AN5403

VR5100 OTP programming instructions

Rev. 1 — 6 April 2017 Application note

Document information

Information	Content
Keywords	AN5403, VR5100, OTP, KITVR5100FRDMPGM
Abstract	This application note provides a detailed description of the VR5100 one time programmable (OTP) function.

Revision history

Rev	Date	Description
1	20170406	initial version

1 Introduction

This application note provides a detailed description of the VR5100 one time programmable (OTP) function. It outlines the system requirements and the instructions needed to program the internal fuses for a selected power-up configuration.

NXP analog ICs are manufactured using the SMARTMOS process, a combinational BiCMOS manufacturing flow that integrates precision analog, power functions and dense CMOS logic together on a single cost-effective die.

2 Hardware considerations

The minimum system requirements for programming the OTP fuses are:

- An I^2C-bus communication bridge for communicating with the VR5100
- A 1.7 V to 3.6 V power supply at VCCI2C (power to I^2C-bus block and pull-up resistors for the SCL and SDA lines).
- A 9.5 V, 100 mA power supply at VDDOTP bypassed by two 10 µF capacitors. See Section 4.2 "OTP programming example" for details.
- An input voltage of 3.3 V at the VIN pin

The KITVR5100FRDMPGM programming board includes all of these features in a USB standalone solution and is plug-and-play compatible with the VR5100 GUI software.

Figure 1 shows the minimum requirements for programming the VR5100. For programming the VR5100 on an application board, observe the hardware constraints outline in Section 3 "Programming the VR5100 on an application board".

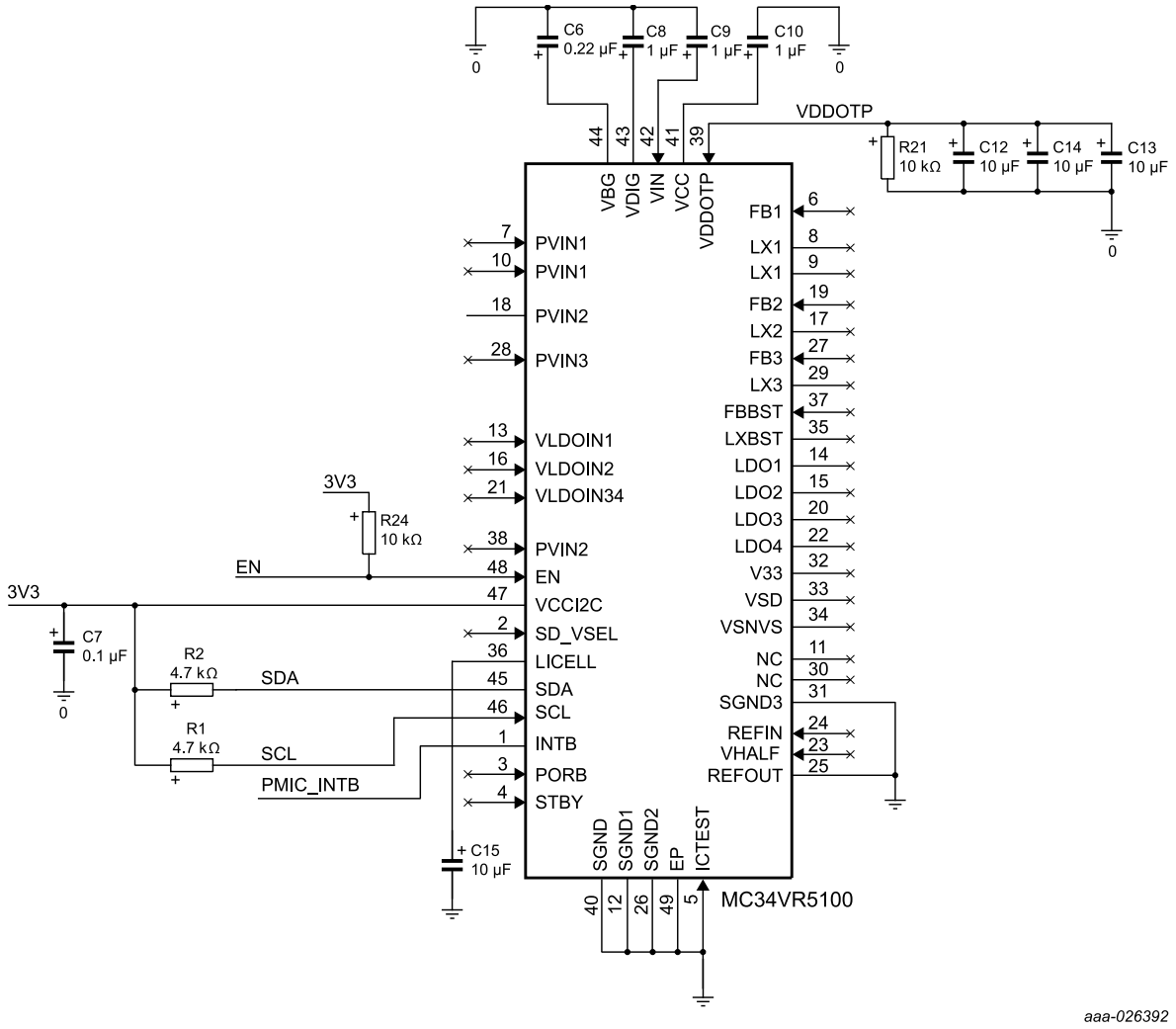

Figure 1. Minimum OTP programming requirements diagram

3 Programming the VR5100 on an application board

When programming the VR5100 in an application board, you must apply voltages at the VIN, VCCI2C, and VDDOTP pins. Make the necessary adjustments to assure that voltages are applied on these rails in a fully populated system board.

3.1 Isolating SCL/SDA

During OTP programming, the VR5100 is connected to an I^2C-bus communications bridge (such as the FRDM-KL25Z) and receives commands via the SCL and SDA pins. In a typical application, the SCL and SDA pins of the VR5100 are connected to the communication ports of an I^2C-bus master, (the processor). Depending on how the ports in the processor are designed, it may or may not be valid to communicate with the VR5100 using an external dongle while the SCL/SDA pins are still connected to the processor. This is particularly true when the processor is unpowered due to a yet-to-be-programmed VR5100.

Use an external programmer to isolate the SCL/SDA lines going to the processor while communicating with the VR5100; see the example in Figure 2. In the normally closed position of the analog switch (NLAS3158 or similar), SCL and SDA on the VR5100 are connected to the processor. When the signal Programmer_Select_O/P is HIGH, SCL and SDA on the VR5100 are connected to the external programming interface. The Programmer_Select_O/P signal can be generated by the programming interface as well.

Note: *Using the analog switch may not be the most cost effective way to isolate the I^2C-bus. Similar functionality can be achieved by using solder shorts or 0 Ω resistors. However, minor rework of the board would be required once OTP programming is completed.*

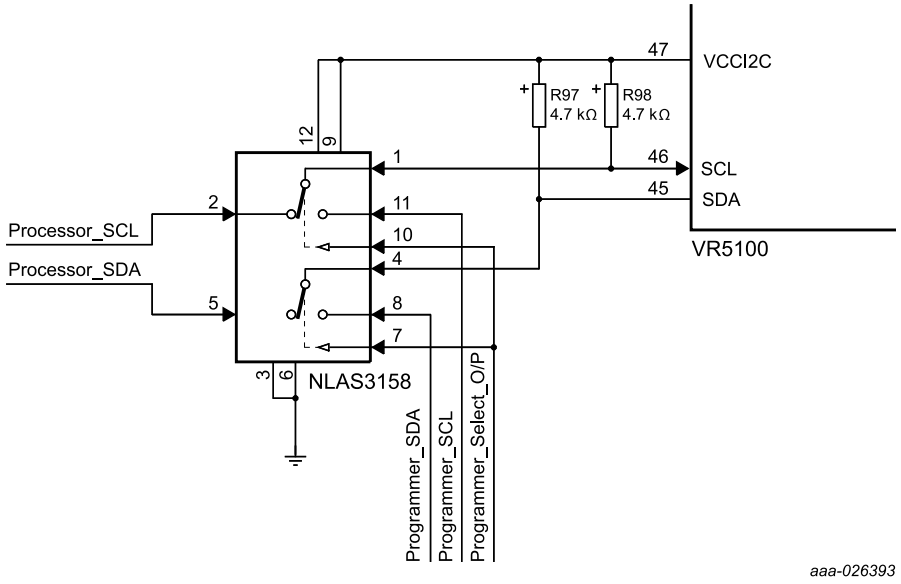

Figure 2. Isolating SCL and SDA using an analog switch

3.2 Programming using the KITVR5100FRDMPGM

The NXP KITVR5100FRDMPGM board provides an ideal platform for programming the VR5100 OTP. The board integrates a 3.3 V LDO to power the VR5100 and a boost converter with a 9.5 V output voltage to generate the OTP programming voltage. An integrated USB-to-I^2C-bus converter allows PC communication with the VR5100 using a NXP supplied GUI. See Figure 3 for a block diagram of the KITVR5100FRDMPGM.

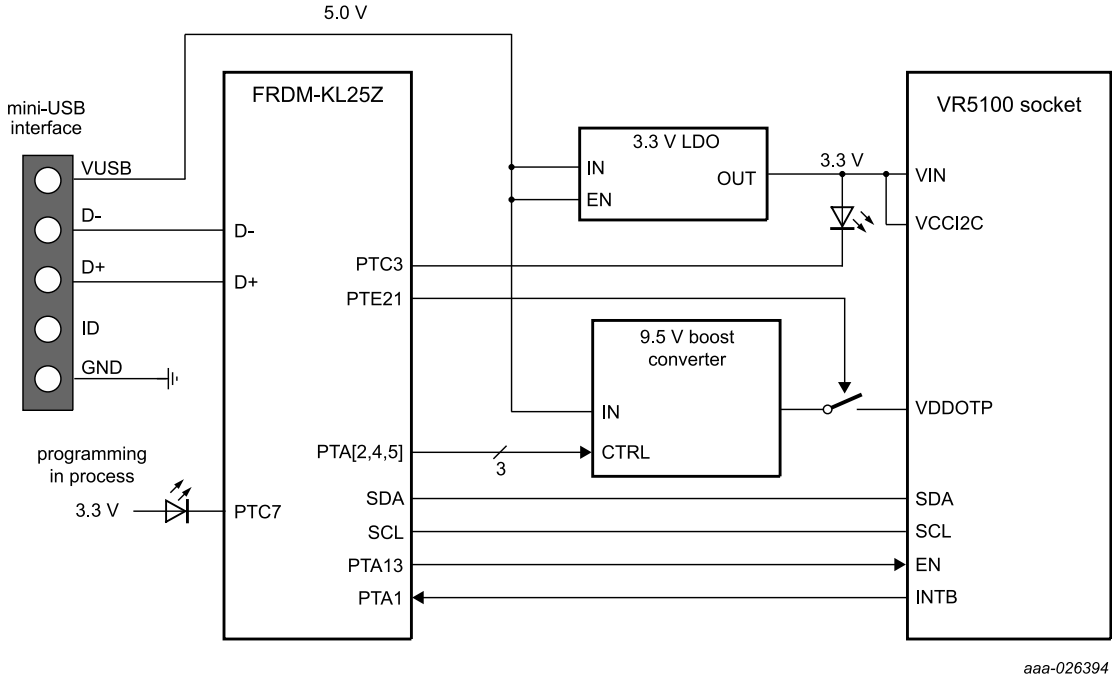

Figure 3. KITVR5100FRDMPGM Rev. D block diagram

3.3 Programming using a generic programmer

Figure 4 shows how to interface a generic programmer with the VR5100 in an application board. For applications that use a single rail for VIN and VCCI2C, the connection is straightforward as shown in Figure 4. However, other loads connected to the 3.3 V rail must not surpass the current rating of the LDO. If that is the case, isolation in the form of an analog switch, a solder short, or a 0 Ω resistor is required.

The following requirements apply when a generic programmer board is used:

- VIN power supply: 3.3 V, 100 mA
- VCCI2C power supply: 1.8 V to 3.3 V, 10 mA
- I^2C-bus master
- GPO signal to control VR5100 EN pin
- GPO signal to control analog switch (Programmer_Select_O/P)
- 9.5 V, 100 mA power supply at VDDOTP bypassed by two 10 µF capacitors. The voltage depends on the silicon revision used. See Section 4.2 "OTP programming example" for details.

Figure 4 illustrates a typical configuration using a generic programmer.

Note: *Using the analog switch may not be the most cost effective way to isolate the I^2C-bus. Similar functionality can be achieved by using solder shorts or 0 Ω resistors. However, minor rework of the board would be required once OTP programming is completed.*

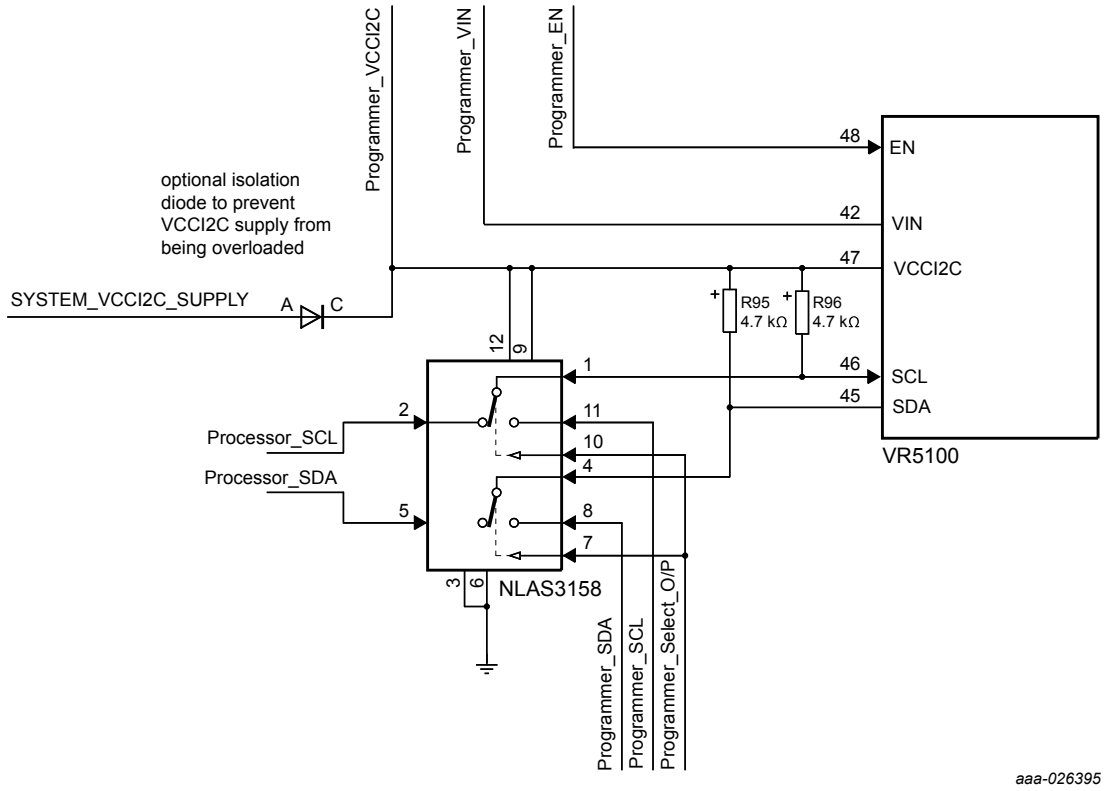

Figure 4. Interfacing a generic programmer to the VR5100 in an application board

4 OTP overview

The regulators in the VR5100 are configurable and are designed for flexibility in a wide variety of applications. One-time-programmable (OTP) fuses in the VR5100 demonstrate this flexibility. Key startup parameters and regulator configuration information can be programmed into the VR5100 to enable it to power the system. These parameters are:

- General
  - I^2C-bus slave address
  - EN pin configuration
  - Regulator startup sequence and timing
  - PORB configuration
- Buck regulators
  - Output voltage
  - Single phase or independent mode configuration
  - Switching frequency
  - Soft start ramp rate
- Boost regulator and LDOs
  - Output voltage

VR5100 starts up based on the contents of the TBBOTP registers. You can load the TBBOTP registers from different sources, as shown in Table 1. The default setting is hard-coded in the VR5100 and is available in all non-programmed and programmed VR5100 devices. Once you complete OTP programming, you can load TBBOTP either from the default values or from the OTP fuses.

The OTP block in the VR5100 also features a try-before-buy (TBB) mode which allows experimenting with different voltages and sequences of the regulators. In the TBB mode, you can write to the TBBOTP registers directly and use them for startup of the VR5100. To maintain the contents of the TBBOTP registers in the absence of the main input supply (VIN), use a coin cell at the LICELL pin.

4.1 Power-up configuration

The VR5100 powers up based on the contents of the TBBOTP registers. Depending on certain pin and bit settings, the TBBOTP registers load from different sources as shown in Table 1.

Table 1. Start-up configuration source and conditions

Source	Condition	Power-up configuration
ROM	VDDOTP pin = VDIG[1]	VR5100 starts up using the factory default settings
TBBOTP registers	VDDOTP pin = 0 V and TBB_POR bit = 1	VR5100 starts up from current values of TBBOTP registers. This is referred to as the try-before-buy mode.
OTP fuses	VDDOTP pin = 0 V and TBB_POR bit = 0	VR5100 starts up from the OTP fuse values

[1] Pull up VDDOTP to VDIG with a 100 kΩ resistor.

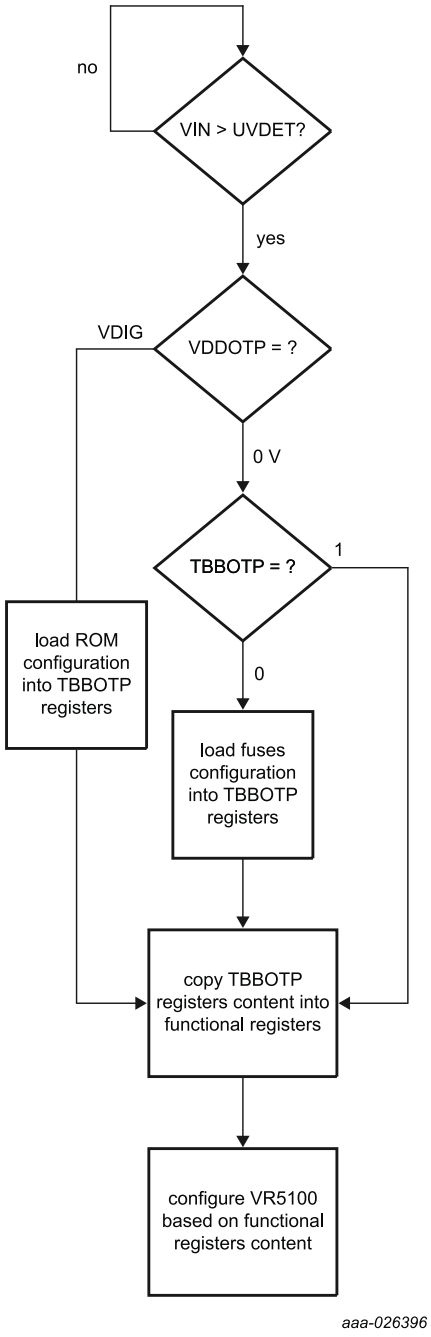

Figure 5. VR5100 power-up flowchart

The TBBOTP registers serve as temporary storage for any of the following:

- The values to be written to the fuses
- The values read from the fuses
- The values to start from during TBB development
- The values read from the default configuration

The TBBOTP registers reside within the extended page 1 of the VR5100 register map.

During a power-up, the TBBOTP registers behave as follows:

- The contents of the TBBOTP registers initialize to zero when a valid VIN is first applied.
- The values that then load into the TBBOTP registers depend on the setting of the VDDOTP pin, and on the value of the TBB_POR.
  - If VDDOTP = VDIG (1.5 V), the TBBOTP values are loaded from ROM.
  - If VDDOTP = 0 V and TBB_POR = 1, the VR5100 starts up from current values of TBBOTP registers.
  - If VDDOTP = 0 V and TBB_POR = 0, the TBBOTP values are loaded from the fuses.

Notice that the initial value of TBB_POR is always '0'.

The contents of the TBBOTP registers may be modified by I^2C-bus. To communicate with I^2C-bus, VIN must be valid and VCCI2C (to which SDA and SCL are pulled up) must be powered by a 1.7 V to 3.6 V supply. VIN or the coin cell voltage must be valid to maintain the contents of the TBBOTP registers. To power on with the contents of the TBBOTP registers, a valid turn-on event must occur with the following conditions:

- A valid VIN
- Optional LICELL
- VDDOTP = 0 V
- TBB_POR = 1

4.2 OTP programming example

The one time programmable memory is controlled by fuses. The startup configuration programmed into the VR5100 depends on the state of these fuses as defined during the OTP programming process.

There are five banks of fuses. Each bank consists of 26 fuses. Of the 26 fuses in a bank, 20 fuses are programmable by the user. The remaining six fuses are redundant fuses that allow implementation of error correction. An error correction code within the VR5100 corrects single bit errors if they occur in the bank.

The programming voltage should have a tolerance of ±3 % and OTP programming should be done at room temperature. For reliability reasons, do not OTP program a given part more than once.

Note: *All code examples in this document represent a script using the KITVR5100FRDMPGM and the associated GUI. Command syntax may vary if the user utilizes a different tool for communication.*

The following is an example of programming the VR5100.

```
//-------------------------------------------------------
// Set VDDOTP = 0 V, EN = HIGH, LICELL = 3.0 V (optional), VIN = 3.3 V,
 VCCI2C = 3.3 V
//-------------------------------------------------------
//=======================================================
// Following commands come from customer based on their requirements
//=======================================================
WRITE_I2C:7F:01 // Access VR5100 EXT Page1 (OTP Registers)
//[Extended Page 1 Registers: 0xA0 - 0xAF] --------------
WRITE_I2C:A0:10 // SW1 OTP Vout = 1.100V
WRITE_I2C:A1:01 // SW1 OTP Sequence = 1
WRITE_I2C:A2:05 // SW1 OTP SwConfig = Single Phase; SW1 OTP Frequency = 2.0MHz
WRITE_I2C:AC:06 // SW2 OTP Vout = 1.800V
WRITE_I2C:AD:02 // SW2 OTP Sequence = 2
WRITE_I2C:AE:01 // SW2 OTP Frequency = 2.0MHz
```

```
//[Extended Page 1 Registers: 0xB0 - 0xBF] --------------
WRITE_I2C:B0:09 // SW3 OTP Vout = 1.350V
WRITE_I2C:B1:05 // SW3 OTP Sequence = 5
WRITE_I2C:B2:01 // SW3 OTP Frequency = 2.0MHz
WRITE_I2C:BC:00 // SWBST OTP Vout = 5.000 Volts
WRITE_I2C:BD:00 // SWBST OTP Sequence = 0
//[Extended Page 1 Registers: 0xC0 - 0xCF] --------------
WRITE_I2C:C0:06 // VSNVS OTP Vout = 3.000 Volts
WRITE_I2C:C8:00 // LDO1 OTP Vout = 1.800 Volts
WRITE_I2C:C9:04 // LDO1 OTP Sequence = 4
WRITE_I2C:CC:0E // LDO2 OTP Vout = 1.500 Volts
WRITE_I2C:CD:04 // LDO2 OTP Sequence = 4
//[Extended Page 1 Registers: 0xD0 - 0xDF] --------------
WRITE_I2C:D0:03 // VSD OTP Vout = 3.300 Volts
WRITE_I2C:D1:04 // VSD OTP Sequence = 4
WRITE_I2C:D4:03 // V33 OTP Vout = 3.300 Volts
WRITE_I2C:D5:03 // V33 OTP Sequence = 3
WRITE_I2C:D8:0F // LDO3 OTP Vout = 3.300 Volts
WRITE_I2C:D9:03 // LDO3 OTP Sequence = 3
WRITE_I2C:DC:0F // LDO4 OTP Vout = 3.300 Volts
WRITE_I2C:DD:03 // LDO4 OTP Sequence = 3
//============================================================
// Following commands are for actual OTP programming
//============================================================
 // PROGRAMMING COMMANDS FOLLOW
//============================================================
//------------------------------------------------------------
WRITE_I2C:7F:02 // Access VR5100 EXT Page2
WRITE_I2C:D0:FF
WRITE_I2C:D8:46
//------------------------------------------------------------
VPGM:ON // Turn ON 9.5V Supply at VDDOTP
DELAY:50 // Adds 50msec delay to allow VPGM time to ramp up
//------------------------------------------------------------
WRITE_I2C:D9:FF // Command to blow fuses
//------------------------------------------------------------
DELAY:750 // Adds 750msec delay to allow programming to complete
VPGM:OFF // Turn off 9.5V Boost Supply
DELAY:500 // Adds delay to allow VPGM to bleed off
EN:LOW // EN LOW to reload new OTP data
DELAY:500
EN:HIG
```

After completing OTP programming by following the above steps, read the registers 0xA0 to 0xE8 in extended page 1 and compare them to the required register values as in the script. Additionally, read the ECC interrupt bit, OTP_ECCI in register 0x0E. If there is an error in the programmed values or if the OTP_ECCI bit is set to logic 1, reject the part because the programming process resulted in errors.

4.3 Try-before-buy mode example

Note: *All code examples in this document represent a script using the KITVR5100FRDMPGM and the associated GUI. Command syntax may vary if the user utilizes a different tool for communication.*

```
//=============================================
//Following commands come from customer based on their requirements
```

```
//================================================
WRITE_I2C:7F:01 // Access VR5100 EXT Page1 (OTP Registers)
//[Extended Page 1 Registers: 0xA0 - 0xAF]
WRITE_I2C:A0:10 // SW1 OTP Vout = 1.100V
WRITE_I2C:A1:01 // SW1 OTP Sequence = 1
WRITE_I2C:A2:05 // SW1 OTP SwConfig = Single Phase; SW1 OTP Frequency = 2.0MHz
WRITE_I2C:AC:06 // SW2 OTP Vout = 1.800V
WRITE_I2C:AD:02 // SW2 OTP Sequence = 2
WRITE_I2C:AE:01 // SW2 OTP Frequency = 2.0MHz
//[Extended Page 1 Registers: 0xB0 - 0xBF] --------------
WRITE_I2C:B0:09 // SW3 OTP Vout = 1.350V
WRITE_I2C:B1:05 // SW3 OTP Sequence = 5
WRITE_I2C:B2:01 // SW3 OTP Frequency = 2.0MHz
WRITE_I2C:BC:00 // SWBST OTP Vout = 5.000 Volts
WRITE_I2C:BD:00 // SWBST OTP Sequence = 0
//[Extended Page 1 Registers: 0xC0 - 0xCF] --------------
WRITE_I2C:C0:06 // VSNVS OTP Vout = 3.000 Volts
WRITE_I2C:C8:00 // LDO1 OTP Vout = 1.800 Volts
WRITE_I2C:C9:04 // LDO1 OTP Sequence = 4
WRITE_I2C:CC:0E // LDO2 OTP Vout = 1.500 Volts
WRITE_I2C:CD:04 // LDO2 OTP Sequence = 4
//[Extended Page 1 Registers: 0xD0 - 0xDF] --------------
WRITE_I2C:D0:03 // VSD OTP Vout = 3.300 Volts
WRITE_I2C:D1:04 // VSD OTP Sequence = 4
WRITE_I2C:D4:03 // V33 OTP Vout = 3.300 Volts
WRITE_I2C:D5:03 // V33 OTP Sequence = 3
WRITE_I2C:D8:0F // LDO3 OTP Vout = 3.300 Volts
WRITE_I2C:D9:03 // LDO3 OTP Sequence = 3
WRITE_I2C:DC:0F // LDO4 OTP Vout = 3.300 Volts
WRITE_I2C:DD:03 // LDO4 OTP Sequence = 3
//[Extended Page 1 Registers: 0xE0 - 0xEF] --------------
WRITE_I2C:E0:00 // EN_CFG=0 (EN is level sensitive); OTP_SWDVS_CLK=0 (25 mV step
 each 2.0 us); OTP_SEQ_CLK_SPEED=0 (Clock speed 500 us increments)
WRITE_I2C:E4:00 // TBB_POR=0 (Try Before Buy disabled)
WRITE_I2C:E8:00 // OTP_PG_EN=0 (RESETBMCU default mode. After start-up,
 RESETBMCU will be released 2.0 ms after the last regulator is enabled)
//[Extended Page 1 Registers: 0xF0 - 0xFF] --------------
WRITE_I2C:FF:08 // I2C_SLV_ADDR[3]=1 (hard coded); I2C_SLV_ADDR[2:0] = 000 (I2C
 Device Address is 0x08)
//================================================
// Following commands are for entering the TBB mode.
//================================================
// TRY-BEFORE-BUY COMMANDS FOLLOW
//================================================
WRITE_I2C:E4:80 // TBB POR=1 (This Enables TBB Mode)
//================================================
EN:LOW // EN LOW
DELAY:500
EN:HIGH // EN HIGH to Start PMIC from desired TBB Configuration
```

4.4 OTP register descriptions

The VR5100 OTP registers consist of 130 fuses arranged in five banks. Each bank contains 26 fuses. Each fuse represents one bit of the TBBOTP register map. Table 2 to Table 6 show the banks, their fuses and the corresponding bits in the register map.

Table 2. Bank 1

Fuses	OTP register name	Register bits	Description
4:0	SW1 VOLT	OTP_SW1_VOLT[4:0]	SW1 power-up voltage
7:5	SW1 SEQ	OTP_SW1_SEQ[2:0]	SW1 power-up sequence
12:8	reserved	reserved	—
15:13	reserved	reserved	—
17:16	SW1 FREQ	OTP_SW1_FREQ[1:0]	SW1 frequency
19:18	reserved	reserved	—
25:20	—	—	ECC check bits for fuse bank 1

Table 3. Bank 2

Fuses	OTP register name	Register bits	Description
3:0	SW2 VOLT	OTP_SW2_VOLT[3:0]	SW2 power-up voltage
6:4	SW2 SEQ	OTP_SW2_SEQ[2:0]	SW2 power-up sequence
10:7	SW3 VOLT	OTP_SW3_VOLT[4:0]	SW3 power-up voltage
13:11	SW3 SEQ	OTP_SW3_SEQ[2:0]	SW3 power-up sequence
15:14	SW2 FREQ	OTP_SW2_FREQ[1:0]	SW2 frequency
17:16	SW3 FREQ	OTP_SW3_ FREQ[1:0]	SW3 frequency
19:18	reserved	reserved	—
25:20	—	—	ECC check bits for fuse bank 2

Table 4. Bank 3

Fuses	OTP register name	Register bits	Description
3:0	LDO1 VOLT	OTP_LDO1_VOLT[3:0]	LDO1 power-up voltage
7:4	LDO1 SEQ	OTP_LDO1_SEQ[3:0]	LDO1 power-up sequence
11:8	LDO2 VOLT	OTP_LDO2_VOLT[3:0]	LDO2 power-up voltage
15:12	LDO2 SEQ	OTP_LDO2_SEQ[3:0]	LDO2 power-up sequence
18:16	VSNVS VOLT	OTP_VSNVS_VOLT[2:0]	VSNVS power-up voltage
19	reserved	reserved	—
25:20	—	—	ECC check bits for fuse bank 3

Table 5. Bank 4

Fuses	OTP register name	Register bits	Description
3:0	LDO3 VOLT	OTP_LDO3_VOLT[3:0]	LDO3 power-up voltage
7:4	LDO3 SEQ	OTP_LDO3_SEQ[3:0]	LDO3 power-up sequence
11:8	LDO4 VOLT	OTP_LDO4_VOLT[3:0]	LDO4 power-up voltage
15:12	LDO4 SEQ	OTP_LDO4_SEQ[3:0]	LDO4 power-up sequence

Fuses	OTP register name	Register bits	Description
18:16	I2C SLAVE ADDR	OTP_I2C_SLAVE_ADDR[2:0]	VR5100 I^2C-bus address
19	OTP PWRGD EN	OTP_PWRGD_EN[0]	fault mode enable
25:20	—	—	ECC check bits for fuse bank 4

Table 6. Bank 5

Fuses	OTP register name	Register bits	Description
1:0	SWBST VOLT	OTP_SWBST_VOLT[1:0]	SWBST power-up voltage
4:2	SWBST SEQ	OTP_SWBST_SEQ[2:0]	SWBST power-up sequence
6:5	V33 VOLT	OTP_V33_VOLT[3:0]	V33 power-up voltage
9:7	V33 SEQ	OTP_V33_SEQ[2:0]	V33 power-up sequence
11:10	VSD VOLT	OTP_VSD_VOLT[1:0]	VSD power-up voltage
14:12	VSD SEQ	OTP_VSD_SEQ[2:0]	VSD power-up sequence
15	SEQ CLK FREQ	OTP_SEQ_CLK_FREQ[0]	SEQ clock frequency selection
16	SWDVS CLK	OTP_SWDVS_CLK[0]	DVS clock selection
17	EN CFG	OTP_EN_CFG[0]	EN level/edge configuration
18	reserved	reserved	—
19	OTP BLOWN	OTP_BLOWN[0]	OTP fuses blown status
25:20	—	—	ECC check bits for fuse bank 2

The TBBOTP registers store data for programming the fuses. These registers are written to and read from using the I^2C-bus interface.

Once the TBBOTP registers are loaded with the correct values, the fuses can then be programmed. Before discussing the programming process, some salient features of the OTP function are described.

4.4.1 TBBOTP registers description

The TBBOTP registers for configuring the switching regulators are listed in Table 12 to Table 18 provide a general description of the TBBOTP registers for all the switching regulators.

Table 7. OTP switching regulators register summary

Register	Address	Output
OTP SW1 VOLT	0xA0	SW1 OTP output voltage set point
OTP SW1 SEQ	0xA1	SW1 OTP power-up sequence selection
OTP SW1 CONFIG	0xA2	SW1 OTP frequency selection
OTP SW2 VOLT	0xAC	SW2 OTP output voltage set point
OTP SW2 SEQ	0xAD	SW2 OTP power-up sequence selection
OTP SW2 FREQ	0xAE	SW2 OTP frequency selection
OTP SW3 VOLT	0xB0	SW3 OTP output voltage set point

Register	Address	Output
OTP SW3 SEQ	0xB1	SW3 OTP power-up sequence selection
OTP SWBST VOLT	0xBC	SWBST OTP output voltage set point
OTP SWBST SEQ	0xBD	SWBST OTP power-up sequence selection

Table 8. OTP SW1 VOLT register description

Name	Bit	Description
OTP_SW1_VOLT	4:0	Sets the SW1 output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to SW1 output voltage configuration table on data sheet for all possible configurations.
UNUSED	7:5	unused

Table 9. OTP SW1 SEQ register description

Name	Bit	Description
OTP_SW1_SEQ	2:0	assigns the power-up sequence slot 0 to 7 for SW1
UNUSED	7:3	unused

Table 10. OTP SW1 CONFIG register description

Name	Bit	Description
OTP_SW1_FREQ	1:0	SW1 OTP frequency configuration 00 = 1.0 MHz 01 = 2.0 MHz 10 = 4.0 MHz 11 = reserved
RESERVED	3:2	01 - this setting must be used
UNUSED	7:4	unused

Table 11. OTP SW2 VOLT register description

Name	Bit	Description
OTP_SW2_VOLT	2:0	Sets the SW2 output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to the respective SW2 output voltage configuration table in the data sheet for all possible configurations.
OTP_SW2_HI	3	0: LOW output voltage range selected (1.50 V to 1.85 V) 1: HIGH output voltage range selected (2.50 V to 3.3 V)
UNUSED	7:4	unused

Table 12. OTP SW2 SEQ register description

Name	Bit	Description
OTP_SW2_SEQ	2:0	assigns the power-up sequence slot 0 to 7 for SW2
UNUSED	7:3	unused

Table 13. OTP SW2 CONFIG register description

Name	Bit	Description
OTP_SW2_FREQ	1:0	SW2 OTP frequency configuration 00 = 1.0 MHz 01 = 2.0 MHz 10 = 4.0 MHz 11 = reserved
UNUSED	7:2	unused

Table 14. OTP SW3 VOLT register description

Name	Bit	Description
OTP_SW3_VOLT	3:0	Sets the SW3 output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to SW3 output voltage configuration table on data sheet for all possible configurations.
UNUSED	7:4	unused

Table 15. OTP SW3 SEQ register description

Name	Bit	Description
OTP_SW3_SEQ	2:0	assigns the power-up sequence slot 0 to 7 for SW3
UNUSED	7:3	unused

Table 16. OTP SW3 CONFIG register description

Name	Bit	Description
OTP_SW3_FREQ	1:0	SW3 OTP frequency configuration 00 = 1.0 MHz 01 = 2.0 MHz 10 = 4.0 MHz 11 = reserved
UNUSED	7:2	unused

Table 17. OTP SWBST VOLT register description

Name	Bit	Description
OTP_SWBST_VOLT	1:0	SWBST OTP output voltage set point 00 = 5.00 V 01 = 5.05 V 10 = 5.10 V 11 = 5.15 V
UNUSED	7:2	unused

Table 18. OTP SWBST SEQ register description

Name	Bit	Description
SWBST_SEQ	4:0	assign the power-up sequence slot 0 to 31 for SWBST
UNUSED	7:5	unused

Table 19 shows a summary of all the registers related to the linear regulators, and Table 20 to Table 22 provide a general bit description of the linear regulator OTP registers.

Table 19. OTP linear regulators register summary

Register	Address	Output
OTP VSNVS VOLT	0xC0	VSNVS OTP output voltage set point
OTP LDO1 VOLT	0xC8	LDO1 OTP output voltage set point
OTP LDO1 SEQ	0xC9	LDO1 OTP power-up sequence selection
OTP LDO2 VOLT	0xCC	LDO2 OTP output voltage set point
OTP LDO2 SEQ	0xCD	LDO2 OTP power-up sequence selection
OTP VSD VOLT	0xD0	VSD OTP output voltage set point
OTP VSD SEQ	0xD1	VSD OTP power-up sequence selection
OTP V33 VOLT	0xD4	V33 OTP output voltage set point
OTP V33 SEQ	0xD5	V33 OTP power-up sequence selection
OTP LDO3 VOLT	0xD8	LDO3 OTP output voltage set point
OTP LDO3 SEQ	0xD9	LDO3 OTP power-up sequence selection
OTP LDO4 VOLT	0xDC	LDO4 OTP output voltage set point
OTP LDO4 SEQ	0xDD	LDO4 OTP power-up sequence selection

Table 20. OTP VSNVS VOLT register description

Name	Bit	Description
OTP_VSNVS_VOLT	2:0	Sets the VSNVS output voltage to be programmed on the OTP fuses and loaded during power-up 000 = RSVD 001 = RSVD 010 = RSVD 011 = RSVD 100 = RSVD 101 = RSVD 110 = 3.0 V 111 = RSVD
UNUSED	7:3	unused

Table 21. OTP LDOx VOLT register description (LDO1, LDO2, LDO3 and LDO4)

Name	Bit	Description
OTP_LDOx_VOLT	3:0	Sets the LDOx output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to the LDOx output voltage configuration table on data sheet for all possible configurations.
UNUSED	7:4	unused

Table 22. OTP LDOx SEQ register description

Name	Bit	Description
OTP_LDOx_SEQ	3:0	assign the power-up sequence slot 0 to 31 for the specific linear regulator
UNUSED	7:4	unused

Table 23. OTP VSD VOLT register description

Name	Bit	Description
OTP_VSD_VOLT	1:0	Sets VSD output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to the VSD output voltage configuration table on data sheet for all possible configurations.
UNUSED	7:2	unused

Table 24. OTP VSD SEQ register description

Name	Bit	Description
OTP_VSD_SEQ	3:0	assign the power-up sequence slot 0 to 31 for the specific linear regulator
UNUSED	7:4	unused

Table 25. OTP V33 VOLT register description

Name	Bit	Description
OTP_V33_VOLT	1:0	Sets V33 output voltage to be programmed on the OTP fuses and loaded during power-up. Refer to the V33 output voltage configuration table on data sheet for all possible configurations.
UNUSED	7:2	unused

Table 26. OTP V33 SEQ register description

Name	Bit	Description
OTP_V33_SEQ	3:0	assign the power-up sequence slot 0 to 31 for the specific linear regulator
UNUSED	7:4	unused

4.4.2 Other OTP bits

Table 27. OTP PU CONFIG1 bits definition

Bit	Name	Description
0	OTP_SEQ_CLK_SPEED	sequence delay between steps 0 = 500 µs 1 = 2000 µs
1	RSVD	reserved
2	OTP_SWDVS_CLK	startup slew rate 0 = 25 mV step each 2.0 µs 1 = 25 mV step each 4.0 µs
3	RSVD	reserved
4	OTP_EN_CFG	set the power on button initial configuration 0 = power button is level sensitive 1 = power button is edge sensitive and turn-off is based on time held LOW
7:5	RSVD	reserved

Table 28. OTP_FUSE_POR1 bits definition

Bit	Name	Description
6:0	RSVD	reserved
7	TBB_POR	prototyping enable bit 0 = prototyping disabled 1 = prototyping enabled

Table 29. OTP PWRGD EN bits definition

Bit	Name	Description
0	OTP_PG_EN	fault mode enable bit 0 = fault mode disabled 1 = fault mode enabled
7:1	RSVD	reserved

Table 30. OTP BLOWN bits definition

Bit	Name	Description
0	OTP_BLOWN	OTP fuses blown status 0 = OTP fuses not blown 1 = OTP fuses blown
7:1	RSVD	reserved

4.4.3 TBBOTP register reloading without turn-on event

After the fuses are programmed, their values may be loaded into the digital control logic without toggling VIN or EN. To update the TBBOTP registers by reloading the fuse values automatically, set bits in the OTP LOAD MASK register depending on the functionality required. Refer to Table 31 for a description of the OTP LOAD MASK register.

Table 31. OTP load mask register

Extended page 1		I^2C-bus data bits							
Address	Register name	7	6	5	4	3	2	1	0
84	OTP LOAD MASK	START	RL_PWRTN	FORCE_PWRCTL	RL_PWRCTL	RL_OTP	RL_OTP_ECC	RL_OTP_FUSE	RSVD
		0	0	0	0	0	0	0	0

Table 32. OTP reload mask register bit description

Bit	Name	Description
0	RSVD	reserved
1	RL_OTP_FUSE	reload the OTP fuse latch from the analog fuse bit 0 = disable loading 1 = enable loading
2	RL_OTP_ECC	Reload the OTP ECC registers. Set this bit irrespective of whether ECC is enabled or disabled. 0 = disable loading 1 = enable loading
3	RL_OTP	reload the TBBOTP registers from the fuses 0 = disable loading 1 = Enable loading of fuses if ECC is disabled. Enable loading of ECC corrected fuses if ECC is enabled.

Bit	Name	Description
4	RL_PWRCTL	reload the power control registers from the TBBOTP registers 0 = disable loading 1 = enable loading
5	FORCE_PWRCTL	forces the power control registers to be reloaded if they are being used to control the regulators 0 = no reload forced 1 = Power control register value affects regulators when the reload sequence is enabled and RL PWRCTL bit is enabled. This is needed when changing output voltage of switching regulators from low-voltage range to high-voltage range.
6	RL_PWRTN	Reloads the register that controls how the EN button works 0 = EN configuration setting does not change until a shutdown and restart event 1 = EN behavior switch to new OTP EN button configuration when START bit is enabled
7	START	reload sequence start bit 0 = reload sequence disabled 1 = starts the reload sequence, when the sequence is done all of the OTP_LOAD_MASK bits are reset

Often only bits 1, 2, and 3 need to be set, as well as the START bit, to reload the TBBOTP registers after the fuses are programmed. Then, check the TBBOTP register values to make sure that the correct values have been loaded from the fuses. Setting bits 4 and 5, updates the regulator parameters immediately. This should be done with caution if EN is already asserted. A EN event triggers a complete reload using the same logic. When a '1' is written to bit 7 of the OTP_LOAD_MASK registers, the VR5100 turns off momentarily and then turns back on to reload the fuses. To reload the fuses without first turning off the VR5100, clear bit 0 of the PWRCTRL_OTP_CTRL register prior to writing to the OTP_LOAD_MASK register. Note that the OTP_LOAD_MASK is register 0x84 in extended page 1 whereas the PWRCTRL_OTP_CTRL is register 0x88 in extended page 2.

4.4.4 Direct OTP fuse read

Setting the OTP_FUSE_READ_EN bit to HIGH allows you to read uncorrected fuse values. If ECC is not enabled, or there is no programming error, the values loaded into the TBBOTP registers are identical to the fuse values. If ECC is enabled and a single-bit error occurs during programming, the fuse values may be different from the values loaded into the TBBOTP registers. The values loaded into the TBBOTP registers are the error-corrected values. Table 33 shows the OTP FUSE READ EN register.

Table 33. OTP fuse read enable register

Extended page 1		I^2C-bus data bits							
Address	Register name	7	6	5	4	3	2	1	0
80	OTP FUSE READ EN	—	—	—	—	—	—	—	OTP_FUSE_READ_EN
		0	0	0	0	0	0	0	0

5 Fuse programming and error correction code (ECC)

5.1 OTP fuse control register

Section 4.2 "OTP programming example" provides an example of a typical OTP programming script. You must write to the OTP_FUSE_CTRLx registers, located in the extended page 2, in order to program fuses. There are ten such registers, one for each bank. See Table 34 and Table 35 for a description of the registers.

Table 34. General OTP fuse control register bits

Extended page 2	I^2C-bus data bits							
Register name	7	6	5	4	3	2	1	0
OTP_FUSE_CTRLx	—	—	—	—	ANTIFUSEx_EN	ANTIFUSEx_LOAD	ANTIFUSEx_RW	BYPASSx

Table 35. OTP fuse control bits description

Bit	Name	Description
0	BYPASSx	multiplexor that selects between the value stored in the digital fuse latch and the value on the TBBOTP register 0 = select from digital latch 1 = select from TBBOTP register
1	ANTIFUSEx_RW	allows programming the fuse bank when VDDOTP is 9.5 V 0 = disable program fuse 1 = enable program fuse
2	ANTIFUSEx_LOAD	Clock input to the digital latch that stores the state of the analog fuse cell, it is active HIGH and is pulsed while the ANTIFUSE_EN bit is HIGH to load the value of the analog fuse state into the digital latch.
3	ANTIFUSEx_EN	turns on the bias to the analog fuse cell so that it can be written to or read from 0 = analog bias disabled 1 = analog bias enabled
4:7	not used	not used

5.2 Error correction code (ECC)

Error correction is enabled for bank 1 and bank 2 and disabled for the other banks by default. However, NXP recommends that it is enabled (set to ON) for all programming operations. When error correction is enabled, a single-bit error per fuse bank is both reported and corrected. A double-bit error per fuse bank is reported but not corrected. Fuses may be programmed without using ECC. However, after verifying that the part is configured properly, ECC may be enabled and the error check bits are programmed.

Double-bit errors can prevent regulators from powering up, or can result in a configuration that does not match the external components. Although such occurrences are rare, it is still a good practice to employ ECC to at least alert the user when this occurs.

Note: *The desired function of the redundant bits must be determined when ECC is configured and its bits are programmed, or the ECC logic attempts to correct the newly programmed redundant bits.*

Section 5.2.1 "ECC interrupt" through Section 5.2.3 "Fuse programming with ECC" are for advanced users. For a simple script that enables ECC, see Section 4.2 "OTP programming example".

5.2.1 ECC interrupt

With ECC enabled, if a single fuse in a bank has the wrong value, the ECC logic corrects that bit and loads the corrected value into the TBBOTP register for that bank. The single-error bit for that bank is set and also the main interrupt ECC bit is set. If two or more bits are in error, in a bank, the ECC is not able to correct them. The double-error bit error for that bank is set and the ECC interrupt bit is set. The single-error and double-error bits may be read from registers 0x8A to 0x8D in the extended page 1 of the register map. The ECC interrupt bit may be read from register, 0xE, on the functional page of the register map.

Table 36. ECC error detection registers

Extended page 1		I²C-bus data bits							
Address	Register name	7	6	5	4	3	2	1	0
8A	OTP ECC SE1	—	—	—	ECC5_SE	ECC4_SE	ECC3_SE	ECC2_SE	ECC1_SE
		X	X	X	0	0	0	0	0
8B	OTP ECC SE2	—	—	—	—	—	—	—	—
		X	X	X	X	X	X	X	X
8C	OTP ECC DE1	—	—	—	ECC5_DE	ECC4_DE	ECC3_DE	ECC2_DE	ECC1_DE
		X	X	X	0	0	0	0	0
8D	OTP ECC DE2	—	—	—	—	—	—	—	—
		X	X	X	X	X	X	X	X

Table 37. OTP ECC SE1 register description

Bit	Name	Default	Description
0	ECC1_SE	0	single error detection in fuse bank 1 0 = no single error detected 1 = single error detected
1	ECC2_SE	0	single error detection in fuse bank 2 0 = no single error detected 1 = single error detected
2	ECC3_SE	0	single error detection in fuse bank 3 0 = no single error detected 1 = single error detected
3	ECC4_SE	0	single error detection in fuse bank 4 0 = no single error detected 1 = single error detected
4	ECC5_SE	0	single error detection in fuse bank 5 0 = no single error detected 1 = single error detected
7:5	RSVD	0	reserved

Table 38. OTP ECC DE1 register description

Bit	Name	Default	Description
0	ECC1_DE	0	dual error detection in fuse bank 1 0 = no single error detected 1 = single error detected
1	ECC2_DE	0	dual error detection in fuse bank 2 0 = no single error detected 1 = single error detected
2	ECC3_DE	0	dual error detection in fuse bank 3 0 = no single error detected 1 = single error detected
3	ECC4_DE	0	dual error detection in fuse bank 4 0 = no single error detected 1 = single error detected
4	ECC5_DE	0	dual error detection in fuse bank 5 0 = no single error detected 1 = single error detected
7:5	RSVD	0	reserved

All interrupts are masked by default. Therefore, after programming the fuses, you should unmask the ECC interrupt to determine if single-bit or double-bit errors exist in any of the banks. To read the error bits, see Table 36 for their location in the registers.

5.2.2 Analyzing a single-bit ECC error

When a single bit error occurs, the ECC check bits indicate which fuse in a given bank is in error. If you require ECC error information, read the check bits for each bank from bits[5:0], in registers 0xE1 to 0xEA, in the extended page 2; see Table 39. For example, if there is an error in bit[5] of fuse bank 3, reading bits[5:0] of register 0xE3 yields a hexadecimal code of 0x15. Refer to Table 41 and Table 42 for a description of the error control registers.

Table 39. ECC error location coding

Bit in error	ECC check bit code
0	07
1	0B
2	0D
3	0E
4	13
5	15
6	16
7	19
8	1A
9	1C
10	23

Bit in error	ECC check bit code
11	25
12	26
13	29
14	2A
15	2C
16	31
17	32
18	34
19	38
20	01
21	02
22	04
23	08
24	10
25	20

5.2.3 Fuse programming with ECC

Table 40 shows the OTP AUTO ECC0 register. After programming the fuses, load their values into the TBBOTP registers. Load the error-corrected values if there was a single bit error in any bank. To view the uncorrected or raw fuse values, see Section 4.4.4 "Direct OTP fuse read". To determine if an error occurred while programming fuses, do any of the following:

- Check the fuse values against what was written.
- Monitor the INTB signal, but first the ECC interrupt must be unmasked.
- Read bits[5:0] from the OTP ECC CTRLx registers in the extended page 2. See Table 39 to decipher single bit error codes and Table 42 for a description of the ECC registers.

Table 40. Automatic ECC mode enable register

Extended page 2		I^2C-bus data bits							
Address	Name	7	6	5	4	3	2	1	0
D0	OTP AUTO ECC0	—	—	—	AUTO_ECC_ BANK5	AUTO_ECC_ BANK4	AUTO_ECC_ BANK3	AUTO_ECC_ BANK2	AUTO_ECC_ BANK1
		—	—	—	0	0	0	1	1

Table 41. ECC control registers in the extended page 2

Address	Name	D7	D6	D5	D4	D3	D2	D1	D0
E1	ECC_CTRL1	RSVD	ECC1_CALC_CIN	ECC1_CIN_TBB[5:0]					
		0	0	0	0	0	0	0	0
E2	ECC_CTRL2	RSVD	ECC2_CALC_CIN	ECC2_CIN_TBB[5:0]					
		0	0	0	0	0	0	0	0
E3	ECC_CTRL3	RSVD	ECC3_CALC_CIN	ECC3_CIN_TBB[5:0]					
		0	0	0	0	0	0	0	0
E4	ECC_CTRL4	RSVD	ECC4_CALC_CIN	ECC4_CIN_TBB[5:0]					
		0	0	0	0	0	0	0	0
E5	ECC_CTRL5	RSVD	ECC5_CALC_CIN	ECC5_CIN_TBB[5:0]					
		0	0	0	0	0	0	0	0

Table 42. ECC_CTRLx registers description

Bit	Name	Default	Description
5:0	ECCx_CIN_TBB	0	ECC error location code; see codes in Table 39
6	ECCx_CALC_CIN	0	calculate the ECC check bit values 0 = calculation disabled 1 = calculation enabled
7	RSVD	0	reserved

6 References

To obtain more information on NXP products and application solutions, go to the following URLs:

Support page	Description	URL
VR5100	product summary page	http://www.nxp.com/VR5100
Analog and power management	homepage	http://www.nxp.com/ANALOG
Power management integrated circuits (PMICs)	homepage	http://www.nxp.com/PMIC

7 Legal information

7.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/ or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/salestermsandconditions.

7.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

NXP — is a trademark of NXP B.V.

the NXP logo — is a trademark of NXP B.V.

Freescale — is a trademark of NXP B.V.

the Freescale logo — is a trademark of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6 April 2017
Document identifier: AN5403