



## ABOUT ADSP-TS101S SILICON ANOMALIES

These anomalies represent the currently known differences between revisions of the TigerSHARC ADSP-TS101S product and the functionality specified in the ADSP-TS101S data sheets and the Hardware Reference books.

### SILICON REVISIONS

A silicon revision number with the form "-x.x" is branded on all parts (see the data sheet for information on reading part branding). The silicon revision can also be electronically read by reading the IDCODE register either via JTAG or DSP code.

The following DSP code can be used to read the register:

```
<UREG> = IDCODE;;
```

Silicon REVISION	IDCODE[31:28]
0.4	0101
0.2	0011

### ANOMALY LIST REVISION HISTORY

The following revision history lists the anomaly list revisions and major changes for each anomaly list revision.

Date	Anomaly List Revision	Applicable Data Sheet Revision	Additions and Changes
06/06/2007	L	B	Added anomalies: 03000355, 03000374
08/18/2005	K	B	Added anomalies: 03000343

#### NR002475L

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

## SUMMARY OF SILICON ANOMALIES

The following table provides a summary of ADSP-TS101S anomalies and the applicable silicon revision(s) for each anomaly.

No.	ID	Description	0.2	0.4
1	03000010	Internal Bus Requests by IALU	x	x
2	03000011	Floating Point Denormal Result	x	x
3	03000044	Wait Cycle on Other Master Read	x	x
4	03000099	JTAG emulation will fail during idle if interrupts are disabled	x	x
5	03000100	IALU operand fetch uses bypassed result of canceled instruction	x	x
6	03000102	TCB CX value is incorrect after block completion	x	x
7	03000105	BTB miss on target of non-quad aligned jump	x	x
8	03000107	DMA Priority - Cluster Bus Channels	x	x
9	03000108	Flyby Data Alignment	x	x
10	03000109	AutoDMA Overrun	x	x
11	03000110	PRC to RAS programming	x	x
12	03000111	Performance monitor count timing	x	x
13	03000114	Link does not detect error if checksum byte is not received	x	x
14	03000115	Fairness in Link to Link Communication is not always maintained	x	x
15	03000117	Wrong Link Transmission Time Out Error	x	x
16	03000120	The Link port does not stop on error	x	x
17	03000121	Address bus contention when mastership is changed	x	x
18	03000129	Stall in the DESPREAD Instruction	x	x
19	03000133	RETI is wrong when two interrupts arrive during IDLE	x	x
20	03000134	Lock of the BTB not functional	x	x
21	03000136	RETI return address corrupted during interrupt nesting	x	x
22	03000137	Conditional predicted RTI or RDS instructions are always partially executed	x	x
23	03000138	Link Transmit with PSIZE=0 fails if NACK'd in the middle	x	x
24	03000139	Interrupts missed when writing to ILATST/ILATCL	x	x
25	03000140	AutoDMA channels priority bit is always set	x	x
26	03000141	Exception is ignored when regprot or performance monitor counter wrap occur with no valid instruction	x	x
27	03000142	Stuck IFIFO when AUTODMA is paused	x	x
28	03000146	LDQM and HDQM pins have pull-downs instead of pull-ups	x	x
29	03000148	L0DIR and L1DIR always drive	x	x
30	03000149	Predicted unaligned branches may fail	x	x
31	03000150	LCLKRAT pins internal pull-downs are insufficient	x	x
32	03000151	Illegal op-codes in the pipeline	x	x
33	03000152	MAC saturation	x	x
34	03000153	Status register missing stall	x	x
35	03000156	AC flag is set when Rs = -Rm	x	x
36	03000157	SDRAM Performance in core accesses	x	x
37	03000158	DMA chaining stops	x	x
38	03000162	Branch from external memory may fail.	x	.
39	03000163	FDEP out of range	x	x
40	03000165	MAC underflow exception followed by another MAC	x	x
41	03000166	No bypass when reading from LSTAT.TST right after writing to LBUFTX	x	x
42	03000167	Chaining DMAs may not work with DMAR external requests	x	x

No.	ID	Description	0.2	0.4
43	03000168	First instruction of an ISR has to be quad-aligned	x	x
44	03000169	First three instruction lines in the ISR can not contain IALU instructions	x	.
45	03000172	RTI, RETI, CJMP and CJMP_CALL must be NP	x	.
46	03000173	RDS command should be performed only at interrupt level	x	x
47	03000174	SDRAM Performance	x	x
48	03000175	SBSRAM access in an MP system must use an ACK controlled buffer	x	x
49	03000176	AC set by instruction $R_s = R_m + R_n$ , $R_a = R_m - R_n$ ; but should be cleared	x	x
50	03000177	Internal Bus Arbiter Performance Issue #1	x	.
51	03000178	BUSLOCK cannot be toggled too quickly	x	x
52	03000179	SDRAM in high memory space should not use A15 as bank select	x	x
53	03000180	Time between exit of SDRAM self refresh to active is too short	x	x
54	03000181	SDRAM precharge with $t_{RP}=5$ takes four instead of five cycles	x	x
55	03000182	BUSLOCK with HBR can cause external bus deadlock	x	x
56	03000232	LCTL3:0 accessible as quads only through the external bus	x	x
57	03000233	Internal Bus Arbiter Performance Issue #2	x	x
58	03000234	Internal Bus Arbiter Performance Issue #3	x	x
59	03000235	One Dimensional DMA chaining fails if DY field is non-zero	x	x
60	03000236	External Port DMA chaining may fail when paused	x	x
61	03000237	loading cjmp from SDRAM misses the interlock	x	x
62	03000268	Global hardware interrupt disable allows interrupts for two cycles	x	x
63	03000288	RETS/DBGE gets wrong value when an exception/emulation arrives together with a reti update	x	x
64	03000302	Reading from SDRAM when SDREN bit in SDRCON is clear causes deadlock	x	x
65	03000318	Cancelled access to uninitialized SDRAM still causes "SDRAM access error"	x	x
66	03000322	Aborted LxSTATC or LxBUFRX read still updates LxSTAT	x	x
67	03000334	MAC instruction of one format followed by compact in different format might cause wrong saturation value	x	x
68	03000335	Incorrect AV flag in 'Rs=compact Rsd +/- Rsd' command	x	x
69	03000336	Incorrect AV flag in 'Rs(d)=ABS (Rs(d) +/- Rs(d))(U(X))' command	x	x
70	03000340	IA illegal address exception is not canceled when the command is aborted	x	x
71	03000343	DLL may lock incorrectly	x	x
72	03000355	BTB invalidate may not invalidate all entries	x	x
73	03000374	Conditional J-IALU compute stall	x	x

Key: x = anomaly exists in revision

. = Not applicable

## DETAILED LIST OF SILICON ANOMALIES

The following list details all known silicon anomalies for the ADSP-TS101S including a description, workaround, and identification of applicable silicon revisions.

### 1. 03000010 - Internal Bus Requests by IALU:

---

#### DESCRIPTION:

If the IALU stalls, and in the instruction pipeline stage I there is an IALU instruction that requests the internal bus, the IALU will request the bus although it will not execute the transaction because of the stall. This may happen if the stalled instruction is one of the following (every reference to J - also applies to K):

```
UREG = immediate;;  
UREG = UREG;;  
Js = Jm +/- Jn (cjmp);;  
[Jm +/-= Jn] = Ureg;;  
Ureg = [Jm +/-= Jn];;  
[Jm +/-= <immediate>] = Ureg;;  
Ureg = [Jm +/-= <immediate>];;
```

This may have a significant performance impact in applications in which the bus is used massively by both IALU (the instructions above) and DMA.

#### WORKAROUND:

Example for the scenario that causes this problem:

```
xr0 = r5 + r6;;  
xr8 = r3 * r0; r6 = [j0 + 4];;
```

Here the second instruction stalls for one cycle because of dependency on r0. The IALU, however, requests the bus even though the entire line is stalled.

Solution - Avoid placing IALU bus accesses on instruction lines that will stall:

```
xr0 = r5 + r6;;  
r6 = [j0 + 4];;  
xr8 = r3 * r0;;
```

#### APPLIES TO REVISION(S):

0.2, 0.4

### 2. 03000011 - Floating Point Denormal Result:

---

#### DESCRIPTION:

If the exact floating point result is not zero, but its absolute value is less than min-norm (mantissa all zero and exponent = 1), the result is always flushed to zero. It is never rounded to min-norm.

#### WORKAROUND:

None

#### APPLIES TO REVISION(S):

0.2, 0.4

**3. 03000044 - Wait Cycle on Other Master Read:**

---

**DESCRIPTION:**

If the ratio between LCLK and CCLK is lower than 3, when data is read from one of the TigerSharc's internal memory by another TigerSharc, there is a wait cycle.

**WORKAROUND:**

None, but the performance penalty can be decreased if the data is packed in quad words. When a quad word is read, there is a wait cycle only on the first data cycle (word or long word depending on the bus width). The next data of the same quad word is ready in time.

**APPLIES TO REVISION(S):**

0.2, 0.4

**4. 03000099 - JTAG emulation will fail during idle if interrupts are disabled:**

---

**DESCRIPTION:**

When trying to initiate jtag emulation operations during idle (or low power idle) the global interrupts (pmask[60] and imask[60]) need to be enabled.

This might cause a problem in two cases:

1. When starting jtag emulation during idle (after reset) and changing the reset value of imask before that (probably a very remote situation).
2. When starting jtag emulation after an idle command and not verifying interrupts are enabled before going into idle state.

**WORKAROUND:**

Keep Global HW interrupt - bit 60 of IMASK set, and bit 60 of PMASK should be cleared before changing to IDLE. This is standard practice before going into idle.

**APPLIES TO REVISION(S):**

0.2, 0.4

**5. 03000100 - IALU operand fetch uses bypassed result of canceled instruction:**

---

**DESCRIPTION:**

On successive IALU instructions on which the first instruction is load/store with post-modify and update and the second instruction uses the same ialu register if the first instruction is canceled due to illegal/non\_aligned address access, the second instruction will still use the bypassed result of the ialu register instead of fetching the old value.

Note1: the ialu register is not updated on the first instruction since the instruction is canceled.

Note2: if an exception is issued on the first instruction, and the exception is handled then there will be no problem since the second instruction will be aborted.

An example of the successive instructions:

```
xr31:28= q[j0+=0x4] ;;  
nop;;  
nop;;  
nop;;  
xr3:0= dab q[j0+=0x4] ;;
```

If j0 is initialized to 0x3 then the first instruction will be canceled because of attempt to access non-quad-aligned address, j0 will still be 0x3 the bypassed value will be 0x7, the second instruction is executed, the accessed address will be 0x7 instead of 0x3 and j0 will be updated to 0xB instead of 0x7.

**WORKAROUND:**

None - This is wrong behavior on user error.

**APPLIES TO REVISION(S):**

0.2, 0.4

**6. 03000102 - TCB CX value is incorrect after block completion:**

---

**DESCRIPTION:**

At the end of the last transfer of the DMA block, the TCB Cx register incorrectly acquires the initial value of CX (from the beginning of the block transfer) instead of the correct value of '0'.

**WORKAROUND:**

None

**APPLIES TO REVISION(S):**

0.2, 0.4

## 7. 03000105 - BTB miss on target of non-quad aligned jump:

---

**DESCRIPTION:**

This is a performance issue only.

When a target of a jump is a non quad aligned fetch of another jump that should cause a BTB hit, the BTB indicates a miss.

Example:

```
if xxx, jump _firstlabel;;  
.  
.  
.  
.align 4;  
NOP;;  
_firstlabel:  
if yyy, _seconlabel;;
```

Even if the second jump is in the BTB, it is referred to as a miss. However, if it is pushed to an aligned address (remove the NOP;;), BTB is a hit.

**WORKAROUND:**

Add the directive .align\_code 4; to quad align the address. In the example above,

```
if xxx, jump _firstlabel;;  
.  
.  
.  
.align_code 4;  
  
_firstlabel:  
if yyy, _seconlabel;;
```

**APPLIES TO REVISION(S):**

0.2, 0.4

## 8. 03000107 - DMA Priority - Cluster Bus Channels:

---

**DESCRIPTION:**

If the DMA channel #3 is being served, and a higher priority DMA channel takes over for a while, when it finishes, the priority should return to the DMA channel #3. However, in case when channels #3, #2 & #1 are all requesting service, where channels #3 & #1 have the high priority bit set, instead of returning to channel #3, the priority goes to channel #1. This is the only failing case.

