

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。



お客様各位

資料中の「三菱電機」、「三菱XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って株式会社日立製作所及び三菱電機株式会社のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。

従いまして、本資料中には「三菱電機」、「三菱電機株式会社」、「三菱半導体」、「三菱XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

注:「高周波・光素子事業、パワーデバイス事業については三菱電機にて引き続き事業運営を行います。」

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

M3XXXXT-PAC

《リロケータブルアセンブラSRA74M編》

ユーザーズマニュアル

Simple Emulator System Package for 740 Family

《安全設計に関するお願い》

三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障または誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

《本資料ご利用に際しての留意事項》

本資料は、お客様が用途に応じた適切な三菱半導体製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社が所有する知的財産権その他の権利の実施、使用を許諾するものではありません。

本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は責任を負いません。

本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は、予告なしに、本資料に記載した製品または仕様を変更することがあります。三菱半導体製品のご購入に当たりましては、事前に三菱電機株式会社、三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社または特約店へ最新の情報をご確認頂きますとともに、三菱電機半導体情報ホームページ (<http://www.semicon.melco.co.jp/>) および三菱開発ツールホームページ (<http://www.tool-spt.maec.co.jp/>) などを通じて公開される情報に常にご注意ください。

本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社はその責任を負いません。

本資料に記載の製品データ、図、表に示す技術的な内容、プログラムおよびアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は、適用可否に対する責任を負いません。

本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、三菱電機株式会社、三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社または特約店へご照会ください。

本資料の転載、複製については、文書による三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社の事前の承諾が必要です。

本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたら三菱電機株式会社、三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社または特約店までご照会ください。

《製品の内容および本書についてのお問い合わせ先》

三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社 マイコンツール部

[技術サポート窓口] support@tool.maec.co.jp

[ユーザ登録窓口] regist@tool.maec.co.jp

[ホームページアドレス] <http://www.tool-spt.maec.co.jp/>

目次

1	SRA74M 操作方法	1
1.1	SRA74M 起動方法	1
1.2	SRA74M 入力パラメータ	3
1.2.1	ソースファイル名 (必須)	3
1.2.2	コマンドパラメータ	3
1.2.3	SRA74M オペレーティングシステムへの戻り値	4
1.3	SRA74M 環境変数	4
2	LINK74M 操作方法	5
2.1	LINK74M 起動方法	5
2.1.1	対話式入力	5
2.1.2	コマンド行入力	7
2.1.3	コマンドファイル入力	8
2.2	LINK74M 入力パラメータ	9
2.2.1	リロケータブルファイル名	9
2.2.2	ライブラリファイル名	9
2.2.3	コマンドパラメータ	10
2.2.4	LINK74M セクション制御	10
2.3	LINK74M オペレーティングシステムへの戻り値	11
2.4	LINK74M 環境変数	11
3	SRA74M 機能	12
3.1	生成ファイル	12
3.1.1	PRN ファイルの構成	14
4	LINK74M 機能	16
4.1	機能	16
4.2	生成ファイル	16
4.3	MAP ファイルの構成	17
5	セクションの機能	19
5.1	セクションの属性	19
5.1.1	アドレス属性	19

5.1.2	物理属性	20
5.1.3	予約セクション	20
5.2	セクションの配置規則	22
5.2.1	セクションの配置例	22
6	ソースプログラムの記述方法	24
6.1	ソースプログラムの構成	24
6.2	行の構成	25
6.2.1	アセンブリ言語命令行	25
6.2.2	疑似命令行	25
6.2.3	コメント行	26
6.3	シンボル/ビットシンボル/ラベル欄の記述方法	26
6.4	コメント欄の記述方法	27
7	アセンブリ言語	28
7.1	アドレッシングモード	28
7.2	オペランドデータ形式	31
7.3	演算子	32
8	疑似命令	34
8.1	疑似命令の機能	34
8.2	アセンブル制御	35
8.3	アドレス制御	36
8.4	リンク制御	37
8.5	リスト制御	38
8.6	疑似命令一覧	38
9	SRA74M エラー	55
9.1	SRA74M エラーの種類	55
9.2	SRA74M システムエラー一覧表	57
9.3	SRA74M アセンブルエラー一覧表	58
9.4	SRA74M ワーニング一覧表	63
10	LINK74M エラー	64
10.1	LINK74M エラーの種類	64
10.2	LINK74M エラーメッセージ一覧表	66

目 次

1.1	SRA74M 起動コマンド入力例	1
1.2	SRA74M 正常終了時の画面表示	2
2.1	LINK74M 起動画面	6
2.2	LINK74M 対話式入力時の画面	6
2.3	LINK74M コマンド行入力例 1	7
2.4	LINK74M コマンド行入力例 2(ライブラリ名の省略)	7
2.5	LINK74M コマンド行入力例 3(コマンドパラメータの省略)	7
2.6	LINK74M コマンドファイルの指定	8
2.7	LINK74M コマンドファイル記述例	8
3.1	プリントファイル出力例	15
4.1	MAP ファイル出力例	18
5.1	リロケータブルファイル構成図	22
5.2	システムメモリマップ	23
9.1	SRA74M エラー表示例	56
10.1	LINK74M エラー表示例	65

表 目 次

1.1	SRA74M コマンドパラメーター一覧表	3
1.2	SRA74M エラーレベル一覧表	4
2.1	LINK74M コマンドパラメーター一覧表	10
2.2	LINK74M エラーレベル一覧表	11
7.1	演算子一覧表	32
9.1	SRA74M システムエラー一覧表	57
9.2	SRA74M アセンブルエラー一覧表	58
9.3	SRA74M ワーニング一覧表	63
10.1	LINK74M エラーメッセージ一覧表	66

はじめに

マニュアルで使用している記号について

\

本マニュアルに掲載されているプログラム例など、一部においてスペシャルページアドレッシングモードを表す‘¥’記号を‘\’で表していることがあります。これは、オペレーティングシステムによって表記が異なるためで、コードは同一ですのでいずれも使用できます。

スペース又はタブコードを示しています。ただし、 は必須であることを示します。

スペース又はタブコードを示しています。 は省略できます。

ライブラリファイルについて

本マニュアルに記述されている「ライブラリファイル」は、740 ファミリ用アセンブラ SRA74 で生成されたファイルを示します。したがって、ライブラリファイルを利用する場合は、必ず SRA74 で生成されたファイルを指定してください。

第 1 章

SRA74M 操作方法

1.1 SRA74M 起動方法

SRA74M は、オペレーティングシステムのプロンプトからコマンド行を入力することにより起動します。図 1.1に、起動コマンドの入力例を示します。

```
A>SRA74M FILENAME -L -S <RET>
```

アセンブル対象の コマンドパラメータ
ソースファイル名

図 1.1: SRA74M 起動コマンド入力例

図 1.2に、SRA74M の正常終了画面を示します。

```
A>SRA74M TEST<RET>
740 Family SRA74M V.1.00.00
Copyright 1996, MITSUBISHI ELECTRIC CORPORATION
AND MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION
All Rights Reserved.

now processing pass 1 ( TEST.A74 )
----*----
now processing pass 2 ( TEST.A74 )
----*----

ERROR   COUNT           00000
WARNING COUNT          00000
TOTAL   LINE ( SOURCE ) 00994 LINES
TOTAL   LINE ( OBJECT ) 00994 LINES
COMMENT LINE ( SOURCE ) 00247 LINES
COMMENT LINE ( OBJECT ) 00147 LINES
OBJECT  SIZE ( Z       ) 00010 (000A) BYTES
OBJECT  SIZE ( R       ) 00053 (0035) BYTES
OBJECT  SIZE ( P       ) 01938 (0792) BYTES

A>
```

図 1.2: SRA74M 正常終了時の画面表示

1.2 SRA74M 入力パラメータ

1.2.1 ソースファイル名 (必須)

1. アセンブル対象のソースファイル名を指定します。指定できる個数は1個です。
2. ファイル拡張子 (.A74) を省略した場合、既定値として拡張子.A74 を選びます。
3. ファイル名をフルネームで指定することにより、.A74 以外の拡張子 (例 .ASM) のファイルもアセンブル可能です。
4. ファイル名にはディレクトリパス指定が可能です。ファイル名のみを指定した場合は、カレントドライブのカレントディレクトリ中のファイル进行处理します。
例) A>SRA74 C:¥WORK¥TEST<RET>

1.2.2 コマンドパラメータ

1. コマンドパラメータはマイナス記号 (-) 及びそれに続く文字で構成されます。
2. 文字は大文字/小文字どちらでも有効です。
3. 各パラメータは同時に複数指定することができます。この場合は、各パラメータをスペースで区切って入力してください。
4. マイナス記号を2つ続けることによりそのコマンドパラメータを無効にすることができます。例えば --L とすると PRN ファイルの生成が無効となります。

表 1.1に、コマンドパラメータの内容を示します。

表 1.1: SRA74M コマンドパラメータ一覧表

コマンドパラメータ	内容
-C	ファイル中の全ての行に対してソースラインデバッグ情報を出力します。
-L	PRN ファイルを生成します。この指定がない場合 PRN ファイルは生成しません。
-S	R74 ファイルにローカルなビットシンボル、シンボル、及びラベルを出力します。

1.2.3 SRA74M オペレーティングシステムへの戻り値

SRA74M では、実行結果を表 1.2に示す 5 つの戻り値に分けてオペレーティングシステムに返します。

表 1.2: SRA74M エラーレベル一覧表

エラーレベル	実行結果の内容
0	正常終了
1	アセンブル対象のソースファイルの内容に関するエラー
2	SRA74M のコマンド入力に関するエラー
3	オペレーティングシステムに関するエラー
4	^ C (コントロールC) 入力による強制終了

1.3 SRA74M 環境変数

SRA74M は、以下の環境変数を使用しています。

- TMP
アセンブルするときに生成するテンポラリファイルの作成ディレクトリ名を指定します。この環境変数が設定されていない場合、テンポラリファイルはソースファイルと同じディレクトリ内に生成されます。環境変数の設定例を以下に示します。

例) `set TMP=A:¥TMP`

- INC
インクルードファイルを検索するディレクトリ名を指定します。この環境変数が設定されていない場合、`.INCLUDE` 疑似命令で指定されているインクルードファイルはカレントディレクトリから検索されます。環境変数の設定例を以下に示します。

例) `set INC=A:¥INC`

第 2 章

LINK74M 操作方法

2.1 LINK74M 起動方法

LINK74M を起動する方法として、操作環境に応じた以下の 3 通りの方式を用意しています。

1. 対話式入力
2. コマンド行入力
3. コマンドファイル入力

2.1.1 対話式入力

対話式入力の特徴を以下に示します。

1. リロケートブルファイル、ライブラリファイル、セクション制御コマンド、コマンドパラメータの順に対話式に入力する方式です。
2. リロケートブルファイルやセクションの数が少ない場合や、試行錯誤的にアドレス設定を試したい場合などに便利な方式です。
3. コマンド行入力やコマンドファイル入力でコマンドの数が不足している時は、自動的にこの方式に移行します。

対話式入力は、オペレーティングシステムのプロンプト状態で LINK74M<RET>と入力することにより起動します。LINK74M は、起動後 図 2.1 のような画面を出力します。

図 2.1 の最後の行が、リロケートブルファイルの入力待ちを示しています。 >> に続いて、リンク対象のリロケートブルファイルを入力してください。対話式入力では、このように順次ライブラリファイル名、セクション制御、コマンドパラメータの入力待ちとなりますので、図 2.2 のように入力していきましょう (複数のファイルを入力する場合、各ファイル名はスペース又はタブで区切ってください。ライブラリファイル名入力、セクション情報、コマンドパラメータについても同様です)。

```
A>LINK74M <RET>
740 Family LINKER V.1.00.00
Copyright 1996, MITSUBISHI ELECTRIC CORPORATION
AND MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION
All Rights Reserved.

Relocatable files (.R74) >>
```

図 2.1: LINK74M 起動画面

```
A>LINK74M <RET>
740 Family LINKER V.1.00.00
Copyright 1996, MITSUBISHI ELECTRIC CORPORATION
AND MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION
All Rights Reserved.

Relocatable files (.R74) >> MAIN SUB<RET>
Libraries          (.LIB) >> UTIL1 UTIL2<RET>
Section information >> WORKRAM=100 PROM=C000 DROM<RET>
Command parameter  >> -O\WORK -M -S<RET>
```

図 2.2: LINK74M 対話式入力時の画面

2.1.2 コマンド行入力

コマンド行入力の特徴を以下に示します。

1. オペレーティングシステムのコマンド入力状態ですべてのコマンド入力を行う方式です。
2. MS-DOS では、コマンドの長さが 127 文字以内に制限されているためリロケータブルファイルやセクションの数が少ない場合に使用できます。
3. バッチファイルや、make ファイル中で実行コマンドを記述する場合に使用できます。
4. 入力パラメータの 4 種類の情報は、';'(カンマ) で区切って入力してください。図 2.2 と同じコマンドをコマンド行で入力した例を図 2.3 に示します。
5. ライブラリファイル名以降のパラメータが不要な場合も、カンマは必ず入力してください。図 2.3 の例で、ライブラリファイルが必要ない場合は図 2.4 のようになります。
6. 特別な場合として、コマンドパラメータを省略する場合のみコマンドパラメータがないことを明確に示すために 2 つのカンマが必要になります。図 2.4 でコマンドパラメータを省略した例を図 2.5 に示します。
7. 入力パラメータの数が不足している時は、自動的に対話式入力状態に移行します。

```
A>LINK74M MAIN SUB, UTIL1 UTIL2, WORKRAM=100 PROM=C000 DROM, -O\WORK -M -S<RET>
```

図 2.3: LINK74M コマンド行入力例 1

```
A>LINK74M MAIN SUB,, WORKRAM=100 PROM=C000 DROM, -O\WORK -M -S<RET>
```

図 2.4: LINK74M コマンド行入力例 2(ライブラリ名の省略)

```
A>LINK74M MAIN SUB,, WORKRAM=100 PROM=C000 DROM,,<RET>
```

図 2.5: LINK74M コマンド行入力例 3(コマンドパラメータの省略)

2.1.3 コマンドファイル入力

コマンドファイル入力の特徴を以下に示します。

1. リンクコマンドを予めエディタによりコマンドファイルとして作成し、起動時にこのファイル名を指定する方式です。
2. コマンドファイル名の指定は、オペレーティングシステムから LINK74M を起動する時に図 2.6のように、ファイル名の前に '@' をつけて指定します。図 2.6の場合、ファイル名 CMD.DAT の内容がコマンドとして実行されます。
3. コマンドファイルに記述する内容は、コマンド行入力と同じです (ただし、LINK74M の起動を指示する LINK74M の部分は不用)。改行コードは無視しますので長いコマンドは複数行に分けて記述できます。図 2.5をコマンドファイルで記述すると図 2.7のようになります。
4. コマンドパラメータが不足している場合、LINK74M は対話式入力状態に移ります。例えば、図 2.7の最後の行の 2 つ目のカンマがない場合 LINK74M は、コマンドパラメータの対話式入力画面になります。

```
A>LINK74M @CMD.DAT<RET>
```

図 2.6: LINK74M コマンドファイルの指定

```
MAIN SUB  
,  
,WORKRAM=100 PROM=C000 DROM  
,,
```

図 2.7: LINK74M コマンドファイル記述例

2.2 LINK74M 入力パラメータ

2.2.1 リロケータブルファイル名

1. リロケータブルファイル名は必ず入力してください。
2. リロケータブルファイルは3つまで指定できます。
3. リロケータブルファイルは、拡張子.R74 を持ったものだけを扱います。コマンド入力では、ファイル拡張子を省略できます。
4. ファイル名は、ディレクトリパスが指定可能です。ファイル名のみを記述した場合、カレントドライブのカレントディレクトリ中のファイル进行处理します。
5. 最初に指定したリロケータブルファイル名が出力ファイル名になります。コマンドパラメータ“-F”で出力ファイル名を指定した場合は、コマンドパラメータ“-F”を優先します。
6. 最初に指定したリロケータブルファイルのある拡張子ディレクトリに出力ファイルを出力します。コマンドパラメータ“-O”でディレクトリを指定した場合は、コマンドパラメータ“-O”を優先します。

2.2.2 ライブラリファイル名

1. LINK74M は、ライブラリファイルを生成しません。既にライブラリファイルが存在している場合のみライブラリファイル名を指定してください。

注意事項

本マニュアルに記述されている「ライブラリファイル」は、740 ファミリー用アセンブラ SRA74 で生成されたファイルを示します。したがって、ライブラリファイルを利用する場合は、必ず SRA74 で生成されたファイルを指定してください。

2. ライブラリファイル名は省略できます。
3. ライブラリファイルは、拡張子.LIB を持ったものだけを扱います。コマンド入力の際は、ファイル拡張子を省略できます。
4. ファイル名には、パス指定ができます。ファイル名のみを指定した場合、カレントディレクトリのファイル进行处理します。
5. ライブラリファイルは、リロケータブルファイル中に解決されないグローバルラベル又はグローバルシンボルがあった場合のみ参照します。

2.2.3 コマンドパラメータ

コマンドパラメータは、リンクの出力ファイル、バージョン確認の有無等の制御を行います。表 2.1に、コマンドパラメータの内容を示します。

表 2.1: LINK74M コマンドパラメータ一覧表

コマンドパラメータ	内容
-A	同名の絶対属性セクションのオーバーラップを許可します。グローバルなメモリ領域を共有結合する場合に利用できます。
-F	出力ファイル名を指定します。指定の書式は次のようになります。 -Fsample
-M	MAP ファイルの出力を行います。(セクション情報のみ)
-MS	MAP ファイルを、グローバルラベル、グローバルビットシンボル、及びグローバルシンボルリスト付きで出力します。
-N	ソースファイル中の疑似命令.OBJ、.LIB により指定された.R74、.LIB ファイルの参照指定情報を無視します。
-O	出力ファイルのディレクトリを指定します。指定の書式は、次のようになります。 -OC:¥USR¥WORK C ドライブの ¥USR¥WORK を出力ディレクトリに指定します。
-S	SYM ファイルの出力を行います。
-V	リロケータブルファイル間のバージョン整合性の確認を行います。ただし、確認を行うためには、アセンブリ言語ソース中で疑似命令.VER によりバージョンの指定を行っておく必要があります。

2.2.4 LINK74M セクション制御

1. セクション情報は省略できます。この場合、読み込んだリロケータブルファイル中で出会ったセクション順に配置されます。
2. セクション情報の指定は、相対属性のセクションに対して行って下さい。絶対属性のセクションは、セクション指定の有無に関わらず疑似命令.ORG で指定した固定のアドレスから配置されます。
3. セクション配置の指定は、アドレスの下位に配置するセクションから順に、セクション名を空白で区切って記述してください。
4. 各セクションの開始アドレスは、該当セクション名の後に '=' (イコール) に続けてアドレスを記述してください。アドレスは、16 進数で指定してください。この時、先頭の '0' と最後の 'H' は不用です。

5. セクション名は、大文字/小文字を区別します。
6. 相対属性のセクションの開始アドレスを省略した場合、0000₁₆ 番地から配置します。ただし、予約セクションである S セクションは、FF00₁₆ から配置します。

注意事項

スペシャルページが FF00₁₆ から始まらないマイクロコンピュータを使用する場合は、必ず S セクションの開始アドレスを指定してください。

7. セクションのアドレスオーバーラップはエラーになります。
ただし、コマンドパラメータ“-A”を指定すると、絶対アドレスをオーバーラップさせることができます。これにより、SFR 領域などの絶対アドレスラベルを外部参照指定せず、疑似命令“.INCLUDE”により、複数のプログラムファイルに読み込むことが可能になります。

2.3 LINK74M オペレーティングシステムへの戻り値

LINK74M では、実行結果を表 2.2 の 5 つの戻り値をオペレーティングシステムに返します。

表 2.2: LINK74M エラーレベル一覧表

エラーレベル	実行結果の内容
0	正常終了
1	リンク対象のリロケータブルファイルの内容に関するエラー
2	LINK74M のコマンド入力に関するエラー
3	オペレーティングシステムに関するエラー
4	LINK74M の機能に関するエラー

2.4 LINK74M 環境変数

- LIB74

LINK74M は、ライブラリファイルを読み込む際に環境変数 LIB74 を参照します。ライブラリファイルが存在するディレクトリパス名を環境変数 LIB74 に設定することにより、コマンド入力においてパス名の指定を省略することができます。以下に環境変数の設定例を示します。

例) `set LIB74=A:¥LIB`

第 3 章

SRA74M 機能

SRA74M は、740 ファミリ用リロケータブルアセンブラです。アセンブリ言語で記述されたソースファイルを LINK74M で処理可能な形式のリロケータブルファイルに変換します。LINK74M は、リロケータブルファイルを機械語ファイルに変換します。

1. 疑似命令 SECTION により、分割された領域に任意の名前（セクション名）を指定できます。リンク時には、ここで指定したセクション名を使用してアドレスを指定できます。
2. 疑似命令 LIB、及び OBJ により、リンク対象ファイル名をソースプログラム中で指定できます（LINK74M 起動時にリンク対象ファイル名の指定が不要になります）。
3. 疑似命令 VER によってバージョンを宣言することで、リンクガリロケータブルファイル間のバージョンを確認します（コマンドオプション“-V”指定時のみ）。

3.1 生成ファイル

SRA74M では、以下の 3 種類のファイルを生成します。なお、コマンドパラメータ等の指定がない場合、ソースファイルと同じディレクトリに生成します。

1. リロケータブルファイル（以下 R74 ファイルと呼びます）
 - 機械語データと、その再配置情報を格納したファイルです。
 - コマンドパラメータの内容に関わらず生成します。但し、アセンブルエラーの発生などによって、SRA74M が異常終了した場合には生成しません。
 - シンボリックデバッグのためのシンボル情報を含んでいます。但し、規定値ではローカルシンボル情報は出力されません。
 - ローカルシンボル情報は、コマンドパラメータ“-S”を指定したときに R74 ファイル中に出力されます。
 - ソースラインデバッグ情報は、コマンドパラメータ“-C”を指定したときに R74 ファイル中に出力されます。
 - ファイル拡張子は、.R74 です。

注意事項

R74 ファイルはバイナリ形式ですので、プリンタ及び画面への出力は、行わないでください。

2. プリントファイル (以下 PRN ファイルと呼びます)

- 処理対象のソースファイルと、その配置アドレス及び生成データを示したファイルです。
- PRN ファイルは、コマンドパラメータ “-L” を指定したときに生成します。
- PRN ファイルは、プリント出力してデバッグなどにお使いください。
- ファイル拡張子は、.PRN です。

3. 一時ファイル (以下 TMP ファイルと呼びます)

- アセンブル作業用一時ファイルです。
- このファイルは、環境変数 TMP が設定されている場合 TMP により指定しているディレクトリ内に生成されます。
- このファイルは、アセンブル終了時に自動的に消去されます。
- ファイル拡張子は、.\$n(n は 1 ~ 5) です。

3.1.1 PRN ファイルの構成

図 3.1に、PRN ファイルの出力例を示します。PRN ファイルは、以下の情報を示しています。

1. プリントファイルの各ページの先頭にはヘッダ情報を出力します。疑似命令".COL"によるカラム数の指定が 132 文字の場合は、ヘッダ部分にアセンブルを実行した時刻を次の形式で表示します。
Mon Dec 16 15:06:42 1996
2. インクルードファイルを読み込んでいる部分には 1 から 4 までの数字を出力します。数字は、インクルードファイルのネスティングレベルを示します。
3. シンボルを参照している行には、"S"を出力します。シンボルは、疑似命令".EQU"で定義された名前を示します。
4. パブリックラベルを参照している行には、"P"を出力します。パブリックラベルは、疑似命令".PUB"で宣言された名前を示します。
5. ローカルラベルを参照している行には、"L"を出力します。ローカルラベルは、ラベルが定義されたファイル内でのみ参照されるラベルを示します。
6. ビットシンボルを参照している行には、"B"を出力します。ビットシンボルは、疑似命令".EQU"で定義されたビット毎につけられた名前を示します。
7. エクスターナルラベルを参照している行には、"E"を出力します。エクスターナルラベルは、疑似命令".EXT"で宣言された名前を示します。
8. プリントファイルの最後には、アセンブル結果情報を出力します。エラー数、ワーニング数、全行数、コメント行数及びセクション毎のメモリ容量を示します。

* 740 Family SRA74M V.1.00.00 *

SEQ.. LOC. OBJ... S NM3....*....4....*....5....SOURCE STATEMENT....*....8..

```

1          0          .INCLUDE zero.h
2          1          .PUB    ram,work1,work2,flags
3          1          .SECTION z
4          1          .ORG    0
5 0000 (0001) 1 ram:    .BLKB  1
6 0001 (0001) 1 work1: .BLKB  1
7 0002 (0001) 1 work2: .BLKB  1
8 0003 (0001) 1 flags:  .BLKB  1
9 0,0003      P 1 bit0  .EQU   0,flags
10 0000       1 zero   .EQU   0
11          0          .SECTION p
12          0          .ZEXT   stack
13          0          .SEXT   sub1
14          0          .PUB    start
15 0000       0 start:
16 0000 D8    0          CLD
17 0001 12    0          CLT
18 0002       0 ramclr:
19 0002 A6FF  0          LDX    $FF
20 0004 A900  S 0          LDA    #zero
21 0006 9500  P 0          STA    ram,X
22 0008 CA    0          DEX
23 0009 10F7  L 0          BPL    ramclr
24 000B       0 main:
25 000B 0F03  B 0          SEB    bit0
26 000D 3C0001 P 0          LDM    #0,work1
27 0010 3C0002 P 0          LDM    #0,work2
28 0013 2200  E 0          JSR    \sub1
29 0015 170302 L 0          BBC    bit0,??loop1
30 0018 80F1  L 0          BRA    main
31 001A       0 ??loop1:
32 001A 80E6  L 0          BRA    ramclr
33          0          .END

```

```

ERROR  COUNT          00000
WARNING COUNT          00000
TOTAL  LINE ( SOURCE ) 00033 LINES
TOTAL  LINE ( OBJECT ) 00033 LINES
COMMENT LINE ( SOURCE ) 00000 LINES
COMMENT LINE ( OBJECT ) 00000 LINES
OBJECT  SIZE ( z      ) 00004 (0004) BYTES
OBJECT  SIZE ( p      ) 00028 (001C) BYTES

```

図 3.1: プリントファイル出力例

第 4 章

LINK74M 機能

LINK74M は、SRA74M で生成したリロケータブルファイルをリンクし、740 ファミリ用機械語データファイルを生成します。

4.1 機能

1. 複数のリロケータブルファイル中の同一セクション名を持つ領域を連結して配置します。セクションとは、プログラムを構成する ROM 領域や RAM 領域のような物理的に異なる性質をもった要素単位を指します。
2. セクションの配置順序やセクション単位ごとの開始アドレスを指定できます。
3. セクション情報及びラベル情報を一覧表示したマップファイルを生成します。

4.2 生成ファイル

LINK74M では、以下の 3 種類のファイルを生成します。

1. 機械語データファイル (以下 HEX ファイルと呼びます)
 - インテル 16 進形式で出力します。
 - ファイル拡張子は、.HEX です。
2. マップファイル (以下 MAP ファイルと呼びます)
 - 各セクションが最終的に配置されたアドレス情報を示します。
 - このファイルはプリンタへ出力して、デバッグ時や各セクションのメモリ容量の把握のためにお使いください。
 - MAP ファイルは、コマンドパラメータ “-M” を指定した時に出力します。
 - ファイル拡張子は、.MAP です。

3. シンボルファイル (以下 SYM ファイルと呼びます)

- デバッグを行う時に必要な各種情報を含んだファイルです。
- SYM ファイルは、コマンドパラメータ “-S” を指定した時に出力します。
- ファイル拡張子は、.SYM です。

4.3 MAP ファイルの構成

図 4.1に、MAP ファイルの出力例を示します。MAP ファイルは、以下の情報を示しています。

1. リロケータブルファイル情報 (セクション単位)

ここには、次の情報を出力しています。

- ATR 部：相対属性か絶対属性かを示します。REL は相対、ABS は絶対をそれぞれ示しています。

アセンブリ疑似命令.ORG(または *=) により開始アドレス指定を行ったものが絶対属性になります。

- TYPE 部：RAM 領域か ROM 領域かを示します。
- START 部：開始アドレスを示します。
- LENGTH 部：領域の大きさをバイト数で示します。
- ライブラリファイルをリンクした場合、ライブラリファイル名とリロケータブルファイル名の両方が示されます。リロケータブルファイル名は、() 内に表示します。

2. グローバルラベルリスト

プログラム中のグローバルラベルとその絶対番地を示します。この部分は、コマンドパラメータ “-MS” を指定した時のみ出力します。

グローバルラベルとは、疑似命令.PUB で宣言したラベルを指します。

3. グローバルシンボルリスト

プログラム中のグローバルシンボルとその絶対番地を示します。この部分は、コマンドパラメータ “-MS” を指定した時のみ出力します。

グローバルシンボルとは、疑似命令.PUB で宣言したシンボルを指します。

4. グローバルビットシンボルリスト

プログラム中のグローバルビットシンボル、ビット値、及び絶対番地を示します。この部分は、コマンドパラメータ “-MS” を指定した時のみ出力します。

グローバルビットシンボルとは、疑似命令.PUB で宣言したビットシンボルを指します。

740 Family LINKER V.1.00.00 MAP FILE Mon Dec 16 18:30:58 1996

SECTION	FILENAME	ATR.	TYPE	START	LENGTH
WORKRAM	MAIN.R74	ABS	RAM	0000	0080
	SUB.R74	REL	RAM	0080	0100
	UTIL.LIB (CALC.R74)	REL	RAM	0180	0008
PROM	MAIN.R74	REL	ROM	C000	1800
	SUB.R74	REL	ROM	D800	1500
	UTIL.LIB (CALC.R74)	REL	ROM	ED00	0820
DROM	MAIN.R74	REL	ROM	F520	0023
	SUB.R74	REL	ROM	F544	0030

GLOBAL LABEL INFORMATION

ADCNT	0030	COUNT	009C	DATA0	00A4
DATA1	00A6	MAIN	C000	TIME	00C6

GLOBAL SYMBOL INFORMATION

GLOBAL BIT SYMBOL INFORMATION

図 4.1: MAP ファイル出力例

第 5 章

セクションの機能

セクションとは、各セクション毎に独立したロケーションアドレスを持つプログラム単位です。リンク時に、セクションを単位としてプログラムアドレスを再配置します。

5.1 セクションの属性

各セクションは、アドレスの配置に関する情報 (再配置可能か固定か) と、物理的な配置に関する情報 (ROM か RAM か) の 2 つの情報を持っています。前者をアドレス属性、後者を物理属性と呼びます。以下では、各属性の内容について説明します。

5.1.1 アドレス属性

アドレス属性には、相対属性と絶対属性の 2 つの属性があります。これらの属性は、ソースプログラムの該当セクション中の疑似命令.Org (または * =) の有無によって決定します。各属性の特徴について以下に示します。

1. 相対属性

- セクション内に疑似命令.Org がない場合、そのセクションは相対属性になります。相対属性のセクションは再配置可能なセクションで、リンク時に開始アドレスを指定することができます。

2. 絶対属性

- セクション内に疑似命令.Org が書かれている場合、そのセクションは絶対属性になります。絶対属性のセクションは、疑似命令.Org で指定した固定のアドレスに配置されます。
- 絶対属性のセクションは、リンク時に開始アドレスの指定を行うことはできません。

複数のリロケータブルファイルに存在する同一名セクションは、異なるアドレス属性を持つことができます。

5.1.2 物理属性

物理属性には、ROM 属性と RAM 属性の 2 つの属性があります。これらの属性は、セクションが配置される領域の物理的性質を示しています。物理属性の特徴を以下に示します。

1. ROM 属性

- アセンブリ言語ソース中にコードを発生するステートメント (LDA 等のアセンブリ言語命令や疑似命令.BYTE 等) が記述されているセクションが ROM 属性になります。
- ROM 属性のセクションはリンクの結果、機械語コードを生成します。
- アドレス属性との混同を避けるためこの属性を ROM タイプと呼びます。

2. RAM 属性

- RAM 属性のセクションはリンクの結果、機械語コードを生成しません。
- アセンブリ言語ソース中に、領域確保疑似命令.BLKB を記述しているセクションが RAM 属性になります。
- アドレス属性との混同を避けるためこの属性を RAM タイプと呼びます。

同一名のセクションは、すべて同一の物理属性でなければいけません。

5.1.3 予約セクション

LINK74M は、セクション Z、S、P、R を予約セクションとして扱います。以下で各々の予約セクションについて説明します。

1. Z セクション

Z セクションは、740 ファミリのゼロページ (00₁₆番地から FF₁₆番地) に配置される RAM 属性のセクションです。SRA74M の疑似命令.SECTION でセクション名 'Z'(小文字でも可) を宣言した場合、そのセクションは Z セクションになります。

Z セクションが相対属性の場合、LINK74M のコマンド入力で先頭アドレスを指定することができます。配置アドレスコマンドの指定を省略した場合は、00₁₆番地から配置します。

2. S セクション

S セクションは、740 ファミリのスペシャルページ (FF00₁₆番地から FFFF₁₆番地) に配置される ROM 属性のセクションです。SRA74M の疑似命令.SECTION でセクション名 'S'(小文字でも可) を宣言した場合、そのセクションは S セクションになります。

S セクションが相対属性の場合、LINK74M のコマンド入力で先頭アドレスを指定することができます。配置アドレスの指定を省略した場合は、FF00₁₆番地から配置します。
注意事項

スペシャルページが FF00₁₆ から始まらないマイクロコンピュータを使用する場合は、必ず開始アドレスを指定してください。

3. P セクション

P セクションは、プログラムまたは固定データを持つ ROM 属性のセクションです。プログラム領域内 (00_{16} 番地から $FFFF_{16}$ 番地) のどの領域にも配置可能です。SRA74M の疑似命令 .SECTION でセクション名 'P' (小文字でも可) を宣言した場合、そのセクションは P セクションになります。

P セクションが相対属性の場合、LINK74M のコマンド入力で配置アドレスの指定を省略すると 00_{16} 番地から配置します。

4. R セクション

R セクションは、RAM 属性のセクションです。プログラム領域内 (00_{16} 番地から $FFFF_{16}$ 番地) のどの領域にも配置可能です。SRA74M の疑似命令 .SECTION でセクション名 'R' (小文字でも可) を宣言した場合、そのセクションは R セクションになります。

R セクションが相対属性の場合、LINK74M のコマンド入力で配置アドレスの指定を省略すると 00_{16} 番地から配置します。

5.2 セクションの配置規則

1. 同一名のセクションはリンク時に、連続して配置されます。同一名セクション間に他の名前のセクションが配置されることはありません。
2. 同一名セクション内での配置順序は、リンクコマンド中のリロケータブルファイル名の指定順序に従います。

5.2.1 セクションの配置例

セクションの配置例を次に示します。各リロケータブルファイルは図 5.1のように、それぞれ RAM 領域、プログラム領域、固定データ領域を持っています。

これらのリロケータブルファイルを図 5.2のようなメモリ空間に配置したい場合、予めアセンブラ疑似命令 SECTION により結合したいセクションに同一の名前を指定します。LINK74M では各セクションの開始アドレスをリンク時に指定できます。リンク時のコマンドによってシステムの最終的なアドレスに対応した機械語データを作成できます。

