

APPENDIX A. KIT-VR5400-TP INTERNAL COMMANDS

This appendix describes the KIT-VR5400-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

Appendix A. KIT-VR5400-TP internal commands:	A-1
Commands:	A-1
Command syntax:	A-2
Option break:	bpop command.....	A-3
Cache operation:	cacheinit and cacheflush commands.....	A-4
Environment setting:	env command.....	A-5
Execution event:	eva command.....	A-7
Access event:	eve command.....	A-8
Help:	help command.....	A-9
Port input:	inb, inh, inw, and ind commands.....	A-10
Initialization:	init command.....	A-11
JTAG read:	jread command.....	A-12
Releasing of an area as a debugger cache area:	nc command.....	A-13
Specification of an area as a debugger cache area:	ncd command.....	A-14
Port output:	outb, outh, outw, and outd commands.....	A-15
CPU reset:	reset command.....	A-16
E.ROM setting:	rom command.....	A-17
TLB:	tlb32 and tlb64 commands.....	A-18
Symbols:	symfile and sym commands.....	A-19
Setting and start of trace:	tron command.....	A-20
Forcible termination of trace:	troff command.....	A-21
Trace display:	trace command.....	A-22
Confirmation of trace setting:	tmode command.....	A-24
Version display:	ver command.....	A-25

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-V831/2-TP and the debugger, causing either device to malfunction.

Command syntax

The basic syntax for the KIT-VR5400-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.

bpopt !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}] [[!]verify] [[!]hispeed]
```

[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[12|25]: Specifies the JTAG clock (12.5 MHz|25 MHz) for N-Wire. Initial value of rete4win32 is 12.5 MHz (Ver. 4.37 or later) or 25 MHz (before Ver. 4.37)).

[!]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.

[[!]hispeed]: Specifies high-speed mode to write data to memory. ! specifies the normal mode.

Remark The high-speed mode can be specified on condition that the ROM probe be connected. If this mode is specified, a control program temporarily located in the ROM emulated is executed in the foreground only when data of 128 bytes or more are contiguously written. A necessary condition for this command is that it can be correctly executed in the ROM space. Use this command after debugging of the hardware has been completed. In normal mode, data can be written to ROM via JTAG.

[work ADDR]: Specifies an area for clearing the cache or executing initialization processing. Be sure to specify the RAM on uncache immediately after starting the system. The VR5400 requires RAM on the user system for cache processing. The VR5400 requires RAM on the user system for cache processing. The monitor uses an area of 128 bytes from a specified address without saving the contents of the area.

Remark As this area is not used unless the cache area is accessed, confirm that the RAM can be accessed in the uncache area before accessing the cache area.

[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:

CPU Settings:

Auto Run = ON (auto)

JTAGCLOCK = 12.5MHz (jtag12)

Verify = verify off (!verify)

Download = Normal (!hispeed)

Signals Mask:

NMI = NO MASK (!nmi)

INT = NO MASK (!int)

Cache Clear Settings:

Work Memory = ffffffff (work ffffffff)

[Examples]

```
env !nmi verify work a0010000 hispeed
```

Specifies masking of NMI, ON of verify, 0xa0010000 as the work address, and high-speed mode.

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-VR5400-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-VR5400-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

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.

outw 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-VR5400-TP.

rom command

[Format]

rom [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m] [rom8|rom16] [bus8|bus16|bus32] [little|big]

[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:	Specifies the bit size of the ROM to be emulated. Sizes from 512K bits to 16M 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.
Little big:	Specifies the endian of the rom data. If little endian is specified during downloading, the binary image of a file is written as is. If big endian is specified, the data of the high-order bytes and low-order bytes are exchanged depending on the bus size of the ROM.

[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 EXIO can be specified as a trigger.

noext: EXIO is not used as a trigger.

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

nega: The falling edge of EXIO 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.

ffff 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= \pm 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 Display type (assemble)... disassembly and display

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.

[Display]

Assembler mode

```

-000005 0002 bfc035c8 34190001 ori    r25, r0, 1      0000 ----
-000005 0003 bfc035cc 3c01a000 lui    r1, a000    0000 ----
-000005 0004 bfc035d0 ac392780 sw     r25, 2780(r1)  0000 ----
-000005 0005 bfc035d4 0ff00da6 jal    bfc03698    0000 ----
-000005 0006 bfc035s8 00000000 nop                                0000 ----
-000002 ---- bfc03698 27bdfff8 addiu  r29, r29, fffffff8  0000 NSEQ
-000002 0001 bfc0369c afbf0004 sw     r31, 4(r29)    0000 ----
-000002 0002 bfc036a0 0ff00369 jal    bfc00da4    0000 ----
-000002 0003 bfc036a4 00000000 nop                                0000 ----
000001 ---- bfc00da4 401b6000 mfc0  r27, $12     0000 NSEQ
000001 0001 bfc00da8 00000000 nop                                0000 ----
000001 0002 bfc00dac 00000000 nop                                0000 ----
000001 0003 bfc00db0 2401ffe  addiu  r1, r0, fffffffe  0000 ----
000001 0004 bfc00db4 003b0824 and   r1, r1, r27     0000 ----
    
```

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.



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-VR5400-TP.