**WORKAROUND:**

None - performance issue

**APPLIES TO REVISION(S):**

0.2, 0.4

## 9. 03000108 - Flyby Data Alignment:

---

**DESCRIPTION:**

A TS mastered read that follows a slow protocol flyby read by the same TS master, gets the data with a shift of one word.

e.g. for quad read:

```
instead of : 0x33333333,22222222,11111111,00000000  
we get      : 0x22222222,11111111,00000000,33333333
```

**WORKAROUND:**

Do not use flyby for slow protocol devices.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 10. 03000109 - AutoDMA Overrun:

---

**DESCRIPTION:**

No additional writes to the AUTODMA register should be performed for four core clock cycles after the AUTODMA channel has completed. Otherwise an external port deadlock may occur.

**WORKAROUND:**

Check the DMA status for the AUTODMA completion, then wait at least four core clock cycles before doing another write to the AUTODMA register.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 11. 03000110 - PRC to RAS programming:

---

**DESCRIPTION:**

The PRC to RAS delay as programmed in the SDRCON results in one more cycle than programmed (as long as the bus mastership is not changed):

00 - 2 cycles.  
01 - 3 cycles.  
10 - 4 cycles.  
11 - 5 cycles.

But the actual delay is one more cycle (i.e.: 3,4,5,6).

This could be a performance issue when the user wants to use 2 cycles delay. On all other cases, the SDRCON should be programmed to one less cycle delay. (e.g. if 4 cycles delay is required, then 3 cycles should be programmed.)

**WORKAROUND:**

Program 1 cycle less than required.  
For 2 cycles PRC to RAS - no workaround

**APPLIES TO REVISION(S):**

0.2, 0.4



**12. 03000111 - Performance monitor count timing:**

---

**DESCRIPTION:**

The performance counter is updated two cycles after the relevant event occurs.  
If the read of PRFCNT is on the next cycle after the relevant event occurs, the result will be wrong.

**WORKAROUND:**

Add `NOOP;; NOOP;;` before reading performance counter

**APPLIES TO REVISION(S):**

0.2, 0.4

**13. 03000114 - Link does not detect error if checksum byte is not received:**

---

**DESCRIPTION:**

No checksum error is detected if no checksum byte and subsequent dummy byte are received in single quad word transfer.  
The following scenario describes the error:

**Conditions:**

The Link receiver LCTLx.VERE=1.

**Sequence:**

1. A Quad Word without checksum byte and subsequent dummy byte at the end is received.
2. LxCLKIN is stopped HIGH.

**The receiver reaction is:**

1. LSTAT.RST[3] is set to "1", a receive interrupt or dma request is issued and LBUFRx is updated.
2. Upon detection of LxCLKIN stopped HIGH the Checksum generation/comparison mechanism is reset and the block is ready to start a new receive/transmit.
3. In this case no checksum error is detected by the receiver.

**WORKAROUND:**

Users should be aware that there might be cases where the transmitted data is not verified even if LCTLx.VERE bit is set.

**APPLIES TO REVISION(S):**

0.2, 0.4

**14. 03000115 - Fairness in Link to Link Communication is not always maintained:**

---

**DESCRIPTION:**

If link port of TigerSharc 1 is transmitting more than 64 quads (with psize=0 for packet size set for 64 quad-words) to link port of TigerSharc 0, TigerSharc 1 pauses after 64 quads but not long enough to allow TigerSharc 0 a token switch. Once the transmitter runs out of data to transmit, it will allow the token switch.

**WORKAROUND:**

If the 64 quads packet size fairness is required, it can be implemented by setting up the DMA and core to do it, i.e. the DMA is programmed to transfer 64 QW and to issue an interrupt when it is done. The interrupt routine is responsible for delaying the new TCB load in order to allow the token switch.  
Chaining might be used, but it cannot guarantee that the second block transmission is delayed enough time to allow a token switch.

**APPLIES TO REVISION(S):**

0.2, 0.4

**15. 03000117 - Wrong Link Transmission Time Out Error:**

---

**DESCRIPTION:**

TigerSHARC can set the Transmission Time Out Error, LSTATx.TER[1], regardless of whether it has data to transmit or is idle.

**WORKAROUND:**

Users should be aware that LSTATx.TER[1] bit can be set even if the transmitter is idle, and handle this error indication accordingly.

**APPLIES TO REVISION(S):**

0.2, 0.4

**16. 03000120 - The Link port does not stop on error:**

---

**DESCRIPTION:**

The link does not stop on error. The link should stop receiving data until the error status is read from LSTATx.

**WORKAROUND:**

Users should be aware that once an error interrupt is issued by the link, new data might be received or transmitted by the time the interrupt is serviced.

**APPLIES TO REVISION(S):**

0.2, 0.4

**17. 03000121 - Address bus contention when mastership is changed:**

---

**DESCRIPTION:**

The output disable and output enable timings can overlap by up to 2.5ns. During the cycle that the mastership is transferred from one master to another, there is a bus idle cycle. There is no conflict on the data bus, but there is potential for conflict on the address bus when the new bus master starts driving. This can result in additional noise in the system.

**WORKAROUND:**

None

**APPLIES TO REVISION(S):**

0.2, 0.4

**18. 03000129 - Stall in the DESPREAD Instruction:**

---

**DESCRIPTION:**

The DESPREAD instruction has a two cycle dependency. Thus, if its result is used in the next instruction, a stall will occur.

Example: The following two code examples

```
TR0 = DESPREAD(R7:4,THR1:0) + TR0;;  
TR0 = DESPREAD(R3:0,THR1:0) + TR0;;
```

and

```
TR0 = DESPREAD(R7:4,THR1:0) + TR0;;  
NOP;;  
TR0 = DESPREAD(R3:0,THR1:0) + TR0;;
```

will execute in the same number of cycles (3).

**WORKAROUND:**

If possible, schedule code in the register-interleaved fashion. In any case, the results of the execution are correct, only performance may suffer.

**APPLIES TO REVISION(S):**

0.2, 0.4

**19. 03000133 - RETI is wrong when two interrupts arrive during IDLE:**

---

**DESCRIPTION:**

When two interrupts arrive separated by one cycle and an idle command reaches pipeline stage EX2 at the same time as the first interrupt service routine's instruction is being fetched (two cycles after the interrupt request), the RETI register will be updated the second time and will receive a value of the first instruction of the interrupt service routine. This will cause the TigerSHARC to enter an infinite loop inside the interrupt service routine.

**WORKAROUND:**

Whenever an idle command is used, disable all hardware interrupts by writing 0 to imask[60] two cycles before the idle command. In the same instruction line as the idle command, enable the interrupts by writing 1 to imask[60]. This will prevent interrupts from occurring at the same time as the idle command reaches stage EX2 of the pipeline.

**APPLIES TO REVISION(S):**

0.2, 0.4

**20. 03000134 - Lock of the BTB not functional:**

---

**DESCRIPTION:**

If BTB is locked and then unlocked (using BTBLK bit in SQCTL), some of the valid entries may be invalidated.

**WORKAROUND:**

Do not use lock of the BTB feature (keep BTBLK bit in SQCTL cleared).

**APPLIES TO REVISION(S):**

0.2, 0.4

**21. 03000136 - RETI return address corrupted during interrupt nesting:**

---

**DESCRIPTION:**

When the first instruction of a nested interrupt service routine reaches execution pipeline stage A while the nested interrupt disable command (`retib = ...`) is at stage E1, the nested interrupt will be served after RETI is restored. Thus, nested interrupt service routine will modify the RETI register but will return one instruction past the restore of that register.

**WORKAROUND:**

When using nested interrupts, the RETI restore command at the end of the interrupt service routine (`retib = ...`) should be executed twice. The first one is to disable nested interrupts and the second one is to restore RETI.

**APPLIES TO REVISION(S):**

0.2, 0.4

**22. 03000137 - Conditional predicted RTI or RDS instructions are always partially executed:**

---

**DESCRIPTION:**

When an RTI or an RDS command is conditional and the condition is false, the interrupt will still be reduced to a subroutine, i.e. the appropriate bits of PMASK will be cleared and the supervisor mode may be reduced to user mode. The branch of the RTI will not occur.

**WORKAROUND:**

Use a conditional jump before RTI/RDS to direct the flow of execution.

**APPLIES TO REVISION(S):**

0.2, 0.4

**23. 03000138 - Link Transmit with PSIZE=0 fails if NACK'd in the middle:**

---

**DESCRIPTION:**

Link port transmission with PSIZE=0 (packet size set for 64 quad-words) fails if there is ever a NACK from the receiver before the packet size limit is encountered. The failure is that not enough edges (14 instead of 16) are generated for the last quad-word before the token enable. The receiver appears to discard this last quad-word for this case.

**WORKAROUND:**

Do not use PSIZE = 0. If arbitration fairness is required, do it via software by breaking the DMA blocks into smaller pieces.

**APPLIES TO REVISION(S):**

0.2, 0.4

**24. 03000139 - Interrupts missed when writing to ILATST/ILATCL:**

---

**DESCRIPTION:**

During the cycle when core writes to ILATST/ILATCL, interrupts are blocked from latching in ILAT and are, therefore, missed.

**WORKAROUND:**

Do not write to ILATST or ILATCL.

**APPLIES TO REVISION(S):**

0.2, 0.4

**25. 03000140 - AutoDMA channels priority bit is always set:**

---

**DESCRIPTION:**

When reading AutoDMA channels value (DC13 and DC14), the priority bit is always set. A user can not write a different value to this bit.

**WORKAROUND:**

Not needed.

**APPLIES TO REVISION(S):**

0.2, 0.4

**26. 03000141 - Exception is ignored when regprot or performance monitor counter wrap occur with no valid instruction:**

---

**DESCRIPTION:**

A register protection and a performance monitor counter wrap sometimes fail to generate an exception. The protection and wrap still work properly, just the exception is not generated. The problem occurs if the cause of the exception happens at the time when the EX2 stage of the processor's pipeline contains no valid instructions.

**WORKAROUND:**

None

**APPLIES TO REVISION(S):**

0.2, 0.4

**27. 03000142 - Stuck IFIFO when AUTODMA is paused:**

---

**DESCRIPTION:**

If the AUTODMA channel is paused and data has been sent to AUTODMA, the IFIFO transactions will be stuck.

**WORKAROUND:**

Do not pause AUTODMA channels.

**APPLIES TO REVISION(S):**

0.2, 0.4

**28. 03000146 - LDQM and HDQM pins have pull-downs instead of pull-ups:**

---

**DESCRIPTION:**

LDQM and HDQM pins have internal pull-downs instead of pull-ups.

**WORKAROUND:**

Connect external pull-ups of 5K or stronger on these pins.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 29. 03000148 - L0DIR and L1DIR always drive:

---

**DESCRIPTION:**

L0DIR and L1DIR always drive, even during reset.

**WORKAROUND:**

L0DIR and L1DIR cannot be connected to any signal that can drive at any time.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 30. 03000149 - Predicted unaligned branches may fail:

---

**DESCRIPTION:**

Non-quad-aligned predicted branches may fail and jump to a wrong address.

**WORKAROUND:**

Ensure that all predicted branches are quad-aligned by inserting `.align_code 4;` directive before them. Alignment can also be automatically performed by the assembler through the use of the assembler switch '-align-branches-lines'.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 31. 03000150 - LCLKRAT pins internal pull-downs are insufficient:

---

**DESCRIPTION:**

LCLKRAT pins internal pull-downs are insufficient to keep these pins low.

**WORKAROUND:**

Use external 1 KOhm pull-down resistors on LCLKRAT pins that are required to be low.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 32. 03000151 - Illegal op-codes in the pipeline:

---

**DESCRIPTION:**

Illegal op-codes in the TigerSHARC's pipeline may cause improper execution. These op-codes can enter the pipeline if memory following an unconditional branch is uninitialized.

**WORKAROUND:**

If code following an unconditional branch is uninitialized, insert 6 single line NOPs following the branch to keep the pipeline filled with valid op-codes.

**APPLIES TO REVISION(S):**

0.2, 0.4

**33. 03000152 - MAC saturation:**

---

**DESCRIPTION:**

When multiplying two integer values that cause saturation, the value accumulated is the truncated multiply value instead of the saturated one. This is true for both signed and unsigned MAC's.

