

付録. A KIT-VR41xx-TP (H) 内部コマンド

本書は、KIT-VR41xx-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
オプションブレーク : BPOPTコマンド.....	A-3
キャッシュ操作 : CACHEINIT, CASHEFLUSHコマンド.....	A-4
環境設定 : ENV, EMEMSTATコマンド.....	A-5
アクセス系イベント : ABP1, ABP2コマンド.....	A-7
実行系イベント : IBP1, IBP2コマンド.....	A-8
ヘルプ : HELPコマンド.....	A-9
ポート入力 : INB, INH, INW, INDコマンド.....	A-10
初期化 : INITコマンド.....	A-11
JTAGリード : JREADコマンド.....	A-12
デバッガキャッシュの解除 : NCDコマンド.....	A-13
デバッガキャッシュの設定 : NCDコマンド.....	A-14
ソフトブレーク禁止領域の設定 : NSBPコマンド.....	A-15
ソフトブレーク禁止領域の解除 : NSBPDコマンド.....	A-16
強制ユーザ領域の設定 : NROMコマンド.....	A-17
強制ユーザ領域の解除 : NROMコマンド.....	A-18
ポート出力 : OUTB, OUTH, OUTW, OUTDコマンド.....	A-19
CPUリセット : RESETコマンド.....	A-20
E. ROMの環境設定 : ROMコマンド(RTE-1000-TP用コマンド).....	A-21
E. ROMの環境設定 : ROM1..ROM4コマンド(RTE-2000(H)-TP用コマンド).....	A-22
TLB : TLB32, TLB64コマンド.....	A-24
SFR : SFRコマンド.....	A-25
シンボル : SYMFILE, SYM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。

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。)

b p o p t コマンド

[書式]

bpopot [seq | [aand|aor] [iand|ior]]

[パラメータ]

seq: シーケンシャル条件を設定します。シーケンシャル条件は、abp1または、ibp1が発生後、abp2または、ibp2の条件成立でブレークします。

[aand|aor]: abp1, abp2の条件を設定します。

aand: abp1とabp2が同時に成立した時にブレークします。

ior: abp1または、abp2のどちらかが成立した時にブレークします。

[iand|ior]: ibp1, ibp2の条件を設定します。

iand: ibp1とibp2が同時に成立した時にブレークします。

ior: ibp1または、ibp2のどちらかが成立した時にブレークします。

[機能]

ブレーク用のイベントの条件を設定します。

ibp1, 2は実行系のイベント、abp1, 2はアクセス系のイベントです。

ibp1, 2, abp1, 2の設定方法はそれぞれのコマンドを参照ください。

[入力例]

bpopot aor

abp1とabp2をオアの条件で使用します。

bpopot seq

abp1, 2, ibp1, 2をシーケンシャル条件で使用します。

cacheinit, cacheflushコマンド

[書式]

```
cacheinit  
cacheflush [ADDRESS [LENGTH]]
```

[パラメータ]

cacheinit キャッシュの初期化を行います。ライトバックは行いませんので、キャッシュの内容は破棄されます。

cacheflush 指定した範囲のキャッシュのフラッシュを行います。ライトバックが指定されている場合は、ライトバックサイクルが発生します。

ADDR: 開始アドレスを16進数で指定します。

LENGTH: フラッシュする空間のバイト数を16進数で指定します。

[機能]

キャッシュ操作の為のコマンドです。

[入力例]

```
cacheflush 80000000 1000  
flush cache addr=80000000 len=00001000  
0x80000000 0x1000バイトのキャッシュの内容をフラッシュします。
```

env, ememstatコマンド

[書式]

```
env [[!]auto] [[!]nmi] [jtag[xxx][. [yyy]] {M|K}] [[!]verify]
  [[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]
  [[!]cresetb] [[!]resetb] [pclock1|pclock2|pclock4]
  [io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]
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。

[[!]nmi: NMI端子のマスク条件を指定します。!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。

[[!]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からリード(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。

[[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]:

外部割込みのマスク条件を指定します。!はマスクしないを意味します。

[[!]cresetb] [[!]resetb]: RESET端子のマスク条件を指定します。!はマスクしないを意味します。

CresetbがColdResetB端子を、resetbがResetB端子を意味します。

[pclock1|pclock2|pclock4]: 常にデフォルトのpclock4でご使用ください。

[io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]: BKTGIO_L端子のモードを指定します。

io_nouse: 使用しません。

io_brkout: ブレーク出力

io_brkin: ブレーク入力

io_trgout: 設定しないでください。

io_trgin: 設定しないでください。

[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。

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(初期値の状態)

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:
BKTGIO_L  = Present
Cotrol Unit = Present
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK = 12.5MHz (jtag12.5M)
Verify     = verify off (!verify)
Signals Mask:
INT0       = NO MASK (!int0)
INT1       = NO MASK (!int1)
INT2       = NO MASK (!int2)
INT3       = NO MASK (!int3)
INT4       = NO MASK (!int4)
TIMER      = NO MASK (!timer)
NMI        = NO MASK (!nmi)
COLDRESETB = NO MASK (!cresetb)
RESETB     = NO MASK (!resetb)
Trace UNIT:
TRCCLK Mode = PClock 1/4 (pclock4)
BKTGIO_L Mode= not use (io_nouse)

```

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:
BKTGIO_L  = Present
Cotrol Unit = Present
CPU Settings:
Auto Run   = ON (auto)
JTAGCLOCK = 25MHz (jtag25M)
Verify     = verify off (!verify)
Signals Mask:
INT0       = NO MASK (!int0)
INT1       = NO MASK (!int1)
INT2       = NO MASK (!int2)
INT3       = NO MASK (!int3)
INT4       = NO MASK (!int4)
TIMER      = NO MASK (!timer)
NMI        = NO MASK (!nmi)
COLDRESETB = NO MASK (!cresetb)
RESETB     = NO MASK (!resetb)
Trace UNIT:
TRCCLK Mode = PClock 1/2 (pclock2)
BKTGIO_L Mode= not use (io_nouse)

>ememstat
Board_num  EMEM_Size  ROM_Probe
=====
ROM1       8Mbyte     Extend Type 2K    << 実装されているモジュールの状態を表示します。
ROM2       32Mbyte     Extend Type 2K    << 実装されているモジュールの状態を表示します。

```

[入力例]

```

env nmi verify timer
    NMIとタイマ割込みをマスク、verify ONを指定をします。
env jtag40m
    JTAGクロックを40MHzに設定します。

```

abp1, abp2コマンド

[書式]

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

[パラメータ]

ADDR: アドレスを16進数で指定します。

AMASK: ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。

data DATA [DMASK]|nodata: データ条件を指定します。

data DATA: dataを前置し、データを16進数で指定します。

DMASK: DATAに対するマスクを指定します。ビット単位で'1'でマスクします。

nodata: データの指定をしません。

asid ASID|noasid: asidを指定します。

asid ASID: ASIDを比較対象に含めます。

noasid: ASIDを比較対象にしません。

aeq|aneq: アドレスの条件を指定します。aeqで指定値を、aneqで指定値の否定を条件にします。

deq|dneq: データの条件を指定します。deqで指定値を、dneqで指定値の否定を条件にします。

read|write|acc: ステータス条件を指定します。

read: ステータス条件としてリードサイクルを指定します。

write: ステータス条件としてライトサイクルを指定します。

acc: ステータスの指定を条件から削除します。

nosize|byte|hword|word|dword: アクセスサイズ条件を指定します。

nosize: アクセスサイズは比較しません。

byte: アクセスサイズとしてバイト条件を指定します。

hword: アクセスサイズとしてハーフワード条件を指定します。

word: アクセスサイズとしてワード条件を指定します。

dword: アクセスサイズとしてダブルワード条件を指定します。

abp{1|2} /del: それぞれの条件を削除します。

[機能]

アクセスサイクルのブレイク用のイベントを指定します。

[使用例]

```
abp1 1000 0 deta 5555 0 hword read
1000h番地からハーフワードで5555hをリードしたサイクルをブレイク条件に指定します。
```

[備考]

abp1とabp2の組み合わせ条件はbpoptで指定します。

i b p 1, i b p 2 コマンド

[書式]

```
ibp{1|2} [ADDR [AMASK]] [asid ASID|noasid] [aeq|aneq]  
ibp{1|2} /del
```

[パラメータ]

ADDR: アドレスを16進数で指定します。
AMASK: ADDRに対するマスクを指定します。ビット単位で'1'でマスクします。
asid ASID|noasid: ASIDを指定します。
asid ASID: ASIDを比較対象に含めます。
noasid: ASIDを比較対象にしません。
aeq|aneq: アドレスの条件を指定します。aeqで指定値を、aneqで指定値の否定を条件にします。
ibp{1|2} /del: それぞれの条件を削除します。

[機能]

実行アドレスのイベントを指定します。

[使用例]

```
ibp1 1000 0  
1000hの命令実行をマスクなしでブレイク条件として指定します。  
ibp2 1000 0ff  
1000hの下位8bitをマスクした実行アドレスをブレイク条件として指定します。  
ibp2 1000 0 asid 10  
asid=10hで1000hの命令実行をマスクなしでブレイク条件として指定します。
```

[備考]

ibp1とibp2の組み合わせ条件はbpoptで指定します。

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。

inb, inh, inw, indコマンド

[書式]

inb [ADDR]

inh [ADDR]

inw [ADDR]

ind [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, indは、アクセスサイズを区別して、リードを行ないます。

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

[使用例]

inb b0000000

b0000000Hからバイト(8-bit)でリードします。

inh 0000000

b0000000Hからハーフワード(16-bit)でリードします。

inw 0000000

b0000000Hからワード(32-bit)でリードします。

ind 0000000

b0000000Hからロングワード(64-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。

[使用例]

```
jread a0000000 100
```

a0000000h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、デバッガ内部に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。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。

[使用例]

```
nc b8000000 100000
```

b8000000hから100000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 b8000000 100000  
No Memory Cache Area  
No. Address Length  
1 b8000000 00100000
```

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: 削除するメモリアドレスの除外エリアのブロック番号を指定します。

[機能]

メモリアドレスの除外エリアを削除します。削除は各メモリアドレス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 をメモリアドレスの除外エリアから削除します。

```
>nc bf000000 100
No Memory Cache Area
No. Address Length
  1 bf000000 00000100
  2 b8000000 00100000
```

```
>ncd 1
No Memory Cache Area
No. Address Length
  1 b8000000 00100000
```

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 a0010000 20000
```

a0010000h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 a0010000 20000
```

```
Num Address Length
```

```
01 a0010000 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 a0100000 00200000  
02 a0400000 00010000
```

```
>nsbpd 1  
Num Address Length  
01 a0400000 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 a0000000 20000
```

a0000000h番地から20000バイトを強制ユーザ領域に指定します。

```
>nrom a0000000 20000
```

No.	Address	Length
1	a0000000	00020000

```
>nrom a0800000 40000
```

No.	Address	Length
1	a0000000	00020000
2	a0800000	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 a0800000 40000  
No. Address Length  
1 a0000000 00020000  
2 a0800000 00040000
```

```
>nromd 1  
No. Address Length  
1 a0800000 00040000
```

outb, outh, outw, outdコマンド

[書式]

```
outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
outd [[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、outdはロングワード単位でアクセスします。

[使用例]

```
outb b800000 12
    bfc00000hへバイトデータ : 12hを1ライトします。
outh b800000 1234
    bfc00000hへハーフワードデータ : 1234hをライトします。
outh b800000 12345678
    bfc00000hへワードデータ : 12345678hをライトします。
outd b800000 123456789abcdef0
    bfc00000hへワードデータ : 123456789abcdef0hをライトします。
```

resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

対象エミュレーション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 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16]
[bus8|bus16|bus32] [little|big]
```

[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、エラーになります。

LENGTH: エミュレートするROMのバイト数 (4バイトの境界単位で指定)

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。

little|big: romデータのエンディアンを指定します。ダウンロード時、little指定時は、ファイルのバイナリイメージをそのままの形で書き込みます。big指定時は、ROMのバスサイズに応じて、上位バイトと下位バイトのデータを入れ替えて書き込みます。

[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に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 bfc0000 40000 1m rom16 bus32 little
```

27C1024(1M-bitの16bit-ROM)でbfc00000hから256Kバイト(40000h)エミュレートします。この場合、バスが32Bitですので、16bit-rom2個をエミュレートします。Romのエンディアンはlittleです。(バイナリのイメージをそのままロードします。)

```
rom bfc00000 40000 2m rom rom16 bus16 big
```

27c2048(2M-bitの16bit-ROM)をbfc00000hから256Kバイト(40000h)エミュレートします。この場合、16bit-rom1個をエミュレートします。Romのエンディアンはbigです。(バイナリのイメージを上位と下位のバイトを入れ替えてロードします。)

<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のバス経路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）になっています。

[入力例]

rom1 bfc00000 400000 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	bfc00000 – bfffffff	16-bit	16-bit	32M-bit	禁止

rom2 bfc40000 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	bfc40000 – bfc7ffff	16-bit	16-bit	2M-bit	許可

rom1 bfc00000 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	bfc00000 – bfc7ffff	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。

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	

t l b 3 2, t l b 6 4 コマンド

[書式]

tlb32 [all|INDEX [MASK HI L00 L01]]

tlb64 [all|INDEX [MASK HI L00 L01]]

[パラメータ]

all: 全てのインデックスの表示を指定します。

INDEX: 特定のインデックスを指定します。

MASK HI L00 L01: 変更時、INDEXで指定したインデックスの内容を指定します。
4つセットで入力してください。

MASK: PageMaskを指定します。

HI: EntryHiを指定します。

L00: EntryLo0を指定します。

L01: EntryLo1を指定します。

[機能]

TLBの内容の表示と変更を行います。

tlb32は、CPUが32bitの時の内容です。

Tlb64は、CPUが64bitの時の内容です。

[使用例]

tlb32 all

全インデックスの内容を表示します。

Tlb32 10

TLB#=10の内容を表示します。

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。

[機能]

SFRレジスタ値の設定と表示を行います。

SFRコマンドでレジスタ名として使用できる名称は、パラメータなしでコマンドを入力することで確認できます。

[使用例]

sfr PIC0

PIC0レジスタの値を表示します。

sfr PIC0 2

PIC0レジスタに2hを設定します。

[制限事項]

VR4181Aにおいて、

SFRコマンドが対象とする内蔵周辺コントローラのレジスタ群はINTGSレジスタの設定によりアドレスが変更できますが、初期値（アドレスは0xbfa00000～）から変更した場合、

SFRコマンドは使用できなくなります。

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）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。
```

verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

ICEの制御用ファームウェアのバージョンを表示します。