AS0140AT

1/4-Inch CMOS Image Sensor and Signal Processor

ON Semiconductor®

www.onsemi.com

General Description

The ON Semiconductor AS0140AT is a 1.0 MP format digital image sensor and image sensor processor for automotive viewing applications. The device includes full auto-functions support (AWB and AE) and ALTM (Adaptive Local Tone Mapping) to enhance HDR video. The AS0140AT implements a high-sensitivity 3.0 μm pixel with DR-Pix™ technology, and advanced noise reduction, to enable excellent low-light performance. It can be operated in interlaced (NTSC or PAL) or progressive modes, and captures images in either linear or high dynamic range modes. The AS0140AT may be operated in video (master) mode or in single frame trigger mode, providing flexibility for multi-camera systems.

Table 1. KEY PERFORMANCE PARAMETERS

Parameter	Typical Value
Optical Format	1/4″
Pixel Size and Type	3.0 μm × 3.0 μm
Active Pixels	1280 (H) × 800 (V) (Entire Array)
NTSC Output	720 (H) × 487 (V)
PAL Output	720 (H) × 576 (V)
Input Clock Range	6−30 MHz
Frame Rate (Note 1)	60 fps at 720p
Color Filter Array	RGB Bayer
Shutter Type	Electronic Rolling Shutter
Output Interface	Analog Composite, up to 16-bit Parallel Digital Output
Output Data Formats	YUV422 8-bit,10-bit, and 10 to 12-bit Tone-mapped Bayer
Maximum Output Clock Frequency	Parallel clock up to 84 MHz
Supply Voltage	VDDIO: 2.8 V Nominal VDD: 1.8 V Nominal VAA: 2.8 V Nominal VDDA_DAC: 3.3 V Nominal
Power Consumption (Typical)	506 mW (Linear Mode NTSC) 527 mW (HDR Mode NTSC)
Package	8.5 mm × 8.5 mm 130-pin BGA
Temperature	Operating Temperature −40°C to 105°C

1. Maximum frame rates depend on output interface and data format configuration used.

Features

- 3.0 μm Pixel with ON Semiconductor DR-Pix Technology
- Superior Low-light Performance
- 45 fps at 1.0 MP, 60 fps at 720p
- Linear or High Dynamic Range Video
- Color Processing Optimized for HDR Video Operation
- Color and Gamma Correction
- Auto Exposure, Auto White Balance, 50/60 Hz Auto Flicker Detection and Avoidance
- Adaptive Local Tone Mapping (ALTM)
- Programmable Spatial Transform Engine (STE)
- Pre-rendered Graphical Overlay
- Two-wire Serial Programming Interface (CCIS)
- Interface to Low-cost Flash or EEPROM through SPI Bus (to Configure and Load Patches, etc.)
- High-level Host Command Interface
- Standalone Operation Supported
- Up to 5 GPIO
- Support for External LED or Xenon Flash
- Fail-safe IO
- Multi-camera Synchronization Support
- Integrated Video Encoder for NTSC/PAL with Overlay Capability and 10-bit I-DAC
- Temperature Sensor

Applications

- Surround, Rear and Front View Cameras
- Blind Spot/Side Mirror Replacement Cameras
- Automotive Viewing/Processing Fusion Cameras

© Semiconductor Components Industries, LLC, 2017

February, 2018 − Rev. 0

ORDERING INFORMATION

Table 2. ORDERABLE PART NUMBERS

Part Number	Description	Orderable Product Attribute Description
AS0140AT2C00XUSM0−DPBR	0° CRA, RGB iEBGA	Dry Pack with Protective Film, Double Side BBAR Glass, Engineering Sample
AS0140AT2C00XUSM0−DRBR	0° CRA, RGB iEBGA	Dry Pack w/o Protective Film, Double Side BBAR Glass, Engineering Sample
AS0140AT2C00XUSM0−TPBR	0° CRA, RGB iEBGA	Tape and Reel with Protective Film, Double Side BBAR Glass, Engineering Sample
AS0140AT2C00XUSM0−TRBR	0° CRA, RGB iEBGA	Tape and Reel w/o Protective Film, Double Side BBAR Glass, Engineering Sample
AS0140AT2C00XUSMH3−GEVB	RGB Headboard	
MARS1−AS0140AT2−GEVB	RGB MARS Board	

Function Overview

Figure 1 shows the typical configuration of the AS0140AT in a camera system. On the host side, a two-wire serial interface is used to control the operation of the AS0140AT, and image data is transferred using the analog or parallel interface between the AS0140AT and the host.

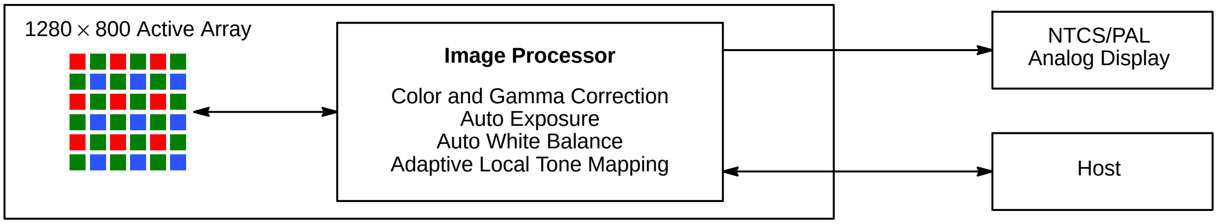

1280 × 800 Active Array
Image Processor
Color and Gamma Correction
Auto Exposure
Auto White Balance
Adaptive Local Tone Mapping
NTCS/PAL
Analog Display
Host

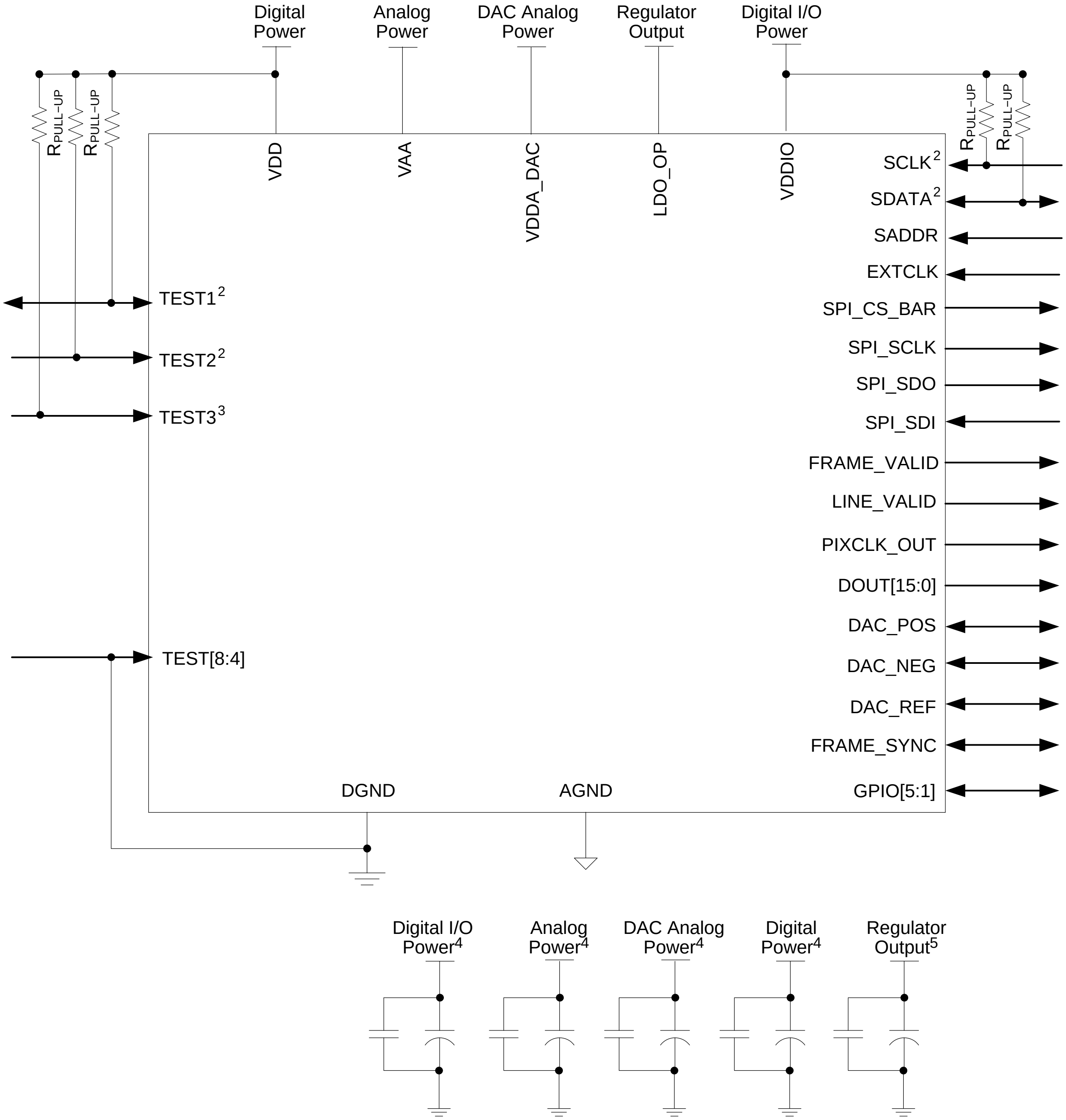
Figure 1. AS0140AT Connectivity

System Interfaces

Figure 2 shows typical AS0140AT device connections.

All power supply rails must be decoupled from ground using capacitors as close as possible to the package.

Table 3 provides pin descriptions for the AS0140AT.


Digital Power
Analog Power
DAC Analog Power
Regulator Output
Digital I/O Power
R PULL−UP
R PULL−UP
R PULL−UP
R PULL−UP
VDD
VAA
VDDA_DAC
LDO_OP
VDDIO
SCLK 2
SDATA 2
SADDR
EXTCLK
SPI_CS_BAR
SPI_SCLK
SPI_SDO
SPI_SDI
FRAME_VALID
LINE_VALID
PIXCLK_OUT
DOUT[15:0]
DAC_POS
DAC_NEG
DAC_REF
FRAME_SYNC
GPIO[5:1]
TEST1 2
TEST2 2
TEST3 3
TEST[8:4]
DGND
AGND
Digital I/O Power 4
Analog Power 4
DAC Analog Power 4
Digital Power 4
Regulator Output 5

Notes:

1. This typical configuration shows only one scenario out of multiple possible variations for this device.
2. ON Semiconductor recommends a 1.5 kΩ resistor value for the two-wire serial interface $R_{PULL\text{-}UP}$. However, greater values may be used for slower two−wire serial transmission speed.
3. ON Semiconductor recommends a 10 kΩ resistor value for TEST3 to avoid potential power−up issues.
4. ON Semiconductor recommends that 0.1 μF and 10 μF decoupling capacitors for each power supply are mounted as close as possible to the pin. Actual values and numbers may vary depending on layout and design consideration.
5. The decoupling capacitors for the regulator input and output should have a value of 1.0 μF. The capacitors should be ceramic and need to have X5R or X7R dielectric.

Figure 2. Typical Device Configuration

Table 3. PIN DESCRIPTION

Pin Number	Pin Name	Type	Description
CLOCK AND RESET			
L9	EXTCLK	Input	Master Input Clock
F1	STANDBY	Input	Standby Mode Control, Active HIGH
B10	RESET_BAR	Input	Master reset signal, active LOW. This signal has an internal pull up.
B8	FRAME_SYNC	Input	This signal is used to synchronize to external sources or multiple cameras together. This signal should be connected to GND if not used.
L12	TRIGGER_OUT	Output	If utilizing trigger modes, TRIGGER_OUT should be connected to the TRIGGER pin; otherwise, this signal should be left unconnected.
H11	TRIGGER	Input	If utilizing trigger modes, TRIGGER_OUT should be connected to the TRIGGER pin; otherwise, this signal should be connected to GND.
REGISTER INTERFACE			
B3	SCLK	Input	SCLK: Two-wire Serial Interface Clock (Host Interface)
C12	SDATA	I/O	Two-wire Serial Interface Data (Host Interface)
A7	SADDR	Input	Selects device address for the two-wire slave serial interface. When connected to GND the device ID is 0x90. When wired to VDDIO, a device ID of 0xBA is selected.
SPI INTERFACE			
J1	SPI_SCLK	Output	Clock Output for Interfacing to an External SPI Flash or EEPROM Memory
A8	SPI_SDI	Input	Data in from SPI flash or EEPROM memory. When no SPI device is fitted, this signal is used to determine whether the AS0140AT should auto-configure: 0: Do not auto-configure; Two-wire interface will be used to configure the device (host-config mode) 1: Auto-configure. This signal has an internal pullup resistor.
A4	SPI_SDO	Output	Data Out to SPI Flash or EEPROM Memory
A5	SPI_CS_BAR	Output	Chip Select Out to SPI Flash or EEPROM Memory
PIXEL DATA OUTPUT			
H1	FRAME_VALID	Output	Frame Valid Output (Synchronous to PIXCLK_OUT)
B4	LINE_VALID	Output	Line Valid Output (Synchronous to PIXCLK_OUT)
B5	PIXCLK_OUT	Output	Pixel Clock Output
E2	DOUT0	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
M10	DOUT1	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
L8	DOUT2	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
E12	DOUT3	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
L10	DOUT4	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
L3	DOUT5	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
D2	DOUT6	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
M9	DOUT7	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
D12	DOUT8	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
J2	DOUT9	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
H2	DOUT10	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
E1	DOUT11	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)

Table 3. PIN DESCRIPTION (continued)

Pin Number	Pin Name	Type	Description
PIXEL DATA OUTPUT			
C2	DOUT12	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
A10	DOUT13	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
G1	DOUT14	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
F12	DOUT15	Output	Pixel Data Output (Synchronous to PIXCLK_OUT)
COMPOSITE VIDEO OUTPUT			
M3	DAC_REF	Output	External Reference Resistor for Video DAC
L7	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.
L6	DAC_NEG	Output	Negative Video DAC Output in Differential Mode
GPIO			
G2	GPIO_1	I/O	General Purpose Digital I/O
C1	GPIO_2	I/O	General Purpose Digital I/O
D1	GPIO_3	I/O	General Purpose Digital I/O
F2	GPIO_4	I/O	General Purpose Digital I/O
A3	GPIO_5	I/O	General Purpose Digital I/O
POWER			
E10, F10, G10, H12, J12	AGND	Supply	Analog Ground
K2, D4, E4, F4, G4, H4, J4, L4, D5, E5, F5, G5, H5, J5, D6, E6, F6, G6, H6, J6, B7, D7, E7, F7, G7, H7, J7, D8, E8, F8, G8, H8, J8, B9, D9, E9, F9, G9, H9, J9, G12	DGND	Supply	Digital Ground
C10, D10, D11, E11	VDDIO	Supply	I/O Supply Power
H10, J10, K10, K12	VAA	Supply	Analog Power
E3, F3, G3, H3, J3, K3, M4, M5, M7	VDD	Supply	Digital Power
M6	VDDA_DAC	Supply	Video DAC Analog Power
A6	LDO_OP	Output	Output from on Chip 1.8 to 1.2 V Regulator
TEST PINS			
F11	TEST1	Input	Must be Pulled Up via 1.5 kΩ to V_{DD} for Normal Operation
G11	TEST2	Input	Must be Pulled Up via 1.5 kΩ to V_{DD} for Normal Operation
A12	TEST3	Input	Recommended Pull Up to V_{DD}
B6	TEST4	Input	Must be Tied to GND for Normal Operation
K11	TEST5	Input	Must be Tied to GND for Normal Operation
L2	TEST6	Input	Must be Tied to GND for Normal Operation
L11	TEST7	Input	Must be Tied to GND for Normal Operation
M8	TEST8	Input	Must be Tied to GND for Normal Operation
A1, B1, K1, L1, M1, A2, B2, M2, L5, A9, A11, B11, C11, J11, M11, B12, M12	NC		

Table 4. PACKAGE PINOUT

	1	2	3	4	5	6	7	8	9	10	11	12
A	NC	NC	GPIO_5	SPI_SDO	SPI_CS_BAR	LDO_OP	SADDR	SPI_SDI	NC	DOUT13	NC	TEST3
B	NC	NC	SCLK	LINE_VALID	PIXCLK_OUT	TEST4	DGND	FRAME_SYNC	DGND	RESET_BAR	NC	NC
C	GPIO_2	DOUT12								VDDIO	NC	SDATA
D	GPIO_3	DOUT6		DGND	DGND	DGND	DGND	DGND	DGND	VDDIO	VDDIO	DOUT8
E	DOUT11	DOUT0	VDD	DGND	DGND	DGND	DGND	DGND	DGND	AGND	VDDIO	DOUT3
F	STANDBY	GPIO_4	VDD	DGND	DGND	DGND	DGND	DGND	DGND	AGND	TEST1	DOUT15
G	DOUT14	GPIO_1	VDD	DGND	DGND	DGND	DGND	DGND	DGND	AGND	TEST2	DGND
H	FRAME_VALID	DOUT10	VDD	DGND	DGND	DGND	DGND	DGND	DGND	VAA	TRIGGER	AGND
J	SPI_SCLK	DOUT9	VDD	DGND	DGND	DGND	DGND	DGND	DGND	VAA	NC	AGND
K	NC	DGND	VDD							VAA	TEST5	VAA
L	NC	TEST6	DOUT5	DGND	NC	DAC_NEG	DAC_POS	DOUT2	EXTCLK	DOUT4	TEST7	TRIGGER_OUT
M	NC	NC	DAC_REF	VDD	VDD	VDDA_DAC	VDD	TEST8	DOUT7	DOUT1	NC	NC

On-Chip Regulator

The AS0140AT has an on-chip regulator, the output from the regulator is 1.2 V.

Power-Up Sequence

Powering up the AS0140AT requires voltages to be applied in a particular order, as seen in Figure 3. The timing requirements are shown in Table 5. The AS0140AT includes a power-on reset feature that initiates a reset upon power up of the AS0140AT.

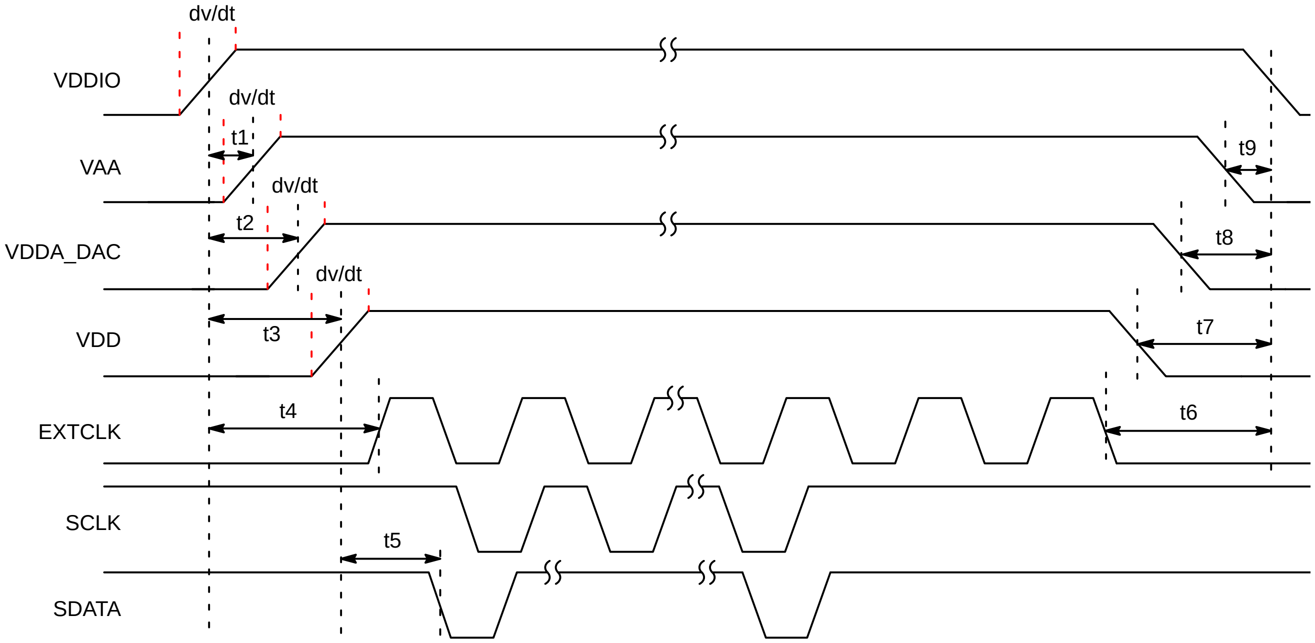


Figure 3. Power-Up and Power-Down Sequence

Table 5. POWER-UP AND POWER-DOWN SIGNAL TIMING

Symbol	Parameter	Min	Typ	Max	Unit
t1	Delay from VDDIO to VAA	0	−	50	ms
t2	Delay from VDDIO to VDDA_DAC	0	−	50	ms
t3	Delay from VDDIO to VDD	0	−	50	ms
t4	EXTCLK Activation	t3 + 1	−	−	ms
t5	First Serial Command	100	−	−	EXTCLK Cycles
t6	EXTCLK Cutoff	t7	−	−	ms
t7	Delay from VDD to VDDIO	0	−	50	ms
t8	Delay from VDDA_DAC to VDDIO	0	−	50	ms
t9	Delay from VAA to VDDIO	0	−	50	ms
dv/dt	Power Supply Ramp Time (Slew Rate)	−	−	0.1	V/μs

1. It is critical that VAA is not powered up after VDD. It must be powered before or at least at the same time. If the case happens that VAA is powered after VDD then sensor may have functionality issues and will experience high current draw on this supply.

Reset

The AS0140AT has three types of reset available:

- A hard reset is issued by toggling the RESET_BAR signal
- A soft reset is issued by writing commands through the two-wire serial interface
- An internal power-on reset

Table 6 shows the output states when the part is in various states.

Table 6. OUTPUT STATES

Name	Hardware States		Firmware States				Notes
	Reset State	Default State	Hard Standby	Soft Standby	Streaming	Idle	
EXTCLK	(Clock Running or Stopped)	(Clock Running)	(Clock Running or Stopped)	(Clock Running)	(Clock Running)	(Clock Running)	Input
RESET_BAR	(Asserted)	(Negated)	(Negated)	(Negated)	(Negated)	(Negated)	Input
SCLK	N/A	N/A	(Clock Running or Stopped)	(Clock Running or Stopped)	(Clock Running or Stopped)	(Clock Running or Stopped)	Input. Must always be driven to a valid logic level.
SDATA	High-impedance	High-impedance	High-impedance	High-impedance			Input/Output. A valid logic level should be established by pull-up.
SADDR	N/A	N/A	N/A	N/A	N/A	N/A	Input. Must always be driven to a valid logic level.
FRAME_SYNC	N/A	N/A	N/A	N/A	N/A	N/A	Input. Must always be driven to a valid logic level.
STANDBY	N/A	(Negated)	(Asserted)	(Negated)	(Negated)	(Negated)	Input. Must always be driven to a valid logic level.
SPI_SCLK	High-impedance	Driven, Logic 0	Driven, Logic 0	Driven, Logic 0			Output
SPI_SDI	Internal Pull-up Enabled	Internal Pull-up Enabled	Internal Pull-up Enabled	Internal Pull-up Enabled			Input. Internal pull-up permanently enabled.
SPI_SDO	High-impedance	Driven, Logic 0	Driven, Logic 0	Driven, Logic 0			Output
SPI_CS_BAR	High-impedance	Driven, Logic 1	Driven, Logic 1	Driven, Logic 1			Output
FV_OUT, LV_OUT, PIXCLK_OUT, DOUT[15:0]	High-impedance	Varied	Driven if Used	Driven if Used	Driven if Used	Driven if Used	Output. Default state dependent of configuration.
DAC_POS DAC_NEG	Varied	Varied	Driven if Used	Driven if Used	Driven if Used	Driven if Used	Output. Default state dependent on configuration. Tie to ground if VDAC not used.
DAC_REF	N/A	N/A	N/A	N/A	N/A	N/A	Output. Requires reference resistor. Tie to ground if VDAC not used.

Table 6. OUTPUT STATES (continued)

Name	Hardware States		Firmware States				Notes
	Reset State	Default State	Hard Standby	Soft Standby	Streaming	Idle	
GPIO[5:2]	High-impedance	Input, then High-impedance	Driven if Used	Driven if Used	Driven if Used	Driven if Used	Input/Output. After reset, these pins are sampled as inputs as part of auto-configuration.
GPIO1	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	
TRIGGER_OUT	N/A	N/A	N/A	N/A	N/A	N/A	Output. Tie to TRRIGER if used; otherwise leave NC.
TRIGGER	N/A	N/A	N/A	N/A	N/A	N/A	Input. Tie to TRRIGER_OUT if used; otherwise tie to ground.
TEST[3:1]	N/A	N/A	N/A	N/A	N/A	N/A	Input. A valid logic level should be established by pull-up.
TEST[8:4]	N/A	N/A	N/A	N/A	N/A	N/A	Input. Must always be driven to GND.

Hard Reset

The AS0140AT enters reset state when the external RESET_BAR is asserted LOW, as shown in Figure 4. All the output signals will be in High-Z state.

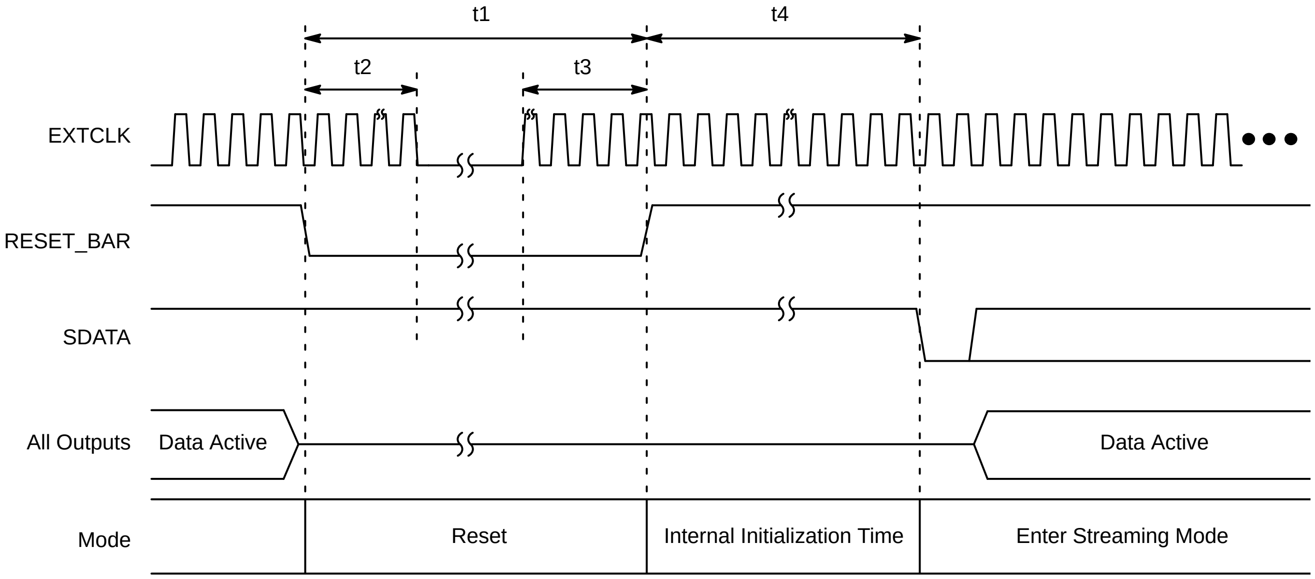


Figure 4. Hard Reset Operation

Table 7. HARD RESET

Symbol	Parameter	Min	Typ	Max	Unit
t1	RESET_BAR Pulse Width	50	−	−	EXTCLK Cycles
t2	Active EXTCLK Required after RESET_BAR Asserted	10	−	−	EXTCLK Cycles
t3	Active EXTCLK Required before RESET_BAR De-asserted	10	−	−	EXTCLK Cycles
t4	First Two-wire Serial Interface Communication after RESET is HIGH	100	−	−	EXTCLK Cycles

Soft Reset

A soft reset sequence to the AS0140AT can be activated by writing to a register through a two-wire serial interface.

Hard Standby Mode

The AS0140AT can enter hard standby mode by using external STANDBY signal, as shown in Figure 5.

Entering Standby Mode

- Assert STANDBY signal HIGH.

Exiting Standby Mode

- De-assert STANDBY signal LOW.

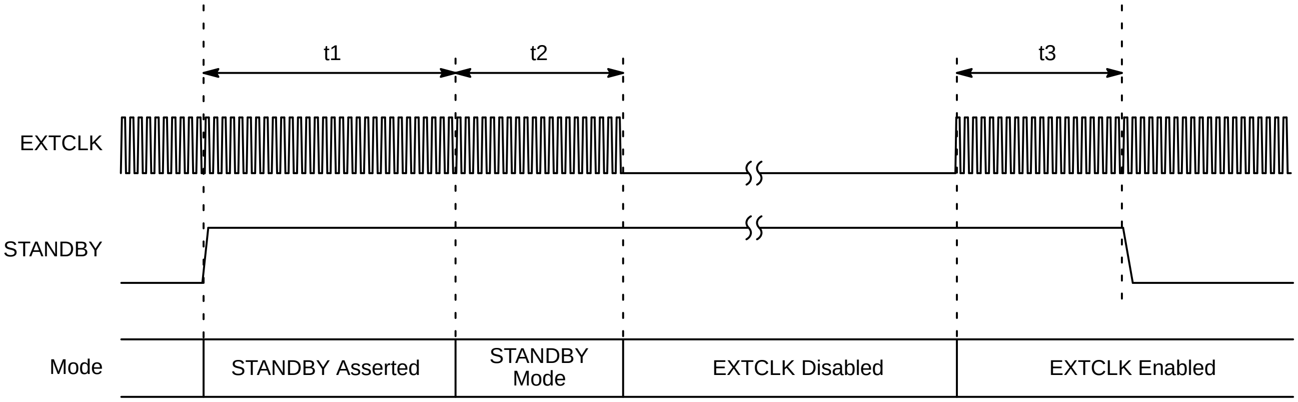


Figure 5. Hard Standby Operation

Table 8. HARD STANDBY SIGNAL TIMING

Symbol	Parameter	Min	Typ	Max	Unit
t1	Standby Entry Complete	−	−	2 Frames	Lines
t2	Active EXTCLK Required after Going into STANDBY Mode	10	−	−	EXTCLKs
t3	Active EXTCLK Required before STANDBY De-asserted	10	−	−	EXTCLKs

Multi-Camera Synchronization Support

The AS0140AT supports multi-camera synchronization through the FRAME_SYNC pin.

The behavior will be different depending if the user is using interlaced or progressive mode.

When using the interlaced modes, on the rising edge of FRAME_SYNC this will cause the output to stop the current frame (A) and during B the image output will be indeterminate. On the falling edge of FRAME_SYNC this will cause the re-synchronization to begin, this will continue for a period (C), during C black fields will be output. The resynchronized interlaced signal will be available at D. During C if the user toggles the FRAME_SYNC input the AS0140AT will ignore it, the user cannot re-synchronize again until at D.

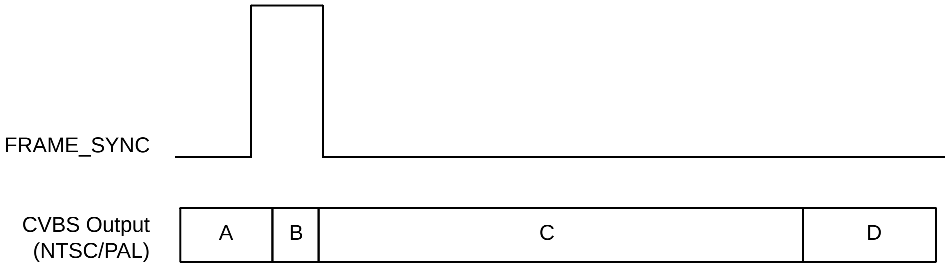


Figure 6. Frame Sync Behavior with Interlaced Mode

When using progressive mode, the host (or controlling entity) 'broadcasts' a sync-pulse to all cameras within the system that triggers capture. The AS0140AT will propagate the signal to the TRIGGER_OUT pin, and subsequently to the attached sensor's TRIGGER pin.

The AS0140AT supports two different trigger modes when using progressive output. The first mode supported is 'single-shot'; this is when the trigger pulse will cause one frame to be output from the AS0140AT (see Figure 7).

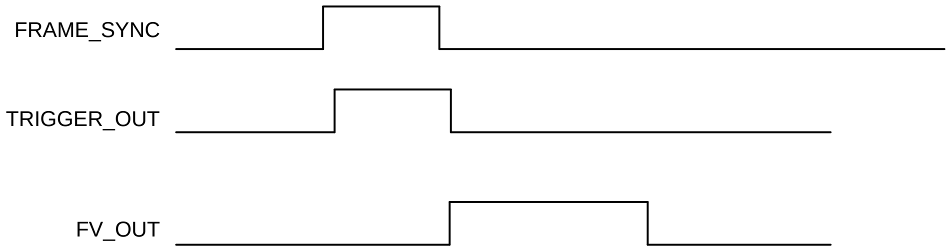
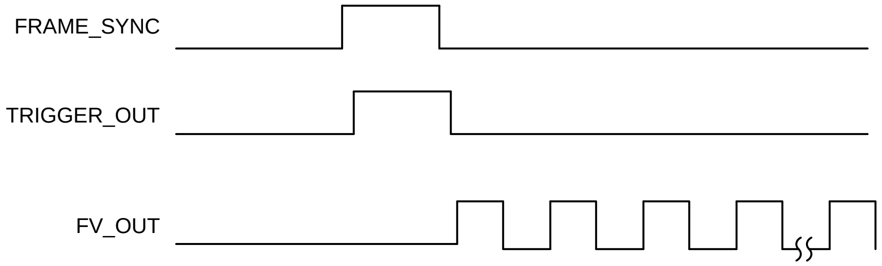


Figure 7. Single-Shot Mode

The second mode supported is called 'continuous', this is when a trigger pulse will cause the part to continuously output frames, see Figure 8. This mode would be especially useful for applications which have multiple sensors and need to have their video streams synchronized (for example, surround view or panoramic view applications).



NOTE: This diagram is not to scale.

Figure 8. Continuous Mode

When two or more cameras have a signal applied to the FRAME_SYNC input at the same time, the respective FRAME_VALID signals would be synchronized within 5 PIXCLK_OUT cycles. This assumes that all cameras have the same configuration settings and that the exposure time is the same.

Image Flow Processor

Image and color processing in the AS0140AT is implemented as an image flow processor (IFP) coded in hardware logic. During normal operation, the embedded microcontroller will automatically adjust the operating parameters. For normal operation of the AS0140AT, streams of raw image data are fed into the color pipeline. The user also has the option to select a number of test patterns to be input instead of sensor data. The IFP is broken down into different sections, as outlined in Figure 9.

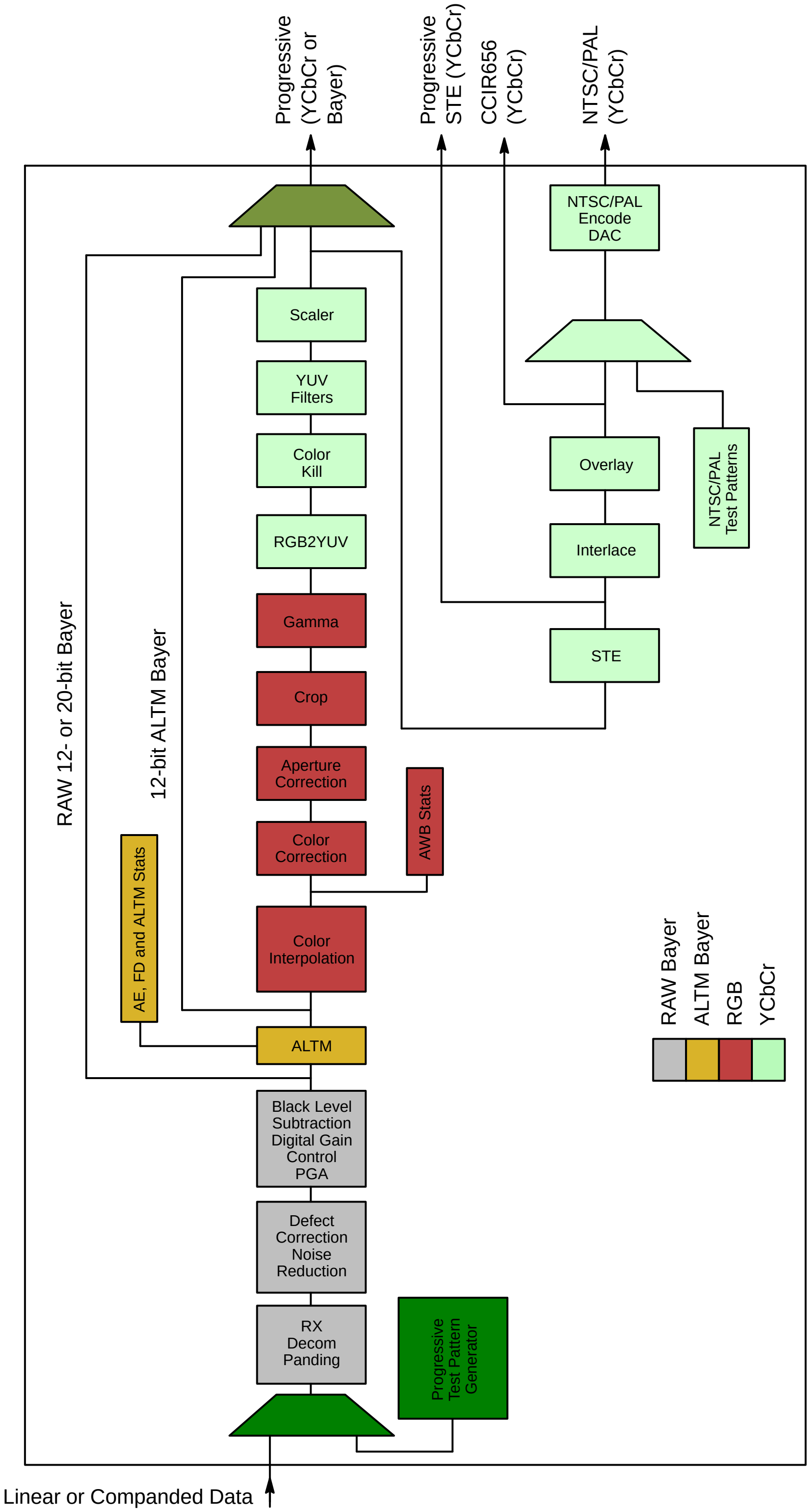

Progressive (YCbCr or Bayer)
Progressive STE (YCbCr)
CCIR656 (YCbCr)
NTSC/PAL (YCbCr)
NTSC/PAL Encode DAC
Scaler
YUV Filters
Color Kill
RGB2YUV
Gamma
Crop
Aperture Correction
Color Correction
AWB Stats
Color Interpolation
ALTM
AE, FD and ALTM Stats
Black Level Subtraction Digital Gain Control PGA
Defect Correction Noise Reduction
RX Decom Panding
Progressive Test Pattern Generator
Overlay
Interlace
STE
NTSC/PAL Test Patterns
RAW 12- or 20-bit Bayer
12-bit ALTM Bayer
RAW Bayer
ALTM Bayer
RGB
YCbCr
Linear or Companded Data

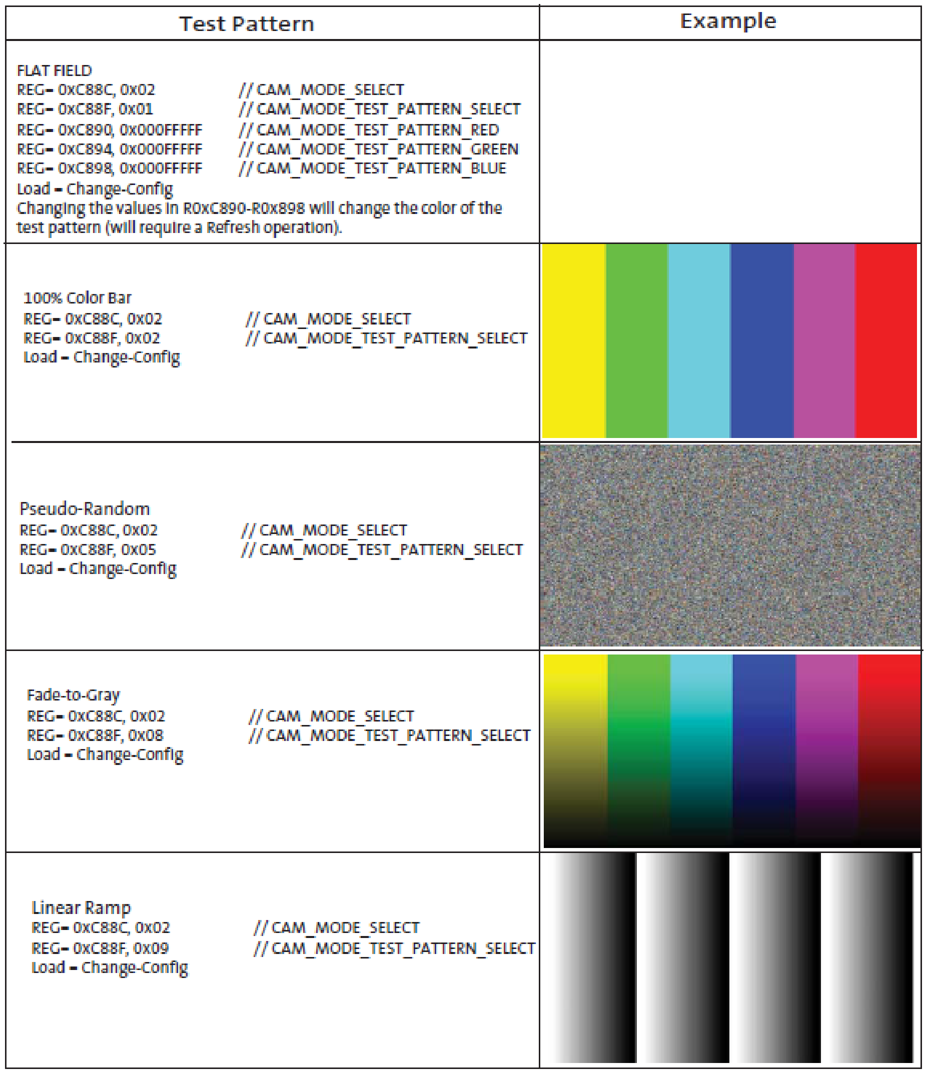
Figure 9. AS0140AT IFP

Test Patterns

The AS0140AT has a number of test patterns that are available when using the progressive NTSC and PAL modes. The test patterns can be selected by programming variables. To enter test pattern mode, set R0xC88F to 0x02 and issue a Change-Config request; to exit this mode, set R0xC88F to 0x00, and issue a Change-Config request.

NTSC and PAL test patterns can only be selected when the device is configured for interlaced operation.

Progressive Test Patterns:



Test Pattern	Example

FLAT FIELD
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x01 // CAM_MODE_TEST_PATTERN_SELECT
REG= 0xC890, 0x000FFFFF // CAM_MODE_TEST_PATTERN_RED
REG= 0xC894, 0x000FFFFF // CAM_MODE_TEST_PATTERN_GREEN
REG= 0xC898, 0x000FFFFF // CAM_MODE_TEST_PATTERN_BLUE
Load = Change-Config
Changing the values in R0xC890-R0x898 will change the color of the test pattern (will require a Refresh operation).	

100% Color Bar
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x02 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config	

Pseudo-Random
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x05 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config	

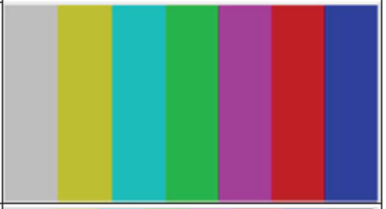
Fade-to-Gray
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x08 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config	

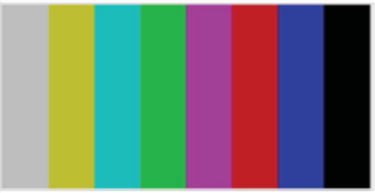
Linear Ramp
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x09 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config	

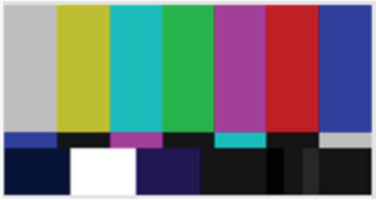
Figure 10. Progressive Test Patterns

NTSC Test Patterns:

Test Pattern	Example

EIA Full Field 7 Color Bars
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x14 // CAM_MODE_TEST_PATTTERN_SELECT
Load = Change-Config
	

EIA Full Field 8 Color Bars
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x15 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config
	

SMPTE EG 1-1990
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x16 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config
	

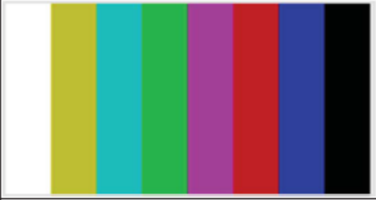
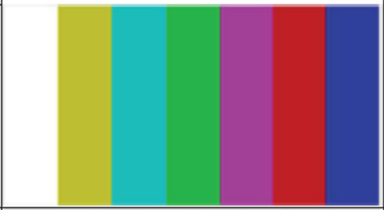
EIA Full Field 8 Color Bars 100 IRE
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x17 // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config
	

Figure 11. NTSC Test Patterns

PAL Test Patterns:

Test Pattern	Example

EBU Full Field 7 Color Bars
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x1E // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config
	

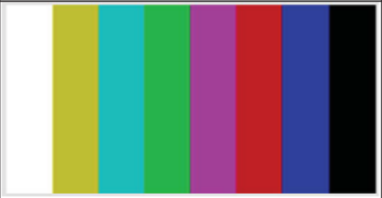
EBU Full Field 8 Color Bars
REG= 0xC88C, 0x02 // CAM_MODE_SELECT
REG= 0xC88F, 0x1F // CAM_MODE_TEST_PATTERN_SELECT
Load = Change-Config
	

Figure 12. PAL Test Patterns

Each NTSC/PAL test pattern consists of seven or eight color bars (white, yellow, cyan, green, magenta, red, blue and optionally black). The Y, Cb and Cr values for each bar are detailed in Table 9.

For the NTSC SMPTE test pattern it is also required to generate −I, +Q, −4 black and +4 black.

Table 9. NTSC/PAL TEST PATTERN VALUES

	Nominal Range	White 100%	White 75%	Yellow	Cyan	Green	Magenta	Red	Blue	Black	−I	−Q	−4 Black	+4 Black
Y	16 to 235	235	180	162	131	112	84	65	35	16	16	16	7	25
Cb	16 to 240	128	128	44	156	72	184	100	212	128	156	171	128	128
Cr	16 to 240	128	128	142	44	58	198	212	114	128	97	148	128	128

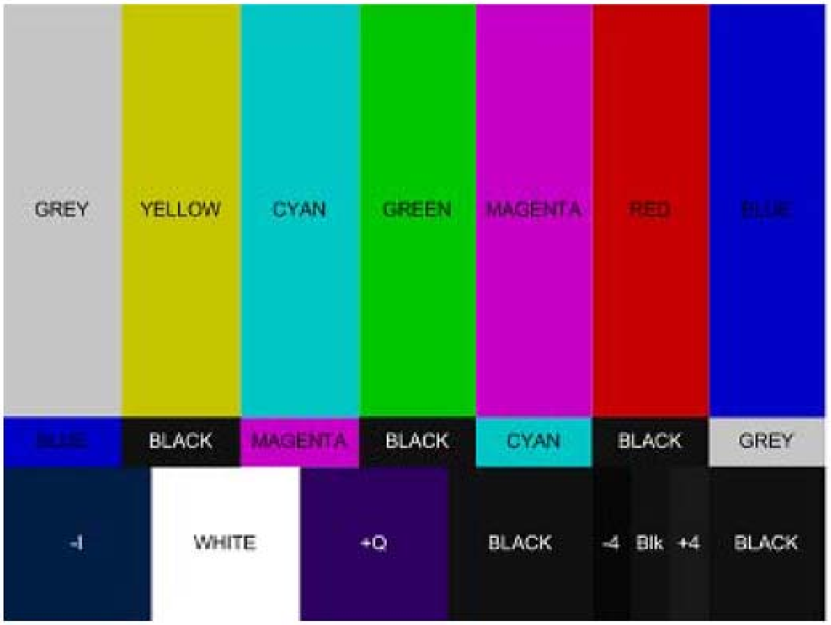


Figure 13. Test Pattern

Defect Correction

Image stream processing commences with the defect correction function immediately after data decompanding.

To obtain defect free images, the pixels marked defective during sensor readout and the pixels determined defective by the defect correction algorithms are replaced with values derived from the non-defective neighboring pixels. This image processing technique is called defect correction.

AdaCD (Adaptive Color Difference)

Automotive applications require good performance in extremely low light, even at high temperature conditions. In these stringent conditions the image sensor is prone to higher noise levels, and so efficient noise reduction techniques are required to circumvent this sensor limitation and deliver a high quality image to the user.

Black Level Subtraction and Digital Gain

After noise reduction, the pixel data goes through black level subtraction and multiplication of all pixel values by a programmable digital gain. Independent color channel digital gain can be adjusted with registers. Black level subtraction (to compensate for sensor data pedestal) is a single value applied to all color channels. 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

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 AS0140AT 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:

$$P_{\text{corrected}}(\text{row}, \text{col}) = P_{\text{sensor}}(\text{row}, \text{col}) \cdot f(\text{row}, \text{col}) \qquad \text{(eq. 1)}$$

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

Adaptive Local Tone Mapping

Real world scenes often have very high dynamic range (HDR) that far exceeds the electrical dynamic range of the imager. Dynamic range is defined as the luminance ratio between the brightest and the darkest object in a scene. In recent years many technologies have been developed to capture the full dynamic range of real world scenes. For example, the multiple exposure method is widely adopted for capturing high dynamic range images, which combines a series of low dynamic range images of the same scene taken under different exposure times into a single HDR image.

Even though the new digital imaging technology enables the capture of the full dynamic range, low dynamic range display devices are the limiting factor. Today's typical LCD monitor has contrast ratio around 1,000:1; however, it is not typical for an HDR image (the contrast ratio for an HDR image is around 250,000:1). Therefore, in order to reproduce HDR images on a low dynamic range display device, the captured high dynamic range must be compressed to the available range of the display device. This is commonly called tone mapping.

Tone mapping methods can be classified into global tone mapping and local tone mapping. Global tone mapping methods apply the same mapping function to all pixels. While global tone mapping methods provide computationally simple and easy to use solutions, they often cause loss of contrast and detail. A local tone mapping is thus necessary in addition to global tone mapping for the reproduction of visually more appealing images that also reveal scene details that are important for automotive safety and surveillance applications. Local tone mapping methods use a spatially variable mapping function determined by the neighborhood of a pixel, which allows it to increase the local contrast and the visibility of some details of the image. Local methods usually yield more pleasing results because they exploit the fact that human vision is more sensitive to local contrast.

ON Semiconductor's ALTM solution significantly improves the performance over global tone mapping. ALTM is directly applied to the Bayer domain to compress the dynamic range from 20-bit to 12-bit. This allows the regular color pipeline to be used for HDR image rendering.

Color Interpolation

In the raw data stream fed by the external sensor to the IFP, each pixel is represented by a 20- or 12-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 ALTM, preserve the one-color-per-pixel nature of the data stream, but after ALTM 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 3×3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. 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.

Traditionally this would have been derived from two sets of CCM, one for Warm light like Tungsten and the other for Daylight (the part would interpolate between the two matrices). This is not an optimal solution for cameras used in a Cool White Fluorescent (CWF) environment. A better solution is to provide three CCMs, which would include a matrix for CWF (interpolation now between three matrices). The AS0140AT offers this feature which will give the user improved color fidelity when under CWF type lighting.

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 gamma correction curve is implemented as a piecewise linear function with 33 knee points, taking 12-bit arguments and mapping them to 10-bit output. The abscissas of the knee points are fixed at 0, 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, 384, 448, 512, 640, 768, 896, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584, and 4096. The 10-bit ordinates are programmable through variables.

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

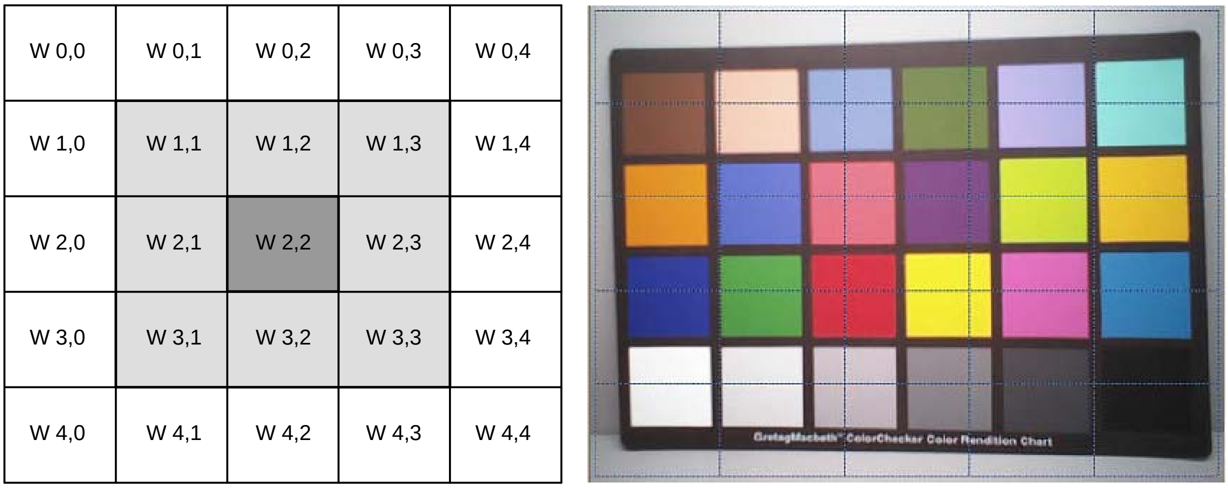
As an optional processing step, noise suppression by one-dimensional low-pass filtering of Y and/or UV signals is possible. A 3- or 5-tap filter can be selected for each signal.

Camera Control and Auto Functions

Auto Exposure

The auto exposure algorithm optimizes scene exposure to minimize clipping and saturation in critical areas of the image. This is achieved by controlling exposure time and analog gains of the external sensor as well as digital gains applied to the image.

Auto exposure is implemented by a firmware algorithm that is running on the embedded microcontroller that analyzes image statistics collected by the exposure measurement engine, makes a decision, and programs the sensor and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into 25 windows organized as a 5 × 5 grid.



W 0,0	W 0,1	W 0,2	W 0,3	W 0,4
W 1,0	W 1,1	W 1,2	W 1,3	W 1,4
W 2,0	W 2,1	W 2,2	W 2,3	W 2,4
W 3,0	W 3,1	W 3,2	W 3,3	W 3,4
W 4,0	W 4,1	W 4,2	W 4,3	W 4,4

Figure 14. 5 × 5 Grid

AE Track Driver

Other algorithm features include the rejection of fast fluctuations in illumination (time averaging), control of speed of response, and control of the sensitivity to small changes. While the default settings are adequate in most situations, the user can program target brightness, measurement window, and other parameters described above.

The driver changes AE parameters (integration time, gains, and so on) to drive scene brightness to the programmable target.

To avoid unwanted reaction of AE on small fluctuations of scene brightness or momentary scene changes, the AE track driver uses a temporal filter for luma and a threshold around the AE luma target. The driver changes AE parameters only if the filtered luma is larger than the AE target step and pushes the luma beyond the threshold.

Auto White Balance

The AS0140AT has a built-in AWB algorithm designed to compensate for the effects of changing spectra of the scene illumination on the quality of the color rendition. The algorithm consists of two major parts: a measurement engine performing statistical analysis of the image and a driver performing the selection of the optimal color correction matrix and IFP digital gain. While default settings of these algorithms are adequate in most situations, the user can reprogram base color correction matrices, place limits on color channel gains, and control the speed of both matrix and gain adjustments. The AS0140AT ATAWB displays the current AWB position in color temperature, the range of which will be defined when programming the CCM matrixes.

The region of interest can be controlled through the combination of an inclusion window and an exclusion window.

Exposure and White Balance Control

The Sensor Manager firmware component is responsible for controlling the application of ‘exposure’ and ‘white balance’ within the system. This effectively means that all control of integration times and gains (whether for exposure or white balance) is delegated to the Sensor Manager. The Auto Exposure (AE) and Auto White Balance (AWB) algorithms use services provided by the Sensor Manager to apply exposure and/or white balance changes.

Dual Band IRCF

For some applications a day/night filter would be switched in/out, this option is an additional cost to the camera system. The AS0140AT supports the use of dual band IRCF, which removes the need for the switching day/night filter. Tuning support is provided for this usage case. Refer to the AS0140AT developer guide for details.

Exposure and White Balance Modes

The AS0140AT supports auto and manual exposure and white balance modes. In addition, it will operate within synchronized multi-camera systems. In this use case, one camera within the system will be the ‘master’, and the others ‘slaves’. The master is used to calculate the appropriate exposure and white balance. This is then applied to all slaves concurrently under host control.

Auto Mode

In Auto Exposure mode the AE algorithm is responsible for calculating the appropriate exposure to keep the desired scene brightness, and for applying the exposure to the underlying hardware. In Auto White Balance mode the AWB algorithm is responsible for calculating the color temperature of the scene and applying the appropriate red and blue gains to compensate.

Triggered Auto Mode

The Triggered Auto Exposure and Triggered Auto White Balance modes are intended for the multi-camera use cases, where a host is controlling the exposure and white balance of a number of cameras. The idea is that one camera is in triggered-auto mode (the master), and the others in host-controlled mode (slaves). The master camera must calculate the exposure and gains, the host then copies this to the slaves, and all changes are then applied at the same time.

Manual Mode

Manual mode is intended to allow simple manual exposure and white balance control by the host. The host needs to set the CAM_AET_EXPOSURE_TIME_MS, CAM_AET_EXPOSURE_GAIN and CAM_AWB_COLOR_TEMPERATURE controls, the camera will calculate the appropriate integration times and gains.

Host Controlled

The Host Controlled mode is intended to give the host full control over exposure and gains.

Flicker Avoidance

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The AS0140AT can be programmed to avoid flicker for 50 or 60 Hz. For integration times below the light intensity period (10 ms for 50 Hz environment), flicker cannot be avoided. The AS0140AT supports an indoor AE mode, that will ensure flicker-free operation.

Flicker Detection

The AS0140AT supports flicker detection, the algorithm is designed only to detect a 50 Hz or 60 Hz flicker source.

Output Formatting

The pixel output data in AS0140AT will be transmitted as an 8/10 bit word over one or two clocks.

Uncompressed YCbCr Data Ordering

The AS0140AT supports swapping YCbCr mode, as illustrated in Table 10.

Table 10. YCbCr OUTPUT DATA ORDERING

Mode	Data Sequence			
Default (No Swap)	Cb_i	Y_i	Cr_i	Y_{i+1}
Swapped CrCb	Cr_i	Y_i	Cb_i	Y_{i+1}
Swapped YC	Y_i	Cbi	Y_{i+1}	Cr_i
Swapped CrCb, YC	Y_i	Cri	Y_{i+1}	Cb_i

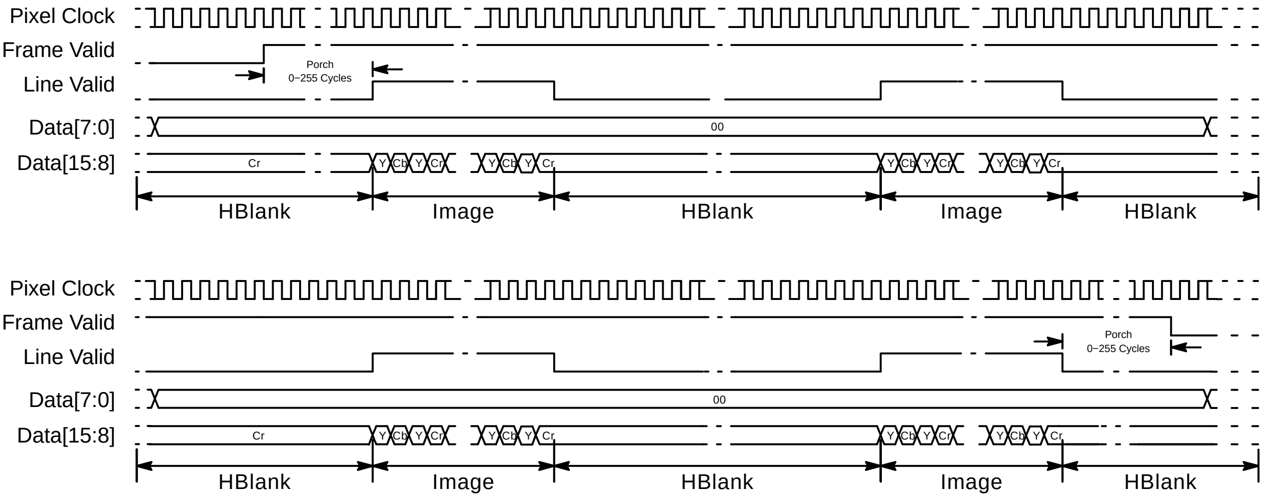
The data ordering for the YCbCr output modes for AS0140AT are shown in Table 11.

Table 11. YCbCr OUTPUT MODES (cam_port_parallel_msb_align=0x1)

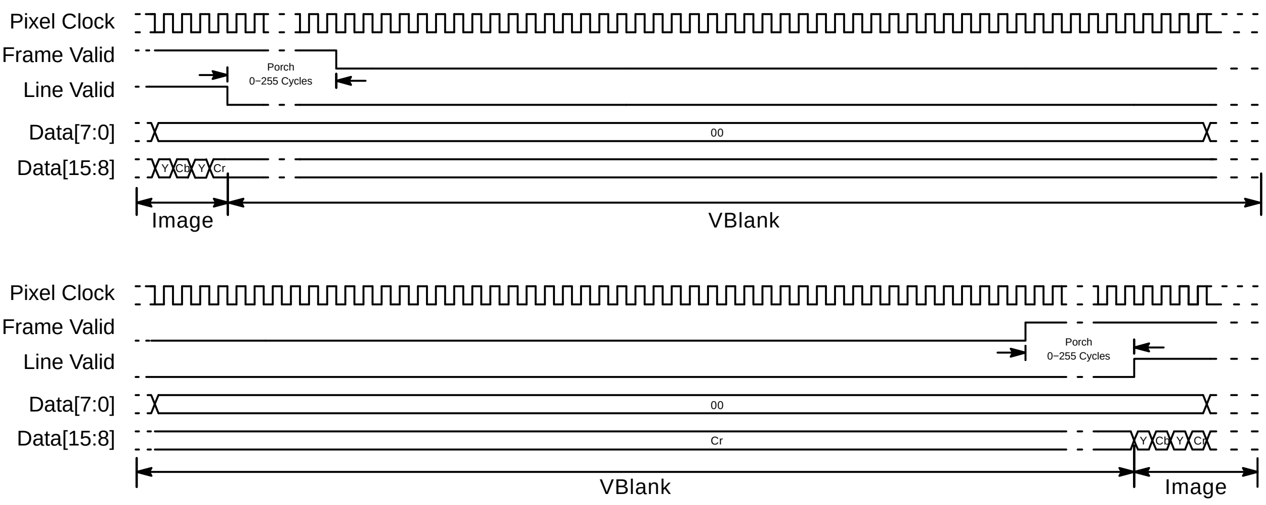
Mode	Byte	Pixel i	Pixel i+1	Notes
YCbCr_422_8_8	Odd (DOUT[15:8])	Cb_i	Cr_i	Data Range of 0−255 (Y = 16−235 and C = 16−240)
	Even (DOUT[15:8])	Y_i	Y_{i+1}	
YCbCr_422_10_10	Odd (DOUT[15:6])	Cb_i	Cr_i	Data Range of 0−1023 (Y = 64−940 and C = 64−960)
	Even (DOUT[15:6])	Y_i	Y_{i+1}	
YCbCr_422_16	Single (DOUT[15:0])	Cb_i_Y_i	Cr_i_Y_{i+1}	Data Range of 0−255 (Y = 16−235 and C = 16−240)

Table 12. YCbCr OUTPUT MODES (cam_port_parallel_msb_align=0x0)

Mode	Byte	Pixel i	Pixel i+1	Notes
YCbCr_422_8_8	Odd (DOUT[7:0])	Cb_i	Cr_i	Data Range of 0−255 (Y = 16−235 and C = 16−240)
	Even (DOUT[7:0])	Y_i	Y_{i+1}	
YCbCr_422_10_10	Odd (DOUT[9:0])	Cb_i	Cr_i	Data Range of 0−1023 (Y = 64−940 and C = 64−960)
	Even (DOUT[9:0])	Y_i	Y_{i+1}	
YCbCr_422_16	Single (DOUT[15:0])	Cb_i_Y_i	Cr_i_Y_{i+1}	Data Range of 0−255 (Y = 16−235 and C = 16−240)


Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0−255 Cycles
00
Cr
Y
Cb
Y
Cr
HBlank
Image
HBlank
Image
HBlank

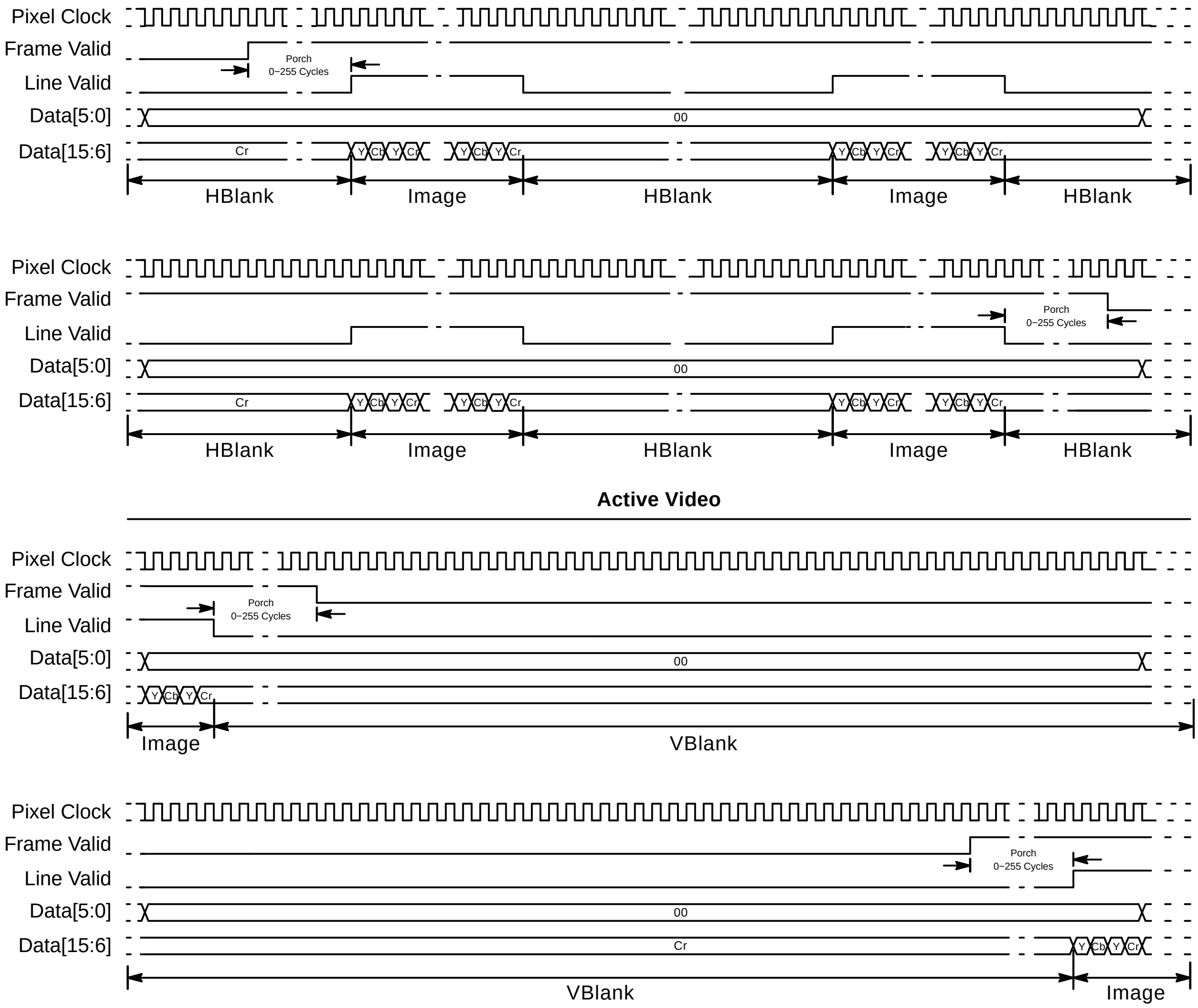
Active Video


Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0−255 Cycles
00
Cr
Y
Cb
Y
Cr
Image
VBlank
VBlank
Image

Notes:

1. Cb Y Cr Y by default.
2. cam_port_parallel_msb_align=0x1

Figure 15. 8-bit YCrCr Output (YCbCr_422_8_8)


Pixel Clock
Frame Valid
Line Valid
Data[5:0]
Data[15:6]
Porch
0−255 Cycles
00
Cr
Y
Cb
HBlank
Image
Active Video
VBlank

Notes:

1. Cb Y Cr Y by default.
2. cam_port_parallel_msb_align=0x1

Figure 16. 10-bit YCrCr Output (YCbCr_422_10_10)

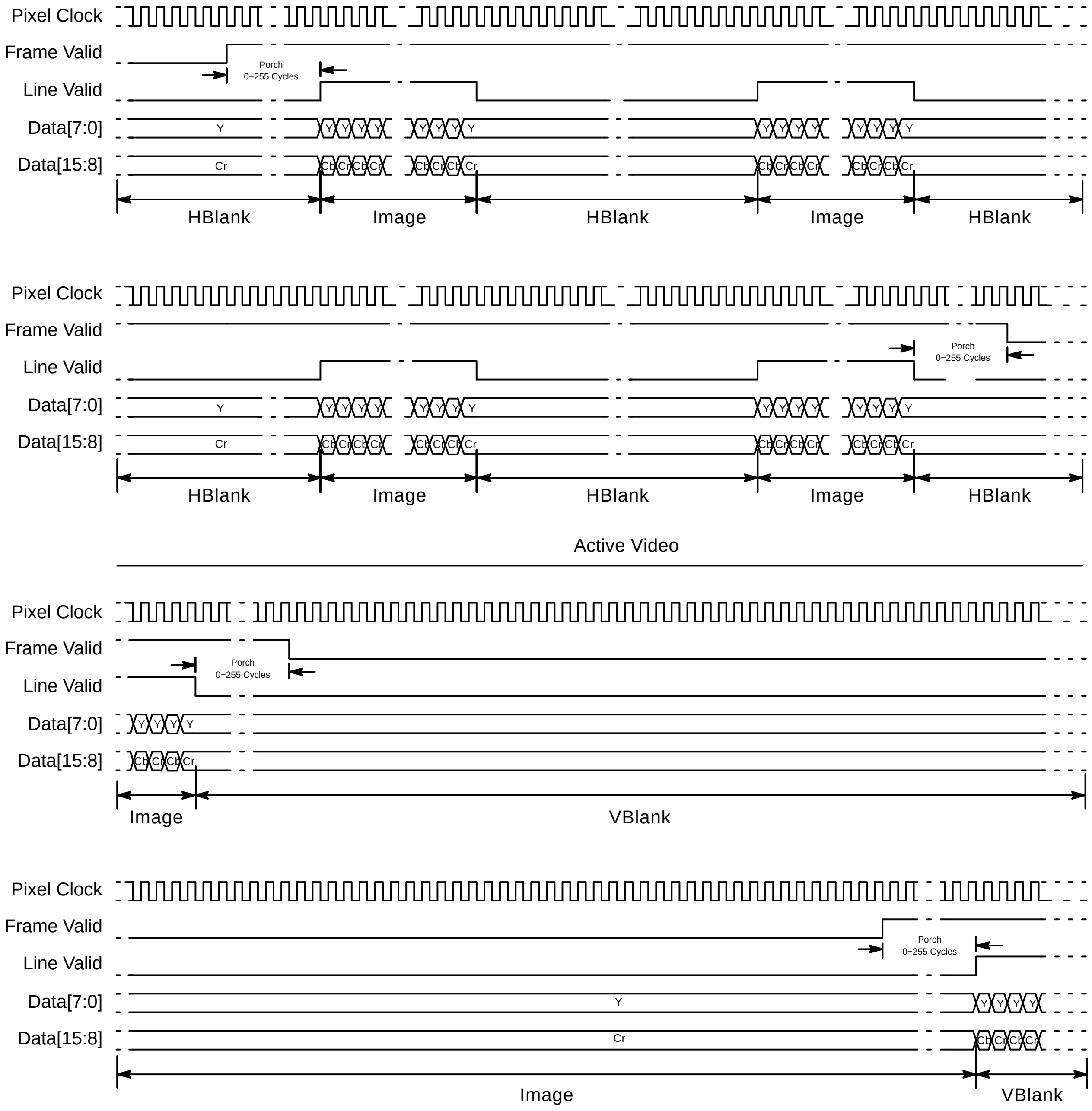

Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0–255 Cycles
Y
Cr
Cb
HBlank
Image
HBlank
Image
HBlank
Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0–255 Cycles
HBlank
Image
HBlank
Image
HBlank
Active Video
Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0–255 Cycles
Image
VBlank
Pixel Clock
Frame Valid
Line Valid
Data[7:0]
Data[15:8]
Porch
0–255 Cycles
Image
VBlank

Figure 17. 16-bit YCrCr Output (YCbCr_422_16)

Progressive STE can output any of the YCbCr modes in Figure 15 to Figure 17.

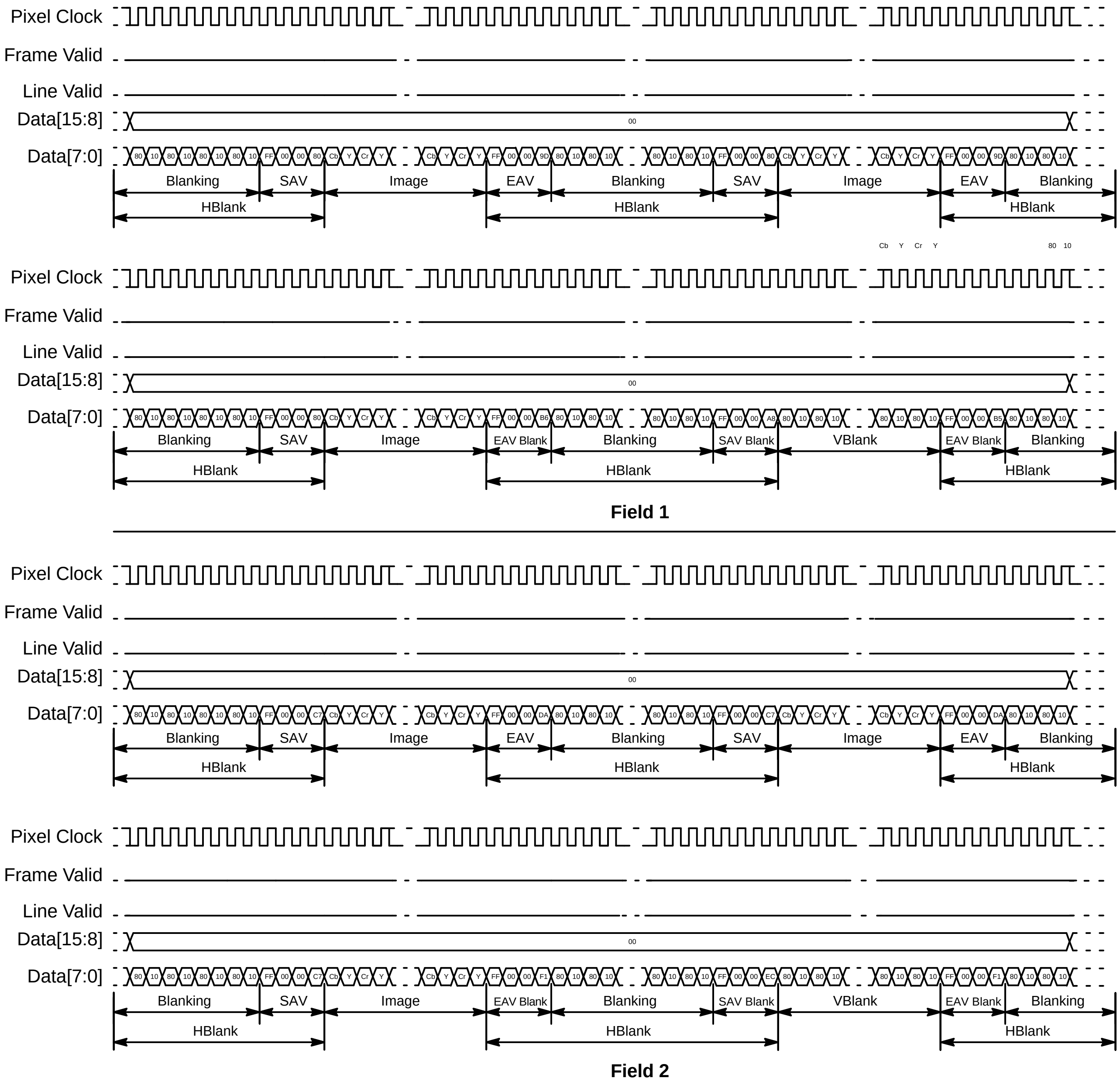

Pixel Clock
Frame Valid
Line Valid
Data[15:8]
00
Data[7:0]
80 10 80 10 80 10 80 10 FF 00 00 80 Cb Y Cr Y
Cb Y Cr Y FF 00 00 9D 80 10 80 10
80 10 80 10 FF 00 00 80 Cb Y Cr Y
Cb Y Cr Y FF 00 00 9D 80 10 80 10
Blanking
SAV
Image
EAV
HBlank
Cb Y Cr Y
80 10
EAV Blank
SAV Blank
VBlank
Field 1
Field 2

Figure 18. Typical CCIR656 Output

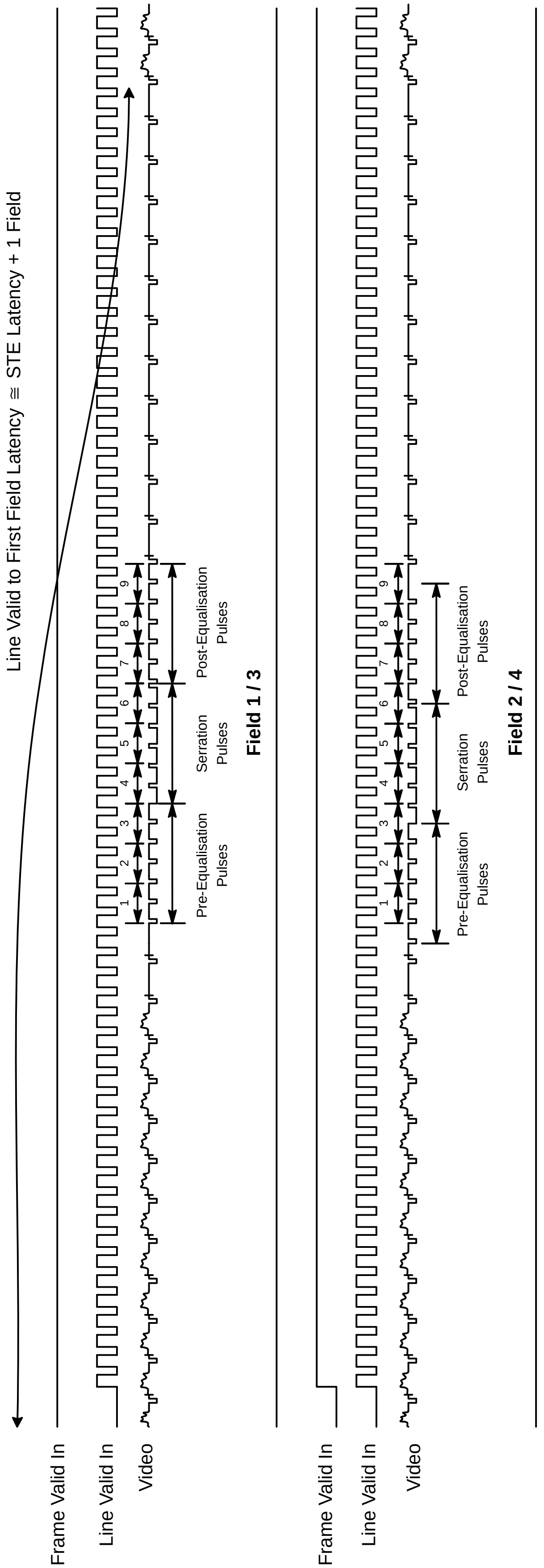

Line Valid to First Field Latency ≅ STE Latency + 1 Field
Frame Valid In
Line Valid In
Video
1
2
3
4
5
6
7
8
9
Pre-Equalisation Pulses
Serration Pulses
Post-Equalisation Pulses
Field 1 / 3
Frame Valid In
Line Valid In
Video
1
2
3
4
5
6
7
8
9
Pre-Equalisation Pulses
Serration Pulses
Post-Equalisation Pulses
Field 2 / 4

Figure 19. Typical CVBS Output (NTSC/PAL)

Bayer Modes

Bayer output modes are only available in progressive output mode before STE. The data ordering for the ALTM Bayer output modes for AS0140AT are shown in Table 13.

Table 13. ALTM BAYER OUTPUT MODES

Mode	Byte	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ALTM_Bayer_10	Single	0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ALTM_Bayer_12	Single	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Table 13 and Table 14 show LSB aligned data; it is possible using register setting to obtain MSB aligned data.

The data ordering for the Bayer output modes for AS0140AT are shown in Table 14.

Table 14. BAYER OUTPUT MODES

Mode	Byte	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Notes
Bayer_12	Single	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	RAW Bayer Data

Sensor Embedded Data

The AS0140AT is capable of passing sensor embedded data in Bayer output mode only.

The AS0140AT Statistics are available through the serial interface. Refer to the developer guide for details.

Spatial Transform Engine (STE)

A spatial transform is defined as a transform in which some pixels are in different positions within the input and output pictures. Examples include zoom, lens distortion correction, turn, rotate, roaming and projection. STE is a fully programmable engine which can perform spatial transforms and eliminates the need for an expensive DSP for image correction.

Lens Distortion Correction

Automotive backup cameras typically feature a wide FOV lens so that a single camera mounted above the center of the rear bumper can present the driver with a view of all potential obstacles immediately behind the full width of the vehicle. Lenses with a wide field of view typically exhibit at least a noticeable amount of barrel distortion.

Barrel distortion is caused by a reduction in object magnification the further away from the optical axis.

For the image to appear natural to the driver, the AS0140AT corrects this barrel distortion and reprocesses the image so that the resulting distortion is much smaller. This is called distortion correction. Distortion correction is the ability to digitally correct the lens barrel distortion and to provide a natural view of objects. In addition, with barrel distortion one can adjust the perspective view to enhance the visibility by virtually elevating the point of viewing objects.

Perspective View

A backup camera has to be able to virtually adjust the vertical perspective as if the camera were placed immediately behind the vehicle pointed directly down, as illustrated in Figure 20. The vertical perspective adjustment may be employed temporarily to assist with parking conditions, or it may be enabled permanently by loading new parameters.

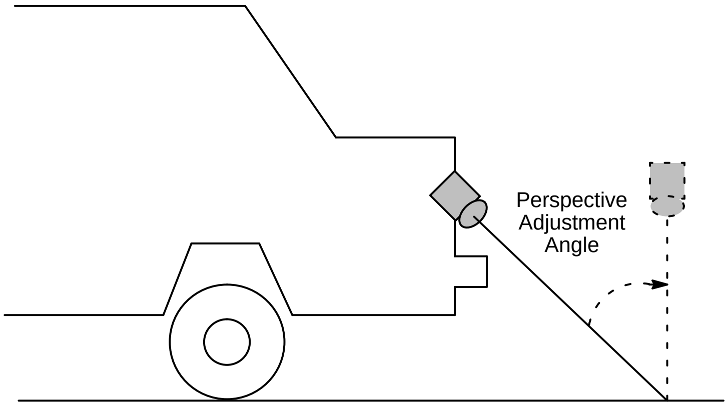


Figure 20. Vertical Perspective Adjustment

Pan, Tilt, Zoom and Rotate

Using the STE it is possible to implement image transforms like Pan, Tilt, Zoom, and Rotate.

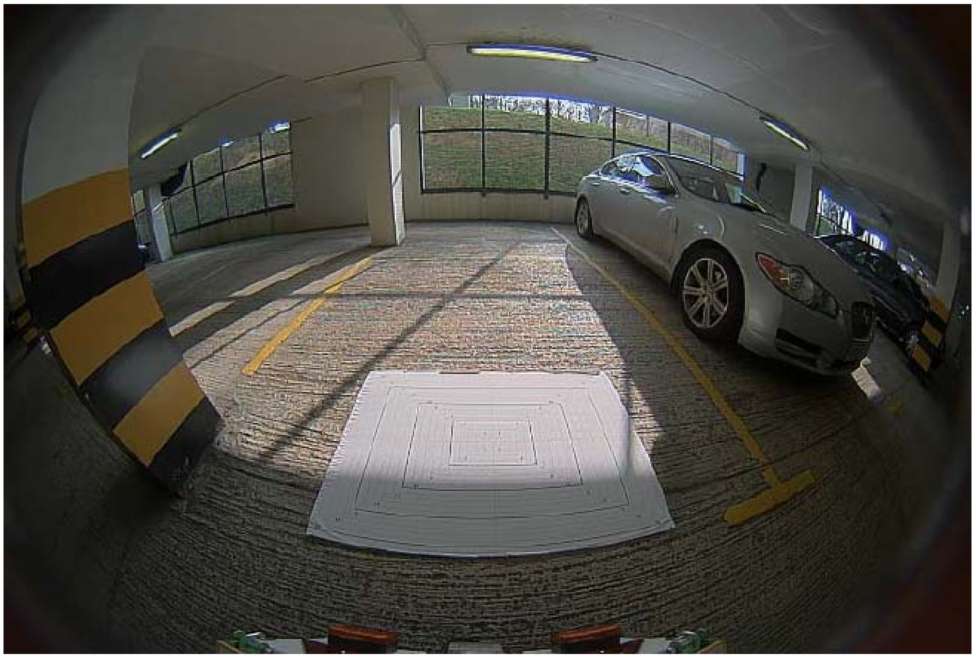


Figure 21. Uncorrected Image

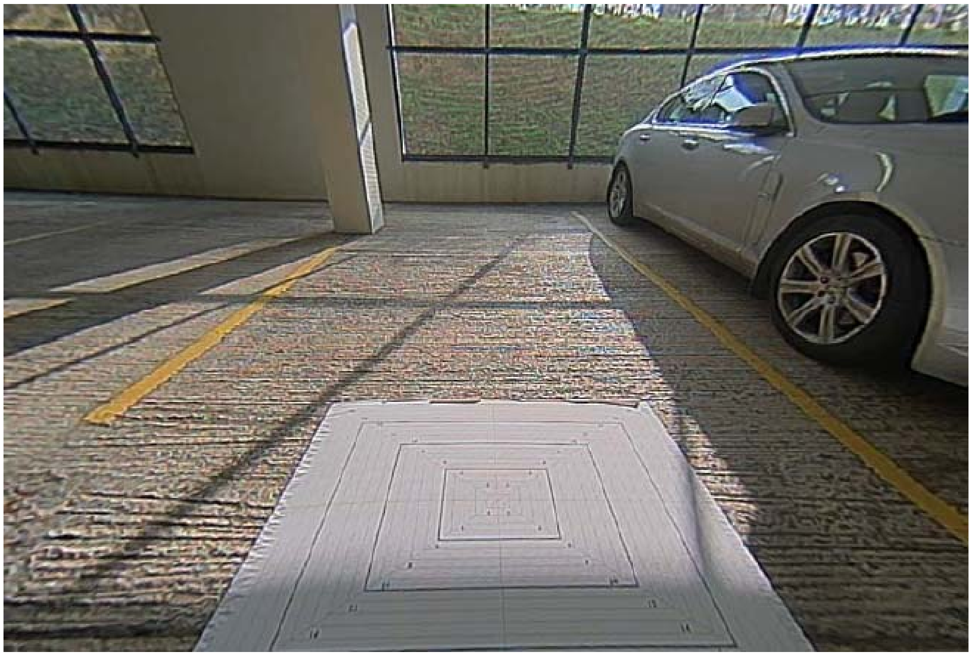


Figure 22. Zoomed

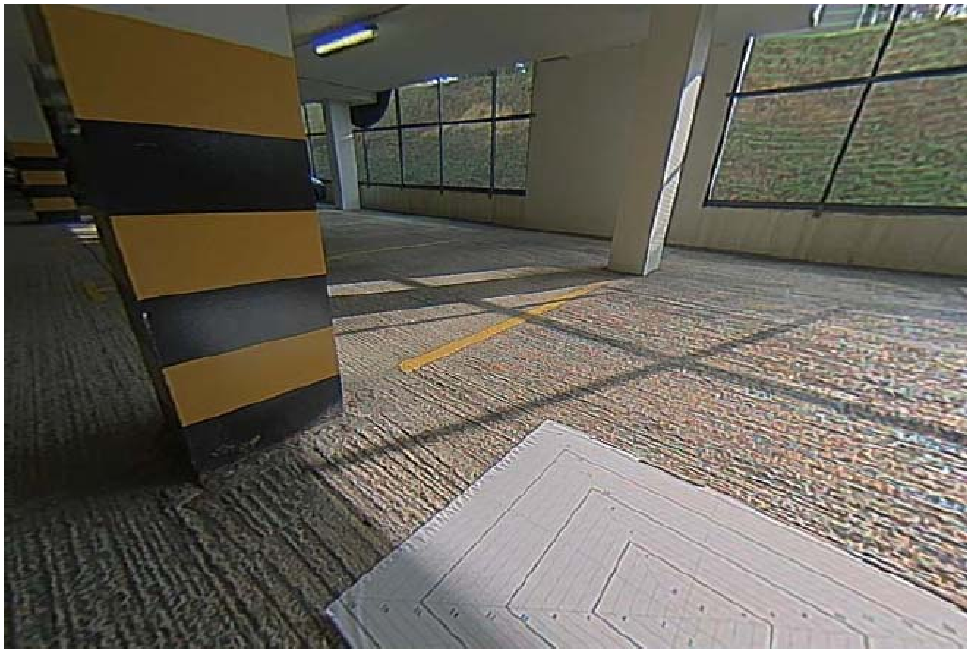


Figure 23. Zoom and Look Left

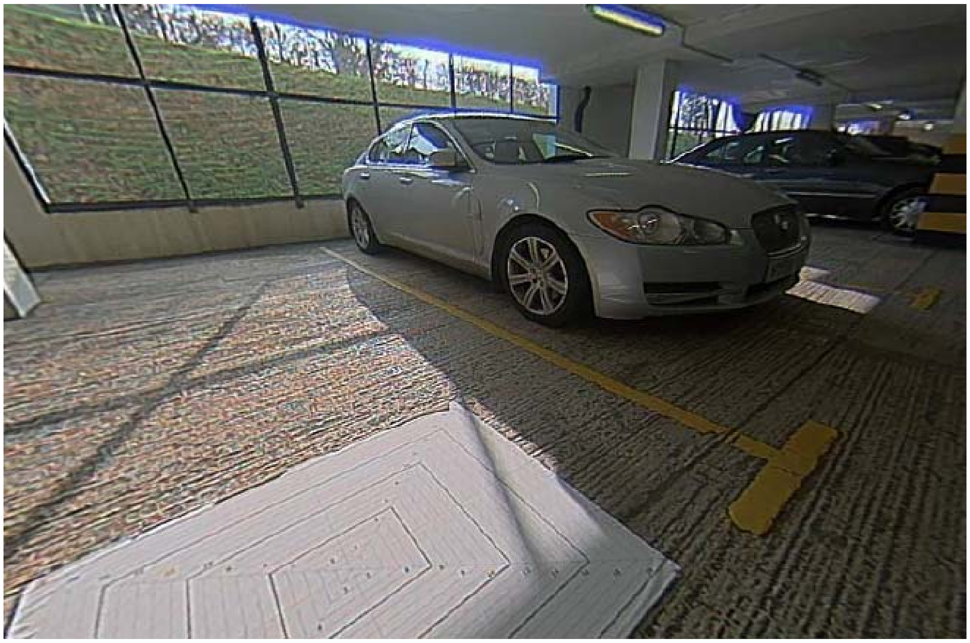


Figure 24. Zoom and Look Right

Multi-Panel

STE supports multi-panel views, these can be 2 or 3 panels. This feature is ideally suited for applications where viewing at a junction is required.

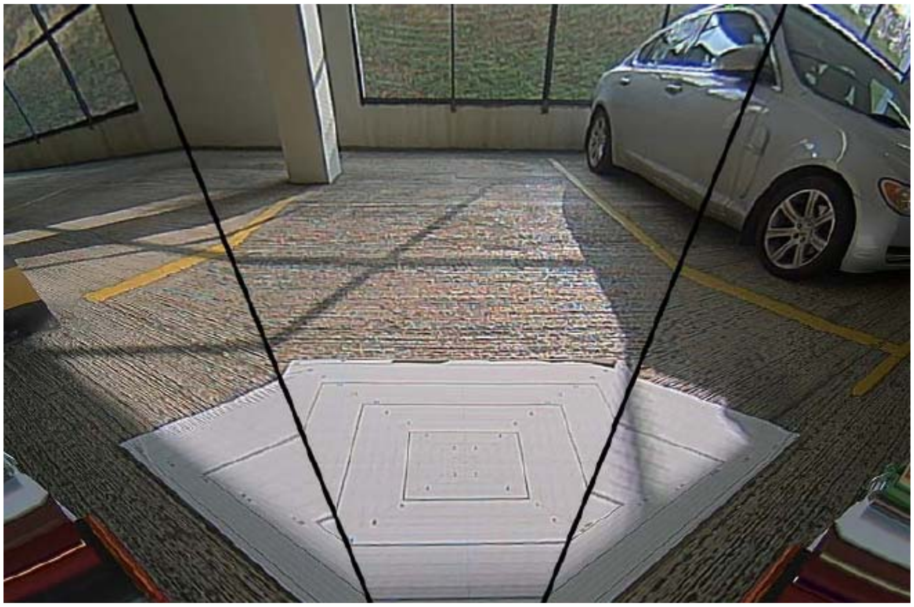


Figure 25. Multi-Panel

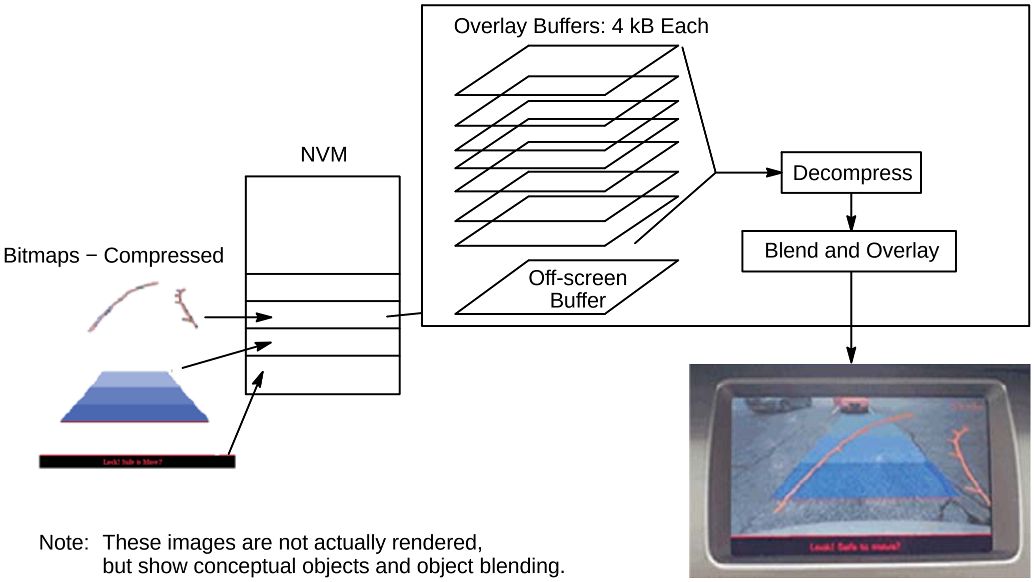
Overlay Capability

Figure 26 highlights the graphical overlay data flow of the AS0140AT. The images are separated to fit into 4 kB blocks of memory after compression.

- Up to seven overlays may be blended simultaneously
- Overlay size up to 720 × 576 pixels rendered
- Selectable readout: rotating order is user programmable
- Dynamic movement through predefined overlay images
- Palette of 32 colors out of 16 million with 16 colors per bitmap
- Each color has a YCbCr (8-8-8 bit) and 8 bits for the Alpha value (Transparency).
- Each layer has a built in fader which when enabled scales the Alpha value for each pixel.
- Blend factors may be changed dynamically to achieve smooth transitions

The overlay engine is controlled through host commands that allow a bitmap to be written piecemeal to memory buffer through the two-wire serial interface, and through a DMA channel direct from SPI Flash memory. Multiple encoding passes may be required to fit an image into a 4 kB 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 seven active overlay buffers.


Overlay Buffers: 4 kB Each
NVM
Decompress
Blend and Overlay
Bitmaps − Compressed
Off-screen Buffer

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

Figure 26. Overlay Data Flow

Serial memory Partition

The contents of Flash/EEPROM memory partition locally into three blocks (see Figure 27):

- 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.

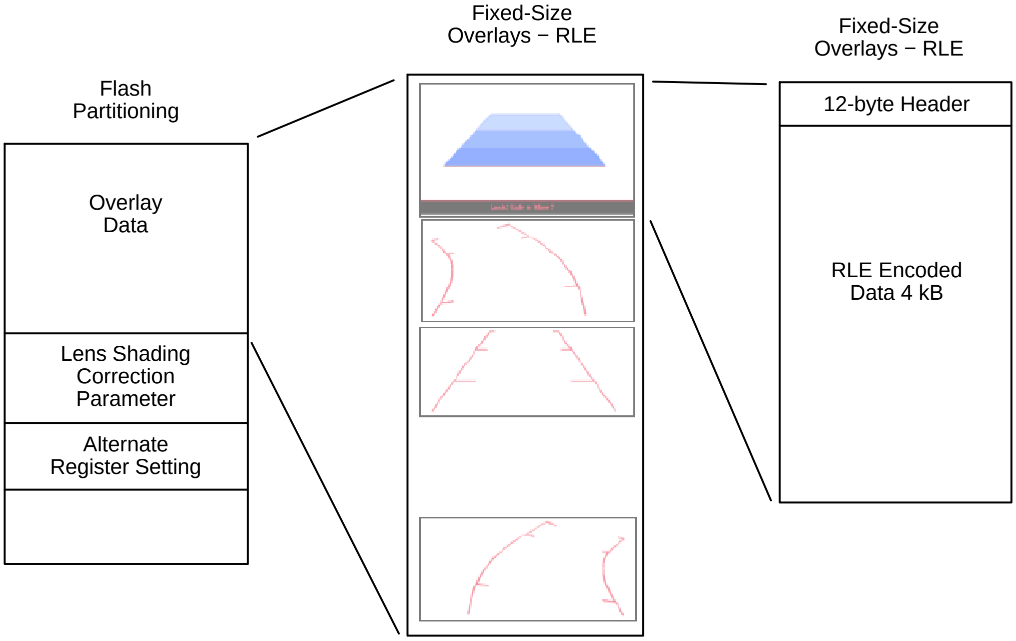

Fixed-Size Overlays − RLE
Fixed-Size Overlays − RLE
Flash Partitioning
12-byte Header
Overlay Data
RLE Encoded Data 4 kB
Lens Shading Correction Parameter
Alternate Register Setting

Figure 27. Memory Partitioning

Overlay Adjustment

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

- The calibration statistics engine supports a windowed 8-bin luma histogram, either row-wise (vertical) or column-wise (horizontal).

- The example calibration statistics function of the firmware 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 also supports a manual mode, which allows the host to access the raw accumulator values directly.

Composite Video Output

The external pin GPIO[3] can 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.

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 28. The differential schematic is shown in Figure 29.

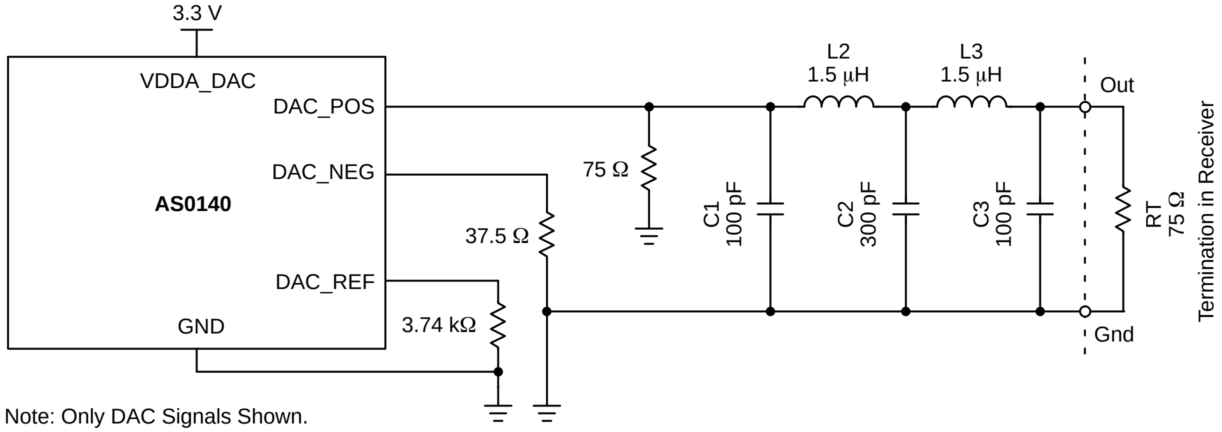

3.3 V
VDDA_DAC
DAC_POS
DAC_NEG
AS0140
DAC_REF
GND
75 Ω
37.5 Ω
3.74 kΩ
L2
1.5 μH
L3
1.5 μH
C1
100 pF
C2
300 pF
C3
100 pF
Out
RT
75 Ω
Gnd
Termination in Receiver
Note: Only DAC Signals Shown.

Figure 28. Single-ended Termination

The DAC is differential, but it may be used to produce single-ended signals provided that the unused (DAC_NEG) output is terminated into a resistance to ground approximately equal to the load on the DAC_POS output. Without this termination, the internal bias circuits will not be kept in their proper operating regions and the dynamic performance of the DAC will be degraded. Termination straight into ground causes all of the power dissipation to occur on the chip, which is undesirable. If a one component saving was absolutely critical, termination straight to ground is a possibility.

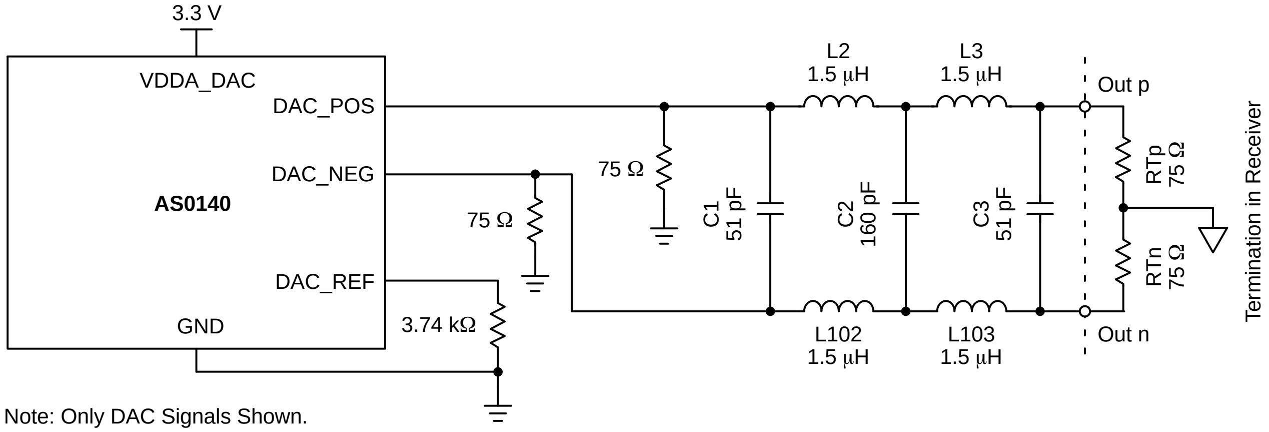
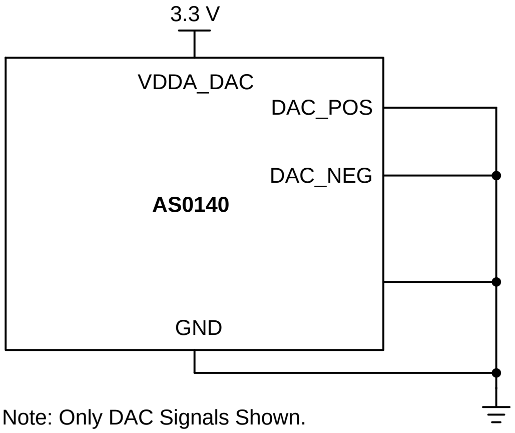

3.3 V
VDDA_DAC
DAC_POS
DAC_NEG
AS0140
DAC_REF
GND
75 Ω
75 Ω
3.74 kΩ
L2
1.5 μH
L3
1.5 μH
C1
51 pF
C2
160 pF
C3
51 pF
L102
1.5 μH
L103
1.5 μH
Out p
Out n
RTp
75 Ω
RTn
75 Ω
Termination in Receiver
Note: Only DAC Signals Shown.

Figure 29. Differential Connection

If the user is not using the analog output then Figure 30 shows how the signals should be connected.



Note: Only DAC Signals Shown.

Figure 30. No DAC

Slave Two-Wire Serial Interface (CCIS)

The two-wire slave serial interface bus enables read/write access to control and status registers within the AS0140AT.

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). SCLK and SDATA are pulled up to VDDIO off-chip by a pull-up resistor of 1.5 kΩ or greater.

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 x14 below. The user can change the slave address by changing a register value.

Table 15. 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 AS0140AT 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 31 shows the typical READ cycle of the host to the AS0140AT. 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.

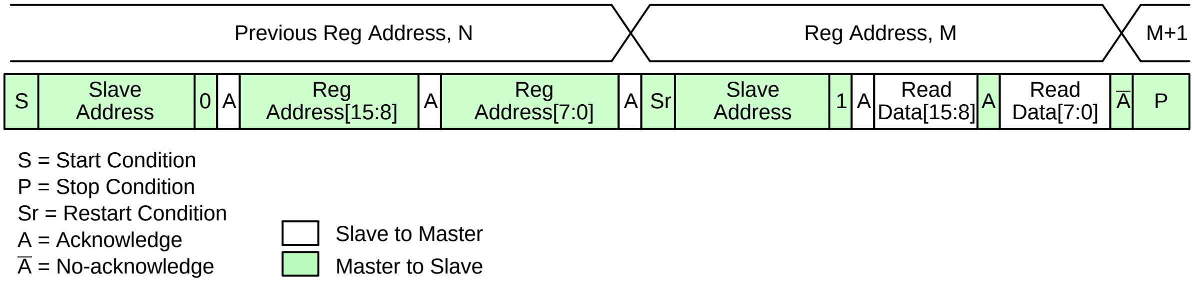

Previous Reg Address, N
Reg Address, M
M+1
S
Slave Address
0
A
Reg Address[15:8]
A
Reg Address[7:0]
A
Sr
Slave Address
1
A
Read Data[15:8]
A
Read Data[7:0]
$\overline{A}$
P
S = Start Condition
P = Stop Condition
Sr = Restart Condition
A = Acknowledge
$\overline{A}$ = No-acknowledge
Slave to Master
Master to Slave

Figure 31. Single READ from Random Location

Single READ from Current Location

Figure 32 shows the single READ cycle without writing the address. The internal address will use the previous address value written to the register.

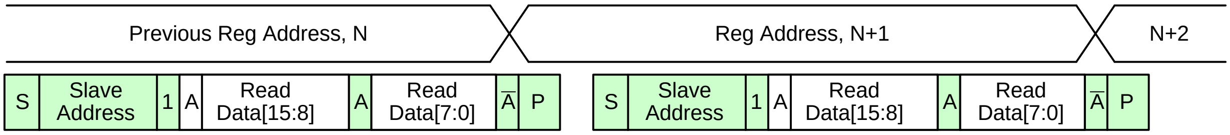

Previous Reg Address, N
Reg Address, N+1
N+2
S
Slave Address
1
A
Read Data[15:8]
A
Read Data[7:0]
$\overline{A}$
P
S
Slave Address
1
A
Read Data[15:8]
A
Read Data[7:0]
$\overline{A}$
P

Figure 32. Single READ from Current Location

Sequential READ, Start from Random Location

This sequence (Figure 33) starts in the same way as the single READ from random location (Figure 31). 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.

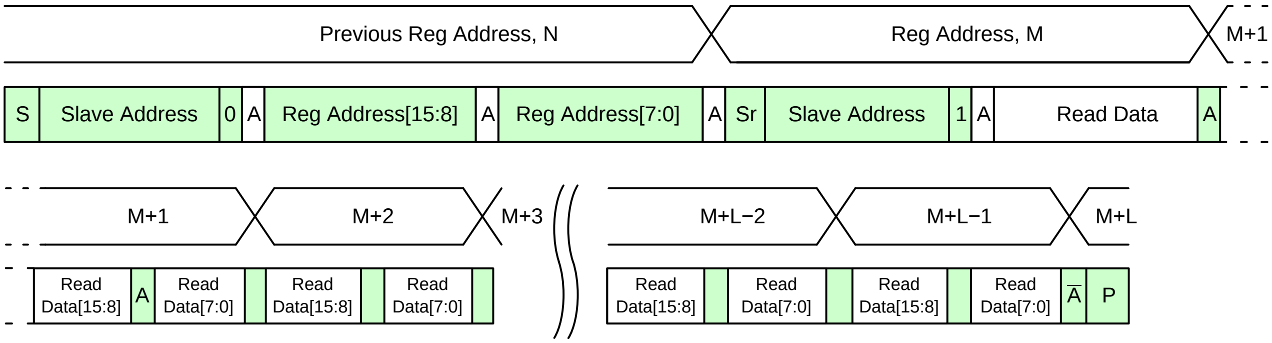

Previous Reg Address, N
Reg Address, M
M+1
S
Slave Address
0
A
Reg Address[15:8]
A
Reg Address[7:0]
A
Sr
Slave Address
1
A
Read Data
A
M+1
M+2
M+3
M+L−2
M+L−1
M+L
Read Data[15:8]
A
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
A̅
P

Figure 33. Sequential READ from Random Location

Sequential READ, Start from Current Location

This sequence (Figure 34) starts in the same way as the single READ from current location (Figure 32). 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.

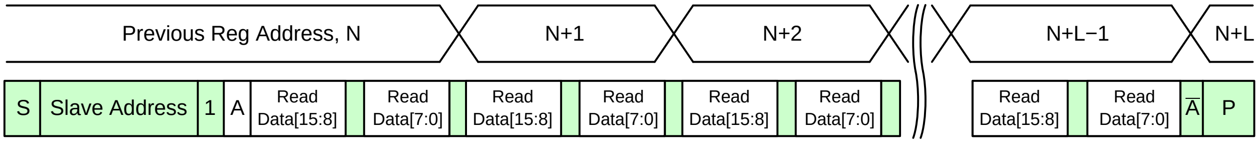

Previous Reg Address, N
N+1
N+2
N+L−1
N+L
S
Slave Address
1
A
Read Data[15:8]
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
Read Data[15:8]
Read Data[7:0]
A̅
P

Figure 34. Sequential READ, Start from Current Location

Single WRITE to Random Location

Figure 35 shows the typical WRITE cycle from the host to the AS0140AT.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.

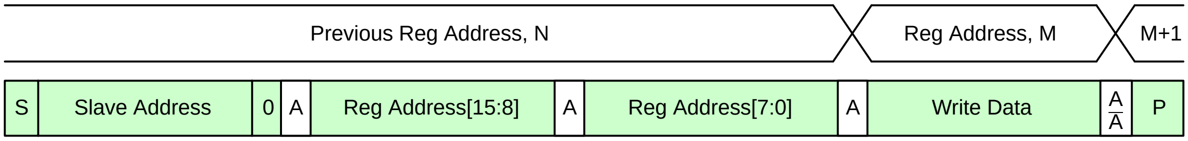

Previous Reg Address, N
Reg Address, M
M+1
S
Slave Address
0
A
Reg Address[15:8]
A
Reg Address[7:0]
A
Write Data
A
A̅
P

Figure 35. Single WRITE to Random Location

Sequential WRITE, Start at Random Location

This sequence (Figure 36) starts in the same way as the single WRITE to random location (Figure 33). 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.

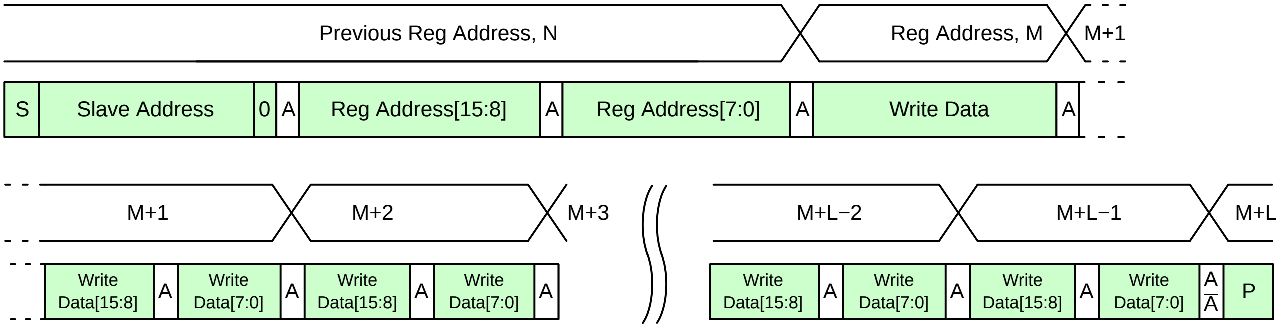

Previous Reg Address, N
Reg Address, M
M+1
S
Slave Address
0
A
Reg Address[15:8]
A
Reg Address[7:0]
A
Write Data
A
M+1
M+2
M+3
M+L−2
M+L−1
M+L
Write Data[15:8]
A
Write Data[7:0]
A
Write Data[15:8]
A
Write Data[7:0]
A
Write Data[15:8]
A
Write Data[7:0]
A
Write Data[15:8]
A
Write Data[7:0]
A
A̅
P

Figure 36. Sequential WRITE, Start at Random Location

Device Configuration

After power is applied and the device is out of reset (either the power on reset, hard or soft reset), it will enter a boot sequence to configure its operating mode. There are essentially three configuration modes: Flash/EEPROM-Config, Auto-Config, and Host-Config.

The AS0140AT firmware supports a System Configuration phase at start-up. This consists of three sub-phases of execution:

Flash detection, then one of:

1. Flash-Config
2. Auto-Config
3. Host-Config

The System Configuration phase is entered immediately following power-up or reset. Then the firmware performs Flash Detection.

Flash Detection attempts to detect the presence of an SPI Flash or EEPROM device:

- If no device is detected, the firmware then samples the SPI_SDI pin state to determine
- The next mode:
  - If SPI_SDI is low, then it enters the Host-Config mode
  - If SPI_SDI is high, then it enters the Auto-Config mode
- If a device is detected, the firmware switches to the Flash-Config mode.

In the Flash-Config mode, the firmware interrogates the device to determine if it contains valid configuration records:

- If no records are detected, then the firmware enters the Auto-Config mode.
- If records are detected, the firmware processes them. By default, when all Flash records are processed the firmware switches to the Host-Config mode. However, the records encoded into the Flash can optionally be used to instruct the firmware to proceed to auto-config, or to start streaming (via a Change-Config).

In the Host-Config mode, the firmware performs no configuration, and remains idle waiting for configuration and commands from the host. The System Configuration phase is effectively complete and the AS0140AT will take no actions until the host issues commands.

The Auto-Config mode uses the GPIO [5..2] pins to configure the operation of the device, such as video format and pedestal (see Table 17, “GPIO Bit Descriptions in Auto-Config”). After Auto-Config completes the firmware switches to the Change-Config mode.

Supported SPI Devices

Table 16 lists supported EEPROM/Flash devices. Devices not compatible will require a firmware patch. Contact ON Semiconductor for additional support.

Table 16. SPI FLASH DEVICES

Manufacturer	Device	Type	Size	Auto-detected	ManuID
Atmel	AT26DF081A	Flash	1 MB	Yes	0x1f4501
Atmel	AT25DF161	Flash	2 MB	Yes	0x1f4602
Sanyo (Note 1)	LE25FW806	Flash	1 MB	Yes	0x622662
ST	M25P05A	Flash	64 kB	Yes	0x202010
ST	M25P16	Flash	2 MB	Yes	0x202015
ST	M95040	EEPROM	512 B	No	0x20ffff
ST	M95020	EEPROM	256 B	No	0x20ffff
ST	M95010	EEPROM	128 B	No	0x20ffff
ST	M95M01	EEPROM	128 kB	No	0x20ffff
Microchip	M25AA080	EEPROM	1 kB	No	0x29ffff
Microchip	M25LC080	EEPROM	1 kB	No	0x29ffff

1. Has been obsoleted.

Table 17. GPIO BIT DESCRIPTIONS IN AUTO-CONFIG

	GPIO[5]	GPIO[4]	GPIO[3]	GPIO[2]
Low (“0”)	Normal	Normal	NTSC	No Pedestal
High (“1”)	Vertical Flip	Horizontal Mirror	PAL	Pedestal

Host Command Interface

The AS0140AT 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. EEPROM or Flash memory is also available to store commands for later execution.

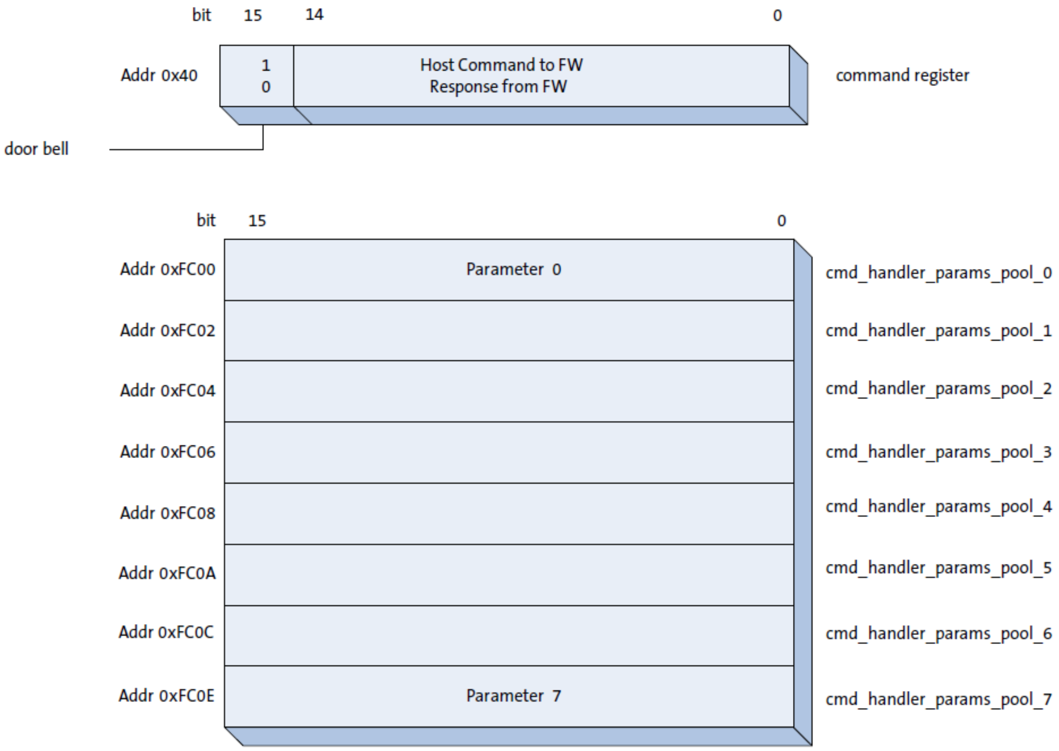

bit
15
14
0
Addr 0x40
1
0
Host Command to FW
Response from FW
command register
door bell
bit
15
0
Addr 0xFC00
Parameter 0
cmd_handler_params_pool_0
Addr 0xFC02
cmd_handler_params_pool_1
Addr 0xFC04
cmd_handler_params_pool_2
Addr 0xFC06
cmd_handler_params_pool_3
Addr 0xFC08
cmd_handler_params_pool_4
Addr 0xFC0A
cmd_handler_params_pool_5
Addr 0xFC0C
cmd_handler_params_pool_6
Addr 0xFC0E
Parameter 7
cmd_handler_params_pool_7

Figure 37. Interface Structure

Command Flow

The host issues a command by writing (through the two wire interface) 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 shared-variable page), then writes the command to Command Register. The firmware’s interrupt handler is invoked, which immediately copies the Command Register contents. 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 become 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.

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. It is strongly recommended that the host tests that the doorbell bit is clear before issuing a command.

Synchronous Command Flow

The typical ‘flow’ for synchronous commands is:

1. The host issues a ‘request’ command to perform an operation.
2. The registered command handler is invoked, validates the command parameters, then performs the operation. The handler returns the command result status to indicate the result of the operation.
3. The host retrieves the command result value, and any associated command response parameters.

Asynchronous Command Flow

The typical ‘flow’ for asynchronous commands is:

1. The host issues a ‘request’ command to start an operation.
2. The registered command handler is invoked, validates and copies the command parameters, then signals a separate task to perform the operation. The handler returns the ENOERR return value to indicate the command was acceptable and is in progress.
3. The host retrieves the command return value – if it is not ENOERR the host knows that the command was not accepted and is not in progress.

4. Subsequently, the host issues an appropriate ‘get status’ command to both poll whether the command has completed, and if so, retrieve any associated response parameters.
5. The registered command handler is invoked, determines the state of the command (via shared variables with the processing task), and returns either ‘EBUSY’ to indicate the command is still in progress, or it returns the result status of the command.
6. The host must re-issue the ‘get status’ command until it does not receive the EBUSY response.

Asynchronous commands exist to allow the Host to issue multiple commands to the various subsystems without having to wait for each command to complete. This prevents the host command interface from being blocked by a long-running command. Therefore, each asynchronous command has a “Get Status” (or similar) command to allow the Host to determine when the asynchronous command completes.

Start-Up Host Command Lock-Out

The AS0140AT firmware implements an internal Host Command ‘lock’. At start-up, the firmware obtains this lock, which prevents the Host from successfully issuing a host command. All host commands will be rejected with EBUSY until the lock is freed.

The firmware releases the Host Command lock when it completes its start-up configuration processing. The time to do this is dependent upon the configuration mechanism. It is recommended that the Host poll the device with the System Manager Get State command until ENOERR is returned.

Once the host can send serial commands it should perform the following sequence.

1. POLL command_register[15] until it clears (This is called the doorbell bit).
2. Continuously issue the SYSMGR_GET_STATE command (0x8101) until the result status is not EBUSY.

Below is some pseudocode that a host could use to implement the above sequence:

```
def systemWaitReadyFollowingReset(numRetries=10):
    """API function: waits for the system to be ready following reset (or powerup)
         - first wait for the doorbell bit to clear - this indicates that the device can
         accept host commands.
         - then wait until the system has completed its configuration phase; the system is
         ready when the SYSMGR_GET_STATE command does not return EBUSY.
         - note the time for the system to be ready is dependent upon the active system
         configuration mode.
         - numRetries is the number of retries before timing-out
         - returns result status code
    """

    # Wait for doorbell bit to clear (indicates device can receive host commands)
    retries = numRetries
    while (0 != retries):
         if (reg.COMMAND_REGISTER.DOORBELL.uncached_value == 0): break # ready to receive
         commands retries -= 1

    if (0 == retries):
         # device failed to respond in time
         return printError(ResultStatus.EIO, 'systemWaitReadyFollowingReset failed
         (doorbell failed to clear)')

    # Wait for the System Manager to complete the System Configuration phase
    retries = numRetries
    while (0 != retries):
         res, currentState = sysmgrGetState()
         if (ResultStatus.ENOERR == res): break # we're done
         if (ResultStatus.EBUSY != res):
            return printError(res, 'systemWaitReadyFollowingReset failed
         (sysmgrGetState failed)')
            retries -= 1

    if (0 == retries):
         # device failed to respond in time
         return printError(ResultStatus.EAGAIN, 'systemWaitReadyFollowingReset failed
         device busy)')

return res
```

Multitasking

The AS0140AT firmware is multitasking; therefore note that it is possible for an internally requested command to be in-progress when the Host issues a command. In these circumstances, the Host command is immediately rejected with EBUSY. The Host should reissue the command after a short interval.

Host Commands

Overview

The AS0140AT supports a number of functional modules or processing subsystems. Each module or subsystem exposes commands to the host to control and configure its operation.

Command Parameters

Command parameters are written to the Parameters Pool shared-variables by the host prior to invoking the command. Similarly, any Command Response parameters are also written back to the Parameters Pool by the firmware.

Result Status Codes

Table 18 shows the result status codes that are written by the Command Handler to the Host Command register, in response to a command.

Table 18. RESULT STATUS CODES

Value	Mnemonic	Typical Interpretation − Each Command may Re-interpret
0x00	ENOERR	No Error – Command was Successful
0x01	ENOENT	No Such Entity
0x02	EINTR	Operation Interrupted
0x03	EIO	I/O Failure
0x04	E2BIG	Too Big
0x05	EBADF	Bad File/Handle
0x06	EAGAIN	Would-block, Try Again
0x07	ENOMEM	Not Enough Memory/Resource
0x08	EACCES	Permission Denied
0x09	EBUSY	Entity Busy, Cannot Support Operation
0x0A	EEXIST	Entity Exists
0x0B	ENODEV	Device Not Found
0x0C	EINVAL	Invalid Argument
0x0D	ENOSPC	No Space/Resource to Complete
0x0E	ERANGE	Parameter Out-of-Range
0x0F	ENOSYS	Operation Not Supported
0x10	EALREADY	Already Requested/Exists

NOTE: Note: Any unrecognized host commands will be immediately rejected by the Command Handler, with result status code ENOSYS.

Summary of Host Commands

Tables 19 through 30 show summaries of the host commands. The commands are divided into the following sections:

- System Manager
- Overlay
- GPIO
- Flash Manager
- STE
- Sequencer
- Patch Loader
- Miscellaneous
- Calibration Stats

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 19. SYSTEM MANAGER HOST COMMAND

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
Config Power Management	0x8102	Synchronous	Configures the Power State of the System

Table 20. OVERLAY HOST COMMANDS

Overlay Host Command	Value	Type	Description
Enable Overlay	0x8200	Synchronous	Enable or Disable the Overlay Subsystem
Get Overlay State	0x8201	Synchronous	Retrieves 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
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
Get String	0x820D	Synchronous	Get the Current Character String
Load String	0x820E	Asynchronous	Load a Character String (from Flash)

Table 21. STE MANAGER HOST COMMANDS

STE Manager Host Command	Value	Type	Description
Config	0x8310	Synchronous	Configure using the Default NTSC or PAL Configuration Stored in ROM
Load Config	0x8311	Asynchronous	Load a Configuration from SPI NVM to the Configuration Cache
Load Status	0x8312	Synchronous	Get Status of a Load Config Request
Write Config	0x8313	Synchronous	Write a Configuration (via CCIS) to the Configuration Cache

Table 22. 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 GPIO 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 NVM
Get GPI Association	0x8405	Synchronous	Retrieve a GPIO Pin Association

Table 23. 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 NVM Subsystem
Read	0x8504	Asynchronous	Read Data from the SPI NVM
Write	0x8505	Asynchronous	Write Data to the SPI NVM
Erase Block	0x8506	Asynchronous	Erase a Block of Data from the SPI NVM
Erase Device	0x8507	Asynchronous	Erase the SPI NVM Device
Query Device	0x8508	Asynchronous	Query Device-specific Information
Status	0x8509	Synchronous	Obtain Status of Current Asynchronous Operation
Config Device	0x850A	Synchronous	Configure the Attached SPI NVM Device

Table 24. SEQUENCER HOST COMMANDS

Sequencer Host Command	Value	Type	Description
Set GPIO Property	0x8400	Synchronous	Set a Property of One or More GPIO Pins
Refresh	0x8606	Asynchronous	Refresh the Automatic Image Processing Algorithm Configuration
Refresh Status	0x8607	Synchronous	Retrieve the Status of the Last Refresh Operation

Table 25. PATCH LOADER HOST COMMANDS

Patch Loader Host Command	Value	Type	Description
Load Patch	0x8700	Asynchronous	Load a Patch from SPI NVM and Automatically Apply
Status	0x8701	Synchronous	Get Status of an Active Load Patch or Apply Patch Request

Table 26. MISCELLANEOUS HOST COMMANDS

Miscellaneous Host Command	Value	Type	Description
Invoke Command Seq	0x8900	Synchronous	Invoke a Sequence of Commands Stored in SPI NVM
Config Command Seq Processor	0x8901	Synchronous	Configures the Command Sequence Processor
Wait for Event	0x8902	Synchronous	Wait for a System Event to be Signalled

Table 27. CALIBRATION STATS HOST COMMANDS

Calibration Stats Host Command	Value	Type	Description
Calib Stats Control	0x8B00	Asynchronous	Start Statistics Gathering
Calib Stats Read	0x8B01	Synchronous	Read the Results Back

Table 28. EVENT MONITOR HOST COMMANDS

Event Monitor Host Command	Value	Type	Description
Event Monitor Set Association	0x8C00	Synchronous	Associate a System Event with a Command Sequence Stored in NVM
Event Monitor Get Association	0x8C01	Synchronous	Retrieve an Event Association

Table 29. CCI MANAGER HOST COMMANDS

CCI Manager Host Command	Value	Type	Description
Get Lock	0x8D00	Asynchronous	Request the CCI Manager Access Lock
Lock Status	0x8D01	Synchronous	Retrieve the Status of the Access Lock Request
Release Lock	0x8D02	Synchronous	Release the CCI Manager Access Lock
Config	0x8D03	Synchronous	Configure the CCI Manager and Underlying CCI Subsystem
Set Device	0x8D04	Synchronous	Set the Target CCI Device Address
Read	0x8D05	Asynchronous	Read One or More Bytes from a 16-bit Address
Write	0x8D06	Asynchronous	Write One or More Bytes to a 16-bit Address
Write Bitfield	0x8D07	Asynchronous	Read-Modify-Write 16-bit Data to a 16-bit Address
CCI Status	0x8D08	Synchronous	Obtain Status of Current Asynchronous Operation

Table 30. SENSOR MANAGER HOST COMMANDS

Sensor Manager Host Command	Value	Type	Description
Discover Sensor	0x8E00	Synchronous	Discover Sensor
Initialize Sensor	0x8E01	Synchronous	Initialize Attached Aensor

Usage Modes

How a camera based on the AS0140AT will be configured depends on what features are used. A back-up camera with dynamic input from the steering system will require a μC with a system bus interface. Flash sizes vary depending on the register and firmware data being transferred − somewhere between 1 kB to 16 MB. The two-wire bus is adequate since only high-level commands are used.

The AS0140AT can be configured by a serial EEPROM or Flash through the SPI Interface.

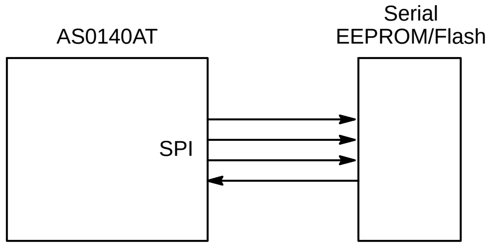


Figure 38. Flash Mode

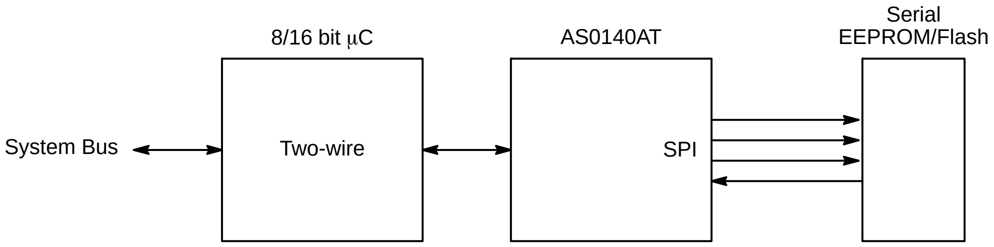


Figure 39. Host Mode with Flash

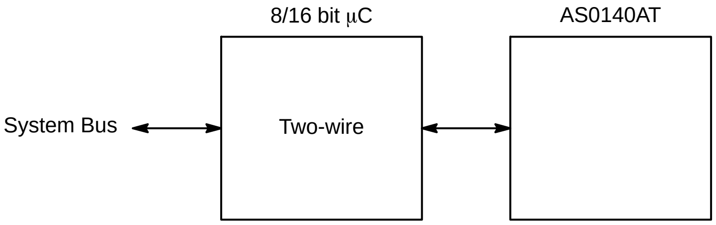


Figure 40. Host Mode

Electrical Specifications

Table 31. ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Min	Max	Unit
V_{DDIO}	I/O Power (2.8V)	−0.3	4	V
V_{AA}	Analog Power (2.8V)	−0.3	4	V
V_{DDA_DAC}	Video Analog DAC Power (3.3V)	−0.3	4	V
V_{DD}	Digital Power (1.8V)	−0.3	2.4	V
V_{IN}	DC Input Voltage	−0.3	V_{DDIO} + 0.3	V
V_{OUT}	DC Output Voltage	−0.3	V_{DDIO} + 0.3	V
T_{ST}	Storage Temperature	−50	150	°C

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

Table 32. ELECTRICAL CHARACTERISTICS AND OPERATING CONDITIONS

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V_{DDIO}	IO Power		2.3	2.8	3.1	V
V_{AA}	Analog Power		2.5	2.8	3.1	V
V_{DDA_DAC}	DAC Analog power		3	3.3	3.6	V
V_{DD}	Digital Power		1.7	1.8	1.98	V
T_A	Functional Operating Temperature (Ambient – T_A)		−40	−	105	°C
T_{ST}	Storage Temperature		−50	−	150	°C

Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

Table 33. AC ELECTRICAL CHARACTERISTICS

(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} = V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C unless otherwise stated)

Symbol	Parameter	Condition	Min	Typ	Max	Unit
f_{EXTCLK}	External Clock Frequency (Note 1)		6	−	30	MHz
D_{EXTCLK}	External Input Clock Duty Cycle		40	50	60	%
t_{EXTCLK_JITTER}	External Input Clock Jitter		−	−	500	ps
t_{PIXCLK_JITTER}	Pixel Clock Output Jitter		−	2.5	−	ns
f_{PIXCLK}	Pixel Clock Frequency (One-clock/Pixel)		6	−	74.125	MHz
	Pixel Clock Frequency (Two-clocks/Pixel)		6	−	84	MHz
$t_{RPIXCLK}$	Pixel Clock Rise Time (10−90%)	C_{LOAD} = 35 pF	−	2.5	5	ns
$t_{FPIXCLK}$	Pixel Clock Fall Time (10−90%)	C_{LOAD} = 35 pF	−	2.5	5	ns
t_{PD}	PIXCLK to Data Valid		−	1	5	ns
t_{PFH}	PIXCLK to FV HIGH		−	1	5	ns
t_{PLH}	PIXCLK to LV HIGH		−	1	5	ns
t_{PFL}	PIXCLK to FV LOW		−	1	5	ns
t_{PLL}	PIXCLK to LV LOW		−	1	5	ns

1. V_{IH}/V_{IL} restrictions apply.

Table 34. DC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	Input HIGH Voltage (Note 1)		V_{DDIO} * 0.8	–	V
V_{IL}	Input LOW Voltage (Note 1)		–	V_{DDIO} * 0.2	V
I_{IN}	Input Leakage Current (Note 2)	V_{IN} = 0 V or V_{IN} = V_{DDIO}	–	10	μA
V_{OH}	Output HIGH Voltage		V_{DDIO} * 0.8	–	V
V_{OL}	Output LOW Voltage		–	V_{DDIO} * 0.2	V

1. V_{IH} and V_{IL} have min/max limitations specified by absolute ratings.
2. Excludes pins that have internal PU resistors.

Table 35. VIDEO DAC ELECTRICAL CHARACTERISTICS

(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} = V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C unless otherwise stated)

Symbol	Parameter	Min	Typ	Max	Unit
DNL	Differential Non-linearity	−	±1	−	LSB
INL	Integral Non-linearity	−	±3	−	LSB
C_{LOAD}	Load Capacitance	−	−	10	pF
OER	Offset Error	−1	−	1	% FS
DGER	Gain Error	−2	−	2	% FS
GER	Absolute Gain Error	−5	−	5	% FS

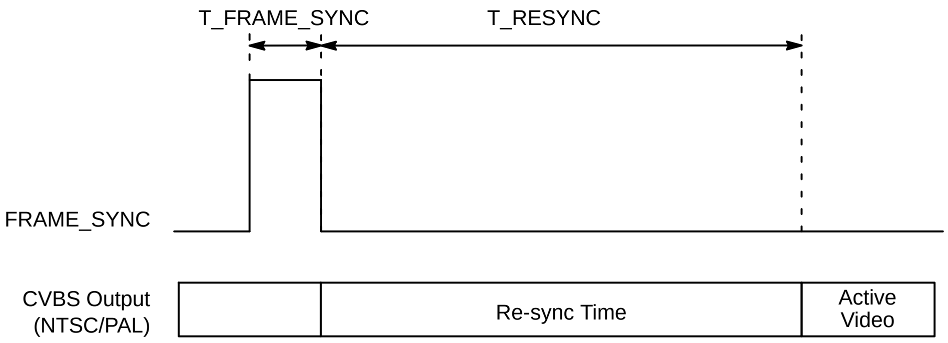

T_FRAME_SYNC
T_RESYNC
FRAME_SYNC
CVBS Output
(NTSC/PAL)
Re-sync Time
Active
Video

Figure 41. Frame_Sync (Interlaced Operation) Diagram

Table 36. FRAME_SYNC (INTERLACED OPERATION) PARAMETERS

Parameter	Name	Conditions	Min	Typ	Max	Unit
T_FRAME_SYNC	T_FRAME_SYNC	−	−	−	3	EXTCLK Cycles
T_RESYNC	T_RESYNC	NTSC	−	100	−	ms
T_RESYNC	T_RESYNC	PAL	−	120	−	ms

NTSC and PAL Signal Parameters

Table 37. NTSC AND PAL SIGNAL PARAMETERS
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} = V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C unless otherwise stated)

Parameter	NTSC	PAL	Unit
Number of Lines per Frame	525	625	
Line Frequency	15734.264	15625	Hz
Field Frequency	59.94	50	Hz
Sync Level	40	43	IRE
Burst Level	40	43	IRE
Black Level	7.5	0	IRE
White Level	100	100	IRE

3. IRE ∼ 7.14 mV.
4. DAC_REF = 3.74 kOhm; Load = 37.5 Ohm.

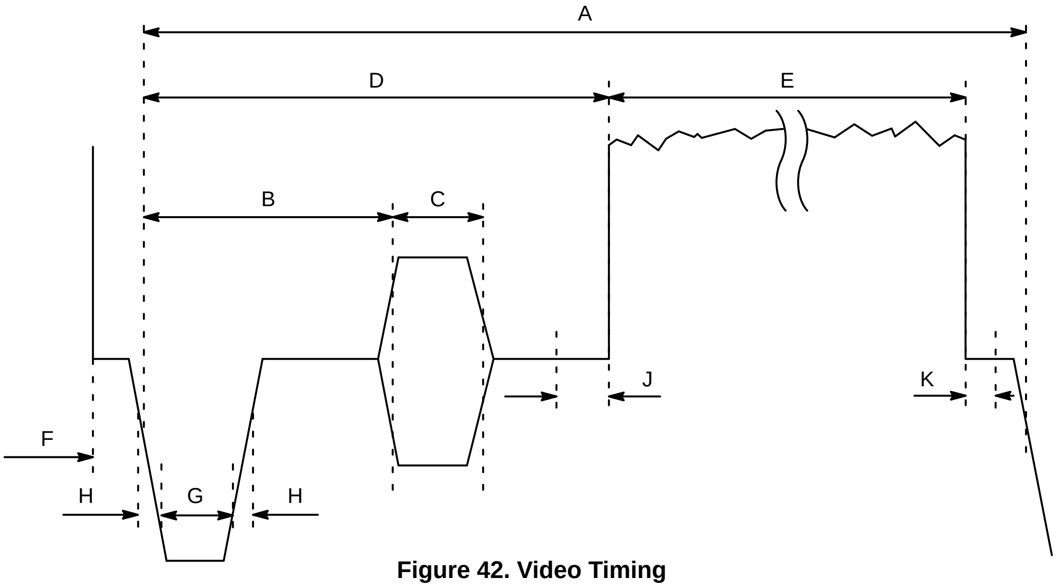

A
D
E
B
C
J
K
F
H
G
H

Figure 42. Video Timing

Table 38. VIDEO TIMING: SPECIFICATION FROM REC. ITU−R BT.470

	Signal	NTSC 27 MHz	PAL 27 MHz	Unit
A	H Period	63.556	64.00	μs
B	Hsync to Burst	4.71 to 5.71	5.60 ±0.10	μs
C	Burst	2.23 to 3.11	2.25 ±0.23	μs
D	Hsync to Signal	9.20 to 10.30	10.20 ±0.30	μs
E	Video Signal	52.655 ±0.20	52 + 0, −0.3	μs
F	Front	1.27 to 2.22	1.5 + 0.3, −0.0	μs
G	Hsync Period	4.70 ±0.10	4.70 ±0.20	μs
H	Sync Rising/Falling Edge	≤ 0.25	0.20 ±0.10	μs

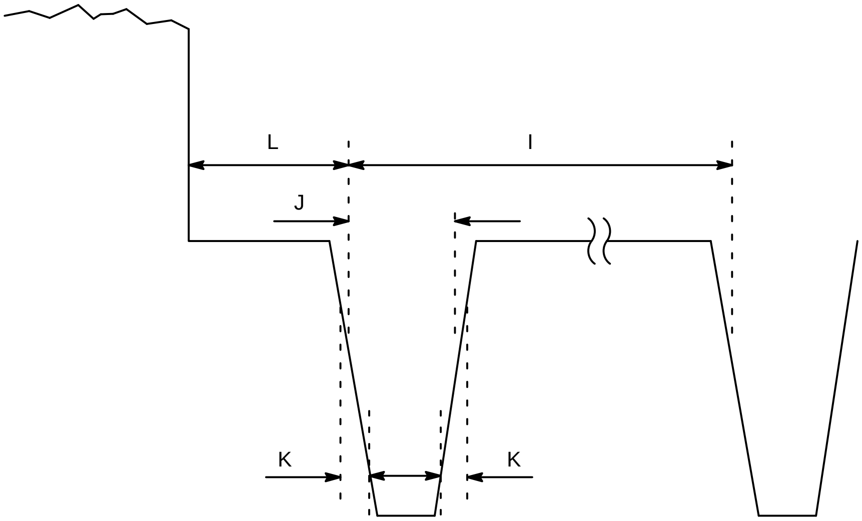


Figure 43. Equalizing Pulse

Table 39. EQUALIZING PULSE: SPECIFICATION FROM REC. ITU−R BT.470

	Signal	NTSC 27 MHz	PAL 27 MHz	Unit
I	H/2 Period	31.778	32.00	μs
J	Pulse Width	2.30 ±0.10	2.35 ±0.10	μs
K	Pulse Rising/Falling Edge	≤ 0.25	0.25 ±0.05	μs
L	Signal to Pulse	1.50 ±0.10	3.0 ±2.0	μs

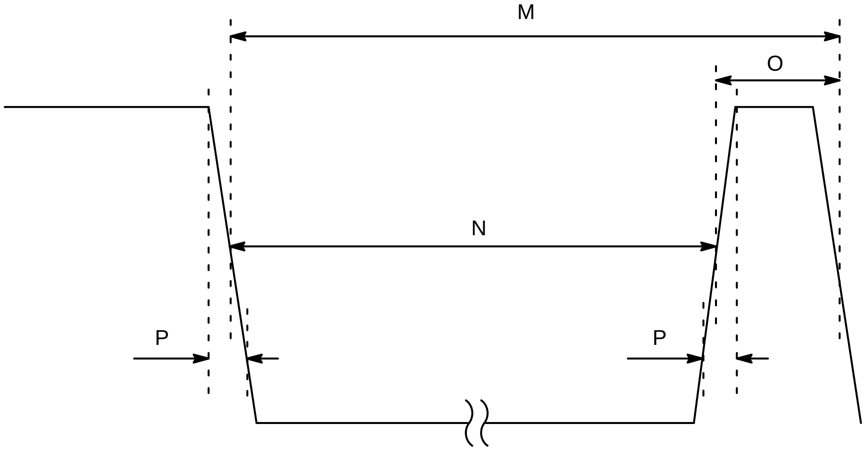


Figure 44. V Pulse

Table 40. V PULSE: SPECIFICATION FROM REC. ITU−R BT.470

	Signal	NTSC 27 MHz	PAL 27 MHz	Unit
M	H/2 Period	31.778	32.00	μs
N	Pulse Width	27.10 (Nominal)	27.30 ±0.10	μs
O	V Pulse Interval	4.70 ±0.10	4.70 ±0.10	μs
P	Pulse Rising/Falling Edge	≤ 0.25	0.25 ±0.05	μs

Table 41. STANDBY CURRENT CONSUMPTION
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} = V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C unless otherwise stated)

Parameter	Conditions	Min	Typ	Max	Unit
Hard Standby with STANDBY pin High	EXTCLK off	−	1.3	5.0	mA
Hard Standby with STANDBY pin High	EXTCLK on	−	2.1	8.0	mA

5. VAA & VDD supplies only.

Table 42. OPERATING CURRENT CONSUMPTION

(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} not included in measurement, V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C, HDR Mode unless otherwise stated)

Symbol	Conditons	Min	Typ	Max	Unit
VDDIO = 2.8 V		TBD	2.8	TBD	V
VAA = 2.8 V		TBD	2.8	TBD	V
VDDA_DAC = 3.3 V		TBD	3.3	TBD	V
VDD = 1.8 V		TBD	1.8	TBD	V
IAA	NTSC HDR Mode	−	32	38	mA
	NTSC Linear Mode	−	32	38	mA
	PAL	−	29	35	mA
	STE Progressive YCbCr_422_10_10	−	41	49	mA
	STE Progressive YCbCr_422_16	−	31	37	mA
	STE Progressive YCbCr_422_8_8	−	41	49	mA
IDDA_DAC	NTSC HDR Mode	−	19	26	mA
	NTSC Linear Mode	−	19	26	mA
	PAL	−	19	26	mA
	STE Progressive YCbCr_422_10_10	−	0	−	mA
	STE Progressive YCbCr_422_16	−	0	−	mA
	STE Progressive YCbCr_422_8_8	−	0	−	mA
IDD	NTSC HDR Mode	−	209	237	mA
	NTSC Linear Mode	−	197	228	mA
	PAL	−	202	230	mA
	STE Progressive YCbCr_422_10_10	−	191	217	mA
	STE Progressive YCbCr_422_16	−	231	264	mA
	STE Progressive YCbCr_422_8_8	−	191	217	mA
Total Power Consumption	NTSC HDR Mode	−	527	619	mW
	NTSC Linear Mode	−	506	603	mW
	PAL	−	505	598	mW
	STE Progressive YCbCr_422_10_10	−	456	528	mW
	STE Progressive YCbCr_422_16	−	504	579	mW
	STE Progressive YCbCr_422_8_8	−	456	528	mW

Table 43. INRUSH CURRENT

Supply	Voltage (V)	Maximum Current (mA)
VAA	2.8	509
VDDA_DAC	3.3	471
VDD	1.8	519
VDDIO	2.8	502

6. Based on maximum external LDO drive of 500 mA.
7. 25°C.

Two-Wire Serial Register Interface

The electrical characteristics of the two-wire serial register interface (SCLK, SDATA) are shown in Figure 45 and Table 44.

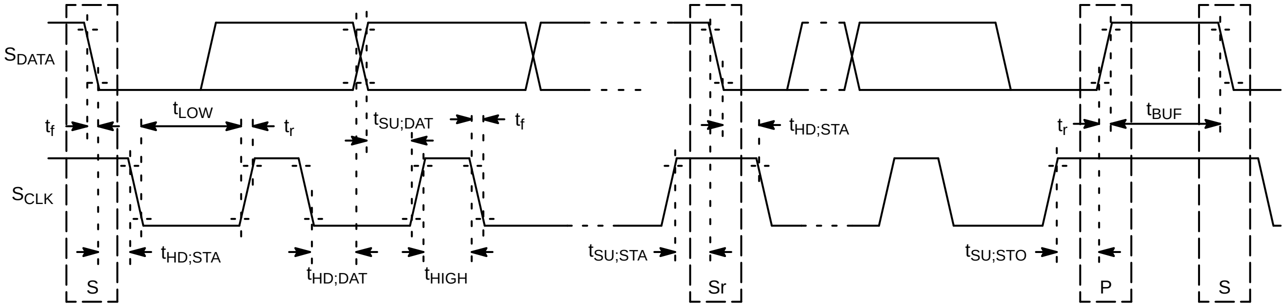


Figure 45. Two-Wire Serial Bus Timing Parameters

Table 44. TWO-WIRE SERIAL BUS CHARACTERISTICS (CCIS)

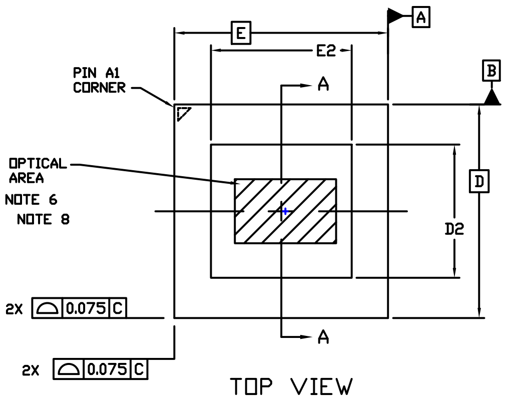
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DDIO} = V_{AA} = 2.8 V, V_{DDA_DAC} = 3.3 V, V_{DD} = 1.8 V, T_A = 25°C, unless otherwise stated)

Parameter	Symbol	Standard-Mode		Fast-Mode		Unit
		Min	Max	Min	Max	
S_{CLK} Clock Frequency	f_{SCL}	0	100	0	400	KHz
Hold Time (Repeated) START Condition, After This Period, the First Clock Pulse is Generated	$t_{HD;STA}$	4.0	–	0.6	–	μs
LOW Period of the S_{CLK} Clock	t_{LOW}	4.7	–	1.3	–	μs
HIGH Period of the S_{CLK} Clock	t_{HIGH}	4.0	–	0.6	–	μs
Set-up Time for a Repeated START Condition	$t_{SU;STA}$	4.7	–	0.6	–	μs
Data Hold Time	$t_{HD;DAT}$	0 (Note 2)	3.45 (Note 3)	0	0.9 (Note 3)	μs
Data Set-up Time	$t_{SU;DAT}$	250	–	100	–	ns
Rise Time of Both S_{DATA} and S_{CLK} Signals (10−90%)	t_r	–	1000	20 + 0.1Cb (Note 4)	300	ns
Fall Time of Both S_{DATA} and S_{CLK} Signals (10−90%)	t_f	–	300	20 + 0.1Cb (Note 4)	300	ns
Set-up Time for STOP Condition	$t_{SU;STO}$	4.0	–	0.6	–	μs
Bus Free Time between a STOP and START Condition	t_{BUF}	4.7	–	1.3	–	μs
Capacitive Load for Each Bus Line	Cb	–	400	–	400	pF
Serial Interface Input Pin Capacitance	C_{IN_SI}	–	3.3	–	3.3	pF
S_{DATA} Max Load Capacitance	C_{LOAD_SD}	–	30	–	30	pF
S_{DATA} Pull-up Resistor	R_{SD}	1.5	4.7	1.5	4.7	kΩ

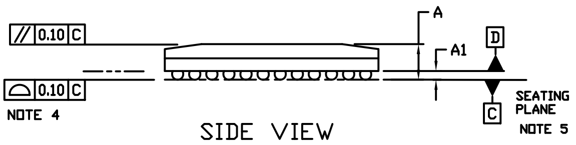
1. All values referred to V_{IHmin} = 0.9 V_{DDIO} and V_{ILmax} = 0.1 V_{DDIO} levels. EXTCLK = 27 MHz.
2. A device must internally provide a hold time of at least 300 ns for the S_{DATA} signal to bridge the undefined region of the falling edge of S_{CLK}.
3. The maximum $t_{HD;DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the t_{LOW} signal.
4. Cb = total capacitance of one bus line in pF.

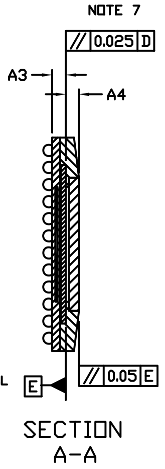
PACKAGE DIMENSIONS

IBGA130 8.5x8.5
CASE 503BK
ISSUE B

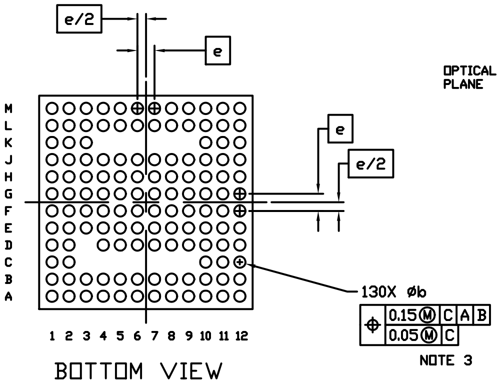


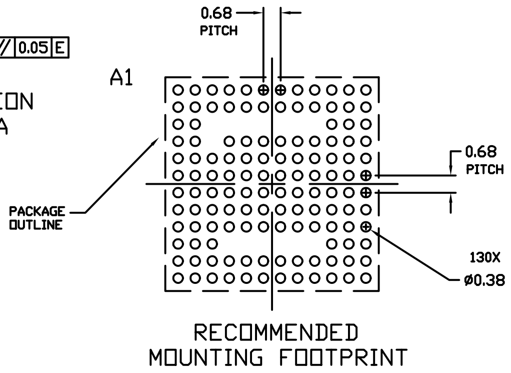
NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: MILLIMETERS
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER PARALLEL TO DATUM C.
4. COPLANARITY APPLIES TO THE SPHERICAL CROWNS OF THE SOLDER BALLS.
5. DATUM C, THE SEATING PLANE, IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
6. MAXIMUM ROTATION OF OPTICAL AREA RELATIVE TO D AND E WILL BE 1°. OPTICAL AREA IS DEFINED BY THE ACTIVE PIXEL ARRAY. REFER TO THE DEVICE DATA SHEET FOR TOTAL ARRAY AND FIRST ACTIVE PIXEL DEFINITIONS.
7. PARALLELISM APPLIES ONLY TO THE OPTICAL AREA.
8. OPTICAL CENTER OFFSET WITH RESPECT TO THE PACKAGE CENTER IS X=166 MICRONS, Y=0 MICRONS ±75 MICRONS.





DIM	MILLIMETERS MIN.	MILLIMETERS MAX.
A	---	1.57
A1	030	0.40
A3	0.495	0.595
A4	0.475	0.575
b	0.40	0.50
D	8.50 BSC	
D2	5.20	5.40
E	8.50 BSC	
E2	5.50	5.70
e	0.68 BSC	





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

ON Semiconductor and ON are trademarks of Semiconductor Components Industries, LLC dba ON Semiconductor or its subsidiaries in the United States and/or other countries. ON Semiconductor owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of ON Semiconductor's product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent−Marking.pdf. ON Semiconductor reserves the right to make changes without further notice to any products herein. ON Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does ON Semiconductor 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. Buyer is responsible for its products and applications using ON Semiconductor products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by ON Semiconductor. "Typical" parameters which may be provided in ON Semiconductor 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. ON Semiconductor does not convey any license under its patent rights nor the rights of others. ON Semiconductor products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use ON Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold ON Semiconductor 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 ON Semiconductor was negligent regarding the design or manufacture of the part. ON Semiconductor is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative