

## 付録. A KIT-VR7701-TP (H) 内部コマンド

本書は、KIT-VR7701-TP (H) 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)

### PARTNER/Winの場合

>&                      << スルーコマンドへの移行します。  
 >#ENV                  << 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。

### GHS-Multiの場合

RTESERV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

コマンド一覧.....	A-1
コマンド書式.....	A-2
オプションブ레이크 : BPOPTコマンド.....	A-3
キャッシュ操作 : CACHEINIT, CASHEFLUSHコマンド.....	A-4
逆アセンブル表示 : DASMコマンド.....	A-5
環境設定 : ENV, EMEMSTATコマンド.....	A-6
実行系イベント : EVAコマンド.....	A-8
アクセス系イベント : EVEコマンド.....	A-9
ヘルプ : HELPコマンド.....	A-10
ポート入力 : INB, INH, INW, INDコマンド.....	A-11
初期化 : INITコマンド.....	A-12
JTAGリード : JREADコマンド.....	A-13
デバッガキャッシュの解除 : NCコマンド.....	A-14
デバッガキャッシュの設定 : NCDコマンド.....	A-15
ソフトブ레이크禁止領域の設定 : NSBPコマンド.....	A-16
ソフトブ레이크禁止領域の解除 : NSBPDコマンド.....	A-17
強制ユーザ領域の設定 : NROMコマンド.....	A-18
強制ユーザ領域の解除 : NROMDコマンド.....	A-19
ポート出力 : OUTB, OUTH, OUTW, OUTDコマンド.....	A-20
CPUリセット : RESETコマンド.....	A-21
E. ROMの環境設定 : ROMコマンド(RTE-1000-TP用コマンド).....	A-22
E. ROMの環境設定 : ROM1..ROM4コマンド(RTE-2000(H)-TP用コマンド)....	A-23
TLB : TLB32, TLB64コマンド.....	A-25
SFRアクセス : SFRコマンド.....	A-26
シンボル : SYMFILE, SYMコマンド.....	A-31
トレースの設定、開始 : TRONコマンド.....	A-32
トレースの強制終了 : TROFFコマンド.....	A-33
トレース表示 : TRACEコマンド.....	A-34
トレースの設定確認 : TMODEコマンド.....	A-36
トレースデータのディレイ調整 : TDATA_DLYコマンド.....	A-37
バージョン表示 : VERコマンド.....	A-38

ご注意:

- 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いずれかの動作が異常になる場合があります。
- 内部コマンドは一部のコマンドを除き 64-Bit のアドレスには対応していません。64-Bit モードでご使用になる場合はそれぞれのデバッガの注意事項を良く読みご使用ください。

## 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ります。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ります。(16進数には演算子は使用できません。)

## bpoptコマンド

### [書式]

```
bpopt [[!]eve] [[!]eva]
```

### [パラメータ]

eve: イベント:eve をブレイク条件に指定します。!は、条件解除を意味します。

eva: イベント:eva をブレイク条件に指定します。!は、条件解除を意味します。

### [機能]

イベント条件をブレイク条件に設定または解除します。

eveは実行系のイベント、evaはアクセス系のイベントです。

Eve, evaの設定方法はそれぞれのコマンドを参照ください。

### [入力例]

```
bpopt eve
```

eveをブレイク条件に設定します。

```
bpopt !eve
```

eveをブレイク条件から解除します。

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。
```

## dasmコマンド

### [書式]

dasm [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]

アセンブラ表示するためのコマンドです。

### [使用例]

dasm bfc0000 100

bfc0000hから100hバイトをアセンブラ表示します。

env, ememstatコマンド

## [書式]

```
env [!]auto [!]nmi [!]int [jtag[xxx]. [yyy]] {M|K} [!]verify]
    [tclkdiv{1|2|4|6|8|16}]
```

```
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。

[!]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。

[jtag[xxx]. [yyy]] {M|K}: JTAGクロックの周波数をMHz, またはKHzの単位で指定します。指定は10KHzから125MHzの間の任意の値が可能ですが、設定されるのは指定値以下の以下の値に丸められます。実際の設定値は表示で確認できます。

RTE-2000-TP : [25MHz, 12.5MHz, 5MHz, 2MHz, 1MHz, 500KHz, 250KHz, 100KHz]

RTE-2000H-TP : [125MHz, 100MHz, 80MHz, 60MHz, 50MHz, 40MHz, 30MHz, 25MHz, 12.5MHz, 5MHz, 2MHz, 1MHz, 500KHz, 250KHz, 100KHz, 50KHz, 25KHz, 10KHz]

注意: 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、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。初期値は25MHzを上限とした動作する最高周波数に自動的に設定します。初期値以上の値に設定する場合はCPUの許容範囲内で設定してください。CPUのスペック以上の周波数を設定した場合の動作は保証できません。

