

Обзор главы

В разделе	Вы найдете	на стр.
9.1	Непосредственная адресация	9–2
9.2	Прямая адресация	9–2
9.3	Косвенная адресация через память	9–3
9.4	Адресные регистры	9–6
9.5	Косвенная регистровая внутризонная адресация	9–7
9.6	Косвенная регистровая адресация с указанием в регистре области памяти	9–10

9.1. Непосредственная адресация

Описание

При непосредственной адресации операнда значение операнда кодируется непосредственно в операции, то есть непосредственно следует значение, с которым должна работать данная операция (например, загрузка). Аналогичным образом операция может предоставить в распоряжение свое собственное значение (например SET, смотрите таблицу 9–1).

Примеры

Таблица 9–1. Непосредственная адресация

Пример	Описание
SET	Установить VKE в 1.
OW W#16#A320	Поразрядное ИЛИ со словом
L 27	Загрузить целое число 27 в АККУ 1.
L 'ABCD'	Загрузить ASCII-символы ABCD в АККУ 1.
L B#(100,12)	Загрузить константу в виде 2 байтов (100 и 12) в АККУ 1.
L C#100	Загрузить BCD-значение 100 в АККУ 1.

9.2. Прямая адресация

Описание

При прямой адресации адрес операнда закодирован в операции, то есть операнд задает адрес значения, которое операция будет обрабатывать. Операнд состоит из двух следующих частей:

- признака операнда (например, "EB" для "входного байта"),
- точного адреса внутри области памяти, задаваемой признаком операнда.

Операнд прямо указывает на адрес значения.

Примеры

Таблица 9–2. Прямая адресация

Пример	Описание
U E 0.0	Выполнить операцию И с входным битом E 0.0.
S L 20.0	Установить бит локальных данных L20.0.
= M 115.4	Присвоить VKE меркерному биту M 115.4.
L EB 0	Загрузить входной байт EB0 в АККУ 1.
L MW 64	Загрузить меркерное слово MW64 в АККУ1.
T DBD 12	Передать содержимое АККУ 1 в двойное слово данных DBD12.

9.3. Косвенная адресация через память

Описание

При косвенной адресации операнд задает адрес значения, которое операция будет обрабатывать. Операнд состоит из двух следующих частей:

- признака операнда (например, "EB" для "входного байта") и
- слова, содержащего номер таймера (T), счетчика (Z), блока данных (DB), функции (FC) или функционального блока (FB), или
- двойного слова, содержащего точный адрес значения внутри области памяти, задаваемой признаком операнда.

Операнд задает адрес значения или номер косвенно через указатель. Это слово или двойное слово может находиться в одной из следующих областей:

- меркер (M)
- блок данных (DB)
- экземпляр блока данных (DI)
- локальные данные (L)

Преимуществом косвенной адресации через память является то, что Вы можете динамически модифицировать операнды команды во время обработки программы.

Использование правильного синтаксиса

Когда Вы работаете с косвенно адресованным через память операндом, который хранится в области памяти блока данных, Вам нужно вначале открыть блок данных, используя операцию открытия блока данных. После этого Вы можете использовать слово или двойное слово данных в качестве косвенно адресованного операнда, как показано в следующем примере:

```
AUF DB10  
L EB [DBD20]
```

Примеры

Таблица 9–3. Косвенная адресация через память

Пример	Описание
U E [MD2] или U E [anna]	Выполнить операцию И с входным битом. Точный адрес находится в двойном меркерном слове MD2 или по обозначенному через "anna" адресу в таблице символов в качестве ссылки на MD2.
= DIX [DBD2]	Присвоить бит VKE биту данных экземпляра блока данных. Точный адрес находится в двойном слове данных DBD2.
AUF DB [LW2]	Открыть блок данных. Номер блока данных находится в слове локальных данных LW2.
O A [LD3] или O A [hans]	Выполнить операцию ИЛИ с выходным битом. Точный адрес находится в двойном слове локальных данных 3 или по обозначенному через "hans" адресу в таблице символов в качестве ссылки на LD3.

