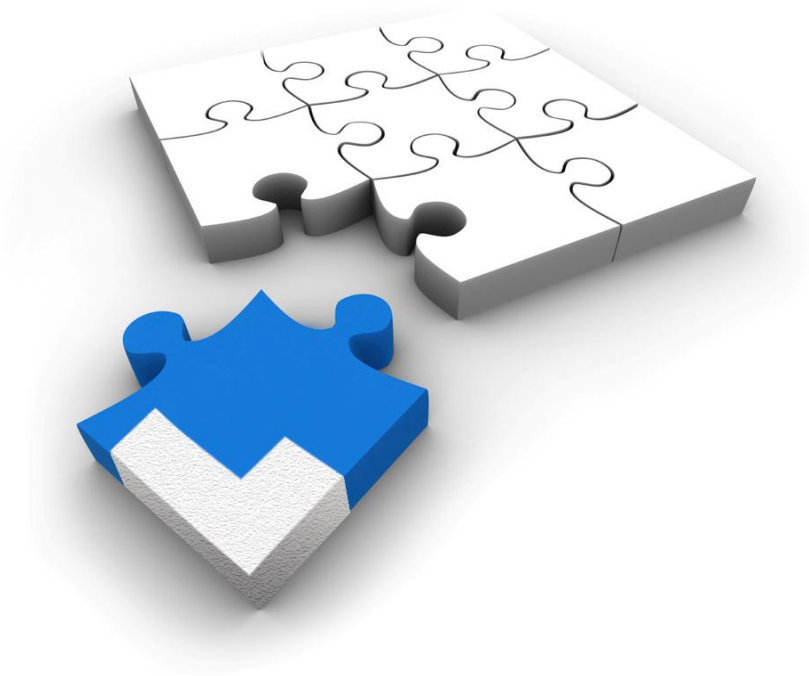
Xylon Linux Framebuffer Driver

For Use with Xylon's Display Controller IP Core – logiCVC-ML Compact Multilayer Video Controller

User's Manual

Version: 2.01.a

Xylon-Linux-FrameBuffer_v2_01_a.doc



1 INTRODUCTION 4
2 LOGICVC-ML COMPACT MULTILAYER VIDEO CONTROLLER 6
3 XYLON FRAMEBUFFER SOFTWARE DRIVER 7
3.1 VERSIONING 7
3.2 FEATURES 7
3.3 FRAMEBUFFER DRIVER FILE STRUCTURE 8
3.4 XYLON FRAMEBUFFER DRIVER TYPES 8
3.4.1 Open Firmware Type 8
3.4.2 Platform Type: 9
3.5 SUPPORT FOR MULTIPLE LOGICVC-ML LAYERS 9
3.6 PIXEL CLOCK SUPPORT 11
3.7 KERNEL BOOT ARGUMENTS 11
3.7.1 Console and video mode settings 11
3.7.2 Mapping of the physical memory to logiCVC-ML layers 12
3.8 FRAMEBUFFER DRIVER CONFIGURATION 12
3.8.1 Open Firmware Framebuffer Driver Configuration 12
3.8.2 Platform Framebuffer driver configuration 15
3.9 FRAMEBUFFER DRIVER MISCELLANEOUS FUNCTIONALITY 17
3.10 FRAMEBUFFER DRIVER MULTIPLE INSTANCES 17
3.11 FRAMEBUFFER DRIVER RESOLUTION SETTINGS 17
3.12 FRAMEBUFFER DRIVER CUSTOM IOCTLS 18
4 BUILDING XYLON FRAMEBUFFER DRIVER FOR XILINX ARM LINUX KERNEL 19
5 XYLON FRAMEBUFFER DTS SNIPPET 21
6 REVISION HISTORY 24

1 INTRODUCTION

Xylon's logicBRICKS library of IP cores optimized for Xilinx programmable devices includes graphics logicBRICKS IP cores for full range implementation of 2D and 3D Graphics Processing Units (GPU) on Xilinx® Zynq™-7000 All Programmable SoC and FPGAs.

Depending on graphics requirements, designers can use one or more logicBRICKS IP cores to build the graphics engine through an easy plug-and-play Xilinx Platform Studio (XPS) design flow. The minimum GPU configuration requires the logiCVC-ML display controller IP core to interface the system processor with the LCD or other types of graphics displays.

The logiCVC-ML Compact Multilayer Video Controller supports many advanced graphic features and enables easy interfacing with different displays. Xylon provides extensive logicBRICKS software support to enable software developers to work efficiently with popular graphic libraries, widget toolkits and familiar development tools.

This User's Manual briefly describes Xylon's Framebuffer software driver optimized for the logiCVC-ML display controller IP core and Linux operating system. The same software driver can be used with the Android™ operating system.

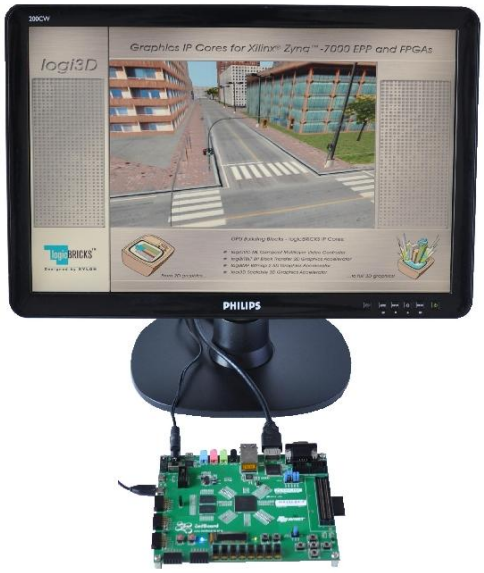


Figure 1: The ZedBoard™ Development Board from Avnet Electronics Marketing - Running Xylon's 3D Graphics Demo – Linux OS

Linux Framebuffer software driver is a standard Linux driver that abstracts the graphic hardware and allows application software to access the hardware through a well-defined interface. Software designers can use it with no need to know anything about the underlying hardware (IP cores) in Xilinx Zynq-7000 All Programmable SoC and FPGA devices.

Where to Get the Xylon Framebuffer Driver?

To get the latest Xylon's Linux software drivers, please visit https://github.com/logicbricks.

How to Try It?

Xylon Framebuffer is standard Linux driver and can be used with any Xilinx FPGA or Zynq-7000 SoC based system running Linux or Android operating systems. Xylon offers pre-verified reference designs for popular evaluation kits. Designs include evaluation logicBRICKS IP cores, hardware design files, complete Linux OS image, software drivers, demo applications and documentation. Please check Xylon's Video Gallery web pages (http://www.logicbricks.com/logicBRICKS-IP-Library/Video-Galleries.aspx) to see the demonstrated graphics demo applications.

Available Xylon reference designs:

1) Graphics Engine for the Xilinx ZC702 Evaluation Kit

http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design/Graphics-for-Xilinx-Zynq-7000.aspx

2) Human Machine Interface (HMI) for the Xilinx ZC702 Evaluation Kit

http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design/HMI-for-Xilinx-Zynq-7000.aspx

3) Graphics Engine for the ZedBoard from Avnet Electronics Marketing

http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design/Graphics-for-Zynq-AP-SoC-ZedBoard.aspx

Additional Xylon logicBRICKS GPU Software Support

logicBRICKS IP cores can be delivered with software drivers for the most popular operating systems: Linux, Microsoft® Windows® Embedded Compact and Android.

For use with the Linux operating system Xylon also offers:

- Direct Frame Buffer (DirectFB) - a thin library that provides hardware graphics acceleration, input device handling and abstraction, integrated windowing system with support for translucent windows and multiple display layers
- OpenGL® ES 1.1 - a royalty-free, cross-platform API for full-function 2D and 3D graphics on embedded systems - including consoles, phones, appliances and vehicles.

For more information about Xylon products, please contact info@logicbricks.com.

2 LOGICVC-ML COMPACT MULTILAYER VIDEO CONTROLLER

The logiCVC-ML IP core is an advanced display graphics controller for LCD and CRT displays, which enables an easy video and graphics integration into embedded systems with Xilinx Zynq-7000 All Programmable SoC and FPGAs.

This IP core is the cornerstone of all 2D and 3D GPUs. Though its main function is to provide flexible display control, it also includes a level of hardware acceleration: alpha blendings, panning, buffering of multiple frames, etc.

- Supports all Xilinx FPGA families
- Supports LCD and CRT displays (easily tailored for special display types)
- 64x1 to 2048x2048 display resolutions
- Support for higher display resolutions available on request
- Supports up to 5 layers; the last one configurable as a background layer
- 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 set for each layer
- Packed pixel layer memory organization
  - RGB pixel color depth 8-bpp, 8-bpp using CLUT, 16bpp Hi-color RGB 565 and True-Color 24bpp
  - YUV pixel color depth 8-bpp using CLUT, 16 bpp, 32 bpp
- Configurable CoreConnect™ PLBv4.6, Xylon XMB or ARM® AMBA® AXI4 memory interface data width (32, 64 or 128)
- Programmable layer memory base address and stride
- Simple programming due to small number of control registers
- Support for multiple output formats:
  - Parallel display data bus: 12x2-bit, 15-bit, 16-bit, 18-bit or 24-bit
  - Digital Video ITU-656: PAL and NTSC
  - LVDS output format: 3 or 4 data pairs plus clock
  - Camera link output format: 4 data pairs plus clock
  - DVI output format
  - YCbCr 4:4:4 or 4:2:2 output format
- Supports synchronization to external parallel input
- HW cursors
- Versatile and programmable sync signals timing
- Double/triple buffering enables flicker-free reproduction
- Display power-on sequencing control signals
- Parametrical VHDL design that allows tuning of slice consumption and features set
- Prepared for Xilinx Platform Studio (XPS) and the EDK

More info in Xylon's web shop: http://www.logicbricks.com/Products/logiCVC-ML.aspx
Datasheet: http://www.logicbricks.com/Documentation/Datasheets/IP/logiCVC-ML_hds.pdf

3 XYLON FRAMEBUFFER SOFTWARE DRIVER

3.1 Versioning

Driver version	Supported logiCVC-ML CORE version	NOTES
1.0	2.05.c	Initial driver version
1.1	2.05.c	Added logiCLK* support
1.2	2.05.c, 3.00.a	YUV support
2.1	3.00.a, 3.01.a, 3.02.a	Added EDID functionality with interface to the ADV7511 V4L2 driver Supported multiple driver instances Resolution settings improvements

Driver History

* Learn more about this auxiliary IP core:
http://www.logicbricks.com/Documentation/Datasheets/IP/logiCLK_hds.pdf

3.2 Features

Xylon Framebuffer Driver is the Linux kernel driver. It supports the following logiCVC-ML display controller's features:

- Resolutions from 64x1 up to 2048x2048
- Up to 5 layers, last configurable as background color (example: video overlaid with an HMI)
- Configurable layers' size and position
- Alpha blending and Color keyed transparency (Pixel, Layer or Color LUT alpha blending)
- Packed pixel layer memory organization
  - RGB
    - pixel color depth 8bpp (Color Look-Up Table (CLUT))
    - 16 bits per pixel HiColor RGB 5-6-5
    - TrueColor 24bpp
  - YUV
    - pixel color depth 8bpp (CLUT table with the AYUV8888 only!)
    - 16 bits per pixel (4:2:2)
    - 32 bits per pixel (4:4:4) : memory layout => AYUV8888
- Programmable layer memory base address and stride
- Double/triple buffering enables flicker-free reproduction
- Display power-on sequencing control signals
- Pixel clock frequency supported with:

- Zynq-7000 Processing System (PS) Clock Generation Module
- Si570 10 MHz – 1.4 GHz DSPLL® from Silicon Laboratories (used on the ZC702 board)
- logiCLK IP core (Please check the logiREF-ZGPU-ZED GPU Reference Design User's Manual, Chapter 6. Video Output Clocking with an auxiliary IP core http://www.logicbricks.com/Documentation/Datasheets/IP/logiREF-ZGPU-ZED_UM.pdf)

- Miscellaneous:

  - Interfacing the ADV7511 HDMI transmitter V4L2 driver to get EDID parameters and initialize the logiCVC-ML display controller IP core accordingly to the preferred display resolution

3.3 Framebuffer Driver File Structure

Xylon Framebuffer (FB) driver is located in the Linux kernel tree folder *drivers/video/xylon/xylonfb*.

Driver folder structure:

- *drivers/video/xylon/xylonfb/core*
  -> Xylon FB driver core, pixel clock generation functionality, IOCTL functionality

- *drivers/video/xylon/xylonfb/misc*
  -> Xylon FB miscellaneous functionality: ADV7511 HDMI transmitter interface driver

- *drivers/video/xylon/xylonfb/of*
  -> Open Firmware driver

- *drivers/video/xylon/xylonfb/platform*
  -> Platform driver

3.4 Xylon Framebuffer Driver Types

Xylon framebuffer driver types:

- Open Firmware type
- Platform type

3.4.1 Open Firmware Type

Linux kernel contains the Open Firmware (OF) module to provide various information and parameters about the underlying hardware to kernel drivers; at the kernel driver's loading time or the kernel module's insertion time. The parameters are loaded from the Device Tree Binary *.dtb* binary file, and stored in memory where they wait for specific OF driver to use them. The Linux kernel must be configured (compiled) to support the OF in order to enable and use the OF information transfers. Device Tree Source *.dts* file is used as an input into the Device Tree parsing script located in the Linux kernel tree that generates the *.dtb* file.

The Open Firmware functionality is especially useful in embedded systems based on programmable FPGA and SoC devices, since the programmable devices can often change internal architecture (configuration). When the SoC/FPGA hardware changes, and the programmable logic device configuration file (.dtb) is separated from the Linux kernel, any hardware parameter can be changed and used by the kernel driver with no need for recompiling Linux kernel driver or Linux kernel driver module. When added, the new hardware (SoC/FPGA IP core) functionality can be supported with new device parameters in the driver's code and the Device Tree Source *.dts* (Device Tree Binary *.dtb)* file. In this case the kernel driver must be recompiled.

3.4.2 Platform Type:

Platform drivers are used for devices that are usually autonomous entities in the system. Xylon logiCVC-ML display controller IP core can be used like that, since the display graphics or the frame buffer console is not an obligatory Linux functionality and can be started at any moment. Platform type Linux drivers have all hardware configuration parameters hardcoded in driver's *platform_device* structure, and can be accessed by specific kernel interface functions.

The platform driver type drivers must be recompiled every time when any hardcoded hardware parameter is changed. Embedded systems usually have Linux kernel compiled at remote *host* machine and not directly on embedded file system, so a fully functional driver can become unusable because of e.g. a change of a single HW parameter which impacts embedded system's behavior.

3.5 Support for multiple logiCVC-ML layers

Xylon Framebuffer driver supports multiple logiCVC-ML layers. The layers' pixel type, blending order and memory locations are defined by the SoC/FPGA design. Framebuffer driver receives the logiCVC-ML layer configuration through the *.dts* or platform in-built structures, depending on the driver type (Open Firmware or Platform). For details see chapters 3.8.1.1 and 3.8.2.1.1.

Xylon Framebuffer driver will create the following Linux kernel device files for each logiCVC-ML layer:

- */dev/fb0*
- */dev/fb1*
- ..
- */dev/fbN*, where N=number of logiCVC-ML layers -1

logiCVC-ML layer is enabled when the */dev/fbN* file is opened and disabled when the file is closed.

By default, the logiCVC-ML layer 0 (*/dev/fb0*) is used for the Linux kernel console.

Other layers can be also selected for the Linux kernel console by setting up the *.dts* parameter "*xylon-video-params->active-layer"*, or in the platform driver *xylonfb_platform_data.active_layer*.

When "*active-layer"* is set to e.g. 1 then the logiCVC-ML layer 1 is represented with the file /dev/fb0 and the other layers will be numbered in sequential order.

Examples:

logiCVC-ML implements four layers and "*xylon-video-params->active-layer"=0*

Layer ordering in the /dev folder:

/dev/fb0 – Layer0
/dev/fb1 – Layer1
/dev/fb2 – Layer2
/dev/fb3 – Layer3

logiCVC-ML implements four layers and "*xylon-video-params->active-layer"=2*

Layer ordering in the /dev folder:

/dev/fb0 – Layer2
/dev/fb1 – Layer0
/dev/fb2 – Layer1
/dev/fb3 – Layer3

3.6 Pixel clock support

Xylon Framebuffer driver supports several pixel clock sources. The Linux kernel configuration menu enables selection of one of the supported options:

Drivers -> Graphics support -> Support for frame buffer devices -> Xylon logiCVC frame buffer support -> Xylon logiCVC pixel clock source:

- **External pixel clock** - not controlled by the driver; assumed externally set pixel clock
- **Zynq PS pixel clock** – internal Zynq PS (Processing System) clock generator is used as the pixel clock source. The PS clock source has a limited precision and many popular pixel clock frequencies cannot be generated exactly, which results by a strange monitor behavior, i.e. no image
- **Xylon logiCLK pixel clock** – support for the auxiliary logiCLK IP core that enables generation of accurate pixel clock frequencies (0.05% tolerance)
- **ZC702 board pixel clock –** support for the Si570 DSPLL chip mounted on the board

The pixel clock source configuration depends on the FPGA/Zynq-7000 SoC design. The logiCVC-ML pixel clock input can be connected to the Zynq PS clock generator, Si570 clock generator on the ZC702 board, the logiCLK auxiliary IP core implemented in the programmable logic, or to some external clock source device.

Xylon Framebuffer driver support is designed to enable the logiCVC-ML IP core use of external (on-board) or internal (in SoC/FPGA) hardware devices to achieve various advanced driver functionality. Xylon's reference designs for Xilinx ZC702 or Avnet Electronic Marketing ZedBoard evaluation/development kits use Xylon Framebuffer driver support to control on-board external HDMI transmitter and to drive standard PC monitors. This support can be expanded on as-needed basis.

3.7 Kernel boot arguments

3.7.1 Console and video mode settings

Linux kernel console is enabled by the following line in the kernel boot arguments:

console=tty0

Default video VESA video mode can be selected from the kernel arguments; please see *Documentation/fb/modedb.txt for details*. Chapter 3.8.1.4 explains video modes priorities.

video=xylonfb:800x600M-32@60

3.7.2 Mapping of the physical memory to logiCVC-ML layers

Xylon Framebuffer driver needs to map large memory areas of contiguous physical memory to the logiCVC-ML layers. Physical memory for the logiCVC-ML layers is statically determined in the SoC/FPGA design by the logiCVC generics configuration.

For example, a single layer with RGB32 pixel format, double buffered, max 1080 vertical resolution requires (STRIDE * VRES * BUFFRES_NUM * BPP = 2048 * 1080 * 2 * 4) 16 MB of memory.

Xylon recommends constraining of the Linux kernel to "see" less memory and setting up of the logiCVC-ML video memory at the end of the available RAM.

The following example assumes 1024 MB of the available RAM, and at the end of the RAM, 256 MB of the memory reserved for the logiCVC-ML and other video applications:

mem=768M

mem=768M – limits Linux kernel system RAM to 768 MB

3.8 Framebuffer Driver Configuration

Xylon Framebuffer driver can be configured (see Chapter 3.4) as follows:

- Open Firmware configuration
- Platform configuration

3.8.1 Open Firmware Framebuffer Driver Configuration

To configure the driver, please position in the Linux kernel configuration menu to:

Drivers -> Graphics support -> Support for frame buffer devices -> Xylon logiCVC frame buffer support

Choose **Xylon frame buffer driver type** and pick:
- **Xylon logiCVC frame buffer Open Firmware driver**

Choose **Xylon logiCVC pixel clock source** and pick the one that fits to your design.

3.8.1.1 logiCVC-ML core configuration

The following parameters are extracted directly from Xylon's reference Zynq-7000 SoC design and the logiCVC-ML IP core generics settings (see the logiCVC-ML Users's Manual or the *logicvc_v2_1_0.mpd* file).

```
compatible = "xylon,logicvc-3.02.a";
reg = <0x40030000 0x6000>;
interrupts = <0 59 4>;
```

```
interrupt-parent = <&gic>;

xlnx,display-interface = <0>;
xlnx,display-color-space = <1>;
xlnx,ip-license-type = <0>;
xlnx,ip-major-revision = <3>;
xlnx,ip-minor-revision = <0>;
xlnx,ip-patch-level = <0>;
xlnx,num-of-layers = <3>;
xlnx,layer-0-type = <0>;
xlnx,layer-0-alpha-mode = <0>;
xlnx,layer-0-data-width = <16>;
xlnx,layer-0-offset = <0>;
xlnx,layer-1-type = <0>;
xlnx,layer-1-alpha-mode = <0>;
xlnx,layer-1-data-width = <24>;
xlnx,layer-1-offset = <3240>;
xlnx,layer-2-type = <0>;
xlnx,layer-2-alpha-mode = <0>;
xlnx,layer-2-data-width = <24>;
xlnx,layer-2-offset = <6480>;
xlnx,layer-3-type = <0>;
xlnx,layer-3-alpha-mode = <0>;
xlnx,layer-3-data-width = <24>;
xlnx,layer-3-offset = <9720>;
xlnx,layer-4-type = <0>;
xlnx,layer-4-alpha-mode = <0>;
xlnx,layer-4-data-width = <24>;
xlnx,layer-4-offset = <12960>;
xlnx,buffer-0-offset = <1080>;
xlnx,buffer-1-offset = <1080>;
xlnx,buffer-2-offset = <1080>;
xlnx,buffer-3-offset = <1080>;
xlnx,buffer-4-offset = <1080>;
xlnx,little-endian = <1>;
xlnx,readable-regs = <1>;
xlnx,row-stride = <2048>;
xlnx,use-background = <1>;
xlnx,use-size-position = <1>;
xlnx,vmem-baseaddr = <0x30000000>;
xlnx,vmem-highaddr = <0x3FFFFFFF>;

pixel-clock-source = <3>;
pixel-data-invert = <0>;
pixel-clock-active-high = <0>;
pixel-component-format = "ARGB";
pixel-component-layer = <0>,<1>,<2>;
active-layer = <0>;
videomode = "1920x1080";
edid {
    preffered-videomode = <0>;
    display-data = <0>;
};
```

NOTE: logiCVC contains other IP core generics that are not relevant to the software driver (e.g. mem-burst). These parameters can be written in the .dts but they will be ignored.

Specific parameters description:

- ***pixel-clock-source*** – source of the pixel clock for the logiCVC-ML; 0 - External; 1 – Zynq PS; 2 - logiCLK; 3 - Si570
- ***pixel-data-invert*** – logiCVC-ML sends inverted pixel data towards the display
- ***pixel-clock-active-high*** – when set to "1" logiCVC-ML sends valid pixel data on rising pixel clock edge
- ***pixel-component-format*** – ARGB or RGBA (RGBA used for Android OS only!)
- ***pixel-component-layer*** – index(es) of logiCVC-ML layer that will have the pixel format changed to RGBA format; **This parameter is valid only if *pixel-component-format = RGBA***
- ***active-layer*** – index of the logiCVC-ML layer used for the system console,
- ***videomode*** – defines video mode in the (HRES)x(VRES)MRim text format ("MRim" string explained in Chapter 3.11); If the text matches the later defined custom video mode, the custom mode is used, otherwise the kernel video mode is used
- ***edid*** – see Chapter 3.9
- ***preffered-videomode*** – when set to "1" the logiCVC-ML gets configured accordingly to display device EDID read from the monitor
- ***display-data*** – set to "1" for displaying EDID in system log

3.8.1.2 xylon-video-params

The following *.dts* node defines specific video mode timing parameters which are not available in kernel video mode timing database.
Section should be placed in *.dts* file only when specific timing parameters are needed, like in case of custom LCDs.

If *xylon-video-params* node contains video mode resolution name which is placed under "*videomode*" parameter, framebuffer driver will take that timings and it will not check for kernel timings database.

```
xylon-video-params {
    1680x1050 {
        name = "1680x1050";
        refresh = <60>;
        xres = <1680>;
        yres = <1050>;
        pixclock-khz = <119000>;
        left-margin = <80>;
        right-margin = <48>;
        upper-margin = <21>;
        lower-margin = <3>;
        hsync-len = <32>;
        vsync-len = <6>;
        sync = <0>;
        vmode = <0>;
    };
};
```

Example: *videomode = "1680x1050 ";*

3.8.1.3 logiCLK configuration

The following parameters are used if the "**Xylon logiCVC pixel clock source**"->"**Xylon logiCLK pixel clock**" is set in the Linux kernel configuration menu. The parameters values are determined by the SoC/FPGA hardware design.

```
clkgen@40010000 {
    compatible = "xylon,clkgen-1.01.a";
    reg = <0x40010000 0xffff>;
    osc-clk-freq-hz = <100000000>;
};
```

NOTE: The logiCLK is Xylon's auxiliary IP core used in the reference design for the ZedBoard from Avnet Electronics Marketing: http://www.logicbricks.com/logicBRICKS/Reference-logicBRICKS-Design/Graphics-for-Zynq-AP-SoC-ZedBoard.aspx

3.8.1.4 Video mode selection priority

1. If the custom video mode is defined in the *.dts xylon-video-params* section, and the *xylon-video-params.videomode* is set to that custom mode, the custom mode is used.

2. If *xylon-video-params.videomode* is not set to a custom video mode, the kernel Framebuffer video mode database is searched for a requested video mode. If this mode is not available, the driver will use the default built-in video mode

3. If there is no *.dts* video mode definition, the video mode can be set from the kernel command line, see 3.7.1

4. If there are both, the *.dts* and kernel line video mode definitions, the *.dts* mode is used

3.8.2 Platform Framebuffer driver configuration

To configure the driver, position in the Linux kernel configuration menu to:

Drivers -> Graphics support -> Support for frame buffer devices -> Xylon logiCVC frame buffer support

Choose **Xylon frame buffer driver type** and pick:
- **Xylon logiCVC frame buffer platform driver**

Choose **Xylon logiCVC pixel clock source** and pick one of the options depending on your pixel clock source.

3.8.2.1 logiCVC-ML IP core configuration

A list of hardcoded platform driver parameters describing the logiCVC-ML IP core's configuration used by the Xylon Framebuffer platform driver must be manually set:

- **struct xylonfb_platform_layer_params logicvc_0_layer_params**

```
/* Framebuffer driver platform layer structure */
struct xylonfb_platform_layer_params
{
    /* Layer memory offset in lines */
    unsigned int offset;
    /* Layer buffer memory offset in lines */
    unsigned short buffer_offset;
    /* Layer type */
    unsigned char type;
    /* Layer bits per pixel */
    unsigned char bpp;
    /* Layer alpha mode */
    unsigned char alpha_mode;
    /* Layer control flags */
    unsigned char ctrl_flags;
};
```

- **struct xylonfb_platform_data logicvc_0_platform_data**

```
/* Framebuffer driver platform data structure */
struct xylonfb_platform_data
{
    struct xylonfb_platform_layer_params *layer_params;
    /* logiCVC video mode */
    char *vmode;
    /* logiCVC Control Register value */
    u32 ctrl_reg;
    /* Physical starting address of the video memory */
    unsigned long vmem_base_addr;
    /* Physical ending address of the video memory */
    unsigned long vmem_high_addr;
    /* Layer row stride in pixels */
    unsigned short row_stride;
    /* Number of logiCVC layers */
    unsigned char num_layers;
    /* logiCVC layer ID for FB console */
    unsigned char active_layer;
    /* Background layer bits per pixel */
    unsigned char bg_layer_bpp;
    /* Background layer alpha mode */
    unsigned char bg_layer_alpha_mode;
    /* Display interface and color space type */
    /* higher 4 bits: display interface, lower 4 bits: display color space */
    unsigned char display_interface_type;
    /* logiCVC specific flags */
    unsigned short flags;
};
```

- **struct resource logicvc_0_resource**

The data structures are located in *drivers/video/xylon/xylonfb/platform/xylonfb-platform.c* file.

3.8.2.2 Video mode configuration

xylonfb_platform_data.vmode defines a video mode string e.g. "*1024x768*" (equivalent to the *xylon-video-params.videomode*).

Xylon Framebuffer driver default video mode resolution is defined by the *struct xylonfb_vmode_data xylonfb_vmode* in the *drivers/video/xylon/xylonfb/core/xylonfb.c* file, which is also a suitable place for custom video mode definitions.

3.8.2.3 Video mode selection priority

1. Kernel boot argument video mode has priority over the internal driver video mode selection.
2. If Kernel boot argument video mode is not available, the internal driver video mode is used.

3.9 Framebuffer driver miscellaneous functionality

The *.dts* file inside the "logicvc" node can implement the "edid" subnode.

If *edid* subnode is present together with the running adv7511 V4L2 (video for Linux) driver in kernel, the framebuffer driver parses *preffered-videomode* and *display-data*.

If *preffered-videomode* is set to "1", framebuffer driver will use adv7511 V4L2 driver interface functionality for handling EDID.

If *display-data* is set to "1", framebuffer driver will log EDID to Linux kernel system log.

If *edid* subnode is not present and the adv7511 V4L2 is running in kernel, the framebuffer driver will skip adv7511 V4L2 driver interface functionality.

NOTE: In multi display configurations, when Linux kernel has multiple framebuffer driver instances running, only one logiCVC node is allowed to contain "edid" subnode used by the framebuffer driver. This subnode must be within the logiCVC node with parameters for the logiCVC-ML device connected to the ADV7511 device - because one ADV7511 is physically connected to one logiCVC.

3.10 Framebuffer driver multiple instances

XylonFB driver supports multi display configuration by running multiple driver instances for multiple logiCVC-ML display controller IP cores. Functionality is implemented in the Open Firmware and platform driver configurations.

Defining multiple driver instances in Open Firmware (OF) driver configuration:

- .dts file contains two or more logiCVC nodes with logiCVC HW parameters
- register and video memory base addresses must be different

3.11 Framebuffer driver resolution settings

.dts file can contain "*xylon-video-params*" node where user can place any custom video timing parameters in standard format supported by kernel frame buffer subsystem and xylonfb OF driver. Parameter "*videomode*" in HxV format is parsed by the OF driver to get information about the requested logiCVC-ML resolution. If there is the same video resolution setup in the "*videomode*" parameter and "*xylon-video-params*" node, video timings will be taken from the "*xylon-video-params*"

node and the logiCVC-ML will be accordingly initialized (the same resolution supported by the kernel VESA database does not care). If a video resolution is not setup in the "*xylon-video-params*" node or the node is not present at all, the framebuffer driver will use video timings from the kernel video mode database (assuming supported video resolution is in that database). In that case the "*videomode*" parameter can contain a few more standard Linux video mode options listed in the Linux kernel modedb.txt

If 'M' is specified in the mode_option argument (after <yres> and before <bpp> and <refresh>, if specified) the timings will be calculated using VESA Coordinated Video Timings instead of looking up the mode from a table.

If 'R' is specified, do a 'reduced blanking' calculation for digital displays.

If 'i' is specified, calculate for an interlaced mode.

If 'm' is specified, add margins to the calculation (1.8% of xres rounded down to 8 pixels and 1.8% of yres).

Sample usage: 1920x1080MR@60 - CVT timings, reduced blanking period
Sample usage: 1920x1080M@60i - CVT timings, interlaced

For setting default logiCVC video mode resolution from kernel command line add following command line parameter for getting e.g. 800x600 32bpp video resolution:

video=xylonfb:800x600M-32@60

3.12 Framebuffer driver custom IOCTLs

- Get layer physical index
- Get / Set layer alpha value
- Enable / Disable / Get / Set layer color transparency
- Get / Set layer size and position
- Get / Set layer video buffer
- Get layer buffer offset
- Get layer buffers number
- Get / Set background color
- Enable / Disable layer external buffer switching
- Read / Write any logiCVC register
- Wait / Read display device EDID

4 BUILDING XYLON FRAMEBUFFER DRIVER FOR XILINX ARM LINUX KERNEL

Xylon Framebuffer users must obtain Xilinx Linux distribution and development tools from the Xilinx GIT server, and use the Linux OS running host computer.

1) Add the following parameters to kernel boot arguments to reserve 256 MB at the end of RAM:

console=tty0 mem=768M

NOTE: this is system specific and depends on logiCVC-ML configuration. The video memory's start must be set to ***0x30000000***
IMPORTANT: Without these parameters, Xylon Framebuffer will not be able to do memory mapping of larger video buffers for logiCVC-ML layers!

The Linux Kernel will report error "***Error xylonfb ioremap REGS 0x…….. FB 0x0***", where FB 0x0 indicates failed memory mapping of video buffers.

Description of the used kernel boot arguments:

console=tty0
-> Linux frame buffer console will work with */dev/tty0*

mem=768M
-> Linux kernel system RAM limited to 768 MB

2) Position on the host computer to the kernel root folder and run the configuration menu:

make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- menuconfig

Follow steps from the chapter 3.8 Framebuffer Driver Configuration.

3) Compile the kernel:

make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-

4) Create uImage:
mkimage -A arm -O linux -C none -T kernel -a 8000 -e 8000 -n Linux-3.10.0-xilinx-trd -d arch/arm/boot/zImage arch/arm/boot/uImage

5) After the compilation, install kernel modules into the RAM filesystem by running the following commands:

sudo mkdir /mnt/ramdisk
gunzip ramdisk.image.gz

sudo mount ramdisk.image /mnt/ramdisk/
sudo rm -r /mnt/ramdisk/lib/modules/
sudo make ARCH=arm CROSS_COMPILE=$CC_PATH/CodeSourcery/
Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin/arm-none-linux-gnueabi-
INSTALL_MOD_PATH=/mnt/ramdisk modules_install
sudo umount /mnt/ramdisk
gzip -9 ramdisk.image
mkimage -A arm -T ramdisk -C gzip -d ramdisk.image.gz uramdisk.image.gz

6) To create the DTB file, please place the *devicetree.dts* file into the kernel root folder

scripts/dtc/dtc -I dts -O dtb -o devicetree.dtb devicetree.dts

7) Copy the following images to root of SD card relative to kernel root folder:

- arch/arm/boot/uImage
- uramdisk8M.image.gz
- devicetree.dtb

5 XYLON FRAMEBUFFER DTS SNIPPET

To add logiCVC-ML display controller support to your Linux running Zynq-7000 SoC or FPGA design with implemented logiCVC-ML IP core, please add this snippet to your .dts file (or copy from provided driver deliverables) and change driver parameters to fit your design.

```
logicvc0: logicvc@40030000 {
    compatible = "xylon,logicvc-3.00.a";
    reg = <0x40030000 0x6000>;
    interrupts = <0 59 4>;
    interrupt-parent = <&gic>;

    xlnx,display-interface = <0>;
    xlnx,display-color-space = <1>;
    xlnx,ip-license-type = <0>;
    xlnx,ip-major-revision = <3>;
    xlnx,ip-minor-revision = <0>;
    xlnx,ip-patch-level = <0>;
    xlnx,num-of-layers = <3>;
    xlnx,layer-0-type = <0>;
    xlnx,layer-0-alpha-mode = <0>;
    xlnx,layer-0-data-width = <16>;
    xlnx,layer-0-offset = <0>;
    xlnx,layer-1-type = <0>;
    xlnx,layer-1-alpha-mode = <0>;
    xlnx,layer-1-data-width = <24>;
    xlnx,layer-1-offset = <3240>;
    xlnx,layer-2-type = <0>;
    xlnx,layer-2-alpha-mode = <0>;
    xlnx,layer-2-data-width = <24>;
    xlnx,layer-2-offset = <6480>;
    xlnx,layer-3-type = <0>;
    xlnx,layer-3-alpha-mode = <0>;
    xlnx,layer-3-data-width = <24>;
    xlnx,layer-3-offset = <9720>;
    xlnx,layer-4-type = <0>;
    xlnx,layer-4-alpha-mode = <0>;
    xlnx,layer-4-data-width = <24>;
    xlnx,layer-4-offset = <12960>;
    xlnx,buffer-0-offset = <1080>;
    xlnx,buffer-1-offset = <1080>;
    xlnx,buffer-2-offset = <1080>;
    xlnx,buffer-3-offset = <1080>;
    xlnx,buffer-4-offset = <1080>;
    xlnx,little-endian = <1>;
    xlnx,readable-regs = <1>;
    xlnx,row-stride = <2048>;
    xlnx,use-background = <1>;
    xlnx,use-size-position = <1>;
    xlnx,vmem-baseaddr = <0x30000000>;
    xlnx,vmem-highaddr = <0x3FFFFFFF>;
```

```
    pixel-clock-source = <3>;
    pixel-data-invert = <0>;
    pixel-clock-active-high = <0>;
    pixel-component-format = "ARGB";
    pixel-component-layer = <0>,<1>,<2>;
    active-layer = <0>;
    videomode = "1920x1080";
    edid {
        preffered-videomode = <0>;
        display-data = <0>;
    };
};

xylon-video-params {
    800x480_TM050RBH01 {
        name = "800x480_TM050RBH01";
        refresh = <60>;
        xres = <800>;
        yres = <480>;
        pixclock-khz = <30000>;
        left-margin = <40>;
        right-margin = <40>;
        upper-margin = <29>;
        lower-margin = <13>;
        hsync-len = <48>;
        vsync-len = <3>;
        sync = <0>;
        vmode = <0>;
    };
    800x480_LQ070Y3LG4A {
        name = "800x480_LQ070Y3LG4A";
        refresh = <60>;
        xres = <800>;
        yres = <480>;
        pixclock-khz = <33330>;
        left-margin = <88>;
        right-margin = <40>;
        upper-margin = <35>;
        lower-margin = <8>;
        hsync-len = <128>;
        vsync-len = <2>;
        sync = <1>;
        vmode = <0>;
    };
    1680x1050 {
        name = "1680x1050";
        refresh = <60>;
        xres = <1680>;
        yres = <1050>;
        pixclock-khz = <119000>;
        left-margin = <80>;
        right-margin = <48>;
        upper-margin = <21>;
```

```
        lower-margin = <3>;
        hsync-len = <32>;
        vsync-len = <6>;
        sync = <0>;
        vmode = <0>;
    };
};
```

6 REVISION HISTORY

Version	Date	Author	Approved by	Note
1.00.a	November 26, 2012	D. Joja		Initial document release
1.02.b	January 11, 2013	D. Joja		Updated to 1.2 driver, minor changes
2.01.a	September 02, 2013	D. Joja		EDID handling functionality, Resolution settings improvements, Multiple driver instance support.