1.9.14.2 HS_HORIZ ONTAL 0

Bit 7 0

0,0 7,0

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

1.9.14.3 HS_VERTI CAL 1

Bit 7 0

0,0 7,0

0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0

1.9.14.4 HS_F DIAGONAL 2

Bit 7 0

0,0 7,0

1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

1.9.14.5 HS_BDIAGONAL 3

Bit 7 0

0,0 7,0

0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

1.9.14.6 HS_CROSS 4

Bit 7 0
0,0 7,0

0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
1	1	1	1	1	1	1	1
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0

1.9.14.7 HS_DIAG CROSS 5

Bit 7 0
0,0 7,0

1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	1

1.9.14.8 Screen Door 8

Bit 7 0
0,0 7,0

0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0

1.9.14.9 SD Wide 9

Bit 7 0
0,0 7,0

1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1

1.9.14.10 Walking Bit (One) A

Bit 7 0
0,0 7,0

1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1

1.9.14.11 Walking Zero B

Bit 7 0
0,0 7,0

0	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
0	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0

1.9.15 XY_SRC_COPY_BLT

This BLT instruction performs a color source copy where the only operands involved is a color source and destination of the same bit width.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is **less than** Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is **less than** Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The ROP value chosen must involve source and no pattern data in the ROP operation.

DWord Bit		Description
0 = BR00	31:29	**Client:** 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 53h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:16	**Reserved.**
	15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	14:12	**Reserved**
	11	**Dest Tiling Enable:** **0 = Tiling Disabled (Linear)1 = Tiling enabled** (Pre-DevSNB : Tile-X only.)
	10: 8	**Reserved**
	7:0	**Dword Length:** 06h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)

DWord Bit		Description
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Dest Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.
5 = BR26	31:16	**Source Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Source X1 Coordinate (Left):** (15:00 = 16 bit signed number)
6 = BR11	31:16	**Reserved**
	15:00	**Source Pitch (double word aligned) and in DWords:** [15:00] 2's complement. For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
7 = BR12	31:00	**Source Base Address:** (base address of the source surface: X=0, Y=0) When Src Tiling is enabled (Bit 15 enabled), this address is limited to 4Kbytes.

1.9.16 XY_ SRC_COPY_CHROMA_BLT

This BLT instruction performs a color source copy with chroma-keying where the only operands involved is a color source and destination of the same bit width.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is **less than** Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is **less than** Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The ROP value chosen must involve source and no pattern data in the ROP operation.

DWord Bit		Description
0 = BR00	31:29	**Client:** 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 73h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:17	**Transparency Range Mode:** (chroma-key)
	16	**Reserved**
	15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	14:12	**Reserved**
	11	**Dest Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10: 08	**Reserved**

DWord Bit		Description
	07:00	**Dword Length:** 08h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Dest Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.
5 = BR26	31:16	**Source Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Source X1 Coordinate (Left):** (15:00 = 16 bit signed number)
6 = BR11	31:16	**Reserved.**
	15:00	**Source Pitch (double word aligned) and in DWords:** [15:00] 2's complement. For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
7 = BR12	31:00	**Source Base Address:** (base address of the source surface: X=0, Y=0) When Src Tiling is enabled (Bit 15 enabled), this address is limited to 4Kbytes.
8 = BR18	31:00	**Transparency Color Low:** (Chroma-key Low = Pixel Greater or Equal)
9 = BR19	31:00	**Transparency Color High:** (Chroma-key High = Pixel Less or Equal)

1.9.17 XY_MONO_SRC_COPY_BLT

This BLT instruction performs a monochrome source copy where the only operands involved is a monochrome source and destination. The source and destination operands cannot overlap therefore the X and Y directions are always forward.

All non-text monochrome sources are word aligned. At the end of a scan line of monochrome source, all bits until the next word boundary are ignored. The monochrome source data bit position field [2:0] indicates the bit position within the first byte of the scan line that should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation. The ROP value chosen must involve source and no pattern data in the ROP operation.
Negative Stride (= Pitch) is NOT ALLOWED.

DWord Bit		Description
0 = BR00	31:29	**Client:** 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 54h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:17	**Monochrome source data bit position of the first pixel within a byte per scan line.**
	16:12	**Reserved.**
	11	**Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10: 08	**Reserved**
	07:00	**Doubleword Length:** 06h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29	**Mono Source Transparency Mode:** (1 = transparency enabled; 0 = use background)
	28:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)

DWord Bit		Description
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.
5 = BR12	31:00	**Source Address:** (address corresponding to DST X1,Y1) (Note no NPO2 change here)
6 = BR18	31:00	**Source Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
7 = BR19	31:00	**Source Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]

1.9.18 XY_MONO_SRC_COPY_ IMMEDIATE_BLT

This instruction allows the Driver to send monochrome data through the instruction stream, eliminating the read latency of the source during command execution.

The IMMEDIATE_BLT data MUST transfer an even number of doublewords and the exact number of quadwords.

All non-text monochrome sources are word aligned. At the end of a scan line of monochrome source, all bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates the bit position within the first byte of the scan line that should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation. The ROP value chosen must involve source and no pattern data in the ROP operation.

The monochrome source data supplied corresponds to the Destination X1 and Y1 coordinates.

Negative Stride (= Pitch) is NOT ALLOWED.

DWord Bit		Description
0 = BR00	31:29	**Client:** 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 71h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:17	**Monochrome source data bit position of the first pixel within a byte per scan line.**
	16:12	**Reserved.**
	11	**Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10: 08	**Reserved**
	07:00	**Dword Length:** 05+ DWL = (Number of Immediate double words)h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29	**Mono Source Transparency Mode:** (1 = transparency enabled; 0 = use background)
	28:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.
5 = BR18	31:00	**Source Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
6 = BR19	31:00	**Source Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
7	31:00	**Immediate Data DW 0:**
8	31:00	**Immediate Data DW 1:**
9 thru DWL+4	S	**Immediate Data DWs 2 through DWORD_LENGTH (DWL):**

1.9.19 XY_FULL_BLT

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source and pattern operands are the same bit width as the destination operand.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is **less than** Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is **less than** Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord Bit		Description
0 = BR00	31:29	**Client:** 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 55h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:16	**Reserved.**
	15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	14:12	**Pattern Horizontal Seed** (pixel of the scan line to start on corresponding to DST X=0)
	11	**Dest Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:08	**Pattern Vertical Seed:** (starting scan line of the 8x8 pattern corresponding to DST Y=0)
	07:00	**Doubleword Length:** 07h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**

DWord Bit		Description
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Dest Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.
5 = BR11	31:16	**Reserved.**
	15:00	**Source Pitch (double word aligned and signed) and in DWords:** [15:00] 2's complement. For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
6 = BR26	31:16	**Source Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Source X1 Coordinate (Left):** (15:00 = 16 bit signed number)
7 = BR12	31:00	**Source Base Address:** (base address of the source surface: X=0, Y=0) When Src Tiling is enabled (Bit 15 enabled), this address is limited to 4Kbytes.
8 = BR15	31:00	**Pattern Base Address:** (28:06 are implemented) (Note no NPO2 change here). The pattern data must be located in linear memory.

1.9.20 XY_FULL_IMMEDIATE_PATTERN_BLT

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source and immediate pattern operands are the same bit width as the destination operand. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64 DWs) for 8, 16, and 32 bpp color patterns.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is **less than** Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is **less than** Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord Bit		Description
0 = BR00	31:29	**Client**: 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 74h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:16	**Reserved.**
	15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	14:12	**Pattern Horizontal Seed:** (pixel of the scan line to start on corresponding to DST X=0)
	11	**Dest Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:8	**Pattern Vertical Seed:** (starting scan line of the 8x8 pattern corresponding to DST Y=0)
	7:0	**Doubleword Length:** 06+ DWL = (Number of Immediate double words)h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Dest Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.
5 = BR11	31:16	**Reserved.**
	15:00	**Source Pitch (double word aligned and signed) and in DWords:** [15:00] 2's complement. For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
6 = BR26	31:16	**Source Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Source X1 Coordinate (Left):** (15:00 = 16 bit signed number)

DWord Bit		Description
7 = BR12	31:00	**Source Base Address:** (base address of the source surface: X=0, Y=0) When Src Tiling is enabled (Bit 15 enabled), this address is limited to 4Kbytes.
8	31:00	**Immediate Data DW 0:**
9	31:00	**Immediate Data DW 1:**
A thru DWL+4	S	**Immediate Data DWs 2 through DWORD_LENGTH (DWL):**

1.9.21 XY_FULL_MONO_SRC_BLT

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source operand is monochrome and the pattern operand is the same bit width as the destination.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text and non-immediate monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the Destination X1 coordinate.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Negative Stride (= Pitch) is NOT ALLOWED

DWord Bit		Description
0 = BR00	31:29	**Client**: 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 56h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:17	**Monochrome source data bit position of the first pixel within a byte per scan line.**
	16:15	**Reserved.**
	14:12	**Pattern Horizontal Seed:** (pixel of the scan line to start on corresponding to DST X=0)
	11	**Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:08	**Pattern Vertical Seed:** (starting address of the 8x8 pattern corresponding to DST Y=0)
	07:00	**Doubleword Length :** 07h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29	**Mono Source Transparency Mode:** (1 = transparency enabled; 0 = use background)
	28:27	**Reserved.**
	26	**Reserved.**

DWord Bit		Description
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.
5 = BR12	31:00	**Mono Source Address:** (address corresponds to DST X1, Y1) (Note no NPO2 change here)
6 = BR18	31:00	**Source Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
7 = BR19	31:00	**Source Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
8 = BR15	31:00	**Pattern Base Address:** (28:06 are implemented **)** (Note no NPO2 change here). The pattern data must be located in linear memory.

1.9.22 XY_FULL_MONO_SRC_ IMMEDIATE_PATTERN_BLT

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source operand is a monochrome and the immediate pattern operand is the same bit width as the destination. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64DWs) for 8, 16, and 32 bpp color patterns.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the destination X1 coordinate.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Negative Stride (= Pitch) is NOT ALLOWED.

DWord Bit		Description
0 = BR00	31:29	**Client**: 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 75h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:17	**Monochrome source data bit position of the first pixel within a byte per scan line.**
	16:15	**Reserved.**
	14:12	**Pattern Horizontal Seed:** (pixel of the scan line to start on corresponding to DST X=0)
	11	**Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:08	**Pattern Vertical Seed:** (starting address of the 8x8 pattern corresponding to DST Y=0)
	07:00	**Doubleword Length :** 06+ DWL = (Number of Immediate double words)h
1 = BR13	31	**Reserved.**
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29	**Mono Source Transparency Mode:** (1 = transparency enabled; 0 = use background)
	28:26	**Reserved.**

DWord Bit		Description
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.
5 = BR12	31:00	**Mono Source Address:** (address corresponds to DST X1, Y1) (Note no NPO2 change here)
6 = BR18	31:00	**Source Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
7 = BR19	31:00	**Source Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
8	31:00	**Immediate Data DW 0:**
9	31:00	**Immediate Data DW 1:**
A thru DWL+4	S	**Immediate Data DWs 2 through DWORD_LENGTH (DWL):**

1.9.23 XY_FULL_MONO_PATTERN_BLT

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The pattern operand is monochrome and the source operand is the same bit width as the destination operand.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is **less than** Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is **less than** Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the pattern foreground color is used in the ROP operation.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Setting both Solid Pattern Select =1 & Mono Pattern Transparency = 1 is mutually exclusive. The device implementation results in NO PIXELs DRAWN.

DWord Bit		Description
0 = BR00	31:29	**Client**: 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 57h
	21:20	**32 bpp byte mask:** (21 =1= write alpha channel; 20=1= write RGB channels)
	19:16	**Reserved.**
	15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	14:12	**Pattern Horizontal Seed:** (pixel of the scan line to start on corresponding to DST X=0)
	11	**Dest Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:08	**Pattern Vertical Seed:** (starting scan line of the 8x8 pattern corresponding to DST Y=0)
	07:00	**Dword Length :** 0Ah
1 = BR13	31	**Solid Pattern Select:** (1 = solid pattern; 0 = no solid pattern)
	30	**Clipping Enable:** (1 = enabled; 0 = disabled)
	29	**Reserved.**
	28:27	**Mono Pattern Transparency Mode:** (1 = transparency enabled; 0 = use background)
	26	**Reserved.**

DWord Bit		Description
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color 10 = 16 bit color (1555) 11 = 32 bit color (565)
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Dest Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.
5 = BR11	31:16	**Reserved.**
	15:00	**Source Pitch (double word aligned and signed) and in DWords:** [15:00] 2's complement. For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
6 = BR26	31:16	**Source Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Source X1 Coordinate (Left):** (15:00 = 16 bit signed number)
7 = BR12	31:00	**Source Base Address:** (base address of the source surface: X=0, Y=0) When Src Tiling is enabled (Bit 15 enabled), this address is limited to 4Kbytes.
8 = BR16	31:00	**Pattern Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
9 = BR17	31:00	**Pattern Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
A = BR20	31:00	**Pattern Data 0:** (least significant DW)
B = BR21	31:00	**Pattern Data 1:** (most significant DW)

1.9.24 XY_FULL_MONO_ PATTERN_MONO_SRC_BLT

The full BLT provides the ability to specify all 3 operands: destination, source, and pattern. The pattern and source operands are monochrome.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the pattern is 0. When the source bit is 1, then the pattern foreground color is used in the ROP operation. The monochrome source transparency mode works identical to the pattern transparency mode.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Setting both Solid Pattern Select =1 & Mono Pattern Transparency = 1 is mutually exclusive. The device implementation results in NO PIXELs DRAWN.

Negative Stride (= Pitch) is NOT ALLOWED.

DWord Bit		Description
0 = BR00	31:29	**Client**: 02h - 2D Processor
	28:22	**Instruction Target (Opcode):** 58h
	21:20	**32 bpp byte mask:** (21 = 1 = write alpha channel; 20 = 1 = write RGB channels)
	19:17	**Monochrome source data bit position of the first pixel within a byte per scan line.**
	16:15	**Reserved.**
	14:12	**Pattern Horizontal Seed:** (pixel of the scan line to start on corresponding to DST X = 0)
	11	**Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
	10:08	**Pattern Vertical Seed:** (starting scan line of the 8x8 pattern corresponding to DST Y = 0)
	07:00	**Doubleword Length :** 0Ah
1 = BR13	31	**Solid Pattern Select:** (1 = solid pattern; 0 = no solid pattern)
	30	**Clipping Enable** (1 = enabled; 0 = disabled)
	29	**Mono Source Transparency Mode:** (1 = transparency enabled; 0 = use background)
	28	**Mono Pattern Transparency Mode:** (1 = transparency enabled; 0 = use background)

DWord Bit		Description
	27:26	**Reserved.**
	25:24	**Color Depth:** 00 = 8 bit color 01 = 16 bit color (565) 10 = 16 bit color (1555) 11 = 32 bit color
	23:16	**Raster Operation:**
	15:00	**Destination Pitch in DWords:** [15:00] 2's complement For Tiled surfaces (bit_11 enabled) this ptich is of 512Byte granularity FOR Tile-X, 128B granularity for Tile-Y, and can be upto 128Kbytes (or 32KDwords).
2 = BR22	31:16	**Destination Y1 Coordinate (Top):** (31:16 = 16 bit signed number)
	15:00	**Destination X1 Coordinate (Left):** (15:00 = 16 bit signed number)
3 = BR23	31:16	**Destination Y2 Coordinate (Bottom):** (31:16 = 16 bit signed number)
	15:00	**Destination X2 Coordinate (Right):** (15:00 = 16 bit signed number)
4 = BR09	31:00	**Destination Base Address:** (base address of the destination surface: X=0, Y=0) When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.
5 = BR12	31:00	**Source Address:** (address corresponding to Dst X1,Y1) (Note no NPO2 change here)
6 = BR18	31:00	**Source Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
7 = BR19	31:00	**Source Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
8 = BR16	31:00	**Pattern Background Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
9 = BR17	31:00	**Pattern Foreground Color:** 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
A =BR20	31:00	**Pattern Data 0:** (least significant DW)
B =BR21	31:00	**Pattern Data 1:** (most significant DW)

1.10 BLT Engine Instruction Field Definitions

This section describes the BLT Engine instruction fields. These descriptions are in the format of register descriptions. These registers are internal and are not readable. Some of these registers are state that is saved and restored for supporting separate software threads.

1.10.1 BR00—BLT Opcode & Control

Memory Offset Address: none
Default: 0000 0000
Attributes: not accessible

BR00 is the last executed instruction DWord 0. Bits [22:5] are written by every DW0 of every instruction. Bits [31:30] and [4:0] are status bits. Bits [28:27] are written from the DW0 [15:14] of a Setup instruction and Bit 29 is written with a 1 when ever a Setup instruction is written. Bit 29 is a decode of the Setup instruction Opcode.

31	30	29	28 ... 24
Rsvd	Clip Inst	Setup Mono Pattern	Instruction Target (Opcode)

23 ... 22	21 ... 20	19 ... 17	16
Instruction Target (Opcode)	32 bpp byte mask	Monochrome Source Start	Bit (0) / Byte (1) Packed

15	14 ... 12	11	10 ... 8
Rsvd	Pattern Horizontal Seed	Tiling Enable	Transparency Range Mode

7 ... 5	4	3	2	1	0
PatternVertical Seed	DST RMW	Color Source	Mono Source	Color Pattern	Mono Pattern

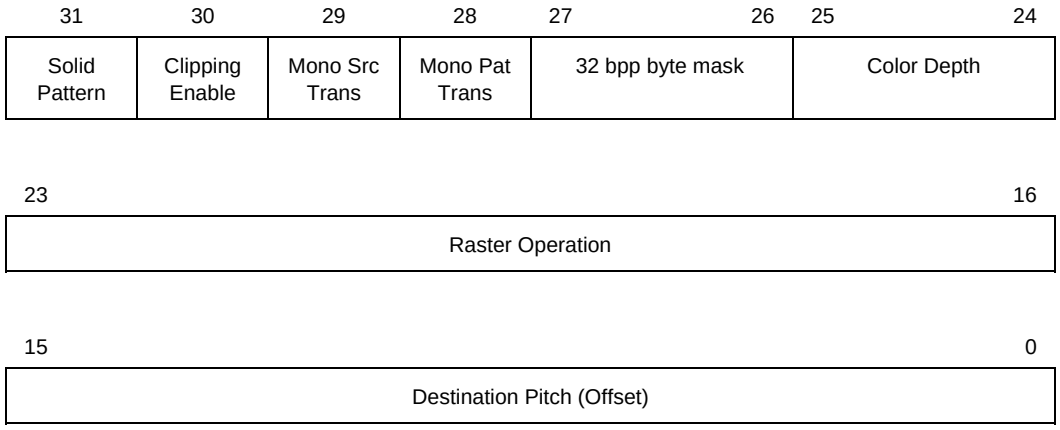
Bit De	scriptions
31	**BLT Engine Busy.** This bit indicates whether the BLT Engine is busy (1) or idle (0). This bit is replicated in the SETUP BLT Opcode & Control register. 1 = Busy 0 = Idle
30	**Setup Instruction Instruction.** The current instruction performs clipping (1).
29	**Setup Monochrome Pattern.** This bit is decoded from the Setup instruction opcode to identify whether a color (0) or monochrome (1) pattern is used with the SCANLINE_BLT instruction. 1 = Monochrome 0 = Color
28:22	**Instruction Target (Opcode).** This is the contents of the Instruction Target field from the last BLT instruction. This field is used by the BLT Engine state machine to identify the BLT instruction it is to perform. The opcode specifies whether the source and pattern operands are color or monochrome.
21:20	**32 bpp byte mask:** 21 = 1 = write alpha channel [31:24]; 20 = 1 = write RGB channels [23:00]. This field is only used for 32bpp.
19:17	**Monochrome Source Start.** This field indicates the starting monochrome pixel bit position within a byte per scan line of the source operand. The monochrome source is word aligned which means that at the end of the scan line all bits should be discarded until the next word boundary.
16	**Bit/Byte Packed .** Byte packed is for the NT driver 0 = Bit 1 = Byte
15	**Src Tiling Enable:** 0 = Tiling Disabled (Linear) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.)
14:12	**Horizontal Pattern Seed.** This field indicates the pattern pixel position which corresponds to X = 0.
11	**Dest Tiling Enable:** 0 = Tiling Disabled (Linear blit) 1 = Tiling enabled (Pre-DevSNB : Tile-X only.) When set to '1', this means that Blitter is executing in Tiled mode. If '0' it means that Blitter is in Linear mode. Pre-Dev Blitter never executes in Tiled-Y mode., DevGT+ Blitter supports both Tile-X and Tile-Y modes. On reset, this bit will be '0'. This definition applies to only X,Y Blits. Non-XY blits (COLOR_BLT, SRC_COPY_BLT), will support only linear mode and will not support tiling and for them this bit will remain reserved.

Bit De	scriptions
10:8	**Transparency Range Mode.** These bits control whether or not the byte(s) at the destination corresponding to a given pixel will be conditionally written, and what those conditions are. This feature can make it possible to perform various masking functions in order to selectively write or preserve graphics data already at the destination. XX0 = No color transparency mode enabled. This causes normal operation with regard to writing data to the destination. 001 = **[Source color transparency]** The **Transparency Color Low:** (Pixel Greater or Equal) (source background register) and the **Transparency Color High:** (Pixel Less or Equal) (source foreground register) are compared to the source pixels. The range comparisons are done on each component (R,G,B) and then logically ANDed. If the source pixel components are not within the range defined by the Transparency Color registers, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation. 011 = **[Source and Alpha color transparency]** The **Transparency Color Low:** (Pixel Greater or Equal) (source background register) and the **Transparency Color High:** (Pixel Less or Equal) (source foreground register) are compared to the source pixels. The range comparisons are done on each component (A,R,G,B) and then logically ANDed. If the source pixel components are not within the range defined by the Transparency Color registers, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation. 101 = **[Destination and Alpha color transparency]** The **Transparency Color Low:** (Pixel Greater or Equal) (source background register) and the **Transparency Color High:** (Pixel Less or Equal) (source foreground register) are compared to the destination pixels. The range comparisons are done on each component (A,R,G,B) and then logically ANDed. If the destination pixels are within the range, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation. 111 = **[Destination color transparency]** The **Transparency Color Low:** (Pixel Greater or Equal) (source background register) and the **Transparency Color High:** (Pixel Less or Equal) (source foreground register) are compared to the destination pixels. The range comparisons are done on each component (R,G,B) and then logically ANDed. If the destination pixels are within the range, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
7:5	**Pattern Vertical Seed.** This field specifies the pattern scan line which corresponds to Y=0.
4	**Destination Read Modify Write.** This bit is decoded from the last instruction's opcode field and Destination Transparency Mode to identify whether a Destination read is needed.
3	**Color Source.** This bit is decoded from the last instructions opcode field to identify whether a color (1) source is used.
2	**Monochrome Source.** This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) source is used.
1	**Color Pattern.** This bit is decoded from the last instructions opcode field to identify whether a color (1) pattern is used.
0	**Monochrome Pattern.** This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) pattern is used.

1.10.2 BR01—Setup BLT Raster OP, Control, and Destination Offset

Memory Offset Address: none
Default: 0000 xxxx
Attributes: State accessible

BR01 contains the contents of the last Setup instruction DWord 1. It is identical to the BLT Raster OP, Control, and Destination Offset definition, but it is used with the following instructions: PIXEL_BLT, SCANLINE_BLT, and TEXT_BLT.



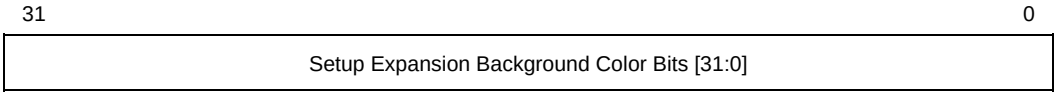
Bit De	scriptions
31	**Solid Pattern Select.** This bit applies only when the pattern data is monochrome. This bit determines whether or not the BLT Engine actually performs read operations from the frame buffer in order to load the pattern data. Use of this feature to prevent these read operations can increase BLT Engine performance, if use of the pattern data is indeed not necessary. The BLT Engine is configured to accept either monochrome or color pattern data via the opcode field. 0 = This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations. 1 = The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.
30	**Clipping Enabled**: 1 = Enabled; 0 = Disabled

Bit De	scriptions
29	**Monochrome Source Transparency Mode.** This bit applies only when the source data is in monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the source data also corresponds will actually be written if that source data bit has the value of 0. This feature can make it possible to use the source as a transparency mask. The BLT Engine is configured to accepted either monochrome or color source data via the opcode field. 0 = This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result. 1 = Wherever a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.
28	**Monochrome Pattern Transparency Mode.** This bit applies only when the pattern data is monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the pattern data also corresponds will actually be written if that pattern data bit has the value of 1. This feature can make it possible to use the pattern as a transparency mask. The BLT Engine is configured to accepted either monochrome or color pattern data via the opcode field. 0 = This causes normal operation with regard to the use of the pattern data. Wherever a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result. 1 = Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.
27:26	**32 bpp byte mask.** 21 = 1 = write alpha channel [31:24]; 20 = 1 = write RGB channels [23:00]. This field is only used for 32bpp.
25:24	**Color Depth.** 00 = 8 Bit Color Depth 01 = 16 Bit Color Depth 10 = 16 Bit Color Depth 11 = 32 Bit Color Depth
23:16	**Raster Operation Select.** These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine. The 8-bit values, and their corresponding raster operations, are intended to correspond to the 256 possible raster operations specified for graphics device drivers. The opcode field must indicate a monochrome source if ROP = F0.

Bit De	scriptions
15:0	**Destination Pitch (Offset).** For non-XY Blits, the signed 16bit field allows for specifying upto ± 32Kbytes signed pitches in bytes (same as before). For X, Y Blits with tiled-X surfaces, the pitch for Destination will be 512Byte aligned and should be programmable upto ± 128Kbytes. For X, Y Blits with Tiled-Y surfaces, the pitch for Destination will be 128Byte aligned and should be programmable upto + 128Kbytes. In this case, this 16bit signed pitch field is used to specify upto ± 32K**DWords**. For X, Y blits with nontiled surfaces (linear surfaces), this 16bit field can be programmed to byte specification of upto ± 32Kbytes (same as before). These 16 bits store the signed memory address offset value by which the destination address originally specified in the Destination Address Register is incremented or decremented as each scan line's worth of destination data is written into the frame buffer by the BLT Engine, so that the destination address will point to the next memory address to which the next scan line's worth of destination data is to be written. If the intended destination of a BLT operation is within on-screen frame buffer memory, this offset is normally set so that each subsequent scan line's worth of destination data lines up vertically with the destination data in the scan line, above. However, if the intended destination of a BLT operation is within off-screen memory, this offset can be set so that each subsequent scan line's worth of destination data is stored at a location immediately after the location where the destination data for the last scan line ended, in order to create a single contiguous block of bytes of destination data at the destination.

1.10.3 BR05—Setup Expansion Background Color

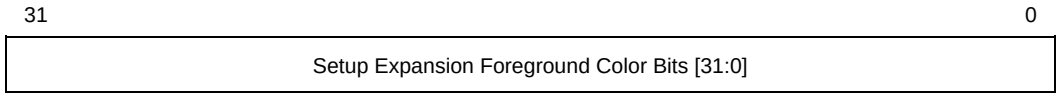
Memory Offset Address: none
Default: None
Attributes: State accessible



Bit De	scriptions
31:0	**Setup Expansion Background Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome pattern or source data for either the SCANLINE_BLT or TEXT_BLT instructions. BR05 is also used as the solid pattern for the PIXEL_BLT instruction. Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

1.10.4 BR06—Setup Expansion Foreground Color

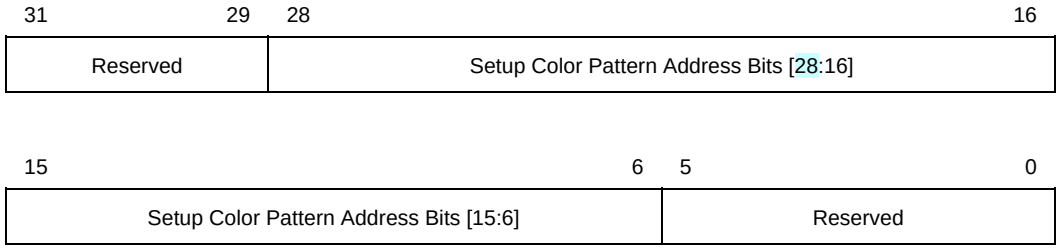
Memory Offset Address: none
Default: None
Attributes: State accessible



Bit De	scriptions
31:24	**Reserved.**
31:0	**Setup Expansion Foreground Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome pattern or source data for either the SCANLINE_BLT or TEXT_BLT instructions. Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

1.10.5 BR07—Setup Color Pattern Address

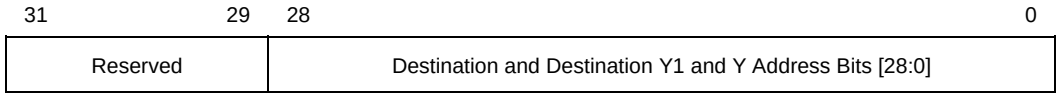
Memory Offset Address: none
Default: None
Attributes: State accessible



Bit De	scriptions
31:29	**Reserved.** The maximum GC graphics address is 512 MBs.
28:6	**Pattern Address.** These 23 bits specify the starting address of the color pattern from the SETUP_BLT instruction. This register works identically to the Pattern Address register, but this version is only used with the SCANLINE_BLT instruction execution. The pattern data must be located in linear memory. The pattern data must be located on a pattern-size boundary. The pattern is always of 8x8 pixels, and therefore, its size is dependent upon its pixel depth. The pixel depth may be 8, 16, or 32 bits per pixel if the pattern is in color (the pixel depth of a color pattern must match the pixel depth to which the graphics system has been set). Monochrome patterns require 8 bytes and is supplied through the instruction. Color patterns of 8, 16, and 32 bits per pixel color depth must start on 64-byte, 128-byte and 256-byte boundaries, respectively.
5:0	**Reserved.** These bits always return 0 when read.

1.10.6 BR09—Destination Address

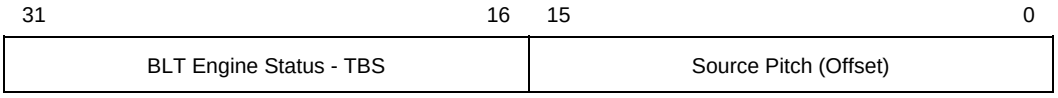
Memory Offset Address: None
Default: None
Attributes: State accessible



Bit De	scriptions
31:29	**Reserved.**
28:0	**Destination Address Bits.** When tiling is enabled for XY-blits, this base address should be limited to 4KB. Otherwise for XY blits, there is no restriction and it is same as before. These 29 bits specify the starting pixel address of the destination data. This register is also the working destination address register and changes as the BLT Engine performs the accesses. Used as the scan line address (Destination Y Address & Destination Y1 Address) for BLT instructions: PIXEL_BLT, SCANLINE_BLT, and TEXT_BLT. In this case the address points to the first pixel in a scan line and is compared with the ClipRect Y1 & Y2 address registers to determine whether the scan line should be written or not. The Destination Y1 address is the top scan line to be written for text. Note that for non-XY blits (COLOR_BLT, SRC_COPY_BLT), this address points to the first byte to be written. This register is always the last register written for a BLT drawing instruction. Writing BR09 starts the BLT engine execution. ***Note:*** Some instructions affect only one scan line (requiring only one coordinate); other instructions affect multiple scan lines and need both coordinates.

1.10.7 BR11—BLT Source Pitch (Offset)

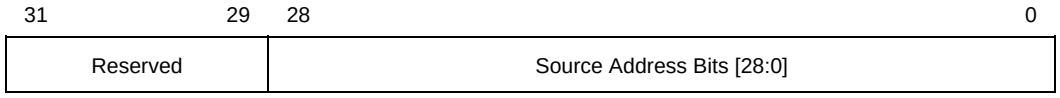
Memory Offset Address: None
Default: None
Attributes: Not accessible



Bit De	scriptions
15:0	**Source Pitch (Offset)** For non-XY Blits with color source operand (SRC_COPY_BLT), the signed 16bit field allows for specifying upto ± 32Kbytes signed pitch in bytes (same as before). For X, Y Blits with tiled-X surfaces, the pitch for Color Source will be 512Byte aligned and should be programmable upto ± 128Kbytes. For X, Y Blits with tiled-Y surfaces, the pitch for Color Source will be 128Byte aligned and should be programmable upto ± 128Kbytes. In this case, this 16bit signed pitch field is used to specify up to upto ± 32K**DWords**. For X, Y blits with nontiled color source surfaces (linear surfaces), this 16bit field can be programmed to byte specification of upto ± 32Kbytes (same as before). When the color source data is located within the frame buffer or AGP aperture, these signed 16 bits store the memory address offset (pitch) value by which the source address originally specified in the Source Address Register is incremented or decremented as each scan line's worth of source data is read from the frame buffer by the BLT Engine, so that the source address will point to the next memory address from which the next scan line's worth of source data is to be read. Note that if the intended source of a BLT operation is within on-screen frame buffer memory, this offset is normally set to accommodate the fact that each subsequent scan line's worth of source data lines up vertically with the source data in the scan line, above. However, if the intended source of a BLT operation is within off-screen memory, this offset can be set to accommodate a situation in which the source data exists as a single contiguous block of bytes where in each subsequent scan line's worth of source data is stored at a location immediately after the location where the source data for the last scan line ended.

1.10.8 BR12—Source Address

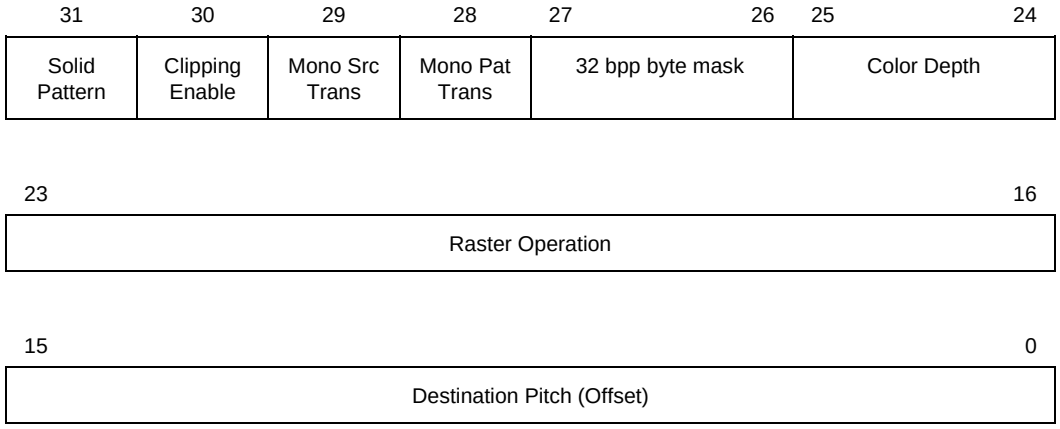
Memory Offset Address: None
Default: None
Attributes: Not accessible



Bit De	scriptions
31:29	**Reserved.** The maximum GC Graphics address is 512 MBs.
28:0	**Source Address Bits [28:0].** When tiling is enabled for XY-blits with Color source surfaces, this base address should be limited to 4KB. Otherwise for XY blits, there is no restriction and it is same as before, including for monosource and text blits. Note that for non-XY blit with Color Source (SRC_COPY_BLT), this address points to the first byte to be read. These 29 bits are used to specify the starting pixel address of the color source data. The lower 3 bits are used to indicate the position of the first valid byte within the first Quadword of the source data.

1.10.9 BR13—BLT Raster OP, Control, and Destination Pitch

Memory Offset Address: None
Default: 0000 xxxx
Attributes: Not accessible



Bit De	scriptions
31	**Solid Pattern Select.** This bit applies only when the pattern data is monochrome. This bit determines whether or not the BLT Engine actually performs read operations from the frame buffer in order to load the pattern data. Use of this feature to prevent these read operations can increase BLT Engine performance, if use of the pattern data is indeed not necessary. The BLT Engine is configured to accept either monochrome or color pattern data via the opcode field. 0 = This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations. 1 = The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.
30	**Clipping Enabled**: 1 = Enabled; 0 = Disabled
29	**Monochrome Source Transparency Mode.** This bit applies only when the source data is in monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the source data also corresponds will actually be written if that source data bit has the value of 0. This feature can make it possible to use the source as a transparency mask. The BLT Engine is configured to accepted either monochrome or color source data via the opcode field. 0 = This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result. 1 = Where a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.

Bit De	scriptions
28	**Monochrome Pattern Transparency Mode.** This bit applies only when the pattern data is monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the pattern data also corresponds will actually be written if that pattern data bit has the value of 1. This feature can make it possible to use the pattern as a transparency mask. The BLT Engine is configured to accepted either monochrome or color pattern data via the opcode in the Opcode and Control register. 0 = This causes normal operation with regard to the use of the pattern data. Where a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result. 1= Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.
25:24	**Color Depth.** 00 = 8 Bit Color Depth 01 = 16 Bit Color Depth 10 = 24 Bit Color Depth 11 = Reserved
23:16	**Raster Operation Select.** These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine. The 8-bit values, and their corresponding raster operations, are intended to correspond to the 256 possible raster operations specified for graphics device drivers. The opcode must indicate a monochrome source operand if ROP = F0.
15:0	**Destination Pitch (Offset).** These 16 bits store the signed memory address offset value by which the destination address originally specified in the Destination Address Register is incremented or decremented as each scan line's worth of destination data is written into the frame buffer by the BLT Engine, so that the destination address will point to the next memory address to which the next scan line's worth of destination data is to be written. If the intended destination of a BLT operation is within on-screen frame buffer memory, this offset is normally set so that each subsequent scan line's worth of destination data lines up vertically with the destination data in the scan line, above. However, if the intended destination of a BLT operation is within off-screen memory, this offset can be set so that each subsequent scan line's worth of destination data is stored at a location immediately after the location where the destination data for the last scan line ended, in order to create a single contiguous block of bytes of destination data at the destination.

1.10.10 BR14—Destination Width & Height

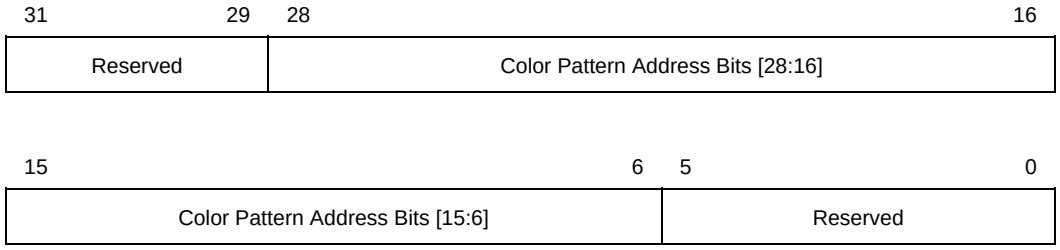
Memory Offset Address: None
Default: None
Attributes: Not accessible

BR14 contains the values for the height and width of the data to be BLT. If these values are not correct, such that the BLT Engine is either expecting data it does not receive or receives data it did not expect, the system can hang.

Bit De	scriptions
31:29	**Reserved.**
28:16	**Destination Height.** These 13 bits specify the height of the destination data in terms of the number of scan lines. This is a working register.
15:13	**Reserved.**
12:0	**Destination Byte Width.** These 13 bits specify the width of the destination data in terms of the number of bytes per scan line. The number of pixels per scan line into which this value translates depends upon the color depth to which the graphics system has been set.

1.10.11 BR15—Color Pattern Address

Memory Offset Address: None
Default: None
Attributes: Not accessible

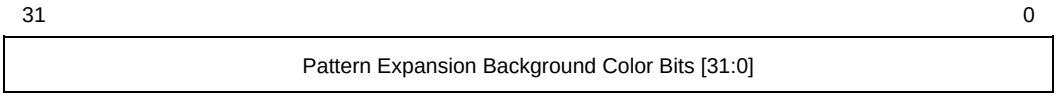


.

Bit De	scriptions
31:29	**Reserved.** The maximum GC graphics address is 512 MBs.
28:6	**Color Pattern Address.** There is no change to the Color Pattern address specification due to Non-Power-of-2 change. It remains the same as before. The pattern data must be located in linear memory. These 23 bits specify the starting address of the pattern. The pattern data must be located on a pattern-size boundary. The pattern is always of 8x8 pixels, and therefore, its size is dependent upon its pixel depth. The pixel depth may be 8, 16, or 32 bits per pixel if the pattern is in color (the pixel depth of a color pattern must match the pixel depth to which the graphics system has been set). Monochrome patterns require 8 bytes and are applied through the instruction. Color patterns of 8, 16, and 32 bits per pixel color depth must start on 64-byte, 128-byte and 256-byte boundaries, respectively.
5:0	**Reserved.** These bits always return 0 when read.

1.10.12 BR16—Pattern Expansion Background & Solid Pattern Color

Memory Offset Address: 40040h
Default: None
Attributes: RO; DWord accessible

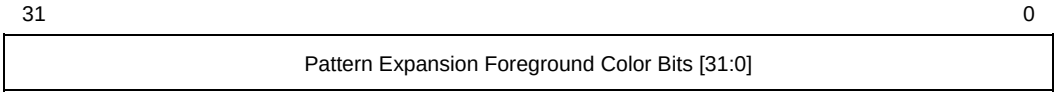


.

Bit De	scriptions
31:0	**Pattern Expansion Background Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome pattern data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

1.10.12.1 BR17—Pattern Expansion Foreground Color

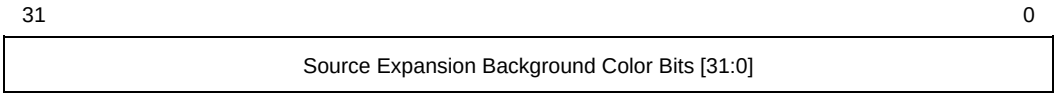
Memory Offset Address: None
Default: None
Attributes: Not accessible



Bit De	scriptions
31:0	**Pattern Expansion Foreground Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome pattern data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

1.10.13BR18—Source Expansion Background, and Destination Color

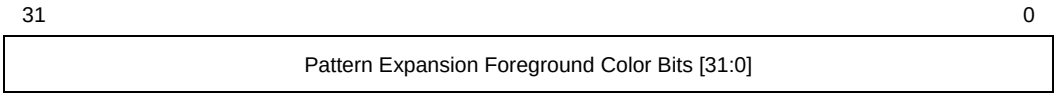
Memory Offset Address: None
Default: None
Attributes: Not accessible



Bit De	scriptions
31:0	**Source Expansion Background Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome source data during BLT operations. This register is also used to support destination transparency mode and Solid color fill. Whether one, two, three, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

1.10.14BR19—Source Expansion Foreground Color

Memory Offset Address: None
Default: None
Attributes: Not accessible



Bit De	scriptions
31:0	**Pattern/Source Expansion Foreground Color Bits [31:0].** These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome source data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

2. Blitter (Blt) Engine Command Streamer

2.1 Registers for Blitter Engine

2.1.1 Introduct ion

Each register is at the same offset from 020000h as its primary counterpart is offset from 02000h.

2.1.2 Virtual Memory Control

2.1.2.1 BLT_PP_DIR_BASE – Page Directory Base Register

BLT_PP_DIR_BASE – Page Directory Base Register	
Register Type:	MMIO_BCS
Address Offset:	22390h
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Trusted Type:	1

This register contains the offset into the GGTT where the (current context's) PPGTT page directory begins. This register is restored with context

Bit De	scription
31:16	**Page Directory Base Offset** Project: All Security: None Default Value: 0h DefaultVaueDesc Format: U15 Address: GraphicsAddress[31:16] Range [0,PPGTT Size - 1 in cachelines] Contains the cacheline (64-byte) address into the GGTT where the page directory begins.
15:0	**Reserved** Project: All Format: MBZ

2.1.2.2 BLT_PP_DCLV – PPGTT Directory Cacheline Valid Register

BLT_PP_DCLV – PPGTT Directory Cacheline Valid Register	
Register Type:	MMIO_BCS
Address Offset:	22398h
Project:	All
Default Value:	00000000h; 00000000h; 00000000h; 00000000h
Access:	RW
Size (in bits):	2x32
Trusted Type:	1

This register can be used to designate entire cachelines of the PPGTT Directory as invalid. Bits that are set indicate the corresponding 16 directory entry group is valid. Note that some or all of the entries could have their valid bits clear, indicating they are invalid

This register can effectively be used to limit the size of a processes' virtual address space. Any access by a process that requires a PD entry in a set that is not enabled in this register will cause a fatal error, and no attempt to fetch the PD entry will be made.

DWord	Bit	Description
0	31:0	PPGTT Directory Cacheline Valid. Project: All Security: None Default Value: 0h DefaultVaueDesc Format: U32 FormatDesc If set, each bit in this register corresponds to 16 valid entries of the page directory. If clear, these entries are considered invalid and fetch of these entries will not be attempted.
1	31:0	**Reserved** Project: All Format: MBZ

2.1.3 Mode and Misc Ctrl Registers

2.1.3.1 BCS_MI_MODE — Mode Register for Software Interface

Address Offset: 2209Ch–2209Fh
Default Value: 0000 0000h
Access: Read/Write
Size: 32 bits

The MI_MODE register contains information that controls software interface aspects of the command parser.

Bit De	scription
31:16	**Masks:** A “1” in a bit in this field allows the modification of the corresponding bit in Bits 15:0
15:12	**Reserved** Read/Write
11	**Invalidate UHPTR enable:** If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.
10	**Reserved** Read/Write
9	**Ring Idle (Read Only Status bit)** 0 = Parser not Idle 1 = Parser Idle ***Writes to this bit are not allowed.***
8	**Stop Ring** 0 = Normal Operation. 1 = Parser is turned off. Software must set this bit to force the Ring and Command Parser to Idle. Software must read a “1” in Ring Idle bit after setting this bit to ensure that the hardware is idle. ***Software must clear this bit for Ring to resume normal operation.***
7:0	**Reserved** Read/Write

2.1.3.2 BCS_NOPID — NOP Identification Register

BCS_NOPID — NOP Identification Register					
Register Type:	MMIO_BCS				
Address Offset:	22094h				
Project:	All				
Default Value:	00000000h				
Access:	RO				
Size (in bits):	32				
Trusted Type:	1				
The NOPID register contains the Noop Identification value specified by the last MI_NOOP instruction that enabled this register to be updated.					
Bit De	**scription**				
31:22	**Reserved**	Project:	All	Format:	MBZ
21:0	**Identification Number** Project: All Security: None Default Value: 0h DefaultVaueDesc This field contains the 22-bit Noop Identification value specified by the last MI_NOOP instruction that enabled this field to be updated				

2.1.3.3 BCS_INST PM—Instruction Parser Mode Register

Address Offset: 220C0h–220C3h
Default Value: 0000 0000h
Access: Read/Write
Size: 32 bits

The BCS_INSTPM register is used to control the operation of the BCS Instruction Parser. Certain classes of instructions can be disabled (ignored) – often useful for detecting performance bottlenecks. Also, "Synchronizing Flush" operations can be initiated – useful for ensuring the completion (vs. only parsing) of rendering instructions.

Programming Notes:

- All Reserved bits are implemented

2.1.3.4 BCS_EXCC—Execute Condition Code Register

BCS_EXCC—Execute Condition Code Register

Register Type:	MMIO_BCS
Address Offset:	22028h
Project:	All
Default Value:	00000000h
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a "1", while instruction is discarded if the condition evaluates to a "0". Once excluded a ring is enabled into arbitration when the selected condition evaluates to a "0".

Bit De	scription
31:18	**Reserved** Project: All Format: MBZ
17	**Mask Bits** Format: Mask[1] This bit serves as a write enable for bit 1. If this register is written with this bit clear the corresponding bit in the field 1 will not be modified. Reading these bits always returns 0s.
16	**Mask Bits** Format: Mask[0] These bits serves as a write enable for bit 0. If this register is written with any of these bits clear the corresponding bit in the field 0 will not be modified. Reading these bits always returns 0s.
15:2	**Reserved** Project: All Format: MBZ
1	**Video Command Streamer Condition Codes** The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).

BCS_EXCC—Execute Condition Code Register	
0	**Render Command Streamer Condition Codes** The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).

2.1.3.5 BRSYNC – Blitter/Render Semaphore Sync Register

BRSYNC – Blitter/Render Semaphore Sync Register	
Register Type:	MMIO_BCS
Address Offset:	22040h
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Trusted Type:	1
This register is written by CS, read by BCS.	
Bit De	**scription**
31:0	**Semaphore Data** Semaphore data for synchronization between blitter engine and render engine..

2.1.3.6 VSYNC – Blitter/Video Semaphore Sync Register

BVSYNC – Blitter/Video Semaphore Sync Register	
Register Type:	MMIO_BCS
Address Offset:	22044h
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Trusted Type:	1
This register is written by VCS, read by BCS.	
Bit De	**scription**
31:0	**Semaphore Data** Semaphore data for synchronization between blitter engine and video codec engine.

2.1.4 Context Submission

Before submitting a context for the first time, the context image must be properly initialized. Proper initialization includes the ring context registers (ring location, head/tail pointers, etc.) and the page directory.

2.1.4.1 BCS_RCCID—Ring Buffer Current Context ID Register

Address Offset: 227C0h–227C4h
Default Value: 00 00 00 00h
Access: Read/Write
Size: 32 bits

This register contains the current "ring context ID" associated with the ring buffer.

Programming Notes:

- The current context registers must not be written directly (via MMIO). The RCCID register should only be updated indirectly from RNCID.

Bit Description	
63:0	See Context Descriptor for BCS

2.1.4.2 VCS_RNCID—Ring Buffer Next Context ID Register

Address Offset: 22700h–22708h
Default Value: 00 00 00 00h
Access: Read/Write
Size: 64 bits

This register contains the *next* "ring context ID" associated with the ring buffer.

Programming Notes:

- The current context (RCCID) register can be updated indirectly from this register on a context switch event. Note that the only time a context switch can occur is when MI_ARB_CHECK enables preemption or the current context runs dry (head pointer becomes equal to tail pointer).

Bit Description	
63:0	See Context Descriptor for BCS

2.1.4.3 Context Status

A context switch interrupt will be sent anytime a context switch change occurs. This is documented in the “GPU Overview” volume, “Memory Data Formats” chapter. A status DW for the context that was just switched away from will be written to the Context Status Buffer in the Global Hardware Status Page. The status contains the context ID and the reason for the context switch. Note that since there will have been no running contexts when the very first (after reset) context is submitted, the Context ID in the first Context Status DWord will be UNDEFINED.

Bit De	scription
31:12	**Context ID.** Contains the context ID copied from the submitted context.
11:8	Reserved: MBZ
7	Reserved: MBZ
6	Reserved: MBZ
5	Reserved: MBZ
4	**Ring Buffer Becoming Empty** Caused context to Switch.
3	Reserved: MBZ
2	Reserved: MBZ
1	**Waiting on a Semaphore** Caused Context to Switch.
0	Reserved: MBZ

2.1.5 BCS_RINGBUF—Ring Buffer Registers

RINGBUF—Ring Buffer Registers	
Register Type:	MMIO_BCS
Address Offset:	22030h
Project:	All
Default Value:	00000000h; 00000000h; 00000000h; 00000000h
Access:	R/W
Size (in bits):	4x32
Trusted Type:	1

These registers are used to define and operate the "ring buffer" mechanism which can be used to pass instructions to the command interface. The buffer itself is located in a physical memory region. The ring buffer is defined by a 4 Dword register set that includes starting address, length, head offset, tail offset, and control information. Refer to the *Programming Interface* chapter for a detailed description of the parameters specified in this ring buffer register set, restrictions on the placement of ring buffer memory, arbitration rules, and in how the ring buffer can be used to pass instructions.

Ring Buffer Head and Tail Offsets must be properly programmed before it is enabled. A Ring Buffer can be enabled when empty.

DWord	Bit	Description
0	31:21	**Reserved** Project: All Format: MBZ
	20:3	**Tail Offset** Project: All Format: U18 This field is written by software to specify where the valid instructions placed in the ring buffer end. The value written points to the QWord *past* the last valid QWord of instructions. In other words, it can be defined as the *next* QWord that software will write instructions into. Software must write subsequent instructions to QWords following the Tail Offset, possibly wrapping around to the top of the buffer (i.e., software can't skip around within the buffer). Note that all DWords prior to the location indicated by the **Tail Offset** must contain valid instruction data – which may require instruction padding by software. See **Head Offset** for more information. QWord Offset
	2:0	**Reserved** Project: All Format: MBZ
1	31:21	**Wrap Count** Project: All Format: U11 This field is incremented by 1 whenever the **Head Offset** wraps from the end of the buffer back to the start (i.e., whenever it wraps back to 0). Appending this field to the **Head Offset** field effectively creates a virtual 4GB Head "Pointer" which can be used as a tag associated with instructions placed in a ring buffer. The Wrap Count itself will wrap to 0 upon overflow. The Wrap Count will get cleared as a result of writes of the Starting Address field.

RINGBUF—Ring Buffer Registers		
	20:2	**Head Offset** Project: All Format: U19 This field indicates the offset of the *next* instruction DWord to be parsed. Software will initialize this field to select the first DWord to be parsed once the RB is enabled. (Writing the Head Offset while the RB is enabled is UNDEFINED). Subsequently, the device will increment this offset as it executes instructions – until it reaches the QWord specified by the **Tail Offset**. At this point the ring buffer is considered "empty". **Programming Notes:** • A RB can be enabled empty or containing some number of valid instructions. • Head Offset is cleared as a result of writes of the Starting Address field.
	1:0	**Reserved** Project: All Format: MBZ
2	31:12	**Starting Address** Project: All Format: Graphics Address[31:12] This field specifies Bits 31:12 of the 4KB-aligned starting Graphics Address of the ring buffer. Address bits 31 down to 29 must be zero. Writing this register also causes the Head Offset to be reset to zero, and the Wrap Count to be reset to zero. All ring buffer pages must map to Main Memory (uncached) pages. Ring Buffer addresses are always translated through the global GTT. Per-process address space can only be used via a batch buffer with the appropriate **Memory Space Select**.
	11:0	**Reserved** Project: All Format: MBZ
3	31:21	**Reserved** Project: All Format: MBZ
	20:12	**Buffer Length** Project: All Format: U9 This field is written by SW to specify the length of the ring buffer in 4 KB Pages. Range = [0 = 1 page = 4 KB, 1FFh = 512 pages = 2 MB]
	11	**RB Wait** Project: All Format: Boolean Indicates that this ring has executed a WAIT_FOR_EVENT instruction and is currently waiting. Software can write a "1" to clear this bit, write of "0" has no effect. When the RB is waiting for an event and this bit is cleared, the wait will be terminated and the RB will be returned to arbitration.
	9:3	**Reserved** Project: All Format: MBZ

RINGBUF—Ring Buffer Registers		
	2:1	**Automatic Report Head Pointer** Project: All Format: U2 This field is written by software to control the automatic "reporting" (write) of this ring buffer's "Head Pointer" register (register DWord 1) to the corresponding location within the Hardware Status Page. Automatic reporting can either be disabled or enabled at 4KB, 64KB or 128KB boundaries within the ring buffer. Format = 0: MI_AUTOREPORT_OFF – Automatic reporting disabled 1: MI_AUTOREPORT_64KB – Report every 16 pages (64KB) 2: MI_AUTOREPORT_4KB – Report every page (4KB) 3: MI_AUTOREPORT_128KB – Report every 32 pages (128KB) When the **Per-Process Virtual Address Space Enable** bit is set and automatic head reporting is desired, this field must be set to option 2 since the ring buffer will be only 16KB in size. The head pointer will be reported to the head pointer location in the PP HW Status Page when it passes each 4KB page boundary.
	0	**Ring Buffer Enable** Project: All Format: Enable This field is used to enable or disable this ring buffer. It can be enabled or disabled regardless of whether there are valid instructions pending.

2.1.5.1 BCS_UHPTR — Pending Head Pointer Register

Address Offset: 22134h–22137h
Default Value: 0000 0000h
Access: Read/Write
Size: 32 bits

Bit De	scription
31:3	**Head Pointer Address**: This register represents the GFX address offset where execution should continue in the ring buffer following execution of an MI_ARB_CHECK command. Format = MI_Graphics_Offset
2:1	Reserved: MBZ
0	**Head Pointer Valid:** 1 = Indicates that there is an updated head pointer programmed in this register 0 = No valid updated head pointer register, resume execution at the current location in the ring buffer This bit is set by the software to request a pre-emption. It is reset by hardware after the head pointer in this register is read. The hardware uses the head pointer programmed in this register at the time the reset is generated.

2.1.6 Interrupt Control Registers

The Interrupt Control Registers described below all share the same bit definition. The bit definition is as follows:

Table 2-1. Bit Definition for Interrupt Control Registers

Bit De	scription
31:4	**Reserved. MBZ:** These bits may be assigned to interrupts on future products/steppings.
7	**Page Fault:** This bit is set whenever there is a pending PPGTT (page or directory) fault.
6	**Reserved. MBZ**
5	**Reserved. MBZ**
4	**MI_FLUSH_DW Notify Interrupt:** The Pipe Control packet (Fences) specified in *3D pipeline* document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.
3	**Blitter Command Parser Master Error:** When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the “Error Status Register” which along with the “Error Mask Register” determine which error conditions will cause the error status bit to be set and the interrupt to occur. **Page Table Error:** Indicates a page table error. **Instruction Parser Error**: The Renderer Instruction Parser encounters an error while parsing an instruction.
2	**Sync Status:** This bit is toggled when the Instruction Parser completes a flush with the sync enable bit active in the INSTPM register. The toggle event will happen after all the graphics engines are flushed. The HW Status DWord write resulting from this toggle will cause the CPU’s view of graphics memory to be coherent as well (flush and invalidate the render cache).
1	**Reserved. MBZ**
0	**Blitter Command Parser User Interrupt:** This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Render Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.

2.1.6.1 HWSTAM — Hardware Status Mask Register

Hardware Status Mask Register	
Register Type:	MMIO_BCS
Address Offset:	22098h
Project:	All
Default Value:	**FFFF FFFFh**
Access:	R/W
Size (in bits):	32
Trusted Type:	1

The HWSTAM register has the same format as the Interrupt Control Registers. The bits in this register are “mask” bits that prevent the corresponding bits in the Interrupt Status Register from generating a “Hardware Status Write” (PCI write cycle). Any unmasked interrupt bit (HWSTAM bit set to 0) will allow the Interrupt Status Register to be written to the ISR location (within the memory page specified by the Hardware Status Page Address Register) when that Interrupt Status Register bit changes state.

Bit De	scription
31:0	**Hardware Status Mask Register** Project: All Default Value: FFFFFFFFh DefaultVaueDesc Format: Array of Masks See the Interrupt Control Register section for bit definitions

4

2.1.6.2 IMR—Interrupt Mask Register

IMR—Interrupt Mask Register	
Register Type:	MMIO_BCS
Address Offset:	220A8h
Project:	All
Default Value:	FFFF FFFFh
Access:	R/W
Size (in bits):	32

The IMR register is used by software to control which Interrupt Status Register bits are “masked” or “unmasked”. “Unmasked” bits will be reported in the IIR, possibly triggering a CPU interrupt, and will persist in the IIR until cleared by software. “Masked” bits will not be reported in the IIR and therefore cannot generate CPU interrupts.

Bit De	scription
31:0	**Interrupt Mask Bits** Project: All Default Value: FFFF FFFFh Format: Array of interrupt mask bits — Refer to Table 1 4 in Interrupt Control Register section for bit definitions This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.

Value Na	me	Description	Project
0h	Not Masked	Will be reported in the IIR	All
1h	Masked	Will not be reported in the IIR	All

2.1.6.3 Hardware-Detected Error Bit Definitions (for EIR, EMR, ESR)

This section defines the Hardware-Detected Error bit definitions and ordering that is common to the EIR, EMR and ESR registers. The EMR selects which error conditions (bits) in the ESR are reported in the EIR. Any bit set in the EIR will cause the Master Error bit in the ISR to be set. EIR bits will remain set until the appropriate bit(s) in the EIR is cleared by writing the appropriate EIR bits with '1'.

The following table describes the Hardware-Detected Error bits:

Table 2-2. Hardware-Detected Error Bits

Bit De	scription
15:5	Reserved: MBZ
4	**Page Table Error**: This bit is set when a Graphics Memory Mapping Error is detected. The cause of the error is indicated (to some extent) in the PGTBL_ER register. Note: This error indications can not be cleared except by reset (i.e., it is a fatal error). 1 = Page table error
1	Reserved.
0	**Instruction Error:** This bit is set when the Renderer Instruction Parser detects an error while parsing an instruction. Instruction errors include: 1) Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported). 2) Defeatured MI Instruction Opcodes: 1: Instruction Error detected Programming Note: [DevBW][DevCL]: The bit for the error mask of this register is reserved. The mask should be set to a value of 1.

2.1.6.3.1 EIR — Error Identity Register

EIR — Error Identity Register			
Register Type:	MMIO_BCS		
Address Offset:	220B0h		
Project:	All		
Default Value:	0000 0000h		
Access:	R/WC		
Size (in bits):	32		
The EIR register contains the persistent values of Hardware-Detected Error Condition bits. Any bit set in this register will cause the Master Error bit in the ISR to be set. The EIR register is also used by software to clear detected errors (by writing a '1' to the appropriate bit(s)).			
Bit De	**scription**		
31:16	**Reserved** Project: All Format: MBZ		
15:0	**Error Identity Bits** Project: All Default Value: 0h Format: Array of Error condition bits See Table 1 5. Hardware-Detected Error Bits This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a '1' to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.		

Value Na	me	Description	Project
1h	Error occurred	Error occurred	All

Programming Notes	Project
Writing a '1' to a set bit will cause that error condition to be cleared. However, the Page Table Error bit (Bit 4) can not be cleared except by reset (i.e., it is a fatal error).	All

2.1.6.3.2 EMR—Error Mask Register

EMR—Error Mask Register	
Register Type:	MMIO_BCS
Address Offset:	220B4h
Project:	All
Default Value:	FFFF FFFFh
Access:	R/W
Size (in bits):	32

The EMR register is used by software to control which Error Status Register bits are “masked” or “unmasked”. “Unmasked” bits will be reported in the EIR, thus setting the Master Error ISR bit and possibly triggering a CPU interrupt, and will persist in the EIR until cleared by software. “Masked” bits will not be reported in the EIR and therefore cannot generate Master Error conditions or CPU interrupts.

Bit De	scription
31:16	**Reserved** Project: All Format: MBZ
15:0	**Error Mask Bits** Project: All Default Value: FFFF FFFFh Format: Array of error condition mask bits See Table 1 5. Hardware-Detected Error Bits This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

Value Na	me	Description	Project
0h	Not Masked	Will be reported in the EIR	All
1h	Masked	Will not be reported in the EIR	All

2.1.6.3.3 ESR—Error Status Register

ESR—Error Status Register	
Register Type:	MMIO_BCS
Address Offset:	220B8h
Project:	All
Default Value:	0000 0000h
Access:	RO
Size (in bits):	32

The ESR register contains the current values of all Hardware-Detected Error condition bits (these are all by definition "persistent"). The EMR register selects which of these error conditions are reported in the persistent EIR (i.e., set bits must be cleared by software) and thereby causing a Master Error interrupt condition to be reported in the ISR.

Bit De	scription
31:16	**Reserved** Project: All Format: MBZ
15:0	**Error Status Bits** Project: All Default Value: 0h Format: Array of error condition bits See Table 1 5. Hardware-Detected Error Bits This register contains the non-persistent values of all hardware-detected error condition bits.

Value Na	me	Description	Project
1h	Error Condition Detected	Error Condition detected	All

2.1.7 Logical Context Support

2.1.7.1 BCS_BB_ADDR—Batch Buffer Head Pointer Register

Address Offset: 022140h–022147h
Default Value: 0000 0000 0000 0000h
Access: Read-Only
Size: 64 bits

This register contains the current QWord Graphics Memory Address of the last-initiated batch buffer.

Bit De	scription
63:32	Reserved: MBZ
31:3	**Batch Buffer Head Pointer:** This field specifies the QWord-aligned Graphics Memory Address where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit will be 0 and this field will be meaningless. .
2:1	Reserved: MBZ
0	**Valid:** 1 = Batch buffer Valid 0 = Batch buffer Invalid

2.1.8 Software Control Bit Definitions

Registers in the range 22XX are not protected from the load register immediate instruction if the command is executed in the non-secure batch buffer.

2.2 Memory Interface Commands for Blitter Engine

2.2.1 Introduct ion

This chapter describes the formats of the "Memory Interface" commands, including brief descriptions of their use. The functions performed by these commands are discussed fully in the *Memory Interface Functions* Device Programming Environment chapter.

This chapter describes MI Commands for the blitter graphics processing engine. The term "for Blitter Engine" in the title has been added to differentiate this chapter from a similar one describing the MI commands for the Media Decode Engine and the Rendering Engine.

The commands detailed in this chapter are used across products within the Gen4 family. However, slight changes may be present in some commands (i.e., for features added or removed), or some commands may be removed entirely. Refer to the *Preface* chapter for product specific summary.

2.2.2 MI_ARB_CHECK

The MI_ARB_CHECK instruction is used to check the ring buffer next context ID register (RNCID) or the UHPTR register. This instruction can be used to pre-empt the current execution of the ring buffer. Note that the valid bit in the RNCID register or the UHPTR register needs to be set for the command streamer to be pre-empted.

Programming Note:

- This instruction can be placed only in a ring buffer, never in a batch buffer.

The instruction format is:

DWord	Bits	Description
0	31:29	**Instruction Type =** MI_INSTRUCTION = 0h
	28:23	**MI Instruction Opcode =** MI_ARB_CHECK = 05h
	22:0	Reserved: MBZ

2.2.3 MI_SUSPEND_FLUSH

MI_SUSPEND_FLUSH			
Project:	All	**Length Bias:**	1
Blocks MMIO sync flush or any flushes related to VT-d while enabled.			

DWord Bit		Description
0	31:29	**Command Type** Default Value: 0h MI_COMMAND Format: OpCode
	28:23	**MI Command Opcode** Default Value: 0Bh MI_SUSPEND_FLUSH Format: OpCode
	22:1	**Reserved** Project: All Format: MBZ
	0	**Suspend Flush** Project: All Default Value: 0h DefaultVaueDesc Format: Enable FormatDesc This field suspends flush due to sync flush or implicit flush generated during VTD enable, disable and IOTLB invalidation.

Value Name	Description	Project
0h	Disable	All
1h	Enable	All