Intel® Iris® Xᵉ MAX Graphics Open Source

Programmer's Reference Manual

For the 2020 Discrete GPU formerly named "DG1"

Volume 14: Workarounds

February 2021, Revision 1.0

Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks

Customer is responsible for safety of the overall system, including compliance with applicable safety-related requirements or standards.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described 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.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

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.

Table of Contents

Workarounds Overview

impact	lineage	title	bspec_wa_details	sku_impact
data_corruption	2202389218	Invalid occlusion query results with "Pixel Shader Does not write to RT" bit	When Pixel Shader Kills Pixel is set, SW must perform a dummy render target write from the shader and not set this bit, so that Occlusion Query is correct.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption,hang	1407528679	while loop cases causing issues in jeu fused mask	Disable Structured Control Flow by setting EnableVISAStructurizer.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	1604061319	Command Streamer not sending flush to VF and SVG after Fence during PipeControl sequence of commands causing hang	In set shader mode 3DSTATE_CONSTANT_* needs to be programmed before BTP_* At CS RTL boundary, this is the order of commands 1. Constant cycle on MCR 2. Fence command 3. BTP on MCR At SVG RTL boundary, this is the order of commands seen because of MCR delay 1. Fence 2. Constant Cycle on MCR 3. BTP on MCR At fence, although fence is a non pipeline state, CS is optimizing the flush and NOT sending the flush.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1604402567	VFURB dropping data in some scenarios involving 256 bit element format	WA Name: WaNo256BitVFCompPacking Component packing of vertex elements associated with 256-bit surface formats is not supported due to a HW bug. WA: All components of vertex elements associated with 256-bit surface formats MUST be enabled.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	1406337848	Atomic operation does not work on compressed data	Software should ensure at the time of the Atomic operation the surface is resolved (uncompressed) state.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1406798080	MMIO remapping feature in command streamer MI_* register access functions doesn't work for certain offsets	WA Name: SelectiveMMIORemapEnable "MMIO Remap Enable" can be enabled only for the "Register Offsets" mentioned in the "MMIO remap table" of a given engine on which the MI commands accessing the MMIO registers are getting executed.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
hang	1406689936	POSH/PTBR workloads can hang if varying tile counts within a tile pass and preemption happens	WA Name: PoshPreemptionTilePassInfoCmd "Tile Count" value programmed must be same in the 3DSTATE_PTBR_TILE_PASS_INFO command programmed for "Start of Tile Pass" and "End of Tile Pass".	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1305657336	OVR Issue where initialize that follows the restart is not deferred causing an invalid page to be allotted for storing the tokens	OVR Issue if pocs_ovr_restart is asserted within 256 clks after the ctx restore is done. WA: The WA could be to do a page pool size mmio write with a value of 0 followed by 256 noops before any page pool restart.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1605787622	DualContext : During CSB update GAM will not have dualcontext information causing issue	Program GAM 0xCE90 Register's Dual Context Mode bits whenever RCU mode control reg 0x14800 is programmed. (same value of bit0 with mask).	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	2205427594	Media compression issue: Issue during Macroblock processing during error concealment can result in page faults/engine soft hang	Use the first valid reference (or the closest reference if POC is available to detect) from reference list if available to fill all unused reference frame address regardless coding type (I, P or B) to prevent potential page fault. If valid reference is not available from reference list, use decode output surface for dummy reference if MMCD is disabled, otherwise make an intermediate allocation as dummy reference. Correspondent reference index needs to be programmed as frame.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1408224581	Semi pipelined flush not backpressuring when stencil buffer state is enabling thread dispatch resulting in hang	Issue: Semi pipelined flush not backpressuring when stencil buffer state is enabling thread dispatch. Workaround: An additional pipe control with post-sync = store dword operation would be required.(w/a is to have an additional pipe control after the stencil state whenever the surface state bits of this state is changing).	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
power	1606766515	RCU should ignore(reset) Media Sampler DOP status of engine which is idle	In Dual Context Mode of operation, a context can get executed on an engine and switch out with Media Sampler DOP Clock Gate Disabled (can be on Render Engine or Compute Engine). In such a scenario the corresponding engine keeps the Media Sampler DOP Clock Gate Disabled until further a context gets submitted resetting the state to Media Sampler DOP Clock Gate Enabled or both the engines go Idle. This will lead to ineffective DOP Clock Gate of Media Sampler. This may happen under following circumstances: • SW didn't submit the workload exercising Media Sampler bracketed between PIPELINE_SELECT with Media Sampler DOP Clock Gate Disable and Enable respectively in a single dispatch. SW may avoid the inefficient Media Sampler DOP Clock Gate Enable by avoiding above mentioned scenarios, i.e • Make workloads accessing Media Sampler are bracketed between PIPELINE_SELECT with Media Sampler DOP Clock Gate Disable and Enable respectively.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
performance	1806565034	Sampler cache can be thrashed in certain cases involving texture arrays resulting in low performance	WA: In Render Surface State, the SurfaceArray bit should not be set unless the depth of the arrayed surface is > 1.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1506855762	OVR causes a Page fault when running out of free pages in PTBR PAGE POOL	The driver has to map 1 page of dummy resource to address PTBR_PAGE_POOL_BASE_ADDRESS + (0xFFFF * 4KB).	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	1606931601	GRF source swap feature for SIMD16 with Src0 scalar and bundle conflict between Src1/Src2 is causing the GRF read issue.	WA: Driver must set E4F4[14]=1 to disable early read/Src Swap.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
	1408981979	Default BCredits on MBUS insufficient to meet required display bandwidth	Issue: Default BCredits on MBUS insufficient to meet required display bandwidth WA: Display MBUS_DBOX_CTL* registers should be programmed with BCredit value of 12 (e.g. 7003C[12:8] = 0xC). Note that there are multiple instances of this register, one for each display pipe (A, B, C, D).. All instances should be programmed to the same value.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1408979724	Coarse Pixel Shading - hang can occur in color pipe if CPS Aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting MMIO register 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption,performance	1409044764	Revert DAPRSS Blend Opt RTL	Disable Float Blend Opt Strict Mode by always setting MMIO register 0x07304 Bit[12] to 0	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	1409077218	LACE histograms don't properly get cleared due to timing issue related to vblank vs pipe enable signals	To mitigate this issue, essentially, we need to keep the LACE gated clock toggling during vblank_rise edge. In order to do this, we could use one of the three options 1. we could move the LACE function disable to after the poll on TRANS_CONF_A to wait till the pipe transcoder is OFF. this way we will still have clocks toggling during vblank_rise and clear the frame_histograms_done flag. 2. The other options is during the modeset disable sequence, we could not disable LACE function and keep it asserted so that the LACE gated clocks continue to toggle. However, this works only if LACE is expected to be enabled back during the re-enable sequence. 3. set clock DCPR gating disable during the modeset disable sequence and re-enable them after-wards.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
