

付録 B RTE-V850E/GPx-IE 内部コマンド

本書は、RTE-V850E/GPx-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・ウィ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
アクセスサイズの指定 : ACCコマンド	A-4
環境設定 : ENVコマンド	A-5
イベント設定状態の表示 : EMODEコマンド	A-6
アクセス系イベントの設定 : EVAコマンド	A-7
実行系イベントの設定 : EVEコマンド	A-8
イベント統合の設定 : EVTコマンド	A-9
外部からのブレーク設定 : EXTBRKコマンド	A-10
フライバイ・リード : FRDコマンド	A-11
フライバイ・ライト : FWRコマンド	A-12
ヘルプ : HELPコマンド.....	A-13
INPUT : INB, INH, INWコマンド.....	A-14
初期化 : INITコマンド.....	A-15
デバッガキャッシュ領域の解除 : NCコマンド.....	A-16
デバッガキャッシュ領域の設定 : NCDコマンド.....	A-17
OUTPUT : OUTB, OUTH, OUTWコマンド.....	A-18
CPUリセット : RESETコマンド.....	A-19
シーケンシャル条件の設定 : SEQコマンド.....	A-20
サブスイッチの設定 : SSWON, SSWOFFコマンド.....	A-21
SFRアクセス : SFR, SFR2コマンド.....	A-23
シンボル : SYMFILE, SYMコマンド.....	A-27
実行時間の表示 : TIMEコマンド.....	A-28
トレースデータ条件 : TD1, TD2, TD3, TD4コマンド	A-29
トレースの環境設定 : TENVコマンド.....	A-30
トリガポイント : TPコマンド.....	A-31
トレーススイッチポイント : TSP1, TSP2コマンド.....	A-32
トレース条件の参照 : TMODEコマンド.....	A-33
トレースの設定&開始 : TRONコマンド.....	A-34
トレースの強制終了 : TROFFコマンド	A-36
トレースの表示 : TRACEコマンド.....	A-37
トレースのファイル書き出し : FTRACEコマンド	A-40
バージョン表示 : VERコマンド	A-41

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

RTE-V850E/GPx-IEの内部コマンドの基本書式を以下に示します。

コマンド名 パラメータ

*パラメータ書式で [] は省略可能を示し、| は択一を意味します。

コマンド名はアルファベットの文字列でパラメータとの間はスペースまたはタブで区切ります。パラメータはアルファベットの文字列または16進数を指定し、各パラメータ間はスペースまたはタブで区切ります。(16進数には演算子は使用できません。)

abp, abp1, abp2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。
```

accコマンド

[書式]

acc [byte|hword|word]

[パラメータ]

byte|hword|word: 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。

[機能]

フライバイ・ライトコマンド(fwr)におけるデータサイズを指定します。

[使用例]

acc byte

fwr 0 12

fwrコマンドでバイトデータを書き込む手順です。

acc word

fwr 0 12345678

fwrコマンドでワードデータを書き込む手順です。

envコマンド

[書式]

```
env [!]auto [!]verify [jtag {25|12|5|2|1|500|250|100}]
    [!]nmi0 [!]nmi1 [!]nmi2 [!]reset [!]hldrq [!]stopz [!]waitz
```

[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 {25|12|5|2|1|500|250|100}: 内部のN-WireのJTAGクロック指定します。
それぞれ以下に対応します。
[25MHz|12.5MHz|5MHz|2MHz|1MHz|500KHz|250KHz|100KHz]

備考: 通常は、初期値の25MHzでご使用ください。1MHzより低い周波数を指定した場合は、デバッグ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。

[!]nmi0, [!]nmi1, [!]nmi2: 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。

外部端子のNMIは、nmi2で指定します。

[!]reset: RESET端子のマスク指定を指定します。!はマスクしないを意味します。

[!]hldrq: 初期値から変更しないでください。

[!]stopz: STOP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。

[!]waitz: 初期値から変更しないでください。

[機能]

envコマンドは、エミュレーション環境の設定とDCUの状態を表示します。

設定は変更が必要なパラメータだけを入力ください。入力の順序は任意です。

但し、同じパラメータを2回入力した場合は、後から入力した値が有効です。

初期値は、以下の通りです。

```
CPU Settings:
Auto Run      = ON (auto)
JTAGCLOCK    = 25MHz (jtag25)
Verify        = verify off (!verify)
Signals Mask:
NMI0          = NO MASK (!nmi0)
NMI1          = NO MASK (!nmi1)
NMI2          = NO MASK (!nmi2)
RESET         = NO MASK (!reset)
HLDRQ         = NO MASK (!hldrq)
STOPZ         = NO MASK (!stopz)
WAITZ         = NO MASK (!waitz)
```

[入力例]

```
env reset !nmi2
    RESETをマスクし、NMIをマスクしません。
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

```

e v a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。
ava 1 /del
EVA#1の条件を解除します。
```

e v e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。

```
ave 1 /del
```

EVE#1の条件を解除します。

e v t 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: トレースのクオ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で指定したチャンネル番号と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で指定したチャンネル番号と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オ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。

extbrkコマンド

[書式]

```
extbrk [disable|posi|nega]
```

[パラメータ]

```
disable: 本機能を使用しない時に指定します。(初期値)  
posi:    立ち上がりエッジを検出してブレーク要求を出します。  
nega:    立ち下がりエッジを検出してブレーク要求を出します。
```

[機能]

外部入力信号 (EXTコネクタ-4pin(EX11))から入力する信号を使って、実行をブレークする機能の指定を行う為のコマンドです。

[使用例]

```
extbrk posi  
立ち上がりエッジを検出してブレーク要求を出します。
```

f r dコマンド

[書式]

```
frd [ADDRESS [LENGTH]]
```

[パラメータ]

ADDR: リードするメモリの開始アドレスを16進数で指定します。
指定できるアドレスは、内蔵RAMの範囲に限られます。

LENGTH: 出力するデータを16進数で指定します。(max100)

[機能]

内蔵RAMの内容を実行中にフライバイでリードするためのコマンドです。

[使用例]

```
frd ffff0000 100
```

0xffff0000から0x100バイトリードします。
表示終了直後にリターンキーを押下することで、連続したアドレスの参照ができます。

f w r コマンド

[書式]

fwr [ADDRESS [DATA]]

[パラメータ]

ADDR: ライトするメモリの開始アドレスを16進数で指定します。
指定できるアドレスは、内蔵RAMの範囲に限られます

DATA: 出力するデータを16進数で指定します。

[機能]

内蔵RAMに対し、実行中にフライバイでライトするためのコマンドです。
データ、及びアクセスのサイズは、accコマンドで指定します。

[使用例]

acc hword
 ハーフワードでアクセスする為には、予め、accコマンドで指定が必要です。

fwr ffff1000 1234
 1000Hへハーフワードのデータ：1234hをライトします。

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。

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]
RTE-V850E/GPx-IEを初期化します。全ての環境設定値は初期化されます。
メモリキャッシュの除外エリアは初期化されません。

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]

RTE-V850E/GPx-IEではメモリ参照の高速化を図るため、ICE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、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
```


n c dコマンド

[書式]

ncd ブロック番号

[パラメータ]

ブロック番号: 削除するメモリキャッシュの除外エリアのブロック番号を指定します。

[機能]

メモリキャッシュの除外エリアを削除します。削除は各メモリキャッシュの除外エリアのブロック番号を指定します。初期値の領域は、決して削除しないでください。変更した場合、コマンドでのI/O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。

s f r , s f r 2 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。

レジスタとして使用できる名称は以下の通りです。

<sfrコマンドでアクセスできるレジスタ一覧>

SFR (R/W):

BPC VSWC IMRO IMROL IMROH IMR1
IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L IMR3H IMR4 IMR4L IMR4H IMR5
IMR5L IMR5H IMR6 IMR6L IMR6H VIC512 VIC1M VIC2M VIC4M PIC000 PIC001
PIC002 PIC003 PIC004 PIC005 PIC006 PIC007 PIC008 PIC009 PIC010 PIC011
PIC012 PIC013 PIC014 PIC015 CMR00 CMR01 CMR02 CMR03 CMR04
CMR05 CMR06 CMR07 CMR08 CMR09 CMR10 CMR11 CMR12 CMR13
CMR14 CMR15 CMF00 CMF01 CMF02 CMF03 CMF04 CMF05 CMF06
CMF07 CMF08 CMF09 CMF10 CMF11 CMF12 CMF13 CMF14 CMF15
CMIC16 CMIC17 CMIC18 CMIC19 CMIC20 CMIC21 CMIC22 CMIC23 CMIC24 CMIC25
CMIC26 CMIC27 CMIC28 CMIC29 CMIC30 CMIC31 PW1C0 PW1C1 PW1C2 PW1C3
PW1C4 PW1C5 PW1C6 PW1C7 PW2C0 PW2C1 PW2C2 PW2C3 PW2C4 PW2C5
PW2C6 PW2C7 MAC1C0 CNER1C1 CNRX1C1 CNTX1C1 CNER1C2 CNRX1C2 CNTX1C2
CS1C1 CS1C2 CS1C3 SR1C1 ST1C1 SR1C2 ST1C2 SR1C3 ST1C3 SR1C4 ST1C4
SR1C5 ST1C5 SOFT1C0 PIC10 SOFT1C1 AD1C0 ADMO
ADEN ADST ADTR ADIS MAR0 MAR1
MAR2 MAR3 MAR4 MAR5 MAR6 MAR7 MAR8 MAR9 MAR10 MAR11 MAR12 MAR13 MAR14
MAR15 SAR0 SAR1 SAR2 SAR3 SAR4 SAR5 SAR6 SAR7 DTCR1 DTCR2 DTCR3 DTCR4
DTCR5 DTCR6 DTCR7 DTCR8 DTCR9 DTCR10 DTCR11 DTCR12 DTCR13 DTCR14 DTCR15
DMAMC0 DMAMC1 DMAS0 DMAS1 DMAS2 DTFR1 DTFR2 DTFR3 DTFR4 DTFR5 DTFR6
DTFR7 P0 P1 P2 P3 P4 P5 P6 P9
P10 P11 P12 P13 P14 P15 PM0 PM1 PM2 PM3 PM4 PM5 PM6 PM9
PM10 PM11 PM12 PM13 PM14 PM15 PMC0 PMC1 PMC2 PMC3 PMC9
PMC10 PMC11 PMC12 PMC13 PMC14 PMC15 PPMC10
PPMC11 PPMC12 PPMC13 PPMC14 PPMC15 PPMC16 PPMC17 CMC10 CMD10 CMC11
CMD11 CMC12 CMD12 CMC13 CMD13 CMC14 CMD14 CMC15 CMD15 CMC16 CMD16 CMC17
CMD17 PPMC20 PPMC21 PPMC22 PPMC23 PPMC24 PPMC25 PPMC26
PPMC27 CMC20 CMD20 CMC21 CMD21 CMC22 CMD22 CMC23 CMD23 CMC24 CMD24
CMC25 CMD25 CMC26 CMD26 CMC27 CMD27 P16
P17 P18 PM16 PM17 PM18 PMC16
PMC17 PMC18 PFC17 PLO PL1 PES0
PES1 NRC PHS CKC INTMO INTM1
ASIM10 ASIM11 ATXB1 ATXBL1 ACKSR1
ABRGC1 ASIM20 ASIM21 PRS2M PRS2CM
ASIM30 ASIM31 ATXB3 ATXBL3 ACKSR3
ABRGC3 ASIM40 ASIM41 ATXB4 ATXBL4 ACKSR4
ABRGC4 ASIM50 ASIM51 ATXB5 ATXBL5 ACKSR5
ABRGC5 TOC0 TOC1 TOC2 TOC3 TMDL0
TMDL1 TMDL2 TMDL3 OS10 OS11 CPCNT8 CPCNT9 CPCNT10 CPCNT11 CPCNT12 CPCNT13
CPCNT14 CPCNT15 CMR0 CMR1 CMR2
CMR3 CMR4 CMR5 CMR6 CMR7 CMR8 CMR9 CMR10
CMR11 CMR12 CMR13 CMR14 CMR15 CMF0 CMF1 CMF2
CMF3 CMF4 CMF5 CMF6 CMF7 CMF8 CMF9 CMF10
CMF11 CMF12 CMF13 CMF14 CMF15 CM16 CM17 CM18

CM19 CM20 CM21 CM22 CM23 CM24 CM25 CM26
 CM27 CM28 CM29 CM30 CM31 SES0 SES1 SES2 SES3 CTXB1
 CTXB10 CTXBL10 CTXB11 CSIM10 CSIM11 CSIL1 CSIC1 CSIS1 CBRGC1 ITC1 CTXB2
 CTXB20 CTXBL20 CTXB21 CSIM20 CSIM21 CSIL2 CSIC2 CSIS2 CBRGC2 ITC2 CTXB3
 CTXB30 CTXBL30 CTXB31 CSIM30 CSIM31 CSIL3 CSIC3 CSIS3 CBRGC3 ITC3 DMAWC0
 DMAWC1

SFR (W) :

PRCMD ATXS2 ATXSL2

SFR (R) :

ISPR ADCR0 ADCR1 ADCR2 ADCR3
 ADCR4 ADCR5 ADCR6 ADCR7 ADCR8 ADCR9 ADCR10 ADCR11 ADCR12 ADCR13 ADCR14
 ADCR15 CKDR ARXB1 ARXBL1 ASIS1
 ASIF1 ASIS2 ARXB2 ARXBL2 ARXB3
 ARXBL3 ASIS3 ASIF3 ARXB4 ARXBL4
 ASIS4 ASIF4 ARXB5 ARXBL5 ASIS5
 ASIF5 GTM CPRO CPF0 CPR1 CPF1
 CP2 CP3 CP4 CP5 CP6 CP7 CP8 CP9
 CP10 CP11 CP12 CP13 CP14 CP15 CRXB1
 CRXB10 CRXBL10 CRXB11 CRXB2 CRXB20
 CRXBL20 CRXB21 CRXB3 CRXB30 CRXBL30
 CRXB31

<sfr2コマンドでアクセスできるレジスタ一覧>

SFR (R/W) :

M_DLC00 M_CTRL00 M_TIME00 M_DATA000
 M_DATA001 M_DATA002 M_DATA003 M_DATA004 M_DATA005 M_DATA006 M_DATA007
 M_ID00 M_IDL00 M_IDH00 M_CONF00 M_DLC01 M_CTRL01 M_TIME01 M_DATA010
 M_DATA011 M_DATA012 M_DATA013 M_DATA014 M_DATA015 M_DATA016 M_DATA017
 M_ID01 M_IDL01 M_IDH01 M_CONF01 M_DLC02 M_CTRL02 M_TIME02 M_DATA020
 M_DATA021 M_DATA022 M_DATA023 M_DATA024 M_DATA025 M_DATA026 M_DATA027
 M_ID02 M_IDL02 M_IDH02 M_CONF02 M_DLC03 M_CTRL03 M_TIME03 M_DATA030
 M_DATA031 M_DATA032 M_DATA033 M_DATA034 M_DATA035 M_DATA036 M_DATA037
 M_ID03 M_IDL03 M_IDH03 M_CONF03 M_DLC04 M_CTRL4 M_TIME04 M_DATA040
 M_DATA041 M_DATA042 M_DATA043 M_DATA044 M_DATA045 M_DATA046 M_DATA047
 M_ID04 M_IDL04 M_IDH04 M_CONF04 M_DLC05 M_CTRL05 M_TIME05 M_DATA050
 M_DATA051 M_DATA052 M_DATA053 M_DATA054 M_DATA055 M_DATA056 M_DATA057
 M_ID05 M_IDL05 M_IDH05 M_CONF05 M_DLC06 M_CTRL06 M_TIME06 M_DATA060
 M_DATA061 M_DATA062 M_DATA063 M_DATA064 M_DATA065 M_DATA066 M_DATA067
 M_ID06 M_IDL06 M_IDH06 M_CONF06 M_DLC07 M_CTRL07 M_TIME07 M_DATA070
 M_DATA071 M_DATA072 M_DATA073 M_DATA074 M_DATA075 M_DATA076 M_DATA077
 M_ID07 M_IDL07 M_IDH07 M_CONF07 M_DLC08 M_CTRL08 M_TIME08 M_DATA080
 M_DATA081 M_DATA082 M_DATA083 M_DATA084 M_DATA085 M_DATA086 M_DATA087
 M_ID08 M_IDL08 M_IDH08 M_CONF08 M_DLC09 M_CTRL09 M_TIME09 M_DATA090
 M_DATA091 M_DATA092 M_DATA093 M_DATA094 M_DATA095 M_DATA096 M_DATA097
 M_ID09 M_IDL09 M_IDH09 M_CONF09 M_DLC10 M_CTRL10 M_TIME10 M_DATA100
 M_DATA101 M_DATA102 M_DATA103 M_DATA104 M_DATA105 M_DATA106 M_DATA107
 M_ID10 M_IDL10 M_IDH10 M_CONF10 M_DLC11 M_CTRL11 M_TIME11 M_DATA110
 M_DATA111 M_DATA112 M_DATA113 M_DATA114 M_DATA115 M_DATA116 M_DATA117
 M_ID11 M_IDL11 M_IDH11 M_CONF11 M_DLC12 M_CTRL12 M_TIME12 M_DATA120
 M_DATA121 M_DATA122 M_DATA123 M_DATA124 M_DATA125 M_DATA126 M_DATA127
 M_ID12 M_IDL12 M_IDH12 M_CONF12 M_DLC13 M_CTRL13 M_TIME13 M_DATA130
 M_DATA131 M_DATA132 M_DATA133 M_DATA134 M_DATA135 M_DATA136 M_DATA137
 M_ID13 M_IDL13 M_IDH13 M_CN13 M_DLC14 M_CTRL14 M_TIME14 M_DATA140
 M_DATA141 M_DATA142 M_DATA143 M_DATA144 M_DATA145 M_DATA146 M_DATA147
 M_ID14 M_IDL14 M_IDH14 M_CONF14 M_DLC15 M_CTRL15 M_TIME15 M_DATA150
 M_DATA151 M_DATA152 M_DATA153 M_DATA154 M_DATA155 M_DATA156 M_DATA157

M_ID15 M_IDL15 M_IDH15 M_CONF15 M_DLC16 M_CTRL16 M_TIME16 M_DATA160
M_DATA161 M_DATA162 M_DATA163 M_DATA164 M_DATA165 M_DATA166 M_DATA167
M_ID16 M_IDL16 M_IDH16 M_CONF16 M_DLC17 M_CTRL17 M_TIME17 M_DATA170
M_DATA171 M_DATA172 M_DATA173 M_DATA174 M_DATA175 M_DATA176 M_DATA177
M_ID17 M_IDL17 M_IDH17 M_CONF17 M_DLC18 M_CTRL18 M_TIME18 M_DATA180
M_DATA181 M_DATA182 M_DATA183 M_DATA184 M_DATA185 M_DATA186 M_DATA187
M_ID18 M_IDL18 M_IDH18 M_CONF18 M_DLC19 M_CTRL19 M_TIME19 M_DATA190
M_DATA191 M_DATA192 M_DATA193 M_DATA194 M_DATA195 M_DATA196 M_DATA197
M_ID19 M_IDL19 M_IDH19 M_CONF19 M_DLC20 M_CTRL20 M_TIME20 M_DATA200
M_DATA201 M_DATA202 M_DATA203 M_DATA204 M_DATA205 M_DATA206 M_DATA207
M_ID20 M_IDL20 M_IDH20 M_CONF20 M_DLC21 M_CTRL21 M_TIME21 M_DATA210
M_DATA211 M_DATA212 M_DATA213 M_DATA214 M_DATA215 M_DATA216 M_DATA217
M_ID21 M_IDL21 M_IDH21 M_CONF21 M_DLC22 M_CTRL22 M_TIME22 M_DATA220
M_DATA221 M_DATA222 M_DATA223 M_DATA224 M_DATA225 M_DATA226 M_DATA227
M_ID22 M_IDL22 M_IDH22 M_CONF22 M_DLC23 M_CTRL23 M_TIME23 M_DATA230
M_DATA231 M_DATA232 M_DATA233 M_DATA234 M_DATA235 M_DATA236 M_DATA237
M_ID23 M_IDL23 M_IDH23 M_CONF23 M_DLC24 M_CTRL24 M_TIME24 M_DATA240
M_DATA241 M_DATA242 M_DATA243 M_DATA244 M_DATA245 M_DATA246 M_DATA247
M_ID24 M_IDL24 M_IDH24 M_CONF24 M_DLC25 M_CTRL25 M_TIME25 M_DATA250
M_DATA251 M_DATA252 M_DATA253 M_DATA254 M_DATA255 M_DATA256 M_DATA257
M_ID25 M_IDL25 M_IDH25 M_CONF25 M_DLC26 M_CTRL26 M_TIME26 M_DATA260
M_DATA261 M_DATA262 M_DATA263 M_DATA264 M_DATA265 M_DATA266 M_DATA267
M_ID26 M_IDL26 M_IDH26 M_CONF26 M_DLC27 M_CTRL27 M_TIME27 M_DATA270
M_DATA271 M_DATA272 M_DATA273 M_DATA274 M_DATA275 M_DATA276 M_DATA277
M_ID27 M_IDL27 M_IDH27 M_CONF27 M_DLC28 M_CTRL28 M_TIME28 M_DATA280
M_DATA281 M_DATA282 M_DATA283 M_DATA284 M_DATA285 M_DATA286 M_DATA287
M_ID28 M_IDL28 M_IDH28 M_CONF28 M_DLC29 M_CTRL29 M_TIME29 M_DATA290
M_DATA291 M_DATA292 M_DATA293 M_DATA294 M_DATA295 M_DATA296 M_DATA297
M_ID29 M_IDL29 M_IDH29 M_CONF29 M_DLC30 M_CTRL30 M_TIME30 M_DATA300
M_DATA301 M_DATA302 M_DATA303 M_DATA304 M_DATA305 M_DATA306 M_DATA307
M_ID30 M_IDL30 M_IDH30 M_CONF30 M_DLC31 M_CTRL31 M_TIME31 M_DATA310
M_DATA311 M_DATA312 M_DATA313 M_DATA314 M_DATA315 M_DATA316 M_DATA317
M_ID31 M_IDL31 M_IDH31 M_CONF31 CSTOP
CGST CGIE CGCS CGINTP C1INTP
C2INTP C1MASKO C1MASKLO C1MASKHO
C1MASK1 C1MASKL1 C1MASKH1 C1MASK2 C1MASKL2 C1MASKH2 C1MASK3 C1MASKL3
C1MASKH3 C1CTRL C1DEF C1IE C1BRP C1SYNC C2MASKO
C2MASKLO C2MASKHO C2MASK1 C2MASKL1 C2MASKH1 C2MASK2 C2MASKL2 C2MASKH2
C2MASK3 C2MASKL3 C2MASKH3 C2CTRL C2DEF C2IE C2BRP C2SYNC

SFR (W) :

SC_STAT00 SC_STAT01 SC_STAT02
SC_STAT03 SC_STAT04 SC_STAT05
SC_STAT06 SC_STAT07 SC_STAT08
SC_STAT09 SC_STAT10 SC_STAT11
SC_STAT12 SC_STAT13 SC_STAT14
SC_STAT15 SC_STAT16 SC_STAT17
NSCST18 SC_STAT19 SC_STAT20 SC_STAT21
SC_STAT22 SC_STAT23 SC_STAT24
SC_STAT25 SC_STAT26 SC_STAT27
SC_STAT28 SC_STAT29 SC_STAT30
SC_STAT31 CGMSS

SFR (R) :

M_STAT00 M_STAT01 M_STAT02 M_STAT03
M_STAT04 M_STAT05 M_STAT06 M_STAT07
M_STAT08 M_STAT09 M_STAT10 M_STAT11
M_STAT12 M_STAT13 M_STAT14 M_STAT15

M_STAT16 M_STAT17 M_STAT18 M_STAT19
M_STAT20 M_STAT21 M_STAT22 M_STAT23
M_STAT24 M_STAT25 M_STAT26 M_STAT27
M_STAT28 M_STAT29 M_STAT30 M_STAT31
CCINTP CGTSC CGMSR C1LAST C1ERC
C1BA C1DINF C2LAST C2ERC C2BA
C2DINF

[機能]

SFRレジスタ値の設定と表示を行います。

[使用例]

sfr P10
P10レジスタの値を表示します。
sfr PM10 0
PM10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）をできます。

[使用例]

```
symfile c:%test%dry%dry.elf  
          c:%test%dryのディレクトリからelfファイル:dry.elfのシンボルを読み込みます。  
sym m  
          mから始まるシンボルを最大30個表示します。
```

t i m e コマンド

[書式]

time [sysclk]

[パラメータ]

sysclk: cpu のシステムクロックを MHz の単位で指定します。小数点以下 2 桁まで有効です。指定しなかった場合のデフォルト値は、64MHz です。

[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。タイマーの値は CPU クロックで 1 回カウントしています。

[備考]

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

[使用例]

```
>time
```

```
Time = 1,139,655,796 (ns) (64.000MHz) [Counter=0x000458f1f3]>
```

直前の実行からブレークまでの時間を表示します。

```
>time 32
```

CPU の動作クロックが初期値の 64MHz 以外の時に変更(この場合は 32MHz)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、「付録. B 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 [subor|suband] [[!]dmatrc] [[!]sfrtrc] [[!]tendbrk] [[!]stop]
      [rtrcb{0|8|16}]|[nrtrcb{0|4|8|12|16|20|24}]
      [nonbranchNN] [[!]phold] [[!]once] [[!]debug]
```

[パラメータ]

subor: サブスイッチとして、セクション条件とクオ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オリファイ条件のアンドを指定します。

[[!]dmatrc: DMAのサイクルをトレースします。!で禁止します。

[[!]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バ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バ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。

[[!]debug: 初期値 (!debug)でご使用ください。

[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、セクションとクオ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、「付録. B 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、「付録. B トレース機能の詳細」を参照ください。

t s p 1, t s p 2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、「付録. B 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、「付録. B 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 = 0..1ffff デレイカウンタ

- トリガ成立後にメモリに取り込むフレーム数を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ー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、オーバー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の立ち下がり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 1ffff
```

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:0x1fff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 ffff
```

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:0xffff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0xffffサイクルがトレースに入ります。

```
tp 1000
```

```
tron !delay ffff
```

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:0xffff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0xffff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0xffff）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、「付録. B 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] [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EX13..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)
DMAW	DMAによるiRAM書き込み発生 (バストレース)
DMAR	DMAによるiRAM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]

トレースに関する詳細は、「付録. B トレース機能の詳細」を参照ください。

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|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。

verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

RTE-V850E/GPx-IEのバージョンを表示します。