# 48-Channel LED PWM Generator with 12-Bit Resolution and 50MHz Cascadable Serial Interface

## FEATURES

- **3V to 5.5V Input Voltage**
- **48 Independent PWM Outputs**
- **TTL/CMOS Logic 50MHz Serial Data Interface**
- **12-Bit (4096 Steps) PWM Width Resolution**
- **6-Bit (64 Steps) PWM Correction (±50% of Programmed PWM Width)**
- **Up to 6.1kHz PWM Frequency (PWMCK = 25MHz)**
- **Phase-Shift Option Reduces Switching Noise**
- **Directly Controls Three LT3595A 16-Channel LED Drivers**
- Diagnostic Information: Sync Error/Open LED Flags
- 56-Pin (5mm × 9mm × 0.75mm) QFN Package

## APPLICATIONS

- Large Screen Display LED Backlighting
- Mono-, Multi-, Full-Color LED Displays
- LED Billboards and Signboards
- Motor Control
- Industrial Control
- Automated Test Equipment
- Robotics

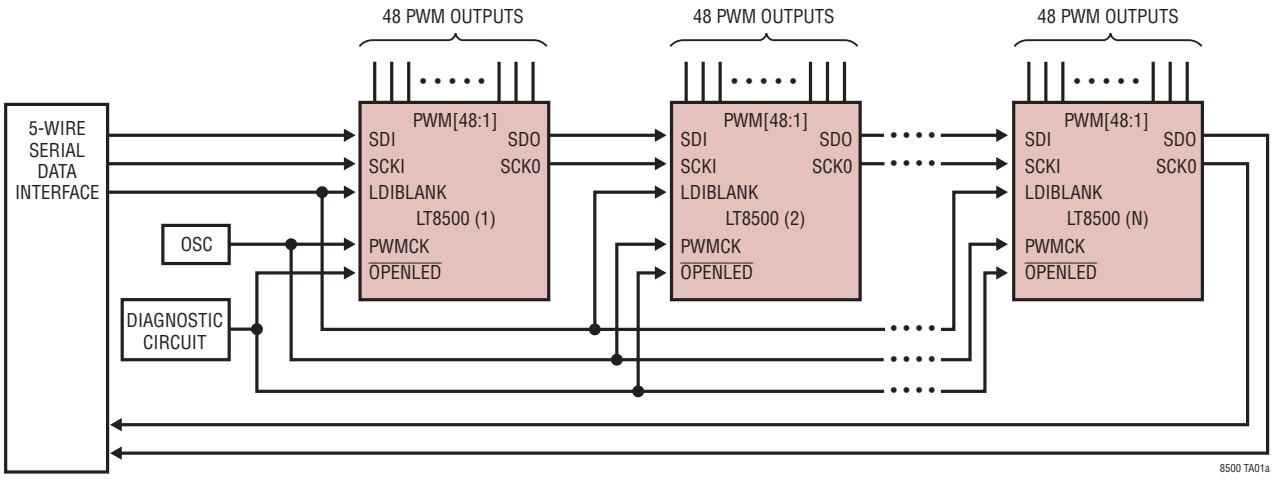
## DESCRIPTION

The LT®8500 is a pulse width modulation (PWM) generator with 48 independent channels. Each channel has an individually adjustable 12-bit (4096-step) PWM register and a 6-bit (64-step) ±50% correction register. All controls are programmable via a simple serial data interface. Three banks of 16-channels each can be configured such that they operate 120° out-of-phase with each other.

The LT8500 features two diagnostic information flags: synchronization error and open LED. The flags are sent, with additional state information, on the serial data interface during status read back. The 50MHz cascadable serial data interface includes buffering and skew-balancing, making the chip suitable for PWM intensive applications such as large screen LCD dynamic backlighting and mono-, multi- and full-color LED displays. The LT8500 is also ideally suited to control three LT3595A LED drivers.



## TYPICAL APPLICATION

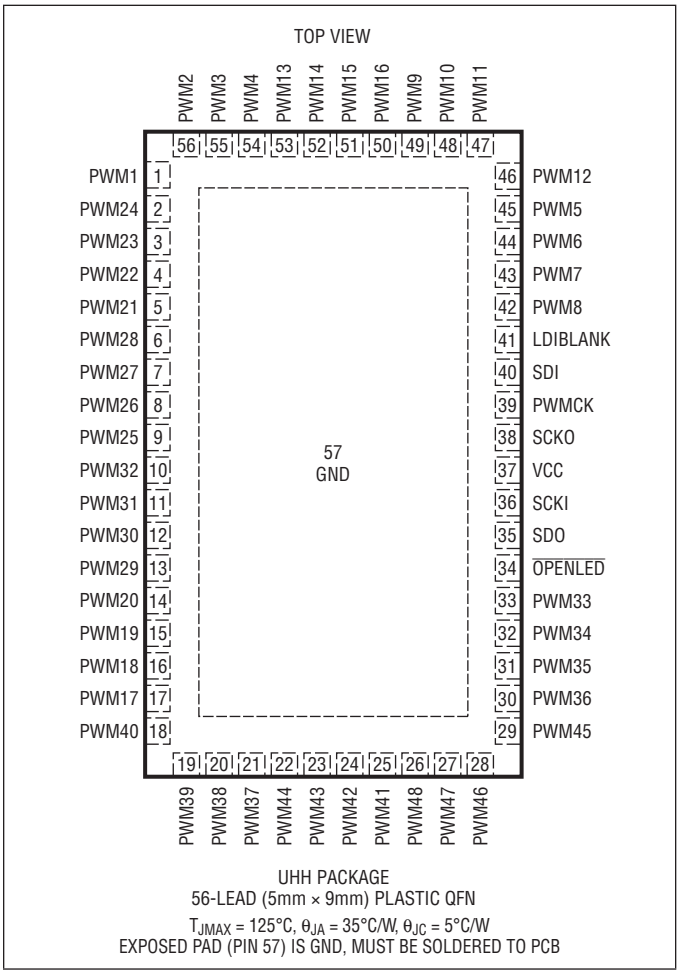


## ABSOLUTE MAXIMUM RATINGS

**(Note 1)**

$V_{CC}$ ..... –0.3V to 6V
SDI, SCKI, PWMCK, $\overline{OPENLED}$,
LDIBLANK..... –0.3V to Lesser of 6V and ($V_{CC}$ + 0.3V)
Operating Junction Temperature Range
(Note 2)..... –40°C to 125°C
Storage Temperature Range ..... –65°C to 150°C

## PIN CONFIGURATION




UHH PACKAGE
56-LEAD (5mm × 9mm) PLASTIC QFN
$T_{JMAX}$ = 125°C, $\theta_{JA}$ = 35°C/W, $\theta_{JC}$ = 5°C/W
EXPOSED PAD (PIN 57) IS GND, MUST BE SOLDERED TO PCB

## ORDER INFORMATION

| LEAD FREE FINISH | TAPE AND REEL | PART MARKING* | PACKAGE DESCRIPTION | TEMPERATURE RANGE |
|---|---|---|---|---|
| LT8500EUHH#PBF | LT8500EUHH#TRPBF | 8500 | 56-Lead (5mm × 9mm) Plastic QFN | –40°C to 125°C |
| LT8500IUHH#PBF | LT8500IUHH#TRPBF | 8500 | 56-Lead (5mm × 9mm) Plastic QFN | –40°C to 125°C |

Consult LTC Marketing for parts specified with wider operating temperature ranges. *The temperature grade is identified by a label on the shipping container.
Consult LTC Marketing for information on non-standard lead based finish parts.

For more information on lead free part marking, go to: http://www.linear.com/leadfree/
For more information on tape and reel specifications, go to: http://www.linear.com/tapeandreel/

## ELECTRICAL CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{CC}$ = 3.3V, unless otherwise noted.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| **Supply** | | | | | | | |
| $V_{CC}$ | $V_{CC}$ Operating Voltage | | ● | 3.0 | | 5.5 | V |
| **Digital Inputs: SCKI, SDI, LDIBLANK, $\overline{\text{OPENLED}}$, PWMCK** | | | | | | | |
| | Input Logic Levels | | | | | | |
| $V_{IH}$ | High Level Voltage | $V_{CC}$ = 5V | ● | 4.0 | | | V |
| | | $V_{CC}$ = 3.3V | ● | 2.7 | | | V |
| $V_{IL}$ | Low Level Voltage | $V_{CC}$ = 5V | ● | | | 1.0 | V |
| | | $V_{CC}$ = 3.3V | ● | | | 0.6 | V |
| $I_{IN}$ | Input Current | Pin Voltage = $V_{CC}$ or GND Excluding $\overline{\text{OPENLED}}$ | | –1 | | 1 | µA |
| $R_{PU}$ | $\overline{\text{OPENLED}}$ Pull-Up Resistor | $V_{CC}$ = 5.5V | | 70 | 100 | 130 | kΩ |
| $C_{IN}$ | Input Capacitance (Note 4) | Pin to GND | | | 3 | | pF |
| **Digital Outputs: SCKO, SDO, PWM[48:1]** | | | | | | | |
| | SDO, SCKO Output Voltages | | | | | | |
| $V_{OH}$ | High Level Voltage | $I_{OUT}$ = –6mA, $V_{CC}$ = 5V | ● | 4.0 | | | V |
| | | $I_{OUT}$ = –3mA, $V_{CC}$ = 3.3V | ● | 2.7 | | | V |
| $V_{OL}$ | Low Level Voltage | $I_{OUT}$ = 6mA, $V_{CC}$ = 5V | ● | | | 1.0 | V |
| | | $I_{OUT}$ = 3mA, $V_{CC}$ = 3.3V | ● | | | 0.6 | V |
| | PWM [48:1] Output Voltages | | | | | | |
| $V_{OH}$ | High Level Voltage | $I_{OUT}$ = –3mA, $V_{CC}$ = 5V | ● | 4.0 | | | V |
| | | $I_{OUT}$ = –1.5mA, $V_{CC}$ = 3.3V | ● | 2.7 | | | V |
| $V_{OL}$ | Low Level Voltage | $I_{OUT}$ = 3mA, $V_{CC}$ = 5V | ● | | | 1.0 | V |
| | | $I_{OUT}$ = 1.5mA, $V_{CC}$ = 3.3V | ● | | | 0.6 | V |

## TIMING CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{CC}$ = 3.3V, and all inputs are rail-to-rail unless otherwise noted.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $f_{SCKI}$ | Data Shift Clock Frequency | SCKI ↑ – SCKI ↑ (Figure 4) | ● | | | 50 | MHz |
| $f_{PWMCK}$ | PWMCK Clock Frequency | PWMCK ↑ – PWMCK ↑ (Figure 5) | ● | | | 25 | MHz |
| $t_{WH-SCKI}$ | Minimum SCKI High Time (Note 3) | SCKI = High | | | 2 | | ns |
| $t_{WL-SCKI}$ | Minimum SCKI Low Time (Note 3) | SCKI = Low | | | 2 | | ns |
| $t_{WH-LDI}$ | LDIBLANK Pulse Duration (LDI Function) | LDIBLANK = High (Figure 4) | ● | 8 | | 5,000 | ns |
| $t_{WH-BLANK}$ | LDIBLANK Pulse Duration (BLANK Function) | LDIBLANK = High (Figure 4) | ● | 50,000 | | | ns |
| $t_{SU-SDI}$ | SDI-SCKI Setup Time (Note 3) | SDI ↑↓ – SCKI ↑ (Figure 4) | ● | 3 | | | ns |
| $t_{HD-SDI}$ | SCKI-SDI Hold Time (Note 3) | SCKI ↑ – SDI ↑↓ (Figure 4) | ● | 1.75 | | | ns |
| $t_{SU-LDI}$ | SCKI-LDIBLANK Setup Time (Note 3) | SCKI ↑ – LDIBLANK ↑ (Figure 4)<br>SCKI 50% Duty Cycle | ● | 10 | | | ns |
| $t_{HD-LDI}$ | LDIBLANK-SCKI Hold Time (Note 3) | LDIBLANK ↓ – SCKI ↑ (Figure 4) | ● | 5 | | | ns |
| $t_{PD-SDO}$ | SCKI-SDO Propagation Delay (Note 3) | SCKI ↑ – SDO ↑↓ (Figure 4) | ● | | 15 | 25 | ns |
| $t_{PD-SCK}$ | SCKI-SCKO Propagation Delay (Note 3) | SCKI ↑ – SCKO ↑ (Figure 4) | ● | | 10 | 20 | ns |
| $t_{HD-SDO}$ | SCKO-SDO Hold Time (Note 3) | SCKO ↑ – SDO ↑↓ (Figure 4) | ● | 2.75 | | | ns |
| $t_{DC-SCK}$ | SCKI-SCKO Duty Cycle Change (Note 4) | Difference Between SCKI = High Time and SCKO = High Time, $C_{LOAD}$ = 25pF | | | –0.2 | | ns |

## TIMING CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{CC}$ = 3.3V, and all inputs are rail-to-rail unless otherwise noted.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $t_{PD-PWM}$ | PWMCK-PWM[48:1] Propagation Delay (Note 3) | PWMCK ↑ – PWM ↑↓ (Figure 5) | ● | | 32 | 50 | ns |
| $t_{R-SDO}$ | SDO, SCKO Rise Time (Note 4) | $C_{LOAD}$ = 25pF, 30% to 70% | | | 2 | | ns |
| $t_{F-SDO}$ | SDO, SCKO Fall Time (Note 4) | $C_{LOAD}$ = 25pF, 70% to 30% | | | 2 | | ns |
| $t_{R-PWM}$ | PWM[48:1] Rise Time (Note 4) | $C_{LOAD}$ = 25pF, 30% to 70% | | | 12 | | ns |
| $t_{F-PWM}$ | PWM[48:1] Fall Time (Note 4) | $C_{LOAD}$ = 25pF, 70% to 30% | | | 12 | | ns |

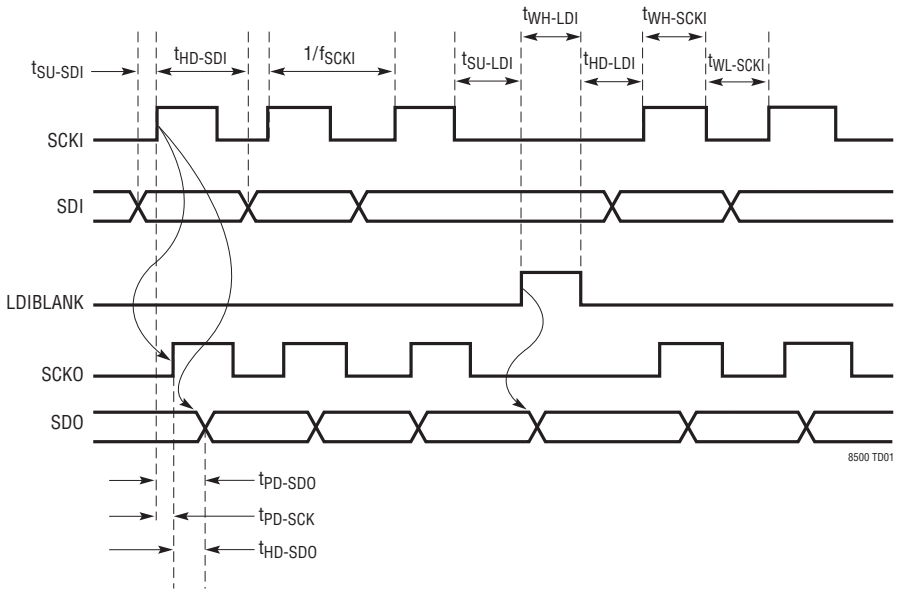
**Note 1:** Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. Exposure to any Absolute Maximum Rating condition for extended periods may affect device reliability and lifetime.

**Note 2:** The LT8500E is guaranteed to meet performance specifications from 0°C to 125°C junction temperature. Specifications over the –40°C to 125°C operating junction temperature range are assured by design, characterization and correlation with statistical process controls. The LT8500I is guaranteed over the full –40°C to 125°C operating junction temperature range.

**Note 3:** Propagation delays, setup/hold times and hi times are measured from 50% to 50%.

**Note 4:** This parameter is correlated to lab measurements and is not subject to production testing.

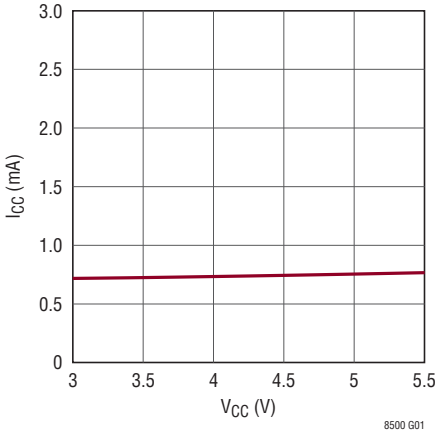
## TIMING DIAGRAM



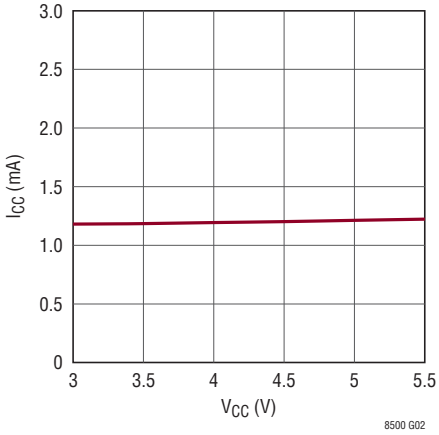
## TYPICAL PERFORMANCE CHARACTERISTICS

**For the $I_{CC}$ vs $V_{CC}$ Graphs, the Following Conditions Apply: 23pF Load on SCKO. PWM Outputs Enabled: Duty Cycle = 1365/4096, 10pF Average Load on PWMs.**

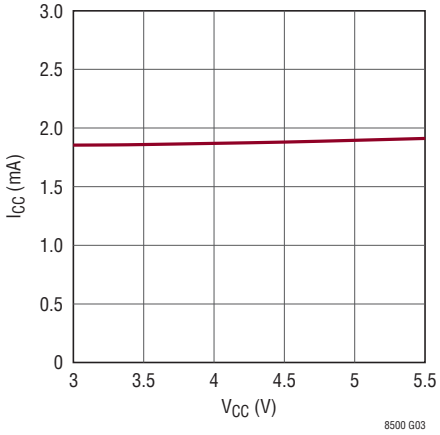
**$I_{CC}$ vs $V_{CC}$, SCKI = 0MHz, SDI = 0MHz, PWMCK = 0MHz**



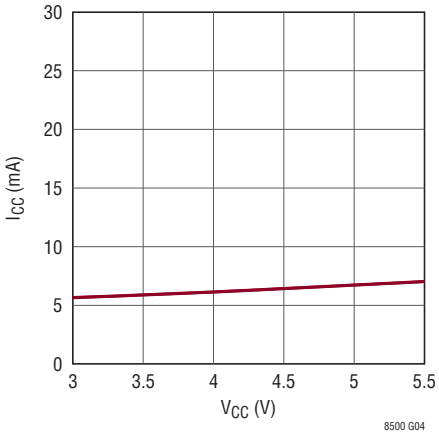

**$I_{CC}$ vs $V_{CC}$, SCKI = 0MHz, SDI = 0MHz, PWMCK = 10MHz**



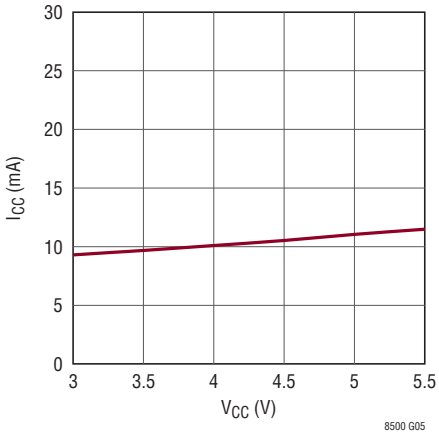

**$I_{CC}$ vs $V_{CC}$, SCKI = 0MHz, SDI = 0MHz, PWMCK = 25MHz**



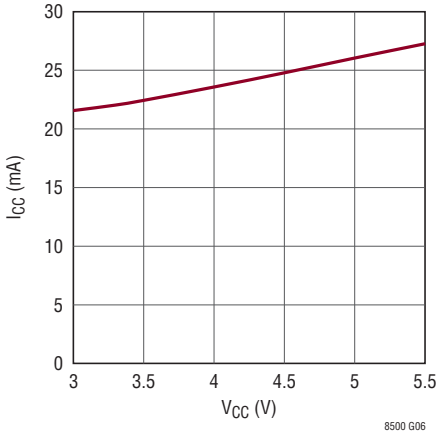

**$I_{CC}$ vs $V_{CC}$, SCKI = 12MHz, SDI = 6MHz, PWMCK = 0MHz**



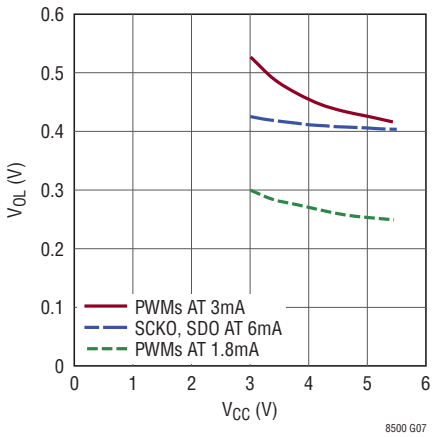

**$I_{CC}$ vs $V_{CC}$, SCKI = 20MHz, SDI = 10MHz, PWMCK = 10MHz**



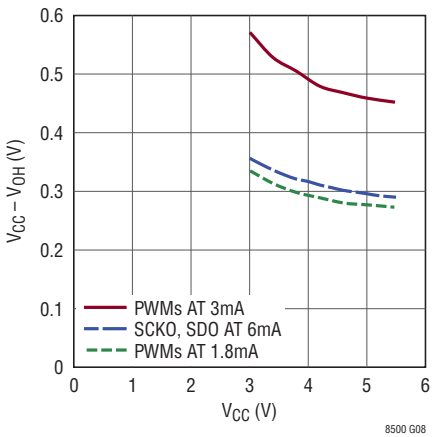

**$I_{CC}$ vs $V_{CC}$, SCKI = 50MHz, SDI = 25MHz, PWMCK = 25MHz**




**$V_{OL}$ vs $V_{CC}$**




**$V_{OH}$ vs $V_{CC}$**



## PIN FUNCTIONS

**PWM[48:1] (Pins 1 to 33, 42 to 56):** Pulse Width Modulated (PWM) Output Pins. Pulse width is determined by comparing the value in the PWMRSYNC latches to an internal PWMCK counter. Outputs are high when the value in the PWMCK counter is less than the value in PWMRSYNC[n]. The PWM frequency is determined by the signal applied to the PWMCK pin.

**$\overline{\text{OPENLED}}$ (Pin 34):** Not Open LED Input Pin. This input passes diagnostic information to the host via the status frame. When used with LT3595A LED drivers, it connects to the wired-OR (open collector) $\overline{\text{OPENLED}}$ outputs which indicate an open in one or more of the LED strings. The user can run a self test on the LT8500 to detect which PWM output is associated with an open LED string, or other fault. This pin has an internal 100kΩ pull-up to the $V_{CC}$ supply rail.

**SDO (Pin 35):** Serial Data Output Pin. This pin is the output of the shift register (SR), and cascades data to downstream chips or returns data to the host.

**SCKI (Pin 36):** Serial Clock Input Pin. This clock pin provides timing for the serial interface and the calculation of PWM values in the correction multiplier. This clock is independent of PWMCK.

**$V_{CC}$ (Pin 37):** Supply Pin. 3.0V to 5.5V. Must be locally bypassed with a capacitor to ground.

**SCKO (Pin 38):** Serial Clock Output Pin. Buffered pass through of SCKI. This pin cascades the clock to the next chip or to the host.

**PWMCK (Pin 39):** PWM Clock Input Pin. This pin provides PWM timing for the outputs. Each PWM signal is generated by counting the pulses on this clock from zero to the calculated value in the PWM synchronization register (PWMRSYNC). This clock is independent of SCKI.

**SDI (Pin 40):** Serial Data Input Pin. This pin provides serial interface data to issue commands and set up the individual PWM channels.

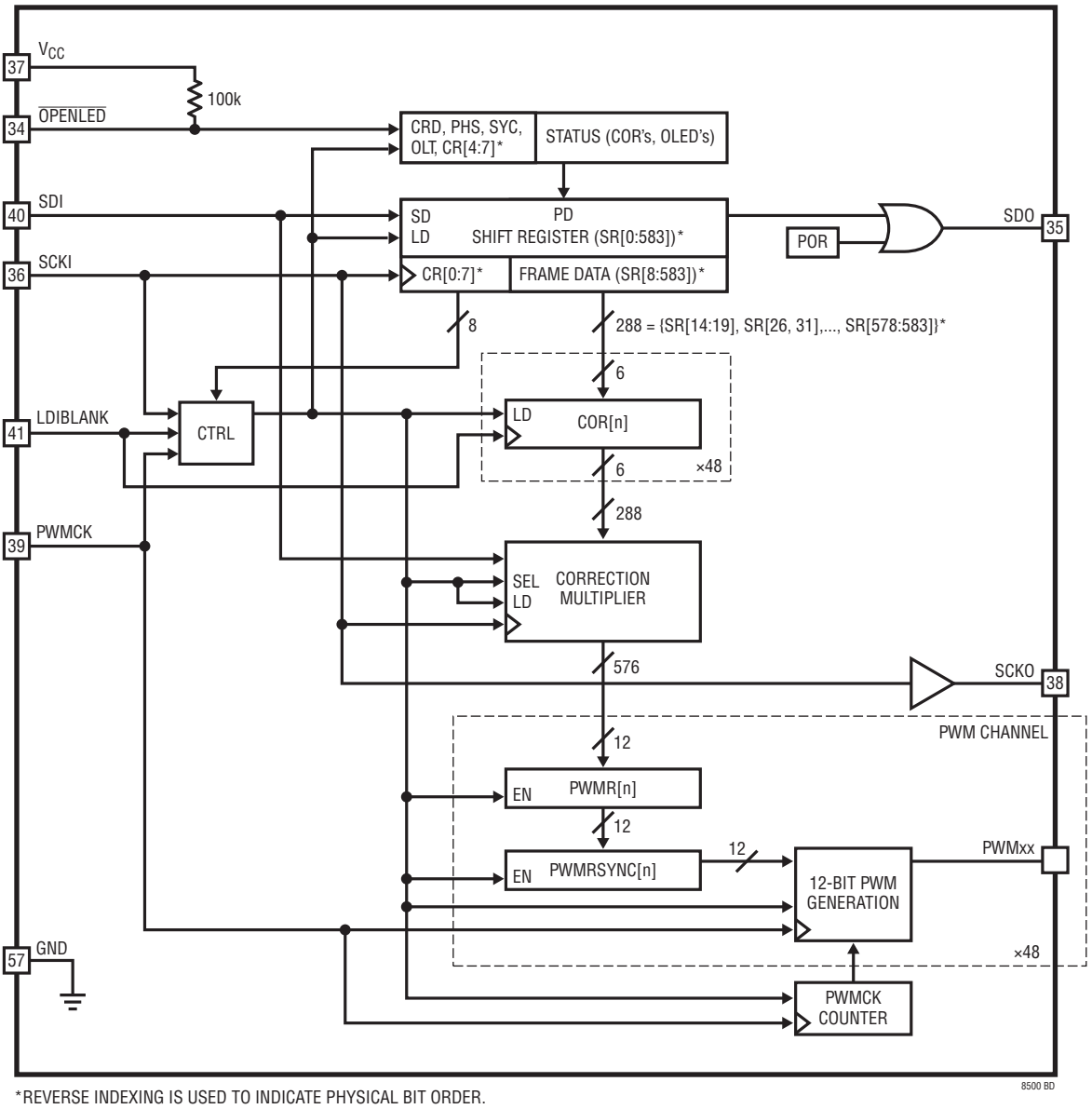
**LDIBLANK (Pin 41):** Latch Data In/Blank Input Pin. This is a dual function pin.

**LDI Function:** The internal LDI signal is directly connected to the LDIBLANK pin. A logic high on the pin always asserts the LDI function. The rising edge of LDIBLANK captures the decoded command field (CMD, CR[7:0]) of the shift register (SR[7:0]). The high level of LDIBLANK latches data from the correction multiplier into the PWM Registers (PWMR). When LDIBLANK is high, status information is loaded into the shift register (SR) to shift out on SDO when the next frame shifts in on SDI. See more details in the Operation section.

**BLANK Function:** Asserting LDIBLANK high for more than 50µs turns off all PWM[48:1] outputs and resets the chip. To avoid inadvertently resetting the chip, do not assert LDIBLANK high for more than 5µs.

**GND (Exposed Pad Pin 57):** This is the ground reference for the chip.

# BLOCK DIAGRAM




*REVERSE INDEXING IS USED TO INDICATE PHYSICAL BIT ORDER.

**Figure 1. Block Diagram**

## OPERATION

### OVERVIEW

The LT8500 controls 48 pulse width modulated (PWM[48:1]) outputs, suitable for control applications such as driving three LT3595A LED drivers. The chip's operation is best understood by referring to the Block Diagram in Figure 1.

The major blocks inside the LT8500 are: a 584-bit shift register (SR[0:583]), 48 6-bit correction registers (COR[1:48]), a correction multiplier, 48 PWM channels and a PWMCK clock counter. Each PWM channel stores data for the associated PWMx output pin and includes a PWM register (PWMR) and a PWM synchronization register (PWMRSYNC). The lower 8 bits of the 584-bit shift register are the command register (CR[0:7]) and the rest of the shift register contains the frame data.

A comparison of a channel's PWMRSYNC register to the PWMCK counter generates the respective PWM output signal. The input of the 584-bit shift register (SR[0]) is connected to the SDI signal. SDI is also an input to the correction multiplier. The output of the 584-bit shift register (SR[583]) is connected to SDO.

The user communicates with the part by controlling the serial interface pins SDI, SCKI and LDIBLANK. A serial data frame, called a command frame, is shifted into the part on SDI using SCKI as the clock signal. At the same time, the status frame is shifted out on SDO. A rising edge on the LDIBLANK pin terminates a frame. A frame consists of a 12-bit data field for each PWM channel, followed by an 8-bit command field, totaling (12 × 48) + 8 = 584 bits. **The data is transmitted with the most significant channel first, and each field is transmitted MSB first.** The frame formats and timing are illustrated in Figures 3 and 4, respectively. There are eight commands, two of which update the PWM[48:1] outputs. The commands are summarized in Table 2. Within this document, command frames will be referred to by the commands they issue, such as "update frame" or "correction frame."

With a 50MHz SCKI, a single frame can be transmitted in 11.7µs (584 SCKIs + LDI), for a frame rate of 85.5kHz. A 25MHz PWMCK creates a PWM period (4096 PWMCKs) of 164µs, or a PWM output frequency of 6.1kHz.

Update frames are used to serially load the 12-bit values for each of the 48 PWM channels. The LT8500 contains a correction multiplier that can automatically scale the 12-bit PWM channel data before it's stored. By default, the correction multiplier is enabled and scales incoming channel data according to:

$$PWM_{OUTn} = CHAN_{n(NOM)} \bullet \left(\frac{2}{3}\right) \bullet \left(\frac{COR_n + 32}{64}\right)$$

where $PWM_{OUTn}$ is the number of PWMCK cycles that PWM*n* is high, $CHAN_{n(NOM)}$ is the *n*th channel field in the frame, and COR*n* is the *n*th programmed correction setting (COR*n* = 0 to 63). See Table 1 for examples.

Otherwise, when the correction multiplier is disabled, the incoming data is stored unchanged:

$$PWM_{OUTn} = CHAN_{n(NOM)}$$

The correction multiplier is disabled by the correction register disable bit (CRD), which is toggled by the correction toggle command (CMD = 0x7X). By default, the correction multiplier is enabled after power-up and the CRD bit is low.

The result generated by the correction multiplier moves to the respective PWMRSYNC register after an update frame. An update frame does this either synchronously or asynchronously. A synchronous update frame will copy the data to the PWMR on the subsequent rising edge of LDI which marks the end of the frame, and then from the PWMR to the PWMRSYNC register at the beginning of a PWM period. A PWM period starts when the free-running PWMCK counter is zero. Otherwise, the asynchronous update frame will copy the data from the correction multiplier, through the PWMR to the PWMRSYNC at the same time, on the subsequent rising edge of LDI which marks the end of the frame.

As soon as the PWMRSYNC registers are updated with their new values, the PWM outputs will reflect the update. As mentioned earlier, the PWMR outputs are generated by comparing the respective PWMRSYNC values to the PWMCK counter.

## OPERATION

### START-UP

The LT8500 is ready to communicate after power-up, if the LDIBLANK pin is low. The PWM[48:1] outputs remain disabled (logic 0) until an output enable frame is sent. The recommended sequence of events for start-up is:

1. Apply power and drive LDIBLANK low. SDO will go low when the on-chip power-on-reset (POR) de-asserts.
2. Send a correction register frame (CMD = 0x20) on the serial interface. This sets the correction factor on each channel.
3. Send an update frame (CMD = 0x00 or CMD = 0x10) on the serial interface. This sets the pulse width of each channel.
4. Send an output enable frame (CMD = 0x30) on the serial interface. This enables the modulated pulses on the PWM[48:1] outputs.

The PWM clock (PWMCK) should be turned on before step 4. The start of a PWM period, when all PWM[48:1] channels turn on, is synchronized to the output enable frame when the outputs are disabled prior to the frame.

### SERIAL DATA INTERFACE

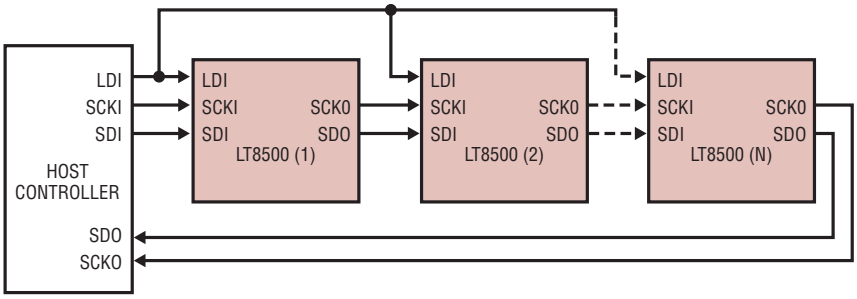
The LT8500 has a 50MHz cascadable serial data interface with full buffering and skew balancing on clock and data. The interface uses a novel 5-wire (LDIBLANK, SCKI, SDI, SCKO, and SDO) topology and can be connected to a variety of digital controllers, such as microcontrollers, digital signal processors (DSPs), or field programmable gate arrays (FPGAs).

#### Topology

Two topologies shown in Figure 2 are supported for cascading the LT8500. For higher speeds and a large number of LT8500s, consider the novel 5-wire topology. For lower speeds and few LT8500s, consider the conventional 4-wire topology. Whichever topology is used, signal integrity should be carefully evaluated, especially for the clocks.

The 5-wire topology eliminates the need for global SCKI routing and reduces the need for buffer insertion for the SCKI signal. Instead, it provides the SCKO signal along with the SDO signal to drive the next chip. The skew inside the chip between the SCKI and SDI signals is balanced internally. The skew outside the chip between the SCKO and SDO signals can be easily balanced by parallel routing

**Novel 5-Wire Topology**




**Conventional 4-Wire Topology**

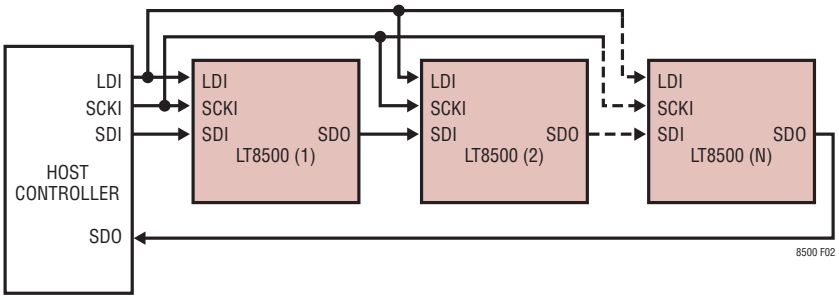



**Figure 2. Serial Interface Topologies**

## OPERATION




**COMMAND REGISTER (CR):**
SEE TABLE 2 FOR COMMAND REGISTER DECODING.

**STATUS FRAME:**
NOL48-NOL1: FAULT = 0, OK = 1
CR[7:4]: LAST COMMAND
OLT: $\overline{\text{OPENLED}}$ TESTED = 1, $\overline{\text{OPENLED}}$ NOT TESTED = 0
SYC: OUT-OF-SYNC = 1, OK = 0
PHS: PHASE-SHIFTED = 1, NOT SHIFTED = 0
CRD: CORRECTION DISABLED = 1, CORRECTION ENABLED = 0

**Figure 3. Serial Data Frame Format**

## OPERATION

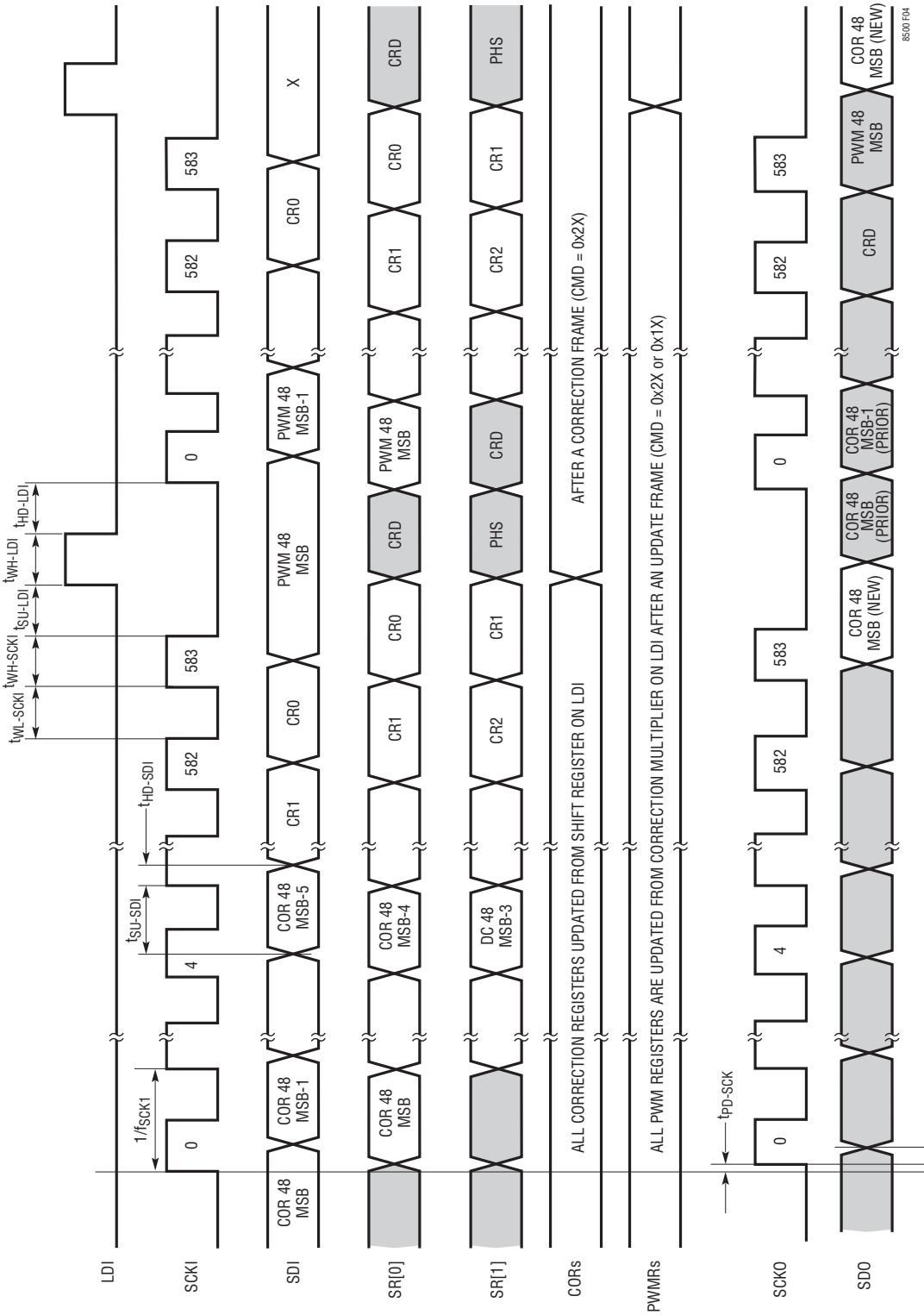



**Figure 4. Serial Data Input and Output Timing Chart**

## OPERATION

these two signals between chips. When properly balanced in this way, the SCKO/SDO timing will meet the timing requirements of SCKI/SDI on the next cascaded chip, enabling faster clock speeds and more chips in cascade. The host controller sends the SDI signal with the SCKI signal, and receives the SDO signal with the SCKO signal. The controller will see skew between SCKI and SCKO, and will need to operate on two clock planes depending on the number of cascaded LT8500s and system timing constraints. A duty cycle change ($t_{DC-SCK}$) will also occur between SCKI and SCKO, limiting the number of LT8500s in a chain, depending on SCKI speed. This change results from a slight difference in propagation delays of the positive and negative edges of SCKI. LDIBLANK skew between chips may require balancing in timing critical systems, otherwise the host should increase the delay between SCKI and LDI to avoid violating LDI to SCKI setup and hold times ($t_{SU-LDI}$ and $t_{HD-LDI}$). In summary, the 5-wire topology extends the maximum number of cascadable chips, boosts the series data interface clock frequency, eliminates global SCKI routing, reduces the need of buffer insertion for SCKI signals, and offers an easier PCB layout. In a low-speed application with a small number of cascaded chips, the 5-wire topology can be simplified to the 4-wire topology by ignoring the SCKO output.

In a 4-wire topology, the LDIBLANK and SCKI signals need global routing while the SDI signal only needs local routing between chips. SCKO is ignored. When a large number of chips are in cascade, or long board traces are used, external clock-tree buffers with corresponding driving capability might be needed for the LDIBLANK and SCKI signals to minimize signal skews. The propagation delay caused by the buffer insertion on the SCKI signal yields the skew between the SCKI and SDI signals, which usually requires balancing. Since both the SDI and SDO signals require the same SCKI signal to send and receive, the propagation delay between the SDI and SDO signals limits the number of chips in cascade and the series data interface clock frequency.

### Communication

Figure 3 shows two command frames sent on SDI, and one status frame received on SDO. All the frames have the same 584-bit length and are transmitted with the most significant channel first, and each field is transmitted with MSB first. The command frames are sent with the SCKI signal and the status frame is received with the SCKO signal. The command field determines the function of a frame, according to Table 2. The status frame consists of the four MSB's of the last command (CR[7:4]), the open LED self test bit (OLT), the synchronization error status bit (SYC), the phase-shift status bit (PHS), the correction register disable status bit (CRD), and individual $\overline{\text{OPENLED}}$ fault bits (NOL[48:1]), as well as each 6-bit correction register (COR[48:1]). Logic zeros fill in the unused bits of the status frame. Refer to Figure 3.

Figure 4 illustrates the timing relationship among serial input and serial output signals in more detail. One correction register frame followed by an update frame is sent through the SDI, SCKI, and LDIBLANK pins. At the same time, two status frames are received through the SDO, SCKO, and LDIBLANK pins. The rising edges of SCKI shift a frame of data into shift register SR[0:583]. After 584 clock cycles, all bits of data sit in the shift register waiting for the LDI signal. An asynchronous LDIBLANK "high" signal captures the decoded 8-bit CMD field (CR[7:0]), executing commands and routing data accordingly. At the same time, a frame of status information, including the 4 MSB's of the CMD field (CR[7:4]), status bits, COR registers, and individual open LED fault flags, is parallel loaded into the 584-bit shift register and will be shifted out as the next frame shifts in.

### LDIBLANK = LDI + BLANK

The LDIBLANK pin is a dual function input, determined by the duration of a logic high on the pin. LDI is the latch data input, which signals the end of a frame and executes the command in the CMD field (CR[7:0]). The BLANK signal turns off the PWM[48:1] outputs and performs a global reset of the part, including the shift register in the serial interface. A logic high on LDIBLANK always asserts LDI, while a logic high greater than the minimum LDIBLANK pulse duration for BLANK ($t_{WH-BLANK}$) also asserts BLANK. BLANK will never be asserted if the pin is held high less than the maximum LDIBLANK pulse duration for LDI ($t_{WH-LDI}$). Between maximum $t_{WH-LDI}$ and minimum $t_{WH-BLANK}$, BLANK becomes asserted at an undetermined time.

## OPERATION

A rising edge on the LDI signal is always interpreted as the end of a frame. The next rising edge of SCKI after the falling edge of LDIBLANK is always interpreted as the start of a new frame. An out-of-sync error bit (SYC) is provided in the status frame to alert the system if the part saw an LDI unexpectedly. This occurs when LDI and SCKI are both hi, or when LDI is hi on other than a frame boundary (n • 584 SCKI's). The SYC bit is for information only, it has no other effect on the part. If the SYC bit is set, none of the other data in the status frame is reliable and the effect of the prior frame is unknown; the LT8500 assumes the system's timing of the LDI is correct and considers the next SCKI as the start of the next frame.

### $\overline{\text{OPENLED}}$

The $\overline{\text{OPENLED}}$ pin provides status information to the host by reporting its state in the status frame. The state of the pin is captured by each rising edge of PWMCK and is reported in two ways. In typical use, the status frame receives the captured state of the pin on the rising edge of the first SCKI after LDIBLANK goes low. This state is duplicated 48 times and reported in the LSB of each PWM channel in the status frame. The state will normally be a logic "1" due to the on-chip pull-up resistor.

Alternatively, the LT8500 supports a diagnostic self test frame (CMD = 0x5X) that reports the $\overline{\text{OPENLED}}$ state differently. In this case, the LT8500 sequentially pulses PWM[1] through PWM[48] high for 64 PWMCK cycles each. The state of the $\overline{\text{OPENLED}}$ pin is captured for each channel while the corresponding PWM pin is high. This by-channel data is shifted out in the status frame as the next frame is shifted in. In addition, the status frame will set the open LED test bit (OLT), indicating that the $\overline{\text{OPENLED}}$ data in the current status frame is from the self test. The status frame will return to typical reporting on the following frame. When the LT8500 is used with the LT3595A, the $\overline{\text{OPENLED}}$ pin and the self test provide a diagnostic routine to identify the location of open LED faults. See "Diagnostic Information Flags" in the Applications Information section.

### OUTPUTS

After power-up or reset, no PWM[48:1] output will turn on until an output enable frame is sent. The 12-bit PWMCK counter is free-running from the PWMCK clock when outputs are enabled. When an output enable frame is sent, the PWMCK counter increments to one on the second rising edge of PWMCK after the rising edge of LDIBLANK, as shown in Figure 5. By default, all outputs with non-zero values in PWMRSYNC will turn on when the PWMCK counter is one. Alternatively, if the phase-shift bit (PHS) is set, the PWM[48:1] outputs will turn on as illustrated in the phase-shift synchronous updates in Figure 6, case A. Further discussion of the phase-shift function follows. Each subsequent rising edge of PWMCK increases the PWMCK counter by one. Any PWM channel will be turned off when its PWMRSYNC value is equal to the value in the PWMCK counter. An output disable frame resets the PWMCK counter immediately after LDI, and turns off all the PWM channels on the next rising edge of PWMCK after LDI. Figure 5 shows the PWM output enable timing chart.

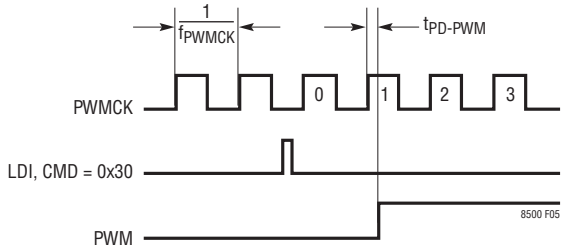



**Figure 5. PWM Output Enable Timing Chart Assumes Outputs Were Previously Disabled**

### PHASE DIFFERENCE BETWEEN 16-CHANNEL BANKS

By default, the rising edges of all PWM[48:1] channels occur on the same rising edge of PWMCK. This event begins a PWM period of 4096 PWMCK cycles. The LT8500 provides a phase-shift toggle command (CMD = 0x6X) to reduce system noise and current spikes resulting from 48 pins switching at once. The function of this command is illustrated in Figure 6, case A. In phase-shift mode, the PWM[48:1] outputs are divided into three 16-channel banks that are 120 degrees out-of-phase with each other within a PWM period. This means that channels PWM[48:33] will turn on with the rising edge of PWMCK(1), then channels PWM[32:17] will turn on with the rising edge of PWMCK(1365), 1/3 of the PWM period, and channels PWM[16:1] will turn on with the rising edge of PWMCK(2730), 2/3 of the PWM period.

## OPERATION

**Table 1. Example PWM Width Calculations (Base 10) with Correction Enabled (CRD = 0)**

| A<br>PWM UPDATE VALUE<br>SENT ON SDI | B<br>PRESCALED PWM<br>(A • 2/3) | C<br>CORRECTION REGISTER<br>(COR) VALUE | D<br>MULTIPLIER<br>(C + 32)/64 | E<br>PWM WIDTH (B•D)<br>(IN UNITS OF $t_{PWMCK}$) |
|---|---|---|---|---|
| 3 | 2 | 63 | 1.484375 | 3 |
| 120 | 80 | 63 | 1.484375 | 119 |
| 120 | 80 | 32 | 1.0 | 80 |
| 120 | 80 | 0 | 0.5 | 40 |
| 1200 | 800 | 63 | 1.484375 | 1188 |
| 1200 | 800 | 32 | 1.0 | 800 |
| 1200 | 800 | 0 | 0.5 | 400 |
| 4095 | 2730 | 63 | 1.484375 | 4052 |
| 4095 | 2730 | 32 | 1.0 | 2730 |
| 4095 | 2730 | 0 | 0.5 | 1365 |

### PWM CALCULATION BY DIGITAL MULTIPLICATION OF CORRECTION REGISTER AND PWM UPDATE VALUES

The correction multiplier is used to automatically scale the 12-bit PWM channel data before storing the PWM update value for the respective channel. The correction multiplier is disabled by the correction register disable bit (CRD), which is toggled by the correction toggle command (CMD=0x7X). When the correction multiplier is disabled, the incoming data is stored unchanged:

$$PWM_{OUTn} = CHAN_{n(NOM)}$$

The correction multiplier is enabled by default (CRD=0) and scales incoming channel data according to:

$$PWM_{OUTn} = CHAN_{n\,(NOM)} \bullet \left(\frac{2}{3}\right) \bullet \left(\frac{COR_n + 32}{64}\right)$$

where $PWM_{OUTn}$ is the number of PWMCK cycles that PWM*n* is high, $CHAN_{n(NOM)}$ is the *n*th channel field in the frame, and COR*n* is the *n*th programmed correction setting (COR*n* = 0 to 63). See Table 1 for examples.

The 6-bit COR value sets a multiplier of 0.5X to ~1.5X (exactly 1.484375, or ((63 + 32)/64)) with 64 values and a midrange, signifying a multiple of 1.0, at 32 (0x20). In order to avoid overflow in the PWM registers when the multiplier is greater than 1.0, the nominal PWM update value (CHAN*n*) is first prescaled on chip by 2/3. This means that the full-scale width for a channel with a multiplier of 1.0 (CHAN*n* = 4095, COR*n* = 32) will result in a $PWM_{OUTn}$ width of 4095 • (2/3) • 1.0 = 2730, not 4095. So, a correction multiplier of ~1.5 (COR*n* = 63) yields a corrected PWM width of 4052 = 4095 • (2/3) • 1.484375. The $PWM_{OUTn}$ width is always rounded to the nearest whole number. Table 1 shows examples of PWM calculations for selected register values. This means the maximum PWM duty cycle with CRD=0 is 4052/4096, and with CRD=1 it is 4095/4096.

### COMMAND DESCRIPTIONS

The LT8500 implements eight commands, outlined in Table 2. The commands (CMD) are encoded in the eight LSB's of a command frame, and so reside in the eight LSB's of the shift register when a frame has been completely shifted in. The command field is executed by the rising edge of LDI. Only the four MSB's of the command field are decoded for commands.

#### Synchronous Update Frame: CMD = 0x0X

A synchronous update frame updates PWM[48:1] with the data in the frame, after processing through the Correction Multiplier. The PWMR is updated when LDIBLANK goes high. The PWMRSYNC register will be written from the PWMR synchronously to the start of the PWM period (on PWMCK 1). This command eliminates shortened PWM "runt" pulses. The value in the PWMRSYNC registers will update the PWM outputs on the next rising edge of PWMCK. Examples are shown in Figure 6, cases B and E.

## OPERATION

**Table 2. Command Register Decoding**

| CMD (CR[7:0]) | NAME | SUMMARY | FRAME DATA |
|---|---|---|---|
| 0000_xxxx | Synchronous Update Frame | Update PWM's Synchronously to PWM Period | PWM Update by Channel |
| 0001_xxxx | Asynchronous Update Frame | Update PWM's Asynchronously to PWM Period | PWM Update by Channel |
| 0010_xxxx | Correction Frame | Set PWM Correction Factor | Correction by Channel |
| 0011_xxxx | Output Enable Frame | Enable PWM Outputs | Don't Care |
| 0100_xxxx | Output Disable Frame | Disable (Drive Low) PWM Outputs | Don't Care |
| 0101_xxxx | Self Test Frame | Initiates Self Test | Don't Care |
| 0110_xxxx | Phase-Shift Toggle Frame | Toggle 16-Channel Bank 120° Phase-Shift (PHS) | Don't Care |
| 0111_xxxx | Correction Toggle Frame | Toggle Correction Disable Bit in Multiplier (CRD) | Don't Care |
| 1xxx_xxxx | Reserved | Do Not Use | – |

### Asynchronous Update Frame: CMD = 0x1X

An asynchronous update frame updates PWM[48:1] with the data in the frame, after processing through the correction multiplier. The PWMR is updated when LDIBLANK goes high. The PWMRSYNC register will be written immediately (asynchronously), through the PWMR, when LDI is high. The value in the PWMRSYNC registers will update the PWM outputs on the next rising edge of PWMCK. Examples are shown in Figure 6, cases C and F.

### Correction Frame: CMD = 0x2X

A correction frame updates the correction registers (COR) with the six MSB's of each channel's data field in the frame. The CORs are used by the correction multiplier to adjust the PWM width, prescaled by 2/3, by a multiplier of between 0.5 and ~1.5. Example PWM width calculations are shown in Table 1. In typical applications, this command will only be run once after power-up to initialize the system. Therefore, a correction frame will not update the PWM outputs. The update frame that follows a correction frame will reflect the COR update.

### Output Enable Frame: CMD = 0x3X

An output enable frame starts a PWM period, and enables the PWM outputs, on the second PWMCK edge after LDIBLANK goes high. There is no effect on either SDO or SCKO. The data in the output enable frame is irrelevant to the command, but allows a daisy chain of LT8500's to function properly.

### Output Disable Frame: CMD = 0x4X

An output disable frame immediately resets the PWMCK counter when LDI goes high, and disables the PWM outputs on the next rising edge of PWMCK. There is no effect on either SDO or SCKO. The data in the output disable frame is irrelevant to the command, but allows a daisy chain of LT8500's to function properly.

### Self Test Frame: CMD = 0x5X

The self test frame can be used for diagnostics on each PWM[48:1], including identifying open LED strings on an LT3595A. After LDIBLANK goes hi, the LT8500 pulses PWM[1] through PWM[48] sequentially for 64 PWMCK cycles each. The state of the $\overline{\text{OPENLED}}$ pin is captured for each channel while the corresponding PWM pin is high. This by-channel data is subsequently shifted out in the status frame. In addition, the status frame will set the open LED test bit (OLT) to confirm that the $\overline{\text{OPENLED}}$ data in the current status frame is from the self test. For all other commands, the state of the $\overline{\text{OPENLED}}$ pin is captured once on the first SCKI of the frame. The same value is then reported in the status frame on all 48 channels. The data in the self test frame is irrelevant to the command, but allows a daisy chain of LT8500's to function properly.

## OPERATION

### Phase-Shift Toggle Frame: CMD = 0x6X

The phase-shift toggle frame toggles the phase-shift (PHS) bit, which is off by default. When PHS is set, it sets the rising edges of the PWM outputs, by banks of 16 channels, out-of-phase with each other by 120 degrees. This means that channels PWM[48:33] will start the PWM cycle with a rising edge at the beginning of a PWM period, then channels PWM[32:17] will start their PWM cycle 1/3 of the time into a PWM period, and channels PWM[16:1] will start 2/3 of the time into a PWM period. The state of the PHS bit is returned in every status frame. The data in the phase-shift toggle frame is irrelevant to the command, but allows a daisy chain of LT8500's to function properly.

### Correction Toggle Frame: CMD = 0x7X

The correction toggle frame toggles the correction register disable (CRD) bit, which is off by default. When CRD is set, it disables use of the correction registers (CORs) in the correction multiplier, instead multiplying the incoming data from SDI by "1." This causes the data in an update frame to reach the PWMRSYNC registers unchanged. The state of the CRD bit is returned in every status frame. The data in the correction toggle frame is irrelevant to the command, but allows a daisy chain of LT8500's to function properly.

### Examples of PWM Updates for Selected Cases

Figure 6 shows examples of the effect of various commands on the PWM output waveforms. These example waveforms assume all three channels shown are always programmed for the same PWM width. For each case, a representative channel is shown from each of the three 16 channel banks, PWM[48:33], PWM[32:17], and PWM[16:1].

Case A illustrates the phase-shift mode in steady-state, with PWM's programmed for a width of 256 PWMCK cycles. PWM[48], from bank 2, rises at the beginning of the PWM period. PWM[32], from bank 1, rises 1/3 of the way into the PWM period of bank 2, or 1365 PWMCK cycles later. PWM[16], from bank 0, rises 2/3 of the way into the PWM period of bank 2, or 2730 PWMCK cycles later.

Case B illustrates a synchronous update frame (CMD = 0x0X) while in phase-shift mode, as in case A. The LDI signal goes active 512 PWMCK cycles into the PWM period, after PWM[48] has turned off. The update frame programs a PWM width of 1024, but the synchronous update command prevents a channel from updating except at the beginning of its PWM period. As a result, PWM[48] remains low until the next PWM period, when the updated width drives it high for 1024 PWMCK cycles. PWM[32] begins its PWM period at PWMCK 1365, and PWM[16] starts at PWMCK 2730, both updated to 1024 PWMCK cycles.

Case C illustrates an asynchronous update frame (CMD = 0x1X) while in phase-shift mode, as in case A. The LDI signal goes active 512 PWMCK cycles into the PWM period, after PWM[48] has turned off. The update frame programs a PWM width of 1024, and because it is an asynchronous update, PWM[48] immediately rises and stays high until PWMCK 1024. PWM[32] and PWM[16] (and all PWM's) are also updated, but no rising edge occurs until their PWM period begins due to the phase-shifting.

Case D illustrates the default (not phase-shifted) mode in steady-state. All PWM outputs rise on the same PWMCK edge at the beginning of the PWM period.

Case E illustrates a synchronous update frame (CMD = 0x0X) without phase-shifting, as in case D. The LDI signal goes active 512 PWMCK cycles into the PWM period, after the PWMs have turned off. The update programs a PWM width of 1024, but the synchronous update command prevents a channel from updating except at the beginning of it's PWM period. As a result, all PWM's remain low until the next PWM period, when the updated width drives them high for 1024 PWMCK cycles.

Case F illustrates an asynchronous update frame (CMD = 0x1X) without phase-shifting, as in case D. The LDI signal goes active 512. PWMCK cycles into the PWM period, after the PWMs have turned off. The update programs a PWM width of 1024, and because it is an asynchronous update, all PWM's immediately rise and stay high until PWMCK 1024.

## OPERATION

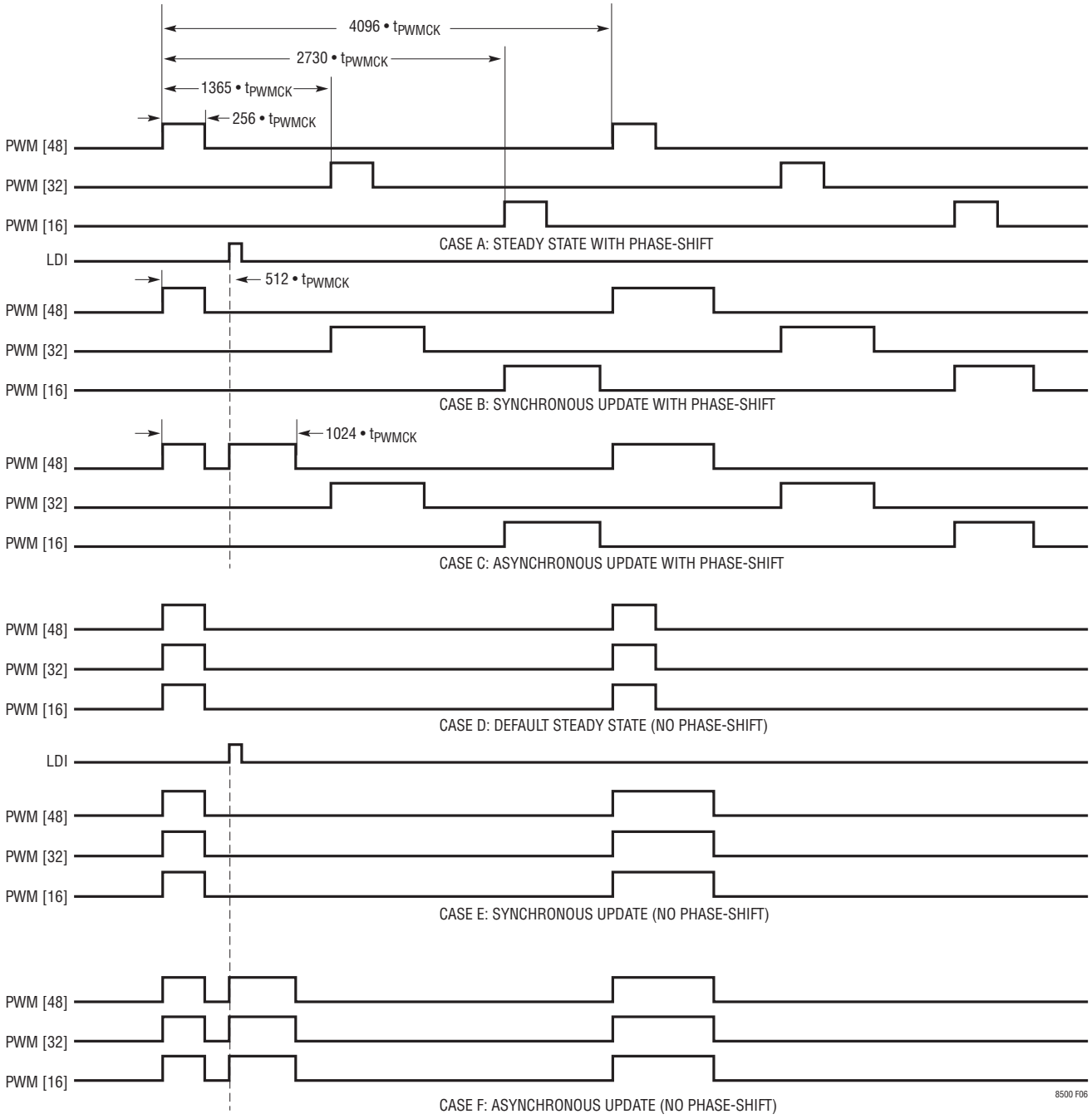



**Figure 6. Examples of PWM Outputs For Selected Command Cases**

## APPLICATIONS INFORMATION

This section is illustrated with an LED dimming application, but is relevant to other applications as well. The LT8500 provides 48 PWM outputs, such as for driving three LT3595A LED drivers. The LT8500 provides an LED dot correction function using digital multiplication of the correction register (COR) and the PWM update value, which is prescaled by 2/3. This results in a dot corrected PWM duty cycle. Optionally, the PWM update can be written directly (unchanged) by setting the correction register disable bit (CMD = 0x7X). When this bit is set, the multiplication is bypassed and dot correction, if any, must be calculated off-chip. The PWM duty cycle in this case will be the nominal value sent in the update frame, divided by 4096. The part provides a status frame with $\overline{\text{OPENLED}}$ and COR data for each channel, and global state data indicating self testing (such as for open LED's), out-of-sync error, phase-shift status, and direct data status. The status frame is shifted out of the part whenever a new frame is shifted in. An on-chip self test is available (CMD = 0x5X) to determine which channel is responsible for a fault, such as open LEDs. The $\overline{\text{OPENLED}}$ pin and self test are especially suited for use with the LT3595A. In this application, the self test will identify which channels have opens in their LED strings. This Applications Information section serves as a guideline for avoiding common pitfalls for the typical application.

### Setting Grayscale by PWM Updates

Although adjusting the LED current changes its luminous intensity, or brightness, it will also affect the color matching between LED channels by shifting the chromaticity coordinate. The best way to adjust the brightness is to control the amount of LED on/off time by pulse width modulation (PWM).

The LT8500 can adjust the brightness for each channel independently. The 12-bit PWM registers (PWMR), used for grayscale (GS) dimming, results in 4095 different brightness steps from 0% to 99.98%. The brightness level, or PWM duty cycle, GS*n*% for channel *n* can be calculated as:

$$GSn\% = \frac{GSR_{n(CALC)}}{4096} \bullet 100\%$$

where $GSR_{n(CALC)}$ is the nth calculated grayscale register (same as PWMR) setting ($GSR_{n(CALC)}$ = 0 to 4052 with dot correction enabled).

### Setting Dot Correction

The LT8500 can adjust the PWM duty cycle for each channel independently. The duty cycle adjustment, also called dot correction, is mainly used to calibrate the brightness deviation between LED channels. The 6-bit (64 values) dot correction registers (DCR, same as COR) adjust each PWM duty cycle from 0.5X to ~1.5X of the duty cycle, prescaled by 2/3, sent to the grayscale register (GSR) according to

$$PWM_{OUTn} = CHAN_{n(NOM)} \bullet \left(\frac{2}{3}\right) \bullet \left(\frac{COR_n + 32}{64}\right)$$

where $PWM_{OUTn}$ is the nth PWM duty cycle, $GSR_{n(NOM)}$ is the nominal grayscale value sent to the nth channel and $DCR_n$ is the nth programmed dot correction setting ($DCR_n$ = 0 to 63).

### Cascading Devices and Determining Serial Data Interface Clock

In a large LCD backlighting or LED display system, multiple LT8500 chips can be easily cascaded to drive all LED drivers, such as the LT3595A, and their associated LED strings. The LT8500 adopts a novel 5-wire topology, which balances clock skew and eases PCB layout.

The time required to send a set of cascaded frames is 584 SCKI cycles per LT8500, plus another cycle time for LDI. Assuming LDI is externally balanced, the minimum serial data interface clock frequency $f_{SCK}$ for a large display system can be calculated as:

$$f_{SCK} = [(n_{CHIPS} \bullet 584) + 1] \bullet f_{REFRESH}$$

where $n_{CHIPS}$ is the number of cascaded LT8500s and $f_{REFRESH}$ is the refresh rate of the whole system.

### Status Frame Information

The status frame is captured and shifted out of SDO as a new data frame shifts in on SDI. The format of a status frame is shown in Figure 5. With the exception of the diagnostic flags (SYC and NOL[48:1]), the data in the status frame does not change without a command from

## APPLICATIONS INFORMATION

the user interface. It can therefore be monitored to confirm proper communication with the chip. The following non-diagnostic status information is continually provided in the status frame: dot correct registers for each channel (COR[48:1]), Open LED Testing bit (OLT), phase-shift bit (PHS), correction register disable (CRD) bit. There are five unused bits, [5:1], in the field associated with each channel, all of which are always set to logic zero.

### Diagnostic Information Flags

The LT8500 features two kinds of diagnostic information flags: global out-of-sync error (SYC) and 48 individual open LED flags (NOL[48:1]).

An out-of-sync error occurs when the part sees an LDI signal unexpectedly, whether before 584 SCKI clocks, or coinciding with SCKI high. Either of these events can corrupt the data and the state of the chip. The SYC bit is available in every status frame to notify the system if an erroneous LDI was seen since the first rising edge of SCKI of the last frame. A series of multiple LDI's between frames, with no SCKI, is not an out-of-sync error. Recovery from an out-of-sync error may require the user to completely rewrite the data and state of the chip. The LDI signal resets the serial interface.

The $\overline{\text{OPENLED}}$ bits, NOL[48:1], are well suited for use with the LT3595A, and indicate an open circuit has been detected on at least one of the 48 LED strings driven by the three LT3595A's. The part monitors the three LT3595A wired-OR $\overline{\text{OPENLED}}$ pins that detect open LED strings for each LT3595A. When one of the LT3595A's detects an open LED string, it will pull $\overline{\text{OPENLED}}$ low during the PWM high time for that LED string. The state of $\overline{\text{OPENLED}}$ is captured by the LT8500 on the rising edge of the first SCKI of a new frame (after LDI). Since SCKI and PWMCK are asynchronous, the detection of an open LED string by this method is a probability function dependent on the frame rate and PWM duty cycle. If a new frame begins when the PWM pin associated with an open LED string is high, the $\overline{\text{OPENLED}}$ pin will be driven low and captured in the status register, but if a new frame begins when the associated PWM pin is low, the $\overline{\text{OPENLED}}$ pin will be pulled high and the status register will capture a default high. When a low $\overline{\text{OPENLED}}$ pin is captured, signaling an open, each of the 48 $\overline{\text{OPENLED}}$ (NOL[48:1]) status flags will be cleared. Upon detecting this condition in the status frame, or as a polling strategy, the host may request an LED self test (CMD = 0x5X), where the LT8500 will test each channel to determine which, if any, is open. The test drives each PWM pin high, one at a time, in order, for 64 PWMCK cycles each, and captures the corresponding value on the $\overline{\text{OPENLED}}$ pin for the associated PWM channel. These results will overwrite the NOL flags in the status frame and the open LED test bit (OLT) will be set in the status frame to indicate that the NOL data in this status frame is given by channel. In the next frame, the OLT bit will be cleared and all 48 NOL bits will again reflect the state of the $\overline{\text{OPENLED}}$ pin.

### PCB Layout Guidelines

The following guidelines should be considered when designing printed circuit boards (PCBs) using the LT8500. These guidelines are more important as clock speeds and daisy chain sizes increase.

1. Match the line lengths and delays between SDI and SCKI to each LT8500.
2. Ensure the timing of LDI to each chip meets SCKI to LDI setup and hold requirements. In a 5-pin topology, SCKI is delayed by each chip in the daisy chain, so LDI may need extra delay to match the delayed SCKI down the chain. See the discussion on topology in the Operation section.
3. Avoid cross talk between the communication signals (SDI, SCKI, LDI, SDO, SCKO) and the PWMs. Even though the PWM's signals toggle at a slow rate, all of their rising edges can occur within a few nanoseconds of each other.
4. Buffer the signals returning to the host if their paths are long.
5. High speed techniques: standard high speed PCB design techniques should be used on high frequency clock and data lines. These include short path lengths, shielding of high speed data cables and traces, minimized parasitic capacitance, and reducing antennas and reflections.
6. A ceramic bypass capacitor should be placed close to the $V_{CC}$ pin.

## TYPICAL APPLICATIONS

Four typical applications are shown in Figures 7 to 10. Figures 7 and 8 illustrate the 5-pin and 4-pin topologies for daisy chains as discussed earlier in this data sheet. Figure 9 illustrates a single LT8500 controlling 48 resistor ballasted LED strings. Figure 10 illustrates a novel use of the LT8500 as a 48-channel digital-to-analog converter (DAC). Using a simple RC filter on each PWM output, the resulting converter has very good error characteristics as shown in the accompanying differential linearity error (DLE) and integrated linearity error (ILE) charts (Figures 11 and 12). The DLE measurements were taken from an all codes test, and were compensated for power supply variation on $V_{CC}$ of less than ±0.01% over the course of the test. The ILE is simply the sum of all previous compensated DLE measurements. The units of the DLE and ILE measurements are in PWM LSB's.

## TYPICAL APPLICATIONS

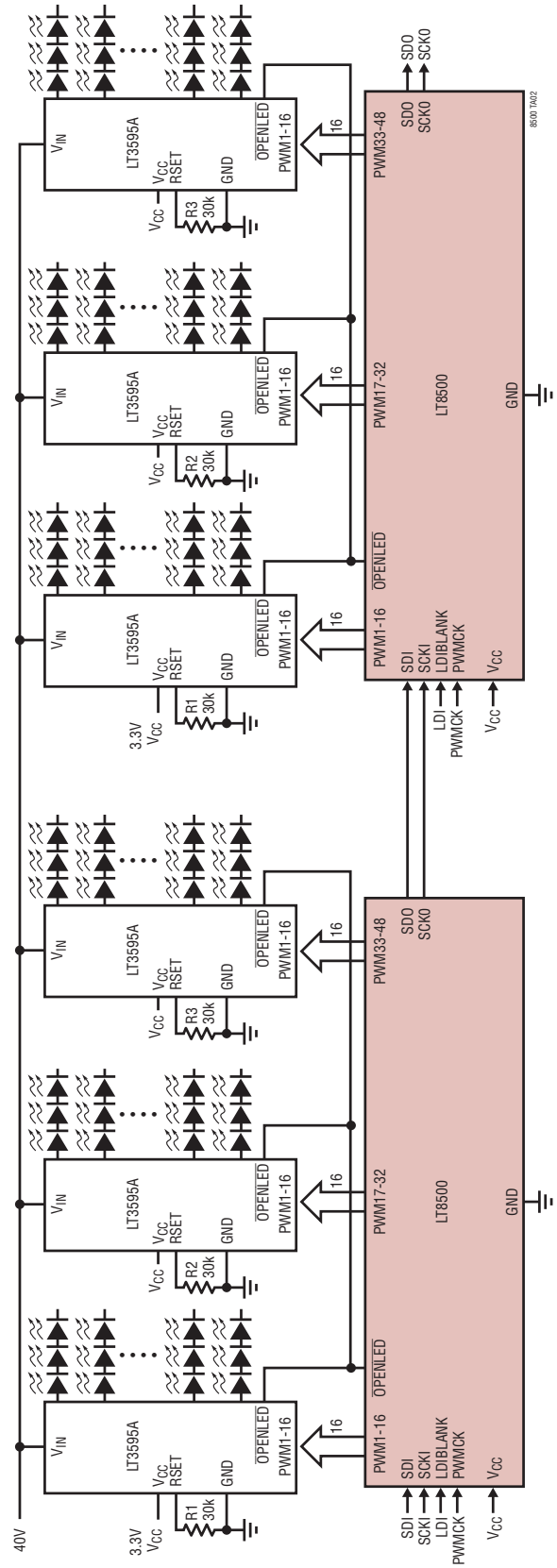



Figure 7. LT8500 Daisy Chain Driving LT3595As Using 5-Pin Topology

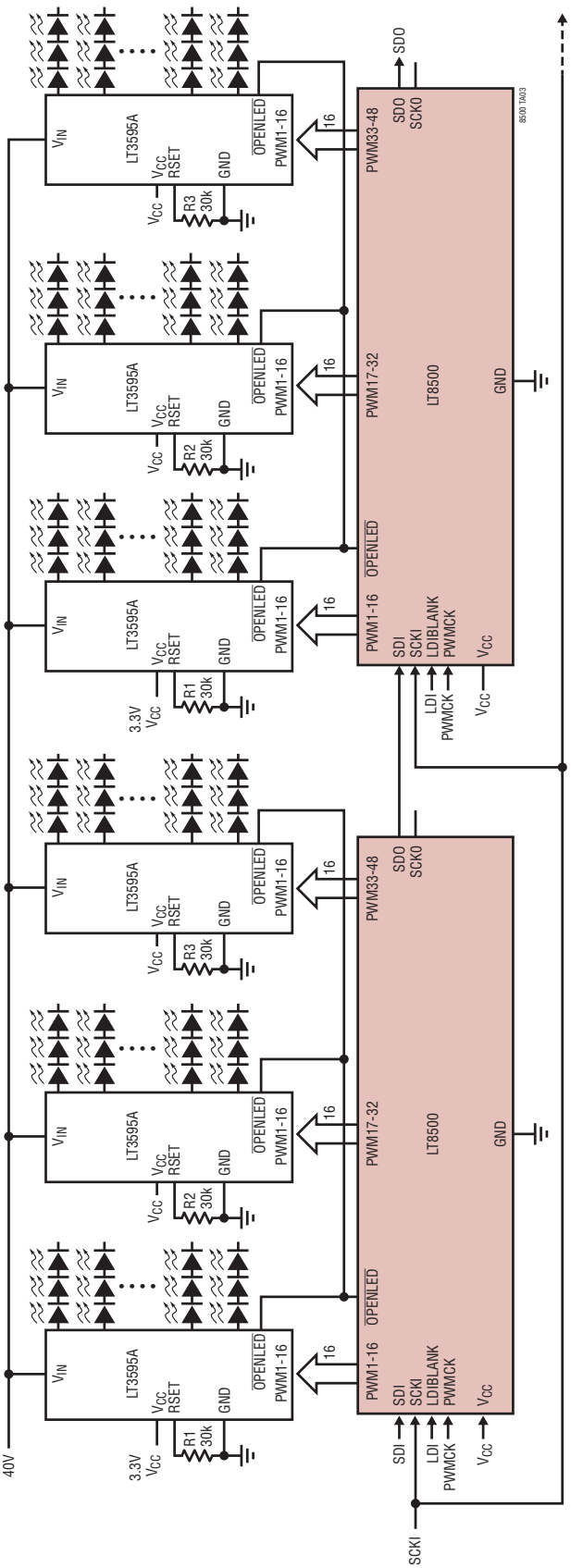



Figure 8. LT8500 Daisy Chain Driving LT3595As Using 4-Pin Topology

## TYPICAL APPLICATIONS

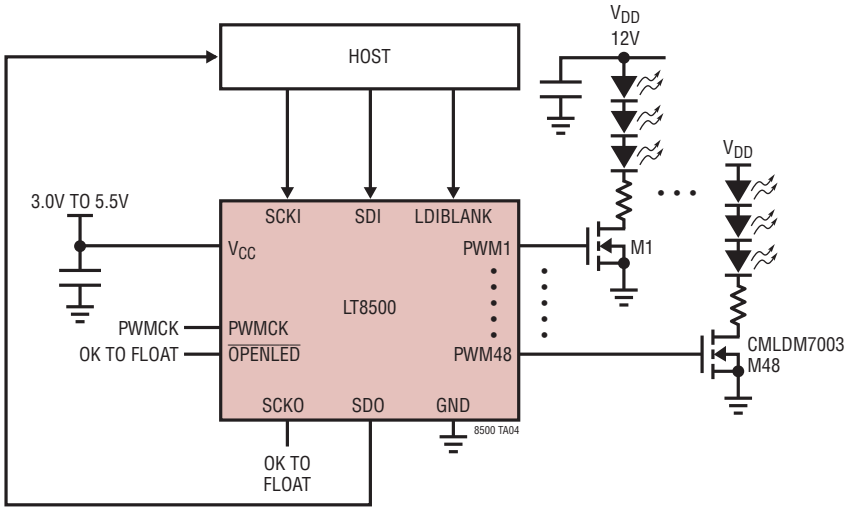



**Figure 9. Single LT8500 Driving 48 Resistor Ballasted LED Strings From a $V_{DD}$ Rail**

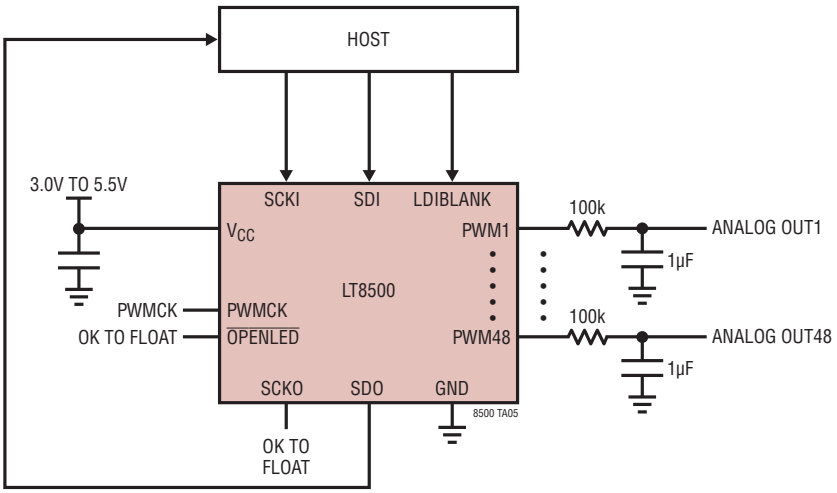



**Figure 10. Single LT8500 Implementing 48 Digital-to-Analog Converter (DAC) Channels**

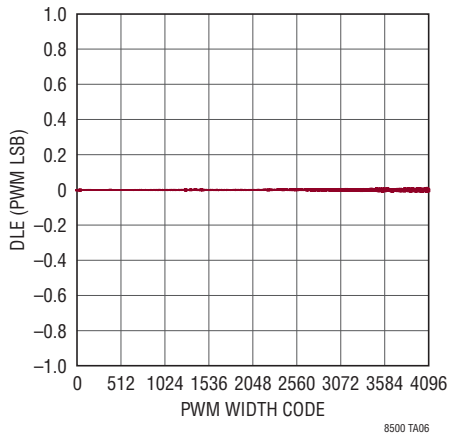



**Figure 11. DAC Differential Linearity Error (DLE)**

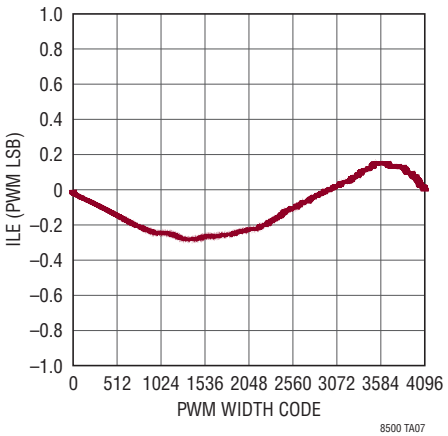
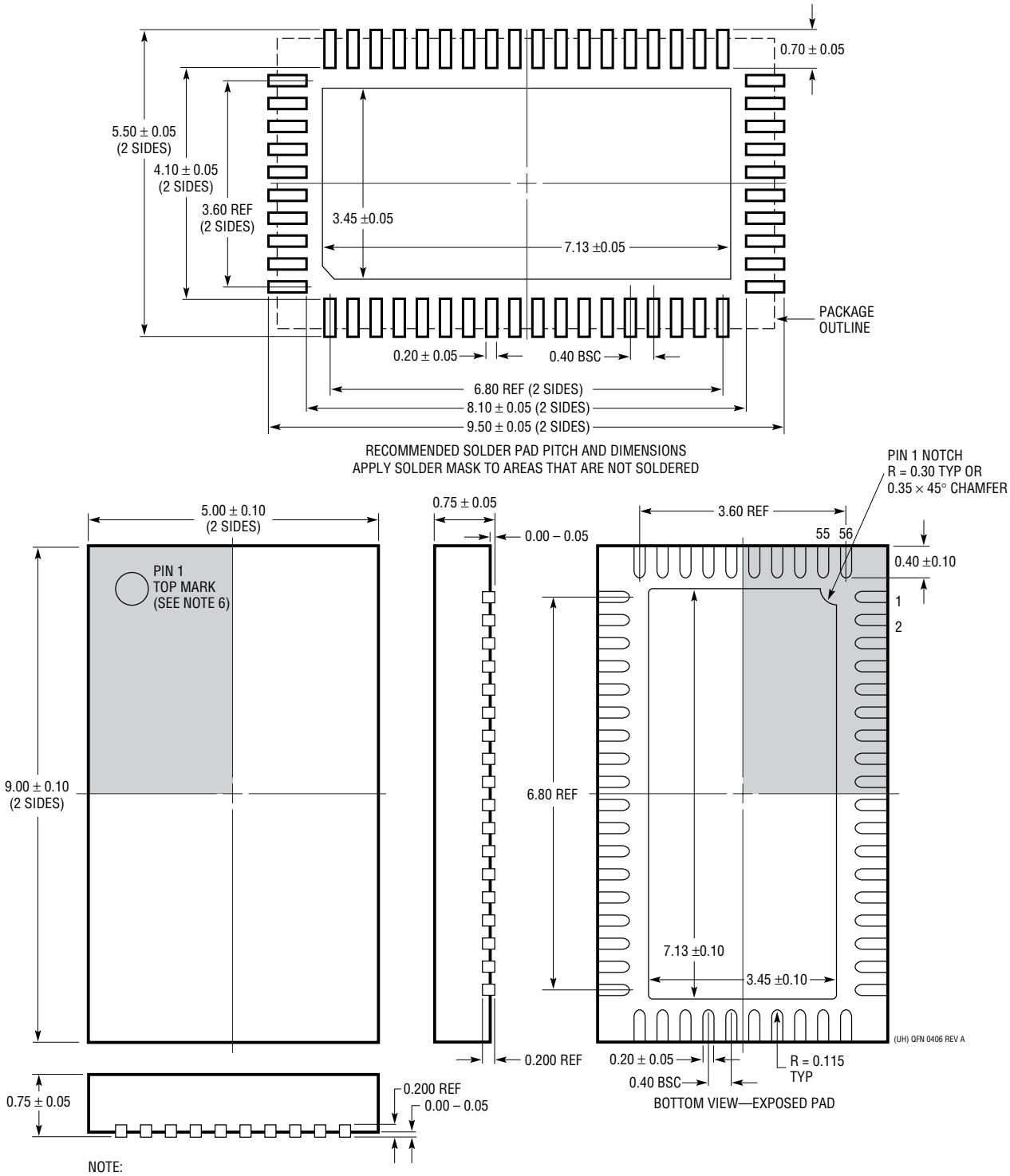



**Figure 12. DAC Integrated Linearity Error (ILE)**

# PACKAGE DESCRIPTION

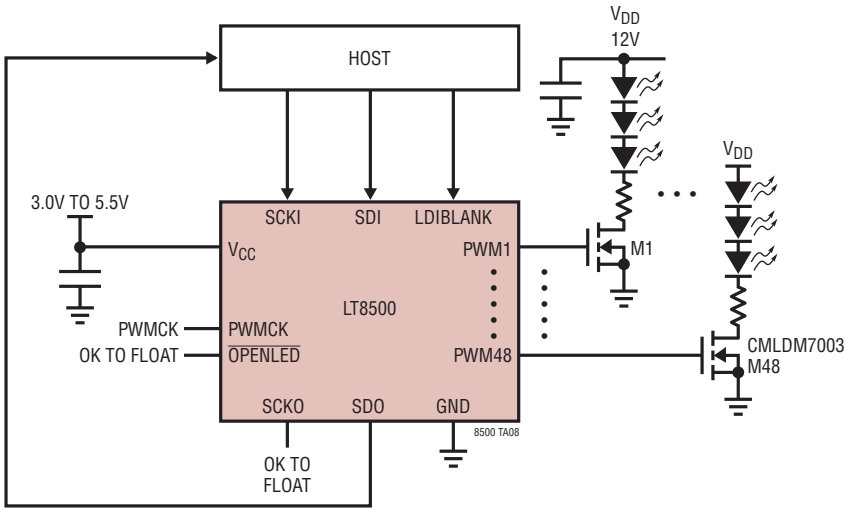
**UHH Package**
**56-Lead (5mm × 9mm) Plastic QFN**
(Reference LTC DWG # 05-08-1727 Rev A)




NOTE:
1. DRAWING IS NOT A JEDEC PACKAGE OUTLINE
2. DRAWING NOT TO SCALE
3. ALL DIMENSIONS ARE IN MILLIMETERS
4. DIMENSIONS OF EXPOSED PAD ON BOTTOM OF PACKAGE DO NOT INCLUDE MOLD FLASH. MOLD FLASH, IF PRESENT, SHALL NOT EXCEED 0.20mm ON ANY SIDE
5. EXPOSED PAD SHALL BE SOLDER PLATED
6. SHADED AREA IS ONLY A REFERENCE FOR PIN 1 LOCATION ON THE TOP AND BOTTOM OF PACKAGE

## TYPICAL APPLICATION

**Single LT8500 Driving 48 Resistor Ballasted LED Strings From a $V_{DD}$ Rail**




## RELATED PARTS

| PART NUMBER | DESCRIPTION | COMMENTS |
|---|---|---|
| LT3746 | 55V, 1MHz 32-Channel Full Featured 30mA Step-Down LED Driver | $V_{IN(MIN)}$ = 6V, $V_{IN(MAX)}$ = 55V, $V_{OUT(MAX)}$ = 13V, Dimming = 5,000:1 True Color PWM, $I_{SD}$ < 1µA, Package 5mm × 9mm QFN-56 |
| LT3595/ LT3595A | 45V, 2.5MHz 16-Channel, 50mA Full Featured Boost LED Driver | $V_{IN(MIN)}$ = 4.5V, $V_{IN(MAX)}$ = 45V, $V_{OUT(MAX)}$ = 45V, Dimming = 5,000:1 True Color PWM, $I_{SD}$ < 1µA, Package 5mm × 9mm QFN-56 |
| LT3754 | 60V, 1MHz Boost 16-Channel, 50mA LED Driver with True Color 3,000:1 PWM Dimming and 2% Current Matching | $V_{IN(MIN)}$ = 4.5V, $V_{IN(MAX)}$ = 40V, $V_{OUT(MAX)}$ = 60V, Dimming = 3,000:1 True Color PWM, $I_{SD}$ < 1µA, Package 5mm × 5mm QFN-32 |
| LT3598 | 44V, 1.5A, 2.5MHz Boost 6-Channel, 30mA LED Driver | $V_{IN(MIN)}$ = 3V, $V_{IN(MAX)}$ = 30V(40$V_{MAX}$), $V_{OUT(MAX)}$ = 44V, Dimming = 1,000:1 True Color PWM, $I_{SD}$ < 1µA, Package 4mm × 4mm QFN-24 |
| LT3599 | 44V, 2A, 2.5MHz Boost 4-Channel, 120mA LED Driver | $V_{IN(MIN)}$ = 3V, $V_{IN(MAX)}$ = 30V(40$V_{MAX}$), $V_{OUT(MAX)}$ = 44V, Dimming = 1,000:1 True Color PWM, $I_{SD}$ < 1µA, Package 4mm × 4mm QFN-24 |


Linear Technology Corporation
1630 McCarthy Blvd., Milpitas, CA 95035-7417
(408) 432-1900 • FAX: (408) 434-0507 • www.linear.com
LT 0611 • PRINTED IN USA