Key Design Features

- Synthesizable, technology independent IP Core for FPGA, ASIC and SoC
- Supplied as human-readable VHDL (or Verilog) source code
- Converts 24-bit RGB digital video to an industry standard 20-bit 4:2:2 SMPTE stream
- Integrated RGB888 to 4:2:2 YCbCr colour-space converter
- All 720p and 1080p (progressive) video formats supported including SMPTE 296M and SMPTE 274M
- All signals synchronous with the pixel clock with full flow-control on the input video pixels
- Small implementation size ideal for all types of FPGA
- Full HD1080p60 video supported on most devices[1]

Applications

- Generation of progressive SMPTE video formats
- Connectivity with a wide range of commercially available video encoder ICs, 3G-SDI, HD-SDI and HDMI transmitters
- Simple and cost-effective method for generating digital video outputs from your FPGA, ASIC or SoC

Generic Parameters

Generic name	Description	Type	Valid range
hd_mode	Output video mode	integer	0: 720p 1: 1080p

Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Pixel clock	rising edge
reset	in	Asynchronous reset	low
underflow	out	Pixel underflow error	high
video_y [9:0]	out	Y channel output	data
video_c [9:0]	out	CbCr channel output	data
video_val	out	SMPTE video output valid	high
pixin [23:0]	in	24-bit RGB888 pixel	data
pixin_vsync	in	Vertical sync in	high
pixin_hsync	in	Horizontal sync in	high
pixin_val	in	Input pixel valid	high
pixin_rdy	out	Ready to accept input pixel (handshake signal)	high

1 Xilinx® 7-series FPGAs used as a benchmark

Block Diagram

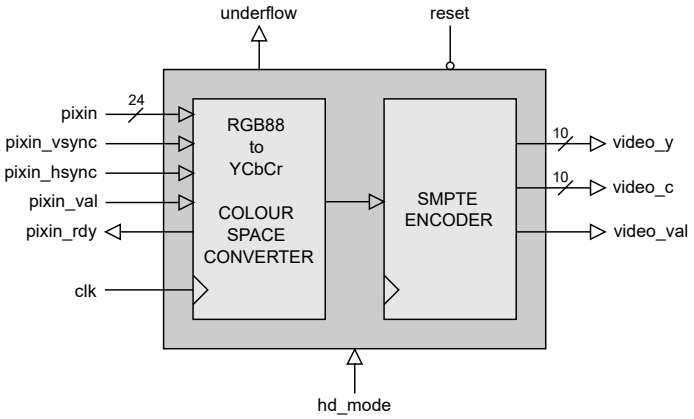

Figure 1: SMPTE Encoder simplified architecture

General Description

The HD_SMPTE_ENCODER IP Core (Figure 1) is a digital video encoder with integrated colour-space converter. The encoder accepts 24-bit RGB pixels from sequential frames. These pixels are then mapped to the YCbCr colour-space and formatted correctly into a SMPTE video output stream.

The encoder begins operation after *reset* is disabled and on detection of the first valid frame. Input pixels are sampled on the rising-edge of *clk* when the *pixin_val* and *pixin_rdy* signals are both asserted high. The *pixin_vsync* flag is coinciendent with the first pixel of a frame. The *pixin_hsync* flag is coincident with the first pixel of a line.

The valid/ready input interface is a streaming interface with full flow-control. It shares a common format with all other Zipcores video IP and allows easy connectivity between IP Cores. The interface also allows simple connectivity with an input FIFO or external frame buffer[2].

The output of the encoder generates an industry standard SMPTE format 20-bit video stream together with a *video_val* signal that is asserted with the first valid byte of the output stream.

If the encoder is starved of pixels during the generation of an active line then the *underflow* flag will be asserted. In the event of an underflow condition, then design must be reset by asserting the *reset* signal low for at least one clock cycle. Operation will then resume as normal when the next input frame is detected.

