

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

Строительно-политехнический колледж

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению лабораторных работ № 1-2
для студентов специальностей

15.02.10 «Мехатроника и мобильная робототехника (по отраслям)»
и 15.02.10 «Мехатроника и робототехника (по отраслям)»
всех форм обучения

УДК 621.38:681.3(07)
ББК 32.97:32.81я7

Составитель В. Н. Коротков

Технологии программирования мехатронных систем: методические указания к проведению лабораторных работ № 1-2 для студентов специальностей 15.02.10 «Мехатроника и мобильная робототехника (по отраслям)» и 15.02.10 «Мехатроника и робототехника (по отраслям)» всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: В. Н. Коротков. — Воронеж: Изд-во ВГТУ, 2025. — 47с.

Методические указания содержат теоретические сведения, необходимые для выполнения лабораторных работ № 1-2. Методические указания разработаны с целью организации и проведения лабораторных занятий по дисциплине «Технологии программирования мехатронных систем».

Предназначены для студентов специальностей 15.02.10 «Мехатроника и мобильная робототехника (по отраслям)» и 15.02.10 «Мехатроника и робототехника (по отраслям)» всех форм обучения.

Подготовлены в электронном виде и содержатся в файле ТПМС_МУ_ЛР1-2.pdf.

Ил. 17. Табл. 12. Библиогр.: 8 назв.

УДК 621.38:681.3(07)
ББК 32.97:32.81я7

Рецензент – В.А. Трубецкой, канд. техн. наук, доц. кафедры электропривода, автоматизации и управления в технических системах ВГТУ

*Издается по решению редакционно-издательского совета
Воронежского государственного технического университета*

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМАХ УПРАВЛЕНИЯ

1.1. Классификация систем управления

Управление роботами и другими мехатронными устройствами и системами осуществляется системой управления. Во взаимодействии с информационной (сенсорной или датчиковой) системой и исполнительными системами она образует систему автоматического управления. Кроме того, через пульт или панель управления роботом может управлять человек-оператор.

Основными способами управления являются программное, адаптивное и интеллектуальное управление [1-3].

Практически только программное управление нашло применение в чистом виде, да и то часто и к нему добавляют элементы адаптации. В целом же все эти три способа управления применяются комплексно. Адаптивное управление строится на базе программного как более высокий уровень управления. Интеллектуальное управление реализуется как надстройка над первыми двумя уровнями. Названия систем управления конкретных роботов обычно определяются основным использованным в ней способом управления.

По степени участия человека-оператора в процессе управления различают системы автоматического, автоматизированного и ручного управления.

По типу движения исполнительных систем существуют системы управления:

- непрерывные (контурные);
- дискретные позиционные (шагами “от точки к точке”);
- дискретные цикловые (по упорам, как правило, с одним шагом по каждой координате).

По управляемым переменным различают системы управления:

- положением (позицией);
- скоростью;
- силой (моментом).

Часто эти способы управления применяют в комбинации либо разные способы по разным координатам, либо с последовательным переходом от одного к другому, либо, в виде функциональной зависимости управляемой переменной от другой (например, управление по силе, величина которой задается как функция от положения).

1.2. Структура и принцип действия цикловой системы программного управления

Цикловое управление имеют практически все пневматические роботы. Также имеются роботы с гидравлическим и электрическим приводом, оснащенные цикловым управлением. Процесс управления отдельными приводами сводится к однократному разгону, движению с постоянной скоростью и торможению при достижении упора. Программирование робота заключается в установке на каждом приводе этих упоров, которые определяют величину переме-

щения по соответствующей степени подвижности, последовательности включений приводов и возможных задержек времени между этими включениями. Все эти операции, кроме установки упоров, проводятся с помощью переключателей или других органов на пульте устройства управления. Вследствие простоты циклового управления для роботов с таким управлением, могут применяться устройства группового управления [1-3].

На рис. 1.1 приведена структура цикловой системы управления роботом.

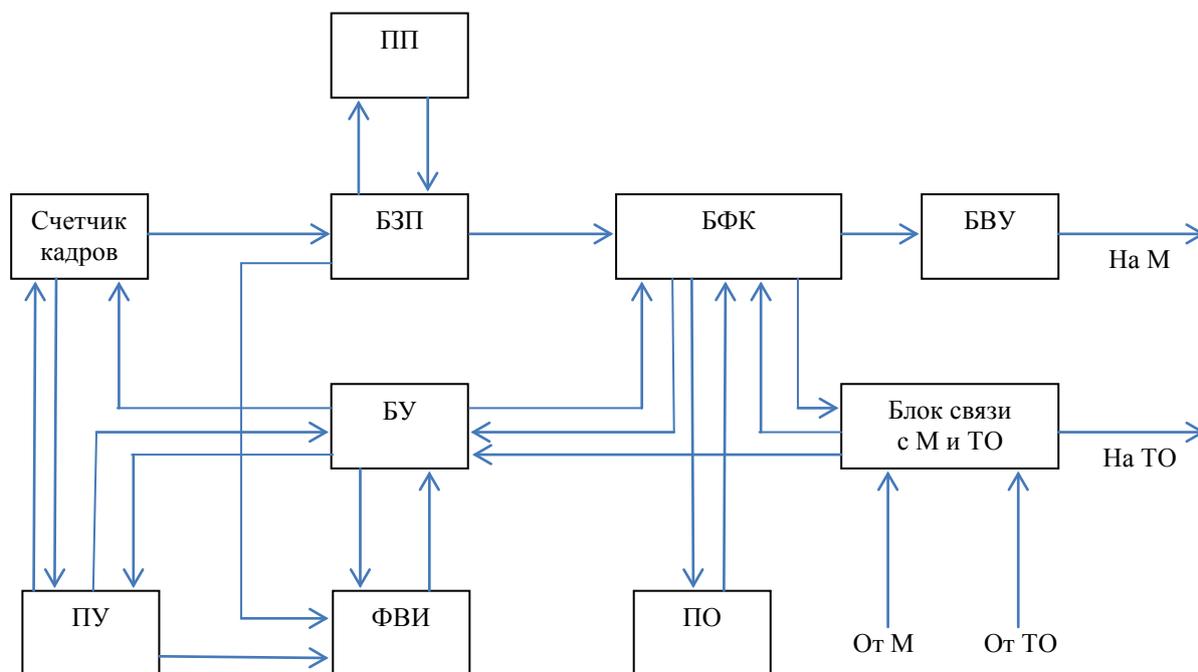


Рис. 1.1. Структура системы циклового программного управления: ПУ – пульт управления; ПП – пульт программирования; БЗП – блок задания программ; БУ – блок управления; ФВИ – формирователь временных импульсов; БФК – блок формирования команд; ПО – пульт обучения; БВУ – блок выходных усилителей; М – манипулятор; ТО – технологическое оборудование

В реальных системах циклового управления роботами пульт управления ПУ, пульт обучения ПО и пульт программирования ПП часто представляли одно устройство. Данная структура системы управления реализовывалась на жесткой логике или на самых простых микропроцессорах. Каждая команда, реализующая одно действие манипулятора, выполняется за один кадр. Полная программа представляет несколько кадров, каждый из которых представлен своей командой.

При функционировании робота подготовленная заранее в процессе его обучения при помощи пульта программирования ПП программа считывается из блока записи программ БЗП в блок управления БУ и выдается в блоки связи с манипулятором и технологическим оборудованием для конкретного формирования сигналов отработки управляющего алгоритма и обмена информацией с обслуживаемыми объектами.

Формирователь временных импульсов формирует тактовые импульсы,

которые, через блок управления БУ, поступают в счетчик кадров.

Счетчик кадров, отсчитав определенное количество импульсов, выдает в блок записи программ БЗП сигнал на смену команды.

Из блока записи программ БЗП на вход блока формирования команд БФК поступает код для формирования очередной команды.

Блок формирования команд БФК формирует на выходах команды для совместно работающего технологического оборудования, которые в виде соответствующих электрических сигналов, через блок связи с манипулятором М и технологическим оборудованием ТО поступают в системы управления технологическим оборудованием. Для управления собственным манипулятором Блок формирования команд БФК выдает сигналы команд на вход блока выходных усилителей БВУ. Блок выходных усилителей БВУ усиливает до необходимого уровня сигналы команд и передает их непосредственно на приводы манипулятора.

Сигналы от датчиков положения и состояние систем манипулятора, а также от систем управления совместно работающего технологического оборудования поступают на входы блока связи с манипулятором и технологическим оборудованием. Исходя из этих сигналов, блок связи с манипулятором и технологическим оборудованием подает сигналы о выполнении текущей команды в блок формирования команд БФК и в блок управления БУ. Блок управления БУ формирует сигнал для смены кадра, который поступает на вход счетчика кадров.

Система управления, построенная по приведенной выше или подобной структуре, является в настоящее время устаревшей. Однако такие системы управления все еще используются на предприятиях, хотя и выходят из употребления.

Выпускаемые в настоящее время системы циклового программного управления строятся на основе программируемых логических контроллеров (ПЛК), которые иначе называются промышленными логическими контроллерами. Сами ПЛК и их программирование будут рассмотрены в одном из следующих разделов. А на рис. 1.2 приведена структура цикловой системы управления на основе ПЛК для работа с тремя степенями подвижности.

Программа для ПЛК разрабатывается и отлаживается на ПЭВМ типа IBM PC/AT в специальной программной среде, которая будет рассмотрена в одном из следующих разделов. Далее готовая программа по интерфейсу связи ИС передается в программируемый логический контроллер ПЛК. Также по интерфейсу связи ИС находящаяся в постоянной памяти программируемого логического контроллера ПЛК программа может быть считана в ПЭВМ для хранения или модификации в специальной программной среде. Интерфейс связи может быть RS-422, RS-485 (с применением специальных преобразователей интерфейса RS-422 – USB, RS-485 – USB), USB и Ethernet. В последнем случае ПЭВМ может управлять сразу несколькими ПЛК, передавая и принимая информацию через коммутаторы и маршрутизаторы локальной вычислительной сети и сети “Интернет”.

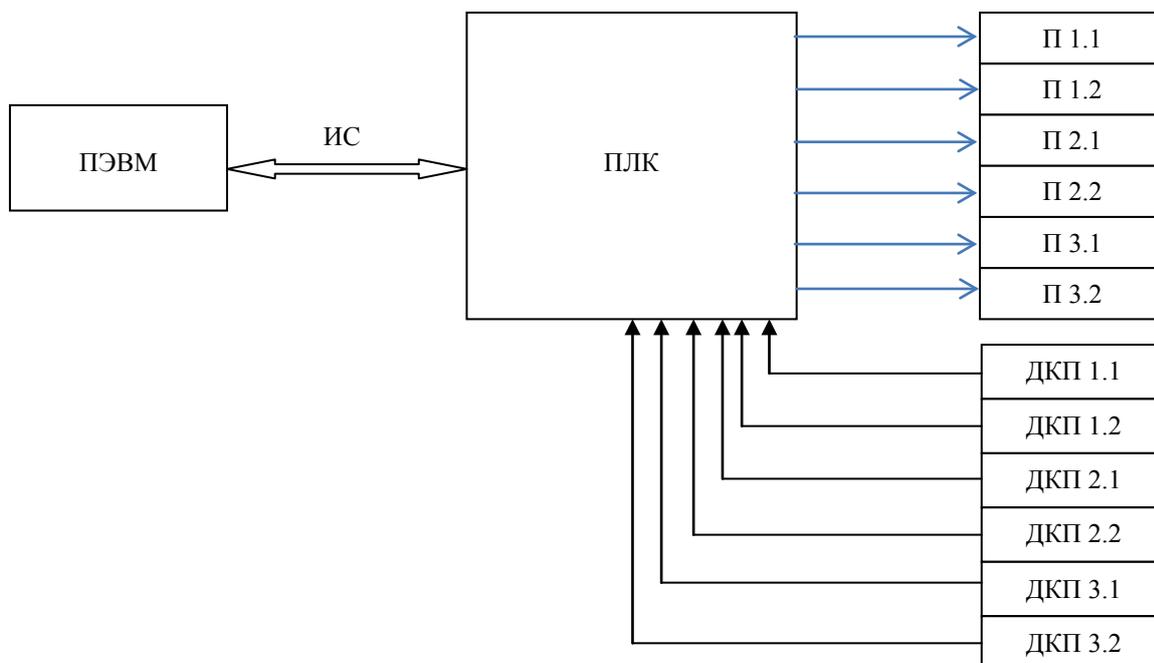


Рис. 1.2. Структура системы циклового программного управления на основе ПЛК: ПЭВМ – персональная электронно-вычислительная машина (как правило типа IBM PC/AT); ИС – интерфейс связи (RS-422, RS-485, USB, Ethernet); ПЛК – программируемый логический контроллер; П1.1 и П1.2 – реверсивный привод степени подвижности № 1; П2.1 и П2.2 – реверсивный привод степени подвижности № 2; П3.1 и П3.2 – реверсивный привод степени подвижности № 3; ДКП1.1 и ДКП1.2 – датчики конечных положений степени подвижности № 1; ДКП2.1 и ДКП2.2 – датчики конечных положений степени подвижности № 2; ДКП3.1 и ДКП3.2 – датчики конечных положений степени подвижности № 3

ПЛК, в соответствии с записанной в его память программой, выдает сигналы управления приводами степеней подвижности робота. Например, подаст сигнал на включение привода П1.1 и, в тоже время, подаст сигнал на выключение привода П1.2 степени подвижности № 1. Это приведет, например, к повороту колонны робота влево до конечного положения.

Затем ПЛК опрашивает состояние датчика конечного положения ДКП1.1 на наличие на его выходе нужного логического сигнала (лог. 0 или лог. 1 – в зависимости от указаний в программе). Как только на выходе датчика конечного положения ДКП1.1 появится нужный логический сигнал, что будет свидетельствовать о нахождении степени подвижности робота в крайнем левом положении, ПЛК снимает сигнал с привода П1.1.

Далее, в соответствии с записанной в ПЛК программой, может быть выдержана нужная временная пауза и включен любой из приводов, кроме П1.1. Также ПЛК по интерфейсу связи ИС может информировать ПЭВМ о выполнении команды и получить новую команду.

Достоинство данной структуры в том, что для управления используется широко распространенная ПЭВМ, а сам ПЛК может управлять не только роботом, но и другим технологическим и бытовым оборудованием после смены управляющей программы, что позволяет выпускать ПЛК в больших количествах.