performance	1409142259	Coarse Pixel Shading - perf issue with floating point render targets if CPS Aware color pipe	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
		optimization is enabled		
data_corruption	1409178076	Coarse Pixel Shading - corruption can occur with R11G11B10_FLOAT render target if CPS Aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409207793	Coarse Pixel Shading - data corruption can occur if CPS Aware color pipe performance optimization enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption,hang	1409217633	Coarse Pixel Shading - hang or data corruption can occur with 16X MSAA if CPS aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409120013	Underrun when FBC is compressing with odd plane size and first segment is only 3 lines	FBC causes screen corruption when plane size is odd for vertical and horizontal. Set 0x43224 bit 14 to 1 before enabling FBC. It is okay to leave it set when FBC is disabled.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409252684	Coarse Pixel shading Data corruption due to dropping CP Subspan with Alpha2Coverage if CPS aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	1409210145	DPT isn't sending VRR enable indication to dcpr while in Push VRR mode	Issue: DPT (transcoder unit) isn't sending VRR enable indication to dcpr (clocks and reset unit) while in Push VRR mode (the primary VRR mode). WA: DPT should send VRR enable indicator to DCPR even while Push mode is enabled.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	1409028688	VP9 VDEnc encode: segmentation within 64x64 block picks wrong segment id	Program same stream-in segmentation id for all four 32x32 blocks of SB64.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
other	1409364714	RCS/POCS/CCS/BCS: Reserved fields in "Instdone" Registers are tied to "0" instead of "1"	Software must ignore the Reserved Fields in the INSTDONE register.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409392000	Data Corruption with Coarse Pixel Shading + Dual Source Blend + Dual SIMD8 pixel shader dispatch	CPS cannot be enabled alongside Dual SIMD8 Dispatch and Dual Source Blend	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	2201730850	Spec clarification: Z Clear Color Location	There was a hole in the definition for Clear value for the case of D24X8 depth surfaces. Added a programming note in RENDER_SURFACE_STATE as well as in Clear Color section describing the need to write the converted value to the lower 16B. Also, this programming note is removed by HW Managed Z Clear.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1408615072	clk gating bug in VS unit can cause UAV counters for HS, GS, TDS to result in hang	Disable the vsunit clock gating. Offset 9434 bit 3	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1606932921	RCS is not waking up fixed function clock when specific 3d related bits are programmed in pipecontrol in compute mode	SW WA to program PIPE_CONTROl with RT Flush and CS Stall prior to PIPE_SELECT to Compute. This will be revisited while implementing dove tailing to wake FFDOP and issue flush to both 3D and compute Pipe	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1607304331	DS GS checker mismatch on UAV increment marker	Workaround is to put a flush in before the TDS UAV coherent primitive if ps is the shader the tds uav is coherent with.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
other	1207137018	3D Tiled-YF surface corruption in MIP tail LODs because of X-adjacent RCC cacheline composition	WaSetMipTailStartLODLargertoSurfaceLOD RCC cacheline is composed of X-adjacent 64B fragments instead of memory adjacent. This causes a single 128B cacheline to straddle multiple LODs inside the TYF MIPtail for 3D surfaces (beyond a certain slot number) , leading to corruption when CCS is enabled for these LODs and RT is later bound as texture. WA: If RENDER_SURFACE_STATE.Surface Type = 3D and RENDER_SURFACE_STATE.Auxiliary Surface Mode != AUX_NONE and RENDER_SURFACE_STATE.Tiled ResourceMode is TYF or TYS, Set the value of RENDER_SURFACE_STATE.Mip Tail Start LOD to a mip that larger than those present in the surface (i.e. 15)	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409342910	[DX12] WGF11RenderTargets failure	Driver should always program Color Clamp Range Based on Table in Pre-Blend Color Clamping.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1409600907	PipeControl with Depth Flush enable can result in hang	“PIPE_CONTROL with Depth stall Enable bit must be set with any PIPE_CONTROL with Depth Flush Enable bit set “	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1409689360	Corruption with FBC and plane enable/disable	Corruption with FBC around plane 1A enabling. In the Frame Buffer Compression programming sequence "Display Plane Enabling with FBC" add a wait for vblank between plane enabling step 1 and FBC enabling step 2.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption,hang	1607854226	Certain Non-Pipelined State commands on RCS should work in PipeSelect compute, but don't because of FFDOP clk gating	Listed commands below are the non-pipeline state commands that may get programmed when PIPELINE_SELECT is set to Media/GPGPU in RenderCS. Due to known HW issue when these commands are executed in Media/GPGPU mode of operation, the new state may not get latched by the destination unit and stale value will prevail. In order to WA this issue SW must temporarily change the PIPELINE_SELECT mode	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>lineage</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td></td><td>to 3D prior to programming of these command and following that shift it back to the original mode of operation to Media/GPGPU. Since all the listed commands are non-pipelined and hence flush caused due to pipeline mode change must not cause performance issues. • STATE_BASE_ADDRESS • STATE_COMPUTE_MODE • 3DSTATE_BINDING_TABLE_POOL_ALLOC Example: Programming with No WA. PIPELINE_SELECT – GPGPU MEDIA_VFE_STATE MEDIA_INTERFACE_DESCRIPTOR_LOAD GPGPU_WALKER 3DSTATE_BINDING_TABLE_POOL_ALLOC MEDIA_VFE_STATE MEDIA_INTERFACE_DESCRIPTOR_LOAD GPGPU_WALKER …….. Programming with WA. PIPELINE_SELECT – GPGPU MEDIA_VFE_STATE MEDIA_INTERFACE_DESCRIPTOR_LOAD GPGPU_WALKER PIPELINE_SELECT – 3D 3DSTATE_BINDING_TABLE_POOL_ALLOC PIPELINE_SELECT – GPGPU MEDIA_VFE_STATE MEDIA_INTERFACE_DESCRIPTOR_LOAD GPGPU_WALKER</td><td></td></tr>
<tr><td></td><td>1507384622</td><td>Depth stats (occlusion query) gives wrong results when using Render Target Independent Rasterization (STATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0) and no pixel shader bound</td><td>When 3DSTATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0, SW should program a dummy pixel shader in case occlusion query is required.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td></td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td></td><td>2205427594</td><td>Media compression issue: Issue during Macroblock processing during error</td><td>Use the first valid reference (or the closest reference if POC is available to detect) from reference list if available to fill all unused</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr></table></td></tr>
</table>

