

付録 . A K I T - V 8 3 1 / 2 - T P 内部コマンド

本書は、K I T - V 8 3 1 / 2 - 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、ディ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ィバッガのマニュアルを参照ください。

P A R T N E R / W i nの場合

> &	< <	スルーコマンドへの移行します。
> # E N V	< <	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。

G H S - M u l t i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

アクセスブレークポイント: A B P , A B P 1 , A B P 2 , A B P 3 , A B P 4コマンド.....	A-3	
特殊レジスタ	: C M C R , D C T R , I C T Rコマンド.....	A-4
環境設定	: E N Vコマンド.....	A-5
ヘルプ	: H E L Pコマンド.....	A-6
I N P U T	: I N B , I N H , I N Wコマンド.....	A-7
初期化	: I N I Tコマンド.....	A-8
J T A Gリード	: J R E A Dコマンド.....	A-9
キャッシュ領域の解除	: N Cコマンド.....	A-10
キャッシュ領域に指定	: N C Dコマンド.....	A-11
O U T P U T	: O U T B , O U T H , O U T Wコマンド.....	A-12
C P Uリセット	: R E S E Tコマンド.....	A-13
E . R O Mの設定	: R O Mコマンド.....	A-14
S F R	: S F Rコマンド.....	A-15
シンボルの読み込み	: S Y M F I L E , S Y Mコマンド.....	A-16
トリガポイントの指定	: T Pコマンド.....	A-17
トレース停止ポイントの指定	: T E Pコマンド(V832のみ).....	A-18
トレース開始ポイントの指定	: T S Pコマンド.....	A-19
トレースデータ条件の設定	: T D 1 , T D 2コマンド.....	A-20
トレースの設定&開始	: T R O Nコマンド.....	A-21
トレースの強制終了	: T R C O F Fコマンド.....	A-23
トレースの表示	: T R A C Eコマンド.....	A-24
バージョン表示	: V E Rコマンド.....	A-26

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。ご使用になるディバッガが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、ディ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-V831/2-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, abp3, abp4 コマンド

[書式]

```
abp[1|2|3|4] [ADDR] [io|mem]
abp [ADDR] [io|mem] /del
abp{1|2|3|4} /del
abpd{1|2|3|4}
```

[パラメータ]

abp: abpコマンドを指定します。abpで入力した場合、未使用のチャンネルを自動的に使用します。
明示的にチャンネルを指定する場合は、abp1,abp2,abp3,abp4を使用します。

ADDR: アドレスを16進数で指定します。

io: i/o空間のアクセスを条件に指定します。

mem: memory空間のアクセスを条件に指定します。

/del: 指定した条件を解除します。

abpdx: xで指定したabpチャンネルを削除します。

[機能]

4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。

[入力例]

```
abp 1000 mem
          1000h番地のメモリアクセスにブレークを設定します。
abp2 2000 io
          1000h番地のioアクセスにブレークを設定します。
abp 2000 io /del
          1000h番地のioアクセスに設定したブレーク条件を解除します。
abp1 /del
          abp1の条件を解除します。(abpd1と同じです)
abpd1
          abp1の条件を解除します。(abp1 /delと同じです)
```

cmcr, dctr, ictr コマンド

cmcr コマンド

[書式]

cmcr [=]value

[機能]

CMCR (キャッシュメモリ・コントロールレジスタ) に値を設定します。

dctr コマンド

[書式]

dctr [all]

[機能]

DCTR レジスタを表示します。

レジスタは256個ありますが、バリッドビットが有効になっているレジスタのみ表示します。

ただし、all を指定した場合は、全てのレジスタ値を表示します。

DCTR レジスタは、I/O 空間 f2000000h-f200ffffh にマップされています。

ictr コマンド

[書式]

ictr [all]

[機能]

ICTR レジスタを表示します。

レジスタは128個ありますが、バリッドビットが有効になっているレジスタのみ表示します。

ただし、all を指定した場合は、全てのレジスタ値を表示します。

ICTR レジスタは、I/O 空間 fa000000h-fa00ffffh にマップされています。

envコマンド

[書式]

```
env [!]auto [!]reset [!]nmi [!]hldrq [!]int{00|01|02|03} [!]int{10|11|12|13}
[jtag{25|12}] [!]verify [inone|istack|iaddr ADDR]
```

[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を行う場合に[Auto],行わない場合に[!auto]を指定します。
- [!]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。
- [!]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。
- [!]hldrq: HLDQR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。
- [!]int{00|01|02|03}: 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。
- [!]int{10|11|12|13}: int10-13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{12|25}: N-WireのJTAGクロック(12.5MHz|25MHz)指定します。Rte4win32 Ver4.37以降の初期値は12.5MHzです。それ以前のバージョンでは、25MHzです。
- [!]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。

[inone|istack|iaddr ADDR]: V832の内蔵命令RAMをアクセスする時に必要なワーク領域を指定します。モニタが使用後は、元のデータに戻します。ブロック転送可能なRAMを指定して下さい。

- inone: 指定されていません。
- istack: 現在のスタックからマイナス方向に32バイト使用します。
- iaddr ADDR: ADDRで指定したアドレスからプラス方向に32バイト使用します。

[機能]

envコマンドは、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。

```
CPU Settings:
Auto Run      = ON (auto)
JTAGCLOCK    = 12.5MHz (jtag12)
Verify       = verify off (!verify)
Signals Mask:
INT00..13    = NO MASK (!int00..13)
NMI          = NO MASK (!nmi)
RESET        = NO MASK (!reset)
HLDQR        = NO MASK (!hldrq)
IRAM Settings:
IRAM work    = stack (istack)
```

[入力例]

```
env reset !nmi
RESETをマスクし、NMIをマスクしません。
```

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、I/Oリードを行ないます。

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

アドレスが省略すると、前回のアドレスが指定されたものとみなします。

[使用例]

inb 1000

1000Hからバイト(8-bit)でI/Oリードします。

inh 1000

1000Hからハーフワード(16-bit)でI/Oリードします。

inw 1000

1000Hからワード(32-bit)でI/Oリードします。

initコマンド

[書式]

init

[パラメータ]

なし

[機能]

KIT-V831/2-TP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。

j r e a dコマンド

[書式]

j read [ADDR [LENGTH]]

[パラメータ]

ADDR: アドレスを 1 6 進数で指定します。

LENGTH: 読み出すバイト数を 1 6 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)から読み出す為のコマンドです。
(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]

```
j read ffff0000 100
```

ffff0000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 - V 8 3 1 / 2 TPではメモリ参照の高速化を図るため、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。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 3 2 バイトです。

[使用例]

```
nc 10000 1000
          10000h番地から1000h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

>ncd 2
No Memory Cache Area
No. Address Length
1 020000 000100
```

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、I/Oライトを行いません。
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。
アドレス、データが省略すると、前回の値が指定されたものとみなします。

[使用例]

```
outb 1000 12
    1000Hへバイトデータ：12hをI/Oライトします。
outh 1000 1234
    1000Hへハーフワードデータ：1234hをI/Oライトします。
outh 1000 12345678
    1000Hへワードデータ：12345678hをI/Oライトします。
```

resetコマンド

[書式]

reset

[パラメータ]

なし

[機能]

KIT-V831/2-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 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m] [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のバイト数を指定します。(4バイトの境界で指定)

512k|1m|2m|4m|8m|16m: エミュレートするROMのBitサイズを指定します。

512K-bitから16M-bitまでのサイズが指定できます。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 C0000 40000 1m rom16 bus16

27C1024(1M-bitの16bit-ROM)を0xc0000から256Kバイト(40000)エミュレートします。

この場合、結果的に16bit-romを2個エミュレートします。

rom fff80000 80000 2m rom rom16 bus32

27c2048(2M-bitの16bit-ROM)を0xffff80000から512Kバイト(80000)エミュレートします。

この場合、結果的に16bit-ROMを2個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。

<V831>

リード・ライトレジスタ:

IGP BCTC DBC DRC PRC ASIM00 ASIM01 CSIMO SIOO
 BRGO BPRMO TMC1 TMC4 TOC1 TOVS PORT PM PC CGC
 IMR IMOD PWCO PWC1 PIC RFC DSA0H DSA1H DSA2H DSA3H
 DSA0L DSA1L DSA2L DSA3L DDA0H DDA1H DDA2H DDA3H
 DDA0L DDA1L DDA2L DDA3L DBC0H DBC1H DBC2H DBC3H
 DBC0L DBC1L DBC2L DBC3L DCHCO DCHC1 DCHC2 DCHC3 DC
 CM4 CC10 CC11 CC12 CC13 TUM1

