

# User Guide: HDMI output peripheral

Ana Klimovic, Bryce Long, Victor Zhang

February 28, 2013

Version for EDK 14.2

## 1 Goals

Use a custom HDMI output peripheral that displays a frame from off-chip memory.

## 2 Examples

You have been given two example projects that use the HDMI output peripheral. The first example uses a resolution of 640x480 and uses RGB565. The second example uses a resolution of 1280x720 and uses RGB888x. The peripheral supports 3 resolutions (640x480, 800x600, and 1280x720) and two colour formats (RGB565 and RGB888x).

## 3 Setup

1. Start XPS and select **Open Project**. Browse to the **system.xmp** file in **ip\hdmi\_out** and open the project. The BSB package is already selected for the *Atlys* board.
2. Click on **Generate Bitstream** and wait for the design to compile. This will take a few minutes.
3. Click on **Export Design** and select to launch Xilinx Software Development Kit. Use the workspace in **ip\workspace**.

4. Create a new Xilinx C Project and select Empty Application. Also create a new BSP project.
5. Copy the source files from `ip\workspace\hdmi_out_test\src` including `lscrip.ld`, the linker file, into your Xilinx C Project.
6. Compile the project and Program the FPGA using the `.elf` file you just compiled.
7. If you have the UART connected, you should see `passed` being printed. This checks that the off-chip memory is connected properly. On the monitor, you should see an animation of a colourful gradient, slowly being covered with white pixels. This animation cycles continuously.

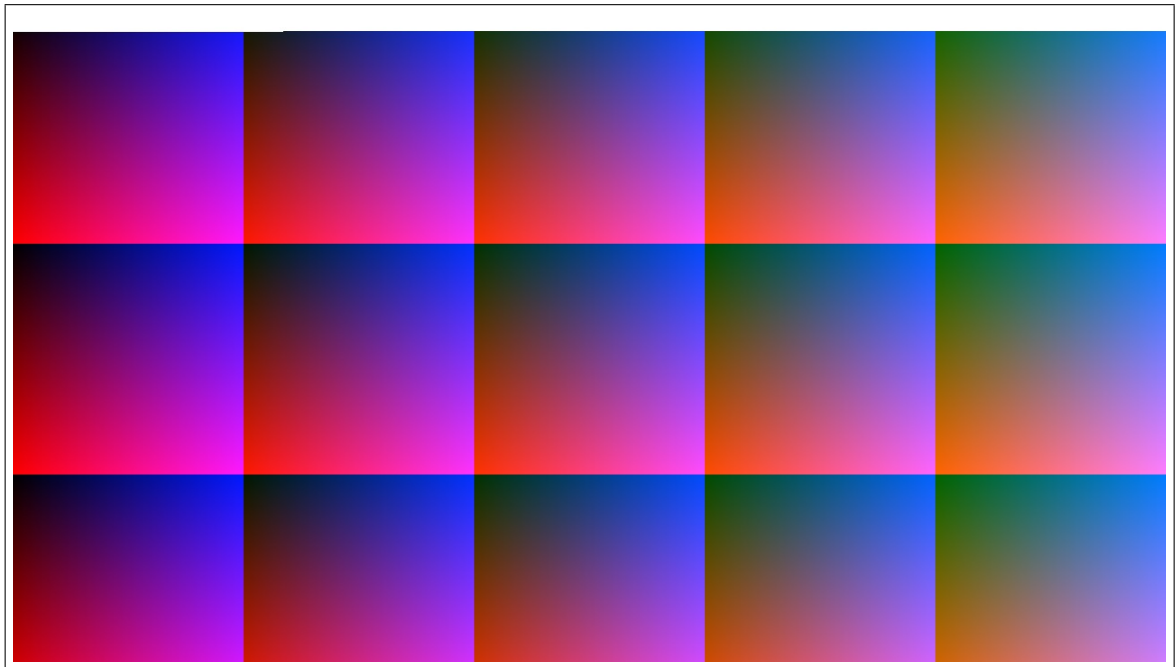


Figure 1: The start of the correct animation (1280x720, RGB888x)

