FPGA/ASIC Display Interface Module

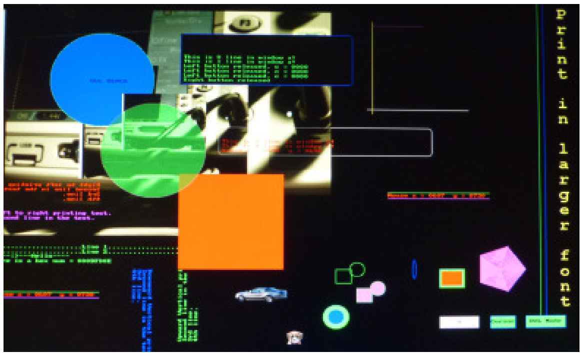
VIDEO OUPUT FAST AND EASY

- LCD/VGA/HDMI/Custom
- Multi Frame Buffer Support
- Multi Overlay Buffer support
- HW Pointer Support
- Full Color BMP Pointer Support
- Configurable Frame Timing
- Configurable Frame size
- HW Transparency/Mixing
- HW Frame Dimming
- FPGA Technology Independent
- Features vs Size Options
- Efficient 16-bit Pixel Format
- Anti-shear Parameter Switching
- Selectable Interrupts on FS/HS
- Configurable HW Block Transfer
- HW Test Pattern Generator
- Multiple Overlay Modes
- Color Key for Block Transfers
- Color Key for BMP Draws

RICH SW API

- Customizable Fonts, Font Draw
- Font Fore&Back Color Settings
- Printf() Directly to Background
- Printf() Directly to Overlay
- Selectable Text Orientation
- TextWindow/Terminal Widget
- Graph Windows, Button WIdgets
- Polygon/Ellipse/Circle Draws
- Selectable Fill vs. Outline Colors
- Bitmap Draws
- Frame Buffer Clearing
- Draw Orientation Selection
- Transparent Color Key Selection
- Transparency Setting Function
- Full Frame Fade Function
- <3k Logic Elements
- Utility for BMP Generation

- Small HDL Based Design
- Fully Synchronous
- Rich SW API
- Fully Pipelined for Highest Performance
- Simple Slave/Register Configuration Interface
- Simple Master Interface to memory (for pixel fetching)
- Easy to Use - Quickest Time from Concept to UI Realization
- Pointer Cache Option for Full Color Pointer Support in HW
- Integrates with other Tectonic and Third Party Video HW modules



Make your FPGA design come alive with vibrant colors and dynamic, rich, output.

The Tectonic series of graphic and video IP modules make is easier than ever to get output from your FPGA/ASIC to the screen. The Display Interface module is a core module that provides the interface directly to an LCD module, or through and external triple DAC or HDMI transmitter, directly to any standard video display.

The module also builds in many must haves for engineers needing to accelerate the time from instantiation to displaying information on the screen in a professional format. Many important features are performed at the hardware level, offloading the processor such as, transparency and mixing of the overlay and the background frame, hardware pointer overlay, full frame dimming, and block memory transfers into selected frame buffer memory.

We Make It Move

RICH SW API

Many Primitive Calls Exposed

SW based and HW based

HW Acceleration on Many Calls

Additional API calls Under Development

Contact:

Sales@logic-tectonics.com

815 975-7070

www.logic-tectonics.com

API (Q2/2015)

Function call	Input	Operation
fill_frame()	Frame buffer pointer, color	Fills the frame memory with the specified color
draw_circle()	Frame buffer pointer, x/y center, fill color, outline color, border thickness	Draws a circle
draw_line()	Frame buffer pointer, start x/y, end xy, color	Draw a line in the selected frame buffer
draw_rectangle()	Frame buffer pointer, upper left corner x/y, y height, x width, fill color, outline color, border thickness	Draw a rectangle in the selected frame buffer
draw_polygon()	Frame buffer pointer, upper x/y center, radius, number of sides, offset angle, color.	Draw a rectangle in the selected frame buffer
put_pixel()	Frame buffer pointer, x,y pixel location, color	This is a primitive api call to draw an individual pixel
bmp_to_fb()	Frame buffer pointer, array containing a 16bit pixel bitmap, x /y location in the frame, transparent pixel value (allowing non rectangular transfers)	Draws a bitmap in the designed location on the screen. Tectonics provides a utility to convert any 24-bit BMP to a compatible data array in c format for including in the users code base.
init_fonts()	Font header file with available fonts.	Initializes and loads the fonts before first use
put_char()	Frame buffer pointer, x/y location to put the character, the character to print, character color, background color, font id	Draw a character at the location specified in the frame buffer specified.
put_string()	Frame buffer pointer, x/y location to put the string, the string (ASCII) to print, character color, background color, font id	Draw a character at the location specified in the frame buffer specified.
prinfb()	Frame buffer pointer, x/y location, text color, background color (including transparent), font id, standard printf() format sting and variable list.	Using standard printf() formatting will print to the selected frame buffer.
bmp_to_cursor()	Bitmap array name, transparent color key	Updates the HW cursor with the bmp image for rich 16 bit color cursor/pointer.
init_textwin()	Frame buffer pointer, text window parameters, lines, columns, font id, colors, etc.	Initializes a text window to allow terminal like scrolling text line output.
printf_textwin()	Text window id, standard printf() format sting and variable list.	Used to print a line to the specified text window.
block_xfer()	Pointer to source fb/mem, pointer to dest fb/mem, source x/y loc, block x/y dimension, dest x/y loc, mode, color, replacement mode, replacement color key	Very fast HW based block transfer method to move data from one location to the next, clear screen, replace with certain color, and can use transparent key for non - rectangular support. Used for fast frame buffer fills/clears, and movement of data for PIP, and other user functions.
use_fb()	Frame buffer pointer	Selects the frame buffer to display/make visible.
set_ovl()	Overlay frame buffer pointer, overlay transparency setting, mode.	Used to select and enable an overlay to be used and the transparency settings and mode. Modes include overlay replacement of background, mix where overlap, or full mix
Init_button(), set_button_state()	Button location, bmp, color, size, etc	Create buttons and then manages the button operation

We Make It Move