

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

С. И. Зиатдинов, Л. А. Осипов

ПРОЕКТИРОВАНИЕ
СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ
ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Учебное пособие

Санкт-Петербург
2002

УДК 681.32.01(075)

ББК 32.973.2

359

Зиатдинов С. И., Осипов Л. А.

359 Проектирование специализированных вычислителей цифровой обработки сигналов: Учеб. пособие / СПбГУАП. СПб., 2002. 75 с.: ил.

Изложены вопросы проектирования специализированных вычислителей на базе микроЭВМ, микроконтроллеров и БИС с жесткой логикой. Даны теоретические основы решения задач цифровой обработки информации.

Учебное пособие предназначено для студентов старших курсов, специализирующихся в области создания вычислительных систем.

Рецензенты:

кафедра радиотехнических систем Санкт-Петербургского
государственного университета телекоммуникаций им. проф. Бонч-Бруевича;
доктор технических наук *В. В. Изранцев*

Утверждено

редакционно-издательским советом университета
в качестве учебного пособия

© СПбГУАП, 2002

© С. И. Зиатдинов, Л. А. Осипов, 2002

ВВЕДЕНИЕ

Специализированные вычислители решают задачи первичной и вторичной цифровой обработки информации.

При первичной обработке обеспечивается преобразование входных сигналов в цифровую форму, обнаружение полезных сигналов в помехах, измерение параметров сигналов, их спектрально-корреляционные преобразования и т. д. Основные методы решения этих задач заключаются в цифровой обработке во временной области и обработке в частотной области (например, цифровая фильтрация с использованием дискретного преобразования Фурье).

Исключительно важным требованием является необходимостью выполнения обработки в реальном масштабе времени, что накладывает жесткие ограничения как на время решения задач, так и на скорость обмена информацией средствами интерфейса.

При вторичной обработке решаются задачи траекторных измерений, распознавание образов, задачи управления, контроля, диагностики и т. д.

В случае цифровой обработки различные алгоритмы могут реализовываться аппаратно, программно и программно-аппаратно.

Основной принцип замены аппаратных средств программными заключается в том, что программы, реализуемые микропроцессорной системой, могут заменить аппаратные средства, которые воспринимают, хранят, обрабатывают и выдают цифровую информацию. Аппаратные и программные средства оказываются, в известной степени, взаимозаменяемыми и соотношения между ними определяются, главным образом, экономическими факторами.

Аппаратная реализация труднее поддается модификации, а программная – отличается гибкостью. Самое существенное ограничение замены аппаратных средств программными – это быстрдействие. Производительность специализированного устройства, выполняющего некоторую функцию, всегда выше производительности устройств, реализующих ту же функцию программно.

В настоящее время реализация специализированных вычислителей, обеспечивающих выполнение заданного алгоритма за отрезок времени, исчисляемый долями и единицами микросекунды, возможна лишь на устройствах с жесткой логикой.

В этом случае эффективный подход к обеспечению обработки в реальном масштабе времени – это создание специализированных средств цифровой обработки на основе БИС и СБИС. В настоящее время появились специализированные микросхемы сложения, вычитания, умножения и деления многоразрядных чисел, представленных в различных форматах.

В то же время в таких областях, как связь, медицина, акустика, бытовая техника и т. д. с успехом можно использовать микропроцессорные системы, обладающие исключительной гибкостью алгоритмов обработки.

В настоящем учебном пособии изложены вопросы разработки специализированных вычислителей как на микропроцессорной основе, так и на базе устройств с жесткой логикой.

1. ОБЩИЕ СВЕДЕНИЯ О СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЯХ

1.1. Обобщенная структура специализированного вычислителя

Укрупненно в состав специализированного вычислителя, независимо от используемой элементной базы, входят устройства ввода информации ($Y_{\text{вв}}$), арифметико-логическое устройство (АЛУ), устройство вывода информации ($Y_{\text{выв}}$). Данная структура показана на рис. 1.1.

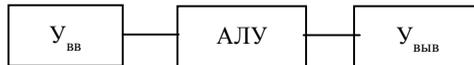


Рис. 1.1

Входная информация может быть представлена как в цифровой, так и в виде непрерывного либо дискретного аналогового сигнала. При цифровом представлении входной информации для ее временного хранения могут использоваться регистры-защелки и программируемые контроллеры ввода данных.

Непосредственный ввод информации может осуществляться как по инициативе спецвычислителя, так и в режиме запроса прерывания от внешнего устройства.

В случае использования аналогового входного сигнала в устройстве ввода происходит его преобразование в цифровую форму с помощью аналого-цифрового преобразователя (АЦП).

Арифметико-логическое устройство, построенное в виде программируемого автомата либо по схеме с жесткой логикой, реализует заданный пользователем алгоритм обработки. Результат обработки выводится через устройства вывода к потребителю. При этом вывод данных может осуществляться как в цифровой форме, так и с предварительным преобразованием с помощью цифроаналогового преобразователя в непрерывный сигнал. Вывод информации реализуется либо по инициативе спецвычислителя, либо в режиме прерывания по запросу внешнего устройства.

Очевидно, что качество выполнения заданного алгоритма всецело определяется быстродействием и точностными характеристиками всех трех устройств, входящих в специализированный вычислитель.

При этом количество операций ввода-вывода мало по сравнению с числом арифметических и логических операций. По этой причине основная нагрузка ложится на арифметико-логическое устройство.

1.2. Обзор современных микропроцессоров

Для реализации алгоритмов обработки широко используются как микропроцессоры и микроконтроллеры, так и микроЭВМ. Такие устройства должны иметь следующие основные функциональные блоки:

микропроцессор (микроконтроллер) или микроЭВМ для выполнения арифметических и логических операций, а также управления;

постоянное запоминающее устройство для хранения программ и констант;

оперативное запоминающее устройство для хранения входных и выходных данных и промежуточных результатов;

интерфейс ввода и вывода для сопряжения вычислителя с внешними устройствами.

Пригодность того или иного микропроцессорного комплекта к решению задач цифровой обработки сигналов определяется рядом тактико-экономических параметров:

- время выполнения команд;
- наличие в системе команд, необходимых для выполнения операций обработки (умножение, деление, сложение, сдвиговые, операции ввода и вывода и т. д.);

- количество регистров общего назначения (сверхоперативной памяти);
- емкость адресуемой памяти, которая определяет максимальный объем обрабатываемой информации;

- наличие системы прерываний (определяет возможность многоканального режима работы).

Программная реализация алгоритмов цифровой обработки сигналов обычно занимает много времени, в первую очередь, из-за длительности выполнения операции умножения. Поэтому она используется в основном для решения задач, не лимитированных жесткими временными рамками. Для решения задач обработки в реальном масштабе времени особенности структуры алгоритмов обработки должны быть учтены возможно более полно в архитектуре вычислительных средств.

Ниже приведены основные технические характеристики современных микропроцессоров и микроконтроллеров.

1.2.1. Микроконтроллеры фирмы Atmel

Фирма Atmel производит две серии микроконтроллеров: серию AT89, совместимую с фактически промышленным стандартом MCS-51, и серию RISC-микроконтроллеров AT90 собственной архитектуры.

Микроконтроллеры серии AT89 имеют следующие основные особенности:

- 8-разрядный процессор, оптимизированный для приложений управления;
- обширные возможности побитовой обработки;
- встроенная FLASH-память программ;
- двунаправленные и индивидуально адресуемые линии ввода-вывода;
- один или несколько 16-разрядных таймеров-счетчиков; полнодуплексный UART;
- разветвленная структура прерываний;
- встроенный тактовый генератор;
- экономичные режимы: IDLE и POWER DOWN;
- встроенная память EEPROM (AT89S);
- последовательный интерфейс SPI (AT89S);
- сторожевой таймер.

Свойства семейства микроконтроллеров представлены в табл. 1.1.

Таблица 1.1

Тип микроконтроллера	AT89 C51	AT89 LV51	AT89 C52	AT89 LV52	AT89 C2051	AT89 C1051	AT89 S8252
Память программ, Кбайт	4	4	8	8	2	1	8
Память данных, байт	128	128	256	256	128	64	256
Память EEPROM, Кбайт	–	–	–	–	–	–	2
Выводы ввода-вывода	32	32	32	32	15	15	32
16-битные таймеры-счетчики	2	2	3	3	2	1	3
UART	+	+	+	+	+	–	+
Источники прерываний	6	6	8	8	6	3	9
Биты защиты	3	3	3	3	2	2	3

Микроконтроллеры AVR серии AT90 с RISC-архитектурой производятся по энергонезависимой КМОП-технологии с низким потреблением

ем по питанию фирмой Atmel. Перепрограммируемая энергонезависимая FLASH-память обеспечивает перепрограммирование контроллеров как с помощью обычного программатора, так и на плате через последовательный интерфейс SPI.

В табл. 1.2 представлен состав микроконтроллеров.

Таблица 1.2

Тип микроконтроллера	AT90S 1200	AT90S 2313	AT90S 4414	AT90S 8515
Количество команд	89	120	120	120
FLASH-память программ, Кбайт	1	2	4	8
Память EEPROM, байт	64	128	256	512
Оперативная память, байт	–	128	256	512
Тактовая частота, МГц	0...16	0...16	0...20	0...20
Количество линий ввода-вывода	15	15	32	32
Аналоговый компаратор	+	+	+	+
8-разрядный таймер-счетчик	+	+	+	+
16-разрядный таймер-счетчик	–	+	+	+
ШИМ	–	–	+	+
UART	–	+	+	+
Последовательный интерфейс SPI	–	–	+	+
Сторожевой таймер	+	+	+	+
Количество выводов	20	20	40	40
Типы корпусов	PDIP, SOIC, SSOP	PDIP, SOIC, SSOP	PDIP, PLCC	PDIP, PLCC

Основные характеристики младшей модели семейства микроконтроллера AT90S1200:

- 89 команд, выполняемых за один такт каждая;
- 32 8-битовых универсальных регистра;
- рабочая частота 0 Гц–16 МГц (минимальный цикл выполнения команды 62,5 нс);
- 16-битовые команды;
- 8-битовые данные;
- 1 Кбайт репрограммируемой FLASH-памяти программ с ресурсом 1000 циклов записи/стирания;
- возможность перепрограммирования на плате через последовательный интерфейс SPI;

- 64 бита электрически перепрограммируемой памяти EEPROM с ресурсом 100000 циклов записи/стирания;
- 15 индивидуально программируемых линий ввода/вывода;
- максимальный втекающий ток – 20 мА;
- максимальный вытекающий ток – 3 мА;
- встроенный аналоговый компаратор;
- 8-разрядный таймер-счетчик с 10-разрядным программируемым предварительным делителем;
- автоматический сброс при подаче напряжения питания;
- сторожевой (Watchdog) таймер с собственным встроенным генератором частотой 1 МГц, обеспечивающим повышенную надежность;
- наличие битов секретности для защиты кодов;
- биты идентификации;
- экономичный режим: IDLE и POWER DOWN;
- энергонезависимая экономичная высокоскоростная КМОП-технология;
- полностью статическая архитектура;
- широкий диапазон рабочего напряжения питания и температуры: 2,7–6,0 В; –5,5–+125 °С;
- низкое энергопотребление:
 - 2 мА (типично для 3 В, 4 МГц);
 - 500 мкА (типично для 3 В, 4 МГц в режиме Idle);
 - 15 мкА (максимально для 3 В в режиме Power down при включенном сторожевом таймере);
 - 1 мкА (максимально для 3 В в режиме Power down при отключенном сторожевом таймере).

1.2.2. PIC-контроллеры фирмы Microchip

Микроконтроллеры семейства PIC (Peripheral Interface Controller) объединяют все передовые технологии микроконтроллеров: электрически программируемые пользователем ППЗУ, минимальное энергопотребление, высокую производительность, хорошо развитую RISC-архитектуру, функциональную законченность и минимальные размеры.

На базе микроконтроллеров фирмы Microchip строятся микроАТС, автоответчики, АОНы, мобильные телефоны, зарядные устройства, факсы, модемы, пейджеры, таймеры, системы сигнализации, измерительные приборы, счетчики воды, газа и электроэнергии, дозиметры, приборы автосигнализации, системы управления зажиганием и впрыском

топлива, приборные панели и радарные детекторы, интеллектуальные датчики, системы управления электродвигателями, промышленные роботы, регуляторы температуры, влажности, давления и т.п., схемы управления принтерами, плоттерами, сетевые контроллеры, сканеры, системы управления аудиосистемами, системы синтеза речевых сообщений, видеогры, системы дистанционного управления, кассовые аппараты и т. д.

В зависимости от производительности и функциональных возможностей микроконтроллеры серии PIC 16/17 подразделяются на три семейства:

- PIC16C5X – базовое семейство с 12-разрядными командами;
- PIC16C6X/7X/8X – расширенное семейство с 14-разрядными командами;
- PIC17CXX – высокопроизводительное семейство с 16-разрядными командами.

Сравнительные характеристики микроконтроллеров семейства PIC приведены в табл. 1.3.

Таблица 1.3

Тип микроконтроллера	PIC12 C673	PIC16 C57	PIC16 C710	PIC17 C42A	PIC17 C752	PIC16 C165
Количество команд	33/35	33	35	58	58	58
Память программ	1024 ×14	2048 ×12	512 ×4	2048 ×16	8192 ×16	8192 ×16
OTP/ROM	ORT	ORT	ORT	ORT	ORT	ORT
Оперативная память, байт	128×8	72×8	36×8	232×8	678×8	512×8
Тактовая частота, МГц	10	20	33	33	33	40
Число линий ввода-вывода	6	20	13	33	50	30
Таймер + сторожевой таймер WDT	1+ WDT	1+ WDT	1+ WDT	4+ WDT	4+ WDT	3+ WDT
Наличие АЦП (количество входов)	4	–	4	–	12	–
Программирование на плате	+	–	+	–	–	+
Умножитель	–	–	–	8?8	8?8	–
Последовательный интерфейс	–	–	–	+	–	–

В таблице использованы следующие обозначения: ORT – однократно программируемый кристалл; ROM – память программ с масочным ПЗУ; WDT – сторожевой таймер.

В качестве примера более подробно рассмотрим технические характеристики микроконтроллеров PIC16C5X:

- 33 команды;
- все команды выполняются за один цикл (200 нс на 20 МГц), кроме команд перехода (2 цикла);
- рабочая частота 0 Гц–20 МГц;
- 12-битовые команды;
- 8-битовые данные;
- 7(8) специальных аппаратных регистров SFR;
- двухуровневый аппаратный стек;
- прямая, непосредственная, косвенная и относительная адресация данных и команд;
- 12(20) линий ввода-вывода с индивидуальной настройкой;
- максимальный втекающий ток – 10 мА;
- максимальный вытекающий ток – 10 мА;
- автоматический сброс при включении;
- 8-битовый таймер/счетчик с 8-битовым программируемым предварительным делителем;
- таймер включения при сбросе;
- сторожевой таймер WDT с собственным встроенным генератором, обеспечивающим повышенную надежность;
- EPROM-бит защиты кода;
- бит идентификации;
- экономичный режим SLEEP;
- выбираемые пользователем биты для установки режима возбуждения встроенного генератора;
- RC-генератор (RC);
- обычный кварцевый резонатор (XT);
- высокочастотный кварцевый резонатор (HS);
- экономичный высокочастотный кристалл;
- встроенное устройство программирования EPROM-памяти программ;
- широкий диапазон напряжений питания и температур:
 - коммерческий: 2,0–6,0 В; 0–+70 °С;
 - промышленный: 2,0–6,0 В; –40–+70 °С;
- низкое энергопотребление:
 - 2 мА (типично для 5 В, 4 МГц);
 - 15 мкА (типично для 3 В, 32 кГц);
 - 1 мкА (в режиме ожидания при 3 В).

1.2.3. Микроконтроллеры SX18AC/SX28AC фирмы Scenix

Микроконтроллеры SX – высокоскоростные 8-разрядные микроконтроллеры с внутренней FLASH-памятью программ объемом 2К слов.

Основные особенности микроконтроллеров:

- скорость выполнения команд 50 MIPS при тактовой частоте 50 МГц;
- выполнение команд за один такт;
- 2048×12 бит FLASH-память с ресурсом 10000 циклов записи;
- совместимость по программному коду и выводам с микроконтроллерами PIC16C5X;
- программирование на плате через вывод OSC;
- пошаговый режим и останов в контрольных точках с использованием вывода OSC2;
- внутренний RC-генератор с частотой 4 МГц с возможностью деления частоты с коэффициентом от 1 до 128;
- аналоговый компаратор;
- все выводы обеспечивают втекающий/вытекающий ток до 30 мА;
- экономичный режим;
- таймер-счетчик;
- каждый вывод можно запрограммировать как вход или выход.

Микроконтроллеры SX имеют специальные возможности, которые обеспечивают снижение стоимости системы и пониженное потребление по питанию. Таймер сброса по подаче питания и сброса устройства устраняет необходимость применения внешней схемы сброса. Имеется возможность выбора одной из пяти конфигураций тактового генератора, включая программируемый внутренний генератор 4 МГц. Энергосберегающий режим SLEEP, сторожевой таймер и возможность защиты кода уменьшают стоимость системы.

Области применения микроконтроллеров SX:

- генерация и измерение частоты;
- работа с сетями;
- контроллеры динамической памяти;
- синтез звука;
- фазоимпульсная и широтно-импульсная модуляция;
- дельта-сигма АЦП;
- спектральный анализ;
- передача/распознавание тонового набора в телефонии;
- квадратурный кодер/декодер;
- видеоконтроллер.

Основные характеристики микроконтроллеров SX приведены в табл. 1.4.

Таблица 1.4

Название	Число вводов	Число линий ввода/вывода	Память программ (слов)	RAM (байт)
SX18AC/SO	18	12	2К	136
SX18AC/DP	18	12	2К	136
SX20AC/SS	20	12	2К	136
SX28AC/SO	28	20	2К	136
SX28AC/DP	28	20	2К	136
SX28AC/SS	28	20	2К	136

1.2.4. Микроконтроллеры фирмы Motorola

Motorola предлагает самую широкую в мире номенклатуру микроконтроллеров, охватывающую практически все области применения и включающую в себя около 300 моделей от простейших дешевых до высокопроизводительных 32-разрядных микроконтроллеров с RISC-ядром и мощной периферией. Микроконтроллеры фирмы Motorola отличаются высоким качеством и надежностью.

Каждый из выпускаемых микроконтроллеров относится к одной из относительно крупных групп или семейств.

