付録. Ａ　ＫＩＴ－ＡＳ８５ＥＰ２－ＴＰ（－Ｈ）　内部コマンド

本書は、ＫＩＴ－ＡＳ８５ＥＰ２－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 A-1
コマンド書式 A-2
アクセス系ブレークポイント　：ＡＢＰ，ＡＢＰ１，ＡＢＰ２コマンド A-3
環境設定　：ＥＮＶ，ＥＭＥＭＳＴＡＴコマンド A-3
アクセス系イベントの設定　：ＥＶＡコマンド A-6
実行系イベントの設定　：ＥＶＥコマンド A-7
イベント統合の設定　：ＥＶＴコマンド A-8
ヘルプ　：ＨＥＬＰコマンド A-9
ＩＮＰＵＴ　：ＩＮＢ，ＩＮＨ，ＩＮＷコマンド A-10
初期化　：ＩＮＩＴコマンド A-11
ＪＴＡＧリード　：ＪＲＥＡＤコマンド A-12
デバッガキャッシュ領域の解除　：ＮＣコマンド A-13
デバッガキャッシュ領域の設定　：ＮＣＤコマンド A-14
ソフトブレーク禁止領域の設定　：ＮＳＢＰコマンド A-15
ソフトブレーク禁止領域の解除　：ＮＳＢＰＤコマンド A-16
強制ユーザ領域の設定　：ＮＲＯＭコマンド A-17
強制ユーザ領域の解除　：ＮＲＯＭＤコマンド A-18
ＯＵＴＰＵＴ　：ＯＵＴＢ，ＯＵＴＨ，ＯＵＴＷコマンド A-19
ＣＰＵリセット　：ＲＥＳＥＴコマンド A-20
Ｅ．ＲＯＭの設定　：ＲＯＭコマンド(RTE-1000-TP 用コマンド) A-21
Ｅ．ＲＯＭの設定　：ＲＯＭ1..ＲＯＭ４コマンド(RTE-2000(H)-TP 用コマンド) A-22
シーケンシャル条件の設定　：ＳＥＱコマンド A-24
ＳＦＲアクセス　：ＳＦＲコマンド A-25
シンボル　：ＳＹＭＦＩＬＥ，ＳＹＭコマンド A-27
トリガポイント　：ＴＰコマンド A-28
トレーススイッチポイント　：ＴＳＰ１，ＴＳＰ２コマンド A-29
トレースデータ条件　：ＴＤ１，ＴＤ２コマンド A-30
トレースの設定＆開始　：ＴＲＯＮコマンド A-31
トレースの強制終了　：ＴＲＯＦＦコマンド A-34
トレースの表示　：ＴＲＡＣＥコマンド A-35
トレースデータのディレイ調整　：ＴＤＡＴＡ＿ＤＬＹコマンド A-38
バージョン表示　：ＶＥＲコマンド A-39

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。
eva/eve/evt/seq の各コマンドは、Rte4win32 V5.10 以上で対応されたコマンドです。

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。）

ａｂｐ，ａｂｐ１，ａｂｐ２コマンド

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

[パラメータ]
```
abp [or|and|seq]: abp1とabp2の組み合わせの条件を指定します。
     or:          abp1 又は、abp2のどちらかの発生でブレークします。
     and:         abp1とabp2が同時に発生した時にブレークします。マスク条件を使用します。
     seq:         abp1発生後、abp2が発生した時にブレークします。
abp[1|2]: abp1または、abp2の条件指定に先立ち入力します。
  ADDR [AMASK]:   アドレス条件の指定
     ADDR:        アドレスを１６進数で指定します。
     AMASK:       アドレスのマスクデータを16進数で指定します。1のビットは、比較の対象に
                  なりません。
data DATA [DMASK]: データ条件の指定
     DATA:        データを16進数で指定します。
     DMASK:       データのマスクデータを16進数で指定します。1のビットは、比較の対象にな
                  りません。
asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。
aeq|aneq: アドレスの比較条件を指定します。
     aeq:         アドレスをイコールで比較します。
     aneq:        アドレスをノットイコールで比較します。
deq|dneq: データの比較条件を指定します。
     deq:         データをイコールで比較します。
     dneq:        データをノットイコールで比較します。
exec|read|write|accs: サイクルの条件を指定します。
     exec:        実行アドレスを指定します。データ条件は無視されます。
     read:        リードサイクルを指定します。
     write:       ライトサイクルを指定します。
     accs:        リードまたはライトサイクルを指定します。
byte|hword|word|nosize: アクセスサイズの指定します。
     byte:        バイトアクセス(8-bit)を指定します。
     hword:       ハーフワードアクセス(16-bit)を指定します。
     word:        ワードアクセス(32-bit) を指定します。
     nosize:      無効を指定します。
abp{1|2} /del: 条件の解除を行います。
     /del:        解除を指定します。
```

[機能]
```
２点ある、アクセス系のブレークポイントの設定または解除します。
実行アドレスの指定もできます。
```

[入力例]
```
abp or
     abp1 or abp2 を指定します。
abp1 1000 aeq exec
     1000h番地の実行にブレークを設定します。
abp2 1000 data 5555 0 aeq deq read hword
     1000h番地からhwordで5555hをリードした時にブレークします。
abp1 /del
     abp1の条件を解除します。
```

ｅｎｖ，ｅｍｅｍｓｔａｔコマンド

[書式]

```
env  [[!]auto] [[!][verify]] [[!]reset] [[!]stopz] [[!]hldrq]
     [jtag[xxx][.[yyy]]{M|K}] [[!]nmi0] [[!]nmi1] [[!]nmi2]
     [rtrcb{0|25|50|75}] [nrtrcb{12|25|37|50}] [[!]iiram_chk]
```

[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。

[!]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。

[!]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。

> 注意：初期値はRte4win32のバージョンによって異なります。
> V5.09.xxまではマスクしない、V5.10.xx以上はマスク状態です。
> マスクを解除して使用する場合､CPUのマニュアル記載の注意事項を必ず参照ください。

[!]stopz: 当該KITでは使用しません。!stopの状態から変更しないでください。

[!]hldrq: HOLDRQ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。

[!]nmi0 : 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。

[!]nmi1 : 当該KITでは使用しません。!nmi1の状態から変更しないでください。

[!]nmi2 : 当該KITでは使用しません。!nmi2の状態から変更しないでください。

[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。

rtrcb{0|25|50|75}: リアルタイムトレース中のオーバフローからの復帰時のﾊﾞｯﾌｧの使用率を指定します。通常、初期値で使用してください。

nrtrcb{12|25|37|50}: 完全トレースモード中のパイプライン停止要求時にﾊﾞｯﾌｧの使用率を指定します。通常は、初期値で使用してください。

[!]iiram_chk: Instruction RAM からプログラム実行を開始する場合、Instruction RAM がリードモードになっていることをチェックするかどうかの指定です。
iiram_chkで有効、!iiram_chkで無効です。

[機能]

envコマンドは、エミュレーション環境の設定とDCUの状態を表示します。
設定は変更が必要なパラメータだけを入力ください。入力の順序は任意です。
但し、同じパラメータを２回入力した場合は、後から入力した値が有効です。
ememstatコマンドはRTE-2000(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-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:
 NMI0         = NO MASK (!nmi0)       <<Rte4win32 V5.10.xx以上では"MASK"です。
 NMI1         = NO MASK (!nmi1)
 NMI2         = NO MASK (!nmi2)
 RESET        = NO MASK (!reset)      <<Rte4win32 V5.10.xx以上では"MASK"が初期値です。
 HLDRQ        = NO MASK (!hldrq)
 STOPZ        = NO MASK (!stopz)
Trace Buffer Usage Settings:
 Realtime     <=  0% (rtrcb0)
 None Realtime>= 12% (nrtrcb12)
Trace UNIT:
 Cotrol Unit  = Enable
 Event Unit   = Enable
  Execute  Event Number = 8
  Access   Event Number = 4
  Sequence Event Number = 1
  Sequence Counter Bit  = 12
IIRAM Settings:
 Mode Check   = Enable  (iiram_chk)

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

[入力例]

env !nmi0 verify

RESETをマスクし、NMIをマスクしません。Verify機能をONにします。

env jtag40m

JTAGクロックを40MHzに設定します。

ｅｖａコマンド

[書式]
eva {1..8} [ADDR] [data DATA [MASK]] [asid ASID|noasid] [eq|lt|gt|ne|lte|gte|ign]
[deq|dneq][read|write|accs] [byte|hword|word|nosize] [/del]

[パラメータ]
eva {1..8}: アクセス系イベントのチャンネルを指定します。(ME2で指定できるchは1-4のみ)
ADDR: アドレスを１６進数で指定します。
data DATA [MASK]: データ条件の指定
DATA: データを16進数で指定します。
MASK: データのマスクデータを16進数で指定します。1のビットは、比較の対象になりません。
asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。
eq|lt|gt|neq|lte|gte|ign:
eq: ADDRで指定したアドレス = イベントアドレスで条件成立を指定します。
lt: ADDRで指定したアドレス > イベントアドレスで条件成立を指定します。
gt: ADDRで指定したアドレス < イベントアドレスで条件成立を指定します。
neq: ADDRで指定したアドレス != イベントアドレスで条件成立を指定します。
lte: ADDRで指定したアドレス => イベントアドレスで条件成立を指定します。
gte: ADDRで指定したアドレス =<イベントアドレスで条件成立を指定します。
ign: ADDRを比較条件として使用しない指定です。
deq|dneq: データの比較条件を指定します。
deq: データをイコールで比較します。
dneq: データをノットイコールで比較します。
read|write|accs: サイクルの条件を指定します。
read: リードサイクルを指定します。
write: ライトサイクルを指定します。
accs: リードまたはライトサイクルを指定します。
byte|hword|word|nosize: アクセスサイズの指定します。
byte: バイトアクセス(8-bit)を指定します。
hword: ハーフワードアクセス(16-bit)を指定します。
word: ワードアクセス(32-bit) を指定します。
nosize: 無効を指定します。
eva {1..8} /del:条件の解除を行います。
/del: 解除を指定します。

[機能]
アクセス系のイベントを設定します。指定したイベントは、EVTコマンドで統合して、ブレークやトレースの条件として使用できます。

[使用例]
eva 1 ffff000 data 55 00 byte read
デフォルトからの指定で、0xffff000番地から0x55のリードサイクルをEVA ch1に設定します。
eva 1 /del
EVA ch1の条件を解除します。

ｅｖｅコマンド

[書式]
eve {1..16} [ADDR] [asid ASID|noasid] [eq||t|gt|neq||te|gte|ign] [/del]

[パラメータ]
eve {1..16}: 実行系イベントのチャンネルを指定します。(ME2で指定できるchは1-8のみ)
ADDR: アドレスを１６進数で指定します。
asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。
eq||t|gt|neq||te|gte|ign:
eq: ADDRで指定したアドレス = イベントアドレスで条件成立を指定します。
lt: ADDRで指定したアドレス > イベントアドレスで条件成立を指定します。
gt: ADDRで指定したアドレス < イベントアドレスで条件成立を指定します。
neq: ADDRで指定したアドレス != イベントアドレスで条件成立を指定します。
lte: ADDRで指定したアドレス => イベントアドレスで条件成立を指定します。
gte: ADDRで指定したアドレス =<イベントアドレスで条件成立を指定します。
ign: ADDRを比較条件として使用しない指定です。
eve {1..16} /del: 条件の解除を行います。
/del: 解除を指定します。

[機能]
実行系のイベントを設定します。指定したイベントは、EVTコマンドで統合して、ブレークやトレースの条件として使用できます。

[使用例]
eve 1 1000
デフォルトからの指定で、0x1000番地の実行をEVE ch1に設定します。
eve 1 /del
EVE ch1の条件を解除します。

ｅｖｔコマンド

[書式]
evt {brk|seqclr|seq1|seq2|seq3|seq4|trcs1|trcs2|trcr|trg|match}
evep{[1][2][3]..[g]} ever{[1][3][5]..[f]} evap{[1][2][3]..[8]}
evar{[1][3][5][7]} [seq|!seq]

[パラメータ]
brk|seqclr|seq1|seq2|seq3|seq4|trcs1|trcs2|trcr|trg|match:
イベントを統合する対象を指定します。
brk: ブレーク条件を指定します。
seqclr: シーケンシャル条件のクリア条件を指定します。
seq1: シーケンシャル条件の初段の条件を指定します。
seq2: シーケンシャル条件の2段目の条件を指定します。
seq3: シーケンシャル条件の3段目の条件を指定します。
seq4: シーケンシャル条件の4段目の条件を指定します。
trcs1: TSP1(トレーススイッチポイント１)の条件を指定します。
trcs2: TSP2(トレーススイッチポイント２)の条件を指定します。
trcr: 区間トレースの条件を指定します。
trg: EVTTRG信号へのトリガ出力の条件を指定します。
match: トレーストリガの条件を指定します。
evep{[1][2][3]..[g]}: eveコマンドで指定したイベントを単独でポイントとして指定します。
数字をつけない場合、解除を意味します。(ME2で指定できるchは1-8のみ)
[1][2][3]..[8]: eveで指定したチェンネル番号と１対１で対応します。
ever{[1][3][5]..[f]}:eveコマンドで指定したイベントを複合してエリアとして指定します。
数字をつけない場合、解除を意味します。(ME2で指定できるchは1,3,5,7のみ)
1: eveで指定したチャンネル1と2の条件を範囲(and条件)として指定します。
3: eveで指定したチャンネル3と4の条件を範囲(and条件)として指定します。
5: eveで指定したチャンネル5と6の条件を範囲(and条件)として指定します。
7: eveで指定したチャンネル7と8の条件を範囲(and条件)として指定します。
evap{[1][2][3]..[8]}: evaコマンドで指定したイベントを単独でポイントとして指定します。
数字をつけない場合、解除を意味します。(ME2で指定できるchは1-4のみ)
[1][2][3]..[8]: evaで指定したチェンネル番号と１対１で対応します。
evar{[1][3][5][7]}: evaコマンドで指定したイベントを複合してエリアとして指定します。
数字をつけない場合、解除を意味します。(ME2で指定できるchは1,3のみ)
1: evaで指定したチャンネル1と2の条件を範囲(and条件)として指定します。
3: evaで指定したチャンネル3と4の条件を範囲(and条件)として指定します。
[!]seq: シーケンシャル条件を指定します。
seq: シーケンシャル条件を指定します。!でシーケンシャル条件を解除します。
seq関連(seccl r,seq1,seq2,...)の条件には、指定できません。

[機能]
eve evaで指定したイベントを何に使うかを指定します。

[使用例]
evt brk evep1234 ever5 evap12 evar3
ブレーク用のイベントとして、eveで指定した1から4をポイントして、5と6を範囲条件とし、evaで指定した1から2をポイントとして、3,4を範囲として使用します。
evt brk evep ever evap evar
ブレーク用のイベントとして指定した、evep ever evap evarを解除します。

[備考]
シーケンシャル条件の詳細は、seqコマンドを参照ください。
トレースのセクションやクォリファイに関する詳細は、本編のトレースの章を参照ください。

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、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

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。尚、ffff000h - ffffffh と 3fff000h - 3ffffffhは、内蔵のsfr領域ですので、初期値として除外エリアに指定されています。

[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
 2  0ffff000   00001000
 3  03fff000   00001000
```

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 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 100000 100
 No Memory Cache Area
 No. Address  Length
 1  00100000   00000100
 2  0ffff000   00001000
 3  03fff000   00001000

>ncd 1
 No Memory Cache Area
 No. Address  Length
 1  0ffff000   00001000
 2  03fff000   00001000
```

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]
nsbpd [ブロック番号|/all]

[パラメータ]
ブロック番号: 削除するソフトウェアブレーク禁止領域のブロック番号を指定します。
/all　　: 全てのソフトウェアブレーク禁止領域を削除します。

[機能]
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、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 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 100000 40000
No. Address    Length
 1  00000000   00020000
 2  00100000   00040000
```

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

[入力例]
rom 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]

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		ﾗｲﾄｲﾈｰﾌﾞﾙ
			バス幅	Bit数	
#3	100000 – 3fffff	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		ﾗｲﾄｲﾈｰﾌﾞﾙ
			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		ﾗｲﾄｲﾈｰﾌﾞﾙ
			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(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	

ｓｅｑコマンド

[書式]
seq [PASS] [step{1|2|3|4}]

[パラメータ]
PASS: シーケンス条件の成立回数を１０進数で指定します。
step{1|2|3|4}: シーケンスの段数を指定します。
step1: seq4->pass_count_decrement
step2: seq3->seq4->pass_count_decrement
step3: seq2->seq3->seq4->pass_count_decrement
step4: seq1->seq2->seq3->seq4->pass_count_decrement

[機能]
シーケンシャル条件の設定をします。
seq1～seq4の条件は、eve,eva,evtで指定します。
シーケンス途中でseqclr条件が成立した場合、そのシーケンスは最初に戻ります。

[使用例]
seq 100 step1
seq1->seq2->seq3->seq4の条件が100回成立した時にseqイベントが発生します。

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。

SFR (R/W):
PAL PALL PAH PAHL PAHH PDH PDHL PDHH PCS PCT PCM PCD PMAL
PMALL PMAH PMAHL PMAHH PMDH PMDHL PMDHH PMCS PMCT PMCM PMCD PMCAL
PMCALL PMCAH PMCAHL PMCAHH PMCDH PMCDHL PMCDHH PMCCS PFCCS PMCCT PFCCT
PMCCM PFCCM PMCCD PFCDH PFCDHL PFCDHH PFCALL CSC0 CSC1 BEC BHC VSWC
ICC ICCL ICCH ICD DSA0L DSA0H DDA0L DDA0H DSA1L DSA1H DDA1L DDA1H
DSA2L DSA2H DDA2L DDA2H DSA3L DSA3H DDA3L DDA3H DBC0
DBC1 DBC2 DBC3 DADC0 DADC1 DADC2 DADC3 DCHC0
DCHC1 DCHC2 DCHC3 DRST IMR0 IMR0L
IMR0H IMR1 IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L IMR3H IMR4 IMR4L
IMR4H IMR5 IMR5L IMR5H P1IC0 P1IC1 P2IC1 P2IC2 P2IC3 P2IC4 P2IC5 P5IC0
P5IC1 P5IC2 P6IC5 P6IC6 P6IC7 PDIC0 PDIC1 PDIC2 PDIC3 PDIC4 PDIC5 PDIC6
PDIC7 PDIC8 PDIC9 PDIC10 PDIC11 PDIC12 PDIC13 PDIC14 PDIC15 PLIC0 PLIC1
OVCIC0 OVCIC1 OVCIC2 OVCIC3 OVCIC4 OVCIC5 CCC0IC0 CCC0IC1 CCC1IC0 CCC1IC1
CCC2IC0 CCC2IC1 CCC3IC0 CCC3IC1 CCC4IC0 CCC4IC1 CCC5IC0 CCC5IC1 CMDIC0
CMDIC1 CMDIC2 CMDIC3 CC10IC0 CC10IC1 CM10IC0 CM10IC1 OV1IC0 UD1IC0
CC11IC0 CC11IC1 CM11IC0 CM11IC1 OV1IC1 UD1IC1 DMAIC0 DMAIC1 DMAIC2
DMAIC3 CSI3IC0 COVF3IC0 CSI3IC1 COVF3IC1 UREIC0 URIC0 UTIC0 UIFIC0
UT0IC0 UREIC1 URIC1 UTIC1 UIFIC1 UT0IC1 ADIC US0BIC US1BIC US2BIC USP2IC
USP4IC RSUMIC PSC ADM0 ADM1 ADM2
ADTS P1 P2 P5 P6 P7 PM1 PM2 PM5
PM6 PM7 PMC1 PMC2 PMC5 PMC6 PMC7
PFC1 PFC2 PFC5 PFC6 PFC7 BCT0
BCT1 DWC0 DWC1 BCC ASC BCP LBS LBC0 LBC1 FWC FIC BMC PRC SCR1
RFS1 SCR3 RFS3 SCR4 RFS4 SCR6 RFS6 CMD0
TMCD0 CMD1 TMCD1 CMD2 TMCD2 CMD3
TMCD3 TMENC10 CM100 CM101 CC100
CC101 CCR10 TUM10 TMC10 SESA10 PRM10 NCW10 TMENC11
CM110 CM111 CC110 CC111 CCR11 TUM11 TMC11 SESA11 PRM11 NCW11
CCC00 CCC01 TMCC00 TMCC01 SESC0 NCWC0
CCC10 CCC11 TMCC10 TMCC11 SESC1 NCWC1
CCC20 CCC21 TMCC20 TMCC21 SESC2 NCWC2
CCC30 CCC31 TMCC30 TMCC31 SESC3 NCWC3
CCC40 CCC41 TMCC40 TMCC41 CCC50
CCC51 TMCC50 TMCC51 OSTS IRAMM
DTFR0 DTFR1 DTFR2 DTFR3 PSMR CKC CKS
UCKC SSCGC DTOC DIFC UB0CTL0
UB0CTL2 UB0STR UB0FIC0 UB0FIC1 UB0FIC2 UB0FIC2L UB0FIC2H UB1CTL0
UB1CTL2 UB1STR UB1FIC0 UB1FIC1 UB1FIC2 UB1FIC2L UB1FIC2H PWMC0
PWM0 PWML0 PWMH0 PWMC1 PWM1 PWML1 PWMH1 INTF1
INTF2 INTF5 INTF6 INTFAL INTFDH INTFDHL INTFDHH INTR1
INTR2 INTR5 INTR6 INTRAL INTRDH INTRDHL INTRDHH CSIM30
CSIC30 SFDB30 SFDB30L SFDB30H SFA30 CSIL30 SFN30 CSIM31
CSIC31 SFDB31 SFDB31L SFDB31H SFA31 CSIL31 SFN31 UF0CS
UF0BC UF0EON UF0EONA UF0EN UF0ENM UF0SDS UF0IM0

```
    UF0IM1 UF0IM2 UF0IM3 UF0IM4 UF0IDR UF0DEND UF0GPR
    UF0MODC UF0AIFN UF0AAS UF0E1IM UF0E2IM UF0E3IM UF0E4IM UF0E7IM UF0E8IM
    UF0DSTL UF0E0SL UF0E1SL UF0E2SL
    UF0E3SL UF0E4SL UF0E7SL UF0E8SL UF0ADRS UF0CNF UF0IF0 UF0IF1 UF0IF2
    UF0IF3 UF0IF4 UF0DSCL UF0DD0 UF0DD1 UF0DD2 UF0DD3 UF0DD4 UF0DD5 UF0DD6
    UF0DD7 UF0DD8 UF0DD9 UF0DD10 UF0DD11 UF0DD12 UF0DD13 UF0DD14 UF0DD15
    UF0DD16 UF0DD17 UF0CIE0 UF0CIE1 UF0CIE2 UF0CIE3 UF0CIE4 UF0CIE5 UF0CIE6
    UF0CIE7 UF0CIE8 UF0CIE9 UF0CIE10 UF0CIE11 UF0CIE12 UF0CIE13 UF0CIE14
    UF0CIE15 UF0CIE16 UF0CIE17 UF0CIE18 UF0CIE19 UF0CIE20 UF0CIE21 UF0CIE22
    UF0CIE23 UF0CIE24 UF0CIE25 UF0CIE26 UF0CIE27 UF0CIE28 UF0CIE29 UF0CIE30
    UF0CIE31 UF0CIE32 UF0CIE33 UF0CIE34 UF0CIE35 UF0CIE36 UF0CIE37 UF0CIE38
    UF0CIE39 UF0CIE40 UF0CIE41 UF0CIE42 UF0CIE43 UF0CIE44 UF0CIE45 UF0CIE46
    UF0CIE47 UF0CIE48 UF0CIE49 UF0CIE50 UF0CIE51 UF0CIE52 UF0CIE53 UF0CIE54
    UF0CIE55 UF0CIE56 UF0CIE57 UF0CIE58 UF0CIE59 UF0CIE60 UF0CIE61 UF0CIE62
    UF0CIE63 UF0CIE64 UF0CIE65 UF0CIE66 UF0CIE67 UF0CIE68 UF0CIE69 UF0CIE70
    UF0CIE71 UF0CIE72 UF0CIE73 UF0CIE74 UF0CIE75 UF0CIE76 UF0CIE77 UF0CIE78
    UF0CIE79 UF0CIE80 UF0CIE81 UF0CIE82 UF0CIE83 UF0CIE84 UF0CIE85 UF0CIE86
    UF0CIE87 UF0CIE88 UF0CIE89 UF0CIE90 UF0CIE91 UF0CIE92 UF0CIE93 UF0CIE94
    UF0CIE95 UF0CIE96 UF0CIE97 UF0CIE98 UF0CIE99 UF0CIE100 UF0CIE101 UF0CIE102
    UF0CIE103 UF0CIE104 UF0CIE105 UF0CIE106 UF0CIE107 UF0CIE108 UF0CIE109
    UF0CIE110 UF0CIE111 UF0CIE112 UF0CIE113 UF0CIE114 UF0CIE115 UF0CIE116
    UF0CIE117 UF0CIE118 UF0CIE119 UF0CIE120 UF0CIE121 UF0CIE122 UF0CIE123
    UF0CIE124 UF0CIE125 UF0CIE126 UF0CIE127 UF0CIE128 UF0CIE129 UF0CIE130
    UF0CIE131 UF0CIE132 UF0CIE133 UF0CIE134 UF0CIE135 UF0CIE136 UF0CIE137
    UF0CIE138 UF0CIE139 UF0CIE140 UF0CIE141 UF0CIE142 UF0CIE143 UF0CIE144
    UF0CIE145 UF0CIE146 UF0CIE147 UF0CIE148 UF0CIE149 UF0CIE150 UF0CIE151
    UF0CIE152 UF0CIE153 UF0CIE154 UF0CIE155 UF0CIE156 UF0CIE157 UF0CIE158
    UF0CIE159 UF0CIE160 UF0CIE161 UF0CIE162 UF0CIE163 UF0CIE164 UF0CIE165
    UF0CIE166 UF0CIE167 UF0CIE168 UF0CIE169 UF0CIE170 UF0CIE171 UF0CIE172
    UF0CIE173 UF0CIE174 UF0CIE175 UF0CIE176 UF0CIE177 UF0CIE178 UF0CIE179
    UF0CIE180 UF0CIE181 UF0CIE182 UF0CIE183 UF0CIE184 UF0CIE185 UF0CIE186
    UF0CIE187 UF0CIE188 UF0CIE189 UF0CIE190 UF0CIE191 UF0CIE192 UF0CIE193
    UF0CIE194 UF0CIE195 UF0CIE196 UF0CIE197 UF0CIE198 UF0CIE199 UF0CIE200
    UF0CIE201 UF0CIE202 UF0CIE203 UF0CIE204 UF0CIE205 UF0CIE206 UF0CIE207
    UF0CIE208 UF0CIE209 UF0CIE210 UF0CIE211 UF0CIE212 UF0CIE213 UF0CIE214
    UF0CIE215 UF0CIE216 UF0CIE217 UF0CIE218 UF0CIE219 UF0CIE220 UF0CIE221
    UF0CIE222 UF0CIE223 UF0CIE224 UF0CIE225 UF0CIE226 UF0CIE227 UF0CIE228
    UF0CIE229 UF0CIE230 UF0CIE231 UF0CIE232 UF0CIE233 UF0CIE234 UF0CIE235
    UF0CIE236 UF0CIE237 UF0CIE238 UF0CIE239 UF0CIE240 UF0CIE241 UF0CIE242
    UF0CIE243 UF0CIE244 UF0CIE245 UF0CIE246 UF0CIE247 UF0CIE248 UF0CIE249
    UF0CIE250 UF0CIE251 UF0CIE252 UF0CIE253 UF0CIE254 UF0CIE255
  SFR (W):
    PRCMD UB0TX UB1TX UF0IC0 UF0IC1
```

[機能]

ＳＦＲレジスタ値の設定と表示を行います。

[使用例]

```
sfr PIC0
      PIC0レジスタの値を表示します。
sfr PIC0 2
      PIC0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。

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。
トレースは、トリガポイントを基点にしてその前後の実行状態を取り込むことができます。
（トリガポイントの使用方法はtronコマンドを参照ください）

[使用例]
tp 100000
　　100000h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を解除してください。

ｔｓｐ１，ｔｓｐ２コマンド

[書式]
tsp{1|2} [ADDR] [asid ASID|noasid] [/del]

[パラメータ]

tsp{1|2}:	tsp1または、tsp2の条件指定に先立ち入力します。
ADDR:	実行アドレスを１６進数で指定します。
asid ASID|noasid:	将来の拡張用です。noasidでご使用ください。
/del:	指定したアドレスを解除します。

[機能]
２点あるトレースのスイッチ・ポイント（アドレス）を指定します。
指定したスイッチ・ポイントにより、トレース情報の取り込み条件をかえることができます。
（取り込み条件の指定方法は、tronコマンドを参照ください）

[使用例]
tsp1 100000
100000hの命令実行をトレースのスイッチ・ポイントとして指定します。

[備考]
このコマンドで指定したスイッチ・ポイントは、tronコマンドを発行した時点で有効になります。

ｔｄ１，ｔｄ２コマンド

[書式]
td{1|2} [ADDR [MASK]] [asid ASID|noasid] [/del]

[パラメータ]
td{1|2}: td1または、td2の条件指定に先立ち入力します。
ADDR: アドレスを指定します。
MASK: アドレスのマスクデータを16進数で指定します。1のビットは、比較の対象になりません。
有効なのは、bit9-bit2のみです。
asid ASID|noasid: 将来の拡張用です。noasidでご使用ください。
/del: 指定したアドレスを解除します。

[機能]
トレースに取り込むデータアクセスサイクルの条件を設定します。トレースは、実行履歴情報と、ここで指定したアドレスへのアクセスサイクルを取り込みます。

[使用例]
td1 100000 ff
1000xxh番地のアクセスサイクルをトレースに取り込みます。

ｔｒｏｎコマンド

[書式]
```
tron [DELAY] [[!]delay] [[!]real] [[!]force] {[[!]evttrcs1] [[!]evttrcs2] |
     [[!]evttrcr]} [tr1_{[0]..[h]}|tr1_all] [tr2_{[0]..[h]}|tr2_all]
     [[!]clock2] [[!]stop] [noext|posi|nega] [[!]td1] [[!]td2] [[!]debug]
```

[パラメータ]

DELAY = 0..xxxx: ディレイカウンタ
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では、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。

[!]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フローする場合があります。
!で非リアルタイム実行モードになります。このモードでは、オーバフローは発生しませんが、実行速度が低下します。

[!]force: トレースの強制開始を指定します。!で強制開始を解除です。その場合は、tsp1の条件によります。

[[!]evttrcs1] [[!]evttrcs2]|[[!]evttrcr]: TSP1のイベント条件を指定します。
evttrcs1とevttrcs2のペアとevttrcrは排他的な設定となります。

- [!]evttrcs1: trcs1をTSP1の条件として使用します。!で使用しません。
- [!]evttrcs2: trcs2をTSP2の条件として使用します。!で使用しません。
- [!]evttrcr: trcrの条件成立中をTSP1、非成立中をTSP2の条件として使用します。!で使用しません。

tr1_{[0]..[h]}|tr1_all: tsp1のスイッチポイント以降に取り込むトレース情報を指定します。

- tr1_{[0]..[h]: 0:Interrupt, 1:Exception, 2:RETI, 3:JMP, 4:JR, 5:JARL,
  6:Condition Jump(not taken), 7:Condition Jump(taken),
  8:CALLT, 9:SWITCH, a:DISPOSE, b:CTRET,
  c:td1 read cycle, d:td1 write cycle,
  e:td2 read cycle, f:td2 write cycle,
  g:tp, h:evt_match
- tr1_all: 全てのトレース情報を取り込みます。

tr2_{[0]..[h]}|tr1_all: tsp2のスイッチポイント以降に取り込むトレース情報を指定します。
通常は何も指定しないことでトレースの取り込みを一時的に停止する目的で使用します。

- tr2_{[0]..[h]: 0:Interrupt, 1:Exception, 2:RETI, 3:JMP, 4:JR, 5:JARL,
  6:Condition Jump(not taken), 7:Condition Jump(taken),
  8:CALLT, 9:SWITCH, a:DISPOSE, b:CTRET,
  c:td1 read cycle, d:td1 write cycle,
  e:td2 read cycle, f:td2 write cycle,
  g:tp, h:evt_match
- tr2_all: 全てのトレース情報を取り込みます。

[!]clock2: トレースのサンプリングクロックを指定します。clock2でVBCLKの1/2を指定します。!で1/1になります。通常!clock2でご使用ください。

[!]stop: stopモード中のトレース出力を指定します。stopでstopモード中のトレースを停止します。!で停止しないの指定になります。

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。

[!]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。

[!]debug: 常時初期値 (!debug)でご使用ください。

[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tron直後より、無条件に3fffdサイクル分トレースします。

```
>tron delay 3fffd                    <<トレース開始
 Trace Settings:
 Delay Count    = 0003fffd
 Trace Mode     = Real Time (real)
 Start Mode     = Force Start (force)
 Delay Mode     = Enable (delay)
 Event trcs1    = Disable (!evttrcs1)
 Event trcs2    = Disable (!evttrcs2)
 Event trcr     = ------
 Sampling cond1= tr1_0123456789abcdefgh
 Sampling cond2= tr2_0123456789abcdefgh
 Trace Clock    = VBCLK (!clock2)
 STOP Mode      = Disable (!stop)
 Ext Trigger    = Disable (noext)
 TD1 Trigger    = Disable (!td1)
 TD2 Trigger    = Disable (!td2)
 Debug Mode     = Disable (!debug)
```

100000h番地の命令実行をトリガにして、トリガ後の取り込を1ffffサイクルでトレースを行います。

```
>tp 100000                           <<トリガの指定
 Trigger Point Settings:
    Address   AISD
 tp 00100000 noasid

>tron !delay 1ffff                   <<トレース開始
 Trace Settings:
 Delay Count    = 0001ffff
 Trace Mode     = Real Time (real)
 Start Mode     = Force Start (force)
 Delay Mode     = Disable (!delay)
 Event trcs1    = Disable (!evttrcs1)
 Event trcs2    = Disable (!evttrcs2)
 Event trcr     = ------
 Sampling cond1= tr1_0123456789abcdefgh
 Sampling cond2= tr2_0123456789abcdefgh
 Trace Clock    = VBCLK (!clock2)
 STOP Mode      = Disable (!stop)
 Ext Trigger    = Disable (noext)
 TD1 Trigger    = Disable (!td1)
 TD2 Trigger    = Disable (!td2)
 Debug Mode     = Disable (!debug)
```

tsp1をトレース開始条件、tsp2をトレース停止条件にして、100000h番地の実行から100100h番地を実行するまでの間の実行履歴をトレースします。

```
>tsp1 100000                          <<開始条件に使用するポイントの設定
 Trace Switch Point Settings:
      Address   AISD
 tsp1 00100000 noasid
 tsp2 /del

>tsp2 100100                          <<停止条件に使用するポイントの設定
 Trace Switch Point Settings:
      Address   AISD
 tsp1 00100000 noasid
 tsp2 00100100 noasid

>tron !force tr1_all tr2_             <<tsp1でall、tsp2でnoneを指定
 Trace Settings:
 Delay Count   = 0000ffff
 Trace Mode    = Real Time (real)
 Start Mode    = Start by tsp1 or evttrcs1 or evttrcr (!force)
 Delay Mode    = Disable (!delay)
 Event trcs1   = Disable (!evttrcs1)
 Event trcs2   = Disable (!evttrcs2)
 Event trcr    = ------
 Sampling cond1= tr1_0123456789abcdefgh
 Sampling cond2= tr2_
 Trace Clock   = VBCLK (!clock2)
 STOP Mode     = Disable (!stop)
 Ext Trigger   = Disable (noext)
 TD1 Trigger   = Disable (!td1)
 TD2 Trigger   = Disable (!td2)
 Debug Mode    = Disable (!debug)
```

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..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。
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
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 asm -15
   Cycle  Sub  Address     Code     Instruction              EXT  Stat
  -00001e ---- 00:0010558e ffbfb7da jarl    00100d68h        1111 JMPS  JARL
  -000014 ---- 00:00100d68 3f460000 st.b    r7,+00h[r6]      1111 JMPD  JARL
*  000000 ---- --:00100d6c 007f     jmp     [lp]             1111 MATCH
   000002 ---- 00:00105592 664003d0 movehi  03d0h,zero,r12   1111 JMPD  JMP
   000002 0001 00:00105596 672ca4b2 ld.h    -05b4eh[r12],r12 1111 SUB
   000002 0002 00:0010559a 6ecc0010 andi    0010h,r12,r13    1111 SUB
   000002 0003 00:0010559e 69e0     cmp     zero,r13         1111 SUB
   00000c ---- 00:001055a0 1d92     be      001055d2h        1111 JMPS  BcondNT
   000016 ---- 00:001055a2 16400380 movehi  0380h,zero,r2    1111 JMPD  BcondNT

>trace -15 ttag1
   Cycle  Sub  Address     Code     Instruction              EXT  Stat
  -00001e ---- 00:0010558e ffbfb7da jarl    00100d68h        1111 JMPS  JARL
                             time = 000,001,448,264.9uS
  -000014 ---- 00:00100d68 3f460000 st.b    r7,+00h[r6]      1111 JMPD  JARL
                             time = 000,001,448,265.3uS
*  000000 ---- --:00100d6c 007f     jmp     [lp]             1111 MATCH
                             time = 000,001,448,265.7uS
   000002 ---- 00:00105592 664003d0 movehi  03d0h,zero,r12   1111 JMPD  JMP
                             time = 000,001,448,267.4uS
   000002 0001 00:00105596 672ca4b2 ld.h    -05b4eh[r12],r12 1111 SUB
   000002 0002 00:0010559a 6ecc0010 andi    0010h,r12,r13    1111 SUB
   000002 0003 00:0010559e 69e0     cmp     zero,r13         1111 SUB
   00000c ---- 00:001055a0 1d92     be      001055d2h        1111 JMPS  BcondNT
                             time = 000,001,448,268.5uS
   000016 ---- 00:001055a2 16400380 movehi  0380h,zero,r2    1111 JMPD  BcondNT
                             time = 000,001,448,268.9uS
```

```
>trace -15 ttag2
   Cycle  Sub  Address     Code     Instruction              EXT  Stat
  -00001e ---- 00:0010558e ffbfb7da jarl     00100d68h        1111 JMPS  JARL
                           time = 000,000,000,002.6uS
  -000014 ---- 00:00100d68 3f460000 st.b     r7,+00h[r6]      1111 JMPD  JARL
                           time = 000,000,000,000.4uS
*  000000 ---- --:00100d6c 007f     jmp      [lp]             1111 MATCH
                           time = 000,000,000,000.2uS
   000002 ---- 00:00105592 664003d0 movehi   03d0h,zero,r12   1111 JMPD  JMP
                           time = 000,000,000,001.7uS
   000002 0001 00:00105596 672ca4b2 ld.h     -05b4eh[r12],r12 1111 SUB
   000002 0002 00:0010559a 6ecc0010 andi     0010h,r12,r13    1111 SUB
   000002 0003 00:0010559e 69e0     cmp      zero,r13         1111 SUB
   00000c ---- 00:001055a0 1d92     be       001055d2h        1111 JMPS  BcondNT
                           time = 000,000,000,001.1uS
   000016 ---- 00:001055a2 16400380 movehi   0380h,zero,r2    1111 JMPD  BcondNT
                           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: 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。

TRGSTART0	STARTパケット発生、サブスイッチがONになった
TRGSTART1	STARTパケット発生、サブスイッチがOFFになった
MATCH	MATCHパケット発生
OVF	オーバーフロー発生
TRCEND	TRCENDパケット発生
JMPD ◇	JMPDパケット発生（◇は後述）
JMPDS ◇	JMPDSパケット発生（◇は後述）
JMPS ◇	JMPSパケット発生（◇は後述）
OPCODE	オペコード・アクセス（実行）発生
DATAW1,2	メモリ書き込み発生（トレース・パケット）
DATAR1,2	メモリ読み出し発生（トレース・パケット）
SUB	サブサイクル

上記の"◇"部分は次の文字列が入ります。これは、分岐要因となった命令もしくは事象です。

NMI/INT	割り込みの発生によるもの
EXP/TRAP	例外の発生によるもの
RETI	当該命令によるもの
JMP	当該命令によるもの
JR	当該命令によるもの
JARL	当該命令によるもの
BcondNT	当該命令によるもの
Bcond	当該命令によるもの
CALLT	当該命令によるもの
SWITCH	当該命令によるもの
DISPOSE	当該命令によるもの
CTRET	当該命令によるもの
FSTART	トレースの強制スタート

*: 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。

ｔｄａｔａ＿ｄｌｙ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制御のファームウェアのバージョンを表示します。