[!]verify: メモリへの書き込み時にリードアウトしてペリファイするかどうか指定します。!はペ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(jread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。

tclkdiv{1|2|4|6|8|16}: CPU動作クロック(Pclock)からのトレースクロックの分周率を指定します。それぞれ、[1/1|1/2|1/4|1/8|1/16]に対応します。但し、1/1だけは“SysClock”に対しての比率です。(=バスクロック)

## [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(H)-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。

RTE-1000-TPの場合

```
Probe:
Unit      : RTE-1000-TP          << 接続している本体を表示します。
Rom Probe : Extend Type        << 接続しているROMプローブのタイプを表示します。
Emem Size : 32Mbyte            << 実装しているエミュレーションメモリの容量を表示します。
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK = 25MHz (jtag25M)
Verify     = verify off (!verify)
Signals Mask:
NMI        = NO MASK (!nmi)
INT        = NO MASK (!int)
Trace Clock Settings:
TRCCLK Div. = 1/4 (tclkdiv4)
```

RTE-2000(H)-TPの場合

```
Probe:
Unit      : RTE-2000(H)-TP      << 接続している本体を表示します
Rom Probe : (use ememstat command)
Emem Size : (use ememstat command)
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK  = 25MHz (jtag25M)
Verify     = verify off (!verify)
Signals Mask:
NMI        = NO MASK (!nmi)
INT        = NO MASK (!int)
Trace Clock Settings:
TRCCLK Div. = 1/4 (tclkdiv4)

>ememstat
Board_num  EMEM_Size  ROM_Probe
=====
ROM1       8Mbyte    Extend Type 2K
```

## [入力例]

```
env !nmi verify
    NMIをマスク、verifyをONの指定をします。
env jtag40m
    JTAGクロックを40MHzに設定します。
```

evaコマンド

## [書式]

```
eva [[!] ADDR [AMASK [[!] DATA [DMASK]|nodata] [byte|hword|word|dword]]]
    [read|write|acc] [{noasid} | {asid ASID}]
eva [noaddr [[!] DATA [DMASK]|nodata] [byte|hword|word|dword]]]
    [read|write|acc] [{noasid} | {asid ASID}]
```

## [パラメータ]

ADDR: アドレスを16進数で指定します。!をつけるとaddrに対するnotを意味します。

AMASK: ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。

noaddr: アドレスの指定を条件から削除します。

[[!] DATA [DMASK]|nodata: データ条件を指定します。

DATA: データを16進数で指定します。!をつけるとDATAに対するnotを意味します。

DMASK: DATAに対するマスクを指定します。ビット単位で'1'でマスクします。

nodata: データの指定を条件から削除します。

byte|hword|word|dword: アクセスサイズ条件を指定します。

byte: アクセスサイズとしてバイト条件を指定します。

hword: アクセスサイズとしてハーフワード条件を指定します。

word: アクセスサイズとしてワード条件を指定します。

dword: アクセスサイズとしてダブルワード条件を指定します。

read|write|acc: ステータス条件を指定します。

read: ステータス条件としてリードサイクルを指定します。

write: ステータス条件としてライトサイクルを指定します。

acc: ステータスの指定を条件から削除します。

noasid|asid ASID:

noasid: ASIDを比較対象にしません。

asid ASID: ASIDを比較対象に含めます。

## [機能]

アクセスサイクルのイベントを指定します。

## [使用例]

```
eva 1000 0 5555 0 hword read
```

1000h番地からハーフワードで5555hをリードしたサイクルをeva条件に指定します。

## [備考]

evaで指定したイベント条件は、ブレークやトレースのトリガ条件として使用できます。

設定したイベント条件は、bpopt, tronを使用してブレークやトレースの条件として使用します。



## e v eコマンド

### [書式]

```
eve [[!] ADDR [AMASK] [{noasid} | {asid ASID}]]
```

### [パラメータ]

ADDR: アドレスを16進数で指定します。!をつけるとaddrに対するnotを意味します。

AMASK: ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。

noasid|asid ASID:

noasid: ASIDを比較対象にしません。

asid ASID: ASIDを比較対象に含めます。

### [機能]

実行アドレスのイベントを指定します。

### [使用例]

```
eve 1000 0
```

1000hの命令実行をマスクなしでイベントとして指定します。

```
eve 1000 0ff
```

1000hの下位8bitをマスクした実行アドレスをイベントとして指定します。

```
eve 1000 asid 10
```

asid=10hで1000hの命令実行をマスクなしでイベントとして指定します。

### [備考]

eveで指定したイベント条件は、ブレークやトレースのトリガ条件として使用できます。

設定したイベント条件は、bpopt, tronを使用してブレークやトレースの条件として使用します。

## h e l p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。

inb, inh, inw, indコマンド

## [書式]

inb [ADDR]

inh [ADDR]

inw [ADDR]

ind [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、リードを行います。

inbはバイト、inhはハーフ・ワード、inwはワード、indはロング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。

ind 0000000

b0000000Hからロングワード(64-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]

- ICEの環境を起動時の状態に初期化します。  
以下を除き、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

j r e a dコマンド

## [書式]

```
jread [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]

```
jread a0000000 100
```

a0000000hから100hバイトをJTAG経由で読み出します。

nc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、ファームウェア内に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には実際にはメモリをリードしません。I/Oを割り付けている空間では、このキャッシュ機能は実際の動作と矛盾しますので、このコマンドで除外エリアとして指定してください。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: 削除するメモリアドレスの除外エリアのブロック番号を指定します。

## [機能]

メモリアドレスの除外エリアを削除します。削除は各メモリアドレス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 をメモリアドレス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
```

nsbpコマンド

## [書式]

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

## [パラメータ]

ADDR: ソフトウェアブレイク禁止領域の開始アドレスを指定します。

LENGTH: ソフトウェアブレイ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]

ソフトウェアブレイクを禁止したい領域を指定します。

ブレイ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わり、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イク禁止領域に指定します。

```
>nsbp a0010000 20000
```

```
Num Address Length
```

```
01 a0010000 00020000
```



nsbpd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イ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イク禁止領域を削除します。

## [機能]

nsbpで指定したソフトウェアブレイ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イ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
```

n r o m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 (H)-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 (H)-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 a0800000 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 a0800000 40000
```

No.	Address	Length
1	a0000000	00020000
2	a0800000	00040000

```
>nromd 1
```

No.	Address	Length
1	a0800000	00040000

outb, outh, outw, outdコマンド

## [書式]

```
outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
outd [[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、ライトを行います。  
outbはバイト、outhはハーフ・ワード、outwはワード、outdはロング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1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。
outd b800000 123456789abcdef0
    bfc00000hへワードデータ : 123456789abcdef0hをライトします。
```

resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

対象エミュレーションCPU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のバウンダリ)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に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) でbfc00000h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bfc00000h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。)

## &lt;備考&gt;

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(H)-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のバウンダリ) に合致していない場合、指定アドレス以下のアドレス領域は非エミュレーション領域になります。

LENGTH: エミュレートするROMのバイト数を指定します。

備考: ADDR, LENGTHで指定できる領域の最小単位は、エミュレーションしているROMのサイズに応じ、以下の通りです。

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

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(32M-Byte)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のアダプタを使用する場合はrom8、DIP-40/42のアダプタ、及び16bit-標準ROMケーブルをそのまま使用する場合は、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]: Write Enable: 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で書込み許可、!wrenで書込み禁止です。初期値は!wrenです。

## [機能]

RTE-2000(H)-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に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 400000 32m rom16 bus16 !wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイェ-ブル
			バス幅	Bit数	
#3	bfc00000 - bfffffff	16-bit	16-bit	32M-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		ライトイェ-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		ライトイェ-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 3 2, t l b 6 4 コマンド

## [書式]

tlb32 [all|INDEX [MASK HI L00 L01]]

tlb64 [all|INDEX [MASK HI L00 L01]]

## [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 L00 L01: 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。

L00: EntryLo0を指定します。

L01: EntryLo1を指定します。

## [機能]

TLBの内容の表示と変更を行います。

tlb32は、CPUが32bitの時の内容です。

tlb64は、CPUが64bitの時の内容です。

## [使用例]

tlb32 all

全インデックスの内容を表示します。

tlb32 10

TLB#=10の内容を表示します。

s f r コマンド

## [書式]

sfr [reg [VAL]]

## [パラメータ]

VAL: SFRのレジスタ値を16進数で指定します。

reg: SFRレジスタ名を指定します。

レジスタとして使用できる名称は以下の通りです。

## SFR (R/W):

SCALLOCCNT SCECC SCLRU SCWRCNT BIUCFG1 BIUCFG2 BLUEVTYPE1 BLUEVTYPE2  
 BLUEVCOUNT1OF BIUENCOUNT2OF BLUEVCOUNT1 BLUEVCOUNT2  
 BIUREVID BIUCPUCLKCNT TIMDATALO  
 TIMDATAHI TIMMODE1 TIMMODE2 ACOCRO  
 ACOCR1 AC1CRO AC1CR1 ECOFR ECOCR  
 ECO EC1FR EC1CR EC1 DCR SYSRCR  
 PCIXRCR DMARCR ETH1RCR ETH2RCR  
 SDTYPE SDTIMING SDMODE SDCNT SDREFCYC SDERRSYNHI SDERRSYNLO SDINITCNT  
 SDERRINSHI SDERRINSLO BCFGR1  
 BCFGR2 BCFGR3 BCFGR4 BCFGR5 BCFGR6 GPIO\_SEL  
 GPIO\_IOSEL1 GPIO\_IOSEL2 GPIO\_IOSEL3 GPIO\_IOSEL4 GPIO\_DATA1  
 GPIO\_DATA2 GPIO\_DATA3 GPIO\_DATA4 MSAO  
 MSA1 MSA2 MSA3 MSB0 MSB1 MSB2 MSB3 MBC0  
 MBC1 MBC2 MBC3 MBBC0 MBBC1 MBBC2  
 MBBC3 MCCRO MCCR1 MCCR2 MCCR3 MICCR MOCRO  
 MOCR1 MOCR2 MOCR3 DEVID VENID  
 STATUS COMMAND CLCODE REVID LTTIMER CALISIZE PTBADDR0L PTBADDR1L  
 PTBADDR1M SUBSYSID SUBSYSVENID CAPPOINT MAX\_LAT  
 MIN\_GNT INTPIN INTLINE CFGADDR CFGDATA CFGATTR PINTACNT PINTXMASK SERRSTAT  
 PERRSTAT PERRMASK OCBERRSTAT OCBERRMASK  
 MSGCNT MSINXTCAP MSICAPID MSGADDR MSGUPADDR MSGDATA PCIXCOMMAND PCIXNXTCAP  
 PCIXCAPID PCIXSTAT PTMEMSEGO PTMEMMSKO PTMEMENSWO PTMEMSEG1 PTMEMMSK1  
 PTMEMENSW1 PMSEGL PMSEGM PRST  
 UNITCNT CNTOTIMER PSLADAFLSH MACG1\_1 MACG2\_1 IPGT\_1 IPGR\_1 CLRT\_1 LMAX\_1  
 LSA2\_1 LSA1\_1 VLTP\_1 MIIC\_1 MADR\_1  
 MWTD\_1 AFR\_1 HT1\_1 HT2\_1 CAR1\_1  
 CAR2\_1 CAM1\_1 CAM2\_1 RBYTC\_1  
 RPKTC\_1 RFCSC\_1 RMCAC\_1 RBCAC\_1 RXCF\_1 RXPFC\_1 RXUOC\_1 RALNC\_1  
 RFLRC\_1 RCDEC\_1 RFCRC\_1 RUNDC\_1 ROVRC\_1 RFRGC\_1 RJBR\_1 R64C\_1  
 R127C\_1 R255C\_1 R511C\_1 R1KC\_1 RMAXC\_1 RVBTC\_1 TBYTC\_1  
 TPCTC\_1 TFCSC\_1 TMCAC\_1 TBCAC\_1 TUCAC\_1 TXPFC\_1 TDFRC\_1 TXDFC\_1  
 TSCLC\_1 TMCLC\_1 TLCLC\_1 TXCLC\_1 TNCLC\_1 TGSEC\_1 TIMEC\_1 XMT\_CFGR\_1  
 XMT\_CTLR\_1 XMT\_DPR\_1 RCV\_CFGR\_1 RCV\_CTLR\_1 RCV\_DPR\_1  
 RCV\_PDR\_1 CCR\_1 MSR\_1 MACG1\_2  
 MACG2\_2 IPGT\_2 IPGR\_2 CLRT\_2 LMAX\_2 LSA2\_2  
 LSA1\_2 VLTP\_2 MIIC\_2 MADR\_2 MWTD\_2  
 AFR\_2 HT1\_2 HT2\_2 CAR1\_2 CAR2\_2  
 CAM1\_2 CAM2\_2 RBYTC\_2 RPKTC\_2  
 RFCSC\_2 RMCAC\_2 RBCAC\_2 RXCF\_2 RXPFC\_2 RXUOC\_2 RALNC\_2 RFLRC\_2  
 RCDEC\_2 RFCRC\_2 RUNDC\_2 ROVRC\_2 RFRGC\_2 RJBR\_2 R64C\_2 R127C\_2  
 R255C\_2 R511C\_2 R1KC\_2 RMAXC\_2 RVBTC\_2 TBYTC\_2  
 TPCTC\_2 TFCSC\_2 TMCAC\_2 TBCAC\_2 TUCAC\_2 TXPFC\_2 TDFRC\_2 TXDFC\_2  
 TSCLC\_2 TMCLC\_2 TLCLC\_2 TXCLC\_2 TNCLC\_2 TGSEC\_2 TIMEC\_2 XMT\_CFGR\_2

```

XMT_CTLR_2 XMT_DPR_2 RCV_CFGR_2 RCV_CTLR_2 RCV_DPR_2
RCV_PDR_2 CCR_2 MSR_2 PRE_A PRE_B PRE_C PRE_D PRE_E PRE_F PRE_G CLK_SEL_A
CLK_SEL_B CLK_SEL_C TC_0_STS TC_0_EN TC_0_DATA0 TC_1_STS
TC_1_EN TC_1_DATA0 TC_1_DATA1 TC_2_STS TC_2_EN TC_2_DATA0 TC_3_STS
TC_3_EN TC_3_DATA0 TC_4_STS TC_4_EN TC_4_DATA0 TC_4_DATA1 TC_5_STS
TC_5_EN TC_5_DATA0 TC_6_STS TC_6_EN TC_6_DATA0 TC_7_STS
TC_7_EN TC_7_DATA0 TC_8_STS TC_8_EN TC_8_DATA0 TC_8_DATA1 TC_8_DATA2
TC_8_DATA3 CH1_LEVEL CH2_LEVEL
CH3_LEVEL CH4_LEVEL CH5_LEVEL CH6_LEVEL CH7_LEVEL CH8_LEVEL CH9_LEVEL
CH10_LEVEL CH11_LEVEL CH12_LEVEL CH13_LEVEL CH14_LEVEL CH15_LEVEL CH16_LEVEL
INT_CLR INT_SET LEVEL_EDGE INT_MASK LEVEL_MASK INT_ACT_LV NMI_CLR
NMI_SET NMI_MASK OUT_SEL UART_1_DLL
UART_1_IER UART_1_DLM UART_1_LCR UART_1_MCR UART_1_LSR UART_1_MSR UART_1_SCR
UART_1_RESET UART_2_DLL UART_2_IER
UART_2_DLM UART_2_LCR UART_2_MCR UART_2_LSR UART_2_MSR UART_2_SCR UART_2_RESET
CSI_MODEREG CSI_CLKSELREG CSI_SOTBREG
CSI_SOTBFREG CSI_CNTREG CSI_INTREG CSI_IFIFOVREG CSI_OFIFOVREG CSI_OFIFOREG
CSI_FIFOTRGREG
SFR (W) :
SCIDXSTTAG SCIDLDTAG SCIDXBINV SCHITWBINV SCTAG MCMD_1 MCMD_2 INT_ACK
UART_1_THR UART_1_FCR UART_2_THR
UART_2_FCR
SFR (R) :
BIUCBERRINFO1 BIUCBERRINFO2 BIUCPIXRSTSTAT TIMCOUNTLO TIMCOUNTHI
SDREFCOUNT SDERRADDR MICR BIST
HDTYPE PINTXSTAT PERRADDRL PERRADDRM
SPCMPERRMSG PTVR_1 MRDD_1 MIND_1 ISR_1 PTVR_2 MRDD_2 MIND_2 ISR_2 INT_PENDING
INT_VECTOR INT_STATUS NMI_PENDING
NMI_STATUS UART_1_RBR UART_1_IIR
UART_2_RBR UART_2_IIR CSI_SIRBREG
CSI_SIRBEREG CSI_SIOREG CSI_IFIFOREG

```

## [機能]

SFRレジスタ値の設定と表示を行います。

## [使用例]

```

sfr GPIO_DATA1
    GPIO_DATA1レジスタの値を表示します。
sfr GPIO_DATA1 2
    GPIO_DATA1 レジスタに2hを設定します。

```

## [備考]

NEC社発行のVR7701のユーザーズマニュアルに記載のレジスタ名とは一部異なる名称のものがあります。それらは以下の通りです。

1. レジスタ名称中の“-“, “#”, “スペース”は”\_”に置き換えています。
2. 次ページの表中のレジスタは略号（NEC社推奨）を採用しています。

略号対応表

略号	フルネーム	アドレス (little endian)
scaloccnt	SC allocate control	0xf f800 0048
scidxsttag	Sc index store tag	0xf f800 0050
scidxldtag	sc index load tag	0xf f800 0058
scidxwbinv	sc index writeback invalidate	0xf f800 0060
schitwbinv	sc hit write back invalidate	0xf f800 0068
sctag	sc tag	0xf f800 0100
scecc	sc ecc	0xf f800 0108
sclru	sc lru	0xf f800 0110
scwrcnt	sc write control	0xf f800 0118
biucfg1	biu config 1	0xf f800 0000
biucfg2	biu config 2	0xf f800 0008
biuevtype1	system biu event type 1	0xf f800 0010
biuevtype2	system biu event type 2	0xf f800 0018
biuevcount1of	system biu event count 1 overflow	0xf f800 0020
biuevcount2of	system biu event count 2 overflow	0xf f800 0028
biuevcount1	system biu event count 1	0xf f800 0030
biuevcount2	system biu event count 2	0xf f800 0038
biurevid	system biu rev ID	0xf f800 0040
biuocberrinfo1	biu-onchipbus error information 1	0xf f800 0080
biuocberrinfo2	biu-onchipbus error information 2	0xf f800 0088
biupcixrststat	biu pci-x reset status	0xf f800 0090
biucpuclkcnt	biu cpu clock control	0xf f800 0098
timdatalo	timer data low	0xf f800 00a0
timdatahi	timer data high	0xf f800 00a8
timcountlo	timer count low	0xf f800 00b0
timcounthi	timer count high	0xf f800 00b8
timmode1	timer mode 1	0xf f800 00c0
timmode2	timer mode 2	0xf f800 00c8
sdtype	sdram type	0xf f800 0200
sdtiming	sdram timing	0xf f800 0208
sdmode	sdram mode	0xf f800 0210
sdcnt	sdram control	0xf f800 0218
sdrefcyc	sdram refresh cycle	0xf f800 0220
sdrefcount	sdram refresh count	0xf f800 0228
sderraddr	sdram error address	0xf f800 0230
sderrsynhi	sdram error syndrome high	0xf f800 0238
sderrsynlo	sdram error syndrome low	0xf f800 0240
sdinitcnt	sdram initialization control	0xf f800 0248
sderrinshi	sdram error insert high	0xf f800 0250
sderrinslo	sdram error insert low	0xf f800 0258
devid	device id	0xf f884 0000
venid	vender id	0xf f884 0000
status	status	0xf f884 0004

command	command	0xf f884 0004
clcode	class code	0xf f884 0008
revid	revision id	0xf f884 0008
bist	bist	0xf f884 000c
hdtype	header type	0xf f884 000c
lttimer	letency timer	0xf f884 000c
calisize	cache line size	0xf f884 000c
ptbaddr0l	pci target base address #0 lsw	0xf f884 0010
ptbaddr0m	pci target base address #0 Msw	0xf f884 0014
ptbaddr1l	pci target base address #1 lsw	0xf f884 0018
ptbaddr1m	pci target base address #1 Msw	0xf f884 001c
subsysid	subsystem id	0xf f884 002c
subsysvenid	subsystem vender id	0xf f884 002c
cappoint	capabilities pointer	0xf f884 0034
max_lat	max_lat	0xf f884 003c
min?gnt	min_gnt	0xf f884 003c
intpin	interrupt pin	0xf f884 003c
intline	interrupt line	0xf f884 003c
cfgaddr	config address	0xf f884 0040
cfgdata	config data	0xf f884 0044
cfgattr	config attribute	0xf f884 0048
pintacnt	pci inta control	0xf f884 0050
pintxstat	pci intx status	0xf f884 0054
pintxmask	pci intx mask	0xf f884 0058
serrstat	serr status	0xf f884 005c
perraddrl	pci error address lsw	0xf f884 0060
perraddrm	pci error address msw	0xf f884 0064
perrstat	pci error sttus	0xf f884 0068
permask	pci error msak	0xf f884 006c
ocberrstat	onchipbus error status	0xf f884 0078
ocberrmask	onchipbus error mask	0xf f884 007c
msgcnt	message control	0xf f884 0080
msinxtpcap	msi next capability	0xf f884 0080
msicapid	msi capability id	0xf f884 0080
msgaddr	message address	0xf f884 0084
msgupaddr	message upper address	0xf f884 0088
msgdata	message data	0xf f884 008c
pcixcommand	pci-x command	0xf f884 0090
pcixnxtcap	pci-x next capability	0xf f884 0090
pcixcapid	pci-x capability id	0xf f884 0090
pcixstat	pci-x status	0xf f884 0094
spcmpermsg	split comp error message	0xf f884 0098
ptmemseg0	pci target memory segment #0	0xf f884 00a0
ptmemmsk0	pci target memory mask #0	0xf f884 00a4
ptmemensw0	pci target memory endian swap #0	0xf f884 00a8

ptmemseg1	pci target memory segment #1	0xf f884 00b0
ptmemmsk1	pci target memory mask #1	0xf f884 00b4
ptmemensw1	pci target memory endian swap #1	0xf f884 00b8
pmsegl	pci master segment lsw	0xf f884 00c0
pmsegm	pci master segment msw	0xf f884 00c4
prst	pci reset	0xf f884 00f0
unitcnt	unit control	0xf f884 00f4
cntotimer	count out timer	0xf f884 00f8
psladafsh	pci slave data flash	0xf f884 00fc

