付録. Ａ　ＫＩＴ－Ｖ８５０Ｅ／ＰＧ２－ＩＥ　内部コマンド

本書は、ＫＩＴ－Ｖ８５０Ｅ／ＰＧ２－ＩＥ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。

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。）

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。
```

[使用例]
```
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，ｉｆｒｏｍｅｎｖ，ｉｆｒｏｍｒｅｆｉｌｌコマンド

[書式]

```
env [[!]auto] [[!][verify]] [jtag[xxx][.[yyy]]{M|K}]
    [[!]reset]
ifromenv [[!]tracecache]
ifromrefill
```

[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、初期値(当該 ICEの場合は12.5MHz)でご使用ください。

[!]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。

[!]tracecache:実行中にトレース表示を行うためにIROM領域への書込みデータをキャッシングする時に指定します。!で指定しないを意味します。初期値は、キャッシングします。

[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。
ifromenvコマンドは、実行中のトレース表示の為の設定をパラメータで指定します。キャッシングを行うと指定した場合は、IROM空間へのリード、ライト時にホストPC上のメモリに同じ値がキャッシング（写像）され、実行中のトレース表示に使用されます。
ifromrefillコマンドは、手動でIROM空間の全データを強制的にキャッシングするためのコマンドです。ifromenvでtracecacheが有効であることが条件です。

envコマンドの初期値は、以下の通りです。

```
CPU Settings:
 Auto Run     = ON (auto)
 JTAGCLOCK    = 12.5MHz (jtag12)
 Verify       = verify off (!verify)
Signals Mask:
 RESET        = NO MASK (!reset)
```

[入力例]

```
env reset
    RESETをマスクします。
env verify
    Verify機能をONにします。
```

ｅｍｏｄｅコマンド

[書式]
emode

[パラメータ]
なし

[機能]
イベントの設定状態を表示します。

[表示例]
以下は、初期状態の表示です。

```
 Event Condition Settings:          << EVTコマンドの設定状態を表示
  evt brk      !seq
  evt seqclr   !seq
  evt seq1     !seq
  evt seq2     !seq
  evt seq3     !seq
  evt seq4     !seq
  evt secon    !seq
  evt secoff   !seq
  evt qualify  !seq
  evt tout     !seq
  evt match    !seq
 Event Settings (execute):<< EVEコマンドの設定状態を表示
      ch Address   ASID   Cmp
  eve  1 /del
  eve  2 /del
  eve  3 /del
  eve  4 /del
  eve  5 /del
  eve  6 /del
  eve  7 /del
  eve  8 /del
 Event Settings (access):           << EVAコマンドの設定状態を表示
      ch Address         Data    D_Mask   ASID   A_Cmp D_Cmp Kind  Size
  eva  1 /del
  eva  2 /del
  eva  3 /del
  eva  4 /del
  eva  5 /del
  eva  6 /del
 Sequence Condigion Settings:       << SEQコマンドの設定状態を表示
  seq 1 step4
```

ｅｖａコマンド

[書式]
eva {1..6} [ADDR] [data DATA [MASK]] [asid ASID|noasid] [eq|lt|gt|neq|lte|gte|ign]
[deq|dneq] [read|write|accs] [byte|hword|word|nosize] [/del]

[パラメータ]
eva {1..6}: アクセス系イベントのチャンネル(1-6)を指定します。
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..6} /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#1に設定します。
eva 1 /del
EVA#1の条件を解除します。

ｅｖｅコマンド

[書式]
eve {1..8} [ADDR] [asid ASID|noasid] [eq|lt|gt|neq|lte|gte|ign] [/del]

[パラメータ]
eve {1..8}: 実行系イベントのチャンネル(1-8)を指定します。
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|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。
eve {1..8} /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#1に設定します。
eve 1 /del
EVE#1の条件を解除します。

ｅｖｔコマンド

[書式]
```
evt {brk|seqclr|seq1|seq2|seq3|seq4|secon|secoff|qualify|tout|match}
    evep{[1][2][3]..[8]} ever{[1][3][5][7]} evap{[1][2][3]..[6]}
    evar{[1][3][5]} [[!]seq]
```

[パラメータ]
```
brk|seqclr|seq1|seq2|seq3|seq4|secon|secoff|qualify|tout|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。
    secon:      トレースのセクション"ON"の条件を指定します。
    secoff:     トレースのセクション"OFF"の条件を指定します。
    qualify:    トレースのクォリファイの条件を指定します。
    tout:       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]..[8]}: 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。
    [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][7]}: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。
    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]..[6]}: 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。
    [1][2][3]..[6]: 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]}: 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。
    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。
    5:          eva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。
 [!]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(secclr,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

ｎｃコマンド

[書式]
nc [[ADDR [LENGTH]]

[パラメータ]
ADDR: キャッシング領域の除外エリアの開始アドレスを指定します。
LENGTH: キャッシング領域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、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、最小のブロックサイズは３２バイトです。
尚、ROM,RAM領域以外は、初期値として除外エリアに指定されていますので、通常変更する必要はありません。

[表示例]
初期値の表示です。

```
>nc
 No Memory Cache Area
 No. Address  Length
 1  00100000   03ef0000
 2  0ffff000   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: 削除するキャッシング領域の除外エリアのブロック番号を指定します。

[機能]
キャッシング領域の除外エリアを削除します。削除は各キャッシング領域の除外エリアのブロック番号を指定します。初期値の領域は、削除しないでください。
変更した場合、コマンドでのIOへのアクセスで、正しい値が読み出せない場合があります。

[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。
＞＞一例ですので、実際には、変更しないでください。

```
>nc
 No Memory Cache Area
 No. Address  Length
 1  00100000   03ef0000
 2  0ffff000   00001000

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

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］
ＣＰＵをリセットします。

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の条件成立が100回成立した時にseqイベントが発生します。

ｓｓｗｏｎ，ｓｓｗｏｆｆコマンド

[書式]
```
ssw{on|off} [{exec_{[0]..[e]}|exec_default}]
            [td{1|2|3|4} {none|read|write|accs|readp|writep|accsp}]
            [evap{1|2|3|4|5|6} {none|read|write|accs|readp|writep|accsp}]
            [evar{1|3|5} {none|read|write|accs|readp|writep|accsp}]
            [all_cycle {none|read|write|accs|readp|writep|accsp}]
```

[パラメータ]

sswon: サブスイッチがON時にトレースに取り込むサイクルを指定するコマンドです。

sswoff: サブスイッチがoff時にトレースに取り込むサイクルを指定するコマンドです。

exec_{[0]..[e]}:実行系のトレースに取り込むサイクルを指定します。
番号との対応付けは、以下の通りです。取り込みを制限した場合、
トレースの逆アセンブル表示は正しく行えない場合があります。

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:tp, d:evt_match, e:opecode

exec_default: 全てのサイクルを取り込みます。('exec_0123456789abcd'と等価)
通常この状態で使用してください。

td{1|2|3|4} {none|read|write|accs|readp|writep|accsp}:
tdコマンドで指定した条件それぞれに対し、取り込むサイクルの種類を
指定します。

- none :取り込みません。
- read :リードサイクルのみを取り込みます。
- write :ライトサイクルのみを取り込みます。
- accs :リードとライトの両方のサイクルを取り込みます。
- readp :リードサイクルとその実行サイクルを取り込みます。
- writep:ライトサイクルとその実行サイクルを取り込みます。
- accsp :リードとライトのサイクルとその実行サイクルを取り込みます。

evap{1|2|3|4|5|6} {none|read|write|accs|readp|writep|accsp}:
evaコマンドで指定したポイント条件それぞれに対し、取り込むサイクル
の種類を指定します。

- none :取り込みません。
- read :リードサイクルのみを取り込みます。
- write :ライトサイクルのみを取り込みます。
- accs :リードとライトの両方のサイクルを取り込みます。
- readp :リードサイクルとその実行サイクルを取り込みます。
- writep:ライトサイクルとその実行サイクルを取り込みます。
- accsp :リードとライトのサイクルとその実行サイクルを取り込みます。

evar{1|3|5} {none|read|write|accs|readp|writep|accsp}:
evaコマンドで指定した範囲条件それぞれに対し、取り込むサイクル
の種類を指定します。

- none :取り込みません。
- read :リードサイクルのみを取り込みます。
- write :ライトサイクルのみを取り込みます。
- accs :リードとライトの両方のサイクルを取り込みます。
- readp :リードサイクルとその実行サイクルを取り込みます。
- writep:ライトサイクルとその実行サイクルを取り込みます。
- accsp :リードとライトのサイクルとその実行サイクルを取り込みます。

all_cycle {none|read|write|accs|readp|writep|accsp}:
無条件に取り込むサイクルの種類を指定します。

- none :取り込みません。
- read :リードサイクルのみを取り込みます。
- write :ライトサイクルのみを取り込みます。
- accs :リードとライトの両方のサイクルを取り込みます。
- readp :リードサイクルとその実行サイクルを取り込みます。
- writep:ライトサイクルとその実行サイクルを取り込みます。
- accsp :リードとライトのサイクルとその実行サイクルを取り込みます。

[機能]
サブスイッチの状態によって、トレースに取り込むサイクルの種類を指定します。

[使用例]
初期値では、サブスイッチがONの時に全てのサイクルを取り込み、OFFの時にサイクルの取り込みを行わないように指定してあります。
これにより、任意の条件でトレースの取り込みをコントロールできます。
以下に初期値の状態を示します。

```
>sswon
Sub-switch ON Settings:
 Trace execute cycle           = exec_0123456789abcd (exec_default)
 td1 Trace cycle   (td1)       = Read and Write cycle with pc value (accsp)
 td2 Trace cycle   (td2)       = Read and Write cycle with pc value (accsp)
 td3 Trace cycle   (td3)       = Read and Write cycle with pc value (accsp)
 td4 Trace cycle   (td4)       = Read and Write cycle with pc value (accsp)
 evap1 Trace cycle (evap1)     = No cycle (none)
 evap2 Trace cycle (evap2)     = No cycle (none)
 evap3 Trace cycle (evap3)     = No cycle (none)
 evap4 Trace cycle (evap4)     = No cycle (none)
 evap5 Trace cycle (evap5)     = No cycle (none)
 evap6 Trace cycle (evap6)     = No cycle (none)
 evar1 Trace cycle (evar1)     = No cycle (none)
 evar3 Trace cycle (evar3)     = No cycle (none)
 evar5 Trace cycle (evar5)     = No cycle (none)
 All access cycle  (all_cycle) = No cycle (none)

>sswoff
Sub-switch OFF Settings:
 Trace execute cycle           = exec_
 td1 Trace cycle   (td1)       = No cycle (none)
 td2 Trace cycle   (td2)       = No cycle (none)
 td3 Trace cycle   (td3)       = No cycle (none)
 td4 Trace cycle   (td4)       = No cycle (none)
 evap1 Trace cycle (evap1)     = No cycle (none)
 evap2 Trace cycle (evap2)     = No cycle (none)
 evap3 Trace cycle (evap3)     = No cycle (none)
 evap4 Trace cycle (evap4)     = No cycle (none)
 evap5 Trace cycle (evap5)     = No cycle (none)
 evap6 Trace cycle (evap6)     = No cycle (none)
 evar1 Trace cycle (evar1)     = No cycle (none)
 evar3 Trace cycle (evar3)     = No cycle (none)
 evar5 Trace cycle (evar5)     = No cycle (none)
 All access cycle  (all_cycle) = No cycle (none)
```

[備考]
サブスイッチに関する詳細は、「付録.Ｂ トレース機能の詳細」を参照ください。

ｓｆｒ,ｓｆｒ２コマンド

[書式]
sfr [reg [VAL]]
sfr2 [reg [VAL]]

[パラメータ]
reg: リロケータブルＳＦＲレジスタ名を指定します。
VAL: ＳＦＲのレジスタ値を１６進数で指定します。

[機能]
ＳＦＲレジスタ値の設定と表示を行います。
ＳＦＲ２コマンドはリロケータブル領域にマップされたレジスタ(CANコントローラ用レジスタ等)を対象としたコマンドで、ＳＦＲコマンドはそれ以外のレジスタを対象としたものです。
それぞれのコマンドでレジスタ名として使用できる名称は、パラメータなしでコマンドを入力することで確認できます。

[使用例]
sfr P1
P1レジスタの値を表示します。
sfr PM1 0
PM1レジスタに0hを設定します。
sfr2 C2CTRC
C2CTRLレジスタを参照します。

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。

ｔｄ１，ｔｄ２，ｔｄ３，ｔｄ４コマンド

[書式]
td{1|2|3|4} [ADDR [MASK]] [asid ASID|noasid] [/del]

[パラメータ]
td{1|2|3|4}: td1,td2,td3,td4の条件指定に先立ち入力します。
ADDR: アドレスを16進数で指定します。
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。トレースへの取り込み条件と、トリガとして使用できます。

[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。

[備考]
トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

ｔｅｎｖコマンド

[書式]
```
tenv [subor|suband] [[!]dmatrc] [[!]sfrtrc] [[!]tendbrk] [[!]stop]
     [rtrcb{0|8|16}]|[nrtrcb{0|4|8|12|16|20|24}]
     [nonbranchNN] [[!]phold] [[!]once]
```

[パラメータ]

subor:	サブスイッチとして、セクション条件とクォリファイ条件のオアを指定します。
suband:	サブスイッチとして、セクション条件とクォリファイ条件のアンドを指定します。
[!]sfrtrc:	常に!sfrtrcで使用ください。
[!]tendbrk:	トレースの終了でブレークします。!でブレークしません。
[!]stop:	ストップモード中、トレースを停止します。!で停止しません。
rtrcb{0|8|16}:	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パケット数を指定します。通常、初期値"0"でご使用ください。
nrtrcb{0|4|8|12|16|20|24}:	完全トレースモード中のパイプライン停止要求時のﾊﾞｯﾌｧの使用パケット数を指定します。通常は、初期値"0"でご使用ください。
nonbranchNN:	連続したアドレスの実行が継続した場合にPC情報をトレースに取り込む間隔を指定します。NNは、 0 – fffの範囲で指定します。 NN=0 は無限大を示します。通常は、初期値"0"でご使用ください。
[!]phold	完全モード（ノンリアルタイムモード）でトレース中、実行が停止したときにその状態を知らせるパケットをトレースに取り込みます。!で取り込みません。
[!]once:	トリガ出力としてトリガ条件成立時に１回だけ出力します。 !では、条件成立時に毎回出力します。

[機能]
トレースの環境設定を行います。

[使用例]
```
tenv subor dmatrc
```
サブスイッチは、セクションとクォリファイのオアで使用し、DMAサイクルをトレースします。

[備考]
トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

ｔｐコマンド

[書式]
tp [ADDR] [asid ASID|noasid] [/del]

[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）
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。

[使用例]
tp 100000
10000h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。
トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

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点あるトレースのセクション・ポイント（アドレス）を指定します。
指定したポイントで、トレース情報の取り込みサイクルをかえることができます。
（取り込み条件の指定は、sswon, sswoffコマンドを参照ください）

[使用例]
tsp1 100000
セクション・ポイント１に100hの命令実行を指定します。

[備考]
トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

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。

[表示例]
以下にデフォルト値を表示例として示します。

```
>tmode
Trace Settings (tron):
 Delay Count   = 0000ffff
 Trace Mode    = Real Time (real)
 Start Mode    = Force Start (force)
 Delay Mode    = Disable (!delay)
 Ext Trigger   = Disable (noext)
 TD1 Trigger   = Disable (!td1)
 TD2 Trigger   = Disable (!td2)
 TD3 Trigger   = Disable (!td3)
 TD4 Trigger   = Disable (!td4)
Trace Settings (tenv):
 Sub switch    = <section> or  <qualify> (subor)
 DMA Trace     = Enable (dmatrc)
 SFR Trace     = Disable (!sfrtrc)
 Trace End BRK = Disable (!tendbrk)
 STOP Mode     = Enable (stop)
 Non-branch    = None (nonbranch0)
 Realtime      = 0 (rtrcb0)
 No Realtime   = 0 (nrtrcb0)
 PHOLD         = Disable (!phold)
 ONCE          = Disable (!once)
 Debug Mode    = Disable (!debug)
Trace Switch Point Settings:
      Address   ASID
 tsp1 /del
 tsp2 /del
Trigger Point Settings:
    Address   ASID
 tp /del
Data Trace Settings:
     Address   A_Mask   ASID
 td1 /del
 td2 /del
 td3 /del
 td4 /del
```

[備考]
トレースに関する詳細は、本編の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] [[!]force] [noext|posi|nega]
     [[!]td{1|2|3|4}]
```

[パラメータ]
```
DELAY = 2..1fffd 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:       トレースの開始条件としてtronの最初から強制的に開始を指定します。!で強
                制開始を解除します。その場合は、tsp1の条件により開始します。強制開始
                を指定した場合もtsp1, tsp2は、有効です。
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。
[!]td3:         トレースデータ条件３(td3)をトリガとして指定します。！で解除します。
[!]td4:         トレースデータ条件４(td4)をトリガとして指定します。！で解除します。
```

[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ィレイモード(delay=on)でトレースを開始します。実行開始直後より、
    ディレイカウンタ値:0x3fffd分の取り込み後、トレースは自動的に終了します。強制ディ
    レイモードでは、トリガは無視されます。
td1 3ffb800 0
tron !delay td1 1ffff
    td1の条件成立をトリガポイントとしてトレースを開始します。!delay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、ディレイカウンタ値: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、
    各0x1ffffサイクルがトレースに入ります。
tp 1000
tron !delay 1ffff
    tpの条件成立をトリガポイントとしてトレースを開始します。!delay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、ディレイカウンタ値: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、各
    0x1ffffサイクルがトレースに入ります。
```

```
tsp1 1000
tsp2 2000
tp 1800
tron !force
```

トレースパケットの取り込み条件は、tsp1の条件成立後はsswonコマンドの指定値に、tsp2の条件成立後はsswoffコマンドの指定値になります。初期値では、sswonコマンドでパケットの取り込み、sswoffで取り込みの停止を指定していますので、この設定では、tsp1で指定した0x1000番地の実行直後より、トレースの取り込みを開始し、tsp2で指定した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。この間にtpで指定した0x1800の実行があった場合、それをトリガポイントとして、ディレイサイクル値（初期値0x1ffff）分のパケットをトレースして取り込みを終了します。

```
tsp1 /del
tsp2 /del
tron force
```

tsp1, tsp2を解除して、強制開始でトレースを開始します。

[備考]

トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

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|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

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。

[表示例]

```
> trace asm -5
   Cycle  Sub  Address     Code     Instruction              EXT  Stat
  -000006 ---- 00:00001806 4200     mov      +00h,r8         0000 JMPD  JMP
  -000006 0001 00:00001808 8f260005 ld.w     +04h[r6],r17    0000 SUB
  -000006 0002 00:0000180c 89e0     cmp      zero,r17        0000 SUB
  -000004 ---- 00:0000180e 259a     bne      00001850h       0000 JMPS  BcondNT
  -000002 ---- 00:00001810 4f26000d ld.w     +0ch[r6],r9     0000 JMPD  BcondNT
  -000002 0001 00:00001814 17860015 ld.bu    +0014h[r6],r2   0000 SUB
* +000000 ---- --:00001818 5f260011 ld.w     +010h[r6],r11   0000 MATCH
  +000001 ---- 00:0000181c 0df5     br       0000183ah       0000 JMPS  Bcond
  +000002 ---- 00:0000183a 700b     mov      r11,r14         0000 JMPD  Bcond
  +000002 0001 00:0000183c 5a5f     add      -01h,r11        0000 SUB

> trace ttag1
   Cycle  Sub  Address     Code     Instruction              EXT  Stat
  -000006 ---- 00:00001806 4200     mov      +00h,r8         0000 JMPD  JMP
                             time = 000,000,284,368.8uS
  -000006 0001 00:00001808 8f260005 ld.w     +04h[r6],r17    0000 SUB
  -000006 0002 00:0000180c 89e0     cmp      zero,r17        0000 SUB
  -000004 ---- 00:0000180e 259a     bne      00001850h       0000 JMPS  BcondNT
                             time = 000,000,284,368.9uS
  -000002 ---- 00:00001810 4f26000d ld.w     +0ch[r6],r9     0000 JMPD  BcondNT
                             time = 000,000,284,368.9uS
  -000002 0001 00:00001814 17860015 ld.bu    +0014h[r6],r2   0000 SUB
* +000000 ---- --:00001818 5f260011 ld.w     +010h[r6],r11   0000 MATCH
                             time = 000,000,284,368.9uS
  +000001 ---- 00:0000181c 0df5     br       0000183ah       0000 JMPS  Bcond
                             time = 000,000,284,368.9uS
```

```
> trace ttag2
   Cycle  Sub  Address     Code     Instruction             EXT  Stat
  -000006 ---- 00:00001806 4200     mov      +00h,r8        0000 JMPD  JMP
                             time = 000,000,000,000.0uS
  -000006 0001 00:00001808 8f260005 ld.w     +04h[r6],r17   0000 SUB
  -000006 0002 00:0000180c 89e0     cmp      zero,r17       0000 SUB
  -000004 ---- 00:0000180e 259a     bne      00001850h      0000 JMPS  BcondNT
                             time = 000,000,000,000.1uS
  -000002 ---- 00:00001810 4f26000d ld.w     +0ch[r6],r9    0000 JMPD  BcondNT
                             time = 000,000,000,000.0uS
  -000002 0001 00:00001814 17860015 ld.bu    +0014h[r6],r2  0000 SUB
* +000000 ---- --:00001818 5f260011 ld.w     +010h[r6],r11  0000 MATCH
                             time = 000,000,000,000.0uS
  +000001 ---- 00:0000181c 0df5     br       0000183ah      0000 JMPS  Bcond
                             time = 000,000,000,000.0uS
```

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。

TRGSTARTON	STARTパケット発生、サブスイッチがONになった
TRGSTARTOFF	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
DATAW	メモリ書き込み発生（トレース・パケット）
DATAR	メモリ読み出し発生（トレース・パケット）
SFRW	SFR書き込み発生（バストレース）
SFRR	SFR読み出し発生（バストレース）
SUB	サブサイクル

"◇"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
STORE	データトレースでWithPCが指定されている場合
LOAD	データトレースでWithPCが指定されている場合
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。

［備考］
envコマンドでiromcacheを有効にしている場合は、IROM領域の実行に限り実行中でもトレースの表示が可能です。
トレースに関する詳細は、本編のトレース機能の詳細を参照ください。

ｆｔｒａｃｅコマンド

[書式]
ftrace statpos endpos filname [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
filname:
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|data] [asm|ttag1|ttag2] [subNN]

[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。

ｔｉｍｅコマンド

[書式]
time

[パラメータ]
なし

[機能]
実行時間計測結果を時間で表示します。実行時間計測のタイマーはＣＰＵが実行を開始する毎に初期化され、ＣＰＵ実行中カウントされます。計測クロックは 50MHz で、20nS の分解能で ns 単位に換算されて表示されます。

[備考]
測定値は実行の開始とブレークのオーバヘッド時間（数クロック）を含みます。

[使用例]
```
>time
 Time = 6,600 (ns) (50.000000MHz) [Counter=0000014a]
                          |                  |_Counter 値(Hex)
                          |_ 計測クロックの周波数(常に 50MHz)
```

ｖｅｒコマンド

［書式］
ver

［パラメータ］
なし

［機能］
ＩＣＥ制御用ファームウェアのバージョンを表示します。