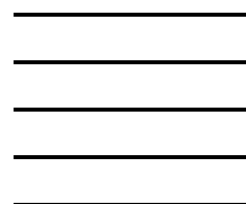


# Разделитель программ для записи в ППЗУ



5



## 5.1. Введение

Программа разделителя программ выделяет ПЗУ части из выходного .EXE файла редактора связей и формирует информацию для использования программаторами ППЗУ.

Программа разделителя программ может генерировать файлы для памяти программ, памяти данных или памяти начальной загрузки. Три файла создаются для памяти программ, чтобы организовать ППЗУ в трехбайтовых словах, соответствующих 24-разрядным инструкциям ADSP-21xx. Два файла создаются для памяти данных, чтобы сгруппировать данные в двухбайтовые слова, и один файл байтового формата создается для памяти начальной загрузки.

Файлы байтовых потоков могут быть созданы для памяти программ и памяти данных, чаще вертикальной организации слов, чем горизонтальной. Программа разделения программ для программаторов ППЗУ может генерировать файлы в форматах **Motorola S** или **Intel Hex**. Кроме того, поддерживается формат **Motorola S2** для вывода байтовых потоков.

## 5.2. Запуск программы разделения

Для запуска программы используется следующая команда:

```
SPL21 файл_образа ППЗУфайл [-ключ ...]
```

При каждом запуске генерируются выходные файлы для одного типа памяти ADSP-21xx – памяти программ, памяти данных или памяти начальной загрузки. Если требуется создать файлы для более чем одного типа памяти, необходимо запустить программу разделителя программ для программаторов ППЗУ несколько раз.

Входной файл *файл\_образа* является .EXE файлом вашей программы, сгенерированным редактором связей. Этот файл должен иметь расширение .EXE, добавленное редактором связей; в противном случае программа разделителя не распознает его.

Каждый раз, когда запускается программа разделителя, необходимо называть имя выходного файла *ППЗУфайл*. При создании файлов для более чем одного типа памяти (PM, DM, BM), необходимо каждый раз выбирать различные имена файлов, чтобы избежать наложения на результаты предшествующих запусков.

При запуске программы разделителя в командной строке могут использоваться пять ключей (один обязательный, остальные дополнительные).

Ключ	Возможные формы	
Тип памяти	<b>-pm, -dm, -bm</b>	(обязательный)
Формат файла ППЗУ	<b>-s, -i, -us, -us2, -ui</b>	(по умолчанию -s)
Размер страниц	<b>-bs pagesize</b>	(по умолчанию 2К)
Границы страниц	<b>-bb boundary</b>	(по умолчанию 2К)
Программа начальной загрузки	<b>-loader</b>	

Ключи, определяющие пространство памяти и формат файла ППЗУ:

<b>-pm</b>	память программ	
<b>-dm</b>	память данных	
<b>-bm</b>	память начальной загрузки	
<b>-s</b>	формат Motorola S	(по умолчанию)
<b>-i</b>	формат Intel Hex	
<b>-us</b>	формат Motorola S, байтовый поток	(только с -pm или -dm)
<b>-us2</b>	формат Motorola S2, байтовый поток	(только с -pm или -dm)
<b>-ui</b>	формат Intel Hex, байтовый поток	(только с -pm или -dm)

Ключи -bs, -bb и -loader описаны в следующих разделах этой главы.

### 5.2.1. Пример создания файлов для памяти PM и DM

Чтобы создать файлы для программаторов ППЗУ для памяти программ и памяти данных, нужно запустить программу разделителя дважды:

```
spl21 fir_sys pmburn -pm
spl21 fir_sys dmburn -dm
```

В приведенном примере имеется файл, созданный редактором связей – FIR\_SYS.EXE, который содержит программы и данные для записи в ППЗУ. Заметьте, что каждому выходному файлу программы разделителя дано уникальное имя. Выходные файлы созданы в формате Motorola S, так как отсутствует ключ указания формата ППЗУ.

### 5.2.2. Пример создания файла только для памяти BM

Многие системы ADSP-21xx используют для хранения программ только память начальной загрузки. Для этих систем программу разделителя запускают один раз, чтобы создать файл для памяти начальной загрузки ППЗУ, например:

```
spl21 iir-sys bootburn -bm -i
```

Здесь используется ключ -i, чтобы сгенерировать файл в формате Intel Hex.

### 5.3. Выходные файлы программы разделителя

Когда установлен ключ `-pm`, программа разделителя для программаторов ППЗУ создает три файла байтовой ширины. Один файл содержит старшие байты 24-разрядных слов и имеет расширение `.BNU`. Второй файл содержит средние байты и имеет расширение `.BNM`, и третий файл содержит младшие байты и имеет расширение `.BNL`.

Когда установлен ключ `-dm`, программа разделителя для программаторов ППЗУ создает два файла байтовой ширины. Один файл содержит старшие байты 16-разрядных слов и имеет расширение `.BNM`. Второй файл содержит младшие байты и имеет расширение `.BNL`. Файл с расширением `.BNU` не создается.

Когда установлен ключ `-bm`, программа разделителя для программаторов ППЗУ создает единственный файл с расширением `.BNM`, который включает программы и данные для всех страниц начальной загрузки. Файлы с расширением `.BNU` и `.BNL` не создаются.

Рис.5.1 показывает выходные файлы разделителя программ для записи в ППЗУ.

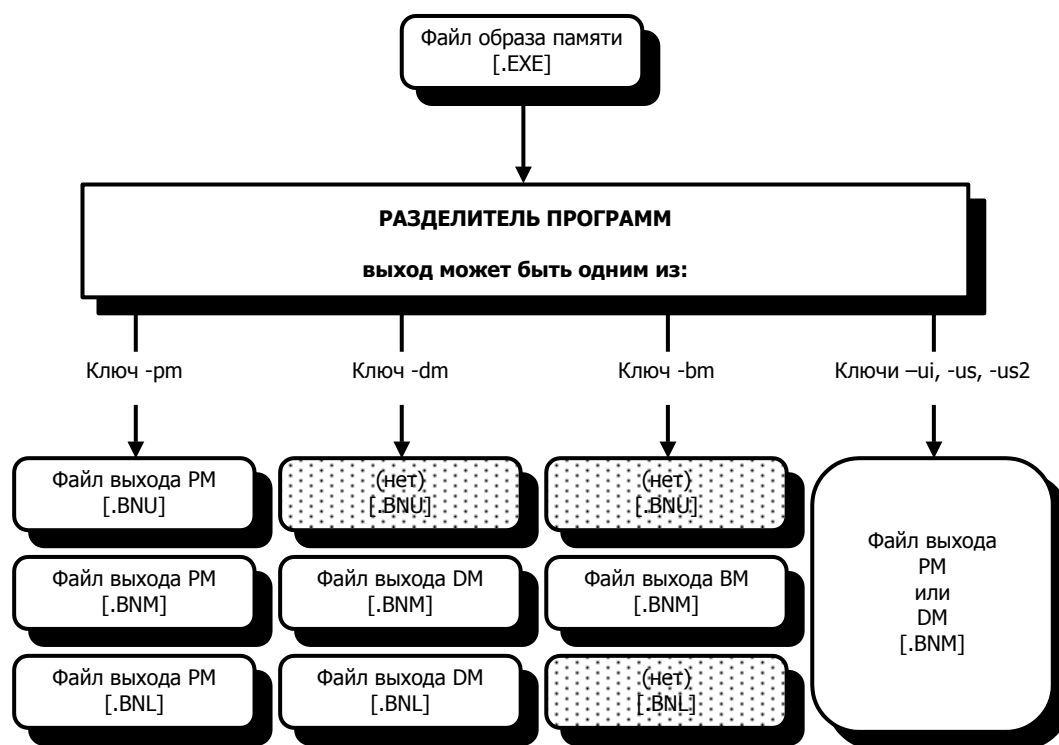


Рис.5.1. Вход/выход разделителя программ для программаторов ППЗУ.

#### 5.3.1. Выходной байтовый поток для РМ и DM

Ключами `-us`, `-us2` или `-ui` выбирается выходной байтовый поток и генерируется единственный файл (`.BNM`) с вертикальной организацией слов в памяти. Этот формат может быть выбран только для памяти программ или памяти данных.

Выходной файл байтового потока построен от самого старшего байта каждого слова к самому(ым) младшему(им) байту(ам), другими словами, старший байт каждого слова находится по младшему адресу. 24-разрядные слова памяти программ требуют последовательности из трех байт, в то время как 16-разрядные слова памяти данных последовательности из двух.

Если были назначены любые абсолютные адреса объектам памяти, такая информация в формате байтового потока будет утеряна.

Нельзя также создавать файлы байтового потока из входной программы, которая содержит несмежные блоки памяти (неперемещаемые сегменты с неиспользуемыми блоками памяти между ними). Программные модули должны быть или перемещаемыми, или их необходимо поместить в смежные блоки памяти.

### 5.3.2. Организация памяти начальной загрузки

Загружаемая память обладает байтовой шириной и организована в вертикальные группы из четырехбайтовых слов. Старший байт каждого слова размещают по младшему адресу четырехбайтовой группы.

Каждое 32-разрядное слово состоит из 24-разрядной инструкции, дополненной байтом. Добавленный байт (0xFF) игнорируется, за исключением первого байта каждой загружаемой страницы. Дополнительный байт первого слова на странице содержит информацию о длине страницы начальной загрузки.

Длины страниц подсчитываются программой разделителя и вставляются в файл образа ППЗУ: для страницы 0 эта величина размещена в ППЗУ по байтовому адресу 0x0003.

Разделитель программ для программаторов ППЗУ подсчитывает длину каждой страницы начальной загрузки следующим образом:

$$\text{Длина страницы} = \frac{\text{Количество 24-разрядных инструкций}}{8} - 1$$

(количество инструкций должно быть округлено до кратного восьми).

Например, нулевая длина страницы показывает восемь слов, представленных 32 последовательными байтами. Максимальная длина страницы = 255 показывает 2048 слов.

Каждая страница начальной загрузки должна содержать количество слов, которое кратно восьми, если необходимо, программа разделителя добавляет дополнительные слова (0xFFFFFFFF) в конце страницы.

## 5.4. Страницы начальной загрузки меньше 2К

Ключи `-bs` и `-bb` разрешают создание загружаемых страниц размером меньше 2К. Системы ADSP-21xx с памятью начальной загрузки обычно используют страницы размером 2К слов. Однако такие системы, как ADSP-2105 и ADSP-2115 требуют, чтобы размер страниц начальной загрузки был 1К. Использование страниц меньшего размера позволит использовать преимущества многостраничных систем с небольшими программами, так как для них необходимо меньше по объему EPROM.

Когда программа разделителя запущена без ключа `-bs`, размер загружаемой страницы по умолчанию формируется равным 2К, что позволяет хранить 2048 слов.

Применяя ключ `-bs` можно создать загрузочные страницы меньшего размера; аргумент *pagesize* может быть величины от 0 до 2048 и должен быть введен в десятичном виде.

$$0 \leq \text{pagesize} \leq 2048 \qquad (\text{по умолчанию} = 2048)$$

Ключ `-bb` используют, чтобы сделать небольшие страницы непрерывными в памяти. Аргумент *boundary* определяет стартовый адрес последующих страниц и может принимать следующие значения:

$$\text{boundary} = 2048, 1024, 512 \text{ или } 256 \qquad (\text{по умолчанию} = 2048)$$

Например, если размер страницы = 1024 и значение ограничения = 2048, то страницы размером 1К будут начинаться с адресов, кратных 2048: 0, 2048, 4096, 6144, 8192. Это приводит к тому, что остаются неиспользованными 1024 адреса пространства ППЗУ.

Заметьте, что программы моделирования ADSP-2101, ADSP-2111 и ADSP-21msp50 в состоянии моделировать только загрузку страниц размером 2К. Если вы создадите загрузочные страницы меньшего размера, вы не сможете использовать команду симулятора LR, чтобы загрузить файл образа ППЗУ. Тем не менее, команда L может быть применена, чтобы загрузить и моделировать исполняемый .EXE файл.

### 5.4.1. Страницы начальной загрузки размером 1К для ADSP-2105/2115

Для создания страниц начальной загрузки размером 1К, непрерывных в памяти, для систем ADSP-2105 или ADSP-2115, используйте ключи `-bs` и `-bb` следующим образом:

```
sp121  файлEXE  файлППЗУ  -bm  -bs 1024  -bb 1024
```

### 5.4.2. Использование адресной линии для памяти начальной загрузки

Если вы используете ключи `-bs` и `-bb` для создания страниц начальной загрузки размером меньше 2К в системах ADSP-2101, ADSP-2111 или ADSP-21msp50, или для создания загружаемых страниц меньших 1К для систем ADSP-2105 или ADSP-2115, необходимо изменить подключение шины адреса к памяти начальной загрузки:

<i>Размер страницы</i>	<i>Отсоединить</i>
1024	A12*
512	A12, A11
256	A12, A11, A10

\* Нет необходимости отключать A12 в системах ADSP-2105 или ADSP-2115.

Эти изменения позволяют процессору адресовать загружаемые страницы меньшие 2К без незаполненных блоков памяти. Обычно (со страницами 2К) ADSP-2101 используют следующие соединения с адресами памяти начальной загрузки:

- D23, D22, A13 выбирают страницы начальной загрузки (0-7);
- A12 – A0 выбирают байтовый адрес (0-8191) на каждой странице.

Отключая A12 и A11, ADSP-2101 может загрузить страницы размером 1/2К. Связи D23, D22, A13 и A10–A0 адресуют тогда полное пространство памяти начальной загрузки в 4К слов.

Чтобы адресовать загружаемые страницы размером 1К ADSP-2105 и ADSP-2115 используют следующие соединения:

- D23, D22, A13 выбирают страницы начальной загрузки (0-7);
- A11–A0 выбирают байтовый адрес (0-4095) на каждой странице.

## 5.5. Использование ключа –loader.

Опция загрузчика (ключ -loader) разделителя программ для записи программаторами ППЗУ создает загружаемую версию программы, которая автоматически инициализирует все необходимое пространство ОЗУ ADSP-21xx во внутренней памяти программ. Чтобы осуществлять инициализацию загрузки каждой следующей страницы, добавляются подпрограммы загрузки памяти.

Стандартная операция загрузки ADSP-2101, ADSP-2111 и ADSP-21msp50 работает при перезапуске только с 2К внутренней памятью программ. Ни внутренняя память данных, ни внешняя память программ, или внешняя память данных не загружаются.

Если ваша программа начальной загрузки требует инициализацию любой из перечисленных выше областей памяти (в ОЗУ), необходимо написать подпрограммы для копирования кодов/данных из внутренней памяти программ, или использовать ключ -loader (можно также использовать запрограммированные ПЗУ устройства для внешней памяти).

Смотрите раздел «Инициализация вашей программы в памяти» в главе 3.

Ключ -loader позволяет также создавать многостраничные программы, которые полностью загружаются при перезапуске – коды и данные загружаются во внутреннюю память программ и копируются из нее в другие области, для каждой страницы одновременно.

Чтобы использовать опцию загрузчика для систем ADSP-2101, ADSP-2111 или ADSP-21msp50 с размером страниц начальной загрузки 2К, вызовите программу разделителя только с ключом `-loader` и ключом формата ППЗУ файла:

```
spl21 файлEXE файлППЗУ -loader [-s] [-i]
```

Чтобы использовать опцию загрузчика для систем ADSP-2105 или ADSP-2115 со страницами начальной загрузки размером 1К, вызовите разделитель программ для записи в ППЗУ с ключом `-loader`, ключом формата ППЗУ файла и ключами `-bs` и `-bb` с аргументами `pagesize` и `boundary`, устанавливающими значение 1024:

```
spl21 файлEXE файлППЗУ -bs 1024 -bb 1024 -loader [-s] [-i]
```

При этом нельзя использовать ключи типа памяти `-bm`, `-pm` или `-dm`. Автоматически генерируется файл с расширением `.BIM`. Выходной файл должен использоваться для программирования загрузочной памяти EPROM. В этом файле включена ваша полная исполняемая программа, включая все коды и данные, которые должны быть скопированы в память программ и память данных ОЗУ.

Для страниц размером 2К процесс загрузки и копирования изображен на Рис.5.2.

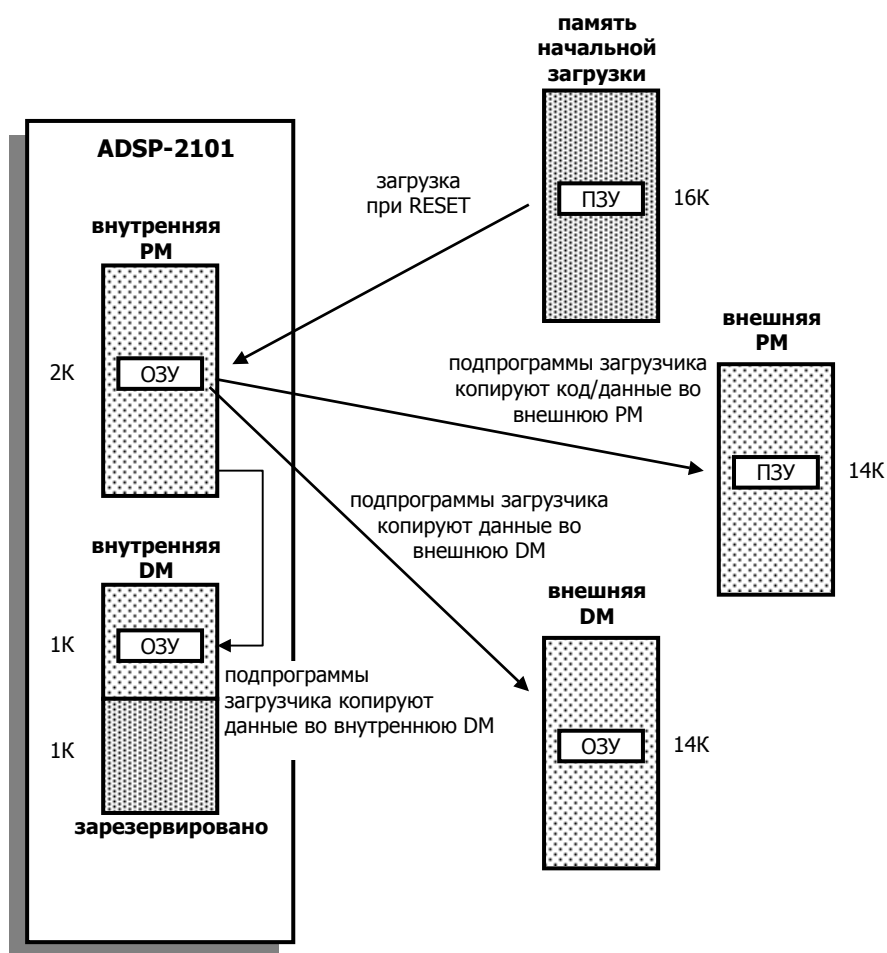


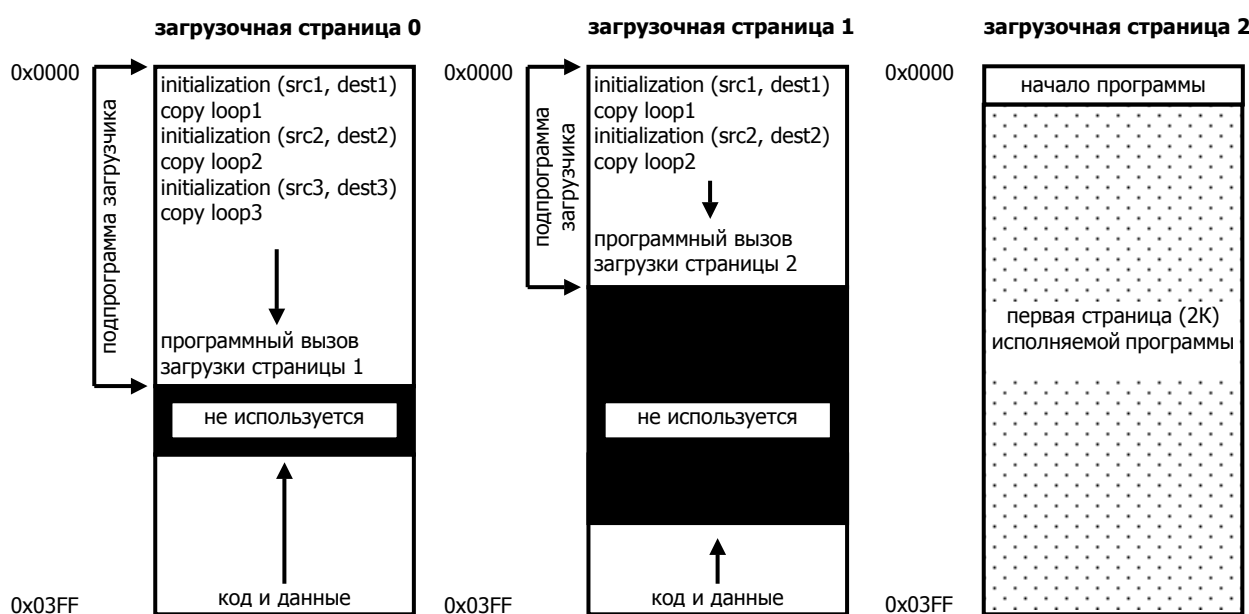
Рис.5.2. Загрузка программы и инициализация памяти.



Ключ `-loader` указывает разделителю программ просматривать входной исполняемый файл для поиска РМ сегментов внешней памяти ОЗУ и DM сегментов внутренней или внешней памяти ОЗУ; при этом создается столько загружаемых страниц, сколько необходимо для того чтобы вместить все коды и данные, не обращая внимание на то, как много страниц объявлено в файле описания системной архитектуры. Вдобавок, небольшие загружающие подпрограммы создаются и размещаются в начале (младшие адреса) каждой страницы.

Рис.5.3. показывает программу, созданную с ключом `-loader`, которая содержится на страницах начальной загрузки 0, 1 и 2. Подпрограммы загрузки на страницах 0 и 1 содержат несколько циклов, каждая из которых после загрузки копирует сегмент кода или данных в соответствующую память программ или данных. Один цикл копирования требуется для каждого отдельного (несвязанного) сегмента.

В конце каждой страницы подпрограмм загрузчика следует последовательность инструкций, которая программно вызывает следующую страницу. Последняя страница (в приведенном примере страница 2) не имеет подпрограмм загрузки - она содержит первую 2К страницу исполняемой программы. После загрузки этой страницы, выполнение программы начнется с адреса 0 во внутренней памяти программ ADSP-21xx:



**Рис.5.3. Пример программы начальной загрузки.**

Ключ `-loader` заставляет программу разделителя размещать коды и данные на каждой загружаемой странице (кроме последней), начиная с самого старшего адреса вниз (Рис.5.3). Неиспользованная область, обычно, остается посередине.

После загрузки страницы 0 сегменты кодов и данных копируются подпрограммами загрузчика в соответствии с их назначениями. Затем страница 0 осуществляет программный вызов страницы 1, где подпрограммы загрузчика выполняют подобные операции. Последовательная загрузка продолжается до тех пор, пока программа не загрузится целиком, максимум до 15К памяти начальной загрузки.



На примере Рис.5.3. осуществляется следующая последовательность событий после перезапуска системы:

1. Страница 0 загружается во внутреннюю память программ ADSP-21xx. Выполняются подпрограммы загрузчика страницы 0, копирующие коды и данные со страницы 0 во внутреннюю DM, внешнюю PM и/или внешнюю DM.
2. Страница 1 загружается подпрограммой загрузчика страницы 0. Выполняются подпрограммы загрузчика страницы 1, копирующие коды и данные. Это завершает процесс инициализации.
3. Страница 2 загружается подпрограммой загрузчика страницы 1. Основная исполняемая программа стартует с адреса 0.

При установке системного управляющего регистра отраженного в памяти для вызова программной начальной загрузки, подпрограммы загрузчика по умолчанию устанавливают в поле BWAIT этого регистра значение 3.

### 5.5.1. Как подготовить программу для начального загрузчика

Если предполагается использование ключа `-loader` программы разделителя, чтобы создать загружаемую программу с автоматической инициализацией, необходимо предпринять следующие шаги:

1. Объявить максимальное допустимое количество загружаемых страниц, восемь, в вашем системном файле архитектуры.
2. Записать ваши ассемблерные модули без использования определителя `/BOOT` в директиве `.MODULE`.
3. Ассемблировать и компоновать.
4. Вызвать разделитель программ для программаторов ППЗУ с ключом `-loader`. Во время обработки показывается отображение процесса созданных страниц (Чтобы сохранить эту информацию воспользуйтесь командами операционной системы для перенаправления текста в файл).

Когда используется опция загрузчика нельзя присваивать программным модулям ассемблерный определитель `/BOOT`. Это позволит программе определить размещение загружаемой страницы для каждого модуля.

Заметьте, что при копировании кодов/данных во внешнюю память программ или данных подпрограммы загрузчика, добавленные к вашей программе, могут записать другие адреса памяти, чем те, что используются основной программой. Например, если основная программа инициализирует адреса `DM[0x03FF]` и `DM[0x0401]`, и определен порт ввода/вывода отображенный по адресу `DM[0x0400]`, подпрограмма загрузчика может записать в порт неопределенную величину. Необходимо гарантировать, что в системе не будет таких ложных записей. Один из способов сделать это состоит в объявлении для всех портов уникального сегмента памяти (директивой `.SEG` системного конфигулятора) в файле описания архитектуры. Если код или данные не размещаются в этом сегменте, порты не будут перезаписаны в течение инициализации памяти.

### 5.5.2. Моделирование программы начальной загрузчика

Так как функция начальной загрузки реализуется программой разделителем, а не редактором связей, необходимо использовать выходной файл `.BNM` программы разделителя для моделирования загружаемой программы. Выходной файл редактора связей не содержит информации о памяти начальной загрузки.

Чтобы загрузить файл `.BNM` в программу моделирования ADSP-21xx или эмулятор, используйте команду LR, вместо команды L.

## 5.6. Файлы HIP загрузки (HIP splitter)

HIP разделитель программ – это утилита, подобная разделителю программ для записи в ППЗУ, которая позволяет создавать файлы, которые могут быть загружены через интерфейс управляющего порта (host interface port – HIP) ADSP-2111 и ADSP-21msp50.

HIP разделитель программ вызывается следующей командой:

```
hspl21 файлEXE [addr] [-boot] [-i]
```

Входной файл *файлEXE* – это файл копии содержимого памяти редактора связей. Этот файл должен иметь расширение `.EXE`, добавленное редактором связей; в противном случае, HIP разделитель программ его не распознает. Дополнительный (необязательный) параметр `addr` характеризует отступление от нулевого стартового адреса для загрузочного HIP файла. Этот параметр должен быть введен десятичным номером, без префикса «0x».

Выходной файл HIP разделителя всегда называется `IMAGEFILE.HIP`. Формат файла подобен файлу `.BNM`, который создает разделитель программ для записи в ППЗУ для памяти программ или памяти начальной загрузки – первый байт загружается в регистр процессора HDR3, содержащий длину страницы. По умолчанию используется формат **Motorola S**. Используя ключ `-i` можно создавать файлы в формате **Intel Hex**.

Если HIP разделитель вызван без ключа `-boot`, `IMAGEFILE.HIP` содержит 24-разрядные слова памяти программ, размещенные в следующем порядке: **старший байт, средний байт, младший байт**. Чтобы разместить эти три байта в порядке, используемом операциями начальной загрузки ADSP-21xx: **старший байт, младший байт, средний байт**, используйте ключ `-boot`.