1.3. Структура и принцип действия позиционной системы программного управления

Типичные роботы с позиционным управлением - промышленные роботы для точечной сварки, сборки и обслуживания различного технологического оборудования.

Эти роботы имеют большое число (от десятков – до сотен) точек позиционирования рабочего органа манипулятора по каждой степени подвижности. В отличие от систем циклового управления здесь точность позиционирования обеспечивается не упорами, а точностью отработки приводами с обратной связью по положению заданных управляющей программой точек позиционирования. Исключение составляют разомкнутые системы на шаговых приводах. Системы программного управления роботом первоначально были заимствованы из систем ЧПУ технологического оборудования, но они существенно сложнее последних, прежде всего из-за большего числа степеней подвижности и их взаимосвязанности. Процесс дискретного позиционного программного управления манипулятором выглядит следующим образом. В устройстве управления хранится управляющая программа, которая состоит из подпрограмм для отдельных приводов каждой степени подвижности. Эти подпрограммы представляют последовательность численных значений шагов позиционирования привода данной степени подвижности.

Отработка управляющей программы заключается в одновременной подаче на все приводы значений очередного шага и отработке приводами этого задания. После того как все приводы остановятся, рабочий орган манипулятора займет соответствующую очередную позицию в пространстве. После этого управляющая программа выдаст команду на выполнение приводами степеней подвижности следующего шага и т.д. В результате рабочий орган манипулятора будет перемещаться шагами по запланированной дискретной траектории.

Программирование, то есть синтез управляющей программы, осуществляется методом обучения на самом роботе или аналитически на ЭВМ. Первый способ программирования, так же в свое время заимствованный у систем ЧПУ технологического оборудования, применительно к манипуляторам имеет два варианта.

В первом варианте оператор в режиме ручного управления отдельными приводами последовательно устанавливает рабочий орган манипулятора в заранее выбранные точки заданной траектории. При этом в каждой такой точке в память устройства управления заносятся значения сигналов с датчиков положения всех степеней подвижности. В результате прохождения всей траектории, в устройстве управления оказывается записанной соответствующая ей управляющая программа. После пробного ее воспроизведения и при необходимости корректировки в отдельных точках программа готова к работе.

Второй вариант программирования методом обучения заключается в перемещении рабочего органа манипулятора рукой оператора и записи при этом показаний датчиков положения приводов степеней подвижности, как в предыдущем варианте. Для выполнения такой операции на рабочем органе предусматриваются специальные ручки, а в конструкции самого манипулятора - возможность отсо-

единения приводов от его механической части, чтобы дать возможность оператору беспрепятственно ее перемещать. Таким образом, этот вариант программирования требует соответствующего изменения конструкции манипулятора.

Аналитический способ программирования позволяет синтезировать управляющие программы на ЭВМ, без использования робота. По существу, в этом случае вместо робота используется его математическая модель, с помощью которой и осуществляется процесс программирования подобно тому, как это делается на реальном роботе. При этом для получения математической модели требуемой точности с робота необходимо регулярно снимать соответствующие характеристики. Такой способ программирования не требует отключения робота на время программирования от технологического процесса, в котором он задействован.

На рис. 1.3 приведена структура позиционной системы управления роботом.

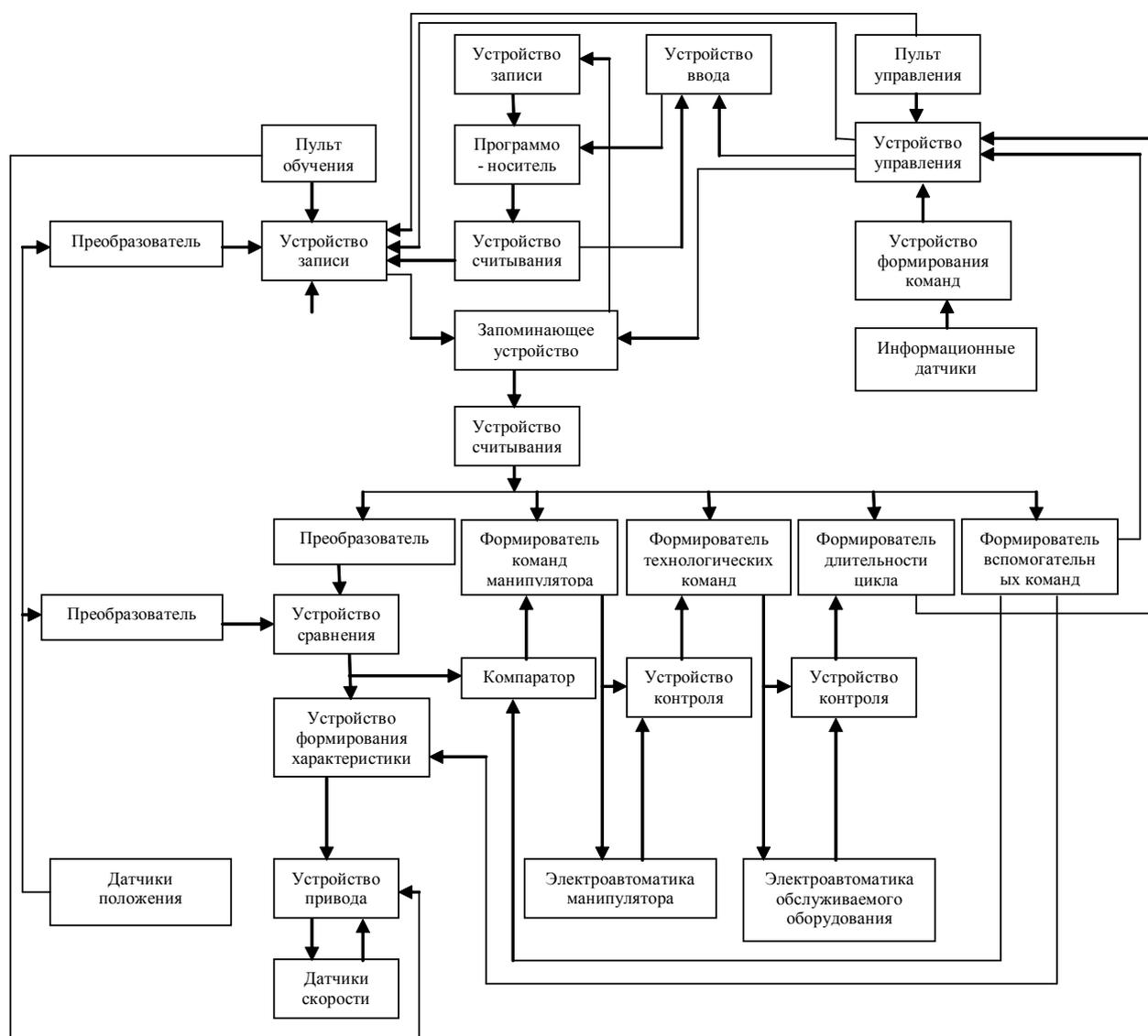


Рис. 1.3. Структура системы позиционного программного управления

В настоящее время системы позиционного управления, построенные по такой структуре, являются устаревшими. Однако, они все еще используются на производстве.

Современной является система позиционного управления, построенная по приведенной на рис. 1.4 структуре.

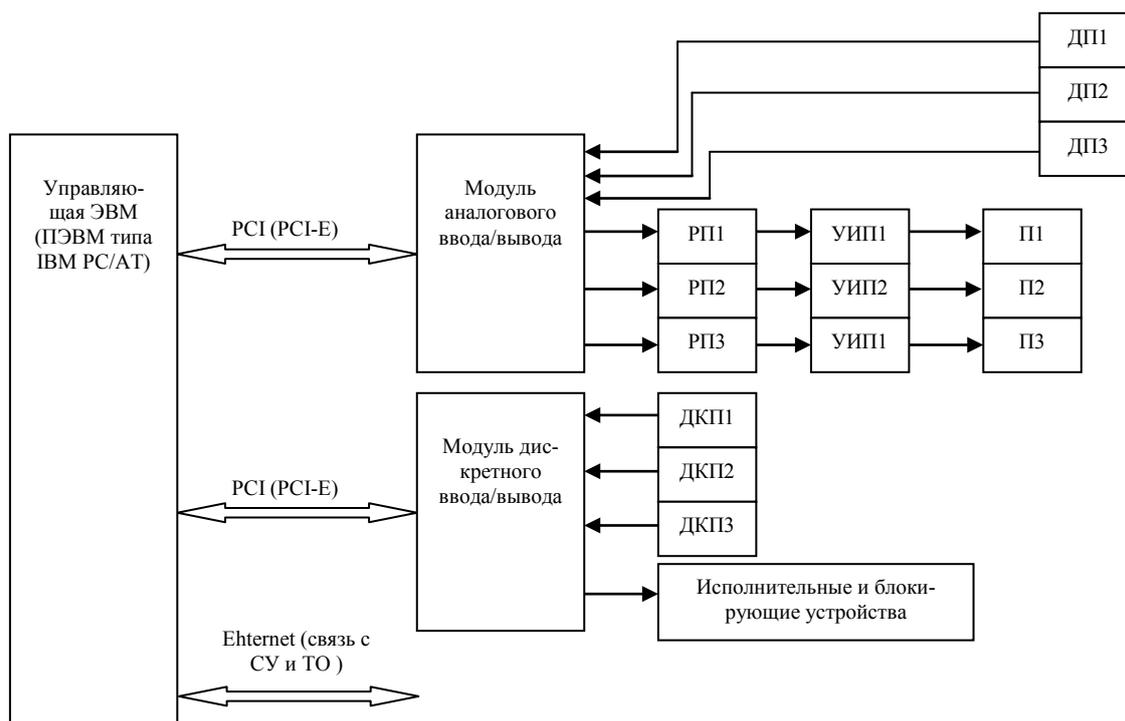


Рис. 1.4. Современная структура системы позиционного программного управления

Здесь основой является управляющая ЭВМ на базе микропроцессоров Intel или AMD в промышленном исполнении или ПЭВМ типа IBM PC/AT. К ней может подключаться специальный пульт управления при помощи одного из стандартных интерфейсов. Также для управления роботом могут применяться штатные средства ввода/вывода в виде клавиатуры, монитора, манипуляторов “мышь” и “джойстик”.

С объектом управления, в качестве которого выступает манипулятор робота, обмен информацией производится через модуль аналогового ввода/вывода и модуль дискретного ввода/вывода. А для связи с системой управления верхнего уровня и с совместно работающим технологическим оборудованием используется интерфейс Ethernet.

В данной структурной схеме предполагается, что регуляторы положения РП1 – РП3 имеют аналоговое управление, а датчики положения ДП1 – ДП3 – потенциометрические с аналоговым выходным сигналом. В случае использования регуляторов положения РП1 – РП3 с цифровым управлением, может использоваться модуль дискретного вывода с требуемой разрядностью или специально спроектированный интерфейсный модуль. В случае же использования импульсных или кодовых датчиков положения ДП1 – ДП3, может использоваться модуль дискретного ввода с требуемой разрядностью или специально спроектированный интерфейсный модуль. В любом случае модуль аналогового (дискретного) ввода/вывода или специальный интерфейсный модуль обмениваются данными с микропроцессором ЭВМ через стандартную шину PCI (устаревая шина) или PCI-E (современная шина).

Управляющая ЭВМ по заданной программе рассчитывает траекторию движения манипулятора и все точки позиционирования для трех (в данном примере) степеней подвижности. Через каналы модуля аналогового ввода/вывода информация о нужных точках позиционирования каждой степени подвижности поступает на входы регуляторов положения степеней подвижности РП1 – РП3. Регуляторы положения РП1 – РП3 рассчитывают управляющее напряжение для управляемых источников питания УИП1 – УИП3 каждой из степеней подвижности. При определении управляющего напряжения учитывается не только величина перемещения каждой степени подвижности, но и направление перемещения.

Управляемые источники питания УИП1 – УИП3 формируют на своих выходах напряжение питания требуемой величины и полярности для двигателей П1 – П3 степеней подвижности.

Двигатели приводов степеней подвижности П1 – П3 приводят в движение степени подвижности манипулятора, что меняет их положение в пространстве и положение друг относительно друга.

Положение каждой из степеней подвижности отслеживается датчиками положения ДП1 – ДП3 степеней подвижности. В конкретные моменты времени информация о положении степеней подвижности считывается управляющей ЭВМ через модуль аналогового ввода/вывода с выходов датчиков положения ДП1 – ДП3.

В зависимости от полученной информации о положении степеней подвижности, управляющая ЭВМ рассчитывает следующие точки позиционирования на траектории движения манипулятора и, через модуль аналогового ввода/вывода, передает информацию о них на входы регуляторов положения РП1 – РП3.

Если степень подвижности манипулятора во время движения достигла одного из конечных своих положений, то это фиксируется одним из датчиков конечных положений ДКП1 – ДКП3. Информация с выходов датчиков ДКП1 – ДКП3 поступает в управляющую ЭВМ через модуль дискретного ввода/вывода.

Через модуль дискретного ввода/вывода управляющая ЭВМ выдает управляющие сигналы на входы управления исполнительных и блокирующих устройств, что, в том числе, предотвращает аварийные ситуации.

1.4. Структура и принцип действия контурной системы программного управления

Типичные роботы с контурным управлением - промышленные роботы для дуговой сварки и резки, для нанесения покрытий. Главное отличие этих роботов от роботов с позиционным управлением в том, что движение по программной траектории осуществляется без остановок и с контролем скорости. Это требует от приводов большего быстродействия и приводит к принципиальному различию их программирования. Если записать управляющую программу для манипулятора с непрерывным управлением методом обучения, перемещая его рабочий орган по требуемой программной траектории на небольшой скорости, а затем воспроизвести эту программу на существенно большей скорости, какая требуется по технологии, то из-за неизбежного динамического запаздывания рабочий орган на всех изгибах траектории будет сходиться с нее. Это ди-

намическая ошибка будет возрастать с увеличением скорости движения. Поэтому управляющие программы при таком методе программирования обучением необходимо корректировать и обрабатывать на заданной реальной скорости, с которой программная траектория должна воспроизводиться. То же относится и к аналитическому программированию, при котором необходима динамическая математическая модель робота.

При программировании систем непрерывного управления методом обучения помимо запоминания непрерывного перемещения приводов нашел применение существенно более простой способ, когда запоминаются только ряд дискретных позиций на программной траектории, а участки траектории между ними формируются при воспроизведении программы с помощью интерполятора в виде стандартных математических функций. Выбор точек на программной траектории производится с учетом кривизны траектории: чем она больше, тем меньше берется расстояние между точками.

Достоинство такого способа программирования в резком сокращении требуемого объема памяти устройства управления, а недостаток - в меньшей точности воспроизведения программной траектории. Аппаратно соответствующее устройство управления представляет собой устройство дискретного позиционного управления, дополненное интерполятором и функцией контроля скорости.

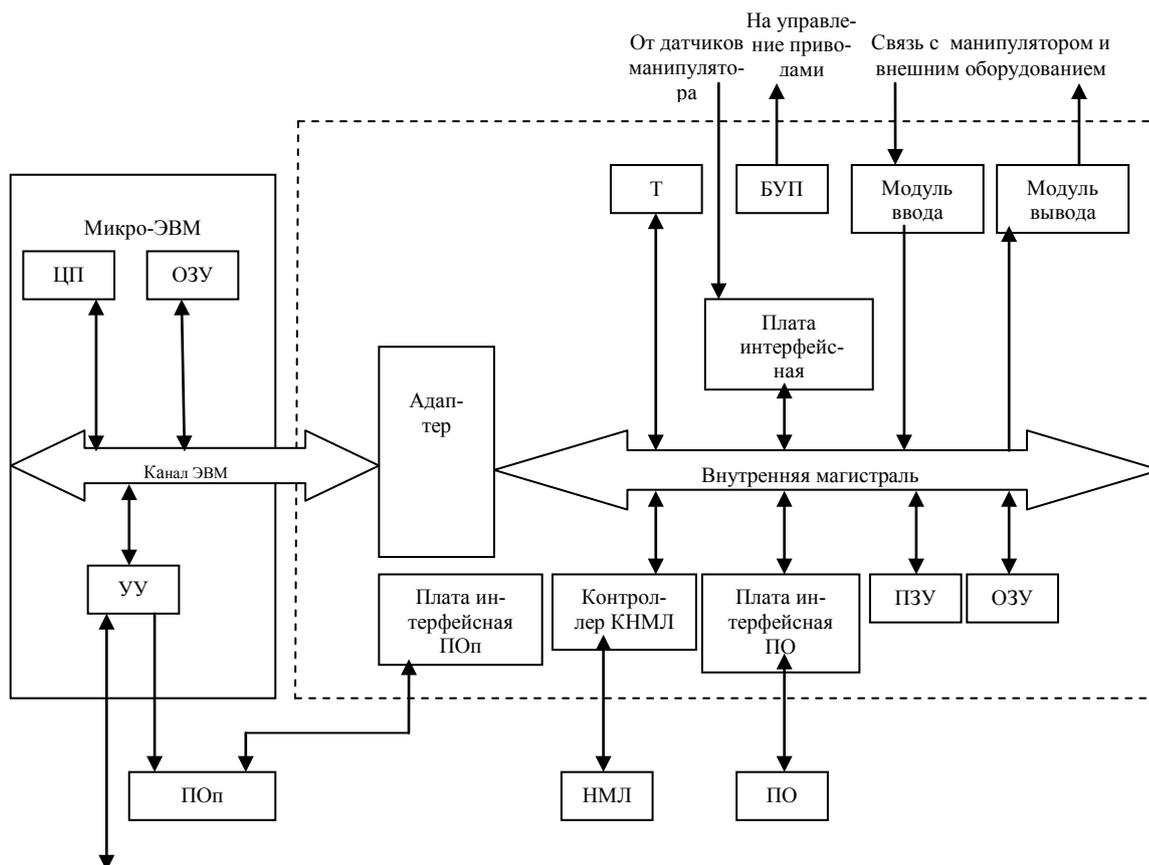


Рис. 1.5. Структура системы контурного управления “УКМ-772”.

ЦП – центральный процессор; ОЗУ – оперативное запоминающее устройство; УУ – управляющее устройство; Т – таймер; БУП – блок управления приводами; ДОС – датчик обратной связи; ПОп – пульт оператора; КНМЛ – кассетный накопитель на магнитной ленте; ПО – пульт обучения; ПЗУ – постоянное запоминающее устройство

На рис. 1.5 представлена структура контурной системы управления на примере системы управления “УКМ-772”, которая в настоящее время является устаревшей, но еще применяется на некоторых промышленных предприятиях.

Система контурного управления УКМ-772 [4] предназначена для управления манипулятором и сварочным оборудованием при автоматизации технологического процесса дуговой сварки в различных отраслях промышленности. Может быть использована для автоматизации окрасочных работ.

Некоторые технические характеристики:

- управление семью следящими электрогидравлическими приводами координатных перемещений;
- выдача на привод управляющего сигнала постоянного напряжения ($+10 \pm 0,5$) ... ($-10 \pm 0,5$) В при токе нагрузки не более 5 мА;
- выдача ряда рабочих скоростей 2,8 - 31,5 мм/с с коэффициентом ряда 1,12;
- точность поддержания контурной скорости - $\pm 5\%$;
- выдача на оборудование шести восьмиразрядных групп технологических команд в виде сигналов постоянного тока не более 0,1 А напряжением ± 24 В;
- приём ответов о выполнении технологических команд в виде сигналов постоянного тока не более 0,03 А напряжением ± 24 В;
- приём сигналов от оборудования на выбор одной из четырёх программ управления в виде сигналов постоянного тока не более 0,03 А напряжением ± 24 В;
- задание десяти режимов работы.

В состав системы входят:

- ЭВМ “Электроника-60”;
- кассетный накопитель КНМЛ “Искра 005-33”
- пульт оператора;
- логический блок;
- аппаратура электропитания и вентиляции.

Нежимы работы:

- “Разметка магнитной ленты”
- “Ввод”;
- “Вывод”;
- “Выбор программы”;
- “Обучение”;
- “Автоматическая работа однократная”;
- “Покадровая отработка”;
- “Поиск кадра”;
- “Контроль”.

Метод программирования – обучение. Носитель программы - кассета МК-60 с магнитной лентой по ГОСТ 20492-75.

Рабочая программа хранится на магнитной ленте. При помощи контроллера кассетного накопителя на магнитной ленте КНМЛ она считывается и, через внутреннюю магистраль, через адаптер, поступает в канал ЭВМ. Центральный процессор ЦП на основе считанной программы выдает данные через канал ЭВМ, через адаптер, через внутреннюю магистраль на блок управления приво-

дами БУП. Блок управления приводами БУП выдает управляющие сигналы на приводы манипулятора, которые перемещаются в заданное по программе положение. От датчиков манипуляторов поступает информация о положении манипулятора и скорости движения звеньев манипулятора в конкретный момент времени. Через плату интерфейсную ДОС, внутреннюю магистраль, адаптер эта информация поступает в канал ЭВМ. Центральный процессор ЦП сравнивает информацию, поступившую от датчиков с заданной по программе, и выдает данные так, как было описано выше.

Задача ЭВМ здесь состоит в формировании управляющей программы в памяти при обучении робота, а также в обработке информации, поступающей с пульта оператора ПОп и от датчиков обратной связи манипулятора, подающих сигналы о фактическом протекании процессов на выходе всей системы. В итоге ЭВМ выдает соответствующие сигналы управления следящим приводам манипулятора и технологическому оборудованию, например сварочному при помощи модуля ввода и модуля вывода.

Во многих роботах с контурной системой управления, когда привод по каждой степени подвижности строится по принципу следящей системы, ставятся аналоговые датчики обратной связи по положению (потенциометрические) и по скорости (тахогенераторы). Для преобразования их выходных сигналов в цифровые применяются аналого-цифровые преобразователи. Большая эффективность системы по быстродействию, точности и плавности движений манипулятора достигается при установке импульсных или кодовых датчиков обратной связи по положению. Тогда, при использовании современных ЭВМ и при использовании современных однокристальных микроконтроллеров получается чисто цифровая система управления приводами. Подобная структура приведена на рис. 1.6.

Здесь основой является управляющая ЭВМ на базе микропроцессоров Intel или AMD в промышленном исполнении или ПЭВМ типа IBM PC/AT. К ней может подключаться специальный пульт управления при помощи одного из стандартных интерфейсов. Также для управления роботом могут применяться штатные средства ввода/вывода в виде клавиатуры, монитора, манипуляторов “мышь” и “джойстик”.

С объектом управления, в качестве которого выступает манипулятор робота, обмен информацией производится через модуль аналогового ввода/вывода и модуль дискретного ввода/вывода. А для связи с системой управления верхнего уровня и с совместно работающим технологическим оборудованием используется интерфейс Ethernet.

Положение исполнительного механизма ИМ1 – ИМ3 каждой из степеней подвижности отслеживается датчиками положения ДП1 – ДП3 (как правило, импульсные или кодовые). Однако информация с их выходов поступает на входы регуляторов положения РП1 – РП3 каждой из степеней подвижности, выполненных на основе однокристальных микроконтроллеров, которые сравнивают заданное в конкретный момент времени положение исполнительного механизма ИМ1 – ИМ3 с текущим. Также регуляторы положения РП1 – РП3 выдают управляющей ЭВМ информацию о положении исполнительного механизма ИМ1 – ИМ3 степеней подвижности.

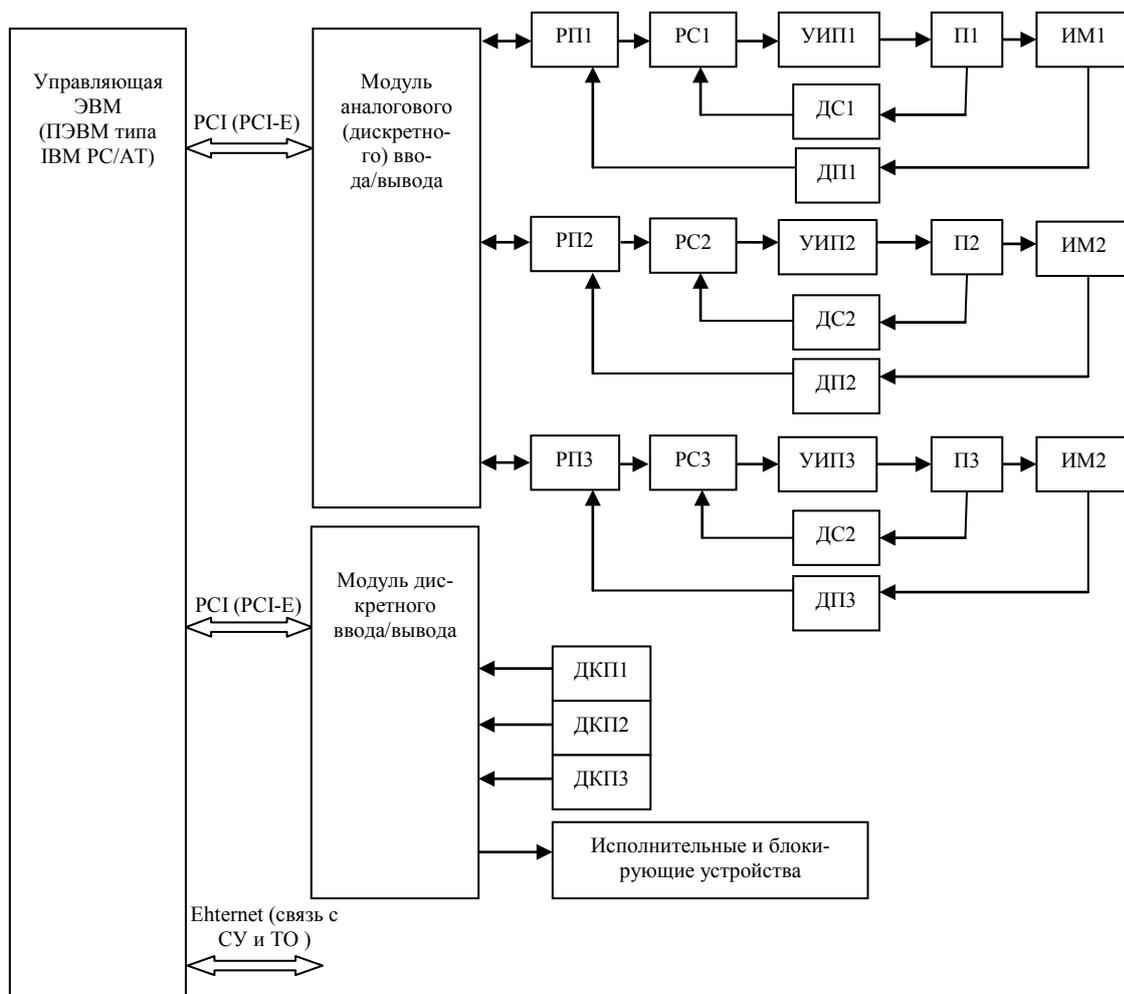


Рис. 1.6. Структура современной системы контурного управления

В зависимости от полученной информации о положении исполнительных механизмов ИМ1 – ИМ3 степеней подвижности, регуляторы положения РП1 – РП3 рассчитывают сигналы задания скоростей перемещения исполнительных механизмов ИМ1 – ИМ3 степеней подвижности, и выдают их на входы регуляторов скорости РС1 – РС3.

Скорость перемещения исполнительного механизма ИМ1 – ИМ3 каждой из степеней подвижности отслеживается датчиками скорости ДС1 – ДС3 (как правило, импульсные или тахогенератор). Информация с их выходов поступает на входы регуляторов скорости РС1 – РС3 каждой из степеней подвижности, выполненных на основе однокристальных микроконтроллеров, которые сравнивают заданную в конкретный момент времени скорость перемещения исполнительного механизма ИМ1 – ИМ3 с текущей.

Регуляторы скорости РС1 – РС3 рассчитывают управляющее напряжение для управляемых источников питания УИП1 – УИП3 каждой из степеней подвижности. При определении управляющего напряжения учитывается не только величина скорости каждого исполнительного механизма ИМ1 – ИМ3 степеней подвижности, но и направление перемещения.

Управляемые источники питания УИП1 – УИП3 формируют на своих выходах напряжение питания требуемой величины и полярности для двигате-

лей П1 – П3 приводов степеней подвижности.

Двигатели приводов степеней подвижности П1 – П3 приводят в движение исполнительные механизмы ИМ1 – ИМ3 степеней подвижности манипулятора, что меняет их положение в пространстве и положение друг относительно друга.

Если степень подвижности манипулятора во время движения достигла одного из конечных своих положений, то это фиксируется одним из датчиков конечных положений ДКП1 – ДКП3. Информация с выходов датчиков ДКП1 – ДКП3 поступает в управляющую ЭВМ через модуль дискретного ввода/вывода.

Через модуль дискретного ввода/вывода управляющая ЭВМ выдает управляющие сигналы на входы управления исполнительных и блокирующих устройств, что, в том числе, предотвращает аварийные ситуации.

1.5. Адаптивное управление роботами

Рассмотренные выше системы программного управления роботами основаны на наиболее простом способе автоматического управления - без обратной связи по фактическому состоянию внешней среды, с которой взаимодействует робот. В связи с этим такие системы применимы только при полностью неизменных на протяжении всего процесса управления внешних условиях работы, а также целях управления и параметрах самого робота. Адаптивное управление осуществляется в функции от параметров внешней среды и поэтому позволяет обеспечить достижение цели управления при непостоянстве или неполной априорной информации об этих параметрах. Примерами простейших задач, которые могут быть решены с помощью адаптивного управления, являются взятие произвольно расположенных или подвижных предметов путем наведения на них захватного устройства манипулятора, выбор и взятие предметов из ряда других по определенным признакам (форма, цвет и т.д.), обход непредвиденных препятствий и т.п. Для осуществления такого управления робот должен быть снабжен сенсорными устройствами, то есть датчиками состояния внешней среды.

При адаптивном управлении максимально используются заранее составленные программы для выполнения тех частей задания, которые могут быть реализованы способом программного управления. Таким образом, в общем случае в системах адаптивного управления используются оба способа управления – программное, и в функции от текущей информации о внешней среде. Примером системы адаптивного управления является система управления промышленным роботом для дуговой сварки. Сам процесс сварки ведется по программе, однако перед этим автоматически осуществляется поиск места стыка свариваемых деталей, затем движение вдоль стыка при определенных ориентации и расстоянии электрода относительно стыка. Эти задачи реализуются с помощью различного типа датчиков угла наклона и расстояния, которые размещены на сварочной головке манипулятора. Другим примером робота с адаптивным управлением является окрасочный робот с простейшей системой технического зрения, которая служит для определения контура очередного окрашиваемого изделия [5, 6].

На рис. 1.7 показана обобщенная структура системы управления оцувст-

вленных роботов, к которым относятся роботы с адаптивным управлением.

Показанная на рис. 1.7 структура включает 5 уровней управления У1 - У5.



Рис. 1.7. Структура системы управления адаптивным роботом

Связь человека-оператора с роботом осуществляется через пульт управления. Оператор выдает роботу задания, контролирует их выполнение и проводит общий контроль за процессом функционирования робота в целом.

Пятый (верхний) уровень автоматического управления У5 анализирует задания, поступающие от оператора или от системы управления верхнего уровня, и определяет последовательность действий робота в соответствии с заданием. На этом уровне анализируется информация о внешней среде, получаемая от сенсорной системы, и синтезируются модели, на базе которых выполняется планирование действия робота. Пятый уровень управления отвечает за функционирование робота как единой системы, обеспечивая реализацию не только основных, профессиональных функций, но и служебных общесистемных задач, которые определяются требованиями к условиям функционирования робота, включая защиту от внешних воздействий и внутренних неполадок, условий безопасности и т.д. Уровень У5 определяет интеллектуальные возможности робота и круг решаемых им задач.

Четвертый уровень управления У4 - это уровень синтеза функционально

законченных сложных действий, в результате которых решается конкретная задача. В соответствии с планом, выработанным для этого на уровне У5, на уровне У4 производится его разбиение на последовательность элементарных типовых операций, которые реализуются нижними уровнями управления. Результатом действия уровня У4 является выдача управлений на уровни У3 и У2. Уровень У4 использует также текущую информацию от сенсорных устройств для оперативной коррекции планов, получаемых с уровня У5.

Третий и второй уровни управления У3 и У2 - это уровни выполнения элементарных операций, на которые могут быть разбиты законченные действия робота. Различие между этими уровнями заключается в том, что на уровне У3 синтезируются адаптивные управления в функции от информации о внешней среде, а на уровне У2 - более простые управления по программе. В связи с этим при синтезе управлений на уровне У3 используются наряду с типовыми программами уровня У2 команды на вход уровня У1 параллельно с управляющими воздействиями с выхода уровня У2. В результате поступившее на вход третьего уровня задание реализуется, во-первых, в виде последовательности типовых программ второго уровня, и во-вторых, в виде совокупности управляющих воздействий непосредственно на отдельные приводы уровня У1. Все эти действия в целом задаются и координируются уровнем У3 в зависимости от текущей информации о внешней среде и состоянии самого робота.

На уровне У2 рассчитываются управляющие воздействия, которые затем поступают на уровень У1, реализующий программное управление приводами.

Нижний уровень управления У1 реализует управление по отдельным степеням подвижности робота и представляет собой систему управления приводами.

Схема системы управления роботом, изображенная на рис. 1.7, является упрощенной. На ней не показаны все прямые связи выходов отдельных уровней управления со входами нижних уровней, кроме ближайшего, а также обратные связи выходов нижних уровней со входами верхних, в том числе информация о завершении отдельных заданий, об аварийных ситуациях и т.п. На схеме не отражены информационные связи отдельных уровней с пультом управления, которые обеспечивают передачу информации о функционировании робота оператору.

Оператор может взаимодействовать с роботом на любом уровне управления, в том числе может выдавать задания непосредственно на уровень У1 путем командного управления каждым приводом отдельно. Такое управление является весьма трудоемким и требует большого навыка. Временное запаздывание в канале связи еще более усложняет работу в этом режиме. В связи с этим к нему прибегают только в тех случаях, когда по каким-либо причинам другие способы управления оказываются неприемлемыми.

При управлении роботом через уровни У2 и У3 оператор заменяет уровень У4, задавая на их входы коды, подлежащих выполнению программно на уровне У2 или адаптивно на уровне У3, элементарных операций, после чего следит за их автоматическим выполнением. Такое управление называется супервизорным. Аналогичным образом оператор может управлять и через уровни У4 и У5, задавая уже не элементарные операции, а более сложные законченные технологические процессы.

Развитием супервизорного способа управления является интерактивное управление, которое включает двухсторонний обмен информацией между человеком и роботом в виде диалога. Робот, получив очередное задание от человека, в свою очередь запрашивает его о необходимых уточнениях или информирует о необходимости откорректировать задание, чтобы сделать его выполнимым. Этот режим управления, таким образом, максимально упрощает функции и уровень умения оператора за счет соответствующего алгоритмического усложнения системы управления робота вплоть до наделения его искусственным интеллектом.

Такая система должна включать не менее трех уровней управления - У1, У2 и У3. Собственно адаптивное управление реализуется уровнем У3 через уровень программного управления У2 или непосредственно воздействуя на уровень У1. В зависимости от степени сложности технологических операций, выполняемых роботом в адаптивном режиме, система адаптивного управления может включать и остальные верхние уровни управления У4 и У5. Однако обязательной принадлежностью эти уровни являются для системы интеллектуального управления.

1.6. Интеллектуальное управление мехатронными системами

Интеллектуальное управление - это следующий, после адаптивного управления, наивысший в отношении алгоритмических возможностей тип управления. Робототехника является одной из областей применения такого управления, которое в теории управления в настоящее время находится в стадии становления.

В теории адаптивного управления, в результате развития от простого к более сложному, стремясь обеспечить автоматическое управление в условиях все большей неопределенности и изменчивости объекта управления и внешней среды, сперва был разработан раздел самонастраивающихся систем управления с автоматической настройкой параметров при неизменной структуре системы, а затем - раздел самоорганизующихся систем управления с автоматическим изменением структуры систем. В рамках самоорганизующихся систем развивалась теория самообучающихся систем управления, посвященная наиболее совершенной форме управления в современной теории управления, которую можно рассматривать как первый шаг к интеллектуальному управлению, поскольку в ее основе лежат идеи формирования целей управления, принятия решений и планирования поведения.

Под интеллектом понимается общая, в основном врожденная познавательная способность, включая умение обрабатывать информацию о внешней среде с построением ее моделей и использовать эту информацию для планирования поведения в условиях заведомой неполноты информации и непредсказуемо изменяющейся внешней обстановки. Принципиальное различие интеллекта искусственного и естественного заключается в том, что первый имитирует естественный интеллект только в части решения определенного типа творческих задач, в то время как естественный интеллект значительно универсальнее

и многограннее, и включает, прежде всего, такие аспекты, как социальную обусловленность, мотивацию и эмоциональность.

К типичным таким задачам искусственного интеллекта относятся, например, игровые и другие подобные задачи, где невозможен полный перебор вариантов, доказательство теорем, перевод с одного языка на другой. Фундаментальной проблемой искусственного интеллекта является создание модели человеческого мозга и разгадка процесса его мышления.

В создании искусственного интеллекта можно выделить два принципиально различных подхода. Первый подход заключается в разработке общего теоретического решения проблемы на основе адекватного математического аппарата. Второй подход близок к эволюционному пути развития естественного интеллекта и заключается в постепенном накоплении частных эвристических находок для решения отдельных конкретных практических задач. В области робототехники на сегодня наиболее перспективным представляется именно такое последовательное накопление решений конкретных задач.

Естественный интеллект - это способность приобретать и целенаправленно использовать знания. Это значит, что интеллект предполагает наличие хорошо организованной памяти. Человеческие знания, как и сам окружающий мир, нечетки и зачастую противоречивы, хотя, содержат вполне однозначные четкие фрагменты.

Соответственно программа создания искусственного интеллекта включает два аспекта - программно-алгоритмический и аппаратный.

Первый аспект заключается в создании баз знаний на языке высокого уровня, близком естественному, и алгоритмов работы с ними, основанных на нечетких представлениях и параллельной обработке. Это требует принципиально нового аппаратного обеспечения, которое, в отличие от современных фон-неймановских компьютеров, должно быть подобно нейронным структурам живых организмов. Сегодня интеллектуальные системы строятся как биотехнические системы, включая системы интеллектуального управления, и пока обладают весьма ограниченными интеллектуальными возможностями. В них используют теорию нечетких множеств и нечеткой логики, различные эвристические алгоритмы и технологии экспертных систем, ассоциативной памяти и технических нейронных сетей. Ведутся работы по созданию так называемых нечетких компьютеров, которые оперируют нечеткими данными и выводами. Это требует создания новой нечеткой элементной базы и соответствующего программного обеспечения.

Главная сфера применения интеллектуального управления - сложные и большие объекты и системы, для которых доступно описание только на семиотическом уровне. К ним, прежде всего, относятся биотехнические системы, включающие человека. Поскольку такие системы обладают естественным интеллектом, управление ими может быть так же только интеллектуальным. Вместе с тем интеллектуальное управление может потребоваться и для достаточно простых объектов, если с их помощью решаются интеллектуальные задачи или если сама задача управления ими требует интеллектуального подхода в силу сложности внешних условий. В робототехнике искусственный интеллект может

потребоваться для решения следующих задач:

- обработка сенсорной информации (фильтрация, сжатие информации, распознавание образов);
- создание моделей внешней среды;
- планирование поведения;
- управление движением;
- создание интеллектуального интерфейса между оператором и роботом.

Между системами интеллектуального и адаптивного управления нет резкой границы. Интеллектуальные системы являются результатом развития адаптивных систем в направлении расширения возможностей автономного выполнения все более сложных заданий во все более неопределенной среде и при все большей неполноте информации, требующейся для выполнения этих заданий. В ходе этой эволюции к настоящему времени созданы пока только адаптивные системы с некоторыми элементами искусственного интеллекта в виде способности воспринимать и анализировать достаточно сложную и изменяющуюся внешнюю среду и принимать адекватные решения по поведению. Для качественного скачка в направлении создания полноценных интеллектуальных систем требуется, прежде всего, новое аппаратное обеспечение на принципиально новой элементной базе.

Повышение уровня искусственного интеллекта связано, прежде всего, с развитием иерархической структуры моделей среды путем формирования все более обобщенных, более абстрактных уровней ее представления.

Соответственно будет развиваться и иерархия в системах, решающих перечисленные выше задачи путем перехода от образов внешней среды, непосредственно воспринимаемых сенсорами системы, ко все более абстрактным образам. Следствием этого будет расширение функциональных возможностей робота благодаря возможности автономного решения все более сложных неалгоритмируемых интеллектуальных задач, включая самоусовершенствование в процессе активного взаимодействия с внешней средой при решении конкретных задач.

Одним из наиболее обобщенных типов моделей среды являются логико-лингвистические модели. Они применяются для наиболее сложных объектов с неоднозначной реакцией на одни и те же ситуации, которые не могут быть описаны формально математически и поэтому описываются эвристически на основе экспертных оценок на языке, близком естественному. Методы искусственного интеллекта могут применяться в системах управления не только в общесистемном контуре управления, то есть для формирования управления, но и для решения различных локальных задач отдельных подсистем робота. В этом случае процесс управления роботом, естественно, не становится интеллектуальным, подобно тому, как в системах адаптивного управления наличие внутренней адаптации в отдельных частях системы не делает управление в целом адаптивным. В этом смысле следует отличать термин “система интеллектуального управления” от более общего термина “интеллектуальная система”, который относится к системам, использующим искусственный интеллект для решения любых задач.

На рис. 1.8 показана обобщенная схема системы интеллектуального управления роботом, которая представляет собой конкретизацию общей схемы управления оцувствленного робота в части применения искусственного интеллекта при решении перечисленных выше пяти задач.

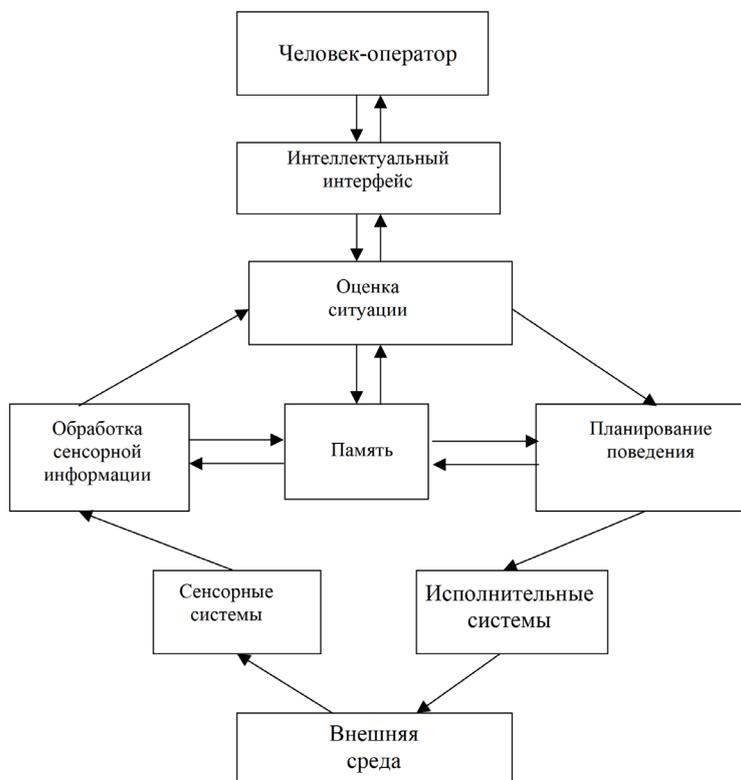


Рис.1.8. Структурная схема системы интеллектуального управления роботом

В центре схемы находится блок памяти, двусторонне связанный с другими системами, перерабатывающими информацию. В этот блок входит база знаний о внешней среде - иерархическая модель внешней среды и база данных, как о внешней среде, так и о самом роботе и об операциях, которые он должен выполнять. Кроме того, специализированные оперативные базы знаний и данных, связанные с центральной памятью, могут иметься в отдельных системах робота.