impact	lineage	title	bspec_wa_details	sku_impact
		concealment can result in page faults/engine soft hang	reference frame address regardless coding type (I, P or B) to prevent potential page fault. If valid reference is not available from reference list, use decode output surface for dummy reference if MMCD is disabled, otherwise make an intermediate allocation as dummy reference. Correspondent reference index needs to be programmed as frame.	ALL | a0 | | driver_permanent_wa
	1809012548	MPEG2 & AVC Encode: As part of encode operation, CONDITIONAL_BATCH_BUFFER_END command fetches Compare data (related to Panic mode/QP) and pushes to hw engine for subsequent frame; sends wrong data if the comparison data was in upper 4 QWORD of cacheline	Software must ensure the “Compare Address” programmed in MI_CONDITIONAL_BATCH_BUFFER_END command for the Compare Data Qword in memory is always within the first 256b of a cacheline (i.e address bit[5] must be ‘0’). MI_STORE_REGISTER_MEM have the same address condition for register image status mask (08b4)and image status data(08b8) so that the mem write is always within the first 256b of a cacheline	sku: ALL | stepping_impacted: a0 | stepping_fixed: | wa_status: driver_permanent_wa
data_corruption,other	1409347922	Color pipe incorrectly counts unlit pixels in some cases when Coarse Pixel Shading is used with CPS aware color pipe optimization enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL | stepping_impacted: a0 | stepping_fixed: | wa_status: driver_permanent_wa
hang	1607934025	Hang if using VEBox for GEC+3DLut and concurrent SFC scaling	The issue is seen when GEC+3dlut in VEBOX are used, for HDR Tone mapping, along with SFC for scaling operation. Software can revert back to the legacy behavior where used AVS for scaling operation post HDR TM in VEBOX.	sku: ALL | stepping_impacted: a0 | stepping_fixed: | wa_status: driver_permanent_wa
	14010480278	DARBFunit early clock gating leading to underrun	Disable clock gating for DARBFunit. Set register offset 0x46530 bit 27 (DARBF Gating Dis) to 1 before first enabling display planes or cursors and keep set. No need to clear after disabling	sku: ALL | stepping_impacted: a0 | stepping_fixed: | wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>lineage</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td></td><td>planes</td><td></td></tr>
<tr><td>security</td><td>14010017096</td><td>[SECURITY] Accumulator is not currently cleared with GRF clear exposing its content to new context.</td><td>Clear ACC register before EOT send mov(16) acc0.0:f 0x0:f</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td></td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>other</td><td>1608008084</td><td>Register reads to 0x6604 is incorrect</td><td>SW is required to only write 0x6604 as the read will not return the correct value if doing a read-modify-write. The default value for this register is zero for all fields and there are no bit masks. Updating this register requires SW to know the previous written value to retain previous programming.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td></td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>performance</td><td>1607983814</td><td>LNCF MOCS settings are cleared on soft reset of RCS/POCS/CCS</td><td>Upon render reset, the driver needs to reprogram LNCFCMOCS0 to LNCFCMOCS31.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td></td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	lineage	title	bspec_wa_details	sku_impact
hang	1306835611	Resubmission of preempted POSH/PTBR workload can hang on Tile_Select command	Workaround for PTBR Workloads: On RenderCS in “Tile Based Rendering Mode” (TileCount > 0) of operation, preemption on 3DPRIMITVE command must be disabled with in a “Tile Pass”. A “Tile Pass” is defined by “Start of Tile Pass” and “End of Tile Pass” programmed by 3DSTATE_PTBR_TILE_PASS_INFO command. This must be done by setting the “Disable Preemption on 3DPRIMTIVE Command” attribute in register bit[10] following a “Start of Tile Pass” command and must be reset following “End of Tile Pass” command. Preemption opportunity on tile boundaries within a “Tile Pass” must be explicitly provided by programming of MI_ARB_CHECK command prior to programming of a 3DSTATE_PTBR_TILE_SELECT command. Workaround for POSH Workloads: On RenderCS in “POSH Mode” (TileCount = 0) of operation, preemption on last two 3DPRIMITVE in submitted work load command must be disabled by adding two extra dummy 3DPRIMITVE command at the end of the workload with associated 3DSTATE_CLIP command such that these dummy 3DPRIMITVE command always generate CULL token from OVR to render command streamer and these are bracketed around disabling the pre-emption on 3Dprimitive with Reg 0x2580[10] programmed. 3DSTATE_CLIP should be programmed with DW2 field ClipMode as 0x3 to indicate REJECT_ALL and dummy 3DPRIMITIVE are programmed with PrimitiveTopologyType as 3DPRIMITIVE_TRILIST so that OVR generated visibility token for this type of workload.	<table><tr><th>sku</th><th>stepping_impacted</th><th>stepping_fixed</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td></td><td>driver_permanent_wa</td></tr></table>

impact	lineage	title	bspec_wa_details	sku_impact
data_corruption	1808850743	EU: goto instruction with uniform predicate in CS SIMD32 kernel does not work as expected	ITo workaround this, a kernel change is proposed. Since hardware is able to turn off channels at goto but unable to change fuse mask correctly, combine the channel enable register with dispatch mask and use it to predicate NoMask instructions. Kernel with workaround looks like below. To ensure the predicate mask has all channels enabled, we can specify the 'any' modifier with the size of the JEU instruction execution size. (W) mov(1) r107.0:uw sr0.4:uw //load the dispatch mask into a temp register. (~f0.0) goto (16|M0) ELSE_UNSTRUCT ELSE_UNSTRUCT or (16|M0) r21.0<1>:uw r21.0<1;1,0>:uw 0x8:uw (W) and(1) f0.0:uw ce0.0:uw r107:uw //and the ce mask and dispatch mask loaded into r107. (W&f0.0.any16h) add (16|M0) r23.0<1>:uw r23.0<1;1,0>:uw 0x0001:uw //predicate the NoMask instruction. 'any' modifier with 16h specified because JEU execution size is 16. goto (16|M0) ELSE_UNSTRUCT END_IF_UNSTRUCT ELSE_UNSTRUCT: join (16|M0) END_IF_UNSTRUCT or (16|M0) r21.0<1>:uw r21.0<1;1,0>:uw 0x10:uw (W) and(1) f0.0:uw ce0.0:uw r107:uw //and the ce mask and dispatch mask again, before every NoMask instruction. If channel enables haven't changed, then once before the first NoMask instruction. (W&f0.0.any16h) add (16|M0) r23.0<1>:uw r23.0<1;1,0>:uw 0x0100:uw //predicate the NoMask instruction. END_IF_UNSTRUCT: join (16|M0) POST_END_IF_UNSTRUCT POST_END_IF_UNSTRUCT: This workaround is needed for all NoMask instructions inside branching instruction blocks where EUs can diverge. Exceptions: if the NoMask instruction execution size is greater than the JEU block executions size like below, an additional instruction is required to ensure flag is written for the upper channels to use. The 'any modifier will not be required in this case.' //JEU block execution size of 16 nop //do | | (W) and(1) f0.0:uw ce0.0:uw r107:uw (W) and(32) (ne)f0.0 f0.0:uw 0xffff:uw //execution size same as NoMask instruction size. Immediate value as wide as the jeu block execution size. (W&f0.0) add(32) ..... | While(16) //JEU block execution size of 4 nop //do | | (W) and(1) f0.0:uw ce0.0:uw r107:uw (W) and(16) (ne)f0.0 f0.0:uw 0xf:uw //execution size same as NoMask instruction execution size which is 16. Immediate value as wide as jeu block execution size which is 4. (W&f0.0) add(16) ..... (f1.0) while(4)	sku | stepping_impacted | stepping_fixed | wa_status ALL | a0 | | driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
hang	14010477008	Display fetch path enters unrecoverable hang from corrupted media compression control codes	Media compression (AKA End2End compression) must be disabled for displayable media surfaces. Do not enable Media Decompression in the display plane control registers.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14010595310	A64 scatter messages incorrectly dispatched to same address if Addr[47:32] differ in a msg among simd lanes	IGC W/A is to avoid such A64 scatter messages by adding a loop around each A64 vector load/store so that on each iteration only lanes with identical high 32-bit addresses will execute.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14010443199	[DAPRSS] Color: DaprSsDaprSc.ss_phase0.cpq_mask.sample_mask Mismatch	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14010698770	[DAPRSS] DAPRSS Sending Blend CData Encoding For Fill CPQ	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	14010755945	PSDunit is dropping MSB of the blend state pointer from SD FIFO	Limit the Blend State Pointer to < 2G	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	22010178259	AUTOCLONE: BW Buddy CTL Register has incorrect default value for TLB Request timeout	Program BW_BUDDY_CTL0 and BW_BUDDY_CTL1 "TLB Request Timer" field to 8h.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1607794140	[3D-WHCK] wgf11resourceaccess workload fail	Before fast clearing any resource, SW must partially resolve the resource i.e. corresponding CCS for the resource MUST NOT be in CLEAR state	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14010818508	[DAPRSS] Data Corruption on R10G10B10_FLOAT_A2_UNORM After Blend2Fill	See the Errata on Pre-Blend Color Clamping	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14010915640	[DAPRSS] Repcol with R10G10B10_FLOAT_A2_UNORM Not Properly	De-feature Repcol Messages	sku; stepping_impacted; stepping_fixed; wa_status

