Key Design Features

- Technology independent soft IP Core for FPGA, SoC and ASIC
- Supplied as human readable VHDL (or Verilog) source code
- Separate Camera Link® Receiver (frame-grabber) and Camera Link® Transmitter (camera) IP Cores
- Supports BASE, MEDIUM and FULL configurations of the Camera Link specification
- Fully configurable clocking via PLL or the clock management resources available on the target device
- Error checking for data misalignment at the Camera Link receiver
- Supports a parallel clock of ~70 MHz or ~500 Mbps / lane on basic FPGAs and SoCs. Higher data rates supported on higher-end devices
- No special interface circuitry required. Connects directly to the standard LVDS I/O resources available on common FPGA and SoC devices

Example Applications

- Real-time / low-latency video interfaces on low-cost devices
- Data streaming interfaces over cable or twisted pair
- General purpose LVDS / SERDES applications

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
config	Camera Link Configuration Setting	integer	0: BASE 1: MEDIUM 2: FULL

Pin-out Description (Receiver / Frame-grabber)

Pin name	*I/O*	*Description*	*Active state*
rst_n	in	Asynchronous reset	low
xc_p/n yc_p/n zc_p/n	in	Camera Link input clocks (BASE / MEDIUM / FULL) 2up/3down/2up duty cycle	LVDS
x0_p/n x1_p/n x2_p/n x3_p/n	in	Camera Link input data Lanes 0 to 3 (BASE configuration)	LVDS
y0_p/n y1_p/n y2_p/n y3_p/n	in	Camera Link input data Lanes 0 to 3 (MEDIUM configuration)	LVDS
z0_p/n z1_p/n z2_p/n z3_p/n	in	Camera Link input data Lanes 0 to 3 (FULL configuration)	LVDS
sys_rst_f	out	Asynchronous reset	low
sys_clk_f	out	System / pixel clock	rising edge
ser_clk_f	out	Serial clock (= sys_clk x 7)	rising edge
pix_a [7:0] pix_b [7:0] pix_c [7:0]	out	Parallel pixel data out (BASE configuration)	data
pix_d [7:0] pix_e [7:0] pix_f [7:0]	out	Parallel pixel data out (MEDIUM configuration)	data
pix_g [7:0] pix_h [7:0] pix_i [7:0]	out	Parallel pixel data out (FULL configuration)	data
pix_dval	out	Pixel valid flag (data en)	high
pix_fval	out	Frame valid flag (vsync)	high
pix_lval	out	Line valid flag (hsync)	high
err_dval	out	Data alignment error flag	high
err_fval	out	Frame alignment error flag	high
err_lval	out	Line alignment error flag	high

Pin-out Description (Transmitter / Camera)

Pin name	*I/O*	*Description*	*Active state*
sys_rst	in	Asynchronous reset	low
sys_clk	in	System / pixel clock	rising edge
ser_clk	in	Serial clock (= sys_clk x 7)	rising edge
pix_a [7:0] pix_b [7:0] pix_c [7:0]	in	Parallel pixel data in (BASE configuration)	data
pix_d [7:0] pix_e [7:0] pix_f [7:0]	in	Parallel pixel data in (MEDIUM configuration)	data
pix_g [7:0] pix_h [7:0] pix_i [7:0]	in	Parallel pixel data in (FULL configuration)	data
pix_dval	in	Pixel valid flag (data en)	high
pix_fval	in	Frame valid flag (vsync)	high
pix_lval	in	Line valid flag (hsync)	high
xc_p/n yc_p/n zc_p/n	out	Camera Link output clocks (BASE / MEDIUM / FULL) 2up/3down/2up duty cycle	LVDS
x0_p/n x1_p/n x2_p/n x3_p/n	out	Camera Link output data Lanes 0 to 3 (BASE configuration)	LVDS
y0_p/n y1_p/n y2_p/n y3_p/n	out	Camera Link output data Lanes 0 to 3 (MEDIUM configuration)	LVDS
z0_p/n z1_p/n z2_p/n z3_p/n	out	Camera Link output data Lanes 0 to 3 (FULL configuration)	LVDS

Copyright © 2022 www.zipcores.com

Block Diagram

CAMERA LINK RX

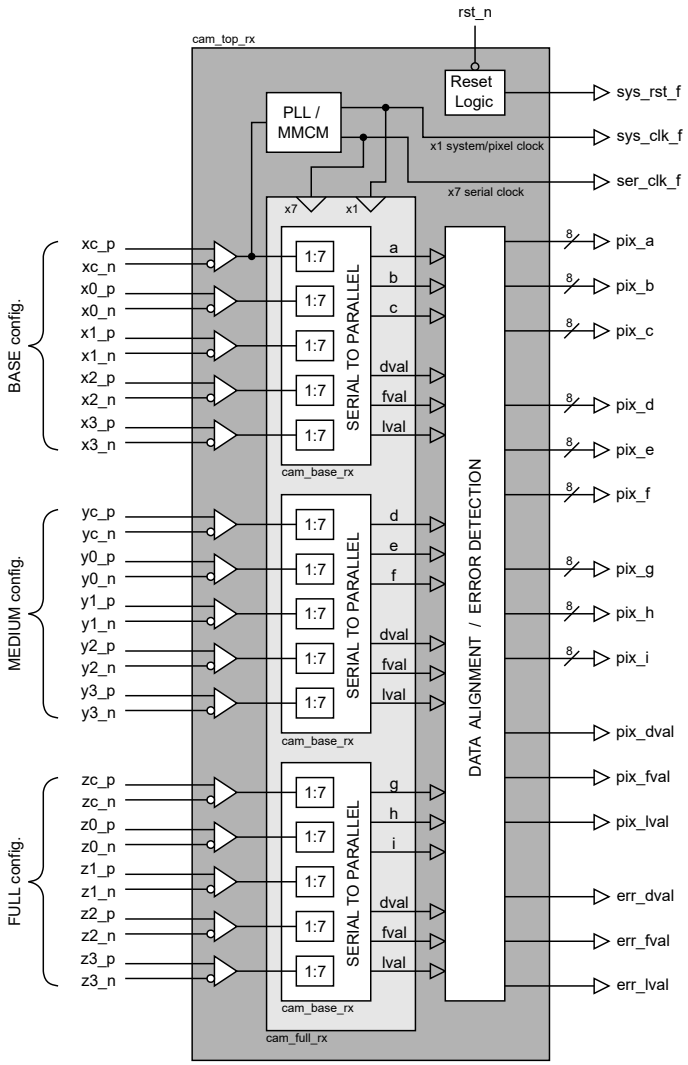

Figure 1: Camera Link Receiver basic architecture

CAMERA LINK TX

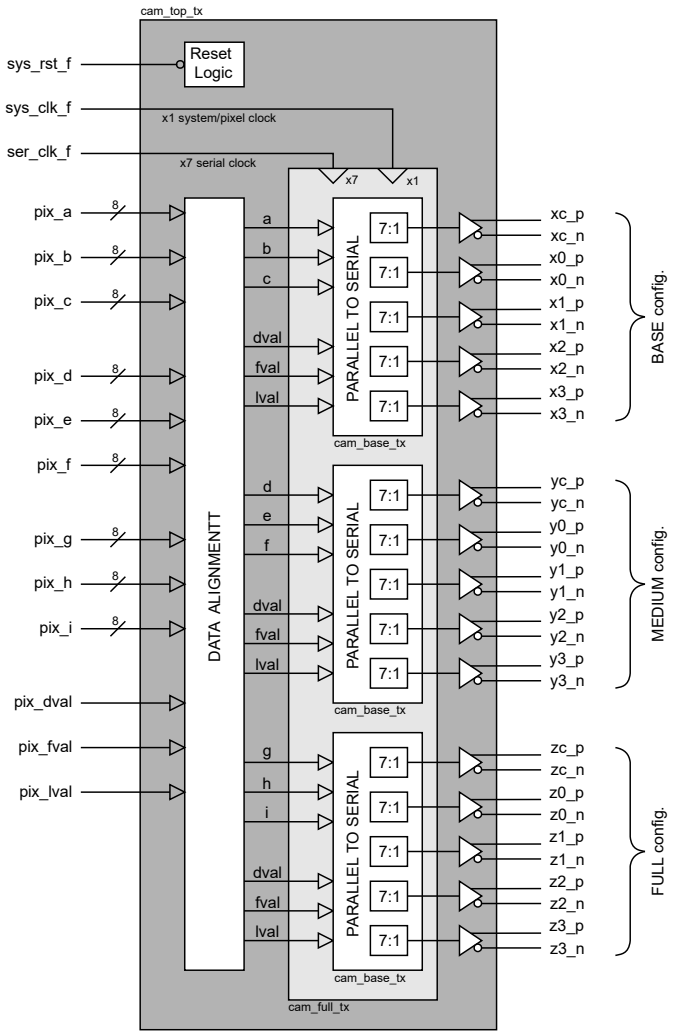

Figure 2: Camera Link Transmitter basic architecture

General Description

The Camera Link® IP Core is a high-speed LVDS transmitter / receiver pair that conforms to the standard Camera Link protocol originally developed by National Semiconductor Corp®.

The design is comprised of an independent transmitter and receiver that may be implemented separately or together as a single transceiver unit. The IP Core may be used in either the BASE, MEDIUM or FULL configurations as defined in the Camera Link specification.

In general, each data lane can support around 500 Mbps (max) per LVDS lane on basic FPGA and SoC devices. This gives a typical throughput of around 2 Gbps over the 4 data lanes (BASE), 4 Gbps over 8 lanes (MEDIUM) or 6 Gbps over 12 lanes (FULL). However, the maximum data rate attained will be dependent on a wide range of factors such as: cable type, cable length, PCB board layout and also the choice of target device.

Figures (1) and (2) describe the basic architecture showing the sub-modules of the design. The following sections explain the Camera Link receiver and transmitter in more detail.

Camera Link Receiver

The receiver or 'frame-grabber' is responsible for de-serialization of the serial input data from the 4 x LVDS data lanes. In order to do the data lane sampling the LVDS input clock is taken from the *xc_p/n* input. This clock is then fed into an internal PLL (or equivalent) to provide the x7 serial clock in order to sample the LVDS data inputs on lanes *x0_p/n* to *x3_p/n* (BASE confg). Likewise, the same serial clock is used to sample inputs on the MEDIUM and FULL data lanes (if implemented).

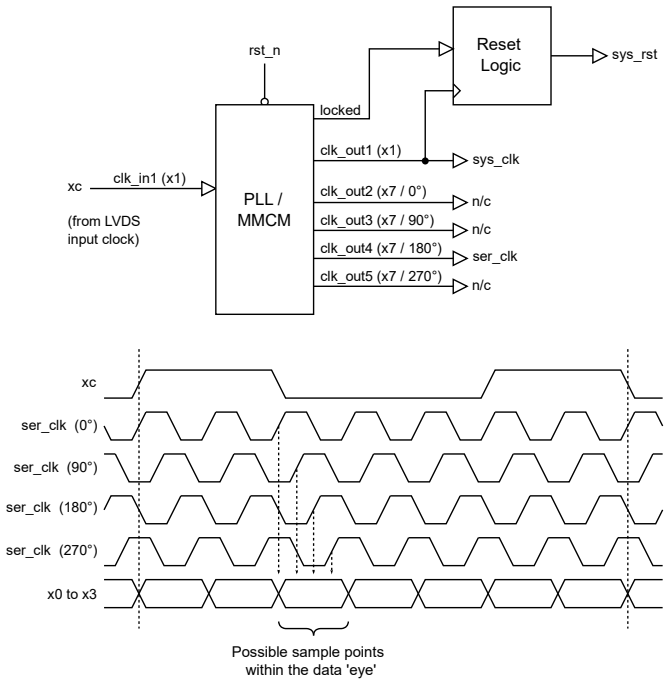

Figure 3: PLL circuit with options for different phase-shifted serial clocks

The data sampling of the LVDS inputs is very sensitive and, as such, care should be taken that the sampling is done close to the middle of the data 'eye'. For instance, all current FPGA and SoC devices allow the clock signal to be phase-shifted by various amounts to permit the best possible sampling point. The source code is provided with an example PLL implementation with 0, 90, 180 and 270 degree phase shifts of the serial clock. Note that some experimentation may be required to find the best phase-shifted clock to use. Figure (3) shows the PLL circuit and the relationship between the data and phase-shifted serial clocks. By default, the inverted serial clock (180 degrees phase-shift) is used in the implementation. The user may choose a different clock by editing the source file *cam_rx_top.vhd*.

Note that the signals *sys_clk*, *ser_clk* and *sys_rst* are also forwarded out of the receiver IP Core and are renamed *sys_clk_f*, *ser_clk_f* and *sys_rst_f* at the ports. These signals may be used by the downstream logic for further video processing. All parallel output data from the receiver is synchronous with the *sys_clk* signal.

Data Alignment Errors

In the case where only the BASE Camera Link configuration is used, then the correct sampling of the LVDS data lanes may be verified by monitoring the *pix_dval*, *pix_fval* and *pix_lval* signals at the output of the receiver. Assuming that there is a valid Camera Link video input signal, then the *dval*, *lval* and *fval* should be clean with the correct horizontal line frequency and frame rate.

