付録．Ａ　ＫＩＴ－Ｖ８３１／２－ＴＰ　内部コマンド

本書は、ＫＩＴ－Ｖ８３１／２－ＴＰ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方法は各デバッガのマニュアルを参照ください。

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

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にのみ補助的にご使用ください。ご使用になるデバッガが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おこし、いずれかの動作が異常になる場合があります。

コマンド書式

ＫＩＴ－Ｖ８３１／２－ＴＰの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。）

ａｂｐ，ａｂｐ１，ａｂｐ２，ａｂｐ３，ａｂｐ４コマンド

［書式］

abp[1|2|3|4] [ADDR] [io|mem]
abp [ADDR] [io|mem] /del
abp{1|2|3|4} /del
abpd{1|2|3|4}

［パラメータ］

abp: abpコマンドを指定します。abpで入力した場合、未使用のチャンネルを自動的に使用します。
明示的にチャンネルを指定する場合は、abp1,abp2,abp3,abp4を使用します。
ADDR: アドレスを１６進数で指定します。
io: i/o空間のアクセスを条件に指定します。
mem: memory空間のアクセスを条件に指定します。
/del: 指定した条件を解除します。
abpdx: xで指定したabpチャンネルを削除します。

［機能］

４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。

［入力例］

abp 1000 mem
1000h番地のメモリアクセスにブレークを設定します。
abp2 2000 io
1000h番地のioアクセスにブレークを設定します。
abp 2000 io /del
1000h番地のioアクセスに設定したブレーク条件を解除します。
abp1 /del
abp1の条件を解除します。（abpd1と同じです）
abpd1
abp1の条件を解除します。（abp1 /delと同じです）

ｃｍｃｒ，ｄｃｔｒ，ｉｃｔｒコマンド

ｃｍｃｒ コマンド

［書式］

cmcr [=]value

［機能］

CMCR（キャッシュメモリ・コントロールレジスタ）に値を設定します。

ｄｃｔｒ コマンド

［書式］

dctr [all]

［機能］

DCTR レジスタを表示します。
レジスタは256個ありますが、バリッドビットが有効になっているレジスタのみ表示します。
ただし、all を指定した場合は、全てのレジスタ値を表示します。
DCTR レジスタは、I/O 空間 f2000000h-f2000fffh にマップされています。

ｉｃｔｒ コマンド

［書式］

ictr [all]

［機能］

ICTR レジスタを表示します。
レジスタは128個ありますが、バリッドビットが有効になっているレジスタのみ表示します。
ただし、all を指定した場合は、全てのレジスタ値を表示します。
ICTR レジスタは、I/O 空間 fa000000h-fa000fffh にマップされています。

ｅｎｖコマンド

［書式］

env [[!]auto] [[!]reset] [[!]nmi] [[!]hldrq] [[!]int{00|01|02|03}] [[!]int{10|11|12|13}]
[jtag{25|12|5|2|1|500|250|100}] [!]verify [inone|istack|iaddr ADDR]

［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。

[!]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。

[!]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。

[!]hldrq: HLDRQ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{00|01|02|03}: INT00-03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{10|11|12|13}: int10-13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{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。
> 尚、RTE-100-TPでは、jtag25,jtag12以外のパラメータは無効です。

[!]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も、CPUからアクセス(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。

[inone|istack|iaddr ADDR]: V832の内蔵命令RAMをアクセスする時に必要なワーク領域を指定します。モニタが使用後は、元のデータに戻します。ブロック転送可能なRAMを指定して下さい。

inone: 指定されていません。

istack: 現在のスタックからマイナス方向に32バイト使用します。

iaddr ADDR: ADDRで指定したアドレスからプラス方向に32バイト使用します。

［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。

初期値は、以下の通りです。

```
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プローブを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     = 12.5MHz (jtag12)
 Verify        = verify off (!verify)
Signals Mask:
 INT00         = NO MASK (!int00)
 INT01         = NO MASK (!int01)
 INT02         = NO MASK (!int02)
 INT03         = NO MASK (!int03)
 INT10         = NO MASK (!int10)
 INT11         = NO MASK (!int11)
 INT12         = NO MASK (!int12)
 INT13         = NO MASK (!int13)
 NMI           = NO MASK (!nmi)
 RESET         = NO MASK (!reset)
 HLDRQ         = NO MASK (!hldrq)
IRAM Settings:
 IRAM work     = stack (istack)
```