impact	lineage	title	bspec_wa_details	sku_impact
		Down-converted		ALL a0 driver_permanent_wa
	1409820462	AV1 ALN LR temp flops need to be reloaded at top of new tile	AV1 decoder will put all the tiles programming into single batch buffer (frame based) [instead of 1 tile per batch buffer]	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
	14010942852	Audio 8K1port - For certain VDSC bpp settings, hblank asserts before hblank_early, leading to a bad audio state	WA details can be found at: Display Engine > North Display Engine Registers > Audio > Audio Programming Sequence under "Audio Hblank Early Sequence"	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1607871015	[AMFS] SW Workarounds for AMFS flush	1.A pipe control flush with "AMFS flush Enable" set and "DC flush enable set" must be sent down the pipe before a context switch, when compute shaders do evaluate. 2. if compute shader does evaluates, and SW needs to flush the AMFS pipe, it has to first send a pipecontrol flush to the compute pipe and then switch to 3D pipe before sending a pipecontrol with "Command Streamer Stall Enable", AMFS flush Enable, and DC flush enable set on it 3. If compute shaders do evaluate, disable premption, until AMFS data is flushed out of all the caches. 4. All shaders that perform evaluates must send a Cache Flush message to the sampler with a non-zero read-length after all evaluates are issued and before End-Of-Thread 5. Compute shaders run on a CCS context must not issue AMFS evaluates. All AMFS evaluates must run in an RCS context	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1607956946	Handle block deref size is part of 3dstate_sf & is non-privileged register bit	Driver will have to correctly program bits [30:29] on every 3dstate_SF programming (driver would have to reprogram this field with all the rest of the fields disabled prior to 3DPRIMITIVE command.) SF Body: [bits 30:29]	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
other	1808692196	Read/write access to OAG registers blocked for non-priv batch buffers from RCS/POCS/CCS; required for certain performance instrumentation cases to work	WA: READ/WRITE ACCESS to OAG Registers 1. Software must use the Force_To_Non_Priv registers to enable Read/WRITE access to the below register offsets RCS: 0xD920 - 0xD93F and 0xDA10 - 0xDA27 (2 ranges) POCS: 0xD920 - 0xD93F and 0xDA10 - 0xDA27 (2 ranges) CCS: 0xD920 - 0xD93F and 0xDA10 - 0xDA27 (2 ranges)	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	14010222001	SW WA for SFC+HEVC+VP9 : Dummy VDCTRL insertion to support b2b DV	VD(HEVC/VP9)-SFC mode (both single and scalability): In case of HEVC/VP9 with SFC enabled, insert 2 dummy VD_CONTROL_STATE packets with data=0 after every HCP_SFC_LOCK VD(HEVC/VP9) scalability mode, insert 2 dummy VD_CONTROL_STATE packets with data=0 before every HCP_TILE_CODING_DW1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	16010904313	*CS: sometimes ctx time stamp register doesn't get restored to value from the engine context image on context switch	The below workaround must be used to overcome the ctx timestamp issue 1. For BCS/VCS/VECS: -- In the Per-Context WABB (workaround batch buffer) Software must program 3 back to back LRM (MI_LOAD_REGISTER_MEM) commands with - For RCS/CCS -- In the Indirect Context Pointer, Software must program 3 back to back LRM (MI_LOAD_REGISTER_MEM) commands with Dw0[19] = 1, Register Address = CTX_TIMESTAMP and Memory Address = LRCA + 108Ch. 2. The first two MI_LOAD_REGISTER_MEM commands must have Dw0 bit 21 = 1 3. The third MI_LOAD_REGISTER_MEM command must have Dw0 bit 21 = 0 4. All three commands must have “Add CS MMIO Start Offset” Dw0[19] = 1 to enable auto addition of CS MMIO Start Offset. For Example in case of RCS, if LRCA for a given context is DEADh the below commands must be programmed in the per-context workaround batch buffer. 1. MI_LOAD_REGISTER_MEM (dw0[19] = 1, dw0[21]= 1, REGISTER ADDR = 3a8h, Memory Address = DEADh + 108Ch 2.	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
			MI_LOAD_REGISTER_MEM (dw0[19] = 1, dw0[21]= 1, REGISTER ADDR = 3a8h, Memory Address = DEADh + 108Ch 3. MI_LOAD_REGISTER_MEM (dw0[19] = 1, dw0[21]= 0, REGISTER ADDR = 3a8h, Memory Address = DEADh + 108Ch	
	14011264657	dupunit not generating line_pop indication for plane with minimum size	WA: Plane horizontal minimum size in PLANE_SIZE register need to be increased according to the following: 8bpp: 18 16bpp: 10 32bpp,yuv212,yuv216: 6 64bpp: 4 NV12: 20 P010,P012,P016: 12	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1607880884	LRR Cmd addr 2360 not correctly remapped	Software must use only MI_LOAD_REGISTER_IMM to program 0x2360 register	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	1409863162	Multicontext: rsi message retrieves inst_Base address from RCS for both contexts	When dual context or dual queue (e.g. async compute) is enabled, SW cannot rely on the RSI message for getting the instruction base address due to this bug. If needed, driver can pass the instruction base address to the kernel as a kernel argument	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	18011246551	Diagonal error propagation for vertical intra refresh on H264 VDEnc	The solution is to disable all prediction modes that uses reference values from not refreshed area. Those are modes 3,7 for 4x4 and modes 0, 2, 3, 4, 5, 7 for 8x8 (due to filtering). In the driver code it looks like: AvcIntra4X4ModeMask = 0x88 AvcIntra8X8ModeMask = 0xBD	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	16010946120	Media Compression : counter overflow leads to premature flush done reporting - can result in corruption due to dirty cachelines not getting evicted when high read latency occurs	Issue: Media compression block can have a counter overflow issue in certain long memory latency scenarios that leads to premature flush and some dirty cachelines don't get evicted. Workaround: At the end of VDBox/VEBox batch buffers which involve access to media compressed buffers, SW must insert an extra MI_FLUSH_DW command and specify an address that is different from the compressed allocation (can be compressed or uncompressed).	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
data_corruption	14011348488	DAPRSS Clamping NaN Inconsistently	Errata: If Pre-Blend Source Only Clamp is enabled and Clamp Range is set to COLORCLAMP_UNORM, hardware will not clamp FLOAT render targets to 0.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	1409364714	RCS/POCS/CCS/BCS: Reserved fields in "Instdone" Registers are tied to "0" instead of "1"	Software must ignore the Reserved Fields in the INSTDONE register.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	1809012548	MPEG2 & AVC Encode: As part of encode operation, CONDITIONAL_BATCH_BUFFER_END command fetches Compare data (related to Panic mode/QP) and pushes to hw engine for subsequent frame; sends wrong data if the comparison data was in upper 4 QWORD of cacheline	Software must ensure the “Compare Address” programmed in MI_CONDITIONAL_BATCH_BUFFER_END command for the Compare Data Qword in memory is always within the first 256b of a cacheline (i.e address bit[5] must be ‘0’). MI_STORE_REGISTER_MEM have the same address condition for register image status mask (08b4)and image status data(08b8) so that the mem write is always within the first 256b of a cacheline	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	14011403231	Coarse Pixel Shading: Hang can occur with CPS Aware color pipe optimization enabled: CPQ sequence sent with no state in case where SubspanValid=true but SubspanValid=false	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang,security	22010271021	3DState programming on RCS while in PIPELINE_SELECT= GPGPU mode can cause system hang due to FFDOP clock gating	Kernel driver should disable FF DOP clk gating via masked write to 20EC[1] = 1.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
data_corruption	22010614185	Corruption may occur with the surface formats B5G5R5X1_UNORM and B5G5R5X1_UNORM_SRGB if Color Blend is enabled	Errata: Corruption may occur with the surface formats B5G5R5X1_UNORM and B5G5R5X1_UNORM_SRGB if Color Blend is enabled.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	1607666336	Semaphore_signal with post sync enable does not send the correct signal data	Due to known HW issue, SW must not set “Post-Sync Operation” field for MI_SEMAPHORE_SIGNAL command	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	16011107343	Hull Shader Control and Header Fifo in TRG going out of sync results in hang	Please insert 3D State HS before every 3D primitive that has HS enabled	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14011466639	PLANE_CC_VAL not getting updated immediately on async flip	Display async flips will not update the clear color value at the right point. Limit use of display clear color to sync flips.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	14011508470	Remove PM Req with unblock/memup + fill support -- SAGV enhancement not working as expected	Set register bits (offset 0x46434) bits 24,25,26,27) to 1 at display initialization to revert the RTL fix.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	22010554215	Incorrect decoding of DW67 in MFX_PIPE_BUF_ADDR state.	Slice size streamout buffer address should be programmed as zero and disable Slice size streamout feature (Slice Stats Streamout Enable in MFX_AVC_IMG state to zero) till the HW bug is fixed. SW has two methods of generating slice size for the frame. Method 1: At the end of each slice, read MFC Bitstream Byte Count register and store in SLICE_SIZE_BUFFER, increment SLICE_SIZE_BUFFER address by 4 bytes. Method 2: SW can parse the bitstream and determine each individual slice size.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	16011163337	HW default value for fusedEU timeout for thread dispatch can hang	The GS Timer Bits [31:24] in the GangTimer Register [MMIO: 0x6604] should be set to 0xE0 (224 decimal	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
		HS / DS		
hang	22010772959	With pixel scoreboard disabled, PSS is creating an extra thread with no slotquads loaded when it sees an FC64 8x8 with a different topology have an overlapping X/Y with two already committed partial threads	Issue: When pixel scoreboard is disabled, PSS is creating an extra thread with no slotquads loaded when it sees an FC64 8x8 with a different topology. This is causing a hang. WA: When SIMD32 is enabled, do not disable pixel scoreboard. In other words, 3DSTATE_PS Bitgroup5[21] = 0 when 3DSTATE_PS Bitgroup5[2] = 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	14011555645	SVSM: Dual Context - Invalidate hang	SW must ensure pipeline is IDLE prior HW or SW executing a state cache invalidation. There are two possible cases SW or HW may cause this to happen: 1) Scheduler must ensure that CCS and RCS are not running in parallel. CCS could invalidate the state cache while RCS is executing and visa-versa. 2) SW must insert a PIPE_CONTROL with CS stall prior to any PIPE_CONTROL with "State Cache Invalidate Enable" bit. Any PIPE_CONTROL with "State Cache Invalidate Enable" bit set will do an invalidation of the state cache prior to flushing the pipe while sampler is active.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
performance	1608975824	HDC issues an uncacheable 'clear' color read when compression is enabled, using MOCS#0 instead of MOCS#3	No w/a is needed for functionality. For performance w/a: KMD should set MOCS[0] as "L3 cacheable". Mocs[0] is usually reserved.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	16010657660	CS power context save/restore doesn't work properly for 0x20E4[2:1]	Driver must program register 20e4[2:1] - with required preemption granularity along with the corresponding mask bits as part of WABB during every context restore.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	14010919138	Hang can occur on VS UAV write when TE-DOP clk gating is enabled	Set Tessellation DOP Gating Disable via bit [19] in the ThreadMode Register [0x020A0]. eg: 0x020A0[19]=0x1	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
hang	2209620131	HWM unit doesn't check for ack response from downstream unit (backpressure) on tile boundaries, results in hang	Real Tile Scale Decoder insert below commands after every HCP_BSD_OBJECT: (Tile boundary) MFX_WAIT (with MFX_Sync_Control_Flag=1) VD_PIPELINE_FLUSH (with HEVC flush + VDcmd flush + HEVC done=1)	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	22010923548	PSD is indicating the first payload phase as null for PSD_REG_P_BARY_PLANE phase	Corruption can exist in Fused SIMD16 threads if R68-R71 is the first phase after R1. This scenario might happen if experimenting with "remove BC" kernel. Enable any phase from R3-R67 to prevent the issue.	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	14011807268	During Object-Level preemption and an odd number of objects VF does not change the Topology correctly in the Ctx Restore	Multiple WAs are proposed for this issue. Details of them are captured below in "workaround_details section". For now, we are continuing with following WA: Disable Mid-Object Preemption for certain topologies. Set 0x2580[0] = 0 or 0x20ec[0]. It is derived from the condition in RTL - object_preempt_en = 0x20e0[14] ? 0x2580[0] : 0x20ec[0]; - 1. TRISTRIP (0x05) - 2. TRISTRIP_REVERSE (0x0c) - 3. LINESTRIP (0x03) - 4. LINESTRIP_ADJ (0x0A) - 5. LINELOOP (0x10) - 6. LINESTRIP_BF (0x11)	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	18011464164	dx10_sdksamples_sc-default-effect-pools-msaa-2_win-skl_main - triangular corruptions	Set Tessellation DOP Gating Disable via bit [19] in the ThreadMode Register [0x020A0]. eg: 0x020A0[19]=0x1	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
	14010733141	HCP + SFC reset doesn't work correctly	This bug would affect the VDBOX-SFC reset sequence. We need to VE-SFC forced lock to get around this issue. Here are the steps 1. .Check MFX-SFC usage 2. If (MFX-SFC usage is 1) { a. Issue a MFX-SFC forced lock b. Wait for MFX-SFC forced lock ack c. Check the MFX-SFC usage bit d. If (MFX-SFC usage bit is 1) Reset VDBOX and SFC else Reset VDBOX f. Release the force lock MFX-SFC } 3. else (check HCP-SFC usage). 4. if(HCP+SFC usage is 1) 1. Issue a VE-SFC forced lock 2. Wait for SFC forced lock ack 3. Check the VE-SFC usage bit 4. If (VE-SFC usage bit is 1)	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
			Reset VDBOX else Reset VDBOX and SFC 5. Release the force lock VE-SFC. else Reset VDBOX	
other	22010537631	PSD RTL bug caught through UVM: disp_reg_addr going to X (PSD_REG_ERR) instead of R67 phase	Corruption can exist in dual-simd8 threads if R66-R71 is the first phase after R1. This scenario might happen if experimenting with "remove BC" kernel. Enable any phase from R3-R65 to prevent the issue.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	16011478345	SVG RTL doesn't correctly handle Push Constant buffer with length 0 when buffer address bit 5 is set; Results in render hang	When using Push Constant Buffers, anytime the buffer length is 0x0, the address of the buffer needs to be set to 0x0.	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	22011054531	Corruption on 3D engine writes to media compressible render target due to incorrect memory cycle type used for read operations when RHWO optimization is enabled	Issue: Checkerboard background on text input and composition rendering across multiple apps WA: 0x7010[14] needs to be set for all media compressed render targets	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	22010493002	Media compression: Decode output writes sometimes sends data as uncompressed but doesn't properly update tile compression status to match, resulting in corruption when data is consumed later	Compression Control Surface should be cleared for destination buffers at the start of the batch buffer for AVC/VC1/JPEG/MPEG2 codecs.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	1508208842	AV1 decode corruption on due to non-deterministic state on exit from reset/power gating	For every AV1 batch buffer, do a force reset/flush on the AV1 pipeline prior to running an Inter workload	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
other	14010826681	Driver writes to SVL register offsets	Disable FF DOP clk gating when accessing registers in SVL unit (range 0x7000-0x7FFC). This	sku; stepping_impacted; stepping_fixed; wa_status

