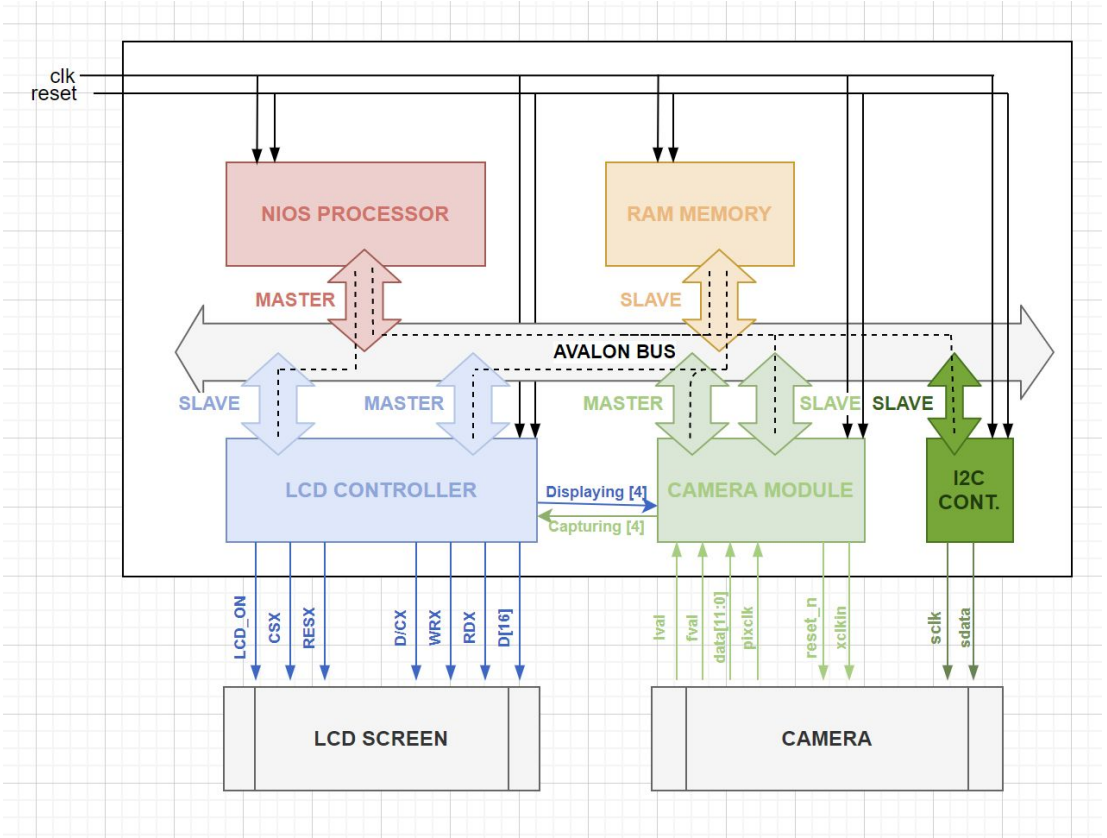
Lab 4.0

TFT-LCD Display + Camera

O. Facklam, J. Schweizer, S. Thirion

Global system

- 4 main modules
- 2 external interfaces: GPIO_0 and GPIO_1
- Synchronization of LCD and CAMERA controller with 2 conduits



Storage in SDRAM

4 buffers, 1 frame each

Access via Avalon Master (4 bytes)

- Write - burst of 80 words (2 pixels = 4 bytes)
- Read - burst of 40 words (2 pixels = 4 bytes)

[3:0] conduits between modules for concurrent access

- ***Conduit displaying*** - buffer being read
- ***Conduit capturing*** - buffer being written in

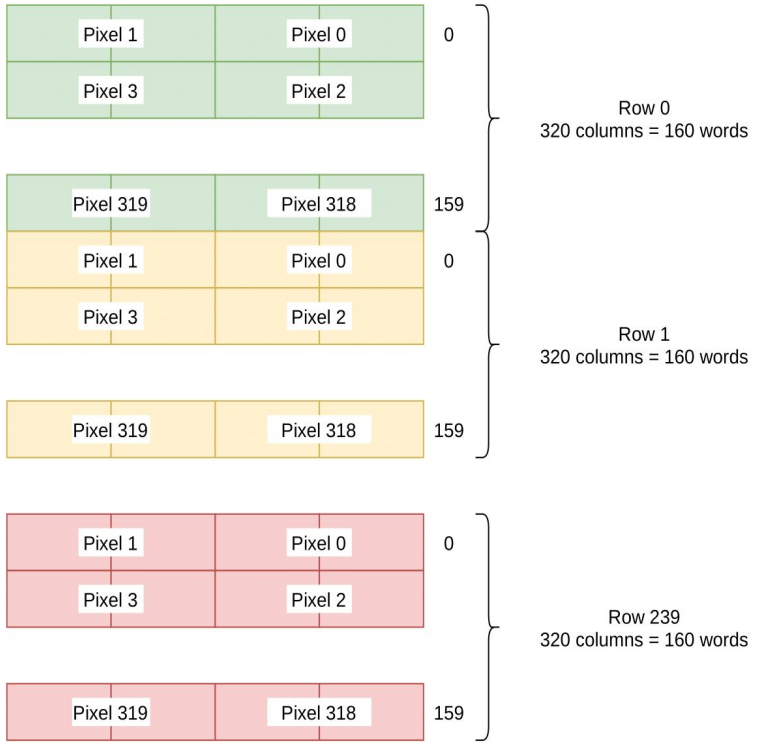


Figure : Organisation of 1 buffer

Camera module (general)

Registers

Buffer config storage

FSM

Synchronization

Acquisition management

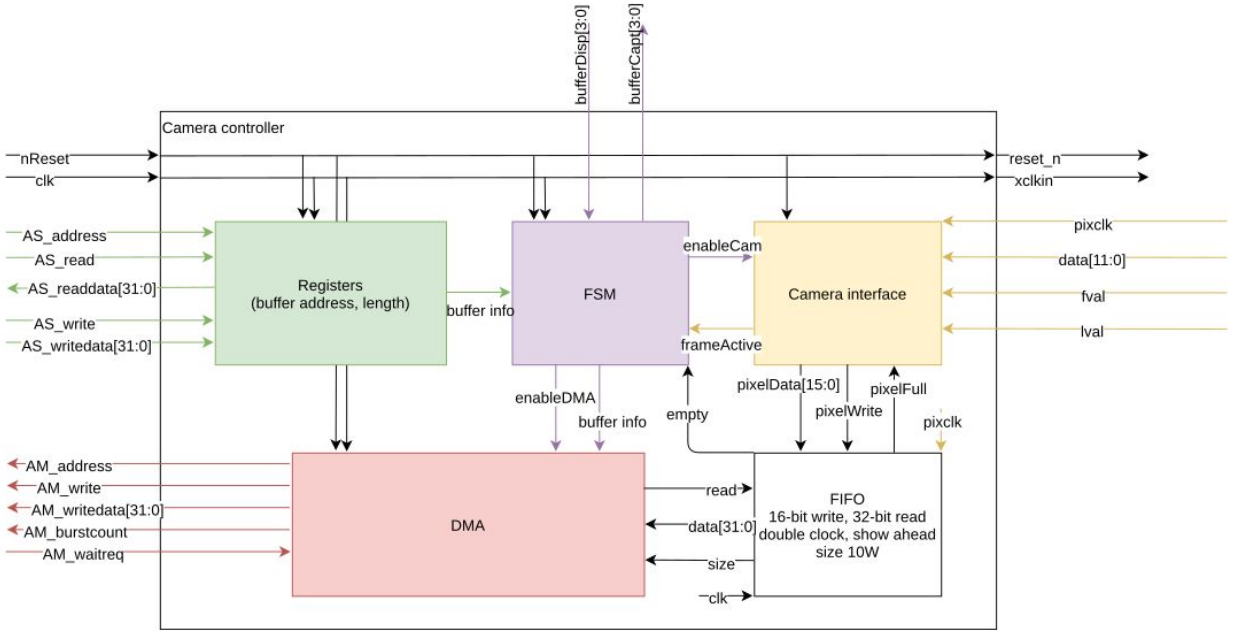
Camera interface

Sampling, debayerization

DMA

FIFO dequeuing

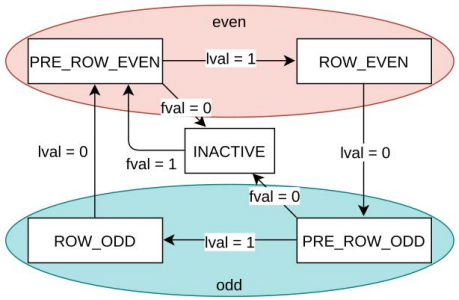
SDRAM write

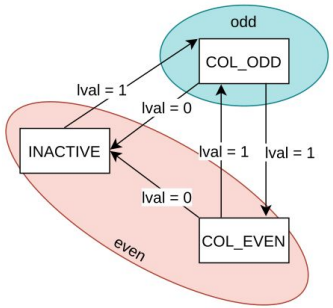


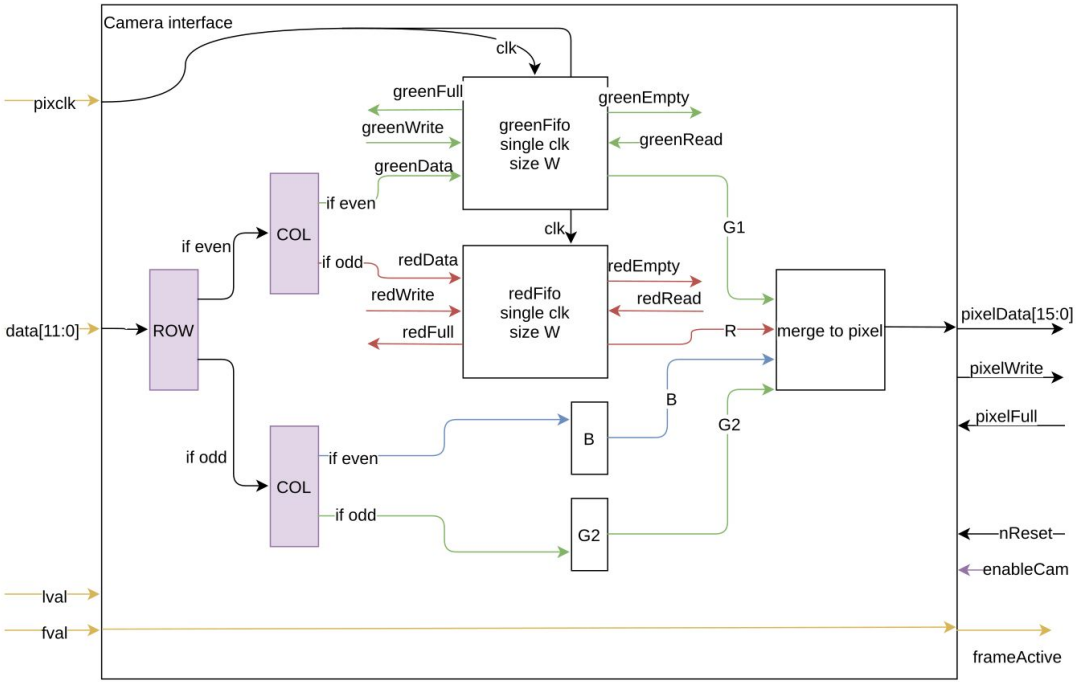
4 external interfaces: Avalon master, Avalon slave, Camera input, LCD synchro conduit

Camera module (camera interface)

- Simple state machine to track row/column parity
- Store even rows in green & red FIFO
- Merge 4 colors during odd rows

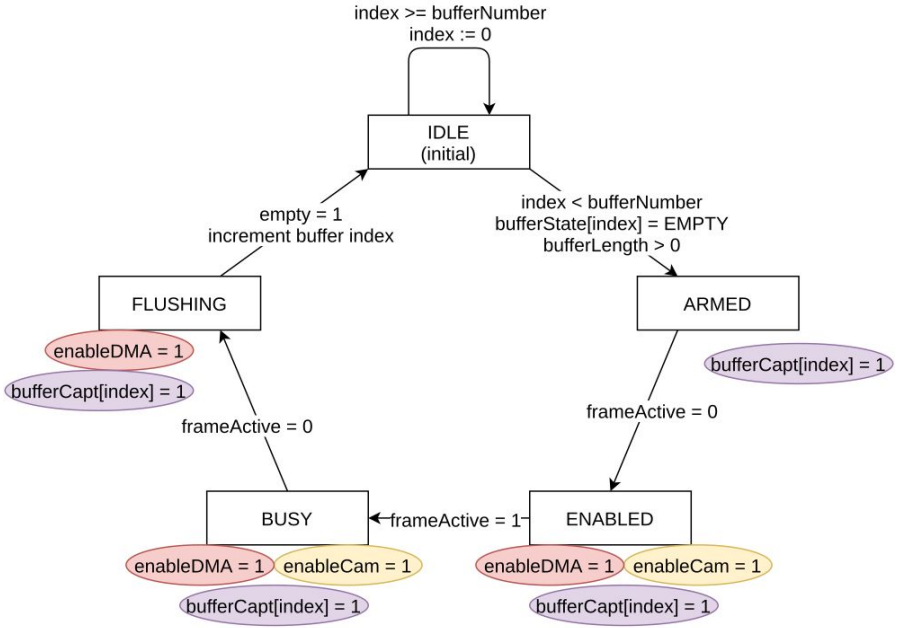




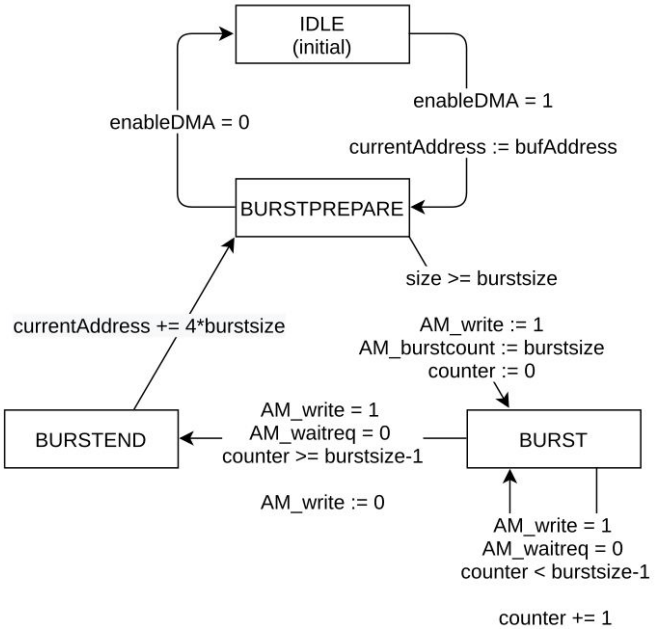


Camera module (FSM & DMA)

FSM: track buffer index & camera frame start



DMA: burst counter & current address registers



LCD module (general)

Address	Register name	Bits	Explanation	Default value
0x00	MODENABLE	0	Enable the lcd controller module	0
0x01	FRAMESEC	0-7	Frame per second of the transfer	25
0x03	BUFFADDR	0-31	Address in memory of the first buffer	0x00000000
0x04	COMMANDLCD	0-15	Send a command to LCD	None
0x05	DATALCD	0-15	Send a data to LCD	None

Master controller

SDRAM read, FIFO enqueuing

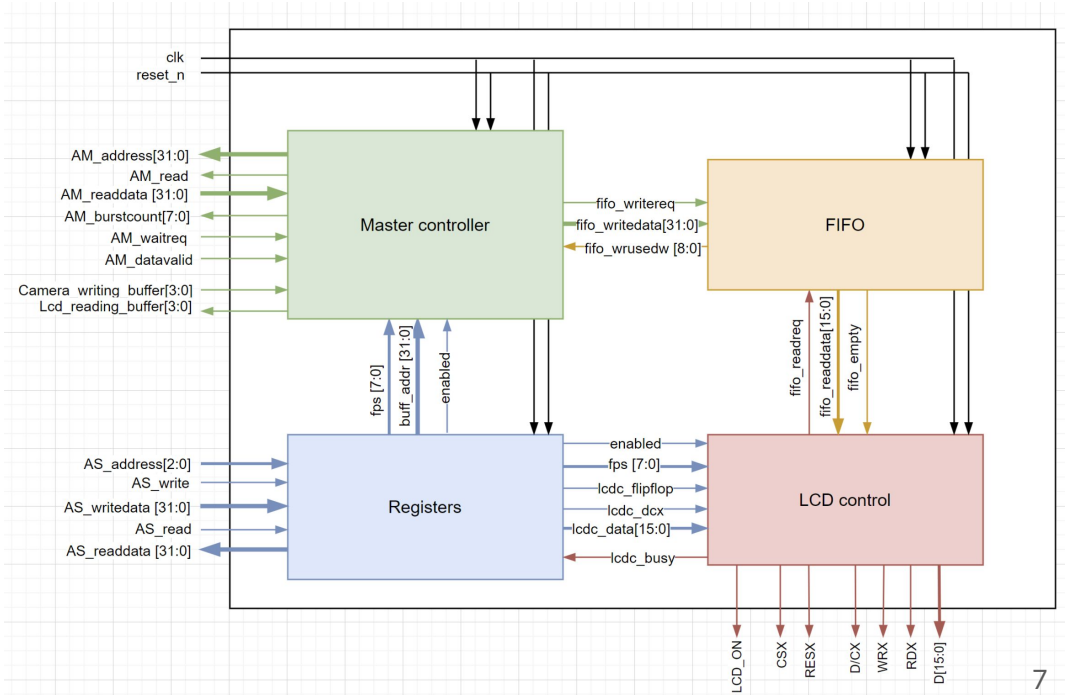
LCD controller

FIFO dequeuing, transmit commands

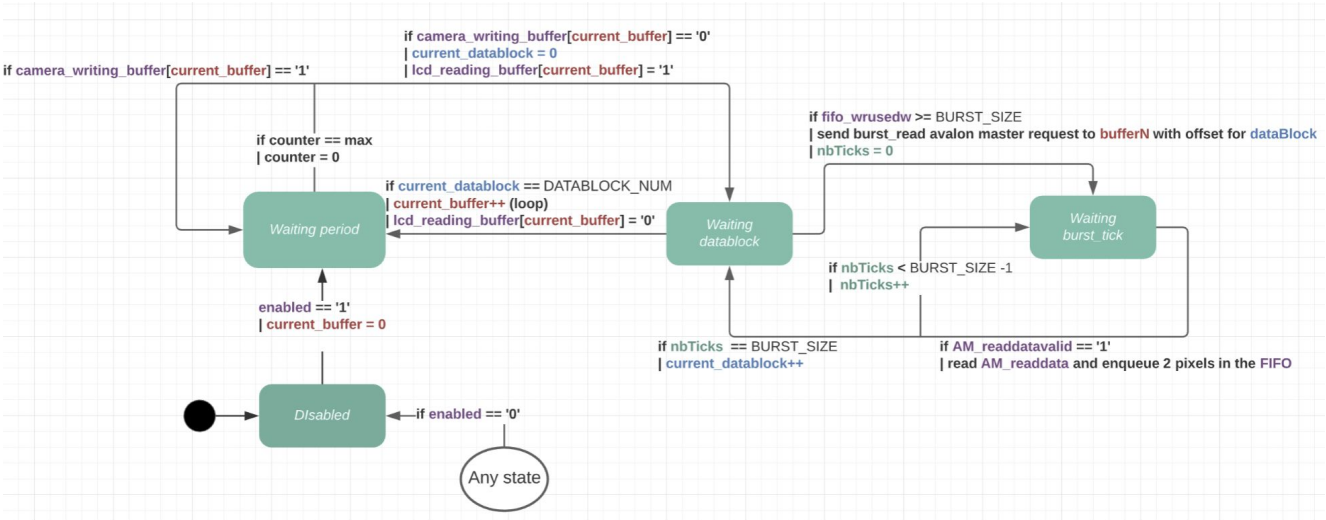
Register

Global variables, transmit commands to LCD controller

4 external interfaces: ***Avalon Master, Avalon Slave, Camera Conduit*** and ***Lcd Output***



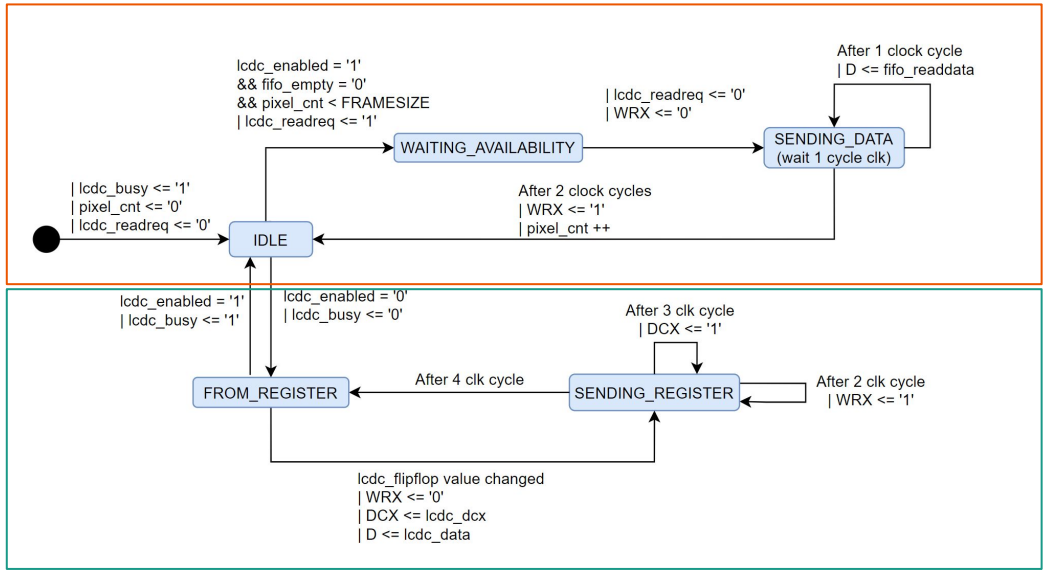
LCD module (master controller)



- Needed : ***Buffer Address & FPS***
- What FPS value? Not <u>too low</u> nor <u>too high</u> ! (it is an upper bound)

LCD module (LCD controller)

- Avoids pulling too fast with *FPS*
- Fastest pixel transmission possible (80 ns / pixel)



Merged system

Test of camera module

- ModelSim validation
- Save picture from camera to SDRAM
- Retrieve picture from SDRAM

Test of LCD module

- ModelSim validation
- Write image from software (commands)
- Set SDRAM buffers to predefined values

Merged component successful ! 🎉

Software programming

- Init of the camera & camera controller:
  - Programming the camera registers (image size, skipping, binning)
  - Programming custom registers (buffer address, number, length), automatically starting capture of images

- Init of the LCD controller
  - Programming custom registers (fps, buffer_address, ...)
  - Programming the LCD screen (pixel format, color, gamma correction, control ...)
  - Displaying small screen boot
  - Starting the display of images

A success story...

… to be improved

- LCD screen & camera configuration (exposure & saturation)
- FPS - find and remove bottleneck (currently ~10 fps)
- Small random lag & saturated pixels when overexposed scenery

Thanks!