

## 付録 . A RTE-V850E/MA1-IE 内部コマンド

本書は、RTE-V850E/MA1-IE の内部コマンドについて記述しています。これらのコマンドは、各ディバグモニタの中でスルーコマンドとして使用できます。スルーコマンドの入力方法はそれぞれのディバグモニタのマニュアルを参照ください。

(例) 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

付録 . A RTE-V850E/MA1-IE 内部コマンド.....	1
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の設定 : ABP , ABP 1 , ABP 2 , ABP 3 , ABP 4 コマンド.....	3
アクセス系トレーストリガの設定 : ATP , ATP 1 , ATP 2 コマンド.....	4
アクセス系トレーストリガの設定 (範囲) : ATP 3 コマンド.....	5
環境設定 : ENV コマンド.....	7
ヘルプ : HELP コマンド.....	9
初期化 : INIT コマンド.....	10
メモリマップの指定 : MAP コマンド.....	11
デバグキャッシュ領域の解除 : NC コマンド.....	12
デバグキャッシュ領域に設定 : NCD コマンド.....	13
CPU のリセット : RESET コマンド.....	14
リアルタイム RAM モニタのベース : RRMB コマンド.....	15
リアルタイム RAM モニタのリード : RRM コマンド.....	16
SFR : SFR コマンド.....	17
シンボル : SYMFILE , SYM コマンド.....	18
実行時間の表示 : TIME コマンド.....	19
実行系トレーストリガの設定 : TP , TP 1 , TP 2 コマンド.....	20
トレースの開始と設定 : TRON コマンド.....	21
トレースの強制終了 : TROFF コマンド.....	22
トレース結果の表示 : TRACE コマンド.....	23
バージョン表示 : VER コマンド.....	25

ご注意：これらのコマンドは、ご使用になりたい機能がディバグ本体にない場合にのみ補助的にご使用ください。ご使用になるディバグ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、ディバグとの間で競合をおこし、いずれかの動作が異常になる場合があります。

## コマンド書式

RTE-V850E/MA1-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, 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 (この指定では、DATA は無視されます)

/del: 設定を解除します。

## [ 機能 ]

アクセス系ブレークポイントを設定または解除します。アクセス系ブレークポイントは 4 点あり、abp コマンドで未使用のブレークポイントのチャンネルに自動的に設定されます。

- ・明示的にチャンネルを指定する場合は、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ークします。(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ークします。

abp2 /del
    abp2 を解除します。

```

atp, atp1, atp2 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のチャンネルに自動的に設定されます。

- ・明示的にチャンネルを指定する場合は、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。
```

atp3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 <= トリガアドレス <= HADDR

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

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。  
              トリガアドレス < 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。  
              トリガアドレス > ADDR

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

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

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

eq:          アドレス値が一致したアドレスを有効とします。  
              トリガアドレス = 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 [romXX] [ramXX] [romless0|romless1|single0|single1]
env [[!]reset] [[!]nmi] [[!]hldrq] [[!]stop] [waitmode{0|1|2|3}] [w0|w16|w32|w64|w128|w256]
env [[!]dmatrace]
```

## [ 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,512K です。  
 RTE-V850E/MA1-IE では、通常 ROM256 でご使用ください。

ram[4|12|28|60]

内蔵 RAM のサイズを指定します。  
 指定できる値は 4K,12K,28K,60K です。  
 RTE-V850E/MA1-IE では、通常 RAM12 でご使用ください。

romless0|romless1|single0|single1:

CPU のモードを指定します。

[[!]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。

[[!]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]:

HLDRQ 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-V850E/MA1-IE では、常に初期値 (マスクする) でご使用ください。

waitmode{0|1|2|3}:

wait のモード指定です。

以下の組み合わせがあります。通常、waitmode1 でご使用ください。

wait mode	External	Emulation
	Memory	Memory
-----+		
waitmode0	EXT-RDY	EXT-RDY
waitmode1	EXT-RDY	2WAIT
waitmode2	0WAIT	2WAIT
waitmode3	0WAIT	0WAIT

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 クロックでバスを強制終了 (タイマオーバ) することを意味します。

[[!]dmatrace:

DMA サイクルをトレースに入れる場合に指定します。! は入れない指定です。

## [ 機能 ]

RTE-V850E/MA1-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。
- ・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。

起動直後の初期値は以下の通りです。

内蔵 ROM 容量	: 256KB
内蔵 RAM 容量	: 12KB
端子マスク	: stop を除き、全て、マスクしない
ウェイトモード	: ウェイトモード 1
ウェイト時間	: 256 クロック
DMA トレース	: 禁止

## [ 使用例 ]

```
env !nmi
```

NMI 端子をマスクしません。

```
env rom512 RAM28
```

内蔵 ROM を 512K バイトに、内蔵 RAM を 28K バイトに指定します。



## 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。

## initコマンド

[書式]

init

[パラメータ]

なし

[機能]

RTE-V850E/MA1-IE を初期化します。メモリキャッシュの除外エリア以外は、全て初期化されます。

## 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。リード/ライト可能です。

rom : エミュレーションメモリを rom として扱います。リードのみ可能です。

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。割り付けのサイズは 1M バイト単位です。

### [注意]

rom としてマップした領域へライトした場合、強制的にブレークがかかりますが、書き込みされたアドレスのデータは破壊されます。

### [使用例]

map 100000 100000 ram

100000h から 1M 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 32 バイト、最小値 32 バイト

## [ 機能 ]

I/O 空間や共有メモリ等、メモリのキャッシングに不適な領域を指定してください。

RTE-V850E/MA1-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ディバッガが内部的に持っています。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ないで内部的に処理しますので高速に処理できる反面、I/O に割り付けている空間においては、このキャッシュ機能は実際の動作と矛盾する場合があります。そこで、IO 空間をこのコマンドでメモリキャッシュの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。以下の空間は、通常、CPU の SFR 空間に指定されていますので、起動時に初期値として入っています。したがって、この空間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。

```
No Memory Cache Area
No. Address Length
1 0ffff000 f0001000
2 03fff000 00001000
```

## [ 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 00010000 00001000
2 0ffff000 f0001000
3 03fff000 0c001000
```

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 10000 1000
No Memory Cache Area
No. Address Length
1 00010000 00001000
2 0ffff000 f0001000
3 03fff000 0c001000
```

```
>ncd 1
No Memory Cache Area
No. Address Length
1 0ffff000 f0001000
2 03fff000 0c001000
```

## resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

RTE-V850E/MA1-IE のエミュレーション CPU をリセットします。

## r rmb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。



sfr コマンド

## [書式]

sfr [reg] [VAL]

## [パラメータ]

VAL: SFR レジスタ値を 16 進数で指定します。

reg: SFR 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:

PAL PAH PDL PCS PCT PCM PCD PBD  
 PMAL PMAH PMDL PMCS PMCT PMCM PMCD PMBD  
 PMCAL PMCAH PMCDL PMCCS PFCCS PMCCT PMCCM PFCCM PMCCD PFCCD PMCBS PMCBD  
 CSC0 CSC1 BSC VSWC DSALO DSAHO  
 DDALO DDAH0 DSAL1 DSAH1 DDAL1 DDAH1 DSAL2 DSAH2 DDAL2 DDAH2 DSAL3 DSAH3  
 DDAL3 DDAH3 DBC0 DBC1 DBC2 DBC3  
 DADC0 DADC1 DADC2 DADC3 DCHC0 DCHC1 DCHC2 DCHC3  
 DRST IMRO IMROL IMROH IMR1 IMR1L IMR1H IMR2 IMR2L IMR2H IMR3 IMR3L  
 IMR3H OVIC00 OVIC01 OVIC02 OVIC03 P00IC0 P00IC1 P01IC0 P01IC1 P02IC0  
 P02IC1 P03IC0 P03IC1 P10IC0 P10IC1 P10IC2 P10IC3 P11IC0 P11IC1 P11IC2  
 P11IC3 P12IC0 P12IC1 P12IC2 P12IC3 P13IC0 P13IC1 P13IC2 P13IC3 CMIC40  
 CMIC41 CMIC42 CMIC43 DMAIC0 DMAIC1 DMAIC2 DMAIC3 CSIIC0 SEIC0 SRIC0  
 STIC0 CSIIC1 SEIC1 SRIC1 STIC1 CSIIC2 SEIC2 SRIC2 STIC2 ADIC  
 PSC ADM0 ADM1 ADM2 P0 P1 P2 P3  
 P4 P5 P7 PM0 PM1 PM2 PM3 PM4  
 PM5 PMCO PMC1 PMC2 PMC3 PMC4  
 PMC5 PFC0 PFC2 PFC3 PFC4 BCT0  
 BCT1 DWCO DWC1 BCC ASC BCP PRC RWC  
 DRC1 SCR1 RFC1 RFS1 DRC3 SCR3 RFC3 RFS3 DRC4 SCR4 RFC4 RFS4 DRC6 SCR6  
 RFC6 RFS6 CMD0 TMD0 CMD1 TMD1  
 CMD2 TMD2 CMD3 TMD3 CCC00 CCC01  
 TMCC00 TMCC01 SESCO CCC10 CCC11 TMCC10 TMCC11 SESC1 CCC20 CCC21  
 TMCC20 TMCC21 SESC2 CCC30 CCC31 TMCC30 TMCC31 SESC3  
 PHS DTFRO DTFR1 DTFR2 DTFR3 PSMR  
 CKC INTMO INTM1 INTM2 INTM3 INTM4  
 CSIMO CSIC0 SOTB0 CSIM1 CSIC1 SOTB1  
 CSIM2 CSIC2 SOTB2 ASIMO TXB0  
 CKSR0 BRGCO ASIM1 TXB1 CKSR1 BRGC1 ASIM2 TXB2  
 CKSR2 BRGC2 PWMCO PWMB0 PWMC1  
 PWMB1

ライトオンリーレジスタ:

PRCMD PHCMD

リードオンリーレジスタ:

DDIS ISPR ADCRO ADCR1 ADCR2 ADCR3  
 ADCR4 ADCR5 ADCR6 ADCR7 ADCROH ADCR1H ADCR2H ADCR3H ADCR4H ADCR5H ADCR6H  
 ADCR7H TMD0 TMD1 TMD2 TMD3 TMC0  
 TMC1 TMC2 TMC3 UNLOCK SIO0 SIO1  
 SIO2 RXB0 ASIS0 ASIF0 ASISE0  
 RXB1 ASIS1 ASIF1 ASISE1 RXB2 ASIS2 ASIF2 ASISE2

## [機能]

SFR レジスタの値を設定・表示します。

## [使用例]c

sfr P3

P3 レジスタの値を表示します。

sfr PMC3 07

PMC3 レジスタに 07H を設定します。

## symfile, symコマンド

### [ 書式 ]

symfile FILENAME :GHS の elf ファイル(.elf)から読み込みを行います  
sym [NAME] :シンボルの表示 ( 30 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 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。また、sym コマンドで読み込んだシンボルの表示 ( 最大 30 個 ) ができます。

### [ 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 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、50MHzです。

### [機能]

実行時間計測結果を時間で表示します。実行時間計測のタイマーはCPUが実行を開始する毎に初期化され、CPU実行中カウントされます。タイマーの値はCPUクロックで1回カウントしています。

### [備考]

測定値は実行の開始とブレークのオーバーヘッド時間(数クロック)を含みます。

### [使用例]

time 40

40MHz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。

## [ 注意 ]

実行開始アドレスには設定できません。設定した場合、実行開始直後の実行は無視されます。

## [ 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][all|qualify|qualify2]
[x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k|x512k
|x1m|x2m|x4m|x8m|x16m|x32m|x64m|x128m|x256m|x512m|x1g|x2g]
```

## [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(7fff)取り込むことができます。

```
[add|cycle] [x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k|x512k
|x1m|x2m|x4m|x8m|x16m|x32m|x64m|x128m|x256m|x512m|x1g|x2g]
```

これらのパラメータは、実行時間計測用タイマのモード設定の為のもので、

add 加算モードを指定します。

トレースタグ かつ値は、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。

トレースタグ かつ値は、サイクル毎の値を表示します。

x1..x2g トレースタグ かつの基本クロックの分周率を指定します。

通常は、1:1となるx1を指定してください。測定値がオーバフローする場合、順次右のパラメータを指定することで、1/2づつ分周比が増えていきます。

例) x16の場合、1/16のクロックで計数することを意味します。

[all|qualify|qualify2] トレースの取り込むモードを設定します。

All 全てのサイクルを取り込みます。

qualify トリガとして指定したサイクルのみをトレースし、デレイカウンタ数分のサイクルを取り込み、トレースを終了します。

qualify2 トリガとして指定したサイクルのみをトレースします。トレースを終了するまで(TROFF または、TRACE コマンドが発行されるまで)トレースは続きます。

## [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時間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= ±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) ...逆アセンブル表示します。

## [ 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:        N M I 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:    トレースバッファのカウンタを 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 E X T 0 , 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、ステップ実行中とリアルタイム実行直後の2フレームとブレーク直前のフレームは不正です。



## verコマンド

[ 書式 ]

ver

[ パラメータ ]

なし

[ 機能 ]

RTE-V850E/MA1-IE のバージョンを表示します。