impact	lineage	title	bspec_wa_details	sku_impact
		sometimes don't work correctly due to FFDOP clk gating	could be done: A. EITHER on a per access basis - save current 20EC[1] polarity, masked write 20EC[1]=1 to disable, write SVL register, masked write to 20EC[1] to restore original polarity. B. OR statically disable FFDOP clk gating all the time via 20EC[1]=1 or 9424[2]=0 from driver boot.	ALL | a0 | | driver_permanent_wa
hang	22011327657	CSB data in hw status page may be stale when read out by SW (memory ordering for CS write vs engine interrupt delivery)	Issue:: HW ensures the write completion are in place before generating the interrupt, however because of latencies involved when SW is processing a interrupt it might end up processing the following CSB update in HWSP for which HW has not yet generated the context switch interrupt. The race between the write pointer update and CSB update for a given context switch itself shows up as the writes to memory from a given client need not be ordered. This issue might show up when fences were removed, prior all these writes were fences and hence were ordered. Workaround: SW on processing an CSB interrupt requiring to process more than one CSB entry, SW must introduce a delay of 30us between CSB fetch and processing. OR SW must process on chip CSB present in CS through MMIO reads.	sku | stepping_impacted | stepping_fixed | wa_status ALL | a0 | | driver_permanent_wa
data_corruption	22011217531	Display underrun can occur on cursor plane if WM0 is used without WM1		sku | stepping_impacted | stepping_fixed | wa_status ALL | a0 | | driver_permanent_wa
hang	22010594632	Hang can occur at context switch with RCS+CCS concurrency due to bug in const cache invalidate @ context switch	Disable Dual Queue(no async computes) and avoid RCS+CCS usage concurrently.	sku | stepping_impacted | stepping_fixed | wa_status ALL | a0 | | driver_permanent_wa
	16011227922	Range based flush does not support 48 bit addressing	Disable 48 bit addressing for all stepping and limit addressing to 47 bits irrespective of dual context enabled or not.	sku | stepping_impacted | stepping_fixed | wa_status ALL | a0 | | driver_permanent_wa

