

CAST



BRC

High Performance
Block-to-Raster
Converter
Xilinx Core

Digital image display devices, both static and video, need image samples on a line-by-line / pixel-by-pixel basis; a scheme well known as raster scan. On the other hand many image processing - transform algorithms work on a block-by-block basis. Typical image processing examples of this kind include types of the $N \times M$ image processing filter matrices such as smoothing filters, edge detection filters, noise reduction filters etc.

In the image transform context a well known example is the 2D-Discrete cosine transform (2D-DCT), especially the 8×8 block 2D-DCT, found among others in the MPEG video compression and in JPEG image compression. Streaming applications and applications that cannot afford a full frame buffer but still wish to display such a block-by-block based algorithm processed image face the need of on-the-fly conversion from blocks to raster scan pixels. Our BRC Block-to-Raster core is designed to be the perfect standalone and on-the-fly conversion solution for applications that need to display decompressed image data that have been compressed using the JPEG image compression algorithm. Its use can be extended also to other applications that need to display incoming rectangle pixel blocks.

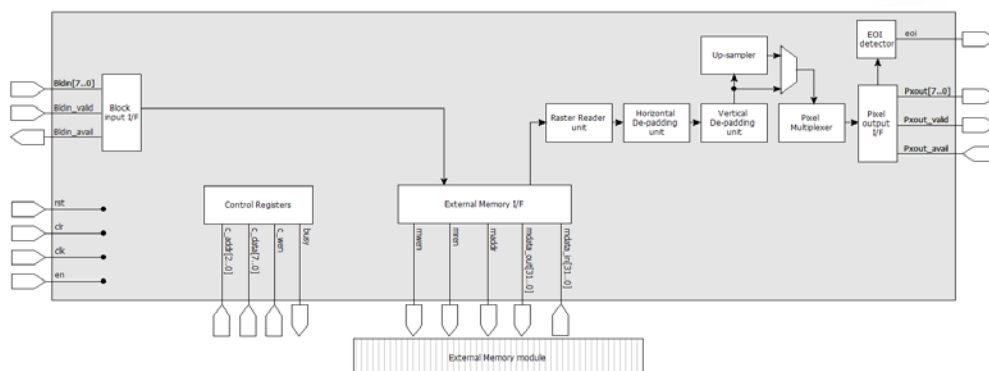
Developed for easy reuse, the BRC is available optimized for several Xilinx devices, with competitive utilization and performance characteristics.

Applications

The core is suitable for implementing a wide range of applications, including:

- JPEG decompression - display applications
- Digital camcorders
- Digital cameras
- Surveillance applications
- Printers

Block Diagram



Features

- JPEG MCU order to raster scan
- Full streaming support
- Supported component sampling factors
 - ▶ 4:4:4
 - ▶ 4:2:2
 - ▶ 4:1:1 (horizontal)
 - ▶ 4:4:4:4 (CMYK)
 - ▶ 1:0:0 (grayscale)
- 8 bit sample precision
- De-padding
- Up-sampling
- Sustained per cycle operation
- Does not insert extra idle cycles and compensates host stalls – perfect for video encoders
- High throughput – over 100 MSamples on FPGA platforms
- Standalone operation
- Fully stallable interfaces
- Low power standby mode via global synchronous register enable
- Hardware proven

Functional Description

The Control Interface is used for programming the operation mode and the conversion-related parameters such as image dimensions, sub-sampling format etc. There is one dedicated interface for each type of data that the core accepts or produces, so there are a total of two data I/O interfaces; the Block input and Pixel output interfaces. In conjunction with these two interfaces and the control registers interface there are two more interfaces. The external memory interface, through which an external memory module that stores conversion intermediate results is accessed, and the system interface.

Core accepts the MCU blocks through the block input interface. The MCU interleaved component blocks are de-multiplexed and they are stored to the external double-buffer memory. Then the external memory is addressed so that image samples are read in raster scan order. According to programmed image parameters regarding image dimensions and sampling format the core, if necessary, performs vertical and/or horizontal de-padding at each block of each image component. The component samples are then interleaved according to the image format specified. Finally the interleaved samples are fed to the pixel output interface. If the core is programmed to do so, it will up-sample its output to 4:4:4 format when applicable (4:2:2 or 4:1:1 input format).

Implementation Results

The following are typical performance and utilization results using Xilinx devices.

Supported Family	Device Tested	Slices	IOBs	GCLK	(f_{max} , MHz)
Virtex-E	V300E-8	1074	124	1	80
Virtex-II	2V500-6	1003	124	1	120
Virtex-II Pro	2VP4-7	1077	124	1	135

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been verified through extensive simulation and rigorous code coverage measurements. It has also been successfully implemented in commercial and prototype systems.

Deliverables

The core includes everything required for successful implementation:

- Post-synthesis EDIF netlist (firm core) optimized for a specific Xilinx device (HDL RTL source code (soft core) is also available)
- Sophisticated self-checking Testbench (Verilog versions use Verilog 2001)
- Simulation script, vectors, and expected results
- Synthesis (soft) or place and route (firm) script
- Comprehensive user documentation