CS-473 Embedded System

Lab Report 4.0

LT24 LCD Controller Implementation

Students:
Chuanfang Ning 320662
Alexander Sigrist 296025
The collaborating group:
Aliaa Diab 302713
Raphael Linsen 322834

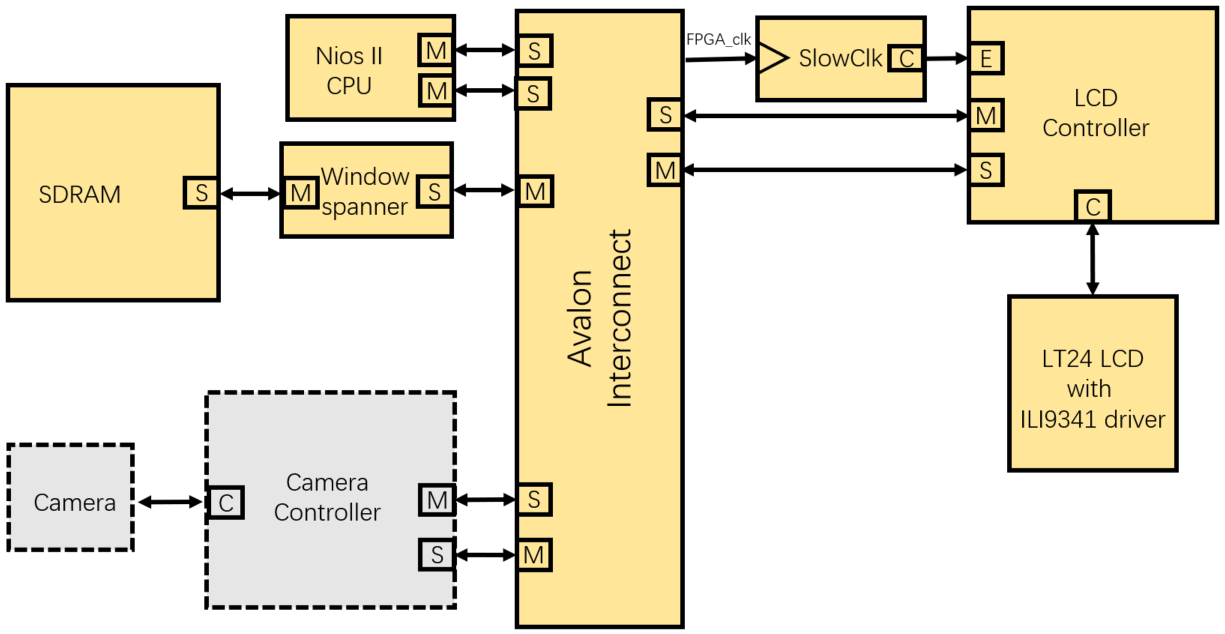
Introduction

Problem statement

In Lab 3.0, the goal is to propose a detailed design for an FPGA-based system which can interface with an LT24 LCD display by Terasic. The design will include a master unit that can initiate transfers on the Avalon bus to fetch image data frames from memory. This LT24 sub-system will later be combined with a sub-system developed by another team that handles reading image data from a TRDB-D5M camera to memory. In order to ensure compatibility and functionality of the full-system, the shared memory resource must be agreed upon and specified in detail.

System Diagrams

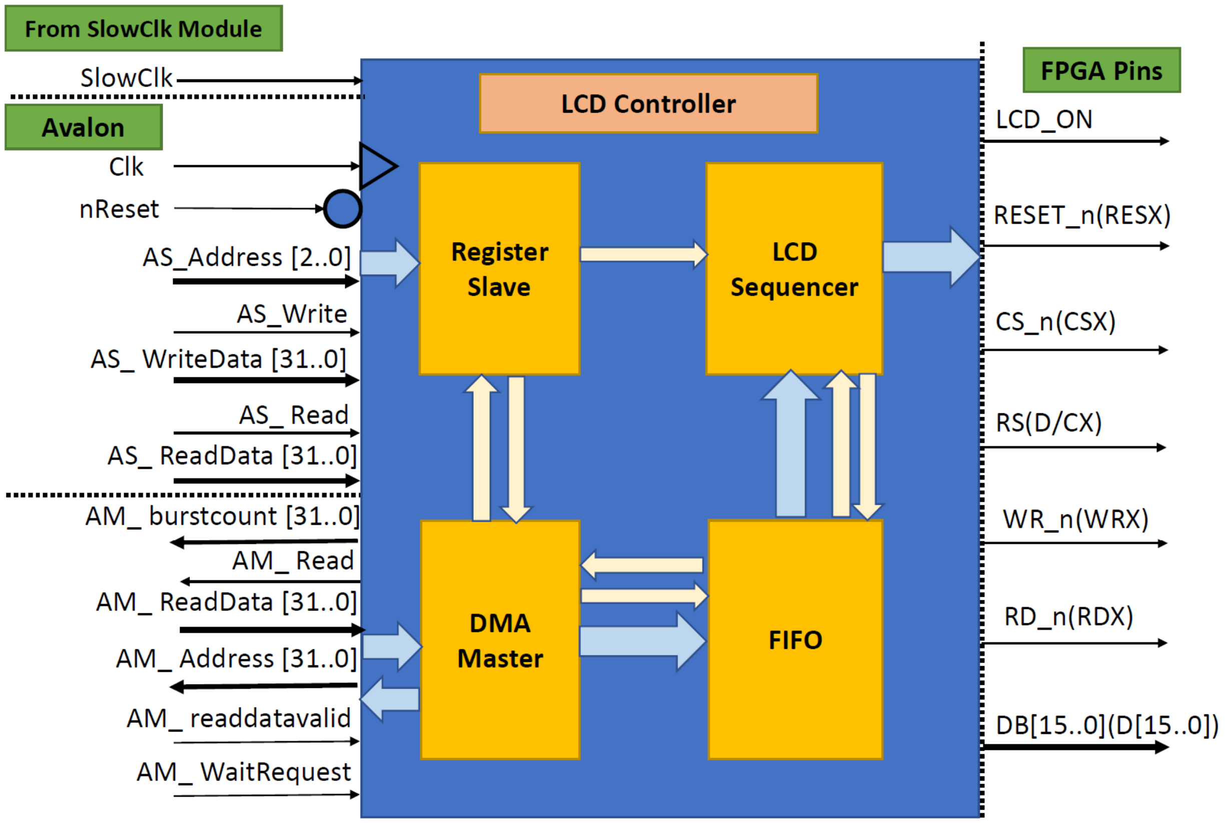
Full System Block Diagram



A SlowClk module, identical to that used in Lab 2.2, will be added to divide the 50 MHz FPGA_clk signal by 4. This will produce a 12.5 MHz clock with a period of 80 ns that will allow the LCD controller submodule FSM to generate low-level 8080-I MCU parallel interface signals with the correct timings.

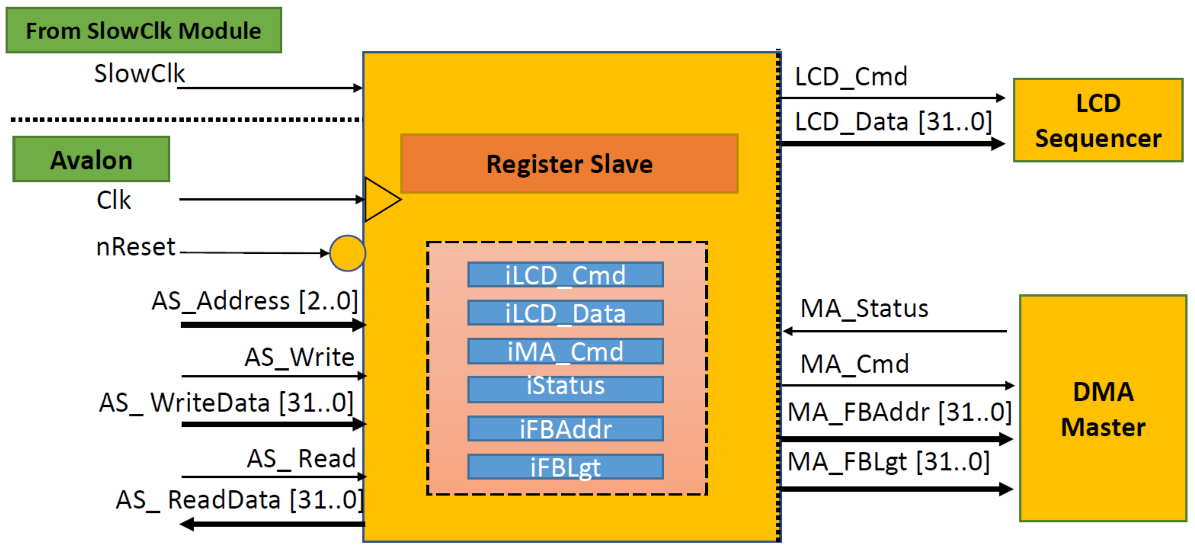
Custom IP Block Diagram

The custom LCD controller IP block is composed of 4 submodules as shown in the figure below: An Avalon slave memory-mapped register interface for programming, an LCD sequencer to send commands and RGB signals, a FIFO (implemented using the included IP block in Quartus) and a Master Controller which implements the DMA functionality.



Slave Register Interface

This block includes an Avalon slave interface which allows the Custom LCD Controller to be programmed by the NIOS II Softcore CPU. All the information about available registers and their addresses are listed in the Custom IP register map. The SlowClk is connected to the Avalon slave register interface submodule. The LCD_Cmd signal will be asserted to the desired value written from the Avalon interface by the NIOS II processor for one SlowClk cycle. Similarly, the MA_Cmd signal will be asserted for one FPGA clk cycle. This allows the command to trigger the respective FSMs to do a ‘single-shot’ execution of an FSM cycle that returns to an idle state. More details in Custom IP finite state machine (FSM) diagram.

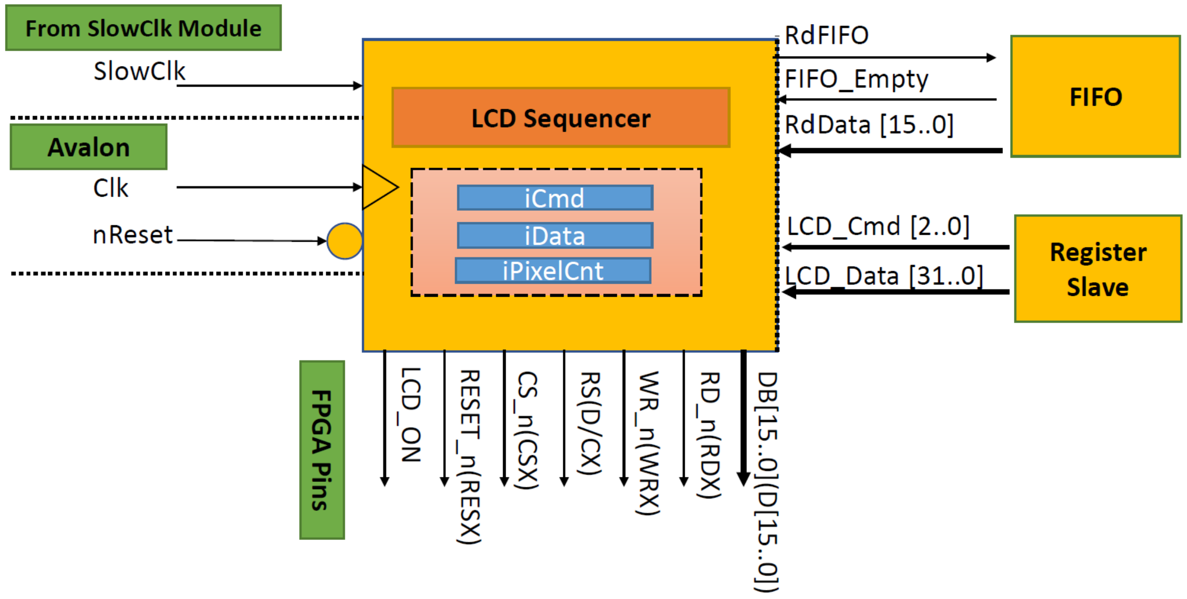


LCD Sequencer

The LCD sequencer submodule is responsible for generating the 8080-I 16b parallel bus MCU interface signals with the correct timing required to communicate with the ILI9341 IC. This controller will have several different modes, which are switched according to the LCD_Cmd register.

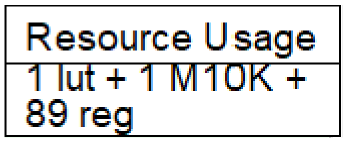
"CPU mode" will be for configuration where the LCD controller submodule will set configuration registers (register address/command then register data/parameters) in the ILI9341 on startup and reset as well as before each new image frame. It is also responsible for hard resetting the ILI9341 by toggling RSX pins. There are 4 sub-modes in the "CPU mode" which respectively send command address, send command data, and set or reset the RESX signal to the ILI9341. Which sub-mode to enter is controlled by asserting the value of LCD_Cmd (valid values are 1, 2, 4, 5).

When the value of LCD_Cmd is asserted to 3 by the Register Slave, the LCD sequencer enters "DMA mode", in which it will automatically dequeue an image data frame from the FIFO and write it to the LCD.



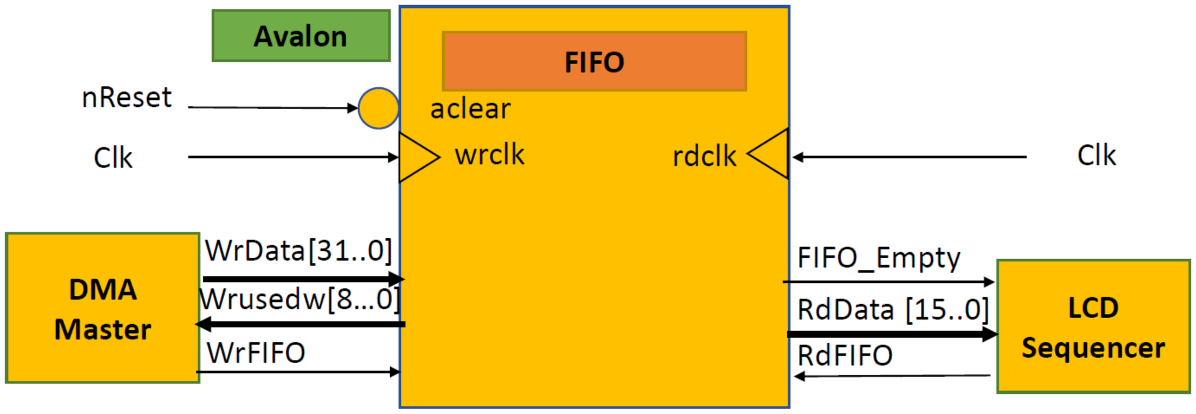
FIFO

The FIFO is configured with a size of 256 words; it was sized to store the results of several burst transfers and to efficiently use the full size of the allocated 10kb memory block in the FPGA. The FIFO is configured to have separate read and write clocks with 32b write width and 16b read width. This will allow the DMA Master to write words from memory (composed of two 16b doublet RGB pixels each) into the FIFO and the LCD sequencer to read 16b RGB pixels from the FIFO in an efficient manner. The used words (wr*usedw[]*), with an additional bit to prevent roll-over, will be exported from the write side as an input to the DMA Master Controller to check if there is sufficient space in the FIFO to store the data from a burst-read from memory. The *empty* signal will be exported on the read side to tell the LCD controller if there are still pixels in the FIFO to read and display. The FIFO is configured in show-ahead synchronous mode so that data appears before the assertion of RdData signal, thus the LCD Sequencer doesn't need to wait for the data to be available to read from FIFO. The aclear register of FIFO should also be configured to implement an asynchronous reset so that the pixel data is cleared in the case of a reset and the system begins from a valid initial state. The FIFO's FPGA resource usage is presented in the figure below:



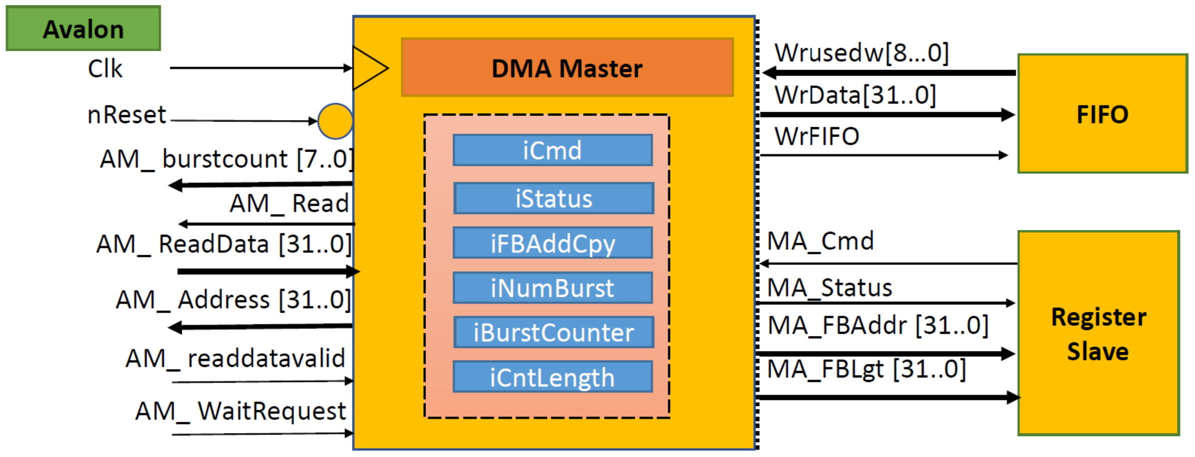
Resource Usage
1 lut + 1 M10K + 89 reg

FIFO Resource Usage



DMA Master Controller

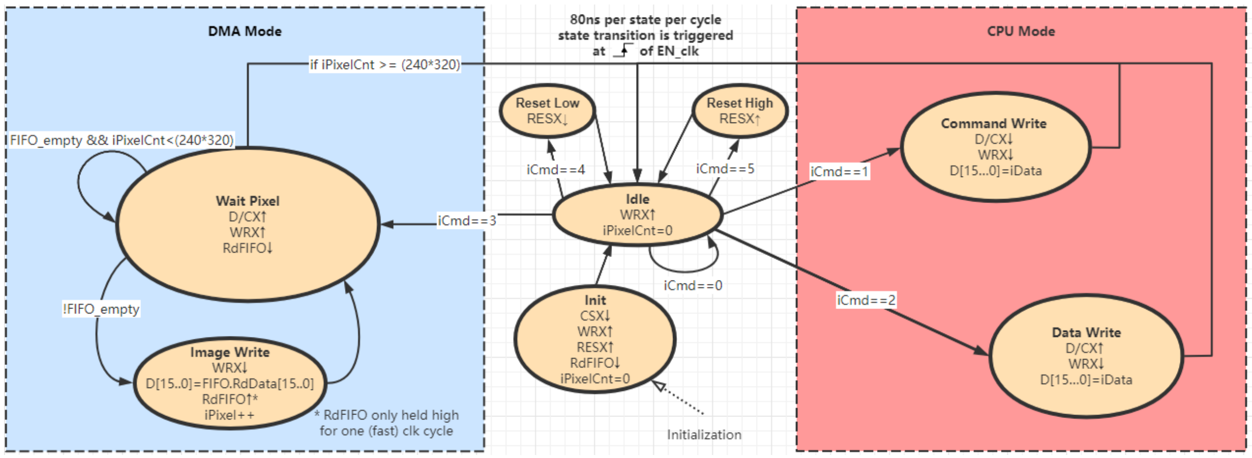
The DMA Master Controller consists of an Avalon master interface that can initiate transfers on the Avalon bus. It is responsible for fetching data from the SDRAM and writing it into the FIFO using burst transfers of 16 word lengths. It checks the w*rusedw[]* signal from the FIFO to ensure there is enough free space to store the data from the burst transfer. This master controller unit is configured using the slave register interface where it gets a starting address of the image frame buffer in memory (MA_*FBAdd*), the length of the data buffer (MA_*FBLgt*) and a signal (MA_Cmd) to start. It performs a total number of $\frac{320\times240\ px}{16\ words/burst\times2\ px/word} = 2400$ burstreads for one frame. At the start of each burstread, it initializes the address and burstcount. At the start of each burst it increments the address and decrements the burstcount. On reaching 1 of burstcount , the DMA Master Controller Starts a new burst until the entire frame is read into the FIFO. At this point it sets a register (MA_Status) in the register interface which the NIOS II processor will poll to determine if the operation is complete. When the operation is complete, the NIOS II will set the FbAddr and FBLgt for the next buffer and start the DMA and the LCD sequencer again.



Custom IP finite state machine (FSM) diagram

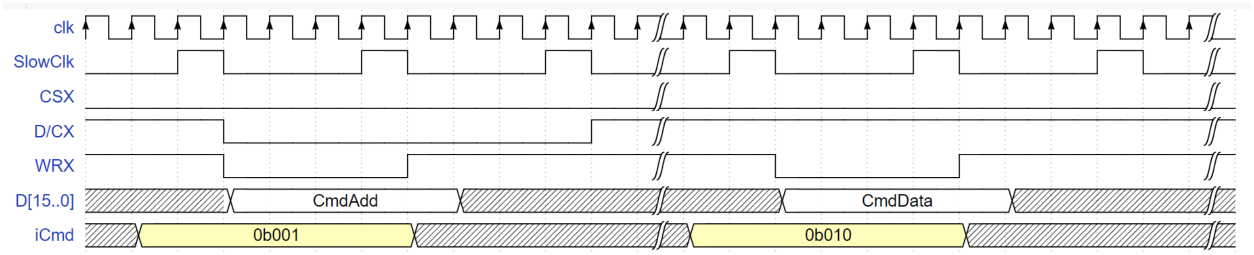
The state machine for the LCD Sequencer is presented below. There are two main operating modes "CPU mode" to perform startup/reset sequence commands (address and data) and "DMA mode" to send RGB data from the FIFO to the ILI9341.

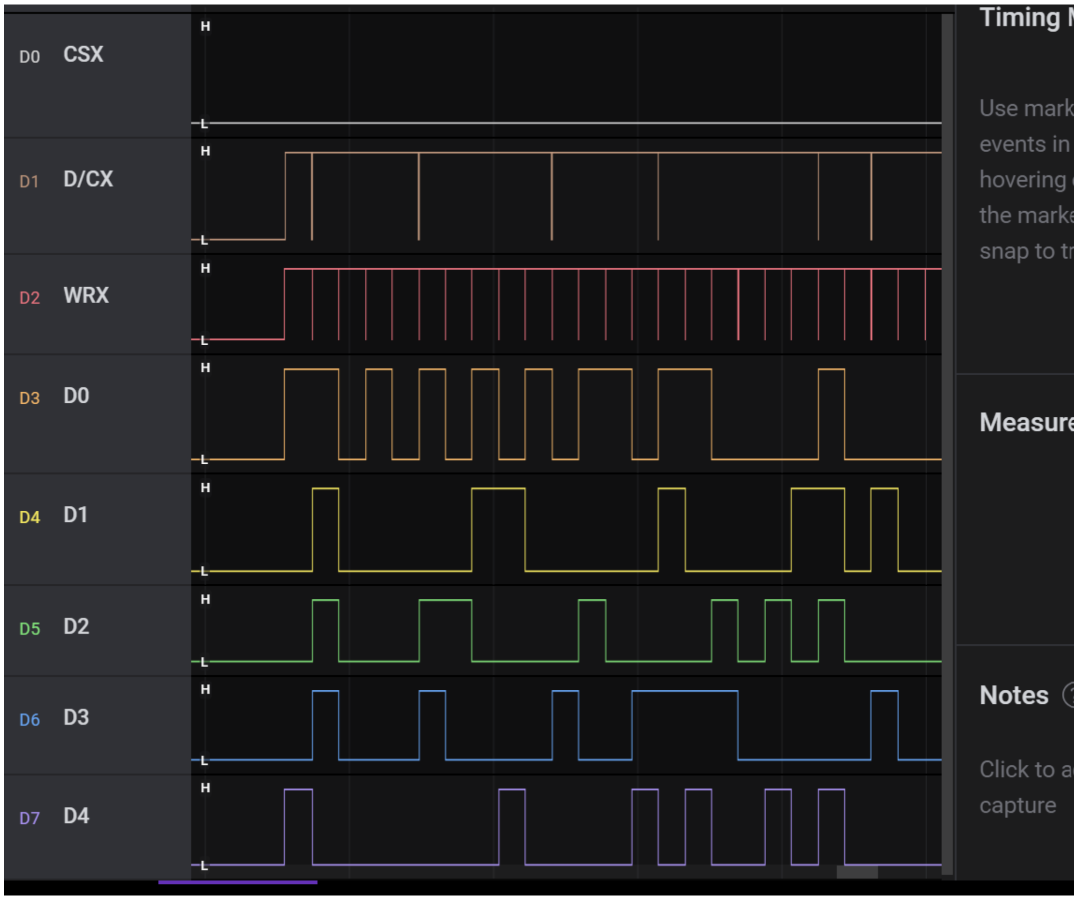
LCD Sequencer State Machine



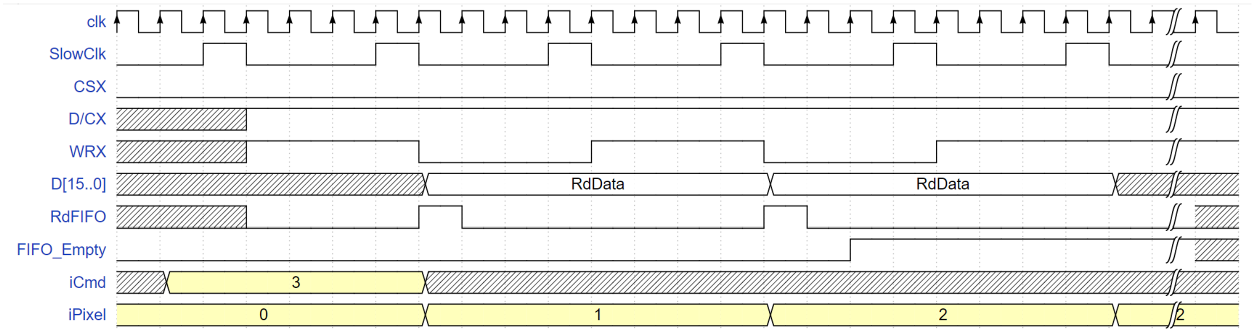
Timing Diagrams

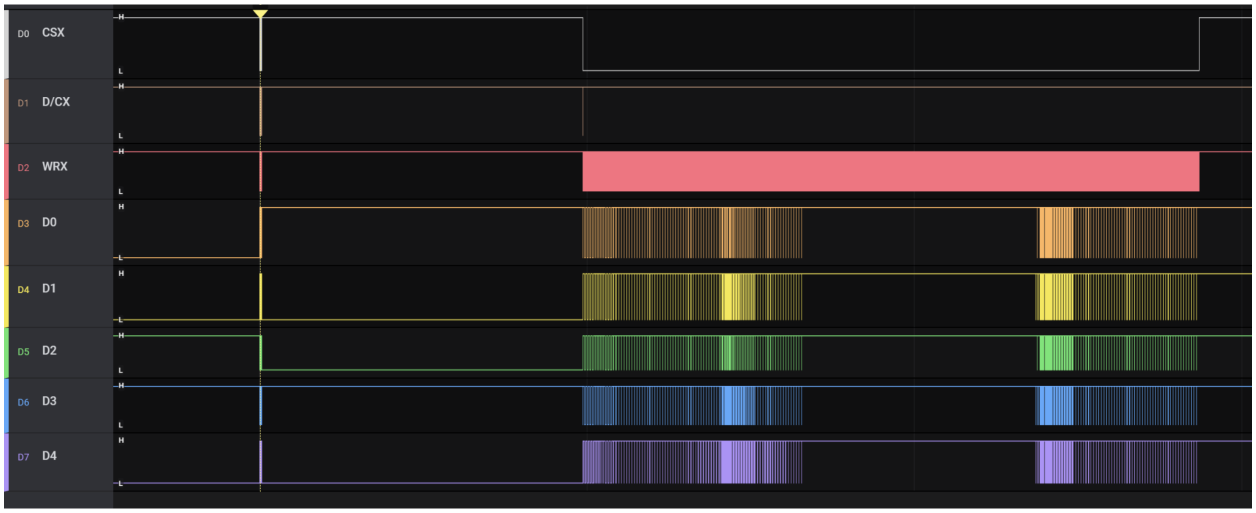
The waveform of Signals in "DMA Mode" and "CPU Mode" are shown below to help understand the FSM of LCD Sequencer.
The following waveform shows the **command address writing** and **command data writing** in "CPU mode" of the LCD Sequencer in design and test:



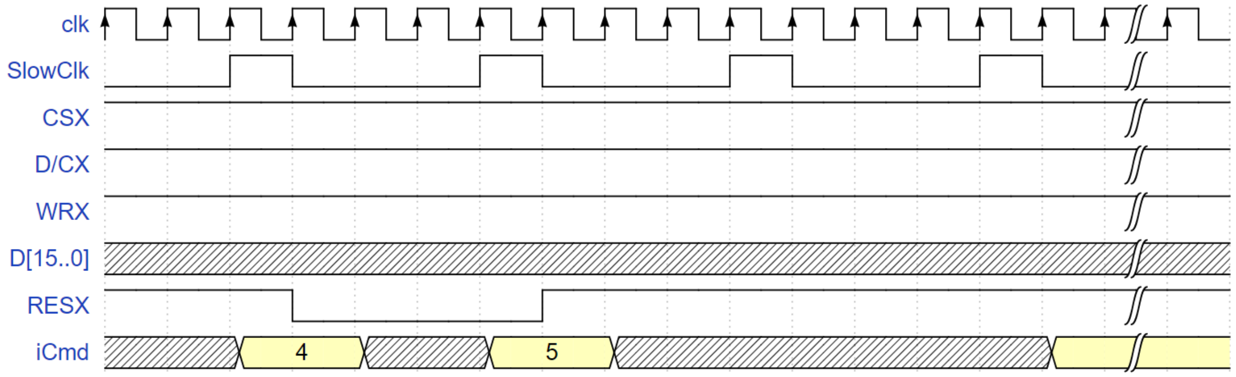


The following waveform shows the "**DMA mode**" of the LCD Sequencer in design and test:

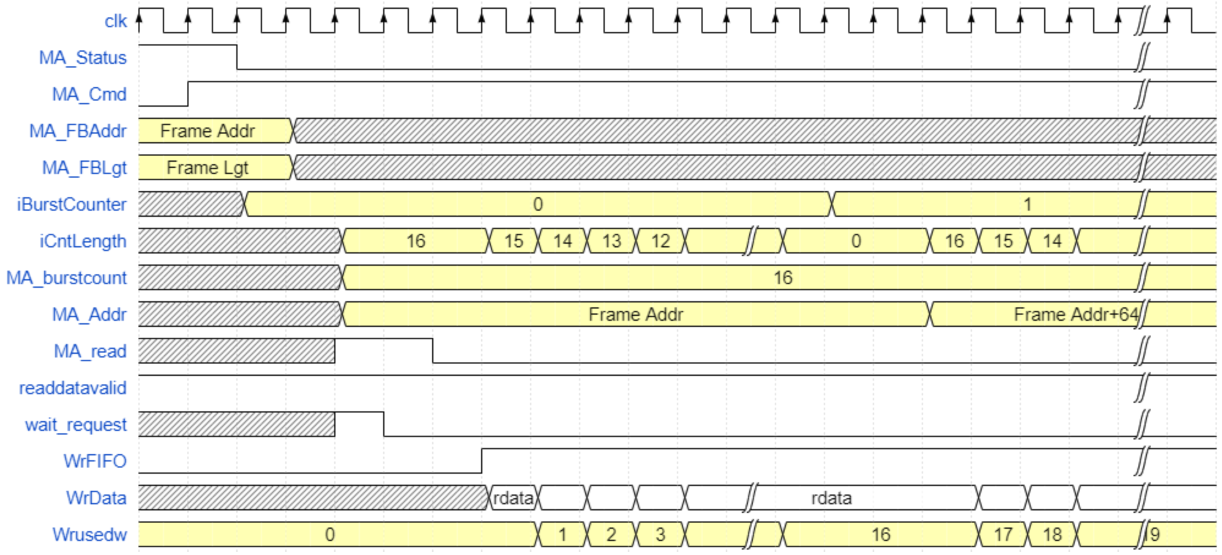




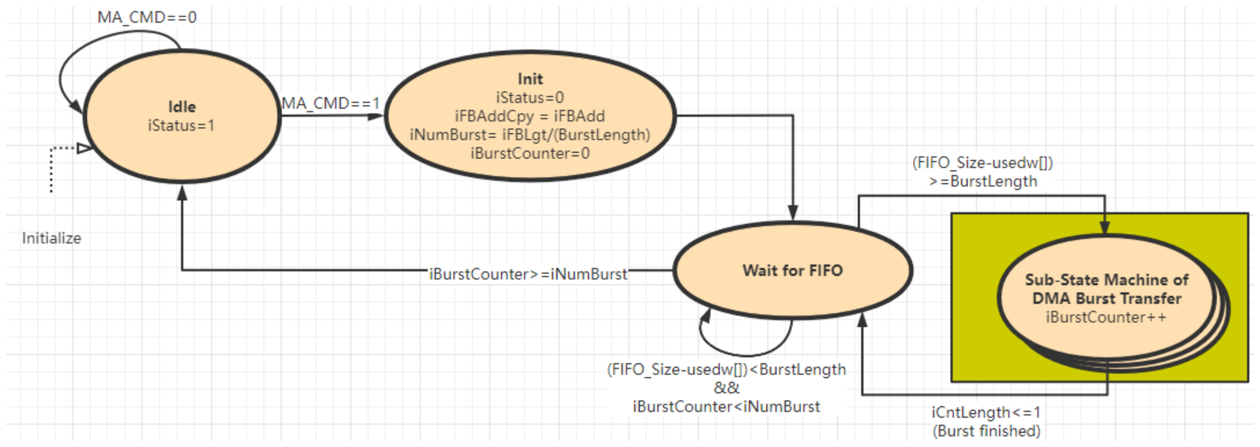
The following waveform shows the **set** and **reset** of the ILI9341 in “CPU mode” of the LCD Sequencer:



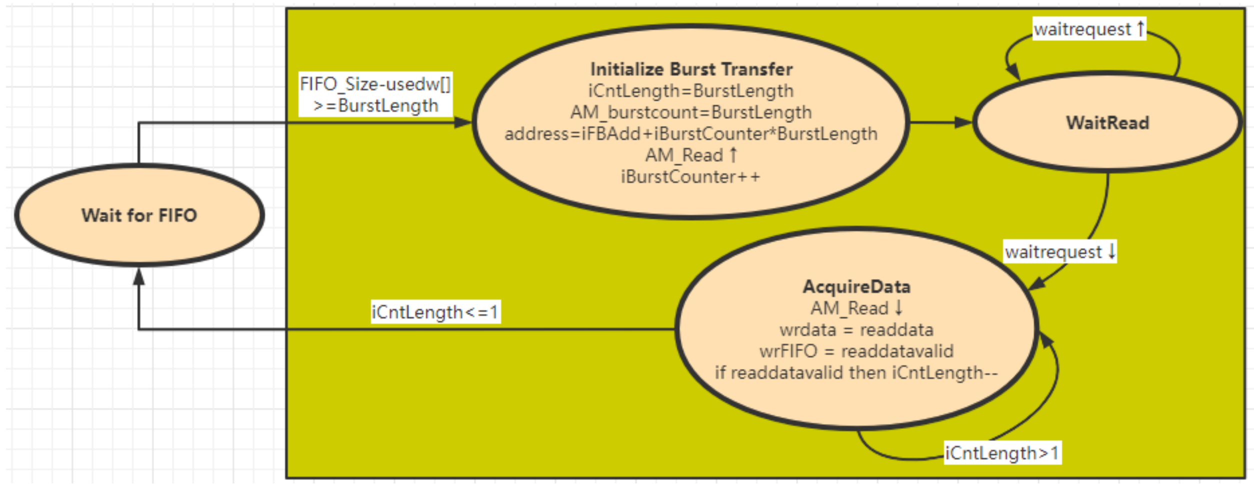
The following waveform shows the timing of the DMA Master:



DMA Master State Machine



Sub-State Machine of DMA Burst Transfer



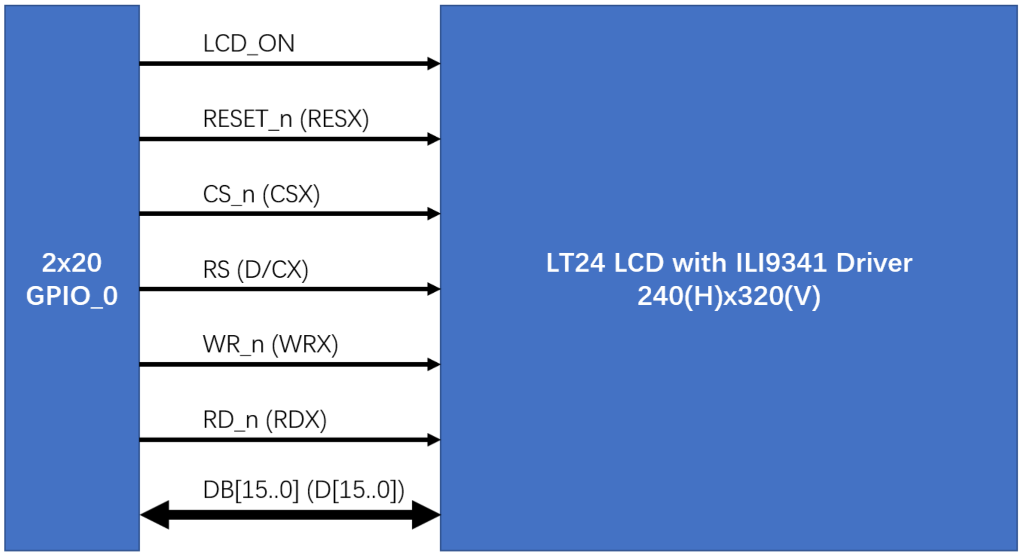
Custom IP register map

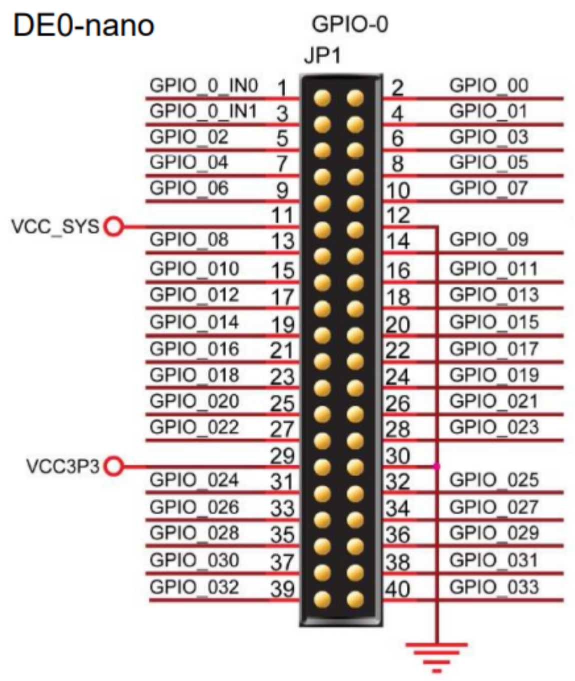
Addr.	Write register	Writedata[31..0]	Read register	Readdata[31..0]	Description
0	LCD_Cmd	→iLCD_Cmd	LCD_Cmd	iLCD_Cmd→	3 bit unsigned integer to start and select the LCD sequencer FSM branch
1	LCD_Data	→iLCD_Data	LCD_Data	LCD_Data→	16 bit Command/Data to write to ILI9341 LCD driver
2	MA_Cmd	→iMA_Cmd	MA_Cmd	iMA_Cmd→	1 bit Command to initiate DMA image read from frame buffer
3	-	Don't care	MA_Status	iStatus→	1 bit Status register, 0: busy. 1: ready for the next frame.
4	MA_FBAddr	→iFBAddr	MA_FBAddr	iFBAddr→	32 bit base address of image frame buffer in memory
5	MA_FBLgt	→iFBLgt	MA_FBLgt	iFBLgt→	32 bit integer for length of image frame buffer. Corresponds to number of words to read from memory
6	-	Don't care	-	0x00	
7	-	Don't care	-	0x00	

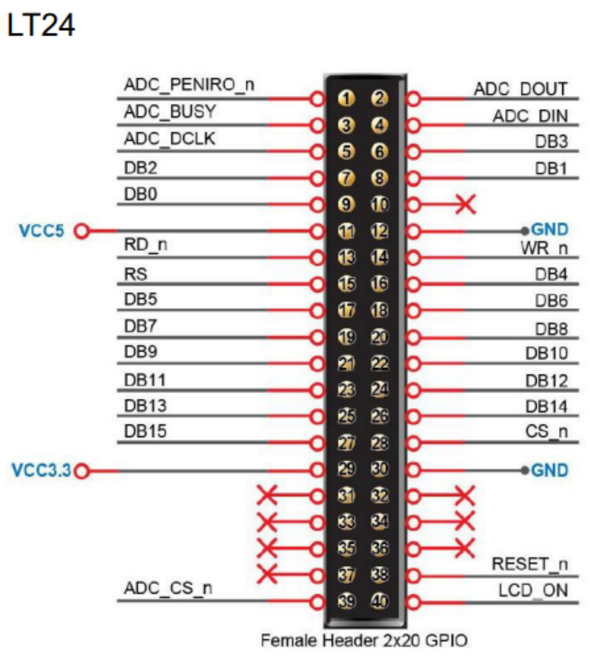
Table 1. Register map of the Avalon Slave Memory Mapped Register Interface

Top-level block diagram Hardware connection

The LT24 camera module will be connected to the GPIO_0 connector of the DE0 Nano FPGA development board. The connections between the custom *LCD Controller* IP and the development board's pins are detailed in the diagram below:

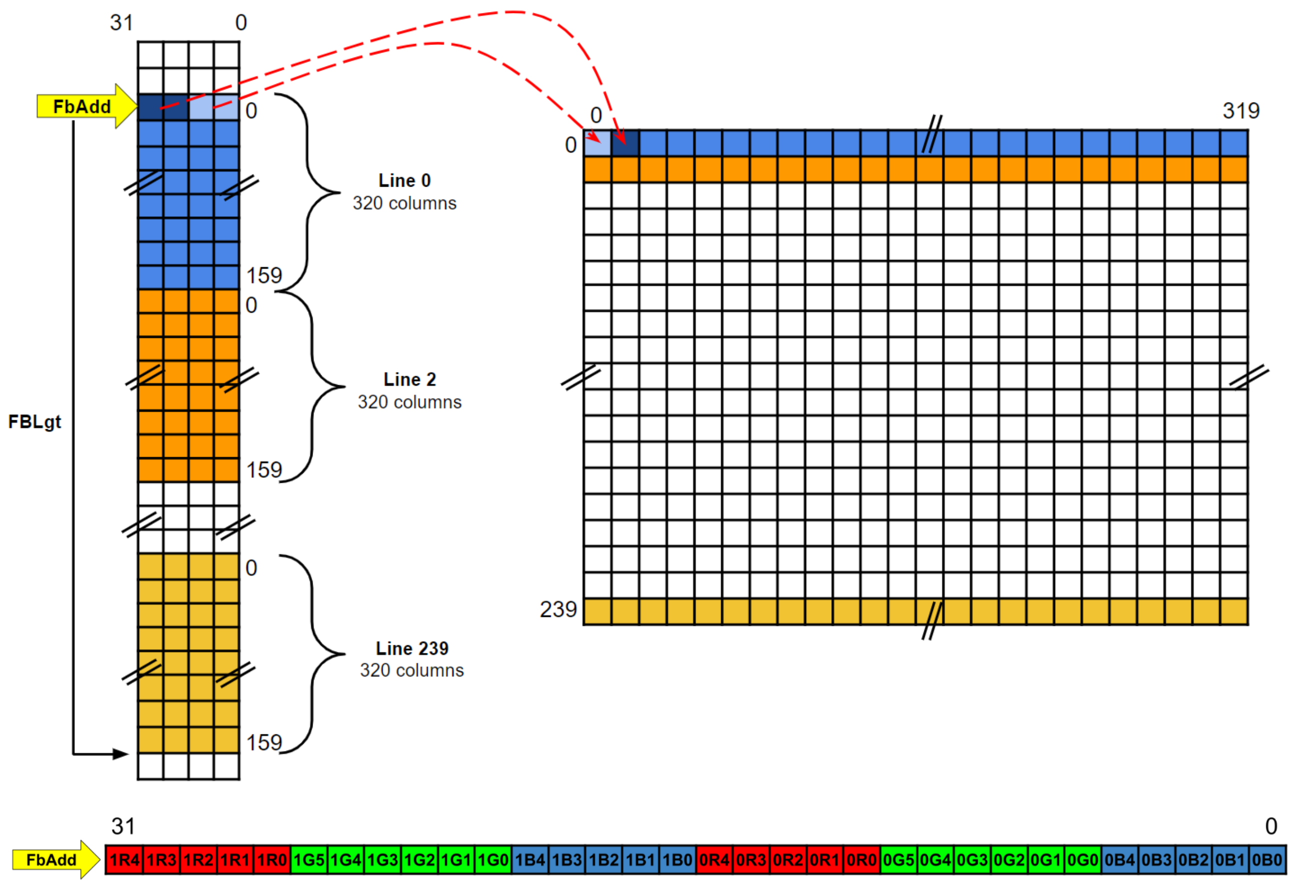






Memory Organization

The memory organization paradigm must be agreed upon with the team implementing the complementary camera DMA to ensure interoperability of the two-subsystems.
The memory will be organized in a 16-bit, 5-6-5 RGB format. Two pixels will be stored as doublets in one 32b word. The image frame buffer organization as well as the organization of one word (example word shown is the case of pixels 0 & 1 stored in the memory address pointed to by FBAdd).


31
0
FbAdd
FBLgt
0
159
Line 0
320 columns
0
159
Line 2
320 columns
0
159
Line 239
320 columns
0
319
0
239
31
0
FbAdd
1R4 1R3 1R2 1R1 1R0 1G5 1G4 1G3 1G2 1G1 1G0 1B4 1B3 1B2 1B1 1B0 0R4 0R3 0R2 0R1 0R0 0G5 0G4 0G3 0G2 0G1 0G0 0B4 0B3 0B2 0B1 0B0

LT24 Control

LT24 Register Initialisation

At the startup of LT24 LCD Display an initialization is needed to set up the parameters of the device. The initialization is done by writing the command address or data to LCD_Data Register and appropriately setting LCD_Cmd Register. The written command will be automatically handled by the Finite State Machine of LCD controller and sent to the LT24 chip respecting the protocol and time constraints. The LCD_ON signal will be permanently routed to 3.3V.
The initialization of the LT24 is done as shown in the steps below:

1. Set-Reset-Set the Hardware Reset signal RESX to the LT24 module.
2. Exit Sleep mode by writing to 0x0011 Command address without data.
3. Registers shown in the table below related to power, gamma are configured to the recommended value according to the lecture slides.
4. The EC (end column) and EP (end page) values in ILI9341 registers 0x2A and 0x2B are set to 319 and 239 respectively.
5. The MACTR register (0x36) is set to ‘X-Y Exchange X Mirror’ mode.

Registers	Address
Power control	0xC0, 0xCF (Level 1), 0xC1 (Level 2)
Power on sequence control	0xED
Driver timing control	0xE8
Power control	0xCB
Pump ratio control	0xF7
Display function control	0xB6
VCM control	0xC5, 0xC7
3-Gamma control	0xF2
Gamma-Set	0x26
Positive/Negative Gamma Correction	0xE0, 0xE1

Table 2. Register Initialisation with default values

6. Other Registers are set according to our specific design needs.

Registers	Address	Details
Pixel Format Register	0x3A	MCU Mode I for 5-6-5 16bit-RGB parallel transfer
Interface Control	0xF6	Internal-clock mode
Memory access control Registers	0x36	Mirror the image with left-top pixel as origin
Column and Page Address Register	0x2A, 0x2B	Changing the image size from 240*320 to 320*240
Frame rate normal mode control	0xCB	70Hz according to our writing speed

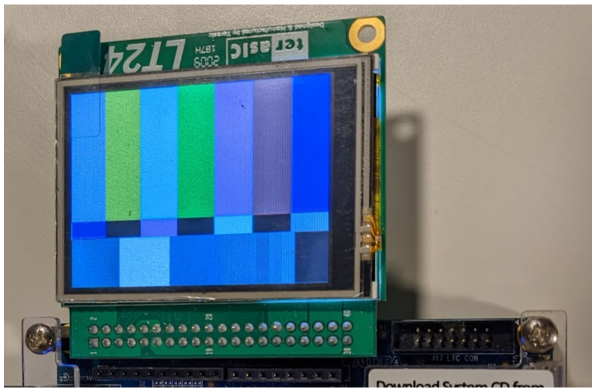
Table 2. Register Initialisation with default values

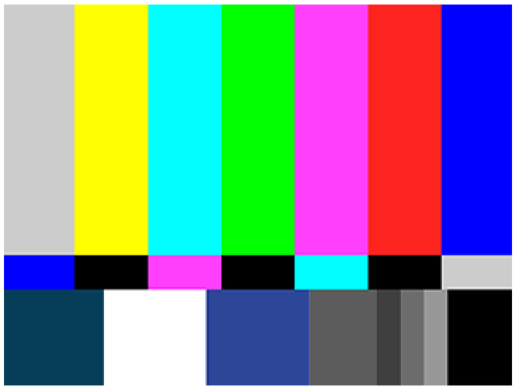
LT24 Data Transfer

After setting up the Registers, the display would be turned on by writing to Display On (0x29). Frame data can be written to the memory after setting the Memory Write Register (0x2C). The chip-select for the ILI9341 is held low continuously, as stated in the datasheet the device will still operate correctly in parallel interface mode.

Conclusion

The design specified in Lab 3 was implemented in hardware on a DE0_Nano Intel FPGA development board with only minor changes and simplifications. After some troubleshooting, the LCD controller subsystem can successfully read an image in the defined format from a specified location in memory and display the contents on the LCD. The display for 1 image takes 12ms, which allows a frame rate of 81Hz.





References

1. Terasic, DE0-Nano-SoC_User_manual https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-5804152209-de0-user-manual.pdf
2. Terasic, LT24 Datasheet. https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=892&FID=527f33a451f2c9a404934446366f5342
3. Terasic, LT24 DE-10 Demonstration https://www.terasic.com.tw/cgi-bin/page/redirect.pl?Url=http%3A%2F%2Fdownload.terasic.com%2Fdownloads%2Fcd-rom%2Fde10-nano%2Fde10-nano_db_demo%2FLT24.zip
4. ILI9341 Datasheet: https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf