Atmel MSL3040/41/50/60/80/86/87/88 Programmers Guide

4-String 120mA and 5/6/8-String 60mA LED Drivers with Integrated Boost Controller and Phase Shifted Dimming

APPLICATION NOTE

Description:

The MSL3040/41/50/60/80/86/87/88 LED drivers offer a complete solution to drive up to eight parallel LED strings at up to 40V. The LED current sinks of the MSL3050, MSL3060, MSL3080, MSL3086, MSL3087, and MSL3088 control up to 60mA each, and the MSL3040 and MSL3041 current sinks control up to 120mA each, for up to 19W of LED power. The MSL3050, MSL3060 and MSL3080 allow parallel driver connection, increasing string current capability. A single resistor sets LED current with string matching and accuracy within ±3%. Each driver varies by the number of current sinks and features (see Table 1 below). Basic features include integrated boost regulator controller, PWM circuitry that allows up to 4095:1 dimming, phase-shifted LED PWM dimming, up to eight LED drive outputs, integrated or separate duty cycle and frequency control inputs, a 1MHz I^2C compatible serial interface and internal registers for PWM dimming control. Additionally, integrated fault detection circuitry acts on string open-circuit and LED short circuit faults, boost regulator over-voltage faults, and die over-temperature faults. The proprietary Efficiency Optimizer minimizes power use while maintaining proper LED current, and supports interconnecting multiple drivers to power more than eight strings while maintaining optimum efficiency.

Table 1. LED Driver Parts Comparison

PART	NUMBER OF LED STRINGS	MAX CURRENT PER STRING	PHASE SHIFTED STRING DRIVERS	INTERNAL BOOST CONTROLLER	RESISTOR SET LED SHORT CIRCUIT THRESHOLD	SEPARATE SYNC INPUT***	BEST FOR
MSL3086	8	60mA	YES	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
MSL3087*	8	60mA	YES	NO	YES	NO	SMALL TV
MSL3088	8	60mA	YES	YES	NO	YES	SMALL TV
MSL3080	8	60mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	4**	120mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	2**	240mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	1**	480mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
MSL3040*	4	120mA	YES	YES	YES	NO	MONITOR, AUTOMOTIVE
MSL3041*	4	120mA	YES	YES	YES	YES	MONITOR, AUTOMOTIVE
MSL3050*	5	60mA	NO	YES	YES	NO	INDUSTRIAL PANEL
	1**	300mA	NO	YES	YES	NO	INDUSTRIAL PANEL
MSL3060*	6	60mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	3**	120mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	2**	180mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL
	1**	360mA	NO	YES	YES	NO	MONITOR, INDUSTRIAL PANEL

* Future product, contact factory for information.
** Drivers without phase shift allow parallel connection of string drive outputs for increased string current.
*** Drivers with separate SYNC input expect two input control signals, one for dimming duty cycle and one for dimming frequency.

Table of Contents

1.0 Document Scope

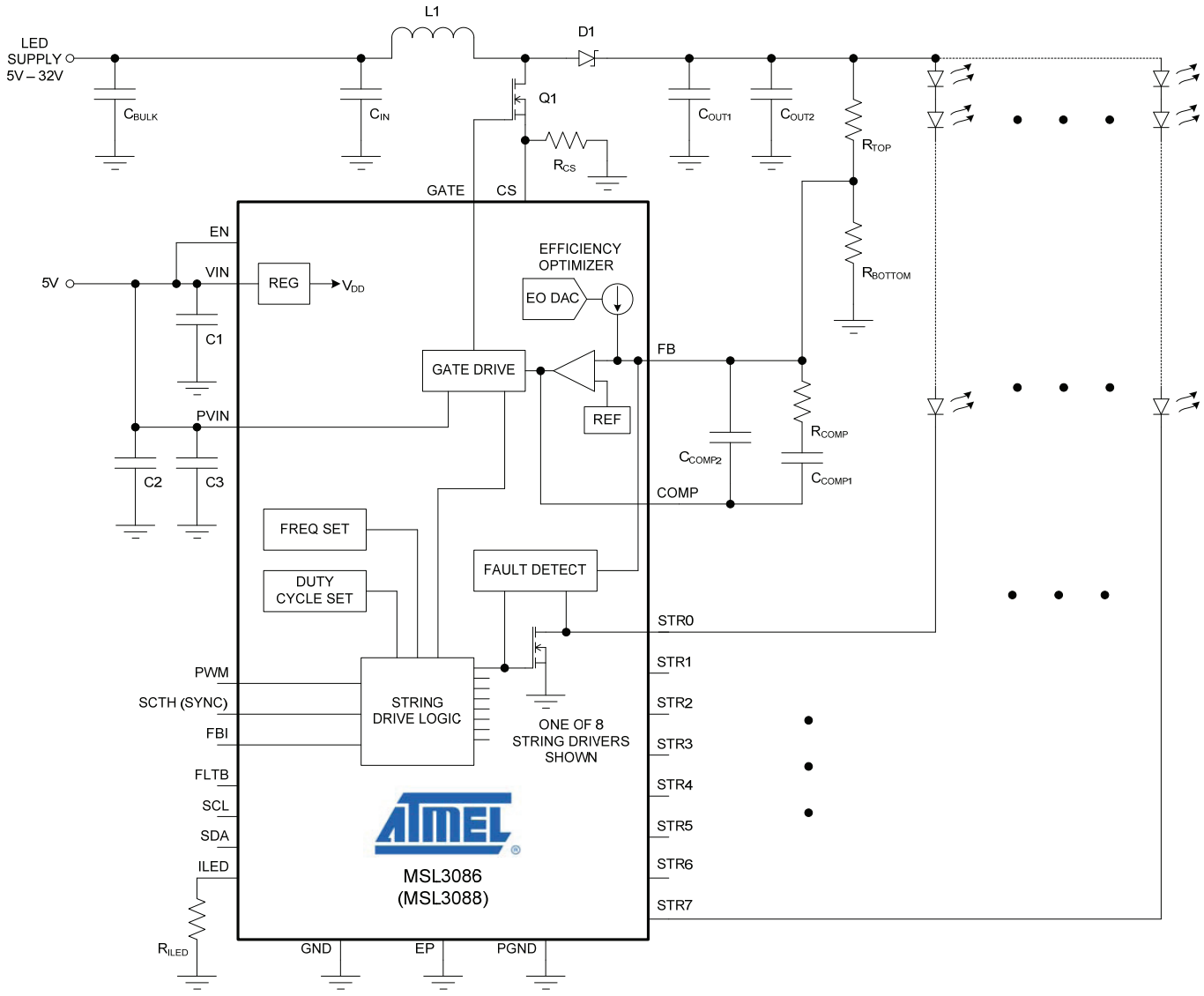
This MSL3040/41/50/60/80/86/87/88 Programmers Guide is an addendum to the datasheets MSL3040/MSL3041, MSL3050/MSL3060/MSL3080 and MSL3086/MSL3087/MSL3088, and gives detailed information and instructions on how to configure, monitor, and control the LED drivers through the I²C compatible serial interface. These drivers operate without I²C serial port access; using the serial port allows access to advanced features, dimming modes, fault detection and monitoring modes, and test and debug features.

