APPENDIX A. KIT-VR4120-TP INTERNAL COMMANDS

This appendix describes the KIT-VR4120-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-VR4120-TP and the debugger, causing either device to malfunction.

Command syntax

The basic syntax for the KIT-VR4120-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 [seq | [aand|aor] [iand|ior]]

[Parameters]

seq: Specifies sequential conditions. Sequential conditions take a break by condition formation of abp2 or ibp2 after abp1 or ibp1 occurring.

[aand|aor]: Specifies abp1 and abp2 conditions.

aand: Break is taken when abp1 and abp2 are materialized simultaneously.

ior : Break is taken when either abp1 or abp2 are materialized.

[iand|ior]: Specifies ibp1 and ibp2 conditions.

iand: Break is taken when ibp1 and ibp2 are materialized simultaneously.

ior : Break is taken when either ibp1 or ibp2 are materialized.

[Function]

Sets or clears an event condition as a break condition.

Ibp1 and ibp2 are execution event and abp1 and abp2 are access event.

For how to set ibpx and abpx, refer to the description of each command.

[Examples]

bpopt aor

Specifies abp1 or abp2 as a break condition.

bpop seq

Specifies abp1, abp2, ibp1 and ibp2 sequential conditions 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] [jtag{25|12|5|2|1|500|250|100}] [[!]verify]
[[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]
[[!]cresetb] [[!]resetb] [pclock1|pclock2|pclock4]
[io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]

[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. And for RTE-100-TP, the parameters other than jtag-25 or jtag12 is invalid.

[!]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.

[[!]int0] [[!]int1] [[!]int2] [[!]int3] [[!]int4] [[!]timer]:
Specifies whether the INTx pin is to be masked. Enter ! if it is not to be masked.

[[!]cresetb] [[!]resetb]:
Specifies whether the RESET pin is to be masked. Enter ! if it is not to be masked.
cresetb is ColdResetB pin. Resetb is ResetB pin.

[pclock1|pclock2|pclock4]: Please always use it by default pclock4.

[io_nouse|io_brkout|io_brkin|io_trgout|io_trgin]: Specifies the mode of BKTGIO_L pin.

io_nouse:	no use
io_brkout:	Break output
io_brkin:	Break in
io_trgout:	Please do not specify.
io_trgin:	Please do not specify.

[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:
 BKTGIO_L       = Present
 Cotrol Unit      = Present
CPU Settings:
 Auto Run        = ON (auto)
 JTAGCLOCK    = 12.5MHz (jtag12)
 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)
```

[Examples]

env nmi verify timer

Specifies masking of NMI and TIMER, ON of verify.

abp1, abp2 command

[Format]

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

[Parameters]

ADDR: Specifies an address in hexadecimal number.
AMASK: Specifies masking of ADDR. ADDR is masked with '1' in bit units.
data DATA [DMASK]|nodata: Specifies a data condition.
data DATA: Specifies data in hexadecimal number.
DMASK: Specifies masking of DATA. DATA is masked with '1' in bit units.
nodata: Deletes specification of data from the condition.
noasid | asid ASID:
noasid: Does not include ASID in subject to comparison.
asid ASID: Includes ASID in subject to comparison.
aeq|aneq: Specifies the condition of an address.
aeq is normal addr. aneq is negative addr
deq|dneq: Specifies the condition of an data.
aeq is normal data. aneq is negative data
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.
nosize|byte|hword|word|dword: Specifies an access size condition.
nosize: Does not include access size in subject to comparison.
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.

abp{1|2} /del: Each condition is deleted.

[Function]

Specifies an event of an access cycle.

[Examples]

abp1 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 combination conditions of abp1 and abp2 are specified by bpopt.

Ibp1, ibp2 command

[Format]

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

ibp{1|2} /del

[Parameters]

ADDR: Specifies an address in hexadecimal number.

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

asid ASID| noasid:

asid ASID: Includes ASID in subject to comparison.

noasid: Does not include ASID in subject to comparison.

ibp{1|2} /del: Each condition is deleted.

[Function]

Specifies an event for an executable address.

[Examples]

ibp1 1000 0

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

Ibp1 1000 0ff

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

Ibp2 1000 asid 10

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

[Remark]

The combination conditions of ibp1 and ibp2 are specified by bpopt.

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-VR4120-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-VR4120-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 break is 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-VR4120-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, two 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.

sfr command

[Format]

sfr [reg [VAL]]

[Parameters]

VAL: Specifies the value for an SFR register in hexadecimal notation.

reg: Specifies an SFR register name.

The following names can be used as register names:

Read/write registers:

BCUCNTREG1 ROMSIZEREG ROMSPEEDREG IO0SPEEDREG IO1SPEEDREG REVIDREG
CLKSPEEDREG BCUCNTREG3 CSIIBALREG CSIIBAHREG CSIIALREG CSIIAHREG CSIOBALREG
CSIOBAHREG CSIOALREG CSIOAHREG FIRBALREG FIRBAHREG FIRALREG FIRAHREG
DMARSTREG DMASENREG DMAMSKREG TDREG DMAABITREG CONTROLREG BASSCNTLREG
BASSCNTHREG CURRENTCNTLREG CURRENTCNTHREG TCINTR CMUCLKMSK MSYSINT1REG
MGIUINTLREG MDSIUINTREG NMIREG SOFTINTREG MSYSINT2REG MGIUINTHREG MFIRINTREG
MPCIINTREG MSCUINTREG MCSIINTREG MBCINTREG PMUINTREG PMUCNTREG PMUINT2REG
PMUCNT2REG PMUWAITREG PMUTCLKDIVREG PMUINTRCLKDIVREG ETIMELREG
ETIMEMREG ETIMEHREG ECMPLREG ECMPMREG ECMPHREG RTCL1LREG RTCL1HREG
RTCL2LREG RTCL2HREG TCLKLREG TCLKHREG RTCINTREG GIUIOSELL
GIUIOSELH GIUPIODL GIUPIODH GIUINTSTATL GIUINTSTATH GIUINTENL GIUINTENH
GIUINTTYPL GIUINTTYPH GIUINTALSELL GIUINTALSELH GIUINTHTSELL GIUINTHTSELH
GIUPODATEN GIUPODATL LEDHTSREG
LEDLTSREG LEDCNTREG LEDASTCREG LEDINTREG CSI_MODEREG
CSI_CLKSELREG CSI_SOTBREG CSI_SOTBFREG CSI_CNTREG CSI_INTREG CSI_IFIFOVREG
CSI_OFIFOVREG CSI_OFIFOREG CSI_FIFOTRGREG RAMBALREG
RAMBAHREG RAMALREG RAMAHREG IOBALREG IOBAHREG IOALREG IOAHREG SDRAMMODEREG
SDRAMCNTREG BCURFCNTREG BCURFCOUNTREG RAMSIZEREG SIUDLL
SIUIE SIUDLM SIULC SIUMC SIULS SIUMS SIUSC SIUIRSEL SIURESET SIUCSEL
DSIUDLL DSIUIE DSIUDLM DSIULC DSIUMC DSIULS DSIUMS DSIUSC FRSTR
DPINTR DPCNTR IMR FSR IRSR1 CRCSR FIRCR MIRCR
DMACR DMAER TXFL MRXF PCIMMAW1REG
PCIMMAW2REG PCITAW1REG PCITAW2REG PCIMIOAWREG PCICONFDREG PCICONFAREG
PCIMAILREG BUSERRADREG INTCNTSTAREG PCIEXACCREG PCIRECONTREG PCIENREG
PCICLKSELREG PCITRDYVREG PCICLKRUNREG
VENDORIDREG DEVICEIDREG COMMABDREG
STATUSREG REVREG CLASSREG CACHELSREG LATTIMEREG MAILBAREG PCIMBA1REG
PCIMBA2REG INTLINEREG INTPINREG
RETVALREG PCIAPCNTREG TIMOUTCNTREG
TIMOUTCOUNTREG ERRLADDRESSREG ERRHADDRESSREG SCUINTRREG

Write-only registers:

SIUTH SIUFC DSIUTH DSIUFC TDR

Read-only registers:

DMAIDLEREG DMAREQREG SYSINT1REG GIUINTLREG DSIUINTREG SYSINT2REG GIUINTHREG
FIRINTREG PCIINTREG SCUINTREG CSIINTREG BCUINTREG RTCL1CNTLREG
RTCL1CNTHREG RTCL2CNTLREG RTCL2CNTHREG TCLKCNTLREG TCLKCNTHREG CSI_SIRBREG
CSI_SIRBEREG CSI_SIOREG CSI_IFIFOREG SIURB
SIUIID DSIURB DSIUIID RDR TXIR
RXIR IFR RXSTS RXFL

[Function]

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

[Examples]

sfr PIC0

The value of the PIC0 register is displayed.

sfr PIC0 2

The value 2h is set in the PIC0 register.

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.

ver command

[Format]

ver

[Parameters]

None

[Function]

The ver command displays the version of KIT-VR4120-TP.