APPENDIX A. KIT-V831/2-TP INTERNAL COMMANDS

This appendix describes the KIT-V831/2-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.

Example: If PARTNER/Win is used,

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

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

Command syntax

The basic syntax for the KIT-V831/2-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.)

abp, abp1, abp2, abp3, and abp4 commands

[Format]

abp[1|2|3|4] [ADDR] [io|mem]
abp [ADDR] [io|mem] /del
abp{1|2|3|4} /del
abpd{1|2|3|4}

[Parameters]

abp: Specifies abp condition.

ADDR:

ADDR specifies an address in hexadecimal notation.

io|mem:

io: Specifies i/o space access as a condition.

mem: Specifies memory space access as a condition.

/del:

Deletes the condition.

Abpd: Deletes the condition by channel.

[Function]

These commands set or delete access break points.
Up to four access break points can be set.
The abp command automatically uses an unused channel.
To specify a channel explicitly, use abp1, abp2, abp3, or abp4, as appropriate.

[Examples]

abp 1000 mem

A break point is set for memory access to address 1000h.

abp2 2000 io

A break point is set for i/o access to address 1000h.

abp1 /del

The condition set by abp1 is deleted.

Abpd1

The condition set by abp1 is deleted.

cmcr, dctr, and itcr commands

cmcr command

[Format]

cmcr [=]value

[Function]

The cmcr command sets a value in the CMCR (cache memory control register).

dctr command

[Format]

dctr [all]

[Function]

The dctr command displays DCTR registers.

There are 256 DCTR registers. This command displays only those registers whose valid bits are effective.

If all is entered, all the registers are displayed.

The DCTR registers are mapped to f2000000h to f2000fffh in the I/O space.

itcr command

[Format]

itcr [all]

[Function]

The itrc command displays the ICTR registers.

There are 128 ICTR registers. This command displays only those registers whose valid bits are effective.

If all is entered, all the registers are displayed.

The ICTR registers are mapped to fa000000h to fa000fffh of the I/O space.

env command

[Format]

env [[!]auto] [[!]reset] [[!]nmi] [[!]hldrq] [[!]int{00|01|02|03}] [[!]int{10|11|12|13}]
[jtag{25|12}] [!]verify [inone|istack|iaddr ADDR]

[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.

[!]reset

This parameter specifies whether the RESET pin is to be masked. Enter ! if it is not to be masked.

[!]nmi

This parameter specifies whether the NMI pin is to be masked. Enter ! if it is not to be masked.

[!]hldrq

This parameter specifies whether the HLDRQ pin is to be masked. Enter ! if it is not to be masked.

[!]int{00|01|02|03}

This parameter specifies that pins INT00 to INT03 are to be masked. Enter ! if they are not to be masked.

[!]int{10|11|12|13}

This parameter specifies that pins int10 to int13 are to be masked. Enter ! if they are not to be masked.

jtag[12|25]

This parameter specifies the JTAG clock for N-Wire. Usually, use jtag25.

[!]verify

This parameter specifies the verification after writing memory is set. Enter ! if it is not to be set.

[inone|istack|iaddr ADDR]

This parameter specifies the work ram area when debugger accesses the V832 internal instruction ram.

inone	: not specified
Istack	: This parameter specifies the area from current 32-byte stack.
Iaddr ADDR	: This parameter specifies the area from ADDR to ADDR + 32-byte.

[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       = 25 MHz (jtag25)
Signals Mask:
  INT00..13       = NO MASK (!int00..13)
  NMI             = NO MASK (!nmi)
  RESET           = NO MASK (!reset)
  HLDRQ           = NO MASK (!hldrq)
```

[Example]

env reset !nmi

RESET is masked while NMI is not.

help command

[Format]

```
help  [command]
```

[Parameter]

command: Specify 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.

[Example]

```
help map
```

A help message for the map command is displayed.

inb, inh, and inw commands

[Format]

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

[Parameter]

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

[Function]

The inb, inh, and inw commands read I/O space.

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

If the address is omitted, the address specified previously with the command is assumed.

[Examples]

inb 1000

I/O space is read in bytes (8-bit units), starting at 1000H.

inh 1000

I/O space is read in half words (16-bit units), starting at 1000H.

inw 1000

I/O space is read in words (32-bit units), starting at 1000H.

init command

[Format]

init

[Parameters]

None

[Function]

The init command initializes RTE-100-TP. All environment values are initialized. A memory cache rejection area is not initialized.

jread command

[Format]

jread [ADDR [LENGTH]]

[Parameters]

ADDR: This parameter specifies an address in hexadecimal notation.

LENGTH: This parameter 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.

[Example]

