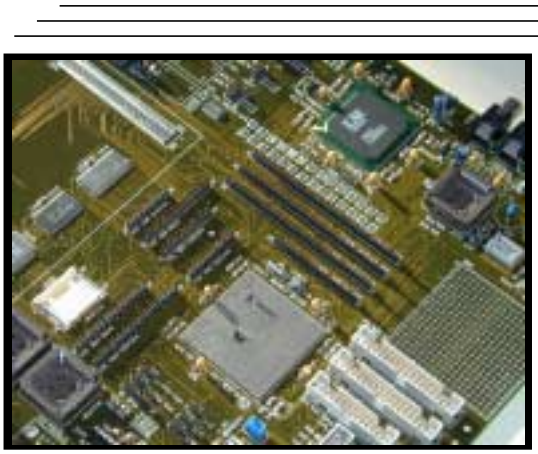
iAH-LCDC

AMBA (AHB) compliant

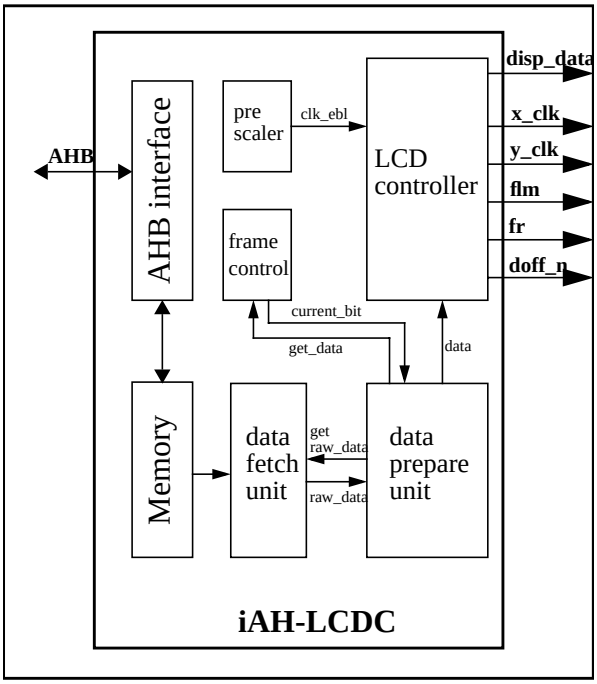
data sheet



Features:

- AHB memory interface for AMBA
- Configurable operating mode:
  B/W or 4 (out of 8)-shade gray scale
- Anti flicker feature for gray scale mode
- Configurable display size
- Display data bus can be configured for 4 or 8bit
- Pixel access
- Display clear command
- Completely independent working
- Needs single port memory only
- Connects to most LCDs of the leading manufacturers

Structure of the iAH-LCDC:



INICORE - the reliable Core and System Provider.
We provide high quality IP, design expertise and leading edge silicon to the industry.

The **iAH-LCDC** is an innovative and easy configurable LCD controller for graphic type LCDs that handles display sizes up to 1024 (W) x 512 (H) pixels. It can be used with most LCDs of the leading manufacturers like Samsung, Hantronix, Kyocera, Sharp and others. Once configured and started the iAH-LCDC works as an independent module in your system and needs no further controlling, giving you more flexibility to take care of the rest of the system.
In addition it supports B/W or gray scale operation. Gray scale is achieved by applying the frame rate control (FRC) technique. A special anti flicker feature lets you set the frame rate individually for odd and even pixels to avoid any flickering.

INICORE's in-depth know how in display controllers and system design makes this module your ideal choice for an easy to use LCD controller.

INICORE offers the structural VHDL iAH-LCDC simulation/synthesis model for the target technology of your choice

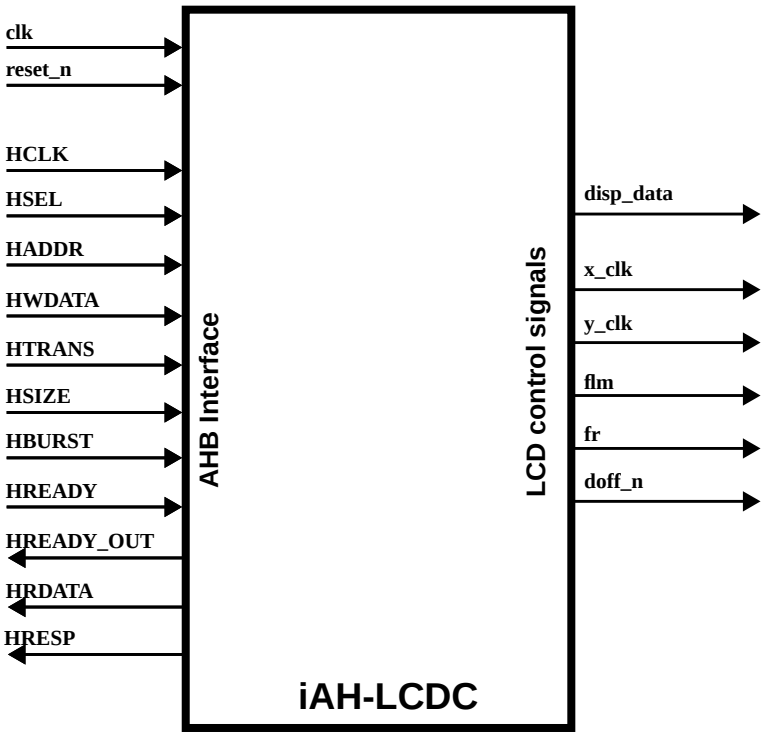
INICORE Inc. (USA)
5600 Mowry School Road, Suite 180,
Newark, CA 94560
Tel: 510 445 1529 Fax: 510 656 0995
E-mail: ask_us@inicore.com
Web: www.inicore.com

INICORE AG (Europe)
Mattenstrasse 6a,CH-2555 Brügg, Switzerland
Tel: ++41 32 374 32 00, Fax: ++41 32 374 32 01
E-mail: ask_us@inicore.ch
Web: www.inicore.ch

Ref.Nr.: V1.0 / 6/8/01

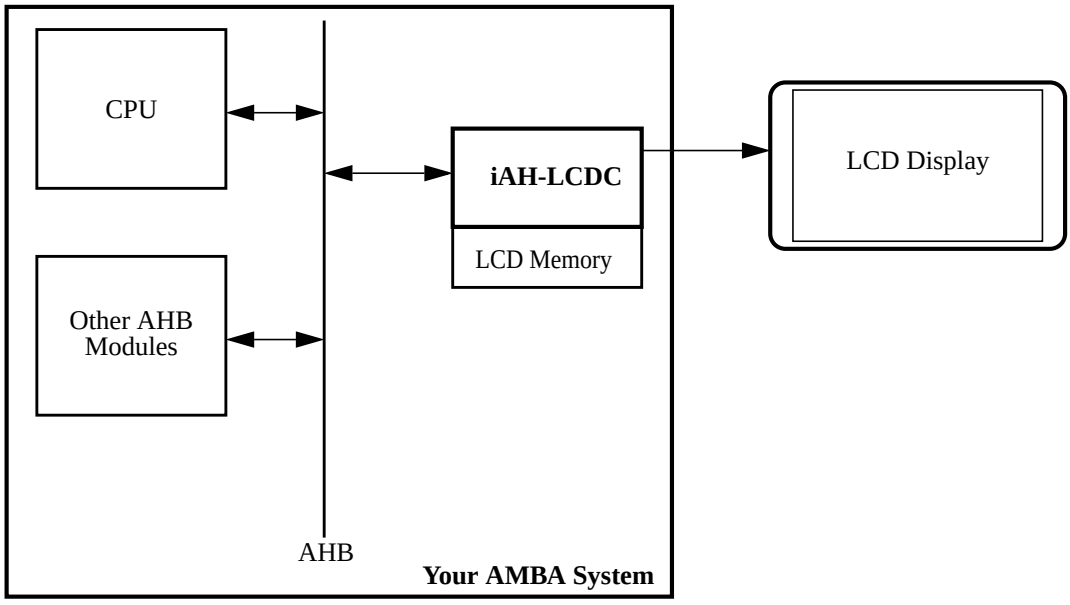
1 Overview

The iAH-LCDC core is a versatile but independent operating LCD controller that gives you the flexibility to use different LCD manufacturers or run it in different modes.



1.1 Typical Application

The following picture shows you a typical application of the iAH-LCDC:



2 IO description

The following part lists the input and output ports of the iAH-LCDC core and gives a short overview of their functionality.

2.1 General inputs

These pins are used to clock and initialize the whole iAH-LCDC core. There are no other clocks in this core.

pin name	type	size	description
clk	in	1	system clock for the LCD controller of the iAH-LCDC
reset_n	in	1	asynchronous system reset, active low

2.2 AHB Interface

These pins are used to connect the iAH-LCDC to the AHB of your system.

pin name	type	size	description
hclk	in	1	AHB clock, used for the interface and memory only. The HCLK may also be used to clock the LCD controller (clk)
hsel	in	1	Select signal for the LCD controller
haddr	in	32	Bus address
hwdata	in	32	Bus write data
htrans	in	2	Transaction type
hsize	in	3	Transaction size
hburst	in	3	Transaction burst desciptor
hready	in	1	hready signal of AHB (after AHB multiplexor)
hready_out	out	1	hready output signal of the LCD controller
hrdata	out	32	Bus read data
hresp	out	2	Bus response (always o.k.)

2.3 LCD Control Signals

These LCD control signals are connected to the LCD driver:

pin name	type	size	description
disp_data	out	8	4 or 8 bit display data depending on configuration (the upper left pixel of the LCD is always the MSB of disp_data, either disp_data[3] or disp_data[7])
x_clk	out	1	x or column clock
y_clk	out	1	y or line clock
flm	out	1	first line marker, is high when first line is displayed
fr	out	1	frame signal, toggles with every new frame
doff_n	out	1	display off signal

3 Functional Description

This chapter explains the functionality of the iAH-LCDC and illustrates it with a few examples. LCD controller is an independent module which reads out the memory at a programmable address and generates data and control signals for the LCD. Various sizes of LCD can be configured, but not all combinations are valid.

3.1 Programming Model

This part describes the registers viewed from the AHB bus. The iAH-LCDC may be placed anywhere in the AHB memory map. The hsel signal is used to mark an access to the LCD controller.

3.1.1 Memory Map

The memory map of the iAH-LCDC is fixed and needs 128kByte address space:

Area / Register	Address	Type	Description
lcd_ctrl	0x00'0000	W	[0]: LCD Operation '0': stop LCD[1] '1': start LCD
		R	[0]: LCD Status '0': LCD controller is idle / stopped '1': LCD controller is running
lcd_x	0x00'0004	R/W	[4:0] X-dimension of the LCD. The real pixel size is: (lcd_x + 1) by 32.
lcd_y	0x00'0008	R/W	[8:0] Y-dimension of the LCD. The real pixel size is: lcd_y + 1

Area / Register	Address	Type	Description
lcd_cfg	0x00'000c	R/W	[3:0]: This is the timing configuration for the LCD control signal generator. It defines the refresh rate and the clock for the LCD. This feature makes the LCD refresh rate independent of the 'clk': 0x0: Fastest clock, clk / 2 0x1: clk / 4 0x2: clk / 6 ... 0xF: clk / 32 [4]: LCD data mode: '0': Use 4bit LCD data output '1': Use 8bit LCD data output [5]: gray scale operation '0': LCD black and white mode '1': LCD grayscale mode [23:16]: Odd pixel frame rate control. This 8 bit register allows to define in which frame which bit of the odd pixels is used. '0': bit0 is displayed '1': bit1 is displayed [31:24]: Even pixel frame rate control. This 8 bit register allows to define in which frame which bit of the even pixels is used. '0': bit0 is displayed '1': bit1 is displayed
lcd_cmd	0x00'0010	W	Command instruction: 0x1: Display clear 0x2: user defined command ...
		R	Busy status: 0x0: Command is executed / controller is ready for next command 0x1: Controller is busy
lcd_pix_access	0x00'0014	R/W	Pixel access: [15:0]: X position of the pixel (0..1023) [29:16]: Y position of the pixel (0..511) [31:30]: Pixel value: b&w mode: [30] ignored, or '0'
LCD Memory	0x01'0000 - 0x01'ffff	R/W	Memory of the LCD controller

1. When the LCD controller is stopped, the LCD is switched off, but the current frame is finished (data is still sent to the LCD) before all control signals go to a defined state and no more requests are issued to the bus controller. The status can be checked.

3.1.2 Programming

The display controller has to be configured first. Then the display controller can be started and the content of the memory is displayed. To modify the image, the processor can write to the LCD memory like to a standard memory. Read access is also supported. Since the LCD memory is a single port architecture, the memory access controller will insert wait-

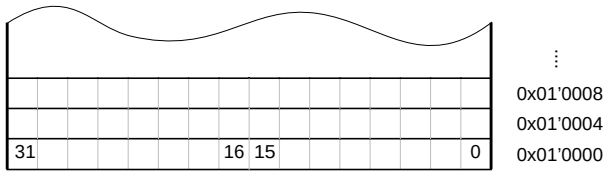
states (maximum 1) when a conflict has been detected. Write cycles will normally take place without wait-states, while read accesses take 1 wait-state. This relaxes the timing on the LCD memory.

The LCD memory can also be accesses pixel by pixel. The register lcd_pix_access can be read or written. The pixel X-address, Y address and pixel value are read / written within one single access. Wait-states are inserted until the pixel has been read / written.

To clear the display, the lcd_cmd register can be used. A write access to this register will execute the corresponding command (e.g. 0x1 for display clear). Then, this register can be polled until the value 0x0 has been read, which signals that the command has been executed. More user specific commands can be added easily.

3.2 Functional Description

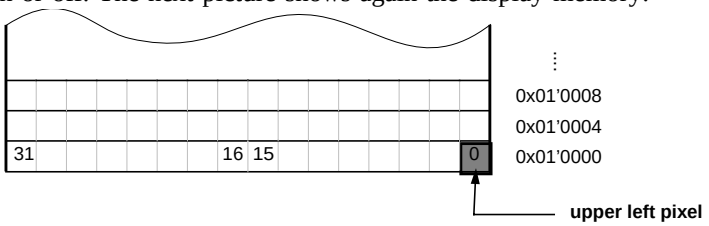
This part shows how the pixels are organized in the LCD memory:



When speaking about bit x, then we are referring to the bit position in the display memory.

3.3 B/W Mode

In B/W mode every bit in the display memory corresponds to a pixel on the display and can be switched on or off. The next picture shows again the display memory:



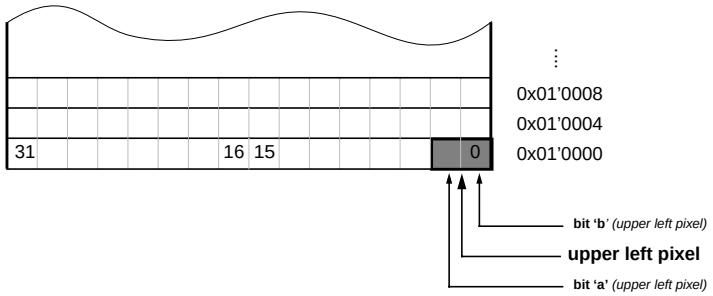
The LCD controller starts at the memory base address 0x01'0000 to fetch word 0, the first 32 pixels, where bit 0 of this word is the upper left pixel. Then, word 1 is fetched, word 2... until it has reached the end of line 0. After that, line 1 is displayed, by fetching word 0 of line 1 at the following address. At the end of a cycle, after displaying the lower right pixel, the LCD controller restarts at 0x01'0000 with the upper left pixel.

3.4 Grayscale Mode

This mode uses 2 consecutive bits per pixel in the display memory. This doubles the size of the needed display memory and doubles as well the number of accesses to this memory. Grayscale mode allows to display 4 (out of 8) different shades of gray including black. This is achieved with an 8-cycle frame rate control system. It also features individually eprogrammable frame cycles for even and odd pixels to avoid flickering. If the desired LCD refresh rate is 70 Hz, that means that the frame rate equals 560 Hz.

The next picture illustrates the display memory for grayscale mode where each pixel needs two consecutive bits:

The principle is the following: The two 8bit values lcd_cfg(31:24) (even pixel frame rate control) and lcd_cfg(23:16) (odd pixel frame rate control) allow to program which of the two bits ('a' or 'b') of which pixel (odd or even) is displayed in which frame (1 to 8).



Depending on the proportion of ‘1’ and ‘0’ in each part of the lcd_cfg register the following shade combinations are possible:

No. of ‘1’	**No. of ‘0’**	**Resulting shade** (provided that each bit ‘b’ = 1 and bit ‘a’ = 0)
0	8	‘white’
1	7	12.5% gray
2	6	25% gray
3	5	37.5% gray
4	4	50% gray
5	3	62.5% gray
6	2	75% gray
7	1	87.5% gray
8	0	‘black’