Intel® Iris® Xe and UHD Graphics Open Source

Programmer's Reference Manual

For the 2020-2021 11th Generation Intel Xeon®, Core™, Celeron®, Pentium® Gold Processors based on the "Tiger Lake" Platform

Volume 14: Workarounds

December 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, with the sole exceptions that a) you may publish an unmodified copy and b) code included in this document is licensed subject to Zero-Clause BSD open source license (0BSD). You may create software implementations based on this document and in compliance with the foregoing that are intended to execute on the Intel product(s) referenced in this document. No rights are granted to create modifications or derivatives of 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

H81 Workarounds

impact	title	bspec_wa_details	sku_impact
	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; wa_status: driver_permanent_wa
data_corruption	HDC L3 write moves forward for a L1 cacheable write when Sampler is stalling and can result in RAW hazard	DW-1 Bit-13 of State Compute Mode register (field name: Disable L1 Invalidate for non-L1-cacheable Writes) must be set to 0 by driver.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
hang	GFXPERF: Shadow of Mordor: frame 253 hang in WW40d model	Set register bit (7018h) bit 13 = 1 when depth buffer is D16_UNORM	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	PSS X-prop issue in quad_valid when we see an unlit poly on the back of a chg marker with no SIMD modes enabled by the programmer	It is unknown if detected X-prop issue can generate Si failures. To avoid any possible issues, set at least one simd enable in 3dstate_ps (e.g. 16 pixel dispatch enable). If no pixel shader is valid, clear 3dstate_ps_extra "pixel shader valid"	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
	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; wa_status: driver_permanent_wa
hang	Coarse Pixel Shading - hang can occur in color pipe if CPS Aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting register bit: 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	Coarse Pixel Shading - perf issue with floating point render targets if CPS Aware color pipe optimization is enabled	Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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 register bit 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Coarse Pixel Shading - data corruption can occur if CPS Aware color pipe performance optimization enabled	Disable CPS Aware color pipe by setting register bit: 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption, hang	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 register bit: 0x07304 Bit[9]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	DPT should send VRR enable indicator to DCPR even while Push mode is enabled.	Package C2 increase when VRR is enabled with push mode. When enabling VRR, before setting TRANS_VRR_CTL VRR Enable, program GT-driver Pcode mailbox with	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		command 0x11 and data low bit 0 = 1 to inform pcode that VRR is enabled. When disabling VRR, after clearing TRANS_VRR_CTL VRR Enable, program GT-driver Pcode mailbox with command 0x11 and data low bit 0 = 0 to inform pcode that VRR is disabled.	
data_corruption	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; wa_status: driver_permanent_wa
data_corruption	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 register bit: 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
performance	Sampler cache can be thrashed in certain cases involving texture arrays resulting in low performance	added a programming note to the Render Surface State BXML saying the Array bit should not be set unless the depth of the arrayed surface is > 1.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	MI_ATOMIC uses wrong address for atomic operation in RCS.	MI_ATOMIC command when programmed with “Inline Data” field set to “0” must have “Dword Length” field of the command set to “9h” and must have Dword3..10 programmed with data as 0x0.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	WGF11RenderTargets failure	Issue: CPQ can be optimally pipelined from DAPRSS to the color pipe in two phases, one for fill and one for blend instead of breaking down the blend CPs into PQs. Due to bug in DAPRSS, when using R11G11B10_FLOAT format, looks like RTL uses blend data instead of fill data during fill phase. WA: Disable CPS aware color pipe by programming register bit for Common Slice Register3 (0x7304) bit 9 to 1.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
hang	Register based invalidations for a given engine don't indicate completion if that engine is in a power domain that is powered down	SW need to always send an OA invalidation following any render /compute or media TLB register based invalidation. The sequence from driver/SW should be: (when issuing any register based invalidation) 1) issue a mmio write to any render/compute/media Inval 2) issue a mmio write to OA Inval.register (0xCEEC) 3) Now poll for respective invalidation completion	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	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.	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption, hang	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 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	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td>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>performance</td><td>[non-RCS] Preempt delay counter should not be reset on heqt restore abort</td><td>Scheduler must check the Context ID on ACTIVE to IDLE switch to make sure which element was preempted even if it is not the last element of the prior submission.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>hang</td><td>[nonRCS] CS should stop making new DMA req once decided to go to RDOP</td><td>Disable RDOP on Semaphore Wait and Wait for event using register bit. OR Disable Pre-Parser around MI_SEMAPHORE_WAIT and MI_WAIT_FOR_EVENT command using MI_ARB_ON_OFF.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td></td><td>Plane with Source keying enabled on format "P010" not going transparent based on color channel selection</td><td>Source keying with source planes in the pixel formats “P010”, “P012”, “P016”, “RGB64 Unit” is not supported;</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>hang</td><td>RCS is not waking up fixed function clock when specific 3d related bits are programmed in pipecontrol in compute mode</td><td>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</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>power</td><td>RCU should ignore(reset) Media Sampler DOP status of engine which is idle</td><td>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</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
		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. OR • Media Sampler Workload got preempted before PIPELINE_SELECT with Media Sampler DOP Clock Gate Enable is executed. SW may avoid the inefficient Media Sampler DOP Clock Gate Enable by avoiding above mentioned scenarios, i.e • Make workloads accessing Media Sampler non-preemptable and ensure they are bracketed between PIPELINE_SELECT with Media Sampler DOP Clock Gate Disable and Enable respectively. Or • Following a context switch status of Active to Idle for a Media Sampler workload from and engine and while other engine is busy, SW must submit a context (dummy no real workload) to the former to reset the Media Sampler DOP Clock Gate to be Enabled.	