All registers are volatile and reset when power is removed, when the driver is turned off by the hardware enable input (EN), or when automatically shut-off due to an over-temperature event. If a configuration other than the default programmed configuration is used, it must be loaded each time the driver is turned on. Note that non-standard configurations are available from the factory, contact the factory for information.

Register default values for each driver are listed in the Register Details section where appropriate.

2.0 Hardware

Figure 1.1: LED Driver Block Diagram



3.0 Basic Operation

All MSL30xx drivers, except the MSL3087, include a boost regulator controller to make a voltage, V_{LED}, that powers the LEDs. The boost regulator accepts an input voltage of 5V to 32V and boosts it up to a maximum of 40V (limited by the voltage stand-off rating of the STR*n* driver outputs). V_{LED} is controlled by a resistive voltage divider, R_{TOP} and R_{BOTTOM} in Figure 1 above, connected to the controller feedback input FB. The Efficiency Optimizer monitors LED driver voltage and drives FB to control the V_{LED} voltage to minimize power use while assuring proper LED current. The MSL3087 controls an external boost regulator with its Efficiency Optimizer output, FBO, which also allows daisy-chaining with other drivers in the family to drive additional strings from a single optimized power supply. Provision is made for boost regulator control loop compensation via the COMP input.

Internally or externally generated PWM dimming signals gate the LED string current-sink outputs STR*n* (the number of strings vary for different members of the MSL30xx family, Table 1 on page 1). A single external PWM signal connected to the PWM input distributes to all strings via the string drive logic (Figure 1), with most parts in the family phase shifting the dimming signals (Table 1). The additional SYNC input allows separate frequency control for the MSL3088 and MSL3041. All drivers allow internal register control of PWM dimming frequency and duty cycle (in the place of external inputs) via the I^2C compatible serial interface. All parts in this family have a requirement of minimum on time of 3µs.

4.0 I²C/SMBus Compatible Serial Interface

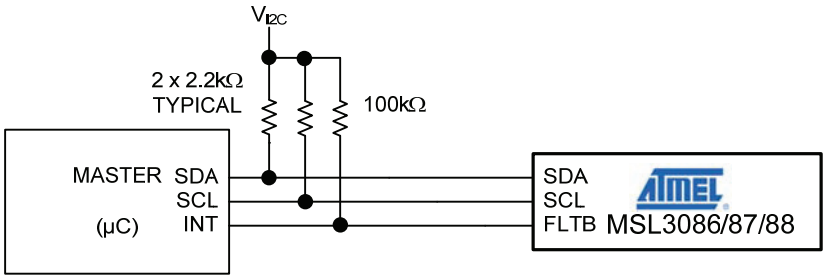
4.1 Overview & Timing

The I²C serial interface allows access to the control and status registers. The Register Map section shows register definitions and the Register Details section shows detailed register functions and default settings. The MSL3040/41/50/60/80/86/87/88 operate as I²C/SMBus slaves that send and receive data to a master. The interface is needed only to allow control and monitoring over some functions and advanced features, and is not required for basic operation. When not using the serial interface, connect SDA and SCL to GND.

The serial interface is suitable for 100kHz, 400kHz and 1MHz communication. The interface uses bi-directional data line SDA and clock input SCL to achieve bidirectional communication between master and slaves. The FLTB fault output optionally alerts the host system to faults. During over temperature shutdown the serial interface is disabled and register settings are reset to their default values.

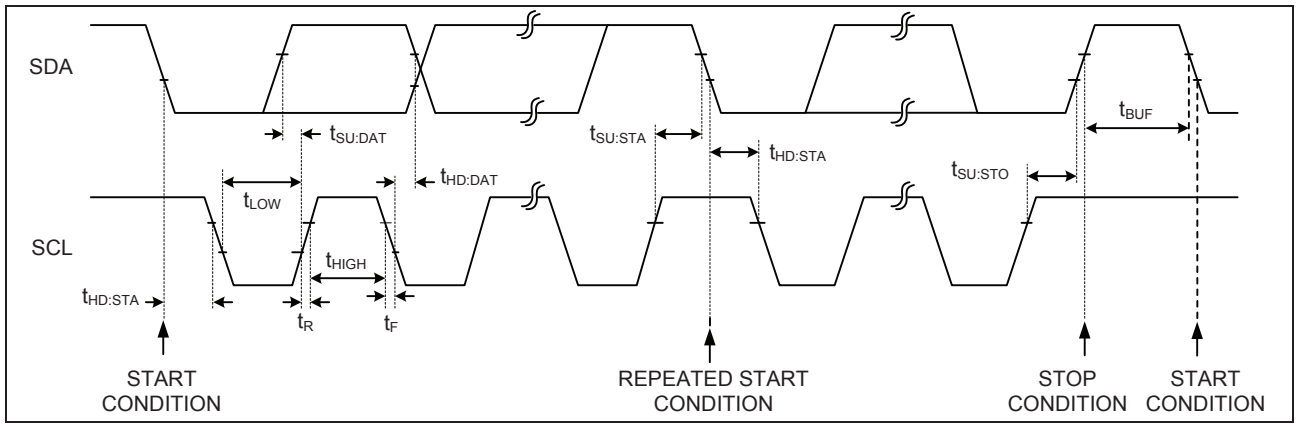
The master, typically a microcontroller, initiates all data transfers, and generates the clock that synchronizes the transfers. SDA operates as both an input and an open-drain output. SCL operates only as a slave input (master output), and does not perform clock-stretching. Use pull-ups on SDA, SCL and FLTB.

Figure 4.1: I²C Interface Connections



A transmission consists of a START condition sent by a master, a 7-bit slave address plus one R/W bit, an acknowledge bit, none or many data bytes each separated by an acknowledge bit, and a STOP condition (Figure 4.2, Figure 4.3 and Figure 4.4).

Figure 4.2: I²C Serial Interface Timing Details



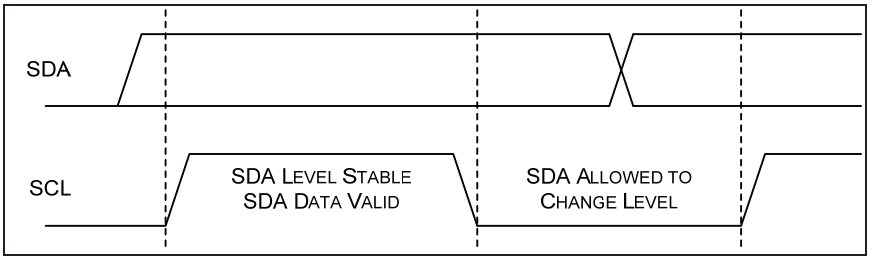
4.2 I²C Bus Timeout

The bus timeout feature (register 0x02 bit D7) allows the MSL3040/41/50/60/80/86/87/88 to reset the serial bus interface if a communication ceases before a STOP condition is sent. If either SCL or SDA is low for more than 30ms (typical), then the transaction terminates, SDA releases and the interface waits for another START condition.

4.3 I²C Bit Transfer

One data bit is transferred during each clock pulse. Ensure SDA is stable while SCL is high.

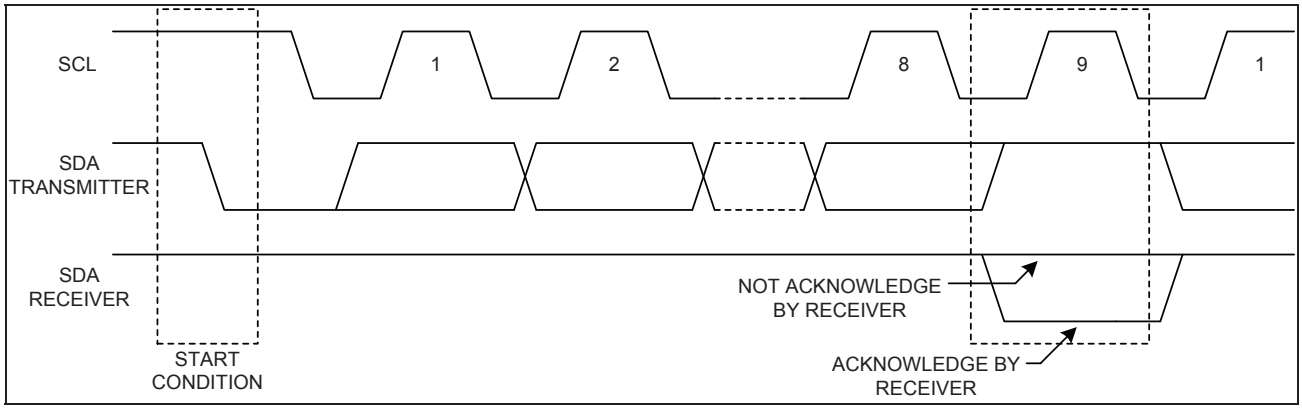
Figure 4.3: I²C Bit Transfer



4.4 I²C START and STOP Conditions

Both SCL and SDA remain high when the interface is free. The master signals a transmission with a START condition by transitioning SDA from high to low while SCL is high. When the master has finished communicating with the slave, it issues a STOP condition by transitioning SDA from low to high while SCL is high. The bus is then free.

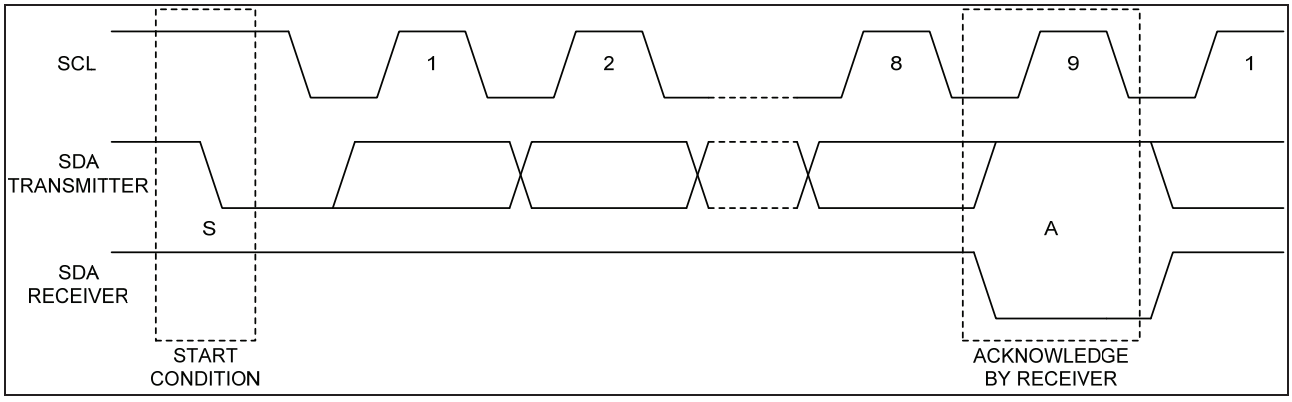
Figure 4.4: I²C START and STOP Conditions



4.5 I²C Acknowledge Bit

The acknowledge bit is a clocked 9th bit which the recipient uses to handshake receipt of each 8-bit byte of data. The master generates the 9th clock pulse, and the recipient holds SDA low during the high period of the clock pulse. When the master is transmitting to the slave, the slave pulls SDA low because the slave is the recipient. When the slave is transmitting to the master, the master pulls SDA low because the master is the recipient.

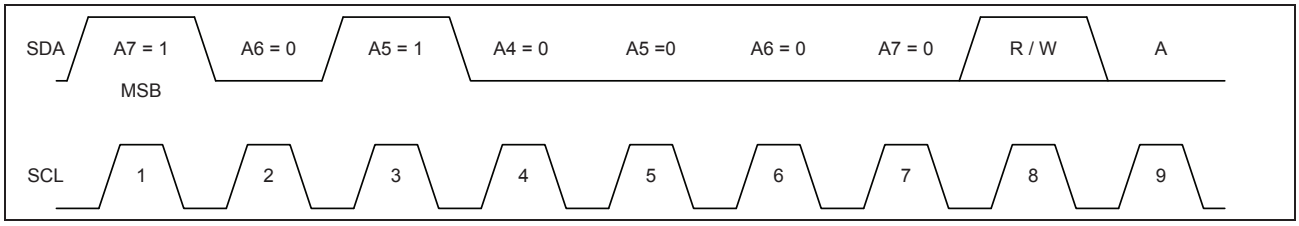
Figure 4.5: I²C Acknowledge



4.6 I²C Slave Address

The MSL3040/41/50/60/80/86/87/88 have a 7-bit long slave address, 0b1010000, followed by an eighth bit, the R/W bit, that combine to make 2 separate 8-bit read and write addresses (i.e. the slave addresses are 0xA0 for write operations and 0xA1 for read operations). The R/W bit is low for a write and high for a read. All MSL3040/41/50/60/80/86/87/88 have the same 0xA0/0xA1 slave address; when using multiple drivers and communicating with them through their serial interfaces, make external provision to route the serial interface to the appropriate driver.

