

# 1/4-Inch Color CMOS NTSC/PAL Digital Image SOC with Overlay Processor

## MT9V136 Datasheet, Rev. J

For the latest datasheet, please visit [www.onsemi.com](http://www.onsemi.com)

### Features

- Low-power CMOS image sensor with integrated image flow processor (IFP) and video encoder
- 1/4-inch optical format, VGA resolution (640H x 480V)
- $\pm 2.5\%$  additional columns and rows to compensate for lens alignment tolerances
- Overlay generator for dynamic bitmap overlay
- Integrated video encoder for NTSC/PAL with overlay capability and 10-bit I-DAC
- Integrated microcontroller for flexibility
- On-chip image flow processor performs sophisticated processing, such as color recovery and correction, sharpening, gamma, lens shading correction, on-the-fly defect correction, auto white balancing, and auto exposure
- Auto black level calibration
- 10-bit, on-chip analog-to-digital converter (ADC)
- Internal master clock generated by on-chip phase-locked loop (PLL)
- Two-wire serial programming interface
- Interface to low-cost Flash through SPI bus
- High-level host command interface
- Stand alone operation support
- Comprehensive tool support for overlay generation and lens correction setup
- Development system with DevWare
- Overlay generation and compilation tools

### Applications

- Analog surveillance CCTV
- Surveillance network IP camera

**Table 1: Key Parameters**

| Parameter             | Typical Value   |
|-----------------------|---|
| Pixel size and type   | 5.6 $\mu\text{m}$ x 5.6 $\mu\text{m}$ active pinned-photodiode with high-sensitivity mode for low-light conditions  |
| Sensor format         | 680H x 512V (includes $\pm 2.5\%$ of rows and columns for lens alignment)   |
| NTSC output           | 720H x 480V   |
| PAL output            | 720H x 576V   |
| Imaging area          | Total array size: 3.584 mm x 2.688 mm   |
| Optical format        | ¼-inch  |
| Frame rate            | 50/60 fields/sec  |
| Sensor scan mode      | Progressive scan  |
| Color filter array    | RGB standard Bayer  |
| Shutter type          | Electronic rolling shutter (ERS)  |
| Automatic Functions   | Exposure, white balance, black level offset correction, flicker avoidance, color saturation control, on-the-fly defect correction, aperture correction  |
| Programmable Controls | Exposure, white balance, horizontal and vertical blanking, color, sharpness, gamma correction, lens shading correction, horizontal and vertical image flip, windowing, sampling rates, GPIO control |

Key parameters are continued on next page.

See details of new features on page 3.

See “Ordering Information” on page 4.

**Table 2: Key Parameters (continued)**

| Parameter                        | Typical Value  |      |
|----------------------------------|--|------|
| Overlay Support <sup>1</sup>     | Utilizes SPI interface to load overlay data from external flash/EEPROM memory with the following features: <ul style="list-style-type: none"> <li>•Overlay Size 360 x 480 pixel rendered into 720 x 480 pixel display format</li> <li>•Up to four (4) overlays may be blended simultaneously</li> <li>•Selectable readout: Rotating order user selected</li> <li>•Dynamic scenes by loading pre-rendered frames from external memory</li> <li>•Palette of 32 colors out of 64,000</li> <li>•8 colors per bitmap</li> <li>•Blend factor dynamically programmable for smooth transitions</li> <li>•Fast Update rate of up to 30 fps</li> <li>•Every bitmap object has independent x/y position</li> <li>•Statistic Engine to calibrate optical alignment</li> <li>•Number Generator</li> </ul> |      |
| Windowing                        | Programmable to any size   |      |
| Max analog gain                  | 0.5–16x  |      |
| ADC                              | 10-bit, on-chip  |      |
| Output interface                 | Analog composite video out, single-ended or differential; 8-, 10-bit parallel digital output   |      |
| Output data formats <sup>1</sup> | Digital: Raw Bayer 8-,10-bit, CCIR656, 565RGB, 555RGB, 444RGB  |      |
| Data rate                        | Parallel: 27 MB/s  |      |
|                                  | NTSC: 60 fields/sec  |      |
|                                  | PAL: 50 fields/sec   |      |
| Control interface                | Two-wire I/F for register interface plus high-level command exchange. SPI port to interface to external memory to load overlay data, register settings, or firmware extensions.  |      |
| Input clock for PLL              | 27 MHz   |      |
| SPI Clock Frequencies            | 4.5 - 9.0 - 18 MHz, programmable   |      |
| Supply voltage                   | Analog: 2.8 V ±5%  |      |
|                                  | Core: 1.8 V ±5%  |      |
|                                  | IO: 2.8V ±5%   |      |
| Power consumption                | Full resolution at 60 fps: <350mW <sup>2</sup>   |      |
| Package                          | 48-pin Ceramic LCC, 11.43mm x 11.43mm, 0.8mm pitch   |      |
| Ambient temperature              | Operating: –30°C to 70°C   |      |
|                                  | Storage: –50°C to +150°C   |      |
| Dark Current                     | < 200e/s at 60°C with a gain of 1  |      |
| Fixed pattern noise              | Column   | < 2% |
|                                  | Row  | < 2% |
| Responsivity                     | 11.9 V/lux-s at 550nm  |      |
| Signal to noise ratio (S/N)      | 45 dB  |      |
| Pixel dynamic range              | 74.6 dB  |      |

- Notes: 1. Graphical overlay is available only in CCIR656 output format.  
2. Analog output enabled; parallel output disabled.

---

## New Features

### Integrated Video Encoder for PAL/NTSC with Overlay Capability

- Composite analog output (NTSC/PAL)
- 8-bit parallel digital output ITU-R BT.656 format
- Raw Bayer format

### On-Chip Overlay Generator

- Static and dynamic overlay graphics with four overlay planes plus number plane
- Support for serial SPI memory up to 16 megabytes
- Number generator
- Overlay blending and x/y positioning
- Overlay position adjustment and statistics engine to calibrate overlay
- Overlay support utilizes SPI interface to load overlay data from external Serial Flash/EEPROM to support the following features:
  - Overlay size 360 x 480 pixel rendered into 720 x 480 pixel display format
  - Up to four overlays may be blended simultaneously
  - Selectable readout: rotating order user selected
  - Dynamic scenes by loading pre-rendered frames from external memory
  - Palette of 32 colors out of 64,000
  - Eight colors per bitmap
  - Blend factor dynamically programmable for smooth transitions
  - Fast update rate of up to 30 fps
  - Every bitmap object has independent x/y position
  - Statistics engine to calibrate optical alignment

## Ordering Information

**Table 3: Available Part Numbers**

| Part Number             | Product Description              | Orderable Product Attribute Description |
|-------------------------|----------------------------------|---|
| MT9V136C12STC-DR        | Color VGA 1/4" SOC, CLCC Package | Dry Pack without Protective Film        |
| MT9V136D00STCK22BC1-200 | Color VGA 1/4" SOC               | Die Sales, 200µm Thickness              |
| MT9V136W00STCK22BC1-750 | Color VGA 1/4" SOC               | Wafer Sales, 750µm Thickness            |

See the ON Semiconductor Device Nomenclature document (TND310/D) for a full description of the naming convention used for image sensors. For reference documentation, including information on evaluation kits, please visit our web site at [www.onsemi.com](http://www.onsemi.com).

**Table of Contents**

Features . . . . . 1

Applications . . . . . 1

New Features . . . . . 3

Ordering Information . . . . . 4

General Description . . . . . 9

Architecture . . . . . 10

System Block Diagram . . . . . 11

Pin Descriptions and Assignments . . . . . 13

SOC Description . . . . . 17

Sensor Active Pixel Array . . . . . 20

Usage Modes . . . . . 31

External Signal Processing . . . . . 33

Slave Two-Wire Serial Interface . . . . . 42

Overlay Capability . . . . . 47

Serial Memory Partition . . . . . 48

Overlay Adjustment . . . . . 49

Overlay Character Generator . . . . . 51

Modes and Timing . . . . . 54

Electrical Specifications . . . . . 68

Spectral Characteristics . . . . . 78

Revision History . . . . . 80

## List of Figures

|            |   |    |
|------------|---|----|
| Figure 1:  | Internal Block Diagram . . . . .  | 10 |
| Figure 2:  | System Block Diagram . . . . .  | 11 |
| Figure 3:  | Using a Crystal Instead of an External Oscillator . . . . .                         | 12 |
| Figure 4:  | Pin Assignments . . . . .   | 15 |
| Figure 5:  | Sensor Core Block Diagram . . . . .   | 17 |
| Figure 6:  | Clear Pixel Array Description . . . . .   | 18 |
| Figure 7:  | Image Capture Example . . . . .   | 19 |
| Figure 8:  | Active Pixel Array . . . . .  | 20 |
| Figure 9:  | Pixel Color Pattern Detail (top right corner) . . . . .                             | 21 |
| Figure 10: | Spatial Illustration of Image Readout . . . . .                                     | 22 |
| Figure 11: | Color Pipeline . . . . .  | 23 |
| Figure 12: | Color Bar Test Pattern . . . . .  | 24 |
| Figure 13: | Color Bars . . . . .  | 25 |
| Figure 14: | Gamma Correction Curve . . . . .  | 28 |
| Figure 15: | Auto-Config Mode . . . . .  | 31 |
| Figure 16: | Flash Mode . . . . .  | 31 |
| Figure 17: | Host Mode with Flash . . . . .  | 32 |
| Figure 18: | Host Mode . . . . .   | 32 |
| Figure 19: | External Signal Processing Block Diagram . . . . .                                  | 33 |
| Figure 20: | Power-Up Sequence – Configuration Options Flow Chart . . . . .                      | 35 |
| Figure 21: | Interface Structure . . . . .   | 37 |
| Figure 22: | Single READ from Random Location . . . . .  | 44 |
| Figure 23: | Single Read from Current Location . . . . .   | 44 |
| Figure 24: | Sequential READ, Start from Random Location . . . . .                               | 45 |
| Figure 25: | Sequential READ, Start from Current Location . . . . .                              | 45 |
| Figure 26: | Single WRITE to Random Location . . . . .   | 45 |
| Figure 27: | Sequential WRITE, Start at Random Location . . . . .                                | 46 |
| Figure 28: | Overlay Data Flow . . . . .   | 47 |
| Figure 29: | Memory Partitioning . . . . .   | 48 |
| Figure 30: | Overlay Calibration . . . . .   | 50 |
| Figure 31: | Internal Block Diagram Overlay . . . . .  | 51 |
| Figure 32: | Example of Character Descriptor 0 Stored in ROM . . . . .                           | 52 |
| Figure 33: | Full Character Set for Overlay . . . . .  | 53 |
| Figure 34: | Single-Ended Termination . . . . .  | 54 |
| Figure 35: | Differential Connection—Grounded Termination . . . . .                              | 55 |
| Figure 36: | CCIR656 8-Bit Parallel Interface Format for 525/60 (625/50) Video Systems . . . . . | 55 |
| Figure 37: | Typical CCIR656 Vertical Blanking Intervals for 525/60 Video System . . . . .       | 56 |
| Figure 38: | Typical CCIR656 Vertical Blanking Intervals for 625/50 Video System . . . . .       | 57 |
| Figure 39: | Primary Clock Relationships . . . . .   | 58 |
| Figure 40: | Typical I/O Equivalent Circuits . . . . .   | 60 |
| Figure 41: | NTSC Block . . . . .  | 61 |
| Figure 42: | Serial Interface . . . . .  | 61 |
| Figure 43: | Digital Output I/O Timing . . . . .   | 62 |
| Figure 44: | Slew Rate Timing . . . . .  | 63 |
| Figure 45: | Configuration Timing . . . . .  | 64 |
| Figure 46: | Power Up Sequence . . . . .   | 65 |
| Figure 47: | Power Down Sequence . . . . .   | 66 |
| Figure 48: | Reset to SPI Access Delay . . . . .   | 66 |
| Figure 49: | Reset to Serial Access Delay . . . . .  | 67 |
| Figure 50: | Reset to AE/AWB Image . . . . .   | 67 |
| Figure 51: | SPI Output Timing . . . . .   | 68 |
| Figure 52: | Video Timing . . . . .  | 74 |
| Figure 53: | Equivalent Pulse . . . . .  | 75 |
| Figure 54: | V Pulse . . . . .   | 76 |
| Figure 55: | Two-Wire Serial Bus Timing Parameters . . . . .                                     | 77 |
| Figure 56: | Quantum Efficiency . . . . .  | 78 |

Figure 57: 48-Pin CLCC Package Outline Drawing.....79

**List of Tables**

Table 1: Key Parameters . . . . . 1  
 Table 2: Key Parameters (continued). . . . . 2  
 Table 3: Available Part Numbers . . . . . 4  
 Table 4: Pin Descriptions . . . . . 13  
 Table 5: Reset/Default State of Interfaces. . . . . 15  
 Table 6: EIA Color Bars (NTSC) . . . . . 25  
 Table 7: EBU Color Bars (PAL). . . . . 25  
 Table 8: NTSC . . . . . 25  
 Table 9: PAL. . . . . 26  
 Table 10: YCbCr Output Data Ordering. . . . . 29  
 Table 11: RGB Ordering in Default Mode . . . . . 29  
 Table 12: 2-Byte Bayer Format . . . . . 29  
 Table 13: SPI Flash Devices . . . . . 35  
 Table 14: SPI Commands Supported . . . . . 36  
 Table 15: GPIO Bit Descriptions . . . . . 36  
 Table 16: System Manager Commands . . . . . 39  
 Table 17: Overlay Host Commands . . . . . 39  
 Table 18: GPIO Host Commands . . . . . 40  
 Table 19: Flash Manager Host Commands. . . . . 40  
 Table 20: Sequencer Host Commands. . . . . 40  
 Table 21: TX Manager Host Commands . . . . . 41  
 Table 22: Two-Wire Interface ID Address Switching . . . . . 42  
 Table 23: Transfer Time Estimate . . . . . 48  
 Table 24: Character Generator Details. . . . . 53  
 Table 25: Field, Vertical Blanking, EAV, and SAV States 525/60 Video System . . . . . 56  
 Table 26: Field, Vertical Blanking, EAV, and SAV States for 625/50 Video System . . . . . 57  
 Table 27: Output Data Ordering in DOUT RGB Mode . . . . . 59  
 Table 28: Output Data Ordering in Sensor Stand-Alone Mode . . . . . 59  
 Table 29: Parallel Digital Output I/O Timing . . . . . 62  
 Table 30: Slew Rate for PIXCLK and DOUT . . . . . 63  
 Table 31: Configuration Timing . . . . . 64  
 Table 32: Power Up Sequence . . . . . 65  
 Table 33: Power Down Sequence . . . . . 66  
 Table 34: RESET\_BAR Delay Parameters. . . . . 67  
 Table 35: SPI Data Setup and Hold Timing. . . . . 68  
 Table 36: Absolute Maximum Ratings . . . . . 69  
 Table 37: Electrical Characteristics and Operating Conditions . . . . . 69  
 Table 38: Video DAC Electrical Characteristics–Single-Ended Mode. . . . . 70  
 Table 39: Video DAC Electrical Characteristics–Differential Mode. . . . . 70  
 Table 40: Digital I/O Parameters. . . . . 71  
 Table 41: Power Consumption – Condition 1 . . . . . 72  
 Table 42: Power Consumption – Condition 2 . . . . . 72  
 Table 43: NTSC Signal Parameters . . . . . 73  
 Table 44: Video Timing . . . . . 74  
 Table 45: Equivalent Pulse . . . . . 75  
 Table 46: V Pulse. . . . . 76  
 Table 47: Two-Wire Serial Bus Characteristics . . . . . 77



## General Description

The ON Semiconductor MT9V136 is a VGA-format, single-chip CMOS active-pixel digital image sensor for surveillance applications. It captures high-quality color images at VGA resolution and outputs NTSC or PAL interlaced composite video.

The VGA CMOS image sensor features ON Semiconductor's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, low power, and integration advantages of ON Semiconductor's advanced active pixel CMOS process technology.

The MT9V136 is a complete camera-on-a-chip. It incorporates sophisticated camera functions on-chip and is programmable through a simple two-wire serial interface or by an attached SPI Flash memory that contains setup information that may be loaded automatically at startup.

The MT9V136 performs sophisticated processing functions including color recovery, color correction, sharpening, programmable gamma correction, auto black reference clamping, auto exposure, 50Hz/60Hz flicker avoidance, lens shading correction, auto white balance (AWB), and on-the-fly defect identification and correction.

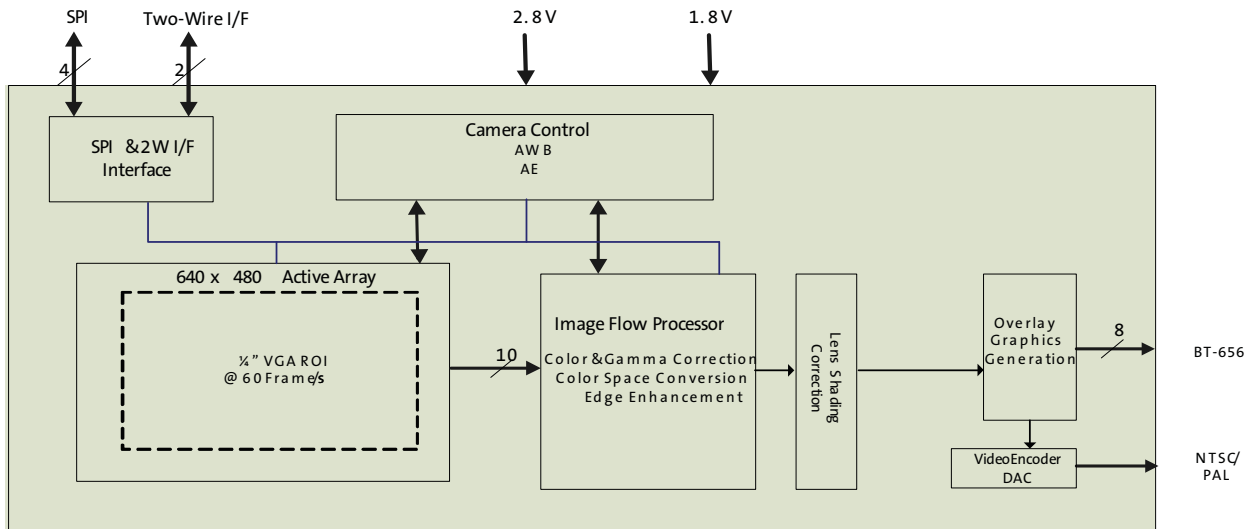
The MT9V136 outputs interlaced-scan images at 30 or 25 fps, supporting both NTSC and PAL video formats. The image data can be output on one or two output ports:

- Composite analog video (single-ended and differential output support)
- Parallel 8-, 10-bit digital

## Architecture

### Internal Block Diagram

Figure 1: Internal Block Diagram

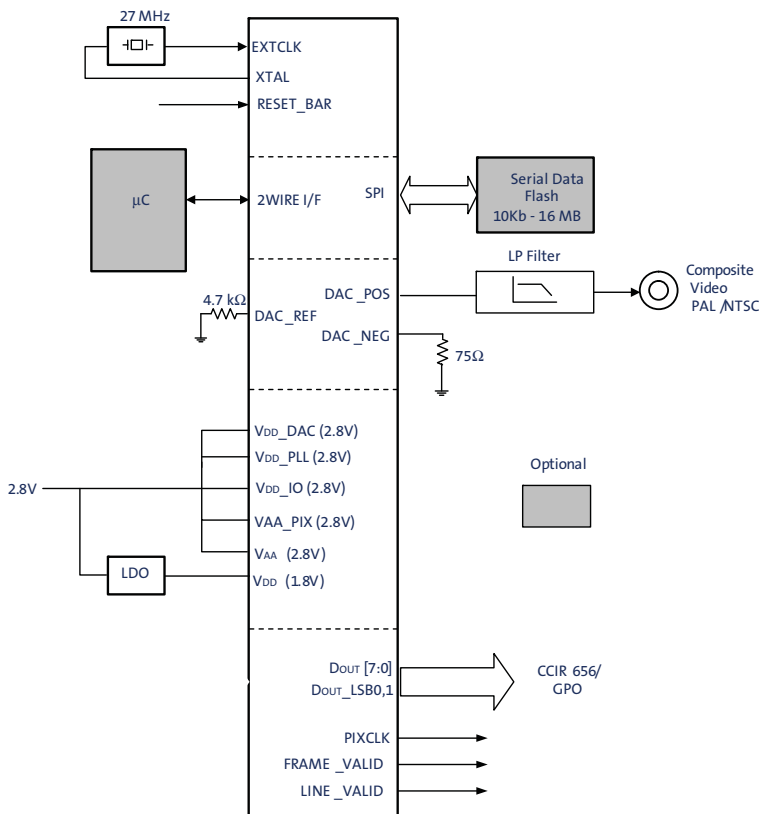


**Note:** The active array is smaller than the sensor array.

## System Block Diagram

The system block diagram will depend on the application. The system block diagram in Figure 2 shows all components; optional peripheral components are highlighted. The optional microcontroller controls the MT9V136 sensor using the two-wire serial bus. Optional components will vary by application. For further details, see the MT9V136 Register and Variable Reference.

Figure 2: System Block Diagram

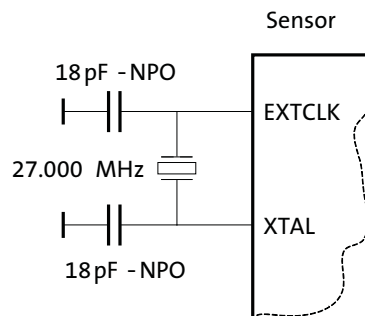


### Crystal Usage

As an alternative to using an external oscillator, a fundamental 27 MHz crystal may be connected between EXTCLK and XTAL. Two small loading capacitors of 15–22pF of NPO dielectric should be added as shown in Figure 3.

ON Semiconductor does not recommend using the crystal option for applications above 85°C. A crystal oscillator with temperature compensation is recommended.

**Figure 3: Using a Crystal Instead of an External Oscillator**



When using Xtal as the clock source, the internal inverter circuit has a 100K bias resistor in parallel to Xtal, which can be connected or disconnected by register 0x0014 bit[14]. The clockin\_bias\_en bit is set to 1 by default.