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）をします。

## [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。
```

tronコマンド

## [書式]

```
tron [DELAY] [[!]delay] [[!]eve] [[!]eva] [noext|nega|posi]
```

## [パラメータ]

DELAY = 0..xxxx: デレイカウンタ <注意> (RTE-1000-TPでは、0..1ffffです)  
トリガ成立後にメモリに取り込むフレーム数を十進数で指定します。

[[!]delay: 強制デレイモードを指定します。!で通常モードの指定に戻ります。  
強制デレイモードでは、TRONコマンドの直後よりトレースを開始し、デレイカウンタ数分のトレースを完了した時点で強制的にトレースを終了するモードです。  
このモード中は、トリガイベントは無視されます。

[[!]eve イベントeveをトレーストリガとして指定します。!で指定を削除します。

[[!]eva イベントevaをトレーストリガとして指定します。!で指定を削除します。

noext|nega|posi: トリガとして外部入力端子 (EXI0) を指定します。

noext: EXI0をトリガとして使用しません。

posi: EXI0の立ち上がりエッジをトリガとして指定します。

nega: EXI0の立ち下がエッジをトリガとして指定します。

## [機能]

トレースの諸設定とトレースバッファをクリアし、トレースの取り込みを開始します。

## [使用例]

```
tron delay 3ffff
```

delayモードで無条件に3ffffサイクル分トレースします。

この場合、tronコマンドの直後よりトレースを開始し、3ffffdhサイクル分トレースして、終了します。

```
tron !delay eve 1ffff
```

delayモードを解除し、eveをトリガポイントにしてトレースを開始します。

トリガ成立後の取り込みサイクルとして、1ffffhを指定します。

この場合、tronコマンド直後よりトレースを開始し、トリガ点を通過後、1ffffhサイクル分トレースして終了します。結果的にトリガを基点にして、前後、約20000hサイクル分の実行履歴がトレースできます。

## [備考]

eve, evaの設定方法はそれぞれのコマンドを参照ください。



t r o f fコマンド

[書式]

troff

[パラメータ]

なし

[機能]

トレースの取り込みを強制的に終了します。

traceコマンド

## [書式]

```
trace [POS] [all|pc|data] [asm] [asm|ttag1|ttag2] [subNN]
```

## [パラメータ]

POS=±0..xxxx: トリガサイクル近辺または終了サイクルを0として、トレースの表示開始位置を16進数で指定します。<注意>(RTE-1000-TPでは、0..1fffdです)

all|pc|data: 取り込んだトレース情報の中から選択して表示するサイクルを指定します。

all: 全てのサイクル

pc: 実行サイクルのみ

data: データサイクルのみ

asm|ttag1|ttag2 表示種別を指定します。

Asm: アセンブラ表示のみ

ttag1: アセンブラ表示+絶対時間でのタイムタグ表示

ttag2: アセンブラ表示+相対時間でのタイムタグ表示

subNN: 実際に取り込まれる一つの分岐情報から連続して逆アセンブルする命令数を16進数で指定します。初期値は80h(ex:sub80)です。

## [機能]

トレースバッファの内容を表示します。

トレース中にこのコマンドを発行した場合、強制的に取り込みを終了します。

## [表示内容]

```
>trace -10 asm
Cycle Sub Address Code Instruction EXT Stat
-00000d ---- bfc00000 0bf00100 j bfc00400 1111 TPC
-000002 ---- bfc00004 00000000 nop 1111 NSEQ
000001 ---- bfc00400 401a6000 mfc0 r26,$12 1111 NSEQ
000001 0001 bfc00404 00000000 nop 1111 ----
000001 0002 bfc00408 001ad502 srl r26,r26,14 1111 ----
000001 0003 bfc0040c 335a0001 andi r26,r26,1 1111 ----
000001 0004 bfc00410 13400003 beq r26,r0,bfc00420 1111 ----
000001 0005 bfc00414 00000000 nop 1111 ----
000004 ---- bfc00420 0ff001cc jal bfc00730 1111 NSEQ
000004 0001 bfc00424 00000000 nop 1111 ----
000007 ---- bfc00730 40806800 mtc0 r0,$13 1111 NSEQ
000007 0001 bfc00734 00000011 mthi r0 1111 ----
000007 0002 bfc00738 00000013 mtlo r0 1111 ----
000007 0003 bfc0073c 0000e025 or r28,r0,r0 1111 ----
000007 0004 bfc00740 0000f025 or r30,r0,r0 1111 ----
000007 0005 bfc00744 3c0ab800 lui r10,b800 1111 ----
000007 0006 bfc00748 81421003 lb r2,1003(r10) 1111 ----
000007 0007 bfc0074c 00000000 nop 1111 ----
000007 0008 bfc00750 30420008 andi r2,r2,8 1111 ----
000007 0009 bfc00754 1c400002 bgtz r2,bfc00760 1111 ----

>trace -10 ttag1
Cycle Sub Address Code Instruction EXT Stat
-00000d ---- bfc00000 0bf00100 j bfc00400 1111 TPC
time= 000,000,000,000.0uS
-000002 ---- bfc00004 00000000 nop 1111 NSEQ
time= 000,000,000,000.5uS
000001 ---- bfc00400 401a6000 mfc0 r26,$12 1111 NSEQ
time= 000,000,000,004.0uS
```

```

000001 0001 bfc00404 00000000 nop 1111 ----
000001 0002 bfc00408 001ad502 srl r26, r26, 14 1111 ----
000001 0003 bfc0040c 335a0001 andi r26, r26, 1 1111 ----
000001 0004 bfc00410 13400003 beq r26, r0, bfc00420 1111 ----
000001 0005 bfc00414 00000000 nop 1111 ----
000004 ---- bfc00420 0ff001cc jal bfc00730 1111 NSEQ
time= 000, 000, 000, 005. 2uS

000004 0001 bfc00424 00000000 nop 1111 ----
000007 ---- bfc00730 40806800 mtc0 r0, $13 1111 NSEQ
time= 000, 000, 000, 011. 9uS

000007 0001 bfc00734 00000011 mthi r0 1111 ----
000007 0002 bfc00738 00000013 mtlo r0 1111 ----
000007 0003 bfc0073c 0000e025 or r28, r0, r0 1111 ----
000007 0004 bfc00740 0000f025 or r30, r0, r0 1111 ----

>trace -10 ttag2
Cycle Sub Address Code Instruction EXT Stat
-00000d ---- bfc00000 0bf00100 j bfc00400 1111 TPC
-000002 ---- bfc00004 00000000 nop 1111 NSEQ
time= 000, 000, 000, 000. 5uS
000001 ---- bfc00400 401a6000 mfc0 r26, $12 1111 NSEQ
time= 000, 000, 000, 003. 5uS
000001 0001 bfc00404 00000000 nop 1111 ----
000001 0002 bfc00408 001ad502 srl r26, r26, 14 1111 ----
000001 0003 bfc0040c 335a0001 andi r26, r26, 1 1111 ----
000001 0004 bfc00410 13400003 beq r26, r0, bfc00420 1111 ----
000001 0005 bfc00414 00000000 nop 1111 ----
000004 ---- bfc00420 0ff001cc jal bfc00730 1111 NSEQ
time= 000, 000, 000, 001. 2uS
000004 0001 bfc00424 00000000 nop 1111 ----
000007 ---- bfc00730 40806800 mtc0 r0, $13 1111 NSEQ
time= 000, 000, 000, 006. 7uS
000007 0001 bfc00734 00000011 mthi r0 1111 ----
000007 0002 bfc00738 00000013 mtlo r0 1111 ----
000007 0003 bfc0073c 0000e025 or r28, r0, r0 1111 ----
000007 0004 bfc00740 0000f025 or r30, r0, r0 1111 ----
000007 0005 bfc00744 3c0ab800 lui r10, b800 1111 ----

```

Cycle:            トレースバッファ内の位置を16進数で相対的に表示しています。トリガポイント位置の近辺または、トレースの最終フレームを0としています。

Sub:               分岐や実行命令数などの情報から解析して生成したサイクルの番号です。

Address:           実行アドレスまたは、バスサイクルのアドレスを表示します。

Code:              命令コードまたは、バスサイクルのデータを表示します。

Instruction:       命令のニーモニックまたは、バスの種類を表示します。

EXT:               外部入力端子EX13.0の状態をビット列で表示します。

Stat:              表示にもとになるトレースパケットの種別を表示します。

          TPC:       命令から追跡できない分岐が発生

          EXP:       例外事象の発生

          LSEQ:      256命令以上、連続した実行が発生

          NSEQ:      分岐が発生

time =             タイムタグの表示

備考：タイムタグは、CPUから分岐情報が出力された時点のものです。分岐情報の出力は、実際の実行時間に対し遅れがあり、この遅れは一定ではありません。したがって、測定値には潜在的な誤差があります。また、実行直後の測定値は不定ですので無視してください。

tmodeコマンド

[書式]

tmode

[パラメータ]

なし

[機能]

トレースの設定状態を表示します。

t d a t a \_ d l yコマンド

## [書式]

tdata\_dly [off|small|medium|large]

## [パラメータ]

off: 補正しません。  
small: 最小の補正をします。  
medium: 中程度の補正をします。  
large: 最大の補正をします。(初期値)

## [機能]

トレースクロックに対するトレースデータのセットアップ時間を調整するためのコマンドです。セットアップ時間はoffが一番小さく、largeが一番大きくなります。なお、実際のセットアップ値は使用するRTE-xxxx-TP本体やケーブルに依存しますので、各本体の仕様を確認ください。

## [補足]

通常は初期値から変更する必要はありませんが、CPUやボードの状態によっては調整が必要になる場合があります。  
このコマンドはRTE-2000(H)-TPでのみ使用できるコマンドです。

verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

ICEの制御用ファームウェアのバージョンを表示します。