This document describes the internal commands of the RTE-V853-IE.   These commands can be used as through commands in each debug monitor.   Refer to the manual of the respective debug monitors for how to use the commands.

(Example)   PARTNER/Win

>&   <<   Selects the through command mode.
>#ENV   <<   Internal command
>&   <<   Exits the through command mode.

Commands

Caution   Do not use these commands when equivalent functions are supported by the debugger.   If you issue a command when the debugger in use has an equivalent function, contention will occur between them, and either the command or the equivalent function is likely to malfunction.

Command syntax

The internal commands of the RTE-V853-IE are explained using the following notation.

Command name   parameter

* A parameter enclosed in [and] is omissible.   The horizontal line (|) separates optional parameters.

The command name is an alphabetic string.   The command name string is separated from the first parameter by a space or tab character.   A parameter is specified using an alphabetic character string or hexadecimal number.   Parameters are separated by a space or tab character.   (No operator can be used for hexadecimal numbers.)

ABP, ABP1, ABP2, ABP3, and ABP4 commands

[Format]

    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]

[Parameters]

    ADDR :   An address value is specified in hexadecimal.
    DATA :   Data to be accessed is specified in hexadecimal.
    MASK :   A data mask is specified in hexadecimal.
    read|write|access   A type of status is specified.

            read   :   Data read
            write   :   Data write
            access : Data access

    byte|hword|word|nosize   The size of data to be accessed is specified.

            byte   :   Byte access
            hword :   Halfword access
            word   :   Word access
            nosize :   No access size

    /del   :   Any previous setting by the command is canceled.

[Function]

The ABP, ABP1, ABP2, ABP3, and ABP4 commands set or cancel an access break point.
There are four access break points.   The abp command specifies an unused break point
automatically.

- The abp1, abp2, abp3, and abp4 commands are used to specify a break point explicitly.
- The data mask is used to specify invalid bits in the data.
- If a mask bit is 1, the corresponding data bit will not be compared.   If the mask bits are
  ffffffff, for example, the accessed data will be utterly ignored.

[Example]

    abp 1020 0 ffffffff access hword

            A break occurs if a halfword at address 1020H is accessed (regardless of the bit
            configuration of the data).

    abp 1020 100 0 write word

            A break occurs if the word 100H is written to address 1020H.

    abp2 /del

            The abp2 command is canceled.

<u>ATP, ATP1, and ATP2 commands</u>

[Format]

    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]

[Parameters]

    ADDR :   An address value is specified in hexadecimal.

    DATA :   Data to be accessed is specified in hexadecimal.

    MASK :   A data mask is specified in hexadecimal.

    read|write|access   A type of status is specified.

              read   :   Data read

              write :   Data write

              access :   Data access

    byte|hword|word|nosize   The size of data to be accessed is specified.

              byte   :   Byte access

              hword :   Halfword access

              word   :   Word access

              nosize :   No access size

    /del   :   Any previous setting by the command is canceled.

[Function]

The ATP, ATP1, and ATP2 commands set or cancel an access trigger point.   There are two access trigger points.   The atp command specifies an unused trigger point automatically.

• The atp1 and atp2 commands are used to specify a trigger point explicitly.

• The data mask is used to specify invalid bits in the data.

• If a mask bit is 1, the corresponding data bit will not be compared.

  Example)   If the mask bits are ffffffff, the accessed data will be utterly ignored.

When any of these commands is issued, the trace buffer will be cleared, and tracing will newly begin.

[Example]

    atp 1020 0 ffffffff access hword

              Triggering occurs if a halfword at address 1020H is accessed (regardless of the bit configuration of the data).

    atp 1020 100 0 write word

              Triggering occurs if the word 100H is written to address 1020H.

    atp2 /del

              The atp2 command is canceled.

ATP3 command   A-5

[Format]

> 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]

[Parameters]

> in|out   An address range is specified.
>
> > in   :   Addresses in a specified range are assumed to be valid.
> >   ADDR <= trigger address <= HADDR
> > out :   Addresses out of a specified range are assumed to be valid.
> >   Trigger address < ADDR, HADDR < trigger address
>
> ADDR   : A lower address value is specified in hexadecimal.
> HADDR : An upper address value is specified in hexadecimal.
>
> <|>|<>|<=|>=|eq   An address condition is specified.
>
> > <   :   Addresses lower than the specified address are made valid.
> >   Trigger address < ADDR
> > >   :   Addresses higher than the specified address are made valid.
> >   Trigger address > ADDR
> > <>   :   Any address not equal to the specified address is made valid.
> >   Trigger address <> ADDR
> > <=   :   Addresses lower than or equal to the specified address are made valid.
> >   Trigger address <= ADDR
> > >=   :   Addresses higher than or equal to the specified address are made valid.
> >   Trigger address >= ADDR
> > eq   :   Any address equal to the specified address is made valid.
> >   Trigger address = ADDR
>
> noaddr   Address conditions are ignored.
>
> DATA   :   Data to be accessed is specified in hexadecimal.
> MASK   :   A data mask is specified in hexadecimal.
> read|write|access   A type of status is specified.
>
> > read   :   Data read
> > write :   Data write
> > access :   Data access

byte|hword|word|nosize   The size of data to be accessed is specified.

> byte   :   Byte access
>
> hword   :   Halfword access
>
> word   :   Word access
>
> nosize   :   No access size

/del   :   Any previous setting by the command is canceled.

[Function]

The ATP3 command sets or cancels a range-of-address access break point.   If in or out is specified as a address range condition, two address values, upper and lower, can be specified.   In any other address range specification, only one address value can be specified.

When the command is issued, the trace buffer will be cleared, and tracing will newly begin.

• The data mask is used to specify invalid bits in the data.

• If a mask bit is 1, the corresponding data bit will not be compared.   If the mask bits are ffffffff, for example, the accessed data will be utterly ignored.

[Example]

atp3 in 1020 1300 0 ffffffff access hword

> Triggering occurs if a halfword at an address between 1020H and 1300H is accessed (regardless of the bit configuration of the data).

atp3 <= 1020 100 0 write word

> Triggering occurs if the word 100H is written to address 1020H or lower.

env command

[Format]

> 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]

[Parameters]

> rom[36|64|128|256]
>
> > The size of internal ROM is specified.   The sizes that can be specified include 32K, 64K, 128K, and 256K.   ROM128 should be specified for the RTE-V853-IE.
>
> ram[1|2|3|4|6|8|10|12|16|20|24|28]
>
> > The size of internal RAM is specified.   The sizes that can be specified include 1K, 2K, 3K, 4K, 6K, 8K, 10K, 12K, 16K, 20K, 24K, and 28K.   RAM4 should be specified for the RTE-V853-IE.
>
> [[romless|single]
>
> > A CPU mode is specified.   The CPU mode that can be specified is the ROM-less mode or single-chip mode.
>
> [[!reset]
>
> > Whether to mask the RESET pin is specified.   Special character ! specifies not to mask the pin.
>
> [[!]wait]
>
> > Whether to mask the WAIT pin is specified.   Special character ! specifies not to mask the pin.
>
> [[!]nmi]
>
> > Whether to mask the NMI pin is specified.   Special character ! specifies not to mask the pin.
>
> [[!]hldrq]
>
> > Whether to mask the HLDRQ pin is specified.   Special character ! specifies not to mask the pin.
>
> [[!]stop]
>
> > For the RTE-V853-IE, keep this parameter at the initial value (to mask).
>
> [w0|w16|w32|w64|w128|w256]
>
> > A wait timeout value is specified.
> >
> > • w0 specifies to wait permanently.
> >
> > • The other values specify that a timeout is to occur after 16 to 256 clock cycles.

[Function]

The env command is used to specify various environmental values for the emulation CPU of the RTE-V853-IE.   A parameter may be omitted if it is unnecessary.

• Parameters may be written in any order.

• If parameters that are mutually exclusive are coded, the last to appear is effective.

• After the init command is issued, the initial values are as follows:

Built-in ROM capacity :   128 K

Built-in RAM capacity :   4 K

CPU mode   :   Single-chip mode

Pin masks   :   No pin is masked.

Wait time   :   256 clock pulses

[Example]

env single !nmi

The single-chip mode is selected.   The NMI pin is not masked.

env romless rom128 w16

The ROM-less mode is selected.   The capacity of built-in ROM is set to 128 K.   The wait time is specified to be 16 clock pulses.

<u>**help command**</u>   A-9

[Syntax]

> help [command]

[Parameters]

> command: The name of the command you want to be helped with. When the command
> name is omitted, the command list is displayed.

[Function]

> The help command displays the help message for each specified command.

[Example]

> help map
>
> >> The information about the map command is displayed.

<u>**init command**</u>

[Syntax]

init

[Parameters]

None

[Function]

The init command initializes the RTE-V853-IE. The mapping information and environmental values are initialized. The rejection area of the memory cache is not initialized.

map command

[Syntax]

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

[Parameters]

> ADDR   :   The address of a location where mapping is to begin
>
> LENGTH   :   The number of bytes to be mapped
>
> [guard|ram|rom|target]:   The mapping attributes are specified.
>
> > guard   :   Specifies a guarded area.   An attempt to access the guarded area results in an error being detected.
> >
> > ram   :   Specifies that the emulation memory be treated as RAM.
> >
> > rom   :   Specifies that the emulation memory be treated as ROM.
> >
> > target   :   Specifies that the memory of the target system be used.

[Function]

> The map command carries out memory mapping.   The V853 emulation memory can be assigned in address spaces above and below 1M.
>
> • The size of memory assignment can be specified in 64-Kbyte units.

[Example]

> map 100000 80000 ram
>
> > 512 Kbytes starting at 100000h are assigned as emulation RAM.
>
> map 0 100000 guard
>
> > 1 Mbyte starting at address 0h is assigned as a guard area.

nc command

[Syntax]

> nc [[ADDR [LENGTH]]

[Parameters]

> [ADDR]   :   The start address of a memory cache rejection area is specified.
>
> [LENGTH] :   The number of bytes in the memory cache rejection area is specified.   The
> default value is 32 bytes.   The minimum value that can be specified is 32
> bytes.

[Function]

> The RTE-V853-IE has an 8-block x 32-byte memory read cache to speed memory reference.
> When memory reference is repeated at the same address, an actual read operation may not
> be carried out.   This cache operation conflicts with the actual operation of an I/O unit if it is
> mapped in memory.   In such a case, the nc command can be used to specify a memory
> cache rejection area.   Up to 8 blocks can be specified as a memory cache rejection area.
> The minimum acceptable block size is 32 bytes.   An address space above 0fff000 is always
> a memory cache rejection area.   So, it is unnecessary to specify this portion as a no-
> memory cache area.

[Example]

> nc 10000 1000
>
> > A 1000-byte area starting at address 10000 is specified as a memory cache
> > rejection area.
>
> >nc 10000 1000
> No Memory Cache Area
> No. Address   Length
> 1   010000   001000
> 2   fff000   001000

ncd command

[Syntax]

>   ncd   block-number

[Parameters]

>   block-number:　The block number for a memory cache rejection area to be deleted is specified.

[Function]

>   The ncd command deletes a memory cache rejection area.　The memory cache rejection area to be deleted is specified using a block number.

[Example]

>   ncd 2
>
>   >   Block 2 is deleted from a memory cache rejection area.
>
>   >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 command

[Syntax]

        reset

[Parameters]

        None

[Function]

        The reset command resets the emulation CPU of the RTE-V853-IE.

rrmb command

[Syntax]

rrmb ADDR

[Parameters]

ADDR: The base address for the real-time RAM monitor is specified.

[Function]

The rrmb command specifies the base address for the real-time RAM monitor. A 1K-byte area starting at the address specified here can be referenced using rrm command, even when the CPU is running.

[Example]

rrmb 10000

A 1-Kbyte area starting at address 10000 is specified as the real-time RAM monitor area.

<u>**rrm command**</u>   A-16

[Syntax]

rrm [ADDR[LENGTH]]

[Parameters]

ADDR   :   The memory reference start address in the real-time RAM monitor is specified.

LENGTH :   The number of bytes (up to 256 bytes) to be read is specified.

[Function]

The rrm command accesses a location in the real-time RAM monitor memory area.   The length can be up to 256 bytes.

[Example]

rrm 10000 30

30H bytes are read from the real-time RAM monitor, starting at address 10000.

sfr command

[Format]

> sfr [reg] [VAL]

[Parameters]

> VAL : An SFR register value is specified in hexadecimal.
>
> reg : An SFR register name is specified. The names that can be used for registers are listed below.

Read/write registers

> P0 P1 P2 P3 P4 P5 P6 P9 P11 PM0 PM1 PM2 PM3 PM4 PM5 PM6 PM9 PM11
>
> PMC0 PMC1 PMC2 PMC3 MM PMC11 PCM PUO DWC BCC PSC CKC SYS BRG0 BPRM0
>
> CSIM0 SIO0 BRGC1 BPRM1 CSIM1 SIO1 BRGC2 BPRM2 CSIM2 SIO2 CSIM3 SIO3
>
> ASIM00 ASIM01 ASIM10 ASIM11 OVIC11 OVIC12 OVIC13 OVIC14
>
> P11IC0 P11IC1 P11IC2 P11IC3 P12IC0 P12IC1 P12IC2 P12IC3
>
> P13IC0 P13IC1 P13IC2 P13IC3 P14IC0 P14IC1 P14IC2 P14IC3
>
> CMIC4 CSIC0 CSIC1 CSIC2 CSIC3 SEIC0 SRIC0 STIC0 SEIC1 SRIC1 STIC1 ADIC
>
> INTM0 INTM1 INTM2 INTM3 INTM4 TOVS
>
> TUM11 TMC11 TOC11 CC110 CC111 CC112 CC113
>
> TUM12 TMC12 TOC12 CC120 CC121 CC122 CC123
>
> TUM13 TMC13 TOC13 CC130 CC131 CC132 CC133
>
> TUM14 TMC14 TOC14 CC140 CC141 CC142 CC143
>
> TMC4 CM4 PWCM PWPR PWM0 PWM0L PWM1 PWM1L ADM0 ADM1 DACS0 DACS1 DAM

Write-only registers

> TXS0 TXS0L TXS1 TXS1L PRCMD

Read-only registers

> ASIS0 RXB0 RXB0L ASIS1 RXB1 RXB1L ISPR TM11 TM12 TM13 TM14 TM4
>
> ADCR0 ADCR0H ADCR1 ADCR1H ADCR2 ADCR2H ADCR3 ADCR3H
>
> ADCR4 ADCR4H ADCR5 ADCR5H ADCR6 ADCR6H ADCR7 ADCR7H

[Function]

> The sfr command sets and displays a value in an SFR register.

[Example]

> sfr TM4
>
> > A value in the TM4 register is displayed.
>
> sfr MM 0a0
>
> > The value 0a0H is set in the MM register.

<u>**symfile and sym commands**</u>

[Format]

> symfile FILENAME : Reads a symbol file (.sym).
> sym [NAME]   : Displays 30 symbols.

[Parameters]

> symfile : File name
> sym   : The first character string in symbols.

[Function]

> The symfile command reads a symbol file specified as FILENAME.   This command targets
> only global symbols.   The command also displays up to 30 symbols.

[Example]

> symfile c:¥test¥dry¥dry.sym
>> The symbol file :dry.sym is read from the directory c:¥test¥dry.
> sym m
>> Up to 30 symbols starting with character m are displayed.

<u>**time command**</u>

[Format]

time [sysclk]

[Parameters]

sysclk

[Function]

The time command displays the result of execution time measurement.   Each time the CPU starts running, the execution time measurement timer is initialized and started to count.   It continues to count, as long as the CPU is running.   The timer has a resolution of two clock cycles.

[Remark]

The measurement value contains the overhead time (several clock cycles) needed in starting and breaking measurement.

TP, TP1, and TP2 commands

[Format]

>   tp   [ADDR] [/del]
>   tp1 [ADDR] [/del]
>   tp2 [ADDR] [/del]

[Parameters]

>   ADDR :   An address value is specified in hexadecimal.
>   /del   :   Any previous setting by the command is canceled.

[Function]

>   The TP, TP1, and TP2 commands set and cancel execution trigger points.   There are two
>   execution trigger points.   The tp command specifies an unused trigger point automatically.
>   • The tp1 and tp2 commands are used to specify a trigger point explicitly.
>   When any of these commands is issued, the trace buffer will be cleared, and tracing will
>   newly begins.

[Example]

>   tp 1020
>
>>   Triggering occurs when an instruction at address 1020H is executed.

<u>**TRON command**</u>

[Format]

>   tron [DELAY] [add|cycle]
>
>   tron [x1|x2|x4|x8|x16|x32|x64|x128|x256|x512|x1k|x4k|x16k|x64k|x256k]

[Parameters]

>   DELAY=0..07fff   Delay counter
>
>>   The number of instruction cycles (delay counter) during which tracing should continue since the occurrence of triggering is specified in hexadecimal.   The trace buffer can hold data for up to 32K cycles.
>
>   x1..x1m   Trace timer counter frequency division ratio
>
>>   The frequency division ratio for the time tag counter clock is specified.   The setting x16 means that the timer unit is multiplied by 16.
>
>   add   The addition mode is specified.
>
>>   In this mode, a value from the previous cycle is accumulated.
>
>   cycle   The cycle mode is specified.
>
>>   The trace timer tag counter value is displayed for an individual cycle.

[Function]

>   The TRON command clears the trace buffer and begins recording trace data.

[Caution]

>   1. Even when the addition mode is specified for the timer counter, the time tag will be cleared at restart if a break occurs.

[Example]

>   tron 100 x16 cycle
>
>>   Tracing is terminated after data for 100h (256) cycles has been recorded since the occurrence of triggering.   The timer unit is multiplied by 16.   Execution results are written to the trace buffer in every cycle.

TROFF command

[Format]

troff

[Parameters]

None

[Function]

The TROFF command forcibly terminates tracing.

TRACE command

[Format]

      trace [POS] [asm]

[Parameters]

      POS= ±0..07fff   Read start position (with the trigger point or end point as 0)

                             The number of cycles counted from the beginning of the trace buffer is

                             specified in hexadecimal.

      asm   Display type (assembler)...displayed in a disassembly form

[Function]

      The TRACE command displays the contents of the trace buffer.

      • Tracing is terminated when this command is issued.

      • To restart tracing, it is necessary to issue the tron command.

[Display] **Assembly mode**

```
     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 : A trigger position is specified in hexadecimal with the first trigger cycle as 0.

neis   :   The PSW flag is displayed.

             n :   NMI flag

             e :   External interrupt flag

             i :   Exception flag

             s :   Saturation flag

Time   :   The trace timer tag counter is displayed in hexadecimal.

Ext   :   External data is displayed in bit units (in sequence of EXT0, 1, 2, 3 from the right).

Address :   An instruction execution address is displayed in hexadecimal.

Code   :   For execution tracing, an instruction code is displayed in hexadecimal. For data tracing, data is displayed in hexadecimal.

Operand :   An instruction is displayed in a disassembly form.

            Data is displayed in the following format:

            Read [address] <- data

            Write data -> [address]

            The number of digits in the data means the data size.

<<Caution>>

The first two frames of the time display contain error.   If add (addition mode) is specified in the tron command, the first frame is not subjected to addition.

<u>**ver command**</u>

    [Syntax]

        ver

    [Parameters]

        None

    [Function]

        The ver command indicates the version of the RTE-V853-IE.

APPENDIX B.   RTE-V853-IE USER INTERFACE CIRCUITS

P00-P07, P10-P17, P20-P27, P30-P37, P110-P117, WAIT-, RESET-

```
                    ○ +5 V
                    ⌇ 47 KΩ
CPU  ◄──────────────●──────────────► USER SYSTEM
```

P40-P47, P50-P57, P60-P3, P90-P96

```
                    ○ +5 V
                    ⌇ 47 KΩ
                              22 Ω
CPU  ◄──────────────●───────WWW──────► USER SYSTEM
```

P70-P77, ANO0, ANO1, AVREF1, AVREF2, AVREF3

```
CPU  ◄──────────────────────► USER SYSTEM
```

NMI

```
CPU ◄────[ EPM7032 ]──●───── USER SYSTEM
                      ⌇ 10 KΩ
                      ⏚
```

CLKOUT

```
              33 Ω
CPU  ──────WWW──────► USER SYSTEM
```

X1, X2

VHCU04 (2/6)   1 MΩ   100 Ω

1 MΩ

CPU

X1

USER SYSTEM

100 Ω

X2

1 MΩ

MODE0, MODE1

+5 V

47 KΩ

CPU ← EPM7032 ── MODE0

USER SYSTEM

NC. ────── MODE1

VDD, VSS

To level sense circuit

VDD

+5 V

10 KΩ

VSS (89 pin)

USER SYSTEM

VSS (excluding 89 pin)

VDD, VSS

CPU   +5 V

AVDD

NC. ────── AVDD

USER SYSTEM

AVSS

AVSS