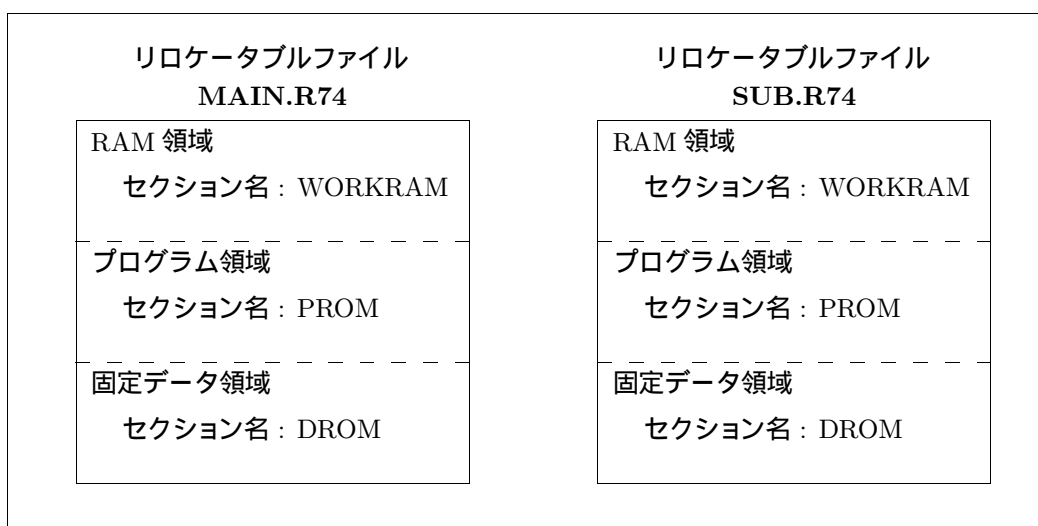


図 5.1: リロケータブルファイル構成図



図 5.2: システムメモリマップ

第 6 章

ソースプログラムの記述方法

6.1 ソースプログラムの構成

ソースプログラムは、行を単位として構成しています。以下に、行記述の規則を示します。

1. 各行は 1 行ごとに完結しています。従って 1 命令を、2 行以上にわたって記述することはできません。
2. 1 行の文字数は、改行コードを含めて最大 256 文字です。SRA74M は、256 文字を越えた部分を無視します。
3. 行は内容別に以下の 3 種類の行に分類できます。
 - アセンブリ言語命令行
740 ファミリのアセンブリ言語命令を記述した行です。
この行は対応する機械語データを生成します。
 - 疑似命令行
740 ファミリの疑似命令を記述した行です。
 - コメント行
この行は SRA74M で処理しませんのでユーザーが自由に使用できます。

6.2 行の構成

本節では、それぞれの行の構成について説明します。

6.2.1 アセンブリ言語命令行

アセンブリ言語命令行の構成を以下に示します。

```
ラベル : オペコード オペランド ; コメント <RET>
```

1. ラベル欄

行を他の場所から参照するためのラベルを記述します。

2. オペコード欄

740 ファミリのアセンブリ言語ニーモニック (以下オペコードと呼びます) を記述します。オペコードは、英大文字/小文字を区別しません。従って NOP、nop、Nop いずれも有効です。

SRA74M はオペコードを予約語として識別しますので、ラベルを記述しない場合は行の先頭から記述できます。

3. オペランド欄

オペコードの処理対象を記述します。

- オペランドにデータが 2 つ以上ある場合、データ間を ‘,’(カンマ) で区切ってください。
- カンマの両側には、スペース又はタブコードが記述できます。

4. コメント欄

この欄は、SRA74M で処理しませんのでユーザーが自由に使用できます。

6.2.2 疑似命令行

疑似命令行の構成を以下に示します。この行の詳細については、第 8 章を参照してください。

```
シンボル/ビットシンボル .EQU オペランド ; コメント <RET>
```

```
ラベル : 疑似命令 オペランド ; コメント <RET>
```

1. シンボル/ビットシンボル欄

疑似命令.EQU により値を割り当てるシンボル、ビットシンボルを記述します。

2. ラベル欄

行を他の場所から参照するためのラベルを記述します。

3. 疑似命令欄

740 ファミリの疑似命令を記述します。疑似命令は、英大文字/小文字を区別しません。従って.END、.end、.End いずれも有効です。

SRA74M は疑似命令を予約語として識別しますので、ラベルを記述しない場合は行の先頭から記述できます。

4. オペランド欄

疑似命令の処理対象を記述します。

- オペランドにデータが 2 つ以上ある場合、データ間を ‘;’ (カンマ) で区切ってください。
- カンマの両側には、スペース又はタブコードが記述できます。

5. コメント欄

この欄は、SRA74M で処理しませんのでユーザーが自由に使用できます。

6.2.3 コメント行

コメント行は、行の最初の文字 (を除く) を ‘;’(セミコロン) で始めてください。コメント行の構成を以下に示します。

```
; コメント <RET>
```

注意事項

コメント行の最初の文字が ‘;’ 以外である場合、SRA74M はその行をコメント行とは認識せずアセンブル処理を行います。その結果、アセンブルエラーが発生したり予期しないコードを生成する可能性がありますので注意してください。

6.3 シンボル/ビットシンボル/ラベル欄の記述方法

この欄は、各命令行とも記述形式は同じです。但し SRA74M は、シンボル、ビットシンボル、ラベルを区別して管理します。以下に記述上の規則を示します。

1. 名前には、英数字、‘_’(アンダーライン)、‘.’(ピリオド)、及び‘?’(クエッションマーク) が使用できます。但し、1 文字目に使用できる文字は英字と ‘_’(アンダーライン) だけです。また、文字数は、‘?’ を含めて最大 255 文字まで使用できます。
2. 大文字/小文字を区別します。従って BIG と Big は異なった名前として判断します。
3. 名前に、予約語は使用できません。SRA74M では、レジスタ名、フラグ名、オペコード、及び疑似命令を予約語として処理します。
4. ‘.’(ピリオド) で始まるラベルは今後の SRA74M 機能拡張のため予約されていますので使用しないでください。

5. “..”(ピリオド 2 個) で始まるラベルの内、以下のラベルは SRA74M で予約されていますので使用しないでください。また、以下のもの以外についても、‘..’ で始まるラベルは今後の SRA74M 機能拡張のため予約されていますので使用しないでください。

- ..0 ~ ..65535

6. “??”(クエッションマーク 2 個) で始まるラベルは、そのセクション内だけで有効なセクション内ローカルラベルとして扱われます。従って、異なるセクションで同一名のラベルが使用できます。参照した場合は、常にそのセクション内のラベルのみが参照されます。

例)

```

        .SECTION PROG1
        ??MAIN:  NOP
                :
                BRA    ??MAIN    セクション名‘PROG1’の‘??MAIN’
                                   ヘジャンプする。

        .SECTION PROG2
        ??MAIN:  NOP
                :
                BRA    ??MAIN    セクション名‘PROG2’の‘??MAIN’
                                   ヘジャンプする。

```

6.4 コメント欄の記述方法

ユーザーの任意の情報を記述することができます。以下に記述形式を示します。

1. コメントの先頭には必ず ‘;’(セミコロン) を付けて記述してください。

第 7 章

アセンブリ言語

7.1 アドレッシングモード

アドレッシングモードは、命令が処理対象データを指定する基本的な方式 (モード) のことです。740 ファミリでは 19 通りのアドレッシングモードがあり、アドレッシングモードごとにオペランド形式が決まっています。以下に、オペランド記述に関するアドレッシングモードについて示します。

1. インプライド

オペコードのみの命令です。オペランドには何も指定しません。

例) BRK

2. アキュムレータ

処理対象データをアキュムレータの内容とする方法です。

例) ASL A

3. イミディエイト

処理対象データをオペランドで直接指定する方法です。オペランドに記述する値は、'#' で始めてください。

例) ADC #IMMDATA

4. ゼロページ

処理対象データとして、ゼロページ領域 (00₁₆ ~ FF₁₆) を指定する方法です。

例) ADC ZWORK

5. ゼロページ X

処理対象データとして、ゼロページアドレスをレジスタ X で修飾する方法です。',' (カンマ) の後にレジスタ名 'X' を記述してください。

例) ADC ZWORK,X

6. ゼロページ Y

処理対象データとして、ゼロページアドレスをレジスタ Y で修飾する方法です。',' (カンマ) の後にレジスタ名 'Y' を記述してください。

例) LDX ZWORK,Y

7. ゼロページインダイレクト

処理対象データをメモリ間接で示す方法です。オペランドには、対象アドレスを格納したゼロページアドレスを記述し、メモリに2バイトの対象アドレスを格納します。オペランドに記述する値は、()で示してください。

例) JMP (ZWORK)

8. ゼロページインダイレクト X

処理対象データをメモリ間接とレジスタ X で修飾して示す方法です。オペランドには、対象アドレスを格納したゼロページアドレスを記述し、メモリに2バイトの対象アドレスを格納します。','(カンマ)の後にレジスタ名 'X' を記述し、オペランドに記述する値は、()で示してください。

例) ADC (ZWORK,X)

9. ゼロページインダイレクト Y

処理対象データをメモリ間接とレジスタ Y で修飾して示す方法です。オペランドには、対象アドレスを格納したゼロページアドレスを記述し、メモリに2バイトの対象アドレスを格納します。オペランドに記述する値は、()で示し、','(カンマ)の後にレジスタ名 'Y' を記述してください。

例) ADC (ZWORK),Y

10. アブソリュート

処理対象データとして、一般ページ領域(0100₁₆ ~ FFFF₁₆)を指定する方法です。

例) ADC WORK

11. アブソリュート X

処理対象データとして、一般ページアドレスをレジスタ X で修飾する方法です。','(カンマ)の後にレジスタ名 'X' を記述してください。

例) ADC WORK,X

12. アブソリュート Y

処理対象データとして、一般ページアドレスをレジスタ Y で修飾する方法です。','(カンマ)の後にレジスタ名 'Y' を記述してください。

例) ADC WORK,Y

13. アブソリュートインダイレクト

処理対象データをメモリ間接で示す方法です。オペランドには、対象アドレスを格納した一般ページアドレスを記述し、メモリに2バイトの対象アドレスを格納します。オペランドに記述する値は、()で示してください。

例) JMP (WORK)

14. スペシャルページ

処理対象データとして、スペシャルページ領域を(FF00₁₆ ~ FFFF₁₆)を指定する方法です。オペランドに記述する値は、'¥' 又は '\ ' で示してください。

例) JSR ¥WORK

15. ゼロページビット

処理対象データとして、ゼロページ領域 ($00_{16} \sim FF_{16}$) の特定ビットを指定する方法です。また、オペランドにビットシンボルを記述すると、そのビットシンボルのビット値とアドレスが参照されます。

例) CLB 0,ZWORK ⇒ ‘ZWORK’ のビット 0 を指定します。

例) CLB BITSYMBOL

16. アキュムレータビット

処理対象データとして、アキュムレータの特定ビットを指定する方法です。また、第 1 オペランドにビットシンボルを記述すると、そのビットシンボルのビット値だけが参照されます。

例) CLB 1,A ⇒ アキュムレータのビット 1 を指定します。

例) CLB BITSYMBOL,A

17. レラティブ

本命令の先頭アドレスに、オペランドの内容を加えた番地にジャンプします。但し、オペランドには相対値そのものを記述することはできません。オペランドに、ラベル又は対象アドレスを記述すると、SRA74M が相対値を計算します。

例) BRA *-12

例) BRA NEXT

18. ゼロページビットレラティブ

処理対象データとして、ゼロページ領域 ($00_{16} \sim FF_{16}$) の特定ビットを指定します。また第 1 オペランドにビットシンボルを記述すると、そのビットシンボルのビット値とアドレスが参照され、そのビットの状態により本命令の先頭アドレスに、最終オペランドの内容を加えた番地にジャンプする方法です。但し、最終オペランドには相対値そのものを記述することはできません。ラベル、又は対象アドレスを記述すると、SRA74M が相対値を計算します。

例) BBC 2,ZWORK,NEXT ⇒ ‘ZWORK’ のビット 2 を指定します。

例) BBC BITSYMBOL,NEXT

19. アキュムレータビットレラティブ

処理対象データとして、アキュムレータの特定ビットを指定します。また第 1 オペランドにビットシンボルを記述すると、そのビットシンボルのビット値だけが参照され、そのビットの状態により本命令の先頭アドレスに、最終オペランドの内容を加えた番地にジャンプする方法です。但し、最終オペランドには相対値そのものを記述することはできません。ラベル、又は対象アドレスを記述すると、SRA74M が相対値を計算します。

例) BBC 3,A,NEXT ⇒ アキュムレータのビット 3 を指定します。

例) BBC BITSYMBOL,A,NEXT

7.2 オペランドデータ形式

オペランドには、以下の4種類のデータ形式が記述できます。

1. 数値定数

- 数値定数の前に、‘+’又は‘-’の演算子をつけて正、又は負の値の表現ができます。‘+’も‘-’もない場合は、正の値として処理します。

- 数値の種類を表す記号と数値との間には、スペース又はタブをいれることはできません。

例) .BYTE \$ 64 ⇒ エラーとなります。

- 数値定数には、2進、8進、10進、16進数が使用できます。

- 2進数 ⇒ 2進数で構成され、先頭に‘%’を付けるか、最後に‘B’、又は‘b’を付けて記述してください。

例) .BYTE %100110

例) .BYTE 100110B

- 8進数 ⇒ 8進数で構成され、先頭に‘@’を付けるか、最後に‘O’、‘o’、又は‘Q’、‘q’を付けて記述してください。

例) .BYTE @70

例) .BYTE 70o

例) .BYTE 70Q

- 10進数 ⇒ 10進数で構成され、特に何も指定しません。23、256のように整数のみで記述してください。

例) .BYTE 100

- 16進数 ⇒ 16進数で構成され、先頭に‘\$’を付けるか、最後に‘H’、又は‘h’を付けて記述してください。先頭が英文字(A ~ F)で始まる場合は先頭に0を付加してください。

例) .BYTE \$64

例) .BYTE 64H

例) .BYTE 0ABH

2. 文字定数

- 文字には、ASCIIコードで定義されている文字が記述できます。

- 文字定数は、’(シングルクォート)又は”(ダブルクォート)で囲んで記述してください。各文字は、7ビットASCIIコード(再上位ビットは0)に対応しています。

例) .BYTE 'A' ⇒ 41₁₆を設定します。

3. 記号定数

- 記号はシンボル、ビットシンボル、ラベル、及び現在のステートメントの先頭を表す '*' の 4 種類があります。ビットシンボルは、割り当てられたビットの値と、そのビットの属するアドレスを、シンボルは絶対値を、ラベル又は '*' は相対又は絶対値をもちます。

例) .WORD SUB ⇒ ラベル SUB のアドレスを設定します。

例) BRA **+2 ⇒ 現在のアドレスに 2 を加えたアドレスへジャンプします。

4. 式

- 式は、数値定数、文字定数、記号定数、演算子の組み合わせで構成します。演算子と各項の間には、必要に応じてスペース又はタブを入れることができます。

例) TBL + 1

- 式は左から右に計算します (演算子の優先順位はありません)。

例) 2*3 ⇒ 結果は 6 になります。

例) 2+6/2 ⇒ 結果は 4 になります。

7.3 演算子

表 7.1 に、オペランドデータ記述で使用可能な演算子の一覧表を示します。

表 7.1: 演算子一覧表

分類	演算子 ¹	内容
単 項 演 算 子	+	正の数を表す
	-	負の数を表す
	!	1 の補数をとる
	<	ラベル、又はシンボルの上位 8 ビットを切り出す
	>	ラベル、又はシンボルの下位 8 ビットを切り出す
	sizeof ²	セクションの大きさを求める
2 項 演 算 子	+	加算
	-	減算
	*	乗算
	/	除算
	&	ビットごとの AND
		ビットごとの OR

注意事項

1. SRA74M では、数値は 16 ビット符号付きの値として取り扱います。また、式の値についても同様に処理を行います。したがって、式の演算結果は次に示す例のようになります。

例)

$$40960 / 512 = \text{FFD0h}$$

演算結果をシンボル値として定義する場合は、16 ビット値の FFD0_{16} が定義され、メモリなどへ代入されるデータ値として利用する場合は、8 ビット値の D0_{16} が代入されません。

演算式をオペランドに記述する際には上記注意事項を考慮して記述願います。

2. 演算は左から右へ行います (演算子の優先順位はありません)。
例) $2+6/2 \Rightarrow$ 結果は 4 になります。
3. SIZEOF の値は、参照セクションがリロケータブルかアブソリュートかに関係なくリンク時に決まります。SIZEOF 演算子は、セクション名にしか使用できません。又、SIZEOF 演算子を用いる場合は、セクション名との間には必ずスペースを入れてください。
例) `.WORD SIZEOF DATA`
4. ビットシンボルに対する演算、“ZEXT”と“.SEXT”で参照されているラベル間の演算はできません。

第 8 章

疑似命令

8.1 疑似命令の機能

疑似命令は、命令が目的とする機械語データを生成するように SRA74M に対する指示を行います。疑似命令の名称は、SRA74M に対する指示を行うものを“宣言”、出力ファイルに影響するものを“指定”と呼んでいます。疑似命令は機能的に以下の 4 つのグループに分類できます。

1. アセンブル制御

- 疑似命令自身はデータを生成しませんが、アセンブル処理の流れを制御します。
- アドレスの更新には影響しません。
- このグループは以下の疑似命令を含みます。

<code>.ASSERT</code>	アセンブルアサーション宣言
<code>.END</code>	プログラム終了宣言
<code>.ERROR</code>	アセンブルエラー宣言
<code>.IF (.ELSE) .ENDIF</code>	条件付きアセンブル
<code>.INCLUDE</code>	ファイル読み込み

2. アドレス制御

- データ設定疑似命令は定数データの生成を行います。
- アドレスの更新を行います。
- このグループは以下の疑似命令を含みます。

<code>.EQU (=)</code>	同義定義
<code>.ORG (*=)</code>	アドレス指定
<code>.BLKB</code>	RAM 領域確保
<code>.BYTE .WORD</code>	データ設定

3. リンク制御

- リンク処理に関する制御を行います。
- このグループは以下の疑似命令を含みます。

<code>.BEXT</code>	<code>.ZBEXT</code>	外部参照指定 (ビットシンボル)	
<code>.EXT</code>	<code>.SEXT</code>	<code>.ZEXT</code>	外部参照指定 (シンボル、ラベル)
<code>.PUB</code>		パブリック指定 (ビット、シンボル、ラベル)	
<code>.SECTION</code>		ROM、及び RAM 領域指定	
<code>.SECTION P</code>		ROM 領域指定 (一般ページ)	
<code>.SECTION R</code>		RAM 領域指定 (一般ページ)	
<code>.SECTION S</code>		ROM 領域指定 (スペシャルページ)	
<code>.SECTION Z(.ZMOD)</code>		RAM 領域指定 (ゼロページ)	
<code>.OBJ</code>	<code>.LIB</code>	リンクファイル名指定	
<code>.VER</code>		バージョン指定	

4. リスト制御

- PRN ファイル出力に関する制御を行います。
- このグループは以下の疑似命令を含みます。

<code>.COL</code>	<code>.LINE</code>	リスト形式 (カラム数/行数) 指定
<code>.LIST</code>	<code>.NLIST</code>	リスト出力/抑止指定
<code>.PAGE</code>		改ページ及びタイトル指定

以下に各グループ別に疑似命令の機能を説明します。

8.2 アセンブル制御

1. アセンブルアサーション宣言

`.ASSERT`

アセンブル作業中にオペランドに記述した文字列を画面に表示します。

2. アセンブル終了宣言

`.END`

ソースプログラムの終りを宣言します。SRA74M は、この行以降の内容を処理しません。

3. アセンブルエラー宣言

`.ERROR`

この疑似命令の、オペランドに記述した文字列を画面に表示しアセンブル作業を終了します。

4. 条件付きアセンブル

.IF (.ELSE) .ENDIF

シンボル値の内容により、アセンブルを行う場所を指定します。複数仕様に対応するプログラムを 1 つのソースプログラムで管理する場合、テストルーチンのアセンブルを制御する場合などに使用できます。

5. ファイル読み込み

.INCLUDE

この命令を記述した場所に、他のファイル内容を読み込みます。大きなソースプログラムを分割して編集する場合に使用できます。

8.3 アドレス制御

1. 同義定義

.EQU、又は =

シンボルに絶対値を定義します。この疑似命令は、シンボルに 0 ~ 7 のビット値と 0000_{16} ~ $FFFF_{16}$ の値を定義します。

2. アドレス宣言

.ORG、又は *=

以降の行のアドレスを宣言します。この疑似命令を記述したセクションは絶対属性となり、リンク時にアドレス指定を行うことはできません。割り込みベクタなど、アドレスが固定している領域に使用できます。

3. 領域確保

.BLKB

オペランドで指定した容量のメモリ領域を RAM 領域に確保します。

4. データ設定

.BYTE .WORD

オペランドで指定したデータを ROM 領域に生成します。

8.4 リンク制御

1. グローバルラベル指定

.BEXT .ZBEXT

外部参照するビットシンボル名を指定します。なお、ここで指定したビットシンボル名は必ず他のファイルでパブリック指定されていなければなりません。

.EXT .SEXT .ZEXT

外部参照するラベル、及びシンボル名を指定します。なお、ここで指定したラベル名は必ず他のファイルでパブリック指定されていなければなりません。

.PUB

このファイル中で定義しているラベル、シンボル、又はビットシンボルを他のファイルから参照できるように指定します。

2. 領域指定

.SECTION

この行以降が、オペランドで指定した名前を持つ領域であることを指定します。又、領域属性はこの命令以降の命令により SRA74M が自動的に決定します。この指定は、他の領域指定命令が現れるまで有効です。

.SECTION P

この行以降が、一般ページの ROM 領域であることを指定します。この指定は、他の領域指定命令が現れるまで有効です。

.SECTION R

この行以降が、一般ページの RAM 領域であることを指定します。この指定は、他の領域指定命令が現れるまで有効です。

.SECTION S

この行以降が、スペシャルページの ROM 領域であることを指定します。この指定は、他の領域指定命令が現れるまで有効です。

.SECTION Z

この行以降が、ゼロページの RAM 領域であることを指定します。この指定は、他の領域指定命令が現れるまで有効です。

3. リンクファイル名指定

.OBJ .LIB

リンク対象の R74 ファイル名、及びライブラリファイル名を指定します。ここで宣言したファイルは、LINK74M(リンカ) が自動的に参照するため、リンク時のコマンド入力を簡略化できます。

4. バージョン指定

.VER

R74 ファイルのバージョンを指定します。LINK74M のコマンドパラメータ “-V” を指定すると R74 ファイル間のバージョンの一致を確認できます。

8.5 リスト制御

1. 改ページ及びタイトル指定

.PAGE

リストの改ページ及びタイトルを指定します。

2. リスト形式指定

.COL .LINE

リストのカラム数、行数を指定します。これらの疑似命令は、ソースファイル中に 1 回だけ記述できます。

3. リスト出力/抑止指定

.LIST .NLIST

PRN ファイルへのリスト出力制御を行います。プログラムの一部をデバッグする場合などで、リストの一部のみが必要なときに使用してください。

8.6 疑似命令一覧

各疑似命令毎にの機能と記述形式を次ページ以降に示します。

.ASSERT

アセンブルアサーション宣言

書式

```
.ASSERT '文字列'
```

内容

- アセンブル作業中に、オペランドに記述した文字列を画面に表示します。

記述例

```
.IF MODE
:
.ELSE
.ASSERT 'MODE is FALSE'
.ENDIF
```

.BEXT

外部参照指定 (一般ページビットシンボル)

書式

```
.BEXT ビットシンボル [, ビットシンボル, ..., ビットシンボル]
```

内容

- オペランドに指定したビットシンボルを外部参照することを指定します。
- 本疑似命令により指定されたビットシンボルは一般ページ内にあるものと解釈します。
- 本疑似命令は、ラベルを参照する行より前に記述してください。

記述例

```
.BEXT BIT0,BIT1,BIT2
:
```

.BLKB

RAM 領域確保 (単位: バイト)

書式`[ラベル:] .BLKB 数式`**内容**

- 数式で指定した大きさの領域を確保します (単位:1 バイト)。
- 数式中に使用するラベル又はシンボルは、この行より前に定義してください。
- オペランドに、リロケータブルな値を持つラベルは記述できません。

記述例

```
label: .BLKB 10 ; 10 バイトの領域を確保します。  
:
```

.BYTE

バイトデータ設定

書式`[ラベル:] .BYTE 数式`**内容**

- 1 バイトの定数データを設定します。
- 2 個以上のデータを設定する場合は、各データを ‘,’ で区切って指定してください。指定できる個数は 255 個までです。
- オペランドに、グローバルラベルを記述できます。

記述例

```
label1: .BYTE 10 ; 0AH を設定します。  
label2: .BYTE 'A', 'B' ; 41H、42H を設定します。  
:
```

.COL

カラム数指定 (既定値は 132)

書式

```
.COL 数式
```

内容

- リストの 1 行の文字数 (80 ~ 132) を指定します。
- 79 以下を指定した場合は 80 に、133 以上を指定した場合は 132 文字になります。
- 本疑似命令は、プログラム中に 1 回だけ記述できます。

記述例

```
.COL 80 ; カラム数を 80 に設定します。  
:
```

.END

プログラム終了宣言

書式

```
.END
```

内容

- ソースプログラムの終了を指定します。
- 本疑似命令以降の行はアセンブルしません。

記述例

```
:  
.END ; プログラムの終了を宣言します。
```

.EQU(又は‘=’)同義定義

書式 1

シンボル .EQU(又は‘=’) 数式

書式 2

ビットシンボル .EQU(又は‘=’) 数式, 数式

内容

- 左辺のシンボルに数値を割り当てます。
- 書式 1 はシンボルに符号付き 16 ビットの整数値を割り当てます。書式 2 はシンボルに 0 ~ 7 のビット値と符号付き 16 ビットの整数値 (アドレス) を割り当てます。
- 数式中使用するラベル又はシンボルは、この行より前に定義してください。
- オペランドにリロケータブルな値を持つラベルは記述できません。
- シンボルについては再設定を行うことができます。
- コマンドパラメータ “-Q” を指定することにより、シンボルの再設定時にワーニングを出力します。

記述例

```
symbol .EQU 1 ; symbol に 1 を設定
:
symbol .EQU 2 ; symbol に 2 を設定
:
symbol .EQU 3 ; symbol に 3 を設定
:
bitsym .EQU 1,23H ; bitsym に 23H のビット 1 を設定
:
```

.ERROR

アセンブルエラー宣言

書式

```
.ERROR '文字列'
```

内容

- 条件アセンブル命令 (.IF) と組み合わせて使用します。
- 条件として有り得ないものが指定された場合に、オペランドに記述した文字列を画面へ表示しアセンブル作業を終了します。

記述例

```
.IF      MODE
      :
.ELSE
.ERROR  'undefined assemble mode'
.ENDIF
```

.EXT

外部参照指定 (一般ページ)

書式

```
.EXT ラベル又はシンボル[, ラベル又はシンボル,..., ラベル又はシンボル]
```

内容

- オペランドに指定したラベルをアブソリュートアドレッシングモードで外部参照することを指定します。ゼロページアドレッシングモードでアセンブルしたい場合には.ZEXTで、スペシャルページアドレッシングモードでアセンブルしたい場合には.SEXTで指定してください。
- 本疑似命令は、ラベルを参照する行より前に記述してください。

記述例

```
.EXT  WORK1,WORK2,WORK3
      :
```

.IF (.ELSE) .ENDIF

条件付きアセンブル

書式

```
.IF 式
  <文 1>
.ELSE
  <文 2>
.ENDIF
```

内容

- オペランドの式には数式または文字列式を記述することができます。
- オペランドの数式が真 (0 でない) なら文 1 をアセンブルし、偽 (0) なら文 2 をアセンブルします。
- オペランドの文字列式が真 (文字列データがある) なら文 1 をアセンブルし、偽 (文字列データがない) なら文 2 をアセンブルします。
- ネスティングは 20 レベルまで可能です (但し、これはホストマシンのメモリ容量に依存します)。
- 文 1、文 2 には複数行が記述できます。
- オペランドに、リロケータブルな値を持つラベルは記述できません。
- オペランドに条件式判定の比較演算子が記述可能です。記述可能な条件判定式を表に示します。

<	より小さい
>	より大きい
==	等しい
!=	等しくない
<=	小さいか等しい
>=	大きいか等しい

- 未定義シンボルを数式中に使用した場合は 0 の値を持つシンボルとして処理されます。また、文字列式中に記述した場合は文字列として処理されます。
- 論理演算子 (' || ', ' && ') によりオペランドに条件式を 6 つまで組み合わせることができます。但し論理演算子の間に優先順位はなく、条件式は先頭から順番に評価されます。

記述例

(1)

```
.IF    FLAG    ; FLAG の値が真なら .ELSE までをアセンブルする
:
:
.ELSE    ; 偽なら .ENDIF までをアセンブルする
:
:
.ENDIF
```

(2)

```
.IF FLAG1 && FLAG2 || FLAG3 && FLAG4
:   この部分は次に示す表のとおりアセンブルの
:   対象となるかどうか決定されます。
.ENDIF
```

FLAG1	FLAG2	FLAG3	FLAG4	結果
真	真	-	-	真
真	偽	真	真	真
真	偽	真	偽	偽
真	偽	偽	-	偽
偽	-	-	-	偽

‘-’ は真偽どちらの場合でも同じ結果を示します。

.INCLUDE

ファイル読み込み

書式

```
.INCLUDE ファイル名
```

内容

- 本疑似命令を記述した場所に、オペランドで指定したファイルの内容を読み込みます。
- ファイル名はフルネームで指定してください。
- ネスティングは 4 レベルまで可能です。
- ネスティングレベルをプリントファイルに出力します。

記述例

```
.INCLUDE TEST.INC ; TEST.INC の内容を読み込む  
:
```

.LIB

ライブラリファイル名指定

書式

```
.LIB ファイル名 [, ファイル名, ..., ファイル名]
```

内容

- リンク対象のライブラリファイル名を指定します。
- 指定可能なファイルは、.LIB 拡張子のファイルだけです。これ以外のファイルを指定した場合は、リンク時にエラーになります。
- 本疑似命令は入れ子にできません。
- ファイル名にディレクトリパス、ファイル拡張子 (.LIB) は記述できません。

記述例

```
.LIB LIB1,LIB2,LIB3  
:
```

.LINE 1 ページ当りの行数指定 (既定値は 54)**書式**

```
.LINE 数式
```

内容

- リストの 1 ページ当りの行数 (5 ~ 255) を指定します。
- 本疑似命令は、プログラム中に 1 回だけ記述できます。

記述例

```
.LINE 60 ; 行数を 60 に設定します。  
:
```

.LIST リスト出力開始 (既定値)**書式**

```
.LIST
```

内容

- PRN ファイルへのリスト出力を行います。
- 疑似命令.NLIST により PRN ファイルへの出力を中止後、リスト出力を再開する場合に使用します。

記述例

```
.NLIST ; リスト出力の抑止を行います。  
: ; ‘.LIST’ までの間を PRN ファイルへ出力しません。  
:  
.LIST ; リスト出力の開始を行います。  
: ; この疑似命令以降を PRN ファイルへ出力します。  
:
```

.NLISTリスト出力抑止

書式

```
.NLIST
```

内容

- PRN ファイルへの出力を抑止します。
- この状態は疑似命令 `.LIST` により解除できます。

記述例

```
.NLIST          ; リスト出力の抑止を行います。
:              ; ‘.LIST’ までの間を PRN ファイルへ出力しません。
:
.LIST           ; リスト出力の開始を行います。
:              ; この疑似命令以降を PRN ファイルへ出力します。
:
```

.OBJリロケータブルファイル名指定

書式

```
.OBJ   ファイル名 [, ファイル名, ..., ファイル名]
```

内容

- リンク対象のリロケータブルファイル名を指定します。
- 指定可能なファイルは、`.R74` 拡張子のファイルだけです。これ以外のファイルを指定した場合、リンク時にエラーになります。
- 本疑似命令は入れ子にできません。
- ファイル名にディレクトリパス、ファイル拡張子 (`.R74`) は記述できません。

記述例

```
.OBJ   OBJ1,OBJ2,OBJ3
:
```

.ORG(又は“*=”)

アドレス宣言 (既定値は 0000H)

書式

`.ORG(又は ‘ ‘*=’ ’) 数式`

内容

- この行以降の開始アドレスを宣言します。
- 指定がない場合、開始アドレスは 0000_{16} として処理します。
- 本疑似命令を記述したセクションは、絶対属性になります。本疑似命令を含まないセクションは、相対属性になります。
- 数式中に使用するラベル又はシンボルは、この行より前に定義してください。
- オペランドに、リロケートブルな値を持つラベルは記述できません。

記述例

```
.ORG    0C000H ; ロケーションを C000H に設定します。
:
*=     0E000H ; ロケーションを E000H に設定します。
```

.PAGE

リスト改ページおよびタイトル指定

書式

`.PAGE [' タイトル']`

内容

- この命令の直前でリストの改ページを行い、オペランドで指定したタイトルをリストのヘッダ部に出力します。タイトルは、' (シングルクォート) 又は " (ダブルクォート) で囲んで記述してください。
- タイトルの文字数は、カラム指定が 80 のとき最大 20 文字、105 から 132 のとき最大 45 文字まで、81 から 104 まではカラム数から 60 を引いた文字数まで、各々使用できます。なおタイトル省略時には改ページのみを行います。

記述例

```
.PAGE   'PROG1' ; PROG1 を PRN ファイルのヘッダ部に出力します。
:
```

.PUB

パブリック指定

書式`.PUB` ビットシンボル又はシンボル又はラベル,**内容**

- オペランドに指定したビットシンボル、シンボル又はラベルを、他のソースファイルから参照可能にします。
- Z セクション及び R セクションのみで構成される RAM 領域のみのファイルでは、総てのラベルはグローバル属性として処理されます。従って、ラベルに対する .PUB 指定は省略することができます。
- 本疑似命令は、ラベル又はシンボルを定義する行より前に記述してください。

記述例

```
.RMOD
.PUB  WORK1,WORK2,WORK3
WORK1: .BLKB  1
WORK2: .BLKB  1
WORK3: .BLKB  1
      :
```

.SECTION

領域指定

書式

`.SECTION` セクション名

内容

- この行以降が、オペランドで指定した名前を持つ領域であることを指定します。
- SRA74M では、オペランドに予約セクション名 (P、R、S、Z) を記述した場合、本疑似命令以降はそれぞれの領域属性として認識され処理されますが、任意のセクション名を記述した場合は、この命令以降の命令により領域属性を決定します。但し、この場合一般ページの ROM 領域、又は一般ページの RAM 領域としてしか認識されません。
- この指定は、他の領域指定命令が現れるまで有効です。
- 同じ名前をもつセクションがファイル内に複数個存在してもかまいません。
- ファイルの先頭に領域指定命令がない場合は、`.SECTION P` が規定値として選択されます。従って、ファイルの最初の ROM 領域に限り、領域指定命令は省略できます。

記述例

```

                .SECTION    DATA    ; DATA セクションを開始します。
datatop:
nulldt: .BLKB    8
        :
                .SECTION    STACK   ; STACK セクションを開始します。
                .BLKB      16
stacktop:
        :
                .SECTION    PROG    ; PROG セクションを開始します。
_init:
        LDX      #stacktop
        TXS
        LDA      #SIZEOF DATA
        LDX      #datatop
        :

```

.SEXT

外部参照指定 (スペシャルページ)

書式

```
.SEXT ラベル又はシンボル [, ラベル又はシンボル, ..., ラベル又はシンボル]
```

内容

- オペランドに指定したラベルを、スペシャルページアドレッシングモード (JSR ㉞のみ)、又はアブソリュートアドレッシングモードで外部参照することを指定します。ゼロページアドレッシングモードでアセンブルしたい場合には、ZEXT で指定してください。
- 本疑似命令は、ラベルを参照する行より前に記述してください。

記述例

```

        .SECTION P
        .EXT   WORK1,WORK2
        .SEXT  SUB
START:
        LDA   WORK1
        JSR   \SUB
        :
```

.VER

プログラムバージョン指定

書式

```
.VER '文字列'
```

内容

- リロケータブルファイルのバージョン指定を行います。
- LINK74M はコマンドパラメータ “-V” が指定されると、各リロケータブルファイル中のバージョン指定の一致を確認します。これによりリロケータブルファイル間のバージョン整合性の確認が行えます。コマンドパラメータ “-V” の詳細については、LINK74M 操作マニュアルを参照してください。
- バージョンの一致は、文字列の比較により確認します。大文字/小文字も区別しますので、注意して記述してください。
- 本疑似命令は、プログラム中に 1 回だけ記述できます。

記述例

```
.VER 'V.1.0' ; バージョン 'V.1.0' を指定します。
:
```

.WORDワードデータ設定

書式

```
[ラベル:] .WORD 数式
```

内容

- 数式のもつ値を設定します (単位:ワード)。
- 2つ以上のデータを設定する場合は、各データを‘,’で区切って指定してください。1行で指定できる個数は16個までです。
- データは下位バイトより設定します。
- オペランドにグローバルラベルを記述できます。

記述例

```
label: .WORD 0E000H ; 00H,E0H を設定します。
      .WORD symbol ; symbol のもつ値を下位バイトより設定します。
      :
```

.ZBEXT外部参照指定 (ゼロページビットシンボル)

書式

```
.ZBEXT ビットシンボル [, ビットシンボル,..., ビットシンボル]
```

内容

- オペランドに指定したビットシンボルを外部参照することを指定します。
- 本疑似命令により指定されたビットシンボルはゼロページ内にあるものと解釈します。
- 本疑似命令は、ラベルを参照する行より前に記述してください。

記述例

```
.ZBEXT BIT0,BIT1,BIT2
      :
```

.ZEXT外部参照指定 (ゼロページ)

書式

```
.ZEXT ラベル又はシンボル[, ラベル又はシンボル,..., ラベル又はシンボル]
```

内容

- オペランドに指定したラベルを、ゼロページアドレッシングモードで外部参照することを指定します。アブソリュートアドレッシングモードでアセンブルしたい場合には.`EXT`、スペシャルページアドレッシングモードでアセンブルしたい場合には.`SEXT`で指定してください。
- 本疑似命令は、ラベルを参照する行より前に記述してください。

記述例

```
        .SECTION Z
        .ZEXT  WORK1,WORK2
        .SEXT  SUB
START:
        LDA   WORK1
        JSR   \SUB
        :
```

第 9 章

SRA74M エラー

9.1 SRA74M エラーの種類

SRA74M 実行時に発生するエラーは、以下の原因によるものがあります。

1. オペレーティングシステムに関するエラー
ディスクやメモリ容量の不足など、SRA74M を実行するオペレーティングシステム環境に関わるエラーです。エラーメッセージ一覧表を参照の上、オペレーティングシステムのコマンドにより対応してください。
2. SRA74M のコマンド行入力に関するエラー
SRA74M 起動時のコマンド行入力に関わるエラーです。本章の内容を確認の上コマンドを再入力してください。
3. アセンブル対象のソースファイルの内容に関するエラー
ラベルの 2 重定義、未定義シンボルの参照などのソースファイルの内容に関わるエラーです。該当箇所のソースファイル内容を修正して再度アセンブルを行ってください。アセンブルエラーを検出した場合 R74 ファイルは生成しません。

SRA74M はエラー及びワーニングを検出すると、図 9.1の形式でエラー内容 (ファイル名、ファイル中の行番号、通し行番号、エラー番号及びエラーメッセージ) を画面と PRN ファイルに出力します。

```
A>SRA74M TEST<RET>
740 Family SRA74M V.1.00.00
Copyright 1996, MITSUBISHI ELECTRIC CORPORATION
AND MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION
All Rights Reserved.

now processing pass 1 ( TEST.A74 )
----*----
now processing pass 2 ( TEST.A74 )
-
  115 E025 EAEA      BCC      LOOP2
TEST.A74 115 ( TOTAL LINE 115 ) Error 18: Relative jump is out of range
  127 E031 EAEA      LDA      #data
TEST.A74 127 ( TOTAL LINE 127 ) Error 22: Value is out of range "data"
----*
  551 E42B EAEA      BRA      TEST2
TEST.A74 551 ( TOTAL LINE 551 ) Error 18: Relative jump is out of range
  593 E4FC EAEAEA    LDA      (work,x          ; data set
TEST.A74 593 ( TOTAL LINE 593 ) Error 23: "()" format error ";"
-----

ERROR   COUNT           00004
WARNING COUNT          00000
TOTAL   LINE ( SOURCE ) 00994 LINES
TOTAL   LINE ( OBJECT ) 00994 LINES
COMMENT LINE ( SOURCE ) 00247 LINES
COMMENT LINE ( OBJECT ) 00147 LINES
OBJECT  SIZE ( Z       ) 00010 (000A) BYTES
OBJECT  SIZE ( R       ) 00053 (0035) BYTES
OBJECT  SIZE ( P       ) 01938 (0792) BYTES

A>
```

図 9.1: SRA74M エラー表示例

9.2 SRA74M システムエラー一覧表

アセンブル中にシステムエラーを検出すると、エラーメッセージを画面に表示しアセンブルを中止します。表 9.1にシステムエラー一覧表を示します。

表 9.1: SRA74M システムエラー一覧表

エラーメッセージ	エラー内容と対策
Usage: SRA74M <filename> [options]	コマンド入力が誤っています。 ⇒ ヘルプ画面を参照して、コマンドを再入力してください。
Can't open xxx ¹	該当ファイルが見つかりません。 ⇒ ソースファイル名を確認して、再入力してください。
Can't create xxx	該当ファイルが生成できません。 ⇒ ドライブ及びディレクトリのパーミッションを確認してください。
Out of disk space	ファイルを出力するためのディスク容量が不足しています。 ⇒ ディスク上に空き領域を作ってください。
Out of heap space	アセンブラが動作するために必要なメモリが不足しています。 ² ⇒ シンボル又はラベルの数を減らすか、ファイルを分割してください。

注意事項

1. MS-DOS においてエラーメッセージのファイル名が \$\$n の場合、CONFIG.SYS の FILES の値が小さいために TMP ファイルを開くことができずにエラーとなることが考えられますので、FILES の値を大きくしてください。
2. SRA74M で扱えるシンボル及びラベルの総数は、SRA74M を実行するシステムでの使用可能メモリ容量に依存しています。

9.3 SRA74M アセンブルエラー一覧表

アセンブルエラーを検出すると、エラーメッセージを画面と PRN ファイルに出力します。表 9.2 にアセンブルエラーとその内容を示します。

表 9.2: SRA74M アセンブルエラー一覧表

エラー番号	エラーメッセージ	エラー内容と対策
1	Already had same statement	ソースファイル中に 1 回しか使用できない疑似命令を、2 回以上使用しています。 例) <code>.LINE 60</code> <code>:</code> <code>.LINE 80</code> ⇒ 宣言を 1 回にしてください。
2	Reference to forward label or symbol	疑似命令が前方のラベル又はシンボルを参照しています。 例) <code>.ORG TOP</code> <code>TOP:</code> ⇒ ラベル又はシンボルの定義を参照以前に行ってください。
3	Division by 0	数式中に 0 による割り算を含んでいます。 ⇒ 数式の記述を確認してください。

エラー番号	エラーメッセージ	エラー内容と対策
4	Illegal operand	オペランドに使用できない文字を含んでいます。 例) LDA #&10 ⇒ オペランドの記述を確認してください。
5	Improper operand type	ニーモニクとオペランドの組み合わせが誤っています。 例) ASL work,Y ⇒ 命令記述形式を確認してください。
6	Invalid label definition	ラベル定義ができない場所でラベル定義を行っています。 例 1) LABEL1: .LINE 60 ⇒ ラベルを削除してください。 例 2) LABEL2: .EQU 100 ⇒ ラベルをシンボルに変更してください。
7	Invalid symbol definition	シンボル定義ができない場所でシンボル定義を行っています。 例) SYMBOL .LINE 60 ⇒ シンボルを削除してください。
8	Out of maximum program size	アドレスが $0FFFF_{16}$ を超えています。 例) .ORG OFFFOH .WORD 1,2,3,4,5,6,7,8,9 ⇒ アドレスが範囲内になるようにプログラムを変更してください。
9	Label or symbol is multiple defined	同一のラベル又はシンボルが 2 回以上定義されています。 例) MAIN: NOP MAIN: NOP ⇒ ラベル名又はシンボル名を確認してください。
10	Nesting error	ネスティングレベルの限界を越えて記述されています。 ⇒ ネスティングレベルの限界を越えないように記述してください。

エラー番号	エラーメッセージ	エラー内容と対策
11	No .END statement	ソースファイル内に.END 文がありません。 ⇒ プログラムの最後に.END 文を記述してください。
12	No symbol definition	シンボルが記述されていません。 例) .EQU 60 ⇒ シンボルを記述してください。
13	No ';' at the top of comment	コメント欄の先頭に ';' (セミコロン) がありません。 例) LDA #CNT counter set ⇒ コメント欄の先頭に ';' を付加してください。
14	Not in conditional block	対応する .IF 文がないのに、.ELSE 又は .ENDIF が記述されています。 (対応する .IF 文がエラーになっている場合もこのエラーが発生します。) 例) .IF DATA1 : .ENDIF : .ELSE : .ENDIF ⇒ .IF 文を確認してください。
15	Operand is expected	命令に必要なオペランドが不足しています。 例) .BYTE ⇒ オペランドの記述を確認してください。
16	Questionable syntax	ニーモニックのスペルに誤りがあります。 例) ADD #DATA ⇒ ニーモニックのスペルを確認してください。
17	Reference to multi defined label or symbol	重複定義されているラベル又はシンボルを参照しています。 例) MAIN: NOP MAIN: NOP BRA MAIN ⇒ ラベル名又はシンボル名を確認してください。

エラー番号	エラーメッセージ	エラー内容と対策
18	Relative jump is out of range	<p>相対ジャンプ命令のジャンプ先アドレスが範囲内にありません。</p> <p>⇒ プログラムの再配置を行うか、命令を変更してください。</p>
19	Label or symbol is reserved word	<p>ラベル又はシンボルに予約語と同一名称が使われています。</p> <p>例) A .EQU 1FFH</p> <p>⇒ ラベル名又はシンボル名を変更してください。</p>
20	Reference to undefined label or symbol	<p>未定義のラベル又はシンボルを参照しています。</p> <p>⇒ ラベル又はシンボルを確認してください。</p>
21	Value error	<p>データ記述形式に誤りがあります。</p> <p>例) ADC #'A</p> <p>⇒ データ記述形式を確認してください。</p>
22	Value is out of range	<p>データの範囲が許容値を越えています。</p> <p>例) ADC #100H</p> <p>⇒ オペランド記述形式を確認してください。</p>
23	"()" format error	<p>左カッコ '(' と右カッコ ')' の数があっていません。</p> <p>例) ADC (WORK</p> <p>⇒ オペランド記述形式を確認してください。</p>
24	Relocatable error	<p>リロケートブルセクションの中で疑似命令 .ORG を記述しています。</p> <p>例)</p> <pre>.SECTION PROG LDA WORK : .ORG 1000H :</pre> <p>⇒ セクションを分けてください。</p>
25	No .SECTION statement	<p>疑似命令 .SECTION を記述していません。</p> <p>⇒ プログラム記述の前に、疑似命令 .SECTION を記述してください。</p>

エラー番号	エラーメッセージ	エラー内容と対策
26	Reference to undefined section	未定義のセクション名を参照しています。 例) LDA #SIZEOF UNDEF ⇒ 該当セクションを確認してください。
27	Section type mismatch	命令又はデータ設定疑似命令 (.BYTE など) が、領域確保命令 (.BLKB など) と混在しています。 例) LDA #WORK .BLKB 1 ⇒ セクションを分けてください。
28	Constant value is required	相対属性のラベル及び外部参照シンボルを用いることはできません。 ⇒ 絶対値を指定してください。

9.4 SRA74M ワーニング一覧表

ワーニングを検出すると、ワーニングメッセージを画面と PRN ファイルに出力します。表 9.3 にワーニングとその内容を示します。

表 9.3: SRA74M ワーニング一覧表

ワーニング番号	ワーニングメッセージ	ワーニング内容と対策
1	Phase warning	<p>1) 疑似命令 .ORG で指定したアドレスが、それ以前のアドレスよりも前になっています。</p> <p>例)</p> <pre>.ORG 0E000H MAIN: LDA WORK :</pre> <pre>.ORG 0C000H</pre> <p>2) 命令がこの行より前方で定義されているラベル又はシンボルを参照しています。</p> <p>例)</p> <pre>LDA WK,X :</pre> <pre>WK .EQU 80H</pre> <p>⇒ ラベル又はシンボルを参照する行より前で定義してください。</p>
2	.END statement in include file	<p>疑似命令 .END をインクルードファイル中に記述しています。</p> <p>⇒ .END をソースファイル中に記述してください。</p>
3	statement has no effect	<p>ステートメントとして意味を持ちません。</p> <p>⇒ プログラムを確認してください。</p>
6	.EQU symbol is multiple defined	<p>疑似命令 .EQU で同一シンボルを 2 回以上定義しています。</p>