Семейство HC05 содержит наибольшее количество модификаций микроконтроллеров (около 180). Области применения микроконтроллеров семейства HC05 являются самые разнообразные устройства связи, автомобильной и бытовой электроники, промышленного управления, компьютерной периферии.

В состав микроконтроллеров семейства HC05 входят: ПЗУ всех типов, ОЗУ, таймеры, АЦП, ШИМ, контроллеры ЖКИ и других дисплеев.

Основные характеристики некоторых типов микроконтроллеров семейства HC05 приведены в табл. 1.5.

Семейство HC08 является следующим шагом в развитии микроконтроллеров фирмы Motorola для массовых приложений и характеризуется повышенной в 5–10 раз производительностью процессора, совместимого с процессором HC05.

Введены дополнительные команды и методы адресации, а также новые функции, как прямой доступ к памяти и элементы цифровой обработки сигналов.

Таблица 1.5

Название	68HC705B5	68HC705B16	68HC705B32	68HC705C8A
ПЗУ	6К	15К	32К	8К
ОЗУ	176	352	528	304
EPROM	–	256	256	–
Таймер	16 bit	16 bit	16 bit	16 bit
I/O	24	32	32	24
Последовательный интерфейс	+	+	+	+
АЦП	8 ch, 8 bit	8 ch, 8 bit	8 ch, 8 bit	–
ШИМ	2 ch, 8 bit	2 ch, 8 bit	2 ch, 8 bit	–
Тактовая частота	4 МГц	4 МГц	4 МГц	4 МГц

Набор модулей в настоящее время включает в себя различные типы ПЗУ и ОЗУ, таймеры, последовательные интерфейсы, АЦП, контроллеры ЖКИ, контроллеры ПДП, силовые и высоковольтные ключи и т. д.

Основные технические характеристики некоторых контроллеров данного семейства представлены в табл. 1.6.

Таблица 1.6

Название	68HC 08 ABO	68HC 08 AZO	68HC 708 LN56	68HC 708 MP16	68HC 708 XL36	68HC 908 AT32
ПЗУ	–	–	56К	16К	36К	32К FLASH
ОЗУ	1К	1К	1280	512	1К	1К
EPROM	512	512	–	–	–	512
Таймер	16 bit, 4 ch	16 bit, 4 ch	16 bit, 4 ch	16 bit, 4 ch	16 bit, 4 ch	16 bit, 6 ch
I/O	50	50	51	37	54	50
Последовательный интерфейс	+	+	+	+	+	+
АЦП	8 bit, 8 ch	8 bit, 8 ch	8 bit, 4 ch	8 bit, 10 ch	–	8 bit, 15 ch
ШИМ	–	–	–	–	–	–
ПДП	–	–	–	–	+	–

Семейство HC11, в отличие от относительно специализированных микроконтроллеров «заказных», содержит около 40 универсальных и

высокопроизводительных микроконтроллеров. Периферийные функции представлены многофункциональными таймерами, АЦП (до 12 каналов и 10 разрядов), встроенным сопроцессором, ускоряющим выполнение умножения и деления на порядок, ШИМ и ЦАП, последовательными интерфейсами, контроллером ПДП, синтезатором тактовой частоты и другими функциями.

Наиболее характерные особенности семейства HC11:

- два 8-битовых или один 16-битовый аккумулятор;
- два 16-битовых индексных регистра;
- программно управляемый режим пониженного энергопотребления;
- операции умножения 8×8 и деление $16/16$;
- внутренняя тактовая частота до 4 МГц.

Некоторые модели семейства содержат встроенный математический сопроцессор, выполняющий 16-битовые операции умножения и деления в 10 раз быстрее, чем процессор.

Технические характеристики некоторых контроллеров семейства HC11 приведены в табл. 1.7.

Таблица 1.7

Название	68HC11AO	68HC711EA9	68HC711E32	68HC711M2
ПЗУ	–	12К	32К	32К
ОЗУ	256	512	2К	1278
EPROM	–	512	512	–
Таймер	16 bit	16 bit	16 bit	16 bit
Последовательный интерфейс	+	+	+	+
АЦП	8 bit, 8 ch			
Тактовая частота	3 МГц	2 МГц	4 МГц	4 МГц
Арифметический сопроцессор	–	–	–	16 bit арифметический сопроцессор ПДП

1.2.5. Микроконтроллеры семейства Z8 фирмы Zilog

Фирма Zilog имеет развитое семейство микроконтроллеров Z8, позволяющее решать различные задачи от создания недорогих автономных устройств и «интеллектуальных» датчиков до управления сложными объектами.

Условно все микроконтроллеры семейства Z8 можно разделить на:

- стандартные;
- стандартные с расширенным набором универсальных функций;
- широкого применения (ССР);
- работающие с пониженным питанием (Z86Lxx);
- специализированные с цифровым сигнальным процессором (DSP);
- устройства с внутрисистемной эмуляцией.

Стандартные микроконтроллеры семейства Z8 имеют все основные функции, существующие в 8-разрядных встраиваемых микроконтроллерах наиболее популярных семейств: MCS-51 фирмы INTEL, PIC17 фирмы Microchip.

Стандартные микроконтроллеры с расширенным набором универсальных функций имеют:

- расширение внутренней памяти программ ROM до 16/22К;
- расширение числа портов ввода/вывода и их функций;
- дополнительный блок таймеров/счетчиков;
- устройство аппаратного умножения 16×16 (17 мкс) и деления $32/16$ (2,5 мкс);
- контроль формата в аппаратном UART;
- аналого-цифровые компараторы или 8-разрядный АЦП и 8-разрядный ЦАП.

Микроконтроллеры широкого применения Z8ССР – это малопотребляющие КМОП-контроллеры, используемые для создания интеллектуальных датчиков, управления низкоскоростными процессорами, управления бытовой техникой.

Микроконтроллеры с пониженным напряжением питания (286L03/06, Z86L29 и т. д.) $V_{cc} = 2,0-3,6$ В. Этим достигается значительное снижение потребляемой мощности, особенно на низких частотах (до 2,4 мВт при частоте 8 МГц – в рабочем режиме и до 2 мкВт – в режиме ожидания с пониженным энергопотреблением).

Это позволяет использовать их в системах, критичных к энергопотреблению, например в системах дистанционного управления.

Микроконтроллер, совмещенный с цифровым сигнальным процессором, например Z86C95, содержит базовое ядро микроконтроллера Z8 без внутреннего ПЗУ и встроенный 16-разрядный DSP (цифровой сигнальный процессор).

Z86C95 предназначен для оцифровки аналоговых сигналов и их цифровой обработки в реальном времени. Он имеет следующие характеристики:

- 512 байт внутреннего ОЗУ для программ и данных DSP-процессора;
- 8 каналов 8-битового АЦП (время преобразования до 1,5 мкс);
- 8-разрядный ЦАП;
- 1 канал ШИМ 40/80 кГц;
- аппаратное умножение 16×16 (1,7 мкс);
- деление 32/16 (2,5 мкс);
- 3 таймера-счетчика.

Основные технические характеристики микроконтроллеров семейства Z8 приведены в табл.1.8.

Таблица 1.8

Название	Z86C04/E04/L04	Z86C21/E21	Z86C95/295	Z86C84
ROM	1	8	–	4
RAM	125	236	236	237
Тактовая частота, МГц	8, 12	4, 12, 16	24, 33, 40	12, 16
I/O	14	32	16+BUS	17
Таймер	2+WDT	2	3	2+WDT
Компараторы	2	–	АЦП+ЦАП	2+АЦП+2 ЦАП
Последовательный интерфейс	–	UART	UART	–

В табл. 1.9 представлены сравнительные технические характеристики микроконтроллеров наиболее популярных семейств: MCS-51 фирмы Intel, PIC17 фирмы Microchip и Z86C11 фирмы Zilog.

Результаты сравнения показывают, что аппаратные возможности этих микроконтроллеров во многом схожи и каких-либо явных преимуществ не имеет ни одно из семейств.

Таблица 1.9

Название	Z86C11	180C51	PIC17C42
Внутреннее ПЗУ	4К×8	4К×8	2К×16
Регистры общего назначения	236	128	232
Регистры специальных функций: управляющие ввода/вывода	16/4	16/4	48
Ввод/вывод: сигналы порты	32 4×8 бит	32 4×8 бит	33 5 портов
Прерывания: источники внешние источники векторы приоритеты маскируемые	8 4 6 48 6	5 2 5 2 5	11 (внутренних и внешних) 4 – 4
Внешняя память (ROM+RAM)	120 Кбайт	124 Кбайт	64 К слов (только память программ)
Стек: указатель стека внутренний стек внешний стек	16 бит + +	8 бит + +	Аппаратный + –
Таймеры-счетчики Предделитель	2 по 8 бит 2 по 6 бит	2 по 16 бит или 2 по 8 бит нет при 16 и 8 бит	3 по 16 бит –
Индексные регистры	236, любой регистр общего назначения	Используется аккумулятор при 8-битовом смещении	Используется 16-битовый индексный регистр

Окончание табл. 1.9

Название	Z86C11	I80C51	PIС17C42
Последовательный интерфейс: полный дуплекс (UART) Прерывания: при передаче при приеме Двойное буферирование	+ + + При приеме	+ Одно в обоих случаях При приеме	+ + + Прием/передача
Быстродействие: время выполнения команды время для длинной команды	1,0 мкс при 16 МГц 2,0 мкс при 16 МГц	1,0 мкс при 16 МГц 2,8 мкс при 16 МГц	0,25 мкс 0,5 мкс при 16 МГц
Тактовая частота	12 и 16 МГц	12 и 16 МГц	16 МГц
Режим пониженного энергопотребления	Сохраняется 256 регистров RAM	Сохраняются первые 128 регистров	Сохраняются состояния портов ввода/вывода
Сохранение контекста	Сохраняются РС и флаги	Сохраняется РС; можно сохранить все регистры	Сохраняется РС
Поддержка разработки	64-выв. кристалл-эмулятор Z86C12; 40-выв. без ПЗУ Z86C91	Эмуляция без отключения кристалла; 40-выв. без ПЗУ i87C51	

2. АРИФМЕТИЧЕСКИЕ ОСНОВЫ МИКРОПРОЦЕССОРНЫХ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ

В данном разделе приводятся несколько примеров программирования простых задач, которые окажутся полезными при проектировании гораздо более сложных прикладных программ.

Программы написаны в системе команд микроЭВМ К1816.

2.1. Сложение двух многобайтовых чисел

Оба слагаемых располагаются во внутреннем ОЗУ, начиная с младшего байта. Начальные адреса слагаемых заданы в R0 и R1. Количество байт в числах задано в регистре R2. Результат размещается по месту первого слагаемого.

Ассемблерная программа имеет следующий вид:

Мнемокод	Комментарии
CLRC	Сброс переноса
M1:MOV A,@R0	Загрузка в аккумулятор текущего байта 1-го слагаемого
ADDC A,@R1	Сложение байта с учетом переноса
MOV@R0,A	Размещение байта результата
INC R0	Продвижение указателей
INC R1	Организация цикла,
DJNC R2,M1	если не все байты просуммированы
RET	Выход из подпрограммы

В данной программе при сложении чисел без знака на переполнение укажет флаг C, а в случае сложения чисел со знаком – флаг OV (К1816BE51).

2.2. Умножение однобайтовых чисел

Рассмотрим программу MULT умножения 8-битовых целых чисел без знака младшими разрядными вперед со сдвигом частичной суммы

вправо. Предполагается следующее распределение внутренних регистров микроЭВМ К1816: множитель находится в ячейке памяти внутреннего ОЗУ, адресуемой через регистр R1; адрес счетчика циклов располагается в регистре R2 и, наконец, 16-разрядное произведение размещается в ячейках памяти внутреннего ОЗУ, адресуемых регистрами R0 и R3.

Мнемокод	Комментарии
MULT: MOV @R3,#0	Обнуление старшего байта произведения
MOV R2,#9	Установка счетчика
M1: MOV A,@R0	Загрузка множителя в аккумулятор
RRC	Очередной бит множителя во флажке C
MOV @R0,A	Возвращение множителя
DJZ R2,M3	Организация цикла, если умножение не закончено
MOV A,@R3	Старший байт произведения в аккумуляторе
JNC M2	Бит множителя равен 0
ADD A,@R1	Суммирование множимого
M2: RRC	Сдвиг частичной суммы
MOV @R3,A	Возвращение старшего байта произведения
JMP M1	Умножение на следующий бит
M3: RET	Выход из подпрограммы умножения

В начале программы производится обнуление ячейки памяти внутреннего ОЗУ с адресом в регистре R3, в которой будут образованы старшие 8 бит произведения. Затем осуществляется установка счетчика числа циклов суммирования частичных сумм. После этого множитель передается в аккумулятор и командой RRC очередной бит множителя передается во флажок C.

Если бит равен единице, то командой ADD осуществляется сложение множимого со старшим байтом произведения, а если бит равен нулю, сложение не производится.

В данной программе младшие биты произведения замещают освободившиеся биты множителя. При этом длина программы составляет 13 байт.

2.3. Умножение однобайтовых чисел со знаком в прямом коде

В данной программе MULTS используется следующее распределение ячеек памяти резидентного ОЗУ микроЭВМ серии К1816: множи-

мое, множитель и счетчик циклов адресуются через регистры общего назначения R0, R1 и R2 соответственно; 16-разрядное произведение размещается в ячейках памяти внутреннего ОЗУ, адресуемых регистрами R0 и R3.

Алгоритм программы содержит этапы:

1) получение знака произведения из знаков множимого и множителя и его сохранение в регистре R4;

2) очистка знаковых битов множителя и множимого (получение модулей);

3) перемножение модулей множимого и множителя методом суммирования частных произведений: путем сдвига произведения вправо и прибавления множимого к произведению для каждого ненулевого бита множителя, выдвигаемого вправо, т. е. младшими разрядами вперед;

4) помещение знака в старший бит произведения.

Ассемблерная программа умножения 8-битовых чисел со знаком в прямом коде имеет следующий вид:

Мнемокод	Комментарии
MULTS: MOV F,@R0	Загрузка множителя в аккумулятор
XRL A,@R1	Определение знака произведения
ANL A,#80	Выделение знакового бита
MOV R4,A	Загрузка знака произведения в регистр R4
Удаление знаков сомножителей для получения их абсолютных значений	
MOV A,@R0	Загрузка множителя в аккумулятор
ANT A,#7F	Очистка знакового бита множимого
MOV @R0,A	Возврат модуля множимого в ОЗУ
MOV A,@R1	Загрузка множителя в аккумулятор
ANT A,#7F	Очистка знакового бита множителя
MOV @R1,A	Возврат модуля множителя в ОЗУ
CALL MULT	Вызов подпрограммы MULT умножения целых чисел без знака
MOV A,R4	Загрузка знака произведения в аккумулятор
ORL A,@R0	Перенос знака в старший бит произведения
MOV @R0,A	Возвращение старшего байта произведения в ОЗУ

2.4. Умножение однобайтовых чисел в дополнительном коде

В данной задаче оба сомножителя X и Y представляются целыми числами в дополнительном коде. При этом возможны следующие четыре случая.

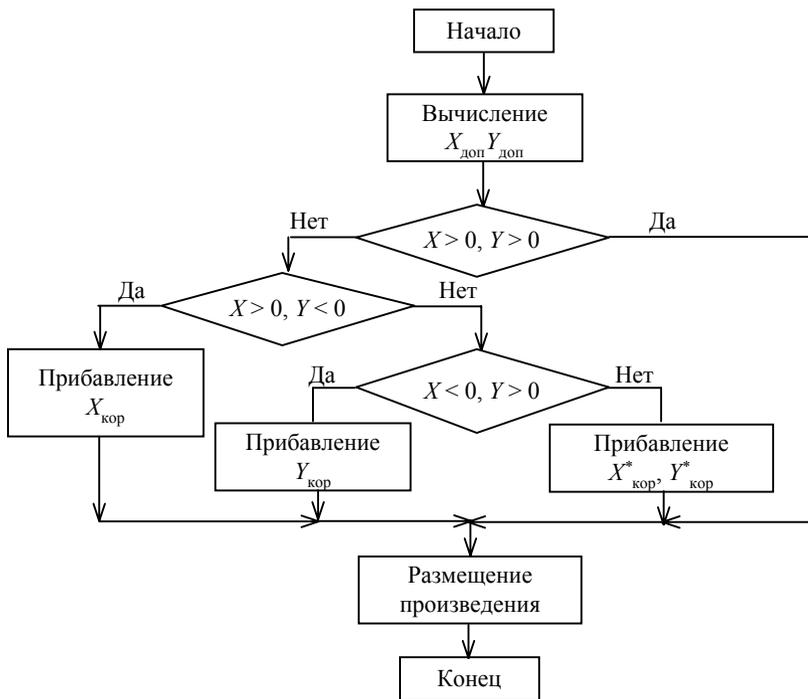


Рис. 2.1

1. Если перемножаются положительные целые числа ($X > 0$, $Y > 0$), то 16-битное произведение P в дополнительном коде совпадает с произведением сомножителей:

$$P_{\text{доп}} = XY.$$

2. Если множимое $X > 0$, а множитель $Y < 0$, то в арифметическое устройство множимое поступает непосредственно (дополнительный код положительного числа совпадает с его прямым кодом), а отрицательный множитель в дополнительном коде $Y_{\text{доп}} = 2^8 - |Y|$. При этом 16-битное произведение в дополнительном коде определяется по выражению

$$P_{\text{доп}} = XY_{\text{доп}} + X_{\text{кор}},$$

где $X_{\text{кор}} = [-X]_{\text{доп}} \cdot 2^8$ – корректирующее слагаемое.

3. При умножении отрицательного множимого $X < 0$ на положительный множитель $Y > 0$, представленных соответственно дополнительным и прямым кодами, 16-битное произведение в дополнительном коде определяется соотношением

$$P_{\text{доп}} = YX_{\text{доп}} + Y_{\text{кор}}.$$

где $Y_{\text{кор}} = [-Y]_{\text{доп}} \cdot 2^8$ – корректирующее слагаемое.

4. Когда $X < 0$ и $Y < 0$, непосредственное перемножение дополнительных кодов строится в соответствии с формулой

$$P_{\text{доп}} = X_{\text{доп}} Y_{\text{доп}} + X_{\text{кор}}^* Y_{\text{кор}}^*,$$

где $X_{\text{кор}}^* = |X| \cdot 2^8$, $Y_{\text{кор}}^* = |Y| \cdot 2^8$.

