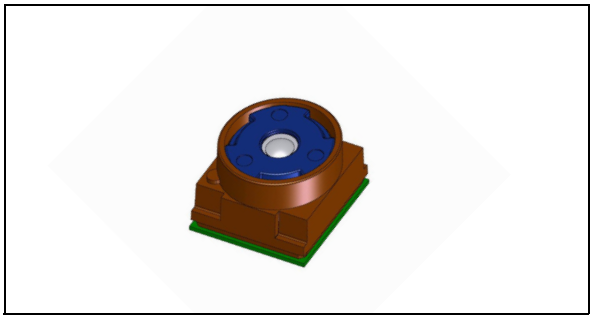
VS6724

2 Megapixel single-chip camera module

Preliminary Data

Features

- 1600H x 1200V active pixels
- Class leading 30 fps UXGA progressive scan
- 2.2 µm pixel size, 1/3.8 inch optical format
- Quad-element plastic lens, F# 3.2, 52° horizontal field of view
- 8.0 x 8.0 x 5.55 mm ultra low profile fixed focus camera module with embedded passives
- RGB Bayer color filter array
- Integrated 10-bit ADC
- Integrated digital image processing functions, including defect correction, lens shading correction, image scaling, interpolation, sharpening, gamma correction and color space conversion
- Embedded hardware JPEG compression (4:2:0 or 4:2:2) delivering 30 fps streaming performance
- Embedded camera controller for automatic exposure control, automatic white balance control, black level compensation, 50/60 Hz flicker cancelling, flashgun support.
- Fully programmable frame rate and output derating functions
- Low power 30 fps up to SVGA for video capture
- ITU-R BT.656-4 YUV (YCbCr) 4:2:2 with embedded syncs, YUV (YCbCr) 4:0:0, RGB 565, RGB 444, JPEG, Bayer 10-bit or Bayer 8-bit output formats
- 8-bit parallel video interface, horizontal and vertical syncs, 80 MHz (max) clock
- Two-wire serial control interface
- On-chip PLL, 6.5 to 27 MHz clock input
- Analog power supply, from 2.4 V to 3.0 V
- Separate I/O power supply, 1.8 V or 2.8 V levels



- Integrated power management with power switch, automatic power-on reset and power-safe pins
- Low power consumption, ultra low standby current
- 24-pin 8.0 mm x 8.0 mm x 5.55 mm shielded socket option

Applications

- Mobile phone, videophone
- PDA, toys
- Medical
- Biometry
- Bar code reader
- Lighting control

Description

The VS6724 is a UXGA resolution CMOS imaging device designed for low power systems, particularly mobile phone applications. Manufactured using ST 0.18 µm CMOS Imaging process, it integrates a high-sensitivity pixel array, digital image processor and camera control functions.

Contents

List of tables

List of figures

1 Overview

1.1 Description

The VS6724 is a UXGA resolution CMOS imaging device designed for low power systems, particularly mobile phone applications.

Manufactured using ST 0.18 µm CMOS Imaging process, it integrates a high-sensitivity pixel array, digital image processor and camera control functions.

The VS6724 is capable of streaming UXGA video up to 30 fps JPEG and up to 15 fps with ITU-R BT.656-4 YCbCr 4:2:2 format. It supports 1.8 V or 2.8 V interface and requires a 2.4 to 3.0 V analog power supply. Typically, the VS6724 can operate as a 2.8 V single supply camera or as a 1.8 V interface / 2.8 V supply camera. The integrated PLL allows for low frequency system clock, and flexibility for successful EMC integration. An input clock is required in the range 6.5 MHz to 27 MHz.

The VS6724 camera module uses ST's second generation "SmOP2" packaging technology: the sensor, lens and passives are assembled, tested and focused in a fully automated process, allowing high volume and low cost production.

The device contains an embedded video processor and delivers fully color processed images at up to 30 fps UXGA JPEG, or up to 30 fps SVGA YCbCr 4:2:2.

The video data is output over an 8-bit parallel bus in JPEG (4:2:2 or 4:2:0), RGB, YCbCr or Bayer formats and the device is controlled via an I²C interface.

The VS6724 also includes a wide range of image enhancement functions, designed to ensure high image quality, these include:

- Automatic exposure control
- Automatic white balance
- Lens shading compensation
- Defect correction algorithms
- Interpolation (Bayer to RGB conversion)
- Color space conversion
- Sharpening
- Gamma correction
- Flicker cancellation
- NoRA noise reduction algorithm
- Intelligent image scaling
- Special effects

1.2 Signal description

The VS6724 has 20 electrical connections which are described in *Table 1*.

The package details and electrical pinout of the flex connector are shown in *Figure 45 on page 115* and *Figure 46 on page 116*.

Table 1. Signal description

Pad name	Type	Description	Reference voltage
GND	PWR	Ground(1)	
HSYNC	OUT	Horizontal synchronization output	VDD
VSYNC	OUT	Vertical synchronization output	VDD
SCL	IN	I²C clock input	VDD
CLK	IN	Master clock input - 6.5MHz to 27MHz	VDD
SDA	I/O	I²C data line	VDD
VDD	PWR	Digital supply	1.8 V OR 2.8 V
AVDD	PWR	Analogue supply	2.4 V to 3.0 V
PCLK	OUT	Pixel qualification clock	VDD
CE	IN	Chip enable signal active HIGH	VDD
DO5	OUT	Data output D5	VDD
DO4	OUT	Data output D4	VDD
GND	PWR	Ground(1)	
DO3	OUT	Data output D3	VDD
DO2	OUT	Data output D2	VDD
DO1	OUT	Data output D1	VDD
DO0	OUT	Data output D0	VDD
DO6	OUT	Data output D6	VDD
DO7	OUT	Data output D7	VDD
FSO	OUT	Flash output	VDD

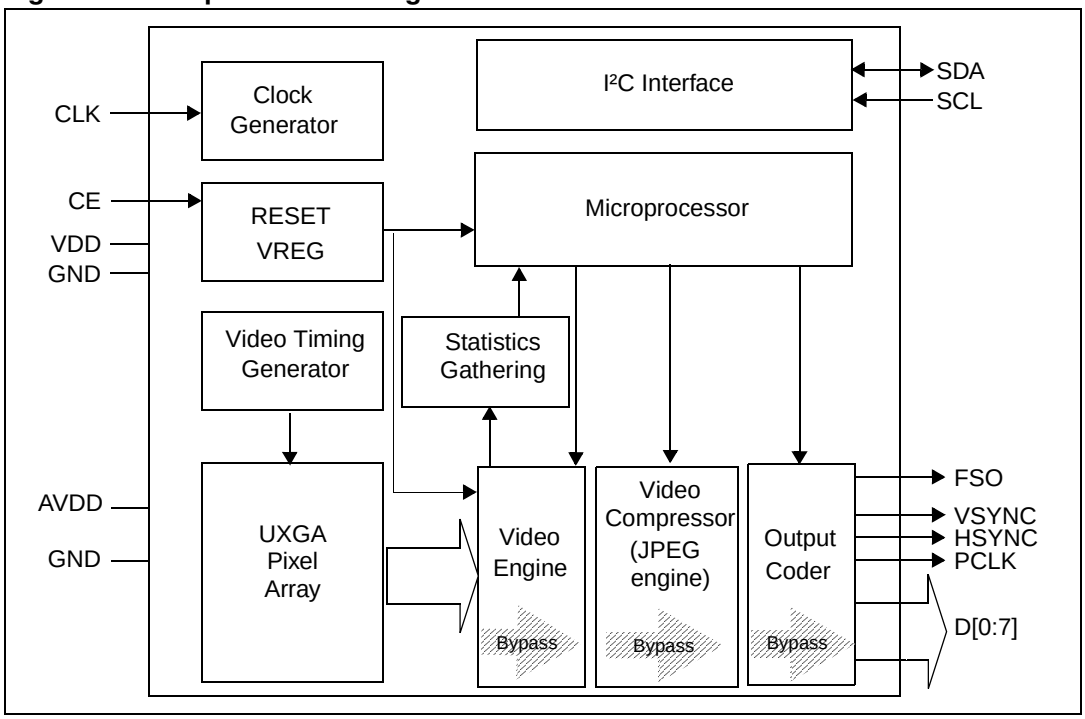
1. The two GND pins are connected together on the device substrate.

2 Functional description

The VS6724 simplified block diagram is shown in *Figure 1*, with the following main blocks:

- UXGA-sized pixel array
- Video pipe
- Statistics gathering unit
- Clock generator
- Microprocessor
- Video timing generator

Figure 1. Simplified block diagram



2.1 Operation

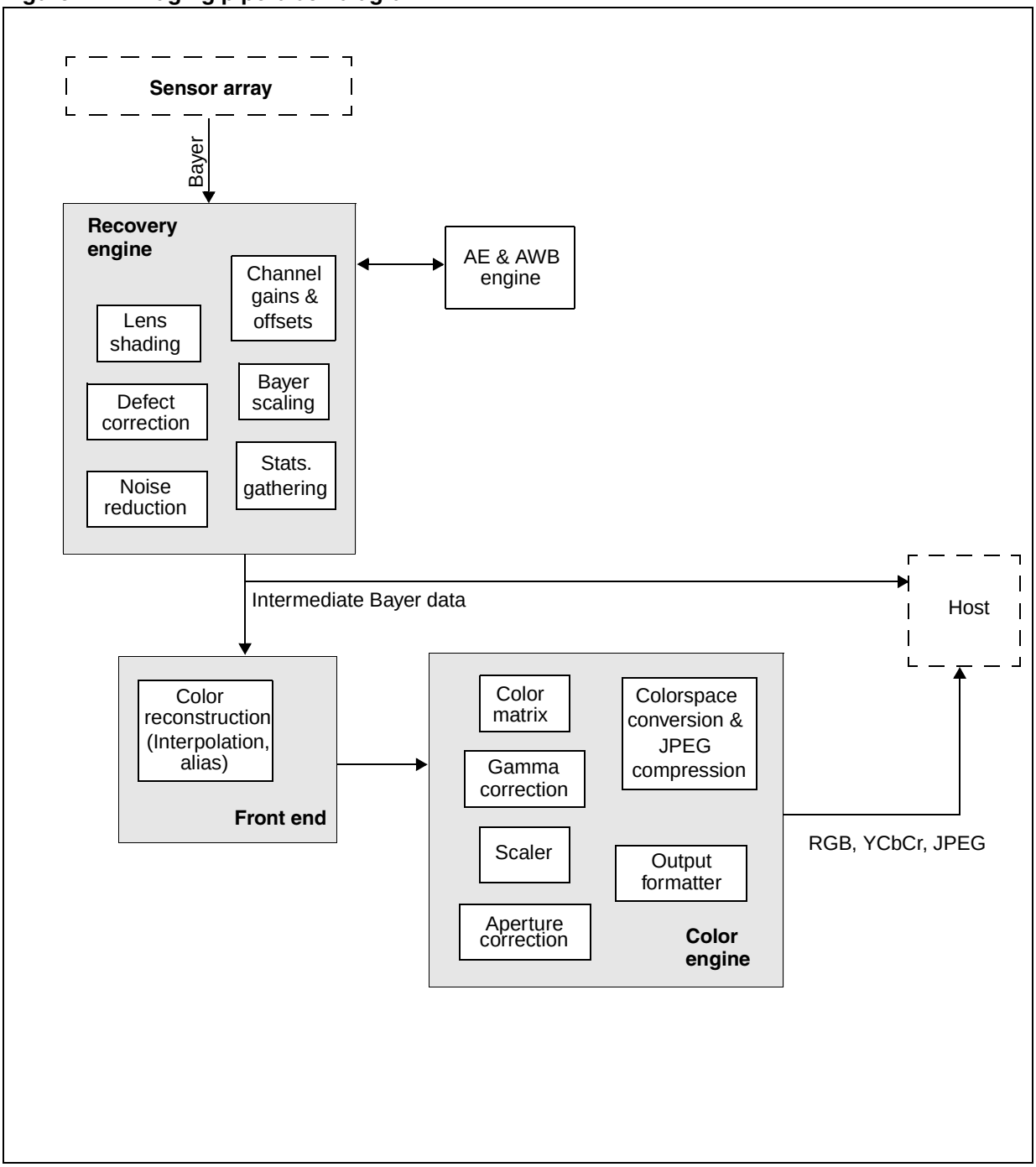
A video timing generator controls a UXGA-sized pixel array to produce raw Bayer images. The analogue pixel information is digitized and passed into the video pipe. The video pipe contains a number of different functions (explained in detail later). At the end of the video pipe, data is output to the host system over an 8-bit parallel interface along with qualification signals.

The whole system is controlled by an embedded microprocessor that is running firmware stored in an internal ROM. The external host communicates with this microprocessor over an I²C interface. The microprocessor does not handle the video data itself but is able to control all the functions within the video pipe. Real-time information about the video data is gathered by a statistics engine and is available to the microprocessor. The processor uses this information to perform real-time image control tasks such as automatic exposure control.

2.2 Imaging pipe diagram

The details of the imaging pipe are shown in *Figure 2*. The main functions contained within this diagram are detailed in this chapter.

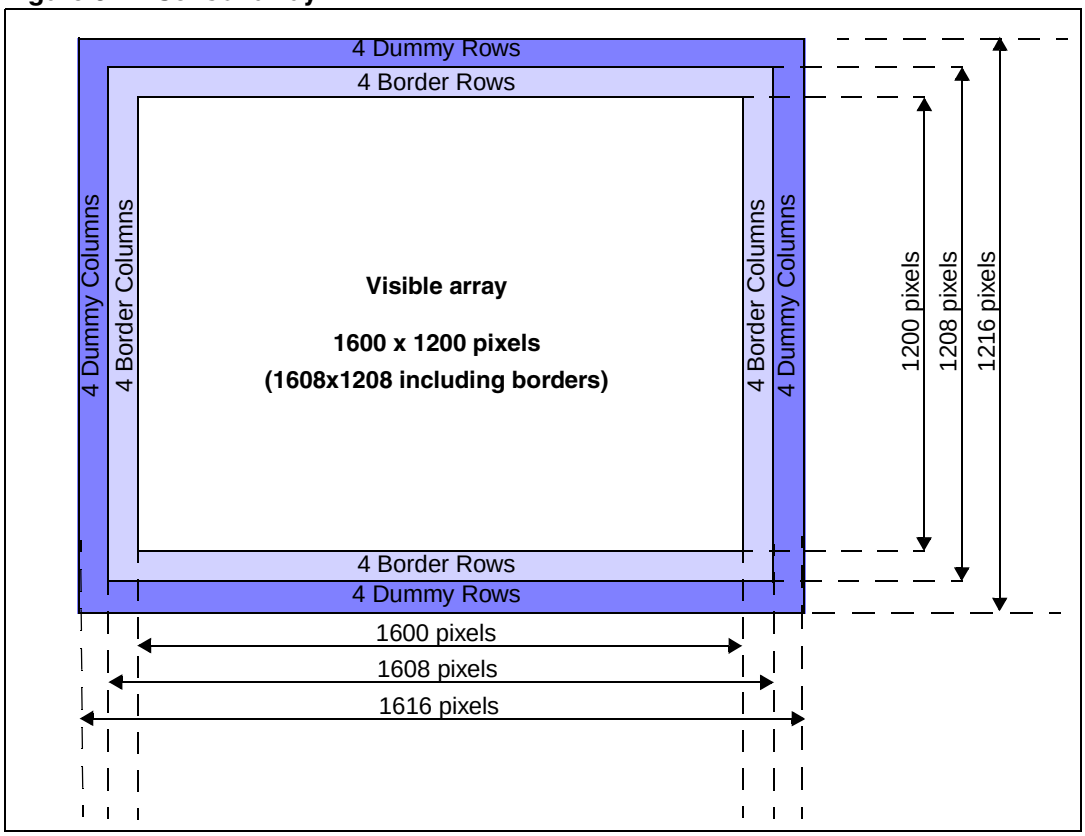
Figure 2. Imaging pipe block diagram



2.3 Sensor array

The VS6724 physical pixel array is 1616 x 1216 pixels (see *Figure 3*), with a pixel size of 2.2 µm by 2.2 µm. The image size read from the array is mode dependent. For UXGA mode the image size read from the array is 1608x1208. For SVGA mode it is 1616x1208.

Figure 3. Sensor array



2.3.1 Sensor mode control

The VS6724 can operate it's sensor array in two modes controlled by register SensorMode within *Mode setup*.

- SensorMode_UXGA - the full array can be read out at 30 fps and the VS6724 can achieve:
  - 30 fps JPEGs up to UXGA
  - 30 fps YCbCr or RGB up to SVGA
  - 15 fps YCbCr or RGB over SVGA
- SensorMode_SVGA_analogue binning - this is a reduced power mode which uses the full array and a technique of analogue binning output SVGA at up to 30 fps.

2.3.2 Horizontal mirror and vertical flip

The image data output from the VS6724 can be mirrored horizontally or flipped vertically (or both).

2.4 Recovery engine (RE)

The RE is used to process raw Bayer input from the sensor array to an intermediate Bayer which is correctly white balanced and noise reduced, and can be supplied to the interpolation engine.

2.4.1 Channel offsets

To achieve the desired black level of Bayer pixel data it is often necessary to apply an offset to the data. This module provides the functionality to apply color channel dependent offset on active pixel data, i.e. not dark or black data.

2.4.2 Channel gains

Color dependent gains are applied to active pixel data within this module as part of the automatic white balance function. The gain inputs are re-synchronized to the first active line.

2.4.3 Lens shading compensation

The lens shading is used to compensate for attenuation in the optical signal caused by primary and micro lens roll off. The module uses internal counters to determine the input pixel position in terms of imaging array, to which it applies a simple polynomial gain function. This gain is dependent on two coefficients which are system specific.

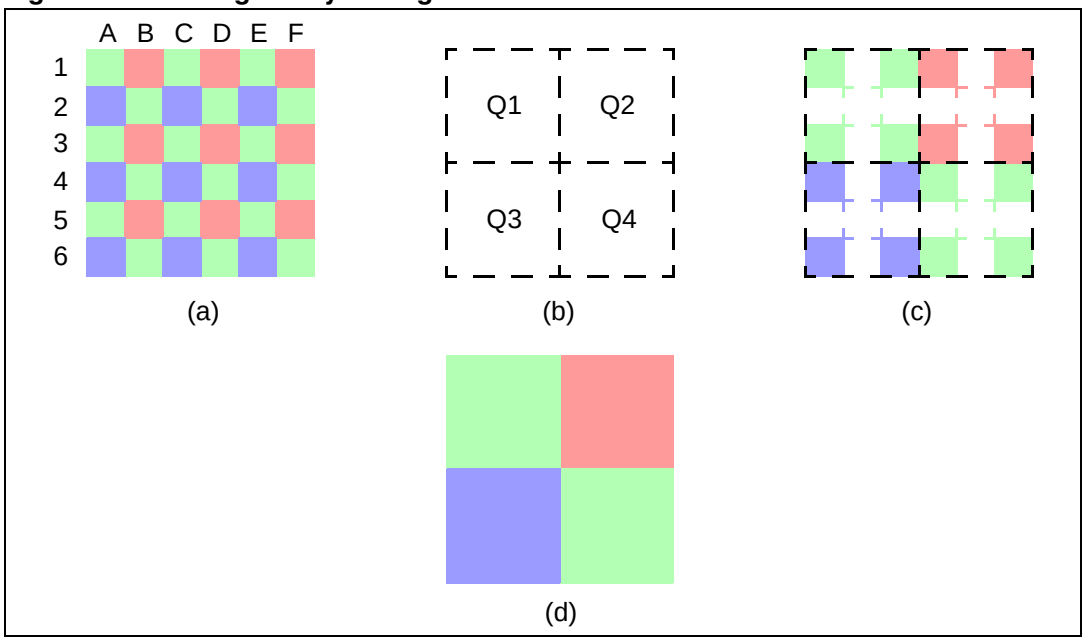
2.4.4 Bayer scaling

The RE architecture includes an optional Bayer scaler/smoothing module which produces a Bayer image fully compatible with existing image reconstruction techniques.

The scaling factor is fixed at downscale-by-3 (in both x & y), with the basic principle of sub-sampling from successive 3x3 blocks to produce a pseudo-Bayer pattern (2x2 Bayer pattern generated from 6x6 Bayer input as illustrated in *Figure 4*).

This module is intended for use during viewfinder modes to provide a lower power (and reduced quality) image for display on LCD.

Figure 4. Scaling of Bayer image



a) is a 6x6 pixel section of a Bayer patterned pixel array.

b) shows four 3x3 pixel regions which divide the 6x6 pixel array into quarters.

c) shows the pixels in each quarter used to calculate the sub-sampled pixel (and the spatial positioning of the sub-sampled pixel shown by the dashed colored line)

d) shows the final sub-sampled 2x2 Bayer data

The sub-sampled pixel value is the average value of the specific color pixels in each quarter.

Smoothing is required in order to regulate the data rate through the image reconstruction chain so as to match any system bandwidth limits.

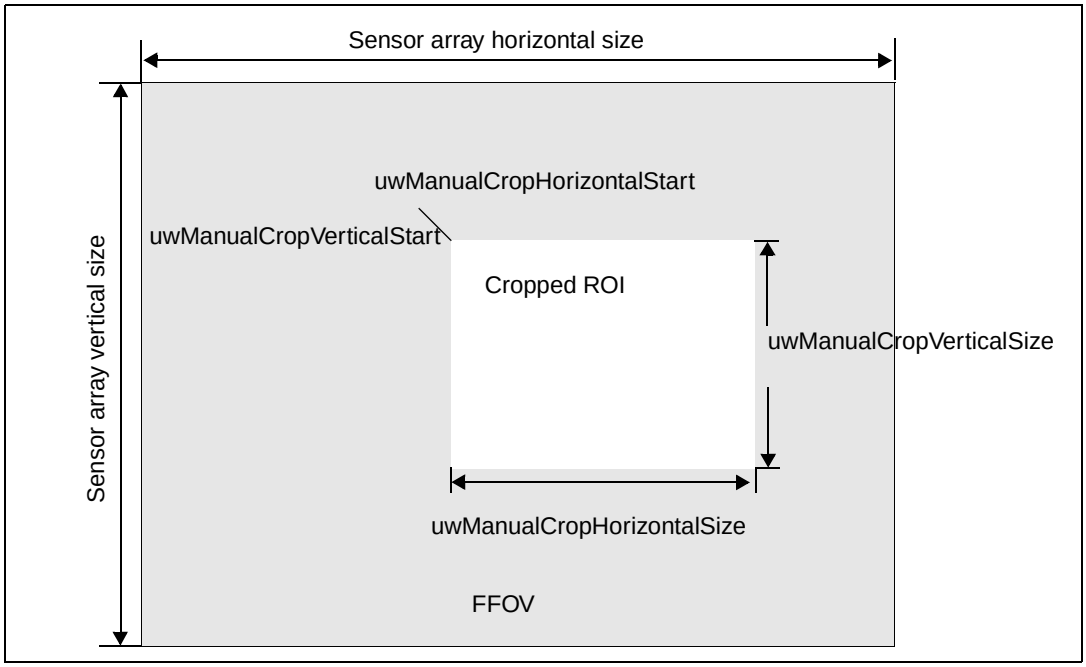
2.4.5 Cropping module

The VS6724 cropping module can be used to define a window of interest within the full UXGA array size, with all pixels falling outside this window of interest appearing as blanking.

The crop block is also used by the scaler to allow configuration of certain output frame dimensions.

The user can set a start location and the required output size. *Figure 5* shows the example with of the crop.

Figure 5. Crop controls



2.4.6 Zoom

The zoom function found in the pipe context control sections can be used to achieve a continuous zoom by simply selecting the commands zoom_in, zoom_out and zoom_stop. It is also possible to perform a single step of zoom by selecting Zoom_step_in and Zoom_step_out.

It is possible to zoom between the sensor size selected and the output size (the output size must be smaller that the sensor mode size, UXGA or SVGA).

2.4.7 Pan

It is possible to pan left, right, up and down when the output size selected is smaller than the sensor size selected (if the output size selected equals the sensor mode size then no pan can take place).

The pan step size in both the horizontal and vertical directions are selectable.

2.4.8 Derating

The VS6724 contains an internal derating module. This is designed to reduce the peak output data rate of the device by spreading the data over the whole frame period and allowing a subsequent reduction in output clock frequency.

The maximum achievable derating factor is x400 for an equivalent scale factor of x20 downscale. As a general rule the allowable derating factor is equal to the square of the scaling factor.

Note: The interline period is not guaranteed to be consistent for all derating ratios. This means the host capture system must be able to cope with use of the sync signals or embedded codes rather than relying on fixed line counts.

2.4.9 Defect correction 1 & 2

The input to the defect correction filters is a 5x5 pixel matrix from which a centre pixel and a neighborhood of 8 pixels is extracted. The 8 pixel neighborhood is used to determine the validity of the corresponding centre pixel. This is achieved using the ranked output of a Batcher-Banyan sort architecture. The defect correction filters perform both detection and weighted correction on all data. Defect correction 1 is weighted to work horizontally and vertically. Defect correction 2 is weighted to operate on the diagonal elements.

2.4.10 Spatial noise reduction

The noise reduction module implements an algorithm based on the human-visual system and adaptive pixel filtering that reduces perceived noise in an image whilst maintaining areas of high definition.

2.4.11 Interpolation

The interpolation module converts Bayer pixel data to RGB and applies an anti-alias filter to the data. It also generates the sharp data used in the aperture correction block.

2.5 Color engine (CE block)

The role of the CE is to process the RGB data from the recovery engine into a specific format requested by the host, e.g. RGB565 for LCD display, or YCbCr for the JPEG encoder.

2.5.1 Color matrix

A matrix color correction transform is performed on the outputs of the scaler, the matrix is detailed in *Figure 6*.

Figure 6. Color matrix

$$\begin{bmatrix} rcof00 & rcof01 & rcof02 \\ rcof10 & rcof11 & rcof12 \\ rcof20 & rcof21 & rcof22 \end{bmatrix}$$

The matrix equations are:

Ro = Rin.rcof00 + Gin.rcof01 + Bin.rcof02

Go = Rin.rcof10 + Gin.rcof11 + Bin.rcof12

Bo = Rin.rcof20 + Gin.rcof21 + Bin.rcof22

Note: The neutral-preserving (“eigen-grey”) property of the matrix can be achieved by ensuring the matrix rows sum to unity.

2.5.2 Aperture correction

The aperture correction module is used to add a certain amount of sharpening to the scaled RGB, it can be programmed to reduce as the light level drops. The aperture correction module also includes a function known as coring which allows control of the minimum magnitude which any edge on the image must exceed before sharpening is applied. This ensures that small edges such as noise are not amplified by the sharpening. The coring can be programmed to increase as the light level drops.

2.5.3 Special effects

The special effects module is used to apply simple transforms onto the input data. The supported effects are: Black and White, Hue, Sepia, Negative, Antique, sketch, caricature and Emboss. There are also a number of color effects which can only be applied to YCbCr data.

2.5.4 Gamma correction

The gamma module is a gain curve applied to each of the three image components from the sharpening module. The shape of the gain curve is fully programmable.

2.5.5 YCbCr conversion

This module performs color space conversion from RGB to YCbCr. It is used to control the contrast and color saturation of the output image as well as the fade to black feature.

2.5.6 Dither

The dither block is used in RGB modes to reduce contouring in the image when the data is truncated to lower ranges. This is achieved by determining the error introduced by truncation and adding/subtracting this residual energy to/from the subsequent pixel.

2.6 Video compression module

The JPEG encoder receives YCbCr data from the image reconstruction engine and encodes it using baseline sequential JPEG technique. It outputs a standard JPEG data stream with all the required markers and segments required by decoders.

2.6.1 Features

- Supports YCbCr 422 as input data format.
- Converts YCbCr 422 to 420 format internally, format selectable.
- Baseline ISO/ IEC 10918-1 JPEG compliance.
- Programmable quantization tables.
- Capable of streaming the UXGA image at a rate of 30 frames per second.
- Manual / automatic compression control via scaled quantization tables.
- Automatic compression control targeted at file size and/or peak bit-rate.
- Progressive high-frequency roll-off option for increased compression.
- Intraframe rate-control via zig-zag sequence truncation.

Quantizer

Quantizer scales the DCT coefficients with programmable luminance and chrominance Q-tables. These Q-tables are programmed according to image characteristics, and modified by the squeeze setting. The larger the squeeze setting, the more severe the quantization and the greater the compression achieved.

Squeeze values can be manually set via the 8-bit register user_squeeze, or automatically set, and dynamically adjusted, by the autosqueeze module.

Entcoder16

The entcoder16 (entropy coder with 16-bit output) converts the quantized data stream to a symbol stream, including differential encoding of DC samples and run-length encoding of zero-valued AC samples according to the JPEG standard, and finally Huffman-encodes the symbol stream using hardwired baseline JPEG tables to compress the image. It also includes the JPEG header and embeds restart interval markers (if requested by nonzero restartinterval setting) as per the JPEG standard.

Autosqueeze

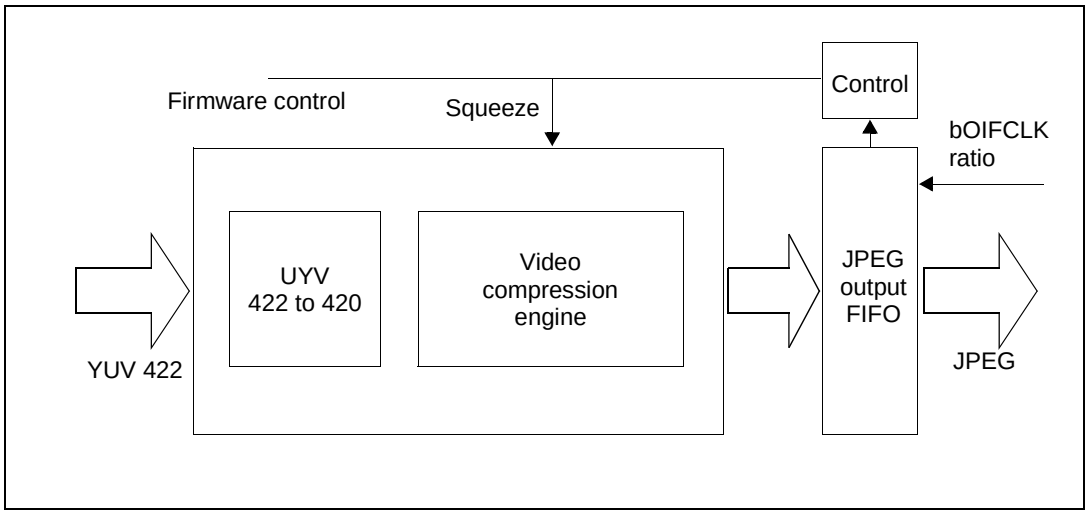
Autosqueeze is the VC's in-built dynamic compression control system. It adjusts the squeeze parameter based on measurements of two key criteria derived from the previous frame: output file size, and FIFO fullness. Squeeze is subsequently driven iteratively towards the optimal value, satisfying the two conditions:

1. FIFO stability, i.e. fullness has not exceeded a user-defined dead-zone, AND
2. output file size <= a user-defined target.

2.6.2 JPEG compression

At the simplest level the video compressor has one control which is the Squeeze setting: this tells the system how hard to compress. A change in compression is achieved by scaling the quantization tables.

Figure 7. QCLK options



The FIFO shown in the figure above is being constantly monitored with fixed break points, if the FIFO fill state passes one of these break points the squeeze value is increased or decreased.

The amount of data in the FIFO is dependant on a number of factors

- The squeeze factor set
- The complexity of the scene
- The speed at which the FIFO is being emptied (bOIFClkRatio)

2.6.3 JPEG squeeze controls

The strength of compression or the level of squeeze operated on the image (and therefore the quality of the final image) can be controlled in two ways via the bJpegSqueezeSettings register (note each context can be set independently);

`bJpegSqueezeSettings0`

0x00 SET_USER_SQUEEZE_MODE

0x01 SET_AUTO_SQUEEZE_MODE

In auto squeeze mode, the JPEG compression engine is trying to achieve a target file size and constantly changes the compression ratio. This is the preferred mode when using the output JPEG to create a video, or in preview mode.

`wJpegTargetFileSize {0x03c3, 0x03c4}`: Input required size in KBytes

In user squeeze mode, the JPEG compression engine will use a constant level of squeeze. This is recommended when trying to capture a high quality image and for snapshot or flashgun modes. For ease of use, the level of squeeze can be adjusted for each pipe context between a high, medium or low level. The value of these squeeze settings can be configured by using the following registers:

`bJpegImageQuality0`

0x00 High quality. Value set in `bHiSqueezeValue` {0x2508}

0x01 Medium quality. Value set in `MedSqueezeValue` {0x250a}

0x02 Low quality. Value set in `bLowSqueezeValue` {0x250c}

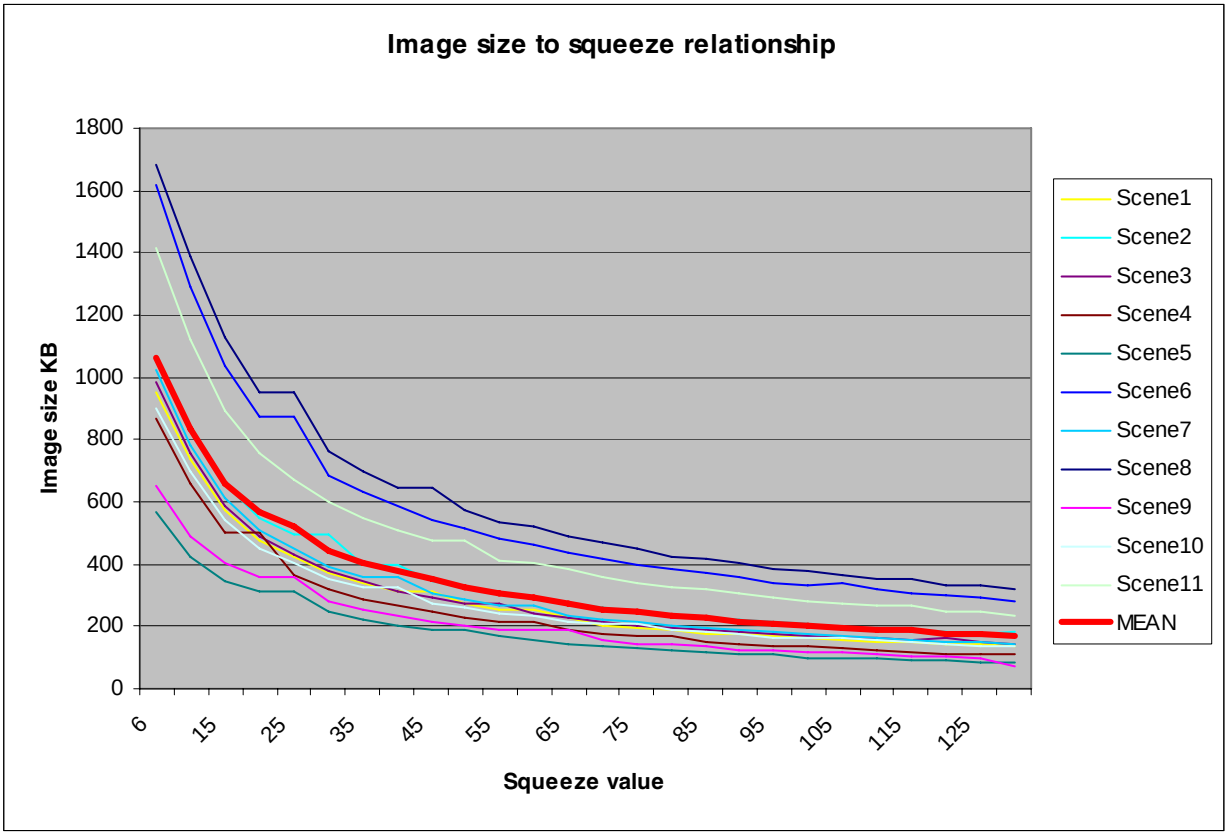
It is also possible to select between the YCbCr formats input to the video compression engine:

`bJpegImageFormat`

0x00 YCbCr 422

0x01 YCbCr 420

Figure 8. JPEG Compression image size to squeeze value relationship



2.6.4 FIFO control

The rate which the FIFO is emptied (and therefore the output PCLK frequency) can be controlled using the bOIFClkRatio register. Changing the FIFO readout rate will in turn change the FIFO's fullness and therefore the squeeze factor used. *Section 2.11.2: PLL operation on page 39* describes the calculations for $PCLK_{max}$ frequency. This frequency is divided by the bOIFClkRatio to give the max JPEG PCLK output from the device:

bOIFClkRatio {0x2514}

1. PCLK is 2 times slower
2. PCLK is 4 times slower
3. PCLK is 8 times slower
4. PCLK is 16 times slower
5. PCLK is 32 times slower
6. PCLK is 64 times slower

2.6.5 JPEG output as a conventional frame

To keep compatibility with conventional frame format, the JPEG frame is output in a series of packets targeted to look like image lines. The Hsync envelopes each packet (line) of data and the Vsync envelopes the complete frame. The start of the frame is indicated by a transition of the Vsync and Hsync. This will occur when the output FIFO has accumulated enough JPEG data for the first line (default line size 512 = 1KBytes). When this line is output, it will then output blocks of blank data (16Byte blocks) until the next 512Bytes of data is available.

The number of bytes per line can be programmed using the following registers, both 16 bit registers should be programmed with the same value:

- wLineLength {MSB 0x2511, LSB 0x2512} default = 512
- wThres {MSB 0x251b, LSB 0x251c}

Note: The number of PCLKs in a line is equal to 2 x the line length and the maximum line length is 2K.

In most cases the JPEG data will not fill the last line of data and padding data is added after the JPEG end of image marker FFD9h filling the remainder of the last line. Note that it is possible that the JPEG data will exactly fill the line and no padding is required.

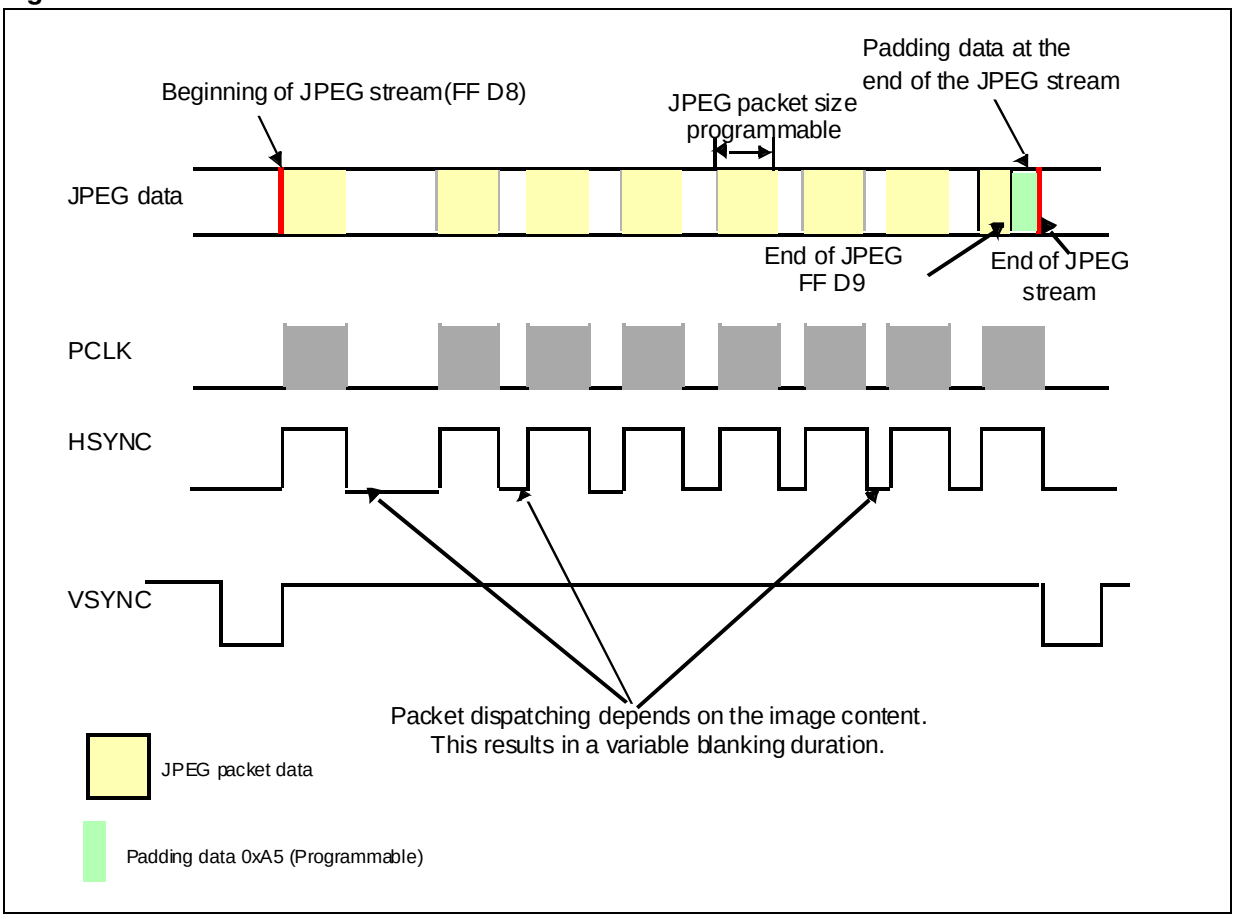
The value of padding bytes can be programmed using the following registers, both registers should be programmed with the same value;

- bJPEG_FILL_VAL {0x23b4} default = 0xA5
- bJPEG_PADDING {0x23b6}

It is not possible to control the timing of the Vsync and Hsync signals in JPEG mode. You can however change the polarity of the signals and select if the Hsync is present or not during the interframe period.

PCLK can be made to be present only during active JPEG data, and therefore not present during the interline or interframe periods. The rising edge of PCLK is always half clock delayed from both Hsync and Vsync.

Figure 9. JPEG frame



Note: The minimum inter-packet (line) blanking time is 16 clock cycles, and is always a multiple of 16 clock cycles

Table 2. JPEG data embedded markers

Marker function	Name	Value
Start of image	SOF	FFD8
Define Quantization tables	DQT	FFDB
State of frame for baseline	DCT SOF	FFC0
Define Huffman Tables	DHT	FFC4
Start of Scan	SOS	FFDA
End of image	End of image	FFD9

2.7 Microprocessor functions

The microprocessor inside the VS6724 handles the I²C communication with the host processor. From this communication it allows the host control over the device via the User interface registers.

Dark calibration: The microprocessor uses information from the array to ensure that the optimal and consistent ‘black’ level is achieved from the output.

Automatic exposure control: Using the information output by the statics gathering engine the appropriate exposure settings for the scene is calculated and using a combination of exposure control, analogue gain and digital gain the device will outputs a correctly exposed image. The host can program the exposure target within a range of EV values or can control the system manually setting a known Exposure time, analogue gain and digital gain.

Flicker cancellation: The 50/60 Hz flicker frequency present in many lighting sources is visible when the integration time is not a integer multiple of this frequency. The VS6724 will adjust the integration time to ensure that the flicker effect is minimized. Note that flicker is present when using a integration time shorter than the period of the lighting frequency (less than 8.333 ms for 60 Hz, and less than 10 ms for 50 Hz).

Automatic white balance: Using the information output by the statics gathering engine the microprocessor adjusts the gains applied to the individual color channels in order to achieve a correctly color balanced image. In addition to the standard white balance operation a constrainer operates on this information which ensures that the white balance achieved fits within the expected color temperature of real life illuminants.

Frame rate control: VS6724 contains a firmware based programmable timing generator. This automatically designs internal video timings, PLL multipliers, clock dividers etc. to achieve a target frame rate with a given input clock frequency.

Optionally an automatic frame rate controller can be enabled. This system examines the current exposure status and adapts the frame rate based on this information. This function is typically useful in low-light scenarios where reducing the frame rate extends the useful integration period. This reduces the need for the application of analog and digital gain and results in better quality images.

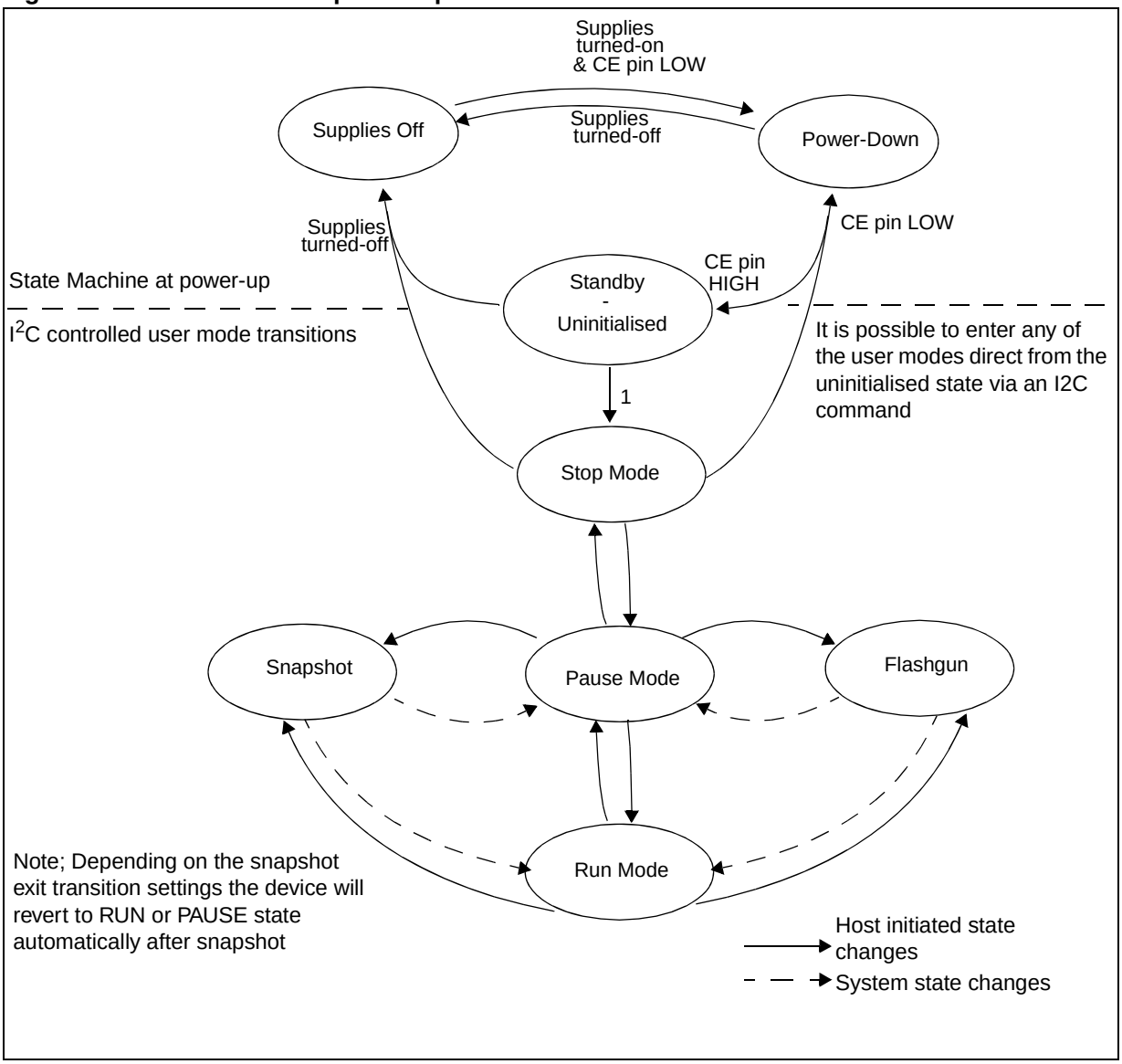
Active noise management: The microprocessor is able to modify certain video pipe functions according to the current exposure settings determined by the automatic exposure controller. The main purpose of this is to improve the noise level in the system under low lighting conditions. Functions which ‘strength’ is reduced under low lighting conditions (e.g. aperture correction) are controlled by ‘dampers’. Functions which ‘strength’ is increased under low lighting conditions are controlled by ‘promoters’. The fade to black operation is also controlled by the microprocessor

Fade to Black: Using programmable levels the microprocessor will fade the output signal to black, this ensures that under the darkest conditions when the image is not of sufficient quality the device will output black. This operation is achieved by scaling the RGB to YCbCr matrix.

2.7.1 Operational mode control

The microprocessor allows high level control of the modes in which the VS6724 can operate, this avoids the need for the host to be concerned with the complexity of controlling the timing or power management during mode changes.

Figure 10. State machine at power -up and user mode transitions



The power down mode is entered and exited by driving the hardware CE signal. Transitions between all other modes are initiated by I²C transactions from the host system or automatically after time-outs.

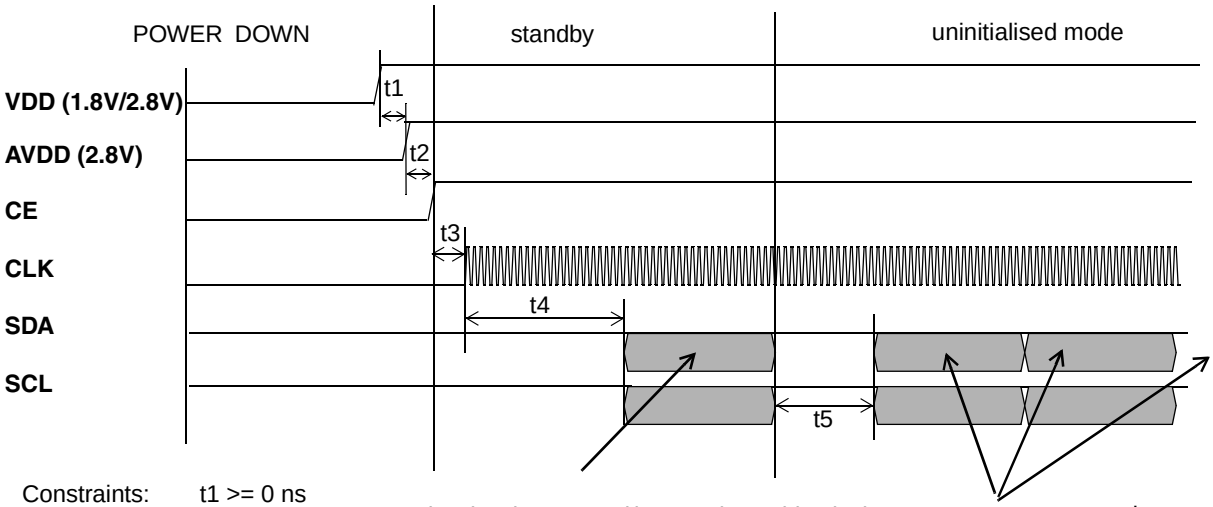
Power Down/Up: The power down state is entered when CE is pulled low or the supplies are removed.
During the power-down state (CE = logic 0)

- The internal digital supply of the VS6724 is shut down by an internal switch mechanism. This method allows a very low power-down current value.
- The device input / outputs are fail-safe, and consequently can be considered high impedance.

During the power-up sequence (CE = logic 1)

- The digital supplies must be on and stable.
- The internal digital supply of the VS6724 is enabled by an internal switch mechanism.
- All internal registers are reset to default values by an internal power on reset cell.

Figure 11. Power up sequence



Constraints:
t1 >= 0 ns
t2 >= 0 ns
t3 >= 0 ns
t4 >= 200 µs
t5 >= 7 ms (based on 12 MHz external clock frequency, delay is inversely proportional to external clock. Max delay for 6 MHz = 14 ms.)

STANDBY mode: The VS6724 enters STANDBY mode when the CE pin on the device is pulled HIGH. Power consumption is very low, most clocks inside the device are switched off. In this state I²C communication is possible when CLK is present and when the microprocessor is enabled.

All registers are reset to their default values. The device I/O pins have a very high-impedance.

Uninitialised = RAW: The initialize mode is defined as supplies present, the CE signal is logic 1 and the microcontroller clock has been activated.

During initialize mode the device firmware may be patched. This state is provided as an intermediary configuration state and is not central to regular operation of the device. The analogue video block is powered down, leading to a lower global consumption

STOP mode: This is a low power mode. The analogue section of the VS6724 is switched off and all registers are accessed over the I²C interface. A run command received in this state automatically sets a transition through the Pause state to the run mode.

If a STOP command is issued while streaming an incomplete frame may be generated, it is recommended that a Pause command is issued and a timeout set in **bTimeToPowerdown** to force an automatic transition to STOP.

The analogue video block is powered down, leading to a lower global consumption.

Pause mode: In this mode all VS6724 clocks are running and all registers are accessible but no data is output from the device. The device is ready to start streaming but is halted. This mode is used to set up the required output format before outputting any data.

The analogue video block is powered down, leading to a lower global consumption

Note: The PowerManagement register can be adjusted in PAUSE mode but has no effect until the next RUN to PAUSE transition.

RUN mode: This is the fully operational mode. In running mode the device outputs a continuous stream of images, according to the set image format parameters and frame rate control parameters. The image size is derived through downscaling of the UXGA image from the pixel array.

ViewLive: this feature allows different sizes, formats and reconstruction settings to be applied to alternate frames of data, while in run mode.

Snapshot mode: The device can be configured to output a single frame according to the size, format and reconstruction settings in the relevant video pipe context. In normal operation this frame will be output, once the exposure, white balance and dark-cal systems are stable. To reduce the latency to output, the user may manually override the stability flags.

The snapshot mode command can be issued in either Run or Stop mode and the device will automatically return previous state after the snapshot is taken. The snapshot mode must not be entered into while ViewLive is selected.

FLASHGUN mode: In flashgun mode, the array is configured for use with an external flashgun. A flash is triggered and a single frame of data is output and the device automatically switches to Pause Mode.

VS6724 supports the following flashgun configurations:

- Torch Mode - user can manually switch on/off the FSO IO pin via a register setting. Independent of mode.
- Pulsed Mode - the flash output is synchronized to the image stream. There are two options available:
  - Pulsed flash with snapshot. Device outputs a single frame synchronized to flash.
  - Pulsed flash with viewfinder. Device outputs a flash pulse synchronized to a single frame in the image stream.
  - In the pulsed mode there are two possible pulse configurations:
  - Single pulse during the interframe period when all image lines are exposed. This is suitable for SCR and IGBT flash configurations. The falling edge of the pulse can be programmed to vary the width of the pulse.
  - Single pulse over entire integration period of frame. This is suitable for LED flash configurations.

Mode transitions

Transitions between operating modes are normally controlled by the host by writing to the *Host interface manager control* register. Some transitions can occur automatically after a time out. If there is no activity in the Pause state then an automatic transition to the Stop state occurs. This functionality is controlled by the Power management register, writing 0xFF disables the automatic transition to Stop.

The users control allows a transition between Stop and Run, at the state level the system will transition through a Pause state.

2.8 Video pipe context configuration

2.8.1 Video pipe setup

The VS6724 has a single video pipe, the control of this pipe can be loaded from either of two possible setups Pipecontext0 and PipeContext1;

PipeContext0 and *PipeContext 1*, control the operations shown below:

- Image size
- Zoom control
- Pan control
- Crop control
- Image format (YCbCr 4:2:2, RGB565, etc....)
- Image controls (Contrast, Color saturation, Horizontal and vertical flip)

2.8.2 Context switching

In normal operation, it is possible to control which pipe context is used and to switch between contexts without the need to stop streaming, the change will occur at the next frame boundary after the change to the register has been made.

For example this function allows the VS6724 to stream an output targeting a display (e.g. QQVGA RGB 444) then switch to capture an image (e.g. UXGA JPEG) with no need to stop streaming or enter any other operating mode.

It is important to note the output size selected for both pipe context must be appropriate to the sensor mode used, i.e. to configure PipeContext0 to QQVGA and PipeContext1 to UXGA the sensor mode must be set to UXGA.

The register *Mode setup* allows selection of the pipe context, by default the PipeContext0 is used.

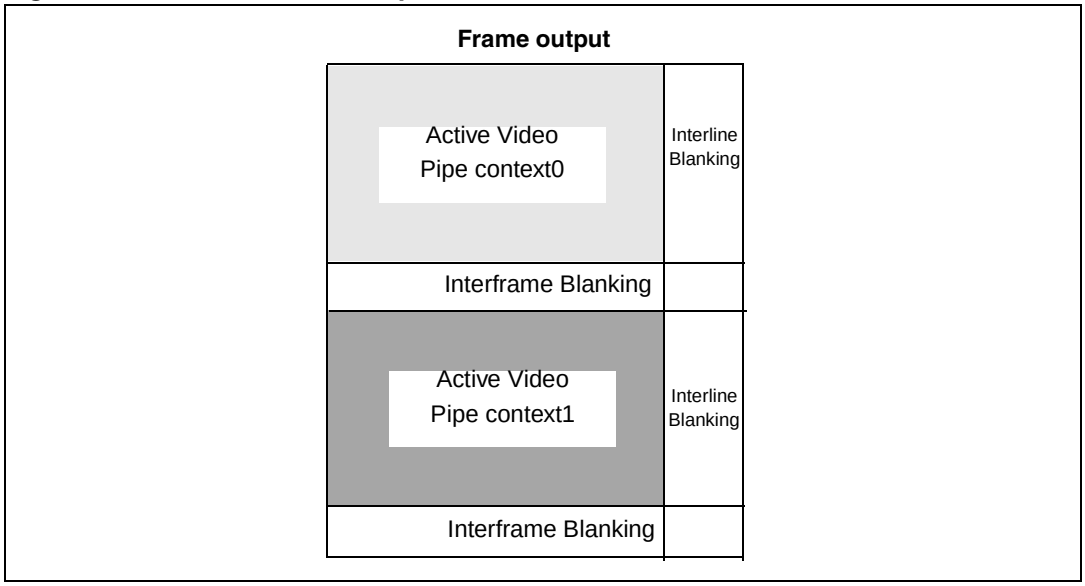
2.8.3 ViewLive operation

ViewLive is an option which allows a different pipe context to be applied to alternate frames of the output data.

The controls for VIewLive function are found in the register bank where the fEnable register allows the host to enable or disable the function and the binitialPipecontext register selects which pipe context is output first.

When ViewLive is enabled the output data switches between *PipeContext0* and *PipeContext 1* on each alternate frame.

Figure 12. ViewLive frame output format



2.9 Output formats

2.9.1 Image size

An output frame consists of a number of active lines and a number of interframe lines. Each line consists of embedded line codes (if selected), active pixel data and interline blank data. Note that by default the interline blanking data is *not* qualified by the PCLK and therefore is not captured by the host system.

The image size can be either the full output from the sensor, depending on sensor mode, or a scaled output, The output image size can be chosen from one of 9 pre-selected sizes or a manual image size can be used.

The VS6724 supports the following data formats:

- JPEG (4:2:2, 4:2:0)
- YCbCr 4:2:2
- YCbCr 4:0:0
- RGB565
- RGB444 (encapsulated as 565)
- RGB444 (zero padded)
- Bayer 10-bit
- Bayer 8-bit

The required data format is selected using the bdataFomat control found in the pipe context registers. The various options available for each format are controlled using the bRgbsetup and bYCbCrSetup registers found in the *Dither control* registers.

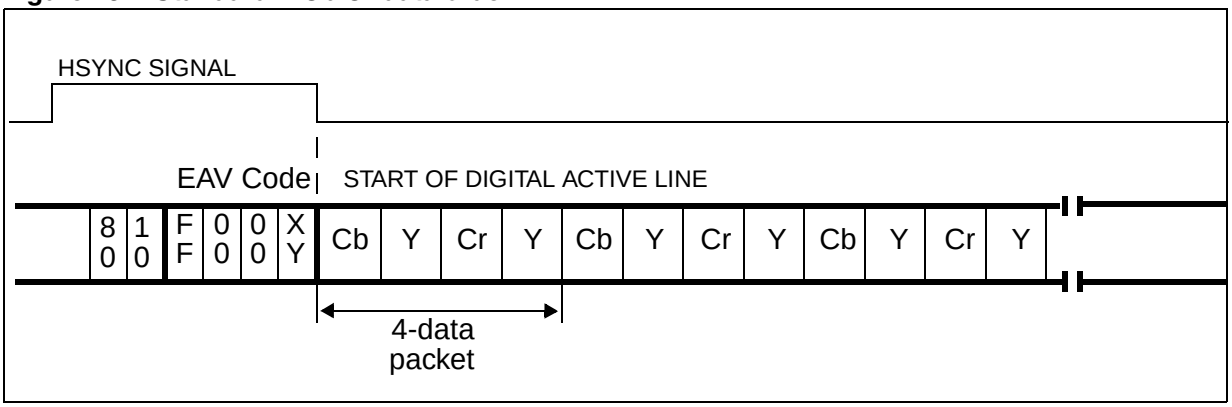
2.9.2 Line / frame blanking data

The values which are output during line and frame blanking are an alternating pattern of 0x10 and 0x80 by default. These values may be changed by writing to the BlankData_MSB and BlankData_LSB registers in the *Dither control* bank.

2.9.3 YCbCr 4:2:2 data format

YCbCr 422 data format requires 4 bytes of data to represent 2 adjacent pixels. ITU601-656 defines the order of the Y, Cb and Cr components as shown in *Figure 13*.

Figure 13. Standard Y Cb Cr data order



The VS6724 bYCbCrSetup register can be programmed to change the order of the components as follows:

Figure 14. Y Cb Cr data swapping options register bYCbCrSetup

	Bit [1] Y first	Bit [0] Cb first	Components order in 4-byte data packet 1st	2nd	3rd	4th
	1	1	Y	Cb	Y	Cr
DEFAULT	0	1	Cb	Y	Cr	Y
	1	0	Y	Cr	Y	Cb
	0	0	Cr	Y	Cb	Y

2.9.4 YCbCr 4:0:0

The ITU protocol allows the encapsulation of various data formats over the link. The following data formats are also proposed encapsulated in ITU601-656 protocol:

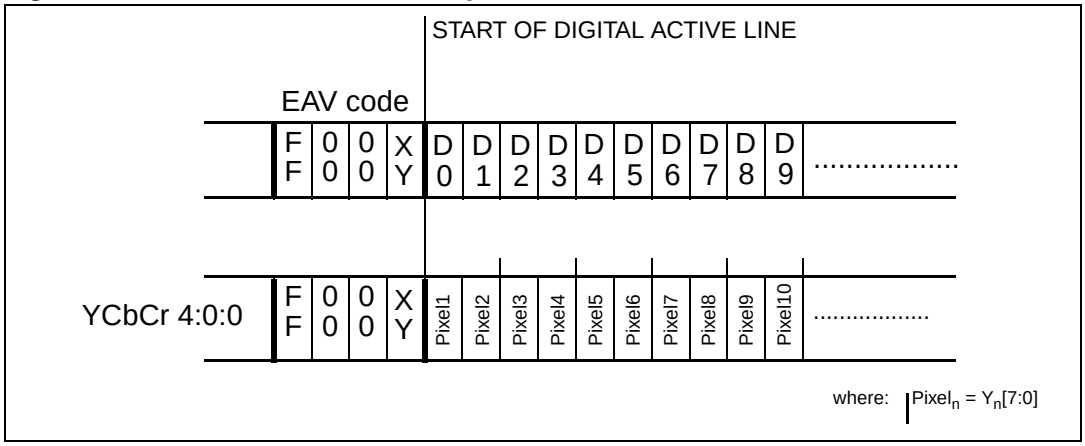
- YCbCr 4:0:0 - luminance data channel

This is done as described in *Figure 15*. In this output mode the output data per pixel is a single byte. Therefore the output PCLK and data rate is halved.

It is possible to reverse the overall bit order of the component through a register programming.

Note: False synchronization codes are avoided in the LSByte by adding or subtracting a value of one, dependent on detection of a 0 code or 255 code respectively.

Figure 15. YCbCr 4:0:0 format encapsulated in ITU stream



See *Dither control* for user interface control of output data formats.

2.9.5 RGB and Bayer 10 bit data formats

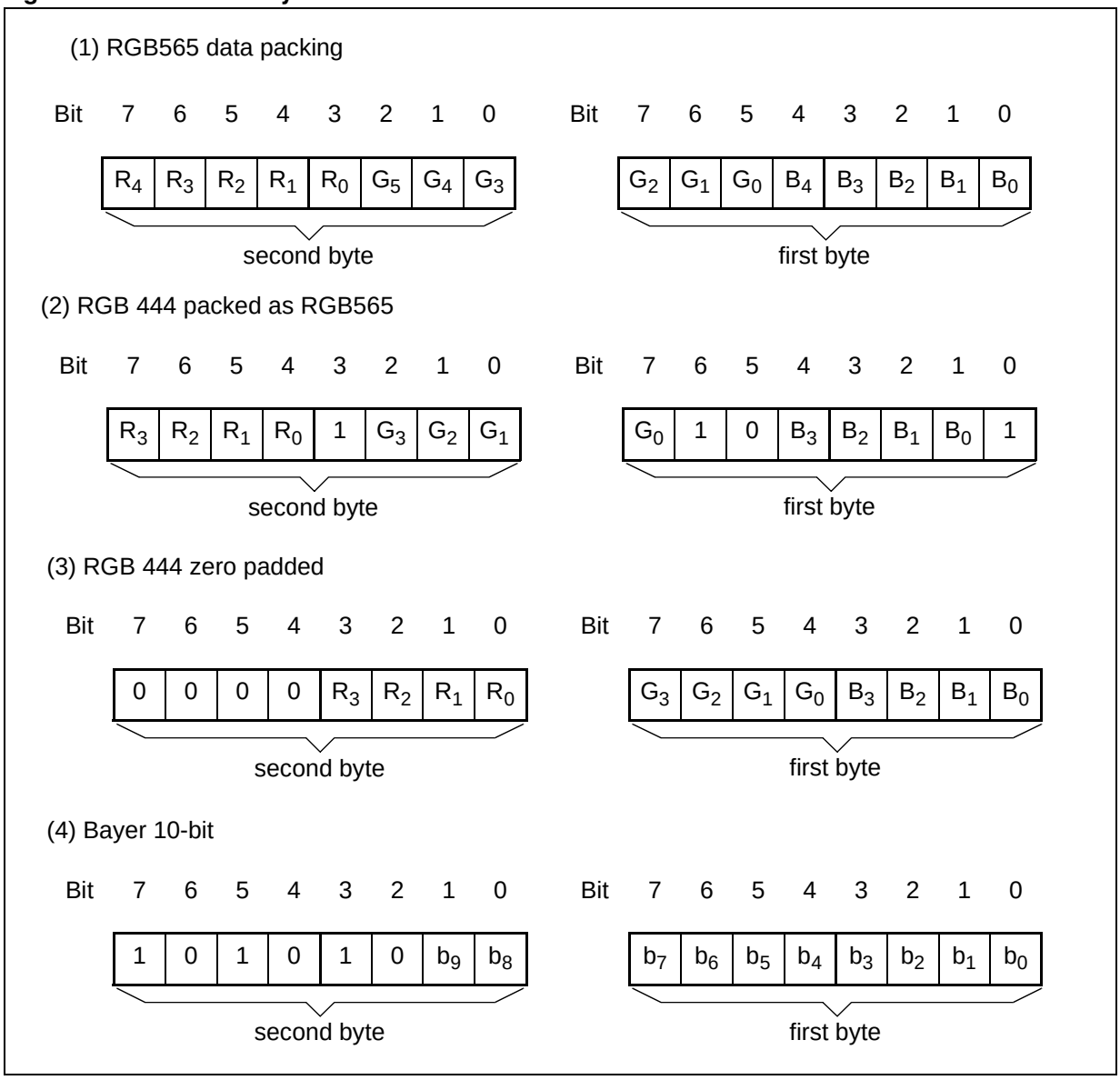
The VS6724 can output data in the following formats:

- RGB565
- RGB444 (encapsulated as RGB565)
- RGB444 (zero padded)
- Bayer 10-bit

Note: Pixels in Bayer 10-bit data output are defect corrected, correctly exposed and white balanced. Any or all of these functions can be disabled.

In each of these modes 2 bytes of data are required for each output pixel. The encapsulation of the data is shown in *Table 16*.

Figure 16. RGB and Bayer data formats



Manipulation of RGB data

It is possible to modify the encapsulation of the RGB data in a number of ways:

- swap the location of the RED and BLUE data
- reverse the bit order of the individual color channel data
- reverse the order of the data bytes themselves

Dithering

An optional dithering function can be enabled for each RGB output mode to reduce the appearance of contours produced by RGB data truncation. This is enabled through the DitherControl register.

2.9.6 Bayer 8-bit

The ITU protocol allows the encapsulation of various data formats over the link. The following data formats are also proposed encapsulated in ITU601-656 protocol:

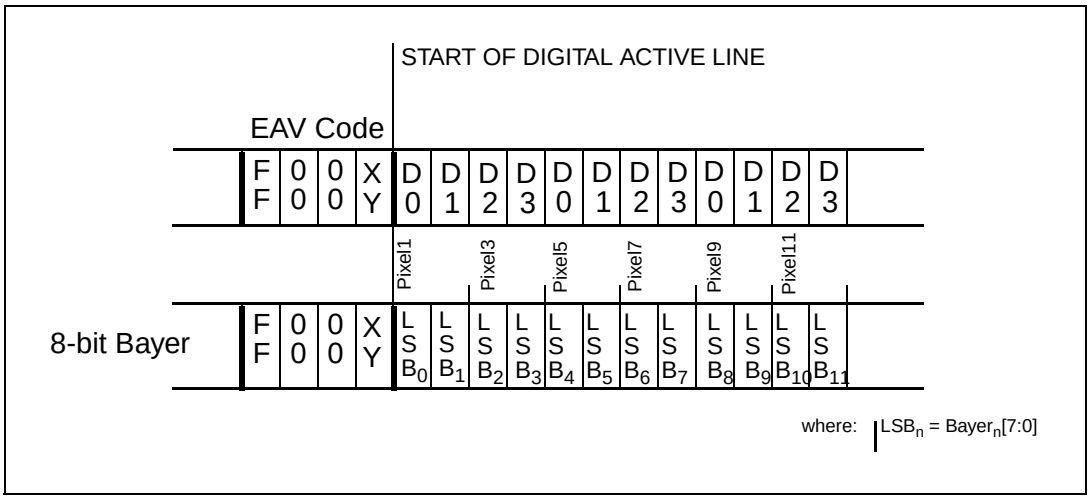
- Truncated from 10-bit

The output is shown in *Figure 17*. In this output mode a single byte is output data per pixel, therefore the output PCLK and data rate is half that of the 10bit modes.

It is possible to reverse the overall bit order of the individual Bayer pixels through a register programming.

Note: False synchronization codes are avoided in the LSByte by adding or subtracting a value of one, dependent on detection of a 0 code or 255 code respectively.

Figure 17. Bayer 8 output



2.10 Data synchronization methods

External capture systems can synchronize with the data output from VS6724 in one of two ways:

1. Synchronization codes are embedded in the output data
2. Via the use of two additional synchronization signals: VSYNC and HSYNC

Both methods of synchronization can be programmed to meet the needs of the host system.

2.10.1 Embedded codes

The embedded code sequence can be inserted into the output data stream to enable the external host system to synchronize with the output frames. The code consists of a 4-byte sequence starting with 0xFF, 0x00, 0x00. The final byte in the sequence depends on the mode selected.

Two types of embedded codes are supported by the VS6724: Mode 1 (ITU656) and Mode 2. The bSyncCodeSetup register is used to select whether codes are inserted or not and to select the type of code to insert.

When embedded codes are selected each line of data output contains 8 additional clocks: 4 before the active video data and 4 after it.

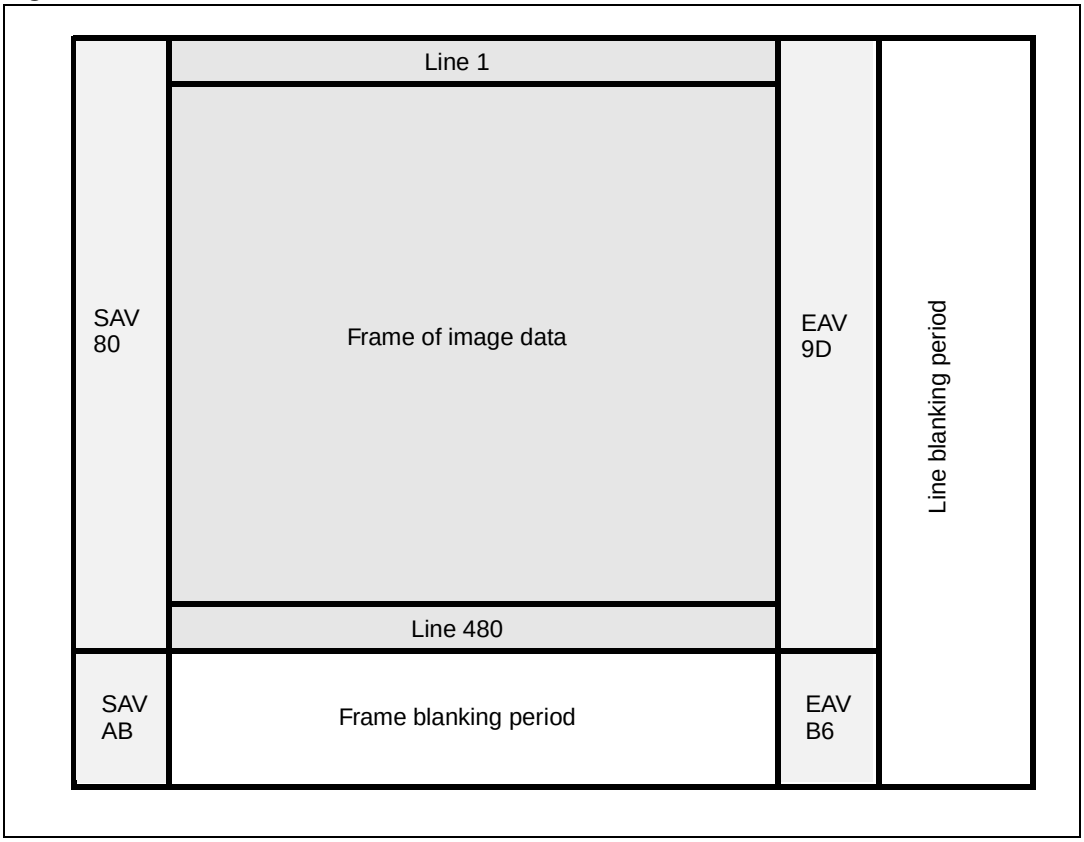
Prevention of false synchronization codes

The VS6724 is able to prevent the output of 0xFF and/or 0x00 data from being misinterpreted by a host system as the start of synchronization data. This function is controlled the bCodeCheckEnable register.

2.10.2 Mode 1 (ITU656 compatible)

The structure of an image frame with ITU656 codes is shown in *Figure 18*.

Figure 18. ITU656 frame structure with even codes



The synchronization codes for odd and even frames are listed in *Table 3* and *Table 4*. By default all frames output from the VS6724 are EVEN. It is possible to set all frames to be ODD or to alternate between ODD and EVEN using the SyncCodeSetup register in the*Dither control* register bank.

Table 3. ITU656 embedded synchronization code definition (even frames)

Name	Description	4-byte sequence
SAV	Line start - active	FF 00 00 80
EAV	Line end - active	FF 00 00 9D
SAV (blanking)	Line start - blanking	FF 00 00 AB
EAV (blanking)	Line end - blanking	FF 00 00 B6

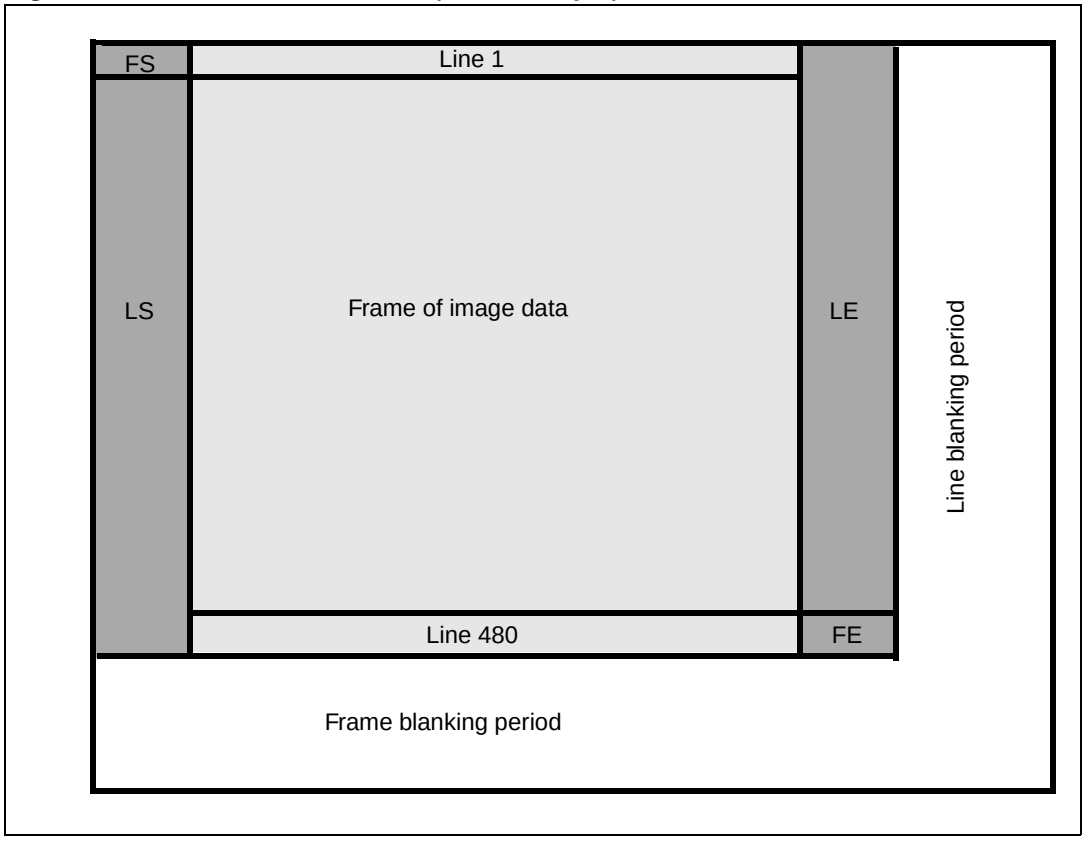
Table 4. ITU656 embedded synchronization code definition (odd frames)

Name	Description	4-byte sequence
SAV	Line start - active	FF 00 00 C7
EAV	Line end - active	FF 00 00 DA
SAV (blanking)	Line start - blanking	FF 00 00 EC
EAV (blanking)	Line end - blanking	FF 00 00 F1

2.10.3 Embedded synchronization EXT-CSI

The structure of a CSI image frame is shown *Figure 19*.

Figure 19. CSI 2 frame structure (VGA example)



For CSI, the synchronization codes are as listed in *Table 5*.

Table 5. CSI - embedded synchronization code definition

Name	Description	4-byte sequence
LS	Line start	FF 00 00 00
LE	Line end	FF 00 00 01
FS	Frame Start	FF 00 00 02
FE	Frame End	FF 00 00 03

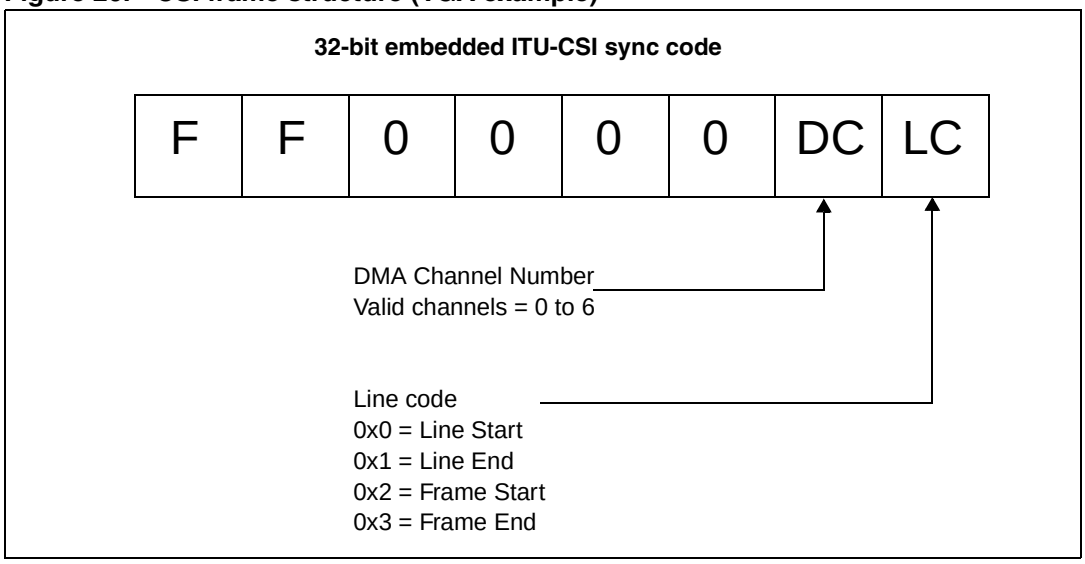
CSI logical DMA channels

The purpose of logical channels is to separate different data flows which are interleaved in the data stream, in the case of the VS6724 this allows the identification of the pipe context used for an image frame. The DMA channel identifier number is directly encoded in the 4-byte CSI embedded sync codes. The receiver can then monitor the DMA channel identifier and de-multiplex the interleaved video streams to their appropriate DMA channel. The bChannelID register can have the value 0 to 6. The DMA channel identifier must be fully programmable to allow the host to configure which DMA channels the different video data stream use.

- Logical channel control

The channel identifier is a part of CSI synchronization code, upper four bits of last byte of synchronization code. *Figure 20. CSI frame structure (VGA example)* illustrates the synchronization code with logical channel identifiers.

Figure 20. CSI frame structure (VGA example)



2.10.4 VSYNC and HSYNC

The VS6724 can provide two programmable hardware synchronization signals: VSYNC and HSYNC. The position of these signals within the output frame can be programmed by the user or an automatic setting can be used where the signals track the active video portion of the output frame regardless of its size.

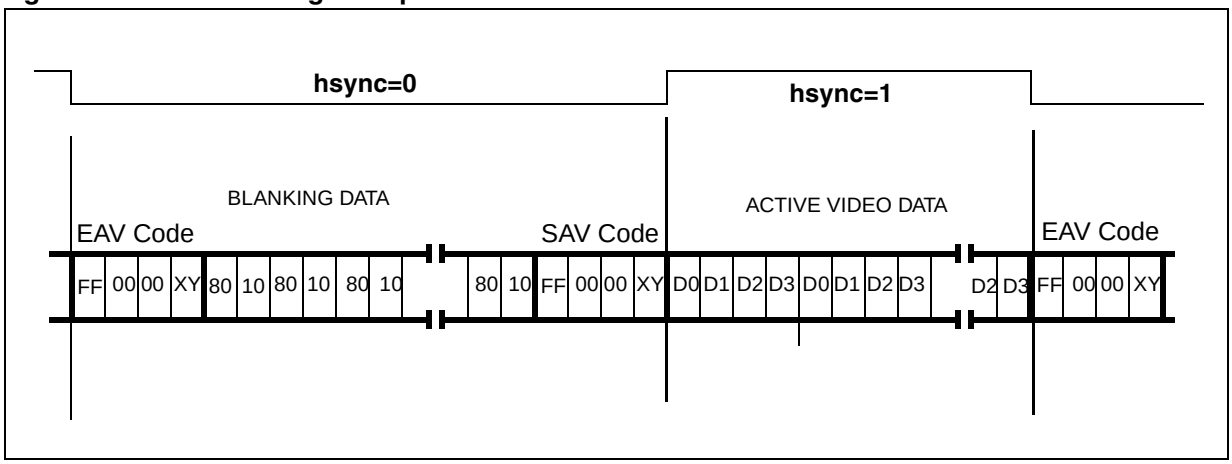
Horizontal synchronization signal (HSYNC)

The HSYNC signal is controlled by the bHSyncSetup register. The following options are available:

- enable/disable
- select polarity
- all lines or active lines only
- manual or automatic

In automatic mode the HSYNC signal envelops all the active video data on every line in the output frame regardless of the programmed image size. Line codes (if selected) fall outside the HSYNC envelope as shown in *Figure 21*.

Figure 21. HSYNC timing example



If manual mode is selected then the pixel positions for HSYNC rising edge and falling edge are programmable. The pixel position for the rising edge of HSYNC is programmed in the bHSyncRising registers. The pixel position for the falling edge of HSYNC is programmed in the bHSyncFalling registers.

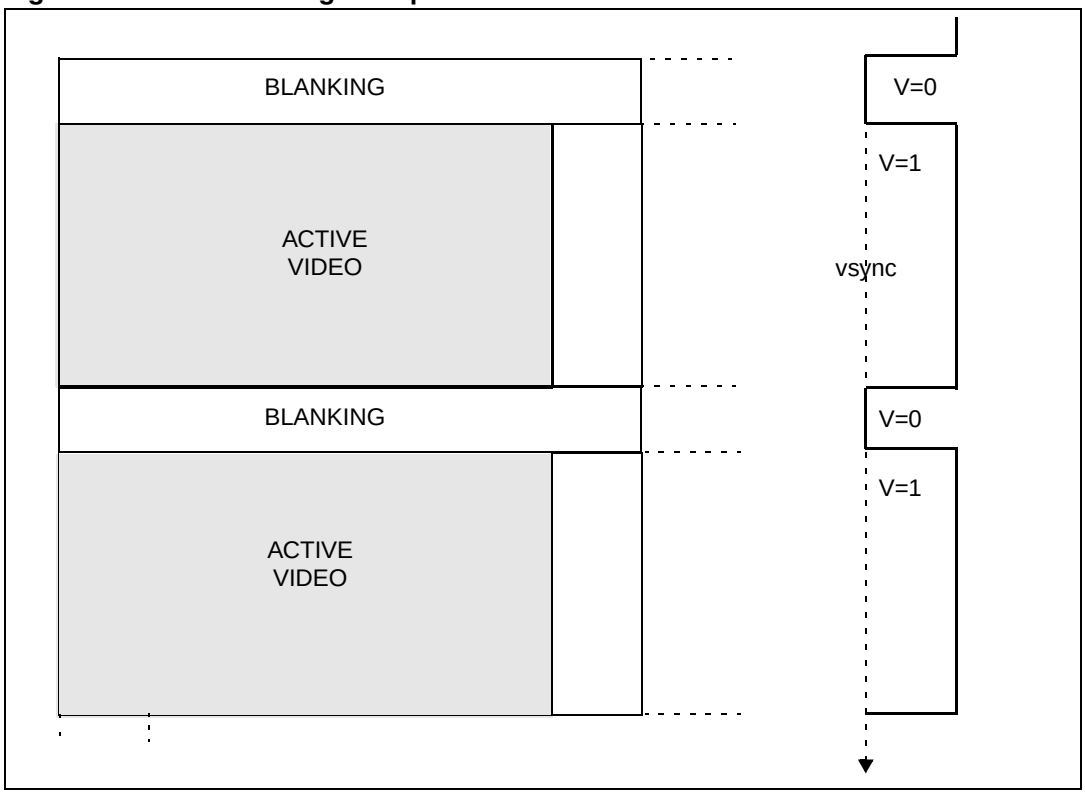
Vertical synchronization (VSYNC)

The VSYNC signal is controlled by the bSyncSetup register. The following options are available:

- enable/disable
- select polarity
- manual or automatic

In automatic mode the VSYNC signal envelops all the active video lines in the output frame regardless of the programmed image size as shown in *Figure 22*.

Figure 22. VSYNC timing example



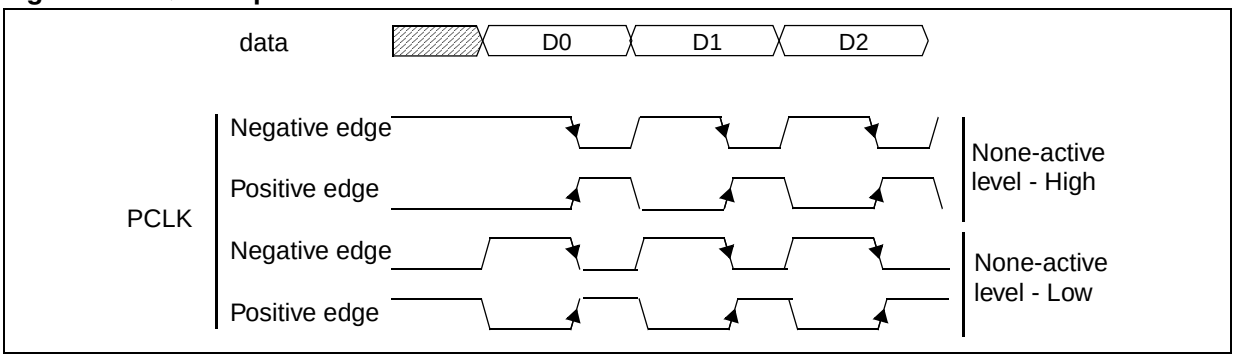
If manual mode is selected then the line number for VSYNC rising edge and falling edge is programmable. The rising edge of VSYNC is programmed in the bVsyncRisingLine registers, the pixel position for VSYNC rising edge is programmed in the bVsyncRisingPixel registers. Similarly the line count for the falling edge position is specified in the bVsyncFallingLine registers, and the pixel count is specified in the bVsyncFallingPixel registers.

2.10.5 Pixel clock (PCLK)

The PCLK signal is controlled by the *Dither control* register. The following options are available:

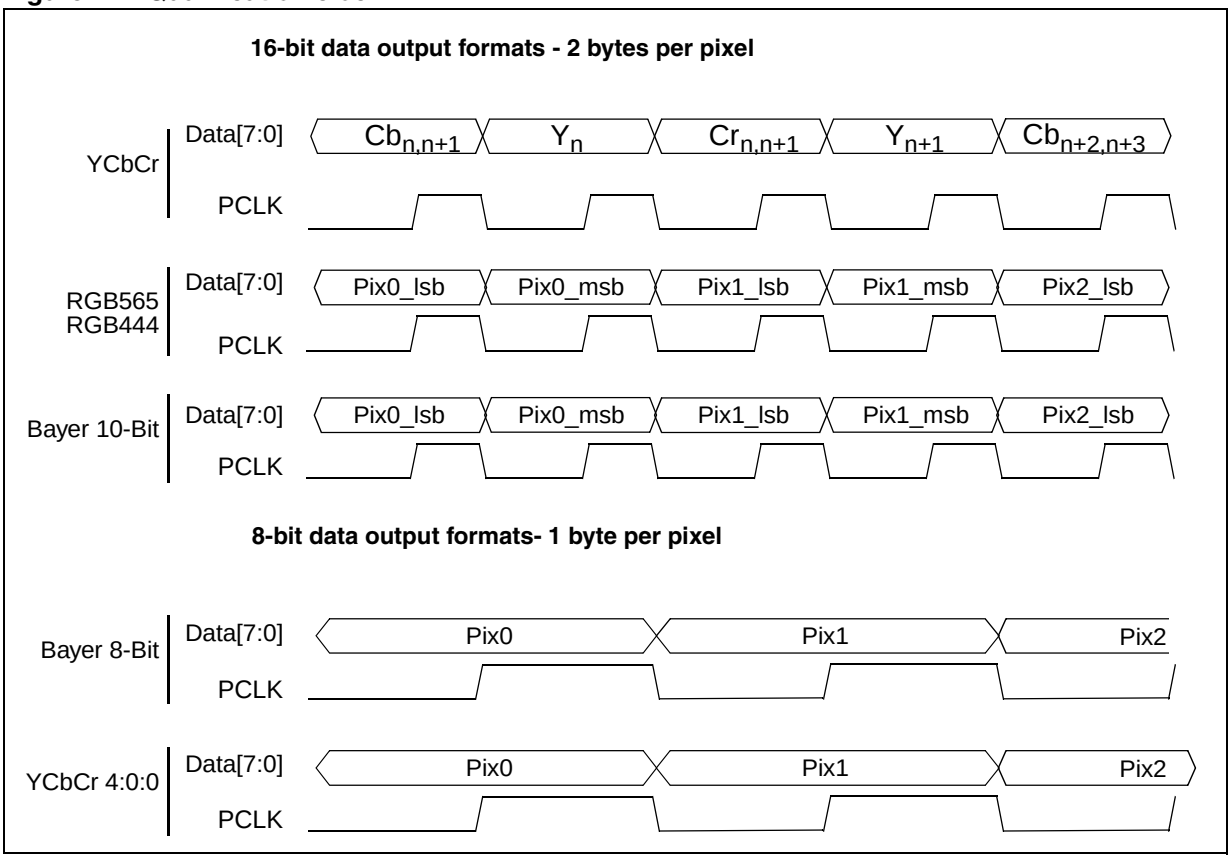
- enable/disable
- select polarity
- select starting phase
- qualify/don't qualify embedded synchronization codes
- enable/disable during horizontal blanking

Figure 23. QCLK options



The YCbCr, RGB and Bayer timings are represented on *Figure 24*, with the associated qualifying PCLK clock. The output clock rate is effectively halved for the Bayer 8-bit and YCbCr4:0:0 modes where only one byte of output data is required per pixel.

Figure 24. Qualification clock



In practice the user is likely to require to write some additional setup information prior to receive the required data output.

2.11 Timing control

2.11.1 Input clock

The VS6724 requires provision of an external reference clock. The external clock should be a DC coupled square wave and may have been RC filtered. The clock input is fail-safe in power down mode.

The VS6724 contains an internal PLL allowing it to produce accurate frame rates from a wide range of input clock frequencies. The allowable input range is from 6.5 MHz to 27 MHz. The external clock frequency must be programmed in the registers found in *Table 19: Video timing parameter host inputs*.

2.11.2 PLL operation

The VS6724 contains a firmware based programmable timing generator which automatically configures the internal video timing, PLL multipliers, clock dividers to achieve a target operation. The timing generator is controlled and constrained by the required PLL frequency, framerate and scaling factor, this in turn effects the output PCLK frequency.

Timing control

The **bSysClkMode** register selects the mode in which the system clock manager works,

- <0> SYSCLK_MODE_NORMAL
- <1> SYSCLK_MODE_USER

In normal mode the VS6724 can achieve UXGA at 30 fps and is based on an internal PLL frequency of 260 MHz, the clock management system ensures a maximum output frequency of 80 Mhz (YCbCr).

In user mode the VS6724 can be configured to allow the VS6724 to meet the input frequency constraints of a host application. The fpUserPLLCLK register sets the maximum frequency generated by the system clock manager and therefore limits the output frequency and is based on the following equation:

$$\textbf{fpUserPLLClk} = 2 * \textbf{PCLK}_{max}$$

The PCLKmax may be a limitation on the host and using the following approximate calculation can help work out the framerate achievable with this output PCLK rate;

$$\textbf{Framerate} = \textbf{PCLK}_{max} / (\textbf{Image size} * \textbf{data format} * 1.10)$$

- Image size = 1600 x 1200 for UXGA sensor mode
- Image size = 800 x 600 for SVGA sensor mode
- Data format = 2 for YCbCr 4:2:2, RGB and Bayer10 (as these are 2 Bytes per pixel)
- Data format = 1 for YCbCr 4:2:0 and Bayer 8 (as these are 1 Byte per pixel)
- (1.10) is a 10% increase to take into account interline and interframe.

2.11.3 Derating

When the scaler is operating the clock manager employs derating to reduce the peak output rate of the device by spreading the data over the full frame period.

The derating employed can be found by dividing the input size to the scaler by the output size of the scaler and rounding up to the nearest even number.

Example: VGA output scaled from the full UXGA array = 1600x1200 / 640x480 = 6.25 rounded up to the nearest even number = 8, therefore the VGA output is 8 times slower than that for the full UXGA output.

3 Host communication - I²C control interface

The interface used on the VS6724 is a subset of the I²C standard. Higher level protocol adaptations have been made to allow for greater addressing flexibility. This extended interface is known as the V2W interface.

3.1 Protocol

A message contains two or more bytes of data preceded by a START (S) condition and followed by either a STOP (P) or a repeated START (Sr) condition followed by another message.

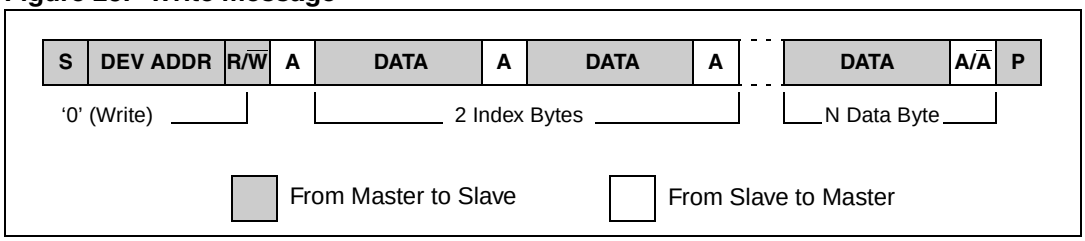
STOP and START conditions can only be generated by a V2W master.

After every byte transferred the receiving device must output an acknowledge bit which tells the transmitter if the data byte has been successfully received or not.

The first byte of the message is called the device address byte and contains the 7-bit address of the V2W slave to be addressed plus a read/write bit which defines the direction of the data flow between the master and the slave.

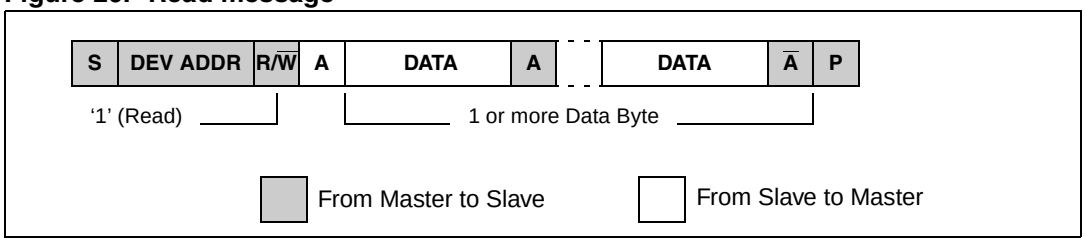
The meaning of the data bytes that follow device address changes depending whether the master is writing to or reading from the slave.

Figure 25. Write message



For the master writing to the slave the device address byte is followed by 2 bytes which specify the 16-bit internal location (index) for the data write. The next byte of data contains the value to be written to that register index. If multiple data bytes are written then the internal register index is automatically incremented after each byte of data transferred. The master can send data bytes continuously to the slave until the slave fails to provide an acknowledge or the master terminates the write communication with a STOP condition or sends a repeated START (Sr).

Figure 26. Read message



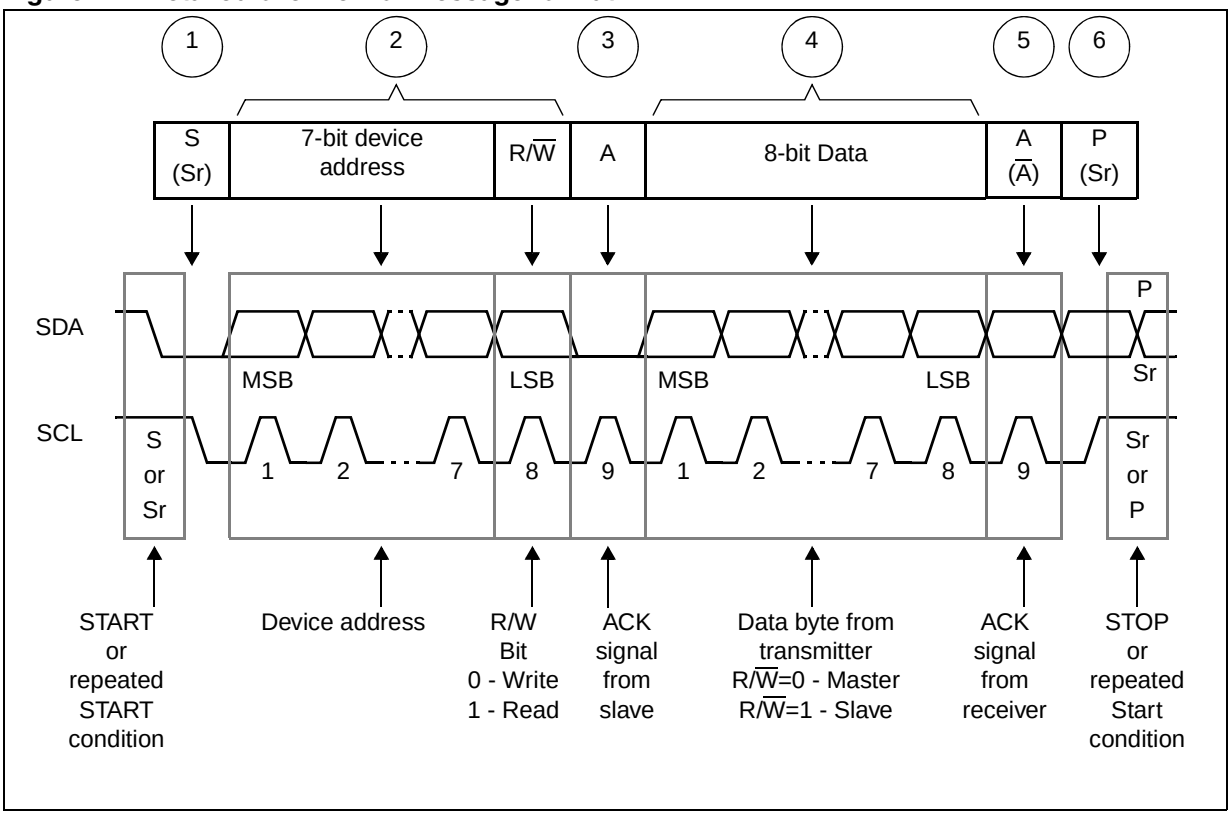
For the master reading from the slave the device address is followed by the contents of last register index that the previous read or write message accessed. If multiple data bytes are read then the internal register index is automatically incremented after each byte of data

read. A read message is terminated by the bus master generating a negative acknowledge after reading a final byte of data.

A message can only be terminated by the bus master, either by issuing a stop condition, a repeated start condition or by a negative acknowledge after reading a complete byte during a read operation.

3.2 Detailed overview of the message format

Figure 27. Detailed overview of message format



The V2W generic message format consists of the following sequence:

1. Master generates a START condition to signal the start of new message.
2. Master outputs, MS bit first, a 7-bit device address of the slave the master is trying to communicate with followed by a R/$\overline{W}$ bit.
   a) R/$\overline{W}$ = 0 then the master (transmitter) is writing to the slave (receiver).
   b) R/$\overline{W}$ = 1 the master (receiver) is reading from the slave (transmitter).
3. The addressed slave acknowledges the device address.
4. Data transmitted on the bus
   a) When a write is performed then master outputs 8-bits of data on SDA (MS Bit first).
   b) When a read is performed then slave outputs 8-bits of data on SDA (MS Bit First).
5. Data receive acknowledge
   a) When a write is performed slave acknowledges data.

b) When a read is performed master acknowledges data.

Repeat 4 and 5 until all the required data has been written or read.

Minimum number of data bytes for a read =1 (Shortest Message length is 2-bytes).

The master outputs a negative acknowledge for the data when reading the last byte of data. This causes the slave to stop the output of data and allows the master to generate a STOP condition.

6. Master generates a STOP condition or a repeated START.

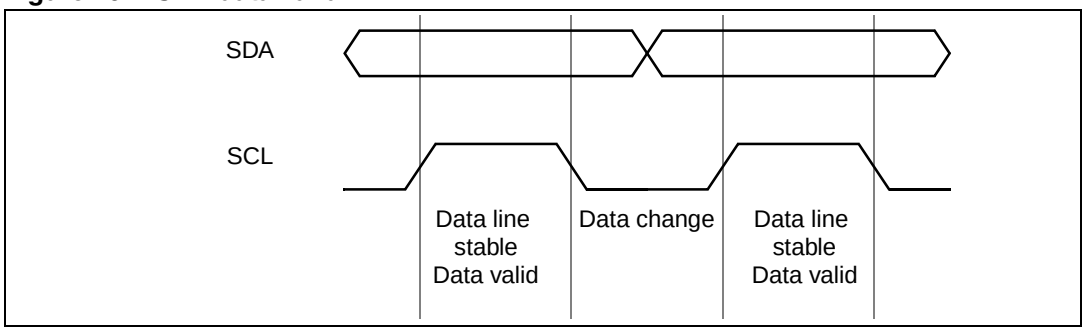
Figure 28. Device addresses

Sensor address	0	0	1	0	0	0	0	R/$\overline{W}$
Sensor write address 20_H	0	0	1	0	0	0	0	0
Sensor read address 21_H	0	0	1	0	0	0	0	1

3.2.1 Data valid

The data on SDA is stable during the high period of SCL. The state of SDA is changed during the low phase of SCL. The only exceptions to this are the start (S) and stop (P) conditions as defined below. (See *I²C slave interface* for full timing specification).

Figure 29. SDA data valid



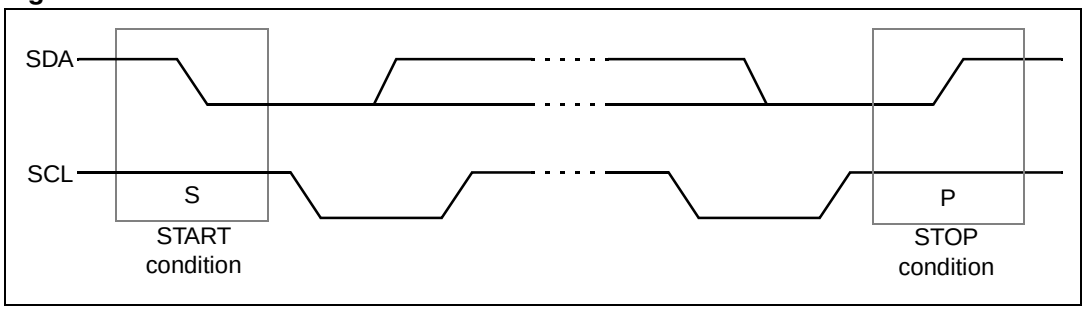
3.2.2 Start (S) and Stop (P) conditions

A START (S) condition defines the start of a V2W message. It consists of a high to low transition on SDA while SCL is high.

A STOP (P) condition defines the end of a V2W message. It consists of a low to high transition on SDA while SCL is high.

After STOP condition the bus is considered free for use by other devices. If a repeated START (Sr) is used instead of a stop then the bus stays busy. A START (S) and a repeated START (Sr) are considered to be functionally equivalent.

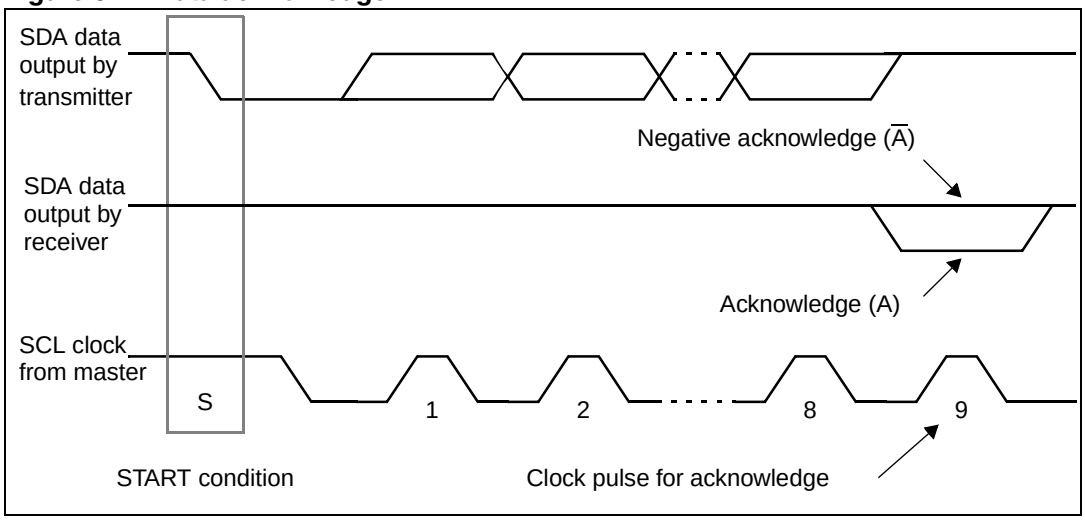
Figure 30. START and STOP conditions



3.2.3 Acknowledge

After every byte transferred the receiver must output an acknowledge bit. To acknowledge the data byte receiver pulls SDA during the 9th SCL clock cycle generated by the master. If SDA is not pulled low then the transmitter stops the output of data and releases control of the bus back to the master so that it can either generate a STOP or a repeated START condition.

Figure 31. Data acknowledge

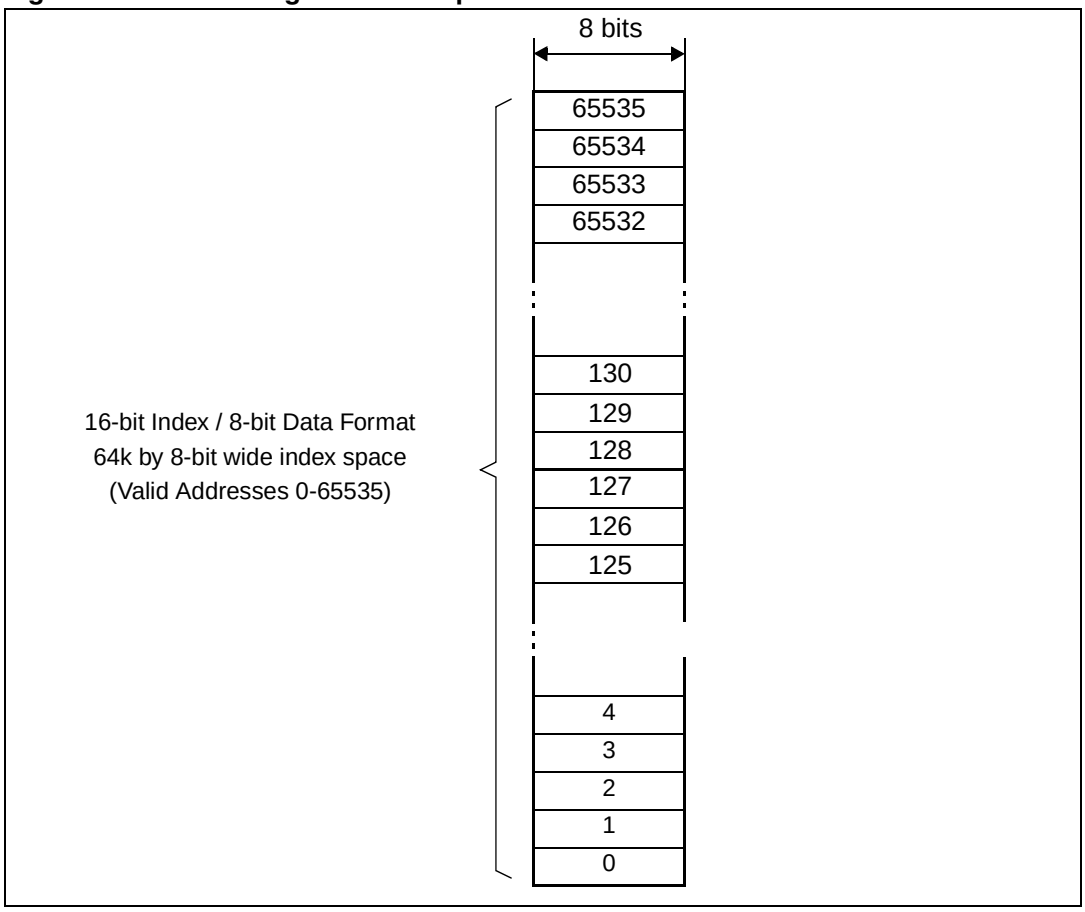


3.2.4 Index space

Communication using the serial bus centres around a number of registers internal to the either the sensor or the co-processor. These registers store sensor status, set-up, exposure and system information. Most of the registers are read/write allowing the receiving equipment to change their contents. Others (such as the chip id) are read only.

The internal register locations are organized in a 64k by 8-bit wide space. This space includes “real” registers, SRAM, ROM and/or micro controller values.

Figure 32. Internal register index space



3.3 Types of messages

This section gives guidelines on the basic operations to read data from and write data to VS6724.

The serial interface supports variable length messages. A message contains no data bytes or one data byte or many data bytes. This data can be written to or read from common or different locations within the sensor. The range of instructions available are detailed below.

- Single location, single byte data read or write.
- Write no data byte. Only sets the index for a subsequent read message.
- Multiple location, multiple data read or write for fast information transfers.

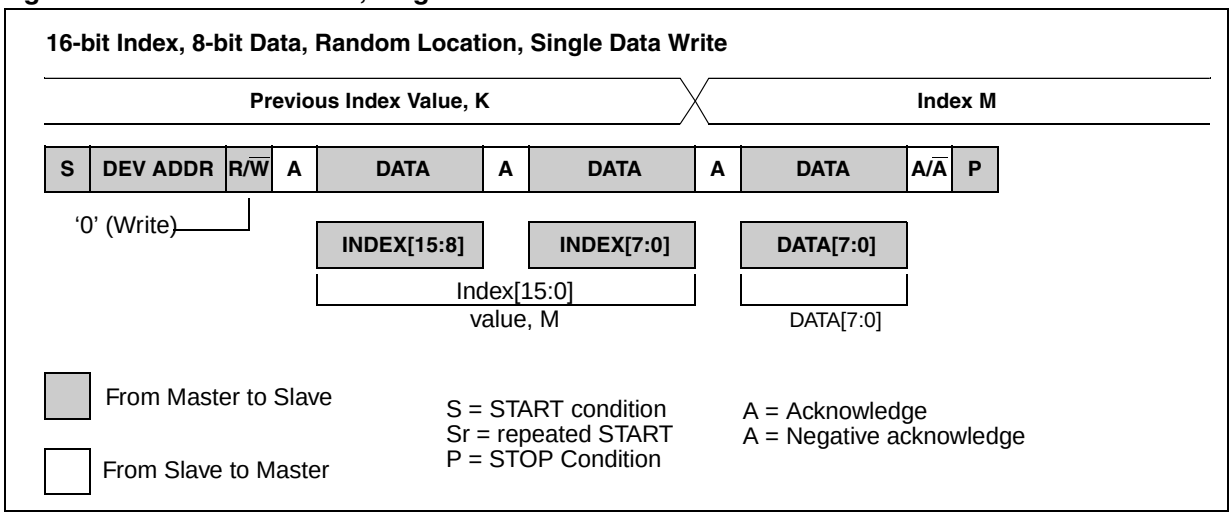
Any messages formats other than those specified in the following section should be considered illegal.

3.3.1 Random location, single data write

For the master writing to the slave the R/$\overline{W}$ bit is set to zero.

The register index value written is preserved and is used by a subsequent read. The write message is terminated with a stop condition from the master.

Figure 33. Random location, single write



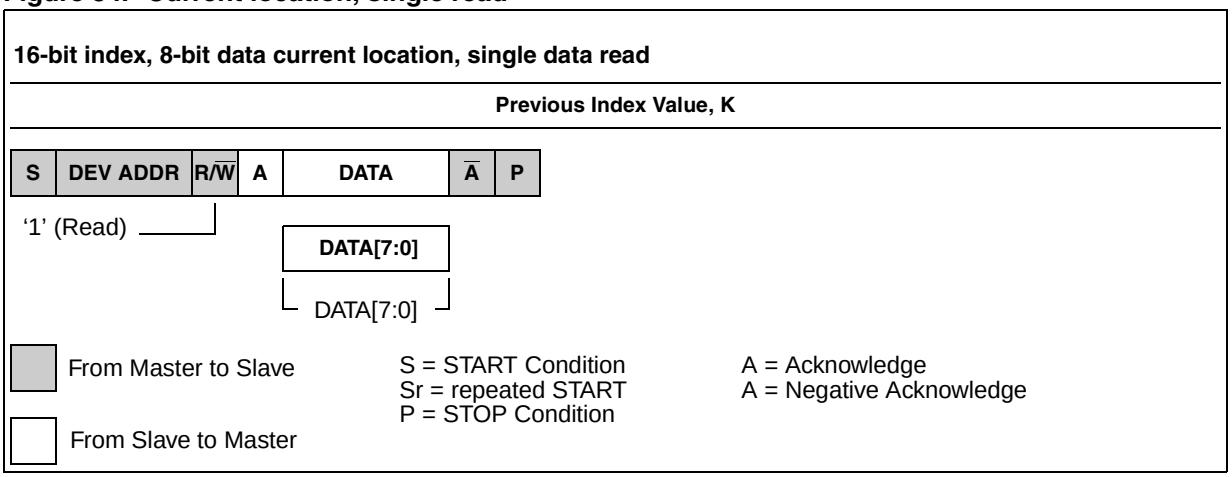
3.3.2 Current location, single data read

For the master reading from the slave the R/$\overline{W}$ bit is set to one. The register index of the data returned is that accessed by the previous read or write message.

The first data byte returned by a read message is the contents of the internal index value and NOT the index value. This was the case in older V2W implementations.

Note that the read message is terminated with a negative acknowledge ($\overline{A}$) from the master: it is not guaranteed that the master will be able to issue a stop condition at any other time during a read message. This is because if the data sent by the slave is all zeros, the SDA line cannot rise, which is part of the stop condition.

Figure 34. Current location, single read

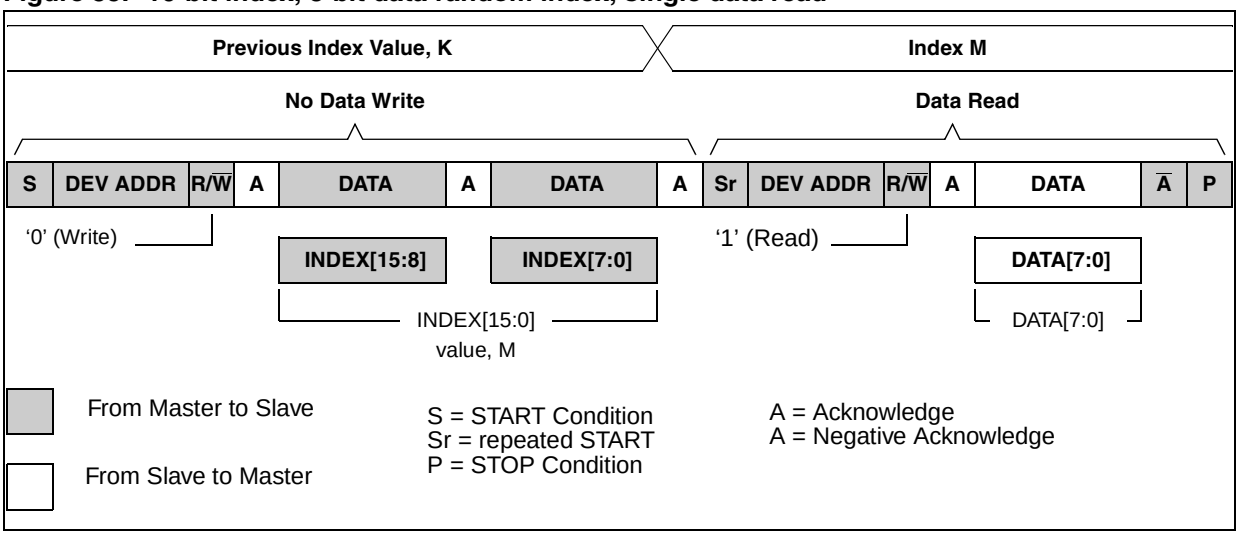


3.3.3 Random location, single data read

When a location is to be read, but the value of the stored index is not known, a write message with no data byte must be written first, specifying the index. The read message then completes the message sequence. To avoid relinquishing the serial to bus to another master a repeated start condition is asserted between the write and read messages.

As mentioned in the previous example, the read message is terminated with a negative acknowledge ($\overline{A}$) from the master.

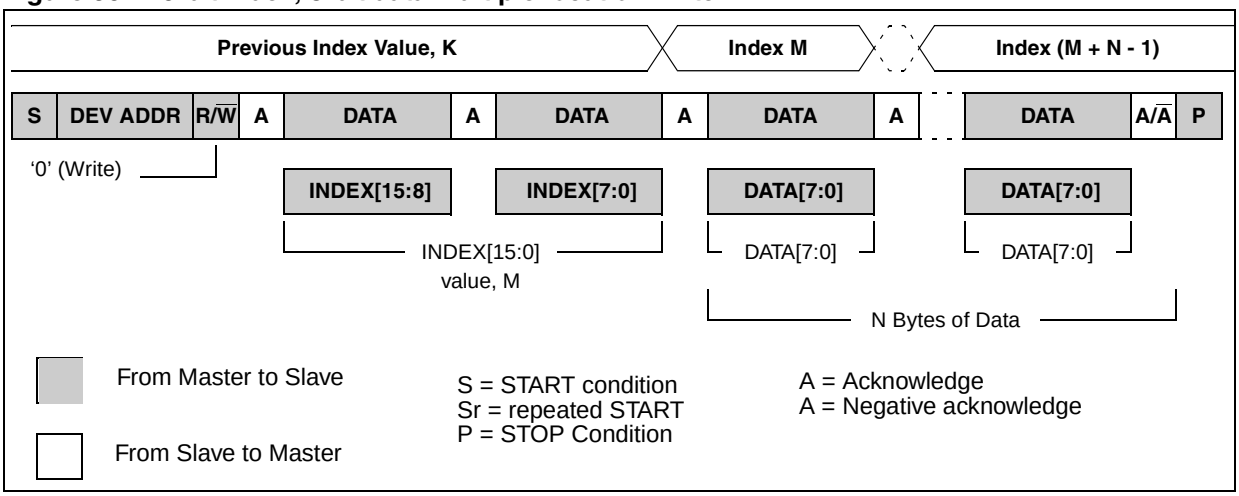
Figure 35. 16-bit index, 8-bit data random index, single data read



3.3.4 Multiple location write

For messages with more than 1 data byte the internal register index is automatically incremented for each byte of data output, making it possible to write data bytes to consecutive adjacent internal registers without having to send explicit indexes prior to sending each data byte.

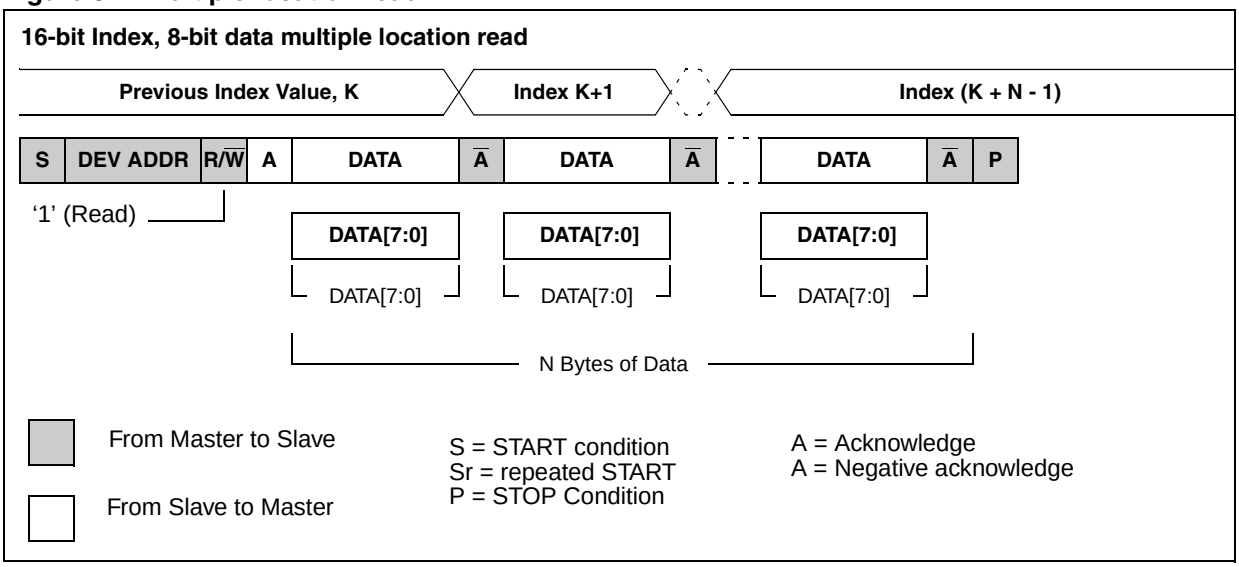
Figure 36. 16-bit index, 8-bit data multiple location write



3.3.5 Multiple location read stating from the current location

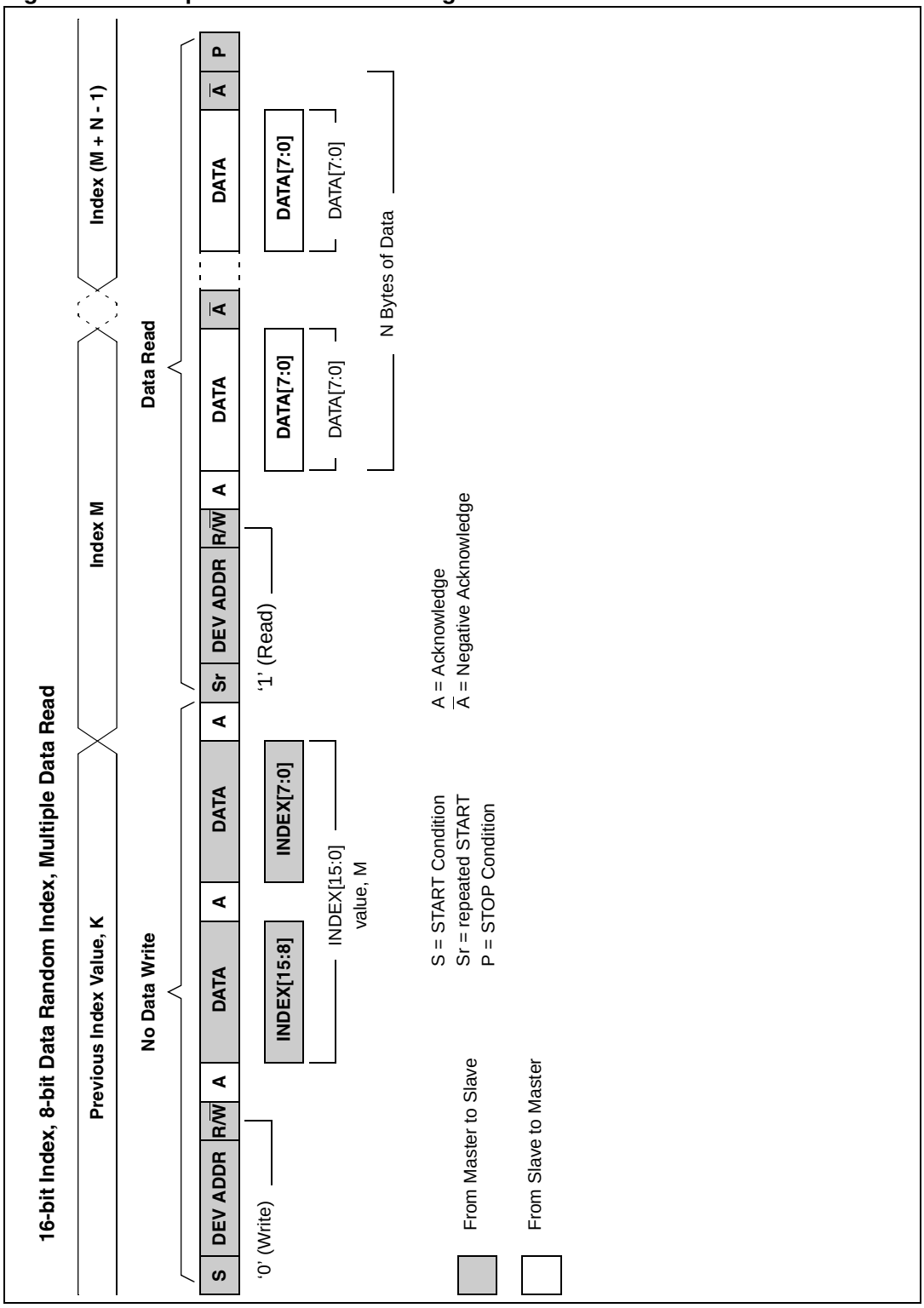
In the same manner to multiple location writes, multiple locations can be read with a single read message.

Figure 37. Multiple location read



3.3.6 Multiple location read starting from a random location

Figure 38. Multiple location read starting from a random location



4 Programming model and register description

4.1 Programming model

The VS6724 addressable register/memory space is configured as shown below. It consists of four principal areas each of which provides a different function:

User Interface: this area starting from address zero provides the user interface. Register reads and writes to this block are passed through the internal micro and processed. All operational control of the device by the host should be performed through these user interface locations.Many of these registers are detailed in *Section 4.2: Register description*.

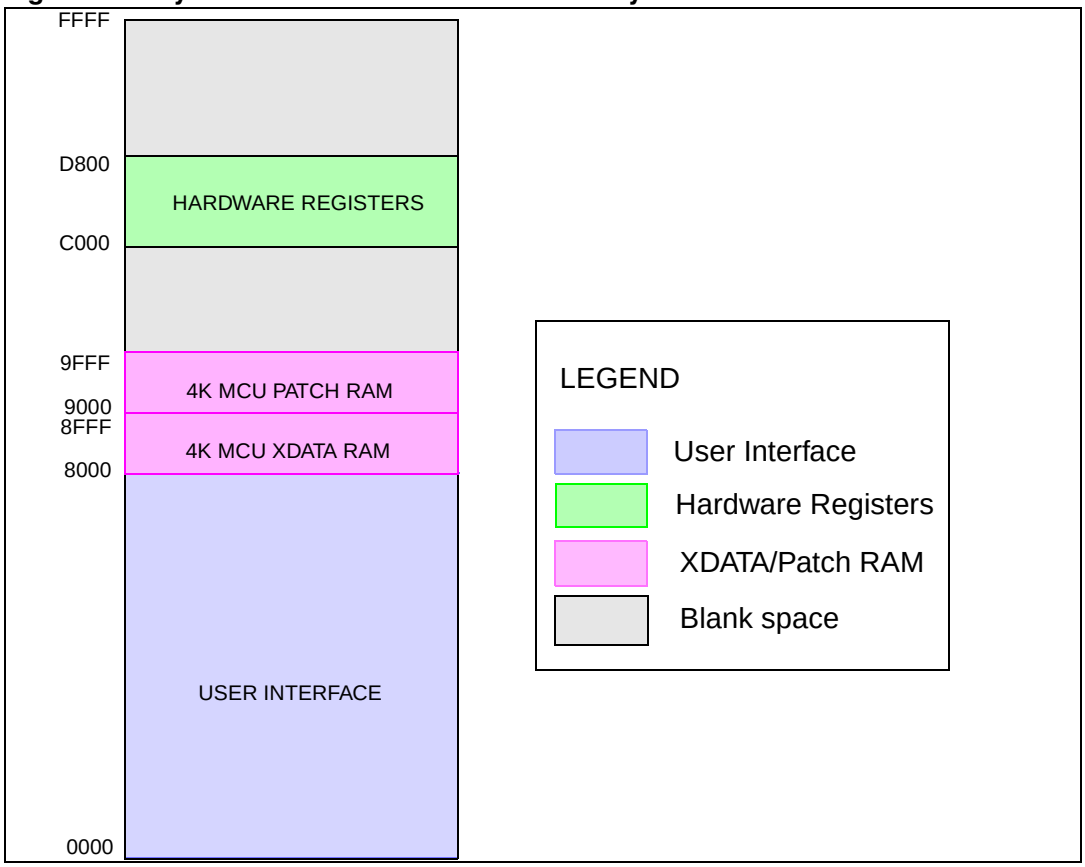
Registers not listed in this datasheet should be considered as reserved or read-only and should NOT be written to, as this may cause unpredictable results.

Hardware registers: provides direct access to hardware registers associated with each functional block of the device. In normal operation, these registers will be accessed under the control of the micro. i.e. they should never be accessed by the host system. However, they may be directly accessed by the host for debug and test purposes.

XDATA RAM: provides temporary variable storage for the on-board micro.

Patch RAM: provides memory space to download firmware patches and modify device operation. This provides a software update mechanism without the need for a new program ROM.

Figure 39. System / Host view of VS6724 memory



4.2 Register description

4.2.1 Register interpreter

Register contents represent different data types as described in *Register naming prefix*.

Table 6. Register naming prefix

Prefix	Description
VP_BYTE = ub	One Byte unsigned data
VP_UINT16 = uw	Two Byte unsigned data
Flag_e(F) = f	One byte data. Only two possible values
Enum = e	One Byte data. Specific to module
VP_FLOAT = fl	Two byte data. Expect value in Floating Point 16 notation
VP_INT8 = sb	One Byte signed data

Registers not listed in this datasheet should be considered as reserved or read-only should NOT be written to, as this may cause unpredictable results.

The VS6724 I²C write address is 0x20.

All I²C locations contain an 8-bit byte. However, certain parameters require 16 bits to represent them and are therefore stored in more than 1 location.

Note: For all 16 bit parameters the MSB register must be written before the LSB register.

The data stored in each location can be interpreted in different ways as shown below. Register contents represent different data types as described in *Table 7*.

Table 7. Data type

Data type	Description
BYTE	Single field register 8 bit parameter
UINT_16	Multiple field registers - 16 bit parameter
FLAG_e	Bit 0 of register must be set/cleared
CODED	Coded register - function depends on value written
FLOAT	Float Value

Float number format

Float 900 is used in ST co-processors to represent floating point numbers in 2 bytes of data. It conforms to the following structure:

Bit[15] = Sign bit (1 represents negative)

Bit[14:9] = 6 bits of exponent, biased at decimal 31

Bit[8:0] = 9 bits of mantissa

4.2.2 Low level control registers

Table 8. Low-level control registers

Index	LowLevelControlRegisters[1]	
0xC003	**MicroEnable**	
	Default value	0x1c
	Purpose	Used to power up the device
	Type	CODED
	Possible values	<0x1c> initial state after low to high transition of CE pin <0x02> Power enable for all MCU Clock- start device
0xC044	**DIO_Enable**	
	Default value	0x00
	Purpose	Enables the digital I/O of the device
	Type	CODED
	Possible values	<0> IO pins in a high impedance state ‘Tri-state’ <1> IO pins enabled

1. Can be controlled in all stable states.

Note: The default values for the above registers are true when the device is powered on, Ext. Clk input is present and the CE pin is high. All other registers can be read when the MicroEnable register is set to 0x02.

4.2.3 User interface register description

Device parameters [read only]

Table 9. Device parameters [read only]

Index	DeviceParameters [read only](1)	
0x0001 (MSByte) 0x0002 (LSByte)	**uwDeviceId**	
	Purpose	device id i.e. 724
	Type	UINT
0x0004	**bFirmwareVsnMajor**	
	Type	BYTE, Silicon cut 1.2 = 0,
0x0006	**bFirmwareVsnMinor**	
	Type	BYTE, Silicon cut 1.2 = 8,
0x0008	**bPatchVsnMajor**	
	Type	BYTE, Silicon cut 1.2 = 0,
0x000a	**bPatchVsnMinor**	
	Type	BYTE, Silicon cut 1.2 = 0,

1. Can be accessed in all stable state.

Host interface manager control

Table 10. Device parameters [read only]

Index	HostInterfaceManagerControl(1)	
0x0180	**bUserCommand**	
	Default value	<0> UNINITIALISED
	Purpose	User level control of operating states
	Type	CODED
	Possible values	<0> UNINITIALISED - powerup default <1> BOOT - the boot command will identify the sensor & setup low level handlers <2> RUN - stream video <3> PAUSE- stop video streaming <4> STOP - low power mode, analogue powered down <3> SNAPSHOT- grab one frame at correct exposure without flashgun <6> FLASHGUN - grab one frame at correct exposure for flashgun

1. Can be controlled in all stable states

Host interface manager status

Table 11. Host interface manager status [read only]

Index	HostInterfaceManagerStatus [read only](1)	
0x0202	**bState**	
	Default Value	<16>_RAW
	Purpose	The current state of the mode manager.
	Type	CODED
	Possible values	<16>_RAW - default powerup state. <33> WAITING_FOR_BOOT - Waiting for ModeManager to signal BOOT event. <34> PAUSED - Booted, the input pipe is idle. <38>WAITING_FOR_RUN - Waiting for ModeManager to complete RUN setup. <49> RUNNING - The pipe is active. <50> WAITING_FOR_PAUSE - The host has issued a PAUSE command. The HostInterfaceManager is waiting for the ModeManager to signal PAUSE processing complete. <64> FLASHGUN - Grabbing a single frame. <80> STOPPED - Low power
0x0204	**bCycles**	
	Purpose	Allows the host to read back a register which indicates that the device is streaming, the output should increment by one on each frame boundary
	Type	Coded
	Possible values	0 -255

1. Can be accessed in all stable states

Run mode control

Table 12. Run mode control

Index	RunModeControl(1)	
0x0280	**fMeteringOn**	
	Default Value:	<1> TRUE
	Purpose	If metering is off the Auto Exposure (AE) and Auto White Balance (AWB) tasks are disabled
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

1. Can be controlled in all stable states

Mode setup

Table 13. Mode setup

Index	ModeSetup	
0x0302	**bNonViewLive_ActivePipeContext (Can be controlled in all stable states)**	
	Default Value:	<0> PipeContext_0
	Purpose	Select the context used for streaming, can be changed on the fly to switch between contexts on next frame boundary
	Type	CODED
	Possible values	<0> PipeContext_0 <1>PipeContext_1
0x0304	**bSnapShot_ActivePipeContext (Can be controlled in all stable states)**	
	Default Value:	<0> PipeContext_0
	Purpose	Select the context for snapshot mode
	Type	CODED
	Possible values	<0> PipeContext_0 <1>PipeContext_1
0x0308	**SensorMode (Must be configured in STOP mode)**	
	Default value	<0>SensorMode_UXGA
	Purpose	Select the different sensor mode
	Type	CODED
	Possible values	<0>SensorMode_UXGA <1>SensorMode_SVGA

PipeContext0

Table 14. PipeContext0

Index	PipeContext0[1]	
0x0380	**bImageSize0 #**	
	Default value	<0> ImageSize_UXGA
	Purpose	required output dimension.
	Type	CODED
	Possible values	<0> ImageSize_UXGA <1> ImageSize_SXGA <2> ImageSize_SVGA <3> ImageSize_VGA <4> ImageSize_CIF <5> ImageSize_QVGA <6> ImageSize_QCIF <7> ImageSize_QQVGA <8> ImageSize_QQCIF <9> ImageSize_Manual - to use ManualSubSample and ManualCrop controls select Manual mode.
0x0383(MSB) 0x0384(LSB)	**uwManualHSize0 #**	
	Default value	0x00
	Purpose	if ImageSize_Manual selected, input required manual H size
	Type	UINT16
0x0387(MSB) 0x0388(LSB)	**uwManualVSize0 #**	
	Default value	0x00
	Purpose	if ImageSize_Manual selected, input required manual V size
	Type	UINT16
0x038b(MSB) 0x038c(LSB)	**uwZoomStepHSize0**	
	Default value	0x01
	Purpose	Set the zoom H step
	Type	UINT16
0x038f(MSB) 0x0390(LSB)	**uwZoomStepVSize0**	
	Default value	0x01
	Purpose	Set the zoom V step
	Type	UINT16

Table 14. PipeContext0 (continued)

Index	PipeContext0[1]	
0x0392	**bZoomControl0**	
	Default value	<0> ZoomStop
	Purpose	control zoom in, zoom out and zoom stop
	Type	C
	Possible values	<0> ZoomStop <1> ZoomStart_In <2> ZoomStart_Out <3> ZoomStep_in <4> ZoomStep_out
0x0395(MSB) 0x0396LSB)	**uwPanSteplHSize0**	
	Default value	0x00
	Purpose	Set the pan H step
	Type	UINT16
0x0399(MSB) 0x039a(LSB)	**uwPanStepVSize0**	
	Default value	0x00
	Purpose	Set the PanV step
	Type	UINT16
0x039c	**bPanControl0**	
	Default value	<0> Pan_Disable
	Purpose	control pandisable, pan right, pan left, pan up, pan down
	Type	C
	Possible values	<0> Pan_Disable <1> Pan_Right <2> Pan_Left <3> Pan_Down <4> Pan_Up
0x039e	**bCropControl0**	
	Default value	<1> Crop_auto
	Purpose	Select cropping manual or auto
	Type	C
	Possible values	<0> Crop_manual <1> Crop_auto
0x03a1(MSB) 0x03a2(LSB)	**uwManualCropHorizontalStart0**	
	Default value	0x00
	Purpose	Set the cropping H start address
	Type	UINT16

Table 14. PipeContext0 (continued)

Index	PipeContext0[1]	
0x03a5(MSB) 0x03a6(LSB)	**uwManualCropHorizontalSize0**	
	Default value	0x00
	Purpose	Set the cropping H size
	Type	UINT16
0x03a9(MSB) 0x03aa(LSB)	**uwManualCropVerticalStart0**	
	Default value	0x00
	Purpose	Set the cropping Vstart address
	Type	UINT16
0x03ad(MSB) 0x03ae(LSB)	**uwManualCropVerticalSize0**	
	Default value	0x00
	Purpose	Set the cropping Vsize
	Type	UINT16
0x03b0	**bImageFormat0 #[2]**	
	Default value	<0> ImageFormat_YCbCr_JFIF
	Purpose	select required output image format.
	Type	CODED
	Possible values	<0> ImageFormat_YCbCr_JFIF <1> ImageFormat_YCbCr_Rec601 <2> ImageFormat_YCbCr_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <3> ImageFormat_YCbCr_400 <4> ImageFormat_RGB_565 <5> ImageFormat_RGB_565_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <6> ImageFormat_RGB_444 <7> ImageFormat_RGB_444_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <9> ImageFormat_Bayer10_ThroughVP <10> ImageFormat_Bayer8_ThroughVP-- to truncate Bayer data to 8 bits data <11> JPEG
0x03b2	**bBayerOutputAlignment0**	
	Default value	<4> BayerOutputAlignment_RightShifted
	Purpose	set Bayer output alignment
	Type	CODED
	Possible values	<4> BayerOutputAlignment_RightShifted <5> BayerOutputAlignment_LeftShifted

Table 14. PipeContext0 (continued)

Index	PipeContext0[1]	
0x03b4	**bContrast0**	
	Default value	0x87
	Purpose	contrast control for both YCbCr and RGB output.
	Type	BYTE
0x03b6	**bcolorSaturation0**	
	Default value	0x78
	Purpose	color saturation control for both YCbCr and RGB output.
	Type	BYTE
0x03b8	**bGamma0**	
	Default value	0x0f
	Purpose	gamma settings.
	Type	BYTE
	Possible values	0 to 31
0x03ba	**fHorizontalMirror0**	
	Default Value:	0x00
	Purpose	Horizontal image orientation flip
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x03bc	**fVerticalFlip0**	
	Default Value:	0x00
	Purpose	Vertical image orientation flip
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x03be	**bChannelID**	
	Default value	0x00
	Purpose	Logical DMA Channel Number
	Type	BYTE
	Possible values	0 to 6

Table 14. PipeContext0 (continued)

Index	PipeContext0[1]	
0x03c0	**bJpegSqueezeSettings**	
	Default value	0x00
	Purpose	Select JPEG compression control
	Type	BYTE
	Possible values	<0> User Squeeze mode - Sets the VC squeeze setting as user squeeze (e.g. quality based setting). Low, medium or high quality factor should be set to override these settings <1> Auto Squeeze mode - Sets the VC squeeze setting as auto squeeze (file size based setting). <2> Auto still capture mode - If the host is taking only one still, the engine calibrates and waits for 5-10 frames to work out compression settings to reach optimum squeeze for target file size.
0x03c3(MSB) 0x03c4(LSB)	**bJpegTargetFileSize**	
	Default value	0x02ee (750KB)
	Purpose	sets target file size KByte
	Type	UINT16
0x03c6	**bJpegImageQuality**	
	Default value	0x00
	Purpose	Sets target JPEG image quality setting for Auto Squeeze mode
	Type	BYTE
	Possible values	<0>High Quality <1> Medium Quality <2> Low Quality
0x03c8	**bJpegImageFormat**	
	Default value	0x00
	Purpose	Sets image format supplied to JPEG engine
	Type	BYTE
	Possible values	<0>YCbCr 422 <1> YCbCr 420
0x03cc	**bMinScalerFactor**	
	Default value	0x10
	Purpose	Sets the minimum scaling factor which can be achieved by the zoom operation, this can be used to limit the max PCLK rate.
	Type	BYTE
	Possible values	0x00 - 0xff

1. Can be controlled in all stable state.
   # denotes registers where changes will only be consumed during the transition to a RUN state.
2. It is possible to switch between any YCbCr (422) mode, RGB mode and Bayer 10bit or move between YCbCr 400 and a Bayer8 mode without a requiring a transition to STOP, it is not possible to move between these groups of modes without first a transition to STOP then a BOOT.

PipeContext 1

Table 15. PipeContext1

Index	PipeContext1[1]	
0x0400	**bImageSize1 #**	
	Default value	<1> ImageSize_UXGA
	Purpose	required output dimension.
	Type	CODED
	Possible values	<0> ImageSize_UXGA <1> ImageSize_SXGA <2> ImageSize_SVGA <3> ImageSize_VGA <4> ImageSize_CIF <5> ImageSize_QVGA <6> ImageSize_QCIF <7> ImageSize_QQVGA <8> ImageSize_QQCIF <9> ImageSize_Manual - to use ManualSubSample and ManualCrop controls select Manual mode.
0x0403(MSB) 0x0404(LSB)	**uwManualHSize1 #**	
	Default value	0x00
	Purpose	if ImageSize_Manual selected, input required manual H size
	Type	UINT16
0x0407(MSB) 0x0408(LSB)	**uwManualVSize1 #**	
	Default value	0x00
	Purpose	if ImageSize_Manual selected, input required manual V size
	Type	UINT16
0x040b(MSB) 0x040c(LSB)	**uwZoomStepHSize1**	
	Default value	0x01
	Purpose	Set the zoom H step
	Type	UINT16
0x040f(MSB) 0x0410(LSB)	**uwZoomStepVSize1**	
	Default value	0x01
	Purpose	Set the zoom V step
	Type	UINT16

Table 15. PipeContext1 (continued)

Index	PipeContext1[1]	
0x0412	**bZoomControl1**	
	Default value	<0> ZoomStop
	Purpose	control zoom in, zoom out, zoom stop
	Type	CODED
	Possible values	<0> ZoomStop <1> ZoomStart_In <2> ZoomStart_Out <3> ZoomStep_in <4> ZoomStep_out
0x0415(MSB) 0x0416(LSB)	**uwPanSteplHSize1**	
	Default value	0x00
	Purpose	Set the pan H step
	Type	UINT16
0x0419(MSB) 0x041a(LSB)	**uwPanStepVSize1**	
	Default value	0x00
	Purpose	Set the PanV step
	Type	UINT16
0x041c	**bPanControl1**	
	Default value	<0> Pan_Disable
	Purpose	control pandisable, pan right, pan left, pan up, pan down
	Type	C
	Possible values	<0> Pan_Disable <1> Pan_Right <2> Pan_Left <3> Pan_Down <4> Pan_Up
0x041e	**bCropControl1**	
	Default value	<1> Crop_auto
	Purpose	Select cropping manual or auto
	Type	C
	Possible values	<0> Crop_manual <1> Crop_auto
0x0421(MSB) 0x0422(LSB)	**uwManualCropHorizontalStart1**	
	Default value	0x00
	Purpose	Set the cropping H start address
	Type	UINT16

Table 15. PipeContext1 (continued)

Index	PipeContext1[1]	
0x0425(MSB) 0x0426(LSB)	**uwManualCropHorizontalSize1**	
	Default value	0x00
	Purpose	Set the cropping H size
	Type	UINT16
0x0429(MSB) 0x042a(LSB)	**uwManualCropVerticalStart1**	
	Default value	0x00
	Purpose	Set the cropping Vstart address
	Type	UINT16
0x042d(MSB) 0x042e(LSB)	**uwManualCropVerticalSize1**	
	Default value	0x00
	Purpose	Set the cropping Vsize
	Type	UINT16
0x0430	**bImageFormat1**[2]	
	Default value	<0> ImageFormat_YCbCr_JFIF
	Purpose	select required output image format.
	Type	CODED
	Possible values	<0> ImageFormat_YCbCr_JFIF <1> ImageFormat_YCbCr_Rec601 <2> ImageFormat_YCbCr_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <3> ImageFormat_YCbCr_400 <4> ImageFormat_RGB_565 <5> ImageFormat_RGB_565_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <6> ImageFormat_RGB_444 <7> ImageFormat_RGB_444_Custom - to use custom output select required RgbToYCbCr OutputSignalRange from 'PipeContext' page. <9> ImageFormat_Bayer10ThroughVP <10> ImageFormat_Bayer8ThroughVP-- to truncate Bayer data to 8 bits data <11> JPEG 4:2:2
0x0432	**bBayerOutputAlignment1**	
	Default value	<4> BayerOutputAlignment_RightShifted
	Purpose	set Bayer output alignment
	Type	CODED
	Possible values	<4> BayerOutputAlignment_RightShifted <5> BayerOutputAlignment_LeftShifted

Table 15. PipeContext1 (continued)

Index	PipeContext1[1]	
0x0434	**bContrast1**	
	Default value	0x87
	Purpose	contrast control for both YCbCr and RGB output.
	Type	BYTE
0x0436	**bcolorSaturation1**	
	Default value	0x78
	Purpose	color saturation control for both YCbCr and RGB output.
	Type	BYTE
0x0438	**bGamma1**	
	Default value	0x0f
	Purpose	gamma settings.
	Type	BYTE
	Possible values	0 to 31
0x043a	**fHorizontalMirror1**	
	Default value	0x00
	Purpose	Horizontal image orientation flip
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x043c	**fVerticalFlip1**	
	Default value	0x00
	Purpose	Vertical image orientation flip
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x043e	**bChannelD**	
	Default value	0x00
	Purpose	Logical DMA Channel Number
	Type	BYTE
	Possible values	0 to 6

Table 15. PipeContext1 (continued)

Index	PipeContext1[1]	
0x0440	**bJpegSqueezeSettings**	
	Default value	0x00
	Purpose	Select JPEG compression control
	Type	BYTE
	Possible values	<0> User Squeeze mode - Sets the VC squeeze setting as user squeeze (e.g. quality based setting). Low, medium or high quality factor should be set to override these settings <1> Auto Squeeze mode - Sets the VC squeeze setting as auto squeeze (file size based setting). <2> Auto still capture mode - If the host is taking only one still, the engine calibrates and waits for 5-10 frames to work out compression settings to reach optimum squeeze for target file size.
0x0443(MSB) 0x0444(LSB)	**bJpegTargetFileSize**	
	Default value	0x02ee (750KB)
	Purpose	sets target file size KByte
	Type	UINT16
0x0446	**bJpegImageQuality**	
	Default value	0x00
	Purpose	Sets target JPEG image quality setting for Auto Squeeze mode
	Type	BYTE
	Possible values	<0>High Quality <1> Medium Quality <2> Low Quality
0x0448	**bJpegImageFormat**	
	Default value	0x00
	Purpose	Sets image format supplied to JPEG engine
	Type	BYTE
	Possible values	<0>YCbCr 422 <1> YCbCr 420
0x044c	**bMinScalerFactor**	
	Default value	0x10
	Purpose	Sets the minimum scaling factor which can be achieved by the zoom operation, this can be used to limit the max PCLK rate.
	Type	BYTE
	Possible values	0x00 - 0xff

1. Can be controlled in all stable state. # denotes registers where changes will only be consumed during the transition to a RUN state.
2. It is possible to switch between any YCbCr (422) mode, RGB mode and Bayer 10bit or move between YCbCr 400 and a Bayer8 mode without a requiring a transition to STOP, it is not possible to move between these groups of modes without first a transition to STOP then a BOOT.

ViewLive control

Table 16. ViewLive control

Index	ViewLiveControl	
0x0480	**fEnable (Can be controlled in all stable states)**	
	Default value	<0> FALSE
	Purpose	set to enable the View Live mode.
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x0482	**bInitialPipeContext (must be setup in PAUSE or STOP mode)**	
	Default value	<0> PipeContext_0
	Purpose	First frame output will be from PipeSetupBank selected by 'bInitialPipeContext'. if ViewLive is enabled the next frame will be from the other PipeContext, otherwise only one PipeContext will be used.
	Type	CODED
	Possible values	<0> PipeContext_0 <1> PipeContext_1

ViewLive status [read only]

Table 17. ViewLive status

Index	ViewLiveStatus [read only]	
0x0500	**CurrentPipeContext**	
	Default value	<0>PipeContext_0
	Purpose	indicates the PipeContext which has most recently been applied to the pixel pipe hardware.
	Type	CODED
	Possible values	<0> PipeContext_0 <1> PipeContext_1

Power management

Table 18. Power management

Index	PowerManagement(1)	
0x0580	**bTimeToPowerdown**	
	Default value	0x0f
	Purpose	Time (mSecs) from entering Pause mode until the system automatically transitions stop mode. 0xff disables the automatic transition.
	Type	BYTE

1. Must be configured in STOP mode

Video timing parameter host inputs

Table 19. Video timing parameter host inputs

Index	VideoTimingParameterHostInputs(1)	
0x0605(MSByte) 0x0606(LSByte)	**uwExternalClockFrequencyNumerator**	
	Default value	0x0c
	Purpose	The External Clock Frequency configuration must match the frequency supplied to the CLK pin; External clock frequency = uwExternalClockFrequencyDenumerator/ bExternalClockFrequencyDenominator
	Type	UINT16
	Possible values	The External clock frequency range is 6 to 27, this represents 6MHz to 27 MHz
0x0608	**bExternalClockFrequencyDenominator**	
	Default value	0x01
	Type	BYTE

1. Should be configured in the RAW state

Video timing parameter FP inputs (read only)

Table 20. Video timing parameter FP inputs (read only)

Index	VideoTimingParameterFPInputs (read only)(1)	
0x0605(MSByte) 0x0606(LSByte)	**uwExternalClockFrequencyMhzNumerator**	
	Purpose	Allows the host to read back the **uwExternalClockFrequency**
	Type	FLOAT

1. Can be read in all operating modes

Video timing control

Table 21. Video timing control

Index	VideoTimingControl(1)	
0x0880	**bSysClkMode**	
	Default value	0x00
	Purpose	User selection between using default system clock frequency or selecting this frequency manually
	Type	CODED
	Possible values	<0>Normal - Default PLL frequency <1>user - Manual selection of PLL frequency
0x0883(MSByte) 0x0884(LSByte)	**bfpUserPLLClk**	
	Default value	0x00
	Purpose	User selectable system clock frequency
	Type	FLOAT - number of MHz
	Possible values	Minimum = 6 Maximum = 270 (Max required for 30fps)

1. Must be controlled in STOP state

Frame dimension parameter host inputs

Table 22. Frame dimension parameter host inputs

Index	FrameDimensionParameterHostInputs(1)	
0x0b80	**bLightingFrequencyHz**	
	Default value	0x64
	Purpose	AC Frequency - used for flicker free time period calculations this mains frequency determines the flicker free time period.
	Type	BYTE
0x0b82	**fFlickerCompatibleFrameLength**	
	Default value	<0> FALSE
	Purpose	flicker_compatible_frame_length
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

1. Can be controlled in all stable states

Static frame rate control

Table 23. Static frame rate control

Index	StaticFrameRateControl(1)	
0x0c81(MSByte) 0x0c82(LSByte)	**uwDesiredFrameRate_Num**	
	Default value	0x1e
	Purpose	Numerator for the Frame Rate
	Type	UINT16
0x0c84	**bDesiredFrameRate_Den**	
	Default value	0x01
	Purpose	Denominator for the Frame Rate
	Type	BYTE

1. Can be controlled in all stable states

Static frame rate status (read only)

Table 24. Static frame rate status

Index	StaticFrameRatestatus(1)	
0x0d01(MSByte) 0x0d02(LSByte)	**fprequestedFramerate_Hz**	
	Purpose	Shows the existing framerate of the system
	Type	FLOAT

1. Can be controlled in all stable states

Automatic frame rate control

Table 25. Automatic frame rate control

Index	AutomaticFrameRateControl[1]	
0x0d80	**bFrameRateMode**	
	Default value	0x00
	Purpose	Defines the mode in which the framerate of the system would work
	Type	
	Possible values	<0> Manual - the system uses the values programmed into the **StaticFrameRateControl** registers <1> Auto - the system automatically adjusts the framerate between the **bUserMinimumFrameRate_Hz** and **bUserMinimumFrameRate_Hz.**
0x0d82	**bImpliedGainThresholdLow_num**	
	Default value	0x02
	Purpose	Numerator for calculation of low threshold of autoframerate control
	Type	BYTE
0x0d84	**bImpliedGainThresholdLow_den**	
	Default value	0x02
	Purpose	Denominator for calculation of low threshold of autoframerate control.
	Type	BYTE
0x0d86	**bImpliedGainThresholdHigh_num**	
	Default value	0x02
	Purpose	Numerator for calculation of High threshold of autoframerate control
	Type	BYTE
0x0d88	**bImpliedGainThresholdHigh_Den**	
	Default value	0x02
	Purpose	Denominator for calculation of high threshold of autoframerate control
	Type	BYTE
0x0d8a	**bUserMinimumFrameRate_Hz**	
	Default value	0x02
	Purpose	Sets the minimum framerate employed when in automatic framerate mode.
	Type	BYTE

Table 25. Automatic frame rate control (continued)

Index	AutomaticFrameRateControl(1)	
0x0d8c	**bUserMinimumFrameRate_Hz**	
	Default value	0x1e
	Purpose	Sets the maximum framerate employed when in automatic framerate mode.
	Type	BYTE
0x0d8e	**bRelativeChange_num**	
	Default value	0x02
	Purpose	Numerator for calculation of relative change in framerate.
	Type	BYTE
0x0d90	**bRelativeChange_den**	
	Default value	0x02
	Purpose	Denominator for calculation of relative change in framerate
	Type	BYTE
0x0e01 0x0e02	**fpImpliedGain (read only)**	
	Purpose	Displays the present value of implied gain
	Type	FLOAT

1. Can be controlled in all stable states

Exposure controls

Table 26. Exposure controls

Index	ExposureControls(1)	
0x1080	**bMode**	
	Default value	<0> AUTOMATIC_MODE
	Purpose	Sets the mode for the exposure algorithm
	Type	CODED
	possible values	<0> AUTOMATIC_MODE - Automatic Mode of Exposure which includes computation of Relative Step <1> COMPILED_MANUAL_MODE - Compiled Manual Mode in which the desired exposure is given and not calculated by algorithm <2> DIRECT_MANUAL_MODE - Mode in which the exposure parameters are input directly and not calculated by compiler <3> FLASHGUN_MODE - Flash Gun Mode in which the exposure parameters are set to fixed values

Table 26. Exposure controls (continued)

Index	ExposureControls[1]	
0x1082	**bMetering**	
	Default value	<0> ExposureMetering_flat
	Purpose	Weights to be associated with the zones for calculating the mean statistics Exposure Weight could Centered, Backlit or Flat
	Type	C
	possible values	<0> ExposureMetering_flat - Uniform gain associated with all pixels <1> ExposureMetering_backlit - more gain associated with centre pixels and bottom pixels <2> ExposureMetering_centred - more gain associated with centre pixels
0x1084	**bManualExposureTime_Num**	
	Default value	0x01
	Purpose	Exposure Time for Compiled Manual Mode in seconds. Num/Den gives required exposure time
	Type	BYTE
0x1086	**bManualExposureTime_Den**	
	Default value	0x1e
	Type	BYTE
0x1089(MSByte) 0x108a(LSByte)	**fpManualFloatExposureTime**	
	Default value	0x59aa (15008)
	Purpose	Exposure Time for the Manual Mode. This value is in microseconds. **iMiscSettings** must be configured
	Type	FLOAT
0x108d(MSByte) 0x108e(LSByte)	**fpColdStartDesiredTime_us**	
	Default value	0x59aa (15008)
	Purpose	Exposure Time for the that the system will use when entering RUN mode
	Type	FLOAT
0x1090	**iExposureCompensation**	
	Default value	0x00
	Purpose	Exposure Compensation - a user choice for setting the runtime target. A unit of exposure compensation corresponds to 1/6 EV. Default value according to the Nominal Target of 30 is 0. Coded Value of Exposure compensation can take values from -25 to 12.
	Type	INT8

Table 26. Exposure controls (continued)

Index	ExposureControls[1]	
0x1092	**iMiscSettings**	
	Default value	0x00
	Purpose	Set Bit [6] to use **fpManualFloatExposureTime** value
	Type	INT8
	Possible Value	0x00 - COMPILED_MANUAL_MODE uses **bManualExposureTime_Num** and **bManualExposureTime_Den** 0x40 - COMPILED_MANUAL_MODE uses **fpManualFloatExposureTime**
0x1095 (MSByte) 0x1096(LSByte)	**uwDirectModeCoarseIntegrationLines**	
	Default value	0x00
	Purpose	Coarse Integration Lines to be set for Direct Mode
	Type	UINT16
0x1099(MSByte) 0x109a(LSByte)	**uwDirectModeFineIntegrationPixels**	
	Default value	0x00
	Purpose	Fine Integration Pixels to be set for Direct Mode
	Type	UINT16
0x109d (MSByte) 0x109e(LSByte)	**fpDirectModeAnalogGain**	
	Default value	0x00
	Purpose	Analog Gain to be set for Direct Mode
	Type	FLOAT
0x10a1 (MSByte) 0x10a2(LSByte)	**fpDirectModeDigitalGain**	
	Default value	0x00
	Purpose	Digital Gain to be set for Direct Mode
	Type	FLOAT
0x10a5 (MSByte) 0x10a6(LSByte)	**uwFlashGunModeCoarseIntLines**	
	Default value	0x00
	Purpose	Coarse Integration Lines to be set for Flash Gun Mode
	Type	UINT16
0x10a9 (MSByte) 0x10aa(LSByte)	**uwFlashGunModeFineIntPixels**	
	Default value	0x00
	Purpose	Fine Integration Pixels to be set for Flash Gun Mode
	Type	UINT16

Table 26. Exposure controls (continued)

Index	ExposureControls[1]	
0x10ad (MSByte) 0x10ae(LSByte)	**fpFlashGunModeAnalogGain**	
	Default value	0x00
	Purpose	Analog Gain to be set for Flash Gun Mode
	Type	FLOAT
0x10b1 (MSByte) 0x10b2(LSByte)	**fpFlashGunModeDigitalGain**	
	Default value	0x00
	Purpose	Digital Gain to be set for Flash Gun Mode
	Type	FLOAT
0x10b4	**fFreezeAutoExposure**	
	Default value	<0> FALSE
	Purpose	Freeze auto exposure
	Type	Flag_e
	possible values	<0> FALSE <1> TRUE
0x10b7 (MSByte) 0x10b8(LSByte)	**fpUserMaximumIntegrationTime**	
	Default value	0x647f (654336)
	Purpose	User Maximum Integration Time in microseconds. This control takes in the maximum integration time that host would like to support. This would in turn give an idea of the degree of “wobbly pencil effect” acceptable to Host.
	Type	FLOAT
0x10bb(MSByte) 0x10bc(LSByte)	**fpRecommendFlashGunAnalogGainThreshold**	
	Default value	0x4200 (4)
	Purpose	Recommend flash gun analog gain threshold value
	Type	FLOAT
0x10c0	**bAntiFlickerMode**	
	Default value	<0> AntiFlickerMode_Inhibit
	Purpose	Anti flicker mode
	Type	CODED
	Possible values	<0> AntiFlickerMode_Inhibit <1> AntiFlickerMode_ManualEnable <2>AntiFlickerMode_AutomaticEnable

1. Can be controlled in all stable states

Exposure algorithm control

Table 27. Exposure algorithm control

Index	ExposureAlgorithmControl[1]	
0x1119(MSByte) 0x111a(LSByte)	**fpDigitalGainFloor**	
	Default value	1.0000
	Purpose	Minimum digital gain applied by the auto exposure system
	Type	FLOAT
	Possible values	Min 1.0000, Max 2.50000
0x111d(MSByte) 0x111e(LSByte)	**fpDigitalGainCeiling**	
	Default value	2.5000
	Purpose	Maximum Digital Gain applied by the auto exposure system
	Type	FLOAT
	Possible values	Min 1.0000, Max 2.50000

1. Can be controlled in all stable states

Exposure status [read only]

Table 28. Exposure status [read only]

Index	ExposureStatus [read only]	
0x1209(MSByte) 0x120a(LSByte)	**uwCoarseIntegrationPending_lines**	
	Purpose	The course period of time over which every pixel is integrating is a multiple of the number of lines times the number of pixels on the line times the pixel period. These registers show the number of lines of integration which will be used for the next frame.
	Type	FLOAT
0x120d(MSByte) 0x120e(LSByte)	**uwFineIntegrationPending_pixels**	
	Purpose	The fine period of integration is detailed as a added of
	Type	FLOAT
0x1211(MSByte) 0x1212(LSByte)	**fpAnalogGainPending**	
	Purpose	Minimum digital gain applied by the auto exposure system
	Type	FLOAT
0x1215(MSByte) 0x1216(LSByte)	**fpDigitalGainPending**	
	Purpose	Maximum Digital Gain applied by the auto exposure system
	Type	FLOAT
0x1219(MSByte) 0x121a(LSByte)	**fpDesiredExposureTime_us**	
	Purpose	Time in µsecond that the system would like to use for correct exposure
	Type	FLOAT

Table 28. Exposure status [read only] (continued)

Index	ExposureStatus [read only]	
0x121d(MSByte) 0x121e(LSByte)	**fpCompiledExposureTime_us**	
	Purpose	Time in µsecond that the system uses
	Type	FLOAT
0x1285(MSByte) 0x1286(LSByte)	**fpFinalDesiredExposureTime_us**	
	Purpose	Absolute time in µsecond that the system would like to use for correct exposure, this is the value which is used to control the dampers
	Type	FLOAT

White balance control

Table 29. White balance control

Index	WBControlParameters(1)	
0x1380	**bWhiteBalanceMode**	
	Default value	<1> AUTOMATIC
	Purpose	For setting Mode of the white balance
	Type	CODED
	possible values	<0> OFF - No White balance, all gains will be unity in this mode <1> AUTOMATIC - Automatic mode, relative step is computed here <3> MANUAL_RGB - User manual mode, gains are applied manually <4> DAYLIGHT_PRESET - DAYLIGHT and all the modes below, fixed value of gains are applied here. <5> TUNGSTEN_PRESET <6> FLUORESCENT_PRESET <7> HORIZON_PRESET <8> MANUAL_color_TEMP <9> FLASHGUN_PRESET
0x1382	**bManualRedGain**	
	Default value	0x00
	Purpose	User setting for Red Channel gain
	Type	BYTE
	possible values	0 to 255 **Applied_Red_Gain** = (1 + **bManualRedGain / 128) / MinGain** Where **MinGain** = the smallest value from either **Applied_Red_Gain**, **Applied_Green_Gain** or **Applied_Blue_Gain**

Table 29. White balance control (continued)

Index	WBControlParameters[1]	
0x1384	**bManualGreenGain**	
	Default value	0x00
	Purpose	User setting for Green Channel gain
	Type	BYTE
	possible values	0 to 255 **Applied_Green_Gain** = (1 + **bManualGreenGain / 128) / MinGain** Where **MinGain** = the smallest value from either **Applied_Red_Gain**, **Applied_Green_Gain** or **Applied_Blue_Gain**
0x1386	**bManualBlueGain**	
	Default value	0x00
	Purpose	User setting for Blue Channel gain
	Type	BYTE
	possible values	0 to 255 **Applied_Blue_Gain** = (1 + **bManualBlueGain / 128) / MinGain** Where **MinGain** = the smallest value from either **Applied_Red_Gain**, **Applied_Green_Gain** or **Applied_Blue_Gain**
0x1388	**bMiscSettings**	
	Default value	0x00
	Purpose	Bit [2] fFreezeWhiteBalance
	Type	BYTE
	possible values	0x00 - Normal 0x04 - Freeze white balance
0x138b (MSByte) 0x138c(LSByte)	**fpFlashRedGain**	
	Default value	0x3e80 (1.250)
	Purpose	When FLASHGUN_PRESET is selected in bWhiteBalanceMode the **fpFlashRedGain** is used.
	Type	FLOAT
0x138f (MSByte) 0x1390(LSByte)	**fpFlashGreenGain**	
	Default value	0x3e00 (1.000)
	Purpose	When FLASHGUN_PRESET is selected in bWhiteBalanceMode the **fpFlashGreenGain** is used.
	Type	FLOAT

Table 29. White balance control (continued)

Index	WBControlParameters[1]	
0x1393(MSByte) 0x1394(LSByte)	**fpFlashBlueGain**	
	Default value	0x3e8a (1.269531)
	Purpose	When FLASHGUN_PRESET is selected in bWhiteBalanceMode the **fpFlashBlueGain** is used.
	Type	FLOAT

1. Can be controlled in all stable states

White balance constrainer controls

Table 30. White balance constrainer controls

Index	WhiteBalanceConstrainerControls[1]	
0x1501(MSByte) 0x1502(LSByte)	**fpRedA**	
	Default value	0.281738
	Purpose	Sets location of Ref. A point on constrainer control
	Type	FLOAT
0x1505(MSByte) 0x1506(LSByte)	**fpBlueA**	
	Default value	0.4223636
	Purpose	Sets location of Ref. A point on constrainer control
	Type	FLOAT
0x1509(MSByte) 0x150a(LSByte)	**fpRedB**	
	Default value	0.4223636
	Purpose	Sets location of Ref. B point on constrainer control
	Type	FLOAT
0x150d(MSByte) 0x150e(LSByte)	**fpBlueB**	
	Default value	0.281738
	Purpose	Sets location of Ref. B point on constrainer control
	Type	FLOAT
0x1511(MSByte) 0x1512(LSByte)	**fpMaximumDistanceAllowedFromLocus**	
	Default value	0.281738
	Purpose	Sets distance from locus within which the white balance is constrained.
	Type	FLOAT

Table 30. White balance constrainer controls (continued)

Index	WhiteBalanceConstrainerControls[1]	
0x1514	**fEnableConstrainedWhiteBalance**	
	Default value	0.4223636
	Purpose	Maximum Digital Gain applied by the auto exposure system
	Type	CODED
	Possible values	<0> FALSE <1> TRUE

1. Can be controlled in all stable states

Image stability [read only]

Table 31. Image stability [read only]

Index	Image stability [read only]	
0x1800	**fWhiteBalanceStable**	
	Default value	0x00
	Purpose	Indicated that white balance system is stable/unstable
	Type	CODED
	Possible values	<0> Unstable <1> Stable
01802	**fExposureStable**	
	Default value	0x00
	Purpose	Indicated that the exposure system is stable/unstable
	Type	CODED
	Possible values	<0> Unstable <1> Stable
0x1808	**fStable**	
	Default value	0x00
	Purpose	Consolidated flag to indicate whether the system is stable/unstable
	Type	CODED
	Possible values	<0> Unstable <1> Stable

Sensor setup

Table 32. Sensor setup

Index	SensorSetup[1]	
0x1885 0x1886	**fpRedtilt**	
	Default value	0x3e00
	Purpose	An offset can be applied to the Red data from the sensor array to ensure help match the response of all the color channels
	Type	BYTE
0x1889 0x188a	**fpGreentilt**	
	Default value	0x3e00
	Purpose	An offset can be applied to the Green data from the sensor array to ensure help match the response of all the color channels
	Type	BYTE
0x188d 0x188e	**fpbluetilt**	
	Default value	0x3e00
	Purpose	An offset can be applied to the Blue data from the sensor array to ensure help match the response of all the color channels
	Type	BYTE
0x1890	**bBlackCorrectionOffset**	
	Default value	0x00
	Purpose	A black correction offset can be added to the sensor pedestal. This maybe adjusted to improve the black level.
	Type	BYTE

1. Can be controlled in all stable states

ExpSensor constants

Table 33. ExpSensor constants

Index	ExpSensorConstants[1]	
0x1901(MSByte) 0x1902(LSByte)	**uwSensorAnalogGainFloor**	
	Default value	0x00
	Purpose	Sets the minimum gain that will be used by the auto exposure system.
	Type	BYTE-Bits [7:0] - NOTE: Bits[16:8 and 3:0] are not used.
	Possible values.	Min Analogue gain = 256 / (256 - **uwSensorAnalogGainFloor)**
0x1905(MSByte) 0x1906(LSByte)	**uwSensorAnalogGainFloor**	
	Default value	0xd0
	Purpose	Sets the maximum gain that will be used by the auto exposure system.
	Type	BYTE-Bits [7:0] - NOTE: Bits[16:8 and 3:0] are not used.
	Possible values	Max Analogue gain = 256 / (256 - **uwSensorAnalogGainFloor)**

1. Can be controlled in all stable states

Flash control

Table 34. Flash control

Index	FlashControl[1]	
0x1980	**bFlashMode**	
	Default value	<0> FLASH_OFF
	Purpose	Select the flash type and on/off
	Type	CODED
	Possible values	<0> FLASH_OFF <1>FLASH_TORCH <2>FLASH_PULSE
0x1983(MSB) 0x1984(LSB)	**uwFlashOffLine**	
	Default value	0x021c (540)
	Purpose	In flash_pulse mode, used to control off line
	Type	UINT16

1. Can be controlled in all stable states

Flash status [read only]

Table 35. Flash status

Index	FlashStatus [read only]	
0x1a00	**fFlashRecommend**	
	Default value	<0> FALSE
	Purpose	This flag is set if the Exposure Control system reports that the image is underexposed and so the flashgun is recommended to the Host controlled by register **fpRecommendFlashGunAnalogGainThreshold**. It is at the discretion of Host to use it or not for the following still grab.
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x1a02	**fFlashGrabComplete**	
	Default value	<0> FALSE
	Purpose	This flag indicates that the FlashGun Image has been grabbed.
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

Lens shading correction

Table 36. Lens shading correction

Index	LensShadingcorrection(1)	
0x1c01 (MSByte) 0x1c02 (LSByte)	**uwHorizontalOffsetR**	
	Default value	0x00
	purpose	Controls lens shading Horizontal Offset for red pixels
	type	VPIP_UINT16
0x1c05 (MSByte) 0x1c06 (LSByte)	**uwHorizontalOffsetGR**	
	Default value	
	purpose	Controls lens shading Horizontal Offset for green_red pixels
	type	VPIP_UINT16
0x1c09 (MSByte) 0x1c0a (LSByte)	**uwHorizontalOffsetGB**	
	Default value	
	purpose	Controls lens shading Horizontal Offset for green_blue pixels
	type	VPIP_UINT16

Table 36. Lens shading correction (continued)

Index	LensShadingcorrection[1]	
0x1c0e (MSByte) 0x1c0f (LSByte)	**uwHorizontalOffsetB**	
	Default value	
	purpose	Controls lens shading Horizontal Offset for blue pixels
	type	VPIP_UINT16
0x1c11 (MSByte) 0x1c12 (LSByte)	**uwVerticalOffsetR**	
	Default value	
	purpose	Controls lens shading Vertical Offset for red pixels
	type	VPIP_UINT16
0x1c15 (MSByte) 0x1c16 (LSByte)	**uwVerticalOffsetGR**	
	Default value	
	purpose	Controls lens shading Vertical Offset for green_red pixels
	type	VPIP_UINT16
0x1c19 (MSByte) 0x1c1a (LSByte)	**uwVerticalOffsetGB**	
	Default value	
	purpose	Controls lens shading Vertical Offset for green_blue pixels
	type	VPIP_UINT16
0x1c1e (MSByte) 0x1c1f (LSByte)	**uwVerticalOffsetB**	
	Default value	
	purpose	Controls lens shading Vertical Offset for blue pixels
	type	VPIP_UINT16
0x1c20	**iR2RCoefficient**	
	Default value	
	purpose	Controls lens shading R2 component for red pixels.
	type	VPIP_INT8
0x1c22	**iR2GRCoefficient**	
	Default value	
	purpose	Controls lens shading R2 component for green_red pixels.
	type	VPIP_INT8

Table 36. Lens shading correction (continued)

Index	LensShadingcorrection[1]	
0x1c24	**iR2GBCoefficient**	
	Default value	
	purpose	Controls lens shading R2 component for green_blue pixels.
	type	VPIP_INT8
0x1c26	**iR2BCoefficient**	
	Default value	
	purpose	Controls lens shading R2 component for blue pixels.
	type	VPIP_INT8
0x1c28	**iR4RCoefficient**	
	Default value	
	purpose	Controls lens shading R4 component for red pixels.
	type	VPIP_INT8
0x1c2a	**iR4GRCoefficient**	
	Default value	
	purpose	Controls lens shading R4 component for green_red pixels.
	type	VPIP_INT8
0x1c2c	**iR4GBCoefficient**	
	Default value	
	purpose	Controls lens shading R4 component for green_blue pixels.
	type	VPIP_INT8
0x1c2e	**iR4BCoefficient**	
	Default value	0x00
	purpose	Controls lens shading R4 component for blue pixels.
	type	VPIP_INT8

Table 36. Lens shading correction (continued)

Index	LensShadingcorrection[1]	
0x1c30	**fDisable**	
	Default value	0x00
	purpose	To enable or disable Lens shading block
	type	Flag_e
	possible values	<0> VPIP_FALSE <1> VPIP_TRUE

1. Can be controlled in all stable states

Defect correction 1 controls

Table 37. Defect correction 1 controls

Index	DefectCorrection1Controls[1]	
0x1c80	**fDisableFilter**	
	Default value	<0> FALSE
	Purpose	Disable defect correction 1
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

1. Can be controlled in all stable state

Defect correction 2 controls

Table 38. Defect correction 2 controls

Index	DefectCorrection 2Controls[1]	
0x1d00	**fDisableFilter**	
	Default value	<0> FALSE
	Purpose	Disable defect correction 2
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

1. Can be controlled in all stable state

Interpolation control

Table 39. Interpolation control

Index	InterpolationControl(1)	
0x1d80	**bAntiAliasFilterSuppress**	
	Default value	0x08
	Purpose	Anti alias filter suppress
	Type	BYTE

1. Can be controlled in all stable state

Color matrix dampers

Table 40. color matrix dampers

Index	colorMatrixDamper(1)	
0x1e00	**fDisable**	
	Default value	<0> FALSE
	Purpose	set to disable color matrix damper and therefore ensure that all the color matrix coefficients remain constant under all conditions.
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x1e03(MSByte) 0x1e04(LSByte)	**fpLowThreshold**	
	Default value	0x67d1 (2000896)
	Purpose	Low Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x1e07(MSByte) 0x1e08(LSByte)	**fpHighThreshold**	
	Default value	0x6862 (2498560)
	Purpose	High Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x1e0b (MSByte) 0x1e0c(LSByte)	**fpMinimumOutput**	
	Default value	0x3acd (0.350098)
	Purpose	Minimum possible damper output for the color matrix
	Type	FLOAT

1. Can be controlled in all stable state

sRGB color matrix

Table 41. sRGB color matrix

Index	sRGBColorMatrix[1] [2]	
0x1e81 (MSByte) 0x1e82(LSByte)	**fpGinR**	
	Default value	
	Purpose	Green in Red element
	Type	FLOAT
0x1e85 (MSByte) 0x1e86 (LSByte)	**fpBinR**	
	Default value	
	Purpose	Blue in Red element
	Type	FLOAT
0x1e89 (MSByte) 0x1e8a (LSByte)	**fpRinG**	
	Default value	
	Purpose	Red in Green element
	Type	FLOAT
0x1e8d (MSByte) 0x1e8e (LSByte)	**fpBinG**	
	Default value	
	Purpose	Blue in Green element
	Type	FLOAT
0x1e91 (MSByte) 0x1e92 (LSByte)	**fpRinB**	
	Default value	0x00
	Purpose	Red in Blue element
	Type	FLOAT
0x1e95 (MSByte) 0x1e96(LSByte)	**fpGinB**	
	Default value	0x00
	Purpose	Green in Blue element
	Type	FLOAT

1. The diagonal elements of the matrix are calculated such at the sun for each row equal unity
2. Can be controlled in all stable states

Peaking control

Table 42. Peaking control

Index	Peaking control[1]	
0x1f00	**bUserPeakGain**	
	Default value	0x0e
	Purpose	controls peaking gain / sharpness applied to the image
	Type	BYTE
0x1f02	**fDisableGainDamping**	
	Default value	<0> FALSE
	Purpose	set to disable damping and therefore ensure that the peaking gain applied remains constant under all conditions
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x1f05 (MSByte) 0x1f06(LSByte)	**fpDamperLowThreshold_Gain**	
	Default value	0x62ac (350208)
	Purpose	Low Threshold for exposure for calculating the damper slope - for gain
	Type	FLOAT
0x1f09 (MSByte) 0x1f0a(LSByte)	**fpDamperHighThreshold_Gain**	
	Default value	0x65d1 (10004488)
	Purpose	High Threshold for exposure for calculating the damper slope - for gain
	Type	FLOAT
0x1f0d (MSByte) 0x1f0e(LSByte)	**fpMinimumDamperOutput_Gain**	
	Default value	0x3d33 (0.799805)
	Purpose	Minimum possible damper output for the gain.
	Type	FLOAT
0x1f10	**bUserPeakLoThresh**	
	Default value	0x1e
	Purpose	Adjust degree of coring. range: 0 - 63
	Type	BYTE
0x1f12	**fDisableCoringDamping**	
	Default value	<0> FALSE
	Purpose	set to ensure that bUserPeakLoThresh is applied to gain block
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

Table 42. Peaking control (continued)

Index	Peaking control[1]	
0x1f14	**bUserPeakHiThresh**	
	Default value	0x30
	Purpose	adjust maximum gain that can be applied. range: 0 - 63
	Type	BYTE
0x1f107(MSByte) 0x1f108(LSByte)	**fpDamperLowThreshold_Coring**	
	Default value	0x624a (300032)
	Purpose	Low Threshold for exposure for calculating the damper slope - for coring
	Type	FLOAT
0x1f1b (MSByte) 0x1f1c(LSByte)	**fpDamperHighThreshold_Coring**	
	Default value	0x656f (900096)
	Purpose	High Threshold for exposure for calculating the damper slope - for coring
	Type	FLOAT
0x1f1f (MSByte) 0x1f20(LSByte)	**fpMinimumDamperOutput_Coring**	
	Default value	0x3a00 (0.2500)
	Purpose	Minimum possible damper output for the Coring.
	Type	FLOAT

1. Can be controlled in all stable states

RGB to YCbCr matrix manual control

Table 43. RGB to YCbCr matrix manual control

Index	RGB to YCbCr matrix (1)	
0x2080	**fRgbToYCbCrManuCtrl**	
	Default value	<0> FALSE
	Purpose	Enables manual RGB to YCbCr matrix
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x2083 (MSByte) 0x2084(LSByte)	**w0_0**	
	Default value	0x00
	Purpose	Row 0 Column 0 of YCbCr matrix
	Type	UINT_16
0x2087 (MSByte) 0x2088(LSByte)	**w0_1**	
	Default value	0x00
	Purpose	Row 0 Column 1 of YCbCr matrix
	Type	UINT_16
0x208b (MSByte) 0x208c(LSByte)	**w0_2**	
	Default value	0x00
	Purpose	Row 0 Column 2 of YCbCr matrix
	Type	UINT_16
0x208f (MSByte) 0x2090(LSByte)	**w1_0**	
	Default value	0x00
	Purpose	Row 1 Column 0 of YCbCr matrix
	Type	UINT_16
0x2093 (MSByte) 0x2094(LSByte)	**w1_1**	
	Default value	0x00
	Purpose	Row 1 Column 1 of YCbCr matrix
	Type	UINT_16
0x2097 (MSByte) 0x2098(LSByte)	**w1_2**	
	Default value	0x00
	Purpose	Row 1 Column 2 of YCbCr matrix
	Type	UINT_16

Table 43. RGB to YCbCr matrix manual control

Index	RGB to YCbCr matrix (1)	
0x209b (MSByte) 0x209c(LSByte)	**w2_0**	
	Default value	0x00
	Purpose	Row 2 Column 0 of YCbCr matrix
	Type	UINT_16
0x209f (MSByte) 0x20a0(LSByte)	**w2_1**	
	Default value	0x00
	Purpose	Row 2 Column 1 of YCbCr matrix
	Type	UINT_16
0x20a3(MSByte) 0x20a4(LSByte)	**w2_2**	
	Default value	0x00
	Purpose	Row 2 Column 2 of YCbCr matrix
	Type	UINT_16
0x20a7 (MSByte) 0x20a8(LSByte)	**YinY**	
	Default value	0x00
	Purpose	Y in Y
	Type	UINT_16
0x20ab (MSByte) 0x20ac(LSByte)	**YinCb**	
	Default value	0x00
	Purpose	Y in Cb
	Type	UINT_16
0x20af (MSByte) 0x20b0(LSByte)	**YinCr**	
	Default value	0x00
	Purpose	Y in Cr
	Type	UINT_16

1. Can be controlled in all stable states

Gamma manual control

Table 44. Gamma manual control

Index	GammaManuControl[1]	
0x2100	**fGammaManuCtrl**	
	Default value	<0> FALSE
	Purpose	Enables manual Gamma Setup
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x2102	**bRPeakGamma**	
	Default value	0x00
	Purpose	Peaked Red channel gamma value
	Type	BYTE
0x2104	**bGPeakGamma**	
	Default value	0x00
	Purpose	Peaked Green channel gamma value
	Type	BYTE
0x2106	**bBPeakGamma**	
	Default value	0x00
	Purpose	Peaked Blue channel gamma value
	Type	BYTE
0x2108	**bRUnPeakGamma**	
	Default value	0x00
	Purpose	Unpeaked Red channel gamma value
	Type	BYTE
0x210a	**bGUnPeakGamma**	
	Default value	0x00
	Purpose	Unpeaked Green channel gamma value
	Type	BYTE
0x210c	**bBUnPeakGamma**	
	Default value	0x00
	Purpose	Unpeaked Blue channel gamma value
	Type	BYTE

1. Can be controlled in all stable states

Fade to black

Table 45. Fade to black

Index	FadeToBlack[1]	
0x2280	**fDisable**	
	Default value	<0> FALSE
	Purpose	Flag_e
	Type	<0> FALSE <1> TRUE
0x2283(MSByte) 0x2284(LSByte)	**fpBlackValue**	
	Default value	0x0000 (0.000)
	Purpose	Black value
	Type	FLOAT
0x2287 (MSByte) 0x2288(LSByte)	**fpDamperLowThreshold**	
	Default value	0x6d56 (6995968)
	Purpose	Low Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x228b (MSByte) 0x228c(LSByte)	**fpDamperHighThreshold**	
	Default value	0x6cdc (11993088)
	Purpose	High Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x228f (MSByte) 0x2290(LSByte)	**fpDamperOutput**	
	Default value	0x0 (0.0000)
	Purpose	Minimum possible damper output.
	Type	FLOAT

1. Can be controlled in all stable states

Dither control

Table 46. Dither control

Index	DitherControl[1]	
0x2300	**bDither Block control**	
	Default value	0x00
	Purpose	Control of dither block
	Type	BYTE
	Possible values	0 -> Control Automatic 1 -> Control Manual

1. Can be controlled in all stable state

Output control

Table 47. Output control

Index	OutputControl[1]	
0x2380	**bYCbCrSetup**	
	Default value	0x00
	Type	CODED
	Purpose	Changes the output order of YCbCr
	flag bits	[[0] Cb_Cr_Order 0-> Cr before Cb 1-> Cb before Cr [1] Y_ C order 0-> Cb or Cr first 1-> Y first
0x2382	**bRgbSetup**	
	Default value	0x00
	Type	CODED
	Purpose	Setup the RGB padding and order
	flag bits	[0]rgb444_format 0-> zero padding 1->not zero padding [3:1] RGB output order 0-> GBR 1-> RBG 2-> BRG 3-> GRB 4-> RGB 5-> BGR
0x238c	**bBlank_Value_1**	
	Default value	0x10
	Type	CODED
	Purpose	First Blanking value used during interline & interframe blanking
	Possible values	<16> BlankingMSB_Default
0x238e	**bBlank_Value_2**	
	Default value	0x80
	Type	CODED
	Purpose	First Blanking value used during interline & interframe blanking
	Possible values	<128> BlankingLSB_Default

Table 47. Output control (continued)

Index	OutputControl(1)	
0x2390	**bHSyncSetup**	
	Default value	0x0b
	Type	CODED
	flag bits	[0] HSyncSetup_sync_en 1-> enable hsync 0-> disable hsync [1] HSyncSetup_sync_pol 1-> hsync active high 0-> hsync active low [2] HSyncSetup_only_activelines 1-> hsync qualifies only active lines 0-> hsync qualifies active and non active lines, (PCLK will be on for non active lines) [3] HSyncSetup_track_henv 1-> automatic tracking hsync, ensures that the Hsync qualifies from the first active pixel to last active pixel 0 -> programmable rise and fall of the hsync signal
0x2392	**bVSyncSetup**	
	Default value	0x07
	Type	CODED
	flag bits	[0] VSyncSetup_sync_en 1-> enable vsync 0-> disable vsync [1] VSyncSetup_pol 1-> vsync active high 0-> vsync active low [2] VSyncSetup_2_sel 1-> automatic tracking the vsync, so that it covers from the first active line to last active line 0->manually set the rising and falling of the vsync signal
0x2395MSByte) 0x2396(LSByte)	**bHsyncRisingH**	
	Default value	0x00
	Type	BYTE
	Purpose	Operates only when **bHSyncSetup** bit [3] = 0
0x2399MSByte) 0x239a(LSByte)	**bHsyncFallingH**	
	Default value	0x00
	Type	BYTE
	Purpose	Operates only when **bHSyncSetup** bit [3] = 0

Table 47. Output control (continued)

Index	OutputControl[1]	
0x239d(MSByte) 0x239e(LSByte)	**bVsyncRisingFine**	
	Default value	0x00
	Type	BYTE
	Purpose	Operates only when **bVSyncSetup** bit [2] = 0
0x23a1(MSByte) 0x23a2(LSByte)	**bVsyncFallingFineH**	
	Default value	0x00
	Type	BYTE
	Purpose	Operates only when **bVSyncSetup** bit [2] = 0
0x23a5(MSByte) 0x23a6(LSByte)	**bVsyncRisingCoarse**	
	Default value	0x00
	Type	UINT_16
	Purpose	Operates only when **bVSyncSetup** bit [2] = 0
0x23a9MSByte) 0x23aa(LSByte)	**bVsyncFallingCoarseH**	
	Default value	0x01
	Type	BYTE
	Purpose	Operates only when **bVSyncSetup** bit [2] = 0
0x23ae	**bSyncCodeSetup**	
	Default value	0x01
	Type	CODED
	flag bits	[0] SyncCodeSetup_ins_code_en 1-> enable embedded codes 0-> disable embedded codes [1] SyncCodeSetup_frame 0=ITU-656, 1 ITU-CSI 0-> ITU-656 codes 1-> ITU-CSI codes [2] SyncCodeSetup_field_bit - ITU-656 only 1-> Next field is odd frame 0-> Next field is even frame [3] SyncCodeSetup_field_tag - ITU-656 only 1-> the sync code for odd and even frames will toggle 0 -> the sync code will be constant for all frames [4] SyncCodeSetup_field_load - ITU-656 only 1-> Load new settings for bits [3:2] 0 -> New settings for bits [3:2] not consumed

Table 47. Output control (continued)

Index	OutputControl(1)	
0x23b0	**bPClkSetup**	
	Default value	0x05
	Type	CODED
	flag bits	[0] PClkSetup_prog_lo 1-> Rising Edge of PCLK 0-> Falling Edge of PCLK [1] PClkSetup_prog_hi 1-> Polarity of PLCK normally high 0-> Polarity of PLCK normally low [2] PClkSetup_sync_en 1-> Enable PCLK during Embedded sync codes 0-> Disable PCLK during Embedded sync codes [3] PClkSetup_hsync_en_n 1-> enable the PCLK outside the HSYNC active period 0-> disable the PCLK outside the HSYNC active period [4] PClkSetup_hsync_en_n_track_internal 1-> use the internal tracked hsync for the qualification of the bit[3] 0-> use the manually set hsync for the qualification of the bit[3] [5] PClkSetup_vsync_n 1-> enable pclk outside VSYNC active period 0-> disable pclk outside VSYNC active period [6] PClkSetup_vsync_n_track_internal 1-> use the internal tracked vsync for the qualification of the bit[5] 0-> use the manually set vsync for the qualification of the bit[5] [7] PClkSetup_freer 1-> Set the PCLK to free run 0 ->Set the PCLK to be controlled
0x23b2	**fPclkEn**	
	Default value	<1> TRUE
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

1. Changes only consumed during a change to RUN mode

NoRA controls

Table 48. NoRA controls

Index	NoRAControls[1]	
0x2400	**fDisable**	
	Default value	<0> NoraCtrl_auto
	Type	Flag_e
	Possible values	<0> NoraCtrl_auto - switches off NoRA for scaled outputs <1> NoraCtrl_ManuDisable - Always off <2> NoraCtrl_ManuEnable - Always on
0x2402	**bUsage**	
	Default value	0x04
	Purpose	
	Type	BYTE
0x240a	**fDisableNoroPromoting**	
	Default value	<0> FALSE
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x240d (MSByte) 0x240e(LSByte)	**fpDamperLowThreshold**	
	Default value	0x6862 (2498560)
	Purpose	Low Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x2411 (MSByte) 0x2412(LSByte)	**fpDamperHighThreshold**	
	Default value	0x6a62 (4997120)
	Purpose	High Threshold for exposure for calculating the damper slope
	Type	FLOAT
0x2415 (MSByte) 0x2416(LSByte)	**MinimumDamperOutput**	
	Default value	0x3a00 (0.2500)
	Purpose	Minimum possible damper output.
	Type	FLOAT

1. Can be controlled in all stable states

Special effects control

Table 49. Special effects control

Index	SpecialEffectsControl(1)	
0x2480	**fNegative**	
	Default value	<0> FALSE
	Purpose	INverse of image generated
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x2482	**fSolarising**	
	Default value	<0> FALSE
	Purpose	Colors reversed and edges enhanced to give solarizing effect
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x2484	**fSketch**	
	Default value	<0> FALSE
	Purpose	Edges of the image enhanced and output to give the impression of a pencil sketch
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE
0x2486	**fEmboss**	
	Default value	<0> FALSE
	Purpose	Uses mathematical functions to create the effect that the image detail is raised above the surface of the background
	Type	Flag_e
	Possible values	<0> FALSE <1> TRUE

Table 49. Special effects control (continued)

Index	SpecialEffectsControl[1]	
0x2488	**fColorEffects**	
	Default value	<0>
	Purpose	Special effect added to the output image
	Type	0 Normal 1 Red only - Red information preserved, all other color info set to mono 2 Yellow only - Red and Green information preserved, blue info set to mono 3 Green only - Green information preserved, all other color info set to mono 4 Blue only - Blue information preserved, all other color info set to mono 5 Black and white - Monochrome 6 Sepia - to mimic the wet film effect of replacing the silver in black and white print with brown silver sulphide 7 Antique - to mimic the yellowing effect when an old black and white photo ages NOTE; All color effects can operate on YCbCr and JPEG formats but only Sepia and B & W operate when using RGB formats.

1. Can be controlled in all stable states

JPEG image characteristics

Table 50. JPEG image characteristics

Index	JPEGImageCharacteristics(1)	
0x23b4	**bJPEG_Fill_Val**	
	Default value	0xa5
	Purpose	Value of padding bytes inserted at the end of the JPEG data sequence to fill the final packet, must equal **bJPEG_Padding**
	Type	BYTE
0x23b6	**bJPEG_Padding**	
	Default value	0xa5
	Purpose	Value of padding bytes inserted at the end of the JPEG data sequence to fill the final packet, must equal **bJPEG_Fill_Val**
	Type	BYTE
0x2508	**bHiSqueezeValue**	
	Default value	0x18
	Purpose	Sets squeeze factor for HIgh Quality
	Type	BYTE
	Possible values	The Squeeze value can be programmed between the range 6 (Highest quality) to 255 (Low quality)
0x2508	**bMedSqueezeValue**	
	Default value	0x20
	Purpose	Sets squeeze factor for Medium Quality
	Type	BYTE
0x2508	**bLowSqueezeValue**	
	Default value	0x28
	Purpose	Sets squeeze factor for Low Quality
	Type	BYTE
0x2511 0x2512	**uwLinelength**	
	Default value	0x0200
	Purpose	Sets the number of bytes of JPEG data are present in each Packet (line), must equal **uwThres**
	Type	WORD

Table 50. JPEG image characteristics (continued)

Index	JPEGImageCharacteristics(1)	
0x2514	**bOIFCLKRatio**	
	Default value	0x1
	Purpose	Determines the maximum PCLK output rate for the JPEG data, in relation to the YUV pixel clock frequency
	Type	BYTE
	Possible values	0 ->PCLK is same frequency YUV pixel clock, for non JPEG //for JPEG 1-> PCLK is 2 times slower 2-> PCLK is 4 times slower 3-> PCLK is 6 times slower 4-> PCLK is 8 times slower 5-> PCLK is 10 times slower 6-> PCLK is 12 times slower 7-> PCLK is 14 times slower 8-> PCLK is 16 times slower 9-> PCLK is 18 times slower 10-> PCLK is 20times slower
0x251b 0x251c	**uwThres**	
	Default value	0x0200
	Purpose	Sets the number of bytes of JPEG data are present in each Packet (line), must equal **uwLinelength**
	Type	WORD
	Possible values	The number of bytes in a line is 2 x the value programmed into this register, the maximum number of bytes in a line is 2KB.

1. Can be controlled in all stable states

5 Optical specifications

Table 51. Optical specifications(1)

Parameter	Min.	Typ.	Max.	Unit
Optical format		1/3.8		inch
Effective focal length	3.7	3.8	3.9	mm
Aperture (F number)		3.2		
Horizontal field of view		50		deg.
Depth of field	60		infinity	cm
TV distortion	-1		1	%

1. All measurements made at 23°C ± 2°C

6 Electrical characteristics

6.1 Absolute maximum ratings

Table 52. Absolute maximum ratings

Symbol	Parameter	Min.	Max.	Unit
T_{STO}	Storage temperature	-40	85	° C
V_{DD}	Digital power supplies	-0.5	3.3	V
AVDD	Analog power supplies	-0.5	3.3	V

Caution: Stress above those listed under “Absolute Maximum Ratings” can cause permanent damage to the device. This is a stress rating only and functional operations of the device at these or other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

6.2 Operating conditions

Table 53. Supply specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
T_{AF}	Operating temperature, functional (Camera is electrically functional)	-30	25	70	° C
T_{AN}	Operating temperature, nominal (Camera produces acceptable images)	-25	25	55	° C
T_{AO}	Operating temperature, optimal (Camera produces optimal optical performance)	5	25	30	° C
V_{DD}	Digital power supplies operating range (@ module pin(1))	1.7	1.8	2.0	V
		2.4	2.8	3.0	V
AVDD	Analog power supplies operating range (@ module pin(1))	2.4	2.8	3.0	V

1. Module can contain routing resistance up to 5 Ω

6.3 DC electrical characteristics

Note: Over operating conditions unless otherwise specified.

Table 54. DC electrical characteristics

Symbol	Description	Test conditions	Min.	Typ.	Max.	Unit
V_{IL}	Input low voltage	except CLK	-0.3		0.3 V_{DD}	V
V_{IL_CLK}	Input low voltage	CLK pin	-0.3		0.5	V
V_{IH}	Input high voltage		0.7 V_{DD}		V_{DD} + 0.3	V
V_{OL}	Output low voltage	I_{OL} < 2 mA I_{OL} < 4 mA			0.2 V_{DD} 0.4 V_{DD}	V
V_{OH}	Output high voltage	I_{OH}< 4 mA	0.8 V_{DD}			V
I_{IL}	Input leakage current Input pins I/O pins	0 < V_{IN} < V_{DD}			 +/- 10 +/- 1	 µA µA
C_{IN}	Input capacitance, SCL	T_A = 25 °C, freq = 1 MHz			6	pF
C_{OUT}	Output capacitance	T_A = 25 °C, freq = 1 MHz			6	pF
$C_{I/O}$	I/O capacitance, SDA	T_A = 25 °C, freq = 1 MHz			8	pF

Table 55. Typical current consumption - sensor mode UXGA - JPEG UXGA@ 30 fps

Symbol	Description	Test conditions	I_{AVDD}	I_{VDD}		Units
			V_{DD} = 2.8V	V_{DD} = 1.8V	V_{DD} = 2.8V	
I_{PD}	supply current in power down mode	CE=0, CLK = 12 MHz	2.34	0.05	0.07	µA
I_{stanby}	supply current in Standby mode	CE=1, CLK = 12 MHz	0.0024	2.41	2.59	mA
I_{Stop}	supply current in Stop mode	CE=1, CLK = 12 MHz	0.0024	2.11	2.44	mA
I_{Pause}	supply current in Pause mode	CE=1, CLK = 12 MHz	0.58	51	57	mA
I_{run}	supply current in active streaming run mode	CE=1, CLK = 12 MHz	29	120	135	mA

Table 56. Typical current consumption - sensor mode UXGA - JPEG UXGA @ 15 fps

Symbol	Description	Test conditions	I_{AVDD}	I_{VDD}		Units
			V_{DD} = 2.8V	V_{DD} = 1.8V	V_{DD} = 2.8V	
I_{PD}	supply current in power down mode	CE=0, CLK = 12 MHz	2.34	0.05	0.07	µA
I_{stanby}	supply current in Standby mode	CE=1, CLK = 12 MHz	0.0024	2.41	2.59	mA
I_{Stop}	supply current in Stop mode	CE=1, CLK = 12 MHz	0.0024	1.98	2.31	mA
I_{Pause}	supply current in Pause mode	CE=1, CLK = 12 MHz	0.58	28.5	31	mA
I_{run}	supply current in active streaming run mode	CE=1, CLK = 12 MHz	29	67	73	mA

Table 57. Typical current consumption - sensor mode UXGA scaled QVGA RGB565 @ 15 fps

Symbol	Description	Test conditions	I_{AVDD}	I_{VDD}		Units
			V_{DD} = 2.8V	V_{DD} = 1.8V	V_{DD} = 2.8V	
I_{PD}	supply current in power down mode	CE=0, CLK = 12 MHz	2.34	0.05	0.07	µA
I_{stanby}	supply current in Standby mode	CE=1, CLK = 12 MHz	0.0024	2.41	2.59	mA
I_{Stop}	supply current in Stop mode	CE=1, CLK = 12 MHz	0.0024	1.93	2.11	mA
I_{Pause}	supply current in Pause mode	CE=1, CLK = 12 MHz	0.58	15.75	16.44	mA
I_{run}	supply current in active streaming run mode	CE=1, CLK = 12 MHz	29	21.46	23.32	mA

Table 58. Typical current consumption - sensor analogue binning SVGA- scaled QVGA RGB565 @ 15 fps

Symbol	Description	Test conditions	I_{AVDD}	I_{VDD}		Units
			V_{DD} = 2.8V	V_{DD} = 1.8V	V_{DD} = 2.8V	
I_{PD}	supply current in power down mode	CE=0, CLK = 12 MHz	2.34	0.05	0.07	µA
I_{stanby}	supply current in Standby mode	CE=1, CLK = 12 MHz	0.0024	2.41	2.59	mA
I_{Stop}	supply current in Stop mode	CE=1, CLK = 12 MHz	0.0024	1.95	2.1	mA
I_{Pause}	supply current in Pause mode	CE=1, CLK = 12 MHz	0.58	13	14	mA
I_{run}	supply current in active streaming run mode	CE=1, CLK = 12 MHz	29	19	20	mA

6.4 External clock

The VS6724 requires an external clock. This clock is a CMOS digital input. The clock input is fail-safe in power down mode.

Table 59. External clock

CLK	Range			Unit
	Min.	Typ.	Max.	
DC coupled square wave		VDD		V
Clock frequency(1)	6.50		27	MHz

1. The **uwExternalClockFrequencyDenumerator** and **bExternalClockFrequencyDenominator** registers must be configured to match the frequency supplied on the CLK pin.

6.5 Chip enable

CE is a CMOS digital input. The module is powered down when a logic 0 is applied to CE. See *Section Figure 11.: Power up sequence* for further information.

6.6 I²C slave interface

VS6724 contains an I²C-type interface using two signals: a bidirectional serial data line (SDA) and an input-only serial clock line (SCL). See *PLL operation* for detailed description of protocol.

Table 60. Serial interface voltage levels(1)

Symbol	Parameter	Standard mode		Fast mode		Unit
		Min.	Max.	Min.	Max.	
V_{HYS}	Hysteresis of Schmitt Trigger Inputs V_{DD} > 2 V V_{DD} < 2V	 N/A N/A	 N/A N/A	 0.05 V_{DD} 0.1 V_{DD}	 - -	 V V
V_{OL1} V_{OL3}	LOW level output voltage (open drain) at 3mA sink current V_{DD} > 2 V V_{DD} < 2V	 0 N/A	 0.4 N/A	 0 0	 0.4 0.2 V_{DD}	 V V
V_{OH}	HIGH level output voltage	N/A	N/A	0.8 V_{DD}		V
t_{OF}	Output fall time from V_{IHmin} to V_{ILmax} with a bus capacitance from 10 pF to 400 pF	-	250	20+0.1C_b(2)	250	ns
t_{SP}	Pulse width of spikes which must be suppressed by the input filter	N/A	N/A	0	50	ns

1. Maximum $V_{IH} = V_{DDmax}$ + 0.5 V
2. C_b = capacitance of one bus line in pF

Figure 40. Voltage level specification

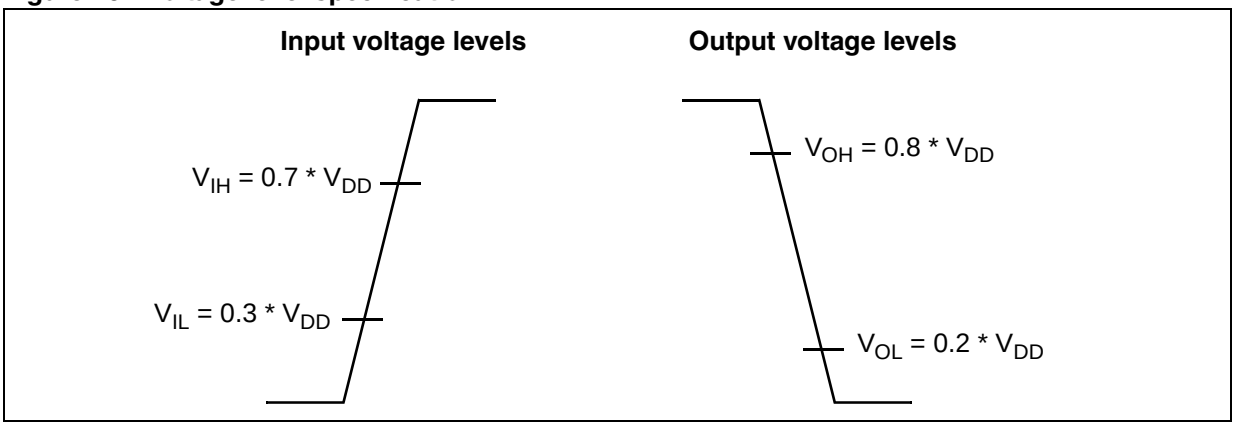


Table 61. Timing specification(1)

Symbol	Parameter	Standard mode		Fast mode		Unit
		Min.	Max.	Min.	Max.	
f_{SCL}	SCL clock frequency	0	100	0	400	kHz
$t_{HD;STA}$	Hold time for a repeated ***start***	4.0	-	0.6	-	µs
t_{LOW}	LOW period of SCL	4.7	-	1.3	-	µs
t_{HIGH}	HIGH period of SCL	4.0	-	0.6	-	µs
$t_{SU;STA}$	Set-up time for a repeated ***start***	4.7	-	0.6	-	µs
$t_{HD;DAT}$	Data hold time (1)	300	-	300	-	ns
$t_{SU;DAT}$	Data Set-up time (1)	250	-	100	-	ns
t_r	Rise time of SCL, SDA	-	1000	$20+0.1C_b$(2)	300	ns
t_f	Fall time of SCL, SDA	-	300	$20+0.1C_b$(2)	300	ns
$t_{SU;STO}$	Set-up time for a ***stop***	4.0	-	0.6	-	µs
t_{BUF}	Bus free time between a ***stop*** and a ***start***	4.7	-	1.3	-	µs
C_b	Capacitive Load for each bus line	-	400	-	400	pF
V_{nL}	Noise Margin at the LOW level for each connected device (including hysteresis)	0.1 V_{DD}	-	0.1 V_{DD}	-	V
V_{nH}	Noise Margin at the HIGH level for each connected device (including hysteresis)	0.2 V_{DD}	-	0.2 V_{DD}	-	V

1. All values are referred to a V_{IHmin} = 0.9 V_{DD} and V_{ILmax} = 0.1 V_{DD}
2. C_b = capacitance of one bus line in pF

Figure 41. Timing specification

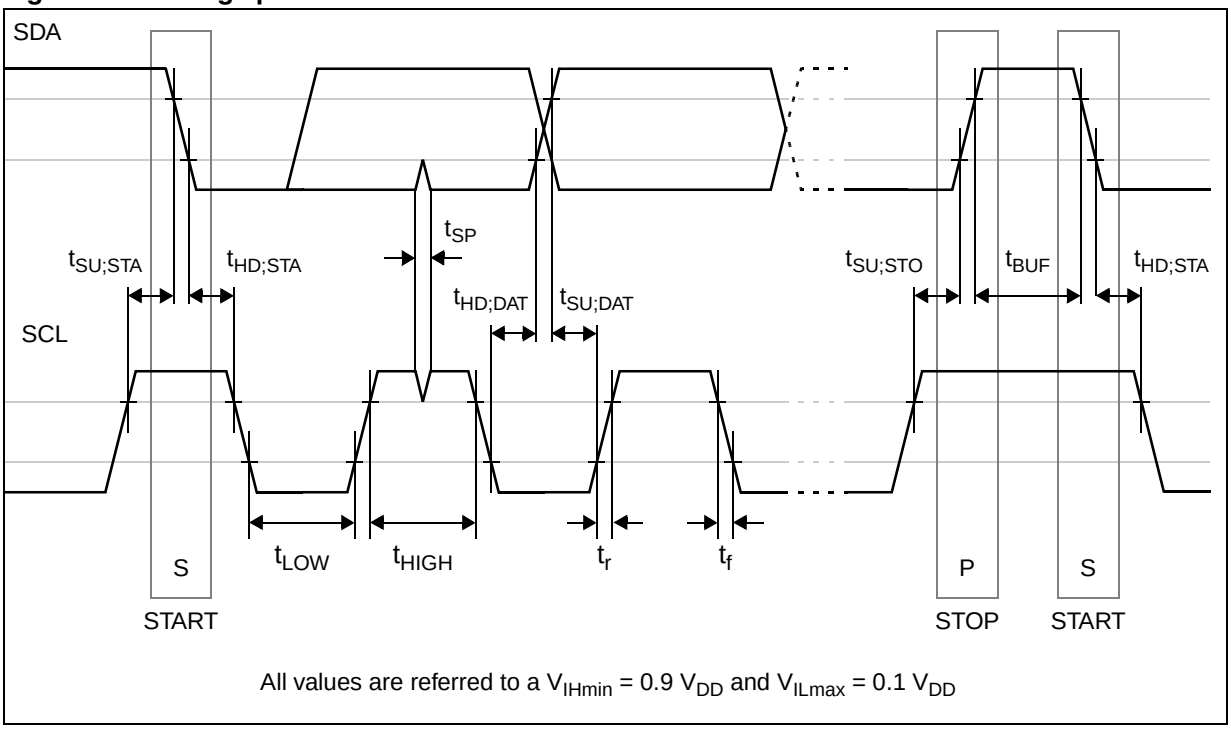
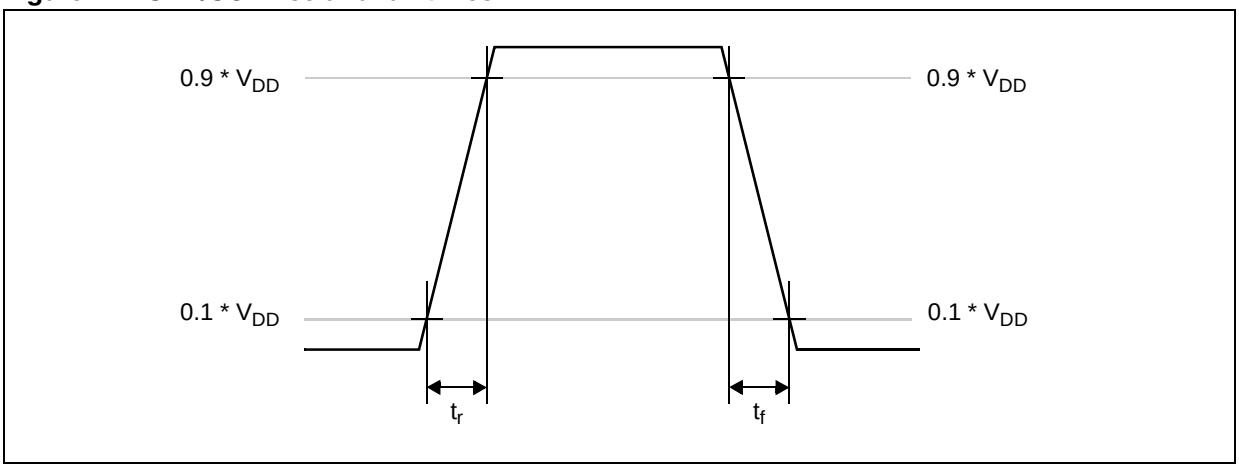


Figure 42. SDA/SCL rise and fall times



6.7 Parallel data interface timing

VS6724 contains a parallel data output port (D[7:0]) and associated qualification signals (HSYNC, VSYNC, PCLK and FSO).

This port can be enabled and disabled (tri-stated) to facilitate multiple camera systems or bit-serial output configurations. The port is disabled (high impedance) upon reset.

Figure 43. Parallel data output video timing

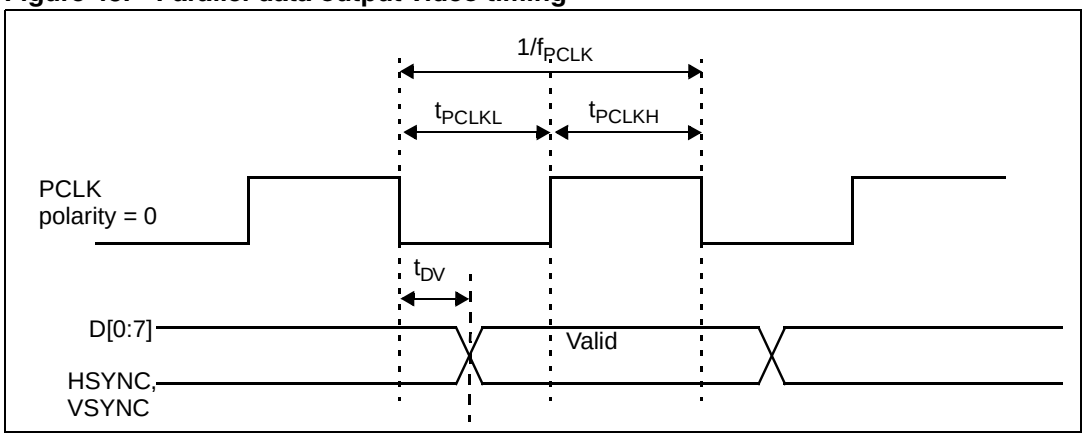
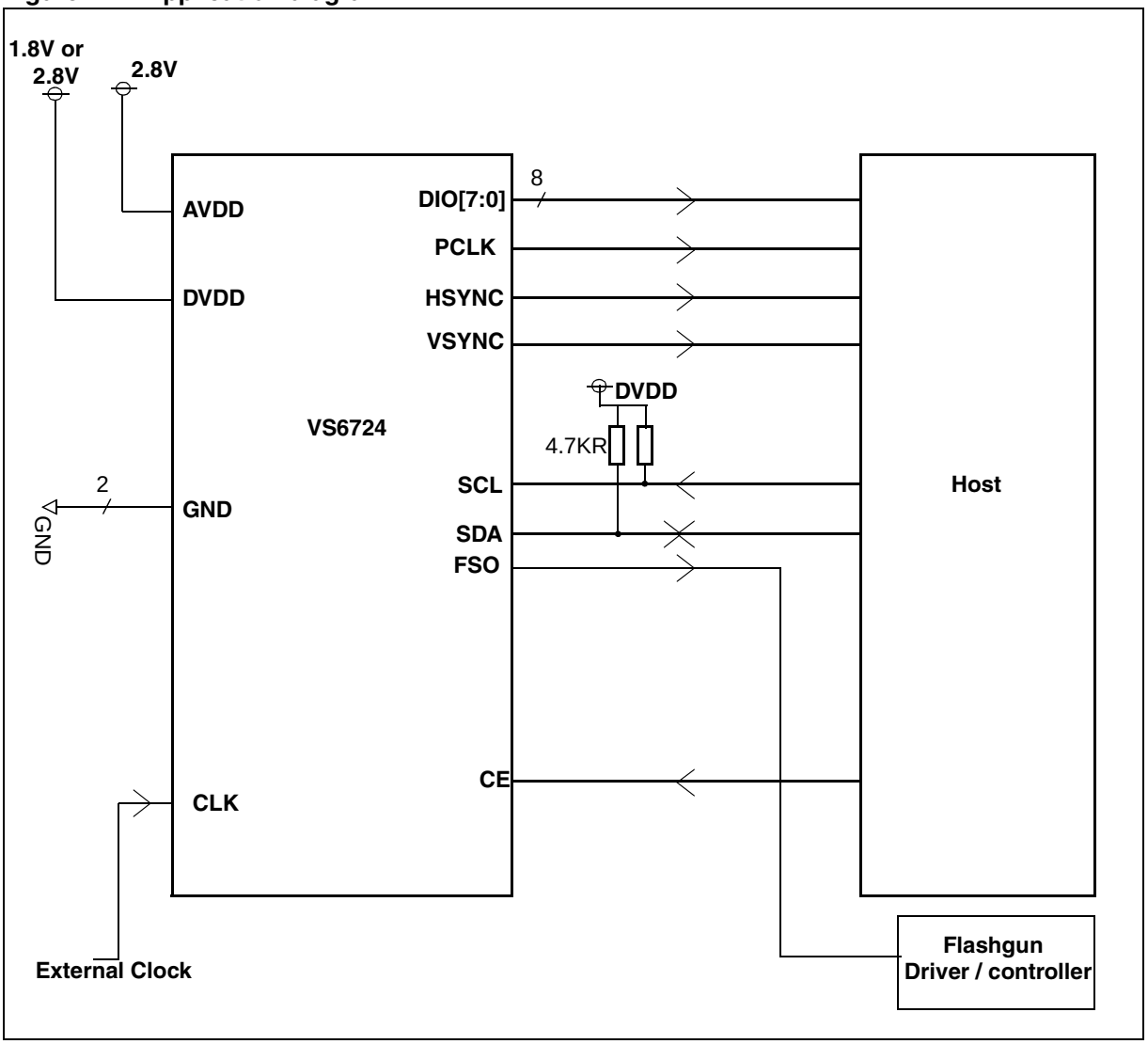


Table 62. Parallel data interface timings

Symbol	Description	Min.	Max.	Unit
f_{PCLK}	PCLK frequency		80	MHz
t_{PCLKL}	PCLK low width	(1/2 * (1/fPCLK)) -1	(1/2 * (1/fPCLK)) +1	ns
t_{PCLKH}	PCLK high width	(1/2 * (1/fPCLK)) -1	(1/2 * (1/fPCLK)) +1	ns
t_{DV}	PCLK to output valid	-1	1	ns

7 Application circuit

Figure 44. Application diagram



8 User precaution

As is common with many CMOS imagers the camera should not be pointed at bright static objects for long periods of time as permanent damage to the sensor may occur.

9 Package mechanical data

Figure 45 and *Figure 46* details the package outline socket module VS6724.

Figure 45. Package outline socket module VS6724

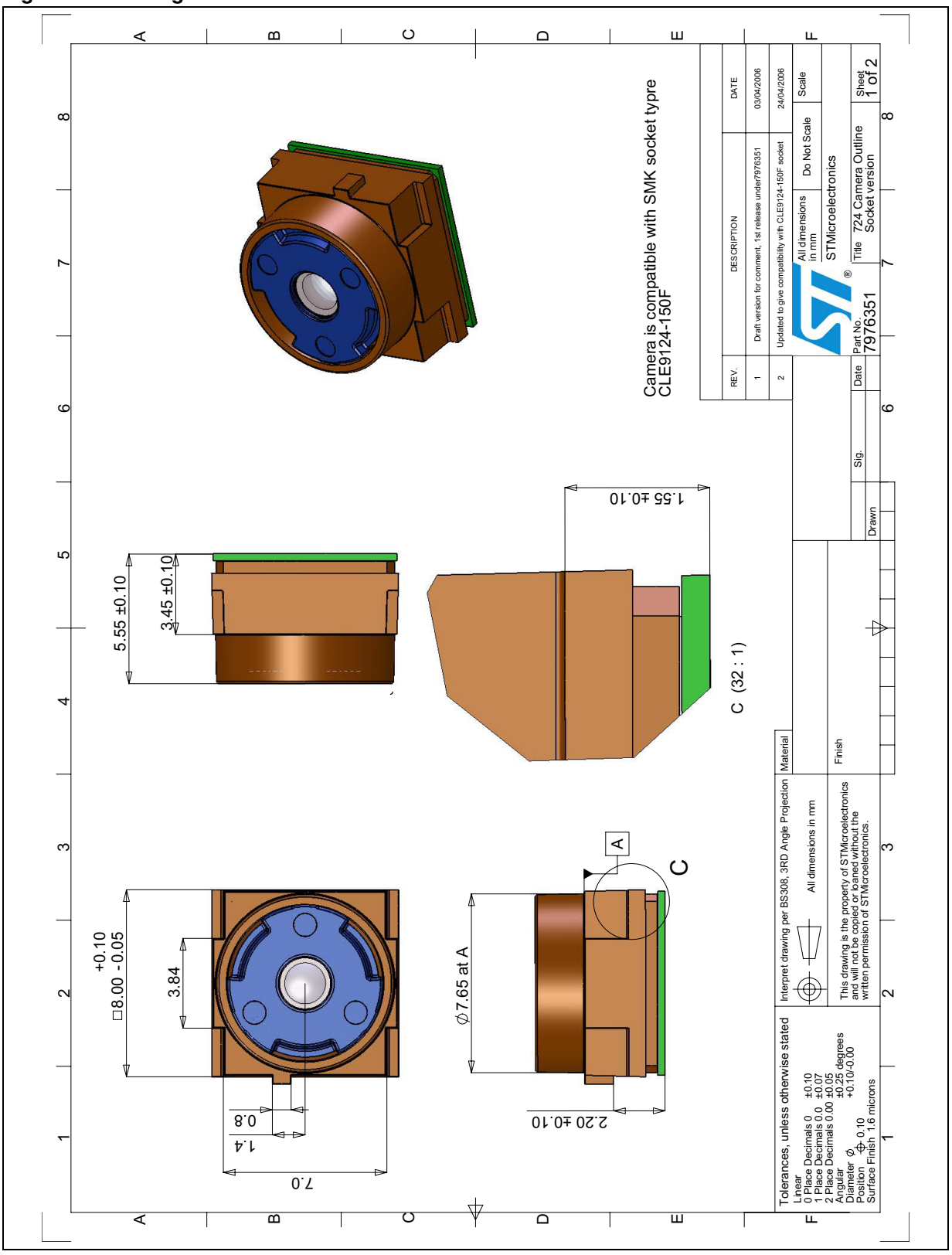
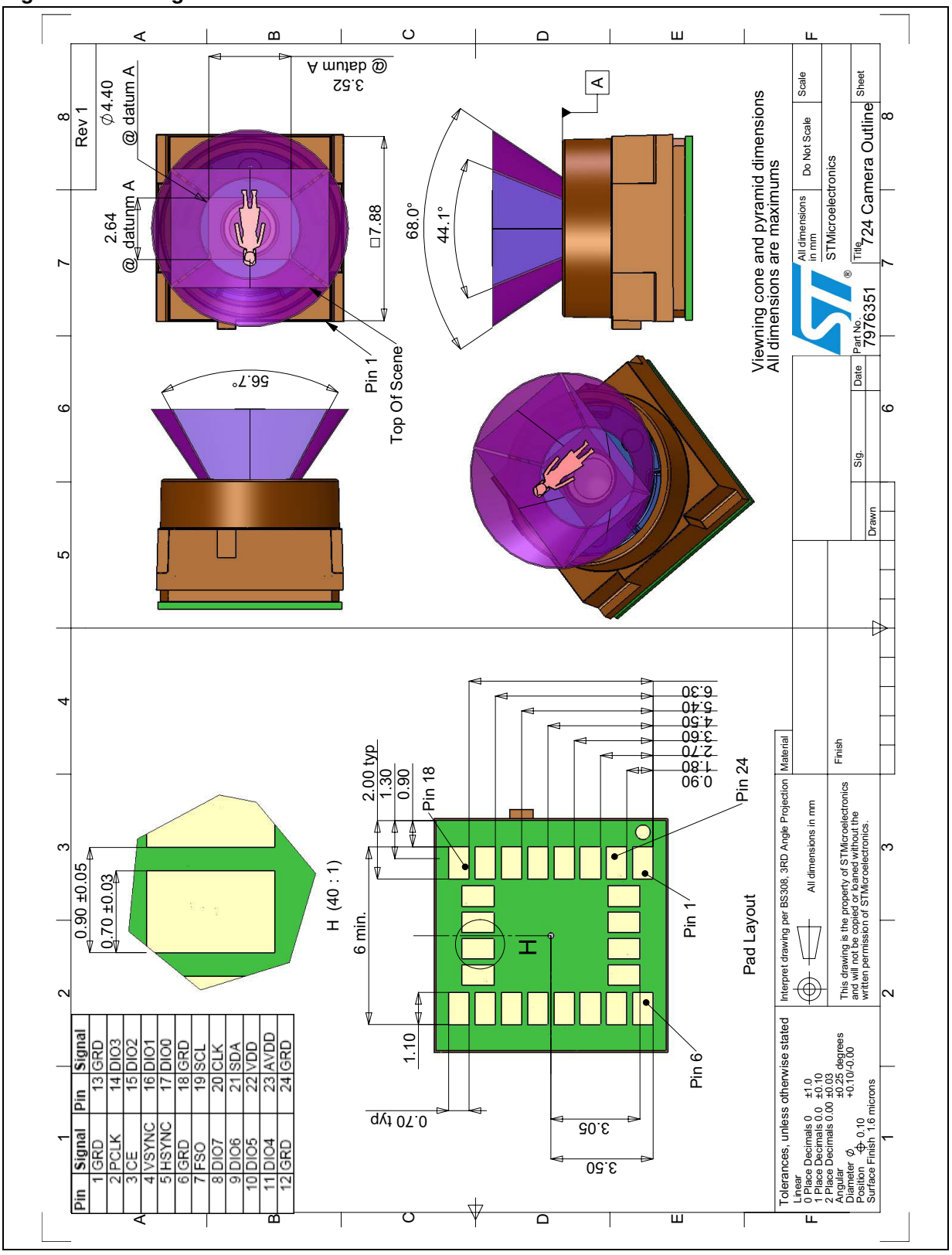


Figure 46. Package outline socket module VS6724



10 Ordering information

Table 63. Order codes

Part number	Package and packing
VS6724Q0FB	SmOP2 - Socket version

11 Revision history

Table 64. Document revision history

Date	Revision	Changes
03-Jul-2007	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZE REPRESENTATIVE OF ST, ST PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS, WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com