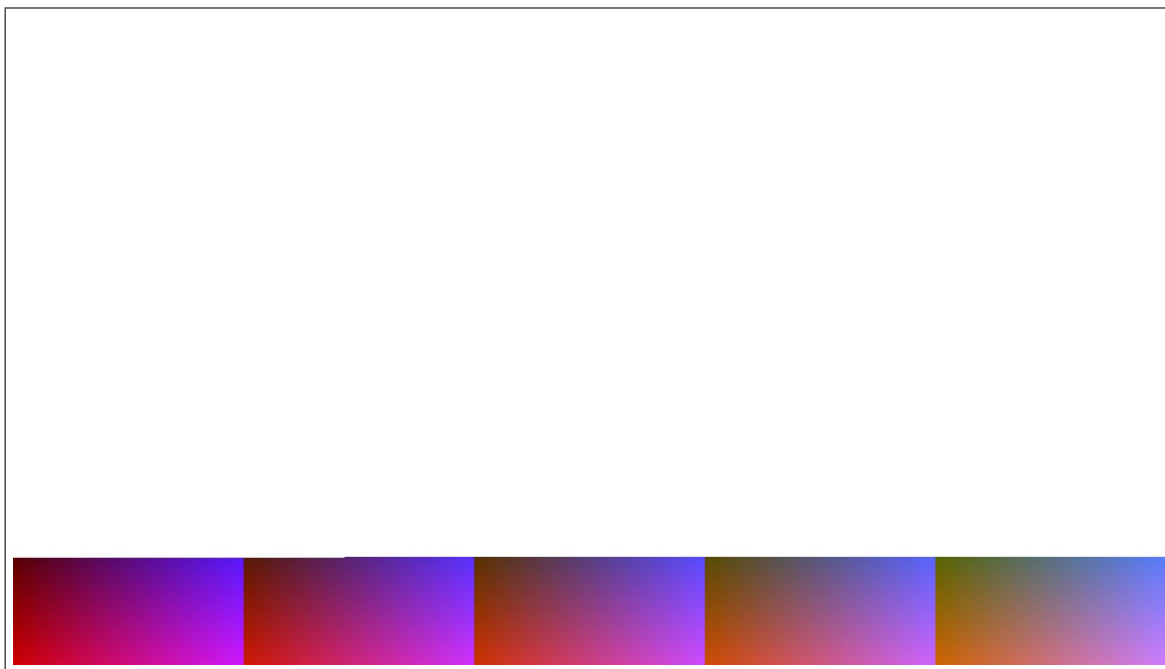


Figure 2: Later on in the animation (1280x720, RGB888x)

## 4 Block Diagram

This sections presents several block diagrams to illustrate how the HDMI output peripheral works. Figure 3 shows the system architecture involving the MicroBlaze, memory, the AXI bus interconnect, the screen and how they interact with the HDMI out IP. Figure 4 zooms in on the HDMI IP itself, showing the main blocks the IP is composed of. Finally, Figure 5 shows the detailed implementation of the user logic that forms the IP.

## 5 Setting parameters

To change resolution, or to change the colour format, open XPS and change the parameters on the hdmi\_out core. **HDMI\_HRES** refers to the horizontal resolution. For example, 1280 refers to a resolution of 1280x720. **HDMI\_NUM\_BYTES\_PER\_PIXEL** refers to the colour format. 2 refers to RGB565 and 4 refers to RGB888x. Note that 640x480 and 800x600 need

