Министерство образования и науки Российской Федерации
Нижнекамский химико-технологический институт (филиал)
Федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Казанский национальный исследовательский технологический университет»

Г.П. Сечина

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНО-ПРАКТИЧЕСКИХ РАБОТ

Нижнекамск
2012

УДК 621.38
С 33

Печатается по решению редакционно-издательского совета Нижнекамского химико-технологического института (филиал) ФГБОУ ВПО «КНИТУ».

Рецензенты:
Саримов Н.Н., кандидат физико-математических наук;
Изотов П.В., нач. АСУТП завода «Этилен».

Сечина, Г.П.
С 33 Микропроцессорные средства : учебно-методическое пособие для выполнения лабораторно-практических работ / Г.П. Сечина. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2012. - 60 с.

Рассмотрены вопросы практического освоения методики программирования в кодах микропроцессора Intel 80x86. Приведены задания для самостоятельной работы студентов.

Предназначено для лабораторного обеспечения курсов «Микропроцессорные средства» по специальности 230102 «Автоматизированные системы обработки информации и управления» и «Микропроцессоры и микро-ЭВМ» для специальности 220301 «Автоматизация технологических процессов и производств».

Подготовлено на кафедре автоматизации технологических процессов и производств Нижнекамского химико-технологического института КНИТУ.

УДК 621.38

Содержание

ВВЕДЕНИЕ

Данное пособие знакомит студентов с арифметическими основами ЭВМ. Рассмотрены системы счисления, правила перевода из одной системы счисления в другую. Приведены способы ручного и машинного счета.

Пособие знакомит студентов с внутренней структурой микропроцессора, системой команд, способами адресации данных форматом команд. Приведены основные характеристики микропроцессора и набор БИС МИК серии 580.

В пособии рассмотрен стандартный узел арифметико-логического устройства АЛУ, приведены набор логических и арифметико-логических операций и особенностей их выполнения в АЛУ.

При разработке пособия использовались работы [1,2,3].

ГЛАВА 1. АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ

1. Общие сведения. Системы счисления.

Совокупность цифр (символов, значков) с их названиями, которые позволяют выразить письменно или устно любое число, вместе с правилами выполнения арифметических действий над числами называется **системой счисления**. Различают непозиционные и позиционные системы счисления.

Непозиционной называется такая система счисления, в которой величина цифры (символа) не зависит от позиции (места), занимаемой ею в записи числа. Примером непозиционной системы счисления является римская система, которую использовали древние римляне, египтяне, вавилоняне и др. народы. Алфавит этой системы состоит из специальных символов, обозначающих следующие величины

I=1; V=5; X=10; L=50;C=100; D=500; M=1000

Например, XXX = 30; число состоит из трех цифр X, каждая из которых независимо от их места в числе равна 10.

Для изображения чисел, отличных от 1, 5, 10, 50, 100, 500 и 1000, приходится комбинировать определенное количество римских цифр, помня при этом, что меньшая по величине цифра, стоящая справа от большей, складывается с большей (например, CXXXIII=133), а стоящая слева - вычитается из большей (например, IX = 10 - 1 = 9).

Арифметические действия над римскими числами выполнять очень трудно, поэтому эта система в настоящее время используется в редких случаях.

Позиционной называется такая система счисления, в которой величина (вес) цифры зависит от позиции (места), которую занимает цифра в записи числа. Примером позиционной системы является десятичная система счисления.

Например, десятичное число 111,11. В зависимости от позиции, занимаемой цифрой 1, ее вес различен в записи числа.

1 ая - 1 сотня,
2 ая - 1 десяток,
3 тья - 1 единица,
4 ая - десятая доля единицы,
5 ая - сотая доля единицы.

В вычислительной технике кроме десятичной системы используются двоичная, восьмеричная и шестнадцатеричная системы счисления. Все они - позиционные.

Десятичная система счисления

Для записи любого числа используются десять цифр от 0 до 9. Общее свойство позиционных систем: вес (величина) цифры при каждом переходе влево от запятой увеличивается во столько раз, чему равно основание системы. При переходе же цифры на один разряд от запятой вправо вес цифры уменьшается во столько раз, чему равно основание системы.

Следовательно, любое десятичное число является компактной записью.

Например: 327,48 это

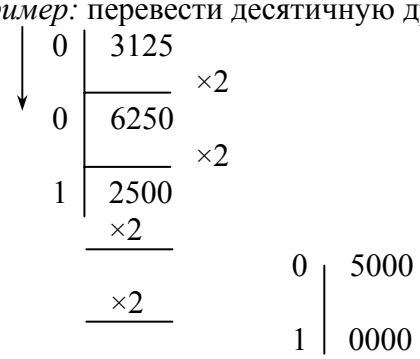
$$3\cdot 100 + 2\cdot 10 + 7 + 4\cdot\frac{1}{10} + 8\cdot\frac{1}{100} \text{ или}$$

$$3\cdot 10^2 + 2\cdot 10^1 + 7\cdot 10^0 + 4\cdot 10^{-1} + 8\cdot 10^{-2}$$

Двоичная система счисления

В этой системе используются только две цифры: 0 и 1. В целой части двоичного числа в самом младшем разряде записывается единица и т.д.

Пример: перевести десятичную дробь в двоичную систему:



$0.3125_{10} = 0.0101_2$

Перевод неправильных дробей осуществляется по приведенным правилам отдельно для целой и дробной части.

Двоичная система используется в ЭВМ всех размеров (от суперЭВМ до микро-ЭВМ), т.к. именно в двоичной форме внутри машины запоминается, перемещается из одного устройства в другое и перерабатывается вся информация: исходные данные и команды программы. Обусловлено это в основном следующим. Чтобы представить в машине цифру, нужно иметь такое устройство (электрическую схему), которое имело бы ровно такое же количество различимых устойчивых состояний равновесия, которое равно основанию системы. Так, чтобы иметь возможность зафиксировать любую десятичную цифру, такое устройство должно иметь 10 состояний. Реализовать технически такое устройство очень сложно. Для фиксирования восьмеричных цифр, устройство должно обладать восемью устойчивыми состояниями (задача технической реализации упрощается), а в случае двоичных цифр - только двумя (что реализовать технически очень просто).

Правила выполнения арифметических действий над двоичными числами оказываются значительно более простыми, чем соответствующие правила для чисел,

представленных в других системах счисления, основание которых больше двух.

Один разряд (цифра) двоичного числа называется битом. Бит является минимальной единицей информации и может содержать либо 0, либо 1. Совокупность трех двоичных цифр, выражающая одну восьмеричную цифру, называют **двоичной триадой.**

0	1	2	3	45	67	- восьмеричная
000	001	010	011	100 101	110 111	–двоичная

Восьмеричная система счисления

В восьмеричной системе счисления используется восемь цифр от 0 до 7

Восьмеричная система используется как вспомогательная при подготовке задач к решению. Она удобна тем, что дает сокращенную запись двоичного числа.

Шестнадцатиричная система счисления

В современных ЭВМ в качестве основной единицы информации принята группа из восьми двоичных разрядов (битов), так называемый **байт** и кратные ему единицы: полбайта, 2 байта, 4 байта и т.д. В связи с этим получила широкое применение шестнадцатиричная система счисления.

В этой системе счисления используется 16 цифр. Младшие цифры , от 0 до 9, совпадают с десятичными (и по написанию, и по величине), а для обозначения остальных цифр используют начальные заглавные буквы латинского алфавита.

Таблица 1

Система счисления			
Шестнадцати-ричная	Десятичная	Восьмеричная	Двоичная
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111
10	16	20	10000
11	17	21	10001
12	18	22	10010

Порядковое число на единицу большее, чем последнее, имеющее собственный символ F, записывается так: берется самая младшая значащая цифра и сдвигается на один разряд влево (факт сдвига обозначается приписыванием нуля справа).

Далее в младшем разряде перебираются подряд все цифры, вплоть до самой старшей (IF), после чего записывается 20 и т.д.

2. Перевод из одной системы счисления в другую неправильных дробей

Осуществляется раздельно для целой и дробной части, после чего обе части (целая и дробная) объединяются в одно число.

3.Арифметические действия с двоичными числами

Сложение двоичных чисел

Сложение двоичных чисел осуществляется следующим образом:

0 + 0 = 0 1 + 0 = 1

0 + 1 = 1 1 + 1 = 10 (два)

Пример. Найти сумму двух чисел 1101 и 101
(1 - единица переноса в старший разряд)

```
  1101 (13)
+  101 (5)
------
 10010 (18)
```

Проверка: $1*2^4 +0*2^3 + 0*2^2 + 1 * 2^1 + 0*2^0 = 1*16 +0*8 +0*4 + +1*2 +0*1 = 18$

Вычитание двоичных чисел

Вычитание двоичных чисел осуществляется следующим образом:

0 - 0 = 0 1 - 1 = 0

1 - 0 = 1 10 - 1 = 1

Пример: Вычесть из 10010 число 101

```
  10010 (18)
− 101 (5)
-------
  01101 (13)
```

Проверка: Аналогична предыдущему примеру.

Умножение двоичных чисел

Умножение двоичных чисел выполняется следующим образом:

0 * 0 = 0 1 * 0 = 0

0 * 1 = 0 1 * 1 = 1

Пример. Перемножить два двоичных числа

```
   1001 (9)
*  101 (5)
-------
   1001
+ 10010
-------
 101101 (45)
```

Проверка. Аналогична предыдущему примеру.

Деление двоичных чисел

Правила деления в двоичной системе аналогичны делению в десятичной системе.

4. Формы представления чисел в машинах

В ЭВМ применяются две формы представления чисел: с фиксированной и плавающей запятой. Соответственно и ЭВМ подразделяются на машины с фиксированной и плавающей запятой.

Машины с фиксированной запятой

При конструировании таких машин заранее устанавливают, какое количество разрядов отводится для целой части числа, а какое для дробной. Обычно, запятая фиксируется перед первым (старшим) цифровым разрядом, и машина оперирует с числами меньше единицы. При решении задач на такой машине число масштабируют так, чтобы в процессе вычислений не получился результат, превышающий единицу или равный единице, т.к. в этом случае наступает переполнение разрядной сетки машины и результат искажается. При выборе количества разрядов в машине ориентируются на требуемую точность вычислений. Для точности вычислений в n-десятичных знаков в машине выбирают 4n-двоичных разрядов. Например, точность вычислений до 10 десятичных знаков обеспечивается разрядной сеткой в 40 двоичных разрядов.

Разрядная сетка машины выглядит следующим образом.

знак числа	2^{-1}	2^{-2}	2^{-3}	… …. …	2^{-n}

Рисунок 1а

1	1	0	1	… … …	1

Рисунок 1б

знак числа	2^{-1}	2^{-2}	2^{-3}	…	2^{n}	знак порядка	2^{1}	2^{2}	…	2^{p}

Рисунок 1в

0	1	1	0	…	1	1	1	0	…	1

Рисунок 1г

Рис.1а - каждый разряд числа записывается в строго определенном месте. Знак числа в машине также представляется двоичными цифрами: (плюс - «0», минус - «1»).

Например. Число - 0,101…1 запишется в разрядной сетке так, как показано на Рис.1б.

Недостатками машин с фиксированной запятой являются:
1. Необходимость предварительного расчета и ввода в машину масштабных коэффициентов, что является довольно сложной работой.
2. Относительная точность работы машины зависит от величины поступающих чисел и является максимальной при проведении действий с максимально возможными числами.

Машины с плавающей запятой

В машинах с плавающей запятой числа представляются в виде двух групп: мантиссы и порядка числа (так называемая нормальная форма представления). Допустим, имеется десятичное число 312,446. Это число можно записать различными способами: $3{,}12446 * 10^{2}$; $0{,}0312446 * 10^{4}$ и т.д. Вообще любое число может быть записано так:

$$N = m * q^{p}, \quad \text{где}$$

m - дробная часть числа (мантисса);
q - основание системы счисления;
p - целое число, называемое порядком числа N (порядок указывает положение запятой в числе).

При различных порядках положение запятой будет различным (по этой причине машины рассматриваемого типа и стали называть машинами с «плавающей запятой»).

Чтобы устранить неоднозначность представления числа применяют нормализованную форму представления чисел. Число считается нормализованным, если его мантисса представляет правильную дробь. В двоичной системе это условие выражается как $0{,}1 \le m < 1$.

Например. Числа $0{,}101*2^{10}$; $0{,}1001*2^{-10}$ являются нормализованными.

Разрядная сетка машины с плавающей запятой представлена на рис.1в.

Например. На рис.1г приводится число +0,110…1*2, записанное в разрядной сетке.

Нормализация чисел в процессе вычислений производится на машине автоматически. При этом мантисса числа сдвигается влево до момента появления в старшем разряде сетки ближайшей единицы. Тут же производится соответствующее уменьшение порядка числа. В случае переполнения разрядной сетки производится нормализация вправо на один разряд.

Большим преимуществом машин с плавающей запятой является больший диапазон представляемых чисел без масштабирования по сравнению с машинами с фиксированной запятой. Количество двоичных разрядов для мантиссы обычно составляет 35 - 45, а для порядка - 6 - 8 разрядов.

Недостатком машин с плавающей запятой является усложнение оборудования, что связано со значительным

усложнением выполнения операций и увеличением времени их выполнения.

5. Изображение отрицательных чисел в прямом и дополнительном кодах

В ЭВМ применяют специальные коды: прямой, обратный и дополнительный. Обратный и дополнительный используются для замены операции вычитания сложением. Прямой код применяется при умножении и делении. Кроме того, он используется для представления отрицательных чисел в запоминающем устройстве машины и ряде устройств, где другие коды не применяются. Изображение положительных чисел во всех кодах совпадает.

Прямой код

Число X в прямом коде изображается [X] пр. Прямой код числа X получается по следующему правилу:

Если X = + 0,X1*X2*X3…, то [X]пр = 0,X1*X2*X3…

Если X = - 0,X1*X2*X3…, то [X]пр = 1,X1*X2*X3…

Прямой код двоичного числа совпадает по изображению с записью самого числа, но в разряде знака ставится 0, если число положительное; и 1, если число отрицательное.

Пример. Пусть имеется число -0,1011,, которое требуется перевести в прямой код

X = -0,1011 и [X]пр = 1,1011

Обратный код

Число X в обратном коде обозначается [X]обр. При X >0 [X]обр = [X]пр = X.

Для отрицательного числа, обратный код получается по следующему правилу: в знаковый разряд числа записывается единица, а в цифровых разрядах нули заменяются единицами, а единицы нулями.

В общем виде отрицательное число X = -0,X1*X2*X3… будет изображаться в обратном коде как [X]обр = $\overline{1, X1 * X2 * X3}$… (черта означает обращение разряда, т.е. инвертирование).

Пример. Перевести число - 0,10001 в обратный код.

X = -0,10001 и [X]обр = 1,01110

Дополнительный код

Дополнительный код числа X обозначается [X]доп. При X > 0, [X]доп = [X]пр = X

Для отрицательного числа дополнительный код получается путем обращения разрядов числа (аналогично тому, как это выполняется в обратном коде), после чего в младший разряд числа прибавляется единица.

Для отрицательного числа X=-0,X1*X2*X3… дополнительный код будет [X]доп = $\overline{1, X1 * X2 * X3}$ +0,0…1

Пример. Перевести число -0,10111 в дополнительный код.

X = -0,10111 и [X]доп = 1,01001

Особенность дополнительного кода заключается в том, что так же как и в десятичной системе, здесь можно брать дополнение любого отрицательного числа до основания системы (до десяти в десятичной и до двух в двоичной системе счисления). При этом необходимо иметь в виду, что дополнение числа до 10 (двойка в двоичном коде) равно его дополнению до 1 плюс 1. Это значит, что дополнение числа до двух можно

получить, взяв дополнение каждого разряда двоичного числа до 1 и добавляя затем 1 в младший разряд.

Так и поступают в машинах, т.к. это дает возможность получать дополнение до 1 в каждом двоичном разряде простым обращением разрядов. Действительно, если поменять нули на единицы и единицы на нули в нашем числе 10011, то получим тот же результат, что и при вычитании его из единиц: 10011-01100. Из всего сказанного вытекает, что

[X]доп = [X]обр +0,0...1

Таким образом, чтобы получить число в дополнительном коде, его сначала преобразуют в обратный код (путем простого обращения разрядов), а затем к младшему разряду полученного числа прибавляют единицу.

Пример. Пусть из 5 требуется вычесть 2.

Обычный метод

а) в десятичной системе:	б) в двоичной системе:
5	101 (5)
– 2	+ 010 (2)
3	011 (3)

В дополнительном коде

а) в десятичной системе:

Возьмем дополнение отрицательного числа (вычитаемого) до десяти (основание десятичной системы) и сложим его с уменьшаемым:

$$\begin{array}{r} 10 \\ -\ 2 \\ \hline 8 \end{array} \qquad\qquad \begin{array}{r} \overline{5} \\ -\ 8 \\ \hline \overline{13} \end{array}$$

Единица переноса отбрасывается и остается результат, равный трем.

б) в двоичной системе:

Возьмем дополнение вычитаемого до двух (основание двоичной системы) и сложим его с уменьшаемым:

$$\begin{array}{r} 1000 \\ -\ 010 \\ \hline 110 \text{ (дополнительный код числа 010)} \end{array} \qquad \begin{array}{r} -\ 101 \\ 110 \\ \hline \underline{1}011 \end{array}$$

Единица переноса отбрасывается и остается результат $011_2 = 3_{10}$

В обратном коде

а) в десятичной системе:

Возьмем дополнение вычитаемого до девяти (основание минус единица) и сложим его с уменьшаемым:

$$\begin{array}{r} -\ 9 \\ 2 \\ \hline 7 \text{ (дополнение числа 2 до 9)} \end{array} \qquad \begin{array}{r} 5 \\ +\ 7 \\ \hline \overset{\curvearrowright}{\underline{1}}2 \\ +\ \quad 1 \\ \hline 3 \end{array}$$

Возникающая единица переноса прибавляется к младшему разряду суммы и получается результат, равный трем.

б) в двоичной системе:

Возьмем дополнение вычитаемого до единицы (основание два минус единица) сложим его с уменьшаемым:

```
  111                          101
− 010                        + 101
-----                        -----
101 (обратный код числа 010)  1010      +
                              ↳1
                              -----
                               011
```

Возникшая единица переноса прибавляется к младшему разряду суммы кодов и получается результат, равный трем.

Из рассмотренного следует:

1. Как дополнительный, так и обратный коды позволяют заменить операцию вычитания сложением.
2. При сложении чисел в дополнительном коде возникающая единица переноса отбрасывается, а при сложении в обратном коде эта единица прибавляется к младшему разряду суммы кодов.

6. Сложение чисел в машинах

Сложение чисел с учетом их знаков на машине представляет собой последовательность следующих действий:

1. Преобразование прямого кода исходных чисел в обратный (дополнительный).
2. Поразрядное сложение кодов.
3. Преобразование результата в прямой код при посылке в другие устройства машины.

Сложение чисел в машинах с фиксированной запятой

Сложение в обратом коде

Обратные коды чисел складываются поразрядно, причем знаковые разряды складываются как разряды мантисс. Если в результате сложения кодов в знаковом разряде возникает единица переноса, она прибавляется к младшему разряду суммы кодов (эта операция называется циклическим переносом). При обратном коде дополнение числа берется до единицы.

При сложении обратных кодов могут встретиться следующие четыре случая:

1**.** X>0, У>0, X+У>0

Этот случай не отображает особенностей обратного кода, т.к. числа положительны, а для положительных чисел обратный код совпадает с прямым.

2**.** X >0, У< О, X+У<0

Рассмотрим пример и будем складывать числа в прямом и обратном кодах и сравнивать результат с ручным счетом.

Ручной счет:	**Сложение в обратном коде на машине:**
X=0,001001	[X]обр=0,001001
У=-0,110001	[У]обр=1,001110
X+У=-0,101000	[X+У]обр=1,010111

Результат получился в обратном коде, т.к. сумма отрицательна. Переведем ее в прямой код:

[1,010111]обр → [1,101000]пр → -0,101000

Результат совпадает с ручным счетом. В данном случае циклический перенос не образуется.

3. X > 0; У < 0**;** X + У > 0

Ручной счет:
X=0,110001
У=-0,001001
X+У=0,101000

Сложение на машине:
[X]обр=0,110001
[У]обр=1,110110
[X+У]обр=10,100111

1

[X+У]обр=0,101000=[X+У]пр

Результат совпадает с ручным счетом. Здесь образуется циклический перенос. Единица переноса, получившаяся в знаковом разряде, прибавляется к младшему разряду суммы обратных кодов.

4. $X < 0$; $У < 0$; $X + У < 0$

Ручной счет:
X=-0,110001
У=-0,001001
X+У=-0,111010

Сложение на машине:
[X]обр=1,001110
[У]обр=1,110110
[X+У]обр=11,000100

1

[X+У]обр=1,000101

Результат получился в обратном коде. Переведем его в прямой код.

[1,000101]обр → [1,111010]пр → -0,111010

Что совпадает с ручным счетом.

Модифицированный обратный код

При неправильно подобранном масштабе результат сложения может оказаться больше единицы.

Например: X=0,101011
У=0,110100
X+У=1,011111 [1, →(-)]=-0,……

Произошло переполнение разрядной сетки (сумма превысила единицу). Машина может воспринять результат как отрицательный.

Для обнаружения переполнения ввели модифицированный обратный код, отличающийся от обычного обратного кода тем, что под знак числа в нем отводится не один, а два разряда .

Запись чисел в модифицированном коде

а) для положительного числа:

X=0,X1 X2 X3… $[X]^{м}_{обр}$ = 00,X1 X2 X3…

б) для отрицательного числа:

X=-0,X1 X2 X3… $[X]^{м}_{обр}$ = 11,X1 X2 X3…

Комбинация знаковых разрядов 01, или 10, будет служить признаком переполнения разрядной сетки.

Тот же пример в модифицированном коде :

X=00,101011
У=00,110100
X+У=01,011111

В случае переполнения машина останавливается.

Сложение чисел в модифицированном обратном коде ничем не отличается от сложения в обычном обратном коде.

Сложение чисел в дополнительном коде

Сложение чисел в дополнительном коде происходит аналогично сложению в обратном коде. Разница в том, что единица переноса, образующаяся при сложении знаковых разрядов, отбрасывается.

Пример: Даны два числа: X=0,101001 и У=-0,001101.
Сложить их в дополнительном коде.

[X]доп=0,101001
[У]доп=1,110011
[X+У]доп=<u>1</u>0,011100=00,011100

Первую единицу (<u>1</u>) отбрасываем.

Модифицированный дополнительный код

Как и в модифицированном обратном коде в модифицированном дополнительном предусматриваются два знаковых разряда :

а) для положительного числа:

X=0,X1 X2 X3… $[X]^{м}_{обр}$ = 00,X1 X2 X3…

б) для отрицательного числа:

X=-0,X1 X2 X3… $[X]^{м}_{обр}$ = 11,$\overline{X1}$ $\overline{X2}$ $\overline{X3}$…

Решим последний пример в модифицированном дополнительном коде:

[X]доп=00,101001
[У]доп=11,110011
[X+У]доп=<u>1</u>00,011100=00,011100

Первую единицу (<u>1</u>) отбрасываем.

Получаем результат с двумя знаковыми разрядами.

При сложении чисел в машинах с плавающей запятой сначала уравниваются порядки слагаемых, а затем складываются их мантиссы. Порядком суммы является общий порядок слагаемых. Уравнивание порядков заключается в том, что меньший порядок числа увеличивается до большего, при этом соответственно изменяется и мантисса.

Мантиссы складываются в одном из модифицированных кодов. Если после сложения результат оказался ненормализованным, то его нормализуют, изменяя соответственно и порядок.

Пример: Сложить два числа: $X=+0,100100\cdot2^{100}$ и
$У= -0,100101\cdot2^{110}$

Проведем вычисления в модифицированном обратном коде.

1. Уравниваем порядки чисел :

X 0 100100 0 100 (до уравнивания порядков)

X 0 001001 0 110 (после уравнивания порядков)

2. Переведем мантиссы чисел в модифицированный обратный код и сложим их :

X 00 001001

У 11 011010

X+У= 11 100011

3. Переведем результат в прямой код:

X+У= 1 011100 0 110

4. Нормализуем его:

X+У= 1 111000 0 101

Результат сложения равен: $X+У=-0,111000\cdot2^{101}$

7. Умножение чисел в машинах

Точность умножения двух чисел зависит от количества значащих цифр произведения, которым ограничиваются при расчетах. В пределе это количество вдвое превышает число значащих цифр сомножителей.

Ограниченное количество разрядов в запоминающем устройстве и других устройствах машины вынуждает, как правило, ограничиваться тем же количеством значащих цифр, которое имели сомножители.

При умножении происходит выход за пределы разрядной сетки. Если сомножители содержат только дробную часть, то возможен выход за пределы разрядной сетки только со стороны младших разрядов. Это допускается, поскольку правила приближенных вычислений рекомендуют оставлять в

произведении столько же значащих цифр, сколько их содержится в наименее точном из сомножителей.

Числа в машинах умножаются в прямом коде по обычным правилам арифметики двоичных чисел. Знак произведения определяется по сумме знаков сомножителей следующим образом:

0 + 0 = 0 0 + 1 = 1
0 + 1 = 1 1 + 1 = 0

В машинах используются два способа умножения.

Первый способ

Пример:

```
  0,1001
* 0,0101
--------
  01001
 00000
01001
00000
---------
0,00101101
```

Умножение начинается с младших разрядов множителя. Умножаем каждый разряд множителя, начиная с младшего, и записываем частное произведение, т.е. множимое, если этот разряд множителя содержит 1. Записывая очередное частное произведение, мы сдвигаем его по отношению к предыдущему на разряд влево. Нули, если разряд множителя содержит 0, обычно не записываются, а просто на лишний разряд влево сдвигается запись очередного частного произведения. Получается, лесенка влево. Вместо этой лесенки можно каждое частное произведение прямо подсуммировать под чертой, сдвигая затем получившуюся сумму вправо. При реализации первого способа умножения на машине так и поступают.

Тот же пример умножения на машине.

Сумматор	Σ	00000000	Множитель
прибавить	(множимое)	10010000	1
	Σ	10010000	
сдвинуть		01001000	
		(не прибавлять)	0
сдвинуть		00100100	
прибавить		10010000	1
	Σ	10110100	
сдвинуть		01011010	
		(не прибавлять)	0
сдвинуть		00101101	
	Σ	00101101	

Результат совпадает с результатом ручного счета.

Подсуммирование частных произведений производится только при наличии единицы в данном разряде множителя. В противном случае осуществляется только сдвиг.

Второй способ

Умножение начинается со старших разрядов множителя, при этом сдвигается множимое (при ручном умножении это даст "лесенку" вправо), а в сумматоре происходит только подсуммирование каждого частного произведения. Чтобы сохранить возможно большее число младших разрядов множимого (при сдвиге они теряются) в машинах, работающих по такому способу, вводят дополнительные разряды в суммирующем устройстве.

Если в первом способе процесс умножения начинается с суммирования, то во втором он начинается со сдвига.

В машинах с плавающей запятой процесс умножения протекает аналогично, с той лишь разницей, что добавляются операции определения порядка произведения путем алгебраического сложения порядков сомножителей и нормализации результата (сама операция умножения производится с мантиссами чисел без выравнивания порядков).

8. Деление чисел в машинах

Если умножение выполняется на машине посредством многократных сдвигов и сложений, то деление, будучи операцией, обратной умножению, выполняется посредством многократных сдвигов и вычитаний. В машинах с фиксированной запятой деление возможно при условии, что делимое по модулю меньше делителя, в противном случае произойдет переполнение разрядной сетки.

Так же как и при ручном счете, разряды частного при делении чисел на машине определяются (начиная со старшего) путем последовательного вычитания делителя из остатка, полученного от предыдущего вычитания. После каждого вычитания остаток, находящийся в сумматоре, сдвигается влево по отношению к делителю. Знак частного, как и при умножении, определяется по сумме знаков делимого и делителя.

Поскольку в машинах операция вычитания не выполняется, последовательное вычитание заменяется сложением остатков с обратным или дополнительным кодом делителя. Остатки также получаются в соответствующем коде.

Ручной способ

Пример: X= + 0,1001; Y= + 0,1101

```
_ 10010     | 1101
  1101      0,1011
 10100
–  1101
  1110
 –1101
     1
```

Здесь после каждого вычитания делитель сдвигается вправо по отношению к делимому. Если остаток после вычитания получается положительный, в разряд частного записывается 1, если отрицательный, то 0. На практике обычно отрицательный остаток не записывается, просто делитель дополнительно сдвигается на один разряд вправо и вычитается из последнего положительного остатка.

Машинный способ

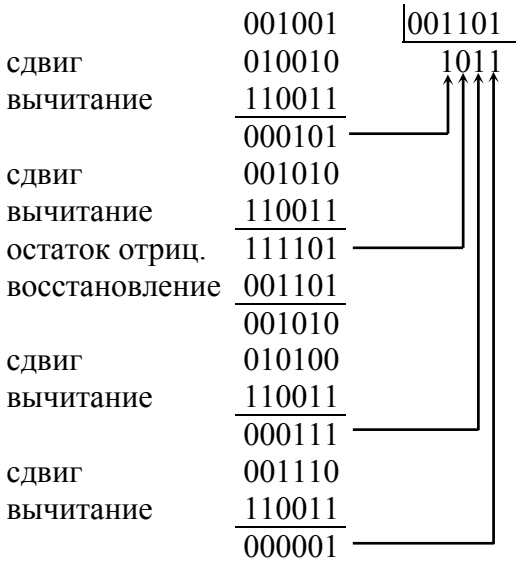
В машинах вместо сдвига делителя вправо осуществляется сдвиг остатка влево, что по сути ничего не изменяет. В случае получения отрицательного остатка в разряде частного записывается 0 и восстанавливается положительный предыдущий остаток путем прибавления к полученному отрицательному остатку делителя в прямом коде. Восстановленный остаток после этого сдвигается дополнительно на один разряд влево (аналогично дополнительному сдвигу делителя вправо при ручном счете).

Пример. (тот же)

Решаем в дополнительном коде.

Процесс деления начинается со сдвига делимого влево на один разряд, после чего из него вычитается делитель, и т.д.

В процессе деления переполнение разрядной сетки блокируется.



Весь процесс выполнения операции деления включает в себя три такта, последовательно повторяющихся до окончания операции:

1-й такт - сдвиг остатка влево

2-й такт - вычитание из сдвинутого остатка делителя (сложение в дополнительном или обратном коде)

3-й такт - запись «1» в соответствующий разряд частного, если остаток после вычитания получился положительным и «0» - если остаток получился отрицательным (в последнем случае в этом же такте производится восстановление остатка).

Пункты 5-8 первой главы выполняются по индивидуальным заданиям.

Результат: 001011

ГЛАВА 2. МИКРОПРОЦЕССОР КАК ОСНОВА ЭВМ

1. Внутренняя структура микропроцессора

Любая ЭВМ предназначена для обработки информации причем, как правило, осуществляет эту обработку опосредовано - представляя информацию в виде чисел. Для работы с числами машина имеет специальную важнейшую часть - ***микропроцессор.*** Это универсальное логическое устройство, которое оперирует с двоичными числами, осуществляя простейшие логические и математические операции, и не просто как придется, а в соответствии с программой, т.е. в заданной последовательности. Для хранения этой заданной последовательности служат запоминающие устройства - ЗУ. ЗУ бывают постоянными - ПЗУ, в которых информация хранится не изменяясь сколь угодно долго, и оперативными - ОЗУ, информация в которых может быть изменена в любой момент в соответствии с результатами ее обработки. Процессор общается с ОЗУ и ПЗУ через так называемое адресное пространство, в котором каждая ячейка памяти имеет свой адрес.

МП состоит из набора регистров памяти различного назначения, которые определенным образом связаны между собой и обрабатываются в соответствии с некоторой системой правил. ***Регистр*** - это устройство, предназначенное для хранения и обработки двоичного кода. К внутренним регистрам процессора относят: счетчик адреса команд, указатель стека, регистр состояний, регистры общего назначения.

Исследование счетчика команд было предложено фон Нейманом. Роль счетчика состоит в сохранении адреса очередной команды программы и автоматическом вычислении адреса следующей. Благодаря наличию программного счетчика в ЭВМ реализуется основной цикл исполнения последовательно расположенных команд программы.

Стек - это особый способ организации памяти, при использовании которого достаточно сохранять адрес последней заполненной ячейки ОЗУ. Именно адрес последней заполненной ячейки ОЗУ и хранится в указателе стека. Стек используется процессором для организации механизма прерываний, обработки обращения к подпрограммам, передачи параметров и временного хранения данных.

В регистре состояний хранятся сведения о текущих режимах работы процессора. Сюда же помещается информация о результатах выполняемых команд, например: равен ли результат нулю, отрицателен ли он, не возникли ли в ходе операции ошибки и т.п. Использование и анализ в этом регистре происходит побитно, каждый бит регистра имеет самостоятельное значение.

Регистры общего назначения (РОН) служат для хранения текущих обрабатываемых данных или их адреса в ОЗУ. У некоторых процессоров регистры функционально равнозначны, в других назначение регистров строго оговаривается. Информация из одного регистра может передаваться в другой.

2. Регистры

Процессоры семейства Intel 80x86 имеют 14 внутренних регистров, используемых для управления выполняющейся программой, для адресации памяти и для обеспечения арифметических вычислений. Каждый регистр имеет длину в одно слово (16 бит), адресуется в языке Assembler по имени и имеет строго определенное назначение:

Сегментные регистры: CS, DS, SS и ES. Каждый сегментный регистр обеспечивает адресацию памяти объемом до 64 Кбайт, которая называется *текущим сегментом*. Сегмент выровнен на границу параграфа и его адрес в сегментном регистре предполагает наличие справа четырех нулевых битов.

Регистр CS. Регистр сегмента кода содержит начальный адрес сегмента кода. Этот адрес плюс значение

смещения в командном указателе IP определяет адрес команды, которая должна быть выбрана для выполнения. Для обычных программ нет необходимости делать ссылки на регистр CS.

Регистр DS. Регистр сегмента данных содержит начальный адрес сегмента данных. Этот адрес плюс значение смещения, определенное в команде, указывают на конкретную ячейку в сегменте данных.

Регистр SS. Регистр сегмента стека содержит начальный адрес сегмента стека.

Регистр ES. Некоторые операции над строками используют дополнительный сегментный регистр для управления адресацией памяти. В данном контексте регистр ES связан с индексным регистром DI. Если необходимо использовать регистр ES, ассемблерная программа должна его инициализировать.

Регистры общего назначения: AX, BX, CX и DX. При программировании на языке Assembler регистры общего назначения являются "рабочими лошадками". Особенность этих регистров состоит в том, что возможна адресация их как одного целого слова или как однобайтовой части. Левый байт является старшей частью (high), а правый - младшей частью (low). Например, двухбайтовый регистр CX состоит из двух однобайтовых CH и CL, и ссылки на регистр возможны по любому из трех имен.

Регистр AX. Регистр AX является основным сумматором и применяется для всех операций ввода-вывода, некоторых операций над строками и некоторых арифметических операций.

Регистр BX. Регистр BX является базовым регистром. Это единственный регистр общего назначения, который может быть использован в качестве

"индекса" для расширенной адресации. Другое общее его применение - вычисления.

Регистр CX. Регистр CX является *счетчиком.* Он необходим для управления циклами и для операций сдвига. Регистр CX используется также и для вычислений.

Регистр DX. Регистр DX является регистром данных. Он применяется для операций ввода-вывода и некоторых операций умножения и деления над большими числами.

Регистровые указатели: SP и BP. Регистровые указатели SP и BP обеспечивают системе доступ к данным в сегменте стека. Реже они используются для операций сложения и вычитания.

Регистр SP. Указатель стека обеспечивает использование стека в памяти, позволяет временно хранить адреса и иногда данные. Этот регистр связан с сегментным регистром SS для адресации стека.

Регистр BP. Указатель базы облегчает доступ к данным (параметрам и адресам), переданным через стек.

Индексные регистры: SI и DI. Оба индексных регистра могут применяться для расширенной адресации и для использования в операциях сложения и вычитания.

Регистр SI. Этот регистр является индексом источника и применяется для некоторых операций над строками. В данном контексте регистр SI связан с регистром DS.

Регистр DI. Этот регистр является индексом назначения и применяется также для строковых операций. В данном контексте регистр DI связан с регистром ES.

Регистр командного указателя: IP. Регистр IP содержит смещение на команду, которая должна быть выполнена. Он

всегда связан с регистром CS. Обычно этот регистр в программе в явной форме не должен применяться.

Флаговый регистр. Девять из 16 бит флагового регистра являются активными и определяют текущее состояние машины и результаты выполнения команд. Многие арифметические операции и команды сравнения изменяют состояние флагов. Назначение флаговых битов следующее:

Флаг	*Назначение*
O (Переполнение)	Указывает на переполнение старшего бита при арифметических командах.
D (Направление)	Обозначает левое или правое направление пересылки или сравнения строковых данных.
I (Прерывание)	Указывает на возможность внешних прерываний.
T (Пошаговый режим)	Обеспечивает возможность работы процессора в пошаговом режиме.
S (Знак)	Содержит результирующий знак при арифметических операциях (0 - плюс, 1 - минус).
Z (Ноль)	Показывает результат арифметических операций и операций сравнения (0 - ненулевой, 1 - нулевой результат).
A (Внешний перенос)	Содержит перенос из 3-го бита для 8-битовых данных, используется для специальных арифметических операций.
P (Контроль четности)	Показывает четность младших 8-битовых данных (1 - четное, 0 - нечетное число).
C (Перенос)	Содержит перенос из старшего бита после арифметических операций, а

также последний бит при сдвигах или циклических сдвигах.

В программах флаговый регистр явно не используется, поэтому не имеется его мнемонического обозначения.

3. Система команд микропроцессора

Несмотря на бурную эволюцию вычислительной техники, основной набор команд довольно слабо изменился. Система команд любой ЭВМ обязательно содержит следующие группы команд обработки информации.

Команды передачи данных (перепись), копирующие информацию из одного места в другое.

Арифметические операции, к которым в основном относят операции сложения и вычитания. Умножение и деление обычно реализуется с помощью специальных программ.

Логические операции, позволяющие компьютеру производить анализ получаемой информации. Простейшими примерами команд рассматриваемой группы могут служить сравнение, а также известные логические операции ***и, или, не***.

Сдвиги двоичного кода влево и вправо. В некоторых случаях сдвиги используются для реализации умножения и деления.

Команды ввода и вывода информации для обмена с внешними устройствами. В некоторых ЭВМ внешние устройства являются специальными служебными адресами памяти, поэтому ввод и вывод осуществляется с помощью команд переписи.

Команды управления, реализующие нелинейные алгоритмы. Сюда относят условный и безусловный переходы, а также команды обращения к подпрограмме (переход с возвратом). Часто к этой группе относят операции по управлению процессором типа останов или нет операции.

Любая команда ЭВМ обычно состоит из двух частей - операционной и адресной. Операционная часть называемая также

кодом операции указывает, какое действие необходимо выполнить с информацией. Операционная часть имеется у любой команды. Адресная часть описывает, где используемая информация хранится и куда поместить результат. В некоторых командах управления работой машины адресная часть может отсутствовать, например, в команде останова.

Код операции можно представить себе как некоторый условный номер в общем списке команд. В основном этот список построен в соответствии с определенными внутренними закономерностями.

Адресная часть обладает значительно большим разнообразием. Основу адресной части составляет операнд. В зависимости от количества возможных операндов команды могут быть одно- и двухадресные. В двухадресных командах результат записывается либо в специальный регистр (сумматор), либо вместо одного из операндов.

4. Способы адресации данных

Способы (или методы) адресации не что иное, как способы указания на те или иные ячейки памяти, с которыми должен манипулировать оператор. Существует много различных методов адресации. Количество их зависит от типа процессора. Наличие большого количества способов адресации обеспечивает высокую гибкость в построении программ и является большим преимуществом системы команд данного типа ЭВМ. Способы адресации практически одинаковы для всех команд, в которых присутствуют операнды. Рассмотрим только три основных метода адресации, которые применяются почти во всех процессорах.

Регистровая адресация. При этом способе операндом является один из регистров общего назначения. Число хранится непосредственно в регистре. Записывается как Rn, где n -номер регистра.

Косвенная адресация. При этом способе адресации в одном из регистров общего назначения содержится не само число, с которым нужно работать, а его адрес, то есть номер ячейки памяти, в котором число находится. Записывается как (Rn), где n - номер регистра.

Автоинкрементная адресация. Этот вид адресации несколько сложнее двух предыдущих. Помимо основного действия (косвенного обращения к ячейке памяти), при использовании этого метода, происходит еще изменение адреса этого обращения. В данном случае увеличивается указатель адреса ячейки памяти, к которой мы обращаемся, то есть содержимое регистра, служащего указателем адреса. Данное увеличение происходит автоматически, без какой-либо команды. Записывается эта адресация как (Rn)+. То, что знак + стоит после имени регистра, намекает на порядок выполнения команды: сначала происходит операция с ячейкой, на которую указывает адрес помещенный в регистр Rn, а потом уже содержимое регистра увеличивается на 2 (если оператор работает со словом, то переход к адресу следующего слова), или на 1 (если оператор работает с байтом, переход к адресу следующего байта). Данный способ адресации применяется для работы с массивами и при использовании стека (например, при использовании подпрограмм).

Существует еще один особый способ адресации, который рассматривается отдельно. Речь идет о работе со стеком. *Стек* - неявный способ адресации данных, при котором информация записывается и считывается только последовательным образом с использованием указателя стека. Стек всегда имеет единственный вход и выход информации - для хранения его адреса и нужен указатель стека. При записи данных в стек процессор проделает следующее:

- уменьшит указатель стека на 2 (целое число занимает в памяти 2 байта);
- запишет данные по полученному адресу.

При извлечении данных из стека процессор проделает следующее:

- считает данные из стека;
- увеличит указатель на 2.

В командах работы со стеком адрес ОЗУ не фигурирует в явном виде. Но при этом молчаливо предполагается, что указатель стека уже задан. При задании указателя надо быть внимательным. Если указатель стека определен неправильно, то запись в стек может разрушить полезную информацию в ОЗУ.

Таким образом, мы рассмотрели способы адресации информации, которые существуют почти во всех типах процессора и которые использованы в имитаторе.

5. Форматы команд микропроцессора

Каждая команда вводится в свою ячейку, имеющую адрес. Размер адресного пространства редактора составляет 1Кб. Начальная ячейка имеет адрес 1000, конечная - 2024. В качестве операнда в одно- и двухадресных командах выступает один из регистров общего назначения. Выделим следующую систему команд.

Одноадресные команды.

Представлены в следующей форме: *Операция операнд ОП1.*

Очистить ОП1- обнуляет значение операнда.

Увеличить на 1 ОП1 - увеличивает значение операнда.

Уменьшить на 1 ОП1 - уменьшает значение операнда.

Двухадресные команды.

Представлены в следующей форме: *Операция первый операнд ОП1, второй операнд ОП2.*

Переслать ОП1 в ОП2 - пересылает значение первого операнда во второй операнд.

Добавить ОП1 к ОП2- добавляет значение первого операнда ко второму операнду, результат во втором операнде.

Вычесть ОП1 из ОП2 - вычитает значение первого операнда из второго операнда, результат во втором операнде.

Сравнить ОП1 с ОП2 - сравнивает разность второго и первого операнда с нулем, значения операндов не меняются, результат влияет на состояние регистра состояний.

Безадресные команды.

Возврат из подпрограммы - осуществляет возврат из подпрограммы в ячейку, следующую за командой, вызвавшей эту подпрограмму. Используется только в подпрограммах

Стоп - команда останова, ставится обязательно в конце программы, после ее выполнения никакие команды не выполняются.

Команды перехода.

Переход на К слов - безусловный переход - осуществляет переход на К слов.

Вызов подпрограммы по адресу - переход на адрес К с запоминанием адреса возврата для команды возврата из подпрограммы.

Если < > переход на К слов - переход на К слов, если результат < >0.

Если = переход на К слов - переход на К слов,если результат =0.

Если >= переход на К слов - переход на К слов, если результат >=0.

Если > переход на К слов - переход на К слов, если результат >0.

Если < переход на К слов - переход на К слов, если результат <0.

Если <= переход на К слов - переход на К слов, если результат <=0.

Работа команды условного перехода осуществляется следующим образом: если анализируемое условие справедливо, то переход происходит. В противном случае никаких действий не производится, а значит, переход игнорируется и процессор, как обычно, выбирает следующую команду. Справедливость анализируемого условия определяется по состоянию регистра

состояний, которое в свою очередь зависит от результата команды сравнения.

ГЛАВА 3. ФУНКЦИИ, РЕАЛИЗУЕМЫЕ АЛУ

В состав различных серий микросхем, лежащих в основе МП входят стандартные узлы арифметико-логических устройств (АЛУ), например, К 155 ИПЗ, К 561 ИПЗ.

Эти ИМС предназначены для действий с двумя четырехразрядными двоичными словами A и B. Результат выполнения логических преобразований и арифметических действий поступает в виде четырехразрядного слова на выходы F.. Кроме того, имеются вход P_0 и выход P сигналов переноса, отдельный выход признака равенства данных A=B и выходы переменных G0 и G1, используемые для наращивания разрядности АЛУ при объединении нескольких микросхем. Для облегчения наращивания разрядности оба сигнала переноса P_0 и P инверсные по отношению к входным сиг-

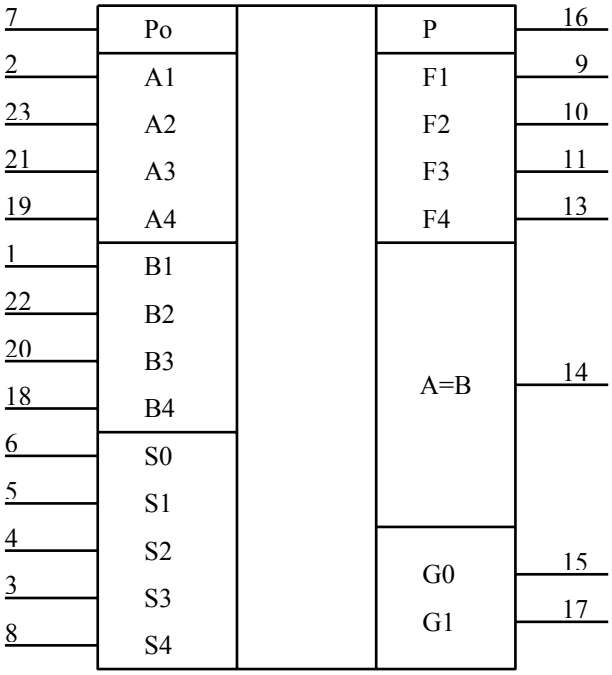


Рис.2.1.Условнографическое изображение ИМС

налам A и B, т.е. когда A и B задаются в положительной логике, сигналу переноса отвечает низкий уровень напряжения.

В зависимости от набора управляющих сигналов S микросхема выполняет одну из логических (при S4=1) или арифметико-логических операций (при S4=0). При этом логические операции выполняются поразрядно над каждой парой одноименных разрядов входных слов (входы и выходы переноса отключаются), а арифметические - над четырехразрядными словами с учетом сигнала.

Таблица 1

S3	S2	S1	S0	Логические операции S4=1	Арифметико-логические операции S4=0
0	0	0	0	$F=\overline{A}$	$F=A+\overline{P}_0$
0	0	0	1	$F=\overline{A}$ V B	$F=AVB+\overline{P}_0$
0	0	1	0	$F=\overline{A}$ * B	$F=AV\overline{B}+\overline{P}_0$
0	0	1	1	F=0	$F=-1+\overline{P}_0$
0	1	0	0	$F=\overline{A*B}$	$F=A+A*B+\overline{P}_0$
0	1	0	1	$F=\overline{B}$	$F=(AVB)+(A*\overline{B})+\overline{P}_0$
0	1	1	0	$F=A\oplus B$	$F=A-B-1+\overline{P}_0$
0	1	1	1	$F=A*\overline{B}$	$F=A*\overline{B}-1+\overline{P}_0$
1	0	0	0	$F=\overline{A}$ V B	$F=A+A*B+\overline{P}_0$
1	0	0	1	$F=\overline{A\oplus B}$	$F=A+B+\overline{P}_0$
1	0	1	0	F=B	$F=(AV\overline{B})+A*B+\overline{P}_0$

1	0	1	1	F=A*B	$F=A*B-1+\overline{P_0}$
1	1	0	0	F=1	$F=A+A+\overline{P_0}$
1	1	0	1	$F=A\vee\overline{B}$	$F=(A\vee B)+A+\overline{P_0}$
1	1	1	0	F=AVB	$F=(A\vee\overline{B})+\overline{P_0}+A$
1	1	1	1	F=A	$F=A-1+\overline{P_0}$

Обозначения:

V - логическое сложение:

* - логическое умножение;

$\oplus$ - сложение по модулю 2

\- арифметическое вычитание

\+ арифметическое сложение

F=A+A - сдвиг влево на один разряд.

При S4=0 АЛУ настраивается на выполнение арифметико-логических операций. Порядок действий при этом такой, что сначала производятся необходимые логические преобразования над входными словами (при этом переносы не учитываются), а затем арифметические действия сложения и вычитания с учетом переноса P_0.

Операция сложения выполняется АЛУ, настроенным управляющими сигналами S4,...,S0 на работу в качестве сумматора. Для того, чтобы не усложнять конструкцию АЛУ, операцию вычитания (для которой был бы нужен специальный вычитатель) заменяют сложением (выполняемым сумматором) уменьшаемого с вычитаемым, представленным в специальном коде.

Рассмотрим это действие подробно.

A	10111	23
B	1111	14
F	1001	9

Вычитание двоичных чисел, записанных в прямом коде подобно вычитанию в десятичной системе:

Стрелками показана операция "Заем", производимая для тех разрядов, в которых вычитаемое больше уменьшаемого. В десятичной системе занимаемая единица старшего разряда равна десяти единицам соседнего младшего разряда, а в двоичной двум единицам младшего разряда.

Для замены операции вычитания операцией сложения приходится представлять вычитаемое В в дополнительном коде. Дополнительный код образуется из обратного (инверсного) кода добавлением к нему единицы. Так четырехразрядное вычитаемое В, представленное в прямом коде Впр=В4 В3 В2 В1, может быть представлено и в обратном коде Вобр= $\overline{B4B3B2B1}$,и в дополнительном коде Вдоп= Вобр + 1.

Очевидно, для четырехразрядных чисел, записанных в этих кодах,справедливы равенства:

Впр + Вобр =1111

Впр+Вдоп =Впр+Вобр +1=1111+1= 10000

Впр = 10000 - Вдоп = 10000 - Вобр - 1

Следовательно, операцию вычитания можно представить в виде

Апр - Впр = Апр + В доп - 10000.

Таким образом, в АЛУ при выполнении операции вычитания входной операнд В преобразуется в дополнительный код, а вычитание числа 10000 производится без помощи специальных схем, только с использованием сигнала переноса в старший пятый разряд. Правда, при этом результат арифметических действий на выходе АЛУ будет также представлен в обратном коде.

ГЛАВА 4. БИС МПК СЕРИИ 580

БИС КР 580ВВ 51

Эта БИС предназначена для организации последовательного обмена информацией между МПС и внешним устройством (последовательный интерфейс). Последовательный обмен необходим в тех случаях, когда нет возможности соединить МПС и периферийные устройства многопроводной (параллельной) линией связи, например,при большом удалении внешнего устройства от МПС.

Перед началом работы в БИС записывается управляющее слово, определяющее формат передаваемого слова (5...8 бит), режим и скорость передачи информации (от 9,6 до 56 К бит/с).

Типичным примером использования этой БИС является её применение для передачи данных по телефонной линии связи.

Состав и функциональная схема БИС КР 580 ВВ 51 аналогичны составу и схеме БИС КР 580 ВВ 55 параллельного интерфейса (рис. 5.1, рис. 5.2).

БИС КР 580 ВВ 55

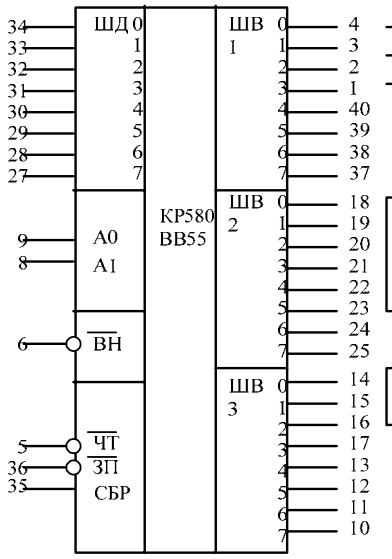


Рис. 4.1. Состав БИС

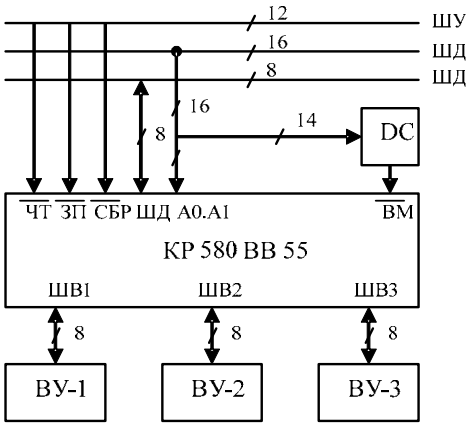


Рис. 4.2. Функциональная схема БИС КР 580 ВВ 55

В состав БИС входят четыре восьмиразрядных буферных регистра (один из которых связан с шиной данных МПС, а три других - с соответствующими периферийными устройствами) и устройство управления (УУ), связанное с управляющей и адресными шинами МП системы.

Управляющие и адресные сигналы задают режимы работы БИС и определяют внешнее устройство, производящее обмен информацией:

ВМ - "выбор микросхемы", разрешающий работу БИС при ВМ=0. При ВМ=1 выводы шин находятся в отключенном состоянии.

АО, А1 - двухразрядный адрес, позволяющий обращаться к одному из трех внешних устройств или регистру устройства управления

ЧТ - "чтение". При ЧТ-0 происходит передача данных от выбранного внешнего устройства на ШД МП системы.

ЗП - "запись". При ЗП=0 данные (или слово управления) передаются с ШД на выбранные адресом внешнее устройство или регистр устройства управления.

СБР - "Сброс" . При СБР=1 все регистры, включая регистр УУ,переводятся в нулевое состояние, а внешние каналы - в режим ввода.

Перед началом работы в регистр устройства управления записывается управляющее слово (производится программирование режима работы), которым определяется режим работы каждого внешнего канала: асинхронный двунаправленный, стробируемый двунаправленный и стробируемый однонаправленный.

БИС КР 580 ВИ 53

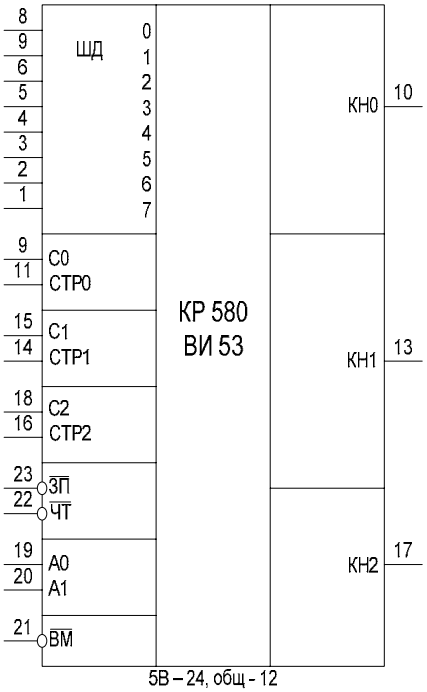


Рис. 4.3. Состав БИС

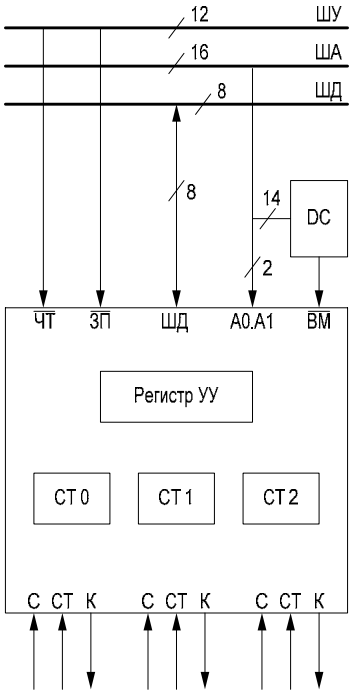


Рис. 4.4. Функциональная Схема

БИС КР 580 ВИ 53 служит для выработки временных интервалов программируемой длительности, может использоваться как счетчик внешних тактовых импульсов, программируемый делитель частоты, двоичный умножитель. В состав БИС входят три независимых 16 - разрядных счетчика (СТ0, СТ1, СТ2) работающих на вычитание в двоичном или двоично-десятичном коде (рис.5.3, рис.5.4). Прием счетных импульсов С1, С2 и С3 разрешается соответствующими стробирующими сигналами СТР0, СТР1, СТР2. Начальные значения счетчиков загружаются с шины данных.

После начала счета при достижении содержимым счетчика нуля вырабатываются сигнал "конец" - КН. В процессе работы содержимое любого счетчика может быть считано на шину данных. Назначение управляющих сигналов то же,что и для БИС КР 580 ВВ 55.

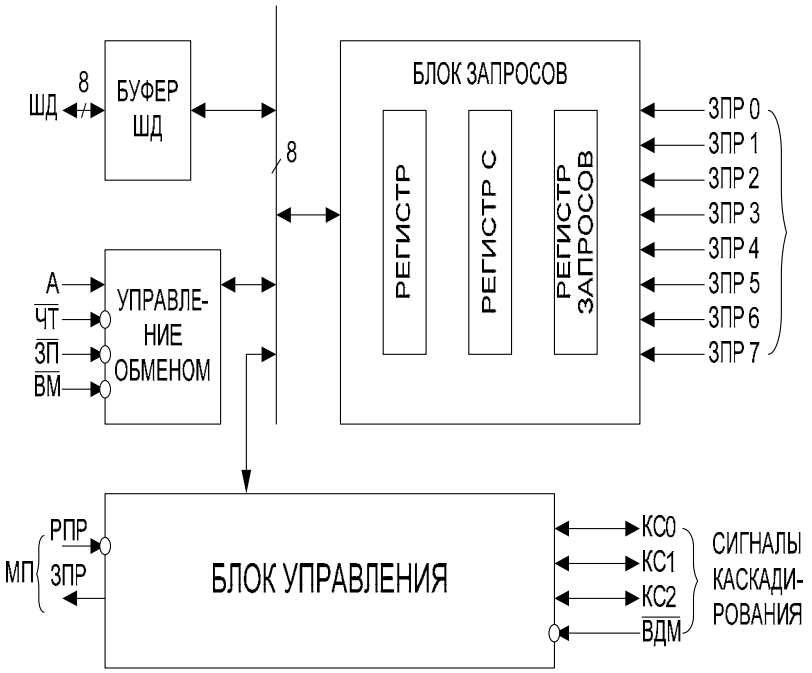
При ВМ=1 выводы шины данных находятся в третьем состоянии БИС,не принимает никаких управляющих сигналов, но работа всех СТ не останавливается. Занесение начального значения и считывание информации из СТ производится по байтам.

БИС КР 580 ВН 59

Эта БИС является восьмиуровневым контроллером приоритетных прерываний. Объединением 8 таких БИС можно создать устройство, реагирующее на запросы прерывания от 64 периферийных объектов. Объединение (каскадирование) производится при помощи трехразряной шины каскадирования КС0, КС1, КС2 и управляющего сигнала ВДМ (ведомый).

При ВДМ= 1 БИС становится ведущей и выходными сигналами КС0...КС2 задает порядковый номер (приоритет) остальным БИС.

При ВДМ=0 сигналы КС0... КС2 являются входными, определяющими адрес ведомой БИС. Назначение сигналов ЧТ, ЗП, ВМ такое же, как и у предыдущих БИС. Сигнал А – адрес регистра, в которой с ШД записывается управляющее слово, определяющее режим работы контроллера (рис. 5.5).



БИС КР 580 ВТ57

Эта БИС предназначена для организации в МПС обмена данными между ЗУ и периферийными устройствами в обход процессора. Контроллер прямого доступа к памяти (ПДП) имеет четыре независимых параллельных канала обмена с различными приоритетами. БИС ПДП в процессе обмена формирует адресное сопровождение ЗУ и сигналы управления обменом. Эта микросхема используется для подключения к МПС быстродействующих УВВ, например, дисковых накопителей данных. Так как в процессе работы канала ПДП процессор отключен от ЗУ, то обычно говорят, что это режим захвата памяти и на МП подается соответствующий сигнал.

ГЛАВА 5. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МП

Основными характеристиками МП являются:

1. ***Длина машинного слова*** (т.е. число битов, обрабатываемых МП в один прием). Обычно она (длина) совпадает с длиной регистров МП. Наибольшее распространение получили 8, 16 и 32 битные МП.

2. ***Количество регистров МП*** . Большинство МП имеет несколько (6) РОН И один РАК. Длина этих регистров совпадает с длиной машинного слова. Некоторые регистры МП (СК, РА, УС, Индексный регистр ИР) бывают длиннее машинного слова в 2 раза. Поэтому в состав команд МП вводятся специальные команды над словами двойной длины. Многие МП имеют РОН, адресуемые не только в одиночку , но и парами (образуя регистр двойной длины), некоторые регистры адреса программно доступны, что позволяет выполнять операции и над адресами.

3. ***Набор выполняемых команд***. В МП обычно используются одноадресные команды, и только для выполнения некоторых операций передачи данных используются двухадресные. Многочисленные (около 200) команды МП обычно разделяют на четыре группы:

а) команды преобразования данных,выполняющие обработку данных и включающие в себя ограниченный набор операций арифметических (сдвига, сравнения) и специальных операций (например, операция уменьшения выполняется через сдвиги и сложение).

б) команды передачи данных пересылают данные из одной части ЭВМ (МП) в другую без изменения передаваемых и включают в себя команды обращения к памяти внутрипроцессорного обмена (регистр-регистр),

команды операции со стеком и операции «ввода-вывода» (обмен с внешними устройствами)

в) команды управления программой изменяют содержимое СК (вызывающее нарушение естественного порядка выполнения команд) и включают в себя команды безусловного и условного перехода операции с подпрограммами, остановку и команду отсутствия операции.

г) команды управления состоянием изменяют значение признаков ЭВМ (МП) без изменения данных и порядка выполнения команд; в регистре признаков МП обычно регистрируются следующие признаки:

S-знак, S=1,ссли результат операции есть отрицательное число.

Z-нуль, Z=1,если результат операции равен 0.

С - перенос, С= 1 ,если формируется единица переноса и старшего разряда (бита) байта.

Р - четность. Р=1.если число единиц в коде результата четно.

4. ***Быстродействие*** выражается количеством операций выполняемых за 1 сек. (иногда выражается длительностью цикла выполнения команды или длительностью такта цикла или же тактовой частотой). Последние два показателя обычно равны соответствующим показателям главной памяти.

5. ***Обьём адресного пространства*** - это диапазон значений адресов главной памяти, которую способен адресовать МП (обычно он превышает объем подключений к МП у главной памяти).

6. ***Количество портов ввода- вывода*** (регистров – контролеров внешних устройств ВУ), их пропускная

способность и механизм прерывания.

7. ***Способы адресации памяти и портов.*** Адресация портов в виду их малочисленности в сравнении с объёмом главной памяти - прямая с помощью одного байта.

Для адресации памяти чаще всего в МП используются следующие виды адресации:

а) непосредственная - когда в команде задается значение операнда; признаком непосредственной адресации является наличие в мнемоническом коде операции в качестве последней буквы символа I, указывающего , что далее в команде содержится операнд или же наличие после мнемокода знака # , указывающего, что величина, перед которой он поставлен, является операндом; непосредственная адресация используется для начальной загрузки счетчиков и адресных регистров.

б) прямая - когда в команде задаётся адрес ячейки главной памяти, содержащей операнд.

в) косвенная регистровая - когда в команде задастся номер регистра МП, содержащего адрес операнда (т.е. номер ячейки , содержащей операнд); признаком косвенной адресации обычно является символ @ , записываемый перед номером регистра.

г) прямая регистровая - когда в команде задаётся номер (обычно буквы В Д и т.д.) регистра МП, содержащего операнд.

д) стековая - когда адрес операнда как бы содержится в самом коде операции ("записывать в стек "или" прочитать из стека"), т.е. адресом всегда является текущее значение содержимого специального регистра указателя стека УС (SP).

Приложение А
Варианты задания по теме
«Функции, реализуемые АЛУ»

Вариант 1. A=0001, B=1000, P0=0

S4	S3	S2	S1	S0
1	1	1	0	0
1	0	0	1	1
0	1	0	0	1
0	0	1	1	0

Вариант 2. A=1010, B=0010, P0=1

S4	S3	S2	S1	S0
1	1	0	0	1
1	0	1	1	0
0	1	1	0	0
0	0	1	1	0

Вариант 3. A=0100, B=0111, P0=1

S4	S3	S2	S1	S0
1	1	1	1	0
1	0	0	1	0
0	1	0	0	0
0	0	1	0	0

Вариант 4. A=1010, B=0001, P0=0

S4	S3	S2	S1	S0
1	1	0	0	0
1	0	1	1	1
0	1	1	1	0
0	0	1	0	1

Вариант 5. A=0101, B=1001, P0=0

S4	S3	S2	S1	S0
1	1	1	1	1
1	0	1	0	0
0	1	1	0	1
0	0	0	1	1

Вариант 6. A=0010, B=1011, P0=0

S4	S3	S2	S1	S0
1	1	0	1	1
1	1	1	1	0
0	0	1	0	1
0	0	1	1	1

Вариант 7. A=1110, B=0001, P0=1

S4	S3	S2	S1	S0
1	0	1	1	0
1	1	1	0	1
0	0	0	0	0
0	0	1	0	0

Вариант 8. A=0110, B=1000, P0=1

S4	S3	S2	S1	S0
1	1	0	1	1
1	1	0	1	0
0	0	1	0	0
0	0	0	0	1

Вариант 9. A=0110, B=1101, P0=1

S4	S3	S2	S1	S0
1	1	0	0	1
1	0	1	0	0
0	1	1	0	0
0	1	0	1	0

Вариант 10. A=1000, B=0010, P0=1

S4	S3	S2	S1	S0
1	0	0	0	1
1	1	0	0	0
0	1	1	0	0
0	1	0	1	0

Вариант 11. A=1100, B=0011, P0=1

S4	S3	S2	S1	S0
1	1	1	0	0
1	0	0	1	1
0	1	1	0	0
0	0	0	1	1

Вариант 12. A=0001, B=1011, P0=0

S4	S3	S2	S1	S0
1	1	1	1	0
1	1	0	1	0
0	1	0	1	1
0	0	1	1	0

Приложение Б
Варианты задания по теме
«Арифметические основы ЭВМ»

1. Сложить в обратном коде:

№ вар.	1		2		3	
	X	Y	X	Y	X	Y
1	0,010001	-0,101000	0,100101	-0,000111	-0,10011	-0,00010
2	0,001010	-0,011001	0,100101	-0,001111	-0,110011	-0,001101
3	0,010001	-0,101010	0,100100	-0,000111	-0,100110	-0,000111
4	0,001101	-0,010101	0,01110	-0,00111	-0,10110	-0,00110
5	0,000101	-0,101100	0,100111	-0,000101	-0,101101	-0,001011
6	0,000101	-0,100111	0,010001	-0,001110	-0,11011	-0,011001
7	0,001010	-0,110010	0,100110	-0,000111	-0,101111	-0,001010
8	0,010111	-0,100010	0,0011101	-0,0001110	-0,01101	-0,10010
9	0,0101001	-0,1001110	0,100101	-0,011101	-0,101011	-0,000110
10	0,00111	-0,01001	0,01101	-0,00111	-0,10111	-0,00101

2. Сложить в дополнительном и модифицированном кодах:

№ вар.	X	Y
1	0,101101	-0,010010
2	0,101001	-0,001010
3	0,1010001	-0,0101110
4	0,110011	-0,010110
5	0,1010001	-0,0101101
6	0,1011101	-0,0100011
7	0,1101101	-0,1101110
8	0,101011	-0,010010
9	0,11001	-0,001010
10	0,1001101	-0,0101010

3. Сложить:

№ вар.	X	Y
1	$+0,011010\cdot2^{100}$	$-0,110010\cdot2^{110}$
2	$+0,1101\cdot2^{100}$	$-0,1101\cdot2^{110}$
3	$+0,011110\cdot2^{100}$	$-0,100110\cdot2^{110}$
4	$+0,101011\cdot2^{100}$	$-0,011101\cdot2^{101}$
5	$+0,011001\cdot2^{100}$	$-0,100101\cdot2^{110}$
6	$+0,110001\cdot2^{100}$	$-0,010101\cdot2^{110}$
7	$+0,1110\cdot2^{101}$	$-0,1010\cdot2^{111}$
8	$+0,100101\cdot2^{100}$	$-0,0110111\cdot2^{110}$
9	$+0,1101\cdot2^{100}$	$-0,1101\cdot2^{111}$
10	$+0,10001\cdot2^{100}$	$-0,011111\cdot2^{110}$

4.Умножить:

№ вар.	X	Y
1	0,0111	0,0101
2	0,0011	0,0101
3	0,0100	0,0011
4	0,010011	0,011101
5	0,0011	0,1001
6	0,0011	0,0101
7	0,001101	0,011010
8	0,011001	0,01101
9	0,1011	0,0110
10	0,0010	0,0111

5. Разделить:

№ вар.	X	Y
1	0,1011001	0,1101100
2	0,100101	0,110111
3	0,10011011	0,11000111
4	0,100111	0,110101
5	0,1001101	0,1110010
6	0,10110011	0,11100011
7	0,010101	0,100101
8	0,1010111	0,1110010
9	0,11001	0,11011
10	0,1000101	0,1101001

Литература

1. *Королев, Л.Н.* Микропроцессоры, микро- и мини-ЭВМ / Л.Н. Королев. – Москва : Издательство МГУ, 1988. - 213 с.
2. *Шапурин, И.И.* Руководство по микроконтроллерам: пер. с англ. под. ред. И.И. Шапурина, С.Б. Лужанского. – М. : Постмаркет, 2001.
3. *Калабеков, Б.А.* Цифровые устройства и МПСистемы : учебник / Б.А. Калабеков. - М.: Темком, 2000. – 336 с.
4. *Арестов, К.А.* Основы электроники и микропроцессорной технике : учебник / К.А. Арестов. – М.: Колос, 2002.
5. *Орлов, И.А*. Основы вычислительной техники и организация вычислительных работ / И.А. Орлов. – М.: Энергия, 1971. – 272с.
6. *Кириличев, А.М.* Основы вычислительной техники : учебник / А.М. Кириличев. – М.: Недра, 1988. – 350 с.
7. *Ямпольский, В.С.* Основы автоматики и электронно-вычислительной техники : учебное пособие для студентов физ.-мат.фак. пед.ин-тов / В.С. Ямпольский. - М.: Просвещение. 1991. - 223 с.

Учебное издание

Сечина Галина Павловна

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНО-ПРАКТИЧЕСКИХ РАБОТ

Корректор Габдурахимова Т.М.
Худ.редактор Федорова Л.Г.

Сдано в набор 02.05.2012
Подписано в печать 06.06.2012.
Бумага писчая. Гарнитура Таймс.
Усл.печ.л. 3,75. Тираж 100.
Заказ №32.

НХТИ (филиал) ФГОУ ВПО «КНИТУ»,
г.Нижнекамск, 423570, ул.30 лет Победы, д.5а.

61