Формат указателя

Есть два возможных формата указателя: слово и двойное слово. В данном руководстве наряду с описаниями операций, допускающих косвенную адресацию через память, содержится также адресная таблица, которая задает формат указателя, то есть слово или двойное слово. Сокращенное обозначение указателя в формате слова заканчивается на W (например, DBW). Рисунок 9–1 показывает указатель формата для слова. Сокращенное обозначение указателя в формате двойного слова заканчивается на D (например, DBD). Рисунок 9–2 показывает указатель формата для двойного слова.



Рис. 9-1. Указатель в формате слова для косвенной адресации через память
Два следующих примера показывают, как Вам работать с указателем в формате слова:

AWL	Объяснение
L +5	Загрузить значение 5 как целое число в АККУ 1.
T MW2	Передать содержимое АККУ 1 в меркерное слово MW2.
AUF DB[MW2]	Открыть блок данных 5.

AWL	Объяснение
AUF DB10	Открыть блок данных DB10.
L +20	Загрузить значение 20 как целое число в АККУ 1.
T DBW10	Передать содержимое АККУ 1 в слово данных DBW10.
U T[DBW10]	Опросить состояние сигнала таймера T20.

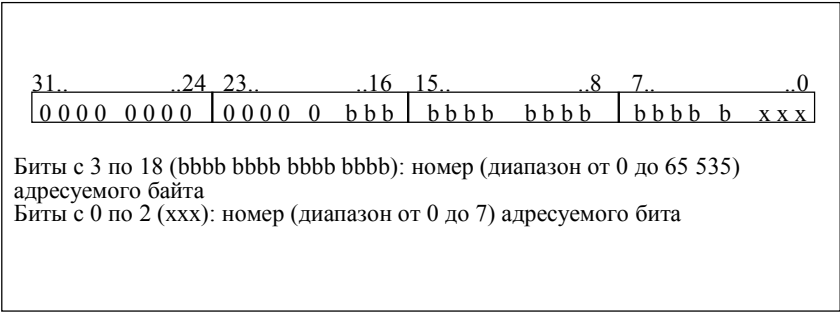


Рис. 9-2. Указатель в формате двойного слова для косвенной адресации

Указание

Если Вы обращаетесь к байту, слову или двойному слову, то вначале убедитесь, что номер бита указателя есть "0".

Два следующих примера показывают, как работать с указателем в формате двойного слова:

AWL	Объяснение
L P#8.7	Загрузить 2#0000 0000 0000 0000 0000 0000 0100 0111 (двоичное значение) в АККУ 1.
T MD2	Запомнить адрес 8.7 в двойном меркерном слове MD2.
U E [MD2]	Контроллер опрашивает вход E 8.7 и присваивает состояние его сигнала выходу A
= A [MD2]	8.7.

AWL	Объяснение
L P#8.0	Загрузить 2#0000 0000 0000 0000 0000 0000 0100 0000 (двоичное значение) в АККУ 1.
T MD2	Запомнить адрес 8 в двойном меркерном слове MD2.
L EB [MD2]	Контроллер загружает входной байт EB8 и передает содержимое в меркерное слово
T MW [MD2]	MW8.

9.4. Адресные регистры

Объяснение	Для некоторых видов косвенной адресации при программировании в AWL необходимы определенные регистры CPU. Эти регистры описаны ниже.
Адресные	Адресные регистры AR 1 и AR 2 содержат внутризонные адреса или регистры 1 и 2 адреса, содержащие указание на область памяти, для операций с косвенной регистровой адресацией. Они имеют разрядность 32 бита (смотрите главы 9.5 и 9.6).
Указатель	<p>При косвенной регистровой адресации используются указатели (смотрите главы 9.5 и 9.6). В распоряжении имеются два следующих вида:</p> <ul style="list-style-type: none">• внутризонные: для доступа внутри области к битам, байтам, словам и двойным словам в областях памяти P, E, A, M, DBX, DIX и L,• с указанием области: для доступа с указанием области памяти к битам, байтам, словам и двойным словам в областях памяти P, E, A, M, DBX, DIX и L.

