

付録 . A K I T - N B 8 5 E - T P 内部コマンド

本書は、K I T - N B 8 5 E - T Pの内部コマンドについて記述しています。これらのコマンドは、デバッガの中でスルーコマンドとして使用できます。スルーコマンドの使用方法は各デバッガのマニュアルを参照ください。

PARTNER/Winの場合

- > & << スルーコマンドへの移行します。
- > #ENV << 内部コマンドの入力です。
- > & << スルーコマンドモードを終了します。

GHS - Multiの場合

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。

コマンド一覧

コマンド一覧.....	1
コマンド書式.....	2
アクセス系ブレークポイント : A B P , A B P 1 , A B P 2 コマンド.....	3
環境設定 : E N V コマンド.....	4
外部入力ブレーク : E X T B R K コマンド.....	5
ヘルプ : H E L P コマンド.....	6
I N P U T : I N B , I N H , I N W コマンド.....	8
初期化 : I N I T コマンド.....	9
J T A G リード : J R E A D コマンド.....	10
デバッガキャッシュ領域の解除 : N C コマンド.....	11
デバッガキャッシュ領域の設定 : N C D コマンド.....	12
ソフトブレーク禁止領域の設定 : N S B P コマンド.....	14
ソフトブレーク禁止領域の設定 : N S B P D コマンド.....	15
強制ユーザ領域の設定 : N R O M コマンド.....	16
強制ユーザ領域の設定 : N R O M D コマンド.....	17
O U T P U T : O U T B , O U T H , O U T W コマンド.....	18
C P U リセット : R E S E T コマンド.....	18
E . R O M の設定 : R O M コマンド.....	19
S F R アクセス : S F R コマンド.....	20
シンボル : S Y M F I L E , S Y M コマンド.....	21
トリガポイント : T P コマンド.....	22
トレーススイッチポイント : T S P 1 , T S P 2 コマンド.....	23
トレースデータ条件 : T D 1 , T D 2 コマンド.....	24
トレースの設定 & 開始 : T R O N コマンド.....	25
トレースの強制終了 : T R O F F コマンド.....	28
トレースの表示 : T R A C E コマンド.....	29
バージョン表示 : V E R コマンド.....	32

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-NB85E-TP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。

[入力例]

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

envコマンド

[書式]

```
env [[!]auto] [[!][verify]] [[!]reset] [[!]stopz] [[!]hldrq]
[jtag{25|12|5|2|1|500|250|100}] [[!]nmi0] [[!]nmi1] [[!]nmi2]
[rtrcb{0|25|50|75}] [nrtrcb{12|25|37|50}] [64m|256m]
[romless|single0|single1] [d0|d1|d2|dauto] [i0|i1|i2|iauto]
```

[パラメータ]

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

[!]verify:メモリへの書き込み時にリードアウトしてベリファイするかどうか指定します。!はベリファイしないを意味します。

備考: ROMをエミュレーションしている領域に対しても、CPUからアクセス(jread相当)しますので、ダウンロード時のテストにも有効です。但し、処理速度が遅くなります。

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

[!]stopz: STOPZ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: VAREQ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。

[!]nmi{00|01|02}: INT00-03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。

jtag{25|12|5|2|1|500|250|100}: 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、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は、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。
尚、RTE-100-TPでは、jtag25, jtag12以外のパラメータは無効です。
初期値は、rte4win32のバージョンによります。
Ver. 5.02以前: 12.5MHz固定です。
Ver. 5.03以降: 動作する最高周波数に自動的に設定します。

