

付録 B KIT-ARM11xx-TP (-H) 内部コマンド

本書は、KIT-ARM11xx-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

コマンド一覧	B-1
コマンド書式	B-2
アクセス方法に関する環境設定	: ACCENVコマンド B-3
ARM系オプション機能の設定	: ARMOPTコマンド B-4
アクセス系ブレークポイント	: ABP, ABP1, ABP2コマンド B-5
実行系ブレークポイント	: IBP, IBP [1..6] コマンド B-6
環境設定	: ENV, EMEMSTATコマンド B-7
ヘルプ	: HELPコマンド B-8
高速ダウンロードの設定	: HSDOWNLOADコマンド B-9
INPUT	: INB, INH, INWコマンド B-10
初期化	: INITコマンド B-11
JTAGリード	: JREADコマンド B-12
デバッガキャッシュ領域の解除	: NCCOMMAND B-13
デバッガキャッシュ領域の設定	: NCDコマンド B-14
ソフトブレーク禁止領域の設定	: NSBPコマンド B-15
ソフトブレーク禁止領域の解除	: NSBPDコマンド B-16
強制ユーザ領域の設定	: NRROMコマンド B-17
強制ユーザ領域の解除	: NROMDコマンド B-18
OUTPUT	: OUTB, OUTH, OUTWコマンド B-19
CPUリセット	: RESETコマンド B-20
E. ROMの設定	: ROM1..ROM4コマンド B-21
シンボル	: SYMFILE, SYMコマンド B-23
ベクタブレーク	: VECTORBPコマンド B-24
イベントの統合[ETM]	: EVTコマンド B-25
コンテキストIDの設定[ETM]	: RSCCIDコマンド B-26
カウンタの設定[ETM]	: RSCCNT [1..4] コマンド B-27
拡張外部入力セレクタの設定[ETM]	: RSCEXTISEL[1.4]コマンド B-28
リソースの設定[ETM]	: RSCSINGL [1..16] コマンド B-29
リソース環境の表示[ETM]	: RSCSTATコマンド B-30
トレースイネーブルの設定[ETM]	: TRCENABLEコマンド B-31
ビューデータの設定[ETM]	: VIEWDATAコマンド B-32
FIFOの設定[ETM]	: FIFOFULLコマンド B-33
トレースの環境の設定[ETM]	: TENVコマンド B-34
トレースの設定状態の表示[ETM]	: TMODEコマンド B-35
トレースの開始[ETM]	: TRONコマンド B-37
トレースの強制終了[ETM]	: TROFFコマンド B-38
トレースの表示[ETM]	: TRACEコマンド B-39
トレースのファイル保存[ETM]	: FTRACEコマンド B-41
トレースデータのディレイ調整[ETM]	: TDATA_DLYコマンド B-42
バージョン表示	: VERコマンド B-43



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おこし、いずれかの動作が異常になる場合があります。[ETM]のコマンドが使用できるのは、CPU に ETM が実装されている場合に限りです。

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。)

accenvコマンド

[書式]

```
accenv [[!]jall] [[!]jsbp]
```

[パラメータ]

[[!]jall: エミュレーション・メモリに割り付けしている空間に対するアクセス方法を指定します。jallはJTAG経由で行い、!jallはホストからエミュレーション・メモリへの直接アクセスを行います。(初期値)

備考: 通常は、初期値でご使用ください。

[[!]jsbp: エミュレーション・メモリに割り付けしている空間に対するソフトウェア・ブレークポイントを設定する際のアクセス方法を指定します。jsbpはJTAG経由で行い、!jsbpはホストからエミュレーション・メモリへの直接アクセスを行います。(初期値)
BIF-Dxx-xxが接続されている場合、jsbpでJTAG経由でソフトウェア・ブレークポイントの設定／解除を行う際のメモリ・ライト時は、エミュレーション・メモリを一時的に書込み可とし、非同期モードに変更してからメモリ・ライトを行います。これは、romコマンドの書込み設定(wren)に依存しませんので、!wrenのままでも問題ありません。

備考: 通常は、初期値でご使用ください。

“accenv jsbp”に設定している場合、ブレーク中にユーザ・システム上のARMコア以外のデバイスがDMA等によりエミュレーション・メモリが割り付いている空間をアクセスした場合、そのアクセスは正常に行えない場合があります。

[機能]

エミュレーション・メモリに対するアクセス方法を指定するコマンドです。

備考: 通常、エミュレーション・メモリに割付けた領域へのアクセスは、エミュレーション・メモリに対しホストから直接アクセスを行い、エミュレーションしているCPUからアクセス(=JTAGアクセス)することはありません。このコマンドは、そのようなケースのアクセスを強制的にエミュレーションしているCPU(JTAG)から行うようにするためのものです。

