Intel® UHD Graphics Open Source

Programmer's Reference Manual

For the 2020 Intel Core™ Processors with Intel Hybrid Technology based on the "Lakefield" Platform

Volume 14: Workarounds

April 2021, Revision 1.0

Notices and Disclaimers

Table of Contents

Workarounds

impact	title	bspec_wa_details	sku_impact
hang	FC Blitter cannot handle a blit whose y-height%4 == 3 and y-height <= 8.	Issue: FC Blitter cannot handle a blit whose y-height%4 == 3 and y-height <= 8. This causes a system hang. WA: These blits must be detected and sent to the legacy blitter engine, not the fast copy engine.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	Message Channel: MCR unit allows read to go to disabled banks during a multicast request by CS to L3	Issue: In the case of a configuration where L3 banks are disabled, reads to L3 bank registers may return zeros instead of the value of the register. The HW does not direct the read based on the banks being disabled. WA: SW must maintain a copy externally to track the value in the case a read/modify write is needed.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	Command Streamer not sending flush to VF and SVG after Fence during PipeControl sequence of commands causing hang	Issue: CS Rtl not sending flush to VF and SVG after Fence during PC sequence of commands causing hang at ffclt WA: In set shader mode 3DSTATE_CONSTANT_* needs to be programmed before BTP_* At CS RTL boundary, this is the order of commands 1. Constant cycle on MCR 2. Fence command 3. BTP on MCR At SVG RTL boundary, this is the order of commands seen because of MCR delay 1. Fence 2. Constant Cycle on MCR 3. BTP on MCR At fence, although fence is a non pipeline state, CS is optimizing the flush and NOT sending the flush.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
performance	HW default value for L3 bank hashing is not optimal for performance	Issue: Register B004 bit[6:0] is used for HASH Control for Address Bit Exclusion, the default value causes reduced performance. WA: Software should program to b'0000001. Issue: Register B404 bit[11:5] are Bank Hash Address Exclude Bits, the default value causes reduced performance. WA: Software should program to b'0000001.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
	Issue with FBC Modify/Clear message generated with color cached in L3 (tile cache or data cache) in PTBR mode.	Workaround for Blitter Engine: "Blitter Tracking with Nuke" is the only FBC functional mode supported by blitter engine. SW must always program the "PPGTT Render Target Base Address Valid for FBC" to value '0' in BCS_ECOSKPD register. This would disable blitter engine from generating modify messages to FBC unit in display. If using Front Buffer rendering via BLT and display FBC compression feature is enabled, software must follow the BLT command that target the front buffer with the following: • Flush • LRI to 0x50380 with data 0x0000_0004 (This causes FBC to recompress the entire buffer after BLT operation). Workaround for Render Engine: "Render Tracking with Nuke" is the only FBC functional mode supported by render engine. SW must always program the FBC_RT_BASE_ADDR_REGISTER_* register in Render Engine to a reserved value (0xFFFF_FFFF) such that the programmed value doesn't match the render target surface address programmed. This would disable render engine from generating modify messages to FBC unit in display. Refer "Frame Buffer Compression" section for more details related to FBC functionality and programming.	sku: ALL; stepping_impacted: a1; stepping_fixed: ; wa_status: driver_permanent_wa
hang	OVR Issue if pocs_ovr_restart is asserted within 256 clks after the ctx restore is done	OVR Issue if pocs_ovr_restart is asserted within 256 clks after the ctx restore is done. WA: The WA could be to do a page pool size mmio write with a value of 0 followed by 256 noops before any page pool restart.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	VFURB dropping data in some scenarios involving *_256 /*_64 format	WA Name: WaNo256BitVFCompPacking 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; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
	Back-to-back send messages from threads on an EU where one of the requests is atomic cycle can result in cycles getting lost/corrupted	WaAtomicDisable Issue: Setting Atomic Control on SEND instruction does not guarantee back-to-back SEND messages. WA: Disable Pick 2nd EU optimization when using Atomic Control on SEND instruction (E48C[7]).	ALL	a0		driver_permanent_wa
data_corruption	3DSTATE_3D_MODE is not implementing modify enables correctly	Issue: 3DSTATE_3D_MODE has 16 of mask(modify enable bits) and 16 data bits. The behavior was expected to be the same as MMIO, when the mask bit is set, then the corresponding data bit could be updated. TDL and CPSS both implemented it such that the value is a ‘1’ when both the mask bit and the data bit is set. This means that SW cannot update a bit without doing a read/write/modify. WA: Driver must always program bits 31:16 of DW1 a value of 0xFFFF. This means if it is only updating 1 field, it must update all the fields to the correct value.	ALL	a0		driver_permanent_wa
	PreemptToIdle being reported with ActiveToIdle bit set in CSB	Issue: PreemptToIdle being reported with ActiveToIdle bit set in CSB WA: SW can ignore ActiveToIdle bit when PreemptToIdle is indicated.	ALL	a0		driver_permanent_wa
	Input Coverage = INNER is incorrectly ANDing sample masks	Issue: While designing CPS and depth coverage mode for input coverage for conservative rasterization implementation changed. Especially input coverage mode = INNER started ANDing sample mask to convservative rast mask. This results in the mis-match wrt 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	ALL	a0		driver_permanent_wa
	3D TYF surface corruption in MIP tail LODs because of X-adjacent RCC cacheline composition	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 Tile-YF 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	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
		!= AUX_NONE and RENDER_SURFACE_STATE.Tiled ResourceMode is Tile-YF or Tile-YS, 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)				
	Incorrect plane CSC coefficients for sRGB to Bt2020	Issue: SDR planes PLANE_COLOR_CTL Plane CSC Mode 100b, RGB709 to RGB2020, uses hardcoded R-Y coefficient of 0.75 instead of 0.625, resulting in incorrect BT2020 color conversion. WA: Limit RGB709 to RGB2020 conversion to the HDR capable planes.	ALL	a0		driver_permanent_wa
	Mid thread preemption hangs with sends instruction	Issue: Sends is having src1 dependency when it hits context save. So EU goes into save with dst and src0 overlapping GRF scoreboard of same sends. The Scoreboard for these registers are set when it restores. Src0 dependency is cleared with sends 2nd pass but dst dependency is not cleared. Sends creates a self-dependency resulting in hang. WA: When src and dest overlap in non-pagefault cases for sends instruction, we must use NoPreempt.	ALL	a0		driver_permanent_wa
	GFXDRV: Hang in cs tsg and eu with OCL workload "test_ocl_3D \half\test_half64.exe vloada_half -w CL_DEVICE_TYPE_GPU"	WA: In the kernel, introduce a dummy dependency on read register after every 3 send instructions. Example: sends null r10 r12 ExDesc Desc sends null r14 r16 ExDesc Desc sends null r18 r20 ExDesc Desc mov r12 r12 {NoMask} à dummy dependency on previous send sends null r22 r24 ExDesc Desc mov r16 r16 {NoMask} à dummy dependency on previous send	ALL	a0	b0	driver_temporary_wa
	Clockgating Issue on EOT & Barrier Ram Valid	Issue: Since EOT logic is not used in clock gating , the clock is gated off when EOT is not being processed because of unavailability of credit from TDC. This will result in barrier for that thread to not get cleared in GW. When the same barrier is programmed again, GW will go out of sync with stale information in it's Ram and would result in falsely clearing dependencies. This will result in a	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		hang. WA: Disable GWL clock gating. Set bit16 for register 0x9524 to 1.	
	Default value of the control bit that controls the dropping out put to the client Data return from Bank is Incorrect.	WA Name: WADisableBankHangMode The L3 error detection can be programmed to hang the GPU on a non-recoverable error due to ECC. The default value of the register currently enables the hang mode which is not desirable. So, a register programming is necessary to disable the hang mode. Kernel driver • Program 0x7034[9] to 'b1 This can be done in the "golden context" image for OS's that use such a construct or programmed into the context image by the kernel driver on a per-context basis at context create time.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	LKF GPGPU Preemption: Predicate_barrier_value mismatch	Predicate barrier - it won’t be used by software	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	POSH/PTBR workloads can hang if varying tile counts within a tile pass and preemption happens	WA Name: PoshPreemptionTilePassInfoCmd WA: Always program the same “Tile Count” value in 3DSTATE_PTBR_TILE_PASS_INFO with “End of Tile Pass” as that was programmed in 3DSTATE_PTBR_TILE_PASS_INFO with “Start of Tile Pass”.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	Invalid occlusion query results with "Pixel Shader Does not write to RT" bit	When Pixel Shader Kills Pixel is set, SW must perform a dummy render target write from the shader and not set this bit, so that Occlusion Query is correct.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	HS Hang & TDG mismatches when dual_instance_enable is zero AND HS is handle limited.	WA Name: WAHSMaxthread Issue: Hang occurs when the number of max threads is less than 2 times the number of instance count. WA: The number of max threads must be more than 2 times the number of instance count.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; 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; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
	Sampler register bit 0xe190[0] to disable DFR doesn't work properly	Issue: There are two registers that were implemented to allow SW to inhibit/disable Sampler DFR feature - E190[0] and 9550[9]. Of these (E190[0]) does not work properly and, if set (==1), can cause hangs. WA: If SW needs to disable it for any reason, SW should use 9550[9]. SW should always leave E190[0] at it HW default (==0, DFR enabled).	**sku** | **stepping_impacted** | **stepping_fixed** | **wa_status** ALL | a0 | | driver_permanent_wa
hang	EU hang with Timespy(DX12) workload	Issue: When a thread is using CE register for reading Mask value and other thread overwrites it, it results in reading of wrong mask value. WA: Do not use the CE register. Driver is able to add a couple extra instructions to work around it.	**sku** | **stepping_impacted** | **stepping_fixed** | **wa_status** ALL | a0 | | driver_permanent_wa
	GPU hangs on one of tessellation vkcts tests with DS not done.	WA: The send cycle, which is a urb write with and eot must be 4 phases long.	**sku** | **stepping_impacted** | **stepping_fixed** | **wa_status** ALL | a0 | | driver_permanent_wa
hang	GAM TLBPend Internal Error seen - results in hang for workloads on GT	TLB miss allocation perf fix is incorrectly detecting TLB hits as Hit on Miss for a corner case, eventually leading to a HANG. Recommendation: Disable the TLB miss allocate stall Perf fix by setting this config bit 0x4AB8[31] to 1.	**sku** | **stepping_impacted** | **stepping_fixed** | **wa_status** ALL | a0 | | driver_permanent_wa
	Hang can occur when POSH is enabled if RCS/POCS concurrently send certain cycles	Below two workaround's to be Implemented by UMD and KMD respectively: 1. WorkAround UMD: WaExplictTdlINPStateAck: SW must program the following MI_LOAD_REGISTER_IMMEDIATE (LRI) sequence after every “STATE_BASE_ADDRESS” command programmed in POSH_START batch buffer. LRI :: MMIO ADD:E700h, DATA: 0x0 <<Address E700h is assumed to be non existing register and hence write to this must not cause any side effect. This to flush the Message Channel Path from POCS to TDL, ensures NP state to TDL>> LRI :: MMIO ADD: 180F0h, DATA: F000_0000 << Clears dirty flag in POCS (TDL0..3), hence POCS will not generate any more implied NP state ack on 3DPRIMTIVE or stalling flushes >> LRI :: MMIO	**sku** | **stepping_impacted** | **stepping_fixed** | **wa_status** ALL | a0 | | driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		ADD: 180F8h, DATA: F000_0000 << Clears dirty flag in POCS (TDL4..7), hence POCS will not generate any more implied NP state ack on 3DPRIMTIVE or stalling flushes >> Register 0x180F0 must be whitelisted by SW. 2. Work Around KMD Add the below sequence as part of the POSH Enabled Per Context WA BB. LRI :: MMIO ADD:E700h, DATA: 0x0 <<Address E700h is assumed to be non-existing register and hence write to this must not cause any side effect. This to flush the Message Channel Path from POCS to TDL, ensures NP state to TDL>> LRI :: MMIO ADD: 180F0h, DATA: F000_0000 << Clears dirty flag in POCS (TDL0..3), hence POCS will not generate any more implied NP state ack on 3DPRIMTIVE or stalling flushes >> LRI :: MMIO ADD: 180F8h, DATA: F000_0000 << Clears dirty flag in POCS (TDL4..7), hence POCS will not generate any more implied NP state ack on 3DPRIMTIVE or stalling flushes >>	
	PSD clock gating causes WM flush done pulse to stay high	Disable PSD clock gating (0x94e4[5]=1)	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	Concurrent accesses to different portions of 128B "cacheline" of a 1D/linear mipmapped surface in RCC from multiple color pipes via different surface bpp descriptions can result in data corruption due to missing indication of shared state to CC in case of evict	Issue: For 1D/linear mip-mapped surfaces, each MIP must be accessed with the same pixel/texel format i.e. re-description of the sub-resource is not allowed. WA: SW should disable format re-description (optimization) for 1D or linear resource copied. Driver W/A (Windows) Tested. No perf impact.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	GT Hang can occur in Subslices PSD when using PTBR due to cycles getting lost due to missing back pressure when fifo full between hardware units	Issue: Normal MSC allocations are followed by a stream of back to back tile markers. MSC stalls further allocations (the interface that also includes tile markers) due to pending read returns from memory. Tile markers and flushes are pushed into an 8-deep FIFO in DAPRSS.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
		These are pulse interfaces from PSD->DAPRSS, so there is no backpressure. Due to backpressure from upstream, the FIFO in DAPRSS becomes full. Writes to the FIFO are qualified with fifo_full, so all subsequent tile markers and a flush from PSD are silently dropped. The drop of the flush leads to a hang. WA: Disable color discard mechanism for PTBR by programming 0x5580[15]=0 before any tile pass.				
hang	Hang can occur in OSB unit (hull shader related) due clock gating issue in certain corner cases	Issue: This is a clock gating issue in OSB (hull shader related) which causes a hang. WA: Disable the hsunit clock gating. Offset 9434 bit 8.	ALL	a0		driver_permanent_wa
	PTBR: Hang can occur in OVR context sort flush if >128 tiles are used	Issue: If there are 128 tiles in over context sort flush, the ovr counter never goes to 128 causing ovr to do context sort flush forever. WA: In PTBR mode a max of 127 tiles can be used.	ALL	a0		driver_permanent_wa
	Media compression issue: Issue during Macroblock processing during error concealment can result in page faults/engine soft hang	For all scenarios, 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 output surface for dummy reference as below: Dx12: disable MMCD, use output for reference, no intermediate buffer allocation needed. Dx11/Dx9/VAAPI: check output, if MMCD is enabled, make an intermediate allocation as dummy reference, otherwise use output, no extra allocation.	ALL	a0		driver_permanent_wa
	TDL unit incorrectly processes Sampler State Prefetch address shift in SARB for 16:6 & 18:8 modes	Issue: Incorrect TDL's SSP address shift in SARB for 16:6 & 18:8 modes. WA: Disable the Sampler state prefetch functionality in the SARB by programming 0xB000[30] to '1'. This is to be done at boot time and the feature must remain disabled permanently.	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	**stepping_impacted**	**stepping_fixed**	**wa_status**
data_corruption	3DSTATE_SAMPLE_PATTERN is not restored as part of POCS CTXT Restore	DisableSmallTriangleCulling for MSRT in PositionCS: Register bit “Disable Small Triangle Culling for MSRT” (0x18088[3]) in POCS must be always set.	ALL	a0		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).	ALL	a0		driver_permanent_wa
	3D mid-object preemption with instancing can result in incorrect handling of InstanceId on resubmission, resulting in hang.	Put the Instance ID enabled element at the last of a vertex.	ALL	a0		driver_permanent_wa
	clk gating bug in VS unit can cause UAV counters for HS, GS, TDS to result in hang	WA: Disable the vsunit clock gating. Offset 9434 bit 3.	ALL	a0		driver_permanent_wa
hang	VF Bank Collision Bug with POSH + instancing - can result in hang in VF	WA: Send 2 dummy 3DSTATE_VERTEX_ELEMENT in RCS whenever 3DSTATE_VF_INSTANCING[i].InstancingEnable = true for any element and the 3dprimitive is a posh enabled draw. Also, we need to disable POSH for the draws which has more than 30 elements since we won’t be able to add these 2 dummy elements in that case.	ALL	a0		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.	ALL	a0		driver_permanent_wa
	Corruption seen for some tesselation workloads when TEDOP clk gating enabled	Issue: Corruption seen on floor around frame 600 to 650for some tesselation workloads when TEDOP clk gating enabled. WA : mmio_write offset 20a0 value 00080000	ALL	a0		driver_permanent_wa
	Depth buffer corruption when state cache invalidate is triggered from POCS	1) Insert PIPE_CONTROL with CS_stall prior to any PIPE_CONTROL with Read Only State Invalidate if engine is still active 2) Workaround Below. Note that the WA can be changed such that RCS can invalidate when POCS is active. enum ePipeSync { eBusy = 0, eIdle = 1, eNeedsSync = 2 };	ALL	a0	c0	driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		ePipeSync poshStatus, rcsStatus; The POCS and RCS command buffers write to their semaphore location to indicate their status to the other pipeline. The Idle is updated after any PipeControl with CS Stall as a Post-Sync operation, the Busy is written before any following commands (state or rendering) – but must check whether the other pipeline needs a sync first. i.e. // Do not start state or rendering operations while the other pipeline is invalidating caches Mutex { SemaphoreWait(OtherPipeline < eNeedsSync); // Lock out the other pipeline from invalidating caches StoreDataImmediate(eBusy); } // State and rendering commands // When the pipeline goes idle PipeControl(CSStall, PostSync Write eIdle); // Pipeline is idle – allow the other pipeline to do any required cache invalidates When an SBA (or Cache Invalidate is needed), the following commands are inserted. // Flush this pipeline, in case both pipelines need to synchronize PipeControl(CSStall, PostSync Write eIdle); // (this might not be needed) // Request the other pipeline waits Mutex { PipeControl(CSStall, Post Sync Write eNeedsSync); } // Wait until the other pipeline acknowledges that it has halted. SemaphoreWait(OtherPipeline >= eIdle); SBA or PipeControl with Cache Invalidate // Remove the request for the other pipeline to wait StoreDataImmediate(eIdle); option use PipeControl(CSStall, PostSyncWrite eIdle); instead – but shouldn't be needed This approach allows both pipelines to invalidate the caches at the same time, so avoids the deadlock scenario. It assumes that we will eventually be doing a CSStall to inject the idle signal, but it can also be injected occasionally (using a predicate as follows) in the middle of rendering operations if we do not have sufficient naturally occurring	

impact	title	bspec_wa_details	sku_impact
		pipeline flushes. Provided there are enough opportunities for the pipeline to complete the handshake without fully draining, we should avoid the cold restart costs. Potential performance enhancement (if needed) – however ending the Tile Pass requires a pipe flush as Tile Pass Info is NP state. Predicate (if OtherPipeline == eNeedsSync) // Use Predicated BBS to skip { // Flush and indicate that this pipeline is idle PipeControl(CSStall, PostSyncWrite eIdle); Mutex { // Wait for the other pipeline to complete its operation SemaphoreWait(OtherPipeline, < eNeedsSync); // Resume operations StoreDataImmediate(eBusy); } } Before starting any other semaphore, the pipeline must be flushed, and the state set to eIdle. Mutex Option 1 (using MI_PREDICATE) A initialized to the value 0 Start Mutex PREDICATE Never SEMA_WAIT (A == 0) ATOMIC Increment A, Read Return -> GPR4 REG2REG GPR4 -> Predicate result PREDICATE Clear (Pred Result[0] = 0) (values are 1 or 2) ATOMIC Decrement A BBS <Start Mutex> PREDICATE Never Got Mutex <BLOCK> End Mutex ATOMIC Decrement A Mutex Option 2 (using Predicated Batch Buffers) <= Preferred A initialized to the value 1 Start Mutex PREDICATE Never SEMA_WAIT (A == 1) ATOMIC Increment A, Read Return -> GPR4 REG2REG GPR4 -> PREDRESULT_1 BBS Predicate Enable (Got Mutex) ATOMIC Decrement A LRI PREDRESULT_1, 1 (for the benefit of GTX) BBS <Start Mutex> (Predicate Enable – for the benefit of GTX) Got Mutex <BLOCK> End Mutex ATOMIC Decrement A	
hang	Hang seen in hull shader unit enabled on some workloads in boundary case	resubmit 3D State HS for every draw call containing Hull Shader.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
	Corruption with FBC and plane enable/disable	Issue: Corruption with FBC around plane 1A enabling. WA: 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.	ALL	a0		driver_permanent_wa
data_corruption,performance	Incorrect default power-on GAPZ arbiter register value.	Register B004 bits [27:22] select the priority of arbitration among RCZ, STC and HiZ in the GAPZ. The priority requested by the Z team is RR(RCZ, STC) < HiZ. However, the default value of this register puts the priority scheme at RCA < STC < HiZ which has been seen to reduce performance in some cases. WA : During driver boot, reprogram the value of this register to '111111b.	ALL	a0		driver_permanent_wa
	Pause register is being set to too small a value. It only triggers for 1 clock, which is under the MCP value	Pause Register decrements every 2 EU clocks, and must be programmed with value greater than or equal to 4.	ALL	a0		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	ALL	a0		driver_permanent_wa
hang	Incorrect read back of 0xA0000-BFFFC	Incorrect readback of VGA space. Read 0x403cc, ignoring the return data, when initializing display.	ALL	a0		driver_permanent_wa
data_corruption	Display underrun when PM fill is issued while an interrupt is pending	Set the register offset 0x46430 bit 22 to 1. This prevents immediate NACK to the Fill PM request while there is a pending interrupt (some other cases where memory bring down is not allowed). Display responds with a NACK after the buffers are filled to top and pm_fill is de-asserted as expected.	ALL	a0		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].	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
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.	ALL	a0		driver_permanent_wa
hang	Non privileged batch buffers cannot use MI* register access commands to read PS Invocation Count and PS Depth Count (required for occlusion queries and pipeline statistics)	WA: Driver should use RCS FORCE_TO_NONPRIV mechanism grant read access to PS invocation count and PS depth count register offsets.	ALL	a0		driver_permanent_wa
performance	Performance counting usage requires write access from non-privileged batch buffer to certain OA registers; HW default blocks such access from non-priv batch	Software must use the Force_To_Non_Priv registers to enable Read/WRITE access to the below register offsets RCS: OAPERF_A20 0x28A0 OAREPORTTRIG6 0x2754 POCS: OAPERF_A19 0x2898 OAREPORTTRIG6 0x2754	ALL	a0		driver_permanent_wa
performance	Performance counting usage requires write access from non-privileged batch buffer to OA registers 0x2898 and 0x28A0 from RCS/POCS respectively; HW default blocks such access from non-priv batche	Kernel driver to add offset 0x2898 (OAPERF_A19) to the list of "whitelisted" registers for W access from RCS (via RCS Force_to_nonpriv register programming). Kernel driver to add offset 0x28A0 (OAPERF_A20) to the list of "whitelisted" registers for W access from POCS (via POCS Force_to_nonpriv register programming).	ALL	a0		driver_permanent_wa
	HDMI symbol mismatch in hblank when pll=4	WA: Do not enable HDMI with 10bpc and pixel repeat 4x.	ALL	a0		driver_permanent_wa
data_corruption	Pipe CSC registers are getting disarmed on reads	Pipe CSC register updates not taking effect as expected. WA Sequence to arm/disarm Pipe CSC registers: Step A: Wait for start of Vblank or safe region before start of Vblank. Step B: Write the CSC registers, CSC_COEFF*, CSC_PREOFF*, and CSC_POSTOFF* with programming set A. Step C: Arm the CSC registers by writing to CSC_MODE register. Step D: Once the registers are armed, do not read the CSC registers, CSC_COEFF*,	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		CSC_PREOFF*, CSC_POSTOFF* until after the next start of Vblank, since reads at this point will disarm the registers and set A programming would not take effect. Step E: Wait for next start of Vblank. Set A programming takes effect here if no reads or writes have occurred since arming. Step F: Reads of the CSC registers can occur here, even though the registers are disarmed, Set A programming is retained. Note: If the CSC registers are not read back, then no adjustment to programming is needed. If the CSC registers, CSC_COEFF*, CSC_PREOFF*, CSC_POSTOFF* are written to with new programming set B, between step C. and step E, then set B will take effect on step E Vblank.	
data_corruption	Transcoder WD tail pointer scan line count corrupted after fault	WA: On receiving a fault, turn WD transcoder off and then back on again before starting another capture.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	Use {NoPreempt} when r2 is used as src0 of sends in regular kernel.	Use of {NoPremept} switch is required whenever r2 is used as src0 for sends instruction. This is to reserve use of r2 for SIP during context save and restore.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
other	Draw with depth test disable/stencil test enabled and with pixel location outside of bound stencil buffer but within viewport leads to page faults or incorrect data	If stencil test is enabled and depth test is not enabled, ensure the clipping/draw rectangle dimensions are clamped to the size of the stencil buffer boundaries.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	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	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
		!= 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)				
	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.	ALL	a0		driver_permanent_wa
hang	South display register access hangs after FLR	Hang when south display registers are accessed after Function Level Reset (FLR), which can be initiated through PCI config (VMM FLR) or through MMIO (driver FLR). Set GMBUS0 Pin Pair Select to 1 at boot and each FLR exit. During GMBUS transactions it can be changed to different values, but return it to 1 after the GMBUS transactions are done.	ALL	a0		driver_permanent_wa
	dupunit not generating line_pop indication for plane with minimum size	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	ALL	a0		driver_permanent_wa
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.	ALL	a0		driver_permanent_wa
other	Mid-frame VTD Disable toggle leading to underrun	BIOS WA: Disable GFX VTD (DMA remapping) in bios until OS boot completes; then it may be re-enabled (or not), depending on the OS default setting.	ALL	a0		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.	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact			
			sku	stepping_impacted	stepping_fixed	wa_status
hang	PS Invocation Count and PS Depth Count unavailable to UMD	SW Must Whitelist PS Invocation Count and PS Depth Count by using the Force_To_Non_Priv registers	ALL	b0		driver_permanent_wa
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.	ALL	a0		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	ALL	a0		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]	ALL	a0		driver_permanent_wa
hang	During MTP flow and when Dispatch Pipeline is idle- Done signal from TDC to TSG going inactive after some extra clocks than what TSG is expecting	SKUs with number of EUs <=32 EU: Disable Mid-Thread Pre-emption	ALL	a0		driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
security	MI_FORCE_WAKEUP and engine reset happen at almost same time, then hang can occur		sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	[GTPM LP] Fixes for L3 not working due to a hang in GFX Flush	Driver needs to program SCRATCH_LNCF2[22:20] = "111" before start of the work load.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
data_corruption	Diagonal error propagation for vertical intra refresh on H264 VDEnc	Issue: For vertical intra refresh on H264 VDEnc, there is an issue with error propagation in diagonal direction when the block is predicted from top-right. The issue still occurs when the deblocking filter is disabled. WA: The solution is to disable all prediction modes that uses reference values from not refreshed area. Those are modes 3,7 for 4x4 and modes 0, 2, 3, 4, 5, 7 for 8x8 (due to filtering). In the driver code it looks like: AvcIntra4X4ModeMask = 0x88 AvcIntra8X8ModeMask = 0xBD	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
	DPST IET data writes do not trigger PSR exit due to lack of write event indication to DMUX	Since the reason for image enhancement not being applied is that the writes to the DPST registers are not causing an exit from PSR. The WA is to do a R/W to PAL_LGC_A_0: 0x4A000 after the Image Enhancement update. This will trigger a PSR exit immediately.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
hang	FLR cycling test causes hang	Intermittent graphics function level reset failures. Set register 0xC2020 bit 15 to 1b in the display initialization sequence before setting NDE_RSTWRN_OPT RST PCH Handshake En to 1b.	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa
performance	[RVP][20H1] 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,	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

impact	title	bspec_wa_details	sku_impact
		will not fit into the memory allocated to FBC, then driver will need to use a more aggressive compression limit factor.	
hang	[PO] 3 Strike on S4 cycles open transaction to GFX MMIO NDE_RSTWRN_OPT (0x46408)	WA: System BIOS to program 0x10_1038[23:16] (DG_CLKREQ_POLICY[CLKREQ_HYST_CNTR]) to 0x4 (default is 0).	sku: ALL; stepping_impacted: a0; stepping_fixed: ; wa_status: driver_permanent_wa

sku	stepping_impacted	stepping_fixed	wa_status
ALL	a0		driver_permanent_wa