Summary

Alpha blending, color keying, layers sizing, layers positioning, and cursors are often misunderstood features of display graphics controllers. This white paper provides a basic overview of the Xylon's logiCVC-ML Compact Multilayer Video Controller IP core for Xilinx® FPGAs and its multilayering capabilities.

Introduction

The logiCVC-ML - Compact Multilayer Video Controller is a graphic video controller optimized for Xilinx FPGAs. This FPGA IP core has rich feature set tailored especially for embedded electronic devices.

It controls TFT flat panel displays and, by means of external video converters, S-Video, Composite Video devices and CRT displays (i.e. VGA monitors).

Multilayer support provides on screen display functions: Alpha Blending, Color Keyed Transparency among layers, Hardware Cursors, and Fast Scrolling and Pan functions. All features are hardware supported and require low CPU processing power.

Interface to frame buffers or video memory is designed for SDRM (SDR, DDR, DDR2) or SRAM implementations. For easier system integration, the logiCVC-ML uses standard CoreConnect™ PLB and OPB buses. The IP requires relatively low memory bandwidth and can be efficiently implemented in low-cost systems featuring Unified Memory Architecture (UMA).

Being designed in parametric VHDL and fully embedded into Xilinx Platform Studio and EDK tools, the logiCVC-ML is very configurable IP core. This tight integration with Xilinx's integrated development environment tremendously shortens IP integration and verification time. Simulation and implementation of logicBRICKS IP cores into targeted designs do not require any particular skills beyond general Xilinx tools knowledge.

Optional processing functions, like bit-block unit, frame grabbing, etc. can be easily added by integration with other graphic logicBRICKS IP cores.

The logiCVC-ML is well proven FPGA IP, tested and used in volume applications serving the most demanding electronics markets.

logiCVC-ML Features

- Supports all latest Xilinx FPGA families
- Supports LCD and CRT displays (easily tailored for special display types)
- 64x1 to 2048x2048 display resolutions
- Higher resolutions supportable on request
- Up to 5 layers; the last on configurable as a background color
- Configurable layers' size, position and offset
- Alpha blending and Color keyed transparency
- Pixel, Layer, or Color Lookup Table (CLUT) alpha blending mode can be independently setup for each layer

- Packed pixel layer memory organization – pixel color depth 8 bpp, 8 bpp using CLUT, 16 bpp Hicolor RGB 5-6-5 and TrueColor 24 bpp
- Configurable PLB or XMB memory interface data width (32, 64 or 128 bits)
- Programmable layer memory base address and stride
- Simple programming of control registers through OPB or PLB interfaces
- Programmable display data bus: 12-bit, 15-bit, 16-bit, 18-bit or 24-bit
- Supports digital video output (ITU-656: PAL and NTSC)
- Supports LVDS and camera link output formats
- Supports synchronization to RGB input (data used for one layer)
- HW cursors
- Double/triple buffering enables flicker free reproduction
- Display power-on sequencing control signals

Multilayer and Alpha Blending Types

The logiCVC-ML simultaneously controls up to 5 independent display layers. The number of layers is configurable at the code's synthesis time. The layers do not have to have the same size (resolution) or pixel color depth. Their positioning within the frame buffer is programmable too. In addition, the layers can be overlaid and scrolled fast.

Overlaying in combination with the Color Key Transparency and the Alpha Blending enables creation of attractive graphics effects, i.e. fading in/out or scroll in/out of computer generated objects over a 'live' video picture displayed on an independent layer.

The creation of such video effects without a hardware support is very CPU processing demanding, and can cause performance limitations in optimized embedded systems.
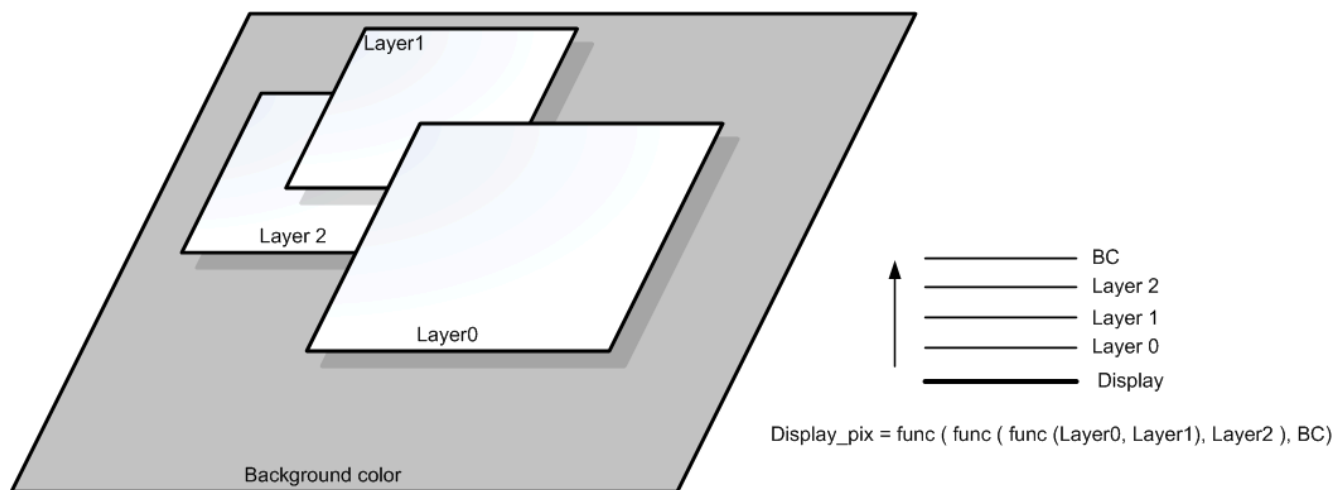

$$Display\_pix = func (func (func (Layer0, Layer1), Layer2), BC)$$

Figure 1: Example configuration with 3 layers

Color Key Transparency

If layers do not overlap, the resulting displayed image is a simple combination of layer images with a fixed background color. The resulting image of overlapping layers can be a combination of transparency and fading defined by Alpha Blending.

Layer transparency is supported by Color Key Transparency function. Each logiCVC-ML controller's layer has an independently programmable color key that can be freely chosen from layer's color palette. Consequently, a number of colors that can be displayed on n-bit layer is 2^n-1.

Layer's pixels set to selected color key values are invisible (transparent) on a screen, and actually replaced by a pixel from the layer underneath. The final pixel shown on the display is a result of multiple transparency settings starting from the background color up to the top layer.

Layer Alpha Blending

The Alpha Blending operations create fade effects. There are three Alpha Blending modes: Layer, Pixel and CLUT. The simplest mode is the Layer Alpha Blending mode. The Layer Alpha Blending operations work simultaneously with all pixels on two layers.

Overlapping pixels of two layers are merged into resulting image following this equation:

$res\_pix(x, y) = alpha(i) * layer(i)\_pix(x, y) + (1 - alpha(i))\ layer(i - 1)\_pix(x, y)$

, where i is a layer's ordinal number between 0 and 4. When $i = 0$, the background color represents the $(i-1)$ layer

alpha(i) is a programmed alpha factor for *layer(i)*. Its value is between 0 and 1, and must be programmed as 8-bit binary value presenting fractional numbers in this range.

x and *y* are pixel positions (horizontal and vertical)

The *alpha(i)* factor set to 1 produces resulting pixel *res_pix(x,y)* equal to the corresponding pixel from the *layer(i)*. The resulting *res_pix(x, y)* pixel is equal to the pixel from the *layer(i - 1)* when *alpha(i)* factor is set to 0. Alpha factor's settings between 0 and 1 produce blended resulting pixels.

The fade out effect from *layer(i)* to *layer(i - 1)* can be programmed by decreasing *alpha(i)* factor from 1 downto 0. The number of blending levels depends on 8-bit normalization and color pixel depth of the particular layer.

The Alpha Blending calculations are executed between layer pairs, starting from the bottom background color and the *layer(0)*. The resulting blended image is further blended by *layer(1)*'s image, and this sequence is executed till the last layer and its alpha factor.

Pixel Alpha Blending

The Pixel Alpha Blending requires different alpha factors assigned to each pixel. Therefore, values of each pixel in a video frame buffer are combinations of the alpha factor and the color value. The Pixel Alpha Blending operations work at the pixel level.

This mode of Alpha Blending enables variable fading of different pixel screen (layer) positions. This feature is commonly used to anti-alias edges of overlapping graphic objects, or graphic object and a 'live' video. An image without visible saw steps can be obtained by setting alpha factors to 1 for pixels in the center of the object, decreasing them from 1 to 0 towards the object's edges, and by setting them to 0 for pixels belonging to the other graphic objects or 'live' video.

The following formula describes the Pixel Alpha Blending:

$res\_pix(x, y) = alpha(x, y) * layer(i)\_pix(x, y) + (1 - alpha(x, y))\ layer(i - 1)\_pix(x, y)$

, where i is a layer's ordinal number between 0 and 4. When $i = 0$, the background color represents the $(i-1)$ layer

alpha(x, y) is the alpha factor for each pixel, in a range of 0 to 1

x and *y* are pixel positions (horizontal and vertical)

The Pixel Alpha Blending mode's disadvantages are its increased memory space consumption, and more important, higher memory bandwidth requirement. A CPU must re-write the entire image (new alpha factors) in order to change fade effects, and this is an additional disadvantage of this blending type.

Even so the Pixel Alpha Blending imposes a bit higher HW requirements; its qualities are unprecedented by other blending modes. The Pixel Alpha Blending is the most sophisticated blending type.

CLUT Alpha Blending

The CLUT Alpha Blending mode is used for blending 8-bit layers. Such layers use 8-bit pixel information, and a color lookup table (CLUT) that expands this 8-bit information to 32-bit pixel value. The expanded pixel value consists of 8-bit alpha factor and 24-bit color value. Effectively such a layer can display 256 selected colors from a 16M color palette. Since each selected color has an adjoined alpha factor, they can be independently faded.

This Alpha Blending type requires a low memory bandwidth and the CPU processing power for fading effects. Since the alpha factors are assigned to particular colors, graphic objects having the same color fade simultaneously. The objects' positioning is irrelevant, and different parts of a layer fade at the same time if they are equally colored.

Layer Size and Positioning

A layer sizing and positioning enables fast hardware scrolling and pan effects. It is important to note that smaller layer's size decreases required memory bandwidth that is usually crucial for overall performance of the embedded electronic device.
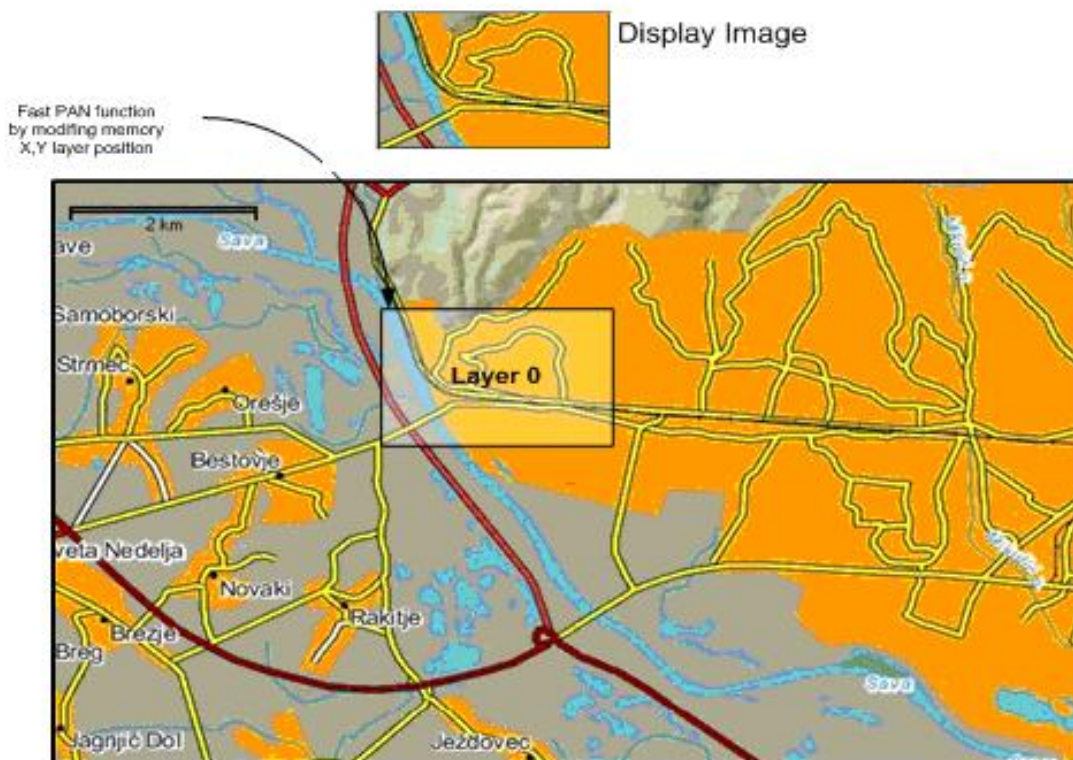

Figure 2: Fast Hardware PAN

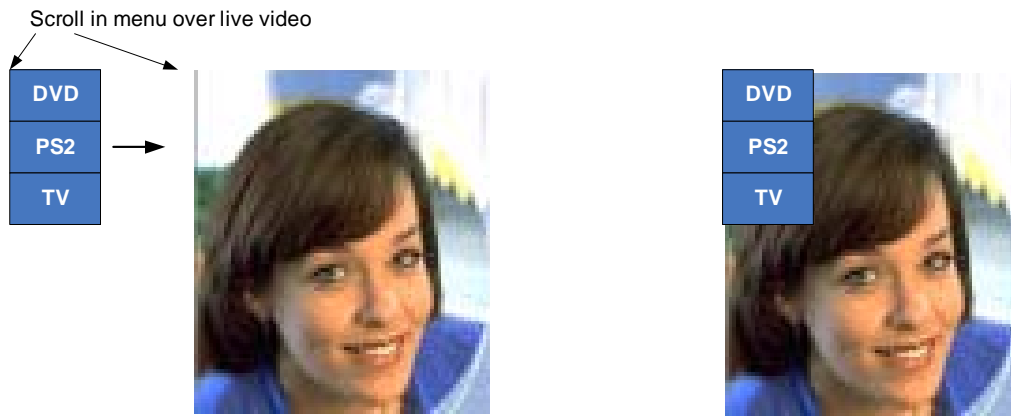Scroll in menu over live video

Figure 3: Scroll in/out function

Hardware image scrolling in combination with Alpha Blending can produce attractive video effects, i.e. fading menus that appear from one side of display and gradually fade in/out.

An illustrative example of scrolling feature would be a simultaneous scrolling of two neighboring graphic windows that scroll at different paces.

HW cursor

Four of five logiCVC-ML controller's layers can be configured as a HW cursor. The HW cursor's pixels are stored in an external video frame buffer. The HW cursor is in fact just a standard logiCVC-ML layer configured for usually small image size. Due to the Cursor's small image size, it consumes very low memory bandwidth. All layer modes are supported; therefore cursor can be 8bpp CLUT, 16bpp or even 24bpp color image with any alpha blending mode.

Revision History

Version	Date	Note
1.00	11.04.2006.	Initial Release
1.10	16.12.2009.	Updated logiCVC-ML info.