Example:

```
XR0 = 0x0000ffff;;
XR1 = 0x0;;
XR2 = 0x0000ffffE;;
XR3 = 0x0;;
xMR1:0 += R3:2 * R1:0 (IUC);; ---- mr0 should be ffff after saturation, instead it is 0001
```

**WORKAROUND:**

None

**APPLIES TO REVISION(S):**

0.2, 0.4

**34. 03000153 - Status register missing stall:**

---

**DESCRIPTION:**

If the result of a compute block operation is written into x/ySTAT in any of the two subsequent instruction lines, the old register value will be written instead. This dependency requires two stalls, but the processor inserts only one.

Example: (for compute X, the same stands for compute Y):

```
xR0 = R1 + R2;;
NOP;; ---- or STALL
xSTAT = xR0;;
```

xSTAT gets the old value of XR0, instead of the value computed in the previous line.

**WORKAROUND:**

Add 2 NOPs when there is a dependency between the result of the compute instruction and the source of the Write Status instruction.

**APPLIES TO REVISION(S):**

0.2, 0.4

**35. 03000156 - AC flag is set when Rs = -Rm:**

---

**DESCRIPTION:**

The ALU instruction

```
B/L/SRs(d) = -Rm(d);;
```

sets the AC flag (x/ystat bit[3]). It should be CLEARED according to spec.

Example:

```
yr4 = 0x380f89;;  
yr5 = 0x0;;  
yr21:20 = -r5:4;;
```

Resulting YSTAT = 0xb, instead of 0x3.  
Which means that the AC flag is set.

**WORKAROUND:**

Since in either case of real or the intended functionality, the flag is always updated with the fixed value, any instruction predicated with the condition can be made unconditional or deleted from the code.

**APPLIES TO REVISION(S):**

0.2, 0.4

**36. 03000157 - SDRAM Performance in core accesses:**

---

**DESCRIPTION:**

If core does SDRAM accesses with gaps in between and the bus is unlocked, the SDRAM controller does not keep the pages open. Thus, it has to open the page for every access which will severely affect its performance.

**WORKAROUND:**

1. Accesses by DMA will have no gaps and will work at full SDRAM speed.
2. If core accesses and performance are required, the bus should be locked first (using BUSLK register).

**APPLIES TO REVISION(S):**

0.2, 0.4

**37. 03000158 - DMA chaining stops:**

---

**DESCRIPTION:**

Chaining of an external port DMA that reverses the direction of transfer may fail.

**WORKAROUND:**

Do not chain external port DMA to external port DMA that reverses the direction of transfer.

**APPLIES TO REVISION(S):**

0.2, 0.4



**38. 03000162 - Branch from external memory may fail.:**

---

**DESCRIPTION:**

A branch executed from external memory may fail.

**WORKAROUND:**

Do not execute code from external memory.

**APPLIES TO REVISION(S):**

0.2

**39. 03000163 - FDEP out of range:**

---

**DESCRIPTION:**

When position field in `LRsd+=fdep` and `Rs+=fdep` instruction with option (ZF) is out of range (>64), the result is 0x0 instead of the previous value. For example, if R4 = 0x4102 (i.e. position=0x41=65 and length=2), then instruction

```
LR1:0 += fdep R3:2 by R4 (ZF);;
```

results in R1:0 = 0. The correct operation would be to leave the value unchanged.

**WORKAROUND:**

Make sure that position is less than or equal to 64.

**APPLIES TO REVISION(S):**

0.2, 0.4

**40. 03000165 - MAC underflow exception followed by another MAC:**

---

**DESCRIPTION:**

When the underflow exception flag in the multiplier (mu) is set (by a mul floating point instruction, or a direct write to the stat register) and the underflow exception is enabled, it will cause the exception (as it should).

If one of the next instructions is a multiplier instruction, it will not clear those bits, hence the program flow will again branch to the exception ISR.

**WORKAROUND:**

Clear the mu bit in the exception routine.

**APPLIES TO REVISION(S):**

0.2, 0.4

**41. 03000166 - No bypass when reading from LSTAT.TST right after writing to LBUFTX:**

---

**DESCRIPTION:**

Reading from LSTAT right after writing to LBUFTX, will result in the old value of TST.

**WORKAROUND:**

Add a NOP or another instruction between the write and the read.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 42. 03000167 - Chaining DMAs may not work with DMAR external requests:

---

**DESCRIPTION:**

If the DMAR external request of the next block is asserted before chaining to it has completed, that DMA channel might become stuck. The reason is that DMA channel accumulates only those requests that occurred after the channel has been enabled (i.e. chained to) again.

**WORKAROUND:**

Ensure that DMAR request is asserted after its DMA has been enabled (chained to).

**APPLIES TO REVISION(S):**

0.2, 0.4

## 43. 03000168 - First instruction of an ISR has to be quad-aligned:

---

**DESCRIPTION:**

If the first instruction of an interrupt service routine is not quad-aligned, the sequencer may fail to execute properly.

**WORKAROUND:**

Quad-align all ISRs with `.align_code 4;` directive. Note that if the ISR is in a separate module, the linker will quad-align it automatically.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 44. 03000169 - First three instruction lines in the ISR can not contain IALU instructions:

---

**DESCRIPTION:**

If an IALU instruction that modifies an IALU register is in the pipeline when an interrupt occurs and the same register is accessed inside the first three instructions of the ISR, the modification outside the ISR may happen twice, once before the interrupt and once after the return from the interrupt.

**WORKAROUND:**

Ensure that the first three instruction lines of an ISR contain no IALU instructions.

**APPLIES TO REVISION(S):**

0.2

## 45. 03000172 - RTI, RETI, CJMP and CJMP\_CALL must be NP:

---

**DESCRIPTION:**

A BTB hit of RTI, RETI, CJMP and CJMP\_CALL may cause the sequencer to branch to a wrong location.

**WORKAROUND:**

All RTI, RETI, CJMP and CJMP\_CALL instructions must have (NP) option to prevent their entry into the BTB.

**APPLIES TO REVISION(S):**

0.2

**46. 03000173 - RDS command should be performed only at interrupt level:**

---

**DESCRIPTION:**

If user and supervisor modes are both used in the system, command RDS should only be performed when the processor is at an interrupt level.

**WORKAROUND:**

1. Run the system in supervisor mode only (NMOD bit in SQCTL set).
2. Check PMASK and execute RDS only if PMASK is not zero.

**APPLIES TO REVISION(S):**

0.2, 0.4

**47. 03000174 - SDRAM Performance:**

---

**DESCRIPTION:**

SDRAM transactions that change pages take additional 6 cycles.

**WORKAROUND:**

None

**APPLIES TO REVISION(S):**

0.2, 0.4

**48. 03000175 - SBSRAM access in an MP system must use an ACK controlled buffer:**

---

**DESCRIPTION:**

If a read from an SBSRAM is attempted after an MP read from another TigerSHARC and the MP read from the TigerSHARC is NACK'd, the command for the read from the SBSRAM will still happen and both the SBSRAM and the TigerSHARC will drive the data bus.

**WORKAROUND:**

Use negated ACK to disable the chip select to the SBSRAM.

**APPLIES TO REVISION(S):**

0.2, 0.4

**49. 03000176 - AC set by instruction  $R_s = R_m + R_n$ ,  $R_a = R_m - R_n$ ; but should be cleared:**

---

**DESCRIPTION:**

Instruction

$R_s = R_m + R_n, R_a = R_m - R_n;$

may set AC flag. It should be cleared.

**WORKAROUND:**

Make sure that code flow does not rely on the AC flag being cleared by the add/subtract instruction.

**APPLIES TO REVISION(S):**

0.2, 0.4

**50. 03000177 - Internal Bus Arbiter Performance Issue #1:**

---

**DESCRIPTION:**

If a DMA requests the OFIFO with low-priority while the OFIFO is full, core virtual bus transfers will not be granted a bus until the OFIFO is no longer full. If the external bus is held off by another bus master (host or another TS101S), the DMA and the core transactions will both be stuck until the OFIFO is allowed to empty one entry.

## Conditions:

- DMA requests the OFIFO with low-priority.
- DMA does not also request an internal bus with high-priority.
- OFIFO is full.
- Core performs virtual bus transfer.

## Result:

Core virtual bus transfer is stalled until the OFIFO is not full.

For a more detailed explanation please refer to EE-190 which can be provided by [dsp.support@analog.com](mailto:dsp.support@analog.com).

**WORKAROUND:**

1. Configure DMA for high-priority OFIFO requests.
2. Configure DMA for high-priority OFIFO requests and pause the DMA for core driven external port transfers.
3. Configure DMA for low-priority OFIFO requests and avoid the OFIFO being full for extended periods of time.
4. Configure DMA for high-priority OFIFO requests and divide large DMAs into smaller blocks that use chaining to setup the TCBs for the next block.

**APPLIES TO REVISION(S):**

0.2

**51. 03000178 - BUSLOCK cannot be toggled too quickly:**

---

**DESCRIPTION:**

If bus is locked (BUSLOCK is set) and is unlocked and then locked again too quickly, it may stay unlocked.

**WORKAROUND:**

After a write to change BUSLOCK, check SYSTAT BUSLOCK bit and wait till the value in SYSTAT has changed correctly before issuing another write.

**APPLIES TO REVISION(S):**

0.2, 0.4

**52. 03000179 - SDRAM in high memory space should not use A15 as bank select:**

---

**DESCRIPTION:**

If 1k page, 64 bit bus SDRAM is used, then TigerSHARC's A15 shouldn't be connected to SDRAM Bank select.

**WORKAROUND:**

In systems that use more than half of the total SDRAM memory space, user should use the TigerSHARC's A15 as one of the higher (A11-..) address pins & not as a bank select.

**APPLIES TO REVISION(S):**

0.2, 0.4

**53. 03000180 - Time between exit of SDRAM self refresh to active is too short:**

---

**DESCRIPTION:**

TigerSHARC's SDRAM controller keeps time of tRAS+tRP (see SDRCON register definitions) between exit of self-refresh and active commands - this may be too short and SDRAM timing specs may not be met.

**WORKAROUND:**

1. Increasing the value of tRAS and tRP in SDRCON can satisfy the timing requirement of SDRAM, but will make all accesses to SDRAM slower.
2. TigerSHARC puts SDRAM into self-refresh mode only when it gives the bus to the Host. Host can take the SDRAM out of self-refresh and delay for the necessary number of cycles before returning the bus to the TigerSHARC.

**APPLIES TO REVISION(S):**

0.2, 0.4

**54. 03000181 - SDRAM precharge with tRP=5 takes four instead of five cycles:**

---

**DESCRIPTION:**

SDRAM precharge with tRP=5 (see SDRCON[10:9] register bits) takes four instead of five cycles when the bus mastership is passed from one TigerSHARC to another.

**WORKAROUND:**

Do not use tRP=5 (see SDRCON[10:9] register bits).

**APPLIES TO REVISION(S):**

0.2, 0.4

**55. 03000182 - BUSLOCK with HBR can cause external bus deadlock:**

---

**DESCRIPTION:**

If the Host request the bus (HBR) and the TigerSHARC assert its Buslock at the same time, then the TS will not execute external transactions, but also will not grant the bus (HBG) to the host.

**WORKAROUND:**

1. Do not use Buslock when there is a Host master in the system.
2. After asserting HBR, Host can sample Buslock pin and de-assert its HBR if Buslock is asserted & HBG is not.

**APPLIES TO REVISION(S):**

0.2, 0.4

**56. 03000232 - LCTL3:0 accessible as quads only through the external bus:**

---

**DESCRIPTION:**

LCTL3:0 can only be accessed as a quad when addressed through multiprocessor space.

**WORKAROUND:**

Access LCTL3:0 from the external bus as quads.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 57. 03000233 - Internal Bus Arbiter Performance Issue #2:

---

**DESCRIPTION:**

If a DMA requests the OFIFO with high-priority while the OFIFO is not full, any core IALU transfers and/or IFIFO low-priority transfers will not be granted a bus until the DMA access to the OFIFO completes.

**Conditions:**

- DMA requests the OFIFO with high-priority.
- OFIFO is not full.
- Core performs any IALU transfer and/or IFIFO performs a low-priority transfer.

**Result:**

Core transfer and/or IFIFO transfer are stalled until the DMA access to the OFIFO completes.

For a more detailed explanation please refer to EE-190 which can be provided by [dsp.support@analog.com](mailto:dsp.support@analog.com).

**WORKAROUND:**

Configure DMA for low-priority OFIFO requests.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 58. 03000234 - Internal Bus Arbiter Performance Issue #3:

---

**DESCRIPTION:**

If a DMA requests the OFIFO with high-priority while the OFIFO is not full and also requests a virtual bus transfer, any core IALU transfers and/or IFIFO low-priority transfers will not be granted a bus until the DMA access to the OFIFO completes.

**Conditions:**

- DMA requests the OFIFO with high-priority.
- DMA also requests a virtual bus transfer (link port to/from external address space only).
- OFIFO is not full.
- Core performs any IALU bus transfer and/or IFIFO performs a low-priority transfer.
- All three busses (0, 1, & 2) are requested by core and/or IFIFO.

**Result:**

Core IALU bus transfer and/or IFIFO transfer are stalled (even when busses are available) until the DMA acquires a bus for the V3/OFIFO access.

For a more detailed explanation please refer to EE-190 which can be provided by [dsp.support@analog.com](mailto:dsp.support@analog.com).

**WORKAROUND:**

Configure DMA for low-priority OFIFO requests.

**APPLIES TO REVISION(S):**

0.2, 0.4

## 59. 03000235 - One Dimensional DMA chaining fails if DY field is non-zero:

---

**DESCRIPTION:**

Non-zero DY field in one dimensional DMA may cause the chaining to fail.

**WORKAROUND:**

When one dimensional DMA is used, DY field must be set to 0.

**APPLIES TO REVISION(S):**

0.2, 0.4

**60. 03000236 - External Port DMA chaining may fail when paused:**

---

**DESCRIPTION:**

If an external port DMA is paused during chaining when one of the two TCB's has chained while the other has not, and then TCBs are modified, un-pausing the DMA will result in a chaining failure and the DMA channel will be stuck.

**WORKAROUND:**

If such a DMA process is required (as it may be in cases of chain insertions), after pausing the type field of the two DP registers must be examined. If both are legal or both are inactive (000), continue with the modification of the TCB's as required. Otherwise (which is the case when one has chained and the other has not), un-pause the channel, wait, pause and check again.

**APPLIES TO REVISION(S):**

0.2, 0.4

**61. 03000237 - loading cjmp from SDRAM misses the interlock:**

---

**DESCRIPTION:**

Loading CJMP from an external address misses the interlock, i.e. if CJMP is used after the load, it may still contain the old value.

**WORKAROUND:**

Load from external memory into an IALU or compute block register and then copy it into CJMP.

```
j10 = [j31+0x4000000];;  
CJMP = j10;;  
if true, cjmp (abs);;
```

**APPLIES TO REVISION(S):**

0.2, 0.4

**62. 03000268 - Global hardware interrupt disable allows interrupts for two cycles:**

---

**DESCRIPTION:**

An interrupt that arrives within two cycles after disabling global interrupts in IMASK (i.e. clearing of IMASK[60]) may still get serviced.

**WORKAROUND:**

1. Disable IMASK[60] at least 2 cycles before code that is intended to be non-interruptible.
2. If IMASK needs to be modified by an interrupt service routine, store IMASK value at the beginning of the routine and restore it at the end of it. User cannot count on the value of IMASK[60] to be set when in an interrupt service routine.

**APPLIES TO REVISION(S):**

0.2, 0.4

**63. 03000288 - RETS/DBGE gets wrong value when an exception/emulation arrives together with a reti update:**

---

**DESCRIPTION:**

When an instruction line which causes an exception/emulation also includes an explicit write to RETI/DBGE/RETS and this write is delayed (due to bus arbitration) RETI/DBGE/RETS will get a wrong value.

**WORKAROUND:**

Restrict writes to RETI/DBGE/RETS from a UREG to prevent the write from stalling. Also, do not parallel writes to RETI/DBGE/RETS with any other instruction that may stall.

**APPLIES TO REVISION(S):**

0.2, 0.4

**64. 03000302 - Reading from SDRAM when SDREN bit in SDRCON is clear causes deadlock:**

---

**DESCRIPTION:**

Reading SDRAM when SDREN bit in SDRCON register is clear will cause deadlock.

**WORKAROUND:**

Ensure that SDRAM is enabled in SDRCON before accessing it.

**APPLIES TO REVISION(S):**

0.2, 0.4

**65. 03000318 - Cancelled access to uninitialized SDRAM still causes "SDRAM access error":**

---

**DESCRIPTION:**

Cancelled access to SDRAM (such as a conditional access with condition evaluated as false) may generate an "SDRAM Error" in SYSTAT and thus a hardware interrupt ("HWILL" in ILAT) if SDRCON has not been initialized.

**WORKAROUND:**

Always initialize SDRCON before performing any accesses to SDRAM, even if the accesses are conditional and the condition is expected to fail.

**APPLIES TO REVISION(S):**

0.2, 0.4

**66. 03000322 - Aborted LxSTATC or LxBUFRX read still updates LxSTAT:**

---

**DESCRIPTION:**

If a read of LxSTATC or LxBUFRX is in the pipeline, but aborted, the Link port Status register is still updated.

Possible reasons for the cancelled transaction are:

- 1) Speculative flow that is aborted.
- 2) Read of LxSTATC or LxBUFRX is conditional and the condition evaluates to false.
- 3) Read of LxSTATC or LxBUFRX is the last instruction before an interrupt routine (and is, thus, flushed).
- 4) Read of LxSTATC or LxBUFRX is in the last 4 instructions before an exception has occurred (and is, thus, flushed).

**WORKAROUND:**

Perform the LxSTATC / LxBUFRX read in an exception trap handler routine.

**APPLIES TO REVISION(S):**

0.2, 0.4



**67. 03000334 - MAC instruction of one format followed by compact in different format might cause wrong saturation value:****DESCRIPTION:**

A MAC instruction of one format followed by a compact instruction in different format might produce incorrect value. For example:

```
yMR3:2 == r13*r16;;
ySr7:6 = compact MR3:0(IS);;
```

may cause the MAC instruction to produce incorrect result. Note that in the example case a 32-bit format MAC is followed by a 16-bit format compact, thus leading to the possible problem.

**WORKAROUND:**

Add a 'nop' instruction line between consecutive instruction lines with different formats of mac and compact instructions.

**APPLIES TO REVISION(S):**

0.2, 0.4

**68. 03000335 - Incorrect AV flag in 'Rs=compact Rsd +/- Rsd' command:****DESCRIPTION:**

In the Rs= COMPACT Rsd +/- Rsd instruction:

AV flag is set if the result of the add/subtract operation causes negative overflow, even if the result after rounding does not cause an overflow.

In the following example, the add/subtract results causes negative overflow, but after rounding, there is no overflow. The AV flag should be cleared but it is not.

```
xr0=0xc100;;
xr1=0x0;;
xr2=0xbe80;;
xr3=0x0;;
xr6=0xa100;;
xr7=0x0;;
xr8=0x2180;;
xr9=0x0;;
xBr10=COMPACT Sr7:6 - Sr9:8;;
xBr5=COMPACT Sr3:2 + Sr1:0;;
```

**WORKAROUND:**

If it is necessary to use the AV status flag from a COMPACT instruction as a condition for another instruction, user should reconfirm in software that the resulting output of COMPACT instruction actually underflowed.

**APPLIES TO REVISION(S):**

0.2, 0.4

**69. 03000336 - Incorrect AV flag in 'Rs(d)=ABS (Rs(d) +/- Rs(d))(U)(X)' command:**

---

**DESCRIPTION:**

The AV flag value is incorrect for the following instructions only:

```
(B/S/L)Rs(d) = ABS (Rm(d) + Rn(d))(X)
(B/S/L)Rs(d) = ABS (Rm(d) - Rn(d))(X)
(B/S/L)Rs(d) = ABS (Rm(d) - Rn(d))(U)
```

- 1) The ABSplus instruction with option (X) should set AV flag only when both operands are 0x80...0; Instead, it is set for every overflow.
- 2) The ABSminus instruction with options (X) & (U), should always clear the AV flag; But currently AV flag remains set for every overflow.

Here are a few examples where the AV flag is set when it should have been cleared:

```
xr3=0xfffffffffe;;
xr8=0x80000001;;
xr7=0x7fffffffef;;
xr14=0x80000000;;
xBr24=abs(r3 + r8)(X);;
xSr12=abs(r7 - r3)(X);;
xr8=abs(r14 - r7)(U);;
```

**WORKAROUND:**

If it is necessary to use the AV status flag from a ABSplus instruction as a condition for another instruction, user should reconfirm in software that both input operands to the instruction were 0x80...0. The AV status flag of ABSminus instruction should not be used as a condition for another instruction.

**APPLIES TO REVISION(S):**

0.2, 0.4

**70. 03000340 - IA illegal address exception is not canceled when the command is aborted:**

---

**DESCRIPTION:**

If an IALU instruction generates a software exception, the exception will still be serviced even if this instruction is aborted.

**WORKAROUND:**

1. Disable software exceptions.
2. Recognize an illegal IALU access in the software exception handler and exit the handler.

**APPLIES TO REVISION(S):**

0.2, 0.4

**71. 03000343 - DLL may lock incorrectly:**

---

**DESCRIPTION:**

The DLL may lock incorrectly under the following scenarios:

- 1) During power-up.
- 2) After power-up if the SCLK input violates specification.

The result of the failure state for the above two scenarios is that the device will violate data sheet AC timing specifications. Once the DLL fails to lock correctly the only proven solution is to cycle power on the device.

This anomaly is device dependent. Systems that have undergone power cycling tests and have not exhibited the failure may be considered safe. The workaround(s) should be implemented for robustness.

**WORKAROUND:**

Workaround:

The DLL will always lock correctly on power up if one of the two following conditions is met:

- 1) SCLK must be stable and in specification before Vdd\_a is powered up.

The implementation can be done in one of two ways:

- a) Power the SCLK clock buffer from a supply separate from Vdd\_io. Power up the supplies in this order: Clock driver (to in-spec condition), then Vdd/Vdd\_a (if Vdd/Vdd\_a are tied together), then Vdd\_io.
- b) Delay Vdd\_a until after SCLK is stable. This can be done by powering Vdd and Vdd\_a from separate supplies and powering-up the supplies in this order: Vdd, then Vdd\_io (with clock driver to in-spec condition), then Vdd\_a. Alternatively, Vdd\_a could be delayed using a large capacitor on the filtering circuit.

-- OR --

- 2) SCLK must be held low until all TS101S power supplies are stable. Once the supplies are stable, SCLK can be enabled, and must immediately be in-spec. It is acceptable for the first pulse of SCLK to be clipped (i.e. the enable to the clock buffer does not need to be phase-aligned to SCLK).

Once DLL lock has been achieved, SCLK must be kept running at a constant frequency. The ONLY exception is that SCLK can be stopped by gating it low. If SCLK frequency must be changed, the procedure is to gate it low, reset the oscillator frequency, and then enable it directly to the in-spec frequency (as in power up option #2 above).

**APPLIES TO REVISION(S):**

0.2, 0.4

**72. 03000355 - BTB invalidate may not invalidate all entries:****DESCRIPTION:**

A btb invalidate command does not flush updates for fetches that are already in the pipeline. The result may be that some entries are not invalidated.

**WORKAROUND:**

In order to ensure that entries in the btb are invalidated correctly, hardware interrupts should be disabled before the btb invalidate command and the following 9 quad-words should not contain predicted branches.

For example:

```
<disable interrupts>
nop;;
btbinv;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
nop;nop;nop;nop;;
<enable interrupts>;;
```

**APPLIES TO REVISION(S):**

0.2, 0.4

**73. 03000374 - Conditional J-IALU compute stall:****DESCRIPTION:**

A conditional (non-IALU) J-IALU compute operation followed (in execution flow) by an instruction with a NOP in the first slot of the instruction line will result in a 3 cycle stall.

Any conditional (non-IALU condition) J-IALU compute instruction such as the following:

```
if <non_IALU_cond>; do, Js1 = F(Jm, Jn)
or
if <non_IALU_cond>, jump/cjmp/cjmp_call/call/reti/rti; else, Js1 = F(Jm, Jn)
```

followed (in execution flow) by any instruction line with a NOP in the first instruction slot such as any of the following:

```
nop;;
nop;inst1;;
nop;inst1;inst2;;
nop;inst1;inst2;inst3;;
```

will result in the 3 cycle stall.

**WORKAROUND:**

For performance critical code sections, ensure that the three instruction lines following (in execution flow) a conditional J-IALU compute instruction do not contain a NOP instruction in the first instruction slot.

**APPLIES TO REVISION(S):**

0.2, 0.4