impact	lineage	title	bspec_wa_details	sku_impact
hang	16011448509	3DSTATE_CONSTANT_ALL command not processed correctly in certain cases	W/A for the S/W is to use 3DSTATE_CONST command for individual shader instead of 3DSTATE_CONST_ALL COMMAND.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	14010915987	DPST IET data writes do not trigger PSR exit due to lack of write event indication to DMUX	Since the reason for image enhancement not being applied is that the writes to the DPST registers are not causing an exit from PSR. The WA is to do a R/W to PAL_LGC_A_0: 0x4A000 after the Image Enhancement update. This will trigger a PSR exit immediately. This register was chosen because it is not DB buffered.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	22010465259	Coarse Pixel Shading: DAPRSS incorrectly sending CPQ with No Pixels Lit, can causing hang/incorrect rendering when CPS Aware color pipe optimization enabled	Disable CPS Aware color pipe by setting MMIO register. 0x07304 Bit[9] to 1	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	22010493298	HiZ corruption issue related to RCZ/L3 interaction when data is in clear state in CC	WA: Set (7018h) bit 14	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	22010931296	Clock gating issue results in rendering corruption	Set Tessellation DOP Gating Disable via bit [19] in the ThreadMode Register [0x020A0]. eg: 0x020A0[19]=0x1	sku: ALL; stepping_impacted: b0; stepping_fixed: ; wa_status: driver_permanent_wa
	22010751166	Underrun can occur in certain cases when FBC is enabled	For non-modulo 4 plane size(including plane size + yoffset), disable FBC when scanline is Vactive - 10	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa