MIPS® Navigator™ Register Edit Guide

Document Number: MD00863
Revision 04.31
June 23, 2011

MIPS Technologies, Inc.
955 East Arques Avenue
Sunnyvale, CA 94085-4521

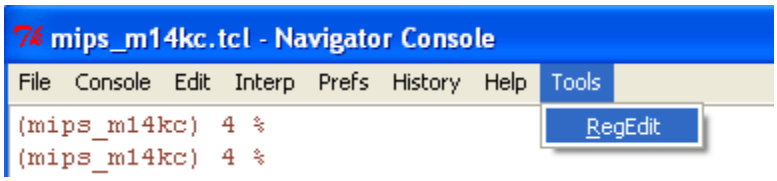
Table of Contents

1. Introduction

RegEdit allows the user to view and change the contents of MIPS core registers and memory mapped locations. Register names, index or address, fields, default values, and symbolic field names are specified in XML in a Register Definition file. Use the example Register Definition files provided to display standard register sets or create your own to display memory mapped peripheral registers for custom devices.

2. Running RegEdit

Before running RegEdit you must first open Navigator Console and connect to the target. You can then start up RegEdit from the "Tools" menu.



RegEdit can also be run from a command shell.

Windows users "cd" to the folder that contains regedit.bat (C:\mips\NavigatorConsole\tools\regedit\win) and enter the regedit.bat command followed by parameters "C:\mips\NavigatorConsole", your probe serial number, and a device number. For example:

```
regedit.bat C:\mips\NavigatorConsole 42278 0
```

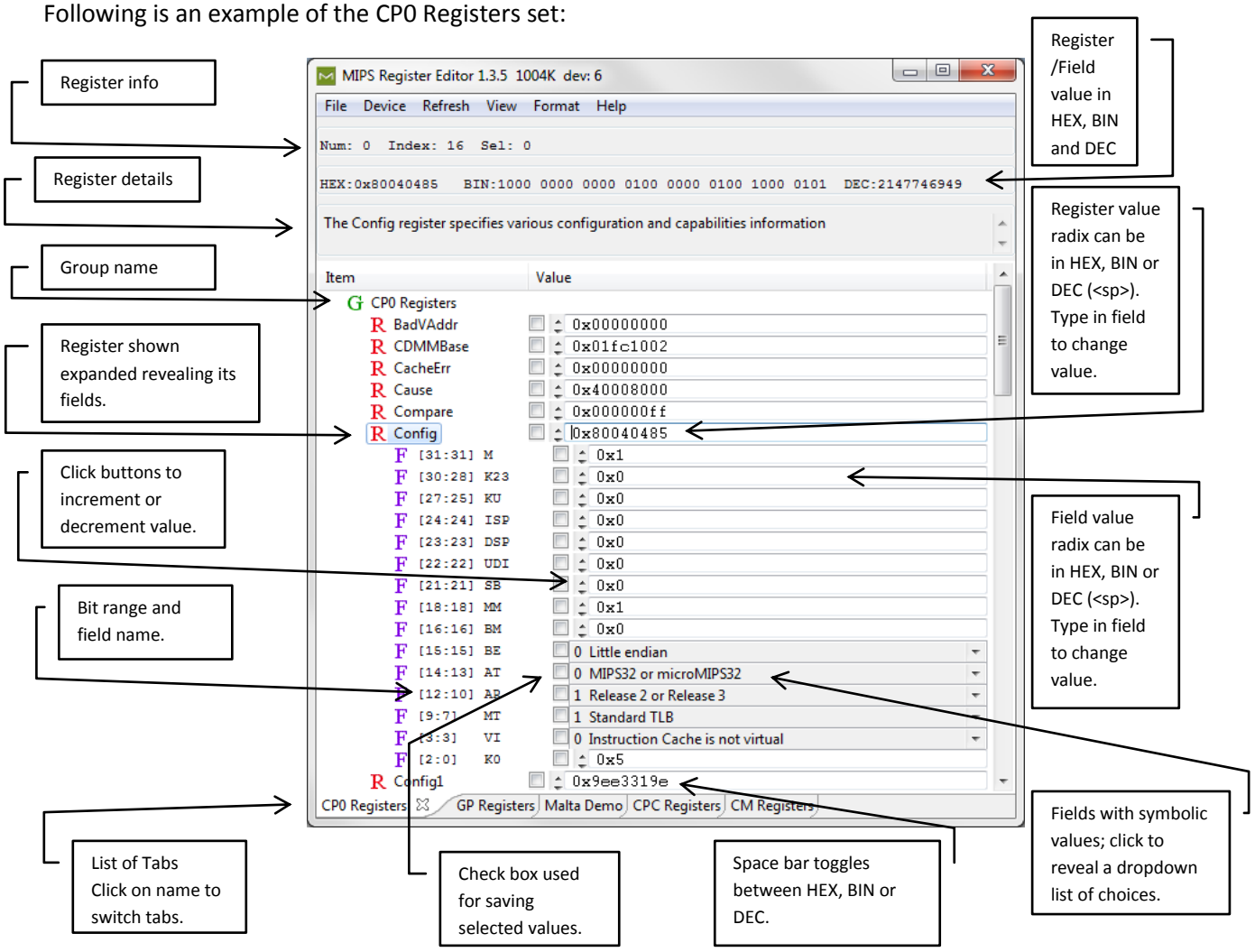
Linux users would "cd" to the directory that contains regedit.sh (/…/mips/NavigatorConsole/tools/regedit/linux) and enter the regedit.sh command followed by parameters "/mips/NavigatorConsole", your probe serial number, and a device number. For example:

```
regedit.sh /mips/NavigatorConsole 42278 0
```

When the RegEdit tool is first opened, it should be populated with a register definition file, master.xml. If that file is not opened automatically, you may open it using the File Menu item, Load Definition File…

3. RegEdit View

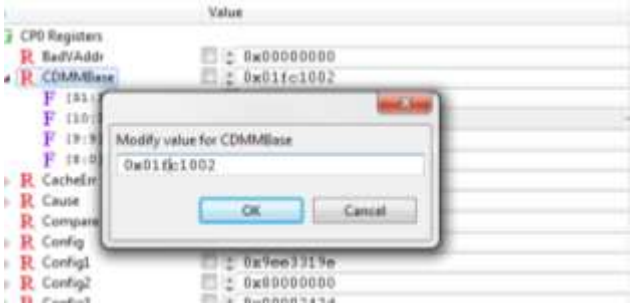
Following is an example of the CP0 Registers set:



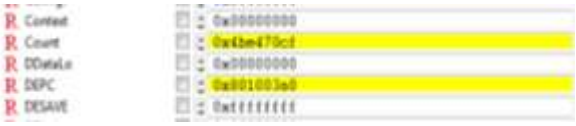
4. Register / Field Values

Each register or field is displayed on a single line with the name on the left and the value on the right.
Clicking the mouse on the register or field name or value brings up its description in the upper part of the window. If the field is writeable typing a digit will immediately write the new value into the register or memory, replacing the old value. You can also increment the digit to the right of the cursor with the '+' key and decrement with the '-' key. Hitting the "r" key will cause the value to be re-read. See other shortcut keys below.

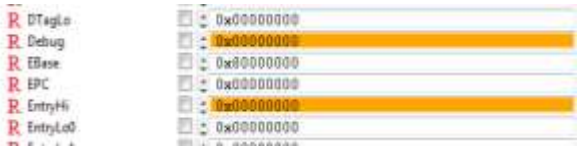
The "m" key will open a dialog, allowing you to modify the entire value before writing it.



After a refresh, the background color is yellow if that particular register has changed since the previous refresh.

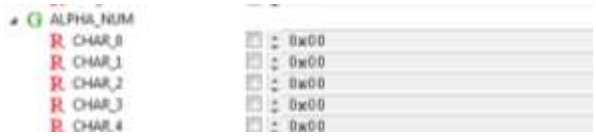


After a refresh, the background color is orange if that particular register could not be read while the processor is running. The value displayed is the old value.



After a refresh, the background color is red if an error occurred reading that particular register. The value displayed will be zero.

A background color of gray indicates that the particular register is write only. The value displayed is the last value written.



5. File Menu

- Load Definition File…
  Opens a dialog that allows the user to pick a Register Definition .xml file to load. Any existing tabs with names matching tabs in the new file will be closed before the file is loaded.

- Save All Values from Current Tab...
  Opens a dialog that allows the user to specify a filename. File will contain an XML representation of all values in the current tab.

- Save All Values from all Tabs...
  Opens a dialog that allows the user to specify a filename. File will contain an XML representation of all values from all tabs.

- Save Checked Values from Current Tab...
  Opens a dialog that allows the user to specify a filename. File will contain an XML representation of all values that have a check in the save checkbox in the current tab.

- Save Checked Values from all Tabs...
  Opens a dialog that allows the user to specify a filename. File will contain an XML representation of all values that have a check in the save checkbox in all tabs.

- Load Saved Values...
  Opens a dialog that allows the user to specify a .xml filename of a file containing values to load. Values specified in the file that are currently defined will be loaded.

- Load Default Values
  Loads “default” values that are specified in the Register Definition File for each register.

- Close All Tabs
  Closes all open tabs and clears file history.

- Exit
  Exits the program.

6. Device Menu

- Select Device
  Select from the menu which device to connect to. The Device number is the same as the JTAG index.

- Select TC
  Select from the menu which TC to connect to. The target Device must be halted to select a TC.

7. Refresh Menu

- Refresh Current Tab from Target
  Reread all values in the current tab.

- Refresh All Tabs from Target
  Reread all values in all tabs.

- Refresh on Halt
  When checked, automatically reread all values in all tabs when the processor halts.

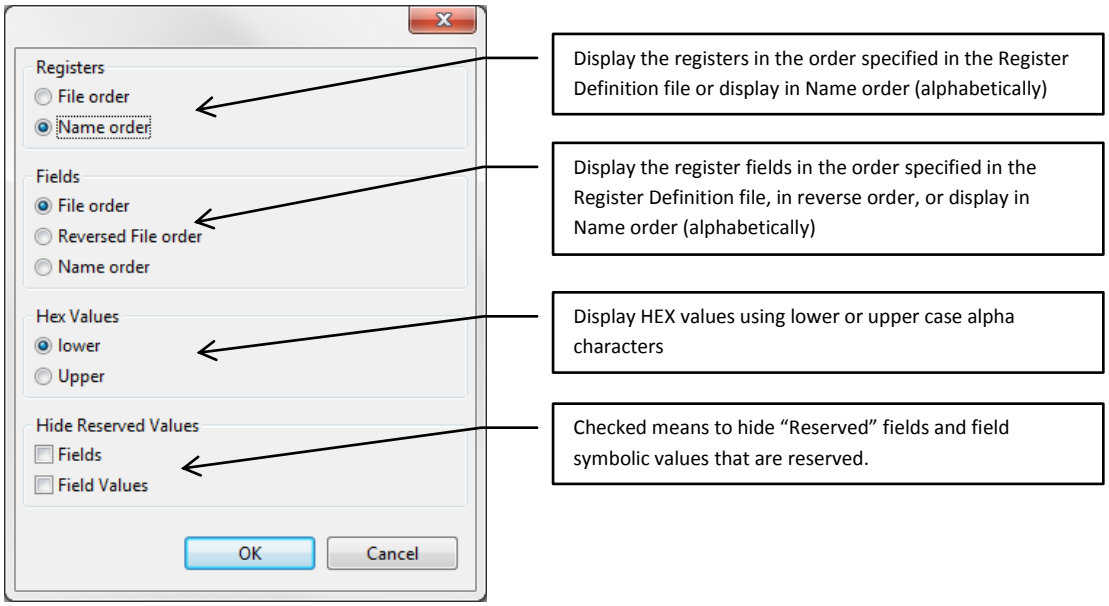
8. View Menu

- Show Info Box
  Enable or disable the display of the Info Box in the top portion of the window
- Show Values Box
  Enable or disable the display of the Values Box in the top portion of the window
- Show Details Box
  Enable or disable the display of the Details Box in the top portion of the window

9. Format Menu

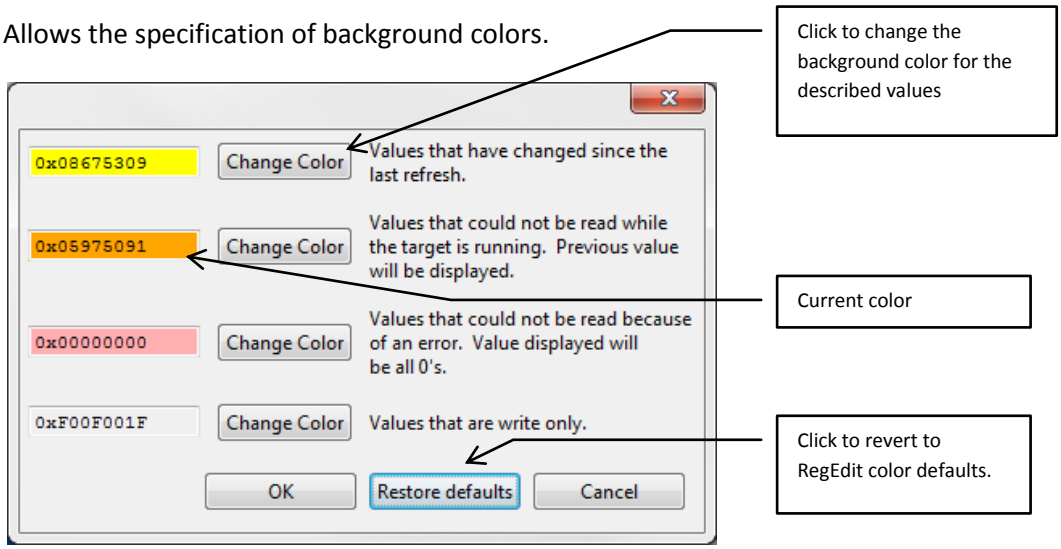
- Display Options…

Allows the specification of how the data is displayed.



- Color Options…

Allows the specification of background colors.



10. Help Menu

- Keyboard Shortcuts… The following keyboard shortcuts are available

Shortcut	**Action**
tab, shift+tab	Move to next/previous register
arrow keys	Navigate
r	Refresh current value
ctrl+r	Refresh current tab
shift+r	Refresh all tabs
space	Change Radix
shift+up, + or =	Increment the digit to the right of the cursor
shift+down, - or _	Decrement the digit to the right of the cursor
!	Flip all bits
z	Zero current value
<	Shift value left, zero fill
>	Shift value right, zero fill
m	Open Modify Value Dialog

- RegEdit Guide

You're reading it now.

- File History

Lists the Register Definition .xml files loaded

11. RegEdit Definition File Description

The definition file is used to describe the registers and memory locations to display. A definition file consists of a root element *tabs*. A *tabs* element contains one or more *tab* elements. A *tab* element contains one or more *group* elements. A *group* element contains one or more register elements, either *cop_register*, *grp_register*, *or mm_register*. A register element contains zero or more *field* elements. A *field* element contains zero or more *field_value* elements.

Following is a list of these elements and their attributes.

Element <tabs>

The outer most root element. Contains namespace references and optionally specifies an XSD validation file. A *tabs* element contains one or more *tab* elements.

Element <tab>

A *tab* element contains definitions for a *tab*. Each *tab* contains one or more group elements.

Attribute	Required	Description
core	no	Used to specify which cores will enable this tab
name	Yes	Name display in Tab
path	no	Path to definition file

```
<tab name="CP0 Registers">
```

Element <group>

A *group* element contains one or more register definitions.

Attribute	Required	Description
name	yes	The name of this group
tooltip	no	Tooltip to display when mouse hovers over group name
desc	no	Description to display when register is selected
base	no	Base address for memory mapped registers
core	no	Specifies which cores are applicable for this group

```
<group name="LED_BAR" tooltip="Malta LED BAR Group" desc="Led Bar Group Desc" base="0xBF000000">
```

Element <cop_register>

A co-processor register definition contains attributes which describe the co-processor register and possible *field* elements that describe fields making up the register. These register values will be color coded orange when they cannot be read. Note: some CP0 registers on processors that contain multiple TCs per VPE cannot be read unless the processor is halted.

Attribute	Required	Description
cp_number	yes	The co-processor number
cp_index	yes	The cop index
cp_select	yes	The cop select
name	yes	The name of this register field
rwflag	yes	Designates if register is read, write or both. Values: “r”, “w”, or “rw”
core	no	Used to specify which cores have this field
default	no	Value that is loaded when doing a Load Default Values
tooltip	no	Tooltip to display when mouse hovers over register name
desc	no	Description to display when register is selected
radix	no	Radix to display value: values are 2, 10 or 16. Defaults to 16 if not present.

```
    <cop_register cp_number="0" cp_index="1" cp_select="1" name="VPEControl" rwflag="rw" tooltip="VPE control
    and status" desc="Per-VPE register containing relatively volatile thread configuration data">
  <field name="YSI" bits="21:21" desc="1 for exception on all 'yield' instructions"/>
  <field name="GSI" bits="20:20" desc="1 for exception on all gating storage accesses"/>
  <field name="EXCPT" bits="18:16" desc="Cause of last thread exception ">
    <field_value name="Thread Underflow" value="0"/>
    <field_value name="Thread Overflow" value="1"/>
    <field_value name="Invalid YIELD Qualifier" value="2"/>
    <field_value name="Gating Storage Exception" value="3"/>
    <field_value name="YIELD Scheduler Exception" value="4"/>
    <field_value name="GS Scheduler Exception" value="5"/>
    <field_value name="Reserved" value="6" reserved="true"/>
    <field_value name="Reserved" value="7" reserved="true"/>
  </field>
  <field name="TE" bits="15:15" desc="When 0 only one TC in this VPE may run"/>
  <field name="TargTC" bits="7:0" desc="Remote TC# which will be accessed by mttr/mftr instructions"/>
</cop_register>
```

Element <gpr_register>

A general purpose register definition contains attributes which describe the general purpose register. These register values will be color coded orange when they cannot be read. Note: for processors that contain multiple TCs per VPE the processor must be halted before general purpose registers can be read. These register values will be color coded orange when they cannot be read.

Attribute	Required	Description
name	yes	The name of this register field
rwflag	yes	Designates if register is read, write or both. Values: “r”, “w”, or “rw”
default	no	Value that is loaded when doing a Load Default Values
tooltip	no	Tooltip to display when mouse hovers over register name
desc	no	Description to display when register is selected
radix	no	Radix to display value: values are 2, 10 or 16. Defaults to 16 if not present.

```
<gpr_register name="s0      (r18)" rwflag="rw" default="0" tooltip="Subroutine register variable" desc="Subroutine
register variable; a subroutine that writes one of these must save the old value and restore it before it exits, so the
calling routine sees the values preserved">
</gpr_register>
```

Element <mm_register>

A memory mapped register definition contains attributes which describe the memory mapped register and possible *field* elements that describe fields of the register. Note: only memory mapped registers with an access size of 4 (32-bits) can be read while the processor is running.

Attribute	Required	Description
address	yes	Memory address; this value is added to the base value specified in the *group* element.
accessize	yes	Access size: 4=32 bit, 2=16 bit, 1=8 bit
datasize	yes	Data size: 4=32 bit, 2=16 bit, 1=8 bit
name	yes	The name of this register
rwflag	yes	Designates if register is read, write or both. Values: “r”, “w”, or “rw”
default	no	Value that is loaded when doing a Load Default Values
tooltip	no	Tooltip to display when mouse hovers over register name
desc	no	Description to display when register is selected
radix	no	Radix to display value: values are 2, 10 or 16. Defaults to 16 if not present.

```
<mm_register address="0x450" accesssize="4" datasize="1" name="CHAR_7" rwflag="w" default="42"
        tooltip="char 7" desc="Char 7">
</mm_register>
```

Element <field>

Register field definition breaks the register into fields. Each field specifies a starting and ending bit. Field definitions can contain *field_value* elements to define symbolic names for the field value.

Attribute	Required	Description
name	yes	The name of this register field
bits	yes	Defines the bits composing this field. Format: “*MSstartBit:LSendBit*” e.g. “31:29” would specify bits 31, 30 and 29
core	no	Specifies which cores have this field.
default	no	Value that is loaded when doing a Load Default Values
desc	no	Description to display when register is selected
tooltip	no	Tooltip to display when mouse hovers over register name
radix	no	Radix to display value: values are 2, 10 or 16. Defaults to 16 if not present.
reserved	no	Set to: reserved=”true” if this field is a reserved field

```
<field name="TDS" bits="21:21" desc="1 when halted on an instruction in branch delay slot"/>
```

```
<field name="DT" bits="20:20" desc="TC is 'dirty' -- has run code, may have changed state"/>
<field name="TCEE" bits="17:17" desc="Per-TC view of Status[CEE] (CorExtend) enable"/>
<field name="DA" bits="15:15" desc="TC available for 'fork'"/>
<field name="A" bits="13:13" desc="TC allocated -- set by fork, TC can't run unless set"/>
<field name="TKSU" bits="12:11" desc="Per-TC view of Status[UM,SM] (privilege level)"/>
<field name="IXMT" bits="10:10" desc="TC is Interrupt-eXeMpt"/>
<field name="TASID" bits="7:0" desc="Per-TC view of EntryHi[ASID]"/>

<field name="Reserved" bits="30:6" reserved="true" />

<field name="Event" bits="11:5" core="34K" desc="Which event should we count?">
```

Element <field_value>

Values and description for register field values

Attribute	Required	Description
name	yes	The name or text that describes this value
reserved	no	Set to: reserved=”true” if this field is a reserved field
value	yes	The value

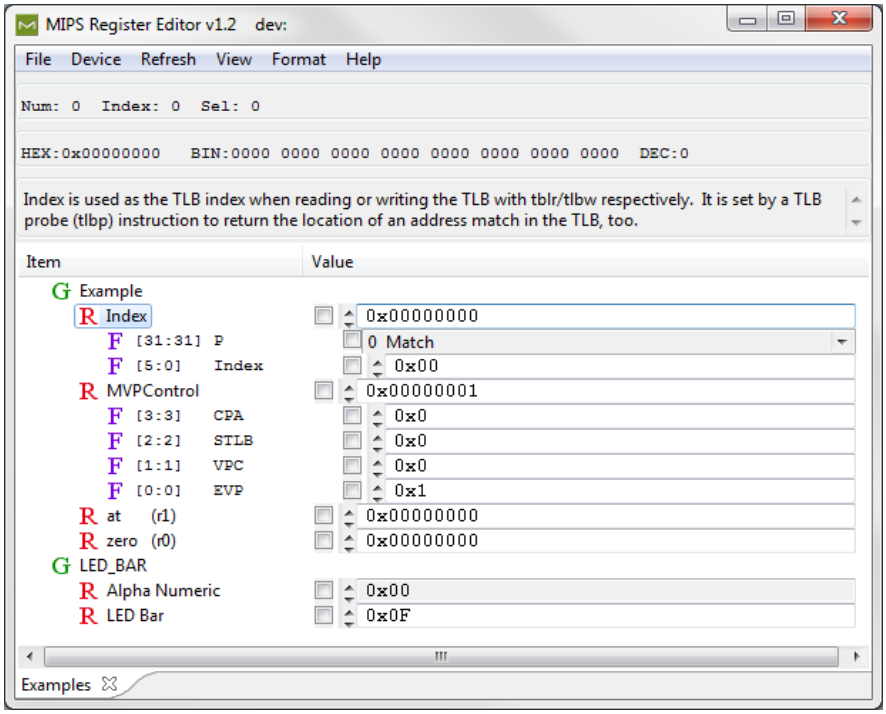
```
<field_value name="Running" value="0"/>
<field_value name="Blocked on WAIT" value="1"/>
<field_value name="Blocked on YIELD" value="2"/>
<field_value name="Blocked on Gating Storage" value="3"/>
```

12. Example Definition File

Following is a file with two cp0 registers, two general purpose registers and two memory mapped registers defined.

```
<tabs>
    <tab name="Examples">
        <group name="Example" tooltip="">
            <cop_register cp_number="0" cp_index="0" cp_select="0" name="Index" rwflag="rw"
                    tooltip="Index into the TLB array" desc="Index is used as the TLB index when
                    reading or writing the TLB with tblr/tlbw respectively.  It is set by a TLB probe (tlbp)
                    instruction to return the location of an address match in the TLB, too.">
                <field name="P" bits="31:31" desc="Probe Failure. Set to 1 by hardware when the
                        previous TLBProbe (TLBP) instruction failed to find a match in the TLB">
                    <field_value name="Match" value="0"/>
                    <field_value name="No match" value="1"/>
                </field>
                <field name="Index" bits="5:0" desc="Index to the TLB entry affected by the
                    TLBRead and TLBWrite instructions"/>
            </cop_register>
            <cop_register cp_number="0" cp_index="0" cp_select="1" name="MVPControl" rwflag="rw"
                    tooltip="CPU-wide multithreading features control" desc="The MVPControl register is
                    instantiated per-processor, and provides an interface for global control and configuration
                    of a multi-VPE MIPS MT 34k">
                <field name="CPA" bits="3:3" desc="1 to permit per-VPE assignment of cache ways"/>
                <field name="STLB" bits="2:2" desc="1 to enable TLB sharing between VPEs"/>
                <field name="VPC" bits="1:1" desc="1 for MT configuration mode"/>
```

```
            <field name="EVP" bits="0:0" desc="The dvpe/evpe bit: 0 disables multithreading, even other VPE"/>
        </cop_register>
        <gpr_register name="zero   (r0)" rwflag="rw" default="0" tooltip="Always Returns Zero"
                desc="Always Returns Zero">
        </gpr_register>
        <gpr_register name="at      (r1)" rwflag="rw" default="0" tooltip="Assembly Temporary"
                desc="Assembly Temporary, Reserved for use by assembly">
        </gpr_register>
    </group>
    <group name="LED_BAR" desc="Malta board registers" base="0xBF000000">
            <mm_register address="0x408" accesssize="4" datasize="1" name="LED Bar" rwflag="rw" />
            <mm_register address="0x410" accesssize="4" datasize="1" name="Alpha Numeric" rwflag="w" />
    </group>
  </tab>
</tabs>
```



13. References

1. MIPS® Navigator™ Console Getting Started Guide
MIPS Document: MD00732

2. MIPS® Navigator™ ICS Getting Started Guide
MIPS Document: MD00708

3. GDB Guide for MIPS® Probes
MIPS Document: MD00734

4. Hardware Guide for MIPS® Probes
MIPS Document: MD00733

5. Console Guide for MIPS® Probes
MIPS Document: MD00705

6. MIPS® Navigator™ Register Edit Guide
MIPS Document: MD00833