

ПЕРЕДАЧА ДАННЫХ 4

4.1 ОБЗОР

В данной главе описываются те элементы процессора, которые управляют обменом данных процессора с внешними устройствами, а также обменом данных между шинами в самом процессоре. Это генераторы адреса данных и устройство обмена данными между шиной данных памяти программы и шиной данных памяти данных - устройство обмена между шинами ДПП-ДПД.

4.2 ГЕНЕРАТОРЫ АДРЕСА ДАННЫХ

В каждом процессоре семейства ADSP-2100 содержится два самостоятельных генератора адреса данных, за счет которых обеспечивается одновременный доступ к памяти данных и памяти программы. Генераторы адреса данных позволяют осуществлять косвенную адресацию данных. Оба генератора могут выполнять автоматическую модификацию адреса. Благодаря циклическим буферам, может выполняться модификация адреса по модулю. Генераторы адреса данных отличаются друг от друга. Генератор адреса данных 1 (DAG1) генерирует только адреса памяти данных, но обеспечивает при этом дополнительную возможность постановки бит в обратном порядке (битреверсную адресацию). Генератор адреса данных 2 (DAG2) генерирует как адреса памяти данных, так и адреса памяти программы, но не способен поддерживать битреверсную адресацию.

При ознакомлении с приводимым ниже описанием внутреннего устройства и работы генераторов адреса данных следует иметь в виду, что в результате развития программного обеспечения процессоров семейства ADSP-2100 появился прямой способ объявления буферов циклическими или линейными и управления положением буфера в памяти. Специальное программирование требуется только для инициализации регистров генераторов адреса данных. См. разделы "Косвенная адресация" и "Адресация по модулю (циклические буферы)" ниже.

4.2.1 Регистры генератора адреса данных

Блок-схема одного генератора адреса данных приведена на рис. 4.1. Имеется три регистровых файла: файл регистров модификации (M), файл индексных регистров (I) и файл регистров длины (L). Каждый из регистровых файлов состоит из четырех регистров разрядностью 14 бит, считывание и запись данных с которых осуществляется через шину ДПД.

Индексные регистры (I0-3 в DAG1, I4-7 в DAG2) содержат действительные адреса, используемые для доступа к памяти. При косвенной адресации

4 ПЕРЕДАЧА ДАННЫХ

данных адрес, находящийся в выбранном индексном регистре (I), становится адресом памяти. Биты адреса на выходе DAG1 могут быть переставлены в обратном порядке за счет установки соответствующего бита режима в регистре состояний режима (MSTAT), как показано ниже, или при помощи команды ENA BIT_REV. Битреверсная адресация (адресация с постановкой бит в обратном порядке) облегчает выполнение БПФ.