注意: jallを指定した状態で、大量のデータをエミュレーション・メモリにダウンロードすると、通常よりも時間がかかります。

[入力例]

```
accenv jall
```

エミュレーション・メモリに対し、JTAG経由でアクセスします。

```
accenv jsbp
```

エミュレーション・メモリ領域のソフトウェア・ブレークポイントの書換えをJTAG経由でアクセスします。

```
accenv !jall !jsbp
```

通常のアクセスに戻します。

armopt コマンド

[書式]

```
armopt [[!]resetrecovery] [[!]resetautorun]
```

[パラメータ]

[!]resetrecovery: ユーザシステムからのリセットでICE制御レジスタがクリアされてしまうGPUにおいて、ユーザプログラム実行中にリセットが入った場合、その時点で強制的にブレークして、ICE制御レジスタの復元を行う時に指定します。!で復元は行いません。(初期値)

[!]resetautorun: resetrecoveryを指定した時に、自動的に実行を再開させる場合に指定します。!で自動的に実行は行いません。

備考: Partnerを使用する場合、resetautorunは指定しないでください。

[機能]

CPUのパワーオンリセットを入れるとICE制御レジスタがクリアされ、その後のICE制御が行えなくなりますので、それを回避するために用意されたコマンドです。

備考: これは実行中にリセットが入った時に機能するものです。この機能を使用する為には、JTAG-IFのSRST-信号が双方向で機能することが条件です。ブレーク中にリセットが入られた場合は、その後デバッガからリセットコマンドを発行してからデバッグを継続してください。

[入力例]

```
armopt resetrecovery !resetautorun
resetrecovery機能をONし、自動再実行は行いません。
armopt !resetrecovery !resetautorun
初期状態に戻します。
```

abp, abp1, abp2コマンド

[書式]

```
abp{1..16} [read|write|accs] [ADDR] [bytesel {0|1} {0|1} {0|1} {0|1}]
           [cidibp{1..16}|nocid] [user|privilege|both_mode]
abp{1..16} /del
```

[パラメータ]

abp{1..16}: abp1または、abp2を条件指定に先立ち入力します。

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。

ADDR: アドレス条件の指定

ADDR: アドレスを16進数で指定します。下位2-bitは無効です。

bytesel {0|1} {0|1} {0|1} {0|1}: バイト単位で比較対象を指定します。

0000: The breakpoint never hits

xxx1: ADDR+0番地を比較対象とします。

xx1x: ADDR+1番地を比較対象とします。

x1xx: ADDR+2番地を比較対象とします。

1xx : ADDR+3番地を比較対象とします。

cidibp{1..16}|nocid: コンテキストIDを指定します。指定できるのはibp5とibp6だけです。

cidibp{1..16}: コンテキストIDを比較対象とするibpを指定します。

cidibp5: ibp5で指定したコンテキストIDを比較対象にします。

Cidibp6: ibp6で指定したコンテキストIDを比較対象にします。

nocid: コンテキストIDは比較対象としません。

user|privilege|both_mode: 実行時の特権レベルを指定します。

user: ユーザモードの指定

privilege: 特権モードの指定

both_mode: 特権レベルを無視

abp{1..16} /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をします。

[入力例]

```
abp1 1000 read bytesel 0011 nocid both_mode
abp1に1000/1001h番地からのリードにブレークを設定します。コンテキストIDと特権レベルは無視します。

abp2 1000 accs bytesel 1111 cidibp5 both_mode
abp2に1000/1001/1002/1003h番地のアクセスにブレークを設定します。
ibp5で指定したコンテキストIDを比較対象にし、特権レベル無視します。
ibp5へのコンテキストID値の設定はこのコマンドよりも先にしておく必要があります。

abp1 /del
abp1の条件を解除します。
```

ibp, abp 1.. 6 コマンド

[書式]

```

ibp{1..16} [exec] [ADDR] [bytesel {0|1}{0|1}{0|1}{0|1}]
           [cidibp{1..16}|nocid] [user|privilege|both_mode]
ibp{1..16} [cid] [CID] [user|privilege|both_mode]
ibp{1..16} /del

```

[パラメータ]

abp{1..16}: ibp1からibp6を条件指定に先立ち入力します。

exec: 実行アドレスを指定対象とします。

cid: コンテキストIDを指定対象とします。指定できるのは、ibp5とibp6だけです。

ADDR: アドレスを16進数で指定します。下位2-bitは無効です。

CID: コンテキストIDを16進数で指定します。

bytesel {0|1}{0|1}{0|1}{0|1}: バイト単位で比較対象を指定します。

0000: The breakpoint never hits

xxx1: ADDR+0番地を比較対象とします。

xx1x: ADDR+1番地を比較対象とします。

x1xx: ADDR+2番地を比較対象とします。

1xx : ADDR+3番地を比較対象とします。

cidibp{1..16}|nocid: コンテキストIDを指定します。指定できるのはibp5とibp6だけです。

cidibp{1..16}: コンテキストIDを比較対照とするibpを指定します。

cidibp5: ibp5で指定したコンテキストIDを比較対象にします。

Cidibp6: ibp6で指定したコンテキストIDを比較対象にします。

nocid: コンテキストIDは比較対象としません。

user|privilege|both_mode: 実行時の特権レベルを指定します。

user: ユーザモードの指定

privilege: 特権モードの指定

both_mode: 特権レベルを無視

abp{1..16} /del: 条件の解除を行います。

/del: 解除を指定します。

[機能]

実行系のブレークポイントの設定または解除をします。ibpの資源は、デバッガが内部的に使用するハードウェアブレークポイントとの共用です。

[入力例]

```

ibp1 exec 1000 bytesel 0011 nocid both_mode
   ibp1に1000/1001h番地の命令実行にブレークを設定します。
   コンテキストIDと特権レベルは無視します。
ibp5 cid 10 user
   ibp5にコンテキストIDとしてユーザモードの0x10を設定します。
ibp1 /del
   ibp1の条件を解除します。

```

env, ememstatコマンド

[書式]

```
env [!]auto [jtag[xxx].[yyy]]{M|K} [!]verify [!]int
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[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。

[!]int: int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。

[機能]

envコマンドは、エミュレーション環境の設定と状態を表示します。

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、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。

```
env
Probe:
Probe:
Unit      : RTE-2000(H)-TP
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:
INT        = NO MASK (!int)

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

[入力例]

```
env verify int
Verify機能をONにし、intをマスクします。
```

```
env !int
intのマスクを解除します。
```

```
env jtag40m
JTAGクロックを40MHzに設定します。
```

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。

hsdownloadコマンド

[書式]

```
hsdownload [[!]enable] [ADDRESS]
```

[パラメータ]

[!]enable: 高速ダウンロード機能を有効にします。!を付けることで無効です。

ADDRESS: 高速ダウンロードの為に高速ダウンロードモジュール(BIF-D03-xx)を割付けたチップセレクトの先頭アドレスを16進数で指定します。

備考：高速ダウンロード機能を有効にする為には、オプションのBIF-D03-xxが必要です。

[使用手順]

- ・高速ダウンロードモジュールとして、BIF-D03-xxを接続し、ボード上でこのモジュールに割り付けるチップセレクトをBIF-D03-xxのSW3で指定します。BIF-D03-xxはエミュレーション・メモリとしての機能もありますが、高速ダウンロードモジュールとして使う場合に必要な設定はSW3だけです。
- ・起動の手順は以下の通りです。
 1. CPUの初期設定(CLKやダウンロード先のMEMCの設定)
 2. 高速ダウンロードモジュール(BIF-D03-xx)にアクセスするためのMEMCの設定
 3. 高速ダウンロード機能のイネーブル処理

```
>hsdownload enable xxxxxxx
```

 - ・xxxxxxは高速ダウンロードモジュールを割付けたチップセレクトの先頭アドレス
 - ・このコマンドを発行した時点でxxxxxx番地で高速ダウンロードモジュール上の資源がアクセスできる状態であることが必要です。
 4. オブジェクトのロード

```
>rd xxxxx.bin
```

 - ・上記はPartnerデバグガの例です。
 5. 高速ダウンロード機能のディゼーブル処理

```
>&hsdownload !enable
```
 6. 高速ダウンロードモジュール(BIF-D03)へアクセスするために行ったMEMCの設定を戻す処理

備考：ご希望により、2~6の処理については、スクリプトを提供します。
CLKの設定やダウンロード先のMEMCの設定はできるだけ高速に設定してください。
遅い設定の場合、性能が十分に発揮されない場合があります。

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、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 1000

1000Hからバイト(8-bit)でリードします。

inh 1000

1000Hからハーフワード(16-bit)でリードします。

inw 1000

1000H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]

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 100000 100
```

100000h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 10000 100  
10000h番地から100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 100000 100  
No Memory Cache Area  
No. Address Length  
1 00100000 00000100
```

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 100000 100
No Memory Cache Area
No. Address Length
1 00100000 00000100
```

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

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 10000 20000  
10000h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 100000 20000  
Num Address Length  
01 00100000 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]

nsbpd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 00100000 00200000  
02 00400000 00010000
```

```
>nsbpd 1  
Num Address Length  
01 00400000 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-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。

[使用例]

```
nrom 0 20000
```

0h番地から20000バイトを強制ユーザ領域に指定します。

```
>nrom 0 20000
No. Address  Length
  1 00000000 00020000
```

```
>nrom 10000 40000
No. Address  Length
  1 00000000 00020000
  2 00100000 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 100000 40000  
No. Address  Length  
 1 00000000 00020000  
 2 00100000 00040000
```

```
>nromd 1  
No. Address  Length  
 1 00100000 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 1000 12
```

1000Hへバイトデータ : 12hを1ライトします。

```
outh 1000 1234
```

1000Hへハーフワードデータ : 1234hをライトします。

```
outw 1000 12345678
```

1000Hへワードデータ : 12345678h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。

rom1..rom4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]

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 100000 300000 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		ライトイネーブル
			バス幅	Bit数	
#3	100000 – 3ffffff	16-bit	16-bit	32M-Bit	禁止

>rom2 140000 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	140000 – 17ffff	16-bit	16-bit	2M-Bit	許可

>rom1 0 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	000000 – 07ffff	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（GPU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	

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。

vectorbpコマンド

[書式]

