TFT-LCD Display + Camera

Design of a TFT Controller
& Camera Controller
For FPGA

Avalon Master/Slave device

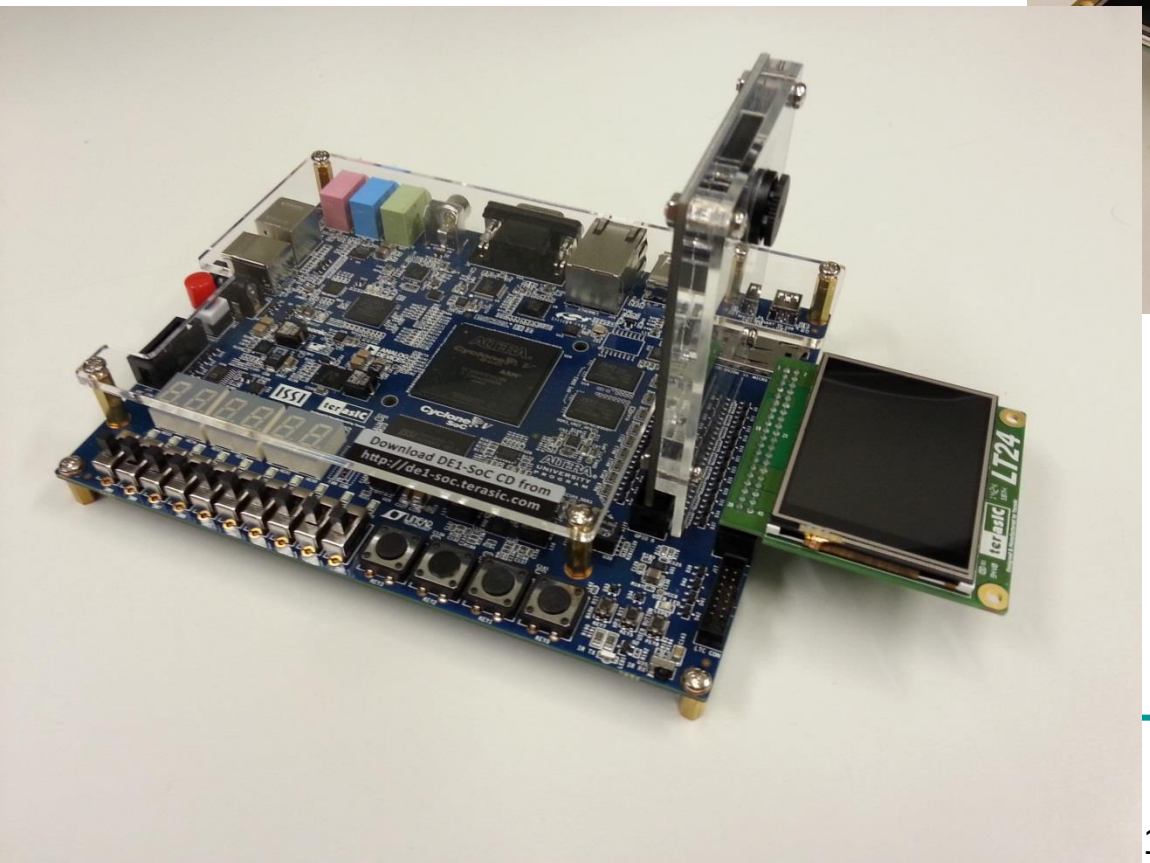
rene.beuchat@epfl.ch
rene.beuchat@hesge.ch

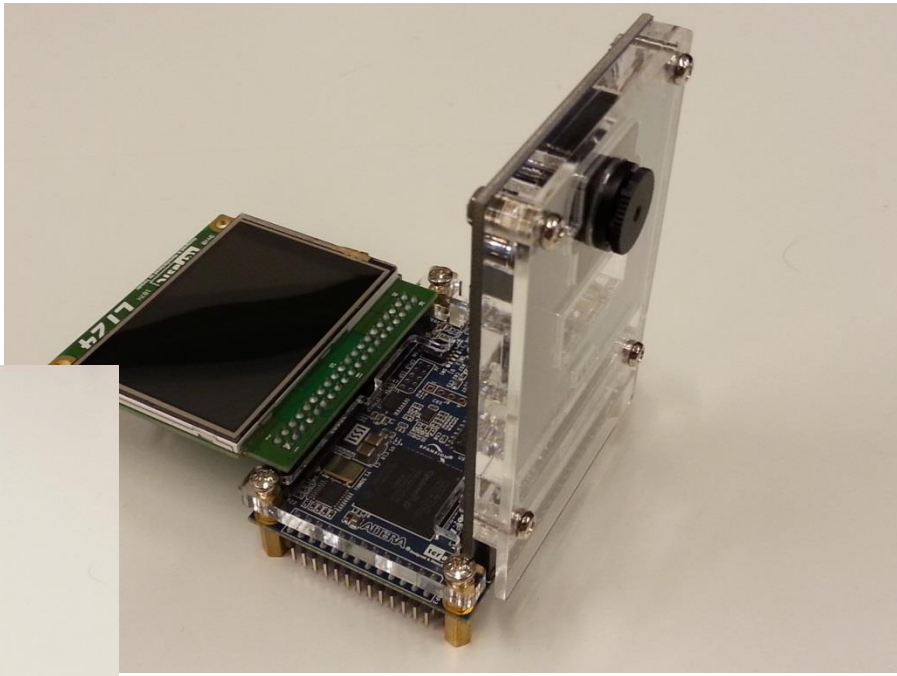
Goals

- The goal is to understand how to design a master unit for a System on FPGA
- 2 masters units have to be designed:
  - a LCD controller
  - a camera controller
- A system with a NIOSII on FPGA has to be build
- A software to control the display and the camera acquisition

Camera (TRDB-D5M) & LCD (LT24)

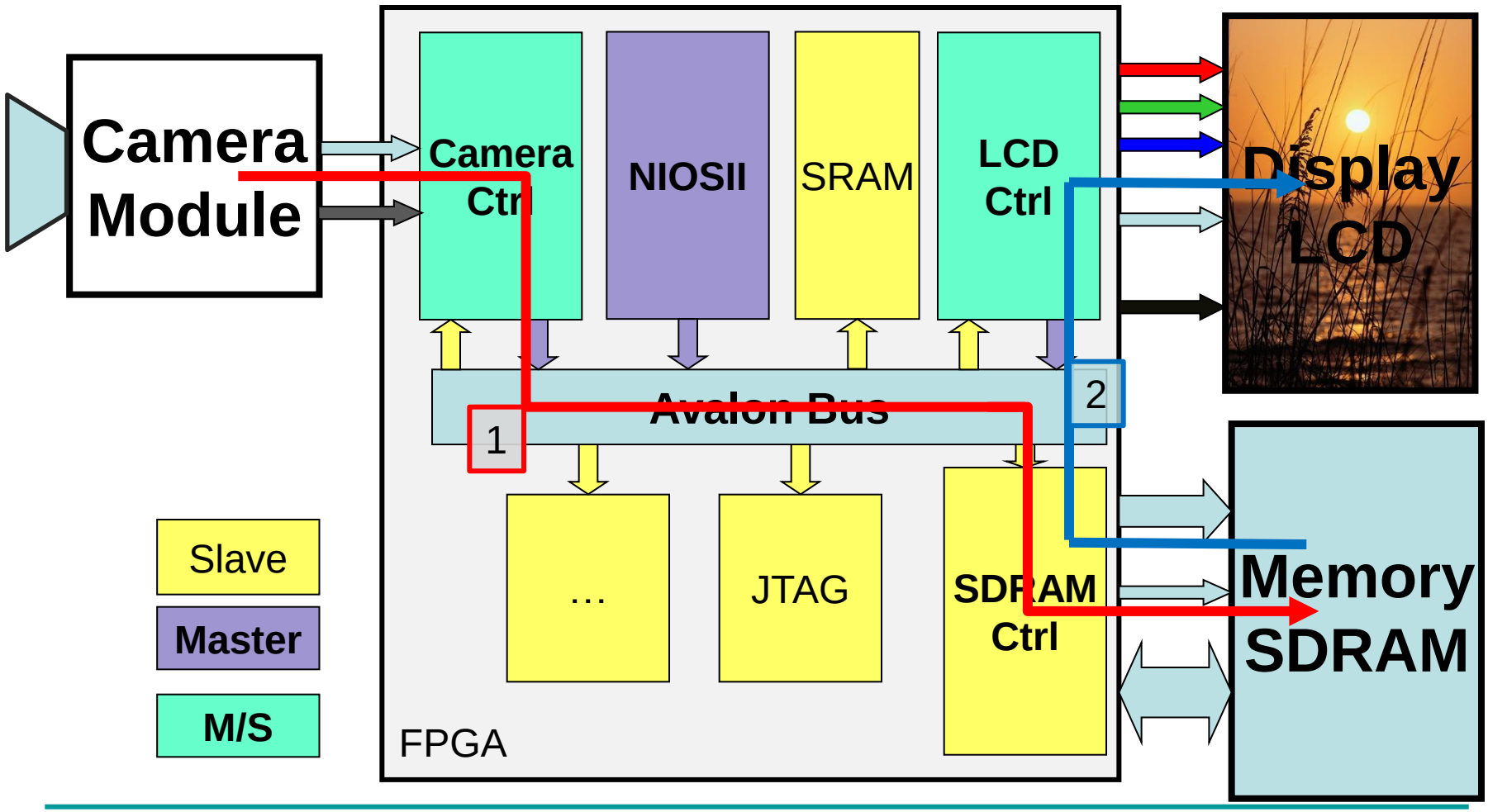
With DE1-SOC





With DE0-nano

Video Controllers in FPGA, General architecture

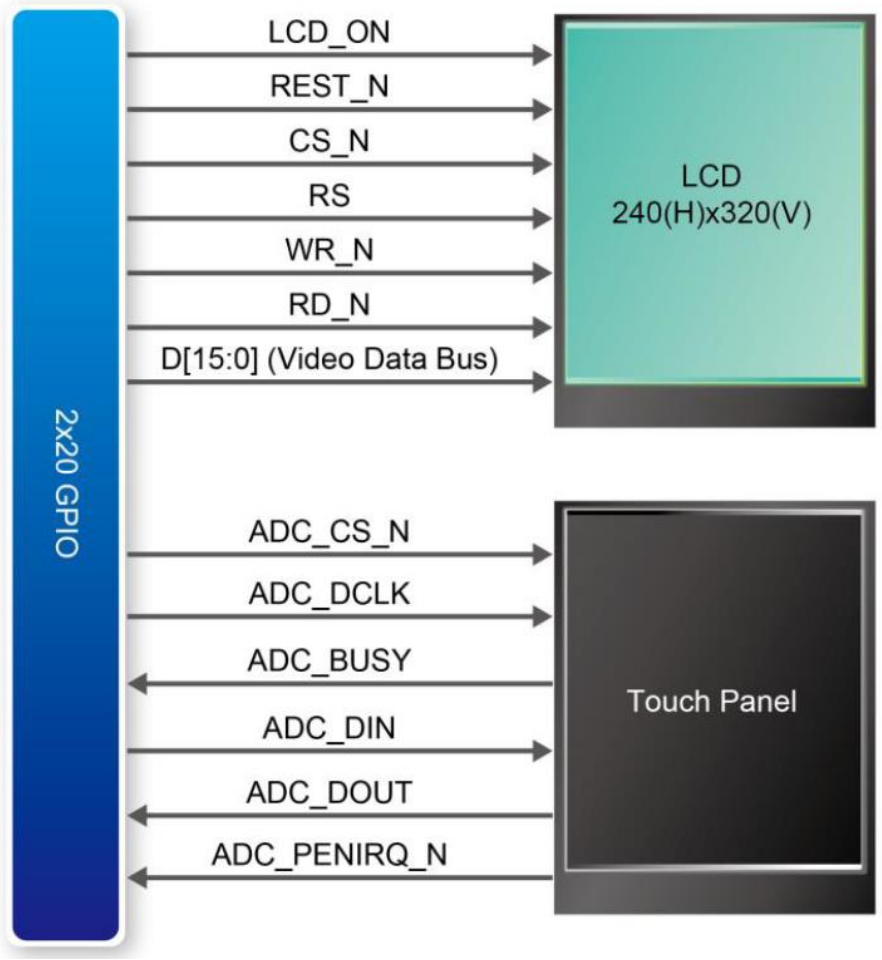


LCD (LT24) specifications

- In this example, the LT24 module from terasic is used (www.terasic.com), characteristics:
  - 240 (H) x 320 (V) RGB 5-6-5 bits/color
  - Touch Panel, resistive technology
  - Small LCD extension on 40 pins connector
  - Simple access to the module as a programmable interface
  - Main controller ILI9341 (Ilitek)

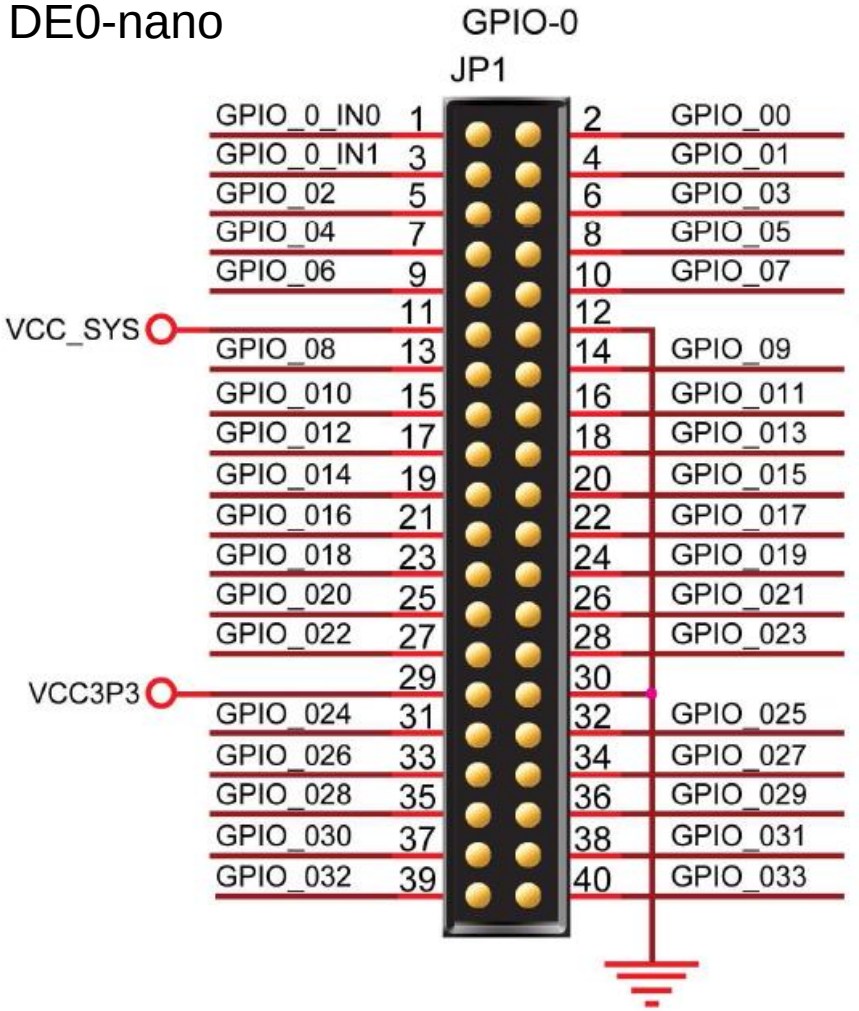
LT24 for DE0-nano & DE1-SOC

- Interface signals:
  - LCD_ON
  - Reset_n
  - CS_n
  - RS (reg. Sel. C/nD)
  - Wr_n
  - Rd_n
  - D[15..0]

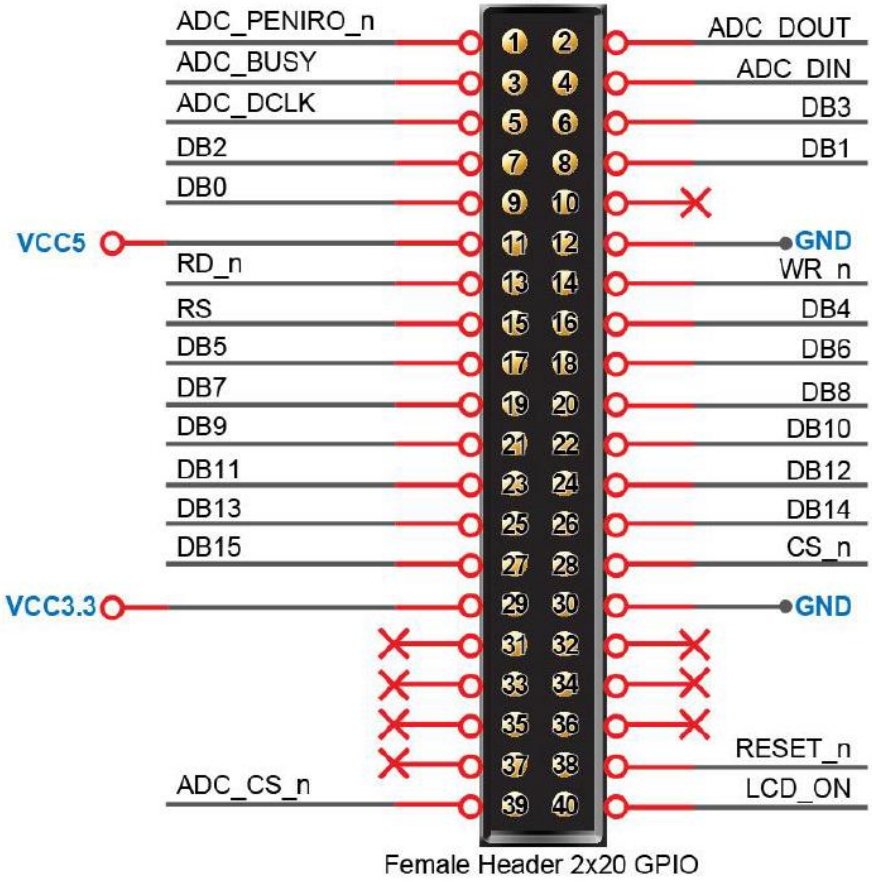


2x20 pins connection

DE0-nano

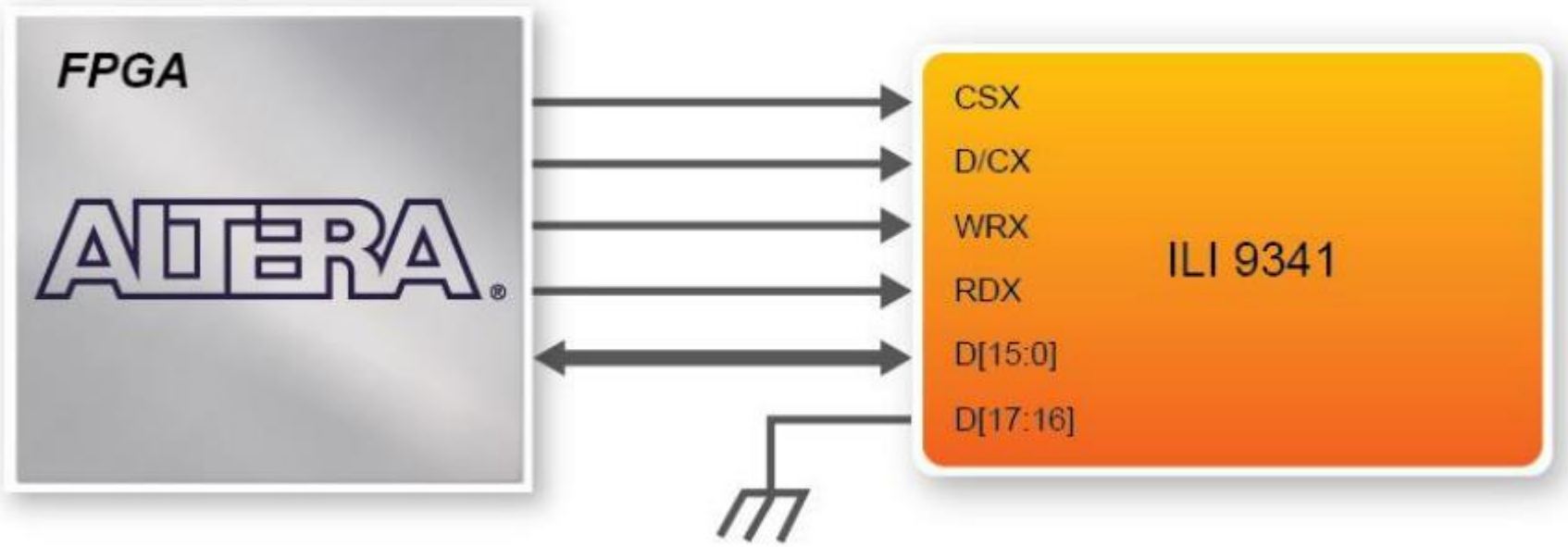


LT24

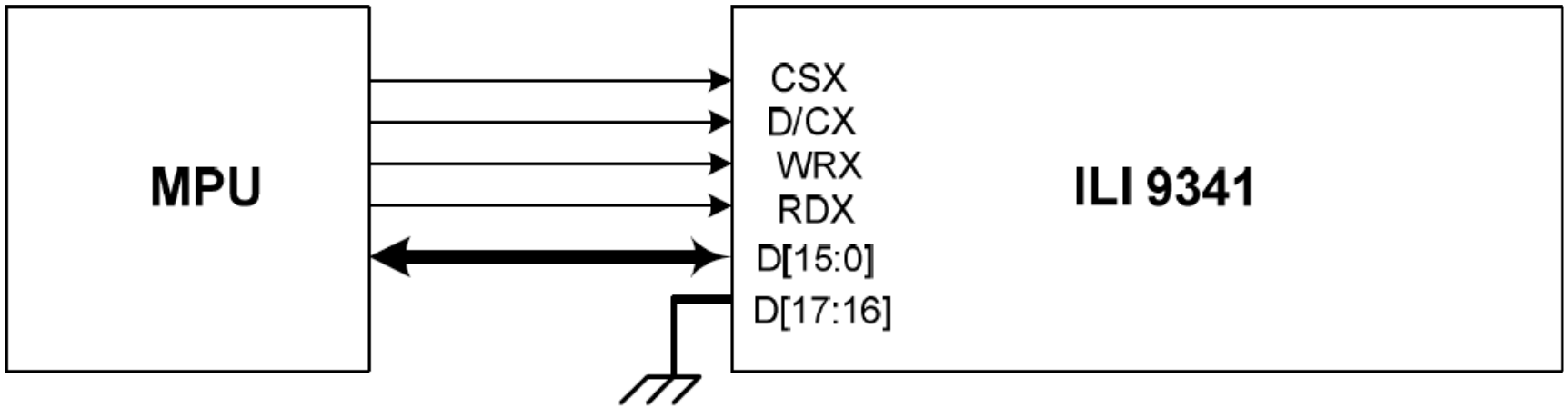


FPGA - LT24 ILI9441 connection

- 65’536 colors 5-6-5 RGB
- 8080-system I, 16 bits interface
- Ref: §7.6.5 ILITEK ILI9341
- http://www.newhavendisplay.com/app_notes/ILI9341.pdf



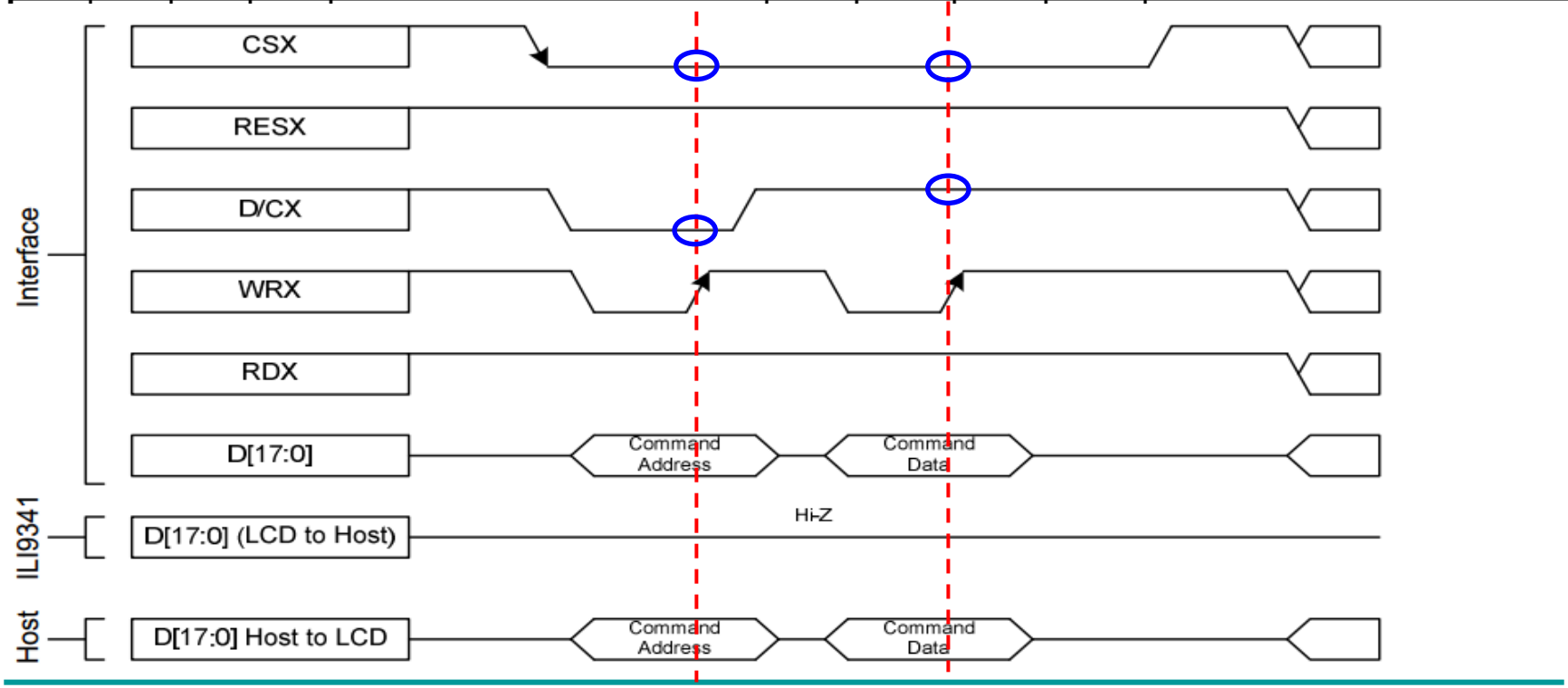
LT24: Bus interface



- CSx : CS_n Chip Select
- D/Cx: D/Cn Data / Command_n
- WRx: WR_n Write Access
- RDx: RD_n Read Access

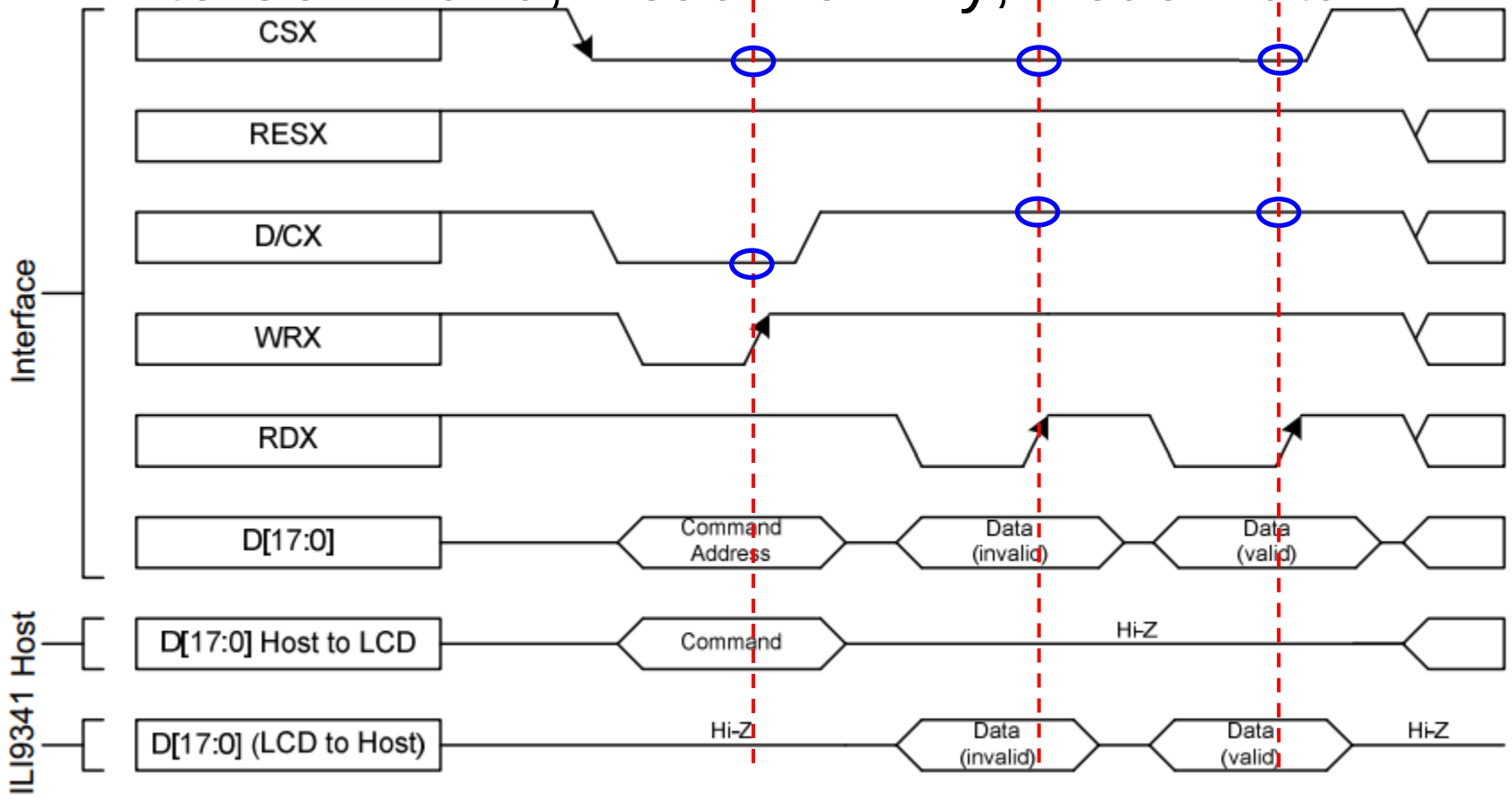
LT24 Timing Diagram Connection, Write Command/Data

IM3	IM2	IM1	IM0	MCU-Interface Mode	CSX	WRX	RDX	D/CX	Function
0	0	0	1	8080 MCU 16-bit bus interface I	"L"	↑	"H"	"L"	Write command code.
					"L"	"H"	↑	"H"	Read internal status.
					"L"	↑	"H"	"H"	Write parameter or display data.
					"L"	"H"	↑	"H"	Reads parameter or display data.



LT24 Timing Diagram Connection, Read Command/Data

- Write Command, Read Dummy, Read Data



LT24 Timing Access

- The interface used is the old fashion 8080-system 16 bits wide.
- The transfers are asynchronous (no clock related) between the FPGA and the module
- The data are sent line by line by burst of 240 pixels continuously

LCD data transfers format, 16 bits data mode

For each line of LCD:

5 bits Blue, **6** bits Green, **5** bits Red, ***IM[3:0]***: "***0001***"

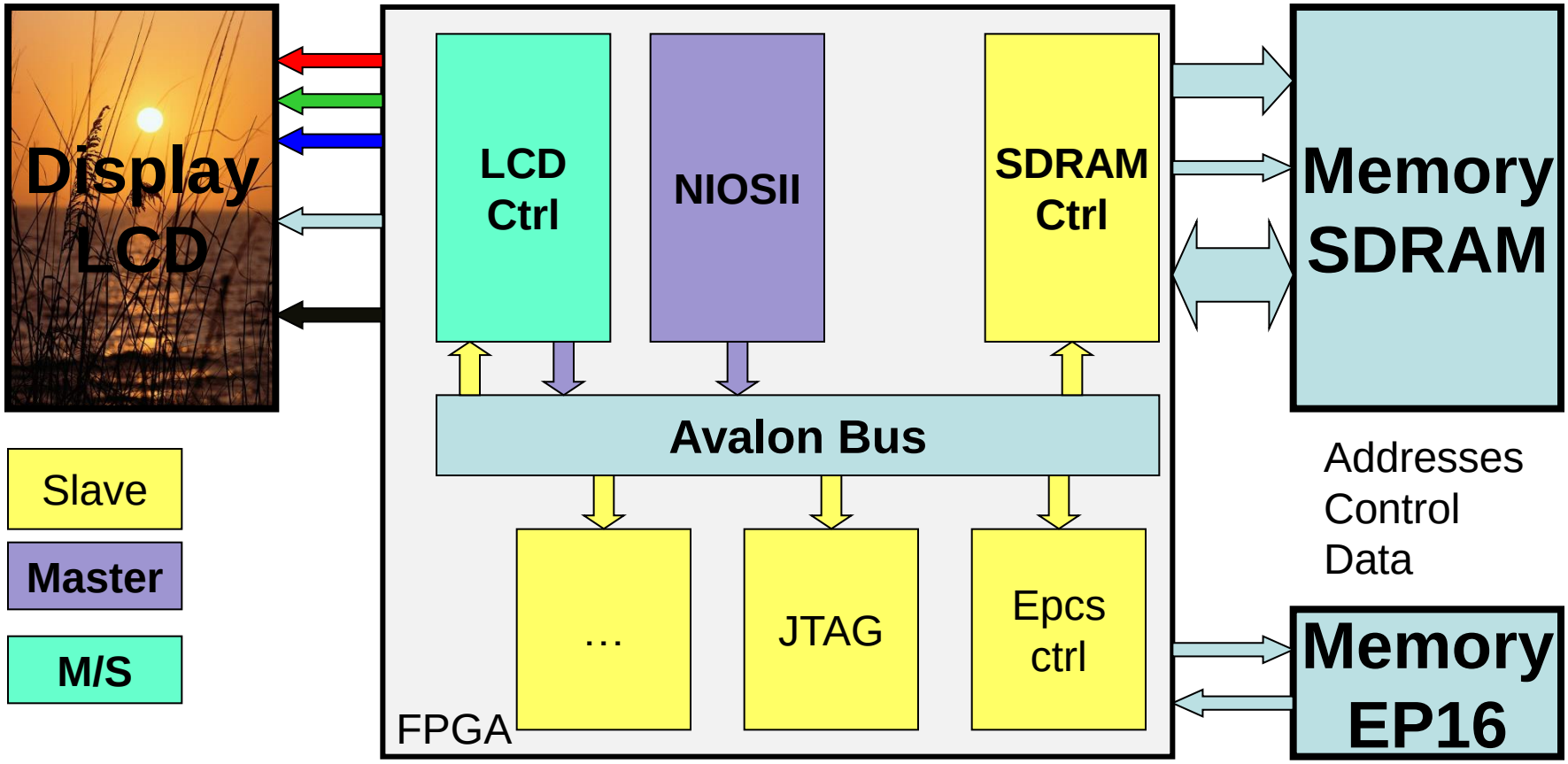
Register ***3A***h: "***xxxx101***", ***MDT[1..0]***: "***00***"

Count	0	1	2	3	…	238	239	240
D/CX	0	1	1	1	…	1	1	1
D15		0R4	1R4	2R4	…	237R4	238R4	239R4
D14		0R3	1R3	2R3	…	237R3	238R3	239R3
D13		0R2	1R2	2R2	…	237R2	238R2	239R2
D12		0R1	1R1	2R1	…	237R1	238R1	239R1
D11		0R0	1R0	2R0	…	237R0	238R0	239R0
D10		0G5	1G5	2G5	…	237G5	238G5	239G5
D9		0G4	1G4	2G4	…	237G4	238G4	239G4
D8		0G3	1G3	2G3	…	237G3	238G3	239G3
D7	C7	0G2	1G2	2G2	…	237G2	238G2	239G2
D6	C6	0G1	1G1	2G1	…	237G1	238G1	239G1
D5	C5	0G0	1G0	2G0	…	237G0	238G0	239G0
D4	C4	0B4	1B4	2B4	…	237B4	238B4	239B4
D3	C3	0B3	1B3	2B3	…	237B3	238B3	239B3
D2	C2	0B2	1B2	2B2	…	237B2	238B2	239B2
D1	C1	0B1	1B1	2B1	…	237B1	238B1	239B1
D0	C0	0B0	1B0	2B0	…	237B0	238B0	239B0

LCD Controller in FPGA

- To control the LCD, a Controller is necessary
- For FPGA4U and in a system based on a NIOSII processor on **Avalon Bus** a specific interface is to be designed.
- It needs to be an:
  - Avalon slave to program it
  - Avalon Master to take display information in memory

LCD Controller in FPGA, General architecture



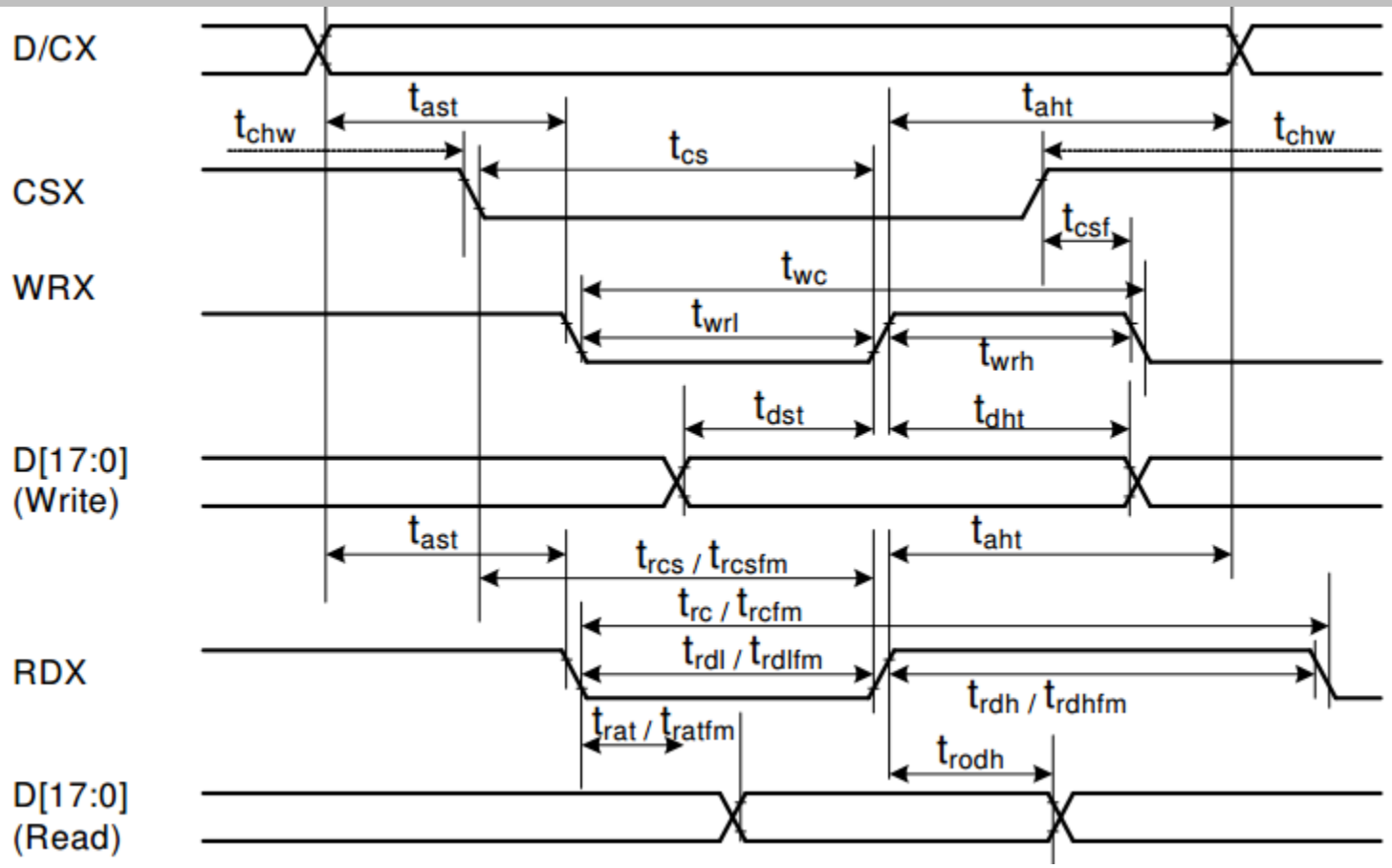
Avalon Slave, registers

- To configure the LCD controller, a number of registers needs to be defined and implemented.
- They need to be programmed by the processor at startup.
- They have default values.
- A command register start the controller and control the power converter and LCD enable.

Reset Initialization §15.1 p.228

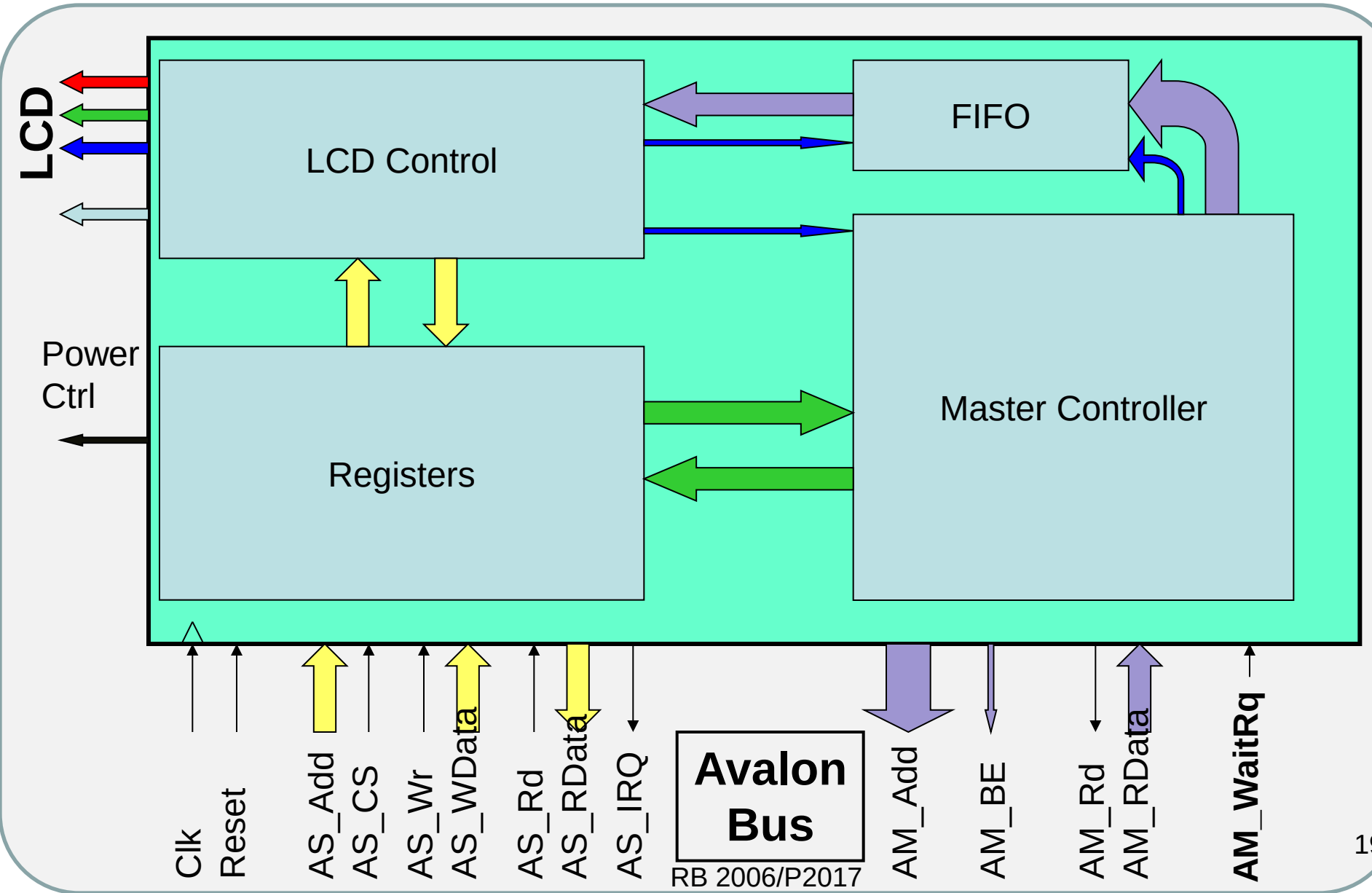
	After Powered ON	After Hardware Reset	After Software Reset
Frame Memory	Random	Repair data	No Change
Sleep	In	In	In
Display Mode	Normal	Normal	Normal
Display	Off	Off	Off
Idle	Off	Off	Off
Column Start Address	0000 h	0000 h	0000 h
Column End Address	00EF h	00EF h	If MADCTL's B5=0:00EF h If MADCTL's B5=1:013F h
Page Start Address	0000 h	0000 h	0000 h
Page End Address	013F h	013F h	If MADCTL's B5 = 0:013F h If MADCTL's B5=1:00EF h
Gamma Setting	GC0	GC0	GC0
Partial Area Start	0000 h	0000 h	0000 h
Partial Area End	013F h	013F h	013F h
Memory Data Access Control	00 h	00 h	No Change
RDDPM	08 h	08 h	08 h
RDDMADCTL	00 h	00 h	No Change
RDDCOLMOD	06 h	06 h	06 h
RDDIM	00 h	00 h	00 h
RDDSM	00 h	00 h	00 h
RDDSDR	00 h	00 h	00 h
TE Output Line	Off	Off	Off
TE Line Mode	Mode 1 (Note 3)	Mode 1 (Note 3)	Mode 1 (Note 3)

Detailed timing diagram



WRX	twc	Write cycle	66	-	ns
	twrh	Write Control pulse H duration	15	-	ns
	twrl	Write Control pulse L duration	15	-	ns

LCD Controller architecture



RB 2006/P2017

LCD Control architecture

- The LCD Controller can be separate in 4 main blocks:
  - **Avalon Slave** to access registers
  - **Avalon Master** to read memory
  - A **FIFO** to receive data from Avalon Master and provide them to LCD control part

    The FIFO guaranties a **continuous** flow from memory the LCD, as memory is shared by multiples masters
  - A **LCD Control** to sequence the LCD signals and send RGB data

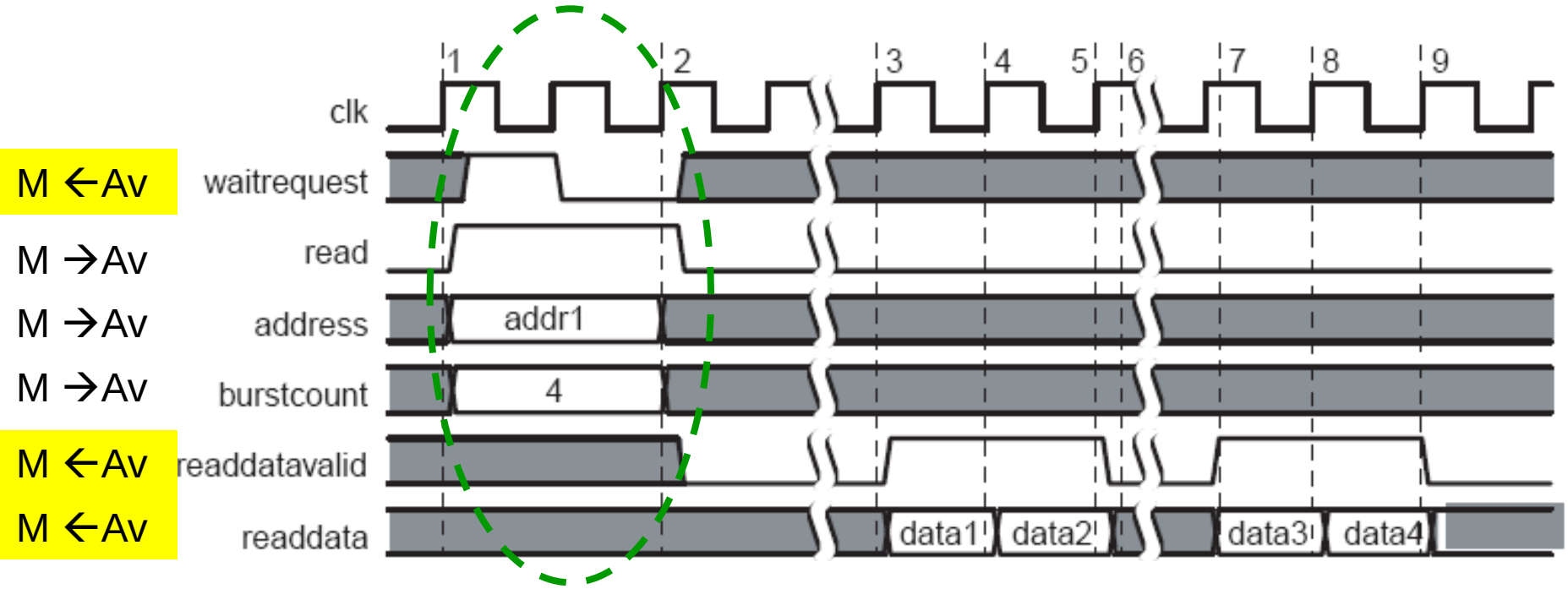
LCD Controller : Avalon Slave - Registers

- The Avalon slave interface is used to:
  - receive and send registers contents from/to a Master Processor (ie: the NIOSII)
  - Send an IRQ at end of frame if enabled, this allows the processor to provide a new buffer address at start of display frame
  - Dispatch the registers content to the modules:
    - Buffer information to master interface
    - Display parameters to LCD Control

LCD Controller : Avalon Master

- The Avalon Master interface is used to:
  1. Receive the Buffer information from Avalon slave part as :
     - Start address of the buffer
     - Length of buffer to read
  2. Synchronize the start of reading with LCD VSync from LCD Ctrl
  3. Read display data from memory through Avalon read accesses
  4. Send the Read data to a synchronization FIFO

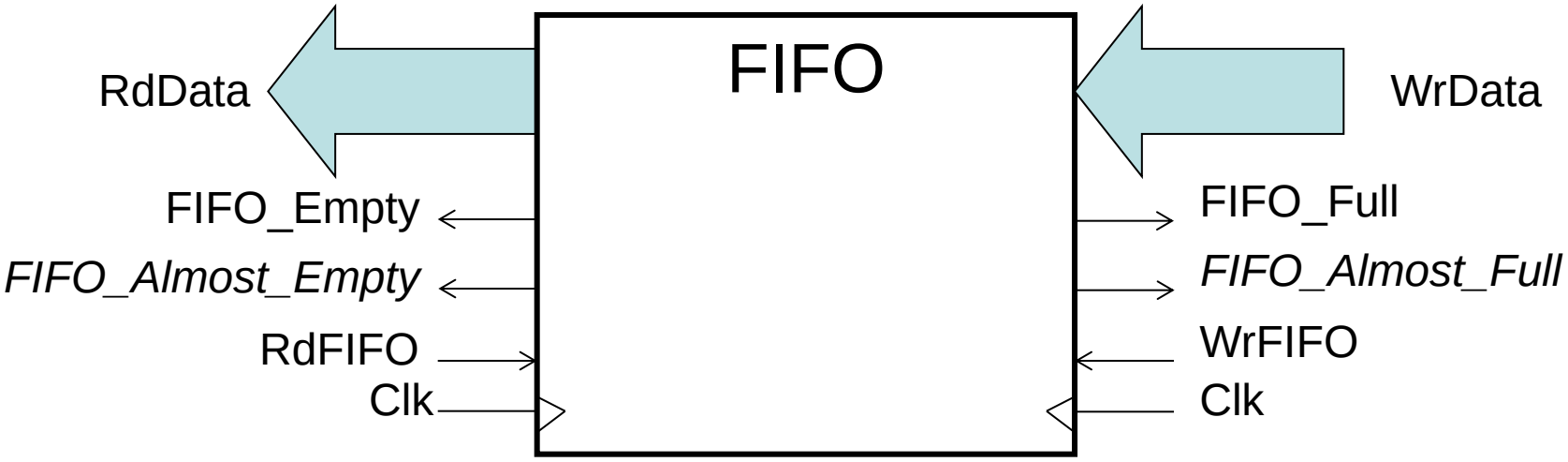
Avalon burst Read
As a Master Unit



Address and BurstCount available for the first cycle only
Read signal only for the first cycle
*The number of burstcount **ReadDataValid** needs to be generated*
The master could start a new transfer in 2

Interface Camera, FIFO

- A FIFO (First In First Out) Memory allows the synchronization between a data producer and a data reader with asynchronous transfers rate.
- Available and configurable with QuartusII MegaWizard



FIFO

- The Output ***FIFO_Almost_Empty*** allows to know if they are still ***xx*** free positions in the FIFO and ***FIFO_Almost_Full*** allows to know if they are already ***xx*** filled positions in the FIFO
- ***xx*** programmable when FIFO is generated with MegaWizard
- Thus it is possible to know that at least Burst Mode Size data are available for read access master transfers to write in FIFO
- The size off the FIFO and …_Almost_... are defined at generation time of the FIFO through MegaWizard

Some ILI9341 Commands

Regulative Command Set													
Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
No Operation	0	1	↑	XX	0	0	0	0	0	0	0	0	00h
Software Reset	0	1	↑	XX	0	0	0	0	0	0	0	1	01h
Read Display Identification Information	0	1	↑	XX	0	0	0	0	0	1	0	0	04h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	ID1 [7:0]								XX
	1	↑	1	XX	ID2 [7:0]								XX
	1	↑	1	XX	ID3 [7:0]								XX
Read Display Status	0	1	↑	XX	0	0	0	0	1	0	0	1	09h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [31:25]							X	00
	1	↑	1	XX	X	D [22:20]			D [19:16]				61
	1	↑	1	XX	X	X	X	X	X	D [10:8]			00
	1	↑	1	XX	D [7:5]			X	X	X	X	X	00
Read Display Power Mode	0	1	↑	XX	0	0	0	0	1	0	1	0	0Ah
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	08
Read Display MADCTL	0	1	↑	XX	0	0	0	0	1	0	1	1	0Bh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	00
Read Display Pixel Format	0	1	↑	XX	0	0	0	0	1	1	0	0	0Ch
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	RIM	DPI [2:0]			X	DBI [2:0]			06
Read Display Image Format	0	1	↑	XX	0	0	0	0	1	1	0	1	0Dh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	D [2:0]			00
Read Display Signal Mode	0	1	↑	XX	0	0	0	0	1	1	1	0	0Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	00
Read Display Self-Diagnostic Result	0	1	↑	XX	0	0	0	0	1	1	1	1	0Fh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:6]		X	X	X	X	X	X	00
Enter Sleep Mode	0	1	↑	XX	0	0	0	1	0	0	0	0	10h
Sleep OUT	0	1	↑	XX	0	0	0	1	0	0	0	1	11h
Partial Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	0	12h
Normal Display Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	1	13h
Display Inversion OFF	0	1	↑	XX	0	0	1	0	0	0	0	0	20h
Display Inversion ON	0	1	↑	XX	0	0	1	0	0	0	0	1	21h
Gamma Set	0	1	↑	XX	0	0	1	0	0	1	1	0	26h
	1	1	↑	XX	GC [7:0]								01
Display OFF	0	1	↑	XX	0	0	1	0	1	0	0	0	28h
Display ON	0	1	↑	XX	0	0	1	0	1	0	0	1	29h
Column Address Set	0	1	↑	XX	0	0	1	0	1	0	1	0	2Ah
	1	1	↑	XX	SC [15:8]								XX
	1	1	↑	XX	SC [7:0]								XX
	1	1	↑	XX	EC [15:8]								XX
	1	1	↑	XX	EC [7:0]								XX
Page Address Set	0	1	↑	XX	0	0	1	0	1	0	1	1	2Bh
	1	1	↑	XX	SP [15:8]								XX
	1	1	↑	XX	SP [7:0]								XX
	1	1	↑	XX	EP [15:8]								XX
	1	1	↑	XX	EP [7:0]								XX

Memory Write	0	1	↑	XX	0	0	1	0	1	1	0	0	2Ch
	1	1	↑	D [17:0]									XX
Color SET	0	1	↑	XX	0	0	1	0	1	1	0	1	2Dh
	1	↑	1	XX			R00 [5:0]						XX
	1	↑	1	XX			Rnn [5:0]						XX
	1	↑	1	XX			R31 [5:0]						XX
	1	↑	1	XX			G00 [5:0]						XX
	1	↑	1	XX			Gnn [5:0]						XX
	1	↑	1	XX			G64 [5:0]						XX
	1	↑	1	XX			B00 [5:0]						XX
	1	↑	1	XX			Bnn [5:0]						XX
	1	↑	1	XX			B31 [5:0]						XX
Memory Read	0	1	↑	XX	0	0	1	0	1	1	1	0	2Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	D [17:0]									XX
Partial Area	0	1	↑	XX	0	0	1	1	0	0	0	0	30h
	1	1	↑	XX	SR [15:8]								00
	1	1	↑	XX	SR [7:0]								00
	1	1	↑	XX	ER [15:8]								01
	1	1	↑	XX	ER [7:0]								3F
Vertical Scrolling Definition	0	1	↑	XX	0	0	1	1	0	0	1	1	33h
	1	1	↑	XX	TFA [15:8]								00
	1	1	↑	XX	TFA [7:0]								00
	1	1	↑	XX	VSA [15:8]								01
	1	1	↑	XX	VSA [7:0]								40
	1	1	↑	XX	BFA [15:8]								00
	1	1	↑	XX	BFA [7:0]								00
Tearing Effect Line OFF	0	1	↑	XX	0	0	1	1	0	1	0	0	34h
Tearing Effect Line ON	0	1	↑	XX	0	0	1	1	0	1	0	1	35h
	1	1	↑	XX	X	X	X	X	X	X	X	M	00
Memory Access Control	0	1	↑	XX	0	0	1	1	0	1	1	0	36h
	1	1	↑	XX	MY	MX	MV	ML	BGR	MH	X	X	00
Vertical Scrolling Start Address	0	1	↑	XX	0	0	1	1	0	1	1	1	37h
	1	1	↑	XX	VSP [15:8]								00
	1	1	↑	XX	VSP [7:0]								00
Idle Mode OFF	0	1	↑	XX	0	0	1	1	1	0	0	0	38h
Idle Mode ON	0	1	↑	XX	0	0	1	1	1	0	0	1	39h
Pixel Format Set	0	1	↑	XX	0	0	1	1	1	0	1	0	3Ah
	1	1	↑	XX	X	DPI [2:0]			X	DBI [2:0]			66
Write Memory Continue	0	1	↑	XX	0	0	1	1	1	1	0	0	3Ch
	1	1	↑	D [17:0]									XX
Read Memory Continue	0	1	↑	XX	0	0	1	1	1	1	1	0	3Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	D [17:0]									XX
Set Tear Scanline	0	1	↑	XX	0	1	0	0	0	1	0	0	44h
	1	1	↑	XX	X	X	X	X	X	X	X	STS [8]	00
	1	1	↑	XX	STS [7:0]								00
Get Scanline	0	1	↑	XX	0	1	0	0	0	1	0	1	45h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X	GTS [9:8]		00
	1	↑	1	XX	GTS [7:0]								00
Write Display Brightness	0	1	↑	XX	0	1	0	1	0	0	0	1	51h
	1	1	↑	XX	DBV [7:0]								00

Read Display Brightness	0	1	↑	XX	0	1	0	1	0	0	1	0	52h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	DBV [7:0]								00
Write CTRL Display	0	1	↑	XX	0	1	0	1	0	0	1	1	53h
	1	1	↑	XX	X	X	BCTRL	X	DD	BL	X	X	00
Read CTRL Display	0	1	↑	XX	0	1	0	1	0	1	0	0	54h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	BCTRL	X	DD	BL	X	X	00
Write Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	0	1	55h
	1	1	↑	XX	X	X	X	X	X	X	C [1:0]		00
Read Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	1	0	56h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X	C [1:0]		00
Write CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	1	1	1	0	5Eh
	1	1	↑	XX	CMB [7:0]								00
Read CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	0	1	1	1	5Fh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	CMB [7:0]								00
Read ID1	0	1	↑	XX	1	1	0	1	1	0	1	0	DAh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	Module's Manufacture [7:0]								XX
Read ID2	0	1	↑	XX	1	1	0	1	1	0	1	1	DBh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver Version [7:0]								XX
Read ID3	0	1	↑	XX	1	1	0	1	1	1	0	0	DCh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver ID [7:0]								XX

Extended Command Set												
Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
RGB Interface Signal Control	0	1	↑	XX	1	0	1	1	0	0	0	0	B0h
	1	1	↑	XX	ByPass_MODE	RCM [1:0]		X	VSPL	HSPL	DPL	EPL	40
Frame Control (In Normal Mode)	0	1	↑	XX	1	0	1	1	0	0	0	1	B1h
	1	1	↑	XX	X	X	X	X	X	X	DIVA [1:0]		00
	1	1	↑	XX	X	X	X	RTNA [4:0]					1B
Frame Control (In Idle Mode)	0	1	↑	XX	1	0	1	1	0	0	1	0	B2h
	1	1	↑	XX	X	X	X	X	X	X	DIVB [1:0]		00
	1	1	↑	XX	X	X	X	RTNB [4:0]					1B
Frame Control (In Partial Mode)	0	1	↑	XX	1	0	1	1	0	0	1	1	B3h
	1	1	↑	XX	X	X	X	X	X	X	DIVC [1:0]		00
	1	1	↑	XX	X	X	X	RTNC [4:0]					1B
Display Inversion Control	0	1	↑	XX	1	0	1	1	0	1	0	0	B4h
	1	1	↑	XX	X	X	X	X	X	NLA	NLB	NLC	02
Blanking Porch Control	0	1	↑	XX	1	0	1	1	0	1	0	1	B5h
	1	1	↑	XX	0	VFP [6:0]							02
	1	1	↑	XX	0	VBP [6:0]							02
	1	1	↑	XX	0	0	0	HFP [4:0]					0A
	1	1	↑	XX	0	0	0	HBP [4:0]					14

Display Function Control	0	1	↑	XX	1	0	1	1	0	1	1	0	B6h
	1	1	↑	XX	X	X	X	X	PTG [1:0]		PT [1:0]		0A
	1	1	↑	XX	REV	GS	SS	SM	ISC [3:0]				82
	1	1	↑	XX	X	X	NL [5:0]						27
	1	1	↑	XX	X	X	PCDIV [5:0]						XX
Entry Mode Set	0	1	↑	XX	1	0	1	1	0	1	1	1	B7h
	1	1	↑	XX	X	X	X	X	0	GON	DTE	GAS	07
Backlight Control 1	0	1	↑	XX	1	0	1	1	1	0	0	0	B8h
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	X	X	X	X	TH_UI [3:0]				04
Backlight Control 2	0	1	↑	XX	1	0	1	1	1	0	0	1	B9h
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	TH_MV [3:0]				TH_ST [3:0]				B8
Backlight Control 3	0	1	↑	XX	1	0	1	1	1	0	1	0	BAh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	X	X	X	X	DTH_UI [3:0]				04
Backlight Control 4	0	1	↑	XX	1	0	1	1	1	0	1	1	BBh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	DTH_MV [3:0]				DTH_ST [3:0]				C9
Backlight Control 5	0	1	↑	XX	1	0	1	1	1	1	0	0	BCh
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX
	1	1	↑	XX	DIM2 [3:0]				X	DIM1 [2:0]			44
Backlight Control 7	0	1	↑	XX	1	0	1	1	1	1	1	0	BEh
	1	1	↑	XX	PWM_DIV [7:0]								0F
Backlight Control 8	0	1	↑	XX	1	0	1	1	1	1	1	1	BFh
	1	1	↑	XX	X	X	X	X	X	LEDONR	LEDONPOL	LEDPWMOPL	00
Power Control 1	0	1	↑	XX	1	1	0	0	0	0	0	0	C0h
	1	1	↑	XX	X	X	VRH [5:0]						26
Power Control 2	0	1	↑	XX	1	1	0	0	0	0	0	1	C1h
	1	1	↑	XX	X	X	X	X	X	BT [2:0]			00
VCOM Control 1	0	1	↑	XX	1	1	0	0	0	1	0	1	C5h
	1	1	↑	XX	X	VMH [6:0]							31
	1	1	↑	XX	X	VML [6:0]							3C
VCOM Control 2	0	1	↑	XX	1	1	0	0	0	1	1	1	C7h
	1	1	↑	XX	nVM	VMF [6:0]							C0
NV Memory Write	0	1	↑	XX	1	1	0	1	0	0	0	0	D0h
	1	1	↑	XX	X	X	X	X	X	PGM_ADR [2:0]			00
	1	1	↑	XX	PGM_DATA [7:0]								XX
NV Memory Protection Key	0	1	↑	XX	1	1	0	1	0	0	0	1	D1h
	1	1	↑	XX	KEY [23:16]								55
	1	1	↑	XX	KEY [15:8]								AA
	1	1	↑	XX	KEY [7:0]								66
NV Memory Status Read	0	1	↑	XX	1	1	0	1	0	0	1	0	D2h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	ID2_CNT [2:0]			X	ID1_CNT [2:0]			XX
	1	↑	1	XX	BUSY	VMF_CNT [2:0]			X	ID3_CNT [2:0]			XX

Read ID4	0	↑	1	XX	1	1	0	1	0	0	1	1	D3h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	0	0	0	0	0	0	0	0	00
	1	↑	1	XX	1	0	0	1	0	0	1	1	93
	1	↑	1	XX	0	1	0	0	0	0	0	1	41
Positive Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	0	E0h
	1	1	↑	XX	X	X	X	X	VP0 [3:0]				08
	1	1	↑	XX	X	X	VP1 [5:0]						0E
	1	1	↑	XX	X	X	VP2 [5:0]						12
	1	1	↑	XX	X	X	X	X	VP4 [3:0]				05
	1	1	↑	XX	X	X	X	VP6 [4:0]					03
	1	1	↑	XX	X	X	X	X	VP13 [3:0]				09
	1	1	↑	XX	X	VP20 [6:0]							47
	1	1	↑	XX	VP36 [3:0]				VP27 [3:0]				86
	1	1	↑	XX	X	VP43 [6:0]							2B
	1	1	↑	XX	X	X	X	X	VP50 [3:0]				0B
	1	1	↑	XX	X	X	X	VP57 [4:0]					04
	1	1	↑	XX	X	X	X	X	VP59 [3:0]				00
	1	1	↑	XX	X	X	VP61 [5:0]						00
	1	1	↑	XX	X	X	VP62 [5:0]						00
	1	1	↑	XX	X	X	X	X	VP63 [3:0]				00
Negative Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	1	E1h
	1	1	↑	XX	X	X	X	X	VN0 [3:0]				08
	1	1	↑	XX	X	X	VN1 [5:0]						1A
	1	1	↑	XX	X	X	VN2 [5:0]						20
	1	1	↑	XX	X	X	X	X	VN4 [3:0]				07
	1	1	↑	XX	X	X	X	VN6 [4:0]					0E
	1	1	↑	XX	X	X	X	X	VN13 [3:0]				05
	1	1	↑	XX	X	VN20 [6:0]							3A
	1	1	↑	XX	VN36 [3:0]				VN27 [3:0]				8A
	1	1	↑	XX	X	VN43 [6:0]							40
	1	1	↑	XX	X	X	X	X	VN50 [3:0]				04
	1	1	↑	XX	X	X	X	VN57 [4:0]					18
	1	1	↑	XX	X	X	X	X	VN59 [3:0]				0F
	1	1	↑	XX	X	X	VN61 [5:0]						3F
	1	1	↑	XX	X	X	VN62 [5:0]						3F
	1	1	↑	XX	X	X	X	X	VN63 [3:0]				0F
Digital Gamma Control 1	0	1	↑	XX	1	1	1	0	0	0	1	0	E2h
1st Parameter	1	1	↑	XX	RCA0 [3:0]				BCA0 [3:0]				XX
:	1	1	↑	XX	RCAx [3:0]				BCAx [3:0]				XX
16th Parameter	1	1	↑	XX	RCA15 [3:0]				BCA15 [3:0]				XX
Digital Gamma Control 2	0	1	↑	XX	1	1	1	0	0	0	1	1	E3h
1st Parameter	1	1	↑	XX	RFA0 [3:0]				BFA0 [3:0]				XX
:	1	1	↑	XX	RFAx [3:0]				BFAx [3:0]				XX
64th Parameter	1	1	↑	XX	RFA63 [3:0]				BFA63 [3:0]				XX
Interface Control	0	1	↑	XX	1	1	1	1	0	1	1	0	F6h
	1	1	↑	XX	MY_EOR	MX_EOR	MV_EOR	X	BGR_EOR	X	X	WEMODE	01
	1	1	↑	XX	X	X	EPF [1:0]		X	X	MDT [1:0]		00
	1	1	↑	XX	X	X	ENDIAN	X	DM [1:0]		RM	RIM	00

Initialization for the LT24 through IL9341 (1)

```
void LCD_Init()
{
        alt_u16 data1,data2;
        alt_u16 data3,data4;
        Set_LCD_RST;
        Delay_Ms(1);
        Clr_LCD_RST;
        Delay_Ms(10);       // Delay 10ms // This delay time is necessary
        Set_LCD_RST;
        Delay_Ms(120);      // Delay 120 ms
//      Clr_LCD_CS;
        LCD_WR_REG(0x0011); //Exit Sleep
        LCD_WR_REG(0x00CF);             // Power Control B
                LCD_WR_DATA(0x0000);            // Always 0x00
                LCD_WR_DATA(0x0081);            //
                LCD_WR_DATA(0X00c0);

        LCD_WR_REG(0x00ED);             //  Power on sequence control
                LCD_WR_DATA(0x0064); // Soft Start Keep 1 frame
                LCD_WR_DATA(0x0003); //
                LCD_WR_DATA(0X0012);
                LCD_WR_DATA(0X0081);

        LCD_WR_REG(0x00E8); // Driver timing control A
                LCD_WR_DATA(0x0085);
                LCD_WR_DATA(0x0001);
                LCD_WR_DATA(0x00798);
```

Initialization for the LT24 through IL9341 (2)

```
LCD_WR_REG(0x00CB); // Power control A
        LCD_WR_DATA(0x0039);
        LCD_WR_DATA(0x002C);
        LCD_WR_DATA(0x0000);
        LCD_WR_DATA(0x0034);
        LCD_WR_DATA(0x0002);

LCD_WR_REG(0x00F7); // Pump ratio control
        LCD_WR_DATA(0x0020);

LCD_WR_REG(0x00EA); // Driver timing control B
        LCD_WR_DATA(0x0000);
        LCD_WR_DATA(0x0000);

LCD_WR_REG(0x00B1); // Frame Control (In Normal Mode)
        LCD_WR_DATA(0x0000);
        LCD_WR_DATA(0x001b);

LCD_WR_REG(0x00B6); // Display Function Control
        LCD_WR_DATA(0x000A);
        LCD_WR_DATA(0x00A2);

LCD_WR_REG(0x00C0);    //Power control 1
        LCD_WR_DATA(0x0005);   //VRH[5:0]

LCD_WR_REG(0x00C1);    //Power control 2
        LCD_WR_DATA(0x0011);   //SAP[2:0];BT[3:0]

LCD_WR_REG(0x00C5);    //VCM control 1
        LCD_WR_DATA(0x0045);       //3F
        LCD_WR_DATA(0x0045);       //3C
```

Initialization for the LT24 through IL9341 (3)

```
LCD_WR_REG(0x00C7);    //VCM control 2
LCD_WR_DATA(0X00a2);

LCD_WR_REG(0x0036);    // Memory Access Control
LCD_WR_DATA(0x0008);// BGR order

LCD_WR_REG(0x00F2);    // Enable 3G
LCD_WR_DATA(0x0000); // 3Gamma Function Disable

LCD_WR_REG(0x0026);    // Gamma Set
LCD_WR_DATA(0x0001); // Gamma curve selected

LCD_WR_REG(0x00E0);    // Positive Gamma Correction, Set Gamma
LCD_WR_DATA(0x000F);
LCD_WR_DATA(0x0026);
LCD_WR_DATA(0x0024);
LCD_WR_DATA(0x000b);
LCD_WR_DATA(0x000E);
LCD_WR_DATA(0x0008);
LCD_WR_DATA(0x004b);
LCD_WR_DATA(0X00a8);
LCD_WR_DATA(0x003b);
LCD_WR_DATA(0x000a);
LCD_WR_DATA(0x0014);
LCD_WR_DATA(0x0006);
LCD_WR_DATA(0x0010);
LCD_WR_DATA(0x0009);
LCD_WR_DATA(0x0000);
```

Initialization for the LT24 through IL9341 (4)

```
LCD_WR_REG(0X00E1);    //Negative Gamma Correction, Set Gamma
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x001c);
                LCD_WR_DATA(0x0020);
                LCD_WR_DATA(0x0004);
                LCD_WR_DATA(0x0010);
                LCD_WR_DATA(0x0008);
                LCD_WR_DATA(0x0034);
                LCD_WR_DATA(0x0047);
                LCD_WR_DATA(0x0044);
                LCD_WR_DATA(0x0005);
                LCD_WR_DATA(0x000b);
                LCD_WR_DATA(0x0009);
                LCD_WR_DATA(0x002f);
                LCD_WR_DATA(0x0036);
                LCD_WR_DATA(0x000f);

LCD_WR_REG(0x002A); // Column Address Set
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x00ef);

 LCD_WR_REG(0x002B); // Page Address Set
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x0000);
                LCD_WR_DATA(0x0001);
                LCD_WR_DATA(0x003f);

LCD_WR_REG(0x003A); // COLMOD: Pixel Format Set
                LCD_WR_DATA(0x0055);
```

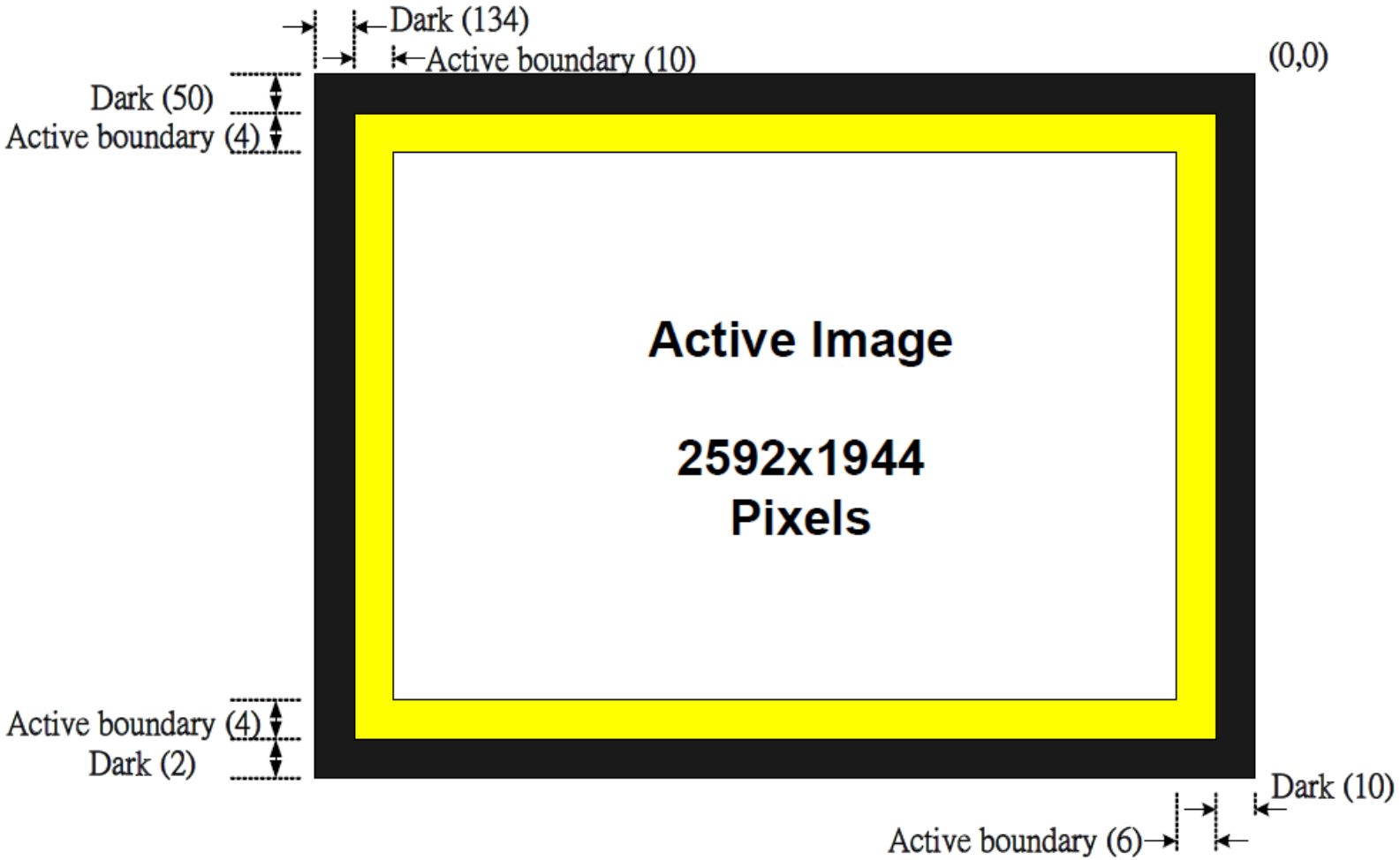
Initialization for the LT24 through IL9341 (5)

```
        LCD_WR_REG(0x00f6); // Interface Control
                LCD_WR_DATA(0x0001);
                LCD_WR_DATA(0x0030);
                LCD_WR_DATA(0x0000);

        LCD_WR_REG(0x0029); //display on

        LCD_WR_REG(0x002c);    // 0x2C

    }
```

Camera interface
Ex: TRDB-D5M (Terasic)

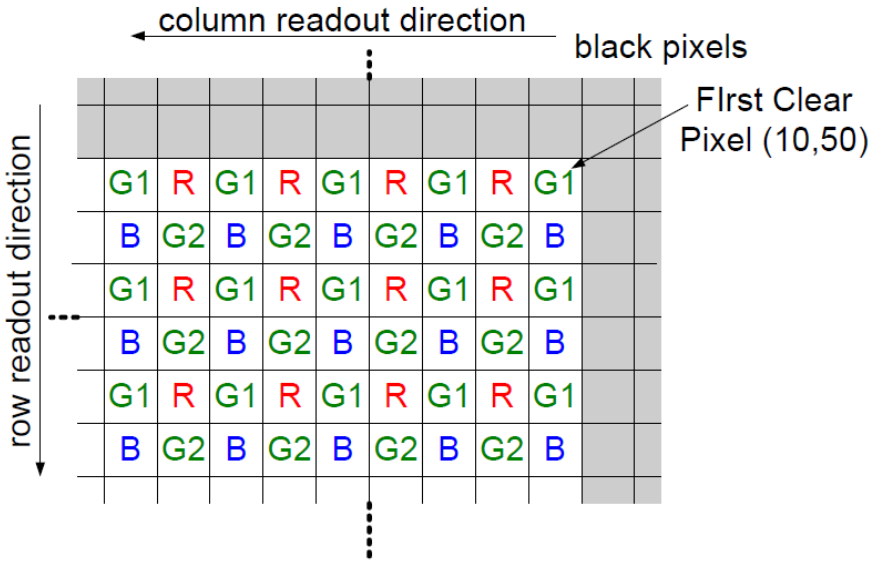
From: TRDB-D5M_Hardware Specification

Camera Pixels organisation (TRDB-D5M)

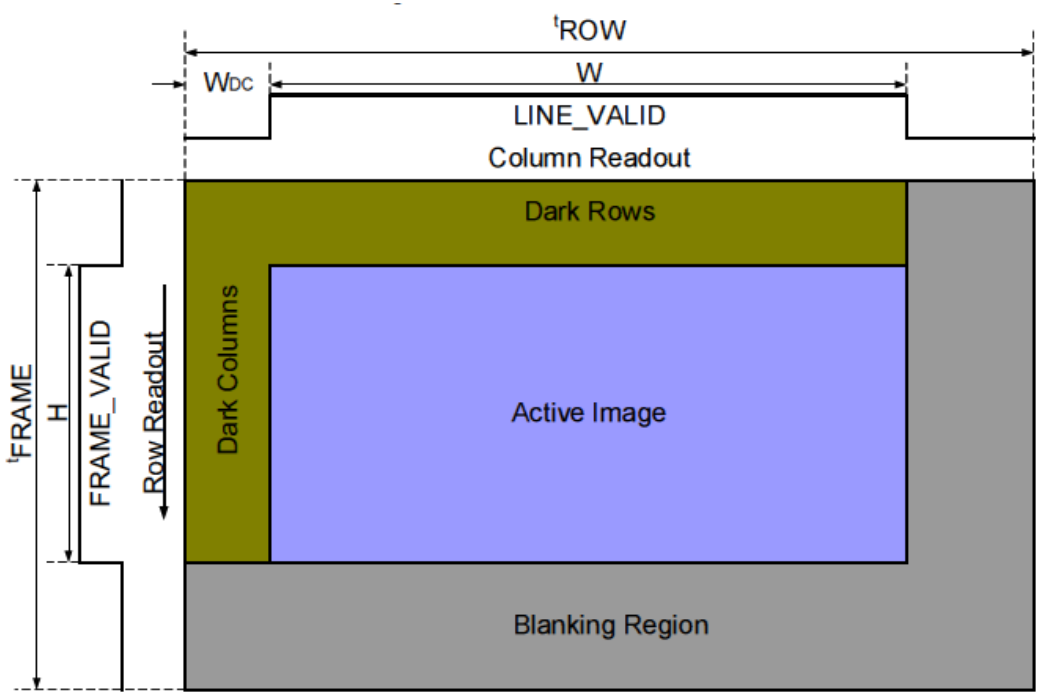


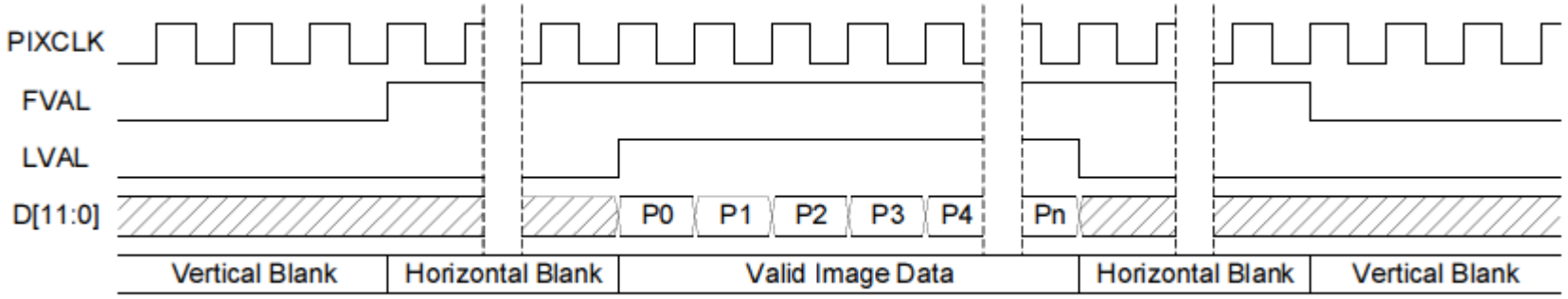
Camera basic Timing (TRDB-D5M)

- A 5Mpixels camera with 12 bits RGB
- 2592 x 1944 active rows from a
- 2752 x 2004 matrix
- Bayer colors Green1 – Red / Blue – Green2



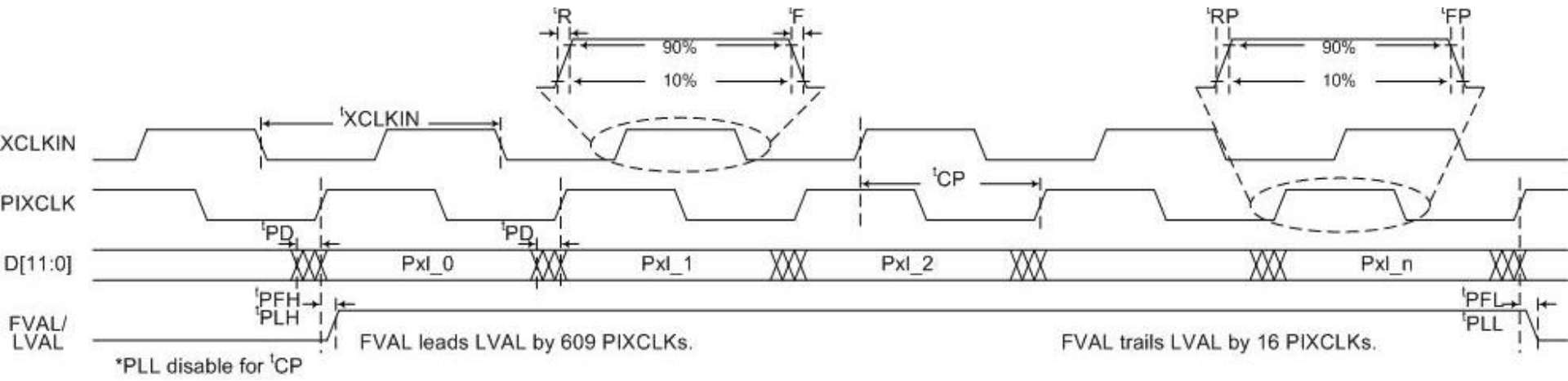
Camera basic Timing (TRDB-D5M)





Camera basic Timing (TRDB-D5M), Clock details

^tCP	XCLKIN to PIXCLK propagation delay	Nominal voltages	11.5	17.7	19.1	ns
^fPIXCLK	PIXCLK frequency	Default	6	—	96	MHz
^tPD	PIXCLK to data valid	Default	0.6	1.2	2.2	ns
^tPFH	PIXCLK to FV HIGH	Default	2.8	3.6	4.6	ns
^tPLH	PIXCLK to LV HIGH	Default	2.2	3.2	4.2	ns
^tPFL	PIXCLK to FV LOW	Default	2.4	3.4	4.2	ns
^tPLL	PIXCLK to LV LOW	Default	2.6	3.4	4.2	ns
CLOAD	Output load capacitance		—	<10	—	pF
CIN	Input pin capacitance		—	2.5	—	pF



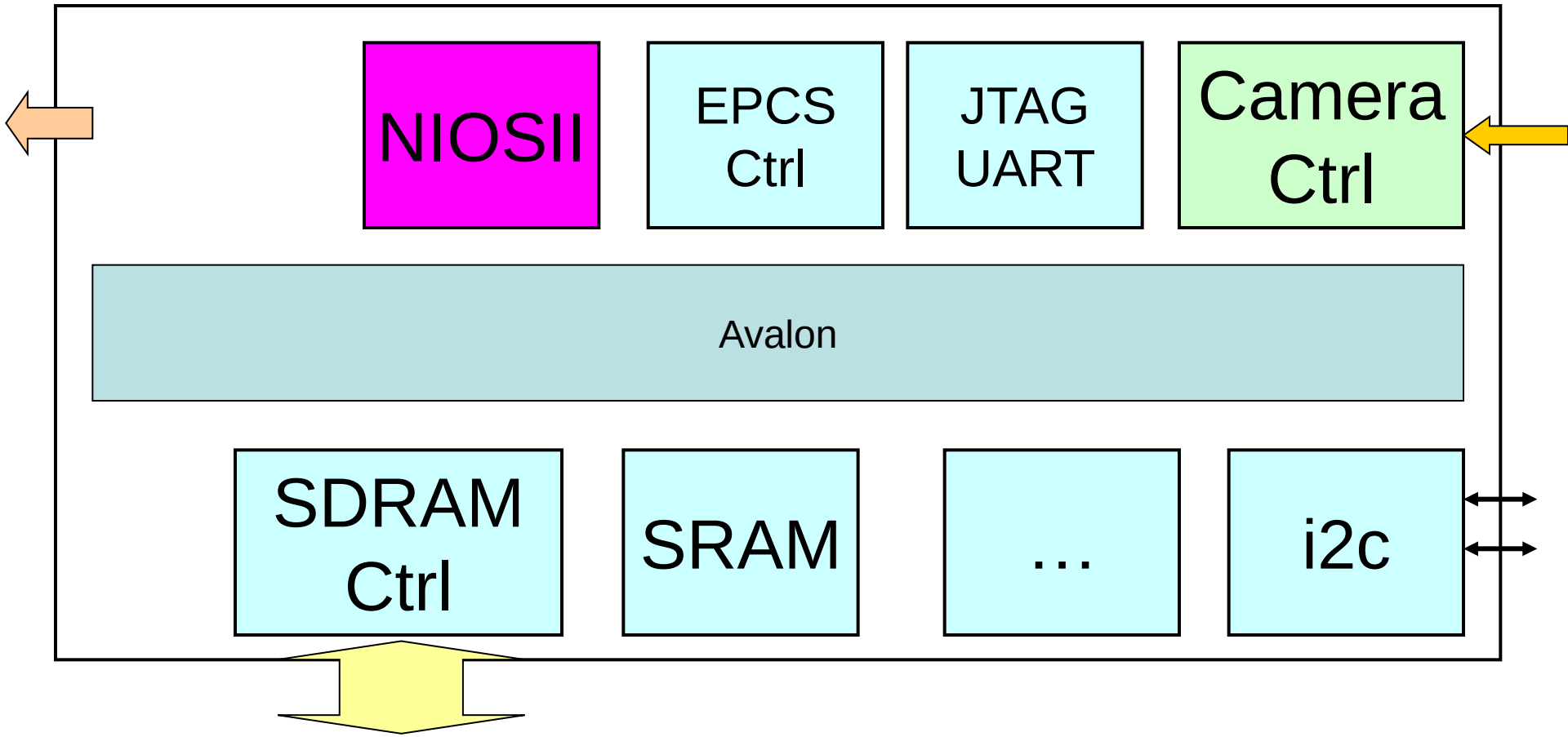
Camera basic Timing (TRDB-D5M)

Parameter	Name	Equation	Default Timing at EXTCLK = 96 MHz
fps	Frame Rate	1/tFRAME	15
^{t}FRAME	Frame Time	(H + max(VB, VBMIN)) × tROW	66ms
^{t}ROW	Row Time	2 × tPIXCLK x max(((W/2) + max(HB, HBMIN)), (41 + 208 x (Row_Bin+1) + 99))	33.5µs
W	Output Image Width	2 × ceil((Column_Size + 1) / (2 × (Column_Skip + 1)))	2592 PIXCLK
H	Output Image Height	2 × ceil((Row_Size + 1) / (2 × (Row_Skip + 1)))	1944 rows
SW	Shutter Width	max(1, (2*16 × Shutter_Width_Upper) + Shutter_Width_Lower)	1943 rows
HB	Horizontal Blanking	Horizontal_Blank + 1	1 PIXCLK
VB	Vertical Blanking	Vertical_Blank + 1	26 rows
HBMIN	Minimum Horizontal Blanking	208 × (Row_Bin + 1) + 64 + (WDC/2)	312 PIXCLK
VBMIN	Minimum Vertical Blanking	max(8, SW - H) + 1	9 rows
^{t}PIXCLK	Pixclk Period	1/fPIXCLK	10.42ns

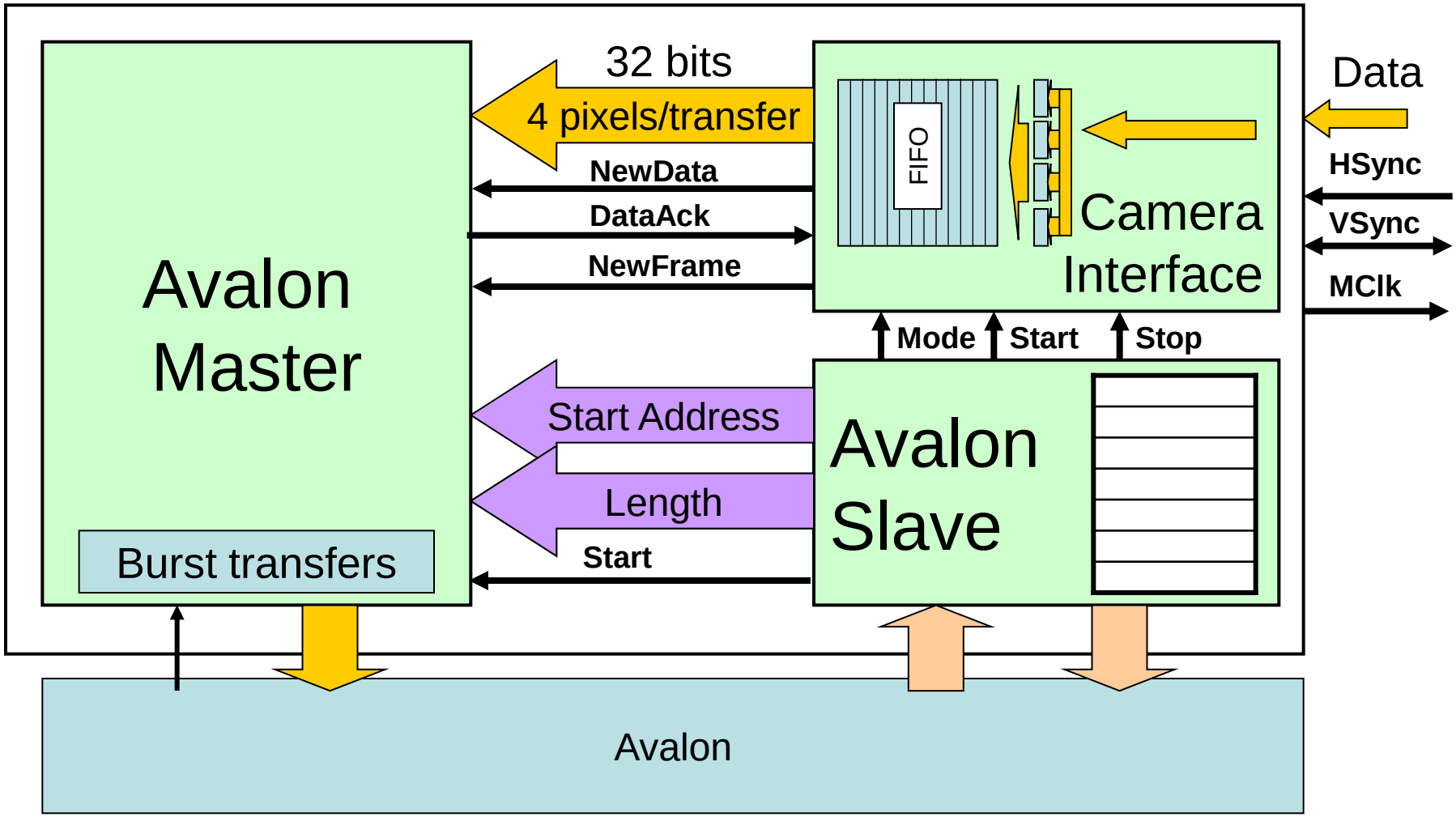
Camera Standard Resolution (TRDB-D5M)

Resolution	Frame Rate	Sub-sampling Mode	Column_Size (R0x04)	Row_Size (R0x03)	Shutter_Width_Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column_Bin (R0x23 [5:4])	Column_Skip (R0x23 [2:0])
2592 x 1944 (Full Resolution)	15.15	N/A	2591	1943	<1943	0	0	0	0
2,048 x 1,536 QXGA	23	N/A	2047	1535	<1535	0	0	0	0
1,600 x 1,200 UXGA	35.2	N/A	1599	1199	<1199	0	0	0	0
1,280 x 1,024 SXGA	48	N/A	1279	1023	<1023	0	0	0	0
	48	skipping	2559	2047		0	1	0	1
	40.1	binning	2559	2047		1	1	1	1
1,024 x 768 XGA	73.4	N/A	1023	767	<767	0	0	0	0
	73.4	skipping	2047	1535		0	1	0	1
	59.7	binning	2047	1535		1	1	1	1
800 x 600 SVGA	107.7	N/A	799	599	<599	0	0	0	0
	107.7	skipping	1599	1199		0	1	0	1
	85.2	binning	1599	1199		1	1	1	1
640 x 480 VGA	150	N/A	639	479	<479	0	0	0	0
	150	skipping	2559	1919		0	3	0	3
	77.4	binning	2559	1919		3	3	3	3

FPGA architecture

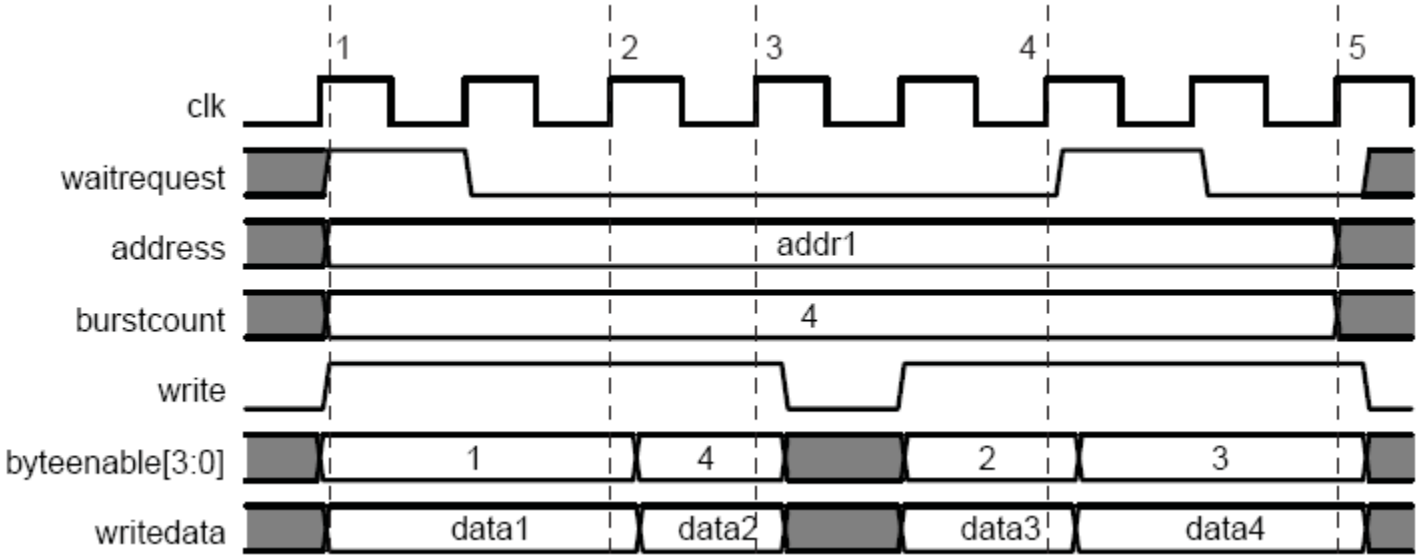


Camera Controller architecture



Camera Interface, signals on Avalon Master

- As a master, burst access allows the transfer of uninterruptable data flow
- The **BurstCount** is provided by the master unit and the number of announced data has to be provided



Camera Interface, signals

- Camera interface :
  - Mclk — Cam_Mclk
  - HSync — Cam_HSync
  - VSync — Cam_VSync
  - CamData[7..0] — Cam_data[7..0]
  - CamReset_n — Cam_Reset_n

- Slave interface → interface programmation
  - Clk — Clk
  - Address — AS_Address[2..0]
  - CSelect — AS_Cs_n
  - Write — AS_Write_n
  - DataWrite[31..0] — AS_Datawr[31..0]
  - Read — AS_Read_n
  - DataRead[31..0] — AS_Datard[31..0]
  - InterruptRequest — AS_IRQ_n
- Master interface → Data transfers to memory :
  - Clk — Clk
  - Address [31..0] — AM_Address[31..0]
  - ByteEnable_n[3..0] — AM_ByteEnable_n[3..0]
  - BurstCount — AM_BurstCount[2..0]
  - Write — AM_Write_n
  - DataWrite[31..0] — AM_Datawr[31..0]
  - WaitRequest — AM_WaitRequest

Camera Interface, slave access: internal registers

Address	Register	Rz value	Size	Description
00h	**CamAddr**	0h	32	Destination Address
04h	**CamLength**	128*101	24	Buffer size in bytes
08h	**CamComm**	00h	8	Command
0Ch	**CamStatus**	00h	8	Status
10h	**CamStart**	0	8	Acquisition enabled
14h	**CamStop**	0	8	Stop acquisition
18h	**CamSnapshot**	0	8	Snapshot, activate VSync

Camera Registers (TRDB-D5M)

- 256 internal registers available through **i2c**
- ~40 used
- Clock programming
- Frame programming

I²c (TRDB-D5M)

- I2c transfers
- Addresses:
  - 0xBA: write ‘1011 1010’
  - 0xBB: read ‘1011 1011’
- 400 kHz
- Register Address on 8 bits

Camera to work (TRDB-D5M), External Clk mode

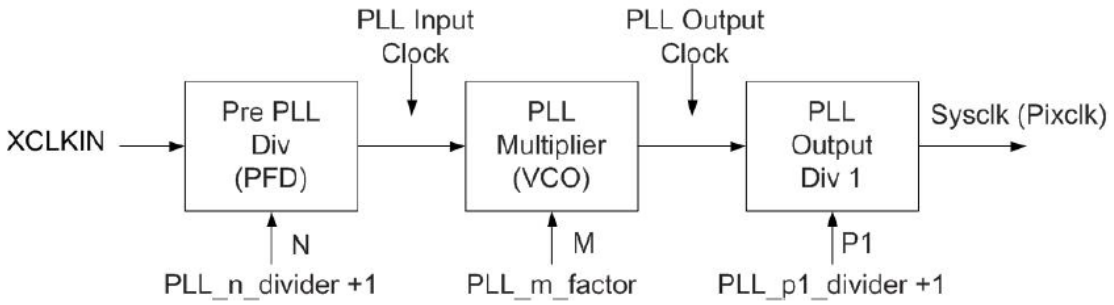
- Activate Reset_n
- Power On
- Desactivate Reset_n
- External Clock provided
  - XClkIn: 6 .. 96 MHz Clock to be provided
  - fPixClk = fXClkIn if Divide_Pixel_Clock =0
  - fPixClk = fXClkIn/(2 x Divide_Pixel_Clock)

Camera to work (TRDB-D5M), PLL mode

- PLL pixel clk generation
- fXClkIn: 6MHz .. 27MHz
- Power PLL
- Set M, N, P1

$$^{f}PIXCLK = (^{f}XCLKIN \times M) / (N \times P1)$$

The PLL control registers must be programmed while the sensor is in the software Standby state. The effect of programming the PLL divisors while the sensor is in the streaming state is UNDEFINED.



Camera to work (TRDB-D5M), PLL mode

- PLL pixel clk generation
- fXClkIn: 6MHz .. 27MHz
- Power PLL
- Set M, N, P1 $^{f}PIXCLK = ({}^{f}XCLKIN \times M) / (N \times P1)$
- 2 MHz < fXClkIn/N < 13.5 MHz
- 180 MHz < (fXClkIn * M) / N < 360 MHz
- M: 16..255
- Use_PLL (Reg0x10[1] = 1) from XClkIn to PLL mode

Camera to work (TRDB-D5M), Skipping

- Skipping of line/column

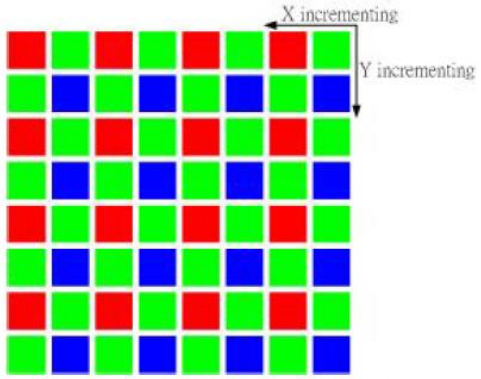


Figure 3.3: Pixel Readout (no skipping)

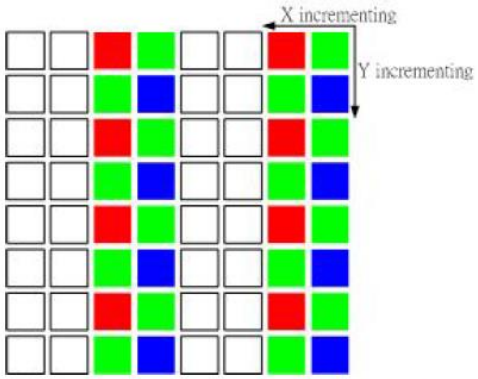


Figure 3.4: Pixel Readout (Column Skip 2X)

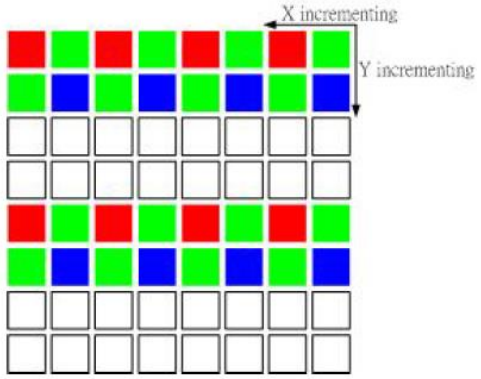


Figure 3.5: Pixel Readout (Row Skip 2X)

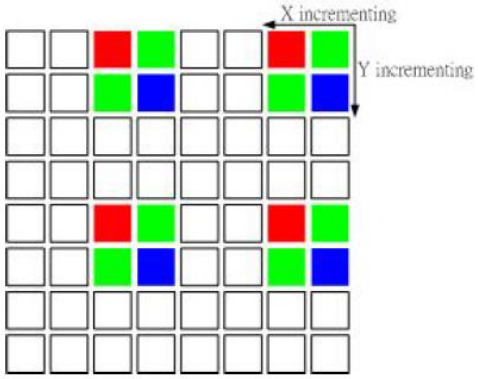


Figure 3.6: Pixel Readout (Column Skip 2X, Row Skip 2X)

Camera to work (TRDB-D5M), Binning

- Binning of line/column

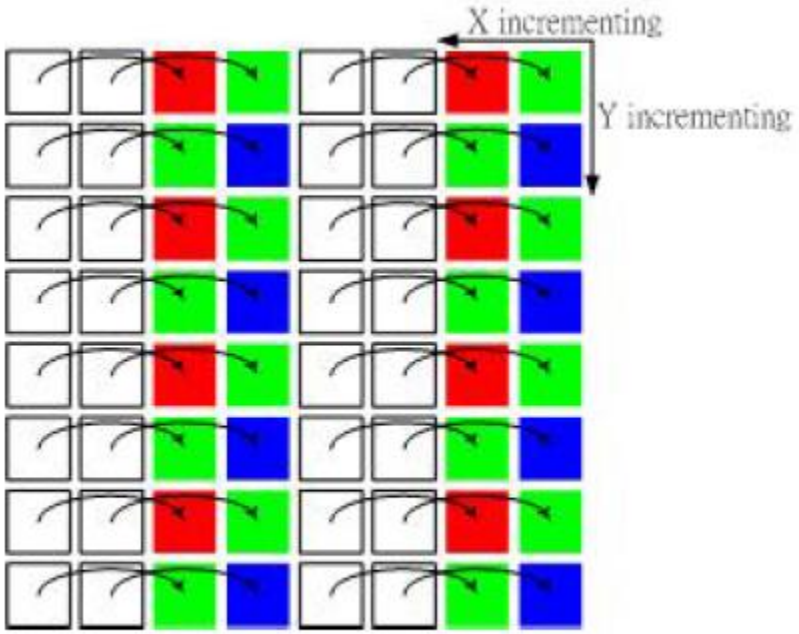


Figure 3.7: Pixel Readout (Column Bin 2X)

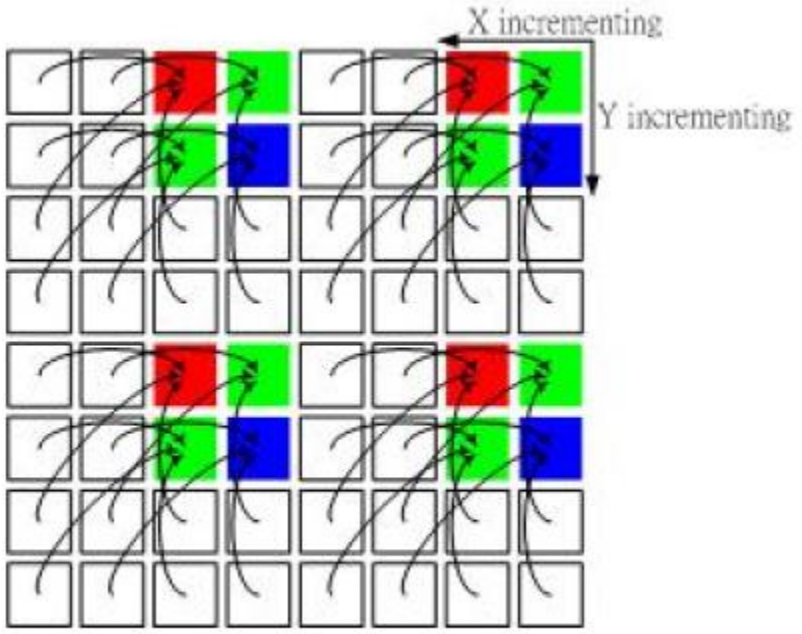
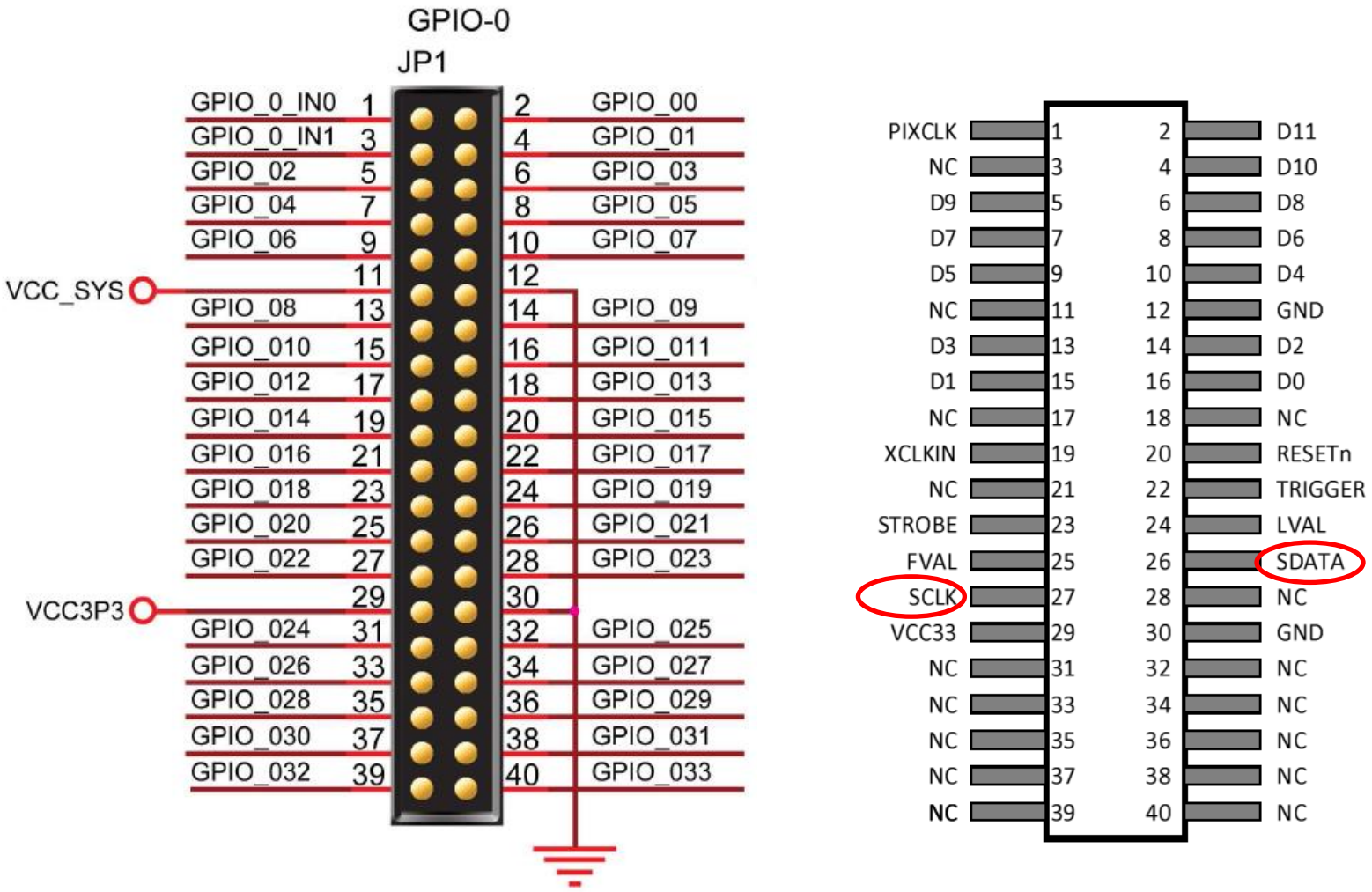


Figure 3.8: Pixel Readout (Column Bin 2X, Row Bin 2X)

DE0-nano / Camera Module pinning

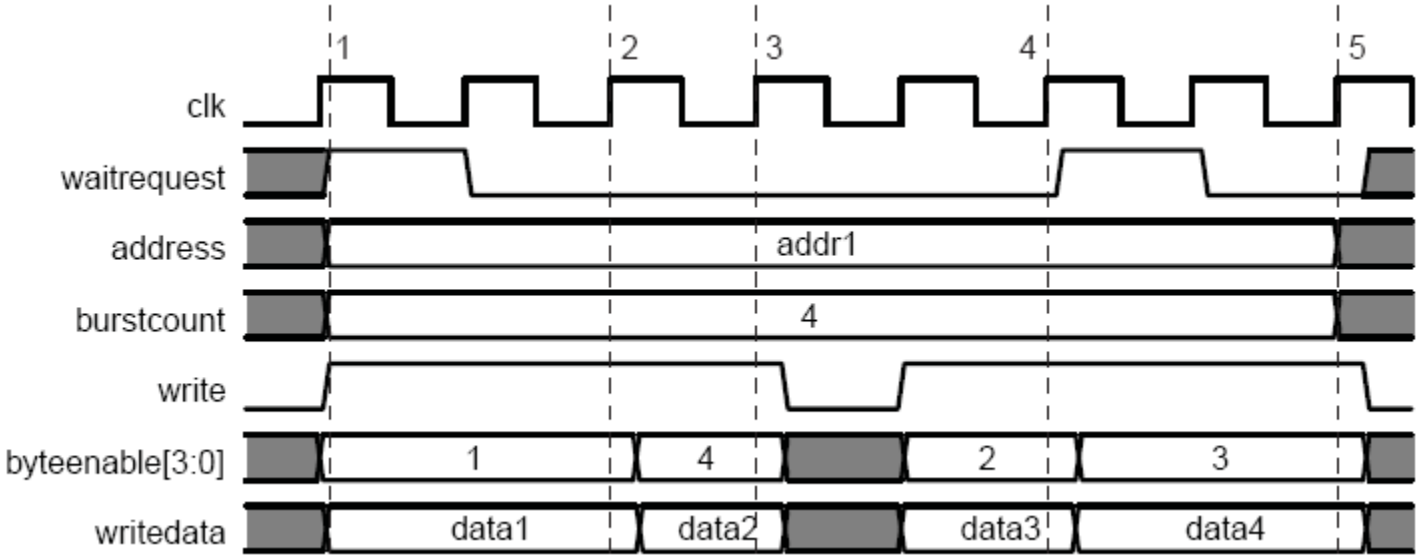


Camera Interface, signals

		Dir, FPGA view
• Camera interface :		
➢ Pixel_Clk	PixClk	In (from Camera to FPGA)
➢ LineValid	Cam_Lval	In
➢ FrameValid	Cam_Fval	In
➢ CamData[11..0]	Cam_data[11..0]	In
➢ CamReset_n	Cam_Reset_n	Out
➢ Strobe		In
➢ Trigger		Out
➢ XCLKIN		Out
• Slave interface → interface programmation		
➢ Clk	Clk	In
➢ Address	AS_Address[2..0]	In
➢ CSelect	AS_Cs_n	In
➢ Write	AS_Write_n	In
➢ DataWrite[31..0]	AS_DataWr[31..0]	In
➢ Read	AS_Read_n	In
➢ DataRead[31..0]	AS_DataRd[31..0]	Out
➢ InterruptRequest	AS_IRQ_n	Out
• Master interface → Data transfers to memory :		
➢ Clk	Clk	In
➢ Address [31..0]	AM_Address[31..0]	Out
➢ ByteEnable_n[3..0]	AM_ByteEnable_n[3..0]	Out
➢ BurstCount	AM_BurstCount[2..0]	Out
➢ Write	AM_Write_n	Out
➢ DataWrite[31..0]	AM_DataWr[31..0]	Out
➢ WaitRequest	AM_WaitRequest	In

Camera Interface, signals on Avalon Master

- As a master, burst access allows the transfer of uninterruptable data flow
- The **BurstCount** is provided by the master unit and the number of announced data has to be provided

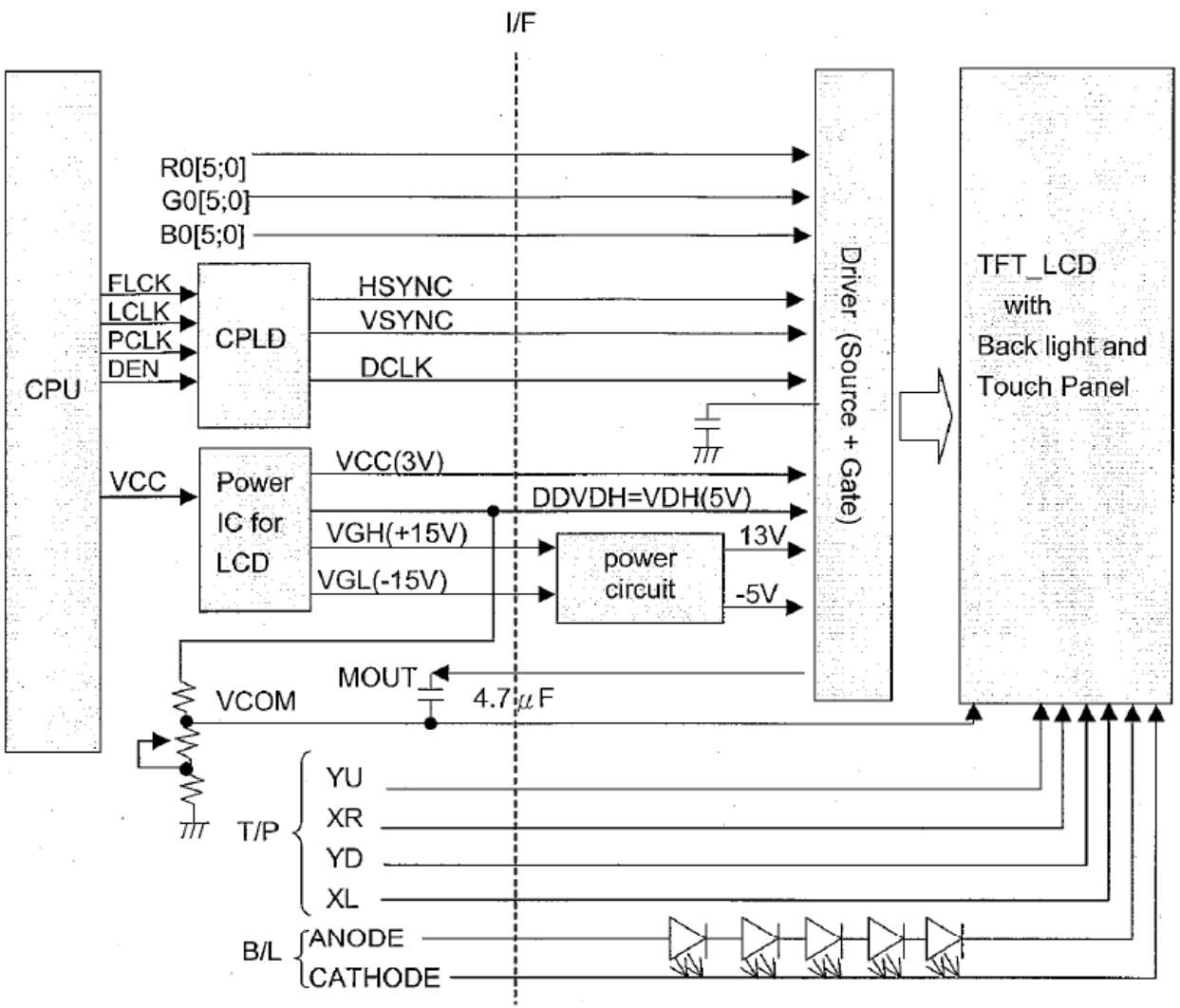


Another LCD: TX-07
Another Protocol

TX07- Interface with TFT-LCD

- This kind of LCD is controlled by a parallel data flow: **RGB** digital intensity
- It's synchronized at Frame level with Vertical Synchronization : **VSync**
- It's synchronized at Line level with Horizontal Synchronization : **HSync**
- It's synchronized by a Clock at pixel level: **DotClk**

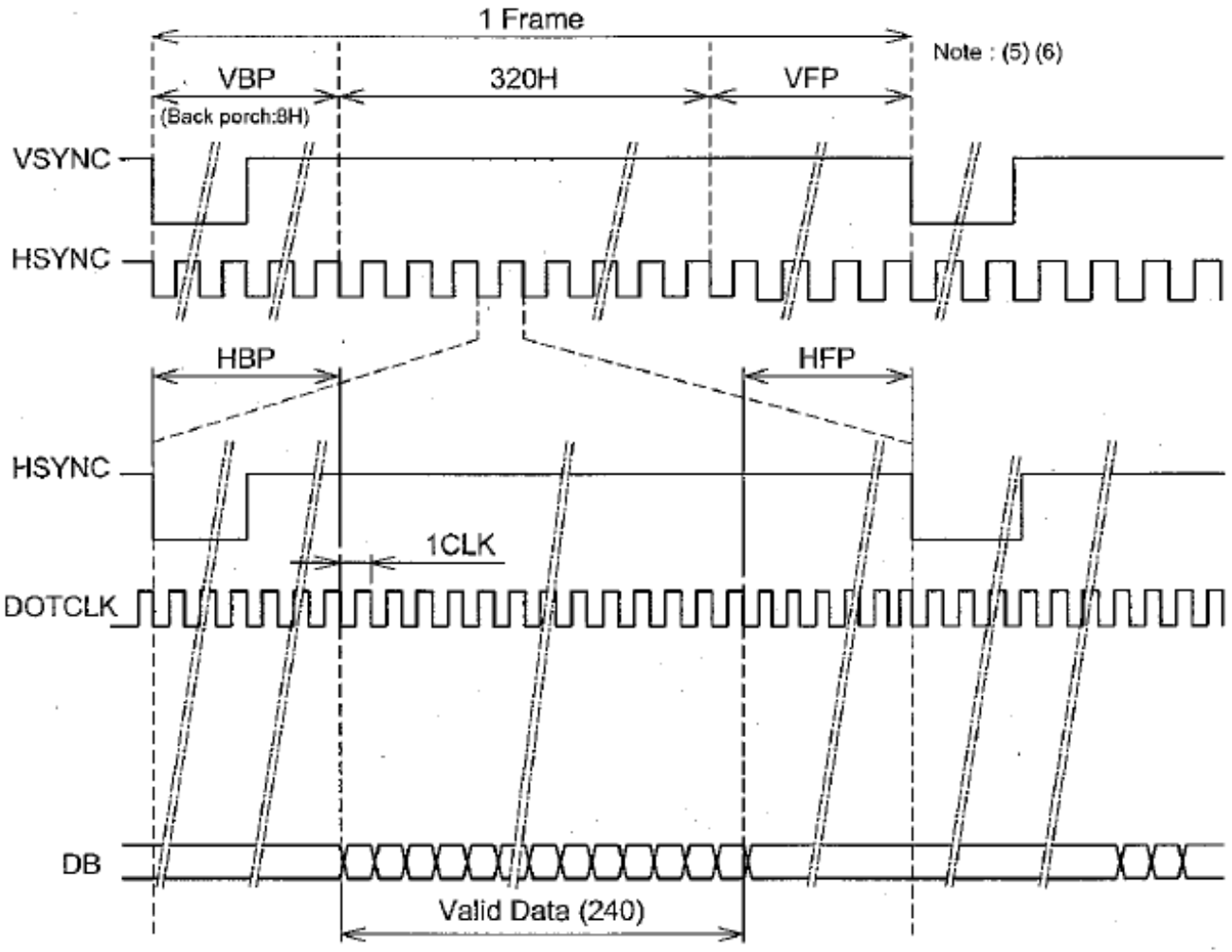
TX07 TFT_LCD


I/F
CPU
R0[5;0]
G0[5;0]
B0[5;0]
FLCK
LCLK
PCLK
DEN
CPLD
HSYNC
VSYNC
DCLK
VCC
Power
IC for
LCD
VCC(3V)
DDVDH=VDH(5V)
VGH(+15V)
VGL(-15V)
power
circuit
13V
-5V
Driver (Source + Gate)
TFT_LCD
with
Back light and
Touch Panel
MOUT
VCOM
4.7μF
T/P
YU
XR
YD
XL
B/L
ANODE
CATHODE

Interface with TFT-LCD

- Some Voltage need to be generated for the LCD:
  - **VGH/VGL @+/- 15V** for internals +13V/-5V
  - **Vcc @ 3V**
  - **VDH @ 5V**
  - **VCOM @ ~2.2V**
- **A DotClk @ 5 ~12 MHz**

TX07 Synchronization



TX07 Synchronization

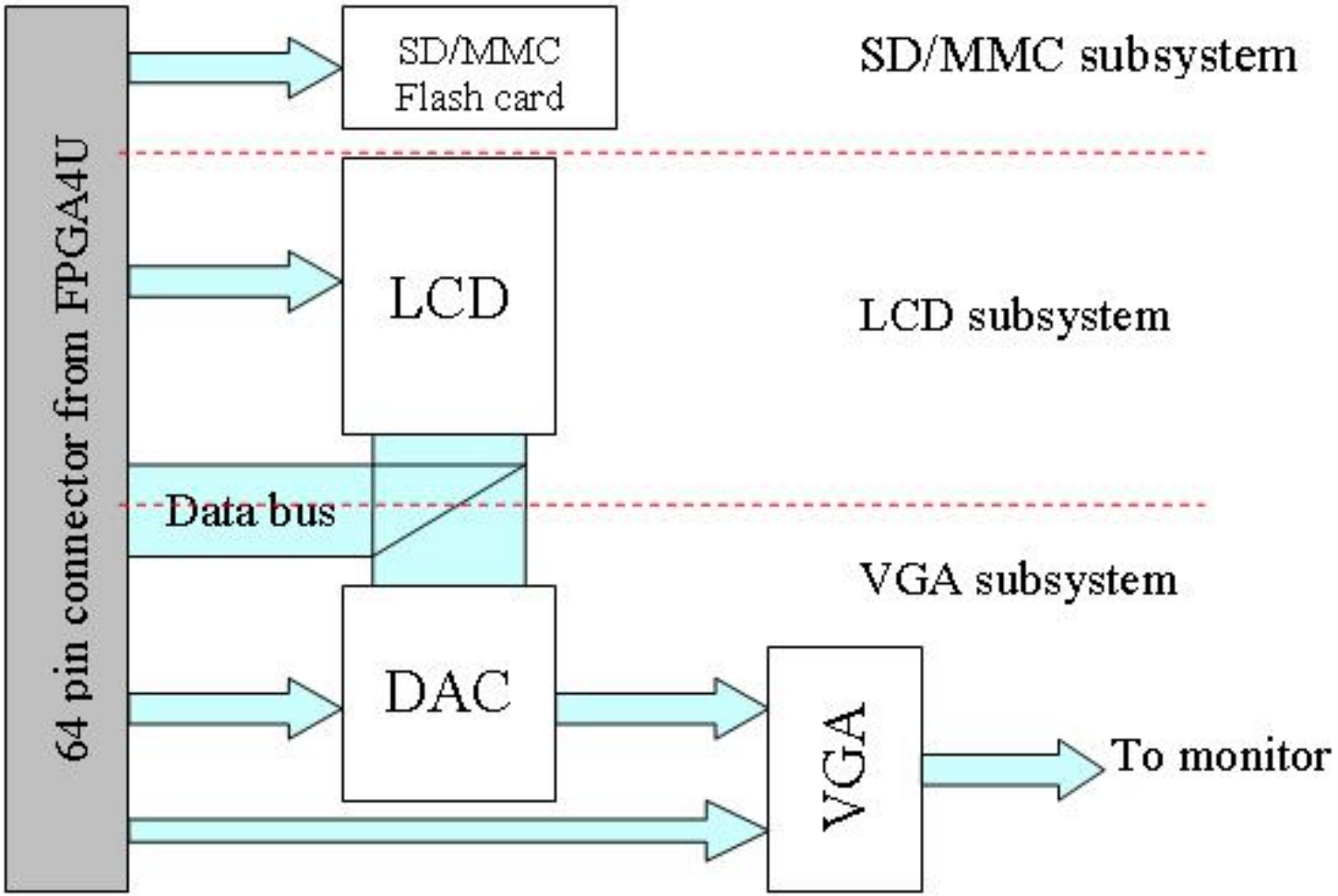
	SYMBOL	MIN.	TYP.	MAX.	UNIT
Back porch for Horizontal	HBP	-	12	-	Clock
Front porch for Horizontal	HFP	15	18 Note(1)	21	Clock
Back porch for Vertical	VBP	-	8 Note(3)	-	HSYNC
Front porch for Vertical	VFP	17	(20) Note(2)	22	HSYNC

Note(1) : (DOTCLK total) - ((Valid data period for Horizontal) + (HBP))
(2) : (HSYNC total) - ((Active Area period) + VBP)

Extension for FPGA4U

- An extension board for FPGA4U allows connection of 2 different LCD from Hitachi:
  - TX06
  - http://www.hitachi-displays-eu.com/doc/TX06D57VM0AAA.pdf
  - **TX07**
  - http://www.hitachi-displays-eu.com/doc/TX07D09VM1CAB.pdf
- And a VGA interface with a DAC
  - http://focus.ti.com/lit/ds/symlink/ths8133b.pdf

ExtLCD for FPGA4U



Avalon Slave, registers

- **FBufAdd**, Address of the frame buffer to display
- **FBLgt**, Length of the frame buffer *(in pixels number, 240 x 320 by default)*
- **DisplayCom**, Command *(Enable display, IRQ at end of Frame, Power Control)*
- **DisplayStat,** Status *(Run, EOF (End Of frame))*
- **HBP** *(Horizontal Back Porch) (default: 12)*
- **HFP** *(Horizontal Front Porch)***:** *(default: 18)*
- **VBP** *(Vertical Back Porch)***:** *(default: 8)*
- **VFP** *(Vertical Front Porch)* **:** *(default: 20)*
- **HData** *(Horizontal Data)* **:** *(default: 240)*
- **VData** *(Vertical Data)* **:** *(default: 320)*
- *…*

Avalon Slave, registers mapping

Add. (Offset)	Name	Reset Val.	Function	Size
0x0	**FBAdd**	0	Address of the frame buffer to display	32
0x1	**FBLgt**	240 x 320	Length of the frame buffer in pixels	32
0x2	**DisplayCom**	0	Command (Start, Stop, IRQ, Power)	8
0x3	**DisplayStat**	0	Status (Run, EOF (End Of frame))	8
0x4	**HBP**	12	*Horizontal Back Porch* (Nb DotClk)	16
0x5	**HFP**	18	*Horizontal Front Porch* (Nb DotClk)	16
0x6	**VBP**	8	*Vertical Back Porch (Nb Lines)*	16
0x7	**VFP**	20	*Vertical Front Porch (Nb Lines)*	16
0x8	**HData**	240	*Horizontal Data* (Nb DotClk)	16
0x9	**VData**	320	*Vertical Data (Nb Lines)*	16
0xA	**HSync**	2	Horizontal Sync Length (Nb DotClk)	16
0xB	**VSync**	7	Vertical Sync Length (Nb Lines)	16
0xC			68	

Power control

- 3 signals allows control of Power on the LCD module. They have to be controlled by the module as 3 Ports bits.
- The **DisplayCom** register controls them
- **StepUp_ON** :
  - Allows internal +5V generation, necessary for VGA and LCD on 45 pins connector
- **LED_ON** :
  - Allows LED back light ON
- **LCD_ON :**
  - Needed for LCD TX07 to work

LCD Controller : LCD Control

- The LCD Control part send the synchronization signals to the LCD:
  - VSync
  - HSync
  - DotClk
  - RGB (3x6 bits/pixel)
- Read the pixels data from FIFO
- Receive information from Avalon slave part through registers interface
- It contains counters for signals and timing generation
- It's based on a state machine to control them

LCD Controller : FIFO

- Writing of data from the Master interface and reading from LCD control have to be perfectly synchronized
- Read access from SDRAM memory can **not** be guaranties at pixel level timing
- We have to guaranty that the global data flow is possible and with which delay!
- A FIFO is an excellent way to allow synchronization between 2 asynchronous units

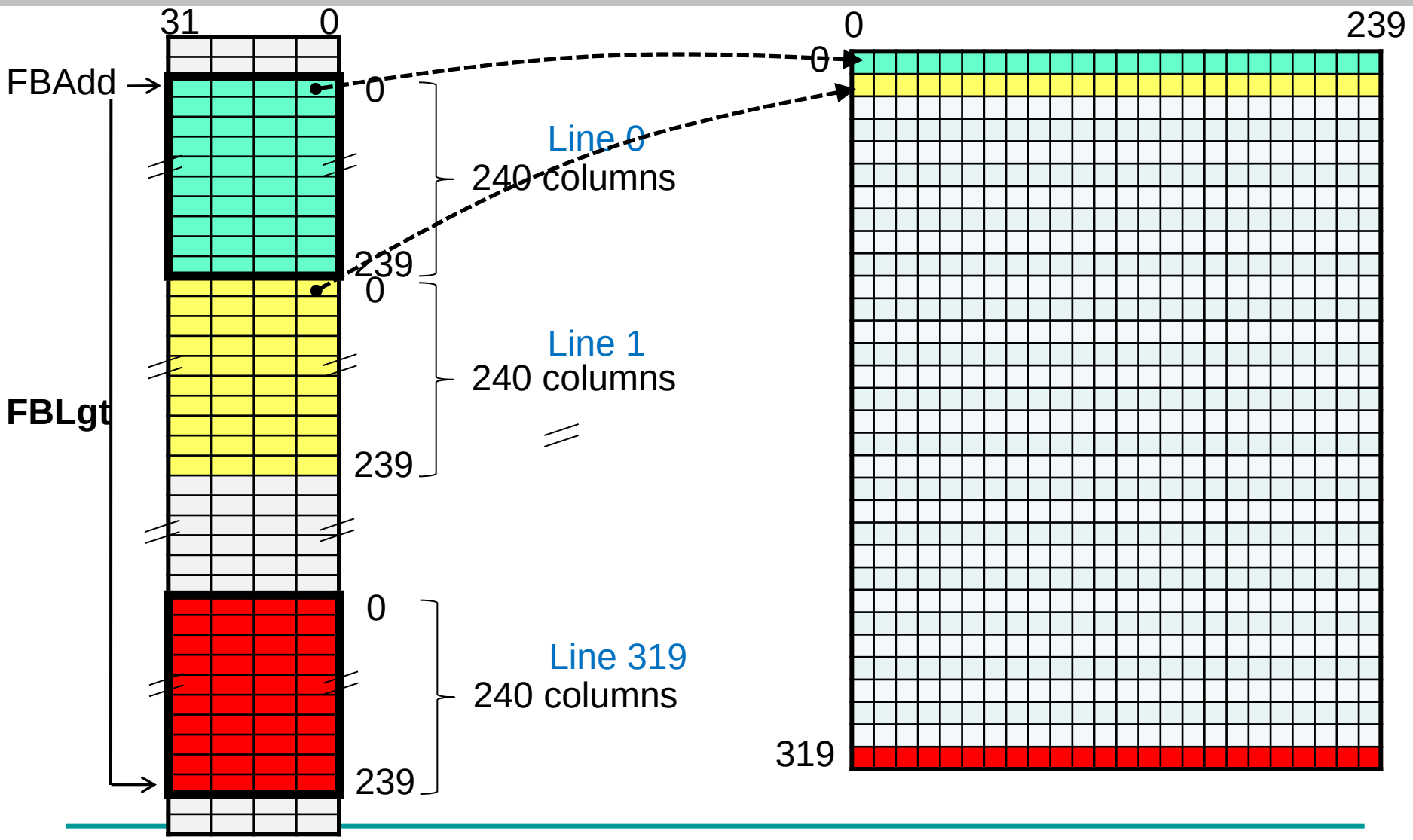
LCD Controller : FIFO

- Rules:
  - The Avalon master try to fulfill the FIFO when **it can**
  - The LCD control unit read it when **it needs.**
- To be efficient, the Avalon master try to make consecutive reads, thus the SDRAM controller can do **burst transfers**.
- The FIFO needs to provide an information to the Avalon master module when it has a minimum of empty positions as a multiple of the burst transfers (4, 8, 16, ..).
- The FIFO send information to the LCD Control module when it has at least 1 available pixel data.

Global questions

- Format of the bitmap in memory
- Color organization
- Pixel resolution

Memory ←→ Display relation

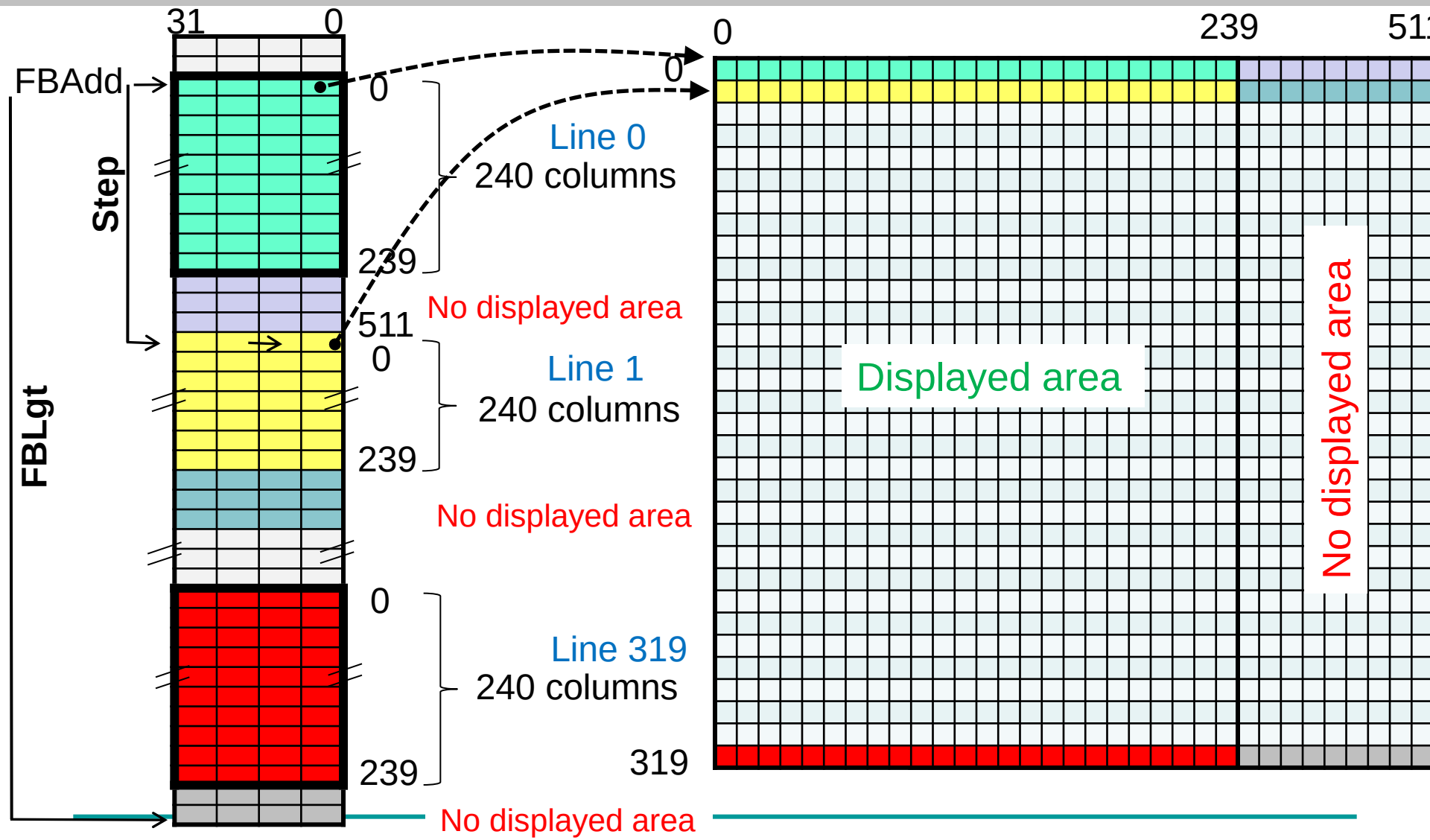


Memory organization

- Visible area
- Shadow area, not displayed
- →Steps between lines

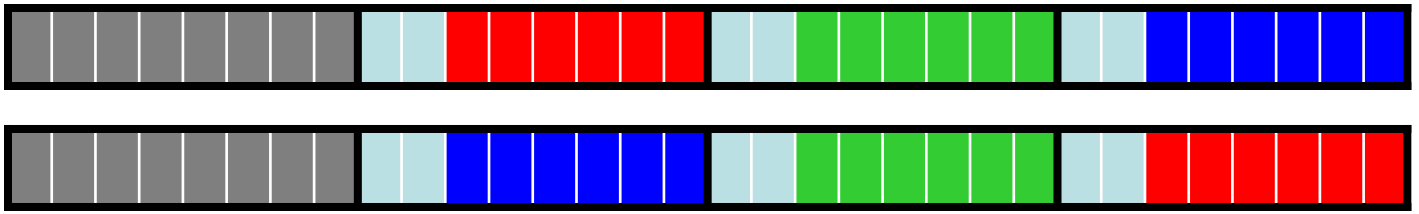
- → Added parameters in registers definition
- → Transmitted to the DMA unit for addresses calculations

Memory ←→ Display relation

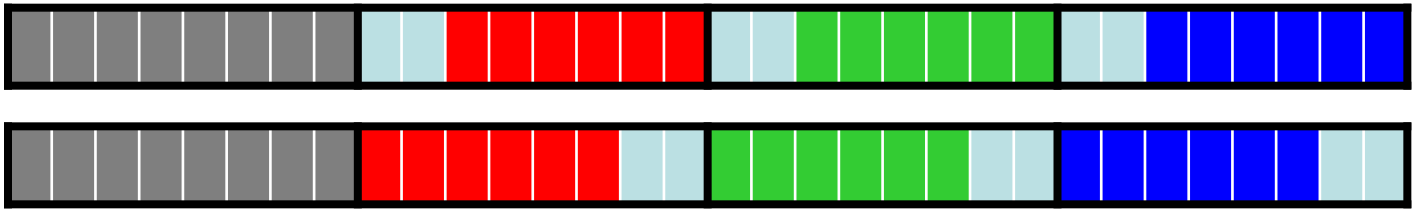


Pixel organization, some choices

- 1 pixel organization:
  - 18 bits/pixel at LCD
  - Ex. **32** bits /pixel at memory
  - RGB or BGR ?

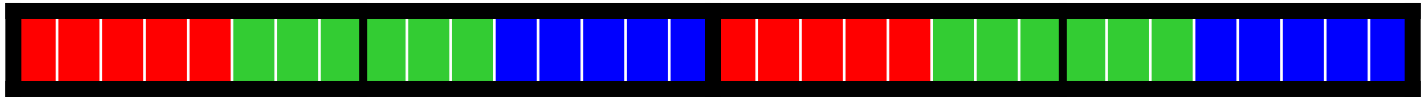


  - 6 bits / Byte: right or left alignment ?

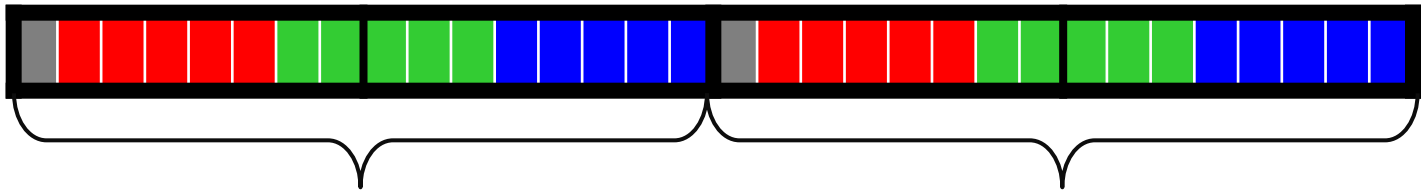


Pixel organization, some choices

- 1 pixel organization:
  - 18 bits/pixel at LCD
  - Ex. **16** bits /pixel at memory, 2 pixels / 32 bits
  - RGB or BGR ?
  - 5-6-5 bits / doublet: RGB, 2 bits lost



  - 5-5-5 bits / doublet: RGB, 3 bits lost



Pixel organization

- Many choices
- Need to be done !
- More bits/pixel 18 on 32 bits→
  - more memory for a frame
  - more bandwidth necessary
  - more colors available 2^{18} : 262144
  - Memory space free for other function
- Less bits/pixel (16)
  - Lost in color resolution 15 bits→ 32’768
  - 16 bits 65’536, more on Green generally