На рис. 2.1 изображена блок-схема программы умножения.

3. СИСТЕМА ПАМЯТИ МИКРОПРОЦЕССОРНЫХ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ

3.1. Классификация запоминающих устройств

Высокие технико-экономические показатели систем памяти (СП) микропроцессоров могут быть обеспечены скорее в результате рациональной организации СП, чем в результате повышения емкости и быстродействия их подсистемы.

Укрупненно запоминающие устройства (ЗУ) делятся на два класса: оперативные ЗУ (ОЗУ), у которых как запись, так и считывание являются рабочими режимами; постоянные ЗУ (ПЗУ), у которых рабочим режимом является только считывание, а запись информации осуществляется особым образом до включения БИС в работу.

В зависимости от способа хранения информации ОЗУ делятся на два типа: статические ОЗУ, ячейки которых выполнены на основе триггерных схем, сохраняющих записанную в них информацию в течение всего времени подачи напряжения питания; динамические ОЗУ, ячейки которых теряют со временем записанную в них информацию, так что ее необходимо периодически возобновлять (регенерировать).

В зависимости от способа записи информации ПЗУ делятся на два класса: ПЗУ, программирование которых производится с помощью специальной маски при изготовлении микросхем, и ПЗУ, программирование которых осуществляется пользователем однократно с помощью программаторов, которые вырабатывают необходимые электрические импульсы; ППЗУ (перепрограммируемые), программирование которых может осуществляться пользователем многократно, причем стирание информации может производиться либо электрическими импульсами, либо с помощью ультрафиолетовых лучей.

Программы, реализующие алгоритмы функционирования микропроцессорного вычислителя, хранятся в ПЗУ. Загрузка программ в ПЗУ производится после их отладки и опытной эксплуатации в электронном устройстве.

На стадии отладки и опытной эксплуатации используются энергонезависимые перепрограммируемые постоянные запоминающие устройства (ППЗУ), позволяющие разработчику многократно вносить изменения в отлаживаемую программу.

Оперативные запоминающие устройства предназначены для хранения промежуточных результатов работы программы. Классификация ОЗУ приведена на рис. 3.1.

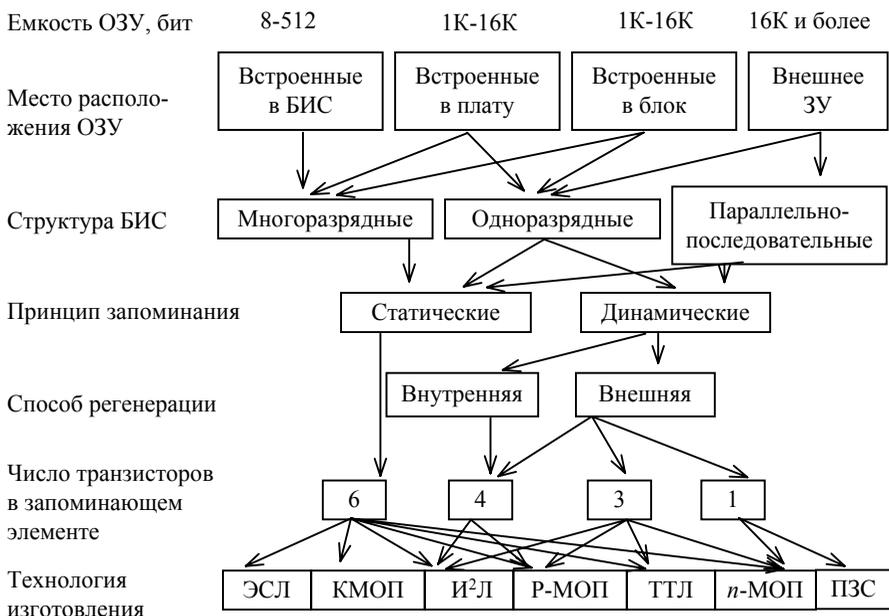


Рис. 3.1

Для снижения общей стоимости следует подбирать информационную емкость памяти, равную требуемой емкости памяти для хранения программ и промежуточных результатов.

Для серийно выпускаемых микропроцессорных вычислителей используются масочно-программируемые ПЗУ. Они выполняются как по МОП, так и по биполярной технологии и загружаются отлаженными программами непосредственно поставщиками.

Выпуск масочно-программируемых ПЗУ экономически целесообразен при крупносерийном производстве (десятки тысяч в год). К достоинствам масочно-программируемых ПЗУ относятся высокая

степень интеграции, низкая потребляемая мощность, хорошие временные параметры.

Самостоятельная загрузка программ производится в ПЗУ с пережигаемыми перемычками. Стоимость таких ПЗУ значительно выше, чем масочно-программируемых, но возможность самостоятельной загрузки делает их весьма распространенными.

Перепрограммируемые постоянные запоминающие устройства выполняются с использованием разных технологий. Большое распространение получили ППЗУ, выполненные на лавинно-инжекторных МОП-приборах с плавающими затворами, электрический контакт к которым отсутствует. Заряд инжектируется в эти затворы электронами с высокой энергией в режиме программирования при подаче на $p-n$ -переход напряжения 28 В. Стирание информации выполняется путем облучения кристалла ультрафиолетовым светом в течение 20 мин.

Как отмечалось выше, ОЗУ подразделяются на динамические и статические. Динамические ОЗУ характеризуются большой информационной емкостью и низкой стоимостью. Информация в динамических ОЗУ хранится в виде электрических зарядов на затворах МОП-транзисторов, при этом необходима их периодическая регенерация.

Статические ОЗУ имеют меньшую степень интеграции, являются более дорогими, чем динамические ОЗУ. Однако простота использования статических ОЗУ в составе микропроцессорного устройства с малым объемом памяти определяет их значительную распространенность.

При выборе элементарной базы для построения памяти МПУ предварительно оценивается требуемое быстродействие элементов памяти. Обычно в основу оценки берут временные характеристики МП. Для снижения стоимости во многих случаях целесообразно выбирать элементы с относительно низким быстродействием. Экономичным является сочетание БИС памяти с номинальным и пониженным быстродействием. Первые БИС применяются для сохранения программ, вторые – для промежуточных, вводимых данных. При этом практически не снижается общее быстродействие устройства. Так обычно в электронном устройстве микропроцессор работает 20–50% общего времени. При этом время обращения к памяти составляет 10–30% общего времени выполнения программ. Если в этих условиях использовать память с быстродействием в два раза меньшим, чем быстродействие МП, то производительность устройства падает только на 25%. Основные характеристики БИС памяти приведены в табл. 3.1 [3].

Таблица 3.1

ЗУ	Обозначение	Технология	Организация	Максимальное время выборки, нс	Потребляемая мощность, мВт/бит
ПЗУ	K505PE3	НСТЛМ	512×8	1,5	0,25
	K556PE6	–	2048×8	–	–
	K556PE5	–	2048×8	850	0,03
	K568PEA	<i>n</i> -МОП	2048×8	850	0,03
	K155PE21	ТТЛ	256×4	60	0,4
	K556PE4	ТТЛШ	256×4	90	0,6
	K500PE149	ЭСТЛ	256×4	35	0,68
ППЗУ	K558PE1	Стеклообразный полупроводник	256×8	5000	0,15
	K505PP1	–	256×8	–	–
	K558PP1	И ² Л	1048×1	5000	0,15
	K558PP11	То же	1024×1	5000	0,3
ОЗУ	K565PY1	<i>n</i> -МОП	4096×1	200	0,12
	K565PY3	То же	4096×1	200	0,06
	K565PY2	– » –	1638×1	450	0,4
	K522PY1	– » –	1024×1	800	0,03
	K541PY1A	И ² Л	4096×1	180	0,16
	K500PY415	ЭСТЛ	1024×1	30	0,7

3.2. Организация памяти

При построении систем памяти наибольшее распространение получили БИС ЗУ с конфигурацией $n \times k$ ($n = 256, 512, 1028, 2048, 4096; k = 1, 4, 8$). Память микропроцессорных вычислителей обычно имеет разрядность, равную или кратную разрядности микропроцессоров. Поэтому для 8-разрядного микропроцессора необходима память с длиной слова 8 бит.

Восьмиразрядную память используют также 4-разрядные и секционные микропроцессоры с микропрограммным управлением.

Необходимая длина слова памяти достигается параллельным включением m БИС памяти, где m – длина слова в битах (обычно $m = 8, 12, 16, 24, 32, 64$), в 8-разрядном микропроцессоре для построения памяти с организацией $n \times 1$. Совокупность восьми одноразрядных линий данных всех восьми БИС ЗУ образует 8-разрядную шину данных системы памяти. Линии адреса и чтения/записи всех БИС включаются параллельно для одновременного обращения ко всем БИС ЗУ.

Часто разрядность шины адреса микропроцессора в два раза больше разрядности ее шины данных. Так, в 8-разрядном микропроцессоре Intel 8085 применение 16-разрядного адреса обеспечивает адресацию 65536

байтов памяти. Два байта адреса называются младшим и старшим байтами адреса. Возможно и другое разделение адресного кода. Младшие разряды адреса A_0 – A_7 называют адресом слова, а старшие разряды A_8 – A_{16} называют адресом страницы. Таким образом, можно адресовать 256 страниц по 256 слов (байтов) в каждой. На рис. 3.2 показана организация ППЗУ емкостью 256×8 бит на основе двух БИС ППЗУ K558PE2 емкостью 256×4 бит. Организация ОЗУ емкостью 2048 бит на основе БИС ОЗУ K565PY2 емкостью 1024×1 бит представлена на рис. 3.3.

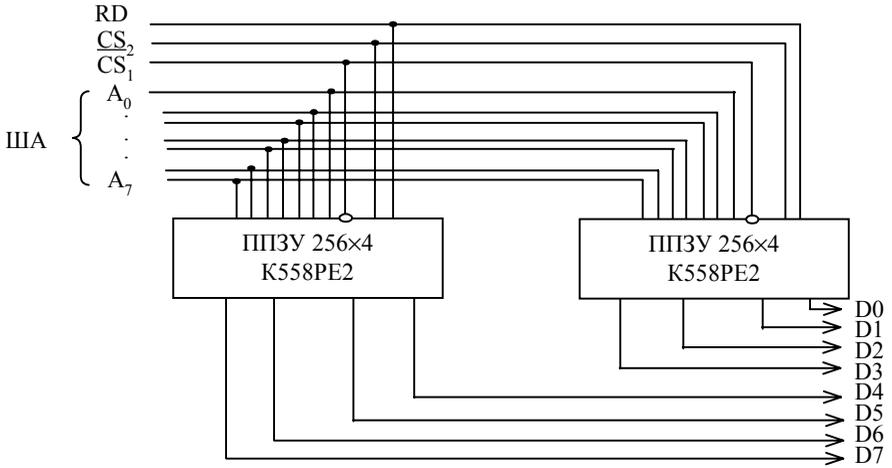


Рис. 3.2

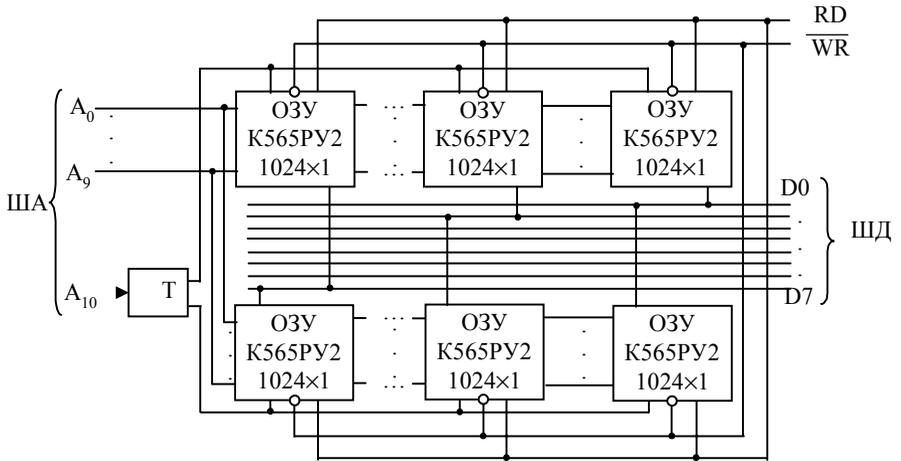


Рис. 3.3

4. ИНТЕРФЕЙС ВВОДА-ВЫВОДА ИНФОРМАЦИИ

4.1. Общие сведения о микросхемах цифроаналоговых и аналого-цифровых преобразователей

Широкое использование микропроцессоров в схемах связи, обработки информации, робототехнике, бытовой радиоэлектронной аппаратуре, управлении технологическими процессами и т. п., состояние которых в большинстве случаев характеризуется непрерывными функциями времени, выдвигает на первый план проблему связи микропроцессоров с объектами.

Поэтому в процессе обработки таких функций важная роль отводится операции преобразования непрерывных (аналоговых) сигналов в цифровую форму и обратно. Это осуществляется при помощи аналого-цифровых (АЦП) и цифроаналоговых преобразователей (ЦАП).

Аналого-цифровой преобразователь обеспечивает сопряжение источников аналоговых сигналов (например, чувствительных элементов систем управления, различных датчиков) с микропроцессорными устройствами обработки, а ЦАП предназначены, в основном, для вывода из процессора результатов обработки информации на управляемые объекты.

В системах обработки информации используются следующие виды аналого-цифровых и цифроаналоговых преобразователей: “угол – код”, “фаза – код”, “напряжение – код”, “время – код” и др.

Совершенно очевидно, что преимущества цифровых методов обработки информации могут быть реализованы лишь в том случае, когда АЦП и ЦАП не вносят в эту обработку ограничений по точности и быстродействию. Эти ограничения удается свести к минимуму при использовании интегральных преобразователей.

В дальнейшем будут рассматриваться наиболее распространенные АЦП и ЦАП типа “напряжение – код” и “код – напряжение” соответственно.

Методы построения и характеристики АЦП и ЦАП подробно изложены в литературе [6].

В настоящее время преобразование с допустимой погрешностью 0,05–0,1% (10–12 двоичных разрядов) обеспечено широкой номенклатурой серийных монолитных микросхем, отличающихся быстродействием и функциональными возможностями.

В этом классе выпускаются [8]:

- преобразователи малого быстродействия с временем установления 5–10 мкс для ЦАП и временем преобразования 5–10 мкс на разряд для АЦП последовательных приближений (серий К572 и К594);

- преобразователи среднего быстродействия с временем установления 0,5–1,0 мкс для ЦАП (К1108ПА1) и временем преобразования 10–20 мкс для АЦП последовательных приближений (К1113ПВ1);

- быстродействующие преобразователи с временем установления 100 нс для ЦАП и временем преобразования 1 мкс для АЦП последовательных приближений (К1108ПВ1);

- сверхбыстродействующие преобразователи – АЦП непосредственно считывания (параллельные) с временем преобразования десятки наносекунд (6- и 8-разрядные АЦП серии К1107, десятиразрядные АЦП МР7685 фирмы “Микро-Пауер”, США).

Средства стыковки с вычислительными системами, в частности с микроЭВМ, зачастую включаются в состав микросхем АЦП и ЦАП. Многие ЦАП содержат буферные регистры, а АЦП – шинные формирователи с тремя состояниями, управляемые стандартными сигналами обмена по магистрали (“Чтение”, “Запись”, “Адрес”), и узлы синхронизации обмена, обеспечивающие прием и генерацию сигналов запуска и готовности. При отсутствии подобных блоков микросхемы АЦП и ЦАП сопрягаются с микроЭВМ через параллельные адаптеры или порты ввода-вывода. В литературе [6] рассмотрен вопрос сопряжения АЦП и ЦАП как в программном режиме, так и в режиме прерывания с микропроцессором серии КР580 с использованием параллельного периферийного адаптера КР580ВВ55.

4.2. Справочные сведения о микросхемах ЦАП и АЦП

Сведения о характеристиках ЦАП (табл. 4.1) и АЦП (табл. 4.2), выпускаемых отечественной промышленностью, приведены ниже.

В табл. 4.1 и 4.2 приняты следующие условные сокращения: $\delta_{\text{дд}}$ – дифференциальная нелинейность; $t_{\text{уст}}$ – время установления; ед. МР – единицы младшего разряда.

Таблица 4.1

Тип микро-схемы	Число двоичных разрядов	$\delta_{\text{дл}} \%$ (ед. МР)	$t_{\text{уст}}^{\text{мкс}}$	Технология	Совместимость с ИМС	Число источников питания
K427ПА1	15	$\pm 0,006$	30	КМДП, бипол.	ТТЛ, КМДП	3
K427ПА2	16	0,0015	5	То же	То же	2
K427ПА3	16	0,0015	10	- » -	- » -	2
K427ПА4	16	0,0015	20	- » -	- » -	4
K572ПА1 (А - Г)	10	$\pm 0,1$	5	КМДП	- » -	1
K572ПА2 (А - В)	12	$\pm 0,025$	15	То же	- » -	2
K594ПА1	12	$\pm 0,012$	3,5	Биполярные	- » -	2
K1108ПА1 (А, Б)	12	$\pm 0,024$	0,4	То же	ТТЛ	2
K1108ПА2	8	$\pm 0,2$	1,5	- » -	ЭСЛ, ТТЛ	2
K1108ПА3	6	$\pm 0,006$	0,1	- » -	То же	2
K1113ПА1	12	$\pm 0,25$	2,5	- » -	ТТЛ	2
K1118ПА1	8	$\pm 0,75$	0,02	- » -	ЭСЛ	1
K1118ПА2 (А, Б)	10	$\pm 0,1$	0,08	- » -	ЭСЛ, ТТЛ	2
K1118ПА3 (А, Б)	8	$\pm 0,5$	0,005	- » -	То же	2
K1118ПА4	10	$\pm 0,35$	0,03	- » -	ЭСЛ	1
KM11488ПА1	10	$\pm 0,75$	1	- » -	ТТЛ	1

Таблица 4.2

Тип микро-схемы	Число двоичных/десятичных разрядов	$\delta_{\text{дл}} \%$ (МР)	$t_{\text{проб}}^{\text{мкс}}$	Технология	Совместимость с микро-схемами	Число источников питания	Дополнительные функциональные возможности
K572ПВ1 (А - В)	12	$\pm 0,1$	170	КМДП	ТТЛ	2	СМ
K572ПВ2 (А - В)	(3,5)	$\pm 0,15$	300	То же	КМДП	2	Ф3
KP572ПВ3	8	$\pm 0,75$	7,5	- » -	ТТЛ	1	МК, СМ
K572ПВ4	8	$\pm 0,5$	25	- » -	То же	1	СМ
K1107ПВ1	6	$\pm 0,5$	0,1	Бипол.	- » -	2	Ф3
K1107ПВ2	8	± 1	0,1	То же	- » -	2	То же
K1107ПВ3	6	$\pm 0,5$	0,02	- » -	- » -	2	НР
K1107ПВ4	8	$\pm 0,5$	0,03	- » -	ТТЛ, ЭСЛ	2	То же
K1107ПВ5	6	$\pm 0,75$	0,02	- » -	ЭСЛ	4	- » -
M1107ПВ5 (А - В)	6	$\pm 0,75$	0,02	- » -	То же	4	- » -
K1108ПВ1 (А, Б)	10	$\pm 0,75$	1,0	- » -	ТТЛ	2	Ф3
K1108ПВ2	12	± 1	0,9	- » -	То же	2	То же
K1113ПВ1	10	$\pm 0,5$	25	- » -	- » -	1	- » -
KM1126ПВ1	8	$\pm 0,75$	30	- » -	- » -	1	- » -

При этом под дифференциальной нелинейностью понимают максимальное отклонение разности двух аналоговых сигналов, соответствующих соседним кодам, от значения МР; время установления $t_{уст}$ – интервал времени от подачи входного кода до момента, когда выходной сигнал достигнет установившегося значения с заданной погрешностью (обычно $\pm 0,5$ МР). Это время определяет общее быстродействие ЦАП.

4.3. Подключение АЦП и ЦАП к микропроцессору

Схема подключения АЦП серии К1113ПВ1 и ЦАП серии К572ПА1 к однокристальной микроЭВМ серии К1816ВЕ48 показана на рис. 4.1. В качестве интерфейса использован многорежимный буферный регистр К589ИР12. Микросхема К1113ПВ1 – 10-разрядный функционально полный АЦП последовательного приближения. Выполнен по КМДП-технологии.

Основные параметры преобразователя при $U_{П1} = 5$ В, $U_{П2} = -15$ В: $\delta_{л} = \delta_{лд} = \pm 0,1\%$; $t_{прб} = 25$ мкс; $R_{вх} = 10$ кОм; $I_{пот} = 30$ мА. Имеет десятиразрядную входную шину D_{0-9} ; $\Gamma/\bar{\Pi}$ – вход гашения и преобразования; U_c – аналоговый вход; Упр.сдв.0 – управление сдвигом нуля; ГД – выход готовности данных.

Запуск преобразователя осуществляется подачей логического 0 на вход $\Gamma/\bar{\Pi}$. Через 25 мкс на выводе ГД появляется сигнал высокого уровня и информация поступает на цифровые выходы. При подаче на вывод $\Gamma/\bar{\Pi}$ логической 1 в регистре последовательного приближения информация стирается и АЦП подготовлен к новому преобразованию.

Входные буферные устройства микросхемы имеют три состояния, позволяющие использовать их для связи с шиной данных микропроцессора. Уровни выходных сигналов соответствуют ТТЛ-микросхемам.

Аналого-цифровой преобразователь может работать с однополярными входными сигналами до 11 В и двуполярными сигналами до $\pm 5,5$ В. Во втором случае вывод Упр.сдв.0 заземляется.

Микросхема К572ПА1 – 10-разрядный преобразователь двоичного кода в ток (ЦАП). Выполнен по КМДП-технологии.

Основные параметры преобразователя при $U_{П} = 15$ В, $U_{ОП} = 10,24$ В: $\delta_{лд} = \pm 0,1\%$; $t_{уст} = 5$ мкс; $I_{пот} = 2$ мА; $U_{вых} = 10$ В. ЦАП имеет десять цифровых входов D_{0-9} ; аналоговые выходы I, \bar{I} ; R – вход отрицательной обратной связи.

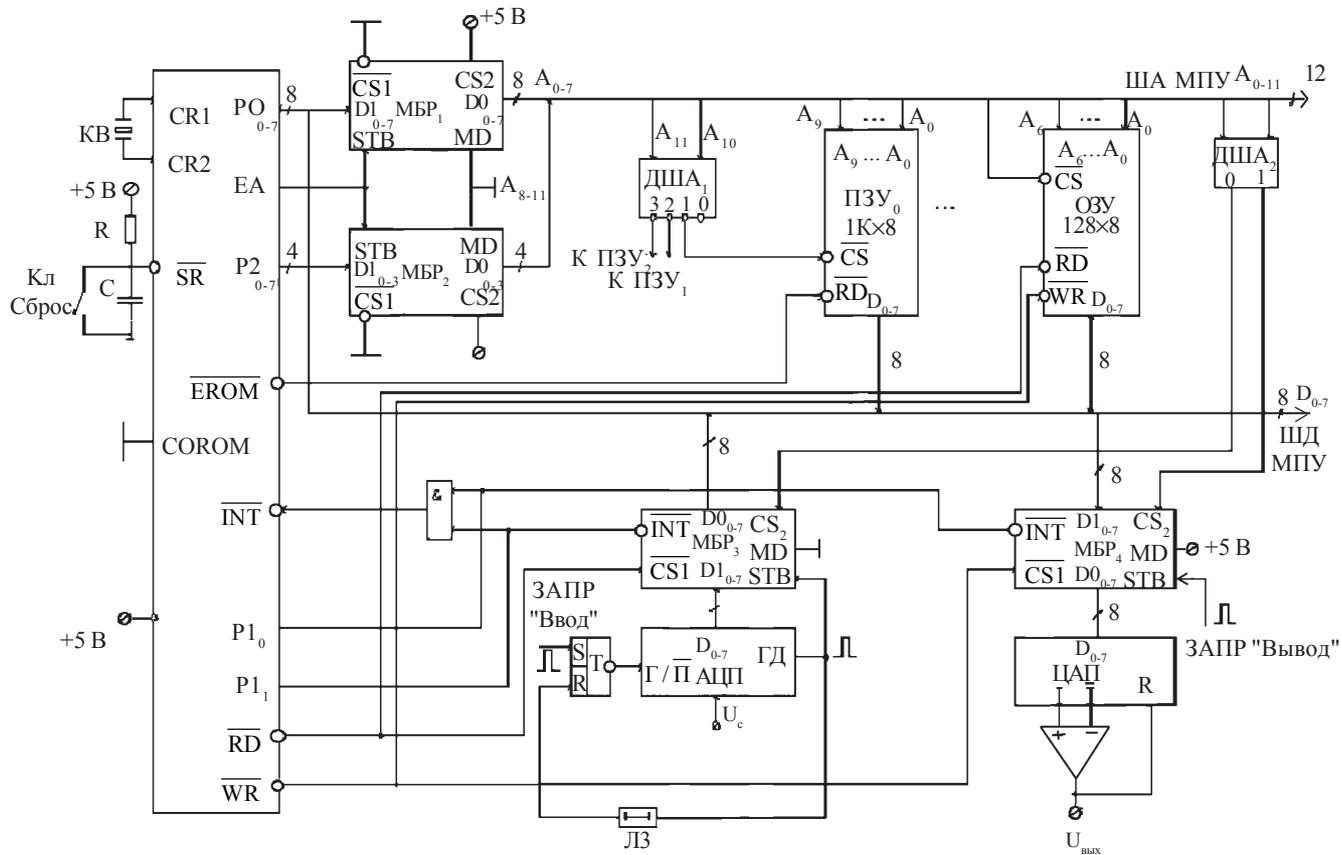


Рис. 4.1

Функциональная схема преобразователя содержит резистивную матрицу $R - 2R$, токовые ключи и согласующее устройство. Для преобразования тока в напряжение на выходе микросхемы устанавливается внешний операционный усилитель ОУ.

В состав устройства обработки информации, изображенного на рис. 4.1, включены однокристалльная микроЭВМ серии K1816BE48; постоянное запоминающее устройство в виде трех БИС ПЗУ₀ – ПЗУ₂ со структурой 1024×8 бит каждое; оперативное запоминающее устройство ОЗУ со структурой 128×8 бит; канал ввода данных, реализованный на многорежимном буферном регистре МБР₃ серии K589ИР12; канал вывода информации на МБР₄, АЦП и ЦАП.

Каналы ввода-вывода адресуются как ячейки памяти внешнего ОЗУ. Системная шина адреса $A_0 - A_{11}$ организована двумя многорежимными регистрами МБР₁ и МБР₂, в которые в первом такте машинного цикла по сигналу ЕА с линий портов $P0_{0-7}$ и $P2_{0-3}$ защелкивается 12-разрядный адрес A_{0-11} . Адресация ячеек памяти ПЗУ осуществляется сигналами с линий $A_0 - A_9$ ША, а адресация 128 ячеек памяти ОЗУ – сигналами с линий $A_0 - A_6$.

Линии $A_{10} - A_{11}$ ША подведены к дешифратору адреса ДША₁, который имеет три выхода.

Сигналом с первого выхода ДША₁ по входу \overline{CS} адресуется БИС ПЗУ₀. Наличие еще двух выходов у дешифратора ДША₁ позволяет дополнительно подключить две БИС ПЗУ₁ и ПЗУ₂ со структурой 1024×8 бит.

Синхронизация чтения данных из ячеек памяти ПЗУ осуществляется сигналом \overline{EROM} микроЭВМ по входу \overline{RD} ПЗУ.

Поскольку каналы ввода-вывода информации организованы как ячейки памяти внешнего ОЗУ, то адресация БИС ОЗУ осуществляется сигналом низкого уровня на линии A_7 по входу выборки кристалла \overline{CS} БИС ОЗУ.

Для адресации каналов ввода-вывода использован дешифратор адреса ДША₂, на который подводятся линии шины адреса A_0, A_7 . При этом дешифратор имеет два выхода, что позволяет адресовать канал ввода и канал вывода данных. В рассматриваемом устройстве канал ввода имеет адрес $A_7 = 1, A_0 = 0$ и адресуется сигналом с нулевого выхода ДША₂ по входу CS_2 регистра МБР₃.

Канал ввода информации имеет адрес $A_7 = 1, A_0 = 1$ и адресуется с первого выхода дешифратора ДША₂ по входу CS_2 регистра МБР₄.

В рассматриваемом микропроцессорном вычислителе ввод-вывод данных осуществляется в режиме прерывания по запросу внешнего устройства.

Ввод-вывод информации. Для ввода данных в режиме прерывания внешнее устройство подает положительный импульс (ЗАПР “ввод”) на вход S R-S триггера T. На выходе триггера устанавливается сигнал низкого логического уровня, который поступает на вход Г/П АЦП и иницирует процесс преобразования аналогового сигнала $U_{\text{вх}}$ в цифровой код. По истечении процесса преобразования ($t_{\text{прб}} = 25$ мкс) на выход ГД АЦП устанавливается высокий уровень напряжения, которым по входу STB МБР₃ цифровой код защелкивается в регистр.

Одновременно на выходе $\overline{\text{INT}}$ МБР₃ формируется сигнал низкого уровня, а сигналом с выхода ГД АЦП по входу R триггер T возвращается в исходное состояние, что обеспечивает готовность ЦАП к следующему циклу преобразования.

Сигнал низкого уровня с выхода $\overline{\text{INT}}$ МБР₃ одновременно подается на вход запроса внешнего прерывания $\overline{\text{INT}}$ микроЭВМ и линию P1₁ порта P1. Если прерывания программно разрешены, то микроЭВМ обращается к программе обработки запроса прерывания, в которой последовательно опрашиваются линии порта P1. При этом выясняется, кто запрашивает режим прерывания – либо устройство ввода, либо устройство вывода данных.

Далее следует команда ввода информации через МБР₃. Выполняя данную команду, микроЭВМ выставляет на системную шину адреса адрес устройства ввода ($A_7 = 1, A_0 = 0$), который декодирует в дешифраторе адреса ДША₂ и в виде логической единицы с нулевого выхода ДША₂ поступает на выход выборки кристалла CS₂ МБР₃. Далее микроЭВМ формирует импульс синхронизации чтения информации из внешнего устройства ОЗУ $\overline{\text{RD}}$, который подается на вход выборки кристалла CS₁ МБР₃. В результате МБР₃ срабатывает и вводимая информация из МБР₃ поступает на вход порта PO и считывается в аккумулятор микроЭВМ. На этом цикл ввода данных завершается.

При выводе информации в режиме прерывания используется регистр МБР₄, выступающий по отношению к микроЭВМ как ячейка памяти внешнего ОЗУ со своим адресом ($A_7 = 1, A_0 = 1$). При этом процесс вывода данных из аккумулятора микроЭВМ в регистр МБР₄ и далее ЦАП по логике полностью совпадает с вводом информации.

5. УСТРОЙСТВА ЦИФРОВОЙ ОБРАБОТКИ ИНФОРМАЦИИ С ЖЕСТКОЙ ЛОГИКОЙ

Наиболее часто встречаемыми операциями при реализации специализированных вычислителей являются операции хранения, арифметического сложения (вычитания) и умножения многоразрядных чисел. Промышленностью выпускается широкий ряд БИС, выполняющих указанные операции.

5.1. Устройства хранения многоразрядных чисел

В качестве устройства хранения (запоминания) многоразрядных двоичных чисел могут быть использованы регистры на базе D-триггеров. На рис. 5.1 приведена схема четырехразрядного регистра серии К531ИР12 [7].

В состав регистра входят четыре синхронных D-триггера с параллельным вводом и выводом данных.

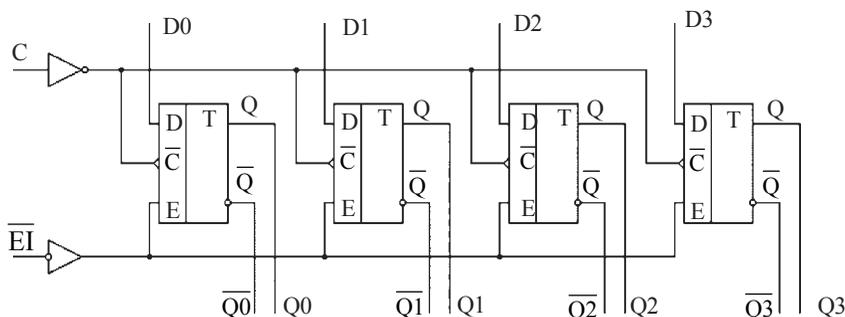


Рис. 5.1

Если на входе EI присутствует напряжение низкого уровня, данные будут загружены в регистр по положительному перепаду на тактовом входе C. Ток потребления более 75 мА, время задержки распространения сигналов от 8 до 10 нс.

На рис. 5.2 изображена схема двухступенчатого регистра на базе D-триггеров. По положительному фронту сигнала на входе C данные с

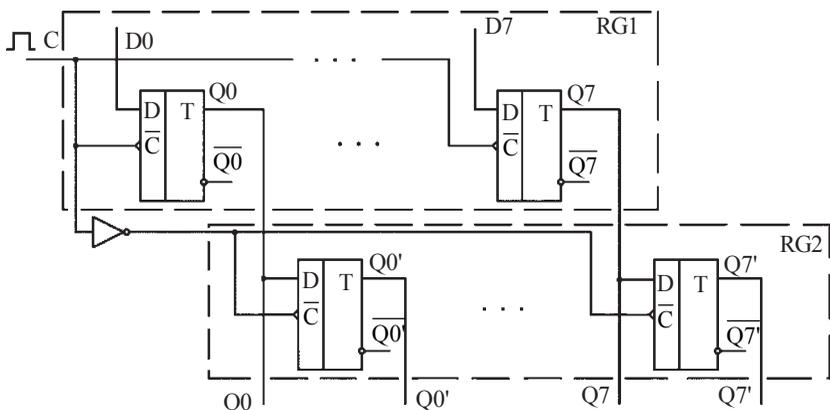


Рис. 5.2

регистра RG1 переписываются в регистр RG2. По заднему фронту данные с линий D0–D7 записываются в регистр RG1.

5.2. Умножитель (16×16) KP1802BP5

Микросхема предназначена для умножения 16-разрядных двоичных модулей или чисел в дополнительном коде и обладает организацией, позволяющей наращивать разрядность обрабатываемых чисел за счет объединения нескольких БИС при использовании дополнительных сумматоров и регистров [3]. Структура БИС приведена на рис. 5.3.

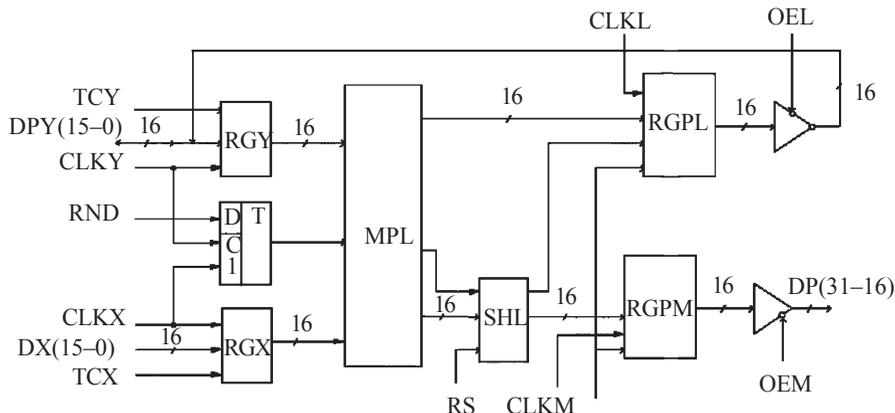


Рис. 5.3

В состав перемножителя входят: 16-разрядный регистр множителя RGY; 16-разрядный регистр множимого RGX; триггер округления T; 16-разрядные регистры старшей и младшей части произведения RGPM, RGPL; 16-разрядный параллельный сдвигатель SHL; два 16-разрядных буфера выдачи произведения; 16-разрядный блок умножения MPL. Описание выводов БИС приведено в табл.5.1.

Таблица 5.1

Обозначение вывода	Назначение вывода
DX (15-0)	Входная параллельная шина множимого X
DPY(15-0)	Двунаправленная параллельная шина множителя Y или младших разрядов произведения P
DP (31-16)	Выходная параллельная шина старших разрядов произведения P
CLKY, CLKX	Входы стробов записи в регистре множителя и множимого
RND	Вход округления
TCY, TCX	Вход задания формы представления чисел Y и X
CLKL, CLKM	Входы стробов записи младших и старших разрядов результата в регистр произведения
OEL, OEM	Входы стробов выдачи младших и старших разрядов произведения
RS	Вход управления сдвигом вправо старшей части произведения
STB	Вход управления передачей чисел через регистры произведения

Микросхема имеет совмещенную двунаправленную шину DPY (множителя/младших разрядов результата произведения).

Микросхема интерпретирует множимое (множитель) как модуль при подачи на вход TXC (TXY) потенциала логической единицы или как число в дополнительном коде при потенциале логического нуля. Но вход RND перед выполнением умножения необходимо подать сигнал, задающий округление произведения до 8 бит при $RND = 1$.

Операция округления производится суммированием единицы к старшему разряду отбрасываемой части и не вносит дополнительной задержки. При этом по сигналу на входе RS осуществляется сдвиг вправо старшей части произведения. Результат операций по сигналам CLKL и CLKM записывается в регистр произведений RGPL и RGPM, построенных на D-триггерах. Высокий потенциал на входе STB позволяет исключить использование регистров и индцировать произведение непосредственно на шинах DPY (15-0) и DP (31-16) по отрицательному

потенциалу стробов OEL и OEM. Запись данных в регистры произведения происходит по положительному стробу на входах CLKL и CLKM. При нулевом потенциале на входах CLKL и CLKM осуществляется режим хранения информации.

Управление работой БИС осуществляется 7-разрядной микрокомандой и тактируется четырьмя синхросериями.

5.3. Параллельный 4-входовый сумматор КР1802ИМ1

Микросхема предназначена для быстрого выполнения арифметического сложения или вычитания четырех 4-разрядных чисел, представленных в дополнительном коде, и обладает секционной организацией, что позволяет наращивать разрядность обрабатываемых чисел по четыре бита за счет объединения нескольких БИС. Структура сумматора показана на рис. 5.4.

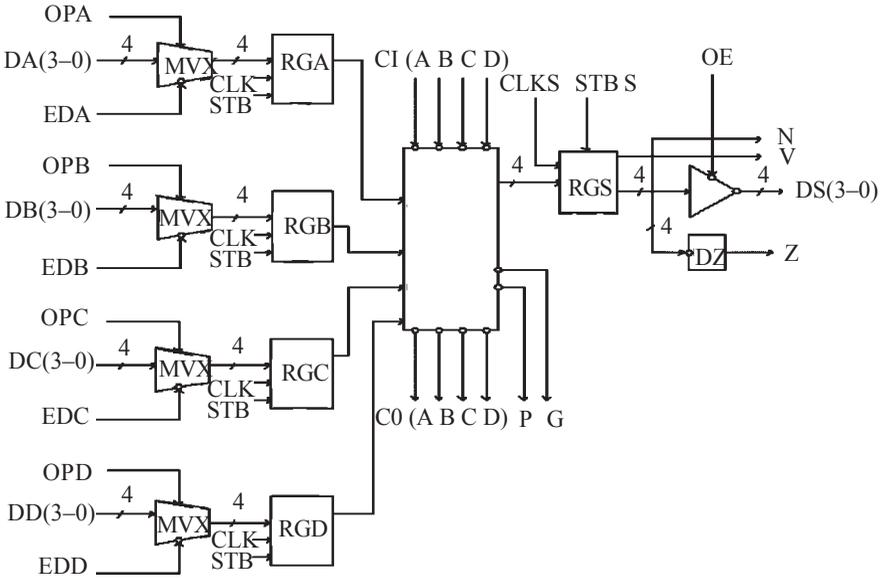


Рис. 5.4

В состав БИС входят: четыре мультиплексора входных данных (MUX); четыре регистра операндов (RGA, RGB, RGC, RGD); 4-входовый 4-разрядный сумматор (SM); 5-разрядный регистр результата (RGS); декодер нуля (DZ); выходной 4-разрядный буфер данных. Описание выводов БИС приведено в табл. 5.2.

Таблица 5.2

Обозначение вывода	Назначение вывода
DA (3-0)	4-разрядная входная шина А
DB (3-0)	4-разрядная входная шина В
DC (3-0)	4-разрядная входная шина С
DD (3-0)	4-разрядная входная шина D
OPA, OPB, OPC, OPD	Входы задания инверсного значения сигналов в мультиплексорах каналов А, В, С, D
EDA, EDB, EDC, EDD	Входы разрешения приема информации с шин DA, DB, DC и DD
OE	Вход разрешения выдачи данных из RGS на магистраль DS
CLK	Вход записи операндов в RGA-RGD
STB	Вход управления записью операндов в RGA-RGD
CLKS	Вход записи результата в RGS
STBS	Вход управления записью результата в RGS
CI(A, B, C, D)	Входы переноса сумматора по каналам А,В,С,D
DS(3-0)	4-разрядная выходная шина результата
CO(A, B, C, D)	Выходы переноса сумматора по каналам А,В,С,D
P, G	Выходы сигналов подготовки и распространения ускоренного переноса
N, V, Z	Выходы знака, признака переполнения и признака нуля результата

Кодировка данных на входных шинах и выходе результата инверсная; на входах (CIA-CID) при суммировании по данному каналу необходимо задавать низкий уровень, а при вычитании – высокий. Операция суммирования или вычитания информации по каждому из каналов задается полярностью данных, пропускаемых через входные мультиплексоры, и в случае их инвертирования для вычитания – прибавлением в младший разряд единицы со входа канального переноса. При этом если на входах (CIA-CID) установлен низкий потенциал, то к сумме прибавляется число 0000, а при высоком потенциале прибавляется число 1111. Микросхема тактируется одной синхросерией CLKs.

6. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

В настоящее время цифровые методы находят самое широкое применение при решении задач рекурсивной и нерекурсивной фильтрации, демпфирования в автоматических следящих системах, вычисления корреляционной функции, спектрального анализа, быстрого преобразования Фурье и т. д.

Использование средств цифровой техники позволяет сравнительно легко получить требуемые частотные и временные характеристики устройств обработки.

6.1. Цифровая фильтрация

При заданной передаточной функции непрерывного фильтра $W(p)$ передаточная функция цифрового фильтра находится заменой $p = \frac{2}{T} \frac{z-1}{z+1}$, где T – период дискретизации или период поступления отсчетов входного сигнала; $z = e^{pT}$ [2].

В общем виде передаточная функция цифрового фильтра в z -плоскости определяется соотношением

$$W(p) = \frac{\sum_{i=0}^n a_i z^{-i}}{1 + \sum_{i=1}^n b_i z^{-i}}, \quad (6.1)$$

где a_i, b_i – весовые коэффициенты нерекурсивной и рекурсивной частей передаточной функции.

Так, например, для фильтра нижних частот (апериодического звена первого порядка) с непрерывной передаточной функцией

$$W_{\text{ФНЧ}}(p) = \frac{K}{1 + pT_{\text{ФНЧ}}}, \quad (6.2)$$

его дискретная передаточная функция имеет вид

$$W_{\text{ФНЧ}}(p) = K_1 \frac{1 + z^{-1}}{1 + b_1 z^{-1}}, \quad (6.3)$$

где $K_1 = K / (1 + \tau)$; $b_1 = -(\tau - 1) / (\tau + 1)$; $\tau = 2T_{\text{ФНЧ}} / T$.

Выражение (6.1) позволяет записать разностное уравнение, определяющее алгоритм работы цифрового фильтра:

$$y[k] = \sum_{i=0}^n a_i x[k-i] - \sum_{i=1}^n b_i y[k-i], \quad (6.4)$$

где $y[k]$, $x[k]$ – отсчеты выходного и входного сигналов фильтра; n – порядок фильтра.

Структурная схема цифрового рекурсивного фильтра изображена на рис. 6.1 [5].

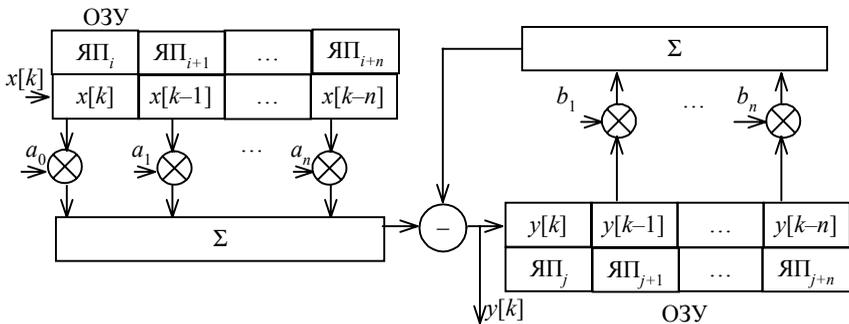


Рис. 6.1

Из представленной схемы цифрового рекурсивного фильтра видно, что его реализация требует организации хранения $(n + 1)$ текущих отсчетов входного сигнала $x[k]$, ..., $x[k - n]$ и $(n + 1)$ отсчетов выходного сигнала фильтра $y[k]$, ..., $y[k - n]$. Данная операция легко выполняется на основе ячеек оперативной памяти ОЗУ (в простейшем случае с использованием $(n + 1)$ -разрядного сдвигающего регистра). С поступлением каждого нового отсчета входного сигнала необходимо выполнить во всех указанных ячейках памяти сдвиг информации, соответствующий операции замещения содержимого ячейки памяти ЯП $_{\alpha}$ содержимым ячейки ЯП $_{\alpha-1}$ с адресом на единицу меньшим адреса ячейки ЯП $_{\alpha}$.

В табл. 6.1 приведены типовые передаточные функции и разностные уравнения фильтров, используемых в цифровых следящих измерителях.

В качестве примера рассмотрим цифровой фильтр с третьей передаточной функцией. При этом положим $K = 2 \text{ с}^{-2}$, $\tau = 1 \text{ с}$, $T = 10 \text{ с}^{-1}$. Тогда соответствующее разностное уравнение имеет вид

$$y[k] = 0,105x[k] + 0,01x[k-1] - 0,095x[k-2] + 2y[k-1] - y[k-2].$$

Дробные весовые коэффициенты затрудняют реализацию цифрового фильтра. Введя масштабный множитель, например 10^3 , удобно коэффициенты разностного уравнения сделать целыми числами.

Тогда

$$y[k] = 10^{-3} \{105x[k] + 10x[k-1] - 95x[k-2] + 2 \cdot 10^3 y[k-1] - 10^3 y[k-2]\}.$$

Масштабирование легко учитывается при выводе результатов расчетов из вычислителя к периферийным устройствам.

В случае использования нерекурсивного фильтра его разностное уравнение имеет вид

$$y[k] = \sum_{i=0}^n a_i x[k-i].$$

Нерекурсивные фильтры значительно проще в реализации, однако обладают значительно худшими амплитудно-частотными характеристиками (АЧХ) по сравнению с рекурсивными фильтрами.

При цифровой фильтрации аналоговых сигналов необходимо предварительно с помощью АЦП отсчеты входного сигнала преобразовать в цифровую форму, а при выводе результата выходные цифровые отсчеты подаются в ЦАП.

При проектировании цифровых фильтров необходимо решить следующие задачи:

- по непрерывному аналогу найти дискретную передаточную функцию;
- определить период дискретности T представления входной и выходной информации;
- выбрать параметры АЦП и ЦАП;
- определить методические и случайные ошибки, связанные с переходом от непрерывной к дискретной передаточной функции;
- выбрать аппаратные средства.

Таблица 6.1

№ п/п	Непрерыв- ный аналог $W(p)$	Дискретная передаточная функция $W(z)$	Разностное уравнение $Y[k] = \sum_{i=0}^n a_i x[k-i] - \sum_{i=1}^n b_i y[k-i]$
1	$\frac{K}{p}$	$\frac{KT}{2} \frac{1+z^{-1}}{1-z^{-1}}$	$y[k] = \frac{KT}{2} x[k] + \frac{KT}{2} x[k-1] + y[k-1]$
2	$\frac{K(1+p\tau_1)}{p(1+p\tau_2)}$	$\frac{KT \left(1 + \frac{2\tau_1}{T}\right) + 2z^{-1} + \left(1 - \frac{2\tau_1}{T}\right) z^{-2}}{2 \left(1 + \frac{2\tau_2}{T}\right) \left[1 - \left(\frac{4\tau_2}{T+2\tau_2}\right) z^{-1} - \left(\frac{T-2\tau_2}{T+2\tau_2}\right) z^{-2}\right]} \rightarrow$ $\rightarrow \frac{\left(1 - \frac{2\tau_1}{T}\right) z^{-2}}{-\left(\frac{T-2\tau_2}{T+2\tau_2}\right) z^{-2}}$	$y[k] = \frac{KT}{2} \left(\frac{T+2\tau_1}{T+2\tau_2}\right) x[k] + \frac{KT^2}{(T+2\tau_2)} x[k-1] +$ $+ \frac{KT}{2} \left(\frac{T-2\tau_1}{T+2\tau_2}\right) x[k-2]$
3	$\frac{K(1+p\tau)}{p^2}$	$\frac{KT^2 \left(1 + \frac{2\tau}{T}\right) + 2z^{-1} + \left(1 - \frac{2\tau}{T}\right) z^{-2}}{2(1-2z^{-1}+z^{-2})}$	$y[k] = \frac{KT}{4} (T+2\tau)x[k] + \frac{KT^2}{2} x[k-1] +$ $+ \frac{KT}{4} (T-2\tau)x[k-2] +$ $+ 2y[k-1] - y[k-2]$

Выбор периода дискретности. Предположим, что по непрерывной передаточной функции $W(p)$ найдено ее дискретное прерывание $W(z)$. С учетом дискретизации передаточная функция цифрового фильтра является периодической с периодом $\omega_{\pi} = \frac{2\pi}{T}$. На рис. 6.2, а и б показаны модули передаточных функций непрерывного и цифрового фильтров соответственно.

При этом один из наиболее простых переходов к определению периода дискретности состоит в том, что ставится требование хорошего совпадения амплитудно-частотных и фазочастотных характеристик (ФЧХ) непрерывного и цифрового фильтров в области низких частот $0 \leq \omega \leq \omega_{гр}$. Граничная частота $\omega_{гр}$ определяет ту точку на АЧХ или ФЧХ, где ошибка совпадения характеристик фильтров не превосходит заданную.

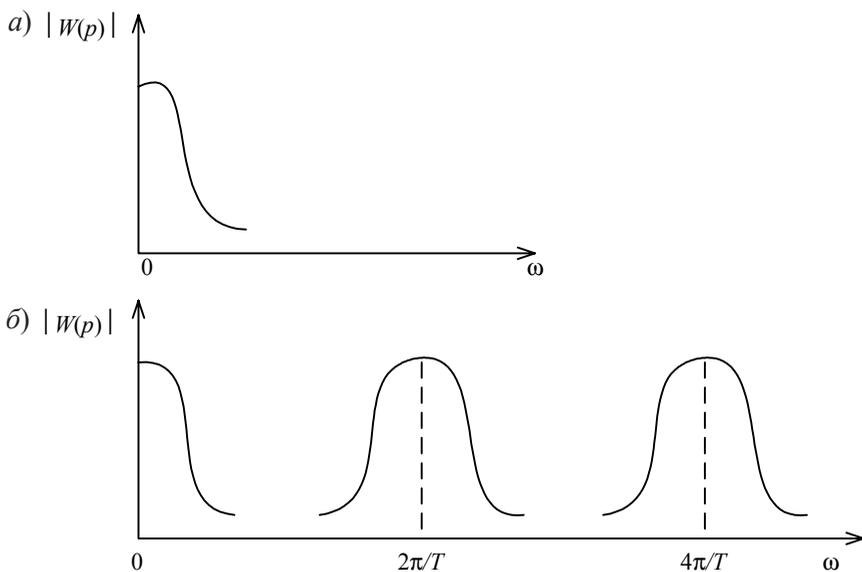


Рис. 6.2

При выборе периода дискретности T приходится находить компромиссное решение с учетом следующих противоречивых требований.

Во-первых, чрезмерное уменьшение величины T при определенном быстродействии вычислителя ограничивает допустимую сложность алгоритма вычислений, которые производятся в реальном масштабе вре-

мени и на каждом такте должны быть выполнены за время, не превышающее значения T .

Во-вторых, увеличение значения T приводит к возрастанию информационных потерь при квантовании непрерывного сигнала, появляются нежелательные динамические искажения полезного сигнала.

Если нахождение дискретной передаточной функции цифрового фильтра проведено по непрерывному прототипу с использованием замены $p = \frac{2z-1}{Tz+1}$, то абсолютная погрешность реализации его желаемой амплитудно-частотной характеристики $A(\omega) = |W(j\omega)|$ на граничной частоте $\omega_{гр}$ составит

$$\Delta A(\omega_{гр}) = A(\omega_{гр}) - A^*(z_{гр}),$$

где $A^*(z_{гр})$ – АЧХ цифрового фильтра на граничной частоте.

Воспользовавшись соотношением (6.1), можно записать

$$A^*(z_{гр}) = \frac{\left| \frac{\sum_{i=0}^n a_i z_{гр}^{-i}}{1 + \sum_{i=1}^n b_i z_{гр}^{-i}} \right|}{\left| \frac{\sum_{i=0}^n a_i e^{-i(j\omega_{гр}T)}}{1 + \sum_{i=1}^n b_i e^{-i(j\omega_{гр}T)}} \right|}$$

или

$$A^*(\omega_{гр}) = \left\{ \frac{\left[\sum_{i=0}^n a_i \cos(i\omega_{гр}T) \right]^2 + \left[\sum_{i=1}^n a_i \sin(i\omega_{гр}T) \right]^2}{\left[1 + \sum_{i=1}^n b_i \cos(i\omega_{гр}T) \right]^2 + \left[\sum_{i=1}^n b_i \sin(i\omega_{гр}T) \right]^2} \right\}^{\frac{1}{2}}; \quad (6.5)$$

зададимся относительной погрешностью реализации АЧХ

$$\varepsilon \leq \frac{|\Delta A(\omega_{гр})|}{A(\omega_{гр})}. \quad (6.6)$$

Данная формула позволяет обоснованно выдвинуть требование к периоду дискретности, исходя из заданной АЧХ непрерывного прототи-

па цифрового фильтра $A(\omega)$, величины граничной частоты $\omega_{\text{гр}}$ и относительной погрешности реализации АЧХ ε .

В качестве примера рассмотрим реализацию цифрового фильтра нижних частот по его непрерывному прототипу (6.2). При этом положении $K = 1$, $T_{\text{ФНЧ}} = 0,1$ с, $\varepsilon = 0,01$, $\omega_{\text{гр}} = 200$ с⁻¹.

