APPENDIX A. KIT-VR4131-TP INTERNAL COMMANDS

This appendix describes the KIT-VR4131-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 provides equivalent functions, a contention may occur between KIT-VR4131-TP and the debugger, causing either device to malfunction.

Command syntax

The basic syntax for the KIT-VR4131-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. (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 hold simultaneously.

aor : Break is taken when either abp1 or abp2 holds.

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

iand: Break is taken when ibp1 and ibp2 hold simultaneously.

ior : Break is taken when either ibp1 or ibp2 holds.

[Function]

This command sets an event condition as a break condition.

ibp1 and ibp2 are execution events and abp1 and abp2 are access events.

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.

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

[Function]

These commands are used to manipulate the cache.

[Examples]

cacheflush 80000000 1000

flush cache addr=80000000 len=00001000

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

env and ememstat commands

[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]

ememstat

[Parameters]

[!]auto: If a break point is set 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.

jtag{25|12|5|2|1|500|250|100}: Specifies the JTAG clock for N-Wire. Each number corresponds to the following JTAG clock.
[25 MHz|12.5 MHz|5 MHz|2 MHz|1 MHz|500 kHz|250 kHz|100 kHz]

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

[!]verify: Specifies whether 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 external interrupt 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 the ColdResetB pin. resetb is the 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 the BKTGIO_L pin.

io_nouse:	No use
io_brkout:	Break output
io_brkin:	Break input
io_trgout:	Please do not specify.
io_trgin:	Please do not specify.

[Function]

The env command sets the emulation environment and displays the DCU status. 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 ememstat command displays the mounting status of the E.MEM board when RTE-2000-TP is used.

Display examples are shown below (status of default value).

With RTE-1000-TP

```
Probe:
 Unit          : RTE-1000-TP      << Displays the main unit 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)
```

With RTE-2000-TP

```
Probe:
 Unit          : RTE-2000-TP                    << Displays the main unit connected.
 Rom Probe     : (use ememstat command)
 Emem Size     : (use ememstat command)
CPU:
 BKTGIO_L      = Present
 Cotrol Unit   = Present
CPU Settings:
 Auto Run      = ON (auto)
 JTAGCLOCK     = 25MHz (jtag25)
 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 1/2 (pclock2)
 BKTGIO_L Mode= not use (io_nouse)

>ememstat
 Board_num     EMEM_Size     ROM_Probe
 ===================================
   ROM1        8Mbyte        Extend Type 2K    << Displays the status of the module mounted.
   ROM2        32Mbyte       Extend Type 2K    << Displays the status of the module mounted.
```

[Examples]

env nmi verify timer

Specifies masking of NMI and TIMER, and enables verify.

abp1 and abp2 commands

[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.
AMASK:	Specifies masking of ADDR. ADDR is masked with '1' in bit units.
data DATA [DMASK]:	Specifies a data condition.
data DATA:	Specifies data in hexadecimal.
DMASK:	Specifies masking of DATA. DATA is masked with '1' in bit units.
asid ASID|noasid:	Specifies asid.
asid ASID:	Includes ASID in subject to comparison.
noasid:	Does not include 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. deq is normal data. dneq is negative data.
read|write|accs:	Specifies a status condition.
read:	Specifies a read cycle as a status condition.
write:	Specifies a write cycle as a status condition.
accs:	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]

These commands specify events for access cycle breaks.

[Examples]

abp1 1000 0 data 5555 0 hword read

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

[Remark]

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

ibp1 and ibp2 commands

[Format]

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

ibp{1|2} /del

[Parameters]

ADDR: Specifies an address in hexadecimal.

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

asid ASID| noasid: Specifies ASID.

asid ASID: Includes ASID in subject to comparison.

noasid: Does not include ASID in subject to comparison.

aeq|aneq: Specifies the condition of an address. aeq is normal addr. aneq is negative addr.

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

[Function]

These commands specify an event for an executable address.

[Examples]

ibp1 1000 0

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

ibp2 1000 0ff

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

ibp2 1000 0 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 require 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: Specifies the address of an input port in hexadecimal.

[Function]

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

[Examples]

inb b0000000
The I/O space is read in bytes (8-bit units), starting at b0000000H.
inh b0000000
The I/O space is read in half words (16-bit units), starting at b0000000H.
inw b0000000
The I/O space is read in words (32-bit units), starting at b0000000H.
ind b0000000
The 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-VR4131-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.

LENGTH: Specifies the number of bytes to be read, in hexadecimal. (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-VR4131-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  b8000000    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 of a software break prohibition area in bytes.
The minimum unit of a specification area is the boundary of word.
The number of the areas which can be specified is a maximum of four.

[Function]

The nsbp command specifies an area to forbid a software break.
When a break point is specified, a debugger implicitly performs a memory test (write access) to an object address.
The state of some flash ROM may change by performing write access and right data may not be read. When this happens, please forbid a software break by this command to prohibit use of write cycles. 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|/all]

[Parameters]

block-number: Specifies the block number 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 minimum unit of the a specification area is as follows.

RTE-1000-TP: 4 bytes

RTE-2000-TP: Depends on the size of the ROM being emulated.

8/16 bits: 128K bytes

32 bits: 256K bytes

(64 bits: 512K bytes)

The number of areas which can be specified is a maximum of four.

[Function]

The nrom command specifies the area when part of ROM emulation area specified by ROM command is mapped to other resources on a user system. Usually, it is not necessary to specify this command. The operations for the specified area are as follows.

- An access from the debugger is forcibly made to the user system.
- The EMEMEN- signal is deasserted inactive (high level) in the cycle for accessing this area during execution (RTE-2000-TP only).

[Examples]

nrom a0000000 20000

A 20000-byte area, starting at a0000000h, is specified as a forced user area.

```
>nrom a0000000 20000
No.   Address     Length
  1   a0000000    00020000

>nrom a0100000 40000
No.   Address     Length
  1   a0000000    00020000
  2   a0100000    00040000
```

nromd command

[Format]

nromd [block-number|/all]

[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 specified by nrom.

[Examples]

nromd 1

Block 1 is deleted from the forced user area.

```
>nrom a000000 40000
No.   Address     Length
  1   a0000000    00020000
  2   a1000000    00040000

>nromd 1
No.   Address     Length
  1   a1000000    00040000
```

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.
DATA: Specifies the data to be output in hexadecimal.

[Function]

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

[Examples]

outb b8000000 12
Byte data 12h is written to b8000000h in the I/O space.
outh b8000000 1234
Half word data 1234h is written to b8000000h in the I/O space.
outw b8000000 12345678
Word data 12345678h is written to b8000000h in the I/O space.
outd b8000000 123456789abcdef0
Long word data 123456789abcdef0h is written to b8000000h in the I/O space.

reset command

[Format]

reset

[Parameters]

None

[Function]

The reset command resets the emulation CPU of KIT-VR4131-TP.

rom command (for RTE-1000-TP)

[Format]

rom [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [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:	Specifies the 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 bit.
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 cable is used, choose rom8; if a DIP-40/42-ROM or STD-16BIT-ROM cable 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 rom data. During a download, when little is specified, the binary image of the file is downloaded as is. When big is specified, the data is downloaded with the high-order and low-order bytes exchanged according to the bus size of ROM.

[Function]

The rom command sets the ROM emulation environment of RTE-1000-TP. ADDR and LENGTH must be input in pairs. Input other parameters only when their values 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 bfc00000 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 ROMs are emulated because the bus is 32 bits wide. The endian of ROM is little. (The binary image is loaded as is.)

rom bfc00000 40000 2m 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.)

<Remark>

Note on area specified by rom command

Access to a range specified by the rom command from the debugger is a direct access to the emulation memory in the tool. As a result, display is performed correctly even if the processor cannot correctly access ROM. It is therefore recommended to read and check data by using the jread command (that reads data via the CPU bus) or write data by setting verify to ON with the env command (download) in the initial stage of debugging.

rom1..rom4 commands (for RTE-2000-TP)

[Format]

rom1 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16] [bus8|bus16|bus32|bus64] [[!]wren]

rom2 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16] [bus8|bus16] [[!]wren]

rom3 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16] [bus8|bus16|bus32] [[!]wren]

rom4 [ADDR [LENGTH]] [512k|1m|2m|4m|8m|16m|32m|64m|128m|256m] [rom8|rom16] [bus8|bus16] [[!]wren]

rom1: This command performs setting of a module including the EMEM board mounted to slot #3.
rom2: This command performs setting of a module including the EMEM board mounted to slot #4.
rom3: This command performs setting of a module including the EMEM board mounted to slot #5.
rom4: This command performs setting of a module including the EMEM board mounted to slot #6.

[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: Specifies the 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 bit.

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 cable is used, choose rom8; if a DIP-40/42-ROM or STD-16BIT-ROM cable is used, choose rom16.

bus8|bus16|bus32|bus64: Specifies the ROM bus size in the system to be emulated. 8 bits, 16 bits, 32 bits, or 64 bits can be specified.
>> [bus64] is a parameter for future use. (It is not used with KIT-VR4131-TP.)

[[!]wren]: This setting is for using the emulation memory as RAM. wren enables writing, and !wren disables writing. The default value is !wren.

[Function]

The rom1 to rom4 commands set the ROM emulation environment of RTE-2000-TP. ADDR and LENGTH must be input in pairs. Input other parameters only when their values 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]

rom1 bfc00000 40000 2m rom16 bus16 !wren

Slot position of EMEM board	Address range	Bus width	ROM		Write enable
			Bus width	Bits	
#3	bfc00000 - bfc3ffff	16 bits	16 bits	2M bits	Disabled

rom2 bfc40000 40000 2m rom16 bus16 wren

Slot position of EMEM board	Address range	Bus width	ROM		Write enable
			Bus width	Bits	
#4	bfc40000 - bfc7ffff	16 bits	16 bits	2M bits	Enabled

rom1 bfc00000 80000 2m rom16 bus32 !wren

Slot position of EMEM board	Address range	Bus width	ROM		Write enable
			Bus width	Bits	
#3 + #4	bfc00000 - bfc7ffff	32 bits	16 bits	2M bits	Disabled

Do not issue the rom2 command at this time.

<Remark>

Note on area specified by rom command

Access to the range specified by the rom1..rom4 commands from the debugger is a direct access to the emulation memory in the tool. As a result, display is performed correctly even if the processor cannot correctly access ROM. It is therefore recommended to read and check data by using the jread command (that reads data via the CPU bus) or write data by setting verify to ON with the env command (download) in the initial stage of debugging.

Relationship between rom command and EMEM board

rom command	Bus width	Slot position of EMEM board	Unusable rom command
rom1	8 bits	#3	
	16 bits	#3	
	32 bits	#3 + #4	rom2
	64 bits	#3 + #4 + #5 + #6	rom2, rom3, rom4
rom2	8 bits	#4	
	16 bits	#4	
rom3	8 bits	#5	
	16 bits	#5	
	32 bits	#5 + #6	rom4
rom4	8 bits	#6	
	16 bits	#6	

tlb32 and tlb64 commands

[Format]

tlb32 [all | INDEX [MASK HI Lo0 Lo1]]

tlb64 [all | INDEX [MASK HI Lo0 Lo1]]

[Parameters]

all: Specifies display of all indexes.

INDEX: Specifies a specific index.

MASK HI Lo0 Lo1:

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.

Lo0: Specifies EntryLo0.

Lo1: Specifies EntryLo1.

[Function]

These commands display and change the contents of TLB.

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

tlb64 displays 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 BCUCNTREG3
CSIIBALREG CSIIBAHREG CSIIALREG CSIOBALREG CSIOBAHREG CSIOALREG FIRBALREG
FIRBAHREG FIRALREG RAMBALREG
RAMBAHREG RAMALREG RAMAHREG IOBALREG IOBAHREG IOALREG IOAHREG DMARSTREG
DMASENREG DMAMSKREG DMAREQREG TDREG DMAABITREG CONTROLREG BASSCNTLREG
BASSCNTHREG CURRENTCNTLREG CURRENTCNTHREG TCINTR CMUCLKMSK MSYSINT1REG
MGIUINTLREG MDSIUINTREG NMIREG SOFTINTREG MSYSINT2REG MGIUINTHREG MFIRINTREG
MPCIINTREG MSCUINTREG MCSIINTREG MBCUINTREG 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 TIMOUTCNTREG
TIMOUTCOUNTREG ERRLADDRESSREG ERRHADDRESSREG SCUINTRREG SDRAMMODEREG
SDRAMCNTREG BCURFCNTREG BCURFCOUNTREG RAMSIZEREG PCIMMAW1REG
PCIMMAW2REG PCITAW1REG PCITAW2REG PCIMIOAWREG PCICONFDREG PCICONFAREG
PCIMAILREG INTCNTSTAREG PCIEXACCREG PCIENREG PCICLKSELREG
PCITRDYVREG PCICLKRUNREG COMMANDREG STATUSREG
CHACHELSREG LATTIMEREG MAILBAREG PCIMBA1REG PCIMBA2REG INTLINEREG INTPINREG
RETVALREG PCIAPCNTRETG DSIUDLL DSIUIE DSIUDLM DSIUFC DSIULC DSIUMC
DSIULS DSIUMS DSIUSC SIURESET LEDHTSREG LEDLTSREG LEDCNTREG LEDASTCREG
LEDINTREG SIUDLL SIUIE SIUDLM
SIUFC SIULC SIUMC SIULS SIUMS SIUSC SIUIRSEL SIURESET SIUCSEL CSI_MODEREG
CSI_CLKSELREG CSI_SOTBREG CSI_SOTBFREG CSI_CNTREG CSI_INTREG CSI_IFIFOVREG
CSI_OFIFOVREG CSI_OFIFOREG CSI_FIFOTRGREG FRSTR
DPINTR DPCNTR IMR FSR IRSR1 CRCSR FIRCR MIRCR
DMACR DMAER TXFL MRXF

Write-only registers:

DSIUTH SIUTH TDR

Read-only registers:

REVIDREG CLKSPEEDREG CSIIAHREG CSIOAHREG FIRAHREG DMAIDLEREG SYSINT1REG
GIUINTLREG DSIUINTREG SYSINT2REG
GIUINTHREG FIRINTREG PCIINTREG SCUINTREG CSIINTREG BCUINTREG RTCL1CNTLREG
RTCL1CNTHREG RTCL2CNTLREG RTCL2CNTHREG TCLKCNTLREG TCLKCNTHREG
BUSERRADREG
PCIRECONTREG VENDORIDREG DEVICEIDREG
REVIDREG CLASSREG DSIURB DSIUIID SIURB SIUIID CSI_SIRBREG CSI_SIRBEREG
CSI_SIOREG CSI_IFIFOREG RDR TXIR
RXIR IFR RXSTS RXFL

[Function]

The sfr command sets and displays the value of the 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]

FILENAME: Specifies file name.

NAME: 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-VR4131-TP.