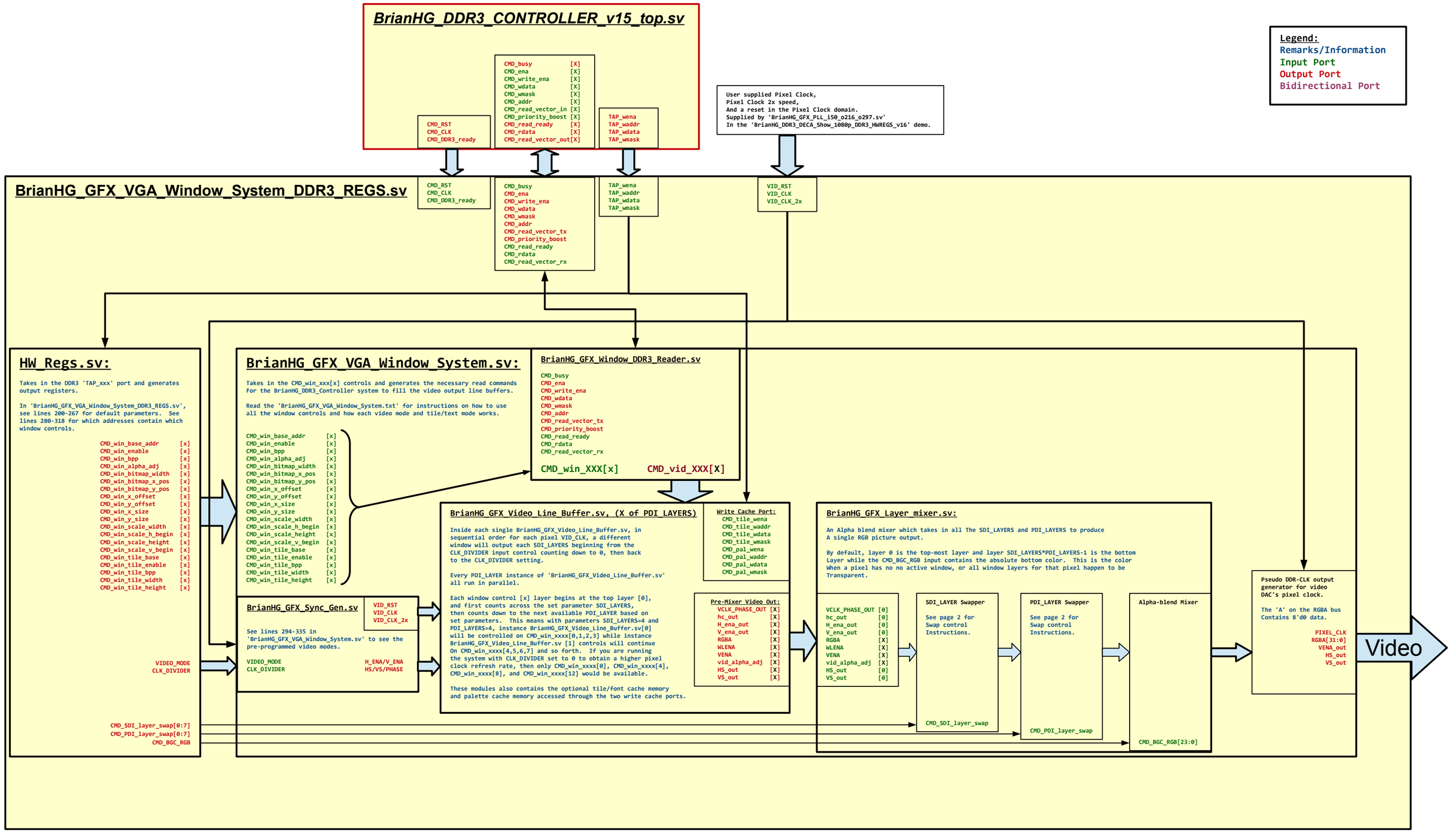
BrianHG_DDR3_CONTROLLER_v15_top.sv
CMD_busy [X]
CMD_ena [X]
CMD_write_ena [X]
CMD_wdata [X]
CMD_wmask [X]
CMD_addr [X]
CMD_read_vector_in [X]
CMD_priority_boost [X]
CMD_read_ready [X]
CMD_rdata [X]
CMD_read_vector_out[X]
CMD_RST
CMD_CLK
CMD_DDR3_ready
TAP_wena
TAP_waddr
TAP_wdata
TAP_wmask
User supplied Pixel Clock,
Pixel Clock 2x speed,
And a reset in the Pixel Clock domain.
Supplied by 'BrianHG_GFX_PLL_i50_o216_o297.sv'
In the 'BrianHG_DDR3_DECA_Show_1080p_DDR3_HWREGS_v16' demo.
Legend:
Remarks/Information
Input Port
Output Port
Bidirectional Port
BrianHG_GFX_VGA_Window_System_DDR3_REGS.sv
VID_RST
VID_CLK
VID_CLK_2x
HW_Regs.sv:
Takes in the DDR3 'TAP_xxx' port and generates
output registers.
In 'BrianHG_GFX_VGA_Window_System_DDR3_REGS.sv',
see lines 200-267 for default parameters. See
lines 280-318 for which addresses contain which
window controls.
CMD_win_base_addr [x]
CMD_win_enable [x]
CMD_win_bpp [x]
CMD_win_alpha_adj [x]
CMD_win_bitmap_width [x]
CMD_win_bitmap_x_pos [x]
CMD_win_bitmap_y_pos [x]
CMD_win_x_offset [x]
CMD_win_y_offset [x]
CMD_win_x_size [x]
CMD_win_y_size [x]
CMD_win_scale_width [x]
CMD_win_scale_h_begin [x]
CMD_win_scale_height [x]
CMD_win_scale_v_begin [x]
CMD_win_tile_base [x]
CMD_win_tile_enable [x]
CMD_win_tile_bpp [x]
CMD_win_tile_width [x]
CMD_win_tile_height [x]
VIDEO_MODE
CLK_DIVIDER
CMD_SDI_layer_swap[0:7]
CMD_PDI_layer_swap[0:7]
CMD_BGC_RGB
BrianHG_GFX_VGA_Window_System.sv:
Takes in the CMD_win_xxx[x] controls and generates the necessary read commands
For the BrianHG_DDR3_Controller system to fill the video output line buffers.
Read the 'BrianHG_GFX_VGA_Window_System.txt' for instructions on how to use
all the window controls and how each video mode and tile/text mode works.
CMD_win_base_addr [x]
CMD_win_enable [x]
CMD_win_bpp [x]
CMD_win_alpha_adj [x]
CMD_win_bitmap_width [x]
CMD_win_bitmap_x_pos [x]
CMD_win_bitmap_y_pos [x]
CMD_win_x_offset [x]
CMD_win_y_offset [x]
CMD_win_x_size [x]
CMD_win_y_size [x]
CMD_win_scale_width [x]
CMD_win_scale_h_begin [x]
CMD_win_scale_height [x]
CMD_win_scale_v_begin [x]
CMD_win_tile_base [x]
CMD_win_tile_enable [x]
CMD_win_tile_bpp [x]
CMD_win_tile_width [x]
CMD_win_tile_height [x]
BrianHG_GFX_Sync_Gen.sv
See lines 294-335 in
'BrianHG_GFX_VGA_Window_System.sv' to see the
pre-programmed video modes.
VIDEO_MODE
CLK_DIVIDER
H_ENA/V_ENA
HS/VS/PHASE
BrianHG_GFX_Window_DDR3_Reader.sv
CMD_busy
CMD_ena
CMD_write_ena
CMD_wdata
CMD_wmask
CMD_addr
CMD_read_vector_tx
CMD_priority_boost
CMD_read_ready
CMD_rdata
CMD_read_vector_rx
CMD_win_XXX[x]
CMD_vid_XXX[X]
BrianHG_GFX_Video_Line_Buffer.sv, (X of PDI_LAYERS)
Inside each single BrianHG_GFX_Video_Line_Buffer.sv, in
sequential order for each pixel VID_CLK, a different
window will output each SDI_LAYERS beginning from the
CLK_DIVIDER input control counting down to 0, then back
to the CLK_DIVIDER setting.
Every PDI_LAYER instance of 'BrianHG_GFX_Video_Line_Buffer.sv'
all run in parallel.
Each window control [x] layer begins at the top layer [0],
and first counts across the set parameter SDI_LAYERS,
then counts down to the next available PDI_LAYER based on
set parameters. This means with parameters SDI_LAYERS=4 and
PDI_LAYERS=4, instance BrianHG_GFX_Video_Line_Buffer.sv[0]
will be controlled on CMD_win_xxxx[0,1,2,3] while instance
BrianHG_GFX_Video_Line_Buffer.sv [1] controls will continue
On CMD_win_xxxx[4,5,6,7] and so forth. If you are running
the system with CLK_DIVIDER set to 0 to obtain a higher pixel
clock refresh rate, then only CMD_win_xxxx[0], CMD_win_xxxx[4],
CMD_win_xxxx[8], and CMD_win_xxxx[12] would be available.
These modules also contains the optional tile/font cache memory
and palette cache memory accessed through the two write cache ports.
Write Cache Port:
CMD_tile_wena
CMD_tile_waddr
CMD_tile_wdata
CMD_tile_wmask
CMD_pal_wena
CMD_pal_waddr
CMD_pal_wdata
CMD_pal_wmask
Pre-Mixer Video Out:
VCLK_PHASE_OUT [X]
hc_out [X]
H_ena_out [X]
V_ena_out [X]
RGBA [X]
WLENA [X]
VENA [X]
vid_alpha_adj [X]
HS_out [X]
VS_out [X]
BrianHG_GFX_Layer_mixer.sv:
An Alpha blend mixer which takes in all The SDI_LAYERS and PDI_LAYERS to produce
A single RGB picture output.
By default, layer 0 is the top-most layer and layer SDI_LAYERS*PDI_LAYERS-1 is the bottom
Layer while the CMD_BGC_RGB input contains the absolute bottom color. This is the color
When a pixel has no no active window, or all window layers for that pixel happen to be
Transparent.
VCLK_PHASE_OUT [0]
hc_out [0]
H_ena_out [0]
V_ena_out [0]
RGBA [X]
WLENA [X]
VENA [X]
vid_alpha_adj [X]
HS_out [0]
VS_out [0]
SDI_LAYER Swapper
See page 2 for
Swap control
Instructions.
CMD_SDI_layer_swap
PDI_LAYER Swapper
See page 2 for
Swap control
Instructions.
CMD_PDI_layer_swap
Alpha-blend Mixer
CMD_BGC_RGB[23:0]
Pseudo DDR-CLK output
generator for video
DAC's pixel clock.
The 'A' on the RGBA bus
Contains 8'd0 data.
PIXEL_CLK
RGBA[31:0]
VENA_out
HS_out
VS_out
Video

BrianHG_GFX_Layer_mixer.sv: Assuming parameters SDI_LAYERS=8 and PDI_LAYERS=8, here is an illustration of how to set the CMD_SDI_layer_swap[0:7] and CMD_PDI_layer_swap[0:7].

Input from each PDI line buffer

Each CMD_SDI_layer_swap[x] where bits [2:0] will 'XOR' swap the source green input channel number to the output red channel number and bits [6:4] will perform an 'ADD' offset from input To output channel number. IE: A horizontal swap.

After the each CMD_SDI_layer_swap[x], a vertical swap is performed using the CMD_PDI_layer_swap[x] for each new SDI_LAYER position. This 'vertical' swap has a separate control for each SDI_LAYER. Bits [2:0] logically 'XOR' the vertical address position while bits [6:4] perform an 'ADD' offset to the vertical position.

New output for each PDI layer to be sent to the Alpha Blend Layer mixer. Layer 0x0 will be the top-most layer.