Errata for MIPS R10000 Microprocessor User's Manual, Version 1.0 (June' 95)

This errata supplements the "MIPS R10000 User's Manual, Version 1.0 released June' 95" by Joe Heinrich

January 5, 1996

MIPS Technologies Inc.
2011 N Shoreline Blvd
PO Box 7311
Mountain View, CA 94039-7311

Page 4:

Last paragraph, 2nd line:
Change:

"....about 6.8 million transistors..."

To:

"....about 6.7 million transistors...".

Page 19:

2nd. para, last line:
Change:

"...... ranging from 4Kbytes to 16Mbytes, inclusive, in increments of 4..."

To:

"...... ranging from 4Kbytes to 16Mbytes, inclusive, in power of 4....."

Page 34:

For VrefByp, change:

"This pin must be tied to VrefSys or Vss." to "This pin must be tied to VrefSys or Vss, preferably, through at least a 100ohm resistor.

Page 35:

In the description for the signal name: SCAAddr(18:0), change

"Duplicated complementary 19-bit bus..." to "Duplicated 19-bit bus..."

Page 38:

In the description for the signal name "PLLDis" change:

"This signal must be tied to Vss." to "This signal must be tied to Vss through a 100 ohm resistor."

Page 54:

Third paragraph:
Change

"...in increments..."

To:

"...in power of 2..."

Page 58:

Third paragraph, 3rd bullet should read as follows:

- If the access misses in both ways of the secondary cache, the data is fetched from the system interface. If the state of the predicted way is found to be invalid, the fetched data is placed in it and the MRU is unchanged. However, if the state of the predicted way is found to be valid then the fetched data is placed into the non-predicted way, and the way prediction table is updated to point to this way since it is now the most-recently-used.

Page 81:

Replace the third paragraph with:

"An outgoing buffer entry containing a coherency data response is ready for issue to the System interface bus when the quadword specified by the corresponding external intervention request is received from the secondary cache. The processor then allows the data to stream from the secondary cache to the System interface bus through the outgoing buffer."

Page 89:

Replace the second sentence of the second paragraph with:

"During the address cycle of processor block read, data typical block write, upgrade, and eliminate requests, the processor asserts **SysCmd[0]**."

Page 102:

"Processor Block Read Request Protocol" section,

1)add the following to the first paragraph:

Before issuing a processor block read request, the processor changes the secondary cache state to Invalid. Additionally, if the secondary cache block former state was DirtyExclusive, a write back is scheduled. Note that if the processor block read request receives an external NACK or ERR completion response, the secondary cache block state remains Invalid.

2) Third paragraph, first sentence should read as:

When multiple, non-conflicting processor requests .....

Page 106:

1) Delete the seventh bullet, "asserting **SysCmd[0]**"

2) Replace the forth paragraph with:

"If the processor block write request results from the writeback of a secondary cache block, the *Dirty Exclusive* secondary cache block former state is driven on **SysAD[2:1],** the secondary cache block way is driven on **SysAD[57]** and **SysCmd[0]** is asserted."

3) Replace the fifth paragraph with:

"If the processor block write request results from a completely gathered uncached accelerated block, the uncached attribute is driven on **SysAD[59:58]** and **SysCmd[0]** is negated."

Page 121:

Add the following at the end of the first paragraph:

The external agent refers to any device attached to the processor system interface. It may be memory interface or cluster coordinator ASIC, or another processor residing on the cluster bus.

Page 126:

At the end of the 2nd para in section *Processor Coherency State Response Protocol*, add the following sentence:

"The processor coherency state responses are issued in an order designated by the external coherency requests and will always be issued before an associated processor coherency data response. Note that processor coherency state responses can be pipelined ahead of the associated processor coherency data responses, and processor coherency data responses can be returned out-of-order. These cases typically arise from external coherency requests hitting outgoing buffer entries."

Page 127:

Replace the last para. with the following paragraphs:

"When **SysStateVal*** is negated, **SysState[0]** provides the processor coherency data response indication. The processor asserts the processor coherency data response indication when there is one or more processor coherency data responses pending issue in the outgoing buffer. Once asserted, the indication is negated when the first double-word of the last pending issue processor coherency data response is issued to the system interface bus. The processor coherency data response indication is not affected by **SysWrRdy***. However, as previously noted the processor may only issue a processor coherency data response when **SysWrRdy*** was asserted two **SysClk** cycles previously.

"Processor coherency data response data is supplied in subblock order, beginning with the quadword-aligned address specified by the corresponding external coherency request. Processor coherency data responses are not necessarily issued in the same order as the external coherency requests; however each processor coherency data response always follows the corresponding processor coherency state response. Note that more than one processor coherency state response may be pipelined ahead of the corresponding processor coherency data responses."

Page 132:

1) Add the following to the end of the first paragraph:

"If the external agent chooses to issue an external coherency request to the processor which causes an external coherency conflict, the external coherency request must be completed before an external response is given to the conflicting processor request."

2) In table 6-29, replace the footnote * with the following:

"Although it is not required, the external agent may choose to issue the conflicting external coherency request to R10000 and the processor will return an *invalid* processor coherency state response."

3) Change the second reference to the footnote, in the "Resolution" column, "*" to "#" and add the following description for the "#" footnote:

"Although it is not required, the external agent may choose to issue the conflicting external coherency request to R10000 and the processor will return an *shared* processor coherency state response."

Page:144

2nd para. of section, *Secondary Cache Clock*
Change:

"The processor provides 6 pairs of differential HSTL clock outputs,...... synchronous SRAMs."

To:

"The processor provides 6 pairs of differential clock outputs,...... synchronous SRAMs. These outputs swing between **VccQSC** and **Vss**."

Page 148:

At the end of section 8.2, add the following paragraph:

If the **virtual SysClk** is used during the reset sequence, the mode bits, **SysGnt***, **SysRespVal***, and **SysReset*** should all be referenced to the **virtual SysClk** that is created with **SysCyc***. This approach will cause the R10000 to come out of reset synchronously with the **virtual SysClk**, which will allow repeatable and lock-step operation (see Section 6.24 for description of **virtual SysClk** operation).

Page 155

In the "Name and Function" column: replace SysClk(5:0) and SysClk*(5:0) to SCClk(5:0) and SCClk*(5:0), respectively.

Page 158:

1) Second to last paragraph, second sentence: Replace "protection" with "detection".
2) Replace the last paragraph with:

"To inform the external agent, the processor asserts **SysUncErr*** for one **SysClk** cycle whenever any of the following uncorrectable errors are detected:

- Primary instruction cache tag array parity errors
- Primary data cache tag array parity errors
- Secondary cache tag array uncorrectable ECC errors
- System interface command bus parity errors
- System interface address/data bus external address cycle uncorrectable ECC errors
- System interface response bus parity errors"

Page 168:

Add the following sentence to the second to last paragraph:

"Additionally, the processor informs the external agent by asserting **SysUncErr*** for one **SysClk** cycle."

Page 169:

Add the following sentence to the fifth paragraph:

"Additionally, the processor informs the external agent by asserting **SysUncErr*** for one **SysClk** cycle."

Page 170:

Add the following sentence to the second to last paragraph:

"Additionally, the processor informs the external agent by asserting **SysUncErr*** for one **SysClk** cycle."

Page 189:

At the end of the page add this paragraph:

"The JTAG operation does not require **DCOk** to be asserted or **SysClk** to be running; however, if **DCOk** is asserted the **SysClk** must run at the specified minimum frequency or the core logic may get damaged."

Page 198:

At the end of the section "Unused Inputs" add the following:

Following input pins may be unused in certain system configurations, and each of them should be tied to VccQSys, preferably, through a resistor of 100 ohm or greater value:

- SysNMI*

Following input pins may be unused in certain system configurations, and each of them should be tied to Vss, preferably, through a resistor of 100 ohm or greater value:

- SysRdRdy*
- SysWrRdy*
- SysGblPerf*
- SysCyc*

Following input pins may be unused in certain system configurations, and each of them should be tied to VccQSys or, preferably, Vss but always through a resistor of 100 ohm or greater value:

- SysADChk(7:0)

Page 203:

In section *De-coupling Capacitance*
Change:

- six VccQSC-Vss
- two VccQSys-Vss

To:

- five VccQSC-Vss
- four VccQSys-Vss

Page 224:

For figures 14-4, 14-19, 14-20, 14-21, 14-22, 14-24, 14-25, 14-26, 14-27, 14-28, 14-29, 14-30, 14-31 add the description of field "0" as follows:

"Reserved. Must be written as zeros, and returns zeros when read."

Page 248:

As a footnote to the *Instruction issued* in the *Event 1* row add the following:

Note *"Issued"* instruction also includes those speculatively issued, but not graduated. *NOP, J(umps), BEQ 0 0*, and all *FP branch* instructions, are not issued to any functional unit; thus, they are included in the *"Issued"* count. Moreover, *Integer MULT* and *DIVIDE* are each counted as two instructions because they write to two destination registers, and take two slots in the *Active List*.

Page 256 & 257:

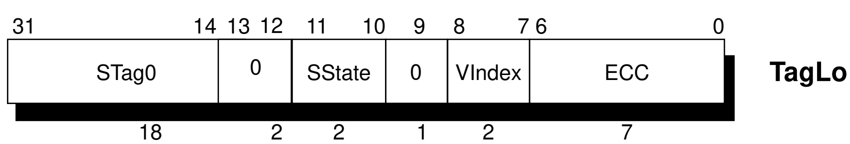
Definition of LRU should read as:

LRU: indicates which way is the least recently used in the set.

Page 258:

1) Figure 14-30:

STag0 field should be bits 31..14 rather than 31..13. The 0 field should be bits 13..12 rather than just 12.



2) In the Secondary Cache Operation section MRU should read as follows:

MRU: indicates which way was the most recently used in the set.

Page 271:

1) At the end of the 1st paragraph add the following:

"The *event specifier registers* are referred as *control registers* in the description of *CP0 register 25*."

2) MFPS and MTPS for the Performance Event Specifier:

Replace the content of bit 0 from "1" to "0".

Page 304:

In the last sentence of the section *Cache Pages*,
Change:

"...., in increments of 4 (that is 4Kbytes, 16Kbytes,....)."

To:

"...., in power of 4 (that is 4Kbytes, 16Kbytes,....)."

Change the title:

Cache Pages

To:

Virtual Pages

Change the title:

Cache Page Size Encodings

To:

Virtual Page Size Encodings