In-System Programming Guide

SLG46824/6/7-A

ISPG-SLG46824/6/7-A

Contents

Figures

Tables

1 Terms and Definitions

ACK	Acknowledge Bit
ERSE	Erase Enable
ERSR	Erase Register
IO	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
NPR	Non-Volatile Memory Read/Write/Erase Protection
NVM	Non-Volatile Memory
PRL	Protect Lock Bit
R/W	Read/Write
RPR	Register Read/Write Protection
SCL	I^2C Clock Input
SDA	I^2C Data Input/Output
WPB	Write Protect Block Bits
WPR	Write Protection Register
WPRE	Write Protect Register Enable

2 Introduction

This document describes the in-system programming procedures for SLG46824, SLG46826, and SLG46827-A (**Note 1**).

Note 1 For the SLG46827-A write protection settings of the device will be permanently disabled once the GreenPAK design is finalized and enters production.

3 Hardware Requirements

3.1 Pinout and Signals

Four pins are required to program the SLG46824/6/7-A: V_{DD}, GND, SCL, and SDA.

The V_{DD} pin requires a voltage ranging from 2.5 V to 5.5 V for Programming (Write) operations, and 2.3 V to 5.5 V for Verification (Read) operations.

The SCL and SDA pins are defined to be standard I^2C signaling. I^2C Fast Mode Plus speed (1 MHz) NVM read and I^2C Fast Mode speed (400 kHz) NVM write communication is supported for these devices. For the timing characteristics for signals on these pins refer to Table 1.

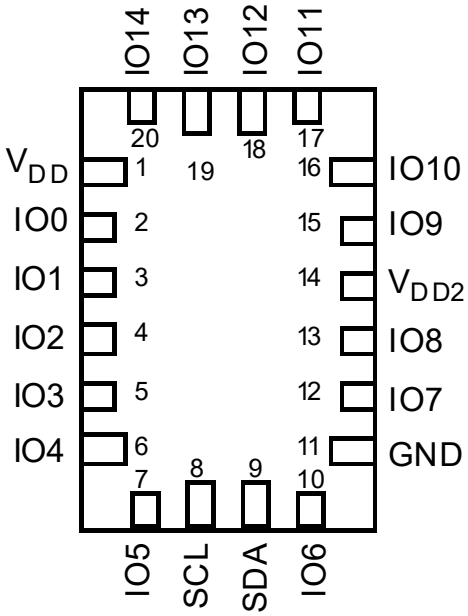

Figure 1: STQFN-20 Pin Configuration

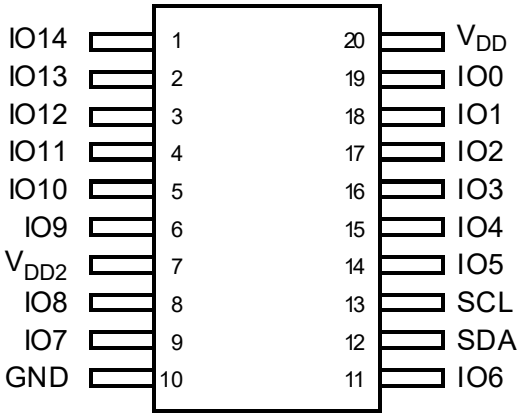

Figure 2: TSSOP-20 Pin Configuration

Table 1: I²C Specifications

Symbol	Parameter	Condition	Fast-Mode			Fast-Mode Plus			Unit
			Min	Typ	Max	Min	Typ	Max	
F_{SCL}	Clock Frequency, SCL	V_{DD} = 2.3 V to 5.5 V	--	--	400	--	--	1000	kHz
t_{LOW}	Clock Pulse Width Low	V_{DD} = 2.3 V to 5.5 V	1300	--	--	500	--	--	ns
t_{HIGH}	Clock Pulse Width High	V_{DD} = 2.3 V to 5.5 V	600	--	--	260	--	--	ns
t_I	Input Filter Spike Suppression (SCL, SDA)	V_{DD} = 2.3 V to 5.5 V	--	--	50	--	--	50	ns
t_{AA}	Clock Low to Data Out Valid	V_{DD} = 2.3 V to 5.5 V	--	--	900	--	--	450	ns
t_{BUF}	Bus Free Time between Stop and Start	V_{DD} = 2.3 V to 5.5 V	1300	--	--	500	--	--	ns
t_{HD_STA}	Start Hold Time	V_{DD} = 2.3 V to 5.5 V	600	--	--	260	--	--	ns
t_{SU_STA}	Start Set-up Time	V_{DD} = 2.3 V to 5.5 V	600	--	--	260	--	--	ns
t_{HD_DAT}	Data Hold Time	V_{DD} = 2.3 V to 5.5 V	0	--	--	0	--	--	ns
t_{SU_DAT}	Data Set-up Time	V_{DD} = 2.3 V to 5.5 V	100	--	--	50	--	--	ns
t_R	Inputs Rise Time	V_{DD} = 2.3 V to 5.5 V	--	--	300	--	--	120	ns
t_F	Inputs Fall Time	V_{DD} = 2.3 V to 5.5 V	--	--	300	--	--	120	ns
t_{SU_STD}	Stop Set-up Time	V_{DD} = 2.3 V to 5.5 V	600	--	--	260	--	--	ns
t_{DH}	Data Out Hold Time	V_{DD} = 2.3 V to 5.5 V	50	--	--	50	--	--	ns

4 Programming Algorithm for NVM Configuration Register Space

The SLG46824, SLG46826, and SLG46827-A programming algorithm for the NVM Configuration space consists of a series of I^2C Sequential Write commands, each of which will program one 16 byte page of NVM memory (Note 2).

Data “1” cannot be re-programmed as data “0” without erasure. Each byte can only be programmed one time without erasure.

Note 2 The functionality of the device is based upon the registers. The registers will not be reloaded from the NVM until power is cycled or a reset command is issued.

The SLG46824, SLG46826, and SLG46827-A can be programmed either with or without an acknowledge polling routine. The acknowledge polling routine is implemented to optimize time sensitive applications that would prefer not to wait the fixed maximum write cycle time (t_{WR}). This method allows the application to know instantly when the NVM write cycle has completed, so a subsequent operation can be started.

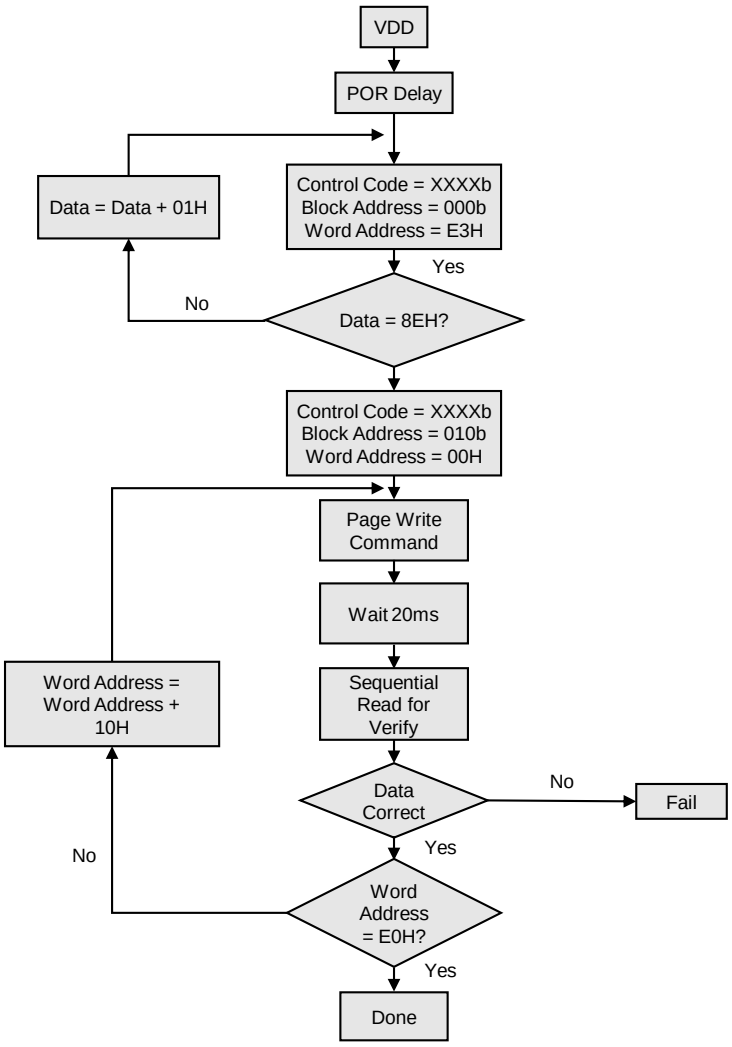

Figure 3: Flowchart for Programming NVM without Acknowledge Polling

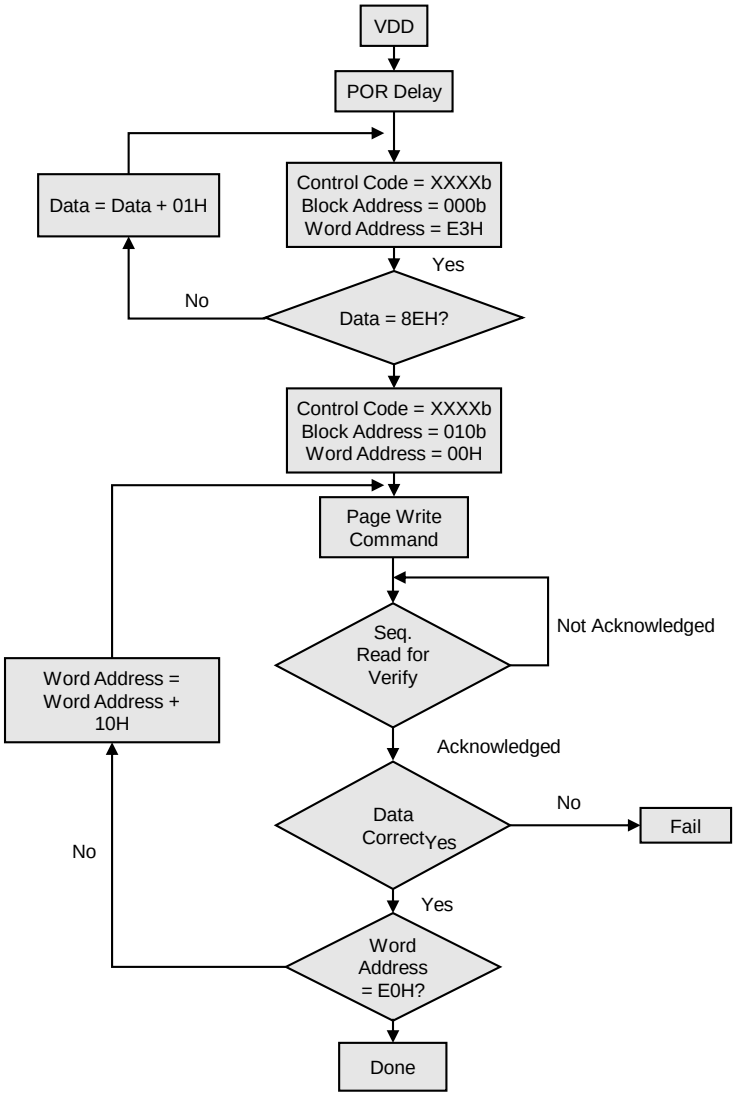

Figure 4: Flowchart for Programming NVM with Acknowledge Polling

5 I²C Signal Specification

5.1 Commands

5.1.1 Write Command

Write access to the NVM is possible by setting A3, A2, A1, A0 to "0000", which allows serial write data for a single page only. Upon receipt of the proper Control Byte and Word Address bytes, the SLG46824/6/7-A will send an ACK. The device will then be ready to receive page data, which is 16 sequential writes of 8-bit data words. The SLG46824/6/7-A will respond with an ACK after each data word is received. The addressing device, such as a Bus Master, must then terminate the write operation with a Stop condition after all page data is written. At that time the GPAK will enter an internally self-timed write cycle, which will be completed within t_{WR} = 20 ms (max). While the data is being written into the NVM Memory Array, all inputs, outputs, internal logic, and I²C access to the Register data will be operational/valid.

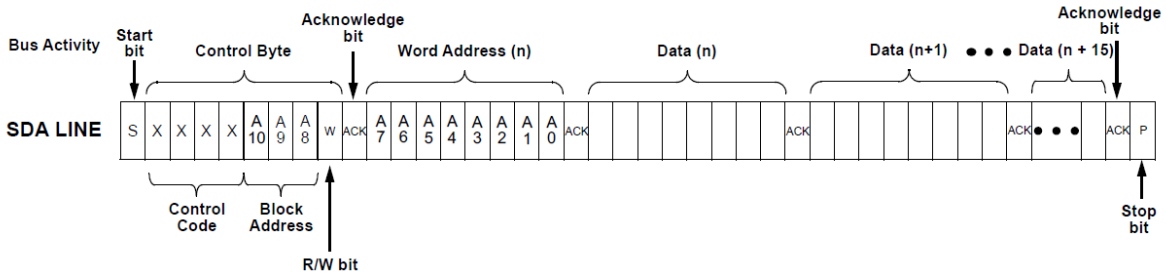

Figure 5: Page Write Command Example

5.1.2 Verify Command

The Random Sequential Read command can be used for verification. The command starts with a Control Byte (with R/W bit set to "0", indicating a write command) and Word Address to set the internal byte address, followed by a Start bit, and then the Control Byte for the read (exactly the same as the Byte Write command). The Start bit in the middle of the command will halt the decoding of a Write command, but will set the internal address counter in preparation for the second half of the command. After the Start bit, the Bus Master issues a second Control Byte with the R/W bit set to "1", after which the SLG46824/6/7-A issues an Acknowledge bit, followed by the requested eight data bits. Once the SLG46824/6/7-A transmits the first data byte, the Bus Master issues an Acknowledge bit. The Bus Master can continue reading sequential bytes of data, and will terminate the command with a Stop condition.

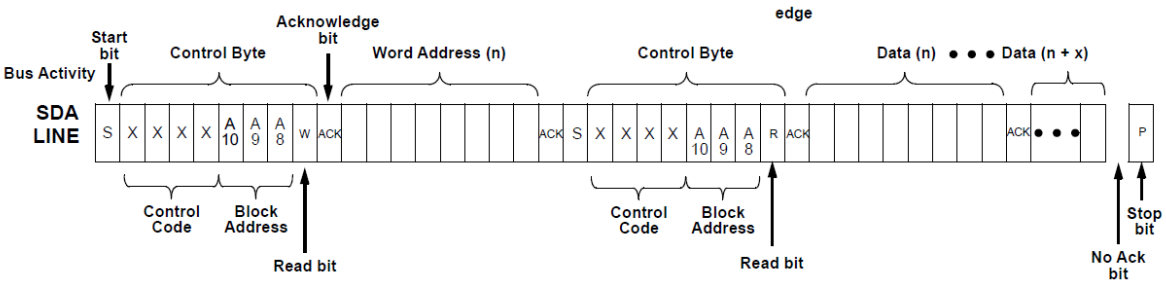

Figure 6: Random Sequential Read Command

5.1.3 Erase Command

The erase scheme allows a 16 byte page in the emulated EEPROM (Note 3) space or the 2K bits NVM chip configuration space to be erased by modifying the contents of the Erase Register (ERSR). When the ERSE bit is set in the ERSR register, the device will start a self-timed erase cycle which will complete in a maximum of t_{ER} = 20 ms (max). Changing the state of the ERSR is accomplished with a

Byte Write sequence with the requirements outlined in this section. The ERSR register is located on the E3H address.

Note 3 Emulated EEPROM is available for SLG46826 only.

Table 2: Erase Register Bit Format

	b7	b6	b5	b4	b3	b2	b1	b0
Page Erase Register	ERSE	--	--	ERSEB4	ERSEB3	ERSEB2	ERSEB1	ERSEB0

Table 3: Erase Register Bit Function Description

Bit	Name		Type	Description
7	ERSE	Erase Enable	W	Setting b7 bit to “1” will start in internal erase cycle on the page defined by ERSEB4-0
6	--	--	--	--
5	--	--	--	--
4	ERSEB4	Page Selection for Erase	W	Define the page address, which will be erased. ERSB4 = 0 corresponds to the Upper 2K NVM used for chip configuration; ERSB4 = 1 corresponds to the 2-k emulated EEPROM (Note 3)
3	ERSEB3		W	
2	ERSEB2		W	
1	ERSEB1		W	
0	ERSEB0		W	

Upon receipt of the proper Device Address and Erase Register Address, the SLG46824/6/7-A will send an ACK. The device will then be ready to receive Erase Register data. The SLG46824/6/7-A will respond with a non-compliant I²C ACK after the Erase Register data word is received. Please reference the SLG46824/6/7-A errata document (revision XC) posted on Renesas’s website for more information. The addressing device, such as a Bus Master, must then terminate the write operation with a Stop condition. At that time, the GPAK will enter an internally self-timed erase cycle, which will be completed within t_{ER} (max 20 ms). While the data is being written into the Memory Array, all inputs, outputs, internal logic, and I²C access to the Register data will be operational/valid.

After the erase has taken place, the contents of ERSE bits will be set to "0" automatically. Erase will be triggered by Stop Bit in I²C command.

5.2 Addressing

Each command to the I²C Serial Communications macrocell begins with a Control Byte. The bits inside this Control Byte are shown in Figure 7. After the Start bit, the first four bits are a control code. Each bit in the control code can be sourced independently from the register or by value defined externally by IO5, IO4, IO3, and IO2. The LSB of the control code is defined by the value of IO2, while the MSB is defined by the value of IO5. The address source (either register bit or PIN) for each bit in the control code is defined by registers [1623:1620]. This gives the User flexibility on the chip level addressing of this device and other devices on the same I²C bus. **The default control code is 0001.** The Block Address is the next three bits (A10, A9, A8), which will define the most significant bits in the addressing of the data to be read or written by the command. The last bit in the Control Byte is the R/W bit, which selects whether a read command or write command is requested, with a "1" selecting for a Read command, and a "0" selecting for a Write command. This Control Byte will be followed by an Acknowledge bit (ACK), which is sent by this device to indicate successful communication of the Control Byte data.

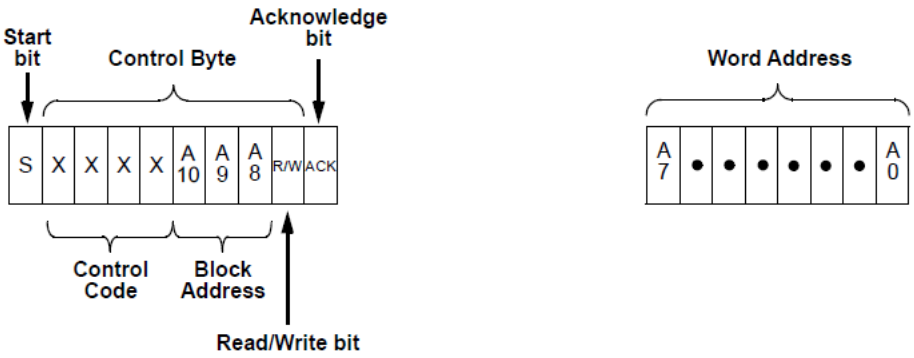

Figure 7: Page Write Command

6 Memory Space

6.1 Memory Map

In the read and write command address structure there are a total of 11 bits of addressing, each pointing to a unique byte of information, resulting in a total address space of 16K bytes. Valid addresses are shown in the memory map in Figure 8.

	I²C Block Address			Memory Space	
Lowest I²C Address = 000h	A10 = 0	A9 = 0	A8 = 0	2 Kbits Register Data Configuration	
	A10 = 0	A9 = 0	A8 = 1	Not Used	
	A10 = 0	A9 = 1	A8 = 0	2 Kbits NVM Data Configuration	16 pages to configure GPAK
	A10 = 0	A9 = 1	A8 = 1	2 Kbits EEPROM (Note 3)	16 pages for emulated EEPROM (Note 3)
Highest I²C Address = 7FFh	A10 = 1	A9 = X	A8 = X	Not Used	

Figure 8: I²C Block Addressing

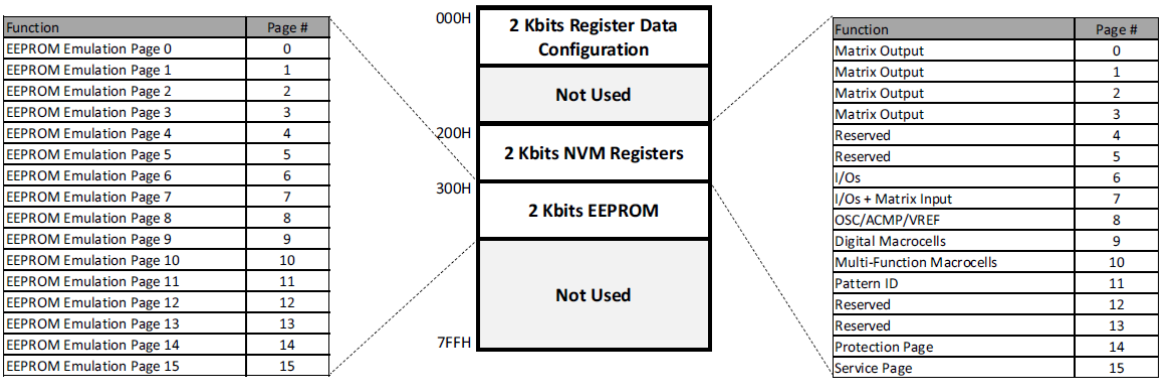

Figure 9: I²C Address Mapping

6.2 Special Pages

GreenPAK's internal NVM memory is divided into 32 pages which hold 16 bytes each. 16 pages are used to configure the GreenPAK, and the other 16 are used for the emulated EEPROM (Note 3) function.

6.2.1 Protection Page

Page #14 inside the 2Kbits NVM Registers memory space is a dedicated protection page. All 16 bytes of Page #14 are dedicated to protection settings, even though only 3 of those bytes are used. The

whole page is used because a page is the smallest NVM section that can be programmed/erased or blocked for erase/program.

Once the User determines the security setting, they can do an NVM page 14 programming (program the security setting and set security page lock bit = 1). The changes to the Protection Page will not be active until the device is reset. The user can do an immediate reset using one of the following methods:

- Power down the device, then power it back on
- Perform a soft reset using an I^2C command

This will allow the memory interface circuit to know that page 14 (security page) is write/erase protected.

6.2.2 Service Page

Page #15 inside the 2Kbits NVM Registers memory space contains reserved information that is preprogrammed during device final test. The information on this page can be Read but not Written by the User. As this page cannot be altered by the User, the programming algorithm does not need to address this page.

7 Programming Algorithm for the Emulated EEPROM Space

The SLG46826 programming algorithm for the emulated EEPROM (Note 3) space consists of a series of I²C Sequential Write commands, each of which will program one 16 byte page of NVM memory.

Data “1” cannot be re-programmed as data “0” without erasure. Each byte can only be programmed one time without erasure.

Programming of the emulated EEPROM (Note 3) follows a similar flow to programming of the NVM with two differences:

- Block Address for emulated EEPROM (Note 3) is 011b (in contrast to the NVM Configuration Registers Block Address, which is 010b)
- With emulated EEPROM (Note 3), all 16 pages are user accessible.

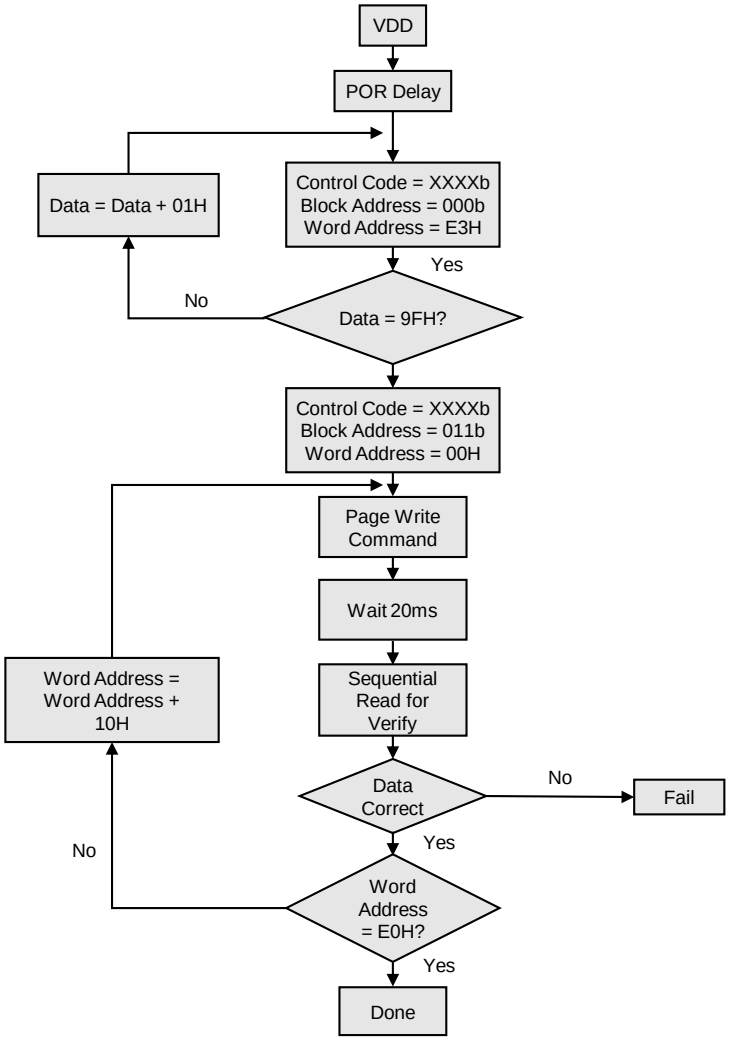

Figure 10: Flowchart for Programming EEPROM without Acknowledge Polling

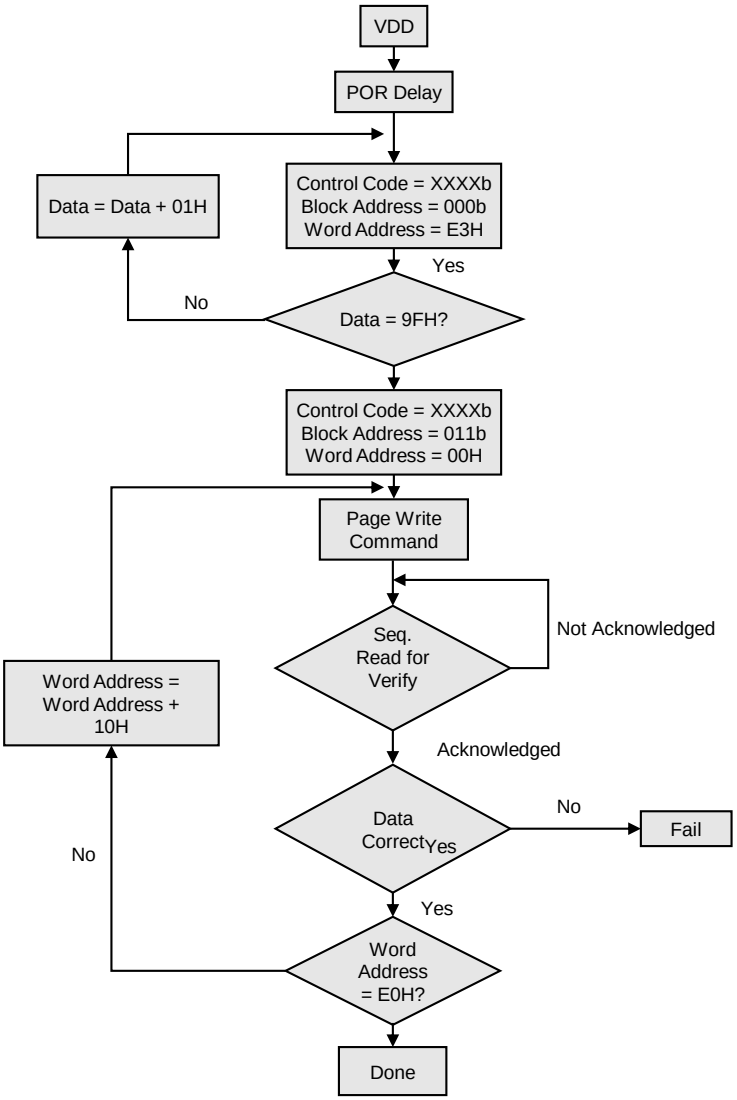

Figure 11: Flowchart for Programming Emulated EEPROM with Acknowledge Polling

8 Protection for Emulated EEPROM

The SLG46826 utilizes a software scheme that allows a portion or the entire emulated EEPROM (Note 3) to be inhibited from being written/ erased to by modifying the contents of the Write Protection Register (WPR). If desired, the WPR can be set so that it may no longer be modified/erased, thereby making the current protection scheme permanent. The status of the WPR can be determined by following a Random Read sequence. Changing the state of the WPR is accomplished with a Byte Write sequence with the requirements outlined in this section.

The WPR register is located on I²C Block Address = 000b, I²C Word Address = E2H.The WPR format is shown in Table 4, and the WPR bit functions are included in Table 5.

Table 4: Write/Erase Protect Register Format

	b7	b6	b5	b4	b3	b2	b1	b0
WPR						WPRE	WPB1	WPB0

Table 5: Write/Erase Protect Register Bit Function Description

Bit	Name		Type	Description
2	WPRE	Write Protect Register Enable	R/W	0: No Software Write Protection enabled (default) 1: Write Protection is set by the state of WPB[1:0] bits
1:0	WPB1	Write Protect Block Bits	R/W	00: Upper quarter of emulated EEPROM (Note 3) is write protected (default) 01: Upper half of emulated EEPROM (Note 3) is write protected 10: Upper 3/4 of emulated EEPROM (Note 3) is write protected 11: Entire emulated EEPROM (Note 3) is write protected
	WPB0		R/W	

Write Protect Enable (WPRE): The Write Protect Enable Bit is used to enable or disable the device Software Write/Erase Protect. A Logic 0 in this position will disable Software Write/Erase Protection, and a Logic 1 will enable this function.

Write Protect Block Bits (WPB1:WPB0): The Write Protect Block bits allow four levels of protection of the Memory Array, provided that the WPRE bit is a Logic 1. If the WPRE bit is a Logic 0, the state of the WPB1:0 bits have no impact on device protection.

Protect Lock Bit (PRL): The Protect Lock Bit is used to permanently lock the current state of the WPR, as well as RPR and NPR. A Logic 0 indicates that the WPR, RPR, and NPR can be modified, whereas a Logic 1 indicates the WPR, RPR, and NPR has been locked and can no longer be modified. The PRL register bit is located at register [1824].

Revision History

Revision	Date	Description
1.4	23-Dec-2022	Added note to Introduction section Fixed typos
1.3	16-Mar-2022	Renesas rebranding SLG46827-A added to In-System Programming Guide
1.2	14-Sep-2020	Updated according to Dialog's Writing Guideline
1.1	4-Mar-2019	Updated Flowcharts Added a reference to the NVM and EEPROM Erase Register erratum in section 3.1.3
1.0	13-Feb-2018	Initial release

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

RoHS Compliance

Renesas Electronics Corporation's suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.