В развернутом виде соотношение (6.6) записываются следующим образом:

$$\left| 1 - \frac{2\sqrt{1 + \omega_{\text{гр}}^2 T_{\text{ФНЧ}}^2} \cos \frac{\omega_{\text{гр}} T}{2}}{\left(1 + \frac{2T_{\text{ФНЧ}}}{T}\right) \sqrt{1 - 2\frac{2T_{\text{ФНЧ}} - T}{2T_{\text{ФНЧ}} + T} \cos \omega_{\text{гр}} T + \left(\frac{2T_{\text{ФНЧ}} - T}{2T_{\text{ФНЧ}} + T}\right)^2}} \right| \leq 0,01.$$

Численным методом из данного соотношения находим, что необходимо обеспечить $T \leq 0,00175$ с.

Можно воспользоваться более простым методом для определения периода дискретизации. Необходимо потребовать, чтобы модуль передаточной функции непрерывного прототипа цифрового фильтра на граничной частоте не превышал бы некоторой малой величины, например 0,05, тогда $T \leq (0,1 - 0,3) \omega_{\text{гр}}^{-1}$.

Для ранее принятых исходных данных получаем $T \leq (0,1 - 0,3)0,005$ с.

Выбор параметров АЦП и ЦАП. Основными параметрами АЦП и ЦАП, подлежащими определению, являются: цена единицы младшего разряда преобразователей δ_1 и δ_2 , а также разрядности преобразователей α_1 и α_2 соответственно.

При выборе цены единицы младшего разряда АЦП δ_1 можно поставить условие, чтобы средний квадрат дополнительной ошибки от квантования по уровню в АЦП не превышал заданного значения:

$$\sigma_{\text{КВ}}^2 = \frac{\delta_1^2}{12} T \Delta f_3,$$

здесь Δf_3 – эквивалентная полоса пропускания непрерывного фильтра для белого шума.

Если известна импульсная характеристика непрерывного фильтра $h(t)$, то ее решаемая функция $h[k]$ определяет алгоритм работы цифрового фильтра

$$y[k] = \sum_{i=0}^m x[k-i]h[i].$$

В этом случае дисперсия шумов квантования на выходе фильтра находится из соотношения

$$\sigma_{\text{кв}}^2 = \frac{\delta_1^2}{12} \sum_{i=0}^m h^2 [i].$$

В данном случае m – число учитываемых отсчетов импульсной характеристики фильтра.

После выбора цены единицы младшего разряда АЦП требуемое число его двоичных разрядов определяется формулой

$$\alpha_1 \geq \log_2 \left[1 + \frac{x_{\text{max}}^{\text{вх}}}{\delta_1} \right],$$

где $x_{\text{max}}^{\text{вх}}$ – максимальное значение входной величины АЦП, которое может быть задано.

Цену единицы младшего разряда ЦАП обычно принимают равной цене единицы младшего разряда АЦП. При выборе характеристик ЦАП следует учесть, что при определенных условиях эффект квантования по уровню проявляться не будет. Если цена единицы младшего разряда ЦАП δ_2 выбрана по формуле $\delta_2 = \delta_1/\beta$, где β – целое число (1, 2, 3, ...), то нет округления до ближайшей ступени ЦАП, так как все возможные значения выходного сигнала будут кратны δ_2 .

При $\beta = 1/2; 1/3; 1/4; \dots$ появляются шумы квантования в ЦАП, которые можно считать белыми с дисперсией

$$\sigma_{\text{к2}}^2 = \frac{\delta_2^2}{12\sigma_{\text{к1}}^2},$$

где $\sigma_{\text{к1}}^2 = \frac{\delta_1^2}{12}$ – дисперсия шумов квантования в АЦП.

Для нахождения требуемого числа разрядов ЦАП можно воспользоваться выражением

$$\alpha_2 \geq \log_2 \left[1 + \frac{y_{\text{max}}^{\text{вых}}}{\delta_2} \right],$$

где $y_{\text{max}}^{\text{вых}}$ – максимальная величина сигнала на выходе ЦАП, которая заранее должна быть оценена для исключения искажений выходного сигнала фильтра.

6.2. Дифференцирование цифровых последовательностей

Общие сведения. Рассмотрим задачу дифференцирования цифровой последовательности $g[n]$, являющейся результатом дискретизации по времени с периодом T и квантования по уровню непрерывной функции времени $g(t)$ [2].

Известно представление оператора z : $z = e^{pT}$.

Отсюда находим, что оператор дифференцирования p определяется соотношением

$$p = \ln z / T.$$

Представление оператора p -степенными рядами дает несколько алгоритмов цифрового дифференцирования, из которых наиболее приемлемым с точки зрения практической реализации является следующий:

$$p = \frac{1}{T} \left[(1 - z^{-1}) + \frac{1}{2}(1 - z^{-1})^2 + \frac{1}{3}(1 - z^{-1})^3 + \dots \right]. \quad (6.7)$$

В практических задачах в выражении (6.7) можно использовать только ограниченное число членов ряда. При этом оператор $(1 - z^{-1})^k$ соответствует взятию обратных разностей $\nabla^k g[n]$.

В результате для первой производной при использовании ограниченного числа членов ряда (6.7) имеем

$$\begin{aligned} \dot{g}_m[n] &= \left(\nabla g[n] + \frac{1}{2} \nabla^2 g[n] + \frac{1}{3} \nabla^3 g[n] + \dots + \frac{1}{m} \nabla^m g[n] \right) T^{-1} = \\ &= \frac{1}{T} \sum_{i=0}^m \frac{1}{i} \nabla^i g[n]. \end{aligned} \quad (6.8)$$

Для получения второй производной необходимо использовать оператор p^2 . Тогда из выражения (6.7) можно получить

$$\ddot{g}_m[n] = \left(\nabla^2 g[n] + \nabla^3 g[n] + \frac{11}{12} \nabla^4 g[n] + \dots + \frac{2}{m} \sum_{i=1}^{m-1} \frac{1}{i} \nabla^m g[n] \right) T^{-2}. \quad (6.9)$$

В формулах (6.8), (6.9) обратные разности имеют вид

$$\begin{aligned} \nabla g[n] &= g[n] - g[n-1] \\ \nabla^2 g[n] &= g[n] - 2g[n-1] + g[n-2] \\ \nabla^3 g[n] &= g[n] - 3g[n-1] + 3g[n-2] - g[n-3]. \end{aligned} \quad (6.10)$$

При учете формулы (6.10) формулы для первой и второй производной записываются следующим образом:

$$\dot{g}_m[n] = T^{-1} \sum_{i=0}^m a_i g[n-i], \quad (6.11)$$

где $a_i = (-1)^i \sum_{k=1}^m \frac{1}{k} c_k^i$ – биномиальные коэффициенты;

$$\ddot{g}_m[n] = T^{-2} \sum_{i=0}^m b_i g[n-i], \quad (6.12)$$

где $b_i = (-1)^i \sum_{k=2}^m \frac{2}{k} c_k^i \sum_{j=1}^{k-1} \frac{1}{j}$.

Методические ошибки дифференцирования. Среднеквадратическая ошибка определения производной стационарного сигнала $g(t)$ в дискретные моменты времени $t = nT$ может быть найдена через математическое ожидание квадрата разности между действительными значениями производной $\dot{g}[n]$ и машинным значением $\dot{g}_m[n]$, вычисленным по формуле (6.11):

$$\begin{aligned} \sigma_m^2 = M \left\{ \left(\dot{g}[n] - T^{-1} \sum_{i=0}^m a_i g[n-1] \right)^2 \right\} = & -\ddot{R}[0] - \frac{2}{T} \sum_{i=0}^m a_i \dot{R}[iT] + \\ & + \frac{2}{T^2} \sum_{i=0}^{m-j} \sum_{j-1}^m a_i a_{i+j} R[jT] + \frac{1}{T^2} \sum_{i=0}^m a_i^2 R[0], \end{aligned} \quad (6.13)$$

где $R(\tau)$ – корреляционная функция сигнала $g(t)$; $\ddot{R}(\tau)$ – корреляционная функция производной сигнала $\dot{g}(t)$; $\dot{R}(\tau)$ – взаимная корреляционная функция сигнала и его производной.

Относительная среднеквадратическая ошибка

$$\nabla_m = \frac{\sigma_m}{\dot{\sigma}_{g1}}, \quad (6.14)$$

где $\dot{\sigma}_g$ – среднеквадратическое значение производной сигнала $\dot{g}(t)$.

Выражения (6.13), (6.14) позволяют выбирать период дискретизации T по заданному значению методической ошибки при известном значе-

нии числа учитываемых обратных разностей или определять необходимое значение m при заданном значении периода дискретности.

Пусть рассматривается сигнал гармонического вида $g(t) = A \sin(\beta t + \psi)$ с амплитудой A , частотой β и с равномерно распределенной на интервале $(0; 2\pi)$ случайной фазой ψ . Для этого сигнала имеем

$$\sigma_g^2 = 0,5A^2; \quad \sigma_{\dot{g}}^2 = 0,5\beta^2 A^2;$$

$$R(\tau) = 0,5A^2 \cos \beta\tau; \quad \dot{R}(\tau) = -0,5\beta A^2 \sin \beta\tau; \quad \ddot{R}(\tau) = -0,5\beta^2 A^2 \cos \beta\tau.$$

Нетрудно показать, что при дифференцировании данного сигнала методические погрешности для $\beta T \ll 1$ можно определить приближенными формулами:

$$\sigma_M^2 \approx \frac{A^2 \beta^{2(m+1)} T^{2m}}{2(m+1)^2};$$

$$\Delta_M = \frac{\sigma_M}{\sigma_{\dot{g}}} = \sqrt{0,5} \frac{\beta^m T}{m+1}.$$

Таким образом, если заданы величины Δm и m , то период дискретности должен удовлетворять условию:

$$T \leq \frac{1}{\beta} \sqrt[m]{(m+1)\Delta_M}.$$

В качестве примера положим $\beta = 1$ рад/с, $m = 2$, $\Delta_M = 0,001 = 0,1\%$. Тогда получим, что $T \leq 0,055$ с. В случае взятия второй производной от стационарного входного сигнала $g(t)$ имеем

$$\begin{aligned} \sigma_M^2 &= M \left\{ (\ddot{g}[n] - \ddot{g}_M[n])^2 \right\} = \\ &= R_2(0) + 2T^{-2} \sum_{i=0}^m b_i R_i[iT] + 2T^{-4} \sum_{i=0}^{m-1} \sum_{j=1}^m b_i b_{i+j} R[jT] + T^{-4} \sum_{i=0}^m b_i^2 R[0]; \end{aligned}$$

$$\Delta_M = \frac{\sigma_M}{\sigma_{\dot{g}}} = \frac{\sigma_M}{\sqrt{R_2(0)}},$$

где $R_2(\tau)$ – корреляционная функция второй производной сигнала $\ddot{g}(t)$; $R_1(\tau)$ – взаимная корреляционная функция входного сигнала и его второй производной.

Приближенная формула относительной погрешности двойного дифференцирования гармонического сигнала $g(t) = A \sin(\beta t + \psi)$ со случайной начальной фазой ψ при $\beta T \ll 1$ записываются следующим образом:

$$\Delta_M = \frac{\sigma_M}{\sigma_{\dot{g}}} \approx \frac{2}{m+1} \sum_{i=1}^m i^{-1} \beta^{m-1} T^{m-1} = \beta^{m-1} T^{m-1} q_1(m). \quad (6.15)$$

В формуле (6.15) введена функция $q_1(m)$, значения которой даны ниже:

m	2	3	4	5	6	7	8
$q_1(m)$	1,0	0,917	0,834	0,761	0,700	0,678	0,630

Формула (6.15) позволяет выбрать период дискретности по заданному значению методической ошибки при известном числе учитываемых обратных разностей. Если задана величина Δm , то необходимо выполнить условие

$$T \leq \frac{1}{\beta} \left[\frac{\Delta_M}{q_1(m)} \right]^{\frac{1}{m-1}}.$$

Требуемые значения периода дискретности при двойном дифференцировании гармонического сигнала с частотой $\beta = 1$ рад/с и $\Delta m = 0,001 = 0,1\%$ приведены ниже:

m	2	3	4	5
$T, \text{с}$	0,001	0,039	0,106	0,190

Влияние шумов квантования. В случае вычисления первой производной в соответствии с выражением (6.11) суммарная дисперсия ошибки квантования

$$\sigma_{\text{кв}}^2 = \frac{\delta_1^2}{12T^2} F_0(m),$$

где δ_1 – цена единицы младшего разряда АЦП. Значение функции $F_0(m)$ и среднеквадратической шумовой ошибки $\sigma_{\text{кв}}$, отнесенной к величине δ_1/T , при различном числе учитываемых обратных разностей приведены ниже:

m	1	2	3	4	5	6	7
$F_0(m)$	2	6,5	14,7	31	68	160	396
$\sigma_{\text{кв}}^2 T / \delta_1$	0,407	0,738	1,11	1,61	2,38	3,66	5,75

Общее число разрядов АЦП выбирается из условия $\alpha_1 \geq \log_2 \left[1 + \frac{g_{\max}^{\text{ВХ}}}{\delta_1} \right]$,

где $g_{\max}^{\text{ВХ}}$ – максимальное значение входной величины АЦП, которое можно найти из соотношения $g_{\max}^{\text{ВХ}} \approx 3\sigma_g$.

При этом цена единицы младшего разряда выходного преобразователя (ЦАП) должна быть равной $\delta_2 = \delta_1 T^{-1}$. Общее число разрядов выходного преобразователя выбирается из условия $\alpha_2 \geq \log_2 \left[1 + \dot{g}_{\max} \delta_1^{-1} T \right]$.

При отбрасывании α младших разрядов в ЦАП ошибка округления не будет влиять на точность при условии

$$\frac{\delta_2^2}{12} < \frac{\delta_1^2}{12T^2} F_0(m),$$

которое сводится к неравенству $\alpha < 1,65 \lg F_0(m)$.

При вычислении второй производной аналогичным образом на основании выражения (6.12) может быть получено выражение для дисперсии шумовой ошибки квантования

$$\sigma_{\text{КВ}}^2 = \frac{\delta_1^2}{12T^4} \sum_{j=0}^m b_j^2 = \frac{\delta_1^2}{12T^4} q_2(m).$$

Значения функции $q_2(m)$ и нормированной среднеквадратической шумовой ошибки $\sigma_{\text{КВ}}$ приведены ниже:

m	2	3	4	5	6	7	8
$g_2(m)$	6,0	46	197	692	$2,88 \cdot 10^3$	$7,35 \cdot 10^3$	$2,47 \cdot 10^3$
$\sigma_{\text{КВ}}^2 T^2 / \delta_1$	0,71	1,96	4,05	7,58	15,5	24,7	45,2

Общее число разрядов АЦП выбирается так же, как и в случаях получения первой производной. Для исключения ошибок округления в ЦАП вычислителя второй производной цена единицы младшего разряда ЦАП должна быть равной $\delta_2 = \delta_1 / T^2$. При отбрасывании α младших разрядов в ЦАП ошибка округления не будет влиять на точность при условии

$$\frac{\delta_2^2}{12} < \frac{\delta_1^2}{12T^4} q_2(m)$$

или $\alpha < 3,3 \lg \sqrt{q_2(m)} = 1,65 \lg q_2(m)$.

Последняя формула позволяет выбрать допустимое закругление ЦАП, что снижает число его разрядов.

6.3. Интегрирование цифровых последовательностей

Общие сведения. Рассмотрим задачу интегрирования непрерывной функции $g(t)$, представленной в виде цифровой последовательности, полученной в результате квантования и по времени, и по уровню [2].

В качестве оператора интегрирования может рассматриваться величина $1/p$, где $p = \frac{d}{dt}$. Для практической реализации целесообразно использовать три приближенные равенства вычисления оператора интегрирования:

$$\frac{1}{p} \approx \frac{T}{z-1} = \frac{Tz^{-1}}{1-z^{-1}}; \quad (6.16)$$

$$\frac{1}{p} \approx \frac{T}{2} \frac{z+1}{z-1} = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}}; \quad (6.17)$$

$$\frac{1}{p} \approx \frac{Tz}{z-1} = \frac{T}{1-z^{-1}}. \quad (6.18)$$

Алгоритмы интегрирования. Обозначим через $f(t)$ интеграл от функции времени $g(t)$ на интервале от 0 до t . Из выражений (6.17), (6.18) нетрудно получить разностные уравнения, определяющие алгоритмы интегрирования:

$$f_M[n] = Tg[n-1] + f_M[n-1]; \quad (6.19)$$

$$f_M[n] = \frac{T}{2}(g[n] + g[n-1]) + f_M[n-1]; \quad (6.20)$$

$$f_M[n] = Tg[n] + f_M[n-1], \quad (6.21)$$

где $g[n], f_M[n]$ – цифровые представления входной и выходной величины интегратора.

Формула (6.19) соответствует интегрированию методом прямоугольников с недостатком, формула (6.20) – интегрированию методом трапеций, а формула (6.21) – интегрированию методом прямоугольников с избытком.

При интегрировании постоянного значения $g(t) = \text{const}$ все три выражения (6.19)–(6.21) дают нулевую ошибку. Если входной сигнал представляет собой линейно возрастающую функцию $g(t) = C_1 t$, $f(t) = C_1 t^2/2$, то ошибка интегрирования на шаге (за время T) для формулы (6.19) равна $C_1 T^2/2$. Формула (6.21) дает в этих же условиях ошибку $C_1 T^2/2$, а формула (6.20) – нулевую ошибку.

Алгоритмы точного интегрирования. Ошибка интегрирования может быть оценена следующим соотношением:

$$F_{\delta}(z) = F_g(z) - W_{\text{и}}(z)G(z), \quad (6.22)$$

где $F_{\delta}(z)$, $F_g(z)$, $G(z)$ – изображение функций $f[n]$, $f_{\text{м}}[n]$ и $g[n]$; $W_{\text{и}}(z)$ – передаточная функция цифрового интегратора.

Из формулы (6.22) следует, что нулевая ошибка интегрирования будет при $W_{\text{и}}(z) = F_g(z)/G(z)$.

Для входного сигнала $g(t) = C_k t^k / k!$ изображение

$$G(z) = \frac{C_k T^k K_k(z)}{k!(z-1)^{k+1}},$$

где $K_k(z)$ – полином [2].

Если $k = 0$ и $g(t) = C_0$, то передаточная функция интегратора с нулевой ошибкой $W_{\text{и}}(z) = Tz^{-1}/(1-z^{-1})$ совпадает со случаем интегрирования по методу прямоугольников (6.16); если $k = 1$ и $g(t) = C_1 t$, то передаточная функция $W_{\text{и}}(z) = 0,5T(1+z^{-1})/(1+z^{-1})$ совпадает со случаем интегрирования по методу трапеций (6.17); если $k = 2$ и $g(t) = C_2 t^2/2$, то передаточная функция интегратора $W_{\text{и}}(z) = T(1+4z^{-1}+z^{-2})/[(1+z^{-2})3]$ соответствует методу Симпсона и дает нулевую ошибку при $k \leq 2$.

Последней передаточной функции соответствует разностное уравнение

$$f_{\text{м}}[n] = T(g[n] + 4g[n-1] + g[n-2])/3 + f_{\text{м}}[n-2]. \quad (6.23)$$

При $k > 2$ передаточные функции интеграторов с нулевой ошибкой дают неустойчивые алгоритмы.

Полученные выше формулы для ошибок интегрирования входного сигнала позволяют выбирать алгоритм и период дискретности из условия допустимого значения этих ошибок.

Входящий во все алгоритмы интегрирования масштабный множитель T , как и в случае дифференцирования, учитывается в выходном преобразователе, цена единицы младшего разряда которого $\delta_2 = T\delta_1$, где δ_1 – цена единицы младшего разряда входного преобразователя. Для исключения ошибок округления целесообразно все коэффициенты сделать целыми числами и дробную часть присоединить к масштабу на выходе.

Влияние шумов квантования. Эффект влияния шумов квантования можно рассматривать двояко: или как дополнительную ошибку интегрирования на каждом шаге, или как накапливающуюся ошибку, вызванную эффектом интегрирования. Для независимых ошибок квантования дисперсия ошибки интегрирования на шаге определяется формулой

$$\sigma_{\tau}^2 = \frac{T^2 \delta_1^2}{12} \sum_{i=0}^m a_i^2,$$

где a_i – коэффициенты при дискретах входного воздействия.

Так, для алгоритма (6.23) имеем

$$\sigma_{\tau}^2 = \frac{T^2 \delta_1^2}{12} \left(\frac{1}{9} + \frac{16}{9} + \frac{1}{9} \right) = \frac{T^2 \delta_1^2}{6}.$$

Накапливающаяся шумовая ошибка интегрирования может быть найдена суммированием дисперсий шумовой ошибки на каждом шаге интегрирования

$$\sigma_{\text{ш}}^2 = n\sigma_{\tau}^2 = t\sigma_{\tau}^2 / T, \quad (6.24)$$

где n – число шагов интегрирования; t – время интегрирования.

Формула (6.24) дает эквивалентный уход интегратора. В отличие от непрерывных интеграторов, у которых выходная величина увеличивается при уходе пропорционально времени, здесь пропорционально времени нарастает дисперсия шумовой ошибки. Этот вид ухода присущ всем цифровым интеграторам и является их отличительной особенностью.

6.4. Микропроцессорный следящий измеритель

На рис. 6.3 показана укрупненная схема цифрового следящего измерителя, состоящего из дискриминатора, АЦП, цифрового фильтра сглаживания и коррекции, ЦАП и синтезатора [5].

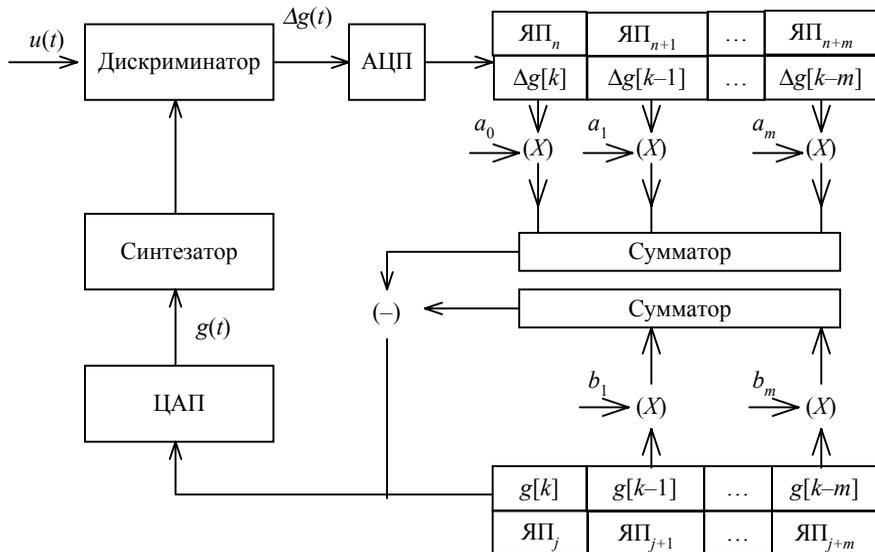


Рис. 6.3

Алгоритм работы цифрового фильтра сглаживания и коррекции определяется разностным уравнением [1]

$$g[k] = \sum_{i=0}^m a_i \Delta g[k-i] - \sum_{i=1}^m b_i g[k-i], \quad (6.25)$$

где $\Delta g[k]$ – цифровые отсчеты (с выхода АЦП) выходного сигнала дискриминатора $\Delta g(t)$; $g[k]$ – выходной сигнал цифрового фильтра; a_i , b_i – весовые коэффициенты.

В уравнении (6.25) сделаем замену: $\Delta g[k]$ на $x[k]$ и $g[k]$ на $y[k]$.

Тогда разностное уравнение принимает вид

$$y[k] = \sum_{i=0}^m a_i x[k-i] - \sum_{i=1}^m b_i y[k-i]. \quad (6.26)$$

Согласно данному уравнению в микропроцессорном устройстве необходимо иметь $(m+1)$ ячейку оперативной памяти для хранения вход-

ных отсчетов $x[k]$ и $(m+1)$ ячейку оперативной памяти для хранения выходных отсчетов $y[k]$. Пусть в $(m+1)$ ячейках памяти ЯП $_n, \dots, \text{ЯП}_{n+m}$ хранятся отсчеты $x[k], \dots, x[k-m]$, а в ячейках ЯП $_j, \dots, \text{ЯП}_{j+m}$ – отсчеты $y[k], \dots, y[k-m]$. С поступлением каждого нового отсчета выходного сигнала дискриминатора необходимо выполнить во всех ячейках памяти сдвиг информации, соответствующий операции замещения содержимого ячейки памяти ЯП $_\alpha$ содержимым ячейки ЯП $_{\alpha-1}$ с адресом, на единицу меньшим.

Для выполнения операций умножения, сложения и вычитания при реализации разностного уравнения (6.26) удобно пользоваться стандартными подпрограммами.

Укрупненно алгоритм работы микропроцессорного фильтра должен содержать следующие операции:

- сдвиг содержимого ячеек памяти ЯП $_n, \dots, \text{ЯП}_{n+m}$ и ЯП $_j, \dots, \text{ЯП}_{j+m}$ с предшествующими отсчетами величин x и y соответственно;
- ввод с АЦП текущего отсчета выходного сигнала дискриминатора $x[k]$;
- умножение отсчетов $x[k], \dots, x[k-m]$ и $y[k], \dots, y[k-m]$ на соответствующие весовые коэффициенты;
- суммирование частных произведений $x[k-i]a_i$ и $y[k-i]b_i$ согласно алгоритму работы фильтра;
- вывод текущего отсчета выходного сигнала фильтра $y[k]$ в ЦАП.

Формат выходных отсчетов дискриминатора. Выходные отсчеты дискриминатора $x[k]$ после АЦП будем представлять целыми знаковыми 8-разрядными двоичными числами (байтами): формат чисел $x[k]$ показан на рис. 6.4.

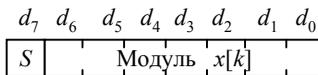


Рис. 6.4

При этом в разрядах $d_0 - d_6$ записывается модуль отсчета $x[k]$ целым двоичным числом в прямом коде, а в разряде d_7 указывается знак (флаг S) числа. При $S = 1$ отсчет $x[k]$ является отрицательным, а при $S = 0$ – положительным.

Формат выходных отсчетов цифрового фильтра. Выходной сигнал цифрового фильтра $y[k]$ будем представлять целыми знаковыми 16-разрядными двоичными числами, формат которых представлен на рис. 6.5.



Рис. 6.5

В разрядах $d_0 - d_n$ задается модуль отсчета $y[k]$ целым двоичным числом в прямом коде. Разряд d_{15} указывает знак числа. При $S = 1$ отсчет $y[k]$ является отрицательным, а при $S = 0$ – положительным.

Далее по тексту подобный формат целых знаковых чисел будем называть “знак-модуль”, считая старший бит числа знаковым (независимо от количества битов, отведенных на его хранение), а все прочие – хранящими модуль числа.

Формат весовых коэффициентов. Весовые коэффициенты a_i и b_i являются дробными числами. В связи с тем, что программная реализация нецелочисленных операций сложения, вычитания и умножения является трудоемкой, а получаемые при этом алгоритмы являются медленно действующими, применение дробных чисел в программе не рекомендуется.

Целесообразно все весовые коэффициенты представить в виде целых чисел, домножив их на общий множитель с последующим округлением их до целых чисел. Соответственно, после окончания расчетов по разностному уравнению, результат необходимо поделить на тот же общий множитель.

Возникает необходимость выбора величины общего множителя. Введем ограничения на величину весовых коэффициентов. На практике для весовых коэффициентов цифрового фильтра достаточно иметь две десятичные цифры до запятой и три десятичные цифры после запятой.

Так как после запятой весовые коэффициенты имеют три значащие цифры, то рекомендуется домножить коэффициенты разностного уравнения на 1024. Выбор в качестве общего множителя числа, представимого в виде 2^n , позволяет значительно упростить и ускорить выполнение операции деления на это число текущего решения разностного уравнения.

Рассмотрим пример. Запишем разностное уравнение (6.26) в конкретной форме:

$$y[k] = 3,771 \cdot x[k] - 6,842 \cdot x[k-1] + 3,091 \cdot x[k-2] + 0,332 \cdot y[k-1] + 0,668 \cdot y[k-2],$$

где $a_0 = 3,771$; $a_1 = -6,842$; $a_2 = 3,091$; $b_1 = 0,332$; $b_2 = 0,668$.

После домножения на 1024 и округления получим:

$$y[k] = 3861 \cdot x[k] - 7006 \cdot x[k-1] + 3165 \cdot x[k-2] + 340 \cdot y[k-1] + 684 \cdot y[k-2], \quad (6.27)$$

где $a_0 = 3861$; $a_1 = -7006$; $a_2 = 3165$; $b_1 = 340$; $b_2 = 684$.

Таким образом, весовые коэффициенты цифрового фильтра a_i и b_i будем представлять целыми знаковыми 16-разрядными двоичными числами, формат которых показан на рис. 6.6. Это позволяет задавать значения коэффициентов исходного разностного уравнения в диапазоне $[-32,0; 32,0]$ с точностью до трех значащих цифр после запятой (с учетом последующего домножения на 1024).

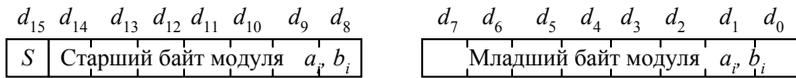


Рис. 6.6

В данном случае в разрядах $d_0 - d_{14}$ записывается модуль коэффициента a_i (b_i) целым двоичным числом в прямом коде, а в разряде d_{15} указывается знак коэффициента: $S = 1$ – коэффициент отрицательный; $S = 0$ – коэффициент положительный.

Реализация арифметических операций. Из полученного разностного уравнения (6.26) видно, что для его реализации необходимы три операции – умножение, сложение и вычитание. Удобно операцию вычитания заменить операцией сложения с использованием отрицательных чисел, представленных в дополнительном коде, т. е. обеспечить выполнение операции алгебраического сложения. Таким образом, для реализации собственно разностного уравнения нужно иметь в программе процедуры умножения, алгебраического сложения и деления результата на фиксированное число (в данном случае – деления на 1024).

Вышеописанные арифметические операции рекомендуется оформить в виде отдельных подпрограмм. Могут быть использованы готовые подпрограммы, тексты которых приведены в приложениях. Таких подпрограмм четыре.

Подпрограмма умножения входного сигнала на весовой коэффициент [4]. Эта подпрограмма вычисления произведения $a_i x[k-i]$ реализует умножение 8-разрядного двоичного значения сигнала $x[k-i]$ на 16-разрядный двоичный коэффициент a_i . Результат умножения является 24-разрядным двоичным числом и формируется в про-

цессе выполнения подпрограммы в аккумуляторе А (старший байт) и регистровой паре HL (младшие два байта), но затем расширяется до 32-разрядного числа (добавлением нулевого старшего байта) и помещается в 4-байтовую ячейку памяти PLUS (см. табл. 6.2), что делается с целью унификации применения подпрограмм умножения.

Алгоритм подпрограммы следующий:

1. Получение знака произведения из знаков операндов и его сохранение в стеке.
2. Очистка знаковых битов операндов (получение модулей).
3. Перемножение операндов методом суммирования частных произведений: путем сдвига произведения вправо и прибавления множимого к произведению для каждого ненулевого бита множителя, выдвигаемого вправо, т. е. старшими разрядами вперед (при этом произведение образуется в регистрах А-HL).
4. Помещение произведения в переменную PLUS.
5. Помещение знака произведения во флажок знака МП.

При этом операцию умножения удобно отобразить схемой, показанной на рис. 6.7.



Рис. 6.7

Подробнее алгоритмы реализации умножения рассмотрены в литературе [4]. Вызов подпрограммы умножения байта на слово осуществляется с адреса 0200H.

Входными регистрами подпрограммы являются регистровая пара DE и регистр-аккумулятор А, хранящие множимое и множитель в прямом коде (знак-модуль). Выходным регистром является флажок знака МП, возвращающий знак произведения, 4-байтовый модуль которого помещается во временную ячейку памяти PLUS.

Подпрограмма сохраняет неизменным значение регистровой пары HL. Текст подпрограммы представлен в прил. 6.1.

Подпрограмма умножения выходного сигнала на весовой коэффициент. Данная подпрограмма реализует вычисление произведений $b_i y[k-i]$, в которых величины b_i и $y[k-i]$ являются 2-байтовыми словами (16-разрядными двоичными целыми числами). Их произведение является 32-разрядным двоичным числом, которое формируется в процессе выполнения подпрограммы в двух регистровых парах – DE и HL (соответственно старшие и младшие два байта), а затем помещается в 4-байтовую ячейку памяти PLUS (см. табл. 6.2).

Подпрограмма алгоритмически реализована аналогично вышеописанной и отличается от нее только длиной множителя (и соответственно распределением регистров). В соответствии с длиной множителя возрастает вдвое число итераций. Соответственно среднее время выполнения данной подпрограммы более чем в 2 раза превышает время выполнения первой подпрограммы, поэтому и следует применять первую подпрограмму везде, где это позволяют размеры операндов.

Вызов подпрограммы умножения слова на слово осуществляется с адреса 0228H.

Входными регистрами подпрограммы являются регистровые пары BC и DE, хранящие множимое и множитель в прямом коде (знак-модуль). Выходным регистром является флажок знака МП, возвращающий знак произведения, 4-байтовый модуль которого помещается во временную ячейку памяти PLUS. Подпрограмма сохраняет неизменным значение регистровой пары HL.

Текст подпрограммы представлен в прил. 6.2.

Подпрограмма сложения/вычитания 4-байтовых чисел. Подпрограмма осуществляет алгебраическое сложение результата выполнения одной из подпрограмм умножения (беззнакового 4-байтового целого в ячейке памяти PLUS) с ячейкой SUM, хранящей сумму $y[k]$ с размещением результата в той же переменной SUM в дополнительном коде. Знак числа, хранимого в PLUS, передается во флаге знака МП.

Алгоритмически подпрограмма состоит из проверки знака слагаемого с последующим разделением на две однотипные процедуры: сложения и вычитания многобайтовых целых чисел, хранимых в памяти и адресуемых указателями в регистровых парах. В данном случае длина операндов фиксирована как 4. Результат операции формируется в дополнительном коде.

Вызов подпрограммы сложения осуществляется с адреса 025CH.

Входным регистром подпрограммы является флажок знака МП, значение которого определяет, какая операция будет выполняться – сложение или вычитание. Выходных регистров нет, результат операции находится в ячейке памяти SUM, хранящей значение $y[k]$.

Текст подпрограммы представлен в прил. 6.3.

Подпрограмма формирования выходного сигнала фильтра. Данная подпрограмма осуществляет деление полученной суммы $y[k]$ на число 1024 для получения выходного значения фильтра. Подпрограмма деления двоичного числа на число, представимое в виде целой степени числа 2, сводится к сдвигу делимого вправо на число бит, равное этой степени (для деления на $1024 = 2^{10}$, надо сдвигать на 10 бит вправо). При этом следует сохранять знак делимого.

Кроме того, предлагаемая подпрограмма реализует округление результата деления до целого (а не отбрасывание дробной части), а также осуществляет проверку возможности превышения результатом деления максимального значения для 2-байтового слова и диагностирует этот факт переходом на процедуру обработки ошибки.

Отметим, что подпрограмма алгебраического сложения образует в ячейке, хранящей $y[k]$, значение суммы в дополнительном коде, в связи с чем перед выполнением собственно деления данная подпрограмма анализирует знак суммы $y[k]$ и в случае отрицательного значения производит его преобразование в прямой код.

Вызов подпрограммы деления осуществляется с адреса 0280H.

Результат деления подпрограмма возвращает в регистровую пару HL в прямом коде (знак-модуль).

Текст подпрограммы представлен в прил. 6.4.

Распределение поля памяти. В распоряжении пользователя до 64 Кбайт ячеек памяти ОЗУ. Согласно заданию на курсовое проектирование наивысший порядок разностного уравнения равен двум, т. е. в памяти обязательно должны присутствовать отсчеты $x[k]$, $x[k-1]$, $x[k-2]$, $y[k]$ (как результат текущего решения разностного уравнения), $y[k-1]$ и $y[k-2]$. Также следует предусмотреть хранение в памяти коэффициентов a_0 , a_1 , a_2 , b_1 и b_2 уравнения (6.26).

