

付録 . A K I T - M I P S 3 2 / 4 K c - T P 内部コマンド

本書は、K I T - M I P S 3 2 / 4 K c - T P の内部コマンドについて記述しています。これらのコマンドは、デバッガの中でスルーコマンドとして使用できます。スルーコマンドの使用の可否及び使用方法は各デバッガのマニュアルを参照ください。（デバッガによっては使用できない場合もあります）

P A R T N E R / W i n の場合

> &	<< スルーコマンドへの移行します。
> # E N V	<< 内部コマンドの入力です。
> &	<< スルーコマンドモードを終了します。

G H S - M u l t i の場合

R T E S E R V を接続後、ターゲット・ウインドウで直接入力できます。

コマンド一覧

付録 . A K I T - M I P S 3 2 / 4 K C - T P 内部コマンド	
コマンド一覧	1
コマンド書式	2
アクセス系ブレーク	: A B P , A B P 1 , A B P 2 コマンド 3
キャッシュ操作	: C A C H E I N I T , C A S H E F L U S H コマンド 4
環境設定	: E N V コマンド 5
ヘルプ	: H E L P コマンド 7
実行系イベント	: I B P , I B P 1 , I B P 3 , I B P 4 コマンド 8
ポート入力	: I N B , I N H , I N W , I N D コマンド 9
初期化	: I N I T コマンド 10
J T A Gリード	: J R E A D コマンド 11
デバッガキャッシュの解除	: N C コマンド 12
デバッガキャッシュの設定	: N C D コマンド 13
ソフトブレーク禁止領域の設定	: N S B P コマンド 13
ソフトブレーク禁止領域の設定	: N S B P D コマンド 14
強制ユーザ領域の設定	: N R O M コマンド 15
強制ユーザ領域の設定	: N R O M D コマンド 16
ポート出力	: O U T B , O U T H , O U T W , O U T D コマンド 17
C P Uリセット	: R E S E T コマンド 19
E . R O Mの環境設定	: R O M コマンド(RTE-1000-TP 用) 20
E . R O Mの環境設定	: R O M1..R O M4 コマンド(RTE-2000-TP 用) 21
T L B	: T L B コマンド 23
シンボル	: S Y M F I L E , S Y M コマンド 24
バージョン表示	: V E R コマンド 25

ご注意：

- 1 . これらのコマンドはご使用になりたい機能がデバッガ本体に有していない場合にのみ補助的にご使用ください。ご使用になるデバッガで同等の機能を有している場合にこれらのコマンドを発行した場合、デバッガとの間で競合を起こしあいの動作が異常になる場合があります。

コマンド書式

内部コマンドの基本書式を以下に示します。

コマンド名 パラメータ

* パラメータ書式で [] は省略可能を示し、| は択一を意味します。

コマンド名はアルファベットの文字列でパラメータとの間はスペースまたはタブで区切れます。パラメータはアルファベットの文字列または16進数を指定し、各パラメータ間はスペースまたはタブで区切れます。（16進数には演算子は使用できません。）

a b p , a b p 1 , a b p 2 コマンド

[書式]

```
abp
abp{1|2} [ADDR [AMASK]] [data DATA [DMASK]] [asid ASID|noasid]
[aeq|aneq] [deq|dneq] [read|write|accs] [byte|hword|word]
abp{1|2} /del
```

[パラメータ]

abp{1|2}: アクセス系のブレーク条件のチャンネル(1-4)をibpに続けて指定します。

ADDR: アドレスを16進数で指定します。

ADDR [AMASK]: アドレス条件の指定

ADDR: アドレスを16進数で指定します。

AMASK: アドレスのマスクデータを16進数で指定します。
“1”のビットは比較の対象になりません。

data DATA [DMASK]: データ条件の指定

DATA: データを16進数で指定します。

DMASK: データのマスクデータを16進数で指定します。
8bit単位で”00”または”FF”を入力してください。
”FF”の8ビットは、比較の対象になりません。

asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。

aeq|aneq: アドレスの比較条件を指定します。

aeq: アドレスをイコールで比較します。

aneq: アドレスをノットイコールで比較します。

deq|dneq: データの比較条件を指定します。

deq: データをイコールで比較します。

dneq: データをノットイコールで比較します。

read|write|accs: サイクルの条件を指定します。

read: リードサイクルを指定します。

write: ライトサイクルを指定します。

accs: リードまたはライトサイクルを指定します。

byte|hword|word|nosize: アクセスサイズの指定します。

byte: バイトアクセス(8-bit)を指定します。

hword: ハーフワードアクセス(16-bit)を指定します。

word: ワードアクセス(32-bit)を指定します。

nosize: 無効を指定します。

abp{1|2} /del: 条件の解除を行います。

/del: 解除を指定します。

[機能]

2点ある、アクセス系のブレークポイントの設定または解除します。

[使用例]

```
abp1 a0001000 0 data 5555 0 aeq deq read hword
a0001000h番地から5555hをリードした時にブレークします。
```

```
abp1 a0001000 0 data 5555 ff00 aeq deq read hword
a0001000h番地からxx55hをリードした時にブレークします。(xxは、Con't Careの意味です)
```

```
abp1 /del
abp1の条件を解除します。
```

cacheinit, cacheflushコマンド

[書式]

```
cacheinit  
cacheflush [ADDRESS [LENGTH]]
```

[パラメータ]

cacheinit キャッシュの初期化を行います。ライトバックは行いませんので、キャッシュの内容は破棄されます。

cacheflush 指定した範囲のキャッシュのフラッシュを行います。ライトバックが指定されている場合は、ライトバックサイクルが発生します。

ADDR: 開始アドレスを16進数で指定します。

LENGTH: フラッシュする空間のバイト数を16進数で指定します。

[機能]

キャッシュ操作のためのコマンドです。

[入力例]

```
cacheflush 80000000 1000  
flush cache addr=80000000 len=00001000  
0x80000000 0x1000バイトのキャッシュの内容をフラッシュします。
```

env, ememstatコマンド

[書式]

```
env [[!]auto] [jtag{25|12|5|2|1|500|250|100}] [[!]nmi] [[!]int] [[!]reset] [[!]verify]
ememstat
```

[パラメータ]

[!]auto: 実行中にブレークポイントを設定した場合一時的にブレークしますが、その後の実行を自動的に行う場合に[Auto], 行わない場合に[!auto]を指定します。
 jtag{25|12|5|2|1|500|250|100}: N-WireのJTAGクロック指定します。それぞれ以下に対応します。
 [25MHz|12.5MHz|5MHz|2MHz|1MHz|500KHz|250KHz|100KHz]

備考: 通常は25MHzまたは、12.5MHzでご使用ください。1MHzより低い周波数を指定した場合は、デバッガの動作が著しく遅くなったり、異常になる場合があります。

[!]nmi: NMI端子のマスク指定を指定します。!はマスクしないを意味します。
 [!]int: INTxx端子のマスク指定を指定します。!はマスクしないを意味します。
 [!]reset: reset端子のマスク指定を指定します。!はマスクしないを意味します。
 [!]verify: メモリへの書き込み時にリードアウトしてベリファイするかどうか指定します。!はベリファイしないを意味します。

備考: ROMをエミュレーションしている領域に対しても、CPUからリード(j read相当)しますので、ダウンロード時のテストにも有効です。但し、処理速度が遅くなります。

[機能]

envコマンドは、エミュレーション環境の設定とDCUの状態を表示します。設定は変更が必要なパラメータだけを入力ください。入力の順序は任意です。但し、同じパラメータを2回入力した場合は、後から入力した値が有効です。

ememstatコマンドはRTE-2000-TPの場合に、E.MEM基板の実装状態を表示するコマンドです。

以下に表示例を示します（初期値の状態）

RTE-1000-TPの場合

```
CPU Settings:
  Auto Run      = ON (auto)
  JTAGCLOCK    = 25MHz (jtag25)
  Verify        = verify off (!verify)

Signals Mask:
  NMI           = NO MASK (!nmi)
  Soft/Hard INT= NO MASK (!int)
  Soft RESET   = NO MASK (!reset)
```

RTE-2000-TPの場合

```
Probe:  
  Unit      : RTE-2000-TP  
  Rom Probe : (use ememstat command)  
  Emem Size : (use ememstat command)  
CPU Settings:  
  Auto Run    = ON (auto)  
  JTAGCLOCK   = 25MHz (jtag25)  
  Verify       = verify off (!verify)  
Signals Mask:  
  NMI          = NO MASK (!nmi)  
  Soft/Hard INT= NO MASK (!int)  
  Soft RESET   = NO MASK (!reset)  
Hardware Bread Point Number:  
  Data Access BP      = 2  
  Instruction Address BP = 4  
  
>ememstat  
Board_num  EMEM_Size  ROM_Probe  
=====  
  ROM1      32Mbyte   Extend Type 2K  
  ROM2      32Mbyte   Extend Type 2K
```

[入力例]

```
env !nmi verify  
NMIをマスク、verifyをONの指定します。
```

helpコマンド

[書式]

```
help [ command ]
```

[パラメータ]

command: コマンド名を指定します。

コマンド名を省略した場合、コマンドの一覧が表示されます。

[機能]

各コマンドのヘルプメッセージを表示します。

[使用例]

```
help map
```

mapコマンドのヘルプを表示します。

ibp, ibp1, ibp2, ibp3, ibp4コマンド

[書式]

```
ibp{1|2|3|4} [ADDR [AMASK]] [asid ASID|noasid] [aeq|aneq]
ibp{1|2|3|4} /del
```

[パラメータ]

ibp{1|2|3|4}: 実行系ブレークのチャンネル(1-4)をibpに続けて指定します。
 ADDR: アドレスを16進数で指定します。
 asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。
 aeq|aneq: アドレスの比較条件を指定します。
 aeq: アドレスをイコールで比較します。
 aneq: アドレスをノットイコールで比較します。
 ibp{1|2|3|4} /del: 条件の解除を行います。
 /del: 解除を指定します。

[機能]

実行系のブレーク条件を設定します。ROM空間の指定も可能です。

[使用例]

```
ibp1 bfc01000 0 aeq
      bfc01000番地の実行でブレークします。
ibp1 /del
      abp1の条件を解除します。
```

inb, inh, inwコマンド

[書式]

```
inb [ADDR]  
inh [ADDR]  
inw [ADDR]
```

[パラメータ]

ADDR: 入力ポートのアドレスを16進数で指定します。

[機能]

inb, inh, inw, indは、アクセスサイズを区別して、リードを行ないます。
inbはバイト、inhはハーフ・ワード、inwはワード単位でアクセスします。

[使用例]

```
inb b0000000  
b0000000Hからバイト(8-bit)でリードします。  
inh 0000000  
b0000000Hからハーフワード(16-bit)でリードします。  
inw 0000000  
b0000000Hからワード(32-bit)でリードします。
```

init コマンド

[書式]

init

[パラメータ]

なし

[機能]

K I T - V R 5 5 0 0 - T P を初期化します。全ての環境設定値は初期化されます。

メモリキャッシュの除外エリアは初期化されません。

j read コマンド

[書式]

```
j read [ADDR [LENGTH]]
```

[パラメータ]

ADDR: アドレスを16進数で指定します。

LENGTH: 読み出すバイト数を16進数で指定します。 (max 100h)

[機能]

ROMコマンドで割り付けたROMエミュレーション領域をJTAG(CPU)から読み出すためのコマンドです。

通常のコマンドではROMエミュレーション領域へのアクセスは内部のメモリに対し直接行っています。

[使用例]

```
j read a0000000 100
```

a0000000hから100hバイトをJTAG経由で読み出します。

n c コマンド

[書式]

```
nc [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: メモリキャッシュの除外エリアの開始アドレスを指定します。

LENGTH: メモリキャッシュの除外エリアのバイト数を指定します。

デフォルト値 32 バイト、最少値 32 バイト

[機能]

メモリ参照の高速化を図るため、デバッガ内部に 8 ブロック * 32 バイトのメモリリードキャッシュを持っています。同一アドレスのメモリ参照などは実際にはメモリをリードしません。メモリに I/O を割り付けている場合は、このキャッシュ機能は実際の動作と矛盾してしまいますので、このコマンドでメモリキャッシュの除外エリアを指定してください。メモリキャッシュの除外エリアは最大 8 ブロック指定でき、最少のブロックサイズは 32 バイトです。

[使用例]

```
nc b8000000 100000
```

b8000000h から 100000 バイトの領域をメモリキャッシュの除外エリアに指定します。

```
>nc b8000000 100000
No Memory Cache Area
No. Address Length
1 b8000000 00100000
```

n c d コマンド

[書式]

ncd ブロック番号

[パラメータ]

ブロック番号: 削除するメモリキャッシュの除外エリアのブロック番号を指定します。

[機能]

メモリキャッシュの除外エリアを削除します。削除は各メモリキャッシュの除外エリアのブロック番号を指定します。

[使用例]

ncd 1

ブロック番号 1 をメモリキャッシュの除外エリアから削除します。

```
>nc bf000000 100
No Memory Cache Area
No. Address Length
1 bf000000 00000100
2 b8000000 00100000
```

```
>ncd 1
No Memory Cache Area
No. Address Length
1 b8000000 00100000
```

n s b pコマンド

[書式]

```
nsbp [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: ソフトウェアブレーク禁止領域の開始アドレスを指定します。

LENGTH: ソフトウェアブレーク禁止領域のバイト数を指定します。

指定領域の最小単位はワードバウンダリです。

また、指定できる領域の数は最大4ヶ所です。

[機能]

ソフトウェアブレークを禁止したい領域を指定します。

ブレークポイントを指定した場合、デバッガは暗黙的に対象アドレスに対し、メモリテスト（ライトアクセス）を行います。

一部のフラッシュROM等、ライトアクセスを行うことでメモリの状態が変り、正しいデータの読み出しが行えなくなる場合等に、ライトサイクルを禁止する目的で指定してください。

通常は、指定する必要はありません。

[使用例]

```
nsbp a0010000 20000
```

a0010000h番地から20000バイトの領域をソフトウェアブレーク禁止領域に指定します。

```
>nsbp a0010000 20000
```

```
Num Address Length
```

```
01 a0010000 00020000
```

n s b p d コマンド

[書式]

```
nsbpd [ ブロック番号 | /all ]
```

[パラメータ]

ブロック番号: 削除するソフトウェアブレーク禁止領域のブロック番号を指定します。

/all : 全てのソフトウェアブレーク禁止領域を削除します。

[機能]

nsbpdで指定したソフトウェアブレーク禁止領域を削除します。

[使用例]

```
nsbpd 1
```

ブロック番号 1 をソフトウェアブレーク禁止領域から削除します。

```
nsbp
```

Num	Address	Length
-----	---------	--------

01	a0100000	00200000
----	----------	----------

02	a0400000	00010000
----	----------	----------

```
>nsbpd 1
```

Num	Address	Length
-----	---------	--------

01	a0400000	00010000
----	----------	----------

nromコマンド

[書式]

```
nrom [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: 強制ユーザ領域の開始アドレスを指定します。

LENGTH: 強制ユーザ領域のバイト数を指定します。

指定領域の最小単位は、以下の通りです。

RTE-1000-TP : 4-byte単位。

RTE-2000-TP : エミュレーションしているROMのサイズに応じます。

- 8/16-bit : 128k-byte単位
- 32-bit : 256k-byte単位
- (64-bit : 512k-byte単位)

また、指定できる領域の数は最大4ヶ所です。

[機能]

ROMコマンドで指定したROMエミュレーション領域内の一部がユーザシステム上の資源にマップされていた場合にその領域を指定します。通常は指定する必要はありません。

指定領域に対する動作は以下の通りです。

- ・デバッガからのアクセスは強制的にユーザシステムに対し行われるようになります。
- ・実行中この領域へのアクセスサイクルでEMEMEN-信号はインアクティブ(Highレベル)になります。

(RTE-2000-TPのみ)

[使用例]

```
nrom a0000000 20000
a0000000h番地から20000バイトを強制ユーザ領域に指定します。
```

```
>nrom a0000000 20000
No. Address Length
1 a0000000 00020000
```

```
>nrom a080000 40000
No. Address Length
1 a0000000 00020000
2 a0800000 00040000
```

n r o m d コマンド

[書式]

nromd [ブロック番号 |/all]

[パラメータ]

ブロック番号: 削除する強制ユーザ領域のブロック番号を指定します。

/all : 全ての強制ユーザ領域のブロックを削除します。

[機能]

nromで指定した強制ユーザ領域を削除します。

[使用例]

ncd 1

ブロック番号 1 を強制ユーザ領域から削除します。

```
>nrom a080000 40000
No. Address Length
1 a0000000 00020000
2 a0800000 00040000
```

```
>nromd 1
No. Address Length
1 a0800000 00040000
```

outb, outh, outwコマンド

[書式]

```
outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
```

[パラメータ]

ADDR: 出力ポートのアドレスを16進数で指定します。

DATA: 出力するデータを16進数で指定します。

[機能]

outb, outh, outwは、アクセスサイズを区別して、ライトを行ないます。

outbはバイト、outhはハーフ・ワード、outwはワード単位でアクセスします。

[使用例]

```
outb b800000 12
      bfc00000hへバイトデータ : 12hをライトします。
outh b800000 1234
      bfc00000hへハーフワードデータ : 1234hをライトします。
outh b800000 12345678
      bfc00000hへワードデータ : 12345678hをライトします。
```

r e s e t コマンド

[書式]

reset

[パラメータ]

なし

[機能]

エミュレーションC P Uをリセットします。

romコマンド(RTE-1000-TP用コマンド)

[書式]

```
rom [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
[bus8|bus16|bus32] [|little|big]
```

[パラメータ]

ADDR [LENGTH]: エミュレーションする領域を指定します。

ADDR: 開始アドレスを指定します。エミュレートするROMの最下位のアドレス (ROMのパウンダリ)に合致していない場合、エラーになります。

LENGTH: エミュレートするROMのバイト数 (4バイトの境界単位で指定)

512k|1m|2m|4m|8m|16m|32m|64m|128m|256m: 1本のROMプローブでエミュレートするROMのBit容量を指定します。512K-bitから256M-bitまでの値が指定できます。
例えば、27C1024の場合は、1Mを指定します。

rom8|rom16: エミュレートするROMのデータビット数を指定します。

8bitと16bitが指定できます。DIP32-ROMケーブルを使用する場合はrom8、DIP-40/42-ROMケーブル、16bit-標準ROMケーブルを使用する場合は、rom16を指定します。

bus8|bus16|bus32: エミュレートするシステムの中でのROMのバスサイズを指定します。

8bit,16bit,32bitが指定できます。

|little|big: romデータのエンディアンを指定します。ダウンロード時、little指定時は、ファイルのバイナリイメージをそのままの形で書き込みます。big指定時は、ROMのバスサイズに応じて、上位バイトと下位バイトのデータを入れ替えて書き込みます。

[機能]

RTE-1000-TPのROMのエミュレーション環境の設定を行います。設定はADDRとLENGTHをペアで入力する以外は変更が必要なパラメータだけ入力できます。入力の順序は任意です。但し、同じパラメータを2回入力した場合は、後から入力した値が有効です。初期値は、LENGTH = 0 (使用しない)になっています。

[入力例]

```
rom bfc0000 40000 1m rom16 bus32 little
```

27C1024(1M-bitの16bit-ROM)をbfc0000hから256Kバイト(40000h)エミュレートします。
この場合、バスが32Bitですので、16bit-rom2個をエミュレートします。Romのエンディアンはlittleです。(バイナリのイメージをそのままロードします。)

```
rom bfc00000 40000 2m rom rom16 bus16 big
```

27C2048(2M-bitの16bit-ROM)をbfc0000hから256Kバイト(40000h)エミュレートします。
この場合、16bit-rom1個をエミュレートします。Romのエンディアンはbigです。
(バイナリのイメージを上位と下位のバイトを入れ替えてロードします。)

<備考>

romコマンドで指定した領域における注意事項

romコマンドで指定した範囲へのデバッガからのアクセスは、ツール内部のエミュレーションメモリに対し直接アクセスしています。その結果、プロセッサから正しくROMにアクセスできない状態においても表示は正しく行われますので、デバッグ初期の段階ではjreadコマンド (CPUのバス経由で読み出すコマンド)を使用して読み出し確認するか、envコマンドでverifyをONにして書き込み(ダウンロード)を行うことをお勧めします。

rom1..rom4 コマンド(RTE-2000-TP 用コマンド)

[書式]

```
rom1 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16|bus32|bus64] [[!]wren]
rom2 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16] [[!]wren]
rom3 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16|bus32] [[!]wren]
rom4 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16] [[!]wren]
```

rom1: スロット#3に実装されたEMEM基板を含むモジュールに対する設定コマンドです。

rom2: スロット#4に実装されたEMEM基板を含むモジュールに対する設定コマンドです。

rom3: スロット#5に実装されたEMEM基板を含むモジュールに対する設定コマンドです。

rom4: スロット#6に実装されたEMEM基板を含むモジュールに対する設定コマンドです。

[パラメータ]

ADDR [LENGTH]: エミュレーションする領域を指定します。

ADDR: 開始アドレスを指定します。エミュレートするROMの最下位のアドレス (ROMのパウンダリ)に合致していない場合、エラーになります。

LENGTH: エミュレートするROMのバイト数を指定します。(4バイトの境界で指定)

512k|1m|2m|4m|8m|16m|32m|64m|128m|256m: 1本のROMプロープでエミュレートするROMのBit容量を指定します。512K-bitから256M-bitまでの値が指定できます。例えば、27C1024の場合は、1Mを指定します。

rom8|rom16: エミュレートするROMのデータビット数を指定します。

8bitと16bitが指定できます。DIP32-ROMケーブルを使用する場合はrom8、DIP-40/42-ROMケーブル、16bit-標準ROMケーブルを使用する場合は、rom16を指定します。

bus8|bus16|bus32|bus64: エミュレートするシステムの中でのROMのバスサイズを指定します。

8bit, 16bit, 32bit, 64bitが指定できます。

>> [64-bit]は将来のためのパラメータです。(本KITでは使用しません)

[[!]wren]: エミュレーションメモリをRAMとして使用する場合の設定です。

wrenで書き込み許可、!wrenで書き込み禁止です。初期値は!wrenです。

[機能]

RTE-2000-TPのROMエミュレーション環境の設定を行います。設定はADDRとLENGTHをペアで入力する以外は変更が必要なパラメータだけ入力できます。入力の順序は任意です。但し、同じパラメータを2回入力した場合は、後から入力した値が有効です。初期値は、LENGTH = 0 (使用しない)になっています。

[入力例]

rom1 bfc00000 40000 2m rom16 bus16 !wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#3	bfc00000 – bfc3ffff	16-bit	16-bit	2M-bit	禁止

rom2 bfc40000 40000 2m rom16 bus16 wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#4	bfc40000 – bfc7ffff	16-bit	16-bit	2M-bit	許可

rom1 bfc00000 80000 2m rom rom16 bus32 !wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#3+#4	bfc00000 – bfc7ffff	32-bit	16-bit	2M-bit	禁止

この時、rom2コマンドは発行しないでください。

< 備考 >

romコマンドで指定した領域における注意事項

rom1..rom4コマンドで指定した範囲へのデバッグからのアクセスは、ツール内部のエミュレーションメモリに対し直接アクセスしています。その結果、プロセッサから正しくROMにアクセスできない状態においても表示は正しく行われますので、デバッグ初期の段階ではjreadコマンド（CPUのバス経由で読み出すコマンド）を使用して読み出し確認するか、envコマンドでverifyをONにして書き込み（ダウンロード）を行うことをお勧めします。

romコマンドとEMEM基板の関係

romコマンド	バス幅	対象EMEM基板の スロット位置	使用できないromコマンド
rom1	8-bit	#3	
	16-bit	#3	
	32-bit	#3+#4	rom2
	64-bit	#3+#4+#5+#6	rom2, rom3, rom4
rom2	8-bit	#4	
	16-bit	#4	
rom3	8-bit	#5	
	16-bit	#5	
	32-bit	#5+#6	rom4
rom4	8-bit	#6	
	16-bit	#6	

t l b コマンド

[書式]

```
tlb [all|INDEX [MASK HI LO0 LO1]]
```

[パラメータ]

all: 全てのインデックスの表示を指定します。

INDEX: 特定のインデックスを指定します。

MASK HI LO0 LO1: 変更時、INDEXで指定したインデックスの内容を指定します。
4つセットで入力してください。

MASK: PageMaskを指定します。

HI: EntryHiを指定します。

LO0: EntryLo0を指定します。

LO1: EntryLo1を指定します。

[機能]

TLBの内容の表示と変更を行います。

[使用例]

```
tlb all
```

全インデックスを内容を表示します。

```
tlb 10
```

TLB#=10の内容を表示します。

symfile, symコマンド

[書式]

symfile FILENAME

sym [NAME]

[パラメータ]

symfile: ファイル名を指定します。

sym: シンボルの先頭文字列を指定します。

[機能]

symfile コマンドは、FILENAMEで指定したelfファイルからシンボルを読み込みます。

対象となるのはグローバルシンボルだけです。

Symコマンドは、読み込んだシンボルの表示（最大30個）をできます。

[使用例]

symfile c:¥test¥dry¥dry.elf

c:¥test¥dryのディレクトリからelfファイル:dry.elfのシンボルを読み込みます。

sym m

mから始まるシンボルを最大30個表示します。

v e r コマンド

[書式]

ver

[パラメータ]

なし

[機能]

KIT-MIPS32/4Kc-TPの関連ソフトのバージョンを表示します。