Technical Note

MT9M019 Power-up Sequence and Sample Register Settings

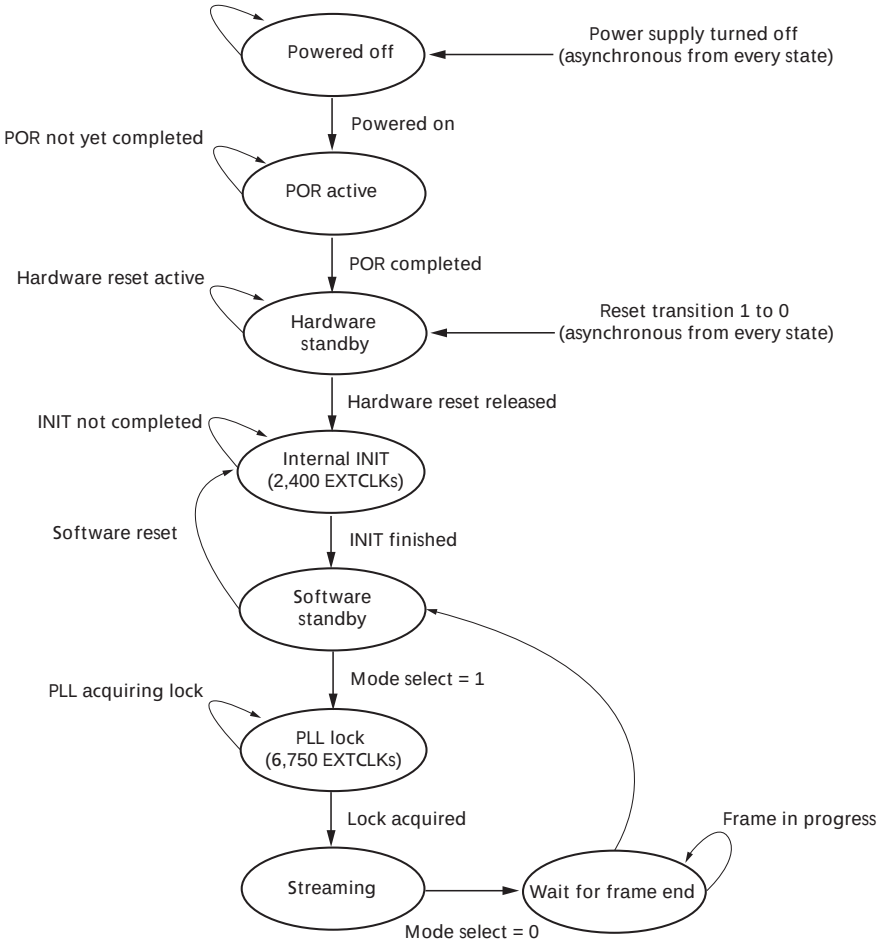
Introduction

This technical note describes the power-up sequence and provides sample register settings for the Micron® MT9M019 CMOS image sensor.

Power-up Sequence

Figure 1 displays a simple flowchart of the MT9M019 power-up sequence. The sequence is described in greater detail in following sections.

Figure 1: MT9M019 System States



The different power supplies need not be turned on in any specific sequence. Once the last supply is stable within the valid ranges specified below, the sensor enters a low-power hardware standby state.

- Analog voltage: 2.40–3.10V (2.80V nominal)
- Digital voltage: 1.70–1.90V (1.80V nominal)

Exit from the hardware standby is controlled by the latter of two events:

1. The negation of the RESET# (XSHUTDOWN) input.
2. A time-out of the internal power-on reset circuit.

When RESET# is asserted, it asynchronously resets the sensor, truncating any frame that is in progress. At that time, the sensor is in its lowest-powered state. When the sensor leaves the hardware standby state, it performs an internal initialization sequence with a duration of 2,400[1] EXTCLK cycles and then enters a low-power software standby state.

The MT9M019 will not respond to read transactions on its two-wire serial interface while the initialization sequence is in progress. Polling a sensor register (R0x0000, for example) allows the user to determine when the initialization sequence has completed. When the sequence is completed, reads will return the operational value for the register (0x14 if R0x0000 is read).

When the sensor leaves software standby mode and enables the voltage-controlled oscillator (VCO) through mode_select = 1, an internal delay will keep the PLL disconnected for 6,750[1] EXTCLK cycles so that the PLL can lock. Once the lock sequence is completed, the sensor enters into the active streaming mode.

Notes: **1.** Cycle times listed here are tentative and subject to change.

Soft Reset Sequence

The MT9M019 has the option of software reset control by writing “1” to the software_reset register (R0x0103). A software reset asynchronously resets the sensor, truncating any frame that is in progress. The sensor briefly enters the hardware standby state and then starts its internal initialization sequence. At that point, the behavior is identical to the power-on reset sequence.

Register Descriptions

The following notes apply to all tables in this section:

Notes: **1.** Hexadecimal values have **0x** in front. All other values are in decimal.
2. Integration time and gain values are examples only. Adjustments might be necessary.
3. Sample frame rates might not be optimal for flicker detection.
4. Delay values (in milliseconds) are examples only. Actual values may vary.

Table 1: Register Descriptions

Register	Description
R0x0103	**software_reset (RW)** This bit always reads as 0. Setting this bit initiates a reset sequence; the frame being generated will be truncated. This register field is an alias of R0x301A–B[0].
R0x301A	**reset_register (RW)**

Table 1: Register Descriptions (continued)

Register	Description
R0x0100	**mode_select (RW)** Setting this bit places the sensor in streaming mode. Clearing this bit places the sensor in a low-power mode. The result of clearing this bit depends upon the operating mode of the sensor. Entry and exit from streaming mode can also be controlled from the signal interface. This register field is an alias of R0x301A[2].
R0x306E	**datapath_select (RW)** SMIA profile mode. Bit 7:0 = profile 0; 1 = profile 1/2
R0x30D4	Reserved.
R0x0112	**ccp_data_format (RW)** [7:0] = The bit-width of the compressed pixel data. [15:8] = The bit-width of the uncompressed pixel data. The value in this register must match one of the valid data_format_descriptor registers (R0x00C2–R0x00C7).
R0x0304	**pre_pll_clk_div** Clock divisor applied to EXTCLK to generate PLL input clock.
R0x0306	**pll_multiplier (RW)** Clock multiplier applied to PLL input clock.
R0x0302	**vt_sys_clk_div (RW)** Clock divisor applied to PLL output clock to generate video timing system clock.
R0x0300	**vt_pix_clk_div (RW)** Clock divisor applied to video timing system clock to generate video timing pixel clock.
R0x030A	**op_sys_clk_div (R)** Clock divisor applied to PLL output clock to generate output system clock. Read-only.
R0x0308	**op_pix_clk_div (RW)** Clock divisor applied to the output system clock to generate the output pixel clock. Allowed values are 1, 2, and 4.
R0x0104	**grouped_parameter_hold (RW)** 0 = Update of multiple registers is synchronized to frame start. 1 = Inhibit register updates. When this bit is returned to 0, all pending register updates will be made on the next frame start. This register field is an alias of R0x301A–B[15].
R0x0400	**scaling_mode (RW)** SMIA scaler mode. 0 = Disable scaler 1 = Enable horizontal scaling 2 = Enable horizontal and vertical scaling
R0x0404	**scale_m (RW)** Scale factor M.
R0x0344	**x_addr_start (RW)** The first column of visible pixels to be read out (not counting any dark columns that may be read). To move the image window, set this register to the starting X value. This register is an alias of R0x3004–5.
R0x0346	**y_addr_start (RW)** The first row of visible pixels to be read out (not counting any dark rows that may be read). To move the image window, set this register to the starting Y value. This register is an alias of R0x3002–5.
R0x0348	**x_addr_end (RW)** The last column of visible pixels to be read out. This register is an alias of R0x3008–9.

Table 1: Register Descriptions (continued)

Register	Description
R0x034A	**y_addr_end (RW)** The last row of visible pixels to be read out. This register is an alias of R0x3006–7.
R0x0382	**x_odd_inc (RW)** Increment applied to odd addresses in X (column) direction. 1 = Normal readout 3 = Read out alternate pixel pairs to halve the amount of horizontal data in a frame (“skip 2x”) This register field is an alias of R0x3040–1[7:5].
R0x0386	**y_odd_inc (RW)** Increment applied to odd addresses in Y (row) direction. 1 = Normal readout 3 = Read out alternate pixel pairs to halve the amount of vertical data in a frame (“skip 2x”) This register field is an alias of R0x3040–1[4:2].
R0x034C	**x_output_size (RW)** Set X output size of the displayed image. Bit[0] is read-only 0. The default value of this register is set to be consistent with the default values of x_addr_end and x_addr_start.
R0x034E	**y_output_size (RW)** Set Y output size of the displayed image. Bit[0] is read-only 0. The default value of this register is set to be consistent with the default values of y_addr_end and y_addr_start. The output image will have two additional rows containing embedded data, in accordance with the frame format descriptors.
R0x0202	**coarse_integration_time (RW)** Integration time programmed in units of line_length_pck. This register is an alias of R0x3012–3
R0x0340	**frame_length_lines (RW)** The number of complete lines (rows) in the output frame, including visible lines and vertical blanking lines. This register is an alias of R0x300A–B.
R0x0342	**line_length_pck (RW)** The number of pixel clock periods in one line (row) time, including visible pixels and horizontal blanking time. This register is an alias of R0x300C–D.
R0x0206	**analog_gain_code_greenR (RW)** The gain code written to this register sets the gain for green pixels on red/green rows of the pixel array. This register is an alias of R0x302A–B.
R0x0208	**analog_gain_code_red (RW)** The gain code written to this register sets the gain for red pixels. This register is an alias of R0x302C–D.
R0x020A	**analog_gain_code_blue (RW)** The gain code written to this register sets the gain for blue pixels. This register is an alias of R0x302E–F.
R0x020C	**analog_gain_code_greenB (RW)** The gain code written to this register sets the gain for green pixels on blue/green rows of the pixel array. This register is an alias of R0x3030–1.

Table 2: Register Values in Specific Modes

Register	Value				
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
R0x0103[1]	0x0001				
R0x301A	0x0018				
R0x0100	0x0000				
R0x306E	0x9080				
R0x30D4	0x0080				
R0x0112[2]	0x0A0A				
R0x0304	0x0002	0x0004	0x0004	0x0004	0x0002
R0x0306	0x0050	0x005A	0x005A	0x005A	0x0050
R0x0302	0x0002				
R0x0300	0x0005				
R0x030A	0x0001				
R0x0308	0x000A				
R0x0104	0x0001				
R0x0400	0x0000				0x0002
R0x0404	0x0010				0x0020
R0x0344	0x0004				

Register	Value				
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
R0x0346	0x0004				
R0x0348	0x0503			0x0501	0x0503
R0x034A	0x0403			0x0401	0x0403
R0x0382	0x0001			0x0003	0x0001
R0x0386	0x0001			0x0003	0x0001
R0x034C	0x0500			0x0280	
R0x034E	0x0400			0x0200	
R0x0202	0x0454		0x0817	0x02CF	0x0454
R0x0340	0x0455		0x0818	0x02D0	0x0455
R0x0342	0x0780	0x0874	0x0D90	0x0680	0x0780
R0x0206	0x0020		0x0008	0x0020	
R0x0208	0x0030		0x000C	0x0030	
R0x020A	0x0030		0x000C	0x0030	
R0x020C	0x0020		0x0008	0x0020	
R0x0104	0x0000				
R0x0100	0x0001				

Notes:
1. Delay after a write access of this register is 300ms.
2. Delay after a write access of this register is 100ms.
3. Mode 1 = SXGA (1280 x 1024) RAW10 30 fps with 16 MHz EXTCLK and 64 MHz PLL.
4. Mode 2 = SXGA (1280 x 1024) RAW10 15 fps with 16 MHz EXTCLK and 36 MHz PLL.
5. Mode 3 = SXGA (1280 x 1024) RAW10 5 fps with 16 MHz EXTCLK and 36 MHz PLL.
6. Mode 4 = 640 x 512 2x Skip RAW10 30 fps with 16 MHz EXTCLK and 36 MHz PLL.
7. Mode 5 = 640 x 512 2x Scaler RAW10 30 fps with 16 MHz EXTCLK and 64 MHz PLL.

Conclusion

For more information on this and other features, refer to the MT9M019 data sheet (visit www.micron.com/imaging to request access to confidential imaging data sheets).

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
prodmktg@micron.com www.micron.com Customer Comment Line: 800-932-4992

Revision History

Rev. B. **8/15/2006**

- Modified PLL settings to meet VCO specifications.
- Made minor adjustments on sample gain values.

Rev. A **7/13/2006**

- Initial release.