

付録 B KIT-V850E2/ME3-TP (—H) 内部コマンド

本書は、KIT-V850E2/ME3-TP (—H) の内部コマンドについて記述しています。これらのコマンドは、デバッガの中でスルーコマンドとして使用できます。スルーコマンドの使用方法は各デバッガのマニュアルを参照ください。

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。

PARTNER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.....	B-1
コマンド書式.....	B-2
アクセス系ブレークポイント : ABP, ABP1, ABP4コマンド.....	B-3
環境設定 : ENV, EMEMSTATコマンド.....	B-4
イベント設定状態の表示 : EMODEコマンド.....	B-5
アクセス系イベントの設定 : EVAコマンド.....	B-6
実行系イベントの設定 : EVEコマンド.....	B-7
イベント統合の設定 : EVTコマンド.....	B-8
ヘルプ : HELPコマンド.....	B-9
INPUT : INB, INH, INWコマンド.....	B-10
初期化 : INITコマンド.....	B-11
割込みマスク : INIT_MASKコマンド.....	B-12
JTAGリード : JREADコマンド.....	B-13
デバッガキャッシュ領域の解除 : NCコマンド.....	B-14
デバッガキャッシュ領域の設定 : NCDコマンド.....	B-15
ソフトブレーク禁止領域の設定 : NSBPコマンド.....	B-16
ソフトブレーク禁止領域の解除 : NSBPDコマンド.....	B-17
強制ユーザ領域の設定 : NROMコマンド.....	B-18
強制ユーザ領域の解除 : NRMDコマンド.....	B-19
OUTPUT : OUTB, OUTH, OUTWコマンド.....	B-20
CPUリセット : RESETコマンド.....	B-21
E. ROMの設定 : ROM1, ROM4コマンド.....	B-22
シーケンシャル条件の設定 : SEQコマンド.....	B-24
サブスイッチの設定 : SSWON, SSWOFFコマンド.....	B-25
サブスイッチ条件の設定 : SSWENVコマンド.....	B-27
SFRアクセス : SFRコマンド.....	B-28
シンボル : SYMFILE, SYMコマンド.....	B-30
トレースの環境設定 : TENVコマンド.....	B-31
トリガポイント : TPコマンド.....	B-32
トレーススイッチポイント : TSP1, TSP2コマンド.....	B-33
トレース条件の参照 : TMODEコマンド.....	B-34
トレースの設定&開始 : TRONコマンド.....	B-35
トレースの強制終了 : TROFFコマンド.....	B-37
トレースの表示 : TRACEコマンド.....	B-38
トレースのファイル書き出し : FTRACEコマンド.....	B-41
トレースデータのディレイ調整 : TDATA_DLYコマンド.....	B-42
バージョン表示 : VERコマンド.....	B-43

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, abp3, abp4 コマンド

[書式]

```
abp [or|seq]
abp{1|2|3|4} [ADDR [AMASK]] [data DATA [DMASK]] [asid ASID|noasid] [aeq|aneq] [deq|dneq]
[exec|read|write|accs] [byte|hword|word|nosize]
abp{1|2|3|4} /del
```

[パラメータ]

abp [or|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。
 seq: abp1発生後、abp2が発生した時にブレークします。
 abp1, 及びabp2は実行アドレス条件で使用してください。

abp{1|2|3|4}: 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。

ata 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。

sid ASID|noasid: 将来の拡張用です。noasidでご使用ください。

eq|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。

eq|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。

xec|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。

yte|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]

4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}]
    [[!]nmi] [[!]reset] [[!]hldrq] [[!]wait]
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。

[[!]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。
[[!]reset: 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。
[[!]hldrq: HLDRQ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。
[[!]wait: 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          << RTE-2000(H)-TPが接続されています。
Rom Probe : (use ememstat command)
Emem Size : (use ememstat command)
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK  = 25MHz (jtag25M)
Verify     = verify off (!verify)
Signals Mask:
NMI        = NO MASK (!nmi)
RESET      = MASK (reset)
HLDRQ      = NO MASK (!hldrq)
WAIT       = NO MASK (!wait)

>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 !nmi1
    RESETをマスクし、NMI1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。
```

e m o d e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 Condition 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。

evt コマンド

[書式]

```

evt {brk|seqclr|seq1|seq2|seq3|seq4|tout|match}
     evp{[1][2][3]..[8]} ever{[1][3][5][7]} evap{[1][2][3]..[6]}
     evar{[1][3][5]} [!]seq]
evt {secon|secoff|qualify}
     evp{[1][2][3]..[8]} ever{[1][3][5][7]} [!]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。
evp{[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 evp1234 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 evp ever evap evar
    ブレーク用のイベントとして指定した、evp ever evap evarを解除します。

```

[備考]

secon, secoff, qualifyに対し、evap, evarパラメータは指定できません。
seqを使用する場合には、seqclr, seq1, seq2, seq3, seq4には、evp/everで指定するイ
ベントのみを指定してください。
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。

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

int_maskコマンド

[書式]

```
int_mask [intNN {enable|disable}]
```

[パラメータ]

intNN: NNが0～63でint0～int63を指定します。

enable|disable

enable: maskを解除します。

disable: maskを設定します。

[機能]

int00～int63の割込みのマスクを指定します。指定した割り込みはCPUに受け付けられなくなります。

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。

[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
```

nsbpコマンド

[書式]

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

[パラメータ]

ADDR: ソフトウェアブ레이크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레이크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레이크を禁止したい領域を指定します。
ブ레이크ポイントを指定した場合、デバッグ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레이크禁止領域に指定します。
```

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


nsbpdコマンド

[書式]

nsbpd [ブロック番号|/all]

[パラメータ]

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

[機能]

nsbpdで指定したソフトウェアブレイク禁止領域を削除します。

[使用例]

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

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

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

n r o mコマンド

[書式]

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

[パラメータ]

ADDR: 強制ユーザ領域の開始アドレスを指定します。
 LENGTH: 強制ユーザ領域のバイト数を指定します。
 指定領域の最小単位は、以下の通りです。
 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、ライトを行いません。
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-ブル
			バス幅	Bit数	
#3	100000 - 3fffff	16-bit	16-bit	32M-Bit	禁止

>rom2 140000 40000 2m rom16 bus16 wren

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

この時、rom2コマンドは発行しないでください。

<備考>

romコマンドで指定した領域における注意事項

rom1..rom4コマンドで指定した範囲へのデバッガからのアクセスは、ツール内部のエミュレーションメモリに対し直接アクセスしています。その結果、プロセッサから正しくROMにアクセスできない状態においても表示は正しく行われますので、デバッグ初期の段階ではjreadコマンド（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。(max4096)

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]}
             {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
 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。
 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)
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_
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. A 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す。

[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):

PAL PALL PAH PAHL PAHH PDH PDHL PDHH PCS PCT PCM PCD PMAL
 PMALL PMAH PMAHL PMAHH PMDH PMDHL PMDHH PMCS PMCT PMCM PMCD PMCAL
 PMCALL PMCAH PMCAHL PMCAHH PMCDH PMCDHL PMCDHH PMCCS PFCGS PMCCT PFCCT
 PMCGM PFCGM PMCCD PFCDH PFCDHL PFCDHH PFCALL CSC0 CSC1 BHC VSWC
 BTSC BTSC L BTSCH CADL CADH CCNT COPR COPRL COPRH DSAOL DSAOH DDAOL
 DDAOH DSA1L DSA1H DDA1L DDA1H DSA2L DSA2H DDA2L DDA2H DSA3L DDA3L
 DDA3H DBC0L DBC0H DBC1L DBC1H DBC2L DBC2H DBC3L DBC3H DADC0 DADC1 DADC2
 DADC3 DC DCHC0 DCHC1 DCHC2 DCHC3 DRST IMRO
 IMROL IMROH IMR1 IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L IMR3H IMR4
 IMR4L IMR4H IMR5 IMR5L IMR5H P1IC0 P1IC1 P2IC1 P2IC2 P2IC3 P2IC4 P2IC5
 P5IC0 P5IC1 P5IC2 P6IC5 P6IC6 P6IC7 PDIC0 PDIC1 PDIC2 PDIC3 PDIC4 PDIC5
 PDIC6 PDIC7 PDIC8 PDIC9 PDIC10 PDIC11 PDIC12 PDIC13 PDIC14 PDIC15 PLIC0
 PLIC1 OVCIC0 OVCIC1 OVCIC2 OVCIC3 OVCIC4 OVCIC5 CCC0IC0 CCC0IC1 CCC1IC0
 CCC1IC1 CCC2IC0 CCC2IC1 CCC3IC0 CCC3IC1 CCC4IC0 CCC4IC1 CCC5IC0 CCC5IC1
 CMDIC0 CMDIC1 CMDIC2 CMDIC3 CC10IC0 CC10IC1 CM10IC0 CM10IC1 OV1IC0
 UD1IC0 CC11IC0 CC11IC1 CM11IC0 CM11IC1 OV1IC1 UD1IC1 DMAIC0 DMAIC1
 DMAIC2 DMAIC3 CS13IC0 COVF3IC0 CS13IC1 COVF3IC1 UREIC0 URIC0 UTIC0
 UIFIC0 UTOIC0 UREIC1 URIC1 UTIC1 UIFIC1 UTOIC1 ADIC USOBIC US1BIC US2BIC
 USP2IC USP4IC RSUMIC PSC ADM0
 ADM1 ADM2 ADTS P1 P2 P5 P6 P7
 PM1 PM2 PM5 PM6 PM7 PMC1 PMC2
 PMC5 PMC6 PMC7 PFC1 PFC2 PFC5
 PFC6 PFC7 BCT0 BCT1 DWCO DWC1 BCC ASC
 BCP LBS FWC FIC BMC PRC SCR1 RFS1 SCR3
 RFS3 SCR4 RFS4 SCR6 RFS6 CMD0
 TMCDO CMD1 TMC1 CMD2 TMC2 CMD3
 TMC3 TMENC10 CM100 CM101 CC100
 CC101 CCR10 TUM10 TMC10 SESA10 PRM10 NCW10 TMENC11
 CM110 CM111 CC110 CC111 CCR11 TUM11 TMC11 SESA11 PRM11 NCW11
 CCC00 CCC01 TMCC00 TMCC01 SESCO NCW0
 CCC10 CCC11 TMCC10 TMCC11 SESC1 NCW1
 CCC20 CCC21 TMCC20 TMCC21 SESC2 NCW2
 CCC30 CCC31 TMCC30 TMCC31 SESC3 NCW3
 CCC40 CCC41 TMCC40 TMCC41 CCC50
 CCC51 TMCC50 TMCC51 IRC IRWE
 DTFR0 DTFR1 DTFR2 DTFR3 CKC CKS
 SSGC DT0C DSEP DIFC0 DIFC1 DIFC2
 DIFC3 UB0CTL0 UB0CTL2 UB0STR
 UB0FIC0 UB0FIC1 UB0FIC2 UB0FIC2L UB0FIC2H UB1CTL0 UB1CTL2 UB1STR
 UB1FIC0 UB1FIC1 UB1FIC2 UB1FIC2L UB1FIC2H PWMCO
 PWMO PWMLO PWMHO PWM1 PWM1L PWM1H INTF1
 INTF2 INTF5 INTF6 INTFAL INTFDH INTFDHL INTFDHH INTR1
 INTR2 INTR5 INTR6 INTRAL INTRDH INTRDHL INTRDHH CSIM30
 CSIC30 SFDB30 SFDB30L SFDB30H SFA30 CSIL30 SFN30 CSIM31
 CSIC31 SFDB31 SFDB31L SFDB31H SFA31 CSIL31 SFN31 UFOCS
 UFOBC UFOE0N UFOE0NA UFOEN UFOENM UFOSDS UFOIMO
 UFOIM1 UFOIM2 UFOIM3 UFOIM4 UFOIDR UFODEND UFOGPR
 UFOMODC UFOA1FN UFOAAS UFOE11M UFOE21M UFOE31M UFOE41M UFOE71M UFOE81M
 UFODSTL UFOE0SL UFOE1SL UFOE2SL
 UFOE3SL UFOE4SL UFOE7SL UFOE8SL UFODSCL
 UFODD0 UFODD1 UFODD2 UFODD3 UFODD4 UFODD5 UFODD6 UFODD7 UFODD8 UFODD9

UFODD10 UFODD11 UFODD12 UFODD13 UFODD14 UFODD15 UFODD16 UFODD17 UFOCIE0
 UFOCIE1 UFOCIE2 UFOCIE3 UFOCIE4 UFOCIE5 UFOCIE6 UFOCIE7 UFOCIE8 UFOCIE9
 UFOCIE10 UFOCIE11 UFOCIE12 UFOCIE13 UFOCIE14 UFOCIE15 UFOCIE16 UFOCIE17
 UFOCIE18 UFOCIE19 UFOCIE20 UFOCIE21 UFOCIE22 UFOCIE23 UFOCIE24 UFOCIE25
 UFOCIE26 UFOCIE27 UFOCIE28 UFOCIE29 UFOCIE30 UFOCIE31 UFOCIE32 UFOCIE33
 UFOCIE34 UFOCIE35 UFOCIE36 UFOCIE37 UFOCIE38 UFOCIE39 UFOCIE40 UFOCIE41
 UFOCIE42 UFOCIE43 UFOCIE44 UFOCIE45 UFOCIE46 UFOCIE47 UFOCIE48 UFOCIE49
 UFOCIE50 UFOCIE51 UFOCIE52 UFOCIE53 UFOCIE54 UFOCIE55 UFOCIE56 UFOCIE57
 UFOCIE58 UFOCIE59 UFOCIE60 UFOCIE61 UFOCIE62 UFOCIE63 UFOCIE64 UFOCIE65
 UFOCIE66 UFOCIE67 UFOCIE68 UFOCIE69 UFOCIE70 UFOCIE71 UFOCIE72 UFOCIE73
 UFOCIE74 UFOCIE75 UFOCIE76 UFOCIE77 UFOCIE78 UFOCIE79 UFOCIE80 UFOCIE81
 UFOCIE82 UFOCIE83 UFOCIE84 UFOCIE85 UFOCIE86 UFOCIE87 UFOCIE88 UFOCIE89
 UFOCIE90 UFOCIE91 UFOCIE92 UFOCIE93 UFOCIE94 UFOCIE95 UFOCIE96 UFOCIE97
 UFOCIE98 UFOCIE99 UFOCIE100 UFOCIE101 UFOCIE102 UFOCIE103 UFOCIE104
 UFOCIE105 UFOCIE106 UFOCIE107 UFOCIE108 UFOCIE109 UFOCIE110 UFOCIE111
 UFOCIE112 UFOCIE113 UFOCIE114 UFOCIE115 UFOCIE116 UFOCIE117 UFOCIE118
 UFOCIE119 UFOCIE120 UFOCIE121 UFOCIE122 UFOCIE123 UFOCIE124 UFOCIE125
 UFOCIE126 UFOCIE127 UFOCIE128 UFOCIE129 UFOCIE130 UFOCIE131 UFOCIE132
 UFOCIE133 UFOCIE134 UFOCIE135 UFOCIE136 UFOCIE137 UFOCIE138 UFOCIE139
 UFOCIE140 UFOCIE141 UFOCIE142 UFOCIE143 UFOCIE144 UFOCIE145 UFOCIE146
 UFOCIE147 UFOCIE148 UFOCIE149 UFOCIE150 UFOCIE151 UFOCIE152 UFOCIE153
 UFOCIE154 UFOCIE155 UFOCIE156 UFOCIE157 UFOCIE158 UFOCIE159 UFOCIE160
 UFOCIE161 UFOCIE162 UFOCIE163 UFOCIE164 UFOCIE165 UFOCIE166 UFOCIE167
 UFOCIE168 UFOCIE169 UFOCIE170 UFOCIE171 UFOCIE172 UFOCIE173 UFOCIE174
 UFOCIE175 UFOCIE176 UFOCIE177 UFOCIE178 UFOCIE179 UFOCIE180 UFOCIE181
 UFOCIE182 UFOCIE183 UFOCIE184 UFOCIE185 UFOCIE186 UFOCIE187 UFOCIE188
 UFOCIE189 UFOCIE190 UFOCIE191 UFOCIE192 UFOCIE193 UFOCIE194 UFOCIE195
 UFOCIE196 UFOCIE197 UFOCIE198 UFOCIE199 UFOCIE200 UFOCIE201 UFOCIE202
 UFOCIE203 UFOCIE204 UFOCIE205 UFOCIE206 UFOCIE207 UFOCIE208 UFOCIE209
 UFOCIE210 UFOCIE211 UFOCIE212 UFOCIE213 UFOCIE214 UFOCIE215 UFOCIE216
 UFOCIE217 UFOCIE218 UFOCIE219 UFOCIE220 UFOCIE221 UFOCIE222 UFOCIE223
 UFOCIE224 UFOCIE225 UFOCIE226 UFOCIE227 UFOCIE228 UFOCIE229 UFOCIE230
 UFOCIE231 UFOCIE232 UFOCIE233 UFOCIE234 UFOCIE235 UFOCIE236 UFOCIE237
 UFOCIE238 UFOCIE239 UFOCIE240 UFOCIE241 UFOCIE242 UFOCIE243 UFOCIE244
 UFOCIE245 UFOCIE246 UFOCIE247 UFOCIE248 UFOCIE249 UFOCIE250 UFOCIE251
 UFOCIE252 UFOCIE253 UFOCIE254 UFOCIE255
 SFR (W) :
 PRCMD UB0TX UB1TX UF0IC0 UF0IC1
 UF0IC2 UF0IC3 UF0IC4 UF0FIC0 UF0FIC1 UFOE0W
 UFOB11 UFOB12 UFOINT1 UFOINT2
 SFR (R) :
 PALH PMALH PMCALH ISPR ADCRO
 ADCROH ADCR1 ADCR1H ADCR2 ADCR2H ADCR3 ADCR3H ADCR4 ADCR4H ADCR5 ADCR5H
 ADCR6 ADCR6H ADCR7 ADCR7H TMD0
 TMD1 TMD2 TMD3 STATUS10 STATUS11
 TMC0 TMC1 TMC2 TMC3 TMC4 TMC5
 LOCKR UB0RXAP UB0RX UB0FIS0 UB0FIS1
 UB1RXAP UB1RX UB1FIS0 UB1FIS1
 SIRB30 SIRB30L SIRB30H SIRB31
 SIRB31L SIRB31H UFOCLR UFOSET
 UFOEPS0 UFOEPS1 UFOEPS2 UFOIS0 UFOIS1 UFOIS2 UFOIS3 UFOIS4 UFODMS0
 UFODMS1 UFOMODS UFOASS UFOEOR
 UFOEOL UFOEOST UFOB01 UFOB01L UFOB02 UFOB02L UFOADRS
 UFOCNF UFO1F0 UFO1F1 UFO1F2 UFO1F3 UFO1F4

[機能]

SFRレジスタ値の設定と表示を行います。

[使用例]

```

sfr PAL
PALレジスタの値を表示します。
sfr PAL 0
PAL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。
```

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}] [[!]trcce] [[!]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。

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。

tclkdiv{1|2|4}: 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|1/4に対応します。

[[!]trcce: 初期値 (trcce)でご使用ください。

[[!]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

```

サブスイッチは、セクションとクオリファイのオアでトレースします。

[備考]

トレースに関する詳細は、「付録. A トレース機能の詳細」を参照ください。

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. A 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. A トレース機能の詳細」を参照ください。

t m o d eコマンド

[書式]

tmode

[パラメータ]

なし

[機能]

トレースの設定状態を表示します。

[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)
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         = 1 (lvsuspend1)
PHOLD                 = Disable (!phold)
ONCE                  = Disable (!once)
TDATA Width           = 16bit (tdwidth16)
TRCCLK Div.           = 1/2 (tclkdiv2)
Debug Mode            = Disable (!debug)
Trace Switch Point Settings:
  Address  ASID
tsp1 /del
tsp2 /del
Trigger Point Settings:
  Address  ASID
tp /del
```

[備考]

トレースに関する詳細は、「付録. A トレース機能の詳細」を参照ください。

tronコマンド

[書式]

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

[パラメータ]

DELAY = 0..xxxx: デレイカウンタ

トリガ成立後にメモリに取り込むフレーム数を16進数で指定します。

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

[[!]real: 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の立ち下がエッジをトリガとして指定します。

[機能]

トレースの設定とトレースバッファをクリアし、トレースの取り込みを開始します。

[使用例]

```
tron
```

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

```
tron delay 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:0x3ffd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。

```
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. A トレース機能の詳細」を参照ください。

t r o f fコマンド

[書式]

troff

[パラメータ]

なし

[機能]

トレースの取り込みを強制的に終了します。

traceコマンド

[書式]

```
trace [POS] [all|pc|data] [asm|ttag1|ttag2] [subNN]
```

[パラメータ]

POS=±0. .xxxx: トリガサイクル近辺または終了サイクルを0として、トレースの表示開始位置を16進数で指定します。

all|pc|data: 取り込んだトレース情報の中から選択して表示するサイクルの指定します。

all: 全てのサイクル

pc: 実行サイクルのみ

data: データサイクルのみ

asm|ttag1|ttag2表示種別を指定します。

Asm: アセンブラ表示のみ

ttag1: アセンブラ表示+絶対時間でのタイムタグ表示

ttag2: アセンブラ表示+相対時間でのタイムタグ表示

subNN: 実際に取り込まれる一つの分岐情報から連続して逆アセンブルする命令数を16進数で指定します。初期値は80h(ex:sub80)です。

[機能]

トレースバッファの内容を表示します。

トレース中にこのコマンドを発行した場合、強制的に取り込みを終了します。

[表示例]

```
> trace asm -10
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
H-000011	----	00:00001e32	500f	mov r15, r10	1111	JMPD Bcond
-000011	0001	00:00001e34	7a5f	add ffffffffh, r15	1111	SUB
-000011	0002	00:00001e36	51e0	cmp r0, r10	1111	SUB
H-00000e	----	00:00001e38	ddcb	bh 00001df0h	1111	JMPS Bcond
H-00000b	----	00:00001df0	57e90001	ld.hu 0000h[r9], r10	1111	JMPD Bcond
-00000b	0001	00:00001df4	880a	mov r10, r17	1111	SUB
-00000b	0002	00:00001df6	8002	mov r2, r16	1111	SUB
-00000b	0003	00:00001df8	89f0	cmp r16, r17	1111	SUB
-000008	----	00:04020000	----4d53	[Hword Read]	1111	DATAR
H-000002	----	00:00001dfa	15a2	bz 00001e1eh	1111	JMPSS Bcond
* +000000	----	---	-----	[Trace Trigger]	1111	MATCH
H+000001	----	00:00001e1e	4a42	add 00000002h, r9	1111	JMPD Bcond
+000001	0001	00:00001e20	1241	add 00000001h, r2	1111	SUB
+000001	0002	00:00001e22	00c2	zxx r2	1111	SUB
+000001	0003	00:00001e24	6002	mov r2, r12	1111	SUB
+000001	0004	00:00001e26	5e80ffff	ori ffffh, r0, r11	1111	SUB
+000001	0005	00:00001e2a	61eb	cmp r11, r12	1111	SUB
H+000003	----	00:00001e2c	05ba	bnz 00001e32h	1111	JMPSS Bcond
H+000005	----	00:00001e32	500f	mov r15, r10	1111	JMPD Bcond
+000005	0001	00:00001e34	7a5f	add ffffffffh, r15	1111	SUB

```
> trace ttag1
```

Cycle	Sub	Address	Code	Instruction	EXT	Stat
H-000011	----	00:00001e32	500f	mov r15, r10	1111	JMPD Bcond
				time = 000, 000, 002, 027. 5uS		
-000011	0001	00:00001e34	7a5f	add ffffffffh, r15	1111	SUB
-000011	0002	00:00001e36	51e0	cmp r0, r10	1111	SUB
H-00000e	----	00:00001e38	ddcb	bh 00001df0h	1111	JMPS Bcond
				time = 000, 000, 002, 027. 6uS		
H-00000b	----	00:00001df0	57e90001	ld.hu 0000h[r9], r10	1111	JMPD Bcond
				time = 000, 000, 002, 027. 6uS		
-00000b	0001	00:00001df4	880a	mov r10, r17	1111	SUB
-00000b	0002	00:00001df6	8002	mov r2, r16	1111	SUB
-00000b	0003	00:00001df8	89f0	cmp r16, r17	1111	SUB
-000008	----	00:04020000	----4d53	[Hword Read]	1111	DATAR
				time = 000, 000, 002, 027. 6uS		

```

H-000002 ---- 00:00001dfa 15a2    bz      00001e1eh      1111 JMPSS Bcond
                        time = 000,000,002,027.7uS
* +000000 ---- -:----- ----- [Trace Trigger]      1111 MATCH
                        time = 000,000,002,027.7uS
H+000001 ---- 00:00001e1e 4a42    add    00000002h, r9   1111 JMPD  Bcond
                        time = 000,000,002,027.7uS
+000001 0001 00:00001e20 1241    add    00000001h, r2   1111 SUB
+000001 0002 00:00001e22 00c2    zxh    r2              1111 SUB
+000001 0003 00:00001e24 6002    mov    r2, r12         1111 SUB
+000001 0004 00:00001e26 5e80ffff ori    fffffh, r0, r11    1111 SUB
+000001 0005 00:00001e2a 61eb    cmp    r11, r12        1111 SUB
H+000003 ---- 00:00001e2c 05ba    bnz    00001e32h      1111 JMPSS Bcond
                        time = 000,000,002,027.7uS
H+000005 ---- 00:00001e32 500f    mov    r15, r10        1111 JMPD  Bcond
                        time = 000,000,002,027.8uS
+000005 0001 00:00001e34 7a5f    add    ffffffffh, r15  1111 SUB
+000005 0002 00:00001e36 51e0    cmp    r0, r10         1111 SUB
H+000007 ---- 00:00001e38 ddcbb  bh     00001df0h         1111 JMPS  Bcond
                        time = 000,000,002,027.8uS

```

> trace ttag2

Cycle	Sub	Address	Code	Instruction	EXT	Stat
H-000011	----	00:00001e32	500f	mov r15, r10	1111	JMPD Bcond
				time = 000,000,000,000.0uS		
-000011	0001	00:00001e34	7a5f	add ffffffffh, r15	1111	SUB
-000011	0002	00:00001e36	51e0	cmp r0, r10	1111	SUB
H-00000e	----	00:00001e38	ddcbb	bh 00001df0h	1111	JMPS Bcond
				time = 000,000,000,000.1uS		
H-00000b	----	00:00001df0	57e90001	ld.hu 0000h[r9], r10	1111	JMPD Bcond
				time = 000,000,000,000.0uS		
-00000b	0001	00:00001df4	880a	mov r10, r17	1111	SUB
-00000b	0002	00:00001df6	8002	mov r2, r16	1111	SUB
-00000b	0003	00:00001df8	89f0	cmp r16, r17	1111	SUB
-000008	----	00:04020000	----4d53	[Hword Read]	1111	DATAR
				time = 000,000,000,000.0uS		
H-000002	----	00:00001dfa	15a2	bz 00001e1eh	1111	JMPSS Bcond
				time = 092,216,911,462.5uS		
* +000000	----	-:-----	-----	[Trace Trigger]	1111	MATCH
				time = 000,000,000,000.0uS		
H+000001	----	00:00001e1e	4a42	add 00000002h, r9	1111	JMPD Bcond
				time = 000,000,000,000.0uS		
+000001	0001	00:00001e20	1241	add 00000001h, r2	1111	SUB
+000001	0002	00:00001e22	00c2	zxh r2	1111	SUB
+000001	0003	00:00001e24	6002	mov r2, r12	1111	SUB
+000001	0004	00:00001e26	5e80ffff	ori fffffh, r0, r11	1111	SUB
+000001	0005	00:00001e2a	61eb	cmp r11, r12	1111	SUB
H+000003	----	00:00001e2c	05ba	bnz 00001e32h	1111	JMPSS Bcond
				time = 000,000,000,000.0uS		
H+000005	----	00:00001e32	500f	mov r15, r10	1111	JMPD Bcond
				time = 000,000,000,000.1uS		

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)
JMPSS ◇	JMPSS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. A トレース機能の詳細」を参照ください。

f t r a c eコマンド

[書式]

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

[パラメータ]

statpos: ファイルに書き出すトレースポジションの開始位置

endpos: ファイルに書き出すトレースポジションの終了位置

filename:

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

```
[all|pc|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。

t d a t a _ d l yコマンド

[書式]

```
tdata_dly [off|small|medium|large]
```

[パラメータ]

off:	補正しません。(初期値)
small:	最小の補正をします。
medium:	中程度の補正をします。
large:	最大の補正をします。

[機能]

トレースクロックに対するトレースデータのセットアップ時間を調整するためのコマンドです。セットアップ時間はoffが一番小さく、largeが一番大きくなります。なお、実際のセットアップ値は使用するRTE-xxxx-TP本体やケーブルに依存しますので、各本体の仕様を確認ください。

[補足]

通常は初期値から変更する必要はありませんが、CPUやボードの状態によっては調整が必要になる場合があります。

verコマンド

[書式]
ver

[パラメータ]
なし

[機能]
I C E制御用のファームウェアのバージョンを表示します。