



380

К.59

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ  
им. Г.И. Будкера СО РАН

В.Р. Козак

№ 6

БИБЛИОТЕКА РАДИОИНЖЕНЕРА  
СИСТЕМА ПРОЕКТИРОВАНИЯ АБЕЛЬ  
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ  
(Информационно-справочный материал)



НОВОСИБИРСК



БИБЛИОТЕКА РАДИОИНЖЕНЕРА  
СИСТЕМА ПРОЕКТИРОВАНИЯ АБЕЛЬ

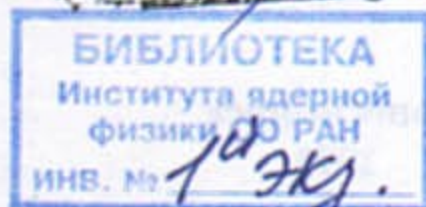
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

(Информационно-справочный материал)

Институт ядерной физики им. Г.И.Будкера  
630090, Новосибирск 90, Россия

АННОТАЦИЯ

Издание предназначено для разработчиков радиоэлектронной аппаратуры. Пособие представляет из себя руководство пользователя АБЕЛЬ-системы проектирования программируемых логических устройств.



Институт ядерной физики им.Г.И.Будкера СО РАН

СОДЕРЖАНИЕ

ГЛАВА 1 ВВЕДЕНИЕ	5
1.1. Соглашения об обозначениях	5
ГЛАВА 2 ЯЗЫКОВОЙ ПРОЦЕССОР	7
2.1. ABEL.BAT, автоматическая обработка	8
2.2. PARSE	9
2.2.1. Файл листинга программы PARSE	11
2.3. TRANSFOR	13
2.4. REDUCE	14
2.5. FUSEMAP	15
2.6. SIMULATE	17
2.6.1. Выходной файл программы SIMULATE	20
2.6.2. Симуляция для проектов с обратной связью	23
2.7. DOCUMENT	31
2.7.1. Выходной файл программы DOCUMENT	32
ГЛАВА 3 УТИЛИТЫ	35
3.1. TOABEL, преобразователь PALASM в АБЕЛЬ	35
3.2. IFLDOC	35
ПРИЛОЖЕНИЕ А "Batch" файл ABEL.BAT	38
ПРИЛОЖЕНИЕ В Поддерживаемые микросхемы	39
ПРИЛОЖЕНИЕ С Сообщения об ошибках	42
С.1. Обычные сообщения об ошибках	42
С.1.1. Ошибки командной строки	42
С.1.2. Ошибки файла устройства	43
С.1.3. Ошибки фатальные	43
С.1.4. Ошибки промежуточных файлов	44
С.1.5. Ошибки внутренние	45



C.1.6. Ошибки логические	46
C.1.7. Ошибки препроцессора	48
C.1.8. Ошибки синтаксические	48
C.2. Нефатальные ошибки симуляции	51
C.3. Сообщения об ошибках программы TOABEL	51
C.4. Ошибки программы IFLDOC	52
C.4.1. Ошибки командной строки	52
C.4.2. Ошибки диагностические	52
C.4.3. Ошибки входного файла	52
C.4.4. Предупреждающие сообщения	53
<b>ПРИЛОЖЕНИЕ D Узлы устройств</b>	<b>54</b>

## ГЛАВА 1 ВВЕДЕНИЕ

ABEL является мощным инструментом для проектирования с программируемой логикой, состоящий из языка проектирования, который легко описывает логические функции, и языкового процессора, который преобразует логическое описание в загрузочные файлы программатора. Эти загрузочные файлы используются чтобы как запрограммировать, так и протестировать программируемые логические микросхемы. В процессе преобразования языковой процессор также:

- выполняет логическую редукцию;
- проверяет правильность синтаксиса;
- проверяет что логические функции могут быть реализованы с помощью выбранной микросхемы;
- моделирует операции запрограммированной микросхемы прежде чем она будет запрограммирована.

### 1.1. СОГЛАШЕНИЯ ОБ ОБОЗНАЧЕНИЯХ

Нижеследующая таблица (1-1) показывает соглашения об обозначениях, используемых в этом руководстве.

Таблица 1-1. Соглашение об обозначениях

Обозначение	Использование
Наклонный шрифт	Отображает ссылки по имени на понятия, содержащиеся в примерах, рисунках, листингах и таблицах.
Кавычки ("")	Окружают понятия отображенные наклонными буквами, когда они содержат пробелы.
<b>БОЛЬШИЕ БУКВЫ</b>	В синтаксических описаниях и диаграммах отражает что слово или буквы большого размера должны быть введены полностью. Понятие может быть введено буквами большими, маленькими или разного размера. Отображает ключевые слова.
маленькие буквы	В синтаксических описаниях и диаграммах отражает что значение или имя вводится пользователем на месте слова или букв маленького размера. Это понятие может быть введено буквами большими, маленькими или разного размера.



Обозначение	Использование
Квадратные скобки [ ]	Окружают понятия которые могут либо вводиться, либо опускаться, по необходимости.
Круглые скобки ( )	Отражает, что предшествующее понятие может быть повторено если это необходимо.
Все другие символы пунктуации	Апострофы, фигурные скобки, запятые и т.п. должны быть введены точно так, как это показано.

Пример кавычек, использованных в тексте:

Спецификация файла "m6809a fus" является неправильной, потому что имя файла и его расширение разделены пробелом. Правильной спецификацией будет "m6809a.fus".

Пример больших и маленьких букв, квадратных скобок используется в синтаксическом описании:

PARSE [-Iin\_file] [-Aarg]...

Должна быть введена команда PARSE, но она может быть введена буквами большими, маленькими либо смешанного размера. Входной файл может быть (но может и не быть) определен печатанием -I либо большой либо маленькой буквой с последующей спецификацией файла на месте in file. Могут быть добавлены аргументы вслед за параметром -A. Параметр -A печатается либо большой либо маленькой буквой. Аргумент вводится пользователем на месте arg. Может быть введено столько аргументов, сколько нужно, но каждый новый аргумент требует использования нового параметра -A. Ниже приводится правильный пример:

parse -Ap16r4 -aGND

Определение командного синтаксиса языкового процессора АБЕЛЬ дается с помощью синтаксического описания. Синтаксическое описание содержит полное и формальное определение команды и определение индивидуальных частей, которые составляют команду. Первая строка синтаксического описания (показана ниже) определяет синтаксис в соответствии с соглашениями о нотациях, перечисленных выше. Строки, следующие за первой, являются элементами этого синтаксиса. Синтаксическое описание размещено справа от этих элементов. Рисунок 1-1 показывает пример синтаксического описания.

Рисунок 1-1. Пример синтаксического описания.

```

REDUCE [-Iin_file][-Oout_file][-RnZ
-Iin_file    преобразуемый код содержится в in_file
-Oout_file   вывод REDUCE записывается в out_file
-Rn          n=0,1,2 или 3, отражает уровень редукции,
             который должен выполняться (по умолчанию 1)
             n=0: нет редукции
             n=1: простая редукция
             n=2: редукция простая и PRESTO
             n=3: редукция простая и PRESTO по выводам

```

## ГЛАВА 2 ЯЗЫКОВОЙ ПРОЦЕССОР

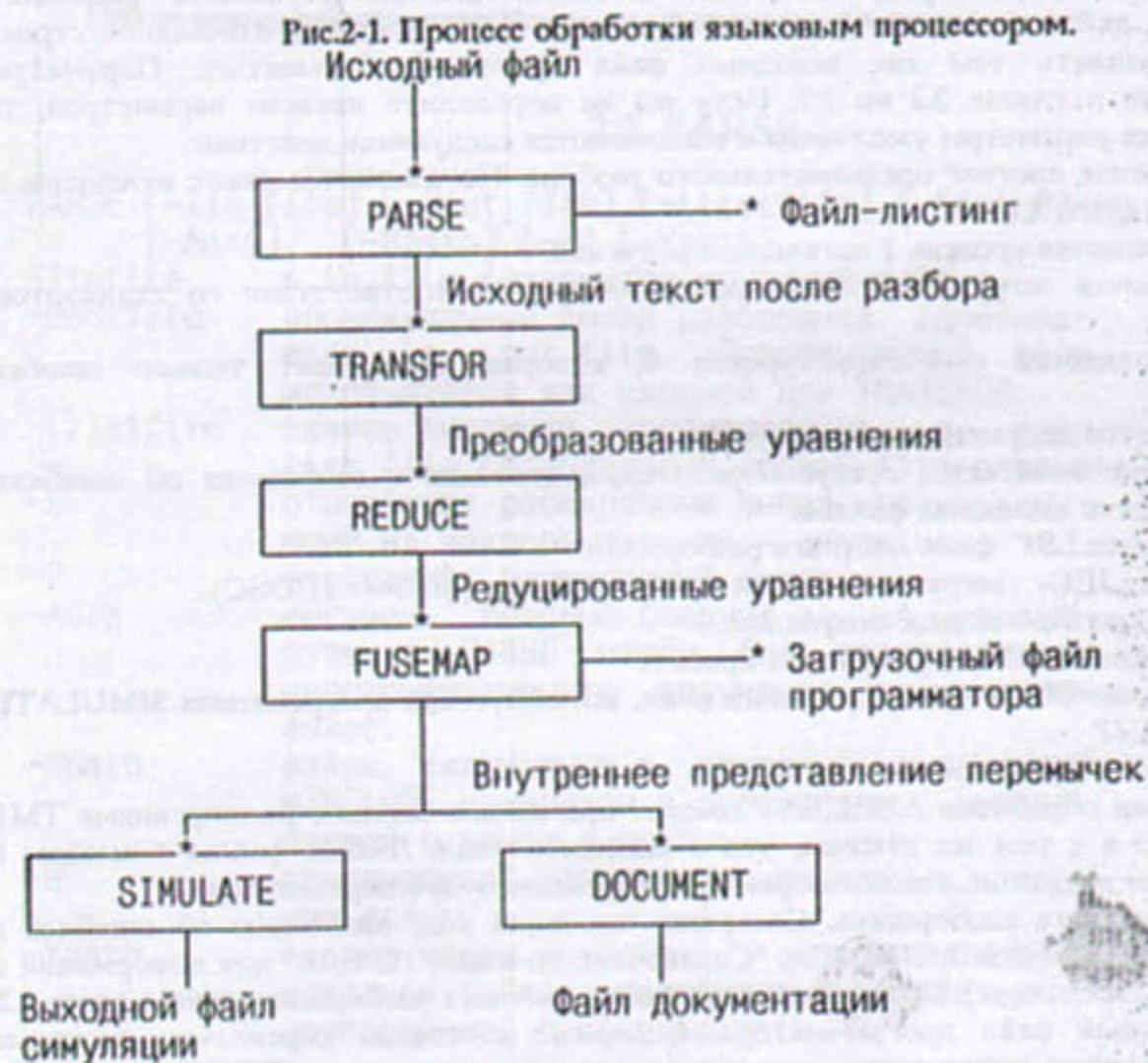
Языковой процессор АБЕЛЬ преобразует логическое описание в загрузочный файл программатора, который может быть загружен в программатор чтобы запрограммировать микросхему. Языковой процессор проверяет ваше логическое описание, выполняет логическую редукцию, симулирует операции запрограммированной микросхемы и создает документацию разработки.

Обработка исходного абелевского файла является шестиступенчатым процессом:

1. PARSE - Читает исходный файл, проверяет правильность синтаксиса, преобразует макросы, исполняет директивы.
2. TRANSFOR - Преобразует описание в промежуточную форму.
3. REDUCE - Выполняет логическую редукцию.
4. FUSEMAP - Создает загрузочный файл программатора.
5. SIMULATE - Симулирует функции запрограммированной микросхемы.
6. DOCUMENT - Создает документацию разработки.

Глава 2.1 описывает в деталях batch-файл ABEL. Главы 2.2 по 2.7 описывают индивидуальные ступени языкового процессора, их режимы и как их включать. Рисунок 2-1 показывает процесс обработки языкового процессора.

Все команды описанные в этой главе должны вводиться в соответствии с надлежащим синтаксисом. Если не указана спецификация для какого-либо входного или выходного файла, DOS будет искать файлы на текущем дисковом.





## 2.1. ABEL.BAT, АВТОМАТИЧЕСКАЯ ОБРАБОТКА

ABEL filename [parameters]

**filename** имя абелевского исходного файла. Расширение файла должно быть "ABL", но не должно вводиться.

**parameters** любой из параметров относящихся к индивидуальным ступеням языкового процессора (за исключением -I, -O или -B), как описано в главах 2.2 по 2.7. Разделяйте параметры пробелами.

ABEL.BAT является простым batch-файлом, который содержит команды чтобы запустить все шесть ступеней языкового процессора автоматически. Входной файл, который вы определяете "filename", должен быть осмысленным исходным файлом. Любые устройства (микросхемы) определенные в исходном файле должны поддерживаться АБЕЛем и файлы-спецификации соответствующих устройств должны находиться на текущем диске. Список поддерживаемых устройств дается в приложении В. Файлы-спецификации устройств поставляются на диске с пакетом АБЕЛЬ и имеют расширение DEV.

Любой из параметров, связанный с этими шестью ступенями языкового процессора, за исключением -I, -O или -B, может быть введен в командной строке чтобы управлять тем как исходный файл будет обрабатываться. Параметры обсуждаются в главах 2.2 по 2.7. Если вы не определите никаких параметров, то принимаются параметры умолчания и выполняются следующие действия:

- \* Создается листинг предварительного разбора. Показывается текст включенный директивой @INCLUDE.
- \* Выполняется уровень 1 логической редукции.
- \* Создается загрузочный файл программатора в соответствии со стандартом JEDEC.
- \* Выполняется симуляция уровня 0, которая показывает только ошибки симуляции.
- \* Создается документация разработки.

На экране печатается статус обработки, статистика и сообщения об ошибках. Создаются пять выходных файлов:

filename.LST файл листинга разборщика  
device.JED загрузочный файл программатора (формат JEDEC)  
filename.SIM выход симуляции  
filename.DOC документация проекта  
filename.OUT промежуточный файл, используемый программами SIMULATE

и DOCUMENT

**ВНИМАНИЕ!**

В течении обработки ABEL.BAT создает временные файлы с расширениями TM1, TM2 и TM3 и с тем же именем, что и исходный файл. Любые файлы с именем и расширением идентичными этим временным файлам будут перезаписаны.

Файл листинга разборщика. Содержит исходный код, сообщения об ошибках и эффект директив @INCLUDE. (См. "Справочник по языку АБЕЛЬ" для информации о макросах и директивах). Полное описание файла листинга разборщика дано в главе 2.2.

Загрузочный файл программатора. Содержит состояния переключек, тестовые вектора и другую информацию, как это определено стандартом JEDEC. Этот файл

загружается в программатор чтобы запрограммировать и протестировать микросхемы программируемой логики. Для каждой микросхемы, определенной в проекте, создается один загрузочный файл программатора.

Файл выхода симуляции. Выходная информация ступени симуляции, отображающая завершена ли симуляция успешно и, если нет, где различаются фактический результат от предсказанного. Смотрите главу 2.6. для большей информации.

Файл документации проекта. Содержит рисунок микросхемы и редуцированные логические уравнения. Этот файл создается ступенью DOCUMENT языкового процессора.

Промежуточный файл. Языковой процессор создает файл с именем filename.OUT который содержит промежуточные выходные данные, созданные программой FUSEMAP. Этот промежуточный файл может быть использован как входной программами SIMULATE и DOCUMENT (см. главы 2.6 и 2.7).

Следующая команда заставляет batch-файл ABEL.BAT обрабатывать исходный файл m6809a.abl и выполнять редукцию уровня 2 (PRESTO):

```
abel m6809.abl -r2
```

На диске будут созданы следующие файлы:

M6809A.LST	файл листинга
U09.JED	загрузочный файл программатора
M6809A.SIM	выход симуляции
M6809A.DOC	документация проекта
M6809A.OUT	промежуточный файл.

U09 является именем устройства, как определено в исходном файле.

## 2.2. PARSE

PARSE [-Iin\_file] [-Oout\_file] [-Llistfile] [-E] [-P] [-Aarg]...[-Hpath] [-W] [-Ypath]

**-Iinfile** в in\_file содержится исходный файл

**-Ooutfile** промежуточный выход разборщика записывается в out\_file. Промежуточный файл используется как входной для TRANSFOR.

**-Llistfile** запись листинга производится в файл list\_file (стандартный формат file\_name.ext).

**-E** отображает расширенный вывод, результирующий из макрорасширений и директив.

**-P** отображает расширенный вывод.

**-Aarg** аргумент, который следует за -A передается в PARSE чтобы быть подставленным вместо фиктивного аргумента в исходном файле.

**-Hpath** файлы, включаемые в исходный директивой @INCLUDE находятся в директории, определяемой с помощью path.

**-W** игнорировать незначимые символы в исходном файле.

**-Ypath** спецификации устройств, используемых в исходном файле, находятся в директории, определяемой с помощью path.



PARSE читает исходный файл, преобразует диаграммы состояний и таблицы истинности в булевские уравнения, транслирует тестовые вектора, расширяет макросы и проверяет корректность синтаксиса. Если обнаруживается какая-либо ошибка, на вашем мониторе отображается тип ошибки и примерное ее местонахождение. Сообщения об ошибках записываются также в файл листинга, если он создан (-L). Промежуточный файл записывается в out file если определено -O. Этот промежуточный файл является входным для TRANSFOR.

Все параметры являются не обязательными.

Пример:

```
parse -aP14L4 -aGND -im6809a.abl -am6809a.par -lm6809a.lst -e
```

Эта команда заставляет PARSE обрабатывать исходный файл m6809.abl, который содержит абелевское логическое описание. Выходные данные записываются в файл m6809.par и создается файл листинга m6809.lst. В файле содержится расширенный вывод, поскольку включен параметр -E. Для подстановки аргументов процессору передаются два аргумента: P14L4 и GND. Запрещенные управляющие символы вызовут ошибки, так как не использовано -W.

Заметим, что команда может быть введена маленькими буквами и что параметры могут быть определены в любом порядке. Каждый параметр обсуждается подробно ниже.

**-I:** Определение входного файла.

Используйте -I чтобы определить файл, содержащий исходный абелевский код. Если -I не используется, то считается что ввод должен производиться из вашего стандартного входного устройства (обычно с клавиатуры).

**-O:** Определение выходного файла.

Используйте -O чтобы определить имя выходного файла. Если выходной файл не определен, вывод будет производиться на ваше стандартное выходное устройство (обычно монитор). Выходной файл содержит разобранный исходный код и используется как входной в процессоре TRANSFOR (глава 23).

**-L:** Создание файла листинга.

-L означает что должен быть создан файл листинга. Файл содержащий листинг определяется непосредственно после параметра -L. Файл листинга содержит разобранный исходный код с сообщениями об ошибках (если там они есть), а также с макрорасширениями и включениями кодов директивами, если определен параметр -E. (Для информации касающейся макросов и директив смотрите Справочник по языку АБЕЛЬ). Если определен параметр -P, файл листинга показывает также директивы, которые вызвали включение кода. Если параметр -L не использован, то никакого файла листинга создано не будет.

**-E:** Показать расширенный вывод.

Параметр -E вызывает запись в файл листинга разобранного и расширенного исходного кода. (Для информации касающейся макросов и директив смотрите Справочник по языку АБЕЛЬ). Если параметр -E не определен, листинг будет содержать исходный файл каким он был до обработки плюс сообщения об ошибках: расширения текста не показываются. Если файл листинга не определен (опущен параметр -L), расширенный вывод будет отображен на вашем терминале.

**-P:** Показать расширенный вывод.

В дополнение к листинговой информации, предоставляемой параметром -E, параметр -P перечисляет директивы, которые добавляют код к исходному. Если параметр -L не включен для генерации файла листинга, директивы и расширенный вывод будут отображены на вашем терминале.

**-A:** Передать аргументы в исходный файл.

Параметр -A позволяет вам передать аргументы в исходный абелевский файл. Эти аргументы подставляются вместо фиктивных аргументов в исходном файле. Аргументов может быть определено так много, как это нужно, но каждому аргументу должен предшествовать параметр -A. Подстановка аргументов обсуждается подробнее в Справочнике по языку АБЕЛЬ.

**-H:** Определить путь для включаемых файлов.

Директива @INCLUDE, описанная в главе 4 Справочника по языку АБЕЛЬ, позволяет вам включить исходный текст из одного файла в другой исходный файл. По умолчанию считается что включаемые файлы находятся в директории умолчания; -H позволяет вам переопределить это значение умолчания и определить где включаемые файлы должны быть найдены. Определяйте дисковод и путь непосредственно после -H. Например, ниже определяется, что включаемые файлы должны быть найдены в директории "examples" на дисковом "B":

```
-Hb:\examples
```

Смотрите ваше руководство по операционной системе для большей информации о спецификациях пути и дисковода.

**-W:** Игнорировать запрещенные управляющие символы.

Использование -W заставляет PARSE игнорировать любые незначимые символы, которые иначе могли бы рассматриваться как ошибочные в исходном файле. Управляющие символы создаются обычно использованием последовательности управляющих клавиш на вашей клавиатуре.

**-Y:** Определить путь файлов-спецификаций микросхем.

Языковой процессор использует файлы-спецификации микросхем совместно с вашим исходным файлом чтобы правильно обработать проект. Параметр Y позволяет вам определить какой дисковод и какая директория содержит файлы микросхем. Определяйте путь непосредственно после -Y.

Пример:

```
-Yb:\examples
```

означает что файлы-спецификации микросхем находятся на дисковом "B" в субдиректории "examples". Если параметр -Y не использован, считается что файлы микросхем находятся в директории умолчания. Смотрите ваше руководство по операционной системе для большей информации о спецификациях пути и дисковода.

## 2.2.1. Файл листинга программы PARSE

Листинг 2-1 показывает файл листинга, созданный ступенью PARSE языкового процессора. Этот листинг был создан обработкой программой PARSE измененной версии исходного файла M6809.ABL. В исходном файле были сделаны изменения чтобы произвести две синтаксические ошибки.



Листинг 2.1. Файл листинга из программы PARSE с ошибками для M6809A.LBL

```

0001 module m6809a
0002 title '6809 memory decode
0003 Jean Designer Data I/O Corp Redmond WA 24 Feb 1984'
0004
0005 U09a device 'P14L4':
0006 A15,A14,A13,A12,A11,A10 pin 1,2,3,4,5,6;
0007 ROM1, IO, ROM2, DRAM pin 14, 15, 16, 17;
0008
0009 H, L, X = 1, 0, .X.;
0010 Address = [A15, A14, A13, A12, A11, A10, X, X, X, X, X, X, X, X];
0011
0012 equations
0013 !DRAM = (Address <= ^hDFFF)
0014
0015 !IO = (Address >= ^hE000) & (Address <= ^hE7FF);

Syntax error: ':' expected

0016
0017 !ROM2 = (Address >= ^hF000) & (Address <= ^hF7FF);
0018
0019 !ROM1 = (Address >= ^hF800);
0020
0021 test_vectors Address -> [ROM1, ROM2, IO, DRAM]

Syntax error: '(' expected

0022 ^h0000 -> [ H, H, H, L ]:
0023 ^h4000 -> [ H, H, H, L ]:
0024 ^h8000 -> [ H, H, H, L ]:
0025 ^hC000 -> [ H, H, H, L ]:
0026 ^hE000 -> [ H, H, L, H ]:
0027 ^hE800 -> [ H, H, H, H ]:
0028 ^hF000 -> [ H, L, H, H ]:
0029 ^hF800 -> [ L, H, H, H ]:
0030 end m6809a

```

Первой ошибкой является отсутствие точки с запятой в уравнении:

```
!DRAM = (Address >= ^hDFFF)
```

Второй ошибкой является отсутствие левой скобки в заголовке таблицы истинности:

```
test_vectors Address -> [ROM1, ROM2, IO, DRAM]
```

Эта команда преобразует разобранный исходный код содержащийся в b:\m6809.par (созданный PARSE) и создает выходной файл b:\m6809.lst.

Эти две ошибки отображаются на вашем мониторе и показываются в файле листинга. Отмечается также приблизительное место в котором может находиться ошибка. Отмечается и тип ошибки.

Листинг 2-2 показывает правильный исходный файл с правильно добавленными скобкой и точкой с запятой.

Листинг 2.2. Корректированный исходный файл M6809A.LBL

```

0001 module m6809a
0002 title '6809 memory decode
0003 Jean Designer Data I/O Corp Redmond WA 24 Feb 1984'
0004
0005 U09a device 'P14L4':
0006 A15,A14,A13,A12,A11,A10 pin 1,2,3,4,5,6;
0007 ROM1, IO, ROM2, DRAM pin 14, 15, 16, 17;
0008
0009 H, L, X = 1, 0, .X.;
0010 Address = [A15, A14, A13, A12, A11, A10, X, X, X, X, X, X, X, X];
0011
0012 equations
0013 !DRAM = (Address <= ^hDFFF);
0014
0015 !IO = (Address >= ^hE000) & (Address <= ^hE7FF);
0016
0017 !ROM2 = (Address >= ^hF000) & (Address <= ^hF7FF);
0018
0019 !ROM1 = (Address >= ^hF800);
0020
0021 test_vectors (Address -> [ROM1, ROM2, IO, DRAM])
0022 ^h0000 -> [ H, H, H, L ]:
0023 ^h4000 -> [ H, H, H, L ]:
0024 ^h8000 -> [ H, H, H, L ]:
0025 ^hC000 -> [ H, H, H, L ]:
0026 ^hE000 -> [ H, H, L, H ]:
0027 ^hE800 -> [ H, H, H, H ]:
0028 ^hF000 -> [ H, L, H, H ]:
0029 ^hF800 -> [ L, H, H, H ]:
0030 end m6809a

```

### 2.3. TRANSFOR

TRANSFOR [-Iin\_file] [-Oout\_file]

-Iinfile в in\_file содержится промежуточный вывод программы PARSE.  
 -Ooutfile вывод TRANSFOR записывается в файл out\_file.

TRANSFOR считывает уравнения сгенерированные программой PARSE и прорабатывает с ними следующее:

1. Заменяет группы эквивалентными уравнениями, не использующими группы.
2. Заменяет все операторы эквивалентными операциями используя только ИЕ, И, ИЛИ и ИСКЛЮЧАЮЩЕЕ ИЛИ.
3. ИЛИ с уравнениями вызывающими множественные присвоения одному идентификатору.
4. Выполняет простую логическую редукцию базирующуюся на следующих правилах:



Правило	Описание
A & 1 = A	A AND 1 = A
A & 0 = 0	A AND 0 = 0
A # 1 = 1	A OR 1 = 1
A # 0 = A	A OR 0 = A
A \$ 1 = !A	A XOR 1 = NOT A
A \$ 0 = A	A XOR 0 = A
A !\$ 1 = A	A XNOR 1 = A
A !\$ 0 = !A	A XNOR 0 = NOT A

Преобразованные уравнения могут затем быть сокращены программой REDUCE. Спецификации файлов, даваемые для in file и out file должны быть правильными файловыми спецификациями для вашей операционной системы.

Пример:

```
TRANSFOR -ib:m6809.par -ob:m6809.tsf
```

## 2.4. REDUCE

```
REDUCE [-Iin_file] [-Oout_file] [-Rn]
```

**-Iinfile** в in\_file содержится промежуточный вывод программы TRANSFOR.

**-Ooutfile** вывод REDUCE записывается в файл out\_file.

**-Rn** n=0, 1, 2 или 3, означает уровень редукции, который должен быть выполнен (по умолчанию -1).

n=0 : нет редукции  
n=1 : простая редукция  
n=2 : простая и PRESTO редукция  
n=3 : простая и PRESTO редукция по каждому выводу

REDUCE сокращает ваше логическое описание так чтобы меньшее число сумм произведений было использовано в микросхеме программируемой логики. REDUCE сокращает логические уравнения из in file и записывает сокращенные уравнения в файл определенный как out file. Уравнения в in file должны быть в форме производимой программой TRANSFOR. Если входной файл не определен, ввод будет осуществляться из вашего стандартного входного устройства (обычно с вашей клавиатуры). Если не определен выходной файл, вывод будет производиться на ваше стандартное выходное устройство (обычно на монитор).

Вы можете выбрать один из четырех уровней редукции: 0, 1, 2 или 3. Если вы не выберете уровень редукции посредством -R, будет выполняться первый уровень редукции.

Уровень редукции 0. Для уровня 0 не выполняется никакой редукции. Преобразованные уравнения проверяются на правильность выходов для определенной микросхемы и записываются в выходной файл.

Уровень редукции 1. Если вы выберете уровень редукции 1 (-R1), логические уравнения преобразуются в форму суммы произведений и затем редуцируются в соответствии со следующими основными правилами булевой алгебры:

Правило	Описание
!0 = 1	NOT 0 = 1
!1 = 0	NOT 1 = 0
A & 0 = 0	A AND 0 = 0
A & 1 = A	A AND 1 = A
A # 0 = A	A OR 0 = A
A # 1 = 1	A OR 1 = 1
A # A = A	A OR A = A
A & A = A	A AND A = A
A & !A = 0	A AND (NOT A) = 0
A # !A = 1	A OR (NOT A) = 1

Уровень редукции 1 устанавливается по умолчанию, если не выбрано никакого другого уровня директивой -R.

Уровень редукции 2. Уровень редукции 2 должен быть использован когда число сумм произведений, использованных чтобы выполнить функцию, близко или более числа сумм произведений доступных для данной микросхемы. Если вы выберете уровень редукции 2, то будет выполнена редукция уровня 1 и логическая редукция PRESTO. PRESTO является техникой логической редукции, разработанной Антонином Свободой, которая уменьшает число сумм входов и произведений.

Пример:

```
REDUCE -r2 -ib:m6809.tsf -ob:m6809.red
```

Эта команда выполняет простую и PRESTO редукции над преобразованными уравнениями, содержащимися в файле b:m6809.tsf (созданном программой TRANSFOR) и записывает сокращенные уравнения в файл с названием b:m6809.red.

Уровень редукции 3. Уровень редукции 3 уменьшает логику связанную с каждым выводом микросхемы по принципу вывод за выводом. Существует два основных преимущества редукции по выводам. Во-первых, редукция по выводам быстрее чем нормальная PRESTO редукция. Во-вторых, этот тип редукции хорошо подходит для таких микросхем как GALы, которые не разделяют произведения между выходами.

## 2.5. FUSEMAP

```
FUSEMAP [-Iin_file] [-Oout_file] [-Jpath] [-Cn] [-Dn] [-Kq]
```

**-Iinfile** в in\_file содержится промежуточный файл, созданный программой REDUCE.

**-Ooutfile** промежуточный выход из FUSEMAP записывается в out\_file.

**-Jpath** определяется дисковод и путь для записи загрузочного файла программатора.

**-Cn** параметр контрольной суммы, n может быть:  
0 опускать STX, ETX и контрольную сумму в загрузочном файле программатора.  
1 записать STX, ETX в загрузочный файл программатора, а контрольную сумму



опустить.

2 записать STX, ETX и контрольную сумму в загрузочном файле программатора.

-Dn определение формата загрузочного файла программатора, где "n" может быть равно 0, отмечая использование стандарта JEDEC для загрузочного файла программатора, или одного из микропроцессорных форматов, перечисленных в таблице 2-1.

-Kq неиспользованные переключки в матрице ИЛИ в IFL или FPLA должны быть оставлены соединенными или разъединенными.

q=Y оставить переключки соединенными.

q=N оставить переключки разъединенными.

FUSEMAP обрабатывает выходные данные программы REDUCE и создает загрузочные файлы программатора, который должен быть загружен в логический программатор, чтобы запрограммировать и протестировать микросхемы. Для каждого устройства определенного в оригинальном исходном файле создается один загрузочный файл. Загрузочный файл содержит состояния переключки для программирования микросхемы и тестовые вектора чтобы протестировать ее, когда она будет запрограммирована. FUSEMAP создает загрузочный файл в формате JEDEC, если вы не определите другой формат параметром -D.

Имя файла каждого загрузочного файла является именем устройства, для которого файл содержит карту переключки. Расширение файла является "JED" для загрузочных файлов в формате JEDEC и "Pxx" для всех других форматов загрузочного файла, где "xx" соответствует числу, определенному параметром -D. Загрузочный файл записывается на дисковод, определенный "Jdrive:" или на дисковод умолчания, если он не определен. FUSEMAP создает также промежуточный файл, определяемый с помощью "out file", который используется на стадиях DOCUMENT и SIMULATE.

Пример:

FUSEMAP -im6809a.red -om6809a.fus

Эта команда запускает FUSEMAP. Читается файл m6809a.red (вывод от REDUCE), по одному загрузочному файлу программатора создается для каждого устройства в исходном файле, и создается выходной файл m6809a.fus. Учитывая что в исходном файле были декларированы три устройства с именами DEVICE1, DEVICE2 и DEVICE3, FUSEMAP создаст следующие файлы:

DEVICE1.JED

DEVICE2.JED

DEVICE3.JED

Каждый файл является загрузочным файлом программатора сформированным в соответствии со стандартом JEDEC, который описывает логику и тестовые вектора, которые нужно запрограммировать в микросхеме.

-J: Определить путь для вывода загрузочного файла программатора.

Параметр -J позволяет вам отметить куда должен быть записан загрузочный файл программатора. Определяйте желаемый дисковод и/или путь после -J.

-C: Параметр контрольной суммы.

Обычно, в начале загрузочного файла программатора помещается STX, а ETX и контрольная сумма помещается в конец файла в соответствии со стандартом JEDEC.

Параметр -C позволяет вам управлять писать ли и каким образом в загрузочный файл программатора STX, ETX и контрольную сумму. -C0 опускает STX, ETX и контрольную сумму в загрузочном файле программатора. -C1 вызывает запись в загрузочный файл символов STX, ETX как обычно, но вместо контрольной суммы записывается 0000. -C2 является значением умолчания и вызывает запись STX, ETX и правильной контрольной суммы.

-D: Определение формата загрузочного файла программатора.

Параметр -D определяет формат загрузочного файла программатора. -D0 является значением умолчания и означает что загрузочный файл программатора должен быть в формате JEDEC. Другие форматы могут быть определены записью за "-D" надлежащего кода микропроцессорного формата (-d83, например). Поддерживаемые микропроцессорные форматы, их коды и расширения, даваемые загрузочным файлам программатора, даются в таблице 2-1.

ПРИМЕЧАНИЕ: Загрузочные файлы программатора не JEDEC формата не содержат тестовых векторов, даже если тестовые вектора были определены в исходном файле.

Таблица 2-1. Коды форматов данных и расширения файлов

Формат	Код	Расширение
Motorola Exorciser	82	.p82
Intel Intellec 8/MDS	83	.p83

-K: Параметр неиспользуемых переключки.

Неиспользуемые переключки в матрице ИЛИ для IFL или FPLA могут быть оставлены либо соединенными, либо разъединенными. Оставление неиспользуемых переключки несоединенными позволяет добавлять логические функции в устройстве или изменять существующий проект в нем. Однако, некоторое улучшение скорости и потребляемой мощности достигается разъединением всех неиспользуемых переключки. Разъединение переключки препятствует любым будущим модификациям устройства.

Параметр -K позволяет вам определить что делать с неиспользуемыми переключками. -Ky оставляет переключки соединенными. -Kn разъединяет неиспользуемые переключки. Если -K не определен, неиспользуемые переключки разъединены.

## 2.6. SIMULATE

SIMULATE [-Iin\_file] [-Oout\_file] [-Tn]  
[-Ndevice] [-Bn1.n2[.n3]] [-Xn] [-Zn]

-Iinfile в in\_file содержится промежуточный файл, созданный программой FUSEMAP или загрузочный файл программатора в формате JEDEC.

-Ooutfile выходной файл симуляции.

-Tn "n" является уровнем трассирования симуляции, где уровни могут быть:

0 : показывать только ошибки

1 : показывать тестовые вектора



- 2 : показывать выходы и тестовые вектора  
 3 : показывать полностью внутреннее содержание микросхемы
- По умолчанию устанавливается уровень трассирования 0, показываются только ошибки.
- Ndevice часть промышленного номера микросхемы. Используется когда симуляция выполняется независимо от АБЕЛЯ.
- Bn1,n2[n3] устанавливает точки разрыва "n1" и "n2" между которыми действует новый уровень трассирования "n3".
- Xn устанавливает безразличное значение используемое для ".X." в тестовых векторах загрузочного файла программатора. "n" может быть 1, 0, N. По умолчанию "n"=0.
- Zn устанавливает логическое значение используемое для ".Z." в тестовых векторах загрузочного файла программатора. "n" может быть 1, 0, N. По умолчанию "n"=1.

SIMULATE использует информацию о проекте и микросхеме чтобы симулировать операции запрограммированного устройства. SIMULATE может использовать либо информацию о проекте, созданную программой FUSEMAP, либо загрузочный файл программатора в стандарте JEDEC чтобы симулировать операции PAL или IFL. Промежуточный выход программы FUSEMAP должен быть использован чтобы симулировать ППЗУ.

SIMULATE не исполняет булевские уравнения и не прикладывает ко входам абелевские таблицы истинности и диаграммы состояний; эта программа имитирует операции микросхемы как будто бы уже запрограммированной информацией, которая содержится во входном файле. Если для SIMULATE используется в качестве входного загрузочный файл программатора, то должна быть предоставлена информация описывающая микросхему параметром -N. Информация о микросхеме содержится в файлах на дистрибутивном диске с расширением файла DEV и с частью промышленного номера в качестве имени. Если для симуляции используется выходной файл программы FUSEMAP, информация описывающая устройство уже содержится в файле и параметр -N не нужен.

Более того, если загрузочный файл программатора использован в качестве входного для SIMULATE, будут имитированы операции только одной микросхемы, связанной с этим загрузочным файлом. Если же в качестве входного файла использован выход FUSEMAP, симулироваться будут операции всех микросхем первоначально определенных в абелевском исходном файле.

Микросхемы с регистровыми выходами должны тактироваться прежде чем выход отразит изменение на входах. Тактовый импульс должен быть определен как один из входов в тестовых векторах для таких микросхем. Выбирая соответствующий уровень трассирования, вы можете увидеть только окончательные выходы для регистровых микросхем, или выходы перед и после тактового импульса. Смотрите главу 2.6.2 для примера симуляции выходов регистровой микросхемы.

Выходы микросхем с обратной связью не всегда могут быть определены одним вычислением функции выходов от входов, но могут потребовать нескольких вычислений, пока выходы стабилизируются. SIMULATE использует метод итераций

чтобы вычислить состояние выходов, вычисляя функцию повторно пока выходы не стабилизируются. Если выходы не стабилизируются после 20 итераций, сообщается об ошибке. Более подробная информация о симулировании микросхем с обратной связью представлена в главе 2.6.2.

Выходная информация, созданная программой SIMULATE, записывается в файл, определенный параметром -O, или на стандартное выходное устройство, если никакого выходного файла не задано. Содержимое этого файла будет очень сильно зависеть от уровня трассирования, задаваемого параметром -T. Смотрите в главах 2.6.1 и 2.6.2 примеры вывода при симуляции.

-T: установить уровень трассирования

-T определяет уровень информации, которая обеспечивается программой SIMULATE. Уровень трассирования может быть 0, 1, 2 или 3. Ошибки выводятся независимо от уровня трассирования.

Уровень 0 показывает конечные состояния выходов (после того как выходы установились) и тестовые вектора только для ошибок.

Уровень 1 показывает конечные состояния выходов и тестовые вектора для микросхемы.

Уровень 2 показывает состояния выходов для каждой итерации симулятора и тестовые вектора. Для регистровых элементов, выходы показаны до и после тактового импульса. Для разработок с обратными связями, выходы показываются для каждой итерации до тех пор, пока состояние выходов не установится.

Уровень 3 показывает состояния внутренних узлов и выходов микросхемы для каждой итерации в дополнении к тестовым векторам. Используйте уровень 3 для большей помощи при определении где и почему случились ошибки симуляции.

ПРИМЕЧАНИЕ: Разработки с ППЗУ могут быть симулированы только с уровнями трассирования 0 и 1.

-N: определить тип микросхемы, используемой для симуляции

Если для симуляции в качестве входного используется загрузочный файл программатора, а не выход FUSEMAP, никакая информация о микросхеме недоступна. -N определяет тип микросхемы, относящейся к этому загрузочному файлу, и информация о микросхеме считывается из файла спецификации микросхемы на дистрибутивном диске. Эта микросхема должна поддерживаться АБЕЛЕМ, и определена промышленным номером, следующим за параметром -N. Список поддерживаемых микросхем может быть найден в приложении В.

Пример:

```
simulate -iU09.jed -om6809a.sim -nP14L4
```

Пример выше запускает симулятор с файлом U09.jed в качестве входного, m6809a.sim в качестве выходного и P14L4.DEV как файла спецификации микросхемы. Если файл P14L4.DEV не может быть найден, будет выдано сообщение об ошибке.

ПРИМЕЧАНИЕ: -N и JEDEC файлы не могут быть использованы для симуляции ППЗУ.

-Bn1,n2[n3]: Установить точки останова

Может быть полезным, в частности в больших проектах, рассмотреть выход симуляции только для некоторых тестовых векторов. -B обеспечивает этот тип селективного трассирования. Должны быть определены начальная и конечная точки разрыва. Точка разрыва определяется номером тестового вектора, на котором разрыв имеет место. После второй точки разрыва уровень трассирования возвращается к своему значению до разрыва.



Новый уровень трассирования может быть определен параметром "n". Если новый уровень трассирования не определен, уровень трассирования между точками разрыва будет на один уровень выше, чем перед точкой разрыва.

Пример:

```
simulate -im6809a.fus -om6809a.sim -B5,8,3
```

Этот пример запускает симулятор с уровнем трассирования по умолчанию равным 0. На пятом тестовом векторе уровень трассирования устанавливается в 3 и остается таким до вектора 8, после которого уровень трассирования сбрасывается в уровень 0.

Пример:

```
simulate -t1 -im6809a.fus -om6809a.sim -b5,8
```

В этом случае точки разрыва снова охватывают вектора 5 и 8. Однако, начальный уровень трассирования в уровень 1 параметром -T и в точках разрыва не определяется никакого нового уровня. Новый уровень трассирования между векторами 5 и 8 (включительно) будет уровнем 2 (на один уровень выше, чем 1).

-X и -Z: Установить значения для состояний "безразличное" и "высокий импеданс"

Значения "безразличное" и "высокий импеданс" используемые при вычислениях тестовых векторов должны быть определены некоторым значением в течении симуляции. Параметры -X и -Z позволяют вам переопределить значения умолчания. По умолчанию для "X" подставляется значение L, а для "Z" подставляется H. Вы можете определить для "X" и "Z" значения 0, 1, L или H. Значения умолчания подставляются только когда "X" и "Z" используются как входы в проекте или как выходы, которые в данный момент используются как входы обратной связи. Выходы, которые не используются для обратной связи, показываются в выходном файле симулятора так, как они описаны в исходном файле, без изменения "X" и "Z".

## 2.6.1. ВЫХОДНОЙ ФАЙЛ ПРОГРАММЫ SIMULATE

Рисунок 2.2 показывает выход программы SIMULATE, созданный при обработке файла M6809A.ABL (листинг 2-2), с указателями на различные части выхода. В целях описания, один из тестовых векторов был изменен чтобы вызвать ошибку, и симуляция была запущена на уровне 0 (показываются только ошибки). Рисунок 2-3 показывает тестовые вектора, использованные для создания ошибок симуляции, в которых четвертый вектор был изменен. Если использованы оригинальные тестовые вектора, показанные в листинге 2-2, никаких ошибок симуляции не будет.

Рисунок 2-2. Выход симуляции для M6809A.ABL

```
ABEL Version 2.02a Data I/O Corp.
6809 memory decode
Jean Designer Data I/O Corp Redmond WA 24 Feb 1984

Simulate device U09a, type 'P14L4'

Output
V 0004 [110000000-000000000-] -> [.....HHHL...]
Test error [.....LHHL...]
```

14 ROM1. L expected

Рисунок 2-3. Тестовые вектора использованные чтобы создать ошибки симуляции

```
test_vectors (Address -> [ROM1,ROM2, IO, DRAM])
~h0000 -> [ H, H, H, L ]
~h4000 -> [ H, H, H, L ]
~h8000 -> [ H, H, H, L ]
~hC000 -> [ L, H, H, L ]
~hE000 -> [ H, H, L, H ]
~hE800 -> [ H, H, H, H ]
~hF000 -> [ H, L, H, H ]
~hF800 -> [ L, H, H, H ]
```

Выход симулятора показывает текущие выходные сигналы и тестовые вектора, использованные чтобы выполнить симуляцию. Текущие выходы, относящиеся к каждому тестовому вектору, показаны в строке, за которой следует строка с тестовым вектором. Тестовый вектор перечисляет входы слева от сочетания "->", и ожидаемый результат. Фактический результат для легкого сравнения показан справа от этого сочетания. Символ "N", встречающийся в тестовых векторах, означает что данный результат не тестируется. Точки "." помещаются в позициях тех выводов, которые не являются выходами.

Листинг 2-3 показывает листинг симуляции уровня 1 для того же самого исходного файла (включая неправильный вектор). Поскольку этот проект не имеет ни обратных связей, ни регистровых выходов, Уровень 2 симуляции не добавляет никакой дополнительной информации и здесь не показан.

Листинг 2.3. Выход симуляции для M6809A.ABL при уровне трассирования 1

```
ABEL Version 2.02a Data I/O Corp.
6809 memory decode
Jean Designer Data I/O Corp Redmond WA 24 Feb 1984

Simulate device U09a, type 'P14L4'

Vector 1
Output -> [.....HHHL...]
V 0001 [000000000-000000000-] -> [.....HHHL...]

Vector 2
Output -> [.....HHHL...]
V 0002 [010000000-000000000-] -> [.....HHHL...]

Vector 3
Output -> [.....HHHL...]
V 0003 [100000000-000000000-] -> [.....HHHL...]

Vector 4
Output -> [.....HHHL...]
V 0004 [110000000-000000000-] -> [.....LHHL...]
Test error
```







Листинг 2.4. Исходный файл: синхронная схема с обратной связью

```

module regfb
title 'Operation of the simulator on devices with feedback
      Data I/O Corp      24 Feb 1984'
      FB1      device 'P16R4';
      pin 1,11;
      pin 2,3,4,5;
      pin 14,13;

      Clk,OE
      INIT,D1,D2,D3
      F1,F2

equations
!F1 := D1 & INIT;
!F2 = D2 & !F1;
ENABLE F2 = D3;

test_vectors ([Clk,OE,INIT,D1,D2,D3] -> [ F1, F2])
[.C., 0, 0, 1, 1, 1] -> [ 1, 1 ];
[.C., 1, 0, 0, 0, 0] -> [.Z.,.Z.];
[.C., 1, 1, 1, 1, 1] -> [.Z., 0 ];
[ 0, 0, 0, 0, 0, 0] -> [ 0, .Z.];

end regfb
    
```

Листинг 2.5. Выход симуляции при уровне трассирования 1: синхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
      Data I/O Corp      24 Feb 1984

Simulate device FB1, type 'P16R4'
    
```

```

Vector 1
Output      -> [.....ZH.....]
V 0001 [C01110000-000000000-] -> [.....NH.....]

Vector 2
Output      -> [.....ZZ.....]
V 0002 [C00000000-100000000-] -> [.....NZ.....]

Vector 3
Output      -> [.....ZL.....]
V 0003 [C11110000-100000000-] -> [.....NL.....]

Vector 4
Output      -> [.....ZZL.....]
V 0004 [000000000-000000000-] -> [.....NZL.....]
    
```

Листинг 2.6. Выход симуляции при уровне трассирования 2: синхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
      Data I/O Corp      24 Feb 1984

Simulate device FB1, type 'P16R4'
    
```

```

Vector 1
Output      -> [.....ZH.....]
Clock      [.....CC.....]
V 0001 [C01110000-000000000-] -> [.....NH.....]

Vector 2
Output      -> [.....ZZ.....]
Clock      [.....CC.....]
V 0002 [C00000000-100000000-] -> [.....NZ.....]

Vector 3
Output      -> [.....ZH.....]
Clock      [.....CC.....]
Output      -> [.....ZL.....]
V 0003 [C11110000-100000000-] -> [.....NL.....]

Vector 4
Output      -> [.....ZZL.....]
V 0004 [000000000-000000000-] -> [.....NZL.....]
    
```

Листинг 2.7. Выход симуляции при уровне трассирования 3: синхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
      Data I/O Corp      24 Feb 1984
    
```

----- FB1 -----

type 'P16R4'

Vector 1  
Input Reg[.....]

Pin	Name	Output Enable	First Fuse	Product Terms
12		N	1824	F F F F F F F F
13	F2	Y	1568	F F F F F F F F
14	F1	Y	1280	F F F F F F F F
15		Y	1024	F F F F F F F F
16		Y	768	F F F F F F F F
17		Y	512	F F F F F F F F
18		N	288	F F F F F F F F
19		N	32	F F F F F F F F

-----



```

OR gate      -> { ..... LLLLLLLL. }
Register     -> { ..... LLLL... }
Output       -> { ..... ZHHHHHZZ. }
Clock        -> { ..... CCCC... }
V 0001 [C01110000-000000000-] -> { ..... NHHNNNNN. }

```

Vector 2

Pin	Name	Output Enable	First Fuse	Product Terms
12		N	1824	F F F F F F F
13	F2	N	1568	F F F F F F F
14	F1	N	1280	F F F F F F F F
15		N	1024	F F F F F F F F
16		N	768	F F F F F F F F
17		N	512	F F F F F F F F
18		N	288	F F F F F F F
19		N	32	F F F F F F F

```

OR gate      -> { ..... LLLLLLLL. }
Register     -> { ..... LLLL... }
Output       -> { ..... ZZZZZZZZ. }
Clock        -> { ..... CCCC... }
V 0002 [C00000000-100000000-] -> { ..... NZZNNNNN. }

```

Vector 3

Pin	Name	Output Enable	First Fuse	Product Terms
12		N	1824	F F F F F F F
13	F2	Y	1568	F F F F F F F
14	F1	N	1280	T F F F F F F F
15		N	1024	F F F F F F F F
16		N	768	F F F F F F F F
17		N	512	F F F F F F F F
18		N	288	F F F F F F F
19		N	32	F F F F F F F

```

OR gate      -> { ..... LLHLLLLL. }
Register     -> { ..... LLLL... }
Output       -> { ..... ZHZZZZZZ. }
Clock        -> { ..... CCCC... }

```

Pin	Name	Output Enable	First Fuse	Product Terms
12		N	1824	F F F F F F F
13	F2	Y	1568	T F F F F F F

Pin	Name	Output Enable	First Fuse	Product Terms
14	F1	N	1280	T F F F F F F F
15		N	1024	F F F F F F F F
16		N	768	F F F F F F F F
17		N	512	F F F F F F F F
18		N	288	F F F F F F F
19		N	32	F F F F F F F

```

OR gate      -> { ..... LHLLLLLL. }
Register     -> { ..... HLLL... }
Output       -> { ..... ZLZZZZZZ. }
V 0003 [C11110000-100000000-] -> { ..... NLZNNNNN. }

```

Vector 4

Pin	Name	Output Enable	First Fuse	Product Terms
12		N	1824	F F F F F F F F
13	F2	N	1568	F F F F F F F F
14	F1	Y	1280	F F F F F F F F F
15		Y	1024	F F F F F F F F F
16		Y	768	F F F F F F F F F
17		Y	512	F F F F F F F F F
18		N	288	F F F F F F F F
19		N	32	F F F F F F F F

```

OR gate      -> { ..... LLLLLLLL. }
Register     -> { ..... HLLL... }
Output       -> { ..... ZZLHHHZZ. }
V 0004 [000000000-000000000-] -> { ..... NZLNNNNN. }

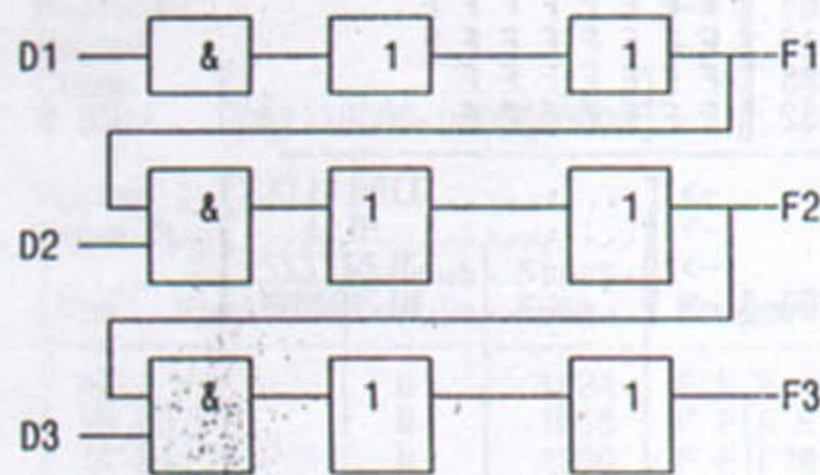
```

В качестве второго примера рисунок 2-6 показывает асинхронную схему, которая требует более чем одну итерацию при симуляции перед тем как выходы установятся. Листинг 2-8 показывает исходный файл описывающий схему, а листинги 2-9 по 2-11 показывают вывод симуляции для уровней трассирования с 1 по 3. Вывод при уровне трассирования 0 не показан, потому что никаких ошибок в нашем проекте нет.

Уровень трассирования 1 показывает конечные состояния выходов после того как они установятся, и тестовые вектора. Уровень трассирования 1 показывает значения выходов при различных итерациях, когда выходы устанавливаются, а также конечные состояния выходов и тестовые вектора. Заметьте, что для входных данных в векторе 2 требуется три итерации, прежде чем выходы установятся. Вектор 1 требует только одну итерацию чтобы обеспечить установившееся значение выходов. Уровень трассирования 3 содержит таблицу выводов, выходы вентилях ИЛИ, выходы регистров и состояния выходов для каждой итерации, пока выходы не установятся.



Рис. 2.6. Асинхронные схемы с обратной связью



Листинг 2-8. Исходный файл: асинхронная схема с обратной связью

```

module feedback
title
'Operation of the simulator on devices with feedback
DATA I/O Corp. 24 Feb 1983'
FB1 device 'P16HD8';
D1,D2,D3 pin 1,2,3;
F1,F2,F3 pin 13,14,15;
equations
F1 = D1;
F2 = D2 & F1;
F3 = D3 & F2;
test_vectors (([D1,D2,D3] -> [F1,F2,F3]))
[ 0, 0, 0] -> [ 0, 0, 0];
[ 1, 1, 1] -> [ 1, 1, 1];
end feedback
    
```

Листинг 2.9. Выход симуляции при уровне трассирования 1: асинхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
DATA I/O Corp. 24 Feb 1983

Simulate device FB1, type 'P16HD8'

Vector 1
Output -> [.....LLLLLLLLL.]
V 0001 [000000000-000000000-] -> [.....NLLLNNNN.]

Vector 2
Output -> [.....LHHHLLLLL.]
V 0002 [111000000-000000000-] -> [.....NHHHNNNN.]
    
```

Листинг 2.10. Выход симуляции при уровне трассирования 2: асинхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
DATA I/O Corp. 24 Feb 1983

Simulate device FB1, type 'P16HD8'

Vector 1
Output -> [.....LLLLLLLLL.]
V 0001 [000000000-000000000-] -> [.....NLLLNNNN.]

Vector 2
Output -> [.....LHLLLLLLL.]
Output -> [.....LHHLLLLLL.]
Output -> [.....LHHHLLLLL.]
V 0002 [111000000-000000000-] -> [.....NHHHNNNN.]
    
```

Листинг 2.11. Выход симуляции при уровне трассирования 3: асинхронная схема с обратной связью

```

ABEL Version 2.02a Data I/O Corp.
Operation of the simulator on devices with feedback
DATA I/O Corp. 24 Feb 1983

----- FB1 -----
type 'P16HD8'

Vector 1
Input Reg[.....]


| Pin | Name | Output Enable | First Fuse | Product Terms   |
|-----|------|---------------|------------|-----------------|
| 12  |      | Y             | 1792       | F F F F F F F F |
| 13  | F1   | Y             | 1536       | F F F F F F F F |
| 14  | F2   | Y             | 1280       | F F F F F F F F |
| 15  | F3   | Y             | 1024       | F F F F F F F F |
| 16  |      | Y             | 768        | F F F F F F F F |
| 17  |      | Y             | 512        | F F F F F F F F |
| 18  |      | Y             | 256        | F F F F F F F F |
| 19  |      | Y             | 0          | F F F F F F F F |


-----
OR gate -> [.....LLLLLLLLL.]
Register -> [.....]
Output -> [.....LLLLLLLLL.]
V 0001 [000000000-000000000-] -> [.....NLLLNNNN.]
    
```







Если никаких параметров не передано программе DOCUMENT, то никакого листинга выведено не будет.

-Q: параметр вывода уравнений.

Параметр -Q управляет типом уравнений, которые должны появиться в выходном файле документации. До трех цифр (0,1,2) может быть определено вслед за -Q, либо флаг -Q может быть повторен с различными номерами уравнений, которые желательно вывести. Например, как "-q02" так и "-q0 -q2" вызывают включение в выходной файл документации первоначальных и редуцированных уравнений.

-F: включение карты переключек и термов

Параметр -F управляет включением в файл карты переключек и термов. -F0 или -F включает полную карту переключек и числа использованных термов. -F1 включает только число использованных термов.

### 2.7.1. ВЫХОДНОЙ ФАЙЛ ПРОГРАММЫ DOCUMENT

Листинг 2-12 показывает вывод из программы DOCUMENT. Этот вывод был создан обработкой исходного файла M6809A.ABL (листинг 2-2) при использовании параметров -V, -F, -G, -S и -Q2. Листинг содержит таблицу символов, редуцированные уравнения, диаграмму микросхемы, карту переключек и тестовые вектора. В карте переключек нетронутые соединения показаны как "X", а расплавленные переключки (нет соединения) показаны черточками. Входы тестовых векторов показаны слева от "->"; выходы показаны справа.

Листинг 2.12. Вывод документации для m6809a.abl

ABEL(tm) Version 2.02a - Document Generator 31-Dec-91 01:04 PM  
 6809 memory decode  
 Jean Designer Data I/O Corp Redmond WA 24 Feb 1984  
 Symbol list for Module m6809a

A10	Pin 6	pos. com
A11	Pin 5	pos. com
A12	Pin 4	pos. com
A13	Pin 3	pos. com
A14	Pin 2	pos. com
A15	Pin 1	pos. com
Address	([A15,A14,A13,A12,A11,A10,..X...X...X... ..X...X...X...X...X...X...X...])	
DRAM	Pin 17	neg. com
H	(1)	
IO	Pin 15	neg. com
L	(0)	
ROM1	Pin 14	neg. com
ROM2	Pin 16	neg. com
U09a	device P14L4	
X	(.X.)	
m6809a	Module Name	

ABEL(tm) Version 2.02a - Document Generator 31-Dec-91 01:04 PM  
 6809 memory decode  
 Jean Designer Data I/O Corp Redmond WA 24 Feb 1984  
 Equations for Module m6809a

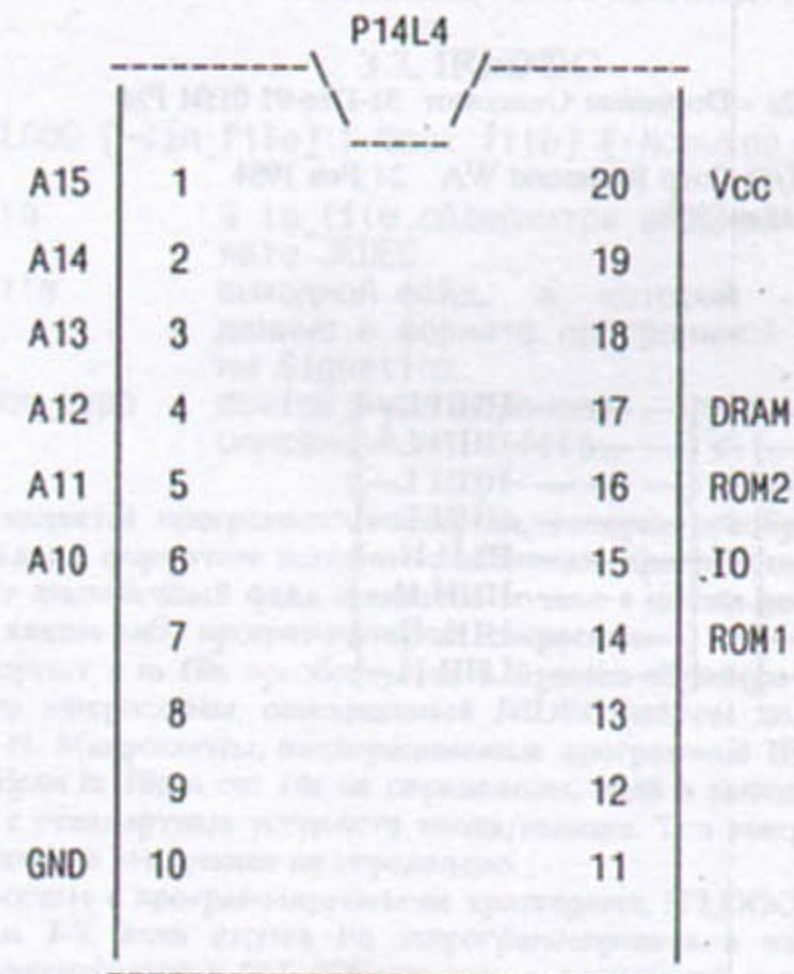
Device U09a

Reduced Equations:

DRAM = !(A13 # A14 # A15);  
 IO = !(A11 & A12 & A13 & A14 & A15);  
 ROM2 = !(A11 & A12 & A13 & A14 & A15);  
 ROM1 = !(A11 & A12 & A13 & A14 & A15);

ABEL(tm) Version 2.02a - Document Generator 31-Dec-91 01:04 PM  
 6809 memory decode  
 Jean Designer Data I/O Corp Redmond WA 24 Feb 1984  
 Chip diagram for Module m6809a

Device U09a





ABEL(tm) Version 2.02a - Document Generator 31-Dec-91 01:04 PM  
6809 memory decode  
Jean Designer Data I/O Corp Redmond WA 24 Feb 1984  
Fuse Map for Module m6809a

Device U09a

```

      0      10      20
0:  ---X---
28: -X-----
56: --X-----
84: XXXXXXXXXXXXXXXXXXXXXXX
112: X-X-X-X- --X-----
140: XXXXXXXXXXXXXXXXXXXXXXX
168: XXXXXXXXXXXXXXXXXXXXXXX
196: XXXXXXXXXXXXXXXXXXXXXXX
224: X-X-X-X- --X-----
252: XXXXXXXXXXXXXXXXXXXXXXX
280: XXXXXXXXXXXXXXXXXXXXXXX
308: XXXXXXXXXXXXXXXXXXXXXXX
336: X-X-X-X- --X-----
364: XXXXXXXXXXXXXXXXXXXXXXX
392: XXXXXXXXXXXXXXXXXXXXXXX
420: XXXXXXXXXXXXXXXXXXXXXXX

```

ABEL(tm) Version 2.02a - Document Generator 31-Dec-91 01:04 PM  
6809 memory decode  
Jean Designer Data I/O Corp Redmond WA 24 Feb 1984  
Test Vectors for Module m6809a

Device U09a

```

1 [0000 00-- ---] -> [--- --- -HHH L--]
2 [0100 00-- ---] -> [--- --- -HHH L--]
3 [1000 00-- ---] -> [--- --- -HHH L--]
4 [1100 00-- ---] -> [--- --- -HHH L--]
5 [1110 00-- ---] -> [--- --- -HLH H--]
6 [1110 10-- ---] -> [--- --- -HHH H--]
7 [1111 00-- ---] -> [--- --- -HHH H--]
8 [1111 10-- ---] -> [--- --- -LHH H--]

```

end of module m6809a

## ГЛАВА 3 УТИЛИТЫ

В этой главе описываются утилиты, поставляемые в дополнение к программному обеспечению ABEL.

### 3.1. TOABEL, ПРЕОБРАЗОВАТЕЛЬ PALASM В ABEL

```

TOABEL -Iinfile -Ooutfile
        -Iinfile  входной файл в стандарте PALASM
        -Ooutfile  выходной файл в стандарте ABEL

```

TOABEL является утилитой конверсии, которую вы можете использовать чтобы преобразовать логическое описание на языке PALASM в логическое описание на языке ABEL. TOABEL создает абелевский исходный файл, который может быть обработан обычным образом языковым процессором АБЕЛЬ. Это означает, что проект может быть редуцирован, симулирован (если есть тестовые вектора), документирован и загружен в логический программатор так же как любой другой исходный абелевский текст.

Заметьте, что тестовые вектора для микросхем с программируемыми входами/выходами (например P16L8 (1556XJ18)) могут потребовать редактирования, так как TOABEL не может определить, является ли вывод входом или выходом.

### 3.2. IFLDOC

```
IFLDOC [-Iin_file] [-Oout_file] [-Ndevice_type]
```

```

-Iinfile      в in_file содержится входной файл в формате JEDEC.
-Ooutfile     выходной файл, в который записываются данные в формате программной таблицы фирмы Signetics.
-Ndevice_type device_type отмечает тип микросхемы, описанной в in_file.

```

IFLDOC является программой конверсии, которая преобразует файлы формата JEDEC в файлы с форматом подобным таблицам программируемой логики фирмы Signetics. Этот листинговый файл создается только с целью документирования; он не используется каким-либо программатором микросхем.

JEDEC-формат в in file преобразуется в Signetics-образную форму и записывается в out file. Тип микросхемы, описываемый JEDEC-файлом должен быть определен параметром -N. Микросхемы, поддерживаемые программой IFLDOC, перечислены в таблице 3-1. Если in file и out file не определены, ввод и вывод будут приниматься и передаваться с стандартных устройств ввода/вывода. Тип микросхемы должен быть определен; значение умолчания не определено.

Для микросхем с программируемыми триггерами, IFLDOC считает что триггера являются типа J-K если строка Fo запрограммирована и индивидуальный режим триггера устанавливается в "A". (Обратитесь к логической схеме PLIM для лучшего понимания вышесказанного).



Таблица 3-1. Устройства, поддерживаемые программой IFLDOC

Signetics  
 F82S100 F82S153 F82S161  
 F82S103 F82S155 F82S162  
 F82S105 F82S157 F82S163  
 F82S151 F82S159 F82S167

Texas Instruments  
 PL839

Пример:  
 ifldoc -ifl2.jed -oifl2.tab -nf82s100

Эта команда запускает IFLDOC чтобы преобразовать данные из формате JEDEC содержащиеся в файле ifl2.jed, в формат таблицы программирования записанный в файл ifl2.tab. Параметр -N определяет, что микросхема, описываемая файлом ifl2.jed, является F82S100 (556PT1). Выходной файл, полученный в результате этой команды, показан на рис. 3-1.

Рисунок 3.1. Пример выходного файла из IFLDOC

Input file: IFL2.JED

T E R M	82S100								POLARITY															
	AND INPUTS (Im)				OR OUTPUTS (Fp)				H H H H H H H H				H H H H H H H H											
	1	1	1	1	1	1			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	A	.	.	.	.	.	.	.
1	-	-	-	-	-	-	-	-	H	L	-	-	-	-	-	-	.	A	.	.	.	.	.	.
2	-	-	-	-	-	-	-	-	L	H	-	-	-	-	-	-	.	A	.	.	.	.	.	.
3	-	-	-	-	-	-	-	-	H	L	-	-	-	-	-	-	.	.	A	.	.	.	.	.
4	-	-	-	-	-	-	-	-	L	H	-	-	-	-	-	-	.	.	A	.	.	.	.	.
5	-	-	-	-	-	-	-	-	-	H	L	-	-	-	-	-	.	.	.	A	.	.	.	.
6	-	-	-	-	-	-	-	-	-	L	H	-	-	-	-	-	.	.	.	A	.	.	.	.
7	-	-	-	-	-	-	-	-	-	-	H	L	-	-	-	-	.	.	.	.	A	.	.	.
8	-	-	-	-	-	-	-	-	-	-	L	H	-	-	-	-	.	.	.	.	A	.	.	.
9	-	-	-	-	-	-	-	-	-	-	-	H	L	-	-	-	.	.	.	.	.	A	.	.
10	-	-	-	-	-	-	-	-	-	-	-	L	H	-	-	-	.	.	.	.	.	A	.	.
11	-	-	-	-	-	-	-	-	-	-	-	-	H	L	-	-	.	.	.	.	.	.	A	.
12	-	-	-	-	-	-	-	-	-	-	-	-	L	H	-	-	.	.	.	.	.	.	.	A
13	-	-	-	-	-	-	-	-	-	-	-	-	-	H	L	-	.	.	.	.	.	.	.	A
14	-	-	-	-	-	-	-	-	-	-	-	-	L	H	-	-	.	.	.	.	.	.	.	A
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.

21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.	.	.	.	.	.



## ПРИЛОЖЕНИЕ А ABEL.BAT

Ниже приводится листинг ABEL.BAT, batch-файла, включенного в пакет АБЕЛЬ. Вы можете модифицировать этот batch-файл, если сочтете нужным.

Листинг А.1. "Batch" файл.

```
REM ABEL Version 2.0 for MS-DOS 2.0 (XT) Oct 11 1984
echo off
\label\system\PARSE -i%1.abl -o%1.tm1 -l%1.lst -y\abel\device %2 %3
%4 %5 %6 %7 %8 %9
if errorlevel 1 goto errparse
\label\system\TRANSFOR -i%1.tm1 -o%1.tm2
if errorlevel 1 goto errtrans
if exist %1.tm1 del %1.tm1
\label\system\REDUCE -i%1.tm2 -o%1.tm3 %2 %3 %4 %5 %6 %7 %8 %9
if errorlevel 1 goto erreduce
if exist %1.tm2 del %1.tm2
\label\system\FUSEMAP -i%1.tm3 -o%1.out %2 %3 %4 %5 %6 %7 %8 %9
if errorlevel 1 goto errfuse
if exist %1.tm3 del %1.tm3
\label\system\SIMULATE -i%1.out -o%1.sim %2 %3 %4 %5 %6 %7 %8 %9
if errorlevel 1 goto errsim
\label\system\DOCUMENT -i%1.out -o%1.doc -g -q2 %2 %3 %4 %5 %6 %7 %8
%9
if errorlevel 1 goto errdoc
goto exit
:errparse
echo Error(s) in PARSE
goto exit
:errtrans
\label\system\DOCUMENT -i%1.tm1 -o%1.doc -g -q0 %2 %3 %4 %5 %6 %7
%8 %9
echo Error(s) in TRANSFOR
goto exit
:erreduce
\label\system\DOCUMENT -i%1.tm2 -o%1.doc -g -q1 %2 %3 %4 %5 %6 %7 %8
%9
echo Error(s) in REDUCE
goto exit
:errfuse
\label\system\DOCUMENT -i%1.tm3 -o%1.doc -g -v -q2 %2 %3 %4 %5 %6 %7
%8 %9
echo Error(s) in FUSEMAP
goto exit
:errsim
\label\system\DOCUMENT -i%1.out -o%1.doc -f -g -v -q2 %2 %3 %4 %5 %6
%7 %8 %9
echo Error(s) in SIMULATE
goto exit
:errdoc
echo Error(s) in DOCUMENT
:exit
```

## ПРИЛОЖЕНИЕ В ПОДДЕРЖИВАЕМЫЕ МИКРОСХЕМЫ

Таблица В-1. Поддерживаемые ПЛИМ, ПМЛ, ППЗУ

ABEL	ALT	AMD	CYP	HAR	HMI	NAT	SIG	TI	VTI
P6L16	-	-	-	-	-	-	-	-	-
P8L14	-	-	-	-	-	-	-	-	-
P10H8	-	-	-	-	10H8	10H8	-	-	-
P10L8	-	-	-	-	10L8	10L8	-	-	-
P10P8	-	-	-	-	10P8	-	-	-	-
P12H6	-	-	-	-	12H6	12H6	-	-	-
P12L6	-	-	-	-	12L6	12L6	-	-	-
P12P6	-	-	-	-	12P6	-	-	-	-
P12H10	-	-	-	-	12H10	-	-	12H10	-
P12L10	-	-	-	-	12L10	12L10	-	12L10	-
P12P10	-	-	-	-	12P10	-	-	12P10	-
P14H4	-	-	-	-	14H4	14H4	-	-	-
P14L4	-	-	-	-	14L4	14L4	-	-	-
P14P4	-	-	-	-	14P4	-	-	-	-
P14H8	-	-	-	-	14H8	-	-	-	-
P14L8	-	-	-	-	14L8	14L8	-	14L8	-
P14P8	-	-	-	-	14P8	-	-	14P8	-
P16C1	-	-	-	-	16C1	16C1	-	-	-
P16CP1	-	-	-	-	16CP1	-	-	-	-
P16H2	-	-	-	-	16H2	16H2	-	-	-
P16L2	-	-	-	-	16L2	16L2	-	-	-
P16P2	-	-	-	-	16P2	-	-	-	-
P16H6	-	-	-	-	16H6	-	-	16H6	-
P16L6	-	-	-	-	16L6	16L6	-	16L6	-
P16P6	-	-	-	-	16P6	-	-	16P6	-
P16H8	-	16H8	-	77215	-	-	-	-	-
P16L8	-	16L8	16L8	77209	16L8	16L8	-	16L8	-
P16P8	-	-	-	16PC8	16P8	-	-	-	-
P16P8H	-	-	-	77216	-	-	-	-	-
P16HD8	-	16HD8	-	-	-	-	-	-	-
P16LD8	-	16LD8	-	-	-	-	-	-	-
P18H4	-	-	-	-	18H4	-	-	18H4	-
P18L4	-	-	-	-	18L4	18L4	-	18L4	-
P18P4	-	-	-	-	18P4	-	-	18P4	-
P18P8	-	18P8	-	-	-	-	-	-	-
P19L8	-	-	-	-	-	-	-	19L8	-
P19L8L	-	-	-	-	-	-	-	19L8	-
P20C1	-	-	-	-	20C1	20C1	-	-	-



ABEL	ALT	AMD	CYP	HAR	MHI	NAT	SIG	TI	VTI
P20CP1	-	-	-	-	20CP1	-	-	-	-
P20H2	-	-	-	-	20H2	-	-	20H2	-
P20L2	-	-	-	-	20L2	20L2	-	20L2	-
P20P2	-	-	-	-	20P2	-	-	20P2	-
P20X4	-	-	-	-	20X4	20X4	-	20X4	-
P20L8	-	-	-	-	20L8	-	-	20L8	-
P20P8E	-	-	-	-	20P8E	-	-	-	-
P20X8	-	-	-	-	20X8	20X8	-	20X8	-
P20L10	-	-	-	-	20L10	20L10	-	20L10	-
P20S10	-	-	-	-	20S10	-	-	-	-
P20X10	-	-	-	-	20X10	20X10	-	20X10	-
P16R4	-	16R4	16R4	16RC4	16R4	16R4	-	16R4	-
P16RP4	-	-	-	-	16RP4	-	-	-	-
P16R6	-	16R6	16R6	16RC6	16R6	16R6	-	16R6	-
P16RP6	-	-	-	-	16RP6	-	-	-	-
P16R8	-	16R8	16R8	16RC8	16R8	16R8	-	16R8	-
P16RP8	-	-	-	-	16RP8	-	-	-	-
P19R4	-	-	-	-	-	-	-	19R4	-
P19R4L	-	-	-	-	-	-	-	19R4	-
P19R6	-	-	-	-	-	-	-	19R6	-
P19R6L	-	-	-	-	-	-	-	19R6	-
P19R8	-	-	-	-	-	-	-	19R8	-
P19R8L	-	-	-	-	-	-	-	19R8	-
P20R4	-	-	-	-	20R4	-	-	20R4	-
P20RS4	-	-	-	-	20RS4	-	-	-	-
P20R6	-	-	-	-	20R6	-	-	20R6	-
P20R8	-	-	-	-	20R8	-	-	20R8	-
P20RS8	-	-	-	-	20RS8	-	-	-	-
P20RS10	-	-	-	-	20RS10	-	-	-	-
P22V10	-	22V10	-	-	-	-	-	-	-
P32R16	-	-	-	-	32R16	-	-	-	-
P32R16C	-	-	-	-	32R16	-	-	-	-
F82S100	-	-	-	-	-	-	82S100	-	-
F82S103	-	-	-	-	-	-	82S103	-	-
F82S105	-	-	-	-	-	-	82S105	-	-
F82S151	-	-	-	-	-	-	82S151	-	-
F82S153	-	-	-	77153	-	-	82S153	-	-
F82S157	-	-	-	-	-	-	82S157	-	-
F82S157	-	-	-	-	-	-	82S157	-	-
F82S159	-	-	-	-	-	-	82S159	-	-
F82S161	-	-	-	-	-	-	82S161	-	-
F82S167	-	-	-	-	-	-	82S167	-	-
FL839	-	-	-	-	-	-	-	74PL839	-
EP300	EP300	-	-	-	-	-	-	-	-
E10P8	-	-	-	-	-	-	-	-	EPL10P8
E12P6	-	-	-	-	-	-	-	-	EPL12P6

ABEL	ALT	AMD	CYP	HAR	MHI	NAT	SIG	TI	VTI
E14P4	-	-	-	-	-	-	-	-	EPL14P4
E16P2	-	-	-	-	-	-	-	-	EPL16P2
RA5P8	-	27S19	-	7603	6331	74S288	82S123	18S030	-
RA8P4	-	27S21	-	7611	6301	74S287	82S129	24S10	-
RA8P8	-	-	-	7625	6309	74S471	82S114	18S22	-
RA9P4	-	27S13	-	7621	6306	74S571	82S1.1	-	-
RA9P8	-	27S29	-	7649	6349	74S472	82S147	18S42	-
RA10P4	-	27S33	-	7643	6353	74S573	82S137	24S41	-
RA10P8	-	27S181	-	7681	6381	87S81	82S181	28S86	-
RA11P4	-	27S185	-	7685	6389	87S185	82S185	24S81	-
RA11P8	-	27S191	-	76161	S1681	87S191	82S191	28S166	-
RA12P4	-	27S41	-	76165	63S1641	-	-	-	-
RA12P8	-	-	-	76321	63S3281	87S321	82S321	-	-

## Сокращения:

- ALT - Altera
- CYP - Cypress
- HAR - Harris
- VTI - VLSI Technology, Inc.
- AMD - Advanced Micro Devices
- MHI - Monolithic Memories, Inc.
- NAT - National Semiconductor
- SIG - Signetics
- TI - Texas Instrument



## ПРИЛОЖЕНИЕ С

### СООБЩЕНИЯ ОБ ОШИБКАХ

Это приложение содержит списки сообщений об ошибках, с которыми вы можете встретиться, когда используете языковой процессор АБЕЛЬ или какие-либо утилиты из программного пакета АБЕЛЬ. Сообщения об ошибках подразделены в четыре субсекции:

- C.1 Сообщения об обычных ошибках
- C.2 Сообщения о нефатальных ошибках симуляции
- C.3 Сообщения об ошибках программы TOABEL
- C.4 Сообщения об ошибках программы IFLDOC

Внутри этих субсекций, сообщения об ошибках расположены в алфавитном порядке, и где это необходимо приводятся дополнительные пояснения. Сочетание "xxx" означает что в сообщении об ошибке в этом месте появляется дополнительный текст или числовое значение. Текстовая информация может значительно зависеть от условий, вызвавших ошибку, поэтому вместо этого текста используется сочетание "xxx".

#### C.1. СООБЩЕНИЯ ОБ ОБЫЧНЫХ ОШИБКАХ

Следующие сообщения об ошибках могут быть отображены на экране в течение исполнения программ АБЕЛЯ. Они перечислены здесь в алфавитном порядке для вашего удобства. Там, где это необходимо, приводятся дополнительные пояснения.

##### C.1.1. ОШИБКИ КОМАНДНОЙ СТРОКИ

Ошибки в командной строке означают что ошибка была сделана когда запускалась одна из программ языкового процессора. Эти ошибки могли иметь место когда программа запускалась непосредственно или из "batch"-файла.

- Illegal argument -x
- Illegal breakpoint
- Illegal option 'xxx', option must start with a '-'
- Illegal type of output file 'xxx', valid = 0, 82 or 83
- Illegal value for checksum 'xxx', valid = 0, 1 or 2
- Illegal value for flag -Xn
- Illegal value for unused terms 'xxx', valid = y,n
- Illegal value for checksum 'xxx', valid = 0, 1 or 2
- Illegal type of output file 'xxx', valid = 0, 82 or 83
- Input and output file names may not be the same
- Input & output file names are the same 'xxx'
- List and input file names may not be the same
- List and output file names may not be the same
- No device specified
- Reduction choices are 0, 1, 2, not n
- Same file used for input and output, 'xxx'

(Должны быть использованы различные файлы для ввода и вывода, для того чтобы избежать перезаписи какой-либо информации).

- Unable to open input file 'xxx'
- Unable to open output file 'xxx'
- 'xxx' default X values are 0 or 1
- 'xxx' default Z values are 0 or 1
- 'xxx' preload options are 0, 1 or 2
- 'xxx' reduction choices are 0, 1 or 2
- 'xxx' trace level are 0, 1, 2, 3
- 'xxx' trace level for breakpoint are 0, 1, 2, 3
- 'xxx' unrecognized parameter

#### C.1.2. ОШИБКИ ФАЙЛА УСТРОЙСТВА

Ошибки файла устройства отображаются в следующем формате:

Device file error 'xxx'

где 'xxx' является номером обозначающим тип ошибки файла устройства. Любая из этих ошибок указывает на ошибку в файле спецификации устройства, аппаратную ошибку или "блоху" в программе.

#### C.1.3. ФАТАЛЬНЫЕ ОШИБКИ

Фатальными ошибками являются ошибки, которые вынуждают программу завершить свою работу. Эти ошибки должны быть исправлены, прежде чем снова запустить программу.

Can't open device file 'xxx'

(Файл указанной микросхемы не может быть найден на дисковом устройстве, или в директории, определенной флагом -J).

Equation for 'xxx' needs reduction

(Используйте следующий, более высокий, уровень редукции чтобы устранить эту ошибку).

Equation for 'xxx' is not sums-of-products

(Существует две причины для этих ошибок:

1. Входной файл программы FUSEMAP не является выходным файлом программы REDUCE.

2. Вы использовали операторы другие чем &, # и ! (и \$ для ППЗУ) и использовали только "-R0"-редукцию в программе REDUCE. Запустите REDUCE с более высоким уровнем редукции).

Error writing JEDEC file

(Это общая ошибка, которая включает ошибки записи во все выходные файлы загрузки программатора).

Error writing list file

Error writing output file

(Выходной файл не может быть дописан далее. Это может произойти когда на диске нет свободного места или данные на диске повреждены.)

Expected a '\*' in JEDEC file

Fuse out of range in JEDEC file

Illegal fuse state in JEDEC file 'xxx'

Illegal operator in equation for 'xxx'



(Используйте следующий, более высокий, уровень редукции чтобы устранить эту ошибку.)

Illegal term element at 'xxx'

(Используйте следующий, более высокий, уровень редукции чтобы устранить эту ошибку.)

Intermediate file not produced by PARSE

Memory overflow

(Переполнение памяти может иметь место по одной из следующих причин:

1. Слишком много уравнений или тестовых векторов.
2. Уравнения слишком сложные.
3. Уравнения имеют слишком много операторов \$.
4. Недостаточно доступной свободной памяти.
5. Слишком много символов.

Если память переполнилась при исполнении программы PARSE, попытайтесь переписать ваши уравнения так, чтобы они имели меньше произведений термов и/или меньше операторов \$ (оператор \$ расширяется в несколько термов каждый.)

Number too large in JEDEC file

(Эта ошибка порождается при чтении JEDEC файла с параметром -N.)

PALs and GPLAs can only be output to a JEDEC file

(Для PAL и GPL тип выходного файла (формат JEDEC) должен быть определен параметром \*-D0\* или оставлен не определенным.)

Premature end of source file

PROMs can not be output to a JEDEC file

(Для ПЗУ тип выходного файла должен быть определен параметром \*-Dxx\*, где \*xx\* не равно 0.)

Too many errors

(PARSE прекращает работу когда в исходном файле обнаружено слишком много ошибок.)

Unable to open input file 'xxx'

Unable to open output file 'xxx'

(FUSEMAP не способна открыть определенный для вывода (записи).)

Vector in JEDEC file wrong size

(Генерируется когда читается файл типа JEDEC используя флаг -N.)

Wrong version of intermediate file.

### C.1.4. ОШИБКИ ПРОМЕЖУТОЧНОГО ФАЙЛА

Промежуточными файлами являются файлы создаваемые одной из программ языкового процессора АБЕЛЬ и используемые одной или несколькими другими программами. Если в промежуточном файле существуют ошибки, то по всей вероятности последующие программы, которым этот файл нужен, не будут правильно работать.

File not produced by FUSEMAP

File not produced by ABEL 1.1

(Этот файл не был создан одной из абелевских программ пакета АБЕЛЬ версии 1.1.)

File not produced by REDUCE

(Промежуточный файл, который определен как входной файл для FUSEMAP не был создан программой REDUCE).

Intermediate file error, SGxxx

(Эти ошибки происходят по следующим причинам:

1. Что-то исказило промежуточный файл.
2. Ошибка в предыдущем проходе.
3. Флаговая ошибка (например -B1)
4. Аппаратная ошибка).

Intermediate file error, FGxxx

(Ошибка обнаружена на входе FUSEMAP. Это может быть вызвано следующими причинами:

1. Использование входного файла созданного программой REDUCE, который содержит ошибки.
2. Аппаратная ошибка).

Intermediate file error, TGxxx

(Ошибка обнаружена на входе TRANSFOR. Это может быть вызвано следующими причинами:

1. Использование входного файла созданного программой PARSE, который содержит ошибки.
2. Входной файл создан не программой PARSE.
3. Аппаратная ошибка).

Intermediate file error, DGxxx

(Эти ошибки означают одно из следующего:

1. Назначенный входной файл не был создан одной из программ языкового процессора.
2. Что-то исказило промежуточный файл.
3. Была ошибка в более ранних проходах.
4. Аппаратная ошибка).

Intermediate file error, RGxxx

(Эти ошибки означают одно из следующего:

1. Пользователь имеет входной файл, который был создан не программой TRANSFOR.
2. Что-то исказило промежуточный файл.
3. Была ошибка в более ранних проходах.
4. Аппаратная ошибка.
5. Промежуточный файл был создан не АБЕЛем версии 1.1).

### C.1.5. ВНУТРЕННИЕ ОШИБКИ

Внутренние ошибки вызываются "блохами" в программах и неисправностями аппаратуры. Вы не должны сталкиваться с внутренними ошибками, но если это произошло, проверьте вашу аппаратуру. Внутренние ошибки отображаются в следующем формате:

Internal error: YYxxx

где YY является двухсимвольным кодом индицирующим программу, которая ошиблась, а \*xxx\* является номером внутренней ошибки.



## C.1.6. ЛОГИЧЕСКИЕ ОШИБКИ

Логические ошибки являются ошибками, связанными с программируемым устройством, и указывают на элементы в вашей разработке, которые не могут быть реализованы на выбранной микросхеме, или которые выполнены некорректно. Логические ошибки могут быть созданы ошибками в исходном файле или ошибками в вашем проекте.

Attribute for 'xxx' is 'latch' but pin is register

(Этот вход имеет неправильный атрибут, присвоенный ему. Перепишите ваше определение атрибутов или используйте другую микросхему.)

Attribute for 'xxx' is 'register' but pin is latch

(Этот вход имеет неправильный атрибут, присвоенный ему. Перепишите ваше определение атрибутов или используйте другую микросхему.)

Cannot use an XOR for 'xxx'

(Оператор ИСКЛЮЧАЮЩЕЕ ИЛИ не поддерживается для вывода 'xxx'. Используйте большее редуцирование чтобы устранить операторы ИСКЛЮЧАЮЩЕЕ ИЛИ.)

Enable for 'xxx' is a pin, not equation

('xxx' является идентификатором вывода или узла.)

Enable of 'xxx' can only be a constant or a pin

(Разрешение этого вывода может быть запрограммировано несколькими способами. Изучите микросхему и измените enable (разрешение) так чтобы оно соответствовало возможностям микросхемы.)

Enable of 'xxx' has already been programmed

(Enable (разрешение) этого вывода является также разрешением другого вывода и уже запрограммировано. Перепишите ваши уравнения чтобы устранить повторное программирование строки.)

Enable of 'xxx' has already been set for this group

(Enable (разрешение) этого вывода может быть сделано только в группе, а разрешение группы, включающей этот вывод, уже установлено. Перепишите ваши уравнения так чтобы разрешение этого вывода не репрограммировалось.)

Error writing hex output file

Illegal constant for output 'xxx'

(Единственные допустимые константы это 0 и 1: 1 для соединенных перемычек, а 0- для расплавленных.)

Illegal pin for enable of 'xxx'

(Разрешение этого вывода может быть запрограммировано несколькими способами. Изучите микросхему и измените enable (разрешение) так чтобы оно соответствовало возможностям микросхемы.)

Inverted input not allowed from 'xxx'

(Вывод (или узел) 'xxx' не может быть входом данного типа. Перепишите ваши уравнения или используйте другой тип.)

Node number 'xxx' is not defined for this device

Number must be 1 or 0 for enable of 'xxx'

(Разрешение этого вывода может быть запрограммировано несколькими способами. Изучите микросхему и измените enable (разрешение) так чтобы оно соответствовало возможностям микросхемы.)

Output 'xxx' used more than once

(Некоторые выводы могут быть выбраны больше чем одним способом. Вы имеете уравнения, которые ссылаются к данному выводу более одного способа. Перепишите ваши уравнения, чтобы использовать только один метод.)

Pin can only be negative logic for enable of 'xxx'

(Разрешение этого вывода может быть запрограммировано несколькими способами. Изучите микросхему и измените enable (разрешение) так чтобы оно соответствовало возможностям микросхемы.)

Pin number 'xxx' not defined for this device

Pin or node 'xxx' is already defined as 'xxx'

Positive input not allowed from 'xxx'

(Этот вход из вывода/узла может быть только одного типа. Перепишите ваши уравнения, чтобы использовать другой тип.)

Signal 'xxx' is not in device 'xxx'

Signal 'xxx' used or defined as combinational

Signal 'xxx' used or defined as registered

Single fuse equation for 'xxx' must be a constant

(Единственные допустимые константы это 0 и 1: 1 для соединенных перемычек, а 0- для расплавленных.)

State diagrams not supported for device 'xxx'

Status of 'xxx' register has already been set for this group

(Выбор типа регистровый/нерегистровый для вывода в выбранной микросхеме может быть сделан только в группе, и это сделано для группы, включающей этот вывод. Перепишите ваши уравнения так чтобы все выходы в этой группе были одного типа ('=' или ':=').)

Too many terms for FPLA array at output 'xxx'

(Уравнение для 'xxx' требует программирования слишком многих термов для ПМЛ. Используйте редукцию более высокого уровня.)

Too many terms for output 'xxx'

(В этой микросхеме недостаточно термов, чтобы запрограммировать ваши уравнения. Используйте редукцию более высокого уровня (например -r2) или перепишите ваши уравнения чтобы использовать меньше термов.)

'xxx' attribute not allowed on this device

'xxx' cannot be enabled

(Вы написали уравнение для разрешения вывода, который не имеет такой функции.)

'xxx' cannot be programmed inverted

(Уравнение для 'xxx' имеет неправильную полярность для указанного вывода и вывод не может быть запрограммирован чтобы изменить полярность. Либо перепишите ваши уравнения, либо используйте редукцию более высокого уровня, которая автоматически преобразует ваши уравнения в правильную полярность.)

'xxx' has no enable

('xxx' является идентификатором вывода или узла.)

'xxx' has no registered feedback

(Вы используете 'xxx' как входной или вывод обратной связи, но он не обладает такими свойствами. Перепишите ваши уравнения так, чтобы они не использовали этот вывод как вход.)

'xxx' is a registered pin

'xxx' is an invalid output

('xxx' является идентификатором вывода или узла.)

'xxx' is not an input pin



(Ваши уравнения пытаются использовать вывод микросхемы как входной или обратную связь, который не является входным выводом и/или не имеет обратной связи.)

'xxx' is not an input/programmable pin

'xxx' is not an output pin

(Вы определили в качестве выхода вывод, который не является выходом.)

'xxx' is not a registered pin

(Вы использовали неправильный тип присвоения для этого вывода. Используйте "\*" для регистровых выводов и "=" для нерегистровых выводов.)

'xxx' is not in sums-of-product form

### C.1.7. ОШИБКИ ПРЕПРОЦЕССОРА

Ошибки препроцессора являются синтаксическими ошибками, обнаруженными программой препроцессора PARSE. Препроцессор обрабатывает абелевский исходный файл, прежде чем он подвергается дальнейшему разбору; препроцессор расширяет макросы и исполняет директивы. Чтобы исправить ошибки препроцессора, необходимо устранить ошибки в исходном файле.

(' expected

;) expected

'=' expected

'xxx' actual arguments expected

Block expected

Constant label expected

Dummy argument expected

Identifier expected

Label expected

Number expected

Number is too large

Radix 'xxx' is not one of 2, 8, 10 or 16

String expected

### C.1.8. СИНТАКСИЧЕСКИЕ ОШИБКИ

Синтаксические ошибки обнаруживаются программой PARSE и отражают отсутствие, некорректность или неполноту элементов и структур в абелевском исходном файле. Исправление синтаксических ошибок производится устранением соответствующих ошибок в исходном файле. Файл-листинг программы PARSE может помочь вам в этом. Файл-листинг содержит указатели, которые отмечают где, приблизительно, находится ошибка, следом за которыми находится сообщение, описывающее ошибку.

)' expected

'->' expected

'xxx' feature not implement

;) expected

'+' or '\*' expected

'>' expected

'>' or '->' expected

;) expected

'=' or ':=' expected

] expected

THEN' expected

'xxx' actual arguments specified on command line

(Число аргументов найденных после ключевого слова MODULE меньше чем число аргументов в командной строке.)

Actual argument length exceeds 'xxx' chars

lock expected

Can't have letters imbedded in a number

Can't map set onto a different sized set

Can't map set onto a non-set element

(Группы не могут присваиваться числам, сигналам или специальным константам; они могут быть присвоены только другим группам.)

Cannot operate on signal 'xxx'

(Невозможно воздействовать на константу в этом контексте.)

Closing '"' of string not found

Closing ')' of block not found

Declaration keyword expected

(Значимые декларационные ключевые слова следующие:

PIN

NOD

MACRO

DEVICE)

Device label expected

Digit not in radix 'xxx'

Don't know what device to use

(Необходимо "IN device id" либо никакая микросхема не была определена.)

Dummy argument 'xxx' not recognized

(Фиктивный аргумент обнаруженный после "?" не был ранее определен.)

Element expected

(Здесь ожидается группа, идентификатор, число, специальная константа или скобочное выражение.)

ELSE' expected

(Конструкция IF-THEN-ELSE требует чтобы был представлен оператор

ELSE.)

Enables are not registered

'GOTO', 'II' OR 'CASE' expected

Identifier expected

Identifier length exceeds 'xxx' chars

Illegal character 'xxx' in source file

('xxx' является десятичным значением забракованного символа.)

Illegal operation on special constant

Inconsistency in number of parameters

(Число меток с левой стороны от оператора декларации не соответствует числу деклараций с правой стороны от ключевого слова.)

Industry part number string expected

Invalid flag string 'xxx'

Keyword 'xxx' is out of context



Label 'xxx' is already defined  
 Label expected  
 Max of 'xxx' elements in set was exceeded  
 Mismatch in number of set elements  
 (Группы с различным числом элементов в них не могут быть обработаны двоичным оператором.)  
 'MODULE' expected  
 Module label doesn't match 'xxx'  
 (Если метка использована после оператора END конца модуля, эта метка должна совпадать с меткой, использованной с оператором MODULE, который начинает модуль.)  
 Multiple mapping to signal 'xxx'  
 (Была сделана попытка присвоить более одного тестового условия одному сигналу.)  
 No more than 'xxx' args are declared  
 (Использовано больше фактических аргументов, чем было определено аргументов фиктивных.)  
 Number expected  
 Numeric overflow  
 (Определено число, которое требует более 32 бит для своего представления.)  
 Only ^b, ^a, ^d or ^h radix allowed  
 Only one label allowed  
 Section keyword or 'END' expected  
 (Секция должна начинаться с одного из следующих ключевых слов:  
 EQUATIONS  
 TRUTH TABLE  
 STATE DIAGRAM  
 TEST VECTORS)  
 Signal not allowed  
 Signal number for 'xxx' is way too large  
 Signal or .X. expected  
 Source line length exceeds 'xxx' chars  
 Special constant must end with a '.'  
 Special constant not allowed  
 String expected  
 String length exceeds 'xxx' chars  
 Undefined compiler directive '@xxx'  
 Undefined label 'xxx'  
 Undefined label 'xxx'; may be 'xxx' was meant  
 Undefined operation on set  
 Undefined operation on signal  
 Undefined special constant 'xxx'  
 Undefined token  
 (В исходном файле обнаружен символ вне правильного контекста.)  
 Unrecognized attribute 'xxx'  
 Unrecognized industry part number 'xxx'  
 Use '\$' for OR instead of '∨'

## C.2. НЕФАТАЛЬНЫЕ ОШИБКИ СИМУЛЯЦИИ

Нефатальные ошибки симуляции это предупреждения вам, что что-то не вполне правильно в вашем проекте, но эта ошибка недостаточно серьезна, чтобы завершить симуляцию. Уделите серьезное внимание этим ошибкам; они могут указывать на ошибки в разработке, которые могут вызвать проблемы в запрограммированной микросхеме.

Device unstable  
 (Микросхема не стабилизировалась за 20 итераций.)  
 Illegal inputs on RS flip-flop, 'xxx'  
 (Изучите справочную информацию на микросхему, чтобы правильно с ней работать. Обратите внимание на узлы, управляющие триггерами.)  
 Illegal JEDEC default value, 'xxx'  
 (Ошибки при чтении JEDEC файла когда используется флаг -N.)  
 Illegal J-K inputs during load, 'xxx'  
 (Изучите справочную информацию на микросхему, чтобы правильно с ней работать. Обратите внимание на узлы, управляющие триггерами.)  
 Illegal K input on D flip-flop, 'xxx'  
 (Изучите справочную информацию на микросхему, чтобы правильно с ней работать. Обратите внимание на узлы, управляющие триггерами.)  
 Illegal P-R inputs during load, 'xxx'  
 (Изучите справочную информацию на микросхему, чтобы правильно с ней работать. Обратите внимание на узлы, управляющие триггерами.)  
 Output not High-Z during load, 'xxx'  
 (Изучите справочную информацию на микросхему, чтобы правильно с ней работать. Обратите внимание на узлы, управляющие триггерами.)  
 Unknown JEDEC identifier, 'xxx'  
 (Ошибки при чтении JEDEC файла когда используется флаг -N.)

## C.3. СООБЩЕНИЯ ОБ ОШИБКАХ ПРОГРАММЫ TOABEL

Ошибки программы TOABEL даны не в алфавитном, а в цифровом порядке.

Fatal error 701: Invalid command line argument  
 Fatal error 702: Duplicate input and output file name  
 Fatal error 703: Error opening input file  
 Fatal error 704: Error opening output file  
 Fatal error 705: Device not supported  
 Fatal error 706: Unexpected end of file encountered  
 Fatal error 707: Error writing output file  
 Syntax error 710: Invalid symbol '  
 Syntax error 711: Not a valid pin identifier  
 Syntax error 712: Duplicate pin identifier  
 Syntax error 713: No device found in input file  
 Syntax error 714: Missing '('  
 Syntax error 715: Too many elements in enable expression  
 Syntax error 716: Expected pin identifier  
 Syntax error 717: ':' expected  
 Syntax error 718: Undefined pin identifier  
 Syntax error 719: Invalid symbol in test vector



## С.4. ОШИБКИ ПРОГРАММЫ IFLDOC

Следующие ошибки могут иметь место при выполнении программы IFLDOC.

### С.4.1. ОШИБКИ КОМАНДНОЙ СТРОКИ

Too many input options of 'xxx'

(Эта ошибка вызывается определением более трех входных параметров. Это может быть вызвано также при наличии пробела внутри параметра.)

Missing -in'xxx'

(Эта ошибка имеет место если "-" не является первым символом параметра, введенного в командной строке.)

Unrecognized parameter, 'xxx'

(Эта ошибка имеет место если в качестве параметра введен символ не "0, o, I, i, N, n".)

Multiple input files

(Во входной строке определено более одного входного файла. Имеет смысл только один.)

Multiple output files

(Во входной строке определено более одного выходного файла. Имеет смысл только один.)

Multiple devices

(Во входной строке определено более одной микросхемы. Имеет смысл только одна.)

No device type specified

(Должен быть определен тип микросхемы.)

Input and output files have the same name

(Входной и выходной файлы должны иметь различные имена.)

Can't open the file named 'xxx'

(Не удалось открыть входной или выходной файл с именем 'xxx'.)

Unknown device type 'xxx'

(Микросхема 'xxx' не поддерживается или тип микросхемы введен неправильно.)

### С.4.2. ДИАГНОСТИЧЕСКИЕ ОШИБКИ

Диагностические ошибки означают что либо произошла аппаратная неисправность, либо в программе есть "блоха". Диагностические ошибки представлены в следующей форме:

Diagnostic Error 'xxx'

### С.4.3. ОШИБКИ ВХОДНОГО ФАЙЛА

Unrecognized field identifier, 'xxx'

(Эта ошибка случается когда во входном JEDEC файле обнаружен нелегальный идентификатор поля.)

Missing '\*' or End of File

(IFLDOC не смог обнаружить "\*" или маркер конца файла в конце поля JEDEC.)

The JEDEC file contains references to fuses beyond the maximum for this device

(Эта ошибка происходит когда JEDEC файл ссылается на большее число перемычек, чем это доступно в данной микросхеме. Например, если JEDEC файл ссылается на перемычку с номером 5000 в микросхеме с 200 перемычками, произойдет эта ошибка. Проверьте и убедитесь, что тип микросхемы, определенный в командной строке, соответствует JEDEC файлу.)

Unrecognized character in JEDEC field, 'x'

(В поле перемычек обнаружен символ отличный от "1" и "0".)

Default fuse value specified after fuse field reached

(Чтобы предотвратить потерю информации о перемычках, перед любым полем перемычек (поле "L") в JEDEC файле должно быть определено значение количества перемычек по умолчанию. Эта ошибка происходит когда значение умолчания обнаружено после поля перемычек.)

### С.4.4. ПРЕДУПРЕЖДАЮЩИЕ СООБЩЕНИЯ

Следующие сообщения являются предупреждениями, что во входном JEDEC файле могут быть ошибки, но эти ошибки не достаточно серьезны, чтобы завершить выполнение программы.

The number of fuses in this device is not equal to the number given in the JEDEC field

(Возможно вы имеете неправильный JEDEC файл для микросхемы, определенной в командной строке. Проверьте соответствие между JEDEC файлом и типом микросхемы.)

The JEDEC file may contain less information than is necessary

(Возможно вы имеете неправильный JEDEC файл для микросхемы, определенной в командной строке. Проверьте соответствие между JEDEC файлом и типом микросхемы.)



## ПРИЛОЖЕНИЕ D УЗЛЫ МИКРОСХЕМ

Таблица D-1 перечисляет узлы, поддерживаемые АБЕЛем вместе с именами, данными этим узлам производителем. Обратитесь также к логическим схемам производителей. Использование узлов обсуждается в СПРАВОЧНИКЕ ПО ЯЗЫКУ АБЕЛЬ и в РУКОВОДСТВЕ ПО ПРИМЕНЕНИЮ.

Таблица D-1. Узлы, используемые АБЕЛем

Номер узла	Имя	Функции
F82S105 (28 выводов) 556PT21		
29	RF0	R-вход триггера F0 (вывод 18)
30	RF1	R-вход триггера F1 (вывод 17)
31	RF2	R-вход триггера F2 (вывод 16)
32	RF3	R-вход триггера F3 (вывод 15)
33	RF4	R-вход триггера F4 (вывод 14)
34	RF5	R-вход триггера F5 (вывод 13)
35	RF6	R-вход триггера F6 (вывод 12)
36	RF7	R-вход триггера F7 (вывод 11)
37	P0	S-вход триггера (и выход) P0
38	P1	S-вход триггера (и выход) P1
39	P2	S-вход триггера (и выход) P2
40	P3	S-вход триггера (и выход) P3
41	P4	S-вход триггера (и выход) P4
42	P5	S-вход триггера (и выход) P5
43	RP0	R-вход триггера P0
44	RP1	R-вход триггера P1
45	RP2	R-вход триггера P2
46	RP3	R-вход триггера P3
47	RP4	R-вход триггера P4
48	RP5	R-вход триггера P5
49	COMP	Дополнительный терм на смещение 44

Номер узла	Имя	Функции
F82S155 (20 выводов)		
21	FC	Управление режимом триггера
22	PB	Установка триггеров группы B
23	RB	Сброс триггеров группы B
24	PA	Установка триггеров группы A
25	RA	Сброс триггеров группы A
26	LB	Загрузка триггеров группы B
27	LA	Загрузка триггеров группы A

28	COMP	Дополнительный узел смещения 32
29	M0	Управление режимом триггера F0 (вывод 14)
30	M1	Управление режимом триггера F1 (вывод 15)
31	M2	Управление режимом триггера F2 (вывод 16)
32	M3	Управление режимом триггера F3 (вывод 17)
33	K0	K-вход триггера F0 (вывод 14)
34	K1	K-вход триггера F1 (вывод 15)
35	K2	K-вход триггера F2 (вывод 16)
36	K3	K-вход триггера F3 (вывод 17)
37	J0	J-вход триггера F0 (вывод 14)
38	J1	J-вход триггера F1 (вывод 15)
39	J2	J-вход триггера F2 (вывод 16)
40	J3	J-вход триггера F3 (вывод 17)

Номер узла	Имя	Функции
F82S157 (20 выводов)		
21	FC	Управление режимом триггера
22	PB	Установка триггеров группы B
23	RB	Сброс триггеров группы B
24	PA	Установка триггеров группы A
25	RA	Сброс триггеров группы A
26	LB	Загрузка триггеров группы B
27	LA	Загрузка триггеров группы A
28	COMP	Дополнительный узел смещения 32
29	M0	Управление режимом триггера F0 (вывод 13)
30	M1	Управление режимом триггера F1 (вывод 14)
31	M2	Управление режимом триггера F2 (вывод 15)
32	M3	Управление режимом триггера F3 (вывод 16)
33	M4	Управление режимом триггера F4 (вывод 17)
34	M5	Управление режимом триггера F5 (вывод 18)
35	K0	K-вход триггера F0 (вывод 13)
36	K1	K-вход триггера F1 (вывод 14)
37	K2	K-вход триггера F2 (вывод 15)
38	K3	K-вход триггера F3 (вывод 16)
39	K4	K-вход триггера F4 (вывод 17)
40	K5	K-вход триггера F5 (вывод 18)
41	J0	J-вход триггера F0 (вывод 13)
42	J1	J-вход триггера F1 (вывод 14)
43	J2	J-вход триггера F2 (вывод 15)
44	J3	J-вход триггера F3 (вывод 16)
45	J4	J-вход триггера F4 (вывод 17)
46	J5	J-вход триггера F5 (вывод 18)



Номер узла	Имя	Функции
F82S159 (20 выводов)		
21	FC	Управление режимом триггера
22	PB	Установка триггеров группы В
23	RB	Сброс триггеров группы В
24	PA	Установка триггеров группы А
25	RA	Сброс триггеров группы А
26	LB	Загрузка триггеров группы В
27	LA	Загрузка триггеров группы А
28	COMP	Дополнительный узел смещения 32
29	M0	Управление режимом триггера F0 (вывод 12)
30	M1	Управление режимом триггера F1 (вывод 13)
31	M2	Управление режимом триггера F2 (вывод 14)
32	M3	Управление режимом триггера F3 (вывод 15)
33	M4	Управление режимом триггера F4 (вывод 16)
34	M5	Управление режимом триггера F5 (вывод 17)
35	M6	Управление режимом триггера F6 (вывод 18)
36	M7	Управление режимом триггера F7 (вывод 19)
37	K0	К-вход триггера F0 (вывод 12)
38	K1	К-вход триггера F1 (вывод 13)
39	K2	К-вход триггера F2 (вывод 14)
40	K3	К-вход триггера F3 (вывод 15)
41	K4	К-вход триггера F4 (вывод 16)
42	K5	К-вход триггера F5 (вывод 17)
43	K6	К-вход триггера F6 (вывод 18)
44	K7	К-вход триггера F7 (вывод 19)
45	J0	Ж-вход триггера F0 (вывод 12)
46	J1	Ж-вход триггера F1 (вывод 13)
47	J2	Ж-вход триггера F2 (вывод 14)
48	J3	Ж-вход триггера F3 (вывод 15)
49	J4	Ж-вход триггера F4 (вывод 16)
50	J5	Ж-вход триггера F5 (вывод 17)
51	J6	Ж-вход триггера F6 (вывод 18)
52	J7	Ж-вход триггера F7 (вывод 19)

Номер узла	Имя	Функции
F82S167 (24 вывода) 556PT22		
25	RF0	R-вход триггера F0 (вывод 9)
26	RF1	R-вход триггера F1 (вывод 10)
27	RF2	R-вход триггера F2 (вывод 11)
28	RF3	R-вход триггера F3 (вывод 13)
29	RF0	R-вход триггера P0 (вывод 14)
30	RP0	R-вход триггера P1 (вывод 15)
31	P7	S-вход триггера (и выход) P7
32	P6	S-вход триггера (и выход) P6
33	P5	S-вход триггера (и выход) P5
34	P4	S-вход триггера (и выход) P4
35	P3	S-вход триггера (и выход) P3
36	P2	S-вход триггера (и выход) P2
37	RP7	R-вход триггера P7
38	RP6	R-вход триггера P6
39	RP5	R-вход триггера P5
40	RP4	R-вход триггера P4
41	RP3	R-вход триггера P3
42	RP2	R-вход триггера P2
43	COMP	Дополнительный терм на смещение 45
EP300		
21	R	терм сброса переключки 2628
22	P	терм сброса переключки 2592
P22V10 (24 вывода)		
25	R	терм сброса переключки 0000
26	P	терм сброса переключки 5764