SMPTE Encoder

The encoder samples the incoming RGB pixels and looks for the first active line in the frame. Once this is detected, the generation of the SMPTE stream begins. Only active input pixels are processed by the encoder. After a system reset, the encoder will revert to it's initial state and stop generating the output stream. Encoding will then resume again with the first active line of the next frame.

2 Please see Zipcores application note: app_note_zc001.pdf for more examples of how to use the valid-ready streaming protocol

When the generic parameter *hd_mode* is set to '0', the encoder expects a 720p video input with 720 active lines and 1280 pixels per line. Conversely, when *hd_mode* is set to '1' then the expected input is 1080p or 1080 lines with 1920 pixels per line. In it's standard configuration, the encoder only supports progressive video refresh rates as defined in the SMPTE 296M and SMPTE 274M specifications.

Note that only active pixels in active lines should be sent to the encoder. The encoder input is not concerned with periods of vertical or horizontal blanking. All video timing information is automatically embedded in the SMPTE output stream.

Colour-space converter

The encoder features an integrated colour-space converter that converts the RGB inputs to YCbCr 4:2:2 format. Chroma values are decimated every second pixel to generate the 4:2:2 video. The colour-space conversion is done according to the following formula:

$$Y = 16 + 0.257\mathrm{R} + 0.504\mathrm{G} + 0.098\mathrm{B}$$
$$Cb = 128 - 0.148\mathrm{R} - 0.291\mathrm{G} + 0.439\mathrm{B}$$
$$Cr = 128 + 0.439\mathrm{R} - 0.368\mathrm{G} - 0.071\mathrm{B}$$

Functional Timing

Example input waveforms are shown in Figure 2. Input pixels and syncs are sampled on a rising clock-edge when *pixin_val* and *pixin_rdy* are both high. When *pixin_val* is low then the inputs are ignored by the encoder. When *pixin_rdy* is low then inputs are stalled.

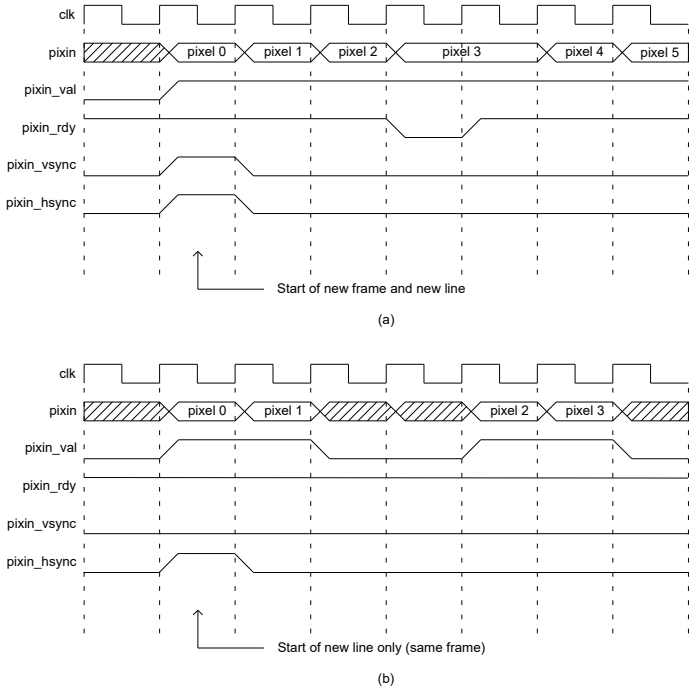

Figure 2: Input waveforms showing start of a new frame (a) and start of a new line (b). Waveform (a) also shows an example stall and (b) shows invalid pixels in the middle of a line

Figure 3 shows an example SMPTE output video stream from the encoder. The 10-bit Y/C values are transferred on a rising clock-edge in parallel when *video_val* is active high.

