

## 付録 . A K I T - V R 5 4 0 0 - T P 内部コマンド

本書は、K I T - V R 5 4 0 0 - 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方法は各デバッガのマニュアルを参照ください。

### 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 - V R 5 4 0 0 - T P 内部コマンド.....	1
コマンド一覧.....	1
コマンド書式.....	2
オプションブレーク : B P O P T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
実行系イベント : E V Aコマンド.....	7
アクセス系イベント : E V Eコマンド.....	8
ヘルプ : H E L Pコマンド.....	9
ポート入力 : I N B , I N H , I N W , I N Dコマンド.....	10
初期化 : I N I Tコマンド.....	11
J T A Gリード : J R E A Dコマンド.....	12
デバッガキャッシュの解除 : N Cコマンド.....	13
デバッガキャッシュの設定 : N C Dコマンド.....	14
ソフトブレーク禁止領域の設定 : N S B Pコマンド.....	15
ソフトブレーク禁止領域の設定 : N S B P Dコマンド.....	16
強制ユーザ領域の設定 : N R O Mコマンド.....	17
強制ユーザ領域の設定 : N R O M Dコマンド.....	18
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.....	19
C P Uリセット : R E S E Tコマンド.....	20
E . R O Mの環境設定 : R O Mコマンド.....	21
T L B : T L B 3 2 , T L B 6 4コマンド.....	22
シンボル : S Y M F I L E , S Y Mコマンド.....	23
トレースの設定、開始 : T R O Nコマンド.....	24
トレースの強制終了 : T R O F Fコマンド.....	25
トレース表示 : T R A C Eコマンド.....	26
トレースの設定確認 : T M O D Eコマンド.....	28
バージョン表示 : V E Rコマンド.....	29

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

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。)

## b p o p tコマンド

### [ 書式 ]

bpopot [[!]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 ]

bpopot eve

eveをブレイク条件に設定します。

bpopot !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 ]

キャッシュ操作の為のコマンドです。

## [ 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コマンド

## [書式]

```
env [!]auto [!]nmi [!]int [jtag{25!12!5!2!1!500!250!100}] [!]verify [!]hispeed
[work 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。

[!]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{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-200-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。

[!]hispeed:メモリへの書き込みにおいて、高速モードを指定します。!でノーマルモードです。

備考：高速モードの指定は、ROMプローブを接続されていることが条件です。このモードを指定した場合、128バイト以上の連続書き込み時に限り、エミュレーションしているROMに一時的に制御プログラムを配置してフォアグラウンドで実行します。CPUがROM上のプログラムを正しく実行できることが必要条件です。ハードウェアのデバッグが完全に完了してからご使用下さい。ノーマルモードは、JTAG経由で書き込みます。

[work ADDR]:キャッシュのクリアや初期化処理を実行する為の領域を指定します。必ず、起動直後にアンキャッシュ上のRAMを指定してください。VR5400では、キャッシュ処理の為にユーザシステム上のRAMが必要です。指定した番地から128バイトの領域を、モニタは保存することなしに使用します。

備考：キャッシュ領域へのアクセスにいかない限りこの領域は使用しませんので、アンキャッシュ領域でRAMがアクセスできることを確認した後にキャッシュ領域へのアクセスを行ってください。

## [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)
Download   = Normal (!hispeed)
Signals Mask:
NMI        = NO MASK (!nmi)
INT        = NO MASK (!int)
Cache Clear Settings:
Work Memory = ffffffff (work ffffffff)
```

[ 入力例 ]

```
env !nmi verify work a0010000 hispeed
```

NMIをマスク、verifyをON、ワークを0xa0010000番地、高速モードの指定をします。

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と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。

## eve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。



## 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, 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、リードを行ないます。

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]

KIT-VR5400-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いません。

### [ 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、最小値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 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
```

## 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



## 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、ロング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 a0000000 2000

a0000000h番地から2000バイトを強制ユーザ領域に指定します。

>nrom a0000000 1000

No.	Address	Length
-----	---------	--------

1	a0000000	00001000
---	----------	----------

>nrom a010000 100

No.	Address	Length
-----	---------	--------

1	a0000000	00001000
---	----------	----------

2	a0010000	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 a0010000 8000
```

No.	Address	Length
1	a0000000	00001000
2	a0010000	00008000

```
>nromd 1
```

No.	Address	Length
1	a0010000	00008000

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、ライトを行いません。

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。

outw 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]

KIT-VR5400-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 [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 ]

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 bfc0000 40000 1m rom16 bus32 little
```

27C1024(1M-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-romを1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-romを1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。

## t l b 3 2 , t l b 6 4 コマンド

### [ 書式 ]

tlb32 [all|INDEX [MASK HI LO0 LO1]]

tlb64 [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。

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。

## 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。
```

tronコマンド

## [ 書式 ]

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

## [ パラメータ ]

DELAY = 0..1ffff デレイカウンタ

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。!で通常の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 1ffff
```

delayモードで無条件に1ffff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、1ffffサイクル分トレースして、終了します。

```
tron !delay eve ffff
```

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、ffff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、ffff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、ffff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。



## 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] [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の指定します。

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-200-TPでは無効です。

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:       命令のニックネームまたは、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。

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。

## verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

KIT-VR5400-TPのバージョンを表示します。