## Pin Descriptions and Assignments

Table 4: Pin Descriptions

| Pin Number                           | Pin Name    | Type         | Description   |
|--------------------------------------|-------------|--------------|---|
| <b>Clock and Reset</b>               |             |              |   |
| 9                                    | EXTCLK      | Input        | Master input clock (27MHz): This either can be a square-wave generated from an oscillator (in which case the XTAL input must be left unconnected) or connected directly to a crystal.   |
| 10                                   | XTAL        | Output       | If EXTCLK is connected to one pin of a crystal, this signal is connected to the other pin; otherwise this signal must be left unconnected.  |
| 12                                   | RESET_BAR   | Input        | Asynchronous active-low reset: When asserted, the device will return all interfaces to their reset state. When released, the device will initiate the boot sequence.  |
| <b>Register Interface</b>            |             |              |   |
| 17                                   | SCLK        | Input        | These two signals implement serial communications protocol for access to the internal register set and memory.  |
| 18                                   | SDATA       | Input/OD     |   |
| 16                                   | SADDR       | Input        | This signal controls the device ID that will respond to serial communication commands.<br>Two-wire serial interface device ID selection:<br>0: 0x90<br>1: 0xBA  |
| <b>SPI Interface</b>                 |             |              |   |
| 22                                   | SPI_SCLK    | Output       | Clock output for interfacing to an external SPI memory such as Flash/EEPROM. Tristated when RESET_BAR is asserted.  |
| 21                                   | SPI_SDI     | Input        | Data in from SPI device. This signal has an internal pull-up resistor.  |
| 20                                   | SPI_SDO     | Output       | Data out to SPI device. Tristated when RESET_BAR is asserted.   |
| 19                                   | SPI_CS_N    | Output       | Chip selects to SPI device. Tristated when RESET_BAR is asserted.   |
| <b>(Parallel) Pixel Data Output</b>  |             |              |   |
| 32                                   | FRAME_VALID | Input/Output | Pixel data from the MT9V136 can be routed out on this interface and processed externally.<br>To save power, these signals are driven to a constant logic level unless the parallel pixel data output or alternate (GPIO) function is enabled for these pins.<br>This interface is disabled by default.<br>The slew rate of these outputs is programmable.<br>These signals can also be used as general purpose input/outputs.   |
| 31                                   | LINE_VALID  | Input/Output |   |
| 33                                   | PIXCLK      | Output       |   |
| 39, 40, 41,<br>42, 43, 44,<br>45, 46 | DOUT[7:0]   | Output       |   |
| 38                                   | DOUT_LSB1   | Input/Output | When the sensor core is running in bypass mode, it will generate 10 bits of output data per pixel. These two pins make the two LSB of pixel data available externally. Leave DOUT_LSB1 unconnected if not used. To save power, these signals are driven to a constant logic level unless the sensor core is running in bypass mode or the alternate function is enabled for these pins. The slew rate of these outputs is programmable. For analog output, the DOUT_LSB0 cannot be left unconnected, and must be strapped to select either NTSC or PAL mode. For more information, see Table 15, "GPIO Bit Descriptions," on page 36. |
| 37                                   | DOUT_LSB0   | Input/Output |   |

**Table 4: Pin Descriptions (continued)**

| Pin Number                          | Pin Name | Type   | Description  |
|-------------------------------------|----------|--------|--|
| <b>Composite Video Output</b>       |          |        |  |
| 6                                   | DAC_POS  | Output | Positive video DAC output in differential mode. Video DAC output in single-ended mode. This interface is enabled by default using NTSC/PAL signaling. For applications where composite video output is not required, the video DAC can be placed in a power-down state under software control. |
| 4                                   | DAC_NEG  | Output | Negative video DAC output in differential mode. Connect to AGND in single-ended mode.  |
| 2                                   | DAC_REF  | Output | External reference resistor for the video DAC.   |
| <b>Manufacturing Test Interface</b> |          |        |  |
| 27                                  | TDI      | Input  | JTAG Test pin (Reserved for Test Mode)   |
| 26                                  | TDO      | Output | JTAG Test pin (Reserved for Test Mode)   |
| 25                                  | TMS      | Input  | JTAG Test pin (Reserved for Test Mode)   |
| 24                                  | TCK      | Input  | JTAG Test pin (Reserved for Test Mode)   |
| 23                                  | TRST_N   | Input  | Connect to GND   |
| <b>Power</b>                        |          |        |  |
| 8, 14, 35, 48                       | DGND     | Supply | Digital ground.  |
| 3                                   | GND_DAC  | Supply | Video DAC GND  |
| 1, 7, 15, 34                        | VDD      | Supply | Supply for VDD core: 1.8V nominal.   |
| 13, 36, 47                          | VDD_IO   | Supply | Supply for digital IOs: 2.8V nominal.  |
| 5                                   | VDD_DAC  | Supply | Supply for video DAC: 2.8V nominal.  |
| 11                                  | VDD_PLL  | Supply | Supply for PLL: 2.8V nominal.  |
| 29                                  | AGND     | Supply | Analog ground.   |
| 28                                  | VAA      | Supply | Analog power: 2.8V nominal.  |
| 30                                  | VAA_PIX  | Supply | Analog pixel array power: 2.8V nominal. Must be at same voltage potential as VAA.  |

## Pin Assignments

Figure 4: Pin Assignments

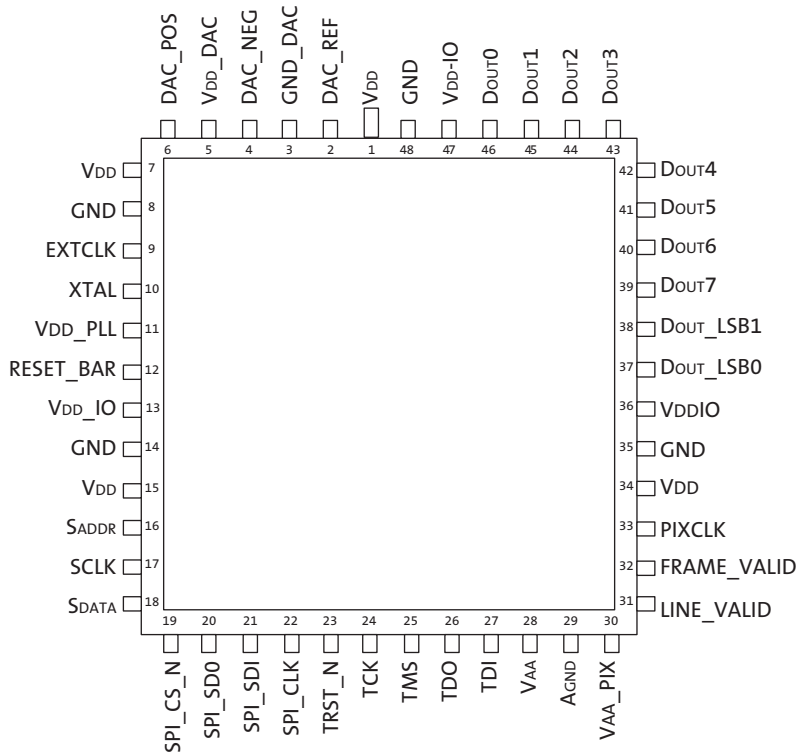


Table 5: Reset/Default State of Interfaces

| Name      | Reset State               | Default State            | Notes   |
|-----------|---------------------------|--------------------------|---|
| EXTCLK    | Clock running or stopped  | Clock running            | Input   |
| XTAL      | N/A                       | N/A                      | Input   |
| RESET_BAR | Asserted                  | De-asserted              | Input   |
| SCLK      | N/A                       | N/A                      | Input. Must always be driven to a valid logic level.  |
| SDATA     | High impedance            | High impedance           | Input/Output. A valid logic level should be established by pull-up resistor.                    |
| SADDR     | N/A                       | N/A                      | Input. Must always be driven to a valid logic level. Must be permanently tied to VDD_IO or GND. |
| SPI_SCLK  | High impedance.           | Driven, logic 0          | Output. Output enable is R0x0032[9].  |
| SPI_SDI   | Internal pull-up enabled. | Internal pull-up enabled | Input. Internal pull-up is permanently enabled.   |
| SPI_SDO   | High impedance            | Driven, logic 0          | Output enable is R0x0032[9].  |
| SPI_CS_N  | High impedance            | Driven, logic 1          | Output enable is R0x0032[9].  |

**Table 5: Reset/Default State of Interfaces (continued)**

| Name        | Reset State              | Default State            | Notes  |
|-------------|--------------------------|--------------------------|--|
| FRAME_VALID | High impedance           | High impedance           | Input/Output. This interface disabled by default. Input buffers (used for GPIO function) powered down by default, so these pins can be left unconnected (floating). After reset, these pins are powered up, sampled, then powered down again as part of the auto-configuration mechanism. See Note 2.  |
| LINE_VALID  |                          |                          |  |
| PIXCLK      | High impedance           | Driven, logic 0          | Output. This interface disabled by default. See Note 1.  |
| DOUT7       |                          |                          |  |
| DOUT6       |                          |                          |  |
| DOUT5       |                          |                          |  |
| DOUT4       |                          |                          |  |
| DOUT3       |                          |                          |  |
| DOUT2       |                          |                          |  |
| DOUT1       |                          |                          |  |
| DOUT0       |                          |                          |  |
| DOUT_LSB1   | High impedance           | High impedance           | Input/Output. This interface disabled by default. Input buffers (used for GPIO function) powered down by default, so these pins can be left unconnected (floating). After reset, these pins are powered-up, sampled, then powered down again as part of the auto-configuration mechanism. For analog output, the DOUT_LSB0 cannot be left unconnected, and must be strapped to select either NTSC or PAL mode. |
| DOUT_LSB0   | High impedance           | Driven, logic 0          |  |
| DAC_POS     | High impedance           | Driven                   | Output. Interface disabled by hardware reset and enabled by default when the device starts streaming.  |
| DAC_NEG     |                          |                          |  |
| DAC_REF     |                          |                          |  |
| TDI         | Internal pull-up enabled | Internal pull-up enabled | Input. Internal pull-up means that this pin can be left unconnected (floating).  |
| TDO         | High impedance           | High impedance           | Output. Driven only during appropriate parts of the JTAG shifter sequence.   |
| TMS         | Internal pull-up enabled | Internal pull-up enabled | Input. Internal pull-up means that this pin can be left unconnected (floating).  |
| TCK         | Internal pull-up enabled | Internal pull-up enabled | Input. Internal pull-up means that this pin can be left unconnected (floating).  |
| TRST_N      | N/A                      | N/A                      | Input. Must always be driven to a valid logic level. Must be driven to GND for normal operation.   |

- Notes:
1. The reason for defining the default state as logic 0 rather than high impedance is this: when wired in a system (for example, on our demo boards), these outputs will be connected, and the inputs to which they are connected will want to see a valid logic level. No current drain should result from driving these to a valid logic level (unless there is a pull-up at the system level).
  2. These pads have their input circuitry powered down, but they are not output-enabled. Therefore, they can be left floating but they will not drive a valid logic level to an attached device.



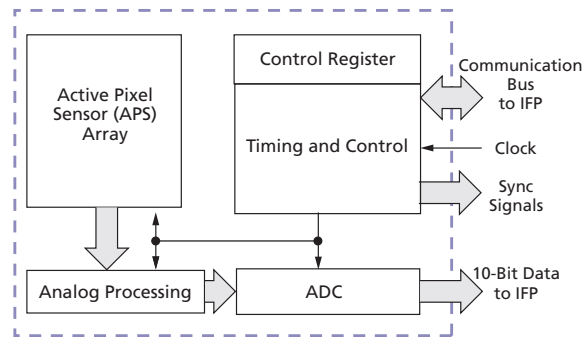
## SOC Description

### Detailed Architecture Overview

#### Sensor Core

The sensor consists of a pixel array, an analog readout chain, a 10-bit ADC with programmable gain and black offset, and timing and control as illustrated in Figure 5.

**Figure 5: Sensor Core Block Diagram**



Pixel Array Structure

The sensor core Clear Pixel Array is configured as 744 columns by 514 rows, as shown in Figure 6. The First Clear pixel can be identifiable under a microscope. The First Active pixel location can be determined from the offset to the First Clear pixel. The Active Pixel Array (680 x 512) is bounded by the First Active pixel and the Last Active pixel. The optical center is the center of the Active Pixel Array. There are inactive pixels around the Active Pixel Array.

Figure 6: Clear Pixel Array Description

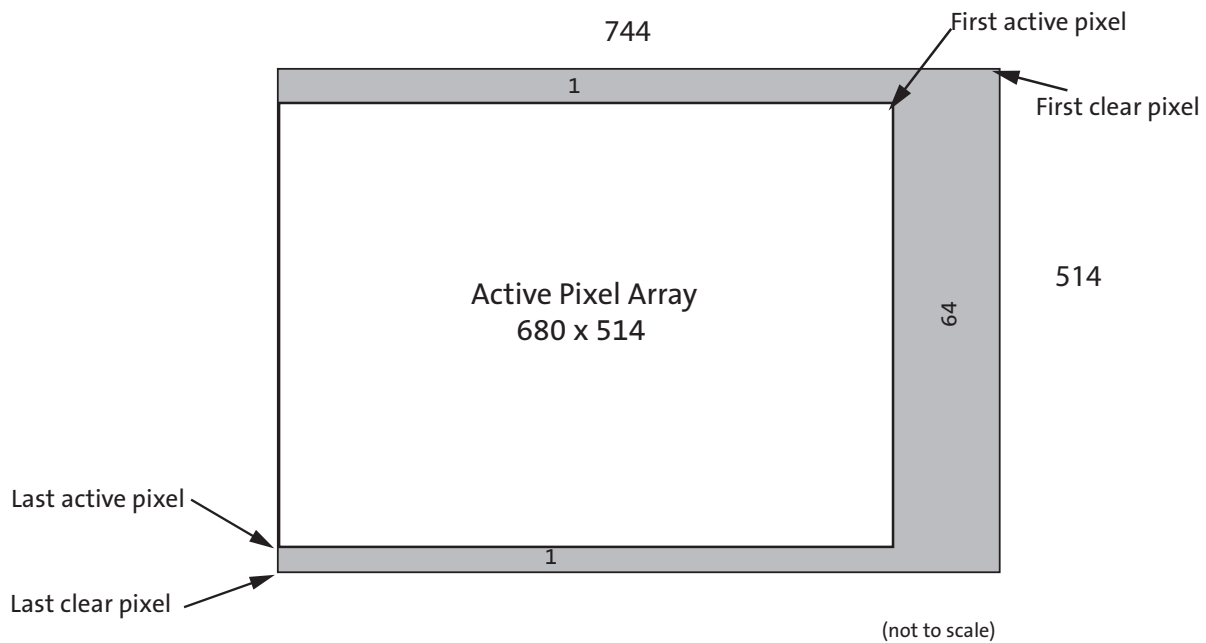
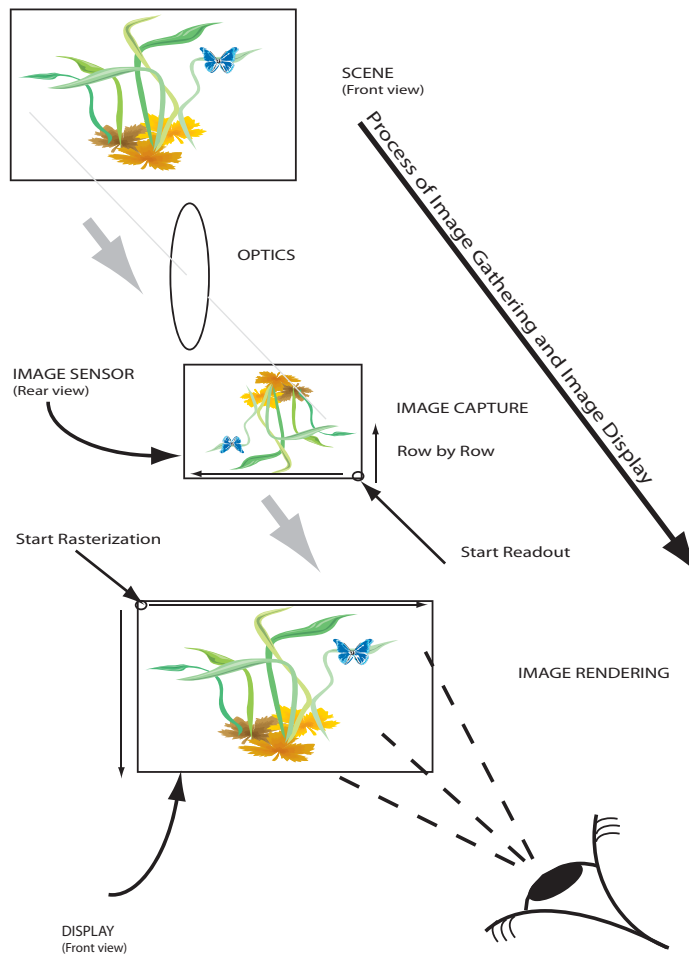


Figure 7 on page 19 illustrates the process of capturing the image. The original scene is flipped and mirrored by the sensor optics. Sensor readout starts at the lower right corner. The image is presented in true orientation by the output display.

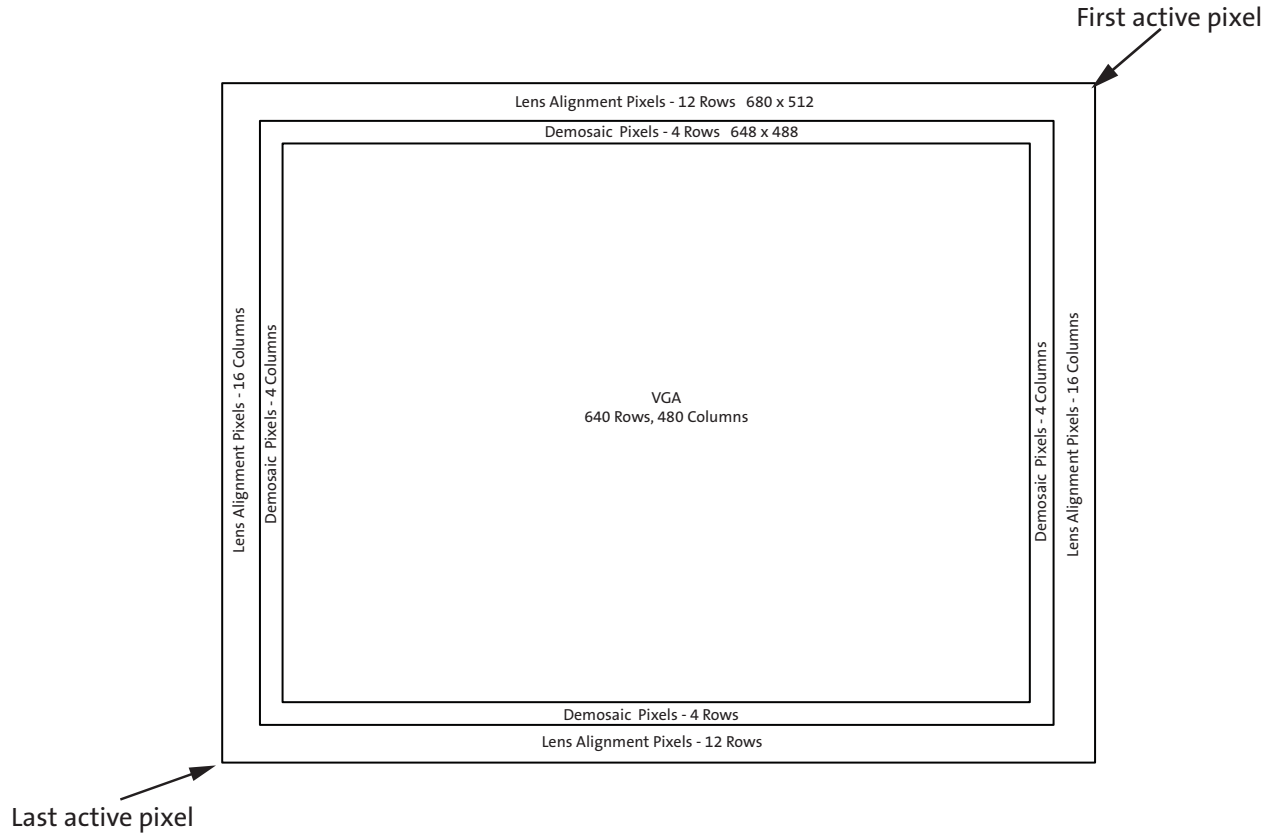
Figure 7: Image Capture Example



## Sensor Active Pixel Array

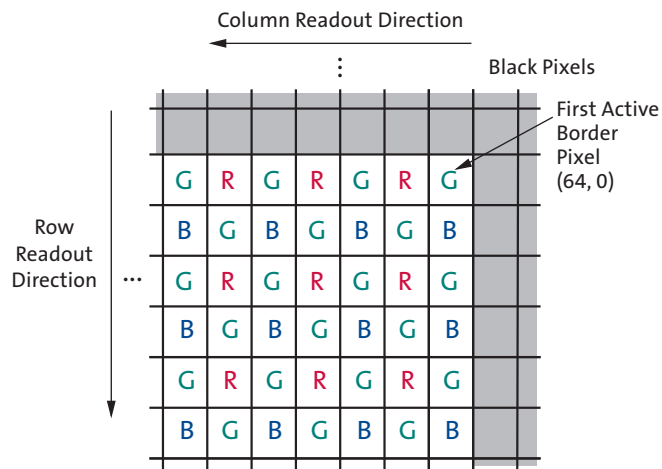
The Active Pixel Array is the area where embedded MCU in SOC programs sensor core hardware registers at R0x3002 - R0x3008. At the First Active pixel location, the Y\_ADDR\_START register (R0x3002) and the X\_ADDR\_START register (R0x3004) are programmed to 0. The default location for these registers are (12, 16) such that the programming image size is at the center of the Active Pixel array. The programming image size at a sensor core is 688 x 488 which includes 4 columns/rows border around it. These extra columns and rows are required for image processing in the image flow processor, IFP. There are additional columns and rows in the Active Pixel array for lens alignment to the optical center to compensate for mechanical tolerance. The programming array (648 x 488) can move +/- 16 columns in the x direction but only +/- 12 rows in the y direction. The window should be moved only at even numbers to preserve the first color.

Figure 8: Active Pixel Array



The range of adjustment is from Row 0 to 22 and Column 0 to 30. There are 4 rows/ columns needed to calculate the RGB values. The window should be moved only at even numbers.

Figure 9: Pixel Color Pattern Detail (top right corner)



## Output Data Format

The sensor core image data are read out in progressive scan order. Valid image data are surrounded by horizontal and vertical blanking, shown in Figure 10.

For NTSC output, the horizontal size is stretched from 640 to 720 pixels. The vertical size is 243 pixels per field; 240 image pixels and 3 dark pixels that are located at the bottom of the image field.

For PAL output, the horizontal size is also stretched from 640 to 720 pixels. The vertical size is 288 pixels per field.

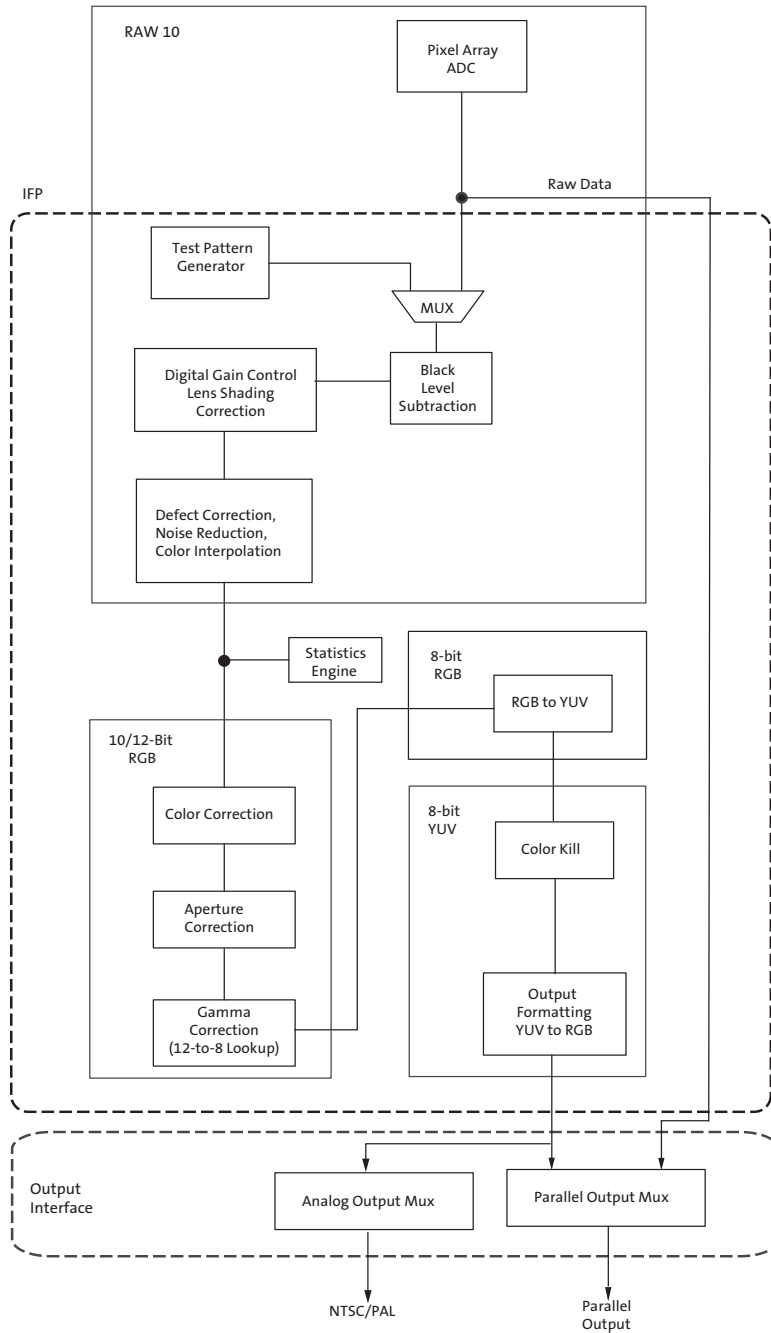
**Figure 10: Spatial Illustration of Image Readout**

|  |  |
|--|--|
| $P_{0,0} P_{0,1} P_{0,2} \dots P_{0,n-1} P_{0,n}$<br>$P_{2,0} P_{2,1} P_{2,2} \dots P_{2,n-1} P_{2,n}$ | 00 00 00 ..... 00 00 00<br>00 00 00 ..... 00 00 00             |
| Valid Image Odd Field  | Horizontal Blanking  |
| $P_{m-2,0} P_{m-2,1} \dots P_{m-2,n-1} P_{m-2,n}$<br>$P_{m,0} P_{m,1} \dots P_{m,n-1} P_{m,n}$         | 00 00 00 ..... 00 00 00<br>00 00 00 ..... 00 00 00             |
| Vertical Even Blanking   | Vertical/Horizontal Blanking                                   |
| $00\ 00\ 00 \dots 00\ 00\ 00$<br>$00\ 00\ 00 \dots 00\ 00\ 00$   | $00\ 00\ 00 \dots 00\ 00\ 00$<br>$00\ 00\ 00 \dots 00\ 00\ 00$ |
| $P_{1,0} P_{1,1} P_{1,2} \dots P_{1,n-1} P_{1,n}$<br>$P_{3,0} P_{3,1} P_{3,2} \dots P_{3,n-1} P_{3,n}$ | 00 00 00 ..... 00 00 00<br>00 00 00 ..... 00 00 00             |
| Valid Image Even Field   | Horizontal Blanking  |
| $P_{m-1,0} P_{m-1,1} \dots P_{m-1,n-1} P_{m-1,n}$<br>$P_{m+1,0} P_{m+1,1} \dots P_{m+1,n-1} P_{m+1,n}$ | 00 00 00 ..... 00 00 00<br>00 00 00 ..... 00 00 00             |
| Vertical Odd Blanking  | Vertical/Horizontal Blanking                                   |
| $00\ 00\ 00 \dots 00\ 00\ 00$<br>$00\ 00\ 00 \dots 00\ 00\ 00$   | $00\ 00\ 00 \dots 00\ 00\ 00$<br>$00\ 00\ 00 \dots 00\ 00\ 00$ |

## Image Flow Processor

Image and color processing in the MT9V136 are implemented as an image flow processor (IFP) coded in hardware logic. During normal operation, the embedded microcontroller will automatically adjust the operation parameters. The IFP is broken down into different sections, as outlined in Figure 11.

Figure 11: Color Pipeline



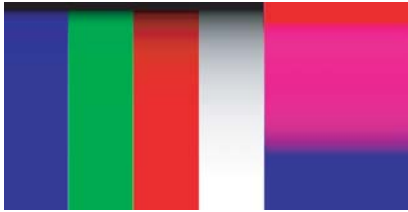




## Test Patterns

During normal operation of the MT9V136, a stream of raw image data from the sensor core is continuously fed into the color pipeline. For test purposes, this stream can be replaced with a fixed image generated by a special test module in the pipeline. The module provides a selection of test patterns sufficient for basic testing of the pipeline.

Test patterns are accessible by programming a register and are shown in Figure 12. ON Semiconductor recommends disabling the MCU before enabling test patterns.

**Figure 12: Color Bar Test Pattern**

| Test Pattern     | Example  |
|------------------|--|
| Flat Field       |    |
| Vertical Ramp    |   |
| Color Bar        |  |
| Vertical Stripes |  |
| Pseudo-Random    |  |

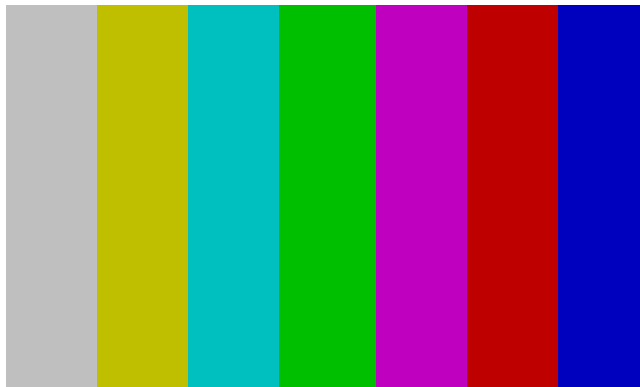


### NTSC/PAL Test Pattern Generation

There is a built-in standard EIA (NTSC) and EBU (PAL) color bars to support hue and color saturation characterization. Each pattern consists of seven color bars (white, yellow, cyan, green, magenta, red, and blue). The Y, Cb and Cr values for each bar are detailed in Tables 6 and 7.

The test pattern is invoked through a Host Command call to the TX Manager. See the MT9V136 Host Command Specification.

**Figure 13: Color Bars**



**Table 6: EIA Color Bars (NTSC)**

|    | Nominal Range | White | Yellow | Cyan | Green | Magenta | Red | Blue |
|----|---------------|-------|--------|------|-------|---------|-----|------|
| Y  | 16 to 235     | 180   | 162    | 131  | 112   | 84      | 65  | 35   |
| Cb | 16 to 240     | 128   | 44     | 156  | 72    | 184     | 100 | 212  |
| Cr | 16 to 240     | 128   | 142    | 44   | 58    | 198     | 212 | 114  |

**Table 7: EBU Color Bars (PAL)**

|    | Nominal Range | White | Yellow | Cyan | Green | Magenta | Red | Blue |
|----|---------------|-------|--------|------|-------|---------|-----|------|
| Y  | 16 to 235     | 235   | 162    | 131  | 112   | 84      | 65  | 35   |
| Cb | 16 to 240     | 128   | 44     | 156  | 72    | 184     | 100 | 212  |
| Cr | 16 to 240     | 128   | 142    | 44   | 58    | 198     | 212 | 114  |

### CCIR-656 Format

The color bar data is encoded in 656 data streams. The duration of the blanking and active video periods of the generated 656 data are summarized in the following tables.

**Table 8: NTSC**

| Line Numbers | Field | Description  |
|--------------|-------|--------------|
| 1-3          | 2     | Blanking     |
| 4-19         | 1     | Blanking     |
| 20-263       | 1     | Active video |
| 264-265      | 1     | Blanking     |
| 266-282      | 2     | Blanking     |
| 283-525      | 2     | Active Video |

**Table 9: PAL**

| Line Numbers | Field | Description  |
|--------------|-------|--------------|
| 1-22         | 1     | Blanking     |
| 23-310       | 1     | Active video |
| 311-312      | 1     | Blanking     |
| 313-335      | 2     | Blanking     |
| 336-623      | 2     | Active video |
| 624-625      | 2     | Blanking     |

### Black Level Subtraction and Digital Gain

Image stream processing starts with black level subtraction and multiplication of all pixel values by a programmable digital gain. Both operations can be independently set to separate values for each color channel (R, Gr, Gb, B). Independent color channel digital gain can be adjusted with registers. Independent color channel black level adjustments can also be made. If the black level subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0.

### Positional Gain Adjustments (PGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The MT9V136 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

### The Correction Function

The correction functions can then be applied to each pixel value to equalize the response across the image as follows:

$$P_{corrected}(row,col) = P_{sensor}(row,col) * f(row,col) \tag{EQ 1}$$

where  $P$  are the pixel values and  $f$  is the color dependent correction functions for each color channel.

### Color Interpolation

In the raw data stream fed by the sensor core to the IFP, each pixel is represented by a 10-bit integer number, which can be considered proportional to the pixel's response to a one-color light stimulus, red, green, or blue, depending on the pixel's position under the color filter array. Initial data processing steps, up to and including the defect correction, preserve the one-color-per-pixel nature of the data stream, but after the defect correction it must be converted to a three-colors-per-pixel stream appropriate for standard color processing. The conversion is done by an edge-sensitive color interpolation module. The module pads the incomplete color information available for each pixel with information extracted from an appropriate set of neighboring pixels. The algorithm used to select this set and extract the information seeks the best compromise between preserving edges and filtering out high frequency noise in flat field areas. The edge threshold can be set through register settings.

### Color Correction and Aperture Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3 x 3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. Since such sums can have up to 12 significant bits, the bit width of the image data stream is widened to 12 bits per color (36 bits per pixel). The color correction matrix can be either programmed by the user or automatically selected by the auto white balance (AWB) algorithm implemented in the IFP. Color correction should ideally produce output colors that are corrected for the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction variables can be adjusted through register settings.

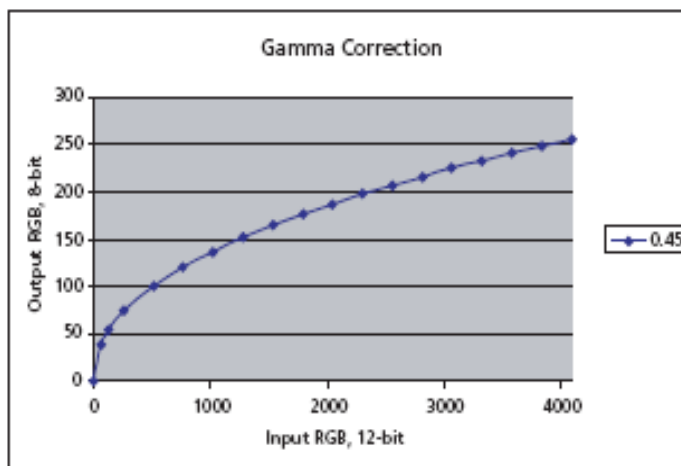
To increase image sharpness, a programmable 2D aperture correction (sharpening filter) is applied to color-corrected image data. The gain and threshold for 2D correction can be defined through register settings.

## Gamma Correction

The MT9V136 IFP includes a block for gamma correction that can adjust its shape based on brightness to enhance the performance under certain lighting conditions. Two custom gamma correction tables may be uploaded corresponding to a brighter lighting condition and a darker lighting condition. At power-up, the IFP loads the two tables with default values. The final gamma correction table used depends on the brightness of the scene and takes the form of an interpolated version of the two tables.

The gamma correction curve (as shown in Figure 14) is implemented as a piecewise linear function with 19 knee points, taking 12-bit arguments and mapping them to 8-bit output. The abscissas of the knee points are fixed at 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. The 8-bit ordinates are programmable through IFP registers.

**Figure 14: Gamma Correction Curve**



## RGB to YUV Conversion

For further processing, the data is converted from RGB color space to YUV color space.

## Color Kill

To remove high- or low-light color artifacts, a color kill circuit is included. It affects only pixels whose luminance exceeds a certain preprogrammed threshold. The U and V values of those pixels are attenuated proportionally to the difference between their luminance and the threshold.

## YUV Color Filter

As an optional processing step, noise suppression by one-dimensional low-pass filtering of Y and/or UV signals is possible. A 3- or 5-tap filter can be selected for each signal.

## YUV-to-RGB/YUV Conversion and Output Formatting

The YUV data stream emerging from the scaling module can either exit the color pipeline as-is or be converted before exit to an alternative YUV or RGB data format.

### Output Format and Timing

#### YUV/RGB Data Ordering

The MT9V136 supports swapping YCbCr mode, as illustrated in Table 10.

**Table 10: YCbCr Output Data Ordering**

| Mode              | Data Sequence   |                 |                  |                  |
|-------------------|-----------------|-----------------|------------------|------------------|
| Default (no swap) | Cb <sub>i</sub> | Y <sub>i</sub>  | Cr <sub>i</sub>  | Y <sub>i+1</sub> |
| Swapped CbCr      | Cr <sub>i</sub> | Y <sub>i</sub>  | Cb <sub>i</sub>  | Y <sub>i+1</sub> |
| Swapped YC        | Y <sub>i</sub>  | Cb <sub>i</sub> | Y <sub>i+1</sub> | Cr <sub>i</sub>  |
| Swapped CbCr, YC  | Y <sub>i</sub>  | Cr <sub>i</sub> | Y <sub>i+1</sub> | Cb <sub>i</sub>  |

The RGB output data ordering in default mode is shown in Table 11. The odd and even bytes are swapped when luma/chroma swap is enabled. R and B channels are bit-wise swapped when chroma swap is enabled.

**Table 11: RGB Ordering in Default Mode**

| Mode (Swap Disabled) | Byte | D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub> |
|----------------------|------|---|
| 565RGB               | Odd  | R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> R <sub>3</sub> G <sub>7</sub> G <sub>6</sub> G <sub>5</sub> |
|                      | Even | G <sub>4</sub> G <sub>3</sub> G <sub>2</sub> B <sub>7</sub> B <sub>6</sub> B <sub>5</sub> B <sub>4</sub> B <sub>3</sub> |
| 555RGB               | Odd  | 0 R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> R <sub>3</sub> G <sub>7</sub> G <sub>6</sub>              |
|                      | Even | G <sub>5</sub> G <sub>4</sub> G <sub>3</sub> B <sub>7</sub> B <sub>6</sub> B <sub>5</sub> B <sub>4</sub> B <sub>3</sub> |
| 444xRGB              | Odd  | R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> G <sub>7</sub> G <sub>6</sub> G <sub>5</sub> G <sub>4</sub> |
|                      | Even | B <sub>7</sub> B <sub>6</sub> B <sub>5</sub> B <sub>4</sub> 0 0 0 0   |
| x444RGB              | Odd  | 0 0 0 0 R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub>   |
|                      | Even | G <sub>7</sub> G <sub>6</sub> G <sub>5</sub> G <sub>4</sub> B <sub>7</sub> B <sub>6</sub> B <sub>5</sub> B <sub>4</sub> |

### Uncompressed 10-Bit Bypass Output

Raw 10-bit Bayer data from the sensor core can be output in bypass mode in two ways:

- Using 8 data output signals (DOUT[7:0]) and GPIO[1:0]. The GPIO signals are the least significant 2 bits of data.
- Using only 8 signals (DOUT[7:0]) and a special 8 + 2 data format, shown in Table 12.

**Table 12: 2-Byte Bayer Format**

| Byte       | Bits Used                   | Bit Sequence  |
|------------|-----------------------------|---|
| Odd bytes  | 8 data bits                 | D <sub>9</sub> D <sub>8</sub> D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> |
| Even bytes | 2 data bits + 6 unused bits | 0 0 0 0 0 0 D <sub>1</sub> D <sub>0</sub>   |

### Readout Formats

Progressive format is used for raw Bayer output.

## Output Formats

### ITU-R BT.656 and RGB Output

The MT9V136 can output processed video as a standard ITU-R BT.656 (CCIR656) stream, an RGB stream, or as unprocessed Bayer data. The ITU-R BT.656 stream contains YCbCr 4:2:2 data with fixed embedded synchronization codes. This output is typically suitable for subsequent display by standard video equipment or JPEG/MPEG compression.

Colorpipe data (pre-lens correction and overlay) can also be output in YCbCr 4:2:2 and a variety of RGB formats in 640 by 480 progressive format in conjunction with LINE\_VALID and FRAME\_VALID.

The MT9V136 can be configured to output 16-bit RGB (565RGB), 15-bit RGB (555RGB), and two types of 12-bit RGB (444RGB). Refer to Table 27 and Table 28 on page 59 for details.

### Bayer Output

Unprocessed Bayer data are generated when bypassing the IFP completely—that is, by simply outputting the sensor Bayer stream as usual, using FRAME\_VALID, LINE\_VALID, and PIXCLK to time the data. This mode is called sensor stand-alone mode.

## Output Ports

### Composite Video Output

The composite video output DAC is external-resistor-programmable and supports both single-ended and differential output. The DAC is driven by the on-chip video encoder output.

### Parallel Output

Parallel output uses either 8-bit or 10-bit output. Eight-bit output is used for ITU-R BT.656 and RGB output. Ten-bit output is used for raw Bayer output.

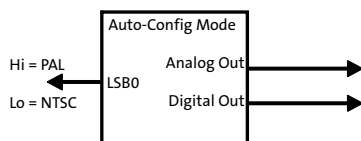
## Usage Modes

How a camera based on the MT9V136 will be configured depends on what features are used. In the simplest case, only an MT9V136 plus an external flash memory, or an 8-bit microcontroller ( $\mu$ C) might be sufficient. Flash sizes vary depending on the data for registers, firmware, and overlay data—somewhere between 10Kb to 16MB. The two-wire bus is adequate since only high-level commands are used to invoke overlays, load registers from memory, or set up lens correction parameters. Overlay data can alternatively be issued by the external  $\mu$ C if the rate of refreshing data is deemed adequate. If there are no commands in the Flash image the device can be in auto configuration mode by which the sensor is set up according to the status of pins FRAME\_VALID, LINE\_VALID and DOUT\_LSB0. For further information, see “Auto-Configuration” on page 34.

In the simplest case no Flash memory or  $\mu$ C is required, as shown in Figure 15. This is truly a single chip operation.

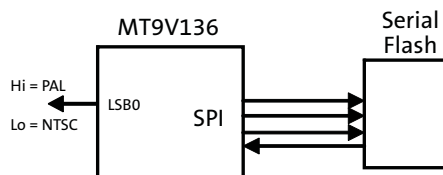
**Note:** Because mandatory patches must be loaded, the Auto-Config mode is not recommended.

**Figure 15: Auto-Config Mode**



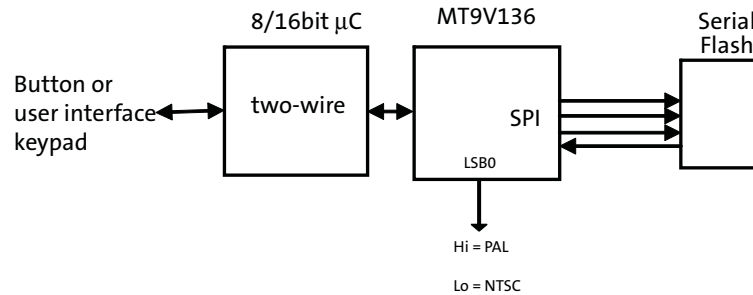
The MT9V136 can be configured by a serial Flash through the SPI Interface.

**Figure 16: Flash Mode**



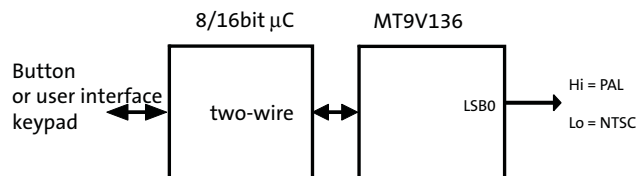
In some applications, button or user interface keypad can trigger overlay images being called by the  $\mu\text{C}$  as shown in Figure 17.

**Figure 17: Host Mode with Flash**



Overlay information may also be passed by the  $\mu\text{C}$  without a need for a Flash memory. However, because the data transfer rate is limited over the two-wire serial bus, the update rate may be slower. However, if overlay images are preloaded into the four on-chip buffers, they may be turned on and off or move location at the frame rate as shown in Figure 18.

**Figure 18: Host Mode**

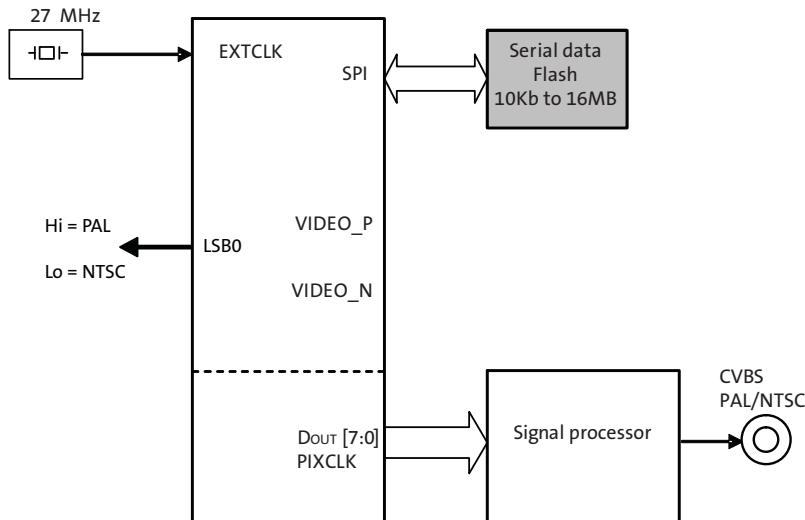




## External Signal Processing

An external signal processor can take data from ITU656 or raw Bayer output format and post-process or compress the data in various formats.

**Figure 19: External Signal Processing Block Diagram**



## Device Configuration

After power is applied and the device is out of reset by de-asserting the RESET\_BAR pin, it will enter a boot sequence to configure its operating mode. There are essentially four modes, two when Flash is present and two when Flash is not present. Figure 20: “Power-Up Sequence – Configuration Options Flow Chart,” on page 35 contains more details on the configuration options.

If Flash is present and:

- A valid Flash device identifier is detected AND the Flash device contains valid configuration records, then
  - Disable Auto-Config
  - Parse Flash Content
  - Load Flash Configuration ->Flash Configuration Mode
- A valid Flash device identifier is detected BUT the Flash device DOES NOT contain valid configuration records, then
  - Enter Auto Configuration.

If Flash is not present and:

- SPI\_SDI == 0, then
  - Enter Host Configuration.
- SPI\_SDI != 0, then
  - Enter Auto Configuration

## Auto-Configuration

The device supports an auto-configuration feature. During system start-up, the device first detects whether an SPI Flash device is attached to the MT9V136. If not, it will then sample the state of a number of GPI inputs including FRAME\_VALID, LINE\_VALID and DOUT\_LSB0. For more information, see Table 15, “GPIO Bit Descriptions,” on page 36. The state of these inputs then determines the configuration of a number of subsystems of the device such as readout mode, pedestal and video format, respectively.

The auto-configuration feature can be disabled by grounding the SPI\_DIN pin. The device samples the state of this pin during the Flash device detection process. If no SPI Flash device is detected (read device ID of 0x00 or 0xFF), OR the SPI\_DIN pin is grounded, then auto-configuration is disabled.

## Flash Configuration Mode

If a valid Flash is detected (by reading device ID other than 0x00 or 0xFF) and the flash device contains valid configuration records, then these configuration records are processed.

## Host Configuration

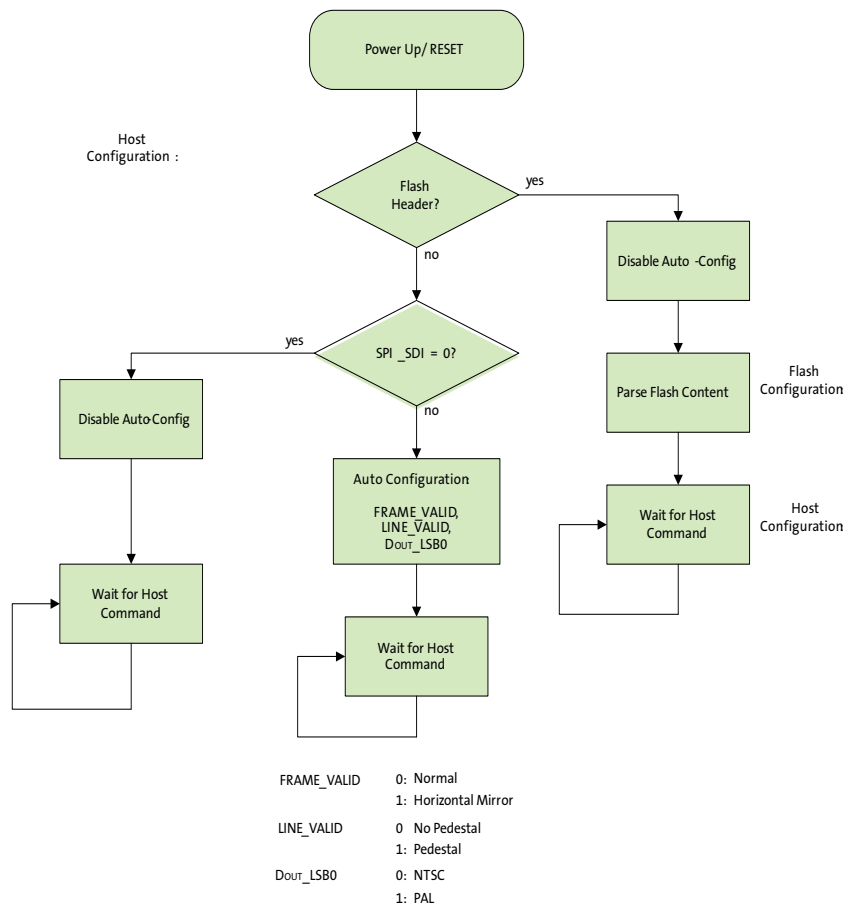
This mode is entered if the SPI\_DIN pin is grounded. The SOC performs no configuration, and remains idle waiting for configuration and instruction from the host.

## Power Sequence

In power-up, the core voltage (1.8V) must trail the IO (2.8V) by a positive number. All 2.8V rails can be turned on at the same time or follow the power-up sequence in Figure 46: “Power Up Sequence,” on page 65.

In power down, the sequence is reversed. The core voltage (1.8V) must be turned off before any 2.8V. Refer to Figure 47: “Power Down Sequence,” on page 66 for details.

**Figure 20: Power-Up Sequence – Configuration Options Flow Chart**



## Supported SPI Devices

Table 13 lists supported Flash devices. Devices not compatible will require a firmware patch. Contact ON Semiconductor for additional support.

**Table 13: SPI Flash Devices**

| Type  | Density | Manufacturer | Device       | Speed (MHz) | Standard        | Temp Range (°F) | Supported |
|-------|---------|--------------|--------------|-------------|-----------------|-----------------|-----------|
| Flash | 8 MB    | Atmel        | AT26DF081A   | 70          | JEDEC/Device ID | -20 to +85      | Yes       |
| Flash | 1 MB    | ST           | M25P10-AVMB3 | 50          |                 | -40 to +125     | Yes       |

## Supported SPI Commands

The SPI commands shown in Table 14 are supported by the MT9V136.

**Table 14: SPI Commands Supported**

| Command                         | Value |
|---------------------------------|-------|
| Read Array                      | 0x03  |
| Block Erase                     | 0xD8  |
| Chip Erase                      | 0xC7  |
| Read Status                     | 0x05  |
| Write status                    | 0x01  |
| Byte Page Program               | 0x02  |
| Write Enable                    | 0x06  |
| Write Disable                   | 0x04  |
| Read Manufacturer and Device ID | 0x9F  |
| (Fast) Read Array               | 0x0B  |

**Table 15: GPIO Bit Descriptions**

|            | GPI[2]<br>(DOUT_LSB0) | GPI[1]<br>(FRAME_VALID) | GPI[0]<br>(LINE_VALID) |
|------------|-----------------------|-------------------------|------------------------|
| Low ("0")  | NTSC                  | Normal                  | No pedestal            |
| High ("1") | PAL                   | Horizontal mirror       | Pedestal               |

### Host Command Interface

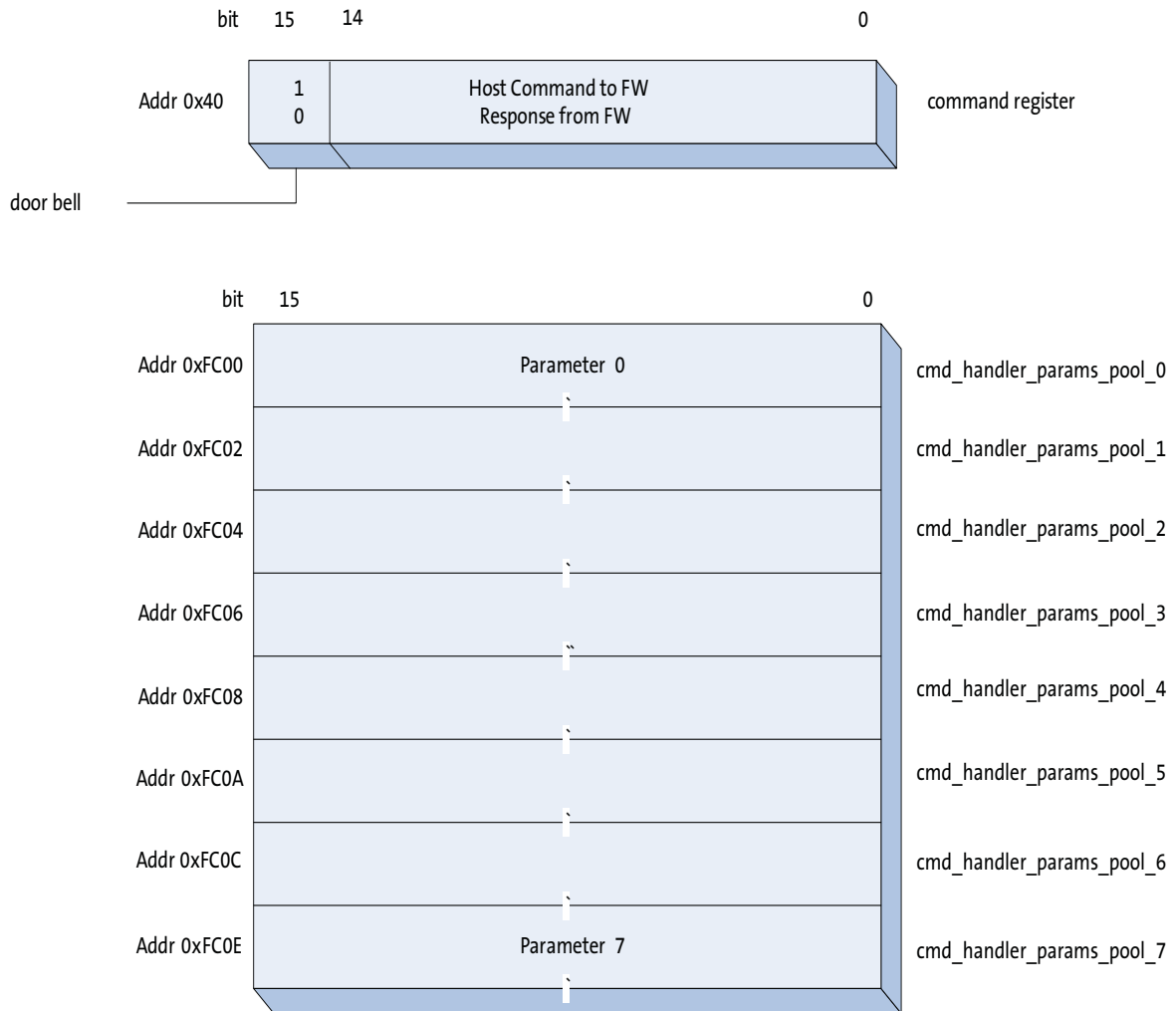
ON Semiconductor’s sensors and SOCs contain numerous registers that are accessed through a two-wire interface with speeds up to 400 kHz.

The MT9V136, in addition to writing or reading straight to/from registers or firmware variables, has a mechanism to write higher level commands, the Host Command Interface (HCI). Once a command has been written through the HCI, it will be executed by on chip firmware and the results are reported back. In general, registers shall not be accessed with the exception of registers that are marked for “User Access.”

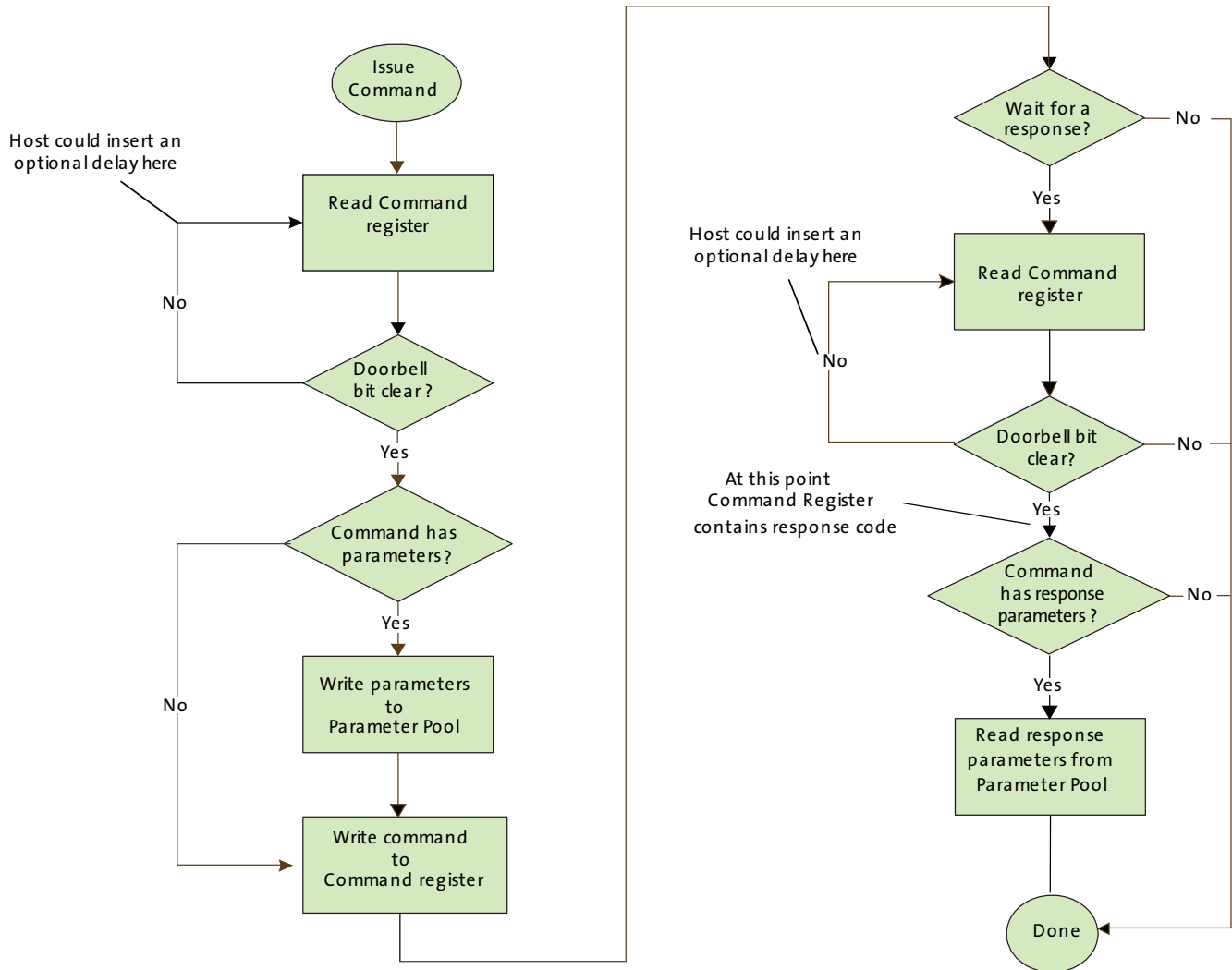
Flash memory is also available to store commands for later execution. Under DMA control, a command is written into the SOC and executed.

For a complete spec on host commands, refer to the MT9V136 Host Command Interface Specification.

**Figure 21: Interface Structure**



### Host Command Process Flow



### Command Flow

The host issues a command by writing (through a two-wire interface bus) to the command register. All commands are encoded with bit 15 set, which automatically generates the host command (doorbell) interrupt to the microprocessor.

Assuming initial conditions, the host first writes the command parameters (if any) to the parameters pool (in the command handler's logical page), then writes the command to command register. The interrupt handler then signals the command handler task to process the command.

If the host wishes to determine the outcome of the command, it must poll the command register waiting for the doorbell bit to be cleared. This indicates that the firmware completed processing the command. The contents of the command register indicate the command's result status. If the command generated response parameters, the host can now retrieve these from the parameters pool.

**Note:** The host must not write to the parameters pool, nor issue another command, until the previous command completes. This is true even if the host does not care about the result of the previous command. Therefore, the host must always poll the command register to determine the state of the doorbell bit, and ensure the bit is cleared before issuing a command.

For a complete command list and further information consult the Host Command Interface Specification.

An example of how (using DevWare) a command may be initiated in the form of a “Preset” follows.

### Set Parallel Mode - Normal (Overlay i656)

All DevWare presets supplied by ON Semiconductor poll and test the doorbell bit after issuing the command. Therefore there is no need to check if the doorbell bit is clear before issuing the next command.

```
REG= 0xFC00, 0x1000 // CMD_HANDLER_PARAMS_POOL_0
REG= 0x0040, 0x8801 // issue command
// POLL COMMAND_REGISTER::DOORBELL => 0x0
```

### Summary of Host Commands

Table 16 on page 39 through Table 21 on page 41 show summaries of the host commands. The commands are divided into the following sections:

- System Manager
- Overlay
- GPIO Host interface
- Flash Manager Host
- Patch Loader Interface
- TX Manager

Following is a summary of the Host Interface commands. The description gives a quick orientation. The “Type” column shows if it is an asynchronous or synchronous command. For a complete list of all commands including parameters, consult the Host Command Interface Specification document.

**Table 16: System Manager Commands**

| System Manager Host Command | Value  | Type         | Description                          |
|-----------------------------|--------|--------------|--------------------------------------|
| Set State                   | 0x8100 | Asynchronous | Request the system enter a new state |
| Get State                   | 0x8101 | Synchronous  | Get the current state of the system  |

**Table 17: Overlay Host Commands**

| Overlay Host Command | Value  | Type        | Description                                 |
|----------------------|--------|-------------|---|
| Enable Overlay       | 0x8200 | Synchronous | Enable or disable the overlay subsystem     |
| Get Overlay State    | 0x8201 | Synchronous | Retrieve the state of the overlay subsystem |
| Set Calibration      | 0x8202 | Synchronous | Set the calibration offset                  |
| Set Bitmap Property  | 0x8203 | Synchronous | Set a property of a bitmap                  |
| Get Bitmap Property  | 0x8204 | Synchronous | Get a property of a bitmap                  |
| Set String Property  | 0x8205 | Synchronous | Set a property of a character string        |

**Table 17: Overlay Host Commands**

| Overlay Host Command | Value  | Type         | Description  |
|----------------------|--------|--------------|--|
| Load Buffer          | 0x8206 | Asynchronous | Load an overlay buffer with a bitmap (from Flash)  |
| Load Status          | 0x8207 | Synchronous  | Retrieve status of an active load buffer operation |
| Write Buffer         | 0x8208 | Synchronous  | Write directly to an overlay buffer                |
| Read Buffer          | 0x8209 | Synchronous  | Read directly from an overlay buffer               |
| Enable Layer         | 0x820A | Synchronous  | Enable or disable an overlay layer                 |
| Get Layer Status     | 0x820B | Synchronous  | Retrieve the status of an overlay layer            |
| Set String           | 0x820C | Synchronous  | Set the character string                           |
| Load String          | 0x820E | Asynchronous | Load a character string (from Flash)               |

**Table 18: GPIO Host Commands**

| GPIO Host Command   | Value  | Type        | Description   |
|---------------------|--------|-------------|---|
| Set GPIO Property   | 0x8400 | Synchronous | Set a property of one or more GPIO pins                               |
| Get GPIO Property   | 0x8401 | Synchronous | Retrieve a property of a GPIO pin                                     |
| Set GPO State       | 0x8402 | Synchronous | Set the state of a GPO pin or pins                                    |
| Get GPIO State      | 0x8403 | Synchronous | Get the state of a GPI pin or pins                                    |
| Set GPI Association | 0x8404 | Synchronous | Associate a GPI pin state with a Command Sequence stored in SPI Flash |

**Table 19: Flash Manager Host Commands**

| Flash Manager Host Command | Value  | Type         | Description  |
|----------------------------|--------|--------------|--|
| Get Lock                   | 0x8500 | Asynchronous | Request the Flash Manager access lock                          |
| Lock Status                | 0x8501 | Synchronous  | Retrieve the status of the access lock request                 |
| Release Lock               | 0x8502 | Synchronous  | Release the Flash Manager access lock                          |
| Config                     | 0x8503 | Synchronous  | Configure the Flash Manager and underlying SPI Flash subsystem |
| Read                       | 0x8504 | Asynchronous | Read data from the SPI Flash                                   |
| Write                      | 0x8505 | Asynchronous | Write data to the SPI Flash                                    |
| Erase Block                | 0x8506 | Asynchronous | Erase a block of data from the SPI Flash                       |
| Erase Device               | 0x8507 | Asynchronous | Erase the SPI Flash device                                     |
| Query Device               | 0x8508 | Asynchronous | Query device-specific information                              |
| Status                     | 0x8509 | Synchronous  | Obtain status of current asynchronous operation                |

**Table 20: Sequencer Host Commands**

| Sequencer Host Command | Value  | Type        | Description                        |
|------------------------|--------|-------------|------------------------------------|
| Set Encoding Mode      | 0x8603 | Synchronous | Set the encoding mode              |
| Enable Horizontal Flip | 0x8604 | Synchronous | Enable or disable horizontal flip  |
| Set Flicker Frequency  | 0x8605 | Synchronous | Set the flicker frequency          |
| Refresh Mode           | 0x8606 | Synchronous | Refresh the Sequencer mode/context |



**Table 21: TX Manager Host Commands**

| TX Manager Host Command | Value  | Type        | Description                        |
|-------------------------|--------|-------------|------------------------------------|
| Config DAC              | 0x8800 | Synchronous | Configure the Video DAC            |
| Set Parallel Mode       | 0x8801 | Synchronous | Configure the Parallel output port |

## Slave Two-Wire Serial Interface

The two-wire serial interface bus enables read/write access to control and status registers within the MT9V136. This interface is designed to be compatible with the MIPI Alliance Standard for Camera Serial Interface 2 (CSI-2) 1.0, which uses the electrical characteristics and transfer protocols of the two-wire serial interface specification.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and used to synchronize transfers.

Data is transferred between the master and the slave on a bidirectional signal (SDATA). SDATA is pulled up to VDD\_IO off-chip by a pull-up resistor in the range of 1.5 to 4.7kΩ resistor.

### Protocol

Data transfers on the two-wire serial interface bus are performed by a sequence of low-level protocol elements, as follows:

- a start or restart condition
- a slave address/data direction byte
- a 16-bit register address
- an acknowledge or a no-acknowledge bit
- data bytes
- a stop condition

The bus is idle when both SCLK and SDATA are HIGH. Control of the bus is initiated with a start condition, and the bus is released with a stop condition. Only the master can generate the start and stop conditions.

The SADDR pin is used to select between two different addresses in case of conflict with another device. If SADDR is LOW, the slave address is 0x90; if SADDR is HIGH, the slave address is 0xBA. See Table 22 below.

**Table 22: Two-Wire Interface ID Address Switching**

| SADDR | Two-Wire Interface Address ID |
|-------|-------------------------------|
| 0     | 0x90                          |
| 1     | 0xBA                          |

### Start Condition

A start condition is defined as a HIGH-to-LOW transition on SDATA while SCLK is HIGH. At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a “repeated start” or “restart” condition.

### Data Transfer

Data is transferred serially, 8 bits at a time, with the MSB transmitted first. Each byte of data is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes.

One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is low and must be stable while SCLK is HIGH.

### Slave Address/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A “0” in bit [0] indicates a write, and a “1” indicates a read. The default slave addresses used by the MT9V136 are 0x90 (write address) and 0x91 (read address). Alternate slave addresses of 0xBA (write address) and 0xBB (read address) can be selected by asserting the SADDR input signal.

### Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data. The protocol used is outside the scope of the two-wire serial interface specification.

### Acknowledge Bit

Each 8-bit data transfer is followed by an acknowledge bit or a no-acknowledge bit in the SCLK clock period following the data transfer. The transmitter (which is the master when writing, or the slave when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As for data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

### No-Acknowledge Bit

The no-acknowledge bit is generated when the receiver does not drive SDATA low during the SCLK clock period following a data transfer. A no-acknowledge bit is used to terminate a read sequence.

### Stop Condition

A stop condition is defined as a LOW-to-HIGH transition on SDATA while SCLK is HIGH.

## Typical Operation

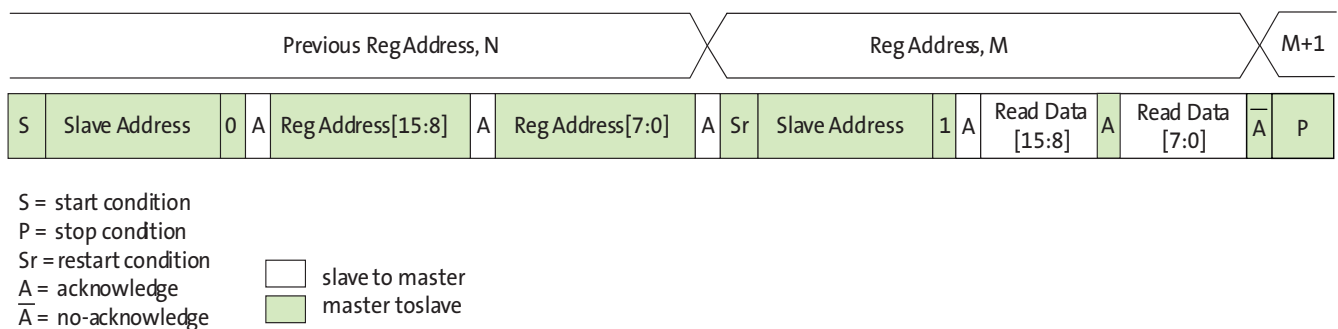
A typical READ or WRITE sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a READ or a WRITE, where a “0” indicates a WRITE and a “1” indicates a READ. If the address matches the address of the slave device, the slave device acknowledges receipt of the address by generating an acknowledge bit on the bus.

If the request was a WRITE, the master then transfers the 16-bit register address to which a WRITE will take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master will then transfer the 16-bit data, as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master stops writing by generating a (re)start or stop condition. If the request was a READ, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, just as in the write request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, 8 bits at a time. The master generates an acknowledge bit after each 8-bit transfer. The data transfer is stopped when the master sends a no-acknowledge bit.

### Single READ from Random Location

Figure 22 shows the typical READ cycle of the host to MT9V136. The first two bytes sent by the host are an internal 16-bit register address. The following 2-byte READ cycle sends the contents of the registers to host.

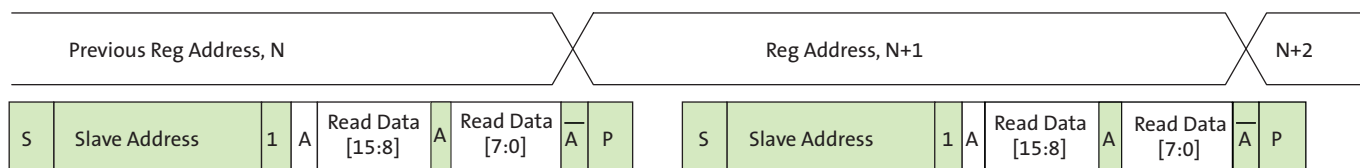
Figure 22: Single READ from Random Location



### Single READ from Current Location

Figure 23 shows the single READ cycle without writing the address. The internal address will use the previous address value written to the register.

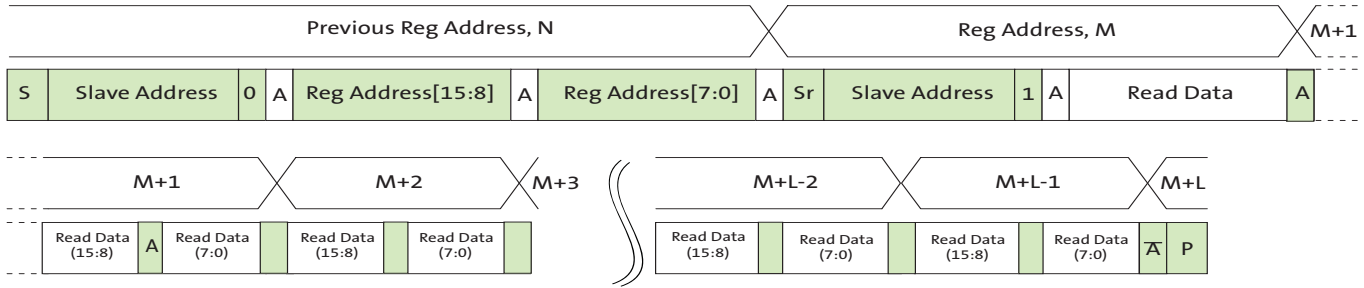
Figure 23: Single Read from Current Location



**Sequential READ, Start from Random Location**

This sequence (Figure 24) starts in the same way as the single READ from random location (Figure 22 on page 44). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until “L” bytes have been read.

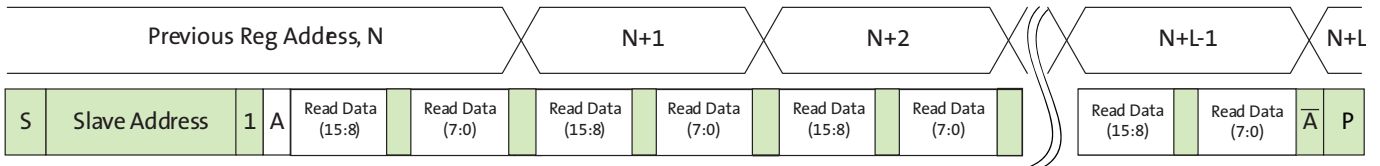
**Figure 24: Sequential READ, Start from Random Location**



**Sequential READ, Start from Current Location**

This sequence (Figure 25) starts in the same way as the single READ from current location (Figure 23). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte reads until “L” bytes have been read.

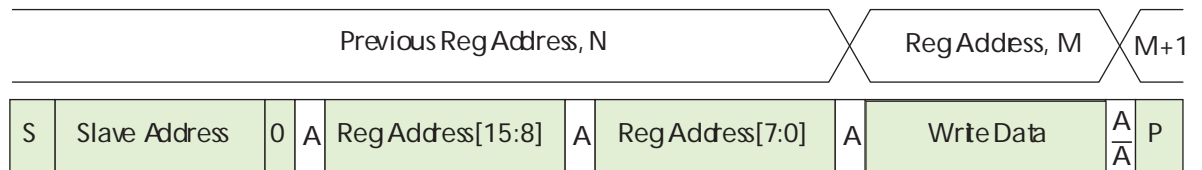
**Figure 25: Sequential READ, Start from Current Location**



**Single Write to Random Location**

Figure 26 shows the typical WRITE cycle from the host to the MT9V136. The first 2 bytes indicate a 16-bit address of the internal registers with most-significant byte first. The following 2 bytes indicate the 16-bit data.

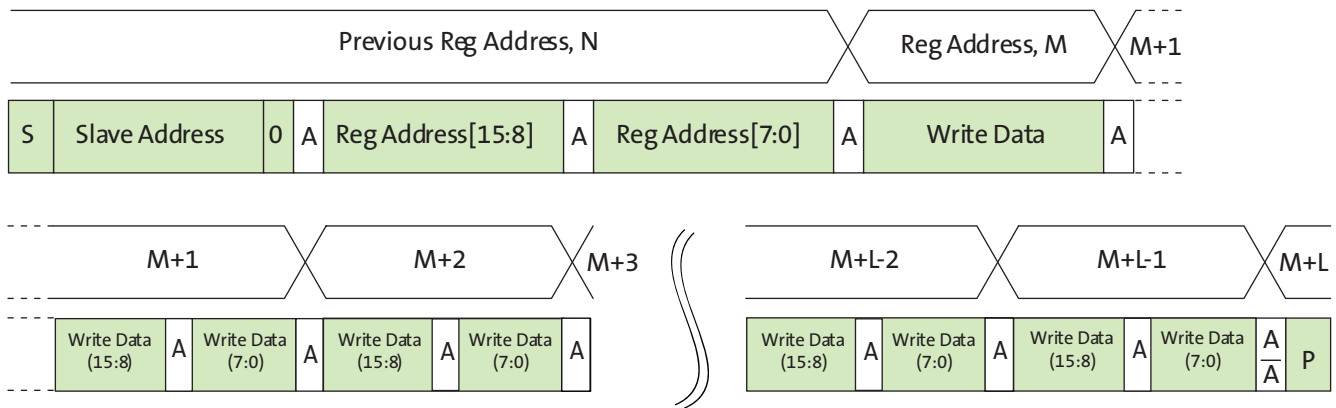
**Figure 26: Single WRITE to Random Location**



### Sequential WRITE, Start at Random Location

This sequence (Figure 27) starts in the same way as the single WRITE to random location (Figure 26). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte writes until “L” bytes have been written. The WRITE is terminated by the master generating a stop condition.

Figure 27: Sequential WRITE, Start at Random Location



## Overlay Capability

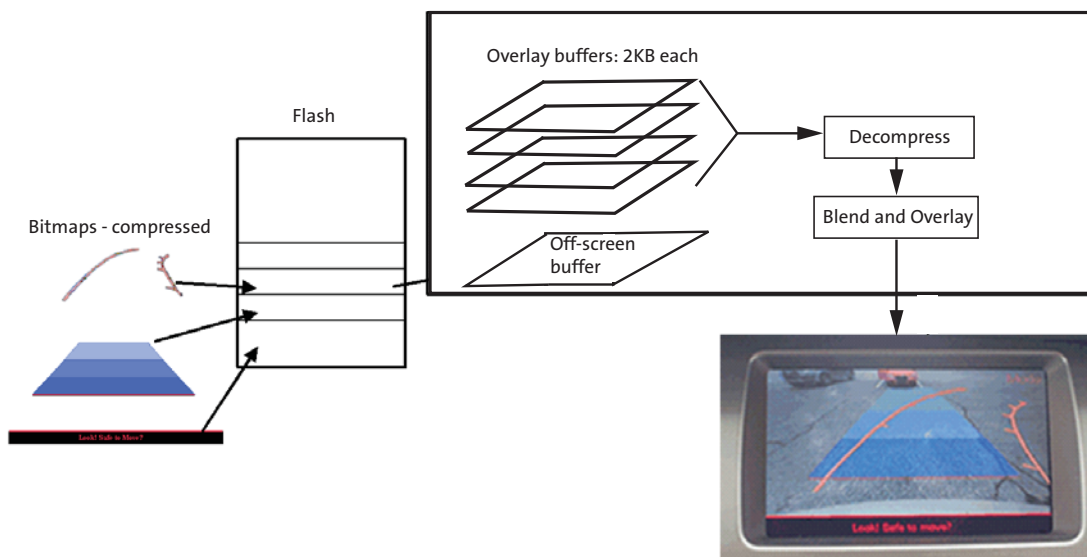
Figure 28 highlights the graphical overlay data flow of the MT9V136. The images are separated to fit into 2KB blocks of memory after compression.

- Up to four overlays may be blended simultaneously
- Overlay size 360 x 480 pixels rendered into a display area of 720 x 480 pixels
- Selectable readout: rotating order is user programmable
- Dynamic movement through predefined overlay images
- Palette of 32 colors out of 64,000 with eight colors per bitmap
- Blend factors may be changed dynamically to achieve smooth transitions

The host commands allow a bitmap to be written piecemeal to a memory buffer through the I<sup>2</sup>C, and through the DMA direct from SPI Flash memory. Multiple encoding passes may be required to fit an image into a 2KB block of memory; alternatively, the image can be divided into two or more blocks to make the image fit. Every graphic image may be positioned in an x/y direction and overlap with other graphic images.

The host may load an image at any time. Under control of DMA assist, data are transferred to the off-screen buffer in compressed form. This assures that no display data are corrupted during the replenishment of the four active overlay buffers.

Figure 28: Overlay Data Flow



Note: These images are not actually rendered, but show conceptual objects and object blending.

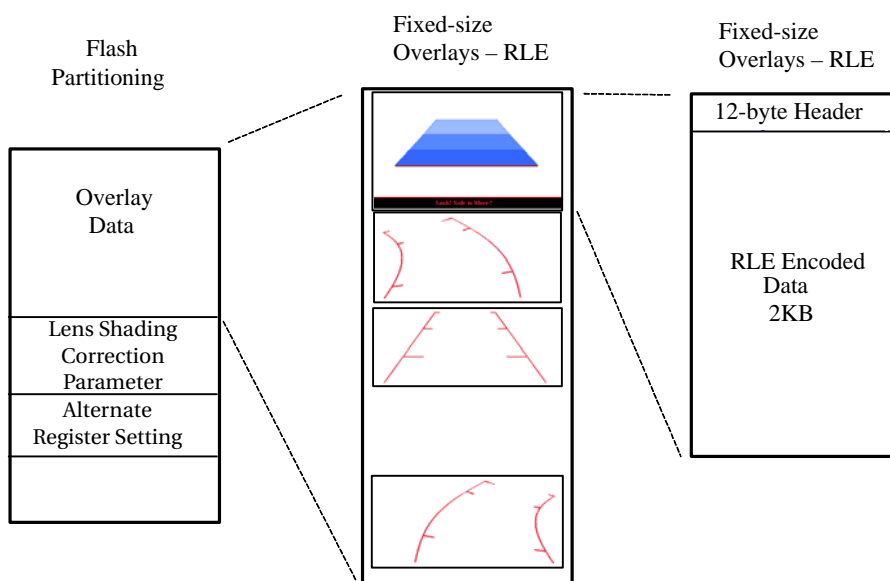
## Serial Memory Partition

The contents of the Flash/EEPROM memory partition logically into three blocks (see Figure 29):

- Memory for overlay data and descriptors
- Memory for register settings, which may be loaded at boot-up
- Firmware extensions or software patches; in addition to the on-chip firmware, extensions reside in this block of memory

These blocks are not necessarily contiguous.

**Figure 29: Memory Partitioning**



For a complete description of memory organization, refer to the MT9V136 SPI Flash Contents Encoding Specification.

## External Memory Speed Requirement

For a 2KB block of overlay to be transferred within a frame time to achieve maximum update rate, the serial memory has to be a certain speed.

**Table 23: Transfer Time Estimate**

| Frame Time | SPI Clock | Transfer Time to 2KB |
|------------|-----------|----------------------|
| 33.3ms     | 4.5 MHz   | 1ms                  |



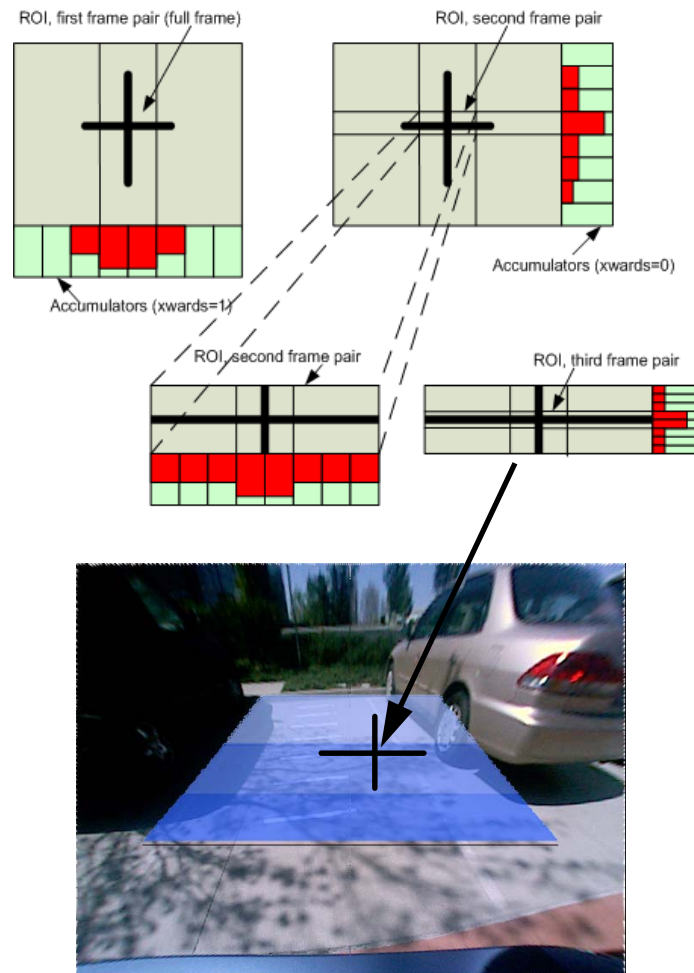
## Overlay Adjustment

To ensure a correct position of the overlay to compensate for assembly deviation, the overlay can be adjusted with assistance from the overlay statistics engine:

- The overlay statistics engine supports a windowed 8-bin luma histogram, either row-wise (vertical) or column-wise (horizontal).
- The example calibration statistics firmware patch can be used to perform an automatic successive-approximation search of a cross-hair target within the scene.
- On the first frame, the firmware performs a coarse horizontal search, followed by a coarse vertical search in the second frame.
- In subsequent frames, the firmware reduces the region-of-interest of the search to the histogram bins containing the greatest accumulator values, thereby refining the search.
- The resultant X, Y location of the cross-hair target can be used to assign a calibration value of offset selected overlay graphic image positions within the output image.
- The calibration statistics patch also supports a manual mode, which allows the host to access the raw accumulator values directly.

**Note:** For the overlay calibration feature to work, load the appropriate patch. See Statistics Engine document.

Figure 30: Overlay Calibration



The position of the target will be used to determine the calibration value that shifts the X,Y position of adjustable overlay graphics.

The overlay calibration is intended to be applied on a device by device basis “in system,” which means after the camera has been installed. ON Semiconductor provides basic programming scripts that may reside in the SPI Flash memory to assist in this effort.

## Overlay Character Generator

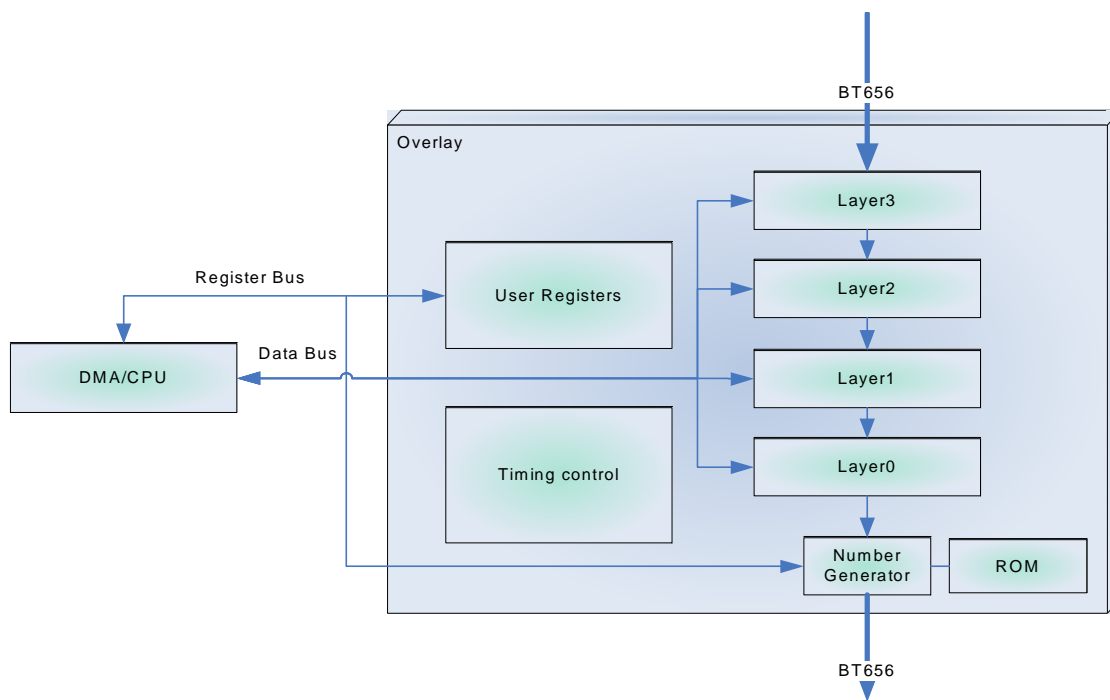
In addition to the four overlay layers, a fifth layer exists for a character generator overlay string.

There are a total of:

- 16 alphanumeric characters available
- 22 characters maximum per line
- 16 x 32 pixels with 1-bit color depth

Any update to the character generator string requires the string to be passed in its entirety with the Host Command. Character strings have their own control properties aside from the Overlay bitmap properties.

**Figure 31: Internal Block Diagram Overlay**



## Character Generator

The character generator can be seen as the fifth top layer, but instead of getting the source from RLE data in the memory buffers, it has a predefined 16 characters stored in ROM.

All the characters are 1-bit depth color and are sharing the same YCbCr look up table.

**Figure 32: Example of Character Descriptor 0 Stored in ROM**

| ROM  | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x02 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x04 | 0  | 0  | 0  | 0  | 0  | 0  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x06 | 0  | 0  | 0  | 0  | 0  | 1  | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x08 | 0  | 0  | 0  | 0  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0x0a | 0  | 0  | 0  | 1  | 1  | 1  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0x0c | 0  | 0  | 0  | 1  | 1  | 1  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0x0e | 0  | 0  | 1  | 1  | 1  | 1  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0x10 | 0  | 0  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0x12 | 0  | 0  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0x14 | 0  | 1  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x16 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x18 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x1a | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x1c | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x1e | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x20 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x22 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x24 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x26 | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x28 | 0  | 0  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x2a | 0  | 0  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x2c | 0  | 0  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x2e | 0  | 0  | 1  | 1  | 1  | 1  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0x30 | 0  | 0  | 0  | 1  | 1  | 1  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0x32 | 0  | 0  | 0  | 1  | 1  | 1  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0x34 | 0  | 0  | 0  | 0  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0x36 | 0  | 0  | 0  | 0  | 0  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0x38 | 0  | 0  | 0  | 0  | 0  | 0  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x3a | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x3c | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x3e | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ...  |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

It can show a row of up to 22 characters of 16 x 32 pixels resolution (32 x 32 pixels when blended with the BT 656 data).

## Character Generator Details

Table 24 shows the characters that can be generated.

**Table 24: Character Generator Details**

| Item             | Quantity | Description  |
|------------------|----------|--|
| 16-bit character | 22       | Coder for one of these characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, (space), :, -, (comma), (period) |
| 1 bpp color      | 1        | Depth of the bit map is 1 bpp  |

It is the responsibility of the user to set up proper values in the character positioning to fit them in the same row (that is one of the reasons that 22 is the maximum number of characters).

**Note:** No error is generated if the character row overruns the horizontal or vertical limits of the frame.

## Full Character Set for Overlay

Figure 33 shows all of the characters that can be generated by the MT9V136.

**Figure 33: Full Character Set for Overlay**

|     |     |     |     |   |   |   |   |
|-----|-----|-----|-----|---|---|---|---|
| 0x0 | 0x4 | 0x8 | 0xC | 0 | 4 | 8 |   |
| 0x1 | 0x5 | 0x9 | 0xD | 1 | 5 | 9 | , |
| 0x2 | 0x6 | 0xA | 0xE | 2 | 6 | : | - |
| 0x3 | 0x7 | 0xB | 0xF | 3 | 7 | / | . |

## Modes and Timing

This section provides an overview of the typical usage modes and related timing information for the MT9V136.

### Composite Video Output

The external pin DOUT\_LSB0 must be used to configure the device for default NTSC or PAL operation. This and other video configuration settings are available as register settings accessible through the serial interface.

#### NTSC

Both differential and single-ended connections of the full NTSC format are supported. The differential connection that uses two output lines is used for low noise or long distance applications. The single-ended connection is used for PCB tracks and screened cable where noise is not a concern. The NTSC format has three black lines at the bottom of each image for padding (which most LCDs do not display).

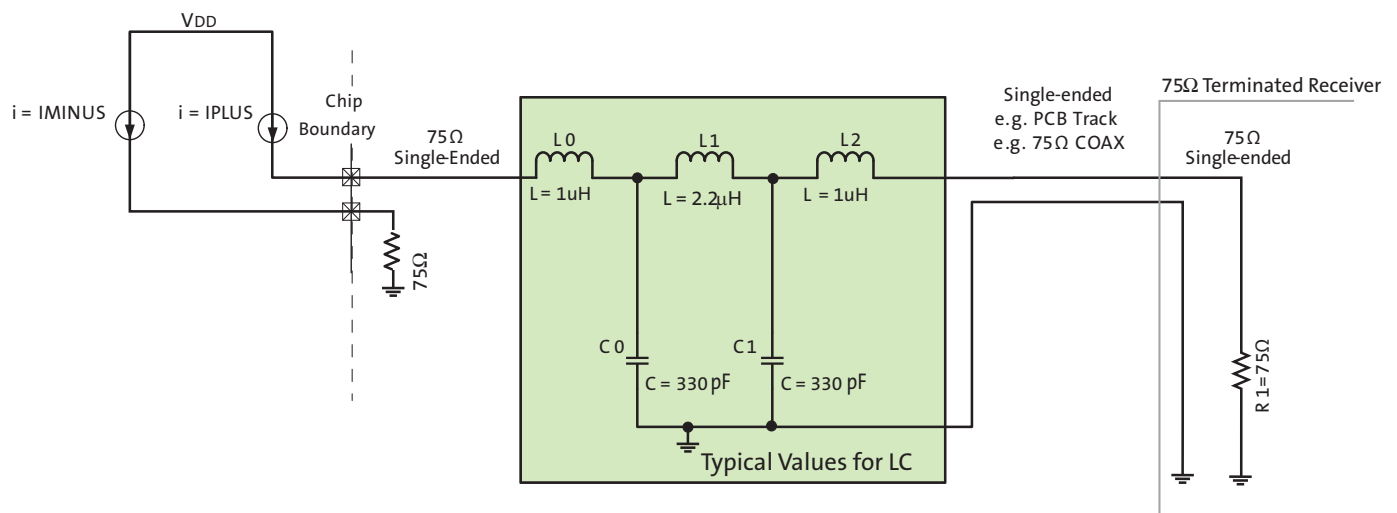
#### PAL

The PAL format is supported with 576 active image rows.

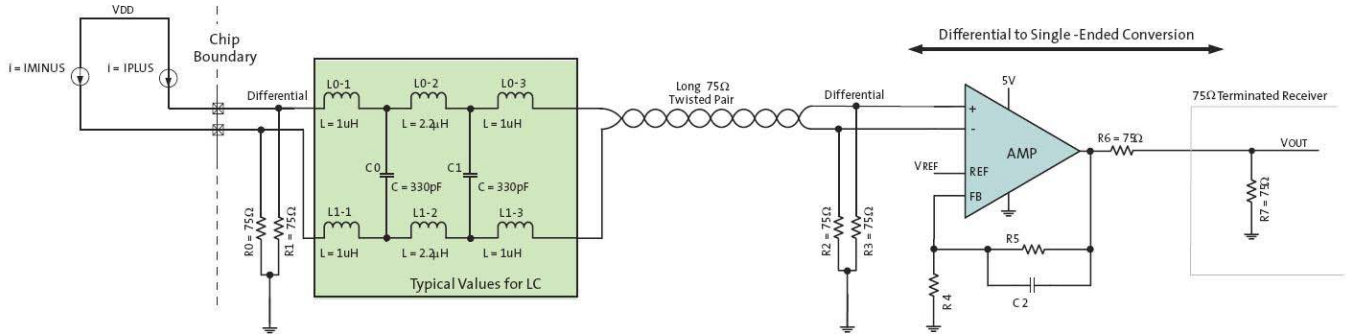
### Single-Ended and Differential Composite Output

The composite output can be operated in a single-ended or differential mode by simply changing the external resistor configuration. For single-ended termination, see Figure 34 on page 54. The differential schematic is shown in Figure 35 on page 55.

**Figure 34: Single-Ended Termination**



**Figure 35: Differential Connection—Grounded Termination**



**Parallel Output (Dout)**

The DOUT[7:0] port supports both progressive and Interlaced mode. Progressive mode (with FV and LV signal) include raw bayer(8 or 10 bit), YCbCr, RGB. Interlaced mode is CCIR656 compliant.

Figure 36 shows the data that is output on the parallel port for CCIR656. Both NTSC and PAL formats are displayed. The blue values in Figure 36 represent NTSC (525/60). The red values represent PAL (625/50).

**Figure 36: CCIR656 8-Bit Parallel Interface Format for 525/60 (625/50) Video Systems**

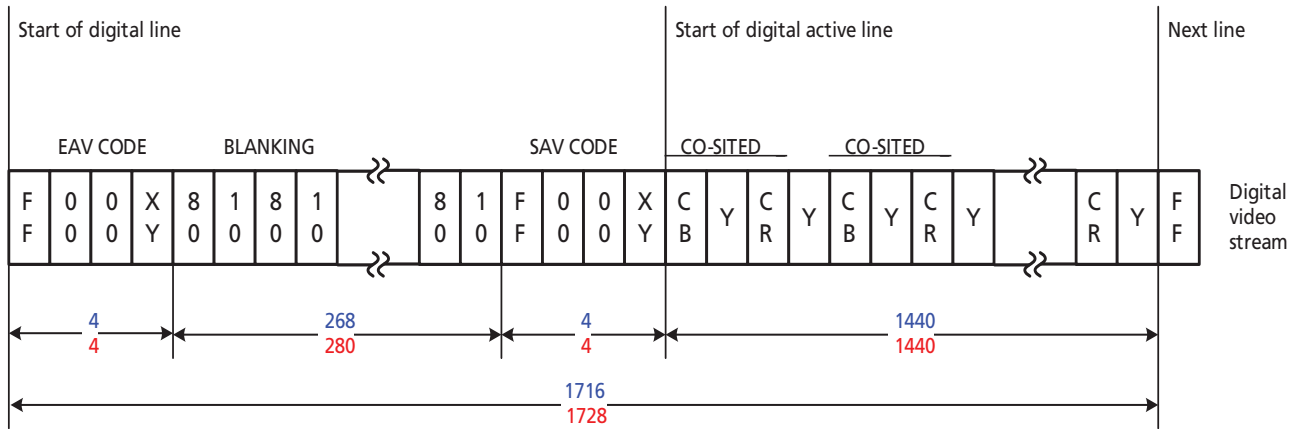
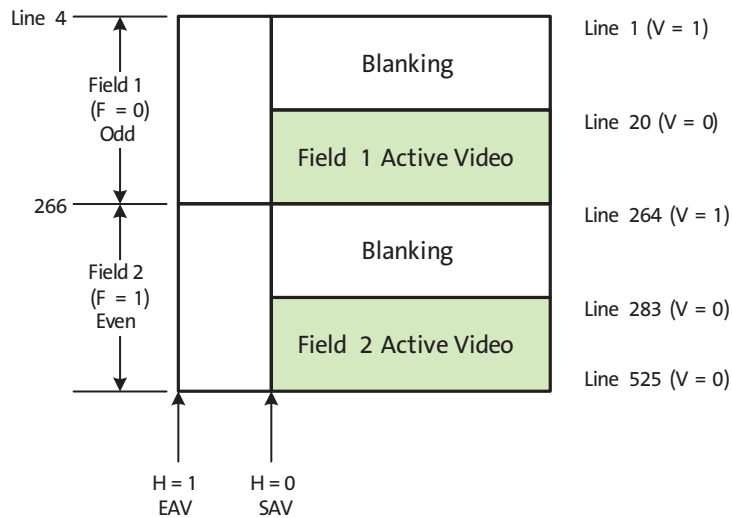


Figure 37 on page 56 shows detailed vertical blanking information for NTSC timing. See Table 25 on page 56 for data on field, vertical blanking, EAV, and SAV states.

**Figure 37: Typical CCIR656 Vertical Blanking Intervals for 525/60 Video System**



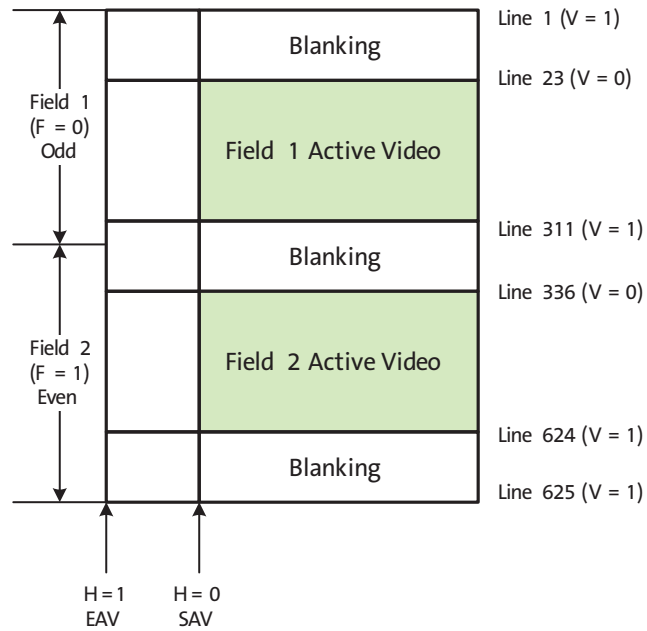
**Table 25: Field, Vertical Blanking, EAV, and SAV States 525/60 Video System**

| Line Number | F | V | H (EAV) | H (SAV) |
|-------------|---|---|---------|---------|
| 1–3         | 1 | 1 | 1       | 0       |
| 4–9         | 0 | 1 | 1       | 0       |
| 20–263      | 0 | 0 | 1       | 0       |
| 264–265     | 0 | 1 | 1       | 0       |
| 266–282     | 1 | 1 | 1       | 0       |
| 283–525     | 1 | 0 | 1       | 0       |

Figure 38 shows detailed vertical blanking information for PAL timing. See Table 26 on page 57 for data on field, vertical blanking, EAV, and SAV states.



**Figure 38: Typical CCIR656 Vertical Blanking Intervals for 625/50 Video System**



**Table 26: Field, Vertical Blanking, EAV, and SAV States for 625/50 Video System**

| Line Number | F | V | H (EAV) | H (SAV) |
|-------------|---|---|---------|---------|
| 1–22        | 0 | 1 | 1       | 0       |
| 23–310      | 0 | 0 | 1       | 0       |
| 311–312     | 0 | 1 | 1       | 0       |
| 313–335     | 1 | 1 | 1       | 0       |
| 336–623     | 1 | 0 | 1       | 0       |
| 624–625     | 1 | 1 | 1       | 0       |

## Reset and Clocks

### Reset

Power-up reset is asserted or de-asserted with the RESET\_BAR pin, which is active LOW. In the reset state, all control registers are set to default values. See “Device Configuration” on page 33 for more details on Auto, Host, and Flash configurations.

Soft reset is asserted or de-asserted by the two-wire serial interface program. In soft-reset mode, the two-wire serial interface and the register bus are still running. All control registers are reset using default values.

### Clocks

The MT9V136 has two primary clocks:

- A master clock coming from the EXTCLK signal.
- In default mode, a pixel clock (PIXCLK) running at  $2 * EXTCLK$ . In raw Bayer bypass mode, PIXCLK runs at the same frequency as EXTCLK.

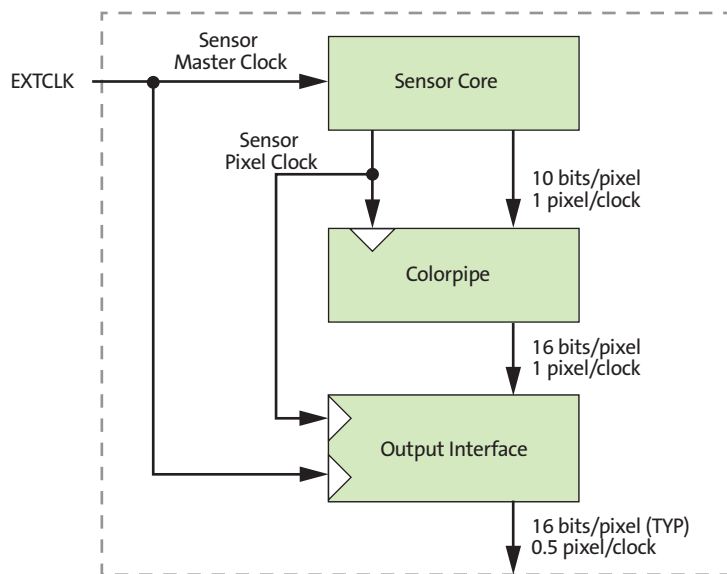
When the MT9V136 operates in sensor stand-alone mode, the image flow pipeline clocks can be shut off to conserve power.

The sensor core is a master in the system. The sensor core frame rate defines the overall image flow pipeline frame rate. Horizontal blanking and vertical blanking are influenced by the sensor configuration, and are also a function of certain image flow pipeline functions. The relationship of the primary clocks is depicted in Figure 39.

The image flow pipeline typically generates up to 16 bits per pixel—for example, YCbCr or 565RGB—but has only an 8-bit port through which to communicate this pixel data.

To generate NTSC or PAL format images, the sensor core requires a 27 MHz clock.

**Figure 39: Primary Clock Relationships**



## Floating Inputs

The following MT9V136 pins cannot be floated:

- SDATA—This pin is bidirectional and should not be floated
- TRST\_N

## Output Data Ordering

**Table 27: Output Data Ordering in DOUT RGB Mode**

| Mode (Swap Disabled) | Byte   | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----------------------|--------|----|----|----|----|----|----|----|----|
| 565RGB               | First  | R7 | R6 | R5 | R4 | R3 | G7 | G6 | G5 |
|                      | Second | G4 | G3 | G2 | B7 | B6 | B5 | B4 | B3 |
| 555RGB               | First  | 0  | R7 | R6 | R5 | R4 | R3 | G7 | G6 |
|                      | Second | G5 | G4 | G3 | B7 | B6 | B5 | B4 | B3 |
| 444xRGB              | First  | R7 | R6 | R5 | R4 | G7 | G6 | G5 | G4 |
|                      | Second | B7 | B6 | B5 | B4 | 0  | 0  | 0  | 0  |
| x444RGB              | First  | 0  | 0  | 0  | 0  | R7 | R6 | R5 | R4 |
|                      | Second | G7 | G6 | G5 | G4 | B7 | B6 | B5 | B4 |

Note: PIXCLK is 54 MHz when EXTCLK is 27 MHz.

**Table 28: Output Data Ordering in Sensor Stand-Alone Mode**

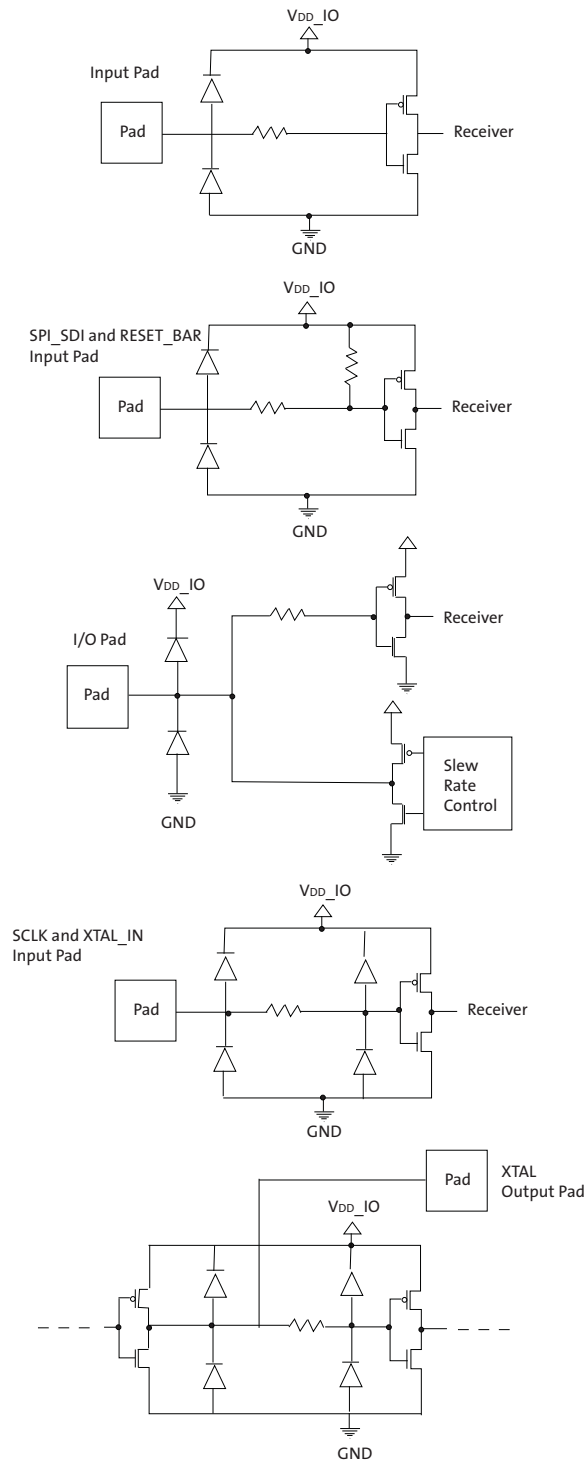
| Mode          | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Dout_LSB1 | Dout_LSB0 |
|---------------|----|----|----|----|----|----|----|----|-----------|-----------|
| 10-bit Output | B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1        | B0        |

Note: PIXCLK is 27 MHz when EXTCLK is 27 MHz.

## I/O Circuitry

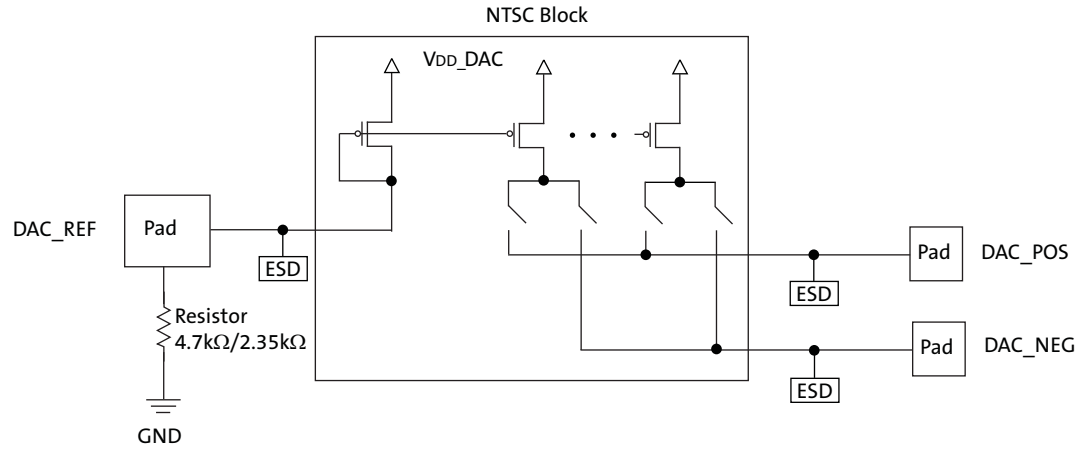
Figure 40 illustrates typical circuitry used for each input, output, or I/O pad.

**Figure 40: Typical I/O Equivalent Circuits**



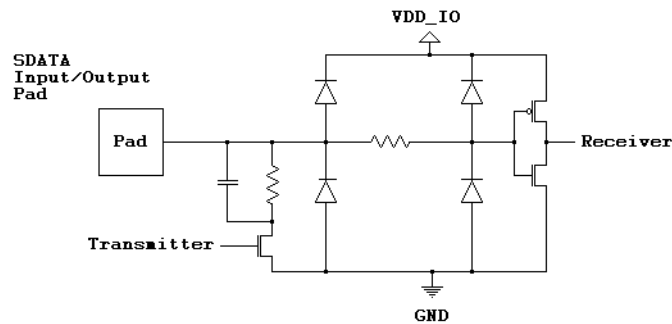
**Note:** All I/O circuitry shown above is for reference only. The actual implementation may be different.

Figure 41: NTSC Block



Note: All I/O circuitry shown above is for reference only. The actual implementation may be different.

Figure 42: Serial Interface



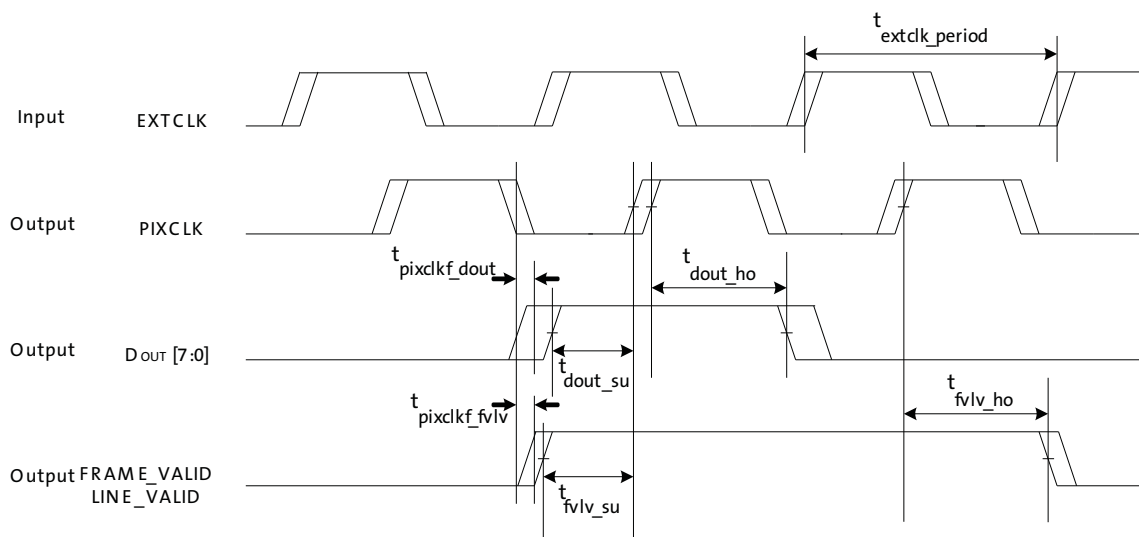
## I/O Timing

### Digital Output

By default, the MT9V136 launches pixel data, FV, and LV synchronously with the falling edge of PIXCLK. The expectation is that the user captures data, FV, and LV using the rising edge of PIXCLK. The timing diagram is shown in Figure 43.

As an option, the polarity of the PIXCLK can be inverted from the default by programming R0x0016[14].

**Figure 43: Digital Output I/O Timing**



**Table 29: Parallel Digital Output I/O Timing**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$ ; Default slew rate

| Signal              | Parameter            | Conditions        | Min | Typ | Max  | Unit |
|---------------------|----------------------|-------------------|-----|-----|------|------|
| EXTCLK              | $f_{extclk}$         | max $\pm 100$ ppm | –   | 27  | –    | MHz  |
|                     | $t_{extclk\_period}$ |                   | –   | 37  | –    | ns   |
|                     | Duty cycle           |                   | 45  | 50  | 55   | %    |
| PIXCLK <sup>1</sup> | $f_{pixclk}$         |                   | –   | 27  | –    | MHz  |
|                     | $t_{pixclk\_period}$ |                   | –   | 37  | –    | ns   |
|                     | Duty cycle           |                   | 45  | 50  | 55   | %    |
| DATA[7:0]           | $t_{pixclkf\_dout}$  |                   | –2  | 0   | 2    | ns   |
|                     | $t_{dout\_su}$       |                   | 8   | –   | 18.5 | ns   |
|                     | $t_{dout\_ho}$       |                   | 8   | –   | 18.5 | ns   |
| FV/LV               | $t_{pixclkf\_fvlv}$  |                   | –2  | 0   | 2    | ns   |
|                     | $t_{fvlv\_su}$       |                   | 8   | –   | 18.5 | ns   |
|                     | $t_{fvlv\_ho}$       |                   | 8   | –   | 18.5 | ns   |

Note: PIXCLK can be inverted from the default by programming R0x0016[14].

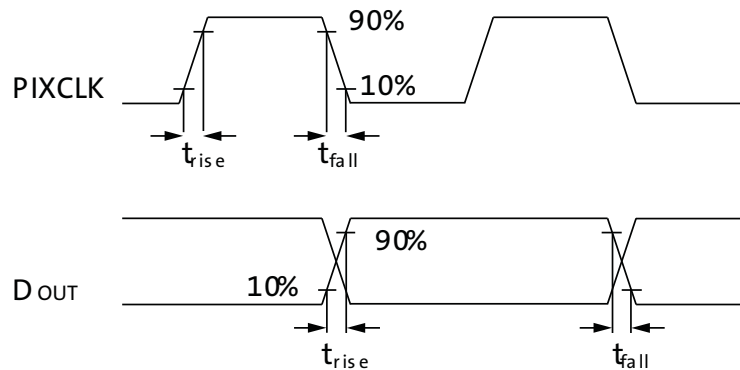
Slew Rate

**Table 30: Slew Rate for PIXCLK and DOUT**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$ ;  $T = 25^\circ\text{C}$ ;  $C_{LOAD} = 40 \text{ pF}$

| PIXCLK       |                   |                   | DOUT[7:0]   |                   |                   | Unit |
|--------------|-------------------|-------------------|-------------|-------------------|-------------------|------|
| R0x30 [10:8] | Typical Rise Time | Typical Fall Time | R0x30 [2:0] | Typical Rise Time | Typical Fall Time |      |
| 000          | 6.5               | 6.3               | 000         | 6.5               | 6.3               | ns   |
| 001          | 4.8               | 4.6               | 001         | 4.8               | 4.6               | ns   |
| 010          | 3.9               | 3.8               | 010         | 3.9               | 3.8               | ns   |
| 011          | 3.7               | 3.7               | 011         | 3.7               | 3.7               | ns   |
| 100          | 3.6               | 3.6               | 100         | 3.6               | 3.6               | ns   |
| 101          | 3.5               | 3.5               | 101         | 3.5               | 3.5               | ns   |
| 110          | 3.4               | 3.4               | 110         | 3.4               | 3.4               | ns   |
| 111          | 3.3               | 3.3               | 111         | 3.3               | 3.3               | ns   |

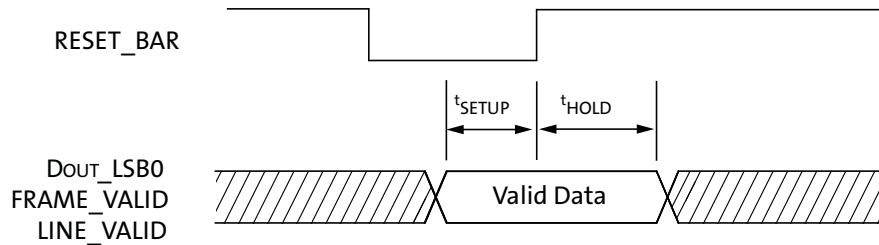
**Figure 44: Slew Rate Timing**



### Configuration Timing

During start-up, the Dout\_LSB0, LV and FV are sampled. Setup and hold timing for the RESET\_BAR signal with respect to DOUT\_LSB0, LV, and FV are shown in Figure 45 and Table 31. These signals are sampled once by the on-chip firmware, which yields a long  $t_{\text{Hold}}$  time.

**Figure 45: Configuration Timing**

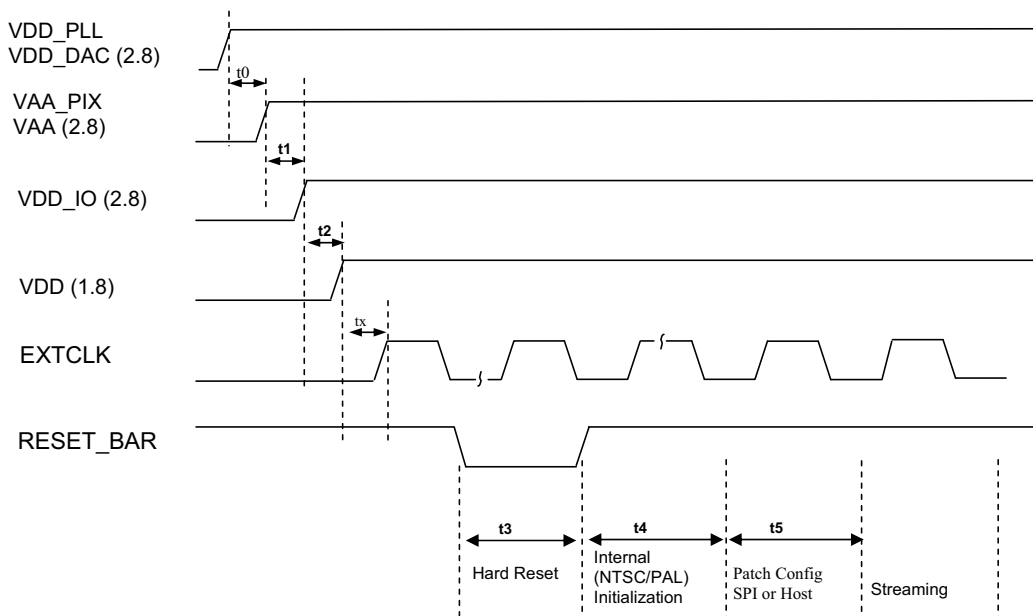


**Table 31: Configuration Timing**

| Signal                             | Parameter          | Min | Typ | Max | Unit          |
|------------------------------------|--------------------|-----|-----|-----|---------------|
| DOUT_LSB0, FRAME_VALID, LINE_VALID | $t_{\text{SETUP}}$ | 0   |     |     | $\mu\text{s}$ |
|                                    | $t_{\text{HOLD}}$  | 50  |     |     | $\mu\text{s}$ |



Figure 46: Power Up Sequence



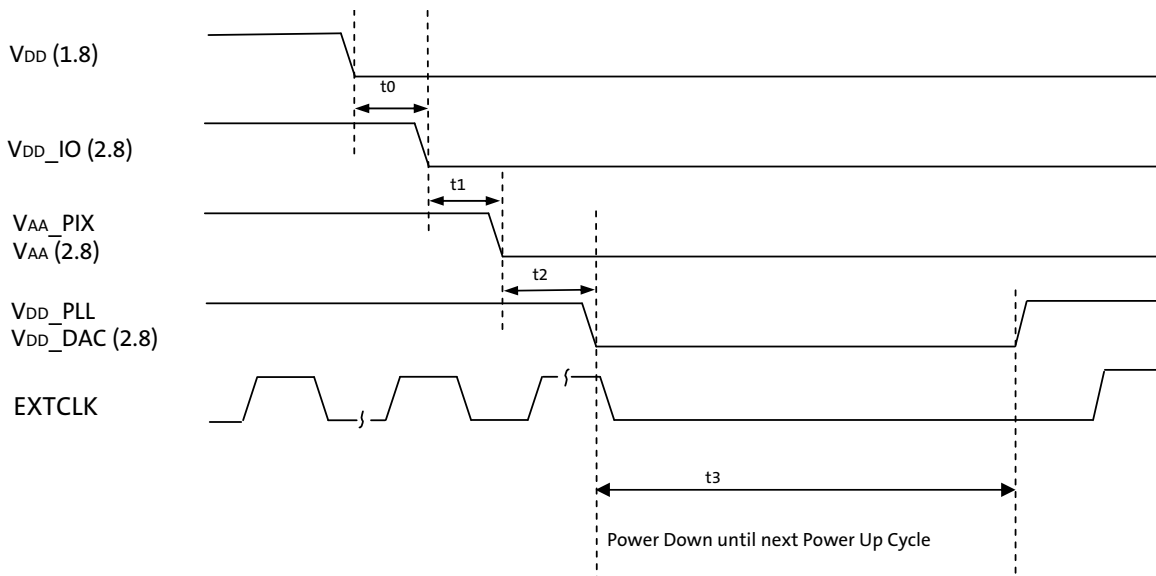
- Notes:
1. RESET\_BAR may not exceed  $V_{DD\_IO} + 0.3V$ .
  2. The 2.8V plane (VAA, VAA\_PIX, VDD\_PLL, VDD\_DAC, VDD\_IO) must remain at a higher voltage than the 1.8V core voltage at all times.

Table 32: Power Up Sequence

| Definition                           | Symbol | Minimum | Typical | Maximum | Unit        |
|--------------------------------------|--------|---------|---------|---------|-------------|
| VDD_PLL to VAA/VAA_PIX               | $t_0$  | 0       | –       | –       | $\mu S$     |
| VAA/VAA_PIX to VDD_IO                | $t_1$  | 0       | –       | –       | $\mu S$     |
| VDD_IO to VDD                        | $t_2$  | 0       | –       | –       | $\mu S$     |
| Xtal settle time                     | $t_x$  | –       | $30^1$  | –       | mS          |
| Hard Reset                           | $t_3$  | $10^2$  | –       | –       | Clock cycle |
| Internal Initialization              | $t_4$  | 50      | –       | –       | mS          |
| Patch Load (SPI or I <sup>2</sup> C) | $t_5$  | –       | $400^3$ | –       | mS          |

- Notes:
1. Xtal settling time is component-dependent (Xtal, Oscillator, etc) and usually takes about 10mS ~100mS.
  2. Hard reset time is the minimum time required after power rails are settled. Ten clock cycles are required for the sensor itself, assuming all power rails are settled. In a circuit where Hard reset is performed by the RC circuit, then the RC time must include the all power rail settle time and Xtal
  3. This is required to load necessary patches via Flash mode (SPI) or Host mode (two-wire serial interface). Loading time varies depending on the number of patches and bus speed.

**Figure 47: Power Down Sequence**

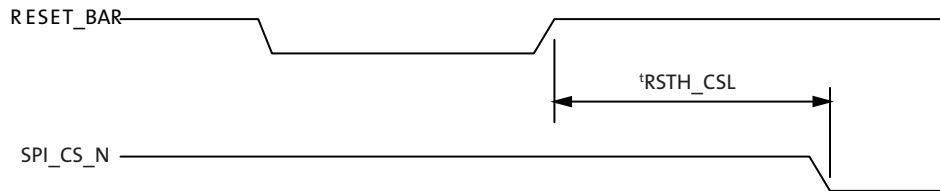


**Table 33: Power Down Sequence**

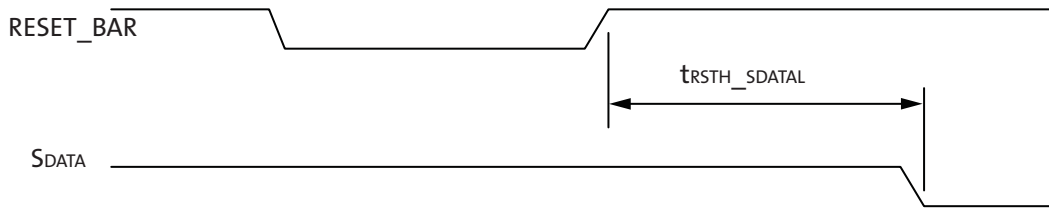
| Definition                          | Symbol | Minimum          | Typical | Maximum | Unit |
|-------------------------------------|--------|------------------|---------|---------|------|
| VDD to VDD_IO                       | t0     | 0                | –       | –       | μS   |
| VDD_IO to VAA/VAA_PIX               | t1     | 0                | –       | –       | μS   |
| VAA/VAA_PIX to VDD_PLL/DAC          | t2     | 0                | –       | –       | μS   |
| Power Down until Next Power Up Time | t3     | 100 <sup>1</sup> | –       | –       | ms   |

(1) t3 is required between power down and next power up time, all decoupling caps from regulators must completely discharged before next power up.

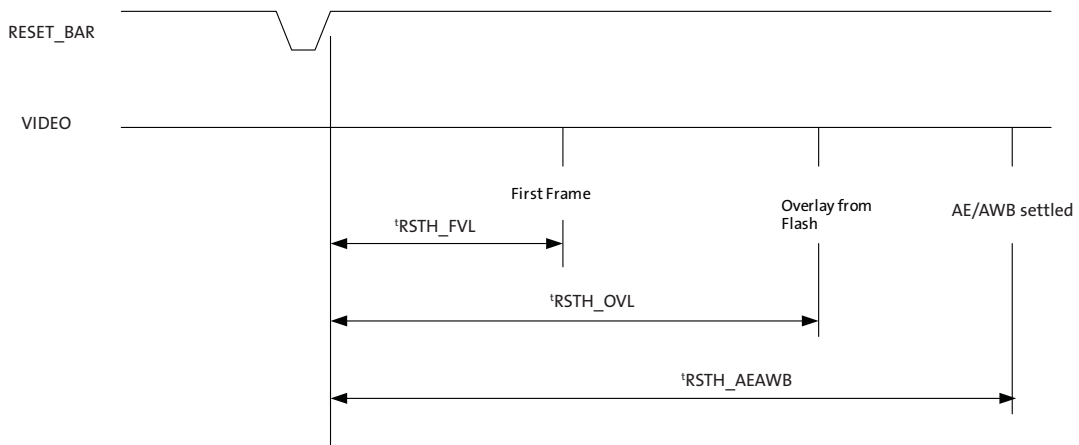
**Figure 48: Reset to SPI Access Delay**



**Figure 49: Reset to Serial Access Delay**



**Figure 50: Reset to AE/AWB Image**



**Table 34: RESET\_BAR Delay Parameters**

| Parameter                        | Name         | Condition | Min | Typ | Max | Unit |
|----------------------------------|--------------|-----------|-----|-----|-----|------|
| Power up delay 2.8V to 1.8V      |              |           | 0.1 | –   | –   | ms   |
| RESET_BAR HIGH to SPI_CS_N LOW   | tRSTH_CSL    |           | 18  | –   | –   | ms   |
| RESET_BAR HIGH to SDATA LOW      | tRSTH_SDATAL |           | 1.8 | –   | –   | ms   |
| RESET_BAR HIGH to FRAME_VALID    | tRSTH_FVL    |           | 235 | –   | –   | ms   |
| RESET_BAR HIGH to first Overlay  | tRSTH_OVL    |           | 235 | –   | –   | ms   |
| RESET_BAR HIGH to AE/AWB settled | tRSTH_AEAWB  |           | –   | 400 | –   | ms   |

## Electrical Specifications

Figure 51: SPI Output Timing



Table 35: SPI Data Setup and Hold Timing

| Parameter       | Description  | Min    | Typ | Max | Units |
|-----------------|--|--------|-----|-----|-------|
| $f_{SPI\_SCLK}$ | SPI_SCLK Frequency   | 1.6875 | 4.5 | 18  | MHz   |
| $t_{su}$        | Setup time   | –      | –   | 110 | ns    |
| $t_{SCLK\_SDO}$ | Hold time  | –      | –   | 110 | ns    |
| $t_{CS\_SCLK}$  | Delay from falling edge of SPI_CS_N to rising edge of SPI_SCLK | –      | 230 | –   | ns    |

**Caution** Stresses greater than those listed in Table 36 may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

**Table 36: Absolute Maximum Ratings**

| Symbol  | Parameter                | Rating |            | Unit |
|---------|--------------------------|--------|------------|------|
|         |                          | Min    | Max        |      |
| VDD     | Digital power (1.8V)     | -0.3   | 2.4        | V    |
| VDD_IO  | I/O power (2.8v)         | -0.3   | 4          | V    |
| VAA     | VAA Analog power (2.8V)  | -0.3   | 4          | V    |
| VAA_PIX | Pixel array power (2.8v) | -0.3   | 4          | V    |
| VDD_PLL | PLL power (2.8V)         | -0.3   | 4          | V    |
| VDD_DAC | DAC power (2.8V)         | -0.3   | 4          | V    |
| VIN     | DC Input Voltage         | -0.3   | VDD_IO+0.3 | V    |
| VOUT    | DC Output Voltage        | -0.3   | VDD_IO+0.3 | V    |
| TSTG    | Storage temperature      | -50    | 150        | °C   |

**Table 37: Electrical Characteristics and Operating Conditions**

| Parameter <sup>1</sup>         | Condition           | Min  | Typ | Max  | Unit |
|--------------------------------|---------------------|------|-----|------|------|
| Core digital voltage (VDD)     | –                   | 1.7  | 1.8 | 1.9  | V    |
| IO digital voltage (VDD_IO)    | –                   | 2.66 | 2.8 | 2.94 | V    |
| Video DAC voltage (VDD_DAC)    | –                   | 2.66 | 2.8 | 2.94 | V    |
| PLL Voltage (VDD_PLL)          | –                   | 2.66 | 2.8 | 2.94 | V    |
| Analog voltage (VAA)           | –                   | 2.66 | 2.8 | 2.94 | V    |
| Pixel supply voltage (VAA_PIX) | –                   | 2.66 | 2.8 | 2.94 | V    |
| Leakage current                | EXTCLK: HIGH or LOW |      |     | 10   | µA   |
| Imager operating temperature   | –                   | –30  |     | +70  | °C   |
| Storage temperature            | –                   | –50  |     | +150 | °C   |

Notes: 1. VAA and VAA\_PIX must all be at the same potential to avoid excessive current draw. Care must be taken to avoid excessive noise injection in the analog supplies if all three supplies are tied together.

**Table 38: Video DAC Electrical Characteristics—Single-Ended Mode**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$

| Parameter         | Condition                    | Min | Typ         | Max  | Unit       |
|-------------------|------------------------------|-----|-------------|------|------------|
| Resolution        |                              | –   | 10          | -    | bits       |
| DNL               |                              | –   | 0.2         | 0.4  | bits       |
| INL               |                              | –   | 0.7         | 3.5  | bits       |
| Output local load | Output pad (DAC_POS)         | –   | 75          | -    | $\Omega$   |
|                   | Unused output (DAC_NEG)      | –   | 0           | -    | $\Omega$   |
| Output voltage    | Single-ended mode, code 000h | –   | .02         | -    | V          |
|                   | Single-ended mode, code 3FFh | –   | 1.30        | -    | V          |
| Output current    | Single-ended mode, code 000h | –   | 0.26        | -    | mA         |
|                   | Single-ended mode, code 3FFh | –   | 17.33       | -    | mA         |
| Supply current    | Estimate                     | –   | -           | 25.0 | mA         |
| DAC_REF           | DAC Reference                | –   | 1.15 +/-0.2 | -    | V          |
| R DAC_REF         | DAC Reference                | –   | 4.7         | -    | K $\Omega$ |

**Table 39: Video DAC Electrical Characteristics—Differential Mode**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$

| Parameter                     | Condition                                       | Min | Typ         | Max  | Unit       |
|-------------------------------|---|-----|-------------|------|------------|
| DNL                           |   | –   | 0.2         | 0.25 | Bits       |
| INL                           |   | –   | 0.8         | 2.5  | Bits       |
| Output local load             | Differential mode per pad (DAC_POS and DAC_NEG) | –   | 37.5        | –    | $\Omega$   |
| Output voltage                | Differential mode, code 000h, pad dacp          | –   | .02         | –    | V          |
|                               | Differential mode, code 000h, pad dacn          | –   | 1.30        | –    | V          |
|                               | Differential mode, code 3FFh, pad dacp          | –   | 1.30        | –    | V          |
|                               | Differential mode, code 3FFh, pad dacn          | –   | .02         | –    | V          |
| Output current                | Differential mode, code 000h, pad dacp          | –   | .53         | –    | mA         |
|                               | Differential mode, code 000h, pad dacn          | –   | 34.7        | –    | mA         |
|                               | Differential mode, code 3FFh, pad dacp          | –   | 34.7        | –    | mA         |
|                               | Differential mode, code 3FFh, pad dacn          | –   | .53         | –    | mA         |
| Differential output, midlevel |   | –   | 0.65        | –    | V          |
| Supply current                | Estimate  | –   | –           | 50   | mA         |
| DAC_REF                       | DAC Reference                                   | –   | 1.15 +/-0.2 |      | V          |
| R DAC_REF                     | DAC Reference                                   |     | 2.35        |      | K $\Omega$ |

**Table 40: Digital I/O Parameters**  
T<sub>A</sub> = Ambient = 25°C; All supplies at 2.8V

| Signal          | Parameter          | Definitions                                    | Condition                                      | Min                      | Typ                | Max                      | Unit |
|-----------------|--------------------|--|--|--------------------------|--------------------|--------------------------|------|
| All Outputs     |                    | Load capacitance                               |  | 1                        | –                  | 30                       | pF   |
|                 |                    | Output signal slew                             | 2.8V, 30pF load                                | –                        | –                  | –                        | V/ns |
|                 |                    |  | 2.8V, 5pF load                                 | –                        | –                  | –                        | V/ns |
|                 | V <sub>OH</sub>    | Output high voltage                            |  | –                        | V <sub>DD_IO</sub> | –                        | V    |
|                 | V <sub>OL</sub>    | Output low voltage                             |  | –0.3                     | –                  | –                        | V    |
|                 | I <sub>OH</sub>    | Output high current                            | V <sub>DD</sub> = 2.8V, V <sub>OH</sub> = 2.4V | –                        | –                  | 8                        | mA   |
| I <sub>OL</sub> | Output low current | V <sub>DD</sub> = 2.8V, V <sub>OL</sub> = 0.4V | –  | –                        | 8                  | mA                       |      |
| All Inputs      | V <sub>IH</sub>    | Input high voltage                             | V <sub>DD</sub> = 2.8V                         | 0.7 * V <sub>DD_IO</sub> | –                  | V <sub>DD_IO</sub> + 0.3 | V    |
|                 | V <sub>IL</sub>    | Input low voltage                              | V <sub>DD</sub> = 2.8V                         | –0.3                     | –                  | 0.3 * V <sub>DD_IO</sub> | V    |
|                 | I <sub>IN</sub>    | Input leakage current                          |  | –2                       | –                  | 2                        | μA   |
|                 | Signal CAP         | Input signal capacitance                       |  | –                        | 3.5                | –                        | pF   |

Notes: 1. All inputs are protected and may be active when All supplies (2.8V and 1.8V) are turned off.

## Power Consumption, Operating Mode

**Table 41: Power Consumption – Condition 1**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$

| Power Plane | Supply | Condition 1  | Typ Power | Max Power | Unit |
|-------------|--------|--------------|-----------|-----------|------|
| VDD         | 1.8    |              | 140.4     | 162       | mW   |
| VDD_IO      | 2.8    | Parallel off | 4.2       | 8.4       | mW   |
| VAA         | 2.8    |              | 89.6      | 112       | mW   |
| VAA_PIX     | 2.8    |              | 1.96      | 5.04      | mW   |
| VDD_DAC     | 2.8    | Single 75(1) | 39.2      | 44.8      | mW   |
| VDD_PLL     | 2.8    |              | 13.44     | 16.8      | mW   |
| Total       |        |              | 288.8     | 349.04    | mW   |

Analog output uses single-ended mode: DAC\_Pos = 75Ω, DAC\_Neg = open, parallel output is disabled.

**Table 42: Power Consumption – Condition 2**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$

| Power Plane | Supply | Condition 2  | Typ Power | Max Power | Unit |
|-------------|--------|--------------|-----------|-----------|------|
| VDD         | 1.8    |              | 140.4     | 162       | mW   |
| VDD_IO      | 2.8    | Parallel on  | 42        | 50.4      | mW   |
| VAA         | 2.8    |              | 89.6      | 112       | mW   |
| VAA_PIX     | 2.8    |              | 1.96      | 5.04      | mW   |
| VDD_DAC     | 2.8    | Single 75(1) | 39.2      | 44.8      | mW   |
| VDD_PLL     | 2.8    |              | 13.44     | 16.8      | mW   |
| Total       |        |              | 326.6     | 391.04    | mW   |

Analog output uses single-ended mode: DAC\_Pos = 75Ω, DAC\_Neg = open, parallel output is enabled.



## NTSC Signal Parameters

**Table 43: NTSC Signal Parameters**

<sup>f</sup>EXTCLK = 27 MHz; VDD = 1.8V; VDD\_IO = 2.8V; VAA = 2.8V; VAA\_PIX = 2.8V;  
VDD\_PLL = 2.8V; VDD\_DAC = 2.8V

| Parameter                            | Conditions | Min      | Typ      | Max      | Units | Notes      |
|--------------------------------------|------------|----------|----------|----------|-------|------------|
| Line Frequency                       |            | 15734.25 | 15734.27 | 15734.28 | Hz    |            |
| Field Frequency                      |            | 59.94    | 59.94    | 59.94    | Hz    |            |
| Sync Rise Time                       |            | 148      | 148      | 148      | ns    |            |
| Sync Fall Time                       |            | 148      | 148      | 148      | ns    |            |
| Sync Width                           |            | 4.74     | 4.74     | 4.74     | μs    |            |
| Sync Level                           |            | 38       | 40       | 42       | IRE   | 2, 4       |
| Burst Level                          |            | 38       | 40       | 42       | IRE   | 2, 4       |
| Sync to Setup<br>(with pedestal off) |            | 9.44     | 9.44     | 9.44     | μs    |            |
| Sync to Burst Start                  |            | 5.33     | 5.33     | 5.33     | μs    |            |
| Front Porch                          |            | 1.33     | 1.33     | 1.33     | μs    |            |
| Black Level                          |            |          | 7.5      |          | IRE   | 1, 2, 4    |
| White Level                          |            |          | 100      |          | IRE   | 1, 2, 3, 4 |

- Notes:
1. Black and white levels are referenced to the blanking level.
  2. NTSC convention standardized by the IRE (1 IRE = 7.14mV).
  3. Encoder contrast setting R0x011 = R0x001 = 0.
  4. DAC ref = 2.35kΩ, load = 37.5Ω.

Figure 52: Video Timing

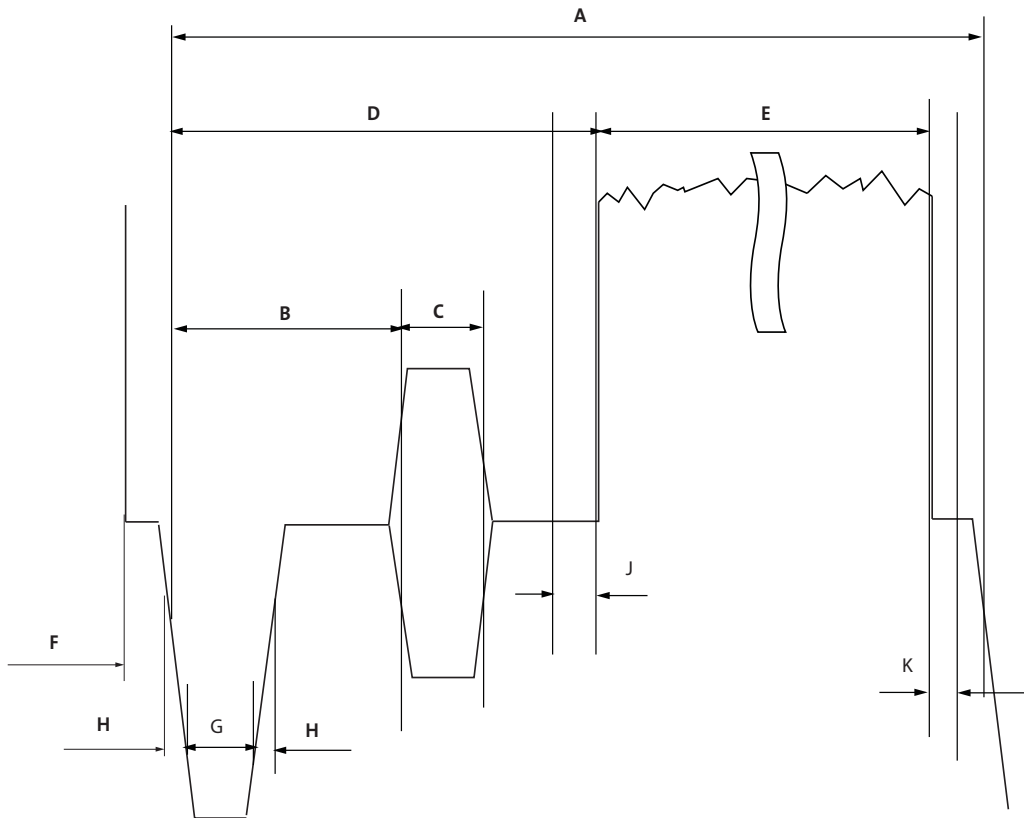


Table 44: Video Timing

|   | Signal                   | NTSC<br>27 MHz | PAL<br>27 MHz | Units  |
|---|--------------------------|----------------|---------------|--------|
| A | H Period                 | 1716           | 1728          | Clocks |
| B | Hsync to burst           | 144            | 153           | Clocks |
| C | burst                    | 63             | 66            | Clocks |
| D | Hsync to Signal          | 255            | 279           | Clocks |
| E | Video Signal             | 1423           | 1413          | Clocks |
| F | Front                    | 36             | 39            | Clocks |
| G | Hsync Period             | 128            | 128           | Clocks |
| H | Sync rising/falling edge | 4              | 4             | Clocks |
| J | Back overscan (BOS)      | 9              | 14            | Clocks |
| K | Front overscan (FOS)     | 8              | 13            | Clocks |

Figure 53: Equivalent Pulse

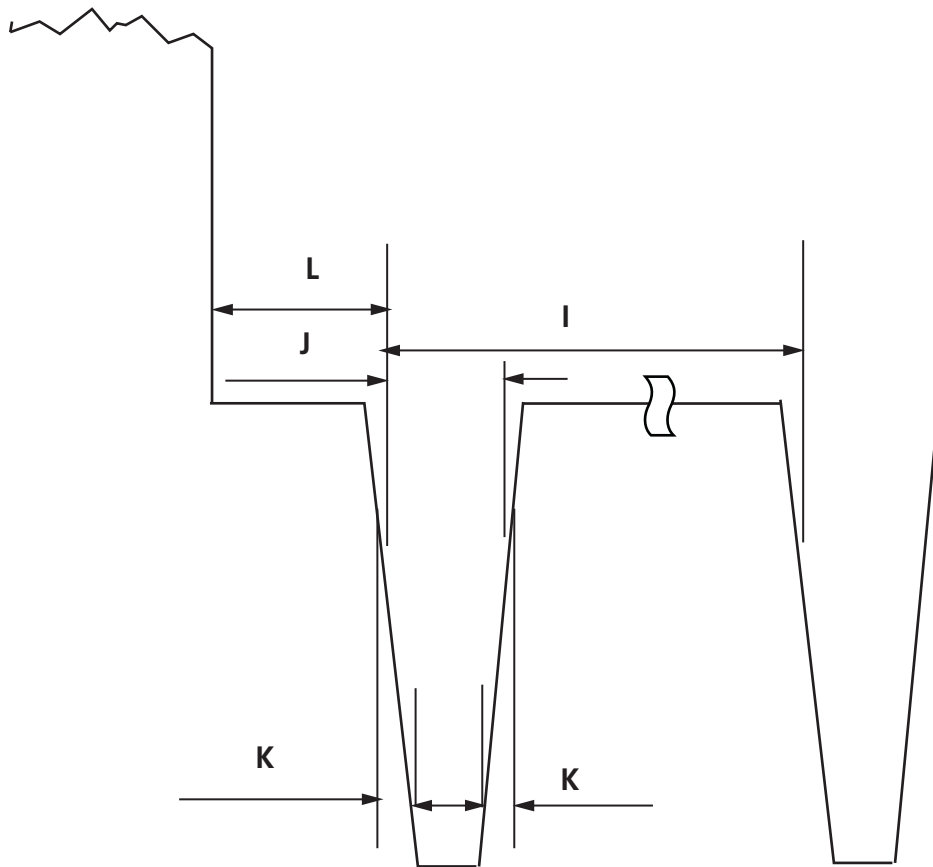


Table 45: Equivalent Pulse

|   | Signal                    | NTSC<br>27 MHz | PAL<br>27 MHz | Units  |
|---|---------------------------|----------------|---------------|--------|
| I | H/2 Period                | 858            | 864           | Clocks |
| J | Pulse width               | 64             | 64            | Clocks |
| K | Pulse rising/falling edge | 4              | 4             | Clocks |
| L | Signal to pulse           | 38             | 41            | Clocks |

Figure 54: V Pulse

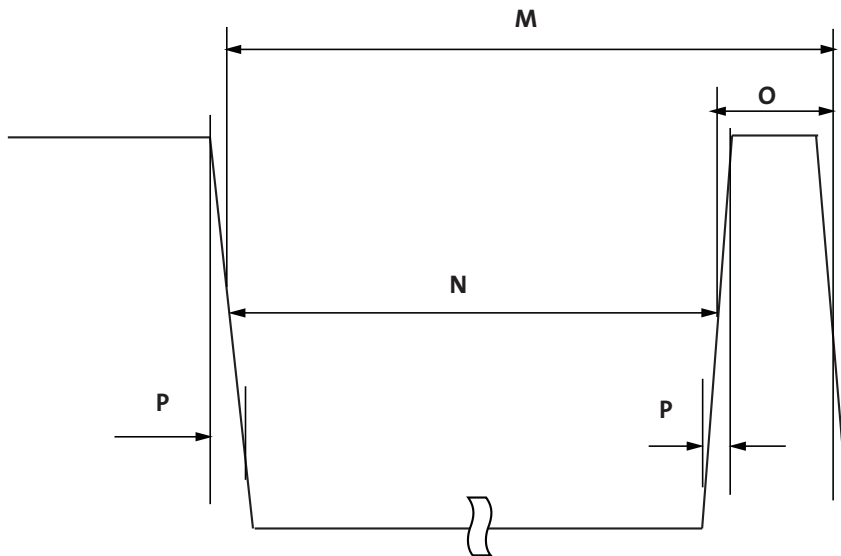


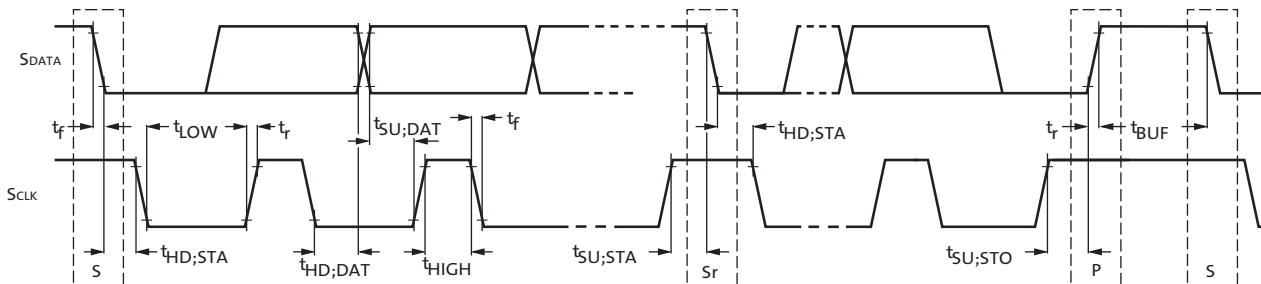
Table 46: V Pulse

|   | Signal                    | NTSC<br>27 MHz | PAL<br>27 MHz | Units  |
|---|---------------------------|----------------|---------------|--------|
| M | H/2 Period                | 858            | 864           | Clocks |
| N | Pulse width               | 730            | 736           | Clocks |
| O | V pulse interval          | 128            | 128           | Clocks |
| P | Pulse rising/falling edge | 4              | 4             | Clocks |

## Two-Wire Serial Bus Timing

Figure 55 and Table 47 describe the timing for the two-wire serial interface.

**Figure 55: Two-Wire Serial Bus Timing Parameters**



**Table 47: Two-Wire Serial Bus Characteristics**

$f_{EXTCLK} = 27 \text{ MHz}$ ;  $V_{DD} = 1.8\text{V}$ ;  $V_{DD\_IO} = 2.8\text{V}$ ;  $V_{AA} = 2.8\text{V}$ ;  $V_{AA\_PIX} = 2.8\text{V}$ ;  
 $V_{DD\_PLL} = 2.8\text{V}$ ;  $V_{DD\_DAC} = 2.8\text{V}$ ;  $T_A = 25^\circ\text{C}$

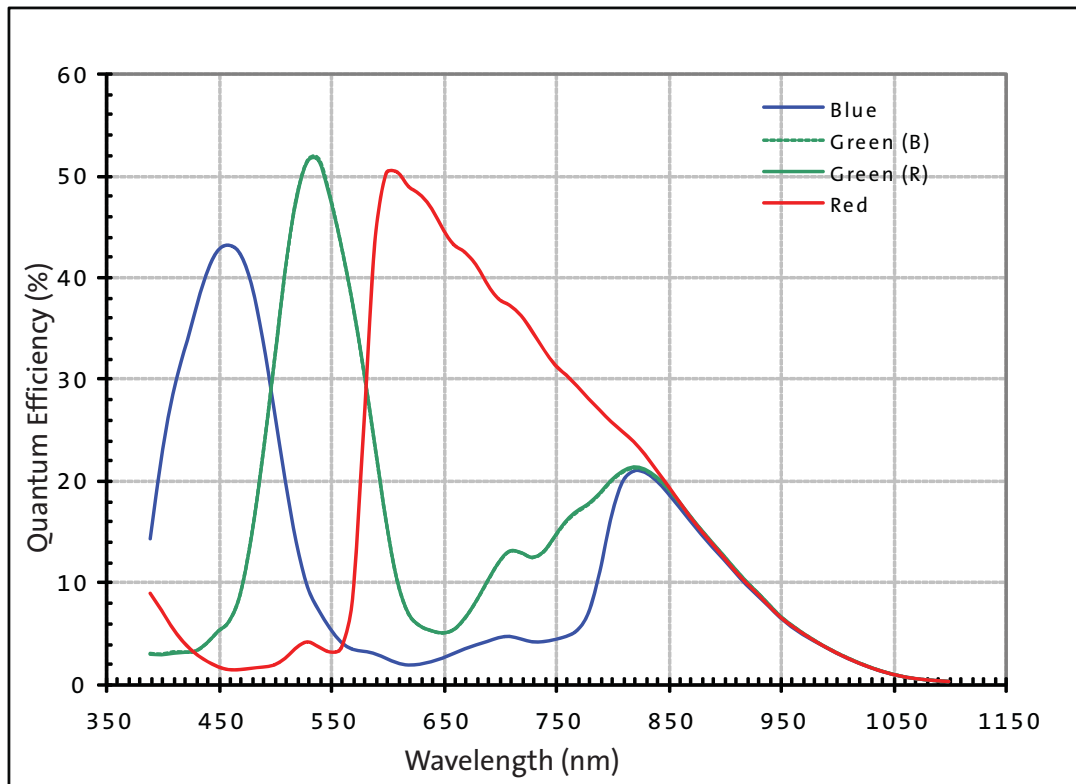
| Parameter  | Symbol         | Standard-Mode  |                   | Fast-Mode        |                  | Unit             |
|--|----------------|----------------|-------------------|------------------|------------------|------------------|
|  |                | Min            | Max               | Min              | Max              |                  |
| SCLK Clock Frequency   | $f_{SCL}$      | 0              | 100               | 0                | 400              | KHz              |
| Hold time (repeated) START condition.<br>After this period, the first clock pulse is generated | $t_{HD;STA}$   | 4.0            | -                 | 0.6              | -                | $\mu\text{S}$    |
| LOW period of the SCLK clock   | $t_{LOW}$      | 4.7            | -                 | 1.3              | -                | $\mu\text{S}$    |
| HIGH period of the SCLK clock  | $t_{HIGH}$     | 4.0            | -                 | 0.6              | -                | $\mu\text{S}$    |
| Set-up time for a repeated START condition   | $t_{SU;STA}$   | 4.7            | -                 | 0.6              | -                | $\mu\text{S}$    |
| Data hold time:  | $t_{HD;DAT}$   | 0 <sup>4</sup> | 3.45 <sup>5</sup> | 0 <sup>6</sup>   | 0.9 <sup>5</sup> | $\mu\text{S}$    |
| Data set-up time   | $t_{SU;DAT}$   | 250            | -                 | 100 <sup>6</sup> | -                | nS               |
| Rise time of both SDATA and SCLK signals   | $t_r$          | -              | 1000              | $20 + 0.1Cb^7$   | 300              | nS               |
| Fall time of both SDATA and SCLK signals   | $t_f$          | -              | 300               | $20 + 0.1Cb^7$   | 300              | nS               |
| Set-up time for STOP condition   | $t_{SU;STO}$   | 4.0            | -                 | 0.6              | -                | $\mu\text{S}$    |
| Bus free time between a STOP and START condition   | $t_{BUF}$      | 4.7            | -                 | 1.3              | -                | $\mu\text{S}$    |
| Capacitive load for each bus line  | $C_b$          | -              | 400               | -                | 400              | pF               |
| Serial interface input pin capacitance   | $C_{IN\_SI}$   | -              | 3.3               | -                | 3.3              | pF               |
| SDATA max load capacitance   | $C_{LOAD\_SD}$ | -              | 30                | -                | 30               | pF               |
| SDATA pull-up resistor   | $R_{SD}$       | 1.5            | 4.7               | 1.5              | 4.7              | $\text{K}\Omega$ |

- Notes:
1. This table is based on I<sup>2</sup>C standard (v2.1 January 2000). Philips Semiconductor.
  2. Two-wire control is I<sup>2</sup>C-compatible.
  3. All values referred to  $V_{IHmin} = 0.9 V_{DD}$  and  $V_{ILmax} = 0.1V_{DD}$  levels. Sensor EXCLK = 27 MHz.
  4. A device must internally provide a hold time of at least 300 ns for the SDATA signal to bridge the undefined region of the falling edge of SCLK.
  5. The maximum  $t_{HD;DAT}$  has only to be met if the device does not stretch the LOW period ( $t_{LOW}$ ) of the SCLK signal.

6. A Fast-mode I<sup>2</sup>C-bus device can be used in a Standard-mode I<sup>2</sup>C-bus system, but the requirement  $t_{SU;DAT} \geq 250$  ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLK signal. If such a device does stretch the LOW period of the SCLK signal, it must output the next data bit to the SDATA line  $t_r \max + t_{SU;DAT} = 1000 + 250 = 1250$  ns (according to the Standard-mode I<sup>2</sup>C-bus specification) before the SCLK line is released.
7.  $C_b$  = total capacitance of one bus line in pF.

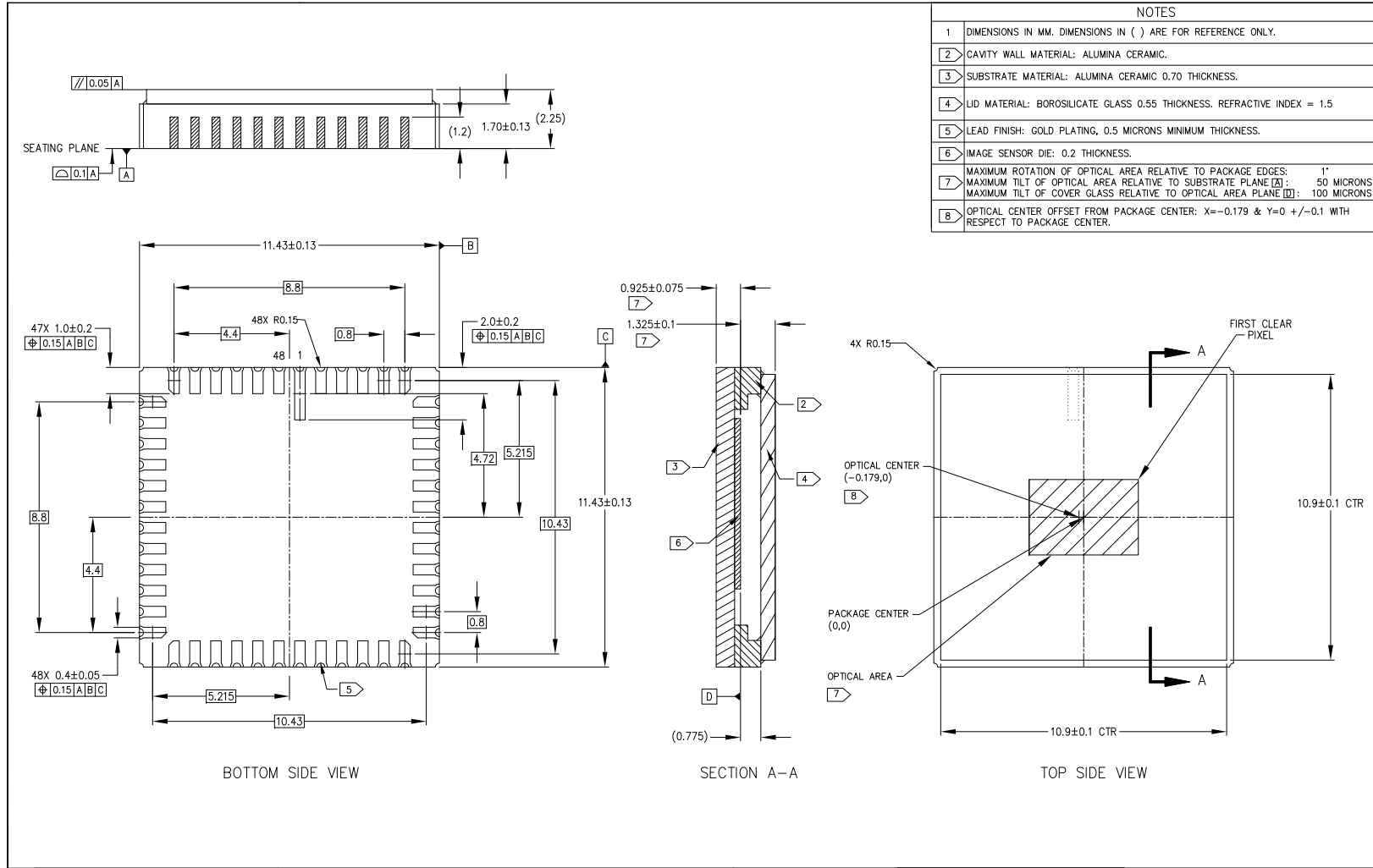
## Spectral Characteristics

Figure 56: Quantum Efficiency



# Package and Die Dimensions

Figure 57: 48-Pin CLCC Package Outline Drawing



## Revision History

|             |   |         |
|-------------|---|---------|
| Rev. J..... | <ul style="list-style-type: none"> <li>• Updated “Ordering Information” on page 4</li> <li>• Updated “Pixel Array Structure” on page 18</li> <li>• Updated “Sensor Active Pixel Array” on page 20</li> <li>• Updated Figure 57: “48-Pin CLCC Package Outline Drawing,” on page 79</li> </ul>  | 6/22/15 |
| Rev. H..... | <ul style="list-style-type: none"> <li>• Updated “Ordering Information” on page 4</li> </ul>  | 4/17/15 |
| Rev. G..... | <ul style="list-style-type: none"> <li>• Converted to ON Semiconductor template</li> <li>• Removed Confidential marking</li> </ul>  | 4/1/15  |
| Rev. F..... | <ul style="list-style-type: none"> <li>• Updated trademarks</li> <li>• Applied updated Aptina template</li> </ul>   | 5/17/12 |
| Rev. E..... | <ul style="list-style-type: none"> <li>• Updated capacitor value in “Crystal Usage” on page 12</li> <li>• Updated Note 4 in Table 43, “NTSC Signal Parameters,” on page 73</li> <li>• Updated Figure 55: “Two-Wire Serial Bus Timing Parameters,” on page 77</li> <li>• Updated Table 47, “Two-Wire Serial Bus Characteristics,” on page 77</li> </ul>  | 6/22/10 |
| Rev.D.....  | <ul style="list-style-type: none"> <li>• Updated package pitch number.</li> </ul>   | 5/25/10 |
| Rev. C..... | <ul style="list-style-type: none"> <li>• Updated title</li> <li>• Updated “Features” on page 1</li> <li>• Updated Table 1, “Key Parameters,” on page 1</li> <li>• Updated Table 6, “Available Part Numbers,” on page 4</li> <li>• Updated “On-Chip Overlay Generator” on page 3</li> <li>• Updated “General Description” on page 9</li> <li>• Moved register tables to a separate document, “MT9V136 Register and Variable Reference”</li> <li>• Updated Table 4, “Pin Descriptions,” on page 13</li> <li>• Updated Figure 4: “Pin Assignments,” on page 15</li> <li>• Updated Table 5, “Reset/Default State of Interfaces,” on page 15</li> <li>• Updated “Pixel Array Structure” on page 18</li> <li>• Deleted first sentence of 2nd paragraph of “Sensor Active Pixel Array” on page 20</li> <li>• Updated Figure 8: “Active Pixel Array,” on page 20</li> <li>• Updated Figure 9: “Pixel Color Pattern Detail (top right corner),” on page 21</li> <li>• Updated Figure 11: “Color Pipeline,” on page 23</li> <li>• Updated “Positional Gain Adjustments (PGA)” on page 26</li> <li>• Updated “Device Configuration” on page 33</li> <li>• Updated “Flash Configuration Mode” on page 34</li> <li>• Updated “Host Configuration” on page 34</li> <li>• Updated “Power Sequence” on page 35</li> </ul> | 4/5/10  |



- Updated Table 13, “SPI Flash Devices,” on page 35
- Updated “Command Flow” on page 38
- Updated “Set Parallel Mode - Normal (Overlay i656)” on page 39
- Updated “Summary of Host Commands” on page 39, including
  - Table 16, “System Manager Commands,” on page 39
  - Table 17, “Overlay Host Commands,” on page 39
  - Table 18, “GPIO Host Commands,” on page 40
  - Table 19, “Flash Manager Host Commands,” on page 40
  - Table 20, “Sequencer Host Commands,” on page 40
  - Table 21, “TX Manager Host Commands,” on page 41
- Updated “Overlay Capability” on page 47
- Updated “Overlay Adjustment” on page 49
- Changed “Overlay Number Generator” heading to “Overlay Character Generator” on page 51
- Updated “Modes and Timing” on page 54
- Updated and moved the section “I/O Circuitry” on page 60
- Updated Table 29, “Parallel Digital Output I/O Timing,” on page 62
- Replaced section on RESET\_N Timing with “Configuration Timing” on page 64
- Updated “Electrical Specifications” on page 68

**Rev. B** ..... **6/26/09**

- Updated to Production status
- Updated “Features” on page 1
- Updated Table 1, “Key Parameters,” on page 1 and Table 2, “Key Parameters (continued),” on page 2
- Updated “Available Part Numbers” on page 3
- Updated Table 4, “Pin Descriptions,” on page 12
- Updated “Pixel Array Structure” on page 18
- Updated “Sensor Pixel Array” on page 18
- Updated “Positional Gain Adjustments (PGA)” on page 23
- Updated “Usage Modes” on page 28
- Moved register tables and section on “How to Access Registers and Variables” to separate document
- Updated “Command Flow” on page 35
- Updated Figure 26: “Single Write to Random Location,” on page 42
- Updated Table 21, “TX Manager Host Commands,” on page 37
- Updated “Overlay Capability” on page 44
- Updated “Overlay Adjustment” on page 46
- Changed “Overlay Number Generator” to “Overlay Character Generator,” on page 48
- Added Table 36, “Absolute Maximum Ratings,” on page 66

A-Pix is a trademark of Semiconductor Components Industries, LLC (SCILLC) or its subsidiaries in the United States and/or other countries.

ON Semiconductor and the ON logo are registered trademarks of Semiconductor Components Industries, LLC (SCILLC) or its subsidiaries in the United States and/or other countries. SCILLC owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of SCILLC’s product/patent coverage may be accessed at [www.onsemi.com/site/pdf/Patent-Marketing.pdf](http://www.onsemi.com/site/pdf/Patent-Marketing.pdf). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. “Typical” parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.



|                   |          |
|-------------------|----------|
| Rev. A .....      | 11/21/08 |
| • Initial release |          |



Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



#### Как с нами связаться

**Телефон:** 8 (812) 309 58 32 (многоканальный)

**Факс:** 8 (812) 320-02-42

**Электронная почта:** [org@eplast1.ru](mailto:org@eplast1.ru)

**Адрес:** 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.