Intel® Open Source HD Graphics Programmers' Reference Manual (PRM)

Volume 16: Workarounds

For the 2014-2015 Intel Atom™ Processors, Celeron™ Processors and Pentium™ Processors based on the "Cherry Trail/Braswell" Platform (Cherryview/Braswell graphics)

June 2015, Revision 1.0

Creative Commons License

Notices and Disclaimers

Table of Contents

Workarounds

Functional Area	Component	Workaround Name	Workaround Description
3D	Data Port Messages		Render Target Write messages require a header when Pixel Shader Computed Depth is enabled.
3D	SURFACE_STATE		The R32_FLOAT, R32G32_FLOAT and R8G8_UNORM surface formats are not handled correctly for the Alpha channel for cases where the sample is off-map or out of bounds. The correct behavior should be to force these Alpha values to 1.0, but actual result is for these formats is a blending of the default color with 1.0.
3D	3D Sampler Message Types		When a surface of type surfaceType_Null is accessed by resinfo, the MIPCount returned is undefined instead of 0.
3D	Message Header		Offu/offv are calculated in normalized space and hence subject to small truncation error.
3D	3D Sampler Messages and Message Types		Offu/offv are calculated in normalized space and hence subject to a small truncation error.
3D	Vertex Fetch Functions		If a 32 bit uscaled or sscaled format is used, then a float format needs to be used so VF will keep the data as is and the kernel needs to convert the format to 32 bit float.
3D		WaDisableAsyncFlipPerfMode	Async flips hang on MI_WAIT_FOR_EVENT following it. WaDuplicateMiDisplayFlip does not seem to help. To workaround this we have enabled MI_SYNC_FLIP always.

3D	Patch Header DW0-7		The Tessellation stage will incorrectly add domain points along patch edges under the following conditions, which may result in conformance failures and/or cracking artifacts:
			
			QUAD domain
			
			INTEGER partitioning
			
			All three Tess Factors in a given U or V direction (for example,, V direction: UEQ0, InsideV, UEQ1) are all exactly 1.0
			
			All three Tess Factors in the other direction are > 1.0 and all round up to the same integer value (e.g, U direction: VEQ0 = 3.1, InsideU = 3.7, VEQ1 = 3.4)
			
			The suggested workaround (to be implemented as part of the post-amble to the HS shader in the HS kernel) is:
			
			if (
			   (TF[UEQ0] > 1.0) ||
			   (TF[VEQ0] > 1.0) ||
			   (TF[UEQ1] > 1.0) ||
			   (TF[VEQ1] > 1.0) ||
			   (TF[INSIDE_U] > 1.0) ||
			   (TF[INSIDE_V] > 1.0))
			{
			   TF[INSIDE_U] = (TF[INSIDE_U] == 1.0) ? 2.0 : TF[INSIDE_U];
			   TF[INSIDE_V] = (TF[INSIDE_V] == 1.0) ? 2.0 : TF[INSIDE_V];

3D		WaPreventHSTessLevelsInterference	Inner tessellation levels can interfere with outer tess levels. The proposed SW workaround is to have the HS compiler, upon seeing INTEGER and QUAD, generate instructions to perform the following logic after or when moving the TFs into the Patch Header:
			if (
			 (TF[UEQ0] > 1.0) ||
			 (TF[VEQ0] > 1.0) ||
			 (TF[UEQ1] > 1.0) ||
			 (TF[VEQ1] > 1.0) ||
			 (TF[INSIDE_U > INSIDE_U] > 1.0) ||
			 (TF[INSIDE_V > INSIDE_V] > 1.0))
			{
			 TF[INSIDE_U|INSIDE_U] = (TF[INSIDE_U|INSIDE_U] == 1.0)
			    ? 2.0: TF[INSIDE_U|INSIDE_U];
			 TF[INSIDE_V|INSIDE_V] = (TF[INSIDE_V|INSIDE_V] == 1.0)
			    ? 2.0: TF[INSIDE_V|INSIDE_V];
			}
			Note that the non-inside TFs are never modified, so this won't impact the HW patch cull testing for 0= or NaN. (This serves as a warning to watch out for NaN comparisons).

3D	GPGPU Indirect Thread Dispatch		The indirect registers are not supposed to be set to 0, but sometimes the kernel computing the value wants no work done and sets them to 0. This does not work correctly, so a workaround in the command stream is needed. The following workaround sets the predicate if IndirectArgs.X=0 or IndirectArgs.Y=0 or IndirectArgs.Z=0

Command	Operation
MI_LOAD_REGISTER_IMM	MI_PREDICATE_RESULT = 0
MI_LOAD_REGISTER_IMM	MI_PREDICATE_SRC1 = 0
MI_LOAD_REGISTER_MEM	MI_PREDICATE_SRC0 = [IndirectArgs].X
MI_PREDICATE	LOAD, OR, SOURCES_EQUAL
MI_LOAD_REGISTER_MEM	MI_PREDICATE_SRC0 = [IndirectArgs].Y