9.5. Косвенная регистровая внутризонная адресация

Описание

При косвенной регистровой внутризонной адресации операнд задает адрес значения, которое операция будет обрабатывать. Операнд состоит из двух следующих частей:

- признака операнда (например, "LD" для "двойного слова локальных данных", смотрите таблицу 8–6),
- адресного регистра и указателя для задания смещения, которое прибавляется к содержимому адресного регистра для того, чтобы установить точный адрес, который операция должна обрабатывать. Указатель задается в виде P#байт.бит.

Операнд указывает на адрес значения косвенно, а именно, через адресный регистр плюс смещение.

Операция, использующая косвенную регистровую внутризонную адресацию, значение в адресном регистре не изменяет.

Вычисление адреса операнда

Операнд операции указывает на значение, которое операция будет обрабатывать. При косвенной регистровой внутризонной адресации операнд указывает на адрес значения косвенно, а именно, через адресный регистр плюс смещение. Рисунок 9–3 показывает, как вычислить адрес операнда для операции присваивания (=) в следующей команде:

$$= A [AR1, P\#1.1]$$

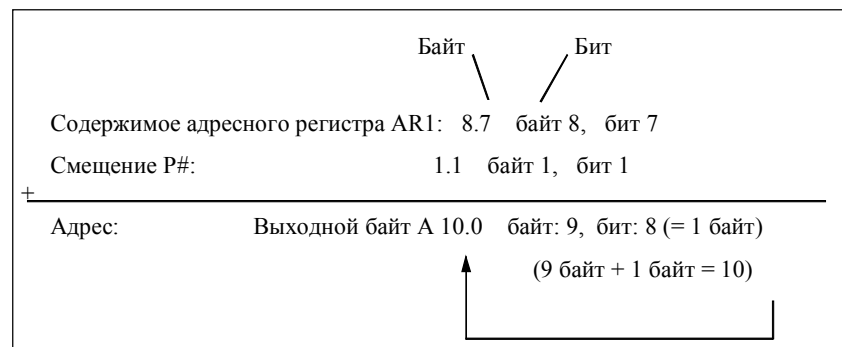


Рис. 9-3. Вычисление адреса выхода A [AR1, P#1.1]

Вы вычисляете адрес операнда, прибавляя байтовый компонент содержимого адресного регистра к байтовому компоненту указателя смещения и прибавляя битовый компонент содержимого адресного регистра к битовому компоненту указателя смещения. При вычислении байтового компонента адреса используйте десятичную систему счисления, при вычислении битового компонента адреса используйте восьмеричную систему счисления (8 бит = 1 байт). Здесь речь может идти о переносе между битовым и байтовым компонентами.

Примеры

Таблица 9–4. Косвенная регистровая внутризонная адресация

U E [AR1, P#4.3]	Выполнить операцию И с входным битом. Адрес рассчитывается как содержимое адресного регистра AR 1 плюс 4 байта плюс 3 бита.
= DIX [AR2, P#0.0]	Присвоить бит VKE биту данных экземпляра блока данных. Адрес находится в адресном регистре AR 2.
L EB [AR1, P#100.0]	Загрузить входной байт в АККУ 1. Адрес рассчитывается как содержимое адресного регистра AR 1 плюс 100 байтов.
T LD [AR2, P#56.0]	<p>Передать содержимое АККУ 1 в двойное слово локальных данных LD. Адрес рассчитывается как содержимое адресного регистра AR 2 плюс 56 байтов.</p> <p>По поводу адресации локальных данных прочитайте, пожалуйста, приведенное ниже предупреждение.</p>



компилятором, и локальными данными, используемых

программистом.

Предупреждение

Возможное переписывание данных, используемых компилятором.

