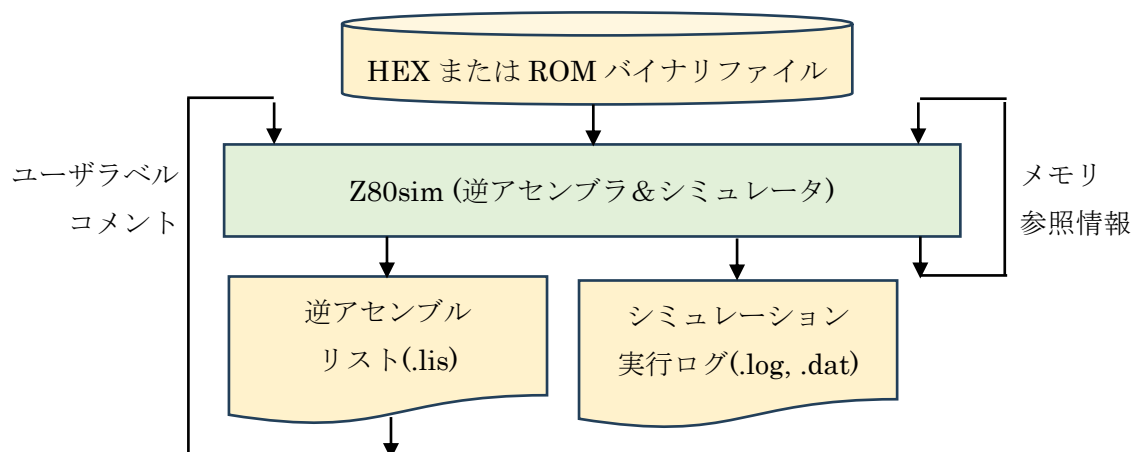


Z80sim : Z80 逆アセンブラ&シミュレータ説明書

(R1.20)

(1) 概要

- ・ Z80 用の HEX ファイルまたは ROM バイナリファイルからの逆解析用です。
- ・ 逆アセンブラは分岐先をたどって解析するため、データ混在があっても正確な逆アセンブル・リストが生成できます。
- ・ メモリデータの参照情報は、静的解析結果だけでなく、シミュレーション時のメモリアクセス結果も反映できます。
- ・ アドレス単位のユーザーラベル定義やコメント追加が容易にできます。
- ・ シミュレータは、1 操作ごとの自動メモリ保存も可能なので巻戻し動作も可能です。
- ・ モニタの推奨解像度は 1920x1080 以上です。
- ・ **jump** 命令をステップオーバーした場合の不具合を修正 ▲R1.20
- ・ メモリ欄の逆順表示に対応(スタック領域用) ▲R1.20



(2) インストール

インストール作業は必要ありません。解凍した Z80sim.exe をクリックして起動します。

アンインストールする場合は、解凍したファイルおよび自動生成される Z80sim.ini を削除してください。

(3) 仕様、制限事項

- ・ 下位アドレスに ROM、上位アドレスに RAM があるシステムに対応します。
ROM/RAM が交互に配置されるシステムには対応しておりません。
- ・ メモリアドレス空間は 64KB です。メモリバンク等には対応しておりません。
- ・ シミュレータで認識可能な IO 空間は 256 バイトです(B レジスタには非対応)。
- ・ エントリ数(関数の入口数)は、16,000 以内。
- ・ ユーザラベルは 31 文字以内、16,000 個以内。
- ・ リファレンス数(参照情報数)は、100,000 以内。シミュレーション時に多量のリファレンスを生成する可能性のある LDIR, LDDR 命令はリファレンス生成対象外です。
- ・ コメント文字数は 1 アドレスに対して 1,000 文字以内、総計 4MB 以内。
- ・ 本版のシミュレータは、次の命令には非対応です。