База знаний о внешней среде содержит как априорную информацию, как вводимую до начала работы, так и оперативную сенсорную, которая приобретается в процессе восприятия окружающей среды при выполнении роботом заданных действий, а так же в процессе его специальных познавательных действий для изучения этой среды. База знаний о внешней среде содержит так же правила, позволяющие моделировать возможные изменения этой среды. Сами знания представляются в виде логических и сетевых моделей среды. Логические модели основаны на аппарате математической логики. Модель строится из системы базовых элементов и системы правил и аксиом.

Все другие блоки схемы имеют иерархическую структуру, уровни которой соединены друг с другом по вертикали снизу вверх. В свою очередь показанные на схеме соединения блоков осуществляются многоканально между одноименными уровнями по горизонтали.

Блок обработки сенсорной информации получает из блока памяти экстраполяцию изменения состояния внешней среды, а передает в него коррекцию этого состояния на уровне непосредственной сенсорной картины среды.

Блок оценки ситуации и блок планирования поведения получают из блока памяти текущую модель внешней среды, а передают в него соответственно ее оценку по определенным критериям и синтезированный план управления движением робота. При синтезе этого плана применяются различные способы решения задач, разработанные в рамках искусственного интеллекта, в том числе:

- поиск решения в пространстве состояний путем нахождения последовательности преобразования исходного состояния в конечное целевое;
- сведением задачи к подзадачам путем последовательного разбиения задачи на подзадачи вплоть до элементарных, решение которых известно;
- поиск в форме решения теоремы путем формулирования задачи как теоремы и ее решения (доказательства) на базе системы аксиом.

Блок интеллектуального интерфейса в общем случае должен быть двусторонне связан со всеми перечисленными выше функциональными блоками.

Обратим внимание на ту особенность рассмотренной обобщенной схемы системы интеллектуального управления, что в ней отсутствует в явном виде блок, ответственный за реализацию способа интеллектуального управления, как это имеет место для адаптивного и программного управления. Объясняется это тем, что искусственный интеллект распределен по всем функциональным блокам схемы в соответствии с перечисленными выше функциями, при реализации которых он может требоваться. В конкретных системах он может присутствовать в любом из этих блоков.

В настоящее время применение искусственного интеллекта в системах управления роботами начинается с введения элементов искусственного интеллекта в системы адаптивного управления на основе применения перечисленных выше интеллектуальных технологий.

1.7. Управление средствами передвижения роботов

Управление передвижением мобильных роботов - это чисто транспортная задача, которая не имеет принципиальной специфики применительно к робототехнике. Это относится и к самой системе передвижения роботов, за исключением маломощных транспортных машин, которые в силу близости их педипуляторов к манипуляторам традиционно являются специальным разделом робототехники.

Рассмотренные схемы адаптивного и интеллектуального управления в равной мере относятся к управлению приводами, как манипуляторов, так и средств передвижения роботов. Однако, последний случай имеет свою специфику, связанную с особенностями зоны передвижения по сравнению с рабочей зоной манипулятора, особенно если задача состоит в передвижении по заранее неизвестной, неподготовленной трассе и на значительные расстояния.

В случае мобильного робота, передвигающегося по произвольной местности, модель этой местности представляет карту местности, которая помимо ап-

риорных сведений составляется и уточняется в ходе движения на основе сенсорной информации. Эта модель должна иметь как минимум два уровня по масштабу: один в пределах досягаемости сенсорных систем и второй для местности непосредственно перед роботом. Первая модель местности служит для прокладки маршрута движения в соответствии с заданной целью. Варианты целей:

- поиск конкретных объектов по заданным признакам;
- достижение определенной точки на местности, заданной координатами;
- движение по заданному маршруту для выполнения определенной технологической операции.

Вторая, более подробная, модель ближайшего участка местности необходима для уточнения трассы непосредственно перед роботом, включая обход мелких и скрытых препятствий, неотмеченных на общей карте местности первого уровня, и обеспечения безопасности движения.

Однако для гарантированного решения последней задачи этого может оказаться недостаточно и тогда применяется специальная система обеспечения безопасности, которая контролирует углы наклона шасси робота, не допуская превышения их критических значений, определяемых условиями опрокидывания, опасные вертикальные провалы и трещины непосредственно перед роботом, оценивает свойства грунта в отношении его проходимости, появление подвижных препятствий, обеспечивает аварийную остановку робота по сигналу дистанционного или контактного датчика препятствий непосредственно перед роботом.

На рис. 1.9 приведена типовая схема системы управления движением мобильного робота.

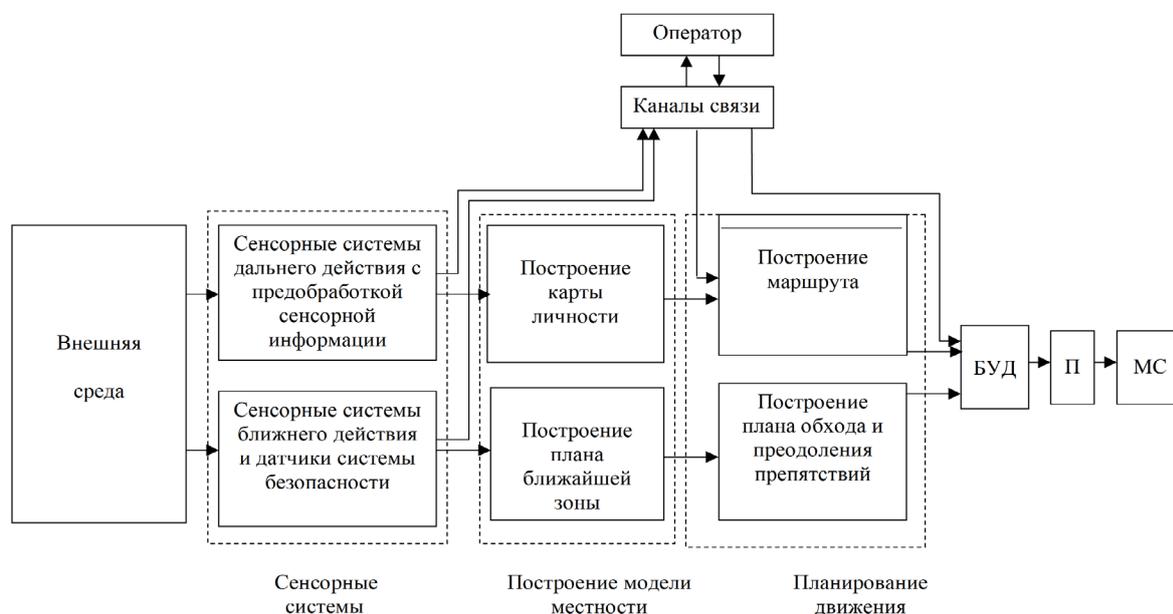


Рис. 1.9. Функциональная схема системы управления движением мобильного робота

Блок управления движением БУД управляет приводами П и приводами поворота шасси. Блок построения маршрута синтезирует траекторию движения, оптимизируя ее, по минимуму расхода энергии, что особенно важно для робо-

тов с автономным энергопитанием, или по минимуму времени выполнения задания, когда в постановку задачи входит условие обеспечения максимального быстродействия. Блок построения карты местности представляет ее в форме, удобной для решения задачи выбора маршрута, в частности, с выделением непреодолимых препятствий и опасных или неясных участков. Если выполнение задачи требует знания большего участка местности, чем дают сенсорные системы в начальном положении робота, то есть до начала движения, карта местности формируется и передается в блок построения маршрута фрагментами по мере передвижения робота, или сразу целиком.

Таковыми же участками последовательно осуществляется и синтез траектории движения. В этом случае первое приближение для всей траектории делается по имеющейся априорной информации о местности или, как минимум, определяется только общее направление движения.

На нижнем общесистемном уровне системы управления находится канал управления движением в ближней зоне, включающий наиболее детальную модель этой зоны и реализующий алгоритмы обеспечения безопасности движения.

Требуемый уровень адаптации и искусственного интеллекта системы определяется степенью неопределенности и сложности местности и характером подлежащих выполнению заданий. При этом основное значение имеет минимально необходимая степень автономности управления роботом, определяемая перечнем его действий, которые должны выполняться без участия оператора. При этом учитывается возможность временной потери связи с оператором, ограниченная пропускная способность каналов связи, неполнота и ограниченная достоверность получаемой оператором от робота информации и ее задержка во времени. Например, если из-за высокого уровня помех в канале передачи оператору не видно изображения панорамы местности перед роботом оператор только эпизодически получает эту информацию, он имеет возможность вмешиваться в автономное движение робота только в порядке контроля и корректировки маршрута движения путем целеуказания отдельных промежуточных точек трассы в моменты ее достоверного наблюдения.

2. ЛАБОРАТОРНАЯ РАБОТА № 1 АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ

2.1. Теоретические сведения о системах счисления

Система счисления - совокупность приемов и правил наименования и обозначения чисел, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов [7].

В любой системе счисления выбирается алфавит, представляющий совокупность символов (букв или цифр), с помощью которого в результате каких-либо операций можно представить любое их количество. Изображение любого количества символов называется числом, а символы алфавита - буквами и циф-

рами. Символы алфавита должны быть разными и значение каждого из них должно быть известно.

В современном мире наиболее распространенной является десятичная система счисления, происхождение которой связано с пальцевым счетом. Она возникла в Индии в XIII в., и была перенесена в Европу арабами. Поэтому десятичную систему счисления стали называть арабской, а используемые для записи чисел цифры, которыми мы теперь пользуемся, - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 - арабскими.

С давних времен для подсчетов и вычислений применялись различные системы счисления. Например, на Древнем Востоке довольно широко была распространена двенадцатеричная система. Благодаря ей число 12 называется дюжиной, а многие предметы и сейчас считают дюжинами. Эта система счисления сохранилась в английской системе мер (например, 1 фут = 12 дюймов) и в денежной системе (1 шиллинг равен 12 пенсам).

В Древнем Вавилоне существовала весьма сложная 60-ричная система. Она, как и 12-ричная система, в какой-то степени сохранилась и до наших дней (в системе измерения времени: 1 ч = 60 мин, 1 мин = 60 с, аналогично в системе измерения углов: $1^\circ = 60'$, $1' = 60''$).

Первые цифры (знаки для обозначения чисел) появились у египтян и вавилонцев. У ряда народов (древние греки, сирийцы, финикийцы) цифрами служили буквы алфавита. Аналогичная система до XVI в. применялась и в России. В Средние века в Европе пользовались системой римских цифр, которые и сейчас часто применяют для обозначения глав, частей, разделов в различного рода документах, книгах, для обозначения месяцев и т. д.

Все системы счисления можно разделить на позиционные и непозиционные.

Непозиционная система счисления - система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения (позиции) в изображении числа. Запись числа A в непозиционной системе счисления D может быть представлена выражением

$$A_D = D_1 + D_2 + \dots + D_N = \sum_{i=1}^N D_i \quad (2.1)$$

где A_D - запись числа A в системе счисления D ;

D_i - символы системы.

Непозиционной системой счисления является самая простая система с одним символом (палочкой). Для изображения какого-либо числа в этой системе надо записать количество палочек, равное данному числу. Например, запись числа 12 в такой системе счисления будет иметь вид: IIIIIIIII. Эта система неэффективна, так как форма записи очень громоздка.

К непозиционной системе счисления относится и римская, символы алфавита которой и обозначаемое ими количество представлены в табл. 2.1.

Таблица 2.1
Римская система счисления

1	2	3	4	5	6	7	8
Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

Запись чисел в этой системе осуществляется по следующим правилам:

- если цифра слева меньше, чем справа, то левая цифра вычитается из правой (IV: $1 < 5$, следовательно, $5 - 1 = 4$, XL: $10 < 50$, следовательно, $50 - 10 = 40$);
- если цифра справа меньше или равна цифре слева, то эти цифры складываются (VI: $5 + 1 = 6$, VIII: $5 + 1 + 1 + 1 = 8$, XX: $10 + 10 = 20$).

Так, число 1964 в римской системе счисления имеет вид MCMLXIV (M – 1000, CM – 900, LX – 60, IV – 4), здесь "девятьсот" получается посредством вычитания из "тысячи" числа "сто", "шестьдесят" - посредством сложения "пятидесяти" и "десяти", "четыре" - посредством вычитания из "пяти" "единицы".

В общем случае непозиционные системы счисления характеризуются сложными способами записи чисел и правилами выполнения арифметических операций.

В настоящее время все наиболее распространенные системы счисления относятся к разряду позиционных.

Систему счисления, в которой значение цифры определяется ее местоположением (позицией) в изображении числа, называют позиционной.

Упорядоченный набор символов (букв и цифр) $\{a_0, a_1, \dots, a_n\}$, используемый для представления любых чисел в заданной позиционной системе счисления, называют ее алфавитом, число символов (цифр) алфавита $p = n + 1$ - ее основанием, а саму систему счисления называют p -ричной.

Основание позиционной системы счисления - количество различных цифр, используемых для изображения чисел в данной системе счисления.

Самой привычной для нас является десятичная система счисления. Ее алфавит – $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, а основание $p = 10$, то есть в этой системе для записи любых чисел используется только десять разных символов (цифр). Эти цифры введены для обозначения первых десяти последовательных чисел, а все последующие числа, начиная с 10 и т. д., обозначаются уже без использования новых цифр. Десятичная система счисления основана на том, что 10 единиц каждого разряда объединяются в одну единицу соседнего старшего разряда, поэтому каждый разряд имеет вес, равный степени 10. Следовательно, значение одной и той же цифры определяется ее местоположением в изображении числа, характеризуемым степенью числа 10. Например, в изображении числа 222,22 цифра 2 повторяется 5 раз, при этом первая слева цифра 2 означает количество сотен (ее вес равен 10^2); вторая - количество десятков (ее вес равен 10), третья - количество единиц (ее вес равен 10^0), четвертая - количество десятых долей единицы (ее вес равен 10^{-1}) и пятая цифра - количество сотых долей единицы (ее вес равен 10^{-2}), то есть число 222,22 может быть разложено по степеням числа 10:

$$222,22 = 2 \times 10^2 + 2 \times 10^1 + 2 \times 10^0 + 2 \times 10^{-1} + 2 \times 10^{-2}$$

Аналогично

$$725 = 7 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

$$1304,5 = 1 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

$$50328,15 = 5 \times 10^4 + 0 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 + 1 \times 10^{-1} + 5 \times 10^{-2}$$

Таким образом, любое число A можно представить в виде полинома путем разложения его по степеням числа 10:

$$A_{10} = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-m} \times 10^{-m} + \dots \quad (2.2)$$

последовательность из коэффициентов которого представляет собой десятичную запись числа A_{10} :

$$A_{10} = a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m} \quad (2.3)$$

Запятая, отделяющая целую часть числа от дробной, служит для фиксации конкретных значений каждой позиции в этой последовательности цифр и является началом отсчета.

2.2. Теоретические сведения о двоичной, восьмеричной и шестнадцатеричной системах счисления

Примеры изображения чисел в данных системах счисления представлены в табл. 2.2.

Таблица 2.2

Представление чисел в разных системах счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
1	2	3	4
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9

10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

В современной вычислительной технике, в устройствах автоматики и связи используется в основном двоичная система счисления, что обусловлено рядом преимуществ данной системы счисления перед другими системами. Так, для ее реализации нужны технические устройства лишь с двумя устойчивыми состояниями, например, материал намагничен или размагничен (магнитные ленты, диски), отверстие есть или отсутствует (перфолента и перфокарта). Этот метод обеспечивает более надежное и помехоустойчивое представление информации, дает возможность применения аппарата булевой алгебры для выполнения логических преобразований информации. Кроме того, арифметические операции в двоичной системе счисления выполняются наиболее просто.

Недостаток двоичной системы - быстрый рост числа разрядов, необходимых для записи больших чисел. Этот недостаток не имеет существенного значения для ЭВМ. Если же возникает необходимость кодировать информацию вручную, например, при составлении программы на машинном языке, то раньше использовали восьмеричную, а в настоящее время используют шестнадцатеричную систему счисления. Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в 3 (восьмеричная) и в 4 (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (числа 8 и 16 - соответственно 3-я и 4-я степени числа 2), а перевод их в двоичную систему счисления и обратно осуществляется гораздо проще, в сравнении с десятичной системой счисления.

Правила выполнения арифметических действий над двоичными числами задаются таблицей сложения, вычитания и умножения (табл. 2.3).

Таблица 2.3

Правила выполнения арифметических операций над двоичными числами

Сложение	Вычитание	Умножение
1	2	3
$0+0=0$	$0-0=0$	$0*0=0$
$0+1=1$	$1-0=1$	$0*1=0$
$1+0=1$	$1-1=0$	$1*0=0$
$1+1=10$	$10-1=1$	$1*1=1$

Правила арифметики во всех позиционных системах счисления аналогичны. В двоичной системе счисления арифметическое сложение происходит по правилу сложения по модулю два с учетом переноса единицы в старший разряд.

Пример. Выполнить операцию арифметического сложения в двоичной системе счисления чисел 13 и 7.

$$13_{10} = 1101_2 \quad 7_{10} = 0111_2$$

$$\text{Решение: } 13 + 7 = 20_{10} \quad 01101 + 0111 = 10100_2$$

При сложении двух единиц результат операции равен нулю и единица переносится в соседний разряд.

Пример. Выполнить операцию арифметического вычитания в двоичной системе счисления чисел 12 и 7.

$$\text{Решение: } 12 - 7 = 5_{10} \quad 1100 - 0111 = 0101_2$$

При вычитании из нулевого разряда в данном разряде образуются две единицы, а в соседних нулевых разрядах возникает единица.

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

Далее рассматривается один из способов перевода числа из двоичной системы счисления в десятичную, который осуществляется следующим образом:

- шаг № 1. Каждый разряд представляется числом 2 в степени n . При этом n – означает номер разряда от нулевого до максимального (например, при 8-ми разрядах n будет находиться в диапазоне от 0 до 7).

- шаг № 2. Для восьми двоичных разрядов представляется: $2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$.

- шаг № 3. Имеется восьмиразрядное число 10011101 в двоичной системе счисления.

- шаг № 4. Для его перевода в десятичную систему счисления делается следующее:

- шаг № 5. В седьмом разряде имеется единица, значит возводится 2 в степень 7 и получается число 128.

- шаг № 6. В шестом разряде имеется ноль, значит возводить 2 в степень 6 не требуется.

- шаг № 7. В пятом разряде имеется ноль, значит возводить 2 в степень 5 не требуется.

- шаг № 8. В четвертом разряде имеется единица, значит возводится 2 в степень 4 и получается число 16.

- шаг № 9. В третьем разряде имеется единица, значит возводится 2 в степень 3 и получается число 8.

- шаг № 10. Во втором разряде имеется единица, значит возводится 2 в степень 2 и получается число 4.

- шаг № 11. В первом разряде имеется ноль, значит возводить 2 в степень 1 не требуется.

- шаг № 12. В нулевом разряде имеется единица, значит возводится 2 в степень 0 и получается 1.

- шаг № 13. Складываются числа, полученные на шагах 5, 8, 9, 10, 12 и получается число: $128 + 16 + 8 + 4 + 1 = 157$.

Перевод двоичного числа 10011101 в шестнадцатиричное производится

следующим образом:

- шаг № 1. Разбивается двоичное число 10011101 на группы по 4 цифры (разряда) справа налево и получается число 1001 1101 = (1001) (1101).

- шаг № 2. Внутри каждой полученной группы производится умножение каждой цифры на 2^n , где n - номер разряда от нулевого до третьего, и получается $(1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0) (1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0)$.

- шаг № 3. Производятся математические действия в скобках и получается $(1 * 8 + 0 * 4 + 0 * 2 + 1 * 1) (1 * 8 + 1 * 4 + 0 * 2 + 1 * 1) = (8 + 0 + 0 + 1) (8 + 4 + 0 + 1)$.

- шаг № 4. Производится сложение чисел в скобках и получается (9) (13).

- шаг № 5. В правых скобках число больше 9, поэтому оно заменяется буквой D.

- шаг № 6. В результате получается число 9D.

В соответствии с табл. 2.2, шестнадцатиричное число 9D, равное двоичному числу 10011101, равно десятичному числу 157.

Перевод десятичных чисел в двоичные осуществляется следующим образом:

- шаг № 1. Имеется десятичное число 194 и имеются 8 двоичных разрядов с номерами от 0 до 7 (справа налево).

- шаг № 2. Возвести 2 в степень 7 и подучить число 128, которое меньше заданного числа 194.

- шаг № 3. Присвоить 7-му двоичному разряду единицу.

- шаг № 4. Вычесть из числа 194 число 128 и получить число 66.

- шаг № 5. Возвести 2 в степень 6 и получить число 64, которое меньше полученного на предыдущем шаге числа 66.

- шаг № 6. Присвоить 6-му двоичному разряду единицу.

- шаг № 7. Вычесть из числа 66 число 64 и получить число 2.

- шаг № 8. Возвести 2 в степень 5 и получить число 32, которое больше полученного на шаге № 7 числа 2.

- шаг № 9. Присвоить 5-му двоичному разряду ноль.

- шаг № 10. Возвести 2 в степень 4 и получить число 16, которое больше полученного на шаге № 7 числа 2.

- шаг № 11. Присвоить 4-му двоичному разряду ноль.

- шаг № 12. Возвести 2 в степень 3 и получить число 8, которое больше полученного на шаге № 7 числа 2.

- шаг № 13. Присвоить 3-му двоичному разряду ноль.

- шаг № 14. Возвести 2 в степень 2 и получить число 4, которое больше полученного на шаге № 7 числа 2.

- шаг № 15. Присвоить 2-му двоичному разряду ноль.

- шаг № 16. Возвести 2 в степень 1 и получить число 2, равное полученному на шаге № 7 числу 2.

- шаг № 17. Присвоить 1-му двоичному разряду единицу.

- шаг № 18. Вычесть из числа 2, полученного на шаге № 7, число 2, полученное на шаге № 16 и получить число 0.

- шаг № 19. Возвести 2 в степень 0 и получить число 1, которое больше полученного на шаге № 18 числа 0.

- шаг № 20. Присвоить 0-му двоичному разряду ноль.

Таким образом, можно получить двоичное число 11000010, которое, согласно табл. 2.1, равно десятичному числу 194.

2.3. Цель лабораторной работы

Целью лабораторной работы является получение и закрепление навыков перевода чисел из одной системы счисления в другую.

2.4. Задание на лабораторную работу

Получить у преподавателя вариант задания, где должно быть указано:

- числа для перевода из десятичной системы счисления в двоичную;
- числа для перевода из двоичной системы счисления в десятичную;
- числа для перевода из десятичной системы счисления в шестнадцатиричную;
- числа для перевода из шестнадцатиричной системы счисления в десятичную;
- числа для перевода из шестнадцатиричной системы счисления в двоичную;
- числа для перевода из двоичной системы счисления в шестнадцатиричную;
- задачи на арифметические действия с числами в двоичной системе счисления;
- задачи на арифметические действия с числами в шестнадцатиричной системе счисления.

2.5. Выполнение лабораторной работы

В ходе выполнения работы требуется, используя приведенные теоретические сведения решить поставленные задачи. Задачи выполнить любым из способов. При решении каждой из задач требуется подробно описать каждый из шагов.

3. ЛАБОРАТОРНАЯ РАБОТА № 2 ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ

3.1. Основы и законы алгебры логики

Основой построения любого устройства, использующего цифровую информацию, являются логические и запоминающие элементы. Логические элементы выполняют простейшие логические операции над цифровой информацией, а запоминающие – ее хранят.

Логическая операция состоит в преобразовании по определенным правилам входной информации в выходную. Сигналы на входах и выходах логических элементов обычно являются двоичными (бинарными), то есть принимают лишь два значения, символически обозначаемые как 0 и 1. Поэтому их также

называют двоичными переменными и обозначают буквами латинского алфавита (входные сигналы x_1, x_2, \dots, x_n , а результат операции, то есть выходной сигнал – y). Переменная x может принимать два значения:

- либо $x = 1$ (событие истинно);
- либо $x = 0$ (событие ложно).

Эти переменные называются также булевыми по имени английского математика Дж. Булля, который в середине XIX в. разработал основные положения алгебры логики.

Различные логические переменные могут быть связаны функциональными зависимостями. Например, $y = f(x_1, x_2)$ указывает на функциональную зависимость логической переменной y от логических переменных x_1 и x_2 , называемых аргументами или входными переменными.

В алгебре логики имеются четыре основных закона:

- переместительный (свойство коммутативности);
- сочетательный (свойство ассоциативности);
- распределительный (свойство дистрибутивности);
- инверсии (закон де Моргана).

Переместительный и сочетательный законы имеют место в обычной алгебре.

Распределительного закона и закона инверсии в обычной алгебре нет.

Соотношения, отображающие основные законы алгебры логики, приведены в табл. 3.1.

Используя основные законы алгебры логики, можно составить ряд правил, которые применяются при анализе сложных логических выражений, приведения их к более простому и удобному виду, что показано в табл. 3.2.

Законам и правилам булевой алгебры присуще свойство симметрии. Все правила, кроме последнего, представлены парой соотношений. В каждой паре одно соотношение вытекает из другого заменой логического сложения логическим умножением и наоборот. Кроме того, все значения “0” заменяются на “1” и наоборот. Это свойство симметрии отражает принцип двойственности булевой алгебры.

Таблица 3.1

Соотношения, отображающие основные законы алгебры логики

№	Закон	Логическое сложение	Логическое умножение
1	2	3	4
1	Переместительный	$X_1 \vee X_2 = X_2 \vee X_1$	$X_1 X_2 = X_2 X_1$
2	Сочетательный	$(X_1 \vee X_2) \vee X_3 = X_1 \vee (X_2 \vee X_3)$	$(X_1 X_2) X_3 = X_1 (X_2 X_3)$
3	Распределительный	$(X_1 \vee X_2) X_3 = X_1 X_3 \vee X_2 X_3$	$X_1 X_2 \vee X_3 = (X_1 \vee X_3)(X_2 \vee X_3)$
4	Инверсии	$\overline{X_1 \vee X_2} = \overline{X_1} \overline{X_2}$	$\overline{X_1 X_2} = \overline{X_1} \vee \overline{X_2}$

Правила алгебры логики

№	Правило	а	б
1	2	3	4
1	Инверсии	$\bar{0} = 1$	$\bar{1} = 0$
2	Неизменности	$X \vee 0 = X$	$X * 1 = X$
3	Универсального и нулевого множества	$X \vee 1 = 1$	$X * 0 = 0$
4	Повторения	$X \vee X = X$	$X * X = X$
5	Дополнительности	$X \vee \bar{X} = 1$	$X * \bar{X} = 0$
6	Склеивания	$X_1 X_2 \vee X_1 \bar{X}_2 = X_1$	$(X_1 \vee X_2)(X_1 \vee \bar{X}_2) = X_1$
7	Двойного отрицания	$\overline{\bar{X}} = X$	

Правило склеивания широко используется при минимизации логических функций с целью их упрощения.

3.2. Преобразование булевых выражений

Рассмотренные законы и правила используются для тождественных преобразований булевых выражений, описывающих логические функции. Булево выражение представляет формулу, состоящую из логических констант и логических переменных, соединенных знаками логических операций. Как и в обычной алгебре, для задания порядка выполнения логических выражений используются скобки. Примером булевого выражения для трех переменных служит формула:

$$f(X_1, X_2, X_3) = (\bar{X}_1 * \bar{X}_2 * X_3 \vee X_1 * \bar{X}_2)(X_1 \vee X_3) \quad (3.1)$$

Считается, что операция логического умножения всегда предшествует операции логического сложения.

Можно преобразовать (3.1), применив законы и правила алгебры логики:

$$\begin{aligned} (\bar{X}_1 * \bar{X}_2 * X_3 \vee X_1 * \bar{X}_2)(X_1 \vee X_3) &= \bar{X}_1 * \bar{X}_2 * X_3 * X_1 \vee \bar{X}_1 * \bar{X}_2 * X_3 * X_3 \vee \\ &\vee X_1 * \bar{X}_2 * X_1 \vee X_1 * \bar{X}_2 * X_3 = \bar{X}_1 * \bar{X}_2 * X_3 \vee X_1 * \bar{X}_2 * X_3 = \bar{X}_2 * X_3 (\bar{X}_1 \vee X_1) \vee \\ &\vee X_1 * \bar{X}_2 = \bar{X}_2 * X_3 \vee X_1 * \bar{X}_2 = \bar{X}_2 (X_3 \vee X_1) \end{aligned} \quad (3.2)$$

Полученное в результате тождественных преобразований выражение (3.2) значительно проще выражения (3.1). Процесс упрощения логического выражения, основанный на тождественных преобразованиях, называется минимизацией булевых выражений.

3.3. Дизъюнктивные нормальные формы

Для записи одной и той же функции алгебры логики можно использовать различные формы. Формы, которые представляют суммы элементарных произведений, называют дизъюнктивными нормальными формами (ДНФ).

Под элементарным понимается такое произведение, в котором сомножителями являются только отдельные переменные или их отрицания. Например, формула

$$f(X_1, X_2, X_3) = \overline{X_2} * X_3 \vee X_1 * \overline{X_2} \quad (3.3)$$

содержит два элементарных произведения, каждое из которых состоит из двух сомножителей. Одна и та же функция может быть представлена множеством различных ДНФ. Существуют такие виды ДНФ, в которых функция может быть записана единственным образом. Такие формы называют совершенными дизъюнктивными нормальными формами (СДНФ).

СДНФ определяется как сумма элементарных произведений, в которых каждая переменная встречается ровно один раз либо с отрицанием, либо без него. Для преобразования функции (3.2) в СДНФ необходимо дополнить каждое элементарное произведение недостающими переменными так, чтобы тождественность преобразования не была нарушена:

$$f(X_1, X_2, X_3) = \overline{X_2} * X_3 \vee X_1 * \overline{X_2} = (X_1 \vee \overline{X_1}) \overline{X_2} * X_3 \vee X_1 * \overline{X_2} (X_3 \vee \overline{X_3}) \quad (3.4)$$

После раскрытия скобок и приведения подобных членов, получится функция, записанная в СДНФ:

$$\begin{aligned} f(X_1, X_2, X_3) &= X_1 * \overline{X_2} * X_3 \vee \overline{X_1} * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * \overline{X_3} = \\ &= X_1 * \overline{X_2} * X_3 \vee \overline{X_1} * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * \overline{X_3} \end{aligned} \quad (3.5)$$

Здесь каждая переменная или ее отрицание содержится по одному разу в каждом элементарном произведении. Функция (3.5) обращается в логическую единицу при трех различных комбинациях значений входных переменных:

- $X_1 = 1, X_2 = 0, X_3 = 1$ – первая комбинация;
- $X_1 = 0, X_2 = 0, X_3 = 1$ – вторая комбинация;
- $X_1 = 1, X_2 = 0, X_3 = 0$ – третья комбинация.

Для каждой комбинации соответствующее элементарное произведение равно единице, для всех остальных комбинаций входных переменных – нулю.

Таблица истинности такой функции (табл. 3.3) содержит три строки, в которых функция равна 1.

Таблица истинности функции

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
1	2	3	4
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Каждой из этих строк соответствует по одной из рассмотренных выше комбинаций входных переменных, то есть таблица истинности имеет столько строк, где функция обращается в 1, сколько элементарных произведений содержит ее СДНФ.

Чтобы написать СДНФ по таблице истинности, необходимо для всех комбинаций входных переменных, обращающих функцию в 1, записать элементарные произведения, инвертируя переменные, принимающие на данной комбинации нулевые значения, а все полученные элементарные произведения соединить знаками логического сложения.

Применив это правило, можно получить

$$f(X_1, X_2, X_3) = \overline{X_1} * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * \overline{X_3} \vee X_1 * \overline{X_2} * X_3 \quad (3.6)$$

3.4. Минимизация логических функций

Рассмотренные тождественные преобразования позволяют существенно упростить выражения логических функций. Каждая логическая функция реализуется с помощью определенного набора устройств. Чем меньше элементов содержит выражение, тем проще схема, реализующая соответствующую ему логическую функцию. Поэтому значительный интерес представляет рассмотрение методов минимизации логических функций.

Наиболее распространенным является метод непосредственных тождественных преобразований. Этот метод, рассмотренный выше, состоит в последовательном применении к логической формуле законов и правил тождественных преобразований алгебры логики. Он пригоден для простых формул, когда последовательность преобразований очевидна. Наиболее часто этот метод применяется для окончательной минимизации выражений, полученных после минимизации их другими методами.

Рассмотрим применение метода непосредственных преобразований на примере минимизации функции трех переменных, заданных ее СДНФ:

$$Y = \overline{X_1} * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * \overline{X_3} \vee X_1 * \overline{X_2} * X_3 \vee X_1 * X_2 * \overline{X_3} \vee X_1 * X_2 * X_3 \quad (3.7)$$

После объединения попарно первого и третьего, второго и третьего, а также четвертого и пятого элементарные произведения:

$$Y = (\overline{X_1} * \overline{X_2} * X_3 \vee X_1 * \overline{X_2} * X_3) \vee (X_1 * \overline{X_2} * \overline{X_3} \vee X_1 * \overline{X_2} * X_3) \vee (X_1 * X_2 * \overline{X_3} \vee X_1 * X_2 * X_3) = \overline{X_2} * X_3 \vee X_1$$

Эта формула гораздо проще исходной СДНФ. В формуле (3.7) в каждой паре объединяемые элементарные произведения различаются лишь одной переменной, которая входит в первое произведение с отрицанием, а во второе – без отрицания. Такие элементарные произведения называются соседними. К соседним произведениям применима операция склеивания, в результате которой уменьшается число суммируемых произведений и на единицу уменьшается число переменных.

3.5. Табличные методы минимизации. Карты Карно

Стремление к алгоритмизации поиска соседних элементарных произведений привело к разработке табличных методов минимизации логических функций. Одним из них является метод, основанный на использовании карт Карно.

Карты Карно – это графическое представление таблиц истинности логических функций. Они представляют собой таблицы, содержащие по 2^n прямоугольных ячеек, где n – число логических переменных.

На рис. 3.1, 3.2 и 3.3 приведены структуры карт Карно для функции

	X_2	0	1
X_1	0	$f(0,0)$	$f(0,1)$
	1	$f(1,0)$	$f(1,1)$

Рис. 3.1. Структура карты Карно для функции двух переменных, трех и четырех переменных, а в табл. 3.4 и 3.5 представлены таблицы истинности для функции двух, трех переменных соответственно.

	X_2X_3	00	01	11	10
X_1	0	$f(0,0,0)$	$f(0,0,1)$	$f(0,1,1)$	$f(0,1,0)$
	1	$f(1,0,0)$	$f(1,0,1)$	$f(1,1,1)$	$f(1,1,0)$

Рис. 3.2. Структура карты Карно для функции трех переменных

	X_3X_4	00	01	11	10
X_1X_2	00	$f(0,0,0,0)$	$f(0,0,0,1)$	$f(0,0,1,1)$	$f(0,0,1,0)$
	01	$f(0,1,0,0)$	$f(0,1,0,1)$	$f(0,1,1,1)$	$f(0,1,1,0)$
	11	$f(1,1,0,0)$	$f(1,1,0,1)$	$f(1,1,1,1)$	$f(1,1,1,0)$
	10	$f(1,0,0,0)$	$f(1,0,0,1)$	$f(1,0,1,1)$	$f(1,0,1,0)$

Рис. 3.3. Структура карты Карно для функции четырех переменных

Таблица 3.4

Таблица истинности для функции двух переменных

X_1	X_2	$f(X_1, X_2)$
1	2	3
0	0	$f(0,0)$
0	1	$f(0,1)$
1	0	$f(1,0)$
1	1	$f(1,1)$

Карта Карно размечается системой координат, соответствующих значениям входных переменных. Например, верхняя строка карты для функции трех переменных соответствует нулевому значению переменной x_1 , а нижняя – ее единичному значению. Каждый столбец этой карты характеризуется значениями двух переменных: x_2 и x_3 . Комбинация цифр, которыми отмечается каждый столбец, показывает, для каких значений переменных x_3 и x_2 вычисляется функция, размещаемая в клетках этого столбца. Так, в случае карты Карно для функции четырех переменных, функция, расположенная в ячейках столбца с координатами 01, вычисляется при значениях переменных $X_3 = 0$, $X_4 = 1$. Функция, расположенная в ячейке на пересечении этого столбца и строки с координатами 11, определяется при наборе входных переменных $X_1 = 1$, $X_2 = 1$, $X_3 = 0$, $X_4 = 1$.

Таблица 3.5

Таблица истинности для функции трех переменных

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
1	2	3	4
0	0	0	$f(0, 0, 0)$
0	0	1	$f(0, 0, 1)$
0	1	0	$f(0, 1, 0)$
0	1	1	$f(0, 1, 1)$
1	0	0	$f(1, 0, 0)$
1	0	1	$f(1, 0, 1)$
1	1	0	$f(1, 1, 0)$
1	1	1	$f(1, 1, 1)$

Если на указанном наборе функция равна 1, то ее СДНФ обязательно содержит элементарное произведение $X_1 * X_2 * \overline{X_3} * X_4$, принимающее при этом наборе единичное значение. Таким образом, ячейки карты Карно, представляющие функцию, содержат столько единиц, сколько элементарных произведений содержится в ее СДНФ, причем каждой единице соответствует одно из элементарных произведений.

Координаты строк и столбцов в карте Карно следуют не в естественном порядке возрастания двоичных кодов, а в порядке 00; 01; 11; 10. Изменение порядка следования наборов кодов сделано для того, чтобы соседние наборы, от-

личающиеся между собой лишь цифрой какого-либо одного разряда, были соседними в геометрическом смысле.

Рассмотрим таблицу истинности (табл. 3.6) и структуру карты Карно, показанную на рис. 3.4, для функции:

$$f(X_1, X_2, X_3) = \overline{X_1} \vee \overline{X_2} * X_3$$

	X_2X_3	00	01	11	10
X_1	0	1	1	1	1
	1	0	1	0	0

Рис. 3.4. Структура карты Карно для функции

Ячейки, в которых функция принимает значение 1, заполняются единицами. Процесс минимизации заключается в формировании прямоугольников, содержащих по 2^k ячеек, где k – целое число. В прямоугольники объединяются соседние ячейки, которые соответствуют соседним элементарным произведениям. Например, на рис. 3.4 объединены ячейки с координатами 001 и 101. При объединении этих ячеек образовался прямоугольник, в котором X_1 изменяет свое значение. Следовательно, оно исчезает при склеивании соответствующих элементарных произведений $\overline{X_1} * \overline{X_2} * X_3$ и $X_1 * \overline{X_2} * X_3$. Ячейки, расположенные в первой строке, содержат 1 и являются соседними. Поэтому они объединяются в прямоугольник, содержащий $2^2 = 4$ ячейки. Переменные X_2 и X_3 в пределах прямоугольника меняют свое значение, следовательно, они исчезнут из результирующего элементарного произведения. Переменная X_1 является неизменной и равной нулю. Таким образом, элементарное произведение, полученное в результате объединения ячеек первой строки, содержит лишь элемент X_1 . Это следует из того, что четырем ячейкам первой строки соответствует сумма четырех элементарных произведений:

$$\begin{aligned} & \overline{X_1} * \overline{X_2} * \overline{X_3} \vee X_1 * \overline{X_2} * X_3 \vee \overline{X_1} * X_2 * X_3 \vee X_1 * X_2 * \overline{X_3} = \\ & = \overline{X_1} * \overline{X_2} (\overline{X_3} \vee X_3) \vee \overline{X_1} * X_2 (X_3 \vee \overline{X_3}) = \overline{X_1} * \overline{X_2} \vee \overline{X_1} * X_2 = \overline{X_1} (\overline{X_2} \vee X_2) = \overline{X_1} \end{aligned}$$

Таблица 3.6
Таблица истинности для функции $f(X_1, X_2, X_3) = \overline{X_1} \vee \overline{X_2} * X_3$

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1

Совокупность прямоугольников, покрывающих все единицы, называют покрытием. Одна и та же ячейка может покрываться несколько раз. Итак, можно сделать следующие выводы:

- формула, получающаяся в результате минимизации логической функции с помощью карт Карно, содержит сумму столько же элементарных произведений, сколько прямоугольников имеется в покрытии;

- чем больше ячеек в прямоугольнике, тем меньше переменных содержится в соответствующем ему элементарном произведении.

Несмотря на то, что карты Карно изображаются на плоскости, соседство ячеек устанавливается на поверхности тора или цилиндра. Верхняя и нижняя границы карты Карно как бы склеиваются, образуя цилиндр. При склеивании боковых границ получается тороидальная поверхность. Поэтому в примере, показанном на рис. 3.5, ячейки с координатами 1011 и 0011 являются соседними и объединяются в прямоугольники. Действительно, указанным ячейкам соответствует сумма элементарных произведений:

$$\overline{X_1} * \overline{X_2} * X_3 * X_4 \vee X_1 * \overline{X_2} * X_3 * X_4 = \overline{X_2} * X_3 * (X_1 * \overline{X_1}) = \overline{X_2} * X_3 * X_4$$

	X_3X_4	00	01	11	10
X_1X_2	00	0	0	1	0
01	00	1	0	0	1
11	00	1	0	0	1
10	00	0	0	1	0

Рис. 3.5. Карта Карно для функции четырех переменных

Аналогично объединяются и остальные четыре единичные ячейки. В результате их объединения можно получить:

$$\begin{aligned} & \overline{X_1} * X_2 * \overline{X_3} * \overline{X_4} \vee X_1 * X_2 * \overline{X_3} * \overline{X_4} \vee \overline{X_1} * X_2 * X_3 * \overline{X_4} \vee X_1 * X_2 * X_3 * X_4 = \\ & = X_2 * \overline{X_3} * \overline{X_4} * (X_1 \vee \overline{X_1}) \vee X_2 * X_3 * \overline{X_4} * (X_1 \vee \overline{X_1}) = X_2 * \overline{X_4} * (X_3 \vee \overline{X_3}) = X_2 * X_4 \end{aligned}$$

Поэтому окончательно:

$$f(X_1, X_2, X_3, X_4) = \overline{X_2} * X_3 * X_4 \vee X_2 * \overline{X_4}$$

Рассмотренные примеры позволяют сформулировать последовательность действий, выполняемых при минимизации логических функций с использованием карт Карно:

- изображается таблица для n переменных и производится разметка ее сторон;

- ячейки таблицы, соответствующие наборам переменных, обращающих функцию в 1, заполняются единицами, остальные ячейки – нулями;

- выбирается наилучшее покрытие таблицы правильными прямоугольниками.

Наилучшим считается такое покрытие, которое образовано минимальным числом прямоугольников, а если таких вариантов несколько, то из них выбирается тот, который дает максимальную суммарную площадь прямоугольников.

Качество минимизации оценивается коэффициентом покрытия:

$$k = \frac{m}{s} \quad (3.8)$$

где m – общее количество прямоугольников;

s – суммарное количество покрытых ячеек.

Покрытие считается тем лучше, чем меньше его коэффициент k .

3.6. Неполностью определенные логические функции

При рассмотрении двоично-десятичных кодов было отмечено, что из 16 возможных комбинаций используются только 10, а остальные комбинации запрещены и возникать не должны. Если каждому разряду поставить в соответствие двоичную переменную, то для двоично-десятичных кодов можно получить шесть запрещенных комбинаций переменных. Они приведены в табл. 3.7. Если функция имеет запрещенные наборы переменных, то ее значения на указанных наборах не определены и в таблице истинности отмечаются знаком *. Например, в таблице для трех переменных представлена функция (табл. 3.8), имеющая три запрещенных набора переменных.

Таблица 3.7

Таблица истинности для функции

Цифра	X_1	X_2	X_3	X_4	Набор	
1	2	3	4	5	6	
0	0	0	0	0	Разрешенный	
1	0	0	0	1		
2	0	0	1	0		
3	0	0	1	1		
4	0	1	0	0		
5	0	1	0	1		
6	0	1	1	0		
7	0	1	1	1		
8	1	0	0	0		
9	1	0	0	1		
10	1	0	1	0	Запрещенный	
1	2	3	4	5		6
11	1	0	1	1		
12	1	1	0	0		
13	1	1	0	1		
14	1	1	1	0		
15	1	1	1	1		

Таблица истинности для функции

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
1	2	3	4
0	0	0	*
0	0	1	1
0	1	0	*
0	1	1	1
1	0	0	0
1	0	1	*
1	1	0	0
1	1	1	1

Функция, я такой и подобной таблицами истинности применяется, например, в дешифраторе для семисегментного индикатора, показывающего цифры от 0 до 9. В этом случае при поступлении на входы двоичных кодовых комбинаций, соответствующих десятичным числам от 10 до 15, на выходах дешифратора должны быть сигналы лог. 0 или лог. 1, при которых ни один сегмент индикатора светиться не должен, соответственно не должна быть показана ни одна цифра.

Двоичные функции, значения которых определены не для всех наборов входных переменных, называются неполностью определенными. На карте Карно ячейки, соответствующие запрещенным наборам переменных, также отмечаются знаком *, что показано на рис. 3.6. При минимизации неполностью определенной функции ее следует доопределить, то есть неопределенные значения ячеек карты Карно произвольным образом заменить единицами или нулями.

На рис. 3.7 показана функция $f(X_1, X_2, X_3)$, все значения * которой заменены единицами. Доопределенная функция имеет вид $f(X_1, X_2, X_3) = \overline{X_1} \vee X_3$ не зависит от X_2 .

Эти примеры показывают возможности упрощения формулы неполностью определенной функции при ее соответствующем доопределении.

Если функция имеет m запрещенных наборов переменных, то может быть выбран тот вариант, при котором формула минимизированной функции будет наиболее простой.

3.7. Логические элементы и логические операции

$X_1 \backslash X_2 X_3$	00	01	11	10
0	*	1	1	*
1	0	*	1	0

Рис. 3.6. Карта Карно для функции трех переменных

Основной операцией цифровой обработки информации является реализация функциональных зависимостей $Y = f(X_1, X_2, \dots, X_n)$, ставящих в соответствие каждой комбинации значений двоичных переменных X_1, X_2, \dots, X_n значение двоичной переменной Y . Функция такого типа называется переключательной или логической. Переключательную функцию можно задать таблицей, в левой части которой перечисляются комбинации значений аргументов, а в правой - значения функции.

$X_1 \backslash X_2 X_3$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Рис. 3.7. Замена знаков * нулями в верхней строке

Переключательную функцию можно задать в аналитической форме, то есть в виде некоторого выражения, описывающего последовательность элементарных операций над аргументами функций. Совокупность переключательных функций, определяющая набор элементарных операций, достаточных для реализации любой переключательной функции, называется базисом. В большинстве случаев необходимые логические преобразования двоичных сигналов выполняются на базе трех элементарных операций:

- логического сложения;
- логического умножения;
- инверсии.

Логическое сложение (дизъюнкция), либо операция ИЛИ обозначается \vee или знаком "+":

$$Y = X_1 \vee X_2 \vee \dots \vee X_n = X_1 + X_2 + \dots + X_n \quad (3.9)$$

Читается так: Y есть X_1 или X_2 или, ..., или X_n . Иначе говоря, Y есть единица, если хотя бы одно из слагаемых равно единице.

Логическое умножение (конъюнкция), либо операция И обозначается символом \wedge или знаком "*":

$$Y = X_1 \wedge X_2 \wedge \dots \wedge X_n = X_1 * X_2 * \dots * X_n \quad (3.10)$$

Читается так: Y есть X_1 и X_2 и, ..., и X_n . Иначе говоря, Y есть единица только тогда, когда все сомножители равны единице.

Логическое отрицание, называемое также инверсией либо операцией НЕ, обозначается чертой над переменной $Y = \overline{X}$, читается так: Y есть не X .

Перечисленные операции образуют так называемую булеву алгебру. Известно, что любую переключательную функцию можно представить аналитическим выражением в булевой алгебре, то есть совокупность логических функций,

состоящая из логических сложения, умножения и отрицания, является базисом.

Правила выполнения логических операций над двоичными переменными для случая двух входных сигналов представлены в табл. 3.9, называемой таблицей истинности. Аналогично определяются операции ИЛИ, И для n входных переменных.

Таблица 3.9

Таблица истинности

Операция ИЛИ			Операция И			Операция НЕ	
X_1	X_2	Y	X_1	X_2	Y	X	Y
0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	0
1	0	1	1	0	0		
1	1	1	1	1	1		

Логическим элементом (ЛЭ) или логической схемой называется устройство, реализующее заданную переключательную функцию. Обычно такое устройство имеет $n \geq 1$ входов и один выход.

На функциональных и структурных схемах ЛЭ условно изображается прямоугольником, внутри которого записана реализуемая им логическая функция, что показано на рис. 3.8.

Функция логического сложения условно обозначается символом "1", умножения – символом "&", инверсия на выходе (входе) ЛЭ – кружком на выходе (входе) прямоугольника. Логический элемент, реализующий операцию ИЛИ, называется либо дизъюнктом, либо элементом ИЛИ, либо сборкой. Логический элемент, реализующий операцию И, называется либо конъюнктом, либо элементом И, либо схемой совпадения. Логический элемент, выполняющий операцию НЕ, называется инвертором.

Также широко применяются комбинированные элементы, реализующие последовательно не одну операцию отрицания дизъюнкции – элемент ИЛИ-НЕ, и операцию отрицание конъюнкции - элемент И-НЕ. Логические функции, реализуемые этими элементами, записываются соответственно:

$$Y = \overline{X_1 + X_2 + \dots + X_n} \quad (3.11)$$

$$Y = \overline{X_1 * X_2 * \dots * X_n} \quad (3.12)$$

Элементы ИЛИ-НЕ, И-НЕ являются универсальными, так как с помощью элементов одного из этих типов, применяя соответствующее соединение элементов друг с другом, можно выполнить любую базисную функцию И, ИЛИ, НЕ.

Важно заметить, что у элементов ИЛИ, И, ИЛИ-НЕ, И-НЕ может быть от двух (минимум) входов и более. Ограничение на количество входов налагает только аппаратная реализация логического элемента [8].

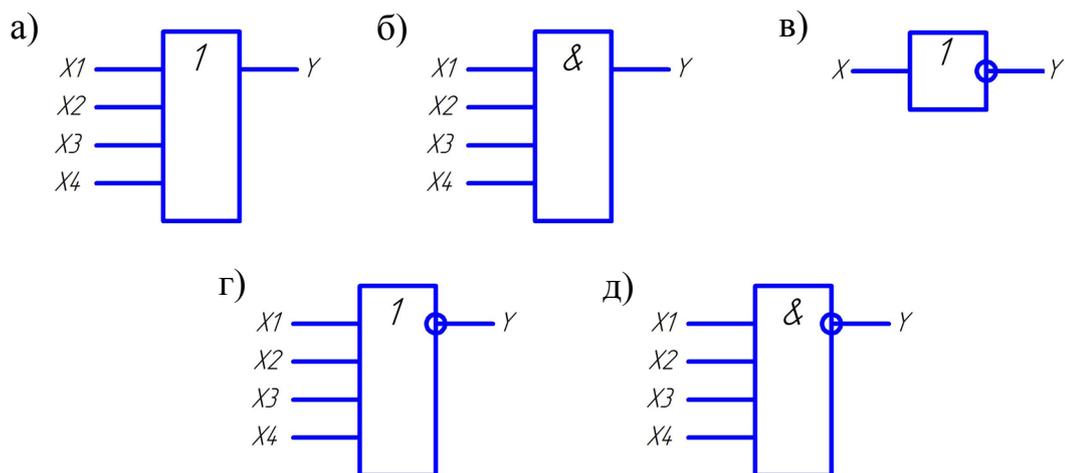


Рис. 3.8. Функциональное обозначение логических элементов.
 а – элемент ИЛИ; б – элемент И; в – элемент НЕ; г – элемент ИЛИ-НЕ;
 д – элемент И-НЕ

3.8. Цель лабораторной работы

Целью лабораторной работы является:

- практическое освоение преобразование булевых (логических) выражений и их минимизация, а также составление карт Карно;
- практическое освоение составления таблицы истинности логической функции;
- составление логической схемы по логической функции и таблице истинности.

3.9. Задание на лабораторную работу

Получить у преподавателя вариант задания с указанием логической функции. Преобразовать и минимизировать заданную логическую функцию. По результатам минимизации составить карту Карно. Составить таблицу истинности получившейся минимизированной функции. Составить из логических элементов логическую схему по получившейся минимизированной функции и ее таблице истинности.

3.10. Выполнение лабораторной работы

В ходе выполнения работы требуется, используя приведенные теоретические сведения решить поставленные задачи. Задачи выполнить любым из способов. При решении каждой из задач требуется подробно описать каждый из шагов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Юревич Е.И. Основы робототехники. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2018. – 415 с.: ил.
2. Н.В. Василенко, К.Д. Никитин, В.П. Пономарев, А.Ю. Смолин. Основы робототехники. Под общей редакцией К.Д. Никитина. Томск. МГП “Раско”. – 1993. - 475 с. ил.
3. Шахворостов, С.А. Роботы в системах автоматизации [Электронный ресурс] : учеб. пособие/ С.А. Шахворостов. – Электрон. текстовые дан. (1 файл: 6 МБ). – Красноярск: Научно-инновационный центр, 2016. – 110 с.
4. Система контурного управления УКМ-772. Техническое описание и руководство по эксплуатации.
5. Е.И. Юревич. Управление роботами и робототехническими системами. Санкт-Петербург. БХВ. - 2000. – 171 с. ил.
6. Литвиненко А.М. Системы управления роботами: Учеб. пособие/ Воронеж. гос. техн. ун-т. Воронеж, 2002. 140 с.
7. Тюрин И.В. Вычислительная техника : учебное пособие / И. В. Тюрин. - Тамбов : Издательский центр ФГБОУ ВО «ТГТУ», 2019. – 112 с.
8. Шило В.Л. Популярныe цифровые микросхемы: справочник. - М.: радио и связь, 1987. – 352.(Массовая радиобиблиотека. Вып. 1111).

ОГЛАВЛЕНИЕ

1. Общие сведения о системах управления.....	3
1.1. Классификация систем управления.....	3
1.2. Структура и принцип действия цикловой системы программного управления.....	3
1.3. Структура и принцип действия позиционной системы программного управления.....	7
1.4. Структура и принцип действия контурной системы программного управления.....	10
1.5. Адаптивное управление роботами.....	15
1.6. Интеллектуальное управление мехатронными системами.....	18
1.7. Управление средствами передвижения роботов.....	22
2. Лабораторная работа № 1. Арифметические основы цифровых устройств.....	24
2.1. Теоретические сведения о системах счисления.....	24
2.2. Теоретические сведения о двоичной, восьмеричной и шестнадцатиричной системах счисления.....	27
2.3. Цель лабораторной работы.....	31
2.4. Задание на лабораторную работу.....	31
2.5. Выполнение лабораторной работы.....	31
3. Лабораторная работа № 2. Логические основы цифровых устройств.....	31
3.1. Основы и законы алгебры логики.....	31
3.2. Преобразование булевых выражений.....	33
3.3. Дизъюнктивные нормальные формы.....	34
3.4. Минимизация логических функций.....	35
3.5. Табличные методы минимизации. Карты Карно.....	36
3.6. Неполностью определенные логические функции.....	40
3.7. Логические элементы и логические операции.....	41
3.8. Цель лабораторной работы.....	44
3.9. Задание на лабораторную работу.....	44
3.10. Выполнение лабораторной работы.....	44
Библиографический список.....	45

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению лабораторных работ № 1-2

для студентов специальностей

15.02.10 «Мехатроника и мобильная робототехника (по отраслям)»

и 15.02.10 «Мехатроника и робототехника (по отраслям)»

всех форм обучения

Составитель

Коротков Виктор Николаевич

Издается в авторской редакции

Компьютерный набор В. Н. Короткова

Подписано к изданию 18.04.2024.

Уч.-изд. л. 2,5.

ФГБОУ ВО «Воронежский государственный технический университет»
394006 г. Воронеж, ул. 20-летия Октября, 84