CMOS Sensor Output Generator

Sahand Kashani

September 20, 2015

1 Core Overview

The `cmos_sensor_output_generator` core is an interface that generates the signals a CMOS sensor would normally output. The core is useful for testing CMOS sensor acquisition systems when the production sensor is not yet available.

The core is configurable at runtime through an Avalon Memory-Mapped (Avalon-MM) interface, and provides a Conduit interface identical to one a CMOS sensor would have.

Additionally, the core comes with a set of C library interfaces that can be used to configure it, as well as to start and stop its operation.

2 Generated Waveform

A CMOS sensor outputs 4 signals with which it is possible to sample its data:

- `clock`
- `frame_valid (1-bit)`
- `line_valid (1-bit)`
- `data (n-bit)`

Figure 1 shows the relationship between the different signals for 2 frames that contains 2 rows and 3 columns each.

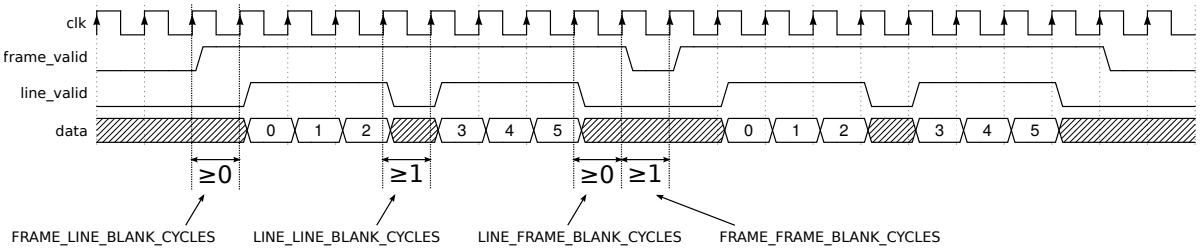

Figure 1: CMOS sensor output signals for two 2×3 frames with a pixel depth of 3 bits. Spacing requirements between the various signals are specified in clock cycles. The labels given to the 4 spacing intervals are the same ones used later in Section 4 to configure the core.

3 Block Diagram

Figure 2 shows a high-level view of the core. The `frame_valid`, `line_valid`, and `data` signals are generated with the same clock frequency as the core's input clock.

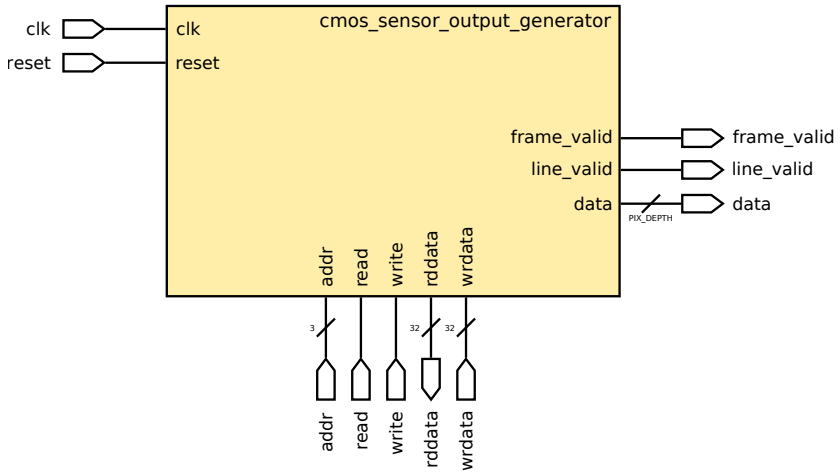

Figure 2: Block diagram.

Note that the core does not output the clock it receives as input as a normal CMOS sensor would do, because it is bad practice to route a clock from an FPGA's clock tree through standard logic in order to output it outside the clock tree.

It is the user's responsibility to use the same clock that is driving the core for any other component down the acquisition pipeline that requires the CMOS sensor's generated output clock.

4 Register Map

Table 1 shows the core's register map.

Offset	Type	Name	Default Value	Minimum
0x00	RW	CONFIG_ROW_SIZE	320	1
0x04	RW	CONFIG_COL_SIZE	240	1
0x08	RW	CONFIG_FRAME_FRAME_BLANK_CYCLES	1	1
0x0C	RW	CONFIG_FRAME_LINE_BLANK_CYCLES	0	0
0x10	RW	CONFIG_LINE_LINE_BLANK_CYCLES	1	1
0x14	RW	CONFIG_LINE_FRAME_BLANK_CYCLES	0	0
0x18	RW	CONFIG_RANDOM	false	N/A
0x1c	WO	COMMAND	N/A	N/A

Table 1: Register map. All registers are 32 bits wide. Note that the CONFIG_ registers can only be modified if the core is not running (you must issue a STOP command before modifying any of these registers).

You can configure the core to output sequential pixel values starting from 0, or to generate a random pixel value instead. Table 2 shows the values needed to configure the pixel value generation logic.

Name	Value	Description
SEQUENTIAL	0	Generate sequential pixel values
RANDOM	1	Generate random pixel values

Table 2: CONFIG_RANDOM Register Definitions.

The core is configured through multiple CONFIG_ registers. If a configuration register is written with a value smaller than those specified in column Minimum of Table 1, then the core has undefined behaviour.

You can start or stop the controller by writing a command to its COMMAND register, shown in Table 3.

Name	Value	Description
STOP	0	Stop generation
START	1	Start generation

Table 3: COMMAND Register Definitions.

5 Qsys Interface

Figure 3 shows the Qsys configuration interface for the core.

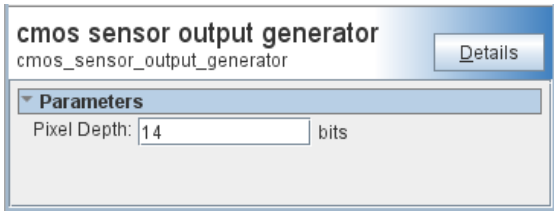

Figure 3: Qsys Configuration Interface.

The pixel depth that is to be outputted is the only parameter that must be set at generation time. All other aspects of the core are configured at runtime through its Avalon-MM interface, detailed in Section 4.