APPENDIX A. KIT-VR5500-TP INTERNAL COMMANDS

This appendix describes the KIT-VR5500-TP internal commands. These commands can be used as through commands in the debugger. For an explanation of using through commands, refer to the manual provided with the debugger.

With PARTNER/Win

```
>&          << Enter through command mode.
>#ENV       << Enter an internal command.
>&          << Exit from through command mode.
```

With GHS-Multi

The through commands can be directly input in the target window after RTESERV has been connected.

Commands

Note These commands can be used only if the debugger does not provide equivalent functions. If these commands are issued when the debugger does provide equivalent functions, a contention may occur between KIT-VR5500-TP and the debugger, causing either device to malfunction.

Command syntax

The basic syntax for the KIT-VR5500-TP internal commands is described below:

command-name parameter(s)

* In parameter syntax, a parameter enclosed in brackets ([ ]) is omissible. A horizontal line (|) indicates that one of the parameters delimited by it must be selected.

A command name must be an alphabetic character string, and be separated from its parameter(s) by a space or tab. A parameter must be an alphabetic character string or hexadecimal number, and be delimited by a space or tab character. (A hexadecimal number cannot contain operators.)

bpopt command

[Format]

bpopt [[!]eve] [[!]eva]

[Parameters]

eve: Specifies event: eve as a break condition. ! clears the specified condition.

eva: Specifies event: eva as a break condition. ! clears the specified condition.

[Function]

Sets or clears an event condition as a break condition.

eve is an execution event and eva is an access event.

For how to set eve and eva, refer to the description of each command.

[Examples]

bpopt eve

Specifies eve as a break condition.

bpop !eve

Clears eve as a break condition.

Cacheinit and cacheflush commands

[Format]

cacheinit

cacheflush [ADDRESS [LENGTH]]

[Parameters]

cacheinit Initializes the cache. The contents of the cache will be lost because write back is not performed.

cacheflush Flushes the cache in a specified range. If write back is specified, a write back cycle is generated.

ADDR: Specifies a start address in hexadecimal number.

LENGTH: Specifies the number of bytes of the space to be flushed in hexadecimal number.

[Function]

This command is used to manipulate the cache.

[Examples]

cacheflush 80000000 1000

flush cache addr=80000000 len=00001000

Flushes the contents of cache of 0x80000000 0x1000 bytes.

env command

[Format]

env [[!]auto] [[!]nmi] [[!]int] [jtag{25|12|5|2|1|500|250|100}] [[!]verify]

[Parameters]

[!]auto: If a break point is encountered during execution, the break point causes a temporary break. Choose [Auto] to automatically perform the subsequent execution. Choose [!auto] to suppress it.

[!]nmi: Specifies whether the NMI pin is to be masked. Enter ! if it is not to be masked.

[!]int: Specifies that pin INTxx is to be masked. Enter ! if they are not to be masked.

jtag{25|12|5|2|1|500|250|100}:

Specifies the JTAG clock for N-Wire. Each number corresponds to the following JTAG clock.

[25MHz|12.5MHz|5MHz|2MHz|1MHz|500KHz|250KHz|100KHz]

> **Remark** Usually, use 25MHz or 12.5MHz. If the frequency lower than 1MHz is specified, the debugger might be slowed down in operation speed or might malfunction.

[!]verify: Specifies the verification after writing memory is set. Enter ! if it is not to be set.

> **Remark** The CPU also reads an area that emulates ROM (jread or equivalent). Therefore, this command is useful for testing the area during downloading. Note, however, that the processing speed slows down.

[Function]

The env command sets the emulation environment. Enter only those parameters that need to be changed. Parameters may be entered in any order. If the same parameter is entered twice, only the last entry is valid.

