

付録. D K I T - V 8 5 0 E / M A 3 - I E 内部コマンド

本書は、K I T - V 8 5 0 E / M A 3 - I E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。

G H S - M u l t iの場合

R T E S E R Vを接続後、ターゲット・ウインドウで直接入力できます。

P A R T N E R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。

コマンド一覧

コマンド一覧.....	D-1
コマンド書式.....	D-2
アクセス系ブレークポイント : ABP, ABP1, ABP2コマンド.....	D-3
環境設定 : ENV, EMEMSTATコマンド.....	D-4
イベント設定状態の表示 : EMODEコマンド.....	D-6
アクセス系イベントの設定 : EVAコマンド.....	D-7
実行系イベントの設定 : EVEコマンド.....	D-8
イベント統合の設定 : EVTコマンド.....	D-9
外部からのブレーク設定 : EXTBRKコマンド.....	D-10
ヘルプ : HELPコマンド.....	D-11
INPUT : INB, INH, INWコマンド.....	D-12
初期化 : INITコマンド.....	D-13
JTAGリード : JREADコマンド.....	D-14
デバッガキャッシュ領域の解除 : NCコマンド.....	D-15
デバッガキャッシュ領域の設定 : NCDコマンド.....	D-16
ソフトブレーク禁止領域の設定 : NSBPコマンド.....	D-17
ソフトブレーク禁止領域の解除 : NSBPDコマンド.....	D-18
強制ユーザ領域の設定 : NRROMコマンド.....	D-19
強制ユーザ領域の解除 : NRROMDコマンド.....	D-20
OUTPUT : OUTB, OUTH, OUTWコマンド.....	D-21
CPUリセット : RESETコマンド.....	D-22
E. ROMの設定 : ROM1. . ROM4コマンド.....	D-23
シーケンシャル条件の設定 : SEQコマンド.....	D-25
サブスイッチの設定 : SSWON, SSWOFFコマンド.....	D-26
サブスイッチ条件の設定 : SSWENVコマンド.....	D-29
SFRアクセス : SFRコマンド.....	D-30
シンボル : SYMFILE, SYMコマンド.....	D-32
トレースデータ条件 : TD1. . TD8コマンド.....	D-33
トレースの環境設定 : TENVコマンド.....	D-34
トリガポイント : TPコマンド.....	D-35
トレーススイッチポイント : TSP1, TSP2コマンド.....	D-36
トレース条件の参照 : TMODEコマンド.....	D-37
トレースの設定&開始 : TRONコマンド.....	D-38
トレースの強制終了 : TROFFコマンド.....	D-40
トレースの表示 : TRACEコマンド.....	D-41
トレースのファイル書き出し : FTRACEコマンド.....	D-44
実行時間計測 : TIMEコマンド.....	D-45
バージョン表示 : VERコマンド.....	D-46

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

KIT-V850E/MA3-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。
```

env, ememstatコマンド

[書式]

```
env [[!]auto] [[!][verify]] [jtag[xxx][. [yyy]] {M|K}]
    [ram{4|12|28|60}]
    [[!]nmi] [[!]intwdt] [[!]resetz] [[!]hldrqz] [[!]waitz]
ememstat
```

[パラメータ]

[[!]auto: 実行中にブレークポイントを設定した場合一時的にブレークしますが、その後の実行を自動的に行う場合に[Auto]、行わない場合に[!auto]を指定します。

[[!]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。

[ram{4|12|28|60}]: 内蔵RAMの容量を設定します。
それぞれ以下に対応します。
[4KB|12KB|28K|60K]

[[!]nmi: nmi信号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。

[[!]intwdt: 内部のintwdt信号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。

[[!]resetz: RESET-信号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。

[[!]hldrqz: HLDQ-信号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: WAIT-信号のマスク指定を指定します。!はマスクしないを意味します。

[機能]

envコマンドは、エミュレーション環境の設定と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E.MEM基板の実装状態を表示するコマンドです。以下に表示例を示します。

```
>env
Probe:
  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)
  CPU Mode   = single (single)
  RAM Size   = 60K (ram60)
Signals Mask:
  NMI        = NO MASK (!nmi)
  INTWDT     = NO MASK (!intwdt)
```

```
RESETZ      = NO MASK (!resetz)
HLDRQZ      = NO MASK (!hldrqz)
WAITZ       = NO MASK (!waitz)
```

```
>ememstat
```

```
Board_num  EMEM_Size  ROM_Probe
```

```
=====
ROM1      32Mbyte    Extend Type 2K
```

<< EMEMボードの実装状態に依存します。

[入力例]

```
env reset !nmi2
RESETをマスクし、NM12をマスクしません。
env verify
Verify機能をONにします。
env jtag40m
JTAGクロックを40MHzに設定します。
```

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 ch1に設定します。

```
ava 1 /del
```

EVA ch1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 ch1に設定します。

```
ave 1 /del
```

EVE ch1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:RSV-IN0) 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ためのコマンドです。

[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。
```

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

j r e a dコマンド

[書式]

```
jread [ADDR [LENGTH]]
```

[パラメータ]

ADDR: アドレスを16進数で指定します。
LENGTH: 読み出すバイト数を16進数で指定します。(max 100h)

[機能]

ROMコマンドで割り付けたROMエミュレーション領域をJTAG(CPU)から読み出すためのコマンドです。(通常のコマンドでは、ROMエミュレーション領域へのアクセスは内部のメモリに対し直接行っています。)

[使用例]

```
jread 100000 100  
100000hから100hバイトをJTAG経由で読み出します。
```

ncコマンド

[書式]

```
nc [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: メモリキャッシュの除外エリアの開始アドレスを指定します。
LENGTH: メモリキャッシュの除外エリアのバイト数を指定します。
 デフォルト値32バイト、最少値32バイト

[機能]

メモリ参照の高速化を図るため、ファームウェア内に8ブロック*32バイトのメモリリードキャッシュを持っています。同一アドレスのメモリ参照など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。

[表示例]

初期値の表示です。

```
>nc
No Memory Cache Area
No. Address   Length
1 1ffff000   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
```


nsbpコマンド

[書式]

```
nsbp [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: ソフトウェアブレイク禁止領域の開始アドレスを指定します。
LENGTH: ソフトウェアブレイク禁止領域のバイト数を指定します。
指定領域の最小単位はハーフワードバウンダリです。
また、指定できる領域の数は最大4ヶ所です。

[機能]

ソフトウェアブレイクを禁止したい領域を指定します。
ブレイクポイントを指定した場合、デバッガは暗黙的に対象アドレスに対し、メモリテスト(ライトアクセス)を行います。
一部のフラッシュROM等、ライトアクセスを行うことでメモリの状態が変わり、正しいデータの読み出しが行えなくなる場合等に、ライトサイクルを禁止する目的で指定してください。
通常は、指定する必要はありません。

[使用例]

```
nsbp 10000 20000  
10000h番地から20000バイトの領域をソフトウェアブレイク禁止領域に指定します。
```

```
>nsbp 100000 20000  
Num Address Length  
01 00100000 00020000
```

nsbpdコマンド

[書式]

nsbpd [ブロック番号|/all]

[パラメータ]

ブロック番号: 削除するソフトウェアブレイク禁止領域のブロック番号を指定します。
/all: 全てのソフトウェアブレイク禁止領域を削除します。

[機能]

nsbpで指定したソフトウェアブレイク禁止領域を削除します。

[使用例]

nsbpd 1
ブロック番号 1 をソフトウェアブレイク禁止領域から削除します。

```
nsbp
Num Address Length
01 00100000 00200000
02 00400000 00010000
```

```
>nsbpd 1
Num Address Length
01 00400000 00010000
```

n r o mコマンド

[書式]

```
nrom [[ADDR [LENGTH]]]
```

[パラメータ]

ADDR: 強制ユーザ領域の開始アドレスを指定します。
 LENGTH: 強制ユーザ領域のバイト数を指定します。
 指定領域の最小単位は、以下の通りです。
 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ROMケーブルのEMEMEN-信号はインアクティブ(Highレベル)になります。

[使用例]

```
nrom 0 20000
```

0h番地から20000バイトを強制ユーザ領域に指定します。

```
>nrom 0 20000
```

No.	Address	Length
1	00000000	00020000

```
>nrom 100000 40000
```

No.	Address	Length
1	00000000	00020000
2	00100000	00040000

n r o m dコマンド

[書式]

nromd [ブロック番号|/all]

[パラメータ]

ブロック番号: 削除する強制ユーザ領域のブロック番号を指定します。
/all: 全ての強制ユーザ領域のブロックを削除します。

[機能]

nromで指定した強制ユーザ領域を削除します。

[使用例]

ncd 1

ブロック番号 1 を強制ユーザ領域から削除します。

```
>nrom 100000 40000
```

No.	Address	Length
1	00000000	00020000
2	00100000	00040000

```
>nromd 1
```

No.	Address	Length
1	00100000	00040000

outb, outh, outwコマンド

[書式]

```
outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
```

[パラメータ]

ADDR: 出力ポートのアドレスを16進数で指定します。
DATA: 出力するデータを16進数で指定します。

[機能]

outb, outh, outwは、アクセスサイズを区別して、ライトを行ないます。
outbはバイト、outhはハーフ・ワード、outwはワード単位でアクセスします。

[使用例]

```
outb 1000 12
    1000Hへバイトデータ : 12hをライトします。
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。

rom1..rom4コマンド

[書式]

```
rom1 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16|bus32|bus64] [[!]  
wren]
rom2 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16] [[!]  
wren]
rom3 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16|bus32] [[!]  
wren]
rom4 [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
      [bus8|bus16] [[!]  
wren]
```

rom1: スロット#3に実装されたEMEM基板を含むモジュールに対する設定コマンドです。
rom2: スロット#4に実装されたEMEM基板を含むモジュールに対する設定コマンドです。
rom3: スロット#5に実装されたEMEM基板を含むモジュールに対する設定コマンドです。
rom4: スロット#6に実装されたEMEM基板を含むモジュールに対する設定コマンドです。

[パラメータ]

ADDR [LENGTH]: エミュレーションする領域を指定します。

ADDR: 開始アドレスを指定します。
エミュレートするROMの最下位のアドレス (ROMのバウンダリ) に合致していない場合、指定アドレス以下のアドレス領域は非エミュレーション領域になります。

LENGTH: エミュレートするROMのバイト数を指定します。

備考: ADDR, LENGTHで指定できる領域の最小単位は、エミュレーションしているROMのサイズに応じ、以下の通りです。

- ・ 8/16-bit : 128k-byte単位
- ・ 32-bit : 256k-byte単位
- ・ 64-bit : 512k-byte単位

512k|1m|2m|4m|8m|16m|32m|64m|128m|256m:

1本のROMプローブでエミュレートするROMのBit容量を指定します。512K-bitから256M-bit(32M-Byte)までの値が指定できます。
例えば、27C1024の場合は、1mを指定します。

rom8|rom16: エミュレートするROMのデータビット数を指定します。

8bitと16bitが指定できます。DIP32のアダプタを使用する場合はrom8、DIP-40/42のアダプタ、及び16bit-標準ROMケーブルをそのまま使用する場合は、rom16を指定します。

bus8|bus16|bus32|bus64:

エミュレートするシステムの中でのROMのバスサイズを指定します。
8bit, 16bit, 32bit, 64bitが指定できます。

>> [64-bit]は将来のためのパラメータです。(本KITでは使用しません)

[[!]
wren]: Write Enable:エミュレーションメモリをRAMとして使用する場合の設定です。
wrenで書込み許可、!wrenで書込み禁止です。初期値は!wrenです。

[機能]

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

[入力例]

>rom1 100000 300000 32m rom16 bus16 !wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#3	100000 - 3ffffff	16-bit	16-bit	32M-Bit	禁止

>rom2 140000 40000 2m rom16 bus16 wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#4	140000 - 17ffff	16-bit	16-bit	2M-Bit	許可

>rom1 0 80000 2m rom rom16 bus32 !wren

対象EMEM基板 スロット位置	アドレス範囲	バス幅	ROM		ライトイネーブル
			バス幅	Bit数	
#3+#4	000000 - 07ffff	32-bit	16-bit	2M-Bit	禁止

この時、rom2コマンドは発行しないでください。

<備考>

romコマンドで指定した領域における注意事項

rom1..rom4コマンドで指定した範囲へのデバッグからのアクセスは、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|5|6|7|8} {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}]
            [evep{1|2|3|4|5|6|7|8} {enable|disable}]
            [ever {1|3|5|7} {enable|disable}]
```

[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:opcode

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|5|6|7|8} {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。
evcp{1|2|3|4|5|6|7|8} {enable|disable} :
    evcpコマンドで指定したポイント条件の取り込みを指定します。
    enableで取り込みます。
ever {1|3|5|7} {enable|disable} :
    everコマンドで指定した範囲条件の取り込みを指定します。
    enableで取り込みます。

```

[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)        = No cycle (none)
td2 Trace cycle (td2)        = No cycle (none)
td3 Trace cycle (td3)        = No cycle (none)
td4 Trace cycle (td4)        = No cycle (none)
td5 Trace cycle (td5)        = No cycle (none)
td6 Trace cycle (td6)        = No cycle (none)
td7 Trace cycle (td7)        = No cycle (none)
td8 Trace cycle (td8)        = 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)
evap7 Trace cycle (evap7)    = No cycle (none)
evap8 Trace cycle (evap8)    = No cycle (none)
evap9 Trace cycle (evap9)    = No cycle (none)
evap10 Trace cycle (evap10)  = No cycle (none)
evap11 Trace cycle (evap11)  = No cycle (none)
evap12 Trace cycle (evap12)  = No cycle (none)
evap13 Trace cycle (evap13)  = No cycle (none)
evap14 Trace cycle (evap14)  = No cycle (none)
evap15 Trace cycle (evap15)  = No cycle (none)
evap16 Trace cycle (evap16)  = No cycle (none)
evap17 Trace cycle (evap17)  = No cycle (none)
evap18 Trace cycle (evap18)  = No cycle (none)
evap19 Trace cycle (evap19)  = No cycle (none)
evap20 Trace cycle (evap20)  = No cycle (none)
evap21 Trace cycle (evap21)  = No cycle (none)
evap22 Trace cycle (evap22)  = No cycle (none)
evap23 Trace cycle (evap23)  = No cycle (none)
evap24 Trace cycle (evap24)  = No cycle (none)
evap25 Trace cycle (evap25)  = No cycle (none)
evap26 Trace cycle (evap26)  = No cycle (none)
evap27 Trace cycle (evap27)  = No cycle (none)
evap28 Trace cycle (evap28)  = No cycle (none)
evap29 Trace cycle (evap29)  = No cycle (none)
evap30 Trace cycle (evap30)  = No cycle (none)
evap31 Trace cycle (evap31)  = No cycle (none)
evap32 Trace cycle (evap32)  = No cycle (none)
evap33 Trace cycle (evap33)  = No cycle (none)
evap34 Trace cycle (evap34)  = No cycle (none)
evap35 Trace cycle (evap35)  = No cycle (none)
evap36 Trace cycle (evap36)  = No cycle (none)
evap37 Trace cycle (evap37)  = No cycle (none)
evap38 Trace cycle (evap38)  = No cycle (none)
evap39 Trace cycle (evap39)  = No cycle (none)
evap40 Trace cycle (evap40)  = No cycle (none)
evap41 Trace cycle (evap41)  = No cycle (none)
evap42 Trace cycle (evap42)  = No cycle (none)
evap43 Trace cycle (evap43)  = No cycle (none)
evap44 Trace cycle (evap44)  = No cycle (none)
evap45 Trace cycle (evap45)  = No cycle (none)
evap46 Trace cycle (evap46)  = No cycle (none)
evap47 Trace cycle (evap47)  = No cycle (none)
evap48 Trace cycle (evap48)  = No cycle (none)
evap49 Trace cycle (evap49)  = No cycle (none)
evap50 Trace cycle (evap50)  = No cycle (none)
evap51 Trace cycle (evap51)  = No cycle (none)
evap52 Trace cycle (evap52)  = No cycle (none)
evap53 Trace cycle (evap53)  = No cycle (none)
evap54 Trace cycle (evap54)  = No cycle (none)
evap55 Trace cycle (evap55)  = No cycle (none)
evap56 Trace cycle (evap56)  = No cycle (none)
evap57 Trace cycle (evap57)  = No cycle (none)
evap58 Trace cycle (evap58)  = No cycle (none)
evap59 Trace cycle (evap59)  = No cycle (none)
evap60 Trace cycle (evap60)  = No cycle (none)
evap61 Trace cycle (evap61)  = No cycle (none)
evap62 Trace cycle (evap62)  = No cycle (none)
evap63 Trace cycle (evap63)  = No cycle (none)
evap64 Trace cycle (evap64)  = No cycle (none)
evap65 Trace cycle (evap65)  = No cycle (none)
evap66 Trace cycle (evap66)  = No cycle (none)
evap67 Trace cycle (evap67)  = No cycle (none)
evap68 Trace cycle (evap68)  = No cycle (none)
evap69 Trace cycle (evap69)  = No cycle (none)
evap70 Trace cycle (evap70)  = No cycle (none)
evap71 Trace cycle (evap71)  = No cycle (none)
evap72 Trace cycle (evap72)  = No cycle (none)
evap73 Trace cycle (evap73)  = No cycle (none)
evap74 Trace cycle (evap74)  = No cycle (none)
evap75 Trace cycle (evap75)  = No cycle (none)
evap76 Trace cycle (evap76)  = No cycle (none)
evap77 Trace cycle (evap77)  = No cycle (none)
evap78 Trace cycle (evap78)  = No cycle (none)
evap79 Trace cycle (evap79)  = No cycle (none)
evap80 Trace cycle (evap80)  = No cycle (none)
evap81 Trace cycle (evap81)  = No cycle (none)
evap82 Trace cycle (evap82)  = No cycle (none)
evap83 Trace cycle (evap83)  = No cycle (none)
evap84 Trace cycle (evap84)  = No cycle (none)
evap85 Trace cycle (evap85)  = No cycle (none)
evap86 Trace cycle (evap86)  = No cycle (none)
evap87 Trace cycle (evap87)  = No cycle (none)
evap88 Trace cycle (evap88)  = No cycle (none)
evap89 Trace cycle (evap89)  = No cycle (none)
evap90 Trace cycle (evap90)  = No cycle (none)
evap91 Trace cycle (evap91)  = No cycle (none)
evap92 Trace cycle (evap92)  = No cycle (none)
evap93 Trace cycle (evap93)  = No cycle (none)
evap94 Trace cycle (evap94)  = No cycle (none)
evap95 Trace cycle (evap95)  = No cycle (none)
evap96 Trace cycle (evap96)  = No cycle (none)
evap97 Trace cycle (evap97)  = No cycle (none)
evap98 Trace cycle (evap98)  = No cycle (none)
evap99 Trace cycle (evap99)  = No cycle (none)
evap100 Trace cycle (evap100) = No cycle (none)
All access cycle (all_cycle) = No cycle (none)
evcp1 Trace (evcp1)          = Disable (disable)
evcp2 Trace (evcp2)          = Disable (disable)
evcp3 Trace (evcp3)          = Disable (disable)
evcp4 Trace (evcp4)          = Disable (disable)
evcp5 Trace (evcp5)          = Disable (disable)
evcp6 Trace (evcp6)          = Disable (disable)
evcp7 Trace (evcp7)          = Disable (disable)
evcp8 Trace (evcp8)          = Disable (disable)
evcp9 Trace (evcp9)          = Disable (disable)
evcp10 Trace (evcp10)        = Disable (disable)
evcp11 Trace (evcp11)        = Disable (disable)
evcp12 Trace (evcp12)        = Disable (disable)
evcp13 Trace (evcp13)        = Disable (disable)
evcp14 Trace (evcp14)        = Disable (disable)
evcp15 Trace (evcp15)        = Disable (disable)
evcp16 Trace (evcp16)        = Disable (disable)
evcp17 Trace (evcp17)        = Disable (disable)
evcp18 Trace (evcp18)        = Disable (disable)
evcp19 Trace (evcp19)        = Disable (disable)
evcp20 Trace (evcp20)        = Disable (disable)
evcp21 Trace (evcp21)        = Disable (disable)
evcp22 Trace (evcp22)        = Disable (disable)
evcp23 Trace (evcp23)        = Disable (disable)
evcp24 Trace (evcp24)        = Disable (disable)
evcp25 Trace (evcp25)        = Disable (disable)
evcp26 Trace (evcp26)        = Disable (disable)
evcp27 Trace (evcp27)        = Disable (disable)
evcp28 Trace (evcp28)        = Disable (disable)
evcp29 Trace (evcp29)        = Disable (disable)
evcp30 Trace (evcp30)        = Disable (disable)
evcp31 Trace (evcp31)        = Disable (disable)
evcp32 Trace (evcp32)        = Disable (disable)
evcp33 Trace (evcp33)        = Disable (disable)
evcp34 Trace (evcp34)        = Disable (disable)
evcp35 Trace (evcp35)        = Disable (disable)
evcp36 Trace (evcp36)        = Disable (disable)
evcp37 Trace (evcp37)        = Disable (disable)
evcp38 Trace (evcp38)        = Disable (disable)
evcp39 Trace (evcp39)        = Disable (disable)
evcp40 Trace (evcp40)        = Disable (disable)
evcp41 Trace (evcp41)        = Disable (disable)
evcp42 Trace (evcp42)        = Disable (disable)
evcp43 Trace (evcp43)        = Disable (disable)
evcp44 Trace (evcp44)        = Disable (disable)
evcp45 Trace (evcp45)        = Disable (disable)
evcp46 Trace (evcp46)        = Disable (disable)
evcp47 Trace (evcp47)        = Disable (disable)
evcp48 Trace (evcp48)        = Disable (disable)
evcp49 Trace (evcp49)        = Disable (disable)
evcp50 Trace (evcp50)        = Disable (disable)
evcp51 Trace (evcp51)        = Disable (disable)
evcp52 Trace (evcp52)        = Disable (disable)
evcp53 Trace (evcp53)        = Disable (disable)
evcp54 Trace (evcp54)        = Disable (disable)
evcp55 Trace (evcp55)        = Disable (disable)
evcp56 Trace (evcp56)        = Disable (disable)
evcp57 Trace (evcp57)        = Disable (disable)
evcp58 Trace (evcp58)        = Disable (disable)
evcp59 Trace (evcp59)        = Disable (disable)
evcp60 Trace (evcp60)        = Disable (disable)
evcp61 Trace (evcp61)        = Disable (disable)
evcp62 Trace (evcp62)        = Disable (disable)
evcp63 Trace (evcp63)        = Disable (disable)
evcp64 Trace (evcp64)        = Disable (disable)
evcp65 Trace (evcp65)        = Disable (disable)
evcp66 Trace (evcp66)        = Disable (disable)
evcp67 Trace (evcp67)        = Disable (disable)
evcp68 Trace (evcp68)        = Disable (disable)
evcp69 Trace (evcp69)        = Disable (disable)
evcp70 Trace (evcp70)        = Disable (disable)
evcp71 Trace (evcp71)        = Disable (disable)
evcp72 Trace (evcp72)        = Disable (disable)
evcp73 Trace (evcp73)        = Disable (disable)
evcp74 Trace (evcp74)        = Disable (disable)
evcp75 Trace (evcp75)        = Disable (disable)
evcp76 Trace (evcp76)        = Disable (disable)
evcp77 Trace (evcp77)        = Disable (disable)
evcp78 Trace (evcp78)        = Disable (disable)
evcp79 Trace (evcp79)        = Disable (disable)
evcp80 Trace (evcp80)        = Disable (disable)
evcp81 Trace (evcp81)        = Disable (disable)
evcp82 Trace (evcp82)        = Disable (disable)
evcp83 Trace (evcp83)        = Disable (disable)
evcp84 Trace (evcp84)        = Disable (disable)
evcp85 Trace (evcp85)        = Disable (disable)
evcp86 Trace (evcp86)        = Disable (disable)
evcp87 Trace (evcp87)        = Disable (disable)
evcp88 Trace (evcp88)        = Disable (disable)
evcp89 Trace (evcp89)        = Disable (disable)
evcp90 Trace (evcp90)        = Disable (disable)
evcp91 Trace (evcp91)        = Disable (disable)
evcp92 Trace (evcp92)        = Disable (disable)
evcp93 Trace (evcp93)        = Disable (disable)
evcp94 Trace (evcp94)        = Disable (disable)
evcp95 Trace (evcp95)        = Disable (disable)
evcp96 Trace (evcp96)        = Disable (disable)
evcp97 Trace (evcp97)        = Disable (disable)
evcp98 Trace (evcp98)        = Disable (disable)
evcp99 Trace (evcp99)        = Disable (disable)
evcp100 Trace (evcp100)      = Disable (disable)
ever1 Trace (ever1)          = Disable (disable)
ever2 Trace (ever2)          = Disable (disable)
ever3 Trace (ever3)          = Disable (disable)

```

```

ever5 Trace      (ever5)    = Disable (disable)
ever7 Trace      (ever7)    = Disable (disable)

```

```
>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)
td5 Trace cycle (td5)    = No cycle (none)
td6 Trace cycle (td6)    = No cycle (none)
td7 Trace cycle (td7)    = No cycle (none)
td8 Trace cycle (td8)    = 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)
evep1 Trace      (evep1)    = Disable (disable)
evep2 Trace      (evep2)    = Disable (disable)
evep3 Trace      (evep3)    = Disable (disable)
evep4 Trace      (evep4)    = Disable (disable)
evep5 Trace      (evep5)    = Disable (disable)
evep6 Trace      (evep6)    = Disable (disable)
evep7 Trace      (evep7)    = Disable (disable)
evep8 Trace      (evep8)    = Disable (disable)
ever1 Trace      (ever1)    = Disable (disable)
ever3 Trace      (ever3)    = Disable (disable)
ever5 Trace      (ever5)    = Disable (disable)
ever7 Trace      (ever7)    = Disable (disable)

```

[備考]

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

sswenvコマンド

[書式]

```
sswenv [[!]tsp1] [[!]tsp2] [[!]secon] [[!]secoff] [[!]qualify]
```

[パラメータ]

```
[[!]tsp1:   tsp1をサブスイッチ・オン条件に指定します。!で指定しません。  
[[!]tsp2:   tsp2をサブスイッチ・オフ条件に指定します。!で指定しません。  
[[!]secon:  セクションON条件をサブスイッチ・オン条件に指定します。!で指定しません。  
[[!]secoff: セクションOFFをサブスイッチ・オフ条件に指定します。!で指定しません。  
[[!]qualify: クオリファイ条件をサブスイッチ・オン/オフ条件に指定します。  
!で指定しません。
```

[機能]

サブスイッチの条件の指定をおこないません。

[使用例]

```
sswenv tsp1 tsp2  
tsp1をサブスイッチ・オン条件に、tsp2をサブスイッチ・オフ条件に指定します。
```

s f r コマンド

[書式]

sfr [reg [VAL]]

[パラメータ]

reg: 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):

SFR (R/W):

PAL PALL PALH PAH PAHL PAHH PDL PDLL PDLH PCS PCT PCM PCD PBD PMAL
 PMALL PMALH PMAH PMAHL PMAHH PMDL PMDLL PMDLH PMCS PMCT PMCM PMCD PMBD
 PMCAL PMCALL PMCALH PMCAH PMCAHL PMCAHH PMCDL PMCDLL PMCDLH PMCCS PFCSS
 PMCCT PFCCT PMCCM PMCCD PMCBD CSC0 CSC1 BEC VSWC
 DSAOL DSAOH DDAOL DDAOH DSA1L DSA1H DDA1L DDA1H
 DSA2L DSA2H DDA2L DDA2H DSA3L DSA3H DDA3L DDA3H DBC0
 DBC1 DBC2 DBC3 DADC0 DADC1 DADC2 DADC3 DCHC0
 DCHC1 DCHC2 DCHC3 IMR0 IMR0L
 IMR0H IMR1 IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L IMR3H WDTIC P00IC0
 P00IC1 P00IC4 P00IC5 P10IC6 P10IC7 P01IC0 P01IC1 P01IC2 P01IC3 P11IC4
 P11IC5 P02IC1 P02IC2 P12IC4 P12IC5 P12IC6 P13IC0 P13IC1 P13IC2 P13IC3
 P13IC4 P13IC7 P05IC0 P05IC1 CMICD0 CMICD1 CMICD2 CMICD3 CM10IC0 CM10IC1
 OVVIC0 OVVIC1 OVVIC2 OVVIC3 DMAIC0 DMAIC1 DMAIC2 DMAIC3 SEIC0 SRIC0
 STIC0 SEIC1 SRIC1 STIC1 SEIC2 SRIC2 STIC2 SEIC3 SRIC3 STIC3 ADIC PSC
 ADM0 ADM1 ADM2 ADTS DAOC0 DAOC1
 DAOM P0 P1 P2 P3 P4 P5 PM0 PM1
 PM2 PM3 PM4 PM5 PM0 PM1 PM2
 PM3 PM4 PM5 PM7 PFC0 PFC1 PFC2
 PFC3 PFC4 PFC5 BCT0 BCT1 DWCO
 DWCO BCC ASC BCP LBS LBC0 LBC1 FWC FIC BMC PRC AHC SCR1
 RFS1 SCR3 RFS3 SCR4 RFS4 SCR6 RFS6 CMD0
 TMCDO CMD1 TMC01 CMD2 TMC02 CMD3
 TMC03 TMENC10 CM100 CM101 CC100
 CC101 CCR10 TUM10 TMC10 SESA10 PRM10 TQOCTL0
 TQOCTL1 TQ0IC0 TQ0IC1 TQ0IC2 TQ0OPT0 TQ0CCR0 TQ0CCR1 TQ0CCR2 TQ0CCR3
 TQ0OPT1 TQ0OPT2 TQ0IC3 TQ0DTC HZA0CTL0
 HZA0CTL1 TPOCTL0 TPOCTL1 TP0IC0 TP0IC1 TP0IC2 TP0OPT0 TP0CCR0 TP0CCR1
 TP1CTL0 TP1CTL1 TP1IC0 TP1IC1 TP1IC2 TP1OPT0 TP1CCR0 TP1CCR1
 TP2CTL0 TP2CTL1 TP2IC0 TP2IC1 TP2IC2 TP2OPT0 TP2CCR0 TP2CCR1
 OST0 WDCS WDTM PFCE0 PFCE1 PFCE2
 PFCE3 PFCE5 SYS DTFR0 DTFR1 DTFR2
 DTFR3 PSMR CKC PCC WDRES CORAD0
 CORAD0L CORAD0H CORAD1 CORAD1L CORAD1H CORAD2 CORAD2L CORAD2H CORAD3
 CORAD3L CORAD3H CORCN DT0C DIFC
 DAKW FLPMC UA0CTL0 UA0CTL1 UA0CTL2
 UA0OPT0 UA0STR UA0TX UA1CTL0 UA1CTL1 UA1CTL2 UA1OPT0 UA1STR UA1TX UA2CTL0
 UA2CTL1 UA2CTL2 UA2OPT0 UA2STR UA2TX UA3CTL0 UA3CTL1 UA3CTL2 UA3OPT0
 UA3STR UA3TX INTF0 INTF1 INTF2
 INTF3 INTF5 NMIF INTRO INTR1 INTR2
 INTR3 INTR5 NMIR CBOCTL0 CBOCTL1
 CBOCTL2 CBOSTR CB0TX CB0TXL CB1CTL0 CB1CTL1 CB1CTL2 CB1STR CB1TX CB1TXL

CB2CTL0 CB2CTL1 CB2CTL2 CB2STR CB2TX CB2TXL IIC
IIC SVA IICCL IICX IICF PRSM PRSCM
SFR (W):
PRCMD PFCMD
SFR (R):
ISPR ADCR0 ADCR1 ADCR2 ADCR3
ADCR4 ADCR5 ADCR6 ADCR7 ADCROH ADCR1H ADCR2H ADCR3H ADCR4H ADCR5H ADCR6H
ADCR7H P7 P8 TMD0 TMD1 TMD2 TMD3
STATUS10 TQOCNT TQODTT1 TQODTT2
TQODTT3 TPOCNT TP1CNT TP2CNT
PFS UAORX UA1RX UA2RX UA3RX CBORX
CBORXL CB1RX CB1RXL CB2RX CB2RXL

[機能]

SFRレジスタ値の設定と表示を行います。

[使用例]

sfr P0
P0レジスタの値を表示します。
sfr PM0 0
PM0レジスタに0h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。

td1..td8コマンド

[書式]

```
td{1|2|3|4|5|6|7|8} [ADDR [MASK]] [asid ASID|noasid] [/del]
```

[パラメータ]

td{1|2|3|4|5|6|7|8}: td1 – td8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]

TD1..TD8で指定した領域のアクセスサイクルをトレースに表示させるためには、sswon/sswoffコマンドでそれぞれについて出力するアクセスサイクルの種類を指定する必要があります。トレースに関する詳細は、「付録. C 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] [[!]sss_st_off] [[!]qss_st_off]
      [[!]sss_on_dly1] [[!]sss_off_dly1]
      [[!]add_pc] [lvresume{0..62}] [lvsuspend{1..62}]
      [nonbranchNN] [[!]phold] [[!]once] [tdwidth{4|8|16|24|48}]
      [tclkdiv{1|2|4}] [[!]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。

[[!]sss_st_off: セクションのサブスイッチをOFF状態から開始します。!でON状態から開始します。

[[!]qss_st_off: クオリファイのサブスイッチをOFF状態から開始します。!でON状態から開始します。

[[!]sss_on_dly1: セクションのサブスイッチがONになるタイミングを条件成立 1 命令後に遅延させます。!で即時ONです。

[[!]sss_off_dly1: セクションのサブスイッチがOFFになるタイミングを条件成立 1 命令後に遅延させます。!で即時OFFです。

[[!]add_pc: Start/Match/Overflow時にPCを出力します。!で出力しません。
lvresume{0..62}: 初期値でご使用ください。
lvsuspend{1..62}: 初期値でご使用ください。

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バ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。

tdwidth{4|8|16|24|48}: トレースデータの幅を指定します。
それぞれ、4-bit|8-bit|16-bit|24-bit|48-bitに対応しますが、KIT-V850E/MA3-IEでは、初期設定(48-bit)でのみご使用ください。

tclkdiv{1|2}: CPUの動作クロックに対するトレースクロックの分周率を指定します。
それぞれ、1/1|1/2に対応しますが、KIT-V850E/MA3-IEでは、初期設定(1/1)でのみご使用ください。

[[!]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
      サブスイッチをセクションとクオリファイのオア条件とします。
```

[備考]

トレースに関する詳細は、「付録. C 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. C 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. C 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 )
TD5 Trigger = Disable (!td5 )
TD6 Trigger = Disable (!td6 )
TD7 Trigger = Disable (!td7 )
TD8 Trigger = Disable (!td8 )
Trace Env Settings :
Sub switch = <section> or <qualify> (subor)
Section Sub Switch at force start = on (!sss_st_off)
Qualify Sub Switch at force start = on (!qss_st_off)
Section Sub Switch turn on delay = immediately (!sss_on_dly1)
Section Sub Switch turn off delay = immediately (!sss_off_dly1)
Add PC infomation = Disable (!add_pc)
Non-branch = None (nonbranch0)
Resume Level = 0 (lvresume0)
Suspend Level = 0 (lvsuspend0)
PHOLD = Disable (!phold)
ONCE = Disable (!once)
TDATA Width = 48bit (tdwidth48)
TRCCLK Div. = 1/1 (tclkdiv1)
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
td5 /del
td6 /del
td7 /del
td8 /del

```

[備考]

トレースに関する詳細は、「付録. C 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|5|6|7|8}]
```

[パラメータ]

DELAY = 0..3ffff デレイカウンタ

トリガ成立後にメモリに取り込むフレーム数を16進数で指定します。

[[!]delay: 強制デレイモードを指定します。!で通常のモードの指定に戻ります。強制デレイモードでは、トレース開始後、デレイカウンタ数分のトレースをした時点で強制的にトレースを終了するモードです。このモード中は、トリガイベントは無視されます。

[[!]real: 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(EX10)を指定します。

noext: EX10をトリガとして使用しません。

posi: EX10の立ち上がりエッジをトリガとして指定します。

nega: EX10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。

[[!]td5: トレースデータ条件5(td5)をトリガとして指定します。!で解除します。

[[!]td6: トレースデータ条件6(td6)をトリガとして指定します。!で解除します。

[[!]td7: トレースデータ条件7(td7)をトリガとして指定します。!で解除します。

[[!]td8: トレースデータ条件8(td8)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 3ffff
```

初期値に対し強制デレイモード(delay=on)でトレースを開始します。実行開始直後より、デレイカウンタ値:0x3ffff分の取り込み後、トレースは自動的に終了します。強制デレイモードでは、トリガは無視されます。

```
td1 3ffb800 0
```

```
sswon td1 accs
```

```
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、デレイカウンタ値:0x1ffffサイクル分取り込んだ後、トレースは自動的に終了します。その結果、トリガポイントの前後、約0x20000サイクルがトレースに入ります。sswonコマンドはtd1のサイクルをトレースに表示させるために必要なコマンドです。

```
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、約0x20000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、「付録. C 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を指定します。

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
_MchkFill8+002ah:						
-000005	----	00:00000766	fd8b	bh	_MchkFill8+001ah(00000756h)	1111 JMPS Bcond
-000003	----	00:00000756	1241	add	00000001h, r2	1111 JMPD Bcond
-000003	0001	00:00000758	7e02ffff	addi	ffffh, r2, r15	1111 SUB
-000001	----	00:0000075c	674f0000	st.b	r12, 0000h[r15]	1111 JMPS STORE
*-000001	----	--:04000000	-----0b	[Byte Write (TD1)]		1111 DATAW
-000001	0002	00:00000760	700a	mov	r10, r14	1111 SUB
-000001	0003	00:00000762	525f	add	ffffffffh, r10	1111 SUB
-000001	0004	00:00000764	71e0	cmp	r0, r14	1111 SUB
+000001	----	00:00000766	fd8b	bh	_MchkFill8+001ah(00000756h)	1111 JMPS Bcond
+000003	----	00:00000756	1241	add	00000001h, r2	1111 JMPD Bcond
+000003	0001	00:00000758	7e02ffff	addi	ffffh, r2, r15	1111 SUB

```
> trace ttag1
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
_MchkFill8+002ah:						
-000005	----	00:00000766	fd8b	bh	_MchkFill8+001ah(00000756h)	1111 JMPS Bcond
						time = 000, 002, 151, 164. 7uS
-000003	----	00:00000756	1241	add	00000001h, r2	1111 JMPD Bcond
						time = 000, 002, 151, 164. 8uS
-000003	0001	00:00000758	7e02ffff	addi	ffffh, r2, r15	1111 SUB
-000001	----	00:0000075c	674f0000	st.b	r12, 0000h[r15]	1111 JMPS STORE
						time = 000, 002, 151, 164. 8uS
*-000001	----	--:04000000	-----0b	[Byte Write (TD1)]		1111 DATAW
-000001	0002	00:00000760	700a	mov	r10, r14	1111 SUB
-000001	0003	00:00000762	525f	add	ffffffffh, r10	1111 SUB

```

-000001 0004 00:00000764 71e0    cmp    r0,r14          1111 SUB
+000001 ---- 00:00000766 fd8b    bh     _MchkFill8+001ah(00000756h)
                                           1111 JMPS Bcond
                                time = 000,002,151,164.9uS
+000003 ---- 00:00000756 1241    add    00000001h,r2    1111 JMPD Bcond
                                time = 000,002,151,164.9uS

> trace ttag2
  Cycle Sub Address Code Instruction EXT Stat
_MchkFill8+002ah:
-000005 ---- 00:00000766 fd8b    bh     _MchkFill8+001ah(00000756h)
                                           1111 JMPS Bcond
                                time = 000,000,000,000.0uS
-000003 ---- 00:00000756 1241    add    00000001h,r2    1111 JMPD Bcond
                                time = 000,000,000,000.1uS
-000003 0001 00:00000758 7e02ffff addi   fffffh,r2,r15        1111 SUB
-000001 ---- 00:0000075c 674f0000 st.b   r12,0000h[r15]      1111 JMPS STORE
                                time = 000,000,000,000.0uS
* -000001 ---- --:04000000 -----0b [Byte Write (TD1)] 1111 DATAW
-000001 0002 00:00000760 700a    mov    r10,r14         1111 SUB
-000001 0003 00:00000762 525f    add    ffffffffh,r10    1111 SUB
-000001 0004 00:00000764 71e0    cmp    r0,r14          1111 SUB
+000001 ---- 00:00000766 fd8b    bh     _MchkFill8+001ah(00000756h)
                                           1111 JMPS Bcond
                                time = 000,000,000,000.1uS
+000003 ---- 00:00000756 1241    add    00000001h,r2    1111 JMPD Bcond
                                time = 000,000,000,000.0uS
+000003 0001 00:00000758 7e02ffff addi   fffffh,r2,r15        1111 SUB

```

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)
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. C 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。大きな範囲を指定した場合、処理に時間がかかります。

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ーヘッド時間（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制御用のファームウェアのバージョンを表示します。