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

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

64m|256m: CPUのアドレスモードを指定します。

64m: 64Mモードを指定します。

256m: 256Mモードを指定します。

romless|single0|single1: CPUの動作モードを指定します。

single0m: シングルモード0 (0番地から内蔵ROM) を指定します。

single1: シングルモード1 (100000h番地から内蔵ROM) を指定します。

Romless: ロムレスモードを指定します。

d0|d1|d2|dauto: データキャッシュの指定します。

d0: データキャッシュなしを指定します。

d1: ダイレクトマップのキャッシュを指定します。

d2: 2-WAYのキャッシュを指定します。

dauto: NB85E-TEGの場合に指定します。自動設定します。

i0|i1|i2|iauto: 命令キャッシュの指定します。

i0: 命令キャッシュなしを指定します。

i1: ダイレクトマップのキャッシュを指定します。

i2: 2-WAYのキャッシュを指定します。

iauto: NB85E-TEGの場合に指定します。自動設定します。

備考: dauto, iautoの指定は、NB85E-TEGを使用した評価ボードに限りです。通常は、実際に実装されているキャッシュにあった指定をしてください。

[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。

```

Probe:
Unit      : RTE-1000-TP          << 接続している本体を表示します。
Rom Probe : Extend Type        << 接続しているROMプローブをタイプを表示します。
Emem Size : 32Mbyte           << 実装しているエミュレーションメモリの容量を表示します。
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK = 12.5MHz (jtag12)
Verify     = verify off (!verify)
CPU Mode   = romless (romless)   << rte4win32の設定に依存します。
Space      = 64M Byte Mode (64m) << rte4win32の設定に依存します。
Signals Mask:
NMIO       = NO MASK (!nmi0)
NMI1       = NO MASK (!nmi1)
NMI2       = NO MASK (!nmi2)
RESET      = NO MASK (!reset)
HLDRQ      = NO MASK (!hldrq)
STOPZ      = NO MASK (!stopz)
Trace Buffer Usage Settings:
Realtime   <= 0% (rtrcb0)
None Realtime>= 12% (nrtrcb12)
Trace UNIT:
Cotrol Unit = Enable
Event Unit  = Enable
Execute Event Number = 8
Access Event Number = 4
Sequence Event Number = 1
Sequence Counter Bit = 12
Cache Mode:
Data       = 2Way (d2)
Instruction = Auto Detect (iauto)

```

[入力例]

```

env reset !nmi
    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。

```

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-1pin(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。

備考：このコマンドは、RTE-100-TPでは無効です。
この機能を使用する場合は、JTAG/N-WireコネクタのA12ピンがCPUのDBINTに接続されていなければなりません。

helpコマンド

[書式]

help [command]

[パラメータ]

command: コマンド名を指定します。

コマンド名を省略した場合、コマンドの一覧が表示されます。

[機能]

各コマンドのヘルプメッセージを表示します。

[使用例]

help map

mapコマンドのヘルプを表示します。

inb, inh, inwコマンド

[書式]

inb [ADDR]

inh [ADDR]

inw [ADDR]

[パラメータ]

ADDR: 入力ポートのアドレスを16進数で指定します。

[機能]

inb, inh, inwは、アクセスサイズを区別して、リードを行います。

inbはバイト、inhはハーフ・ワード、inwはワード単位でアクセスします。

[使用例]

inb 1000

1000Hからバイト(8-bit)でリードします。

inh 1000

1000Hからハーフワード(16-bit)でリードします。

inw 1000

1000Hからワード(32-bit)でリードします。

initコマンド

[書式]

init

[パラメータ]

なし

[機能]

K I T - N B 8 5 E - T P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。

jreadコマンド

[書式]

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

[パラメータ]

ADDR: アドレスを16進数で指定します。

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

[機能]

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

[使用例]

```
jread 100000 100
```

100000hから100hバイトをJTAG経由で読み出します。

ncコマンド

[書式]

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

[パラメータ]

ADDR: メモリキャッシュの除外エリアの開始アドレスを指定します。

LENGTH: メモリキャッシュの除外エリアのバイト数を指定します。

デフォルト値 3 2 バイト、最少値 3 2 バイト

[機能]

KIT-NB85E-TPではメモリ参照の高速化を図るため、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には実際にはメモリをリードしません。I/Oを割り付けている空間では、このキャッシュ機能は実際の動作と矛盾しますので、このコマンドで除外エリアとして指定してください。メモリキャッシュの除外エリアは最大8ブロック指定でき、最少のブロックサイズは32バイトです。尚、ffff000h - ffffffffhと3fff000h - 3fffffffhは、内蔵のsfr領域ですので、初期値として除外エリアに指定されています。

[使用例]

```
nc 10000 100
```

10000h番地から100バイトの領域をメモリキャッシュの除外エリアに指定します。

```
>nc 100000 100
```

```
No Memory Cache Area
```

```
No. Address Length
```

```
1 00100000 00000100
```

```
2 0ffff000 00001000
```

```
3 03fff000 00001000
```

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。

[使用例]

ncd 1

ブロック番号 1 をメモリキャッシュの除外エリアから削除します。

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

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

nsbpコマンド

[書式]

nsbp [[ADDR [LENGTH]]]

[パラメータ]

ADDR: ソフトウェアブレ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

nromコマンド

[書式]

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

[パラメータ]

ADDR: 強制ユーザ領域の開始アドレスを指定します。
LENGTH: 強制ユーザ領域のバイト数を指定します。
指定領域の最小単位は、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]

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他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。

指定領域は、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。
通常は、指定する必要はありません。

[使用例]

```
nrom 0 2000  
0h番地から2000バイトを強制ユーザ領域に指定します。
```

```
>nrom 0 1000  
No. Address Length  
1 00000000 00001000
```

```
>nrom 10000 100  
No. Address Length  
1 00000000 00001000  
2 00010000 00000100
```

nromdコマンド

[書式]

nromd [ブロック番号|/all]

[パラメータ]

ブロック番号: 削除する強制ユーザ領域のブロック番号を指定します。

/all : 全ての強制ユーザ領域のブロックを削除します。

[機能]

nromで指定した強制ユーザ領域を削除します。

[使用例]

ncd 1

ブロック番号 1 を強制ユーザ領域から削除します。

```
>nrom 10000 8000
```

No.	Address	Length
1	00000000	00001000
2	00010000	00008000

```
>nromd 1
```

No.	Address	Length
1	00010000	00008000

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1バイトライトします。
outh 1000 1234
    1000Hへハーフワードデータ：1234hを2バイトライトします。
outw 1000 12345678
    1000Hへワードデータ：12345678hを4バイトライトします。
```

resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

KIT-NB85E-TPの対象エミュレーションCPUをリセットします。

romコマンド

[書式]

```
rom [ADDRESS [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
[bus8|bus16|bus32]
```

[パラメータ]

ADDR [LENGTH]: エミュレーションする領域を指定します。

ADDR: 開始アドレスを指定します。エミュレートするROMの最下位のアドレス (ROMのパウンダリ)に合致していない場合、エラーになります。

LENGTH: エミュレートするROMのバイト数を指定します。(4バイトの境界で指定)

512k|1m|2m|4m|8m|16m|32m|64m|128m|256m: 1本のROMプローブでエミュレートするROMのBit容量を指定します。512K-bitから256M-bitまでの値が指定できます。例えば、27C1024の場合は、1Mを指定します。

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

8bitと16bitが指定できます。DIP32-ROMケーブルを使用する場合はrom8、DIP-40/42-ROMケーブル、16bit-標準ROMケーブルを使用する場合は、rom16を指定します。

bus8|bus16|bus32: エミュレートするシステムの中でのROMのバスサイズを指定します。

8bit,16bit,32bitが指定できます。

[機能]

ROMのエミュレーション環境の設定を行います。設定は変更が必要なパラメータだけを入力してください。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、LENGTH = 0 (使用しない) になっています。

[入力例]

```
rom 100000 40000 1m rom16 bus16
```

27C1024(1M-bitの16bit-ROM)を100000hから256Kバイト(40000h)エミュレートします。
この場合、結果的に16bit-romを2個使用してエミュレートします。