For implementations that use the MEDIUM and FULL configurations, then there are three error flags that may be monitored to check for correct data alignment between streams. These are: *err_dval*, *err_fval* and *err_lval*. These error flags will be asserted if there is a skew between the data. In this case, then it indicates that the phase-shift of the serial clock should be adjusted as described in the previous paragraph.

Camera Link Transmitter

The transmitter performs the reciprocal operation to the receiver. In order for correct operation, then the user must supply two separate clocks. These are *sys_clk* which is the system clock that is synchronous with the parallel input data and the he signal *ser_clk* which is the serial clock. The system clock and serial clock do not need to be phase-aligned, but the serial clock must be an exact integer multiple of the system clock with the relationship:

$$\mathrm{ser\_clk} = \mathrm{sys\_clk} * 7$$

For the purposes of clock generation then the user may use the same 'cam_pll.vhd' component as the receiver. However, it is recommended that the user re-generate the PLL component for optimum performance[1].

After system reset, transmission of data begins on a rising clock-edge of *sys_clk*. The parallel data words are read on consecutive system clock cycles and the data is serialized in accordance with the Camera Link specification. All parallel input data (pixels and syncs) are synchronous with the *sys_clk* signal.

Note on Camera Link Configurations

If the user wishes to use the IP Core (Rx or Tx) in BASE configuration only, then the MEDIUM and FULL input signals should be tied to logic '0' and MEDIUM and FULL outputs left unconnected in the design. Likewise, if the user wishes to use the IP Core in MEDIUM configuration only, then the FULL input signals should be tied to logic '0' and the outputs left unconnected.

1 Most EDA tools (e.g. Xilinx/Vivado) feature applications that allow easy generation of PLL or MMCM components with the desired parameters.

Functional Timing

Figure (4) shows the standard Camera Link timing specifications for the BASE, MEDIUM and FULL configurations. Both the receiver and the transmitter IP Cores follow this exact specification. Note that the design assumes that the input clocks *xc_p/n*, *yc_p/n* and *zc_p/n* are derived from the same clock source and are phase-aligned at the camera outputs.

Note that the serial data rate is exactly 7x the parallel data rate. The parallel clock has a 4:3 duty cycle and is aligned with the serial data in a 2up / 3down / 2up configuration as shown.

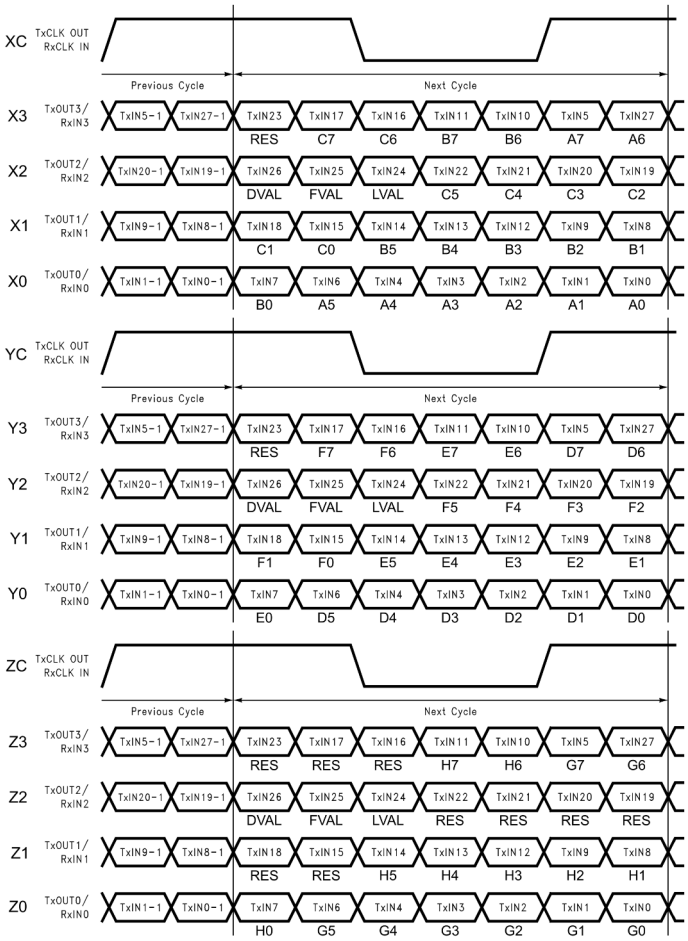

Figure 4: Camera Link input and output timing diagrams: BASE, MEDIUM and FULL configurations

Source File Description

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

Source file	***Description***
cam_ibuf.vhd	LVDS input buffer
cam_obuf.vhd	LVDS output buffer
cam_pll.vhd	PLL for clock generation
cam_deserializer.vhd	7:1 bit de-serializer
cam_base_rx.vhd	Camera Link base config Rx
cam_full_rx.vhd	Camera Link full config Rx
cam_top_rx.vhd	Camera Link top-level receiver
cam_serializer.vhd	1:7 bit serializer
cam_base_tx.vhd	Camera Link base config Tx
cam_full_tx.vhd	Camera Link full config Tx
cam_top_tx.vhd	Camera Link top-level transmitter
cam_model.vhd	Camera Link model for testing
cam_pulse_ext.vhd	Pulse extender
cam_reset_retime.vhd	Reset retiming circuit
cam_link_if.vhd	Top-level Rx/Tx pair for testing only
cam_link_if_bench.vhd	Top-level Rx/Tx pair testbench

Functional Testing

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

1. cam_ibuf.vhd
2. cam_obuf.vhd
3. cam_pll.vhd
4. cam_deserializer.vhd
5. cam_base_rx.vhd
6. cam_full_rx.vhd
7. cam_top_rx.vhd
8. cam_serializer.vhd
9. cam_base_tx.vhd
10. cam_full_tx.vhd
11. cam_top_tx.vhd
12. cam_model.vhd
13. cam_pulse_ext.vhd
14. cam_reset_retime.vhd
15. cam_link_if.vhd
16. cam_link_if_bench.vhd

The testbench instantiates the Camera Link receiver and transmitter in a back-to-back configuration in a single top-level component called ‘cam_link_if.vhd’. The testbench is called ‘cam_link_if_bench.vhd’.

The top level testbench includes a Camera Link model called 'cam_model.vhd' that drives the Camera Link receiver with randomized signals on the LVDS inputs. During the course of the simulation, the testbench captures the output LVDS signals from the transmitter and compares these to the original inputs. If there is a mismatch then the corresponding error flag 'xx_err' is asserted high.

The simulation runs for around 1 ms.

Synthesis and Implementation

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

Transmitter top-level component:

- cam_top_tx.vhd
  - cam_full_tx.vhd
    - cam_base_tx.vhd
      - cam_serializer.vhd
  - cam_obuf.vhd

Receiver top-level component:

- cam_top_rx.vhd
  - cam_full_rx.vhd
    - cam_base_rx.vhd
      - cam_deserializer.vhd
  - cam_ibuf.vhd
  - cam_pll.vhd
  - cam_reset_retime.vhd
  - cam_pulse_ext.vhd

The Camera Link IP Core is technology independent with the exception of the differential LVDS I/O buffers and PLL component which must be specific to the FPGA, SoC or ASIC process being used. As a benchmark, synthesis results have been provided for the AMD / Xilinx® 7-series FPGAs. Synthesis results for other FPGAs and technologies can be provided on request.

Another recommendation is to ensure that the I/O registers are placed in the pads of the target device. This may be specified as an additional attribute in the RTL source code or specified in the constraints file - for instance the 'xdc' or 'sdc' file in the synthesis tool[2]. Placing the inputs in the I/O of the device will ensure more reliable data capture and timing results.

Finally, it's also recommended that the user place the Camera Link input and output pins in a localized area – that is, not spread out around the die. This will reduce timing skew between the input and output data.

Trial synthesis results are shown with the generic parameter, *config* = 2. This corresponds to the FULL Camera Link implementation. Resource usage is specified after place and route of the design and is listed for the top-level component 'cam_link_if.vhd' which contains both the Receiver and Transmitter IP Cores.

XILINX® 7-SERIES FPGAS

Resource type	***A-7***	***K-7***	***V-7***	***V-US+***
Slice Register	553	553	553	535
Slice LUTs	186	186	186	190
Block RAM	0	0	0	0
DSP48	0	0	0	0
Occupied Slices	170	159	160	79 (CLB)
Sys clk freq (est.)	70 MHz	80 MHz	90 MHz	100 MHz

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision.	29/10/2021
1.1	Migration of source-code to standard IP Core library. General code clean-up.	07/03/2022
1.2	Added functionality to optimize the circuit when operating in BASE, MEDIUM and FULL configurations.	20/12/2022

2 Example constraints files may be provided on request according to the chosen synthesis tool.