## HDMI Out IP System Integration

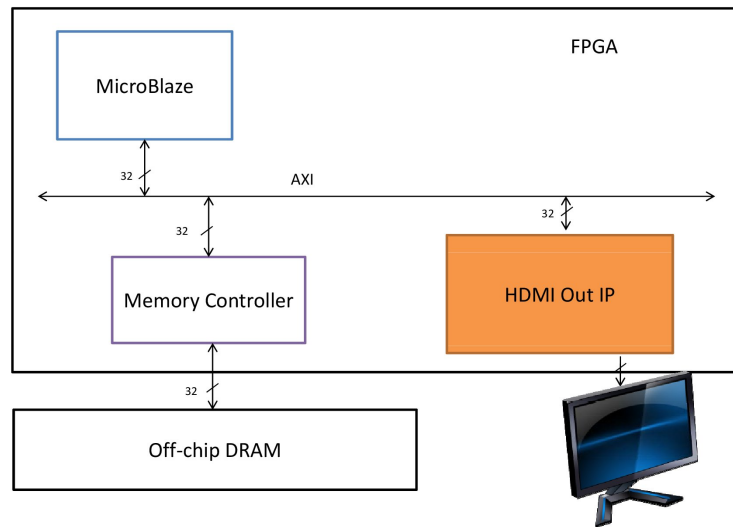


Figure 3: System Integration Block Diagram

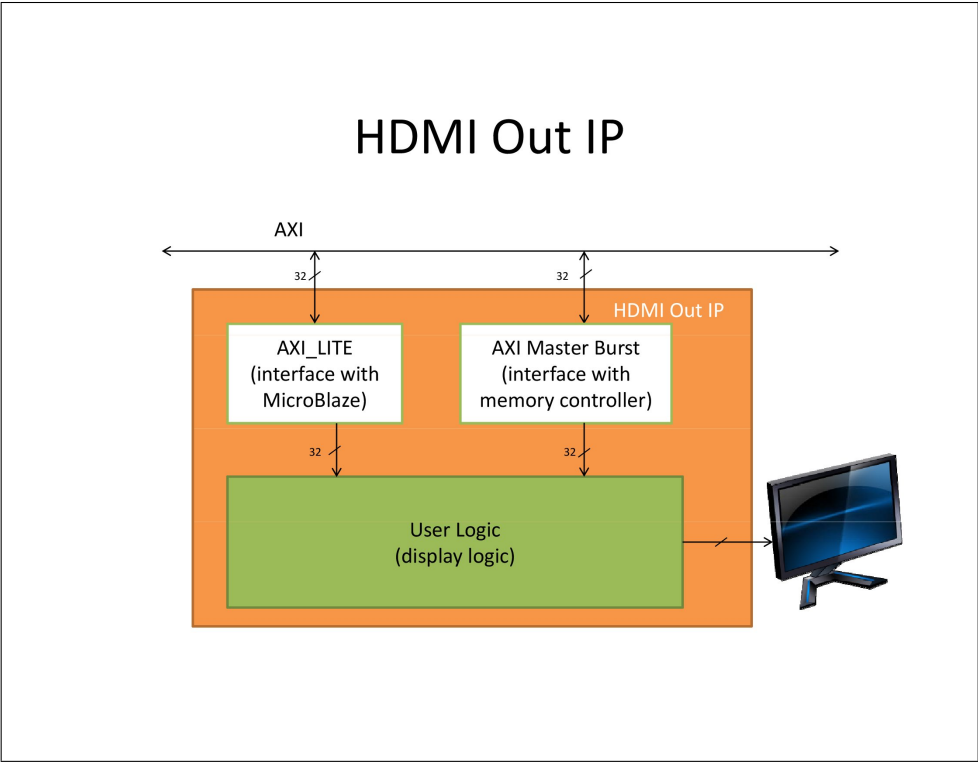


Figure 4: IP Block Diagram

## HDMI Out User Logic

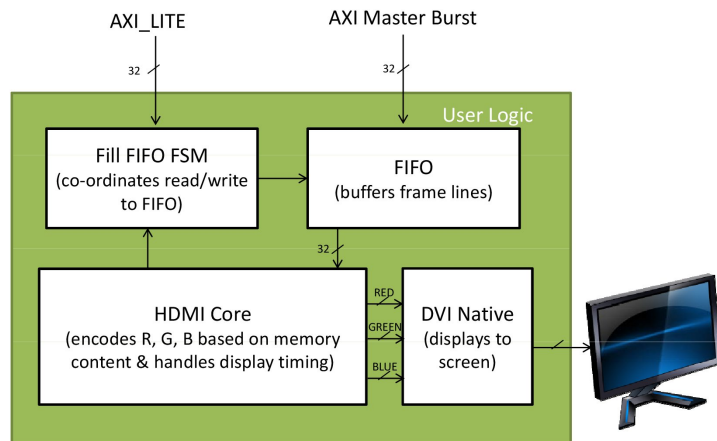


Figure 5: User Logic block diagram

an input clock of 25MHz, while the 1280x720 resolution needs an input clock of 75MHz. The two projects have the correct clock setup. If you change the parameters, you may also need to change the software. Note that line stride is specified in pixels, so line stride may not need to change if only the colour format is different.

## 6 Software registers

**Software register 0** This is the line stride register. The line stride is measured in # of pixels.

**Software register 1** This is the address register. Set it to the beginning of the frame you would like to display.

**Software register 2** This is the enable register. Set it to 1 to enable drawing from the specified frame address.

See the C code example in `ip\workspace\hdmi_out_test\src\main.c` for an example on how to set the software registers.

## 7 Caveats

Note, the RESET button does not work in resetting the design. The video output will not be stable.

## 8 Contact

For more information, feature requests, or bug reports, contact us at [ece532gpu2@googlegroups.com](mailto:ece532gpu2@googlegroups.com). For the latest version of the code, go to <https://github.com/molohov/gpu2> and click ZIP for 1280x720, RGB888x or go to [https://github.com/molohov/gpu2/tree/bryce\\_25mhz](https://github.com/molohov/gpu2/tree/bryce_25mhz) and click ZIP for 640x480, RGB565.