Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-14797

Spec Title: FREQUENTLY ASKED QUESTIONS ABOUT IBIS5 - AN6004

Sunset Owner: Evelyn Beard (EYB)

Replaced by: None

Frequenty Asked Questions about IBIS5

AN6004

OBSOLETE

The IBIS5 device is part of Cypress's industrial high-performance image sensor family. This application note lists the frequently asked questions (FAQs) by customers evaluating the IBIS5.

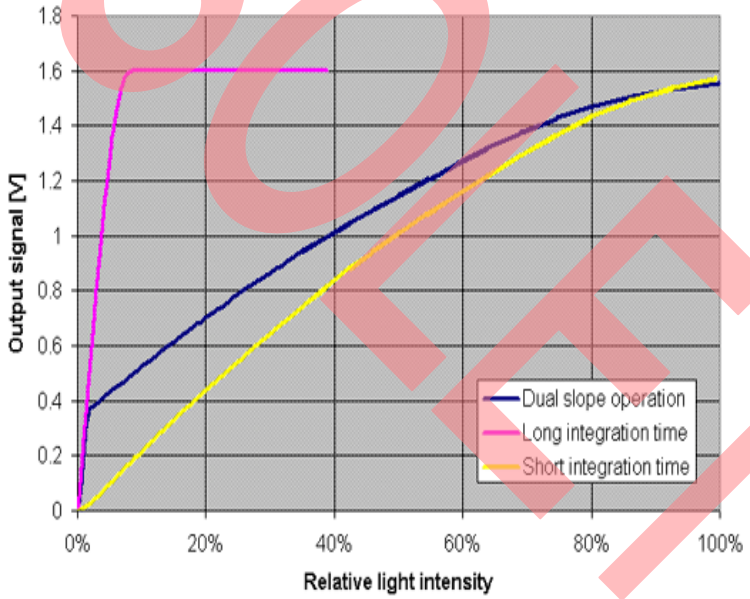
Separate application notes cover some of these topics in more detail.

1. How does the dual or multiple slope extended dynamic range mode work?

Dual slope is a method to extend the dynamic range of a normally linear-transfer imager, by combining the images taken with long integration time (dark areas of a scene) and short integration time (bright areas of a scene). The resulting electro-optical transfer curve is bi-linear.

Multiple slope is an extension of dual slope, resulting in a multi-linear transfer curve with multiple knee points. Figure 1 provides more details.

Figure 1. Multiple Slope Integration



2. How does one program signaling and windowing in X-and Y-direction?

The integrated timing and control circuit of IBIS5 allows windows of any size or region of interest (ROI) in any position within the pixel array to be read out. This speeds up the readout time with an increase in frame rate. Windowing is possible in both X- and Y-direction. Four Internal registers are required for windowing. The signaling and register settings are described in Table 1.

Figure 2. Windowing

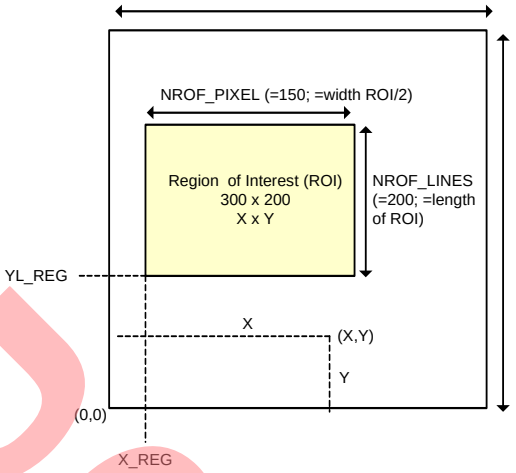


Table 1. Register Names and Comments

Register Name	Comment
NROF_PIXELS	Number of pixels to count in a row; equal to width of (ROI / 2) – 1
NROF_LINES	Number of lines to count; equal to length of ROI – 1
X_REG	X start position; equal to X / 2 (maximum = 639)
YL_REG	Y start position; equal to Y

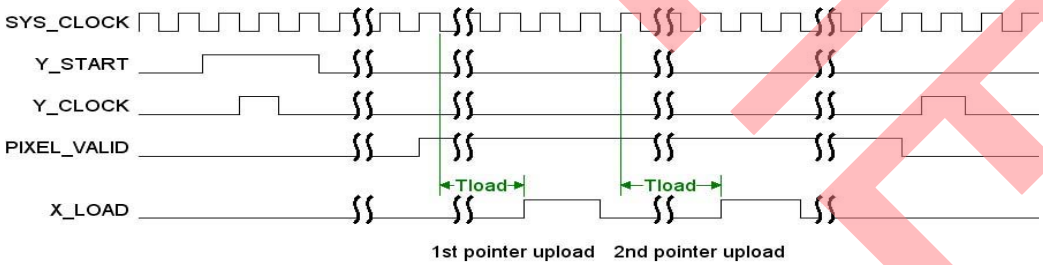
Note: YR_REG register = YL_REG register.

All registers have a minimum value of 0 and maximum value of ('endvalue'-1). For example, to get an ROI of 300 × 200 with starting points X = 500 and Y = 500, the following register values are required:

NROF_PIXELS: 149
NROF_LINES: 199
X_REG: 249
YL_REG / YR_REG: 499

If these registers are loaded, the correct ROI is read out and no special signaling need to be applied. It is also possible to make jumps in both X- and Y-direction during read out. Making an X-jump during read out of a row is described below.

Figure 3. Windowing in the X-direction



An X_LOAD pulse overrides the internal X_SYNC signal, loading a new X-pointer (stored in the X_REG register) into the X-shift-register. The X_LOAD pulse must appear on the falling edge of SYS_CLOCK and remain two SYS_CLOCK cycles high overlapping two rising edges of SYS_CLOCK. The new X-pointer is loaded on one of the rising edges of SYS_CLOCK .

T_{load} is defined from the register load to the rising edge of X_LOAD. It depends on the settling time of the register and the X-decoder. Simulations show that T_{load} should be longer than 1 s. Make sure that when X-jumps are required the X_REG register is loaded fast enough.

The actual time to load the register depends on the interface mode that is used. The parallel interface is the fastest interface available.

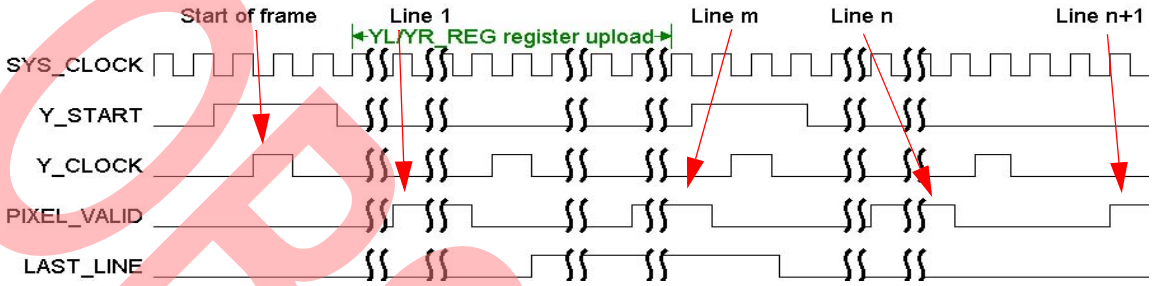
OBSOLETE

Table 2. Interface Modes and Loads

Interface Mode	T_{load} (s)
Parallel interface	1 (about 40 SYS_CLOCK cycles)
Serial 3 Wire	200 (at 200 kHz data rate)

Y-jumps are done in a similar way as described below.

Figure 4. Windowing in the Y-direction



A new Y-pointer can by loaded into the Y-shift-register, by reapplying the Y_START pulse after loading a new Y-pointer value into the YL_REG and YR_REG registers. Every time a Y_START pulse appears, a frame calibration (slow or fast depending on the selected mode) of the output amplifier is done.

3. What is the difference between the YL_REG and YR_REG registers? Should they both be loaded with the same value?

YL_REG is the start position of the left Y-pointer; YR_REG is the start position of the right Y-pointer. In almost all cases both registers should be loaded with the same value (start Y-value of each frame). Only in rolling shutter mode, it is useful to set different values for both registers.

4. What is the shortest integration time that can be programmed?

When SS_START is asserted, it takes four granulated clock cycles to apply all control signals to the sequencer, reset the image core, and start integration. During the four initial clock cycles, the INT_TIME register starts counting. This means that the minimal integration time is set when INT_TIME is set to 5 (INT_TIME always ≥ 5). This minimal integration time is one granulated SS-sequencer clock cycle.

Granularity MSB	Granularity LSB	Minimum Integration Time
0	0	800 ns
0	1	1.6 µs
1	0	3.2 µs
1	1	6.4 µs

The minimal integration time programmable in rolling shutter mode is the read out time of a row.

Integration time = (25 ns × (#pixels/row)) + 3.5 µs = 34.5 µs (for a row of 1240 pixels)

5. Can SYS_CLOCK be clocked above 40 MHz?

The IBIS5 is tested at 40 MHz, but the device has been verified to be functional with acceptable image quality at 58 MHz. As expected, noise figures will increase with frequency and a nonlinear response of the grey values occur at these high frequencies.

6. Is there FPN correction on-chip?

The IBIS5 image sensor provides on-chip double sampling FPN correction.

7. What is the need for two analog outputs (PXL_OUT1 and PXL_OUT2)?

Two outputs were initially designed for debugging reasons. It is possible to use both the outputs to send even and odd pixels (or lines). There is no increase in speed gain when using both outputs as both outputs run at half the output rate.

OBSOLETE

8. Is there any pipeline delay?

There is a pipeline delay of seven clock cycles at the digital output (ADC_OUT<9…0>). This delay is due to a delay of five clock cycles in the analog part of the image sensor and a delay of two clock cycles due to the ADC.

Figure 5. Pipeline Delay

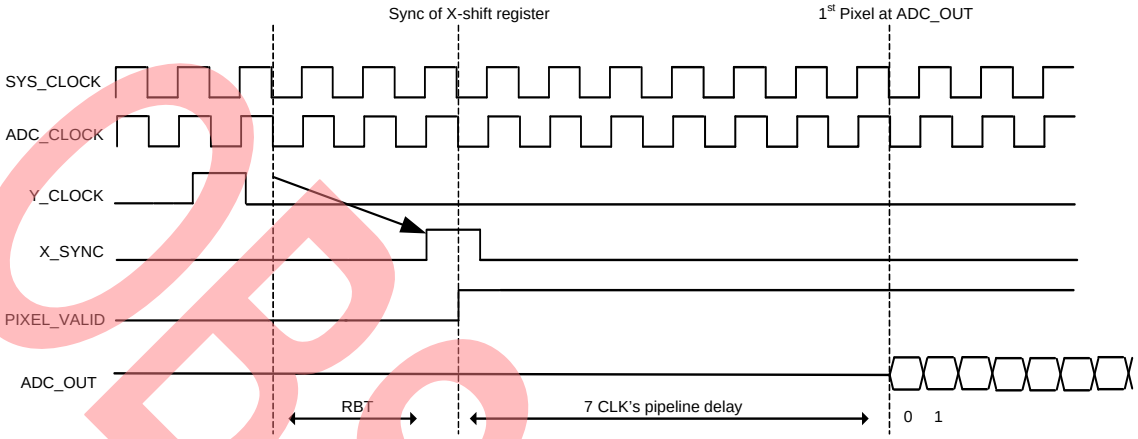
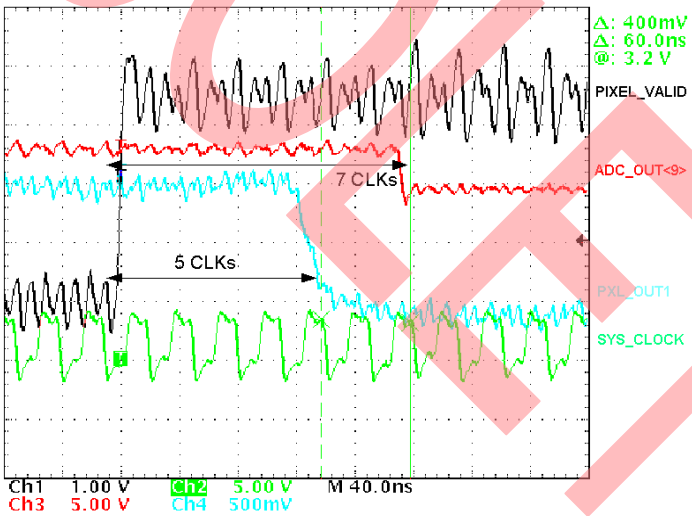


Figure 5 shows that it takes seven clock cycles after the rising edge of PIXEL_VALID to get the output word of the first pixel at ADC_OUT<9…0>. Figure 6 shows this delay measured on an oscilloscope (the analog output swing is not the full output swing).

Figure 6. Pipeline Delay on Oscilloscope



OBSOLETE

9. Is it possible to connect all supplies with +3.3 V?

The IBIS5 sensor is functional when all supplies are connected to +3.3 V (VDDD, VDDA, VDDC, VDDH, VDDR_LEFT, and VDDR_RIGHT). The output swing and SNR will decrease. Recommended supply considerations are shown in Table 3.

Table 3. Recommended Supplies

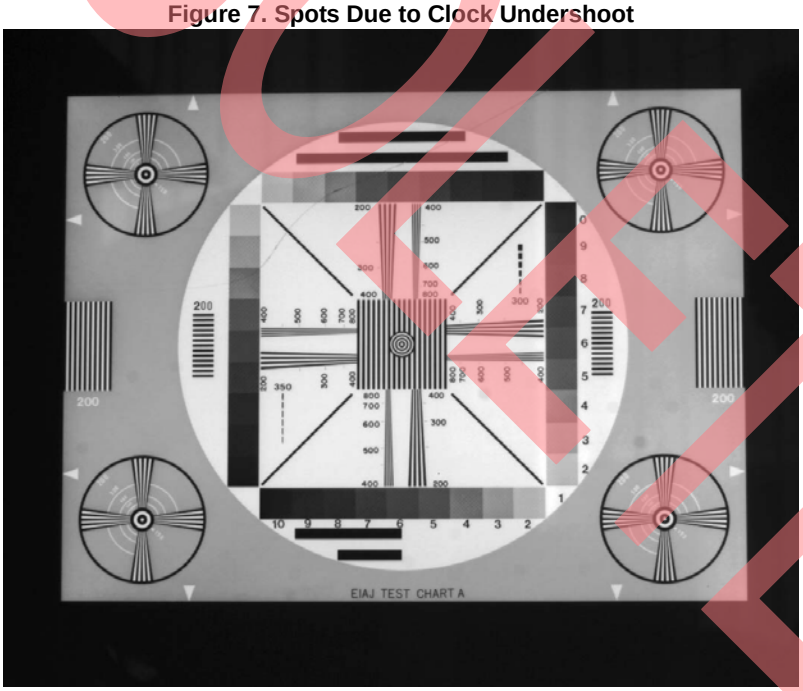
Name	Value
VDDH	4.5 V
VDDR_LEFT	4.5 V
VDDC	3.0 V
VDDD	3.3 V
VDDA	3.3 V
GNDD	0.0 V
GNDA	0.0 V
GND_AB	<0.5 V

10. What is the cause of white spots/areas on the bottom of the images from IBIS5-A(E)?

The white spots on the bottom of the images, as seen in Figure 7, are due to undershoot on the ADC_CLOCK input (pin 58). If the undershoot on this signal is below –0.6 V, the ESD protection diode will be forward biased and current is injected into the substrate of the chip. These free electrons are captured by the photo diodes in the pin area. The AE devices suffer more from this because they have a thicker epi-layer, which collects more electrons generated deep into the substrate.

The solution is to add a small capacitor (47 pF for instance) to remove any undershoot on the ADC_CLOCK input signal.

Figure 7. Spots Due to Clock Undershoot



OBSOLETE

11. Why do images tend to saturate towards the bottom and what is PLS?

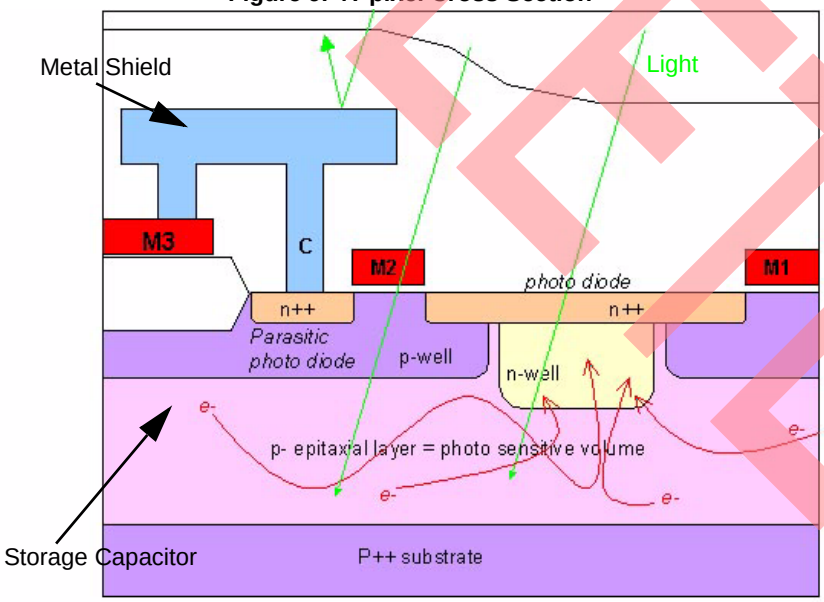
This effect is called parasitic light sensitivity (PLS) and is only apparent in global shutter mode. PLS is seen as a gradient in the image, see Figure 8.

Figure 8. PLS Gradient



PLS is the sensitivity of the storage capacitor, which holds the voltage value of the photodiode, within a pixel to light after the integration time but before the data is read out. The image rows are read out from top to bottom; therefore, the storage capacitors of the bottom rows are more exposed to light than the top capacitors and tend to saturate. To cancel this effect, the image sensor should be shielded from light after the integration time. There is a metal shield on top of the storage capacitor to minimize this effect. Figure 9 is a cross section of the 4T-pixel.

Figure 9. 4T-pixel Cross Section



12. What causes line irregularities (horizontally) that change from frame to frame?

This artifact is possibly caused by noise that is present on the supply voltages. Because the IBIS5 does not have on-chip power supply rejection (PSRR), all noise from power supplies are transferred to the analog pixel signal. As the pixels are sampled line by line it will be visible as line noise.

OBSOLETE

It is recommended to use a low noise, low dropout linear regulator over a switching regulator. Ensure that every sensor supply pin is decoupled with 100 nF as close as possible to the sensor (preferably < 1 cm).

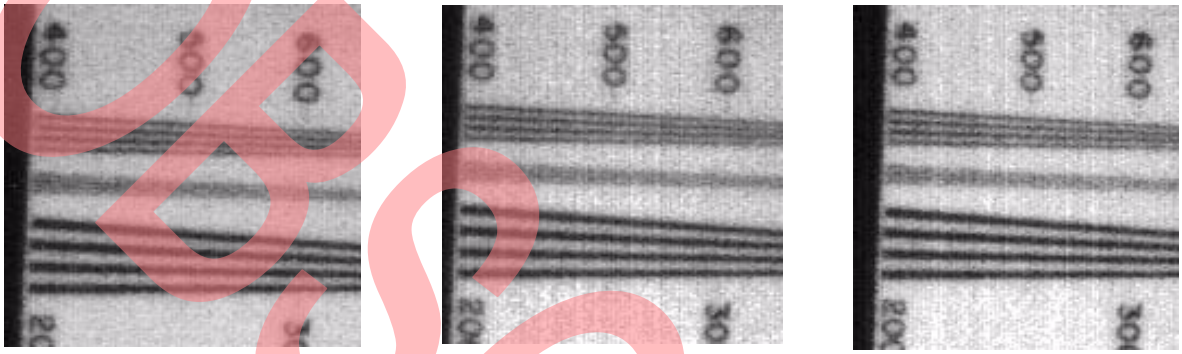
13. Which parameters can be uploaded safely during operation?

Only parameters involving start positions of ROI (X_REG and YL_REG) can be uploaded during operation. Other parameters should be uploaded after the whole integration and readout sequence is completed to avoid any sequencer problems.

14. There are column non-uniformities. Is this normal?

If the Y- read out pointer is in the process of selecting a line during the global reset of the next frame, the reset of columns is corrupted. The corrupted image is visible after integration. This artifact results in column non-uniformities.

Figure 10. Test Chart Images with (center, right) and without (left) Column Non-uniformities

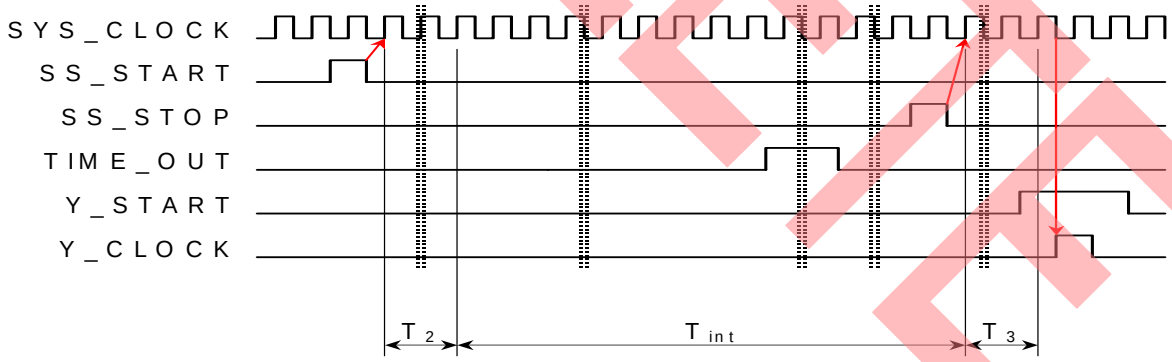


On the left image, the Y- read out pointer is completely shifted out of the pixel array so that the pointer does not select a line during global reset, which results in no column non-uniformities. In the middle and right images the Y- read out pointer is selecting a line during global reset, which introduces column non-uniformities.

Column non-uniformities as shown in Figure 10 can also be introduced by different intervals between successive images. The following recommended timing also solves this non-uniformity.

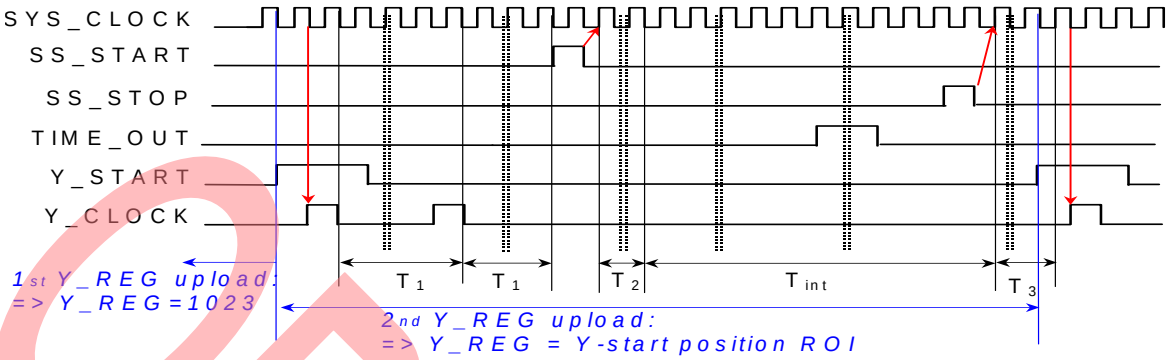
To make sure that the Y- read out pointer does not select a line during global reset, an adapted timing in synchronous shutter mode (single slope mode) is recommended. Figure 11 shows the standard timing as explained in the data sheets of IBIS5-1300. Figure 12 shows the adapted timing.

Figure 11. Standard SS Single Slope Integration Timing



OBSOLETE

Figure 12. Adapted Timing



T_1 Should be minimal 1 row blanking time (the row blanking time depends on the X-granularity; refer to product data sheet: Basic operation timing).

T_2 During this time, the SS-sequencer applies the control signals to reset the image core and start integration. This takes four granulated SS-sequencer clock periods. The integration time counter starts counting at the first rising edge after the falling edge of SYS_CLOCK.

T_3 The SS-sequencer puts the image core in a readable state. It takes two granulated SS-sequencer clock periods.

T_{int} The real integration or exposure time.

To make sure that no line is selected during the global reset, value 1023 (decimal value; hexadecimal value 3FF) must be uploaded to the YL_REG and YR_REG registers before the assertion of the first Y_START. This selects the last line when Y_START is asserted with Y_CLOCK; pulsing the second Y_CLOCK shifts the Y- read out pointer out of the shift register before the global reset occurs. Between the falling edge of the first Y_START and the rising edge of the SS_STOP pulse, upload the YL(R)_REG registers (registers 5 and 6) with the start row of the ROI.

Note that the additional Y_START sequence adds a delay between a trigger event and the actual start of integration. This delay is equal to twice the row blanking time (2 × 3.5 µs = 7 µs with minimal X-granularity). Figure 12 indicates time T_2, which takes another four SS granular clock periods after SS_START before the pixels actual start to integrate.

15. Why does the TIME_OUT signal appear even when SS_START is not asserted?

The TIME_OUT signal is generated by a counter, which is driven by the system clock. The counter asserts the TIME_OUT signal whenever the value of INT_TIME is reached. SS_START only sets the counter to '0'. In other words, if you apply the system clock, the TIME_OUT signal appears when the counter reaches the value of INT_TIME.

16. What is the cause of the banding in the image in rolling shutter mode?

The banding shown in Figure 13 is caused when vertical blanking is used to increase the integration time (delaying the assertion of Y_START). When the left Y-shift register points to the end of the image array, a Y_START pulse resets this shift register. If Y_START is not asserted, the left Y-shift register is clocked out of the array while the right Y-shift register keeps resetting new lines. This causes a reset voltage offset in the pixel. Both Y-shift registers should always point in the array to eliminate this effect. When integration times longer than the frame readout time are required, apply horizontal blanking; this means, program additional delay times at the end of each row. It is important that the Y_START and Y_CLOCK pulses are always asserted with the same interval.

OBSOLETE

Figure 13. Banding Caused by Vertical Blanking



17. **Is there a VHDL test bench available for the IBIS5 sequencer?**

    A VHDL test bench for the IBIS5 sequencer is available on demand.

18. **Are there any design guides for the IBIS5-B PCB?**

    The schematics of our evaluation kits can be used as a reference design.

19. **What is the fill factor of the IBIS5 and how is it improved?**

    The fill factor is approximately 50 percent and this is achieved by use of our patented n-well pixel technology. This technology turns most of the chip's silicon into a light-sensitive area.

20. **Is there a demo system available and at what cost?**

    Demo systems will not be available for sale after Q1, 2011. However, they can be rented for evaluation. Contact imagesensors@cypress.com for more details.

OBSOLETE

Document History Page

Document Title: Frequently Asked Questions about IBIS5
Document Number: 001-14797

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	988843	RGL	04/20/2007	New application note
*A	1770467	VDS	12/03/2007	Converted to new template
*B	3110144	NVEA	12/13/2010	Template updates. Edited Figure 2, Figure 3, Figure 4, and Figure 5.
*C	4113912	MTA	09/04/13	Obsolete specs.

OBSOLETE

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com

© Cypress Semiconductor Corporation, 2007-2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

All product and company names mentioned in this document may be trademarks of their respective holders.

OBSOLETE