Если Вы обращаетесь к временным локальным данным, используя абсолютную адресацию, то это может привести к конфликту между данными, используемыми компилятором, и локальными данными. Возможно, что затем Вы перепишите некоторые из данных, компилятором. (Например, компилятор использует локальные данные для передачи формальных параметров). Локальные данные, используемые компилятором, связаны с символическими данными, которые определяются

Если Вы желаете обратиться к временным локальным данным, то рекомендуется выбирать символическую, а не абсолютную адресацию.

Формат указателя

Для косвенной регистровой внутризонной адресации в распоряжении имеется только один возможный формат указателя: двойное слово. Это двойное слово содержит операнд, закодированный как адрес бита. В данном руководстве в дополнение к описаниям операций, допускающих косвенную регистровую внутризонную адресацию, содержится также адресная таблица, которая характеризует формат указателя, то есть двойное слово. Сокращенное обозначение указателя в формате двойного слова заканчивается на D (например, DBD). На рисунке 9-4 Вы видите формат указателя для двойного слова.

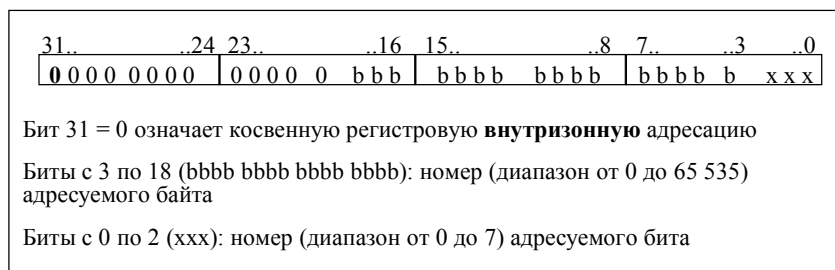


Рис. 9-4. Указатель в формате двойного слова для регистровой косвенной внутризонной адресации

Указание

Если Вы обращаетесь к байту, слову или двойному слову, то вначале убедитесь, что номер бита указателя есть "0".

Два следующих примера показывают, как работать с указателем в формате двойного слова:

AWL	Объяснение
L P#8.7	Загрузить указатель в формате двойного слова на адрес бита 8.7 в АККУ 1.
LAR1	Сохранить указатель в формате двойного слова на адрес бита 8.7 в адресном регистре AR 1.
U E [AR1, P#0.0]	CPU добавляет к содержимому адресного регистра AR 1 (8.7) смещение (P#0.0) и использует этот адрес как операнд операции И с битом. Содержимое AR1 остается неизменным.
= A [AR1, P#1.1]	CPU присваивает результат логической операции (VKE) адресу (A 10.0). CPU вычисляет этот операнд, складывая содержимое адресного регистра AR1 (8.7) и смещение (P#1.1).

AWL	Объяснение
L P#8.0	Загрузить указатель в формате двойного слова на адрес бита 8.0 в АККУ 1.
LAR2	Сохранить указатель в формате двойного слова на адрес бита 8.0 в адресном регистре AR 2.
L EB [AR2, P#2.0]	CPU загружает входной байт EB10 в АККУ 1.
T MW [AR2, P#200.0]	CPU передает содержимое АККУ 1 в меркерное слово MW208.
	Адрес 208 рассчитывается как 8 (AR 2) плюс 200 (смещение). Это дает 208.

9.6. Косвенная регистровая адресация с указанием в регистре области памяти

Описание

При косвенной регистровой адресации с указанием в регистре области памяти, операнд задает адрес значения, которое операция будет обрабатывать, и состоит из двух следующих частей:

- признака операнда, задающего размер объекта данных (например, "В" для "байта", смотрите таблицу 8–8). Область памяти задается в битах 24, 25 и 26 адресного регистра.
- адресного регистра и указателя для задания смещения, которое прибавляется к содержимому адресного регистра для того, чтобы установить точный адрес, который операция должна обрабатывать. Указатель задается в виде P#байт.бит.

Операнд указывает на адрес значения косвенно, а именно, через адресный регистр плюс смещение.

Операция, которая использует косвенную регистровую адресацию с указанием области памяти, значение в адресном регистре не изменяет.

Вычисление адреса операнда

Операнд операции указывает на значение, которое операция будет обрабатывать. При косвенной регистровой адресации с указанием области памяти, операнд указывает на адрес значения косвенно, а именно, через адресный регистр плюс смещение. Рисунок 9–5 показывает, как вычислить адрес операнда для операции присваивания (=) в следующей команде:

= [AR1, P#1.1]

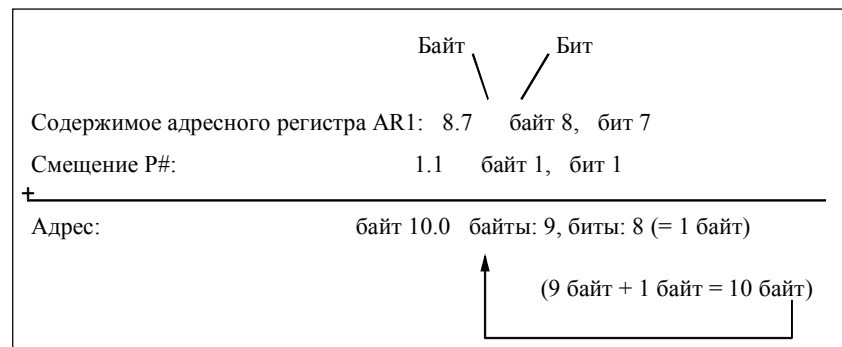


Рис. 9-5. Вычисление адреса [AR1, P#1.1]

Вы вычисляете адрес операнда, прибавляя байтовый компонент содержимого адресного регистра к байтовому компоненту указателя смещения и прибавляя битовый компонент содержимого адресного регистра к битовому компоненту указателя смещения. При вычислении байтового компонента адреса используйте десятичную систему счисления, при вычислении битового компонента адреса используйте восьмеричную систему счисления (8 бит = 1 байт). Здесь речь может идти о переносе между битовым и байтовым компонентами.

Пример

Таблица 9–5 дает примеры косвенной регистровой адресации, охватывающей области. Операнд должен содержать дополнительный признак области в битах 24, 25 и 26 указателя. Запрашиваемая информация находится в адресном регистре.

Таблица 9–5. Косвенная регистровая адресация с указанием области памяти

Пример	Описание
U [AR 1, P#4.3]	Выполнить операцию И с битом. Адрес вычисляется как содержимое адресного регистра AR 1 плюс 4 байта плюс 3 бита. Область памяти бита задана в битах 24, 25 и 26 адресного регистра AR 1.
= [AR 2, P#0.0]	Присвоить бит VKE биту. Адрес бита находится в адресном регистре AR 2. Область памяти бита задана в битах 24, 25 и 26 адресного регистра AR 2.
L B [AR 1, P#100.0]	Загрузить байт в АККУ 1. Адрес вычисляется как содержимое адресного регистра AR 1 плюс 100 байтов. Область памяти байта задана в битах 24, 25 и 26 адресного регистра AR 1.
T D [AR 2, P#56.0]	Передать содержимое АККУ 1 в двойное слово. Адрес двойного слова вычисляется как содержимое адресного регистра AR 2 плюс 56 байтов. Область памяти двойного слова задана в битах 24, 25 и 26 адресного регистра AR 2.

Таблица 9–6 перечисляет значения двоичного кода в битах 24, 25 и 26 указателя, которые характеризуют эту область.

Таблица 9–6. Признак области для косвенной регистровой адресации с указанием области памяти

Признак области (область памяти)	Двоичный код в битах 26, 25, и 24
P (периферийная область входов и выходов)	000
E (отображение процесса на входах)	001
A (отображение процесса на выходах)	010
M (меркеры)	011
DBX (блок данных)	100
DIX (экземпляр блока данных)	101
(Локальные данные вызывающего блока, т.е. локальные данные непосредственного предшественника в иерархии вызовов).	111

Формат указателя

Для косвенной регистровой адресации с указанием области памяти в распоряжении имеется только один возможный формат указателя: двойное слово. Это двойное слово содержит операнд, закодированный как адрес бита. В данном руководстве в дополнение к описаниям операций, допускающих косвенную регистровую адресацию с указанием области памяти, содержится также адресная таблица, которая характеризует формат указателя, то есть двойное слово. Сокращенное обозначение указателя в формате двойного слова заканчивается на D (например, DBD). На рисунке 9–6 Вы видите формат указателя для двойного слова.

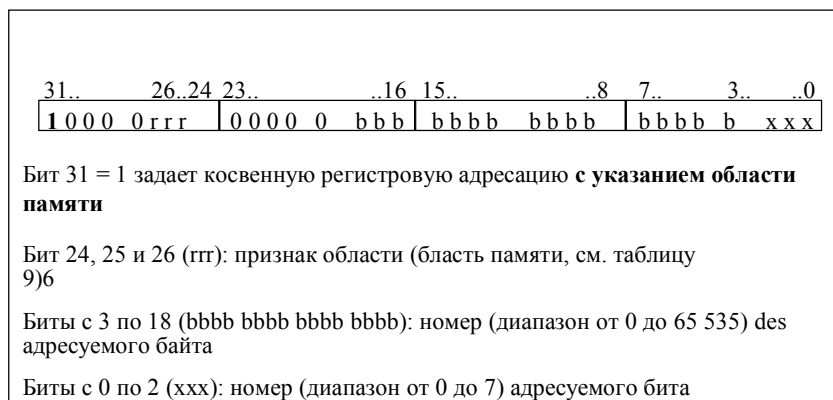


Рис. 9-6. Указатель в формате двойного слова для косвенной регистровой адресации с указанием области памяти

Указание

Если Вы обращаетесь к байту, слову или двойному слову, то вначале убедитесь, что номер бита указателя есть "0".

К локальным данным нельзя обращаться посредством косвенной регистровой адресации с указанием области памяти!

Два следующих примера показывают, как работать с указателем в формате двойного слова:

AWL	Объяснение
L P#E8.7	Загрузить указатель в формате двойного слова на адрес бита E 8.7 в АККУ 1. Сохранить указатель в формате двойного слова на адрес бита E 8.7 в AR 1.
LAR1	Загрузить указатель в формате двойного слова на адрес бита A 8.7 в АККУ 1. Запомнить указатель в формате двойного слова на адрес бита A 8.7 в AR 2.
L P#A8.7	CPU складывает содержимое адресного регистра AR1 (P# E 8.7) и смещение (P#0.0) и использует операнд, на который указывает результат (E 8.7), в качестве операнда операции И с битом. Содержимое AR1 остается неизменным.
LAR2	
U [AR1, P#0.0]	CPU присваивает результат логической операции (VKE) операнду (A 10.0). CPU вычисляет этот операнд, складывая содержимое адресного регистра AR2 (P#A 8.7) и смещение (P#1.1) и деактивизируя указатель. Содержимое AR2 остается неизменным.
= [AR2, P#1.1]	

AWL	Объяснение
L P#E8.0	Загрузить указатель в формате двойного слова на адрес бита E 8.0 в АККУ 1.
LAR2	Сохранить указатель в формате двойного слова на адрес бита E 8.0 в адресном регистре AR 2.
L P#M8.0	Загрузить указатель в формате двойного слова на адрес бита M 8.0 в АККУ 1. Сохранить указатель в формате двойного слова на адрес бита M 8.0 в адресном регистре AR 1.
LAR1	CPU загружает входной байт EB10 в АККУ 1.
L B [AR2, P#2.0]	CPU передает содержимое АККУ 1 в меркерное слово MW208.
T W [AR1, P#200.0]	Входной байт 10 вычисляется как 8 (AR 2) плюс 2 (смещение). Меркерное слово 208 вычисляется как 8 (AR 1) плюс 200 (смещение). Это дает 208.