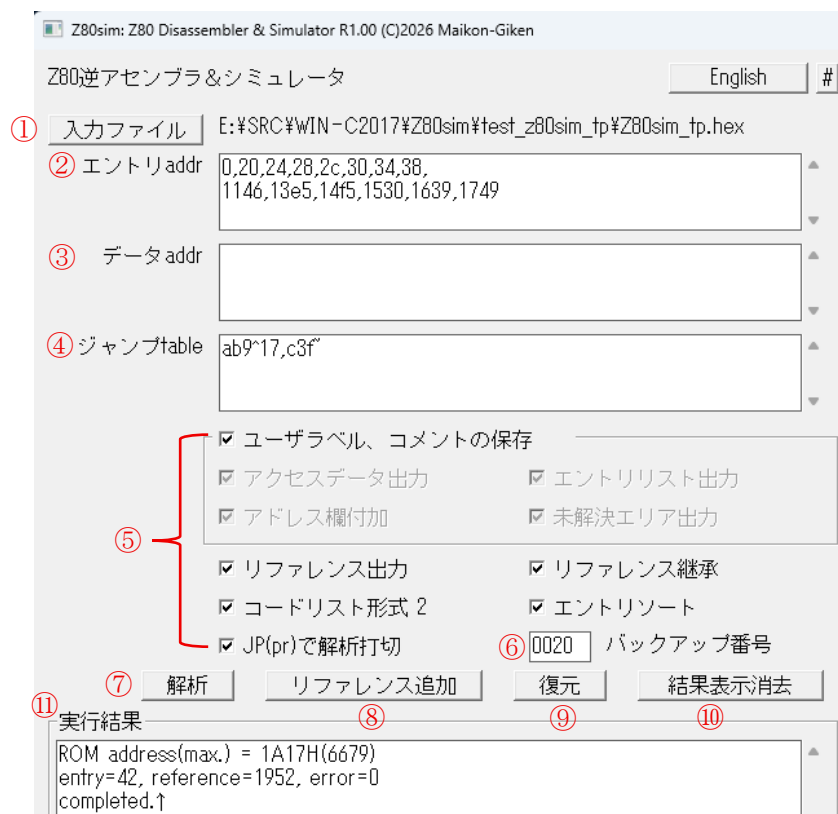
RLD,RRD,CPI,CPIR,CPD,CPDR,

INI,INIR,IND,INDR,OUTI,OTIR,OUTD,OTDR,DAA

- ・ 本版のシミュレータは、フラグは C,Z,S のみ対応です。従って、PO, PE による分岐は正常に動作しません。

(4) 操作方法

(4.1) 逆センブラ部(画面左側)の操作



①入力ファイル

インテル HEX またはバイナリ(ROM イメージ)のファイルを指定します。

拡張子が HEX, hex であれば HEX ファイル、それ以外はバイナリと見なします。

②エントリ addr

エントリは C 言語関数の入口アドレスに相当します。最初は 0(リセット解除アドレス)のみの登録でも構いません。リセット解除後の初期化ルーチンを解読し、割込みアドレスを使用している場合は、それを追加で登録します。

CALL で呼び出しているアドレスは自動的にエントリとして認識しますので、記載の必要はありません。ただしリストに出力する順序を変えたい場合は指定可能です。

さらに JP(HL)によるジャンプ先があれば登録します。ただし ROM のジャンプテーブルを参照している場合は、④ジャンプ table に登録の方が簡単です。

デフォルトではここに記述した順に逆アセンブル・リストを出力しますが、□エントリソートにチェックを付けるとエントリアドレスの小さい順に並び変えて出力します。

【注意】アドレスは 16 進数です。カンマ区切りで並べます。空白は入れないでください。また改行したい場合は、必ずカンマの直後で改行してください。

③データ addr

データとしてアクセスしているアドレスが判明していれば以下のように記述します。アドレスは 16 進数、個数は 10 進数で指定します。

- ・1 バイトデータ型(DB) : 開始アドレス#個数
- ・1 バイト文字型(DS) : 開始アドレス@個数
- ・2 バイトデータ型(DW) : 開始アドレス\$個数

シミュレーションせずに RAM に初期値を設定したい場合は以下の記述にします。

- ・1 バイトデータをコピー : 開始アドレス%個数!コピーする 16 進数値
- ・1 バイト単位領域コピー : コピー元アドレス&個数!コピー先アドレス
- ・2 バイト単位領域コピー : コピー元アドレス=個数!コピー先アドレス