Кроме того, в памяти следует предусмотреть ячейки для хранения промежуточных результатов расчета подпрограмм арифметических операций. В предложенных в приложениях подпрограммах это 4-байтовые ячейки PLUS и SUM, в которых хранятся (соответственно) значения модуля результата последней операции умножения и значение $y[k]$ (т. е. домноженное на 1024).

Таким образом, при применении предложенных в приложениях подпрограмм получаем распределение памяти, приведенное в табл. 6.2.

Следует заметить, что данное распределение памяти является только примерным и возможно произвольное распределение ячеек памяти ОЗУ для хранения переменных программы. При использовании предложен-

Таблица 6.2

Адрес ячейки памяти	Размер в байтах	Обозначение в тексте	Описание, комментарий	
0000H 01FFH	512		Рекомендуемое местоположение основной программы расчета	
0200H 0227H	40		Подпрограмма умножения входных сигналов $x[k]$, $x[k - 1]$, $x[k - 2]$ на весовой коэффициент (умножение байта на слово) (*)	
0228H 025BH	52		Подпрограмма умножения выходных сигналов $y[k - 1]$, $y[k - 2]$ на весовой коэффициент (перемножение 2-байтовых слов) (*)	
025CH 027FH	36		Подпрограмма алгебраического сложения четырехбайтовых чисел (*)	
0280H 02C2H	68		Подпрограмма вычисления выходного сигнала фильтра (деления на 1024) (*)	
0400H	4	(PLUS)	Ячейка для модуля результата операций умножения (*)	
0404H	4	$y[k]$ (SUM)	Ячейка для промежуточного значения результата расчета (*)	
0410H	2	a_0	Кoeffициент a_0 Кoeffициент a_1 Кoeffициент a_2 Кoeffициент b_1 Кoeffициент b_2	Младший байт хранится по меньшему адресу, а старший – по большему адресу
0412H	2	a_1		
0414H	2	a_2		
0416H	2	b_1		
0418H	2	b_2		
0420H	1	$x[k - 2]$	Область памяти, отводимая для хранения трех пар входных и выходных сигналов фильтра. При этом выходные сигналы, имеющие размер 2 байта, хранятся младшим байтом по меньшему адресу, старшим байтом – по большему адресу	
0421H	2	$y[k - 2]$		
0423H	1	$x[k - 1]$		
0424H	2	$y[k - 1]$		
0426H	1	$x[k]$		
0427H	2	$y[k]$		
0429H				

ных в приложениях подпрограмм (без их модификации) обязательными из приведенных в табл. 6.2 являются только отмеченные (*).

Следует также заметить, что стартовый адрес всех приведенных подпрограмм совпадает с начальным адресом их размещения в памяти.

В данном примере предполагается, что с адреса 0000 до 03FFH размещается ПЗУ программ, а с адреса 0400H располагаются ячейки ОЗУ.

Приложение 6.1

Подпрограмма осуществляет умножение слова, помещенного в регистровую пару DE, на байт в аккумуляторе A. В данной работе применяется для умножения $x[k]$ помещаемого в A, на коэффициент, помещаемый в регистровую пару DE.

Подпрограмма формирует модуль произведения в ячейке PLUS (4 байта по адресу 0400H), а знак – во флажке знака МП.

Адрес и код	Мнемокод	Комментарии
-------------	----------	-------------

1) предварительная часть – вычисление знака произведения и получение модулей множителей:

0200:	E5	PUSH H	Сохранение в стеке значения HL
	6F	TMOV L, A	Сохранение значения A в L -
	AA	XRA D	Определение знака результата
	F5	PUSH PSW	и сохранение его в стеке
			Удаление знаков множителей (для
			получения их абсолютных значений):
	7A	MOV A,D	Старший байт DE – в аккумулятор
	E6	AMI 7FH	Очистка знакового бита
	7F		("7FH – битовая маска)
	57	MOV D, A	Возврат байта в D
	7D	MOV A,I,	Восстановление A из L
	E6	ANI 7FH	Очистка знакового бита
	7F		(7FH – битовая маска)

2) основная часть – собственно умножение:

020B:	21	LXI H, 0	Очистка места для произведения
	00		в регистровой паре HL
	00		
	06	MVI B,8	Счетчик сдвигов (битов множителя)
	08		(число битов множителя – 8)

0210:	29	DAD H	Сдвиг влево младших двух байтов произведения
	17	RAL	Продолжение сдвига в старший байт
	D2	JNC 0218H	Анализ бита множителя и обход
	18		прибавления частного произведения,
	02		если бит множителя нулевой
	19	DAD D	Прибавление частного произведения
	CE	ACI 0	Прибавление только флага переноса,
	00		как единицы к старшему байту
0218:	05	DCR B	Декремент счетчика битов множителя
	C2	JNZ 0210H	Конец собственно умножения
	10		если счетчик достиг нуля, а иначе -
	02		переход на начало цикла

3) заключительная часть подпрограммы – сохранение результата:

021C:	22	SHLD 0400H	Запись младшего слова результата
	00		в ячейку PLUS
	04		
	6F	MOV L,A	Передача старшего байта произведения в L
	26	MVI H,O	Очистка старшего байта PLUS
	00		(записью нуля в регистр H)
	22	SHLD 0402H	Запись HL в старшую половину
	02		ячейки PLUS
	04		
	F1	POP PSW	Чтение знака произведения из стека
	E1	POP H	Восстановление из стека HL
0227:	C9	RET	Возврат из подпрограммы

Приложение 6.2

Подпрограмма осуществляет перемножение слов, помещенных в регистровой паре DE (множитель) и BC (множимое). В данной работе применяется для умножения $y[k]$ на весовой коэффициент.

Подпрограмма формирует модуль произведения в ячейке PLUS (4 байта по адресу 0400H), а знак – во флажке знака МП.

Адрес и код	Мнемокод	Комментарии
-------------	----------	-------------

1) предварительная часть – вычисление знака произведения и получение модулей множителей:

0228:	E5	PUSH H	Сохранение в стеке значения HL
	7A	MOV A, D	Определение знака результата
	A8	XRA B	“исключающим или” старших байтов в A

F5	PUSH PSW	Сохранение знака результата в стеке
7A	MOV A, D	Передача старшего байта в A
E6	ANI 7FH	Очистка знакового бита
7F		(7FH – битовая маска)
57	MOV D, A	Возврат байта в D
78	MOV A, B	Аналогично для второго сомножителя:
E6	ANI 7FH	Очистка знакового бита
7F		(7FH – битовая маска)
47	MOV B, A	Возврат байта в B (получены абсолютные значения сомножителей)

2) основная часть – собственно умножение:

0234:	21	LXI H,0	Очистка места для произведения
	00		в регистровой паре HL
	00		
	3E	MVI A, 16	Счетчик сдвигов (битов множителя)
	10		(число битов множителя – 16)
0239:	EB	XCHG	Обмен содержимых HL и DE
	29	DAD H	Сдвиг множителя влево
023B:	EB	XCHG	Обратный обмен HL и DE
	02	JNC 0244H	Анализ выдвинутого бита множителя
	44		и обход, если он нулевой
	02		
	09	DAD B	Прибавление частного произведения
	D2	JNG 0244H	Проверка наличия переноса
	44		
	02		
	13	INX D	Прибавление к старшему слову переноса
			от прибавления частного произведения
0244:	3D	DCR A	Декремент счетчика сдвигов
	CA	JZ 0252H	Если все биты множителя обработаны,
	39		то переход на конец умножения
	02		
	29	DAD H	Сдвиг влево произведения
	D2	JNC 0239H	Проверка наличия переноса
	39		и переход на начало цикла, если
	02		переноса нет
	EB	XCHG	Иначе – загрузка множителя из DE в HL
	29	DAD H	Сдвиг множителя влево
	23	INX H	и прибавление переноса из произведения
	C3	JMP 023BH	Продолжение цикла умножения
	3B		

0252:	02 22 00 04 EВ 22 02 04 F1 E1	SHLD 0400H XCHG SHLD 0402H POP PSW POP H	Запись младшего слова результата в младшую половину ячейки PLUS Передача старшего слова в HL Запись старшего слова результата Чтение знака произведения из стека Восстановление из стека HL
025B:	C9	RET	Возврат из подпрограммы

Приложение 6.3

Подпрограмма прибавляет результат умножения, хранимый в ячейке PLUS (4 байта по адресу 0400H), к сумме в ячейке SUM (4 байта по адресу 0404H) с учетом знака результата умножения, передаваемого во флаге знака МП. Результат остается в ячейке SUM в дополнительном коде. В данной работе в ячейке SUM формируется $y[k]$.

Адрес и код	Мнемокод	Комментарии
025C: E5	PUSH H	Сохранение в стеке значения HL
21	LXI H,0400H	Загрузка указателя на слагаемое (для косвенной адресации ячейки PLUS)
04		
11	LXI D,0404H	Указатель на слагаемое-сумму (для косвенной адресации ячейки SUM)
04		
06	MV1 B, 4	Инициализация счетчика байтов (складываем 4-байтовые числа)
04		
FA	JM 0274H	Анализ знака и выбор: сложение или вычитание (переход на вычитание при знаке минус во флажке знака МП, который не меняется предыдущими командами подпрограммы)
74		
02		

Вариант “сложение” (реализуется при нулевом флажке знака МП):

B7	ORA A	Очистка флажка переноса
0269: 1A	LDAX D	Загрузка байта в A по адресу в DE
8E	ADC M	Прибавление косвенно адресуемого байта с учетом флажка переноса
12	STAX D	Сохранение байта суммы по адресу в DE
23	INX H	Продвижение указателей к следующим
13	INX D	байтам слагаемых и суммы

	05	DCR B	Декремент счетчика байтов
	C2	JNZ 0269H	Перевод к началу цикла, если счетчик не достиг нуля
	69		
	02		
	E1	POP H	Восстановление HL из стека
0273:	C9	RET	Выход из подпрограммы

Вариант “вычитание” (реализуется при нулевом флажке знака МП):

0274:	B7	ORA A	Очистка флажка переноса
0275:	1A	LDAX D	Загрузка байта в А по адресу в DE
	9E	SBB M	Вычитание косвенно адресуемого HL-байта с учетом флажка переноса (заема)
	12	STAX D	Сохранение байта суммы по адресу [DE]
	23	INX H	Продвижение указателей к следующим байтам
	13	INX D	
	05	DCR B	Декремент счетчика байтов
	C2	JNZ 0275H	Переход к началу цикла, если счетчик не достиг нуля
	75		
	02		
	E1	POP H	Восстановление HL из стека
027F:	C9	RET	Выход из подпрограммы

Приложение 6.4

Подпрограмма осуществляет деление суммы в ячейке SUM (4 байта ОЗУ по адресу 0404H) на 1024 с помещением результата в регистровую пару HL.

Адрес и код	Мнемокод	Комментарии
0280: 3A	LDA 0407H	Загрузка старшего байта суммы в А
07		
04		
E6	AMI 80H	Проверка бита знака суммы (80H – маска бита знака)
80		
CA	JZ 029FH	Проверка знака суммы и переход к собственно делению, если сумма положительная
9F		
02		

Преобразование отрицательной суммы в прямой код:

	21	LXI H,0404H	HL – указатель на ячейку SUM (для косвенной адресации)
	04		
	04		

	06	MVI B, 4	Инициализация счетчика байтов суммы (ячейка SUM 4-байтовая)
	04		
	37	STC	Загрузка единицы во флажок переноса
02BE:	7E	MOV A, M	Загрузка в A байта суммы
	2F	CMA	Инvertирование его
	DE	SBI 0	Вычитание из суммы единицы, загруженной во флажок переноса (заема)
	00		
	77	MOV M,A	Байт помещается на прежнее место
	23	INX H	Продвижение указателя
	05	DCR B	Декремент счетчика байтов
	C2	JNZ 028EH	Конец цикла преобразования, если счетчик достиг нуля
	8E		
	02		
	CD	CALL 029FH	Деление результата на 1024 (вызов основной части подпрограммы как другой подпрограммы)
	9F		
	02		
	F6	ORI 80H	Установка знакового бита (80H – маска знакового бита)
	80		
	67	MOV H,A	Запись старшего байта в HL
029E:	C9	RET	Возврат из подпрограммы для случая отрицательной суммы

Собственно деление положительной суммы в ячейке SUM на 1024:

029F:	2A	LHLD 0406H	Загрузка старшего слова суммы в HL (ячейка SUM, старшие два байта)
	06		
	04		
	7C	MOV A, I	Передача старшего байта в A
	E6	AMI 7EH	Анализ наличия переполнения в SUM (7EH – маска недопустимых битов)
	7E		
	C2	JNZ 02C2H	При переполнении – переход на ошибку
	C2		
	02		
	EB	XCHG	Передача старшего слова суммы в DE
	2A	LHLD 0404H	Загрузка младшего слова в HL (ячейка SUM, младшие два байта)
	04		
	04		
	7A	MOV A, D	Подготовка к сдвигу слова вправо
	1F	RAR	Выдвижение младшего бита D во флаг переноса
	7B	MOV A,E	Загрузка в A байта из регистра E
	1F	RAR	и сдвиг его на один бит вправо
	47	MOV B, A	Временное сохранение старшего байта
	7C	MOV A, H	Загрузка следующего байта и
	1F	RAR	его сдвиг вправо

6F	MOV L, A	Запись в младший байт
78	MOV A, B	Восстановления в A старшего байта
B7	ORA A	Только очистка флага переноса
1F	RAR	Последний сдвиг старшего байта
67	MOV H, A	Запись старшего байта в H
7D	MOV A, L	Загрузка в A младшего байта
1F	RAR	(в L попадает бит, выдвинутый из H)
CE	ACI 0	Прибавление бита с весом 1/2 для
00		округления (этот бит во флаге переноса)
6F	MOV L, A	Сохранение младшего байта
7C	MOV A, H	
CE	ACI 0	Учет возможности переноса в старший
00		байт при округлении
67	MOV H, A	Сохранение старшего байта
02C1: C9	RET	Возврат из подпрограммы деления (в случае
		положительной суммы – это выход из под-
		программы, в случае отрицательной суммы –
		это продолжение обработки)
		Обработка ошибки переполнения:
02C2: 76	HLT	процедура обработки ошибки
00	NOP	переполнения может быть выполнена
00	NOP	занесением в эти три байта команды перехода
		на нужный адрес

Библиографический список

1. *Зиятдинов С. И., Красильникова О. И., Кулыгина Л. А.* Электронные устройства на базе микропроцессоров широкого применения: Текст лекций/ СПб ГУАП. СПб., 1995.
2. *Бесекерский В.А.* Цифровые автоматические системы. М.: Наука, 1976.
3. *Хвоц С. Т., Варлинский Н. Н., Попов Е. А.* Микропроцессоры и микро-ЭВМ в системах автоматического управления: Справочник/ Под ред. *С. Т. Хвоца*. Л.: Машиностроение, 1987.
4. *Злобин В. К., Григорьев В. Л.* Программирование арифметических операций в микропроцессорах: Учеб. пособие для технических вузов. М.: Высш. шк., 1991.
5. Применение микропроцессоров в радиосистемах управления: Учеб. пособие/ Под ред. *В. А. Бесекерского*. М.: Изд-во МПИ, 1991.
6. *Зиятдинов С. И., Кулыгина Л. А.* Радиоэлектронное оборудование летательных аппаратов: Текст лекций/ ЛИАП. Л., 1988.
7. *Шило В. Л.* Популярныe цифровые микросхемы: Справочник. М.: Радио и связь, 1988.
8. *Грушвицкий Р. И., Мурсаев А. Х., Смолов В. Б.* Аналого-цифровые периферийные устройства микропроцессорных систем. Л.: Энергоатомиздат, 1989.

ОГЛАВЛЕНИЕ

Введение	3
1. ОБЩИЕ СВЕДЕНИЯ О СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЯХ ...	5
1.1. Обобщенная структура специализированного вычислителя	5
1.2. Обзор современных микропроцессоров	6
1.2.1. Микроконтроллеры фирмы Atmel	7
1.2.2. PIC-контроллеры фирмы Microchip	9
1.2.3. Микроконтроллеры SX18AC/SX28AC фирмы Scenix	12
1.2.4. Микроконтроллеры фирмы Motorola	13
1.2.5. Микроконтроллеры семейства Z8 фирмы Zilog	15
2. АРИФМЕТИЧЕСКИЕ ОСНОВЫ МИКРОПРОЦЕССОРНЫХ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ	20
2.1. Сложение двух многобайтовых чисел	20
2.2. Умножение однобайтовых чисел	20
2.3. Умножение однобайтовых чисел со знаком в прямом коде	21
2.4. Умножение однобайтовых чисел в дополнительном коде	23
3. СИСТЕМА ПАМЯТИ МИКРОПРОЦЕССОРНЫХ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ	25
3.1. Классификация запоминающих устройств	25
3.2. Организация памяти	28
4. ИНТЕРФЕЙС ВВОДА-ВЫВОДА ИНФОРМАЦИИ	30
4.1. Общие сведения о микросхемах цифроаналоговых и аналого- цифровых преобразователей	30
4.2. Справочные сведения о микросхемах ЦАП и АЦП	31
4.3. Подключение АЦП и ЦАП к микропроцессору	33
5. УСТРОЙСТВА ЦИФРОВОЙ ОБРАБОТКИ ИНФОРМАЦИИ С ЖЕСТКОЙ ЛОГИКОЙ	37
5.1. Устройства хранения многоразрядных чисел	37
5.2. Умножитель (16×16) КР1802ВР5	38
5.3. Параллельный 4-входовый сумматор КР1802ИМ1	40
6. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	42
6.1. Цифровая фильтрация	42
6.2. Дифференцирование цифровых последовательностей	50
6.3. Интегрирование цифровых последовательностей	55
6.4. Микропроцессорный следящий измеритель	58
Приложение 6.1	66
Приложение 6.2	67
Приложение 6.3	69
Приложение 6.4	70
Библиографический список	73

Учебное издание

Зиятдинов Сергей Ильич
Осипов Леонид Андроникович

ПРОЕКТИРОВАНИЕ
СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ
ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Учебное пособие

Редактор *А. В. Подчепалева*
Компьютерная верстка *А. Н. Колешко*

Сдано в набор 19.08.02. Подписано к печати 18.10.02. Формат 60×84 1/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 4,2. Усл. кр.-отг. 4,3. Уч. -изд. л. 4,5. Тираж 150 экз. Заказ № 443

Редакционно-издательский отдел
Отдел электронных публикаций и библиографии библиотеки
Отдел оперативной полиграфии
СПбГУАП

190000, Санкт-Петербург, ул. Б. Морская, 67