The initial values are as follows:

```
Probe:
 Unit     : RTE-1000-TP        << Displays the main chassis connected.
 Rom Probe : Extend Type       << Displays the ROM probe type connected.
 Emem Size : 32Mbyte           << Displays the size of emulation memory implemented
CPU Settings:
 Auto Run          = ON (auto)
 JTAGCLOCK         = 12.5MHz (jtag12)
 Verify            = verify off (!verify)
 Signals Mask:
 NMI               = NO MASK (!nmi)
 INT               = NO MASK (!int)
```

[Examples]

env !nmi verify

Specifies masking of NMI and ON of verify.

eva command

[Format]

eva [[!] ADDR [AMASK [[!] DATA [DMASK]|nodata] [byte|hword|word|dword]]]
[read|write|acc] [{noasid} | {asid ASID}]

eva [noaddr [[!] DATA [DMASK]|nodata] [byte|hword|word|dword]]]
[read|write|acc] [{noasid} | {asid ASID}

[Parameters]

ADDR:	Specifies an address in hexadecimal number. ! negates addr.
AMASK:	Specifies masking of ADDR. ADDR is masked with '1' in bit units.
noaddr:	Deletes specification of addresses from the condition.
[!] DATA [DMASK] | nodata:	Specifies a data condition.
DATA:	Specifies data in hexadecimal number. ! negates DATA.
DMASK:	Specifies masking of DATA. DATA is masked with '1' in bit units.
nodata:	Deletes specification of data from the condition.
byte|hword|word|dword:	Specifies an access size condition.
byte:	Specifies a byte condition as access size.
hword:	Specifies a half-word condition as access size.
work:	Specifies a word condition as access size.
dword:	Specifies a double-word condition as access size.
read|write|acc:	Specifies a status condition.
read:	Specifies a read cycle as a status condition.
write:	Specifies a write cycle as a status condition.
acc:	Deletes the specification of a status from the condition.
noasid | asid ASID:	
noasid:	Does not include ASID in subject to comparison.
asid ASID:	Includes ASID in subject to comparison.

[Function]

Specifies an event of an access cycle.

[Examples]

eva 1000 0 5555 0 hword read

Specifies the cycle in which 5555h is read in half-word units from address 1000h as an eva condition.

[Remark]

The event condition specified by eva can be used as a trigger condition for break or trace.
The specified event condition is used as a condition for break or trace, using bpopt or tron.

eve command

[Format]

eve [[!] ADDR [AMASK] [{noasid}|{asid ASID}]]

[Parameters]

ADDR: Specifies an address in hexadecimal number. ! negates addr.

AMASK: Specifies masking of ADDR. ADDR is masked with '1' in bit units.

noasid | asid ASID:

noasid: Does not include ASID in subject to comparison.

asid ASID: Includes ASID in subject to comparison.

[Function]

Specifies an event for an executable address.

[Examples]

eve 1000 0

Specifies execution of the instruction at address 1000h as an event without mask.

eve 1000 0ff

Specifies an executable address 1000h with the low-order 8 bits masked as an event.

eve 1000 asid 10

Specifies execution of the instruction at address 1000h with asid = 10h as an event.

[Remark]

The event condition specified by eve can be used as a trigger condition for break or trace.

The specified event condition is used as a condition for break or trace, using bpopt or tron.

help command

[Format]

help [command]

[Parameters]

command: Specifies the name of the command for which you required help. If this parameter is omitted, a list of commands is displayed.

[Function]

The help command displays a help message for a specified command.

[Examples]

help map

A help message for the map command is displayed.

inb, inh, inw, and ind commands

[Format]

inb [ADDR]
inh [ADDR]
inw [ADDR]
ind [ADDR]

[Parameters]

ADDR: This parameter specifies the address of an input port in hexadecimal notation.

[Function]

The inb, inh, inw, and ind commands read I/O space.
The inb command accesses I/O space in bytes, inh in half words, inw in words, and ind in long words.

[Examples]

inb b0000000
I/O space is read in bytes (8-bit units), starting at b0000000H.
inh 0000000
I/O space is read in half words (16-bit units), starting at b0000000H.
inw 0000000
I/O space is read in words (32-bit units), starting at b0000000H.
ind 0000000
I/O space is read in long words (64-bit unit), starting at b0000000H.

init command

[Format]

init

[Parameters]

None

[Function]

The init command initializes KIT-VR5500-TP. All environment values are initialized.
A memory cache rejection area is not initialized.

jread command

[Format]

jread [ADDR [LENGTH]]

[Parameters]

ADDR: Specifies an address in hexadecimal notation.

LENGTH: Specifies the number of bytes to be read, in hexadecimal notation. (Max: 100h)

[Function]

The jread command reads the ROM emulation area allocated by the ROM command, via JTAG (the CPU).

Access to the ROM emulation area by ordinary commands is performed directly on internal memory.

[Examples]

jread a0000000 100

100h bytes, starting at a0000000h, are read via JTAG.

nc command

[Format]

nc [[ADDR [LENGTH]]

[Parameters]

ADDR: Specifies the start address of a memory cache rejection area.

LENGTH: Specifies the length of the memory cache rejection area in bytes. The default value is 32 bytes. The allowable minimum value is also 32 bytes.

[Function]

To ensure quick memory access, KIT-VR5500-TP provides a memory read cache of 8 blocks * 32 bytes. When the same memory address is accessed more than once, the read operation is not actually performed. This cache operation conflicts with the actual operation when an I/O unit is mapped onto memory. In such a case, specify a memory cache rejection area by using the nc command. Up to eight blocks can be specified as a memory cache rejection area. The allowable minimum block size is 32 bytes.

[Examples]

nc b8000000 100000

A 100000-byte area, starting at b8000000h, is specified as a memory cache rejection area.

```
>nc b8000000 100000
No Memory Cache Area
No. Address  Length
1  b8000000     00100000
```

ncd command

[Format]

ncd block-number

[Parameters]

block-number: Specifies the block number for a memory cache rejection area to be deleted.

[Function]

The ncd command deletes a memory cache rejection area. Specify the block number corresponding to the memory cache rejection area to be deleted.

[Examples]

ncd 1

Block 1 is deleted from the memory cache rejection area.

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

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

nsbp command

[Format]

nsbp [[ADDR [LENGTH]]

[Parameters]

ADDR: Specifies the start address of a software break prohibition area.

LENGTH: Specifies the length software break prohibition area in bytes. The minimum unit of a specification area is the boundary of half word. The number of the areas which can be specified is a maximum of four.

[Function]

An area to forbid a software breakis specified. When a break point is specified, a debugger performs a memory test (write access) to an object address. The state of a memory changes by performing write access and it may stop reading the right data in a part of flash ROM. When such, please forbid a software break by this command. Usually, it is not necessary to specify.

[Examples]

nsbp a0010000 20000

A 20000-byte area, starting at a0010000h, is specified as a software break prohibition area.

```
>nsbp a0010000 20000
Num Address  Length
01  a0010000 00020000
```

nsbpd command

[Format]

nsbpd block-number

[Parameters]

block-number: Specifies the block of the software break prohibition area to be deleted.

/all: Specifies all software break prohibition area to be deleted.

[Function]

The nsbpd command deletes the software break prohibition area specified by nsbp.

[Examples]

nsbpd 1

Block1 is deleted from a software break prohibition area.

```
>nsbp
Num Address  Length
01  a0100000 00200000
02  a0400000 00010000

>nsbpd 1
Num Address  Length
01  a0400000 00010000
```

nrom command

[Format]

nrom [[ADDR [LENGTH]]

[Parameters]

ADDR: Specifies the start address of a forced user area.

LENGTH: Specifies the length of a forced user area in bytes. The number of the areas which can be specified is a maximum of four.

[Function]

The area is specified when the map of the part in ROM emulation area specified by ROM command is carried out to other resources on a user system. Usually, it is not necessary to specify.

[Examples]

nrom a0000000 2000

A 2000-byte area, starting at a0000000h, is specified as a forced user area.

```
>nrom a0000000 1000
No. Address    Length
1  a0000000   00001000

>nrom 10000 100
No. Address    Length
1  a0000000   00001000
2  a0010000   00000100
```

nromd command

[Format]

nromd block-number

[Parameters]

block-number: Specifies the block number for the forced user area to be deleted.

/all: Specifies all the forced user area to be deleted.

[Function]

The nromd command deletes the forced user area by nrom.

[Examples]

nromd 1

Block1 is deleted from the forced user area.

```
>nrom a0010000 8000
No. Address   Length
1  a0000000   00001000
2  a0010000   00008000

>nromd 1
No. Address   Length
1  a0010000   00008000
```

outb, outh, outw, and outd commands

[Format]

outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
outd [[ADDR] DATA]

[Parameters]

ADDR: Specifies the address of an output port in hexadecimal notation.
DATA: Specifies the data to be output in hexadecimal notation.

[Function]

The outb, outh, outw, and outd commands writes data to the I/O space.
The outb command accesses I/O space in bytes, outh in half words, outw in words, and outd in long words.

[Examples]

outb b800000 12
Byte data 12h is written to bfc00000h in the I/O space.
outh b800000 1234
Half word data 1234h is written to bfc00000h in the I/O space.
outh b800000 12345678
Word data 12345678h is written to bfc00000h in the I/O space.
outd b800000 123456789abcdef0
Long word data 123456789abcdef0h is written to bfc00000h in the I/O space.

reset command

[Format]

reset

[Parameters]

None

[Function]

The reset command resets the emulation CPU of KIT-VR5500-TP.

rom command

[Format]

rom [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [bus8|bus16|bus32]

[Parameters]

ADDR [LENGTH]: Specifies an area to be emulated.

ADDR: Specifies a start address. An error occurs if the specified start address does not match the lowest address of the ROM to be emulated (boundary of the ROM).

LENGTH: Number of bytes of the ROM to be emulated. (Must be specified in boundary units of 4 bytes.)

512k|1m|2m|4m|8m|16m|32m|64m|128m|256m: Specifies the bit size of the ROM to be emulated. Sizes from 512K bits to 256M bits can be specified. For the 27C1024, for example, specify 1M bits.

rom8|rom16: Specifies the number of data bits of the ROM to be emulated. Either 8 bits or 16 bits can be specified. If a DIP-32-ROM probe is used, choose rom8; if a DIP-40/42-ROM probe is used, choose rom16.

bus8|bus16|bus32: Specifies the ROM bus size in the system to be emulated. 8 bits, 16 bits, or 32 bits can be specified.

[Function]

The rom command sets the ROM emulation environment. Enter only the parameters that need to be changed. Parameters may be entered in any order. If the same parameter is entered twice, only the last entry is valid. The initial value of LENGTH is 0 (not used).

[Examples]

rom bfc0000 40000 1m rom16 bus32 little

The 256K bytes (40000h) of the 27C1024 (16-bit ROM with a size of 1M bit), starting at bfc00000h, are emulated. Consequently, one 16-bit ROM is emulated. The endian of Rom is little (the binary image is loaded as is).

rom bfc00000 40000 2m rom rom16 bus16 big

The 256K bytes (40000h) of the 27C2048 (16-bit ROM with a size of 2M bits), starting at bfc00000h, are emulated. Consequently, one 16-bit ROM is emulated. The endian of Rom is big (the binary image is loaded with the high-order and low-order bytes exchanged).

<Note>

Access to a rage specified by the rom command is a direct access to the emulation memory in the tool. Therefore, the processor may not correctly access the ROM even if it seems correct in terms of display. In this case, confirm by using the jread command, or write (download) data by setting verify to ON with the evn command.

tlb32 and tlb64 commands

[Format]

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

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

[Parameters]

all: Specifies display of all indexes.

INDEX: Specifies a specific index.

MASK HI L00 L01:

Specifies the contents of the index specified by INDEX for change. Input all four of these parameters as a set.

MASK: Specifies PageMask.

HI: Specifies EntryHi.

L00: Specifies EntryLo0.

L01: Specifies EntryLo1.

[Function]

Displays and changes the contents of TLB.

tlb32 is the contents when a 32-bit CPU is used.

Tlb64 is the contents when a 64-bit CPU is used.

[Examples]

tlb32 all

Displays the contents of all indexes.

Tlb32 10

Displays the contents of TLB# = 10.

symfile and sym commands

[Format]

symfile FILENAME

sym [NAME]

[Parameters]

symfile: Specifies file name.

sym: Specifies first character string in the symbols to be displayed.

[Function]

The symfile command reads symbols from the elf file specified by the FILENAME parameter.

Only global symbols can be read.

The sym command displays up to 30 symbols that have been read.

[Examples]

symfile c:\test\dry\dry.elf

Symbols are read from the elf file dry.elf in the c:\test\dry directory.

sym m

Up to 30 symbols that begin with "m" are displayed.

tron command

[Format]

tron [DELAY] [[!]delay] [[!]eve] [[!]eva] [noext|nega|posi]

[Parameters]

DELAY = 0..1ffff delay counter

Specifies the number of frames in memory that are to be loaded in response to a trigger, in decimal notation.

[!]delay: Specifies forced delay mode. Enter !delay to return to normal mode.
In forced delay mode, trace is started immediately after the TRON command and is forcibly stopped when tracing by the counts of the delay counter has been completed. In this mode, trigger events are ignored.

[!] eve Specifies event eve as a trace trigger. ! deletes specification.

[!] eva Specifies event eva as a trace trigger. ! deletes specification.

noext|nega|posi: The external input pin EXI0 can be specified as a trigger.

noext: EXI0 is not used as a trigger.

posi: The rising edge of EXI0 is specified as a trigger.

nega: The falling edge of EXI0 is specified as a trigger.

[Function]

The tron command clears the trace buffer and the settings of trace, and begins loading trace data.

[Examples]

tron delay 1ffff

Trace is unconditionally performed for 1ffff cycles in delay mode.
In this example, trace is started immediately after the tron command. Trace continues for 1ffff cycles and then stops.

tron !delay eve ffff

Clears the delay mode and starts trace using eve as a trigger point.
ffffh is specified as a cycle to load data after the trigger has been satisfied.
In this case, trace is started immediately after the tron command, passes the trigger point, continues for ffff cycles, and then stops. Consequently, execution history of ffff cycles before and after the trigger point can be traced.

[Remark]

For information on how to set eve and eva, refer to the description of each command.

troff command

[Format]

troff

[Parameters]

None

[Function]

The troff command forcibly terminates the loading of trace data.

trace command

[Format]

trace [POS] [all|pc|data] [asm] [subNN]

[Parameters]

POS=±0..1ffff Specifies the trace display start position in hexadecimal notation, assuming the vicinity of a trigger cycle or the ending cycle to be 0.

[all|pc|data] Specifies the cycle in loaded trace information that is to be displayed.

- all: All cycles
- pc: Execution cycles only
- data: Data cycles only

asm|ttag1|ttag2 Specifies the display type

- Asm: Displays disassembled listing
- ttag1: Displays disassembled listing and Time Tag in absolute time format.
- ttag2: Displays disassembled listing and Time Tag in relative time format.

Remark The ttag1|ttag2 specification is not available for RTE-100-TP.

subNN: The number of instructions to be disassembled in succession from an item of information to actually be loaded, in hexadecimal notation. The initial value is 80h (sub80).

[Function]

The trace command displays the contents of the trace buffer.

Issuing this command during trace terminates the recording process.

```
>trace -10 asm
  Cycle    Sub  Address    Code        Instruction              EXT Stat
 -00000d ----   bfc00000   0bf00100    j    bfc00400            1111 TPC
 -000002 ----   bfc00004   00000000   nop                      1111 NSEQ
  000001 ----   bfc00400   401a6000   mfc0  r26,$12            1111 NSEQ
  000001 0001 bfc00404   00000000   nop                      1111 -----
  000001 0002 bfc00408   001ad502   srl   r26,r26,14          1111 -----
  000001 0003 bfc0040c   335a0001   andi  r26,r26,1           1111 -----
  000001 0004 bfc00410   13400003   beq   r26,r0,bfc00420    1111 -----
  000001 0005 bfc00414   00000000   nop                      1111 -----
  000004 ----   bfc00420   0ff001cc    jal   bfc00730         1111 NSEQ
  000004 0001 bfc00424   00000000   nop                      1111 -----
  000007 ----   bfc00730   40806800   mtc0  r0,$13           1111 NSEQ
  000007 0001 bfc00734   00000011   mthi  r0                 1111 -----
  000007 0002 bfc00738   00000013   mtlo  r0                 1111 -----
  000007 0003 bfc0073c   0000e025   or    r28,r0,r0          1111 -----
  000007 0004 bfc00740   0000f025   or    r30,r0,r0          1111 -----
  000007 0005 bfc00744   3c0ab800   lui   r10,b800          1111 -----
  000007 0006 bfc00748   81421003   lb    r2,1003(r10)       1111 -----
  000007 0007 bfc0074c   00000000   nop                      1111 -----
  000007 0008 bfc00750   30420008   andi  r2,r2,8           1111 -----
  000007 0009 bfc00754   1c400002   bgtz  r2,bfc00760       1111 -----

>trace -10 ttag1
  Cycle    Sub  Address    Code        Instruction              EXT  Stat
 -00000d ----   bfc00000   0bf00100    j    bfc00400            1111 TPC
                    time= 000,000,000,000.0uS
 -000002 ----   bfc00004   00000000   nop                      1111 NSEQ
                    time= 000,000,000,000.5uS
  000001 ----   bfc00400   401a6000   mfc0  r26,$12            1111 NSEQ
                    time= 000,000,000,004.0uS
```

```
  000001 0001              bfc00404   00000000              nop
           1111 -----
  000001 0002 bfc00408   001ad502   srl  r26,r26,14         1111 -----
  000001 0003 bfc0040c   335a0001   andi r26,r26,1          1111 -----
  000001 0004 bfc00410   13400003   beq  r26,r0,bfc00420   1111 -----
  000001 0005 bfc00414   00000000   nop                     1111 -----
  000004 ---- bfc00420   0ff001cc    jal  bfc00730          1111 NSEQ
                    time= 000,000,000,005.2uS
  000004 0001 bfc00424   00000000   nop                     1111 -----
  000007 ---- bfc00730   40806800   mtc0 r0,$13             1111 NSEQ
                    time= 000,000,000,011.9uS
  000007 0001 bfc00734   00000011   mthi r0                 1111 -----
  000007 0002 bfc00738   00000013   mtlo r0                 1111 -----
  000007 0003 bfc0073c   0000e025   or   r28,r0,r0          1111 -----
  000007 0004 bfc00740   0000f025   or   r30,r0,r0          1111 -----

>trace -10 ttag2
  Cycle    Sub  Address    Code       Instruction           EXT  Stat
 -00000d ---- bfc00000    0bf00100   j    bfc00400         1111 TPC
 -000002 ---- bfc00004    00000000   nop                   1111 NSEQ
                    time= 000,000,000,000.5uS
  000001 ---- bfc00400    401a6000   mfc0 r26,$12          1111 NSEQ
                    time= 000,000,000,003.5uS
  000001 0001 bfc00404   00000000   nop                     1111 -----
  000001 0002 bfc00408   001ad502   srl  r26,r26,14         1111 -----
  000001 0003 bfc0040c   335a0001   andi r26,r26,1          1111 -----
  000001 0004 bfc00410   13400003   beq  r26,r0,bfc00420   1111 -----
  000001 0005 bfc00414   00000000   nop                     1111 -----
  000004 ---- bfc00420   0ff001cc    jal  bfc00730          1111 NSEQ
                    time= 000,000,000,001.2uS
  000004 0001 bfc00424   00000000   nop                     1111 -----
  000007 ---- bfc00730   40806800   mtc0 r0,$13             1111 NSEQ
                    time= 000,000,000,006.7uS
  000007 0001 bfc00734   00000011   mthi r0                 1111 -----
  000007 0002 bfc00738   00000013   mtlo r0                 1111 -----
  000007 0003 bfc0073c   0000e025   or   r28,r0,r0          1111 -----
  000007 0004 bfc00740   0000f025   or   r30,r0,r0          1111 -----
  000007 0005 bfc00744   3c0ab800   lui  r10,b800          1111 -----
```

Cycle: Relative positions in the trace buffer are displayed in hexadecimal notation. The vicinity of the trigger point or the trace end frame is assumed to be 0.

Sub: Cycle numbers generated by analyzing branching and number-of-executed-instruction information.

Address: Execution addresses or bus cycle addresses are displayed.

Code: Instruction code or bus cycle data is displayed.

Instruction: Instruction mnemonics or bus types are displayed.

EXT: The states of external input pins EXI3 to EXI0 are displayed as bit strings.

Stat: The types of trace packets on which display is based are displayed.

- TPC: Branch that cannot be traced from an instruction occurs.
- EXP: Occurrence of an exception
- LSEQ: Contiguous execution of 256 instructions or more occurs.
- NSEQ: Branch occurs.

time = Displays Time Tag

Remark The Time Tag is registered, when CPU outputs branch information. The output of branch information has some delay from the time of actual execution, and the delay might vary time to time. Thus, themeasurement value of Time Tag has some difference in its nature. Especially, please ignore the measurement result immediately after the execution, as it has unbounded difference.

tmode command

[Format]

tmode

[Parameters]

None

[Function]

The tmode command displays the setting status of the trace.

ver command

[Format]

ver

[Parameters]

None

[Function]

The ver command displays the version of KIT-VR5500-TP.