```
jread ffff0000 100
```

100h bytes, starting at ffff0000, are read via JTAG.

nc command

[Format]

```
nc  [[ADDR [LENGTH]]
```

[Parameters]

ADDR: The start address of a memory cache rejection area is specified.

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

[Function]

To ensure quick memory access, KIT-V831/2-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.

[Example]

```
nc 10000 1000
```

A 1000-byte area, starting at 10000, is specified as a memory cache rejection area.

```
>nc 10000 1000
  No Memory Cache Area
  No. Address  Length
  1  010000    001000
  2  fff000    001000
```

ncd command

[Format]

ncd block-number

[Parameter]

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

[Function]

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

[Example]

ncd 2

Block 2 is deleted from the memory cache rejection area.

```
>nc
  No Memory Cache Area
  No. Address  Length
  1  020000   000100
  2  010000   001000

>ncd 2
  No Memory Cache Area
  No. Address  Length
  1  020000   000100
```

outb, outh, and outw commands

[Format]

```
outb [[ADDR] DATA]
outh [[ADDR] DATA]
outw [[ADDR] DATA]
```

[Parameters]

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

DATA: This parameter specifies the data to be output in hexadecimal notation.

[Function]

The outb, outh, and outw commands writes data to the I/O space.

The outb command accesses I/O space in bytes, outh in half words, and outw in words.

If the address and data are omitted, those previously specified with the command are assumed.

[Examples]

```
outb 1000 12
```

Byte data 12h is written to 1000H in the I/O space.

```
outh 1000 1234
```

Half word data 1234h is written to 1000H in the I/O space.

```
outh 1000 12345678
```

Word data 12345678h is written to 1000H in the I/O space.

reset command

[Format]

reset

[Parameters]

None

[Function]

The reset command resets the emulation CPU of RTE-100-TP.

rom command

[Format]

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

[Parameters]

ADDRESS [LENGTH]

ADDR: Lowest address of the ROM to be emulated. (An error occurs if it does not match a ROM boundary.)

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

Specify 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 bit.

rom8|rom16

Specify 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

Specify the ROM bus size of the system to be emulated. The system supports the specification of 8 bits, 16 bits, or 32 bits.

[Function]

The rom command sets the ROM 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, the last entry of the parameter is valid. The initial value of LENGTH is 0 (not used).

[Example]

rom C0000 40000 1m rom16 bus16

The 256K bytes (40000) of the 27C1024 (16-bit ROM with a size of 1M bit), starting at 0xc0000, are emulated. Consequently, two 16-bit ROMs are emulated.

rom fff80000 80000 2m rom rom16 bus32

The 512K bytes (80000) of the 27C2048 (16-bit ROM with a size of 2M bits), starting at 0xfff80000, are emulated. Consequently, two 16-bit ROMs are emulated.

<Note>

Access to the range specified by the rom command results in direct access to internal emulation memory. Access to addresses outside the range is performed via the processor bus.

sfr command

[Format]

sfr [reg [VAL]]

[Parameters]

VAL: The value for an SFR register is specified in hexadecimal notation.

reg: An SFR register name is specified.
The following names can be used as register names:

<V831>

Read/write registers:

IGP BCTC DBC DRC PRC ASIM00 ASIM01 CSIM0 SIO0
BRG0 BPRM0 TMC1 TMC4 TOC1 TOVS PORT PM PC CGC
IMR IMOD PWC0 PWC1 PIC RFC DSA0H DSA1H DSA2H DSA3H
DSA0L DSA1L DSA2L DSA3L DDA0H DDA1H DDA2H DDA3H
DDA0L DDA1L DDA2L DDA3L DBC0H DBC1H DBC2H DBC3H
DBC0L DBC1L DBC2L DBC3L DCHC0 DCHC1 DCHC2 DCHC3 DC
CM4 CC10 CC11 CC12 CC13 TUM1

Write-only registers:

TXS0L ICR TXS0

Read-only registers:

ASIS0 RXB0L IRR RXB0 TM1 TM4

<V832>

Read/write registers:

PORT PM PC
BCTC DBC PWC0 PWC1 RFC PRC
DSA0H DSA0L DDA0H DDA0L DBC0H DBC0L DCHC0
DSA1H DSA1L DDA1H DDA1L DBC1H DBC1L DCHC1
DSA2H DSA2L DDA2H DDA2L DBC2H DBC2L DCHC2
DSA3H DSA3L DDA3H DDA3L DBC3H DBC3L DCHC3 DC
TOVS TUM1 TMC1 TOC1 CC10 CC11 CC12 CC13 TMC4 CM4
ASIM00 ASIM01 CSIM0 SIO0 BRG0 BPRM0
IGP IMR IMOD
CGC PMR
PORTA PAM PAC PORTB PBM PBC
PIC0 PIC1 SDC

Write-only registers:

TXS0 TXS0L ICR SDM

Read-only registers:

TM1 TM4 ASIS0 RXB0 RXB0L IRR

[Function]

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

[Examples]

sfr IGP

The value of the IGP register is displayed.

sfr CM4 2

The value 2h is set in the CM4 register.

symfile and sym commands

[Format]

symfile FILENAME	>> Reads an elf file (.elf).
sym [NAME]	>> Displays symbols (up to 30).

[Parameters]

symfile:	File name
sym:	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.

tp command

[Format]

tp [ADDR]

[Parameter]

ADDR: This parameter specifies an even-numbered address in hexadecimal notation. (A0 is always corrected to 0.)

[Function]

The tp command specifies a trace trigger point.

[Example]

tp ffff0000

The execution of the instruction at ffff0000h is specified as a trigger point.

[Note]

If delay mode is specified with the tron command, the trigger point specification is ignored. Delay mode can be canceled by entering tron !delay.

tsp command

[Format]

```
tsp [ADDR] [/del]
```

[Parameters]

ADDR:	This parameter specifies an address in hexadecimal notation.
/del:	This parameter cancels the specified address.

[Function]

The tsp command specifies a trace start point (address).

[Example]

```
tsp ffff0000
```

The execution of the instruction at ffff0000h is specified as a trace start address.

[Note]

Trace information may overflow when it is output from the CPU. By specifying a trace start point so that trace is started immediately before the event to be traced, trace information can be prevented from overflowing.

If a start address is not specified, trace starts forcibly at the point the TRON command is issued.

The start address specified with this command is effective when TRON is issued.

td1 and td2 commands

[Format]

```
td1 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]
td2 [DADDR [ignore|ioread|iowrite|ioacc|memread|memwrite|memacc] [/del]
```

[Parameters]

DADDR:

This parameter specifies an address in hexadecimal notation. The address is corrected such that it corresponds to a 4-byte boundary.

ignore|ioread|iowrite|ioacc|memread|memwrite|memacc:

Specify the desired status.

ignore:	don't care
ioread:	Read from I/O space
iowrite:	Write to I/O space
ioacc:	Read and write to and from I/O space
memread:	Read from memory space
memwrite:	Write from memory space
memacc:	Read and write to and from memory space

/del:

This parameter cancels the setting.

[Function]

The td1 and td2 commands set the data cycles to be recorded by trace.

[Example]

```
td1 fe000000 memread
```

Reads from address fe000000h in memory are traced.

tron command

[Format]

tron [DELAY] [[!]delay] [noreal|real] [noignore|ev{[0]|..|[8]}] [noext|nega|posi]

[Parameters]

DELAY = 0..1ffff delay counter

This parameter specifies the number of frames in memory that are to be recorded in response to a trigger, in decimal notation.

[!]delay: This parameter specifies forced delay mode. Enter !delay to return to normal mode. In forced delay mode, trace is forcibly terminated when the number of frames specified by the delay counter are recorded after trace starts. In this mode, trigger events are ignored.

[noreal|real]:

This parameter specifies trace mode.

real: Real-time mode

normal: Non-real-time mode. (Currently not available; if it is chosen, a forcible break occurs immediately before overflow, and reexecution is not performed.)

noignore|ev{[0]|..|[8]}

This parameter specifies the events that are not to be recorded by trace.

noignore: All events are to be recorded. Normal specification.

ev0..8:

ev0: Exception occurrence information is not to be recorded.

ev1: Interrupt occurrence information is not to be recorded.

ev2: Condition Jump occurrence information is not to be recorded.

ev3: PC relative occurrence information is not to be recorded.

ev4: JAL occurrence information is not to be recorded.

ev5: RETI to occurrence information is not to be recorded.

ev6: RETI from occurrence information is not to be recorded.

ev7: Jump register indirect to occurrence information is not to be recorded.

ev8: Jump register indirect from occurrence information is not to be recorded.

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 recording trace data.

[Examples]

Trace is unconditionally performed for 1ffff cycles in delay mode.

```
>tron delay 1ffff
Trace Settings:
 Start Address = Force
 Delay Count   = 0001fffd
 Trace Mode    = Real Time (real)
 Delay Mode    = Enable (delay)
 Ignore Event  = None (noignore)
 Ext Trigger   = Disable (noext)
 Data Trace 1  = Disable (ignore)
 Data Trace 2  = Disable (ignore)
 Trig Address  = Disable
```

Data trace is performed on IOREAD from address 100h, with the execution of the instruction at address fe00000h as a trigger. ffffh is set as the delay counter (DELAY).

```
>tp fe000000                              <<Trigger specification
 Trig Address  = fe000000

rte3>td1 100 ioread                       <<Data trace specification
 Data Trace 1  = 00000100 I/O Read (ioread)
 Data Trace 2  = Disable (ignore)

>tron ffff                                <<Start of trace
Trace Settings:
 Start Address = Force
 Delay Count   = 0000ffff
 Trace Mode    = Real Time (real)
 Delay Mode    = Disable (!delay)
 Ignore Event  = None (noignore)
 Ext Trigger   = Disable (noext)
 Data Trace 1  = 00004444 I/O Read (ioread)
 Data Trace 2  = Disable (ignore)
```

trcoff command

[Format]

troff

[Parameters]

None

[Function]

The trcoff command forcibly terminates the recording of trace data.

trace command

[Format]

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

[Parameters]

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

[all|pc|data] The cycle in recorded trace information that is to be displayed is specified.

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 recorded, 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

```
  Cycle   Sub  Address  Code     Instruction                                  EXT  Stat
 -000018 0000 fffffff0 20bcff4f movhi  4fffh,r0,r1                            0000 FTRC
 -000018 0001 fffffff4 21a00000 movea  0000h,r1,r1                            0000 ----
 -00000e 0000 fffffff8 0118     jmp    [r1]                                   0000 JREG
* -00000a 0000 4fff0000 1070     ldsr   r0,DPC                                0000 JREG
  000002 0000 4fff0018 00a8f247 jr     4fff480ah                              0000 JMPR
  000002 0001 4fff480a a0bf074e movhi  4e07h,r0,r29                           0000 ----
  000002 0002 4fff480e bda30002 movea  0200h,r29,r29                          0000 ----
  000002 0003 4fff4812 bddf7400 st.w   r29,0074h[r29]                         0000 ----
  000002 0004 4fff4816 1ddc7c00 st.w   r0,007ch[r29]                          0000 ----
  000006 0000 4fff481a ffaf80b8 jal    4fff009ah                              0000 JAL
  000006 0001 4fff009a 1ddc0000 st.w   r0,0000h[r29]                          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.

RD#1: Occurrence of a read cycle of data trace (dt1)

TRIG: Occurrence of a trigger address

FAIL: Occurrence of a failure to record trace data

JMPR: Occurrence of the branch origination address of the PC relative branch instruction
JAL: Occurrence of a branch origination address due to the JAL instruction
RETI1: Occurrence of a branch origination address due to the RTEI instruction
JREG1: Occurrence of a branch origination address due to the register indirect branch instruction
FTRC: Start of trace
WR#1: Occurrence of a write cycle of data trace (dt1)
RETI2: Occurrence of a branch address due to the RETI instruction
JREG2: Occurrence of a branch address due to the register indirect branch instruction
INTR: Occurrence of branching due to a maskable interrupt
EXP: Occurrence of branching due to NMI or Exceptions
CJMP: Occurrence of a branch address due to the conditional branch instruction
RD#2: Occurrence of a read cycle of data trace (dt2)
WR#2: Occurrence of a write cycle of data trace (dt2)

ver command

[Format]

```
ver
```

[Parameters]

None

[Function]

The ver command displays the version of KIT-V831/2-TP.