Power System Management Addressing

Michael Jones

INTRODUCTION

The foundation of all PMBus applications, including LTC® Power System Management (PSM), is the ability for the PMBus master (system host) to communicate with all PMBus slaves (PSM controllers, PSM managers, PSM µModules®, and PMBus monolithic devices) on the bus. Every slave on the bus must have a unique address that does not conflict with other devices.

The bus master must also be able to communicate with PSM slaves in a few less than obvious situations:

- Address discovery
- Global actions
- Multiphase rails
- Invalid NVM
- Bus MUXes

Device addressing is achieved with a combination of base registers plus external address select (ASEL) pins, as well as special global, rail, ARA, and other special addresses.

This Application Note will present the fundamental design principles underlying the LTC PSM family, details on product family differences, as well practical examples and advice. Special cases, such as invalid NVM, will also be discussed.

The benefit to you is a design that works on day one, and works even when things go wrong. For example, if you are writing to the NVM with LTpowerPlay™, and power is lost during the NVM write, your design will be recoverable. Furthermore, it will be recoverable in the field should you choose to implement "In Flight Update" found in LTC's Linduino reference code[1]. Finally, you will be able to recognize symptoms of degenerate systems and fix them.

Once you have an understanding of how to implement LTC PSM addressing, you will be able to design reliable systems quickly.

BASIC PMBus OPERATION

PMBus is a serial communication standard that is an extension of SMBus, which is similar to I^2C. Two open-drain wires, SCL and SDA, support a bidirectional communication bus with masters and slaves. Masters are devices that control communication and are typically a microcontroller or FPGA. Slaves are devices that the master controls, and they are typically a small integrated circuit, in our case a power supply manager like LTC2977 or a power supply controller like LTC3880.

A system can have more than one master, but it is rare in practice. Usually there are multiple slaves. A master directs communication to a single slave at a time by using an address, even in a system with one slave. This means every slave must have a unique address for proper system function.

LTC Power System Management devices use an EEPROM along with resistors on pins to set the unique address of each device (slave). Therefore, part of addressing is ensuring that if any EEPROMs do not have valid data, a master can repair the system to the state where each device (slave) has its unique address.

The following sections explain these address configuration mechanisms in detail, including how to select, design, and repair addresses.

Note 1. In Flight Update is a programming engine (C Code) that applies ISP/HEX files that are exported by LTpowerPlay. Example code for In Flight Update is found in the Linduino release is ported to firmware.

BASIC PMBus ADDRESSING

PMBus addressing is defined by the SMBus standard that the PMBus standard refers to. SMBus addressing is the same as the I^2C standard. For clarity, this discussion is limited to the SMBus standard.

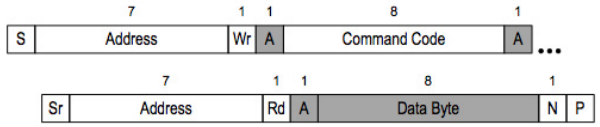

Figure 1. Read Byte Protocol (SMBus 3.0 Standard Figure 29)

Consider Read Byte protocol, defined in Figure 29 of the SMBus 3.0 standard. The address of any transaction proceeds after the start bit (S), and concludes before the ACK bit (A). In between the (S) and (A) are 8 bits, and the first 7 bits are the address, and one bit is used to indicate write (Wr) or read (Rd).

Seven bits means there are 128 potential addresses. In this Application Note, addresses are written without the (Wr/Rd) bit as follows:

0x00 to 0x7F (7-bit addressing)[2]

Sometimes programmers like to write the addresses with the extra (Wr/Rd) bit holding a zero, for instance:

0x00 to 0xFE (8-bit addressing)

These are all the even numbers. Oscilloscopes and spy tools, like the Total Phase Beagle, use the same form as this application note, and it is natural to use 0x00 to 0x7F, so we shall do the same. However, be careful when engineers give you an address and don't indicate the format. The given address maybe left-shifted by one, thus appearing twice the size it really is.

Note that Read Byte protocol uses the Address twice, but with a repeated start (Sr) preceding the second address. A repeated start is part of all read transactions. In this case, the same address must be used for (S) and (Sr).

Address Map

Not all addresses are available for PSM slaves, because the SMBus standard reserves some of them. Appendix C of the SMBus 3.0 Specification has a table of pre-assigned addresses. It is not necessary to know the intended use of them, so a simpler table will do.

Table 1. SMBus Address Map

Address	Description
0x00	General Call or START
0x01 to 0x0B	Battery etc
0x1C	Alert Response Address (ARA)
0x1D to 0x27	Open for use
0x28	Zone Read
0x29 to 0x2B	Open for use
0x2C to 0x2D	Reserved
0x2E to 0x36	Open for use
0x37	Zone Write
0x37 to 0x3F	Open for use
0x40 to 0x43	Reserved
0x44	Reserved
0x45 to 0x47	Open for use
0x48 to 0x4B	Prototyping
0x61	Default SMBus address
0x62 to 0x77	Open for use
0x78 to 0x7B	10 bit address
0x7C to 0x7F	Reserved

The SMBus Address Map shown in Table 1, uses a simple coding scheme. Conservative designs only use addresses that have a white background in the Description column. This avoids all the reserved and special addresses. PSM designs can use all the addresses with a white background in the Address column, with the possible exception of addresses 0x28 and 0x37 that were added for PMBus 3.1 zone operations.

Note 2. Some addresses are reserved.

PSM Global Addresses

There are additional addresses used by PSM that cannot be assigned to any devices, even devices that are not PSM, without causing system level problems.

The first special addresses are the global addresses, and they are 0x5A and 0x5B. These are addresses used by the bus master to talk to more than one device at a time. The address 0x5A is a global address that is not paged, for the LTC388X family. The address 0x5B is a paged global address for the LTC388X PSM family and a paged global address for the LTC297X family.

If a PSM device had its address set to either global address, and the bus master sent commands to it, all PSM devices on the bus would respond to the command. LTpowerPlay would also exhibit unexpected behavior.

The third global address is 0x7C. LTC388X PSM family devices will respond to this address if they have CRC errors in their EEPROM. Therefore do not use this address.

Other Global Addresses

Non-PSM devices may also have global addresses. These addresses must be avoided when assigning PSM addresses.

Rail Addresses

Some PSM devices have a special address called a rail address. Multiple devices or pages are simultaneously addressed with a rail address. The main function of a rail address is to allow the bus master to communicate with several phases of a rail with one command. For example, setting the VOUT voltage of a multiphase rail.

A rail address is set with a register called MFR_RAIL_ADDRESS (0xFA). The default value for this register is 0x80, which disables rail addressing. Setting the rail address to any other value enables rail addressing.

The bus master treats a rail address as a device on the bus, even though it is not a separate device. The bus master cannot tell the difference. Therefore, rail addresses are part of the overall address map of the system and must not conflict with other addresses.

Channel Addresses

There is one last special address called a channel address. It is also set with a register like a rail address, using MFR_CHANNEL_ADDRESS (0xD8). Channel addresses add an address to the bus that points to a specific page. Therefore, the PAGE register does not have to be used when using a channel address.

The bus master treats a channel address as another device on the bus and it cannot conflict with other addresses.

ADDRESS PLANNING

Address planning is simple. Create a spreadsheet with all the devices and addresses listed, using all the following address types:

- Normal addresses
- Global addresses
- Rail addresses
- Channel addresses
- Special addresses (ARA)
- Non-PSM addresses

No addresses should overlap, except global addresses, and all addresses should be unique when they have a common base address.

SETTING ADDRESSES

Although very similar for the most part, there are subtle differences in how addresses are assigned, and how devices behave, between the LTC388X DC/DC controller family and the LTC297X manager family. There are also minor differences within the families. However, the principles of operation are similar, and all devices can be considered generically before getting into specific differences.

For all PSM devices, the actual address of the PSM device on the bus is a combination of a register value, called a BASE ADDRESS, and optional modification by resistors attached to the pins of the device, called ASEL pins, for Address Select Pins.

There are several reasons for using ASEL pins instead of pre-programming devices. First, pre-programming takes time and has a cost, and it may not be desirable to pre-program the devices in general. Many devices also have configuration pins to set the output voltage and that may be all that is required. Second, if a device has a CRC NVM error, it would lose its address, and there would be no way for the bus master to uniquely communicate with each device to reprogram them.

Therefore, the solution is ASEL pins and a base address. A bus master can communicate with all devices using the global address 0x5B. If the bus master can communicate with all devices, it can set the BASE ADDRESS because all PSM devices support the MFR_I2C_BASE_ADDRESS[3] register. The bus master can then force all devices to read their address pins, and all devices will have unique addresses known by the bus master. Once the bus master can communicate with each device individually, the EEPROM can be reprogrammed.

This means the following, and is the most important thing in this Application Note to remember:

> A*ll devices on the bus must have a unique address when they all share a common BASE ADDRESS, and their ASEL pins have been read and applied.*

If this rule is followed, a system can be repaired with LTpowerPlay, Linduino reference code, or custom firmware if any NVM corruption occurs. Therefore, unsoldering Linear Technology PSM devices is never required to repair a system.

Following are the details of the product families for a full understanding of how to set up addresses, and how to work within parameters of the address map.

LTC388X PSM Family

The LTC388X family of PSM device addresses are set by a combination of the BASE ADDRESS value and the value obtained from the ASEL pin(s). Some LTC388X devices have a single ASEL pin, and some have two ASEL pins, so they are treated differently.

Single ASEL pins

Table 2. LTC3880 ASEL

R_{TOP} (kΩ)	R_{BOTTOM} (kΩ)	SLAVE ADDRESS	LSB HEX
0 or Open	Open	EEPROM	
10	23.2	xyz_1111	F
10	15.8	xyz_1110	E
16.2	20.5	xyz_1101	D
16.2	17.4	xyz_1100	C
20	17.8	xyz_1011	B
20	15	xyz_1010	A
20	12.7	xyz_1001	9
20	11	xyz_1000	8
24.9	11.3	xyz_0111	7
24.9	9.09	xyz_0110	6
24.9	7.32	xyz_0101	5
24.9	5.76	xyz_0100	4
24.9	4.32	xyz_0011	3
30.1	3.57	xyz_0010	2
30.1	1.96	xyz_0001	1
Open	0	xyz_0000	0

Table 2, shows the ASEL pins from the LTC3880 data sheet. The LTC3880 has a single ASEL pin. The three “xyz” bits in the slave address column are the BASE ADDRESS, which is stored in the MFR_I2C_BASE_ADDRESS (0xE6) register. The ASEL pin programs the 4 LSBs from the table, except in the case where there are no resistors on the ASEL pin. In that case, all 7 bits of the BASE ADDRESS are used.

Only use open on the ASEL pins when there is a single device on the bus. The ASEL pins must be used in any multi-device application, to program the address. Using the single ASEL pin to program the device limits the number of addresses on the bus to 16.

A bus MUX provides a solution for expanding the number of addresses and is discussed in the bus segmentation section of this application note.

Dual ASEL Pins

Dual ASEL Pins expand the 16 address limit to 127.

Note 3. MFR_I2C_BASE_ADDRESS is named differently in some devices, but is always command code 0xE6.

Table 3. LTC3882 ASEL

R_{TOP} (kΩ)	R_{BOT} (kΩ)	ASEL1		ASEL0	
		LTC3882 DEVICE ADDRESS BITS[6:4]		LTC3882 DEVICE ADDRESS BITS[3:0]	
		BINARY	HEX	BINARY	HEX
0 or Open	Open	from EEPROM		from EEPROM	
10	23.2			1111	F
10	15.8			1110	E
16.2	20.5			1101	D
16.2	17.4			1100	C
20	17.8			1011	B
20	15			1010	A
20	12.7			1001	9
20	11			1000	8
24.9	11.3	111	7	0111	7
24.9	9.09	110	6	0110	6
24.9	7.32	101	5	0101	5
24.9	5.76	100	4	0100	4
24.9	4.32	011	3	0011	3
30.1	3.57	010	2	0010	2
30.1	1.96	001	1	0001	1
Open	0	000	0	0000	0

Table 3, shows the ASEL pins from the LTC3882 data sheet. If ASEL1 is set to "from EEPROM," ASEL0 works just like the single ASEL behavior of the LTC3880. Resistors connected to the ASEL1 pin control the 3 MSBs of the device address. This expands the number of unique addresses to 127 values.

A system comprised of single and dual ASEL pin devices, may have up to 16 of each single ASEL pin device type, and as many dual ASEL devices as there are available addresses.

LTC297X PSM Family

The main difference of the LTC297X family is that the ASEL pins select values that are added to a BASE ADDRESS. Added means addition, not replacing bits or masking.

Table 4. LTC2975 ASEL

ADDRESS PINS		DESCRIPTION	HEX DEVICE ADDRESS		BINARY DEVICE ADDRESS							
ASEL1	ASEL0		7-Bit	8-Bit	6	5	4	3	2	1	0	R/W
X	X	Alert Response	0C	19	0	0	0	1	1	0	0	1
X	X	Global	5B	B6	1	0	1	1	0	1	1	0
L	L	N = 0	5C*	B8	1	0	1	1	1	0	0	0
L	NC	N = 1	5D	BA	1	0	1	1	1	0	1	0
L	H	N = 2	5E	BC	1	0	1	1	1	1	0	0
NC	L	N = 3	5F	BE	1	0	1	1	1	1	1	0
NC	NC	N = 4	60	C0	1	1	0	0	0	0	0	0
NC	H	N = 5	61	C2	1	1	0	0	0	0	1	0
H	L	N = 6	62	C4	1	1	0	0	0	1	0	0
H	NC	N = 7	63	C6	1	1	0	0	0	1	1	0
H	H	N = 8	64	C8	1	1	0	0	1	0	0	0

H = Tie to V_{DD33}, NC = No Connect = Open or Float, L = Tie to GND,
X = Don't Care
*MFR_I2C_BASE_ADDRESS = 7bit 0x5C (Factory Default)

Table 4, shows the LTC2975 address look-up table. These ASEL pins have three states, high, low, or unconnected. Two pins select 9 different values that are shown as "N = ". In this table the BASE ADDRESS is 0x5C, and when N = 0, the address is 0x5C.

This table also has an 8-bit column. This is the address being used in the Application Note, shifted one to the left as previously discussed, for those used to this convention.

CRC MISMATCHES

All Linear Technology PSM devices have EEPROM, which store settings that determine the output voltage, supervisor limits, and basic operation. Any errors in the memory could cause damage to the loads. Linear's PSM devices have a 10-year EEPROM retention specification. However, unanticipated system conditions during an EEPROM write, such as high temperature or supply voltage collapse may cause the write to fail, corrupting the EEPROM contents.

The purpose of CRC is to verify that the EEPROM contents are correct. A valid CRC guarantees that the device operates safely, and an invalid CRC inhibits the device operation, and notifies the host with a PMBus ALERTB.

How Does CRC Work?

When a PSM device is first powered on, it transfers the contents of the EEPROM to RAM, because RAM is the operational memory of the device. When this happens, the device calculates a CRC of the RAM contents, and then compares against the stored CRC in the EEPROM.[4] If the two CRCs match, the part operates; if they do not match, the device will report a CRC fault, and remain in the reset state.

Connecting the GPIOB/FAULTB pins of all PSM devices, and programming the devices to share faults and inhibit operation during faults will cause any CRC within the whole system to prevent power delivery of all rails in the system. This is generally the best system design unless a multi-fault analysis has proved that all combinations of faults are safe. When all rails are off, analysis is much simpler, and the system becomes much safer, because any fault results in a complete power-off of all rails.

LTC388X CRC

If an LTC388X device has a CRC mismatch, its address will become 0x7C. If other LTC388X devices have a CRC mismatch, they will also be at 0x7C at the same time. These devices will remain at 0x7C until their NVMs are rewritten with valid data, and they go through initialization, which occurs from a MFR_RESET (0xFD) command or power cycle.

LTC297X CRC

If an LTC297X device has a CRC mismatch, its value may be one of two addresses. A simple case is the default base address 0x5C, which occurs when any portion of the entire EEPROM has a CRC mismatch.

Because the LTC297X processes EEPROM in blocks, and blocks near the bottom may have valid CRC, and blocks near the top may not, it is also possible to have an address that is the stored base address.

This is different than the LTC388X, because the LTC388X does a complete CRC verification of the entire EEPROM before determining the address, while the LTC297X determines the address incrementally.

Fixing CRC Mismatches

In general, CRC mismatches are never manually fixed. LTpowerPlay can easily fix CRC mismatches with its programming tool, and In Flight Update will also fix CRC mismatches. However, it is helpful to understand how they fix CRC to better understand addressing.

The first step to fix CRC mismatches is to gain access to all devices with CRC mismatches individually. This means sending PMBus commands to restore the individual addresses.

A MFR_I2C_BASE_ADDRESS command is issued to the global address 0x5B. Then a sequence of commands is sent that force all devices to read their ASEL pins.

Note: It is possible to target one device with a CRC mismatch if you can determine its current address, such as 0x7C for LTC388X or the default base address of a LTC297X. However, it is far simpler to use a global address, as it causes no harm to other devices. Furthermore, if the bus is not well planned and a LTC297X lands on a non-PMBus device's address, targeting may have side effects.

The second step is to send data to the device. In theory, every command the device supports could be used, but it is more efficient to send bulk data via an MFR_EE_DATA command. LTpowerPlay and In Flight Update both use this command.

Step one will not work on a poorly planned address design that doesn't use the ASEL pins to program all devices to unique addresses. If setting the MFR_I2C_BASE_ADDRESS causes more than one PSM device to have the same address, there is no way to use MFR_EE_DATA to repair a single device, because any devices that share the same address will be programmed with the same data.

BUS SEGMENTATION

There are two main reasons to segment a PMBus:

1. Addressing
2. Capacitance/Speed

Bus segmenting for addressing typically only occurs in very large systems.

Note 4. The PSM device creates the CRC in its EEPROM when an operation writes data to EEPROM, either via RAM, or externally.

Capacitance problems may occur in very large systems with long bus routing when the bus master is operating at 400kHz. This can cause a timing specification violation, or interfere with performance goals.

Typically, an LTC4306 multiplexer is used. The key feature of the LTC4306 is that it can connect individual segments one at a time, or all segments at once.

When repairing CRC mismatches, it is used to connect the bus master to one segment at a time. When used by firmware, it is used to connect all segments into a single bus. When the segments connected as a single bus, all addresses must be unique.

It will help to walk though a typical example and see how to satisfy the repair process as well as act as a single bus.

Bus Segmentation Example

The example system is comprised of:

- 4 segments
- Each segment has 16 LTC3880 devices
- An LTC4306 multiplexer

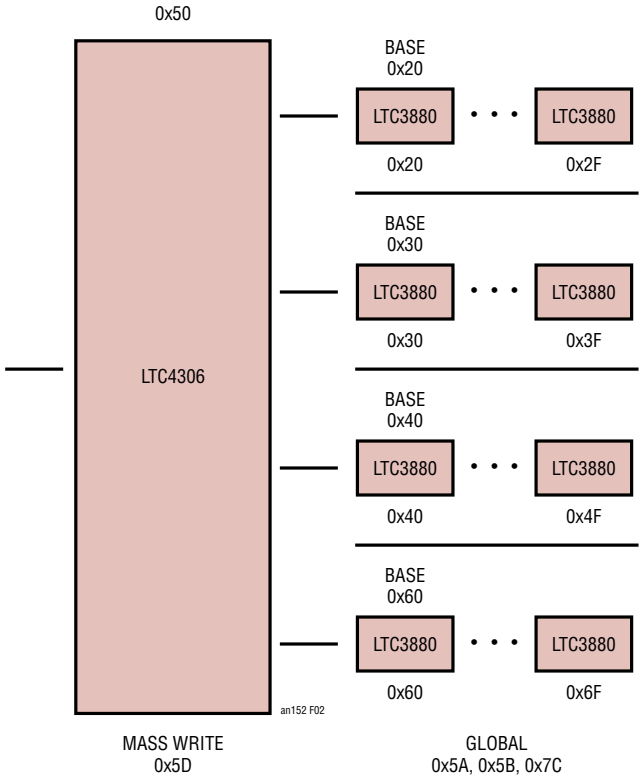

Figure 2. Bus Segmentation Example

The LTC4306 address is set to 0x50 using 3 ASEL pins. It has a global Mass Write Address, at 0x5D. The LTC3880s have global addresses of 0x5A and 0x5B. This design uses 4 base addresses for the 4 segments: 0x20, 0x30, 0x40, and 0x60. This avoids the 0x50 to 0x5F address range to which all Linear Technology PSM device addresses are programmable. In addition, most Linear Technology I^2C/SMBus devices support this address range. An LTC4316 I^2C device easily makes any other device compatible with this addressing range.

Multiphase rails may also be addressed in the 0x5X range, or addressed at available addresses below 0x10, or above 0x6F.

Firmware Operation

The Board Management Controller will connect the input PMBus to all output PMBus segments by setting register 3 of the LTC4306 to 0xF0. In this state, there are 64 devices on the bus with addresses: 0x20 to 0x2F, 0x30 to 0x3F, 0x40 to 0x4F, and 0x60 to 0x6F. The global addresses 0x5A and 0x5B are available for global operations. No devices that operate normally without CRC mismatches will respond to address 0x7C.

System Programming

System programming is the initial programming of the bus segments using LTpowerPlay or In Flight Update. The LTC4306 will connect one segment at a time by setting register 3 to values 0x80, 0x40, 0x20 and 0x10 respectively.

LTpowerPlay or In Flight Update will program the particular segment that is connected by the LTC4306. The MFR_I2C_BASE_ADDRESS for the connected segment is set using the global address 0x5B. The ASELs pin resistors ensure a unique address for each device on the connected segment.

The ASEL pin resistors for the respective devices on each segment are the same and select an address modifier of 0x00 to 0x0F. Connecting one segment at a time to set the MFR_I2C_BASE_ADDRESS for each segment individually results in a unique device address for all 64 devices.

Note: If devices with 2 ASEL pins are used, the segments may contain more than 16 devices, because 2 ASEL pins allow creation of up to 128 devices with one base address.

Fixing CRC Mismatches

This case is identical to the System Programming case. LTpowerPlay and In Flight Update treat the programming process the same. In both cases MFR_I2C_BASE_ADDRESS is used during programming.

Each segment is connected one by one and each segment is programmed.

Interaction of Firmware Operation and System Programming

The LTC4306 defaults to all segments disconnected. Therefore, if the Board Management Controller (BMC) is held in reset for system debugging, a DC1613 PMBus-to-USB controller is connected, and LTpowerPlay is running, the DC1613 will not be connected to any PSM devices on the bus.

There are two ways to make sure the Firmware does not interfere with LTpowerPlay. The first way is to have a debug mode for the BMC, such that the BMC comes out of reset and connects all the segments, and then pauses. This enables LTpowerPlay to interact with the full bus for system bring up and debug.

When the BMC pauses, it will prevent interactions with LTpowerPlay. Even though LTpowerPlay can multi-master, commands that rely on the PAGE command will interact. Therefore, pausing prevents both the firmware and LTpowerPlay from getting erroneous telemetry.

The second way is to hold the BMC in reset during debug, and allow LTpowerPlay to control the MUX. This is typically the safest case, because when the BMC is released from reset, for example after turning LTpowerPlay off, it will set up the multiplexer to the proper state regardless of the current state.

When the BMC and LTpowerPlay alternate between controlling the bus, neither master should make assumptions about the multiplexer state. Any time a major action is taken, such as when either master continues from a pause state, it should first set the multiplexer to the proper state.

SUMMARY

Power system management address planning is not complicated, but attention must be paid to special cases of programming and system recovery from CRC mismatches. All Linear Technology PSM devices are compatible.

All Linear Technology PSM devices combine a BASE ADDRESS and modification with ASEL pins, to create a unique address. All PSM devices are subject to a CRC mismatch if power is lost during programming. Setting the BASE ADDRESS, reprogramming the EEPROM, and resetting the device will fully recover a system. LTpowerPlay and In Flight Update automatically handle recovery of a proper address plan.

Address planning is assigning the BASE ADDRESSes and ASEL pins such that there is never a case where two devices are permanently at the same address. Using devices with 2 ASEL pins or segmenting the bus with an I^2C multiplexer such as the LTC4306 allows design of larger systems.

Address planning must occur before the PCB design because the ASEL pins and bus segmentation relies on resistors and pins.

If you have questions regarding your particular design, please consult your local Linear Technology Field Application Engineer. Your local FAE can review your address plan before you begin PCB design. You should also download and use the full design checklist[6] to ensure that all best practices are followed.

Note 5. A DC1613 is used with LTpowerPlay to configure and debug PSM designs.

Note 6. http://cds.linear.com/docs/en/product-info/LTC297XDesignChecklist.pdf
http://cds.linear.com/docs/en/product-info/LTC388XDesignChecklist.pdf

Linear Technology Corporation
1630 McCarthy Blvd., Milpitas, CA 95035-7417
(408) 432-1900 • FAX: (408) 434-0507 • www.linear.com
LT 0716 • PRINTED IN USA