第 10 章

LINK74M エラー

10.1 LINK74M エラーの種類

LINK74M 実行時に発生するエラーは、以下の原因によるものがあります。

1. オペレーティングシステムに関するエラー
ディスクやメモリ容量の不足等、LINK74M を実行するオペレーティングシステム環境に関わるエラーです。
2. LINK74M のコマンド行入力に関するエラー
LINK74M 起動時のコマンド行入力に関わるエラーです。本章の内容を確認の上コマンドを再入力してください。
3. リンク対象のリロケータブルファイルの内容に関するエラー
グローバルラベルの 2 重定義、外部ラベルの未定義等のリロケータブルファイルの内容に関わるエラーです。該当箇所のソースプログラム等を確認して、必要があればアセンブルから再実行してください。
4. LINK74M の機能に関するエラー
SRA74M とのバージョン不整合等によるエラーです。

LINK74M は、図 10.1 の形式でエラー内容を画面に表示します。本章のエラー番号順のエラー一覧表を参照の上処理してください。

```
A>LINK74M MAIN SUB,,WORKRAM=100 PROM=C000 DROM,,  
740 Family LINKER V.1.00.00  
Copyright 1996, MITSUBISHI ELECTRIC CORPORATION  
AND MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION  
All Rights Reserved.
```

```
now processing pass 1  
processing "MAIN.R74"  
ERROR NO.2: Out of heap space
```

```
A>
```

図 10.1: LINK74M エラー表示例

10.2 LINK74M エラーメッセージ一覧表

リンクエラーを検出するとエラーメッセージを画面に出力します。表 10.1にリンクエラーメッセージ一覧表を示します。

表 10.1: LINK74M エラーメッセージ一覧表

エラー番号	エラーメッセージ	エラー内容と対策
0	xxx file not found	入力指定されたファイルが見つかりません。 入力指定には、疑似命令.OBJ 又は.LIB によるものも含まれます。 ⇒ ファイル名を正しく入力してください。
1	Invalid command input	入力コマンドのパラメータ数が 5 個以上か、 コマンドの入力パラメータが 2048 文字以上です。 ⇒ 上記の範囲以内で再入力してください。
2	Out of heap space	リンクが動作するために必要なメモリが不足しています。 ⇒ パブリックシンボル数を減らしてください。
3	Invalid section information	セクション情報の指定が誤っています。 ⇒ “セクション名=アドレス” の形式で再入力してください。
4	Invalid parameter input "xxx"	コマンドパラメータの指定が誤っています。 ⇒ 正しく入力してください。
5	Non relocatable file name	リロケータブルファイル名の入力がありません。 ⇒ ファイル名を入力してください。
6	Internal error	LINK74M の内部エラーが発生しました。 ⇒ LINK74M 購入先に御連絡ください。

エラー番号	エラーメッセージ	エラー内容と対策
7	xxx relocatable format is mismatch	.R74 ファイルのリロケートブル形式バージョンが違います。 ⇒ リンク対象の.R74 ファイルを LINK74M と同じバージョンの SRA74M で作成してください。
8	Program version is different	疑似命令.VER で宣言されているプログラムバージョンが一致していません。 ⇒ リンク対象のリロケートブルファイルの疑似命令.VER によるバージョン指定を一致させるか、コマンドパラメータ “_V” の指定を解除してください。
9	Unresolved label "xxx" in xxx	該当セクション内で、外部参照宣言されたラベル又はシンボルが定義されていません。 ⇒ 該当ラベル又はシンボルがパブリック宣言されているプログラムをリンクしてください。
10	"xxx" is multiple defined in xxx. others in xxx	該当ラベル又はシンボルが二重定義です。 ⇒ 該当ラベル又はシンボル名を変更してください。
11	Location overlap. SECTION=xxx ADDRESS=xxx in xxx	該当セクションのアドレス空間がオーバーラップしています。 ⇒ 該当セクションのアドレス配置を確認して、アドレスオーバーラップが発生しないようにしてください(絶対属性のセクションの場合は、ソースファイル中の疑似命令.ORG を変更してください)。
12	SECTION xxx is an absolute	絶対属性のセクションに対して、セクション制御コマンドで先頭アドレスが指定されています。 ⇒ コマンド入力でのアドレス指定をやめるか、該当セクションを相対属性に変更してください。

エラー番号	エラーメッセージ	エラー内容と対策
14	Can't find SECTION xxx	該当セクションが存在しません。 ⇒ セクション情報を正しく指定してください(セクション名は大文字/小文字を区別しますのでご注意ください)。
15	Can't create xxx	該当ファイルが生成できません。 ⇒ コマンドパラメータ “-O” の指定を確認して再入力してください。
16	File seek error xxx	該当ファイルがシークできません。 ⇒ このエラーは、オペレーティングシステムにかかわるエラーです。一般にディスク装置のハードウェア的な異常による場合が多いと考えられます。
17	Expression value is out of range. SECTION=xxx ADDRESS=xxx OFFSET=xxx	該当箇所の演算結果が制限値を超えています(エラー箇所の情報として、セクション名、絶対アドレス、セクションの先頭からのオフセットを表示します)。 ⇒ 制限値を越えないよう、プログラムを変更してください。
18	Out of disk space	ファイルを出力するためのディスク領域が不足しています。 ⇒ ディスク上に空き領域を作ってください。
19	Relative jump is out of range. SECTION=xxx ADDRESS=xxx OFFSET=xxx	該当箇所の相対ジャンプ先アクセス対象に届きません。 ⇒ ジャンプ先のラベルが、範囲内にくるようにプログラムを変更してください(エラー箇所の情報として、セクション名、絶対アドレス、セクションの先頭からのオフセットを表示します)。

エラー番号	エラーメッセージ	エラー内容と対策
22	Out of maximum program size	プログラム領域が最大領域 64K バイト (FFFF ₁₆) を超えました。 ⇒ プログラム容量を縮小してください。
23	Section type mismatch in SECTION xxx	該当セクションで ROM タイプと RAM タイプが混在しています。 ⇒ 同一セクションは、ROM タイプ又は RAM タイプに統一してください。
25	Expression is out of ZERO page. SECTION=xxx ADDRESS=xxx OFFSET=xxx	ゼロページアドレッシングで処理された計算式の結果が、00 ₁₆ から FF ₁₆ の範囲を越えました。 ⇒ 計算結果がゼロページの範囲内になるように変更して下さい。
26	Expression is out of SPECIAL page. SECTION=xxx ADDRESS=xxx OFFSET=xxx	スペシャルページアドレッシングで処理された計算式の結果が、FF00 ₁₆ から FFFF ₁₆ の範囲を越えました。 ⇒ 計算結果がスペシャルページの範囲内になるように変更して下さい。
27	label "xxx" type is mismatch.	外部参照したラベルの種類 (ラベルまたはビットシンボル) が宣言と異なっています。 ⇒ 外部参照の疑似命令を正しく宣言して下さい。
28	Section "xxx" information is out of range.	コマンド入力で指定したセクションの開始アドレスが範囲外です。 ⇒ 開始アドレスを正しく指定して下さい。
29	Bit value is out of range. SECTION=xxx ADDRESS=xxx OFFSET=xxx	ビット値の計算式の結果が 0 から 7 の範囲を越えました。 ⇒ 計算結果が範囲内になるように変更して下さい。

予約語

予約語一覧

記号

..0 ~ ..65535 ラベル
.BEXT 疑似命令
.BLKB 疑似命令
.BYTE 疑似命令
.COL 疑似命令
.D0 ~ .D65535 ラベル
.ELSE 疑似命令
.END 疑似命令
.ENDFUNC 疑似命令
.ENDIF 疑似命令
.ENDIO 予約疑似命令
.ENDM 予約疑似命令
.ENDPROC 予約疑似命令
.ENDRAM 予約疑似命令
.EQU 疑似命令
.ERROR 疑似命令
.EXITM 予約疑似命令
.EXT 疑似命令
.F0 ~ .F65535 ラベル
.FUNC 疑似命令
.I0 ~ .I65535 ラベル
.IF 疑似命令
.INCLUDE 疑似命令
.IO 予約疑似命令
.LIB 疑似命令
.LINE 疑似命令

.LIST 疑似命令
.LISTM 疑似命令
.LOCAL 予約疑似命令
.MACRO 予約疑似命令
.NLIST 疑似命令
.NLISTM 疑似命令
.OBJ 疑似命令
.ORG 疑似命令
.PAGE 疑似命令
.PMOD 疑似命令
.PROCINT 予約疑似命令
.PROCMAIN 予約疑似命令
.PROCSUB 予約疑似命令
.PROGRAMME 予約疑似命令
.PUB 疑似命令
.RAM 予約疑似命令
.REPEAT 予約疑似命令
.REPEATC 予約疑似命令
.REPEATI 予約疑似命令
.RMOD 疑似命令
.S0 ~ .S65535 ラベル
.SECTION 疑似命令
.SEXT 疑似命令
.SMOD 疑似命令
.VER 疑似命令
.WORD 疑似命令
.ZBEXT 疑似命令
.ZEXT 疑似命令
.ZMOD 疑似命令
??0 ~ ??65535 ラベル

A

A アキュムレータ
ADC ニーモニック
AND ニーモニック
ASL ニーモニック

B

BBC ニーモニック
BBS ニーモニック
BCC ニーモニック
BCS ニーモニック

BEQ ニーモニック
BIT ニーモニック
BIT_A0 予約語
BIT_A1 予約語
BIT_A2 予約語
BIT_A3 予約語
BIT_A4 予約語
BIT_A5 予約語
BIT_A6 予約語
BIT_A7 予約語
BMI ニーモニック
BNE ニーモニック
BPL ニーモニック
BRA ニーモニック
BREAK 予約語
BRK ニーモニック
BVC ニーモニック
BVS ニーモニック

C

C キャリーフラグ
CASE 予約語
CLB ニーモニック
CLC ニーモニック
CLD ニーモニック
CLI ニーモニック
CLT ニーモニック
CLV ニーモニック
CMP ニーモニック
COM ニーモニック
CONTINUE 予約語
CPX ニーモニック
CPY ニーモニック

D

D 10 進モードフラグ
DEC ニーモニック
DEX ニーモニック
DEY ニーモニック
DIV ニーモニック
DO 予約語

E

ELSE 予約語
ENDIF 予約語
ENDS 構造化命令
EOR ニーモニック
EVER 予約語

F

FOR 予約語
FST ニーモニック

I

I 割り込み禁止フラグ
IF 予約語
INC ニーモニック
INX ニーモニック
INY ニーモニック

J

JMP ニーモニック
JSR ニーモニック

L

LDA ニーモニック
LDM ニーモニック
LDX ニーモニック
LDY ニーモニック
LSR ニーモニック

M

MUL ニーモニック

N

N ネガティブフラグ
NEXT 予約語
NOP ニーモニック

O

ORA ニーモニック

P

P プログラムカウンタ
PHA ニーモニック
PHP ニーモニック

PLA ニーモニク
PLP ニーモニク

R

ROL ニーモニク
ROR ニーモニク
RRF ニーモニク
RTI ニーモニク
RTS ニーモニク

S

S スタックポインタ
SBC ニーモニク
SEB ニーモニク
SEC ニーモニク
SED ニーモニク
SEI ニーモニク
SET ニーモニク
STA ニーモニク
STP ニーモニク
STX ニーモニク
STY ニーモニク
SWITCH 予約語

T

T X 修飾演算フラグ
TAX ニーモニク
TAY ニーモニク
TST ニーモニク
TSX ニーモニク
TXA ニーモニク
TXS ニーモニク
TYA ニーモニク

V

V オーバフローフラグ

W

WHILE 予約語
WIT ニーモニク
WITH_C 予約語

X

X インデックスレジスタ X

Y

Y インデックスレジスタ Y

Z

Z ゼロフラグ

M3xxxT-PAC ユーザーズマニュアル《リロケータブルアセンブラ-SRA74M 編》

第1版:1997年1月16日発行

第2版:2001年7月26日発行

資料番号:MSD-740PAC-UA-010726

Copyright © 1997, 2001 三菱電機株式会社

Copyright © 1997, 2001 三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社

三菱電機株式会社

三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社

M3XXXXT-PAC 《リロケータブルアセンブラ SRA74M 編》
ユーザーズマニュアル



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668