



Intel® Open Source HD Graphics, Intel Iris™ Graphics, and Intel Iris™ Pro Graphics

Programmer's Reference Manual

For the 2015 - 2016 Intel Core™ Processors, Celeron™ Processors, and Pentium™ Processors based on the "Skylake" Platform

Volume 8: Workarounds

May 2017, Revision 1.0



Creative Commons License

You are free to Share - to copy, distribute, display, and perform the work under the following conditions:

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **No Derivative Works.** You may not alter, transform, or build upon this work.

Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Implementations of the I2C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

* Other names and brands may be claimed as the property of others.

Copyright © 2017, Intel Corporation. All rights reserved.



Table of Contents

| | |
|----------------------------------|----------|
| Workarounds | 1 |
| Display Workarounds | 6 |



Workarounds

This page lists all BSpec narrative workarounds for BXT. Note that the functional area for each item is listed below, so you can search on this value or other content on this page. Also note that a "BSpec ID" has been assigned to make it easier to reference these items.

| BSpec ID | Functional Area/Component | | Submitted By | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|-------------------|--------------|---------------------------------------|---|-----------------|
| 0302 | 3D | | | WaCompressedResourceRequiresConstVA21 | GFXDRV: Hot spotting issue with render target compression. WA: Align lossless compressed resource allocations to 2MB or have fixed virtual addresses. This WA is specifically for steppings where HW will not fix. | All |
| 0303 | 3D | MEDIA_STATE_FLUSH | | | A MEDIA_STATE_FLUSH with no options must be added after a GPGPU_WALKER command if pooled mode is enabled. | |
| 0522 | KMD | | | WaldleLiteRestore | SW must always ensure ring buffer head pointer is not equal to tail pointer of a context, whenever it is submitted to HW for execution. WA: Driver should not submit a context with head = tail. | BXT:C0 |
| 0523 | KMD | | | WaldleLiteRestore | SW must always ensure "Force Restore Bit" in the context descriptor is set for a preempted context that is getting resubmitted. WA: Driver should not submit a context with head = tail. | BXT:C0 |



| BSpec ID | Functional Area/Component | | Submitted By | Workaround Name | Workaround Description | Valid Steppings |
|----------|----------------------------|----------------------|--------------|-----------------|---|-----------------|
| 0525 | Command Stream Programming | | | N/A | On BXT "RS enabled Batch Buffer Per Context" can be programmed with MI_SEMAPHORE_WAIT command in register poll mode. Resource streamer stops executing the "RS enabled Batch Buffer Per Context" on encountering MI_SEMAPHORE_WAIT command. SW must ensure all the commands meant to be executed by Resource Streamer from "RS enabled Batch Buffer Per Context" must be programmed prior to programming the MI_SEMAPHORE_WAIT command. | All |
| 0891 | GPGPU | Midthread preemption | Jim Valerio | | When a pooled workload is mid-thread preempted, followed by RC6 or Render power gate, then the mid-thread restore process does not restore pooling-enabled signal soon enough. This causes context corruption and possibly a hang. WA: Insert "Media_Pool_State" command in the WA batch buffer that is run before a context restore (MMIO 0x21C4). | ALL |

Workarounds



| BSpec ID | Functional Area/Component | | Submitted By | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------------|--|--------------|---------------------|--|-----------------|
| 0908 | SFC | SFC Crop Limitation for VEBOX+SFC Mode | Karthik N | WaDisableSFCSrcCrop | <p>Below are the cases to switch from SFC to Render for VEBOX+SFC mode</p> <p>Case 1. ((SurfaceHeight > 1120) && (Top > 1120))</p> <p>Case 2. ((SurfaceHeight > 1120) && (Bottom < SurfaceHeight))</p> <p>Case 3. ((SurfaceHeight > 1120) && (Left > 0))</p> <p>Case 4. ((SurfaceHeight > 1120) && (Right < SurfaceWidth))</p> | ALL |
| 0909 | GS PrimID bug with Tessellation | Peter Doyle | | | GS Clock gating must be disabled under the following conditions: Tessellation enabled, GS enabled, GS PrimitiveID enabled. | BXT: ALL |
| 0911 | CPD | Bus Hang | Jim Valerio | | <p>Hang occurs with a change of CPD Exit ordering of messages. Reverting that change avoids the hang.</p> <p>WA: Set 0xA194[9]=0 to revert the change. This register is set in BIOS and locked.</p> | BXT:ALL |



| BSpec ID | Functional Area/Component | | Submitted By | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|----------------|----------------|-----------------|---|-----------------|
| 0913 | GPGPU | | Hema C Nalluri | | <p>Hang occurs when CPD flows happen concurrently when CS is reading MMIO outside GT.</p> <p>WA: Reads to MMIO registers outside GT (Register Address > 0x40000) are not supported through MI commands (MI_LOAD_REGISTER_REG, MI_STORE_REGISTER_MEM) programmed in command buffer.</p> | BXT:ALL |
| 0915 | HDC | Atomic Counter | Jim Valerio | | <p>Hang occurs with Atomic Counter message with binary operations (i.e. have a separate data operand), in some dynamic load situations.</p> <p>WA: Replace use of hidden counter with an explicit counter location, and then use a typed or untyped Dword Atomic operation message instead.</p> | ALL |
| 0921 | GTI | GAM | Niran Cooray | | <p>When running legacy contexts with GTTC enabled, faults on upper levels of page table entries can result in a GAM hang. Only applicable when faults are supported.</p> <p>WA: Disable GTT Cache</p> | ALL |

Workarounds



| BSpec ID | Functional Area/Component | | Submitted By | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|-----|---------------------|------------------------------|--|-----------------|
| 0924 | Blitter | | Andrew Vanderheyden | DisallowOddSizedSmallFCBlits | <p>FC Blitter cannot handle a blit whose y-height%4 == 3 and y-height <= 8. This causes a system hang.</p> <p>WA:</p> <p>These blits must be detected and sent to the legacy blitter engine, not the fast copy engine.</p> | ALL |
| 0925 | GTI | GAM | Niran Cooray | | <p>Clock gating issue in gamtwarb fub in gamtunit as its not accounting for the evicting cycle present in the ingress fifo btw GAMD and GAMT. Due to this the write buffer ID is used for multiple write transactions - which is not expected in GAM and results in a logic hang.</p> <p>We could have this scenario if we have S/W register based WCP/RCP invalidation [write to register 4AAC], instead of fence based invalidation.</p> <p>WA:</p> <p>Disable clock gating for the Write arbiter logic(0x4A08[10]=1) through GFX driver</p> | ALL |



Display Workarounds

This page lists all workarounds for Display. Note that the functional area for each item is listed below, so you can search on this value or other content on this page. Also note that a "BSpec ID" has been assigned to make it easier to reference these items.

| BSpec ID | Functional Area/Component | | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|-----------|-----------------|--|-----------------|
| 0839 | Display | MIPI | | <p>While MIPI is enabled in dual link mode (DPI MIPI_PORT_CTRL EN=1 or DBI writing a frame), the pipe attached to MIPI must have at least one plane or cursor enabled or pipe scaling enabled. Otherwise the MIPI image will shift and won't correct itself until MIPI is disabled and PG1 is disabled and re-enabled.</p> <p>BXT-Bstep: In the disable sequence, Planes can be disabled after disabling Device ready and MIPI port control</p> <p>BXT-Astep: Scalar as well needs to be ON before we start BXT B-step disable sequence.</p> | All |
| 0854 | Display | Backlight | | <p>Backlight PWM may stop in the asserted state, causing backlight to stay fully on.</p> <p>WA: Before disabling PWM, set 0x46530 bit 13 for PWM1 or bit 14 for PWM2. The bits can remain set without harm.</p> | SIWA_FROM_B0 |



| BSpec ID | Functional Area/Component | | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|----------------|-----------------|--|-----------------|
| 0855 | Display | MIPI DSI | | <p>MIPI DSI dual link front back mode has image shifts if all planes and cursor on the pipe are disabled and pipe scaling (panel fitting) is disabled.</p> <p>WA: The pipe attached to MIPI DSI dual link in front back mode must always have at least one plane or cursor enabled or pipe scaling (panel fitting) enabled. The added plane can be hidden by using plane alpha to make it transparent.</p> | All |
| 0904 | Display | VGA | | <p>Corruption with some VGA modes and clock frequencies.</p> <p>WA: Set the bits 30:29 of MMIO register 0x41004 to 01b.</p> | ALL |
| 1110 | Display | FBC + PSR/PSR2 | | <p>Missing flips when FBC is enabled with PSR link off/PSR2 deep sleep scenarios.</p> <p>WA: When FBC is enabled with PSR/PSR2, set bit 30 of MMIO register 0x420CC to 1b.</p> | ALL |



| BSpec ID | Functional Area/Component | | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|------------------------|-----------------|--|-----------------|
| 1124 | Display | Panel Power Sequencing | | <p>Incorrect panel power up delays</p> <p>WA: Wait at least 100us between programming PP_ON_DELAYS and enabling Power State Target in PP_CONTROL, or disable dpls clock gating before programming PP_ON_DELAYS and leave disabled until after enabling Power State Target in PP_CONTROL.</p> <p>North display dpls clock gate disable is register 0x46530 bits 17 and 16.</p> <p>South display dpls clock gate disable is register 0xC2020 bit 29.</p> | All |
| 1128 | Display | MIPI DSI | | <p>MIPI DSI Rcomp failures which can cause image corruption at hot temperatures.</p> <p>BXT WA: In the MIPI Enable Sequence, just after the step that sets P_CR_GT_DISP_PWRON_0_2_0_GTTMMADR MIPIO_RST_CTRL to 0x1, program DSI_PHY_DW6 HS_OVR_EN=0x1 and HS_OVR_VALUE=0xA.</p> | All |
| 1135 | Display | IPC | | <p>Display underrun when IPC is enabled.</p> <p>WA: The Line Time programmed in the WM_LINETIME register should be half of the actual calculated Line Time.</p> <p>Programmed Line Time = 1/2*Calculated Line Time</p> | All |

Workarounds



| BSpec ID | Functional Area/Component | | Workaround Name | Workaround Description | Valid Steppings |
|----------|---------------------------|-----|-----------------|--|-----------------|
| 1136 | Display | PSR | | <p>Display underrun with PSR single frame update or PSR2, and planes with less than watermark level 7.</p> <p>WA: When using PSR single frame update or PSR2, all enabled planes must have enabled up to watermark level 7. If any plane cannot support level 7, then single frame update or PSR2 cannot be enabled.</p> | All |