hang	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: b0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	Semaphore_signal with post sync enable does not send the correct signal data to GUC	Due to known HW issue, SW must not set “Post-Sync Operation” field for MI_SEMAPHORE_SIGNAL command	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	Display software needs to configure SSC enable in a new PLL register	DPLL SSC enable is not correctly hooked up to DPLL_CFGCR0 SSC Enable field. WA: Use DPLL_SSC sscen field to enable SSC instead of DPLL_CFGCR0 SSC enable field.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption, hang	while loop cases causing issues in jeu fused mask	Disable Structured Control Flow by setting EnableVISAStructurizer.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	HDC: HDCTLB tdl_mode bits incorrectly decoded for hdctlbl3arb	DW-1 Bit-13 and Bit-12 of State Compute Mode register (bitfield names: Coherent access L1 Cache Disable, Disable L1 Invalidate for non-L1-cacheable Writes) must be set to 0 by driver. Coherent access L1 cacheability can be still controlled by MOCS value.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
performance	AMFS : Multi Eval perf test is having traffic only on 3 TSL ports instead of all 6 ports during the 2nd half of the run	0x7300[6] should be set to 1	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption, other	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 register bit: 0x07304 Bit[9]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
	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 planes	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
security	Accumulator is not currently cleared with GRF clear exposing its content to new context.	Clear ACC register before EOT send mov(16) acc0.0:f 0x0:f	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	3DMARK_NIGHTRAID_DX12 - CS not done on PIPE_CONTROL	WA/Mitigation: · Register bit to disable PC deref enhancement 0xe4f4[8]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	Register reads to 0x6604 is incorrect	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.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Multicontext preemption tests hang with sampler, sc & hdc not done	Disable GSYNC.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
performance	LNCF MOCS settings are cleared on soft reset of RCS/POCS/CCS	Upon render reset, the driver needs to reprogram LNCFCMOCS0 to LNCFCMOCS31. Programming note: WAReprogramMOCS: Upon render reset the driver needs to reprogram the LNCF MOCS Register.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
power	AMFS Evaluate via Compute CS hangs if FFDOP clk gating is enabled	1. if compute shaders do evaluate, SW must program register 0x20ec[1] to 1 2. Shaders must not do Evaluate, in VF (Virtual Function) mode.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		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)	

impact	title	bspec_wa_details	sku_impact
data_corruption	3DMark IceStorm/IceStormExtreme Demo - corruptions	To avoid sporadic corruptions “Set 0x7010[9] when Depth Buffer Surface Format is D16_UNORM , surface type is not NULL & 1X_MSAA”	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	pipe3D: register bit flexing : TDL is blocking deref when 8th bit of tdl chicken	Issue: When the push constant deref pipelining is disabled, it can result in some performance drop (0.5% to 1% for certain workloads). WA: Disable Push constant Buffer E48C[9]=1	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
data_corruption	TRTT Aliased Buffers Data Mismatch - Possible race condition between Mem Wr and HDC Flush	A "HDC fence" message must be inserted before the EoT of a compute, 3D or a pixel shader thread, if there is any HDC memory write requests from the thread. [L3 cache flush from the fence message is NOT needed].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	CONDDBG indx matching is for same register only	Issue: If a data pattern is detected in the thread dispatch data, the conditional debug feature should set breakpoint. The conditional debug feature can catch it on multiple data patterns, but it will only match if those data patterns are on the same data phase (which corresponds to a grf register). WA: CONDDBG indx matching is for same register only.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	MI_SET_PREDICATE with # of slices does not work in the condition that the slices are not contiguious	As part of the Dual Context feature the predication evaluation based on the number of slices is deprecated from MI_SET_PREDICATE. This functionality was deprecated previously.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
hang	VFURB dropping data in some scenarios involving 256 bit element format	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; wa_status: driver_permanent_wa
other	Input Coverage = INNER is incorrectly ANDing sample masks	Issue: While designing CPS and depth coverage mode for input coverage for conservative rasterization, implementation changed. This was noticed especially as input coverage mode = INNER started ANDing sample mask to conservative rasterization mask. This resulted in a mis-match write to the spec. WA: Have PS	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td>compiler logically OR input coverage mask to infer if a pixel is fully covered when INPUT_COVERAGE_MASK_MODE = INNER</td><td></td></tr>
<tr><td>data_corruption</td><td>HDC RTL does not support 16-bit typed atomics</td><td>SW must not generate 16bit Typed atomic messages. Also, there must not be API support for 16bit Typed Atomics</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>Blitter RAW hazard between blits</td><td>"For two sequential fast copy blits when the source of the second blit is the destination of the first blit or they overlap a Flush must be inserted between the two blits (there can be one or more Fast Color blt between those two fast copy blits)."</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>hang</td><td>EU instructions: Indirect address access with Acc destination doesn't work correctly on fused EU pair</td><td>WA: Shader compiler should not generate EU instruction that has both indirect addressing and Acc destination. Indirect addressing can be used with non-Acc destinations; Acc destination can be used in cases other than indirect addressing.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>other</td><td>Atomic operation does not work on compressed data</td><td>Driver must make sure there is not atomic operation done on compressed data. For DX API, this means compression will be disabled for any SINT/UINT surfaces. For OCL, compression is allowed on untyped surfaces. But it is the responsibility of the driver to check kernels for any atomic operations, and resolve the surfaces that could be accessed by atomic, before the kernel launch.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
data_corruption	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; wa_status: driver_permanent_wa
hang	EU hang can occur if regular send instructions are followed by URB atomic	WaName: WaResolveDepBeforeAtomics When multiple sends to low priority bus (obus) are present before an Atomic chain of sends to high priority bus (sbus), MA switches grants to high priority bus after the first low priority grant and never goes back to grant the remaining low priority requests. WA: If Atomic chain ends with EOT then resolve all SBID dependencies before the Atomic chain of instructions (sync.allrd), else if Atomic chain does not end with EOT then resolve all SBID dependencies present within the Atomic chain before starting the chain.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
other	HW default polarity for Sampler Small PL is "disabled" - not optimal for power	Issue: To ensure optimal power in 3D Sampler. WA:Enable bit 15 of E18C.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	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; wa_status: driver_permanent_wa
data_corruption	Incorrect blue channel value when sampling from R32G32_FLOAT surface with border texture addressing mode	Issue: When sampling from an R32G32_FLOAT surface with border texture addressing mode, there is an issue where the blue channel value is missing. WA: Set the shader channel select to 1.0 (instead of 0) for the missing blue channel.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	3D Tiled-YF surface corruption in MIP tail LODs because of X-adjacent RCC cacheline composition	WaSetMipTailStartLODLargertoSurface LOD 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; wa_status: driver_permanent_wa
performance	Perf: 3DMark11 Shadowmap : TDS dual dispatch issue	mmio offset 6604h bits 23:16 must be set to 4h	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	HS Hang & TDG mismatches when dual_instance_enable is zero AND HS is handle limited.	Restricting the min number of input handles to 256+128 (?) and output handles to 8 when instancing is enabled	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
data_corruption	3D: AMFS: pipe3d : Same virtual address is being sent/used by amfs for different LODs, resulting in dropping of some LODs resulting in corruption	BitField: Procedural Texture 11: Name: Procedural Texture Description This bit, when set, indicates that the associated surface is a procedural texture which is used for AMFS. This bit can be ENABLED for the following surface types: SURFTYPE_2D arrayed / non-arrayed, SURFTYPE_3D non-arrayed, SURFTYPE_CUBE arrayed/ non arrayed, and surftype = NULL. This bit can be set for the pixel formats that are supported has typed UAVs as per the DX spec. Therefore, writes from only HDC are supported to Procedural Textures. This bit cannot be ENABLED for the following surface types: SURFTYPE_3D arrayed, SURFTYPE_BUFFER Description This bit cannot be ENABLED for SURFTYPE_SCRATCH. ProgrammingNote This bit cannot be set when surface walk (tiling mode) is legacy Y This bit cannot be set when Tiled Resource Mode = TileYS and LOD >= MIP tail LOD	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	Cs - Gam Deadlock after Root Entry Not Present Fault	WA Name: NoResetReadynessHandShake SW must not do Reset Readiness Handshake as part of the reset recovery on an CAT error.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] Color: DaprSsDaprSc.ss_phase0.cpq_mask.sample_mask Mismatch	Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] DAPRSS Sending Blend CData Encoding For Fill CPQ	Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	PSDunit is dropping MSB of the blend state pointer from SD FIFO	Limit the Blend State Pointer to < 2G	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[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: b0; wa_status: driver_temporary_wa
data_corruption	[GT1] [DAPRSS] Data Corruption on R10G10B10_FLOAT_A2_UNORM After Blend2Fill	See the Errata on Pre-Blend Color Clamping	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[GT1] [DAPRSS] Repcol with R10G10B10_FLOAT_A2_UNORM Not Properly Down-converted	De-feature Repcol Messages	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td>PCH display clock remains active when it shouldn't; impact to power and sleep state residency</td><td>Display driver should set and clear register offset 0xC2000 bit #7 as last step in programming south display registers in preparation for entering S0ix state, or set 0xC2000 bit #7 on S0ix entry and clear it on S0ix exit.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>[AMFS] SW Workarounds for AMFS flush</td><td>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</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
hang	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; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
data_corruption	Page Faults: Write access to page marked as read only results in write being dropped, but fault may not be reported	Errata: WR permission faults may not be reported for write access to Read Only pages. SW can choose not to use read only pages OR just live with the fact that write accesses can be silently dropped without permission fault reporting.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Hangs can occur if using constant cache invalidate command with RCS+CCS concurrency	If the intention of “constant cache invalidate” is to invalidate the L1 cache (which can cache constants), use “HDC pipeline flush” instead of Constant Cache invalidate command. Some units bypass the L3 cache when they access memory - CS, MediaFF and Guc. When data sharing (e.g. semaphore) between these units and a shader is needed, the L3 cache may need to be	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td>invalidated using a pipe-control CS command with a "Const cache Invalidate" set. In these cases, the w/a should be to set the "State $ invalidate" in the pipecontrol command, in addition to the "HDC pipeline flush". Setting "state $ invalidate" will also invalidate the RO section (including constants) of L3 cache. So, WA should be applied to GT1 all steppings.</td><td></td></tr>
<tr><td>other</td><td>Vertex fetch unit can fetch past end of vertex buffer resulting in page faults</td><td>Add one extra page to the vertex buffer when in sequential mode.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>*CS: sometimes ctx time stamp register doesn't get restored to value from the engine context image on context switch</td><td>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.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td>MI_LOAD_REGISTER_MEM (dw0[19] = 1, dw0[21]= 1, REGISTER ADDR = 3a8h, Memory Address = DEADh + 108Ch 2. 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</td><td></td></tr>
<tr><td></td><td>dupunit not generating line_pop indication for plane with minimum size</td><td>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</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption, performance</td><td>Sampler power context is not saved/restored</td><td>Issue: Sampler Power Context save operation doesn't work correctly. This means that if driver writes to any of the following offsets (E100, E180, E184, E188, E18C, E190, E194) with non-default values will not be persisted across Render power gating/RC6. There are some cases we already know of where driver is expected/required to write non-default values for correct functional operation and best performance: All E18C[0] E18C[15] In addition to these, more cases may be identified later where driver wants/needs to program these registers with non-default values and needs to have that programming be restored after render/RC6 power gating. Workaround: KMD to configure RC6 WA BB for RCS (CTX_WA_PTR) if not already enabled; allocate buffer to contain the commands and ensure it is pinned in GGTT. In the RC6 WA BB, include LRI command that writes to any offsets which require non-default values. More specifically, if KMD programs any of the 7 offsets identified above during driver boot and/or after engine reset, those same offset/value pairs must also include that offset/value in an LRI command in the RC6 WA BB for RCS. Note that all 7 of these offsets are masked registers (upper 16b mask; lower 16b value)- driver only needs to enable the mask bits</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td></td><td></td><td>for the specific bits it wants to program to non-default value (e.g. for the value for E18C could be 0x1001_1001 (e.g. mask bits 31 & 16 set to allow the values in bit 15 & 0 to take).</td><td></td></tr>
<tr><td>other</td><td>Multicontext: rsi message retrieves inst_Base address from RCS for both contexts</td><td>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</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>Disable DFR</td><td>SW must disable DFR, (permanent work around); that by setting DFRRATIOEN9550[9] -> 1)</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>DAPRSS Clamping NaN Inconsistently</td><td>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.</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>other</td><td>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</td><td>VCS WA: To sample image status from read return data during MI_CONDTIONAL_BATCH_BUFFER_END command The conditional Batch buffer End should have address such that it will always have expected data (image status data) in lower half cacheline . The MI_STORE_REGISTER_MEM have the same address condition for register 08b4 and 08b8 (image status data) so that the mem write is for lower half CL. EX. MI_STORE_REGISTER_MEM 0000_08b4 lw_address up_address MI_STORE_REGISTER_MEM 0000_08b8 lw_address+4 up_address MI_CONDTIONAL_BATCH_BUFFER_EN</td><td><table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
		D Data lw_address up_address	
hang	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 register bit. 0x07304 Bit[9]	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang,security	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; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
data_corruption	PLANE_CC_VAL not getting updated immediately on async flip	Display async flips will not update the clear color value at the right point. The potential workarounds: WA1: KMD must convert async flip to sync flip upon clear color change. WA2: UMD must do partial resolve upon color clear change before submitting the flip to Display, KMD keeps async as async flip.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Remove PM Req with unblock/memup + fill support -- SAGV enhancement not working as expected	SAGV fill timeout. Set 0x46434 bits 24 ,25, 26, and 27 to 1 at display initialization .	sku: ALL; stepping_impacted: b0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	Unexpected ResInfo results with LOD out of bounds.	When doing a resinfo message SW needs to check if any of the LOD values in an aligned 4 channel group is different: channel 0-3, 5-7, 8-11 , etc. If any of them are different It must sequence the resinfo message so that there is at most one unique LOD per valid channel in each 4 pixel group.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	HW default value for fusedEU timeout for thread dispatch can hang HS / DS	The GS Timer Bits [31:24] in the GangTimer Register [MMIO: 0x6604] should be set to 0xE0 (224 decimal)	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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	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; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
performance	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; wa_status: driver_permanent_wa
data_corruption	CS power context save/restore doesn't work properly for 0x20E4[2:1]	Driver must program FF_SLICE_CS_CHICKEN2 register 20e4[2:1] - with required preemption granularity along with the corresponding mask bits as part of WABB during every power context restore.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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: a0; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
data_corruption	During Object-Level preemption and an odd number of objects VF does no 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". Due to perf regression of disabling object-level preemption per topo, a blanket disable can be used instead. Disable Object Preemption Set 0x2580[0] = 0 or 0x20ec[0]. It is derived from the condition in RTL - object_preempt_en = 0x20e0[14] ? 0x2580[0] : 0x20ec[0];	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	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: a0; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
hang	SVG RTL doesn't correctly handle Push Constant buffer with length 0 when buffer address bit 5 is set; Results in render hang	Issue: SVG RTL not zeroing out address bit 5 when the Push Constant buffer length is 0. This is causing additional derefs to be generated. WA: Two options WA1 Program the Push constant buffer address in the Push constant command to be cacheline aligned i.e. make sure bit 5 of the address is set to 0, if any of the 4 push constant buffer length is programmed to be 0 for that constant buffer address. If the above WA is difficult to do, then please do this more generic WA WA2 Program the Push constant buffer address to be always cacheline aligned irrespective of buffer length i.e. make sure bit 5 of the address is set to 0 always in PC command programming.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Corruption in viewmask token coming into CL for POSH enabled workloads when TE DOP is disabled	Disable TEDOP Clock Gating with register bit 20A0 bit 19 set to 1 at boot + Disable POSH for draw calls with PRIM Replication OR PRIM ID enabled	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	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	0x7010[14] needs to be set for all media compressed render targets	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	3DSTATE_CONSTANT_ALL command not processed correctly in certain cases	1. The easiest W/A for the S/W is to use 3DSTATE_CONST command for individual shader instead of 3DSTATE_CONST_ALL COMMNAD. 2. To W/A this issue and to still use 3DSTATE_CONST_ALL command and not lose out on perf, we have to restrict the “Pointer to constant Buffer” filed to always have the address bits [12:8] as zero. Note this is just restricting the start address and CS can still prefetch CL as mentioned in size field. 3. If this address bits (Pointer to constant Buffer[12:8]) needs to be used, then only for those address range we can switch to shader specific push constant commands and rest address can still use 3DSTATE_CONST_ALL.	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
other	Driver writes to SVL register offsets sometimes don't work correctly due to FFDOP clk gating	Disable FF DOP clk gating when accessing registers in SVL unit (range 0x7000-0x7FFC). This could be done: 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. OR statically disable FFDOP clk gating all the time via 20EC[1]=1 or 9424[2]=0 from driver boot. FFDOP is already being	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
		required to be applied all the time as security workaround for another issue (hard hang if non-priv BB sends 3D STATE command while pipeline_select is in GPGPU mode). As such, the simpler static w/a (option B, specifically 20EC[1] version) is preferred for simplicity/consistency.	
	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: c0; wa_status: driver_permanent_wa
hang	CSB data in hw status page may be stale when read out by SW (memory ordering for CS write vs engine interrupt delivery)	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: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Display underrun can occur on cursor plane if WM0 is used without WM1	Bug in the register unit which results in WM1 register used when only WM0 is enabled on cursor. A similar bug was fixed in the planes in 11p5, but Cursor was missed. Software workaround is when only WM0 enabled on cursor, copy contents of CUR_WM_0[30:0] (exclude the enable bit) into CUR_WM_1[30:0]	sku: ALL; stepping_impacted: b0; wa_status: driver_permanent_wa
hang	PSS flush done does not comprehend PSD state change, it only comprehends all PS threads completed.	WA: Insert a csstall after every 10 draws. Performance impact of this w/a on select DX9 workloads has been found to be negligible.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	PCH display HPD IRQ is not detected with default filter value	WA: Program 0xC7204 (PP_CONTROL) bit #0 to '1' to enable workaround and clear to disable it. Driver shall enable this WA when external display is connected and remove WA when display is unplugged or before going into sleep to allow CS entry. Driver shall not enable WA when eDP is connected.	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
hang	Battlefield 4 + AA causing hang in MTunit	Fast Clear must not be used on 8- nor 16-bit-per-sample, MSAA color surfaces (e.g. B5G6R5, R8G8, R16, R8, etc. MSFMT_MSS surfaces), unless the following is true . . . 1. Surface Format is R8G8_UNORM, R16_UNORM, or R16_FLOAT. 2. Surface Width is a multiple of 8. 3. Surface Height is a multiple of 4. 4. Either . . . A. Surface is Tile64 (which is always the case for MSFMT_MSS on Tile64 platforms). B. Surface is TileYF or TileYS. C. Surface Horizontal Alignment is either HALIGN_8 or HALIGN_16. When implementing SW WA for this bug . . . If a surface meets 1+2+3 but not A/B, please also create the surface as HALIGN_8.	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
hang	Blank screen seen with 4 MST Displays	Issue: Blank screen seen with 4 MST Displays WA: If MST master is being enabled, clear DP VC Payload Bit before start of MST enable sequence and set is as part of regular MST enable sequence. If MST slave is being added to the MST primary transcoder, keep the VC Payload allocate bit of slave stream set throughout the MST slave enable sequence. Clear DP VC Payload Bit only for MST/DP2.0 case before Wait for ACT Sent Status Handshake during Disable Sequence Keep DP VC Payload Bit ON as part of HDMI/DVI Enable/Disable Sequence. When enabling non-MST cases (eDP/DP-SST/HDMI/DVI), MstTransportSelect in TRANS_DDI_FUNC_CTL must be programmed to match the assigned pipe.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	Depth stats (occlusion query) gives wrong results when using Render Target Independent Rasterization (STATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0) and no pixel shader bound	When 3DSTATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0, SW should program a dummy pixel shader in case occlusion query is required.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	3DMark - Firestrike - corruption in OOTB run	WA: Program maximum of 1536 handles for GS.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	Display junk and underrun on Pipe A while playing video using MTA with PSR2 enabled.	Issue: Display junk and underrun on Pipe A while playing video using MTA/ while launching edge browser/ opening folders/ interacting with windows icons and taskbar. WA: Set bit 0x46430[23]=0x1 whenever delayed Vblank is used.	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td>driver_permanent_wa</td></tr></table>
other	[MPEG2] Panic mode issue	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	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	LRR Cmd addr 2360 not correctly remapped	Software must use only MI_LOAD_REGISTER_IMM to program 0x2360 register	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
hang	OVR does not send init_abort to POCS when it runs out of free pages	WA: “ Only in the POSH pipeline, add N NOPs after 3DSTATE_PTBR_TILE_PASS_INFO with end of tile bit set (N = 5 * Num of PTBR Tiles programmed)“	<table><tr><th>sku</th><th>stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
data_corruption	Read data corruption due to delayed Writes with State Access --	WA: Driver can enable HDC L1 cacheability for "read-only" buffers only. Setting a "read-write" buffer as L1 cacheable can corrupt memory data. L1 cacheability is set by programming MOCS[6:1] = [48, 59] (in decimal).	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Hang due to deadlock created by RHWO scenario with RHWO optimization enabled.	WA: Disable RHWO by setting 0x7010[14] by default except during resolve pass.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
	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; wa_status: driver_permanent_wa
data_corruption	3D Surface Type Height\Width Restricted to 2047 in render_surface_state	WA: Max Height and Width of a 3D Surface Type is 2047.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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 register bit: 0x07304 Bit[9].	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
other	Display combo PHY DPLL and thunderbolt PLL fractional divider error	Display DPLL and TBT PLL fractional divider value is shifted when reference is 38.4 MHz, giving slightly incorrect frequencies. Workaround when reference is 38.4 MHz, divide by 2 the value programmed into registers DPLL*_CFGCR0 and TBTPLL_CFGCR0 field DCO Fraction. Example, original DCO Fraction value of 0x7000h must be divided to 0x3800h.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
performance	Edp panel will flicker when system idle at desktop with specific background picture	WA: The driver needs to program the FBC_STRIDE (0x43228) and enable the override stride once. The override stride should be programmed with : Compressed buffer seg stride (in CLs) = ceiling[(at least plane width in pixels * 4 * 4) / (64 * compression limit factor)] + 1 If the CFB size computed by: CFB size (in bytes) = Compressed buffer seg stride * Ceiling(MIN(FBC compressed vertical limit/4, plane vertical source size/4)) * 64, will not fit into the memory allocated to FBC, then driver will need to use a more aggressive compression limit factor.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
security	MI_FORCE_WAKEUP and engine reset happen at almost same time, then hang can occur	Prior to doing a reset, SW/FW must ensure command streamer is stopped. Setting both the ring stop and preparser enable bit in the below registers will cause the command streamer to halt. Note preparser is only enabled for RCS and CCS command streamers but bit exists in all CS's. MI_MODE set bit 8. GFX_MODE set bit 10.	sku: ALL; stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Panel Flicker after press F11 or Alt+Tab switch tasks under system.	Corruption seen when FBC is first enabled. After setting the FBC enable, wait for the next start of vblank, then write the plane 1A surface address register.	sku: ALL; stepping_impacted: b0; wa_status: driver_permanent_wa

UP3_UP4_H35 Workarounds

impact	title	bspec_wa_details	sku_impact
other	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; wa_status: driver_permanent_wa
hang	VFURB dropping data in some scenarios involving 256 bit element format	Issue: 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; wa_status: driver_permanent_wa
other	Input Coverage = INNER is incorrectly ANDing sample masks	Issue: While designing CPS and depth coverage mode for input coverage for conservative rasterization, implementation changed. This was noticed especially as input coverage mode = INNER started ANDing sample mask to conservative rasterization mask. This resulted in a mis-match write to the spec. WA: Have PS compiler logically OR input coverage mask to infer if a pixel is fully covered when INPUT_COVERAGE_MASK_MODE = INNER	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Blitter RAW hazard between blits	"For two sequential fast copy blits when the source of the second blit is the destination of the first blit or they overlap a Flush must be inserted between the two blits (there can be one or more Fast Color blt between those two fast copy blits)."	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	EU instructions: Indirect address access with Acc destination doesn't work correctly on fused EU pair	WA: Shader compiler should not generate EU instruction that has both indirect addressing and Acc destination. Indirect addressing can be used with non-Acc destinations; Acc destination can be used in cases other than indirect addressing.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	Sel Denorm Failure in Mixed Mode	WaDenormFlushWithRoundUp When half-float denormals are disabled (i.e. flushed to zero) and Rounding mode is set to "Rnd towards +Inf", output denormals gets flushed to zero during float to half-float conversion. WA: Compiler must not generate instructions with the above combination.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	EU hang can occur if regular send instructions are followed by URB atomic	WaName: WaResolveDepBeforeAtomics When multiple sends to low priority bus (obus) are present before an Atomic chain of sends to high priority bus (sbus), MA switches grants to high priority bus after the first low priority grant and never goes back to grant the remaining low priority requests. WA: If Atomic chain ends with EOT then resolve all SBID dependencies before the Atomic chain of instructions (sync.allrd), else if Atomic chain does not end with EOT then resolve all SBID dependencies present within the Atomic chain before starting the chain.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
	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; wa_status: driver_permanent_wa
other	HW default polarity for Sampler Small PL is "disabled" - not optimal for power	Issue: To ensure optimal power in 3D Sampler. WA:Enable bit 15 of E18C.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	Incorrect blue channel value when sampling from R32G32_FLOAT surface with border texture addressing mode	Issue: When sampling from an R32G32_FLOAT surface with border texture addressing mode, there is an issue where the blue channel value is missing. WA: Set the shader channel select to 1.0 (instead of 0) for the missing blue channel.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption,hang	While loop cases causing issues in jeu fused mask	Issue: One EU executes while loop sequence, other EU breaks out. However, due to NoMask after endif, both EUs end up executing mov and send. JEU Fused Mask not correct in HW. WA: Disable Structured Control Flow by setting EnableVISAStructurizer.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	HDC: HDCTLB tdl_mode bits incorrectly decoded for hdctlbl3arb	DW-1 Bit-13 and Bit-12 of State Compute Mode register (bitfield names: Coherent access L1 Cache Disable, Disable L1 Invalidate for non-L1-cacheable Writes) must be set to 0 by driver. Coherent access L1 cacheability can be still controlled by MOCS value.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	HDC L3 write moves forward for a L1 cacheable write when Sampler is stalling and can result in RAW hazard	DW-1 Bit-13 of State Compute Mode register (field name: Disable L1 Invalidate for non-L1-cacheable Writes) must be set to 0 by driver.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	PSS X-prop issue in quad_valid when we see an unlit poly on the back of a chg marker with no SIMD modes enabled by the programmer	It is unknown if detected X-prop issue can generate Si failures. To avoid any possible issues, set at least one simd enable in 3dstate_ps (e.g. 16 pixel dispatch enable). If no pixel shader is valid, clear 3dstate_ps_extra "pixel shader valid"	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
	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; wa_status: driver_permanent_wa
hang	Coarse Pixel Shading - hang can occur in color pipe if CPS Aware color pipe optimization is enabled	Issue: Hang can occur in color pipe if CPS Aware color pipe optimization is enabled. WA: Register bit for Common Slice Register3 (0x7304) bit 9 can be set to disable CPS Aware Color Pipe.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	Coarse Pixel Shading - perf issue with floating point render targets if CPS Aware color pipe optimization is enabled	Issue: In CPS enabled cases, some extra cycle in daprss to daprsc. WA:Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Coarse Pixel Shading - corruption can occur with R11G11B10_FLOAT render target if CPS Aware color pipe optimization is enabled	Issue: If CPs within CPQ have different blend enables, the CPQ can be optimally pipelined from DAPRSS to the color pipe in two phases, one for fill and one for blend instead of breaking down the blend CPs into PQs. WA: Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Coarse Pixel Shading - data corruption can occur if CPS Aware color pipe performance optimization enabled	Issue: If CPs within CPQ have different blend enables, the CPQ can be optimally pipelined from DAPRSS to the color pipe in two phases, one for fill and one for blend instead of breaking down the blend CPs into PQs. WA: Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	DPT should send VRR enable indicator to DCPR even while Push mode is enabled.	Package C2 increase when VRR is enabled with push mode. When enabling VRR, before setting TRANS_VRR_CTL VRR Enable, program GT-driver Pcode mailbox with command 0x11 and data low bit 0 = 1 to inform pcode that VRR is enabled. When disabling VRR, after clearing TRANS_VRR_CTL VRR Enable, program GT-driver Pcode mailbox with command 0x11 and data low bit 0 = 0 to inform pcode that VRR is disabled.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
data_corruption	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 register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
hang	Register based invalidations for a given engine don't indicate completion if that engine is in a power domain that is powered down	SW need to always send an OA invalidation following any render /compute or media TLB register based invalidation. The sequence from driver/SW should be: (when issuing any register based invalidation) 1) issue a mmio write to any render/compute/media Inval 2) issue a mmio write to OA Inval.register (0xCEEC) 3) Now poll for respective invalidation completion	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	MI_ATOMIC uses wrong address for atomic operation in RCS.	MI_ATOMIC command when programmed with “Inline Data” field set to “0” must have “Dword Length” field of the command set to “9h” and must have Dword3..10 programmed with data as 0x0.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
data_corruption	Blend HW incorrectly uses color clamp range in cases where blend=enabled, Preblend source only clamp= disabled	Driver should always program Color Clamp Range Based on Table in Pre-Blend Color Clamping.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
other	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 an kernel argument	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	WM dropping transactions on AMFS_TXT_PTR-only state change	AMFS 3-pass failure. If only AMFS State is programmed, it gets dropped. It will not be a problem if another state like WM state is programmed. Workaround: An additional pipe control with post-sync = store dword operation would be required when programming the AMFS_TXT_PTR state.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	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; wa_status: driver_permanent_wa
performance	Perf: 3DMark11 Shadowmap : TDS dual dispatch issue	mmio offset 6604h bits 23:16 must be set to 4h	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	HS Hang & TDG mismatches when dual_instance_enable is zero AND HS is handle limited.	Restricting the min number of input handles to 256+128 (?) and output handles to 8 when instancing is enabled	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	3D: AMFS: pipe3d : Same virtual address is being sent/used by amfs for different LODs, resulting in dropping of some LODs resulting in corruption	BitField: Procedural Texture 11: Name: Procedural Texture Description This bit, when set, indicates that the associated surface is a procedural texture which is used for AMFS. This bit can be ENABLED for the following surface types: SURFTYPE_2D arrayed / non-arrayed, SURFTYPE_3D non-arrayed, SURFTYPE_CUBE arrayed/ non arrayed, and surftype = NULL. This bit can be set for the pixel formats that are supported has typed UAVs as per the DX spec. Therefore, writes from only HDC are supported to Procedural Textures. This bit cannot be ENABLED for the following surface types: SURFTYPE_3D arrayed, SURFTYPE_BUFFER Description This bit cannot be ENABLED for SURFTYPE_SCRATCH. ProgrammingNote This bit cannot be set when surface walk (tiling mode) is legacy Y This bit cannot be set when Tiled Resource Mode = TileYS and LOD >= MIP tail LOD	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	Emulation: PTBR tests are hanging with WMFE and IZ	S.W Workaround There is a potential software workaround for the issue by doing these 2 steps 1) setting the force thread dispatch enable(bits 20:19) in the 3dstate_WM_body state to be set to Force_OFF (value of 1) along with the first WM_HZ_OP state cycle 2) The second WM_HZ_OP state which is required by programming sequencing to complete the HZ_OP operation can reprogram the 3dstate_WM_body to set to NORMAL(value of 0).	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
	Plane with Souce Window keying enabled on format "P010" not going transparent based on color channel selection	Source keying with source planes in the pixel formats “P010”, “P012”, “P016”, “RGB64 Unit” is not supported;	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
performance	AMFS : Multi Eval perf test is having traffic only on 3 TSL ports instead of all 6 ports during the 2nd half of the run	Issue: AMFS not sending TS EOT to TDC causing it not properly load balance and utilize idle EUs in the system. Multi Eval perf test is having traffic only on 3 TSL ports instead of all 6 ports during the 2nd half of the run. WA: 0x7300[6] should be set to 1.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	Test CoarsePixelShading_817250 failing with FATAL_ERROR at GT due to X-propagation from RCC unit	Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td>power</td><td>RCU should ignore(reset) Media Sampler DOP status of engine which is idle</td><td>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. OR • Media Sampler Workload got preempted before PIPELINE_SELECT with Media Sampler DOP Clock Gate Enable is executed. SW may avoid the inefficient Media Sampler DOP Clock Gate Enable by avoiding above mentioned scenarios, i.e • Make workloads accessing Media Sampler non-preemptable and ensure they are bracketed between PIPELINE_SELECT with Media Sampler DOP Clock Gate Disable and Enable respectively. Or • Following a context switch status of Active to Idle for a Media Sampler workload from and engine and while other engine is busy, SW must submit a context (dummy no real workload) to the former to reset the Media Sampler DOP Clock Gate to be Enabled.</td><td><table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>data_corruption</td><td>GRF source swap feature for SIMD16 with Src0 scalar and bundle conflict between Src1/Src2 is causing the GRF read issue.</td><td>WA: Driver must set E4F4[14]=1 to disable early read/Src Swap.</td><td><table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td>hang</td><td>RCS is not waking up fixed function clock when specific 3d related bits are programmed in pipecontrol in compute mode</td><td>SW WA to program PIPE_CONTROl with RT Flush and CS Stall prior to PIPE_SELECT to Compute.</td><td><table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
data_corruption	Vs-CL Edge Flag mismatch - revert fix	Disable component packing when edgeflag is enabled.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	*CS does active to idle transition in certain timing cases with pending lite restore and subsequent preempt with other context AND HW preemption delay enabled	Recommendation 1: Scheduler when detects a pending pre-emption and receives Active2Idle should make sure which elements are pending. Scheduler must check the Context ID on ACTIVE to IDLE switch to make sure which element was preempted even if it is not the last element of the prior submission. Recommendation 2: Pre-emption Delay as part of the SW Scheduler instead enabled in HW.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	B0: Multicontext preemption tests hang with sampler, sc & hdc not done	Disable GSYNC.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr>
<td>data_corruption,hang</td>
<td>Certain Non-Pipelined State commands on RCS should work in PipeSelect compute, but don't because of FFDOP clk gating</td>
<td>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 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>
<table>
<tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr>
<tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr>
</table>
</td>
</tr>
</table>

impact	title	bspec_wa_details	sku_impact
data_corruption	[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; wa_status: driver_permanent_wa
data_corruption	LRR Cmd addr 2360 not correctly remapped	Software must use only MI_LOAD_REGISTER_IMM to program 0x2360 register	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Hang if using VEBox for GEC+3DLut and concurrent SFC scaling	This usage was not initially planned as there was no pre-si validation done. However, this was enabled on Si directly and the issue was found for a corner case schmoo. The usage did not involve SFC, instead AVS was used. This was changed to SFC. Hence, the workaround would involve reverting back to AVS usage too.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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 [bits 30:29]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
performance	LNCF MOCS settings are cleared on soft reset of RCS/POCS/CCS	WAReprogramMOCS: Upon render reset, the driver needs to reprogram LNCFCMOCS0 to LNCFCMOCS31 Programming note: WAReprogramMOCS: Upon render reset the driver needs to reprogram the LNCF MOCS Register.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	Register reads to 0x6604 is incorrect	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.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	[B0] *CS Changes: CS should stop making new DMA req once decided to go to RDOP	Issue: When Semaphore/Wait for event does not get satisfied, power management logic in CS might decide to Initiate Idle clock gating flows. WA: Disable RDOP for all Wait For Events like MI_SEMAPHORE_WAIT MI_WAIT_FOR_EVENT_2 MI_WAIT_FOR_EVENT.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	*CS Runlist fix for Use HW pointer reloading completed context	Ongoing execution of contexts in the hardware for a given engine can be stopped and make engine go idle by writing “Preempt to Idle” to the EXECLIST_CONTROL register. Following the “Preempt to Idle” flow, resume can be issued by writing “Load” with “Use HW Element Pointer” to the EXECLIST_CONTROL register. Due to known Hardware issue, “Load” with “Use HW Element Pointer” is not functional, hence following a “Preempt to Idle” flow, SW must do a fresh submission to the “Execlist Submit Queue” by submitting the required contexts to be submit queue followed by a “Load” to the EXECLIST_CONTROL register.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	imagestatus based on upper DW of half CL data	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; wa_status: driver_permanent_wa
other	Vertex fetch unit can fetch past end of vertex buffer resulting in page faults	WA: Add one extra page to the vertex buffer when in sequential mode.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
performance	Sampler cache can be thrashed in certain cases involving texture arrays resulting in low performance	added a programming note to the Render Surface State BXML saying the Array bit should not be set unless the depth of the arrayed surface is > 1.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	Read/write access to OAG registers blocked for non-priv batch buffers from RCS/POCS/CCS; required for certain performance instrumentation cases to work	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; wa_status: driver_permanent_wa
other	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; wa_status: driver_permanent_wa
hang	[SVM][B0 Revisit] Cs - Gam Deadlock after Root Entry Not Present Fault	WA Name: NoResetReadynessHandShake SW must not do Reset Readiness Handshake as part of the reset recovery on a CAT error.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	Display software needs to configure SSC enable in a new PLL register	DPLL SSC enable is not correctly hooked up to DPLL_CFGCR0 SSC Enable field. WA: Use DPLL_SSC sscen field to enable SSC instead of DPLL_CFGCR0 SSC enable field.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Spec clarification: Z Clear Color Location	There was a hole in the BSPEC definition for Clear value for the case of D24X8 depth surfaces. Added a programming note to BSPEC 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 HSD 397398 which is HW Managed Z Clear.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
	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 planes	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
security	[SECURITY] Accumulator is not currently cleared with GRF clear exposing its content to new context.	Clear ACC register before EOT send mov(16) acc0.0:f 0x0:f	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Corruption is seen on the top part of the Edge browser during Netflix AVC/HEVC playback at 4K resolution.	Resolve Compressed buffers prior to submission on Render Pipe for Protected Render scenarios.	sku: ALL; Stepping_impacted: a0; wa_status: driver_temporary_wa
power	AMFS Evaluate via Compute CS hangs if FFDOP clk gating is enabled	1. if compute shaders do evaluate, SW must program register 0x20ec[1] to 1 2. Shaders must not do Evaluate, in VF (Virtual Function) mode.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	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 | wa_status ALL | a0 | driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	[DX11][Corruption] 3DMark IceStorm/IceStormExtreme Demo - corruptions	To avoid sporadic corruptions “Set 0x7010[9] when Depth Buffer Surface Format is D16_UNORM , surface type is not NULL & 1X_MSAA”	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	TRTT Aliased Buffers Data Mismatch - Possible race condition between Mem Wr and HDC Flush	A "HDC fence" message must be inserted before the EoT of a compute, 3D or a pixel shader thread, if there is any HDC memory write requests from the thread. [L3 cache flush from the fence message is NOT needed].	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] Color: DaprSsDaprSc.ss_phase0.cpq_mask.sample_mask Mismatch	Disable CPS Aware color pipe by setting register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] DAPRSS Sending Blend CData Encoding For Fill CPQ	Disable CPS Aware color pipe by setting register bit: 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	PSDunit is dropping MSB of the blend state pointer from SD FIFO	Limit the Blend State Pointer to < 2G	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] Data Corruption on R10G10B10_FLOAT_A2_UNORM After Blend2Fill	See the Errata on Pre-Blend Color Clamping	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	Unexpected ResInfo results with LOD out of bounds.	When doing a resinfo message SW needs to check if any of the LOD values in an aligned 4 channel group is different: channel 0-3, 5-7, 8-11 , etc. If any of them are different It must sequence the resinfo message so that there is at most one unique LOD per valid channel in each 4 pixel group.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	[DAPRSS] Repcol with R10G10B10_FLOAT_A2_UNORM Not Properly Down-converted	Defeature Repcol Messages	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
	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; wa_status: driver_permanent_wa
	PCH display clock remains active when it shouldn't; impact to power and sleep state residency	Display driver should set and clear register offset 0xC2000 bit #7 as last step in programming south display registers in preparation for entering S0ix state, or set 0xC2000 bit #7 on S0ix entry and clear it on S0ix exit.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td>hang</td><td>Hangs can occur if using constant cache invalidate command [original HSD]</td><td>If the intention of “constant cache invalidate” is to invalidate the L1 cache (which can cache constants), use “HDC pipeline flush” instead of Constant Cache invalidate command. Some units bypass the L3 cache when they access memory - CS, MediaFF and Guc. When data sharing (e.g. semaphore) between these units and a shader is needed, the L3 cache may need to be invalidated using a pipe-control CS command with a "Const cache Invalidate" set. In these cases, the w/a should be to set the "State $ invalidate" in the pipecontrol command, in addition to the "HDC pipeline flush". Setting "state $ invalidate" will also invalidate the RO section (including constants) of L3 cache.</td><td><table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
<tr><td></td><td>dupunit not generating line_pop indication for plane with minimum size</td><td>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</td><td><table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table></td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
data_corruption, performance	Sampler power context is not saved/restored	Issue: Sampler Power Context save operation doesn't work correctly. This means that if driver writes to any of the following offsets (E100, E180, E184, E188, E18C, E190, E194) with non-default values will not be persisted across Render power gating/RC6. There are some cases we already know of where driver is expected/required to write non-default values for correct functional operation and best performance: All E18C[0]: E18C[15] In addition to these, more cases may be identified later where driver wants/needs to program these registers with non-default values and needs to have that programming be restored after render/RC6 power gating. Workaround: KMD to configure RC6 WA BB for RCS (CTX_WA_PTR) if not already enabled; allocate buffer to contain the commands and ensure it is pinned in GGTT. In the RC6 WA BB, include LRI command that writes to any offsets which require non-default values. More specifically, if KMD programs any of the 7 offsets identified above during driver boot and/or after engine reset, those same offset/value pairs must also include that offset/value in an LRI command in the RC6 WA BB for RCS. Note that all 7 of these offsets are masked registers (upper 16b mask; lower 16b value)- driver only needs to enable the mask bits for the specific bits it wants to program to non-default value (e.g. for the value for E18C could be 0x1001_1001 (e.g. mask bits 31 & 16 set to allow the values in bit 15 & 0 to take).	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	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.	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
hang	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 register bit: 0x07304 Bit[9]	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
data_corruption	PLANE_CC_VAL not getting updated immediately on async flip	Display async flips will not update the clear color value at the right point. The potential workarounds: WA1: KMD must convert async flip to sync flip upon clear color change. WA2: UMD must do partial resolve upon color clear change before submitting the flip to Display, KMD keeps async as async flip.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	B0+ Remove PM Req with unblock/memup + fill support -- SAGV enhancement not working as expected	SAGV fill timeout. Set 0x46434 bits 24 ,25, 26, and 27 to 1 at display initialization.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
hang	Hang on VECS reset due to clk gating issue in IECP	WA: Disable IECP clkgating by writing to 0x1C3F10[22]=1 and 0x1D3F10[22]=1	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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". Due to perf regression of disabling object-level preemption per topo, a blanket disable can be used instead. Disable Object Preemption Set 0x2580[0] = 0 or 0x20ec[0]. It is derived from the condition in RTL - object_preempt_en = 0x20e0[14] ? 0x2580[0] : 0x20ec[0];	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
data_corruption	[B0]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; wa_status: driver_permanent_wa
data_corruption	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: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	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) Reset VDBOX else Reset VDBOX and SFC 5. Release the force lock VE-SFC. else Reset VDBOX	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	3DSTATE_CONSTANT_ALL command not processed correctly in certain cases	1. The easiest W/A for the S/W is to use 3DSTATE_CONST command for individual shader instead of 3DSTATE_CONST_ALL COMMNAD. 2. To W/A this issue and to still use 3DSTATE_CONST_ALL command and not lose out on perf, we have to restrict the “Pointer to constant Buffer” filed to always have the address bits [12:8] as zero. Note this is just restricting the start address and CS can still prefetch CL as mentioned in size field. 3. If this address bits (Pointer to constant Buffer[12:8]) needs to be used, then only for those address range we can switch to shader specific push constant commands and rest address can still use 3DSTATE_CONST_ALL.	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	[Cayucos] Checkerboard background on text input and composition rendering across multiple apps.	0x7010[14] needs to be set for all media compressed render targets	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
other	Driver writes to SVL register offsets sometimes don't work correctly due to FFDOP clk gating	Disable FF DOP clk gating when accessing registers in SVL unit (range 0x7000-0x7FFC). This could be done: 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. OR statically disable FFDOP clk gating all the time via 20EC[1]=1 or 9424[2]=0 from driver boot. FFDOP is already being required to be applied all the time as security workaround for another issue (hard hang if non-priv BB sends 3D STATE command while pipeline_select is in GPGPU mode). As such, the simpler static w/a (option B, specifically 20EC[1] version) is preferred for simplicity/consistency.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
	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: c0; wa_status: driver_permanent_wa
data_corruption	Display underrun can occur on cursor plane if WM0 is used without WM1	Bug in the register unit which results in WM1 register used when only WM0 is enabled on cursor. A similar bug was fixed in the planes in 11p5, but Cursor was missed. Software workaround is when only WM0 enabled on cursor, copy contents of CUR_WM_0[30:0] (exclude the enable bit) into CUR_WM_1[30:0]	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
other	Depth stats (occlusion query) gives wrong results when using Render Target Independent Rasterization (STATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0) and no pixel shader bound	When 3DSTATE_RASTER::ForcedSampleCount != NUMRASTSAMPLES_0, SW should program a dummy pixel shader in case occlusion query is required.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	PSS flush done does not comprehend PSD state change, it only comprehends all PS threads completed.	WA: Insert a csstall after every 10 draws. Performance impact of this w/a on select DX9 workloads has been found to be negligible.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	Battlefield 4 + AA causing hang in MTunit	Fast Clear must not be used on 8- nor 16-bit-per-sample, MSAA color surfaces (e.g. B5G6R5, R8G8, R16, R8, etc. MSFMT_MSS surfaces), unless the following is true . . . 1. Surface Format is R8G8_UNORM, R16_UNORM, or R16_FLOAT. 2. Surface Width is a multiple of 8. 3. Surface Height is a multiple of 4. 4. Either . . . A. Surface is Tile64 (which is always the case for MSFMT_MSS on Tile64 platforms). B. Surface is TileYF or TileYS (which I don't think Windows or Mesa UMD's are currently using, apart from D3D Sparse Resources). C. Surface Horizontal Alignment is either HALIGN_8 or HALIGN_16. When implementing SW WA for this bug . . . If a surface meets 1+2+3 but not A/B, please also create the surface as HALIGN_8.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
other	PCH display HPD IRQ is not detected with default filter value	WA: Program 0xC7204 (PP_CONTROL) bit #0 to '1' to enable workaround and clear to disable it. Driver shall enable this WA when external display is connected and remove WA when display is unplugged or before going into sleep to allow CS entry. Driver shall not enable WA when eDP is connected.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	[MF][20h1] Blank screen seen with 4 MST Displays	Issue: Blank screen seen with 4 MST Displays WA: If MST primary is being enabled, clear DP VC Payload Bit before start of MST enable sequence and set is as part of regular MST enable sequence. If MST secondary is being added to the MST primary transcoder, keep the VC Payload allocate bit of secondary stream set throughout the MST secondary enable sequence. Clear DP VC Payload Bit only for MST/DP2.0 case before Wait for ACT Sent Status Handshake during Disable Sequence Keep DP VC Payload Bit ON as part of HDMI/DVI Enable/Disable Sequence. When enabling non-MST cases (eDP/DP-SST/HDMI/DVI), MstTransportSelect in TRANS_DDI_FUNC_CTL must be programmed to match the assigned pipe.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	TE DOP disable with idle flush enabled causes CS/CL/SVG hang	Disable IDLE flush during boot with the below sequence. Disable idlemsg via 2050[0], poll for CS FSM csbase+2AC[3:0] = 0 to show idle program TE DOP reenable idlemsg via 2050[0]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	[21H2][Surface Lucca]:Display junk and underrun on Pipe A while playing video using MTA with PSR2 enabled.	Issue: Display junk and underrun on Pipe A while playing video using MTA/ while launching edge browser/ opening folders/ interacting with windows icons and taskbar. WA: Set bit 0x46430[23]=0x1 whenever delayed Vblank is used.	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td>driver_permanent_wa</td></tr></table>
hang	Semaphore_signal with post sync enable does not send the correct signal data to GUC	Due to known HW issue, SW must not set “Post-Sync Operation” field for MI_SEMAPHORE_SIGNAL command	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr></table>
data_corruption	[RCS/CCS] 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. 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	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td>driver_permanent_wa</td></tr></table>

impact	title	bspec_wa_details	sku_impact
data_corruption	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; wa_status: driver_permanent_wa
hang,security	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; wa_status: driver_permanent_wa
performance	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; wa_status: driver_permanent_wa
hang	HW default value for fusedEU timeout for thread dispatch can hang HS / DS	The GS Timer Bits [31:24] in the GangTimer Register [MMIO: 0x6604] should be set to 0xE0 (224 decimal)	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
data_corruption	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 power context restore.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	SVG RTL doesn't correctly handle Push Constant buffer with length 0 when buffer address bit 5 is set; Results in render hang	Issue: SVG RTL not zeroing out address bit 5 when the Push Constant buffer length is 0. This is causing additional derefs to be generated. WA: Two options WA1 Program the Push constant buffer address in the Push constant command to be cacheline aligned i.e. make sure bit 5 of the address is set to 0, if any of the 4 push constant buffer length is programmed to be 0 for that constant buffer address. If the above WA is difficult to do, then please do this more generic WA WA2 Program the Push constant buffer address to be always cacheline aligned irrespective of buffer length i.e. make sure bit 5 of the address is set to 0 always in PC command programming.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa
data_corruption	Corruption in viewmask token coming into CL for POSH enabled workloads when TE DOP is disabled	Disable TEDOP Clock Gating with register bit 20A0 bit 19 set to 1 at boot + Disable POSH for draw calls with PRIM Replication OR PRIM ID enabled	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa

<table>
<tr><th>impact</th><th>title</th><th>bspec_wa_details</th><th>sku_impact</th></tr>
<tr><td>data_corruption</td><td>[BCS/VCS/VECS/POCS] Sometimes ctx time stamp register doesn't get restored to value from the engine context image on context switch</td><td>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. 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</td><td>
<table>
<tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr>
<tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr>
</table>
</td></tr>
<tr><td>data_corruption</td><td>[CS RTL] Timestamp Reporting in RCS compute mode</td><td>WA: - In compute Mode of operation for RCS, Driver must not program PIPE_CONTROL with Post_Sync “Write Timestamp” and Protected Memory Enable/Disable bits set in the same command. - Driver must instead use a separate Pipe_Control command to perform Post-Sync with “Write Timestamp” And another pipe_control command for protection memory enable/disable bits.</td><td>
<table>
<tr><th>sku</th><th>Stepping impacted</th><th>wa_status</th></tr>
<tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr>
</table>
</td></tr>
<tr><td>data_corruption</td><td>Read data corruption due to delayed Writes with State Access -- Sporadic failures in dEQP-VK.subgroups.basic.graphics.subgroupbarrier test</td><td>WA: Driver can enable HDC L1 cacheability for "read-only" buffers only. Setting a "read-write" buffer as L1 cacheable can corrupt memory data. L1 cacheability is set by programming MOCS[6:1] = [48, 59] (in decimal).</td><td>
<table>
<tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr>
<tr><td>ALL</td><td>a0</td><td>driver_permanent_wa</td></tr>
</table>
</td></tr>
</table>

impact	title	bspec_wa_details	sku_impact
hang	Hang due to deadlock created by RHWO scenario with RHWO optimization enabled.	WA: Disable RHWO by setting 0x7010[14] by default except during resolve pass.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
	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; wa_status: driver_permanent_wa
data_corruption	Page Faults: Write access to page marked as read only results in write being dropped, but fault may not be reported	Errata: WR permission faults may not be reported for write access to Read Only pages. SW can choose not to use read only pages OR just live with the fact that write accesses can be silently dropped without permission fault reporting.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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 register bit. 0x07304 Bit[9]	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
other	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 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; wa_status: driver_permanent_wa
data_corruption	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; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
other	Display combo PHY DPLL and thunderbolt PLL fractional divider error	Display DPLL and TBT PLL fractional divider value is shifted when reference is 38.4 MHz, giving slightly incorrect frequencies. Workaround when reference is 38.4 MHz, divide by 2 the value programmed into registers DPLL*_CFGCR0 and TBTPLL_CFGCR0 field DCO Fraction. Example, original DCO Fraction value of 0x7000h must be divided to 0x3800h.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	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	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: b0; wa_status: driver_permanent_wa
hang	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; wa_status: driver_permanent_wa
other	PSD is indicating the first payload phase as null for PSD_REG_P_BARY_PLANE phase	Ensure that for a PS thread dispatch (3DSTATE_PS_EXTRA[31]), when any one of the bits of 3DSTATE_PS_EXTRA[21:19] is set (requesting Z, W, BARY P/NP planes), we must have at least one of the following bits set - 3DSTATE_WM_BODY BitGroup0 [16:11] (Bary Interpolation Modes) 3DSTATE_PS_EXTRA[24] : Pixel Shader uses source Z 3DSTATE_PS_EXTRA[23]: Pixel Shader uses source W 3DSTATE_PS_EXTRA[18]: Pixel Shader uses Subsample Offsets 3DSTATE_PS_EXTRA[1:0]: Pixel Shader uses Coverage Mask 3DSTATE_PS BitGroup5 [4]: Position X/Y Offset	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	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 MFX codecs.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
hang	CSB data in hw status page may be stale when read out by SW (memory ordering for CS write vs engine interrupt delivery)	Driver initializes CSB data[0...11] with -1 during GPU initialization; - When driver receives the interrupt, it will try to read out the value of every new CSB data, if the value is -1, driver will reread it continuously in 50us until the value is not equal to -1; - After getting the value of every new CSB data, driver will write -1 back into current CSB data offset position; - If the valid value could not be read out in 50us, one warning will be given;	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
performance	dp panel will flicker when system idle at desktop with specific background picture	WA: The driver needs to program the FBC_STRIDE (0x43228) and enable the override stride once. The override stride should be programmed with : Compressed buffer seg stride (in CLs) = ceiling[(at least plane width in pixels * 4 * 4) / (64 * compression limit factor)] + 1 If the CFB size computed by: CFB size (in bytes) = Compressed buffer seg stride * Ceiling(MIN(FBC compressed vertical limit/4, plane vertical source size/4)) * 64, will not fit into the memory allocated to FBC, then driver will need to use a more aggressive compression limit factor.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
security	MI_FORCE_WAKEUP and engine reset happen at almost same time, then hang can occur	Prior to doing a reset, SW/FW must ensure command streamer is stopped. Setting both the ring stop and preparser enable bit in the below registers will cause the command streamer to halt. Note preparser is only enabled for RCS and CCS command streamers but bit exists in all CS's. MI_MODE set bit 8 GFX_MODE set bit 10	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
data_corruption	3DMark - Firestrike - corruption in OOTB run	WA: Program maximum of 1536 handles for GS.	sku: ALL; Stepping_impacted: a0; wa_status: driver_permanent_wa
hang	Panel Flicker after press F11 or Alt+Tab switch tasks under system.	Corruption seen when FBC is first enabled. After setting the FBC enable, wait for the next start of vblank, then write the plane 1A surface address register.	sku: ALL; Stepping_impacted: b0; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
data_corruption	Page Fault when small number of pixels using Sampler Feedback are rendered	Any kernel that contains AMFS evaluate (WriteSamplerFeedback) operations must issue two sampler cache flush messages after all evaluate operations are sent and before the kernel EOT message. The first sampler cache flush message must have a zero-length return. This is used to signal EOT to the AMFS unit. The second sampler cache flush message must have a non-zero-length return. This is used to block the kernel EOT until all AMFS operations are flushed out of the sampler. Failure to do both sampler cache flush messages can result in HW hangs and/or spurious page faults.	<table><tr><th>sku</th><th>Stepping_impacted</th><th>wa_status</th></tr><tr><td>ALL</td><td>b0</td><td>driver_permanent_wa</td></tr></table>