```
rom 0 40000 2m rom rom16 bus32
```

27c2048(2M-bitの16bit-ROM)を0x0から256Kバイト(40000h)エミュレートします。
この場合、結果的に16bit-ROMを1個使用してエミュレートします。

< 備考 >

romコマンドで指定した範囲へのアクセスは、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ます。したがって、表示時、正しく見えていても、プロセッサから正しくROMにアクセスできない場合もあります。この場合、jreadコマンドを使用して確認するか、envコマンドでverifyをONにして書き込み(ダウンロード)を行って下さい。そうすることにより、エミュレーションしているROMの内容をCPUのバスを介して読み出すことでチェックできます。

s f rコマンド

[書式]

sfr [reg [VAL]]

[パラメータ]

VAL: S F Rのレジスタ値を16進数で指定します。

reg: S F Rレジスタ名を指定します。

レジスタとして使用できる名称は以下の通りです。

リード・ライトレジスタ:

CSC0 CSC1 BPC BSC BEC BHC VSWC
 DSA0L DSA0H DDA0L DDA0H DSA1L DSA1H DDA1L DDA1H
 DSA2L DSA2H DDA2L DDA2H DSA3L DSA3H DDA3L DDA3H
 DBC0 DBC1 DBC2 DBC3
 DADC0 DADC1 DADC2 DADC3 DCHC0 DCHC1 DCHC2 DCHC3
 DRST IMR0 IMR1 IMR2 IMR3
 PICO..PIC63
 PSC BCT0 BCT1 DWCO DWC1 BCC ASC PRC RWC
 DRC0 SCRO RFC0 RFS0 DRC1 SCR1 RFC1 RFS1
 DRC2 SCR2 RFC2 RFS2 DRC3 SCR3 RFC3 RFS3
 DRC4 SCR4 RFC4 RFS4 DRC5 SCR5 RFC5 RFS5
 DRC6 SCR6 RFC6 RFS6 DRC7 SCR7 RFC7 RFS7
 ICC ICI ICD

ライトオンリーレジスタ:

PRCMD

リードオンリーレジスタ:

DDIS ISPR

[機能]

S F Rレジスタ値の設定と表示を行います。

[使用例]

sfr PICO

PICOレジスタの値を表示します。

sfr PICO 2

PICOレジスタに2hを設定します。

symfile, symコマンド

[書式]

symfile FILENAME

sym [NAME]

[パラメータ]

symfile: ファイル名を指定します。

sym: シンボルの先頭文字列を指定します。

[機能]

symfile コマンドは、FILENAMEで指定したelfファイルからシンボルを読み込みます。

対象となるのはグローバルシンボルだけです。

Symコマンドは、読み込んだシンボルの表示（最大30個）をできます。

[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。

tpコマンド

[書式]

tp [ADDR]

[パラメータ]

ADDR: 偶数アドレスを16進数で指定します。(A0は、常に0に補正されます)

[機能]

トレースのトリガポイントを指定します。

トレースは、トリガポイントを基点にしてその前後の実行状態を取り込むことができます。

(トリガポイントの使用方法はtronコマンドを参照ください)

[使用例]

tp 100000

100000hの命令実行をトリガポイントとして指定します。

[注意事項]

tronコマンドでdelay modeが指定されている場合、トリガポイントの指定は無視されます。

この場合、tron !delayと入力してdelay modeを解除してください。

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点あるトレースのスイッチ・ポイント(アドレス)を指定します。

指定したスイッチ・ポイントにより、トレース情報の取り込み条件をかえることができます。

(取り込み条件の指定方法は、tronコマンドを参照ください)

[使用例]

```
tsp1 100000
```

100000hの命令実行をトレースのスイッチ・ポイントとして指定します。

[備考]

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

td1, td2 コマンド

[書式]

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

[パラメータ]

td{1|2}: td1または、td2の条件指定に先立ち入力します。

ADDR: アドレスを指定します。

MASK: アドレスのマスクデータを16進数で指定します。1のビットは、比較の対象になりません。有効なのは、bit9-bit2のみです。

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

/del: 指定したアドレスを解除します。

[機能]

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

[使用例]

td1 100000 ff

1000xxh番地のアクセスサイクルをトレースに取り込みます。

tronコマンド

[書式]

```
tron [DELAY] [!]delay [!]real [!]force { [!]evttcrs1 [!]evttcrs2 |
 [!]evttcr} [tr1_{[0]..[h]}|tr1_all] [tr2_{[0]..[h]}|tr2_all]
 [!]clock2 [!]stop [noext|posi|nega] [!]td1 [!]td2 [!]debug
```

[パラメータ]

DELAY = 0..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: トレースの強制開始を指定します。!で強制開始を解除です。その場合は、tsp1の条件によります。

[!]evttcrs1 [!]evttcrs2|[!]evttcr: 初期値(!)の状態でご使用ください。

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

tr1_{[0]..[h]}: 0:Interrupt, 1:Exception, 2:RET1, 3:JMP, 4:JR, 5:JARL,
6:Condition Jump(not taken), 7:Condition Jump(taken),
8:CALLT, 9:SWITCH, a:DISPOSE, b:CTRET,
c:td1 read cycle, d:td1 write cycle,
e:td2 read cycle, f:td2 write cycle,
g:tp, h:evt_match

tr1_all: 全てのトレース情報を取り込みます。

Tr2_{[0]..[h]}|tr2_all: tsp2のスイッチポイント以降に取り込むトレース情報を指定します。

Tr2_{[0]..[h]}: 0:Interrupt, 1:Exception, 2:RET1, 3:JMP, 4:JR, 5:JARL,
6:Condition Jump(not taken), 7:Condition Jump(taken),
8:CALLT, 9:SWITCH, a:DISPOSE, b:CTRET,
c:td1 read cycle, d:td1 write cycle,
e:td2 read cycle, f:td2 write cycle,
g:tp, h:evt_match

tr2_all: 全てのトレース情報を取り込みます。

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

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

noext|nega|posi: トリガとして外部入力端子(EXI0)を指定します。

noext: EXI0をトリガとして使用しません。

posi: EXI0の立ち上がりエッジをトリガとして指定します。

nega: EXI0の立ち下がエッジをトリガとして指定します。

[!]td1: トレースデータ条件1(td1)をトリガとして指定します。!で解除します。

[!]td2: トレースデータ条件2(td2)をトリガとして指定します。!で解除します。

備考: [!]td1 [!]td2は、RTE-100-TPでは、無効です。

td1とtd2の条件が重複するサイクルを指定している場合、トリガの条件は、td1を指定してください。td2では、トリガがかからない場合があります。

[!]debug: 常時初期値(!debug)でご使用ください。

[機能]

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

[使用例]

delayモードでtron直後より、無条件に1ffffサイクル分トレースします。

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

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

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

>tron !delay ffff                << トレース開始
Trace Settings:
Delay Count   = 000ffff
Trace Mode    = Real Time (real)
Start Mode    = Force Start (force)
Delay Mode    = Disable (!delay)
Event trcs1   = Disable (!evttrcs1)
Event trcs2   = Disable (!evttrcs2)
Event trcr    = -----
Sampling cond1= tr1_0123456789abcdefgh
Sampling cond2= tr2_0123456789abcdefgh
Trace Clock   = VBCLK (!clock2)
STOP Mode     = Disable (!stop)
Ext Trigger   = Disable (noext)
TD1 Trigger   = Disable (!td1)
TD2 Trigger   = Disable (!td2)
Debug Mode    = Disable (!debug)
```

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

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

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

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

troffコマンド

[書式]

troff

[パラメータ]

なし

[機能]

トレースの取り込みを強制的に終了します。

traceコマンド

[書式]

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

[パラメータ]

POS= \pm 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

備考：ttag1|ttag2の指定は、RTE-100-TPでは無効です。

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

[機能]

トレースバッファの内容を表示します。

トレース中にこのコマンドを発行した場合、強制的に取り込みを終了します。

[表示内容]

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

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

```

000016 ---- 00:001055a2 16400380 movehi 0380h,zero,r2    1111 JMPD  BcondNT
                                time = 000,001,448,268.9uS

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

```

Cycle: トレースバッファ内の位置を16進数で相対的に表示しています。トリガポイント位置の近辺または、トレースの最終フレームを0としています。

Sub: 分岐や実行命令数などの情報から解析して生成したサイクルの番号です。

Address: 実行アドレスまたは、バスサイクルのアドレスを表示します。

Code: 命令コードまたは、バスサイクルのデータを表示します。

Instruction: 命令の二モニクまたは、バスの種類を表示します。

EXT: 外部入力端子EXI3..0の状態をビット列で表示します。

Stat: 表示にもとになるトレースパケットの種別を表示します。

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

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

NMI/INT	割り込みの発生によるもの
EXP/TRAP	例外の発生によるもの
RETI	当該命令によるもの
JMP	当該命令によるもの
JR	当該命令によるもの
JARL	当該命令によるもの
BcondNT	当該命令によるもの
Bcond	当該命令によるもの
CALLT	当該命令によるもの
SWITCH	当該命令によるもの

DISPOSE	当該命令によるもの
CTRET	当該命令によるもの
FSTART	トレースの強制スタート

*: トリガポイント (多少ずれる場合があります)

time = タイムタグの表示

備考：タイムタグは、CPUから分岐情報が出力された時点のものです。分岐情報の出力は、実際の実行時間に対し遅れがあり、この遅れは一定ではありません。したがって、測定値には潜在的な誤差があります。また、実行直後の測定値は不定ですので無視してください。

verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

KIT-NB85E-TPのバージョンを表示します。