

Key Design Features

- Synthesizable, technology independent VHDL Core
- Simple FIFO input interface
- Programmable RGB channel data width - e.g. to support 8-bit or 10-bit DACs
- Highly versatile, supporting direct connection to a wide range of video DACs, Transceivers, and DVI / HDMI CODECs.
- VSYNC, HSYNC, CSYNC, DE and BLANK outputs
- Fully programmable timing parameters
- Supports industry standard and fully custom video modes
- Resolutions up to 4096 x 4096 pixels
- Independent system and pixel clocks supporting frequencies in excess of 350 MHz on most FPGA platforms
- Compatible with all Zipcores video IP

Applications

- High definition Digital or Analogue video
- Computer monitors and flat-panel displays
- Digital TV and multimedia solutions

Generic Parameters

Generic name	Description	Type	Valid Range
dw	Pixel data width	integer	$< 2^{16}$
h_s	Horizontal sync pulse duration	integer	$< 2^{16}$
h_bp	Horizontal back-porch duration	integer	$< 2^{16}$
h_fp	Horizontal front-porch duration	integer	$< 2^{16}$
h_disp	Active pixels per line	integer	$< 2^{16}$
h_line	Duration of whole line	integer	$< 2^{16}$
v_s	Vertical sync pulse duration	integer	$< 2^{16}$
h_bp	Vertical back-porch duration	integer	$< 2^{16}$
v_fp	Vertical front-porch duration	integer	$< 2^{16}$
v_disp	Active lines per frame	integer	$< 2^{16}$
v_frame	Duration of whole frame	integer	$< 2^{16}$

Block Diagram

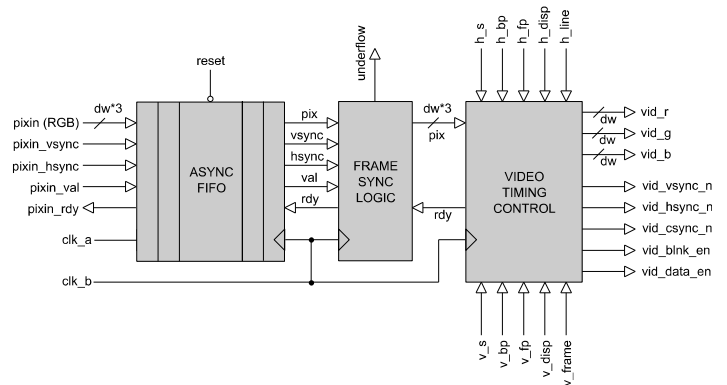


Figure 1: Video timing generator architecture

Pin-out Description

Pin name	I/O	Description	Active state
clk_a	in	System clock	rising edge
clk_b	in	Pixel/DAC clock	rising edge
reset	in	Asynchronous reset	low
underflow	out	Indicates pixel underflow (synchronized to clk_b)	high
pixin [dw*3:0]	in	RGB input 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
vid_r [dw - 1:0]	out	Video out - RED	data
vid_g [dw - 1:0]	out	Video out - GREEN	data
vid_b [dw - 1:0]	out	Video out - BLUE	data
vid_vsync_n	out	Video VSYNC	low
vid_hsync_n	out	Video HSYNC	low
vid_csync_n	out	Video CSYNC	low
vid_blnk_en	out	Video BLANK enable	high
vid_data_en	out	Video DATA enable	high

General Description

VID_TIMING_GEN is a fully configurable video timing generator with the ability to support any video resolution up to 4096 x 4096 pixels in size. The module is compatible with a wide range of video DACs, CODECs and transceivers and provides a flexible solution for displaying digital or analogue video on an external TV, monitor or flat panel display. The module is also capable of clock speeds in excess of 350MHz making it ideal for high-bandwidth HD video solutions.

Input pixels and syncs are read on the rising edge of *clk_a* (the system clock) when *pixin_val* and *pixin_rdy* are both active high. The input signal *pixin_vsync* is coincident with the first active pixel in a frame and the signal *pixin_hsync* is coincident with the first active pixel in a line. (N.B. These sync signals should not be confused with true video timing signals. Their purpose is to delineate the first pixel in a frame and the first pixel in a line only).

After resynchronizing the input pixels to the pixel-clock domain (*clk_b*), the next module is responsible for locking the video signal to the first frame of video. Once frame-lock is achieved, pixels are supplied on demand to the video timing control unit. This module generates the correct RGB video, sync and blanking information depending on the generic parameter settings.

Generic timing parameters

The generic timing parameters determine the duration of the output video, syncs and blanking. They also determine the relative position of the active video signal between syncs. Timing parameters are specified as an integer number of *pixels* (or pixel clocks) for the horizontal timing. For the vertical timing parameters, values are specified as an integer number of *lines*. The timing parameters (Figures 2 and 3) must be specified correctly for the chosen video mode.

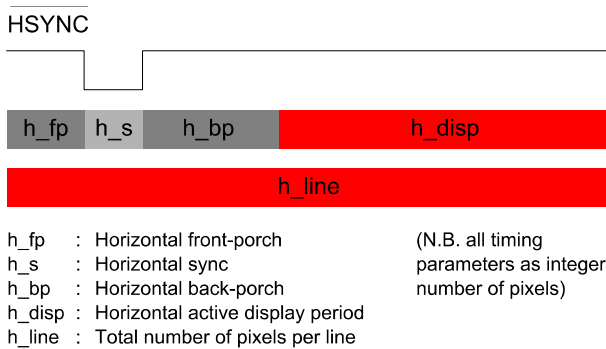


Figure 2: Horizontal timing parameters

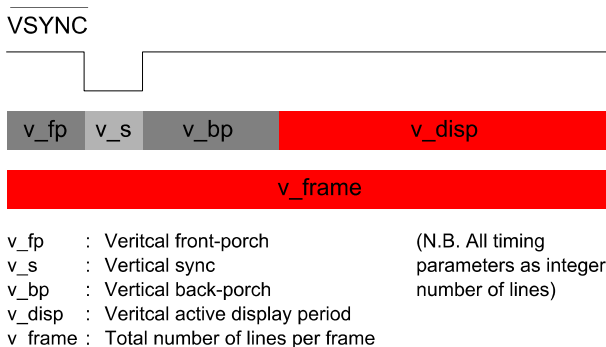


Figure 3: Vertical timing parameters

As an example, consider a standard XGA output (1024 x 768) with a screen refresh rate of 60 Hz and a pixel-clock frequency of 65 MHz. The following tables describe the horizontal and vertical timing settings according to the VESA specification.

HORIZONTAL TIMING (XGA at 60Hz)

Generic parameter	Description	Value
h_fp	Front-porch	24
h_s	Sync pulse	136
h_bp	Back-porch	160
h_disp	Visible area	1024
h_line	Whole line	1344

VERTICAL TIMING (XGA at 60Hz)

Generic parameter	Description	Value
v_fp	Front-porch	3
v_s	Sync pulse	6
v_bp	Back-porch	29
v_disp	Visible area	768
v_frame	Whole frame	806

By modifying the respective horizontal and vertical timing parameters, a diverse range of standard and fully-custom video modes may be displayed.

Standard video timing output signals

The output video timing signals are compatible with standard off-the-shelf video DACs and CODECs. In addition to separate VSYNC and HSYNC signals, the module also provides a Composite SYNC signal that is often used to provide sync information on the GREEN signal of the output video.

In addition, two separate enable signals are provided. These are the video data enable (DE) and blanking enable (BLNK). The data enable is high during active pixels. Conversely, the blanking enable is high when active video is not present. The data enable signal is commonly used when interfacing to DVI and HDMI Transceivers in order to enable valid pixels into the device.

Pixel input/output rate considerations

During normal operation, it is important to ensure that the input pixel rate is sufficient to sustain the chosen video mode. If at any point, the video timing generator detects an internal buffer underflow, then the *underflow* flag is asserted. Once underflow has occurred, then subsequent pixels will be out of sync and the output display will become corrupted.

The output may be reset by asserting the system reset signal for at least one *clk_a* cycle. This will result in a resynchronization of the video signal and the recovery of a clean video output display within a single frame period.

Functional Timing

Input video timing

RGB pixels are sampled at the module input according to the valid-ready pipeline protocol. Figure 4 shows the signalling at the input of the video signal generator at the start of a new frame. The first pixel of a new frame begins with *pixout_vsync* and *pixout_hsync* asserted high together with the first pixel. Note that when *pixin_rdy* is asserted low, then input pixels must be held-off, otherwise pixels will be lost and the output video will become out of sync¹.

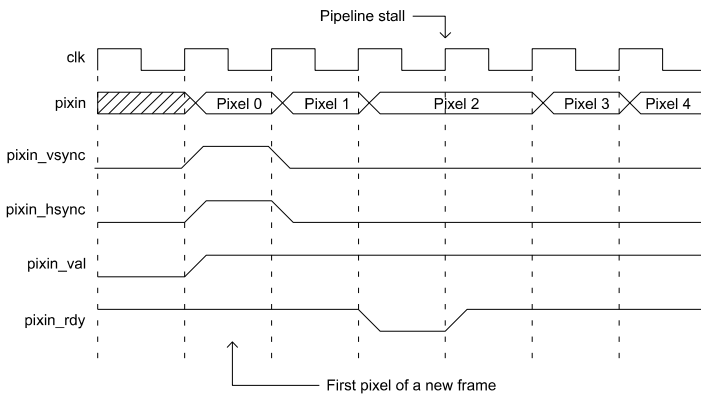


Figure 4: First input pixel in a frame

For the first pixel in a new line (Figure 5) the signal *pixout_hsync* is asserted high with *pixout_vsync* held low. Note that as well as a pipeline stall, Figure 5 also shows an invalid pixel condition. When *pixin_val* is low, the input pixel and input syncs (if present) are ignored.

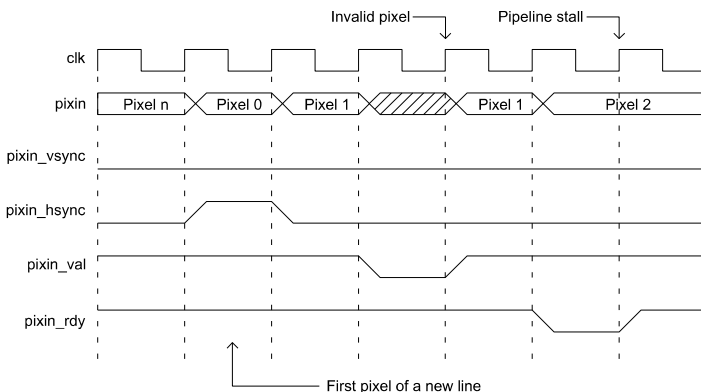


Figure 5: First input pixel of a line

Output video timing

Figures 6 and 7 show the video timing waveforms for a complete frame and a complete line respectively. The position of the active video and the length and duration of the syncs is fully customizable depending on the generic parameter settings.

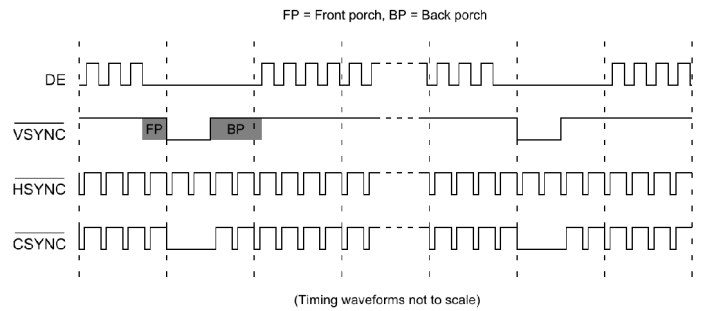


Figure 6: Output video timing (FRAME)

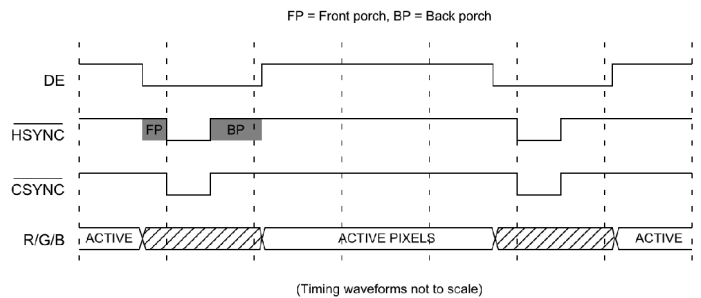


Figure 7: Output video timing (LINE)

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
video_file_reader.vhd	Video source file reader
vid_timing_fifo.vhd	Asynchronous pixel FIFO
vid_timing_sof.vhd	Start of frame sync module
vid_timing_cont.vhd	Video timing controller
vid_timing_gen.vhd	Top-level component
vid_timing_gen_bench.vhd	Top-level test bench

¹ Please refer to Zipcores application note: app_note_zc001.pdf for a more detailed explanation of the valid-ready pipeline protocol.

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. video_file_reader.vhd
2. vid_timing_fifo.vhd
3. vid_timing_sof.vhd
4. vid_timing_cont.vhd
5. vid_timing_gen.vhd
6. vid_timing_bench.vhd

The VHDL testbench instantiates the VID_TIMING_GEN component and the user may modify the generic timing parameters as required. In the example provided, the test is configured to give an industry standard VGA output display at 640x480 resolution and a 60Hz refresh rate.

The simulation must be run for at least 100 ms during which time the output syncs and RGB output pixels are captured in the file *video_out.txt*.

The output text file follows a simple format which defines the state of signals: *vid_vsync_n*, *vid_hsync_n*, *vid_r*, *vid_g* and *vid_b* on a clock-by-clock basis. The simulation output is shown in Figure 8.



Figure 8: VGA simulation output showing regions of video blanking

Synthesis

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

- vid_timing_gen.vhd
 - vid_timing_cont.vhd
 - vid_timing_fifo.vhd
 - vid_timing_sof.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx Virtex 5 and the Altera Stratix III series of FPGA devices. The lowest and highest speed grade devices have been chosen in both cases for comparison.

Trial synthesis results are shown in the following tables. The design was synthesized with the generic parameters set as follows: *dw* = 8, *h_s* = 96, *h_bp* = 48, *h_fp* = 16, *h_disp* = 640, *h_line* = 800, *v_s* = 2, *v_bp* = 33, *v_fp* = 10, *v_disp* = 480, *v_frame* = 525.

Resource usage is specified after Place and Route.

VIRTEX 5

Resource type	Quantity used
Slice register	106
Slice LUT	172
Block RAM	0
DSP48	0
Clock frequency (worst case)	305 MHz
Clock frequency (best case)	350 MHz

STRATIX III

Resource type	Quantity used
Register	135
ALUT	90
Block Memory bit	0
DSP block 18	0
Clock frequency (worse case)	300 MHz
Clock frequency (best case)	350 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	14/12/2009
1.1	Revised the reset scheme. New data enable signal. Added section on video signal output timing.	16/04/2011
1.2	Removed resync_sof signal - now use system reset instead. Updated synthesis results in line with source code optimizations for faster frame sync.	05/12/2011