

## WZeta アセンブラ wzasm 仕様

wzasm は WZeta 命令セットのアセンブラです。

### 1 定義領域

コードの先頭で宣言文を記述します。

MEMORY、MODEL、USE

MEMORY	CPU に搭載されるメモリ
MODEL	搭載されたメモリをプログラムとデータに配分する方法
USE	オプションの設定

#### 1-1 MEMORY

定義領域に次のように記述

MEMORY n
----------

n: 2~128、192 の数字。n KB のメモリを搭載。192KB を指定した場合、メモリモデルは先頭から 128KB がプログラム、続く 64KB がデータとなります。

#### 1-2 MODEL

定義領域に次のように記述

MODEL xxx
-----------

TINY	プログラムメモリとデータメモリで同じアドレスを共有
HALF	データメモリの先頭を搭載メモリの 2 分の一とする
1/2	データメモリの先頭を搭載メモリの 2 分の一とする
1/n	データメモリの先頭を搭載メモリの後ろから n 分の一とする

#### 1-3 USE

OFFSET	プログラムで使用する OFFSET(レジスタ・バンク)を宣言
INTREG	TINY モデル専用。INT 命令に割り当てられている 32~63 のメモリアドレスを不使用な場合、データレジスタとして使えるようにする。
PREFIX	PREFIX 命令で使用するメモリを宣言して自動割当されないようにする

OFFSET

定義領域に次のように記述

USE OFFSET n
--------------

非 TINY モード

n=1	OFFSET 0,1 を使用
n=2	OFFSET 0,1,2 を使用
n=3	OFFSET 0,1,2,3 を使用

TINY モード

n=T0123	OFFSET 0,1,2,3 を使用。.HEAD 領域は使用不可
n=T023	OFFSET 0 と 2、3 を使う。.HEAD 領域は 128～255 バイト
n=T03	OFFSET 0 と 3 を使う。.HEAD 領域は 128～383 バイト

INTREG

定義領域に次のように記述

USE INTREG <16bit の数字>
------------------------

bit n が 1 の場合、メモリアドレス  $32+n \times 2$ 、 $32+n \times 2+1$  が REG で割当可能

bit 0 が 1 の場合、メモリアドレス 32、33 が REG で割当可能となる。

bit 1 が 1 の場合、メモリアドレス 34、35 が REG で割当可能となる。

bit 15 が 1 の場合、メモリアドレス 62、63 が REG で割当可能となる。

PREFIX

定義領域に次のように記述

USE PREFIX n0 n1 n2 ... nx
----------------------------

PREFIX 命令で使うことがないメモリを指定した場合、エラーになる

文字定義

#	以降をコメント
^	ラベルの参照
&	MEM、FAR のデータの参照。2 バイトのデータアドレス 2 バイトの場合、変数名の最後に .H を付けると上位バイトのアドレス .L を付けると下位バイトのアドレス
%	REG データの参照。データアドレスではなく、そのアドレスが示すメモリを指す。 [m]形式。メモリをレジスタのようにプログラムできる。
@	定数に名前を付ける。

## セクション設定

.DEAD	6～31 バイトの領域。デッドスペースの意味から DEAD になっている。
.INT0	割込み 0 番のアドレスを先頭とした領域 INT 命令のジャンプテーブルのコードを記述可能
.INT1	割込み 1 番のアドレスを先頭とした領域
.HEAD	メモリの先頭にある小さなプログラム領域。 先頭近くのコードには 1 バイトアドレスの分岐が可能。
.HMAC	ハードマクロ領域
.PROG	プログラム領域
.DATA	データメモリにデータを格納するデータを置く アドレスはメモリモデルによって決まる。

## メモリマップ

### TINY モデル

アドレス	内容
0	ハードマクロのオペランド
1	0、つまり NOP 命令。0 番地のメモリに依存することなく NOP 命令
2-5	.PROG への分岐命令が置かれる。(4 バイトの分岐命令)
6-31	データ
32-63	割込のジャンプテーブル。データ可能
64-127	データ(127 は 1 バイト指定ではアクセス不可)
128-511	.HEAD 領域
512 以降	.HMAC 領域 .PROG 領域、続いて.DATA 領域

### HALF モデル 搭載メモリ 2KB の場合

アドレス	内容
0-5	.PROG への分岐命令が置かれる。
6-31	.DEAD 領域。 .DEAD、 .INTx がなければ.HEAD 領域開始。
32-511	32-63 は割込のジャンプテーブル。 .INT0、 .INT1 につづいて.HEAD 領域
512-1023	.HMAC 領域に続いて.PROG 領域
1024-	.DATA 領域

## DATA セクションの記述形式

REG n (変数名) REG n (変数名) 初期値 ... REG n (変数名) &固定アドレス REG n (変数名) &固定アドレス 初期値 REG n (変数名) !アライメント REG n (変数名) !アライメント 初期値	データ領域の先頭から 126 バイトまでのメモリを割当ます。1 バイトのアドレスで指定できる。0 バイト目はハードマクロ命令が利用するため 1 バイト目から割り当てられる。TINY モデルでは 12 バイト目から。INTx のプログラムで使われる領域を避けて割り当てられる。
MEM n (変数名) MEM n (変数名) 初期値 ... MEM n (変数名) &固定アドレス MEM n (変数名) &固定アドレス 初期値 MEM n (変数名) !アライメント MEM n (変数名) !アライメント 初期値	REG の割当ての後に MEM の割当てを開始する。データ領域の先頭から 127 バイトから 0x7FFF も割当てが可能。
FAR n (変数名) FAR n (変数名) 初期値 ... FAR n (変数名) &固定アドレス FAR n (変数名) &固定アドレス 初期値 FAR n (変数名) !アライメント FAR n (変数名) !アライメント 初期値	REG、MEM の割当ての後に FAR の割当てを開始する。 データ領域の 0x8000～0xFFFF も割当ることが可能
STR NEAR (変数名) (文字列) STR FAR (変数名) (文字列)	NEAR は MEM を使って領域確保、FAR は FAR を使って領域確保。(文字列)はダブルクォーテーションで囲われた文字列。文字列の直後には'¥0'が追加される。
ALIAS (alias 名) (original 名) [offset]	別名定義(エイリアス) offset は original からの加算値(省略可能)

n は 1～0x10000 までの数字

初期値は 1 バイト単位、スペースで区切る。リトルエンディアン。

初期値の先頭のみ¥を記述して改行して初期値を記述することが可能。

固定アドレスを割当てる場合はアドレスの前に&をつけて記述

アライメントされたアドレスを割当てる場合はアドレスの前に!をつけて記述

REG で割当てた変数名は%を付けると[m]のメモリを表す。  
変数名の頭に L&を付けた場合、割当てたアドレスの下位バイト  
変数名の頭に H&を付けた場合、割当てたアドレスの上位バイトをしめす

MEM 1 SUM

として SUM の値を A レジスタのロードするには

LD B,&SUM.L

LD A,[&SUM.H,B]

FAR 1 SUM

として SUM の値を A レジスタのロードするには

LD A,&SUM.H

LD B,&SUM.L

LD A,[A:B]

疑似命令

WZeta の命令セットにはない命令ですが命令とほぼ同じ疑似命令

名前	展開する命令列
EXIT n	wzasm で-e オプションを使うと n のアドレスに相対ジャンプ -e オプションを付けない場合 JR 0 と同じ シミュレーションなどでの利用に便利。verilog や C のシミュレータでは C レジスタの内容を exit code として扱う。正常終了の場合は 0、0 以外の場合はシミュレーションのプロセスを異常終了させる場合がある。

## C シミュレータ専用疑似命令

C 言語によるシミュレータ wzsims のみ有効な疑似命令。アセンブル結果に命令コードを出力しない。レジスタやメモリの内容を表示させる。表示のタイミングは疑似命令の次にある命令が実行される前であることを注意。条件分岐命令の直後に疑似命令を置いても分岐してしまうと疑似命令は実行されず表示されない。

名前	展開する命令列
\$PRINT A	A レジスタの内容を画面(ファイル)に出力
\$PRINT B	B レジスタの内容を画面(ファイル)に出力
\$PRINT C	C レジスタの内容を画面(ファイル)に出力
\$PRINT ABC	ABC レジスタの内容を画面(ファイル)に出力
\$PRINT PSW	PC、MC、IC、ABC レジスタの内容を画面(ファイル)に出力 PC: プログラムカウンタ MC: ハードマクロプログラムカウンタ IC: 割込みプログラムカウンタ
\$PRINT (アドレス) n	メモリの内容を n バイト表示
\$PRINT (変数名) n	メモリ上に確保された変数のデータを n バイト表示
\$CAPI n	ファンクション番号 n 番の関数をシミュレータが実行

## V シミュレータ専用疑似命令

verilog 言語によるシミュレータでのみ有効な疑似命令。実際の命令コードは NOP 命令。wzsims は NOP のオペランドを解釈してレジスタの出力処理などデバッグに役立てる。実機では NOP 命令として処理されるため実機でコードを、そのまま実行することも可能。

名前	展開する命令列
PRINT n	A、B、C レジスタの内容を表示。 アドレス $(n-1) \times 256 \sim n \times 256 - 1$ までの 256 バイトの値を表示

## 禁止命令

SETMC 命令は割込み中のみ利用可能な命令です。SETPC 命令も同様ですが、ハードマクロ命令中でも動作します。ただしハードマクロ命令終了のマクロビットと一緒に SETPC 命令をすることはできません。通常命令

通常プログラム領域(.PROG)に置かれた命令を割込み中実行する場合があるので.PROGに SETPC 命令を置くことは可能ですが、実行しても確実に正しい動作をしません。