Version 2.0

Nintendo Ultra64 RDP Command Summary

Silicon Graphics Computer Systems, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94043-1389

Last Modified 4/11/96

Set Color Image

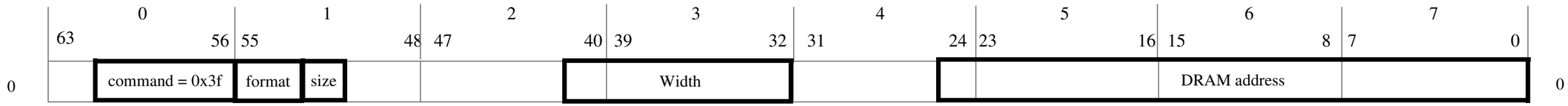


Table 1: Set Color Image Command Format

Field	Word	Bits	Description
command	0	61-56	Command identifier
format	0	55-53	Image data format, 0 = rgba, 1 = yuv, 2 = Color Indx, 3 = IA, 4 = I
size	0	52-51	Size of pixel/texel color element, 0 = 4b, 1 = 8b, 2 = 16b, 3 = 32b
Width	0	41-32	Width of image in pixels, image width = width + 1
DRAM adrs	0	25-0	Base address (top left corner) of image in DRAM, in bytes

Table 2: Legal Color Image Types/Sizes

Type	8b	16b	32b
RGBA		4	4
YUV			
Clr Indx	4		

Set Color Image Usage Notes:

Read/Modify/Write of 32b color image with depth buffer must be done in two cycle mode.

Set Texture Image

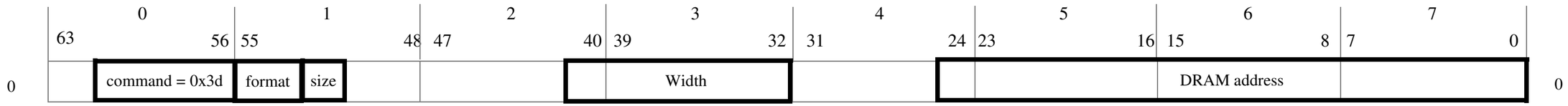


Table 3: Set Texture Image Command Format

Field	Word	Bits	Description
command	0	61-56	Command identifier
format	0	55-53	Image data format, 0 = rgba, 1 = yuv, 2 = Color Indx, 3 = IA, 4 = I
size	0	52-51	Size of pixel/texel color element, 0 = 4b, 1 = 8b, 2 = 16b, 3 = 32b
Width	0	41-32	Width of image in pixels, image width = width + 1
DRAM adrs	0	25-0	Base address (top left corner) of image in DRAM, in bytes

Table 4: Legal Texture Image Types/Sizes

Type	4b	8b	16b	32b
RGBA			4	4
YUV			4	
Clr Indx	4	4		
IA	4	4	4	
I	4	4		

Set Z Image

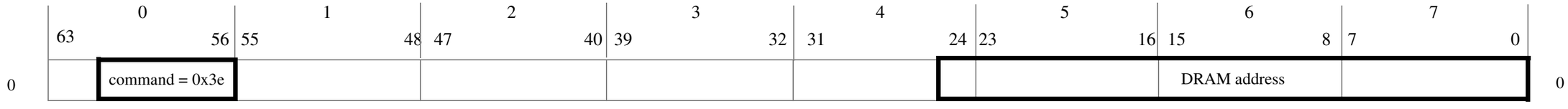


Table 5: Set Mask Image Command Format

Field	Word	Bits	Description
command	0	61-56	Command identifier
DRAM adrs	0	25-0	Base address (top left corner) of image in DRAM, in bytes

Set Tile

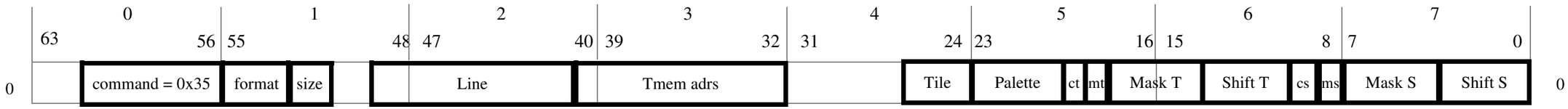


Table 6: Set Tile Command Format

Field	Word	Bits	Description
command	0	61-56	Command identifier
format	0	55-53	Image data format, 0 = rgba, 1 = yuv, 2 = Color Indx, 3 = IA, 4 = I
size	0	52-51	Size of pixel/texel color element, 0 = 4b, 1 = 8b, 2 = 16b, 3 = 32b, 4 = Other
Line	0	49-41	size of tile line in 64b words, max of 4KB
Tmem Adrs	0	40-32	Starting Tmem address for this tile in words (64b), 4KB range
tile	0	26-24	Tile descriptor index
Palette	0	23-20	Palette number for 4b Color Indexed texels. This number is used as the MS 4b of an 8b index.
ct	0	19	clamp enable for T direction
mt	0	18	mirror enable for T direction
Mask T	0	17-14	Mask for wrapping/mirroring in T direction. If this field is zero then clamp, otherwise pass (mask) LSBs of T address.
Shift T	0	13-10	Level of Detail shift for T addresses
cs	0	9	clamp enable bit for S direction
ms	0	8	mirror enable bit for S direction
Mask S	0	7-4	Mask for wrapping/mirroring in S direction. If this field is zero then clamp, otherwise pass (mask) LSBs of S address.
Shift S	0	3-0	Level of Detail shift for S addresses

Set Tile Usage Notes:

For YUV textures, Tile Line (number of Tmem words per tile line) is Line = (Width + 7) >> 3 (8b texels) because although the image texels are 16 bit, the Tmem Y texels are 8 bit and the Tmem UV texels are 16 bit at 1/2 the width in S.

YUV mask and mask/mirror are undefined

No mirroring for 32b RGBA images.

Wrap on all but YUV images.

YUV texture images are stored as interleaved 8 bit YUYV as 16 bits per texel. Software must specify texture coordinates and tile coordinates and sizes which are even numbers in S (lsb == 0), so that a pair UV values are available. Software must specify a Tile Base address in the low half of Tmem, and load a tile which fits in the low half (where 2048 < Base + W * H) in 8b Y's. Software can't have CI TLUT's and YUV textures coexisting in Tmem.

32b RGBA, YUV Set_Tile type should indicate 16b texels.

Mask == 1 means pass bit 0 so minimum mask width is 2 texels

Load Tile

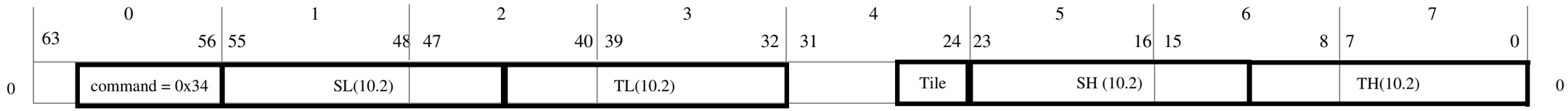


Table 7: Load Tile

Field	Word	Bits	Description
command	0	61-56	Command identifier
SL	0	55-44	Low S coordinate of tile in image
TL	0	43-32	Low T coordinate of tile in image
Tile	0	26-24	Tile descriptor index
SH	0	23-12	High S coordinate of tile in image
TH	0	11-0	High T coordinate of tile in image

Load Tile Usage Notes:

4b textures should be loaded as bytes (they must be byte aligned) using the Set_Image_Texture *type* field **when loading more than 4k texels**. This means the Load_Block parameters will have 8b texel units. The Set_Tile *type* field can be used to set the proper 4b type. The Set_Tile_Sz can be used to set the proper SL,SH,TL,TH values **after** the Load_Block. In general, textures can be loaded as one type and used as another type by proper manipulation of the Set_Image_Texture and Set_Tile *format,size* fields. Normally, during a load, the tile texel size and image pixel size should match.

YUV SL must be even and SH must be odd, which means the width is even, to provide a UV value at each texel. For bilinear interpolation, tile SH should be adjusted to tile laod SH-1, so that clamping occurs at a UV value.

For YUV and 32b RGBA images, software must specify a Tile Base address in the low half of Tmem, and load a tile which fits in the low half (where 2048 < Base + W * H in 8b Y’s).

For YUV and 32b RGBA images, can’t have color index TLUT’s in Tmem.

Fractional coordinates for L and H terms must usually be the same. Useful for subpixel coordinate offset in multi-tile operations.

Load Block

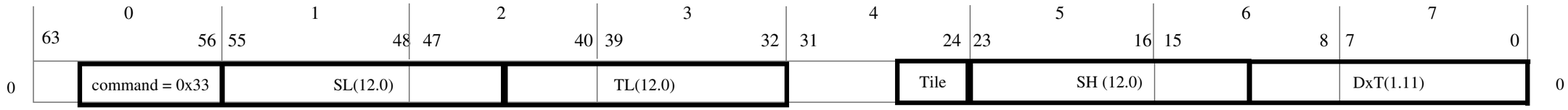


Table 8: Load Block

Field	Word	Bits	Description
command	0	61-56	Command identifier. Load Block loads a Tmem tile with a single memory “span” from SL,TL to SH,TL. During the tile load, the T coordinate is incremented by DxT every 8 Tmem bytes, in order to perform odd-line swapping and line strides.
SL	0	55-44	Low S coordinate of tile in image
TL	0	43-32	Low T coordinate of tile in image, the two MSBs of this number should be zero (i.e., a 10-bit number)
Tile	0	26-24	Tile descriptor index
SH	0	23-12	High S coordinate of tile in image
DxT	0	11-0	unsigned T increment value

Load Block Usage Notes:

A ceiling function must be performed on the 1.11 DxT field of Load_Block. That is, if any number has non-zero bits to the right of the 11-bit fraction, then a ceiling operation should be performed on the the number. For example, a 12 texel (16b/texel) wide texture would have a DxT of 1/3. The 11b fraction would be 1/3 * 2048 or 682 2/3. The ceiling is 683.

The texture image width must be multiples of 8 bytes. For example, a 4 bit texel texture must have an image width of n*16.

Each Load_Block command should be followed by Set_Tile_Sz command to set the actual tile SH,TH values.

4b textures should be loaded as bytes (they must be byte aligned) using the Set_Image_Texture *type* field **when loading more than 4k texels**. This means the Load_Block parameters will have 8b texel units. The Set_Tile *type* field can be used to set the proper 4b type **after** the Load_Block. The Set_Tile_Size can be used to set the proper SL,SH,TL,TH values **after** the Load_Block. In general, textures can be loaded as one type and used as another type by proper manipulation of the Set_Image_Texture and Set_Tile *format,size* fields. Normally, during a load, the tile texel size and image pixel size should match.

YUV texture images are stored as interleaved 8 bit YUYV as 16 bits per texel. Software must specify texture coordinates and tile coordinates and sizes which are even numbers in S (lsb == 0), so that a pair UV values are avaiable. Software must specify a Tile Base address in the low half of Tmem, and load a tile which fits in the low half (where 2048 < Base + W * H) in 8b Y’s. Software can’t have CI TLUT’s and YUV textures coexisting in

Tmem.

In Load_Block, the Tile line is the number of words to skip for each T. That is, zero for a contiguous tile.

When Load_Block is used to load multiple tiles of different widths (such as a mipmap pyramid), the image data in memory must be ordered for Tmem interleaved access. This means that for odd lines (t & 1), the two longs (32b) in each double (64b) must be swapped. Since Load_Block lines must consist of an integral number of double words, this is not effected by width. Note that Load_Tile performs this interleave during load, and Load_Block can perform this interleave by computing aT coordinate from DxT in the command. For memory image data which is interleaved, DxT should be zero.

Set Tile Size

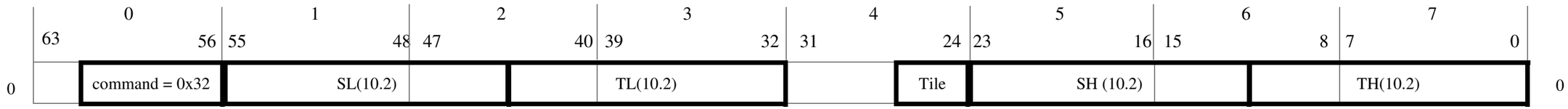


Table 9: Set Tile Size

Field	Word	Bits	Description
command	0	61-56	Command identifier
SL	0	55-44	Low S coordinate of tile in image
TL	0	43-32	Low T coordinate of tile in image
Tile	0	26-24	Tile descriptor index
SH	0	23-12	High S coordinate of tile in image
TH	0	11-0	High T coordinate of tile in image

Set Tile Size Usage Notes:

For YUV textures SL must be even and SH must be odd (this means the width is an even number of texels). Hardware will clamp to SH-1, that is, the last even texel, in order to have a valid UV at the last texel. If linear interpolation is performed, the max S texture coordinate must be odd, and SH is the max S + 2 (an odd SH),in order to supply a valid UV at the max S+1.

Load Tlut

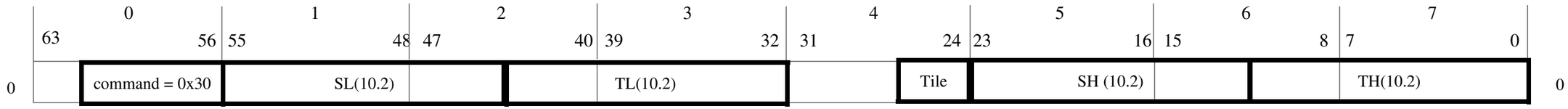


Table 10: Load Tlut

Field	Word	Bits	Description
command	0	61-56	Command identifier, this command is used to initiate a load from DRAM of a Indexed Texture Lookup Table (TLUT). This table dereferences color indexed texels before texture filtering.
SL	0	55-44	low index into table (0-255), fractional bits should be zero
TL	0	43-32	normally zero
Tile	0	26-24	Tile descriptor index
SH	0	23-12	high index into table (0-255), fractional bits should be zero
TH	0	11-0	normally zero

Load Tlut Usage Notes:

Use the set_image_texture to define the dram address, with a 16b type. Use set_tile to define the Tmem address. Reference that tile in load_tlut. The Texture Coordinate unit should get a DxS of 4 due to quadrication of table when loading.

Tmem Address must be in high half (msb == 1) of Tmem.

Tmem address in 64b words.

Triangle commands of various types are formed by concatonating groups of coefficients as shown in the table below. The order for concatonation is from left to right in the table. The formats for each group of coefficients are shown on the following pages.

Table 11: Triangle Commands

Command	Edge	Shade	Texture	ZBuffer
Non-ShadedTriangle, 0x08	4			
Shade Triangle, 0x0c	4	4		
Texture Triangle, 0x0a	4		4	
Shade, Texture Triangle, 0x0e	4	4	4	
Non-Shaded, ZBuff Triangle, 0x09	4			4
Shade, ZBuff Triangle, 0x0d	4	4		4
Texture, ZBuff Triangle, 0x0b	4		4	4
Shade, Texture, ZBuff Triangle, 0x0f	4	4	4	4

Edge Coefficients

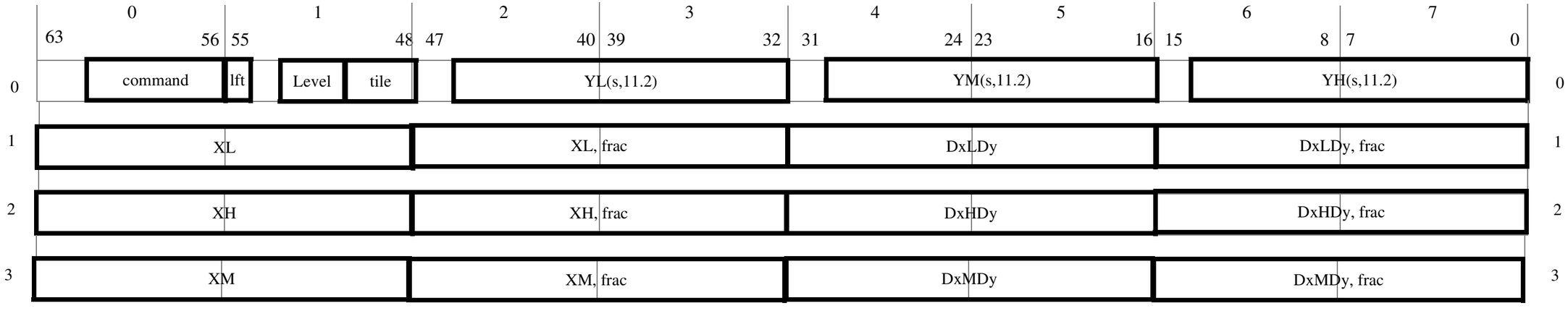


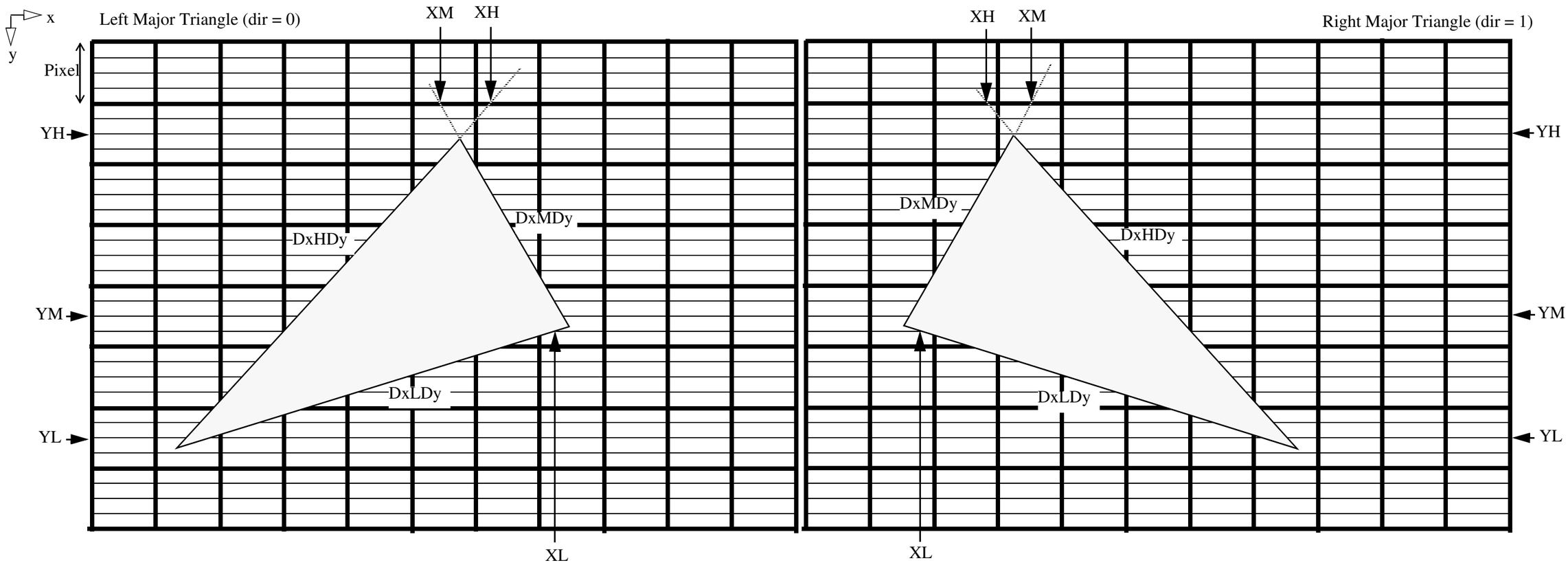
Table 12: Edge Coefficients

Field	Word	Bits	Description
command	0	61-56	Command identifier
lft	0	55	Left major flag, 0 = left major, 1 = right major
Level	0	53-51	number of mip-maps minus one
tile	0	50-48	Tile descriptor index. Used to reference texture for this primitive.
YL	0	45-32	Y coordinate of low minor edge
YM	0	29-16	Y coordinate of mid minor edge
YH	0	13-0	Y coordinate of major edge
XL	1	63-48	X coordinate of low edge, integer
XL, frac	1	47-32	X coordinate of low edge, fraction
DxLDy	1	31-16	Inverse slope of low edge, integer
DxLDy, frac	1	15-0	Inverse slope of low edge, fraction

Table 12: Edge Coefficients

Field	Word	Bits	Description
XH	2	63-48	X coordinate of major edge, integer
XH, frac	2	47-32	X coordinate of major edge, fraction
DxHDy	2	31-16	Inverse slope of major edge, integer
DxHDy, frac	2	15-0	Inverse slope of major edge, fraction
XM	3	63-48	X coordinate of middle edge, integer
XM, frac	3	47-32	X coordinate of middle edge, fraction
DxMDy	3	31-16	Inverse slope of middle edge, integer
DxMDy, frac	3	15-0	Inverse slope of middle edge, fraction

The edge coefficients are calculated at specific points on the view screen. The diagram below shows where each term is located. In general, Y terms are calculated to at least subpixel (1/4 pixel) resolution. The XM and XH are calculated where the H and M edges intersect the previous scan line. XL is calculated where the L edge intersects the next subpixel at or below the mid vertex.



Shade Coefficients

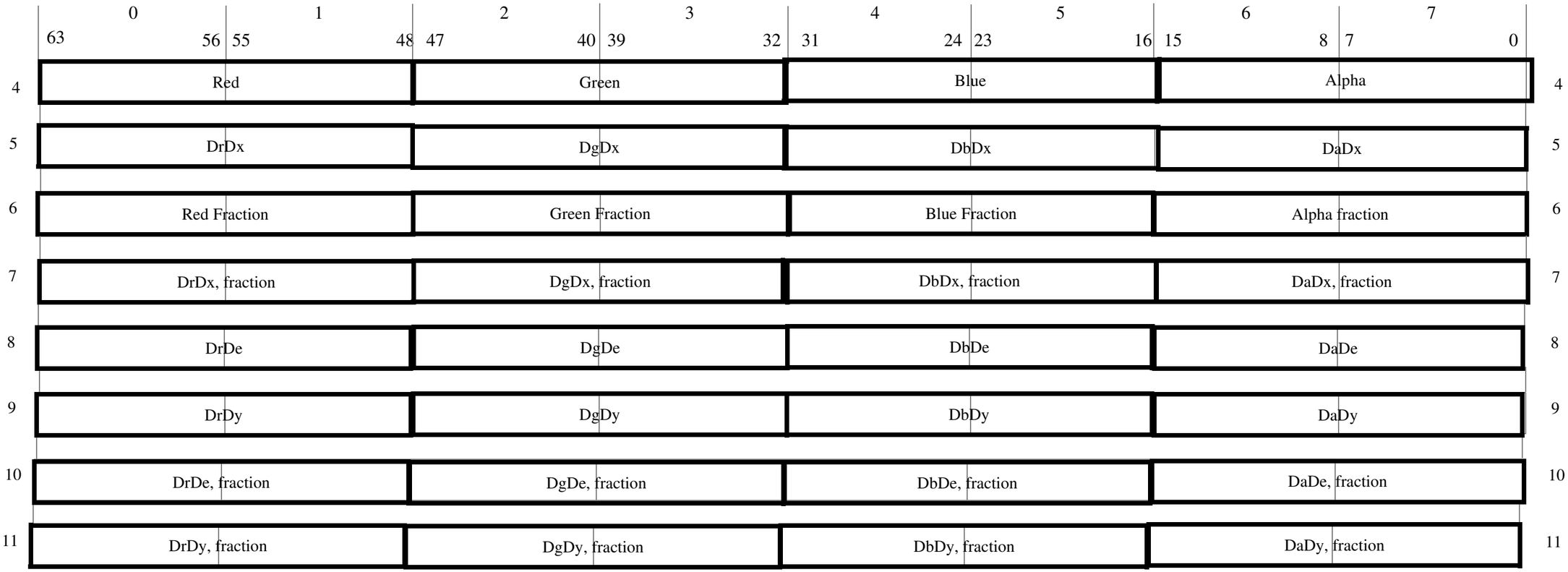


Table 13: Shade Coefficients

Field	Word	Bits	Description
Red	4	63-48	Red color component, integer
Green	4	47-32	Green color component, integer
Blue	4	31-16	Blue color component, integer
Alpha	4	15-0	Alpha color component, integer
DrDx	5	63-48	Change in red per change in X coordinate, integer
DgDx	5	47-32	Change in green per change in X xcoordinate, integer

Table 13: Shade Coefficients

Field	Word	Bits	Description
DbDx	5	31-16	Change in blue per change in X coordinate, integer
DaDx	5	15-0	Change in alpha per change in X coordinate, integer
Red, frac	6	63-48	Red color component, fraction
Green, frac	6	47-32	Green color component, fraction
Blue, frac	6	31-16	Blue color component, fraction
Alpha, frac	6	15-0	Alpha color component, fraction
DrDx, frac	7	63-48	Change in red per change in X coordinate, fraction
DgDx, frac	7	47-32	Change in green per change in X xcoordinate, fraction
DbDx, frac	7	31-16	Change in blue per change in X coordinate, fraction
DaDx, frac	7	15-0	Change in alpha per change in X coordinate, fraction
DrDe	8	63-48	Change in red along the edge, integer
DgDe	8	47-32	Change in green along the edge, integer
DbDe	8	31-16	Change in blue along the edge, integer
DaDe	8	15-0	Change in alpha along the edge, integer
DrDy	9	63-48	Change in red per change in Y coordinate, integer
DgDy	9	47-32	Change in green per change in Y coordinate, integer
DbDy	9	31-16	Change in blue per change in Y coordinate, integer
DaDy	9	15-0	Change in alpha per change in Y coordinate, integer
DrDe	10	63-48	Change in red along the edge, fraction
DgDe	10	47-32	Change in green along the edge, fraction
DbDe	10	31-16	Change in blue along the edge, fraction
DaDe	10	15-0	Change in alpha along the edge, fraction
DrDy	11	63-48	Change in red per change in Y coordinate, fraction
DgDy	11	47-32	Change in green per change in Y coordinate, fraction
DbDy	11	31-16	Change in blue per change in Y coordinate, fraction
DaDy	11	15-0	Change in alpha per change in Y coordinate, fraction

Texture Coefficients

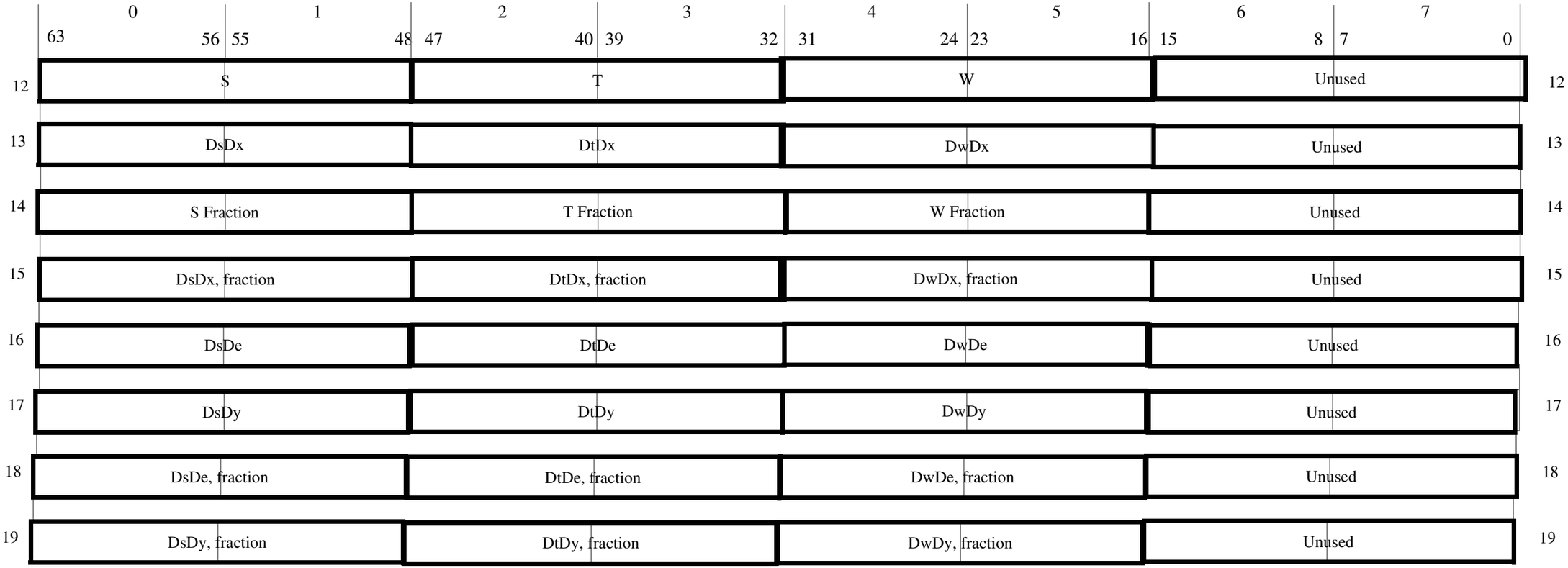


Table 14: Texture Coefficients

Field	Word	Bits	Description
S	12	63-48	S texture coordinate, integer
T	12	47-32	T texture coordinate, integer
W	12	32-16	Normalized inverse depth, integer
DsDx	13	63-48	Change in S per change in X coordinate, integer
DtDx	13	47-32	Change in T per change in X xcoordinate, integer
DwDx	13	31-16	Change in W per change in X coordinate, integer

Table 14: Texture Coefficients

Field	Word	Bits	Description
S, frac	14	63-48	S texture coordinate, fraction
T, frac	14	47-32	T texture coordinate, fraction
W, frac	14	32-16	Normalized inverse depth, fraction
DsDx, frac	15	63-48	Change in S per change in X coordinate, fraction
DtDx, frac	15	47-32	Change in T per change in X xcoordinate, fraction
DwDx, frac	15	31-16	Change in W per change in X coordinate, fraction
DsDe	16	63-48	Change in S along the edge, integer
DtDe	16	47-32	Change in T along the edge, integer
DwDe	16	31-16	Change in W along the edge, integer
DsDy	17	63-48	Change in S per change in Y coordinate, integer
DtDy	17	47-32	Change in T per change in Y coordinate, integer
DwDy	17	31-16	Change in W per change in Y coordinate, integer
DsDe, frac	18	63-48	Change in S along the edge, fraction
DtDe, frac	18	47-32	Change in T along the edge, fraction
DwDe, frac	18	31-16	Change in W along the edge, fraction
DsDy, frac	19	63-48	Change in S per change in Y coordinate, fraction
DtDy, frac	19	47-32	Change in T per change in Y coordinate, fraction
DwDy, frac	19	31-16	Change in W per change in Y coordinate, fraction

ZBuffer Coefficients

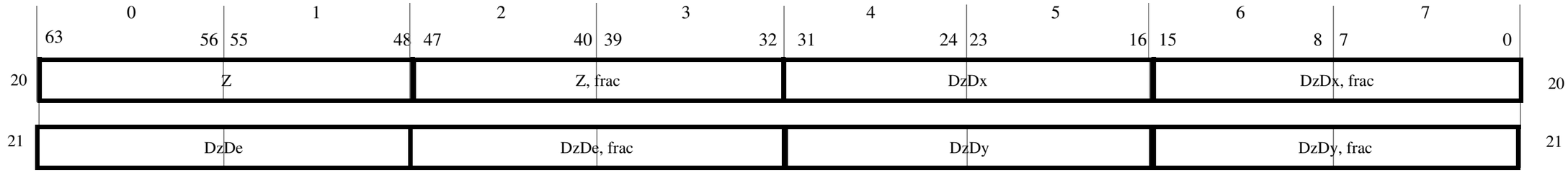


Table 15: ZBuffer Coefficients

Field	Word	Bits	Description
Z	20	63-48	Inverse Depth, integer
Z, frac	20	47-32	Inverse Depth, fraction
DzDx	20	31-16	Change in Z per change in X coordinate, integer
DzDx, frac	20	15-0	Change in Z per change in X coordinate, fraction
DzDe	21	63-48	Change in Z along major edge, integer
DzDe, frac	21	47-32	Change in Z along major edge, fraction
DzDy	21	31-16	Change in Z per change in Y coordinate, integer
DzDy, frac	23	15-0	Change in Z per change in Y coordinate, fraction

Fill Rectangle

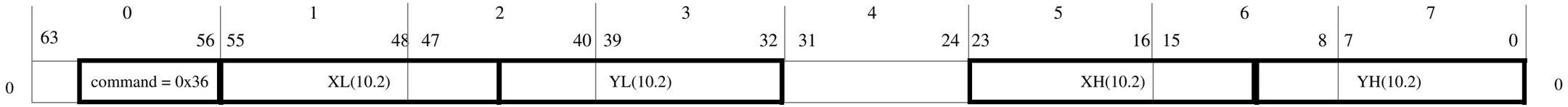


Table 16: Fill Rectangle

Field	Word	Bits	Description
command	0	61-56	Command identifier
XL	0	55-44	X coordinate of bottom right corner of rectangle
YL	0	43-32	Y coordinate of bottom right corner of rectangle
XH	0	23-12	X coordinate of top left corner of rectangle
YH	0	11-0	Y coordinate of top left corner of rectangle

Texture Rectangle

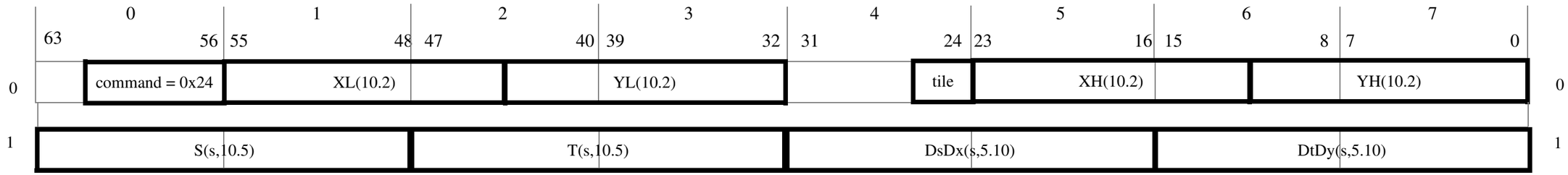


Table 17: Texture Rectangle

Field	Word	Bits	Description
command	0	61-56	Command identifier
XH	0	55-44	X coordinate of top left corner of rectangle
YH	0	43-32	Y coordinate of top left corner of rectangle
tile	0	26-24	Tile descriptor index
XL	0	23-12	X coordinate of bottom right corner of rectangle
YL	0	11-0	Y coordinate of bottom right corner of rectangle
S	1	63-48	S texture coordinate at top left corner of rectangle
T	1	47-32	T texture coordinate at top left corner of rectangle
DsDx	1	31-16	Change in S per change in X coordinate
DtDy	1	15-0	Change in T per change in Y coordinate

Texture Rectangle Usage Notes:

To copy an image, set cycle_type to “copy” in Set_Other_Modes, and set the combination of Set_Tile’s “shift S” and DsDx to step by 4 texels. No texture filtering, color combining, or blending operations are available for copied texels. Write enables may be generated using threshold compares of the alpha channel of each copied texel.

Texture Rextangle and Copy mode:

No Z-buffer or anti-aliasing in copy mode.

4b, YUV, and 32b RGBA textures cannot be directly copied.

4b images are expanded to 8b images before copying.

4b and 8b images can be copied to an 8b color image only.

16b image can be copied to a 16b color image only.

Note that 4b, YUV, and 32b RGBA images can be copied by using 8b or 16b color images and correctly scaling image coordinates and width.

Texture Rectangle Flip

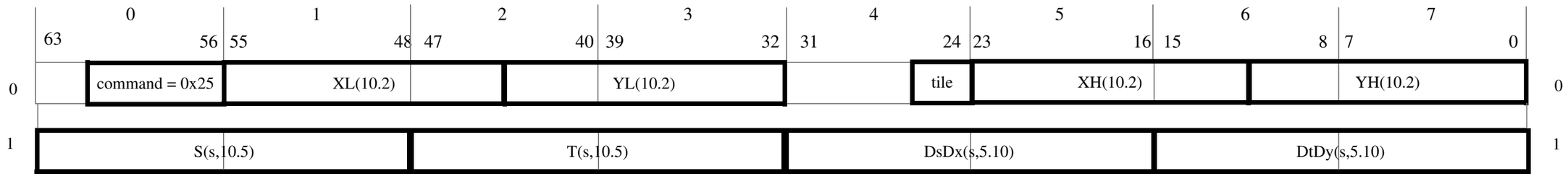


Table 18: Texture Rectangle Flip

Field	Word	Bits	Description
command	0	61-56	Command identifier, same as Texture Ractangle except hardware swaps S/T and DsDx/DtDy.
XH	0	55-44	X coordinate of top left corner of rectangle
YH	0	43-32	Y coordinate of top left corner of rectangle
tile	0	26-24	Tile descriptor index
XL	0	23-12	X coordinate of bottom right corner of rectangle
YL	0	11-0	Y coordinate of bottom right corner of rectangle
S	1	63-48	S texture coordinate at top left corner of rectangle
T	1	47-32	T texture coordinate at top left corner of rectangle
DsDx	1	31-16	Change in S per change in X coordinate
DtDy	1	15-0	Change in T per change in Y coordinate

Set Combine Mode

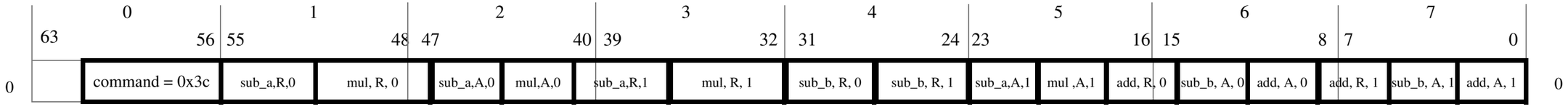


Table 19: Set Combine Mode

Field	Word	Bits	Description
command	0	61-56	Command identifier
sub_a, R, 0	0	55-52	sub_A input, RGB components, cycle 0
mul, R, 0	0	51-47	multiply input, RGB components, cycle 0
sub_a, A, 0	0	46-44	sub_A input, Alpha component, cycle 0
mul, A, 0	0	43-41	multiply input, Alpha component, cycle 0
sub_a, R, 1	0	40-37	sub_A input, RGB components, cycle 1
mul, R, 1	0	36-32	multiply input, RGB components, cycle 1
sub_b, R, 0	0	31-28	sub_B input, RGB components, cycle 0
sub_b, R, 1	0	27-24	sub_B input, RGB components, cycle 1
sub_a, A, 1	0	23-21	sub_A input, Alpha component, cycle 1
mul, A, 1	0	20-18	multiply input, Alpha component, cycle 1
add, R, 0	0	17-15	adder input, RGB components, cycle 0
sub_b, A, 0	0	14-12	sub_B input, Alpha components, cycle 0
add, A, 0	0	11-9	adder input, Alpha components, cycle 0
add, R, 1	0	8-6	adder input, RGB components, cycle 1
sub_b, A, 1	0	5-3	sub_B input, Alpha components, cycle 1
add, A, 1	0	2-0	adder input, Alpha components, cycle 1

Set_Combine_Mode Usage Notes:

The Color Combiner implements the equation: (A - B) * C + D on each color. RGB and Alpha channels have separate mux selects. In addition, there are separate mux selects for cycle 0 and cycle 1. If the RDP is configured for

one cycle mode, set the cycle 0 and cycle 1 mux selects to the same value.

\

Set Other Modes

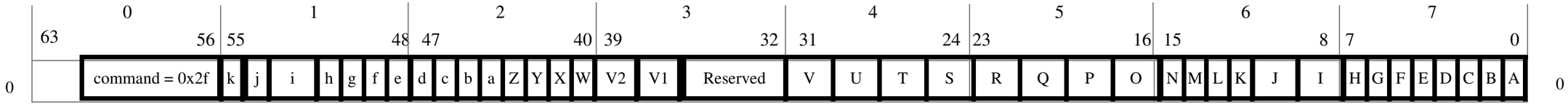


Table 20: Set Other Modes

Field	Word	Bits	Description
command	0	61:56	Command identifier
k: atomic_prim	0	55	Force primitive to be written to frame buffer before read of following primitive.
j:Reserved	0	54	This mode bit is not currently used but may be in the future
i: cycle_type	0	53:52	Display pipeline cycle control mode: 0 = 1 cycle, 1 = 2 cycle, 2 = Copy, 3 = Fill
h: persp_tex_en	0	51	enable perspective correction on texture
g: detail_tex_en	0	50	enable detail texture
f: sharpen_tex_en	0	49	enable sharpened texture
e: tex_lod_en	0	48	enable texture Level of Detail (LOD)
d: en_tlut	0	47	enable lookup of texel values from TLUT. Meaningful if texture type is index, tile is in low Tmem, TLUT is in high Tmem, and color image is RGB.
c: Tlut type	0	46	Type of texels in table, 0 = 16b RGBA(5/5/5/1), 1 = IA(8/8)
b: sample_type	0	45	determines how textures are sampled: 0 = 1x1 (Point Sample), 1 = 2x2. Note that copy (point sample 4 horizontally adjacent texels) mode is indicated by cycle_type.
a: mid_texel	0	44	indicates texture filter should do a 2x2 half texel interpolation, primarily used for MPEG motion compensation processing.
Z: bi_lerp_0	0	43	1 = bi_lerp, 0 = color convert operation in texture filter. Used in cycle 0
Y: bi_lerp_1	0	42	1 = bi_lerp, 0 = color convert operation in texture filter. Used in cycle 1
X: convert_one	0	41	Color convert texel that was the output of the texture filter on cycle 0, used to qualify bi_lerp_1
W: key_en	0	40	Enables chroma keying

Table 20: Set Other Modes

Field	Word	Bits	Description
V2: rgb_dither_sel	0	39:38	0 = magic square matrix (preferred if filtered) 1 = “standard” bayer matrix (preferred if not filtered) 2 = noise (as before) 3 = no dither
V1: alpha_dither_sel	0	37:36	0 = pattern 1 = ~pattern 2 = noise 3 = no dither
Reserved	0	35:32	Reserved for future use, default value is 0xf
V: b, m1a, 0	0	31:30	Blend modeword, multiply 1a input select, cycle 0
U: b, m1a, 1	0	29:28	Blend modeword, multiply 1a input select, cycle 1
T: b, m1b, 0	0	27:26	Blend modeword, multiply 1b input select, cycle 0
S: b, m1b, 1	0	25:24	Blend modeword, multiply 1b input select, cycle 1
R: b, m2a, 0	0	23:22	Blend modeword, multiply 2a input select, cycle 0
Q: b, m2a, 1	0	21:20	Blend modeword, multiply 2a input select, cycle 1
P: b, m2b, 0	0	19:18	Blend modeword, multiply 2b input select, cycle 0
O: b, m2b, 1	0	17:16	Blend modeword, multiply 2b input select, cycle 1
N: reserved	0	15	This mode bit is not currently used, but may be in the future
M: force_blend	0	14	force blend enable
L: alpha_cvg_select	0	13	use cvg (or cvg*alpha) for pixel alpha
K: cvg_times_alpha	0	12	use cvg times alpha for pixel alpha and coverage
J: z_mode[1:0]	0	11:10	0: opaque, 1: interpenetrating, 2: transparent, 3: decal
I: cvg_dest[1:0]	0	8:9	0: clamp (normal), 1: wrap (was assume full cvg), 2: zap (force to full cvg), 3:save (don’t overwrite memory cvg).
H: color_on_cvg	0	7	only update color on coverage overflow (transparent sufraces)
G: image_read_en	0	6	enable color/cvg read/modify/write memory access
F: z_update_en	0	5	enable writing of Z if color write enabled
E: z_compare_en	0	4	conditional color write enable on depth comparison
D: antialias_en	0	3	if not force blend, allow blend enable - use cvg bits
C: z_source_sel	0	2	choose between Primitive Z and pixel Z

Table 20: Set Other Modes

Field	Word	Bits	Description
B: dither_alpha_en	0	1	use random noise in alpha compare, otherwise use blend alpha in alpha compare
A: alpha_compare_en	0	0	conditional color write on alpha compare

Set_Other_Modes Usage Notes:

YUV Copy mode is not supported.

Fill mode in “Cycle Type” means replicate the Fill Color (see Set_Fill_Color).

Initialize the Z-buffer by setting the color image to point to the Z-buffer (Set_Color_Image) and filling with initial depth value.

Multi-tile mixed color index thru TLUT and other not supported, because TLUT effects all modes.

Set Env Color

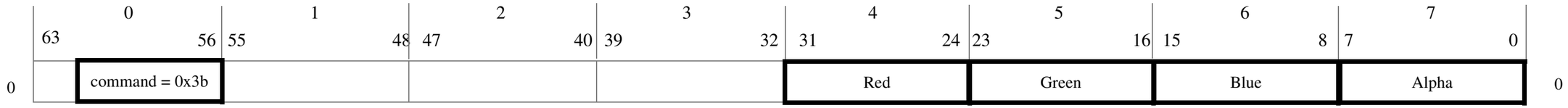


Table 21: Set Env Color

Field	Word	Bits	Description
command	0	61-56	Command identifier
Red	0	31-24	Red Component
Green	0	23-16	Green Component
Blue	0	15-8	Blue Component
Alpha	0	7-0	Alpha Component

Set Prim Color

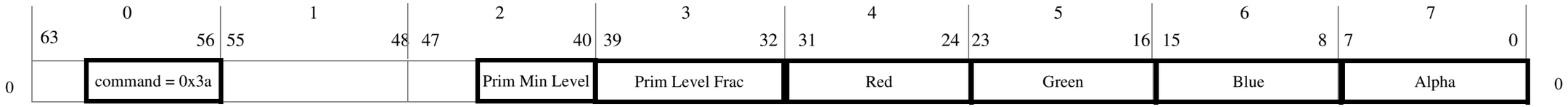


Table 22: Set Prim Color

Field	Word	Bits	Description
command	0	61-56	Command identifier
Prim Min Level	0	44-40	Minimum clamp for LOD fraction when in detail or sharpen texture modes, fixed point 0.5.
Prim Level Frac	0	39-32	Level of Detail fraction for primitive, used primarily in multi-tile operations for rectangle primitives, 0.8.
Green	0	23-16	Green Component
Blue	0	15-8	Blue Component
Alpha	0	7-0	Alpha Component

Set Blend Color

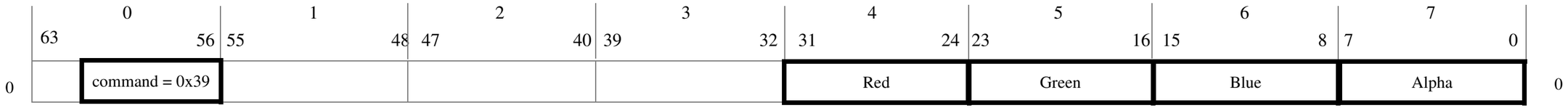


Table 23: Set Blend Color

Field	Word	Bits	Description
command	0	61-56	Command identifier
Red	0	31-24	Red Component
Green	0	23-16	Green Component
Blue	0	15-8	Blue Component
Alpha	0	7-0	Alpha Component

Set Fog Color

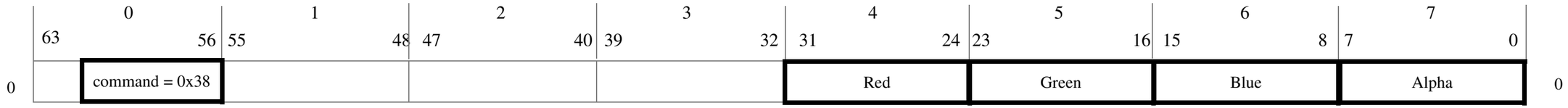


Table 24: Set Fog Color

Field	Word	Bits	Description
command	0	61-56	Command identifier
Red	0	31-24	Red Component
Green	0	23-16	Green Component
Blue	0	15-8	Blue Component
Alpha	0	7-0	Alpha Component

Set Fill Color

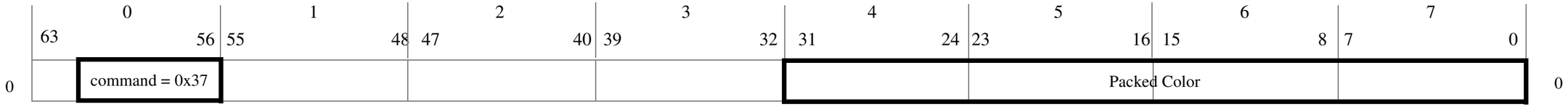


Table 25: Set Fill Color

Field	Word	Bits	Description
command	0	61-56	Command identifier
Packed Color	0	31-0	Packed Color, For example, if the Color Image was set be 16b RGBA, then the fill color would be two horizontally adjacent 16b RGBA pixels.

Set Prim Depth

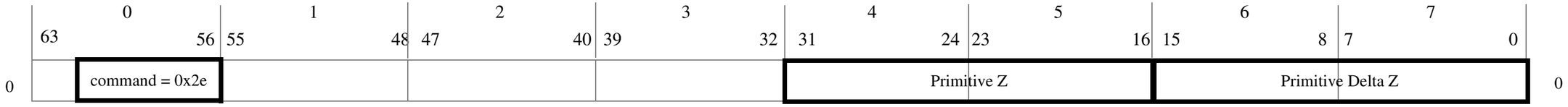


Table 26: Set Prim Depth

Field	Word	Bits	Description
command	0	61-56	Command identifier
Primitive Z	0	31-16	Primitive Z
Primitive Delta Z	0	15-0	Primitive Delta Z

Set Scissor

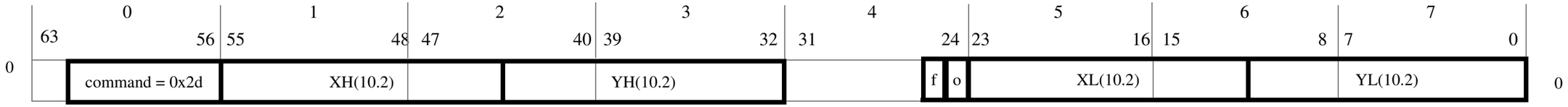


Table 27: Set Scissor

Field	Word	Bits	Description
command	0	61-56	Command identifier
XH	0	55-44	X coordinate of top left corner of scissor box in screen space.
YH	0	43-32	Y coordinate of top left corner of scissor box in screen space.
f	0	25	scissor field, enables scissoring of odd or even lines for interlaced displays
o	0	24	odd line: 0 = keep even line, 1 = keep odd line, indicates whether all odd lines or all even lines should be skipped (for interlaced displays).
XL	0	23-12	X coordinate of bottom right corner of scissor box in screen space.
YL	0	11-0	Y coordinate of bottom right corner of scissor box in screen space.

Set Convert

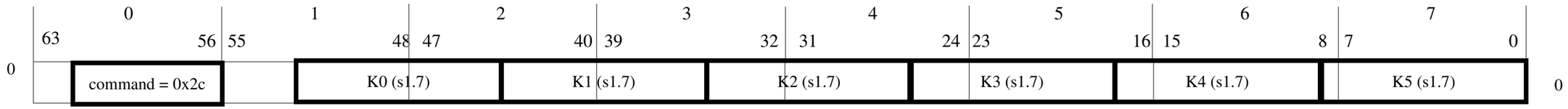


Table 28: Set Convert

Field	Word	Bits	Description
command	0	61-56	Command identifier, this command updates the coefficients for converting YUV pixels to RGB. Conceptually the equations are: R = C0*(Y-16) + C1*V G = C0*(Y-16) - C2*U - C3*V B = C0*(Y-16) + C4*U
K0	0	45-53	K0 term of YUV-RGB conversion matrix
K1	0	36-44	K1 term of YUV-RGB conversion matrix
K2	0	27-35	K2 term of YUV-RGB conversion matrix
K3	0	18-26	K3 term of YUV-RGB conversion matrix
K4	0	9-17	K4 term of YUV-RGB conversion matrix
K5	0	0-8	K5 term of YUV-RGB conversion matrix

Set_Convert Usage Notes:

In the hardware, the color conversion is done in two stages. In the texture filter (TF), the following equation is performed:

R’ = Y + K0*V
G’ = Y + K1*U + K2*V
B’ = Y + K3*U

In the color combiner, the following equations are performed:

R = (R’ - K4)*K5 + R’
G = (G’ - K4)*K5 + G’
B = (B’ - K4)*K5 + B’

Where (CX terms as defined in the table above):

K0 = C1 / C0
K1 = C2 / C0
K2 = C3 / C0
K3 = C4 / C0
K4 = 16 + 16 / (C0 - 1.0)
K5 = C0 - 1.0

Typical Values for YUV to RGB conversion:

K0 = 175
K1 = -43
K2 = -89
K3 = 222
K4 = 114
K5 = 42

Set Key R

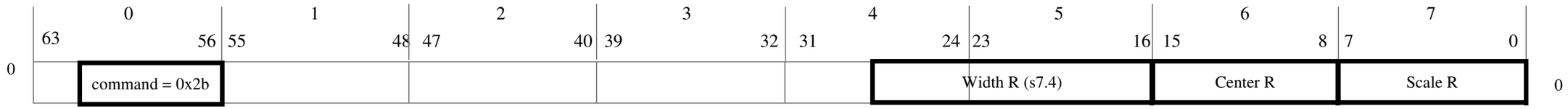


Table 29: Set Key R

Field	Word	Bits	Description
command	0	61-56	Command identifier, This command set the coefficients used for Red keying. The equation used for keying is: KeyR = clamp(0.0, -abs((R - Center) * Scale) + Width, 1.0). The Key Alpha is the minimum of the KeyR, KeyG, KeyB.
Width R	0	27-16	(Size of half the key window including the soft edge)*scale. If width > 1.0, then keying is disabled for that channel.
Center R	0	15-8	Defines color or intensity at which key is active, 0-255
Scale R	0	7-0	(1.0 / (size of soft edge). For hard edge keying, set scale to 255.

Set Key GB

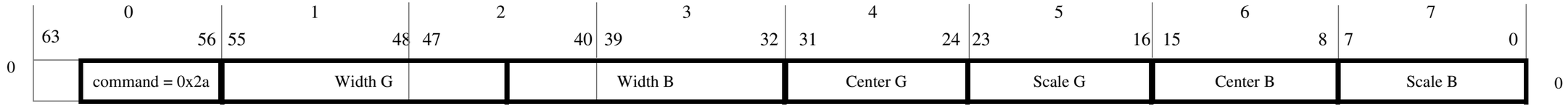


Table 30: Set Key GB

Field	Word	Bits	Description
command	0	61-56	Command identifier, This command set the coefficients used for Green/Blue keying. Conceptually, the equation used for keying is: KeyG/B = clamp(0.0, -abs((G/B - Center) * Scale) + Width, 1.0). The Key Alpha is the minimum of the KeyR, KeyG, KeyB.
Width G	0	55-44	(Size of half the key window including the soft edge)*scale. If width > 1.0, then keying is disabled for that channel.
Width B	0	43-32	(Size of half the key window including the soft edge)*scale. If width > 1.0, then keying is disabled for that channel.
Center G	0	31-24	Defines color or intensity at which key is active, 0-255
Scale G	0	23-16	(1.0 / (size of soft edge). For hard edge keying, set scale to 255.
Center B	0	15-8	Defines color or intensity at which key is active, 0-255
Scale B	0	7-0	(1.0 / (size of soft edge). For hard edge keying, set scale to 255.

Set_Key_XX Usage Notes:

In the hardware, the keying equation is performed in two stages. In the Color Combiner (CC), the equation peformed is:

Key’ = (pixel - Center) * Scale + 0

In the Alpha Fixup unit (AF), the equation performed is:

Key = clamp(0, -abs(Key’) + Width, 1.0)
KeyAlpha = MIN(KeyR, KeyG, KeyB)

In two-cycle mode, the keying operation must be specified in the second cycle (that is, the key alpha is not available as a combine operand).

Sync Full

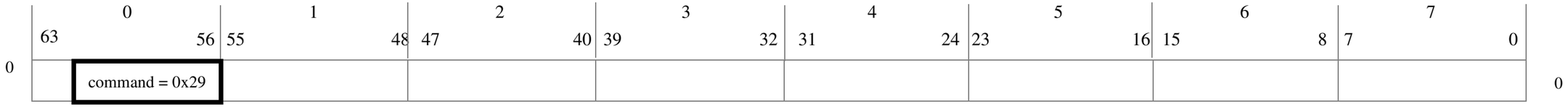


Table 31: Sync Full

Field	Word	Bits	Description
command	0	61-56	Command identifier, This command stalls the RDP until the last dram buffer is read or written from any preceeding primitive. It is typically only needed if the memory data is to be reused, like switching display buffers, or writing a color_image to be used as a texture_image, or for consistent r/w access to an RDP w/r image from the cpu.

Sync Load

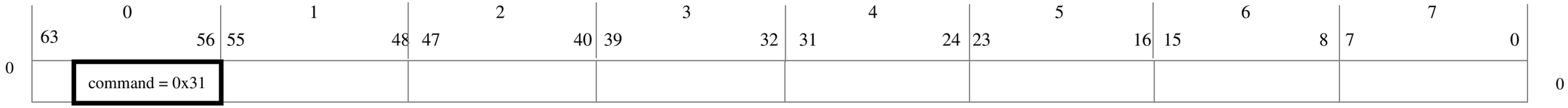


Table 1: Sync Load

Field	Word	Bits	Description
command	0	61-56	Command identifier, This command stalls the execution of load commands (loadTLUT, load tile, load block) until preceeding primitives has completely finished. Usually preceed all load commands by sync load.

Sync Pipe

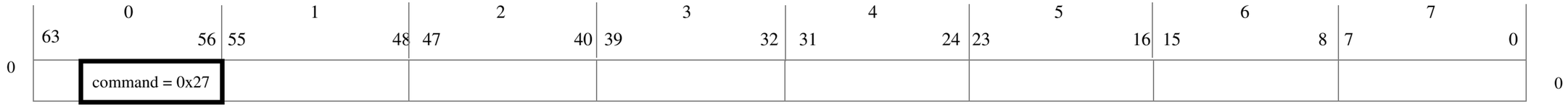


Table 32: Sync Pipe

Field	Word	Bits	Description
command	0	61-56	Command identifier, General attributes (other than prim_color/depth) which are being read by up to two preceeding primitives should be preceeded by sync_pipe, which stalls until the most recent primitive is past the last usage of any attribute. Only one sync_pipe is needed before any number of attribute commands. Software can optimize sync_pipe usage if it knows what is being read, for example, a set_texture_image can follow tri's or rects in preparation for a load_tile without a sync_pipe, because tris or rects don't use the texture_image attribute. [in general, the RDP is optimized for a number of primitives rendered with the same attribute setting. If attributes change per primitive, performance will degrade slightly.

Sync Tile

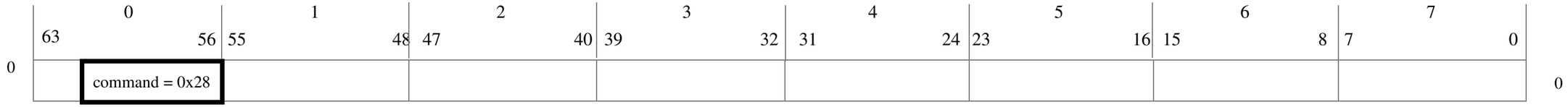


Table 33: Sync Tile

Field	Word	Bits	Description
command	0	61-56	Command identifier, Allows synchroniztion between commands that write to the same tile descriptor that an immediately previous command is reading.

No Op

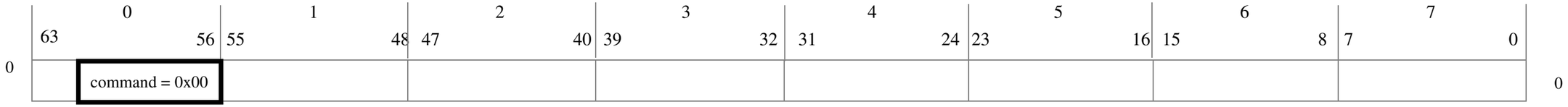


Table 34: No Op

Field	Word	Bits	Description
command	0	61-56	Command identifier, This command has no effect on rdp command execution but is useful for padding command buffers.

Index