1/3-Inch CMOS Digital Image Sensor

AR0330 Developer Guide Rev. E

For the latest data sheet, refer to Aptina's Web site: www.aptina.com

AR0330
Developer Guide

Table of Contents

List of Figures

List of Tables

Introduction

This Developer Guide provides instructions towards how to configure the sensor PLL, output interface. (parallel, HiSPi, or MIPI), the analog gain, and the GRR mode. This document also describes how to identify the version of the on-chip one-time programmable memory (OTPM) that has been programmed to the sensor during production.

Note: This document references the AR0330 data sheet, errata, and register reference.

Background

The functionality of the AR0330 1/3-Inch CMOS digital image sensor can be configured through register settings. One set of these registers is made available to the user to configure the sensor to their use case. Another set of confidential reserved registers and an instruction set for the sensor sequencer are also configured by Aptina in order to optimize the sensor performance.

Sensor Configuration

The sensor configuration discussed in this tech-note can be grouped into the following sections:

1. Sensor PLL and interface configuration (parallel, MIPI, or HiSPi)
2. Sensor Readout Modes

Other registers used for features such as the global reset used for image snapshots, gain and exposure control, and the sensor slave-mode are not described in this technical note but are referenced in the sensor data sheet and register reference.

Reserved Registers

Aptina stores the required reserved register writes into OTPM during the sensor production test. The sensor will then configure the reserved registers from the OTPM when the sensor is reset or powered-on. This process allows Aptina to better control the registers that are required to be written to the sensor.

It is important that sensor drivers are written so that they will check the described registers that will indicate the sensor OTPM version. Based on the OTPM version, the sensor driver should only program the reserved registers noted in the current version of this document. The sensor driver should also be written so that it will check for future versions of the sensor OTPM.

Warning! Overwriting reserved registers programmed by the OTPM may lead to reduced performance in the AR0330 CMOS image sensor.

Sensor Configuration

The sections below describe specific topics towards programming the AR0330 CMOS image sensor. The configuration files (AR0330-REV1.ini, AR0330-REV2.ini) provided with the Aptina DevSuite can be used as a reference for configuring the sensor.

Note: Some sections relevant to sensor programming are described in the data sheet.

1. Changing Sensor Modes (reference data sheet section: "Changing Sensor Modes")
2. Configure Sensor Modes (reference data sheet sections: "Sensor Readout", "Subsampling", and "Sensor Frame Rate")
3. Integration and Gain control (reference data sheet sections: "Pixel Sensitivity" and "Gain Stages")

Sensor Default Configuration

The sensor registers are programmed to the following default configuration:

- The sensor interface is configured to use the HiSPi interface in the Streaming S protocol mode.
- The PLL is configured to a readout clock (CLK_PIX) of 98 Mpixel/s based on a 24MHz input clock (EXTCLK). The output data rate per lane with the 24 MHz input clock will be 588 Mbps/lane at 12-bit per pixel.
- The sensor mode in context A is 2304x1296 at 59.94 fps and 2304x1536 at 50.72 fps in context B.

PLL Configuration

The sensor PLL and output interface should be configured when the sensor streaming is disabled. Reference the following sections when programming the sensor:

1. Programming the sensor PLL. (reference data sheet section: "Sensor PLL")
2. Program the sensor output interface.

Interface Configuration

This subsection describes the registers that must be programmed to enable each interface.

[A] Enabling the Parallel or Serial Interface

Reference the register R0x301A described in the sensor register reference document.

- Enable the parallel interface by setting "parallel_en" (R0x301A[7]) to "1" and "smia_serial_dis" (R0x31AE[12]) to "0". R0x31AE must also be set to 0x301.
- Enable the serial (MIPI or HiSPi) interface by setting "parallel_en" (R0x301A[7]) to "0" and "smia_serial_dis" (R0x301A[12]) to "1".

[B] Configure the sensor to use HiSPi or MIPI and the Number of Data-Lanes

Reference the "Serial Configuration" and "Compression" section in the data sheet.

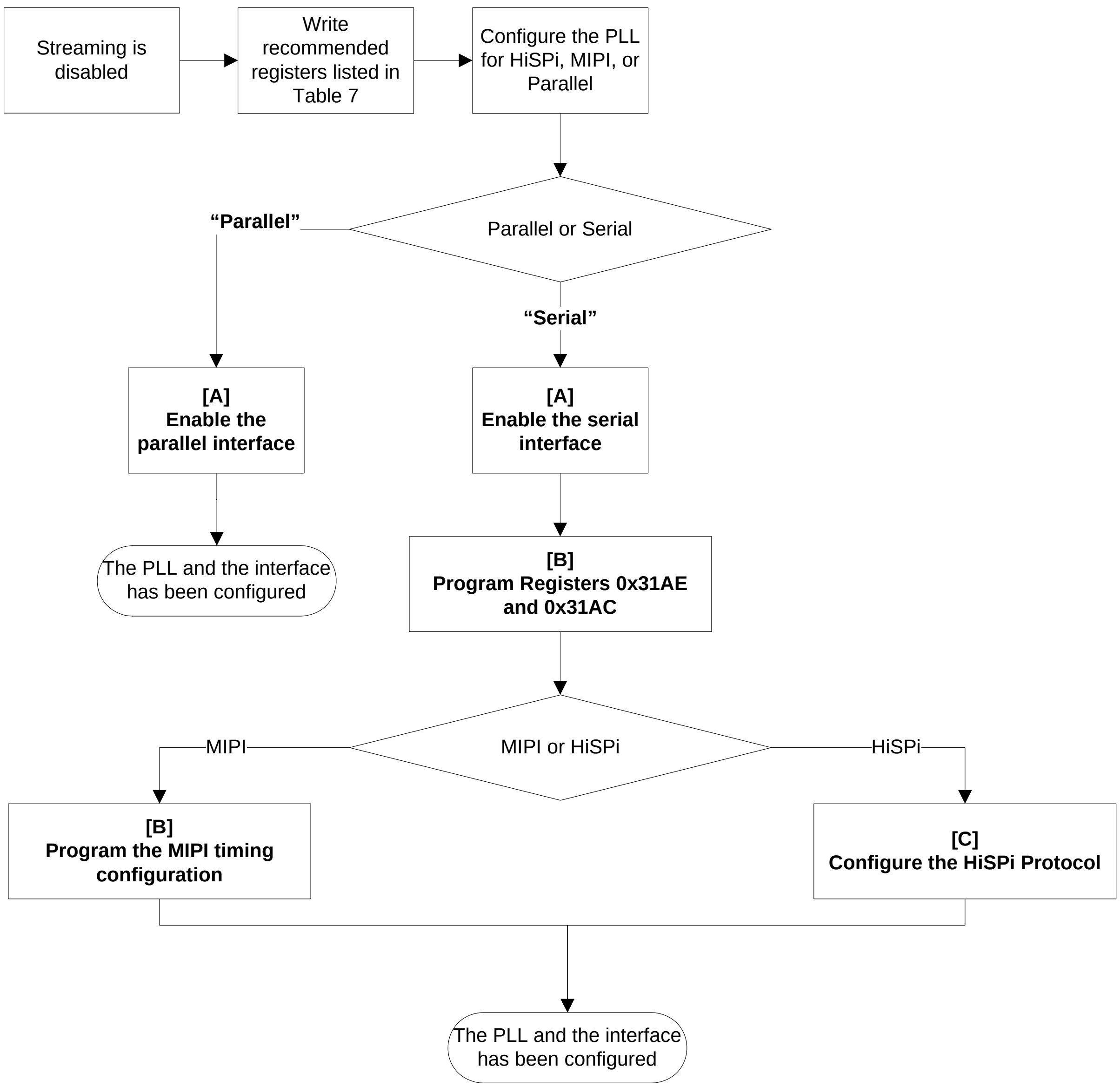
- Program R0x31AE to configure the sensor to use HiSPi or MIPI as well as the number of data lanes. (reference data sheet section: "Serial Configuration")
- Program R0x31AC to configure the expected data format identifying the data bit-depth and the possible use of the A-law compression (reference data sheet section: "Serial Configuration and Compression").
- Program the MIPI Timing configuration if using the MIPI interface.

[C] Configure the HiSPi Output Interface

The HiSPi output interface configuration is described in register R0x31C6 in the sensor register reference document. Further information can be found in the "HiSPi Protocol Specification V2.0."

- Configure the HiSPi protocol using register R0x31C6.

Figure 1: PLL and Interface Configuration



Gain Tables and ISO Speed

The recommended gain configuration per ISO speed is listed in Table 1.

Table 1: ISO Gain Table

ISO Speed	Necessary Gain	ISO Speed	Necessary Gain
ISO 157	1.00x	ISO 800	5.40x
ISO 200	1.30x	ISO 1600	10.90x
ISO 400	2.70x	ISO 3200	21.80x

A recommended gain table that uses both the sensor coarse and fine analog gain steps are listed in the table below.

Table 2: Recommended Sensor Analog Gain Tables

COARSE_GAIN		FINE_GAIN		Total Gain		COARSE_GAIN		FINE_GAIN		Total Gain	
Reg Value	Gain (x)	Reg Value	Gain (x)	(x)	(dB)	Reg Value	Gain (x)	Reg Value	Gain (x)	(x)	(dB)
0	1	0	1.00	1.00	0.00	0	1x	15	1.88	1.88	5.49
0	1	1	1.03	1.03	0.26	1	2x	0	1.00	2.00	6.00
0	1	2	1.07	1.07	0.56	1	2x	2	1.07	2.13	6.58
0	1	3	1.10	1.10	0.86	1	2x	4	1.14	2.29	7.18
0	1	4	1.14	1.14	1.16	1	2x	6	1.23	2.46	7.82
0	1	5	1.19	1.19	1.46	1	2x	8	1.33	2.67	8.52
0	1	6	1.23	1.23	1.80	1	2x	10	1.45	2.91	9.28
0	1	7	1.28	1.28	2.14	1	2x	12	1.60	3.20	10.10
0	1	8	1.33	1.33	2.50	1	2x	14	1.78	3.56	11.02
0	1	9	1.39	1.39	2.87	2	4x	0	1.00	4.00	12.00
0	1	10	1.45	1.45	3.25	2	4x	4	1.14	4.57	13.20
0	1	11	1.52	1.52	3.66	2	4x	8	1.33	5.33	14.54
0	1	12	1.60	1.60	4.08	2	4x	12	1.60	6.40	16.12
0	1	13	1.68	1.68	4.53	3	8x	0	1.00	8.00	18.00
0	1	14	1.78	1.78	5.00						

Sequencer Programming

The sequencer digital block determines the order and timing of operations required to sample pixel data from the array. During sensor initialization, the sequencer will be programmed into memory from the sensor OTPM (one-time programmable memory). The OTPM is configured during production.