```
vectorbp [[!]reset] [[!]undef] [[!]soft_int] [[!]prefetch] [[!]data]  
        [[!]irq] [[!]fiq]
```

[パラメータ]

[[!]reset:	リセット例外の発生でブレークします。!で解除します。
[[!]undef:	未定義命令例外の発生でブレークします。!で解除します。
[[!]soft_int:	ソフトウェア割込み例外の発生でブレークします。!で解除します。
[[!]prefetch:	プリフェッチアボート例外の発生でブレークします。!で解除します。
[[!]data:	データアボート例外の発生でブレークします。!で解除します。
[[!]irq:	IRQ割込み例外の発生でブレークします。!で解除します。
[[!]fiq:	FIQ割込み例外の発生でブレークします。!で解除します。

[機能]

vectorbpコマンドは例外発生時のベクタへの分岐でブレークする条件を設定するコマンドです。

[入力例]

```
vectorbp undef  
        未定義命令例外の発生でブレークします。  
vectorbp !undef  
        未定義命令例外の発生でのブレークを解除します。
```


e v t コマンド

[書式]

```
evt [EVENT_NAME [[!]always|only|and|or] [rsca A_RESOURCE] [rscb B_RESOURCE]
```

[パラメータ]

EVENT_NAME: イベント名を指定します。

trctrig: トレーストリガ条件

trcenable: トレースイネーブル条件

viewdata: ビューデータ条件

count{1|2|3|4}: カウンタのデクリメント条件

reload{1|2|3|4}: カウンタのリロード条件

seq{1to2|1to3|2to1|2to3|3to1|3to2}: シーケンサ条件

extout{1|2|3|4}: 外部出力(External Output)条件

[[!]always|only|and|or]: リソースの組み合わせを指定します。

always: イベントを常に有効とします。!で条件に従います。

only: A_RESOURCE条件だけでイベント成立とします。

and: A_RESOURCEとB_RESOURCEのAND条件をイベント成立とします。

or: A_RESOURCEとB_RESOURCEのOR条件をイベント成立とします。

rsca: A_RESOURCEの指定の直前に必要です。

rscb: B_RESOURCEの指定の直前に必要です。

A_RESOURCE, B_RESOURCE: リソースの指定です。B_RESOURCEだけに!条件を付けることはできません。

[[!]rscsngl{1..16}: シングルコンパレータのリソースを指定します。!で否定条件です。

[[!]rscrng{1..8}: レンジコンパレータのリソースを指定します。!で否定条件です。

[[!]rscnt{1|2|3|4}: カウンタを指定します。カウンタ値が条件成立。!で否定条件です。

[[!]rscseq{1|2|3}: シーケンサのステートを指定します。!で否定条件です。

[[!]rscmmd{1..16}: メモリマップデコーダを指定します。!で否定条件です。

[[!]rscexti{1|2|3|4}: 外部入力(External Input)を指定します。!で否定条件です。

[[!]rscoid{1|2|3}: コンテキストID・リソースを指定します。!で否定条件です。

[[!]trcstartstop: トレースのスタート/ストップ条件をしています。!で否定条件です。

[[!]rscexextisel{1|2|3|4}: 拡張外部入力セクタを指定します。!で否定条件です。

[機能]

トレースをコントロールするためにイベントに対し、そのリソースを指定するコマンドです。

[使用例]

```
evt trctrig only rsca rscsngl1
```

トレーストリガとしてシングルコンパレータのリソース 1 を指定します。

[備考]

設定できる各リソースの最大数は、rscstatコマンドで確認してください。

ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

r s c c i d [1 . . 3] , r s c c i d _ m a s k コマンド

[書式]

```
rscid{1|2|3} [CONTEXTID]
```

```
rscid_mask [MASK]
```

[パラメータ]

CONTEXTID: コンテキストIDを16進数(32-bit)で指定します。

MASK: コンテキストIDのビットマスク値を16進数(32-bit)で指定します。
1のビットをマスク(無視)します。

[機能]

コンテキストID・リソースに値を指定します。

[使用例]

```
rscid1 12345678
```

コンテキストID・リソース#1に0x12345678を設定します。

```
Rscid_mask ffff0000
```

コンテキストID・リソースの上位16-bitをマスク(無視)します。

[備考]

設定できるリソースの最大数は、rscstatコマンドで確認してください。

r s c c n t [1 . . 4] コマンド

[書式]

```
rscnt{1|2|3|4} [COUNT]
```

[パラメータ]

COUNT: カウンタの初期値を指定します。(0x1 - 0xFFFF)

[機能]

カウンタの初期値を設定するためのコマンドです。

[使用例]

```
rscnt1 1000
```

カウンタ#1に0x1000を設定します。

[備考]

設定できるリソースの最大数は、rscstatコマンドで確認してください。

rscexextisel [1.. 4] コマンド

[書式]

```
rscexextisel {1|2|3|4} [exextiNN]
```

[パラメータ]

exextiNN: 拡張外部入力信号の番号 (NN=0..255) を指定します。

[機能]

拡張外部入力信号セレクタ・リソースを指定します。

[使用例]

```
rscexextisel1 exexti55]
```

拡張外部入力信号セレクタ・リソース#1に拡張外部入力信号#55を指定します。

[備考]

設定できるリソースの最大数は、rscstatコマンドで確認してください。

r s c s n g l [1 . . 1 6] コマンド

[書式]

```
rscsngl {1..16} [ADDR] [amask {0|1|3}] [fetch|exec|exec_e|exec_n|accs|read|write]
        [{nodata | {deq|dneq} DATA [DATA_MASK]]]
        [rscid{1|2|3}|norscid] [[!]exact
```

[パラメータ]

[ADDR] [amask {0|1|3}]: 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 {0|1|3}]: 下位2-bitのアドレスのマスクを指定します。
0-マスクなし、1-A0をマスク、3-A[1..0]をマスク

fetch|exec|exec_e|exec_n|accs|read|write: サイクル条件の指定

fetch: 命令フェッチを指定します。

exec: 命令の実行を指定します。

exec_e: 条件が成立した実行を指定します。

exec_n: 条件が成立しなかった実行を指定します。

accs: readとwriteの両方のサイクルを指定します。

read: readサイクルを指定します。

write: writeサイクルを指定します。

nodata | {deq|dneq} DATA [DATA_MASK]: データ条件の指定

nodata: データ条件は無視します。

deq: データの一致条件を指定します。

dneq: データの不一致条件を指定します。

DATA: データを32-bit、6進数で指定します。

DMASK: データのマスクデータを32-bit、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。

rscid{1|2|3}|norscid: コンテキストIDの指定

rscid{1|2|3}: コンテキストID・リソースを指定します。

norscid: コンテキストID・リソースは指定しません。

[[!]exact: exact機能の指定をします。!を付けると指定しません。

[機能]

シングルコンパレータとレンジコンパレータの各リソースの条件を指定するためのコマンドです。

レンジコンパレータは以下のシングルコンパレータの組み合わせです。

rscrng1 – rscsngl1, rscsingl2

rscrng2 – rscsngl3, rscsingl4

rscrng3 – rscsngl5, rscsingl6

rscrng4 – rscsngl7, rscsingl8

[使用例]

```
rscsngl1 1000 amask3 write deq 12345678 0
    シングルコンパレータ#1に対し、以下を設定します。
    0x10000番地に対する0x12345678の書込み
rscsngl2 100 amask0 exec
    シングルコンパレータ#2に対し、以下を設定します。
    0x100番地の命令実行
```

[備考]

データ条件が設定できるのは、奇数のコンパレータだけです。

設定できるリソースの最大数は、rscstatコマンドで確認してください。

r s c s t a tコマンド

[書式]

rscstat

[パラメータ]

なし

[機能]

CPUに実装されているETMのリソース数を表示します。

[使用例]

以下は、OMAP2420の場合です。

```
>rscstat
Number of Resources:
Address Comparator (rscsngl) : 8
Data Comparator   : 2
Counter           (rsccnt)  : 2
Sequencer         (rscseq)  : 1
ContextID Comparator (rscid) : 1
Memory Map Decoder (rscmmd) : 0
External Input     (rscexti) : 0
External Output    (rscext0) : 0
Extended External Input Selector
(rsceextisel) : 0
Extended External Input
(rsceexti) : 0
```

trcenableコマンド

[書式]

```
trcenable [include|exclude] [rscsngl_{[1]..[g]}] [rscrng_{[1]..[4]}]
          [rscmmd_{[1]..[g]}]
          [[!]always] [start rscsngl_{[1]..[g]}] [stop rscsngl_{[1]..[g]}]
```

[パラメータ]

include|exclude: 一致条件のモードの指定
 include: リソース条件が一致した状態をトレースイネーブルとして指定。
 exclude: リソース条件が一致しなかった状態をトレースイネーブルとして指定。
 rscsngl_{[1]..[g]}: 一致条件としてシングルコンパレータのリソースを指定します。
 rscrng_{[1]..[4]}: 一致条件としてレンジコンパレータのリソースを指定します。
 rscmmd_{[1]..[g]}: 一致条件としてメモリマップデコーダを指定します。
 [!]always [start rscsngl_{[1]..[g]}] [stop rscsngl_{[1]..[g]}]: 区間条件の指定
 [!]always: 区間トレース条件を常に有効として指定します。!で区間条件のリソースが有効になります。
 start rscsngl_{[1]..[g]}: 区間トレース条件の開始条件として、シングルコンパレータのリソースを指定します。
 stop rscsngl_{[1]..[g]}: 区間トレース条件の停止条件として、シングルコンパレータのリソースを指定します。

[機能]

トレースを取り込む条件を指定するためのコマンドです。イベント条件(evtコマンド)とtrcenableコマンドの一致条件と区間条件の全てが成立している時に行います。したがって、片方の条件だけを指定する場合は、他方は常に成立の状態にしておく必要があります。(一致条件では、excludeを、区間条件では、alwaysを使います。)

[使用例]

```
evt trcenable always
trcenable include rscsngl_12 rscrng_2 always
          シングルコンパレータ#1と#2、レンジコンパレータ#2に一致したサイクルをトレースイネーブルとして指定します。区間はalwaysとすることで無条件に成立状態にします。
trcenable exclude rscsngl_1 rscrng_2 start rscsngl_1 stop rscsngl_2
          シングルコンパレータ#1を開始条件、#2を停止条件を区間トレース条件としてトレースイネーブルに指定します。イベントの一致条件は指定リソースを削除し、excludeを指定することで無条件に成立状態にします。
```

[備考]

リソースに設定できる最大数は、rscstatコマンドで確認してください。
 メモリマップデコーダは、CPUの仕様に依存します。
 ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

viewdataコマンド

[書式]

```
viewdata [!include | include [rscsngl_{[1]..[g]}] [rscrng_{[1]..[4]}]
[rscmmd_{[1]..[g]}]]
[exclude [rscsngl_{[1]..[g]}] [rscrng_{[1]..[4]}] [rscmmd_{[1]..[g]}]]
```

[パラメータ]

```
!include | include [rscsngl_{[1]..[g]}] [rscrng_{[1]..[4]}]: 一致条件の指定
!include:          一致条件としてリソース条件を無視します。
include:           一致条件としてリソース条件を指定する時に必要です。
rscsngl_{[1]..[g]}: シングルコンパレータのリソースを指定します。
rscrng_{[1]..[4]}:   レンジコンパレータのリソースを指定します。
rscmmd_{[1]..[g]}:   メモリマップデコーダを指定します。
[exclude [rscsngl_{[1]..[g]}] [rscrng_{[1]..[4]}] [rscmmd_{[1]..[g]}]: 不一致条件の指定
exclude:          不一致条件のリソース指定時に必要です。
rscsngl_{[1]..[g]}: シングルコンパレータのリソースを指定します。
rscrng_{[1]..[4]}:   レンジコンパレータのリソースを指定します。
rscmmd_{[1]..[g]}:   メモリマップデコーダを指定します。
```

[機能]

アクセスサイクルのデータをトレースに取り込む条件を指定するためのコマンドです。一致条件と不一致条件がそれぞれ指定でき、両方の条件が成立した状態で取り込みます。

[使用例]

```
viewdata include rscsngl_12 always
          シングルコンパレータ#1と#2に一致したサイクルをviewdataとして指定します。
```

[備考]

リソースに設定できる最大数は、rscstatコマンドで確認してください。
メモリマップデコーダは、CPUの仕様に依存します。
ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

f i f o f u l l コマンド

[書式]

```
fifofull [include|exclude] [rscrng_{[1].. [4]}] [rscmmd_{[1].. [g]}]]
```

[パラメータ]

include|exclude: リソースの一致／不一致の条件指定

include: リソース条件が一致した状態をfifofullとして指定。

exclude: リソース条件が一致していない状態をfifofullとして指定。

rscrng_{[1].. [4]}] [rscmmd_{[1].. [g]}]:

rscsngl_{[1].. [g]}: シングルコンパレータのリソースを指定します。

rscrng_{[1].. [4]}: レンジコンパレータのリソースを指定します。

rscmmd_{[1].. [g]}: メモリマップデコーダを指定します。

[機能]

完全モード(tronコマンド!realパラメータ)でトレースを取り込む条件を指定します。

[使用例]

```
fifofull include rscrng_2
```

レンジコンパレータ#2に一致したサイクルをfifofullとして指定します。

[備考]

リソースに設定できる最大数は、rscstatコマンドで確認してください。これはCPUに組み込まれたETMの仕様に依存します。

メモリマップデコーダは、CPUの仕様に依存します。

ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

t e n vコマンド

[書式]

```
tenv [etm|etb] [td_none|td_addr|td_data|td_both] [tclkdiv{1|2|4|6|8}]
      [cid_none|cid_{8|16|32}bit] [cp_none|cp_always|cp_viewdata] [syncNN]
      [fifoNN] [mmdNN]
```

[パラメータ]

etm|etb: トレース資源としてETMとETBを指定します。

EtM: etmはICE上のトレースメモリに記録した情報を元にトレース表示を行います。

Etb: etmはCPU内のトレースメモリに記録した情報を元にトレース表示を行います。

td_none|td_addr|td_data|td_both: データトレースの種類を指定

td_none: データトレースなしを指定します。

td_addr: アドレスのみを指定します。

td_data: データのみを指定します。

td_both: アドレスとデータの両方を指定します。

tclkdiv{1|2|4|6|8}: 初期値でご使用ください。

cid_none|cid_{8|16|32}bit: コンテキストIDのトレース幅を指定します。

cid_none: コンテキストIDはトレースしません。

cid_{8|16|32}bit: それぞれ8bit, 16bit, 32bitでトレースします。

cp_none|cp_always|cp_viewdata: コプロセッサレジスタのトレースの有無を指定します。

cp_none: コプロセッサレジスタをトレースしません。

cp_always: コプロセッサレジスタを常にトレースします。

cp_viewdata: コプロセッサレジスタをViewDataが有効の時だけトレースします。

syncNN: 初期値でご使用ください。

fifoNN: 初期値でご使用ください。

mmdNN: メモリマップデコーダの制御レジスタの設定です。(0 - 255)

[機能]

トレースの環境を表示と設定を行うコマンドです。

[使用例]

```
tenv
```

設定状態の表示です。以下は、起動直後の初期値の表示例です。

```
>tenv
```

```
Trace Env Settings :
```

```
Trace Mode           = ETM (etm)
Data Trace           = Address and Data (td_both)
TRACECLK Div.       = 1/4 (tclkdiv4)
Context ID           = Not Trace (cid_none)
Copro. Register Transfer = Not Trace (cp_none)
Synchronization Period = 1024 (sync1024)
FIFOFULL Min. Remain Level = 1 (fifo1)
MMD Control Register Value = 0 (mmd0)
TRACEPKT Bit Width   = 8 bit (tdwidth8)
```

```
tenv td_none
```

データトレースを行いません。

[備考]

メモリマップデコーダの制御レジスタは、CPUの仕様に依存します。

ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

t m o d e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確認するためのコマンドです。

[使用例]

以下は、起動直後の初期値の表示例です。

```
>tmode
Trace Settings:
  Delay Count      = 01ffff
  Delay Mode       = Disable (!delay)
  Trace Mode       = Real Time (real)
  Ext Trigger      = Disable (noext)
Trace Env Settings :
  Data Trace              = Address and Data (td_both)
  TRACECLK Div.          = 1/1 (tclkdiv1)
  FIFOFULL Min. Remain Level = 1 (fifo1)
  MMD Control Register Value = 0 (mmd0)
  TRACEPKT Bit Width     = 8 bit (tdwidth8)
  Trace Out Context ID Bits = Disable (cid0)
Event Settings:
  evt trctrig !always
  evt trcenable always
  evt viewdata always
Trace Enable Settings:
Include/Exclude Regions:
  Mode                : Exclude (exclude)
  Single Comparator   : nothing
                      (rscsngl_)
  Range Comparator    : nothing
                      (rscrng_)
  Memory Map Decoder  : nothing
                      (rscmmd_)
Trace Start/Stop Resource:
  Mode                : Always Start (always)
  Start Single Comp.  : nothing
                      (start rscsngl_)
  Stop Single Comp.   : nothing
                      (stop rscsngl_)
View Data Settings:
Include Regions:
  Mode                : Include region Ignore (!include)
  Single Comparator   : nothing
                      (rscsngl_)
  Range Comparator    : nothing
                      (rscrng_)
  Memory Map Decoder  : nothing
                      (rscmmd_)
Exclude Regions:
```

```
Single Comparator      : nothing
                        (rscsngl_)
Range Comparator       : nothing
                        (rscrng_)
Memory Map Decoder     : nothing
                        (rscmmd_)
FIFO Full Settings:
Mode                   : Include (include)
Range Comparator       : nothing
                        (rscrng_)
Memory Map Decoder     : nothing
                        (rscmmd_)
Option Break 2 Settings :
Trace Trigger Break   : Disable (!trctr)
```

[備考]

ETMとトレースの関係は、付録 A トレース機能の詳細を参照ください。

tronコマンド

[書式]

```
tron [DELAY] [[!]delay] [[!]real|real2] [noext|posi|nega]
```

[パラメータ]

DELAY = 0..xxxx: 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16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、トレース開始後、デ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。

[[!]real|real2: トレースの取り込みモードを指定します。

[[!]real: トレース中の実行モードを指定します。realでリアルタイム実行モードです。リアルタイム実行モードでは、トレース情報がオーバーフローする場合があります。!で完全モード(非リアルタイム実行モード)になります。このモードでは、トレースのオーバーフローは発生しませんが、実行速度が低下します。

Real2: リアルタイム実行モードですが、オーバーフローするとデータトレースを優先的に排除し、実行状態のトレースをできるだけトレースするモードです。

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

noext: EXIOをトリガとして使用しません。

posi: EXIOの立ち上がりエッジをトリガとして指定します。

nega: EXIO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
```

初期値でtronした場合、トレースは強制的に開始し、トレースを強制的に終了するまでトレースします。ブレーク後trace表示させた場合、ブレーク直前の実行までの実行状態が表示できます。

```
tron delay 3fffd
```

初期値に対し強制デレイモード(delay=on)でトレースを開始します。実行開始直後より、デレイカウンタ値:0x3fffd分の取り込み後、トレースは自動的に終了します。強制デレイモードでは、トリガは無視されます。

```
rscsngl1 1000 amask0 exec
```

```
evt trctrng onlya rsca rscsngl1
```

```
tron !delay 1ffff
```

0x1000番地の実行をトリガポイントとしてトレースを開始します。!delay, 1ffffは、変更していなければ指定する必要はありません。トリガ成立後は、デレイカウンタ値:0x1ffffサイクル分取り込んだ後、トレースは自動的に終了します。その結果、トリガポイントの前後、約0x20000サイクルがトレースに入ります。

```
rscsngl1 1000
```

```
rscsngl2 2000
```

```
troenable !always start rscsngl_1 stop rscsngl_2
```

```
tron
```

0x1000番地の実行直後より、トレースの取り込みを開始し、0x2000番地の実行で一時的にトレースの取り込みを中止します。

[備考]

トレースに関する詳細は、「付録. A トレース機能の詳細」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|ttag1|ttag2]
```

[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

[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表示します。

[表示例]

```
> trace -8 pc asm
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
-000008	----	01000a48	e321f050	MSR cpsr_ctl, #50	0000	IE
* -000001	----	01000a50	eafffecc	B 1000588	0000	IE
* +000000	----	-----	-----	[Trigger]	0000	TRIG
+000001	----	01000588	e92d4008	STMDB r13!, {r3, r14}	0000	IE

```
> trace -8 all asm
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
-000008	----	01000a48	e321f050	MSR cpsr_ctl, #50	0000	IE
-000008	0001	01000a4c	e59fd008	LDR r13, 1000A5C		IE
-000007	----	01000a5c	01007000	[data access]	0000	DATA(0)
* -000001	----	01000a50	eafffecc	B 1000588	0000	IE
* +000000	----	-----	-----	[Trigger]	0000	TRIG
+000001	----	01000588	e92d4008	STMDB r13!, {r3, r14}	0000	IE
+000002	----	01006ff8	-----2c	[data access]	0000	DATA(0)
+000007	----	-----	-----	[data access]	0000	DATA(0)

```
> trace -8 ttag1
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
-000008	----	01000a48	e321f050	MSR cpsr_ctl, #50	0000	IE
				time = 000,000,000,084.8uS		
-000008	0001	01000a4c	e59fd008	LDR r13, 1000A5C		IE
-000007	----	01000a5c	01007000	[data access]	0000	DATA(0)
				time = 000,000,000,084.8uS		
* -000001	----	01000a50	eafffecc	B 1000588	0000	IE
				time = 000,000,000,086.1uS		
* +000000	----	-----	-----	[Trigger]	0000	TRIG
				time = 000,000,000,086.1uS		
+000001	----	01000588	e92d4008	STMDB r13!, {r3, r14}	0000	IE
				time = 000,000,000,086.5uS		
+000002	----	01006ff8	-----2c	[data access]	0000	DATA(0)
				time = 000,000,000,086.5uS		

```

> trace -8 ttag2
  Cycle Sub  Address Code      Instruction          EXT Stat
-000008 ----  01000a48 e321f050 MSR      cpsr_ctl, #50      0000 IE
                                time = 000,000,000,011.0uS
-000008 0001  01000a4c e59fd008 LDR      r13, 1000A5C      IE
-000007 ----  01000a5c 01007000 [data access]      0000 DATA(0)
                                time = 000,000,000,011.0uS
* -000001 ----  01000a50 eafffecc B        1000588            0000 IE
                                time = 000,000,000,000.3uS
* +000000 ----  ----- [Trigger]          0000 TRIG
                                time = 000,000,000,000.3uS
+000001 ----  01000588 e92d4008 STMDB  r13!, {r3, r14}  0000 IE
                                time = 000,000,000,000.4uS
+000002 ----  01006ff8 -----2c [data access]      0000 DATA(0)
                                time = 000,000,000,000.4uS

```

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: 命令実行以外のサイクルです。

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。

: トリガポイント (±10サイクル程度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。

[備考]

トレースに関する詳細は、「付録. A トレース機能の詳細」を参照ください。

f t r a c eコマンド

[書式]

```
ftrace statpos endpos filename [trace_options]
```

[パラメータ]

statpos: ファイルに書き出すトレースポジションの開始位置

endpos: ファイルに書き出すトレースポジションの終了位置

filename:

trace_options: 以下のパラメータが指定できます。意味は、traceコマンドと同じです。

[all|pc] [asm|ttag1|ttag2]

[機能]

トレースバッファの内容をファイルに書き出します。

[注意]

このコマンドは、処理を開始しますと途中でキャンセルできませんので、パラメータの入力には十分ご注意ください。大きな範囲を指定した場合、処理に時間がかかります。

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。(初期値)
large: 最大の補正をします。

[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。

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。