ADE3800

## Analog LCD Display Engine for XGA and SXGA Resolutions with Embedded LVDS and RSDS Transmitters

## Feature Overview

- Programmable Context Sensitive ${ }^{\text {TM }}$ Filtering
- High-quality Up-scaling and Down-scaling
- Integrated 10-bit Triple Channel ADC/PLL
- IQSync ${ }^{\text {TM }}$ AutoSetup
- Integrated Programmable Timing Controller
- Integrated LVDS Transmitters
- Integrated Pattern Generator
- Perfect Picture ${ }^{\text {TM }}$ Technology
- sRGB 3D Color Warp
- High performance OSD supporting 1- to 4-bpp, proportional fonts
- Advanced EMI reduction features
- Serial ${ }^{2} \mathrm{C}$ interface

■ Low power $0.15 \mu \mathrm{~m}$ process technology

- Low cost 100-pin LQFP and 128-pin LQFP packages
- Lead-free versions available in 2005.


## General Description

ADE3800 devices are a family of highly-integrated display engine ICs, enabling the most advanced, flexible, and cost-effective system-on-chip solutions for analogonly input LCD display applications.

The ADE3800 covers the full range of XGA and SXGA analog-only monitor applications using LVDS or RSDS interface.

The ADE3800 family is software compatible.


## LCD Scaler Product Selector

| Product | Output Format Support |  |  |  |
| :--- | :---: | :---: | :---: | :---: |
|  |  | Resolution | RSDS/TCON | LVDS |
| ADE3800XL |  | Up to XGA 75 Hz |  | Yes |
| ADE3800XT | 100 LQFP | Up to XGA 75 Hz | Yes |  |
| ADE3800SXL | 100 LQFP | Up to SXGA 75 Hz |  | Yes |
| ADE3800SXT | 100 LQFP | Up to SXGA 75 Hz | Yes |  |

## Context Sensitive ${ }^{\text {TM }}$ Scaler

■ Sharper text with Edge Enhancement

- Programmable coefficients for unique customization
- From 5:1 upscale to 2:1 downscale

■ Independent X - Y axis zoom and shrink

## Analog RGB input

■ 140 MHz 10-bit ADC
■ Ultra low jitter digital Line Lock PLL

- Composite Sync and Sync on Green built-in support


## IQsync ${ }^{\text {TM }}$ AutoSetup

■ AutoSetup configures phase, clock, level, and position

- Automatically detects activity on input
- Compatible with all standard VESA and GTF modes


## Perfect Picture ${ }^{\text {TM }}$ Technology

■ Video \& Picture highlight zone
■ Supports up to 4 different windows

- Independent window controls for contrast, brightness and color


## Perfect Color ${ }^{\text {TM }}$ Technology

- Programmable 3D Color Warp

■ Digital brightness, contrast, hue, and saturation gamma controls

- Simple white point control
- Compatible with sRGB standard
- True color dithering for 18 and 24-bit panels
- Temporal and spatial dithering

■ 30-bit programmable gamma table

## OSD Engine

■ 12 KB RAM based $12 \times 18$ characters

- 1, 2, 3, 4-bit per pixel color characters
- Multiple Windows

■ Bordering, shadowing, transparency, fade-in and fade-out effects

■ Supports font rotation

- Up to full screen size, multiple windows

■ 64-entry TrueColor LUT with alpha-blending

## Programmable Timing Controller (TCON)

- Highly programmable support for XGA and SXGA smart panels
- RSDS split line support for SXGA smart panels

■ Supports 18, 24, 36, and 48-bit RSDS outputs

- Advanced Flicker Detection and Reduction
- 8 programmable timing signals for row/column control
- Wide range of drivers \& TCON compatibility


## Integrated LVDS Transmitters

■ Dual 4 channel $6 / 8$ bit LVDS transmitters
■ Programmable channel swapping
■ Programmable channel polarity

- Programmable group channel swapping for flexibility in board layout
■ Programmable output swing control


## Advanced EMI Reduction Features

■ Flexible data transition minimization, single and dual

- Differential clock and signals

■ Spread spectrum - programmable digital FM modulation of the output clock with no external components

## Output Format

■ Supports resolutions up to SXGA @ 75 Hz
■ Supports resolution above SXGA (1280x1024) with convenient input and output pixel clocks
■ Supports 6 or 8-bit Panels
■ Supports single or double pixel wide formats

## Table of Contents

Chapter 1 Important Information .....  6
Chapter 2 General Description ..... 7
Chapter 3 Pin Descriptions ..... 11
Chapter 4 Register Description by Block ..... 18
4.1 Global Control (GLBL) ..... 18
4.2 Frequency Synthesizer (FSYN) ..... 20
4.2.1 Dotclock vs Outclock ..... 20
4.3 Analog-to-Digital Converter (ADC) ..... 21
4.3.1 216 MHz Frequency Synthesizer ..... 22
4.3.2 Sync-on-Green (SOG) ..... 22
4.4 Analog Dithering (ADTH) ..... 29
4.4.1 Function ..... 29
4.4.2 Addressing Technique ..... 29
4.4.3 Output Amplitude Control ..... 29
4.4.4 Miscellaneous ..... 29
4.5 Line Lock PLL (LLK) ..... 30
4.6 Sync Retiming (SRT) ..... 33
4.6.1 Coast Signal ..... 33
4.7 Input Sync Measurement (SMEAS) ..... 36
4.7.1 Input Sync - Activity Detection ..... 36
4.7.2 Input Sync - Measurement ..... 37
4.7.3 Fast Mute ..... 37
4.8 Sync Multiplexer (SMUX) ..... 43
4.9 Data Measurement (DMEAS) ..... 47
4.9.1 Function Summary ..... 47
4.9.2 Window Control ..... 47
4.9.3 Algorithm Control ..... 47
4.9.4 Mode Control ..... 48
4.9.5 Edge Intensity ..... 48
4.9.6 Pixel Sum ..... 48
4.9.7 Min / Max ..... 48
4.9.8 Pixel Cumulative Distribution (PCD) ..... 49
4.9.9 H Position Min / Max ..... 49
4.9.10 V Position Min / Max ..... 50
4.9.11 DE Size ..... 50
4.10 Scale (SCL) ..... 53
4.10.1 Frame Synchronization ..... 54
4.10.2 Context Description ..... 57
4.10.3 Scale Kernel Example ..... 58
4.11 Pattern Generator (PGEN) ..... 59
4.11.1 Overview ..... 59
4.11.2 Color Mask Sequencer ..... 60
4.11.3 $8 \times 8$ Grid Layout with Optional Resets ..... 62
4.11.4 Borders ..... 63
4.11.5 TCON Window Control ..... 64
4.12 sRGB (SRGB) ..... 68
4.12.1 Parametric Gamma, Digital Contrast / Brightness on Multiple Windows ..... 68
4.12.2 Color Space Warp ..... 68
4.13 Gamma (GAM) ..... 71
4.14 On-Screen Display (OSD) ..... 72
4.14.1 Implementation ..... 76
4.14.2 Color LUT Calculation ..... 78
4.14.3 Alpha Blending ..... 78
4.14.4 RAM Memory ..... 81
4.15 Flicker (FLK) ..... 88
4.15.1 Function ..... 88
4.16 Adaptive Phase Control (APC) ..... 92
4.16.1 Function ..... 92
4.16.2 Addressing Technique ..... 92
4.16.3 Dither threshold Control ..... 92
4.17 Output Mux (OMUX) ..... 94
4.17.1 Output Data ..... 98
4.17.2 Output Clocks ..... 101
4.17.3 Clock Sources and Timing Considerations ..... 101
4.18 Timing Controller (TCON) ..... 102
4.19 LVDS/RSDS Features ..... 113
4.19.1 Output Channels ..... 114
4.20 Pulse Width Modulation (PWM) ..... 119
$4.21 \quad I^{2} \mathrm{C}$ Block Transfer (I2CBKT) ..... 120
4.21.1 Transfer Setup and Start ..... 120
4.21.2 Transfer Progress ..... 120
4.21.3 Concurrent I2C Transfers ..... 120
$4.22 \quad \mathrm{I}^{2} \mathrm{C}$ Registers and RAM Addresses ..... 122
4.22.1 I2C Transfer Format ..... 122
4.22.2 Dedicated RAM Areas per Block ..... 122
4.22.3 Multi-byte Registers ..... 122
Chapter 5 Electrical Specifications ..... 124
5.1 Absolute Maximum Ratings ..... 124
5.2 Nominal Operating Conditions ..... 124
5.3 Preliminary Thermal Data ..... 126
5.4 Preliminary DC Specifications ..... 126
5.4.1 LVTTL 5-Volt Tolerant Inputs with Hysteresis ..... 126
5.4.2 LVTTL 3-Volt Tolerant Inputs with Hysteresis ..... 126
5.4.3 LVTTL 5-Volt Tolerant I/O with Hysteresis ..... 126
5.4.4 LVTTL 3-Volt Tolerant I/O ..... 127
5.4.5 LVTTL 3-Volt Tolerant I/O ..... 127
5.5 LVDS Outputs ..... 127
5.6 RSDS Outputs ..... 127
5.7 ADC Characteristics ..... 128
Chapter 6 Package Mechanical Data ..... 129
6.1 100 Pin LQFP ..... 129
6.2128 Pin LQFP ..... 130
Chapter 7 Scaler Equations ..... 132
Chapter 8 ADE3800 vs ADE3700 ..... 133
Chapter 9 Glossary ..... 136
Chapter 10 Revision History ..... 137

## 1 Important Information

- XCLK: Crystal oscillator, usually 27 MHz .
- INCLK: ADC Sampling clock frequency, depends on input video mode pixel rate.
- DOTCLK and OUTCLK: Related to Panel Output Pixel Rate.
- SCLK: Scale Clock used for the line buffer Ram and picture zooming.
- If some bit fields are missing, these bits are marked as "reserved":
- return 0 when read, but it is also the user's duty to mask them upon readout, to ensure compatibility with later device releases
- must be written to 0 when the whole register is written
in all cases, the default reset value always prevails
- An asterisk denotes the default reset value for the corresponding bit(s).
- Unless all addresses and registers values are in hexadecimal.
- "not sticky" means dynamically updated (set or reset) by hardware, not a static bit.
- A "sticky" bit, once set remains set until the user clears it.
- When a value is followed by "typ" this means it is a typical value and PVT dependent.
- If a time or delay value does not have "min/typ/max" information, it is proportional to the XCLK frequency.
- Any register names containing HW are shadow registers: they report which value is currently being used by the chip.
- When a register bit field list has one bold option, it is the only choice for normal mode of operation.
- TCON must always be programmed for any panel type.
- Values spread out over several registers are organised as follows:

| 32-bit values |  | 24-bit values |  | 16-bit values |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| -0 | LSB | _L or _0 | LSB | _L | LSB |
| -1 |  | _M or_1 | MSB | _U | USB |
| -2 |  | _U or _2 | USB |  |  |
| _3 | USB |  |  |  |  |

## 2 General Description

The ADE3800 family of devices is capable of implementing all of the advanced features of todays LCD monitor products. For maximum flexibility, an external microcontroller (MCU) is used for controlling the ADE3800 and other monitor functions.

Figure 1: ADE3800 Block Diagram


The ADE3800 architecture unburdens the MCU from all data-intensive pixel manipulations, providing an optimal blend of features and code customizing without incurring the cost of a 16-bit processor or memory. The key interactions between the monitor MCU and the ADE3800 can be broken down into the features shown in Table 1.

Table 1: ADE3800 Features (Sheet 1 of 2)

| Feature | Description of ADE3800 Operation | Blocks Used | Pages |
| :---: | :---: | :---: | :---: |
| Power-up / Initialize | When power is first applied, the ADE3800 is asynchronously reset from a pin. The MCU typically programs the ADE3800 with a number of default values and sets up the ADE3800 to identify activity on any of the input pins. All preconfigured values and RAMs, such as line-lock PLL settings, OSD characters, LCD timing values (output sequencer), scale kernels, gamma curves, sRGB color warp, APC dithering, output pin configuration (OMUX), etc. can be preloaded into the ADE3800. The typical end state is that the ADE3800 is initialized into a low power mode, ready to turn active once the power button is pressed. | GLBL <br> SMEAS <br> LLK <br> ADC <br> OSD <br> SCALER <br> GAMMA <br> SRGB <br> TCON <br> APC <br> OMUX | $\begin{aligned} & \hline 18 \\ & 36 \\ & 30 \\ & 21 \\ & 72 \\ & 53 \\ & 71 \\ & 68 \\ & 68 \\ & 102 \\ & 92 \\ & 94 \end{aligned}$ |
| Activity Detect | When the monitor has been powered on, the inputs can be monitored for active video sources. Based on the activity monitors, the MCU chooses an input or power down state. | SMEAS | 36 |

Table 1: ADE3800 Features (Sheet 2 of 2)

| Feature | Description of ADE3800 Operation | Blocks Used | Pages |
| :---: | :---: | :---: | :---: |
| Sync / Timing Measurement | Once an input source is selected, all available information on frequencies and line/pixel counts is measured for the selected source and made available to the MCU. | SMEAS | 36 |
| Mode Set | Once the MCU has determined the matching video mode or calculated a video mode using a GTF algorithm, the datapath is programmed to drive the flat panel. Clock frequencies for the internal memory and datapath are also set at this time. | GLBL <br> LLK <br> SRT <br> SMUX <br> SCALER | $\begin{aligned} & 18 \\ & 30 \\ & 33 \\ & 43 \\ & 53 \end{aligned}$ |
| Autotune | When the MCU calls for an autotune, the MCU sets up an iterative loop to search for the best phase, gain, offset, etc. At each step of the loop, the MCU kicks off a test in which the ADE3800 performs extensive statistical analysis of the incoming data stream. The results of the analysis are made available to the MCU which is responsible for the optimization algorithm. | DMEAS <br> LLK <br> ADC <br> SMUX <br> SRT | $\begin{aligned} & 47 \\ & 30 \\ & 21 \\ & 43 \\ & 33 \end{aligned}$ |
| Digital Contrast / Brightness | In response to user OSD control, the MCU can program single 8-bit registers that set brightness and contrast for each color channel independently. | SRGB | 68 |
| White Point Control | In response to user OSD control, the MCU can program three 8-bit registers that set the white point for the output. | SRGB | 68 |
| GAMMA Adjustment | The MCU can program the gamma RAMs to implement 10-bit accurate color transformations to match the panel color characteristics. | GAMMA | 71 |
| sRGB Control | Allows simple, intuitive color control for parametric gamma correction and 3D color cube warping. | SRGB | 68 |
| Pattern Generation | For production testing, the ADE3800 can be programmed by the MCU to output a wide set of test patterns. | PGEN | 59 |
| Flicker Reduction | For Smart Panel applications, the MCU can set up the flicker detection block to report any correlation with the polarity inversion signal. The MCU can then change the polarity inversion to a non-correlating pattern to eliminate flicker. | $\begin{aligned} & \text { FLICKER } \\ & \text { TCON } \end{aligned}$ | $\begin{array}{\|l\|} \hline 88 \\ 102 \end{array}$ |
| Backlight Control | The ADE3800 provides two PWM outputs for direct control of the power components in a typical backlight. The MCU sets up the registers and enables the function. | PWM | 119 |
| Low Power State | To enter a low power state, the MCU can gate off most of the clocks and put the analog blocks into a low power standby state. | GLBL | 18 |

The following table gives a brief description of each block of the ADE3800:
Table 2: ADE3800 Block Descriptions

| Block |  |
| :--- | :--- |
| Global Control (GLBL) | Responsible for selecting clock sources, power control, I2C control and block by block <br> synchronous reset generation |
| Frequency Synthesizer (FSYN) | Generates the output clock (also known as the dot clock \& DCLK) and the scaler clock <br> (SCLK). Frequency modulation, phase control, and pulse extension (duty cycle control) of <br> the output clock are also provided. |
| Analog-to-Digital Converter (ADC) | Has the following features: <br> - Supports input clocks up to 140MHz (SXGA 75Hz) <br> - Adjustable analog amplifier bandwidth <br> - Differential RGB input path for noise immunity <br> - Built-in Sync-on-Green support |
| - Individual RGB clock delay control |  |
| - Power down control |  |
| Analog Dithering (ADTH) | - Linear and independent Gain/Offset adjustment |

Table 2: ADE3800 Block Descriptions

| Block | Description |
| :--- | :--- |
| Output Mux (OMUX) | An extension of the ADE3700 output mux block. The major changes are: <br> - LVDS controls <br> - RSDS split line buffer |
| Timing Controller (TCON) | Provides timing for Smart Panel applications and other applications that are sensitive to <br> output synchronization timing. The timing unit is based on horizontal and vertical counters, <br> which are locked with the output video stream. |
| LVDS/RSDS Features | Has the following features: <br> - Power down <br> - Output swing and common mode programmable |
| - Individual channel programmable delay |  |
| - Pulse Width Modulation (PWM) | Generates two signals that can be used to control backlight inverter switching power <br> components directly. It is derived from XCLK and can be powered up independently of the <br> DOTCLK and INCLK domains. |
| $1^{2} \mathrm{C}$ Block Transfer (I2CBKT) | Allows the internal I2 C parallel bus to be driven by an xclk state machine to perform rapid <br> block transfers between internal addresses. |
| $1^{2} \mathrm{C}$ Registers and RAM Addresses | Memory mapping of all RAM and register locations accessible by ${ }^{2} \mathrm{C}$ ( C. |

## 3 Pin Descriptions

Figure 2: LQFP100 Pinout Diagram


Figure 3: LQFP128 Pinout Diagram


Table 3: Analog Input Signals (Sheet 1 of 2)

| LQFP100 | LQFP128 | Name | Input/ <br> Output | Description |
| :---: | :---: | :--- | :---: | :--- |
| 43 | 53 | INR+ | I | Positive ADC Red Channel Input |
| 44 | 54 | INR- | I | Negative ADC Red Channel Input |

Table 3: Analog Input Signals (Sheet 2 of 2)

| LQFP100 | LQFP128 | Name | Input/ <br> Output | Description |
| :---: | :---: | :--- | :---: | :--- |
| 38 | 48 | ING+ | I | Positive ADC Green Channel Input |
| 39 | 49 | ING- | I | Negative ADC Green Channel Input |
| 33 | 43 | INB+ | I | Positive ADC Blue Channel Input |
| 34 | 44 | INB- | I | Negative ADC Blue Channel Input |
| 29 | 38 | VSYNC | I | Vertical Sync Input Signal |
| 30 | 39 | HSYNC | I | Horizontal Sync or Composite Sync Input Signal |

Table 4: Output Signals and TCON Signals (Sheet 1 of 2)

| LQFP100 | LQFP128 | Name | Input/ Output | Description |
| :---: | :---: | :---: | :---: | :---: |
|  | 25 | RSDS0- | 0 | RSDS Channel 0 Data - |
|  | 26 | RSDS0+ | 0 | RSDS Channel 0 Data + |
|  | 23 | RSDS1- | 0 | RSDS Channel 1 Data - |
|  | 24 | RSDS1+ | 0 | RSDS Channel 1 Data + |
|  | 21 | RSDS2- | 0 | RSDS Channel 2 Data - |
|  | 22 | RSDS2+ | 0 | RSDS Channel 2 Data + |
| 16 | 19 | RSDS3- | 0 | RSDS Channel 3 Data - |
| 17 | 20 | RSDS3+ | 0 | RSDS Channel 3 Data + |
|  | 14 | RSDS4- | 0 | RSDS Channel 4 Data - |
|  | 15 | RSDS4+ | 0 | RSDS Channel 4 Data + |
| 11 | 12 | RSDS5- | 0 | RSDS Channel 5 Data - |
| 12 | 13 | RSDS5+ | 0 | RSDS Channel 5 Data + |
|  | 10 | RSDS6- | 0 | RSDS Channel 6 Data - |
|  | 11 | RSDS6+ | 0 | RSDS Channel 6 Data + |
| 9 | 8 | RSDS7- | 0 | RSDS Channel 7 Data - |
| 10 | 9 | RSDS7+ | 0 | RSDS Channel 7 Data + |
|  | 81 | RSDS8- | 0 | RSDS Channel 8 Data - |
|  | 82 | RSDS8+ | 0 | RSDS Channel 8 Data + |
|  | 83 | RSDS9- | 0 | RSDS Channel 9 Data - |
|  | 84 | RSDS9+ | 0 | RSDS Channel 9 Data + |
|  | 85 | RSDS10- | 0 | RSDS Channel 10 Data - |
|  | 86 | RSDS10+ | 0 | RSDS Channel 10 Data + |
|  | 87 | RSDS11- | 0 | RSDS Channel 11 Data - |
|  | 88 | RSDS11+ | 0 | RSDS Channel 11 Data + |
|  | 92 | RSDS12- | 0 | RSDS Front Side CLK- |
|  | 93 | RSDS12+ | 0 | RSDS Front Side CLK+ |
|  | 97 | RSDS13- | 0 | RSDS Channel 13 Data- |
|  | 98 | RSDS13+ | 0 | RSDS Channel 13 Data+ |

Table 4: Output Signals and TCON Signals (Sheet 2 of 2)

| LQFP100 | LQFP128 | Name | Input/ Output | Description |
| :---: | :---: | :---: | :---: | :---: |
|  | 99 | RSDS14- | O | RSDS Channel 14 Data- |
|  | 100 | RSDS14+ | 0 | RSDS Channel 14 Data+ |
|  | 101 | RSDS15- | 0 | RSDS Channel 15 Data- |
|  | 102 | RSDS15+ | 0 | RSDS Channel 15 Data+ |
| 98 | 125 | OUTO+ | 0 | +LVDS Channel 0 Differential Data Output or RSDS Channel 16 Data + |
| 99 | 126 | OUT0- | 0 | -LVDS Channel 0 Differential Data Output or RSDS Channel 16 Data - |
| 96 | 123 | OUT1+ | 0 | +LVDS Channel 1 Differential Data Output or RSDS channel 17 Data + |
| 97 | 124 | OUT1- | 0 | -LVDS Channel 1 Differential Data Output or RSDS Channel 17 Data - |
| 94 | 121 | OUT2+ | 0 | +LVDS Channel 2 Differential Data Output or RSDS Channel 18 Data + |
| 95 | 122 | OUT2- | 0 | -LVDS Channel 2 Differential Data Output or RSDS Channel 18 Data - |
| 90 | 117 | OUT3+ | 0 | +LVDS Channel 3 Differential Data Output or RSDS Channel 19 Data + |
| 91 | 118 | OUT3- | 0 | -LVDS Channel 3 Differential Data Output or RSDS Channel 19 Data - |
| 85 | 112 | OUT4+ | 0 | +LVDS Channel 4 Differential Data Output or RSDS Back Side CLK- |
| 86 | 113 | OUT4- | 0 | -LVDS Channel 4 Differential Data Output or RSDS Back Side CLK+ |
| 83 | 110 | OUT5+ | 0 | +LVDS Channel 5 Differential Data Output or RSDS Channel 22 Data + |
| 84 | 111 | OUT5- | 0 | -LVDS Channel 5 Differential Data Output or RSDS Channel 22 Data - |
| 81 | 108 | OUT6+ | 0 | +LVDS Channel 6 Differential Data Output or RSDS Channel 23 Data + |
| 82 | 109 | OUT6- | 0 | -LVDS Channel 6 Differential Data Output or RSDS Channel 23 Data - |
| 77 | 104 | OUT7+ | 0 | +LVDS Channel 7 Differential Data Output or RSDS Channel 24 Data + |
| 78 | 105 | OUT7- | 0 | -LVDS Channel 7 Differential Data Output or RSDS Channel 24 Data - |
| 92 | 119 | OUTCLK0+ | 0 | +LVDS Channel A Differential Clock Output or RSDS Channel 20 Data + |
| 93 | 120 | OUTCLK0- | 0 | -LVDS Channel A Differential Clock Output or RSDS Channel 20 Data - |
| 79 | 106 | OUTCLK1+ | 0 | +LVDS Channel B Differential Clock Output or RSDS Channel 25 Data + |
| 80 | 107 | OUTCLK1- | 0 | -LVDS Channel B Differential Clock Output or RSDS Channel 25 Data - |
| 58 | 70 | TCON0 | 0 | TCON Output 0 or PWM B Output |
| 59 | 71 | TCON1 | 0 | TCON Output 1 or PWM A Output |
| 60 | 72 | TCON2 | 0 | TCON Output 2 |
| 61 | 73 | TCON3 | 0 | TCON Output 3 |
| 62 | 74 | TCON4 | 0 | TCON Output 4 |
| 63 | 75 | TCON5 | 0 | TCON Output 5 |
| 64 | 76 | TCON6 | 0 | TCON Output 6 |
| 65 | 77 | TCON7 | 0 | TCON Output 7 |

Table 5: System Controls (Sheet 1 of 2)

| LQFP100 | LQFP128 | Name | Input/ <br> Output | Description |
| :---: | :---: | :---: | :---: | :--- |
| 47 | 59 | XTAL_OUT | O | Crystal Oscillator output |
| 48 | 60 | XTAL_IN | I | Crystal Oscillator input |

Table 5: System Controls (Sheet 2 of 2)

| LQFP100 | LQFP128 | Name | Input/ Output | Description |
| :---: | :---: | :---: | :---: | :---: |
| 23 | 32 | XCLK | I/O | Crystal clock buffered output. Controlled by XCLK_EN pin |
| 28 | 37 | XCLK_EN | I | Crystal clock output enable. <br> When connected to 3.3 V , the XCLK output is active <br> When connected to Ground, the XCLK output is disabled |
| 25 | 34 | RESETN2 | 1 | Reset 2 input ${ }^{\text {a }}$. Active Low |
| 53 | 65 | RESETN | I | Reset input ${ }^{1}$. Active Low |
| 54 | 66 | SDA | I/O | I2C Data ${ }^{\text {b }}$. Open drain |
| 56 | 68 | SCL/CSYNC | I | I2C Clock $^{\text {c }}$ or Composite Sync Input Signal |
| 26 | 35 | SDA2 | I/O | I2C 2 Data $^{2}$. Open drain |
| 27 | 36 | $\begin{aligned} & \text { SCL2/EXT_SOG/ } \\ & \text { CSYNC } \end{aligned}$ | I | I2C 2 Clock $^{3}$ or Composite Sync Input Signal |
| 24 | 33 | TST_SCAN | 1 | Reserved for test. Should be connected to Digital Ground |

a. RESETN and RESETN2 pins are ORed together internally. The pin which is not used must be connected to ground.
b. The SDA and SDA2 pins share the same internal bi-directional control. The pin that is not used reverts as output and must be left floating or connected to a pull-up resistor.
c. This device has two RESET/I2C ports (RESETN/SCL/SDA or RESETN2/EXT_SOG/SDA2) to facilitate PCB layout. The state of the two RESET pins determines which RESET/I2C port is active. The RESET pin that is held in the low state disables that RESET/I2C port for normal RESET/I2C operations. However, the disabled ports SCL input (either SCL or EXT_SOG) can be used as a CSYNC input from an external CSYNC extractor. If this CSYNC input is not required, then the unused SCL pin should be connected to ground

Table 6: Digital Section Power Supply Pins (Sheet 1 of 2)

| LQFP100 | LQFP128 | Name | Description |
| :---: | :---: | :--- | :--- |
| 5 | 4 | DVDD18 | Digital 1.8V Supply |
| 6 | 5 | DVDD18 | Digital 1.8V Supply |
| 7 | 6 | DGND | Digital Ground |
| 8 | 7 | DGND | Digital Ground |
| 18 | 27 | DVDD18 | Digital 1.8V Supply |
| 19 | 28 | DVDD18 | Digital 1.8V Supply |
| 20 | 29 | DGND | Digital Ground |
| 21 | 30 | DGND | Digital Ground |
| 22 | 31 | DVDD33 | Digital 3.3V Supply |
| 55 | 67 | DGND | Digital Ground |
| 57 | 69 | DVDD33 | Digital 3.3V Supply |
| 66 | 78 | DGND | Digital Ground |
| 67 |  | DGND | Digital Ground |
| 68 | 79 | DVDD18 | Digital 1.8V Supply |
| 69 | 80 | DVDD18 | Digital 1.8V Supply |

Table 6: Digital Section Power Supply Pins (Sheet 2 of 2)

| LQFP100 | LQFP128 | Name | Description |
| :---: | :---: | :--- | :--- |
| 70 |  | DVDD18 | Digital 1.8V Supply |
| 72 | 94 | DGND | Digital Ground |
| 73 | 95 | DGND | Digital Ground |
| 74 | 96 | DVDD18 | Digital 1.8V Supply |
| 75 |  | DVDD18 | Digital 1.8V Supply |

Table 7: Analog Section Power Supply Pins

| LQFP100 | LQFP128 | Name |  |
| :---: | :---: | :--- | :--- |
| 31 | 41 | AVDD | Analog 1.8V Supply |
| 35 | 45 | AVDD | Analog 1.8V Supply |
| 36 | 46 | AVDD | Analog 1.8V Supply |
| 40 | 50 | AVDD | Analog 1.8V Supply |
| 32 | 42 | AGND | Analog Ground |
| 37 | 47 | AGND | Analog Ground |
| 41 | 51 | AGND | Analog Ground |
| 42 | 52 | AGND | Analog Ground |
| 45 | 55 | AVDD | Analog 1.8V Supply |
| 46 | 58 | XGND | Crystal Oscillator Ground |
| 49 | 61 | XVDD18 | Crystal Oscillator 1.8V Supply |
| 50 | 62 | PGND | PLL Ground |
| 51 | 63 | PVDD18 | PLL 1.8V Supply |
| 52 | 64 | PGND | PLL Ground |

Table 8: Output Section Power Supply Pins (Sheet 1 of 2)

| LQFP100 | LQFP128 | Name | Description |
| :---: | :---: | :--- | :--- |
| 2 | 1 | PLLVDD18 | Output PLL 1.8V Supply |
| 3 | 2 | SGND | Output PLL Ground. Should be connected to Output Ground |
| 4 | 3 | OVDD18 | Output Multiplexer 1.8V Supply |
| 13 | 16 | OVDD18 | Output Multiplexer 1.8V Supply |
| 14 | 17 | VRH | LVDS/RSDS reference voltage. Connect to external capacitor to ground |
| 15 | 18 | VRL | LVDS/RSDS reference voltage. Connect to external capacitor to ground |
|  | 40 | EPGND | Exposed Pad Ground. Connect to Output Ground |
|  | 56 | EPGND | Exposed Pad Ground. Connect to Output Ground |
| 71 | 57 | EPGND | Exposed Pad Ground. Connect to Output Ground |
| 76 | 103 | OVDD18 | Exposed Pad Ground. Connect to Output Ground |
| 87 | 115 | VRL | Output Multiplexer 1.8V Supply |
| 88 | 114 | VRH | LVDS/RSDS reference voltage. Connect to external capacitor to ground |

Table 8: Output Section Power Supply Pins (Sheet 2 of 2)

| LQFP100 | LQFP128 | Name | Description |
| :---: | :---: | :--- | :--- |
| 89 | 116 | OVDD18 | Output Multiplexer 1.8V Supply |
|  | 91 | OVDD18 | Output Multiplexer 1.8V Supply |
|  | 89 | VRL | LVDS/RSDS reference voltage. Connect to external capacitor to ground |
|  | 90 | VRH | LVDS/RSDS reference voltage. Connect to external capacitor to ground |
| 100 | 127 | OVDD18 | Output Multiplexer 1.8V Supply |
| 1 | 128 | PLLVDD18 | Output PLL 1.8V Supply |

## 4 Register Description by Block

### 4.1 Global Control (GLBL)

The Global Control block is responsible for:

- Selecting clock sources
- Power control
- ${ }^{2} \mathrm{C}$ control
- Block by block synchronous reset generation.

The global control block runs in the crystal clock (XCLK) domain, which is required to be active for programming. In general for all ADE3800 blocks, $I^{2} \mathrm{C}$ register access operates in the XCLK domain; exceptions are the internal RAMS which require the appropriate clock domain to be active (e.g. dotclk for OSD RAMs), refer to Table 44.

Table 9: Global Control Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| GLBL_REV_ID | 0000 | R | [7:0] | 0x83 | REV_ID: Chip Revision ID |
| GLBL_CLK_SRC_SEL_0 | 0001 | R/W | [6:4] | 00 | DOTCLK_SRC_SEL: DOTCLK source select <br> 0: Crystal Clock <br> 1: XCLK pin (test only) <br> 2: FM freq synth half speed $(1 \mathrm{ppc})^{\text {a }}$ <br> 3: FM freq synth full speed $(2 \mathrm{ppc})^{1}$ <br> 4: SCLK frequency synthesizer <br> 5-7: Reserved |
|  |  |  | [2:0] |  | INCLK_SRC_SEL: input clock source select <br> 0: Crystal Clock <br> 1: XCLK pin (test only) <br> 2: LLPLL phase controlled SRC (normal) <br> 3: LLPLL fixed phase clock (test only) <br> 4: LLPLL control clock (test only) <br> 5-7: Reserved |
| GLBL_CLK_SRC_SEL_1 | 0002 | R/W | [6:4] | 00 | OUTCLK_SRC_SEL: panel output clock source select <br> 0: Crystal Clock <br> 1: XCLK pin (test only) <br> 2: FM freq synth half speed $(1 \mathrm{ppc})^{\mathrm{b}}$ <br> 3: FM freq synth full speed $(2 \mathrm{ppc})^{1}$ <br> 4: SCLK frequency synthesizer <br> 5-7: Reserved |
|  |  |  | [2:0] |  | SCLK_SRC_SEL: scaler clock source select <br> 0: crystal clock <br> 1: XCLK pin (test only) <br> 2: FM freq synth half speed <br> 3: FM freq synth full speed <br> 4: Fixed freq synth (normal) <br> 5: LVDS pll output (test only) <br> 6: LVDS pll input (test only) <br> 7: Reserved |

Table 9: Global Control Registers (Sheet 2 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| GLBL_CLK_INV | 0003 | R/W | [4] | 00 | AFE_CLK_INV: invert ADC sample clock |
|  |  |  | [3] |  | OUTCLK_INV: invert output clock |
|  |  |  | [2] |  | SCLK_INV: invert SCLK |
|  |  |  | [1] |  | DOTCLK_INV: invert DOTCLK |
|  |  |  | [0] |  | INCLK_INV: invert INCLK |
| GLBL_CLK_ENAB_0 | 0004 | R/W | [7] | FF | DOTCLK_FLK_EN: enable DOTCLK to the FLK block |
|  |  |  | [6] |  | DOTCLK_OSD_EN: enable DOTCLK to the OSD block |
|  |  |  | [5] |  | DOTCLK_PGEN_EN: enable DOTCLK to the PGEN block |
|  |  |  | [4] |  | DOTCLK_EN: enable DOTCLK upstream of FLK, OSD, and PGEN enable |
|  |  |  | [3] |  | INCLK_DFT_EN: enable INCLK to DFT test circuits |
|  |  |  | [2] |  | INCLK_DMEAS_EN: enable INCLK to DMEAS block |
|  |  |  | [1] |  | INCLK_EN: enable INCLK upstream of DMEAS and DFT enable |
|  |  |  | [0] |  | ALL_VIDEO_CLK_EN: override block enable (FLK, OSD, PGEN, DFT, DMEAS) for test |
| GLBL_CLK_ENAB_1 | 0005 | R/W | [1] | 03 | OUTCLK_EN: enable output clock |
|  |  |  | [0] |  | SCLK_EN: enable scaler clock |
| GLBL_SRST_0 | 0006 | R/W | [7] | 00 | TCON_SRST: reset the TCON block |
|  |  |  | [6] |  | SCL_SRST: reset the SCALER block |
|  |  |  | [5] |  | SMUX_SRST: reset the SMUX block |
|  |  |  | [4] |  | DMEAS_SRST: reset the DMEAS block |
|  |  |  | [3] |  | SMEAS_SRST: reset the SMEAS block |
|  |  |  | [2] |  | SRT_SRST: reset the SRT block |
|  |  |  | [1] |  | ADTH_SRST: reset the ADTH block |
|  |  |  | [0] |  | ADC_SRST: reset the digital logic in the ADC block |
| GLBL_SRST_1 | 0007 | R/W | [7] | 00 | DFT_SRST: reset DFT (test) circuits |
|  |  |  | [6] |  | OMUX_SRST: reset the OMUX block |
|  |  |  | [5] |  | APC_SRST: reset the APC block |
|  |  |  | [3] |  | OSD_SRST: reset the OSD block |
|  |  |  | [1] |  | PGEN_SRST: reset the PGEN block |
|  |  |  | [0] |  | OSQ_SRST: reset the OSQ portion of the SCALER block |
| GLBL_I2C_CTRL | 0008 | R/W | [2] | 00 | I2C_AUTO_INC_OFF: disable I2C autoincrement |
|  |  |  | [1] |  | I2C_SDA_PMOS_ON: SDA PMOS enable ${ }^{\text {c }}$ |
|  |  |  | [0] |  | BYPASS_I2C_FILTER: bypass antiglitch filter |
| GLBL_BPAD_EN | 0009 | R/W | [7:0] | 03 | Reserved |
| GLBL_COMP_CTRL | 000A | R/W | [0] | 01 | COMPEN_EN: enable slew-rate compensation |
| GLBL_XTAL_CTRL | 000B | R/W | [0] | 01 | I2C_MUXA_XTAL_EN: enable the crystal oscillator ${ }^{\text {d }}$ |
| GLBL_TST_CTRL | 000C | R/W | [7:0] | 00 | Reserved |

Table 9: Global Control Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| GLBL_AZWC_CTRL | 000F | R/W | [7:2] | 0 | Reserved |
|  |  |  | [1] | 0 | Auto Zero Window Control and Clamp synchronization <br> 0: Synchronization on INCLK <br> 1: Synchronization on DOTCLK |
|  |  |  | [0] | 0 | Reserved |
| DFT_DEL_REF | OFOB | R | [7:0] |  | Returns chip speed and gate propagation delay (number of gates propagation per XCLK period) |

a. Refer to OMUX_CTRLO[0] and also to Table 12.
b. Refer to OMUX_CTRLO[0] and also to Table 12.
c. If set, this bit puts the SDA output in push-pull mode (instead of open drain) to achieve higher ${ }^{12} \mathrm{C}$ speed.
d. If reset, the device is put in shutdown mode (lowest possible power consumption) but can only exit from that mode with an external reset or a power on/off.

### 4.2 Frequency Synthesizer (FSYN)

The Frequency Synthesizer block generates the output clock, the dot clock and the scaler clock (SCLK). Frequency modulation, phase control, and pulse extension (duty cycle control) of the output clock are also provided.
For consistency and ease of use, both clocks are programmed by means of a single-parameter the phase rate value derived from the desired frequency.

### 4.2.1 Dotclock vs Outclock

Dot clock (also known as DOTCLK or DCLK) is an internal clock; there are no associated I2C registers.
Out clock is the pixel clock that drives the LCD panel:

- When driving 2 pixels per clock, out clock and dot clock are identical
- When driving 1 pixel per clock the out clock frequency is half the dot clock frequency (phase rate is proportional to clock period which is the inverse of frequency).
Refer to Table 12: Clock Relationship.
Table 10: FSYN Frequency Synthesizer Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| FSYN_CTRL | 0850 | R/W | $[0]$ | 00 | frequency modulation <br> $0^{*}:$ off <br> $1:$ on |
| FSYN_PR_OTCLK_0 | 0851 | R/W | $[7: 0]$ | 00 | output clock phase rate <br> = $2^{\wedge} 21 *$ XCLK_FREQ / OUT_CLK_FREQ |
| FSYN_PR_OTCLK_1 | 0852 | R/W | $[7: 0]$ | 00 | [5:0] |
| FSYN_PR_OTCLK_2 | 0853 | R/W | [7:0] | 00 | RSDS clock-data skewcontrol (no meaning in LVDS) <br> LSB = 289ps |
| FSYN_OFFSET | 0854 | R/W |  |  |  |

Table 10: FSYN Frequency Synthesizer Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| FSYN_FM_AMPLITUDE | 0855 | R/W | $[7: 0]$ | 00 | frequency modulation amplitude <br> LSB $=4.5 p s$ |
| FSYN_FM_PERIODX64 | 0856 | R/W | $[7: 0]$ | 80 | frequency modulation period <br> LSB $=1.184$ us |
| FSYN_PULSE_HIGH_EXT | 0857 | R/W | $[7]$ | 00 | enable pulse extend <br> $0^{*}:$ disabled <br> $1:$ enabled |
|  |  | R/W | [2:0] |  | lise extend value <br> LSB $=0.3 n s ~(t y p) ~$ |

Table 11: FSYN_PR_SK Registers

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| FSYN_PR_SK_0 | 0860 | R/W | $[7: 0]$ | 00 | sclk phase rate <br> $=2^{\wedge 115 ~ * ~ x c l k \_f r e q ~ / ~ s c l k \_f r e q ~}$ <br> Set sclk $=140 \mathrm{MHz}$ <br> i.e. FSYN_PR_SK_1/0 $=18 A F h ~ @ ~ x c l k ~$ |
| FSYN_PR_SK_1 | 0861 | R/W | $[7: 0]$ | 00 |  |

Table 12: Clock Relationship

|  | 1 ppc | 2 ppc |
| :--- | :--- | :--- |
| FSYN_OUTCLK_FREQ | $2 x$ DOTCLK_FREQ | DOTCLK_FREQ |
| DOTCLK SOURCE SEL | FSYN_OUTCLK_DIV2 (half speed) | FSYN_OUTCLK (full speed) |
| GLBL_CLK_SRC_SEL_0[6:4] | 2 | 3 |
| GLBL_CLK_SRC_SEL_1[6:4] | 3 | 3 |
| FSYN_PR_OTCLK | $22^{\wedge} 1^{*}$ XCLK_FREQ / 2x DOTCLK_FREQ | $22^{\wedge 21 ~ * ~ X C L K \_F R E Q ~ / ~ D O T C L K \_F R E Q ~}$ |

### 4.3 Analog-to-Digital Converter (ADC)

The Analog-to-Digital block has the following features:

- Supports input clocks up to 140 MHz (SXGA 75 Hz )
- Adjustable analog amplifier bandwidth
- Differential RGB input path for noise immunity
- Built-in Sync-on-Green support
- Individual RGB clock delay control
- Power down control
- Linear and independent Gain/Offset adjustment.


## GAIN CONTROL

Red, Green, and Blue channels have independent control registers: ANA_ADC_RED_0, ANA_ADC_GRN_0, and ANA_ADC_BLU_0, respectively.

8-bit control covers amplitudes from $0.35 \mathrm{~V}(00)$ to 1.05 V (FF) in steps of 2.74 mV .
OFFSET CONTROL
Red, Green, and Blue channels have independent control registers: ANA_ADC_RED_1, ANA_ADC_GRN_1, and ANA_ADC_BLU_1, respectively.
6 -bit control covers a range of $\pm 92.8 \mathrm{mV}$ in steps of 2.9 mV .

### 4.3.1 $\quad 216 \mathrm{MHz}$ Frequency Synthesizer

The FS216 (controlled by the ANA_FS216_CTRL register) is the system PLL that drives the SCLK and DCLK frequency synthesizers (refer to Section 4.2: Frequency Synthesizer (FSYN)) and the LLK, by generating two different reference clock frequencies, 216=27x8 MHz (FSYN) and 54=27x2 MHz (LLK), based on XCLK.
For normal operation with a 27 MHz crystal, this register should be programmed to OA .
The control register also allows for different crystal frequencies, power down, and optional use of an external PLL.

### 4.3.2 Sync-on-Green (SOG)

It is necessary to tune the analog SOG circuit in order to secure a valid HSync that can be used by the Line Lock PLL; the LLK may then be programmed to generate an in-clock. The ADC clamp relies on in-clock and may only be enabled once this step is complete. Clamp pulse is used to set the ADC black level reference voltage. In normal operation, the SOG signal is clamped by the ADC clamp, and this clamp is not available during the initial tuning. For the initial tuning phase, instead of the ADC clamp, the SOG clamp (pull down current) is used to clamp the input SOG signal. Once the tuning has been accomplished, and there is a valid reference HSync and in-clock, the SOG clamp may be disabled and the ADC clamp may be enabled.
There are therefore 2 states of sync-on-green operation: the initial state, which employs the SOG clamp, and the normal (or locked) state, which employs the ADC clamp.

### 4.3.2.1 Initial SOG Clamp State

At power up, set:

- ANA_ADC_SOG_1[0] = 0 (power down bit; apply power to SOG),
- ANA_ADC_SOG_1[3] = 1 (enable SOG clamp pull down current),
- ANA_ADC_GRN_2[1] = 1 (ADC clamp off; must be the same as ANA_ADC_SOG_1[3]), and adjust ANA_ADC_SOG_0[4:0] \& ANA_ADC_SOG_1[7:4] until one of the three comparators detects a SOG signal. Select a SOG signal to be the reference HSync to which the Line Lock PLL will lock.
The normal value of the pull down current is 1.1 uA and can be adjusted with
ANA_ADC_SOG_1[2:1]. Either ANA_ADC_SOG_1[0] = 1 or ANA_ADC_SOG_1[3] = 0 will turn off the pull down current.

The ADC clamp signal is generated in digital circuitry.
Figure 4: Initial SOG Clamp Phase


### 4.3.2.2 SOG Lock State

Set:

- ANA_ADC_SOG_1[0] remains 0,
- ANA_ADC_GRN_2[1] = 0 (ADC clamp on; must be the same as ANA_ADC_SOG_1[3]).
- ANA_ADC_SOG_1[3] = 0 (disable SOG clamp pull down current),

This enables the ADC Clamp circuit and disables the SOG Clamp (this is the recommended order it is better to have overlap than no clamp at all). The comparators will continue to compare the input signal with the reference voltages and provide a correct SOG signal. Comparator threshold voltages can be adjusted to optimize noise immunity if necessary.

The ideal ADC clamp signal would be greater than 1 us wide and placed precisely between the SOG pulse and video data. Any overlap or misalignment will alter the Green offset level internally and comparators may lose track of SOG signal.

Figure 5: SOG Lock Phase


## Level Adjustment

All 3 comparator thresholds and clamp voltage are moved up or down together by changing registers. These cannot be individually adjusted.

- To shift up:
- Set ANA_ADC_SOG_1[7:4] = 0F
- Adjust ANA_ADC_SOG_0[4:0] to a higher value. (The default is $0, \sim 8.8 \mathrm{mV}$ per increment.)
- To shift down:
— Set ANA_ADC_SOG_0[4:0] = 0b00000
- Adjust ANA_ADC_SOG_1[7:4] to a lower value. (The default is 0F, $\sim 10 \mathrm{mV}$ per decrement; a value of 00 is invalid.)
To power down SOG, set ANA_ADC_SOG_1[0] = 1 .
Note: The SMEAS block can still detect SOG activity while the ADC is powered down.
There are three SOG analog voltage comparators that generate the SOG0, SOG1, and SOG2 digital signals. These signals are then sent to the LLK, SRT, SMEAS, and SMUX blocks.
For SOG support the SMEAS block has:
- Three 8-bit edge counters (used to detect activity)
- Four 4-bit delay counters (used to tune the comparator reference voltages)

The 4 delay counters measure the time (in XCLKs) between the leading and trail edges of the SOG signals, as follows:
d1: delay count from SOG[2] falling edge to SOG[1] falling edge
d2: delay count from SOG[1] falling edge to SOG[0] falling edge
d3: delay count from SOG[0] rising edge to SOG[1] rising edge
d4: delay count from SOG[1] rising edge to SOG[2] rising edge


If there is no leading edge for a particular delay counter, the result is 0 .
If both edges are within the same XCLK period, the result is 1 .
When the counter reaches a value of $0 F$, it stops.
The delay and activity registers are used together to tune the SOG sampling level.
The delay measurements are controlled by the activity detection control registers which may be used to select either:

- One-shot: one sync pulse measurement; when done, hold result until next measurement is started; or
- Free-run: continuously measures, results are dynamically updated.

There are 8 possible cases as listed in the figure below.
The N is a whole number from 1 to E representing a stable delay. $\mathrm{F} / 0$ is a whole number between 0 and $F$ representing a delay that varies in time (because Green data is being measured). 1 in the activity column means stable activity is detected, 0 means permanent no activity, and X indicates video dependence.

|  | Sample position | Comparators output Wave form | 3 Active | Delay counters d1-d2-d3-d4 |
| :---: | :---: | :---: | :---: | :---: |
| Case 0: <br> A. ref. set too low; <br> B. No sync tip; <br> C. ref. set too high <br> but no active video |  |  | 0-0-0 | 0-0-0-0 |
| Case 1: <br> 1 right <br> 2 too low |  |  | 1-0-0 | F-0-0-0 |
| Case 2: <br> 2 right <br> 1 too low |  | $\square$ | 1-1-0 | N-F-0-N |
| Case 3: <br> All 3 inputs are sampled from right positions |  |  | 1-1-1 | N-N-N-N |
| Case 4: <br> 1 too high <br> 2 right |  |  | X-1-1 | F/0-N-N-F |
| Case 5: <br> 2 too high <br> 1 right |  |  | X-X-1 | F/0-F/0-F-F/0 |
| Case 6: all 3 too high |  | 吅$\square$ <br> $m$$\sqrt{7}$ | X-X-X | F/0-F/0-F/0-F/0 <br> (distinguished <br> from Case 3 <br> with $\mathrm{x} / \mathrm{h}$ \& $\mathrm{x} / \mathrm{v}$ <br> measurements) |
| Case 7: <br> Sync tip too small, only middle one right |  |  | X-1-0 | F/0-F-0-F |

Table 13: ADC Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_FS216_CTRL ${ }^{\text {a }}$ | 0040 | R/W | [4:3] | 01 | xtal freq multiplier, ndiv <br> $0: \mathrm{f}_{\mathrm{xclk}}=54 \mathrm{MHz}{ }^{\mathrm{b}}$ <br> $1^{*}: f_{\text {xclk }}=27 \mathrm{MHz}\left(\right.$ normal) ${ }^{2}$ <br> 2: $\mathrm{f}_{\mathrm{xclk}}=13.5 \mathrm{MHz}^{2}$ <br> 3: reserved |
|  |  |  | [2] |  | external pll $0^{*}$ : internal 1: external |
|  |  |  | $\begin{array}{\|l\|} \hline[1] \end{array}$ |  | pll select <br> $0^{*}$ : disabled <br> 1: enabled |
|  |  |  | [0] |  | disable FS216 analog VCO <br> $0^{\star}$ : enabled <br> 1: disabled |
| ANA_ADC_PWDN | 0050 | R/W | [0] | 01 | AFE power control 0 : on <br> 1*: off |
| ANA_ADC_SOG_0 | 0051 | R/W | [4:0] | 00 | SOG level detection \& clamp <br> Up when ADCSOG1[7:4]=1 <br> OmV to +282 mV , at $\sim 8.8 \mathrm{mV}$ per step <br> 00000*: 0mV <br> 11111: +282mV |
| ANA_ADC_SOG_1 | 0052 | R/W | [7:4] | 01 | SOG level detection \& clamp <br> Down when ANA_ADC_SOG_O[4:0]=0 <br> OmV to -340mV, 10mV per step <br> $0^{*}$ : disabled <br> 1: -340 mV <br> F: OmV |
|  |  |  | [3] |  | Enable SOG clamp \& pull down current $0^{*} \text { : off }$ 1: on |
|  |  |  | [2:1] |  | SOG pull down current adjust MAX/TYP/MIN <br> 00*: 1.4/1.1/0.8 uA <br> 01: 0.7/0.5/0.4 uA <br> 10: 5.3/4.1/3. 1 uA <br> 11: 2.7/2.1/1.6 uA |
|  |  |  | [0] |  | $\begin{array}{\|l\|} \hline \text { SOG power control } \\ \text { 0: on } \\ \text { 1*: off } \end{array}$ |

Table 13: ADC Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_ADC_BIAS | 0053 | R/W | [5] | 01 | ADC Band gap power control $\begin{aligned} & 0^{\star} \text { : on } \\ & \text { 1: off } \end{aligned}$ |
|  |  |  | [4:3] |  | IREF adjustment for internal bias, when ADCBIAS[2:1]=01 (or 11) $00^{*}: 600 u A$ <br> 01: 750uA <br> 10: 300uA <br> 11: 450uA |
|  |  |  | [2:1] |  | Must be set to 01 |
|  |  |  | [0] |  | ADC power control <br> 0: on <br> $1^{*}$ : off |
| ANA_ADC_RED_0 | 0054 | R/W | [7:0] | 7F | GAIN CONTROL <br> $2.74 \mathrm{mV} /$ step <br> 00: 0.35V <br> FF: 1.05 V |
| ANA_ADC_RED_1 | 0055 | R/W | [7] | OF | VREF <br> $0^{*}$ : internal <br> 1: external |
|  |  |  | [5:0] |  | OFFSET CONTROL: $2.9 \mathrm{mV} / \mathrm{step}$ |
| ANA_ADC_RED_2 | 0056 | R/W | [6:4] | 00 | Channel Skew control LSB = 200ps(typ) |
|  |  |  | [3:2] |  | Amp bandwidth adjust <br> 00*: BW=250MHz (min) <br> 01: $B W=150 \mathrm{MHz}$ (min) <br> 10: reserved <br> 11: $\mathrm{BW}=40 \mathrm{MHz}(\mathrm{min})$ |
|  |  |  | [1] |  | Clamp Control 0*: enabled 1: disabled |
|  |  |  | [0] |  | ADC Dithering (ADTH block) $0^{*}$ : disabled <br> 1: enabled |
| ANA_ADC_GRN_0 | 0057 | See ANA_ADC_RED_0. |  |  |  |
| ANA_ADC_GRN_1 | 0058 | See ANA_ADC_RED_1. |  |  |  |
| ANA_ADC_GRN_2 | 0059 | See ANA_ADC_RED_2. |  |  |  |
| ANA_ADC_BLU_0 | 005A | See ANA_ADC_RED_0. |  |  |  |
| ANA_ADC_BLU_1 | 005B | See ANA_ADC_RED_1. |  |  |  |
| ANA_ADC_BLU_2 | 005C | See ANA_ADC_RED_2. |  |  |  |

a. Normal value for ANA_FS216_CTRL is OAh.
b. When $x$ clk $=27 \mathrm{MHz}$

### 4.4 Analog Dithering (ADTH)

The ADTH block generates a 3-bit dither pattern ADTH_OUT[2:0] to tune the 10-bit resolution of the ADC block.

Note: ADTH_OUT[2:0] is not a register but the generated 3-bit dither output of the ADTH block.

### 4.4.1 Function

The ADTH block consists of a $32 \times 32 \times 3$ bit look up table (LUT). It represents one dither matrix, which can be read using a programmable addressing technique as well as a programmable output amplitude control. When ADTH_MAT_CTRL[0] is zero or during the clamp pulse ADTH_OUT[2:0] = 3. During vertical blanking ADTH_OUT[2:0] is set to ADTH_TEST_DITHER[2:0] to provide a feedback mechanism for calibration.

### 4.4.2 Addressing Technique

The ADTH block offers a programmable addressing technique to generate various temporal dither patterns. ADTH_FRAME_CTRL [7:4] is a 4-bit increment value, which defines the horizontal/vertical displacement of the dither matrix from frame to frame (precisely at rising edge of CLAMP_IN and at falling edge of VENAB).
After (ADTH_FRAME_CTRL [3:0] + 1) number of frames the horizontal/vertical displacement position will be reset to zero/zero, only when_ADTH_FRAME_CTRL [3:0]> 0 .

Note: $\quad$ To set the frame accumulator to zero, program ADTH_FRAME_CTRL [7:4] to zero and program ADTH_FRAME_CTRL [3:0] to 1. ADTH_FRAME_CTRL [7:4] can be independently activated in the horizontal and vertical dimensions using ADTH_MAT_CTRL [2] and ADTH_MAT_CTRL [3], respectively.

### 4.4.3 Output Amplitude Control

The 3-bit LUT output value can be scaled to a reduced dither amplitude using ADTH_MAT_CTRL [5:4]. After adding the ADTH_MAT_CTRL [7:6] to the (reduced) dither amplitude the final 3-bit amplitude is output as ADTH_OUT[2:0].

### 4.4.4 Miscellaneous

During the ADC clamp pulse, the output of the ADTH block is muted; that is the output value is set to 3 (ADTH_OUT[2:0] = 3). In addition, ADTH_CLAMP_CTRL[7:4] delays the clamp pulse by 0 to 15 clock cycles while muting, and ADTH_CLAMP_CTRL[3:0] adds 0 to 15 clock cycles of muting after the falling edge of the clamp pulse.
For AFE dither calibration, ADTH_OUT[2:0] can be programmed via ADTH_TEST_DITHER to a static value during vertical blanking.

Table 14: ADTH Registers

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ADTH_MAT_CTRL | 03D0 | R/W | [7:6] | 01 | amplitude_offset adth_out[2:0] = (dither_amplitude + amplitude_offset) $\% 8$ |
|  |  |  | [5:4] |  | dither_amplitude <br> $0^{*}$ : dither amplitude range: 0-7 <br> 1: dither amplitude range: 0-6 <br> 2: dither amplitude range: 0-5 <br> 3: dither amplitude range: 0-4 |
|  |  |  | [3] |  | 1: vertical start position of dither matrix changes by FRAME_OFFSET |
|  |  |  | [2] |  | 1: horizontal start position of dither matrix changes by FRAME_OFFSET |
|  |  |  | [1] |  | Clamp polarity. To be set to 1 . |
|  |  |  | [0] |  | 0: adth_out[2:0] = 3 <br> $1^{*}$ : AFE dither amplitude enabled |
| ADTH_FRAME_CTRL | 03D1 | R/W | $[7: 4]$ | 00 | frame_offset <br> Offset the start position of the dither matrix from frame to frame by frame_offset. <br> See frame_len. |
|  |  |  | [3:0] |  | frame_len <br> Reset dither matrix start position after frame_len +1 number of frames when frame_len $>0$. <br> See frame_offset. |
| ADTH_CLAMP_CTRL | 03D2 | R/W | [7:4] | 00 | clamp_begin <br> Delay and mute the clamp pulse by 0-15 clock cycles Note: adth_out[2:0] = 3 during clamping/muting |
|  |  |  | [3:0] |  | clamp_end <br> Mute after the end of clamp pulse for 0-15 clock cycles <br> Note: adth_out[2:0] = 3 during clamping/muting |
| ADTH_TEST_DITHER | 03D3 | R/W | [2:0] | 00 | For AFE dither amplitude (voltage) calibration. During vertical blanking adth_out[2:0] = test_dither |

### 4.5 Line Lock PLL (LLK)

The LLK generates the ADC input pixel sampling clock from an incoming HSync source and a multiplying factor (MFACTOR, aka Clock). The loop filter parameters and skew (aka Phase) can be tuned. The phase can be adjusted in steps of 72 ps. The minimum LLK generated clock frequency is 13.5 MHz .

The PLL filter has two states with independent filter parameters: Fast and Slow. If while in the Fast state the phase detector error count remains below a programmable threshold (LLK_LOCK_TOL) for a programmable number of input lines (LLK_LOCK_LI NE_NB), the PLL changes to the Slow state. While in this state, the Slow filter coefficients apply. In the event that phase detector errors should exceed LLK_LOCK_TOL for one or more lines, the PLL returns to the Fast state in one line, and Fast filter coefficients again apply.

The digital loop filter is controlled by two parameters: A and B. The A and B parameters control the response of the 2nd order digital filter. $A$ and $B$ are exponential coefficients. The relationship of these numbers to the classic 2nd order damping and natural frequency are as follows:

- Damping $=2^{\wedge}(\mathrm{AE}-8) * \operatorname{SQRT}\left(5^{*}\right.$ MFACTOR / (2^(BE+4)))
- Natural Frequency $=$ SQRT(MFACTOR * 5 * $2^{\wedge}($ BE-30))

Note: Typical value for the $A$ and $B$ parameters is $66 h$.
The synthesized HSync supplied to SMUX is $50 \%$ duty cycle.
Table 15: Line Lock PLL Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| LLK_CTRL | 0800 | R/W | [6] | 00 | $0^{*}$ : use slow filter when coarse error is zero <br> 1: use slow filter when lock condition is achieved |
|  |  | R/W | [5] |  | mfactor shadow control <br> 0 *: simple shadow. Apply new mfactor when mfactor_u is written. <br> 1: shadow transfer on in_venab falling edge. |
|  |  | R/W | [4] |  | 0*: lock to rising edge of input HSync <br> 1: lock to falling edge of input HSync |
|  |  | R/W | [3:1] |  | input HSync select 0*: HSYNC pin <br> 1: SOGO <br> 2: SOG1 <br> 3: SOG2 <br> 4: EXT_SOG |
|  |  | R/W | [0] |  | LLK pll free run enable |
| LLK_SYNC_OFFSET_MODE | 0801 | R/W | [3] | 06 | manual resync mode <br> The LLK pll requires a resync after any change of mfactor or offset. Writing to this bit causes a one-time resync of the PLL accumulator (cleared by H/W). |
|  |  | R/W | [2] |  | resync every frame mode ${ }^{\text {a }}$ |
|  |  | R/W | [1] |  | resync on in_venab falling edge ${ }^{1}$ |
|  |  | R/W | [0] |  | resync on in_venab rising edge ${ }^{1}$ |
| LLK_MFACTOR_L | 0802 | R/W | [7:0] | 80 | mfactor[7:0] = in_htotal |
| LLK_MFACTOR_U | 0803 | R/W | [3:0] | 02 | mfactor[11:8] |
| LLK_PHASE_RATE_INIT | 0804 | R/W | [7:0] | 80 | pll phase rate init freq = xclk_freq * 128 / phase_rate_init. |
| LLK_TC_AEF | 0805 | R/W | [3:0] | OA | time constant A when out of lock |
| LLK_TC_BEF | 0806 | R/W | [3:0] | OA | time constant B when out of lock |
| LLK_TC_AES | 0807 | R/W | [3:0] | 06 | time constant A when in lock |
| LLK_TC_BES | 0808 | R/W | [3:0] | 06 | time constant B when in lock |
| LLK_LOCK_TOL | 0809 | R/W | [7:0] | 20 | error limit for determining lock. LSB $=150 \mathrm{ps}$ (typ) |
| LLK_LOCK_LINE_NB | 080A | R/W | [7:0] | 30 | line count for determining lock. <br> - set when error is < lock_tol for lock_line_nb of lines. <br> - cleared if error exceeds lock_tol. |

Table 15: Line Lock PLL Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| LLK_OFFSET_L <br> LLK_OFFSET_U | $\begin{aligned} & 080 \mathrm{~B} \\ & 080 \mathrm{C} \end{aligned}$ | R/W <br> R/W | $\left[\begin{array}{l} {[7: 0]} \\ {[1: 0]} \end{array}\right.$ | 00 <br> 00 | phase offset [7:0] of adc sample clock. LSB is xclk_period/512 = 72ps. <br> phase offset |
| LLK_PULSE_HIGH_EXT | 080D | R/W | [7] | 00 | inclk pulse extend enable |
|  |  | R/W | [2:0] |  | inclk pulse extend value. LSB $=0.3 \mathrm{~ns}$ (typ) |
| LLK_PHASE_RATE_MIN | 080E | R/W | [7:0] | 14 | phase rate minimum. Sets the upper frequency limit of the PLL. <br> phase_rate_min = xclk_freq * 128 / max_inclk_freq. |
| LLK_STAT_LINE_NB_L <br> LLK_STAT_LINE_NB_U | $\begin{array}{l\|} \hline 080 \mathrm{~F} \\ 0810 \end{array}$ | $\begin{aligned} & \hline R / W \\ & R / W \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 40 \\ & 00 \end{aligned}$ | number of lines over which statistics are gathered number of lines over which statistics are gathered |
| LLK_STAT_SUM_ABS_MAX_L LLK_STAT_SUM_ABS_MAX_U | $\begin{aligned} & \hline 0811 \\ & 0812 \end{aligned}$ | $\begin{aligned} & \hline R / W \\ & R / W \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 60 \\ & 00 \end{aligned}$ | limit for sum of absolute errors |
| LLK_STAT_MAX_ABS_MAX | 0813 | R/W | [7:0] | 04 | limit for absolute error |
| LLK_DEADZONE | 0814 | R/W | [3:0] | 02 | coarse error deadzone, normal operation $=2$. |
| LLK_STATUS | 0830 | R | [4] | 00 | max absolute error exceeded limit, not sticky |
|  |  | R | [3] |  | sum of absolute errors exceeded limit, not sticky |
|  |  | R | [2] |  | pll filter overflow condition, not sticky |
|  |  | R | [1] |  | coarse error is zero status, not sticky |
|  |  | R | [0] |  | lock status, not sticky |
| ```LLK_STATUS_PHASE_RATE_I_ O LLK_STATUS_PHASE_RATE_I_ 1 LLK_STATUS_PHASE_RATE_I_ 2 LLK_STATUS_PHASE_RATE_I_ 3``` | $\begin{aligned} & 0831 \\ & 0832 \\ & 0833 \\ & 0834 \end{aligned}$ | R <br> R/W <br> R/W <br> R/W | [7:0] <br> [7:0] <br> [7:0] <br> [5:0] | 00 <br> 00 <br> 00 <br> 00 | pll phase rate, free running readout. ${ }^{\text {b }}$ |
| LLK_STATUS_SUM_ABS_L LLK_STATUS_SUM_ABS_U | $\begin{array}{\|l\|} \hline 0835 \\ 0836 \end{array}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | sum of absolute errors readout |
| LLK_STATUS_MAX_ABS | 0837 | R | [7:0] | 00 | max absolute error readout |
| LLK_MFACTOR_HW_L LLK_MFACTOR_HW_U | $\begin{aligned} & \hline 0842 \\ & 0843 \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 80 \\ & 02 \end{aligned}$ | mfactor shadow hw readout |
| LLK_TEST | 084F | R/W | [7:0] | 00 | reserved |

a. recommended setting for bits $[2: 0]=110 \mathrm{~b}$
b. phase rate period (in picoseconds) is:

$$
\text { haseRate }(p s)=\frac{\text { LIkStatusPhaseRate }[28 . .0] \times 2^{-21} \times X T A L p e r i o d(p s}{128}
$$

Example:
LLK_STATUS_PHASE_RATE_I_[28:0] = 03335BDF , XTAL = 27 MHz
phase rate period $=7.409 \mathrm{~ns}$
frequency $=134.97 \mathrm{MHz}$

### 4.6 Sync Retiming (SRT)

The Sync Retiming block retimes synchronization signals (e.g. HSync and VSync) into either the XCLK or in-clock domains.
SRT provides the following:

- Retimes all sync signals going to SMEAS into the xclk domain
- Extracts vertical sync from composite sync signals
- Divides sclk by up to 1024 for activity detection purposes (SMEAS)
- Generates a delayed version of vertical sync from a mux-selectable vertical sync source
- Generates a coast signal in the xclk domain for the LLPLL
- Measures the effect of the filter on marginal composite sync signals and returns a bad_filter flag
- Retimes horizontal and vertical syncs into the inclk domain.


### 4.6.1 Coast Signal

In composite or SOG sync mode, HSYNC pulses may not exist during the VSYNC pulse signal and will cause the LLK to unlock and loose track of HSYNC signal. Coarse signal (also known as LLK Inhibit/Free Run signal) is used to generate a vertical pulse that wraps around the incoming VSYNC.
Coast pulse reference (0) is either edge of VSYNC, and its set and reset values are expressed in XCLK units.

Figure 6: Vertical sync extraction and filtering


Table 16: Sync Retiming Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SRT_CSYNC_INV | 01E0 | R/W | [2] | 00 | invert vert sync signal extracted from internal SOG comparator (sog_vs_inv) |
|  |  | R/W |  |  |  |
|  |  | R/W | [1] |  | invert vert sync extracted from composite sync signal on HSync pin (csync_vs_inv) |
|  |  | R/W | [0] |  | invert filtered vert sync (filt_vs_inv) |
| SRT_CSYNC_THR_L | 01E1 | R/W | [7:0] | $80$ | composite sync vertical sync extractor threshold (this is the narrowest HSync signal sent $+50 \%$ as a safety margin) refer to Figure 7 |
| SRT_CSYNC_THR_U | 01E2 | R/W | [3:0] | 00 |  |
| SRT_VSYNC_SEL | 01E3 | R/W | [2:0] | 00 | filtered vert sync source select <br> 0*: VSYNC pin <br> 1: vsync from composite HSYNC pin <br> 2: vsync from composite SOG[0] comparator <br> 3: vsync from composite SOG[1] comparator <br> 4: vsync from composite SOG[2] comparator <br> 5: vsync from alternate SOG source <br> 6-7: Reserved |
| SRT_VSYNC_THR_L SRT_VSYNC_THR_U | $\begin{aligned} & \hline 01 \mathrm{E} 4 \\ & \text { 01E5 } \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 80 \\ & 00 \end{aligned}$ | filtered vert sync delay |
| SRT_COAST_VS_SEL | 01E6 | R/W | [3] | 00 | coast signal trigger edge $0^{*}$ : rising edge of selected VSync <br> 1: falling edge of selected VSync |
|  |  | R/W | [2:0] |  | source selection for coast VSync trigger 0*: VSYNC pin <br> 1: vsync from composite HSYNC pin <br> 2: vsync from composite SOG[0] comparator <br> 3: vsync from composite SOG[1] comparator <br> 4: vsync from composite SOG[2] comparator <br> 5: filtered and delayed vsync (normal) <br> 6: vsync from alternate SOG source <br> 7: Reserved |
| ```SRT_COAST_RISE_L SRT_COAST_RISE_M SRT_COAST_RISE_U``` | $\begin{aligned} & \hline 01 \mathrm{E} 7 \\ & 01 \mathrm{E} 8 \\ & 01 \mathrm{E} 9 \end{aligned}$ | R/W <br> R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & \hline 00 \\ & 00 \\ & 00 \end{aligned}$ | rising edge of coast, in XCLKs from vsync trigger |
| ```SRT_COAST_FALL_L SRT_COAST_FALL_M SRT_COAST_FALL_U``` | $\begin{aligned} & 01 \mathrm{EA} \\ & 01 \mathrm{~EB} \\ & 01 \mathrm{EC} \end{aligned}$ | R/W <br> R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \\ & 00 \end{aligned}$ | falling edge of coast, in XCLKs from vsync trigger |
| SRT_HS_CTRL | 01EE | R/W | [4] | 00 | Edge of inclk on which to sample horizontal sync: $0^{*}$ : rising edge <br> 1: falling edge (normal) |
|  |  | R/W | [2:0] |  |  |

Table 16: Sync Retiming Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SRT_VS_SEL | 01EF |  | [5:4] | 00 | sclk div prescaler for SMEAS: $\begin{aligned} & \text { 0*: } 128 \\ & 1: 256 \\ & \text { 2: } 512 \\ & \text { 3: } 1024 \end{aligned}$ |
|  |  | R | [3] |  | Bad csync threshold. Change SRT_CSYNC_THR until this is stable low. |
|  |  | R/W | [2:0] |  | vert sync source select for re-sampling into inclk domain for SMUX: <br> 0*: VSYNC pin <br> 1: vsync from composite HSYNC pin <br> 2: vsync from composite SOG[0] comparator <br> 3: vsync from composite SOG[1] comparator <br> 4: vsync from composite SOG[2] comparator <br> 5: filtered and delayed vsync (normal) <br> 6: vsync from alt SOG source pin <br> 7: reserved |
| SRT_COAST_RISE_HW_L | 01F0 | R | [7:0] |  | Shadow read back |
| SRT_COAST_RISE_HW_M | 01F1 | R | [7:0] |  |  |
| SRT_COAST_RISE_HW_U | 01F2 | R | [7:0] |  |  |
| SRT_COAST_FALL_HW_L | 01F3 | R | [7:0] |  |  |
| SRT_COAST_FALL_HW_M | 01F4 | R | [7:0] |  |  |
| SRT_COAST_FALL_HW_U | 01F5 | R | [7:0] |  |  |

Note: All thresholds are in XCLK units.

Figure 7: VSync Up/Down Counter


### 4.7 Input Sync Measurement (SMEAS)

The SMEAS block monitors input activity and measures input sync signals from all sources. All unused and reserved bits return as zero. SMEAS operates in the crystal clock (xclk) domain.
Input Sync Functions:

- Activity Detection: detects input activity
- Measurement: measures sync period and width


### 4.7.1 Input Sync - Activity Detection

The activity block measures all sync sources in parallel. An active channel is defined as having a programmable number of rising edges within a programmable number of xclk cycles (= sample period). Activity limits are set per channel class: clkdiv1k and HSync; vsync. The activity results are updated each sample period.
Software can select either:

- One shot: one time measurement
- Free Run: continuously running measurements


### 4.7.2 Input Sync - Measurement

One set of (HSync, vsync) can be selected for measurement.
Software can request measurements in one of two ways:

- One shot - one time measurement
- Free Run - continuously running measurements.

The measurement block also compares the measured sync signals to programmable limits.

- Xclks per vsync different by more than +/- 2^(xclk_vtol_exp[3:0])
- Xclks per HSync different by more than +/- 2^(xclk_htol_exp[3:0])
- HSyncs per vsync different by more than +/- HSync_vtol[3:0]
- polarity.

Range check flags will be set when the measurements exceed the programmed tolerances. The flags will be updated on the completion of each measurement in Free Run mode. The flags maintain their state at the completion of a measurement while in One Shot mode. When a measurement is started (asserting the Measurement Start bit) the range check flags are cleared.

There are timeout registers to detect the absence of sync signals.
The measurement block registers are grouped into four main categories:

- Timeouts \& Tolerances
- Measurements (obtained by a one-shot or free-run mode of operation)
- Reference values
- Flags (indicators that measurements have timed out or measurements compared to reference values exceed tolerances).


### 4.7.3 Fast Mute

The fast mute block continuously monitors one selected HSync signal and compares its period with an independent reference value and tolerance. A fast mute flag is set as soon as the measured period is outside the tolerance for more than 1,2 or 3 times in a row.
The fastmute range check flag can be combined with other reference checking flags with a mask-or function to make a sticky bit to mute the screen rapidly in the event of a mode change or dropped signal.

Note: $\quad$ Timeout and Tolerance use Horizontal and Vertical measurements. These can either be the Horizontal or Vertical syncs from an Analog input or the local generated Horizontal Enable and/or Vertical Enable.

Table 17: SMEAS Register Definitions (Sheet 1 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_ACT_CTRL | 0100 | R/W | [3] | 00 | Free-run enable |
|  |  | R/W | [2] |  | Freeze results during free run mode. No meaning in one shot mode. <br> $0^{*}$ : Do not freeze. New result will be available on the next and subsequent toggle of the polling bit. <br> 1: Freeze the current results. The polling bit will still toggle and the block continues to free run; however, results will not update. |
|  |  | R/W | [1] |  | Activity detection start. <br> In one-shot mode setting this bit triggers the start of a measurement. This bit is reset to zero when the measurement is complete. No meaning in free run mode. |
|  |  | R/W | [0] |  | Activity detection mode. <br> $0^{*}$ : free-run mode <br> 1: one-shot mode |
| SMEAS_ACT_H_SMPTM_L SMEAS_ACT_H_SMPTM_U | $\begin{aligned} & \hline 0101 \\ & 0102 \end{aligned}$ | $\begin{aligned} & \hline \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Sample period value for clock or HSync activity. Xclks [7:0] Sample period value for clock or HSync activity. Xclks [15:8] |
| SMEAS_ACT_V_SMPTM_L SMEAS_ACT_V_SMPTM_U | $\begin{aligned} & 0103 \\ & 0104 \end{aligned}$ | $\begin{aligned} & \hline R / W \\ & R / W \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Sample period value for vsync activity. Xclks / 256 [7:0] <br> Sample period value for vsync activity. Xclks / 256 [15:8] |
| SMEAS_ACT_H_MINEDGE | 0105 | R/W | [7:0] | 00 | Minimum edge count value for clk or HSync activity. |
| SMEAS_ACT_V_MINEDGE | 0106 | R/W | [7:0] | 00 | Minimum edge count value for vsync activity. |
| SMEAS_H_TMOT_L <br> SMEAS_H_TMOT_U | $\begin{aligned} & 0107 \\ & 0108 \end{aligned}$ | R/W <br> R/W | $\left[\begin{array}{l} {[7: 0]} \\ {[7: 0]} \end{array}\right.$ | $00$ $00$ | Timeout counter value for clk or horizontal measurement. xclks [7:0] <br> Timeout counter value for clk or horizontal measurement. xclks [15:8] |
| $\begin{aligned} & \text { SMEAS_V_TMOT_L } \\ & \text { SMEAS_V_TMOT_U } \end{aligned}$ | $\begin{aligned} & 0109 \\ & 010 A \end{aligned}$ | R/W <br> R/W | $\left[\begin{array}{l} {[7: 0]} \\ {[7: 0]} \end{array}\right.$ | 00 00 | Timeout counter value for vertical measurement. xclks / 256 [7:0] <br> Timeout counter value for vertical measurement. xclks / 256 [15:8] |
| SMEAS_CLEAR | 0110 | R/W | $\begin{array}{\|l\|} \hline[1] \\ \hline[0] \\ \hline \end{array}$ | 00 | clears SMEAS_STATUS_RANGE[7] sticky bit only. <br> Must be reset by software. <br> clears timeouts, measurements. <br> Must be reset by software. |

Table 17: SMEAS Register Definitions (Sheet 2 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_H_CTRL | 0111 | R/W | [5] | 00 | Measures HSync in the condition of no VSync |
|  |  | R/W | [4] |  | Free-run enable |
|  |  | R/W | [3] |  | Edge measurement selection for horizontal period events. <br> $0^{*}$ : rising edge. <br> 1: negative edge. |
|  |  | R/W | [2] |  | Freeze results during free run mode. No meaning in one shot mode. <br> 0*: Do not freeze the results in free run mode. New results will be available on the next and subsequent toggle of the polling bit. <br> 1: Freeze the current results in free run mode. The polling bit will still toggle and the block continues to free run; however, results will not update. |
|  |  | R/W | [1] |  | In free-run mode it enables measurements. In one-shot mode it triggers the start of a measurement and is reset to zero when the measurement is complete. |
|  |  | R/W | [0] |  | 0*: free-run mode. <br> 1: one-shot mode. |
| SMEAS_V_CTRL | 0112 | R/W | [4] | 00 | Free-run enable |
|  |  | R/W | [3] |  | Edge measurement selection for vertical period events. <br> $0^{*}$ : rising edge. <br> 1: negative edge. |
|  |  | R/W | [2] |  | Freeze results during free run mode. No meaning in one shot mode. <br> $0^{*}$ : Do not freeze the results in free run mode. New result will be available on the next and subsequent toggle of the polling bit. <br> 1: Freeze the current results in free run mode. The polling bit will still toggle and the block continues to free run; however, results will not update. |
|  |  | R/W | [1] |  | In free-run mode it enables measurements. In one-shot mode it triggers the start of a measurement and is reset to zero when the measurement is complete. |
|  |  | R/W | [0] |  | $0^{*}$ : free-run mode. <br> 1: one-shot mode. |

Table 17: SMEAS Register Definitions (Sheet 3 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_H_SEL | 0113 | R/W | [6:4] | 00 | Fastmute input select <br> 0*: HSync <br> 1: HSync generated from LLK <br> 2: EXT_SOG <br> 3: SOG[0] <br> 4: SOG[1] <br> 5: SOG[2] <br> 6,7: reserved |
|  |  | R/W | [3:0] |  | H measurement input select <br> 0*: HSync <br> 1: HSync generated from LLK <br> 2: EXT_SOG <br> 3: SOG[0] <br> 4: SOG[1] <br> 5: SOG[2] <br> 6-A: reserved <br> B: inclk / 1024 (for test only) <br> C: dotclk / 1024 (for test only) <br> D: TCON enab (for test only) <br> E: TCON HSync (for test only) <br> F: sclk_div (for test only) |
| SMEAS_V_SEL | 0114 | R/W | [7:4] | 00 | Vertical high level duration measurement input select 0*: VSYNC pin <br> 1: extracted Vsync from HSYNC pin composite sync <br> 2: extracted Vsync from EXT_SOG composite sync <br> 3: filtered vsync from SRT block (normal condition) <br> 4: SOG[0] extracted vsync <br> 5: SOG[1] extracted vsync <br> 6: SOG[2] extracted vsync <br> 7-F: reserved |
|  |  | R/W | [3:0] |  | V measurement input select <br> 0*: VSYNC pin <br> 1: extracted Vsync from HSYNC pin composite sync <br> 2: extracted Vsync from EXT_SOG composite sync <br> 3: filtered vsync from SRT block (normal condition) <br> 4: SOG[0] extracted vsync <br> 5: SOG[1] extracted vsync <br> 6: SOG[2] extracted vsync <br> 7-F: reserved |

Table 17: SMEAS Register Definitions (Sheet 4 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_STATUS_MASK ${ }^{\text {a }}$ | 0119 | R/W | [7] | 00 | Enable mute function to respond to SMEAS_STATUS_RANGE[6] (hpol). |
|  |  | R/W | [6] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[5] (vpol). |
|  |  | R/W | [4] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[4] (fastmute). |
|  |  | R/W | [3] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[3] (xpervhi). |
|  |  | R/W | [2] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[2] (hperv). |
|  |  | R/W | [1] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[1] (xperh). |
|  |  | R/W | [0] |  | Enable mute function to respond to SMEAS_STATUS_RANGE[0] (xperv). |
| SMEAS_H_NUM_LINES | 011A | R/W | [7:0] | 00 | Number of lines to measure for Horizontal period per Xclks, actual value $=$ programmed value +1 . Range 1-256. Provides for a more accurate measurement. |
| SMEAS_H_SKIP_L | 011B | R/W | $[7: 0]$ | $00$ | Number of horizontal reference edges to skip from selected vertical reference edge before starting horizontal measurement. |
| SMEAS_H_SKIP_U | 011C | R/W | [3:0] | 00 |  |
| SMEAS_HV_SKEW ${ }^{\text {b }}$ | 011D | R | [7:0] |  | Returns the minimum number of xclks between edges of the selected hsync and vsync. Does not care about polarity. Free running, updates once per frame. |
| SMEAS_XK_HTOL_EXP | 012C | R/W | [3:0] | 00 | Horizontal tolerance value. +/- $2^{\wedge} \mathrm{n}$ xclks, $\mathrm{n}=[0 . .15]$ |
| SMEAS_XK_VTOL_EXP | 012D | R/W | [3:0] | 00 | Vertical tolerance value. $+/-2^{\wedge} n x c l k s$, $x k \_$v_high counter use this tolerance value as well. $n=[0 . .15]$ |
| SMEAS_HSYNC_VTOL | 012E | R/W | [3:0] | 00 | Horizontal per Vertical tolerance value. +/-n H(rising,falling) per V(rising,falling) |
| SMEAS_FASTMU_CTRL | 0130 | R/W | [6:5] | 00 | fastmute coast <br> $0^{*}$ : Ilk coast (normal) <br> 1: inverted venab <br> 2, 3: no coast (always active) |
|  |  | R/W | [2:1] |  | error count <br> $0^{*}$ : first error sets fastmute flag <br> 1: two errors in a row needed to set fastmute <br> 2: three errors in a row needed to set fastmute <br> 3: reserved |
|  |  | R/W | [0] |  | fastmute enable |
| SMEAS_POL | 0131 | R | [1] |  | Horizontal polarity 0 : active low (-), $1=$ active high (+) |
|  |  | R | [0] |  | Vertical polarity 0 : active low (-), $1=$ active high (+) |
| SMEAS_FASTMU_TOL | 0134 | R/W | [3:0] | 00 | Tolerance for fast mute check +/-n xclks, $\mathrm{n}=[0 . .15]$ |

Table 17: SMEAS Register Definitions (Sheet 5 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_STATUS_MASK2 | 0135 | R/W | [1] | 00 | Enable mute function to respond to SMEAS_STATUS_RANGE2[1]. |
|  |  | R/W | [0] |  | Enable mute function to respond to SMEAS_STATUS_RANGE2[0]. |
| SMEAS_ACT_POLLING | 013F | R | [0] | 00 | Activity detection polling bit. <br> Toggles when new results are ready in free-run. Undefined in one-shot mode. |
| SMEAS_ANA_ACT | 0140 | R | [7] | 00 | SOG2 is active |
|  |  | R | [6] |  | SOG1 is active |
|  |  | R | [5] |  | SOG0 is active |
|  |  | R | [4] |  | EXT_SOG pin is active |
|  |  | R | [3] |  | Comp vsync from EXT_SOG pin is active |
|  |  | R | [2] |  | Comp vsync from HSYNC pin is active |
|  |  | R | [1] |  | HSYNC pin is active |
|  |  | R | [0] |  | VSYNC pin is active |
| SMEAS_SOG_DLY12 | 0141 | R | [7:4] | 00 | d2: delay in xclks between SOG1 \& SOG2 falling edges |
|  |  | R | [3:0] |  | d1: delay in xclks between SOG0 \& SOG1 falling edges |
| SMEAS_SOG_DLY34 | 0142 | R | [7:4] | 00 | d4: delay in xclks between SOG1 \& SOG0 rising edges |
|  |  | R | [3:0] |  | d3: delay in xclks between SOG2 \& SOG1 rising edges |
| SMEAS_ANA_STUCK | 0143 | R | [4] | 00 | EXT_SOG is stuck at 1 (high)/0(low) |
|  |  | R | [3] |  | Comp vsync from EXT_SOG is stuck at 1(high)/0(low) |
|  |  | R | [2] |  | Comp vsync from HSYNC pin is stuck at 1(high)/0(low) |
|  |  | R | [1] |  | HSYNC pin is stuck at 1(high)/0(low) |
|  |  | R | [0] |  | VSYNC pin is stuck at 1(high)/0(low) |
| $\begin{aligned} & \text { SMEAS_XK_PER_H_L } \\ & \text { SMEAS_XK_PER_H_M } \\ & \text { SMEAS_XK_PER_H_U } \end{aligned}$ | $\begin{array}{\|l\|} \hline 0146 \\ 0147 \\ 0148 \end{array}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & \hline 00 \\ & 00 \\ & 00 \end{aligned}$ | Xclks per Horizontal [7:0] (result = actual - 2) <br> Xclks per Horizontal [15:8] <br> Xclks per Horizontal [23:16] |
| $\begin{aligned} & \text { SMEAS_XK_PER_V_L } \\ & \text { SMEAS_XK_PER_V_M } \\ & \text { SMEAS_XK_PER_V_U } \end{aligned}$ | $\begin{aligned} & 0149 \\ & 014 \mathrm{~A} \\ & 014 \mathrm{~B} \end{aligned}$ | $\begin{aligned} & \hline R \\ & R \\ & R \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & \hline 00 \\ & 00 \\ & 00 \end{aligned}$ | Xclks per Vertical [7:0] <br> Xclks per Vertical [15:8] <br> Xclks per Vertical [23:16] |
| SMEAS_H_PER_V_L SMEAS_H_PER_V_U | $\begin{aligned} & 014 C \\ & 014 D \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{array}{\|l\|} \hline[7: 0] \\ {[7: 0]} \end{array}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Horizontal per Vertical [7:0] <br> Horizontal per Vertical [15:8] |
| $\begin{aligned} & \text { SMEAS_XK_V_HI_L } \\ & \text { SMEAS_XK_V_HI_M } \\ & \text { SMEAS_XK_V_HI_U } \end{aligned}$ | $\begin{aligned} & \hline 014 \mathrm{E} \\ & 014 \mathrm{~F} \\ & 0150 \end{aligned}$ | $\begin{aligned} & \hline \mathrm{R} \\ & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & \hline 00 \\ & 00 \\ & 00 \end{aligned}$ | Xclks per V high |
| SMEAS_REF_FASTMU_L SMEAS_REF_FASTMU_U | $\begin{aligned} & 0132 \\ & 0133 \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Fastmute reference, xclks per hsync, one line only |
| SMEAS_STATUS_TMOT | 0151 | R | [1] | 00 | Indicates that the horizontal measurement timed out. Can only be cleared by sync reset or smeas all_clear. |
|  |  | R | [0] |  | Indicates that the vertical measurement timed out. Can only be cleared by sync reset or smeas all_clear. |

Table 17: SMEAS Register Definitions (Sheet 6 of 6)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMEAS_STATUS_RANGE | 0152 | R | [7] | 00 | The meas_sticky_status bit is an OR of the STATUS_MASK bits ANDed with their corresponding non-sticky range status flags. This bit is sticky and can only be cleared by a write to SMEAS_CLEAR[1]. The sticky bit goes to the scaler as a signal to blank the output screen. |
|  |  | R | [6] |  | Indicates that the hpol measurement does not currently equal the reference value. Not sticky. |
|  |  | R | [5] |  | Indicates that the vpol measurement does not currently equal the reference value. Not sticky. |
|  |  | R | [4] |  | Indicates that the fastmute measurement is currently exceeding the ref+tol. Not sticky. |
|  |  | R | [3] |  | Indicates that the xclks per vhi measurement is currently exceeding the ref+tol. Not sticky. |
|  |  | R | [2] |  | Indicates that the horizontal per vertical measurement is currently exceeding the ref+tol. Not sticky. |
|  |  | R | [1] |  | Indicates that the xclks per horizontal measurement is currently exceeding the ref+tol. Not sticky. |
|  |  | R | [0] |  | Indicates that the xclks per vertical measurement is currently exceeding the ref+tol. Not sticky. |
| SMEAS_MEAS_POLLING | 0153 | R | [1] | 00 | Horizontal measurement polling bit. <br> Toggles upon completion of each measurement in free-run mode while SMEAS_H_CTRL[1] = 1. Undefined in one-shot mode. |
|  |  | R | [0] |  | Vertical measurement polling bit. <br> Toggles upon completion of each measurement in free-run mode while SMEAS_V_CTRL[1] = 1. Undefined in one-shot mode. |
| SMEAS_STATUS_RANGE2 | 0155 | R | [1] |  | indicates the current state of the line buffer pointer crossing error check in the scaler. |
|  |  | R | [0] |  | indicates the current state of the output sequencer trigger-out-of-range error check |

a. The Mask can apply in any mode of operation, it does not need to only apply to the Sticky bit setting.
b. Adjust VSYNC delay and/or filtering in the SRT block to achieve an hv_skew >= 6 to prevent vsync jitter sensitivity in the SMUX and SMEAS blocks.

### 4.8 Sync Multiplexer (SMUX)

The SMUX block provides the ability to:

- Clamp (ADC Black level capture) pulse generation.
- Generate Data Enable from incoming HSync/Vsync signals.
- Select which sync source is used as internal reference.

Vertical enable (venab) and clamp are always generated.

Synthesized signals are generated relative to the reference signal and selected edge.
Clean picture position wrap around is supported in both horizontal and vertical directions (+/- half a line in horizontal and +/- half a frame in vertical).
Programmed position and size values must be less than the respective horizontal and vertical totals.
Figure 8: Block Diagram


Table 18: Sync Multiplexer Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMUX_CTRL_0 | 0200 | R | [7] | 00 | toggle on vsync edge as programmed in bit 5 |
|  |  | R/W | [6] |  | $0^{*}$ : clamp on all lines <br> 1: clamp not during coast |
|  |  | R/W | [5] |  | v edge select <br> 0 *: falling <br> 1: rising |
|  |  | R/W | [4] |  | $h$ edge select $0^{*}$ : falling <br> 1: rising |
|  |  | R/W | [3:0] |  | input select <br> 0*: Ilk_HSync, srt_vsync (normal) <br> 1: HSYNC input signal, VSYNC input signal <br> 2-E: reserved <br> F: HSync = TCON.SRTD6 output <br> VSync $=$ TCON.SRTD7 output |

Table 18: Sync Multiplexer Registers (Sheet 2 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SMUX_CTRL_1 | 0201 | R/W | [7] | 00 | shadow event edge select 0*: falling <br> 1: rising |
|  |  | R/W | [6:4] |  | register shadow event <br> $0^{*}$ : no event (upper byte write) <br> 1: in_venab <br> 2: in_enab <br> 3: vtrigger <br> 4: vtrigger count $\neq 0$ <br> 5-7: reserved |
|  |  | R/W | [3:0] |  | output select <br> Must be set to 0 |
| $\begin{aligned} & \text { SMUX_CLAMP_POS_L } \\ & \text { SMUX_CLAMP_POS_U } \end{aligned}$ | $\begin{aligned} & 0202 \\ & 0203 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | 00 <br> 00 | clamp pulse position relative to HSync reference edge |
| SMUX_CLAMP_WIDTH_L <br> SMUX_CLAMP_WIDTH_U | $\begin{aligned} & 0204 \\ & 0205 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | clamp width in inclks |
| $\begin{aligned} & \text { SMUX_HPOS_L } \\ & \text { SMUX_HPOS_U } \end{aligned}$ | $\begin{aligned} & 0206 \\ & 0207 \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | 00 <br> 00 | horizontal data position relative to HSync reference edge |
| SMUX_HPIX_L <br> SMUX_HPIX_U | $\begin{aligned} & 0208 \\ & 0209 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | horizontal data width |
| SMUX_VPOS_L <br> SMUX_VPOS_U | 020A 020B | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | 00 <br> 00 | vertical trigger position in lines relative to vsync reference edge. Should be used for changing position to minimize screen glitches. |
| SMUX_VPIX_L <br> SMUX_VPIX_U | $\begin{aligned} & \text { 020C } \\ & \text { 020D } \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | vertical data height |
| SMUX_VTRIG_L <br> SMUX_VTRIG_U | $\begin{aligned} & \text { 020E } \\ & \text { 020F } \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | 00 <br> 00 | delay in lines from smux_vpos to the first line of a new frame |
| SMUX_CLAMP_POS_HW_L SMUX_CLAMP_POS_HW_U | $\begin{aligned} & 0210 \\ & 0211 \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | shadow readback |
| SMUX_CLAMP_WIDTH_HW_L SMUX_CLAMP_WIDTH_HW_U | $\begin{aligned} & 0212 \\ & 0213 \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | shadow readback |
| SMUX_HPOS_HW_L <br> SMUX_HPOS_HW_U | $\begin{aligned} & 0214 \\ & 0215 \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | shadow readback |
| SMUX_HPIX_HW_L <br> SMUX_HPIX_HW_U | $\begin{aligned} & 0216 \\ & 0217 \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | shadow readback |

Table 18: Sync Multiplexer Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| SMUX_VPOS_HW_L | 0218 | R | $[7: 0]$ | 00 | shadow readback |
| SMUX_VPOS_HW_U | 0219 | $R$ | $[3: 0]$ | 00 |  |
| SMUX_VPIX_HW_L | 021 A | R | $[7: 0]$ | 00 | shadow readback |
| SMUX_VPIX_HW_U | 021 B | R | $[3: 0]$ | 00 |  |
| SMUX_VTRIG_HW_L | 021C | R | $[7: 0]$ | 00 | shadow readback |
| SMUX_VTRIG_HW_U | 021D | R | $[3: 0]$ | 00 |  |

Note: A shadow readback register retains the previously programmed value until the relevant event occurs. There is one shadow readback register for each register in the SMUX block.

Table 19: Horizontal Parameters


Table 20: Vertical Parameters


### 4.9 Data Measurement (DMEAS)

DMEAS provides a number of pixel measurement functions for autosetup (find the best phase, ADC sampling clock, picture auto-position) and autocolor (autolevel, ADC analog range tuning for black and white calibration).
Most DMEAS measurement functions are performed within a programmable input image boundary defined by the top left and bottom right window coordinate registers. The image boundary can be full screen.
DMEAS also includes an annex block called DE Size and is decribed at the end of this spec.
All unused or reserved bits will return as zero.
The DMEAS block only processes the 7 MSBs of the 10 bit ADC outputs. Consequently the maximum pixel value seen by DMEAS is FE.
The horizontal and vertical position measurements are relative to the selected reference sync edges and must be offset before programming SMUX image position, refer to Chapter 4.8: Sync Multiplexer (SMUX) for more information.

### 4.9.1 Function Summary

The algorithms grouped together are executed simultaneously.
The Color, Threshold, Mode Control, Window Control, and Output registers are shared for all measurements, and are used according to the algorithm selected to measure.

| Algorithm | Mode Ctrl | Result | Color | Thresh | Window Control |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Edge Intensity 00 Pixel Sum 00 | One Shot One Shot | 32 bit edge_out 32 bit psum_out | R/G/B/All R/G/B/All | Yes <br> No | Yes Yes |
| Min / Max 01 <br> Pcd 01 | One Shot One Shot | 8 bit min / 8 bit max 24 bit pcd_out | R/G/B/All <br> R/G/B/All | No Yes | Yes <br> Yes |
| Hpos / Vpos 02 | One Shot | 12 bit Hpos_Min 12 bit Hpos_Max 12 bit Vpos_Min 12 bit Vpos_Max | All | Yes | Yes |
| De_Size 03 | One Shot / Free Run | 16 bit De_Size_out <br> 1 bit De_Mismatch | None | No | No |

### 4.9.2 Window Control

All measurements occur within a window in a single frame. The window is defined by the upper left ( $\min \_x$, min_y) and lower right (max_x, max_y) corners (inclusive). Window coordinates are relative to Sync pulses. A window defined from ( 0,0 ) - (FFF, FFF) would go from sync to sync (full screen). The sync reference edge selection is programmable.

### 4.9.3 Algorithm Control

The available measurements are described in detail below. Most algorithms can be run over each or all color channels. Most algorithms also contain a threshold value to zero out noise and / or amplify edges.

Algorithm, Color, Threshold, or Window Control changes should be made at the end of a valid measurement, otherwise they will corrupt the current measurement in progress:

- set DMEAS parameters for the desired measurement
- start the measurement (don't change the parameters above)
- wait until measurement completion.


### 4.9.4 Mode Control

All measurements (except De_Size) are performed in One Shot mode. For De_Size measurement, software can request measurements in one of two ways:

- One Shot - synchronous with respect to the Micro Controller
- Free Run - asynchronous with respect to the Micro Controller

Note: The block indicates when a measurement is valid.

- In One Shot mode the measurement is completed through an Auto Clear of the Start condition.
- In Free Run mode when the measurement is completed a polling bit toggles. A freeze bit is provided to freeze the results. Measurements still continue while result registers are frozen.


### 4.9.5 Edge Intensity

The Edge Intensity measurement is the sum of the absolute value of the delta between adjacent pixels. A programmable threshold is applied to zero out noise and amplify edges.

## Equation:

Delta_val = abs(pixelA - pixelB) - threshold;
Delta_val = Delta_val < 0 ? 0: Delta_val;
Sum += Delta_val;
For all 3 color channels:
Sum += Delta_val on Red channel + Delta_val on Green channel + Delta_val on Blue channel
The measurement includes all transitions inside the defined window.
Measurement Window: The Edge Intensity is computed over a defined window as described in Window Control.
Color Channels: A specific color channel (R/G/B) or all color channels (All) can be applied to the Edge Intensity.
Result: The output at the end of the measurement is a 32-bit number.

### 4.9.6 Pixel Sum

The Pixel Sum is the sum of all selected pixels for either a specific color channel or all color channels.

Measurement Window: The Pixel sum is computed over a defined window as described in Window Control.
Color Channels: A specific color channel (R/G/B) or all color channels (All) can be applied to the Pixel Sum.
Result: The output at the end of the measurement is a 32 bit number.

### 4.9.7 Min / Max

The Min / Max reports the minimum and maximum pixel found.

Measurement Window: The Min / Max value is found over a defined window as described in Window Control.

Color Channels: A specific color channel (R/G/B) or all color channels (All) can be applied to the Min / Max value.

Result: The output at the end of the measurement is two 8 bit numbers, the Minimum Pixel value and the Maximum Pixel value.

### 4.9.8 Pixel Cumulative Distribution (PCD)

PCD function reports the total number of pixels greater than (or less than) a programmable threshold.

To switch between pixels greater than or pixel less than the threshold, a control bit is provided in the Mode register when requesting a measurement.
Measurement Window: The PCD value is calculated over a defined window as described in Window Control.

Color Channels: A specific color channel (R/G/B) or all color channels (All) can be applied to the PCD function.

Result: The output at the end of the measurement is a 24 bit number.

### 4.9.9 H Position Min / Max

Horizontal position measures the start and end of video data in inclks relative to the reference edge of HSync.

Data horizontal start is defined as the number of inclks between the selected edge of HSync and the "first data pixel".
First data pixel definition is either:

1. First pixel > a programmable threshold value (normal)
2. First pixel with the absolute value (current pixel - previous pixel) is > a programmable threshold value
Data horizontal end is defined as the number of inclks between reference edge of HSync and the "last data pixel plus one". The search for the last pixel ends at the end of a window.
Last data pixel plus one is either:
3. Pixel after the last pixel that is > a programmable threshold value (normal)
4. Last pixel with the absolute value (current pixel - previous pixel) is > a programmable threshold value
To switch between the two threshold methods used in the first and last pixel, a control bit is provided in the DMEAS_MODE_CTRL register when requesting a measurement.
The first and last pixels are measured for each line, and the earliest first and latest last for the selected pixel area are reported out at the end of the measurement.
Measurement Window: The First / Last pixel on a line is found over a defined window as described in Window Control.

Color Channels: All color channels are used to find the First / Last pixel on a line.
Result: The output at the end of the measurement is two 12 bit numbers, H position Min and H position Max.

### 4.9.10 V Position Min / Max

Vertical position measures the start and end of video data in lines relative to the reference edge of vsync.
Data vertical start is defined as the number of lines between the selected edge of vsync and the "first data pixel".
First data pixel definition is either:

1. First pixel > a programmable threshold value (normal)
2. First pixel with the absolute value (current pixel - previous pixel) is > a programmable threshold value

Data vertical end is defined as the number of lines between reference edge of vsync and the "last data pixel plus one". The search for the last pixel ends at the end of a window.
Last data pixel plus one is either:

1. Pixel after the last pixel that is > a programmable threshold value (normal)
2. Last pixel with the absolute value (current pixel - previous pixel) is > a programmable threshold value
To switch between the two threshold methods used in the first and last pixel, a control bit is provided in the DMEAS_MODE_CTRL register when requesting a measurement.
Measurement Window: The selected pixel area range for $y$ the range is vsync to vsync. The selected range for x is not applicable.
Color Channels: All color channels are used to find the First / Last line in a frame.
Result: The output at the end of the measurement is two 12 bit numbers, V position Min and V position Max.

### 4.9.11 DE Size

DE Size measures the number of inclks per $D E$.
At the end of the measurement (DE falling edge), the measured value is compared to a programmed expected value +/- a programmed threshold. If the expected value is within the threshold the DE_size_mismatch flag is not set. If the measured size is outside of the threshold the DE_size_mismatch flag is set.
In free run mode the results are updated every line. The DE_size_mismatch flag is set at DE falling edge and reset at DE rising edge.
In One shot mode the results are updated once and stay that way until they are cleared by software. The DE_size_mismatch flag can only be cleared when the reset flag bit is set by software.
Result: 16 bit measured value.

Table 21: DMEAS Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| DMEAS_AEC_CTRL | 0900 | R/W | [7:6] | 00 | lolor selection <br> 00*: All <br> 01: Red <br> 10: Green <br> 11: Blue |

Table 21: DMEAS Registers (Sheet 2 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DMEAS_MODE_CTRL | 0901 | R/W | [7] | 00 | DE reset <br> $0^{*}$ : do not reset the de_mismatch_flag <br> 1: reset the de_mismatch_flag |
|  |  | R/W | [6] |  | DE freeze <br> 0 *: update I2C registers after every measurement in free run mode <br> 1: freeze DE size results in I2C registers and do not update while this bit is active |
|  |  | R/W | [5] |  | DE one shot <br> $0^{*}$ : free run mode. <br> 1: one shot mode. <br> Applies only to DE_Size measurement. All other measurements are always in One_shot mode. |
|  |  | R/W | [3] |  | h_v_pos_sel / pcd_sel <br> - if algorithm = 01 (pcd_sel) <br> $0^{*}$ : pixel < threshold <br> 1: pixel >= threshold <br> - if algorithm = 10 (h_v_pos_sel) <br> 0*: pixel > threshold (normal) <br> 1: abs (pixel - previous pixel) > threshold |
|  |  | R/W | [2] |  | DMEAS all clear <br> All internal result registers are cleared when this bit is set. This bit is self clearing. |
|  |  | R | [1] |  | DMEAS polling bit. <br> Toggles at the end of each measurement in free-run mode. Undefined in one-shot mode. |
|  |  | R/W | [0] |  | DMEAS start <br> Data measurement start. This bit is auto-cleared by HW when the measurement is completed. |
| DMEAS_THRESHOLD | 0902 | R/W | [7:0] | 00 | Threshold value to use for selected algorithm. |
| DMEAS_WIN_MIN_X_L DMEAS_WIN_MIN_X_U | $\begin{aligned} & 0903 \\ & 0904 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Minimum X for window control to use with all algorithms. |
| DMEAS_WIN_MAX_X_L DMEAS_WIN_MAX_X_U | $\begin{aligned} & 0905 \\ & 0906 \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \text { FF } \\ & 00 \end{aligned}$ | Maximum X for window control to use with all algorithms. |
| DMEAS_WIN_MIN_Y_L DMEAS_WIN_MIN_Y_U | $\begin{aligned} & 0907 \\ & 0908 \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Minimum Y for window control to use with all algorithms. |
| DMEAS_WIN_MAX_Y_L DMEAS_WIN_MAX_Y_U | $\begin{aligned} & 0909 \\ & 090 \mathrm{~A} \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \text { FF } \\ & 00 \end{aligned}$ | Maximum Y for window control to use with all algorithms. |
| DMEAS_DE_REF_L DMEAS_DE_REF_H | $\begin{aligned} & 090 \mathrm{~B} \\ & 090 \mathrm{C} \end{aligned}$ | $\begin{aligned} & \mathrm{R} \\ & \mathrm{R} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | DE size expected result |
| DMEAS_DE_TOL | 090D | R | [7:0] | 00 | DE tolerance value |

Table 21: DMEAS Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DMEAS_DATA_0 | 090E | R | [7:0] | 00 | Refer to Table 22 below |
| DMEAS_DATA_1 | 090F | R | [7:0] | 00 |  |
| DMEAS_DATA_2 | 0910 | R | [7:0] | 00 |  |
| DMEAS_DATA_3 | 0911 | R | [7:0] | 00 |  |
| DMEAS_DATA_4 | 0912 | R | [7:0] | 00 |  |
| DMEAS_DATA_5 | 0913 | R | [7:0] | 00 |  |
| DMEAS_DATA_6 | 0914 | R | [7:0] | 00 |  |
| DMEAS_DATA_7 | 0915 | R | [7:0] | 00 |  |

Table 22: DMEAS Output Registers Assignment

|  | alg_sel = 00 | alg_sel = 01 | alg_sel = 10 | alg_sel = 11 |
| :--- | :--- | :--- | :--- | :--- |
| DMEAS_DATA_0 | edge_out [7:0] | min_out [7:0] | hpos_min [7:0] | de_size_out [7:0] |
| DMEAS_DATA_1 | edge_out [15:8] | max_out [7:0] | hpos_min [11:8] | de_size_out [15:8] |
| DMEAS_DATA_2 | edge_out [23:16] | pcd_out [7:0] | hpos_max [7:0] | de_mismatch_flag |
| DMEAS_DATA_3 | edge_out [31:24] | pcd_out [15:8] | hpos_max [11:8] | N/A |
| DMEAS_DATA_4 | psum_out [7:0] | pcd_out [23:16] | vpos_min [7:0] | N/A |
| DMEAS_DATA_5 | psum_out [15:8] | N/A | vpos_min [11:8] | N/A |
| DMEAS_DATA_6 | psum_out [23:16] | $\mathrm{N} / \mathrm{A}$ | vpos_max [7:0] | $\mathrm{N} / \mathrm{A}$ |
| DMEAS_DATA_7 | psum_out [31:24] | $\mathrm{N} / \mathrm{A}$ | vpos_max [11:8] | $\mathrm{N} / \mathrm{A}$ |

### 4.10 Scale (SCL)

ADE scales input video to output panel resolution without external video frame memory. This requires tuning of the panel timing parameters to make the vertical active time match the panel's.

Features:

- Separable $3 \mathrm{~V} \times 4 \mathrm{H}$ polyphase filter:
- 3 line filter for H resolutions $<=1024$
- 2 line filter for H resolutions > 1024
- independent H \& V kernel register storage
- 64 phases are interpolated from 6 V or 10 H reference points
- symmetric kernels only
- coefficients range from -2 to +1 63/64
- Simple pointer collision feedback mechanism
- 2-way $3^{\text {rd }}$ generation context sensitive filtering
- Background color management

For formulae to program the registers refer to Chapter 7: Scaler Equations on page 132.

### 4.10.1 Frame Synchronization

Due to the limited pixel memory of the chip, the output active video needs to be perfectly synchronized with the input active video. This mode of operation is called Frame Lock.

Figure 9: Frame Lock Operation


Table 23: Scale Registers (Sheet 1 of 3)

| Register Name | Addr | R/W | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| SCL_SRC_HPIX_L | $0 A 00$ | R/W | $[7: 0]$ | 00 | input horizontal resolution |
| SCL_SRC_HPIX_U | $0 A 01$ | R/W | $[3: 0]$ | 00 |  |
| SCL_SRC_VPIX_L | $0 A 02$ | R/W | $[7: 0]$ | 00 | input vertical resolution |
| SCL_SRC_VPIX_U | $0 A 03$ | R/W | $[3: 0]$ | 00 |  |
| SCL_SCALEFACH_L | $0 A 04$ | R/W | $[7: 0]$ | 00 | 17-bit horizontal scale factor |
| SCL_SCALEFACH_M | $0 A 05$ | R/W | $[7: 0]$ | 00 |  |
| SCL_SCALEFACH_U | $0 A 06$ | R/W | $[0]$ | 00 |  |
| SCL_SCALEFACV_L | $0 A 07$ | R/W | $[7: 0]$ | 00 | 17-bit vertical scale factor |
| SCL_SCALEFACV_M | 0A08 | R/W | $[7: 0]$ | 00 |  |
| SCL_SCALEFACV_U | 0A09 | R/W | $[0]$ | 00 |  |
| SCL_ORIGHPOS_L | 0A0A | R/W | $[7: 0]$ | 00 | 2's complement , signed number |
| SCL_ORIGHPOS_U | 0A0B | R/W | $[7: 0]$ | 00 | 27-bit horizontal position of the first output pixel |

Table 23: Scale Registers (Sheet 2 of 3)

| Register Name | Addr | R/W | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SCL_H_KERNEL_0 | 0A10 | R/W | [7:0] | 00 | Horizontal filter kernel <br> 2's complement, signed numbers ranging from -2 to +1 63/64 |
| SCL_H_KERNEL_1 | 0A11 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_2 | 0A12 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_3 | 0A13 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_4 | 0A14 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_5 | 0A15 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_6 | 0A16 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_7 | 0A17 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_8 | 0A18 | R/W | [7:0] | 00 |  |
| SCL_H_KERNEL_NORM | OA19 | R/W | [7:0] | 40 | 2's complement, signed number used to normalize the H filter kernel (usually 64) |
| SCL_V_KERNEL_0 | 0A1A | R/W | [7:0] | 00 | Vertical filter kernel <br> 2's complement, signed numbers ranging from -2 to +1 63/64 <br> Has a $1 / 2$ line shift compared to hkernel and must be programmed to a 2 line kernel when in_hpixel > 1024. |
| SCL_V_KERNEL_1 | 0A1B | R/W | [7:0] | 00 |  |
| SCL_V_KERNEL_2 | 0A1C | R/W | [7:0] | 00 |  |
| SCL_V_KERNEL_3 | 0A1D | R/W | [7:0] | 00 |  |
| SCL_V_KERNEL_4 | OA1E | R/W | [7:0] | 00 |  |
| SCL_V_KERNEL_NORM | 0A1F | R/W | [7:0] | 40 | 2's complement, signed number used to normalize the V filter kernels (usually 64) |
| SCL_BGCOLOR_B | 0A20 | R/W | [7:0] | 00 | Blue component of background color, refer to Figure 21 |
| SCL_BGCOLOR_G | 0A21 | R/W | [7:0] | 00 | Green component of background color, refer to Figure 21 |
| SCL_BGCOLOR_R | 0A22 | R/W | [7:0] | 00 | Red component of background color, refer to Figure 21 |
| SCL_BGCOLOR_CTRL | 0A23 | R/W | [7] | 00 | 1: force background color |
|  |  | R/W | [4] |  | Mute color select: <br> $0^{*}$ : black <br> 1: use background color when SMEAS_STATUS_RANGE[7] is high |
|  |  | R/W | [3:2] |  | $0 *$ : line replicate 1 <br> 1: line replicate 2 <br> 2: line replicate 3 <br> 3: vertical border blend |
|  |  | R/W | [1:0] |  | $0^{*}$ : pixel replicate 1 <br> 1: pixel replicate 2 <br> 2: pixel replicate 3 <br> 3: horizontal border blend |
| SCL_PTR_PRE_L | 0A24 | R | $[7: 0]$ | 00 | The minimum difference of the write pointer and the first of three read pointers; updated every frame LSB $=4$ pixels <br> Not valid when SCL_CONTROL[3] = 1 |
| SCL_PTR_PRE_U | 0A25 | R | [3:0] | 00 |  |

Table 23: Scale Registers (Sheet 3 of 3)

\begin{tabular}{|c|c|c|c|c|c|}
\hline Register Name \& Addr \& R/W \& Bits \& Rst \& Description \\
\hline \[
\begin{aligned}
\& \text { SCL_PTR_POST_L } \\
\& \text { SCL_PTR_POST_U }
\end{aligned}
\] \& \begin{tabular}{l}
0A26 \\
0A27
\end{tabular} \& R

$R$ \& $$
[7: 0]
$$

[3:0] \& | 00 |
| :--- |
| 00 | \& The minimum difference of the write pointer and the last of three read pointers; updated every frame LSB $=4$ pixels <br>

\hline \multirow[t]{5}{*}{SCL_CONTROL} \& \multirow[t]{5}{*}{OA28} \& R/W \& [4] \& \multirow[t]{5}{*}{00} \& allow trigger delay count to be retriggered by SMUX vtrig (normal $=0$ ) <br>

\hline \& \& R/W \& [3] \& \& | use two tap vertical filter $0^{*}$ : in hpixel <= 1024 |
| :--- |
| 1: in_hpixel > 1024 |
| ptr_pre is invalid in this case | <br>

\hline \& \& R/W \& [2] \& \& allow output sequencer to be retriggered before vtotal_min (normal $=0$ ) <br>
\hline \& \& R/W \& [1] \& \& completes the current frame then stops the sequencer. Poll the vcount register to determine when frame has stopped. <br>
\hline \& \& R/W \& [0] \& \& enable scaler timing engine (output sequencer) <br>
\hline SCL_VCOUNT \& 0A29 \& R \& [7:0] \& \& output sequencer vertical counter >> 4 <br>

\hline | SCL_HTOTAL_L |
| :--- |
| SCL_HTOTAL_U | \& | 0A2A |
| :--- |
| 0A2B | \& | R/W |
| :--- |
| R/W | \& \[

[7: 0]
\]

[3:0] \& \begin{tabular}{l}
00 <br>
00

 \& 

desired output htotal - 1 <br>
out_htotal should be even <br>
note: out_henab should be a multiple of 4 for RSDS dual
\end{tabular} <br>

\hline | SCL_VTOTAL_MIN_L |
| :--- |
| SCL_VTOTAL_MIN_U | \& \[

$$
\begin{aligned}
& \text { OA2C } \\
& \text { OA2D }
\end{aligned}
$$

\] \& | R/W |
| :--- |
| R/W | \& \[

$$
\begin{aligned}
& {[7: 0]} \\
& {[3: 0]}
\end{aligned}
$$

\] \& \[

$$
\begin{aligned}
& 00 \\
& 00
\end{aligned}
$$
\] \& minimum vcount before a new frame can be started <br>

\hline \[
$$
\begin{aligned}
& \text { SCL_VTOTAL_MAX_L } \\
& \text { SCL_VTOTAL_MAX_U }
\end{aligned}
$$

\] \& | OA2E |
| :--- |
| 0A2F | \& | R/W |
| :--- |
| R/W | \& \[

$$
\begin{aligned}
& {[7: 0]} \\
& {[3: 0]}
\end{aligned}
$$

\] \& | 00 |
| :--- |
| 00 | \& vcount at which the output sequencer will self trigger to maintain a minimum frame rate to the panel <br>


\hline | SCL_TRIGGER_DLY_L |
| :--- |
| SCL_TRIGGER_DLY_M |
| SCL_TRIGGER_DLY_U | \& \[

$$
\begin{aligned}
& \text { OA30 } \\
& \text { OA31 } \\
& \text { OA32 }
\end{aligned}
$$

\] \& | R/W |
| :--- |
| R/W |
| R/W | \& \[

$$
\begin{aligned}
& {[7: 0]} \\
& {[7: 0]} \\
& {[3: 0]}
\end{aligned}
$$

\] \& \[

$$
\begin{aligned}
& 00 \\
& 00 \\
& 00
\end{aligned}
$$
\] \& time in xclks from SMUX vtrig to when vcount/hcount of the output sequencer are reset to 0,0 . <br>

\hline | SCL_LINE_START_L |
| :--- |
| SCL_LINE_START_U | \& | 0A33 |
| :--- |
| 0A34 | \& | R/W |
| :--- |
| R/W | \& [7:0]

[3:0] \& \& | pipe start value |
| :--- |
| $=4.5$-origin_hpos*4096/scale_factor_h -(21.5+5*4096/ pipe_rate)*sclk_period/dotclk_period |
| If pipe_rate $=0$, use 4096 . | <br>

\hline \multirow[t]{2}{*}{SCL_CONTEXT_0} \& \multirow[t]{2}{*}{OA35} \& R/W \& [6:1] \& \multirow[t]{2}{*}{00} \& context sharp offset <br>
\hline \& \& R/W \& [0] \& \& enable context function (normal) <br>
\hline \multirow[t]{3}{*}{SCL_CONTEXT_1} \& \multirow[t]{3}{*}{0A36} \& R/W \& [7:6] \& \multirow[t]{3}{*}{00} \& context smooth slope (recommended $=1,2,3$ ) <br>
\hline \& \& R/W \& [5:4] \& \& context sharp slope <br>
\hline \& \& R/W \& [3:0] \& \& context sharp clip <br>
\hline
\end{tabular}

For proper scale operation, the SCLK frequency must be programmed so that:

1 SCLK_FREQ is greater than the max of DCLK_FREQ and (I N_HPI XEL x DCLK_FREQ) / DEST_HPI XEL;
2 SCLK_FREQ < 140 MHz
3 SCL_LINE_START > 0; and
4 SCL_PIPE_RATE <= 4096
The frame synchronization between input and output can be fine tuned using the line buffer pointer crossing feedback registers, SCL_PTR_PRE and SCL_PTR_POST. By adjusting the SCL_TRIGGER_DLY, pointer crossing can be eliminated.

### 4.10.2 Context Description

The context function allows the scaler to mix the output of three filters (sharp, normal kernel and smooth) on a per pixel basis depending on the local contrast in a 3 V 44 H area. The sharpening suppresses ringing / overshoots.
Those 3 kernels: Smooth, User (defined with H and V kernel coefficients) and Sharp run in parallel and can be blended together to finally generate a panel pixel.

If Context is disabled, only User Kernel is used.
If Context is enabled, then the blending of the 3 kernels follows the diagram below. The horizontal axis is the context:

- Context $0=$ All neighbour pixels $(3 x 3)$ have almost same RGB values (greyscale).
- Context $F=$ All neighbour pixels (3x3) have very different RGB values (1x1 Black and White checker pattern).

Context is used along with I2C programmable coefficients to make the kernel blending ratio, as the drawing below shows.

Refer to the context mixing equations for more details. The vertical axis has 63 steps. $(63=100 \%)$.

Only 2 kernels can be blended together. Smooth wins over Sharp.


## Context Mixing Equations:

contrast $=\max (\max (\mathrm{R} 0, \mathrm{R} 1, .)-.\min (\mathrm{R} 0, \mathrm{R} 1, .),. \max (\mathrm{G} 0, \mathrm{G} 1, .)-.\min (\mathrm{G} 0, \mathrm{G} 1, .),. \max (\mathrm{B} 0, \mathrm{~B} 1, .)-$.
$\min (B 0, B 1, .).), 6 b$ value, [0..63]
sharp_mix $=$ clip((contrast >> (3-sharp_slope)) - sharp_offset, 0 , sharp_clip), 4b value, [0..15]

normal_mix $=16$ - sharp_mix - smooth_mix, [1,,16]
Note: It is recommended to enable the context feature all the time with:

- SCL_CONTEXT_0 = 01
- SCL_CONTEXT_1 = 80


### 4.10.3 Scale Kernel Example

Recommended kernel is:

- Nearest Neighbor for 1X scale modes (no scaling)
- Cubic for > 1X scale modes (upscaling)
- Bilinear for < 1X scal modes (downscaling)

| Register | Address | No Scaling | Down Scaling | Up Scaling |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
|  |  | Nearest Neighbor | Bilinear | "-0.7" Cubic | "-0.5" Cubic |
| H_KERNEL_0 | $0 A 10$ | 00 | 00 | 00 | 00 |


| Register | Address | No Scaling | Down Scaling <br> Bilinear | Up Scaling |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Nearest Neighbor |  | "-0.7" Cubic | "-0.5" Cubic |
| H_KERNEL_1 | 0A11 | 00 | 00 | FE | FE |
| H_KERNEL_2 | 0A12 | 00 | 00 | FA | FC |
| H_KERNEL_3 | OA13 | 00 | 00 | F9 | FB |
| H_KERNEL_4 | OA14 | 00 | 00 | FF | 00 |
| H_KERNEL_5 | OA15 | 00 | 10 | 10 | OF |
| H_KERNEL_6 | OA16 | 20 | 20 | 26 | 24 |
| H_KERNEL_7 | 0A17 | 40 | 30 | 39 | 38 |
| H_KERNEL_8 | OA18 | 40 | 40 | 41 | 40 |
| H_KERNEL_NORM | OA19 | 40 | 40 | 40 | 40 |
| V_KERNEL_0 | 0A1A | 00 | 00 | FB | FA |
| V_KERNEL_1 | 0A1B | 00 | 00 | F9 | FA |
| V_KERNEL_2 | OA1C | 00 | 00 | FF | FE |
| V_KERNEL_3 | 0A1D | 00 | 10 | 10 | OD |
| V_KERNEL_4 | 0A1E | 00 | 20 | 23 | 22 |
| V_KERNEL_NORM | 0A1F | 40 | 40 | 40 | 40 |

Note: Upscaling and downscaling can be simultaneously combined horizontally and vertically.

### 4.11 Pattern Generator (PGEN)

The PGEN block can generate graphic patterns to support debug and test tasks for LCD panels such as horizontal or vertical bicolor stripes, bicolor checkers, color bars, gray scales or color scales. It is also possible to pass through the RGB signal coming from the SCL block.

Note: $\quad$ The PGEN block is located before the sRGB color management block.

### 4.11.1 Overview

The following features of the PGEN block overlap each other like layers, defining display priorities:

- Bars (lowest display priority)
- Cells and Grids
- Borders
- TCON Window Control (highest display priority)

Bars and cells are freely programmable in size and independently of each other.
A border is a horizontal or vertical borderline. If enabled, it has priority over the above settings. Above all, aTCON window, if enabled, restrains all PGEN settings to a given display area.

### 4.11.2 Color Mask Sequencer

### 4.11.2.1 Bars and Groups

A bar is the basic graphic element of the PGEN. A bar group is based on two independently programmable 24 bit RGB colors named CO and C 1 and programmed into:

- For C0: PGEN_PO_COLOR _R_C0, PGEN_PO_COLOR _G_CO, PGEN_PO_COLOR _B_CO
- For C1: PGEN_P0_COLOR _R_C1, PGEN_P0_COLOR _G_C1, PGEN_P0_COLOR _B_C1

Each color C 0 and C 1 is assigned to 1 to 8 consecutive bars. The number of bars minus 1 is programmable in PGEN_P0_MODE, bits [7:5] for C0 and [4:2] for C1:

| Bar group |  |  |  |  |  |  | One group per colour <br> CO and C1 |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Bar 0 | Bar 1 | Bar 2 | Bar 3 | Bar 4 | Bar 5 | Bar 6 | Bar 7 | Up to 8 bars per colour |

### 4.11.2.2 Bar Width, Height and Offset Control

Bar's height and width are programmable, respectively in PGEN_PO_WDTH and PGEN_PO_HGHT (16-bit wide). The actual number of displayed bars depends on the bar width, height and the panel resolution. The bars are numbered in incremental fashion from left to right, top to bottom.
If the combined size of all bars in a group is smaller than the display area, each of the C0 and C1 bar groups is replicated across the display, as long as the bars still fit in the display area:


The height and width of a bar can range anywhere from 1 pixel (checkerboard) to full screen.
Additionally, an offset in both directions can be programmed respectively in registers
PGEN_PO_WDTH_X_OFFSET and PGEN_PO_HGHT_Y_OFFSET. It shifts the top left corner (1st bar of C0 group) off the display area.
Note: $\quad$ The offset value, for each direction, must be less than the corresponding bar size.

### 4.11.2.3 Color Masks

Each bar can filter any R G B component of its assigned C0 or C1 color, by means of 3 mask bits per bar in registers PGEN_PO_SEQ_COLO_COL1 (bars 0 \& 1) thru PGEN_P0_SEQ_COL6_COL7 (bars 6 \& 7). The color is "ANDed" with the mask:

- if either R G B bit is reset, the corresponding colour component is blocked
- if set, the colour component is not blocked


## Example:

PGEN_COLOR_CO_B = PGEN_COLOR _CO_G = PGEN_COLOR _CO_R = FF sets C0 to white PGEN_PO_MODE is set to AC:

- Number of bars in C0 = PGEN_P0_MODE[7:5] +1 = 6 (bars 0 to 5)
- Number of bars in C1 = PGEN_PO_MODE[4:2] +1 = 4 (bars 0 to 3 )

PGEN_PO_SEQ_COLO_COL1 = 42:

- Bar 0 filters $G$ and $B$ components but lets $R$ pass: this 1 st bar is displayed in red
- Bar 1 filters $R$ and $B$ components but lets $G$ pass: this $2 n d$ bar is displayed in green

PGEN_PO_SEQ_COL2_COL3 = 17:

- Bar 2 filters $R$ and $G$ components but lets $B$ pass: this 3rd bar is displayed in blue
- Bar 3 does not filter any of the R G B components: this 4th bar is displayed in white

PGEN_PO_SEQ_COL4_COL5 $=77$ : bars 4 and 5 do not filter R G B and are displayed in white.
PGEN_PO_SEQ_COL6_COL7 is don't care, since a maximum of 6 bars is used by C0 and C1.
Across the display, 6 bars [red] [green] [blue] [white] [white] [white] (from C0 group) are now displayed, followed by 4 bars [red] [green] [blue] [white] (from C1 group), then again 6 bars from CO group etc.. until the right border of the display area is reached:


The bars also repeat vertically.

### 4.11.2.4 Gradient Control

The gradient control registers modify the colors C 0 and C 1 as follows:

- PGEN_PO_GRADDELTA_R: increment the Red value by this register value
- PGEN_PO_GRADDELTA_G: increment the Green value by this register value
- PGEN_PO_GRADDELTA_B: increment the Blue value by this register value
- PGEN_PO_GRADSTEP_X: apply the increment value to each color every $X$ horizontal pixels
- PGEN_P0_GRADSTEP_Y: apply the increment value to each color every $Y$ vertical lines

Note: $\quad$ The values wrap over FF: for example, a value of FF for GRADDELTA will decrease the color by 1 (if GRADDELTA was 50: $50+F F=4 F=G R A D D E L T A-1$ )
All kinds of color shades can be achieved by wisely using the above parameters.

### 4.11.3 $8 \times 8$ Grid Layout with Optional Resets

A cell is a graphic element grouped by 8 in a grid. A set of 8 Grid Registers PGEN_GRIDO to PGEN_GRID7 represents an $8 \times 8$ bitmap where each bit represents one rectangular cell: this makes a total grid of $8 \times 8$ cells.
Each cell either displays the bar pattern defined above, or the input video signal, depending on the value in its corresponding PGEN_GRID register:


All cells have the same size, defined by one horizontal and one vertical grid pitch registers PGEN_GRID_X and PGEN_GRID_Y (16-bit wide).
Additionally, an offset in both directions can be programmed respectively in registers PGEN_PO_WDTH_X_OFFSET and PGEN_PO_HGHT_Y_OFFSET. It shifts the top left corner (1st cell of Grid 0 ) off the display area.

Note: $\quad$ The offset value, for each direction, must be less than the corresponding cell size.

The actual number of displayed cells depends on the programmed cell size:

- If it makes the complete $8 \times 8$ grid bigger than the total display area, only the cells or part of cells that are included in the display area are displayed. Any cell (on the right and bottom sides) outside the display area is ignored and not displayed
- If it makes the complete $8 \times 8$ grid smaller than the total display area, the $8 \times 8$ pattern repeats itself across the entire display area, both vertically and horizontally

Figure 10: $8 \times 8$ Grip Mapping Example


### 4.11.3.1 Cell Reset

When PGEN_PO_MODE[1] bit is set, the bar counters will be reset to bar 0, and gradients color counters will be reset to the default color value CO , each time a new grid cell is reached.
This is to be combined with bar offset settings (refer to Section 4.11.2.2: Bar Width, Height and Offset Control and the example provided hereafter). For example, this will affect all patterns with non-zero values for PGEN_PO_GRADSTEP_X and/or PGEN_PO_GRADSTEP_Y.

### 4.11.3.2 Color CO Replacement

When PGEN_PO_MODE[0] bit is set, the input video signal takes the place of color C 0 . In that case, non-zero gradients will apply the increment value to each R G B color of the input signal, not C0.
Note: If the displayed picture has noticiable jitter, the input $R G B$ values are not stable and may generate heavy noise on screen when the gradient applied to $R G B$ values rolls over from FF to 00.

### 4.11.4 Borders

The border generator adds a single pixel wide borderline to the panel area. There are 4 edges: top, bottom, left and right. Each edge can be enabled independently, and programmed to one of 8 basic colors using a 3-bit RGB mask:

Table 24: Borders Colors

| Colour | Red | Green | Blue | Value |
| :---: | :---: | :---: | :---: | :---: |
| Black | 0 | 0 | 0 | $\mathbf{0}$ |
| Blue | 0 | 0 | 1 | $\mathbf{1}$ |
| Green | 0 | 1 | 0 | $\mathbf{2}$ |
| Cyan | 0 | 1 | 1 | $\mathbf{3}$ |
| Red | 1 | 0 | 0 | $\mathbf{4}$ |
| Magenta | 1 | 0 | 1 | $\mathbf{5}$ |
| Yellow | 1 | 1 | 0 | $\mathbf{6}$ |

Table 24: Borders Colors

| Colour | Red | Green | Blue | Value |
| :---: | :---: | :---: | :---: | :---: |
| White | 1 | 1 | 1 | $\mathbf{7}$ |

The borders override the graphic pattern. In addition, the left and right edges override the top and bottom ones: for example, when both left and top side borders are enabled, the upper left corner has the color of the left side border.

## Example:

PGEN_ENAB $=01$ enables PGEN
PGEN_X_TOT_L $=00$, PGEN_X_TOT_H $=05$ considering that the panel is 1280 pixels wide PGEN_Y_TOT_L $=00$, PGEN_Y_TOT_H $=04$ considering that the panel is 1024 pixels high PGEN_B_TOP_BOT = EE adds a yellow horizontal borderline to top and bottom of display area PGEN_B_LFT_RHT = 9A adds a blue vertical borderline to the left and a green one to the right

### 4.11.5 TCON Window Control

Normally, the whole PGEN block is enabled if its global enable bit PGEN_ENAB[0] is set.
If it is not set but the bit PGEN_ENAB[1] is set instead, the programmed pattern will show only inside a rectangular window defined by the associated TCON signal TCON_X_PGEN. Outside this window, the input video stream will be displayed as generated by the scaler.
Note: If the global enable bit PGEN_ENAB[0] is set, it has priority over PGEN_ENAB[1].


Table 25: Pattern Generator Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PGEN_ENAB | 0600 | R/W | [1] | 00 | Window control via TCON signal $0^{*}$ : disable, use global enable bit 0 below 1: enable PGEN by TCON_X_PGEN |
|  |  | R/W | [0] |  | Global PGEN enable bit <br> 0 *: disable <br> 1: enable <br> (this bit overrides bit 1 above) |
| PGEN_X_TOT_L | 0601 | R/W | [7:0] | 00 | screen total horizontal size in pixels |
| PGEN_X_TOT_U | 0602 | R/W | [3:0] | 00 |  |

Table 25: Pattern Generator Registers (Sheet 2 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PGEN_Y_TOT_L | 0603 | R/W | [7:0] | 00 | screen total vertical size in lines |
| PGEN_Y_TOT_U | 0604 | R/W | [3:0] | 00 |  |
| PGEN_B_TOP_BOT | 0605 | R/W | [7] | 00 | top border enable bit |
|  |  | R/W | [6:4] |  | top border R G B color enable bits |
|  |  | R/W | [3] |  | bottom border enable bit |
|  |  | R/W | [2:0] |  | bottom border R G B color enable bits |
| PGEN_B_LFT_RHT | 0606 | R/W | [7] | 00 | left border enable bit |
|  |  | R/W | [6:4] |  | left border R G B color enable bits |
|  |  | R/W | [3] |  | right border enable bit |
|  |  | R/W | [2:0] |  | right border R G B color enable bits |
| PGEN_GRIDO | 0607 | R/W | [7:0] | 00 | grid 's row 0 <br> 0 : select P0 (bar pattern) <br> 1: select input signal (from scaler) |
| PGEN_GRID1 | 0608 | R/W | [7:0] | 00 | grid 's row 1 |
| PGEN_GRID2 | 0609 | R/W | [7:0] | 00 | grid 's row 2 |
| PGEN_GRID3 | 060A | R/W | [7:0] | 00 | grid 's row 3 |
| PGEN_GRID4 | 060B | R/W | [7:0] | 00 | grid 's row 4 |
| PGEN_GRID5 | 060C | R/W | [7:0] | 00 | grid 's row 5 |
| PGEN_GRID6 | 060D | R/W | [7:0] | 00 | grid 's row 6 |
| PGEN_GRID7 | 060E | R/W | [7:0] | 00 | grid 's row 7 |
| PGEN_GRID_X_L <br> PGEN_GRID_X_U | $\begin{aligned} & 060 F \\ & 0610 \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | grid cells width, in pixels |
| PGEN_GRID_Y_L <br> PGEN_GRID_Y_U | $\begin{aligned} & 0611 \\ & 0612 \end{aligned}$ | $\begin{aligned} & R / W \\ & R / W \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | grid cells height, in lines |
| PGEN_GRID_X_OFFSET_L <br> PGEN_GRID_X_OFFSET_U | $\begin{aligned} & 0613 \\ & 0614 \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | grid's horizontal offset, in pixels |
| PGEN_GRID_Y_OFFSET_L <br> PGEN_GRID_Y_OFFSET_U | $\begin{aligned} & 0615 \\ & 0616 \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | grid's vertical offset, in lines |
| PGEN_PO_MODE | 0617 | R/W | [7:5] | 00 | number of bars in CO (actual number -1) |
|  |  | R/W | [4:2] |  | number of bars in C 1 (actual number -1) |
|  |  | R/W | [1] |  | cell reset enable |
|  |  | R/W | [0] |  | video replaces C0 enable |
| PGEN_PO_COLOR_B_C0 | 0618 | R/W | [7:0] | 00 | color C0 - blue |
| PGEN_PO_COLOR_G_C0 | 0619 | R/W | [7:0] | 00 | color C0 - green |
| PGEN_PO_COLOR_R_CO | 061A | R/W | [7:0] | 00 | color C0 - red |
| PGEN_P0_COLOR_B_C1 | 061B | R/W | [7:0] | 00 | color C1 - blue |
| PGEN_PO_COLOR_G_C1 | 061C | R/W | [7:0] | 00 | color C1 - green |

Table 25: Pattern Generator Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PGEN_PO_COLOR_R_C1 | 061D | R/W | [7:0] | 00 | color C1 - red |
| PGEN_P0_SEQ_COLO_COL1 | 061E | R/W | [6:4] | 00 | bar 0: R G B color mask |
|  |  | R/W | [2:0] |  | bar 1: R G B color mask |
| PGEN_PO_SEQ_COL2_COL3 | 061F | R/W | [6:4] | 00 | bar 2: R G B color mask |
|  |  | R/W | [2:0] |  | bar 3: R G B color mask |
| PGEN_PO_SEQ_COL4_COL5 | 0620 | R/W | [6:4] | 00 | bar 4: R G B color mask |
|  |  | R/W | [2:0] |  | bar 5: R G B color mask |
| PGEN_P0_SEQ_COL6_COL7 | 0621 | R/W | [6:4] | 00 | bar 6: R G B color mask |
|  |  | R/W | [2:0] |  | bar 7: R G B color mask |
| PGEN_PO_WDTH_L PGEN_PO_WDTH_U | $\begin{aligned} & 0622 \\ & 0623 \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \hline 00 \\ & 00 \end{aligned}$ | bar width, in pixels |
| PGEN_PO_HGHT_L PGEN_PO_HGHT_U | $\begin{aligned} & 0624 \\ & 0625 \end{aligned}$ | $\begin{aligned} & \hline \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | bar height, in lines |
| PGEN_PO_WDTH_X_OFFSET_L PGEN_PO_WDTH_X_OFFSET_U | $\begin{aligned} & 0626 \\ & 0627 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | bar horizontal offset, in pixels |
| PGEN_PO_HGHT_Y_OFFSET_L PGEN_PO_HGHT_Y_OFFSET_U | $\begin{aligned} & 0628 \\ & 0629 \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | bar vertical offset, in lines |
| PGEN_PO_GRADDELTA_B | 062A | R/W | [7:0] | 00 | blue gradient delta |
| PGEN_PO_GRADDELTA_G | 062B | R/W | [7:0] | 00 | green gradient delta |
| PGEN_PO_GRADDELTA_R | 062C | R/W | [7:0] | 00 | red gradient delta |
| PGEN_PO_GRADSTEP_X | 062D | R/W | [7:0] | 00 | gradient horizontal step, in pixels |
| PGEN_PO_GRADSTEP_Y | 062E | R/W | [7:0] | 00 | gradient vertical step, in lines |

## EXAMPLES

All examples assume that the display panel size is $1280 \times 1024$ and no pattern is preset, therefore:

- PGEN_X_TOT_L = 00, PGEN_X_TOT_H = 05
- PGEN_Y_TOT_L = 00, PGEN_Y_TOT_H = 04
- All other registers are 00
- A stable picture is being displayed


## Example 1

PGEN_GRID0 $=$ PGEN_GRID7 $=00$ generated pattern is enabled on all 8 cells of grid 0 (top) and grid 7 (bottom)
PGEN_GRID1.. $6=7 \mathrm{E}$ generated pattern is enabled on 1 st and 8 th cells only of grid 1 thru 6
PGEN_GRID_X_L = $1280 / 8$ cells per grid across screen = A0, PGEN_GRID_X_H = 00
PGEN_GRID_Y_L = $1024 / 8$ lines across screen $=80$, PGEN_GRID_Y_H $=00$
PGEN_PO_MODE = 00 color C0 uses 1 bar (bar 0) only

PGEN_P0_SEQ_COLO_COL1 = 70 bar 0 does not block any of the R G B colors
PGEN_PO_COLOR_B_C0 $=00$
PGEN_PO_COLOR_G_C0 = FF define color CO as light green
PGEN_PO_COLOR_B_C0 $=00$
PGEN_ENABLE = 01 enable PGEN
This displays a thick green block that surrounds the original picture in the center.
Now, if PGEN_P0_GRADDELTA_G = FF and PGEN_P0_GRADSTEP_X = 05, the solid green is turned into one linear horizontal shade of green, evenly spread over the horizontal axis from left (light green) to right (black).
Additionally, if PGEN_PO_MODE $=02$, the gradient registers are preset to color C0 each time a new grid cell is displayed; this gives 8 distinct shades of green ( 1 per cell) across the display.

## Example 2

PGEN_GRIDO, 2, 4, $6=00$ all cells of these grids display the pattern
PGEN_GRID1, 3, 5, 7= 80 1st cell of these grids displays the real picture
PGEN_GRID_X_L = 00, PGEN_GRID_X_H = 05 the 1st cell takes the entire display width
PGEN_GRID_Y_L $=1024 / 8$ lines across screen $=80$, PGEN_GRID_Y_H $=00$ one cell takes $1 / 8$ th of the display height, so that all 8 grids will be displayed
PGEN_PO_MODE = 80 color C0 uses 4 bars (bars 0123 )
PGEN_PO_SEQ_COLO_COL1 = 74

- bar 0 does not block any of the R G B colors (displays CO as is)
- bar 1 blocks $G$ and $B$ colors (displays $R$ only)

PGEN_PO_SEQ_COL2_COL3 = 21

- bar 2 blocks $R$ and $B$ colors (displays $G$ only)
- bar 3 blocks $R$ and $G$ colors (displays B only)

PGEN_P0_COLOR_B_C0 = PGEN_P0_COLOR_G_C0 = PGEN_P0_COLOR_B_C0 = 00: C0 is black
PGEN_PO_GRADDELTA_R = PGEN_PO_GRADDELTA_G $=$ PGEN_PO_GRADDELTA_B $=01:$ R G $B$ color components of C0 gradually increase from left to right
PGEN_PO_GRADSTEP_X = 05: shade is evenly spread over the horizontal axis
PGEN_ENABLE = 01 enable PGEN
This displays a complex pattern made of 8 horizontal rows:

- 1 st row (= bar 0) displays a shade of white, from left (black) to right (white)
- 3rd row (= bar 1) displays a shade of red, from left (black) to right (light red)
- 5th row (= bar 2) displays a shade of green, from left (black) to right (light green)
- 7th row (= bar 3) displays a shade of blue, from left (black) to right (light blue)
- 2nd, 4th, 6th and 8th rows display the original picture

When displaying the same pattern from an external pattern generator, since each row of each color is displayed side by side with the same reference shade row generated by the PGEN, defects can be spotted immediately. This is a very useful test to see possible ADC or panel defects.

### 4.12 sRGB (SRGB)

The sRGB block performs two primary functions:

1. Parametric gamma correction on multiple windows or full screen, used for video enhancement in a window and digital contrast/brightness control. The window coordinates are set by TCON registers.
2. 3D color cube warping RGB color space.

The entire backend of the ADE3800 (from Scaler output to the APC) has a 10 bit database including the sRGB block. The sRGB controls correspond to the 8 MSBs of the data.

### 4.12.1 Parametric Gamma, Digital Contrast / Brightness on Multiple Windows

The function can be applied to the entire window by programming the window control to full screen. Each color channel acts independently. Simple digital contrast and brightness can be programmed using this hardware function. The desired window coordinates are programmed into the TCON.

Note: If both Gamma1 and Gamma2 are enabled, Gamma1 has priority over Gamma2.

### 4.12.2 Color Space Warp

The 8 corners of the color cube are independently controlled in 3D space with smooth interpolation of intermediate colors. Registers are 2's complement color deltas.
For example:

- to make WHITE more like RED, program SRGB_WHITE_R to a small positive value.
- to turn RED into GREEN, set Gain $=2$ in SRGB_CTRLO[7:6], then SRGB_RED_R $=0 \times 80$ (-128) to block the red, and SRGB_RED_G=0x30 (the higher the value (up to 0x7F) the brighter the green).

Figure 11: Color Space Warp


The step value for each color delta depends on the gain setting in SRGB_CTRL0[7:6], as follows:
Table 26: Color Space Warp Gain Control

| SRGB_CTRLO[7:6] | Gain | Step Size | Color Delta Range |
| :---: | :---: | :---: | :---: |
| 0 | 1 | 0.5 | $[-64 ;+63]$ |
| 1 | 2 | 1 | $[-128 ;+127]$ |
| 2 | 4 | 2 | $[-256 ;+255]$ |

Note: It is recommended to limit the range of all red/green/blue correction registers and black/red/green/ blue/yellow/cyan/magenta/white delta registers to $[-64 . .+63]$ to avoid color overflow/underflow computation.

Table 27: sRGB Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SRGB_CTRLO | OD00 | R/W | [7:6] | 00 | Gain control of $s R G B$ coeff values <br> $0^{*}$ : gain $=1$ (half step) <br> 1: gain $=2$ (single step) <br> 2: gain $=4$ (double step) |
|  |  | R/W | [5:4] |  | 00*: gamma2 disabled <br> 01: gamma2 full screen <br> 10: gamma2 windowed <br> 11: reserved |
|  |  | R/W | [3:2] |  | 00*: gamma1 disabled <br> 01: gamma1 full screen <br> 10: gamma1 windowed <br> 11: reserved |
|  |  | R/W | [1:0] |  | 00*: srgb disabled <br> 01: srgb full screen <br> 10: srgb windowed <br> 11: reserved |
| SRGB_CTRL1 | 0D01 | R/W | [4] | 00 | 0*: dither pattern disabled (normal) |
| SRGB_CTRL2 | 0D02 | R/W | [3] | 00 | White point saturation inside gamma2 window $0^{*}$ : disabled <br> 1: enabled |
|  |  | R/W | [2] |  | White point saturation inside gamma1 window $0^{*}$ : disabled <br> 1: enabled |
|  |  | R/W | [1] |  | White point saturation inside srgb window $0^{*}$ : disabled <br> 1: enabled |
|  |  | R/W | [0] |  | White point saturation over full screen <br> $0^{*}$ : disabled <br> 1: enabled |
| SRGB_BLACK_B | 0D03 | R/W | [7:0] | 00 | black point bluedelta |
| SRGB_BLACK_G | 0D04 | R/W | [7:0] | 00 | black point green delta |
| SRGB_BLACK_R | 0D05 | R/W | [7:0] | 00 | black point red delta |
| SRGB_RED_B | 0D06 | R/W | [7:0] | 00 | red point blue delta |
| SRGB_RED_G | 0D07 | R/W | [7:0] | 00 | red point green delta |
| SRGB_RED_R | 0D08 | R/W | [7:0] | 00 | red point red delta |
| SRGB_GREEN_B | 0D09 | R/W | [7:0] | 00 | green point bluedeltablue |
| SRGB_GREEN_G | ODOA | R/W | [7:0] | 00 | green point green delta |
| SRGB_GREEN_R | ODOB | R/W | [7:0] | 00 | green point reddelta |
| SRGB_BLUE_B | ODOC | R/W | [7:0] | 00 | blue point bluedelta |
| SRGB_BLUE_G | ODOD | R/W | [7:0] | 00 | blue point green delta |
| SRGB_BLUE_R | ODOE | R/W | [7:0] | 00 | blue point red delta |

Table 27: sRGB Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| SRGB_YELLOW_B | ODOF | R/W | [7:0] | 00 | yellow point bluedelta |
| SRGB_YELLOW_G | 0D10 | R/W | [7:0] | 00 | yellow point green delta |
| SRGB_YELLOW_R | 0D11 | R/W | [7:0] | 00 | yellow point red delta |
| SRGB_CYAN_B | 0D12 | R/W | [7:0] | 00 | cyan point bluedelta |
| SRGB_CYAN_G | 0D13 | R/W | [7:0] | 00 | cyan point green delta |
| SRGB_CYAN_R | 0D14 | R/W | [7:0] | 00 | cyan point red delta |
| SRGB_MAGENTA_B | 0D15 | R/W | [7:0] | 00 | magenta point bluedelta |
| SRGB_MAGENTA_G | 0D16 | R/W | [7:0] | 00 | magenta point green delta |
| SRGB_MAGENTA_R | 0D17 | R/W | [7:0] | 00 | magenta point red delta |
| SRGB_WHITE_B | 0D18 | R/W | [7:0] | 00 | white point bluedelta |
| SRGB_WHITE_G | 0D19 | R/W | [7:0] | 00 | white point green delta |
| SRGB_WHITE_R | 0D1A | R/W | [7:0] | 00 | white point red delta |
| SRGB_WSAT_LIM_B | 0D1B | R/W | [7:0] | FF | White point saturation value for the bluecomponent |
| SRGB_WSAT_LIM_G | 0D1C | R/W | [7:0] | FF | White point saturation value for the green component |
| SRGB_WSAT_LIM_R | 0D1D | R/W | [7:0] | FF | White point saturation value for the red component |
| SRGB_GAMMA1_CON_B | 0D1E | R/W | [7:0] | 00 | first parametric contrast correction, bluecomponent |
| SRGB_GAMMA1_CON_G | 0D1F | R/W | [7:0] | 00 | first parametric contrast correction, green component |
| SRGB_GAMMA1_CON_R | 0D20 | R/W | [7:0] | 00 | first parametric contrast correction, red component |
| SRGB_GAMMA1_BRI_B | 0D21 | R/W | [7:0] | 00 | first parametric brightness correction, bluecomponent |
| SRGB_GAMMA1_BRI_G | 0D22 | R/W | [7:0] | 00 | first parametric brightness correction, green component |
| SRGB_GAMMA1_BRI_R | 0D23 | R/W | [7:0] | 00 | first parametric brightness correction, red component |
| SRGB_GAMMA1_GAM_B | 0D24 | R/W | [7:0] | 00 | first parametric gamma correction, bluecomponent |
| SRGB_GAMMA1_GAM_G | 0D25 | R/W | [7:0] | 00 | first parametric gamma correction, green component |
| SRGB_GAMMA1_GAM_R | 0D26 | R/W | [7:0] | 00 | first parametric gamma correction, red component |
| SRGB_GAMMA2_CON_B | 0D27 | R/W | [7:0] | 00 | second parametric contrast correction, bluecomponent |
| SRGB_GAMMA2_CON_G | 0D28 | R/W | [7:0] | 00 | second parametric contrast correction, green component |
| SRGB_GAMMA2_CON_R | 0D29 | R/W | [7:0] | 00 | second parametric contrast correction, red component |
| SRGB_GAMMA2_BRI_B | 0D2A | R/W | [7:0] | 00 | second parametric brightness correction, bluecomponent |
| SRGB_GAMMA2_BRI_G | 0D2B | R/W | [7:0] | 00 | second parametric brightness correction, green component |
| SRGB_GAMMA2_BRI_R | 0D2C | R/W | [7:0] | 00 | second parametric brightness correction, red component |
| SRGB_GAMMA2_GAM_B | 0D2D | R/W | [7:0] | 00 | second parametric gamma correction, bluecomponent |
| SRGB_GAMMA2_GAM_G | OD2E | R/W | [7:0] | 00 | second parametric gamma correction, green component |
| SRGB_GAMMA2_GAM_R | 0D2F | R/W | [7:0] | 00 | second parametric gamma correction, red component |

### 4.13 Gamma (GAM)

The Gamma block implements three independent 256 point gamma curves for each of $R, G$, and $B$ channels.
Its features are as follows:

- $256 \times 8 \mathrm{~b}$ table per color channel stores 2's complement difference to straight line
- 10b input/output ( 0 to 1023 ), range of delta $=-128$ to $+127\left(+/-1 / 8^{\text {th }}\right.$ full scale)
- double LUT amplitude control to change range to $2 x$ (delta $=-256$ to +254 )
- fast write mode for loading 3 tables with the same data
- glitch free write mode
- clipping on output to $[0,1023]$
- programmable offset_value added from offset_position0 to offset_position1 (inclusive).
gamma_out_r = gamma_in_r + lut_r + (offset_position0 <= gamma_in_r <= offset_position1) ? offset_value: 0
gamma_out_g = gamma_in_g + lut_g + (offset_position0 <= gamma_in_g <= offset_position1) ? offset_value: 0
gamma_out_b = gamma_in_b + lut_b + (offset_position0 <= gamma_in_b <= offset_position1) ? offset_value: 0

Table 28: Gamma Registers

| I2C Address Label | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| GAM_CTRL | 0C00 | R/W | [3] | 00 | $\begin{aligned} & 0^{*}: \text { delta range }=-128 \text { to }+127 \\ & \text { 1: delta range }=-256 \text { to }+254 \end{aligned}$ |
|  |  |  | [2] |  | $0^{*}$ : i2c to RAM transfer at selected i2c address only 1: i2c to RAM transfer the same value to Red, Green, and Blue RAMs when selecting Red RAM addresses |
|  |  |  | [1] |  | $0^{*}$ : Write i2c to RAM allowed during active video <br> 1: Write i2c to RAM during video blanking only (shadowed) |
|  |  |  | [0] |  | 0*: gamma bypassed <br> 1: gamma enabled |
| GAM_POSITIONO | 0C01 | R/W | [7:0] | 00 | ```IF (gamma_in/4 >= offset_position0 && gamma_in/4 <= offset_position1) THEN offset = offset_value * 16 ELSE offset = 0 (gamma_out = gamma_in + lut + offset)``` |
| GAM_POSITION1 | 0 CO 2 | R/W | [7:0] | 00 | See offset_position0 for details |
| GAM_OFFSET | 0C03 | R/W | [5:0] | 00 | Multipled by 16. 2's complement number represents -512 to +496 inclusive. <br> See offset_position0 for details |

Table 29: Gamma LUT RAM addresses

| I2C Address | Memory Contents |
| :---: | :---: |
| $1000-10 F F$ | Red RAM |
| $1100-11 \mathrm{FF}$ | Green RAM |
| $1200-12 \mathrm{FF}$ | Blue RAM |

Note: RAM ACCESS REQUIRES DOTCLK >= XCLK (refer to Chapter 4.22: $I^{2} C$ Registers and RAM Addresses)

### 4.14 On-Screen Display (OSD)

The On-Screen Display block has the following features:

- Registers 4900-4915 are shadowed and are updated on the falling edge of out_venab.
- Pointers for the global RAM refer to 24 bit word locations.
- Pointers for the color LUT RAM refer to 32 bit word locations.
- Write access to the RAMs is shadowed.
- Read access to the global RAM is shadowed.
- Display list must be in top to bottom order for consistent operation. One RAM block 4096x24 is used for the full operation of the OSD, and is internally subdivided for character use or display list with the ability to set up the pointers through I2C.
- The characters can be displayed anywhere on the screen.
- H/V position is programmable per row
- Global Alpha blending for all the characters displayed as well as Alpha blending per color with 16 levels
- H/V flip per character
- $1 \mathrm{bpp} / 2 \mathrm{bpp} / 3 \mathrm{bpp} / 4 \mathrm{bpp}$ characters supported.
- Rotation support
- Color LUT of 64 colors (24bit RGB True Color + 4 bit alpha).

Table 30: OSD Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| OSD_RAM | $1700-$ <br> $46 F F$ | R/W |  |  | I2C address space allocated for OSD Ram |
| OSD_CLUT | $4700-$ <br> $47 F F$ | R/W |  |  | I2C address space allocated for OSD CLUT |

Table 30: OSD Registers (Sheet 2 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| OSD_CTRLO | 4900 | R/W | [7:4] | 00 | Global Alpha $\text { LSB }=1 / 15$ |
|  |  | R/W | [3] |  | Rotation <br> $0 *: 12 \mathrm{H} \times 18 \mathrm{~V}$ char <br> 1: $18 \mathrm{H} \times 12 \mathrm{~V}$ char |
|  |  | R/W | [2] |  | TCON Highlight Window Palette Index |
|  |  | R/W | [1] |  | TCON Highlight Window Enable |
|  |  | R/W | [0] |  | OSD enable |
| OSD_CTRL1 | 4901 | R/W | [7] | 00 | OSD List Pointer Select |
|  |  | R/W | [6:0] |  | Total OSD Rows |
| OSD_GLBL_X_OFFSET_L OSD_GLBL_X_OFFSET_U | $\begin{aligned} & 4902 \\ & 4903 \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Global OSD Xpos offset in pixels |
| OSD_GLBL_Y_OFFSET_L OSD_GLBL_Y_OFFSET_U | $\begin{aligned} & 4904 \\ & 4905 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Global OSD Ypos offset in pixels |
| $\begin{aligned} & \text { OSD_CP_1BPP_L } \\ & \text { OSD_CP_1BPP_U } \end{aligned}$ | $\begin{aligned} & 4906 \\ & 4907 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | 1bpp Char Pointer |
| OSD_CP_2BPP_L OSD_CP_2BPP_U | $\begin{aligned} & 4908 \\ & 4909 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | 2bpp Char Pointer |
| OSD_CP_3BPP_L OSD_CP_3BPP_U | $\begin{aligned} & \text { 490A } \\ & \text { 490B } \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | 3bpp Char Pointer |
| $\begin{aligned} & \text { OSD_CP_4BPP_L } \\ & \text { OSD_CP_4BPP_U } \end{aligned}$ | $\begin{aligned} & \text { 490C } \\ & \text { 490D } \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | 4bpp Char Pointer |
| $\begin{aligned} & \text { OSD_DLPO_L } \\ & \text { OSD_DLPO_U } \end{aligned}$ | $\begin{aligned} & 490 \mathrm{E} \\ & 490 \mathrm{~F} \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & \hline[7: 0] \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Display List Pointer0 |
| $\begin{aligned} & \text { OSD_DLP1_L } \\ & \text { OSD_DL1_U } \end{aligned}$ | $\begin{aligned} & 4910 \\ & 4911 \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Display List Pointer1 |
| OSD_CLUT_1BPP | 4912 | R/W | [4:0] | 00 | Base Color LUT for 1bpp |
| OSD_CLUT_2BPP | 4913 | R/W | [4:0] | 00 | Base Color LUT for 2bpp |
| OSD_CLUT_3BPP | 4914 | R/W | [4:0] | 00 | Base Color LUT for 3bpp |
| OSD_CLUT_4BPP | 4915 | R/W | [4:0] | 00 | Base Color LUT for 4bpp |

Table 30: OSD Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| OSD_OSD_CTRLO_HW | 4920 | R | [7:0] | 00 | HW Shadow Readback |
| OSD_OSD_CTRL1_HW | 4921 | R | [7:0] | 00 |  |
| OSD_GLBL_X_OFFSET_HW_L | 4922 | R | [7:0] | 00 |  |
| OSD_GLBL_X_OFFSET_HW_U | 4923 | R | [3:0] | 00 |  |
| OSD_GLBL_Y_OFFSET_HW_L | 4924 | R | [7:0] | 00 |  |
| OSD_GLBL_Y_OFFSET_HW_U | 4925 | R | [3:0] | 00 |  |
| OSD_CP_1BPP_HW_L | 4926 | R | [7:0] | 00 |  |
| OSD_CP_1BPP_HW_U | 4927 | R | [3:0] | 00 |  |
| OSD_CP_2BPP_HW_L | 4928 | R | [7:0] | 00 |  |
| OSD_CP_2BPP_HW_U | 4929 | R | [3:0] | 00 |  |
| OSD_CP_3BPP_HW_L | 492A | R | [7:0] | 00 |  |
| OSD_CP_3BPP_HW_U | 492B | R | [3:0] | 00 |  |
| OSD_CP_4BPP_HW_L | 492C | R | [7:0] | 00 |  |
| OSD_CP_4BPP_HW_U | 492D | R | [3:0] | 00 | HW Shadow Readback |
| OSD_DLPO_HW_L | 492E | R | [7:0] | 00 |  |
| OSD_DLPO_HW_U | 492F | R | [3:0] | 00 |  |
| OSD_DLP1_HW_L | 4930 | R | [7:0] | 00 |  |
| OSD_DLP1_HW_U | 4931 | R | [3:0] | 00 |  |
| OSD_CLUT_1BPP_HW | 4932 | R | [4:0] | 00 |  |
| OSD_CLUT_2BPP_HW | 4933 | R | [4:0] | 00 |  |
| OSD_CLUT_3BPP_HW | 4934 | R | [4:0] | 00 |  |
| OSD_CLUT_4BPP_HW | 4935 | R | [4:0] | 00 |  |

Figure 12: OSD RAM


### 4.14.1 Implementation

| Row Type $\mathbf{0}$ Attributes: (total 48 bits) |  |  |
| :--- | :--- | :--- |
| [Y Position] | 12 bits | (HPOS) |
| [X Position] | 12 bits | (YPOS) |
| [Type of Row] | 2 bits | (TR) |
| [Char/Row] | 7 bits | (CR) |
| [Palette] | 1 bits | (PI) |
| [FlipHV] | 2 bits | (HVF) |
| [CharDepth0] | 2 bits | (CD0) |
| [CharDepth1] | 2 bits | (CD1) |
| [BG] | 4 bits | (BG) |
| $[F G]$ | 4 bits | (FG) |


| Row Type $\mathbf{0}$ - Character Attributes: (total 8 bits) |  |  |
| :--- | :--- | :--- |
| [CharID] | 8 bits | (CID) |


| Row Type 1 Attributes: (total 48 bits) |  |  |
| :--- | :--- | :--- |
| [Y Position] | 12 bits | (HPOS) |
| [X Position] | 12 bits | (YPOS) |
| [Type of Row] | 2 bits | (TR) |
| [Char/Row] | 7 bits | (CR) |
| [Palette] | 1 bits | (PI) |
| [FlipHV] | 2 bits | (HVF) |
| [CharDepth0] | 2 bits | (CD0) |
| [CharDepth1] | 2 bits | (CD1) |
| [BG] | 4 bits | (BG) |
| $[F G]$ | 4 bits | (FG) |


| Row Type 1-Character Attributes: (total 12 bits) |  |  |
| :--- | :--- | :--- |
| [CharID] | 8 bits | (CID) |
| [FlipHV] | 2 bits | (HVF) |
| [CharDepthIndex] | 1 bits | (CD) |
| [PaletteIndex] | 1 bits | (PI) |

Note: The Character Attribute [CharDepthIndex] selects which of the 2 char depths will be used from RowAttribute [CharDepth0] or RowAttribute [CharDepth1].

Note: Only two types of char depths can be used, and they are specified in RowAttribute [CharDepth0].
Row Type 2 Attributes: (total 48 bits)

| [Y Position] | 12 bits | (HPOS) |
| :--- | :--- | :--- |
| [X Position] | 12 bits | (YPOS) |
| [Type of Row] | 2 bits | (TR) |
| [Char/Row] | 7 bits | (CR) |
| [Palette] | 1 bits | (PI) |
| [FlipHV] | 2 bits | (HVF) |
| [CharDepth0] | 2 bits | (CD0) |
| NCharDepth1] | 2 bits | (CD1) |
| NOT USED |  |  |
| $[$ NG $]$ | 4 bits | (BG) |
|  | 4 bits | (FG) |


| Row Type 2 - Character Attributes: (total 16 bits) |  |  |
| :--- | :--- | :--- |
| [CharID] | 8 bits | (CID) |
| [FlipHV] | 2 bits | (HVF) |
| [CharDepth] | 2 bits | (CD) |
| $[P a l e t t e I n d e x] ~$ | 4 bits | (PI) |

Figure 13: Display List Memory Structure (all the bits are packed)


Note: All Row Attributes are assigned as shown:

| YPOS[23:12] |  |  |  | XPOS[11:0] |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| FG[23:20] | BG[19:16] | CD1[15:14] | CD0[13:12] | HVF[11:10] | PI[9] | CR[8:2] | TR[1:0] |

Note: Character Attributes for Row Type 0 are assigned as shown:

| $\operatorname{CID}[7: 0]$ |
| :--- |

Note: Character Attributes for Row Type 1 are assigned as shown:

| PI[11] | CD[10] | HF[9] | VF[8] | CID[7:0] |
| :--- | :--- | :--- | :--- | :--- |

Note: Character Attributes for Row Type 2 are assigned as shown:

| PI[15:12] | CD[11:10] | HF[9] | VF[8] | CID[7:0] |
| :--- | :--- | :--- | :--- | :--- |

### 4.14.2 Color LUT Calculation

Color pointers in the CLUT [5:0], where:
$\mathrm{PI}=4$-bit Palette Index (RT0/RT1 have 1--bit PI; RT2 has 4bit PI) ;
P1, P2, P3, P4 = 5-bit programmable pointers, clut_1bpp, clut_2bpp, clut_3bpp, clut_4bpp, respectively;
PixelData $=2$-bit, 3-bit or 4-bit value depending on the character depth 2 bpp , 3 bpp or 4 bpp , respectively;
C = 4-bit background/foreground color (used only for 1bpp characters);
tcon = OSD_CTRLO[1] * tcon_window where tcon_window is a signal from the TCON block specifying the window to be highlighted.

|  | RowType0 | RowType1 | RowType2 |
| :---: | :---: | :---: | :---: |
| 1bpp | (tcon*OSD_CTRLO[2]*32 + !tcon*32*PI + P1 + BC) \% 64 | (tcon*OSD_CTRLO[2]*32 + !tcon*32*PI + P1 + BC) \% 64 | (tcon*OSD_CTRLO[2]* ${ }^{*}$ ( $\left.+\mathrm{PI} \% 8\right)^{*} 4+$ <br> !tcon*4*PI + P1 + BC) \% 64 |
| 2bpp | (tcon*OSD_CTRLO[2]*32 + !tcon*32*P1 + P2 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]*32 + Itcon*32*PI + P2 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]* $(8+\mathrm{Pl} \% 8)^{*} 4+$ !tcon*4*PI + P2 + PixelData) \% 64 |
| 3bpp | (tcon*OSD_CTRLO[2]*32 + !tcon*32*PI + P3 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]*32 + <br> !tcon*32*PI + P3 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]* $(8+\mathrm{Pl} \% 8)^{*} 4+$ !tcon*4*PI + P3 + PixelData) \% 64 |
| 4bpp | (tcon*OSD_CTRLO[2]*32 + !tcon*32*PI + P4 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]*32 + Itcon*32*PI + P4 + PixelData) \% 64 | (tcon*OSD_CTRLO[2]* $(8+\mathrm{PI} \% 8)^{*} 4+$ !tcon*4*PI + P4 + PixelData) \% 64 |

### 4.14.3 Alpha Blending

4-bit Alpha is applied to the OSD providing 16 levels ( $6.25 \%$ steps) of blending.
Alpha $=0 F$ :no blending at all ( $100 \%$ OSD data).
Alpha $=00:$ full blending ( $100 \%$ input video).
The OSD region has a 4-bit global alpha and each RGB has a 4-bit color alpha.

Total alpha is calculated as follows:
r_o = (r_i[9:0] + ( (\{lut_data[23:16],lut_data[23:22]\} - r_i[9:0]) * total_alpha) )
g_o = (g_i[9:0] + ( (\{lut_data[15:8] ,lut_data[15:14]\} - g_i[9:0]) * total_alpha) )
b_o = (b_i[9:0] + ( (\{lut_data[7:0] ,lut_data[7:6]\} -b_i[9:0]) * total_alpha) )
total_alpha_selector[7:0] = (glbl_alpha[3:0] * color_alpha[3:0])
The total alpha read from a LUT of 32 entries that are normalized, where the range is total_alpha = $0,1,2,3,4 \ldots 16$; and only the 5 msb 's of total_alpha_selector[7:3] are used as select.
i.e. 16 represents 1.0 "no alpha blending at all". Figure 14 shows how the Alpha Blending is constructed:

Figure 14: OSD Alpha Blending



Figure 15: Global Memory Assignment


### 4.14.4 RAM Memory

## Character Memory:

$175 \times 12 \times 18 \times 1 \mathrm{bpp}=37800$ bits
$42 \times 12 \times 18 \times 4$ bpp $=36288$ bits
The total character storage RAM is estimated based on supporting $175 \times 1 \mathrm{bpp}$ and $42 \times 4 \mathrm{bpp}$ characters.
Total RAM allocated for Character storage $=>\mathbf{7 4 0 8 8}$ bits

## Display List:

Row Attr. 48 bits $\times 15$ rows $=720$ bits
Char Attr. 16 bits $\times 30$ chars $\times 15$ rows $=7200$ bits
The total display list is estimated based on the current OSD size of $30 \times 15$ characters.
Total Display List Memory => 7920 bits
TOTAL OSD Estimated RAM Memory: $\mathbf{8 2 0 0 8}$ bits (RAM selected 98304 bits)

## Color LUT:

$64 \times 32=2048$ bits
For a $30 \times 15$ character display the OSD block global RAM has room remaining for:

$$
\begin{aligned}
& 255=>1 \text { bpp (room for } 418 \text { char, but only } 255 \text { can be addressed w/ 8bit CID) or, } \\
& 209=>2 b p p \text { or, } \\
& 139=>3 b p p \text { or, } \\
& 104=>4 \text { bpp }
\end{aligned}
$$

Global 24b RAM is programmed in the following order:

## Example

WRITE 00 [Data] -> ram_addr 0 [23: 16]
WRITE 01 [Data] -> ram_addr 0 [15: 8]
WRITE 02 [Data] -> ram_addr 0[7: 0]
WRITE 03 [Data] -> ram_addr 1 [23: 16]
WRITE 04 [Data] -> ram_addr 1 [15: 8]
... and so on .....
Similarly, the Color LUT 32b RAM, is programmed in the following order:

## Example

WRITE 00 [Alpha] $\rightarrow$ lut_addr 0 [27:24]
WRITE 01 [ R ] -> lut_addr 0 [23:16]
WRITE 02 [ G ] -> lut_addr 0 [15: 8]
WRITE 03 [ B ] $\rightarrow$ lut_addr 0 [ $7: 0$ ]
WRITE 04 [Alpha] -> lut_addr 1 [27:24]
... and so on .....

Each character is programmed into the RAM starting with the upper left pixel, and it continues going to the right bottom. For example, programming of a 1 bpp character " B " will be as follows:


| i2c comm. | i2c <br> address data |  |  |
| :---: | :---: | :---: | :---: |
| WRITE | 00 |  | -> ram_address 0 [23:16] |
| WRITE | 01 |  | -> ram_address 0 [15 :8] |
| WRITE | 02 |  | -> ram_address $0\left[\begin{array}{ll}7 & \text { :0] }\end{array}\right.$ |
| WRITE | 03 |  | -> ram_address 1 [23:16] |
| WRITE | 04 |  | -> ram_address 1 [15:8] |
| WRITE | 05 | 18 | -> ram_address 1 [7 :0] |
| WRITE | 06 | 60 | -> ram_address 2 [23:16] |
| WRITE | 07 |  | -> ram_address 2 [15:8] |
| WRITE | 08 |  | -> ram_address $2\left[\begin{array}{ll}7 & \text { :0] }\end{array}\right.$ |
| WRITE | 09 |  | -> ram_address 3 [23:16] |
| WRITE | 0a |  | -> ram_address 3 [15:8] |
| WRITE | Ob |  | -> ram_address 3 [ 7 : 0 ] |
| WRITE | Oc |  | -> ram_address 4 [23:16] |
| WRITE | Od |  | -> ram_address 4 [15:8] |
| WRITE | Oe |  | -> ram_address $4\left[\begin{array}{ll}7 & \text { :0] }\end{array}\right.$ |
| WRITE | Of |  | -> ram_address 5 [23:16] |
| WRITE | 10 |  | -> ram_address 5 [15:8] |
| WRITE | 11 |  | -> ram_address 5 [ 7 : 0 ] |
| WRITE | 12 |  | -> ram_address 6 [23:16] |
| WRITE | 13 |  | -> ram_address 6 [15:8] |
| WRITE | 14 |  | -> ram_address $6\left[\begin{array}{ll}7 & : 0\end{array}\right]$ |
| WRITE | 15 |  | -> ram_address 7 [23:16] |
| WRITE | 16 |  | -> ram_address 7 [15:8] |
| WRITE | 17 |  | -> ram_address $7\left[\begin{array}{ll}7 & : 0\end{array}\right]$ |
| WRITE | 18 |  | -> ram_address 8 [23:16] |
| WRITE | 19 |  | -> ram_address 8 [15:8] |
| WRITE | 1a |  | -> ram_address 8 [7 :0] |

Character Data RAM packing is done as follows:

1bpp NON ROTATED

| $16 \quad 15$ | 8 |
| :---: | :--- |
| Line 0 | Line 1 |
| Line 2 | Line 3 |
| Line 4 | Line 5 |
| Line 6 | Line 7 |
| Line 8 | Line 9 |
| Line 10 | Line 11 |
| Line 12 | Line 13 |
| Line 14 | Line 15 |
| Line 16 | Line 17 |

2bpp NON ROTATED


3bpp NON ROTATED


3bpp ROTATED

bits [2:0] are NOT USED

4bpp NON ROTATED


4bpp ROTATED

### 4.15 Flicker (FLK)

The Flicker block computes a nonlinear correlation of LCD polarity inversion patterns and the LCD output data stream and provides the correlation results as scores to the microcontroller via I2C. The MCU polls this block regularly. In response to a high score, the MCU can adjust the polarity signal generated in the TCON to cancel the visual flicker that arises from correlated pixel and polarity patterns.
Figure 16 shows a block diagram of the flicker module and its connectivity with the neighboring modules.

Figure 16: Block Diagram


### 4.15.1 Function

A Walsh $8 \times 8$ function is used to compare the detected pattern, where each one of the 8 functions represents a pattern. All patterns are considered to be vertically, where horizontally the pixels are assumed to be alternating its RGB components.

Only 4 of the patterns can be measured at one time, and they are selected by means of WF_SHIFT[2:0] by programming the number of patterns shifted i.e.

- if WF_SHIFT $=00$ then the 4 results are meas0, meas1, meas2, meas3;
- if WF_SHIFT = 01 then the 4 results are meas1, meas2, meas3, meas4;
- if WF_SHIFT = 05 then the 4 results are meas5, meas6, meas7, meas0; and so on.

The score that is registered at the end of a measurement is the delta intensity between the RGB components on pixels that are alternating horizontally and match one or more of the defined 8 patterns. Since the flickering effect occurs most of the time around the $50 \%$ of the color intensity, two functions are used to get the delta difference between the RGB components, one is normalized at $50 \%$, and the other is normalized at $100 \%$. The selection between the two can be programmed by the FLICKER_CTRLO[5] => 0/1 (100/50\%) normalization.

The horizontal setting of the RGB component of each pixel is represented by the FLICKER_CTRLO[2:0], and for any pattern, maximum scores are calculated by having the correct
distribution of the color components. By default, we assume the most frequent setting is +-+ or -+-, which means FLICKER_CTRLO[2:0] are programmed to either 101 or 010.

A calculation is done after the number of frames programmed in FRAME_CNT_MAX have passed. With each frame the calculation is performed only on a horizontal portion of the image on all lines. The size of that horizontal portion (in pixels) is determined by the value programmed in the HBLOCK_SIZE included in the following formula:

2 ^ (3 + hblock_size)
For calculation of flicker patterns on the whole image, the result of this formula multiplied by FRAME_CNT_MAX should be equal to the line length (in pixels), although that is not a constraint.

By splitting the image calculation to smaller horizontal portions, the local scores are banked (saved) at the end of each portion, hence enabling a reverse pattern within a line to be detected. The smaller that horizontal portion is, the better chance of detecting pattern reversals within a line. Taking that into account, the smaller the horizontal portion is, the more frames needed to finish the full image pattern scan. The minimum horizontal portion can be 8 pixels, and the maximum can be the size of the line. Vertically, the flicker block is defined to have a resolution of 8 lines, so no programming is needed to define the vertical portion, it banks automatically every 8 lines, and it goes through all lines every frame.

The free_run/freeze_scores bit FLICKER_CTRLO[4] enables the final calculation to be fed to the I2C registers. This bit does not regulate all the internal flicker calculation, but only the update of the I2C registers.

The output results are stored in four 32 bit registers with addresses described in the table. The higher the score is, the more that pattern is present in the image (each 32 bit register represents 1 pattern). Whichever pattern is detected most, the TCON is advised to cancel the flicker by switching the pixel polarity which is the opposite of the pattern detected.

The following figure shows all patterns that can be detected by this flicker block.
Figure 17: $8 \times 8$ Walsh basis function set

| + | + | + | + | + | + | + | + |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| + | - | + | - | + | - | + | - |
| + | + | - | - | + | + | - | - |
| + | - | - | + | + | - | - | + |
| + | + | + | + | - | - | - | - |
| + | - | + | - | - | + | - | + |
| + | + | - | - | - | - | + | + |
| + | - | - | + | - | + | + | - |

Figure 18 shows an overview of the scanning of the RGB and updating of the registers diagram:
Figure 18: Scanning Overview


The number of frames used to complete one full measurement and update the I2C registers is programmed into FRAME_CNT_MAX as shown below.


Table 31: FLK Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FLICKER_CTRLO | OCA1 | R/W | [5] | 25 | 0 : straight line uniform function <br> $1^{*}$ : straight line hill function (normal) |
|  |  | R/W | [4] |  | $0^{*}$ : free run <br> 1: freeze scores <br> Set to a 1 when the micro controller is reading multibyte scores to prevent update corruption. |
|  |  | R/W | [2:0] |  | -horizontal polarity pattern (even/odd pixels) $\begin{aligned} & 000:-R-G-B /+R+G+B \\ & 001:-R-G+B /+R+G-B \\ & 010:-R+G-B /+R-G+B \\ & 011:-R+G+B /+R-G-B \\ & 100:+R-G-B /-R+G+B \\ & 101 *:+R-G+B /-R+G-B \\ & 110:+R+G-B /-R-G+B \\ & 111:+R+G+B /-R-G-B \end{aligned}$ <br> -If input data is in RGB format program flicker_ctrl0 to 5 or 2 to get maximum score |

Table 31: FLK Registers (Sheet 2 of 2)

| HBLOCK_SIZE | 0CA2 | R/W | [3:0] | 00 | Size in bits of horizontal window $=2^{\wedge}$ ( $3+$ hblock_size $)$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FRAME_CNT_MAX | 0СA3 | R/W | [7:0] | 08 | -Number of frames to complete one measurement <br> -Total number of pixs in a line would be: <br> frame_cnt_max * ( $2^{\wedge}$ (3+ hblock_size) ) <br> -example: hblock_size $=0$; frame_cnt_max $=8$; <br> means that it will take 8 frames to finish the calculation. For each frame only one portion of the image is being calculated on. The size of that portion is $2 \wedge$ ( $3+$ hblock_size), in this case 8 pixels. This means that the calculated line length $=8$ pix window * 8 frames $=64$ pixels |
| WF_SHIFT | 0CA4 | R/W | [2:0] | 00 | Selector of which 4 of the Walsh function is measuring |
| FLICKER_MEASO | 0CB1 - B4 | R/W | [31:0] | 00 | Score reg showing pattern matching pattern 0 |
| FLICKER_MEAS1 | 0CB5 - B8 | R/W | [31:0] | 00 | Score reg showing pattern matching pattern 1 |
| FLICKER_MEAS2 | 0CB9 - BC | R/W | [31:0] | 00 | Score reg showing pattern matching pattern 2 |
| FLICKER_MEAS3 | OCBD - C0 | R/W | [31:0] | 00 | Score reg showing pattern matching pattern 3 |

### 4.16 Adaptive Phase Control (APC)

The APC block generates a 2-bit dither pattern for an 8-bit panel or a 4-bit dither pattern for a 6-bit panel to visually improve the amplitude resolution of the 10-bit RGB output signal.

### 4.16.1 Function

The heart of the APC block consists of a $32 \times 32 \times 4$ bit lookup table (LUT). It represents one threshold matrix, which can be read using a programmable addressing technique as well as a programmable dither threshold control. The panel depth APC_CTRLO[1] should match the bit depth of the panel and is not masked by APC enable APC_CTRLO[0]. When APC_CTRLO[0] is cleared, the dither pattern is set to zero.

### 4.16.2 Addressing Technique

The APC block offers an I ${ }^{2}$ C programmable addressing technique to generate various temporal dither patterns. The frame offset APC_CTRL1[7:4] is a 4-bit increment value, which defines the horizontal/vertical displacement of the dither matrix from frame to frame. After the frame length APC_CTRL1[3:0] + 1 number of frames, both horizontal and vertical displacement positions will be reset to zero, only when the frame length APC_CTRL1[3:0] $>0$.
Note: To set the frame accumulator to zero, the frame offset APC_CTRL1[7:4] must be programmed to 0 , and the frame length APC_CTRL1[3:0] to 1.
The frame offset can be independently activated in the horizontal and vertical dimension using respectively APC_CTRLO[5] and APC_CTRLO[6]. In addition, APC_CTRLO[7] enables a horizontal displacement increment of the frame offset APC_CTRL1[7:4] per color component.

### 4.16.3 Dither threshold Control

When the panel depth APC_CTRLO[1] is set to 0 , the 4 -bit LUT output value maps to a 2 -bit value for 8 -bit panels.

APC_CTRLO[4] enables symmetric clipping of white levels respectively black levels for 6-bit panels as well as 8 -bit panels.

RGB offset APC_CTRLO[3] enables a different dither amplitude offset for each color component. When the frame inversion APC_CTRLO[2] is set to 1, the dither amplitude is inverted every other frame.
A Matlab file is provided to generate a variety of different threshold matrices.
Table 32: APC Registers

| Register Name | Addr | Mode | Bits | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |

### 4.17 Output Mux (OMUX)

The OMUX block formats the 1 ppc 24bpp data stream from the data path into a single or 2 ppc pixel stream for the flat panel using RSDS or LVDS signaling at the pins.

Table 33: OMUX Registers (Sheet 1 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| OMUX_CTRLO | 0C30 | R/W | [7:4] | 00 | RGB data channel reordering: <br> 0 : no changes on RGB data <br> 2: Right shift 2 bits <br> A: Right rotate 2 bits <br> C: Right rotate 4 bits <br> E: Right rotate 6 bits <br> All other values: reserved |
|  |  | R/W | [3] |  | 1: flip MSB to LSB per color (8 bits) |
|  |  | R/W | [2] |  | 1: swap $R$ and $B$ data |
|  |  | R/W | [1] |  | $0 *$ : <br> - in 1ppc, A channel active <br> - in 2ppc, Left on A, Right on B <br> 1: <br> - in 1ppc, B channel active <br> - in 2ppc, Left on B, Right on A |
|  |  | R/W | [0] |  | $\begin{aligned} & 0^{*}: 1 \mathrm{ppc} \\ & 1: 2 \mathrm{ppc} \end{aligned}$ <br> Forced to 1 ppc in LVDS debug or RSDS mode (refer to OMUX_TEST register) |
| OMUX_CTRL1 | 0C31 | R/W | [7] | 00 | LVDS reserved bit $0 *$ : previous bit 1: TCON[7] |
|  |  | R/W | [6] |  | 1: LVDS channel 0 to channel 3 flip and channel 4 to channel 7 flip |
|  |  | R/W | [0] |  | 1: LVDS outputs active (see Table 34) |
| OMUX_CTRL2 | 0C32 | R/W | [7] | 00 | 1: invert LVDS channel 7 |
|  |  | R/W | [6] |  | 1: invert LVDS channel 6 |
|  |  | R/W | [5] |  | 1: invert LVDS channel 5 |
|  |  | R/W | [4] |  | 1: invert LVDS channel 4 |
|  |  | R/W | [3] |  | 1: invert LVDS channel 3 |
|  |  | R/W | [2] |  | 1: invert LVDS channel 2 |
|  |  | R/W | [1] |  | 1: invert LVDS channel 1 |
|  |  | R/W | [0] |  | 1: invert LVDSchannel 0 |

Table 33: OMUX Registers (Sheet 2 of 3)


Table 33: OMUX Registers (Sheet 3 of 3)

| Register Name | Addr | Mode | Bits | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| OMUX_CTRL7 | 0 C 37 | R/W | [7] | 00 | 1: invert RSDS clock 1 (RSDS data pair 12) |
|  |  | R/W | [6] |  | $0^{*}$ : normal LVDS PLL clock if LVDS mode (normal) <br> 1: invert LVDS PLL clock if LVDS mode, or invert RSDS clock 0 (RSDS data pair 21) if RSDS mode |
|  |  | R/W | [4] |  | 1: invert LVDS output DE |
|  |  | R/W | $\begin{aligned} & {[1]} \\ & {[0]} \end{aligned}$ |  | TCON remapped to PWM <br> TCON[1] = pwm_a enable <br> TCON[0] = pwm_b enable |
| OMUX_HALF_LINE_L | $0 \mathrm{C} 38$ | $\mathrm{R} / \mathrm{W}$ | $[7: 0]$ | 00 | RSDS split buffer half line address = out_hpixel/2. <br> out_hpixel has to be multiples of 4 . E.g. for SXGA panel (1280) the value is 640 |
| OMUX_HALF_LINE_U | 0C39 | R/W | [3:0] | 00 |  |
| OMUX_TEST | 0С3A | R/W | [1] | 00 | 1: enable RSDS debug mode |
|  |  | R/W | [0] |  | 1: enable LVDS debug mode |

Table 34: OMUX_CTRL Output Modes

| OUTPUT MODE | OMUX_CTRL1 [0] | OMUX_CTRL3 [0] |
| :--- | :--- | :--- |
| idle | 0 | 0 |
| LVDS mode | 1 | 0 |
| RVDS mode | 0 | 1 |

The omux architecture consists of 2 main blocks as shown in Figure 19.
Figure 19: OMUX Architecture


The split line buffer can delay and re-interleave the input pixel stream so that a 2 ppc output can drive both the first and the half line pixels simultaneously. This is commonly used for TCON applications where the column drivers are split into two groups (left and right halves of the screen) and driven at $1 / 2$ the pixel rate. Control signals need to be similarly delayed in the TCON to account for the $1 / 2$ line temporal shift. Latency is not important as long as the timing relationship between HSync, vsync, enable and data is preserved at the output.

Figure 20: Mux block diagram


### 4.17.1 Output Data

## LVDS

56 bits of LVDS data are arranged as shown in Table 35:
Table 35: LVDS output data

| LVDS <br> Output | LVDS Data | [6] | [5] | [4] | [3] | [2] | [1] | [0] |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| OUT0 | Ivds_data_o[6:0] | AR0 | AR1 | AR2 | AR3 | AR4 | AR5 | AG0 |
| OUT1 | Ivds_data_o[13:7] | AG1 | AG2 | AG3 | AG4 | AG5 | AB0 | AB1 |
| OUT2 | Ivds_data_o[20:14] | AB2 | AB3 | AB4 | AB5 | HS | VS | DE |
| OUT3 | Ivds_data_o[27:21] | AR6 | AR7 | AG6 | AG7 | AB6 | AB7 | AReserved |
| OUT4 | Ivds_data_o[34:28] | BR0 | BR1 | BR2 | BR3 | BR4 | BR5 | BG0 |
| OUT5 | Ivds_data_o[41:35] | BAG1 | BG2 | BG3 | BG4 | BG5 | BB0 | BB1 |
| OUT6 | Ivds_data_o[48:42] | BB2 | BB3 | BB4 | BB5 | HS | VS | DE |
| OUT7 | Ivds_data_o[55:49] | BR6 | BR7 | BG6 | BG7 | BB6 | BB7 | BReserved |

## MSB-LSB Flip

If omux_ctrl1[6] is equal to 1 , data are flipped as follows:
Ivds_data_out[27:0] =
\{lvds_data_o[6:0],Ivds_data_o[13:7],Ivds_data_o[20:14],Ivds_data_o[27:21]\}
Ivds_data_out[55:28] =
\{Ivds_data_o[34:28],Ivds_data_o[41:35],Ivds_data_o[48:42],Ivds_data_o[55:49]\}

## RSDS 128 pin and 100 pin

In RSDS mode, 24/48 data bits are combined into $12 / 24$ pairs for 1 ppc and 2 ppc modes, respectively.
The split line buffer is to be run in 2 ppc RSDS mode 128 pin only.



| OUTPUT INTERFACE |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| PIN \# (LQFP 128) | PIN \# (LQFP 100) | (RSDS INPUT NAME) PIN NAME | OUTPUT MODE |  |  |  |  |  |
|  |  |  | LVDS |  | RSDS (LQFP-128) |  | RSDS (LQFP-100) |  |
| 70 | 58 | TCONO | pwm_en ? <br> pwm_b: tcon0 | TCON SIGNALS | pwm_en ? <br> pwm_b: tcon0 | TCON SIGNALS | pwm_en ? <br> pwm_b: tcon0 | TCON SIGNALS |
| 71 | 59 | TCON1 | pwm_en? <br> pwm_a: tcon1 |  | pwm_en? <br> pwm_a: tcon1 |  | pwm_en? <br> pwm_a: tcon1 |  |
| 72 | 60 | TCON2 | tcon2 |  | tcon2 |  | tcon2 |  |
| 73 | 61 | TCON3 | tcon3 |  | tcon3 |  | tcon3 |  |
| 74 | 62 | TCON4 | tcon4 |  | tcon4 |  | tcon4 |  |
| 75 | 63 | TCON5 | tcon5 |  | tcon5 |  | tcon5 |  |
| 76 | 64 | TCON6 | tcon6 |  | tcon6 |  | tcon6 |  |
| 77 | 65 | TCON7 | tcon7 |  | tcon7 |  | tcon7 |  |

## Debug Mode

If LVDS debug mode is enabled (omux_test[0] = 1), LVDS output data will be set to a static 7-bit pattern which is programmed in omux_ctr14[6:0]

If RSDS debug mode is enabled (omux_test[1] = 1 ), RSDS output data will be set to a static pattern which is programmed in omux_ctr14[1:0].

### 4.17.2 Output Clocks

Output clock (to LVDS PLL) for both functional and test modes is the divide-by-2 clock generated inside omux. This clock is flopped on the falling edge of fsyn_outclk providing a $1 / 4$ phase offset between clock and data.

RSDS output clocks 0 \& 1 are set to fsyn_outclk_div2_dly for both functional and test modes. This clock has a programmable delay offset from the fsyn_outclk_div2. This is to ensure that data will meet the setup/hold requirements at the destination (panel.)
The out_enab signal (from the TCON block) must be programmed so that its left (rising) edge is odd in 2 ppc RSDS mode.

### 4.17.3 Clock Sources and Timing Considerations

The omux block operates on dotclk with the exception of omux_mux which runs on fsyn_outclk.
Table 2.4 describes the relationship between fsyn_outclk, fsyn_outclk_div2 and dotclk.
Table 36: Clock relationship

|  | $\mathbf{1}$ ppc | 2 ppc |
| :---: | :--- | :--- |
| fsyn_outclk_freq | 2x dotclk_freq | dotclk_freq |
| dotclk source sel | fsyn_outclk_div2 half <br> speed | fsyn_outclk full speed |
| GLBL_CLK_SRC_SEL_0[6:4] | 2 | 3 |
| GLBL_CLK_SRC_SEL_1[6:4] | 3 | 3 |
| FSYN_PR_OTCLK | $2^{\wedge 22 ~ * ~ x c l k \_f r e q ~ / ~}$ <br> dotclk_freq | $2^{\wedge 21 ~ * ~ x c l k \_f r e q ~ / ~}$ <br> dotclk_freq |

### 4.18 Timing Controller (TCON)

The Timing Controller block provides all output timing signals for panel applications.
Features include:

- comparator, pulse and window functions
- LC polarity inversion function generator
- separate logic and output crossbars
- out_HSync, out_vsync and out_enab generation
- register shadowing

Figure 21: Output timing


Figure 22: TCON schematic


Figure 23: Toggle Generator


The toggle generator facilitates the synthesis of polarity signals from internal TCON signals; the horizontal TCON_COMP and vertical TCON_PULSE signals. The selected inputs supply clock and enable signals (resp.) for a 2-bit incrementing counter and a toggle flop that output 3 toggle and 1 polarity signals. The vlen variable sets the counter maximum, which controls the vertical sequence. Input and vlen selection are all in the TCON_POLARITY_CTRL register.
Common types of polarity signals are given below. For synchronization of polarity and vtog_count, a special sync mode should be entered for one frame to initialize the polarity pattern relative to the first line of vmask.

| counter, decoder |  | frame number |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| vlen | vtoggle | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |  |  |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |
|  | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |  |  |
|  | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |  |  |
|  | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |
| 2 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  |  |
|  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |  |  |
|  | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |  |  |

Table 37: Polarity programming examples.

| polarity type | vmask pulse length | vlen |
| :--- | :--- | :--- |
| pol toggle every line, invert frame to frame, steady during vblank (2 <br> frame sequence) | odd, usually vpixel+1 or vpixel-1 | 0 |
| polarity toggle every other line, invert frame to frame, steady during <br> vblank (2 frame sequence) | odd*2, usually vpixel+2 or vpixel -2 | 1 |
| polarity toggle every 3 ${ }^{\text {rd }}$ line, invert frame to frame, steady during <br> vblank (2 frame sequence) | odd*3 | 2 |
| polarity toggle every other line, walking pattern (4 frame sequence) | odd, usually vpixel+1 or vpixel-1 | 1 |

Table 38: Video Pipeline Latency information

| Block | Output pixel video pipeline latency <br> (in per block dotclk units) |
| :--- | :--- |
| PGEN (*) | $+3(+16$ vs TCON window H values) |
| SRGB (*) | $+6(+13$ vs TCON window H values) |
| GAMMA | +3 |
| OSD (*) | $+3(+4$ vs TCON window H values) |
| APC | +1 |
| TCON | Zero Reference |
| LVDS (pixel delay up to LVDS Tx) | 1 ppc: 5 pixels, 2 ppc 6 pixels |
| RSDS (delay up to the RSDS pads) | $1 \mathrm{ppc}: 5$ pixels, <br> $2 p p c ~ w / ~ s p l i t ~ l i n e ~ b u f f e r=640: ~$ <br> $640+12$ pixels |

(*): Block having a window control feature

Table 39: Register Map (Sheet 1 of 7)

| Register Name | Addr. | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_CTRL | OB00 | [6:4] | R/W | 00 | out_venab source selection <br> 0*: out_venab generated from out_enab (normal) <br> 1: tcon_pgen <br> 2: window venab[0] <br> 3: window venab[1] <br> 4: window venab[2] <br> 5: window venab[3] <br> 6-7: reserved |
|  |  | [3:2] | R/W |  | i2c block transfer (not tcon) event selection <br> $0 *$ : (hcount $==0$ ) \&\& (vcount $==0)$ <br> 1: (hcount $==0$ ) <br> 2: sttd0 <br> 3: sttd1 |
|  |  | [0] | R/W |  | TCON[7:0] output enable. Internal signals are always active. |
| TCON_POLARITY_CTRL | OB01 | [7:6] | R/W | 00 | vlen = toggle/polarity line sequence length (desired - 1) |
|  |  | [5:4] | R/W |  | vtoggle / polarity horizontal reference (1 of 4 comparators) |
|  |  | [2:0] | R/W |  | polarity vmask selection <br> 0 *: pulse 0 <br> 1: pulse 1 <br> 2: pulse 2 <br> 3: pulse 3 <br> 4: pulse 4 <br> 5: pulse 5 <br> 6: pulse 0 , reset vtog_count to 0 at rising edge of vmask, polarity reset to 0 <br> 7: pulse 0 , resync vtog_count to 1 at rising edge of vmask, polarity reset to 0 Note: pulse type must be vertical |
| TCON_INV_0 | 0B02 | [7] | R/W | 00 | invert output tcon7 |
|  |  | [6] | R/W |  | invert output tcon6 |
|  |  | [5] | R/W |  | invert output tcon5 |
|  |  | [4] | R/W |  | invert output tcon4 |
|  |  | [3] | R/W |  | invert output tcon3 |
|  |  | [2] | R/W |  | invert output tcon2 |
|  |  | [1] | R/W |  | invert output tcon1 |
|  |  | [0] | R/W |  | invert output tcono |

Table 39: Register Map (Sheet 2 of 7)

| Register Name | Addr. | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_INV_1 | 0B03 | [7] | R/W | 00 | invert output osd_lut |
|  |  | [6] | R/W |  | invert output pgen |
|  |  | [5] | R/W |  | invert output gamma_2 |
|  |  | [4] | R/W |  | invert output gamma_1 |
|  |  | [3] | R/W |  | invert output srgb |
|  |  | [2] | R/W |  | invert output out_enab |
|  |  | [1] | R/W |  | invert output out_vsync |
|  |  | [0] | R/W |  | invert output out_HSync |


| TCON_SHADOW_CTRL | 0B04 | [7:4] |  | 00 | shadow target 00*: comp 0 01: comp 1 02: comp 2 03: comp 3 04: pulse 0 05: pulse 1 06: pulse 2 07: pulse 3 08: pulse 4 09: pulse 5 0A: window 0 OB: window 1 OC: window 2 0D: window 3 0E: polarity OF: reserved |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | [3:2] | R/W |  | ```tcon shadow event selection 0*: (hcount == 0) && (vcount == 0) 1: (hcount == 0) 2: srtd0 3: srtd1``` |
|  |  | [1] | R/W |  | shadow transfer enable <br> - set to transfer at next event <br> - bit is automatically cleared when transfer is complete |
|  |  | [0] | R/W |  | shadow enable |
| TCON_SHADOW_BUF_0 | 0B05 | [7:0] | R/W | 00 | shadow buffer 0 |
| TCON_SHADOW_BUF_1 | 0B06 | [4:0] | R/W | 00 | shadow buffer 1 |
| TCON_SHADOW_BUF_2 | 0B07 | [7:0] | R/W | 00 | shadow buffer 2 |
| TCON_SHADOW_BUF_3 | 0B08 | [7:0] | R/W | 00 | shadow buffer 3 |

Table 39: Register Map (Sheet 3 of 7)

| Register Name | Addr. | Bits | Mode | Rst | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TCON_SHADOW_BUF_4 | 0B09 | $[7: 0]$ | R/W | 00 | shadow buffer 4 |
| TCON_SHADOW_BUF_5 | 0B0A | $[3: 0]$ | R/W | 00 | shadow buffer 5 |
| TCON_SHADOW_BUF_6 | 0B0B | $[7: 0]$ | R/W | 00 | shadow buffer 6 |
| TCON_SHADOW_BUF_7 | 0B0C | $[4: 0]$ | R/W | 00 | shadow buffer 7 |


| TCON_COMP_0_L TCON_COMP_0_U | $\begin{aligned} & \text { OB10 } \\ & \text { OB11 } \end{aligned}$ | [7:0] <br> [4] [3:0] | R/W <br> R/W <br> R/W | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | count comparison value <br> $0 *$ : horizontal count compare <br> 1: vertical count compare count comparison value |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_COMP_1_L <br> TCON_COMP_1_U | $\begin{aligned} & \text { OB12 } \\ & \text { OB13 } \end{aligned}$ | refer to TCON_COMP_0 |  |  |  |
| TCON_COMP_2_L <br> TCON_COMP_2_U | $\begin{aligned} & \text { OB14 } \\ & \text { 0B15 } \end{aligned}$ | refer to TCON_COMP_0 |  |  |  |
| TCON_COMP_3_L TCON_COMP_3_U | $\begin{aligned} & \text { OB16 } \\ & \text { OB17 } \end{aligned}$ | refer to TCON_COMP_0 |  |  |  |


| TCON_PULSE_O_SET_L TCON_PULSE_O_SET_U | $\begin{aligned} & \text { OB18 } \\ & \text { 0B19 } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | set point compare value set point compare value |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_PULSE_O_RST_L TCON_PULSE_O_RST_U | $\begin{aligned} & \text { OB1A } \\ & \text { 0B1B } \end{aligned}$ | [7:0] <br> [7:6] <br> [5:4] <br> [3:0] | R/W <br> R/W <br> R/W <br> R/W | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | reset point compare value <br> for vertical pulses, 1 of 4 comparators is selected to define the horizontal change point <br> $0^{*}$ : horizontal pulse <br> 1: vertical pulse <br> 2,3: single point, set=h, rst=v <br> reset point compare value |
| TCON_PULSE_1_SET_L <br> TCON_PULSE_1_SET_U <br> TCON_PULSE_1_RST_L <br> TCON_PULSE_1_RST_U | $\begin{aligned} & \text { OB1C } \\ & \text { OB1D } \\ & \hline \text { OB1E } \\ & \text { OB1F } \end{aligned}$ | refer to TCON_PULSE_0 |  |  |  |
| TCON_PULSE_2_SET_L TCON_PULSE_2_SET_U TCON_PULSE_2_RST_L TCON_PULSE_2_RST_U | $\begin{aligned} & \text { OB20 } \\ & \text { OB21 } \\ & \hline \text { OB22 } \\ & \text { OB23 } \end{aligned}$ | refer to TCON_PULSE_0 |  |  |  |
| TCON_PULSE_3_SET_L <br> TCON_PULSE_3_SET_U <br> TCON_PULSE_3_RST_L <br> TCON_PULSE_3_RST_U | $\begin{aligned} & \text { OB24 } \\ & \text { OB25 } \\ & \hline \text { OB26 } \\ & \text { OB27 } \end{aligned}$ | refer to TCON_PULSE_0 |  |  |  |

Table 39: Register Map (Sheet 4 of 7)

| Register Name | Addr. | Bits | Mode | Rst |
| :--- | :--- | :--- | :--- | :--- |


| TCON_WINDOW_0_LEFT_L TCON_WINDOW_0_LEFT_U | $\begin{aligned} & \text { OB30 } \\ & \text { 0B31 } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | left edge compare count left edge compare count |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_WINDOW_O_RIGHT_L TCON_WINDOW_O_RIGHT_U | $\begin{aligned} & \text { 0B32 } \\ & \text { 0B33 } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | $\begin{aligned} & \text { R/W } \\ & \text { R/W } \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | right edge compare count right edge compare count |
| TCON_WINDOW_0_TOP_L TCON_WINDOW_0_TOP_U | $\begin{aligned} & \text { 0B34 } \\ & \text { 0B35 } \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[3: 0]} \end{aligned}$ | R/W <br> R/W | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | top edge compare count top edge compare count |
| TCON_WINDOW_0_BOTTOM_L TCON_WINDOW_0_BOTTOM_U | $\begin{aligned} & \text { OB36 } \\ & \text { 0B37 } \end{aligned}$ | [7:0] <br> [4] <br> [3:0] | R/W <br> R/W <br> R/W | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | bottom edge compare count <br> $0^{*}$ : window <br> 1: pulse start at (left, top), end at (right, bottom) bottom edge compare count |
| TCON_WINDOW_1_LEFT_L TCON_WINDOW_1_LEFT_U | $\begin{aligned} & \text { OB38 } \\ & \text { 0B39 } \end{aligned}$ | refer to TCON_WINDOW_0 |  |  |  |
| TCON_WINDOW_1_RIGHT_L TCON_WINDOW_1_RIGHT_U | $\begin{aligned} & \text { OB3A } \\ & \text { OB3B } \end{aligned}$ |  |  |  |  |
| TCON_WINDOW_1_TOP_L TCON_WINDOW_1_TOP_U | $\begin{aligned} & \text { OB3C } \\ & \text { OB3D } \end{aligned}$ |  |  |  |  |
| TCON_WINDOW_1_BOTTOM_L TCON_WINDOW_1_BOTTOM_U | 0B3E <br> 0B3F |  |  |  |  |
| TCON_WINDOW_2_LEFT_L TCON_WINDOW_2_LEFT_U | $\begin{aligned} & \text { OB40 } \\ & \text { 0B41 } \end{aligned}$ | refer to TCON_WINDOW_0 |  |  |  |
| TCON_WINDOW_2_RIGHT_L TCON_WINDOW_2_RIGHT_U | $\begin{aligned} & \text { OB42 } \\ & \text { 0B43 } \end{aligned}$ |  |  |  |  |
| TCON_WINDOW_2_TOP_L TCON_WINDOW_2_TOP_U | $\begin{aligned} & \text { OB44 } \\ & \text { 0B45 } \end{aligned}$ |  |  |  |  |
| TCON_WINDOW_2_BOTTOM_L TCON_WINDOW_2_BOTTOM_U | $\begin{aligned} & \text { OB46 } \\ & \text { 0B47 } \end{aligned}$ |  |  |  |  |

Table 39: Register Map (Sheet 5 of 7)

\left.| Register Name | Addr. | Bits | Mode | Rst |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TCON_WINDOW_3_LEFT_L |  |  |  |  |  |
| TCON_WINDOW_3_LEFT_U |  |  |  |  |  |$\right)$

Table 39: Register Map (Sheet 6 of 7)

| Register Name | Addr. | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_X_SRTD_0_A | 0B80 | [4:0] | R/W | 00 | 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: window 09: window 0A: window OB: window OC: vtoggle OD: vtoggle OE: vtoggle OF: polarity 10: sttd0 <br> 11: srtd1 <br> 12: srtd2 <br> 13: srtd3 <br> 14: srtd4 <br> 15: srtd5 <br> 16: srtd6 <br> 17: srtd7 <br> 18: srtd8 <br> 19: srtd9 <br> 1A: srtd10 <br> 1B: srtd11 <br> 1C: comp0 <br> 1D: comp1 <br> 1E: comp2 <br> 1F: comp3 |
| TCON_X_SRTD_0_B | 0B81 |  |  |  |  |
| TCON_X_SRTD_1_A | 0B82 |  |  |  |  |
| TCON_X_SRTD_1_B | 0B83 |  |  |  |  |
| TCON_X_SRTD_2_A | 0B84 |  |  |  |  |
| TCON_X_SRTD_2_B | 0B85 |  |  |  |  |
| TCON_X_SRTD_3_A | OB86 |  |  |  |  |
| TCON_X_SRTD_3_B | 0B87 |  |  |  |  |
| TCON_X_SRTD_4_A | 0B88 |  |  |  |  |
| TCON_X_SRTD_4_B | 0B89 |  |  |  |  |
| TCON_X_SRTD_5_A | 0B8A |  |  |  |  |
| TCON_X_SRTD_5_B | 0B8B |  |  |  |  |
| TCON_X_SRTD_6_A | 0B8C |  |  |  |  |
| TCON_X_SRTD_6_B | 0B8D |  |  |  |  |
| TCON_X_SRTD_7_A | OB8E |  |  |  |  |
| TCON_X_SRTD_7_B | 0B8F |  |  |  |  |
| TCON_X_SRTD_8_A | 0B90 |  |  |  |  |
| TCON_X_SRTD_8_B | 0B91 |  |  |  |  |
| TCON_X_SRTD_9_A | 0B92 |  |  |  |  |
| TCON_X_SRTD_9_B | OB93 |  |  |  |  |
| TCON_X_SRTD_10_A | 0B94 |  |  |  |  |
| TCON_X_SRTD_10_B | 0B95 |  |  |  |  |
| TCON_X_SRTD_11_A | 0B96 |  |  |  |  |
| TCON_X_SRTD_11_B | 0B97 |  |  |  |  |
| TCON_X_SRTD_12_A | 0B98 |  |  |  |  |
| TCON_X_SRTD_12_B | 0B99 |  |  |  |  |
| TCON_X_SRTD_13_A | 0B9A |  |  |  |  |
| TCON_X_SRTD_13_B | 0B9B |  |  |  |  |
| TCON_X_SRTD_14_A | 0B9C |  |  |  |  |
| TCON_X_SRTD_14_B | 0B9D |  |  |  |  |
| TCON_X_SRTD_15_A | 0B9E |  |  |  |  |
| TCON_X_SRTD_15_B | 0B9F |  |  |  |  |

Table 39: Register Map (Sheet 7 of 7)

| Register Name | Addr. | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| TCON_X_0 | OBAO | [4:0] | R/W | 00 | output selection for tcon pin 0 <br> 00*: 0 <br> 01: 1 <br> 02: pulse0 <br> 03: pulse1 <br> 04: pulse2 <br> 05: pulse3 <br> 06: pulse4 <br> 07: pulse5 <br> 08: window0 <br> 09: window1 <br> 0A: window2 <br> OB: window3 <br> OC: vtoggle0 <br> 0D: vtoggle1 <br> 0E: vtoggle2 <br> 0F: polarity <br> 10: srtd8 <br> 11: srtd9 <br> 12: srtd10 <br> 13: srtd11 <br> 14: srtd12 <br> 15: srtd13 <br> 16: srtd14 <br> 17: srtd15 <br> 18-1F: reserved <br> [7:1] - Reserved <br> [0] - LVDS_DE_SOURCE_SELECT <br> 0: Use DE generated by TCON_OENAB <br> 1: Use DE generated by TCON_OSD_LUT <br> (without 16 pixels latency) |
| TCON_X_1 | OBA1 |  |  |  |  |
| TCON_X_2 | 0BA2 |  |  |  |  |
| TCON_X_3 | OBA3 |  |  |  |  |
| TCON_X_4 | OBA4 |  |  |  |  |
| TCON_X_5 | OBA5 |  |  |  |  |
| TCON_X_6 | 0BA6 |  |  |  |  |
| TCON_X_7 | 0BA7 |  |  |  |  |
| TCON_X_OHSYNC | 0BA8 |  |  |  |  |
| TCON_X_OVSYNC | 0BA9 |  |  |  |  |
| TCON_X_OENAB | OBAA |  |  |  |  |
| TCON_X_GAMMA_A | OBAB |  |  |  |  |
| TCON_X_GAMMA_B | OBAC |  |  |  |  |
| TCON_X_SRGB | OBAD |  |  |  |  |
| TCON_X_PGEN | OBAE |  |  |  |  |
| TCON_X_OSD_LUT | OBAF |  |  |  |  |
| SCL_TCON_I2C_SPARE_REG | 0x0A37 | [0] | R/W | 0 |  |

I2C shadow mode is supported for individual comparators, pulses and windows. New values are loaded into the shadow buffer area by slow I2C then the transfer command and shadow target are written into tcon_shadow_ctrl. At the next event, the data is transferred in a single clock cycle.

Table 40: Shadow Mapping

| source | comparator | pulse | window |
| :--- | :--- | :--- | :--- |
| tcon_shadow_buf_0[7:0] | tcon_comp_X[7:0] | tcon_pulse_X_set[7:0] | tcon_window_X_left[7:0] |
| tcon_shadow_buf_1[4:0] | tcon_comp_X[12:8] | tcon_pulse_X_set[11:8] | tcon_window_X_left[11:8] |
| tcon_shadow_buf_2[7:0] | NA | tcon_pulse_X_rst[7:0] | tcon_window_X_right[7:0] |
| tcon_shadow_buf_3[6:0] | NA | tcon_pulse_X_rst[15:8] | tcon_window_X_right[11:8] |
| tcon_shadow_buf_4[7:0] | NA | NA | tcon_window_X_top[7:0] |
| tcon_shadow_buf_5[3:0] | NA | NA | tcon_window_X_top[11:8] |
| tcon_shadow_buf_6[7:0] | NA | NA | tcon_window_X_bottom[7:0] |
| tcon_shadow_buf_7[4:0] | NA | NA | tcon_window_X_bottom[12:8] |

## TCON Example

The following is an example of a basic TCON script:

```
WriteByte (TCON_CTRL_EN, 0x01);// enable TCON output
// vsync start at vcount = 0, end at vcount = 1
WriteWord (TCON_PULSE_0_SET, 0x0000);// pulse 0 set = 0 (12 bit value)
WriteWord (TCON_PULSE_0_RST, 0x1001);// pulse 0 reset = 0x001 (12 bit),
// vertical pulse, comparator 0
// HSync start at hcount = 4, end at hcount = 6
WriteWord (TCON_PULSE_1_SET, 0x0004);// pulse 1 set = 0x004 (12 bit value)
WriteWord (TCON_PULSE_1_RST, 0x0006);// pulse 1 reset = 0x006, horiz pulse
// data enable start at upper left (31H,1V), ending at lower right (1311H, 1025V)
// for a 1280 x 1024 output enable
WriteWord (TCON_WINDOW_0_LEFT, 0x001F);// window 0 left edge comparison
// count = 0x01F (12 bit value)
WriteWord (TCON_WINDOW_0_RIGHT, 0x051F);// right edge count = 0x51F
WriteWord (TCON_WINDOW_0_TOP, 0x0001);// top edge count = 1
WriteWord (TCON_WINDOW_0_BOTTOM, 0x0400);// bottom edge = 0x400, window type
```

// select pulses and window for oHSync, ovsync, oenab
WriteByte (TCON_X_OHSYNC, 0x03) ;// HSync on TCON pin 0 is pulse 1
WriteByte (TCON_X_OVSYNC, 0x02);// vsync on TCON pin 0 is pulse 0
WriteByte (TCON_X_OENAB, 0x08); // out enable on pin 0 is window 0

### 4.19 LVDS/RSDS Features

The LVDS/RSDS block supports the following modes:

- LVDS 1 ppc
- 4 data channels +1 clock channel $-40 \mathrm{MHz}-85 \mathrm{MHz}$
- LVDS 2 ppc
- 8 data channels +2 clock channels $-40 \mathrm{MHz}-70 \mathrm{MHz}$
- RSDS 1 ppc
- 12 data channels +1 clock channel $-13.5 \mathrm{MHz}-85 \mathrm{MHz}$
- RSDS 2 ppc (128 pin package only)
-24 data channels +2 clock channels $-13.5 \mathrm{MHz}-70 \mathrm{MHz}$ Its features are as follows:
- Power down modes
- Programmable output swing and common mode voltage
- Per channel programmable delay
- Programmable LVDS clock output polarity


### 4.19.1 Output Channels

## 128 Pin Package

- 16 channels dedicated RSDS;
- 10 channels shared by LVDS or RSDS
- LVDS (1ppc): 4 data +1 clock $=5$ (others are unused)
- LVDS (2ppc): 8 data +2 clock $=10$
— RSDS: 10 data (both 1ppc and 2ppc)


## 100 Pin Package

- 3 channels dedicated to RSDS,
- 10 channels shared by LVDS or RSDS
- LVDS (1ppc): 4 data +1 clock $=5$ (others are unused)
- LVDS (2ppc): 8 data +2 clock $=10$
- RSDS: 10 data (1ppc on channel A only)

Table 41: LVDS/RSDS Registers (Sheet 1 of 5)

| Register Name | Address | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_LVDSANAO | 0060 | [7] | R/W | 84 | PLL Manual/Auto Select <br> 0 : manual (using ANA_LVDSANAO[1:0]) <br> 1*: auto |
|  |  | [6] | R/W |  | PLL Comparator Current Select <br> 0*: 300uA (normal) <br> 1: 200uA |
|  |  | [5:4] | R/W |  | PLL Charge Pump Current Select <br> 0*: 10uA (normal) <br> 1: 25uA <br> 2: 50 uA <br> 3: 100uA (fast response) |
|  |  | [1:0] | R/W |  | PLL Manual Range Select (enabled by ANA_LVDSANAO[7]) <br> 0*: 25uA (slowest) <br> 1: 75uA <br> 2: 125uA <br> 3: 175uA (fastest) |

Table 41: LVDS/RSDS Registers (Sheet 2 of 5)

| Register Name | Address | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_LVDSANA1 | 0061 | [7:6] | R/W | 00 | Bit 3 Data Interface Delay Adjustment, see Bit 0 |
|  |  | [5:4] | R/W |  | Bit 2 Data Interface Delay Adjustment, see Bit 0 |
|  |  | [3:2] | R/W |  | Bit 1 Data Interface Delay Adjustment, see Bit 0 |
|  |  | [1:0] | R/W |  | Bit 0 Data Interface Delay Adjustment <br> 0*: Ops (normal) <br> 1: 90ps <br> 2: 210ps <br> 3: 460ps |
| ANA_LVDSANA2 | 0062 | [7] |  | CO | PLL power control 0 : on $1^{*}$ : off |
|  |  | [6] |  |  | PLL Global Data Interface Delay <br> 0 : no delay <br> 1*: delay (normal) |
|  |  | [5:4] | R/W |  | Bit 6 Data Interface Delay Adjustment, see Bit 0 |
|  |  | [3:2] | R/W |  | Bit 5 Data Interface Delay Adjustment, see Bit 0 |
|  |  | [1:0] | R/W |  | Bit 4 Data Interface Delay Adjustment, see Bit 0 |
| ANA_LVDSANA4 | 0064 | [6:4] | R/W | 01 | LVDS Clock Skew LSB $=135$ ps (typ) |
|  |  | [3] |  |  | LVDS Clock Skew Enable $0^{*}$ : no delay (normal) 1: delay |
|  |  | [2] |  |  | LVDSclkout1 output polarity $0^{*}$ : normal <br> 1: invert |
|  |  | [1] |  |  | LVDSclkout0 output polarity 0*: normal <br> 1: invert |
|  |  | [0] |  |  | LVDS \& RSDS Master Power Control (Overrides ANA_LVDSANA5[7], ANA_LVDSANA6[7], and ANA_LVDSANA2[7]) |

Table 41: LVDS/RSDS Registers (Sheet 3 of 5)

| Register Name | Address | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_LVDSANA5 | 0065 | [7] | R/W | CO | LVDS B power control <br> (for LVDS Channel [7:4], LVDS clk 1) <br> 0 : on <br> $1^{*}$ : off |
|  |  | [6] |  |  | LVDS A power control <br> (for LVDS Channel [3:0], LVDS clk 0) <br> 0 : on <br> 1*: off |
|  |  | [5] |  |  | Output mode select <br> 0*: RSDS (also powers down PLL) <br> 1: LVDS |
|  |  | [4:0] |  |  | LVDS Iref Bias current setting <br> 10000: 420uA <br> 00011: 168uA <br> 00010: 178uA <br> 00001: 189uA <br> 00000*: 201uA (normal) <br> 11111: 202uA <br> 11110: 216.3uA <br> 11101: 233uA <br> 11100: 252uA |
| ANA_LVDSANA7 | 0067 | [7] | R | 00 | LVDS Channel [7:4] power status |
|  |  | [6] |  |  | LVDS Channel [3:0] power status |
|  |  | [5] |  |  | LVDS/RSDS/PLL Global Power status |
|  |  | [4] |  |  | PLL powerdown status = [ANA_LVDSANA4[0] OR ANA_LVDSANA2[7] OR (NOT ANA_LVDSANA5[5])] |
|  |  | [3] |  |  | PLL up status |
|  |  | [2] |  |  | PLL down status |
|  |  | [1:0] |  |  | PLL range status |

Table 41: LVDS/RSDS Registers (Sheet 4 of 5)

| Register Name | Address | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_LVDSSW_VC | 0068 | [6:4] | R/W | 00 | LVDS \& RSDS Output Common Mode Adjustment $\begin{aligned} & 0^{*}: 1.093 \mathrm{~V} \\ & 1: 1.119 \mathrm{~V} \\ & \text { 2: } 1.145 \mathrm{~V} \\ & \text { 3: } 1.171 \mathrm{~V} \text { (normal) } \\ & \text { 4: } 1.197 \mathrm{~V} \\ & \text { 5: } 1.223 \mathrm{~V} \\ & \text { 6: } 1.259 \mathrm{~V} \\ & \text { 7: } 1.274 \mathrm{~V} \end{aligned}$ |
|  |  | [3:0] |  |  | LVDS \& RSDS Swing Adjustment $\begin{aligned} & \text { 0*: 170mV (normal) } \\ & \text { F: } 475 \mathrm{mV} \\ & \text { LSB }=20 \mathrm{mV} \text { (typ) } \end{aligned}$ |
| ANA_LVDSCOMPV | 0069 | [6:4] |  | 00 | VRL regulator current adjust 0*: off <br> 1: 18uA (normal) <br> 2: 36uA <br> 3: 54uA <br> 4: 72uA <br> 5: 90uA <br> 6: 108uA <br> 7: 126uA |
|  |  | [2:0] |  |  | VRH regulator current adjust 0*: off <br> 1: 18uA (normal) <br> 2: 36uA <br> 3: 54uA <br> 4: 72uA <br> 5: 90uA <br> 6: 108uA <br> 7: 126uA |

Table 41: LVDS/RSDS Registers (Sheet 5 of 5)

| Register Name | Address | Bits | Mode | Rst |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| ANA_LVDS_DLY_0 | 006A | [6:4] | R/W | 33 | LVDSDLYCH1 | LVDS/RSDS output skew adjust |
|  |  | [2:0] |  |  | LVDSDLYCH0 |  |
| ANA_LVDS_DLY_1 | 006B | [6:4] | R/W | 33 | LVDSDLYCLK0 | $\begin{aligned} & \text { 0: 176ps (typ) } \\ & \text { 1: 104ps (typ) } \\ & \text { 2: 73ps (typ) } \end{aligned}$ |
|  |  | [2:0] |  |  | LVDSDLYCH2 |  |
| ANA_LVDS_DLY_2 | 006C | [6:4] | R/W | 33 | LVDSDLYCH4 | 3*: 50ps (typ) |
|  |  | [2:0] |  |  | LVDSDLYCH3 | 4: 39ps (typ) |
| ANA_LVDS_DLY_3 | 006D | [6:4] | R/W | 33 | LVDSDLYCH6 | 5: 18ps (typ) <br> 6: 7ps (typ) <br> 7: no delay (normal) |
|  |  | [2:0] |  |  | LVDSDLYCH5 |  |
| ANA_LVDS_DLY_4 | 006E | [6:4] | R/W | 33 | LVDSDLYCH7 |  |
|  |  | [2:0] |  |  | LVDSDLYCLK1 |  |
| ANA_RSDS_DLY_0 | 0070 | [6:4] | R/W | 33 | RSDSDLYCH1 |  |
|  |  | [2:0] |  |  | RSDSDLYCH0 |  |
| ANA_RSDS_DLY_1 | 0071 | [6:4] | R/W | 33 | RSDSDLYCH3 |  |
|  |  | [2:0] |  |  | RSDSDLYCH2 |  |
| ANA_RSDS_DLY_2 | 0072 | [6:4] | R/W | 33 | RSDSDLYCH5 |  |
|  |  | [2:0] |  |  | RSDSDLYCH4 |  |
| ANA_RSDS_DLY_3 | 0073 | [6:4] | R/W | 33 | RSDSDLYCH7 |  |
|  |  | [2:0] |  |  | RSDSDLYCH6 |  |
| ANA_RSDS_DLY_4 | 0074 | [6:4] | R/W | 33 | RSDSDLYCH9 |  |
|  |  | [2:0] |  |  | RSDSDLYCH8 |  |
| ANA_RSDS_DLY_5 | 0075 | [6:4] | R/W | 33 | RSDSDLYCH11 |  |
|  |  | [2:0] |  |  | RSDSDLYCH10 |  |
| ANA_RSDS_DLY_6 | 0076 | [6:4] | R/W | 33 | RSDSDLYCH13 |  |
|  |  | [2:0] |  |  | RSDSDLYCH12 |  |
| ANA_RSDS_DLY_7 | 0077 | [6:4] | R/W | 33 | RSDSDLYCH15 |  |
|  |  | [2:0] |  |  | RSDSDLYCH14 |  |

Table 42: LVDS / RSDS Power Configurations

| State | PLL | LVDS <br> Output | RSDS <br> Output | ANA_LVDSANA4[0] <br> Master Power Ctrl | ANA_LVDSANA5[5] <br> Output Mode Sel | ANA_LVDSANA2[7] <br> PLL Power Ctrl |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| All Off | off | off | off | 1 | X | X |
| LVDS On | on | on | off | 0 | 1 | 0 |
| RSDS On | off | off | on | 0 | 0 | X |

### 4.20 Pulse Width Modulation (PWM)

The Pulse Width Modulation block generates two signals that can be used to control backlight inverter switching power components directly. It is derived from XCLK and can be powered up independently of the DOTCLK and INCLK domains. The frequency, duty cycle, polarity and overlap/ non-overlap are programmable. The output frequency can be free-running or locked to the output vsync signal.

Table 43: PWM Registers (Sheet 1 of 2)

| Register Name | Addr | Mode | Bits | Default | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PWM_CTRLO | 01A0 | R | [7] | 00 | PWM status <br> $0^{*}$ : unlocked <br> 1: locked |
|  |  | R/W | [6] |  | 0*: lock to CYCLES_PER_FRAME from the free-running state machine <br> 1: lock to CYCLES_PER_FRAME register setting |
|  |  | R/W | [5] |  | PWM_A polarity <br> $0^{*}$ : active low <br> 1: active high |
|  |  | R/W | [4] |  | PWM_B polarity <br> $0^{*}$ : active low <br> 1: active high |
|  |  | R/W | [3] |  | 0*: normal operation <br> 1: force both PWM outputs to polarity settings of bits 5 and 4 |
|  |  | R/W | [2] |  | $0^{*}$ : change period or duty cycle at the end of the current cycle <br> 1: smooth change, period or duty cycle increment/decrement every PWM_STEP_DELAY cycle |
|  |  | R/W | [1] |  | $0^{*}$ : free-running <br> 1: lock to out_vsync |
|  |  | R/W | [0] |  | 0*: disable PWM output <br> 1: enable PWM output |
| PWM_CTRL1 | 01A1 | R/W | [7:4] | 00 | Lock $2^{\text {nd }}$ order gain (power of 2) $0^{*}$ : max <br> 3: typical <br> F: min. |
|  |  | R/W | [3:0] |  | Lock gain (power of 2) 0*: max <br> 6: typical <br> F: min. |
| PWM_PERIOD_L PWM_PERIOD_U | $\begin{aligned} & \text { 01A2 } \\ & \text { 01A3 } \end{aligned}$ | $\begin{aligned} & \hline R / W \\ & R / W \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Period-2 in free-running mode, in XCLKs |
| PWM_DUTY_L PWM_DUTY_U | $\begin{aligned} & \hline 01 \mathrm{~A} 4 \\ & \text { 01A5 } \end{aligned}$ | $\begin{aligned} & \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Duty cycle of PWM in XCLKs |
| PWM_OVERLAP_L PWM_OVERLAP_U | $\begin{aligned} & \text { 01A6 } \\ & \text { 01A7 } \end{aligned}$ | $\begin{aligned} & \hline \mathrm{R} / \mathrm{W} \\ & \mathrm{R} / \mathrm{W} \end{aligned}$ | $\begin{aligned} & {[7: 0]} \\ & {[7: 0]} \end{aligned}$ | $\begin{aligned} & 00 \\ & 00 \end{aligned}$ | Non-overlap of PWMs in XCLKs |

Table 43: PWM Registers (Sheet 2 of 2)

| Register Name | Addr | Mode | Bits | Default | Description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| PWM_STEP_DELAY | 01 A8 | R/W | $[7: 0]$ | 00 | In smooth change mode, the number of <br> cycles skipped before the period/duty <br> registers are incremented/decremented |
| PWM_CYCLES_PER_FRAME_L <br> PWM_CYCLES_PER_FRAME_U | 01 A9 | 01AA | R/W | $[7: 0]$ | 00 |
| $[7: 0]$ | 00 | The number of cycles per frame in frame lock <br> mode when not using the internally generated <br> cycles per frame from a previous free-running <br> mode |  |  |  |

### 4.21 I²C Block Transfer (I2CBKT)

The block transfer function allows the internal $\mathrm{I}^{2} \mathrm{C}$ parallel bus to be driven by an xclk state machine to perform fast block transfers between internal addresses without any MCU software overhead.
Transfer speed is approximately 2MByte per second under typical conditions.

### 4.21.1 Transfer Setup and Start

Writing the bit I2CBKT_CTRL[0] to 1 initiates the transfer, according to all source and destination parameters (addresses, length):

- Length for source is programmable to allow repeated patterns/fills, such as filling an entire area with the same byte(s)
- An increment register for the destination allows to fill it only every $\mathrm{n}^{\text {th }}$ byte

Depending on the increment value, the destination length must be programmed as follows:

- If I2CBKT_CTRL[3:2]=0 ( or =1 with I2CBKT_INC=1): DESLEN = nb of bytes to transfer
- If I2CBKT_CTRL[3:2]=1 with I2CBKT_INC>1: DESLEN = (nb of bytes to transfer *INC) - 1

The transfer can either take place immediately, or be initiated by a number of selectable events coming from SMUX or TCON, as programmed in I2CBKT_CTRL[6:4].
Transfers can occur between RAM or registers or both, but cannot take place in the own registers of the I2CBKT block (refer to Section 4.21.3: Concurrent I2C Transfers below).
Source and destination addresses cannot overlap.
Data can be either transferred from source to destination (one way) or swapped between them, depending on I2CBKT_CTRL[1].

### 4.21.2 Transfer Progress

The status bit I2CBKT_STATUS[0] is set to 1 by hardware as soon as the transfer actually starts, and falls back to 0 when the transfer is completed.
Note: It is the software's duty to write I2CBKT_CTRL[0] to 0 upon transfer completion, before preparing any new subsequent I2CBKT transfer.

### 4.21.3 Concurrent $\mathrm{I}^{2} \mathrm{C}$ Transfers

While the I2CBKT block is operating, only $I^{2} \mathrm{C}$ accesses from MCU to the I2CBKT registers listed below are allowed: any I ${ }^{2} \mathrm{C}$ access to other adresses will take priority and stop the I2CBKT transfer in progress in an unknown state (there is no way to tell which bytes have been transferred up to that point).

It is therefore strongly recommended to wait until the I2CBKT transfer in progress is completed, before initiating any ${ }^{2} \mathrm{C}$ access other than polling the I2CBKT_STATUS register.
Note: In case of need, a clean way to stop the current I2CBKT transfer is to write I2CBKT_CTRL[0] to 0 .

Table 44: I2C Block Transfer Registers

| Register Name | Addr | Bits | Mode | Rst | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| I2CBKT_INC | 0021 | [7:0] | R/W | 00 | destination address increment, 1 to 255 allowed |
| I2CBKT_SRCLEN_L | 0022 | [7:0] | R/W | 00 | length of source block, in bytes. <br> If source length < destination length, the source data is repeated |
| I2CBKT_SRCLEN_U | 0023 | [7:0] | R/W | 00 |  |
| I2CBKT_DESLEN_L | 0024 | [7:0] | R/W | 00 | length of block transfer, in bytes. <br> Include effect of increment if I2CBKT_CTRL[3:2] = 1 |
| I2CBKT_DESLEN_U | 0025 | [7:0] | R/W | 00 |  |
| I2CBKT_SRC_L | 0026 | [7:0] | R/W | 00 | source starting address |
| I2CBKT_SRC_U | 0027 | [7:0] | R/W | 00 |  |
| I2CBKT_DES_L | 0028 | [7:0] | R/W | 00 | destination starting address |
| I2CBKT_DES_U | 0029 | [7:0] | R/W | 00 |  |
| I2CBKT_CTRL | 002A | [6:4] | R/W | 00 | transfer start condition select (level sensitive) <br> 0*: immediate <br> 1: when in_henab $=0$ <br> 2: when out_henab $=0$ <br> 3: when in_venab $=0$ <br> 4: when out_venab $=0$ <br> 5: tcon_i2c_transfer = 1 (refer to TCON_CTRL[3:2]) |
|  |  | [3:2] | R/W |  | increment mode <br> $0^{*}$ : source +1 , dest + 1 <br> 1: source + 1, dest + inc (as set in I2CBKT_INC) <br> 2: reserved <br> 3: reserved |
|  |  | [1] | R/W |  | $0 *$ : one way transfer from source to destination <br> 1: swap source and destination |
|  |  | [0] | R/W |  | $0^{*}$ : end of transfer, or stop transfer in progress 1: start transfer according to condition bits [6:4] Must be set and cleared by software |
| I2CBKT_PULSE | 002B | [7:4] | R/W | 31 | read pulse width (reserved) |
|  |  | [3:0] | R/W |  | write pulse width (reserved) |
| I2CBKT_STATUS | 002C | [0] | R | 00 | Transfer status <br> 0*: block transfer completed <br> 1: block transfer in progress |

## EXAMPLE

Fill every other byte of the entire OSD_RAM with a byte previously stored at address 4700 : I2CBKT_SRC_L = 00, I2CBKT_SRC_U = 47: start address where the data is located I2CBKT_SRCLEN_L = 01, I2CBKT_SRCLEN_U = 00: only 1 byte to transfer from source

I2CBKT_DES_L = 00, I2CBKT_DES_U = 17: destination start address (OSD_RAM) where the data will be written

I2CBKT_INC = 02: skip every other byte
I2CBKT_DESLEN_L = FF, I2CBKT_DESLEN_U = 5F: $(46 F F-1700+1)=12288$ bytes to transfer means destination length $=(12288 \mathrm{x}$ increment $)-1=5$ FFF
I2CBKT_CTRL $=05$ : immediate transfer with source +1 and destination +2

### 4.22 $I^{2} \mathrm{C}$ Registers and RAM Addresses

The $\mathrm{I}^{2} \mathrm{C}$ own address of the device (also called "ADE_ID") is A8h.

### 4.22.1 $\quad{ }^{2} \mathrm{C}$ Transfer Format

All $I^{2} \mathrm{C}$ addresses, registers and RAM, are 16-bit wide.
Address LSB must be transferred first, followed by MSB then the data, as in the following ${ }^{2} \mathrm{C}$ write access example:

| Start | ADE_ID <br> (A8) | Ack | Register <br> Address <br> LSB | Ack | Register <br> Address <br> MSB | Ack | Data 1 | Ack | $\ldots$..further data... | Stop |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |

a. All Ack bits are returned by the device.

### 4.22.2 Dedicated RAM Areas per Block

Table 45: ${ }^{2}$ C RAM Addresses

| Name | Description | Block | Clock Condition ${ }^{\text {a }}$ | Start <br> Addr | End <br> Addr | Size | Size in Bytes |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| GAM_R | Gamma Red LUT | GAMMA | dotclk >= sclk | 1000 | 10FF | 256x8b | 256 |
| GAM_G | Gamma Green LUT |  |  | 1100 | 11FF | 256x8b | 256 |
| GAM_B | Gamma Blue LUT |  |  | 1200 | 12FF | 256x8b | 256 |
| OSD_RAM | Characters RAM Area | OSD |  | 1700 | 46FF | 4096x24b | 12288 |
| OSD_CLUT | Color LUT |  |  | 4700 | 47FF | 64x32b | 256 |
| SCL_RAM_1 | Line Buffers | SCL | sclk >= xclk | 9000 | A700 | 1024x42b | n/a |
| SCL_RAM_2 |  |  |  | A800 | BFFF | 1024x42b | n/a |
| OMUX | In RSDS Mode only ${ }^{\text {b }}$ | OMUX | dotclk >= sclk | E300 | F1FF | 640x48b | 3840 |

a. The relevant clock condition must be met to grant access to that block's registers and RAM.
b. In RSDS mode: OMUX uses this RAM area for internal computation purposes, it should not be otherwise modified by any means.
In LVDS mode, this RAM is free of use, and can be used as a temporary storage or working area for example.

### 4.22.3 Multi-byte Registers

Data are read back in the order of how they were written.
All values spread out over several registers are organised as follows:

| 32-bit values |  | 24-bit values |  | 16-bit values |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| _0 | LSB | _L or _0 | LSB | _L | LSB |
| -1 |  | _M or _1 | MSB | _U | USB |
| -2 |  | _U or _2 | USB |  |  |
| -3 | USB |  |  |  |  |

They are all LSB aligned, except for OMUX which is MSB aligned.
When the RAM width is not a multiple of 8, zeros will be returned for the non-meaningful bits.

## Example of LSB aligned RAM

If addresses 9000-9005 are written with the values F0-F5, the contents of SCL_RAM_1 (at word address 0 ) are as follows:

| $[41: 40]$ | $[39: 32]$ | $[31: 24]$ | $[23: 16]$ | $[15: 8]$ | $[7: 0]$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 01 | F4 | F3 | F2 | F1 | F0 |

A read from address 9000 will return F0; a read from address 9001 will return F1, etc.
Note: A read from 9005 returns the value 01 (as opposed to F5) since there are only 2 meaningful bits of data at this address.

## Example of MSB aligned RAM (OMUX only)

If addresses E300-E305 are written with the values F0-F5 respectively, the contents of the OMUX RAM (at word address 0 ) are as follows:

| $[47: 40]$ | $[39: 32]$ | $[31: 24]$ | $[23: 16]$ | $[15: 8]$ | $[7: 0]$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| F0 | F1 | F2 | F3 | F4 | F5 |

A read from address E300 will return F0, a read from address E301 will return F1, and so on.

## 5 Electrical Specifications

### 5.1 Absolute Maximum Ratings

| Symbol | Parameter | Min. | Typ. | Max. | Unit |
| :--- | :--- | :---: | :---: | :---: | :---: |
| AVDD <br> DVDD18 <br> XVDD18 <br> OVDD18 <br> PVDD18 <br> PLLVDD18 | 1.8 V Supply Voltages |  |  |  |  |
| DVDD33 |  |  |  | 1.95 | V |
| $\mathrm{~V}_{\text {ESD }}$ | 3.3V Supply Voltages |  |  | 3.6 | V |
| $\mathrm{~V}_{\text {IN5VTOL }}$ | Electrostatic Protection (Human Body Model) |  |  | 2 | kV |
| $\mathrm{V}_{\text {IN3VTOL }}$ | Max voltage on 5 volt tolerant input pins |  |  | 6.1 | V |
| $\mathrm{~T}_{\text {STG }}$ | Max voltage on 3.3 volt tolerant input pins | -40 |  | +150 | ${ }^{\circ} \mathrm{C}$ |
| $\mathrm{T}_{\text {OPER }}$ | Storage temperature | 0 |  | +70 | ${ }^{\circ} \mathrm{C}$ |
| $\mathrm{T}_{\mathrm{J}}$ | Operating Temperature | -40 |  | +125 | ${ }^{\circ} \mathrm{C}$ |

### 5.2 Nominal Operating Conditions

| Symbol | Parameter | Min. | Typ. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| AVDD DVDD18 XVDD18 OVDD18 PVDD18 PLLVDD18 | 1.8V Supply Voltages | 1.71 | 1.8 | 1.89 | V |
| DVDD33 | 3.3V Supply Voltages | 3.135 | 3.3 | 3.465 | V |
| $\mathrm{f}_{\text {XTAL }}$ | Crystal Frequency |  | 27 |  | MHz |
| $\mathrm{P}_{\text {XGA75LVDS }}$ | Power Consumption using XGA75Hz input and driving a XGA LVDS panel (1 pixel per clock) |  | 0.75 |  | W |
| P XGA75RSDS | Power Consumption using XGA75Hz input and driving a XGA RSDS panel (1 pixel per clock) |  | 0.70 |  | W |
| PsxGA75LVDS | Power Consumption using SXGA75Hz input and driving a SXGA LVDS panel (2 pixels per clock) |  | 1.10 |  | W |
| PSXGA75RSDS | Power Consumption using SXGA75Hz input and driving a SXGA RSDS panel (2 pixels per clock) |  | 1.00 |  | W |
| P PWRDN | Power Consumption in Power Down Mode | 0.04 | 0.05 |  | W |
| $\mathrm{I}_{\text {AVDDX75LVDS }}$ | AVDD Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 220 |  | mA |
| $\mathrm{I}_{\text {DVDD18X75LVDS }}$ | DVDD18 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 150 |  | mA |
| IXVDD18X75LVDS | XVDD18 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 2.5 |  | mA |
| IOVDD18X75LVDS | OVDD18 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 35 |  | mA |


| Symbol | Parameter | Min. | Typ. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{I}_{\text {PVDD18X75LVDS }}$ | PVDD18 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 5 |  | mA |
| IPLLVDD18X75LVDS | PLLVDD18 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 2.5 |  | mA |
| IDVDD33X75LVDS | DVDD33 Supply Current, (XGA75Hz input and XGA LVDS panel) |  | 2 |  | mA |
| $\mathrm{I}_{\text {AVDDX75RSDS }}$ | AVDD Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 220 |  | mA |
| $\mathrm{I}_{\text {DVDD18X75RSDS }}$ | DVDD18 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 150 |  | mA |
| ${ }^{\text {XVDD18X75RSDS }}$ | XVDD18 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 2.5 |  | mA |
| IOVDD18X75RSDS | OVDD18 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 10 |  | mA |
| $\mathrm{I}_{\text {PVDD18X75RSDS }}$ | PVDD18 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 5 |  | mA |
| IPLLVDD18X75RS DS | PLLVDD18 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 2.5 |  | mA |
| $\mathrm{I}_{\text {DVDD } 33 X 75 R S D S}$ | DVDD33 Supply Current, (XGA75Hz input and XGA RSDS panel) |  | 2 |  | mA |
| $\mathrm{I}_{\text {AVDDSX75LVDS }}$ | AVDD Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 225 |  | mA |
| $\mathrm{I}_{\text {DVDD18SX75LVDS }}$ | DVDD18 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 260 |  | mA |
| IXVDD18SX75LVDS | XVDD18 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 2.5 |  | mA |
| lovDD18SX75LVDS | OVDD18 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 70 |  | mA |
| IPVDD18SX75LVDS | PVDD18 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 5 |  | mA |
| IPLLVDD18SX75LV DS | PLLVDD18 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 2.5 |  | mA |
| $\mathrm{I}_{\text {DVDD33SX75LVDS }}$ | DVDD33 Supply Current, (SXGA75Hz input and SXGA LVDS panel) |  | 3.5 |  | mA |
| $\mathrm{I}_{\text {AVDDSX75RSDS }}$ | AVDD Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 225 |  | mA |
| $\mathrm{I}_{\text {DVDD18SX75RSDS }}$ | DVDD18 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 250 |  | mA |
| IXVDD18SX75RSDS | XVDD18 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 2.5 |  | mA |
| IovDD18SX75RSDS | OVDD18 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 20 |  | mA |
| IPVDD18SX75RSDS | PVDD18 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 5 |  | mA |
| IPLLVDD18SX75R SDS | PLLVDD18 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 2.5 |  | mA |
| $\mathrm{I}_{\text {DVDD } 33 S X 75 R S D S}$ | DVDD33 Supply Current, (SXGA75Hz input and SXGA RSDS panel) |  | 3.5 |  | mA |

### 5.3 Preliminary Thermal Data

| Symbol | Parameter | Min. | Typ. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{R}_{\text {thJA }}$ | Junction-to-Ambient Thermal Resistance (LQFP100): <br> Soldered exposed pad <br> Unsoldered exposed pad |  |  | $\begin{aligned} & 20 \\ & 29 \end{aligned}$ | 으/W |
| $\mathrm{R}_{\text {thJA }}$ | Junction-to-Ambient Thermal Resistance (LQFP128): <br> Soldered exposed pad Unsoldered exposed pad |  |  | $\begin{aligned} & 25.1 \\ & 32.6 \end{aligned}$ | 으/W |

### 5.4 Preliminary DC Specifications

Test Conditions: DVDD33 = 3.3V, DVDD18 = AVDD $=$ OVDD18 $=$ PVDD18 $=$ XVDD18 $=$ PLLVDD18 $=1.8 \mathrm{~V}$ and $\mathrm{T}_{\mathrm{AMB}}=25^{\circ} \mathrm{C}$

### 5.4.1 LVTTL 5-Volt Tolerant Inputs with Hysteresis

HSYNC, VSYNC, SCL, RESETN, EXT_SOG, RESETN2, XCLKEN

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{V}_{\mathrm{IH}}$ | High Level Input Voltage |  | 2.0 |  |  | V |
| $\mathrm{~V}_{\mathrm{IL}}$ | Low Level Input Voltage |  |  |  | 0.8 | V |
| $\mathrm{~V}_{\text {HYST }}$ | Schmitt Trigger Hysteresis |  | 0.4 |  |  | V |

### 5.4.2 LVTTL 3-Volt Tolerant Inputs with Hysteresis

TST_SCAN

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{V}_{\mathrm{IH}}$ | High Level Input Voltage |  | 2.0 |  |  | V |
| $\mathrm{~V}_{\mathrm{IL}}$ | Low Level Input Voltage |  |  |  | 0.8 | V |
| $\mathrm{~V}_{\text {HYST }}$ | Schmitt Trigger Hysteresis |  | 0.4 |  |  | V |

### 5.4.3 LVTTL 5-Volt Tolerant I/O with Hysteresis

SDA, SDA2

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{V}_{\mathrm{IH}}$ | High Level Input Voltage |  | 2.0 |  |  | V |
| $\mathrm{~V}_{\mathrm{IL}}$ | Low Level Input Voltage |  |  |  | 0.8 | V |
| $\mathrm{~V}_{\mathrm{HYST}}$ | Schmitt Trigger Hysteresis |  | 0.4 |  |  | V |
| $\mathrm{~V}_{\mathrm{OH}}$ | High Level Output Voltage |  | 3.15 |  |  | V |
| $\mathrm{~V}_{\mathrm{OL}}$ | Low Level Output Voltage |  |  |  | 0.15 | V |
| $\mathrm{I}_{\text {OUT }}$ | Output Current |  |  |  | 8 | mA |

### 5.4.4 LVTTL 3-Volt Tolerant I/O

XCLK

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{V}_{\mathrm{IH}}$ | High Level Input Voltage |  | 2.0 |  |  | V |
| $\mathrm{~V}_{\mathrm{IL}}$ | Low Level Input Voltage |  |  |  | 0.8 | V |
|  |  |  |  |  |  |  |
| $\mathrm{~V}_{\mathrm{OH}}$ | High Level Output Voltage |  | 3.15 |  |  | V |
| $\mathrm{~V}_{\mathrm{OL}}$ | Low Level Output Voltage |  |  |  | 0.15 | V |
| $\mathrm{I}_{\text {OUT }}$ | Output Current |  |  |  | 8 | mA |

### 5.4.5 LVTTL 3-Volt Tolerant I/O

TCON [7:0]

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{V}_{\mathrm{IH}}$ | High Level Input Voltage |  | 2.0 |  |  | V |
| $\mathrm{~V}_{\mathrm{IL}}$ | Low Level Input Voltage |  |  |  | 0.8 | V |
|  |  |  |  |  |  |  |
| $\mathrm{~V}_{\mathrm{OH}}$ | High Level Output Voltage |  | 3.15 |  |  | V |
| $\mathrm{~V}_{\mathrm{OL}}$ | Low Level Output Voltage |  |  |  | 0.15 | V |
| $\mathrm{I}_{\text {OUT }}$ | Output Current |  |  |  | 2 | mA |

### 5.5 LVDS Outputs

OUT [7:0], OUT [7:0]b, OUTCLK [1:0], OUTCLK [1:0]b

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $V_{\text {LVDS-DIFF }}$ | LVDS Differential Output Voltages | $R_{\text {L }}=100$ ohm | 250 | 345 | 450 | mV |
| $V_{\text {LVDS-CM }}$ | LVDS Common Mode Output Voltage | $\mathrm{R}_{\mathrm{L}}=100$ ohm | 1.125 | 1.25 | 1.375 | V |
| $\Delta_{\text {LVDS-DIFF }}$ | Change in $V_{\text {LVDS-DIFF }}$ between complimentary output <br> states | $\mathrm{R}_{\mathrm{L}}=100$ ohm |  |  | 35 | mV |
| $\Delta_{\text {LVDS-CM }}$ | Change in $V_{\text {LVDS-CM }}$ between complimentary output <br> states | $\mathrm{R}_{\mathrm{L}}=100$ ohm |  |  | 35 | mV |

### 5.6 RSDS Outputs

RSDS [7:0], RSDS [7:0]b, OUT [7:0], OUT [7:0]b, OUTCLK [1:0], OUTCLK [1:0]b, RSDS [15:8], RSDS [15:8]b

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :---: | :--- | :--- | :---: | :---: | :---: | :---: |
| V RSDS-DIFF $^{\text {R }}$RSDS Differential Output <br> Voltage | RSDS mode | 100 | 200 | 400 | mV |  |


| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :---: | :--- | :--- | :---: | :---: | :---: | :---: |
| V RSDS-CM | RSDS Common Mode Output <br> Voltage | 680 ohm +50 ohm external <br> termination to 1.3 V | 1.1 | 1.3 | 1.5 | V |
| Trise, Tfall | RSDS Transition Time to $90 \%$ | $\mathrm{C}_{\mathrm{L}}=30 \mathrm{pF}$ |  |  | 3 | ns |

### 5.7 ADC Characteristics

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| INL | ADC Integral Nonlinearity (7-bit) | Without Dithering | 0.6 | 1.0 | 1.5 | LSB |
| DNL | ADC Differential Nonlinearity (7bit) | Without Dithering No missing codes | 0.3 | 0.6 | 1.0 | LSB |
| $\Sigma_{\text {NOB }}$ | Effective Number of Bits | $\begin{aligned} & \mathrm{V}_{\text {INADC }}=1 \mathrm{MHz} \\ & \text { sinusoidal, } 0.5 \mathrm{~V}_{\mathrm{PP}}-1 \mathrm{~V}_{\mathrm{PR}} \\ & \mathrm{~F}_{\text {SAMPLING }}=20 \mathrm{MHz} \end{aligned}$ | 6.2 | 6.6 | 6.8 | Bit |
| THD | Total Harmonic Distortion | $\begin{aligned} & \hline \mathrm{V}_{\text {INADC }}=1 \mathrm{MHz} \\ & \text { sinusoidal, } 0.5 \mathrm{~V}_{\mathrm{PP}}-1 \mathrm{~V}_{\mathrm{PP}} \\ & \mathrm{~F}_{\text {SAMPLING }}=20 \mathrm{MHz} \end{aligned}$ | -48 | -44 | -41 | dB |
| $\mathrm{V}_{\text {INADC }}$ | ADC Input Voltage Range |  | 0.5 |  | 1 | Vp-p |
| $\mathrm{R}_{\text {INADC }}$ | ADC Input Resistance |  |  | 200 |  | Kohms |
| $\mathrm{C}_{\text {INADC }}$ | ADC Input Capacitance |  |  | 12 |  | pF |
| $\mathrm{F}_{\text {ADC }}$ | ADC Sample Frequency |  | 20 |  | 140 | MHz |
| ADC gain step | ADC Gain Step Size |  |  | 0.05 |  | dB |
| ADC offset step | ADC Offset Step Size |  |  | 2.9 |  | mV |

## 6 Package Mechanical Data

### 6.1 100 Pin LQFP



Note: $\quad$ The Pin 1 corner is at an angle, while the others are $90^{\circ}$.

|  | Dimensions (mm) |  |  | Dimensions (inches) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Min. | Typ. | Max. | Min. | Typ. | Max. |
| A |  |  | 1.60 |  |  | 0.063 |
| A1 | 0.05 |  | 0.15 | 0.002 |  | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| b | 0.17 | 0.22 | 0.27 | 0.007 | 0.009 | 0.011 |
| c | 0.09 |  | 0.20 | 0.004 |  | 0.008 |
| D | 15.80 | 16.00 | 16.20 | 0.622 | 0.63 | 0.638 |
| D1 | 13.80 | 14.00 | 14.20 | 0.543 | 0.551 | 0.559 |
| D2 | 2.00 | 3.9 |  | 0.079 | 0.154 |  |
| D3 |  | 12.00 |  |  | 0.472 |  |
| E | 15.80 | 16.00 | 16.20 | 0.622 | 0.63 | 0.638 |
| E1 | 13.80 | 14.00 | 14.20 | 0.543 | 0.551 | 0.559 |
| E2 | 2.00 | 3.9 |  | 0.079 | 0.154 |  |
| E3 |  | 12.00 |  |  | 0.472 |  |
| e |  | 0.50 |  |  | 0.02 |  |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.03 |


|  | Dimensions (mm) |  |  | Dimensions (inches) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Min. | Typ. | Max. | Min. | Typ. | Max. |
|  |  | 1.00 |  |  | 0.039 |  |
| K | 0 | 3.5 | 7 | 0 | 0.138 | 0.276 |
| Ccc |  |  | 0.08 |  |  | 0.003 |

### 6.2 128 Pin LQFP



Note: The Pin 1 corner is at an angle, while the others are $90^{\circ}$.

|  | Dimensions (mm) |  | Dimensions (inches) |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Min. | Typ. | Max. | Min. | Typ. | Max. |
| A |  |  | 1.600 |  |  | 0.063 |
| A1 |  |  | 0.150 |  |  | 0.006 |
| A2 | 1.400 | 1.350 | 1.450 | 0.055 | 0.053 | 0.057 |
| b | 0.220 | 0.170 | 0.270 | 0.009 | 0.007 | 0.011 |
| D | 22.000 |  |  | 0.866 |  |  |
| D1 | 20.000 |  |  | 0.787 |  |  |
| D2 |  | 3.9 |  |  | 0.154 |  |


|  | Dimensions (mm) |  | Dimensions (inches) |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Min. | Typ. | Max. | Min. | Typ. | Max. |
| E | 16.000 |  |  | 0.623 |  |  |
| E1 | 14.000 |  |  | 0.551 |  |  |
| E2 |  | 3.9 |  |  | 0.154 |  |
| E | 0.500 |  |  | 0.020 |  | 0.030 |
| L | 0.600 | 0.450 | 0.750 | 0.024 | 0.0178 |  |
| L1 | 1.000 |  |  | 0.040 |  | 0.000 |
| K |  | 0.000 | 7.000 |  | 0.275 |  |

## 7 Scaler Equations

| Dclk_PR = FSYN_PR_OTCLK $\cdot(3-$ NumPPC $)$ | $f_{\text {Sclk }}=140 \mathrm{MHz}$, and Sclk_PR $=18 \mathrm{AF}$ |
| :---: | :---: |
| $\text { dest }_{-} \text {hpos }=\frac{H P_{\text {out }}-\text { dest_hpixel }^{2}}{2}$ | $\text { dest }_{-} v p o s=\frac{V P_{\text {out }}-\text { dest__ }^{\text {vpixel }}}{2}$ |
| WinTop $=10$ | $\text { WinBot }=\left\{\begin{array}{l} D E_{-} \text {panels } \Rightarrow 2, \\ \text { WinTop }+V P_{\text {out }}-1, \text { otherwise } \end{array}\right.$ |
| WinLeft $=\max \left(48, H T_{\text {out }}-\min H B_{\text {out }}-8\right)$ | WinRight $=$ WinLeft + HP ${ }_{\text {out }}$ |
| $\text { ScaleFactorH }=\frac{H P_{\text {in }} \ll 16}{\text { dest_hpixel }}$ | $\text { ScaleFactor } V=\frac{V P_{i n} \ll 16}{\text { dest_vpixel }}$ |
| $\text { pipe _rate }=\frac{\text { Sclk_PR } \cdot \text { ScaleFactorH }}{\text { Dclk_PR >> } 2}$ |  |
| origin_hpos $=-((($ WinLeft + dest_hpos $) \cdot 2-23) \cdot$ ScaleFactorH +24$) \gg 13$ |  |
| origin_vpos $=-(($ WinTop + dest_vpos $) \cdot 2-1) \cdot$ ScaleFactorV $) \gg 13$ |  |
| $\text { LineStart }=(\text { WinLeft }+ \text { dest_hpos }) \ll 4-\frac{(43 \ll 9}{D c}$ | $\left.\frac{\text { Sclk__ }_{-} P R}{P R}-\frac{1 \ll 21}{\text { ScaleFactorH }}-136\right) \gg 4$ |
| $\begin{array}{r} \text { InitTrigDelay }=\frac{\left(\left(\left(H P_{\text {in }}>1024\right) ? 5: 9\right)+\right.\text { SMUX_VT }}{S M E A S \_H \_N U M} \\ \left(( \text { WinTop } + \text { dest_vpos } ) \cdot H T _ { \text { out } } \cdot \left(\text { Dclk_P }^{\prime} P R\right.\right. \end{array}$ | $\frac{I G \ll 2) \cdot x c l k s \_p e r \_h s y n c}{}-$ <br> 6)) >> 15 |
| SCL_TRIGGER_DLY = InitTrigDelay;// wait 3 frames wPtrMinPost = SCL_PTR_POST; <br> // feedback loop to tune trigger delay: <br> while ( ! ( 8 < wPtrMinPost < 16) ) \{// landing zone is ( 8 , wDiff = wPtrMinPost > (512 + 12) ? wPtrMinPost - (1024 dwTriggerDelay = dwTriggerDelay + ((( (xclk_freq / inc SCL_TRIGGER_DLY = max (dwTriggerDelay, 1); // wait 2 frames for new trigger delay to take effect wPtrMinPost = SCL_PTR_POST; \} | \& mid point is 12 <br> + 12): wPtrMinPost - 12; _freq) * 3 ) * wDiff) / 2); |

## 8 ADE3800 vs ADE3700

The following gives an overview of the main differences between ADE3700 and ADE3800:

## Package

- 100 pin LQFP for LVDS $1 \& 2$ channels and RSDS 1 channel application
- 128 pin LQFP for RSDS 2 pixel per clock support
- 5 V tolerant inputs do not have internal pull-up resistors


## I2C Interface

- Add Block Transfer for fast internal data move/swap/copy


## Registers

- Now all registers runs on XCLK
- RGB register address ordering is reversed to BGR


## Analog Front End (ADC \& SOG)

- New ADC design with higher performances
- Add Internal SOG Sync Stripper with bypass option (external SOG TTL pin)
- SOG activity can operate while ADC Power is down (wakeup from DPMS by SOG support)
- Per channel skew control
- Analog Filter bandwidth programmable
- Gain and Offset independent and linear
- 10-bit ADC using Analog Dithering Technique (ADTH)


## Line Lock PLL (LLK)

- Synthesized Internal HSync has 50\% duty cycle
- Phase step is 4 times more precise
- Phase range can exceed one clock period delay
- Lock filter removed
- Fewer registers, simplifies some programming
- Clock and Phase are both shadowed
- FM Modulation amplitude step is 16 times more precise


## Sync Measurement (SMEAS)

- Remove out of range register
- Add Fast Mute function
- Group all the fast mute flags in SMEAS with sticky bit and enable


## Sync Mux (SMUX)

- Set/Reset replaced by Pos/Size references
- Add Vtrigger to make framelock reference in the center of the frame, allowing easier artifact free implementation on wide picture position changes


## Data Measurement (DMEAS)

- Scratch pad register removed
- DMEAS uses only the 7 MSB Color data information for processing


## Scaler Zoom (SCALE)

- Simpler kernel programming, fewer registers
- New context sensitive scaler function
- H \& V Sharpness control
- Includes former OSEQ functionality
- TCON generates the panel output and reference internal signals


## Pattern Generator (PGEN)

- Only one pattern engine (P0); P1 is input video
- TCON Windowing control added


## Color Transformations (SRGB)

- $2 x$ and $4 x$ Delta magnification options added
- RGB programmable max clipping function added


## Gamma Correction (GAM)

- 10 bit input and 10 bit output with bypass option
- Relative 8 bit 2 s -complement value delta tables (3x256)
- $2 x$ Delta magnification option, with limiting
- RGB Offset within an input range option available


## On Screen Display (OSD)

- New concept
- Support 1,2,3,4 bpp characters
- 64 True color palette with 4 bit alpha blending per color
- Common RAM shared between text and font
- Text can be displayed anywhere on the screen
- OSD Position is in pixel and line unit
- Per character H and V flip
- Up to 1024 character support


## Flicker Detector (FLICKER)

- Only 4 of the 8 scores are measured in one shot
- Output Dithering (APC)
- New design
- Mode flexibility and performance improved
- LSB justified (for 6 bit output, MSBs are zero)


## Output Mux (OMUX)

- Per Pin Delay removed
- Gate Speed monitoring register removed
- Bit rotate function added


## Timing Controller (TCON)

- Simplified, easier programming
- Common functions hardcoded
- Easy to use LC Polarity Inversion signal generator
- 16 SRTD gates vs 32 in ADE 3700
- Scaler Output Data Enable signal must come from TCON (no OSEQ block anymore)
- No OCF control anymore
- Resync on H or V Sync edges no longer needed
- Comparator, Pulse and Window generators


## LVDS/RSDS (new block added)

- Skew control
- $40-85 \mathrm{MHz}$ clock range
- Flexible output mapping
- Pair swap, bit sequence reversing option


## 9 Glossary

| AFE | Analog Front End, this includes the ADC and SOG circuitries |
| :---: | :---: |
| Bare Panel | see Smart Panel |
| bpp | bit per pixels (OSD Font: 4bpp = 16 color characters) |
| DFT | design For Test block to output certain internal signals (otherwise not available) |
| in_enab | in_henab \& in_venab = Input active area signal* |
| in_henab | input horizontal active pixel signal* |
| in_venab | input vertical active line signal* |
| LVDS | low voltage differential signaling video interface to LCD panel |
| out_enab | out_henab \& out_venab = Output panel active area signal* |
| out_henab | output panel horizontal active pixel signal* |
| out_venab | output panel vertical active line signal* |
| ppc | pixels per clock ( $2 \mathrm{ppc}=$ dual wide panel bus interface) |
| PVT | parameters that depend on Process (chip), Voltage (power) and Temperature (board) |
| RSDS | reduced swing differential signaling video interface to LCD panel |
| SIP Panel | see Smart Panel |
| Smart Panel | panel without built-in TCON using TTL or RSDS input video interface, additional timing signals must be provided for proper operation. |
| SOG | sync on Green type signal |
| sRGB | standard RGB, color matching between display and real life |
| SRTD | Set-Reset-Toggle-Delay programmable gate in TCON |
| Standard Pan | I panel with built-in TCON using LVDS or TTL input video interface |
| TCON | timing controller function |
| TMDS | transition minimized differential signaling video interface from DVI digital video input |

[^0]
## 10 Revision History

Table 46: Summary of Modifications

| Date | Version | Description |
| :---: | :---: | :--- |
| 14 February 2003 | 0.1 | First Draft. |
| 03 June 2003 | 0.2 | Second Draft. |
| 05 September 2003 | 0.3 | Major updates to Chapter 3: Pin Descriptions, Chapter 4: Register Description by Block and <br> Chapter 6: Package Mechanical Data. |
| 24 October 2003 | 0.4 | Major updates to all chapters. |
| November 2003 | 1.0 | First Issue. |
| January 2004 | 1.1 | Document changed from target specification to datasheet. |
| April 2004 | 1.3 | Corrections to Figure 2: LQFP100 Pinout Diagram on page 11 and Figure 3: LQFP128 Pinout <br> Diagram on page 12 and to pin numbers in table on Chapter 4.17.1: Output Data. |
| May 2004 | 2.0 | Major updates to Chapter 3: Pin Descriptions. New pin assignments impact on <br> Figure 2: LQFP100 Pinout Diagram on page 11, Figure 3: LQFP128 Pinout Diagram on page 12, <br> Table 3: Analog Input Signals on page 12 and Table 7: Analog Section Power Supply Pins on <br> page 16. |
| June 2004 | 2.1 | Corrected TCON_INV_1 descriptions in Table 39: Register Map. Major updates to <br> Chapter 5: Electrical Specifications. |
| 09 November 2004 | 3.0 | Chapter 3: Pin Descriptions - Replaced pin 1 (QFP100) and pin 128 (QFP 128) name with <br> PLLVDD18. <br> Chapter 4: Register Description by Block - Few changes applied in register description for GLBL, <br> OMUX, SMUX and TCON. <br> Chapter 5: Electrical Specifications - Many values modified following the device full <br> characterization. |
| February 2005 | 3.1 | Updated information in Table 7: Analog Section Power Supply Pins on page 16, <br> Chapter 4.12.1: Parametric Gamma, Digital Contrast / Brightness on Multiple Windows and <br> Chapter 4.12.2: Color Space Warp. |
| 11 Apr 2005 | 3.2 | Updated Chip Revision ID and added GLBL_AZWC_CTRL register values in Table 9: Global <br> Control Registers on page 18. |

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.
All other names are the property of their respective owners
© 2005 STMicroelectronics. All Rights Reserved.
STMicroelectronics Group of Companies
Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

000 «ниоКРсистемс» - это оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов. Реализуемая нашей компанией продукция насчитывает более полумиллиона наименований.

Благодаря этому наша компания предлагает к поставке практически не ограниченный ассортимент компонентов как оптовыми, мелкооптовыми партиями, так и в розницу.

Благодаря развитой сети поставщиков, помогаем в поиске и приобретении экзотичных или снятых с производства компонентов.

Наша компания это:

- Гарантия качества поставляемой продукции
- Широкий ассортимент
- Минимальные сроки поставок
- Техническая поддержка
- Подбор комплектации
- Индивидуальный подход
- Гибкое ценообразование
- Работаем по 275 ФЗ


[^0]:    * All enab type signals are active high

