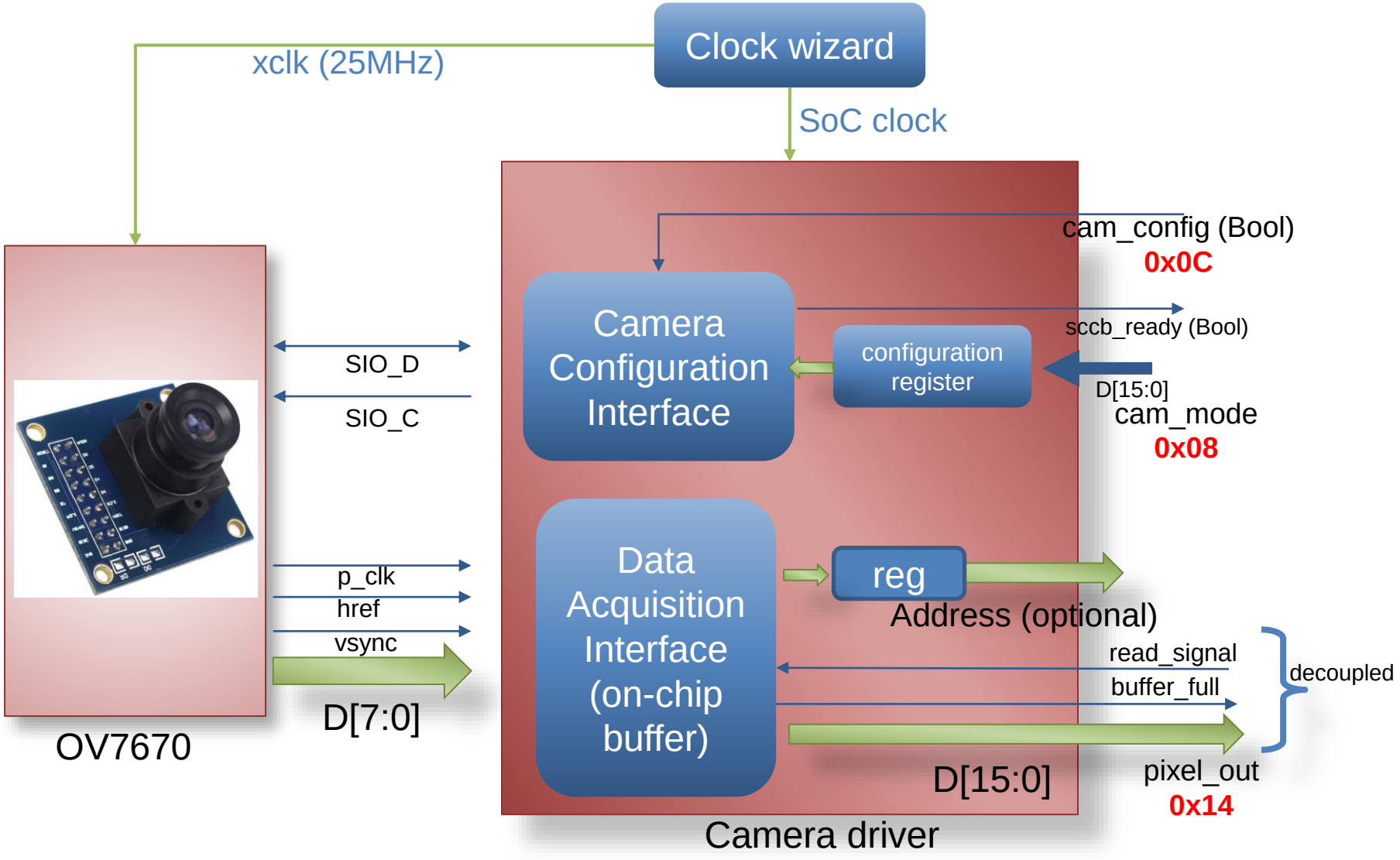
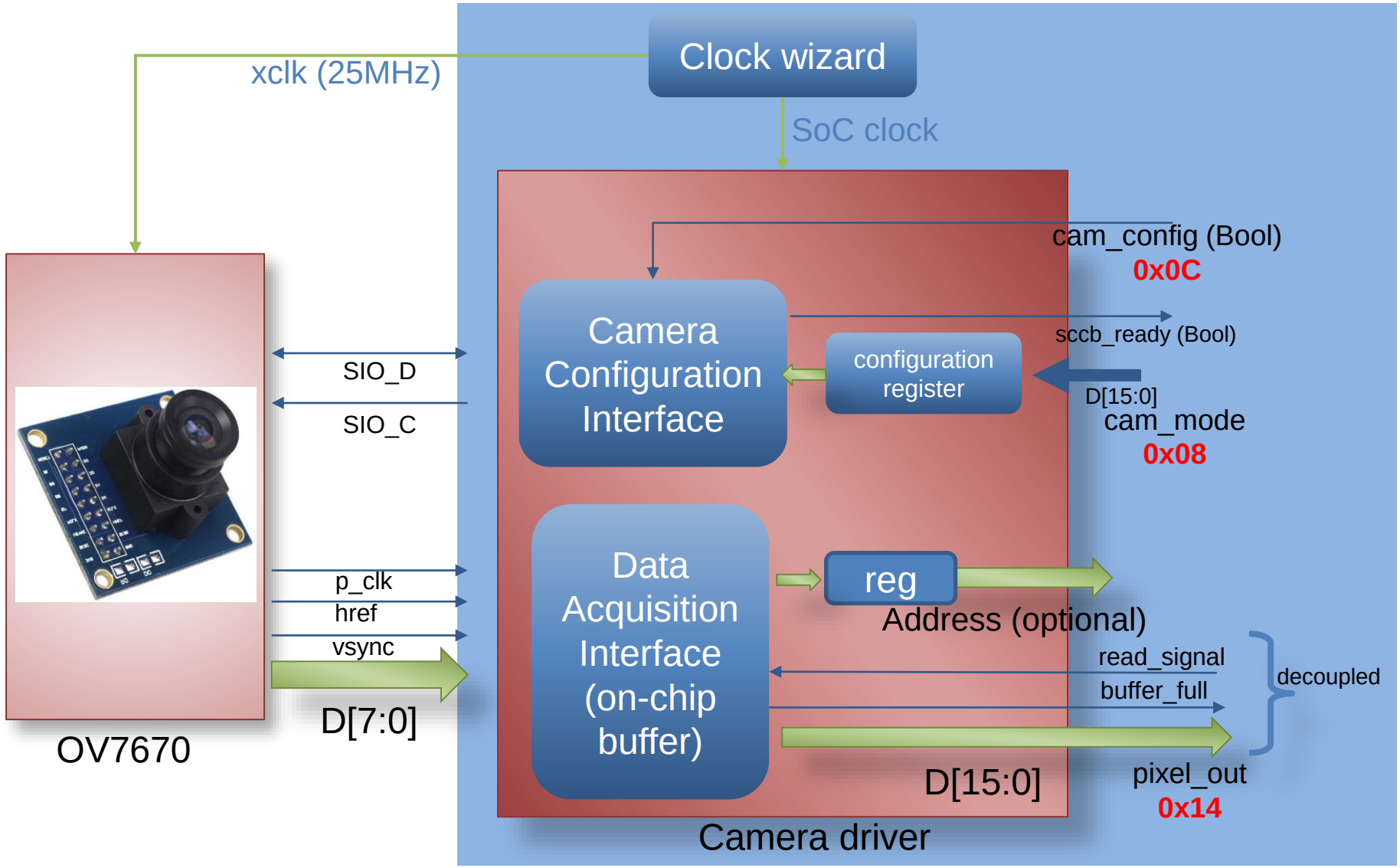
Camera interface top module



Camera interface top module



Camera interface external IO pins

pin names	pin type	frequency	Function
XCLK	Output	24MHz (typ)	System clock Input
SIOC	Output	<400kHz	SCCB serial clock
SIOD	Output		SCCB serial data
HREF	Input		HREF output
VSYNC	Input		Vertical sync output
PCLK	Input		Pixel clock output
D[7:0]	Input		YUV/RGB video component output bits

Total number of IO pins needed for the interface: 14 pins +2 power pins

Since we only need to configure the camera without the concern to read contents of control registers via the SCCB interface, SIOD is unidirectional

=> no tri-state IO pin is used here

CPI Parameters and MMIO registers

```
case class CPIParams(
  address: BigInt =0x10003000,
  useAXI4: Boolean = false,
  img_width: Int=640,
  img_height: Int=480,
  byte_per_pixel: Int =2,
  SCCB_FREQ_kHz: Float=100,
  CLK_FREQ_MHz: Float= 50,
  pixel_width: Int=16
)
```

```
object CPIMOMI{
  val cam_status=0x00   //wire
  val cam_capture=0x04
  val cam_mode=0x08
  val cam_config=0x0C
  val set_image_resolution=0x10
  val returned_image_resolution=0x14
  //val read_frame=0x18
  val pixel=0x18
}
```

- img_width and img_height is used to specify the depth of the frame buffer
  - Img_width=640 and img_height=480 if want to capture the whole VGA frame

- Register mapped IO of the camera interface

OV7670 memory-mapped registers

Address (offset)	Name	Data type associated with the MMIO	Functions
0x00	cam_status	One hot (3b)	Holds the sccb status, buffer status
0x04	capture_frame	Bool	Generates a signal to capture an image. This signal is held when the camera is busy
0x08	cam_mode Cat(addr,data)	16 bit	Used to configure the camera
0x0C	config_cam	Bool	Used to enable the sccb interface to transmit a new mode
0x10	cam_resolution	19 bits	Specifies the resolution for a mode, configured by software
0x14	cam_resolution feedback	19 bits	Returns the resolution of an image based on PCLK, HREF, VSYNC
0x1C	acquired pixel	De-coupled	Has the valid-ready interface to read image from the buffer

Instructions for SCCB testing

- Module class name:

  SCCB_interface(CLK_FREQ_MHz: Float, SCCB_FREQ_kHz: Float).

- Test class name:

  SCCB_iinterface_test(dut: SCCB_interface)(n_of_random_test: Int)

  n_of_random_test: the number of times to test the interface with random configuration

- To run the simulation, call the test class as follows in a main (either app or Matchers to generate vcd file:

  new SCCB_interface_tests(c)(n_of_random_test=50) for example

- Test result is displayed as follows

```
[info] [0.002] SEED 1630050299422
blue texts indicate the testing results
test result: 20 tests passed over 20 being tested
```

Instructions for testing capture_module

- Module name:

  class camera_module(img_width:Int, img_height: Int,
  byte_per_pixel: Int)

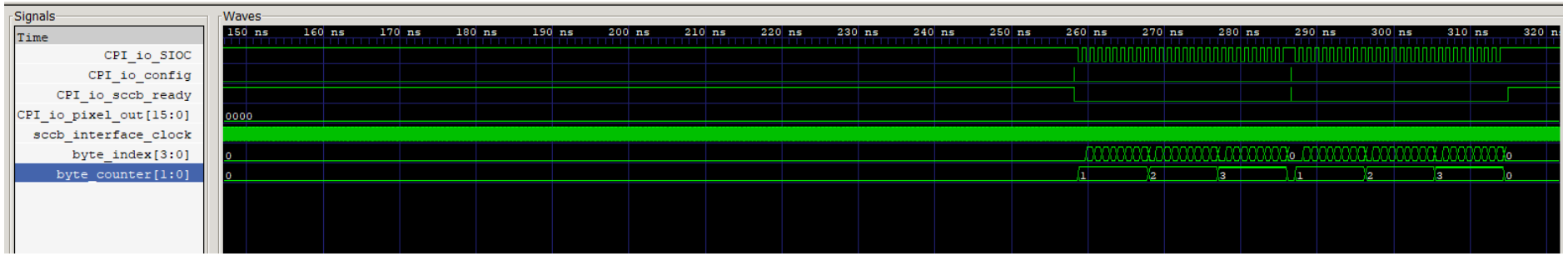
- Test class name:

  class capture_module_tester(dut:camera_module)(n:Int)

  Where n is a scalar, it must be greater than 1 to simulate pclk

  – There are two txt files in this test:
    - The first one, random_frame_0v7670_ver1.txt, is used to transmit each byte of a pixel to the DUT (2 consecutive values equal to one pixel)
    - The second one, pixel_val_frame_ver1.txt, holds standard pixel values of 2 consecutive values in the previous file. This file is then used to check with pixel values read from DUT's buffer with the "expect" function

Software configuration for controlling the SCCB interface



Waveform of the SIOC and SIOD when reset the camera and configure RGB565 mode