FOCUS enhancements
superior video technology

# FS401, FS403 PC to TV Video Scan Converters

## Features

- Frame rate Conversion †
- Programmable 2D scaling †
- Pan and Zoom †
- Advanced 2-D flicker filter †
- Frame-store memory controller
- Supports Multiple Progressive Input Resolutions †

–Minimum 640x 400
–Maximum 2048 x 1536

- Supports Interlaced Input

–XGA and above

- Input refresh rates up to 150Hz
- Multiple Output Standards

–NTSC, NTSC-EIAJ, PAL-B/G/H/I †
–Composite, S-Video, SCART
–RGB, YUV

▪Standard NTSC and PAL
▪Super NTSC & PAL
▪VGA Progressive
▪SVGA Progressive
▪NTSC Progressive
▪PAL 100Hz Interlaced

- Automatically detects input active video area
- Automatically selects the best output and scaling for any input resolution
- Programmable sharpness, brightness, contrast and color saturation
- Customizable On Screen Display via glueless integration with Zilog and Philips OSD Microprocessors (FS403)
- C, H, and V Sync tri-state outputs
- H and V Sync monitoring for DPMS Support
- Exceeds all PC97 and PC98 requirements
- General Purpose Output Pins (2 on FS401, 7 on FS403)
- Genlock (FS403)
- 8-bit A/D converters with frequency adaptive input filtering support
- 10-bit output D/A converters
- Digital RGB Inputs (FS403)
- $I^2C^‡$ compatible port controls (SIO)
- 100 pin PQFP (FS401)
- 128 pin PQFP (FS403)
- 3.3V operation
- RoHS Compliant

†Note: Covered under US Patent # 5,862,268, # 5,905,536, # 5,966,184 and/or patents pending.

## Description

The FS400 family is a fourth generation video scan converter. It accepts many input resolutions and rates and converts them to NTSC or PAL standards compliant with SMPTE-170M and CCIR-656 standards. Also available as output options are VGA 640 x 480 at 60Hz progressive, SVGA 800 x 600 at 60Hz progressive, and 100 Hz interlaced. The chip has a programmable down scaler to fit the incoming resolution to the output display format. Within the FS400 are capture and encoder engines separated by the frame buffer memory controller. Required external components are minimal: a single 16M SDRAM memory, clocks and passive parts.

Analog progressive RGB inputs are digitized and converted to the YUV 4:2:2 format. Vertical scaling and flicker filtering are implemented at the computer frame rate ahead of the frame store interface.

Interlaced input is supported for XGA resolution and above. In this mode, only the first field is processed.

The Flicker Filter is an advanced 2 dimensional filter that enhances text quality. Flicker Filter parameters are programmable to allow user tradeoffs between flicker and sharpness.

The FS400 family contains controls for programmable sharpness, brightness, contrast, and color saturation. These controls allow output to be tuned to match user desires and tastes.

Frame rate conversion is implemented by a Frame Store Controller that interfaces with an external SDRAM frame store memory.

YUV 4:2:2 data is recovered from the memory at the outgoing frame rate. Data is scaled prior to the

digital video encoder that generates Y/C and Composite Video outputs. For RGB and YUV outputs, the encoder may be bypassed via a YUV to RGB transcoder for SCART compatible video, and for output to VGA or SVGA displays.

The FS400 has built in capability to automatically detect the incoming video mode and automatically select optimal sampling and scaling parameters. The chip can detect the location of the active video in the input, and can automatically center the input on the TV screen. All parameters can be read and written via the SIO serial port.

The FS403 has support for glueless integration with Zilog and Philips On Screen Display (OSD) microcontrollers. The OSD interface allows a customized on screen user interface that can contain opaque or halftone video backgrounds.

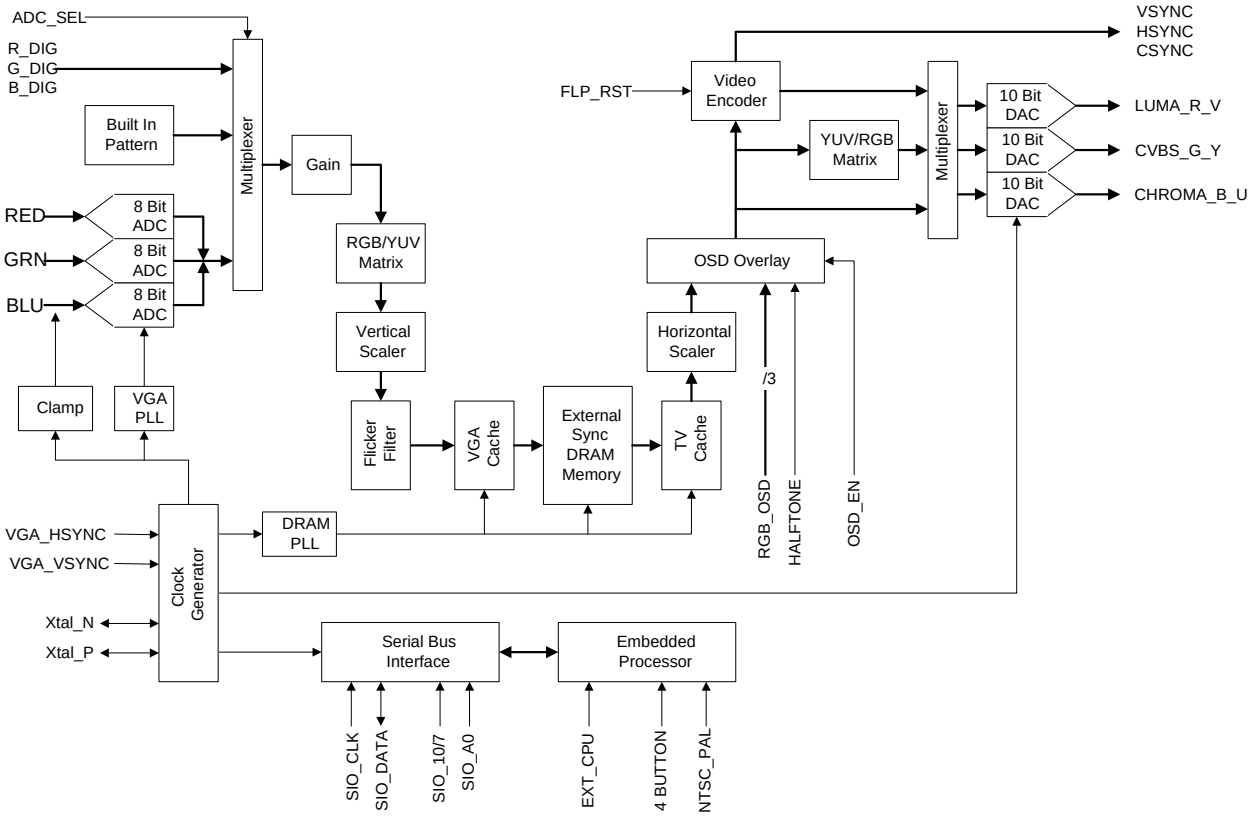
The FS403 has direct digital inputs, bypassing the built-in ADCs.

Power is derived from +3.3V digital and analog supplies. Packages are 100-lead (FS401) or 128-lead (FS403) Plastic Quad Flat Pack (PQFP).

## Applications

- PC video out
- PC ready TV's
- Video Text Displays
- Web Appliances
- PC-to-TV Scan Converter Peripherals
- Video Kiosks

# Architectural Block Diagram



## Contents

## Tables



## Figures

# 1. Architectural Overview

Overall design principles are included in this section. Details of how to use and setup the FS400 are included in the *Functional Description* section, starting on *page 75*.

RGB video inputs are asynchronously converted to either NTSC/PAL, YUV or RGB video formats. Architecturally, the FS400 is divided into five major sections:

1. Video Capture Engine
2. Clock Processor
3. Frame Store Controller
4. Video Encoder Engine
5. Serial Bus Interface

Besides power and a few external passive components, the FS400 requires only a single 16M external SDRAM and external clocks to implement a high quality video scan converter.

Either analog or digital inputs (FS403 only) with separate horizontal and vertical sync signals are accepted. Analog VGA video must be RGB. Digital video (FS403 only) must be 24-bit RGB clocked by external clock, VGACK_IN.

A wide range of resolution formats can be accepted, including common standards such as 320x240, 640x400, 720x400, 640x480, 800x600, 832x624, 1024x768, 1152x864, 1280x1024, and 1600x1200. Incoming RGB signals are converted to either the NTSC or PAL TV Standards, 100Hz PAL, or progressive scan VGA, SVGA, or NTSC. Output video format can be selected to be either composite and Y/C (NTSC and PAL only), or RGB or YUV (all standards).

Incoming frame rate may range to over 150 Hz according to the table below. The Video Capture engine runs asynchronously relative to the Video Encoder Engine. An external frame store memory separates the two engines with write and read access controlled by the FS400.

Transformation operations include overscan, underscan, pan and zoom. Scaling operations are separated by the frame store with vertical down-sampling incorporated into the Capture Engine and horizontal up-sampling incorporated into the Encoder Engine.

## 1.1 Video Capture Engine

Triple 8-bit A/D converters digitize the analog RGB inputs at rates of up to 50 MHz. Internal A/D sample clock, ADCK is derived from a phase locked loop referenced to the leading edge of horizontal sync. Either positive or negative sync polarity is accepted.

The selected input (A/D converter outputs or digital RGB) is transcoded by the color matrix into a 16-bit $YC_RC_B$ 4:2:2 format. A vertical scaler filters the number of incoming video lines by the selected scaling factor. A flicker filter averages lines to eliminate flicker between lines or boundaries.

The Video Capture Engine is programmable as to the number of horizontal samples it takes. The limiting factor in the sample rate is the A/D Converters. By programming fewer samples per line, higher incoming data rates can be accommodated. The following table illustrates the capability:

| **Active Samples** | **720 CCIR 601** | **640** | **500** |
|---|---|---|---|
| **Maximum Line Frequency** | 56kHz | 63kHz | 80kHz |
| **640 x 480** | 106Hz | 119Hz | 152Hz |
| **800 x 600** | 89Hz | 100Hz | 128Hz |
| **1024 x 768** | 70Hz | 78Hz | 100Hz |
| **1152 x 864** | 59Hz | 66Hz | 85Hz |
| **1280 x 1024** | 53Hz | 59Hz | 76Hz |
| **1600 x 1200** | 44Hz | 50Hz | 64Hz |

## 1.2 Frame Store Memory Controller

Inserted between the capture and the encoder engines the frame store has two functions: 1) to act as a reservoir of pixels to match the incoming frame rate to the outgoing field or frame rate; 2) to support vertical scaling by allowing lines to be written into the frame store intermittently, but read out at a constant rate.

Frame store clock, FS_CK is derived from the OSC1 clock by a second phase locked loop.

## 1.3 Video Encoder Engine

Pixels are retrieved from the external frame store memory asynchronously relative to the incoming frames. Outgoing video timing is set to the selected TV (NTSC or PAL) or progressive scan standard.

Incoming data sampling is normally set to fill complete lines in the Frame Store Memory. Horizontal scaling is applied to pixels exiting the Frame Store. Pixels may be routed through either a digital video encoder or a $YC_RC_B$-to-RGB transformation matrix. Either output is connected to a triple 10-bit D/A converter to generate the video output that may be Composite Video and Y/C, RGB or YUV.

Encoder Engine timing is derived from many clock sources. These standards are shown below.

| Encoder Standard | Total Pixels | Total Lines | Vertical Mode | Clock Rate | Line Freq. |
|---|---|---|---|---|---|
| NTSC | 910 | 525 | 60I | 14.318 MHz | 15.73 KHz |
| PAL | 1135.0064 | 625 | 50I | 17.734 MHz | 15.62 KHz |
| RGB NTSC | 910 | 525 | 60I | 14.318 MHz | 15.73 KHz |
| RGB PAL | 1135 | 625 | 50I | 17.734 MHz | 15.62 KHz |
| Super RGB NTSC | 1280 | 525 | 60I | 20.140 MHz | 15.73 KHz |
| Super RGB PAL | 1280 | 625 | 50I | 20.000 MHz | 15.62 KHz |
| VGA | 800 | 525 | 60P | 25.175 MHz | 31.5 KHz |
| SVGA | 1024 | 625 | 60P | 38.400 MHz | 37.5 KHz |
| Progressive NTSC | 910 | 525 | 60P | 28.636 MHz | 31.5 KHz |
| 100Hz PAL | 1135 | 625 | 100I | 35.468 MHz | 31.25 KHz |

## 1.4 Serial Control Port

FS400 setup is programmed by 39 16-bit registers that are accessible via the $I^2C^{\ddagger}$ compatible serial port (SIO). Status and Revision ID can also be read from the registers.

$^{\ddagger}$Note: $I^2C$ is a registered trademark of Philips Corporation. The FS400 SIO bus is similar but not identical to Philips $I^2C$ bus.

# 1.5 Typical System Configurations

## 1.5.1 External Scan Converter

The FS403 has been optimized for scan converter designs. It provides a maximum amount of flexibility while minimizing system cost. When combined with a Zilog Z902xx Family or Philips P8xC055/145/845 Families, the FS403 provides overlaid On-Screen Display pixels specified by the programmer without any additional external components thus minimizing expense, complexity, and size while maximizing flexibility and features.

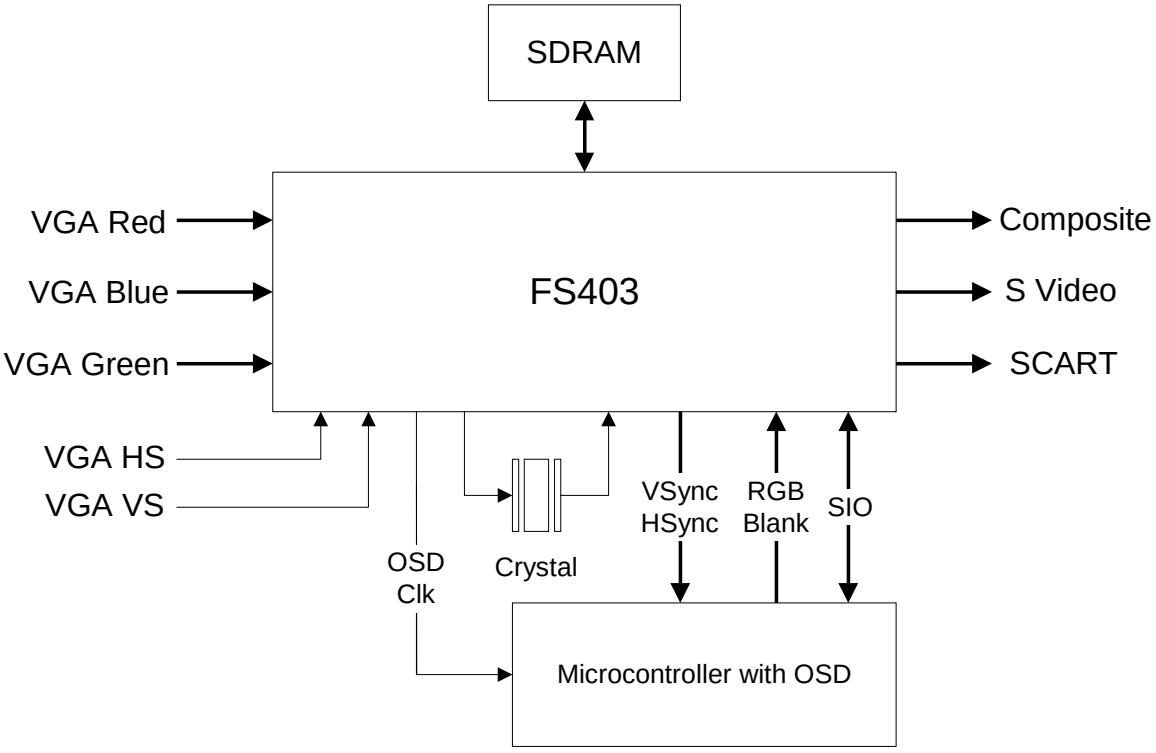



**Figure 1: External Scan Converter Block Diagram**

### 1.5.2 Embedded Television Interface

The FS401 has been optimized for television designs. With its built in microprocessor, the FS401 can run freely with minimal control from the television processor while providing complete plug-n-play capability. Simple commands can be sent to the FS401 via the SIO bus to implement remote control functions such as zoom, pan, sizing, positioning, and video quality control (such as brightness, contrast, saturation, flicker, and sharpness). The only components required are the FS401 and a single SDRAM for a truly minimal incremental system cost.

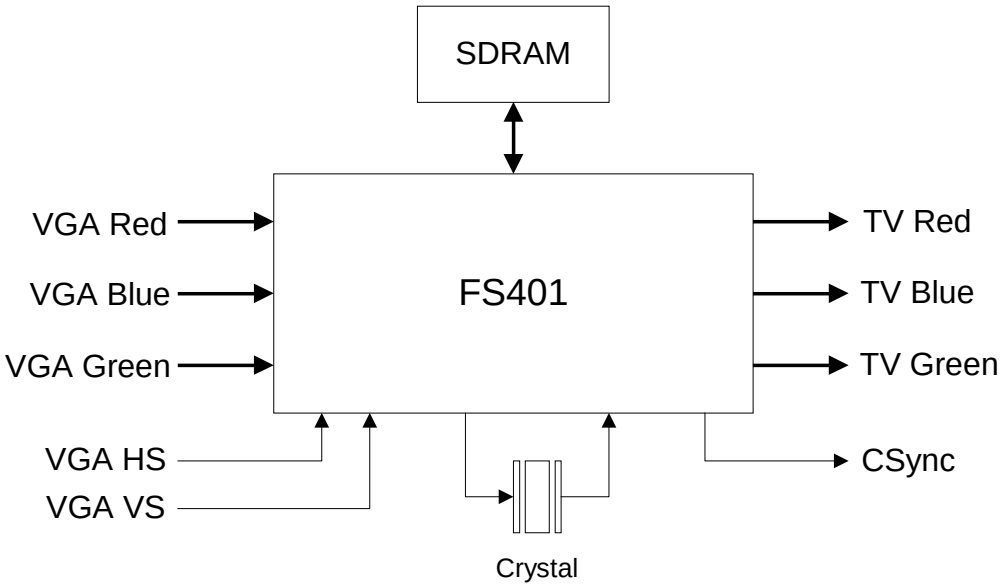



**Figure 2: Embedded Television Design Block Diagram**

### 1.5.3 Deleted

*Figure intentionally left blank.*

**Figure 3: Deleted**

### 1.5.4 Professional and Pro-Consumer Video Designs

In the Professional and Pro-Consumer Video Market, Video quality, video timing accuracy, and Genlock are very important features. Also the system will use external ADCs and PLLs of the highest quality. Genlock to an external studio Black Burst (black screen TV picture) will be used to synchronize the scan converter to the rest of the video studio.

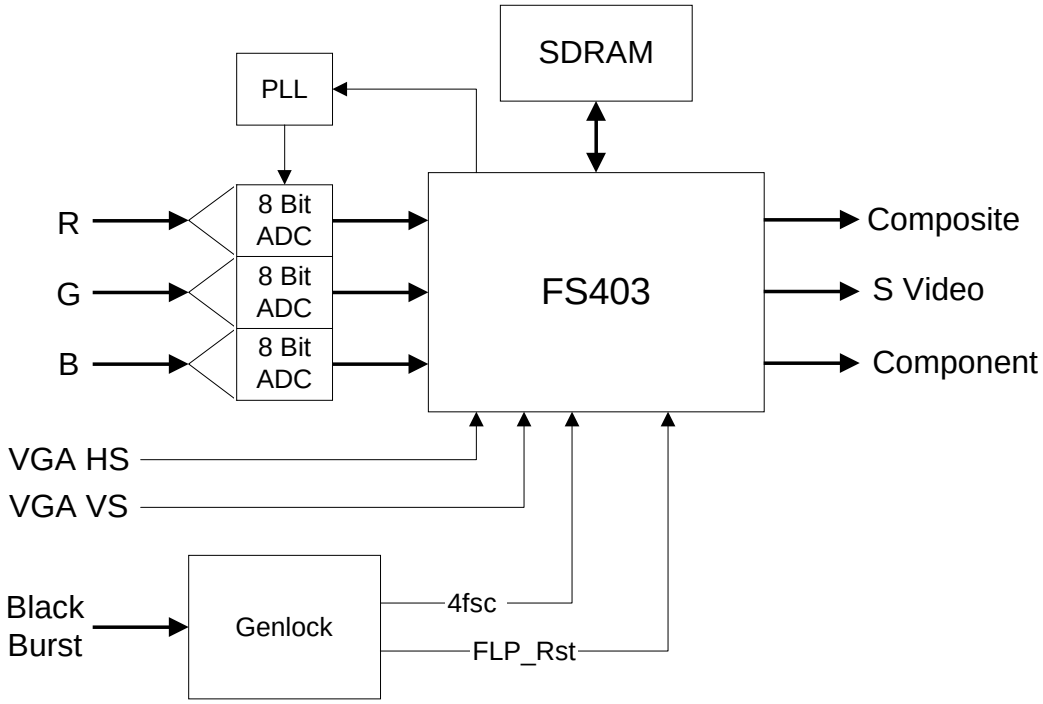
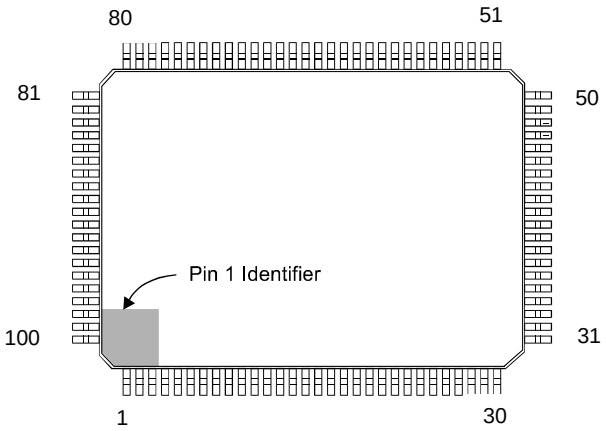



**Figure 4: Professional & Pro-Consumer Video Design Block Diagram**

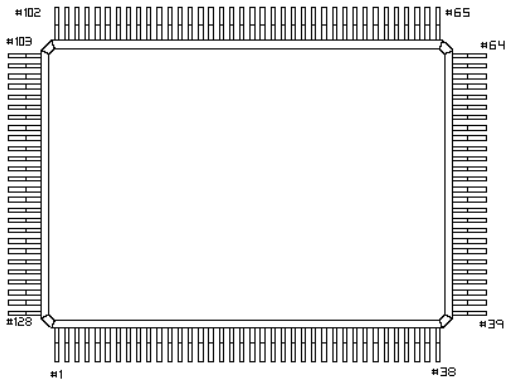



# 2. Pin Assignments

## 2.1 100-Lead PQFP Package (FS401)

**Table 1. Pin Designations (FS401, 100-pin package)**

| Pin | Name | Pin | Name | Pin | Name | Pin | Name |
|---|---|---|---|---|---|---|---|
| 1. | $V_{DDPF}$ | 31. | TV_VSYNC | 51. | $V_{SS}$ | 81. | GPO6 |
| 2. | OSC1 | 32. | TV_HSYNC | 52. | EXTVGASEL\ | 82. | $V_{DD}$ |
| 3. | OSC1BUF | 33. | $V_{DDAD}$ | 53. | VGACLKIN | 83. | A0 |
| 4. | OSC2 | 34. | $V_{SSAD}$ | 54. | $V_{DD}$ | 84. | A1 |
| 5. | OSC2BUF | 35. | $V_{SSAD}$ | 55. | $V_{SS}$ | 85. | A2 |
| 6. | $V_{SS}$ | 36. | R_IN | 56. | $V_{SS}$ | 86. | A3 |
| 7. | $V_{SSDA}$ | 37. | $V_{DDAD}$ | 57. | DQM | 87. | A4 |
| 8. | $C_{COMP}$ | 38. | $V_{DDAD}$ | 58. | WE\ | 88. | A5 |
| 9. | $I_{REF}$ | 39. | G_IN | 59. | INTCPUEN | 89. | A6 |
| 10. | $V_{DDDA}$ | 40. | $V_{SSAD}$ | 60. | RAS\ | 90. | A7 |
| 11. | $C_{BYPASS}$ | 41. | $V_{TOUT}$ | 61. | CAS\ | 91. | A8 |
| 12. | Y/R/V | 42. | $V_{ADCREF}$ | 62. | D0 | 92. | A9 |
| 13. | $V_{DDDA}$ | 43. | $V_{SSAD}$ | 63. | D1 | 93. | A10 |
| 14. | $V_{SSDA}$ | 44. | B_IN | 64. | D2 | 94. | A11 |
| 15. | CVBS/G/Y | 45. | $V_{DDAD}$ | 65. | D3 | 95. | $V_{DD}$ |
| 16. | $V_{DDDA}$ | 46. | CLAMP_REF | 66. | D4 | 96. | GPO0 |
| 17. | C/B/U | 47. | $V_{DDPA}$ | 67. | D5 | 97. | $V_{SSPF}$ |
| 18. | $V_{DDDA}$ | 48. | $V_{SSPA}$ | 68. | D6 | 98. | RAMCK_SEL\ |
| 19. | $V_{DD}$ | 49. | VS_IN | 69. | D7 | 99. | RAMCK_OUT |
| 20. | CSYNC | 50. | HS_IN | 70. | D8 | 100. | RAMCK_IN |
| 21. | $V_{SS}$ | | | 71. | D9 | | |
| 22. | $V_{SS}$ | | | 72. | D10 | | |
| 23. | SIOCLK | | | 73. | D11 | | |
| 24. | SIODATA | | | 74. | $V_{DD}$ | | |
| 25. | $SIOA_{10/7}$ | | | 75. | $V_{SS}$ | | |
| 26. | $SIOA_0$ | | | 76. | D12 | | |
| 27. | $V_{DD}$ | | | 77. | D13 | | |
| 28. | Reserved ($V_{SS}$) | | | 78. | D14 | | |
| 29. | Reserved ($V_{SS}$) | | | 79. | D15 | | |
| 30. | RESET | | | 80. | $V_{SS}$ | | |




## 2.2 128-Lead PQFP Package (FS403)

**Table 2. Pin Designations (FS403, 128-pin package)**

| Pin | Name | Pin | Name | Pin | Name | Pin | Name |
|---|---|---|---|---|---|---|---|
| 1. | OSC1 | 33. | $V_{DD}$ | 65. | VGACLKDIV | 97. | D12 |
| 2. | OSC1BUF | 34. | Reserved ($V_{SS}$) | 66. | EXTVGASEL\ | 98. | D13 |
| 3. | OSC2 | 35. | FLP_RST | 67. | VGACLKIN | 99. | D14 |
| 4. | OSC2BUF | 36. | Reserved ($V_{SS}$) | 68. | $V_{DD}$ | 100. | D15 |
| 5. | $V_{SS}$ | 37. | RESET | 69. | $V_{SS}$ | 101. | $V_{SS}$ |
| 6. | $V_{SSDA}$ | 38. | R0/TV_VSYNC | 70. | G3 | 102. | B7/GPO7 |
| 7. | $C_{COMP}$ | 39. | R1/TV_HSYNC | 71. | G2 | 103. | B6/GPO6 |
| 8. | $I_{REF}$ | 40. | R2/OSDEN | 72. | G1 | 104. | B5/GPO5 |
| 9. | $V_{DDDA}$ | 41. | R3 | 73. | G0 | 105. | B4/GPO4 |
| 10. | $C_{BYPASS}$ | 42. | $V_{DDAD}$ | 74. | $V_{SS}$ | 106. | $V_{DD}$ |
| 11. | Y/R/V | 43. | $V_{SSAD}$ | 75. | DQM | 107. | A0 |
| 12. | $V_{DDDA}$ | 44. | $V_{SSAD}$ | 76. | WE\ | 108. | A1 |
| 13. | $V_{SSDA}$ | 45. | R_IN | 77. | INTCPUEN | 109. | A2 |
| 14. | CVBS/G/Y | 46. | $V_{DDAD}$ | 78. | RAS\ | 110. | A3 |
| 15. | $V_{DDDA}$ | 47. | $V_{DDAD}$ | 79. | CAS\ | 111. | A4 |
| 16. | C/B/U | 48. | G_IN | 80. | $V_{DD}$ | 112. | A5 |
| 17. | $V_{DDDA}$ | 49. | $V_{SSAD}$ | 81. | D0 | 113. | A6 |
| 18. | $V_{DD}$ | 50. | $V_{TOUT}$ | 82. | D1 | 114. | A7 |
| 19. | OSDCLK | 51. | $V_{ADCREF}$ | 83. | D2 | 115. | A8 |
| 20. | CSYNC | 52. | $V_{SSAD}$ | 84. | D3 | 116. | A9 |
| 21. | $V_{SS}$ | 53. | B_IN | 85. | D4 | 117. | A10 |
| 22. | $V_{SS}$ | 54. | $V_{DDAD}$ | 86. | D5 | 118. | A11 |
| 23. | EXADSEL | 55. | CLAMP_REF | 87. | D6 | 119. | $V_{DD}$ |
| 24. | G6/OSDB | 56. | R4 | 88. | D7 | 120. | B3/GPO3 |
| 25. | G5/OSDG | 57. | R5 | 89. | $V_{SS}$ | 121. | B2/GPO2 |
| 26. | G4/OSDR | 58. | R6 | 90. | $V_{DD}$ | 122. | B1/GPO1 |
| 27. | G7/OSDHT | 59. | R7 | 91. | D8 | 123. | B0/GPO0 |
| 28. | $V_{SS}$ | 60. | $V_{DDPA}$ | 92. | D9 | 124. | $V_{SSPF}$ |
| 29. | SIOCLK | 61. | $V_{SSPA}$ | 93. | D10 | 125. | RAMCK_SEL\ |
| 30. | SIODATA | 62. | VS_IN | 94. | D11 | 126. | RAMCK_OUT |
| 31. | $SIOA_{10/7}$ | 63. | HS_IN | 95. | $V_{DD}$ | 127. | RAMCK_IN |
| 32. | $SIOA_0$ | 64. | $V_{SS}$ | 96. | $V_{SS}$ | 128. | $V_{DDPF}$ |

# 3. Pin Descriptions

| Pin Name | Pin Number FS401/FS403 | Type/Value | Pin Function Description |
|---|---|---|---|
| **Clocks** | | | |
| OSC1 | 2/1 | TTL input | **Oscillator Number 1.** Default clock input to the encoder. Input from an external oscillator; or one pin of a crystal connected between OSC1 and OSC1BUF. NTSC subcarrier frequency can be derived from this $4f_{SC}$ clock. OSC1 must be connected (see Table 4. Clock Connections). |
| OSC1BUF | 3/2 | LVTTL output | **Oscillator 1 Buffer.** If a crystal is used, one pin is connected to OSC1, the other to OSC1BUF. Float this pin if OSC1 is connected to an oscillator. |
| OSC2 | 4/3 | TTL input | **Oscillator Number 2.** Input from an external oscillator; or one pin of a crystal connected between OSC2 and OSC2BUF. PAL subcarrier frequency can be derived from this $4f_{SC}$ clock. Ground if unused. |
| OSC2BUF | 5/4 | LVTTL output | **Oscillator 2 Buffer.** If a crystal is used, one pin is connected to OSC2, the other to OSC2BUF. Float this pin if unused. |
| HS_IN | 50/63 | TTL input | **VGA Horizontal Sync Input.** Active HIGH or active LOW polarity is sensed. |
| VS_IN | 49/62 | TTL input | **VGA Vertical Sync Input.** Active HIGH or active LOW polarity is sensed. |
| **Global Controls** | | | |
| INTCPUEN | 59/77 | TTL input | **Internal CPU enable.** INTCPUEN enables the internal CPU when high. |
| FLP_RST 403 only | x/35 | TTL input | **Field, Line and Pixel Reset.** Resets video encoder engine to the start location on the outgoing frame. Used for genlock. Active high. |
| RESET | 30/37 | TTL input | **Reset.** Resets internal state machines and initializes default register values. Active high. |
| **A/D Converter Interface** | | | |
| R_IN, G_IN, B_IN | 36/45, 39/48, 44/53 | 700 or 1000 mV | **Analog red, blue and green inputs.** AC coupled RGB video input signals. Nominal voltage range is 0.7 or 1.0 Volt peak-to-peak (selectable). Inputs are clamped to ground when HS_IN is active. |
| $V_{ADCREF}$ | 42/51 | 700 or 1000 mV | **A/D Converter Top Reference Voltage Input.** Input to voltage follower that supplies current to A/D converter reference resistors. Range is 0.5 - 2.0 volts. |
| $V_{TOUT}$ | 41/50 | 700 or 1000 mV | **A/D Converter Top Reference Voltage Output.** Output of the internal $V_T$ buffer and direct connection to the A/D reference resistor ladder. If an external A/D voltage is used, it must be capable of driving the 100 Ω load of the ladder. If the internal reference is used, it must be connected to an external 0.1 µF de-coupling capacitor. |
| VGACLKIN | 53/67 | TTL input | **A/D VGA converter clock input.** Analog-to-digital converter external clock input if EXTVGASEL\ = L. |
| EXTVGASEL\ | 52/66 | TTL input | **A/D VGA clock select.** Selects the A/D clock source: EXTVGASEL\ = H: internal phase-locked loop; EXTVGASEL\ = L: external clock applied to VGACLKIN. |

| Pin Name | Pin Number FS401/FS403 | Type/Value | Pin Function Description |
|---|---|---|---|
| VGACLKDIV 403 only | x/65 | LVTTL output | **A/D clock divided by N.** VGACLKIN divided by N for connection to external phase-locked loop controller. Polarity selectable. |
| CLAMP_REF | 46/55 | LVTTL output | **VGA clock reference output.** Clamp output signal. CLAMP_REF = H, when the internal clamp is active during the horizontal sync period, HS_IN. CLAMP_REF can be used for a reference input for an external PLL. Polarity selectable. |
| EXADSEL 403 only | x/23 | TTL input | **Analog/Digital RGB in select.** LOW selects internal A/D converters; HIGH selects external digital $R_{7-0}G_{7-0}B_{7-0}$ inputs. If LOW, $B_{7-0}$ become general purpose outputs. This pin has an internal pull-down and should be left open if the internal A/D is desired. |
| **Digital RGB Inputs or OSD/GPO Pins** | | | |
| $R_{7-3}$ 403 only | x/59-56, x/41 | TTL input | **Digital red inputs.** Top 5 bits of 8-bit red input data. No connection if not used. |
| $R_2$/OSDEN 403 only | x/40 | TTL input | **Digital red input/OSD enable.** Bit 2 of red input data if EXADSEL is high. Else, OSD enable pin (high enabled). No connection if not used. |
| $R_1$/ TV_HSYNC 403 only | x/39 | TTL input/ LVTTL output | **Digital red input/TV_HSYNC.** Bit 1 of red input data if EXADSEL is high. Else, digital horizontal sync for YUV, RGB/SCART, and progressive video outputs. No connection if not used. |
| $R_0$/ TV_VSYNC 403 only | x/38 | TTL input/ LVTTL output | **Digital red input/TV_VSYNC.** Bit 0 of red input data if EXADSEL is high. Else, digital vertical sync for YUV, RGB/SCART, and progressive video outputs. No connection if not used. |
| $G_7$/OSDHT 403 only | x/27 | TTL input | **Digital green input/OSD half tone.** Bit 7 of green input data if EXADSEL is high. Else, OSD half tone input pin. No connection if not used. |
| $G_4$/OSDR 403 only | x/26 | TTL input | **Digital green input/OSD red.** Bit 4 of green input data if EXADSEL is high. Else, OSD red input pin. No connection if not used. |
| $G_5$/OSDG 403 only | x/25 | TTL input | **Digital green input/OSD green.** Bit 5 of green input data if EXADSEL is high. Else, OSD green input pin. No connection if not used. |
| $G_6$/OSDB 403 only | x/24 | TTL input | **Digital green input/OSD blue.** Bit 6 of green input data if EXADSEL is high. Else, OSD blue input pin. No connection if not used. |
| $G_{3-0}$ 403 only | x/70-73 | TTL input | **Digital green input.** Bottom 4 bits of 8-bit green input data. No connection if not used. |
| $B_{7-0}$ 403 only | x/102-105, x/120-123 | TTL input | **Digital blue input.** 8-bit blue input data if EXADSEL is high. GPO if not used (see GPO pin definition below). No connection if not used. |
| $GPO_{7-1}$ | x/102, 81/103, x/104, x/105, x/120, x/121, x/122 | LVTTL output | **General Purpose Outputs.** 2 (FS401) or 8 bits (FS403) of general purpose outputs set by SIO commands to the GPO register. No connection if not used. |
| $GPO_0$ | 96/123 | LVTTL output | **General Purpose Output 0.** When the internal CPU is enabled, this bit signals when the frequency of the VGA input is high or low. Otherwise, it is a normal GPO bit. |

| Pin Name | Pin Number FS401/FS403 | Type/Value | Pin Function Description |
|---|---|---|---|
| OSDCLK 403 only | x/19 | LVTTL output | **OSD clock.** Output clock to external microprocessor to synchronize OSD data. |
| **Video Outputs** | | | |
| Y/R/V | 12/11 | analog video | **Video output.** As programmed by Command Register $OFMT_{1-0}$ bits:<br>00 Luminance component Y of S-video.<br>01 Red component of RGB.<br>1X V component of YUV |
| CVBS/G/Y | 15/14 | analog video | **Video output.** As programmed by Command Register $OFMT_{1-0}$ bits:<br>00 Composite video.<br>01 Green component of RGB.<br>1X Y component of YUV. |
| C/B/U | 17/16 | analog video | **Video output.** As programmed by Command Register $OFMT_{1-0}$ bits:<br>00 Chrominance component of S-video.<br>01 Blue component of RGB.<br>1X U component of YUV. |
| CSYNC | 20/20 | LVTTL output | **Composite sync output.** Digital composite sync for YUV and RGB/SCART video outputs. High impedance (tri-state) when powered down. |
| TV_HSYNC | 32/39 | LVTTL output | **Horizontal sync output.** Digital horizontal sync for YUV, RGB/SCART, and progressive video outputs. High impedance (tri-state) when powered down. Note: multiplexed with $R_1$ on FS403 (see above). |
| TV_VSYNC | 31/38 | LVTTL output | **Vertical sync output.** Digital horizontal sync for YUV, RGB/SCART, and progressive video outputs. High impedance (tri-state) when powered down. Note: multiplexed with $R_0$ on FS403 (see above). |
| **D/A Voltage Reference** | | | |
| $C_{COMP}$ | 8/7 | +1.235 V | **Voltage reference input/output.** If unconnected, except for a 0.1µF capacitor to $V_{DDDA}$ for noise decoupling, the internal 1.235 Volt band-gap reference will be supplied to the three D/A Converters. An external 1.235 volt reference connected to $C_{COMP}$ will override the internal voltage reference. |
| $I_{REF}$ | 9/8 | 392/787Ω | **Current Reference.** A resistor between $I_{REF}$ and $V_{SSDA}$ sets the current range of the D/A converters. Use 392Ω for a 37.5Ω load and 787Ω for a 75Ω load. |
| $C_{BYPASS}$ | 11/10 | 0.1 µF | **Bypass Capacitor.** A 0.1µF capacitor must be connected between $C_{BYPASS}$ and $V_{DDDA}$ to reduce noise at the D/A outputs. |

| Pin Name | Pin Number FS401/FS403 | Type/Value | Pin Function Description |
|---|---|---|---|
| **Frame Buffer Port** | | | |
| $D_{15-0}$ | 79-76,73-62/ 100-97,94-91,88-81 | TTL input/ LVTTL output | **Data port, frame store memory.** 16-bit data bus for 16Mb SDRAM. |
| $A_{11-0}$ | 94-83/ 118-107 | LVTTL output | **Address port, frame store memory.** 12-bit address bus for 16Mb SDRAM. |
| RAS\ | 60/78 | LVTTL output | **Row address strobe.** RAS\ output for 16Mb SDRAM frame store memory. |
| CAS\ | 61/79 | LVTTL output | **Column address strobe.** CAS\ output for 16Mb SDRAM frame store memory. |
| WE\ | 58/76 | LVTTL output | **Write enable.** WE\ output for 16Mb SDRAM frame store memory. |
| DQM | 57/75 | LVTTL output | **Data Qualify.** Qualify data for 16Mb SDRAM frame store read/write operations. State is set by the memory access mode:<br>DQM = L, enables memory outputs for read operations and exposes memory inputs for write operations;<br>DQM = H, disables memory outputs and masks memory inputs |
| RAMCK_IN | 100/127 | TTL input | **Frame store clock input.** Frame store input clock to be routed to RAMCK_OUT if RAMCK_SEL\ = L. |
| RAMCK_SEL\ | 98/125 | TTL input | **Frame store clock select.** Selects either internal or external clock for 16Mb SDRAM frame store memory. If RAMCK_SEL\ = H, the internal clock synthesized from OSC1 is selected. With RAMCK_SEL\ = L, the RAMCK_IN input is selected. |
| RAMCK_OUT | 99/126 | LVTTL output | **Frame store clock output**. RAMCK_OUT for 16Mb SDRAM frame store memory. |
| **Serial Port** | | | |
| $SIOA_{10/7}$ | 25/31 | TTL input | **Serial address length select.** Selects the length of the serial address:<br>$SIOA_{10/7}$ = H: 10-bits<br>$SIOA_{10/7}$ = L: 7-bits |
| $SIOA_0$ | 26/32 | TTL input | **Serial data address bit 0.** . Selects the serial bus address:<br>$SIOA_0$ = H: 0x6A, 276<br>$SIOA_0$ = L: 0x4A, 224 |
| SIODATA | 24/30 | TTL input | **Serial data.** Data line of the serial port. |
| SIOCLK | 23/29 | TTL input | **Serial clock.** Clock line of the serial port. |
| **Power and Ground** | | | |
| $V_{DDDA}$ | 10, 13, 16, 18/ 9, 12, 15, 17 | +3.3 V | **D/A Converter Power.** |
| $V_{DDAD}$ | 33, 37, 38, 45/ 42, 46, 47, 54 | +3.3 V | **A/D Converter Power.** |
| $V_{DDPA}$ | 47/60 | +3.3 V | **A/D Sample Clock Phase Lock Loop Power.** |
| $V_{DDPF}$ | 1/128 | +3.3 V | **Frame Store Clock Phase Lock Loop Power.** |
| $V_{DD}$ | 19, 27, 54, 74, 82, 95/ 18, 33, 68, 80, 90, 95, 106, 119 | +3.3 V | **Digital Power.** 3.3 volt power for memory interface. |
| $V_{SSDA}$ | 7, 14/6, 13 | 0 V | **D/A Converter Ground.** |

| Pin Name | Pin Number FS401/FS403 | Type/Value | Pin Function Description |
|---|---|---|---|
| $V_{SSAD}$ | 34, 35, 40, 43/ 43, 44, 49, 52 | 0 V | **A/D Converter Ground.** |
| $V_{SSPA}$ | 48/61 | 0 V | **A/D Sample Clock Phase Lock Loop Ground.** |
| $V_{SSPF}$ | 97/124 | 0 V | **Frame Store Clock Phase Lock Loop Ground.** |
| $V_{SS}$ | 6, 21, 22, 28, 29, 51, 55, 56, 75, 80/ 5, 21, 22, 28, 34, 36, 64, 69, 74, 89, 96, 101 | 0 V | **Digital ground.** |

# 4. Control Register Definitions

## 4.1 Control Register Functions

Control register functions are summarized in Table 3. Each internal register is up to 16-bits wide. To access a register two 8-bit data words must be transferred. In the case of a write, writing the second byte loads the internal 16-bit register. Most registers have type read/write. Read only registers access the internal status of the FS400. Write-only registers initiate the transfer of data.

There are two methods for an external processor to control the FS400. The first is to directly program the FS400 SIO hardware registers. The external processor would sample continuously the IHC and IVC registers to monitor the size and speed of the input video. Additionally, it would read the SHV, EHV, SVV, and EVV to determine the size of the active area of the input image (input calibration). Lastly, the external processor would have to calculate the values for the ILS, VSC, IHS, IHA, HSC, OHO, and OVO registers.

The second method is to allow the FS400's internal micro-controller to do most of the required work. The external processor writes to a set of high level registers (software registers) to direct the internal micro-controller's actions. The internal micro-controller takes care of all the real-time sampling and calculations for the lower level registers (hardware registers).

When enabled, the internal micro-controller takes control of many of the hardware registers. These registers are GPO, ILS, VSC, IVO, IHS, IHA, HSC, IHO, CR, CRE, OHO, OVO, SHV, EHV, SVV, EVV, AVT, and FLK. The external processor should not write to these registers while the internal micro-controller is enabled.

The internal micro-controller also provides high level registers (software registers). These registers are described in the following sections. The interface the software registers is identical to the hardware registers via the SIO.

Additionally, the external processor is allowed to write to the following hardware registers: CON, BRT, SHP, and CSC.

# 4.2 Internal Micro-Controller Programming

## 4.2.1 Input Calibration

The internal micro-controller needs to know what portion of the input video signal contains active video. Determining this information is referred to as Input Calibration. The input calibration information is used by the internal micro-controller to calculate the low-level registers that scale the input image to fit on the output device.

The internal micro-controller stores the input calibration information as 4 ratios that describe the active video area in relation to the total input video signal. These 4 ratios are:

| | |
|---|---|
| SHV | (Number of pixel times from HSync to the start of the active video) / (The total pixel times from HSync to HSync) * 65536 |
| HAT | (Number of active pixels) / (The total pixel times from HSync to HSync) * 65536 |
| SVV | (Number of lines from VSync to the start of the active video) / (The total lines from VSync to VSync) * 65536 |
| VAT | (Number of active lines) / (The total lines from VSync to VSync) * 65536 |

For example, if the input video signal has 1138 pixel times from HSync to HSync, only 1024 of the pixels are active, and the first active pixel is at pixel time 57, then SHV and HAT would be:

SHV = 57 / 1138 * 65536

HAT = 1024 / 1138 * 65536

There are two methods of determining the input calibration: auto input calibration and manual input calibration. Additionally, the internal micro-controller can store the input calibration for 4 video modes.

### 4.2.1.1 Auto Input Calibration

When the FS400 is in auto input calibration mode, then the internal micro-controller will use the active video registers (SHV, EHV, SVV, and EVV) to determine the input calibration. The internal micro-controller will sample the active video registers until their values stabilize for at least 600 milliseconds.

The internal micro-controller will wait up to 3 seconds for the active video registers to stabilize. If the active video register do not stabilize, the internal micro-controller will revert to the default input calibration information to display the image. Additionally, if the input calibration values calculated from the active video registers are outside of the minimums and maximums allowed, the internal micro-controller will revert to the default input calibration information. The internal micro-controller will continue to sample the active video registers. If the registers do stabilize at a later time, the default input calibration will be replaced with the calculated input calibration.

After the initial input calibration is acquired, the internal micro-controller continues to sample the active video registers. If the input calibration calculated from the new sample indicates a larger active video area, then the new input calibration is used.

Auto input calibration is enabled by clearing the DSICAL bit in the SCR register. This is the default mode of operation. There is nothing else an external controller need to do to allow automatic calibration to work. However, an external controller can modify the behavior of auto input calibration by changing the following configuration values via the CCR and CDR registers.

| | |
|---|---|
| CCR_DEF_SHV | Default Start of Horizontal Video |
| CCR_MIN_SHV | Minimum Start of Horizontal Video |
| CCR_MAX_SHV | Maximum Start of Horizontal Video |
| CCR_DEF_HAT | Default Horizontal Active |
| CCR_MIN_HAT | Minimum Horizontal Active |

| | |
|---|---|
| CCR_MAX_HAT | Maximum Horizontal Active |
| CCR_DEF_SVV | Default Start of Vertical Video |
| CCR_MIN_SVV | Minimum Start of Vertical Video |
| CCR_MAX_SVV | Maximum Start of Vertical Video |
| CCR_DEF_VAT | Default Vertical Active |
| CCR_MIN_VAT | Minimum Vertical Active |
| CCR_MAX_VAT | Maximum Vertical Active |

An external controller can reset the input calibration and force the internal micro-controller to reacquire the input calibration information by setting the RSTICAL bit in the SCR register.

#### 4.2.1.2 Manual Input Calibration

The auto input calibration can be by-passed by setting the DSICAL bit in the SCR register. When DSICAL is set, the internal micro-controller will use the manual input calibration values. An external controller can modify the manual input calibration values by changing the following configuration values via the CCR and CDR registers:

CCR_SHV Start of Horizontal Video
CCR_HAT Horizontal Active
CCR_SVV Start of Vertical Video
CCR_VAT Vertical Active

The external controller must set the MICALNEW bit in the SCR register after each change to the manual input calibration values. The internal controller will not recognize a change in the state of DSICAL until the next video mode change or until the RSTICAL bit in the SCR register is set.

The manual input calibration values do not change when a new video mode is detected, except when Input Calibration Tables are enabled. If an external processor needs to write different manual input calibration information for new video modes, it must poll the MCC bit in the SSR register to determine when a new video mode is present. Next, the external processor can read the CCR_IVC and CCR_IHC configuration values to determine the new video mode and then write the appropriate input calibration information.

The auto input calibration can be by-passed by setting the DSICAL bit in the SCR register. When DSICAL is set, the internal micro-controller will use the manual input calibration values. An external controller can modify the manual input calibration values by changing the following configuration values via the CCR and CDR registers.

| | |
|---|---|
| CCR_SHV | Start of Horizontal Video |
| CCR_HAT | Horizontal Active |
| CCR_SVV | Start of Vertical Video |
| CCR_VAT | Vertical Active |

The external controller must set the MICALNEW bit in the SCR register after each change to the manual input calibration values.

The internal controller will not recognize a change in the state of DSICAL until the next video mode change or until the RSTICAL bit in the SCR register is set.

The manual input calibration values do not change when a new video mode is detected, except when Input Calibration Tables are enabled. If an external processor needs to write different manual input calibration information for new video modes, it must poll the MCC bit in the SSR register to determine when a new video mode is present. Next, the external processor can read the CCR_IVC and CCR_IHC configuration values to determine the new video mode and then write the appropriate input calibration information.

#### 4.2.1.3 Input Calibration Tables

The internal micro-controller can store input calibration information for four video modes when in both manual and auto input calibration mode. Setting the ENICTBL bit in the SCR register enables this feature.

During a video mode change, the internal micro-controller will save the accumulated input calibration of the last video mode and its IVC and IHC values into the input calibration table. Next, the internal micro-controller will search the table entries for an entry that match the IHC and IVC of the new mode. If a match is found, then the input calibration information is retrieved from the table.

Setting the RSTICAL bit in the SCR register will clear the table entry for the current mode.

The input calibration table can be read and modified by accessing the following configuration values via the CCR and CDR registers.

CCR_ICAL_TABLE_x_IVC
CCR_ICAL_TABLE_x_IHC
CCR_ICAL_TABLE_x_HAT
CCR_ICAL_TABLE_x_SHV
CCR_ICAL_TABLE_x_VAT
CCR_ICAL_TABLE_x_SVV

x = 0, 1, 2, or 3

### 4.2.2 Output Calibration

The internal micro-controller needs to know the location of the first displayable pixel and line and how many pixels and lines the output device can display. This information is referred to as the Output Calibration.

The internal micro-controller has built-in default output calibration that works well for supported output devices. However, an external controller can change the output calibration to suit a specific output device.

The starting pixel and starting line locations are stored in HOHOS and HOVOS. The external controller may read and write these registers at any time. Any changes will take effect upon the next output frame.

The number of pixels and lines that can be displayed are stored in the TVP (TV Pixels) and TVL (TV Lines) registers. The external controller may read and write these registers at any time. Any changes will take effect after the internal controller recalculates the low level registers.

Additionally, the FS400 is capable of display a 75% color bar pattern to assist the user in manually adjusting the output calibration. Setting the OCAL bit in the SCR register enables the 75% color bar pattern.

Any change to TV_CLK, PROG_INT, or PAL_NTSC fields of the HCRS register causes the internal micro-controller to initialize the HOHOS, HOVOS, TVL, and TVP registers to defaults appropriate to the new output mode.

### 4.2.3 Zoom

There are two zooming methods. The first is to set the ZOOM bit in the HCRES register for 2x-zoom. The second is to use the HSS (Horizontal Scale Step) and the VSS (Vertical Scale Step) registers for variable zoom.

Setting the ZOOM bit in the HCRES register causes the pixels to double in size. The HPP (Horizontal Pan Position) and the VPP (Vertical Pan Position) are used to pan the zoomed image.

The pixels double in size in the vertical direction. The internal micro-controller also attempts to double pixels in the horizontal direction. This is possible for most video modes; however, with very large input images and very small output devices the pixels will be scaled to something less than 2x horizontally.

The HSS (Horizontal Scale Step) and the VSS (Vertical Scale Step) registers can be used to scale the image up or down (larger or smaller). This method allows for a variable zoom that works independently in the horizontal and vertical directions. The HPO (Horizontal Position Offset) and the VPO (Vertical Position Offset) are used to pan the variable zoomed image.

The ZOOM bit and the HSS and VSS register pair effects are mutually exclusive. They do not compound each other. The ZOOM bit is the highest priority. If the ZOOM bit is set, then the image is in the 2x-zoom mode, regardless of the state of HSS and VSS. When the ZOOM bit is cleared, HSS and VSS control the zoom effect.

The minimum and maximum values of HSS and VSS vary depending upon the size of the input image and the size of the output device. The internal micro-controller checks the limits of these registers each time they change. If the values in the HSS and VSS registers are out of range, the internal micro-controller will overwrite the register with either the minimum value or the maximum value. However, it can take up to 25ms before the internal micro-controller does this. Therefore, an external controller should not read HSS or VSS until 25ms after writing to one of these registers.

### 4.2.4 Panning

When the ZOOM bit in the HCRES register is cleared HPO (Horizontal Position Offset) and VPO (Vertical Position Offset) control panning. HPP (Horizontal Pan Position) and VPP (Vertical Pan Position) control panning when the ZOOM bit in the HCRES register is set.

The HPP and HPO values are expressed as a signed ratio of the horizontal total * 32768. The VPP and VPO values are expressed as a signed ratio of the vertical total * 32768. These signed ratios are added to the start of active video ratios (SHV and SVV) to compute a new start of active video.

The CCR_HPO_STEP and CCR_VPO_STEP configuration values contain the percentage of the horizontal and vertical totals that represent one pixel. The processor can read CCR_HPO_STEP and add or subtract the value from HPO or HPP to move the image one pixel horizontally. The CCR_VPO_STEP value is added or subtracted from the VPO and VPP to move the image one pixel vertically.

The minimum and maximum values of HPO, VPO, HPP and VPP vary depending upon the size of the input image. The internal micro-controller checks the limits of these values each time they change. If the values in the HPO, VPO, HPP and VPP registers are out of range, the internal micro-controller will overwrite the register with either the minimum value or the maximum value. However, it can take up to 25ms before the internal micro-controller does this. Therefore, an external controller should not read HPO, VPO, HPP or VPP until 25ms after writing to one of these registers.

### 4.2.5 Picture Control

The following registers control the picture quality.

| | |
|---|---|
| CON | Contrast Coefficient |
| BRT | Brightness Coefficient |
| SHP | Sharpness Coefficient |
| CSC | Color Saturation Coefficient |

CON, BRT, SHP, and CSC are low level hardware registers. However, the internal micro-controller does not modify the contents of these registers. An external controller is free to modify them as needed.

Additionally, there are several bits in the control registers that are used to control picture quality.

In the HCRS register:

| | |
|---|---|
| LNTCH | Luminance Notch Filter |
| CBP | Chroma Bandpass Filter |
| PEDSTL | US NTSC Black Pedestal |

In the HCRES register:

| | |
|---|---|
| RGBGAIN | 1x or 1.43x RGB Input Gain |
| FREEZE | Freeze the Input Image |

In the SCR register

| | |
|---|---|
| SFLK | Flicker Filter |

### 4.2.6 Video Mode Changes

A video mode change starts when the internal micro-controller detects a significant change in either IVC (Input Vertical Count) or IHC (Input Horizontal Count). The internal micro-controller samples IVC and IHC approximately every 50ms. When a change is detected and is stable for several sample periods, the mode change process is started and the MCS (Mode Change Start) bit is set in the SSR register.

Additionally, a mode change is initiated when the RSTVMODE bit is set in the SCR register or when the OCAL bit in the SCR register is cleared.

If the input calibration tables are enabled (ENICTBL bit in the SCR register is set), then the accumulated input calibration for the old video mode is stored in the input calibration table. Next, the input calibration table is searched for an entry that matches the IVC and IHC of the new mode. If one is found, then the input calibration is retrieved from the table.

If manual input calibration is enabled (DSICAL bit in SCR register is set) and the no input calibration information was found in the input calibration table, then the input calibration information is retrieved from the manual input calibration configuration values.

If auto input calibration is enabled (DSICAL bit in SCR register is clear) and the no input calibration information was found in the input calibration table, then the internal micro-controller begins sampling of the active video registers (SHV, EHV, SVV, and EVV).

Once the input calibration information is obtained, the internal micro-controller calculates the values for the low-level hardware registers and programs them. At this point the MCS bit in the SSR is cleared and the MCC (Mode Change Complete) bit in the SSR is set.

A video mode change does not change the state of any other software registers. This includes the FREEZE and ZOOM bits in the HCRES register and the HPO, VPO, HPP, VPP, HSS, and VSS registers. If the external controller modifies any of these, it might be desirable for the external controller to return them to there reset state after a video mode change is complete.

An external controller can detect a video mode change by polling the MCC bit in the SSR. The MCC bit remains set until an external controller sets the CLRMCC bit in the SCR register. The recommended polling frequency is at least once per second, but not more than 20 times per second.

### 4.2.7 By-passing the Internal Micro-Controller

When the FS400's built-in microprocessor is enabled, it is responsible for programming the following low-level hardware registers:

| | |
|---|---|
| IHO | Input Horizontal Offset |
| IVO | Input Vertical Offset |

| | |
|---|---|
| IHA | Input Horizontal Active |
| IHS | Input Horizontal Samples |
| OHO | Output Horizontal Offset |
| OVO | Output Vertical Offset |
| VSC | Vertical Scaling Coefficient |
| CR | Control Register |
| CRE | Control Register Extended |
| HSC | Horizontal Scaling Coefficient |
| FLK | Flicker Filter |
| AVT | Active Video Threshold |

The internal micro-controller also accesses the following read-only low-level hardware registers:

| | |
|---|---|
| IVC | Input Vertical Count |
| IHC | Input Horizontal Count |
| SHV | Start Horizontal Video |
| EHV | End Horizontal Video |
| SVV | Start Vertical Video |
| EVV | End Vertical Video |

When the internal micro-controller is enabled, an external controller should only access the following registers:

| | |
|---|---|
| SCR | Software Control Register |
| SSR | Software Status Register |
| HCRS | Hardware Control Register Shadow |
| HCRES | Hardware Control Register Extended Shadow |
| HPO | Horizontal Position Offset |
| VPO | Vertical Position Offset |
| HSS | Horizontal Scale Step |
| VSS | Vertical Scale Step |
| HPP | Horizontal Pan Position |
| VPP | Vertical Pan Position |
| TVP | TV Pixels |
| TVL | TV Lines |
| CCR | Configuration Command Register |
| CDR | Configuration Data Register |
| HOHOS | Hardware Output Horizontal Offset Shadow |
| HOVOS | Hardware Output Vertical Offset Shadow |
| CON | Contrast Coefficient |
| BRT | Brightness Coefficient |
| SHP | Sharpness Coefficient |
| CSC | Color Saturation Coefficient |

An external controller can disable the internal micro-controller by clearing the ENABLE bit in the SCR register. When disabled (either through software or using the INTCPUEN pin), the internal micro-controller will no longer access any of the FS400 registers. The external controller becomes responsible for programming all of the low-level hardware registers.

The internal micro-controller will only respond to the RESET and ENABLE bits in the SCR register while it is disabled. Changes to any of the other software registers (SCR, HCRS, HCRES, HPO, VPO, HSS, VSS, HPP, VPP, TVP, TVL, CCR, CDR, HOHOS, HOVOS) will have no effect until the internal micro-controller is re-enabled.

The micro controller can be re-enabled by setting the ENABLE bit in the SCR register. The internal micro-controller will begin processing at the same point that it was disabled at. Any changes to the software registers will be processed. The low-level hardware register will eventually be re-programmed.

It is also possible to disable two portions of the internal micro-controllers program code. These are the video mode detect and the auto input calibration program code.

The video mode detect program code is disabled by setting the DSVMDET bit in the SCR register. The video mode detect code usually samples the IHC and IVC registers looking for a stable change to indicate a new video mode. When this code is disabled, no new video modes will be detected. An external processor could sample IHC and IVC or use another method to determine a video mode change. The external processor would then have to program the CCR_IHC and the CCR_IVC configuration values via the CCR and CDR and then set the RSTVMODE bit in the SCR to force a video mode update. The video mode detect code can be re-enabled at any time by clearing the DSVMET bit in the SCR register.

Setting the DSICAL bit in the SCR register disables the input calibration program code. This referred to as manual input calibration mode and is discussed in section 4.2.1.2 above.

### 4.2.8 Special Internal Micro-Controller SIO Requirements

The interface to the internal micro-controller's software registers is identical to the hardware registers via the SIO. However, the internal micro-controller is not able to process code while an external processor is accessing the SIO. Therefore, repeated back to back SIO access can starve the internal micro-controller of processing time. This is not a concern for short bursts of back to back SIO accesses, but long loops of repeated polling of an SIO register should be avoided.

If it is necessary to repeatedly poll a register via the SIO, it is recommended that there be a delay of about 1 to 5 millisecond between SIO accesses. This will allow the internal micro-controller to continue it's processing in a timely manor.

One example of the need to repeated poll an SIO register is when an external processor writes a CCR_READ command to the CCR register. The external processor must wait until the CCR register is set to CCR_COMPLETE (0) before it could read the data from the CDR. A programmer might be tempted to write a loop that continuously reads the CCR register until it is set to CCR_COMPLETE. This would slow down or even stop the processing in the internal micro-controller. To avoid this, add at least a 1 to 5ms delay into the loop to allow the internal micro-controller to continue processing.

## 4.3 Disabling the Internal Micro-Controller

To be added later.

## 4.4 Control Register Definitions

### Table 3. Control Register Map

| Function | | | | |
|---|---|---|---|---|
| **Reg.** | **Bit #** | **Name** | **Type** | **Reset Value** |
| **Input Horizontal Offset** | | | | |
| 0 | 7-0 | $IHO_{7-0}$ | R/W | 80 (128.) |
| 1 | 2-0 | $IHO_{10-8}$ | R/W | 00 |
| **Input Vertical Offset** | | | | |
| 2 | 7-0 | $IVO_{7-0}$ | R/W | 05 |
| 3 | 3-0 | $IVO_{11-8}$ | R/W | 00 |
| **Input Horizontal Active Width** | | | | |
| 4 | 7-0 | $IHAW_{7-0}$ | R/W | 00 (512.) |
| 5 | 1-0 | $IHAW_{9-8}$ | R/W | 02 |
| **Input Lines Stored** | | | | |
| 6 | 7-0 | $ILS_{7-0}$ | R/W | DC (220.) |
| 7 | 1-0 | $ILS_{9-8}$ | R/W | 00 |
| **Input Horizontal Samples (per line)** | | | | |
| 8 | 7-0 | $IHS_{7-0}$ | R/W | FE (766.) |
| 9 | 2-0 | $IHS_{10-8}$ | R/W | 02 |
| **Input Horizontal Count** | | | | |
| A | 7-0 | $IHC_{7-0}$ | R | |
| B | 1-0 | $IHC_{9-8}$ | R | |
| **Input Vertical Count (lines)** | | | | |
| C | 7-0 | $IVC_{7-0}$ | R | |
| D | 3-0 | $IVC_{11-8}$ | R | |
| **Vertical Scaling Coefficient** | | | | |
| E | 7-0 | $VSC_{7-0}$ | R/W | 00 |
| F | - | - | R/W | 00 |
| **Command Register** | | | | |
| 10 | 7-0 | $CR_{7-0}$ | R/W | 00 |
| 11 | 5-0 | $CR_{13-8}$ | R/W | 00 |
| **Status Register** | | | | |
| 12 | 7-0 | $SR_{7-0}$ | R | 00 |
| 13 | 7-0 | $SR_{15-8}$ | R | 02 |
| **Command Register Extended** | | | | |
| 14 | 7-0 | $CRE_{7-0}$ | R/W | 00 (256.) |
| 15 | 3-0 | $CRE_{11-8}$ | R/W | 01 |
| **Start Horizontal Active VGA** | | | | |
| 16 | 7-0 | $SHV_{7-0}$ | R | |
| 17 | 2-0 | $SHV_{10-8}$ | R | |
| **End Horizontal Active VGA** | | | | |
| 18 | 7-0 | $EHV_{7-0}$ | R | |
| 19 | 2-0 | $EHV_{10-8}$ | R | |
| **Start Vertical Active VGA** | | | | |
| 1A | 7-0 | $SVV_{7-0}$ | R | |
| 1B | 2-0 | $SVV_{10-8}$ | R | |
| **End Vertical Active VGA** | | | | |
| 1C | 7-0 | $EVV_{7-0}$ | R | |
| 1D | 2-0 | $EVV_{10-8}$ | R | |
| **Active Video Threshold** | | | | |
| 1E | 7-0 | $AVT_{7-0}$ | R/W | 00 |
| 1F | - | - | R/W | 00 |

| Function | | | | |
|---|---|---|---|---|
| **Reg.** | **Bit #** | **Name** | **Type** | **Reset Value** |
| **Output Horizontal Offset** | | | | |
| 20 | 7-0 | $OHO_{7-0}$ | R/W | C0 (192.) |
| 21 | 1-0 | $OHO_{9-8}$ | R/W | 00 |
| **Output Vertical Offset** | | | | |
| 22 | 7-0 | $OVO_{7-0}$ | R/W | 20 (32.) |
| 23 | 1-0 | $OVO_{9-8}$ | R/W | 00 |
| **Horizontal Scaling Coefficient** | | | | |
| 24 | 7-0 | $HSC_{7-0}$ | R/W | 00 |
| 25 | 0 | $HSC_{8}$ | R/W | 00 |
| **Contrast Coefficient** | | | | |
| 26 | 5-0 | $CON_{5-0}$ | R/W | 20 (32.) |
| 27 | - | - | - | 00 |
| **Brightness Coefficient** | | | | |
| 28 | 7-0 | $BRT_{7-0}$ | R/W | 00 |
| 29 | - | - | - | 00 |
| **Sharpness Coefficient** | | | | |
| 2A | 4-0 | $SHP_{4-0}$ | R/W | 00 |
| 2B | - | - | - | 00 |
| **Flicker Filter Coefficient** | | | | |
| 2C | 4-0 | $FLK_{4-0}$ | R/W | 00 |
| 2D | - | - | - | 00 |
| **Color Saturation Coefficient** | | | | |
| 2E | 5-0 | $CSC_{5-0}$ | R/W | 20 (32.) |
| 2F | - | - | - | 00 |
| **General Purpose Outputs** | | | | |
| 34 | 7-0 | $GPO_{7-0}$ | R/W | 00 |
| 35 | - | - | - | 00 |
| **Software Control Register** | | | | |
| 60 | 7-0 | $SCR_{7-0}$ | R/W | 02 (33794.) |
| 61 | 7-0 | $SCR_{15-8}$ | R/W | 84 |
| **Software Status Register** | | | | |
| 62 | 7-0 | $SSR_{7-0}$ | R | |
| 63 | 7-0 | $SSR_{15-8}$ | R | |
| **Hardware Control Register Shadow** | | | | |
| 64 | 7-0 | $HCRS_{7-0}$ | R/W | 80 (128.) |
| 65 | 7-0 | $HCRS_{15-8}$ | R/W | 00 |
| **Hardware Control Register Extended Shadow** | | | | |
| 66 | 7-0 | $HCRES_{7-0}$ | R/W | 20 (800.) |
| 67 | 7-0 | $HCRES_{15-8}$ | R/W | 03 |
| **Horizontal Position Offset** | | | | |
| 68 | 7-0 | $HPO_{7-0}$ | R/W | 00 |
| 69 | 7-0 | $HPO_{15-8}$ | R/W | 00 |
| **Vertical Position Offset** | | | | |
| 6A | 7-0 | $VPO_{7-0}$ | R/W | 00 |
| 6B | 7-0 | $VPO_{15-8}$ | R/W | 00 |
| **Horizontal Scale Step** | | | | |
| 6C | 7-0 | $HSS_{7-0}$ | R/W | 00 |
| 6D | 7-0 | $HSS_{16-8}$ | R/W | 00 |
| **Vertical Scale Step** | | | | |
| 6E | 7-0 | $VSS_{7-0}$ | R/W | 00 |
| 6F | 7-0 | $VSS_{15-8}$ | R/W | 00 |

| Function | | | | |
|---|---|---|---|---|
| **Reg.** | **Bit #** | **Name** | **Type** | **Reset Value** |
| **Horizontal Pan Position** | | | | |
| 70 | 7-0 | $HPP_{7-0}$ | R/W | 00 |
| 71 | 7-0 | $HPP_{15-8}$ | R/W | 00 |
| **Vertical Pan Position** | | | | |
| 72 | 7-0 | $VPP_{7-0}$ | R/W | 00 |
| 73 | 7-0 | $VPP_{15-8}$ | R/W | 00 |
| **TV Pixels** | | | | |
| 74 | 7-0 | $TVP_{7-0}$ | R/W | 92 (658.) |
| 75 | 7-0 | $TVP_{15-8}$ | R/W | 02 |
| **TV Lines** | | | | |
| 76 | 7-0 | $TVL_{7-0}$ | R/W | 9C (412.) |
| 77 | 7-0 | $TVL_{15-8}$ | R/W | 01 |
| **Configuration Command Register** | | | | |
| 78 | 7-0 | $CCR_{7-0}$ | R/W | 00 |
| 79 | 7-0 | $CCR_{15-8}$ | R/W | 00 |
| **Configuration Data Register** | | | | |
| 7A | 7-0 | $CDR_{7-0}$ | R/W | 00 |
| 7B | 7-0 | $CDR_{15-8}$ | R/W | 00 |
| **Hardware Output Horizontal Offset Shadow** | | | | |
| 7C | 7-0 | $HOHOS_{7-0}$ | R/W | B2 (178.) |
| 7D | 7-0 | $HOHOS_{15-8}$ | R/W | 00 |
| **Hardware Output Vertical Offset Shadow** | | | | |
| 7E | 7-0 | $HOVOS_{7-0}$ | R/W | 26 (38.) |
| 7F | 7-0 | $HOVOS_{15-8}$ | R/W | 00 |

Following reset at power-up, the status of the internal registers is as indicated under “Reset Value” above. Note set all unused bits are set to zero and all unused register bits are set to zero for readback.

## 4.5 Control Registers Definitions

In the following definitions, range is defined as:

{min value : [max value]}

### 4.5.1 IHO - Input Horizontal Offset

#### Input Horizontal Offset Low (0)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IHO_7$ | $IHO_6$ | $IHO_5$ | $IHO_4$ | $IHO_3$ | $IHO_2$ | $IHO_1$ | $IHO_0$ |

#### Input Horizontal Offset High (1)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $IHO_{10}$ | $IHO_9$ | $IHO_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 1, 0 | 2-0, 7-0 | $IHO_{10-0}$ | **Input horizontal offset bits [10-0].** Horizontal displacement of the image in pixels from the leading edge of horizontal sync. Programming a value greater than IHS is illegal, preventing any pixels from being written into the Frame Store. |

**Range:** {0 : [IHS-1]}

## 4.5.2 IVO - Input Vertical Offset

### Input Vertical Offset Low (2)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IVO_7$ | $IVO_6$ | $IVO_5$ | $IVO_4$ | $IVO_3$ | $IVO_2$ | $IVO_1$ | $IVO_0$ |

### Input Vertical Offset High (3)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $IVO_{11}$ | $IVO_{10}$ | $IVO_9$ | $IVO_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 3, 2 | 3-0, 7-0 | $IVO_{11-0}$ | **Input vertical offset bits [11:0].** Vertical displacement of the image in lines from the leading edge of vertical sync plus a one line bias. |

**Range:** {0 : [IVC - 1]}

Note: Input Vertical Count (IVC) must be read to obtain the register value.

### 4.5.3 IHAW - Horizontal Active Width

#### Horizontal Active Width Low (4)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IHAW_7$ | $IHAW_6$ | $IHAW_5$ | $IHAW_4$ | $IHAW_3$ | $IHAW_2$ | $IHAW_1$ | $IHAW_0$ |

#### Horizontal Active Width High (5)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | $IHAW_9$ | $IHAW_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 5, 4 | 1-0, 7-0 | $IHAW_{9-0}$ | **Horizontal active width [9:0].** Number of incoming pixels to be stored in the Frame Store Memory following extraction from the incoming active video area. |

**Range:** {0 : [1023] }

**Note:** IHAW optimizes the number of stored pixels. Line store capacity limits the maximum value of IHAW to 1023 active pixels per video line

**Restrictions:**

1. IHAW < IHS-1
2. NTSC:

$$IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

3. Zoom NTSC:

$$2 * IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

4. PAL:

$$IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

5. Zoom PAL:

$$2 * IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

### 4.5.4 ILS - Input Lines Stored

#### Input Lines Stored Low (6)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $ILS_7$ | $ILS_6$ | $ILS_5$ | $ILS_4$ | $ILS_3$ | $ILS_2$ | $ILS_1$ | $ILS_0$ |

#### Input Lines Stored High (7)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | $ILS_9$ | $ILS_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 7, 6 | 1-0, 7-0 | $ILS_{9-0}$ | **Input lines stored bits [9:0].** Number of incoming lines per field to be stored in the Frame Store Memory following extraction from the incoming active video area. |

**Range:** {0 : [(IVC - 1)]}

**Restrictions:**

1. At all times:

$$ILS < \left( \frac{IVC - 1}{1 - \frac{VSC}{64}} \right)$$

2. NTSC :

   ILS + 2*OVO < 525

3. PAL:

   ILS + 2*OVO < 625

### 4.5.5 IHS - Input Horizontal Samples

#### Input Horizontal Samples Low (8)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IHS_7$ | $IHS_6$ | $IHS_5$ | $IHS_4$ | $IHS_3$ | $IHS_2$ | $IHS_1$ | $IHS_0$ |

#### Input Horizontal Samples High (9)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $IHS_{10}$ | $IHS_9$ | $IHS_8$ |

| Register | Bit# | Name | Description |
|---|---|---|---|
| 9, 8 | 2-0, 7-0 | $IHS_{10-0}$ | **Input horizontal line samples bits [10:0].** Terminal count of the number of clocks per horizontal line between incoming horizontal sync pulses. Internal ADCK phase-locked loop is programmed with this value. |

**Range:** 16MHz ≤ IHS**Incoming Horizontal Line Frequency* ≤ 50MHz

### 4.5.6 IHC - Input Horizontal Count

#### Input Horizontal Count Low (A)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IHC_7$ | $IHC_6$ | $IHC_5$ | $IHC_4$ | $IHC_3$ | $IHC_2$ | $IHC_1$ | $IHC_0$ |

#### Input Horizontal Count High (B)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | $IHC_9$ | $IHC_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| B, A | 1-0, 7-0 | $IHC_{9-0}$ | **Input horizontal count bits [9:0] (read only).** Number of $4f_{SC}$ clock pulses per horizontal line. $4f_{SC}$ is the selected OSC clock (17.734 MHz for PAL or 14.318 MHz for NTSC). IHC can be read to determine the incoming horizontal line frequency for auto selection of the incoming video format. |

**Range:** {0:[1023]}

### 4.5.7 IVC - Input Vertical Count

#### Input Vertical Count Low (C)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $IVC_7$ | $IVC_6$ | $IVC_5$ | $IVC_4$ | $IVC_3$ | $IVC_2$ | $IVC_1$ | $IVC_0$ |

#### Input Vertical Count High (D)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $IVC_{11}$ | $IVC_{10}$ | $IVC_9$ | $IVC_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| D, C | 3-0, 7-0 | $IVC_{11-0}$ | **Input vertical line count bits [11:0] (read only).** Terminal count of the number of incoming lines per frame. IVC is used to determine the incoming vertical refresh frequency for auto selection of the incoming video format. |

**Range:** {0:[4095]}

## 4.5.8 VSC – Vertical Scaling Coefficient

### Vertical Scaling Coefficient (E)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VSC_7$ | $VSC_6$ | $VSC_5$ | $VSC_4$ | $VSC_3$ | $VSC_2$ | $VSC_1$ | $VSC_0$ |

### Vertical Scaling Coefficient (F)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| E | 7-0 | $VSC_{7-0}$ | **Vertical scaling coefficient bits [7:0], m.** Vertical scaling factor = (1 - m/ 256). m is the line reduction number modulo 256 due to vertical filtering. |
| F | - | - | **Load.** Writing to register F activates register E data. |

**Range:** {0:[192]}

**Restrictions:**

1. At all times:

$$ILS < \frac{1}{2} * \left( \frac{IVC - 1}{1 - \frac{VSC}{64}} \right)$$

2. NTSC :

   ILS + OVO < 259

3. PAL:

   ILS + OVO < 310

**Example:** 480 line VGA image mapped to 400 line image.

VSC = (1-(400/480))*256 = 42.667

Use 42 for the Coefficient value

### 4.5.9 CR - Command Register

## Command Register Low (10)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PEDSTL | CBP | LNTCH | YCOFF | COMPOFF | ADCOFF | CLKOFF | RESET |

## Command Register High(11)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $OFMT_1$ | $OFMT_0$ | PAL_NTSC | PROG_INT | $TV\_CLK_1$ | $TV\_CLK_0$ |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 10 | 0 | RESET | 0: run<br>1: reset video pipeline (control registers are not affected) |
| 10 | 1 | CLKOFF | 0: FS400 clocks powered on.<br>1: FS400 clocks powered off. |
| 10 | 2 | ADCOFF | 0: A/D converter powered on.<br>1: A/D converter powered off. |
| 10 | 3 | COMPOFF | 0: Composite D/A converter powered on.<br>1: Composite D/A converter powered off. |
| 10 | 4 | YCOFF | 0: Y/C & Chroma D/A converter powered on.<br>1: Y/C & Chroma D/A converter powered off. |
| 10 | 5 | LNTCH | 0: Bypass luminance notch filter.<br>1: Insert luminance notch filter. |
| 10 | 6 | CBP | 0: Bypass chroma bandpass filter.<br>1: Insert chroma bandpass filter. |
| 10 | 7 | PEDSTL | 0: Bypass pedestal<br>1: Insert US NTSC black pedestal |
| 11 | 3-0 | $TV\_CLK_{1-0}$ | TV Clock Select<br>00 = $4F_{SC}$ crystal or oscillator<br>01 = non-subcarrier crystal or oscillator<br>10 = $8F_{SC}$ crystal or oscillator |
| 11 | 2 | PROG_INT | 0 = Interlaced Output<br>1 = Progressive Output |
| 11 | 3 | PAL_NTSC | 0 = NTSC<br>1 = PAL |
| 11 | 5-4 | $OFMT_{1-0}$ | Output Format<br>00 = Y/C and Composite Video<br>01 = RGB<br>1x = YUV |

**Typical Output Mode Settings:**

| TV Mode | TV_CLK | PROG_INT | PAL_NTSC |
|---|---|---|---|
| NTSC | 0 | 0 | 0 |
| PAL | 0 | 0 | 1 |
| VGA | 2 | 1 | 0 |
| SVGA | 2 | 1 | 1 |
| 100Hz PAL | 1 | 0 | 1 |
| Progressive NTSC (720x480) | 1 | 1 | 0 |
| Super NTSC (1280x480) | 2 | 0 | 0 |
| Super PAL (1280x575) | 2 | 0 | 1 |

## 4.5.10 SR – Status Register

### Status Register Low (12)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $REVID_3$ | $REVID_2$ | $REVID_1$ | $REVID_0$ |

### Status Register High (13)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 12 | 3-0 | $REVID_{3-0}$ | **Chip revision identification.** Starts at 0.<br>Revision History: April 1999 – Revision 0 |
| 13 | 6-0 | $FAMILY_{6-0}$ | **Family ID.** Indicates the product family for chip identification. |

## 4.5.11 CRE – Command Register Extended

### Command Register Extended Low (14)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | VGAINTDET | VGAREFPOL | VGACLKPOL | FREEZE | RGBGAIN | ZOOM |

### Command Register Extended High (15)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EVVCLR | SVVCLR | EHVCLR | SHVCLR | BIPTALL | $BIPGEN_2$ | $BIPGEN_1$ | $BIPGEN_0$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 14 | 5 | VGAINTDET | 0: Ignore interlaced input<br>1: Detect interlaced input |
| 14 | 4 | VGAREFPOL (read only) | 0 = VGA syncs reference polarity rising edge<br>1 = VGA syncs reference polarity falling edge |
| 14 | 3 | VGACLKPOL (read only) | 0 = VGA clock reference polarity rising edge<br>1 = VGA clock reference polarity falling edge |
| 14 | 2 | FREEZE | 0: Continuous update<br>1: Freeze frame |
| 14 | 1 | RGBGAIN | 0: 1x input gain<br>1: 1.43 input gain |
| 14 | 0 | ZOOM | 0: Zoom off<br>1: Zoom on |
| 15 | 7 | EVVCLR | 0: run<br>1: clear EVV and re-detect |
| 15 | 6 | SVVCLR | 0: run<br>1: clear SVV and re-detect |
| 15 | 5 | EHVCLR | 0: run<br>1: clear EHV and re-detect |
| 15 | 4 | SHVCLR | 0: run<br>1: clear SHV and re-detect |
| 15 | 3 | BIPTALL | 0: Normal BIPGEN image size (512x512)<br>1: Tall BIPGEN image size (512x1024) |
| 15 | 2-0 | $BIPGEN_{2-0}$ | Built-in Pattern Generator:<br>000 = Disabled<br>001 = All Waveforms<br>010 = 100% Color Bars<br>011 = 75% Color Bars<br>100 = Cross Hatch<br>101 = Red Gradient<br>110 = Green Gradient<br>111 = Blue Gradient |

**Notes:**
When ZOOM = 1, FLK and SHP registers are disabled, and IHAW should be reduced by ½.

### 4.5.12 Start Horizontal Active VGA

#### Start Horizontal Active VGA Low (16)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $SHV_7$ | $SHV_6$ | $SHV_5$ | $SHV_4$ | $SHV_3$ | $SHV_2$ | $SHV_1$ | $SHV_0$ |

#### Start Horizontal Active VGA High (17)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $SHV_{10}$ | $SHV_9$ | $SHV_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 17, 16 | 2-0, 7-0 | $SHV_{10-0}$ | **Start Horizontal Active VGA bits [10:0] (read only).** Number of $4f_{SC}$ clocks from the start of a line to the first detected active video. |

**Range:** {0:[2047]}

## 4.5.13 End Horizontal Active VGA

### End Horizontal Active VGA Low (18)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $EHV_7$ | $EHV_6$ | $EHV_5$ | $EHV_4$ | $EHV_3$ | $EHV_2$ | $EHV_1$ | $EHV_0$ |

### End Horizontal Active VGA High (19)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $EHV_{10}$ | $EHV_9$ | $EHV_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 19, 18 | 2-0, 7-0 | $EHV_{10-0}$ | **End Horizontal Active VGA bits [10:0] (read only).** Number of $4f_{SC}$ clocks from the start of a line to the last detected active video. |

**Range:** {0:[2047]}

## 4.5.14 Start Vertical Active VGA

### Start Vertical Active VGA Low (1A)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $SVV_7$ | $SVV_6$ | $SVV_5$ | $SVV_4$ | $SVV_3$ | $SVV_2$ | $SVV_1$ | $SVV_0$ |

### Start Vertical Active VGA High (1B)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $SVV_{10}$ | $SVV_9$ | $SVV_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 1B, 1A | 2-0, 7-0 | $SVV_{10-0}$ | **Start Vertical Active VGA bits [10:0] (read only).** Number of $4f_{SC}$ clocks from the start of a frame to the first detected active video. |

**Range:** {0:[2047]}

## 4.5.15 End Vertical Active VGA

### End Vertical Active VGA Low (1C)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $EVV_7$ | $EVV_6$ | $EVV_5$ | $EVV_4$ | $EVV_3$ | $EVV_2$ | $EVV_1$ | $EVV_0$ |

### End Vertical Active VGA High (1D)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $EVV_{10}$ | $EVV_9$ | $EVV_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 1D, 1C | 2-0, 7-0 | $EVV_{10-0}$ | **End Vertical Active VGA bits [10:0] (read only).** Number of $4f_{SC}$ clocks from the start of a frame to the last detected active video. |

**Range:** {0:[2047]}

### 4.5.16 Active Video Threshold

#### Active Video Threshold Low (1E)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $AVT_7$ | $AVT_6$ | $AVT_5$ | $AVT_4$ | $AVT_3$ | $AVT_2$ | $AVT_1$ | $AVT_0$ |

#### Active Video Threshold High (1F)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 1F, 1E | 7-0 | $AVT_{7-0}$ | **Active Threshold bits [7:0]** The minimum value that should be considered not blank on the RGB input lines. This is used by the FS400 to compare to the input RGB when checking for active video to determine the SHV, EHV, SVV, and EVV values. |

**Range:** {0:[255]}

## 4.5.17 OHO - Output Horizontal Offset

### Output Horizontal Offset Low (20)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $OHO_7$ | $OHO_6$ | $OHO_5$ | $OHO_4$ | $OHO_3$ | $OHO_2$ | $OHO_1$ | $OHO_0$ |

### Output Horizontal Offset High (21)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $OHO_{10}$ | $OHO_9$ | $OHO_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 21, 20 | 10-0 | $OHO_{10-0}$ | **Output horizontal offset [10:0].** Horizontal displacement of the outgoing active video area image in pixels from the leading edge of outgoing horizontal sync. |

**Range:** NTSC: {126,[ 454]}
PAL: {177, [ 566]}

**Restrictions:**

1. IHAW < IHS-1
2. NTSC:

$$IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

3. Zoom NTSC:

$$2 * IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

4. PAL

$$2 * IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

5. Zoom PAL:

$$2 * IHAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

## 4.5.18 OVO – Output Vertical Offset

### Output Vertical Offset Low (22)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $OVO_7$ | $OVO_6$ | $OVO_5$ | $OVO_4$ | $OVO_3$ | $OVO_2$ | $OVO_1$ | $OVO_0$ |

### Output Vertical Offset High (23)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | $OVO_9$ | $OVO_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 23, 22 | 9-0 | $OVO_{9-0}$ | **Output vertical offset [9:0].** Vertical displacement of the outgoing active video area in lines from the beginning of the equalization pulses: line 1/263 for NTSC; line 311/623 for PAL. |

**Range:** NTSC: {20,[ 262]}
PAL: {25,[ 312]}

**Restrictions:**

1. At all times:

$$ILS < \frac{1}{2} * \left( \frac{IVC - 1}{1 - \frac{VSC}{64}} \right)$$

2. NTSC :

   ILS + OVO < 259

3. PAL:

   ILS + OVO < 310

## 4.5.19 HSC – Horizontal Scaling Coefficient

### Horizontal Scaling Coefficient (24)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HSC_7$ | $HSC_6$ | $HSC_5$ | $HSC_4$ | $HSC_3$ | $HSC_2$ | $HSC_1$ | $HSC_0$ |

### Horizontal Scaling Coefficient (25)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | $HSC_8$ |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 25, 24 | 8-0 | $HSC_{8-0}$ | **Horizontal scaling coefficient, n [5:0].** Horizontal scaling factor = (1 + n/ 128) |

**Range:** {0:[383]}

**Restrictions:**

1. IHAW < IHS-1
2. NTSC:

$$HAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

3. Zoom NTSC:

$$2 * HAW * \left(1 + \frac{HSC}{64}\right) + OHO < 894$$

4. PAL:

$$HAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

5. Zoom PAL:

$$2 * HAW * \left(1 + \frac{HSC}{64}\right) + OHO < 1125$$

**Example:** HSC = 3F hex; n = 63; Scaling factor = (1 + 63/128) = 1.49.

## 4.5.20 Contrast Coefficient

### Contrast Coefficient Low (26)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $CON_5$ | $CON_4$ | $CON_3$ | $CON_2$ | $CON_1$ | $CON_0$ |

### Contrast Coefficient High (27)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 26 | 5-0 | $CON_{5-0}$ | **Contrast Coefficient bits [5:0]** The Contrast Coefficient provides a luma gain adjustment in the Luma Encoder. Contrast has an implicit denominator of 32. |

**Range:** {0:[63]}

Example: CON = 21 hex; 33 decimal; has a luma gain of 33/32 = 1.03 for a 3% increase in contrast.
CON = 1f hex; 31 decimal; has a gain of 31/32 = .97; for a 3% decrease in contrast

## 4.5.21 Brightness Coefficient

### Brightness Coefficient Low (28)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $BRT_7$ | $BRT_6$ | $BRT_5$ | $BRT_4$ | $BRT_3$ | $BRT_2$ | $BRT_1$ | $BRT_0$ |

### Brightness Coefficient High (29)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 29, 28 | 7-0 | $BRT_{7-0}$ | **Brightness Coefficient bits [7:0]** The Brightness Coefficient provides an offset adjustment in the Luma Encoder. BRT is a signed 8-bit number. |

**Range:** {-128:[127]}

Example: BRT = FF hex; -1 decimal; for a slight decrease in the average picture level.

## 4.5.22 Sharpness Coefficient

### Sharpness Coefficient Low (2A)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $SHP_4$ | $SHP_3$ | $SHP_2$ | $SHP_1$ | $SHP_0$ |

### Sharpness Coefficient High (2B)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 2B, 2A | 4-0 | $SHP_{4-0}$ | **Sharpness Coefficient bits [4:0]** The Sharpness Coefficient combines with the Flicker Filter Coefficient to provide a wide range of edge enhancement and detail vs. flicker. The Sharpness Coefficient controls the intensity of the edge enhancement (peaking function). Sharpness has an implicit denominator of 16. |

**Range:** {0:[31]}

Example: SHP = 10 hex; 16 decimal; for a sharpness of 100% (flat diagonal frequency response)

## 4.5.23 Flicker Filter Coefficient

### Flicker Filter Coefficient Low (2C)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $FLK_4$ | $FLK_3$ | $FLK_2$ | $FLK_1$ | $FLK_0$ |

### Flicker Filter Coefficient High (2D)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 2D, 2C | 4-0 | $FLK_{4-0}$ | **Flicker Filter Coefficient bits [4:0]** The Flicker Filter Coefficient combines with the Sharpness Coefficient to provide a wide range of edge enhancement and detail vs. flicker. The Flicker Filter Coefficient controls the width and height of the edge enhancement (impulse function). Flicker Filter has an implicit denominator of 16. |

**Range:** {0:[21]}

Filter Coefficients: FLK/64, 1-FLK/32, FLK/64

Example: FLK = 10 hex; 16 decimal; for a ¼, ½, ¼, three line average flicker filter.
FLK = 0 hex; 0 decimal; for 0, 1, 0, no flicker filter.

Note: If FLK is set to a value greater than 21, then the weight factor applied to the lines above and below the current line are greater than the current line creating a double line effect. Therefore, values > 21 are not recommended as they will create visual artifacts, but the FS400 does not prohibit these settings.

## 4.5.24 Color Saturation Coefficient

### Color Saturation Coefficient Low (2E)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $CSC_5$ | $CSC_4$ | $CSC_3$ | $CSC_2$ | $CSC_1$ | $CSC_0$ |

### Color Saturation Coefficient High (2F)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 2F, 2E | 5-0 | $CSC_{5-0}$ | **Color Saturation Coefficient bits [5:0]** The Color Saturation Coefficient provides a gain adjustment in the Chroma Encoder. Saturation has an implicit denominator of 32. |

**Range:** {0:[63]}

Example: CSC = 21 hex; 33 decimal; has a luma gain of 33/32 = 1.03; for a 3% increase in saturation.
CSC = 1f hex; 31 decimal; has a gain of 31/32 = .97; for a 3% decrease in saturation

## 4.5.25 General Purpose Outputs

### General Purpose Outputs Low (34)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $GPO_7$ | $GPO_6$ | $GPO_5$ | $GPO_4$ | $GPO_3$ | $GPO_2$ | $GPO_1$ | $GPO_0$ |

### General Purpose Outputs High (35)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Register | Bit# | Bit Name | Description |
|---|---|---|---|
| 34 | 7-0 | $GPO_{7-0}$ | **General Purpose Output bits [7:0].** The General Purpose Output bits control the GPO pins on the FS400. The FS403 has all 8 GPO outputs available while the FS401 has only $GPO_6$ and $GPO_0$ available. On readback, the value read is the value written, not necessarily the voltage level. |
| 35 | 7-0 | $0_{7-0}$ | **Reserved:** Always set these bits to 0. Any other value will place the FS400 into a manufacturing test mode. |

**Note:** When the internal microprocessor is running, $GPO_0$ is under the control of the microprocessor (see CCR_FLT_FREQ in section 4.5.40).

In order to modify the GPO register, one must first change the Configuration Command Register CCR_GPO at CCR address 0x20. After the CCR_GPO register is programmed, the exact same GPO value must be programmed into the General Purpose Outputs register (see above).

Bit 0 of the GPO register is always reserved. When programming the CCR_GPO register it is ok to write an 8-bit value. The system will ignore bit 0. When programming the General Purpose Outputs register (see above), first read the register to get the current state of bit 0, change bits 1-7 as desired, then write the new value into the register.

## 4.5.26 SCR – Software Control Register

### Software Control Register Low (60)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ENICTBL | RSTCALC | RSTICAL | RSTVMODE | CLRMCC | OCAL | ENABLE | RESET |

### Software Control Register High (61)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $SFLK_3$ | $SFLK_2$ | $SFLK_1$ | $SFLK_0$ | MICALNEW | CNTRSML | DSVMDET | DSICAL |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 60 | 0 | RESET | 0: Normal operation<br>1: Force software reset<br>The internal micro-controller clears this bit. |
| 60 | 1 | ENABLE | 0: Disable internal micro-controller<br>1: Enable internal micro-controller<br>The internal micro-controller will not access any FS400 SIO registers while it is disabled. |
| 60 | 2 | OCAL | 0: Normal operation.<br>1: Turns on the output calibration image (color bars). |
| 60 | 3 | CLRMCC | 0: Leave Mode Change Complete (MCC) status bit unchanged<br>1: Clear MCC (the internal micro-controller clears this bit). |
| 60 | 4 | RSTVMODE | 0: Normal operation<br>1: Reset video mode. Forces the internal micro-controller to re-determine the size and speed of the current video mode and to re-acquire the input calibration.<br>The internal micro-controller clears this bit. |
| 60 | 5 | RSTICAL | 0: Normal operation<br>1: Resets the input calibration. This forces the internal micro-controller to re-acquire the input calibration. If input calibration table is enabled (ENICTBL = 1), then the table entry for this video mode is also reset.<br>The internal micro-controller clears this bit. |
| 60 | 6 | RSTCALC | 0: Normal operation<br>1: Forces the internal micro-controller to re-calculate the low level FS400 registers for the current mode. This does not affect the video mode or input calibration.<br>The internal micro-controller clears this bit. |
| 60 | 7 | ENICTBL | 0: Input calibration is not stored in the input calibration table.<br>1: Input calibration is stored in the input calibration table. |
| 61 | 0 | DSICAL | 0: Normal operation<br>1: Disables the input calibration processing.<br>DSICAL is used to bypass the internal micro-controller's input calibration sampling algorithm.<br>NOTE: This bit is used only when an external processor wants to bypass the internal micro-controller's input calibration algorithm. It should be set to 0 for most applications. |
| 61 | 1 | DSVMDET | 0: Normal operation<br>1: Disable video mode detect. The internal micro-controller will not sample IHC and IVC to detect a video mode change.<br>NOTE: This bit is used only when an external processor wants to bypass the internal micro-controller's video mode detect algorithm. It |

| | | | |
|---|---|---|---|
| | | | should be set to 0 for most applications. |
| 61 | 2 | CNTRSML | 0: Small input images are scaled horizontally to fill the output device.<br>1: Small input images are centered on the output device. |
| 61 | 3 | MICALNEW | 0: Normal operation<br>1: There is new manual input calibration information.<br>The internal micro-controller clears this bit. |
| 61 | 7-4 | $SFLK_{3-0}$ | Adjusts the flicker filter from 0 (no flicker filter) to 8 (maximum flicker filter). SFLK is different from the hardware register FLK. SFLK is normalized for compression. |

**Reset Value:** 0x8402

## 4.5.27 SSR – Software Status Register

### Software Status Register Low (62)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | HSYNCACT | VSYNCACT | ICALAQRD | MCC | MCS | ERROR | READY |

### Software Status Register High (63)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $CHKSUM_7$ | $CHKSUM_6$ | $CHKSUM_5$ | $CHKSUM_4$ | $CHKSUM_3$ | $CHKSUM_2$ | $CHKSUM_1$ | $CHKSUM_0$ |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 62 | 0 | READY | 0: Internal controller is initializing after reset<br>1: Internal controller is ready |
| 62 | 1 | ERROR | 0: The video mode parameters are all within range<br>1: A video mode parameter is out of range and the mode cannot be calculated. The following conditions cause an error:<br>IHC < 32<br>IVC < 32<br>HAT < 20%<br>VAT < 20%<br>TVP < 128<br>TVL < 128 |
| 62 | 2 | MCS | Mode Change Start<br>0: Video mode change is not in progress.<br>1: Video mode change is in progress.<br>The internal micro-controller sets this bit to 1 after it detects a change in the size or speed of the input video mode. The controller clears this bit after it has completed the calculations for the new mode. |
| 62 | 3 | MCC | Mode Change Complete<br>0: Video mode change is not complete<br>1: Video mode change is complete<br>NOTE: This bit is set when a mode change is complete and cleared by writing a 1 to the CLRMCC bit in the SCR |
| 62 | 4 | ICALAQRD | Input Calibration Acquired<br>0: The internal micro-controller has not yet read an acceptable set of values from the SHV, EHV, SVV, and EVV registers and is using the default values for HAT, SHV, VAT, and SVV.<br>1: The internal micro-controller has read acceptable input calibration values and is using them. |
| 62 | 5 | VSYNCSACT | 0: No VSync activity during last 100ms interval.<br>1: VSync activity during last 100ms interval was detected. |
| 62 | 6 | HSYNCACT | 0: No HSync activity during last 100ms interval.<br>1: HSync activity during last 100ms interval was detected. |
| 63 | 7-0 | $CHKSUM_{7-0}$ | Internal micro-controller ROM and RAM checksum.<br>NOTE: The checksum for version 30 is 0xAF. |

**NOTE:** It is possible to write to this register, however doing so will produce unpredictable results.

## 4.5.28 HCRS – Hardware Control Register Shadow

### Hardware Control Register Shadow Low (64)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PEDSTL | CBP | LNTCH | YCOFF | COMPOFF | ADCOFF | CLKOFF | 0 |

### Hardware Control Register Shadow High (65)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $OFMT_1$ | $OFMT_0$ | PAL_NTSC | PROG_INT | $TV\_CLK_1$ | $TV\_CLK_0$ |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 64 | 1 | CLKOFF | 0: FS400 clocks powered on.<br>1: FS400 clocks powered off. |
| 64 | 2 | ADCOFF | 0: A/D converter powered on.<br>1: A/D converter powered off. |
| 64 | 3 | COMPOFF | 0: Composite D/A converter powered on.<br>1: Composite D/A converter powered off. |
| 64 | 4 | YCOFF | 0: Y/C & Chroma D/A converter powered on.<br>1: Y/C & Chroma D/A converter powered off. |
| 64 | 5 | LNTCH | 0: Bypass luminance notch filter.<br>1: Insert luminance notch filter. |
| 64 | 6 | CBP | 0: Bypass chroma bandpass filter.<br>1: Insert chroma bandpass filter. |
| 64 | 7 | PEDSTL | 0: Bypass pedestal<br>1: Insert US NTSC black pedestal |
| 65 | 1-0 | $TV\_CLK_{1-0}$ | TV Clock Select<br>00 = $4F_{SC}$ crystal or oscillator<br>01 = non-subcarrier crystal or oscillator<br>10 = $8F_{SC}$ crystal or oscillator |
| 65 | 2 | PROG_INT | 0 = Interlaced Output<br>1 = Progressive Output |
| 65 | 3 | PAL_NTSC | 0 = NTSC<br>1 = PAL |
| 65 | 5-4 | $OFMT_{1-0}$ | Output Format<br>00 = Y/C and Composite Video<br>01 = RGB<br>10 = YUV<br>11 = YUV |

**Reset Value:** 0x0080

**Typical Output Mode Settings:**

| TV Mode | TV_CLK | PROG_INT | PAL_NTSC |
|---|---|---|---|
| NTSC | 0 | 0 | 0 |
| PAL | 0 | 0 | 1 |
| VGA | 2 | 1 | 0 |
| SVGA | 2 | 1 | 1 |
| 100Hz PAL | 1 | 0 | 1 |
| Progressive NTSC (720x480) | 1 | 1 | 0 |
| Super NTSC (1280x480) | 2 | 0 | 0 |
| Super PAL (1280x575) | 2 | 0 | 1 |

NOTE: Any change to TV_CLK, PROG_INT, or PAL_NTSC causes the internal micro-controller to initialize the HOHOS, HOVOS, TVL, TVP, and CCR_FREQUENCY registers to defaults appropriate to the new output mode.

## 4.5.29 HCRES – Hardware Control Register Extended Shadow

### Hardware Control Register Extended Shadow Low (66)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | VGAINTDET | 0 | 0 | FREEZE | RGBGAIN | ZOOM |

### Hardware Control Register Extended Shadow High (67)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $BIPGEN_2$ | $BIPGEN_1$ | $BIPGEN_0$ |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 66 | 0 | ZOOM | 0: Normal operation<br>1: Zoom mode |
| 66 | 1 | RGBGAIN | 0: 1x input gain<br>1: 1.43x input gain |
| 66 | 2 | FREEZE | 0: Continuous update<br>1: Freeze frame |
| 66 | 5 | VGAINTDET | 0: Ignore interlaced input.<br>1: Detect interlaced input. |
| 67 | 2-0 | BIPGEN | Built-in Pattern Generator:<br>000 = Invalid<br>001 = All Waveforms<br>010 = 100% Color Bars<br>011 = 75% Color Bars<br>100 = Cross Hatch<br>101 = Red Gradient<br>110 = Green Gradient<br>111 = Blue Gradient<br>NOTE: Only has effect when OCAL bit in SCR register is set. |

**Reset Value:** 0x0320

## 4.5.30 HPO – Horizontal Position Offset

### Horizontal Position Offset Low (68)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HPO_7$ | $HPO_6$ | $HPO_5$ | $HPO_4$ | $HPO_3$ | $HPO_2$ | $HPO_1$ | $HPO_0$ |

### Horizontal Position Offset High (69)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HPO_{15}$ | $HPO_{14}$ | $HPO_{13}$ | $HPO_{12}$ | $HPO_{11}$ | $HPO_{10}$ | $HPO_9$ | $HPO_8$ |

**Description:**
This offsets the horizontal position of the input capture window. A positive number moves the capture widow to the right and a negative number moves the input capture window to the left.

HPO only has meaning if the ZOOM bit in HCRES is 0.

If the value written to HPO is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the HPO register. An external processor can then read HPO to determine if the last value written to HPO was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:** {-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.31 VPO – Vertical Position Offset

### Vertical Position Offset Low (6A)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VPO_7$ | $VPO_6$ | $VPO_5$ | $VPO_4$ | $VPO_3$ | $VPO_2$ | $VPO_1$ | $VPO_0$ |

### Vertical Position Offset High (6B)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VPO_{15}$ | $VPO_{14}$ | $VPO_{13}$ | $VPO_{12}$ | $VPO_{11}$ | $VPO_{10}$ | $VPO_9$ | $VPO_8$ |

**Description:**
This offsets the vertical position of the input capture window. A positive number moves the capture widow down and a negative number moves the input capture window up.

VPO only has meaning if the ZOOM bit in HCRES is 0.

If the value written to VPO is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the VPO register. An external processor can then read VPO to determine if the last value written to VPO was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:**
{-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.32 HSS – Horizontal Scale Step

### Horizontal Scale Step Low (6C)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HSS_7$ | $HSS_6$ | $HSS_5$ | $HSS_4$ | $HSS_3$ | $HSS_2$ | $HSS_1$ | $HSS_0$ |

### Horizontal Scale Step High (6D)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HSS_{15}$ | $HSS_{14}$ | $HSS_{13}$ | $HSS_{12}$ | $HSS_{11}$ | $HSS_{10}$ | $HSS_9$ | $HSS_8$ |

**Description:**
This offsets the horizontal scale from its nominal value. A positive number stretches the image and a negative number compresses the image.

If the value written to HSS is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the HSS register. An external processor can then read HSS to determine if the last value written to HSS was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:**
{-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.33 VSS – Vertical Scale Step

### Vertical Scale Step Low (6E)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VSS_7$ | $VSS_6$ | $VSS_5$ | $VSS_4$ | $VSS_3$ | $VSS_2$ | $VSS_1$ | $VSS_0$ |

### Vertical Scale Step High (6F)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VSS_{15}$ | $VSS_{14}$ | $VSS_{13}$ | $VSS_{12}$ | $VSS_{11}$ | $VSS_{10}$ | $VSS_9$ | $VSS_8$ |

**Description:**
This offsets the vertical scale from its nominal value. A positive number stretches the image and a negative number compresses the image.

If the value written to VSS is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the VSS register. An external processor can then read VSS to determine if the last value written to VSS was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:**
{-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.34 HPP – Horizontal Pan Position

### Horizontal Pan Position Low (70)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HPP_7$ | $HPP_6$ | $HPP_5$ | $HPP_4$ | $HPP_3$ | $HPP_2$ | $HPP_1$ | $HPP_0$ |

### Horizontal Pan Position High (71)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HPP_{15}$ | $HPP_{14}$ | $HPP_{13}$ | $HPP_{12}$ | $HPP_{11}$ | $HPP_{10}$ | $HPP_9$ | $HPP_8$ |

**Description:**
This controls the horizontal pan window position. A value of 0 will set the pan window to the left edge of the input image. As HPP increases, the pan window will move to the right.

HPP only has meaning if the ZOOM bit in HCRES is 1.

If the value written to HPP is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the HPP register. An external processor can then read HPP to determine if the last value written to HPP was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:**
{-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.35 VPP – Vertical Pan Position

### Vertical Pan Position Low (72)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VPP_7$ | $VPP_6$ | $VPP_5$ | $VPP_4$ | $VPP_3$ | $VPP_2$ | $VPP_1$ | $VPP_0$ |

### Vertical Pan Position High (73)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $VPP_{15}$ | $VPP_{14}$ | $VPP_{13}$ | $VPP_{12}$ | $VPP_{11}$ | $VPP_{10}$ | $VPP_9$ | $VPP_8$ |

**Description:**
This controls the vertical pan window position. A value of 0 will set the pan window to the top edge of the input image. As VPP increases, the pan window will move down.

VPP only has meaning if the ZOOM bit in HCRES is 1.

If the value written to VPP is beyond the internally calculated limit, the maximum (or minimum in case of a negative number) will be written back to the VPP register. An external processor can then read VPP to determine if the last value written to VPP was beyond the limit.

NOTE: It can take up to 25ms for the internal micro-controller calculates the limits for HPO, VPO, HSS, VSS, HPP, and VPP. These registers should not be read until 10ms after writing to any of these registers.

**Range:**
{-32768 : +32767} 2's Complement

**Reset Value:** 0x0000

## 4.5.36 TVP – TV Pixels

### TV Pixels Low (74)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $TVP_7$ | $TVP_6$ | $TVP_5$ | $TVP_4$ | $TVP_3$ | $TVP_2$ | $TVP_1$ | $TVP_0$ |

### TV Pixels High (75)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $TVP_{15}$ | $TVP_{14}$ | $TVP_{13}$ | $TVP_{12}$ | $TVP_{11}$ | $TVP_{10}$ | $TVP_9$ | $TVP_8$ |

**Description:**
Tells the internal controller the total number of usable pixels on the output device in a horizontal line.

If the value written to TVP is beyond the limit, the new mode will not be calculated and the ERROR bit in SSR will be set.

Any change to TV_CLK, PROG_INT, or PAL_NTSC fields in the HCRS register causes the internal micro-controller to initialize TVP register to a default value appropriate to the new output mode.

| TV Mode | TVP |
|---|---|
| NTSC | 658 |
| PAL | 820 |
| VGA | 640 |
| SVGA | 800 |
| 100Hz PAL | 908 |
| Progressive NTSC (720x480) | 658 |
| Super NTSC (1280x480) | 914 |
| Super PAL (1280x575) | 914 |

**Range:**
{+128 : +2047}

**Reset Value:** 0x0292 (658.)

## 4.5.37 TVL – TV Lines

### TV Lines Low (76)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $TVL_7$ | $TVL_6$ | $TVL_5$ | $TVL_4$ | $TVL_3$ | $TVL_2$ | $TVL_1$ | $TVL_0$ |

### TV Lines High (77)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $TVL_{15}$ | $TVL_{14}$ | $TVL_{13}$ | $TVL_{12}$ | $TVL_{11}$ | $TVL_{10}$ | $TVL_9$ | $TVL_8$ |

**Description:**
Tells the internal controller the total number of usable lines on the output device.

If the value written to TVL is beyond the limit, the new mode will not be calculated and the ERROR bit in SSR will be set.

Any change to TV_CLK, PROG_INT, or PAL_NTSC fields in the HCRS register causes the internal micro-controller to initialize TVL register to a default value appropriate to the new output mode.

| TV Mode | TVL |
|---|---|
| NTSC | 412 |
| PAL | 496 |
| VGA | 480 |
| SVGA | 600 |
| 100Hz PAL | 540 |
| Progressive NTSC (720x480) | 412 |
| Super NTSC (1280x480) | 412 |
| Super PAL (1280x575) | 496 |

**Range:**
{+128 : +2047}

**Reset Value:** 0x019C (412.)

## 4.5.38 CCR – Configuration Command Register

### Configuration Command Register Low (78)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $ADDR_7$ | $ADDR_6$ | $ADDR_5$ | $ADDR_4$ | $ADDR_3$ | $ADDR_2$ | $ADDR_1$ | $ADDR_0$ |

### Configuration Command Register High (79)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $CMD_3$ | $CMD_2$ | $CMD_1$ | $CMD_0$ |

| Reg | Bit# | Bit Name | Description |
|---|---|---|---|
| 78 | 7-0 | $ADDR_{7-0}$ | Parameter Address<br>This field contains the address of the configuration parameter to be accessed. |
| 79 | 3-0 | $CMD_{3-0}$ | Command<br><br>0000 Command complete. The internal controller writes this.<br>0001 Read from the configuration parameter.<br>After the command is complete CDR contains the data.<br>0010 Write to the configuration parameter.<br>The CDR must contain the data to write before CCR is written. |

The CCR and CDR registers are used to access constants and variables inside the internal micro-controller. The definitions of these parameters are found in section 4.5.40.

To read from a configuration parameter, write 0001 to the command field and the configuration parameter address in the address field of the CCR. These values must be written with one operation. Next, poll the CCR until it becomes zero. Be sure to include a delay between polling reads to prevent stopping the internal CPU from having access to the SIO registers. Then read the CDR to retrieve the value of the specified configuration parameter.

To write to a configuration parameter, write the data to the CDR. Next, write 0010 to the command field and the configuration parameter address in the address field of the CCR. When the CCR because zero, the write operation is complete. Be sure to include a delay between polling reads to prevent stopping the internal CPU from having access to the SIO registers.

**Reset Value:** 0x0000

### 4.5.39 CDR – Configuration Data Register

## Configuration Data Register Low (7A)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CDR | $CDR_6$ | $CDR_5$ | $CDR_4$ | $CDR_3$ | $CDR_2$ | $CDR_1$ | $CDR_0$ |

## Configuration Data Register High (7B)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CDR | $CDR_{14}$ | $CDR_{13}$ | $CDR_{12}$ | $CDR_{11}$ | $CDR_{10}$ | $CDR_9$ | $CDR_8$ |

**Description:**
Contains the value of the configuration parameter after a CDR read command is complete.
Contains the new value of a configuration parameter before a CDR write is executed.

**Reset Value:** 0x0000

## 4.5.40 HOHOS – Hardware Output Horizontal Offset Shadow

### Hardware Output Horizontal Offset Shadow Low (7C)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HOHOS_7$ | $HOHOS_6$ | $HOHOS_5$ | $HOHOS_4$ | $HOHOS_3$ | $HOHOS_2$ | $HOHOS_1$ | $HOHOS_0$ |

### Hardware Output Horizontal Offset Shadow High (7D)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HOHOS_{15}$ | $HOHOS_{14}$ | $HOHOS_{13}$ | $HOHOS_{12}$ | $HOHOS_{11}$ | $HOHOS_{10}$ | $HOHOS_9$ | $HOHOS_8$ |

**Description:**
Horizontal displacement of the outgoing active video area image in pixels from the leading edge of outgoing horizontal sync.

Any change to TV_CLK, PROG_INT, or PAL_NTSC fields in the HCRS register causes the internal micro-controller to initialize HOHOS register to a default value appropriate to the new output mode.

| TV Mode | HOHOS |
|---|---|
| NTSC | 178 |
| PAL | 230 |
| VGA | 132 |
| SVGA | 176 |
| 100Hz PAL | 180 |
| Progressive NTSC (720x480) | 178 |
| Super NTSC (1280x480) | 290 |
| Super PAL (1280x575) | 290 |

**Reset Value:** 0x00B2 (178.)

## 4.5.41 HOVOS – Hardware Output Vertical Offset Shadow

### Hardware Output Vertical Offset Shadow Low (7E)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HOVOS_7$ | $HOVOS_6$ | $HOVOS_5$ | $HOVOS_4$ | $HOVOS_3$ | $HOVOS_2$ | $HOVOS_1$ | $HOVOS_0$ |

### Hardware Output Vertical Offset Shadow High (7F)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| $HOVOS_{15}$ | $HOVOS_{14}$ | $HOVOS_{13}$ | $HOVOS_{12}$ | $HOVOS_{11}$ | $HOVOS_{10}$ | $HOVOS_9$ | $HOVOS_8$ |

**Description:**
Vertical displacement of the outgoing active video area in lines from the beginning of the equalization pulses: line 1/263 for NTSC; line 311/623 for PAL.

Any change to TV_CLK, PROG_INT, or PAL_NTSC fields in the HCRS register causes the internal micro-controller to initialize HOVOS register to a default value appropriate to the new output mode.

| TV Mode | HOVOS |
|---|---|
| NTSC | 38 |
| PAL | 44 |
| VGA | 40 |
| SVGA | 36 |
| 100Hz PAL | 38 |
| Progressive NTSC (720x480) | 38 |
| Super NTSC (1280x480) | 38 |
| Super PAL (1280x575) | 44 |

**Reset Value:** 0x0026 (38.)

## 4.6 Configuration Values

| Name | Addr | R/W | Size | Reset Value | Description |
|---|---|---|---|---|---|
| CCR_VERSION | 0x00 | R | 8 | 30 | The internal micro-controller's version number. (First silicon = 30, 4/99) |
| CCR_FREQUENCY | 0x01 | R/W | 8 | 112 | The speed of the FS400 expressed as: (Clock Speed / 14.31mHz) * 32<br>The reset value assumes the FS400 clock speed is 50mHz. |
| CCR_FLT_FREQ | 0x02 | R/W | 8 | 63 | The analog filter frequency threshold. This value is expressed as :<br>(Threshold Clock Speed / 14.31mHz) * 32<br>The reset value assumes a 28mHz threshold. When the IHS register exceeds this frequency, $GPO_0$ is set to a 1. |
| CCR_AVT | 0x03 | R/W | 8 | 64 | The value the internal micro-controller writes into the FS400 AVT register when RGBGAIN is off. |
| CCR_AVT_RGBGAIN | 0x04 | R/W | 8 | 128 | The value the internal micro-controller writes into the FS400 AVT register when RGBGAIN is on. |
| CCR_MIN_HAT | 0x05 | R/W | 16 | 32768 | Minimum HAT (Horizontal Active) value allowed. Values of HAT lower than this will cause the input calibration algorithm to use the default HAT value (DEF_HAT).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MAX_HAT | 0x06 | R/W | 16 | 64881 | Maximum HAT (Horizontal Active) value allowed. Values of HAT higher than this will cause the input calibration algorithm to use the default HAT value (DEF_HAT).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_DEF_HAT | 0x07 | R/W | 16 | 55706 | The default HAT (Horizontal Active) value is used when the internal micro-controller has not been able to acquire a valid set of horizontal input calibration values.<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MIN_SHV | 0x08 | R/W | 16 | 1311 | Minimum SHV (Start Horizontal Video) value allowed. Values of SHV lower than this will cause the input calibration algorithm to use the default SHV value (DEF_SHV).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MAX_SHV | 0x09 | R/W | 16 | 32767 | Maximum SHV (Start Horizontal Video) value allowed. Values of SHV higher than this will cause the input calibration algorithm to use the default SHV value (DEF_SHV).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_DEF_SHV | 0x0A | R/W | 16 | 9830 | The default SHV (Start Horizontal Video) value is used when the internal micro-controller has not been able to acquire a valid set of horizontal input calibration |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | values.<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MIN_VAT | 0x0B | R/W | 16 | 32768 | Minimum VAT (Vertical Active) value allowed. Values of VAT lower than this will cause the input calibration algorithm to use the default VAT value (DEF_VAT).<br>Expressed as a percentage vertical total * 65536. |
| CCR_MAX_VAT | 0x0C | R/W | 16 | 64881 | Maximum VAT (Vertical Active) value allowed. Values of HAT higher than this will cause the input calibration algorithm to use the default HAT value (DEF_HAT).<br>Expressed as a percentage vertical total * 65536. |
| CCR_DEF_VAT | 0x0D | R/W | 16 | 63898 | The default VAT (Vertical Active) value is used when the internal micro-controller has not been able to acquire a valid set of vertical input calibration values.<br>Expressed as a percentage vertical total * 65536. |
| CCR_MIN_SVV | 0x0E | R/W | 16 | 1311 | Minimum SVV (Start Vertical Video) value allowed. Values of SVV lower than this will cause the input calibration algorithm to use the default SVV value (DEF_SVV).<br>Expressed as a percentage vertical total * 65536. |
| CCR_MAX_SVV | 0x0F | R/W | 16 | 32767 | Maximum SVV (Start Vertical Video) value allowed. Values of SVV higher than this will cause the input calibration algorithm to use the default SVV value (DEF_SVV).<br>Expressed as a percentage vertical total * 65536. |
| CCR_DEF_SVV | 0x10 | R/W | 16 | 1638 | The default SVV (Start Vertical Video) value is used when the internal micro-controller has not been able to acquire a valid set of vertical input calibration values.<br>Expressed as a percentage vertical total * 65536. |
| CCR_COUNTER | 0x11 | R/W | 8 | 0 | Reads the internal micro-controller's counter. The counter runs at a frequency of OSC1 / 786432. If OSC1 is 14.31mHz, the COUNTER frequency is 18.20Hz. |
| CCR_PIXEL_SIZE | 0x12 | R/W | 16 | | The size of 1 horizontal pixel expressed as a percentage of the horizontal total * 65536. This value is useful in computing offset of the manual input calibration values. |
| CCR_LINE_SIZE | 0x13 | R/W | 16 | 0 | The size of 1 vertical line expressed as a percentage of the vertical total * 65536. This value is useful in computing offset of the manual input calibration values. |
| CCR_MICAL_HAT | 0x14 | R/W | 16 | 55706 | The HAT (Horizontal Active) value is used when the internal micro-controller is in manual input calibration mode (DSICAL bit in SCR is set).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MICAL_VAT | 0x15 | R/W | 16 | 63898 | The VAT (Vertical Active) value is used |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | when the internal micro-controller is in manual input calibration mode (DSICAL bit in SCR is set).<br>Expressed as a percentage of the vertical total * 65536. |
| CCR_MICAL_SHV | 0x16 | R/W | 16 | 9830 | The SHV (Start Horizontal Video) value is used when the internal micro-controller is in manual input calibration mode (DSICAL bit in SCR is set).<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_MICAL_SVV | 0x17 | R/W | 16 | 1638 | The SVV (Start Vertical Video) value is used when the internal micro-controller is in manual input calibration mode (DSICAL bit in SCR is set).<br>Expressed as a percentage of the vertical total * 65536. |
| CCR_HSS_MIN | 0x18 | R/W | 16 | | Minimum value of HSS for the current video mode. |
| CCR_HSS_MAX | 0x19 | R/W | 16 | | Maximum value of HSS for the current video mode. |
| CCR_VSS_MIN | 0x1A | R/W | 16 | | Minimum value of VSS for the current video mode. |
| CCR_VSS_MAX | 0x1B | R/W | 16 | | Maximum value of VSS for the current video mode. |
| CCR_HPX_MIN | 0x1C | R/W | 16 | | In zoom mode (ZOOM bit in HCRES is set), contains the minimum value of HPP for the current video mode. When not in zoom mode, contains the minimum value of HPO for the current mode. |
| CCR_HPX_MAX | 0x1D | R/W | 16 | | In zoom mode (ZOOM bit in HCRES is set), contains the maximum value of HPP for the current video mode. When not in zoom mode, contains the maximum value of HPO for the current mode. |
| CCR_VPX_MIN | 0x1E | R/W | 16 | | In zoom mode (ZOOM bit in HCRES is set), contains the minimum value of VPP for the current video mode. When not in zoom mode, contains the minimum value of VPO for the current mode. |
| CCR_VPX_MAX | 0x1F | R/W | 16 | | In zoom mode (ZOOM bit in HCRES is set), contains the maximum value of VPP for the current video mode. When not in zoom mode, contains the maximum value of VPO for the current mode. |
| CCR_GPO | 0x20 | R/W | 16 | 0 or 1 | The value written to the FS400 GPO register. The internal micro-controller controls the state of bit 0. All other bits are available for general-purpose output. |
| CCR_IHC | 0x21 | R/W | 16 | | The internal micro-controller's version of the FS400 IHC register. |
| CCR_IVC | 0x22 | R/W | 16 | | The internal micro-controller's version of the FS400 IVC register. |
| CCR_HAT | 0x23 | R/W | 16 | | The current HAT value. This can either be the default HAT (DEF_HAT) or a HAT calculated from the SHV and EHV registers.<br>Expressed as a percentage of the |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | horizontal total * 65536. |
| CCR_SHV | 0x24 | R/W | 16 | | The current SHV value. This can either be the default SHV (DEF_SHV) or a SHV calculated from the SHV and EHV registers.<br>Expressed as a percentage of the horizontal total * 65536. |
| CCR_VAT | 0x25 | R/W | 16 | | The current VAT value. This can either be the default VAT (DEF_VAT) or a VAT calculated from the SVV and EVV registers.<br>Expressed as a percentage of the vertical total * 65536. |
| CCR_SVV | 0x26 | R/W | 16 | | The current SVV value. This can either be the default SVV (DEF_SVV) or a SVV calculated from the SVV and EVV registers.<br>Expressed as a percentage of the vertical total * 65536. |
| CCR_ICAL_TABLE_0_IVC | 0x27 | R/W | 16 | 0 | The IVC value that this table entry represents. |
| CCR_ICAL_TABLE_0_IHC | 0x28 | R/W | 16 | 0 | The IHC value that this table entry represents. |
| CCR_ICAL_TABLE_0_HAT | 0x29 | R/W | 16 | 0 | The accumulated HAT value used for this video mode. |
| CCR_ICAL_TABLE_0_SHV | 0x2A | R/W | 16 | 0 | The accumulated SHV value used for this video mode. |
| CCR_ICAL_TABLE_0_VAT | 0x2B | R/W | 16 | 0 | The accumulated VAT value used for this video mode. |
| CCR_ICAL_TABLE_0_SVV | 0x2C | R/W | 16 | 0 | The accumulated SVV value used for this video mode. |
| CCR_ICAL_TABLE_1_IVC | 0x2D | R/W | 16 | 0 | The IVC value that this table entry represents. |
| CCR_ICAL_TABLE_1_IHC | 0x2E | R/W | 16 | 0 | The IHC value that this table entry represents. |
| CCR_ICAL_TABLE_1_HAT | 0x2F | R/W | 16 | 0 | The accumulated HAT value used for this video mode. |
| CCR_ICAL_TABLE_1_SHV | 0x30 | R/W | 16 | 0 | The accumulated SHV value used for this video mode. |
| CCR_ICAL_TABLE_1_VAT | 0x31 | R/W | 16 | 0 | The accumulated VAT value used for this video mode. |
| CCR_ICAL_TABLE_1_SVV | 0x32 | R/W | 16 | 0 | The accumulated SVV value used for this video mode. |
| CCR_ICAL_TABLE_2_IVC | 0x33 | R/W | 16 | 0 | The IVC value that this table entry represents. |
| CCR_ICAL_TABLE_2_IHC | 0x34 | R/W | 16 | 0 | The IHC value that this table entry represents. |
| CCR_ICAL_TABLE_2_HAT | 0x35 | R/W | 16 | 0 | The accumulated HAT value used for this video mode. |
| CCR_ICAL_TABLE_2_SHV | 0x36 | R/W | 16 | 0 | The accumulated SHV value used for this video mode. |
| CCR_ICAL_TABLE_2_VAT | 0x37 | R/W | 16 | 0 | The accumulated VAT value used for this video mode. |
| CCR_ICAL_TABLE_2_SVV | 0x38 | R/W | 16 | 0 | The accumulated SVV value used for this video mode. |
| CCR_ICAL_TABLE_3_IVC | 0x39 | R/W | 16 | 0 | The IVC value that this table entry represents. |
| CCR_ICAL_TABLE_3_IHC | 0x3A | R/W | 16 | 0 | The IHC value that this table entry |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | represents. |
| CCR_ICAL_TABLE_3_HAT | 0x3B | R/W | 16 | 0 | The accumulated HAT value used for this video mode. |
| CCR_ICAL_TABLE_3_SHV | 0x3C | R/W | 16 | 0 | The accumulated SHV value used for this video mode. |
| CCR_ICAL_TABLE_3_VAT | 0x3D | R/W | 16 | 0 | The accumulated VAT value used for this video mode. |
| CCR_ICAL_TABLE_3_SVV | 0x3E | R/W | 16 | 0 | The accumulated SVV value used for this video mode. |

# 5. Functional Description

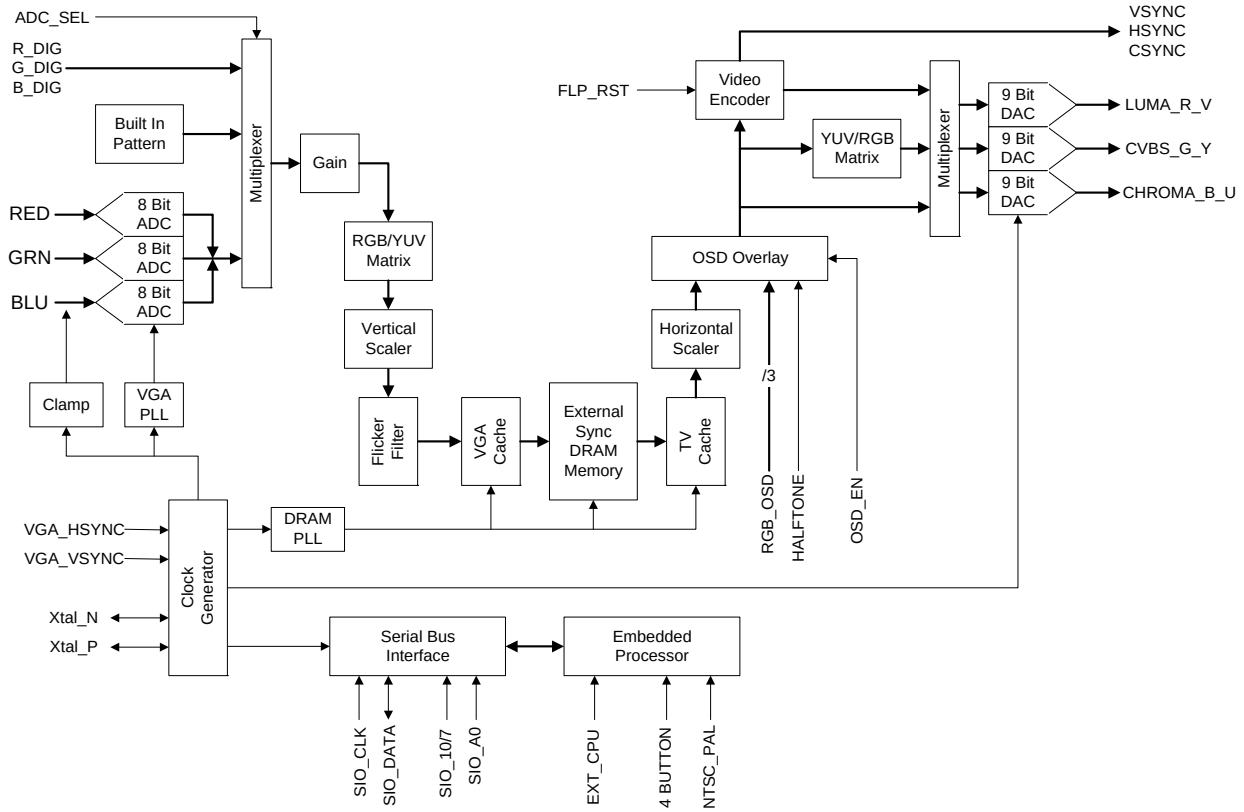



**Figure 5. Functional Block Diagram**
**($RGB_{7-0}$ and A/D_SEL\ omitted on FS401)**

Details of how to connect and setup the FS400 are included in this section. Overall design principles are in the *Architectural Overview* section. Operation of the FS400 is divided into four sections:

- Capture Engine
- Frame Store Controller
- Encoder Engine
- Serial Control Port
- Embedded Microprocessor

## 5.1 Capture Engine

A/D Converters, Built In Pattern Generator, RGB/YUV Matrix, Vertical Scaler and Flicker Filter comprise the Capture Engine.

### 5.1.1 Timing and Control

Timing of the Capture Engine is derived from the Input Control Block which contains a series of counters and decoders synchronized to the A/D sample clock, ADCK. ADCK is derived from a phase locked loop referenced to the leading edge of horizontal sync.

Sync polarity is auto-detected by sensing the leading edge of horizontal sync HS_IN and vertical sync, VS_IN. This edge is the reference for the phase-locked loop tracking the horizontal pixel count and the vertical line counter.

Registers that interface with the Capture Timing and Control block are:

1. IHS
4. SHV (read only)
7. EVV (read only)

2. IHC (read only)
3. IVC (read only)
5. EHV (read only)
6. SVV (read only)
8. AVT

Capture Control also coordinates hand off of data to the Frame Store Controller.

Input Horizontal Samples, IHS is the 11-bit terminal count of the number of pixels per horizontal line between sync pulses. IHS is the value programmed into the ADCK phase-locked-loop. If, for example there are to be 800 samples per incoming line, then IHS must be programmed to be 799.

Input Horizontal Count, IHC is the number of encoder clock pulses on the selected OSC clock that occur between horizontal sync pulses. This count is stored in the IHC register that can be read via the serial bus for automatic detection of the format of incoming video.

Input Vertical Count, IVC is the 12-bit terminal count of the number of lines that occur between the vertical sync pulses. This count is stored in the IVC register that can be read via the serial bus for automatic detection of the format of incoming video.

Start Horizontal Active VGA, SHV is the number of ADC clock pulses that occur from the start of the horizontal sync pulse to the detection of incoming active video. This is used to permit an accurate positioning of the incoming VGA on the TV.

End Horizontal Active VGA, EHV is the number of ADC clock pulses that occur from the start of the horizontal sync pulse to the end of detected incoming active video. This is used to permit an accurate positioning of the incoming VGA on the TV.

Start Vertical Active VGA, SVV is the number of lines that occur from the start of the vertical sync pulse to the detection of incoming active video. This is used to permit an accurate positioning of the incoming VGA on the TV.

End Vertical Active VGA, EVV is the number of lines that occur from the start of the vertical sync pulse to the end of detected incoming active video. This is used to permit an accurate positioning of the incoming VGA on the TV.

### 5.1.2 Clamps

Incoming RGB video signals must be AC coupled to the A/D converters. Preceding each A/D converter is an FET clamp switch which establishes the black reference level of each video signal by shorting the A/D converter input to ground when the clamp signal is active. Clamp timing is derived internally from the HS_IN input. A digital output, CLAMP_REF = H, when the clamp is active. CLAMP_REF and VGACLKDIV polarity are set by the VGAREFPOL bit in the CRE register.

### 5.1.3 Analog-to-Digital Converters

Bottom reference voltage of the A/D converters is ground. Top reference voltage, $V_T$ is a high impedance input that is applied via voltage followers to the reference ladder network of each A/D converter. $V_T$ must be de-coupled with a 0.1µF capacitor to ground.

$V_T$ can be derived from the internal reference voltage, $V_{REF}$ by splitting the resistor connected to $I_{REF}$ as described in the 5.3.5 Digital-to-Analog Converters section.

To avoid aliasing effects, incoming RGB video signals should be filtered by a low pass filter prior to the AC coupling capacitor. Filter cutout frequency should be set at half the highest expected sampling rate of the A/D converter clock, ADCK. A simple two element RC filter is adequate. 20MHz is typically used as a cut off frequency.

Phase of ADCK is set by the Command Register Extended VGACLKPOL bit. By flipping the phase of the sampling clock by 180°, the sampling points can be positioned closer to the center of incoming pixels. *Figure 6* shows optimum sampling of VGA pixels on the rising edge of the ADCLK signal, when synchronous sampling is chosen.

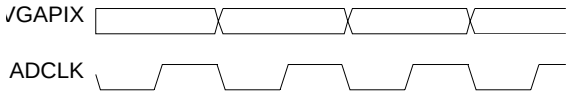



**Figure 6. FAZE Sets ADCK Sampling Edge on Incoming Pixels**

A/D converter power can be disconnected by setting the CR register ADCOFF bit.

### 5.1.4 24-bit Digital RGB Port (FS403 only)

Extra pins are included on the FS403 package to incorporate a 24-bit TTL compatible RGB input port. Either analog or digital inputs can be selected by the level on the EXADSEL pin. With the 24-bit RGB port enabled, incoming pixels may be derived directly from an external digital RGB source or from analog RGB via triple 8-bit A/D converters.

### 5.1.5 Built In Pattern Generator

BIPGEN in the CRE register selects one of several test patterns instead of the incoming video. If the “All Waveforms” option is set (001), the pattern looks as follows:

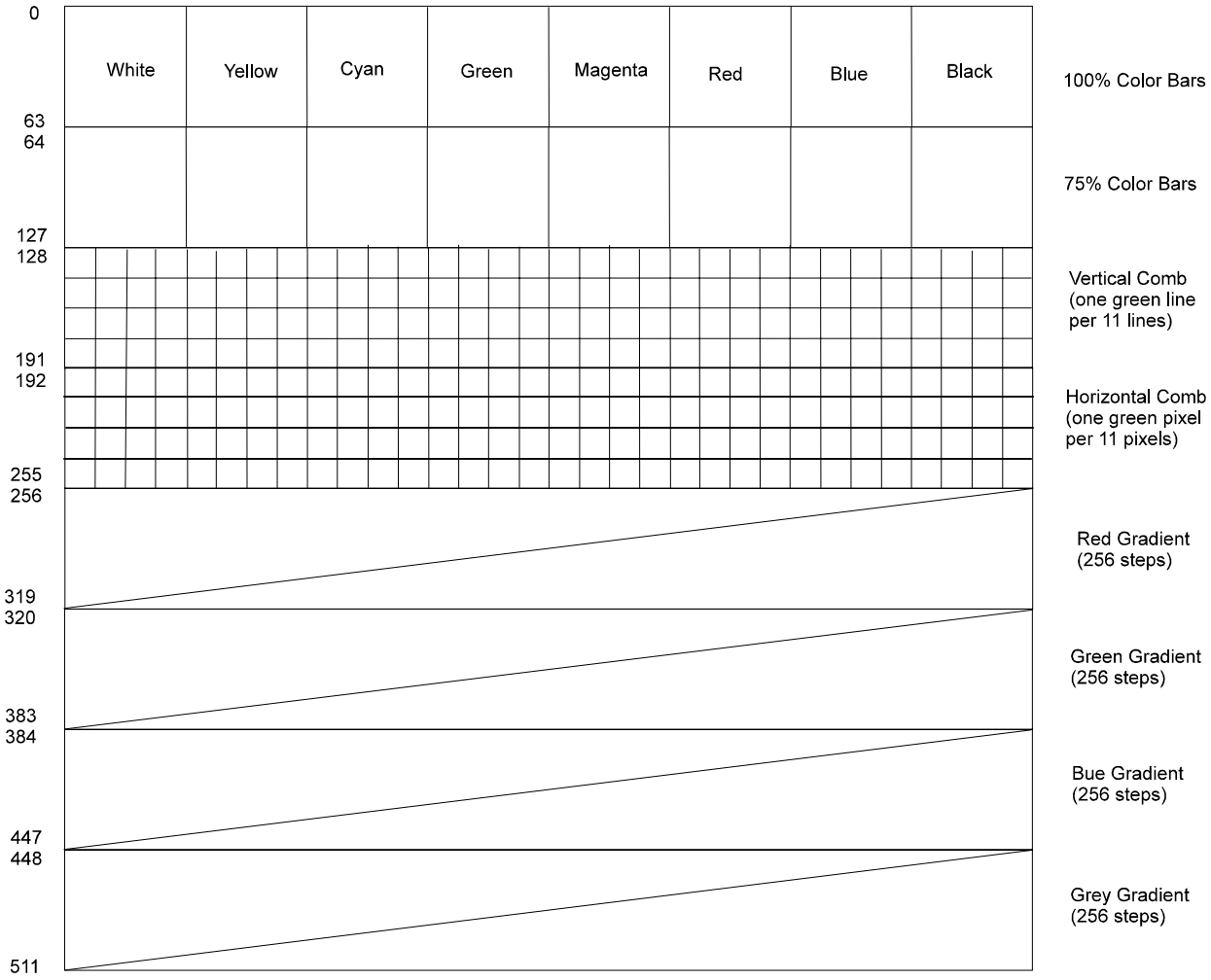



**Figure 7: BiPGEN Image**

The 512x512 chart is inserted into a 768x768 pixel frame, displaced 128 pixels from the left boundary and 128 pixels from the upper boundary. Color bars are each 64 pixels wide.

### 5.1.6 Digital RGB Multiplexer

The EXADSEL pin in conjunction with BIPGEN in the CRE register controls a triple 24-bit multiplexer that selects the source of RGB data to be supplied to the digital gain block. 24-bit RGB data can be accepted from:

- TTL compatible RGB input port
- A/D converter outputs
- BiPGen, the internal test pattern generator

### 5.1.7 RGB Gain

Following each A/D converter is a digital gain stage that allows the A/D converter to accommodate either 700 mV or 1000 mV RGB input video signals. Gain is set by the RGBGAIN bit of the Command Register. When using RGBGAIN, $V_{ADCREF}$ should be set to 1000 mV.

### 5.1.8 RGB/YUV Matrix

Pixels are converted from the 24-bit RGB format to the 24-bit YUV format by an RGB/YUV matrix. UV data is filtered and decimated prior to realignment with Y data forming a 16-bit YUV422 data stream.

### 5.1.9 Vertical Scaler

16-bit YUV422 data from the Transcoder is passed to the Vertical Scaler that operates on columns of pixels. Programming the VSC registers sets vertical scale factor. VSC is the Vertical Scale Coefficient, n which determines the vertical scaling factor:

Vertical Scaling = (1 – VSC/256)

n is the reduction (caused by the scaling filter), modulo 256, in the number incoming vertical lines. With a range: $0 \leq n \leq 192$; $1 \leq VSC \leq 1/4$.

### 5.1.10 Flicker Filter

The FS400 flicker filter includes a variable vertical filter response in addition to a sharpness function.

The vertical averaging part of the filter is decomposed into an impulse function (no filtering) and a vertical flicker function. Adjusting the FLK Coefficient modifies the vertical filter from no filtering (FLK = 0) to a Three Line Average (FLK = 21), giving the user a wide choice in filtering options. A value of 16 provides the normal 3 line flicker filter response of ¼, ½, and ¼.

In addition to the variable vertical settings, the FS400 flicker filter has a sharpness function. This function is a two dimensional peaking function with accentuates the joint high vertical - high horizontal spatial frequencies.

The three line variable two dimensional flicker filter is formed by summing the unit impulse function with a vertical flicker function scaled by FLK (Flicker Coefficient) and the peaking function which is scaled by SHP (Sharpness Coefficient). The FLK and SHP variables have 5 bits of resolutions, with a usable a range from 0 to 21/32 for FLK, and 0 to 31/32 for SHP.

The perception of flicker varies with the angle of the flicker source to the horizontal. Experimentally it was determined that the perceived flicker threshold was between 20 and 35 degrees. Above 35 degrees few viewers can notice a line flicker and below 20 degrees most viewers can.

The vertical response of the flicker filter varies with the angle from horizontal and only modifies parts of the video input that contain flicker. The plot of the filter response is shown below at 0, 14, 27, and 45 degrees from horizontal.

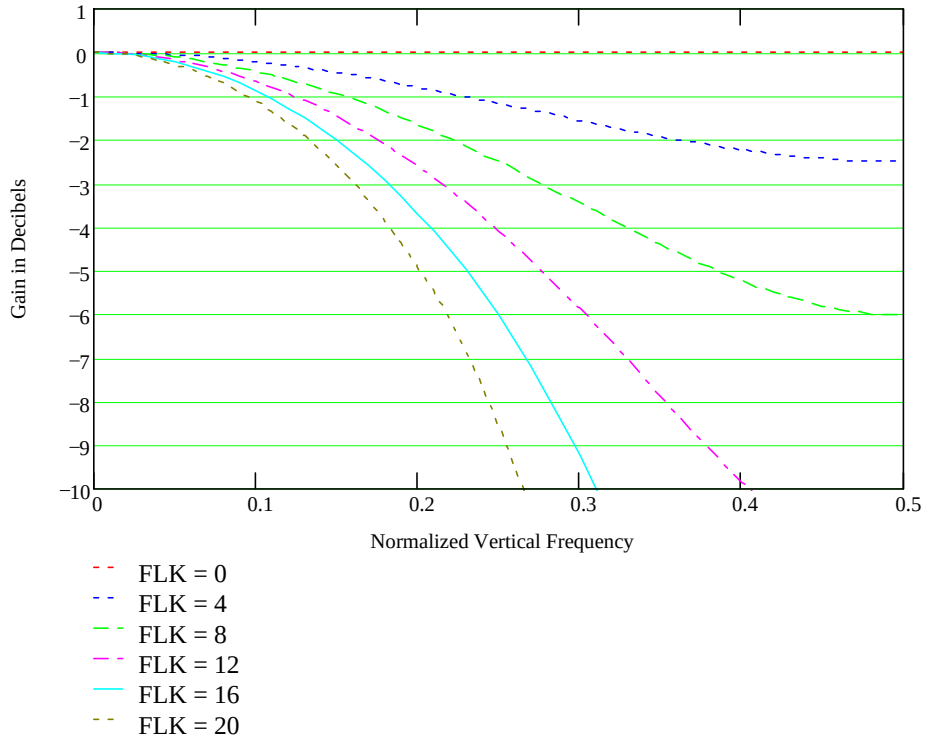



**Figure 8: Two Dimensional Flicker Filter Response (FLK=0,4,8,12,16,20; SHP=0)**

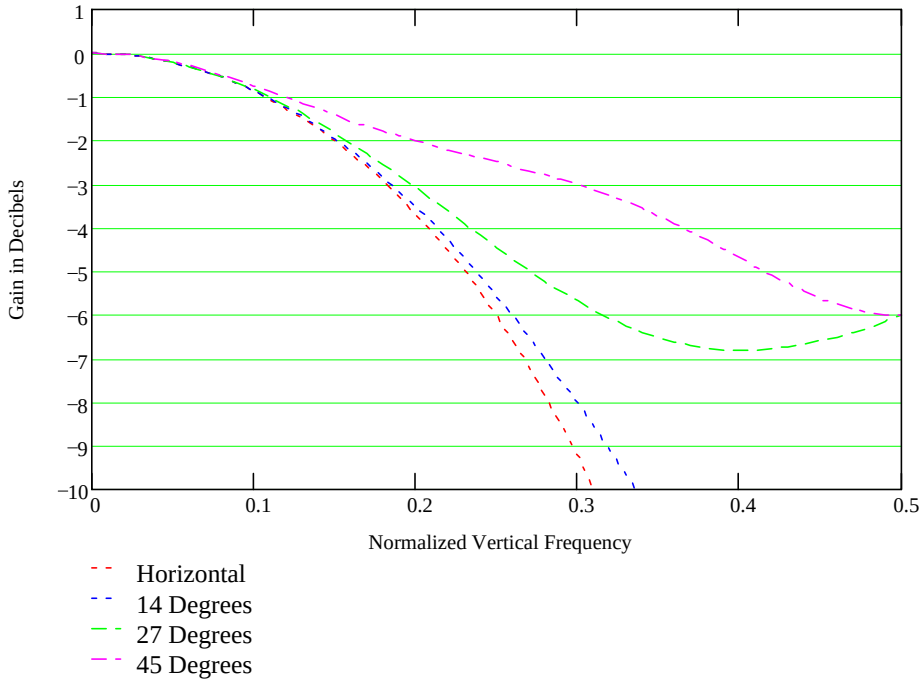



**Figure 9: FLK = 16, SHP = 8; Response at Horizontal, 14, 27, 45 Degrees**

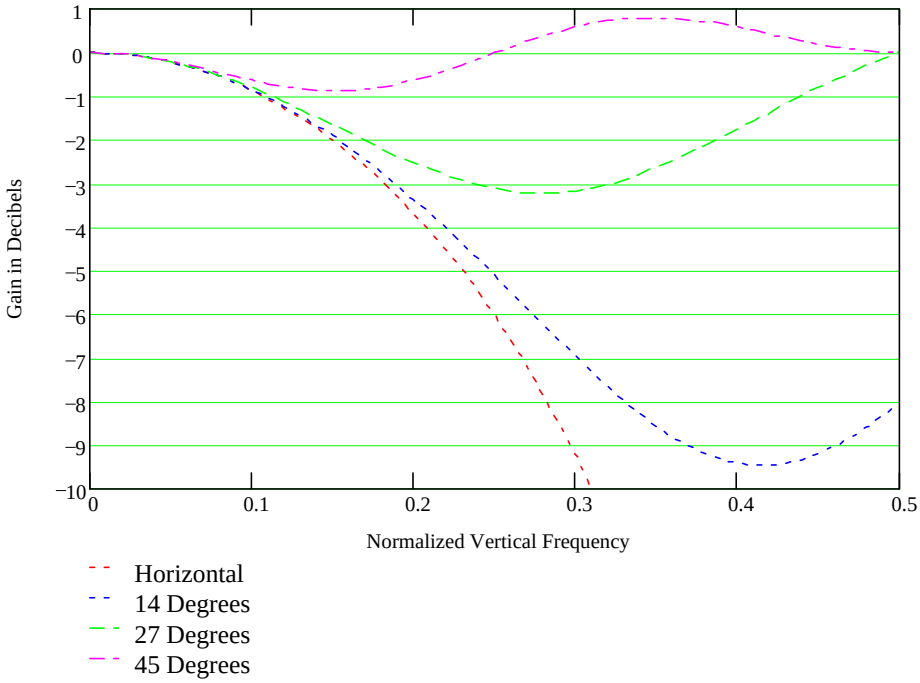



**Figure 10: FLK = 16, SHP = 16; Response at Horizontal, 14, 27, 45 Degrees**

## 5.2 Frame Store Controller

The Frame Store Controller (FSC) coordinates pixel data transfer between the FS400 and the Frame buffer. For normal operation, a 16 Mbit SDRAM is connected to the frame store controller port. Supported SDRAM parts include:

| | |
|---|---|
| NEC | μPD451616G5-A10 |
| NEC | μPD4516161AG5-A10B |
| Samsung | KM416S1120A-G/F10 |
| Samsung | KM416S1020CT-F10 |
| LG | GM72V161621CT-10 |
| MOSEL VITALIC | V-54C316162T-10 |
| Hyundai | HY57V161610BTC-10 |

The refresh of data in the SDRAM is accomplished through the continual update of the video output display. Therefore, any SDRAM selected must meet the frame rate maximum delay of the selected video. The worst case delay time is for PAL systems (25 frames/s) requiring a retention time of 40ms or greater.

Data from the Capture Engine is written into the frame store in parallel with the extraction of data for the encoder engine. Read and write buffers at the data port allow data to be transferred in bursts without interruption of the overall flow of pixels. Bandwidth of the FSC bus is 80 MHz or 100 MHz, sufficient to support maximum rate write cycles from the Capture Engine simultaneously with full rate Encoder Engine read cycles.

### 5.2.1 SDRAM Interface

The SDRAM Interface is designed to use a wide variety of 1Mx16 SDRAM parts. The critical parameters to use in selecting an SDRAM are listed below:

*<To be supplied later>*

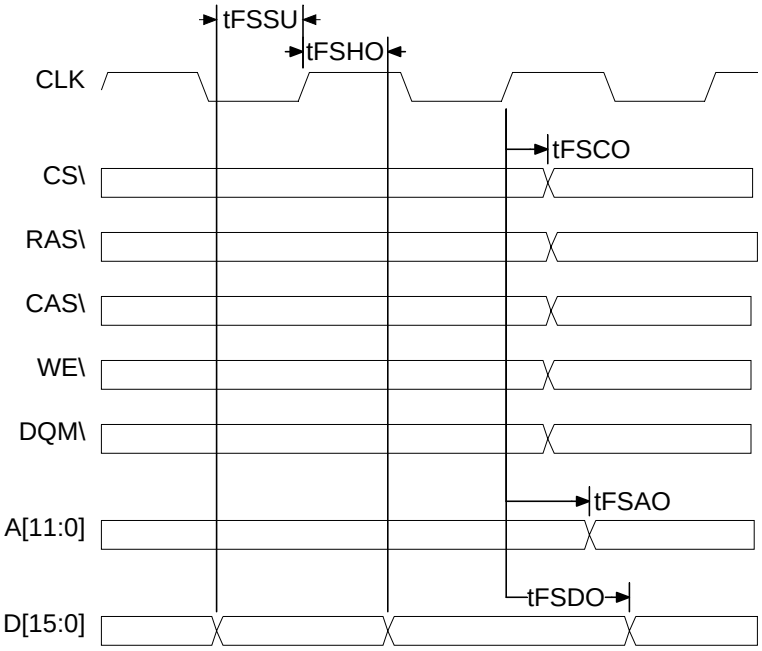



**Figure 11. Timing Parameter Definition, SDRAM Interface**

### 5.2.2 Phase Locked Loop

All SDRAM access is synchronized to the selected $4f_{SC}$ clock derived from the OSC1 clock input. The $4f_{SC}$ clock is divided by two, then depending upon the outgoing video standard selection, either NTSC or PAL, multiplied x9, x11, or x14 by the Frame Store Controller Phase Locked Loop to create the 80 or 100 MHz SDRAM clock.

### 5.2.3 Input Offset and Size Control

Besides synchronizing all SDRAM access activities, the frame store controller also coordinates several offset and size functions. These values are programmed into internal registers that control the setup of the Frame Store Controller.

Input Horizontal Offset, IHO is an eleven bit value that sets the horizontal displacement of the captured active video relative to the horizontal sync as show in *Figure 12*.

Input Vertical Offset, IVO is a twelve bit value that sets the vertical displacement of the captured active video relative to the vertical sync as show in *Figure 12*.

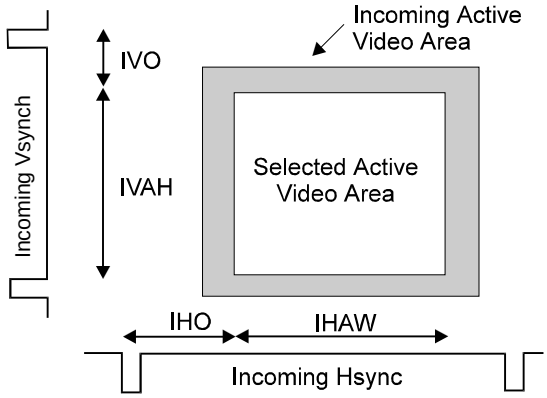



**Figure 12. Input Offset and Size Definitions**

Input Horizontal Active Width, IHAW is the 10-bit terminal count of the number of horizontal pixels to be inserted into frame store memory. (see *Figure 12*)

Input Vertical Active Height, IVAH is the 11-bit terminal count of the number of vertical pixels to be inserted into frame store memory, which is equal to 2*ILS. (see *Figure 12*)

## 5.2.4 Output Offset and Size Control

Synchronization and timing of outgoing video pixel data is predetermined by the selection of the video format. However, the location of the active video area must be selected by programming the offset registers. Leading edges of outgoing horizontal and vertical sync define the dimensions of the outgoing frame as shown in *Figure 13*. Offsets OVO and OHO define the position of the active video area within the outgoing frame.

Output Horizontal Offset, OHO is the 11-bit terminal count of the horizontal displacement of the outgoing active video in pixels relative to the leading edge of horizontal sync. (see *Figure 13*)

Output Vertical Offset, OVO is the 10-bit terminal count of the vertical displacement of the outgoing active video in pixels relative to the beginning of vertical sync equalization pulses. (see *Figure 13*)

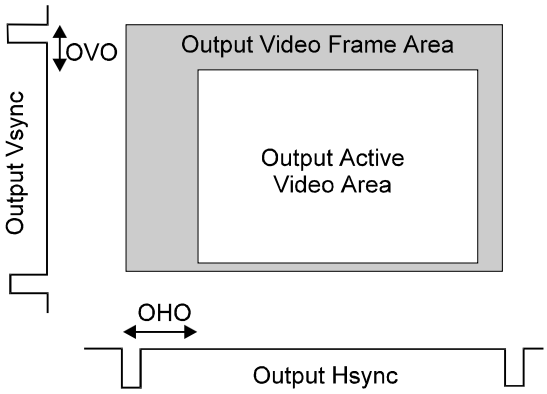



**Figure 13. Output Horizontal and Vertical Offset Definitions**

## 5.2.5 Freeze Frame

Setting the FREEZE bit of the Command Register can interrupt writing to the Frame Store. Capture will continue until the end of the current frame.

### 5.2.6 Zoom

When zoom is activated by setting the ZOOM bit to 1 in the Command Register, pixels are 2X replicated in the vertical and the horizontal directions. Pixels are replicated vertically by the Capture Engine that duplicates the loading of each line into the Frame Store. If, for example, only 120 separate lines are accepted, then 240 lines are written to each field contained the Frame Store Memory. When the zoomed frame segment is retrieved, the number of horizontal pixels is doubled by duplicating each pixel twice as the pixels are read from the Frame Store Memory. IHAW should be set to ½ the non-zoomed IHAW.

Offset may be applied to a zoomed image through the IHO and IVO registers as shown in *Figure 14*

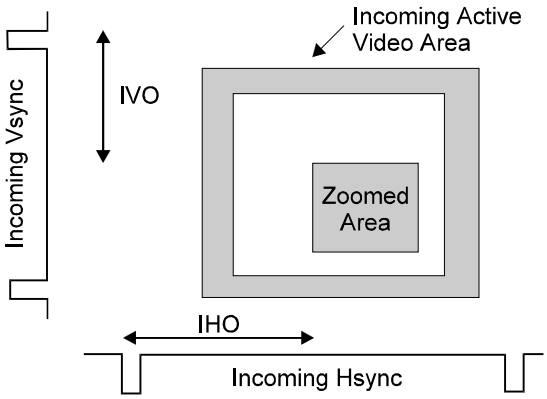



**Figure 14. Zoomed image showing offsets**

## 5.3 Encoder Engine

### 5.3.1 Timing and Control

Timing of the Encoder Engine is synchronized by the Encoder Timing and Control Block. Frame store clock, RAMCK_OUT is derived from the OSC1 clock by the SDRAM phase locked loop.

A clock signal must always be applied to the default clock input, OSC1. Depending upon the outgoing TV standard requirements, one of many possible clock combinations can be selected as depicted in the table below.

| Output Standard | OSC1 | OSC2 |
|---|---|---|
| NTSC and PAL | 14.31818 MHz | 17.734475 MHz |
| NTSC only | 14.31818 MHz | $V_{SSPF}$ |
| PAL only | Connect to OSC2 | 17.734475 MHz |
| Super RGB NTSC only | 14.31818 MHz | 20.140 MHz |
| Super RGB PAL only | 17.734475 MHz | 20.000 MHz |
| NTSC and VGA or VGA only | 14.31818 MHz | 25.175 MHz |
| NTSC and SVGA or SVGA only | 14.31818 MHz | 38.400 MHz |
| NTSC and Progressive NTSC | 14.31818 MHz | 28.63636 MHz |
| Progressive NTSC only | 14.31818 MHz | 28.63636 MHz |
| 100Hz PAL only | 17.734475 MHz | 35.46895 MHz |

**Table 4. Clock Connections**

If other combinations are desired (such as PAL and 100Hz), then OSC2 must be multiplexed with the different frequencies desired while OSC1 is connected to a 14.31818 MHz or 17.734475 MHz input. General principles apply as follows:

- NTSC, VGA, SVGA, and Progressive NTSC assume a 14.31818 MHz clock on OSC1
- PAL must be on OSC2
- 100Hz PAL interlaced assumes 17.734475 on OSC1 and 35.46895 on OSC2

- Multiplexing alternatives on OSC2 must be done with an external multiplexer (perhaps controlled by a GPO bit)

### 5.3.2 Horizontal Scaler

The Frame Store Controller passes pixels extracted from the external frame store memory to the Horizontal Scaler.

Programming the HSC register sets horizontal scale factor. HSC is the Horizontal Scaling Coefficient, m that determines the horizontal scaling factor:

$$HSF = (1 + m/128)$$

With a range: $0 \leq m \leq 384$; $1 \leq HSF \leq 1 + 384/128$

### 5.3.3 Digital Video Encoder

For Composite Video and Y/C outputs, pixels from the Horizontal Scaler are routed to the Video Encoder. NTSC (SMPTE 170M) and PAL (CCIR 624) standards are preprogrammed into the Video Encoder to preset horizontal and vertical timing, subcarrier frequency, and chrominance phase.

Setting the Command Register $OFMT_{1-0}$ bits to 00 selects Composite Video and Y/C outputs, setting it to 01 selects RGB and bypasses the encoder, and setting it to 10 or 11 selects component YUV directly.

Setting the Command Register $TV\_CLK_{1-0}$ selects the output clock mode per the table in the Command Register description (section 4.5.9).

Setting the Command Register PROG_INT bit switches the output mode from interlaced (0) to progressive (1).

Setting the Command Register PAL_NTSC bit switches the output mode from NTSC (0) to PAL (1).

Command Register bit, CBP inserts a 18% (0.64 MHz for NTSC; 0.8 MHz for PAL) $f_{SC}$ bandpass filter centered at $f_{SC}$ following the chroma modulator.

Command Register bit, LNTCH insert a luminance notch filter in the Y channel prior to the Composite Video summer and the Y output.

The Brightness Coefficient (BRT) provides an offset adjustment in the Luma Encoder. BRT is a signed 8-bit number.

The Contrast Coefficient (CON) provides a luma gain adjustment in the Luma Encoder. Contrast has an implicit denominator of 32.

The Color Saturation Coefficient (CSC) provides a gain adjustment in the Chroma Encoder. Saturation has an implicit denominator of 32.

### 5.3.4 YUV/RGB Matrix

For RGB outputs, pixels from the Horizontal Scaler are routed to the YUV/RGB matrix. Matrix coefficients are programmed by setting the $OFMT_{1-0}$ bits in the Command Register. For a YUV output, the $OFMT_{1-0}$ bits can be set to bypass the YUV/RGB matrix.

### 5.3.5 Digital-to-Analog Converters

Three 10-bit D/A converters accept data from either the video encoder or the YUV-to-RGB transcoder. Each output is a current source connected to the analog $V_{DDDA}$ supply. Current is injected into external resistor to develop the output voltage. Typically the DC load is 37.5Ω formed from two 75Ω resistors and a low pass filter. A 75Ω load may be selected to minimize power dissipation.

Peak output current of the D/A converters is established by $V_{REF}$ and an external resistor connected between $R_{REF}$ and ground. Peak current is approximately 11 times the current through the reference resistor.

An internal 1.276 volt reference is buffered from $V_{REF}$ by a resistor, to enable $V_{REF}$ to be overridden by an external voltage. Output current may be calibrated by resistor selection or by setting a potentiometer attached to $R_{REF}$.

For 1.3 volt video, with a 37.5Ω load, the correct value of $R_T$ is 389Ω. With a 75Ω load, the correct value of $R_T$ is 778 ohm. This calculation is shown in the following example:

$$R_T = 1024 / 97 * V_{REF} \Big/ \left( V_{DAC} \Big/ R_{LOAD} \right)$$

$$R_T = 1024 / 97 * 1.276V \Big/ \left( 1.3V \Big/ 37.5\Omega \right)$$

$$R_T = 389\Omega$$

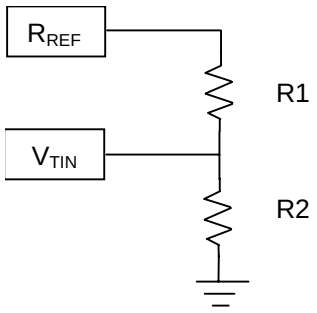



**Figure 15. $R_{REF}$ and $V_{TIN}$ Setup**

$V_{TIN}$ can be developed by splitting $R_T$ into 2 resistors, R1 and R2, wired so that $R_T$ = R1+R2.

To minimize DAC noise, a bypass capacitor must be connected from $C_{BYPR}$ to an adjacent $V_{DDDA}$ pin.

Disabling the power supplied to unused D/A converters may conserve power. Command Register Extended bit COMPOFF controls Composite Video D/A converter power. Command Register Extended bit YCOFF controls Y/C D/A converter.

### 5.3.6 On-Screen Display (FS403 only)

The FS403 has a simple eight wire interface for an On Screen Display (OSD) interface. A single input line is available for R, G, and B, giving the OSD programmer a choice of the eight 100% amplitude, 50% saturation Color Bar colors. Another input enables the overlay of the OSD into the video path, with a fifth line to half the intensity of the overlaid VGA video for better OSD contrast.

The chip also provides the Horizontal Sync, Vertical Sync, and 2Fsc clock for the microcontroller synchronization. The Zilog Z90211 and the Philips P83C055 families are directly supported.

Because the microcontroller operates at half the speed of the FS403 video encoder, a horizontal pixel from the microcontroller will be replicated to form two pixels in the FS403. Since the FS403 input hold times cannot be met by the OSD, the inputs are buffered using the 2Fsc clock."

## 5.4 Serial Control Port (R-Bus)

All FS400 register access is via a 2-wire serial control interface. Either 7 or 10-bit addressing may be used with two addresses available for each type of addressing scheme. (see Table 5. Serial Port Addresses).

Two signals comprise the bus: clock (SIOCLK) and bi-directional data (SIODATA). The FS400 acts as a slave for receiving and transmitting data over the serial interface. The maximum clock rate is 800 KHz.

Data received or transmitted on the SIODATA line must be stable for the duration of the positive-going SIOCLK pulse. Data on SIODATA may change only when SIOCLK = L. An SIODATA transition while SIOCLK = H is interpreted as a start or stop signal.

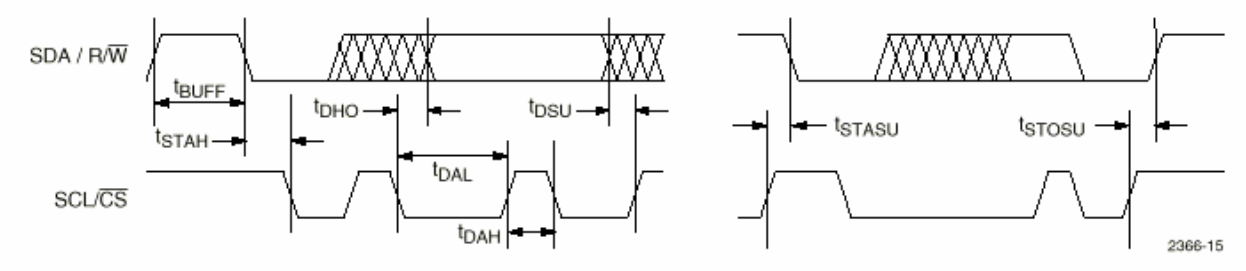



**Figure 16. Serial Port Read/Write Timing**

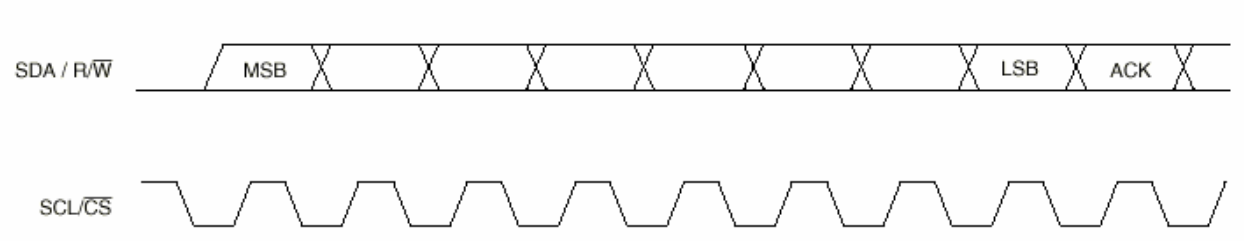



**Figure 17. Serial Interface – Typical Byte Transfer**

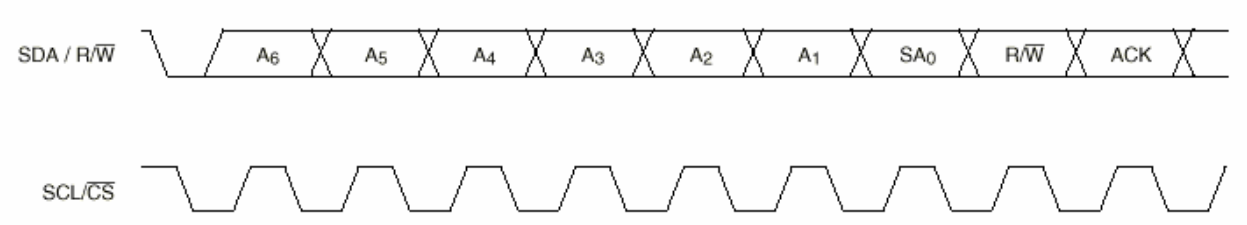



**Figure 18. 7-bit Slave Address with Read/Write\ Bit**

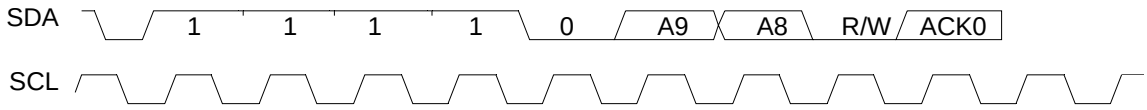



**Figure 19. 10-bit address transfer, upper two bits**

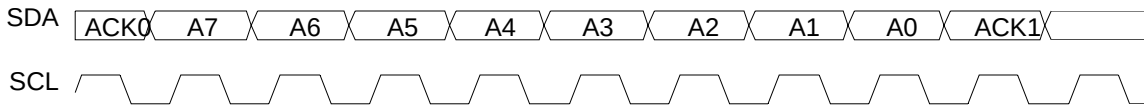



**Figure 20. 10-bit address transfer, lower eight bits**

There are five steps within a serial bus cycle:

1. Start signal
2. Slave address byte
3. Pointer register address byte
4. Data byte to read or write
5. Stop signal

When the serial interface is inactive (SIOCLK = H and SIODATA = H) communications are initiated by sending a start signal. The start signal (Figure 16, left waveform) is a HIGH-to-LOW transition on SIODATA while SIOCLK is HIGH. This signal alerts all slaved devices that a data transfer sequence is imminent.

For 7-bit addressing, the first eight bits of data transferred after a start signal comprise a seven bit slave address and a single R/W bit (Read = H, Write = L). As shown in *Figure 17*, the R/W bit indicates the direction of data transfer, read from or write to the slave device. If the transmitted slave address matches the address of the FS400 (set by the state of the $SIOA_0$ and $SIOA_{10/7}$ input pins in Table 5. Serial Port Addresses, the FS400 acknowledges by bringing SIODATA LOW on the 9th SIOCLK pulse (see Figure 18). If the addresses do not match, the FS400 does not acknowledge.

With 10-bit addressing (see Figure 19 and Figure 20), data is still transferred in 8-bit chunks. The upper two bits of the ten bit address are transferred as the lower two bits of the first byte along with the reserved sequence 11110 in the upper five bits and the R/W bit. The lower eight bits are transferred in the second byte without a R/W bit. Subsequent data reads or writes follow the 7-bit transfer sequences.

For each byte of data read or written, the MSB is the first bit of the sequence.

| $SIOA_{10/7}$ | $SIOA_0$ | Address |
|---|---|---|
| 0 | 0 | 4A |
| 0 | 1 | 6A |
| 1 | 0 | 224 |
| 1 | 1 | 276 |

**Table 5. Serial Port Addresses**

### 5.4.1 Data Transfer via Serial Interface

If a slave device, such as the FS400 does not acknowledge the master device during a write sequence, SIODATA remains HIGH so the master can generate a stop signal. If the master device does not acknowledge the FS400 during a read sequence, the FS400 interprets this as “end of data”. SIODATA remains HIGH so the master can generate a stop signal.

To write data to a specific FS400 control register, the 8-bit pointer must be loaded with the address of the target control register after the slave address has been established. Value of the pointer is the base address for subsequent write operations. The pointer address auto-increments by after each control register data transfer. If more bytes are transferred than there are available addresses, the address will not increment and remain at its maximum value of 6 and send an acknowledge signal, ACK.

Data is read from the control registers of the FS400 in a similar manner, except that two data transfer operations are required:

1. Write the slave address byte with bit R/W = L.
2. Write the pointer byte.
3. Write the slave address byte with bit R/W = H
4. Read the control register indexed by the pointer.

Preceding each slave write, there must be a start cycle. Following the pointer byte there should be a stop cycle. Sequential registers may be accessed by repeated read cycles since pointer auto-increments after each byte transfer. After the last read, there must be a stop cycle comprising a LOW-to-HIGH transition of SIODATA while SIOCLK is HIGH. (see *Figure 16*, right waveform)

A repeated start signal occurs when the master device driving the serial interface generates a start signal without first generating a stop signal to terminate the current communication. This is used to change the mode of communication (read, write) between the slave and master without releasing the serial interface lines.

### 5.4.2 Serial Interface Read/Write Examples

Examples below show how serial bus cycles can be linked together for single and multiple register read and write access cycles. For sequential register accesses, each ACK handshake initiates further SIOCLK clock cycles from the master to transfer the next data byte.

#### 5.4.2.1 Write to one control register

- Start signal
- Slave Address byte (R/W bit = LOW)
- Pointer Address byte
- Data byte to register
- Stop signal

#### 5.4.2.2 Write to two consecutive control registers

- Start signal
- Slave Address byte (R/W bit = LOW)
- Pointer Address byte
- Data byte to register
- Data byte to register (pointer address + 1)
- Stop signal

#### 5.4.2.3 Read from one control register

- Start signal
- Slave Address byte (R/W bit = LOW)
- Pointer Address byte
- Start signal
- Slave Address byte (R/W bit = HIGH)
- Data byte from register
- Stop signal

#### 5.4.2.4 Read from two consecutive data registers

- Start signal
- Slave Address byte (R/W bit = LOW)
- Data byte to register
- Start signal
- Slave Address byte (R/W bit = HIGH)
- Data byte from pointer address
- Data byte from (pointer address + 1)
- Stop signal

## 5.5 Embedded Microprocessor

The configuration of the FS400 can be set by either an external source driving the SIO interface or internally using the built-in micro-controller as both share the serial bus. If the external pin INTCPUEN is active, the internal microprocessor has control.

# 6. Specifications

## 6.1 Absolute Maximum Ratings

**(beyond which the device may be damaged)[1]**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| **Power Supply Voltages** | | | | |
| $V_{DD}$ (Measured to $V_{SS}$) | -0.5 | 3.3 | 4.6 | V |
| $V_{DDAD}$ (Measured to $V_{SSAD}$) | -0.5 | 3.3 | 4.6 | V |
| $V_{DDPA}$ and $V_{DDPF}$ (Measured to $V_{SSPA}$ and $V_{SSPF}$) | -0.5 | 3.3 | 4.6 | V |
| $V_{DDDA}$ (Measured to $V_{SSDA}$) | -0.5 | 3.3 | 4.6 | V |
| $V_{SSAD}$, $V_{SSPA}$, $V_{SS}$, $V_{SSPA}$, $V_{SSDA}$ (delta) | -0.5 | | 0.5 | V |
| **Digital Inputs** | | | | |
| 3.3 V logic applied voltage (Measured to $V_{SS}$)[2] | -0.5 | | $V_{DD}$ + 0.5 | V |
| Forced current [3, 4] | -10.0 | | 10.0 | mA |
| **Analog Inputs** | | | | |
| Applied Voltage (Measured to $V_{SSAD}$)[2] | -0.5 | | $V_{DDDA}$ + 0.5 | V |
| Forced current [3, 4] | -10.0 | | 10.0 | mA |
| **Digital Outputs** | | | | |
| 3.3 V logic applied voltage (Measured to $V_{SS}$)[2] | -0.5 | | $V_{DD}$ + 0.5 | V |
| Forced current [3, 4] | -6.0 | | 6.0 | mA |
| Short circuit duration (single output in HIGH state to ground) | | | 1 | second |
| **Temperature** | | | | |
| Operating, Ambient | -20 | | 110 | °C |
| Junction | | | 150 | °C |
| Lead Soldering (10 seconds) | | | 300 | °C |
| Vapor Phase Soldering (1 minute)[1] | | | 220 | °C |
| Storage[1] | -65 | | 150 | °C |
| Electrostatic Discharge[5] | | | ±150 | V |

**Notes:**

1. Functional operation under any of these conditions is NOT implied. Performance and reliability are guaranteed only if Operating Conditions are not exceeded.
2. Applied voltage must be current limited to specified range.
3. Forcing voltage must be limited to specified range.
4. Current is specified as conventional current flowing into the device.
5. EIAJ test method.

## 6.2 Operating Conditions

| Parameter | | Min | Nom | Max | Units |
|---|---|---|---|---|---|
| $V_{DD}$ | Digital Power Supply Voltage | 3.15 | 3.3 | 3.45 | V |
| $V_{DDAD}$, $V_{DDDA}$ | A/D and D/A Supply Voltage | 3.15 | 3.3 | 3.45 | V |
| $V_{DDPA}$, $V_{DDPF}$ | PLLs Supply Voltage | 3.15 | 3.3 | 3.45 | V |
| $A_{GND}$ | Analog Ground (Measured to $D_{GND}$) | -0.1 | 0 | 0.1 | V |
| $V_{RT}$ | Reference Voltage, Top | 0.5 | 0.75 | 2.0 | V |
| $V_{IN}$ | Analog Input Range | 0 | | VRT | V |
| $V_{REF}$ | External Reference Voltage | | 1.276 | | V |
| $I_{REF}$ | D/A Converter Reference Current ($I_{REF}$ = $V_{REF}$/$R_{REF}$, flowing out of the $R_{REF}$ pin) | | 3.15 | | mA |
| $R_{REF}$ | Reference Resistor, $V_{REF}$ = Nom | | 392 | | Ω |
| $R_L$ | DAC Total Output Load Resistance | | 37.5 | | Ω |
| $T_A$ | Ambient Temperature, Still Air | 0 | | 70 | °C |

## 6.3 Electrical Characteristics

| Parameter | | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| **Power Supply Currents** | | | | | | |
| $I_{DD}$ | 3.3 volt current | FSCK–IN = 80MHz | | 159 | | mA |
| $I_{DDAD}$ | 3.3 volt Analog current | ADXCK=40MHz | | 66 | | mA |
| $I_{DDDA}$ | 3.3 volt Analog current | ADXCK=40MHz | | 90 | | mA |
| $I_{DDPA}$ | 3.3 volt Analog current | ADXCK=40MHz | | 4 | | mA |
| $I_{DDPF}$ | 3.3 volt Analog current | ADXCK=40MHz | | 16 | | mA |
| $I_{DDT}$ | 3.3 volt Total Current | CKNTSC=20MHz | | 350 | | mA |
| **Digital Inputs and Outputs** | | | | | | |
| $C_I$ | Input Capacitance | | | 5 | 10 | pF |
| $C_O$ | Output Capacitance | | | 10 | | pF |
| $I_{IH}$ | Input Current, HIGH | $V_{DD}$ = max.,<br>$V_{IN}$ = max. | | | ±10 | µA |
| $I_{IL}$ | Input Current, LOW | $V_{DD}$ = max.,<br>$V_{IN}$ = 0 V | | | ±200 | µA |
| $I_{ILP}$ | Input Current, LOW with pull-up | $V_{DD}$ = max.,<br>$V_{IN}$ = 0 V | -100 | | | µA |
| $V_{IHTTL}$ | Input Voltage, Logic HIGH (TTL) | | 2.0 | | | V |
| $V_{ILTTL}$ | Input Voltage, Logic LOW (TTL) | | | | 0.8 | V |
| $I_{OH}$ | Output Current, Logic HIGH | | | | -2.0 | mA |
| $I_{OL}$ | Output Current, Logic LOW | | | | 2.0 | mA |
| $V_{OH}$ | Output Voltage, HIGH | $I_{OH}$ = -2mA | 2.4 | | | V |
| $V_{OL}$ | Output Voltage, LOW | $I_{OL}$ = 2mA | | | 0.4 | V |
| **Analog Inputs** | | | | | | |
| $C_{AI}$ | A/D Input Capacitance | ADCLK = LOW<br>ADCLK = HIGH | | 4<br>12 | | pF<br>pF |
| $R_{IN}$ | A/D Input Resistance | | 500 | 1000 | | kΩ |
| $I_{CB}$ | A/D Input Current | | | | ±15 | µA |
| $V_{RO}$ | Voltage Reference Output | Internal Reference | 1.15 | 1.276 | 1.40 | V |
| $I_{RO}$ | VREF Output Current | External $V_{REF}$ | -150 | | +150 | µA |
| **Analog Outputs** | | | | | | |
| $V_{OC}$ | Video Output Compliance | | -0.4 | | 2 | V |
| $R_{OUT}$ | Video Output Resistance | | | 15 | | kΩ |
| $C_{OUT}$ | Video Output Capacitance | $C_{OUT}$ = 0 mA,<br>Freq. = 1 MHz | | 15 | | pF |
| $I_{OS}$ | Short-Circuit Current | | -20 | | -80 | mA |

## 6.4 Switching Characteristics

| Parameter | | Conditions | Min | Typ[1] | Max | Unit |
|---|---|---|---|---|---|---|
| **Clocks** | | | | | | |
| $f_{CKIN\_N}$ | NTSC Reference Clock Frequency | | | 14.31818 | | MHz |
| $f_{CKIN\_P}$ | PAL Reference Clock Frequency | | | 17.734475 | | MHz |
| $f_{XTOL}$ | Reference Clock Frequency Tolerance | | | 50 | 50[2] | ppm |
| $t_{PWH}$ | Reference Clock Pulse Width, HIGH | | | 40 | | ns |
| $t_{PWL}$ | Reference Clock Pulse Width, LOW | | | 40 | | ns |
| Reset | Assert time on RESET to reset the part | | 8 | | | Clocks |
| **Incoming Syncs** | | | | | | |
| $f_H$ | HS_IN frequency | | 24 | | 100 | KHz |
| $f_V$ | VS_IN frequency | | 50 | | 130 | Hz |
| $N_H$ | Number of lines per frame | | 100 | | 4095 | |
| $t_{PWHS}$ | HS_IN Pulsewidth | | 1 | | | µs |
| $t_{VS-HS}$ | VS_IN to HS_IN Delay | | 0 | | | ns |
| $t_{DS}$ | Sync Delay (VGA Sync to Sync Out) | | | 100 | | ns |
| **Video Output** | | | | | | |
| $t_{DOV}$ | Analog Output Delay ($4f_{SC}$ clock to Video Out) | | | | 30 | ns |
| $t_R$ | D/A Output Current Risetime (10% to 90%) | | | | 10 | ns |
| $t_F$ | D/A Output Current Falltime (90% to 10%) | | | | 10 | ns |
| SKEW | D/A to D/A Skew | | -3 | 0 | 3 | ns |
| **SDRAM Frame Buffer Interface** | | | | | | |
| $t_{FCKL}$ | FS_CK pulse width, LOW | | 5.0 | | | ns |
| $t_{FCKH}$ | FS_CK pulse width, HIGH | | 5.0 | | | ns |
| $t_{FSCO}$ | FS_CK to RAS\, CAS\, WE\ DQM\ out delay | | 2.0 | | | ns |
| $t_{FDSU}$ | FS_CK to data setup time | | 3.0 | | | ns |
| $t_{FDHO}$ | FS_CK to data hold time | | 3.0 | | | ns |
| $t_{FDO}$ | FS_CK to data out delay | | 2.0 | | | ns |
| $t_{FAO}$ | FS_CK to address out delay | | 2.0 | | | ns |
| **Serial Microprocessor Interface** | | | | | | |
| $t_{DAL}$ | SIOCLK Pulse Width, LOW | | | 1.3 | | µs |
| $t_{DAH}$ | SIOCLK Pulse Width, HIGH | | | 0.6 | | µs |
| $t_{STAH}$ | SIODATA Start Hold Time | | | 0.6 | | µs |
| $t_{STASU}$ | SIOCLK to SIODATA Setup Time (Stop) | | | 0.6 | | µs |
| $t_{STOSU}$ | SIOCLK to SIODATA Setup Time (Start) | | | 0.6 | | µs |
| $t_{BUFF}$ | SIODATA Stop Hold Time Setup | | | 1.3 | | µs |
| $t_{DSU}$ | SIODATA to SIOCLK Data Setup Time | | | 300 | | ns |
| $t_{DHO}$ | SIODATA to SIOCLK Data Hold Time | | | 300 | | ns |

**Notes:**
1. Values shown in Typ column are typical for $V_{DD} = V_{DDA} = +3.3V$ and $T_A = 25°C$
2. TV subcarrier acceptance band is ± 300 Hz.

## 6.5 System Performance Characteristics

| Parameter | | Conditions | Min | Typ[1] | Max | Unit |
|---|---|---|---|---|---|---|
| **A/D Converter Input** | | | | | | |
| $E_{LI}$ | A/D Integral Linearity Error, Independent | $V_{RT}$ = 0.7V | | ±1 | | LSB |
| $E_{LD}$ | A/D Differential Linearity Error | $V_{RT}$ = 0.7V | | ±1 | | LSB |
| $E_{OT}$ | Offset Voltage, Top | $V_{RT} – V_{IN}$ for most positive code transition | | 150 | | mV |
| $E_{OB}$ | Offset Voltage, Bottom | $V_{IN}$ for most negative code transition | | 150 | | mV |
| **D/A Converter Output** | | | | | | |
| RES | D/A Converter Resolution | | 10 | 10 | 10 | Bits |

**Notes:**

1. Values shown in Typ column are typical for $V_{DD} = V_{DDA}$ = +3.3V and $T_A$ = 25°C

# 7. Application Notes

## 7.1 Circuit Example - PC

RGB video signals and the vertical and horizontal sync signals are intercepted by tapping connections to the VGA connector. Typically, control of the FS400 will be through the serial interface. S-video and Composite Video outputs are fed to connector that should be located at the board edge. Power is derived from +3.3 volt analog and digital supplies. It is recommended that the analog supply be clean of noise.

Input impedance of the A/D converters exceeds 500K and will not load VGA RGB lines. Shunt 100pF capacitors and 110Ω series resistors form a low pass filter with 15 MHz cutoff. Filtering the incoming video has two functions:

1. equalizing the intensities of vertical lines
2. eliminating sampling noise.

Outgoing video is filtered by low pass Bessel filters with 10 MHz cutoff. An optional low pass filter has two functions:

1. removal of steps from the Composite Video output
2. limiting the bandwidth of outgoing video.

For higher definition of horizontal frequencies, a sinx/x correction filter should be incorporated to compensate for 1 dB sampling loss of the D/A converters. Schottky diode clamps protect the FS400 from high voltage transients.

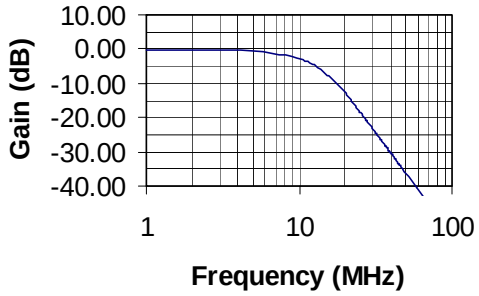



**Figure 21. Video Filter Response**

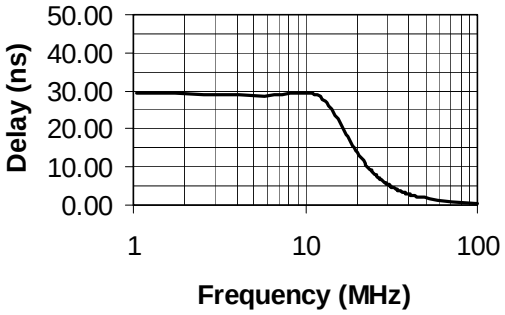



**Figure 22. Video Filter Delay**

$V_{TIN}$ is derived from the bias voltage across $R_{REF}$ that is established by the internal reference voltage available at $V_{REF}$. $R_{REF}$ is split into two resistors with series resistance to set the $R_{REF}$ current and ratio to establish the $V_{TIN}$ voltage.

## 7.2 FS400 Design and Layout Considerations

Careful circuit design and layout are key factors that insure a successful implementation of the FS400 in a product. The following guidelines will help insure that your design yields the best possible results.

### 7.2.1 Video Input to A-D Converters

- Consider using a higher capacitor value and a lower resistor value in the input filter. For example, use 100Ω and 100pF instead of 220Ω and 47pF. This will reduce noise at the ADC input.
- Place the input lowpass filter shunt capacitor as close as possible to the input pin on the FS400. This capacitor acts as a reservoir of charge for the ADC's input sampling circuitry, and limits the voltage kickback from the sampling process.
- Make the video input traces to the A-D converters as short as possible. Do not route other signal traces (especially clocks!) parallel to these traces, or close to the ADC inputs of the FS400. External interference coupled into the video can easily be discerned by the eye at very low levels.
- Consider adding diodes to $V_{DDAD}$ and $V_{SSAD}$ on these lines to reduce the risk of ESD damage to the FS400.

### 7.2.2 Input ADC Phase Lock Loop

- The analog supply for the ADC PLL should always be clean and noise free to insure minimum jitter in the PLL. This applies regardless of the PLL used, internal or external. Do not power other circuitry from the PLL supply.
- When using the internal PLL, the supply line $V_{DDPA}$ should be decoupled with a series resistor of 150Ω and a 4.7μF tantalum capacitor. If 50/60Hz ripple is an issue, consider using 47 or 100μF. Always have a 1000pF to 0.1μF capacitor to remove high frequency noise.
- Do not run other traces (especially clocks) near an external PLL.
- Use a solid ground plane under the FS400.

### 7.2.3 Memory Clock Phase Lock Loop

- When using an external PLL for the memory clock, series terminate the clock line, and keep the traces as short as possible to reduce EMI.
- Avoid using an even multiple of the $4f_{SC}$ clock as your memory clock. This increases EMI since multiples of the clocks that overlap are additive.

### 7.2.4 External SDRAM Interface

- Keep the traces from the FS400 to the external SDRAM as short as possible. Series terminate the RAS\, CAS\, and FSCK_OUT as close as possible to the FS400. Consider series terminating the address lines as well for further EMI reduction.
- Keep other traces, especially analog traces, away from the traces associated with the SDRAM. In addition to 0.1μF bypass capacitors, consider adding 100 to 1000pF capacitors to reduce higher frequency noise on the power supply.

### 7.2.5 HSYNC and VSYNC

- The HSYNC and VSYNC inputs to the FS400 should be low pass filtered. 150Ω and 100pF work well. Consider termination on these lines if the input cable length gives rise to significant reflections, undershoot and overshoot.
- Consider adding diodes to $V_{DD}$ and $V_{SS}$ on these lines to reduce the risk of ESD damage to the FS400.

### 7.2.6 Video Output Filters

- To reduce step noise on the D/A converter outputs, and to lower EMI, consider placing the 75Ω termination resistors and the first capacitor of the output filter close to the video output pins of the FS400.

### 7.2.7 Analog Power Supply Bypassing, Filtering, and Isolation

- When possible, it is recommended that the analog supply voltages be fed from a linear voltage regulator. Switching power supply noise, and noise from the digital plane can induce visible artifacts into the displayed video. Always provide sufficient filtering and high frequency bypassing to insure that power supply noise is minimized for visual as well as EMI reasons.
- Noise from the digital plane must be kept isolated from the analog plane. Consider using a 1Ω to 2Ω resistor with a 100μF capacitor as a filter network. The voltage drop to the analog plane is minimal, and this effectively minimizes clock noise from the digital plane.
- It is recommended that each power supply section be isolated with a ferrite bead and a 4.7μF capacitor. Where the power pins are so close together that the 0.1μF bypass capacitors are adjacent, consider changing one of the adjacent capacitors to 100 to 1000pF to reduce higher frequency noise on the power supply.

### 7.2.8 Power and Ground

Within the FS400, separate power is routed to functional sections: A/D converters, phase locked loops, D/A converters, digital processors and digital drivers. To minimize power consumption, a +3.3 volt supply is used for the external SDRAM interface and the internal line buffers. All ground pins should be connected to a common ground plane. Power pins should be segregated into analog and digital sections.

Clean analog power should be applied to the $V_{DDAD}$, $V_{DDPA}$, $V_{DDPF}$, and $V_{DDDA}$ pins. A 0.1 μF capacitor should be placed adjacent to each group of pins. The capacitor connected to $C_{BYPR}$ is critical, and it must be connected to $V_{DDDA}$ to minimize noise at the D/A converter outputs. Chip capacitors are recommended.

Digital power may be derived from system digital +3.3 volts. If necessary insert a ferrite bead in series with the supply trace. A 47 μF capacitor should be placed across the common +3.3 VDC for $V_{DD}$, $V_{DDAD}$ and $V_{DDDA}$ to act as a reservoir for heavy currents drawn by D/A converters and memory. At least one 0.1 μF capacitor should be located adjacent to $V_{DD}$ pins along each side of the FS400 to supply transient currents.

## 7.3 Interfacing to the FS400 in a Mixed Voltage Environment

As many devices designed today, the FS400 is powered by +3.3 Volts. However, 5 Volt devices are still very common today and will continue to be used for some time in the future. To meet this interface requirement the FS400 has 5 Volt tolerant inputs. Unfortunately, the FS400 does not have 5 Volt tolerant outputs. When a bi-directional I/O pin is used on the FS400, care must be taken when driving the device.

### 7.3.1 5 to 3.3 Volt Translation

A series resistor is the simplest way to safely drive an I/O pin of the FS400 . This input resistor limits the input current to under the 10mA maximum limit when the input is raised towards the 5 Volt supply.

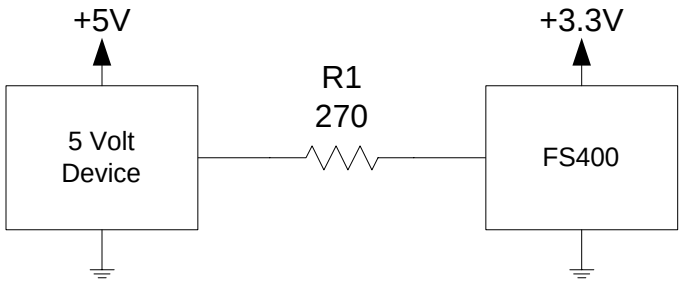



**Figure 23. 5 to 3.3 Volt Translation using a Resistor**

In applications where a series resistor limits the speed or the drive capability of the I/O, a pass gate can be used instead of a resistor. The pass gate (Q1) provides a low impedance path between the devices when the switching voltage is low and a high impedance path as the switching voltage approaches the supply rail.

In the circuit below, D1 biases the gate of the MOSFET (Q1) down to 4.3 Volts so Q1 will turn off when its input exceeds 3.3 Volts. This limits the high state input current, into the FS400, to under the 10mA maximum value.

Note that the MOSFET source is connected to the lower supply device and the MOSFET drain is connected to the higher supply device. This configuration reverse biases the MOSFET drain to bulk diode in the high switching state.

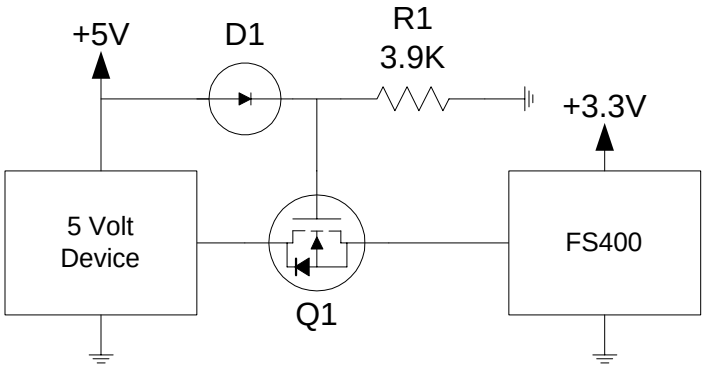



**Figure 24. 5 to 3.3 Volt Translation using a MOSFET**
**Q1 = BSS138, D1 = 1N4148**

### 7.3.2 SIO Bus Interfacing

The SIO bus was developed previous to 3.3V logic processes. The SIO bus input voltage specification is 1.5 Volts for $V_{IL}$ and 3.0 Volts for $V_{IH}$. The FS400 is built on a 3.3 Volt process and has 5 Volt tolerant inputs with a $V_{IL}$ of 0.8 Volts and a $V_{IH}$ of 2.0 Volts.

For most applications this voltage difference is not an issue as the output drive low specification ($V_{OL}$) of the SIO bus and the FS400 are both 0.4 Volts. However, in heavily loaded SIO busses the output $V_{OL}$ is not always preserved.

An easy way to regain the 0.7 Volt difference in the $V_{IL}$ specification of the FS400 and the SIO bus is to bias the FS400's input negative by a diode drop (D1). The diode can be biased by a long-tail resistor pair or a current source pair. Shown below is the long-tail pair:

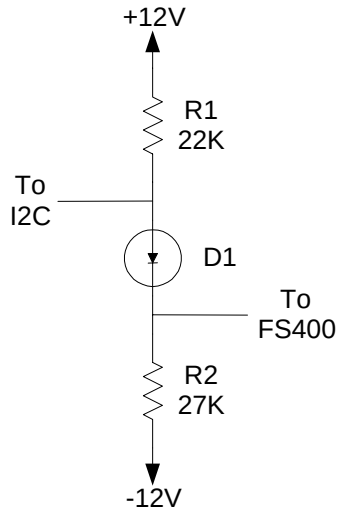



**Figure 25. SIO Translation Using Long-tail Resistors**
**D1 = 1N4148**

The long-tail pair is a simple circuit but has the disadvantage of requiring higher voltage power supplies. Also, these supplies may have to power up in a specific sequence so the surround circuits are not over-voltage.

The translation circuit below requires only one 5 Volt power supply and has no special sequence requirements. In addition, the circuit offers a high impedance load (Q1 become reverse biased) to the SIO bus when its power supply is removed. Unfortunately, it requires more parts. In applications where transistors are more readily available, R2 and R3 can be replaced with diode connected transistors.

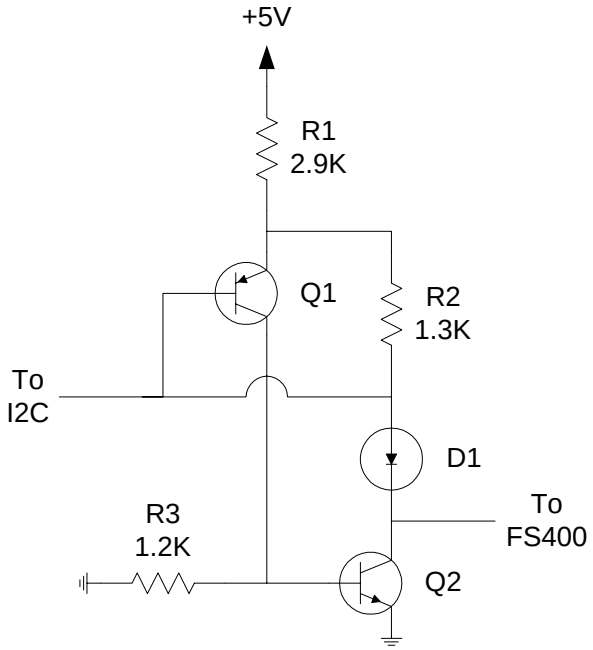



**Figure 26. SIO Translation Using Current Mirrors**
**D1 = 1N4148, Q1 = 2N3906, Q2 = 2N3904**

For applications with more than one supply, combinations of the above two circuits can be used. However, the simplest approach to this problem is to limit the loading on the SIO bus when possible. When this is not possible, some of the SIO passive loads can be replaced with active ones. This will increase the SIO access speed without increasing the SIO output low drive current.

In lightly loaded SIO busses (where $V_{OL}$ is easily reached), a single MOSFET can be used for translation on the SIO SDA line (Since the FS400's input for SCL is 5 Volt tolerant, no additional parts are required on that input). Pull-ups on both supply sides assure that the output on each device can reach its corresponding upper limit. The pass gate (Q1) provides a low impedance connection during the low switching voltage.

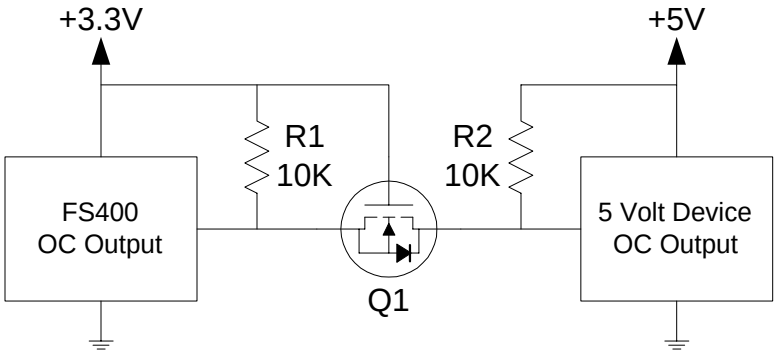



**Figure 27. SIO (Open Collector) Translation using a MOSFET**
**Q1 = BSS138**

Q1 also provides isolation when the 3.3 Volt power to the FS400 is removed. The pass gate (Q1) gate to source voltage is negative and the bulk diode is off. This effectively removes the FS400's Serial Data line from the SIO bus system and allows the higher voltage SIO section to operate normally.

# 8. Mechanical Dimensions

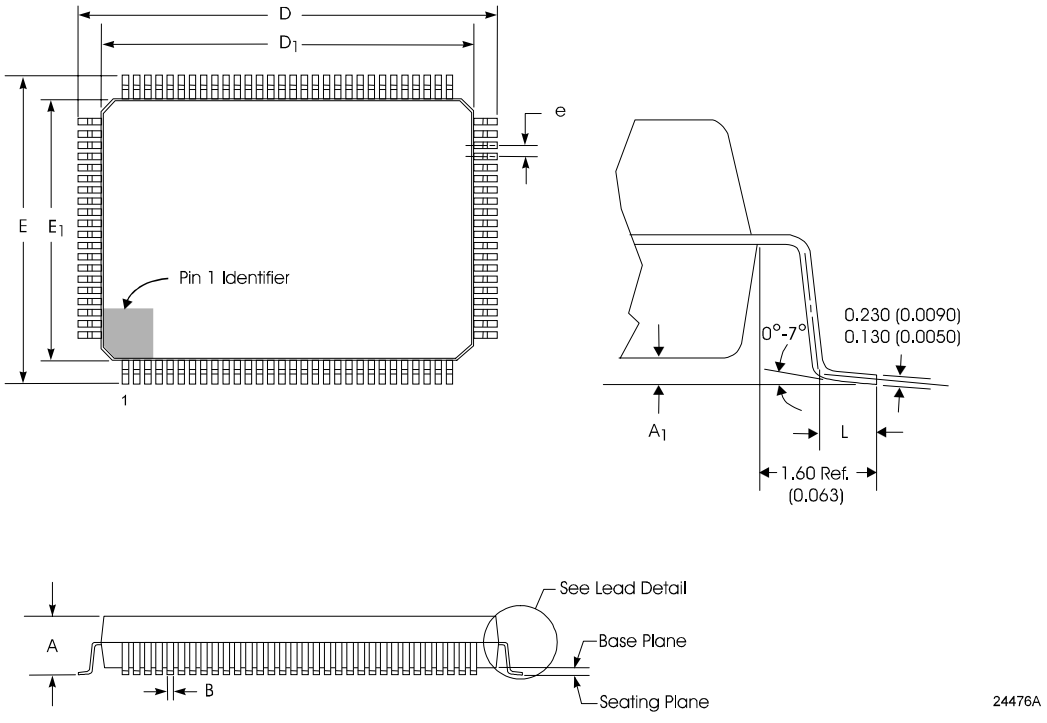
## 8.1 100-Lead PQFP (KH) Package - FS401LF

Package is RoHS Compliant.

| Symbol | Millimeters | | Notes |
|---|---|---|---|
| | Min. | Max. | |
| A | | 3.00 | |
| A1 | 0.05 | - | |
| A2 | 2.55 | 2.75 | |
| B | 0.25 | 0.40 | 3,5 |
| C | 0.10 | 0.25 | 5 |
| D | 23.60 | 24.20 | |
| $D_1$ | 19.80 | 20.20 | |
| E | 17.30 | 18.20 | |
| E1 | 13.80 | 14.20 | |
| e | 0.65 BSC | | |
| L | 0.60 | 1.00 | 4 |
| N | 100 | | |
| ND | 30 | | |
| NE | 20 | | |
| α | 0 | 8° | |
| ccc | - | | |

**Notes:**

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Controlling Dimension is millimeters
3. Dimension “B” does not include dambar protrusion. Allowable dambar protrusion shall be .08mm (.003in.) maximum in excess of the “B” dimension. Dambar cannot be located on the lower radius or the foot.
4. “L” is the length of terminal for soldering to a substrate.
5. “b” & “C” include lead finish thickness.



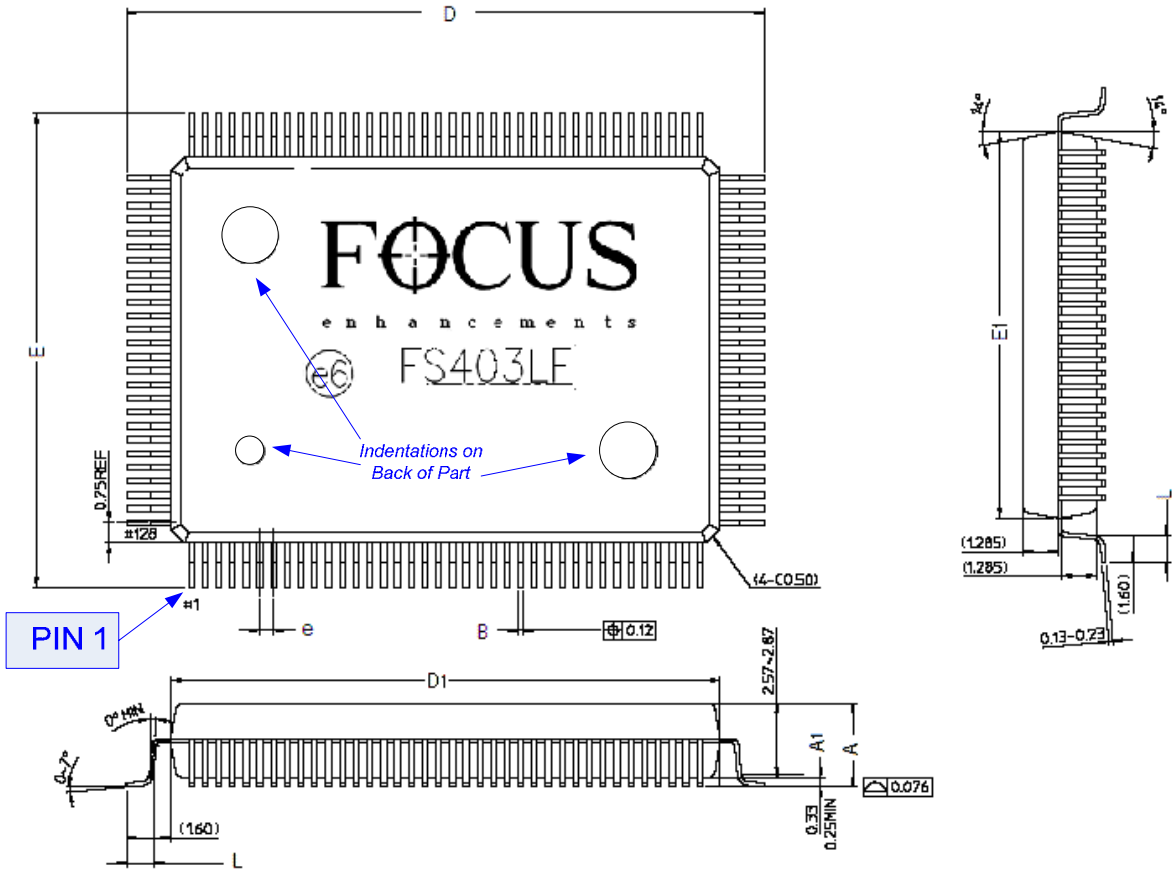
## 8.2 128-Lead PQFP Package, FS403 LF

Package is RoHS Compliant.

| Symbol | FS403LF | |
|---|---|---|
| | Min. | Max. |
| A | - | 3.40 |
| A1 | 0.25 | |
| B | 0.13 | 0.28 |
| D | 23.20 BSC | |
| $D_1$ | 19.80 | 20.00 |
| E | 17.2 BSC | |
| E1 | 13.80 | 14.10 |
| e | 0.50 BSC | |
| L | 0.73 | 1.03 |
| N | 128 | |
| ND | 38 | |
| NE | 26 | |

**Notes:**

6. All dimensions and tolerances conform to ANSI Y14.5M-1982.
7. Controlling Dimension is millimeters
8. Dimension “B” does not include dambar protrusion. Allowable dambar protrusion shall be .08mm (.003in.) maximum in excess of the “B” dimension. Dambar cannot be located on the lower radius or the foot.
9. “L” is the length of terminal for soldering to a substrate.
10. “B” includes lead finish thickness.



# 9. Revision History

3/3/99: First Release, V1.0

4/2/99: Second Release, V1.1: p. 1, added DPMS support & adaptive input filtering selection; p. 13, added $GPO_0$ description; p. 17, moved OHO & OVO to list of registers used by on-board processor; p. 19-24, corrected typo on register names & added better DSICAL description; p. 66, added reminder about loop delay.

7/16/99: Third Release, V1.2: p. 10, 11, 16: documented need to ground “reserved” pins; p. 11 & 13, EXADSEL sense changed from early silicon to high true input (high selects external A/D on FS403); p. 19-24, corrected typo on register names & added better DSICAL description, description of new SCR bits, new HOHOS & HOVOS registers, SFLK change from register to bits in SCR; p. 26-27, new software reset values and new HOHOS, HOVOS registers & SCR bits; p. 38, proper VGAINTDET definition; p. 53, new SCR definitions; p. 55-57, new SSR, HCRS, and HCRES bit definitions; p. 58-63, HPO, VPO, HSS, VSS, HPP, VPP all require up to 25ms (vs. 10ms) to calculate limits; ; p. 68-69, new HOHOS & HOVOS registers definitions; p. 71-73, new Configuration Register definitions; p. 87, comment about Philips vs. FOCUS SIO addressing added; p. 88, errors in table corrected (p. 15 was correct, not table 5); p. 103, part number change.

9/2/99: Fourth Release, V1.3: p. 13, documented internal pull-down on EXADSEL; p. 14, corrected $C_{COMP}$ pull down; p. 96, deleted reference to FS_CKIN to FC_CKOUT resistor, resistor not recommended; p. 103, package marking.

7/25/00: Fifth Release, V1.4: p. 1, new patent received; p. 5, corrected reference to external VGACKIN source; p. 19 section 4.2.1.2, corrected first section regarding DSICAL bit (deleted reference to ICALRDY bit); p. 25, Status Port changed to Status Register; p. 84, corrected oscillator settings for Super NTSC and Super PAL; p. 91, 92: $V_{REF}$ changed to 1.276V typical, new min/max.

1/23/03: Sixth Release, V1.5: Table 2, corrected FS403 signal assignments on pin 24 (now $G_6$) and pin 26 (now $G_4$). Section 3, corrected descriptions of FS403 pin 24 (now $G_6$) and pin 26 (now $G_4$). Section 4.5.25, refined the definition of the GPO register.

11/27/05: Seventh Release, V1.6 Table 8.2 modified to include FS403LF dimensions. Removed unreferenced rows in that table.

01/17/2007: Eight Release, V1.7. Leaded versions of FS401 and FS403 discontinued – removed all references to leaded version of these parts . FS402 discontinued, all references to FS402 removed from data sheet. Page 8, section 1.5.3 (FS402 application description) deleted. Page 52, removed reference to FS402 in register 34 Description. Page 75, removed reference to FS402 in Figure 5 description. Page 85, clarification on last sentence in section 5.3.6. Page 101, removed reference to FS402 in section 8.1. Page 102, removed reference to FS403 (leaded version), updated package drawing to eliminate confusion on pin 1 location. Page 104, removed reference to parts 444-2121, 444-2122, 444-2123 (non ROHS compliant parts), updated corporate contact information.

# 10. Ordering Information

| Order Number | Temperature Range | Screening | Package | Package Marking |
|---|---|---|---|---|
| 444-2121LF | 0°C to 70°C | Commercial | 100 Lead PQFP | FS401LF |
| 444-2123LF | 0°C to 70°C | Commercial | 128 Lead PQFP | FS403LF |

444-2121LF and 444-2123LF are RoHS compliant.

Additionally, a FS400 Hardware Development Kit is available for purchase, Order # 444-2020.

## 10.1 Package Markings:

FOCUS
Enhancements
FS40XLF
<Date Code (YYWWLA) and Revision Letter>
<fab lot id>
where X = 1 or 3.

**Please forward suggestions and corrections as soon as possible to info@focusinfo.com. The information herein is accurate to the best of FOCUS' knowledge, but not all specifications have been characterized or tested at the time of the release of this document. Parameters will be updated as soon as possible and updates made available.**

For further information on the FS400 family of scan converters, contact your Focus representative and request reference schematics, application notes, and source code (as appropriate).



### Critical Applications Policy

Focus Enhancements components are not designed for use in Critical Applications. Critical Applications are products whose use may involve risks of death, personal injury, severe property damage or environmental damage or life support applications, devices, or systems, wherein a failure or malfunction of the component can reasonably be expected to result in death or personal injury. The user of Focus Enhancements components in Critical Applications assumes all risk of such use and indemnifies Focus Enhancements against all damages.

**Focus enhancements – Semiconductor Group**
22867 NW Bennett St., Suite #200
Hillsboro, OR 97124 U.S.A.
Phone: (503) 615-7700
Fax: (503) 615-4232
Website: www.FOCUSsemi.com