# MAX17201/MAX17205/ MAX17211/MAX17215

# Stand-Alone ModelGauge m5 Fuel Gauge with SHA-256 Authentication

## General Description

The MAX1720x/MAX1721x are ultra-low power stand-alone fuel gauge ICs that implement the Maxim ModelGauge™ m5 algorithm without requiring host interaction for configuration. This feature makes the MAX1720x/MAX1721x excellent pack-side fuel gauges. The MAX17201/MAX17211 monitor a single cell pack. The MAX17205/MAX17215 monitor and balance a 2S or 3S pack or monitor a multiple-series cell pack.

To prevent battery pack cloning, the ICs integrate SHA-256 authentication with a 160-bit secret key. Each IC incorporates a unique 64-bit ID.

The ModelGauge™ m5 algorithm combines the short-term accuracy and linearity of a coulomb counter with the long-term stability of a voltage-based fuel gauge, along with temperature compensation to provide industry-leading fuel gauge accuracy. The IC automatically compensates for cell aging, temperature, and discharge rate, and provides accurate state of charge (SOC) in milliampere-hours (mAh) or percentage (%) over a wide range of operating conditions. As the battery approaches the critical region near empty, the ModelGauge m5 algorithm invokes a special error correction mechanism that eliminates any error. The ICs provide accurate estimation of time-to-empty and time-to-full, Cycle+™ age forecast, and three methods for reporting the age of the battery: reduction in capacity, increase in battery resistance, and cycle odometer.

The ICs provide precision measurements of current, voltage, and temperature. Temperature of the battery pack is measured using an internal temperature measurement and up to two external thermistors supported by ratiometric measurements on auxiliary inputs. A Maxim 1-Wire® (MAX17211/MAX17215) or 2-wire I²C (MAX17201/MAX17205) interface provides access to data and control registers. The ICs are available in lead-free, 3mm x 3mm, 14-pin TDFN packages.

## Applications

- Smartphones and Tablets
- Portable Game Players
- e-Readers
- Digital Still and Video Cameras
- Handheld Computers and Terminals
- Portable Medical Equipment
- Handheld Radios

## Benefits and Features

- ModelGauge m5 Algorithm
  - Eliminates Error when Approaching Empty Voltage
  - Eliminates Coulomb-Counter Drift
  - Current, Temperature, and Age Compensated
  - Does Not Require Empty, Full, or Idle States
  - No Characterization Required for EZ Performance (See the *ModelGauge m5 EZ Performance* Section)
  - Cycle+ Age Forecasting Observes Lifespan
- Nonvolatile Memory for Stand-Alone Operation
  - Learned Parameters and History Logging
  - Up to 75 Words Available for User Data
- Precision Measurement System
  - No Calibration Required
- Time-to-Empty and Time-to-Full Estimation
- Temperature Measurement
  - Die Temperature
  - Up to Two External Thermistors
- Multiple Series Cell Pack Operation
- Low Quiescent Current
  - MAX172x1: 18µA Active, 9µA Hibernate
  - MAX172x5: 25µA Active, 12µA Hibernate
- Alert Indicator for Voltage, SOC, Temperature, Current, and 1% SOC Change
- High-Speed Overcurrent Comparators
- Predicts Remaining Capacity Under Theoretical Load
- SHA-256 Authentication
- Maxim 1-Wire or 2-Wire (I²C) Interface
- SBS 1.1 Compatible Register Set

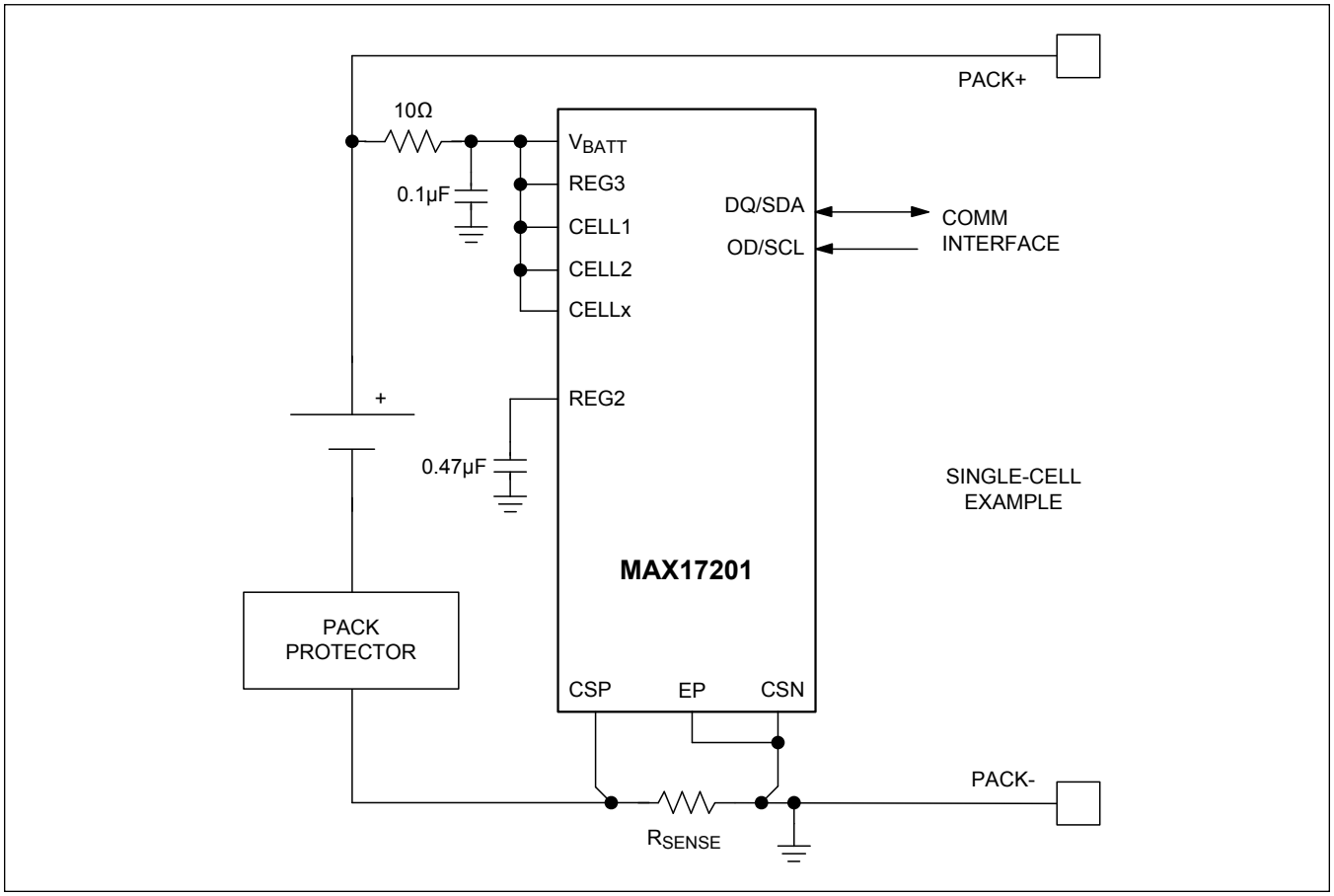
***Ordering Information* appears at end of data sheet.**

*ModelGauge and Cycle+ are trademarks and 1-Wire is a registered trademark of Maxim Integrated Products, Inc.*

*19-8424; Rev 0; 2/16*

maxim integrated™

## Simplified Block Diagram



## TABLE OF CONTENTS

General Description . . . 1
Simplified Block Diagram . . . 2
Absolute Maximum Ratings . . . 13
Package Information . . . 13
Electrical Characteristics . . . 13
Typical Operating Characteristics . . . 19
Pin Configuration . . . 23
Pin Description . . . 23
Functional Diagram . . . 24
Detailed Description . . . 24
- Register Description Conventions . . . 25
- Standard Register Formats . . . 25
  - Device Reset . . . 25
  - Nonvolatile Backup and Initial Value . . . 25
  - Register Naming Conventions . . . 25
- Alternate Initial and Factory Default Values . . . 26
- Typical Operating Circuits . . . 26
  - Single-Cell Typical Operating Circuit (MAX17201/MAX17211 Only) . . . 26
  - Multicell Typical Operating Circuits (MAX17205/MAX17215 Only) . . . 27
  - Cell Balancing Typical Operating Circuits (MAX17205/MAX17215 Only) . . . 28
- ModelGauge m5 Algorithm . . . 28
  - ModelGauge m5 Registers . . . 29
  - ModelGauge m5 Algorithm Output Registers . . . 29
  - RepCap Register (005h) . . . 29
  - RepSOC Register (006h) . . . 32
  - FullCapRep Register (035h) . . . 32
  - TTE Register (011h) . . . 32
  - TTF Register (020h) . . . 32
  - Age Register (007h) . . . 32
  - Cycles Register (017h) . . . 32
  - TimerH Register (0BEh) . . . 32
  - FullCap Register (010h) . . . 32
  - nFullCapNom Register (1A5h) . . . 32
  - RCell Register (014h) . . . 32
  - VRipple Register (0BCh) . . . 33
  - nSOC Register (1AEh) . . . 33
  - nVoltTemp Register (1AAh) . . . 33
  - ModelGauge m5 EZ Performance . . . 33

## TABLE OF CONTENTS (CONTINUED)

OCV Estimation and Coulomb Count Mixing . . . 35
Empty Compensation. . . . 37
End-of-Charge Detection . . . 38
Fuel Gauge Learning . . . 39
Converge-to-Empty . . . 41
Determining Fuel Gauge Accuracy . . . 42
Initial Accuracy. . . . 42
Cycle+ Age Forecasting. . . . 42
nAgeFcCfg Register (1D2h) . . . 42
AgeForecast Register (0B9h) . . . 42
Age Forecasting Requirements . . . 43
Enabling Age Forecasting . . . 44
Battery Life Logging. . . . 44
Life-Logging Data Example . . . 44
Determining Number of Valid Logging Entries . . . 46
Reading History Data. . . . 47
History Data Reading Example . . . 47
ModelGauge m5 Algorithm Model Registers. . . . 49
nXTable0 (180h) to nXTable11 (18Bh) Registers . . . 49
nOCVTable0 (190h) to nOCVTable11 (19Bh) Registers . . . 49
nQRTable00 (1A0h) to nQRTable30 (1A3h) Registers . . . 49
nFullSOCThr Register (1C6h) . . . 49
nVEmpty Register (19Eh). . . . 49
nDesignCap Register(1B3h) . . . 49
nRFastVShdn Register (1D5h). . . . 50
nIChgTerm Register (19Ch) . . . 50
nRComp0 Register (1A6h). . . . 50
nTempCo Register (1A7h) . . . 50
nIAvgEmpty Register (1A8h) . . . 50
ModelGauge m5 Algorithm Configuration Registers. . . . 51
nFilterCfg Register (19Dh) . . . 51
nRelaxCfg Register (1B6h) . . . 51
nLearnCfg Register (19Fh). . . . 52
nMiscCfg Register (1B2h) . . . 52
nTTFCfg Register (1C7h) . . . 53
nConvgCfg Register (1B7h) . . . 53
nRippleCfg Register (1B1h) . . . 53
ModelGauge m5 Algorithm Additional Registers. . . . 54

## TABLE OF CONTENTS (CONTINUED)

Timer Register (03Eh) . . . 54
dQAcc Register (045h) . . . 54
dPAcc Register (046h) . . . 54
QResidual Register (00Ch) . . . 54
VFSOC Register (0FFh) . . . 54
VFOCV Register (0FBh) . . . 54
QH Register (4Dh) . . . 54
AvCap Register (01Fh) . . . 54
AvSOC Register (00Eh) . . . 54
MixSOC Register (00Dh) . . . 54
MixCap Register (00Fh) . . . 55
VFRemCap Register (04Ah) . . . 55
FStat Register (03Dh) . . . 55
Modes of Operation . . . 56
Status and Configuration Registers . . . 57
DevName Register (021h) . . . 57
nROMID0 (1BCh)/nROMID1 (1BDh)/nROMID2 (1BEh)/nROMID3 (1BFh) Registers . . . 57
nRSense Register (1CFh) . . . 58
nPackCfg Register (1B5h) . . . 58
nNVCfg0 Register (1B8h) . . . 59
nNVCfg1 Register (1B9h) . . . 60
nNVCfg2 Register (1BAh) . . . 61
ShdnTimer Register (03Fh) . . . 62
nConfig Register (1B0h) . . . 62
Status Register (000h) . . . 64
Status2 Register (0B0h) . . . 65
nHibCfg Register (1B4h) . . . 65
CommStat Register (061h) . . . 66
Cell Balancing (MAX17205/MAX17215 Only) . . . 66
Cell Balancing Window of Operation . . . 66
Cell Balancing Order and Thresholds . . . 67
Cell Balancing Current . . . 68
Cell Balancing Duty Cycle . . . 68
Analog Measurements . . . 68
Voltage Measurement . . . 68
Voltage Measurement Timing . . . 69
VCell Register (009h) . . . 69
AvgVCell Register (019h) . . . 69

## TABLE OF CONTENTS (CONTINUED)

MaxMinVolt Register (01Bh) . . . 70
Cell1 (0D8h)/Cell2 (0D7h)/Cell3 (0D6)/Cell4 (0D5h) Registers . . . 70
AvgCell1 (0D4h)/AvgCell2 (0D3h)/AvgCell3 (0D2h)/AvgCell4 (0D1h) Registers . . . 70
CellX Register (0D9h) . . . 70
Batt Register (0DAh) . . . 70
Current Measurement . . . 71
Current Measurement Timing . . . 71
MaxMinCurr Register (01Ch) . . . 72
nCGain Register (1C8h) . . . 72
CGTempCo Register (16Dh) . . . 72
Copper Trace Current Sensing . . . 73
Temperature Measurement . . . 73
Temperature Measurement Timing . . . 73
AvgTA Register (016h) . . . 74
MaxMinTemp Register (01Ah) . . . 75
nTCurve Register (1C9h) . . . 75
nTGain (1CAh) Register/nTOff (1CBh) Register . . . 75
Temp1 (134h)/Temp2 (13Bh)/IntTemp (135h) Registers . . . 76
AvgTemp1 (137h)/AvgTemp2 (139h)/AvgIntTemp (138h) Registers . . . 76
AIN0 Register (027h) . . . 76
At-Rate Functionality . . . 76
AtRate Register (004h) . . . 76
AtQResidual Register (0DCh) . . . 76
AtTTE Register (0DDh) . . . 76
AtAvSOC Register (0CEh) . . . 76
AtAvCap Register (0DFh) . . . 76
Overcurrent Comparators . . . 77
nODSCTh Register (18Eh) . . . 77
nODSCCfg Register (18Fh) . . . 78
Alert Function . . . 78
nVAlrtTh Register (1C0h) . . . 79
nTAlrtTh Register (1C1h) . . . 79
nSAlrtTh Register (1C2h) . . . 79
nIAlrtTh Register (1C3h) . . . 80
Memory . . . 80
ModelGauge m5 Memory Space . . . 81
Nonvolatile Memory . . . 82
Shadow RAM . . . 82

## TABLE OF CONTENTS (CONTINUED)

Nonvolatile Memory Commands . . . 83
COPY NV BLOCK [E904h] . . . 83
NV RECALL [E001h] . . . 83
HISTORY RECALL [E2XXh] . . . 83
Nonvolatile Block Programming . . . 84
Determining the Number of Remaining Updates . . . 84
General-Purpose Memory . . . 84
Memory Locks . . . 89
NV LOCK [6AXXh] . . . 89
Locking Memory Blocks . . . 89
Reading Lock State . . . 89
Smart Battery Compliant Operation . . . 89
SBS Compliant Memory Space (MAX1720x Only) . . . 90
sFirstUsed Register (136h) . . . 90
sCell1 (13Fh)/sCell2 (13Eh)/sCell3 (13Dh)/sCell4 (13Ch) Registers . . . 90
sAvgCell1 (14Fh)/sAvgCell2 (14Eh)/sAvgCell3 (14Dh)/sAvgCell4 (14Ch) Registers . . . 90
sAvCap Register (167h) . . . 91
sMixCap Register (168h) . . . 91
sManfctInfo Register (170h) . . . 91
Nonvolatile SBS Register Back-Up . . . 91
nSBSCfg Register (1BBh) . . . 91
nDesignVoltage Register (1D3h) . . . 91
nCGain and Sense Resistor Relationship . . . 92
SHA-256 Authentication . . . 92
Authentication Procedure . . . 92
Alternate Authentication Procedure . . . 93
Secret Management . . . 94
Single Step Secret Generation . . . 94
Multistep Secret Generation Procedure . . . 95
All OEMs . . . 95
Determining the Number of Remaining Updates . . . 96
Authentication Commands . . . 96
Compute MAC Without ROM ID [3600h] . . . 96
Compute MAC with ROM ID [3500h] . . . 96
Compute Next Secret Without ROM ID [3000h] . . . 96
Compute Next SECRET WITH ROM ID [3300h] . . . 97
CLEAR SECRET [5A00h] . . . 97

## TABLE OF CONTENTS (CONTINUED)

LOCK SECRET [6000h] . . . 97
Device Reset . . . 97
Reset Commands . . . 97
Hardware Reset [000Fh to Address 060h] . . . 97
Fuel Gauge Reset [0001h to Address 0BBh] . . . 97
Communication . . . 97
2-Wire Bus System (MAX17201/MAX17205 Only) . . . 97
Hardware Configuration . . . 98
I/O Signaling . . . 98
Bit Transfer . . . 98
Bus Idle . . . 98
START and STOP Conditions . . . 98
Acknowledge Bits . . . 98
Data Order . . . 99
Slave Address . . . 99
Read/Write Bit . . . 99
Bus Timing . . . 99
I2C Protocols . . . 100
I2C Write Data Protocol . . . 100
I2C Read Data Protocol . . . 100
SBS Protocols . . . 101
SBS Write Word Protocol . . . 101
SBS Read Word Protocol . . . 102
SBS Write Block Protocol . . . 102
SBS Read Block Protocol . . . 102
Packet Error Checking . . . 103
1-Wire Bus System (MAX17211/MAX17215 Only) . . . 103
Hardware Configuration . . . 104
64-Bit Net Address (ROM ID) . . . 105
I/O Signaling . . . 105
Reset Time Slot . . . 105
Write Time Slots . . . 105
Read Time Slots . . . 107
Transaction Sequence . . . 107
Net Address Commands . . . 107
Read Net Address [33h] . . . 107
Match Net Address [55h] . . . 107
Skip Net Address [CCh] . . . 107

## TABLE OF CONTENTS (CONTINUED)

Search Net Address [F0h] . . . 107
1-Wire Functions . . . 107
Read Data [69h, LL, HH] . . . 108
Write Data [6Ch, LL, HH] . . . 108
Summary of Commands . . . 109
Appendix A: Reading History Data Psuedo-Code Example . . . 110
Appendix B: Layout Guidelines . . . 113
Ordering Information . . . 114
Revision History . . . 115

## LIST OF FIGURES

Figure 1. Single-Cell Schematic . . . 26
Figure 2. Multicell Schematics . . . 27
Figure 3. Cell-Balancing Circuit Schematics . . . 28
Figure 4. Merger of Coulomb Counter and Voltage-Based Fuel Gauge . . . 29
Figure 5. ModelGauge m5 Block Diagram . . . 30
Figure 6. ModelGauge m5 Registers . . . 31
Figure 7. nSOC (1AEh) Format When nNVCfg2.enSOC = 1 . . . 33
Figure 8. nVoltTemp (1AAh) Format When nNVCfg2.enVT = 1 . . . 33
Figure 9. EZ Configuration Performance Histogram . . . 34
Figure 10. EZ Configuration Performance vs. Test Conditions . . . 34
Figure 11. Voltage and Coulomb Count Mixing . . . 35
Figure 12. ModelGauge m5 Typical Accuracy Example . . . 36
Figure 13. Handling Changes in Empty Calculation . . . 37
Figure 14. Blocking False End of Charge Events . . . 38
Figure 15. FullCapRep Learning at End of Charge . . . 39
Figure 16. FullCapNom Learning . . . 40
Figure 17. Converge to Empty . . . 41
Figure 18. nAgeFcCfg Register (1D2h) Format . . . 42
Figure 19. Benefits of Age Forecasting . . . 43
Figure 20. Sample Life-Logging Data . . . 45
Figure 21. Write Flag Register and Valid Flag Register Formats . . . 46
Figure 22. nFullSOCThr (1C6h)/FullSOCThr (013h) Format . . . 49
Figure 23. VEmpty (03Ah)/nVEmpty (19Eh) Format . . . 49

## LIST OF FIGURES (CONTINUED)

Figure 24. nRFastVshdn (1D5h) Format . . . . . 50
Figure 25. FilterCfg (029h)/nFilterCfg (19Dh) Format . . . . . 51
Figure 26. RelaxCfg (02Ah)/nRelaxCfg (1B6h) Format . . . . . 51
Figure 27. LearnCfg (028h)/nLearnCfg (19Fh) Format. . . . . 52
Figure 28. MiscCfg (02Bh)/nMiscCfg (1B2h) Format . . . . . 52
Figure 29. nTTFCfg (1C7h) Format . . . . . 53
Figure 30. nConvgCfg (1B7h) Format . . . . . 53
Figure 31. nRippleCfg (1B1h) Format. . . . . 53
Figure 32. FStat (03Dh) Format. . . . . 55
Figure 33. Cell Relaxation Detection . . . . . 55
Figure 34. Flowchart of Operating Modes . . . . . 56
Figure 35. DevName (021h) Format. . . . . 57
Figure 36. nROMID (1BCh to 1BFh) Format . . . . . 57
Figure 37. PackCfg (0BDh)/nPackCfg (1B5h) Format . . . . . 58
Figure 38. nNVCfg0 (1B8h) Format . . . . . 59
Figure 39. nNVCfg1 (1B9h) Format . . . . . 60
Figure 40. nNVCfg2 (1BAh) Format . . . . . 61
Figure 41. ShdnTimer (03Fh) Format . . . . . 62
Figure 42. nConfig (1B0h) Format . . . . . 62
Figure 43. Config (01Dh) Format . . . . . 62
Figure 44. Config2 (0BBh) Format. . . . . 62
Figure 45. Status (000h) Format . . . . . 64
Figure 46. Status2 (0B0h) Format . . . . . 65
Figure 47. HibCfg (0BAh)/nHibCfg (1B4h) Format . . . . . 65
Figure 48. CommStat (061h) Format . . . . . 66
Figure 49. Cell Balancing Window of Operation . . . . . 66
Figure 50. 2S and 3S Balancing Circuits . . . . . 67
Figure 51. Cell Balancing Duty Cycle . . . . . 68
Figure 52. MaxMinVolt (01Bh)/nMaxMinVolt (1ACh) Format . . . . . 70
Figure 53. MaxMinCurr (01Ch)/nMaxMinCurr (1ABh) Format . . . . . 72
Figure 54. nCGain Register (1C8h) Format . . . . . 72
Figure 55. MaxMinTemp (01Ah)/nMaxMinTemp (1ADh) Format . . . . . 75
Figure 56. Overcurrent Comparator Diagram . . . . . 77
Figure 57. ODSCTh Register (0F2h) and nODSCTh Register (18Eh) Formats. . . . . 77
Figure 58. ODSCCfg Register (0F3h) and nODSCCfg Register (18Fh) Formats . . . . . 78
Figure 59. VAlrtTh (001h)/nVAlrtTh (1C0h) Format . . . . . 79
Figure 60. TAlrtTh (002h)/nTAlrtTh (1C1h) Format. . . . . 79
Figure 61. SAlrtTh (003h)/nSAlrtTh (1C2h) Format . . . . . 79

## LIST OF FIGURES (CONTINUED)

Figure 62. IAlrtTh (0B4h)/nIAlrtTh (1C3h) Format . . . . . 80
Figure 63. Shadow RAM and Nonvolatile Memory Relationship . . . . . 83
Figure 64. Format of LOCK Command . . . . . 89
Figure 65. Format of Lock Register (07Fh) . . . . . 89
Figure 66. nSBSCfg (1BBh) Format . . . . . 91
Figure 67. Procedure to Verify a Battery . . . . . 92
Figure 68. Battery Authentication Without a Host-Side Secret . . . . . 93
Figure 69. Single Step Secret Generation Example . . . . . 94
Figure 70. Multistep Secret Generation Example . . . . . 95
Figure 71. 2-Wire Bus Interface Circuitry . . . . . 98
Figure 72. 2-Wire Bus Timing Diagram . . . . . 99
Figure 73. Example I²C Write Data Communication Sequence . . . . . 100
Figure 74. Example I²C Read Data Communication Sequence . . . . . 101
Figure 75. Example SBS Write Word Communication Sequence . . . . . 101
Figure 76. Example SBS Read Word Communication Sequence . . . . . 102
Figure 77. Example SBS Read Block Communication Sequence . . . . . 103
Figure 78. PEC CRC Generation Block Diagram . . . . . 103
Figure 79. 1-Wire Bus Interface Circuitry . . . . . 104
Figure 80. 1-Wire Initialization Sequence . . . . . 105
Figure 81. 1-Wire Write and Read Time Slots . . . . . 106
Figure 82. Example 1-Wire Communication Sequences . . . . . 108
Figure 83. Sample Board Layout . . . . . 113

## LIST OF TABLES

Table 1. ModelGauge Register Standard Resolutions . . . . . 25
Table 2. Minimum and Maximum Cell Sizes for Age Forecasting . . . . . 43
Table 3. Life Logging Register Summary . . . . . 44
Table 4. Decoding History Page Flags . . . . . 47
Table 5. Reading History Data . . . . . 47
Table 6. Reading History Page Flags . . . . . 48
Table 7. Recommended nRSense Register Values for Common-Sense Resistors . . . . . 58
Table 8. Fuel Gauge Temperature Input . . . . . 59
Table 9. Cell Balancing Thresholds . . . . . 67
Table 10. Voltage Measurement Timing . . . . . 69
Table 11. Current Measurement Timing . . . . . 71
Table 12. Current Measurement Range and Resolution vs. Sense Resistor Value . . . . . 71

## LIST OF TABLES (CONTINUED)

Table 13. Copper Trace Sensing . . . . . 73
Table 14. Temperature Measurement Timing . . . . . 74
Table 15. Register Settings for Common Thermistor Types . . . . . 75
Table 16. Top-Level Memory Map . . . . . 80
Table 17. Individual Registers. . . . . 81
Table 18. ModelGauge m5 Register Memory Map . . . . . 81
Table 19. Nonvolatile Register Memory Map . . . . . 82
Table 20. History Recall Command Functions. . . . . 83
Table 21. Number of Remaining Config Memory Updates . . . . . 84
Table 22. Nonvolatile Memory Configuration Options . . . . . 85
Table 23. SBS Register Space Memory Map . . . . . 90
Table 24. SBS to Nonvolatile Memory Mapping . . . . . 91
Table 25. nCGain Register Settings to Meet SBS Compliance . . . . . 92
Table 26. Number of Remaining Secret Updates . . . . . 96
Table 27. 2-Wire Slave Addresses . . . . . 99
Table 28. Valid SBS Read Block Registers . . . . . 102
Table 29. 1-Wire Net Address Format . . . . . 105
Table 30. All Function Commands . . . . . 109

## Absolute Maximum Ratings

$V_{BATT}$ to CSN (MAX17201/MAX17211).....-0.3V to +6V
$V_{BATT}$ to CSN (MAX17205/MAX17215) .....-0.3V to +22V
ALRT1 to CSN.....-0.3V to +17V
CELL1 to CSN..... -0.3V to $V_{CELL2}$ + 0.3V
CELL2 to CELL1 ..... -0.3V to $V_{BATT}$ + 0.3V
REG3 to $V_{BATT}$ (MAX17201/MAX17211) ..... 0V to 0V
REG3 to CSN (MAX17205/MAX17215).....-0.3V to +6V
AIN1, AIN2 to CSN.....-0.3V to +6V
THRM, CELLx to CSN ..... -0.3V to $V_{REG3}$ + 0.3V
REG2 to CSN.....-0.3V to +2.2V
CSP to CSN .....-2V to +2V
DQ/SDA, OD/SCL to CSN.....-0.3V to +6V
Continuous Source Current for THRM .....20mA
Continuous Sink Current for DQ/SDA, ALRT1 .....20mA
Continuous Sink Current for BATT, CELL1, CELL2 .....50mA
Operating Temperature Range..... -40°C to +85°C
Storage Temperature Range..... -55°C to +125°C
Lead Temperature (soldering 10s) .....+300°C
Soldering Temperature (reflow).....+260°C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## Package Information

For the latest package outline information and land patterns (footprints), go to **www.maximintegrated.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

**14 TDFN-EP**

| | |
|---|---|
| Package Code | T1433+2C |
| Outline Number | **21-0137** |
| Land Pattern Number | **90-0063** |
| **Thermal Resistance, Single Layer Board:** | |
| Junction-to-Ambient ($\theta_{JA}$) | 54°C/W |
| Junction-to-Case Thermal Resistance ($\theta_{JC}$) | 8°C/W |
| **Thermal Resistance, Four Layer Board:** | |
| Junction-to-Ambient ($\theta_{JA}$) | 41°C/W |
| Junction-to-Case Thermal Resistance ($\theta_{JC}$) | 8°C/W |

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to **www.maximintegrated.com/thermal-tutorial**.

## Electrical Characteristics

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| **POWER SUPPLY** | | | | | | |
| Supply Voltage | $V_{BATT}$ | MAX17201/MAX17211 (Notes 1, 2) | 2.3 | | 4.9 | V |
| | | MAX17205/MAX17215 (Notes 1, 2) | 4.2 | | 20 | |
| Startup Voltage | $V_{BATT}$ | MAX17201/MAX17211 (Note 1) | | 2.85 | 3.0 | V |
| Shutdown Supply Current | $I_{DD0}$ | Single cell, shutdown mode (Note 3) | | 0.7 | 1.5 | µA |
| | | Multiple cell, shutdown mode (Note 3) | | 1.5 | 3.0 | |

## Electrical Characteristics (continued)

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Hibernate Supply Current | $I_{DD1}$ | Hibernate mode average current, single cell (Note 3) | | 9 | 20 | µA |
| | | Hibernate mode average current, multiple cell (Note 3) | | 12 | 25 | |
| Active Supply Current | $I_{DD2}$ | MAX17201/MAX17211, not including thermistor measurement current (Note 3) | | 18 | 35 | µA |
| | | MAX17205/1MAX17215, not including thermistor measurement current (Note 3) | | 25 | 40 | |
| Regulation Voltage | $V_{REG2}$ | | | 1.8 | | V |
| | $V_{REG3}$ | MAX17205/MAX17215 only | | 3.4 | | |
| **ANALOG-TO-DIGITAL CONVERSION** | | | | | | |
| Voltage Measurement Error | $V_{GERR}$ | $T_A$ = +25°C (Note 4) | -12.5 | | +12.5 | mV |
| | | (Note 4) | -25 | | +25 | |
| | | $T_A$ = +25°C (Note 5) | -12.5 | | +12.5 | |
| | | (Note 5) | -25 | | +25 | |
| | $VB_{GERR}$ | $T_A$ = +25°C (Note 6) | -30 | | +30 | |
| | | (Note 6) | -100 | | +100 | |
| | $VX_{GERR}$ | $T_A$ = +25°C (Note 7) | -0.2 | | +0.2 | % of Reading |
| | | (Note 7) | -0.5 | | +0.5 | |
| Voltage Measurement Resolution | $V_{LSB}$ | Individual cell | | 78.125 | | µV |
| | $VB_{LSB}$ | $V_{BATT}$ pin | | 1.25 | | mV |
| | $VX_{LSB}$ | CELLx pin | | 78.125 | | µV |
| Voltage Measurement Range | $V_{FS}$ | Individual cell | 2.3 | | 4.9 | V |
| | $VB_{FS}$ | $V_{BATT}$ pin | 4.2 | | 20.0 | |
| | $VX_{FS}$ | CELLx pin | 0.92 | | 2.0 | |
| Current Measurement Offset Error | $I_{OERR}$ | $V_{CSP}$ = 0V, long-term average (Note 2) | -2.0 | -0.7 | +0.5 | µV |
| Current Measurement Gain Error | $I_{GERR}$ | CSP between -50mV and +50mV | -1 | | +1 | % of reading |
| Current Measurement Resolution | $I_{LSB}$ | | | 1.5625 | | µV |
| Current Measurement Range | $I_{FS}$ | | | ±51.2 | | mV |
| Internal Temperature Measurement Error | $TI_{GERR}$ | | | ±1 | | °C |
| Internal Temperature Measurement Resolution | $TI_{LSB}$ | AIN1 to CSN, AIN2 to CSN | | 0.00391 | | °C |

## Electrical Characteristics (continued)

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Auxiliary Ratiometric Measurement Error | $TE_{GERR}$ | | -0.5 | | +0.5 | % of reading |
| Auxiliary Ratiometric Measurement Resolution | $TE_{LSB}$ | | | 0.001526 | | % |
| **INPUT/OUTPUT** | | | | | | |
| Output Drive High, THRM | $V_{OH}$ | $I_{OH}$ = -1mA, $V_{REG3}$ = 2.3V | $V_{REG3}$ - 0.1 | | | V |
| Output Drive Low, ALRT1, SDA/DQ | $V_{OL}$ | $I_{OL}$ = 4mA, $V_{REG3}$ = 2.3V | | | 0.4 | V |
| Input Logic-High, SCL/OD, SDA/DQ | $V_{IH}$ | | 1.5 | | | V |
| Input Logic-Low, SCL/OD, SDA/DQ | $V_{IL}$ | | | | 0.44 | V |
| Battery-Detach Detection Threshold | $V_{DET}$ | AIN1 to CSN, as a fraction of the voltage between THRM and CSN, AIN1 rising | 91 | 95 | 99 | % |
| Battery-Detach Detection Threshold Hysteresis | $V_{DET-HYS}$ | AIN1 falling | | 1 | | % |
| Battery-Detach Comparator Delay | $t_{TOFF}$ | AIN1 step from 70% to 100% of THRM voltage to ALRT1 falling, Config register Alrtp = 0, Ber = 1, FTHRM = 1 | | | 100 | µs |
| **COMPARATORS** | | | | | | |
| Overcurrent Threshold Offset Error | $OC_{OE}$ | OD or SC comparator | -2.5 | | +2.5 | mV |
| Overcurrent Threshold Gain Error | $OC_{GE}$ | OD or SC comparator | -5.0 | | +5.0 | % of threshold |
| Over Current Comparator Delay | $OC_{DLY}$ | OD or SC comparator, 20mV minimum input overdrive, delay configured to minimum | | 2 | | µs |
| **RESISTANCE AND LEAKAGE** | | | | | | |
| Leakage Current, AIN1, AIN2 | $I_{LEAK}$ | AIN1, AIN2 < REG3 | -0.2 | | 0.2 | µA |
| Leakage Current, CELLx | $I_{LEAK}$ | $V_{CELLx}$ < 2.0V (Note 2) | -60 | ±5 | +60 | nA |
| Leakage Current, CELL1, CELL2, CSP, ALRT1, THRM | $I_{LEAK}$ | $V_{ALRT1}$ < 15V, THRM < REG3 | -1 | | +1 | µA |
| Input Resistance, CELL2, CELL1, CSP | $R_{VAD}$ | Resistance to CSN during voltage sampling | | 1 | | MΩ |
| Input Resistance, CELLx | $R_{CELLx}$ | | | 400 | | MΩ |

## Electrical Characteristics (continued)

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Cell-Balancing Resistance | $R_{BAL}$ | $V_{BATT}$ = 12.6V, $I_{BAL}$ = 50mA, between $V_{BATT}$-CELL2, CELL2-CELL1, and CELL1-CSN | 3 | 9 | 20 | Ω |
| Input Pulldown Current | $I_{PD}$ | $V_{SDA}$, $V_{SCL}$ pins = 0.4V | 0.05 | 0.2 | 0.4 | µA |
| **TIMING** | | | | | | |
| Time-Base Accuracy | $t_{ERR}$ | $T_A$= +25°C | -1 | | +1 | % |
| SHA Calculation Time | $t_{SHA}$ | | | 4.5 | 10 | ms |
| THRM Precharge Time | $t_{PRE}$ | Time between turning on the THRM pullup and AIN1 or AIN2 analog-to-digital conversions | 8.48 | | | ms |
| Power-on-Reset Time | $t_{POR}$ | (Note 2) | | | 10 | ms |
| Task Period | $t_{TP}$ | | | 351.5 | | ms |
| **NONVOLATILE MEMORY** | | | | | | |
| Nonvolatile Access Voltage | $V_{NVM}$ | For block programming and recalling, applied on $V_{BATT}$ (MAX17201/MAX17211) | 3.0 | | | V |
| | | For block programming and recalling, applied on $V_{BATT}$ (MAX17205/MAX17215) | 4.2 | | | |
| Programming Supply Current | $I_{PROG}$ | Current from $V_{BATT}$ for block programming | | 4 | 10 | mA |
| Block Programming Time | $t_{BLOCK}$ | | | 368 | 7360 | ms |
| Page Programming Time | $t_{UPDATE}$ | SHA secret update or learned parameters update | | 64 | 1280 | ms |
| Nonvolatile Memory Recall Time | $t_{RECALL}$ | | | | 5 | ms |
| Write Capacity, Configuration Memory | $n_{CONFIG}$ | (Notes 2, 8, 9) | | 7 | | writes |
| Write Capacity, SHA Secret | $n_{SECRET}$ | (Notes 2, 8, 9) | | 5 | | writes |
| Write Capacity, Learned Parameters | $n_{LEARNED}$ | (Notes 2, 8, 9) | | 202 | | writes |
| Data Retention | $t_{NV}$ | (Note 2) | 10 | | | years |
| **1-WIRE INTERFACE, REGULAR SPEED** | | | | | | |
| Time Slot | $t_{SLOT}$ | | 60 | | 120 | µs |
| Recovery Time | $t_{REC}$ | | 1 | | | µs |
| Write-0 Low Time | $t_{LOW0}$ | | 60 | | 120 | µs |
| Write-1 Low Time | $t_{LOW1}$ | | 1 | | 15 | µs |
| Read-Data Valid | $t_{RDV}$ | | | | 15 | µs |

## Electrical Characteristics (continued)

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Reset-Time High | $t_{RSTH}$ | | 480 | | | µs |
| Reset-Time Low | $t_{RSTL}$ | | 480 | | | µs |
| Presence-Detect High | $t_{PDH}$ | | 15 | | 60 | µs |
| Presence-Detect Low | $t_{PDL}$ | | 60 | | 240 | µs |
| **1-WIRE INTERFACE, OVERDRIVE SPEED** | | | | | | |
| Time Slot | $t_{SLOT}$ | | 6 | | 16 | µs |
| Recovery Time | $t_{REC}$ | | 1 | | | µs |
| Write-0 Low Time | $t_{LOW0}$ | | 6 | | 16 | µs |
| Write-1 Low Time | $t_{LOW1}$ | | 1 | | 2 | µs |
| Read-Data Valid | $t_{RDV}$ | | | | 2 | µs |
| Reset-Time High | $t_{RSTH}$ | | 48 | | | µs |
| Reset-Time Low | $t_{RSTL}$ | | 48 | | | µs |
| Presence-Detect High | $t_{PDH}$ | | 2 | | 6 | µs |
| Presence-Detect Low | $t_{PDL}$ | | 8 | | 24 | µs |
| **2-WIRE INTERFACE** | | | | | | |
| SCL Clock Frequency | $f_{SCL}$ | (Note 10) | 0 | | 400 | kHz |
| Bus Free Time Between a STOP and START Condition | $t_{BUF}$ | | 1.3 | | | µs |
| Hold Time (Repeated) START Condition | $t_{HD:STA}$ | (Note 11) | 0.6 | | | µs |
| Low Period of SCL Clock | $t_{LOW}$ | | 1.3 | | | µs |
| High Period of SCL Clock | $t_{HIGH}$ | | 0.6 | | | µs |
| Setup Time for a Repeated START Condition | $t_{SU:STA}$ | | 0.6 | | | µs |
| Data Hold Time | $t_{HD:DAT}$ | (Notes 12, 13) | 0 | | 0.9 | us |
| Data Setup Time | $t_{SU:DAT}$ | (Note 12) | 100 | | | ns |
| Rise Time of Both SDA and SCL Signals | $t_R$ | | 5 | | 300 | ns |
| Fall Time of Both SDA and SCL Signals | $t_F$ | | 5 | | 300 | ns |
| Setup Time for STOP Condition | $t_{SU:STO}$ | | 0.6 | | | µs |
| Spike Pulse Width Suppressed by Input Filter | $t_{SP}$ | (Note 14) | | | 50 | ns |

## Electrical Characteristics (continued)

($V_{BATT}$ = 2.3V to 4.9V (MAX17201/MAX17211) 4.2V to 20V (MAX17205/MAX17215), $T_A$ = -40°C to 85°C, unless otherwise noted. Typical values are $T_A$ = +25°C. See Figure 1, Figure 2, and Figure 3. Limits are 100% tested at $T_A$ = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.)

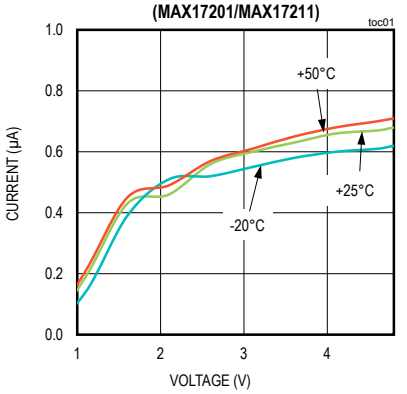
| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Capacitive Load for Each Bus Line | $C_B$ | | | | 400 | pF |
| SCL, SDA Input Capacitance | $C_{BIN}$ | | | 6 | | pF |

**Note 1:** All voltages are referenced to CSN.
**Note 2:** Specification is guaranteed by design (GBD), and not production tested.
**Note 3:** $T_A$ < +50°C, $V_{BATT}$ = 4.9V for single cell or 20V for multiple cell.
**Note 4:** Single cell, CELL1 to CSP cell voltage between 2.3V and 4.9V.
**Note 5:** Multiple cell, $V_{BATT}$ to CELL2, CELL2 to CeLL1, or CELL1 to CSP, cell voltages between 2.3V and 4.9V; for voltages between 2.3V and 4.9V; for two cells, CELL2 must be shorted to CELL1.
**Note 6:** Multiple cell, total $V_{BATT}$ voltage, $V_{BATT}$ = 4.2V to 20V.
**Note 7:** The MAX17205/MAX17215 only CELLx to CSP, per cell voltage of 2.3V to 4.9V.
**Note 8:** Write capacity numbers shown have one write subtracted for the initial write performed during manufacturing test to set nonvolatile memory to a known value.
**Note 9:** Due to the nature of one-time programmable memory, write capacity cannot be production tested. Follow the nonvolatile memory and SHA secret update procedures detailed in the data sheet.
**Note 10:** Timing must be fast enough to prevent the IC from entering shutdown mode due to bus low for a period greater than the shutdown timer setting.
**Note 11:** $f_{SCL}$ must meet the minimum clock low time plus the rise/fall times.
**Note 12:** The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the low period ($t_{LOW}$) of the SCL signal.
**Note 13:** This device internally provides a hold time of at least 100ns for the SDA signal (referred to the minimum $V_{IH}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.
**Note 14:** Filters on SDA and SCL suppress noise spikes at the input buffers and delay the sampling instant.

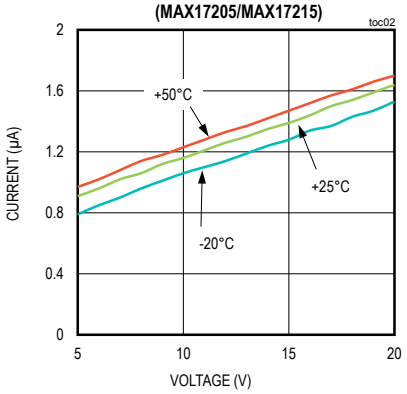
## Typical Operating Characteristics

($T_A$ = +25°C, unless otherwise noted.)

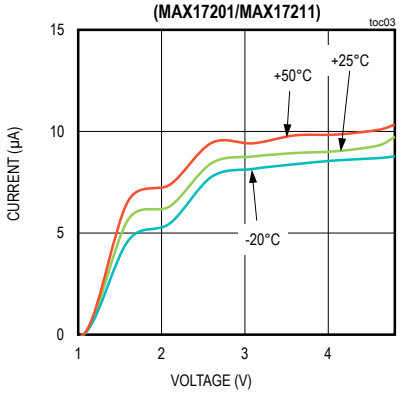
SHUTDOWN CURRENT vs. SUPPLY VOLTAGE
(MAX17201/MAX17211)



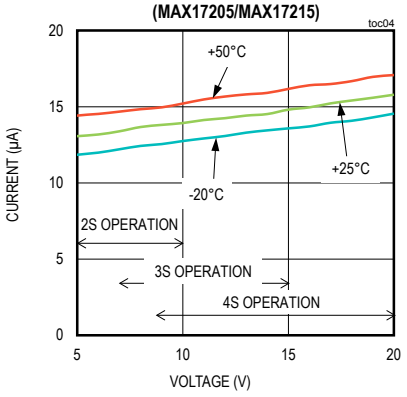

SHUTDOWN CURRENT vs. SUPPLY VOLTAGE
(MAX17205/MAX17215)



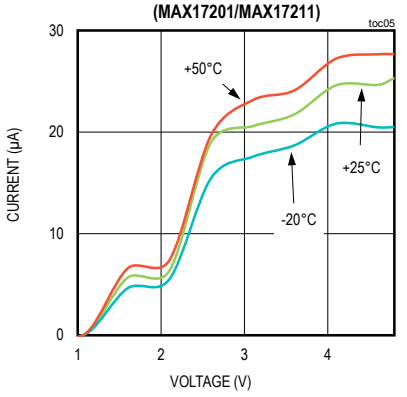

HIBERNATE CURRENT vs. SUPPLY VOLTAGE
(MAX17201/MAX17211)



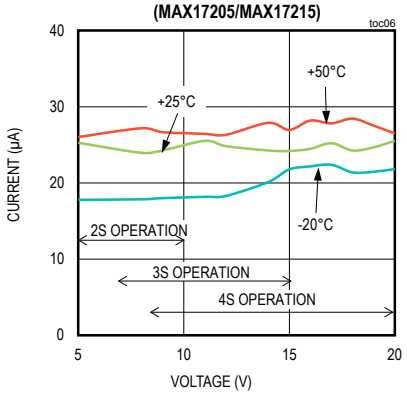

HIBERNATE CURRENT vs. SUPPLY VOLTAGE
(MAX17205/MAX17215)



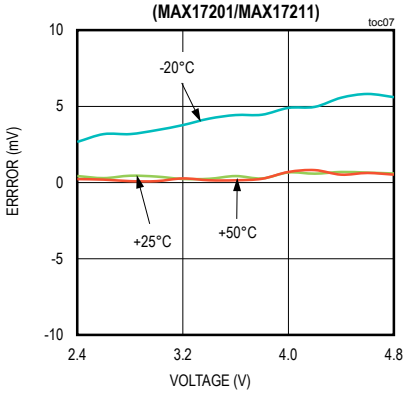

ACTIVE CURRENT vs. SUPPLY VOLTAGE
(MAX17201/MAX17211)



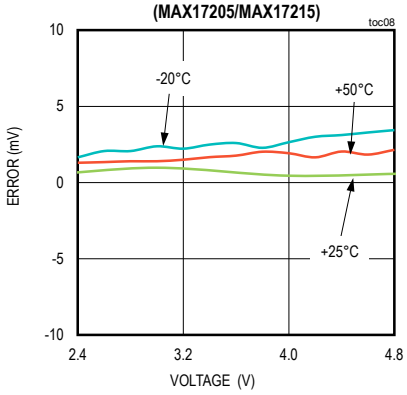

ACTIVE CURRENT vs. SUPPLY VOLTAGE
(MAX17205/MAX17215)



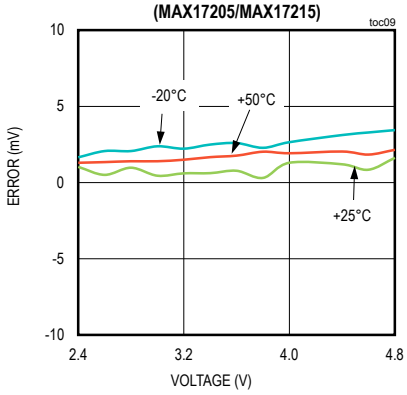

CELL1 VOLTAGE ADC ERROR
(MAX17201/MAX17211)




CELL1 VOLTAGE ADC ERROR
(MAX17205/MAX17215)



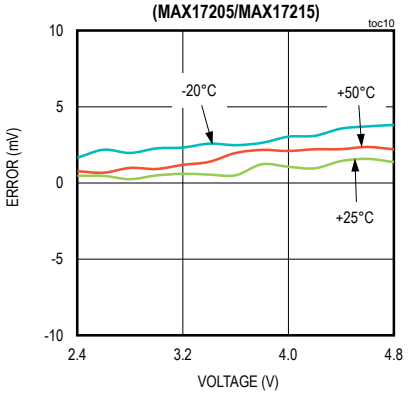

CELL2 VOLTAGE ADC ERROR
(MAX17205/MAX17215)



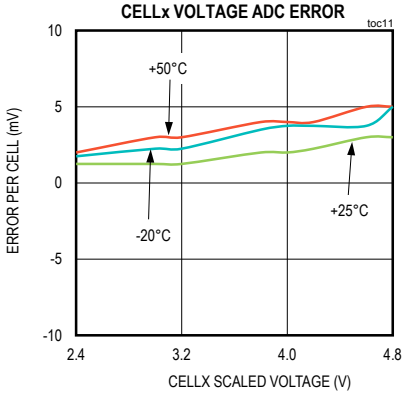
## Typical Operating Characteristics (continued)

($T_A$ = +25°C, unless otherwise noted.)

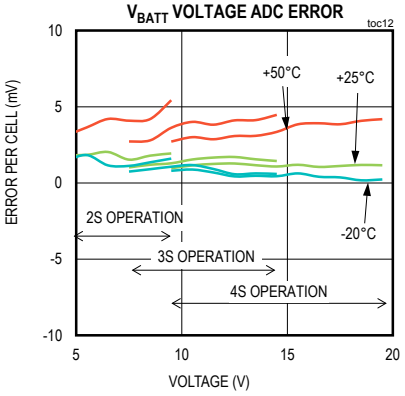
CELL3 VOLTAGE ADC ERROR (MAX17205/MAX17215)



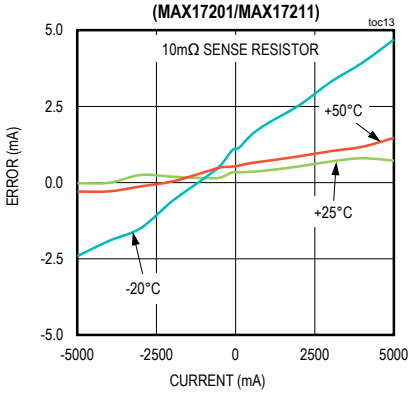

CELLx VOLTAGE ADC ERROR



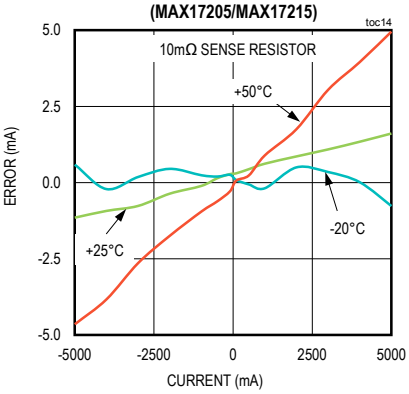

$V_{BATT}$ VOLTAGE ADC ERROR



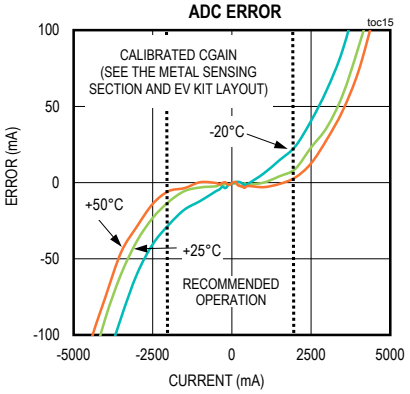

CURRENT ADC ERROR (MAX17201/MAX17211)



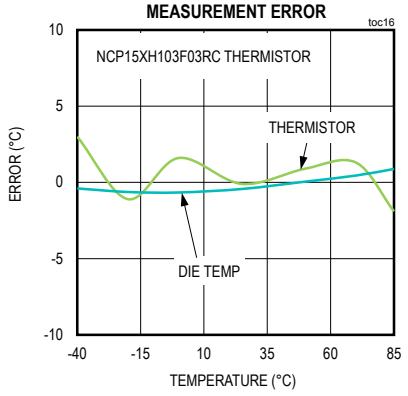

CURRENT ADC ERROR (MAX17205/MAX17215)



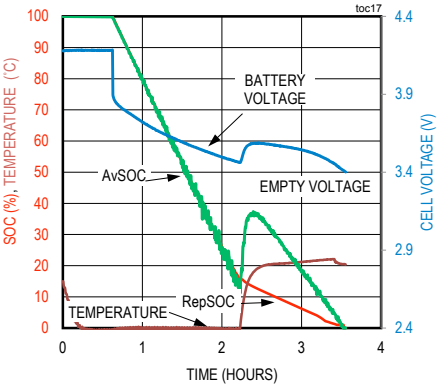

METAL TRACE CURRENT SENSE ADC ERROR



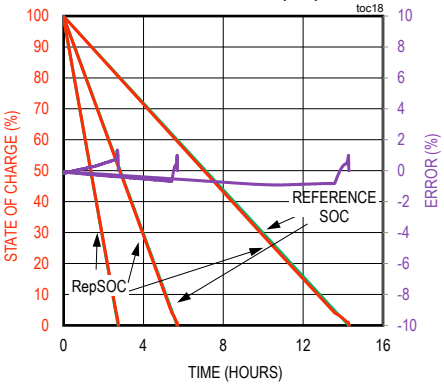

DIE TEMPERATURE AND THERMISTOR MEASUREMENT ERROR




RESPONSE TO TEMPERATURE TRANSIENT AT CONSTANT CURRENT LOAD



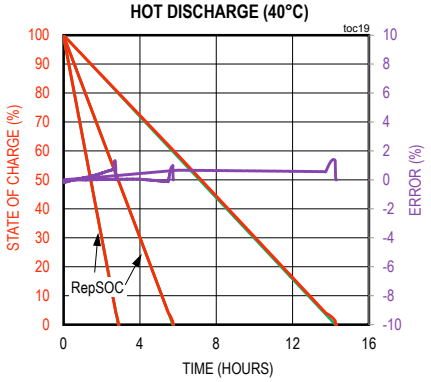

COLD DISCHARGE (0°C)



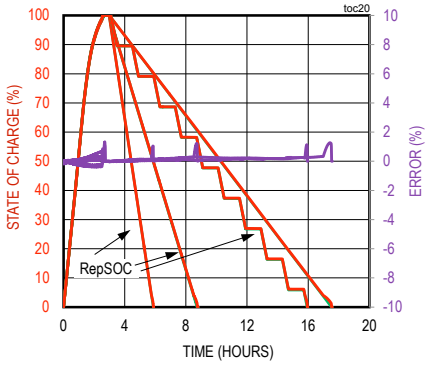
## Typical Operating Characteristics (continued)

($T_A$ = +25°C, unless otherwise noted.)

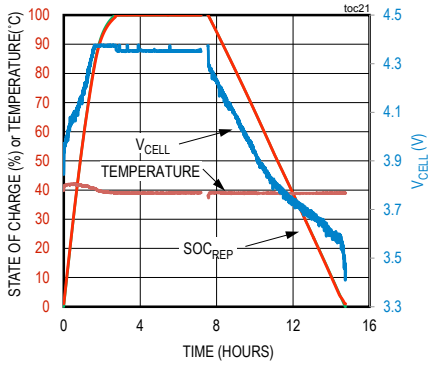
HOT DISCHARGE (40°C)



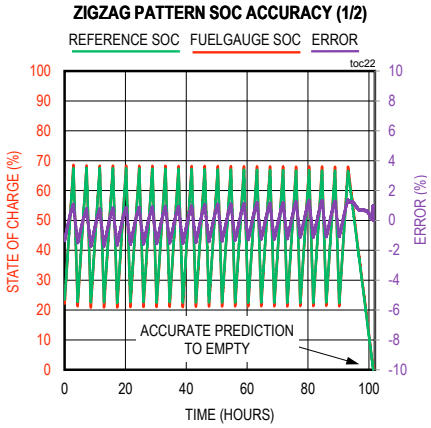

CHARGE AND DISCHARGE (+20°C)



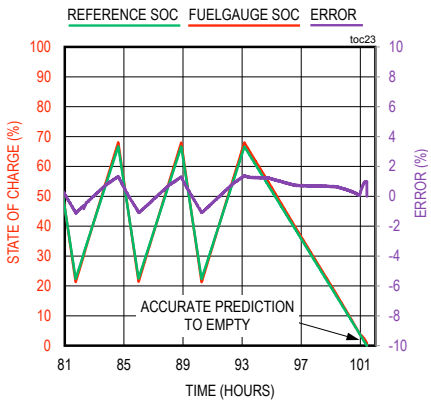

CHARGE AND DISCHARGE IN ACTUAL SYSTEM



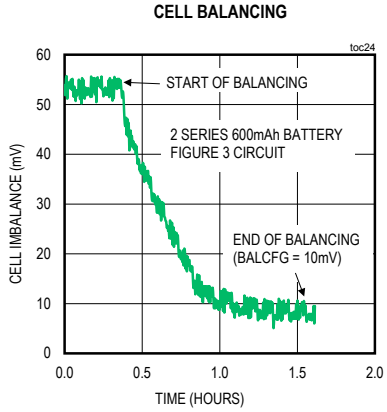

ZIGZAG PATTERN SOC ACCURACY (1/2)




ZIGZAG PATTERN SOC ACCURACY (2/2)



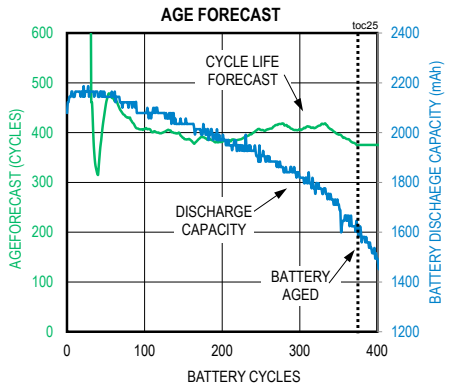

CELL BALANCING



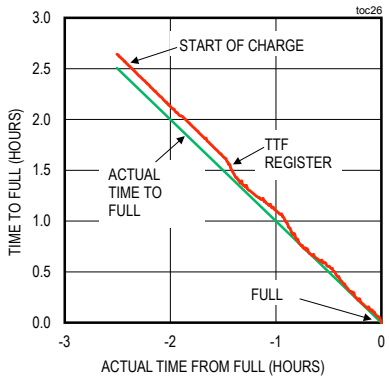
## Typical Operating Characteristics (continued)

($T_A$ = +25°C, unless otherwise noted.)

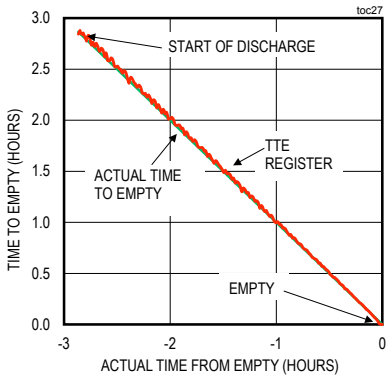
AGE FORECAST



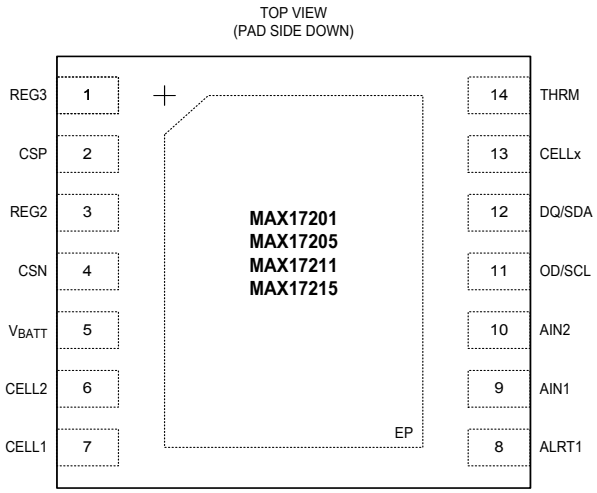

TIME TO FULL PERFORMANCE




TIME TO EMPTY PERFORMANCE



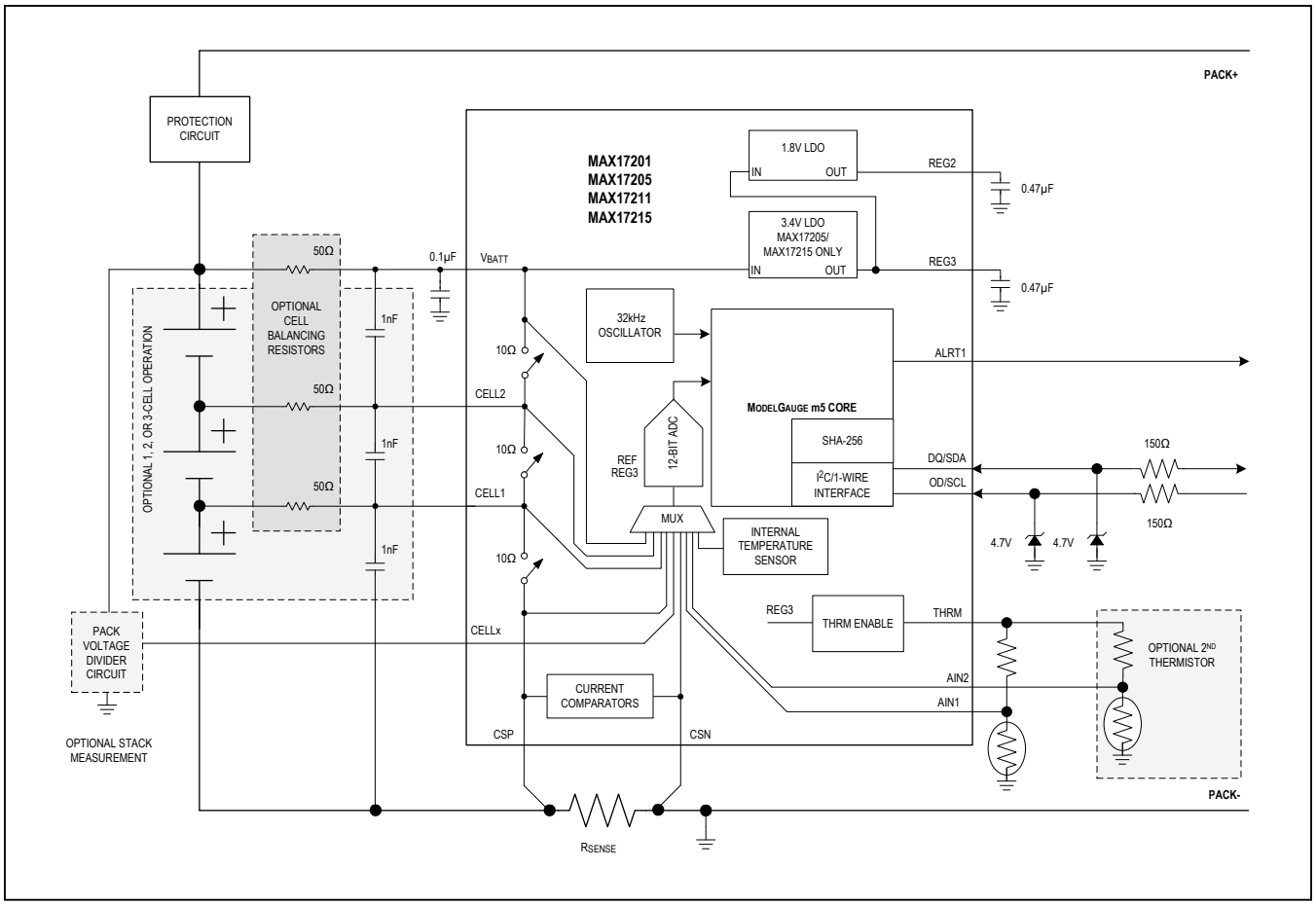
## Pin Configuration




## Pin Description

| PIN | NAME | FUNCTION |
|---|---|---|
| 1 | REG3 | Internal 3.4V Regulator Output. For the MAX17205/MAX17215, bypass with an external 0.47µF capacitor. For the MAX17201/MAX17211, connect REG3 to $V_{BATT}$. |
| 2 | CSP | Current Measurement Positive Sense Point. Kelvin connect to cell side of sense resistor. |
| 3 | REG2 | Internal 1.8V Regulator Output. Bypass with an external 0.47µF capacitor to CSN. |
| 4 | CSN | Device Ground and Current Measurement Negative Sense Point. Kelvin connect to load side of sense resistor. |
| 5 | $V_{BATT}$ | Power-Supply and Battery Voltage Sense Input. Connect to positive terminal of cell stack. Bypass with RC filter to CSN. |
| 6 | CELL2 | Voltage Sense Input for Measuring Cell Voltage of Second or Middle Cell. Series resistance determines balancing current. Also acts as the external divider gate drive when measuring pack voltage on CELLx pin. |
| 7 | CELL1 | Voltage Sense Input for Measuring Voltage of Bottom Cell. Series resistance determines balancing current. |
| 8 | ALRT1 | Programmable Alert Output |
| 9 | AIN1 | Auxiliary Voltage Input 1. Auxiliary voltage input from external thermal-measurement network. |
| 10 | AIN2 | Auxiliary Voltage Input 2. Auxiliary voltage input from external thermal-measurement network. |
| 11 | OD/SCL | Serial Clock Input for $I^2C$ Communication or Speed Selection for 1-Wire Communication. Input only. For $I^2C$ communication, connect to the clock terminal of the battery pack. Connect to CSN for standard speed 1-wire communication. Connect to REG3 pin for overdrive 1-wire communication. OD/SCL has an internal pulldown (IPD) for sensing pack disconnection. |
| 12 | DQ/SDA | Serial Data Input/Output for Both 1-Wire and $I^2C$ communication modes. Open-drain output driver. Connect to the DATA terminal of the battery pack. DQ/SDA has an internal pulldown (IPD) for sensing pack disconnection. |
| 13 | CELLx | High-Impedance Voltage Measurement Channel. Connect to an external voltage divider for measuring cell stacks larger than 4S. |
| 14 | THRM | Thermistor Bias Connection. Connect to the high side of the thermistor resistor-divider circuit. THRM biases to REG3 voltage during AIN1 and AIN2 measurement. |
| — | EP | Exposed Pad. Connect directly to CSN. |

## Functional Diagram




## Detailed Description

The MAX1720x/MAX1721x ultra-low power stand-alone fuel gauge ICs that implement the ModelGauge m5 algorithm without requiring host interaction for configuration. This feature makes the MAX1720x/MAX1721x an excellent pack-side fuel gauge. Voltage of the battery pack is measured at the BATT, CELL2, CELL1, CELLx, and CSP connections. Current is measured by an external sense resistor placed between the CSP and CSN pins. An external resistive voltage-divider network allows the IC to measure temperature of the battery pack by monitoring the AIN1 and AIN2 pins. The THRM pin provides a strong pullup for the resistor-divider that is disabled internally when temperature is not being measured. Internal die temperature of the ICs is also measured. The MAX17201/MAX17211 monitor a single-cell pack. The MAX17205/MAX17215 monitor individual cells of a 2S or 3S pack or the entire stack voltage of any number of multiple-series cells.

The ModelGauge m5 algorithm combines the short-term accuracy and linearity of a coulomb-counter with the long-term stability of a voltage-based fuel gauge, along with temperature compensation to provide industry-leading fuel gauge accuracy. Additionally, the algorithm does not suffer from abrupt corrections that normally occur in coulomb-counter algorithms, since tiny continual corrections are distributed over time. The MAX1720x automatically compensates for aging, temperature, and discharge rate and provides accurate state of charge (SOC) in milliampere-hours (mAh) or percentage (%) over a wide range of operating conditions. Fuel gauge error always converges to 0% as the cell approaches empty. The ICs provide accurate estimation of time-to-empty and time-to-full and provide three methods for reporting the age of the battery: reduction in capacity, increase in battery resistance, and cycle odometer. In addition, age forecasting allows the user to estimate the expected lifespan of the cell.

To prevent battery clones, the ICs integrate SHA-256 authentication with a 160-bit secret key. Every IC also incorporates a 64-bit unique identification number (ROM ID).

Communication to the host occurs over a Maxim 1-wire (MAX17211/MAX17215) or standard I²C interface (MAX17201/MAX17205). OD/SCL is an input from the host, and DQ/SDA is an open-drain I/O pin that requires an external pullup. The ALRT1 pin is an output that can be used as an external interrupt to the host processor if certain application conditions are detected.

For additional reference material, refer to the following application notes:

Application Note 6258: *MAX1720x/MAX1721x Battery Pack Implementation Guide*

Application Note 6259: *MAX1720x/MAX1721x System Side Implementation Guide*

Application Note 6260: *MAX1720x/MAX1721x Host Software Implementation Guide*

## Register Description Conventions

The following sections define standard conventions used throughout the data sheet to describe register functions and device behavior. Any register that does not match one of the following data formats is described as a special register

## Standard Register Formats

Unless otherwise stated during a given register's description, all IC registers follow the same format depending on the type of register. See Table 1 for the resolution and range of any register described hereafter. Note that current and capacity values are displayed as a voltage and must be divided by the sense resistor to determine amps or amp-hours. It is strongly recommended to use the nRSense (1CFh) register to store the sense resistor value for use by host software.

### Device Reset

Device reset refers to any condition that would cause the ICs to recall nonvolatile memory into RAM locations and restart operation of the fuel gauge. Device reset refers to initial power up of the IC, temporary power loss, or reset through the software power-on-reset command.

### Nonvolatile Backup and Initial Value

All configuration register locations have nonvolatile memory backup that can be enabled with control bits in the nNVCfg0, nNVCfg1, and nNVCfg2 registers. If enabled, the associated registers are initialized to their corresponding nonvolatile register value after device reset. If nonvolatile backup is disabled, the register restores to an alternate initial value instead. See each register description for details.

### Register Naming Conventions

Register addresses are described throughout the document as 9-bit internal values from 000h to 1FFh. These addresses must be translated to 8-bit values for the MAX1720x (I²C) or 16-bit external values for the MAX1721x (1-Wire). See the *Memory* section for details.

Register names that start with a lower case n, such as nPackCfg for example, indicate the register is a nonvolatile memory location. Register names that start with a lower case s indicate the register is part of the SBS compliant register block.

## Table 1. ModelGauge Register Standard Resolutions

| REGISTER TYPE | LSB SIZE | MINIMUM VALUE | MAXIMUM VALUE | NOTES |
|---|---|---|---|---|
| Capacity | 5.0µVh/$R_{SENSE}$ | 0.0µVh | 327.675mVh/$R_{SENSE}$ | Equivalent to 0.5mA with a 0.010Ω sense resistor. |
| Percentage | 1/256% | 0.0% | 255.9961% | 1% LSb when reading only the upper byte. |
| Voltage | 0.078125mV | 0.0V | 5.11992V | — |
| Current | 1.5625µV/$R_{SENSE}$ | -51.2mV/$R_{SENSE}$ | 51.1984mV/$R_{SENSE}$ | Signed 2's complement format. Equivalent to 156.25µA with a 0.010Ω sense resistor. |
| Temperature | 1/256°C | -128.0°C | 127.996°C | Signed 2's complement format. 1°C LSb when reading only the upper byte. |
| Resistance | 1/4096Ω | 0.0 Ω | 15.99976Ω | — |
| Time | 5.625s | 0.0s | 102.3984h | — |
| Special | — | — | — | Format details are included with the register description. |

## Alternate Initial and Factory Default Values

If nonvolatile backup of a memory location is disabled, that location initializes to its alternate initial value after reset. The factory default value is the initial value stored in nonvolatile memory at the factory. See Table 22.

## Typical Operating Circuits

The typical operating circuit for the MAX1720x/MAX1721x depends on the series cell count of the cell stack to be monitored and which features of the ICs are desired. The following sections show the five most common typical operating circuits when mounted inside of a cell pack.

### Single-Cell Typical Operating Circuit (MAX17201/MAX17211 Only)

In the single-cell operating circuit example shown in Figure 1, the MAX17201/MAX17211 is mounted outside of the protector circuit to allow communication to the pack even when the protection FETs are disabled. Take care to avoid exceeding the maximum operating voltage on any pin under fault conditions. The single-cell versions of the ICs do not contain a 3.4V regulator. REG3 must be connected directly to the $V_{BATT}$ pin. A single thermistor circuit measures cell temperature leaving the AIN2 input available for additional ADC measurements if desired. Resistors and clamping diodes protect all input pins from ESD. The exposed pad should be connected directly to the CSN pin.

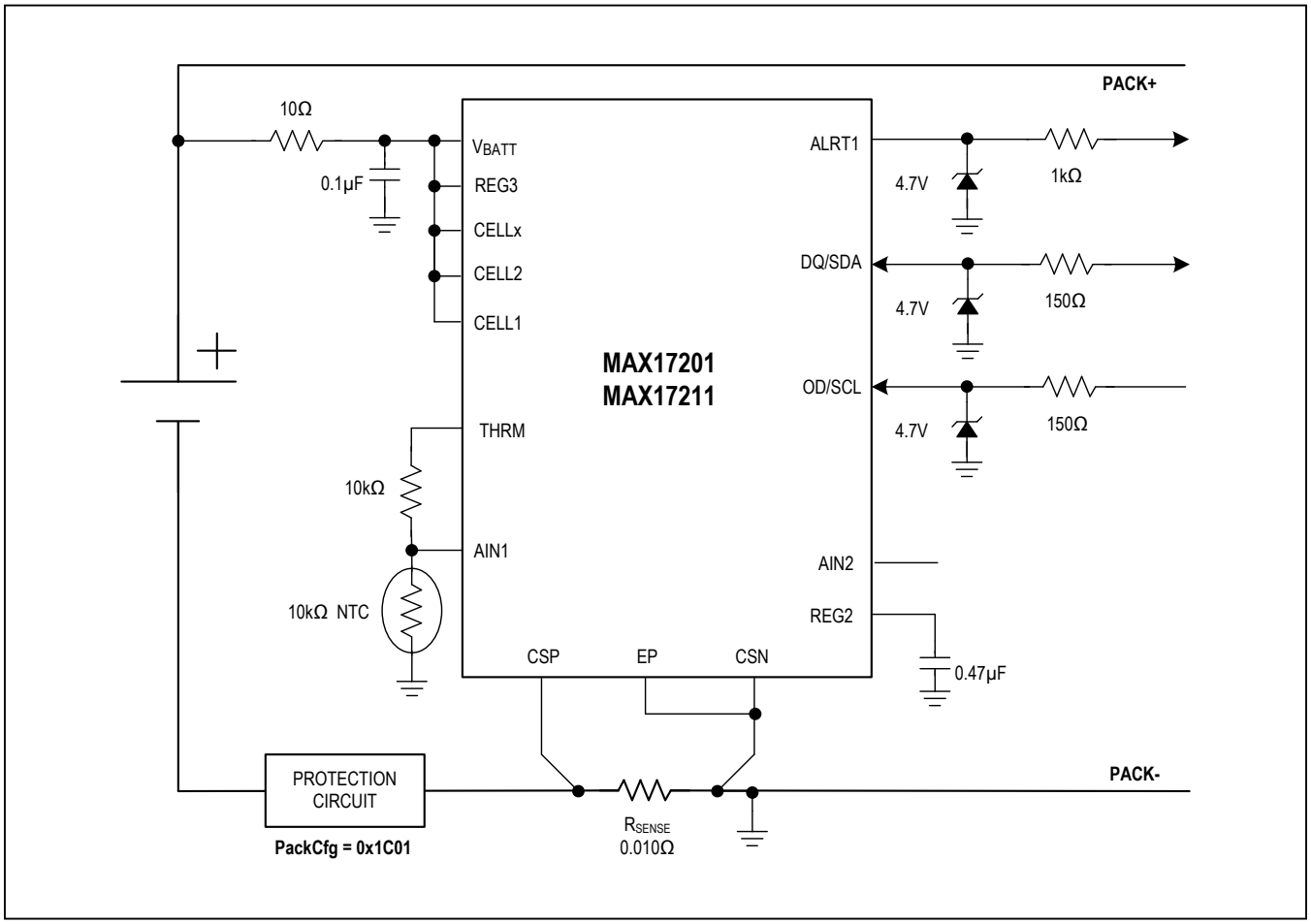



*Figure 1. Single-Cell Schematic*

### Multicell Typical Operating Circuits (MAX17205/MAX17215 Only)

The MAX17205/MAX17215 is designed for use in multicell packs as shown in Figure 2. When used in conjunction with a low-side protection circuit, individual cell measurements and cell balancing are not available. For 2S to 4S configuration packs, the ICs monitor the entire stack voltage from the $V_{BATT}$ pin. Set PackCfg.BtEn = 1, CxEn = 0 and ChEn = 0 for operation with this schematic.

For packs that are larger than 4S, a resistor divider circuit is used. The CELLx pin measures the divided stack voltage and CELL2 controls the divider circuit to limit current drain. Set PackCfg.CxEn = 1, BtEn = 0, ChEn = 0 for operation with this schematic. For cell stacks larger than 4S, a simple regulator circuit clamps the voltage on $V_{BATT}$ below the maximum allowed. Take care to avoid exceeding the maximum operating voltage on any pin under fault conditions.

The resistor divider on CELLx should be calculated as follows:

Resistor from CELLx to CSN: 200kΩ

Resistor from CELLx to $V_{BATT}$: (2.5 * number of cells - 1) x 200kΩ

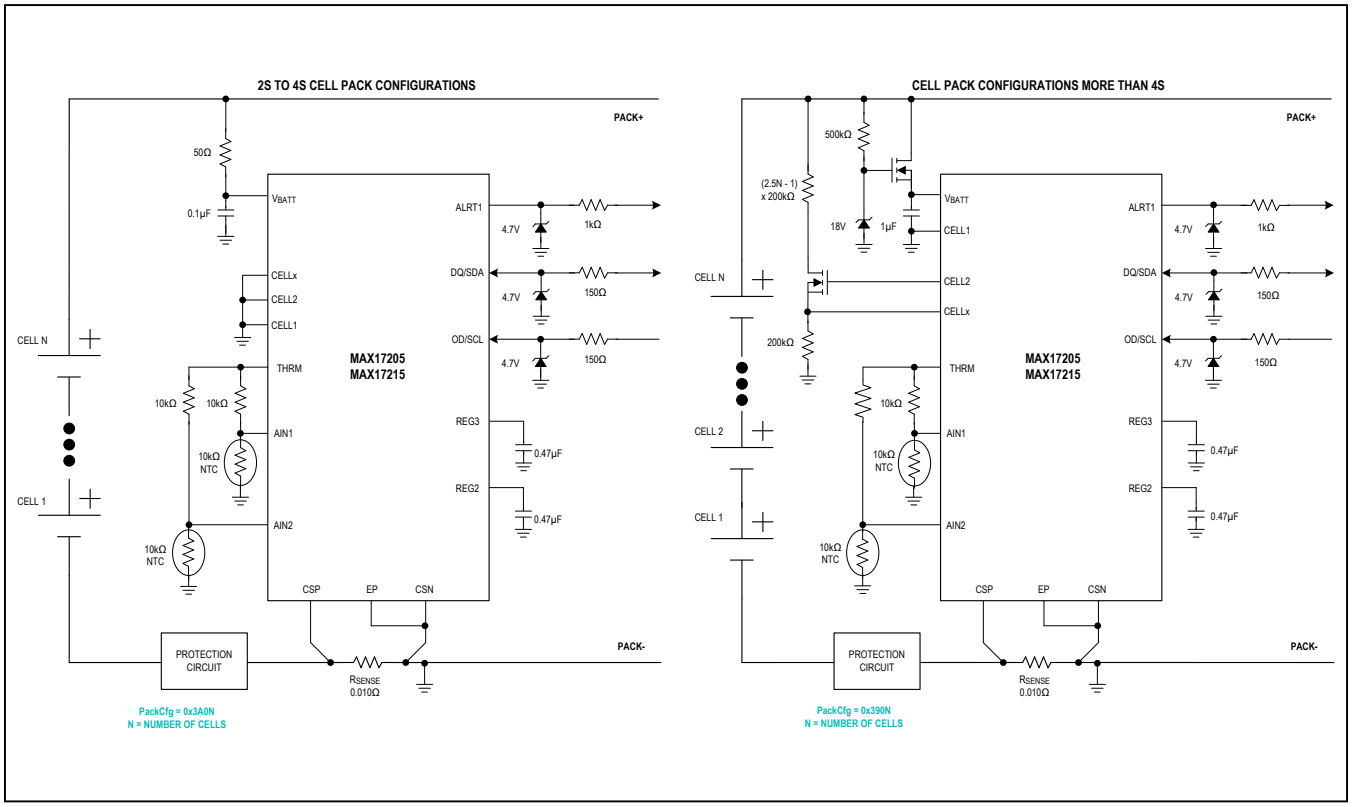



*Figure 2. Multicell Schematics*

### Cell Balancing Typical Operating Circuits (MAX17205/MAX17215 Only)

In multicell configuration packs, if the MAX17205/MAX17215 is used in conjunction with a high-side protection circuit, the ICs can monitor the individual cell voltages and perform cell balancing as shown in Figure 3. In the following 2S and 3S examples, the voltage of each cell is monitored independently. External 100Ω resistors on the CELL1 and CELL2 pins limit current through the balancing circuits to approximately 30mA. Balancing resistors are added to every other pin so that each balancing loop contains a single limiting resistor. Leave the CELLx pin open. Set PackCfg.ChEn = 1 for operation with this schematic.

## ModelGauge m5 Algorithm

Classical coulomb-counter-based fuel gauges have excellent linearity and short-term performance. However, they suffer from drift due to the accumulation of the offset error in the current-sense measurement. Although the offset error is often very small, it cannot be eliminated, causes the reported capacity error to increase over time, and requires periodic corrections. Corrections are usually performed at full or empty. Some other systems also use the relaxed battery voltage to perform corrections. These systems determine the true state of charge (SOC) based on the battery voltage after a long time of no current flow. Both have the same limitation: if the correction condition is not observed over time in the actual application, the error in the system is boundless. The performance of classic coulomb counters is dominated by the accuracy of such corrections. Voltage measurement based SOC estimation has accuracy limitations due to imperfect cell modeling, but does not accumulate offset error over time.

The ICs include an advanced voltage fuel gauge (VFG) that estimates open-circuit voltage (OCV), even during current flow, and simulates the nonlinear internal dynamics of a Li+ battery to determine the SOC with improved accuracy. The model considers the time effects of a battery caused by the chemical reactions and impedance in the battery to determine SOC. This SOC estimation does not accumulate offset error over time.

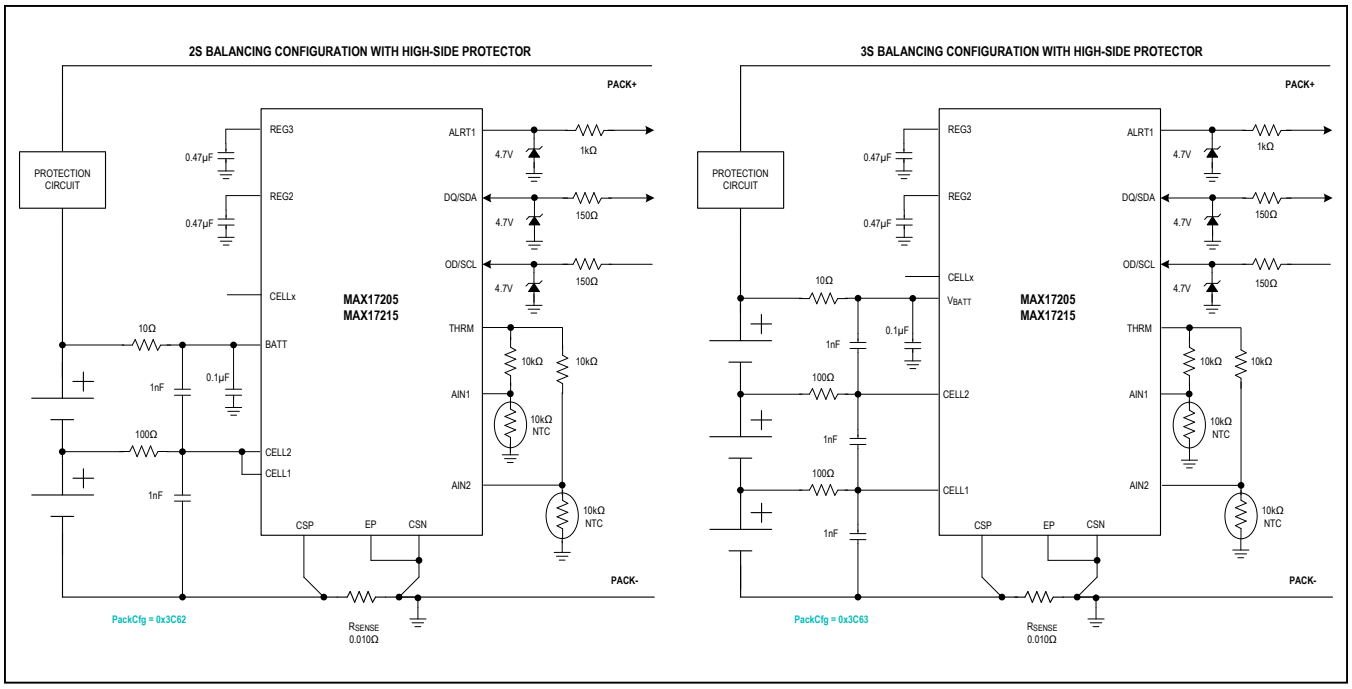



*Figure 3. Cell-Balancing Circuit Schematics*

The ModelGauge m5 algorithm combines a high-accuracy coulomb counter with a VFG. See Figure 4 and Figure 5. The complementary combined result eliminates the weaknesses of both the coulomb counter and the VFG while providing the strengths of both. A mixing algorithm weighs and combines the VFG capacity with the coulomb counter and weighs each result so that both are used optimally to determine the battery state. In this way, the VFG capacity result is used to continuously make small adjustments to the battery state, canceling the coulomb-counter drift.

The ModelGauge m5 algorithm uses this battery state information and accounts for temperature, battery current, age, and application parameters to determine the remaining capacity available to the system. As the battery approaches the critical region near empty, the ModelGauge m5 algorithm invokes a special error correction mechanism that eliminates any error.

The ModelGauge m5 algorithm continually adapts to the cell and application through independent learning routines. As the cell ages, its change in capacity is monitored and updated and the voltage-fuel-gauge dynamics adapt based on cell-voltage behavior in the application.

### ModelGauge m5 Registers

For accurate results, ModelGauge m5 uses information about the cell and the application as well as the real-time information measured by the IC. Figure 6 shows inputs and outputs to the algorithm grouped by category. Analog input registers are the real-time measurements of voltage, temperature, and current performed by the IC. Application-specific registers are programmed by the customer to reflect the operation of the application. The Cell Characterization Information registers hold characterization data that models the behavior of the cell over the operating range of the application. The Algorithm Configuration registers allow the host to adjust performance of the IC for its application. The Learned Information registers allow an application to maintain accuracy of the fuel gauge as the cell ages. The register description sections describe each register function in detail.

### ModelGauge m5 Algorithm Output Registers

The following registers are outputs from the ModelGauge m5 algorithm. The values in these registers become valid 480ms after the ICs are reset.

### RepCap Register (005h)

Register Type: Capacity

Nonvolatile Backup: None

RepCap or reported capacity is a filtered version of the AvCap register that prevents large jumps in the reported value caused by changes in the application such as abrupt changes in temperature or load current. See the Fuel-Guage *Empty Compensation* section for details.

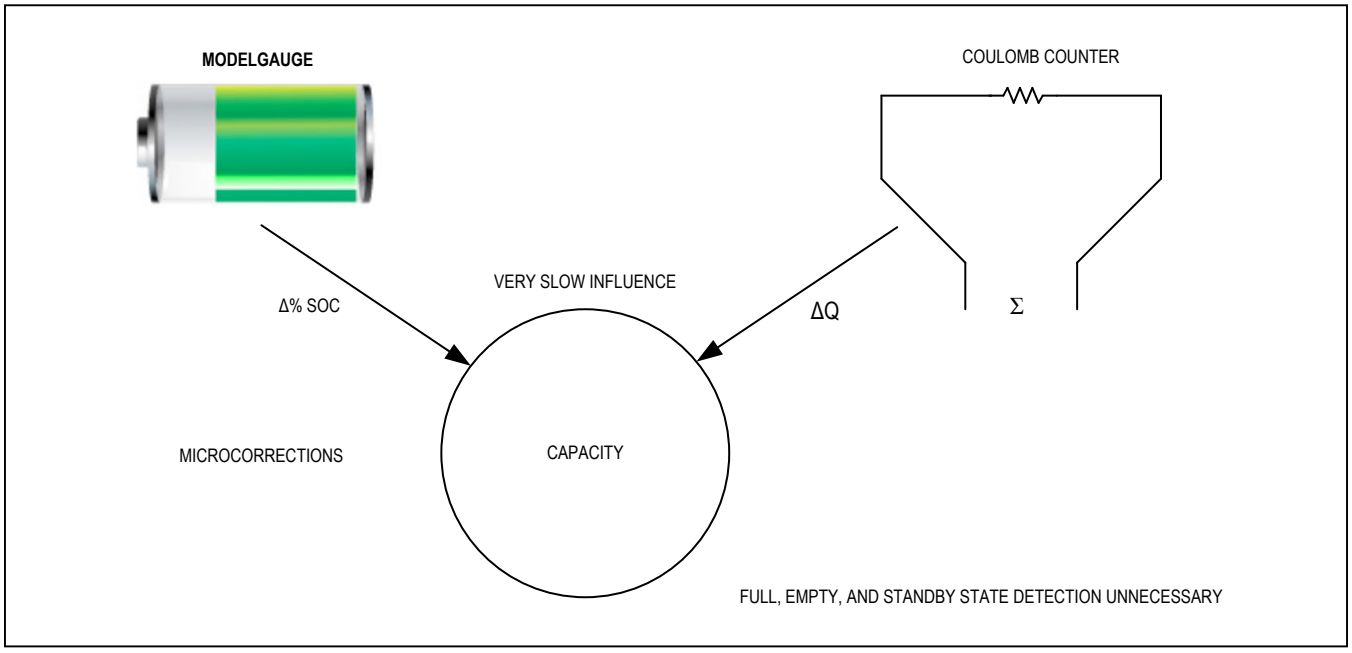



*Figure 4. Merger of Coulomb Counter and Voltage-Based Fuel Gauge*

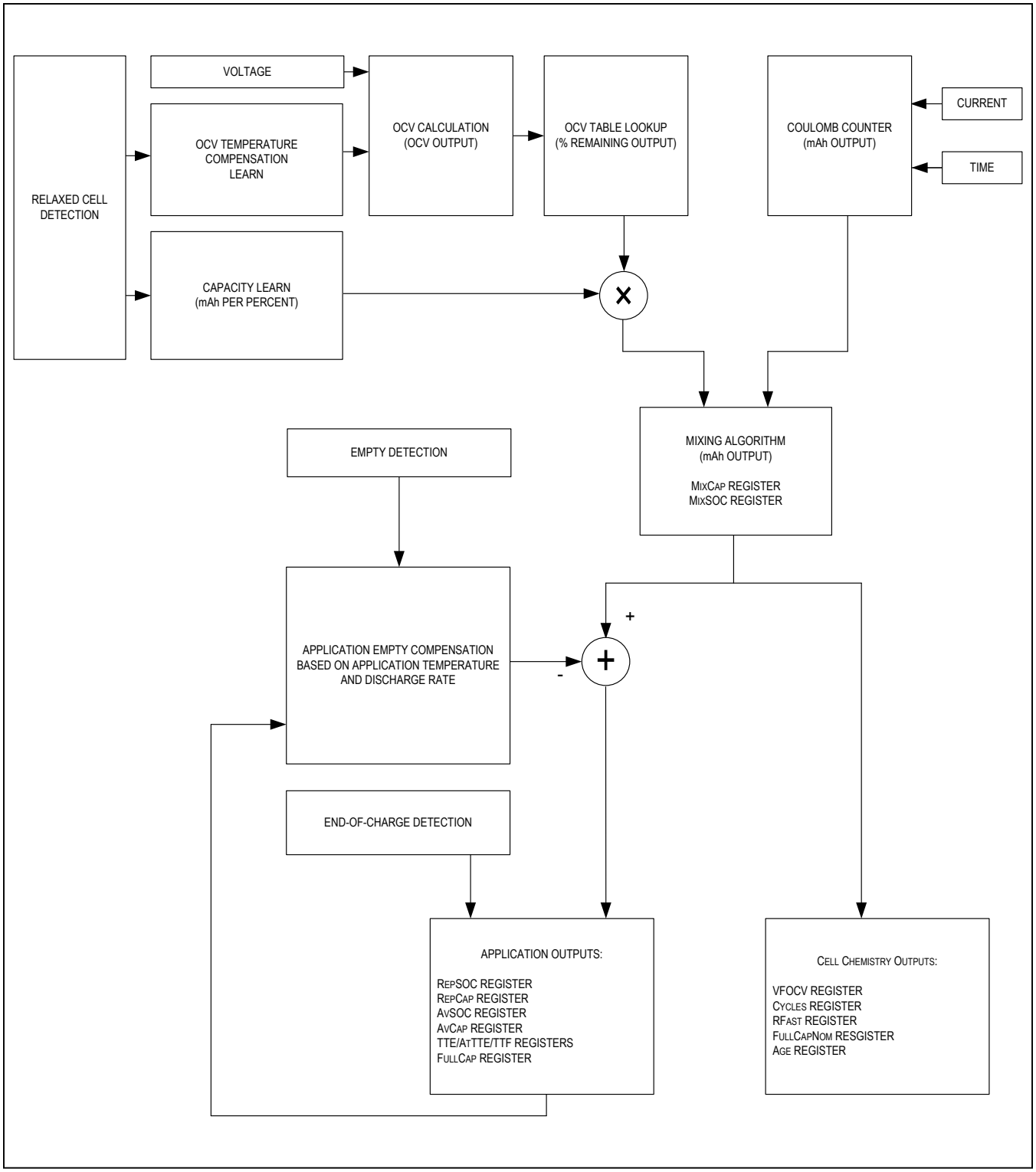



*Figure 5. ModelGauge m5 Block Diagram*

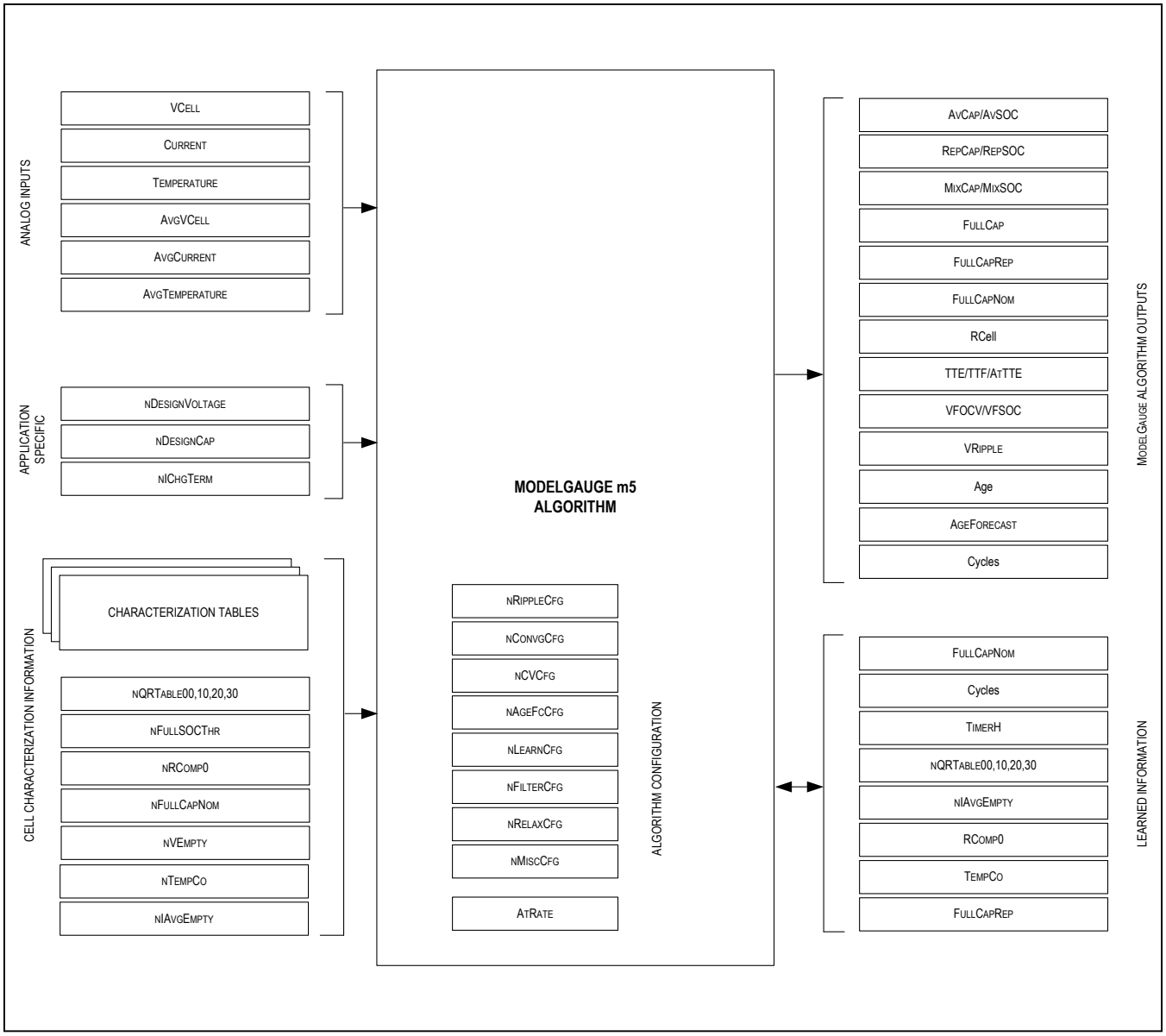



*Figure 6. ModelGauge m5 Registers*

### RepSOC Register (006h)

Register type: Percentage

Nonvolatile backup: None

RepSOC is a filtered version of the AvSOC register that prevents large jumps in the reported value caused by changes in the application such as abrupt changes in load current. RepSOC corresponds to RepCap and FullCapRep. RepSOC is intended to be the final state of charge percentage output for use by the application. See the *Empty Compensation* section for details.

### FullCapRep Register (035h)

Register Type: Capacity

Nonvolatile Backup and Restore From: nFullCapRep (1A9h) or nFullCapNom (1A5h). See nNVCfg2.EnFC.

This register reports the full capacity that goes with RepCap, generally used for reporting to the user. A new full-capacity value is calculated at the end of every charge cycle in the application.

### TTE Register (011h)

Register Type: Time

Nonvolatile Backup: None

The TTE register holds the estimated time to empty for the application under present temperature and load conditions. The TTE value is determined by dividing the AvCap register by the AvgCurrent register. The corresponding AvgCurrent filtering gives a delay in TTE empty, but provides more stable results.

### TTF Register (020h)

Register Type: Time

Nonvolatile Backup: None

The TTF register holds the estimated time to full for the application under present conditions. The TTF value is determined by learning the constant-current and constant-voltage portions of the charge cycle based on experience of prior charge cycles. Time to full is then estimated by comparing present charge current to the charge termination current. Operation of the TTF register assumes all charge profiles are consistent in the application. See Graph 26 in the *Typical Operating Characteristics* section for sample performance.

### Age Register (007h)

Register Type: Percentage

Nonvolatile Backup: None

The Age register contains a calculated percentage value of the application's present cell capacity compared to its expected capacity. The result can be used by the host to gauge the battery pack health as compared to a new pack of the same type. The equation for the register output is:

Age register = 100% x (FullCapNom register/DesignCap register)

### Cycles Register (017h)

Register Type: Special

Nonvolatile Backup and Restore: nCycles (1A4h)

The Cycles register maintains a total count of the number of charge/discharge cycles of the cell that have occurred. The result is stored as a percentage of a full cycle. For example, a full charge/discharge cycle results in the Cycles register incrementing by 100%. The Cycles register has a full range of 0 to 10485 cycles with a 16.0% LSb. Cycles is periodically saved to nCycles to provide a long term nonvolatile cycle count.

### TimerH Register (0BEh)

Register Type: Special

Nonvolatile Backup and Restore: nTimerH (1AFh) if nNVCfg2.enT is set

Alternate Initial Value: 0x0000

This register allows the ICs to track the age of the cell. An LSb of 3.2 hours gives a full-scale range for the register of up to 23.94 years. If enabled, this register is periodically backed up to nonvolatile memory as part of the learning function.

### FullCap Register (010h)

Register Type: Capacity

Nonvolatile Restore: Derived from nFullCapNom (1A5h)

This register holds the calculated full capacity of the cell based on all inputs from the ModelGauge m5 algorithm including empty compensation. A new full-capacity value is calculated continuously as application conditions change.

### nFullCapNom Register (1A5h)

Register Type: Capacity

Nonvolatile Backup and Restore: FullCapNom (023h)

This register holds the calculated full capacity of the cell, not including temperature and empty compensation. A new full-capacity nominal value is calculated each time a cell relaxation event is detected. This register is used to calculate other outputs of the ModelGauge m5 algorithm.

### RCell Register (014h)

Register Type: Resistance

Nonvolatile Backup: None

Initial Value: 0x0290

The RCell register displays the calculated internal resistance of the cell, or average internal resistance of each cell in the cell stack. RCell is determined by comparing open-circuit voltage (VFOCV) against measured voltage (VCell) over a long time period while under load or charge current.

### VRipple Register (0BCh)

Register Type: Special

Nonvolatile Backup: None

Initial Value: 0x0000

The VRipple register holds the slow average RMS value of VCell register reading variation compared to the AvgVCell register. The default filter time is 22.5s. See the *nRippleCfg Register (1B1h)* description. VRipple has an LSb weight of 1.25mV/128.

### nSOC Register (1AEh)

Register Type: Special

Nonvolatile Backup: VFSOC and MixSOC registers if nNVCfg2.enSOC = 1

This register has dual functionality depending on configuration settings. If nNVCfg2.enSOC = 1, this register provides nonvolatile backup of the MixSOC and VFSOC registers as shown in Figure 7.

Alternatively, if nNVCfg0.enAF = 1, this register stores a filtered version of the FullCapNom register value to be used with the Age Forecasting algorithm. Regardless of which option is enabled, this register is periodically saved to nonvolatile memory as part of the learning function. If neither option is enabled, this register can be used as general-purpose user memory.

### nVoltTemp Register (1AAh)

Register Type: Special

Nonvolatile Backup: AvgVCell and AvgTA registers if nNVCfg2.enVT = 1

This register has dual functionality depending on configuration settings. If nNVCfg2.enVT = 1, this register provides nonvolatile back up of the AvgVCell and AvgTA registers as shown in Figure 8.

Alternatively, if nNVCfg0.enAF = 1, this register stores an accumulated age slope value for use with the Age Forecasting algorithm. Regardless of which option is enabled, this register is periodically saved to nonvolatile memory as part of the learning function. If neither option is enabled, this register can be used as general purpose user memory.

### ModelGauge m5 EZ Performance

ModelGauge m5 EZ performance provides plug-and-play operation for the ICs. While the MAX1720x/MAX1721x can be custom tuned to the applications battery through a characterization process for ideal performance, the ICs have the ability to provide reasonable performance for most applications with no custom characterization required. Figure 9 and Figure 10 show the performance of the ModelGauge m5 algorithm in applications using only the default cell model information.

While EZ performance provides good performance for most cell types, some chemistries such as lithium-iron-phosphate (LiFePO4) and Panasonic NCR/NCA series cells require a custom model characterization for best performance.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MixSOC Upper Byte | | | | | | | | VFSOC Upper Byte | | | | | | | |

*Figure 7. nSOC (1AEh) Format When nNVCfg2.enSOC = 1*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AvgVCell Upper 9 Bits | | | | | | | | | AvgTA Upper 7 Bits | | | | | | |

*Figure 8. nVoltTemp (1AAh) Format When nNVCfg2.enVT = 1*

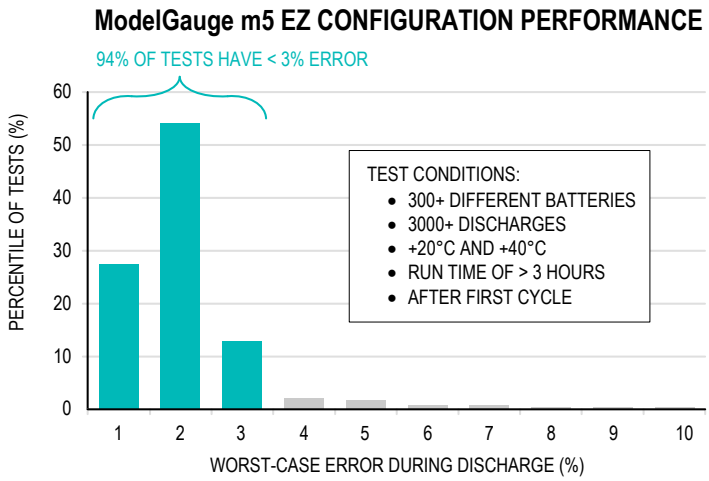



*Figure 9. EZ Configuration Performance Histogram*

| DESCRIPTION | AFTER FIRST CYCLE* (%) | AFTER SECOND CYCLE* (%) |
|---|---|---|
| Tests with error less than 3% | 94 | 95 |
| Tests with error less than 5% | 97 | 98 |
| Tests with error less than 10% | 99.5 | 99.7 |

*TEST CONDITIONS: +20ºC AND +40ºC, RUN TIME OF > 3 HOURS.

*Figure 10. EZ Configuration Performance vs. Test Conditions*

### OCV Estimation and Coulomb Count Mixing

The core of the ModelGauge m5 algorithm is a mixing algorithm that combines the OCV state estimation with the coulomb counter. After a power-on reset of the IC, coulomb-count accuracy is unknown. The OCV state estimation is weighted heavily compared to the coulomb count output. As the cell progresses through cycles in the application, coulomb-counter accuracy improves and the mixing algorithm alters the weighting so that the coulomb-counter result is dominant. From this point forward, the IC switches to servo mixing. Servo mixing provides a fixed magnitude continuous error correction to the coulomb count, up or down, based on the direction of error from the OCV estimation. This allows differences between the coulomb count and OCV estimation to be corrected quickly. See Figure 11.

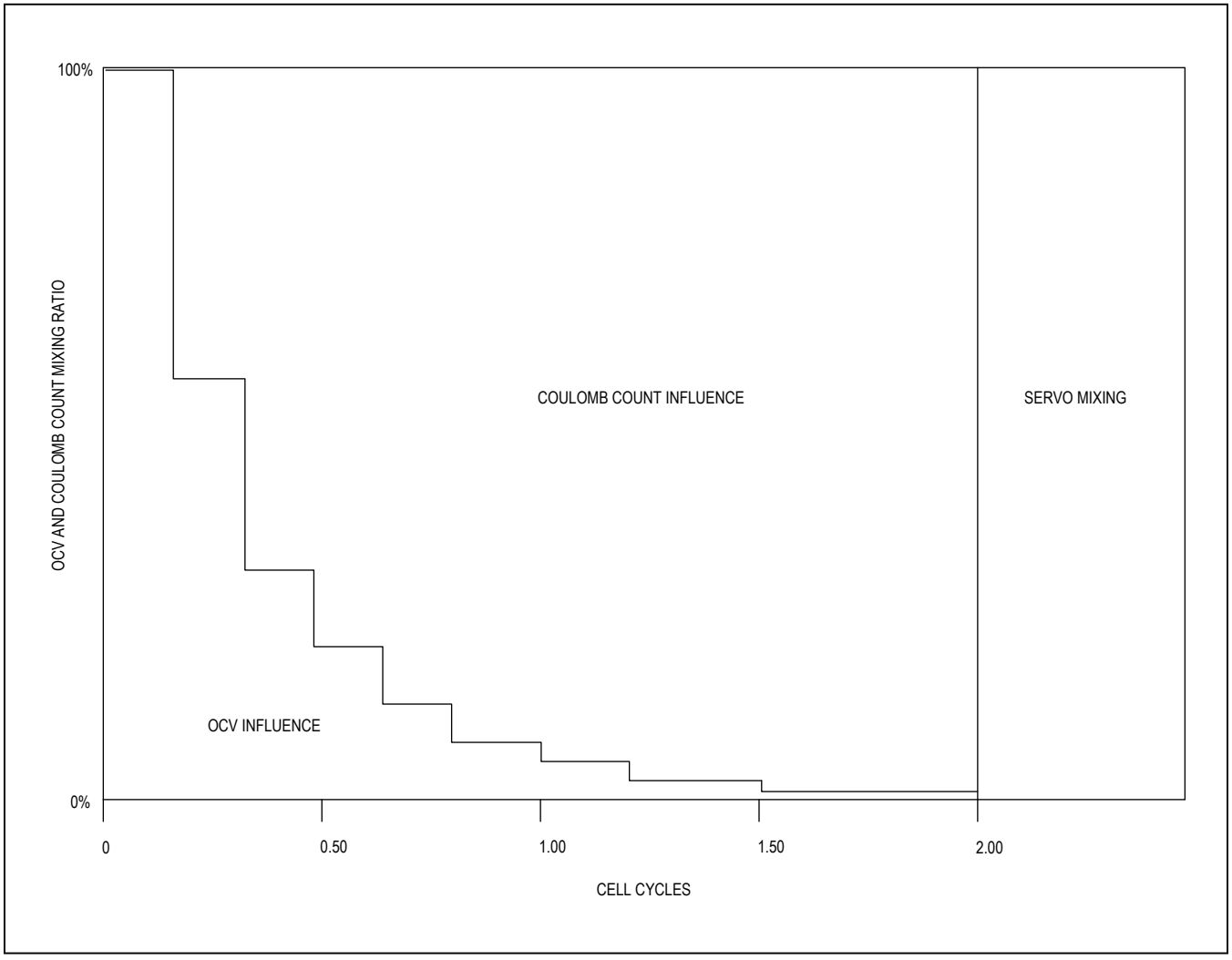



*Figure 11. Voltage and Coulomb Count Mixing*

The resulting output from the mixing algorithm does not suffer accumulation drift from current measurement offset error and is more stable than a stand-alone OCV estimation algorithm. See Figure 12. Initial accuracy depends on the relaxation state of the cell. The highest initial accuracy is achieved with a fully relaxed cell.

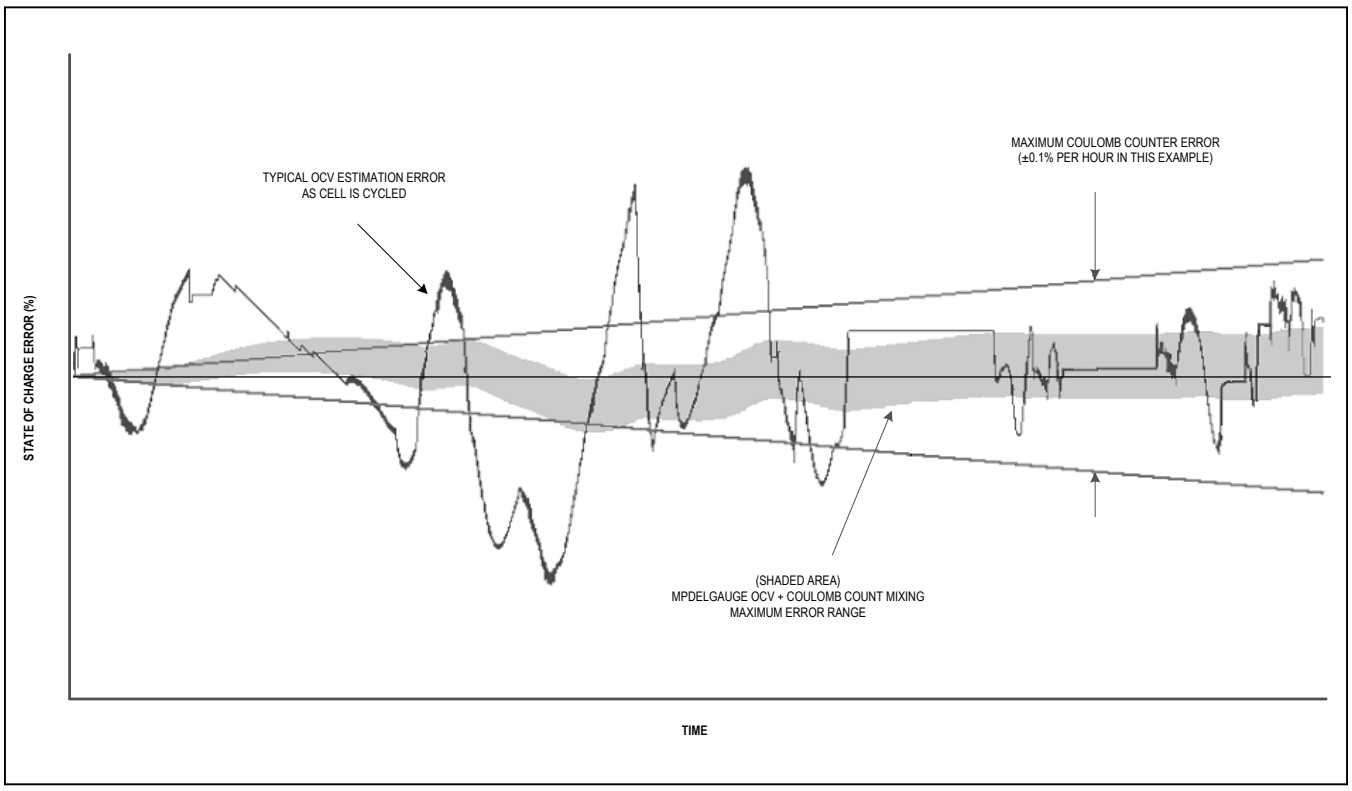



*Figure 12. ModelGauge m5 Typical Accuracy Example*

### Empty Compensation

As the temperature and discharge rate of an application changes, the amount of charge available to the application also changes. The ModelGauge m5 algorithm distinguishes between remaining capacity of the cell and remaining capacity of the application and reports both results to the user.

The MixCap output register tracks the charge state of the cell. This is the theoretical mAh of charge that can be removed from the cell under ideal conditions—extremely low discharge current and no concern for cell voltage. This result is not affected by application conditions such as cell impedance or minimum operating voltage of the application. ModelGauge m5 continually tracks the expected empty point of the application in mAh. This is the amount of charge that cannot be removed from the cell by the application because of minimum voltage requirements and internal losses of the cell. The ICs subtracts the amount of charge not available to the application from the MixCap register and reports the result in the AvCap register.

Since available remaining capacity is highly dependent on discharge rate, the AvCap register can be subject to large instantaneous changes as the application load current changes. The result can increase, even while discharging, if the load current suddenly drops. This result, although correct, can be very counter-intuitive to the host software or end user. The RepCap output register contains a filtered version of AvCap that removes any abrupt changes in remaining capacity. RepCap converges with AvCap over time to correctly predict the application empty point while discharging or the application full point while charging. Figure 13 shows the relationship of these registers.

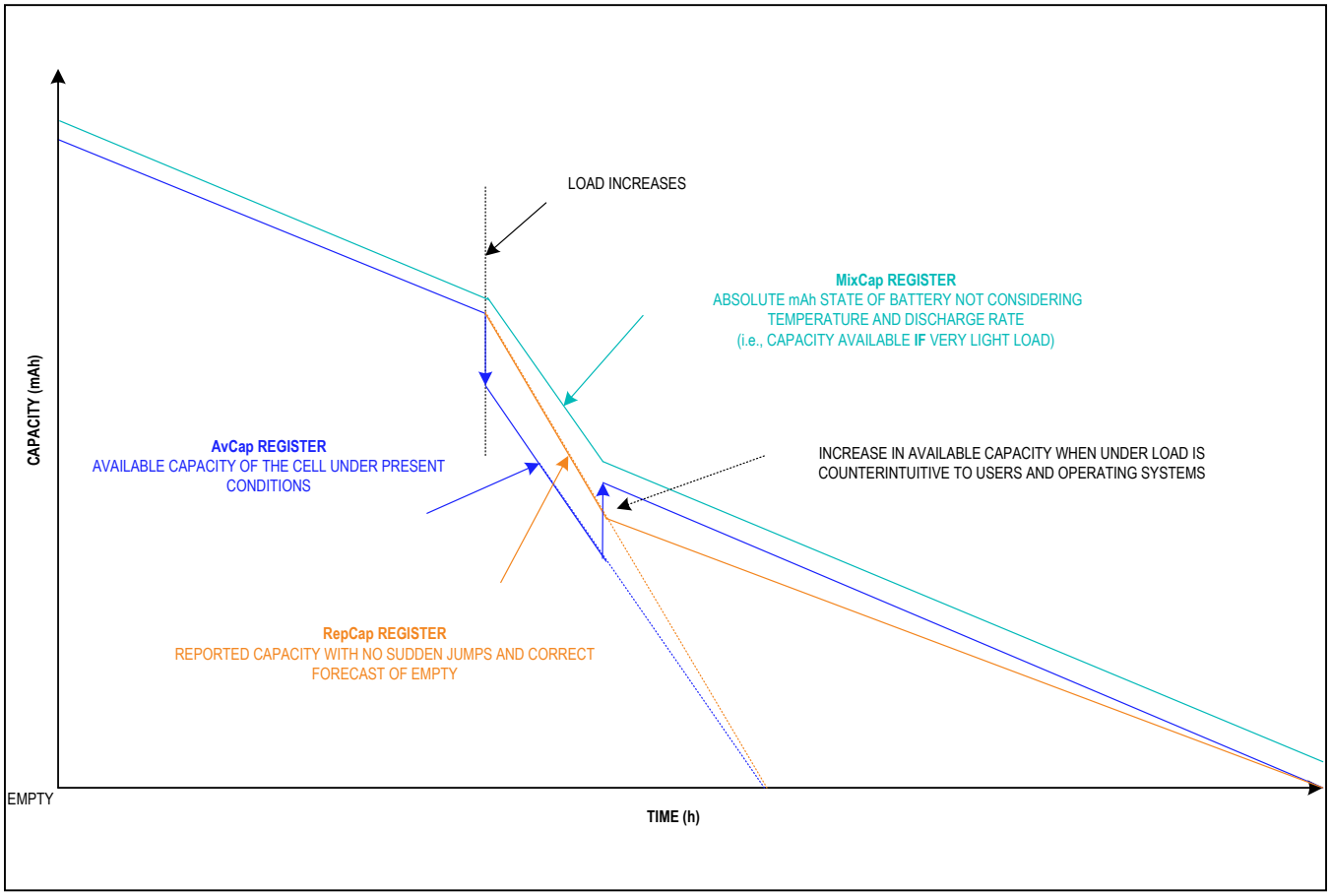



*Figure 13. Handling Changes in Empty Calculation*

### End-of-Charge Detection

The ICs detect the end of a charge cycle when the application current falls into the band set by the IChgTerm register value while the VFSOC value is above the FullSOCThr register value. By monitoring both the Current and AvgCurrent registers, the device can reject false end-of-charge events such as application load spikes or early charge-source removal. See Figure 14. When a proper end-of-charge event is detected, the device learns a new FullCapRep register value based on the RepCap register output. If the old FullCapRep value is too high, it is adjusted on a downward slope near the end of charge as defined by the MiscCfg.FUS setting until it reaches RepCap. If the old FullCapRep is too low, it is adjusted upward to match RepCap. This prevents the calculated state of charge from ever reporting a value greater than 100%. See Figure 15.

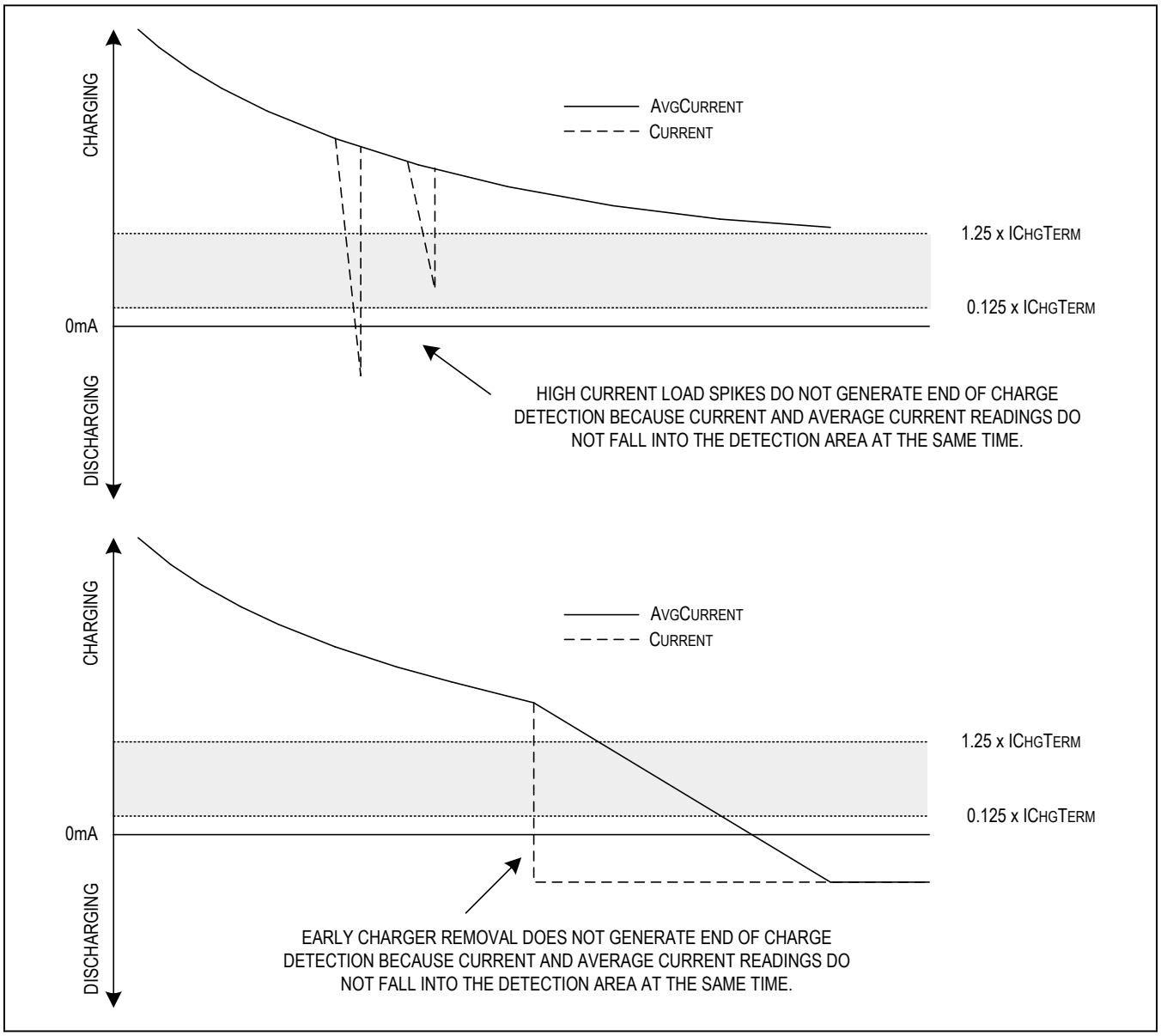



*Figure 14. Blocking False End of Charge Events*

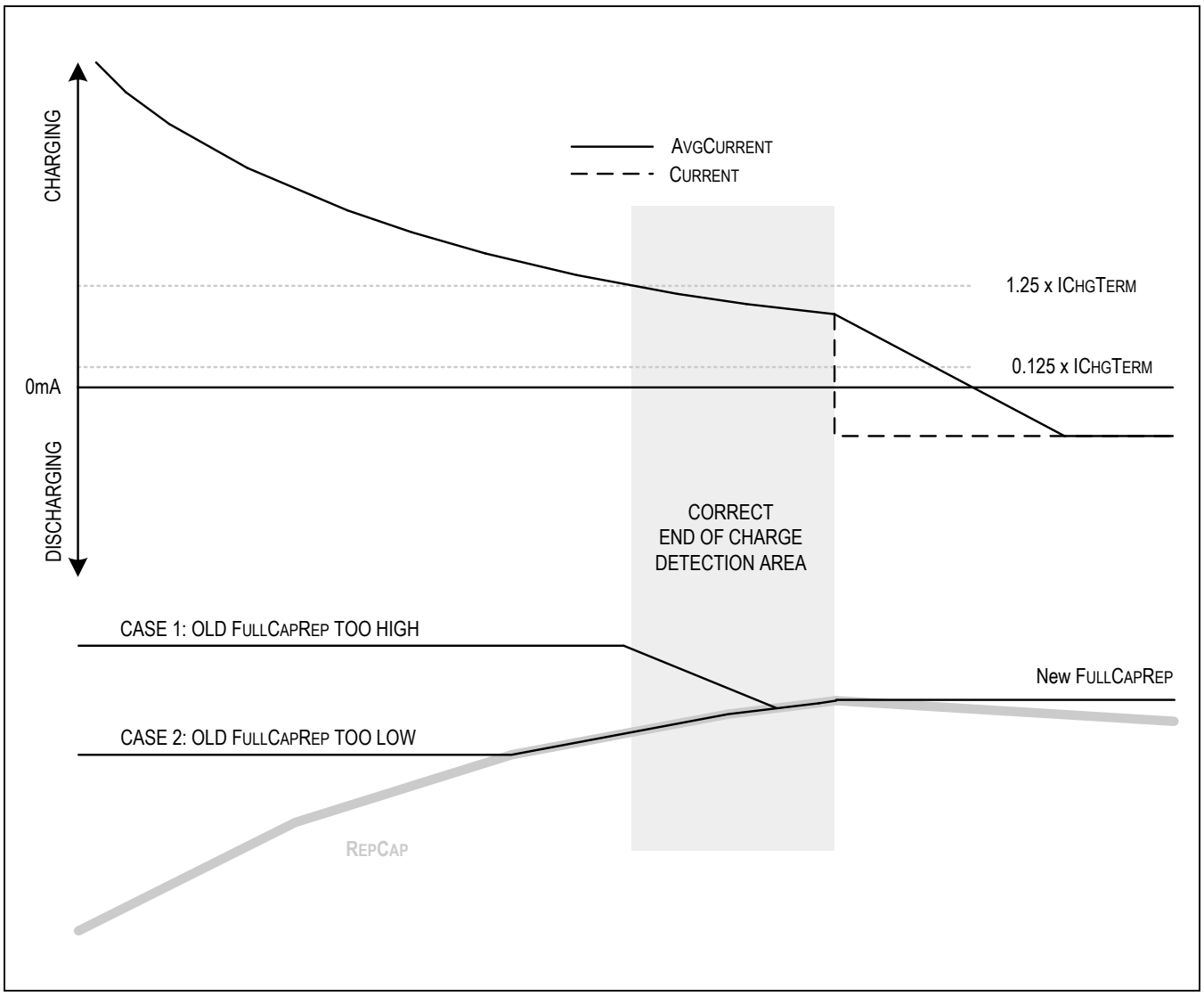



*Figure 15. FullCapRep Learning at End of Charge*

Charge termination is detected by the IC when the following conditions are met:

• VFSOC register > FullSOCThr register
• AND IChgTerm x 0.125 < Current register < IChgTerm x 1.25
• AND IChgTerm x 0.125 < AvgCurrent register < IChgTerm x 1.25

### Fuel Gauge Learning

The ICs periodically make internal adjustments to cell characterization and application information to remove initial error and maintain accuracy as the cell ages. These adjustments always occur as small corrections to prevent instability of the system and prevent any noticeable jumps in the fuel-gauge outputs. Learning occurs automatically without any input from the host. In addition to estimating the battery's state of charge, the ICs observe the battery's relaxation response and adjusts the dynamics of the voltage fuel gauge. Registers used by the algorithm include:

• **Application Capacity (FullCapRep register).** This is the total capacity available to the application at full, set through the IChgTerm and FullSOCThr registers as described in the *End-of-Charge Detection* section. See the *FullCap Register (010h)* description.

• **Cell Capacity (FullCapNom register).** This is the total cell capacity at full, according to the voltage fuel gauge. This includes some capacity that is not available to the application at high loads and/or low temperature. The ICs periodically compare percent change based on an open circuit voltage measurement vs. coulomb-count change as the cell charges and discharges, maintaining an accurate estimation of the pack capacity in mAh as the pack ages. See Figure 16.

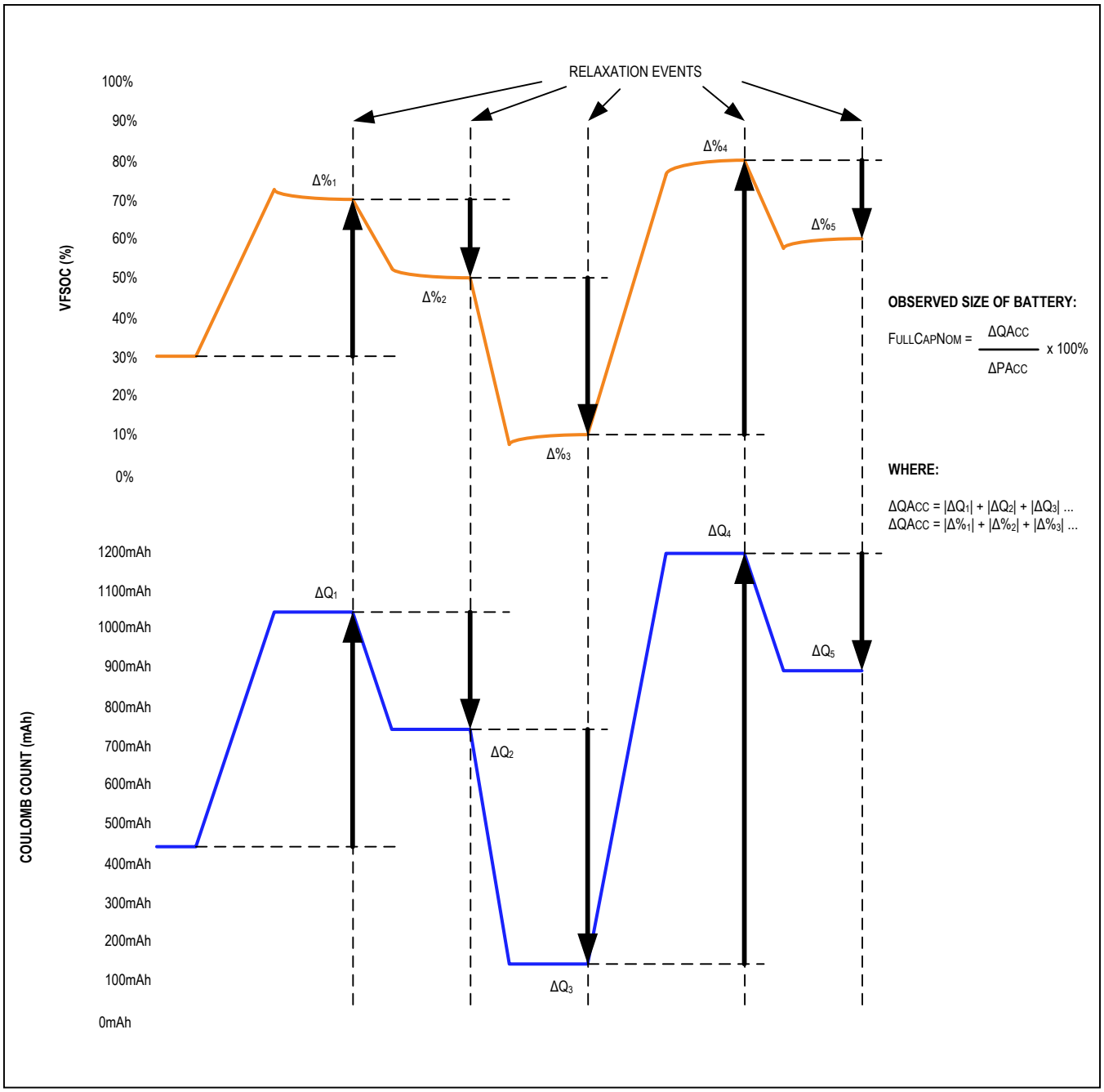



*Figure 16. FullCapNom Learning*

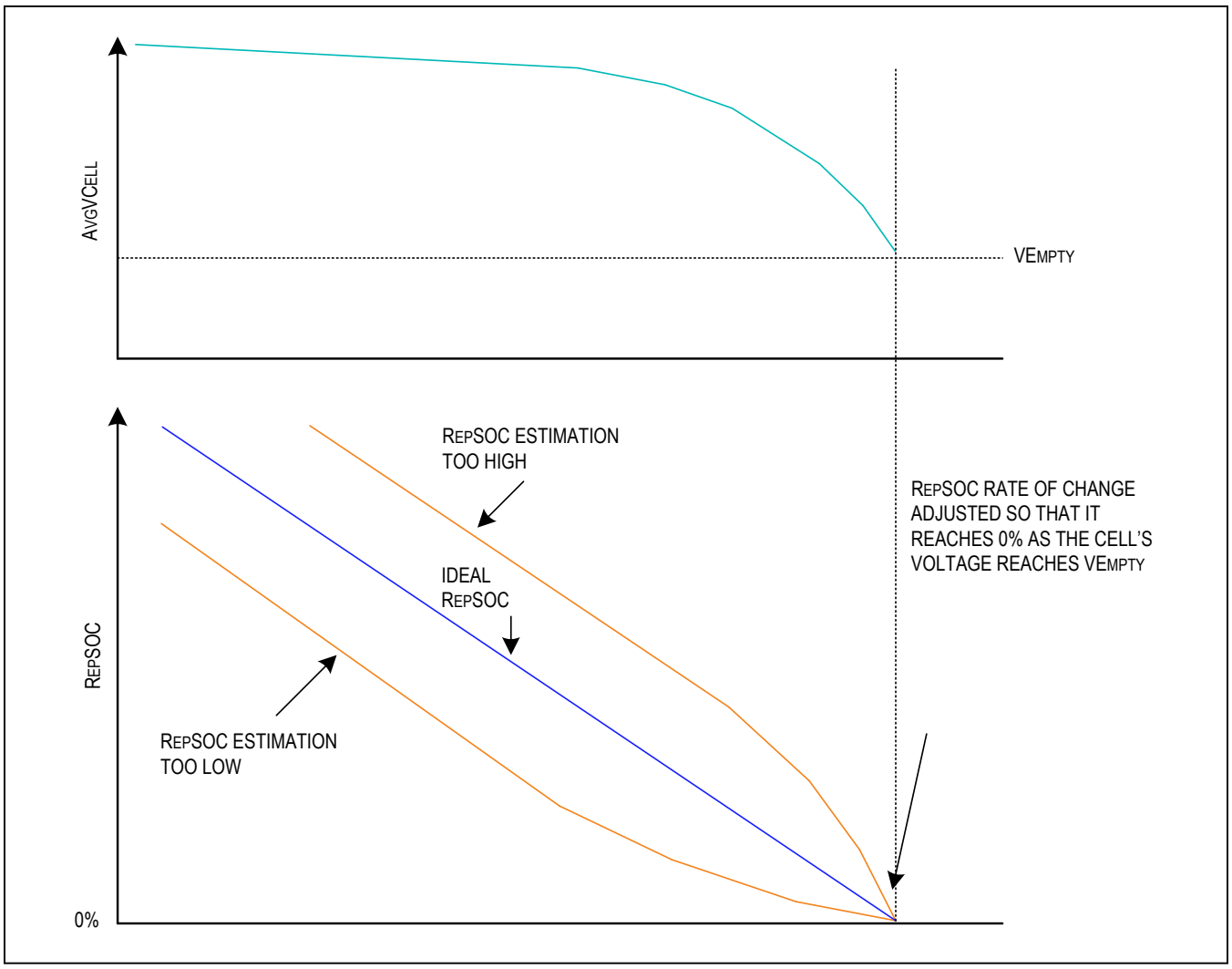



*Figure 17. Converge to Empty*

• **Voltage Fuel-Gauge Adaptation.** The ICs observe the battery's relaxation response and adjust the dynamics of the voltage fuel gauge. This adaptation adjusts the RComp0 register during qualified cell relaxation events.

• **Empty Compensation.** The ICs update internal data whenever an empty cell is detected (VCell < VEmpty) to account for cell age or other cell deviations from the characterization information.

### Converge-to-Empty

The MAX1720x/MAX1721x include a feature that guarantees the fuel gauge output converges to 0% as the cell voltage approaches the empty voltage. As the cell's voltage approaches the expected empty voltage (AvgVCell approaches VEmpty) the ICs smoothly adjust the rate of change of RepSOC so that the fuel gauge reports 0% at the exact time that the cell's voltage reaches empty. This prevents unexpected shutdown or an early 0% SOC reported by the fuel gauge. See Figure 17. In addition, the fuel gauge limits RepSOC to not go below 1% until AvgVCell crosses VEmpty.

### Determining Fuel Gauge Accuracy

To determine the true accuracy of a fuel gauge, as experienced by end users, the battery should be exercised in a dynamic manner. The end-user accuracy cannot be understood with only simple cycles. To challenge a correction-based fuel gauge, such as a coulomb counter, test the battery with partial loading sessions. For example, a typical user can operate the device for 10 minutes and then stop use for an hour or more. A robust test method includes these kinds of sessions many times at various loads, temperatures, and duration. Refer to the Application Note 4799: *Cell Characterization Procedure for a ModelGauge m3/ModelGauge m5 Fuel Gauge.*

### Initial Accuracy

The ICs use the first voltage reading after power-up or after cell insertion to determine the starting output of the fuel gauge. It is assumed that the cell is fully relaxed prior to this reading; however, this is not always the case. If there is a load or charge current at this time, the initial reading is compensated using the characterized internal impedance of the cell (RFast register) to estimate the cell's relaxed voltage. If the cell was recently charged or discharged, the voltage measured by the IC may not represent the true state of charge of the cell, resulting in initial error in the fuel gauge outputs. In most cases, this error is minor and is quickly removed by the fuel gauge algorithm during the first hour of normal operation.

### Cycle+ Age Forecasting

A special feature of the ModelGauge m5 algorithm is the ability to forecast the number of cycles a user can get out of the cell during its lifetime. This allows an application to adjust a cell's charge profile over time to meeting the cycle life requirements of the cell. See Figure 19. The algorithm monitors the change in cell capacity over time and calculates the number of cycles it takes for the cell's capacity to drop to a predefined threshold of 85% of original. Remaining cycles below 85% of the original capacity are unpredictable and not managed by age forecasting.

### nAgeFcCfg Register (1D2h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register. The recommended default value is 0xD5E3.

The nAgeFcCfg register is used to configure age forecasting functionality. Register data is nonvolatile and is typically configured only once during pack assembly. Figure 18 shows the register format.

**DeadTargetRatio:** Sets the remaining percentage of initial cell capacity where the cell is considered fully aged. DeadTargetRatio can be adjusted between 75% and 86.72% with an LSb of 0.7813%. For example, if age forecasting was configured to estimate the number of cycles until the cell's capacity dropped to 85.1574% of when it was new, DeadTargetRatio should be programmed to 1101b.

**CycleStart:** Sets the number of cell cycles before age forecasting calculations begin. CycleStart has a range of 0.00 to 81.92 cycles with an LSb of 0.64 cycles. Since age forecasting estimation becomes more accurate over time, most applications use a default value of 30 cycles.

**0:** Always write this location 0.

**1:** Always write this location 1.

### AgeForecast Register (0B9h)

Register Type: Special

Nonvolatile Backup: None

The AgeForecast register displays the estimated total cycle life of the application cell. The AgeForecast value should be compared against the Cycles (017h) register to determine the estimated number of remaining cell cycles. This is accomplished by accumulating the capacity loss per cycle as the cell ages. The result will become more accurate with each cycle measured. The AgeForecast register has a full range of 0 cycles to 10485 cycles with a 0.16 cycle LSb. This register is recalculated from learned information at power-up.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DeadTargetRatio | | | | CycleStart | | | | | | | 0 | 0 | 0 | 1 | 1 |

*Figure 18. nAgeFcCfg Register (1D2h) Format*

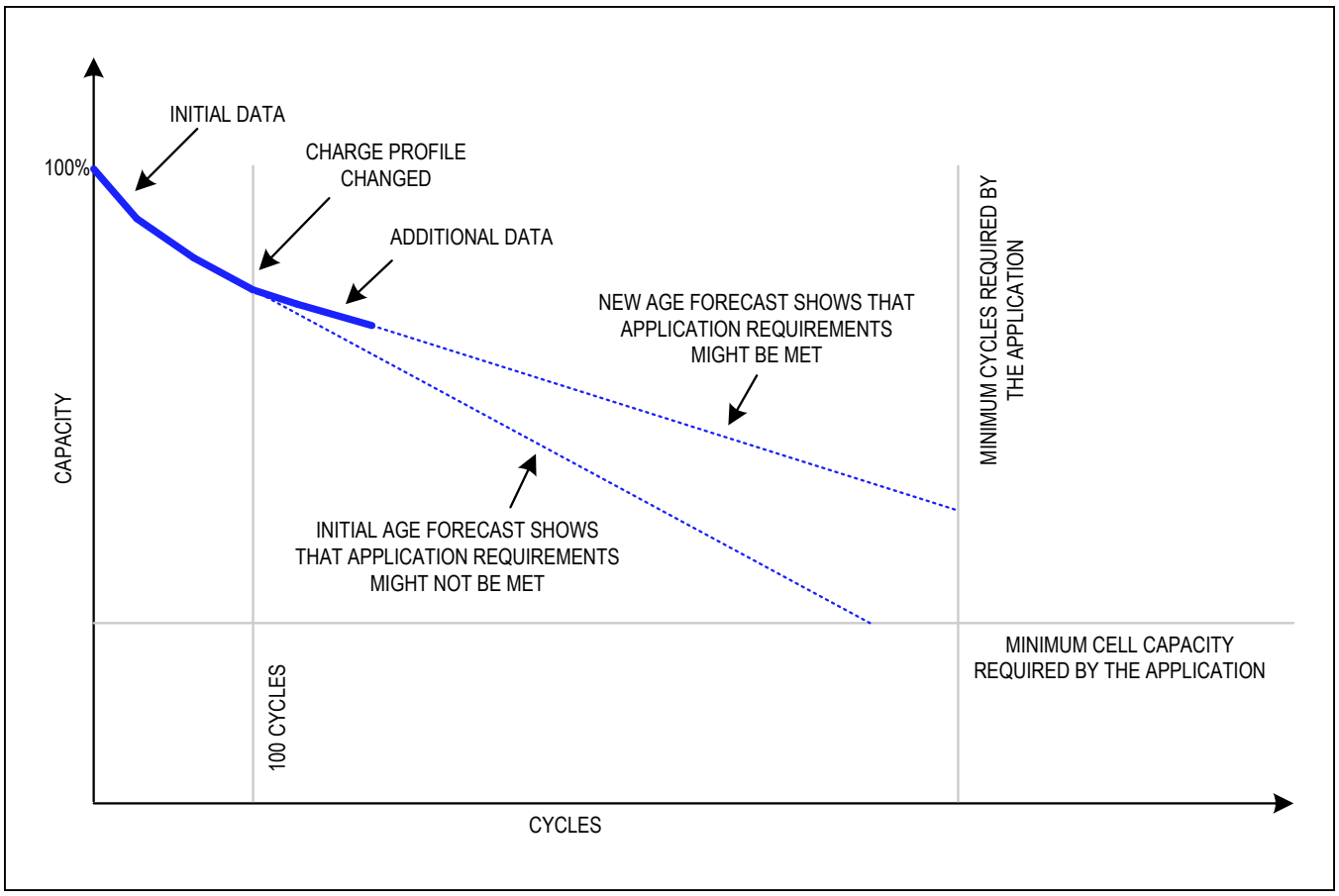



*Figure 19. Benefits of Age Forecasting*

## Table 2. Minimum and Maximum Cell Sizes for Age Forecasting

| SENSE RESISTOR (Ω) (nCGain=0x4000) | MINIMUM CELL SIZE FOR FORECASTING (mAh) | MAXIMUM CELL SIZE FOR FORECASTING (mAh) |
|---|---|---|
| 0.005 | 1600 | 5000 |
| 0.010 | 800 | 2500 |
| 0.020 | 400 | 1250 |

### Age Forecasting Requirements

There are several requirements for proper operation of the age forecasting feature as follows:

- There is a minimum and maximum cell size that the age forecasting algorithm can handle. Table 2 shows the allowable range of cell sizes that can be accurately age forecasted depending on the size of the sense resistor used in the application. Note this range is different from the current and capacity measurement range for a given sense resistor. See the *Current Measurement* section for details.
- Age forecasting requires a minimum of 100 cycles before achieving reasonable predictions. Ignore the age forecasting output until then.
- Age forecasting requires a custom characterized battery model to be used by the IC. Age forecasting is not valid when using the ModelGauge m5 EZ model.

### Enabling Age Forecasting

The following steps are required to enable the age forecasting feature:

1) 1) Set nNVCfg2.enVT = 0 and nNVCfg2.enSOC = 0. These functions conflict with age forecasting and must be disabled.
2) 2) Set nSOC to the value of nFullCapNom.
3) 3) Set nVoltTemp to 0x0001.
4) 4) Set nNVCfg0.enAF = 1 to begin operation.

### Battery Life Logging

The MAX1720x/MAX1721x has the ability to log learned battery information providing the host with a history of conditions experienced by the cell pack over its life time. The IC can store up to 203 snapshots of page 1Ah in nonvolatile memory. Individual registers from page 1Ah are summarized in Table 3. Their nonvolatile backup must be enabled in order for logging to occur. See each register's detailed description in other sections of this data sheet. The logging rate default is once every 10.5 equivalent cell cycles but can be adjusted from 0.5 cycles up to 64.5 cycles using the nNVCfg2.CYCLESpSAVE setting.

### Life-Logging Data Example

Figure 20 shows a graphical representation of sample history data read from an IC. Analysis of this data can provide information of cell performance over its lifetime as well as detect any application anomalies that might have affected performance.

## Table 3. Life Logging Register Summary

| REGISTER ADDRESS | REGISTER NAME | FUNCTION |
|---|---|---|
| 1A0h | nQRTable00 | Learned characterization information used to determine when the cell pack is empty under application conditions. |
| 1A1h | nQRTable10 | |
| 1A2h | nQRTable20 | |
| 1A3h | nQRTable30 | |
| 1A4h | nCycles | Total number of equivalent full cycles seen by the cell since assembly. |
| 1A5h | nFullCapNom | Calculated capacity of the cell independent of application conditions. |
| 1A6h | nRComp0 | Learned characterization information related to the voltage fuel gauge. |
| 1A7h | nTempCo | |
| 1A8h | nIAvgEmpty | Typical current seen in by the application a the point where the cell reaches empty. |
| 1A9h | nFullCapRep | Calculated capacity of the cell under present application conditions. |
| 1AAh | nVoltTemp | The average voltage and temperature seen by the IC at the instance of learned data backup. If age forecasting is enabled, this register contains different information. |
| 1ABh | nMaxMinCurr | Maximum and minimum current, voltage, and temperature seen by the IC during this logging window. |
| 1ACh | nMaxMinVolt | |
| 1ADh | nMaxMinTemp | |
| 1AEh | nSOC | Calculated present state of charge of the battery pack at the instance of learned data backup. If age forecasting is enabled, this register contains different information. |
| 1AFh | nTimerH | Total elapsed time since cell pack assembly not including time spent in shutdown mode. |

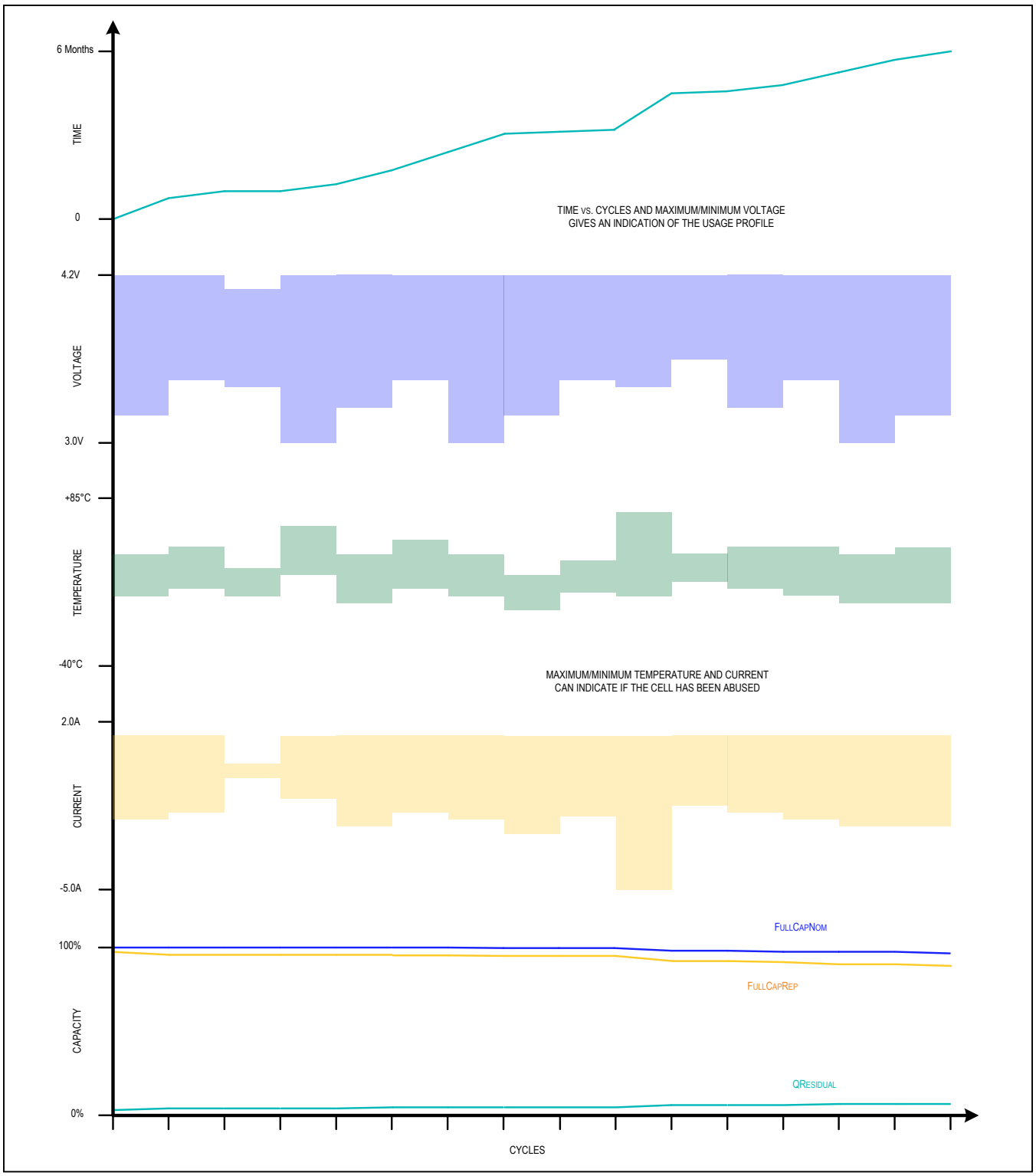



*Figure 20. Sample Life-Logging Data*

### Determining Number of Valid Logging Entries

While logging data, the ICs begin on history page 1 and continue until all history memory has been used at page 203. Prior to reading history information out of the ICs, the host must determine which history pages has been written and which, if any, had write errors and should be ignored. Each page of history information has two associated write flags that indicate if the page has been written and two associated valid flags that indicate if the write was successful. The HISTORY RECALL command [0xE2XX] is used to load the history flags into page 0Eh of IC memory where the host can then read their state. Table 6 shows which command and which page the 0Eh address has the flag information for a given history page. For example ,to see the write flag information of history pages 1-8, send the 0xE2FB command then read address 0E1h. To see the valid flag information of pages 1–8, send the 0xE2FC command and then read address 0EBh.

Once the write flag and valid flag information is read from the IC, it must be decoded. Each register holds two flags for a given history page. Figure 21 shows the register format. The flags for a given history page are always spaced 8 bits apart from one another. For example, history page 1 flags are always located at bit positions D0 and D8, history page 84 flags are at locations D3 and D11, etc. Note that the last flag register contains information for only 3 pages, in this case the upper 5 bits of each byte should be ignored.

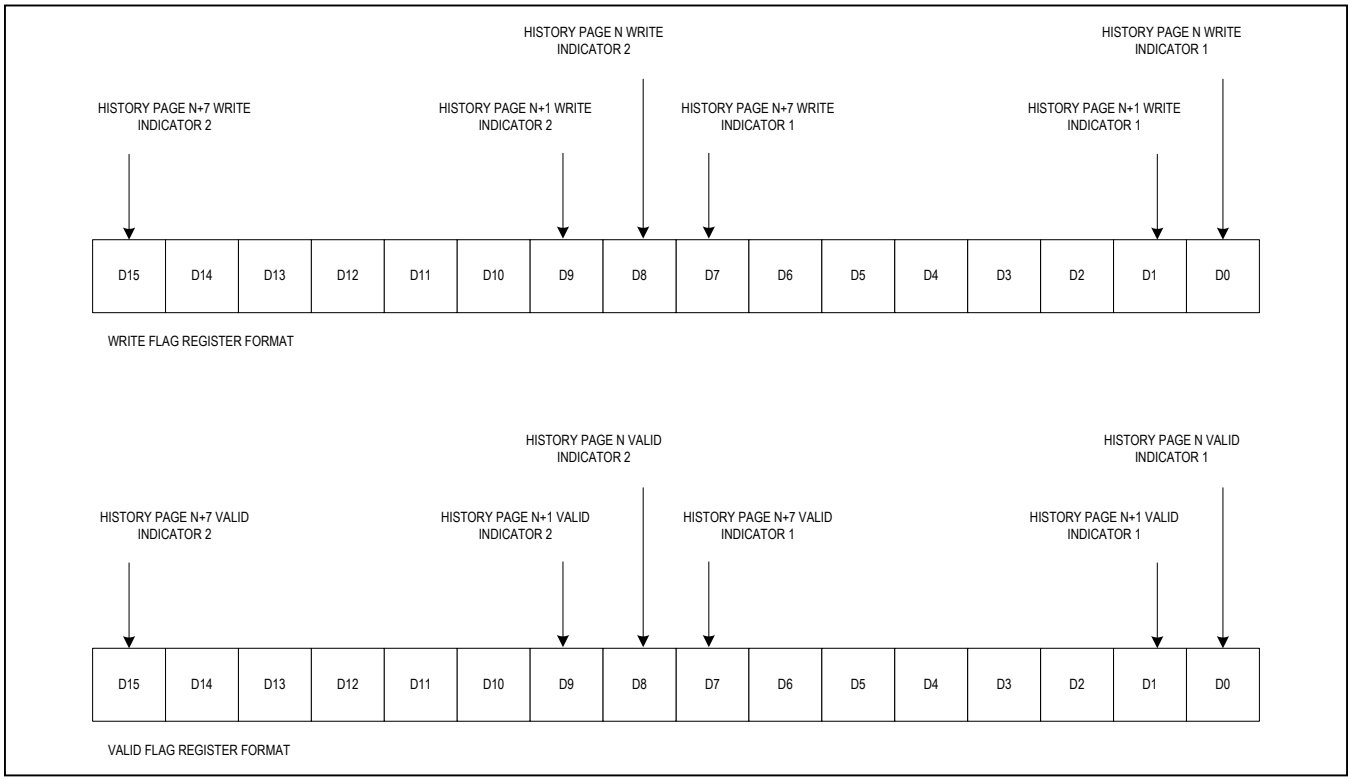



*Figure 21. Write Flag Register and Valid Flag Register Formats*

Once all four flags for a given history page are known, the host can determine if the history page contains valid data. If either write flag is set then data has been written to that page by the IC. If both write flags are clear, the page has not yet been written. Due to application conditions, the write may not have been successful. Next check the valid flags. If either valid flag is set, the data should be considered good. If both valid flags are clear then the data should be considered bad and the host should ignore it. Table 4 shows how to decode the flags.

### Reading History Data

Once all pages of valid history data have been identified, they can be read from the ICs using the HISTORY RECALL command. Table 5 shows the command and history page relationship. After sending the command, wait $t_{RECALL}$, then read the history data from the IC's page 0Eh. Each page of history data has the same format as page 1Ah. For example, nCycles is found at address 1A4h and nCycles history are at 0E4h, nTimerH is located at address 1AFh, and nTimerH history is located at address 0EFh, etc.

### History Data Reading Example

The host reads the life-logging data from a given IC. The host must first determine how many history pages have been written and if there are any errors. To start checking history page 1, the host sends 0xE2FB to the command register, waits $t_{RECALL}$, then reads location 0E1h. If either the D0 or the D8 bit in the read data word is a logic 1, the host knows that history page 1 contains history data.

## Table 4. Decoding History Page Flags

| WRITE INDICATOR 1 | WRITE INDICATOR 2 | VALID INDICATOR 1 | VALID INDICATOR 2 | PAGE STATUS |
|---|---|---|---|---|
| 0 | 0 | X | X | Page Empty. |
| 1 | X | 0 | 0 | Write Failure. Page has Bad Data. |
| | | 1 | X | Write Success. Page has Good Data. |
| | | X | 1 | |
| X | 1 | 0 | 0 | Write Failure. Page has Bad Data. |
| | | 1 | X | Write Success. Page has Good Data. |
| | | X | 1 | |

## Table 5. Reading History Data

| COMMAND | HISTORY PAGE RECALLED TO PAGE 0EH |
|---|---|
| 0xE226 | Page 1 |
| 0xE227 | Page 2 |
| ... | ... |
| 0xE2F0 | Page 203 |

The host can then check page 2 (bits D1 and D9) up to page 7 (bits D7 and D15). The host continues on to pages 8 to 16 by reading location 0E2h, and then repeating individual bit testing. This process is repeated for each command and address listed in Table 6 until the host finds a history page where both write flags read logic 0. This is the first unwritten page. All previous pages contain data, all following pages are empty.

The host must now determine which, if any, of the history pages have bad data and must be ignored. The above process is repeated for every location looking at the valid flags instead of the write flags. Any history page where both valid flags read logic 0 is considered bad due to a write failure and that page should be ignored. Once the host has a complete list of valid written history pages, commands 0xE226 to 0xE2F0 can be used to read the history information from page 0Eh for processing.

**Note:** This example was simplified in order to describe the procedure. A more efficient method would be for the host to send a history command once and then read all associated registers. For example, the host could send the 0xE2FC command once and then read the entire memory space of 0E0h to 0EFh that would contain all write flags for pages 121 to 203 (0E0h to 0EAh) and all valid flags for pages 1 to 40 (0EBh to 0EFh). This applies for all 0xE2XX history commands.

See *Appendix A: Reading History Data Psuedo-Code Example* for a psuedo-code example of reading history data.

## Table 6. Reading History Page Flags

| ASSOCIATED HISTORY PAGES | COMMAND TO RECALL WRITE FLAGS | WRITE FLAG ADDRESS | COMMAND TO RECALL VALID FLAGS | VALID FLAG ADDRESS |
|---|---|---|---|---|
| 1–8 | 0xE2FB | 0E1h | 0xE2FC | 0EBh |
| 9–16 | | 0E2h | | 0ECh |
| 17–24 | | 0E3h | | 0EDh |
| 25–32 | | 0E4h | | 0EEh |
| 33–40 | | 0E5h | | 0EFh |
| 41–48 | | 0E6h | 0xE2FD | 0E0h |
| 49–56 | | 0E7h | | 0E1h |
| 57–64 | | 0E8h | | 0E2h |
| 65–72 | | 0E9h | | 0E3h |
| 73–80 | | 0EAh | | 0E4h |
| 81–88 | | 0EBh | | 0E5h |
| 89–96 | | 0ECh | | 0E6h |
| 97–104 | | 0EDh | | 0E7h |
| 105–112 | | 0EEh | | 0E8h |
| 113–120 | | 0EFh | | 0E9h |
| 121–128 | 0xE2FC | 0E0h | | 0EAh |
| 129–136 | | 0E1h | | 0EBh |
| 137–144 | | 0E2h | | 0ECh |
| 145–152 | | 0E3h | | 0EDh |
| 153–160 | | 0E4h | | 0EEh |
| 161–168 | | 0E5h | | 0EFh |
| 169–176 | | 0E6h | 0xE2FE | 0E0h |
| 177–184 | | 0E7h | | 0E1h |
| 185–192 | | 0E8h | | 0E2h |
| 193–200 | | 0E9h | | 0E3h |
| 201–203 | | 0EAh | | 0E4h |

### ModelGauge m5 Algorithm Model Registers

The following registers are inputs to the ModelGauge algorithm and store characterization information for the application cells as well as important application specific specifications. They are described only briefly here. Contact Maxim for information regarding cell characterization.

### nXTable0 (180h) to nXTable11 (18Bh) Registers

Register Type: Special

Nonvolatile Restore: There are no associated restore locations for these registers.

Cell characterization information used by the ModelGauge algorithm to determine capacity versus operating conditions. This table comes from battery characterization data. These are nonvolatile memory locations. These registers are free to user memory if using ModelGauge m5 EZ.

### nOCVTable0 (190h) to nOCVTable11 (19Bh) Registers

Register Type: Special

Nonvolatile Restore: There are no associated restore locations for these registers.

Cell characterization information used by the ModelGauge algorithm to determine capacity versus operating conditions. This table comes from battery characterization data. These are nonvolatile memory locations. These registers are free to user memory if using ModelGauge m5 EZ.

### nQRTable00 (1A0h) to nQRTable30 (1A3h) Registers

Register Type: Special

Nonvolatile Backup and Restore: QRTable00 to QRTable30 (012h, 022h, 032h, 042h)

The nQRTable00 to nQRTable30 register locations contain characterization information regarding cell capacity that is not available under certain application conditions.

### nFullSOCThr Register (1C6h)

Register Type: Percentage

Nonvolatile Restore: FullSOCThr (013h) if nNVCfg0.enFT is set

Alternate Initial Value: 95% (0x5F05)

The nFullSOCThr register gates detection of end-of-charge. VFSOC must be larger than the nFullSOCThr value before nIChgTerm is compared to the AvgCurrent register value. The recommended nFullSOCThr register setting for most custom characterized applications is 95%. For EZ Performance applications, the recommendation is 80% (0x5005). See the *nIChgTerm Register (19Ch)* description and the *End-of-Charge Detection* section for details. Figure 22 shows the register format.

### nVEmpty Register (19Eh)

Register Type: Special

Nonvolatile Restore: VEmpty (03Ah) if nNVCfg0.enVE is set.

Alternate Initial Value: 0xA561 (VE=3.3V, VR=3.88V)

The nVempty register sets thresholds related to empty detection during operation. Figure 23 shows the register format.

**VE:** Empty Voltage. Sets the voltage level for detecting empty. A 10mV resolution gives a 0 to 5.11V range. This value is written to 3.3V after reset if nonvolatile backup is disabled.

**VR:** Recovery Voltage. Sets the voltage level for clearing empty detection. Once the cell voltage rises above this point, empty voltage detection is reenabled. A 40mV resolution gives a 0 to 5.08V range. This value is written to 3.88V after reset if nonvolatile backup is disabled.

### nDesignCap Register(1B3h)

Register Type: Capacity

Nonvolatile Restore: DesignCap (018h) if nNVCfg0.enDC is set.

Alternate Initial Value: FullCapRep register value

The nDesignCap register holds the expected capacity of the cell. This value is used to determine age and health of the cell by comparing against the measured present cell capacity.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nFullSOCThr | | | | | | | | | | | | | 1 | 0 | 1 |

*Figure 22. nFullSOCThr (1C6h)/FullSOCThr (013h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VE | | | | | | | | | VR | | | | | | |

*Figure 23. VEmpty (03Ah)/nVEmpty (19Eh) Format*

### nRFastVShdn Register (1D5h)

Register Type: Special

Nonvolatile Restore: RFast (015h) and VShdnCfg (0B8h) if nNVCfg1.enRFVSH is set

Alternate Initial Value: RFast defaults 0x0500 (312mΩ) and VShdnCfg defaults to 0x007D (2.5V)

When enabled the nRFastVShdn register is used to configure the initial values for the RFast and VShdnCfg registers. If nNVCfg1.enRFVSH is clear, nRFastVShdn can be used for general purpose data storage. Figure 24 shows the format.

**nRFast:** Restores to the RFast register using the following equation:

RFast = (nRFastVshdn AND 0xFF00) >> 4

**nVShdn**: Restores to the VShdnCfg register using the following equation:

VShdnCfg = nRFastVShdn AND 0x00FF

The RFast register value is used by the ModelGauge m5 algorithm to compensate an initial open-circuit voltage starting point if the IC is powered up or reset while the cell stack is under load and not relaxed. RFast is a unit-less scalar with an LSb of (100 x $R_{SENSE}$)/4096. The initial value of 0x0500 gives a default RFast value of 312.5 mΩ with a 10mΩ sense resistor.

The VShdnCfg register sets the voltage level at which the IC will enter shutdown mode. If the AvgVCell register voltage value that represents the lowest voltage of the cell stack falls below the VShdnCfg register value the IC will halt operation and enter shutdown mode. The VShdnCfg register has an LSb weight of 20mV. The initial value of 0x007D gives a default VShdnCfg value of 2.5V. See the *Modes of Operation* section.

### nIChgTerm Register (19Ch)

Register Type: Current

Nonvolatile Restore: IChgTerm (01Eh) if nNVCfg0.enICT is set.

Alternate Initial Value: 1/3rd the hex value of the nFullCapNom register (corresponds to C/9.6)

The nIChgTerm register allows the device to detect when a charge cycle of the cell has completed. nIChgTerm should be programmed to the exact charge termination current used in the application. The device detects end of charge if all the following conditions are met:

• VFSOC Register > FullSOCThr Register

• AND IChgTerm x 0.125 < Current Register < IChgTerm x 1.25

• AND IChgTerm x 0.125 < AvgCurrent Register < IChgTerm x 1.25

See *End-of-Charge Detection* section for more details.

### nRComp0 Register (1A6h)

Register Type: Special

Nonvolatile Restore: RComp0 (038h)

The nRComp0 register holds characterization information critical to computing the open-circuit voltage of a cell under loaded conditions.

### nTempCo Register (1A7h)

Register Type: Special

Nonvolatile Restore: TempCo (039h)

The nTempCo register holds temperature compensation information for the nRComp0 register value.

### nIAvgEmpty Register (1A8h)

Register Type: Current

Nonvolatile Backup and Restore: IAvgEmpty (036h) if nNVCfg2.enIAvg is set.

Alternate Initial Value: 0x0100

This register stores the typical current experienced by the fuelgauge when empty has occurred. If enabled, this register is periodically backed up to nonvolatile memory as part of the learning function.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nRFast | | | | | | | | nVShdn | | | | | | | |

*Figure 24. nRFastVshdn (1D5h) Format*

### ModelGauge m5 Algorithm Configuration Registers

The following registers allow operation of the ModelGauge m5 algorithm to be adjusted for the application. It is recommended that the default values for these registers be used.

### nFilterCfg Register (19Dh)

Register Type: Special

Nonvolatile Restore: FilterCfg (029h) if nNVCfg0.enFCfg is set.

Alternate Initial Value: 0x0EA4

The nFilterCfg register sets the averaging time period for all ADC readings for mixing OCV results and coulomb count results. It is recommended that these values are not changed unless absolutely required by the application. Figure 25 shows the nFilterCfg register format.

**CURR:** Sets the time constant for the AvgCurrent register. The default POR value of 0100b gives a time constant of 5.625s. The equation setting the period is:

AvgCurrent time constant = 45s x $2^{(CURR-7)}$

**VOLT:** Sets the time constant for the AvgVCell register. The default POR value of 010b gives a time constant of 45.0s. The equation setting the period is:

AvgVCell time constant = 45s x $2^{(VOLT-2)}$

**MIX:** Sets the time constant for the mixing algorithm. The default POR value of 1101b gives a time constant of 12.8 hours. The equation setting the period is:

Mixing Period = 45s x $2^{(MIX-3)}$

**TEMP:** Sets the time constant for the AvgTA register. The default POR value of 0001b gives a time constant of 1.5min. The equation setting the period is:

AvgTA time constant = 45s x $2^{TEMP}$

**1:** Write these bits to 0.

### nRelaxCfg Register (1B6h)

Register Type: Special

Nonvolatile Restore: RelaxCfg (02Ah) if nNVCfg0.enRCfg is set.

Alternate Initial Value: 0x2039

The nRelaxCfg register defines how the ICs detect if the cell is in a relaxed state. See Figure 33. For a cell to be considered relaxed, current flow through the cell must be kept at a minimum while the change in the cell's voltage over time, dV/dt, shows little or no change. If AvgCurrent remains below the LOAD threshold while VCell changes less than the dV threshold over two consecutive periods of dt, the cell is considered relaxed. Figure 26 shows the nRelaxCfg register format:

**LOAD:** Sets the threshold, which the AvgCurrent register is compared against. The AvgCurrent register must remain below this threshold value for the cell to be considered unloaded. Load is an unsigned 7-bit value where 1 LSb = 50µV. The default value is 800µV.

**dV:** Sets the threshold, which VCell is compared against. If the cell's voltage changes by less than dV over two consecutive periods set by dt, the cell is considered relaxed; dV has a range of 0 to 40mV where 1 LSb = 1.25mV. The default value is 3.75mV.

**dt:** Sets the time period over which change in VCell is compared against dV. If the cell's voltage changes by less than dV over two consecutive periods set by dt, the cell is considered relaxed. The default value is 1.5 minutes. The comparison period is calculated as:

Relaxation Period = $2^{(dt-8)}$ x 45s

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | TEMP | | | MIX | | | | VOLT | | | CURR | | | |

*Figure 25. FilterCfg (029h)/nFilterCfg (19Dh) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOAD | | | | | | | dV | | | | | dt | | | |

*Figure 26. RelaxCfg (02Ah)/nRelaxCfg (1B6h) Format*

### nLearnCfg Register (19Fh)

Register Type: Special

Nonvolatile Restore: LearnCfg (028h) if nNVCfg0.enLCfg is set.

Alternate Initial Value: 0x2603

The nLearnCfg register controls all functions relating to adaptation during operation. The factory default value for nLearnCfg is 0x2602. Figure 27 shows the register format:

**0:** Bit must be written 0. Do not write 1.

**1:** Bit must be written 1. Do not write 0.

**Filt Empty:** Empty Detect Filter. This bit selects whether empty is detected by a filtered or unfiltered voltage reading. Setting this bit to 1 causes the empty detection algorithm to use the AvgVCell register. Setting this bit to 0 forces the empty detection algorithm to use the VCell register. This default value is 0.

**LS:** Learn Stage. See Figure 11. The learn stage value controls the influence of the voltage fuel gauge on the mixing algorithm. The learn stage defaults to 0h, making the voltage fuel gauge dominate. The learn stage then advances to 7h over the course of two full cell cycles to make the coulomb counter dominate. Host software can write the learn stage value to 7h to advance to the final stage at any time. Writing any value between 1h and 6h is ignored.

### nMiscCfg Register (1B2h)

Register Type: Special

Nonvolatile Restore: MiscCfg (02Bh) if nNVCfg0.enMC is set

Alternate Initial Value: 0x3870

The nMiscCfg control register enables various other functions of the device. The nMiscCfg register default values should not be changed unless specifically required by the application. Figure 28 shows the register format:

**0:** Bit must be written 0. Do not write 1.

**1:** Bit must be written 1. Do not write 0.

**SACFG:** SOC Alert Config. SOC Alerts can be generated by monitoring any of the SOC registers as follows. SACFG defaults to 00 at power-up:

0 0 SOC alerts are generated based on the RepSOC register.

0 1 SOC alerts are generated based on the AvSOC register.

1 0 SOC alerts are generated based on the MixSOC register.

1 1 SOC alerts are generated based on the VFSOC register.

**MR:** Mixing Rate. This value sets the strength of the servo mixing rate after the final mixing state has been reached (> 2.08 complete cycles). The units are MR0 = 6.25µV, giving a range up to 19.375mA with a standard 0.010Ω sense resistor. Setting this value to 00000b disables servo mixing and the IC continues with time-constant mixing indefinitely. The default setting is 18.75µV or 1.875mA with a standard sense resistor.

**FUS:** Full Update Slope. This field prevents jumps in the RepSOC and FullCapRep registers by setting the rate of adjustment of FullCapRep near the end of a charge cycle. The update slope adjustment range is from 2% per 15 minutes (0000b) to a maximum of 32% per 15 minutes (1111b).

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | LS | | | 0 | Filt Empty | 1 | 0 |

*Figure 27. LearnCfg (028h)/nLearnCfg (19Fh) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FUS | | | | 1 | 0 | MR | | | | | 1 | 0 | 0 | SACFG | |

*Figure 28. MiscCfg (02Bh)/nMiscCfg (1B2h) Format*

### nTTFCfg Register (1C7h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

Alternate Initial Value: CV_HalfTime = 0xA00 (30 minutes) and CV_MixCap = 75% x FullCapNom

The nTTFCfg register configures parameters related to the time to full (TTF) calculation. There is no associated RAM register location that this register is recalled into after device reset. These parameters can be tuned for best TTF performance during characterization by Maxim. Figure 29 shows the register format.

**nCV_HalfTime:** Sets the HalfTime value with an LSb of 45s giving a full-scale range of 0 to 192 minutes.

**nCV_MixCapRatio:** Sets the MixCapRatio with an LSb of 1/256 giving a full-scale range of 0 to 0.9961.

### nConvgCfg Register (1B7h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nConvgCfg register configures operation of the converge to empty feature. The recommended value for nConvgCfg is 0x2241. Figure 30 shows the nConvgCfg register format. The nNVCfg1.CTE bit must be set to enable converge to empty functionality. If nNVCfg1.CTE is clear this register can be used as general purpose data storage.

**RepL_per_stage:** Adjusts the RepLow threshold setting depending on the present learn stage using the following equation. This allows the RepLow threshold to be at higher levels for earlier learn states. RepL_per_stage has an LSb of 1% giving a range of 0% to 7%.

RepLow Threshold = RepLow Field Setting +7% x RepL_per_stage.

**MinSlopeX:** Sets the amount of slope-shallowing that occurs when RepSOC falls below RepLow. MinSlopeX LSb corresponds to a ratio of 1/16 giving a full range of 0 to 15/16.

**VoltLowOff:** When the AvgVCell register value drops below the VoltLow threshold, RepCap begins to bend downwards by a ratio defined by the following equation. VoltLowOff has an LSb of 20mV giving a range of 0 to 620mV.

(AvgVCell - VEmpty)/VoltLowOff

**RepLow:** Sets the threshold below which RepCap begins to bend upwards. The RepLow field LSb is 2% giving a full- scale range from 0% to 30%.

### nRippleCfg Register (1B1h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nRippleCfg register configures ripple measurement and ripple compensation. The recommended value for this register is 0x0204. Figure 31 shows the register format.

**NR:** Sets the filter magnitude for ripple observation as defined by the following equation giving a range of 1.4 seconds to 180 seconds.

Ripple Time Range = 1.4 seconds • $2^{NR}$

**kDV:** Sets the corresponding amount of capacity to compensate proportional to the ripple.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nCV_HalfTime | | | | | | | | nCV_MixCapRatio | | | | | | | |

*Figure 29. nTTFCfg (1C7h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RepLow | | | | VoltLowOff | | | | | MinSlopeX | | | | RepL_per_stage | | |

*Figure 30. nConvgCfg (1B7h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kDV | | | | | | | | | | | | | NR | | |

*Figure 31. nRippleCfg (1B1h) Format*

### ModelGauge m5 Algorithm Additional Registers

The following registers contain intermediate ModelGauge m5 data that can be useful for debugging or performance analysis. The values in these registers become value 480ms after the IC is reset.

#### Timer Register (03Eh)

Register Type: Special

Nonvolatile Backup: None

Initial Value: 0x0000

This register holds timing information for the fuel gauge. It is available to the user for debug purposes. The Timer register LSb is equal to 175.8ms giving a full-scale range of 0 to 3.2 hours.

#### dQAcc Register (045h)

Register Type: Capacity (16mAh/LSB)

Nonvolatile Backup: Translated from nFullCapNom

Alternate Initial Value: 0x0017 (368mAh)

This register tracks change in battery charge between relaxation points. It is available to the user for debug purposes.

#### dPAcc Register (046h)

Register Type: Percentage (1/16% per LSB)

Nonvolatile Backup: None

Initial Value: 0x0190 (25%)

This register tracks change in battery state of charge between relaxation points. It is available to the user for debug purposes.

#### QResidual Register (00Ch)

Register Type: Capacity

Nonvolatile Backup: None

The QResidual register displays the calculated amount of charge in mAH that is presently inside of, but cannot be removed from the cell under present application conditions. This value is subtracted from the MixCap value to determine capacity available to the user under present conditions (AvCap).

#### VFSOC Register (0FFh)

Register Type: Percentage

Nonvolatile Backup: None

The VFSOC register holds the calculated present state of charge of the battery according to the voltage fuel gauge.

#### VFOCV Register (0FBh)

Register Type: Voltage

Nonvolatile Backup: None

The VFOCV register contains the calculated open-circuit voltage of the cell as determined by the voltage fuel gauge. This value is used in other internal calculations.

#### QH Register (4Dh)

Register Type: Capacity

Nonvolatile Backup: None

Alternate Initial Value: 0x0000

The QH register displays the raw coulomb count generated by the device. This register is used internally as an input to the mixing algorithm. Monitoring changes in QH over time can be useful for debugging device operation.

#### AvCap Register (01Fh)

Register Type: Capacity

Nonvolatile Backup: None

The AvCap register holds the calculated available capacity of the cell pack based on all inputs from the ModelGauge m5 algorithm including empty compensation. The register value is an unfiltered calculation. Jumps in the reported value can be caused by changes in the application such as abrupt changes in load current or temperature. See the *Empty Compensation* section for details.

#### AvSOC Register (00Eh)

Register Type: Percentage

Nonvolatile Backup: None

The AvSOC register holds the calculated available state of charge of the cell based on all inputs from the ModelGauge m5 algorithm including empty compensation. The AvSOC percentage corresponds with AvCap and FullCapNom. The AvSOC register value is an unfiltered calculation. Jumps in the reported value can be caused by changes in the application such as abrupt changes in load current or temperature. See the *Empty Compensation* section for details.

#### MixSOC Register (00Dh)

Register Type: Percentage

Nonvolatile Backup: None

The MixSOC register holds the calculated present state of charge of the cell before any empty compensation adjustments are performed. MixSOC corresponds with MixCap and FullCapNom. See the *Empty Compensation* section for details.

### MixCap Register (00Fh)

Register Type: Capacity

Nonvolatile Backup: None

The MixCap register holds the calculated remaining capacity of the cell before any empty compensation adjustments are performed. See the *Empty Compensation* section for details.

### VFRemCap Register (04Ah)

Register Type: Capacity

Nonvolatile Backup: None

The VFRemCap register holds the remaining capacity of the cell as determined by the voltage fuel gauge before any empty compensation adjustments are performed. See the *Empty Compensation* section for details.

### FStat Register (03Dh)

Register Type: Special

Nonvolatile Backup: None

The FStat register is a read-only register that monitors the status of the ModelGauge algorithm. Do not write to this register location. Figure 32 is the FStat register format:

**DNR:** Data Not Ready. This bit is set to 1 at cell insertion and remains set until the output registers have been updated. Afterwards, the ICs clear this bit indicating the fuel gauge calculations are now up to date. This takes between 445ms and 1.845s depending on whether the ICs are in a powered state prior to the cell-insertion event.

**RelDt2:** Long Relaxation. This bit is set to a 1 whenever the ModelGauge m5 algorithm detects that the cell has been relaxed for a period of 48 to 96 minutes or longer. This bit is cleared to 0 whenever the cell is no longer in a relaxed state. See Figure 33.

**FQ:** Full Qualified. This bit is set when all charge termination conditions have been met. See the *End-of-Charge Detection* section for details.

**EDet:** Empty Detection. This bit is set to 1 when the IC detects that the cell empty point has been reached. This bit is reset to 0 when the cell voltage rises above the recovery threshold. See the VEmpty register for details.

**RelDt:** Relaxed Cell Detection. This bit is set to a 1 whenever the ModelGauge m5 algorithm detects that the cell is in a fully relaxed state. This bit is cleared to 0 whenever a current greater than the Load threshold is detected. See Figure 33.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | RelDt | EDet | FQ | RelDt2 | X | X | X | X | X | DNR |

*Figure 32. FStat (03Dh) Format*

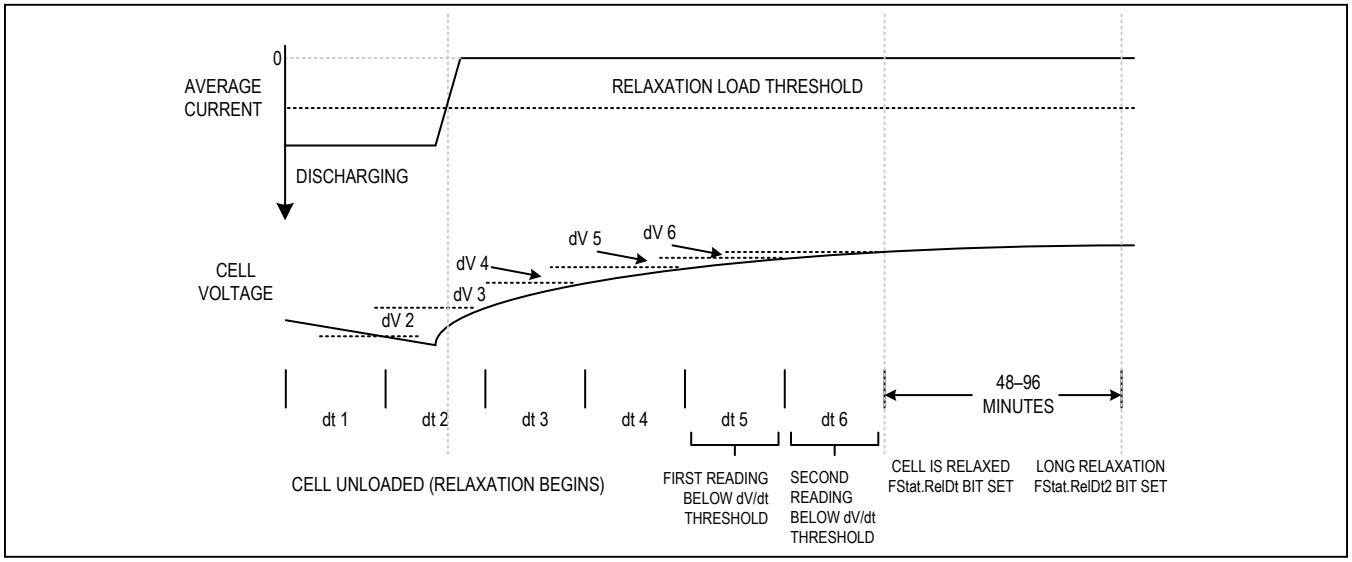



*Figure 33. Cell Relaxation Detection*

## Modes of Operation

The ICs operate in one of three power modes: shutdown, hibernate, and active. While in active mode, the ICs operate as a high-precision fuel gauge with temperature, voltage, auxiliary inputs, current, and accumulated current measurements acquired continuously, and the resulting values updated in the measurement registers. Hibernate mode is a reduced power consumption and reduced activity, but full function version of active mode with the balance between power consumption and activity configurable by the application. In shutdown mode, the internal LDO regulators are disabled, all activity stops, all register and fuel-gauge output values are lost, but nonvolatile configuration values are preserved. Any learned information not yet stored to nonvolatile memory is lost. Figure 34 shows the relationship between the different operating modes.

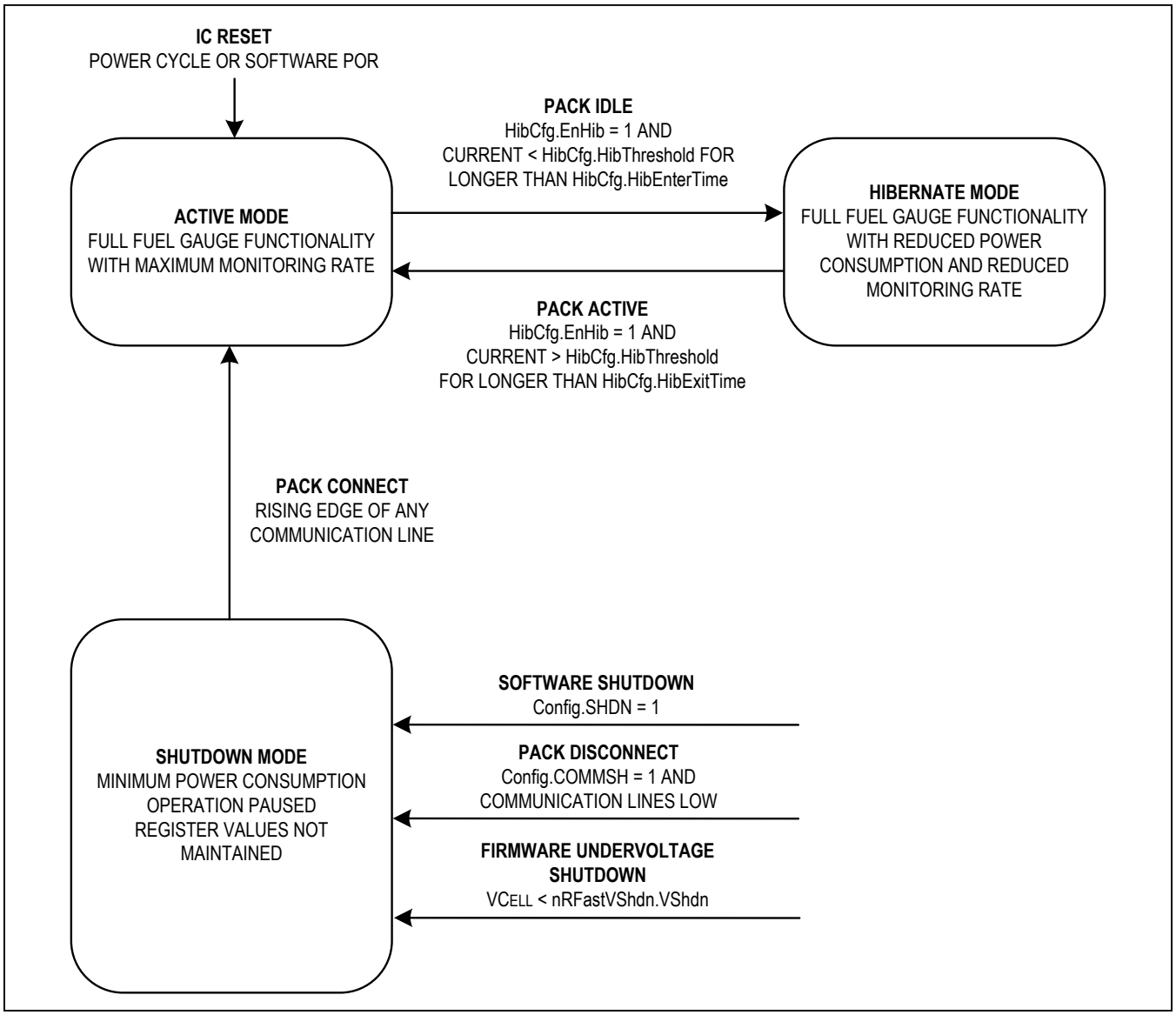



*Figure 34. Flowchart of Operating Modes*

Entering hibernate mode:

- Pack Idle. Hibernate mode must be enabled by setting HibCfg.EnHib = 1. The ICs then enter hibernate mode if the absolute value of the Current register falls below the HibThreshold setting for longer than HibEnterTime. See the HibCfg register for details.

Exiting hibernate mode:

- Pack Active. The ICs return to active mode if the absolute value of the Current register rises above the HibThreshold setting for longer than HibExitTime. See the HibCfg register for details.

Entering shutdown mode (from active mode or hibernate mode):

- Software Shutdown. Software shutdown can be forced by setting Config.SHDN = 1 and waiting longer than the ShdnTimer register value (default 45s).
- Firmware Undervoltage Shutdown. The ICs enter shutdown mode if the VCell register value representing the lowest voltage of the cell stack falls below the nRFastVShdn.VShdn threshold setting (2.5V by default).
- Pack Disconnect. The IC enters shutdown if Config.COMMSH = 1 and communication lines are open (logic-low) for longer than the ShdnTimer period. This option is not recommended.

These shutdown entry modes are all programmable according to the application. Shutdown events are gated by the ShdnTimer register, which allows a long delay between the shutdown event and entering the mode. By behaving this way, the IC takes the best reading of the relaxation voltage.

Exiting Shutdown Mode (ICs always exit into active mode):

- Pack Connect. The ICs return to active mode on the rising edge of any communication line.
- IC Reset. If the ICs are power cycled or the software RESET command is sent, the ICs return to active mode of operation.

IC communication is always allowed in any mode of operation. See the detailed descriptions of the ShdnTimer, HibCfg, and Config registers. If the IC was previously in shutdown mode, starting communication wakes it up and the data is invalid for 550ms.

## Status and Configuration Registers

The following registers control IC operation not related to the fuel gauge such as power-saving modes, nonvolatile backup, and ALRT1 pin functionality.

### DevName Register (021h)

Register Type: Special

Nonvolatile Backup: None

The DevName register holds device type and firmware revision information. This allows host software to easily identify the type of IC being communicated to. Figure 35 shows the DevName register format.

**Device:** Indicates the device type as follows:

1h = MAX17201 or MAX17211

5h = MAX17205 or MAX17215

**Revision:** Indicates the firmware revision inside the ICs.

### nROMID0 (1BCh)/nROMID1 (1BDh)/nROMID2 (1BEh)/nROMID3 (1BFh) Registers

Register Type: Special

Nonvolatile Restore: There are no associated restore locations for these registers

Each MAX1720x/1x IC contains a unique 64 bit identification value that is contained in the nROMID registers. Note this is the same ID that can be read using the MAX17211/15 1-Wire ROM ID commands. The unique ID can be reconstructed from the nROMID registers as shown in Figure 36.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Revision | | | | | | | | | | | | Device | | | |

*Figure 35. DevName (021h) Format*

| NROMID3[15:0] | NROMID2[15:0] | NROMID1[15:0] | NROMID0[15:0] |
|---|---|---|---|
| ROM ID [63:48] | ROM ID [47:32] | ROM ID [31:16] | ROM ID [15:0] |

*Figure 36. nROMID (1BCh to 1BFh) Format*

### nRSense Register (1CFh)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nRSense register is the designated location to store the sense resistor value used by the application. This value is not used by the IC as all current and capacity information is reported in terms of µV and µVH. Host software can use the nRSense register value to convert current and capacity information into mA and mAH. It is recommended that the sense resistor value be stored with an LSb weight of 10µΩ giving a range of 10µΩ to 655.35mΩ. Table 7 shows recommended register settings based on common-sense resistor values.

### nPackCfg Register (1B5h)

Register Type: Special

Nonvolatile Restore: PackCfg (0BDh)

The nPackCfg register configures the voltage and temperature inputs to the ADC and also to the fuel gauge. nPackCfg configuration must match the pack hardware for proper operation of the IC. See the *Typical Operating Circuits* section for recommended nPackCfg settings based on operating circuit configuration. The default factory setting for nPackCfg is 0x0C01 for the MAX172x1 and 0x0A02 for the MAX172x5. Figure 37 shows the register format.

**NCELLS:** Number of Cells. This field configures the ICs for the number of cells in series in the cell pack. This field value must match the cell stack size for proper operation of the fuel gauge and other IC functions.

**0:** Always write 0.

**BALCFG:** Balancing Config. This field sets the cell balancing voltage threshold. When set to 0 cell balancing is disabled. When set to any other value, cell balancing begins when a voltage delta as determined by the following equation is detected. Note there are other criteria for determining the start of cell balancing. See the *Cell Balancing (MAX17205/MAX17215 Only)* section for details.

Balancing Threshold = $1.25mV \cdot 2^{BALCFG}$

**CxEn:** CELLx Channel Enable. Set to 1 to enable voltage measurements of the CELLx pin. Voltage measured from CELLx will be used as an input to the fuel gauge if CxEn = 1 and ChEn = 0, regardless of BtEn state.

**BtEn:** BATT Channel Enable. Set to 1 to enable voltage measurements of the $V_{BATT}$ pin. Voltage measured from $V_{BATT}$ will be used as an input to the fuel gauge only if BtEn = 1, CxEn = 0, and ChEn = 0.

**ChEn:** CELL Channel Enable. Set to 1 to enable voltage measurements of the CELL1, CELL2, and $V_{BATT}$ pins. Voltage measured from these pins is used as an input to the fuel gauge if ChEn = 1 regardless of CxEn and BtEn states.

**TdEn:** Die Temperature Enable. Set to 1 to enable internal temperature measurements.

**A1En:** AIN1 Channel Enable. Set to 1 to enable temperature measurements on the AIN1 pin.

**A2En:** AIN2 Channel Enable. Set to 1 to enable temperature measurements on the AIN2 pin.

**FGT:** Fuel Gauge Temperature Input Select. FGT in combination with the TdEn, A1en, and A2en bits determines which temperature measurement is used by the fuel gauge. This allows multiple temperature inputs to be measured and still control which one is the input to the fuel gauge. Table 8 shows how the fuel gauge input is selected.

## Table 7. Recommended nRSense Register Values for Common-Sense Resistors

| SENSE RESISTOR | NRSENSE REGISTER |
|---|---|
| 0.005Ω | 0x01F4 |
| 0.010Ω | 0x03E8 |
| 0.020Ω | 0x07D0 |

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FGT | 0 | A2En | A1En | TdEn | ChEn | BtEn | CxEn | BALCFG | | | 0 | NCELLS | | | |

*Figure 37. PackCfg (0BDh)/nPackCfg (1B5h) Format*

## Table 8. Fuel Gauge Temperature Input

| FGT | A2EN | A1EN | TDEN | FUEL GAUGE INPUT/TEMP REGISTER |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | Internal/DieTemp (135h) |
| 1 | 0 | 1 | 0 | AIN1/Temp1 (134h) |
| 0 | 1 | 0 | 0 | AIN2/Temp2 (13Bh) |
| 0 | 1 | 0 | 1 | AIN2/Temp2 (13Bh) |
| 0 | 0 | 1 | 1 | Internal/DieTemp (135h) |
| 1 | 0 | 1 | 1 | AIN1/Temp1 (134h) |
| 1 | 1 | 1 | 0 | AIN1/Temp1 (134h) |
| 0 | 1 | 1 | 0 | Temp1 (134h) and Temp2 (13Bh) |
| 1 | 1 | 1 | 1 | AIN1/Temp1 (134h) |
| 0 | 1 | 1 | 1 | Temp1 (134h) and Temp2 (13Bh) |
| All Other Configurations | | | | Illegal Configuration |

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| enOCV | enX | 0 | 0 | enCfg | enFCfg | enRCfg | enLCfg | enICT | enCG | enVE | enDC | enMC | enAF | enHCfg | enSBS |

*Figure 38. nNVCfg0 (1B8h) Format*

### nNVCfg0 Register (1B8h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nNVCfg0 register manages nonvolatile memory backup of device and fuel gauge register RAM locations. Each bit of the nNVCfg0 register, when set, enables a given register location to be restored from a corresponding nonvolatile memory location after reset of the IC. If nonvolatile restore of a given register is not enabled, that location initializes to a default value after reset instead. See the individual register descriptions for details. The factory default value for nNVCfg0 register is 0x0100. Figure 38 shows the nNVCfg0 register format.

**enSBS:** Enable SBS. This bit enables SBS functions of the IC. When set, all registers accessed with the SBS 2-Wire address will be regularly updated. When this bit is clear all SBS related nonvolatile configuration register locations can be used as general-purpose user memory.

**enHCfg:** Enable HibCfg restore. Set this bit to enable HibCfg register to be restored after reset by the nHibCfg register. When this bit is cleared, HibCfg restores with its alternate initialization value and nHibCfg can be used for general-purpose data storage.

**enAF:** Enable Age Forecasting. Set this bit to enable Age Forecasting functionality. When this bit is clear nAgeFcCfg can be used for general purpose data storage. When set, nSOC and nVoltTemp become repurposed for age forecasting data. When enAF is set to 1, nNVCfg2.enVT and nNVCfg2.enSOC must be 0 for proper operation.

**enMC:** Enable MiscCfg restore. Set this bit to enable MiscCfg register to be restored after reset by the nMiscCfg register. When this bit is cleared, MiscCfg restores with its alternate initialization value and nMiscCfg can be used for general- purpose data storage.

**enDC:** Enable DesignCap restore. Set this bit to enable DesignCap register to be restored after reset by the nDesignCap register. When this bit is cleared, DesignCap restores with its alternate initialization value and nDesignCap can be used for general-purpose data storage.

**enVE:** Enable VEmpty restore. Set this bit to enable VEmpty register to be restored after reset by the nVEmpty register. When this bit is cleared, VEmpty restores with its alternate initialization value and nVEmpty can be used for general purpose data storage.

**enCG:** Enable CGain and COff restore. Set this bit to enable CGain and COff registers to be restored after reset by the nCGain register. When this bit is cleared, CGain and COff restore with their alternate initialization values and nCGain can be used for general-purpose data storage.

**enICT:** Enable IChgTerm restore. Set this bit to enable IChgTerm register to be restored after reset by the nIChgTerm register. When this bit is cleared, IChgTerm restores to a value of 1/3rd of the nFullCapNom register and nIChgTerm can be used for general-purpose data storage.

**enLCfg:** Enable LearnCfg restore. Set this bit to enable LearnCfg register to be restored after reset by the nLearnCfg register. When this bit is cleared, LearnCfg restores with its alternate initialization value and nLearnCfg can be used for general-purpose data storage.

**enRCfg:** Enable RelaxCfg restore. Set this bit to enable RelaxCfg register to be restored after reset by the nRelaxCfg register. When this bit is cleared, RelaxCfg restores with its alternate initialization value and nRelaxCfg can be used for general-purpose data storage.

**enFCfg:** Enable FilterCfg restore. Set this bit to enable FilterCfg register to be restored after reset by the nFilterCfg register. When this bit is cleared, FilterCfg restores with its alternate initialization value and nFilterCfg can be used for general-purpose data storage.

**enCfg:** Enable Config and Config2 restore. Set this bit to enable Config and Config2 registers to be restored after reset by the nConfig register. When this bit is cleared, Config and Config2 restore with their alternate initialization values and nConfig can be used for general-purpose data storage.

**0:** This location must remain 0. Do not write this location to 1.

**enX:** Enable XTable restore. Set this bit to enable nXTable register locations to be used for cell characterization data. When this bit is cleared, the ICs use the default cell model and all nXTable register locations can be used as general-purpose user memory.

**enOCV:** Enable OCVTable restore. Set this bit to enable nOCVTable register locations to be used for cell characterization data. When this bit is cleared, the ICs use the default cell model and all nOCVTable register locations can be used as general-purpose user memory.

### nNVCfg1 Register (1B9h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nNVCfg1 register manages nonvolatile memory restore of device and fuel gauge register RAM locations. Each bit of the nNVCfg1 register, when set, enables a given register location to be restored from a corresponding nonvolatile memory location after reset of the ICs. If nonvolatile backup of a given register is not enabled, that location initializes to a default value after reset instead. See the individual register descriptions for details. Figure 39 shows the nNVCfg1 register format.

**0:** This location must remain 0. Do not write this location to 1.

**enCTE:** Enable Converge-to-Empty. Set this bit to enable the nConvgCfg register settings to affect the converge to empty functionality of the IC. When this bit is clear, converge-to-empty is disabled and nConvgCfg can be used for general- purpose data storage.

**enCrv:** Enable Curve Correction. Set this bit to enable curvature correction on thermistor readings, improving thermistor translation performance to -40°C to +80°C (instead of -10°C to +50°C). Note that enCrv and enMtl are mutually exclusive functions. Do not set both enCrv and enMtl at the same time.

**enAT:** Enable Alert Thresholds. Set this bit to enable IAlrtTh, VAlrtTh, TAlrtTh, and SAlrtTh registers to be restored after reset by the nIAlrtTh, nVAlrtTh, nTAlrtTh, and nSAlrtTh registers respectively. When this bit is cleared, these registers restore with their alternate initialization values and the nonvolatile locations can be used for general-purpose data storage.

**enTTF:** Enable time to full configuration. Set to 1 to enable nTTFCfg (configures CVMixCap and CVHalftime) for tuning of time-to-full performance. Otherwise, CVMixCap and CVHalftime restore to their alternate initialization values and nTTFCfg can be used for general-purpose data storage.

**enODSC:** Enable OD and SC overcurrent comparators. Set this bit to enable ODSCTh and ODSCCfg registers to be restored after reset by the nODSCTh and nODSCCfg registers. When this bit is cleared, ODSCTh and ODSCCfg restore with their alternate initialization values (comparators disabled) and nODSCTh and nODSCCfg can be used for general- purpose data storage.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| enTGO | enMtl | enFTh | enRFVSH | enODSC | 0 | 0 | 0 | 0 | 0 | 0 | enTTF | enAT | enCrv | enCTE | 0 |

*Figure 39. nNVCfg1 (1B9h) Format*

**enRFVSH:** Enable RFast and VShdnCfg restore. Set this bit to enable RFast and VShdnCfg registers to be restored after reset by the nRFastVShdn register. When this bit is cleared, RFast and VShdnCfg restore with their alternate initialization values and nRFastVShdn can be used for general-purpose data storage.

**enFTh:** Enable FullSOCThr configuration restore. Set this bit to enable FullSOCThr register to be restored after reset by the nFullSOCThr register. When this bit is cleared, FullSOCThr restore with its alternate initialization value and nFullSOCThr can be used for general-purpose data storage.

**enMtl:** Enable CGTempCo restore. Set this bit to enable CGTempCo register to be restored after reset by the nTCurve register. When this bit is cleared, CGTempCo restores with its alternate initialization value. nTCurve can be used for general-purpose data storage if both enCrv and enMtl are clear. Do not set both enCrv and enMtl at the same time.

**enTGO:** Enable TGain and TOff restore. Set this bit to enable TGain and TOff registers to be restored after reset by the nTGain and nTOff registers. When this bit is cleared, TGain and TOff restore with their alternate initialization values. nTGain and nTOff can then be used for general-purpose data storage.

### nNVCfg2 Register (1BAh)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nNVCfg2 register manages nonvolatile memory backup and restore of device and fuel gauge register RAM locations. Each bit of the nNVCfg2 register, when set, enables a given register location to be restored from or backed up to a corresponding nonvolatile memory location after reset of the ICs. If nonvolatile backup of a given register is not enabled, that location initializes to a default value after reset instead. See the individual register descriptions for details. Figure 40 shows the nNVCfg2 register format.

**CYCLESpSAVE:** Cycles per Save. This field defines the number of equivalent full cell cycles between backup operations. A backup operation occurs each time the Cycles register exceeds the nCycles register plus the CYCLESpSAVE value. CYCLESpSAVE has an LSb weight of 0.5 cycles giving a full range of 0.5 to 64.5 cycles (with an offset of +0.5 cycles). For example, set CYCLESpSAVE to 19d to automatically save every 10 cycles. Since the MAX1720x/1x provides 202 history writes, choose CYCLESpSAVE carefully according to the maximum battery cycle lifespan. For example, if the battery supports 1000 cycles, set CYCLESpSAVE to at least 5 cycles (CYCLESpSAVE = 9).

**enMet:** Enable metal current sensing. Setting this bit to 1 enables temperature compensation of current readings for allowing copper trace current sensing. This also forces the PackCfg.TdEn bit to 1 after reset of the ICs to guarantee internal temperature measurements occur. See also nNVCfg1.enMtl, which enables nTCurve register operation for adjustment of the current sensing temperature coefficient.

**enIAvg:** Enable IAvgEmpty backup and restore. Set this bit to enable IAvgEmpty register to be restored after reset by the nIAvgEmpty register. When this bit is clear IAvgEmpty will restore with its alternate initialization value and nIAvgEmpty can be used as general-purpose memory.

**enFC:** Enable FullCap and FullCapRep backup and restore. Set this bit to enable FullCap and FullCapRep registers to be restored after reset by the nFullCapRep register and FullCapRep to backup to nFullCapRep. When this bit is clear FullCap and FullCapRep registers restore from the nFullCapNom register. nFullCapRep can then be used as general purpose user memory.

**enVT:** Enable Voltage and Temperature backup. Set this bit to enable storage of AvgVCell and AvgTA register information into the nVoltTemp register during save operations. There is no corresponding restore option. When nNVCfg2.enVT and nNVCfg0.enAF are clear nVoltTemp can be used as general purpose memory. Note that enVT should not be set simultaneously with nNVCfg0.enAF (AgeForecasting).

**enMMC:** Enable MinMaxCurr Backup. Set this bit to enable storage of MinMaxCurr register information into the nMinMaxCurr register during save operations. There is no corresponding restore option. When this bit is clear nMinMaxCurr can be used as general-purpose memory.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| enT | enSOC | enMMT | enMMV | enMMC | enVT | enFC | enIAvg | enMet | CYCLESpSAVE | | | | | | |

*Figure 40. nNVCfg2 (1BAh) Format*

**enMMV:** Enable MinMaxVolt Backup. Set this bit to enable storage of MinMaxVolt register information into the nMinMaxVolt register during save operations. There is no corresponding restore option. When this bit is clear nMinMaxVolt can be used as general-purpose memory.

**enMMT:** Enable MinMaxTemp Backup. Set this bit to enable storage of MinMaxTemp register information into the nMinMaxTemp register during save operations. There is no corresponding restore option. When this bit is clear nMinMaxTemp can be used as general-purpose memory.

**enSOC:** Enable MixSOC and VFSOC Backup. Set this bit to enable storage of MixSOC and VFSOC register information into the nSOC register during save operations. There is no corresponding restore option. When this bit and nNVCfg0.enAF are clear nSOC can be used as general purpose memory. Note that enSOC should not be set simultaneously with nNVCfg0.enAF (AgeForecasting).

**enT:** Enable TimerH backup and restore. Set this bit to enable TimerH register to be backed up and restored by the nTimerH register. When this bit is cleared, TimerH restores with its alternate initialization value and nTimerH can be used as general-purpose memory.

### ShdnTimer Register (03Fh)

Register Type: Special

Nonvolatile Backup: None

Initial Value: 0x0000

The ShdnTimer register sets the timeout period from when a shutdown event is detected until the device disables the regulators and enters low-power mode. Figure 41 shows the ShdnTimer register format.

**CTR:** Shutdown Counter. This register counts the total amount of elapsed time since the shutdown trigger event. This counter value stops and resets to 0 when the shutdown timeout completes. The counter LSb is 1.4s

**THR:** Sets the shutdown timeout period from a minimum of 45s to a maximum of 1.6h. The default POR value of 0h gives a shutdown delay of 45s in active mode. In default hibernate mode, the minimum shutdown time is 12 minutes. The equation setting the period is:

Shutdown Timeout Period = 175.8ms • $2^{(8+THR)}$ in active mode

### nConfig Register (1B0h)

Register Type: Special

Nonvolatile Restore: Config (01Dh) and Config2 (0BBh) if nNVCfg0.enCfg is set.

Alternate Initial Value: 0x2210 for Config, 0x0050 for Config2

The nConfig register holds all shutdown enable, alert enable, and temperature enable control bits. Writing a bit location enables the corresponding function within one task period. Figure 42, Figure 43, and Figure 44 show the register formats.

**0:** Bit must be written 0. Do not write 1.

**1:** Bit must be written 1. Do not write 0.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| THR | | | CTR | | | | | | | | | | | | |

*Figure 41. ShdnTimer (03Fh) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | SS | TS | VS | ALRTp | AINSH | Ten | Tex | SHDN | COMMSH | ALSH | 1 | FTHRM | Aen | dSOCen | TAlrtEn |

*Figure 42. nConfig (1B0h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | SS | TS | VS | ALRTp | AINSH | Ten | Tex | SHDN | COMMSH | 0 | ETHRM | FTHRM | Aen | Bei | Ber |

*Figure 43. Config (01Dh) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | dSOCen | TAlrtEn | 0 | 1 | 0 | 0 | 0 | POR_CMD |

*Figure 44. Config2 (0BBh) Format*

**Ber:** Enable alert on battery removal when the ICs are mounted host side. When Ber = 1, a battery-removal condition, as detected by the AIN pin voltage, triggers an alert. Note that if this bit is set to 1, the ALSH bit should be set to 0 to prevent an alert condition from causing the device to enter shutdown mode.

**Bei:** Enable alert on battery insertion when the ICs are mounted host side. When Bei = 1, a battery-insertion condition, as detected by the AIN pin voltage, triggers an alert. Note that if this bit is set to 1, the ALSH bit should be set to 0 to prevent an alert condition from causing the device to enter shutdown mode.

**Aen:** Enable alert on fuel-gauge outputs. When Aen = 1, exceeding any of the alert threshold register values by current, voltage, or SOC triggers an alert. This bit affects the ALRT1 pin operation only. The Smx, Smn, Tmx, Tmn, Vmx, Vmn, Imx, and Imn bits of the Status register (000h) are not disabled. Note that if this bit is set to 1, the ALSH bit should be set to 0 to prevent an alert condition from causing the device to enter shutdown mode.

**FTHRM:** Force Thermistor Bias Switch. This allows the host to control the bias of the thermistor switch or enable fast detection of battery removal. Set FTHRM = 1 to always enable the thermistor bias switch. With a standard 10kΩ thermistor, this adds an additional ~200µA to the current drain of the circuit.

**ETHRM:** Enable Thermistor. Set to logic 1 to enable the automatic THRM output bias and AIN1/AIN2 measurement.

**COMMSH:** Communication Shutdown. Set to logic 1 to force the device to enter shutdown mode if both SDA and SCL are held low (MAX1720x) or DQ is held low (MAX1721x) for more than timeout of the ShdnTimer register. This also configures the device to wake up on a rising edge of any communication. Note that if COMMSH and AINSH are both set to 0, the device wakes up an edge of any of the DQ/SDA or OD/SCL pins. See the *Modes of Operation* section.

**SHDN:** Shutdown. Write this bit to logic 1 to force a shutdown of the device after timeout of the ShdnTimer register (default 45s delay). SHDN is reset to 0 at power-up and upon exiting shutdown mode. In order to command shutdown within 45s, first write HibCFG = 0x0000 to enter active mode.

**Tex:** Temperature External. When set to 1, the fuel gauge requires external temperature measurements to be written from the host. When set to 0, the IC's own measurements as configured by the PackCfg register are used instead.

**Ten:** Enable Temperature Channel. Set to 1 and set ETHRM or FTHRM to 1 to enable temperature measurements as defined in the PackCfg register.

**AINSH:** AIN1 Pin Shutdown. Set to 1 to enable device shutdown when the IC is mounted host side and the battery is removed. The ICs enter shutdown if the AIN1 pin remains high (AIN1 > $V_{THRM}$ - $V_{DET}$) for longer than the timeout of the ShdnTimer register. This also configures the device to wake up when AIN1 is pulled low on cell insertion. Note that if COMMSH and AINSH are both set to 0, the device wakes up an edge of any of the DQ/SDA or OD/SCL pins.

**ALRTp:** ALRT1 Pin Polarity. If ALRTp = 0 the ALRT1 pin is active low and if ALRTp = 1 the ALRT1 pin is active high.

**VS:** Voltage ALRT1 Sticky. When VS = 1, voltage alerts can only be cleared through software. When VS = 0, voltage alerts are cleared automatically when the threshold is no longer exceeded.

**TS:** Temperature ALRT1 Sticky. When TS = 1, temperature alerts can only be cleared through software. When TS = 0, temperature alerts are cleared automatically when the threshold is no longer exceeded.

**SS:** SOC ALRT1 Sticky. When SS = 1, SOC alerts can only be cleared through software. When SS = 0, SOC alerts are cleared automatically when the threshold is no longer exceeded.

**POR_CMD:** Firmware Restart. Set this bit to 1 to restart IC firmware operation without performing a recall of nonvolatile memory into RAM. This allows different IC configurations to be tested without changing nonvolatile memory settings. This bit is set to 0 at power-up and automatically clears itself after firmware restart. See *Reset Commands.*

**TAlrten:** Temperature Alert Enable. Set this bit to 1 to enable temperature based alerts. Write this bit to 0 to disable temperature alerts. This bit is set to 1 at power-up.

**dSOCen:** 1% SOC Change Alert Enable. Set this bit to 1 to alert output on 1% SOC change. This bit is set to 0 at power-up.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Br | Smx | Tmx | Vmx | Bi | Smn | Tmn | Vmn | dSOCi | Imx | X | X | Bst | Imn | POR | X |

*Figure 45. Status (000h) Format*

**Status Register (000h)**

Register Type: Special

Nonvolatile Backup: None

Initial Value: 0x0002

The Status register maintains all flags related to alert thresholds and battery insertion or removal. Figure 45 shows the Status register format.

**POR:** Power-On Reset. This bit is set to a 1 when the device detects that a software or hardware POR event has occurred. This bit must be cleared by system software to detect the next POR event. POR is set to 1 at power-up.

**Imn:** Minimum Current Alert Threshold Exceeded. This bit is set to a 1 whenever a Current register reading is below the minimum IAlrtTh value. This bit is cleared automatically when current rises above minimum IAlrtTh value. Imn is set to 0 at power-up.

**Bst:** Battery Status. Useful when the IC is used in a host side application. This bit is set to 0 when a battery is present in the system and set to 1 when the battery is absent. Bst is set to 0 at power-up.

**Imx:** Maximum Current Alert Threshold Exceeded. This bit is set to a 1 whenever a Current register reading is above the maximum IAlrtTh value. This bit is cleared automatically when current falls below maximum IAlrtTh value. Imx is set to 0 at power-up.

**dSOCi:** State of Charge 1% Change Alert. This is set to 1 whenever the RepSOC register crosses an integer percentage boundary such as 50.0%, 51.0%, etc. Must be cleared by host software. dSOCi is set to 0 at power-up.

**Vmn:** Minimum Voltage Alert Threshold Exceeded. This bit is set to a 1 whenever a VCell register reading is below the minimum VAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config.VS bit description. Vmn is set to 0 at power-up.

**Tmn:** Minimum Temperature Alert Threshold Exceeded. This bit is set to a 1 whenever a Temperature register reading is below the minimum TAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config.TS bit description. Tmn is set to 0 at power-up.

**Smn:** Minimum SOC Alert Threshold Exceeded. This bit is set to a 1 whenever SOC falls below the minimum SAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config. SS and MiscCFG.SACFG bit descriptions. Smn is set to 0 at power-up.

**Bi:** Battery Insertion. Useful when the IC is used in a host-side application. This bit is set to a 1 when the device detects that a battery has been inserted into the system by monitoring the AIN1 pin. This bit must be cleared by system software to detect the next insertion event. Bi is set to 0 at power-up.

**Vmx:** Maximum Voltage Alert Threshold Exceeded. This bit is set to a 1 whenever a VCell register reading is above the maximum VAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config.VS bit description. Vmx is set to 0 at power-up.

**Tmx:** Maximum Temperature Alert Threshold Exceeded. This bit is set to a 1 whenever a Temperature register reading is above the maximum TAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config.TS bit description. Tmx is set to 0 at power-up.

**Smx:** Maximum SOC Alert Threshold Exceeded. This bit is set to a 1 whenever SOC rises above the maximum SAlrtTh value. This bit may or may not need to be cleared by system software to detect the next event. See Config. SS and MiscCFG.SACFG bit descriptions. Smx is set to 0 at power-up.

**Br:** Battery Removal. Useful when the ICs are used in a host-side application. This bit is set to a 1 when the system detects that a battery has been removed from the system. This bit must be cleared by system software to detect the next removal event. Br is set to 0 at power-up.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | Hib | PORSkip |

*Figure 46. Status2 (0B0h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EnHib | HibEnterTime | | | HibThreshold | | | | 0 | 0 | 0 | HibExitTime | | HibScalar | | |

*Figure 47. HibCfg (0BAh)/nHibCfg (1B4h) Format*

### Status2 Register (0B0h)

Register Type: Special

Nonvolatile Backup: None

Initial Value: 0x0000

The Status2 register maintains status of hibernate mode. Figure 46 shows the Status register format.

**PORSkip:** This bit is set when initialization of the IC's RAM is skipped during reset. This occurs under conditions where the RAM values are already valid prior to a reset such as when the firmware restart command is sent to the ICs by software.

**Hib:** Hibernate Status. This bit is set to a 1 when the device is in hibernate mode or 0 when the device is in active mode. Hib is set to 0 at power-up.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

### nHibCfg Register (1B4h)

Register Type: Special

Nonvolatile Restore: HibCfg (0BAh) if nNVCfg0.enHCfg is set.

Alternate Initial Value: 0x890B (5.625 second hibernate mode)

The nHibCfg register controls hibernate mode functionality. The ICs enters hibernate mode if the measured system current falls below the HibThreshold setting for longer than the HibEnterTime delay. While in hibernate mode, the ICs reduce its operating current by slowing down its task period as defined by the HibScalar setting. The ICs automatically return to active mode of operation if current readings go above the HibThreshold setting for longer than the HibExitTime delay. Figure 47 shows the register format.

**0:** Bit must be written 0. Do not write 1.

**HibScalar:** Sets the task period while in hibernate mode based on the following equation:

$$\text{Hibernate Mode Task Period(s)} = 702\text{ms} \times 2^{(\text{HibScalar})}$$

**HibExitTime:** Sets the required time period of consecutive current readings above the HibThreshold value before the IC exits hibernate and returns to active mode of operation.

$$\text{Hibernate Mode Exit Time(s)} = (\text{HibExitTime} + 1) \times 702\text{ms} \times 2^{(\text{HibScalar})}$$

**HibThreshold:** Sets the threshold level for entering or exiting hibernate mode. The threshold is calculated as a fraction of the full capacity of the cell using the following equation:

$$\text{Hibernate Mode Threshold(mA)} = (\text{FullCap}(\mu\text{VH})/(0.8\text{ hours} \times R_{SENSE})/2^{(\text{HibThreshold})}$$

**HibEnterTime:** Sets the time period that consecutive current readings must remain below the HibThreshold value before the ICs enter hibernate mode as defined by the following equation. The default HibEnterTime value of 000b causes the ICs to enter hibernate mode if all current readings are below the HibThreshold for a period of 5.625 seconds, but the ICs could enter hibernate mode as quickly as 2.812s.

$$2.812\text{s} \times 2^{(\text{HibEnterTime})} < \text{Hibernate Mode Entry Time} < 2.812\text{s} \times 2^{(\text{HibEnterTime} + 1)}$$

**EnHib:** Enable Hibernate Mode. When set to 1, the ICs enter hibernate mode if conditions are met. When set to 0, the ICs always remain in active mode of operation.

### CommStat Register (061h)

Register Type: Special

Nonvolatile Backup: None

The CommStat register tracks the progress and error state of any command sent to the Command register. Figure 48 shows the register format.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

**NVBusy:** This read only bit tracks if nonvolatile memory is busy or idle. NVBusy defaults to 0 after reset indicating nonvolatile memory is idle. This bit is set after a nonvolatile related command is sent to the command register and clears automatically after the operation completes.

**NVError:** This bit indicates the results of the previous SHA-256 or nonvolatile memory-related command sent to the command register. This bit is set if there was an error executing the command. Once set, the bit must be cleared by software to detect the next error.

## Cell Balancing (MAX17205/MAX17215 Only)

To maintain equal charge on all cells inside a 2S or 3S pack, the MAX17205/MAX17215 implements internal 10Ω cell-balancing MOSFETs. While charging, if the ICs detect that the voltage of a cell or cells is higher than the average voltage of the cell pack as determined by PackCfg.BALCFG setting, the ICs enable an internal shunt FET to discharge current from the corresponding cell. The small difference in charging current will balance the charge level of all cells in the pack over time.

### Cell Balancing Window of Operation

When enabled, cell balancing occurs only during a window near the end of a charge cycle when certain conditions are met. First, the voltage fuelgauge state of charge (VFSOC) register value must be larger than the FullSOCThr register value indicating the pack is nearly full. Second, the AvgCurrent register value must be above 0xFFF8 but less than 4 times the IChgTerm register value indicating the charge cycle is nearing completion. Cell balancing continues as long as VFSOC remains above FullSOCThr. Figure 49 shows the window of when cell balancing may occur.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X | X | X | X | X | X | NVError | NVBusy | X |

*Figure 48. CommStat (061h) Format*

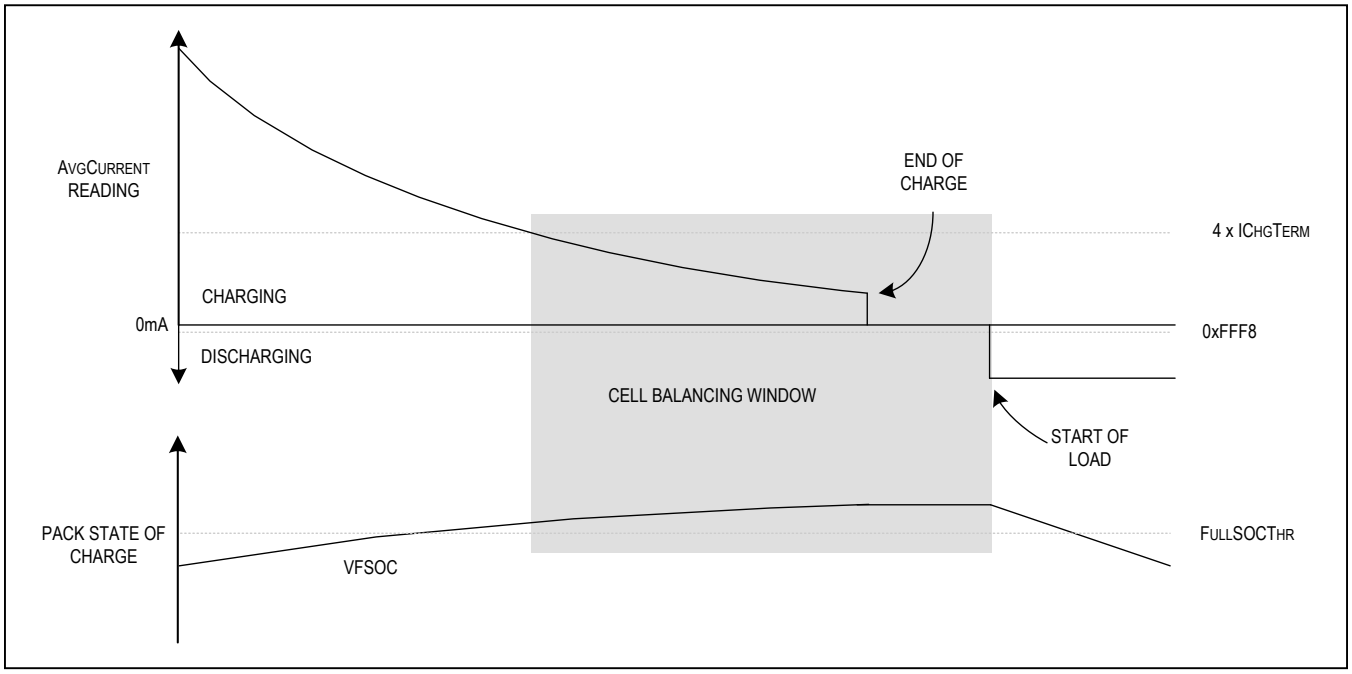



*Figure 49. Cell Balancing Window of Operation*

### Cell Balancing Order and Thresholds

The IC balances only one cell at a time, in order, starting with the highest voltage cell in the pack. As soon as the cell balancing window is entered the maximum and minimum average cell voltages are calculated. If the difference from max to min is more than the threshold defined by the PackCfg.BALCFG setting, the corresponding internal balancing switch is enabled to reduce charging current flow through the highest cell. Table 9 shows all balancing threshold levels determined by BALCFG. If enabled, the recommended balancing threshold is 011b or 10.0mV.

### Cell Balancing Circuits

Figure 50 shows the equivalent balancing circuits for 2S and 3S cell packs. Internal cell-balancing MOSFETs allow current to be drawn from an individual cell in the pack during charge. To limit current during cell balancing, an external resistor must be added in series with the CELL1 and CELL2 pins. If these resistors are not installed, power in excess of the IC package maximum current rating could be drawn leading to failure.

## Table 9. Cell Balancing Thresholds

| BALCFG VALUE | AVGCELL(1:4)–VCELL(1:4) (mV) | BALCFG VALUE | AVGCELL(1:4) - VCELL(1:4) (mV) |
|---|---|---|---|
| 000 | Balancing disabled | 100 | 20.0 |
| 001 | 2.5 | 101 | 40.0 |
| 010 | 5.0 | 110 | 80.0 |
| 011 | 10.0 | 111 | 160.0 |

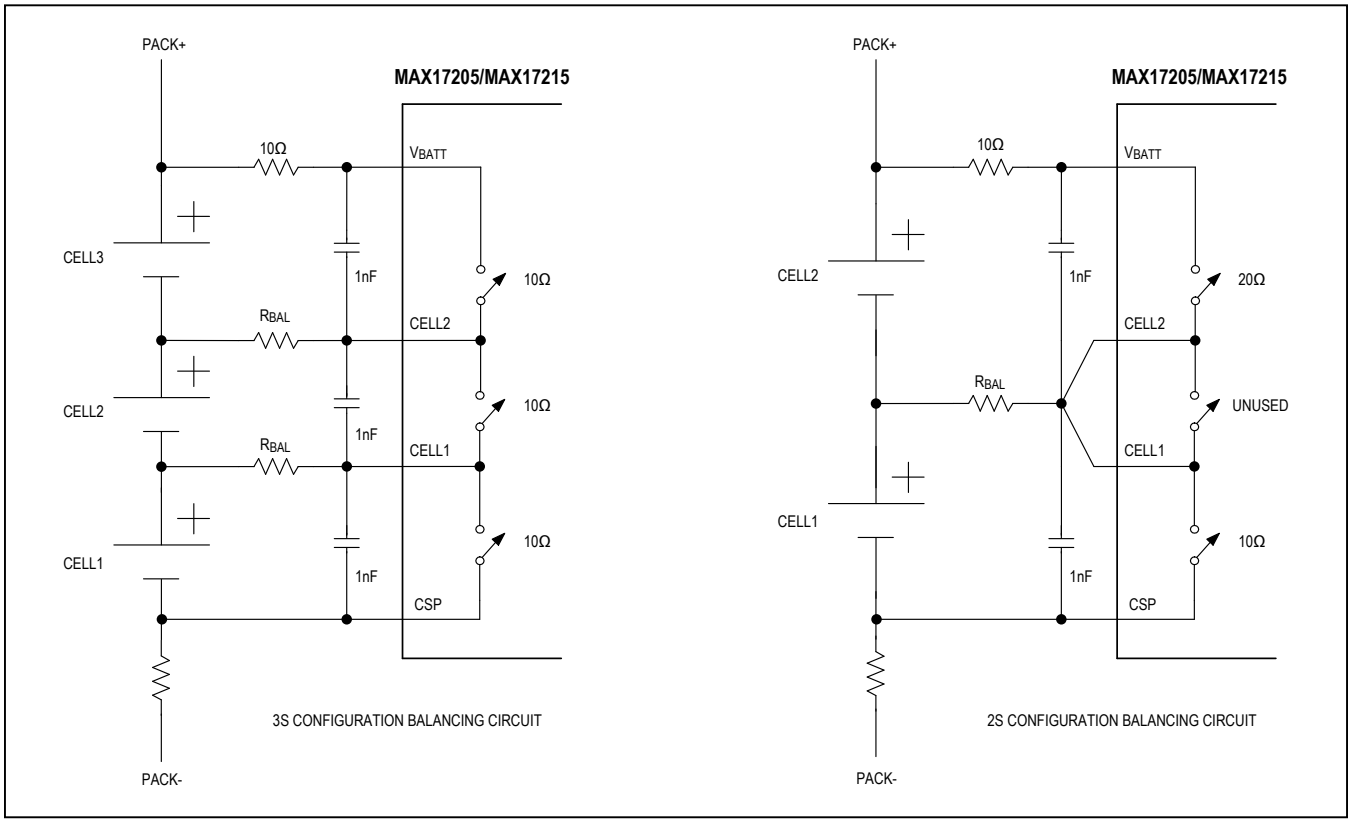



*Figure 50. 2S and 3S Balancing Circuits*

### Cell Balancing Current

External series resistors on the CELL1 and CELL2 pins are required to limit current flow when balancing. The value of these resistors should be selected to prevent exceeding the maximum rated current for these pins as listed in the *Electrical Characteristics* table. The balancing currents will not be the same for each cell. They can be calculated as follows. Remember to size these resistors to handle the power dissipated by balancing.

Bottom Cell: $IBAL_{MAX} = V_{CELL-MAX}/(R_{BAL}+R_{Switch})$

Middle Cell: $IBAL_{MAX} = V_{CELL-MAX}/(R_{BAL}+R_{Switch})$

Top Cell: $IBAL_{MAX} = V_{CELL-MAX}/(R_{BAL}+R_{SWITCH}+10\Omega)$

where:

$R_{SWITCH}$ is 10Ω (typ)

$V_{CELL-MAX}$ is the maximum cell voltage during charging

$R_{BAL}$ is the external series resistor to limit current

Note that a minimum resistance of $R_{BAL}$ should be kept above 75Ω to prevent the balancing current from exceeding the absolute maximum current ratings for the $V_{BATT}$, CELL1, CELL2, and CSP pins.

### Cell Balancing Duty Cycle

The ICs temporary interrupt cell balancing to prevent interference with voltage and current measurements. Balancing is disabled 45ms minimum prior to making a measurement to allow for settling of the external filter on the pin. This pause occurs once every task loop of the processor and has minimal impact on the average balancing current as shown in Figure 51.

## Analog Measurements

The MAX1720x/MAX1721x monitors cell pack voltage, cell pack current, cell pack temperature, and the voltage of each cell individually. This information is provided to the fuel guage algorithm to predict cell capacity and also made available to the user. Note that ADC related register information is not maintained while the IC is in shutdown mode. The following register information is invalid until the first measurement cycle after the IC returns to active mode of operation.

### Voltage Measurement

The MAX1720x/MAX1721x can monitor each cell in the cell stack independently using the CELL1, CELL2, and $V_{BATT}$ pins. The voltage of the entire stack in 2S to 4S configurations is also monitored directly on the $V_{BATT}$ pin. For larger cell stacks, the CELLx pin monitors an resistor divider circuit. The cell balancing switches are managed by firmware to ensure that balancing is disabled during the sample window. Averages of individual cell voltages are also tracked as well as the minimum and maximum voltages seen by the fuel gauge. See the individual voltage register descriptions for details.

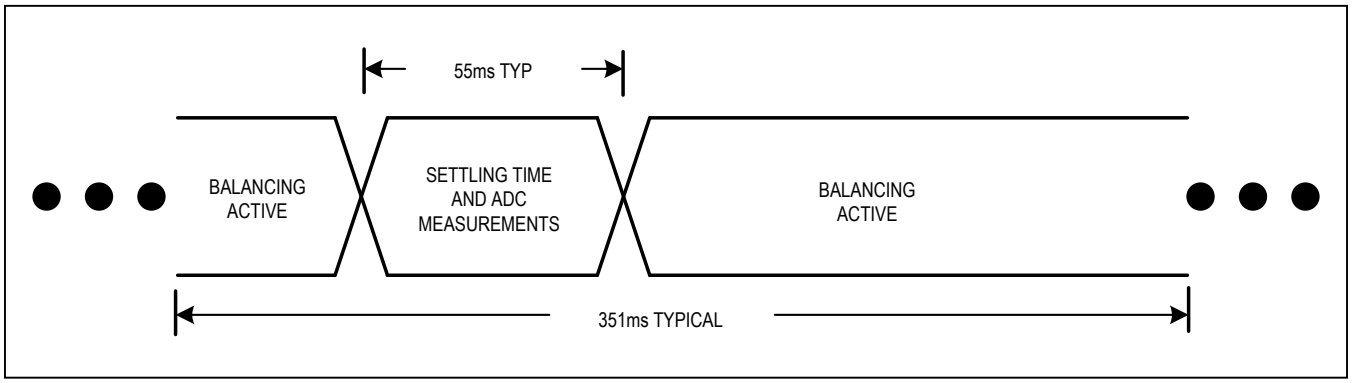



*Figure 51. Cell Balancing Duty Cycle*

### Voltage Measurement Timing

All analog measurements made by the IC are done so through a single ADC with many input channels. ADC measurement order and firmware post processing determine when a valid reading becomes available to the user. In addition, not all channels are measured each time through the firmware task loop. Selection options for enabled channels creates a large number of possible timing options. Table 10 shows the timing for all voltage measurements made by the IC for typical pack configurations. Table values assume the corresponding voltage measurement channel has been enabled through configuration. All times in this table are considered typical.

### VCell Register (009h)

Register Type: Voltage

Nonvolatile Backup: None

Each update cycle, the lowest reading from all cell voltage measurements is placed in the VCell register. VCell is used as the voltage input to the fuel gauge algorithm.

### AvgVCell Register (019h)

Register Type: Voltage

Nonvolatile Backup: None

The AvgVCell register reports an average of the VCell register readings. The time period for averaging is configurable from a 12-second to 24-minute time period. See the FilterCfg register description for details on setting the time filter. The first VCell register reading after power-up or exiting shutdown mode sets the starting point of the AvgVCell register. Note that when a cell relaxation event is detected, the averaging period changes to the period defined by the RelaxCfg.dt setting. The register reverts back to its normal averaging period when a charge or discharge current is detected.

## Table 10. Voltage Measurement Timing

| APPLICATION | nPackCfg SETTING | REGISTER | FIRST UPDATE AFTER RESET (ms)* | UPDATE RATE IN ACTIVE MODE (ms)** | UPDATE RATE IN HIBERNATE MODE (s)*** |
|---|---|---|---|---|---|
| **VOLTAGES** | | | | | |
| 1S Cell Pack | 0x1C01 | Cell1, VCell | 150 | 351 | 5.625 |
| | | AvgCell1 | 700 | | |
| | | AvgVCell | 450 | | |
| 2S–4S Cell Pack w/o Channel Measurements | 0x3A0N**** | Batt, VCell | 150 | 351 | 5.625 |
| | | AvgVCell | 450 | | |
| High-Cell Count Pack | 0x390N**** | CellX, VCell | 150 | 351 | 5.625 |
| | | AvgVCell | 450 | | |
| 2S Cell Pack with Channel Measurements | 0x3C62 | Cell1, Cell2, VCell | 200 | 702 | 11.25 |
| | | AvgCell1, AvgCell2 | 750 | | |
| | | AvgVCell | 450 | 351 | 5.625 |
| 3S Cell Pack with Channel Measurements | 0x3C63 | Cell1, Cell2, Cell3, VCell | 250 | 1053 | 16.875 |
| | | AvgCell1, AvgCell2, AvgCell3 | 800 | | |
| | | AvgVCell | 450 | 351 | 5.625 |

**AvgCell1, AvgCell2, AvgCell3, and AvgVCell registers are initialized using a single reading instead of an average.*
***Not all registers update at the same time. Updates are staggered to one channel per task period. Update order is CellX, Cell1, Cell2, Cell3, then Batt.*
****Hibernate mode update times assume the default HibCfg.HibScalar (0xB, 5.625s) setting of 16 task periods.*
*****N indicates number of cells in series.*

### MaxMinVolt Register (01Bh)

Register Type: Special

Nonvolatile Backup: saves to nMaxMinVolt (1ACh) if nNVCfg2.enMMV is set (does not restore from nonvolatile).

Initial Value: 0x00FF

The MaxMinVolt register maintains the maximum and minimum of VCell register values since device reset. Each time the voltage registers update, they are compared against these values. If the new reading is larger than the maximum or less than the minimum, the corresponding value is replaced with the new reading. At power-up, the maximum voltage value is set to 00h (the minimum) and the minimum voltage value is set to FFh (the maximum). Therefore, both values are changed to the voltage register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x00FF. The maximum and minimum voltages are each stored as 8-bit values with a 20mV resolution. Figure 52 shows the register format.

**MaxVCell:** Maximum VCell register reading (20mV resolution)
**MinVCell:** Minimum VCell register reading (20mV resolution)

MaxMinVolt is not cumulative across the entire battery lifetime. After each periodic nonvolatile-memory save, MaxMinVolt resets to 0x00FF to find the next max/min volt across the next segment of battery life. This behavior helps provide a useful log across the battery lifetime where each log segment shows the maximum and minimum voltage experienced across only that segment.

### Cell1 (0D8h)/Cell2 (0D7h)/Cell3 (0D6)/Cell4 (0D5h) Registers

Register Type: Voltage

Nonvolatile Backup: None

These registers maintain the measured or calculated voltage of each cell in the stack. If PackCfg.ChEn = 1 the ICs monitor each cell ($V_{BATT}$-$V_{CELL2}$, $V_{CELL2}$ - $V_{CELL1}$, and $V_{CELL1}$ - $V_{CSP}$) and reports results in the Cell1, Cell2, and Cell3 registers. The measurements for the lowest cell in the stack, second cell, and third cell are placed in Cell1, Cell2, and Cell3 registers respectively. If PackCfg.ChEn = 0, PackCfg.CxEn = 0, and PackCfg.BtEn = 1 then each register up to Cell4 is populated with Batt/PackCfg.NCELLS. If PackCfg.ChEn = 0, PackCfg.BtEn = 0, and PackCfg.CxEn = 1 then each register up to Cell4 is populated with the Cellx reading.

### AvgCell1 (0D4h)/AvgCell2 (0D3h)/AvgCell3 (0D2h)/AvgCell4 (0D1h) Registers

Register Type: Voltage

Nonvolatile Backup: None

The AvgCell1, AvgCell2, AvgCell3, and AvgCell4 registers report an 8-sample filtered average of the corresponding Cell1, Cell2, Cell3, and Cell4 register readings, respectively.

### CellX Register (0D9h)

Register Type: Voltage

Nonvolatile Backup: None

The CellX register reports two times the voltage measured on the CELLx pin ($V_{CELLx}$ - $V_{CSP}$). The external resistor divider on the CELLx pin should always divide to 2/5 of the voltage of a single cell.

### Batt Register (0DAh)

Register Type: Special

Nonvolatile Backup: None

The Batt register reports the voltage of the entire cell stack in 2S to 4S configuration. The register has a range 0 to 81.92V but is limited by the maximum voltage of the VBATT pin. It has an LSb resolution of 1.25mV.

If PackCfg.ChEn = 1, Batt is calculated by Cell1 + Cell2 + Cell3.

Otherwise, if PackCfg.CxEn = 1, Batt is calculated by Cellx x PackCfg.NCELLS, giving a maximum output of 76.8V when using 15-cell configuration.

Otherwise, if PackCfg.BtEn = 1, Batt is a direct measurement of the $V_{BATT}$ pin with up to 20.48V range.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MaxVCell | | | | | | | | MinVCell | | | | | | | |

*Figure 52. MaxMinVolt (01Bh)/nMaxMinVolt (1ACh) Format*

### Current Measurement

The MAX1720x/MAX1721x monitors the current flow through the cell pack by measuring the voltage between the CSN and CSP pins over a ±51.2mV range. In active mode, updates occur in intervals of 351.5ms. In hibernate mode the update interval is set by the HibCfg register. The cell balancing switches are managed by firmware to ensure that balancing is disabled during the sample window. All ICs are calibrated for current-measurement accuracy at the factory. However, if the application requires, Current register readings can be adjusted by changing the COff and CGain registers using nCGain.

If the application uses a sense resistor with a large temperature coefficient such as a copper metal board trace, current readings can be adjusted based on the temperature measured by the ICs. The CGTempCo register stores a percentage per °C value that is applied to current readings if the nNVCfg2.enMet bit is set. If nNVCfg1.enMtl = 0, the default temperature coefficient of copper is used for temperature adjustments. If enMtl = 1, the CGTempCo register value is used for temperature adjustments.

Additionally, the IC maintains a record of the minimum and maximum current measured by the ICs and an average current over a time period defined by the host. Contents of the Current and AvgCurrent registers are indeterminate for the first conversion cycle time period after IC power-up.

See *Appendix B: Layout Guidelines* for the recommended board layout to minimize current sense error.

### Current Measurement Timing

Current measurements are always enabled regardless of nPackCfg settings. Table 11 shows the timing for current measurements made by the ICs. All times in this table are considered typical.

### Current Register (00Ah)

Register Type: Current

Nonvolatile Backup: None

The IC measures the voltage between the CSP and CSN pins and the result is stored as a two's complement value in the Current register. Voltages outside the minimum and maximum register values are reported as the minimum or maximum value. The register value should be divided by the sense resistance to convert to amps. The value of the sense resistor determines the resolution and the full-scale range of the current readings. Table 12 shows range and resolution values for typical sense resistances.

### AvgCurrent Register (00Bh)

Register Type: Current

Nonvolatile Backup: None

The AvgCurrent register reports an average of Current register readings over a configurable 0.7-second to 6.4-hour time period. See the FilterCfg register description for details on setting the time filter. The first Current register reading after returning to active mode sets the starting point of the AvgCurrent filter.

## Table 11. Current Measurement Timing

| APPLICATION | nPackCfg SETTING | REGISTER | FIRST UPDATE AFTER RESET (ms)* | UPDATE RATE IN ACTIVE MODE (ms) | UPDATE RATE IN HIBERNATE MODE (s)** |
|---|---|---|---|---|---|
| **CURRENTS** | | | | | |
| Any | Any | CurrentCurrent | 150 | 351 | 5.625 |
| | | AvgCurrent | 150 | 351 | 5.625 |

**AvgCurrent register is initialized using a single reading instead of an average.*
***Hibernate mode update times assume the default HibCfg.HibScalar (0xB, 5.625s) setting of 16 task periods.*

## Table 12. Current Measurement Range and Resolution vs. Sense Resistor Value

| SENSE RESISTOR (CGAIN=0X0400) (Ω) | CURRENT REGISTER RESOLUTION (µA) | CURRENT REGISTER RANGE (A) |
|---|---|---|
| 0.005 | 312.5 | ±10.24 |
| 0.010 | 156.25 | ±5.12 |
| 0.020 | 78.125 | ±2.56 |

### MaxMinCurr Register (01Ch)

Register Type: Special

Nonvolatile Backup: Periodically saves to nMaxMinCurr (1ABh) if nNVCfg2.enMMC is set, but does not restore from nonvolatile memory.

Alternate Initial Value: 0x807F

The MaxMinCurr register maintains the maximum and minimum Current register values since the last IC reset or until cleared by the host software. Each time the Current register updates, it is compared against these values. If the reading is larger than the maximum or less than the minimum, the corresponding value is replaced with the new reading. At power-up, the maximum current value is set to 80h (the minimum) and the minimum current value is set to 7Fh (the maximum). Therefore, both values are changed to the Current register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x807F. The maximum and minimum voltages are each stored as two's complement 8-bit values with 0.4mV/$R_{SENSE}$ resolution. Figure 53 shows the register format.

**MaxCurrent:** Maximum Current register reading (0.40mV/$R_{SENSE}$ resolution)
**MinCurrent:** Minimum Current register reading (0.40mV/$R_{SENSE}$ resolution)

MaxMinCurr is not cumulative across the entire battery lifetime. After each periodic nonvolatile-memory save, MaxMinCurr resets to 0x807F to find the next maximum and minimum current across the next segment of battery life. This behavior helps provide a useful log across the battery lifetime in which each log segment shows the maximum and minimum current experienced across only that segment.

### nCGain Register (1C8h)

Register Type: Special

Nonvolatile Restore: CGain (02Eh) and COff (02Fh) if nNVCfg0.enCG is set.

Alternate Initial Value: CGain = 0x0400 and COff = 0x0000

The CGain and COff registers adjust the gain and offset of the current measurement result. The current measurement ADC is factory trimmed to data sheet accuracy without the need for the user to make further adjustments. The default power-up settings for CGain and COff apply no adjustments to the Current register reading. For specific application requirements, the CGain and COff registers can be used to adjust readings as follows:

Current Register = Current ADC Reading × (CGain Register/0400h) + COff Register

The nonvolatile backup of CGain and COff is combined into a single register formatted as shown in Figure 54.

**nCOff:** Value to restore into the COff register giving a range of -32 to +31 LSbs. COff signs extend if the value is negative.

**nCGain:** Value to restore into bits D11 to D2 of the CGain register as in the following equation. nCGAIN can be a negative value. CGain signs extend to maintain polarity.

CGain = (nCGain AND 0xFFC0) >> 4

### CGTempCo Register (16Dh)

Register Type: Special

Alternate Initial Value: 0x20C8

If nNVCfg1.enCrv = 0 and nNVCfg2.enMet = 1 then use CGTempCo to adjust current measurements for temperature. CGTempCo has a range of 0% to 3.1224% per °C with a step size of 3.1224/0x10000 percent per °C. If the nNVCfg1.enMtl bit is clear, CGTempCo defaults to a value of 0x20C8 or 0.4% per °C, which is the approximate temperature coefficient of a copper trace. If the nNVCfg1.enMtl bit is set, CGTempCo restores from nTCurve (1C9h) after the ICs reset to allow a custom-sense resistor temperature coefficient to be used. Note that nNVCfg1.enCrv and nNVCfg2.enMet cannot be enabled simultaneously.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MaxCurrent | | | | | | | | MinCurrent | | | | | | | |

*Figure 53. MaxMinCurr (01Ch)/nMaxMinCurr (1ABh) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nCGain | | | | | | | | | | nCOff | | | | | |

*Figure 54. nCGain Register (1C8h) Format*

## Copper Trace Current Sensing

The MAX1720x/1x has the ability to measure current using a copper board trace instead of a traditional sense resistor. The main difference being the ability to adjust to the change in sense resistance over temperature. To enable copper trace current sensing, set the following configuration bits: nNVCfg1.enCrv=0 and nNVCfg2.enMet=1. The IC's default temperature adjustment is 0.4% per °C, but can be adjusted using the nTCurve register if nNVCfg1.enMtl=1. Note that copper trace current sensing cannot be enabled at the same time as thermistor curve adjustment. For 1-ounce copper, a length to width ratio of 6:1 will create a 0.0035Ω sense resistor which is suitable for most applications. Table 13 summarizes the IC setting for copper trace sensing.

### Temperature Measurement

The IC can be configured to measure its own internal die temperature and up to two external NTC thermistors. In addition the IC can be configured to use any single input or an average of both thermistors as the temperature input to the fuel gauge calculations. See the nPackCfg register for details.

Thermistor conversions are initiated by connecting the THRM and REG3 pins internally. This enables the active pullup to the external voltage-divider network. After the pullup is enabled, the IC waits for a settling period of $t_{PRE}$ prior to making measurements on the AIN1 or AIN2 pins. Measurement results are compared to the voltage of the THRM pin and converted to a ratiometric value from 0 to 100%. The active pullup is disabled when temperature measurements are complete. This feature limits the time the external resistor-divider network is active and lowers the total amount of energy used by the system.

The ratiometric results are converted to temperature using the temperature gain (TGain), temperature offset (TOff), and temperature curve (nTCurve) register values each time the AIN1 and AIN2 pins are measured. Internal die temperature measurements are factory calibrated and are not affected by TGain, TOff, nTCurve register settings. Additionally the IC maintains a record of the minimum and maximum temperature measured, and an average temperature over a time period defined by the host.

### Temperature Measurement Timing

Temperature measurement channels are individually enabled using the nPackCfg register. ADC measurement order and firmware post processing determine when a valid reading will become available to the user. In addition, not all channels are measured each time through the firmware task loop. Selection options for enabled channels creates a large number of possible timing options. Table 14 shows the timing for all temperature measurements made by the IC for some typical pack configurations. All times in this table are considered typical.

## Table 13. Copper Trace Sensing

| PARAMETER | SETTING | RESULT |
|---|---|---|
| nNVCfg1.enCRV | 0 | Thermistor curve compensation disabled |
| nNVCfg1.enMet | 1 | Sense resistor temperature compensation enabled |
| nNVCfg2.enMlt | 0 | Sense resistor temperature compensation set to default of 0.4% per °C (typical copper). |
| nRense | 0x012C | Sense resistor indicator to host software set to 0.0035Ω as demonstrated on the evaluation kit hardware. |
| $R_{SENSE}$ Size | 6:1 | A 6:1 length to width ratio of 1oz copper gives a resistance of 0.0035Ω. |

## Table 14. Temperature Measurement Timing

| APPLICATION | nPackCfg SETTING | REGISTER | FIRST UPDATE AFTER RESET (ms)* | UPDATE RATE IN ACTIVE MODE** (ms) | UPDATE RATE IN HIBERNATE MODE*** (s) |
|---|---|---|---|---|---|
| **TEMPERATURES** | | | | | |
| Die Temperature Only | nPackCfg.TdEn = 1<br>nPackCfg.A1En = 0<br>nPackCfg.A2En = 0<br>nPackCfg.FGT = 0 | Temp, IntTemp, AvgIntTemp | 550 | 703 | 5.625 |
| | | AvgTA | | 351 | |
| Single Thermistor Only | nPackCfg.TdEn = 0<br>nPackCfg.A1En = 1<br>nPackCfg.A2En = 0<br>nPackCfg.FGT = 0 | Temp, Temp1, AvgTemp1 | 550 | 703 | 5.625 |
| | | AvgTA | | 351 | |
| Die Temperature and Single Thermistor, Fuel Gauge Thermistor 1 | nPackCfg.TdEn = 1<br>nPackCfg.A1En = 1<br>nPackCfg.A2En = 0<br>nPackCfg.FGT = 0 | IntTemp, Temp1, Temp, IntTemp, Temp1, Temp, AvgIntTemp, AvgTemp1 | 550 | 1406 | 11.25 |
| | | AvgTA | | 351 | 5.625 |
| All Channels Active, Fuel Gauge Thermistor 1 | nPackCfg.TdEn = 1<br>nPackCfg.A1En = 1<br>nPackCfg.A2En = 1<br>nPackCfg.FGT = 1 | IntTemp, Temp1, Temp2, Temp, AvgIntTemp, AvgTemp1,AvgTemp2 | 550 | 2109 | 16.875 |
| | | AvgTA | | 351 | 5.625 |
| All Channels Active, Average of Both Thermistors for Fuel Gauging | nPackCfg.TdEn = 1<br>nPackCfg.A1En = 1<br>nPackCfg.A2En = 1<br>nPackCfg.FGT = 0 | Temp | 550 | Twice in 2109**** | Twice in 16.875**** |
| | | IntTemp, Temp1, Temp2, AvgIntTemp, AvgTemp1,AvgTemp2 | | 2109 | 16.875 |
| | | AvgTA | | 351 | 5.625 |

**All enabled registers are initialized using the first temperature reading the IC makes even if it is from a different temperature input.*
***Not all registers update at the same time. Updates are staggered to one channel per task period. Update order is IntTemp, Temp1, Temp2.*
****Hibernate mode update times assume the default HibCfg.HibScalar setting of 16 task periods.*
*****Update interval in active mode alternates between 703ms and 1406ms. Update interval in hibernate mode alternates between 5.625s and 11.25s*

### Temp Register (008h)

Register Type: Temperature

Nonvolatile Backup: None

The Temp register value is selected from Temp1, Temp2, IntTemp, or alternating between Temp1 and Temp2 registers as determined by the nPackCfg register setting. The Temp register is the input to the fuel gauge algorithm. Contents of Temp are indeterminate for the first conversion cycle time period after IC power-up.

### AvgTA Register (016h)

Register Type: Temperature

Nonvolatile Backup: None

The AvgTA register reports an average of the readings from the Temp register. Averaging period is configurable from 6 minutes up to 12 hours as set by the FilterCfg register. The first Temp register reading after returning to active mode sets the starting point of the averaging filters.

### MaxMinTemp Register (01Ah)

Register Type: Special

Nonvolatile Backup: periodically saves to nMaxMinTemp (1ADh) if nNVCfg2.enMMT is set, but does not restore from nonvolatile memory.

Alternate Initial Value: 0x807F

The MaxMinTemp register maintains the maximum and minimum Temp register (008h) values since the last fuel-gauge reset or until cleared by host software. Each time the Temp register updates, it is compared against these values. If the reading is larger than the maximum or less than the minimum, the corresponding values are replaced with the new reading. At power-up, the maximum value is set to 80h (minimum) and the minimum value is set to 7Fh (maximum). Therefore, both values are changed to the Temp register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x807F. The maximum and minimum temperatures are each stored as two's complement 8-bit values with 1°C resolution. Figure 55 shows the format of the register.

**MaxTemperature:** Maximum Temp register reading (1ºC resolution)

**MinTemperature:** Minimum Temp register reading (1ºC resolution)

MaxMinTemp is not cumulative across the entire battery lifetime. After each periodic nonvolatile memory save, MaxMinTemp resets to 0x807F to find the next maximum and minimum temperatures across the next segment of battery life. This behavior helps provide a useful log across the battery lifetime where each log segment shows the maximum and minimum temperature experienced across only that segment.

### nTCurve Register (1C9h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register

If nNVCfg1.enCrv = 1 and nNVCfg2.enMet = 0 then nTCurve applies thermistor measurement curvature correction to allow thermistor measurements to be accurate over a wider temperature range. A ±3°C accuracy can be achieved over a -40°C to 85°C operating range. See Table 15 for recommended nTCurve values. If nNVCfg1.enCrv = 0 and nNVCfg2.enMet = 0 this location can be used to general purpose data storage.

### nTGain (1CAh) Register/nTOff (1CBh) Register

Register Type: Special

Nonvolatile Restore: TGain (02Ch) and TOff (02Dh) if nNVCfg1.enTGO is set

Alternate Initial Value: TGain = 0xEE56 and TOff = 0x1DA4 (associated with nTCurve = 0x0025)

External NTC Thermistors generate a temperature related voltage to be measured by the AIN1 and AIN2 inputs. See the typical operating circuits for examples of thermistor circuits. The nTGain, nTOff, and nTCurve registers are used to calculate temperature from the measurement of the AIN1 and AIN2 pins with an accuracy of +/-3°C over a range of -40°C to 85°C. Table 15 lists the recommended nTGain, nTOff, and nTCurve register values for common NTC thermistors.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MaxTemperature | | | | | | | | MinTemperature | | | | | | | |

*Figure 55. MaxMinTemp (01Ah)/nMaxMinTemp (1ADh) Format*

## Table 15. Register Settings for Common Thermistor Types

| THERMISTOR | R25C (KΩ) | BETA | RECOMMENDED NTGAIN | RECOMMENDED NTOFF | RECOMMENDED NTCURVE |
|---|---|---|---|---|---|
| Murata NCP15XH103F03RC | 10 | 3435 | 0xEE56 | 0x1DA4 | 0x0025 |
| Fenwal 197-103LAG-A01 | 10 | 3974 | 0xF49A | 0x16A1 | 0x0064 |
| TDK Type F | 10 | 4550 | 0xF284 | 0x18E8 | 0x0035 |

**Temp1 (134h)/Temp2 (13Bh)/IntTemp (135h) Registers**

Register Type: Temperature

Nonvolatile Backup: None

If enabled in the nPackCfg register, the Temp1, Temp2, and IntTemp registers contain the temperature readings from the AIN1 thermistor, AIN2 thermistor, and internal die temperature respectively. Contents of each register are indeterminate for the first conversion cycle time period after IC power-up. These registers display temperature in degrees Celsius starting at absolute zero, -273ºC or 0ºK with an LSb of 0.1ºC.

**AvgTemp1 (137h)/AvgTemp2 (139h)/AvgIntTemp (138h) Registers**

Register Type: Temperature

Nonvolatile Backup: None

The AvgTemp1, AvgTemp2, and AvgIntTemp registers report a 4-sample filtered average of the corresponding Temp1, Temp2, and IntTemp registers respectively. The first of each temperature register reading after startup sets the starting point of the averaging filters. These registers display temperature in degrees Celsius starting at absolute zero, -273ºC or 0ºK with an LSb of 0.1ºC.

**AIN0 Register (027h)**

Register Type: Special

Nonvolatile Backup: None

External temperature measurements on the AIN1 and AIN2 pins are compared to the THRM pin voltage. The IC stores the result as a ratio-metric value from 0% to 100% in the AIN0 register with an LSB of 0.0122%. The nTGain, nTOff, and nTCurve register values are then applied to this ratio-metric reading to convert the result to temperature. The value stored in the AIN0 register alternates between measurements made on the AIN1 and AIN2 pins during operation.

## At-Rate Functionality

The AtRate function allows host software to see theoretical remaining time or capacity for any given load current. AtRate can be used for power management by limiting system loads depending on present conditions of the cell pack. Whenever the AtRate register is programmed to a negative value indicating a hypothetical discharge current the AtQResidual, AtTTE, AtAvSOC, and AtAvCap registers display theoretical residual capacity, time to empty, state of charge, and available capacity respectively. Host software should wait two full task periods (703ms minimum in active mode) after writing the AtRate register before reading any of the result registers.

**AtRate Register (004h)**

Register Type: Current

Nonvolatile Backup: None

Host software should write the AtRate register with a negative two's-complement 16-bit value of a theoretical load current prior to reading any of the at-rate output registers.

**AtQResidual Register (0DCh)**

Register Type: Capacity

Nonvolatile Backup: None

The AtQResidual register displays the residual charge held by the cell at the theoretical load current level entered into the AtRate register.

**AtTTE Register (0DDh)**

Register Type: Time

Nonvolatile Backup: None

The AtTTE register can be used to estimate time to empty for any theoretical current load entered into the AtRate register. The AtTTE register displays the estimated time to empty for the application by dividing AtAvCap by the AtRate register value.

**AtAvSOC Register (0CEh)**

Register Type: Percentage

Nonvolatile Backup: None

The AtAvSOC register holds the theoretical state of charge of the cell based on the theoretical current load of the AtRate register. The register value is stored as a percentage with a resolution of 0.0039% per LSB. If a 1% resolution state-of-charge value is desired, the host can read only the upper byte of the register instead.

**AtAvCap Register (0DFh)**

Register Type: Capacity

Nonvolatile Backup: None

The AtAvCap register holds the estimated remaining capacity of the cell based on the theoretical load current value of the AtRate register. The value is stored in terms of µVh and must be divided by the application sense-resistor value to determine the remaining capacity in mAh.

## Overcurrent Comparators

The MAX1720x/1x contains two programmable fast overcurrent comparators called OD and SC that allow spikes in system current to be detected. Both comparators have programmable threshold levels and programmable debounced delays. They can be enabled independently of each other and can be enabled to give over-current indication on the ALRT1 pin. See Figure 56.

The OD comparator threshold can be programmed from 0mV to -155mV with 5mV resolution (0 to -15.5A with 0.5A resolution using 10mΩ sense resistor), and has a programmable delay from 0 to 14.6ms with 0.97ms resolution. The SC comparator threshold can be programmed from 0mV to -310mV with 10mV resolution (0 to -31A with 1A resolution using 10mΩ sense resistor), and has a programmable delay from 0 to 915µs with a 61µs resolution. Either comparator can be configured for zero delay (2µs typical propagation) that results in a nonlatching behavior on the ALRT1 pin.

The ODSCTh register sets the threshold levels where each comparator will trip. The ODSCCfg register enables each comparator and sets their debounce delays. The ODSCCfg register also maintains indicator flags of which comparator has been tripped. These register settings are maintained in nonvolatile memory if the nNVCfg1. enODSC bit is set.

### nODSCTh Register (18Eh)

Register Type: Special

Nonvolatile Restore: ODSCTh (0F2h) if nNVCfg1.enODSC is set

Alternate Initial Value: 0x0000

The nODSCTh register sets the current thresholds for each alert. Bits D4 through D0 set the ODTH (over-discharge threshold), and D12 to D8 set the SCTH (short-circuit threshold). The format of the registers is shown in Figure 57.

**SCTH:** Short-Circuit Threshold Setting. Sets the short circuit threshold to a value between 0mV and -155mV with a step size of -5mV. The SCTH bits are stored as a 2's-complement value with 0x1F = 0mV and 0x00 = -155mV.

**ODTH:** Over-discharge Threshold Setting. Sets the over-discharge threshold to a value between 0mV and -77.5mV with a step size of -2.5mV. The ODTH bits are stored as a 2's-complement value with 0x1F = 0mV and 0x00 = -77.5mV.

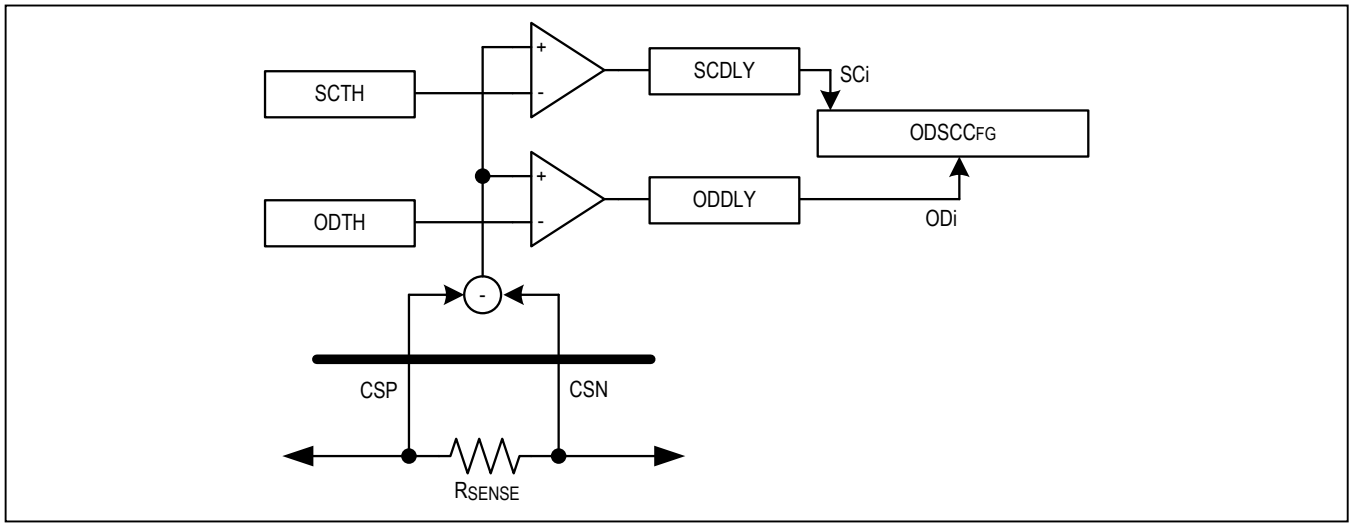



*Figure 56. Overcurrent Comparator Diagram*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | SCTH | | | | | X | X | X | ODTH | | | | |

*X: Don't care.*

*Figure 57. ODSCTh Register (0F2h) and nODSCTh Register (18Eh) Formats*

### nODSCCfg Register (18Fh)

Register Type: Special

Nonvolatile Restore: ODSCCfg (0F3h) if nNVCfg1.enODSC is set.

Alternate Initial Value: 0x0000

The nODSCCfg register configures enabling of the Over Discharge and Short Circuit alerts, as well as setting their latching and delay behavior. The alert indicators are also displayed and cleared in this register. The format of the register is shown in Figure 58.

**SCDLY:** Sets the latching and delay for the Short Circuit detection from 2µs up to 917µs. If this field set to 0, the alert will happen typically 2µs after the threshold is tripped with no latching. Any non zero setting causes the indicator to latch after a delay of 2µs + 61µs x SCDLY.

**SCen:** Short Circuit Alert Enable. Set to 1 to enable alerts when the SCTh value is exceeded.

**SCi:** Short Circuit Indicator. This bit is set to 1 to indicate the short-circuit threshold was detected. Write to zero to clear this bit if SCDLY is not 0. If SCDLY is 0, the alert clears automatically when the alert condition is removed.

ODDLY**:** Sets the latching and delay for the Over Discharge detection from 2µs up to 14.657ms. If this field is set to 0, the alert will happen typically 2µs after the threshold is tripped with no latching. Any non zero setting causes the indicator to latch after a delay of 2µs + 977µs x ODDLY.

ODen**:** Over Discharge Alert Enable. Set to 1 to enable alerts when the ODTh value is exceeded.

**ODi:** Over Discharge Indicator. This bit is set to 1 to indicate the over-discharge threshold was detected. Write to zero to clear this bit if ODDLY is not 0. If ODDLY is 0, the alert clears automatically when the alert condition is removed.

## Alert Function

The Alert Threshold registers allow interrupts to be generated by detecting a high or low voltage, current, temperature, or state of charge. Interrupts are generated on the ALRT1 pin open-drain output driver. An external pullup is required to generate a logic-high signal. Note that if the pin is configured to be logic-low when inactive, the external pullup increases current drain. The ALRTp bit in the Config register sets the polarity of the ALRT1 pin output. Alerts can be triggered by any of the following conditions:

• Battery removal—($V_{AIN1}$ > $V_{THRM}$ - $V_{DET}$) and battery removal detection enabled (Ber = 1).

• Battery insertion—($V_{AIN1}$ < $V_{THRM}$ - $V_{DET-HYS}$) and battery insertion detection enabled (Bei = 1).

• Over/undervoltage—VAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).

• Over/undertemperature—TAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).

• Over/undercurrent—IAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).

• Over/under SOC—SAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).

To prevent false interrupts, the threshold registers should be initialized before setting the Aen bit. Alerts generated by battery insertion or removal can only be reset by clearing the corresponding bit in the Status (000h) register. Alerts generated by a threshold-level violation can be configured to be cleared only by software, or cleared automatically when the threshold level is no longer violated. See the Config (01Dh) register description for details of the alert function configuration.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCi | SCen | X | X | SCDLY | | | | ODi | ODen | X | X | ODDLY | | | |

*X: Don't care.*

*Figure 58. ODSCCfg Register (0F3h) and nODSCCfg Register (18Fh) Formats*

### nVAlrtTh Register (1C0h)

Register Type: Special

Nonvolatile Restore: VAlrtTh (001h) if nNVCfg1.enAT is set.

Alternate Initial Value: 0xFF00 (disabled)

The nVAlrtTh register shown in Figure 59 sets upper and lower limits that generate an ALRT1 pin interrupt if exceeded by the VCell register value. The upper 8 bits set the maximum value and the lower 8 bits set the minimum value. Interrupt threshold limits are selectable with 20mV resolution over the full operating range of the VCell register. At power-up, the thresholds default to their maximum settings unless they are configured to be restored from nonvolatile memory instead by setting the nNVCfg1.enAT bit.

**VMAX:** Maximum voltage threshold. An alert is generated if the VCell register reading exceeds this value. This field has 20mV LSb resolution.

**VMIN:** Minimum voltage threshold. An alert is generated if the VCell register reading falls below this value. This field has 20mV LSb resolution.

### nTAlrtTh Register (1C1h)

Register Type: Special

Nonvolatile Restore: TAlrtTh (002h) if nNVCfg1.enAT is set

Alternate Initial Value: 0x7F80 (Disabled)

The nTAlrtTh register shown in Figure 60 sets upper and lower limits that generate an ALRT1 pin interrupt if exceeded by the Temp register value. The upper 8 bits set the maximum value and the lower 8 bits set the minimum value. Interrupt threshold limits are stored in 2's-complement format with 1ºC resolution over the full operating range of the Temp register. At power-up, the thresholds default to their maximum settings unless they are configured to be restored from nonvolatile memory instead by setting the nNVCfg1.enAT bit.

**TMAX:** Maximum temperature threshold. An alert is generated if the Temp register reading exceeds this value. This field is signed 2's complement format with 1°C LSb resolution.

**TMIN:** Minimum temperature threshold. An alert is generated if the Temp register reading falls below this value. This field is signed 2's complement format with 1°C LSb resolution.

### nSAlrtTh Register (1C2h)

Register Type: Special

Nonvolatile Restore: SAlrtTh (003h) if nNVCfg1.enAT is set.

Alternate Initial Value: 0xFF00 (disabled)

The nSAlrtTh register shown in Figure 61 sets upper and lower limits that generate an ALRT1 pin interrupt if exceeded by the selected RepSOC, AvSOC, MixSOC, or VFSOC register values. See the MiscCFG.SACFG setting for details.The upper 8 bits set the maximum value and the lower 8 bits set the minimum value. Interrupt threshold limits are selectable with 1% resolution over the full operating range of the selected SOC register. At power-up, the thresholds default to their maximum settings unless they are configured to be restored from nonvolatile memory instead by setting the nNVCfg1.enAT bit.

**SMAX:** Maximum state of charge threshold. An alert is generated if the selected SOC register reading exceeds this value. This field has 1% LSb resolution.

**SMIN:** Minimum state of charge threshold. An alert is generated if the selected SOC register reading falls below this value. This field has 1% LSb resolution.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VMAX | | | | | | | | VMIN | | | | | | | |

*Figure 59. VAlrtTh (001h)/nVAlrtTh (1C0h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMAX | | | | | | | | TMIN | | | | | | | |

*Figure 60. TAlrtTh (002h)/nTAlrtTh (1C1h) Format*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMAX | | | | | | | | SMIN | | | | | | | |

*Figure 61. SAlrtTh (003h)/nSAlrtTh (1C2h) Format*

### nIAlrtTh Register (1C3h)

Register Type: Special

Nonvolatile Restore: IAlrtTh (0B4h) if nNVCfg1.enAT is set.

Alternate Initial Value: 0x7F80 (Disabled)

The nIAlrtTh register shown in Figure 62 sets upper and lower limits that generate an ALRT1 pin interrupt if exceeded by the Current register value. The upper 8 bits set the maximum value and the lower 8 bits set the minimum vaue. Interrupt threshold limits are selectable with 400µV resolution over the full operating range of the Current register. At power-up, the thresholds default to their maximum settings unless they are configured to be restored from nonvolatile memory instead by setting the nNVCfg1.enAT bit.

**IMAX:** Maximum current threshold. An alert is generated if the current register reading exceeds this value. This field is signed 2's complement with 400µV LSb resolution to match the upper byte of the Current register.

**IMIN:** Maximum current threshold. An alert is generated if the current register reading falls below this value. This field is signed 2's complement with 400µV LSb resolution to match the upper byte of the Current register.

## Memory

The memory space of the MAX1720x/1x is divided into 32 pages each containing 16 registers where each register is 16 bits wide. Registers are addressed using an internal 9bit range of 000h to 1FFh. Externally registers are accessed with an 8-bit address for 2-wire communication (MAX1720x) or 16-bit address for 1-wire communication (MAX1721x). Registers are grouped by functional block. See the functional descriptions for details of each register's functionality. Certain memory blocks can be permanently locked to prevent accidental overwrite. See the *Locking Memory Blocks* section for details. Table 16 shows the full memory map of the ICs. Note that some individual user registers are located on RESERVED memory pages in Table 17. These locations can be accessed normally while the remainder of the page is considered RESERVED. Memory locations listed as RESERVED should never be written to. Data read from RESERVED locations is not defined.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IMAX | | | | | | | | IMIN | | | | | | | |

*Figure 62. IAlrtTh (0B4h)/nIAlrtTh (1C3h) Format*

## Table 16. Top-Level Memory Map

| REGISTER PAGE | LOCK | DESCRIPTION | MAX1720X | | | MAX1721X |
|---|---|---|---|---|---|---|
| | | | 2-WIRE SLAVE ADDRESS | 2-WIRE PROTOCOL | 2-WIRE EXTERNAL ADDRESS RANGE | 1-WIRE EXTERNAL ADDRESS RANGE |
| 00h | | ModelGauge m5 DATA BLOCK | 6Ch | I²C | 00h–4Fh | 0000h–004Fh |
| 01h–04h | LOCK2 | | | | | |
| 05h–0Ah | | RESERVED | — | — | — | — |
| 0Bh | LOCK2 | ModelGauge m5 DATA BLOCK (continued) | 6Ch | I²C | B0h–BFh | 00B0h–00BFh |
| 0Ch | SHA | SHA MEMORY | 6Ch | I²C | C0h–CFh | 00C0h–00CFh |
| 0Dh | LOCK2 | ModelGauge m5 DATA BLOCK (continued) | 6Ch | I²C | D0h–DFh | 00D0h–00DFh |
| 0Eh–0Fh | | RESERVED | — | — | — | — |
| 10h–17h | | SBS DATA BLOCK | 16h | SBS | 00h–7Fh | — |
| 18h–19h | LOCK3 | NONVOLATILE MEMORY | 16h | I²C | 80h–DFh | 0180h-01DFh |
| 1Ah–1Bh | LOCK1 | | | | | |
| 1Ch | LOCK4 | | | | | |
| 1Dh | LOCK5 | | | | | |
| 1Eh | | NONVOLATILE HISTORY | 16h | I²C | E0h–EFh | 01E0h-01EFh |
| 1Fh | | RESERVED | — | — | — | — |

## Table 17. Individual Registers

| REGISTER PAGE | DESCRIPTION | MAX1720X | | | MAX1721X |
|---|---|---|---|---|---|
| | | 2-WIRE SLAVE ADDRESS | 2-WIRE PROTOCOL | 2-WIRE EXTERNAL ADDRESS RANGE | 1-WIRE EXTERNAL ADDRESS RANGE |
| 060H | COMMAND REGISTER | 6CH | I²C | 60H | 0060H |
| 061h | CommStat REGISTER | 6Ch | I²C | 61h | 0061h |
| 07Fh | Lock REGISTER | 6Ch | I²C | 7Fh | 007Fh |
| 0F2h | ODSCTh REGISTER | 6Ch | I²C | F2h | 00F2h |
| 0F3h | ODSCCfg REGISTER | 6Ch | I²C | F3h | 00F3h |
| 0FBh | MODELGAUGE m5 REGISTER VFOCV | 6Ch | I²C | FBh | 00FBh |
| 0FFh | MODELGAUGE m5 REGISTER VFSOC | 6Ch | I²C | FFh | 00FFh |

## Table 18. ModelGauge m5 Register Memory Map

| WORD | 00xh | 01xh | 02xh | 03xh | 04xh | 0Bxh | 0Dxh |
|---|---|---|---|---|---|---|---|
| 0h | Status | FullCap | TTF | | | Status2 | |
| 1h | VAlrtTh | TTE | DevName | | | | AvgCell4 |
| 2h | TAlrtTh | QRTable00 | QRTable10 | QRTable20 | QRTable30 | | AvgCell3 |
| 3h | SAlrtTh | FullSocThr | FullCapNom | | | | AvgCell2 |
| 4h | AtRate | RCellRCell | | | | IAlrtTh | AvgCell1 |
| 5h | RepCap | RFast | | FullCapRep | dQAcc | | Cell4 |
| 6h | RepSOC | AvgTA | | IAvgEmpty | dPAcc | | Cell3 |
| 7h | Age | Cycles | AIN0 | | | | Cell2 |
| 8h | Temp | DesignCap | LearnCfg | RComp0 | | VShdnCfg | Cell1 |
| 9h | VCell | AvgVCell | FilterCfg | TempCo | | AgeForecast | CellX |
| Ah | Current | MaxMinTemp | RelaxCfg | VEmpty | VFRemCap | HibCfg | Batt |
| Bh | AvgCurrent | MaxMinVolt | MiscCfg | | | Config2 | |
| Ch | QResidual | MaxMinCurr | TGain | | | VRipple | AtQResidual |
| Dh | MixSOC | Config | TOff | FStat | QH | PackCfg | AtTTE |
| Eh | AvSOC | IChgTerm | CGain | Timer | | TimerH | AtAvSOC |
| Fh | MixCap | AvCap | COff | ShdnTimer | | | AtAvCap |

### ModelGauge m5 Memory Space

Registers that relate to functionality of the ModelGauge m5 fuel gauge are located on pages 00h-04h and are continued on pages 0Bh and 0Dh. See the *ModelGauge m5 Algorithm* section for details of specific register operation. These locations (other than page 00h) can be permanently locked by setting LOCK2. Unnamed register locations are reserved locations and should not be written to. See Table 18.

### Nonvolatile Memory

Certain ModelGauge m5 and device configuration values are stored in nonvolatile memory to prevent data loss if the ICs lose power. The MAX1720x/MAX1721x internally update page 1Ah values over time based on actual performance of the ModelGauge m5 algorithm. The host system does not need to access this memory space during operation. Nonvolatile data from other accessible register locations is internally mirrored into the nonvolatile memory block automatically. Note that nonvolatile memory has a limited number of writes. User accessible configuration memory is limited to 7 writes. Internal and external updates to page 1Ah as the fuel gauge algorithm learns is limited to 202 writes. Do not exceed these write limits. Table 19 shows the nonvolatile memory register map.

### Shadow RAM

Nonvolatile memory is never written to or read from directly by the communication interface. Instead, data is written to or read from shadow RAM memory located at the same address. Copy and recall commands are used to transfer data between the nonvolatile memory and the shadow RAM. Figure 63 describes this relationship. Nonvolatile memory recall occurs automatically at IC power up and software POR.

## Table 19. Nonvolatile Register Memory Map

| WORD | 18xh | 19xh | 1Axh* | 1Bxh | 1Cxh | 1Dxh |
|---|---|---|---|---|---|---|
| 0h | nXTable0 | nOCVTable0 | nQRTable00 | nConfig | nVAlrtTh | nUser1D0 |
| 1h | nXTable1 | nOCVTable1 | nQRTable10 | nRippleCfg | nTAlrtTh | nUser1D1 |
| 2h | nXTable2 | nOCVTable2 | nQRTable20 | nMiscCfg | nSAlrtTh | nAgeFcCfg |
| 3h | nXTable3 | nOCVTable3 | nQRTable30 | nDesignCap | nIAlrtTh | nDesignVoltage |
| 4h | nXTable4 | nOCVTable4 | nCycles | nHibCfg | nUser1C4 | nUser1D4 |
| 5h | nXTable5 | nOCVTable5 | nFullCapNom | nPackCfg | nUser1C5 | nRFastVShdn |
| 6h | nXTable6 | nOCVTable6 | nRComp0 | nRelaxCfg | nFullSOCThr | nManfctrDate |
| 7h | nXTable7 | nOCVTable7 | nTempCo | nConvgCfg | nTTFCfg | nFirstUsed |
| 8h | nXTable8 | nOCVTable8 | nIAvgEmpty | nNVCfg0 | nCGain | nSerialNumber0 |
| 9h | nXTable9 | nOCVTable9 | nFullCapRep | nNVCfg1 | nTCurve | nSerialNumber1 |
| Ah | nXTable10 | nOCVTable10 | nVoltTemp | nNVCfg2 | nTGain | nSerialNumber2 |
| Bh | nXTable11 | nOCVTable11 | nMaxMinCurr | nSBSCfg | nTOff | nDeviceName0 |
| Ch | nUser18C | nIChgTerm | nMaxMinVolt | nROMID0** | nManfctrName0 | nDeviceName1 |
| Dh | nUser18D | nFilterCfg | nMaxMinTemp | nROMID1** | nManfctrName1 | nDeviceName2 |
| Eh | nODSCTh | nVEmpty | nSOC | nROMID2** | nManfctrName2 | nDeviceName3 |
| Fh | nODSCCfg | nLearnCfg | nTimerH | nROMID3** | nRSense | nDeviceName4 |

**Locations 1A0h to 1AFh are updated automatically by the ICs each time it learns.*
***The ROM ID is unique to each IC and cannot be changed by the user.*

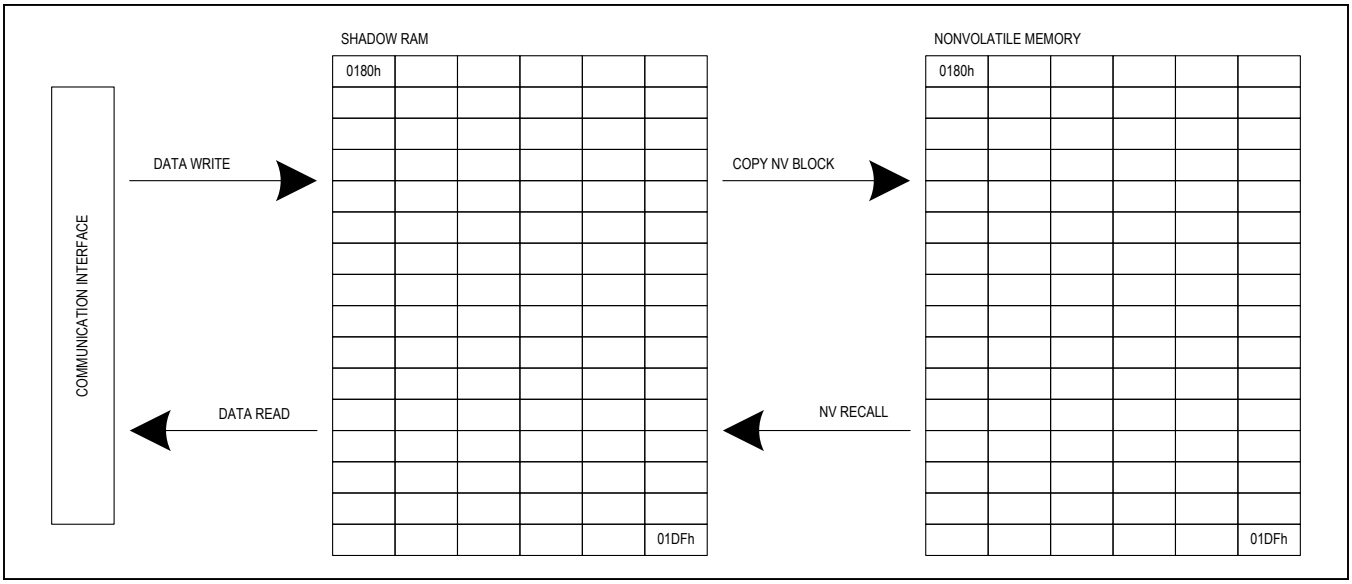



*Figure 63. Shadow RAM and Nonvolatile Memory Relationship*

## Table 20. History Recall Command Functions

| COMMAND | FUNCTION |
|---|---|
| 0xE2FE | Recall indicator flags to determine remaining SHA-256 secret updates or clears |
| 0xE2FA | Recall indicator flags to determine remaining configuration memory writes |
| 0xE2FB, 0xE2FC | Recall indicator flags to determine remaining battery life logging updates |
| 0xE2FB–0xE2FE | Recall indicator flags to determine battery life logging update errors |
| 0xE226–0xE2F0 | Recall battery life logging information |

### Nonvolatile Memory Commands

The following commands are used to copy or recall data from the nonvolatile memory. All commands are written to the Command register at memory address 060h to perform the desired operation. The CommStat register can be used to track the status of the request.

### COPY NV BLOCK [E904h]

This command copies the entire block from shadow RAM to nonvolatile memory addresses 180h to 1DFh excluding the unique ID locations of 1BCh to 1BFh. After issuing this command, the host must wait $t_{BLOCK}$ for the operation to complete. The configuration memory can be copied a maximum of 7 times. Note that the supply voltage must be above $V_{NVM}$ for the operation to complete successfully.

### NV RECALL [E001h]

This command recalls the entire block from nonvolatile memory to shadow RAM addresses 180h to 1DFh. This is an low power operation that will take up to $t_{RECALL}$ to complete. Note that the supply voltage must be above $V_{NVM}$ for the operation to complete successfully.

### HISTORY RECALL [E2XXh]

This command copies history data into page 0Eh of memory. After issuing this command, the host must wait $t_{RECALL}$ for the operation to complete before reading page 0Eh. Table 20 shows what history information can be recalled. See the *SHA-256 Authentication*, *Battery Life Logging*, and *Memory* sections for details on how to decode this information.

### Nonvolatile Block Programming

The host must program all nonvolatile memory locations at the same time by using the Copy NV Block command. The host first writes all desired nonvolatile memory shadow RAM locations to their desired values, then sends the Copy NV Block command, and then waits $t_{BLOCK}$ for the copy to complete. The CommStat.NVError bit should be read to determine if the copy command executed successfully. Afterwards the host should send the power on reset sequence to reset the IC and have the new nonvolatile settings take effect. Note that configuration memory is limited to $n_{BLOCK}$ total write attempts. The recommended full sequence is:

1. Write desired memory locations to new values.
2. Clear CommStat.NVError bit.
3. Write 0xE904 to the Command register 0x060 to initiate a block copy.
4. Wait $t_{BLOCK}$ for the copy to complete.
5. Check the CommStat.NVError bit. If set, repeat the process. If clear, continue.
6. Write 0x000F to the Command register 0x060 to POR the IC.
7. Wait $t_{POR}$ for the IC to reset.
8. Write 0x0001 to Counter Register 0x0BB to reset firmware.
9. Wait $t_{POR}$ for the firmware to restart.

### Determining the Number of Remaining Updates

The configuration memory can only be updated 7 times by the user (First update occurs during manufacturing test). The number of remaining updates can be calculated using the following procedure:

1. Write 0xE2FA to the Command register (060h).
2. Wait $t_{RECALL}$.
3. Read memory address 0EDh.
4. Decode address 0EDh data as shown in Table 21. Each block write has redundant indicator flags for reliability. Logically OR the upper and lower bytes together then count the number of 1s determine how many updates have already been used. The first update occurs in manufacturing test prior to shipping to the user.

### General-Purpose Memory

There are 7 nonvolatile memory words labelled nUser that are dedicated to general purpose user data storage. Most other nonvolatile memory locations could also be used as general purpose storage if their normal function is disabled. The nNVCfg0, nNVCfg1, and nNVCfg2 registers control which nonvolatile memory functions are enabled and disabled. Table 22 shows which nNVCfg bits control different IC functions and the effects when the bit is set or cleared. See the nNVCfg register descriptions for complete details. Do not convert a nonvolatile register to general purpose memory space if that register's function is used by the application.

## Table 21. Number of Remaining Config Memory Updates

| ADDRESS 0EDh DATA | LOGICAL OR OF UPPER AND LOWER BYTES | NUMBER OF UPDATES USED | NUMBER OF UPDATES REMAINING |
|---|---|---|---|
| 000000010000000xb | 00000001b | 1 | 7 |
| 0000001x000000xxb | 00000011b | 2 | 6 |
| 000001xx00000xxxb | 00000111b | 3 | 5 |
| 00001xxx0000xxxxb | 00001111b | 4 | 4 |
| 0001xxxx000xxxxxb | 00011111b | 5 | 3 |
| 001xxxxx00xxxxxxb | 00111111b | 6 | 2 |
| 01xxxxxx0xxxxxxxb | 01111111b | 7 | 1 |
| 1xxxxxxxxxxxxxxxb | 11111111b | 8 | 0 |

## Table 22. Nonvolatile Memory Configuration Options

| ADDRESS | REGISTER NAME | FACTORY DEFAULT | CONTROL BIT(S) | FUNCTION WHEN CONTROL BIT IS SET | FUNCTION WHEN CONTROL BIT(S) CLEARED |
|---|---|---|---|---|---|
| 180h | nXTable0 | 0x0000 | nNVCfg0.enX | 180h–18Bh Hold Custom Cell Model Information | IC Uses Default Cell Model, 180h–18Bh Become Free User Memory |
| 181h | nXTable1 | 0x0000 | | | |
| 182h | nXTable2 | 0x0000 | | | |
| 183h | nXTable3 | 0x0000 | | | |
| 184h | nXTable4 | 0x0000 | | | |
| 185h | nXTable5 | 0x0000 | | | |
| 186h | nXTable6 | 0x0000 | | | |
| 187h | nXTable7 | 0x0000 | | | |
| 188h | nXTable8 | 0x0000 | | | |
| 189h | nXTable9 | 0x0000 | | | |
| 18Ah | nXTableA | 0x0000 | | | |
| 18Bh | nXTableB | 0x0000 | | | |
| 18Ch | nUser18C | 0x0000 | N/A | Always Free User Memory | |
| 18Dh | nUser18D | 0x0000 | | | |
| 18Eh | nODSCTh | 0x0000 | nNVCfg1 enODSC | nODSCCfg→ODSCCfg | Overcurrent Comparators Default Disabled, 18Eh-18Fh Become Free User Memory |
| 18Fh | nODSCCfg | 0x0000 | | | |
| 190h | nOCVTable0 | 0x0000 | nNVCfg0.enOCV | 190h–19Bh Hold Custom Cell Model Information | IC Uses Default Cell Model, 190h–19Bh Become Free User Memory |
| 191h | nOCVTable1 | 0x0000 | | | |
| 192h | nOCVTable2 | 0x0000 | | | |
| 193h | nOCVTable3 | 0x0000 | | | |
| 194h | nOCVTable4 | 0x0000 | | | |
| 195h | nOCVTable5 | 0x0000 | | | |
| 196h | nOCVTable6 | 0x0000 | | | |
| 197h | nOCVTable7 | 0x0000 | | | |
| 198h | nOCVTable8 | 0x0000 | | | |
| 199h | nOCVTable9 | 0x0000 | | | |
| 19Ah | nOCVTableA | 0x0000 | | | |
| 19Bh | nOCVTableB | 0x0000 | | | |
| 19Ch | nIChgTerm | 0x0000 | nNVCfg0.enICT | nIChgTerm→IChgTerm | IChgTerm = FullCapRep/3, 19Ch Becomes Free User Memory |
| 19Dh | nFilterCfg | 0x0000 | nNVCfg0.enFCfg | nFilterCfg→FilterCfg | FilterCfg = 0x0EA4, 19Dh Becomes Free User Memory |
| 19Eh | nVEmpty | 0x0000 | nNVCfg0.enVE | nVEmpty→VEmpty | VEmpty = 0xA561, 19Eh Becomes Free User Memory |
| 19Fh | nLearnCfg | 0x2602 | nNVCfg0.enLCfg | nLearnCfg→LearnCfg | LearnCfg=0x2603, 19Fh Becomes Free User Memory |

## Table 22. Nonvolatile Memory Configuration Options (continued)

| ADDRESS | REGISTER NAME | FACTORY DEFAULT | CONTROL BIT(S) | FUNCTION WHEN CONTROL BIT IS SET | FUNCTION WHEN CONTROL BIT(S) CLEARED |
|---|---|---|---|---|---|
| 1A0h | nQRTable00 | 0x3C00 | N/A | nQRTable30→QRTable30 | |
| 1A1h | nQRTable10 | 0x1B80 | | | |
| 1A2h | nQRTable20 | 0x0B04 | | | |
| 1A3h | nQRTable30 | 0x0885 | | | |
| 1A4h | nCycles | 0x0000 | | Always nCycles→Cycles | |
| 1A5h | nFullCapNom | 0x0BB8 | | Always nFullCapNom→FullCapNom | |
| 1A6h | nRComp0 | 0x1070 | | Always nRComp0→RComp0 | |
| 1A7h | nTempCo | 0x263D | | Always nTempCo→TempCo | |
| 1A8h | nIAvgEmpty | 0x0000 | nNVCfg2.enIAvg | nIAvgEmpty→IAvgEmpty | IAvgEmpty = -1 x nFullCap-Nom, 1A8h Becomes Free User Memory |
| 1A9h | nFullCapRep | 0x0BB8 | nNVCfg2.enFC | nFullCapRep→FullCapRep | nFullCapNom→FullCapRep, 1A9h Becomes Free User Memory |
| 1AAh | nVoltTemp | 0x0000 | (nNVCfg0.enAF = 0) | AvgVCell→nVoltTemp and AvgTA→nVoltTemp at each backup event | Voltage and Temperature Logging disabled, 1AAh Becomes Free User Memory |
| | | | (nNVCfg2.enVT = 0) | nVoltTemp stores Age Forecasting learned information | Age Forecasting disabled, 1AAh Becomes Free User Memory |
| 1ABh | nMaxMinCurr | 0x807F | nNVCfg2.enMMC | MaxMinCurr→nMaxMinCurr at each backup event | 1ABh Becomes Free User Memory |
| 1ACh | nMaxMinVolt | 0x00FF | nNVCfg2.enMMV | MaxMinVolt→nMaxMinVolt at each backup event | 1ACh Becomes Free User Memory |
| 1ADh | nMaxMinTemp | 0x807F | nNVCfg2.enMMT | MaxMinTemp→nMaxMinTemp at each backup event | 1ADh Becomes Free User Memory |
| 1AEh | nSOC | 0x0000 | nNVCfg2.enSOC (nNVCfg0.enAF=0) | SOC→nSOC at each backup event | 1AEh Becomes Free User Memory |
| | | | (nNVCfg2.enSOC = 0) | nSOC stores Age Forecasting backup information | Age Forecasting disabled, 1AEh Becomes Free User Memory |
| 1AFh | nTimerH | 0x0000 | nNVCfg2.enT | TimerH→nTimerH at each backup event | 1AFh Becomes Free User Memory |
| 1B0h | nConfig | 0x0000 | nNVCfg0.enCfg | nConfig→Config2 | Config = 0x2214, Config2 = 0x0050, 1B0h Becomes Free User Memory |
| 1B1h | nRippleCfg | 0x0204 | N/A | Always nRippleCfg→RippleCfg | |
| 1B2h | nMiscCfg | 0x0000 | nNVCfg0.enMC | nMiscCfg→MiscCfg | 1B2h Becomes Free User Memory |
| 1B3h | nDesignCap | 0x0000 | nNVCfg0.enDC | nDesignCap→DesignCap | FullCapRep→DesignCap, 1B3h Becomes Free User Memory |

## Table 22. Nonvolatile Memory Configuration Options (continued)

| ADDRESS | REGISTER NAME | FACTORY DEFAULT | CONTROL BIT(S) | FUNCTION WHEN CONTROL BIT IS SET | FUNCTION WHEN CONTROL BIT(S) CLEARED |
|---|---|---|---|---|---|
| 1B4h | nHibCfg | 0x0000 | nNVCfg0.enHCfg | nHibCfg→HibCfg | HibCfg = 0x890B, 1B4h Becomes Free User Memory |
| 1B5h | nPackCfg | 0x0C01 for MAX172x1 0x0A02 for MAX172x5 | N/A | Always nPackCfg→PackCfg | |
| 1B6h | nRelaxCfg | 0x0000 | nNVCfg0.enRCfg | nRelaxCfg→RelaxCfg | RelaxCfg=0x2039, 1B6h Becomes Free User Memory |
| 1B7h | nConvgCfg | 0x2241 | nNVCfg1.enCTE | Converge-to-Empty Enabled | Converge-to-Empty Disabled, 1B7h Becomes Free User Memory |
| 1B8h | nNVCfg0 | 0x0100 | N/A | Always the Nonvolatile Memory Control Registers | |
| 1B9h | nNVCfg1 | 0x0006 | | | |
| 1BAh | nNVCfg2 | 0xFF0A | | | |
| 1BBh | nSBSCfg | 0x0002 | nNVCfg0.enSBS | SBS Functions Enabled | SBS Functions Disabled, 1BBh Becomes Free User Memory |
| 1BCh | nROMID0 | Varies | N/A | Always the Unique 64-bit ID | |
| 1BDh | nROMID1 | Varies | | | |
| 1BEh | nROMID2 | Varies | | | |
| 1BFh | nROMID3 | Varies | | | |
| 1C0h | nVAlrtTh | 0x0000 | nNVCfg1.enAT | nVAlrtTh→VAlrtTh | VAlrtTh = 0xFF00, 1C0h Becomes Free User Memory |
| 1C1h | nTAlrtTh | 0x0000 | | nTAlrtTh→TAlrtTh | TAlrtTh = 0x7F80, 1C1h Becomes Free User Memory |
| 1C2h | nSAlrtTh | 0x0000 | | nSAlrtTh→SAlrtTh | SAlrtTh = 0xFF00, 1C2h Becomes Free User Memory |
| 1C3h | nIAlrtTh | 0x0000 | | nIAlrtTh→IAlrtTh | IAlrtTh = 0x7F80, 1C3h Becomes Free User Memory |
| 1C4h | nUser1C4 | 0x0000 | N/A | Always Free User Memory | |
| 1C5h | nUser1C5 | 0x0000 | | | |
| 1C6h | nFullSOCThr | 0x0000 | nNVCfg1.enFTh | nFullSOCThr→FullSOCThr | FullSOCThr = 0x5005, 1C6h Becomes Free User Memory |
| 1C7h | nTTFCfg | 0x0000 | nNVCfg1.enTTF | nTTFCfg configures time to full calculation | Time-to-Full Default Configuration, 1C7h Becomes Free User Memory |
| 1C8h | nCGain | 0x0000 | nNVCfg1.enCG | nCGain.nCGain→CGainnCGain. nCOff→COff | CGain = 0x0400, COff = 0x0000, 1C8h Becomes Free User Memory |
| 1C9h | nTCurve | 0x0025 | (nNVCfg1.enCrv = 0) | nTCurve→CGTempCo | CGTempCo = 0x20C8, 1C9h Becomes Free User Memory |
| | | | (nNVCfg2.enMet = 0) | Thermistor Curvature Controlled by nTCurve | 1C9h Becomes Free User Memory |

## Table 22. Nonvolatile Memory Configuration Options (continued)

| ADDRESS | REGISTER NAME | FACTORY DEFAULT | CONTROL BIT(S) | FUNCTION WHEN CONTROL BIT IS SET | FUNCTION WHEN CONTROL BIT(S) CLEARED |
|---|---|---|---|---|---|
| 1CAh | nTGain | 0x0000 | nNVCfg1.enTGO | nTGain→TGain | TGain = 0xEE56, 1CAh Becomes Free User Memory |
| 1CBh | nTOff | 0x0000 | | nTOff→TOff | TOff = 0x1DA4, 1CBh Becomes Free User Memory |
| 1CCh | nManfctrName0 | 0x0000 | nNVCfg0.enSBS | nManfctrName[2:0]→sManfctrName | 1CCh-1CEFh Become Free User Memory |
| 1CDh | nManfctrName1 | 0x0000 | | | |
| 1CEh | nManfctrName2 | 0x0000 | | | |
| 1CFh | nRSense | 0x03E8 | N/A | Always Sense Resistor value | |
| 1D0h | nUser1D0 | 0x0000 | N/A | Always Free User Memory | |
| 1D1h | nUser1D1 | 0x0000 | | | |
| 1D2h | nAgeFcCfg | 0xD5E3 | nNVCfg0.enAF | Configures Age Forecast Feature | 1D2h Becomes Free User Memory |
| 1D3h | nDesignVoltage | 0x0000 | nNVCfg0.enSBS | nDesignVoltage→sDesignVolt | 1D3h Becomes Free User Memory |
| 1D4h | nUser1D4 | 0x0000 | N/A | Always Free User Memory | |
| 1D5h | nRFastVShdn | 0x0000 | nNVCfg1.enRFVSH | nRFastVShdn. Rfast→RFastnRFastVshdn. VShdn→VShdnCfg | VShdnCfg = 0x007D, RFast = 0x0500, 1D5h Becomes Free User Memory |
| 1D6h | nManfctrDate | 0x0000 | nNVCfg0.enSBS | nManfctrDate→sManfctrDate | 1D6h Becomes Free User Memory |
| 1D7h | nFirstUsed | 0x0000 | | nFirstUsed→sFirstUsed | 1D7h Becomes Free User Memory |
| 1D8h | nSerialNumber0 | 0x0000 | | nSerialNumber[2:0]→sSerialNumber | 1D8h-1DAFh Become Free User Memory |
| 1D9h | nSerialNumber1 | 0x0000 | | | |
| 1DAh | nSerialNumber2 | 0x0000 | | | |
| 1DBh | nDeviceName0 | 0x0000 | | nDeviceName[4:0]→sDeviceName | 1DBh-1DFh Become Free User Memory |
| 1DCh | nDeviceName1 | 0x0000 | | | |
| 1DDh | nDeviceName2 | 0x0000 | | | |
| 1DEh | nDeviceName3 | 0x0000 | | | |
| 1DFh | nDeviceName4 | 0x0000 | | | |

### Memory Locks

ModelGauge m5 RAM Registers and all nonvolatile memory locations can be permanently locked to prevent accidental data loss in the application. Locking a memory block only prevents future writes to the locations. Reading locked locations is still allowed. **Note that locking a memory location is permanent so carefully choose all desired locks before sending the NV LOCK command.** The SHA secret is stored in separate secure non-readable memory. There is a different command for locking the SHA secret and its state is not displayed in the Lock register. See the *SHA-256 Authentication* section for details. Once a lock bit is set, it can never be cleared. Figure 64 shows which lock bits correspond to which memory blocks of the IC.

### NV LOCK [6AXXh]

This command permanently locks a block or blocks of memory. To set a lock, send 6AXXh to the Command register where the lower 5 bits of the command determine which locks will be set. Figure 64 shows a detailed format of the NV LOCK command. Set each individual LOCK bit to 1 to LOCK the corresponding register block. Set the LOCK bit to 0 to do nothing at this time. For example writing 6A02h to the Command register sets LOCK2. Writing 6A1Fh sets all five locks. Writing 6A00h sets no locks.

**LOCK1:** Locks register pages 1Ah, 1Bh

**LOCK2:** Locks register pages 01h, 02h, 03h, 04h, 0Bh, 0Dh

**LOCK3:** Locks register pages 18h, 19h

**LOCK4:** Locks register pages 1Ch

**LOCK5:** Locks register pages 1Dh

### Locking Memory Blocks

Prior to sending the lock command the CommStat. NVError bit should be cleared and after the command is sent the CommStat.NVError bit should be read to determine if the lock command executed successfully. Note that locking memory blocks is a permanent operation. The recommended full sequence is:

5) 1) Clear CommStat.NVError bit.

6) 2) Write 0x6AXX to the Command register 0x060 to lock desired blocks.

7) 3) Wait $t_{UPDATE}$ for the copy to complete.

8) 4) Check the CommStat.NVError bit. If set, repeat the process.

### Reading Lock State

The Lock register at address 07Fh reports the state of each lock. See Figure 65 for the format of the Lock register. If a LOCK bit is set the corresponding memory block is locked. If the LOCK bit is cleared the corresponding memory block is unlocked. Note that the SHA-256 Secret lock state cannot be read through this register.

**X:** Don't Care

**1:** LOCK is set

**0:** LOCK is clear

## Smart Battery Compliant Operation

The MAX17201/MAX17205 is compliant to the Smart Battery Specification v1.1 when nNVCfg0.enSBS =1. Enabling SBS operation does not interfere with normal operation of the IC. SBS formatted registers are accessed at slave address 16h, memory addresses 100h to 17Fh using SBS protocols. SBS functionality can be configured using the nSBSCfg and nDesignVoltage registers.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | LOCK5 | LOCK4 | LOCK3 | LOCK2 | LOCK1 |

*Figure 64. Format of LOCK Command*

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X | X | X | X | LOCK5 | LOCK4 | LOCK3 | LOCK2 | LOCK1 |

*Figure 65. Format of Lock Register (07Fh)*

### SBS Compliant Memory Space (MAX1720x Only)

The MAX1720x contains an SBS v1.1 Compliant memory space on pages 10h to 17h that can be accessed using the Read Word, Write Word, and Read Block commands at 2-Wire slave address 16h. Table 23 lists the SBS compliant registers. See the SBS 1.1 Specification for details of registers at addresses 100h to 12Fh. Registers with colored boxes in the table are shared between SBS and normal IC functions and are always readable regardless of IC settings. Their format is described in the *Analog Measurements* section of the data sheet. All other registers on pages 13h to 17h are described in this section. Unnamed register locations are reserved and should not be written to.

### sFirstUsed Register (136h)

This register contains a mirror of the value stored in nonvolatile memory address 1D7h.

### sCell1 (13Fh)/sCell2 (13Eh)/sCell3 (13Dh)/sCell4 (13Ch) Registers

These registers contain the same cell voltage information displayed in Cell1 (0D8h) to Cell4 (0D5h) respectively with SBS compliant formatting. In the sCell registers 1 LSb = 1mV giving a full scale range of 0.0V to 65.535V.

### sAvgCell1 (14Fh)/sAvgCell2 (14Eh)/sAvgCell3 (14Dh)/sAvgCell4 (14Ch) Registers

These registers contain the same average cell voltage information displayed in AvgCell1 (0D4h) to AvgCell4 (0D1h) respectively with SBS compliant formatting. In the sCell registers 1 LSb = 1mV giving a full scale range of 0.0V to 65.535V.

## Table 23. SBS Register Space Memory Map

| WORD | 10xh | 11xh | 12xh | 13xh | 14xh | 15xh | 16xh | 17xh |
|---|---|---|---|---|---|---|---|---|
| 0h | sManfct Access | sFullCap | sManfc-trName* | — | — | — | — | sManfctInfo** |
| 1h | sRemCa-pAlarm | sRunTTE | sDevice-Name* | — | — | — | — | — |
| 2h | sRemTime-Alarm | sAvgTTE | sDev Chemistry* | — | — | — | — | — |
| 3h | sBatteryMode | sAvgTTF | sManfct-Data** | — | — | — | — | — |
| 4h | sAtRate | sCharging-Current | — | Temp1 | — | — | — | — |
| 5h | — | sCharging-Voltage | — | IntTemp | — | — | — | — |
| 6h | sAtTTE | sBattery Status | — | sFirstUsed | — | — | — | — |
| 7h | sAtRateOK | sCycles | — | AvgTemp1 | — | — | sAvCap | — |
| 8h | sTemperature | sDesignCap | — | AvgIntTemp | — | — | sMixCap | — |
| 9h | sPack Voltage | sDesignVolt | — | AvgTemp2 | — | — | — | — |
| Ah | sCurrent | sSpecInfo | — | — | — | — | — | — |
| Bh | sAvg Current | sManfct-Date | — | Temp2 | — | — | — | — |
| Ch | sMaxError | sSerial-Number** | — | sCell4 | sAvgCell4 | — | — | — |
| Dh | sRelSOC | — | — | sCell3 | sAvgCell3 | — | CGTempCo | — |
| Eh | sAbsSOC | — | — | sCell2 | sAvgCell2 | — | — | — |
| Fh | sRemCap | — | — | sCell1 | sAvgCell1 | — | — | — |

**Location is read as ASCII data using the read block command.*
***Location is read as hexadecimal data using the read block command.*

### sAvCap Register (167h)

This register contains the same information as the AvCap (01Fh) register. It is formatted for SBS compliance where 1 LSb = 1.0mAH giving a full-scale range of 0.0mAh to 65535mAh.

### sMixCap Register (168h)

This register contains the same information as the MixCap (00Fh) register. It is formatted for SBS compliance where 1 LSb = 1.0mAH giving a full-scale range of 0.0mAh to 65535mAh.

### sManfctInfo Register (170h)

The sManfctInfo register is accessed using the SBS protocol read block command. This register is intended to provide custom information, and is outside the SBS 1.1 definition. This register function is not supported in the MAX17201/MAX17205.

### Nonvolatile SBS Register Back-Up

When SBS mode operation is enabled by setting nNVCfg0.enSBS = 1, data from several nonvolatile memory locations is translated into SBS memory space. Table 24 lists these translations. Note that when performing an SBS Read Block command, the IC automatically generates the size data byte by counting the number of sequential non-zero data bytes stored in the corresponding nonvolatile memory locations. The nonvolatile memory only needs to store the actual data to be read by an SBS Read Block command. If SBS mode of operation is disabled, these locations become available for general purpose nonvolatile data storage.

### nSBSCfg Register (1BBh)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register.

The nSBSCfg register manages settings for SBS mode operation of the IC. If nNVCfg0.enSBS = 0 and SBS mode is not used, this register can be used as general-purpose data storage. Figure 66 shows the register format.

**MECfg:** Configures sMaxError register output when operating in SBS mode.

00: Always report 0% error

01: Always report 1% error

10: Report actual experienced errror

11: Always report 3% error

**CapMd:** Selects sBatteryMode.CapMd bit default setting when operating in SBS mode. CapMd will reset to 0 every time a pack removal occurs as detected by floating communication lines.

### nDesignVoltage Register (1D3h)

Register Type: Special

Nonvolatile Restore: There is no associated restore location for this register

The nDesignVoltage register holds a value representing the expected average cell voltage of the design. This value is used for SBS mWh calculations. The nDesignVoltage register has an LSb of 1mV giving a full-scale range of 0mV to 65.535V.

## Table 24. SBS to Nonvolatile Memory Mapping

| NONVOLATILE MEMORY ADDRESS | NONVOLATILE MEMORY REGISTER NAME | SBS MEMORY ADDRESS | SBS REGISTER NAME |
|---|---|---|---|
| 1D6h | nManfctrDate | 1Bh | sManfctrDate |
| 1D7h | nFirstUsed | 36h | sFirstUsed |
| 1CCh-1CEh | nManfctrName[2:0] | (Read Block Command) | sManfctrName |
| 1D8h-1DAh | nSerialNumber[2:0] | (Read Block Command) | sSerialNumber |
| 1DBh-1DFh | nDeviceName[4:0] | (Read Block Command) | sDeviceName |

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CapMd | X | X | X | X | X | X | X | X | X | X | X | X | MECfg | | X |

*X: Don't care. This bit is undefined and can be logic 0 or 1.*

*Figure 66. nSBSCfg (1BBh) Format*

### nCGain and Sense Resistor Relationship

To meet SBS compliance, current and capacity registers in the SBS memory space must have a an LSb bit weight of 1.0mA or 1.0mAh. The current gain must be adjusted based on the application sense resistor value to set the proper bit weight. Table 25 shows the proper nCGain value to use for the most common-sense resistor values. This is the default register value only. It does not include any offset trim or custom gain adjustment. Note that changing the nCGain register affects the gain reported by the standard ModelGauge current and capacity registers.

## SHA-256 Authentication

Authentication is performed using a FIPS 180-4 compliant SHA-256 one-way hash algorithm on a 512-bit message block. The message block consists of a 160-bit secret, a 160-bit challenge and 192 bits of constant data. Optionally, the 64-bit ROM ID replaces 64 of the 192 bits of constant data used in the hash operation. Contact Maxim for details of the message block organization.

The host and the IC both calculate the result based on the mutually known secret. The result of the hash operation is known as the message authentication code (MAC) or message digest. The MAC is returned by the IC for comparison to the host's MAC. Note that the secret is never transmitted on the bus and thus cannot be captured by observing bus traffic. Each authentication attempt is initiated by the host system by writing a 160-bit random challenge into the SHA memory address space 0C0h to 0C9h. The host then issues the compute MAC or compute MAC with ROM ID command. The MAC is computed per FIPS 180-4, and stored in address space 0C0h to 0CFh overwriting the challenge value.

## Table 25. nCGain Register Settings to Meet SBS Compliance

| SENSE RESISTOR VALUE | NCGAIN REGISTER VALUE | CORRESPONDING CGAIN REGISTER VALUE |
|---|---|---|
| 0.0025 Ω | 0x4000 | 0x0400 |
| 0.005 Ω | 0x2000 | 0x0200 |
| 0.010 Ω | 0x1000 | 0x0100 |

Note that the results of the authentication attempt are determined by host verification. Operation of the ICs is not affected by authentication success or failure.

### Authentication Procedure

Figure 67 shows how a host system verifies the authenticity of a connected battery. The host first generates a random 160 bit challenge value and writes the challenge to IC memory space 0C0h-0C9h. The host then sends the Compute MAC with ROM ID (3500h) or Compute MAC without ROM ID (3600h) to the Command register 060h and wait $t_{SHA}$ for computation to complete. Finally, the host reads the MAC from memory space 0C0h-0CFh to verify the result. This procedure requires the secret to be maintained on the host side as well as in the battery. The host must perform the same calculations in parallel to verify the battery is authentic.

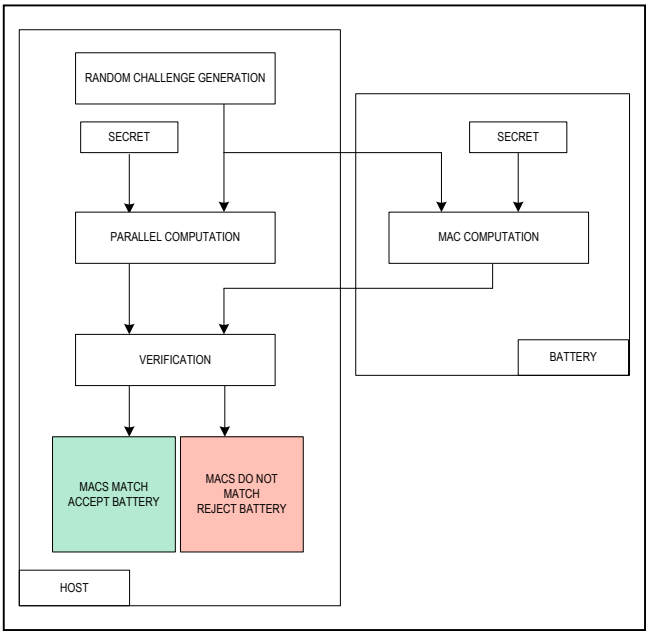



*Figure 67. Procedure to Verify a Battery*

### Alternate Authentication Procedure

Figure 68 shows an alternative method of battery authentication which does not require the host to know the secret. In this method, each host device knows a challenge and MAC pair that match the secret stored in an authentic battery, but each host device uses a different pair. This eliminates the need for special hardware on the host side to protect the secret from hardware intrusion. A battery could be cloned for a single host device, but creating a clone battery that works with any host would not be possible without knowing the secret.

The authentication process for this method is less complex. The host simply writes the challenge to the IC memory space 0C0h-0C9h. The host then sends the Compute MAC without ROM ID (3600h) to the Command register 060h. Note that Compute MAC with ROM ID Command is not valid for this authentication method. The host then waits $t_{SHA}$ for computation to complete and reads the MAC from memory space 0C0h-0CFh to verify the result.

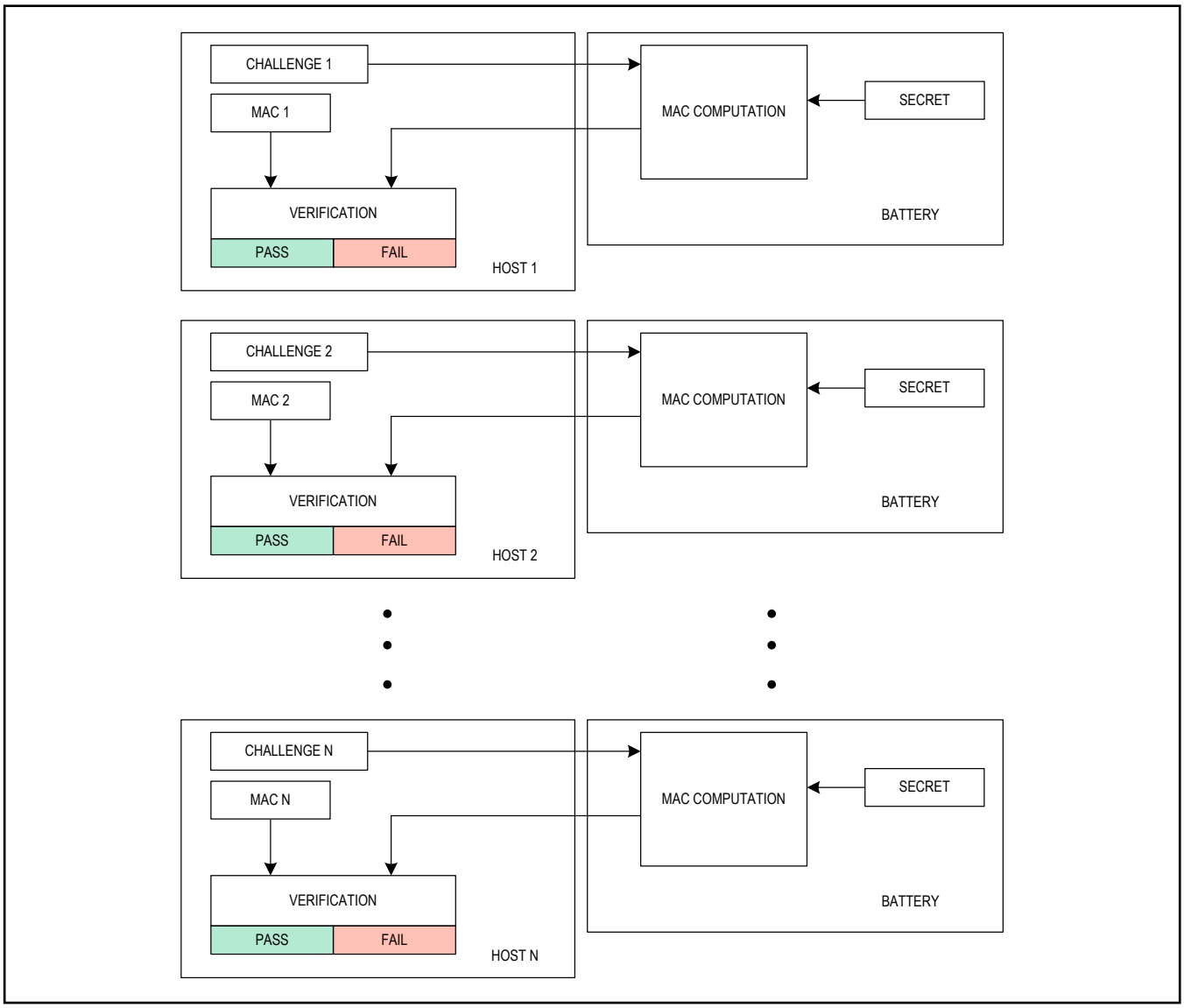



*Figure 68. Battery Authentication Without a Host-Side Secret*

### Secret Management

The secret value must be programmed to a known value prior to performing authentication in the application. The secret cannot be written directly. Instead, the user must generate a new internal secret by performing a SHA computation with the old internal secret and a seed value sent as a challenge. To prevent any one entity from knowing the complete secret value, the process can be repeated multiple times by sending additional challenge seeds and performing additional computations.

Note that secret memory can only be changed a maximum of $n_{SECRET}$ times including erase operations and nonvolatile memory updates are not guaranteed. See the $n_{SECRET}$ write limit in the *Electrical Characteristics* table. Any secret update operation that fails does not change the secret value stored in the ICs, but consumes one of the available limited updates. Be careful not to use up all secret memory during the generation process. Maxim strongly recommends permanently locking the secret after it has been generated.

### Single Step Secret Generation

The single step secret generation procedure should be used in production environments where the challenge seed value can be kept confidential, for example when there are no OEM manufacturing steps or situations where an outside individual or organization would need to know the challenge seed. Use the following sequence to program the ICs. Since the secret cannot be read from the ICs, a parallel computation must be performed externally in order the calculate the stored secret. Figure 69 shows an example single step secret generation operation. Note that new units have their secret value already cleared to all 0s.

1. Clear the CommStat.NVError bit.
2. Write a challenge seed value to the SHA memory space 0C0h–0C9h.
3. Write Compute Next Secret with ROM ID 3300h or Compute Next Secret without ROM ID 3000h to the Command register 060h.
4. Wait $t_{SHA}$ + $t_{UPDATE}$ for computation to complete and new secret to be stored.
5. If CommStat.NVError is set, return to step 1. Otherwise continue.
6. Verify the secret has been generated correctly with a test challenge at this time. If verification fails return to step 1. Also, see the *Determining the Number of Remaining Updates* section to verify enough nonvolatile memory writes remain in order to repeat the process.
7. Write Lock Secret 6000h to the Command register 060h. **Note this operation cannot be reversed.**
8. Wait $t_{UPDATE}$ for secret to lock permanently.

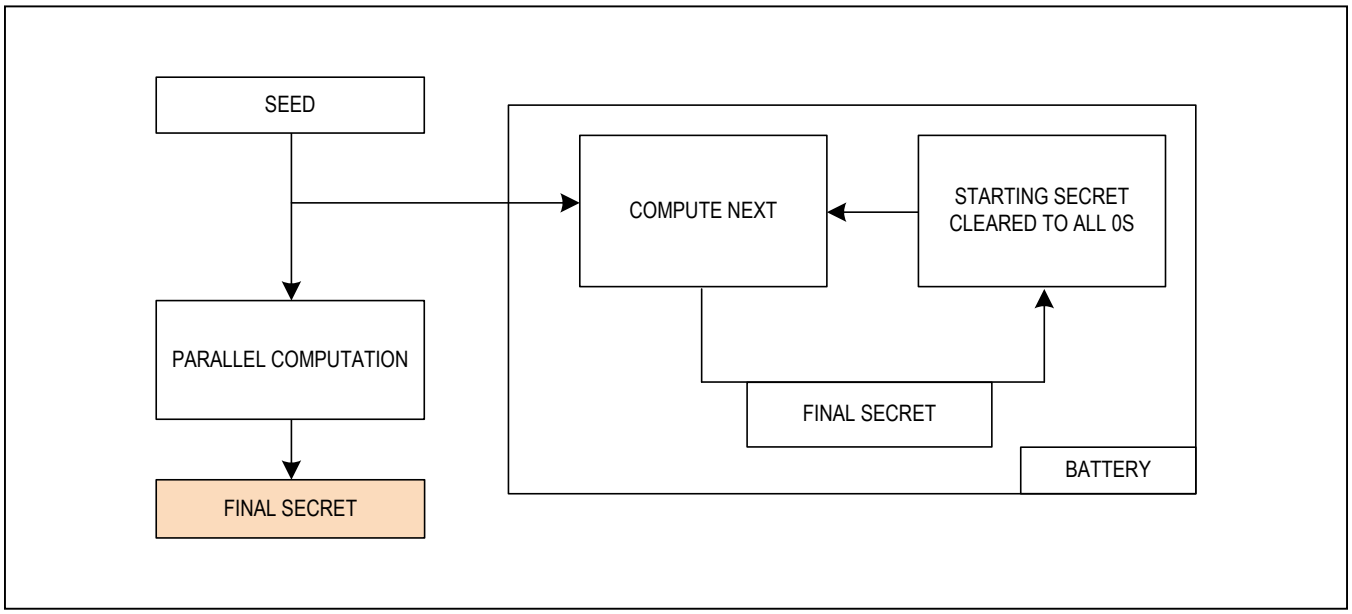



*Figure 69. Single Step Secret Generation Example*

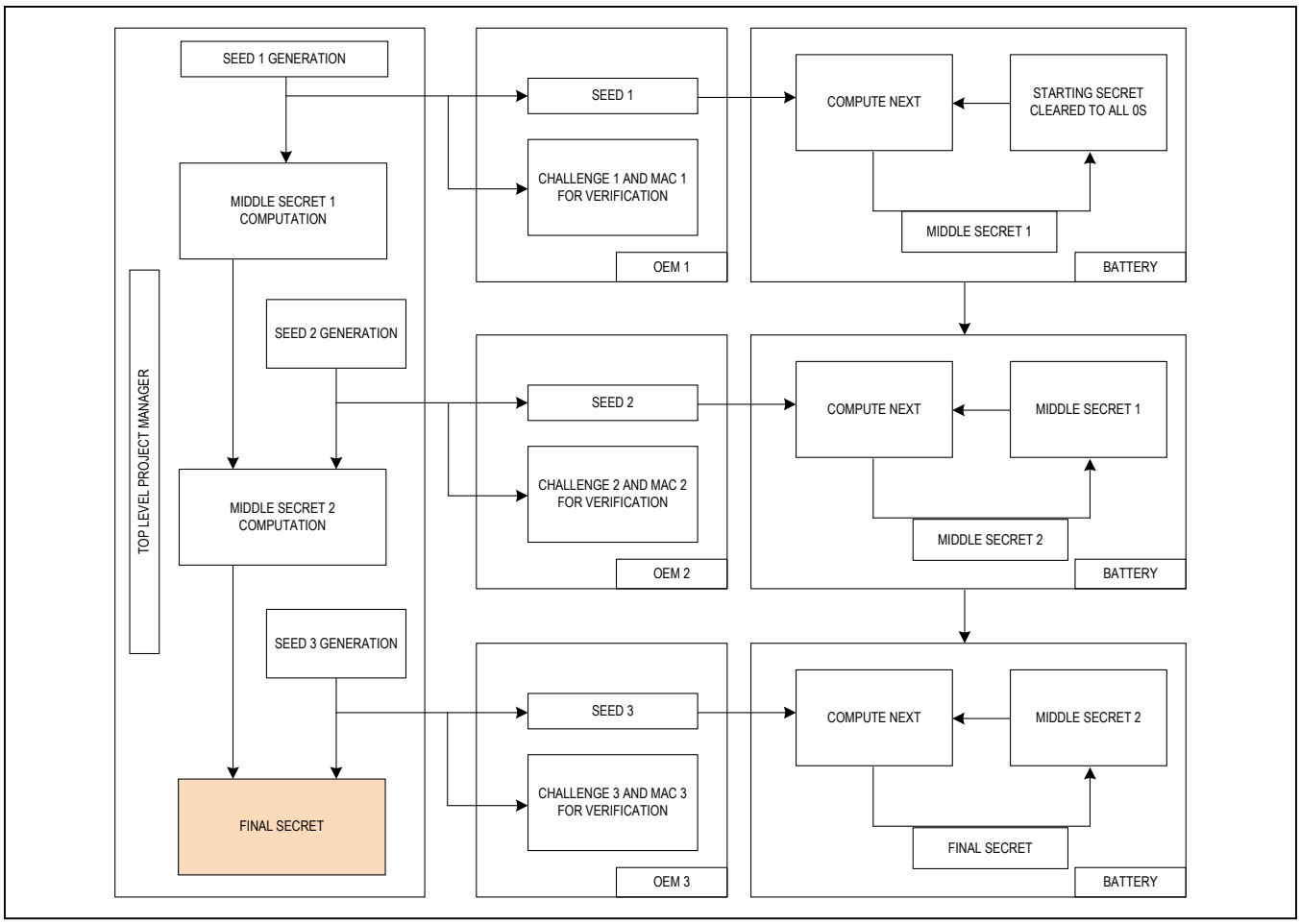



*Figure 70. Multistep Secret Generation Example*

### Multistep Secret Generation Procedure

The multistep secret generation procedure should be used in environments where an outside individual or organization would need to know the challenge seed such as OEM manufacturing. The multistep procedure is more complicated, but allows a secret to be stored inside the ICs without providing any information to an OEM manufacturer that could jeopardize secret integrity. Figure 70 shows an example where three OEM manufacturers are each provided with a seed value for a Compute Next operation. The final secret value stored inside the ICs are known only to the top-level manager who knows all seed values and has performed the computation separately. Use the following procedures when generating a multistep secret. Note that the secret can only be updated or cleared $n_{SECRET}$ times total. New units have their secret value already cleared to all 0s.

### All OEMs

1. Clear the CommStat.NVError bit.
2. Write challenge seed value to the SHA memory space 0C0h–0C9h.
3. Write Compute Next Secret with ROM ID 3300h or Compute Next Secret without ROM ID 3000h to the Command register 060h.
4. Wait $t_{SHA}$ + $t_{UPDATE}$ for computation to complete and new secret to be stored.
5. If CommStat.NVError is set, return to step 1. Otherwise, continue.
6. Verify the secret has been generated correctly with a test challenge at this time. If verification fails return to step 1. Also see the *Determining the Number of Remaining Updates* section to verify enough nonvolatile memory writes remain in order to repeat the process.

**Last OEM:**

1. Follow the procedure in the *All OEMs* section for the final secret update.
2. Write Lock Secret 6000h to the Command register 060h. **Note this operation cannot be reversed.**
3. Wait $t_{UPDATE}$ for secret to lock permanently.

**Top Level:**

1. Generate all seed values to provide to OEMs.
2. Perform SHA calculations separately to determine what the final secret is after all manufacturing steps.
3. Keep final secret value secure.

### Determining the Number of Remaining Updates

The internal secret can only be updated or cleared $n_{SECRET}$ times total. The number of remaining updates can be calculated using the following procedure:

1. Write 0xE2FE to the Command register (060h).
2. Wait $t_{RECALL}$.
3. Read memory address 0E6h.
4. Decode address 0E6h data as shown in Table 26. Each secret update has redundant indicator flags for reliability. Logically OR the upper and lower bytes together then count the number of 1s to determine how many updates have already been used. The first update occurs in manufacturing test to clear the secret memory prior to shipping to the user.

### Authentication Commands

All SHA authentication commands are written to memory address 060h to perform the desired operation. Writing the challenge or reading the MAC is handled by accessing the SHA memory space on page 0Ch through direct write and read operations.

### Compute MAC Without ROM ID [3600h]

The challenge value must be written to the SHA memory space prior to performing a compute MAC. This command initiates a SHA-256 computation without including the ROM ID in the message block. Instead, the ROM ID portion of the message block is replaced with a value of all 1s. Since the ROM ID is not used, this command allows the use of a master secret and MAC response independent of the ROM ID. The ICs compute the MAC in $t_{SHA}$ after receiving the last bit of this command. After the MAC computation is complete, the host can read the MAC from the SHA memory space.

### Compute MAC with ROM ID [3500h]

The challenge value must be written to the SHA memory space prior to performing a compute MAC. This command is structured the same as the compute MAC without ROM ID, except that the ROM ID is included in the message block. With the unique ROM ID included in the MAC computation, the MAC is unique to each unit. After the MAC computation is complete, the host can read the MAC from the SHA memory space.

### Compute Next Secret Without ROM ID [3000h]

This command initiates a SHA-256 computation and uses the resulting MAC as the next or new secret. The hash operation is performed with the current 160-bit secret and the new 160-bit challenge. Logical 1s are loaded in place of the ROM ID. The last 160 bits of the MAC are used as the new secret value. The host must allow $t_{SHA}$ after issuing this command for the SHA calculation to complete, then wait $t_{UPDATE}$ for the new secret value to be stored in nonvolatile memory. During this operation, the SHA memory space is not updated. Note that the old secret value must be known prior to executing this command in order to calculate what the new secret value is.

## Table 26. Number of Remaining Secret Updates

| ADDRESS 0E6H DATA | LOGICAL OR OF UPPER AND LOWER BYTES | NUMBER OF UPDATES USED | NUMBER OF UPDATES REMAINING |
|---|---|---|---|
| 000000010000000xb | 00000001b | 1 | 5 |
| 0000001x000000xxb | 00000011b | 2 | 4 |
| 000001xx00000xxxb | 00000111b | 3 | 3 |
| 00001xxx0000xxxxb | 00001111b | 4 | 2 |
| 0001xxxx000xxxxxb | 00011111b | 5 | 1 |
| 001xxxxx00xxxxxxb | 00111111b | 6 | 0 |

### Compute Next SECRET WITH ROM ID [3300h]

This command initiates a SHA-256 computation and uses the resulting MAC as the next or new secret. The hash operation is performed with the current 160-bit secret, the 64-bit ROM ID, and the new 160-bit challenge. The last 160 bits of the output MAC are used as the new secret value. The host must allow $t_{SHA}$ after issuing this command for the SHA calculation to complete, then wait $t_{UPDATE}$ for the new secret value to be stored in nonvolatile memory. During this operation, the SHA memory space is not updated. Note that the old secret value must be known prior to executing this command in order to calculate what the new secret value is.

### CLEAR SECRET [5A00h]

This command sets the 160-bit secret to all 0s. The host must wait $t_{UPDATE}$ for the IC to write the new secret value to nonvolatile memory. This command uses up one of the secret write cycles.

### LOCK SECRET [6000h]

This command write protects the secret to prevent accidental or malicious overwrite of the secret value. The secret value stored in nonvolatile memory becomes permanent. The host must wait $t_{UPDATE}$ for the lock operation to complete.

SHA-256 Lock state is not shown in the Lock register. Lock state can be verified by reading nonvolatile memory history using the following sequence:

1. Send 0xE2FA to the Command register (060h)
2. Wait for $t_{RECALL}$.
3. Read memory address 0ECh

If address 0ECh is 0x0000 then the secret is not locked. If address 0ECh is anything other than 0x0000 then the secret is permanently locked.

## Device Reset

There are two levels of reset for the IC. A full reset restores the ICs to their power-up state the same as if power had been cycled. A fuel gauge reset resets only the fuel gauge operation without resetting IC hardware. This is useful for testing different configurations without writing nonvolatile memory. Use the following sequences to reset the ICs.

### Full Reset

1. Reset IC hardware by writing 000Fh to the Command register at 060h.
2. Wait $t_{POR}$ for reset to take affect.
3. Reset IC fuel gauge operation by writing 0001h to the Config2 register at 0BBh.
4. Wait $t_{POR}$ for the fuel gauge to reset.

### Fuel Gauge Reset

1. Reset IC fuel gauge operation by writing 0001h to the Config2 register at 0BBh.
2. Wait $t_{POR}$ for the fuel gauge to reset.

### Reset Commands

There are two commands that can be used to reset either the entire IC or just operation of the fuel gauge. Note that the reset fuel gauge command is written to Config2 instead of to the Command register.

### Hardware Reset [000Fh to Address 060h]

Send the hardware reset command to the Command register to recall all nonvolatile memory into shadow RAM and reset all hardware based operations of the IC. This command should always be followed by the reset fuel gauge command to fully reset operation of the IC.

### Fuel Gauge Reset [0001h to Address 0BBh]

The fuel gauge reset command resets operation of the ICs without restoring nonvolatile memory values into shadow RAM. This command allows different configurations to be tested without using one of the limited number of nonvolatile memory writes.

## Communication

This section covers communication protocols and summarizes all special commands used by the ICs. The MAX17201/MAX17205 communicates over a 2-wire interface using either I²C or SBS protocols depending on memory address selected by the host. The MAX17211/MAX17215 communicates using the Maxim 1-Wire interface.

### 2-Wire Bus System (MAX17201/MAX17205 Only)

The MAX17201/MAX17205 uses a 2-Wire bus system to communicate by both standard I²C protocol or by SBS smart battery protocol. The slave address used by the host to access the ICs determines which protocol is used and what memory locations are available to read or write. The following description applies to both protocols. See the I²C and SBS Bus System descriptions for specific protocol details.

### Hardware Configuration

The 2-wire bus system supports operation as a slave-only device in a single or multi-slave, and single or multi-master system. Up to 128 slave devices may share the bus using 7-bit slave addresses. The 2-wire interface consists of a serial data line (SDA) and serial clock line (SCL). SDA and SCL provide bidirectional communication between the ICs and a master device at speeds up to 400kHz. The ICs' SDA pin operates bi-directionally. When the ICs receive data SDA operates as an input. When the IC returns data SDA operates as an open-drain output with the host system providing a resistive pull-up. See Figure 71. The IC always operates as a slave device, receiving and transmitting data under the control of a master device. The master initiates all transactions on the bus and generates the SCL signal, as well as the START and STOP bits that begin and end each transaction.

### I/O Signaling

The following individual signals are used to build byte level 2-Wire communication sequences.

### Bit Transfer

One data bit is transferred during each SCL clock cycle, with the cycle defined by SCL transitioning low to high and then high to low. The SDA logic level must remain stable during the high period of the SCL clock pulse. Any change in SDA when SCL is high is interpreted as a START or STOP control signal.

### Bus Idle

The bus is defined to be idle, or not busy, when no master device has control. Both SDA and SCL remain high when the bus is idle. The STOP condition is the proper method to return the bus to the idle state.

### START and STOP Conditions

The master initiates transactions with a START condition by forcing a high-to-low transition on SDA while SCL is high. The master terminates a transaction with a STOP condition by a low-to-high transition on SDA while SCL is high. A Repeated START condition can be used in place of a STOP then START sequence to terminate one transaction and begin another without returning the bus to the idle state. In multimaster systems, a repeated START allows the master to retain control of the bus. The START and STOP conditions are the only bus activities in which the SDA transitions when SCL is high.

### Acknowledge Bits

Each byte of a data transfer is acknowledged with an acknowledge bit (ACK) or a no acknowledge bit (NACK). Both the master and the IC slave generate acknowledge bits. To generate an acknowledge, the receiving device must pull SDA low before the rising edge of the

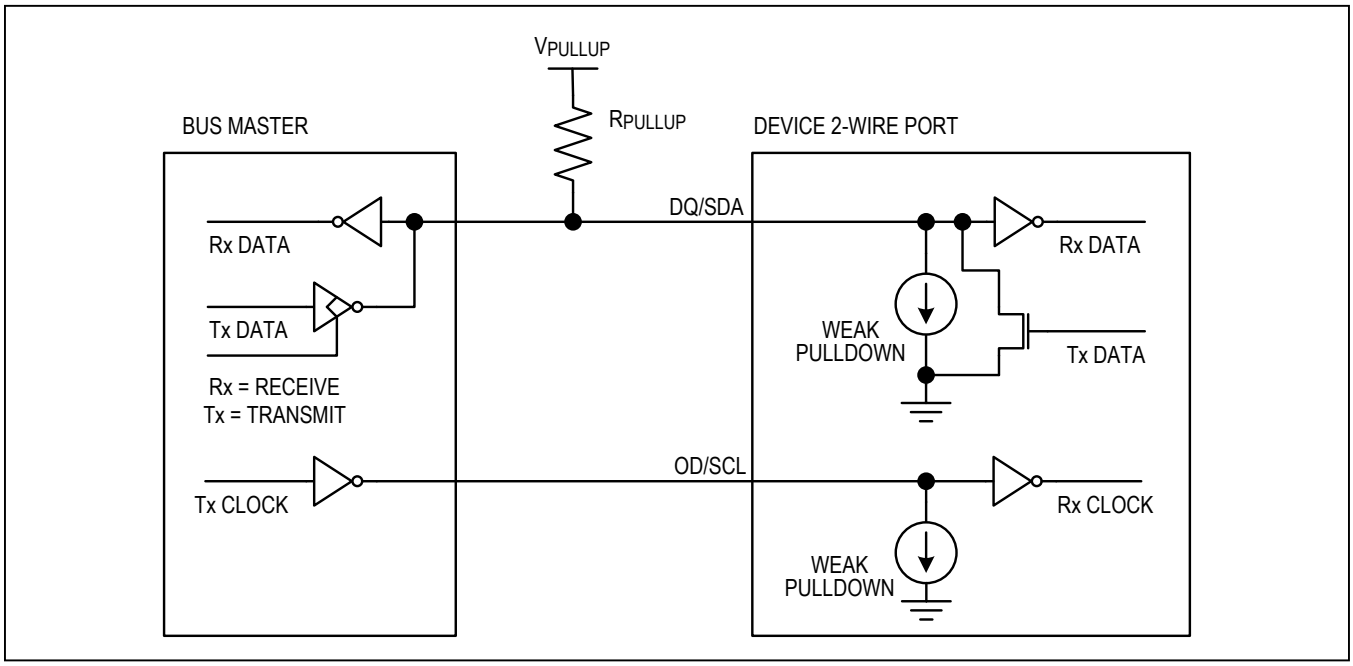



*Figure 71. 2-Wire Bus Interface Circuitry*

acknowledge-related clock pulse (ninth pulse) and keep it low until SCL returns low. To generate a no acknowledge, the receiver releases SDA before the rising edge of the acknowledge-related clock pulse and leaves SDA high until SCL returns low. Monitoring the acknowledge bits allows for detection of unsuccessful data transfers. An unsuccessful data transfer can occur if a receiving device is busy or if a system fault has occurred. In the event of an unsuccessful data transfer, the bus master should reattempt communication. If a transaction is aborted mid-byte, the master should send additional clock pulses to force the slave IC to free the bus prior to restarting communication.

### Data Order

With 2-wire communication, a byte of data consists of 8 bits ordered most significant bit (MSb) first. The least significant bit (LSb) of each byte is followed by the Acknowledge bit. IC registers composed of multibyte values are ordered least significant byte (LSB) first.

### Slave Address

A bus master initiates communication with a slave device by issuing a START condition followed by a slave address and the read/write (R/W) bit. When the bus is idle, the ICs continuously monitor for a START condition followed by its slave address. When the ICs receive a slave address that matches its slave address, they respond with an acknowledge bit during the clock period following the R/W bit. The MAX1720x supports the slave addresses shown in Table 27.

### Read/Write Bit

The R/W bit following the slave address determines the data direction of subsequent bytes in the transfer. R/W = 0 selects a write transaction, with the following bytes being written by the master to the slave. R/W = 1 selects a read transaction, with the following bytes being read from the slave by the master.

### Bus Timing

The IC is compatible with any bus timing up to 400kHz. See the *Electrical Characteristics* table for timing details. No special configuration is required to operate at any speed. Figure 72 shows an example of standard 2-wire bus timing.

## Table 27. 2-Wire Slave Addresses

| SLAVE ADDRESS | PROTOCOL | ADDRESS BYTE RANGE | INTERNAL MEMORY RANGE ACCESSED |
|---|---|---|---|
| 6Ch | I²C | 00h–FFh | 000h–0FFh |
| 16h | SMBUS | 00h–7Fh | 100h–17Fh |
| | I²C | 80h–FFh | 180h–1FFh |

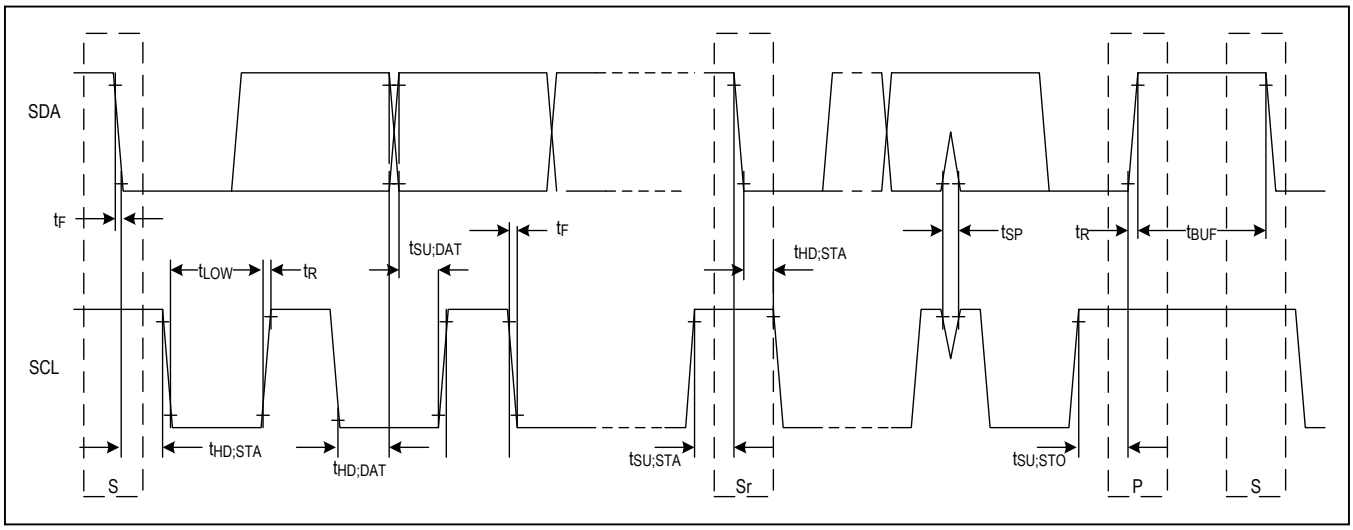



*Figure 72. 2-Wire Bus Timing Diagram*

### I²C Protocols

The following 2-wire communication protocols must be used by the bus master to access MAX17201/MAX17205 memory locations 000h to 1FFh. Addresses 000h to 0FFh and from 180h to 1FFh can be read continuously. Addresses 100h to 17Fh must be read one word at a time. These protocols follow the standard I²C specification for communication.

### I²C Write Data Protocol

The write data protocol is used to transmit data to the ICs at memory addresses from 000h to 1FFh. Addresses 000h to 0FFh and 180h and 1FFh can be written as a block. Addesses 100h to 17Fh must be written one word at a time. The memory address is sent by the bus master as a single byte value immediately after the slave address. The MSB of the data to be stored is written immediately after the memory address byte is acknowledged. Because the address is automatically incremented after the least significant bit (LSb) of each word received by the IC, the MSB of the data at the next memory address can be written immediately after the acknowledgment of the LSB of data at the previous address. The master indicates the end of a write transaction by sending a STOP or Repeated START after receiving the last acknowledge bit. If the bus master continues an autoincremented write transaction beyond address 0FFh or 1FFh, the ICs ignore the data. Data is also ignored on writes to read-only addresses but not reserved addresses. Do not write to reserved address locations. See Figure 73 for an example write data communication sequence.

### I²C Read Data Protocol

The read data protocol is used to transmit data from IC memory locations 000h to 1FFh . Addresses 000h to 0FFh and 180h to 1FFh can be read as a block. Addresses 100h to 17Fh must be read as individual words. The memory address is sent by the bus master as a single byte value immediately after the slave address. Immediately following the memory address, the bus master issues a REPEATED START followed by the slave address. The MAX17201/MAX17205 acknowlege the address and begin transmitting data. A word of data is read as two separate bytes that the master must ACK. Because the address is automatically incremented after the least significant bit (LSb) of each word received by the IC, the MSB of the data at the next memory address can be read immediately after the acknowledgment of the LSB of data at the previous address. The master indicates the end of a read transaction by sending a NACK followed by a STOP. If the bus master continues an autoincremented read transaction beyond memory address 0FFh or 1FFh, the IC transmits all 1s until a NACK or STOP is received. Data from reserved address locations is undefined. See Figure 74 for an example read data communication sequence.

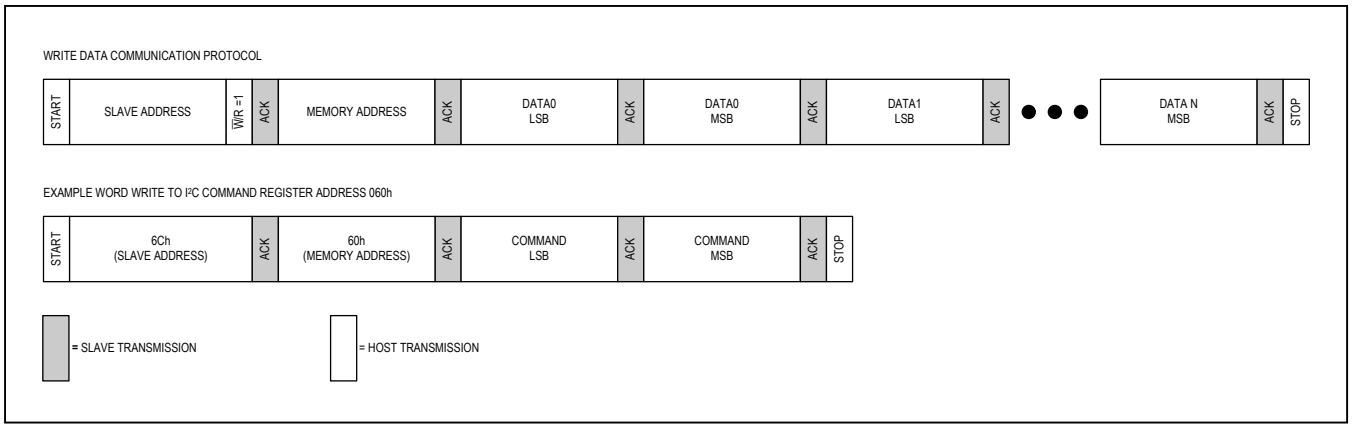



*Figure 73. Example I²C Write Data Communication Sequence*

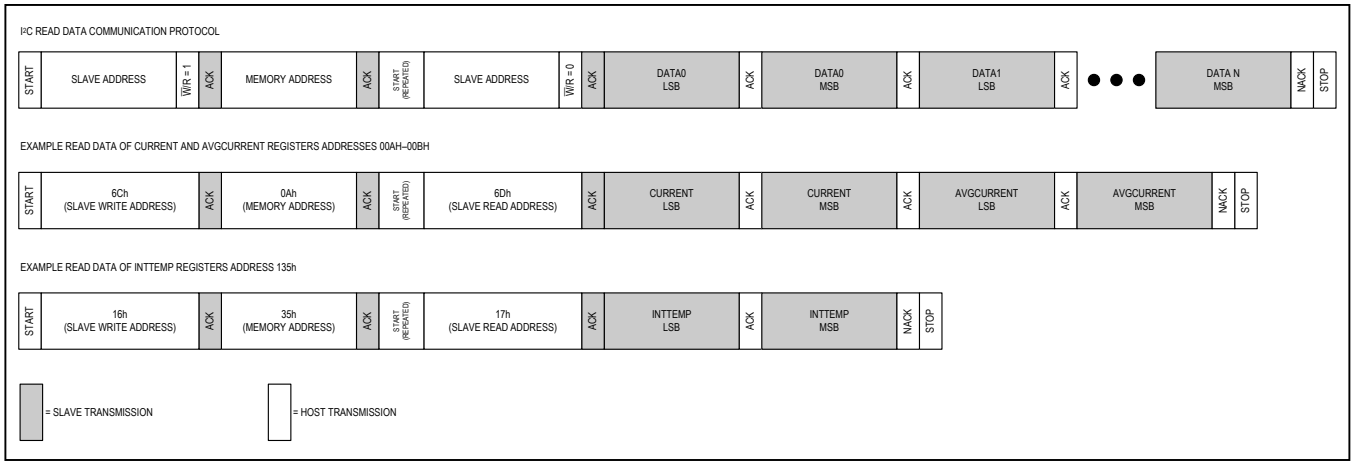



*Figure 74. Example I²C Read Data Communication Sequence*

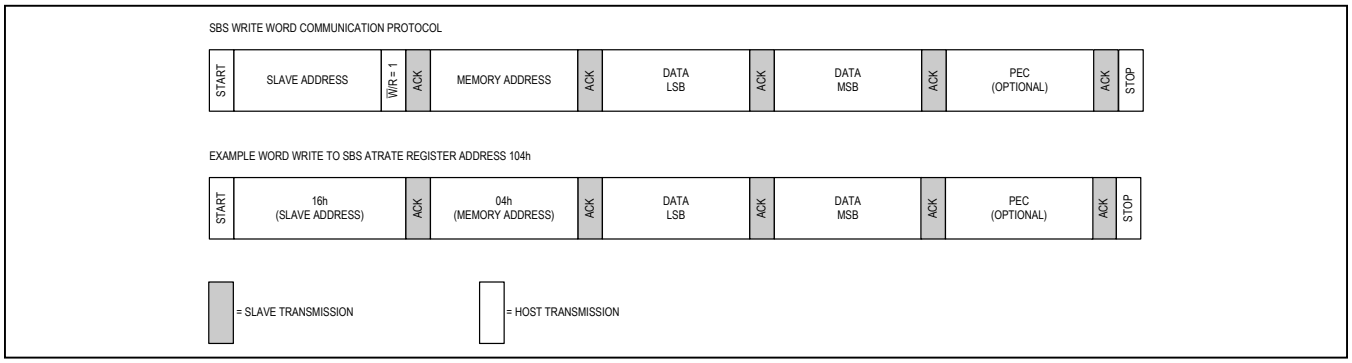



*Figure 75. Example SBS Write Word Communication Sequence*

## SBS Protocols

The following 2-wire communication protocols must be used by the bus master to access MAX17201/MAX17205 memory locations 100h to 17Fh. These protocols follow the smart battery specification for communication.

### SBS Write Word Protocol

The Write Word protocol is used to transmit data to the IC memory addresses between 100h and 17Fh that do not require the write block protocol. The memory address is sent by the bus master as a single-byte LSB value immediately after the slave address, the MSb of the address is omitted. The LSB of the data to be stored is written immediately after the memory address byte is acknowledged, followed by the MSB. A PEC byte can follow the data word, but the data word is written without checking the validity of the PEC. The master indicates the end of a write transaction by sending a STOP or Repeated START after receiving the last acknowledge bit. Data is ignored on writes to read-only addresses but not reserved addresses. Do not write to reserved address locations. The write word protocol should not be used to write to addresses supported by the write block protocol, use write block at these locations instead. See Figure 75 for an example Write Word communication sequence.

### SBS Read Word Protocol

The Read Word protocol is used to read data from the IC at memory addresses between 100h and 17Fh. The memory address is sent by the bus master as a single byte LSB value immediately after the slave address, the MSb of the address is ignored. The LSB of the data is read immediately after the memory address byte is acknowledged, followed by the MSB. A PEC byte follows the data word. The master indicates the end of a write transaction by sending a STOP or repeated START after not acknowledging the last received byte. Data from reserved address locations is undefined. The read word protocol should not be used to read from addresses supported by the read block protocol, use read block at these locations instead. See Figure 76 for an example read word communication sequence.

### SBS Write Block Protocol

The SBS Write Block protocol is not supported by the MAX17201/MAX17205. Use the write data command sequence to the corresponding nonvolatile memory locations to update write/read block register locations. See Table 24.

### SBS Read Block Protocol

The Read Block protocol is similar to the read word protocol except the master reads multiple words of data at once. A data size byte is transmitted by the ICs immediately after the memory address byte and before the first byte of data to be read. The read block protocol is only supported at the register locations shown in Table 28. PEC error checking is provided by the read block protocol if nNVCfg0.enSBS = 1. Figure 77 shows an example Read Block communication sequence.

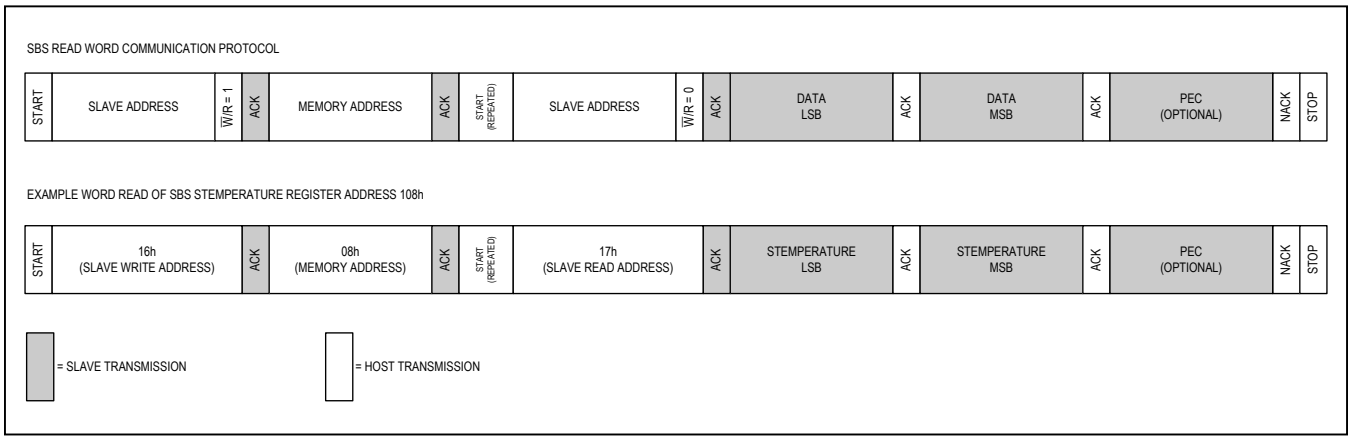



*Figure 76. Example SBS Read Word Communication Sequence*

## Table 28. Valid SBS Read Block Registers

| ADDRESS | REGISTER | SIZE BYTE MAX VALUE | FORMAT |
|---|---|---|---|
| 0120h | sManfctName | 0Ah | ASCII String |
| 0121h | sDeviceName | 0Ch | ASCII String |
| 0122h | sDevChemistry | 05h | ASCII String |
| 0123h | sManfctData | 1Ah | Hexadecimal |
| 011Ch | sSerialNumber | 08h | Hexadecimal |
| 0170h | sManfctInfo | 18h | Hexadecimal |

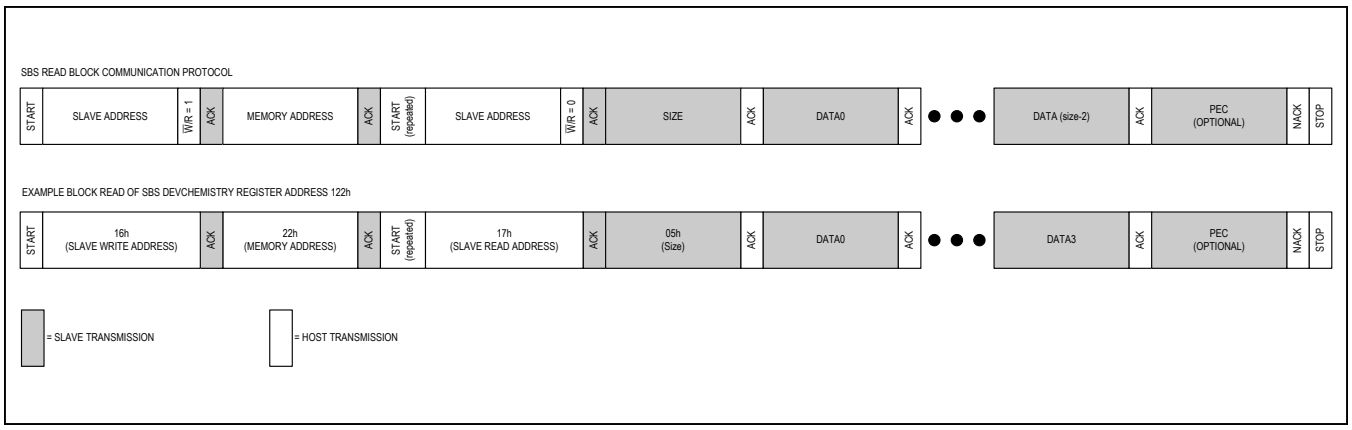



*Figure 77. Example SBS Read Block Communication Sequence*

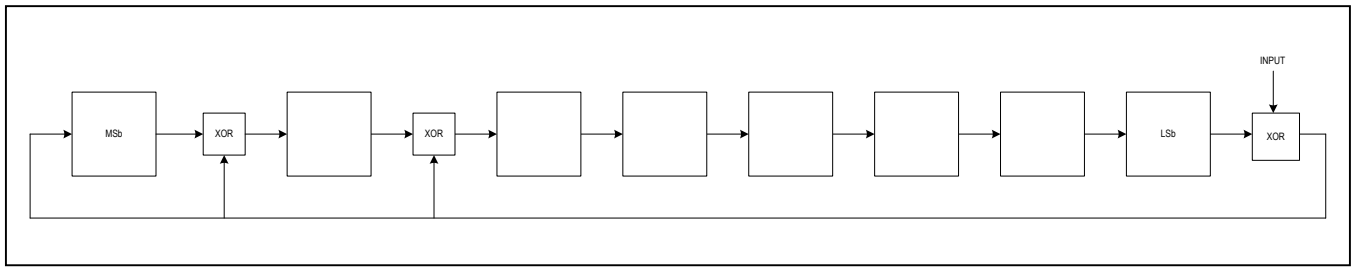



*Figure 78. PEC CRC Generation Block Diagram*

### Packet Error Checking

SBS read functions support packet error checking (PEC) if nNVCfg0.enSBS is enabled. The host system is responsible for verifying the CRC value it receives and taking action as a result. SBS write functions accept a PEC byte but complete the write function regardless of the value of the PEC.

The CRC can be generated by the host using a circuit consisting of a shift register and XOR gates as shown in Figure 78, or it can be generated in software using the polynomial $X^8 + X^2 + X^1 + 1$. See the *Smart Battery Compliant Operation* section for more information.

### 1-Wire Bus System (MAX17211/MAX17215 Only)

The MAX17211/MAX17215 communicate to a host through a Maxim 1-Wire interface. The 1-Wire bus is a system that has a single bus master and one or more slaves. A multidrop bus is a 1-Wire bus with multiple slaves, while a single-drop bus has only one slave device. In all instances, these ICs are slave devices. The bus master is typically a microprocessor in the host system. The discussion of this bus system consists of five topics: 64-bit net address, CRC generation, hardware configuration, transaction sequence, and 1-Wire signaling.

### Hardware Configuration

Because the 1-Wire bus has only a single line, it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1-Wire bus must connect to the bus with open-drain or three-state output drivers. The MAX17211/MAX17215 use an open-drain output driver as part of the bidirectional interface circuitry shown in Figure 79. If a bidirectional pin is not available on the bus master, separate output and input pins can be connected together. Communication speed is controlled by the OD/SCL pin. Connect OD/SCL to PACK- to enable communication at standard speed. Connect OD/SCL to the REG3 pin to enable communication at overdrive speed.

The 1-Wire bus must have a pull-up resistor on the host side of the bus. A value of between 2kΩ and 5kΩ is recommended for most applications. The idle state for the 1-Wire bus is logic high. If, for any reason, a bus transaction must be suspended, the bus must be left in the idle state to properly resume the transaction later. Note that if the bus is left low for more than $t_{LOW0}$, slave devices on the bus begin to interpret the low period as a reset pulse, effectively terminating the transaction.

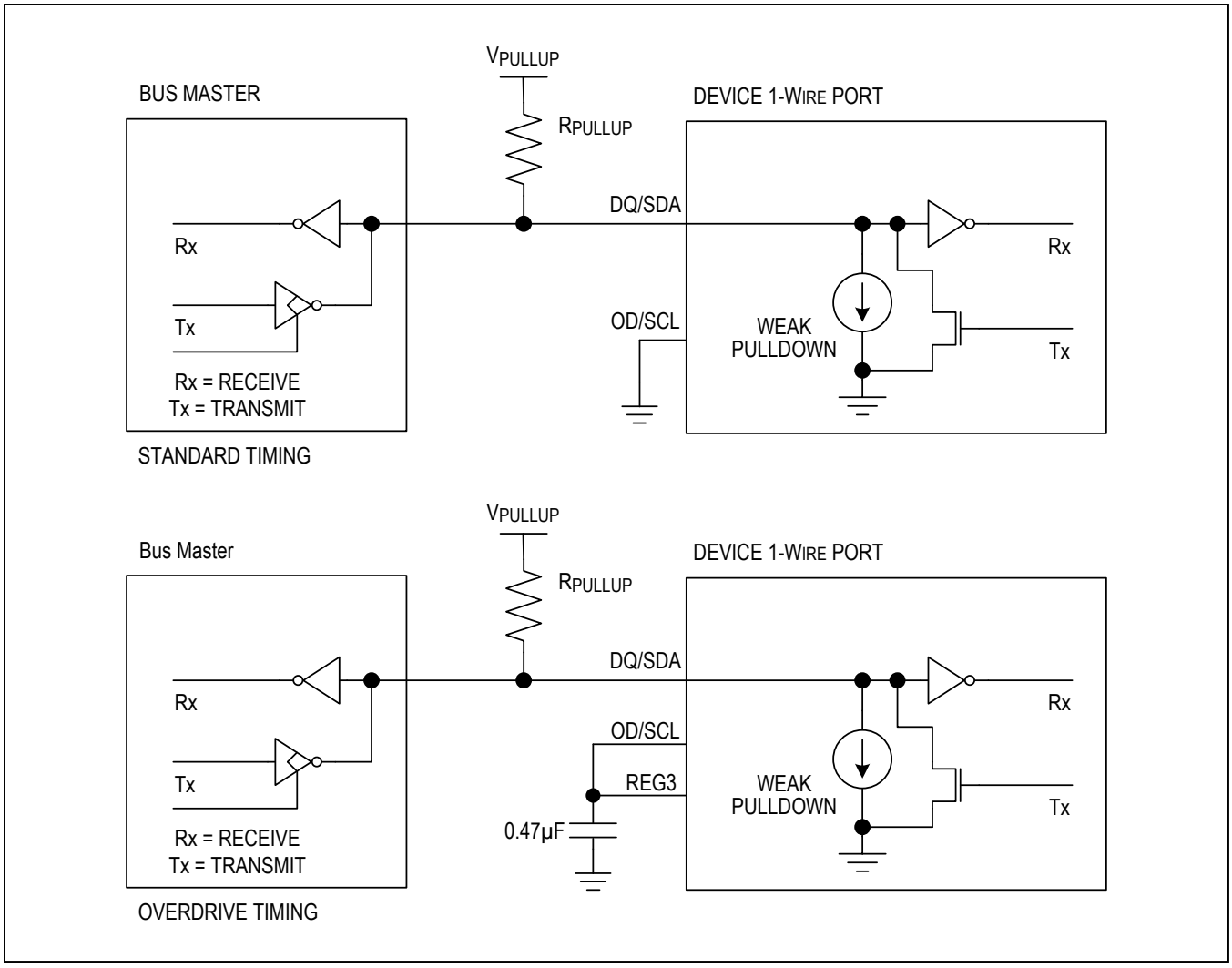



*Figure 79. 1-Wire Bus Interface Circuitry*

### 64-Bit Net Address (ROM ID)

The 1-Wire net address is 64 bits in length. The term net address is synonymous with the ROM ID or ROM code terms used in other 1-Wire documentation. The value of the net address is stored in nonvolatile memory and cannot be changed. In a 1-wire standard net address, the first eight bits of the net address are the 1-Wire family code. This value is the same for all ICs of the same type. The next 48 bits are a unique serial number. The last eight bits are a cyclic redundancy check (CRC) of the first 56 bits. Table 29 details the net address data format. The 64-bit net address and the 1-Wire I/O circuitry built into the device enable the MAX17211/MAX17215 to communicate through the 1-Wire protocol detailed in this data sheet.

### I/O Signaling

The 1-Wire bus requires strict signaling protocols to ensure data integrity. The four protocols used by the MAX17211/MAX17215 are as follows: the initialization sequence (reset pulse followed by presence pulse), write 0, write 1, and read data. The bus master initiates all signaling except for the presence pulse.

### Reset Time Slot

The initialization sequence required to begin any communication with the MAX17211/MAX17215 is shown in Figure 80. The bus master transmits (Tx) a reset pulse for $t_{RSTL}$. The bus master then releases the line and goes into receive mode (Rx). The 1-Wire bus line is then pulled high by the pullup resistor. After detecting the rising edge on the DQ pin, the MAX17211/MAX17215 waits for $t_{PDH}$ and then transmits the presence pulse for $t_{PDL}$. A presence pulse following a reset pulse indicates that the MAX17211/MAX17215 is ready to accept a net address command.

### Write Time Slots

## Table 29. 1-Wire Net Address Format

| MSB: 8-BIT CRC | 48-BIT SERIAL NUMBER | LSB: 8-BIT FAMILY CODE (26H) |
|---|---|---|

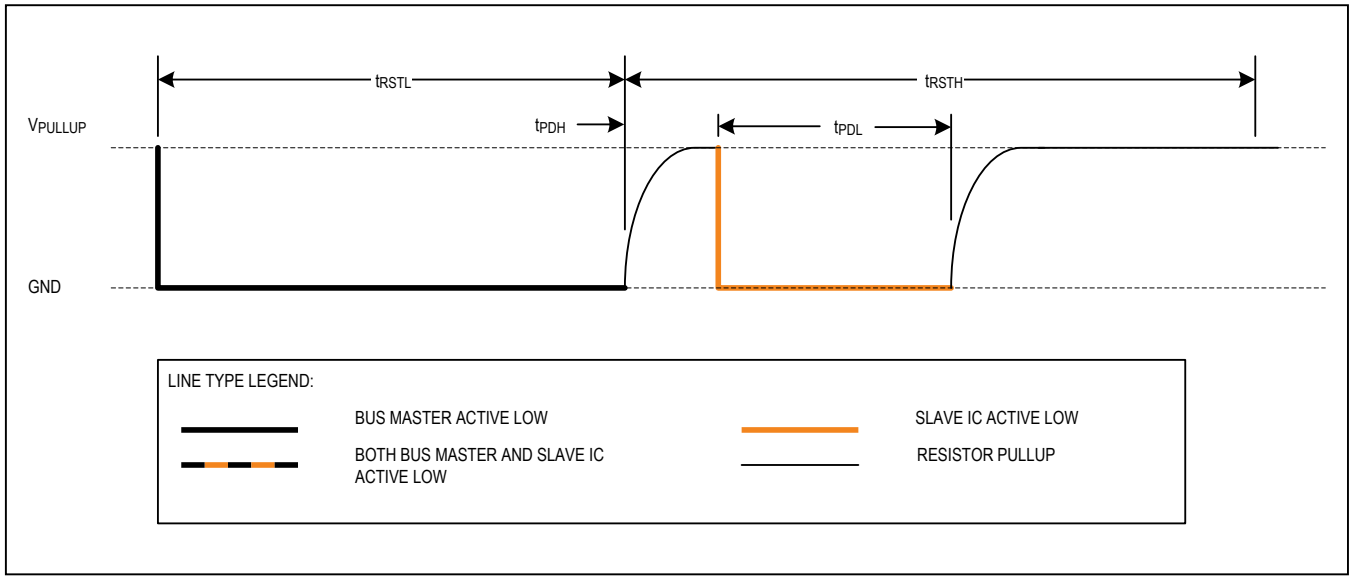



*Figure 80. 1-Wire Initialization Sequence*

A write-time slot is initiated when the bus master pulls the 1-Wire bus from a logic-high (inactive) level to a logic-low level. There are two types of write-time slots: write 1 and write 0. All write-time slots must be $t_{SLOT}$ in duration with a 1µs minimum recovery time, $t_{REC}$, between cycles. The MAX17211/MAX17215 samples the 1-Wire bus line between $t_{LOW1_MAX}$ and $t_{LOW0_MIN}$ after the line falls. If the line is high when sampled, a write 1 occurs. If the line is low when sampled, a write 0 occurs. The sample window is illustrated in Figure 81. For the bus master to generate a write-1 time slot, the bus line must be pulled low and then released, allowing the line to be pulled high less than $t_{RDV}$ after the start of the write time slot. For the host to generate a write 0 time slot, the bus line must be pulled low and held low for the duration of the write-time slot.

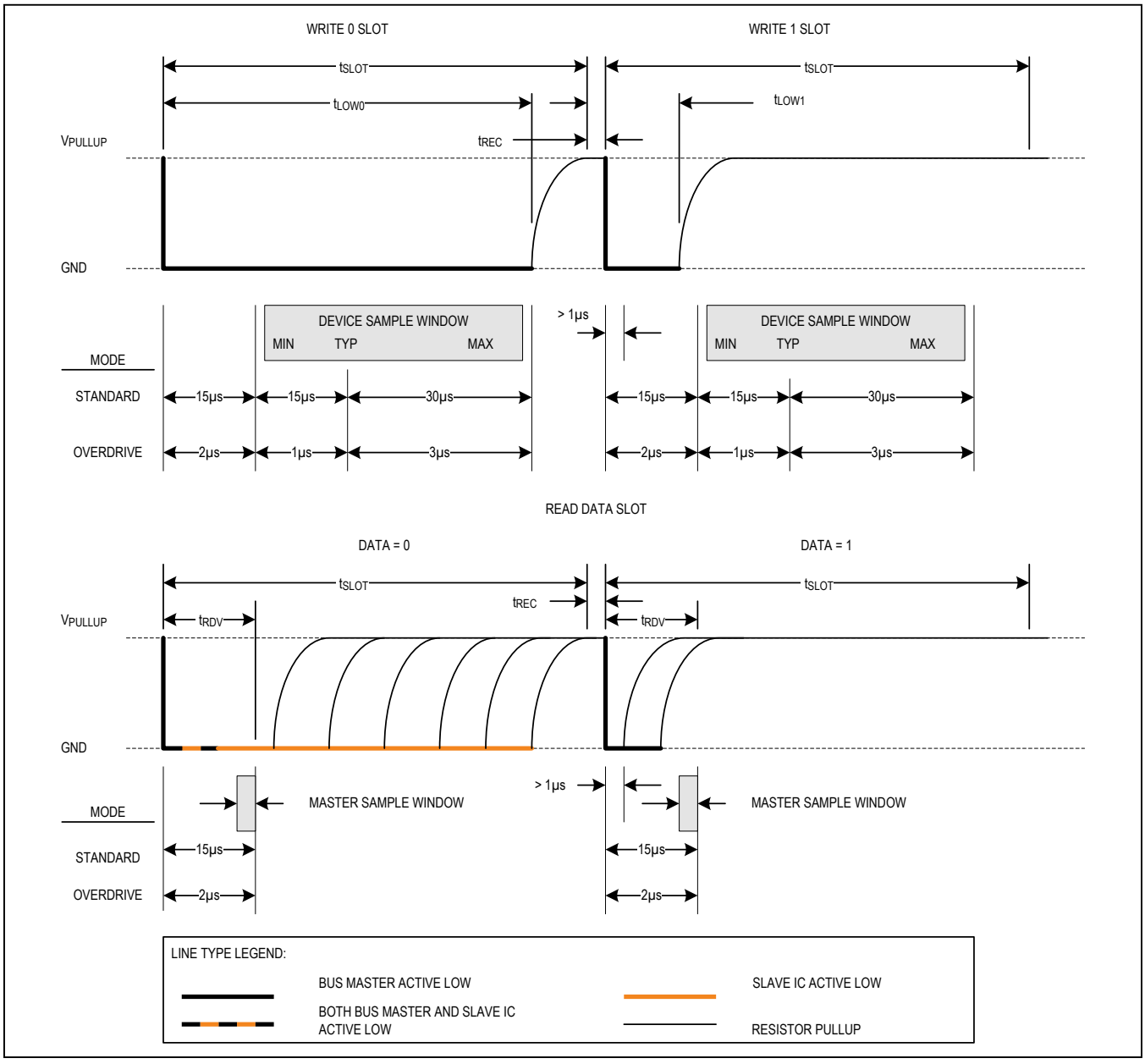



*Figure 81. 1-Wire Write and Read Time Slots*

### Read Time Slots

A read-time slot is initiated when the bus master pulls the 1-Wire bus line from a logic-high level to a logic-low level. The bus master must keep the bus line low for at least 1µs and then release it to allow the MAX17211/15 to present valid data. The bus master can then sample the data $t_{RDV}$ from the start of the read-time slot. By the end of the read-time slot, the MAX17211/15 releases the bus line and allows it to be pulled high by the external pullup resistor. All read-time slots must be $t_{SLOT}$ in duration with a 1µs minimum recovery time, $t_{REC}$, between cycles. See Figure 81 and the timing specifications in the *Electrical Characteristics* table for more information.

### Transaction Sequence

The protocol for accessing the MAX17211/MAX17215 through the 1-Wire port is as follows:

- Initialization
- Net address command
- Function command(s)
- Data transfer (not all commands have data transfer)

### Net Address Commands

Once the bus master has detected the presence of one or more slaves, it can issue one of the net address commands described in the following paragraphs. The name of each net address command (ROM command) is followed by the 8-bit op code for that command in square brackets.

### Read Net Address [33h]

This command allows the bus master to read the MAX17211/MAX17215's 1-Wire net address. This command can only be used if there is a single slave on the bus. If more than one slave is present, a data collision occurs when all slaves try to transmit at the same time (open-drain produces a wired-AND result).

### Match Net Address [55h]

This command allows the bus master to specifically address one MAX17211/MAX17215 on the 1-Wire bus. Only the addressed MAX17211/MAX17215 respond to any subsequent function command. All other slave devices ignore the function command and wait for a reset pulse. This command can be used with one or more slave devices on the bus.

### Skip Net Address [CCh]

This command saves time when there is only one MAX17211/15 on the bus by allowing the bus master to issue a function command without specifying the address of the slave. If more than one slave device is present on the bus, a subsequent function command can cause a data collision when all slaves transmit data at the same time.

### Search Net Address [F0h]

This command allows the bus master to use a process of elimination to identify the 1-Wire net addresses of all slave devices on the bus. The search process involves the repetition of a simple three-step routine: read a bit, read the complement of the bit, then write the desired value of that bit. The bus master performs this simple three-step routine on each bit location of the net address. After one complete pass through all 64 bits, the bus master knows the address of one device. The remaining devices can then be identified on additional iterations of the process. Refer to Chapter 5 of the *Book of iButton® Standards* for a comprehensive discussion of a net address search, including an actual example.

### 1-Wire Functions

After successfully completing one of the net address commands, the bus master can access the features of the MAX17211/MAX17215 with either a read data or write data function command described in the following paragraphs. Any other IC operation such as a compute MAC operation is accomplished by writing to the COMMAND register. See the *Summary of Commands* section for details.

*iButton is a registered trademark of Maxim Integrated Products, Inc.*

### Read Data [69h, LL, HH]

This command reads data from the MAX17211/MAX17215 starting at memory address HHLL. Any memory address from 0000h to 01FFh is a valid starting address. The LSb of the data in address HHLL is available to be read immediately after the MSb of the address has been entered. Because the address is automatically incremented after the MSb of each byte is received, the LSb of the data at address HHLL+ 1 is available to be read immediately after the MSb of the data at address HHLL. If the bus master continues to read beyond address 01FFh, data is will be undefined. Addresses labeled reserved in the memory map contain undefined data values. The read data command can be terminated by the bus master with a reset pulse at any bit boundary. Reads from nonvolatile memory addresses return the data in the shadow RAM. A recall data command is required to transfer data from nonvolatile memory to the shadow RAM. See the *Summary of Commands* section for more details. See Figure 82 for an example read data communication sequence.

### Write Data [6Ch, LL, HH]

This command writes data to the MAX17211/15 starting at memory address HHLL. Any memory address from 0000h to 01FFh is a valid starting address. The LSb of the data to be stored at address HHLL can be written immediately after the MSb of address has been entered. Because the address is automatically incremented after the MSb of each byte is written, the LSb to be stored at address HHLL + 1 can be written immediately after the MSb to be stored at address HHLL. If the bus master continues to write beyond address 01FFh, the data is ignored by the IC. Writes to read-only addresses and locked memory blocks are ignored. Do not write to RESERVED address locations. Incomplete bytes are not written. Writes to unlocked nonvolatile memory addresses modify the shadow RAM. A Copy NV Block command is required to transfer data from the shadow RAM to nonvolatile memory. See the *Summary of Commands* section for more details. See Figure 82 for an example Write Data communication sequence.

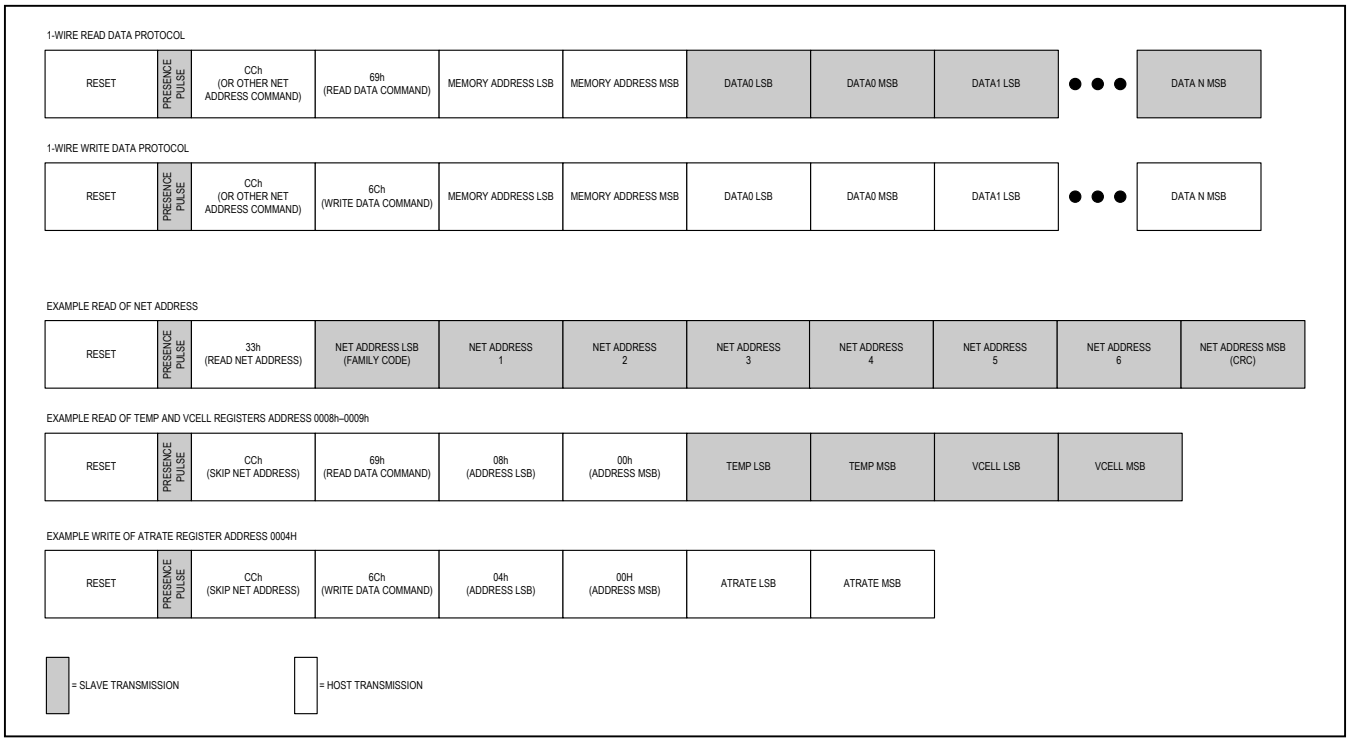



*Figure 82. Example 1-Wire Communication Sequences*

### Summary of Commands

Any operation other than writing or reading a memory location is executed by writing the appropriate command to the Command or Config2 registers. Table 30 lists all function commands understood by the MAX1720x/MAX1721x. For both 1-wire and 2-wire communication, the function command must be written to the Command (060h) or Config2 (0BBh) registers. Device commands are described in detail in their respective sections of the data sheet.

## Table 30. All Function Commands

| COMMAND | TYPE | REGISTER | HEX | DESCRIPTION |
|---|---|---|---|---|
| Compute MAC Without ROM ID | SHA | 060h | 3600h | Computes hash operation of the message block with logical 1s in place of the ROM ID. |
| Compute MAC with ROM ID | SHA | 060h | 3500h | Computes hash operation of the message block including the ROM ID. |
| Compute Next Secret Without ROM ID | SHA | 060h | 3000h | Computes hash operation of the message block with logical 1s in place of the ROM ID. The result is then stored as the new secret. |
| Compute Next Secret with ROM ID | SHA | 060h | 3300h | Computes hash operation of the message block including the ROM ID. The result is then stored as the new secret. |
| Clear Secret | SHA | 060h | 5A00h | Resets the SHA-256 Secret to a value of all 0s |
| Lock Secret | SHA | 060h | 6000h | Permanently locks the SHA-256 Secret. |
| Copy NV Block | Memory | 060h | E904h | Copies all shadow RAM locations to nonvolatile memory at the same time. |
| NV Recall | Memory | 060h | E001h | Recalls all nonvolatile memory to RAM. |
| History Recall | Memory | 060h | E2XXh | Recalls a page of nonvolatile memory history into RAM page 1Eh. |
| NV Lock | Memory | 060h | 6AXXh | Permanently locks an area of memory. See the lock section for details. |
| Hardware Reset | Reset | 060h | 000Fh | Recalls nonvolatile memory into RAM and resets the IC hardware. Fuel gauge operation is not reset. |
| Fuel Gauge Reset | Reset | 0BBh | 0001h | Restarts the fuel gauge operation without affecting nonvolatile shadow RAM settings. |

## Appendix A: Reading History Data Psuedo-Code Example

The following psuedo-code can be used as a reference for reading history data from the IC. The code first reads all flag information, tests all flag information, then reads all valid history data into a two-dimensional array. Afterwards, the HistoryLength variable will indicate the depth of the history array data.

```
Int WriteFlags[26];
Int ValidFlags[26];
Boolean PageGood[203];
Int HistoryData[203][16];
Int HistoryLength;
Int word, position, flag1, flag2, flag3, flag4;
//Read all flag information from the IC
WriteCommand(0xE2FB);
Wait(tRECALL);
WriteFlags[0] = ReadData(0x0E1);
WriteFlags[1] = ReadData(0x0E2);
WriteFlags[2] = ReadData(0x0E3);
WriteFlags[3] = ReadData(0x0E4);
WriteFlags[4] = ReadData(0x0E5);
WriteFlags[5] = ReadData(0x0E6);
WriteFlags[6] = ReadData(0x0E7);
WriteFlags[7] = ReadData(0x0E8);
WriteFlags[8] = ReadData(0x0E9);
WriteFlags[9] = ReadData(0x0EA);
WriteFlags[10] = ReadData(0x0EB);
WriteFlags[11] = ReadData(0x0EC);
WriteFlags[12] = ReadData(0x0ED);
WriteFlags[13] = ReadData(0x0EE);
WriteFlags[14] = ReadData(0x0EF);
WriteCommand(0xE2FC);
Wait(tRECALL);
WriteFlags[15] = ReadData(0x0E0);
WriteFlags[16] = ReadData(0x0E1);
WriteFlags[17] = ReadData(0x0E2);
WriteFlags[18] = ReadData(0x0E3);
WriteFlags[19] = ReadData(0x0E4);
WriteFlags[20] = ReadData(0x0E5);
WriteFlags[21] = ReadData(0x0E6);
WriteFlags[22] = ReadData(0x0E7);
WriteFlags[23] = ReadData(0x0E8);
```

```
WriteFlags[24] = ReadData(0x0E9);
WriteFlags[25] = ReadData(0x0EA);
ValidFlags[0] = ReadData(0x0EB);
ValidFlags[1] = ReadData(0x0EC);
ValidFlags[2] = ReadData(0x0ED);
ValidFlags[3] = ReadData(0x0EE);
ValidFlags[4] = ReadData(0x0EF);
WriteCommand(0xE2FD);
Wait(tRECALL);
ValidFlags[5] = ReadData(0x0E0);
ValidFlags[6] = ReadData(0x0E1);
ValidFlags[7] = ReadData(0x0E2);
ValidFlags[8] = ReadData(0x0E3);
ValidFlags[9] = ReadData(0x0E4);
ValidFlags[10] = ReadData(0x0E5);
ValidFlags[11] = ReadData(0x0E6);
ValidFlags[12] = ReadData(0x0E7);
ValidFlags[13] = ReadData(0x0E8);
ValidFlags[14] = ReadData(0x0E9);
ValidFlags[15] = ReadData(0x0EA);
ValidFlags[16] = ReadData(0x0EB);
ValidFlags[17] = ReadData(0x0EC);
ValidFlags[18] = ReadData(0x0ED);
ValidFlags[19] = ReadData(0x0EE);
ValidFlags[20] = ReadData(0x0EF);
WriteCommand(0xE2FE);
Wait(tRECALL);
ValidFlags[21] = ReadData(0x0E0);
ValidFlags[22] = ReadData(0x0E1);
ValidFlags[23] = ReadData(0x0E2);
ValidFlags[24] = ReadData(0x0E3);
ValidFlags[25] = ReadData(0x0E4);
//Determine which history pages contain valid data
For loop = 0 to 202
{
   word = (int)( loop/8 );
   position = loop % 8 ; //remainder
   flag1 = (WriteFlags[word] >> position) & 0x0001;
   flag2 = (WriteFlags[word] >> (position+8)) & 0x0001;
```

```
    flag3 = (ValidFlags[word] >> position) & 0x0001;
    flag4 = (ValidFlags[word] >> (position+8)) & 0x0001;
    if (flag1 || flag2) && (flag3 || flag4)
      PageGood[loop] = True;
    else
      PageGood[loop] = False;
}
//Read all the history data from the IC
HistoryLength = 0;
For loop = 0 to 202
{
    if(PageGood[loop]) == TRUE
    {
      SendCommand(0xE226 + loop);
      Wait(tRECALL);
      HistoryData[HistoryLength][0] = ReadData(0x0E0);
      ...
      HistoryData[HistoryLength][15] = ReadData(0x0EF);
      HistoryLength++;
    }
}
```

## Appendix B: Layout Guidelines

Proper circuit layout as shown in Figure 83 is essential for voltage, temperature, and current measurement accuracy. The recommended layout guidelines are as follows:

1) CSN and CSP traces should make Kelvin connections to the sense resistor. Current is measured differentially through the CSN and CSP pins. Any shared high-current paths on these traces will affect current measurement gain accuracy.

2) Connect the exposed pad directly to the CSN pin.

3) REG2 and REG3 capacitor trace loop area should be minimized. They should be connected to the CSN pin as close as possible to the IC. Run only a single CSN trace to the sense resistor. The helps filter any noise from the internal regulated supply.

4) All other ground connections should be kept separate from the CSN trace and connect at PACK- away from the sense resistor.

5) There are no limitations on any other IC connection. Other IC pins as well as any external components mounted to these pins have no special layout requirements.

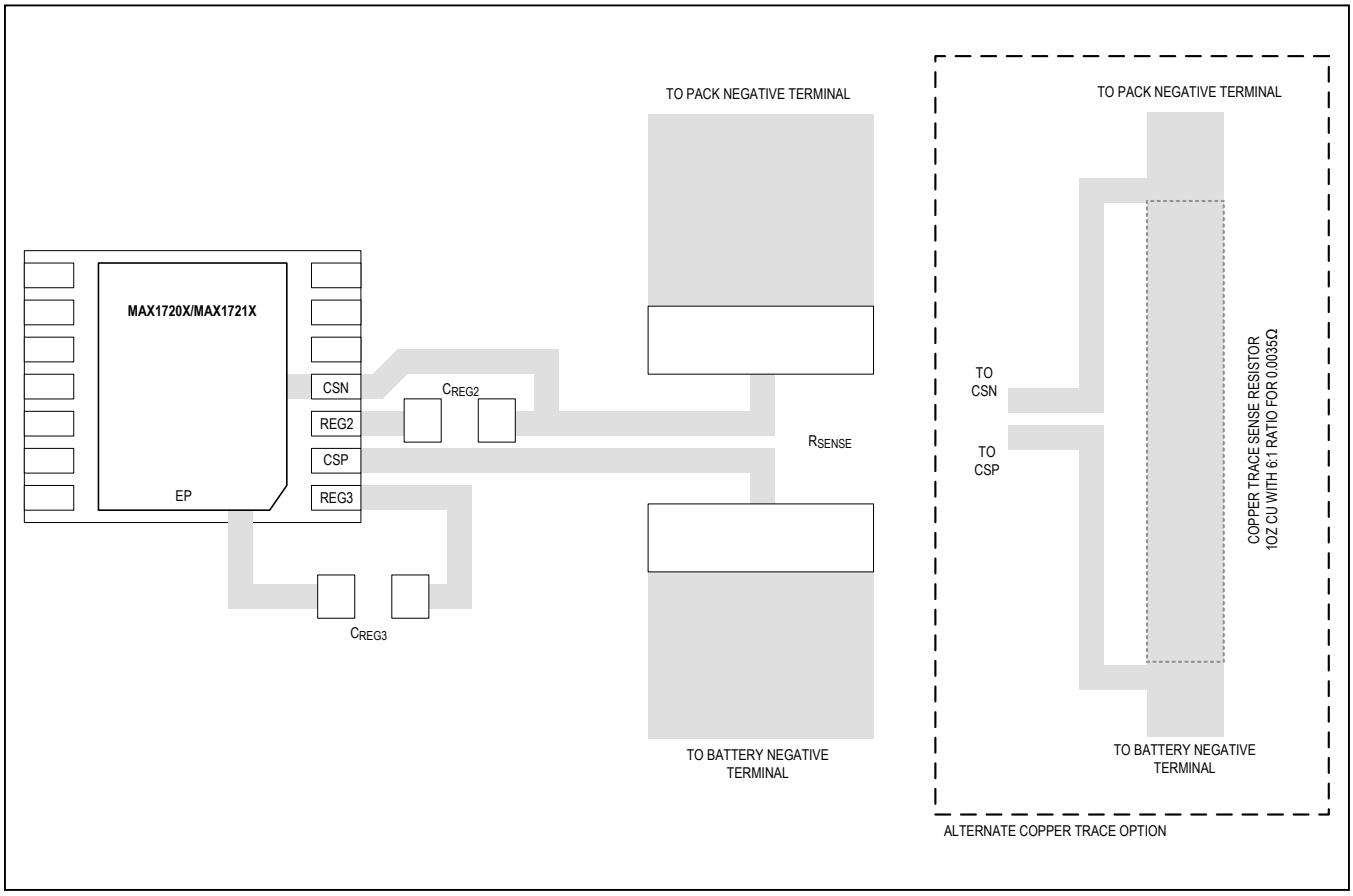



*Figure 83. Sample Board Layout*

## Ordering Information

| PART | CELL COUNT | INTERFACE | TEMP RANGE | PIN-PACKAGE |
|---|---|---|---|---|
| MAX17201G+ | 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17201G+T | 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17201G+00E** | 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17201G+T0E** | 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17205G+ | > 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17205G+T | > 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17205G+00E** | > 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17205G+T0E** | > 1S | I²C | -40°C to +85°C | 14 TDFN-EP* |
| MAX17211G+ | 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17211G+T | 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17211G+00E** | 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17211G+T0E** | 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17215G+ | > 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17215G+T | > 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17215G+00E** | > 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |
| MAX17215G+T0E** | > 1S | 1-Wire | -40°C to +85°C | 14 TDFN-EP* |

*+Denotes a lead(Pb)-free/RoHS-compliant package.*

*T = Tape and reel.*

**EP = Exposed pad.*

***00E or T0E indicates IC supports EZ performance model only. No custom battery characterization.*

## Revision History

| REVISION NUMBER | REVISION DATE | DESCRIPTION | PAGES CHANGED |
|---|---|---|---|
| 0 | 2/16 | Initial release | — |

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

*Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.*

Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc. © 2016 Maxim Integrated Products, Inc. 