ライトオンリーレジスタ:

TXSOL ICR TXS0

リードオンリーレジスタ:

ASISO RXBOL IRR RXBO TM1 TM4

<V832>

リード・ライトレジスタ:

PORT PM PC
 BCTC DBC PWCO PWC1 RFC PRC
 DSA0H DSA0L DDA0H DDA0L DBC0H DBC0L DCHCO
 DSA1H DSA1L DDA1H DDA1L DBC1H DBC1L DCHC1
 DSA2H DSA2L DDA2H DDA2L DBC2H DBC2L DCHC2
 DSA3H DSA3L DDA3H DDA3L DBC3H DBC3L DCHC3 DC
 TOVS TUM1 TMC1 TOC1 CC10 CC11 CC12 CC13 TMC4 CM4
 ASIM00 ASIM01 CSIMO SIOO BRGO BPRMO
 IGP IMR IMOD
 CGC PMR
 PORTA PAM PAC PORTB PBM PBC
 PICO PIC1 SDC

ライトオンリーレジスタ:

TXS0 TXSOL ICR SDM

リードオンリーレジスタ:

TM1 TM4 ASISO RXBO RXBOL IRR

[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 IGP

IGPレジスタの値を表示します。

sfr IGP 2

IGP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。

t p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。

[使用例]

tp ffff0000

ffff0000h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を解除して下さい。

t e pコマンド(V832のみ)

[書式]

```
tep [ADDR] [/del]
```

[パラメータ]

ADDR: アドレスを16進数で指定します。

/del: 指定したアドレスを解除します。

[機能]

トレースの停止ポイント(アドレス)を指定します。

[使用例]

```
tsp ffff0000
```

ffff0000hの命令実行をトレースの開始アドレスとして指定します。

[備考]

トレース情報は、CPUからの出力時点でオーバーフローを起こす場合があります。このような場合、トレースしたい事象の直前でトレースを開始するように設定することで、目的に近辺でのオーバーフローを回避できます。

開始アドレスを指定しない場合は、tronコマンドを発行した時点で強制的に開始します。

このコマンドで指定した開始アドレスは、tronを発行した時点で有効になります。

t s pコマンド

[書式]

tsp [ADDR] [/del]

[パラメータ]

ADDR: アドレスを 16 進数で指定します。

/del: 指定したアドレスを解除します。

[機能]

トレースの開始ポイント (アドレス) を指定します。

[使用例]

tsp ffff0000

ffff0000hの命令実行をトレースの開始アドレスとして指定します。

[備考]

トレース情報は、CPUからの出力時点でオーバーフローを起こす場合があります。このような場合、トレースしたい事象の直前でトレースを開始するように設定することで、目的に近辺でのオーバーフローを回避できます。

開始アドレスを指定しない場合は、tronコマンドを発行した時点で強制的に開始します。

このコマンドで指定した開始アドレスは、tronを発行した時点で有効になります。

td1, td2 コマンド

[書式]

```
td1 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]  
td2 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]
```

[パラメータ]

DADDR: アドレスを16進数で指定します。

4バイトのバウンダリに補正されます。

ignore|ioread|iowrite|ioacc|memread|memwrite|memacc: ステータスを指定します。

ignore: don't care

ioread: I/O空間へのリード

iowrite: I/O空間へのライト

ioacc: I/O空間へのリード/ライト

memread: memory空間へのリード

memwrite: memory空間へのライト

memacc: memory空間へのリード/ライト

/del: 設定の解除

[機能]

トレースに取り込むデータサイクルの条件を設定します。

[使用例]

```
td1 fe000000 memread
```

fe000000h番地のからのメモリリードをトレースします。

tronコマンド

[書式]

```
tron [DELAY] [[!]delay] [noreal|real] [noignore|ev{[0]..|[8]}] [noext|nega|posi]
```

[パラメータ]

DELAY = 0..1ffff: デレイカウンタです。

トリガ成立後にメモリの取り込むフレーム数を十進数で指定します。

[[!]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。

[noreal|real]:

トレースのモードを指定します。

real: リアルタイム実行モード

normal: 非リアルタイム実行モード (rte4win32 ver4.35以上で対応しています。)
この設定では、トレース取り込み中、ブレークが頻繁に介入しますので、実行速度が10倍以上遅くなる場合があります。

noignore|ev{[0]..|[8]}

トレースに取り込まないイベントを指定します。

noignore: 全てのイベントを取り込みます。通常指定です。

ev0..8:

ev0: Exception の発生情報を取り込みません。

ev1: Interrupt の発生情報を取り込みません。

ev2: Condition Jump の発生情報を取り込みません。

ev3: PC relative の発生情報を取り込みません。

ev4: JAL の発生情報を取り込みません。

ev5: RETI to の発生情報を取り込みません。

ev6: RETI from の発生情報を取り込みません。

ev7: Jump register indirect to の発生情報を取り込みません。

ev8: Jump register indirect from の発生情報を取り込みません。

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。

[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モードで無条件に1ffffサイクル分トレースします。

```
>tron delay 1ffff
Trace Settings:
Start Address= Force
Delay Count = 0001ffff
Trace Mode = Real Time (real)
Delay Mode = Enable (delay)
Ignore Event = None (noignore)
Ext Trigger = Disable (noext)
Data Trace 1 = Disable (ignore)
Data Trace 2 = Disable (ignore)
Trig Address = Disable
```

fe00000h番地の命令実行をトリガにして、100h番地IOREADをデータトレースします。トリガの取り込みサイクル(DELAY)は、ffffhを指定します。

```
>tp fe000000                                << トリガの指定
Trig Address = fe000000

rte3>td1 100 ioread                          << データトレースの指定
Data Trace 1 = 00000100 I/O Read (ioread)
Data Trace 2 = Disable (ignore)

>tron ffff                                    << トレースの開始
Trace Settings:
Start Address= Force
Delay Count = 0000ffff
Trace Mode = Real Time (real)
Delay Mode = Disable (!delay)
Ignore Event = None (noignore)
Ext Trigger = Disable (noext)
Data Trace 1 = 00004444 I/O Read (ioread)
Data Trace 2 = Disable (ignore)
```

t r c o f f 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] [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を指定します。

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 表示種別 (アセンブル) ... 逆アセンブルで表示します。

subNN: 実際に取り込まれる一つの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] : アセンブラモード

Cycle	Sub	Address	Code	Instruction	EXT	Stat
-000018	0000	ffffff0	20bcff4f	movhi 4ffffh,r0,r1	0000	FTRC
-000018	0001	ffffff4	21a00000	movea 0000h,r1,r1	0000	----
-00000e	0000	ffffff8	0118	jmp [r1]	0000	JREG
*-00000a	0000	4fff0000	1070	ldsr r0,DPC	0000	JREG
000002	0000	4fff0018	00a8f247	jr 4fff480ah	0000	JMPR
000002	0001	4fff480a	a0bf074e	movhi 4e07h,r0,r29	0000	----
000002	0002	4fff480e	bda30002	movea 0200h,r29,r29	0000	----
000002	0003	4fff4812	bddf7400	st.w r29,0074h[r29]	0000	----
000002	0004	4fff4816	1ddc7c00	st.w r0,007ch[r29]	0000	----
000006	0000	4fff481a	ffaf80b8	jal 4fff009ah	0000	JAL
000006	0001	4fff009a	1ddc0000	st.w r0,0000h[r29]	0000	----

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: 命令のニックネームまたは、バスの種類を表示します。

EXT: 外部入力端子EX13..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類を表示します。

RD#1: データトレース(dt1)のリードサイクルの発生

TRIG: トリガアドレスの発生

FAIL: トレースデータの取りこぼしが発生

JMPR: PC相対分岐命令の分岐元アドレスの発生

JAL: JAL命令による分岐元アドレス発生

RET11: RTEI命令による分岐元アドレスの発生

JREG1: レジスタ間接分岐命令による分岐元アドレスの発生

FTRC: トレースの開始

WR#1: データトレース(dt1)のライトサイクルの発生
RET12: RETI命令による分岐先アドレスの発生
JREG2: レジスタ間接分岐命令による分岐先アドレスの発生
INTR: マスカブル割込みによる分岐の発生
EXP: 例外事象、または、NMIによる分岐の発生
CJMP: 条件分岐命令による分岐元アドレスの発生
RD#2: データトレース(dt2)のリードサイクルの発生
WR#2: データトレース(dt2)のライトサイクルの発生

verコマンド

[書式]

ver

[パラメータ]

なし

[機能]

KIT-V831/2-TPのバージョンを表示します。