			MI_PREDICATE	LOAD, OR, SOURCES_EQUAL
			MI_LOAD_REGISTER_MEM	MI_PREDICATE_SRC0 = [IndirectArgs].Z
			MI_PREDICATE	LOADINV, OR, SOURCES_EQUAL
			WALKER	Predicate Enable = 1
3D	TypedUntyped Surface ReadWrite and TypedUntyped Atomic Operation		The Typed Surface Read message returns 0 in all channels for out-of-bounds accesses.	
3D	Shared Local Memory (SLM)		SLM accesses are incorrectly treated as out-of-bounds when the General State Buffer Size in STATE_BASE_ADDRESS is set to zero. The workaround is to set the General State Buffer Size to a non-zero value.	
3D	Media State and Primitive Commands		The MEDIA_STATE_FLUSH command is updated to optionally specify all the resources required for the next thread group via an interface descriptor – if the resources are not available the group cannot start.	
3D	Shared Local Memory (SLM)		When SLM memory is disabled in the L3 configuration, SLM accesses are correctly treated as out-of-bounds but the out-of-bounds data returned may not be zero.	
3D		WaDisableVFUnitClockGating	VF drawing too many verts on preemption restore.	
3D	Data Port Messages		For Atomic Counter OPS other than INC, DEC, and PREDEC, the header is forbidden and not optional as indicated in the table.	
3D		WaDividePSInvocationCountBy4	Invocation counter is 4 times actual.	
			WA: SW to divide HW reported PS Invocations value by 4.	
3D		WaVSRefCountFullforceMissDisable	Hang; For GS\VS workloads. WA: Set FF_MODE - Thread Mode Register, 0x020A0: DS Reference Count Full Force Miss Enable, bit 19 = 0, VS Reference Count Full Force Miss Enable, bit 15 = 0. HW issue that requires VS_REFERENCE_COUNT_FULL_FORCE_MISS_ENABLE (#15) bit of MMIO 0x20a0 to be clear. Since this bit is always clear by default, the driver shouldn't do anything as long as the WA is enabled and should set this bit if the WA is disabled. To be done at the boot time.	

3D		WaHizAmbiguateRequiredNonAlignedBeforeRender	Corruption. WA: Must do HiZ resolve (ambiguate) on any portion of the HiZ buffer that is not 8x4 aligned before rendering or clearing.
3D	Atomic Counter Operation Message Descriptor		For Atomic Counter OPS other than INC, DEC, or PREDEC, the message header is forbidden and not optional.
3D		WaIndirectDispatchPredicate	GFXDRV: Hang in GT2 WW03d Emulation - GPGPU _WALKER HANG. WA: Use MI_PREDICATE commands to predicate an indirect dispatch whose thread groups in any dimension = 0. GPGPU pipe3d: vfe handle counter underflow for dimx eq 0 followed by curbe workload.
3D	Universal Input Message Phases		WA: The following text needs to be maintained so that we can bring back the feature in the next opportunity. Will be used for Field 16x8 Enabled: This field enables 16x8 interlaced–block partitioning for MPEG-2. Note: Enabling Field 16x8 prevents use of sub-partitions types 8x16, 4x4, 4x8 and 8x4, RefAccess and SrcAccess must be 0 and SrcSize must be 16x16 (00). Field8x8 and Field16x8 are mutually exclusive.
3D	Notification Registers		The sub-register numbers for n0.0 and n0.2 are swapped on a write, that is, a destination of n0.0 is required to update n0.2 and n0.2 is required to update n0.0.
3D		AccWrEnNotAllowedToAcc1With16bit	Implicit write (AccWrEn) to acc1 (for example, H1, Q3) not allowed with 16-bit data type (for example, hf, w). Acc1 dependency is not set as EUTC doesn't detect it as write to 2nd half accumulator for half float Q3 and takes it as acc0 dest. It is using nib_ctrl[1] to determine which part of acc is being written. this works for float dest but for hf destination nib_ctrl[2] should be used.
3D		WaEnsureMemCoherencyBeforeLoadRegisterMem	MI_STORE_REG_MEM followed by MI_LOAD_REGISTER_MEM does not always stall before the LOAD. Requires that a sync event (like MI_ATOMIC or PIPE_CONTROL with CS stall) be issued between the events to ensure that the memory has been forced to be coherent. See B-Spec for restriction.

3D		WaFastClearZ16K	The definition of the HiZ based fast clear does not specify the ability to clear all the way in X and Y to 16K. The last HiZ in the X and Y would be only partially initialized. Workaround suggested is to clear as four (2x2) sets of 4k x 4k clears. The rectangle size in the HZ_OP packet can only be programmed up to 16k-1 pixels in x and y direction. We need one more bit in each direction to be able to clear the whole thing.
3D		WaFlushOpOnCSStall	When a pipecontrol with stall bit is set, an ISP bit is enabled and there is no post sync operation: CS is not doing a flush or dummy atomic while fulsim does and there is a mismatch. Restriction: if stall bit is set, any one of the post sync-operation should be set.
3D		WaRestoreFCandMSGRegistersFromUpperOword	PipeGT Page Faults: Incorrect gpgpu_phase in EUGA causes MEU save cycles to be saved incorrectly.
3D		WaOGLGSVertexReorderingTriStripAdjOnly	GS vtxram read pointer increments incorrectly if control topology value matches tristrip_adj* decoding value...value (6'b001100 , 6'b010101). This causes full to not be generated properly and eventually corruption of the GS internal storage structure..
3D		WaDSRefCountFullforceMissDisable	DS_REFERENCE_COUNT_FULL_FORCE_MISS_ENABLE (#19) bit of MMIO 0x20a0 needs to be clear in all steppings. Since this bit is always clear by default, the driver shouldn't do anything as long as the WA is enabled and should set this bit if the WA is disabled. To be done at the boot time.
3D		WaGrfDepClearOnOutstandingSamplerInGpgpuContextSave	Feature: GPGPU Pre-emption Non-page faulting. Outstanding sampler messages to EU, pre-emption occurs before data return. Hang condition. WA: Software workaround in SIP: a. After jumping to save routine, check "fault status" bit in SR0. b. If it is set, that means there are faulted GRF registers and EU is in page fault mode, so we do not need to wait for them before saving. Jump to normal save routine. c. If it is not set, that means there are no faulted GRF registers, but there may still be outstanding sampler registers to wait for. d. Use mov instructions for each GRF register, with null as the destination, to check dependency on each register, to make sure they have returned. e. Continue with normal save routine.

3D		WaGuardbandSize	Cref precision loss in Z interpolator (src_cref mismatch causing HIZ memdiff). WA: Limit guardband to -16k to +16k.
3D		WaDisableRSBeforeBTPoolDisable	BT edits are dropped in the R S if the pool is disabled. WA: Disable RS before BT pool is disabled.
3D		WaForcenonzeroSBEOutputAttributeCount	WA: Clamp SF number of output attributes to min value of 1 to avoid GS running out of handles.
3D	LOD Information		WA: No range checks are performed on LOD from the Address Payload. The low 4 bits of LOD in the Address Payload are used to right shift the width, height, and depth values.
3D		WaSubtract1FromMaxNoOfThreads	Subtract 1 from Max No of Threads per PSD (Per Slice Dispatch) because the PSD RTL incorrectly adds 1 to the max thread programmed. So when programming "MaximumNumberOfThreadsPerPSD" in 3DSTATE_PS command, always subtract 1 from the value to be programmed.
3D		WaUseNonPrivRegisterForObjectLevelPreemption	Use INSTPM(20c0) MMIO regs to disable Mid-Object pre-emption for draw call.
3D		WaStateBindingTableOverfetch	HW over-fetches two cache lines of binding table indices. When using the resource streamer, SW needs to pad binding table pointer updates with an additional two cache lines.
3D	Floating-Point Support		When converting from float to int, rounding mode RZ must be used.
3D		WaVfPostSyncWrite	Set post sync op of write for PIPE_CONTROL when only VF cache invalidate set.
3D		WaCsStallBeforenonzeroInstanceCount	Issue CS Stall when going from 3DSTATE_HS zero instance count or HS disabled to 3DSTATE_HS non-zero instance count.
3D	Byte Scattered ReadWrite		The stateless model is not supported. A workaround is to use the A64 Byte Scattered message, adding the General State Base Address into each address offset in the message. However, the A64 message will not perform A32 stateless bounds checking.

3D	TypedUntyped Surface ReadWrite and TypedUntyped Atomic Operation		Tile W surfaces must be of format R8_UINT and only support SIMD8. Furthermore, only the RED channel can be enabled.
3D	3D Sampler Message Types		Message sample+killpix cannot be used with Fault & Stream mode.
3D	Stateless Model		A64 Stateless accesses are incorrectly treated as out-of-bounds when the General State Buffer Size in STATE_BASE_ADDRESS is set to zero. The workaround is to set the General State Buffer Size to a non-zero value.
3D	Message Mode: M0.2, bit 31:30		BYTE_MASK is not supported.
3D			PSD variable dispatch with per sample 4x and 8x - restriction from previous project has to be removed. WA: Enable only SIMD8 dispatch mode when in per-sample mode.
3D		WsVfPostSyncWrite	Hang: If the push constant is committed using PIPE_CONTROL with only VF cache invalidation then CS commits the push constant but does not send dirty bits to shaders. Thus the shaders will not see the dirty bits and if there is new push constant command followed by a stalling flush, CS will commit the second push constant and wait for the previous push constant deref. This causes the flush to hang as CS waits for "push constant done" in flush completion. WA:"Post Sync Operation" must be enabled to "Write Immediate Data" or "Write PS Depth Count" or "Write Timestamp".
3D	Media State and Primitive Commands		Every 8 MEDIA_OBJECT or MEDIA_OBJECT_WALKER commands that use indirect payload must be followed by a MEDIA_STATE_FLUSH or PIPE_CONTROL.
3D	Extended Math Function		Mixed mode operations are not supported
3D		WaForceMinMaxGSThreadCount	GS being stalled can cause the fftid to go over max threads causing undefined scratch space to be used. WA: Limit the number of handles to the number of threads, with some GS performance loss. Set min/max threads to 8 for GS. Should be handled in USC/IGC.

3D		WaGrfScoreboardClearInGpgpuContextSave	Need to use stop_done pulse to clear grf scoreboard on save. Logic exists to restore grf scoreboard based on MDE data being restored to MEU. WA: Software workaround in SIP: State register special handling against page fault issue; change is requested by EU team. In Context save sr0.1 register is stored in temporary register, temporary register is masked and sent to csr buffer, next sr0.1 is cleared. In context restore sr0.1 is restored as one of the last registers (just before r0 restore and exception clear).
3D			WA: A PIPE_CONTROL with CS_STALL must be sent whenever the HS_STATE.InstanceCount changes from 0 (no instancing to > 0 (instancing).
3D		WaAdditionalMovWhenSrc1ModOnMulMach	A source modifier must not be used on src1 for the mul/mach macro operations. WA: Use extra move instead of src modifier for src1.
3D		WaRestoreFC4RegisterDW0fromDW1	GPGPU Pre-emption - Execution Mask not being saved/restored correctly (memdiff). WA: SIP routine has to correct the address while restoring. Flow control register FC4 has to be restored from DW1.
3D		WaScalarAtomic	This is a performance improvement implemented as a W/A. Improves append counter updates from 1/6 clks (L3 limit) to 16/6 clks.
3D		WaRestoreFc0RegistersWithOffset	GPGPU Pre-emption - EUStress nested if memdiff. WA: Flow control register FC0 has to be restored with offsets. Restores fc0.4 to fc0.31 register with special offset in pre-emption context restore.
3D	Media State and Primitive Commands		Two MEDIA_STATE_FLUSH commands need to be used to ensure that the flush is complete.
3D	Extended Math Function		When both srcs are NAN, FDIV produces denorminator NAN as output.
3D		WaThreadSwitchAfterCall	GPGPU Pre-emption - CALL Inst ruction Hang. WA: Follow every call by a dummy non-JEU and non-send instruction with a SWITCH for both cases whether a subroutine is taken or not.
3D	MEDIA_STATE_FLUSH		A MEDIA_STATE_FLUSH with no options must be added after a GPGPU_WALKER command which doesn't use either SLM or barriers.

3D		WaNearestFilterLODClamp	DX10.1 LOD clamping VS Max LOD DX case.
			
			Workarounds:
			DX:
			  If (mipfilter_nearest)
			   MaxLOD = floor(MaxLOD)
			   MinLOD = floor(MinLOD)
			OGL:
			  If (mipfilter_nearest)
			   lodbiad = lodbiad - 0.000001b
			
			Dx9 - (Not Required for Dx9. Max always set to to 14.0)

3D	GPGPU Context Switch Workarounds		After either a MI_SET_CONTEXT or a PIPE_CONTROL with Generic Media State Clear, there must be a MEDIA_VFE_STATE command before any pre-emptable command. The parameter of this MEDIA_VFE_STATE command can be set to default values.
3D	GPGPU Indirect Thread Dispatch		CURBE should be used for the payload when using indirect dispatch rather than indirect payload
3D			SLM accesses are incorrectly treated as out-of-bounds when the General State Buffer Size in STATE_BASE_ADDRESS is set to zero. The workaround is to set the General State Buffer Size to a non-zero value. When SLM memory is disabled in the L3 configuration, SLM accesses are correctly treated as out-of-bounds but the out-of-bounds data returned may not be zero.
3D	3D Sampler Message Types		If Surface Format is R10G10B10_SNORM_A2_UNORM and Gather4 Source Channel Select is alpha channel, the returned value may be incorrect.
3D	Programming Media Pipeline - Command Sequence		A MEDIA_STATE_FLUSH needs to be placed right before the MI_BATCH_BUFFER_END of any batch buffer that uses MEDIA_OBJECT.
3D	Depth Buffer Clear		To fast depth clear a full 16k, in the X and/or Y dimensions, the clear operation must be sectioned into rectangles smaller in X and Y than 16k pixels. For example to clear a 16k x 16k surface perform four (2x2) 4k x 4k clears with the proper address offsets.

3D	LOD Information		The LOD is in-bounds if LOD < MIPCount and if MinLOD + LOD < 15. If LOD is not in-bounds then 0 is returned for the width, height, and depth values.
3D		WaClearArfDependenciesBeforeEot	BattleForge3 hang - flag register dependency not cleared after EOT. WA: Source ARF registers before EOT.
Blitter	BLT Programming Restrictions		There are two suggested software workarounds to perform coherent overlapping BLTs: (a) The Source and Destination Base Address registers must hold the same value (without alignment restriction) and (b) the Source and Destination Pitch registers (BR11, BR13) must both be a multiple of 64 bytes. Or if (a) is not possible do overlapping source copy BLTs as two blits, using a separate intermediate surface.
Blitter		WaUse3dBlitForFBCMMIOProgramming	In this hang, BCS is executing LRI to an external GT address (address 50380, data 00000002). So instead of sending the cycle through the message channel, it will go through GAB. At the same time, BCS receives a GO=0 message from PM and this will bring the GO flag down which in turn will block the memory cycle from going out. The deadlock here is that the same state machine (in CSCFG) drives the GO ACK response and the external GT message.
Display	Clocks		CDCLK frequency change sequence needs added steps to prevent clock glitches. See North Display Engine Registers, CDCLK Sequence for details.
Display	Clocks		Very intermittent chance of failure of the display spread/bent clock reference. WA: Adjust the display spread/bent clock reference programming sequence when enabling/disabling HDMI, DVI, and DisplayPort. See ICLKIP Programming.
Display	Clocks		The PCH display clock stops when BIOS enables PCH ISCLK PLL shutdown feature, causing backlight flicker and other problems. WA: When the ISCLK PLL shutdown feature is enabled, set a clock gate disable C2020h bit 12 before using Aux channel B/C/D, GMBUS, GTC, or panel power sequencing. See South Display Engine Registers for details.

Display	Display		VGA may lockup and give a black screen randomly with some memory configs and/or memory tests. WA: See VGA_CONTROL for workaround bits since they change with project and stepping.
Display	Display		HDMI and DVI with audio are not supported when HSYNC Start is programmed equal to HBLANK Start.
Display	Display		FDI failures after training causes blank screen or hang. WA: Transcoder timing generator override bits must be set and cleared at certain point when enabling/disabling PCH transcoder. See BSpec Mode Set Sequences for exact workaround details.
Display	DisplayPort		There could be a rare case where the eDP modeset does not happen correctly if the link is already in "send normal pixels" when the pip e is enabled. WA: For eDP, do not set Normal Pixel at the end of Link Training. See DisplayPort Enable Sequence.
Display	DisplayPort		DP MST output incorrect for certain M and N and VC payload size values. WA: VC payload must be multiple of 4 in x1 lane config, 2 in x2, 1 in x4. See M/N Values.
Display	DisplayPort		DDIA Aux channel transactions get intermittent NAK errors with some receivers. WA: Increase DDI_AUX_CTL_A bits 27:26 Time out timer value to 600us 01b when doing DDIA aux transactions.
Display	DisplayPort		Big FIFO mode conflicts with enabling of DP Port Sync mode. WA: Set 0x45280 bits 2:1=11b before enabling port sync mode. See DisplayPort Enable Sequence.
Display	DisplayPort		Chance of Aux Channel command failures when using Lynxpoint:H. WA: For the PCH Aux channels (Aux B/C/D) use an aux divider value of 63 decimal (03Fh). If there is a failure, retry at least three times with 63, then retry at least three times with 72 decimal (048h). See South Display Engine Registers, DP_AUX_CTL.

Display	FDI		FDI Rx delay default value incorrect, causing FDI failures. WA: Set FDI delay to 90h before enabling FDI.
Display	FDI		FDI Rx TP1 time default value incorrect, causing FDI training failures. WA: Set TP1 to TP2 time to 48 clocks before enabling.
Display	FDI		Added programming needed to reset FDI mPHY IOSF-SB during mode set sequences. WA: See Mode Set Sequences for details.
Display	FDI		Added programming needed to control FDI Rx Pwrdn bits during CRT mode set sequences. WA: See Mode Set Sequences for details.
Display	FDI		Added programming needed to setup FDI mPHY registers at boot and in mode set sequences. WA: See Mode Set Sequences.
Display	General		TRANS_CONF Transcoder State may incorrectly show it is enabled before the transcoder has been enabled for the first time after reset or power well disable and enabled. This should not impact the normal usage of this field during the mode set disable sequence.
Power	IPS	WaIpsWaitForPcodeOnDisable	It can take up to 42ms for pcode to complete IPS disabling. The driver sequence to disable IPS needs to account for this delay before disabling the last plane on the pipe, otherwise there will be corruption from disabling planes before IPS is completely disabled. See North Display Engine Registers, Intermediate Pixel Storage for details.
Power	IPS		Chance of screen corruption if IPS and display planes are not enabled and disabled in a specific sequence. WA: IPS cannot be enabled until at least one plane has been enabled for at least one vertical blank. IPS must be disabled while there is still at least one plane enabled. See North Display Engine Registers, Intermediate Pixel Storage.

Display	Panel fitter	WaPanelFitterDownscale	When using panel fitter downscaling (pipe source size is larger than panel fitter window size) the maximum supported pixel rate will be reduced by the downscale amount, and watermarks must be adjusted. Use panel fitter scale amount when calculating maximum pixel rate and watermarks.
Display	Panel power sequencing	WaVDDOverrideT4Power	When software clears the panel power sequencing VDD override bit from 1 to 0 (disable VDD override) it must ensure that T4 power cycle delay is met before setting the bit to 1 again, else panel ma y be damaged. WA: Use software timers to ensure T4 delay is met or use full panel power enable and not the VDD override.
Display	Power Off Sequence		On CDV when using Self Refresh mode and one pipe is active while the second pipe is being enabled or disabled, the workaround sequence below needs to be followed. This does not need to done when both pipes are being enabled or disabled at the same time. When turning off a second pipe: Disable second pipe using the power off sequence as normal. Anytime after the step in the disable sequence where there is a wait for pipe off status, re-enable Self Refresh if desired (register 0x20E0 bit 15 set to 1).
Display	Power On Sequence		On CDV when using Self Refresh mode and one pipe is active while the second pipe is being enabled or disabled, the workaround sequence below needs to be followed. This does not need to done when both pipes are being enabled or disabled at the same time. When turning on a second pipe: Disable Self Refresh (register 0x20E0 bit 15 set to 0). Wait for vblank from first pipe (frame is ended and the pipe will use only his part at the ddbm during next frames) Enable second pipe using the mode switch enable sequence as normal.
Power	PSR		PSR single frame update with sprite - Mask the sprite enabled when using single frame update with sprite. See North Display Engine Registers, SRD_CTL register for details.

Power	PSR		PSR single frame update - Mask register write events when using single frame update. See North Display Engine Registers SRD_CTL register for details.
KMD, Media		WaVeboxSliceEnable	Workaround required for checking Single Slice Enable flag in VEBOX State.
KMD, Media	Decode	WaJPEGHeightAlignYUV422H2YToNV12	Specific to JPEG decode. If the input is YUV422H_2Y and output is NV12 format, output picture height should be aligned by 16 bytes.
KMD, Media	Encode	WaAddMediaStateFlushCmd	Add Media State Flush command after Media Object Walker command. This WA is for a preemption hang, if the last command in the media batch buffer isn't a media-state-flush then a pre-emption on the following GPGPU batch buffer can cause a hang. A Media_State_flush command is required to be added at the end of media command buffer or the batch buffer with MEDIA_OBJECT/WALKER. Required for RCS engine in media pipeline.
KMD, Media	Encode	WaEnableVMEReferenceWindowCheck	WA for limiting the minimum downscaled surface dimensions to 48x48.
KMD, Media	HuC	WaGetBits	When doing a getbits greater than 8 bits, or doing an exponential golomb read (signed or unsigned), and there is a startcode at a 16 byte aligned address, and start code detection is enabled and the current location in the barrel shifter plus the number of bits to shift is a multiple of 8, the barrel shifter may shift 1 time too many causing the loss of a byte. For example assume the stream is stopped and these are the values in the stream: 0000 0106 0102 f08c 8000 0000 0125. The first 8 bytes are in the barrel shifter (from left: 00 00 01 06 01 02 f0 8c). Doing a getbits24 will cause the 00 00 01 bytes to be removed, and (unfortunately) the 06 byte will get shifted out of barrel shifter and lost. The SW WA is to only use getbits8 or less (so instead of getbits24, do 3 getbits8). And to do exponential golomb decodes manually in FW, not by a read of the registers at 0xFF00_0084 and 0xFF00_0088.
KMD, Media	HuC	WaHucStreamoutEnable	To prevent corruption in the case where a HUC workload with bitstream streamout DISABLE followed by another HUC workload with bitstream streamout ENABLE, a dummy HUC workload (bitstream streamout needs to be set ENABLE) is inserted directly before the HUC workload with bitstream ENABLE.
KMD, Media	Media (Decode)	WaInsertAVCFrameForFormatSwitchToJPEG	Insert a dummy AVC frame before a JPEG frame to WA format switch issue when Decode format is switched to JPEG.

Memory Views	Graphics Memory Address Type: GSM		Due to a workaround, first 4KB of DSM has to be reserved for GFX hardware use during render engine execution.
			Alternative W/A: There is another possibility (100% effective) via disabling the 2nd port accesses to URB for FF clients. This w/a is not suggested as it will limit the vertex rate to 1 vertex/cycle in GT3. Set x2090[7]="1"
Memory Views	Memory Interface		32b PPGTT has no implications
Memory Views	Opaque Textures (DXT1_RGB)		The behavior of this format is not compliant with the OGL spec. As a workaround, the Surface Format should be set to BC1 and the Shader Channel Select A should be set to SCS_ONE.
Power	DPST		The DPST Histogram Event status may not clear if it is written with a 1b to clear it and the Histogram Interrupt enable field is changed from 0b to 1b in the same MMIO write. WA: To guarantee the event status is cleared, separate the single MMIO write into two writes.
Power	FBC		First line of FBC getting corrupted when FBC compressed frame buffer offset is programmed to zero. Command streamers are doing flush writes to base of stolen. WA: New restriction to program FBC compressed frame buffer offset to at least 4KB. See FBC_CFB_BASE in North Display Registers. Additionally the driver will need to avoid using this first 4KB of graphics data stolen memory for anything else.
Power	IPS	WaIpsDisableOnAsyncFlips	Enabling IPS increases the flip completion time results in lower frame rates or performance. WA: Disable IPS for Async flips.
Power	IPS	WaIpsUpdateOnPlaneStatusChang e	Programming requirement: IPS to be enabled only after display plane is enabled. WA: IPS cannot be enabled until at least one plane has been enabled for at least one vertical blank. IPS must be disabled while there is still at least one plane enabled. See North Display Engine Registers Intermediate Pixel Storage.

Power	Turbo		Downward busyness interrupt not working when RC6 is enabled. Symptoms include missing interrupts making the feature generally unreliable. WA: SW should use another mechanism to decide when to requires a change in GT frequency. A suggestion would be to use PP1_C0_CORE_CLOCK as busyness counter. The register PUSHBUS_CONTROL - Push Bus Metric Control, MMIO: 0/2/0, Address 0A248h, bit 2, C0 Residency Time Enable controls this capability. It can be used to report GT activity that can then be used to determine how active GT has been over a given time period.
Power	RC6	WaRsUseTimeoutMode	EI counters are not reliable through RC6 entry / exit, and the resulting behavior effectively causes counters to lose time spent in RC6. WA: Use hardware RC6 Timeout Mode to replace hardware RC6 EI Mode to meet RC6 duty cycle targets. By setting the promotion time value to 800 microseconds, the RC timeout mode provides reasonable power / performance. The value may need to be tuned for specific workloads / systems. 1. Set RC_CTRL0 - Render C State Control 1, MMIO: 0/2/0, Address: 0A090h, bits 27:28 to 01b (time out mode). 2. Set RC_PROMO_TIME1 - RC Promotion Timer for RC6, MMIO: 0/2/0, Address: 0A0b8h to 0x271. (800 microsecond promotion timer value).
Power	PSR	WaMaskVBIduringSFU	PSR SFU not working during media playback. WA: Set PIPE_MISC MMIO 70030h bit20 to 1, masking VBI tracking during SFU enabled scenario.
Power	IPS	WaIpsDisableOnCdClockThreshold	WA:Disable IPS when dotclock > 95% of cdclk
Display	eDP	N/A	eDP link training failures if package C6 is entered when display on/off power well is disabled. The PCU thinks eDP is completely disabled and the display LCPLL can be shutoff, but eDP is actually partially enabled and needs the LCPLL. WA:Program SRD_CTL_EDP Max Sleep Time to 0 before starting eDP link training. This will tell the PCU to not turn off the LCPLL. Restore SRD_CTL_EDP Max Sleep Time after enabling TRANS_CONF_EDP.

Display	Dither	N/A	Dither pattern stops as soon as the transcoder configuration register is written to disable transcoder. Dither should continue until the vblank when transcoder actually disables. WA: A workaround is only needed if it is important to have a correct dither pattern in the final frame as display output is disabled. Any workaround must be specific to the implementation of the operating system and driver.
KMD		WaIdleLiteRestore	SW must always ensure ring buffer head pointer is not equal to tail pointer of a context, whenever it is submitted to HW for execution.
KMD	Mid-batch Preemption	WaDisableCtxRestoreArbitration	[Render CS Only][Execlist Mode of Scheduling]: SW must ensure arbitration is switched off while context restore is in progress for any given context. This is achieved by disabling arbitration by programming MI_ARB_ON_OFF to "Arbitration Disable" in RCS_INDIRECT_CTX buffer and by enabling back the arbitration by programming MI_ARB_ON_OFF to "Arbitration Enable" as the last command prior to MI_BATCH_END in the BB_PER_CTX_PTR buffer of every context submitted. Note that RCS_INDIRECT_CTX_OFFSET could be set to default value or any other legitimate value as per the programming notes of the register definition. • Arbitration disable by programming MI_ARB_ON_OFF (Arbitration Disabled) in RCS_INDIRECT_CTX buffer. • Arbitration enabled by programming MI_ARB_ON_OFF (Arbitration Enabled) as the last command prior to MI_BATCH_BUFFER_END in BB_PER_CTX_PTR buffer.
Media	HUC	N/A	When the bit stream ends in just the sc prefix (0x00 0x00 0x01) and no start code, if FW stops there then does a fast forward, the HUC_BS_COUNTER1 value will increment by 32 bits instead of 24 bits. When this case is encountered, at the end of a stream, the FW should read the HUC_BSD_SCD_STATUS register and if bit 1 is set, then FW should deduct 1 from the HUC_BS_STREAM_OBJ_COUNTER1 value.
Blitter		WaUse3DBlitForNonCLAlignedLinearSrcDst	Color source and destinations must be restricted to CL alignment for linear addressing.

Media	HuC	WaStreamoutFlush	There is a firmware workaround required whenever streamout is enabled. Listing it down, to ensure FW has implemented it in a similar way. This would only impact widevine (since streamOut is enabled) 1. After bitstream processing, FW needs to issue a fast forward to end of stream, and wait for fast forward operation to complete 2. FW issues a write followed by read transaction to memory 3. FW issues a flush operation, and polls on BS status register for flush to complete FW clears all bits in BS status register
3D		WaGT3ResendDepthStencilAndHierDepthBufferState	Issue is BSPEC specifies X Max and Y Max are "exclusive" when they need to be "inclusive". Here is the BSPEC note for Clear Rectangle X Max in 3DSTATE_WM_HZ_OP: 'Specifies Xmax value of (exclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with X coordinates greater than or equal to Xmax will be not be affected.'
Power	FBC		FBC may intermittently fail to update some lines after an image change, causing stale data to appear. The stale data can be prevented by manually invalidating FBC after flips and enabling FBC. Manual invalidation sequence: For (i=0 to 127) { Write 0x50344 = (i<<16) } The manual invalidation needs to happen in the frame that the image changes or FBC enables, between the start of vertical blank at the beginning of that frame and the start of vertical blank at the end of that frame. Recommended sequence: After an async flip, sync flip, or enabling FBC, wait for the next start of vblank (sync flips can alternatively wait for flip done), then perform the manual invalidation sequence.
KMD		WaSkipStolenMemoryFirstPage	WA to skip the first page of stolen memory due to sporadic HW write on *CS Idle.
KMD		WaForceContextSaveRestoreNonCoherent	To avoid a potential hang condition with TLB invalidation driver should enable masked bit 5 of MMIO 0x7300 at boot.

GMM		WaIommuCCInvalidationHang	When using IOMMU, IOTLB invalidation followed by a Context Cache invalidation can cause a hang. Workaround is to add a NOP (Device TLB invalidation is considered a NOP by HW) everytime before doing a context cache invalidation. This is done by writing dev_iotlb_inv_dsc to invalidation queue OR writing to scratch register for register based invalidation.(Scratch register[0x4F104] used for invalidation queue workaround(IommuUncoreUnavailable)) - Refer Bspec for descriptor format - Search "dev_iotlb_inv_dsc".
GMM		Wa32bitGeneralStateOffset	Allocations of type Scratchflat/Instructionflat can only be referenced by 32 bit offset in a Graphics Segment. When Graphics memory size is >2GB, 32 bit offsets cannot address the 64 bit aperture segment. This workaround is used to define 32bit Segment to be used only for ScracthFlat/InstructionFlat heaps in >2GB configurations.
GMM		Wa32bitInstructionBaseOffset	Allocations of type Scratchflat/Instructionflat can only be referenced by 32 bit offset in a Graphics Segment. When Graphics memory size is >2GB, 32 bit offsets cannot address the 64 bit aperture segment. This workaround is used to define 32bit Segment to be used only for ScracthFlat/InstructionFlat heaps in >2GB configurations.
KMD		WaForceEnableNonCoherent	Must Force Non-Coherent whenever executing a 3D context. This is a workaround for a possible hang in the unlikely event a TLB invalidation occurs during a PSD flush. Set masked bit 4 in 0x7300 during boot.
KMD		WaDisableSRGBGammaToLinearConv	WA is to avoid Gamma to Linear conversion when source surface is SRGB and dest is non SRGB as per DX10's Present DDI expectation. In such a case, treat source as non-SRGB that will do bit-for-bit transfer from source to dest. E.g. if source format is B8G8R8A8_UNORM_SRGB and destination format is non-srgb then treat source as B8G8R8A8_UNORM instead by programming the surface state accordingly.
KMD		WaClearRenderResponseMasks	Need to set the correct render response masks to save power. HW power-on default is not good. Applies only to unmask primary plane flip and sprite flip done on A, B and C pipes. Done by clearing bit 1,2, 9, 10, 15 and 20 of MMIO 0x44050. Must be done at boot and all save/restore paths.
KMD		Wa4x4STCOptimizationDisable	HIZ/STC hang in hawx frames. W/A: Disable 4x4 RCPFE-STC optimization and therefore only send one valid 4x4 to STC on 4x4 interface. This will require setting bit 6 of reg. 0x7004. Must be done at boot and all save/restore paths.

KMD		WaModifyVFEStateAfterGPGPUPreemption	GPGPU preemption hang or corruption issue. This WA must be applied before re-submitting a GPGPU preempted workload. SW needs to do two things. 1) SW should detect if VFE unit in the context image have LRI commands to 0x54AC and 0x54B0. SW does that by checking if the LRI header at context image offste (0x0E90) <in VFE> is 0x11001003 and the two MMIO offsets in this LRI is 0x54AC and 0x54B0. If so, do this- D0 (LRI header): 0x1100_1003 -> 0x1100_ 1001 D1 (1st MMIO offset): 0x0000_54AC -> Leave this dword untouched D2 (1st MMIO offset value): 54AC_DATA -> Leave this dword untouched D3 (2nd MMIO offset): 0x0000_54B0 -> Overwrite to 0x0 D4 (2nd MMIO offset value): 54B0_DATA -> Overwrite to 0x0 2) Apart from it SW must also increment the Batch Buffer Address by Walker Command Length so that CS moves to the command following it. Size of Walker should be 0xF * sizeof(DWORD) = 0x3C Bytes.
KMD	RTL	WaBlockMsgChannelDuringGfxReset	When GPU hangs occurs, inside gfx driver TDR routine the driver writes to the reset register and after that it tries to read back the reset register and GPU never response back so system crashes. As a WA set 0x9424 bits 2,3,4,5,6,9 on boot.
KMD	RTL	WaFlushCoherentL3CacheLinesAtContextSwitch	Coherent L3 cache lines are not getting flushed during context switch which is causing issues like corruption. Need to set bit 21 of MMIO b118, then send PC with DC flush and then reset bit 21 of b118. This programming sequence needs to be part of the indirect context WA BB.
KMD		WaDisableSDEUnitClockGating	WA for GPGPU workload hang for which requirement is to disable SDE Unit clock gating. This is done by setting bit 14 of MMIO 9430.
KMD		WaIncreaseTagClockTimer	Clock gating causes wrong read adresss from State Array during RO invalidation causing Mem-Diff. So WA requires programing L3 tag clock timing register. At boot, write 0xB10Ch, bits [23:20] programmed to 1000b
KMD		WaDisableLiteRestore	There is a bug that can surface in Lite-Restore scenarios causing a hang. Driver needs to prevent lite restore scenarios from occuring. There are many ways to do this. 1) Driver can always set the 'Force Full Restore' bit in the context desciptor when writing to the submit port. 2) The driver can wait for the Active->Idle interrupt before submitting any new exeuction lists.
KMD		WaSetVfGuardbandPreemptionVertexCount	Workaround for potential 3d preemption bug that can cause data corruption. Driver should write to register 0x83A4 (preemption vertex count) and set a value of 0x20. At boot, write 0xffff0020 to 0x83a4 (it’s a masked register).

KMD		WaProgramL3SqcReg1Default	Program the default initial value of the L3SqcReg1 for performance. This requires writing 0x784000 to MMIO b100.
KMD		WaSkipInvalidSubmitsFromOS	For Invalid submits from OS - simply report fence completion without submitting the DMA buffer to GPU.
KMD		WaTempDisableDOPClkGating	This WA is associated with another WA WaProgramL3SqcReg1Default (which requires doing a cpu write to 0xb100 offset with value 0x784000). Before applying WaProgramL3SqcReg1Default WA, SW should temporarily force disable render GOP clock gating by setting MMIO 0x9424 [bit 0] to 0. After WaProgramL3SqcReg1Default has been applied, S/W should restore 0x9424 with its old value.
3D		WaSendDummyVFEafterPipelineSelect	TSG unit writes null entries into context which can hang restore CS and TSG. WA: Send dummy VFE immediately after GPGPU pipeline select. We dont use Generic MediaStateClear at all.
3D			For indirect dispatch there is handle counter underflow and it leads to hang. WA: GPGPU walker need to use only curbe while doing indirect dispatch.
3D			Draw call doesn't send verticies to compute shader because 0-vertex draw call, preempted. Handful of resons that VS would send no verticies. WA: Pipe control with CS stall only when the UAV is using a vertex shader.
3D		WaClearCCStatePriorPipelineSelect	Architecture hole; on GPGPU context restore, at the end of the context when CS sends a null prim, SVG and SARB does a state prefetch; by the time the data returns from memory, CS gates the FF clock. WA: In GPGPU mode, color cal state should not have valid bits. Before switching pipelines, send null CC state pointers.
3D			FF clocks will be gated, EU kernels are requesting from render cache to have writes from memory. WA: Disbale FF CLK gating when GPGPU Kernel require RC access.
3D		WaAvoidPMAStall	PMA bit is getting turned on when color traffic is in flight in RCPFE. WA: Put color flush before changing the PMA bit before stencil write cases.
3D			Increase the size of sub register from 8 to 16. For MOVI, did not change the code to accommodate. As a result, SW is not able to use this. WA: Use 2 SIMD16 instructions instead of 1 SIMD32. Will have slight performance hit.

3D			Write collision hang condition. When we have both remainder and quotient in same instruction, had write port collision on GRF, drop the dependency clear resulting in hang. WA: Break integer divide into separate quotient and remainder instructions.
3D			When DCS re-allocates handle ID, SDE flops but valids are not used in clock wakeup equations. When reallocation happens, handles lost. WA: Disable SDEunit clock gating. UCGCTL6[14]=1 (09430h[14])
3D			When we divide and denominator is DNORM, and DNORM is enalbed, should produce infinity, but result is 0. Works OK if DNORMs are disabled, legacy features are fine. WA: Don't use DNORM on FDIV.
3D			GA is sending wrong data to fpu for src0 indirect addressing WA:When an instruction has a source region spanning two registers and a destination regioning contained in one register, one of the following must be true: The destination region is entirely contained in the lower Oword of a register. The destination region is entirely contained in the upper Oword of a register. The destination elements are evenly split between the two OWords of a register AND evenly split between the two source registers.
3D			MsaaBasic hang with single subspan dispatch enabled. WA: QtrCtrl must not be used for jmpi instruction.
3D			DF --> f format conversion for align16 has wrong emask calculation when the source is immediate. WA: In Align16 mode, format conversion from double-float to floats is not allowed when source is immediate data.
3D			SMOV instruction with byte data type. If this is used, corruption. Data read / write is not as expected. WA: Disable SMOV with byte.

3D		WaDisableObjectLevelPreemtionForVertexCount	Preemption the clock after the last vertex of draw call with less verticies than in the guardband, VF will indicate that it's incomplete, when it is actually compelte. WIll redraw the last instance. WA: For regular draw calls, when # od verticies is less than the GB, dissble mid-object preemption.
3D		WaKVMNotificationOnConfigChange	DPR currently sends a "decreg" signal to DPCEUNIT for TRANS_CONF_EDP. The signal is expected only to pulse when a write on TRANS_CONF_EDP has occurred, but the signal is actually assigned to logic directly out of the automated register code which is read and write accesses.
3D		WaForceDX10BorderColorFor64BPTTextures	Dgunit sending wrong border color data for DX9 and 64 bpt. WA: Force xmajor for 64bpt non-aniso in DX9 mode.