［入力例］

```
env reset !nmi jtag25
```

RESETをマスクし、NMIをマスクしません。
JTAGクロックを25MHzに設定します。

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。

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は、アクセスサイズを区別して、I/O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。
アドレスが省略すると、前回のアドレスが指定されたものとみなします。

［使用例］

inb 1000
　　1000Hからバイト(8-bit)でI/Oリードします。
inh 1000
　　1000Hからハーフワード(16-bit)でI/Oリードします。
inw 1000
　　1000Hからワード(32-bit)でI/Oリードします。

ｉｎｉｔコマンド

［書式］
init

［パラメータ］
なし

［機能］
ＫＩＴ－Ｖ８３１／２－ＴＰ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。

ｊｒｅａｄ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)から読み出す為のコマンドです。（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 ffff0000 100

ffff0000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］

ＫＩＴ－Ｖ８３１／２－ＴＰではメモリ参照の高速化を図るため、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下さい。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 10000 1000

10000h番地から1000h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 10000 1000
 No Memory Cache Area
 No. Address   Length
 1   010000    001000
 2   fff000    001000
```

ｎｃｄ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 2

ブロック番号２をメモリキャッシュの除外エリアから削除します。

```
>nc
 No Memory Cache Area
 No. Address   Length
 1   020000    000100
 2   010000    001000

>ncd 2
 No Memory Cache Area
 No. Address   Length
 1   020000    000100
```

ｎｓｂｐ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ハーフ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 10000 20000

10000h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 100000 20000
Num Address  Length
01  00100000 00020000
```

ｎｓｂｐｄコマンド

［書式］

nspbd [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］

nsbp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  00100000 00200000
02  00400000 00010000

>nsbpd 1
Num Address  Length
01  00400000 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、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］

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他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。
指定領域は、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。
通常は、指定する必要はありません。

［使用例］

nrom 0 2000

0h番地から2000バイトを強制ユーザ領域に指定します。

```
>nrom 0 1000
No. Address     Length
 1  00000000    00001000

>nrom 10000 100
No. Address     Length
 1  00000000    00001000
 2  00010000    00000100
```

ｎｒｏｍｄ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 10000 8000
No. Address     Length
 1  00000000    00001000
 2  00010000    00008000

>nromd 1
No. Address     Length
 1  00010000    00008000
```

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、I/O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。
アドレス、データが省略すると、前回の値が指定されたものとみなします。

［使用例］

outb 1000 12
　　1000Hへバイトデータ：12hをI/Oライトします。
outh 1000 1234
　　1000Hへハーフワードデータ：1234hをI/Oライトします。
outh 1000 12345678
　　1000Hへワードデータ：12345678hをI/Oライトします。

ｒｅｓｅｔコマンド

［書式］
reset

［パラメータ］
なし

［機能］
ＫＩＴ－Ｖ８３１／２－ＴＰの対象エミュレーションＣＰＵをリセットします。

ｒｏｍコマンド

［書式］

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

［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。

［機能］

ROMのエミュレーション環境の設定を行います。設定は変更が必要なパラメータだけを入力してください。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。初期値は、LENGTH = 0（使用しない）になっています。

［入力例］

rom 100000 40000 1m rom16 bus16

27C1024(1M-bitの16bit-ROM)を100000hから256Kバイト(40000h)エミュレートします。この場合、結果的に16bit-romを２個使用してエミュレートします。

rom 0 40000 2m rom rom16 bus32

27c2048(2M-bitの16bit-ROM)を0x0から256Kバイト(40000h)エミュレートします。この場合、結果的に16bit-ROMを１個使用してエミュレートします。

＜備考＞

ｒｏｍコマンドで指定した範囲へのアクセスは、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ます。したがって、表示時、正しく見えていても、プロセッサから正しくROMにアクセスできない場合もあります。この場合、jreadコマンドを使用して確認するか、envコマンドでverifyをONにして書き込み（ダウンロード）を行って下さい。そうすることにより、エミュレーションしているROMの内容をCPUのバスを介して読み出すことでチェックできます。

ｓｆｒ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。

<V831>

リード・ライトレジスタ:

IGP BCTC DBC DRC PRC ASIM00 ASIM01 CSIM0 SIO0
BRG0 BPRM0 TMC1 TMC4 TOC1 TOVS PORT PM PC CGC
IMR IMOD PWC0 PWC1 PIC RFC DSA0H DSA1H DSA2H DSA3H
DSA0L DSA1L DSA2L DSA3L DDA0H DDA1H DDA2H DDA3H
DDA0L DDA1L DDA2L DDA3L DBC0H DBC1H DBC2H DBC3H
DBC0L DBC1L DBC2L DBC3L DCHC0 DCHC1 DCHC2 DCHC3 DC
CM4 CC10 CC11 CC12 CC13 TUM1

ライトオンリーレジスタ:

TXS0L ICR TXS0

リードオンリーレジスタ:

ASIS0 RXB0L IRR RXB0 TM1 TM4

<V832>

リード・ライトレジスタ:

PORT PM PC
BCTC DBC PWC0 PWC1 RFC PRC
DSA0H DSA0L DDA0H DDA0L DBC0H DBC0L DCHC0
DSA1H DSA1L DDA1H DDA1L DBC1H DBC1L DCHC1
DSA2H DSA2L DDA2H DDA2L DBC2H DBC2L DCHC2
DSA3H DSA3L DDA3H DDA3L DBC3H DBC3L DCHC3 DC
TOVS TUM1 TMC1 TOC1 CC10 CC11 CC12 CC13 TMC4 CM4
ASIM00 ASIM01 CSIM0 SIO0 BRG0 BPRM0
IGP IMR IMOD
CGC PMR
PORTA PAM PAC PORTB PBM PBC
PIC0 PIC1 SDC

ライトオンリーレジスタ:

TXS0 TXS0L ICR SDM

リードオンリーレジスタ:

TM1 TM4 ASIS0 RXB0 RXB0L IRR

［機能］

ＳＦＲレジスタ値の設定と表示を行います。

［使用例］

sfr IGP

IGPレジスタの値を表示します。

sfr IGP 2

IGPレジスタに2hを設定します。

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。

ｔｐコマンド

［書式］

tp [ADDR]

［パラメータ］

ADDR: 偶数アドレスを１６進数で指定します。（A0は、常に０に補正されます）

［機能］

トレースのトリガポイントを指定します。

［使用例］

tp ffff0000

ffff0000hの命令実行をトリガポイントとして指定します。

［注意事項］

tronコマンドでdelay modeが指定されている場合、トリガポイントの指定は無視されます。
この場合、tron !delayと入力してdelay modeを解除して下さい。

ｔｅｐコマンド(V832 のみ)

［書式］

tep [ADDR] [/del]

［パラメータ］

ADDR: 命令実行アドレスを１６進数で指定します。

/del: 指定したアドレスを解除します。

［機能］

トレースの停止ポイント（アドレス）を指定します。

［使用例］

tep ffff0000

ffff0000hの命令実行をトレースの停止アドレスとして指定します。

［備考］

トレース情報は、CPUからの出力時点でオーバフローを起こす場合があります。このような場合、トレースしたい事象の直前でトレースを開始するように設定することで、目的に近辺でのオーバフローを回避できます。

開始アドレスを指定しない場合は、tronコマンドを発行した時点で強制的に開始します。

このコマンドで指定した開始アドレスは、tronを発行した時点で有効になります。

ｔｓｐコマンド

［書式］

tsp [ADDR] [/del]

［パラメータ］

ADDR: 命令実行アドレスを１６進数で指定します。

/del: 指定したアドレスを解除します。

［機能］

トレースの開始ポイント（アドレス）を指定します。

［使用例］

tsp ffff0000

ffff0000hの命令実行をトレースの開始アドレスとして指定します。

［備考］

トレース情報は、CPUからの出力時点でオーバフローを起こす場合があります。このような場合、トレースしたい事象の直前でトレースを開始するように設定することで、目的に近辺でのオーバフローを回避できます。

開始アドレスを指定しない場合は、tronコマンドを発行した時点で強制的に開始します。

このコマンドで指定した開始アドレスは、tronを発行した時点で有効になります。

ｔｄ１，ｔｄ２コマンド

［書式］

td1 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]
td2 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]

［パラメータ］

DADDR: アドレスを１６進数で指定します。
４バイトのバウンダリに補正されます。

ignore|ioread|iowrite|ioacc|memread|memwrite|memacc: ステータスを指定します。

ignore:	don't care
ioread:	IO空間へのリード
iowrite:	IO空間へのライト
ioacc:	IO空間へのリード／ライト
memread:	memory空間へのリード
memwrite:	memory空間へのライト
memacc:	memory空間へのリード／ライト

/del: 設定の解除

［機能］

トレースに取り込むデータサイクルの条件を設定します。

［使用例］

td1 fe000000 memread
fe000000h番地のからのメモリリードをトレースします。

ｔｒｏｎコマンド

［書式］

tron [DELAY] [[!]delay] [noreal|real] [noignore|ev{[0]|..|[8]}] [noext|nega|posi]
[[!]td1] [[!]td2]

［パラメータ］

DELAY = 0..1ffff: ディレイカウンタです。
トリガ成立後にメモリの取り込むフレーム数を十進数で指定します。

[!]delay: 強制ディレイモードを指定します。!で通常のモードの指定に戻ります。
強制ディレイモードは、トレース開始後、ディレイカウンタ数分のトレースを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。

[noreal|real]: トレースのモードを指定します。

real: リアルタイム実行モード

normal: 非リアルタイム実行モード（rte4win32 ver4.35以上で対応しています。）
この設定では、トレース取り込み中、ブレークが頻繁に介在しますので、実行速度が10倍以上遅くなる場合があります。

noignore|ev{[0]|..|[8]}
トレースに取り込まないイベントを指定します。

noignore: 全てのイベントを取り込みます。通常の指定です。

ev0..8:

ev0: Exception の発生情報を取り込みません。

ev1: Interrupt の発生情報を取り込みません。

ev2: Condition Jump の発生情報を取り込みません。

ev3: PC relative の発生情報を取り込みません。

ev4: JAL の発生情報を取り込みません。

ev5: RETI to の発生情報を取り込みません。

ev6: RETI from の発生情報を取り込みません。

ev7: Jump register indirect to の発生情報を取り込みません。

ev8: Jump register indirect from の発生情報を取り込みません。

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。

[!]td1: トレースデータ条件１(td1)をトリガとして指定します。！で解除します。

[!]td2: トレースデータ条件２(td2)をトリガとして指定します。！で解除します。

> 備考：[[!]td1] [[!]td2]は、RTE-100-TPでは、無効です。
> td1とtd2の条件が重複するサイクルを指定している場合、トリガの条件は、td1を指定してください。td2では、トリガがかからない場合があります。

［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］

delayモードで無条件に1ffffサイクル分トレースします。

```
>tron delay 1ffff
Trace Settings:
 Start Address= Force
 Delay Count  = 0001fffd
 Trace Mode   = Real Time (real)
 Delay Mode   = Enable (delay)
 Ignore Event = None (noignore)
 Ext Trigger  = Disable (noext)
 TD1 Trigger  = Disable (!td1)
 TD2 Trigger  = Disable (!td2)

 Data Trace 1 = Disable (ignore)
 Data Trace 2 = Disable (ignore)
 Trig Address = Disable
```

fe00000h番地の命令実行をトリガにして、100h番地IOREADをデータトレースします。
トリガ後の取り込みサイクル(DELAY)は、ffffhを指定します。

```
>tp fe000000                                  <<トリガの指定
 Trig Address = fe000000

rte3>td1 100 ioread                           <<データトレースの指定
 Data Trace 1 = 00000100 I/O Read (ioread)
 Data Trace 2 = Disable (ignore)

>tron ffff                                    <<トレースの開始
Trace Settings:
 Start Address= Force
 Delay Count  = 0000ffff
 Trace Mode   = Real Time (real)
 Delay Mode   = Disable (!delay)
 Ignore Event = None (noignore)
 Ext Trigger  = Disable (noext)
 TD1 Trigger  = Disable (!td1)
 TD2 Trigger  = Disable (!td2)

 Data Trace 1 = 00004444 I/O Read (ioread)
 Data Trace 2 = Disable (ignore)
```

ｔｒｏｆｆ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|ttag1|ttag2] [subNN]

［パラメータ］

POS=±0..1ffff	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。
[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

備考：ttag1|ttag2の指定は、RTE-100-TPでは無効です。

subNN:	実際に取り込まれる一つの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］：アセンブラモード

