

付録. A KIT-V850E/MA3-TP (H) 内部コマンド

本書は、KIT-V850E/MA3-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。

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

RTESERV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.....	A-1
コマンド書式.....	A-2
アクセス系ブレークポイント : ABP, ABP1, ABP2コマンド.....	A-3
環境設定 : ENV, EMEMSTATコマンド.....	A-4
ヘルプ : HELPコマンド.....	A-6
内蔵フラッシュROM書込み : IFROM... コマンド.....	A-7
INPUT : INB, INH, INWコマンド.....	A-9
初期化 : INITコマンド.....	A-10
JTAGリード : JREADコマンド.....	A-11
デバッガキャッシュ領域の解除 : NCコマンド.....	A-12
デバッガキャッシュ領域の設定 : NCDコマンド.....	A-13
ソフトブレーク禁止領域の設定 : NSBPコマンド.....	A-14
ソフトブレーク禁止領域の解除 : NSBPDコマンド.....	A-15
強制ユーザ領域の設定 : NRROMコマンド.....	A-16
強制ユーザ領域の解除 : NRROMDコマンド.....	A-17
OUTPUT : OUTB, OUTH, OUTWコマンド.....	A-18
CPUリセット : RESETコマンド.....	A-19
E. ROMの設定 : ROMコマンド (RTE-1000-TP 用コマンド).....	A-20
E. ROMの設定 : ROM1.. ROM4コマンド (RTE-2000 (H)-TP 用コマンド).....	A-21
SFRアクセス : SFRコマンド.....	A-23
シンボル : SYMFILE, SYMコマンド.....	A-25
実行時間計測 : TIMEコマンド.....	A-26
バージョン表示 : VERコマンド.....	A-27

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。IFROMxxx コマンドは、Rte4win32 V5.10.xx 以上で対応しています。

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, ememstatコマンド

[書式]

```
env [!]auto [!][verify] [jtag[xxx][. [yyy]] [M|K]]
    [!]nmi [!]intwdt [!]resetz [!]hldrqz [!]waitz
```

[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。

[!]nmi : NMI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。

[!]intwdt : INTWDT(ウォッチドッグからのNMI割込み)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-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。

[!]hldrqz : HLDRQ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。

[!]waitz : WAIT-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[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。

[機能]

envコマンドは、エミュレーション環境の設定と状態を表示します。

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RTE-2000(H)-TPの場合に、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。

RTE-1000-TPの場合

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プローブの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 = 25MHz (jtag25M)

Verify = verify off (!verify)

CPU Mode = single (single)

Signals Mask:

NMI = NO MASK (!nmi)

```

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

```

RTE-2000 (H)-TPの場合

```

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)
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       8Mbyte    Extend Type 2K

```

[入力例]

```

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

```

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。

ifromxxxコマンド

[書式]

```

ifromfreq [FREQ_KHZ]
ifromenv [[!]write] [[!]verify] [[!]tracecache]
ifromclear
ifromflush
ifromwrite S_RECODE_FILENAME
ifromverify S_RECODE_FILENAME

```

[パラメータ]

[[!]write]: 内蔵フラッシュROMに対する自動書き込みモードを指定します。writeでON、!writeでOFFです。初期値はwriteです。

[[!]verify]: 自動書き込みモードがONになっている時に書き込み後のVerifyを指定します。verifyで”する”, !verifyで”しない”です。初期値は!verifyです。

[[!]tracecache]: 実行中にトレース表示を行うためにIROM領域への書き込みデータをキャッシングする時に指定します。!で指定しないを意味します。初期値はtracecacheです。

FREQ_KHZ: CPUへ入力しているクロックの周波数をKHz単位で入力します。

S_RECODE_FILENAME: Sレコードのオブジェクトファイルを入力します。

[機能と使用例]

```

ifromfreq [FREQ_KHZ]

```

CPUへ入力しているクロックの周波数をKHz単位で入力します。
書き込みコマンドに先立ち、必ず指定してください。

8MHzを入力している場合の入力例

```

>ifromfreq 8000
CPU Input Frequency Setting:
CPU Input Frequency : 8.000MHz (8000)

```

```

ifromenv write !verify

```

自動書き込みモードの設定をするコマンドです。
入力例は自動書き込みをONにし、Verifyはoffにします。

- ・書き込みはプログラム実行開始時にまとめて処理されます。
- ・書き込みには数十秒かかる場合があります。

```

ifromclear

```

自動書き込みモード中、キャッシュの状態を強制的にINVALIDにするコマンドです。

```

ifromflush

```

自動書き込みモード中、DIRTY領域のキャッシュのデータを、強制的にフラッシュROMに書き込むコマンドです。

```

ifromwrite S_RECODE_FILENAME

```

Sレコードのオブジェクトファイルを指定して、内蔵フラッシュに書き込みます。

- ・Sレコード以外のフォーマットには対応していません。
- ・書き込み前に内蔵のフラッシュは全領域イレーズされます。

- ・内蔵フラッシュの領域以外のアドレスのオブジェクトが含まれている場合、無視します。
- ・書き込みには数十秒かかる場合があります。
- ・表示は全ての処理が完了した時点でまとめて行われます。

w:¥ma3¥ram_test.sを内蔵フラッシュに書込む場合の入力例

```
>ifromwrite w:¥ma3¥ram_test.s
CPU Input Frequency : 8.000MHz (8000)
The following file is written to Internal FlashROM.
"w¥ma3¥ram_test.s"
Write Data Range : 00000100 - 00003fbe
Erase all data in Internal FlashROM (00000000 - 0007ffff).
Write Complete.
Internal Verify Complete.
Verify Complete.
```

ifromverify S_RECODE_FILENAME

Sレコードのオブジェクトファイルを指定して、内蔵フラッシュの内容と比較します。

- ・Sレコード以外のフォーマットには対応していません。
- ・内蔵フラッシュの領域以外のアドレスのオブジェクトが含まれている場合、無視します。
- ・処理には数十秒かかる場合があります。
- ・表示は全ての処理が完了した時点でまとめて行われます。

w:¥ma3¥ram_test.sを内蔵フラッシュの内容と比較する場合の例

```
>ifromverify w:¥ma3¥ram_test.s
The following file is compared with Internal FlashROM.
"w:¥ma3¥ram_test.s"
Verify Data Range : 00000100 - 00003fbe
Verify Complete.
```

注意：ID850NWを使用する場合の注意事項

ID850NWでは、バックスラッシュ(“¥”記号)がコマンド入力にそのまま使用できません。
ファイルのパスの入力は以下の方法で行ってください。

方法1. パスの中括弧でくる

-> (NECTools32) 2 % ie ifromwrite {w:¥ma3¥ram_test.s}

方法2. バックスラッシュを重ねる

-> (NECTools32) 2 % ie ifromwrite w:¥¥ma3¥¥ram_test.s

[備考]

詳細は、ユーザズ・マニュアル「7章. 内蔵フラッシュROM・サポート機能」を参照ください。

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

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。)

内蔵フラッシュROMの領域をこのコマンドで読み出した場合、自動書き込みの為のキャッシュの値ではなく、実フラッシュROMの内容が読み出せます。

[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]

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 * 3 2 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 3 2 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: ソフトウェアブレーク禁止領域の開始アドレスを指定します。

LENGTH: ソフトウェアブレー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]

ソフトウェアブレークを禁止したい領域を指定します。

非指定領域でブレー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ーク禁止領域に指定します。

```
>nsbp 100000 20000
```

```
Num Address Length
```

```
01 00100000 00020000
```

n s b p d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。

RTE-1000-TP : 4-byte単位。

RTE-2000(H)-TP : 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EMEMEN-信号はインアクティブ(Highレベル)になります。(RTE-2000(H)-TPのみ)

[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。

outw 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]

エミュレーションCPUをリセットします。

romコマンド(RTE-1000-TP用コマンド)

[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]

RTE-1000-TPのROMの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変更が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領域における注意事項

rom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。

rom1..rom4コマンド(RTE-2000(H)-TP用コマンド)

[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に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）になっています。

[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 – 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	

s f r コマンド

[書式]

sfr [reg [VAL]]

[パラメータ]

VAL: SFRのレジスタ値を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。

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 PFCGS
 PMCCT PFCCT PMCCM PMCCD PMCBD CSC0 CSC1 BEC 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 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
 OVPI0 OVQIC OVPI1 OVPI2 DMAIC0 DMAIC1 DMAIC2 DMAIC3 SEIC0 SRIC0
 STIC0 SEIC1 SRIC1 STIC1 SEIC2 SRIC2 STIC2 SEIC3 SRIC3 STIC3 ADIC PSC
 ADM0 ADM1 ADM2 ADTS DA0CS0 DA0CS1
 DA0M 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
 DWC1 BCC ASC BCP LBS LBC0 LBC1 FWC FIC BMC PRC AHC SCR1
 RFS1 SCR3 RFS3 SCR4 RFS4 SCR6 RFS6 CMD0
 TMCDO CMD1 TMCD1 CMD2 TMCD2 CMD3
 TMCD3 TMENC10 CM100 CM101 CC100
 CC101 CCR10 TUM10 TMC10 SESA10 PRM10 TQOCTL0
 TQOCTL1 TQ01OC0 TQ01OC1 TQ01OC2 TQ00PT0 TQ0CCR0 TQ0CCR1 TQ0CCR2 TQ0CCR3
 TQ0OPT1 TQ0OPT2 TQ01OC3 TQ0DTC HZA0CTL0
 HZA0CTL1 TPOCTL0 TPOCTL1 TP01OC0 TP01OC1 TP01OC2 TP00PT0 TPOCCR0 TPOCCR1
 TP1CTL0 TP1CTL1 TP11OC0 TP11OC1 TP11OC2 TP10PT0 TP1CCR0 TP1CCR1
 TP2CTL0 TP2CTL1 TP21OC0 TP21OC1 TP21OC2 TP20PT0 TP2CCR0 TP2CCR1
 OST5 WDC5 WDTM PFCE0 PFCE1 PFCE2
 PFCE3 PFCE5 SYS DTFRO DTFR1 DTFR2
 DTFR3 PSMR CKC PCC WDRES CORAD0
 CORAD0L CORAD0H CORAD1 CORAD1L CORAD1H CORAD2 CORAD2L CORAD2H CORAD3
 CORAD3L CORAD3H CORCN DTOC DIFC
 DAKW FLPMC UA0CTL0 UA0CTL1 UA0CTL2
 UA00PT0 UA0STR UA0TX UA1CTL0 UA1CTL1 UA1CTL2 UA10PT0 UA1STR UA1TX UA2CTL0
 UA2CTL1 UA2CTL2 UA20PT0 UA2STR UA2TX UA3CTL0 UA3CTL1 UA3CTL2 UA30PT0

UA3STR UA3TX INTF0 INTF1 INTF2
 INTF3 INTF5 NMIF INTRO INTR1 INTR2
 INTR3 INTR5 NMIR CBOCTL0 CBOCTL1
 CBOCTL2 CB0STR CB0TX CB0TXL CB1CTL0 CB1CTL1 CB1CTL2 CB1STR CB1TX CB1TXL
 CB2CTL0 CB2CTL1 CB2CTL2 CB2STR CB2TX CB2TXL IIC
 IICC SVA IICCL IICX IICF PRSM PRSCM
 SFR (W):
 PRCMD PFCMD
 SFR (R):
 ISPR ADCR0 ADCR1 ADCR2 ADCR3
 ADCR4 ADCR5 ADCR6 ADCR7 ADCR0H ADCR1H ADCR2H ADCR3H ADCR4H ADCR5H ADCR6H
 ADCR7H P7 P8 TMD0 TMD1 TMD2 TMD3
 STATUS10 TQOCNT TQODTT1 TQODTT2
 TQODTT3 TPOCNT TP1CNT TP2CNT
 PFS UA0RX UA1RX UA2RX UA3RX CB0RX
 CB0RXL CB1RX CB1RXL CB2RX CB2RXL
 IICS

[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 55

PALレジスタに55h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。
```

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、JTAGCLK を 2 分周して使用され、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 = 10,320 (ns) (12.500000MHz) [Counter=00000081] << JTAGCLK=25MHz時の表示です。

	_Counter 値 (Hex)
	_ 計測クロックの周波数 (JTAGCLK の 1/2)

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。