ANALOG DEVICES

AN-1306
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADP1050 and ADP1051 EEPROM Programming and Calibration in the Power Supply Manufacturing Environment

by Rico Huang and James Xie

INTRODUCTION

The ADP1050 and ADP1051 have a built in EEPROM controller that is used to communicate with the embedded 8000, 8-bit bytes EEPROM. The EEPROM is partitioned into two major blocks: the information block and the main block. The information block contains 128 8-bit bytes for internal use only, and the main block contains 8000, 8-bit bytes. The main block is further partitioned into 16 pages. Each page contains 512 bytes.

Page 0 of the main block is reserved for storing the default command settings. The default command settings are preprogrammed in the ADP1050 and ADP1051 production lines. Page 1 of the main block is reserved for storing user command settings. During the manufacturing of the power supply, the user command settings are programmed in this page. The programming of commands includes two main steps: program the user command settings from the HEX file to the registers and then save the user command settings from the registers to Page 1 of the EEPROM using the STORE_USER_ALL command.

Page 3 is reserved for storing the power board parameter settings (board settings) accessing of the ADP1050/ADP1051 GUI software. The programming of the board parameter settings is optional. The user (in particular, the power supply designer) decides whether or not to program. Programming is conducted by a page write of the HEX file to Page 3. A page erase action is needed before the programming.

This application note specifies the EEPROM programming and calibration for the ADP1050/ADP1051 during power supply manufacturing. Firstly, the HEX file used for EEPROM programming is generated by the ADP1050/ADP1051 GUI software in the power supply development stage. Secondly, using the generated HEX file, the EEPROM in the ADP1050/ADP1051 can be programmed during power supply manufacturing via an EEPROM programmer. Finally, the calibration procedure is presented. This application note provides the hardware and software guidelines for EEPROM programming and calibration.

Full details are provided in the ADP1050 or ADP1051 data sheets; the data sheets should be consulted in conjunction with this application note.

TABLE OF CONTENTS

REVISION HISTORY

2/15—Rev. 0 to Rev. A

Changes to Introduction Section ... 1
Changes to Figure 1 Caption ... 3
Changes to Figure 3 and Figure 4 ... 6
Changes to Programming Procedures for the User Command Settings Section ... 8
Changes to Calibration and Test Procedures Section ... 11

5/14—Revision 0: Initial Version

GENERATING THE HEX FILE

Follow these steps to generate a HEX file:

1. When the debugging of the power supply prototype is completed, connect the GUI software to the ADP1050/ADP1051 devices on the power board.
2. After the chip password is unlocked, click **Generate a Hex File of the Commands and Board Settings** in the GUI software, as shown in Figure 1. By clicking this button, all user settings, including the user command settings and board settings, are stored in a HEX file. Indicate the folder where this HEX file is to be saved when the appropriate window appears.
3. Save the generated HEX file.

This HEX file, together with the bill of materials (BOM), printed circuit board (PCB) Gerber files, and other documents, is released to the power supply production line.

The HEX file is composed of the user command settings and board settings. The user command settings represent the user settings of the PMBus™ commands (from Register 0x01 to Register 0xFE6F). The board settings represent the settings of the power board parameters, including the following:

- Turn ratio of main transformer
- Topology type
- Rectifier type
- Output inductance and its DCR (single or dual stage)
- Output capacitance and its ESR (single or dual stage)
- Nominal input voltage
- Input voltage sense divider
- Turn ratio of the current transformer
- Input current sense resistor
- Nominal output current
- Output voltage sense divider
- Output current sense resister

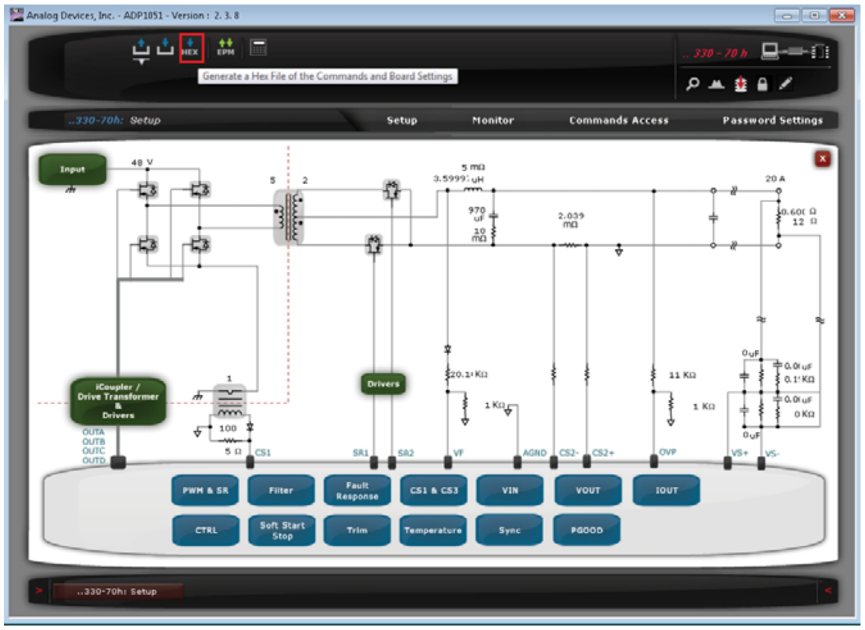
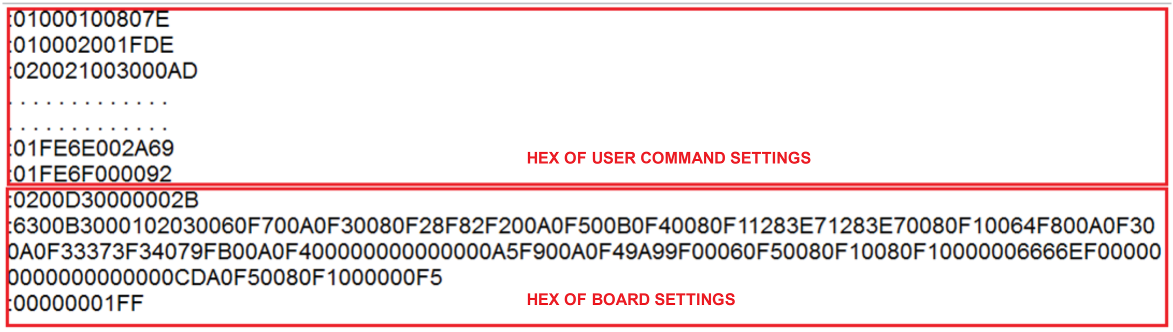


Figure 1. Analog Devices, Inc. – ADP1051 – Version: 2.3.8: Generating a HEX File

Figure 2 shows the HEX file for the user command settings (Register 0x01 to Register 0xFE6F) and board settings.

Example:

```
:01000100807E              // Register 0x01 value is 1-byte value 0x80.
:0200D30000002B          // Register 0xD3 value is word (2-byte) value 0x0000. In the
ADP1050/ADP1051, this means the EEPROM address offset of 0x0000.
:6300B3000102030060 // Page write to Page 3, totally 99 bytes (0x63).
:00000001FF                // Stop.
```



```
:01000100807E
:010002001FDE
:020021003000AD
. . . . . . . . . . . . .
. . . . . . . . . . . . .
:01FE6E002A69
:01FE6F000092
:0200D30000002B
:6300B3000102030060F700A0F30080F28F82F200A0F500B0F40080F11283E71283E70080F10064F800A0F30
0A0F33373F34079FB00A0F400000000000000A5F900A0F49A99F00060F50080F10080F10000006666EF00000
0000000000000CDA0F50080F1000000F5
:00000001FF
```

Figure 2. HEX File Example

EEPROM PROGRAMMING METHODS

There are two kinds of mainstream programming methods: programming by the automatic test equipment (ATE) setup after the ADP1050/ADP1051 devices are soldered, and programming by the EEPROM programmer before the device is soldered. You can use other programming methods, such as the system microcontroller, for the EEPROM programming.

METHOD 1: PROGRAMMING AFTER SOLDERING

Start by building or ordering an ATE setup. This setup provides the functions of EEPROM programming, trimming and calibration, and system test. The equipment includes the PC, a dc source, an electric load, voltage meters, current meters, fixtures, and test jigs. This equipment is connected by the communications interfaces like GPIB, PMBus, and so on. The programming software can be implemented by the LabVIEW®, C++, or other similar software. The procedures are as follows:

1. Do the PCB soldering and assembly of the power board. The ADP1050/ADP1051 devices are soldered on the power board.
2. Command the ATE setup to program the EEPROM in the device.
3. Command the ATE setup to turn on the dc source and turn on the power supply unit (PSU).
4. Do the calibration and test.

The calibration procedures are provided in the data sheets.

METHOD 2: PROGRAMMING BEFORE SOLDERING

Generally, you should order customized EEPROM programmers from a programming solutions provider, such as Data I/O Corporation. Alternatively, you can develop your own EEPROM programmer. The procedures are as follows:

1. Remove the ADP1050/ADP1051 devices from the package.
2. Program the EEPROM of the device (the programmer supports single part or multiple devices programming at the same time).
3. Return the devices to the tubes.
4. Solder and assemble.
5. Power up the power board and then perform the calibration and system test.

Note that if calibration is not needed, this method significantly increases the production efficiency.

REFERENCE SCHEMATIC FOR PROGRAMMING METHOD 2

The reference schematic for the EEPROM programming of the ADP1050 is shown in Figure 3. The reference schematic for the EEPROM programming of ADP1051 is shown in Figure 4. In both figures, J1 is the V_{DD} connector and J2 is the PMBus connector.

