

# CMOS Sensor Acquisition

Sahand Kashani

February 28, 2016

## 1 Core Overview

The `cmos_sensor_acquisition` core is a configurable Qsys system that makes it simple to connect a CMOS sensor to a host processor & memory system.

It combines 3 sub-cores to create a *single*-component CMOS sensor acquisition system easily instantiable in Qsys:

- `cmos_sensor_input`
- `dc_fifo`
- `msgdma`

The core comes with a set of C library interfaces that can be used to configure it and start its various operations.

## 2 CMOS Sensors

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.

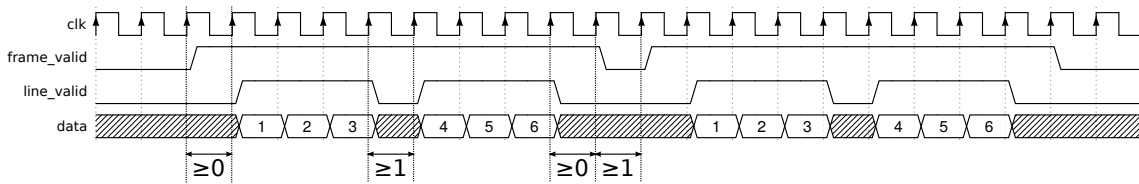


Figure 1: CMOS sensor output signals for two  $3 \times 2$  frames with a pixel depth of 3 bits. Spacing requirements between the various signals are specified in clock cycles.

### 3 Block Diagram

Figure 2 shows the connections between the 3 main building blocks that compose the `cmos_sensor_acquisition` core.

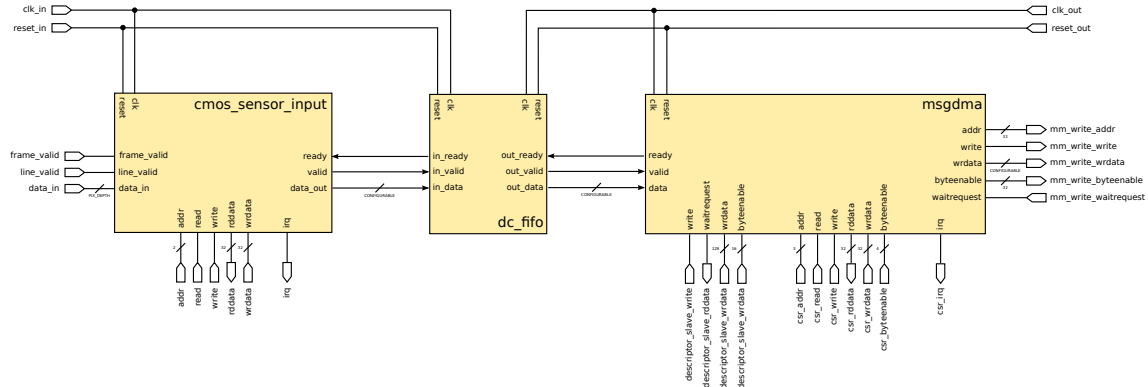


Figure 2: High-level block diagram.

### 4 Qsys Interface

Figure 3 shows the Qsys configuration interface for the core.

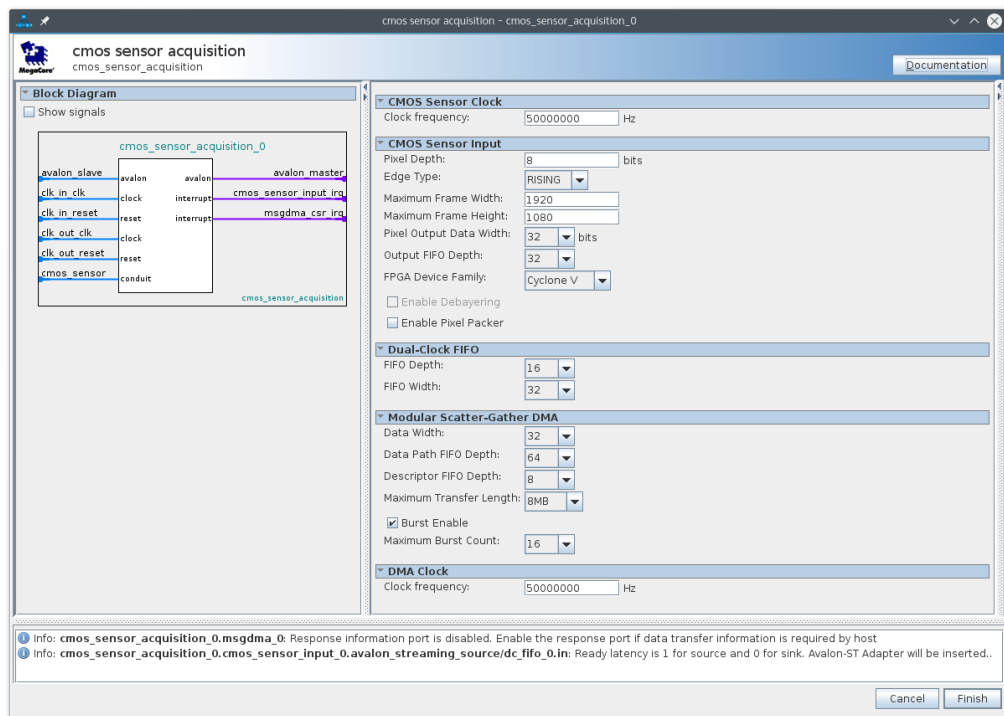


Figure 3: Qsys Configuration Interface.

It can be configured through 17 parameters, shown in Table 1.

Core	Parameter	Type	Values	Default Value
cmos_sensor_input	PIX_DEPTH	Positive	1, 2, 3, ..., 32	8
	SAMPLE_EDGE	String	"RISING", "FALLING"	"RISING"
	MAX_WIDTH	Positive	2, 3, 4, ..., 65535	1920
	MAX_HEIGHT	Positive	1, 2, 3, ..., 65535	1080
	OUTPUT_WIDTH	Positive	8, 16, 32, ..., 1024	32
	FIFO_DEPTH	Positive	8, 16, 32, ..., 1024	32
	DEVICE_FAMILY	String	"Cyclone V", "Cyclone IV E"	"Cyclone V"
	DEBAYER_ENABLE	Boolean	FALSE, TRUE	FALSE
	PACKER_ENABLE	Boolean	FALSE, TRUE	FALSE
dc_fifo	FIFO_DEPTH	Positive	16, 32, 64, ... , 4096	16
	FIFO_WIDTH	Positive	8, 16, 32, ... , 1024	32
msgdma	DATA_WIDTH	Positive	8, 16, 32, ... , 1024	32
	DATA_FIFO_DEPTH	Positive	16, 32, 64, ... , 4096	64
	DESCRIPTOR_FIFO_DEPTH	Positive	8, 16, 32, ... , 1024	8
	MAX_BYTE	Positive	1KB, 2KB, 4KB, ..., 2GB	8MB
	BURST_ENABLE	Boolean	FALSE, TRUE	TRUE
	MAX_BURST_COUNT	Positive	2, 4, 8, ... , 1024	16

Table 1: Qsys parameters.

## 5 Results

*All benchmarks results below were obtained using the default core parameter values shown in Table 1.*

The system was benchmarked on a "Cyclone IV E"-class device, and was able to successfully capture a frame with a resolution of  $1920 \times 1080$  pixels under the following timing constraints:

- 10 MHz cmos\_sensor\_input clock
- FRAME\_FRAME\_BLANK = 1
- FRAME\_LINE\_BLANK = 0
- LINE\_LINE\_BLANK = 1
- LINE\_FRAME\_BLANK = 0
- cmos\_sensor\_input packer disabled.
- 50 MHz msgdma clock

Note that no camera on the market can actually output frames with such tight frame-frame, frame-line, line-line, and line-frame timings. The benchmark was rather performed for demonstration purposes to show that the design is able to capture frames under tight constraints.

Furthermore, note that the cmos\_sensor\_input packer was *not* enabled, which would have allowed one to divide the pressure on the memory system by at least a factor of 2. It is recommended to always enable this option, but again, it was left out for demonstration purposes to show the performance of the core.

## 6 Qsys bugs

Bug rd05212011\_256 may cause issues for the users of the `cmos_sensor_acquisition` core. This bug is caused by the verilog wrapper generated by Qsys to surround the VHDL entity of the `cmos_sensor_input` core contained in this design.

Essentially, the `cmos_sensor_input` core has a few `Boolean` generic parameters, but the generated verilog code passes 0 (or 1) instead of `"false"` (or `"true"`). VHDL being a strongly-typed language, these values are rejected by the synthesizer.

To get around this issue, you must modify the generated verilog wrapper file where the core is instantiated, and replace the boolean parameters by `"false"` (or `"true"`) where required.

For example, you would replace the following instantiation code:

```
1      cmos_sensor_input #(
2          .PIX_DEPTH      (8),
3          ...
4          .DEBAYER_ENABLE (0),
5          .PACKER_ENABLE  (1)
6      ) cmos_sensor_input_0 (
7          .clk             (clk_in_clk_clk),
8          .reset           (rst_controller_reset_out_reset),
9          ...
10     );
```

with the following (changes shown in **red**):

```
1      cmos_sensor_input #(
2          .PIX_DEPTH      (8),
3          ...
4          .DEBAYER_ENABLE ("false"),
5          .PACKER_ENABLE  ("true")
6      ) cmos_sensor_input_0 (
7          .clk             (clk_in_clk_clk),
8          .reset           (rst_controller_reset_out_reset),
9          ...
10     );
```

Remember to keep the quotes, as they are *necessary*.