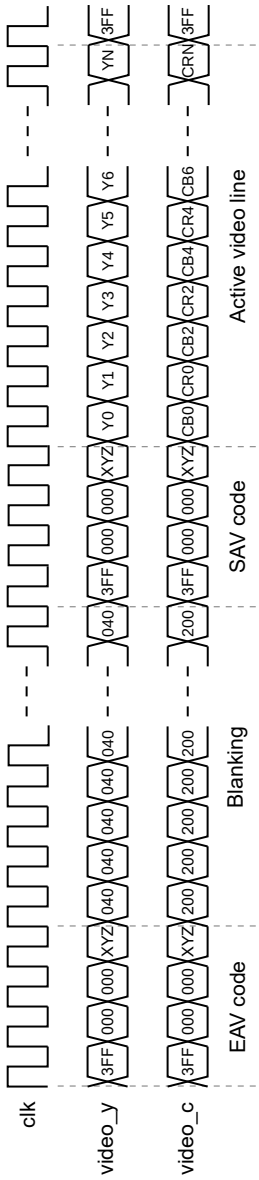

Figure 3: Typical SMPTE 296M/274M output stream (video_val signal not shown)

Source File Description

All source files are provided as text files coded in VHDL. The following table gives a brief explanation of each file.

Source file	Description
pipeline_reg.vhd	Pipeline register element
fifo_sync.vhd	Synchronous FIFO
video_file_reader.vhd	Input video text file reader
hd_smpte_enc_sof.vhd	Field sync component
hd_smpte_enc_csc.vhd	Colour-space converter
hd_smpte_enc_422.vhd	Chroma re-sampler
hd_smpte_enc_fmt.vhd	Main SMPTE formatter
hd_smpte_encoder.vhd	Top-level component
hd_smpte_encoder_bench.vhd	Top-level testbench

Functional Testing

An example VHDL testbench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. pipeline_reg.vhd
2. fifo_sync.vhd
3. hd_smpte_enc_sof.vhd
4. hd_smpte_enc_csc.vhd
5. hd_smpte_enc_422.vhd
6. hd_smpte_enc_fmt.vhd
7. hd_smpte_encoder.vhd
8. hd_smpte_encoder_bench.vhd
9. video_file_reader.vhd

The VHDL testbench instantiates the HD_SMPTE_ENCODER IP Core component with the video format set to either 720p or 1080p. The source video for the simulation is generated by the text file reader component. This component reads a text-based file which contains the RGB pixels and sync flags on consecutive lines. The text file is called *video_in.txt* and should be placed in the top-level simulation directory.

The simulation must be run for at least 50 ms during which time the output SMPTE stream is captured to a text file called *smpte_out.txt*.

Figure 4 shows the results after encoding the a 720p video source into an SMPTE stream. Note that the image also shows the video blanking regions and the vertical blue lines where the EAV and SAV codes are positioned in the video stream.

Figure 4: Output image from the simulation showing the active video frame and blanking regions of the SMPTE stream

Synthesis and Implementation

The files required for synthesis and the design hierarchy is shown below:

- hd_smpte_encoder.vhd
  - hd_smpte_enc_sof.vhd
  - hd_smpte_enc_csc.vhd
    - pipeline_reg.vhd
  - hd_smpte_enc_422.vhd
    - pipeline_reg.vhd
  - fifo_sync.vhd
    - pipeline_reg.vhd
  - hd_smpte_enc_fmt.vhd

The IP Core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® 7-series FPGAs. Synthesis results for other FPGAs and technologies can be provided on request.

If the *video_y* and *video_c* output video signals are designed to be external signals, then we recommend placing these signals in the I/O or pads of the device. They should also be placed locally together to limit the timing skew between individual bits. Other than this, there are no special constraints required for synthesis. The IP Core is completely technology independent.

Trial synthesis results are shown with the generic *hd_mode* parameter set to '0' for 720p video. The resource usage is specified after place and route.

XILINX® 7-SERIES FPGAS

Resource type	Artix-7	Kintex-7	Virtex-7
Slice Register	232	232	222
Slice LUTs	270	268	267
Block RAM	2	2	2
DSP48	9	9	9
Occupied Slices	116	113	119
Clock freq. (approx)	250 MHz	350 MHz	450 MHz

Revision History

Revision	*Change description*	*Date*
1.0	Initial revision	22/02/2018
1.1	First official release	30/10/2019