When referring to these schematics,

1. Place the decoupling capacitors of the VDD pin (C1 and C2) and the decoupling capacitor of the VCORE pin (C3) as close to the device as possible.
2. Connect Resistor R1 (10 kΩ, 0.1% tolerance) from the RES pin of the ADP1050/ADP1051 to the AGND pin.
3. The pull-up R6 and R7 resistors (4.7 kΩ) for the SDA and SCL signals and the R12 and R14 serial resistors (10 Ω) of SDA and SCL signals are optional. This increases the noise immunity of the PMBus communication.

For more information on the connection, see the ADP1050 or ADP1051 data sheets.

The electric specifications and PMBus communication timing information is provided in the ADP1050 and ADP1051 data sheets.

PMBUS SPECIFICATIONS AND TIMINGS

The PMBus protocol is built upon the system management bus (SMBus) which is built upon the 2-wire communication interface, I^2C. The PMBus dictates a maximum bus speed of 400 kHz and has built in timeouts important for critical systems. Unlike other standards, the PMBus defines a substantial number of domain specific commands rather than just saying how to communicate using user defined commands.

Follow the specifications and the PMBus communication timing diagram in the ADP1050 or ADP1051 data sheets.

REFERENCE SCHEMATICS

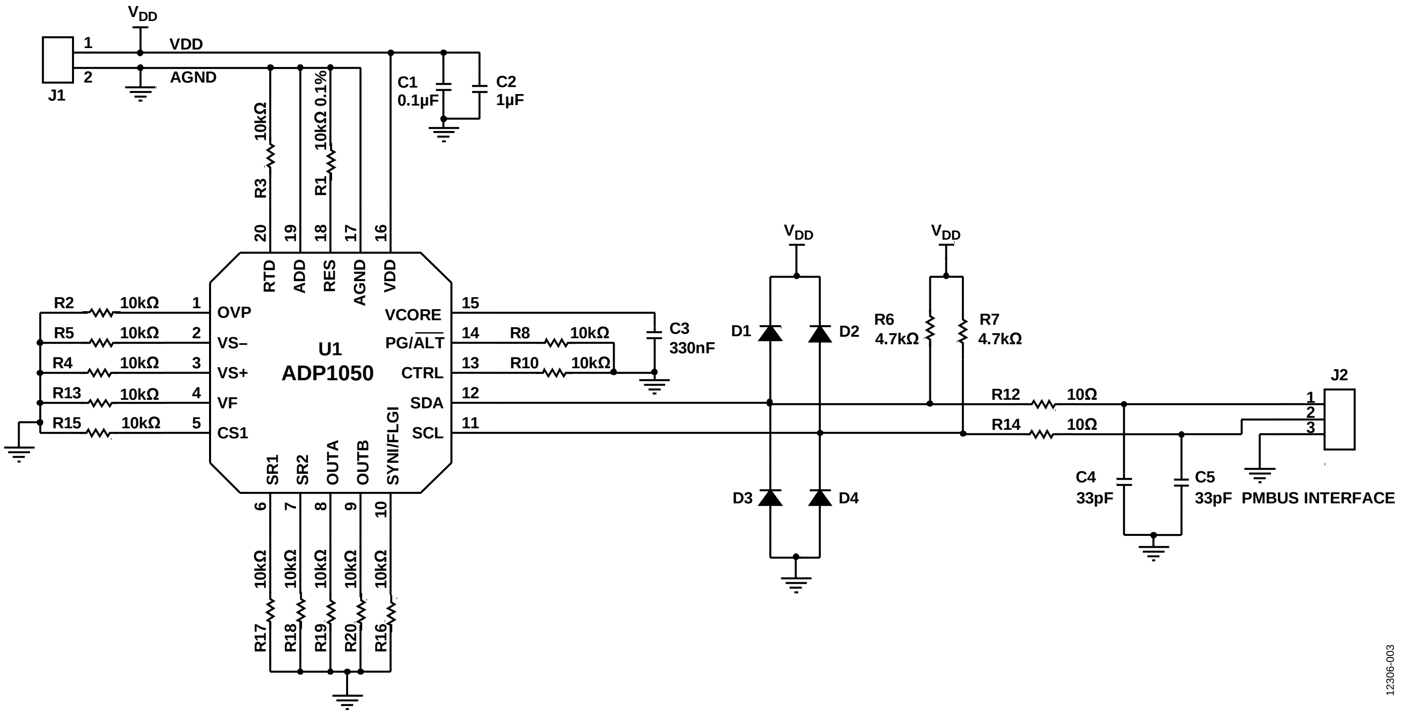


Figure 3. Reference Schematic for the ADP1050 EEPROM Programming

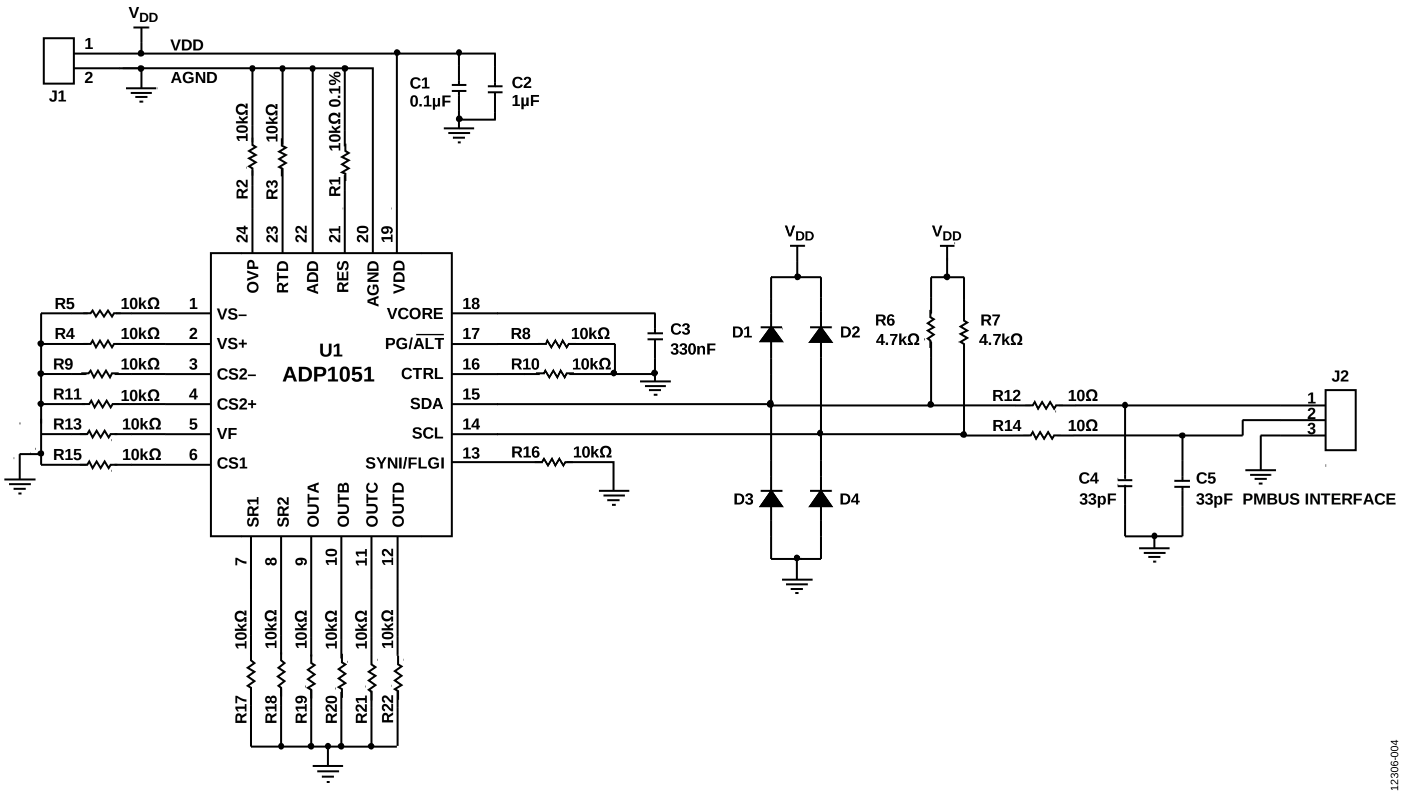


Figure 4. Reference Schematic for the ADP1051 EEPROM Programming

EEPROM PROGRAMMING

ELECTRIC SEQUENCING

Figure 5 shows the electric sequences for the EEPROM programming. The board settings are used for the access of the ADP1050 or ADP1051 GUI. Programming the board settings into the EEPROM is optional.

Programming Sequencing of the User Command Settings

The steps for programming the user command settings are as follows:

1. V_{DD} is powered up to 3.3 V. The SCL and SDA signals are pulled high from the low state with an interval of t_1 after V_{DD} reaches the UVLO level. There is no special requirement on the interval of t_1. After the SCL and SDA signals are pulled high for an interval of t_2, the PMBus communication starts.

   $t_1 + t_2 \geq 60$ ms

2. The PMBus communication occurs during the interval of t_3. All values in the HEX file of the user command settings are written into the registers of ADP1050/ADP1051 and the EEPROM is unlocked for access. The STORE_USER_ALL command (Register 0x15) is executed to save the user command settings to the EEPROM before t_3 ends.
3. There is no PMBus communication during the interval of t_4 to wait for the EEPROM erase operation.

   $t_4 \geq 40$ ms

4. The EEPROM is locked during the interval of t_5.
5. The SCL and SDA signals are pulled low after a delay of t_6. There is no special requirement for the interval of t_6.
6. The V_{DD} is powered down with a delay of t_7 after the SCL and SDA signals are pulled low. There is no special requirement on the interval of t_7. The target is to ensure that V_{DD} is powered down later than the SCL and SDA signals being pulled down.
7. The V_{DD} voltage and SCL and SDA signals keep low level for an interval of t_8.

   $t_8 = 10$ ms

8. At the end of t_8, V_{DD} is powered up. The V_{DD} keeps high level during t_9. There is no special requirement on the interval of t_9. After the SCL and SDA signals are pulled high for an interval of t_{10}, the PMBus communication starts.

   $t_9 + t_{10} \geq 60$ ms

9. During the interval of t_{11}, the register values are read back to check that the writing is correct.
10. The SCL and SDA signals keep high level during the interval of t_{12}. There is no special requirement on the interval of t_{12}.
11. V_{DD} is powered down with a delay of t_{13} after SCL and SDA signals are pulled down. There is no special requirement on the interval of t_{13}.

Programming Sequencing of Board Settings

The steps for programming the board settings are as follows:

1. The V_{DD}, SCL, and SDA voltage levels keep low level for an interval of t_{14}.

   $t_{14} = 10$ ms

2. Then, V_{DD} is powered up to 3.3 V. The SCL and SDA signals are pulled high from the low state with an interval of t_{15} after V_{DD} reaches the UVLO level. There is no special requirement on the interval of t_{15}. After the SCL and SDA signals are pulled high for an interval of t_{16}, the PMBus communication starts.

   $t_{15} + t_{16} \geq 60$ ms

3. The PMBus communication happens during the interval of t_{17}. Before t_{17} ends, the EEPROM_PAGE_ERASE command (Register 0xD4) is executed to erase Page 3.
4. There is no PMBus communication during the interval of t_{18} to wait for the EEPROM erase operation.

   $t_{18} \geq 40$ ms

5. The EEPROM write operation happens during t_{19}. At the end of t_{19}, the EEPROM is locked.
6. The SCL and SDA signals are pulled down after a delay of t_{20}. There is no special requirement on the interval of t_{20}.
7. V_{DD} is powered down with a delay of t_{21} after the SCL and SDA signals are pulled down. There is no special requirement on the interval of t_{21}.
8. The V_{DD} voltage and the SCL and SDA signals keep low level for an interval of t_{22}.

   $t_{22} = 10$ ms

9. At the end of t_{22}, V_{DD} is powered up. V_{DD} keeps high level during t_{23}. There is no special requirement on the interval of t_{23}. After the SCL and SDA signals are pulled high for an interval of t_{24}, the PMBus communication starts.

   $t_{23} + t_{24} \geq 60$ ms

10. During the interval of t_{25}, the values of the board settings are read back to check that the writing is correct.
11. The SCL and SDA signals keep high level during the interval of t_{26}. There is no special requirement on the interval of t_{26}.
12. V_{DD} is powered down with a delay of t_{27} after the SCL and SDA are pulled down. There is no special requirement on the interval of t_{27}.

PROGRAMMING PROCEDURES

Programming Procedures for the User Command Settings

1. Load the HEX file into the buffer of the programmer.
2. During the interval of t_3, execute the following actions:
   a. Write the default chip password 0xFFFF two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip.
   b. Check if Register 0xFEA0[7] = 1. If it is not 1, go back to Step a. If it is 1, go to Step c.
   c. This step is an optional step. Read the Analog Devices factory trim register values of the ADP1050/ADP1051 devices and store them in the buffer of the programmer. The target is to avoid the factory trim values being overwritten. However, there is a trim password, which can protect the factory trim values from being overwritten. These factory trim registers are 0xFE14, 0xFE15, 0xFE16, 0xFE17, 0xFE20, 0xFE28, 0xFE2A, 0xFE2B, 0xFE2C and 0xFE2D. Note that 0xFE15, 0xFE16, and 0xFE17 are applicable only to the ADP1051.
   d. Write the HEX file values of user command settings from the buffer into the corresponding registers.

      For the ADP1050, write the HEX file values of the user command settings from the buffer into the corresponding registers:

      0x01, 0x02, 0x21, 0x24, 0x25, 0x26, 0x27, 0x29, 0x2A, 0x33, 0x35, 0x36, 0x40, 0x41, 0x44, 0x45, 0x4F, 0x50, 0x5E, 0x5F, 0x60, 0x61, 0x64, 0x99, 0x9A, 0x9B, 0xD8, 0xD9, 0xFA, 0xFB, 0xFE00, 0xFE01, 0xFE02, 0xFE03, 0xFE05, 0xFE07, 0xFE08, 0xFE09, 0xFE0B, 0xFE0C, 0xFE0D, 0xFE0E, 0xFE0F, 0xFE11, 0xFE12, 0xFE13, 0xFE19, 0xFE1A, 0xFE1B, 0xFE1D, 0xFE1E, 0xFE1F, 0xFE25, 0xFE26, 0xFE29, 0xFE2F, 0xFE30, 0xFE31, 0xFE32, 0xFE33, 0xFE38, 0xFE39, 0xFE3A, 0xFE3B, 0xFE3C, 0xFE3D, 0xFE3E, 0xFE3F, 0xFE40, 0xFE41, 0xFE42, 0xFE43, 0xFE4A, 0xFE4B, 0xFE4C, 0xFE4D, 0xFE4E, 0xFE4F, 0xFE50, 0xFE52, 0xFE53, 0xFE54, 0xFE55, 0xFE57, 0xFE58, 0xFE59, 0xFE62, 0xFE63, 0xFE64, 0xFE65, 0xFE67, 0xFE68, 0xFE69, 0xFE6A, 0xFE6B, 0xFE6C, 0xFE6D, 0xFE6F

      For the ADP1051, write the HEX file values of the user command settings from the buffer into the corresponding registers:

      0x01, 0x02, 0x21, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2A, 0x33, 0x35, 0x36, 0x38, 0x40, 0x41, 0x44, 0x45, 0x46, 0x47, 0x48, 0x4F, 0x50, 0x5E, 0x5F, 0x60, 0x61, 0x64, 0x99, 0x9A, 0x9B, 0xD8, 0xD9, 0xFA, 0xFB, 0xFE00, 0xFE01, 0xFE02, 0xFE03, 0xFE05, 0xFE07, 0xFE08, 0xFE09, 0xFE0B, 0xFE0C, 0xFE0D, 0xFE0E, 0xFE0F, 0xFE11, 0xFE12, 0xFE13, 0xFE19, 0xFE1A, 0xFE1B, 0xFE1C, 0xFE1D, 0xFE1E, 0xFE1F, 0xFE25, 0xFE26, 0xFE29, 0xFE2F, 0xFE30, 0xFE31, 0xFE32, 0xFE33, 0xFE34, 0xFE35, 0xFE36, 0xFE37, 0xFE38, 0xFE39, 0xFE3A, 0xFE3B, 0xFE3C, 0xFE3D, 0xFE3E, 0xFE3F, 0xFE40, 0xFE41, 0xFE42, 0xFE43, 0xFE44, 0xFE45, 0xFE46, 0xFE47, 0xFE48, 0xFE49, 0xFE4A, 0xFE4B, 0xFE4C, 0xFE4D, 0xFE4E, 0xFE4F, 0xFE50, 0xFE51, 0xFE52, 0xFE53, 0xFE54, 0xFE55, 0xFE56, 0xFE57, 0xFE58, 0xFE59, 0xFE5A, 0xFE5B, 0xFE5C, 0xFE5D, 0xFE5E, 0xFE5F, 0xFE60, 0xFE62, 0xFE63, 0xFE64, 0xFE65, 0xFE66, 0xFE67, 0xFE68, 0xFE69, 0xFE6A, 0xFE6B, 0xFE6C, 0xFE6D, 0xFE6E, 0xFE6F
   e. Write the default EEPROM password 0xFF two consecutive times to the EEPROM_PASSWORD command (Register 0xD5) to unlock the EEPROM.
   f. Check if Register 0xFEA2[3] = 1. If it is not 1, go back to the previous step. If it is 1, go to next step.
   g. Execute the STORE_USER_ALL command (Register 0x15) to save the user command settings to the EEPROM.
3. Wait for an interval of t_4.
4. During the interval of t_5, write a value different from 0xFF to the EEPROM_PASSWORD command (Register 0xD5) to lock the EEPROM.
5. Power down V_{DD} for an interval of t_8.
6. Power up V_{DD} again.
7. During the interval of t_{11}, execute the following actions:
   a. Write the latest chip password (it is 0xFFFF if it was not changed) two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip.
   b. This is an optional step. Skip this step if Step 2c is not executed.
      Read the trim register values of the following registers: 0xFE14, 0xFE15, 0xFE16, 0xFE17, 0xFE20, 0xFE28, 0xFE2A, 0xFE2B, 0xFE2C and 0xFE2D. Note that 0xFE15, 0xFE16, 0xFE17 apply only to the ADP1051. Compare them to the values stored in the buffer in Step 2c to ensure that the trim values are not over-written.
   c. Read the registers listed in Step 2d and compare the values in HEX file to ensure that the values are written correctly.
8. Power down V_{DD} of the ADP1050/ADP1051 after the SCL and SDA signals are pulled low.

Programming Procedure for the Board Settings

1. During the interval of t_{17}, execute the following:
   a. Write the latest password (0xFFFF, by default) two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip. If the chip password has been changed, use the last chip password to unlock the chip.
   b. Check if Register 0xFEA0[7] = 1. If it is not 1, go back to the previous step. If it is 1, proceed to next step.
   c. Write the default EEPROM password 0xFF two consecutive times to the EEPROM_PASSWORD command (Register 0xD5) to unlock the EEPROM.
   d. Check if Register 0xFEA2[3] = 1. If it is not 1, go to Step c again. If it is 1, proceed to Step e.
   e. Erase Page 3 through the EEPROM_PAGE_ERASE command (Register 0xD4).
2. Wait for an interval of t_{18}.
3. During the interval of t_{19}, execute the following:
   a. Write the address offset 0x0000 to the EEPROM_PAGE_ERASE command (Register 0xD3).
   b. Write the HEX of board settings to Page 3 of the EEPROM using the EEPROM_DATA_03 command (Register 0xB3) to program multiple bytes in series.
   c. Write a value different from 0xFF to lock the EEPROM.
4. Power down V_{DD} during the interval of t_{22}.
5. Power up V_{DD} again.
6. During the interval of t_{25}, execute the following:
   a. Write the latest password (0xFFFF by default) two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip. If the chip password has been changed, use the last chip password to unlock the chip.
   b. Read the values of the board settings from Page 3 of the EEPROM and compare them to the HEX file of the board settings to ensure that the writes are correct.
      - Write the return number to the EEPROM_NUM_RD_BYTES command (Register 0xD2).
      - Then, write the address offset to the EEPROM_ADDR_OFFSET command (Register 0xD3).
      - Finally, use the EEPROM_DATA_03 command (Register 0xB3) to read multiple bytes in series from Page 3 of the EEPROM.
7. Pull down the SCL and SDA signs and then power down V_{DD} if all values are correct.

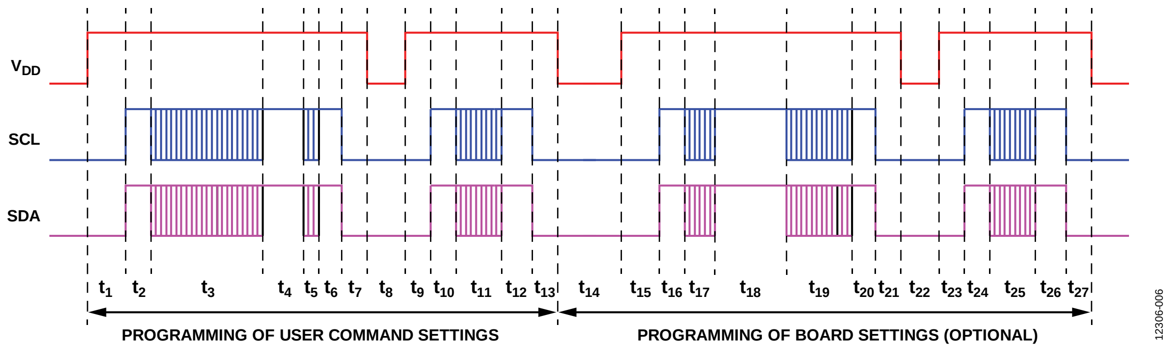

VDD
SCL
SDA
t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16 t17 t18 t19 t20 t21 t22 t23 t24 t25 t26 t27
PROGRAMMING OF USER COMMAND SETTINGS
PROGRAMMING OF BOARD SETTINGS (OPTIONAL)
12306-006

Figure 5. Electric Sequences for EEPROM Programming

POWER SUPPLY CALIBRATION AND TEST

ELECTRIC SEQUENCINGS OF THE CALIBRATION AND TEST

All ADP1050/ADP1051 devices are factory trimmed. If the part is not trimmed in the power supply manufacturing environment, it is recommended that components with a 0.1% or better tolerance be used to meet data sheet specifications.

Figure 6 shows the electric sequences for the calibration. The V_{DD} source is not needed to be applied separately on the PSU because the PSU generates the V_{DD} voltage itself.

The steps for the calibration and test are as follows:

1. The input voltage, V_{IN}, is applied on the PSU. The PSU is powered up and the output voltage can be detected by the voltage meter. The SCL and SDA signals are pulled high from the low state with an interval of t_1 after the V_{IN} reaches the target voltage.

   $t_1 \geq 10$ ms
2. After the SCL and SDA signals are pulled high for an interval of t_2, the PMbus communication starts.

   $t_1 + t_2 \geq 70$ ms
3. The PMBus communication happens during the interval of t_3. All calibrations on the current, voltage, and temperature are processed in this step. The EEPROM is unlocked for access. The STORE_USER_ALL command (Register 0x15) is executed to save the user command settings to the EEPROM before t_3 ends.
4. There is no PMBus communication during the interval of t_4 to wait for the EEPROM erase operation.

   $t_4 \geq 40$ ms
5. The EEPROM is locked during the interval of t_5.
6. The SCL and SDA signals are pulled low after a delay of t_6. There is no special requirement on the interval of t_6.
7. The V_{IN} voltage is disconnected with a delay of t_7 after the SCL and SDA signals are pulled low. There is no special requirement on the interval of t_7. The target is to ensure the V_{DD} of the ADP1050/ADP1051 is powered down later than SCL and SDA signals being pulled down.
8. The V_{IN} voltage and the SCL and SDA signals keep low level for an interval of t_8.

   $t_8 = 10$ ms
9. The V_{IN} voltage is applied on the PSU during the interval of t_9. The SCL and SDA signals are pulled high from the low state with an interval of t_1 after V_{IN} reaches the target voltage.

   $t_9 \geq 10$ ms
10. After the SCL and SDA signals are pulled high for an interval of t_{10}, the PMBus communication starts.

    $t_9 + t_{10} \geq 70$ ms
11. During the interval of t_{11}, all testing on the current, voltage and temperature is processed in this step.
12. The SCL and SDA signals keep high levels during the interval of t_{12}. There is no special requirement on the interval of t_{12}.
13. The V_{IN} voltage is disconnected to the PSU with a delay of t_{13} after SCL and SDA signals are pulled down. There is no special requirement on the interval of t_{13}.

CALIBRATION AND TEST PROCEDURES

1. During the interval of t_3, execute the following actions:
   a. Write the latest password (the default is 0xFFFF) two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip.
   b. Check if Register 0xFEA0[7] = 1. If it is not 1, go back to the previous step. If it is 1, proceed to the next step.
   c. Write the default trim password, 0xFF, two consecutive times to the TRIM_PASSWORD command (Register 0xD6) to unlock the trim registers for write access.
   d. Read Register 0xFE14 value N. Write back a value N + 1 to this register. If the return value from this register is not N + 1, go back to the previous step. If the return value is N + 1, write a value N to this register and proceed to the next step.
   e. Do the calibration and trimming. Follow the calibration procedures based on the information about power supply calibration and trim in the ADP1050/ADP1051 data sheets. These trim registers include Analog Devices factory trim registers (0xFE14, 0xFE15, 0xFE16, 0xFE17, 0xFE20, 0xFE28, 0xFE2A, 0xFE2B, 0xFE2C, 0xFE2D) and none factory trim registers (0x23 and 0xFE59). Note that the 0xFE15, 0xFE16, and 0xFE17 trim registers are only applicable for the ADP1051.
   f. Write the default EEPROM password, 0xFF, two consecutive times to the EEPROM_PASSWORD command (Register 0xD5) to unlock the EEPROM.
   g. Check if Register 0xFEA2[3] = 1. If it is not 1, go back to Step f. If it is 1, proceed to the next step.
   h. Execute the STORE_USER_ALL command (Register 0x15) to save the user command settings to the EEPROM.
2. Wait for an interval of t_4.
3. During the interval of t_5, write a value different from 0xFF to the EEPROM_PASSWORD command (Register 0xD5) to lock the EEPROM.
4. Pull down the SCL and SDA signals and then disconnect the V_{IN} voltage.
5. Connect the V_{IN} voltage to the PSU.
6. During the interval of t_{11}, execute the following actions:
   a. Write the latest chip password (the default is 0xFFFF) two consecutive times to the CHIP_PASSWORD command (Register 0xD7) to unlock the chip.
   b. Collect the voltage and current values through the voltage meters and current meters.
   c. Collect the voltage and current readings through the PMBus commands.
   d. If needed, change the output voltage and output current, and then repeat the test as described in the previous two steps.
   e. Compare the readings with the values from the meters and check the accuracy of the readings.
7. Disconnect the V_{IN} voltage after the SCL and SDA signals are pulled low.

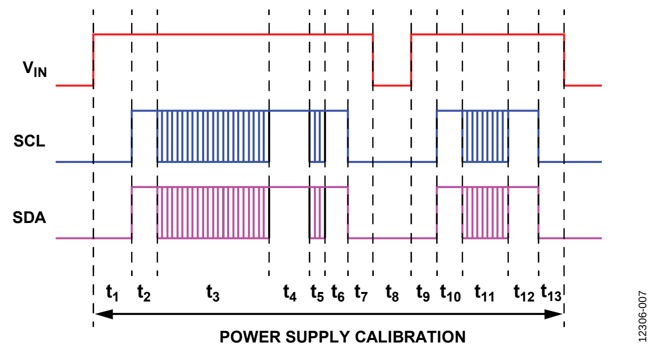


Figure 6. Electric Sequences for Power Supply Calibration and Test

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).