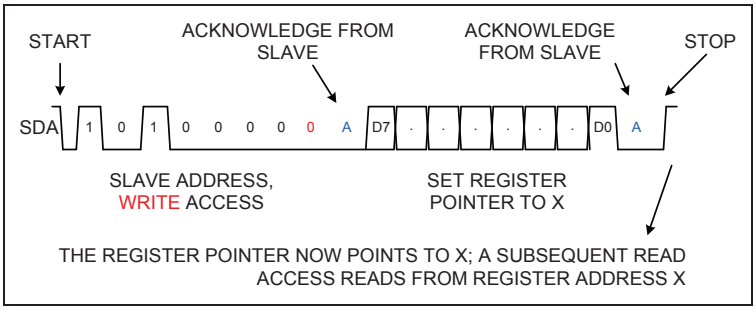
Figure 4.6: I²C Slave Address



4.7 I²C Message Format for Writing to the MSL3040/41/50/60/80/86/87/88

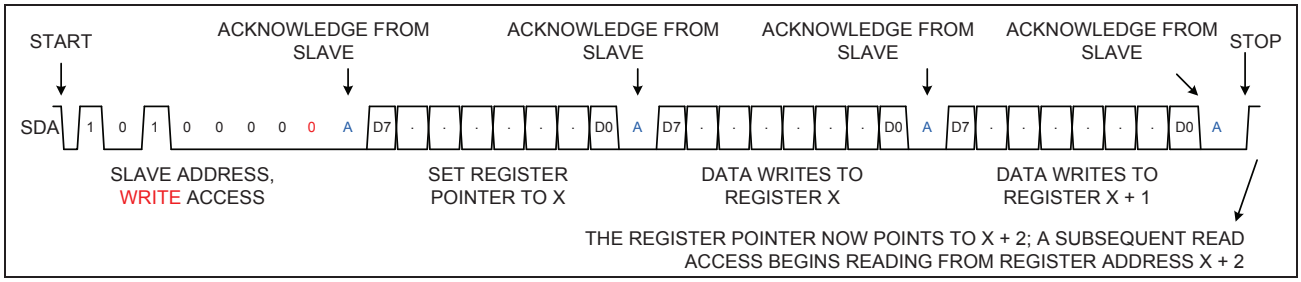
A write to the MSL3040/41/50/60/80/86/87/88 contains the slave address, the R/W bit cleared to 0, and at least 1 byte of information. The first byte of information is the register address byte. The register address byte is stored as a register pointer, and determines which register the following byte is written into. If the MSL3040/41/50/60/80/86/87/88 detect a STOP condition after the register address byte is received, then it takes no further action beyond setting the register pointer.

Figure 4.7: I²C Writing a Register Pointer



When no STOP condition is detected, the byte transmitted after the register address byte is a data byte, and is placed into the register pointed to by the register address byte. To simplify writing to multiple consecutive registers, the register pointer auto-increments during each following acknowledge period; further data bytes transmitted before a STOP condition fill subsequent registers.

Figure 4.8: I²C Writing Two Data Bytes

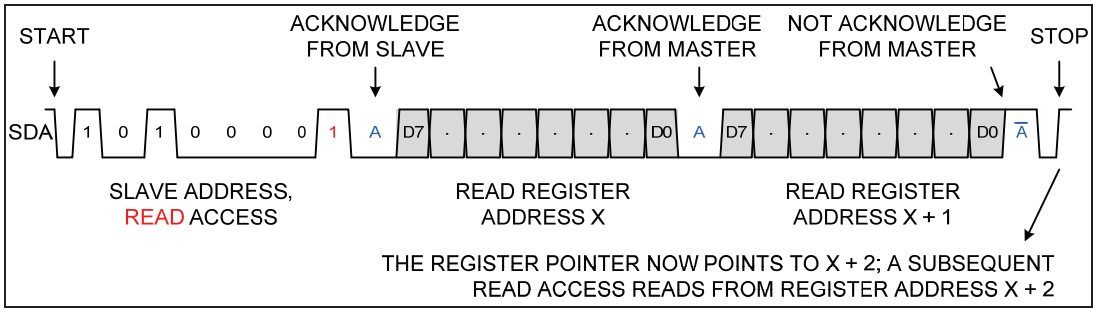


4.8 I²C Message Format for Reading Registers

Read the registers using one of two techniques.

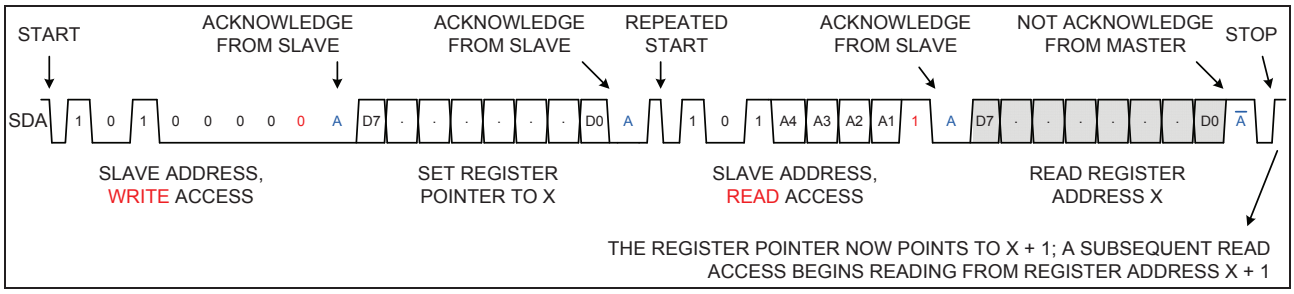
The first technique begins the same way as a write, by setting the register address pointer as shown in Figure 4.7, including the STOP condition (note that even though the final objective is to read data, the R/W bit is first sent as a write because the address pointer byte is being written). Follow the Figure 4.7 transaction by that shown in Figure 4.9, with a new START condition and the slave address, this time with the R/W bit set to 1 to indicate a read. Then, after the slave initiated acknowledge bit, clock out as many bytes as desired, separated by master initiated acknowledges. The pointer auto-increments during each master initiated acknowledge period. End the transmission with a not-acknowledge followed by a stop condition.

Figure 4.9: I²C Reading Register Data with Preset Register Pointer



The second read technique is illustrated in Figure 4.10. Set the register pointer as shown in Figure 4.6 without sending a STOP condition, send a repeated START condition after the second acknowledge bit, then send the slave address again with the R/W bit set to 1 to indicate a read. Then clock out the data bytes separated by master initiated acknowledge bits. The register pointer auto-increments during each master initiated acknowledge period. End the transmission with a not-acknowledge followed by a stop condition. Use this technique for buses with multiple masters, because the read sequence is performed in one continuous transaction.

Figure 4.10: I²C Reading Register Data Using a Repeated START



4.9 Register Map

The I²C slave ID is 0xA0 for read and 0xA1 for write. Do not change bits that are not described.

ADDRESS AND REGISTER NAME		FUNCTION	REGISTER DATA							
			D7	D6	D5	D4	D3	D2	D1	D0
Configuration Registers										
0x00	strEnRg	LED String Enables	strEn[7:0]							
0x01	sysCtrlRg	System Control	-	eoEn	-	-	actOnBst OVFlt	actOnStr SCFlt	actOnStr OCFlt	-
0x02	fltEnRg	Fault Detect Enable	I²CTimeOutEn	-	-	-	bstOVFltEn	strSCEn	strOCEn	fboOCFltEn
0x03	strFltEnRg	String Fault Enable	fltEnStr[7:0]							
0x04	sCThCtrlRg	SC Threshold control	-	-	scQualDly	scDbncDly	-	-	scThrshLvl[1:0]	
Power / Fault Status Registers										
0x05	fltStsRg	Fault Status	fltBDrv	-	-	-	bstOVFltDet	strSCDet	strOCDet	fboOCFltDet
0x06	strOCStsRg	String Open Circuit Fault Status	strOC[7:0]							
0x07	strSCStsRg	String Short Circuit Fault Status	strSC[7:0]							
0x08	strEnStsRg	String Enable Status after auto fault handling	strEnSts[7:0]							
0x09	bstStsRg	Boost/Boot-Load Status	adPwmEn	blRecovDone	blDone	-	-	bstOVFlt	bstPwrGood	bstSftStrtDone
Efficiency Optimizer Status										
0x0C	eoDacRg	EO DAC Read back	eoDac[7:0]							
0x0D	eoStsRg	EO Status	-	-	-	fboOCFlt	-	eoDacAct	eoCal	eoInitCal
PWM Control Registers										
0x10	pwmCtrlRg	PWM Control	-	dcMsrMode	phaShft	phaShftpairs	intDuty	intFreq	pwmDrct	syncPol
0x11	freqPhaRg	Internal Frequency and Phase	freqPha [7:0]							
0x12			freqPha[15:8]							
0x13	dutyCycRg	Internal Duty Cycle	dutyCyc[7:0]							
0x14			freqMul[1:0]		-	-	dutyCyc[11:8]			
0x20	Reserved	Reserved	'0x00'							
0x21			-				'0x0'			
0x22	Reserved	Reserved	'0x00'							
0x23			-				'0x0'			
Power Management Register										
0x7F	sleepRg	Put part to sleep and enable power savings	sleep	slpPwrSv	-	-	-	-	-	-
Efficiency Optimizer Control Registers (MSL3087 only)										
0x84	eoCtrl0Rg	EO Control Register 0	hdrmStep[1:0]		aCalDly[1:0]		stepDly[1:0]		iSinkConfDly[1:0]	
0x85	eoCtrl1Rg	EO Control Register 1	decrStep[1:0]		incrStep[1:0]		iCalPWM	aCal100	aCalEn	iChkDis

4.10 Register Details

The following sections describe the MSL3040/41/50/60/80/86/87/88 registers. Factory default settings are listed where appropriate. Where default values differ between driver types, the defaults for each are listed. Bits labelled "-" are reserved; do not change them, defective operation may result. Bits labelled "x" are not used, and so may be written to with any value without adverse effects on operation.

String Enable Register 0x00

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x00	strEnRg		strEn7	strEn6	strEn5	strEn4	strE3	strEn2	strEn1	strEn0
MSL3040/41		0x0F	0	0	0	0	1	1	1	1
MSL3050		0x1F	0	0	0	1	1	1	1	1
MSL3060		0x3F	0	0	1	1	1	1	1	1
MSL3080/86/87/88		0xFF	1	1	1	1	1	1	1	1

StrEn*n*: Set String Enable bits to 1 to enable, and clear them to 0 to disable the corresponding LED driver strings. The Efficiency Optimizer and the fault logic ignore disabled outputs.

4.11 System Control Register 0x01

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x01	sysCtrlRg	0x4E	x	eoEn	x	x	actOnBst OVFlt	actOnStr SCFlt	actOnStr OCFlt	x

- **eoEn:** Efficiency Optimizer Enable; Set to 1 to enable the Efficiency Optimizer to dynamically control the boost regulator output voltage, Clear to 0 to disable control of the boost regulator output voltage.
- **actOnBstOVFlt:** Act On Boost Over Voltage Fault; Set to 1 to turn off all LED strings upon detection of a boost over-voltage condition; this fault is non-latching, strings turn on when the fault goes away. Clear to 0 to ignore boost over-voltage condition. This actOnBstOVFlt bit is ignored when bstOVFltEn (bit D3 of 0x02) = 0.
- **actOnStrSCFlt:** Act On String Short Circuit Fault; Set to 1 to turn off LED strings that detect an LED short circuit fault, Clear to 0 to leave LED strings on that detect an LED short circuit fault. Set the short circuit threshold voltage with the Short Circuit Threshold bits D0 and D1 of register 0x04. The Efficiency Optimizer ignores outputs that are turned off by an LED short circuit fault. This actOnStrSCFlt bit is ignored when strSCEn (bit D2 of 0x02) = 0.
- **actOnStrOCFlt:** Act On String Open Circuit Fault; Set to 1 to stop driving LED strings that are open circuit, Clear to 0 to continue driving LED strings that are open circuit. Regardless of the state of this bit, the Efficiency Optimizer disregards strings that are open circuit. When actOnStrOCFlt is low the phase engine does not re-calculate phase shifts when a string open circuit occurs. actOnStrOCFlt bit is ignored when strOCEn (bit D1 of 0x02) = 0.

4.12 Fault Enable Register 0x02

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x02	fltEnRg	0x8F	I²CTime OutEn	x	x	x	bstOVFltEn	strSCEn	strOCEn	fboOCFltEn

- **I²CTimeOutEn:** I²C Time Out Enable. Set to 1 to enable I²C bus transaction timeout when the bus is stalled beyond 30ms, Clear to 0 to disallow I²C bus time out.
- **bstOVFltEn:** Boost Over-Voltage Fault Enable; Set to 1 to have a boost over-voltage fault pull FLTB low, Clear to 0 to not pull FLTB low when a boost over-voltage faults occurs. This fault is non-latching.
- **StrSCEn:** String Short Circuit Fault Enable; Set to 1 to have LED short circuit faults pull FLTB low, Clear to 0 to have LED short circuit faults not pull FLTB low.
- **strOCEn:** String Open Circuit Fault Enable; Set to 1 to have string open circuit faults pull FLTB low, Clear to 0 to have string open circuit faults not pull FLTB low. When this bit is zero and a string open circuit fault occurs, the Efficiency Optimizer attempts to bring the string back in to current regulation by increasing the boost regulator output voltage up to its highest value, where it remains without indicating a fault condition. In this state fictitious LED short circuit faults may occur.
- **fboOCFltEn:** Feedback Out Open Circuit Fault Enable; Set to 1 to have feedback open circuit faults pull FLTB low, Clear to 0 to have feedback open circuit faults not pull FLTB low. The feedback connection is labelled FBO on the MSL3087, and FB on all others.

4.13 String Fault Enable Register 0x03

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x03	strFltEnRg	0xFF	strFltEn7	strFltEn6	strFltEn5	strFltEn4	strFltEn3	strFltEn2	strFltEn1	strFltEn0

- **strFltEn*n*:** Individual string fault enable bits. Set to 1 to enable string fault detection, Clear to 0 to disable string fault detection.

4.14 Short Circuit Threshold Control Register 0x04

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x04	sCThCtrlRg	0b0000 00__	x	x	scQualDly	scDbncDly	-	x	scThrshLvl[1:0]	

- **scQualDly:** LED Short Circuit Qualification Delay; Clear to 0 for a 256ms short circuit qualification delay, Set to 1 for a 512ms short circuit qualification delay. An LED short circuit must last for the full qualification delay time to be flagged as a fault.
- **scDbncDly:** LED Short Circuit De-bounce Delay; Clear to 0 for a 2µs short circuit de-bounce delay, Set to 1 for a 4µs short circuit de-bounce delay.
- **scThrshLvl[1:0]:** Short Circuit Threshold Setting; 0b00 = 4.9V, 0b01 = 5.8V, 0b10 = 6.8V, 0b11 = 7.6V. scThrshLvl auto-programs when EN is taken high. The value scThrshLvl takes is based on the value of the resistor connected from SCTH to GND, R_{SCTH} (Table 2). MSL3088 has no SCTH input; these bits default to 0b10 = 6.8V.

Table 4.1: Short Circuit Threshold Resistor (R_{SCTH})

R_{SCTH}	scThrshLvl[1:0]	Threshold Voltage
1.0kΩ (or GND)	0b00	4.9V
27kΩ	0b01	5.8V
68kΩ	0b10	6.8V
330kΩ (or OPEN)	0b11	7.6V

4.15 Fault Status Register 0x05

ADDRESS AND REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x05	fltStsRg	Read Only Clear on read	fltBDrv	x	x	x	bstOV FltDet	strSCDet	strOCDet	fboOC FltDet

- **fltBDrv:** FLTB Driver; equals 1 when any fault is detected (any bit D0 through D3 is 1). When fltBDrv = 1, the hardware FLTB output is low.
- **bstOVFltDet:** Boost Over-Voltage Fault Detected; equals 1 when boost over-voltage fault is detected (non-latching).
- **strSCDet:** String Short Circuit Fault Detected; sets to 1 when any string short circuit fault is detected.
- **strOCDet:** String Open Circuit Fault Detected; sets to 1 when any string open circuit fault is detected.
- **fboOCFltDet:** FB Open Circuit Fault Detected; sets to 1 when FB open circuit fault is detected.

Clear faults by reading fault register 0x05 and writing 0xFF to registers 0x06 and 0x07, or by toggling EN low then high.

4.16 String Open Circuit Status Register 0x06

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x06	strOCStsRg	Read-only Clear on write	strOC7	strOC6	strOC5	strOC4	strOC3	strOC2	strOC1	strOC0

- **strOC*n*:** Each bit sets to 1 to indicate that the driver detected an open circuit fault on the corresponding string; read to determine which strings are faulted. Clear faults by reading fault register 0x05 and writing a '1' to the bit showing a fault, or by toggling EN low then high.

4.17 String Short Circuit Status Register 0x07

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x07	strSCStsRg	Read-only Clear on write	strSC7	strsC6	strSC5	strSC4	strSC3	strSC2	strSC1	strSC0

- **strSC*n*:** Each bit sets to 1 to indicate that the driver detected a short circuit fault on the corresponding string; read to determine which strings are faulted. Clear faults by reading fault register 0x05 and writing a '1' to the bit showing a fault, or by toggling EN low then high.

4.18 String Enable Status Register 0x08

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x08	strEnSts	Read-only	strEnSts7	strEnSts6	strEnSts5	strEnSts4	strEnSts3	strEnSts2	strEnSts1	strEnSts0

- **strEnSts*n*:** Each bit sets to 1 to indicate that the corresponding string is turned off due to the enable register 0x00 or due to faults.

4.19 Boost/Boot-Load Status Register 0x09

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x09	bstStsRg	Read-only	-	blRecovDone	blDone	x	x	bstOVFlt	bstPwrGood	bstSftStrtDone

- **blRecovDone:** Boot-Load Recovery Complete. Sets to 1 when all trim and control registers are ready; signals the boost regulator to begin soft start.
- **blDone:** Boot-load complete; sets to 1 after the initial register values are loaded.
- **bstOVFlt:** Boost Over Voltage Fault; sets to 1 when boost controller over-voltage fault is detected (not latching).
- **bstPwrGood:** Boost Power Good; sets to 1 when boost output power is in regulation.
- **bstSftStrtDone**: Boost Soft Start Done. Sets to 1 when the boost soft-start is complete; drivers are disabled until bstSftStrDone transitions high.

4.20 Efficiency Optimizer DAC Readback Register 0x0C

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x0C	eoDacRg	Read Only	eoDacRg(7:0							

- **eoDacRg[7:0]:** Reports the EO DAC setting, which determines EO (FBO on MSL3087, FB on all other drivers) output current at 1.1µA per LSB, which in turn controls the boost regulator output voltage.

4.21 Efficiency Optimizer Status Register 0x0D

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x0D	eoStsRg	Read-only	-	-	-	fboOCFlt	-	eoDacAct	eoCal	eoInitCal

- **fboOCFlt:** Feedback Out Open Circuit Fault; equals 1 when the driver detects an open circuit at feedback (FBO on MSL3087, FB on all other drivers).
- **eoDacAct:** Efficiency optimizer DAC active; equals 1 when feedback (FBO on MSL3087, FB on all other drivers) is sourcing current to the boost regulator feedback node to control the boost regulator output voltage.
- **eoCal:** Efficiency Optimizer Calibration; equals 1 when the EO is calibrating. The EO automatically re-calibrates V_{OUT} every 1 second. An EO Calibration will also occur immediately when it detects a loss of string current regulation.
- **eoInitCal:** Efficiency Optimizer Initial Calibration; equals 1 when the EO is performing an initial calibration. Initial Calibration takes a maximum of 0.52 seconds.

4.22 PWM Control Register 0x10

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x10	pwmCtrlRg		x	dcMsrMode	phaShft	phaShftPairs	intDuty	intFreq	pwmDrct	syncPol
MSL3040		0x72	0	1	1	1	0	0	1	0
MSL3041		0x70	0	1	1	1	0	0	0	0
MSL50/60/86/87		0x62	0	1	1	0	0	0	1	0
MSL3080		0x42	0	1	0	0	0	0	1	0
MSL3088		0x60	0	1	1	0	0	0	0	0

- **dcMsrMode:** Duty Cycle Measure Mode; determines response to a PWM input of less than 20Hz. When dcMsrMode = 1 and the PWM input is 20Hz or greater, the PWM dimming outputs take the duty cycle of the PWM input signal. When dcMsrMode = 1 and the PWM input is less than 20Hz the PWM dimming output duty cycle is fixed at 0% when the input is low, and is fixed at 100% when the input is high. When dcMsrMode = 0 and the PWM input is 20Hz or greater, the PWM dimming signals take the duty cycle of the PWM input signal. When dcMsrMode = 0 and the PWM input is less than 20Hz, the PWM dimming output duty cycle is fixed at the last valid measured duty cycle before the input fell to less than 20Hz. This allows a PWM input to set the output duty cycle (with a signal above 20Hz) and then cease while the outputs continue PWM dimming of the LED strings. When pwmDrct = 1 and phaShft = 1, this bit needs to be set to 1.
- **phaShft:** Phase Shift; Set to 1 to phase shift the string PWM dimming, Clear to 0 for synchronous PWM dimming. Phase shift control is valid regardless of the settings of bits D1, D2 and D3 of this register.
- **phaShftPairs:** Phase Shift in Pairs; Set to 1 to phase shift the string PWM dimming so that adjacent strings operate as pairs, synchronizing STR0 to STR1, STR2 to STR3, STR4 to STR5 and STR6 to STR7. Clear to 0 for independent string PWM dimming phase.
- **intDuty:** Internal Duty Cycle; Set to 1 to use registers 0x13 and 0x14 to set the PWM dimming duty cycle, Clear to 0 to use the signal at the PWM input to set the duty cycle. pwmDrct (Bit D1) = 1 overrides this bit. See Table 3 below.
- **intFreq:** Internal Frequency; Set to 1 to use registers 0x11 and 0x12 to set the PWM dimming frequency, Clear to 0 to use the signal at the PWM input (SYNC input for MSL3041 and MSL3088) to set the PWM dimming frequency. pwmDrct (Bit D1) = 1 overrides this bit. See Table 3 below.
- **pwmDrct:** PWM Direct; Set to 1 to use the signal at the PWM input to control both the PWM dimming frequency and duty cycle, Clear to 0 to use settings indicated by intDuty (bit D2) and intFreq (bit D3) to determine PWM dimming frequency and duty cycle. See Table 3 below.
- **syncPol:** Sync Polarity; Set to 1 to synchronize string PWM dimming to the rising edge of the signal at the SYNC input, Clear to 0 to synchronize to the falling edge. This bit is valid only for the MSL3041 and MSL3088, and ignored by the other drivers.

Table 4.2: Internal/External Frequency and Duty Cycle Selection

REGISTER BITS			MSL3041/50/60/80/86/87 STRING DRIVERS GET		MSL3040/88 STRING DRIVERS GET	
pwmDrct	intDuty	intFreq	DUTY CYCLE	FREQUENCY	DUTY CYCLE	FREQUENCY
			INFORMATION FROM		INFORMATION FROM	
0	0	0	PWM INPUT	PWM INPUT	PWM INPUT	SYNC INPUT
0	0	1	PWM INPUT	0x11, 0x12	PWM INPUT	0x11, 0x12*
0	1	0	0x13, 0x14	PWM INPUT	0x13, 0x14*	SYNC INPUT
0	1	1	0x13, 0x14*	0x11, 0x12*	0x13, 0x14*	0x11, 0x12*
1	x	x	PWM INPUT	PWM INPUT	PWM INPUT	PWM INPUT

*Assure that unused PWM and SYNC inputs are held to logic low.
x = don't care
NOTE: Regardless of the dimming technique used a minimum on-time of 3µs is required.

4.23 PWM Frequency/Phase Registers 0x11 and 0x12

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x11	freqPhaRg	0x00	freqPha[7:0]							
0x12		0x00	freqPha[15:8]							

- **freqPha[15:0]:** PWM Frequency or Phase Setting Registers. When internal frequency is selected via register 0x10, the PWM dimming frequency equals 20MHz divided by the decimal value of this 16-bit register pair. When using an external signal to set the PWM dimming frequency (SYNC), this register pair determines the delay time between the edge of the input signal transition and the beginning of the first enabled string PWM dimming on-time, where the delay time is 50ns times the decimal value of the lower 12-bits; when phase shifting is enabled, subsequent strings are phase delayed using the first string as the zero-time reference.

4.24 PWM Duty Cycle Registers 0x13 and 0x14

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x13	dutyCycRg	0x00	dutyCyc[7:0]							
0x14		0x00	freqMul[1:0]		x	x	dutyCyc[11:8]			

- **dutyCyc:** 12-bit PWM Duty Cycle Registers. When internal duty cycle is selected, using register 0x10, PWM dimming duty cycle is a linear relation where 0x000 = 0% and 0xFFF = 100%. A minimum on-time of 3 µs is required.
- **freqMul:** SYNC Frequency Multiplier; acts on both internally and externally generated PWM frequency. The PWM control signal frequency (either applied to PWM or SYNC inputs, or internally generated) is multiplied by the decimal value of these two bits, taken as LSB = 2^0 and MSB = 2^1, plus 1 to generate the string PWM dimming frequency. This allows synchronized PWM dimming at multiples of the LCD panel refresh rate. For instance, when freqMul = 0b00 the multiplication factor = 0*(2^1) + 0*(2^0) + 1 = 1.

4.25 Reserved registers 0x20-0x23

These four registers are reserved and should maintain a value of 0x00 for all four bytes.

4.26 Sleep Registers 0x7F

ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x7F	sleepRg	0x00	Sleep	slpPwrSv	x	x	-	-	x	x

- **sleep and slpPwrSv:** Sleep and Sleep Power Save; use these bits together to enter/exit low power sleep mode. When sleep = slpPwrSv = 1 the driver is asleep and input current is reduced to less than 1.8mA, when sleep = slpPwrSv = 0 the driver operates fully. When asleep, all the registers maintain their states and the I²C remains active. Table 4 presents low power controls and behaviours.

Table 4. Low Power Controls and Behaviors

EXTERNAL INPUT	CONTROL BITS		BEHAVIOR					
						RECOVERY		
EN*	sleep	slpPwrSv	LEDS	BOOST REGULATOR GATE DRIVER	I_{VIN}	MAX RECOVERY TIME	INITITAL CALIBRATION CYCLE	SOFT START
0	x	x	OFF	OFF	<10µA	520ms	YES	YES
1	0	x	ON	ON	<20mA	NA	NA	NA
1	1	0	OFF	ON**	<10mA	520ms	YES	NA
1	1	1	OFF	OFF	<1.8mA	520ms	YES	YES

NA = Not Applicable
x = Don't care
* Taking EN from 0 to 1 returns all register bits to their factory default values.
** V_{LED} goes to $V_{OUT(MAX)}$; this state is available but not recommended.

4.27 Efficiency Optimizer Control Registers 0x84 and 0x85*

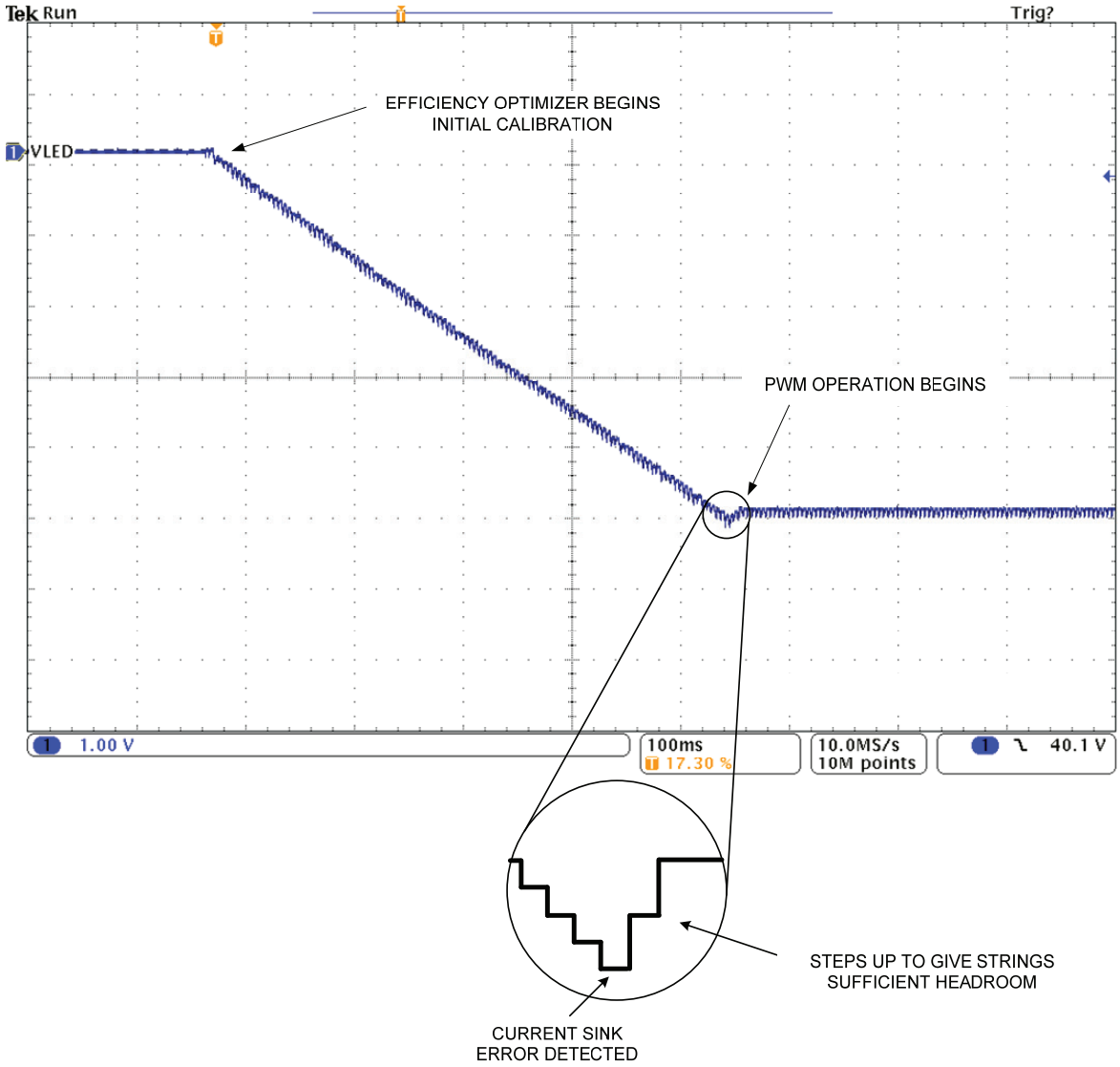
ADDRESS & REGISTER NAME		DEFAULT	D7	D6	D5	D4	D3	D2	D1	D0
0x84	eoCtrl0Rg	0x00	hdrmStep[1:0]		aCalDly[1:0]		stepDly[1:0]		iSinkConfDly[1:0]	
0x85	eoCtrl1Rg	0x02	decrStep[1:0]		incrStep[1:0]		iCaPWM	aCal100	aCalEn	iChkDis

*These registers are available only on the MSL3087 and determine how the FBO output controls the external boost regulator.

An initial power supply voltage calibration occurs when the EO is started; the EO is started at power-up, when exiting from sleep or when EN is taken high.

- **hdrmStep:** Headroom Step. Bits D6 and D7 of 0x84 set the number of steps the EO takes when raising the boost regulator output voltage after detecting a current sink error during any calibration cycle; 0b00 = 6 steps, 0b01 = 3 steps, 0b10 = 9 steps and 0b11 = 12 steps. Fewer steps improves efficiency, while more steps offers better immunity from power supply transients and noise. incrStep, bits D4 and D5 of 0x85, controls the FBO current change of each step.
- **aCalDly:** Auto-Calibration Delay. Bits D4 and D5 of 0x84 set the time delay between consecutive EO auto-recalibration cycles; 0b00 = 1s, 0b01 = 0.5s, 0b10 = 2s and 0b11 = 4s.
- **stepDly:** Step Delay. Bits D2 and D3 of 0x84 set the delay time between consecutive boost regulator output voltage corrections; 0b00 = 2ms, 0b01 = 1ms, 0b10 = 4ms and 0b11 = 8ms. Set the correction delay to longer than the power supply settling time.
- **iSinkConfDly:** Current Sink Confirmation Delay. Bits D0 and D1 of 0x84 set the current sink error confirmation delay time; 0b00 = 0.5µs, 0b01 = 0.25µs, 0b10 = 1µs, 0b11 = 2µs. The current sink error confirmation delay is the time that the current sink error must persist before the EO recognizes it as a current sink error condition. This prevents erroneous current sink error detection due to noise and other transients. The driver uses a current sync error to determine that LED string voltage is too low and needs to be increased to maintain current regulation.
- **decrStep:** Decrement Step. Bits D6 and D7 control the size of the EO voltage control current changes that happen during Efficiency Optimizer initial calibration; 0b00 = 3 LSBs, 0b01 = 1 LSB, 0b10 = 2 LSBs and 0b11 = 4 LSB. Each current change equals the decrStep[1:0] value multiplied by the 1.1µA FBO DAC LSB current value. For example, if decrStep[1:0] = 0b10 then the step setting is 2 LSBs, and each current step is 2 * 1.1µA = 2.2µA. The change in V_{LED} for each step is equal to the change in EO voltage control current times the value of the top resistor in the boost regulator output voltage divider, R_{TOP} (see datasheet for details).
- **incrStep:** Increment Step. Bits D4 and D5 of 0x85 control the size of the EO voltage control current correction steps that it takes after a current sink error is detected. Each correction step current size equals the decimal equivalent of incrStep[1:0] plus 1 multiplied by the 1.1µA FBO DAC LSB current value. For example, if incrStep[1:0] = 0b10 then the step setting is 3, and each current step is 3 * 1.1µA = 3.3µA. The number of correction steps taken is controlled by hdrmStep, bits D6 and D7 of 0x84. The change in V_{LED} for each step is equal to the change in EO voltage control current times the value of the top resistor in the boost regulator output voltage divider, R_{TOP} (see datasheet for details).
- **iCalPWM:** Initialize Calibration with PWM. Set to 1 to use PWM dimming during EO initial calibration. Clear to 0 to use 100% duty cycle during EO initial calibration; when initial calibration is complete the driver begins PWM dimming.
- **aCal100:** Auto Re-Calibration at 100% Duty Cycle. Bit D2 of 0x85; Set to 1 to use 100% duty cycle during auto recalibration. Clear to 0 to use the user controlled PWM dimming during auto recalibration.
- **aCalEn:** Automatic Re-Calibration Enable. Bit D1 of 0x85; Set to 1 to enable automatic EO boost regulator output voltage recalibration, Clear to 0 to not auto-recalibrate after initial calibration is complete. EO automatic recalibration periodically minimizes the boost regulator output voltage after initial calibration to maintain optimal efficiency as the LED string voltage changes with time or temperature. aCalDly bits D4 and D5 of 0x84 set the delay time between consecutive EO recalibration cycles. Regardless of the aCalEn setting, the EO always maintains sufficient LED string current sink headroom by raising the boost regulator output voltage should a current regulation error occur.
- **iChkDis:** Current Check Disable. Bit D0 of 0x85; Set to 1 to disable current sink error detection for the EO, Clear to 0 to enable current sink error detection. The EO uses a current sync error to determine that LED string voltage is too low and needs to be increased to maintain current regulation. Use iChkDis as a debugging or testing tool, not for final production code. Clear eoEn (bit D6 in register 0x01) to 0 before setting iChkDis to 1. When iChkDis = 1 and eoEn = 1 the EO output forces the string power supply to its minimum output voltage, possibly causing fictitious string short circuit faults. Setting iChkDis = 1 disables open circuit fault detection of all LED strings.

Figure 4.11. Efficiency Optimizer Initial Calibration Cycle



Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (+81)(3) 3523-3551
Fax: (+81)(3) 3523-7581