

## 付録 . A RTE - V 8 5 2 - I E 内部コマンド

本書は、RTE - V 8 5 2 - 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、各デバッグモニタの中でスルーコマンドとして使用できます。それぞれの使用方法はそれぞれのデバッグモニタのマニュアルを参照下さい。

(例) 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。

### コマンド一覧

アクセス系ブレークポイント設定 : ABP , ABP 1 , ABP 2 , ABP 3 , ABP 4 コマンド	....A3
アクセス系トレーストリガ設定 # 1 : ATP , ATP 1 , ATP 2 コマンド	.....A4
アクセス系トレーストリガ設定 # 2 : ATP 3 コマンド	.....A5
環境設定 : ENV コマンド	.....A7
ヘルプ : HELP コマンド	.....A8
初期化 : INIT コマンド	.....A9
メモリマップ指定 : MAP コマンド	.....A10
キャッシュ領域の解除 : NC コマンド	.....A11
キャッシュ領域の指定 : NCD コマンド	.....A12
リセットコマンド : RESET コマンド	.....A13
リアルタイムラムモニタのベース : RRMB コマンド	.....A14
リアルタイムラムモニタの読み出し : RRM コマンド	.....A15
SFR : SFR コマンド	.....A16
シンボルファイルの読み込み : SYMFILE , SYM コマンド	.....A17
実行時間の表示 : TIME コマンド	.....A18
実行系とりがポイントの指定 : TP , TP 1 , TP 2 コマンド	.....A19
トレースの開始と設定 : TRON コマンド	.....A20
トレースの強制終了 : TROFF コマンド	.....A21
トレース結果の表示 : TRACE コマンド	.....A22
バージョンの表示 : VER コマンド	.....A24

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下さい。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それらのコマンドを発行した場合、デバッガとの間で競合をおこし、いずれかの動作が異常になる場合があります。

## コマンド書式

RTE - V 8 5 2 - I E 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, ABP3, ABP4 コマンド

## [ 書式 ]

```
abp [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp1 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp2 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp3 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
abp4 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
```

## [ パラメータ ]

ADDR: アドレス値を 16 進数で指定します。  
 DATA: アクセスデータを 16 進数で指定します。  
 MASK: データマスクを 16 進数で指定します。  
 read|write|access ステータスを指定します。  
   read: データリード  
   write: データライト  
   access: データアクセス  
 byte|hword|word|nosize アクセスデータサイズを指定します。  
   byte: バイトアクセス  
   hword: ハーフワードアクセス  
   word: ワードアクセス  
   nosize: アクセスサイズなし  
 /del: 設定を解除します。

## [ 機能 ]

アクセス系ブレイクポイントを設定または解除します。アクセス系ブレイクポイントは 4 点あり、abp コマンドで未使用のブレイクポイントに自動的に設定されます。

- ・明示的にポイントを指定する場合は、abp1,abp2,abp3,abp4 を使います。
  - ・データマスクは、各ビットでデータに対する無効ビットを指定します。
  - ・マスクビットが 1 の場合、対応するビットデータは比較されません。
- 例えば、マスクビットに ffffffff を指定した場合、アクセスデータは無視されます。

## [ 使用例 ]

```
abp 1020 0 ffffffff access hword
    1020H 番地へのハーフワードアクセスでブレイクします。(データは無視)
```

```
abp 1020 100 0 write word
    1020H 番地への 100H のワードライトでブレイクします。
```

```
abp2 /del
    abp2 を解除します。
```

A T P , A T P 1 , A T P 2 コマンド

## [ 書式 ]

```
atp [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp1 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp2 [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
```

## [ パラメータ ]

ADDR: アドレス値を 16 進数で指定します。  
 DATA: アクセスデータを 16 進数で指定します。  
 MASK: データマスクを 16 進数で指定します。  
 read|write|access ステータスを指定します。  
   read: データリード  
   write: データライト  
   access: データアクセス  
 byte|hword|word|nosize アクセスデータサイズを指定します。  
   hword: ハーフワードアクセス  
   word: ワードアクセス  
   nosize: アクセスサイズなし  
 /del: 設定を解除します。

## [ 機能 ]

アクセス系トリガポイントを設定または解除します。アクセス系トリガポイントは 2 点あり、atp コマンドで未使用のトリガポイントに自動的に設定されます。

- ・明示的にポイントを指定する場合は、atp1,atp2 を使います。
- ・データマスクは、各ビットでデータに対する無効ビットを指定します。
- ・マスクビットが 1 の場合、対応するビットデータは比較されません。

例) マスクビットに ffffffff を指定した場合、アクセスデータは無視されます。  
 このコマンドの発行により、トレースバッファはクリアされ、新しくトレースを開始します。

## [ 使用例 ]

```
atp 1020 0 ffffffff access hword
    1020H 番地へのハーフワードアクセスをトリガとします。(データは無視)
```

```
atp 1020 100 0 write word
    1020H 番地への 100H のワードライトでトリガとします。
```

```
atp2 /del
    atp2 を解除します。
```

A T P 3 コマンド

## [ 書式 ]

```
atp3 [in|out] [ADDR [HADDR [DATA [MASK]]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp3 [<|>|<<|>>|<=>|=eq] [ADDR [DATA [MASK]]] [read|write|access] [byte|hword|word|nosize] [/del]
atp3 noaddr [DATA [MASK]] [read|write|access] [byte|hword|word|nosize] [/del]
```

## [ パラメータ ]

in|out     アドレスの範囲を指定します。

in:         指定したアドレスの範囲内を有効とします。  
            ADDR <= トリカ'アドレス= HADDR

out:         指定したアドレスの範囲外を有効とします。  
            トリカ'アドレス ADDR, HADDR < トリカ'アドレス

ADDR:     下位アドレス値を 16 進数で指定します。

HADDR:    上位アドレス値を 16 進数で指定します。

<|>|<<|>>|<=>|=eq     アドレス条件を指定します。

<:         指定したアドレスより小さいアドレスを有効とします。  
            トリカ'アドレス ADDR

>:         指定したアドレスより大きいアドレスを有効とします。  
            トリカ'アドレス ADDR

<>:         アドレス値が不一致したアドレスを有効とします。  
            トリカ'アドレス> ADDR

<=:         指定したアドレスより以下を有効とします。  
            トリカ'アドレス= ADDR

>=:         指定したアドレスより以上を有効とします。  
            トリカ'アドレス= ADDR

eq:         アドレス値が一致したアドレスを有効とします。  
            トリカ'アドレス ADDR

noaddr     アドレス条件を無視します。

DATA:     アクセスデータを 16 進数で指定します。

MASK:     データマスクを 16 進数で指定します。

read|write|access     ステータスを指定します。

read:     データリード

write:    データライト

access:   データアクセス

byte|hword|word|nosize     アクセスデータサイズを指定します。

byte:     バイトアクセス

hword:    ハーフワードアクセス

word:     ワードアクセス

nosize:   アクセスサイズなし

/del:     設定を解除します。

## [ 機能 ]

範囲アドレスアクセス系トリガポイントを設定または解除します。範囲条件として、in,out を指定した場合は下位、上位アドレスの2点を指定します。その他の範囲指定はアドレス値は1点です。

このコマンドの発行により、トレースバッファはクリアされ、新しくトレースを開始します。

- ・データマスクは、各ビットでデータに対する無効ビットを指定します。
  - ・マスクビットが1の場合、対応するビットデータは比較されません。
- 例えば、マスクビットに ffffffff を指定した場合、アクセスデータは無視されます。

## [ 使用例 ]

```
atp3 in 1020 1300 0 ffffffff access hword
```

1020H 番地から 1300H 番地内へのハーフワードアクセスをトリガとします。

(データは無視)

```
atp3 <= 1020 100 0 write word
```

1020H 番地以下のアドレスへの 100H のワードライトでトリガとします。

envコマンド

## [ 書式 ]

```
env rom[32|64|128|256] ram[1|2|3|4|6|8|10|12|16|20|24|28] [romless|single]
    [[!]reset][[!]wait][[!]nmi][[!]hldrq] [w0|w16|w32|w64|w128|w256]
```

## [ パラメータ ]

```
rom[32|64|128|256]:
```

内蔵ROMのサイズを指定します。

指定できる値は 32K,64K,128K,256K です。

RTE-V852-IE ではROM32、またはROM128 でご使用下さい。

```
ram[1|2|3|4|6|8|10|12|16|20|24|28]
```

内蔵RAMのサイズを指定します。

指定できる値は 1K,2K,3K,4K,6K,8K,10K,12K,16K,20K,24K,28K です。

RTE-V852-IE では、RAM1、またはRAM3 でご使用下さい。

```
[[romless|single]
```

CPUのモードを指定します。ROMレスモードまたはシングルチップモード。

```
[[!reset]
```

RESET端子のマスクを指定します。!はマスクしないを意味します。

```
[[!]wait]
```

WAIT端子のマスクを指定します。!はマスクしないを意味します。

```
[[!]nmi]
```

NMI端子のマスクを指定します。!はマスクしないを意味します。

```
[[!]hldrq]
```

HLD R Q端子のマスクを指定します。!はマスクしないを意味します。

```
[[!]stop]
```

RTE-V852-IE では、常に初期値（マスクする）でご使用下さい。

```
[w0|w16|w32|w64|w128|w256]
```

ウェイトのタイムオーバー時間を指定します。

・w0はウェイトし続けるを意味します。

・その他は16から256クロックでタイムオーバーすることを意味します。

## [ 機能 ]

RTE-V852-IEのエミュレーションCPUの各種環境値を設定します。それぞれのパラメータは設定するものだけを指定します。

・パラメータの順序は任意です。

・ただし、択一のパラメータを同時に指定した場合は後から指定したものが有効になります。

・initコマンド発行後の初期値は以下の通りです。

V852の場合の初期値を以下に示します。( )内はV851の場合です。

内蔵ROM容量 : 128K ( 32K )

内蔵RAM容量 : 3K ( 1K )

CPUモード : シングルチップモード

端子マスク : 全て、マスクしない ( stopのみはマスクする )

ウェイト時間 : 256クロック

## [ 使用例 ]

```
env single !nmi
```

シングルチップモードに設定し、NMI端子をマスクしません。

```
env romless rom128 w16
```

ROMレスモード、内蔵ROMを128Kに、ウェイトを16クロックに指定します。

## h e l pコマンド

### [ 書式 ]

help [ command ]

### [ パラメータ ]

command: コマンド名を指定します。

コマンド名を省略した場合、コマンドの一覧が表示されます。

### [ 機能 ]

各コマンドのヘルプメッセージを表示します。

### [ 使用例 ]

help map

map コマンドのヘルプを表示します。



## initコマンド

[書式]

init

[パラメータ]

なし

[機能]

RTE - V 8 5 2 - I Eを初期化します。マッピング情報、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。

## mapコマンド

### [書式]

map [ADDR LENGTH] [guard|ram|rom|target]

### [パラメータ]

ADDR: マッピングを開始するアドレス。

LENGTH: マッピングを行うバイト数。

[guard|ram|rom|target]: マッピング属性を指定します。

guard : ガードエリアを指定します。アクセスを行うとエラーになります。

ram : エミュレーションメモリをramとして扱います。

rom : エミュレーションメモリをromとして扱います。

target : ターゲットシステムのメモリを使用します。

### [機能]

メモリマッピングを行います。エミュレーションメモリは1 M以下の空間に1 Mバイト、1 M以上の空間の一つの1 Mバイトの空間に割り付け可能です。

- ・ 割り付けのサイズは各64 Kバイト単位です。

### [使用例]

map 100000 80000 ram

100000h から 512k バイトをエミュレーション ram に割り付けます。

map 0 100000 guard

0h 番地から 1m バイトをガードエリアとして割り付けます。

## 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 ]

RTE-V852-IE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は実際にはメモリをリードしません。メモリにI/Oを割り付けている場合は、このキャッシュ機能は実際の動作と矛盾してしまいますので、このコマンドでメモリキャッシュの除外エリアを指定して下さい。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。ただし、アドレス0fff000より上位の空間は常にメモリキャッシュの除外エリアになっています。したがって、この部分はノーメモリキャッシュ領域として指定する必要はありません。

### [ 使用例 ]

```
nc 10000 1000
```

10000 番地から 1000 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 10000 1000
```

```
No Memory Cache Area
```

```
No. Address Length
```

```
1 010000 001000
```

```
2 fff000 001000
```

## 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 2

ブロック番号 2 をメモリキャッシュの除外エリアから削除します。

>nc

No Memory Cache Area

No. Address Length

1 020000 000100

2 010000 001000

3 fff000 001000

>ncd 2

No Memory Cache Area

No. Address Length

1 020000 000100

2 fff000 001000

## resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

RTE - V 8 5 2 - I EのエミュレーションCPUをリセットします。

## r r m bコマンド

### [ 書式 ]

rmb ADDR

### [ パラメータ ]

ADDR: リアルタイムRAMモニタのベースアドレスの指定。

### [ 機能 ]

リアルタイムRAMモニタのベースアドレスを指定します。ここで指定したアドレスから1Kバイトの領域はCPU実行中でもrmbコマンドを使用してメモリ参照が可能です。

### [ 使用例 ]

rmb 10000

10000番地から1KバイトをリアルタイムRAMモニタ領域に指定します。

## rrmコマンド

### [ 書式 ]

rrm [ADDR[LENGT]]

### [ パラメータ ]

ADDR: リアルタイムRAMモニタ内のメモリ参照を行う開始アドレスを指定します。

LENGTH: リードするバイト数を指定します。(最大256バイト)

### [ 機能 ]

リアルタイムRAMモニタ領域内のメモリを参照します。レンジは最大256バイトです。

### [ 使用例 ]

rrm 10000 20

10000 番地から 30H バイト、リアルタイムRAMモニタ領域からリードします。

s f r コマンド

## [書式]

sfr [reg] [VAL]

## [パラメータ]

VAL: S F R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

P0 P1 P2 P3 P4 P5 P6 P9 P10 PM0 PM1 PM2 PM3 PM4 PM5 PM6 PM9 PM10  
 PMC0 PMC2 PMC3 MM PMC10 DWC BCC PSC SYS BRG0 BPRM0 CSIM0 SIO0 BRG1  
 BPRM1 CSIM1 SIO1 CSIM2 SIO2 ASIM00 ASIM01  
 OVIC1 P1IC0 P1IC1 P1IC2 P1IC3 CMIC4 CSIC0 SEIC0 SRIC0  
 STIC0 P0IC0 P0IC1 P0IC2 P0IC3 CSIC1 CSIC2  
 INTM0 INTM1 INTM2 TOVS TUM1 TMC1 TOC1 CC10 CC11 CC12 CC13 TMC4 CM4

## ライトオンリーレジスタ

TXS0 TXS0L PRCMD

## リードオンリーレジスタ

ASIS0 RXB0 RXB0L ISPR TM1 TM4

## [機能]

S F Rレジスタの値を設定・表示します。

## [使用例]

sfr TM4  
 TM4 レジスタの値を表示します。  
 sfr MM 0a0  
 MM レジスタに 0a0H を設定します。



## symfile, symコマンド

### [ 書式 ]

symfile FILENAME ::GHS 社の ELF ファイルからグローバルシンボルを読み込みます  
sym [NAME] :シンボルの表示 ( 3 0 個 ) を行います

### [ パラメータ ]

symfile: ファイル名  
sym: シンボルの先頭文字列

### [ 機能 ]

symfile コマンドは、FILENAME で指定した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。(最大: 8 1 5 0 シンボル)  
sym コマンドは、読み込んだシンボルの表示 (最大 3 0 個) を行います。

### [ 使用例 ]

symfile c:\test\dry\dry.elf  
          c:\test\dry のディレクトリの dry.elf から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。

## time コマンド

### [ 書式 ]

time [sysclk]

### [ パラメータ ]

sysclk: cpu のシステムクロックを MHz の単位で指定します。小数点以下 2 桁まで有効です。指定しなかった場合のデフォルト値は、33.33MHz です。

### [ 機能 ]

実行時間計測結果を時間で表示します。実行時間計測のタイマーは CPU が実行を開始する毎に初期化され、CPU 実行中カウントされます。タイマーは 2 クロックの分解能でカウントした値します。

### [ 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 33.33

33.33MHz のシステムクロックで実行した時の時間を表示します。

TP, TP1, TP2 コマンド

## [ 書式 ]

tp [ADDR] [/del]

tp1 [ADDR] [/del]

tp2 [ADDR] [/del]

## [ パラメータ ]

ADDR: アドレス値を 16 進数で指定します。

/del: 設定を解除します。

## [ 機能 ]

実行系トリガポイントを設定または解除します。

実行系トリガポイントは 2 点あり、tp コマンドで未使用のトリガポイントに自動的に設定されます。

・明示的にポイントを指定する場合、tp1, tp2 を使用します。

このコマンドの発行により、トレースバッファはクリアされ、新しくトレースを開始します。

## [ 使用例 ]

tp 1020

1020H 番地の命令実行をトリガとします。

TRONコマンド

## [ 書式 ]

```
tron [DELAY] [add|cycle]
```

```
tron [x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k]
```

## [ パラメータ ]

DELAY=0..07fff デレイカウンタ

トリガ成立後に取り込む命令サイクル数（デレイカウンタ）を16進数で指定します。トレースバッファは最大32Kサイクル取り込むことができます。

x1..x1m トレースタイムタグカウンタの分周率

タイムタグカウンタクロックの分周率を指定します。

x16 ならタイマの単位を16倍することを意味します。

add 加算モードを指定します。

前サイクルからの累積値を表示します。

cycle サイクルモードを指定します。

トレースタイムタグカウンタ値は、サイクル毎の値を表示します。

## [ 機能 ]

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。

## [ 注意 ]

1. タイマカウンタを加算モードに設定した場合も、途中でブレイクした場合のタイムタグは、再実行時に一度クリアされます。

## [ 使用例 ]

```
tron 100 x16 cycle
```

トリガ成立後 100h (256) サイクル分トレースの取り込みによりトレースを終了します。タイマの単位は16倍でサイクル毎の実行サイクルをトレースバッファに書き込みます。

## 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] [asm]

## [ パラメータ ]

POS= $\pm 0..07fff$  読み出しを開始する位置 (トリガポイントまたは終了点が0)  
 トレースバッファの先頭からのサイクル数を16進数で指定します。

asm 表示種別 (アセンブラ) ...逆アセンブラ表示で表示します。

## [ 機能 ]

トレースバッファの内容を表示します。

- ・このコマンドを発行するとトレースの取り込みは終了します。
- ・再度トレースを開始するには、tron コマンドを発行します。

## [ 表示内容 ] : アセンブラモード

```

Frame neis Time Ext  Address  Code   Operand
_start:
-0003 --i- 0003 0000 00000800 401e0000 movehi 0000h,zero,sp
-0002 --i- 0003 0000 00000804 231efcef movea  -1004h,sp,sp
-0001 --i- 0001 0000 00000808 40360000 movehi 0000h,zero,r6
+0000 --i- 0001 0000 0000080c 26365c11 movea  +115ch,r6,r6
+0001 --i- 0001 0000 00000810 6600      jmp    [r6]
main:
+0002 --i- 0003 0000 0000115c 5c1a      add    -04h,sp
+0003 --i- 0001 0000 0000115e 63ff0100 st.w   lp,+00h[sp]
                                00000246 Write 00000246h->[00ffeff8h]
main+0006h:
+0004 --i- 0001 0000 00001162 bfff64f8 jarl   RegChkInit(000009c6h)
RegChkInit:
+0005 --i- 0003 0000 000009c6 501a      add    +10h,sp
+0006 --i- 0001 0000 000009c8 63ff0d00 st.w   lp,+0ch[sp]
                                00001166 Write 00001166h->[00ffeff4h]
+0007 --i- 0001 0000 000009cc 63b70900 st.w   r22,+08h[sp]
                                00000000 Write 00000000h->[00ffeff0h]
+0008 --i- 0001 0000 000009d0 63af0500 st.w   r21,+04h[sp]
                                00000908 Write 00000908h->[00ffefech]

```

Frame: 最初のトリガサイクルを0とした相対位置を16進数で表示します。

neis: P S Wのフラグを表示します。

n: NMIフラグ  
 e: 外部割り込みフラグ  
 i: 例外フラグ  
 s: 飽和フラグ

Time: トレースタイムアウトを16進数で表示します。

Ext: 外部データをビット単位で表示します。(右からEXT0, 1, 2, 3の順です)

Address: 命令実行のアドレスを16進数で表示します。

Code: 実行の場合は命令コード、データの場合はデータを16進数で表示します。  
Operand: 命令を逆アセンブル表示します。  
データの場合は  
Read [アドレス]<- データ  
Write データ-> [アドレス]  
と表示します。データの桁数はデータサイズを意味します。

## 《注意》

time 表示は、最初の2フレームは誤差を含みます。また、tron コマンドで add (加算モード) を指定した場合は、最初のフレームは加算対象になりません。

## verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

RTE - V 8 5 2 - I Eのバージョンを表示します。



< 改訂履歴 >

1997/10/16      symfile コマンド：読み込みファイルを sym ファイルから elf ファイルに  
                  変更。