【注意】カンマ区切りで複数の指定が出来ます。空白は入れないでください。また改行したい場合は、必ずカンマの直後で改行してください。

④ジャンプ table

JP(pr)命令のジャンプ先が ROM データとして格納されている場合は、以下の記述により対象アドレスを自動的にエントリアドレスまたは分岐先として認識します。

開始アドレス\$個数	エントリアドレス(アドレスに制約なし)
JP(pr)命令直後のアドレス^個数	分岐先
JP(pr)命令直後のアドレス~	分岐先から必ずここに戻る場合

データ addr の設定と重なった部分は、こちらの設定を優先します。

【注意】カンマ区切りで複数の指定ができます。空白は入れないでください。また改行したい場合は、必ずカンマの直後で改行してください。

⑤チェックボックス

☐ ユーザラベル、コメントの保持

出力した逆アセンブル・リストに追記したラベルやコメントを次のリストに継承したい場合にチェックします。ここをチェックすると以下4つも自動的にチェックしたことになります。

☐ アクセスデータ出力

チェックが無いとデータリストには③データ **addr** で指定した情報のみ出力します。チェックすると静的解析によってデータと判定したアドレスも出力します。

☐ エントリリスト出力

エントリアドレス(関数の入口アドレス)の一覧を表示します。☐ エントリソートのチェックに関わらず、エントリアドレスの小さい順に出力します。

☐ アドレス欄付加

コードリストの全ての行の先頭にアドレスを表示します。

☐ 未解決エリア出力

チェックするとプログラムコードとしてもデータとしても認識されていないアドレスとデータをコードリストに出力します。データ区分は **NA** となります。ただしデータとして **FF** が連続する領域は出力しません。

☐ リファレンス出力

チェックするとデータリストやエントリリストの各アドレスを参照しているアドレスを出力します。参照アドレスが属しているエントリアドレスにユーザラベルが定義されている場合は、アドレス@ラベル名の形式で出力します。

☐ リファレンス継承

チェックすると、前回リストにあるリファレンスを次のリストに継承します。

チェックが無い場合、静的解析で得たリファレンスだけをリストに出力します。

☐ コードリスト形式 2

チェックが無い場合は、タブ(4 文字単位)を使って位置調整したコードリストを出力します。

チェックすると、空白文字を使って位置調整したコードリストを出力します。

☐ エントリソート

チェックが無い場合は、まず②エントリ **addr** に記載した順、次に③ジャンプ **table** に登録した順、そして解析中に **CALL** 命令から自動登録したエントリの順、でコードリストを出力します。

チェックすると、エントリアドレスの小さい順に出力します。

□JP(pr)で解析打切

チェックが無い場合は、JP (HL)命令の次のアドレスも同じエントリに属するコードとして解析を続けます(分岐先アドレスを別の場所に格納している方式に適合)。

JP (HL)命令の次に分岐先アドレスを配置している場合はチェックを付けて解析を打ち切ります。チェックを付けた場合でも④ジャンプ table で分岐先登録した場合は、分岐先を継続して解析します。

⑥バックアップ番号

前回出力した逆アセンブル・リストを自動的に保存しておきたい場合は番号を付けます。0～9999 の範囲で付けます。

解析ボタンまたはリファレンス追加ボタンを押すと、新しいリストを出力する前にこの番号のファイルに保存します。保存すると番号を自動的に+1 します。

復元ボタンを押すと番号を自動的に-1 し、その番号のファイルをリストにします。

【★★★★注意★★★★】別画面のテキストエディタでリストを編集する場合、「保存」せずに解析・リファレンス追加・復元ボタンを押すと編集内容が消えます(エディタから「読み直しますか?」と出た時に読み直さず保存すれば維持できるはずです)。

⑦解析ボタン

逆アセンブル・リストを生成します。□ユーザラベル、コメントの保存にチェック時は、前回リストのラベル、コメントを継承します。□リファレンス継承にチェック時は、前回リストのリファレンスを継承します。

シミュレータ画面のレジスタ欄、IO 欄、メモリ欄は初期化します。

⑧リファレンス追加ボタン

シミュレーションで新たに得られたリファレンスを逆アセンブル・リストに追加します。またメモリ欄に表示している内容をデータリストの RAM 内容として出力します。

【注意】②～④の入力内容に変更があっても反映しません。

⑨復元ボタン

バックアップ番号を指定している場合は、番号を-1 して、その番号のファイルを出力リストとして復元します。

⑩結果表示消去ボタン

実行結果の内容を消去します。

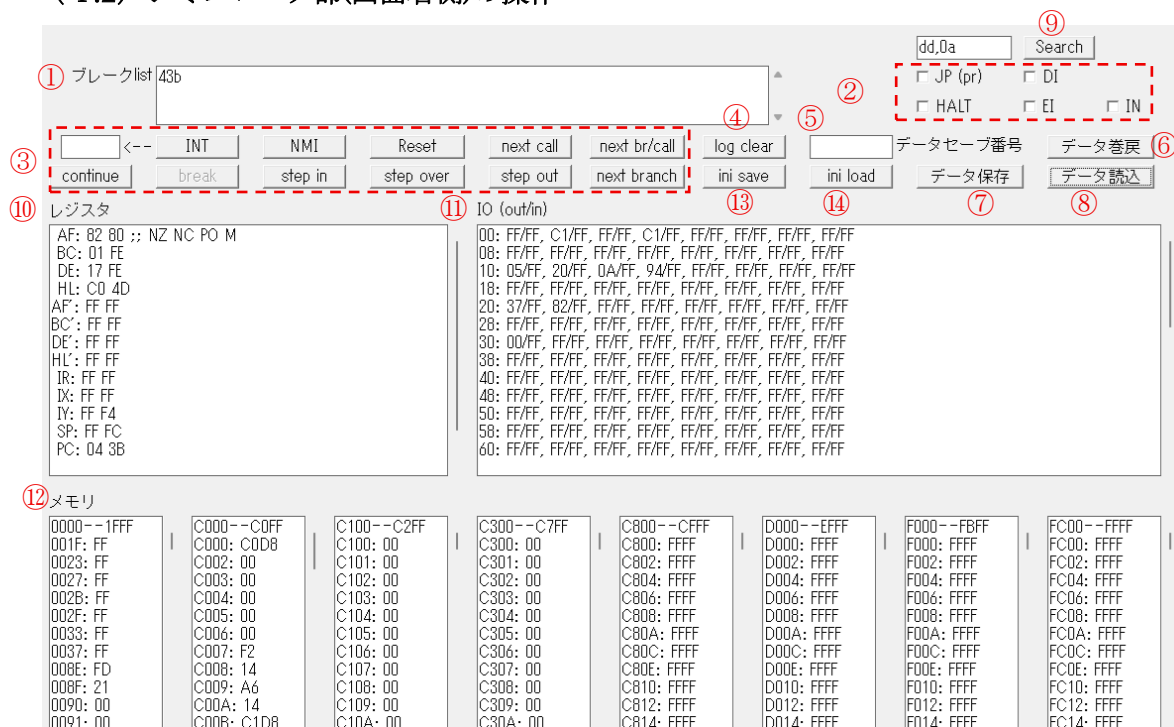
⑪実行結果

エラー情報やサマリ情報を表示します。

⑫ini save(パラメータ保存)/ini load(パラメータ復元)ボタン

各種設定内容を保存／復元できます。→ シミュレータ部参照

(4.2) シミュレータ部(画面右側)の操作



①ブレーク list

カンマ区切りで複数のブレーク条件を指定できます。1 つでも条件が合致すればシミュレーションを停止します。空白は入れないでください。また改行したい場合は、必ずカンマの直後で改行してください。

条件の記述方法(アドレスは 16 進数、回数は 10 進数)

- ・ 命令実行回数の指定(リスト先頭限定) @命令実行回数
- ・ 命令実行アドレス検出 アドレス@検出回数

1 回目で停止の場合はアドレスのみ記述で良いです。命令実行直前で停止します。

- ・ データアクセス検出 アドレス AKD@アクセス回数

指定のメモリまたは IO をアクセス終了後に停止します。

A : アクセス方法を限定したい場合は、以下の記号を記述します。

w メモリ書込み r メモリ読出し o IO 書込み i IO 読出し

限定が無ければメモリの読み書きが対象になります。

K : データ内容を限定したい場合は、以下の記号を記述します。

1 バイト比較 = 等しい < 小さい > 大きい ! 等しくない

2 バイト比較 == 等しい << 小さい >> 大きい !! 等しくない

D : 16 進数で比較値を記述します。

@アクセス回数 : 1 回目で停止の場合は省略できます。

②停止命令のチェックボックス

チェックを付けた命令を検出すると、実行直前に停止します。

③シミュレーション実行関連ボタン群

[continue] レジスタ画面の PC に表示されているアドレスから実行を開始します。

実行開始後は、ブレーク list、停止命令チェック、[break] ボタンが有効になります(以下のボタンでも同様)。

[step in] 1 命令だけ実行して停止します。

[step over] CALL 命令の場合は、CALL 命令の次のアドレス直前まで実行して停止します。CALL 命令以外は、1 命令だけ実行して停止します。

[step out] 現在のエントリを終了する命令(RET 命令)を検出したら停止します。

CALL によって呼ばれた別エントリの RET 命令は停止せず実行します。

[next call] CALL 命令を検出するまで実行します。

[next branch] 分岐命令を検出するまで実行します。

[next br/call] 分岐命令または CALL 命令を検出するまで実行します。

[Reset] レジスタ欄の PC の値を 0 にして実行を開始します。

[NMI] アドレス 66H を CALL します。

[INT] 左の欄で指定したアドレスを CALL します。ただし割込み禁止状態の場合は、許可状態になるまで保留します。保留中は左の欄は非アクティブになります。保留中にもう一度[INT]ボタンを押すと割込み要求をキャンセルします。

[break] シミュレーションを停止します。

④ログ消去ボタン

シミュレーションを実行すると必ず自動生成したログファイル(.log)に結果を追記します。消したい場合はこのボタンを押します。保管したい場合はファイル名をリネームしてください。

⑤データセーブ番号

0～8 桁以内の数値を指定すると、シミュレーション実行直前ごとにレジスタ欄、IO 欄、メモリ欄の表示内容を自動的に保存します。保存すると番号を+1 します。

⑥データ巻戻ボタン

押すとデータセーブ番号を-1 し、その番号の保存内容をレジスタ欄、IO 欄、メモリ欄に読み込みます。

⑦データ保存ボタン

レジスタ欄、IO 欄、メモリ欄の表示内容をファイル名を指定して保存します。

⑧データ読込ボタン

指定したファイルからレジスタ欄、IO 欄、メモリ欄の内容を復元します。

⑨Search ボタン

左のボックスで指定した値がメモリ内に無いか探します。2 バイトから 16 バイトの範囲で、カンマで区切って 1 バイトごとに 16 進数で指定します。

⑩レジスタ欄

レジスタの内容を表示します。シミュレーション停止中は内容を書き換え可能です。

【注意】書換える場合、フォーマット(空白位置など)を変更しないようにします。

⑪IO 欄

IO の内容を表示します。シミュレーション停止中は内容を書き換え可能です。

出力/入力順に並んでいます。

【注意】書換える場合、フォーマット(/や空白位置など)を変更しないようにします。

⑫メモリ欄

8 つのメモリ領域を表示します。シミュレーション停止中は内容を書き換え可能です。表示するメモリ範囲は、先頭 1 行目で次のように指定します。

開始アドレス—終了アドレス

アドレスは必ず 4 桁で指定し、必ず 2 個の・で区切ってください。指定が有効になるのは、シミュレーション実行後もしくは解析ボタンを押したときです。

データは、アドレス 4 桁：空白 データ(1 または 2 バイト) の並びです。1 バイトで表示されているデータを 2 バイトに書き換える場合は、次の行(次のアドレス)を削除する必要があります。逆に 2 バイトで表示されているアドレスに 1 バイトのデータを書く場合は、次の行に次のアドレスとデータを追加する必要があります。

開始アドレスが終了アドレスより大きい場合(例えば FFFF—F000)、アドレスを逆順に表示します。データ単位は 2 バイトに限定されます▲R1.20

【注意】書換える場合、フォーマット(空白位置など)を変更しないようにします。

⑬ini save(パラメータ保存)ボタン

ブレーク list やエントリ addr など画面上の各種設定値をファイル名を指定して保存します。

⑭ini load(パラメータ復元)ボタン

指定した ini ファイルから画面上の各種設定値を復元します。

(5) 逆センプル・リストの見方とラベル、コメント追記方法

(5-1) 逆アセンプル・リストの見方

リストは次の 5 セクションから構成しています。

PARAMETER : エントリ addr、データ addr、ジャンプ table および各種設定

DATA LIST : データとしてアクセスするメモリ領域のアドレス詳細

ENTRY LIST : エントリ addr、ジャンプ table および自動検出した CALL アドレスの一覧。□エントリリスト出力にチェックを付けてない場合は出力しません。

CODE LIST : 逆アセンプル・リスト本体

USER LABEL LIST : ラベル名順のリスト。□ユーザラベル、コメントの保存にチェックを付けてない場合は出力しません。

①PARAMETER

逆アセンプラ部の各種設定値の他に、下記の例の形式で解析サマリを出力します。

ROM address(max.) = 1A17(6679)

Total entry = 42, Detected reference = 1952, Backup number = 0019

ERROR count = 0, LABEL duplication = 0

②DATA LIST

アドレスごとに以下の順で出力します。

アドレス : データ区分 データ ;=ラベル ;<リファレンス ;;コメント

アドレス 16 進数 4 桁

データ区分 DA=ジャンプ table で指定したジャンプ先、

DB=1 バイト数値データ、DS=文字データ、DW=2 バイト数値データ、

EQ=アドレスを引用しているが、ジャンプ先かデータかは不明、

NA=未解析アドレス(FF が連続する場合は出力しません)、

データ DA,DW は 2 バイトの並び、EQ は状況により 2 バイトまたは 1 バイト、
その他は 1 バイトの並び。DA, DW, EQ のデータに対してラベルがある
場合は、データの直後に{ラベル}を出力します▲R1.20

ラベル アドレスに対して定義されたラベル

リファレンス このアドレスを参照している命令のアドレス並び。その命令が属
するエントリにラベルが定義されている場合は、「アドレス@ラベル」と
表示。

コメント 行末までがコメント扱い

同じデータ区分でラベル、リファレンス、コメントのいずれも無いアドレスが連続する
場合は、データをカンマで区切って並べて出力します。

③ENTRY LIST

アドレスごとに以下の順で出力します。

アドレス： 呼出し区分 属性 ;=ラベル ;<リファレンス ;;コメント

アドレス 16 進数 4 桁

呼出し区分 JUMP=ジャンプ table で指定したジャンプ先、
CALL=CALL 命令で呼び出されているアドレス、
LIST=エントリ addr で指定したアドレス

属性 J ジャンプ table 指定
G CALL 命令もしくはエントリ addr 指定
L JP,JR 命令の飛び先にもなっている場合

ラベル アドレスに対して定義されたラベル

リファレンス このアドレスを参照している命令のアドレス並び。その命令が属
するエントリにラベルが定義されている場合は、「アドレス@ラベル」
と表示。

コメント 行末までがコメント扱い

④CODE LIST

エントリ(C 言語の関数に相当)ごとにヘッダ行があり、それに続けて逆アセンブルし
たコードが並びます。

ヘッダ行にはリファレンスを出力しています。

各コード行はアドレスごとに以下の順で出力します。

アドレス | ラベル： オペコード オペランド ;*各種情報 ;;コメント

アドレス 16 進数 4 桁

ラベル ユーザ定義ラベルがあればそれを出力
ユーザ定義ラベルの無いジャンプ先には自動生成ラベルを出力

オペコード Z80 ニモニック

オペランド 数値の場合はユーザ定義ラベルまたは 16 進数で出力
分岐先や CALL 先は、ユーザ定義ラベルか自動生成ラベルを出力

各種情報 ILIGAL JP/CALL ADDRESS 飛び先が命令の 2 バイト目以降
JUMP TO OTHER ENTRY 飛び先が他のエントリ(関数)
CONTINUE TO OTHER ENTRY 他のエントリ(関数)に続く
OPRAND ALREADY CHECHED オペランドが他で解析済み
CAN NOT GET THE ADDRESS JP(pr)命令を検出

コメント 行末までがコメント扱い

⑤USER LABEL LIST

ラベル名順に以下の順で出力します。

アドレス；ラベル

(5-2) ラベル、コメント追記方法

①ラベルの追記方法

デーリスト、エントリリスト、コードリストの各行の内容の後ろにユーザーラベル定義を記述することができます。:=と記述し続けて 31 文字以内のラベル名を記述します。

使用可能文字：英数字 () * + - . / < = > ? [] ^ _ ! # \$ % & { | } ~▲R1.20

ラベルの後ろは、改行、空白、タブのいずれかにしてください。

エントリリストとコードリストで同じアドレスに対して別ラベル名を定義した場合は、コードリストの定義を優先します。

リファレンス(<に続く内容)の後ろにラベル定義を記述した場合、次のリストではリファレンスの前に移動します。

データリストで、アドレス表示されていないデータにラベルを付ける場合は、手書きで新しい行を追加します。書式は次の通りです。

アドレス:¥t データ区分¥t データの値 :=ラベル

アドレス 16 進数 4 桁

: 半角の:

¥t タブコード

データ区分 DA=アドレスを示す 2 バイト、DW=2 バイト数値、

DB または DS=1 バイト数値または文字

データ値 ダミー指定なので適当に FFFF とか記載します。

:= この手前は空白またはタブコードとしてください。

②コメントの追記方法

::から行末までをコメントとして登録します。

デーリスト、エントリリスト、コードリストの各アドレス行に対応して登録します。

エントリリストとコードリストで異なるコメントを付けた場合は、コードリストの内容を優先します。

データリストで、アドレス表示されていないデータにコメントを付ける場合は、手書きで新しい行を追加します。方法は①を参照してください(:=を::に変更)。

【注意】改行の直前にカンマを記述しないでください。次の行もコメント扱いになります。

(6) 解析の手引き

- ①最初に「エントリ addr」として 0 を登録して「解析」します。

これにより少なくとも初期化ルーチンが逆アセンブルできるはずです。

- ②初期化ルーチンの中に割込みに関する記述があれば、そのアドレスを「エントリ addr」に登録して「解析」します。

- ③次に逆アセンブル・リスト(CODE LIST)の中から「CAN NOT GET THE ADDRESS」と表示されている行を探します。JP(HL)で分岐しているので、その付近のコードを解読して飛び先を推定します。

一番わかりやすいのは、ROM エリア上のベース・アドレスから計算によって分岐先を格納しているアドレスを読込んで HL に格納しジャンプするコードです。この場合は、DATA LIST を見てベース・アドレスから何個が飛び先アドレスか推定します。このベース・アドレスと個数を「ジャンプ table」に登録して「解析」します。

分かりにくいのは、RAM エリアの変数から飛び先を読込んで HL に格納しジャンプするコードです。変数の設定を近くで行っていれば容易に解読できますが、マルチタスクっぽいコードの場合は遠くで設定している場合が多く、その場合は DATA LIST からあたりを付けます。データ区分 EQ に続いてデータ区分 NA が続いているデータ群は、EQ のアドレスをエントリとするコードの可能性があります。特に NA の中に RET 命令(C9)が出現していたら、かなりの確率でコードと考えられます。この EQ のアドレスを「エントリ addr」に登録して「解析」します。

- ④全部のコードを解析できなくてもシミュレータを動かすことができます。最初は Reset で開始します。もし JP(HL)の飛び先が未解析のアドレスであればそこで停止するので、そのアドレスを「エントリ addr」に登録して「解析」すれば良いです。

(7) 重要な更新情報

R1.20 2026/08/23 jump 命令をステップオーバーした場合の不具合を修正

R1.00 2026/08/03 新規公開

(8) サポート

問い合わせ先: L00-micon あとまく memoad.jp (注: あとまくを@に変えて下さい)
できるだけタイトル先頭に【サポート依頼】を付けて下さい。

72 時間経過しても何の返事もない場合は、再メールをお願いします。

追加情報がある場合は、[マイコン技研](#) の「技術資料」ページに掲載します。

2026 年 8 月 23 日 マイコン技研 澤田 明