Advanced Image Coprocessor for onsemi's Sensors

6.5 mm x 6.5 mm Package Data Sheet

AP1302

Table 1. KEY PERFORMANCE PARAMETERS

Parameter		Value
Primary camera interface		Up to 4–lane MIPI (up to 1.2 Gbps/lane)
Primary camera input format		RAW6, RAW8, RAW10, RAW12
Output interface		Up to 4–lane MIPI (up to 1.2 Gbps/lane)
Output format		YUV422 8–bit, YUV420 8–bit, RGB888, RGB565, RGB555, JPEG, RAW8, RAW10, RAW12, MJPEG
Secondary camera interface		Up to 3–lane MIPI (up to 1.2 Gbps/lane)
Secondary camera input format		RAW6, RAW8, RAW10, RAW12
Maximum resolution		4224x3156 (13 MP)
Maximum frame rate		30 fps at 13Mp, 120 fps at 1080p
Maximum output clock frequency		MIPI clock up to 600 MHz (1200 Mbps DDR)
Maximum color processing frequency		450 Mpixels per second
Supply voltage	CORE	1.2 V nominal ± 5%
	I/O	1.8 V nominal ± 10%
	PLL	1.2 V nominal ± 5%
	MIPI	1.2 V ± 5%
Operating temperature		–30°C to +70°C (ambient) –30°C to +85°C (junction)
Process		55 nm
ESD Susceptibility		2000 V HBM

Features

- Optimized for operation with **onsemi**'s Clarity+™ Pixel technology sensors as well as Bayer pattern CMOS Image Sensors (CIS)
- Up to 13 MP (4224x3156) sensor support
- Designed to support **onsemi** sensors that provide video interlaced HDR (iHDR).
- 450 MP/second processing (13 MP at 30/1080p at 120/720p at 240 – subject to sensor limitations
- Color gaining and gamma correction
- Frame rate reduction, image resizing and clipping
- Auto exposure, auto white balance, auto focus, auto flicker detection and mitigation
- Adaptive Tone Mapping, Local Tone Mapping
- Face detection
- Flexible synchronization and triggering
- Smooth digital zoom and panning
- Test pattern generator
- Programmable JPEG encoder with EXIF header support
- MJPEG Video Compression
- SPEEDTAGS Encode support
- Two–wire serial interface (I^2C) for sensor and peripheral control and register access supporting Standard (100 kbps), Fast mode (400 kbps), FM+ (1 Mbps) and HS (3.4 Mbps)
- Dual on–chip 32–bit RISC processor cores
- Dual sensor support (second camera or 3D bridge application)
- 6 MIPI data lanes shared between two sensor interfaces (4+2/ 3+3)
- Four–wire serial interface (SPI slave) for register access supporting up to 25 Mbps
- 12 GPIOs (shared functionality with SPI master and slave, second I^2C master)
- Fail–safe IO, programmable slew–rate control

Applications

- IoT
- Drones
- Machine vision
- AR/VR/MR
- Video conferencing

Note: onsemi will only support its own image sensors on this device.

This document, and the information contained herein, is CONFIDENTIAL AND PROPRIETARY and the property of Semiconductor Components Industries, LLC., dba **onsemi**. It shall not be used, published, disclosed or disseminated outside of the Company, in whole or in part, without the written permission of **onsemi**. Reverse engineering of any or all of the information contained herein is strictly prohibited.
© 2013, SCILLC. All Rights Reserved.

Table 2. AVAILABLE PART NUMBERS

Part Number	Description
AP1302CSSL00SMGA0−DR	6.5x 6.5 x 0.8(mm) VFBGA Package Part
AP1302CSSL00SMGAH3−GEVB	Product demo board
AGBAN8CS−GEVB	Dual camera adapter board

GENERAL DESCRIPTION

The **onsemi** AP1302 is a high−performance, ultra−low power in−line, digital image and video stream processor with advanced DSC and DVC features, supporting image resolutions up to 13 megapixels (4224x3156) and frame rates up to 13 MP at 30 fps, 1080p at 120 fps, and 720p at 240 fps. AP1302 supports various advanced features such as face detection. It enables full auto−functions support (AWB, AE, AF) to produce DSC quality images. The AP1302 supports both sub−LVDS (MIPI) sensor (input) and host (output) interface. MIPI D−PHY version v1.1 and CSI−2 version v1.1 is supported.

AP1302 interfaces to CMOS imaging sensors and performs all the necessary operations required to capture state−of−the−art 13 MP images including JPEG compression, capture 1080p video streams, and preview generation. AP1302 is optimized for operation with AR1335 and other **onsemi** sensors with MIPI interface (contact local **onsemi** FAE for details).

AP1302 is designed to support **onsemi**'s innovative and advanced sensors with Clarity+ Pixel technology, which increases the low−light sensitivity and performance of the camera system. AP1302 is also designed to support **onsemi**'s sensors that provide video interlaced HDR (iHDR) capability (i.e.. AR1335), allowing capture of high dynamic range scenes at video recording speed.

An on−chip RISC processor and ROM allow it to be operated with minimal external control. All advanced features are supported with on−chip memory and do not require any external full or partial frame buffer.

FUNCTIONAL OVERVIEW

Figure 1 shows the typical configuration of the AP1302 in a camera module. The AP1302 captures the data from primary or secondary sensor (one at a time) through two MIPI receivers, processes the image, and sends the processed data to the host. The AP1302 uses sensor interface 0 to connect to the primary sensor. Sensor interface 0 can use up to four high−speed sub−LVDS MIPI data lanes capable of transfer speed up to 1.2 Gbps each. The second optional sensor is connected through sensor interface 1 using up to three sub−LVDS MIPI lanes. The I^{2}C master interface is used to control the primary and secondary sensor operation.

The host uses I^{2}C (Standard, Fast Mode, FM+ and HS supported) or SPI Slave (up to 48 Mbps) interfaces to access the AP1302 internal registers and control the operation of the AP1302. Image data is transferred using four MIPI data lanes between the AP1302 and the host processor.

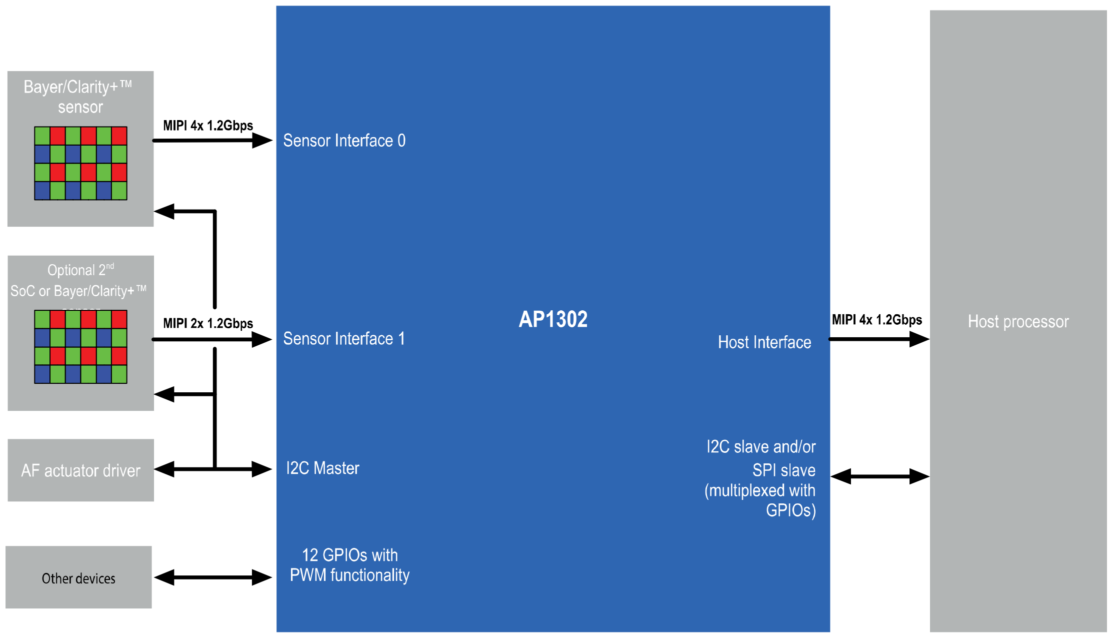


Figure 1. AP1302 Connectivity

Imaging sensors are controlled through the I^{2}C Master, which could also be used to control various types of auto−focus (VCM) drivers. There are also 12 GPIO signals that can be used to control other optional devices. Please contact local FAE support for list of supported devices and possibility to add support for new devices.

Figure 2 shows the 3D bridging application connectivity. In this application, the AP1302 is connected to two identical sensors processing two independent streams simultaneously and sending processed data to the host. The Max width of each sensor is 2112 pixels. The first sensor is connected to sensor interface 0 using up to 4 MIPI lanes and is used for single sensor snapshot or as a left sensor for 3D bridging application. The second sensor is only used as a right sensor for 3D bridging application. Note that the AP1302 only does the processing of two image streams from two sensors and does not merge the streams into one 3D video stream. 3D merging is done on the host; the AP1302 only performs bridging function. Two I^{2}C master interfaces are used to control the two sensors in the 3D bringing application. Both sensors need to be of the same type. The AP1302 will issue exactly same commands and send them synchronously via two I^{2}C interfaces to the sensors. In this way the operation of the two sensors will be kept synchronous up to ±1 line.

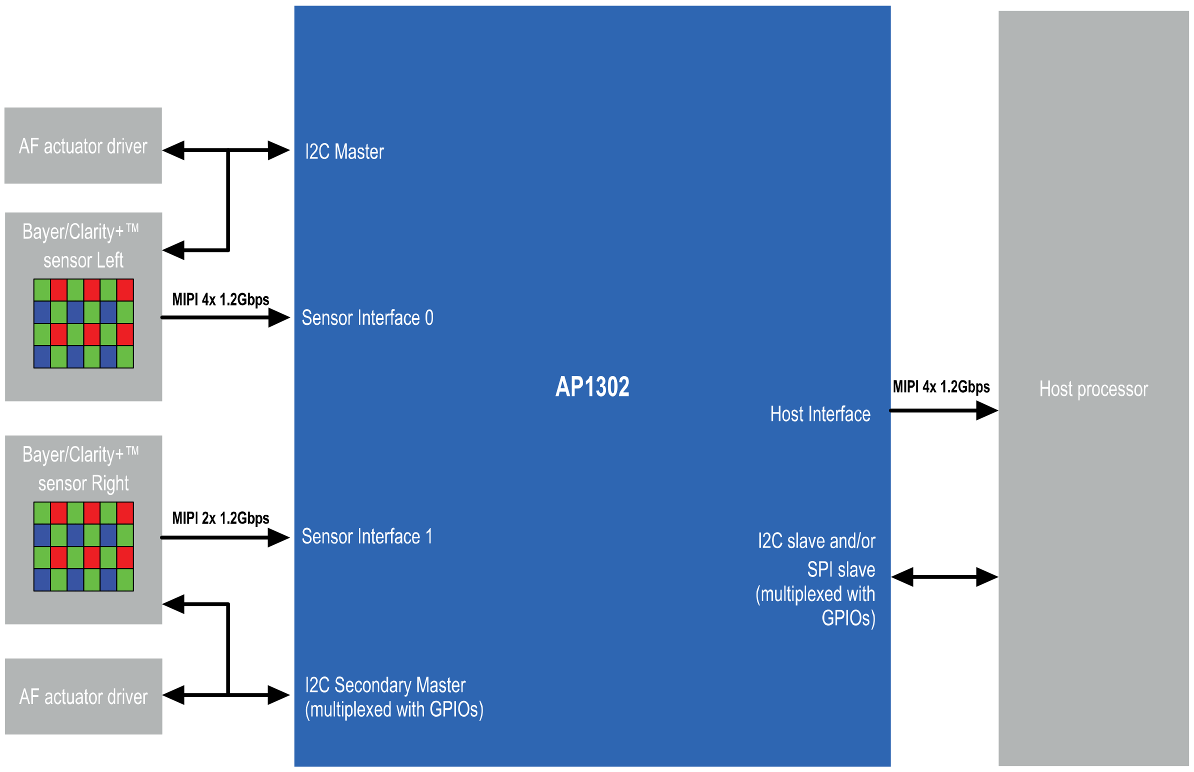


Figure 2. AP1302 3D−Assist Camera Connectivity

Embedded 32−bit CPU is used to run the automatic functions and system control implemented as firmware executed from on−chip ROM and/or RAM. Automatic functions are used for determination of proper exposure (Auto Exposure (AE)), determination of optimal dynamic range mapping function (Adaptive Tone Mapping (ATM)), detection and correction of the light source illuminant (Auto White Balance (AWB)); detection of focal distance and the lens system control (Auto Focus (AF)) and detection and compensation of flickering light source (Flicker Detection (FD)). These auto functions are implemented in firmware running on the 32−bit CPU. System control firmware is used to control the hardware and system resources and provide an interface to the host processor for the configuration and control of the chip.

Advanced features such as face detection are executed using a combination of hardware and programmable engine to maximize performance and flexibility. Image data and statistics required are stored on−chip. Production calibration data can be stored in one−time programmable memory (OTPM) inside the sensor.

FUNCTIONAL DESCRIPTION

Figure 3 shows the internal structure of the AP1302:

- Image Processing sub−system
- Statistics engine
- Control processor subsystem
- Advanced features processor subsystem
- Sensor interface
- Test pattern generator
- Host interface
- Clock and Control

- Production test and debug

The image processing and image enhancement core of the design includes the image preprocessing, color processing and pixel processing engines, the statistics gathering engine, and the JPEG encode engine.

The advanced feature processor includes the 32−bit processor, as well as its associated program and data memories. The advanced feature processor is part of the chip data path and can operate on and manipulate pixel data.

The processor subsystem includes a 32−bit processor to perform all imaging sensor control, control of external focus and zoom actuators, all on−chip real−time control, and to present a simplified API to the host interface in form of a basic register set. All program code for the processor, as well as the settings for imaging sensors and actuators is stored on−chip ROM memory. The processor subsystem also includes a scratch pad RAM.

AP1302 supports sub−LVDS data interfaces to the sensor and the host. I^2C master (also referred to as SIPM) is used to control the sensor and the AF driver while I^2C slave (also referred to SIPS) allows for host control of AP1302. The I^2C master interface may be optionally used to interface to an external EEPROM device. This device has 12 ports that may be configured as GPIO and can be used to implement PWM, to control mechanical shutter, zoom or the secondary sensor. All IO cells are fail−safe.

An on−chip, low−jitter PLL with input clock of 10 MHz – 54 MHz and a combination of clock dividers is used to generate all the clock signals and frequencies required for the chip core and interfaces. The core of the design is completely synchronous and operates as a single domain. Control circuitry and gating is employed as appropriate to control switching and therefore power dissipation. The maximum operational frequency of the device is 450/2 = 225 MHz.

AP1302 uses an external 1.8 V supply for its interfaces and a 1.2 V (nominal) external supply for the core.

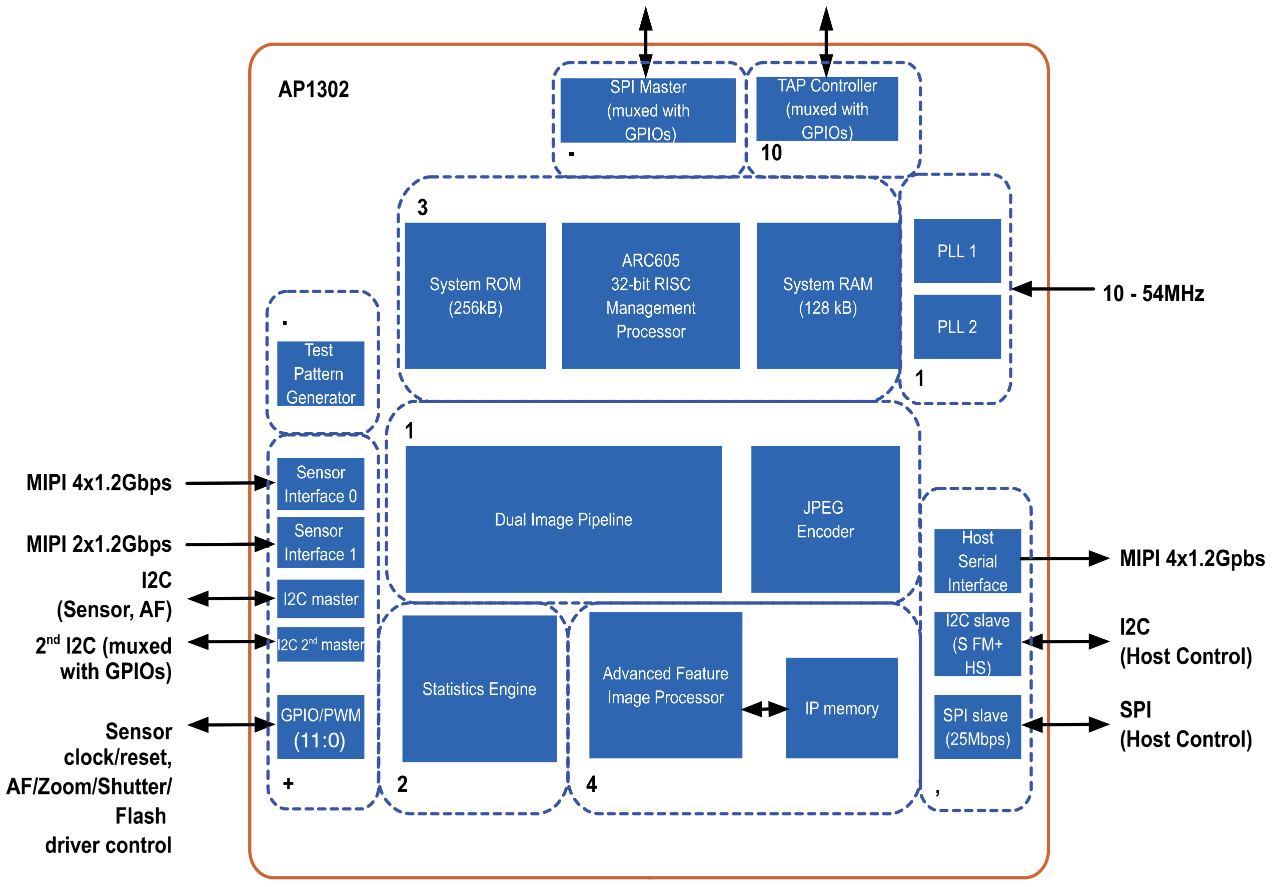


Figure 3. AP1302 Internal Block Diagram

Image Processing Pipeline and Image Statistics Engine

The image processing pipeline includes:

- Color Processing (CF): black level correction, RGB gains, lens shading and color shading correction, GrGb global correction
- Bad Pixel Correction (BPC): bad pixel correction and Bayer de−noise engine (two 5x5 kernels one for each T1/T2 iHDR plane)
- Interleaved High Dynamic Range (HDR): reconstruction of high dynamic range image (14 bits)

out of two 10bit planes, one with long exposure time (T1) and one for short exposure time (T2), motion compensation module

- Pixel Processor (PP): GrGb local correction, de−mosaic, luminance de−noise and sharpening (7x7 kernel)
- Denoise/Sharpening: de−noise and sharpening in RGB/RCB domain (28x28 kernel for chrominance, 7x7 for luminance), chroma coring, anti−fringing
- Pixel Mixer (PM): flare removal, color correction, gamut compression, tone mapping
- Gamma: very precise gamma (exponential) engine
- Curves: post gamma S−curve LUT (32 points)
- Special Effects (SFX): special effects module for color transformations (like sepia) and edge based transformations (like emboss)
- Preferred Color Rendering (PCR): preferred color rendering
- Resample: scale up using bilinear interpolation and/or scale down using averaging algorithm. Max output size limitation:
  - Resample 0: 4224x3156
  - Resample 1: 2304x1296
  - Resample 2: 428x240
- Color Conversion (CCONV): conversion of spaces from YUV to RGB or JFIF version of YUV suitable for JPEG compression
- Color Formatting (CF): Formatting the output stream, RGB565, RGB555, RGB444, YUV422, different orders of color components

The image processing pipe including the JPEG engine is duplicated, which means that there are two instances of each processing block. Incoming frame is divided into left and right halves with 16 pixels of overlap. Each instance up to the JPEG engine does the processing of one half of the image. The image halves are then merged and the merged image is sent to the host interface directly or compressed in the JPEG engine. In case JPEG compression is used, the image is first reordered to form JPEG MCU blocks. All even blocks are processed by one JPEG instance and all odd blocks are processed by the other JPEG instance. Again the outputs from the two JPEG instances are merged before the JPEG stream is sent to host interface.

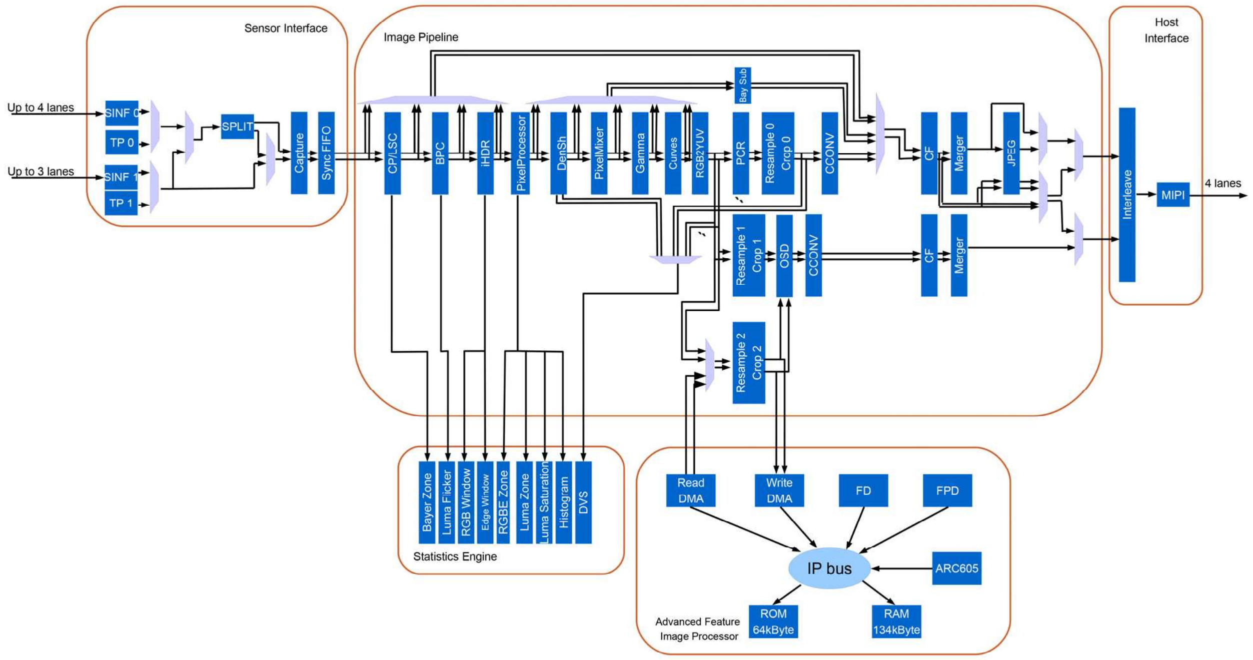


Figure 4. AP1302 Image Pipeline Block Diagram

Max frequency of the IPIPE engine and JPEG is 225 MHz, which means that maximum pixel rate into the IPIPE is 450 Mpix/sec (two instances of the IPIPE working in parallel). Each block in the IPIPE has an acknowledge signal which can be used to stall the incoming stream of pixels coming from the block in front. Similarly each block in the IPIPE can be stalled by the block positioned later in the IPIPE. Blocks have different delays, blocks which has to form the kernels of pixels contains memory for holding several lines of the image (called line buffers) and this way the delay of such block can be several lines. The blocks up to resample blocks do not affect the data rate.

Resample then can reduce or increase the pixel rate on its output (depending whether it is operating in scale down or scale up mode). If operating in scale up mode and is the input pixel stream is at its maximum for a given frequency (meaning that every clock a pixel is received on the resample input), resample will then lower the acknowledge signal and this way block the incoming stream to be able to output the generated pixels. As a consequence, the Sync FIFO module will have to buffer the data which are coming from the sensor. Sync FIFO has capability to buffer one line of data.

This effectively means that it can buffer the data over one line time (which is measured from start on the line to the start of the next one). If sync FIFO gets overflowed, this is a recoverable error and it is reported in warning register (this means that ISP will still be running, but image is corrupted).

Another such bottleneck is the host MIPI interface, which can run at a different clock frequency than IPIPE and this way the pixel rate can be lower (or higher, but this will not cause any issues in terms of data flow).

At the end of the image pipe there are three resample (scaling) and crop engines. Two are used for for primary (main) and secondary (auxiliary, bubble) output image. The third one is used for Advanced Image Processor features (e.g. face detection).

Black Level (BL)

Sensors typically provide predefined data pedestal for zero signal, to prevent noise clipping. AP1302 can subtract that black level and optionally provide noise filtering (on negative data) to prevent clipping. Optionally, black level can vary based on the scene conditions, for example, a lower black level for low light conditions.

Lens Shading Correction

Lens vignetting and optical crosstalk can produce reduced signal levels near borders and/or corners of the image. The AP1302 lens shading correction (LSC) module generates a gain map for compensation of lens vignetting (shading) and pixel crosstalk. Four gain maps are generated, one for each CFA channel.

The gain map curve is obtained using interpolation between reference points−vertices. These are equidistantly placed along the horizontal and equidistantly along the vertical axis, forming an $n \times m$ vertex matrix. The n and m are adjustable and the maximum number of vertices is determined by hardware resources; typical setting would be: $n = 7$, $m = 7$. Some of the vertices fall outside of the image coordinates and can be adjusted to tune the corner gain map.

Bad Pixel Correction

To increase sensor production yield, certain defective pixels are tolerated. The bad pixel correction block uses a 5 x 5 kernel and is able to remove the following defects:

- Single physical pixels inside 3 x 3 physical (and limited amount of bad pixels inside 9 x 9 area)
- Single pixels in same color channel inside 5 x 5 physical area
- Double pixels in same color channel inside 5 x 5 physical area

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

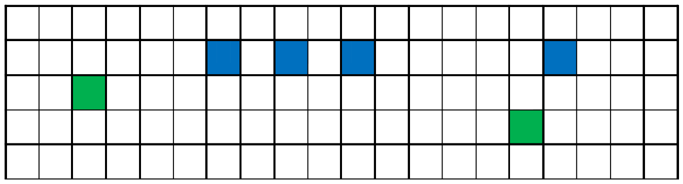


Figure 5. Single Physical Bad Pixel Defect Examples

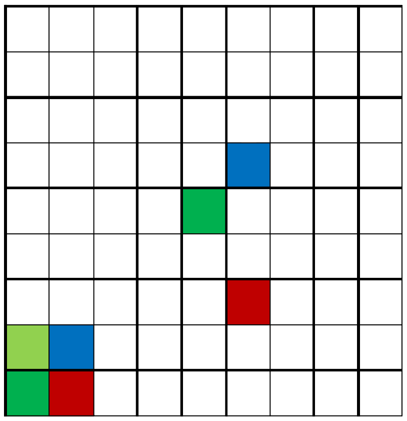


Figure 6. Single Same Color Channel Bad Pixel Defect Examples

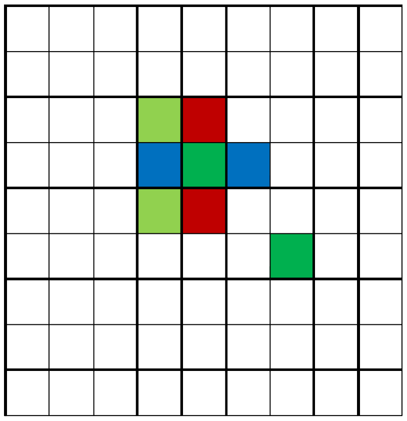


Figure 7. Double Same Color Channel Bad Pixel Defect Example

AP1302 provides several methods for defining the defective pixels:

- Automatic detection
- Single physical defects
- Single Bayer defects
- Double Bayer defects

Bad pixel detection strength is controlled by absolute and relative pixel value threshold along with bounded flat field detection. Furthermore detection of hot and cold defective pixels can be biased. All of these parameters can be adjusted based on the light level.

The detection is sensitive to gradients and can be configured to detect various natural image detail, including highlights. Detection and correction of single physical pixels is generally better than that for those in Bayer plane.

Pixel Processor

Due to crosstalk and other effects, green pixels in red and green pixels in blue rows can slightly differ. This step compensates for these imbalances locally, using 7 x 7 kernel. The algorithm is similar to flat field correction, but the AP1302 algorithm is also capable of compensating for the

GR−GB disparities efficiently on edges and high resolution detail.

Demosaicing

Demosaicing is one of the most important steps for achieving the high−quality images.

Due to the sensor's physical limitations, only one color channel is captured at the same pixel location. In almost all sensors today, the image input data is captured in Bayer color format:

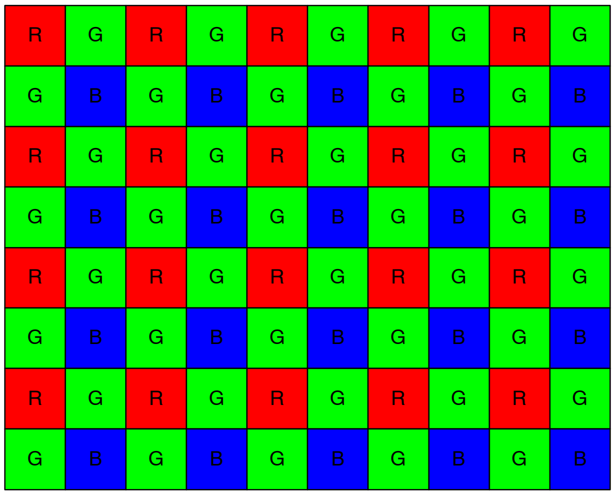


Figure 8. Bayer Array (CFA)

AP1302 uses a 7 x 7 region to interpolate the missing colors. The algorithm tries to preserve all the edges and details and can be tuned to handle various module inefficiencies, which could influence the demosaicing (crosstalk, chromatic aberration, and so on).

Denoise

The amount of random noise increases with shorter exposure times and higher analog or digital gains as well as module form factor. Effective noise removal is required to render visually pleasant images.

AP1302 has implemented the following denoise schemes:

- Bayer filtering (to remove sensor readout related noise and artifacts) occurs during BPC correction and is therefore named BPCDEN; it uses a 5 x 5 kernel. Use parameters starting with BPCDEN_ and BPCTRI_ to tune the module.
- Edge−adaptive multifrequency luminance sigma filter (to control the edge SFR). Uses 7 x 7 kernel. Use parameters starting with PM_ to tune the module.
- Luminance Edge detection and reconstruction in the denoise/sharpening − (DENSH) module. Uses 7 x 7 kernel.
- Edge−adaptive multifrequency chrominance filter. Uses 28 x 28 kernel. Use parameters starting with PM_ to tune the module.
- High performance edge−adaptive chroma filtering (to remove color random noise). Use DENSH_ parameters to tune the module. Uses 7 x 7 kernel.

White Balance

White balance adjust white point to match the sRGB white point.

Sharpening

The lens system together with crosstalk modify the edge SFR, visually reducing the sharpness. There are two blocks that try to compensate for such effects:

- Edge−adaptive multifrequency chrominance filter. Uses 28 x 28 kernel. Use parameters starting with PM_ to tune the module.
- High performance edge−adaptive chroma filtering. Use DENSH_ parameters to tune the module. Uses 5 x 5 kernel.

Illumination Based Color Correction and Saturation Control

This block ensures correct color rendering for the given scene using 3 x 3 illuminant−dependent color correction matrix. The dark areas and highlights are carefully rendered to provide visually pleasant saturated areas.

Tone Mapping

Tone mapping implements several techniques to capture and map the high dynamic range scenes to display:

- ATM (adaptive tone mapping) adjusts the tone mapping curve according to the scene
- LTM (local tone mapping) locally adjusts contrast in order to provide more details in dark and bright areas

Gamma Correction

This block converts the color space by adjusting the gamma in very high precision. Typically the conversion is done to sRGB color space.

Curves

To ensure custom gamma, tuning curves can be applied independently to each color channel. This block can also bypass gamma correction and provide 32 configurable (either equidistant or with higher density in low−key) interpolation points.

Preferred Colors

This block can remap colors based on a custom color map. This is especially useful for mapping colors like sky and foliage to their memory color locations.

Color Conversion

AP1302 uses 3x3 configurable color conversion matrix to output the image in the desired format. By default the following output formats are supported:

- sRGB (identity)
- JFIF YCbCr
- CCIR601 YCbCr

Resampler and Crop

This block will crop the input image so as to match the proper aspect ratio and settings. It will as well perform the appropriate downscaling or upscaling so as to match the target output size.

When upscaling this block will output more data than what is coming in, therefore must properly set the frame rate to prevent any overflow.

There are three instances of this block: two for primary and secondary image streams and additional one for advanced features block (e.g. face detection). Figure 9 shows sample usages:

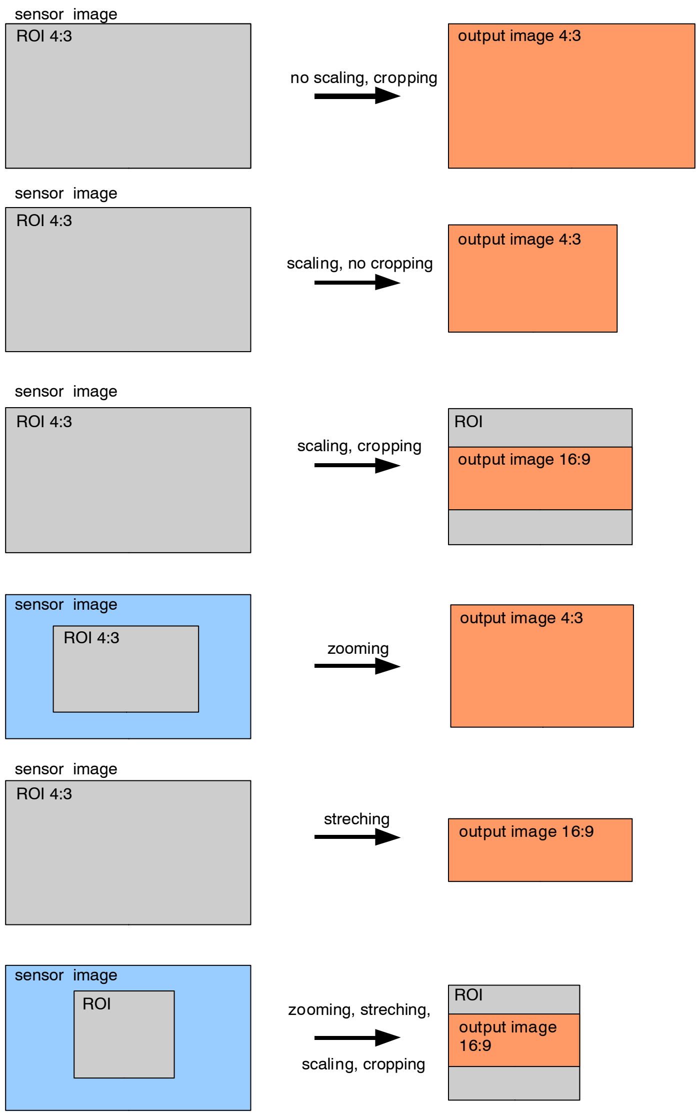


Figure 9. Sample Usages

JPEG Compression Engine

The JPEG engine in AP1302 is composed of two identical instances. An image is first broken to even and odd MCU blocks. All even MCU blocks are processed by one JPEG instance and all odd MCU blocks are processed by the other JPEG instance. Output streams from both engines are then merged together to form a proper JEPG stream.

Restart markers can be enabled. Only restart marker interval of 1 MCU is supported, which means that restart markers are inserted after every MCU. SCALADO speed tags are also supported. When SCALADO speed tags option is turned on, restart marker insertion needs to be disabled.

In 3D bridging application, JPEG instances can be used to compress frames from both sensors independently forming two separated streams.

Statistics Capture

The statistics engine gathers various statistical data about the image from various parts of the image processing pipe. Most of the gathered data is transferred directly to the main CPU memory for further processing using DMA. Part of the data is stored in statistics engine registers and the CPU needs to read them. Statistics data is used by the auto function running on the main CPU. Typical statistical data include:

- AE statistics
- AWB statistics
- Flicker statistics
- RGB zone statistics

Face Detection

Built−in face detection module detects faces on a frame−by−frame basis. Firmware then uses this information to enumerate and track faces. This information in turn helps auto−functions:

- Auto Exposure to properly expose faces.
- Auto White Balance to improve the image colors.
- Auto Focus for faster and more accurate focusing.

Face detection can also be used to automatically zoom into and track faces.

Faces can be detected at speed up to about 30FPS, although detection rate will reduce if smaller faces need to be detected.

Special Effects

The following effects can be optionally enabled in AP1302:

- Normal
- Alien
- Antique
- Black and White (B&W)
- Emboss
- Emboss color
- Grayscale
- Negative
- Blueish
- Greenish
- Reddish
- Posterize1
- Posterize2
- Sepia1
- Sepia2
- Sketch
- Solarize
- Vivid

Auto Image Control

Auto image control functions include the 3As (AE, AWB, AF), and flicker detection, adaptive− tone−mapping (ATM), local−tone−mapping (LTM), and so on.

Auto White Balance

In auto white balance mode, the algorithm uses on−chip RGB zone statistics to detect the illuminant or white point and compensate for it. There are 16 x 16 programmable zones that provide RGB statistics. The algorithm determines for each zone if the statistics would be acceptable or not.

Generally, the algorithm rejects those zones that are determined to be of strong color. To make this determination, the AWB algorithm uses a set of calibration data from the calibration profile, which consists of a set of measured white points from standard illuminants, and bounds for each illuminant that define the region of acceptable colors. Using RGB statistics from the accepted zones, the AWB algorithm estimates a white point for the image. For images where none of the zones are accepted, for example, images that consist entirely of strong colors, the algorithm would decide that there is no white point, and it will use history from previous frames as the white point for the current image. In addition, the algorithm uses information of local scene classes to help determine the white point. The algorithm can accept green and color temperature offset from the calibration profile for each of the standard illuminants. As a result, it is possible, for example, to make the images “warm” for tungsten lighting and neutral for other illuminants.

In manual mode, illuminant or white point can be specified. Additionally ‘measure white−point’ functionality is supported, which performs a gray−world algorithm on the input scene.

Both the auto and manual white settings control the color matrix. In the calibration profile, a calibrated color matrix is defined for each of the standard illuminants. The color matrix that is used for any particular moment in the pipeline is a result of interpolation based on the color temperature for that moment.

Auto Exposure (AE)

Images may be captured under various light conditions. Exposure time must match lighting conditions and has to put the sensor in the most sensitive acquisition range.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

Both auto exposure and manual exposure are supported. A rectangle of interest may be specified for AE and is fully programmable. AP1302 supports different metering modes.

- Center−weighted metering mode where the scene brightness is calculated by taking the average luminance value at the center of the frame (divided into equal zones) and the objects in the center are properly exposed.
- Matrix metering mode splits the frame into many segments and measures exposure value based on the average luma of each segment and its respective weighting.
- Spot metering mode is predominantly utilized in high contrast scenes where the main subject in center of the frame is properly exposed. The window specified is smaller than that of the center− weighted mode.

  The algorithm for AE is configurable and is executed by embedded CPU.

Flicker Detection

Flicker detection algorithm can automatically detect 50 Hz and 60 Hz flicker and compensate for it. The detection time is programmable.

The flicker can also be removed manually.

Auto Focus (AF)

If the module has an AF actuator, AP1302 can drive it through an external driver. The AF algorithm uses programmable zones to detect focus position in the scene and can support a wide variety of actuators, including voicecoil, MEMS, liquid lens, Heliomorph, and so on. The algorithm can perform a single or two−pass search to find the best focus position. In manual mode, arbitrary position can be set.

In case of reliable actuator AP1302 can also perform continuous AF.

SYSTEM OPERATION

This section describes the operation of the AP1302.

Boot Procedure

After power up host needs to load FW patch along with configuration, calibration and tuning settings to AP1302 in order to start frame processing. FW patch and settings are divided into high priority and low priority group:

- High priority group contains essential FW functions and settings needed for basic system operation and needs to be loaded before frame processing is started. Frames processed with only high priority group loaded are not exhibit best image quality, but the frames are of correct size and timing so host can utilize them order to synchronize the receiver to the AP1302 output stream.
- Low priority group contains patches and settings that are not influencing the basic operation of the chip, instead they contain functions and settings that ensure best possible frame processing with highest image quality. Low priority can therefore be loaded after frame processing is already in progress.

  All this is compiled into single continuous binary called bootdata. Host does not need to be aware of these steps or the content of bootdata.

Boot Procedure Using Bootdata

Bootdata is a compressed binary that contains all the necessary configuration to get AP1302 up and running with highest image quality:

- Patches (high and low priority)
- Configuration (output size, format, etc.)
- Tuning
- Calibration

Bootdata is provided in hex format as part of the configuration XML (generated by build script provided by ON Semi). Bootdata is content of a section with a tag <dump> with the ”name” attribute ”bootdata”.

All data inside bootdata content is compressed in order to reduce the loading time. AP1302 decompresses and interprets data on the fly as it receives it. Since there is no need to send the register address when loading register values, this further reduces the loading time. Below is an example of bootdata XML content:

```
<dump     name="bootdata"     crc="0xf0c7"     checksum="0xffff"     pll_init_size="4094"
system_freq_in="0x00320000" scratchpad_size="0x1cf4">
  05000014 00000000 00000008
  6029 0002 00010000
  03000810 00000000 00005a60
  38000008 e4303976 4086a501 b20cf035 e10a0704 0dd007ba 274410f2 7f00f4fb
  f8b01017 208f849a 840a367a 0a586421 0c0380b4 7802fe0d 00402eea 5b11c4d6
  5c972058 038b20d7 b05487c5 6523da13 823f1d83 886fbff0 0913c8ef c40939ba
  ...
  545c 00a0
  6011 0003 b150 b598 bccc
  00000000 06134159
</dump>
```

AP1302 initialization procedure is as follows:

1. After power up sequence, poll chip_version_reg until value 0x0256 is read.
2. Set system_freq_in (input clock) and hinf_mipi_freq_tgt (target MIPI output clock) registers according to your system requirements.
3. Optionally, host can also set the following registers:
   – primary_sensor_sip
   – secondary_sensor_sip
   – sensor_select
   – preview_hinf_ctrl
   See ”Boot time configuration adjustments” below for details.
4. Load bootdata binary content starting from address 0x8000, wrapping around at 0x9FFF until finished. In other words: load bootdata in 8KB chunks. each time starting at address 0x8000.
5. Write 0xFFFF to bootdata_stage register to inform AP1302 that bootdata was fully loaded.
6. To confirm that AP1302 has been successfully initialized, poll bootdata_stage register until value reads back as 0xFFFF. If there was a problem, then bootdata_stage will not be updated. In this case check error register and other debug registers. See developer guide for more details.
7. Optionally reconfigure width, height and any other settings and enable MIPI output if it was disabled earlier.

Boot Time Configuration Adjustments

Bootdata binaries come pre−configured for the most common hardware configuration (as seen on reference schematics in Appendix A) and firmware can do some level of auto−detection for few variations. User also has an option of modifying relevant registers before loading bootdata. This way of non−standard hardware wiring configurations can be supported, or overhead of auto−detection can be avoided without having to modify the bootdata binary itself.

To achieve this, reset, clock, I2C IDs and number of MIPI lanes (between sensor and AP1302) can be explicitly set with the following registers:

- primary_sensor_sip (to configure I2C bus, slave ID and number of MIPI lanes)
- secondary_sensor_sip (same for secondary sensor, if applicable)
- sensor_select (to configure which clock and reset signals need to be enabled)

By default, AP1302 will start streaming with all four MIPI output lanes as soon as possible (even before bootdata binary is fully loaded). If less than four lanes are desired, or if user wants to disable MIPI output altogether until after the boot, then preview_hinf_ctrl register can be set accordingly prior loading bootdata.

Disabling the MIPI output (preview_hinf_ctrl = 0x10) before boot also allows host to adjust width, height and other settings after boot but before actually enabling the MIPI output.

By disabling the output host can also avoid first few frames with suboptimal image quality, since AWB and AE will not be accurate without few frames of history.

Output Size, Cropping and Scaling

The first choice that user typically needs to make is the output resolution. Independent of the sensor resolution, AP1302 will scale full sensor FOV to the selected resolution, configured with preview_width and preview_height registers. This can be either downscaling or upscaling (if the output resolution is larger than the sensor).

If the aspect ratio of the sensor doesn’t match the configured AP1302 output resolution, then image will be center cropped in either vertical direction (maintaining full horizontal FOV) or horizontal direction (maintaining full vertical FOV) as appropriate to maintain the correct aspect ratio of the final image.

If additional cropping is desired then digital zoom, pan and tilt functionality can be used (dz_tgt_fct, dz_center_x, dz_center_y). Note that, in AP1302 terms, using digital zoom does not necessarily imply upscaling. Upscaling is only necessary when number of physical sensor pixels within the selected region is lower than the configured AP1302 output size.

When higher frame rate or reduced power consumption is required, one can use one of the various subsampling modes offered by the sensor (preview_sensor_mode). Using sensor subsampling modes typically mean inferior image in terms of details, but if final output size is considerably smaller (scaled down by AP1302), then difference is negligible. Note that selecting different sensor mode does not affect AP1302 output size or FOV since scaling is automatically adjusted.

Here are few example configurations:

Table 3.

Output Size	Digital Zoom	Sensor Readout Mode	Actual Sensor Readout Size	Actual Scaling Factor
preview_width x preview_height	dz_tgt_fct	preview_sensor_mode	sensor_x0, sensor_x1, sensor_y0, sensor_y1	out0_sf_x, out0_sf_y
1920 x 1080	1.0x	4208 x 3120 (AR1335 Native)	4208 x 2368 100% HFOV 76%VFOV	1920/4208 = 2367/1080 = ~0.456x
1920 x 1080	1.0x	2104 x 1560 (AR1335 Bin2)	2104 x 1184 100% HFOV 76% VFOV	1920/2104 = 1184/1080 = ~0.913x
1920 x 1080	1.234x	4208 x 3120 (AR1335 Native)	3410 x 1918 81% HFOV 61.5% HFOV	1920/3410 = 1080/1918 = ~0.563x
1920 x 1080	1.234x	2104 x 1560 (AR1335 Bin2)	1706 x 910 81% HFOV 61.5% HFOV	1920/3410 = 1080/1918 = ~1.126x (upscaling)
1920 x 1080	2.1917x	4208 x 3120 (AR1335 Native)	1920 x 1080 45.5% HFOV 34.5% VFOV	1920/1920 = 1080/1080 = 1.0x
500 x 500	1.0x	4208 x 3120 (AR1335 Native)	3120 x 3120 74% HFOV 100% VFOV	500/3120 = 500/3120 = ~0.16x

Contexts

When changing multiple parameters at once it is generally recommended to use atomic register (see Register Reference) to apply multiple settings at once. For historic reasons, AP1302 also offers 3 sets of registers, referred to as contexts, that can be selected with a single write to a ctrl[cntx] register. Despite the name (preview, snapshot, video), there is no functional difference between the contexts. For simplicity, it is recommended to always use preview context and utilize atomic register functionality to switch between different output configurations.

Bubble

In addition to primary (main) image stream, an additional secondary (bubble) image stream can be generated from the same input image stream received from sensor (preview_out_fmt[iis]). Bubble image is outputted simultaneously with main image through the same MIPI interface. It has independent size (bubble_width & bubble_height), output format (bubble_out_fmt) and independent zoom configuration (bubble_dz_tgt, bubble_dz_center_x, bubble_dz_center_y). Reference FOV can be relative to the primary stream or full sensor FOV (bubble_out_fmt[roi_extend]). Here are few example use cases where bubble image can be used:

Videoconferencing applications with one video stream to display full FOV and another to focus on the faces.
Machine vision applications where smaller video stream is needed for AI and another, high resolution video stream for the user.

Machine vision applications where smaller full FOV video stream is needed for object detection and another independent stream is used to zoom into any detected object.

Output Formats

AP1302 supports number of different output formats (preview_out_fmt):

- JPEG 422
- JPEG 420
- YUV 422
- YUV 420
- YUV 400 (monochrome)
- RGB 888
- RGB 565
- RGB 555
- Bayer−8/10/12/16 bypassed or processed

JPEG

JPEG is the most bandwidth efficient output format supported by AP1302. Fully formatted JFIF output with all the necessary headers and footers, including EXIF information, can be generated. JPEG can be based on either YUV422 or YUV420 data. With YUV420, maximum width is limited to 2112 pixels.

MJPEG

JPEG engine can also be used to generate MJPEG compressed video. While compression ratio is not as high as with H.264 or H.265, one can achieve less than 1 bit per pixel without obvious image quality degradation or even less for less image quality concerned applications. By default, JPEG is optimized for high quality still images. To optimize for video compression, reduce jpeg_ctrl[qual].

YUV

AP1302 supports 8−bit YUV data formats in either BT.601 or JFIF color space. Sampling can be:

- 4:2:2 (effectively 16 bits / pixel)
- 4:2:0 (effectively 12 bits / pixel)
- 4:0:0 (effectively 8 bits / pixel, luminance only)

RGB

RGB encoding formats are less efficient than YUV, but often more suitable for certain applications, like machine vision:

- RGB888 (effectively 24 bits / pixel, 8 bits per color)
- RGB565 (effectively 16 bits / pixel, 6 bits for green)
- RGB555 (effectively 16 bits / pixel, like RGB565 with green least significant bit always 0)

Bayer

Bayer formats can be used to get raw image sensor data. Supported bit depts are 8, 10, 12 or 16 bits per pixel, selectable independent of sensor bit depth. Image pipeline is bypassed, no image processing is applied, and image statistics are not collected. This also means that auto functions are not working.

It is also possible to generate Bayer formatted image from fully processed and scaled image. This can be useful when receiver does not support other, more efficient, formats. Auto functions are fully functional in this mode.

Status Structure

Along with the image data, AP1302 can append the status structure at the end of the frame, containing metadata of the current frame.

Status structure, when enabled, contains various metadata about the image and it is appended to the end of every frame sent out. Which metadata is part of the status structure and its format can be configured by the host using a simple description list. A status structure description list consists of entries, each entry contains a base address of a first register to be copied into the status structure and the number of bytes to be copied. Each entry also defines whether the base address is part of the Basic register address space or Advanced register address space. In case entry describes the Basic register space, the address is 16 bit and if the entry describes the Advances register space, the address is 32 bit. Description list ends with the entry containing all zeroes.

There can be up to four different lists defined. Status structure description lists should be placed into the Scratchpad memory. The address of the first entry for each list should be put into a one of the four ss_head_pointer basic registers. Each description list can be enabled individually with corresponding bit in preview_ss[enable] register. Figure 10 shows the status structure description list organization.

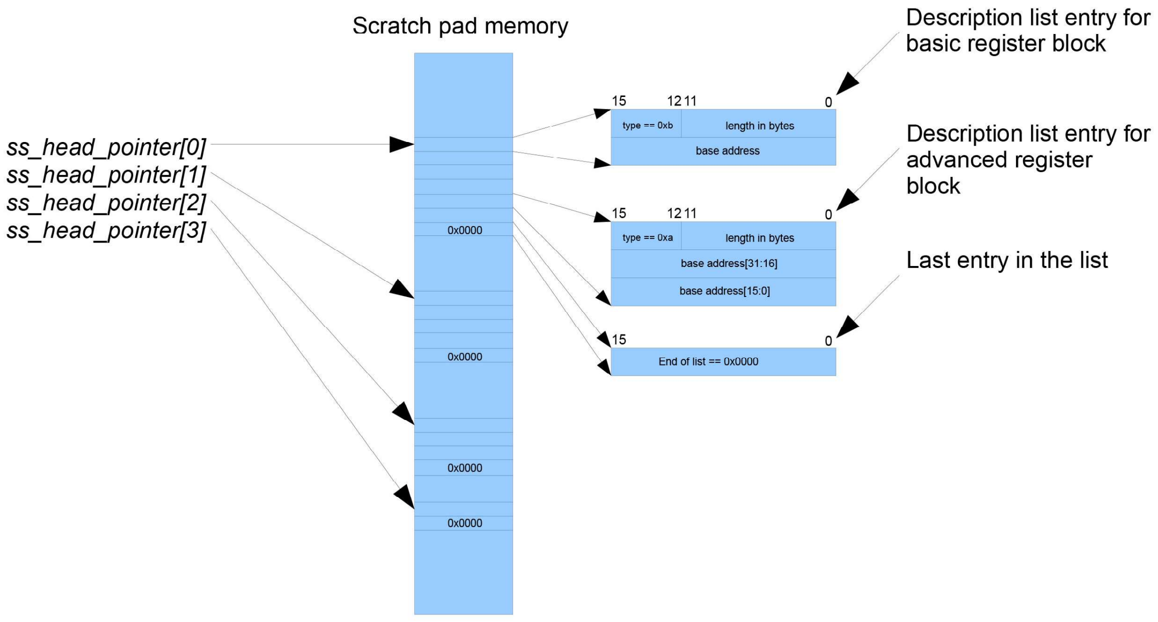


Figure 10. Status Structure Description Lists

Based on the status structure descriptor lists and other settings in preview_ss registers, the status structure is built. Every entry in a given list creates one record in the status structure. The records are concatenated one after another as they are organized in the lists and lists are organized from the ss_head_pointer[0] to ss_head_pointer[3] (depending on which list is enabled in preview_ss[enable] field). Additionally all records can preceded with SOSS marker (if enabled in preview_ss[soss]) and followed by records size (if enabled in preview_ss[size_e]), status filed (if enabled in _ss[status_e]) and EOSS marker (if enabled in _ss[eoss]). Status structure organization is shown in Figure 11.

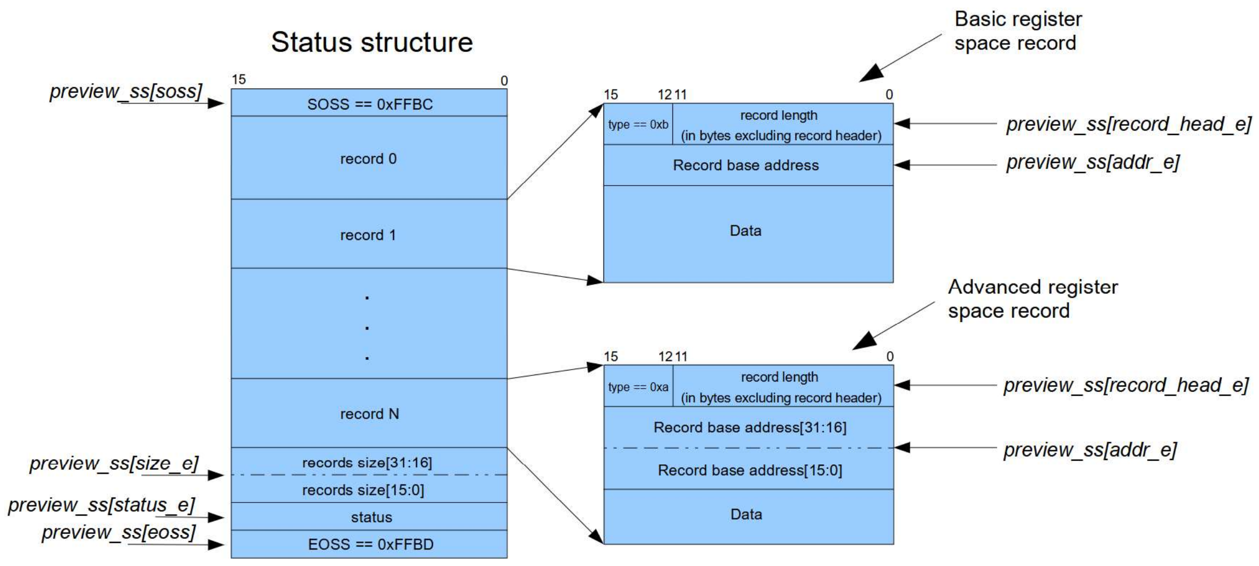


Figure 11. Status Structure

In any given output format, status structure is always aligned to the end of the frame (meaning that the last byte out of AP1302 is last byte of the status structure). In spoof modes, this means that certain padded data are inserted between the image data and the status structure if needed.

Few predefined status structure description lists are available in common profile (see Developer Guide for details on profiles). These can be modified or used as examples to define new ones.

Data Streams Interleaving

There are up to three data streams that AP1302 generates for each frame it receives from the sensor:

- Main image
- Bubble image
- Status structure

These data streams are transmitted simultaneously through a single MIPI interface. There are two methods to interleave the data.

MIPI Interleaving

MIPI interface specification defines Virtual Channel (VC) and Data Type (DT) as part of Data Identifier (DI) byte of MIPI packet header. This information enables receiver to de−interleave multiple data streams received though the same physical interface. This interleaving mode can be enabled by setting preview_hinf_ctrl[mipi_mode] bit. Each of the three data streams generated by AP1302 can have unique VC and/or DT. This is controlled with preview_mipi_ctrl and preview_mipi_ii_ctrl registers.

Note that preview_hinf_ctrl[spoof] bit should be kept at 1, so that main image stream packets are properly sized to maintain MIPI protocol compliance.

Spoof Mode Interleaving

This is an alternative method for transmitting multiple streams for receivers that cannot support interleaving at MIPI protocol level. Instead, a “virtual frame” can be constructed. From MIPI protocol perspective, this virtual frame looks like a normal frame, but the content of this frame is combined data from all the enabled streams. Zero padding is added as needed at the end of the image data to match the required frame size.

Receiver can decode the content as such:

- If bubble image is enabled, then image data from the two streams is formatted into special packets. These packets are prepended with a special 4−byte header consisting of ID and length, 2 bytes each. ID is 0 for main image and 1 for bubble image. Note that these are custom packets, unrelated to MIPI packets. These custom packets are within the payload of the MIPI frame.
- Status structure is always at the very end of the frame. Padding is inserted prior the status structure as required to ensure this.

Figure 12 shows an example of such a frame:

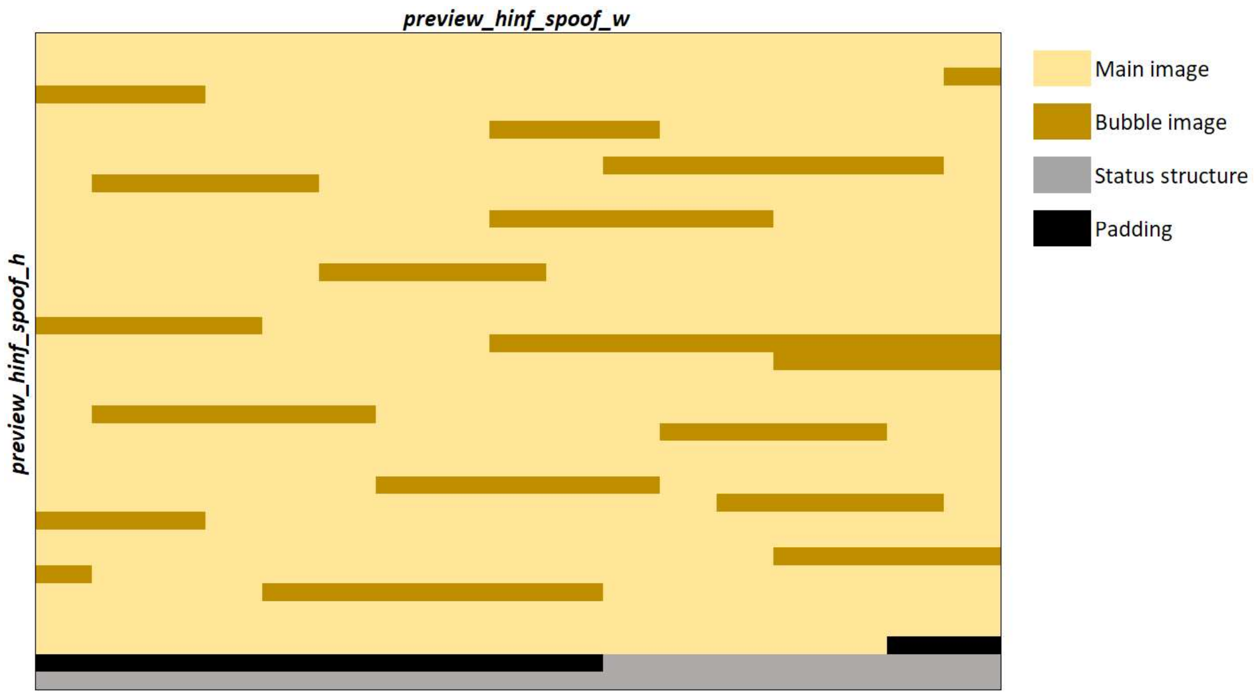


Figure 12. Spoof Frame Example with Main Image, Bubble Image and Status Structure

To enable spoof mode, first preview_hinf_ctrl[spoof] bit needs to be set. In fact, this bit must always be set to maintain constant packet size throughout the frame, as required by MIPI specification.

There are two parameters that user can control:

- preview_hinf_spoof_w (“spoof width”) defines number of bytes in MIPI image data packet payload.
- preview_hinf_spoof_h (“spoof height”) defines the number of packets.

In most cases, it is recommended to leave these parameters at 0, and let AP1302 firmware determine the most suitable values. In most cases, spoof width is set according to the main image width (width * bytes per pixel). There are two exceptions:

- With JPEG, spoof width is set to 2048B.
- If bubble image is enabled, spoof width is set to 2048B.

Spoof height is determined based on the amount of data generated each frame. With only the main image enabled, this means that spoof height matches image height. Except with JPEG, where the amount of data varies depending on the scene, so spoof height also varies on frame−to−frame basis.

It's generally okay to set a custom spoof width. But spoof height parameter needs to be used with caution − too big values can cause excessive padding. Because padding can only start after image is fully processed, it effectively increases vertical blanking requirements and limit the achievable FPS. This is especially challenging with JPEG output where amount of generated image data can vary widely.

Synchronization and Triggering

AP1302 implements very flexible synchronization scheme. Periodic synchronization signal connects to AP1302 GPIO. AP1302 will then measure the signal and synchronize sensor to it. AP1302 continues to measure both, synchronization signal and sensor output, and applies any adjustment necessary to maintain accurate synchronization.

Sensor synchronization is achieved solely through I2C communication. No sensor support for triggering is required. This means that synchronization can be achieved with any sensor supported by AP1302. The accuracy will vary depending on how accurate the sensor timing controls are, but most sensors support adjustment in pixel clock granularity, so sub−microsecond accuracy can be easily achieved.

This synchronization method is also independent of the oscillators used. Because timings are being actively measured and corrected, it does not matter if synchronization signal, AP1302 and sensor are running on separate oscillators.

Because the synchronization is not hardwired, it is possible to synchronize to arbitrary point in sensor integration and readout sequence. While typical sensor allows you to only synchronize to the start of integration or start of readout, AP1302 allows synchronization, for example, to the center of integration (trigger_ctrl[sync_mode]). This allows you to minimize any motion related differences between independent cameras with different integration time, different timings or even entirely different sensors. For example, one can synchronize rolling shutter with global shutter sensor.

In addition to synchronization, it is also possible to control integration time with pulse width or period (trigger_active_* and trigger_period_*).

Using the status structure, it is possible to append timestamp that corresponds to start of readout, start of integration or center of integration (timestamp_ctrl).

INTERFACES

The AP1302 has the following interfaces to interact with the external devices:

- Sensor Serial MIPI interfaces
  - Up to 4x 1.2 Gbps data lanes to receive data from primary sensor
  - Up to 3x 1.2 Gbps data lanes to receive data from secondary sensor
- Host serial MIPI interface (4x1.2 Gbps)
- I2C slave interface for host to control AP1302
- SPI slave interface for host to control AP1302 (optional)
- Two I2C master interfaces for AP1302 to control sensors, AF and other devices
- SPI master interface for future use
- GPIO/PWM controls

Sensor Interfaces

The sensor interface consists of two MIPI receiver ports. Each receiver has its own clock lane and they share 6 data lanes as shown in the Figure 12. Each lane supports up to 1.2 Gbps bandwidth. In normal operation mode, the primary and optional secondary sensor can be connected to the AP1302. The primary sensor can use up to 4 (in the order: S0_D0, S0_D1, S0_D2, S01_D32) data lanes and secondary sensor can use up to 3 data lanes – the last when not used by the primary sensor (in the order S1_D0, S1_D1, S01_D32).

In 3D bridging application, two sensors of the same type can be connected to the two receivers with the same order of data lanes. The left sensor is connected to the receiver 0 and the right sensor is connected to the receiver 1. In this case, the secondary sensor I^{2}C master is used to control the right sensor. This ensures that all the register writes to the sensor are executed synchronously while there is still opportunity to read the registers from the two sensors (like LSC configuration data from OTPM inside the two sensors).

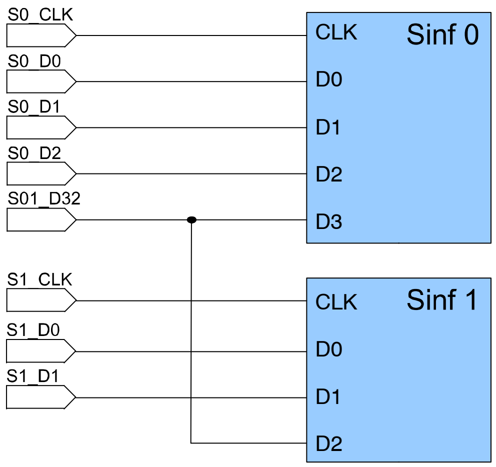


Figure 13. Data Lane Sharing Between Two Sensor Interfaces

Host Interface

The host interface consists of one clock and four data MIPI lanes capable of 1.2 Gbps each. When less than 4 lanes are required, data lanes needs to be disabled in order of 3,2,1 (that is, if just two lanes need to be used, these two lanes are 0 and 1).

Multiple data streams can be output thorugh the same MIPI interface as discussed in Data streams interleaving section.

I^{2}C Slave Interface

The AP1302 is controlled by the host through the I^{2}C or SPI interface where host processor is a master and AP1302 is a slave on the bus. The I^{2}C slave interface of the AP1302 can support Standard–mode (up to 100 kbps), Fast–mode (up to 400 kbps), Fast–mode Plus (up to 1 Mbps) and High speed mode (up to 3.4 Mbps).

The I^{2}C slave ID can be configured as:

- 0x78 (write) and 0x79 (read) or
- 0x7A (write) and 0x7B (read)

To select the ID AP1302 checks GPIO[11] right after reset:

- GPIO[11] == 0: ID = 0x78/0x79
- GPIO[11] == 1: ID = 0x7A/0x7B

Note that GPIO[11] can still be used for other purposes after I^{2}C ID was latched after reset. If used for other purposes, pull up/down resistor needs to be used to pull the GPIO[11] line up/down during and immediately after reset.

The I^{2}C slave has a digital glitch filter. This filter is controlled by SIPS_FILTER register. Filter can filter glitches out of SCLK signal based on median 3 filter (two input needs to be “1” to output “1”; otherwise output is “0”). SDATA signal has a programmable delay in order to avoid race conditions. Filter uses SYS clock which is beforehand divided. Filter is disabled after reset. Figure 14 shows the I^{2}C glitch filter scheme.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

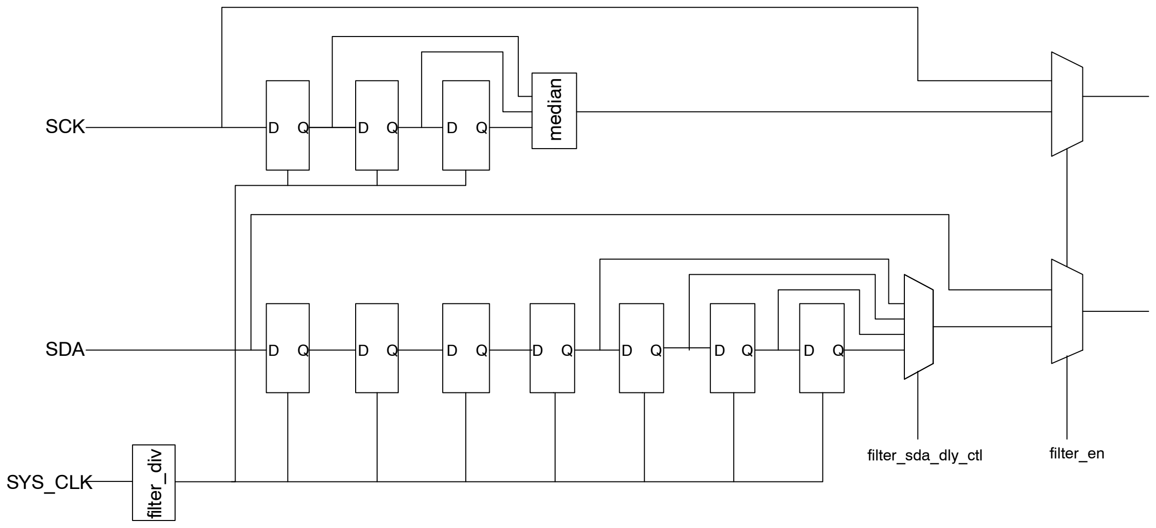


Figure 14. I2C Slave Digital Glitch Filter

Host sees the AP1302 through a 16−bit address space called Basic register space. Every I2C access starts with a 16−bit address, following by a read or write data transaction. Data transaction can consists of one or more bytes (each consecutive byte will be read from or written to incremented address from the address that was specified in the address transaction).

Advanced Register Space

Host can access Advanced register space indirectly through the Basic register space. 4 kBytes of Basic register space is mapped to Advanced register space as shown in Figure 15.

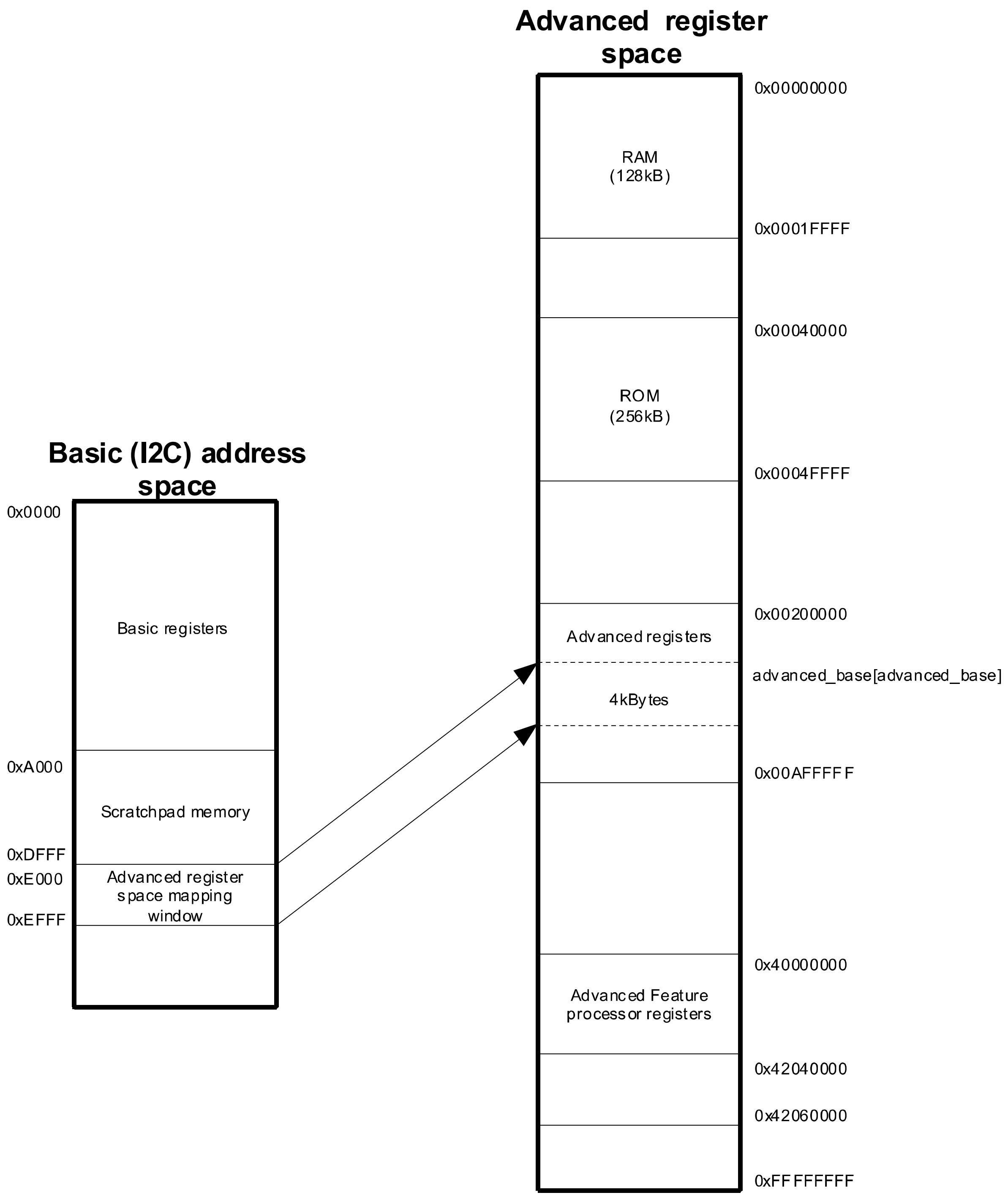


Figure 15. Advanced Register Space to Basic Register Space Mapping

Base address inside Advanced register space to which this 4 kb window is mapped is defined in Advanced space base pointer register.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

Table 4. ADVANCED SPACE BASE POINTER REGISTER

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	Reserved												advanced_base			
RST	0x0												0x0			
R/W	rw												rw			
ADD R	0xF038															
SYN C	none															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	advanced_base															
RST	0x0															
R/W	rw															
ADDR	0xF03A															
SYNC	none															

Len	Bits	Name	Reset	Description
16	19:0	advanced_base	0x0	This field defines the Advanced system address that Basic page starting at 0xE000 is mapped to (used to map advanced address space).

```
Example of setting GPIO3 to 1 by setting ADV_GPIO_DIR[3] and ADV_GPIO_DO[3] bits:
REG= 0xF038, 0x002A0000 // ADVANCED_BASE [ADV_GPIO_DIR]
REG= 0xE004, 0x0000069A // ADV_GPIO_DIR
REG= 0xE000, 0x0000049A // ADV_GPIO_DO
```

SPI Slave Interface

Host can control AP1302 either through the I^{2}C or the SPI bus. In both cases, the host acts as master and AP1302 acts as a slave. When SPI slave interface is used, the host sees the same Basic register set through the SPI interface as it is visible through the I^{2}C interface.

Every access on SPI bus starts with the Basic register space 16−bit address, followed by an 8−bit control byte (holding 1 for WRITE and 0 for READ) and then one or more data bytes. Figure 16 shows write and read cycle waveforms. All data are transferred MSB first. SPI mode 0 is adopted.

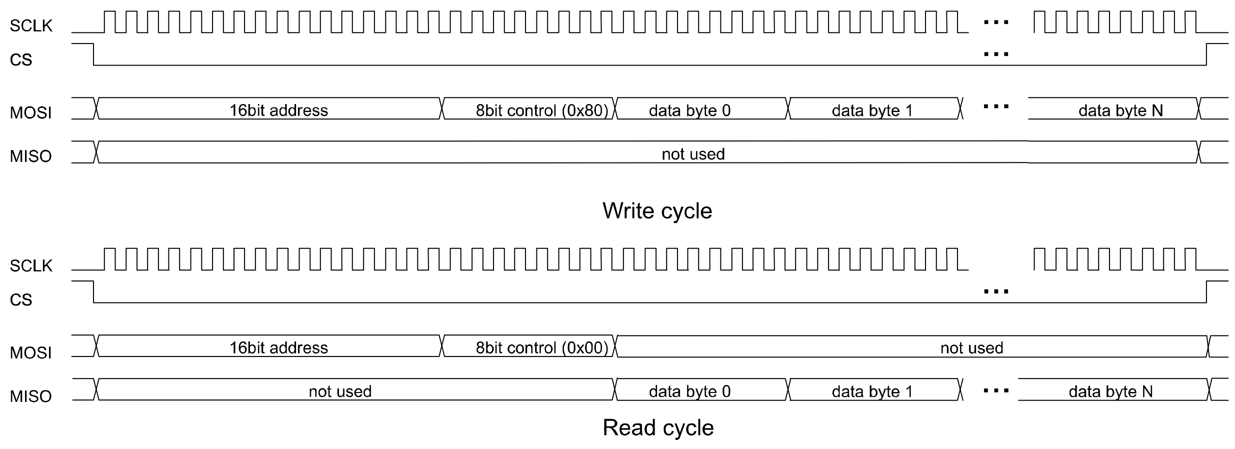


Figure 16. SPI Write and Read Data Access

SPI Slave interface of AP1302 is by default disabled. It can be enabled through advanced registers (please refer to Advanced Register Space). These are the advanced register writes necessary to enable SPI Slave interface:

```
// SPI Slave interface: Enabling.
REG= 0xF038, 0x002B0000 // ADVANCED_BASE
REG= 0xE020, 0x03333000 // ADV_SYS_STBY_GPIO_OSEL
REG= 0xE028, 0x19919111 // ADV_SYS_STBY_GPIO_IN_MASK
```

SPIS clock has a limitation depending on SYS clock. SYS clock, which internal system bus is running on, needs to be fast enough to make sure that write data are written before the new write access comes over SPIS bus. This means that write access needs to finish before then next 16 bits are received over SPIS bus (assuming burst writes). Similarly, on read access the data needs to be read before they need to be sent over the SPI bus back to the host, which means that there are 7 SPI cycles available to read the data over system bus. Since the system bus operation in not entirely predictable (when SPIS, which is master on internal bus, asserts bus request, the CPU or any other master just started a long access over the bus, which means that the SPI will have to wait), worst case is assumed, which is 20 SYS clock cycles for access. Since SPI might have to wait for one access that just started and then to finish its own access, in worst case 40 SYS cycles needs to be compared against 7 SPIS cycles, which gives a minimum ratio of SYS/SPIS >= 40/7 =~ 6.

While the above holds true for normal operation, the conditions are different when the system boots up. The CPU or any other masters are not performing any access over the system bus at that time. Also, when loading the bootdata code, the host only performs writes. In this particular case the SPIS clock can match the SYS clock, so minimum ration can be lowered to SYS/SPIS >= 1.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

Clocking Scheme

For maximum flexibility, AP1302 deploys a complex clocking scheme, with two independent PLLs and number of dividers and clock domains.
Despite the complexity, configuration is trivial for the user. All the settings are calculated by the AP1302 firmware at boot time, user only needs to set couple of registers to inform firmware about the input clock and target MIPI speed. See Boot procedure section for details.

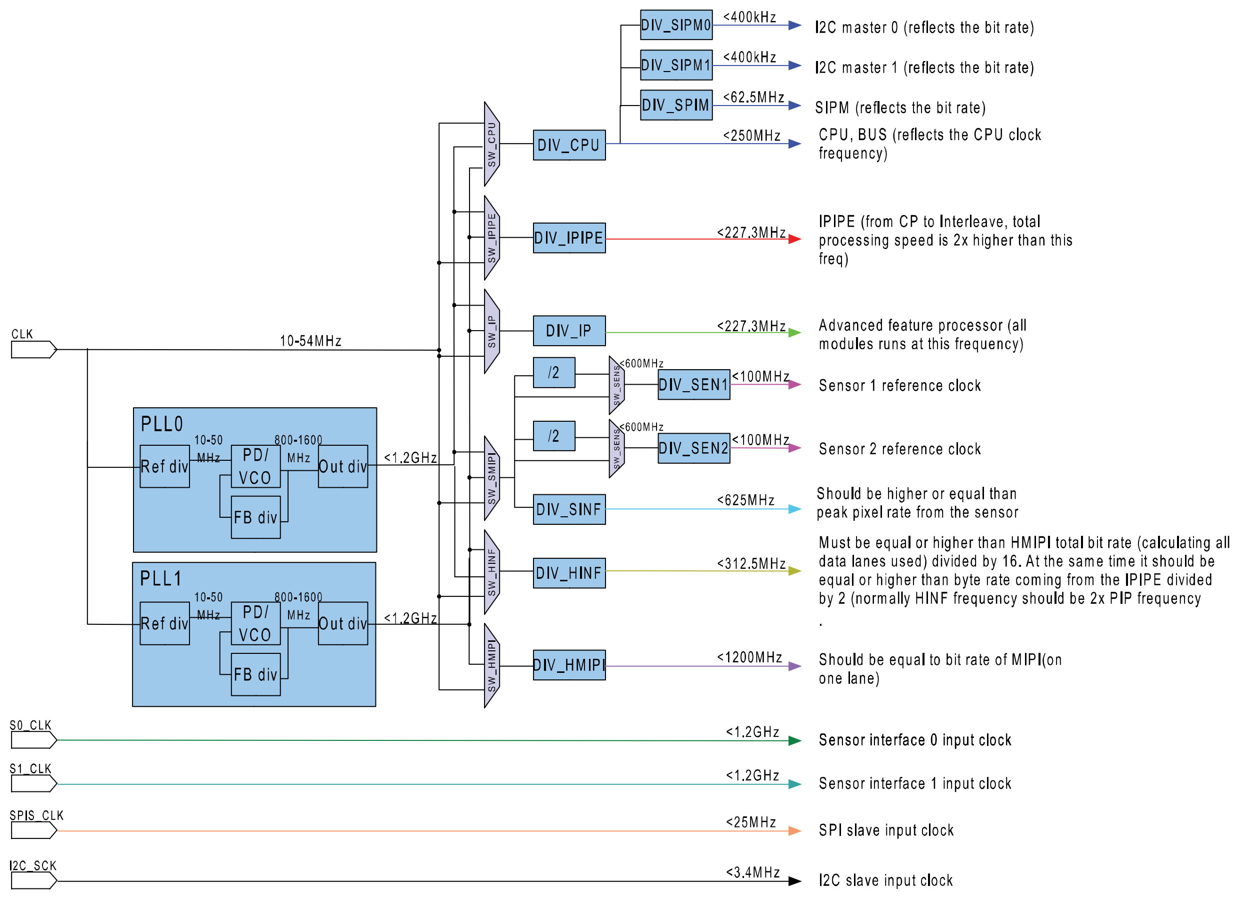

DIV_SIPM0
<400kHz
I2C master 0 (reflects the bit rate)
DIV_SIPM1
<400kHz
I2C master 1 (reflects the bit rate)
DIV_SPIM
<62.5MHz
SIPM (reflects the bit rate)
SW_CPU
DIV_CPU
<250MHz
CPU, BUS (reflects the CPU clock frequency)
SW_IPIPE
DIV_IPIPE
<227.3MHz
IPIPE (from CP to Interleave, total processing speed is 2x higher than this freq)
CLK
10-54MHz
SW_IP
DIV_IP
<227.3MHz
Advanced feature processor (all modules runs at this frequency)
/2
SW_SENS
<600MHz
DIV_SEN1
<100MHz
Sensor 1 reference clock
/2
SW_SENS
<600MHz
DIV_SEN2
<100MHz
Sensor 2 reference clock
PLL0
10-50 MHz
800-1600 MHz
Ref div
PD/ VCO
Out div
FB div
<1.2GHz
SW_SMIPI
DIV_SINF
<625MHz
Should be higher or equal than peak pixel rate from the sensor
SW_HINF
DIV_HINF
<312.5MHz
Must be equal or higher than HMIPI total bit rate (calculating all data lanes used) divided by 16. At the same time it should be equal or higher than byte rate coming from the IPIPE divided by 2 (normally HINF frequency should be 2x PIP frequency
.
PLL1
10-50 MHz
800-1600 MHz
Ref div
PD/ VCO
Out div
FB div
<1.2GHz
SW_HMIPI
DIV_HMIPI
<1200MHz
Should be equal to bit rate of MIPI(on one lane)
S0_CLK
<1.2GHz
Sensor interface 0 input clock
S1_CLK
<1.2GHz
Sensor interface 1 input clock
SPIS_CLK
<25MHz
SPI slave input clock
I2C_SCK
<3.4MHz
I2C slave input clock

Figure 17. Clocking Scheme

POWER MODES

The AP1302 can be in one of the following power modes:

- Power down
- Core power down
- Standby
- Normal operation

Power Up

AP1302 is in power down state if all core power and IO power supplies are removed. AP1302 can maintain I^{2}C slave and GPIO interface IOs in HighZ state during power down state exercising its IO fail−safe feature. To bring the chip in normal operational mode, apply power to the chip and assert RESET. Figure 18 shows power−up sequence that guarantees fail−safe functionality.

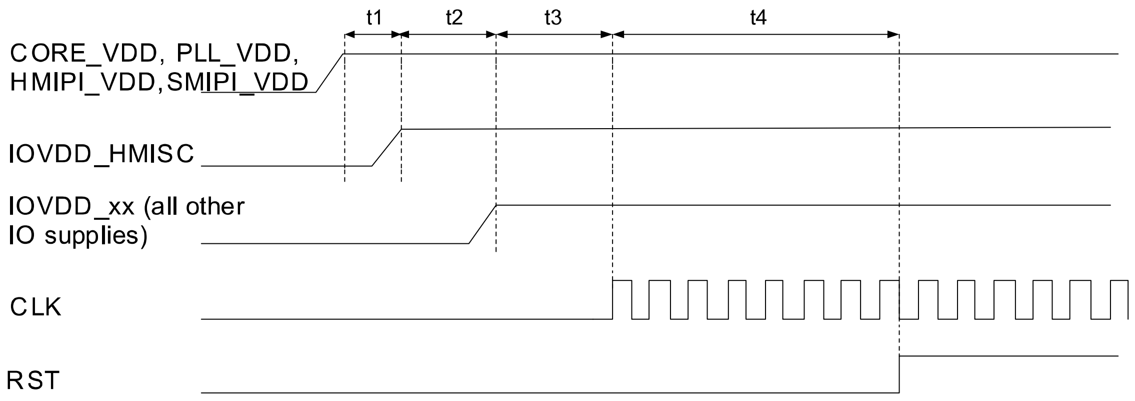


Figure 18. Power−Up Sequence (1.2V −> 1.8V), Fail−safe Guaranteed

Table 5. POWER UP SEQUENCE TIMING

Symbol	Parameter	Condition	Value	Unit
			Min	
t1	CORE_VDD to IOVDD_MISC delay		200	μs
t2	IOVDD_MISC to IOVDD_xx delay (Note 1)		200	μs
t3	Power stable to clock active delay		200	μs
t4	Clock active to reset release delay		1	μs

1. Supplying IOVDD_HMISC all others IO supplies (and keeping it after all other IO supplies) ensures that the logic, which is controlling the GPIOs and I^{2}C IOs, is properly reset before IO power is supplied (or when IO power is cut off) as IOVDD_HMISC is supplying the RESET input pad. If the fail−safe IO option is not required, then IOVDD_HMISC can be ramped up/down together with other IOVDD supplies, which might cause glitches when power up/down sequence is on.

The voltage ramp up order in the power−up sequence can also be reversed if the STANDBY pin is asserted before voltage ramp−up and de−asserted after voltage stabilizes and RESET is applied as shown in Figure 19. Note that when reversed power up sequence is used, the fail−safe functionality is not guaranteed.

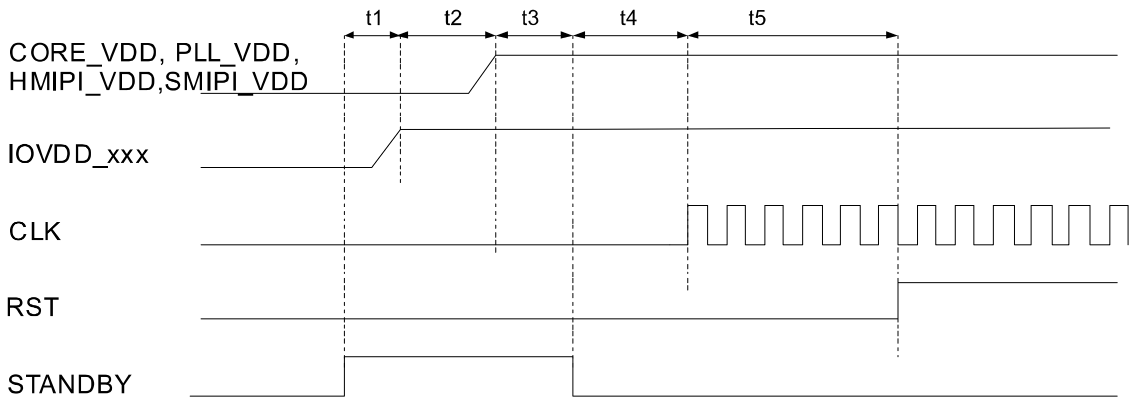


Figure 19. Reversed Power−Up Sequence (1.8V −> 1.2V), Fail−safe Not Guaranteed

Table 6. REVERSED POWER−UP SEQUENCE TIMING

Symbol	Parameter	Condition	Value	Unit
			Min	
t1	STANDBY ON to IOVDD_xxx		200	μs
t2	IOVDD_xxx to CORE_VDD delay		200	μs
t3	Power stable to STANDY OFF delay		200	μs
t4	STANDBY OFF to clock active delay		200	μs
t5	Clock active to reset release delay		1	μs

Power Down

Figure 20 shows the power down sequence.

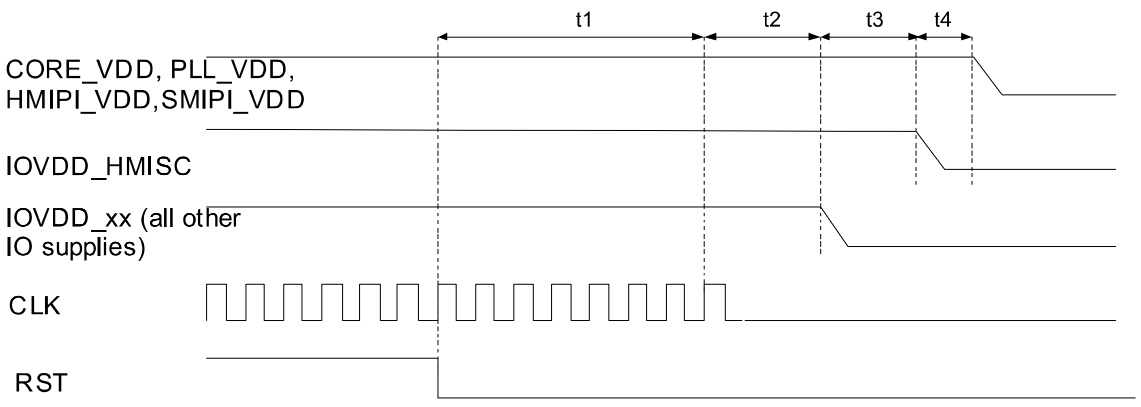


Figure 20. Power−Down Sequence (1.8 V −> 1.2 V), Fail−safe Guaranteed

Table 7. POWER−DOWN SEQUENCE TIMING

Symbol	Parameter	Condition	Value	Unit
t1	Reset active to clock nonactive delay		0	ns
t2	Clock nonactive to IOVDD_xx cutoff		1	μs
t3	IOVDD_xx to IOVDD_HMISC cut off delay (Note 2)		200	μs
t4	IOVDD_HMISC to CORE_VDD cut off delay		200	μs

2. Supplying IOVDD_HMISC before all other IO supplies (and keeping it after all other IO supplies) ensures that the logic, which is controlling the GPIOs and I^2C IOs, is properly reset before power for these IOs is supplied (or during power for these IOs is cut off) as IOVDD_HMISC is supplying the RESET input pad. If fail−safe IO functionality is not required, then IOVDD_HMISC can be ramp up/down together with the rest IOVDD supplies, which might cause glitches on the I^2C and GPIOs when the power up or power down sequence is on.

Voltage ramp down order can also be reversed in power down sequence, but in this case the STANDBY signal needs to be asserted. Note that fail−safe functionality is not guaranteed with reversed power down sequence.

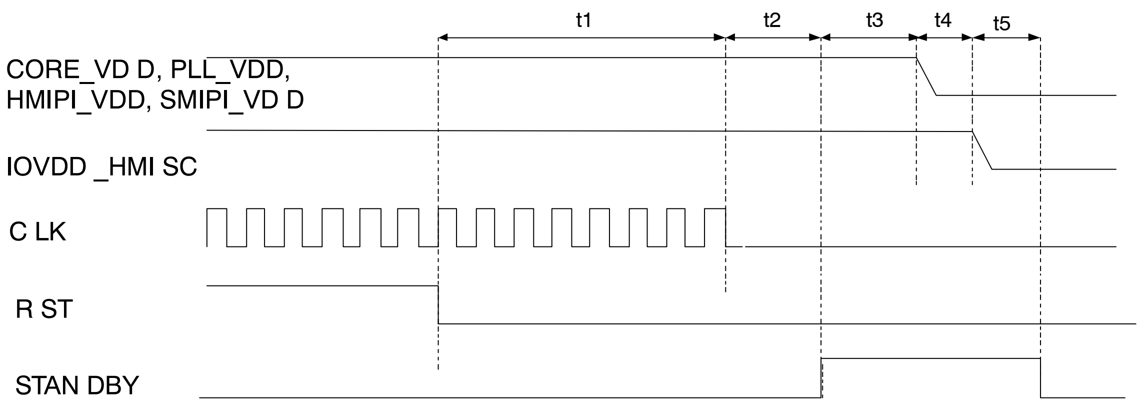


Figure 21. Reversed Power−Down Sequence (1.2 V −> 1.8 V), Fail−safe Not Guaranteed

Table 8. REVERSED POWER−DOWN SEQUENCE TIMING

Symbol	Parameter	Condition	Value	Unit
t1	Reset active to clock nonactive delay	−	0	ns
t2	Clock inactive to STANDBY ON	−	1	μs
t3	STANDBY ON to CORE_VDD cut off delay (Note 3)	−	200	μs
t4	CORE_VDD to IOVDD_xxx cut off delay	−	200	μs
t5	IOVDD_xxx to STANDBY OFF delay	−	200	μs

3. Supplying IOVDD_HMISC before all others IO supplies (and kept it after all others IO supplies), ensures that the logic, which is controlling the GPIOs and I^2C IOs is properly reset before power these IOs is supplied (or during power for these IOs is cut off) as IOVDD_HMISC is supplying the RST input pad. If fail−safe IO functionality is not required, then IOVDD_HMISC can be ramp up/down together with rest IOVDD supplies, which might cause glitches on the I^2C and GPIOs when the power up or power down sequence is on.

Core Power Down

I^2C and GPIO interfaces are using fail−safe IOs. This means that I^2C and GPIO signals, which are in HighZ state before power down sequence is started (when reset is applied), maintain HighZ state during the power down state. When the AP1302 is brought back to operational mode, the GPIO signals, which are in HighZ state during reset (refer to Table 1 for GPIO signal reset state), and the I^2C signals will maintain HighZ state during the power up sequence. I^2C signals maintain HighZ state until the host performs the first access via I^2C.

The AP1302 enters core power down state if core power is removed. IOs are still powered during core power down state in order for GPIO signals to maintain configured state (HighZ/driving 1/ driving 0). This is achieved by asserting STANDBY signal just before removing core power, which will latch in the IO configuration state inside the IO voltage domain. This way, the GPIO signals will maintain the state that they were assuming just before asserting the STANDBY signal for the core power down state duration.

To enter core power down state host must:

1. Configure the GPIO in the desired state.
2. Make sure that AP1302 is not being accessed on the SIPS bus.
3. Enter standby mode by asserting STANDBY signal.
4. Cut off core power.

In core power down state, the chip consumes the least amount of power. Core partition of the chip is in power down state and all configuration data is lost. Reference clock signal CLK may be disabled by the host in this state.

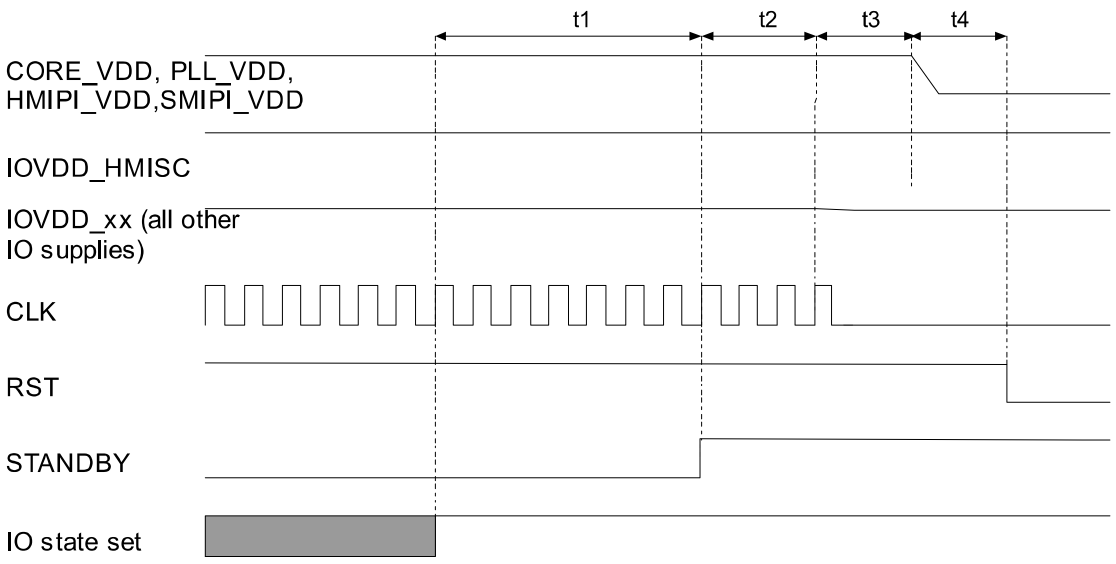

t1
t2
t3
t4
CORE_VDD, PLL_VDD, HMIPI_VDD,SMIPI_VDD
IOVDD_HMISC
IOVDD_xx (all other IO supplies)
CLK
RST
STANDBY
IO state set

Figure 22. Entering Core Power Down State

Table 9. CORE POWER–DOWN SEQUENCE TIMING

Symbol	Parameter	Condition	Value	Unit
t1	IO state setting to STANDBY delay		1	μs
t2	STANDBY to clock non–active delay		10	μs
t3	Clock non–active to CORE_VDD cut–off delay		1	μs
t4	CORE_VDD cut–off to RESET delay		200	μs

When the chip is brought from core power down state by applying the core power, the clock must be enabled and reset must be applied so that internal logic is set to a known state. Note that when the reset condition is applied, the GPIO signals will enter reset state, which might differ from the state specified by the host before STANDBY was asserted (refer to Table 2 for GPIO reset states). Host must do the initialization of the chip including the GPIO state again.

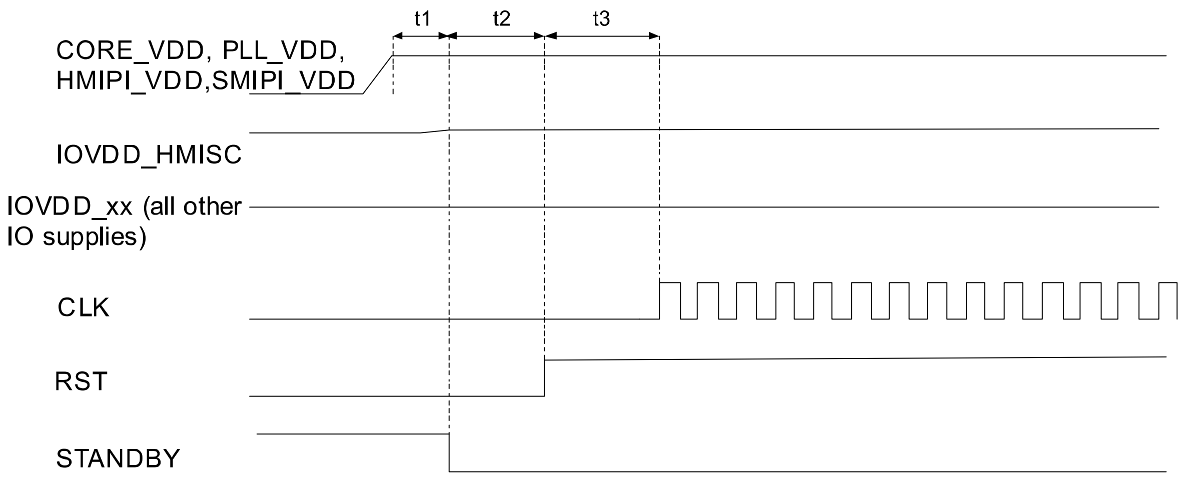

t1
t2
t3
CORE_VDD, PLL_VDD, HMIPI_VDD,SMIPI_VDD
IOVDD_HMISC
IOVDD_xx (all other IO supplies)
CLK
RST
STANDBY

Figure 23. Exiting Core Power Down State

Table 10. EXITING CORE POWER−DOWN SEQUENCE TIMING

Symbol	Parameter	Condition	Value			Unit
			Min	Typ	Max	
t1	CORE_VDD stable to STANDBY off delay	–	200	–	500000	μs
t2	STANDBY off to RESET off delay	–	0	–	500000	μs
t3	RESET off to clock active delay	–	0	–	500000	μs

Standby

Standby mode can be entered using the sys_start register (stall_mode = 2/3, stall_en = 1). All the clocks are gated, except for the system clock which is slowed down to bare minimum required to keep the register access working. PLLs and MIPI IOs are shut down. Sensor and any lens driver can be shut down too, to save power on the whole camera subsystem (stall_mode = 3).

To save even more power, reference clock to AP1302 can be stopped during standby if register access is not needed.

Note that use of STANDBY pin for standby mode purposes is obsolete and the signal should be kept low during standby.

GPIOs

GPIOs are used to control various external devices mostly related to sensor modules, like mechanical shutters, optical zoom stepper motors, flash devices and similar. There are 12 GPIOs all together in AP1302. GPIOs are sharing the pins with some external interfaces as listed in Table 11.

Table 11. GPIO 2ND FUNCTION MAPPING

GPIO	Shared Functions	Reset Value
0	SEN_CLK1	HighZ
1	SHUTDOWN1	0
2	SEN_CLK2	HighZ
3	SHUTDOWN2, SPIS_CLK	HighZ
4	SPIS_MISO	1
5	SPIS_MOSI, 2ND_I2C_SCL	HighZ
6	SPIS_CS, 2ND_I2C_SDA	HighZ
7	SPIM_CLK	1
8	SPIM_MISO	HighZ
9	SPIM_MOSI	0
10	SPIM_CS	1
11	I2C slave address select	HighZ

All GPIOs are capable of PWM functions. There are three independent PWM engines, each can drive four GPIOs. Each engine can have up to eight states, which means it can do a sequence of eight different GPIO states with four GPIOs and it can automatically cycle through the sequence.

GPIOs are not controlled by the FW when they are not used as interfaces in the 2nd function. This way host has direct control over the GPIOs and can set the direction, read the value when GPIO is configured as input, and set the driving value when GPIO is configured as output. This way the host can use GPIOs as the reset signal for sensors, driving LED flash and similar functions independently of the AP1302 FW control loop.

GPIOs have the ability to retain state during Standby and Core power down mode. GPIO also have fail−safe functionality, which means that they will remain in the High−Z state when the chip will be powered down.

GENERAL PHYSICAL SPECIFICATIONS

- Package type: VFBGA
- Package thickness: 0.80 mm (maximum)
- Package size: 6.5 ±0.10 mm x 6.5 ±0.10 mm
- Ball count: 120
- Ball diameter: 0.3 ±0.05 mm
- Min. ball pitch: 0.50 mm

Table 12 describes all signals of the AP1302.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

Table 12. SIGNAL DESCRIPTIONS

Signal	Ball Assignments	Direction	Description
CLK	M2	Input	Master Clock input.
RST	L1	Input	System Reset; Asynchronous assertion, synchronously released
TESTMODE	B1	Input	Test mode.
STANDBY	K1	Input	Device Stand−by mode enable input.
S1_D1_DP	A9	Input	Secondary sensor serial interface pixel data input – Lane 1 (sub−LVDS)
S1_D1_DN	B9	Input	Secondary sensor serial interface pixel data input – Lane 1 (sub−LVDS)
S1_CLK_DP	A7	Input	Secondary Sensor Serial Interface Clock input (sub−LVDS)
S1_CLK_DN	B7	Input	Secondary Sensor Serial Interface Clock input (sub−LVDS)
S1_D0_DP	A8	Input	Secondary Sensor Serial Interface Pixel Data input– Lane 0 (sub−LVDS)
S1_D0_DN	B8	Input	Secondary Sensor Serial Interface Pixel Data input t– Lane 0 (sub– LVDS)
S0_CLK_DP	A4	Input	Primary Sensor Serial Interface Clock input (sub−LVDS)
S0_CLK_DN	B4	Input	Primary Sensor Serial Interface Clock input (sub−LVDS)
S0_D0_DP	A2	Input	Primary Sensor Serial Interface Pixel Data input – Lane 0 (sub−LVDS)
S0_D0_DN	B2	Input	Primary Sensor Serial Interface Pixel Data input – Lane 0 (sub−LVDS)
S0_D1_DP	A3	Input	Primary Sensor Serial Interface Pixel Data input – Lane 1 (sub−LVDS)
S0_D1_DN	B3	Input	Primary Sensor Serial Interface Pixel Data input – Lane 1 (sub−LVDS)
S0_D2_DP	A5	Input	Primary Sensor Serial Interface Pixel Data input – Lane 2 (sub−LVDS)
S0_D2_DN	B5	Input	Primary Sensor Serial Interface Pixel Data input – Lane 2 (sub−LVDS)
S01_D32_DP	A6	Input	Primary Sensor Serial Interface Pixel Data input – Lane 3 (sub−LVDS)
S01_D32_DN	B6	Input	Primary Sensor Serial Interface Pixel Data input – Lane 3 (sub−LVDS)
GPIO[11]	A11	Bi−Directional	General Purpose IO
GPIO[10]	J12	Bi−Directional	General Purpose IO
GPIO[9]	K12	Bi−Directional	General Purpose IO
GPIO[8]	L12	Bi−Directional	General Purpose IO
GPIO[7]	B12	Bi−Directional	General Purpose IO
GPIO[6]	L4	Bi−Directional	General Purpose IO
GPIO[5]	M4	Bi−Directional	General Purpose IO
GPIO[4]	M6	Bi−Directional	General Purpose IO
GPIO[3]	M5	Bi−Directional	General Purpose IO
GPIO[2]	A10	Bi−Directional	General Purpose IO
GPIO[1]	C10	Bi−Directional	General Purpose IO
GPIO[0]	B10	Bi−Directional	General Purpose IO

Table 12. SIGNAL DESCRIPTIONS

Signal	Ball Assignments	Direction	Description
VSS	E2, E3, E5, E6, E7, E8, F2, F3, F5, F6, F7, F8, F10, F11, G2, G3, G5, G6, G7, G8, G10, G11, H2, H3, H5, H6, H7, H8, H10, H11, L5, K5	Power	Common Ground
HMIPI_VSS	K6, K7, K8, L6	Power	Host MIPI VSS
SMIPI_VSS	C4, C5, C6, C7	Power	Sensor MIPI VSS
PLL0_VSS	K4	Power	PLL0 VSS
PLL1_VSS	L2	Power	PLL1 VSS
CORE_VDD	C9, D1, E1, E10, E11, E12, F1, F12, G1, G12, H1, H12, J1, J2, J3	Power	Core Power Supply; 1.2 V Nominal
PLL0_VDD	L3	Power	PLL0 Power Supply; 1.2 V Nominal
PLL1_VDD	K3	Power	PLL1 Power Supply; 1.2 V Nominal
IOVDD_GPIOH	M3	Power	GPIO Power; 1.8V Nominal
IOVDD_GPIO	K11	Power	GPIO Power, 1.8V nominal
IOVDD_HMISC	K2	Power	Host MISC Interface IO Power; 1.8 V Nominal
SMIPI_VDD	C2, C3, D2, D3	Power	Sensor MIPI Power Supply; 1.2 V Nominal
IOVDD_SMISC	C8	Power	Sensor MISC interface IO Power; 1.8VNominal
IOVDD	C1	Power	IO Power Supply 1.8 V Nominal
HMIPI_VDD	J10, J11, K9, K10	Power	Host MIPI IO Power Supply, 1.2 V Nominal
H_CLK_DP	L7	Output	Host Serial Interface Clock Output
H_CLK_DN	M7	Output	Host Serial Interface Clock Output
H_D0_DP	L11	Output	Host Serial Interface Data Output – Lane 0
H_D0_DN	M11	Output	Host Serial Interface Data Output – Lane 0
H_D1_DP	L10	Output	Host Serial Interface Data Output – Lane 1
H_D1_DN	M10	Output	Host Serial Interface Data Output – Lane 1
H_D2_DP	L9	Output	Host Serial Interface Data Output – Lane 2
H_D2_DN	M9	Output	Host Serial Interface Data Output – Lane 2
H_D3_DP	L8	Output	Host Serial Interface Data Output – Lane 3
H_D3_DN	M8	Output	Host Serial Interface Data Output – Lane 3
SIPS_SCL	C12	Bi–Directional	I2C Slave Clock
SIPS_SDA	D12	Bi–Directional	I2C Slave Data
SIPM_SCL	B11	Bi–Directional	I2C Master Clock
SIPM_SDA	C11	Bi–Directional	I2C Master Data
I2C_VSS	D11	Power	I2C Slave ground
IOVDD_SIPS	D10	Power	I2C Slave interface IO Power; 1.8 V Nominal
Total 120 Balls			

4. IO power supplies are grouped.
   Group A.) IOVDD_SIPS
   Group B.) IOVDD, IOVDD_HOST and IOVDD_HMISC Group C.) IOVDD_GPIO, IOVDD_GPIOH, IOVDD_SMISC.
5. The supplies within one group should have the same voltage, but can be powered on/off independently.

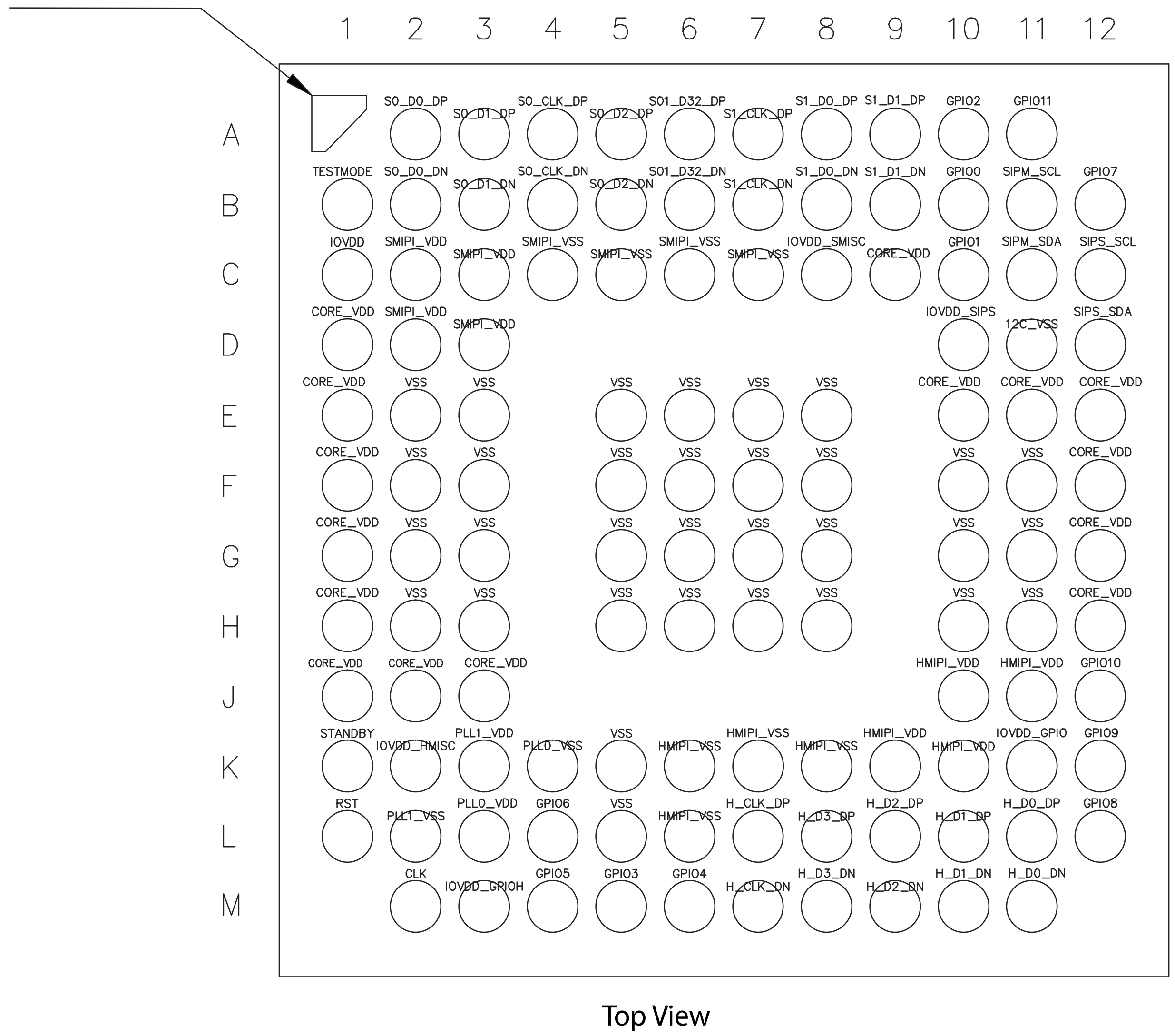


Figure 24. Ball Map Diagram

POWER CONSUMPTION MEASUREMENT SUMMARY

Table 13. TYPICAL POWER CONSUMPTION

	Input Resolution	1.2 V Current (mA)	1.8 V Current (mA)	Power (mW)
Standby	N/A	4	0.7	6
Standby, input clock stopped	N/A	3.5	0.5	5
13MPix @ 30FPS JPEG	4208x3120	570	2	690
13MPix @ 15FPS	4208x3120	340	2	410
4K @ 30FPS	4208x2367	450	2	540
4K @ 30FPS	3840x2160	340	2	410
1080p @ 30FPS HQ	4208x2367	410	2	500
1080p @ 30FPS HQ	3840x2160	330	2	390
1080p @ 30FPS	1920x1080	190	2	230
1080p @ 60FPS	1920x1080	270	2	330

NOTE: Power supply capable of 1 A or more is recommended for 1.2 V rail to cover the peak power consumption.

ELECTRICAL CHARACTERISTICS

Table 14. DC ELECTRICAL DEFINITIONS AND CHARACTERISTICS

Power Source	Standard Voltage	Minimum Voltage	Typical Voltage	Maximum Voltage	Unit
CORE_VDD	Core digital circuit power supply	1.14	1.2	1.26	V
SMIPI_VDD	Power for Sensor MIPI interface	1.14	1.2	1.26	V
PLL_VDD	PLL circuit power supply	1.14	1.2	1.26	V
IOVDD_SMISC	Power for sensor I/O interfaces	1.62	1.8	1.98	V
IOVDD_HMISC	Power for host I/O interfaces	1.62	1.8	1.98	V
IOVDD_GP_3_0	Power for GPIO interfaces	1.62	1.8	1.98	V
IOVDD_SIPS	Power for the I^{2}C interfaces	1.62	1.8	1.98	V
IOVDD_HOST	Power for the host I/O interfaces	1.62	1.8	1.98	V
IOVDD	Power for general I/O interfaces	1.62	1.8	1.98	V
HMIPI_VDD	Power for the Host MIPI interfaces	1.14	1.2	1.26	V

NOTE: Power supply capable of 1 A or more is recommended for 1.2 V rail to cover the peak power consumption.

Table 15. ABSOLUTE MAXIMUM RATINGS

Parameter	Value	Unit
Operating temperature (ambient)	–30 to +70	°C
Operating temperature (junction)	–30 to +85	°C
Storage temperature	–50 to +150	°C
Max voltage (1.2 V domain)	1.6	V
Max voltage (1.8 V domain)	3.6	V

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.

Master Clock Input (MCLK) Characteristics

Table 16. MCLK CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit
Fextclk	Input Clock Freq	10	54	MHz
IIH	Input High Leakage Current	0	5	μA
VIH	Input High Voltage	0.7*IOVDD	–	V
VIL	Input Low Voltage	–	0.3*IOVDD	V

Table 17. DC ELECTRICAL CHARACTERISTICS OF CONTROL SIGNALS (RST, STANDBY, CLK)

Symbol	Parameter	Min	Typ	Max	Unit
VIH	Input High voltage	0.7*IOVDD_HMISC			V
VIL	Input Low voltage			0.3*IOVDD_HMISC	V

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

I^{2}C Signal AC/DC Specifications

Table 18. I^{2}C SIGNAL SPECIFICATIONS

Parameter	Description	Min	Max	Unit
fSCL	SCL clock frequency	100	400	kHz
tSDHR	Data hold time	0	–	ns
tSDSR	Data setup time	0	–	ns
tSRTH	Setup time for start condition	0.6	–	us
tr_sclk	SCLK rise time	–	300 (120 for FM+)	ns
tf_sclk	SCLK fall time	6.5	300 (120 for FM+)	ns
tr_sdata	SDATA rise time	–	300 (120 for FM+)	ns
tf_sdata	SDATA fall time	6.5	300 (120 for FM+)	ns
VOL	Low level data output voltage	–	0.2 * IOVDD	V
VIL_sclk	SCLK input low voltage	–	0.3 * IOVDD	V
VIH_sclk	SCLK input high voltage	0.7 * IOVDD	–	V
VIL_sdata	SDATA input low voltage	–	0.3 * IOVDD	V
VIH_sdata	SDATA input high voltage	0.7 * IOVDD	–	V

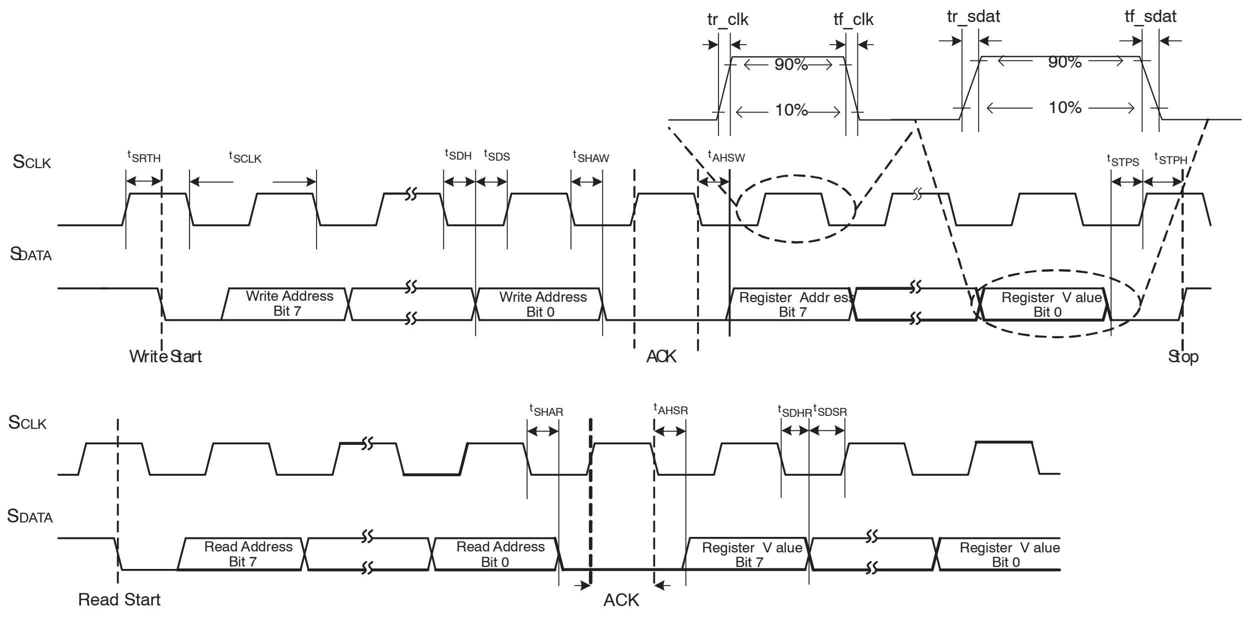


Figure 25. Two Wire Serial Bus Timing Parameters

Table 19. FOR I^{2}C MASTER BUS READ AND WRITE MODES

Measurement conditions: fSCL = 400 kHz; IOVDD=1.8 V; CORE_VDD=1.2 V; T_J = 55°C; Output load < 100 pF

Parameter	Description	Min	Typ	Max	Unit
fSCL	SCL clock frequency	100	400	400	kHz
tLOW	Low period of SCL		1.26	–	μs
tHIGH	High period of SCL		1.23	–	μs
tF	Fall time of SDA and SCL		2.34	–	ns

Table 19. FOR I²C MASTER BUS READ AND WRITE MODES
Measurement conditions: fSCL = 400 kHz; IOVDD=1.8 V; CORE_VDD=1.2 V; T_J = 55°C; Output load < 100 pF

Parameter	Description	Min	Typ	Max	Unit
tR	Rise time of SDA and SCL		91.41	–	ns
VHU:DAT	Data hold time	0	344.20	–	ns
VSU:DAT	Data setup time	100	854.87	–	ns

MIPI Electrical Characterization

Table 20. MIPI ELECTRICAL CHARACTERIZATION RESULTS

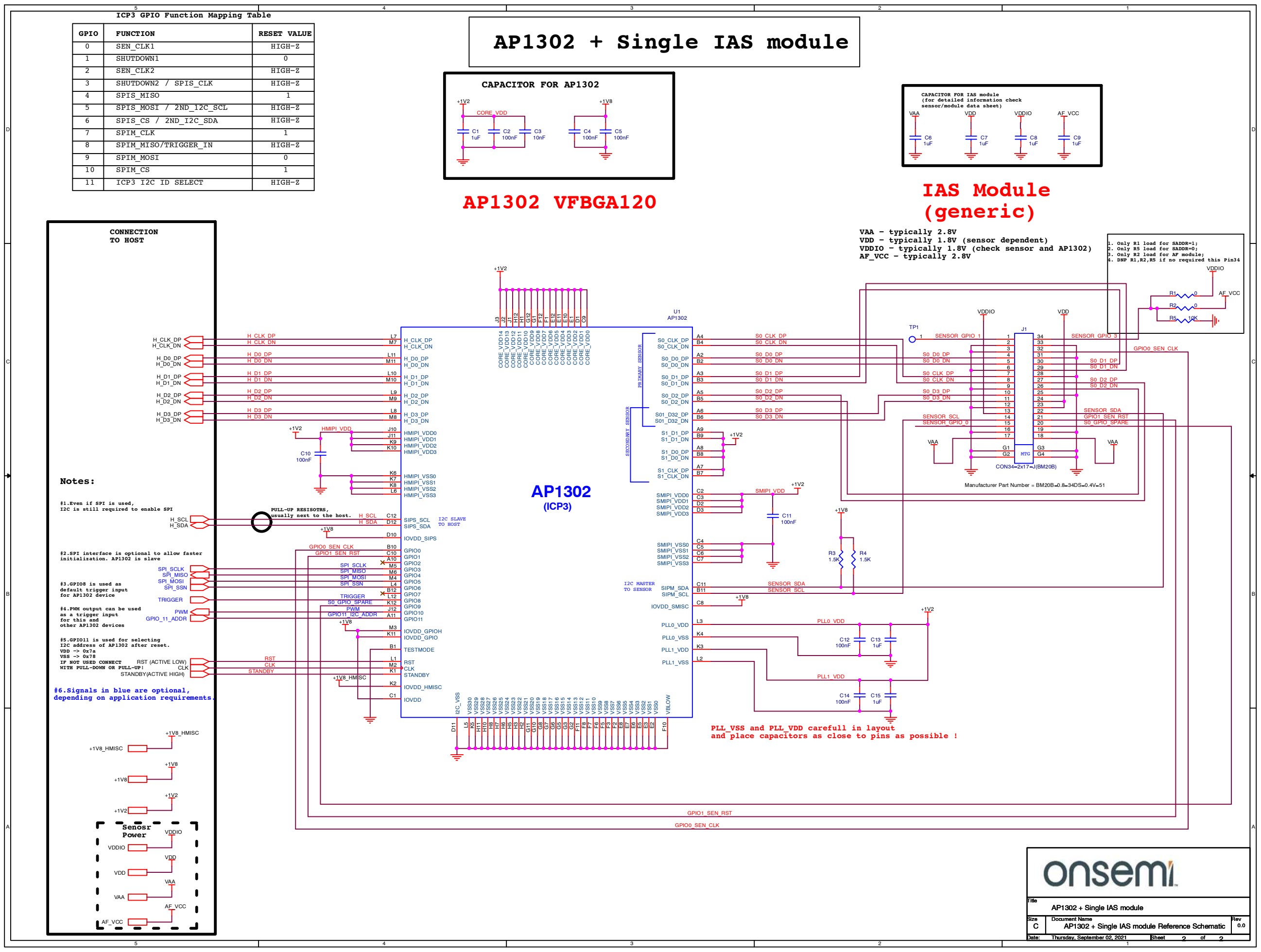
Parameter	Notes	Typ	Unit
VCMTX	HS transmit static common−mode voltage	195	mV
$\lvert \Delta V_{CMTX}(1,0) \rvert$	V_{CMTX} mismatch when output is differential−1 or differential−0	1	mV
$\lvert V_{OD} \rvert$	HS transmit differential voltage	195	mV
$\lvert \Delta V_{OD} \rvert$	V_{OD} mismatch when output is differential−1 or differential−0	5	mV
VOHHS	HS output high voltage	295	mV
ZOS	Single ended output impedance	52	Ω
ΔZ_{OS}	Single ended output impedance mismatch	5	%
ΔVCMTX(HF)	Common level variations 50~450 MHz	5	mVRMS
ΔVCMTX(LF)	Common level variations above 450 MHz	8	mVPEAK
t_R	HS: 20%~80% rise time	350	ps
t_F	HS: 20%~80% fall time	350	ps
TRLP/TFLP	LP: 15%~85% Rise time and fall time	21	ns
TREOT	LP: 30%~85% Rise time and fall time	27	ns
TLP−PULSE−TX	Pulse width of the first LP exclusive−OR clock	170	ns
TLP−PULSE−TX	Pulse width of others	170	ns
TLP−PER−TX	Period of the LP exclusive−OR clock	360	ns

NOTE: MIPI timing diagrams will be supplied. For definitions of these signals, please look up in the MIPI specifications.

CONFIDENTIAL AND PROPRIETARY
NOT FOR PUBLIC RELEASE

APPENDIX A: AP1302 REFERENCE SCHEMATICS USING STANDARD IMAGER ACCESS SYSTEM (IAS) MODULE SENSOR FOR REFERENCE

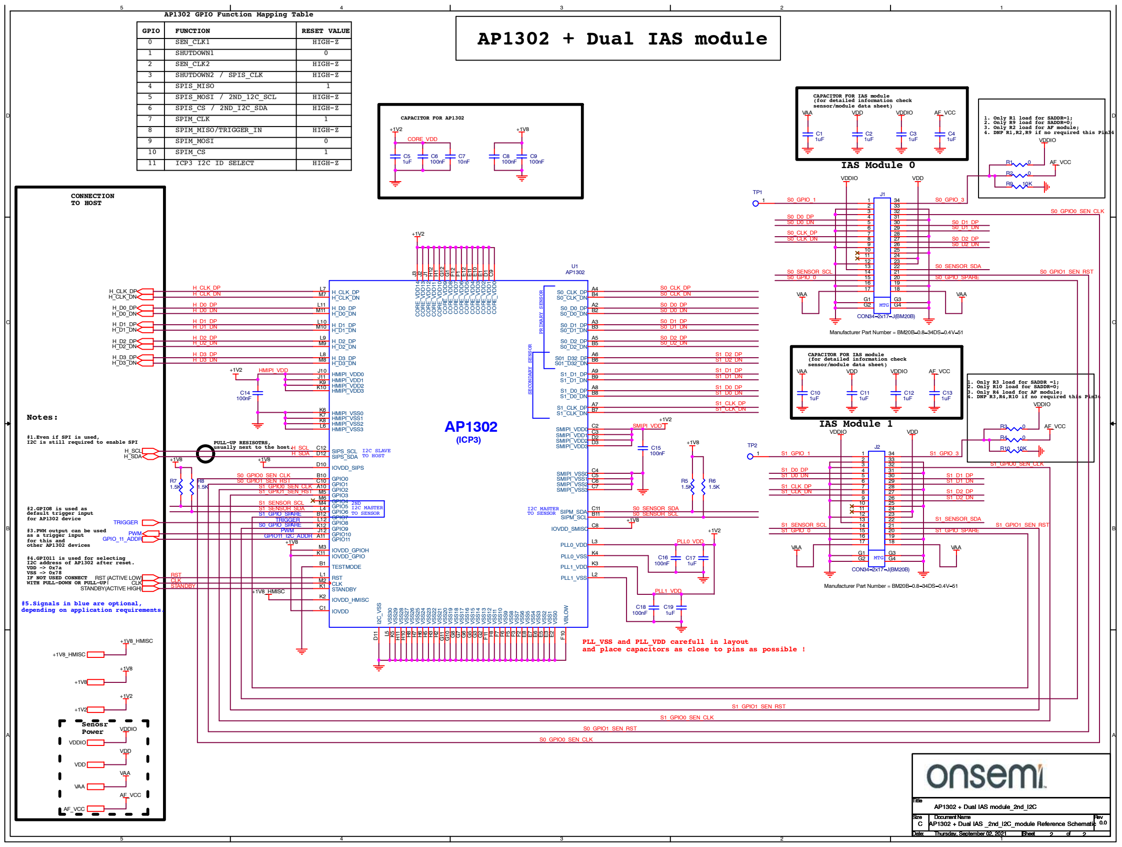
Note: Check your sensor part documentation for details on sensor schematics.


AP1302 + Single IAS module
CAPACITOR FOR AP1302
AP1302 VFBGA120
CAPACITOR FOR IAS module (for detailed information check sensor/module data sheet)
IAS Module (generic)
VAA – typically 2.8V
VDD – typically 1.8V (sensor dependent)
VDDIO – typically 1.8V (check sensor and AP1302)
AF_VCC – typically 2.8V
1. Only R1 load for SADDR=1;
2. Only R5 load for SADDR=0;
3. Only R2 load for AF module;
4. DNP R1,R2,R5 if no required this Pin34
CONNECTION TO HOST
AP1302
(ICP3)
Notes:
#1.Even if SPI is used, I2C is still required to enable SPI
PULL-UP RESISTORS, usually next to the host.
#2.SPI interface is optional to allow faster initialization. AP1302 is slave
#3.GPIO8 is used as default trigger input for AP1302 device
#4.PWM output can be used as a trigger input for this and other AP1302 devices
#5.GPIO11 is used for selecting I2C address of AP1302 after reset. VDD -> 0x7a VSS -> 0x78 IF NOT USED CONNECT WITH PULL-DOWN OR PULL-UP!
#6.Signals in blue are optional, depending on application requirements.
PLL_VSS and PLL_VDD carefull in layout and place capacitors as close to pins as possible !
Sensor Power
Manufacturer Part Number = BM20B-0.8-34DS-0.4V-51
CON34-2x17-J(BM20B)
onsemi
AP1302 + Single IAS module
AP1302 + Single IAS module Reference Schematic
Thursday, September 02, 2021

ICP3 GPIO Function Mapping Table

GPIO	FUNCTION	RESET VALUE
0	SEN_CLK1	HIGH-Z
1	SHUTDOWN1	0
2	SEN_CLK2	HIGH-Z
3	SHUTDOWN2 / SPIS_CLK	HIGH-Z
4	SPIS_MISO	1
5	SPIS_MOSI / 2ND_I2C_SCL	HIGH-Z
6	SPIS_CS / 2ND_I2C_SDA	HIGH-Z
7	SPIM_CLK	1
8	SPIM_MISO/TRIGGER_IN	HIGH-Z
9	SPIM_MOSI	0
10	SPIM_CS	1
11	ICP3 I2C ID SELECT	HIGH-Z

Figure 26. AP1302 Reference Schematics (using standard single IAS (Imager Access System) connector for reference)


AP1302 + Dual IAS module
CAPACITOR FOR AP1302
CONNECTION TO HOST
Notes:
#1.Even if SPI is used, I2C is still required to enable SPI
#2.GPIO8 is used as default trigger input for AP1302 device
#3.PWM output can be used as a trigger input for this and other AP1302 devices
#4.GPIO11 is used for selecting I2C address of AP1302 after reset. VDD -> 0x7a VSS -> 0x78 IF NOT USED CONNECT WITH PULL-DOWN OR PULL-UP!
#5.Signals in blue are optional, depending on application requirements.
PULL-UP RESISTORS, usually next to the host.
AP1302 (ICP3)
CAPACITOR FOR IAS module (for detailed information check sensor/module data sheet)
IAS Module 0
IAS Module 1
Manufacturer Part Number = BM20B-0.8-34DS-0.4V-51
PLL_VSS and PLL_VDD carefull in layout and place capacitors as close to pins as possible !
Senosr Power
onsemi
AP1302 + Dual IAS module_2nd_I2C

AP1302 GPIO Function Mapping Table

GPIO	FUNCTION	RESET VALUE
0	SEN_CLK1	HIGH-Z
1	SHUTDOWN1	0
2	SEN_CLK2	HIGH-Z
3	SHUTDOWN2 / SPIS_CLK	HIGH-Z
4	SPIS_MISO	1
5	SPIS_MOSI / 2ND_I2C_SCL	HIGH-Z
6	SPIS_CS / 2ND_I2C_SDA	HIGH-Z
7	SPIM_CLK	1
8	SPIM_MISO/TRIGGER_IN	HIGH-Z
9	SPIM_MOSI	0
10	SPIM_CS	1
11	ICP3 I2C ID SELECT	HIGH-Z

Figure 27. AP1302 Reference Schematics (using dual standard IAS (Imager Access System) connectors for reference)

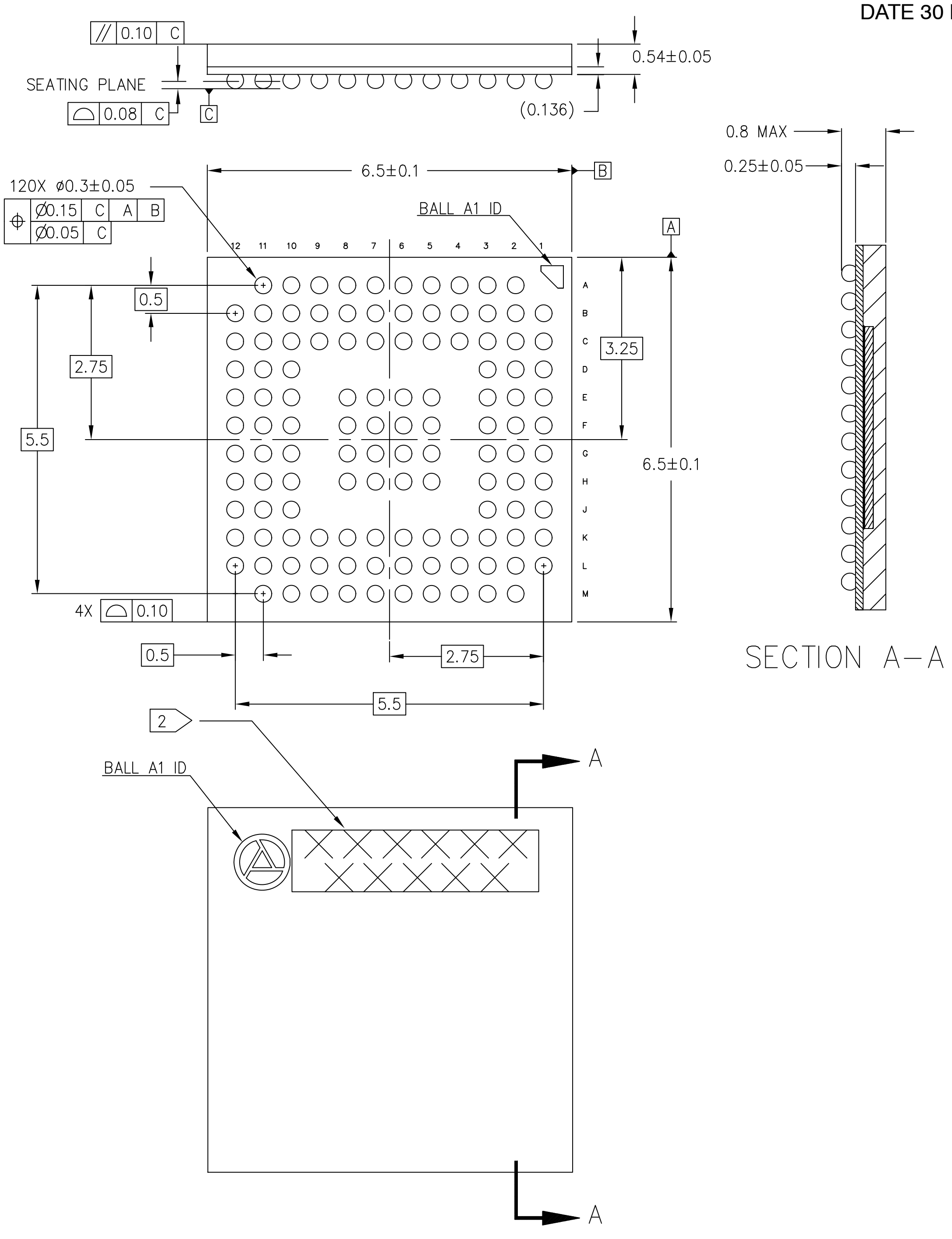
Clarity+ is a trademark of Semiconductor Components Industries, LLC dba “**onsemi**” or its affiliates and/or subsidiaries in the United States and/or other countries.

MECHANICAL CASE OUTLINE
PACKAGE DIMENSIONS

ON Semiconductor®

VFBGA120 6.5x6.5
CASE 138AK
ISSUE O

DATE 30 DEC 2014



1	DIMENSIONS ARE IN MM. DIMENSIONS IN ( ) ARE FOR REFERENCE ONLY
2	THIS AREA RESERVED FOR LASER MARKING.

DOCUMENT NUMBER:	**98AON93702F**	Electronic versions are uncontrolled except when accessed directly from the Document Repository. Printed versions are uncontrolled except when stamped "CONTROLLED COPY" in red.
DESCRIPTION:	**VFBGA120 6.5X6.5**	**PAGE 1 OF 1**

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 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. ON Semiconductor does not convey any license under its patent rights nor the rights of others.

onsemi, onsemi, and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "**onsemi**" or its affiliates and/or subsidiaries in the United States and/or other countries. **onsemi** owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of **onsemi**'s product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent−Marking.pdf. **onsemi** reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as−is" and **onsemi** makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does **onsemi** 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 **onsemi** products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by **onsemi**. "Typical" parameters which may be provided in **onsemi** 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. **onsemi** does not convey any license under any of its intellectual property rights nor the rights of others. **onsemi** 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 **onsemi** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **onsemi** 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 **onsemi** was negligent regarding the design or manufacture of the part. **onsemi** is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

ADDITIONAL INFORMATION

TECHNICAL PUBLICATIONS:
Technical Library: www.onsemi.com/design/resources/technical−documentation
onsemi Website: www.onsemi.com

ONLINE SUPPORT: www.onsemi.com/support
For additional information, please contact your local Sales Representative at www.onsemi.com/support/sales

◊