If the sequencer needs to be programmed differently than its automatically initialized setting, then the sequencer digital block can be reprogrammed using the following instructions:

Program a new sequencer:

1. Place the sensor in standby.
2. Write 0x8000 to R0x3088 (seq_ctrl_port).
3. Write each instruction incrementally to R0x3086. Each write must be 16-bit consisting of two bytes {Byte[N], Byte[N+1]}.
4. If the sequencer consists of an odd number of bytes, set the last byte to “0”.

Read the instructions stored in the sequencer.

1. Place the sensor in standby.
2. Write 0xC000 to R0x3088 (seq_ctrl_port).
3. Sequentially read 2-bytes at a time from R0x3086.

Sequencer Update and Frame Rate Calculations

The sequencer digital block of the AR0330 determines the order and timing of operations required to sample pixel data from the array. Therefore, changes to the sequencer will affect the ADC Readout Limitation (sample) clock periods used in the row period (T_{ROW}) calculation.

The sequencer "B" currently required for the AR0330 Rev 2 sensor uses 1242 clocks. This length will not affect the maximum frame-rate of modes reading the full width of the sensor without subsampling (for example, 1080p+EIS, 3M Still) as the Digital Readout and Output Interface limitations will determine the row period.

Refer to the "Row Period" section of the AR0330 Data Sheet for an explanation of the T_{ROW} limitations listed in Figure 2 and Figure 3 on page 10. The programmed sequence that has a default length of 1242 clocks should be used unless the desired mode requires a shorter ADC readout limitation as described below.

Shortening ADC Readout Limitation

A modification to the sequencer is available to enable the maximum frame-rate for high speed video modes such as the WVGA + EIS Slow-motion (1152x648 at 120fps) where the ADC Readout limitation will determine the row period. Shortening the ADC Readout limitation will enable higher frame-rates but may result in black level shading and column fixed pattern noise. This procedure is valid only for sequencer "B" (Rev2 sensors only).

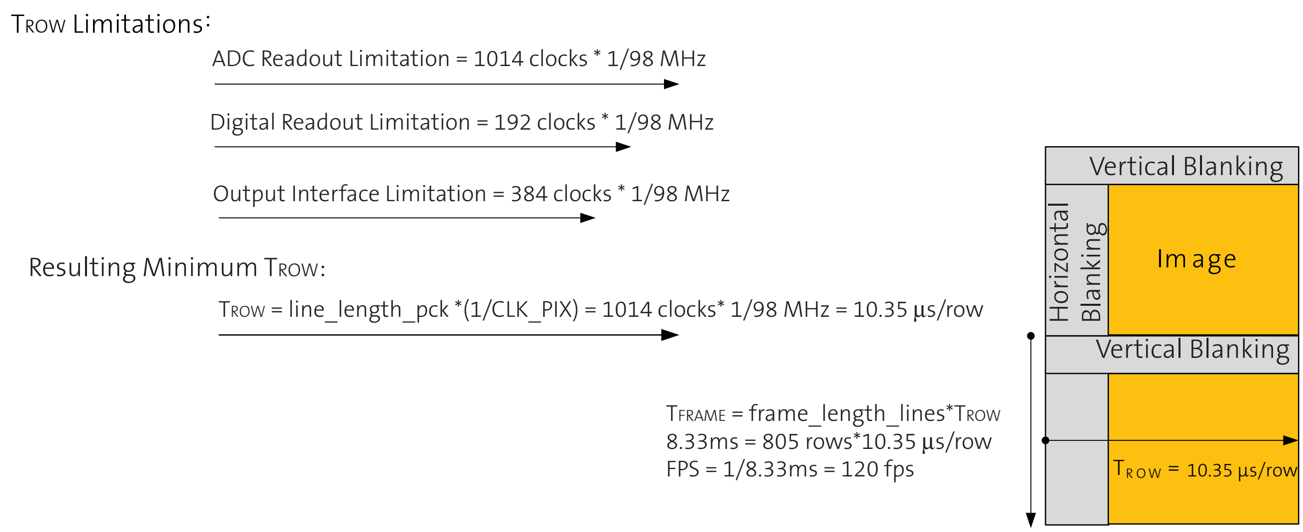
The steps required to change the ADC Readout limitation is the following:

1. The sensor must be in standby (R0x301A[2] = 0).
2. Set the sequencer control port (seq_ctrl_port) to the address of the instruction that must be modified.
3. Update the address instruction.
4. Enable streaming.