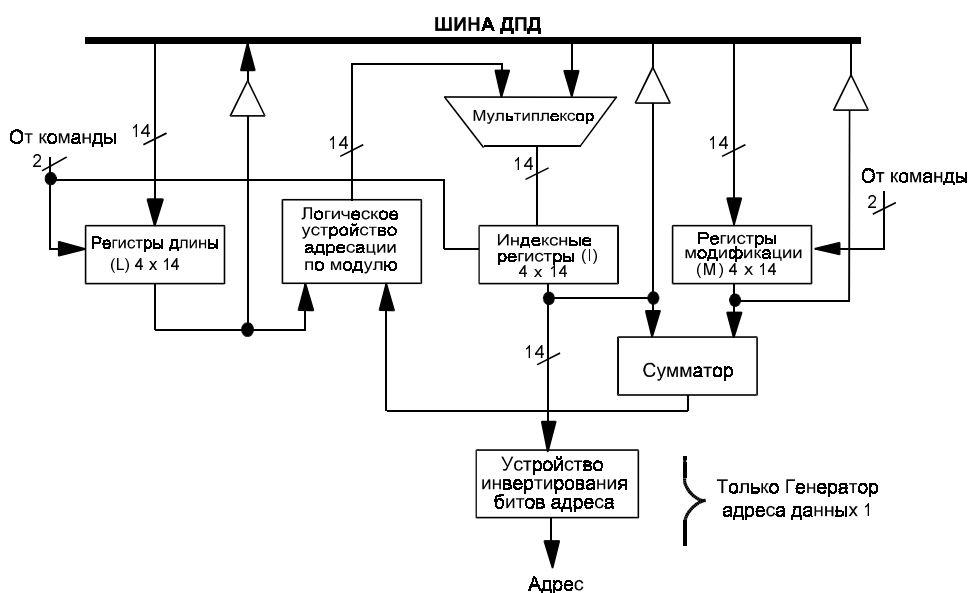


Рис. 4.1 Блок-схема генератора адреса данных

Генераторы адреса данных используют пост-модификацию; после доступа к данным при использовании косвенной адресации содержимое заданного регистра M (M0-3 в DAG1, M4-7 в DAG2) добавляется к содержимому заданного регистра I для генерирования обновленного значения регистра I. Выбор регистров M и I осуществляется независимо в каждом генераторе адреса данных. Другими словами, любой регистр из индексных регистров I0-3 может быть модифицирован содержимым любого из регистров модификации M0-3 со всеми возможными комбинациями, но не может быть модифицирован значениями из регистров DAG2 (M4-7). Модифицирующие значения, содержащиеся в регистрах модификации M,

ПЕРЕДАЧА ДАННЫХ 4

являются числами со знаком, так что следующий адрес может быть и больше и меньше предыдущего.

Генераторы адреса данных поддерживают как адресацию с линейным, так и с циклическим буфером. *Содержимое регистра L, соответствующего регистру I (например, регистр L0 будет соответствовать I0), определяет, какой режим адресации используется для этого индексного регистра I.* При адресации с циклическим буфером регистр L инициализируется длиной буфера. При адресации с линейным буфером логическое устройство адресации по модулю блокируется за счет установки 0 в соответствующем регистре L. Каждый раз, когда выбирается регистр I, соответствующий регистр L обеспечивает устройство адресации по модулю информацией о длине буфера. Если сумма значений регистра M и регистра I пересекает границу буфера, модифицированное значение регистра I вычисляется логическим устройством адресации по модулю с использованием значения регистра длины L.

Все регистры генератора адреса данных (I, M, L) могут загружаться и считываться с младших 14 бит шины ДПД. Так как содержимым регистров I и L считаются числа без знака, два старших бита шины ДПД заполняются при считывании этих регистров нулями. Содержимым регистра M являются числа со знаком, поэтому старшие два бита шины ДПД дополняются по знаку.

4.2.2 Косвенная адресация

В процессорах семейства ADSP-2100 предусмотрено два режима адресации для памяти данных: прямая и косвенная. Косвенная адресация осуществляется за счет загрузки адреса в индексный регистр (I) и выбора одного из доступных регистров модификации (M).

Регистры длины (L) обеспечивают более легкую адресацию с циклическим возвратом указателя циклического буфера. Создание циклического буфера становится возможным только, когда регистр L не равен нулю. Для линейной (т.е. нециклической) косвенной адресации регистр L, соответствующий используемому регистру I, должен быть равен нулю.

Не следует полагать, что правильная инициализация регистров длины L производится автоматически; после перезапуска процессора в регистрах I, M и L содержатся случайные значения. Ваша программа должна предусматривать инициализацию регистров L, соответствующих всем используемым регистрам I.

4 ПЕРЕДАЧА ДАННЫХ

4.2.2.1 Инициализация регистров длины L для осуществления нециклической адресации

Запись в регистр длины L любого значения, не равного нулю, активирует логическое устройство адресации по модулю. Для осуществления линейной косвенной адресации, регистр L должен содержать значение 0, чтобы заблокировать логическое устройство адресации по модулю.

Ниже приводится простой пример задания линейной косвенной адресации:

```
I3 = 0 x 3 8 0 0 ;
M2 = 0 ;
L3 = 0 ;
AX0 = DM ( I3 , M2 ) ;
```

В следующем примере переменная памяти используется для хранения указателя адреса:

```
.VAR/DM/RAM  addr_ptr;    { переменная содержит адрес }
I3=DM(addr_ptr);          { для загрузки регистра I3 }
                                { используется          прямая
                                ад { ресация }

L3=0;                     { блокирование  циклической }
                                { адресации }

M1=0;                     { значение регистра I3 не }
                                { подвергается  пост-моди- }
                                { фикации }

AX0=DM(I3,M1);            { для загрузки регистра AX0 }
                                { используется  косвенная  }
                                { адресации }
```

4.2.3 Адресация по модулю (циклические буферы)

Логическое устройство адресации по модулю выполняет автоматическую адресацию по модулю для достижения доступа к содержимому циклических буферов. Для вычисления следующего адреса логическое устройство адресации по модулю использует:

- Текущий адрес ячейки памяти, находящийся в индексном регистре I (без знака).
- Модифицирующее значение, находящееся в регистре модификации M (со знаком).
- Длину буфера, значение которой находится в регистре длины L (без знака).

ПЕРЕДАЧА ДАННЫХ 4

- Базовый адрес буфера.

Используя эту поступающую в устройство информацию, следующий адрес вычисляется по формуле:

$$\text{Следующий адрес} = (I + M - B) \text{По модулю}(L) + B$$

где:

- I = текущий адрес
- M = модифицирующее значение (со знаком)
- B = базовый адрес
- L = длина буфера
- M + 1 = модифицированный адрес

Используемые входные значения проверяются по условию:

$$|M| < L$$

Выполнение данного условия гарантирует, что не будет более одного циклического возврата за операцию.

4.2.4 Вычисление базового адреса

Базовый адрес циклического буфера длины L равен 2^n или кратен 2^n , где n удовлетворяет условию:

$$2^{n-1} < L \leq 2^n$$

Другими словами, базовый адрес есть "округление" L к ближайшей степени двойки 2^n (или кратного ему числа). Данное правило подразумевает, что определенное число бит базового адреса должны быть нулями.

На практике вам не понадобится самим вычислять n; редактор связей автоматически помещает циклический буфер по правильному адресу.

4.2.4.1 Базовый адрес циклического буфера. Пример 1

Предположим, что длина циклического буфера равна восьми. Согласно правилу, длина буфера (восемь) должна быть меньше или равна значению 2^n ; следовательно n должно быть равно или больше трех. В левой части неравенства определяется, что длина буфера должна быть больше, чем 2^{n-1} ; следовательно n должно быть равно или меньше трех. Единственным значением, удовлетворяющим обоим частям неравенства, является три. Действительными являются базовые адреса кратные 2^3 ; таким образом в данном примере действительными базовыми адресами будут числа кратные восьми: 0x0008, 0x0010, 0x0018, и т. д.

4 ПЕРЕДАЧА ДАННЫХ

4.2.4.2 Базовый адрес циклического буфера. Пример 2

Для второго примера возьмем значение длины буфера равное семи. Оба правила дают одно и то же значение n , а именно три. Следовательно, при длине буфера равной семи действительными базовыми адресами для всех процессоров семейства ADSP-2100 будут числа кратные восьми: 0x0008, 0x0010, 0x0018 и т. д.

4.2.4.3 Работа циклического буфера. Пример 1

Предположим, что $I0 = 5$, $M0 = 1$, $L0 = 3$, а базовый адрес = 4. Следующий адрес вычисляется следующим образом:

$$(I0 + M0 - B) \text{ по модулю } (L0) + B = (5 + 1 - 4) \text{ по модулю } 3 + 4 = 6$$

Последовательные вычисления адреса при использовании $I0$ для косвенной адресации дают последовательность: 5, 6, 4, 5, 6, 4, 5... При $M0 = -1$ (0x3FFF) использование $I0$ для вычисления следующего адреса даст последовательность: 5, 4, 6, 5, 4, 6, 5, 4...

4.2.4.4 Работа циклического буфера. Пример 2

Предположим, что $I0 = 9$, $M0 = 3$, $L0 = 5$, базовый адрес = 8. Буфер, содержащий 5 слов, находится в ячейках с 8 по 12 включительно. Следующий адрес вычисляется следующим образом:

$$(I0 + M0 - B) \text{ по модулю } L0 + B = (9 + 3 - 8) \text{ по модулю } 5 + 8 = 12$$

Последовательные вычисления адреса при использовании $I0$ для косвенной адресации дают последовательность: 9, 12, 10, 8, 11, 9...

4.2.5 Битреверсная адресация

Логическое устройство инвертирования бит адреса предназначено, прежде всего, для использования в вычислениях БПФ, когда биты значений или вводятся или генерируются в обратном порядке. Постановка бит в обратном порядке (инвертирование) возможно только для адресов, сгенерированных DAG1. Точкой поворота является средняя точка 14-разрядного адреса, между битами 6 и 7, как показано на приводимой на следующей странице схеме:

ПЕРЕДАЧА ДАННЫХ 4

Отдельные адресные линии ($ADDR_N$)

Нормальный порядок бит	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Обратный порядок бит	00	01	02	03	04	05	06	07	08	09	10	11	12	13

Режим битреверсной адресации разрешается и блокируется при помощи установки соответствующего бита в регистре состояния режима (MSTAT). При разрешении данного режима биты всех адресов, сгенерированных с использованием индексных регистров I0-3, ставятся в обратном порядке на выходе. (Модифицирующие значения сохраняются в обычном порядке после обновления адреса). Этот режим сохраняется до сброса соответствующего бита состояния.

Значения адресов с разрядностью менее 14 бит также можно инвертировать. Следует определить первый адрес и записать при инициализации в регистр M значение, вычисляемое для модификации инвертированного выходного значения регистра I в нужном интервале. Это значение равно:

$$2^{(14-N)}$$

где N - число бит, которые вы хотите поставить в обратном порядке в выходном значении. Подробный пример такого типа разбирается в главе, посвященной БПФ, в первом томе издания *"Digital Signal Processing Applications Using the ADSP-2100 Family"*.

4.3. ПРОГРАММИРУЕМЫЙ ДОСТУП К ДАННЫМ

Программные средства отладки процессоров семейства ADSP-2100 поддерживают объявление и использование такой простой структуры данных, как одномерные массивы. Массив может содержать одно единственное значение (переменная) или несколько значений (массив). Кроме того, массив может использоваться как циклический буфер. Ниже приводится разъяснение каждого из названных случаев и даются примеры правильного объявления и использования данных. Полную информацию о синтаксисе этих и других директив ассемблера можно найти в руководстве *"ADSP-2100 Family Assembler Tools Manual"*.

4.3.1. Переменные и массивы

Массивы являются базовой структурой данных для процессоров семейства ADSP-2100. В литературе слово "массив" и выражение "буфер данных" используются как взаимозаменяемые понятия. Массивы всегда объявляются при помощи

4 ПЕРЕДАЧА ДАННЫХ

директив ассемблера, к ним можно обращаться косвенно или по имени, они могут инициализироваться непосредственными значениями, указанными в директивах или значениями, взятыми из внешних файлов данных, а также могут быть линейными или циклическими с автоматическим циклическим возвратом.

Массив объявляется при помощи следующей директивы:

```
.VAR/DM coefficients[128];
```

Эта директива объявляет массив из 128 значений разрядностью по 16 бит, расположенный в памяти данных (DM). Специальные операторы ^ и % являются ссылкой на адрес или длину массива, соответственно. Это может быть сделано следующим образом:

```
IO = ^ coefficients;           {указывает на адрес буфера}  
LO = 0;                       {установка регистра L в 0}  
MX0 = DM ( IO , M0 );         {загружает в регистр MX0 значение из  
                               буфера}
```

По этой команде в регистр MX0 загружается начальное значение буфера *coefficients*, находящееся в памяти данных. Используя автоматическую постмодификацию значений регистров генератора адреса данных, можно организовать цикл для выполнения второй из приведенных выше команд и так последовательно загрузить все значения из буфера.

В противоположном случае, когда нужно адресовать только первую ячейку массива, во многих обстоятельствах можно непосредственно использовать имя буфера в качестве метки, например:

```
MX0 = DM ( coefficients );
```

Редактор связей заменяет метку на действительный адрес. Кроме того, можно полностью инициализировать весь массив/буфер из файла данных, используя для этого директиву .INIT:

```
.INIT coefficients: <filename.dat>;
```

Эта директива производит считывание данных из указанного файла данных (*filename.dat*) в массив во время компоновки. Подобная возможность поддерживается только симулятором - данные не могут непосредственно загружаться во внутреннюю память на кристалле за счет аппаратных средств начальной загрузки.

Массив или буфер данных длиной в единицу является простой однословной переменной, которая должна объявляться соответствующим образом:

```
.VAR/DM coefficients;
```


ПЕРЕДАЧА ДАННЫХ 4

4.3.2 Циклические буферы

Для задач цифровой обработки сигналов часто требуется использовать циклические буферы. Функция организации таких буферов выполняется генераторами адреса данных процессора с использованием регистров длины (L). Сначала следует объявить циклический буфер:

```
.VAR/DM/CIRC coefficients[128];
```

Такое объявление указывает редактору связей, что этот буфер должен быть помещен в правильном интервале адресов. Далее следует инициализировать регистр L (длины), для чего обычно используется оператор ассемблера % (или постоянная), а также регистры I и M, как в нижеследующем примере:

```
L0 = % coefficients;      {длина циклического буфера}  
I0 = ^ coefficients;     {указание адреса буфера}  
  
M0 = 1;                  {увеличивается на 1 ячейку каждый раз}
```

А теперь оператор вида:

```
MX0 = DM(I0, M0);       {загрузка в регистр MX0 значения из буфера}
```

- помещенный в цикл, задает непрерывное считывание в цикле из буфера *coefficients* с автоматическим циклическим возвратом.

4.4 УСТРОЙСТВО ОБМЕНА МЕЖДУ ШИНАМИ ДПП-ДПД

Устройство обмена данными между шинами ДПП и ДПД, связывающее эти шины, позволяет осуществлять передачу данных между ними в обоих направлениях. Так как шина данных памяти программы (ДПП) имеет разрядность 24 бита, тогда как разрядность шины данных памяти данных (ДПД) составляет 16 бит, напрямую могут передаваться только 16 старших бит шины ДПП. Оставшиеся 8 бит загружаются (или берутся) из внутреннего регистра (RX). Этот регистр может напрямую загружаться или считываться с шин данных, когда требуется передача всех 24 бит.

Заметим, что при одновременном считывании данных из памяти программы и памяти данных, от старших 16 бит шины ДПП к регистрам Y вычислительных устройств назначена специальная магистраль. Эта магистраль предназначена только для считывания и не использует схему обмена между

4 ПЕРЕДАЧА ДАННЫХ

шинами ДПП-ДПД. Она показана на блок-схемах отдельных вычислительных устройств.

4.4.1 Блок-схема обмена ДПП-ДПД

На рис. 4.2 приводится блок-схема устройства обмена данными между шинами ДПП и ДПД. В ней обеспечивается два вида соединений.

Первое соединение представляет собой однонаправленную магистраль между шинами. Оно состоит из двух буферов с тремя состояниями, которые соединяют шину ДПД со старшими 16 битами шины ДПП. Обычно один из этих двух буферов используется при обмене данными между памятью программы и одним из регистров, подсоединенным к шине ДПД. Именно такая магистраль применяется для записи данных в память программы; она не показана на блок-схемах отдельных вычислительных устройств.

Второе соединение осуществляется через регистр РХ. Регистр РХ имеет разрядность 8 бит и загружается либо 8 младшими битами шины ДПД или 8 младшими битами шины ДПП. Содержимое этого регистра может также считываться на 8 младших бит каждой вышеназванной шины.

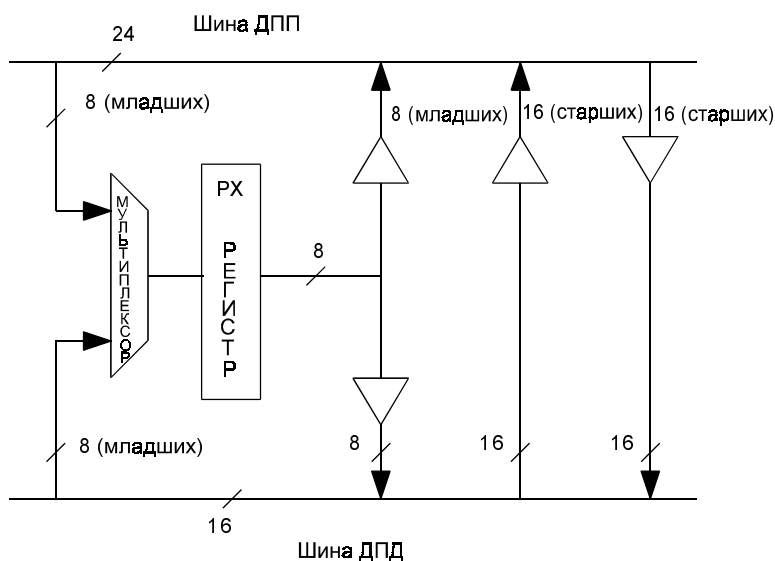


Рис. 4.2 Устройство обмена ДПП-ДПД

Обмен данными между регистром РХ и шиной ДПП осуществляется следующим образом:

ПЕРЕДАЧА ДАННЫХ 4

1. Регистр PX автоматически загружается с шины ДПП каждый раз, когда данные (не команда) считываются с памяти программы на любой регистр, например:

$$AX0 = PM(I4, M4);$$

В данном примере старшие 16 бит 24-разрядного слова памяти программы загружаются в регистр $AX0$, а младшие 8 бит автоматически загружаются в регистр PX .

2. Содержимое регистра PX автоматически считывается как 8 младших бит при записи данных в память программы, например:

$$PM(I4, M4) = AX0;$$

В данном примере 16 бит регистра $AX0$ записываются как старшие 16 бит 24-разрядного слова памяти программы, а 8 бит, содержащиеся в регистре PX автоматически записываются как 8 младших бит этого слова.

При доступе к регистру PX с шины ДПД:

1. Регистр PX может загружаться при помощи команды передачи данных, в которой PX в явной форме задан как регистр назначения. При этом используются младшие 8 бит, тогда как старшие 8 бит данных отбрасываются.

$$PX = AX0;$$

2. Содержимое регистра PX может считываться при помощи команды передачи данных, в которой PX в явной форме задан как исходный регистр. Старшие 8 бит, считываемые с регистра, записываются нулями.

$$AX0 = PX;$$

Всегда, когда содержимое любого регистра записывается в память программы, из исходного регистра берутся 16 старших бит, а содержимое регистра PX автоматически добавляется в качестве младших 8 бит. В случае необходимости сохранения младших 8 бит данных, передаваемых в память программы через шину ДПП, следует прежде загрузить регистр PX с шины ДПД, а затем начинать операцию считывания в память программы.

4 ПЕРЕДАЧА ДАННЫХ