```
>trace -30 asm
   Cycle   Sub   Address   Code          Instruction                    EXT   Stat
  -000032 0000 fffffff0 bc20ffff    movhi ffffh,r0,r1              1111 FTRC
  -00001e 0000 fffffff0 bc20ffff    movhi ffffh,r0,r1              1111 RETI2
  -00001e 0001 fffffff4 a0210000    movea 0000h,r1,r1                1111 -----
  -000014 0000 fffffff8 1801        jmp   [r1]                       1111 JREG1
* -00000a 0000 ffff0000 7010        ldsr  r0,DPC                     1111 JREG2
   000000 0001 ffff0002 7011        ldsr  r0,DPSW                     1111 -----
   000000 0002 ffff0004 7000        ldsr  r0,EIPC                    1111 -----
   000000 0003 ffff0006 7001        ldsr  r0,EIPSW                    1111 -----
   000000 0004 ffff0008 7002        ldsr  r0,FEPC                    1111 -----
   000000 0005 ffff000a 7003        ldsr  r0,FEPSW                    1111 -----

>trace -30 ttag1
   Cycle   Sub   Address   Code          Instruction                    EXT   Stat
  -000032 0000 fffffff0 bc20ffff    movhi ffffh,r0,r1              1111 FTRC
                              time = 000,000,000,000.0uS
  -00001e 0000 fffffff0 bc20ffff    movhi ffffh,r0,r1              1111 RETI2
                              time = 000,000,742,703.0uS
  -00001e 0001 fffffff4 a0210000    movea 0000h,r1,r1                1111 -----
  -000014 0000 fffffff8 1801        jmp   [r1]                       1111 JREG1
                             time = 000,000,742,703.3uS
* -00000a 0000 ffff0000 7010        ldsr  r0,DPC                     1111 JREG2
```

```
                              time = 000,000,742,704.7uS
   000000 0001 ffff0002 7011        ldsr   r0,DPSW                  1111 -----
   000000 0002 ffff0004 7000        ldsr   r0,EIPC                  1111 -----
   000000 0003 ffff0006 7001        ldsr   r0,EIPSW                 1111 -----

>trace -30 ttag2
    Cycle   Sub   Address   Code        Instruction                    EXT   Stat
   -000032 0000 fffffff0 bc20ffff    movhi  ffffh,r0,r1              1111 FTRC
   -00001e 0000 fffffff0 bc20ffff    movhi  ffffh,r0,r1              1111 RETI2
                              time = 000,000,000,001.6uS
   -00001e 0001 fffffff4 a0210000    movea  0000h,r1,r1                1111 -----
   -000014 0000 fffffff8 1801        jmp    [r1]                       1111 JREG1
                              time = 000,000,000,000.3uS
 * -00000a 0000 ffff0000 7010        ldsr   r0,DPC                     1111 JREG2
                              time = 000,000,000,001.4uS
   000000 0001 ffff0002 7011        ldsr   r0,DPSW                  1111 -----
   000000 0002 ffff0004 7000        ldsr   r0,EIPC                  1111 -----
   000000 0003 ffff0006 7001        ldsr   r0,EIPSW                 1111 -----
   000000 0004 ffff0008 7002        ldsr   r0,FEPC                  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EXI3..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。

- RD#1: データトレース(dt1)のリードサイクルの発生
- TRIG: トリガアドレスの発生
- FAIL: トレースデータの取りこぼしが発生
- JMPR: PC相対分岐命令の分岐元アドレスの発生
- JAL: JAL命令による分岐元アドレス発生
- RETI1: RTEI命令による分岐元アドレスの発生
- JREG1: レジスタ間接分岐命令による分岐元アドレスの発生
- FTRC: トレースの開始
- WR#1: データトレース(dt1)のライトサイクルの発生
- RETI2: RETI命令による分岐先アドレスの発生
- JREG2: レジスタ間接分岐命令による分岐先アドレスの発生
- INTR: マスカブル割込みによる分岐の発生
- EXP: 例外事象、または、NMIによる分岐の発生
- CJMP: 条件分岐命令による分岐元アドレスの発生
- RD#2: データトレース(dt2)のリードサイクルの発生
- WR#2: データトレース(dt2)のライトサイクルの発生
- *: トリガポイント（多少ずれる場合があります）

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。

ｖｅｒコマンド

［書式］
ver

［パラメータ］
なし

［機能］
ＫＩＴ－Ｖ８３１／２－ＴＰのバージョンを表示します。