Radiation-hardened SPI NOR flash (CYRS17B512) configuration options for Kintex UltraScale FPGAs from AMD

About this document

Scope and purpose

This document describes how to use the Infineon radiation-hardened CYRS17B512 SPI flash devices to configure the Kintex UltraScale family of devices from AMD.

A specific patched version of the Vivado IDE is required to use the CYRS17B512 NOR flash with AMD devices. This application note focuses on programming and configuration using the CYRS17B512 device. It does not provide information when using CYRS17B01G or CYRS17B02G devices as they are not currently supported in the Vivado IDE for AMD FPGAs.

For further technical support, contact the team.

Intended audience

This document is intended for system integrators and developers to build and deploy systems using AMD Kintex UltraScale FPGAs with Infineon radiation-hardened NOR flash devices for configuration memory.

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **3**
2 SPI connection basics **4**
3 SPI NOR flash connections **6**
4 Programming options **8**
4.1 Programming based on the toolchain from AMD 8
4.1.1 Bit file generation 8
4.1.2 Programming the flash device 12
4.2 Direct programming the SPI NOR flash 13
4.3 Standalone SPI NOR flash programmer 14
References **16**
Revision history **17**
Disclaimer **18**

1 Introduction

Infineon's SPI flash programming solutions for the CYRS17B512 device is now available for use with aerospace- and defense-rated FPGAs from AMD. These radiation-hardened (Rad Hard) SPI flash devices can be used in single SPI (x1), quad SPI (x4), or Dual Quad SPI (2x4) configuration options. Currently, only the Rad Hard SPI NOR flash devices with 512-Mbit density are supported for the configuration solution.

There are two programming options for CYRS17B512:

- **Vivado IDE: Vivado version 2024.1 with a software patch (Vivado v2024.1_AR000038133)**. The patch and installation instructions will be provided by AMD after opening a customer support case with AMD requesting the software patch. Only the Vivado 2024.1 version of the Vivado IDE works with this software patch
- **Standalone programmer:** Standalone SPI NOR flash programming solution that uses a standalone NOR flash programmer from EmbeddedComputers

Along with the programming options listed above, Infineon also provides a mezzanine board (IFXDK-RHNOR-V) for the Rad Hard NOR flash configuration solution that is compatible with the alpha data development platform for space-grade FPGAs (ADA-SDEV-KIT2). This mezzanine card is set up to use the programming approach based on the Vivado toolchain from AMD.

Because the Vivado 2024.1 software patch was developed specifically to support the Infineon Rad Hard SPI NOR flash, there are limitations as to which programming cables will work with the software patch. The table below shows the supported/unsupported programming cables.

Table 1 Supported/unsupported programming cables

Programming cable	Is the cable supported?
AMD programming cable	Supported
Digilent programming cable	Supported
SmartLynq data cable	NOT supported

2 SPI connection basics

Serial Peripheral Interface (SPI) is a simple 4-wire synchronous interface protocol that enables a master device and one or more slave devices to intercommunicate. The SPI bus consists of four signal wires:

- The Master Out Slave In (MOSI) signal generated by the master (data out to the slave)
- The Master In Slave Out (MISO) signal generated by the slave (data in to the master)
- The Serial Clock (SCK) signal generated by the master to synchronize data transfers
- The Slave Select (SS#) signal generated by the master to select individual slave devices; also known as Chip Select (CS#) or Chip Enable (CE#)

The Quad Serial Peripheral Interface (QSPI) expands the data bus to four bits and adds QSPI commands. Two QSPI devices can be used in parallel to provide an 8-bit data bus for the SPI Dual Quad SPI (DQSPI) configuration interface; an additional Slave Select is also used for this connection.

Following the SPI/QSPI/DQSPI protocol, the master is assigned to the FPGA device and the slave to the SPI NOR flash device, as shown in Figure 1. Per these connections, the SPI NOR flash is available to configure the FPGA at power-up.

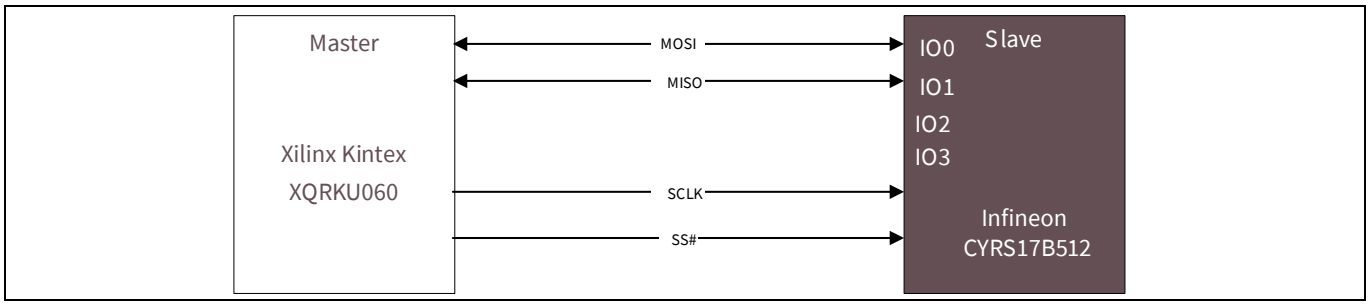


Figure 1 Direct configuring FPGA to interface with SPI NOR flash

Figure 2 shows the SPI NOR flash device connection when used in Quad SPI connection mode.

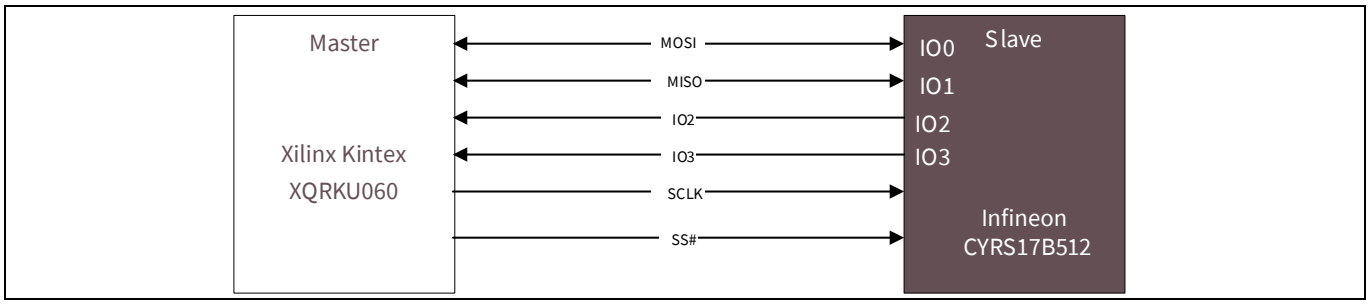


Figure 2 QSPI configuration connection with SPI NOR flash

Figure 3 shows the Dual Quad SPI device connection using two Quad SPI NOR flash devices in parallel.

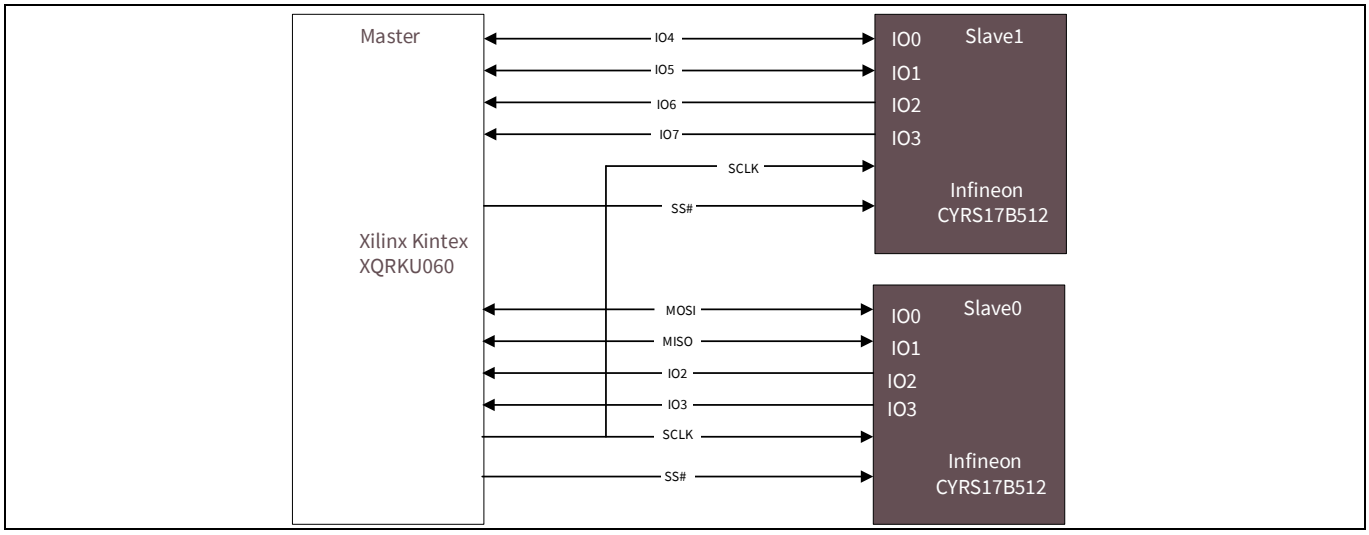


Figure 3 DQSPI configuration connection with 2 x 512 Mb SPI NOR flash

3 SPI NOR flash connections

A connection table for SPI NOR flash SPI pins to FPGA configuration pins is shown below. Some FPGA pins will not have a corresponding SPI NOR flash pin but are essential in configuring the FPGA to the correct mode and I/O voltage level. CYRS17B512 devices support 1.8 V and 3.3 V I/O CMOS voltage levels.

The pin connection table is based on the Infineon CYRS17B512 devices.

Table 2 Pin connections

SPI x1 pin name	SPI x4 pin name	SPI x8 pin name	FPGA pin name	Description
SI/IO0	SI/IO0	SI/IO0	D00_MOSI	Master Out Slave In, when in 1-bit SPI mode. D[0] of D[3:0] when in Quad SPI mode
SO/IO1	SO/IO1	SO/IO1	D01_DIN	Slave Out Master In when in 1-bit SPI mode. D[1] of D[3:0] when in Quad SPI mode. D[1] of D[7:0] when in Dual Quad SPI mode
NA	IO2	IO3	D02	D[2] of D[3:0] when in Quad SPI mode. D[2] of D[7:0] when in Dual Quad SPI mode
NA	IO3	IO4	D03	D[3] of D[3:0] when in Quad SPI mode. D[3] of D[7:0] when in Dual Quad SPI mode
NA	NA	IO4	D04	D[4] of D[7:0] when in Dual Quad SPI mode
NA	NA	IO5	D05	D[5] of D[7:0] when in Dual Quad SPI mode
NA	NA	IO6	D06	D[6] of D[7:0] when in Dual Quad SPI mode
NA	NA	IO7	D07	D[7] of D[7:0] when in Dual Quad SPI mode
CS#	CS#	CS1#	FCS_B	Active LOW Chip Select for D[3:0], connect with a pull-up to VCCO_0 ≤ 4.7 kΩ
NA	NA	CS2#	FCS2_B	Active LOW Chip Select for D[7:4], connect with a pull-up to VCCO_0 ≤ 4.7 kΩ
SCK	SCK	SCK1#	CCLK	FPGA-sourced configuration clock
NA	NA	SCK2#	–	In DQSPI mode, connect this clock pin on the SPI flash to CCLK
–	–	–	CFGBVS	This pin's value determines the I/O operating voltage range of Bank0. Tie this pin to the corresponding value to operate at 1.8 V or 3.3 V on the I/O

SPI x1 pin name	SPI x4 pin name	SPI x8 pin name	FPGA pin name	Description
–	–	–	M[2:0]	Mode configuration bits set to M[2:0] =001, master SPI
–	–	–	PROGRAM_B	Active LOW input. When pulsed LOW, the FPGA configuration is cleared, and new configuration is initiated
–	–	–	INIT_B	Active LOW FPGA initialization pin. After initialization is completed, the FPGA device will move this pin to High-Z state and begin the FPGA configuration. External masters can force this pin LOW to delay the configuration by driving it LOW actively. This pin is connected with a pull-up to VCCO_0 ≤ 4.7 kΩ
–	–	–	DONE	When DONE is HIGH, it indicates that the FPGA programming was completed. This pin is connected with a pull-up to VCCO_0 ≤ 4.7 kΩ

4 Programming options

There are three ways to program the SPI NOR flash; they are discussed in detail in the following sections:

- **Programming based on the Vivado toolchain from AMD:** This is achieved by communicating to the FPGA through the JTAG interface and programming the flash indirectly through the FPGA
- **Direct programming the SPI NOR flash:** An additional master is connected to the SPI bus
- **Standalone SPI NOR flash programmer:** The SPI NOR flash is programmed using a third-party standalone NOR flash programmer

4.1 Programming based on the toolchain from AMD

The Vivado IDE from AMD can be used to generate a bit file to program through the JTAG interface of the FPGA using the programming cable from AMD.

The Vivado 2024.1 + software patch from AMD supports the generation of the bit file and programming for the Kintex XCKU060 FPGA from AMD. In this mode, the tools program the bit file through the configuration cable from AMD connected to the JTAG interface of the FPGA. The FPGA SPI configuration interface is then used to program the incoming bit file on the JTAG interface to the SPI NOR flash.

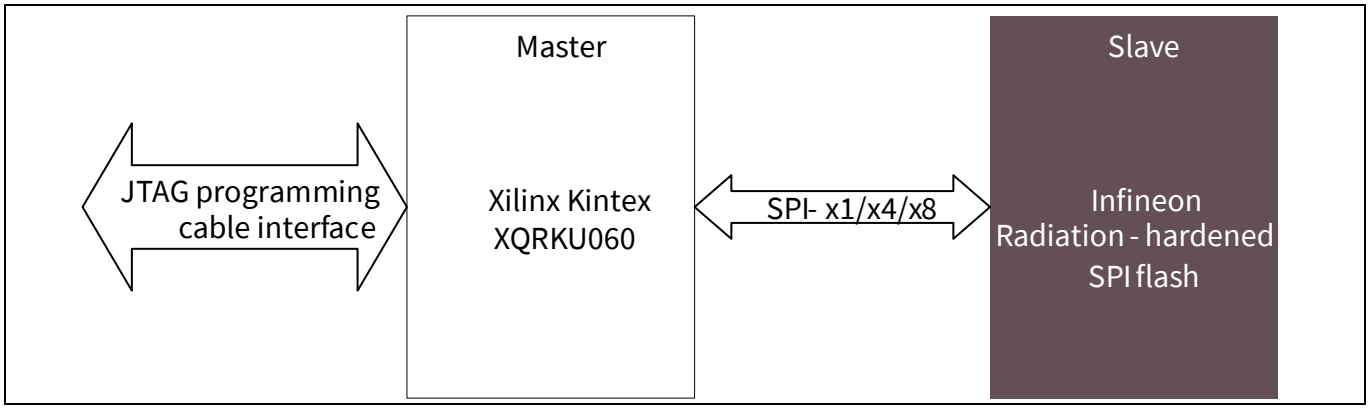


Figure 4 Programming based on the toolchain from AMD

4.1.1 Bit file generation

In the Vivado IDE, after the implementation has been completed, before generating the bit file, choose the SPI configuration method (i.e., SPI, QSP, DQSPI, master serial mode, etc.).

1. Open the implemented design in the Vivado IDE from AMD
2. Right-click on the **Generate bitstream** option and select **Bitstream Settings**

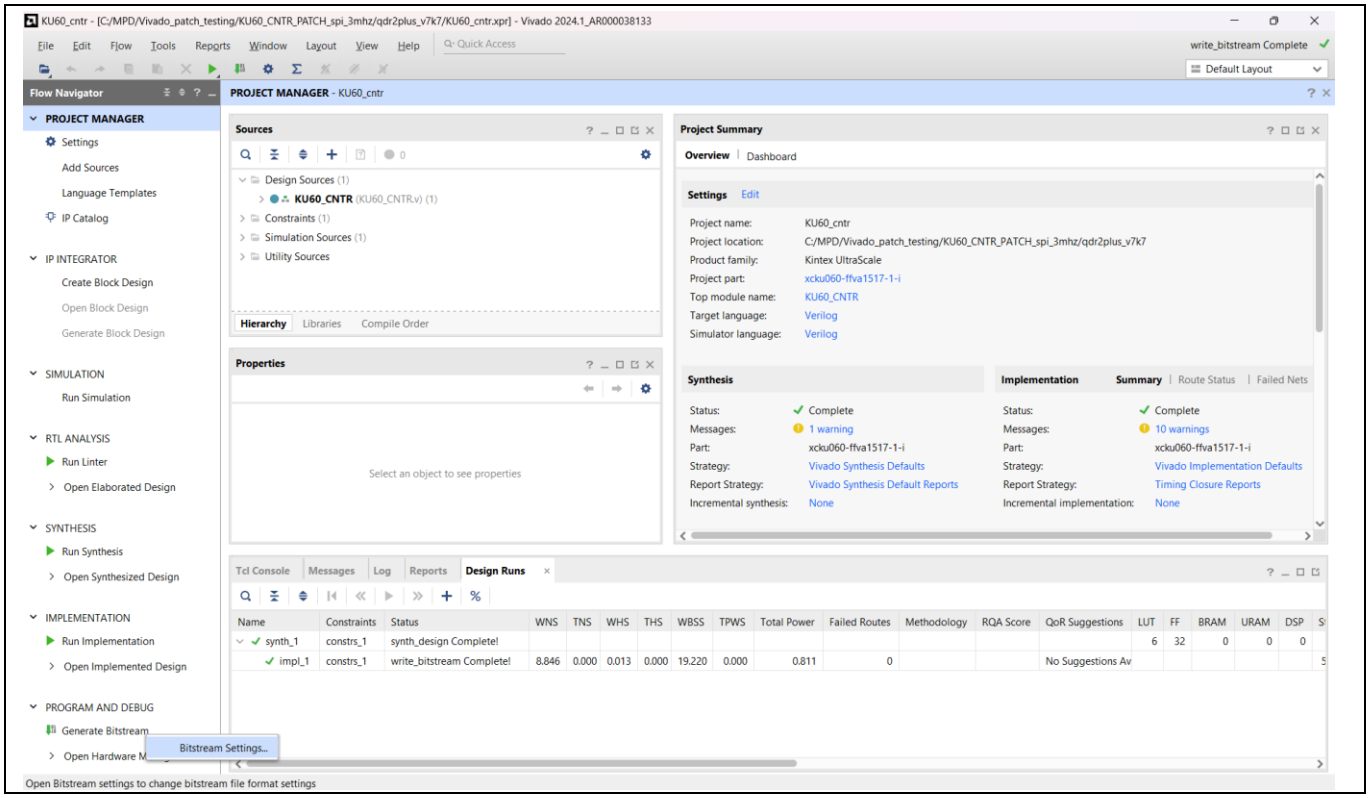


Figure 5 Bitstream configuration settings selection in Vivado IDE from AMD

3. In the pop-up window, select the *.bin* file options and click the **Configure additional bitstream settings** link

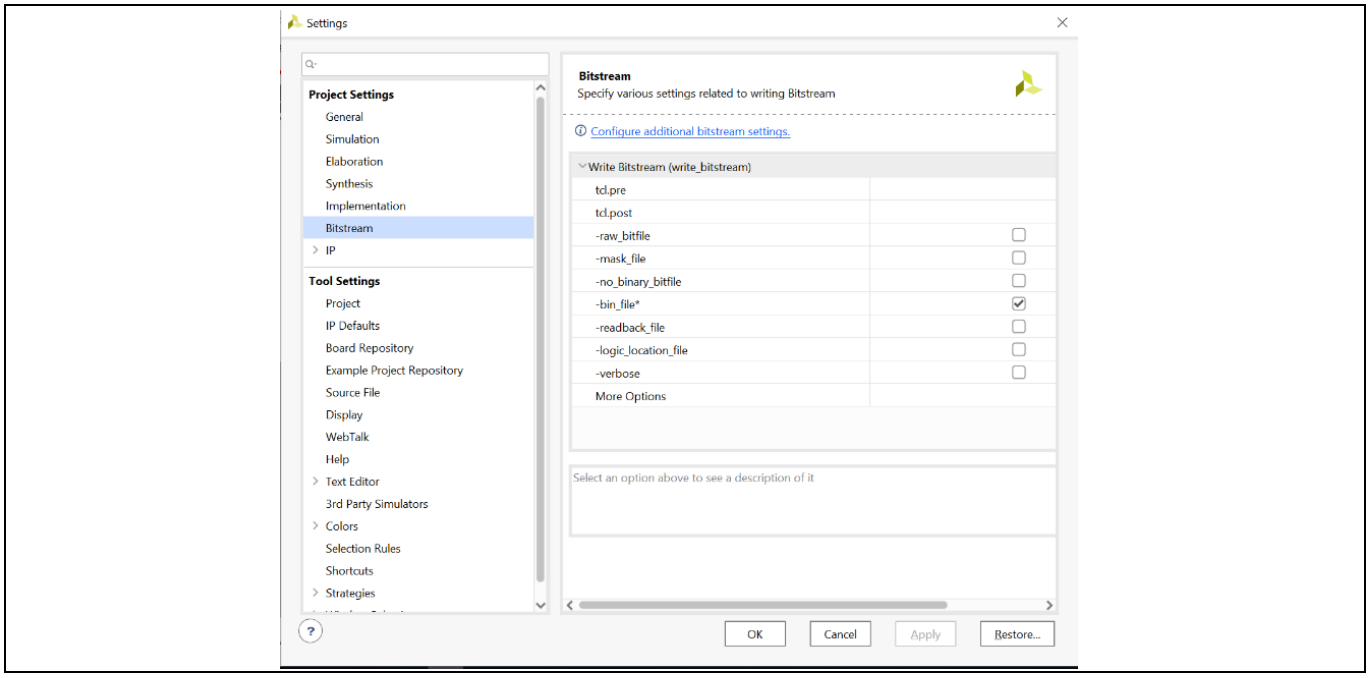


Figure 6 Configure additional bitstream settings in Vivado IDE from AMD

4. In the next window, select the **Configuration** option on the left pane. All configuration options should now be seen on the right pane of the window

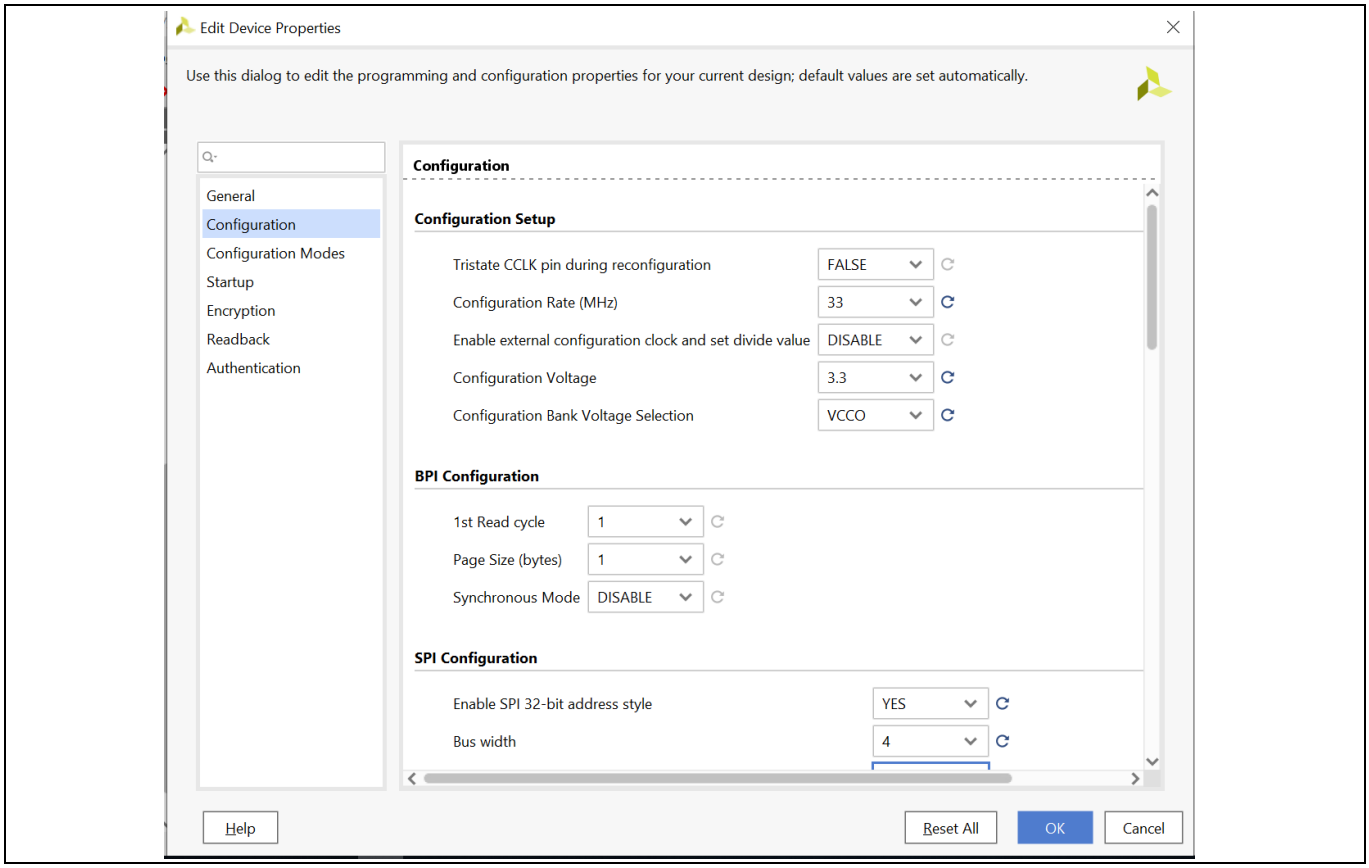


Figure 7 Configuration of clock and voltage for configuration memory in the bitstream in Vivado IDE from AMD

5. In the configuration setup, change the following for default values:

- **Configuration Rate (MHz):** 33 MHz (The clock frequency will vary depending on the board design and other factors that may limit the maximum frequency of operation for the configuration interface.)
- **Configuration Voltage:** Can be 3.3 V or 1.8 V
- **Configuration Bank Voltage Selection:** VCCO

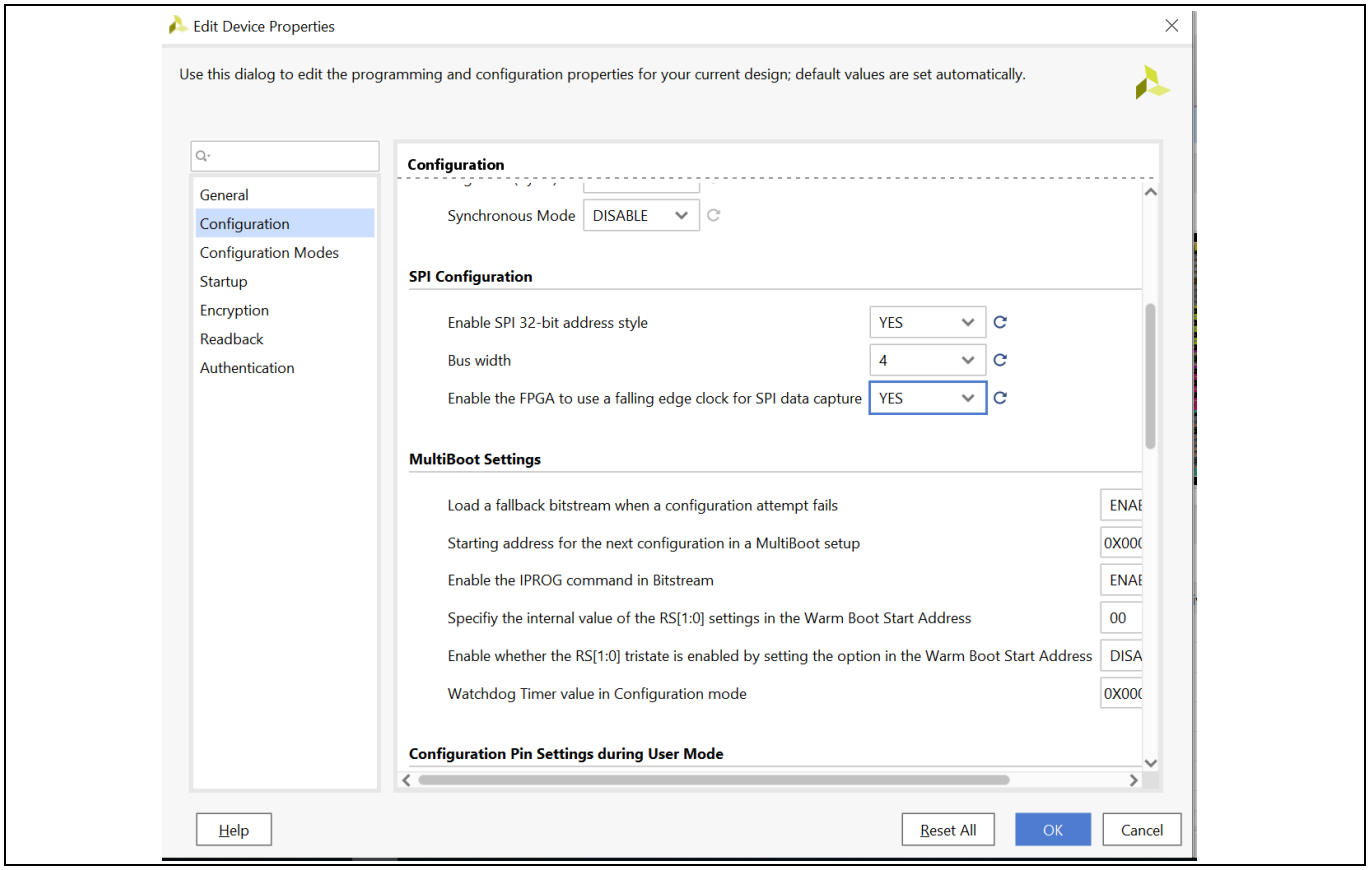


Figure 8 Configuration memory SPI interface type for bitstream transmission in Vivado IDE from AMD

6. In the SPI configuration section, change the following:
   Enable SPI 32-bit address style: YES
   Bus width: 4 (Set for QSPI; if DQSPI, set to 8.)
   Enable the FPGA to use a falling edge clock for the SPI data capture: YES
7. Select **Configuration modes** in the left panel of the window and select the corresponding SPI mode (in this case, **Master SPI x4**)
8. Click **OK** to accept the settings and then click **OK** to close the setting window

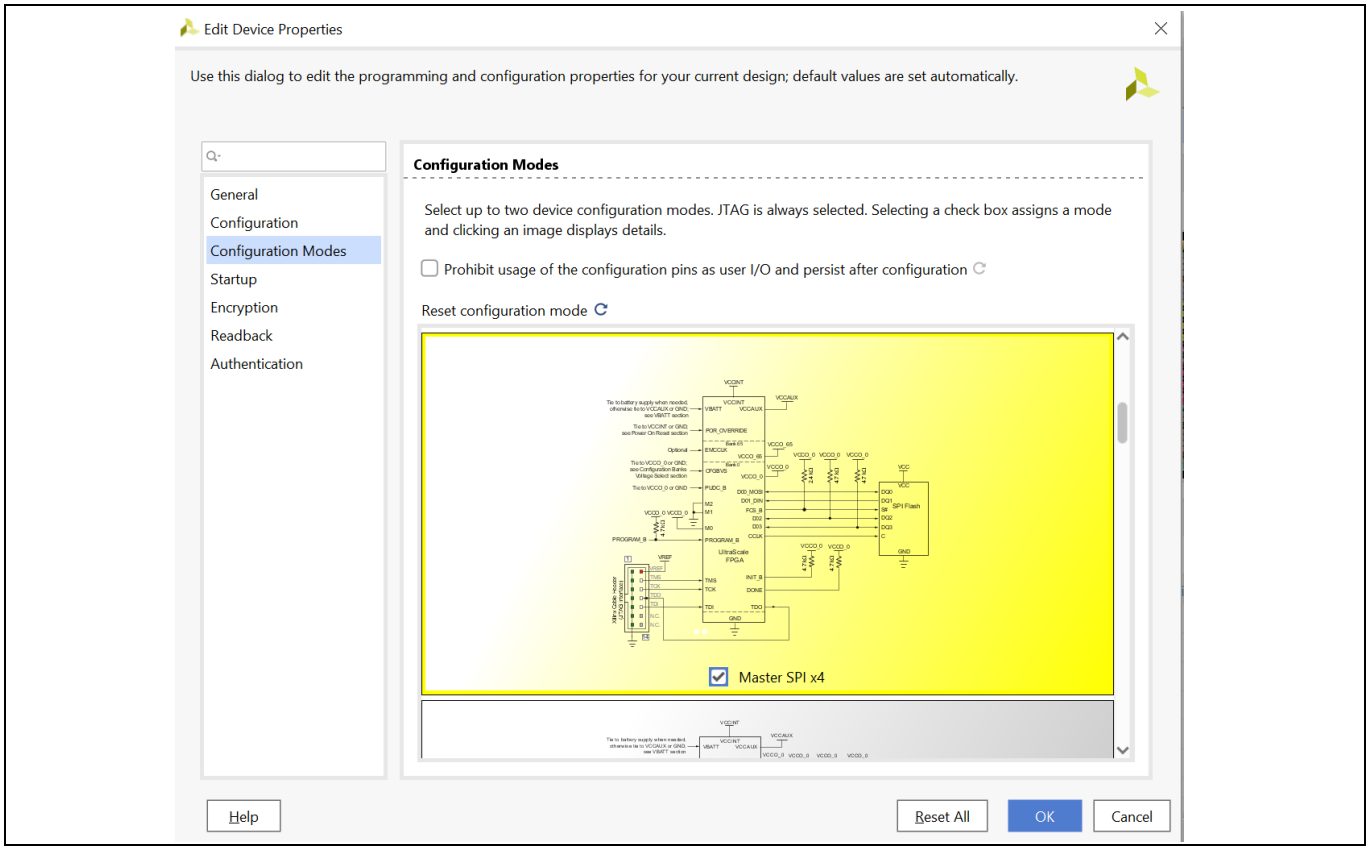


Figure 9 SPI configuration mode selection for the FPGA in Vivado IDE from AMD

9. Generate the bitstream. The *.bit* and *.bin* files will be generated with the QSPI programming options
10. Follow the same steps to generate the DQSPI programming file

4.1.2 Programming the flash device

1. Generate the bit file using the flow in the Vivado IDE flow from AMD. Note that version 2024.1 with Vivado v2024.1_AR000038133 patch is the only version of Vivado that supports the Rad-Hard SPI NOR flash natively
2. Open **Hardware Manager** in the **Program and Debug** section of the IDE
3. Connect to the hardware and set up the FPGA device from AMD. Select the XCKU060 Kintex radiation-hardened device from AMD
4. Select **cyrs17b512-spi-x1_x2_x4 or cyrs17b512-spi-x1_x2_x4_x8** to add the appropriate configuration memory

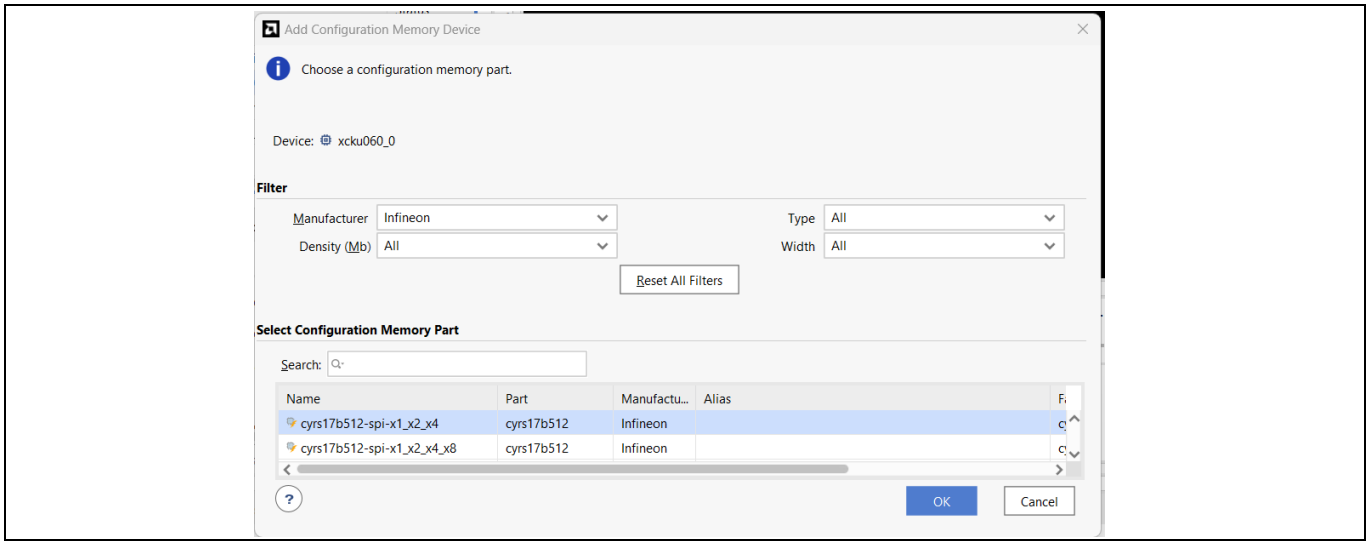


Figure 10 Infineon radiation-hardened SPI NOR flash selection in Vivado IDE from AMD

5. Follow the instructions in the IDE to select the *.bin* file and program the SPI NOR flash device

4.2 Direct programming the SPI NOR flash

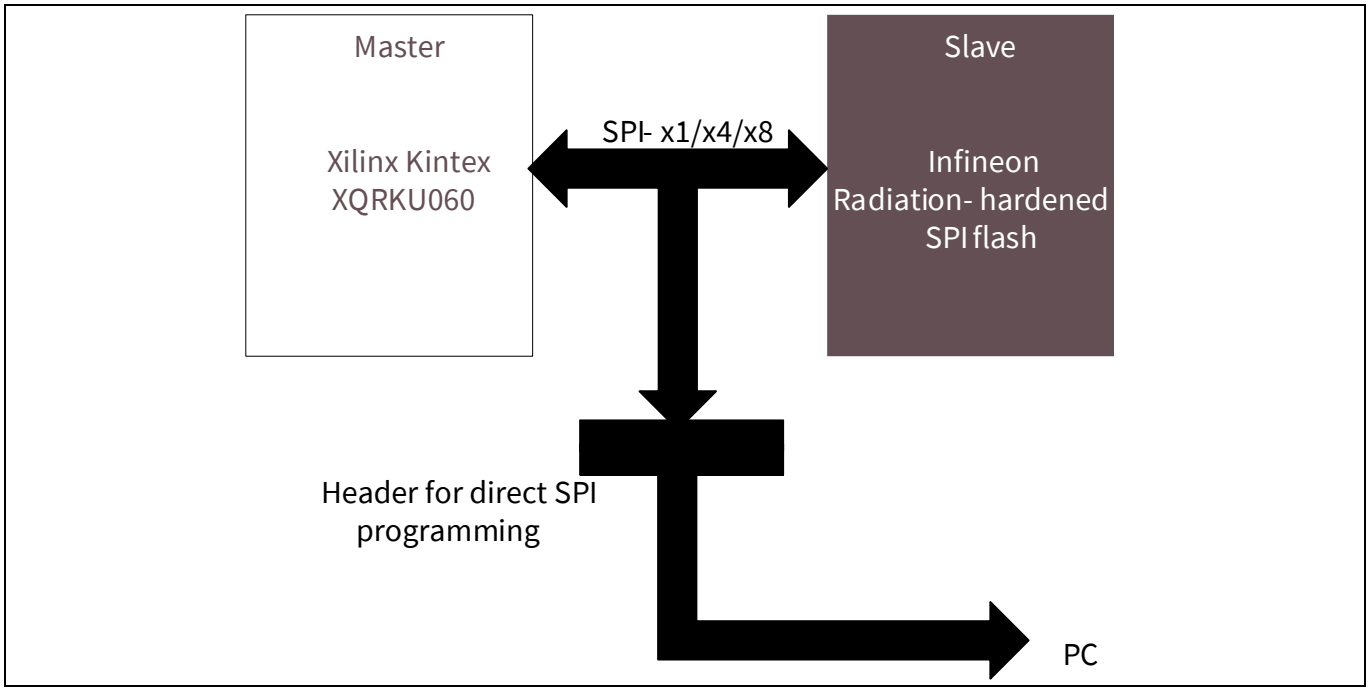


Figure 11 Direct programming of SPI NOR flash

In this mode, the SPI NOR flash can be programmed directly by another SPI master such as a microcontroller or PC.

The FPGA supplies the CCLK output from its internal oscillator to drive the clock input of the SPI NOR flash. The In-System Programming Control (ISP) signal is used to move the FPGA SPI interface signals to High-Z state during in-system programming to re-program the configuration data in the SPI NOR flash.

To read the configuration data from the SPI NOR flash at power-up, the FPGA from AMD must issue a read command to the SPI NOR flash. This is done automatically based on the MODE setting. The following figure shows the connection for this programming option.

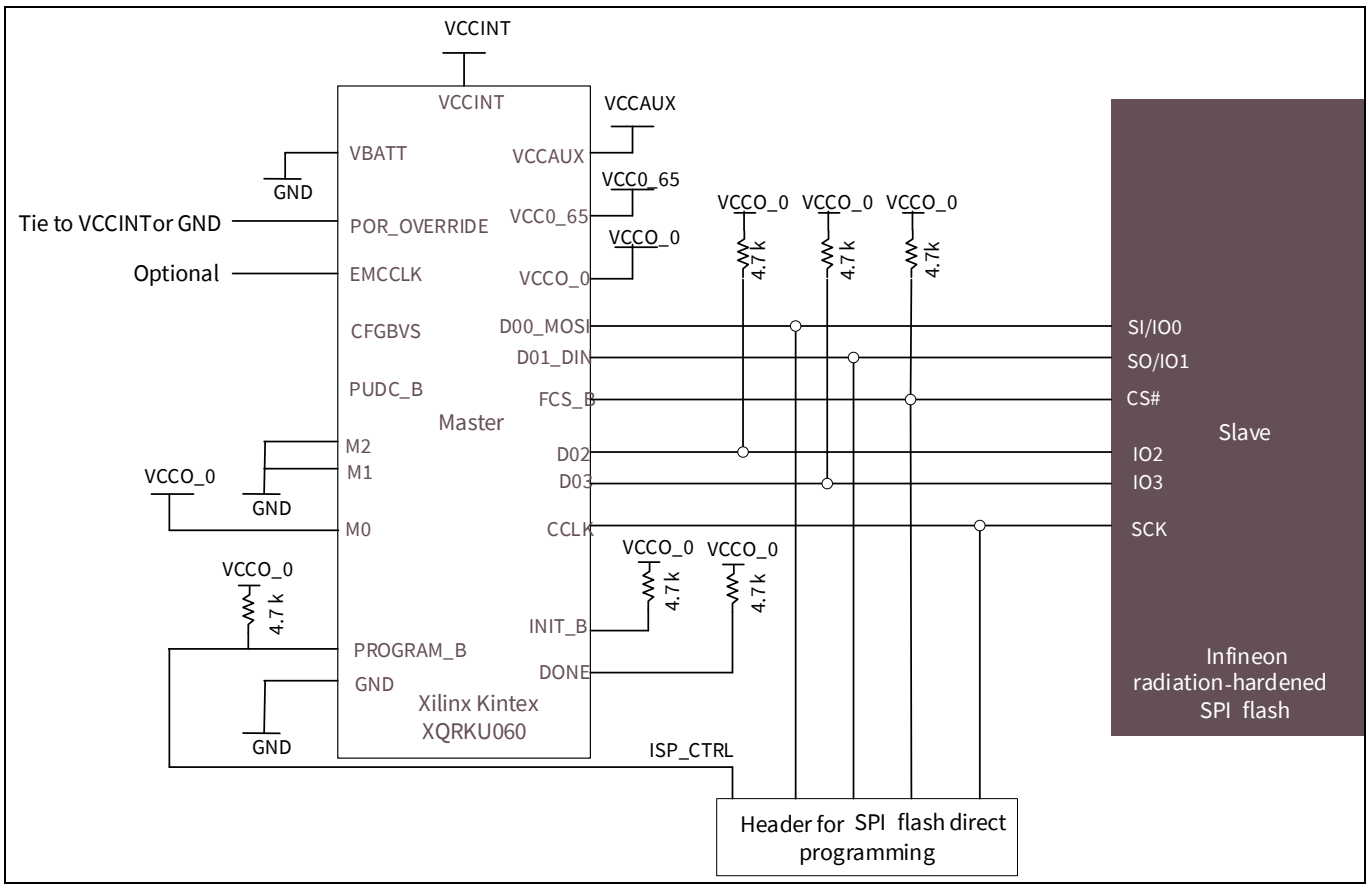


Figure 12 Connections for SPI direct programming

4.3 Standalone SPI NOR flash programmer

A standalone SPI NOR flash programmer can also be used to program the SPI NOR flash device. Generic off-the-shelf programmers are widely available such as the Rad Hard SPI NOR flash device socket board from EmbeddedComputers that will plug into their FlashcatUSB-Mach programmer.

Using the FlashcatUSB software, the SPI NOR flash device can be programmed from a computer with a USB port. Install the USB driver and run the software as instructed in the user manual. When the FlashcatUSB-Pro programmer from EmbeddedComputers is connected to the USB port, it will detect the flash device in the socket. With the flash device detected, the software will create a **Flash** tab. All flash read and write operations can be done through this tab.

The following image shows the FlashcatUSB-Pro programmer board and socket board from EmbeddedComputers.

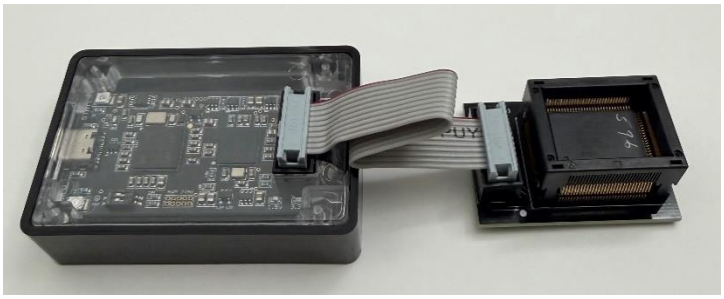


Figure 13 FlashcatUSB-Pro programmer + socket board from EmbeddedComputers

In the Vivado IDE from AMD, generate a *.bin* file that can be used by the FlashcatUSB software from EmbeddedComputers. Do the following to program the SPI NOR flash:

1. Run the *FlashcatUSB.exe* program
2. Connect the FlashcatUSB programmer from EmbeddedComputers with the SPI flash plugged into the programmer board. The **Status** tab shows the software connected to the FlashcatUSB programmer from EmbeddedComputers
3. Click on the **SPI Flash** tab
4. Click **Write Data to memory**
5. On the window, select the *.bin* file
6. Verify that the number of bytes displayed in the pop-up window is correct, and then click **OK**

The software begins programming the SPI flash. Wait for the "Write operation complete” status on the bottom of the window.

References

[1] Infineon Technologies AG: *CYRS17B512 datasheet*; Available online

[2] Infineon Technologies AG: *Radiation hardened non-volatile and volatile memory portfolio webpage*; Available online

[3] EmbeddedComputers: *FlashcatUSB-Pro user guide*; Available online

[4] AMD: *UltraScale architecture configuration user guide (UG570)*; Available online

[5] AMD: *SPI configuration and flash programming in UltraScale FPGAs (XAPP1233)*; Available online

Revision history

Document revision	Date	Description of changes
**	2026-01-13	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-01-13

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-42434 Rev. **

Important Notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability.

Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury. However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer.

Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG).

If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited. For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit https://www.infineon.com