Example: Shortening the sequencer to 1014 clocks by adjusting the length of a delay instruction:

1. The sensor has been placed in standby.
2. Write to address 0xBA in the sequencer. R0x3088 = 0x80BA
3. Change the instruction at address 0xBA. R0x3086 = 0x0253

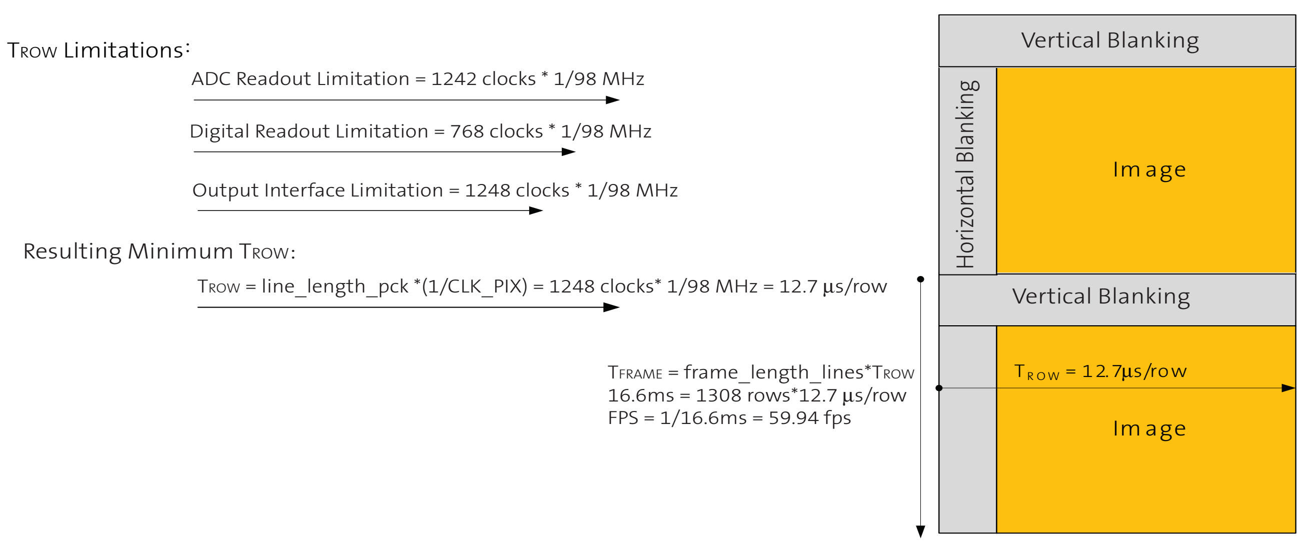
Figure 2: Sequencer Shortened to 1014 Clocks Enabling the 1152x648 at120fps Mode



Example: Returning the sequencer to 1242 clocks by adjusting the same delay instruction:

1. The sensor has been placed in standby.
2. Write to address 0xBA in the sequencer. R0x3088 = 0x80BA
3. Change the instruction at address 0xBA. R0x3086 = 0xE653

Figure 3: Sequencer Lengthened to 1242 Clocks for the 2304x1296 at 60fps Mode



GRR MODE

Entering and Exiting the GRR mode

The sensor must be placed in Standby (R0x301A[2]=0) before the GRR can be called. The GRR can be called immediately after the sensor is placed in the standby state.

The GRR frame can be started using two methods:

1. Configure the GRR_MODE (R0x30CE[0]=1) to TRUE and then enable streaming (R0x301A[2]=1).
2. Configure the GPI_EN (R0x301A[8]=1) so that the TRIGGER pin is enabled. Transition the input signal into the TRIGGER pin from low to high.

The sensor must be placed back into standby before calling another GRR frame.

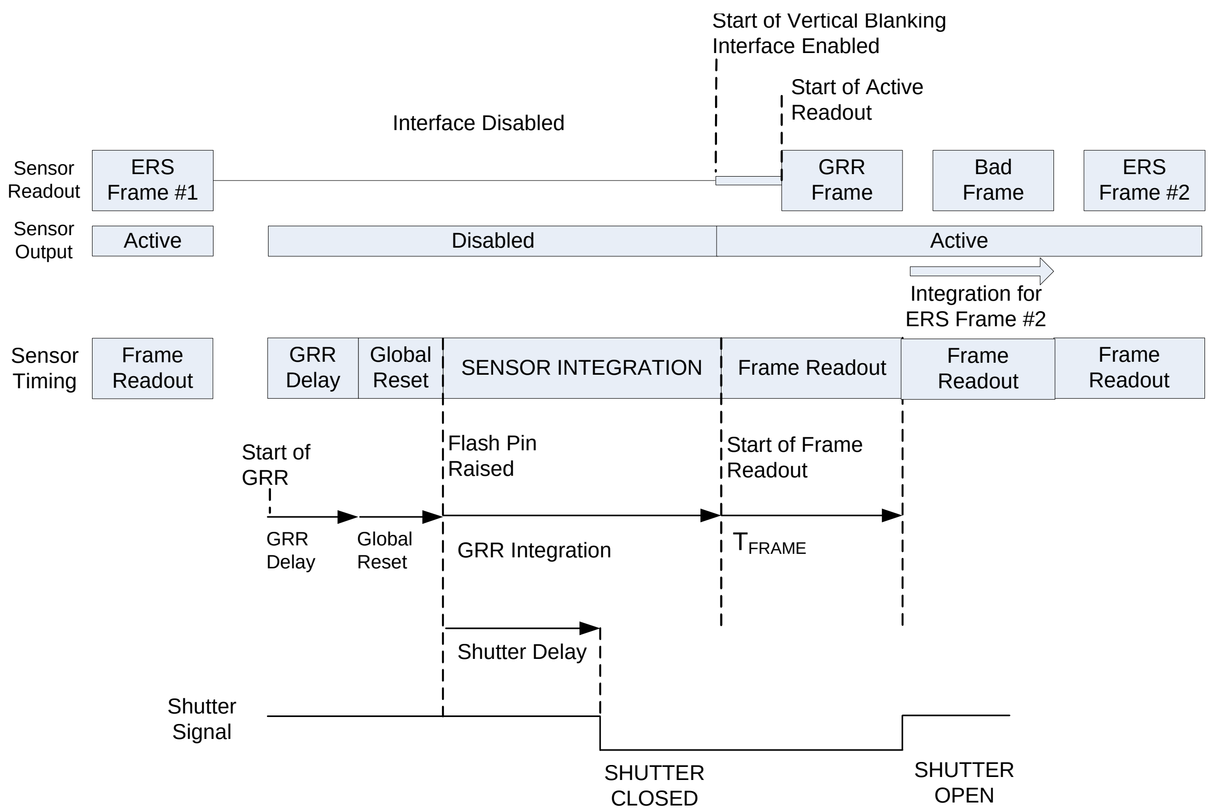
The sensor can be configured to ERS mode by setting GRR_MODE (R0x30CE[0]=0) to FALSE. If this is configured while streaming is enabled (that is, during a GRR frame readout) then an incorrectly exposed frame (a bad frame) will be output after the GRR frame and before the first properly exposed ERS frame.

Table 3: Shutter Control During GRR

GRR Shutter Controls	Description
Enable/Disable Shutter	The shutter GPIO can be configured to be “always open” (shutter_always_open=1) or “always closed” (shutter_disable=1).
Shutter Pulse Mode	When ext_shut_pulsed is TRUE, the shutter pin remains low until the end of the shutter delay where it will pulse HIGH. The pulse width can be set by the variable ext_shut_pulse_width that is measured in number of clock cycles. If ext_shut_pulsed is FALSE, the shutter pin will toggle HIGH when GRR_MODE is set to HIGH. It will toggle to LOW at the end of the shutter delay.

Timing Control for GRR

Figure 4: AR0330 GRR Snapshot Timing



Note: GRR_MODE must be configured to 0 during the GRR frame integration or readout for the sensor to enter the ERS mode after the GRR frame.

Table 4: GRR Timing Control

GRR Timing Parameter	Definition
GRR Delay	Delay between external trigger and global reset in number of rows controlled by gr_delay.
Global Reset	The sensor will reset all pixels in the array during the global reset period. This will last 13 row periods (13 x T_{ROW}).
GRR Integration	GRR Integration is defined as the period following the GRR reset when the pixels are integrating and the mechanical shutter is open. The length of this period is defined in row periods: $T_{COARSE\_INTEGRATION}$ = coarse_integration_time x T_{ROW}
Shutter Delay	The shutter delay period determines length of the GRR Integration period in which the mechanical shutter is open. The length of this period is defined in row periods: $T_{SHUTTER\_DELAY}$ = ext_shut_delay x T_{ROW}
Frame Period	The frame period is defined in the data sheet under T_{FRAME}=1/(CLK_PIX) ×[frame_length_lines × line_length_pck + extra_delay]

Note: TROW is defined as "=line_length_pck/CLK_PIX" where line_length_pck is the number of clocks per row and CLK_PIX is the readout clock frequency.

One-Time Programmable Memory

The AR0330 CMOS sensor includes internal one-time programmable memory (OTPM). The OTPM is used to store register addresses and recommended values. During the sensor initialization sequence, the sensor will reference the register addresses listed in the OTPM and program these registers with the register values listed in the OTPM.

The use of the OTPM data helps to reduce the number of register writes required by the user when initializing the Aptina sensor.

Table 5: OTPM Data Programmed During Production

OTPM Data	Use of OTPM Data During the Initialization Sequence:
The instruction-set used in the sequencer digital block.	This instruction set is programmed into the AR0330 sequencer during the initialization sequence.
Calibration data specific to the sensor for pixel sampling and analog readout determined during the production tests.	This calibration data is programmed to “reserved” sensor registers.
Sensor revision and register bits that identify the CRA revision.	This data is programmed to register 0x31FA. (reference data sheet section: Spectral Characteristics)
Changes to the sensor default register configuration.	The register addresses and values stored in OTPM are written to reserved sensor registers.

Table 6: OTPM Version Table

OTPM Version	[R0x300E]	[R0x30F0]	[R0x3072]	Required Actions:
Version #1 (AR0330 Rev1)	0x10	0x1200	0x0000	Program Recommendations V1 (see Table 7)
Version #2 (AR0330 Rev 2.0)	-	0x1208	0x0000	Program Recommendations V2 (see Table 7)
Version #3 (AR0330 Rev 2.1)	0x20	0x1208	0x0006	Program Recommendations V3 (see Table 7)
Version #4 (AR0330 Rev 2.1)	0x20	0x1208	0x0007	Program Recommendations V4 (see Table 7)
Version #5 (AR0330 Rev 2.1)	0x20	0x1208	0x0008	Program Recommendations V5 (see Table 7)

Note: The original AR0330 Rev2.0 samples did not have R0x300E programmed correctly.

Figure 5: Sensor Configuration Depending on OTPM Version

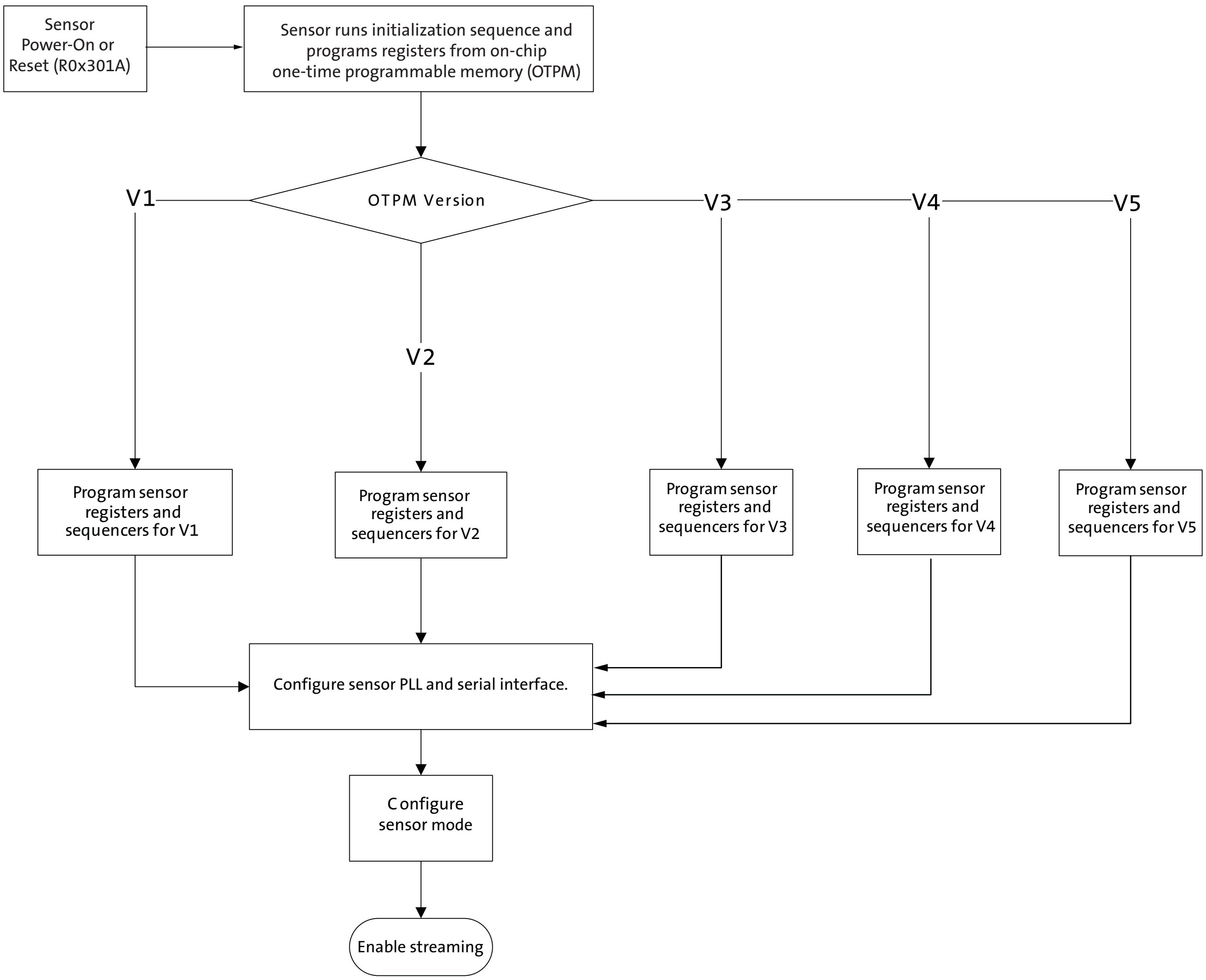


Table 7: Recommended Default Register Changes and Sequencer
Dependent on the Sensor OTPM

	AR0330 Rev 1	AR0330 Rev 2.0	AR0330 Rev 2.1	AR0330 Rev 2.1	AR0330 Rev 2.1
OTPM Version	**V1**	**V2**	**V3**	**V4**	**V5**
R0x30BA	0x2C	0x2C	-	-	-
R0x30FE	0x0080	0x0080	-	-	-
R0x31E0	0x0003	0x0003	0x0003	0x0003	-
R0x3ECE[0:7]	0xFF	0xFF	-	-	-
R0x3ED0	0xE4F6	0xE4F6	-	-	-
R0x3ED2	0x0146	0x0146	0x0146	0x0146	0x0146
R0x3ED4	0x8F6C	0x8F6C	0x8F6C	-	-
R0x3ED6	0x66CC	0x66CC	0x66CC	0x66CC	-
R0x3ED8	0x8C42	0x8C42	0x8C42	0x8C42	-
R0x3EDA	0x88BC	0x88BC	0x88BC	0x88BC	0x88BC
R0x3EDC	0xAA63	0xAA63	0xAA63	0xAA63	0xAA63
R0x3EDE	0x22C0	0xAA04	-	-	-
R0x3EE0	0x1500	0x15F0	-	-	-
R0x3EE6	0x0080	0x008C	-	-	-
R0x3EE8	0x2027	0x2024	-	-	-
R0x3EEA	0x001D	0xFF1F	-	-	-
R0x3F06	0x046A	0x046A	-	-	-
R0x305E	0x00A0	0x00A0	0x00A0	0x00A0	0x00A0
Sequencer Version	Sequencer A	Sequencer B	Sequencer Patch 1	-	-

Note: The symbol "-"indicates that the register value or sequencer is programmed by the sensor OTPM or that its power-on default value is correct; it should not be overwritten.

Table 8: "Sequencer A" Recommended for the AR0330 Rev1 Sensor

Writes 1-20	Writes 21-40	Writes 41-60	Writes 61-80	Writes 81-100	Writes 101-120	Writes 121-140	Writes 141-148
0x4540	0xFF3D	0x0D60	0x294E	0x285D	0x035C	0x2798	0x4017
0x6134	0xFF3D	0x5754	0x1718	0x27FA	0x0045	0x4B12	0x1027
0x4A31	0xEA62	0x1709	0x26A2	0x170E	0x4027	0x4452	0x9817
0x4342	0x2728	0x5556	0x5C03	0x2681	0x9017	0x5117	0x2022
0x4560	0x3627	0x4917	0x1744	0x5300	0x2A4A	0x0260	0x4B12
0x2714	0x083D	0x145C	0x2809	0x17E6	0x0A43	0x184A	0x442C
0x3DFF	0x6444	0x0945	0x27F2	0x5302	0x160B	0x0343	0x2C2C
0x3DFF	0x2C2C	0x0045	0x1714	0x1710	0x4327	0x1604	0x2C00
0x3DEA	0x2C2C	0x8026	0x2808	0x2683	0x9445	0x4316	
0x2704	0x4B01	0xA627	0x164D	0x2682	0x6017	0x5843	
0x3D10	0x432D	0xF817	0x1A26	0x4827	0x0727	0x1659	
0x2705	0x4643	0x0227	0x8317	0xF24D	0x9517	0x4316	
0x3D10	0x1647	0xFA5C	0x0145	0x4E28	0x2545	0x5A43	
0x2715	0x435F	0x0B5F	0xA017	0x094C	0x4017	0x165B	
0x3527	0x4F50	0x5307	0x0727	0x0B17	0x0827	0x4327	
0x053D	0x2604	0x5302	0xF317	0x6D28	0x905D	0x9C45	
0x1045	0x2684	0x4D28	0x2945	0x0817	0x2808	0x6017	
0x4027	0x2027	0x6C4C	0x8017	0x014D	0x530D	0x0727	

Table 8: "Sequencer A" Recommended for the AR0330 Rev1 Sensor (continued)

Writes 1-20	Writes 21-40	Writes 41-60	Writes 61-80	Writes 81-100	Writes 101-120	Writes 121-140	Writes 141-148
0x0427	0xFC53	0x0928	0x0827	0x1A17	0x2645	0x9D17	
0x143D	0x0D5C	0x2C28	0xF217	0x0126	0x5C01	0x2545	

Table 9: "Sequencer B" Recommended for the AR0330 Rev2.0 Sensor

Writes 1-20	Writes 21-40	Writes 41-60	Writes 61-80	Writes 81-100	Writes 101-120	Writes 121-140	Writes 141-158
0x4A03	0x3D10	0x1647	0x6045	0x27FA	0x4C0B	0x0727	0x4316
0x4316	0x2715	0x435F	0x0045	0x45A0	0x6017	0x9D17	0x5A43
0x0443	0x3527	0x4F50	0x8026	0x1707	0x2027	0x2545	0x165B
0x1645	0x053D	0x2604	0xA627	0x27FB	0xF217	0x4017	0x4345
0x4045	0x1045	0x2684	0xF817	0x1729	0x535F	0x0827	0x4027
0x6017	0x4027	0x2027	0x0227	0x4580	0x2808	0x985D	0x9C45
0x2045	0x0427	0xFC53	0xFA5C	0x1708	0x164D	0x2645	0x6017
0x404B	0x143D	0x0D5C	0x0B17	0x27FA	0x1A16	0x5C01	0x0727
0x1244	0xFF3D	0x0D57	0x1826	0x1728	0x1627	0x4B17	0x9D17
0x6134	0xFF3D	0x5417	0xA25C	0x5D17	0xFA26	0x0A28	0x2545
0x4A31	0xEA62	0x0955	0x0317	0x0E26	0x035C	0x0853	0x4017
0x4342	0x2728	0x5649	0x4427	0x8153	0x0145	0x0D52	0x1027
0x4560	0x3627	0x5307	0xF25F	0x0117	0x4027	0x5112	0x9817
0x2714	0x083D	0x5302	0x2809	0xE653	0x9817	0x4460	0x2022
0x3DFF	0x6444	0x4D28	0x1714	0x0217	0x2A4A	0x184A	0x4B12
0x3DFF	0x2C2C	0x6C4C	0x2808	0x1026	0x0A43	0x0343	0x442C
0x3DEA	0x2C2C	0x0928	0x1616	0x8326	0x160B	0x1604	0x2C2C
0x2704	0x4B01	0x2C28	0x4D1A	0x8248	0x4327	0x4316	0x2C00
0x3D10	0x432D	0x294E	0x2683	0x4D4E	0x9C45	0x5843	
0x2705	0x4643	0x5C09	0x1616	0x2809	0x6017	0x1659	

Note: This sequencer is the same as the Ver.5 OTPM sequencer.

Sequencer Patch 1

The steps required to write sequencer patch # 1:

1. The sensor must be in standby (R0x301A[2] = 0).
2. Set the sequencer control port (seq_ctrl_port) to the address of the instruction that must be modified.
3. Update the address instruction.
4. Enable streaming.

Example: Implementing Sequencer Patch # 1 (Modification to the Existing Sequencer) Instruction:

1. The sensor has been placed in standby.
2. Write to address 0x0C in the sequencer. R0x3088 = 0x800C
3. Change the instruction at address 0x0C. R0x3086 = 0x2045

Binning Mode Optimization

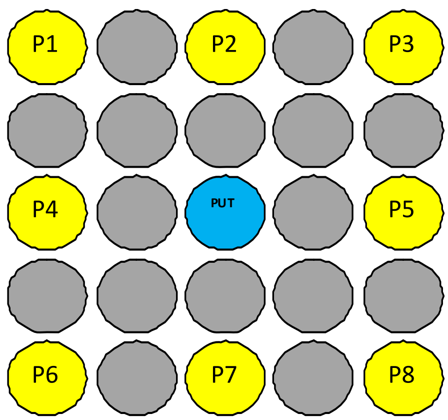
When the sensor is used in binning mode (R0x3040 = 0x1200 or 0x3000), the following settings will improve the matching of the two green channels:

- R0x3ED4 = 0x8F3C
- R0x3ED6 = 0x33CC

2d Single Pixel Defect Correction

The AR0330 is capable of correcting single pixel defects in two dimensions. 2d single pixel defect correction may be enabled by setting register 0x31E0[9] = 1. The default power up state is for defect correction to be enabled.

Figure 6: Pixels Used by Defect Correction Algorithm



The on-the-fly single pixel defect correction is based on the maximum and minimum values of pixels P1 through P8 (see Figure 6) surrounding the pixel under test (PUT). The correction algorithm is implemented as follows:

1. For each PUT, form an 8-neighboring pixel set as shown in Figure #.
2. Sort {P1,P2,…P8}, and determine max and min pixel values
3. Detect and Correct:
   3a. If PUT>max, PUT is replaced by max
   3b. If PUT<min, PUT is replaced by min
   3c. Else, PUT is unchanged

The first two and last two rows/columns of the array are not corrected.

Conclusion

The sensor PLL and interface can be configured to use the HiSPi, MIPI, or parallel interface. The AR0330 supporting documents including the data sheet, the register reference, and the errata can be referenced when programming the sensor.

During the sensor initialization, the sequencer and a set of reserved registers will be programmed to the sensor from the on-chip OTPM. It is important that the camera software driver checks the OTPM version of the sensor and only programs the reserved registers and sequencer specified in this document.

Revision History

Rev. E **11/14/13**

- Applied updated Aptina template
- Updated Figure 2: “Sequencer Shortened to 1014 Clocks Enabling the 1152x648 at120fps Mode,” on page 10
- Updated Figure 3: “Sequencer Lengthened to 1242 Clocks for the 2304x1296 at 60fps Mode,” on page 10
- Updated Table 7, “Recommended Default Register Changes and Sequencer,” on page 15
- Updated Table 9, ““Sequencer B” Recommended for the AR0330 Rev2.0 Sensor,” on page 16
- Added “Binning Mode Optimization” on page 17
- Added “2d Single Pixel Defect Correction” on page 17

Rev. D **5/16/13**

- Updated values for R0x3ED2 to 0x0146 in Table 7, “Recommended Default Register Changes and Sequencer,” on page 15

Rev. C **1/23/12**

- Updated to Table 1, “ISO Gain Table,” on page 8
- Updated Table 6, “OTPM Version Table,” on page 13
- Updated Figure 5: “Sensor Configuration Depending on OTPM Version,” on page 14
- Updated Table 7, “Recommended Default Register Changes and Sequencer,” on page 15

Rev. B **8/11/11**

- Updated to Production
- Updated “Sensor Default Configuration” on page 6
- Updated “Sequencer Update and Frame Rate Calculations” on page 9
- Updated Figure 2, Sequencer Shortened to 1014 Clocks Enabling the 1152x648 at120fps Mode and Figure 3: “Sequencer Lengthened to 1242 Clocks for the 2304x1296 at 60fps Mode,” on page 10
- Updated Table 6, “OTPM Version Table,” on page 13
- Updated Figure 5: “Sensor Configuration Depending on OTPM Version,” on page 14
- Updated Table 7, “Recommended Default Register Changes and Sequencer,” on page 15
- Updated Table 9, ““Sequencer B” Recommended for the AR0330 Rev2.0 Sensor,” on page 17
- Added “Sequencer Patch 1” on page 16

Rev. A **5/26/11**

- Initial release

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.aptina.com

