LCD Description

- Color display
- 2.4'' Diagonal
- 240 x 320 pixels
- Up to 18 bit resolution (We will use 16)
- ILI9341 controller
- Controller allows several interfaces. We will use the 8080-series parallel interface with 16 bits.
- Available documentation:
  - LCD documentation (practical and very short)
  - ILI9341 controller documentation (very complete... too much)

8080 Series Parallel Interface

- The LCD interface has 7 signals :
  - DB (16 bits) for data bus (also used for commands).
  - Read control signal (RD_n)
  - Write control signal (WR_n)
  - Chip-Select (CS_n)
  - D/Cx control signal indicates whether in the bus we are sending data or commands. (D_C_n)
  - LCD_Reset_n,
  - Interface Mode signal allowing to select either an interface on 8 or 16 bits (IM0)

Commands (or registers…) [1]

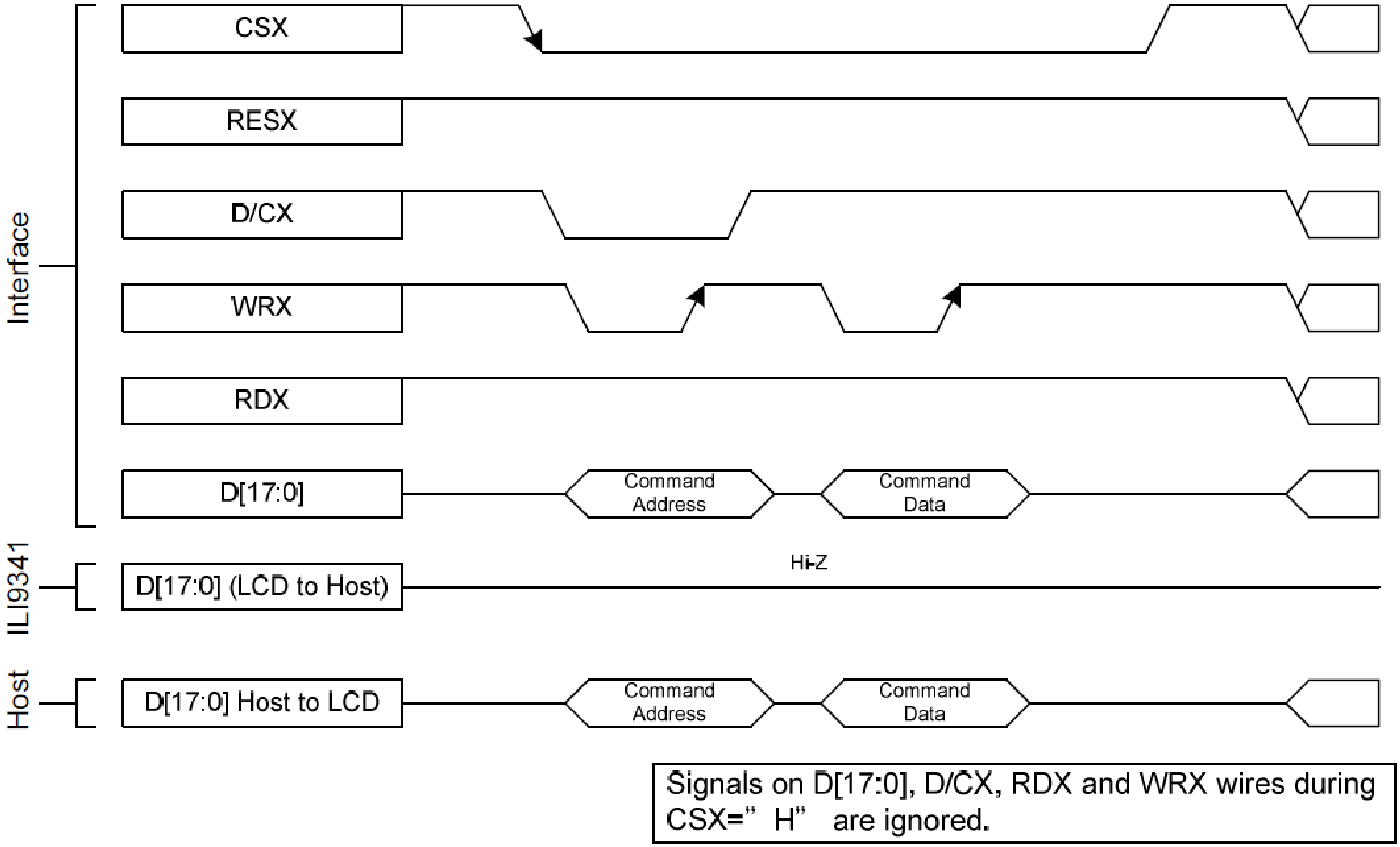
- As set of commands allow you to initialise power seting, adjust gamma setings, select resolution (18 bits, 16 bits,...), transfer mode, screen size, scrolling, etc.
- These commands are performed in two accesses : first a register address followed by the data to be writen to the register.
- Adresses and data must be sent in 16 bits. We will thus use the 16 bits interface. It is easier and faster.

Commands (or registers…) [2]

- You can either study every register to write your initialisation… or use the *init_LCD()* function available in moodle.
- This initialisation sets a pixel format BGR on 16 bits assigning 5,6,and 5 bits respectively for each color.
- For sending pixels you must initially write to the register 0x002C and then send 240x320 pixels to fill the screen. Each pixel sent in a 16-bit transfer.

Your hardware interface must...

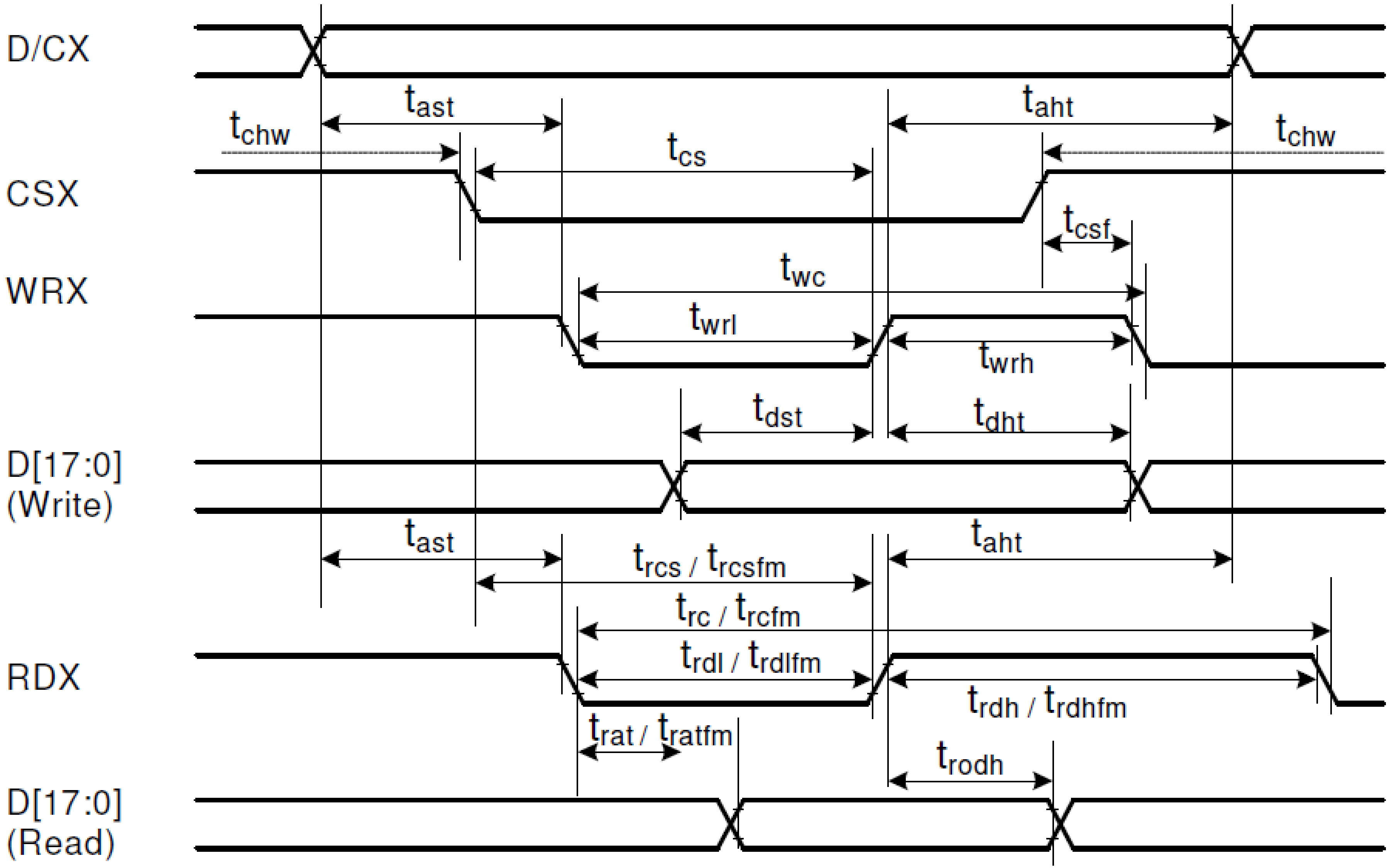
- Translate Avalon to LCD format accesses :



Signals on D[17:0], D/CX, RDX and WRX wires during CSX=” H” are ignored.

- Respecting the following timing :

-



Signal	Symbol	Parameter	min	max	Unit	Description
DCX	tast	Address setup time	0	-	ns	
	taht	Address hold time (Write/Read)	0	-	ns	
CSX	tchw	CSX “H” pulse width	0	-	ns	
	tcs	Chip Select setup time (Write)	15	-	ns	
	trcs	Chip Select setup time (Read ID)	45	-	ns	
	trcsfm	Chip Select setup time (Read FM)	355	-	ns	
	tcsf	Chip Select Wait time (Write/Read)	10	-	ns	
WRX	twc	Write cycle	66	-	ns	
	twrh	Write Control pulse H duration	15	-	ns	
	twrl	Write Control pulse L duration	15	-	ns	
RDX (FM)	trcfm	Read Cycle (FM)	450	-	ns	
	trdhfm	Read Control H duration (FM)	90	-	ns	
	trdlfm	Read Control L duration (FM)	355	-	ns	
RDX (ID)	trc	Read cycle (ID)	160	-	ns	
	trdh	Read Control pulse H duration	90	-	ns	
	trdl	Read Control pulse L duration	45	-	ns	
D[17:0], D[17:10]&D[8:1], D[17:10], D[17:9]	tdst	Write data setup time	10	-	ns	For maximum CL=30pF For minimum CL=8pF
	tdht	Write data hold time	10	-	ns	
	trat	Read access time	-	40	ns	
	tratfm	Read access time	-	340	ns	
	trod	Read output disable time	20	80	ns	

Hardware interface

- Define a register model.
  - How many « registers » to be addressed from software ?
- Define an architecture with the following behavior :
  - When *avalon_CS* and *avalon_WR* detected :
    - Block the avalon bus with *waitrequest* signal
    - Generate LCD control signals (*LCD_WRn* and *LCD_D_Cn*) respecting timing from the datasheet. They can be generated from a state machine or a counter followed by a decoder.
- Slow control signals like LCD_Reset_n, LCD_CS_n and IM0 can be generated from a GPIO.