Golden System Reference Design Demo

User Guide

FPGA-UG-02205-1.1

February 2024

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AXI4	Advanced eXtensible Interface 4
BSP	Board Support Package
CRC	Cycling Redundancy Check
DMA	Direct Memory Access
DRAM	Dynamic Random Access Memory
FreeRTOS	Free Real-Time Operating System
FPGA	Field Programmable Gate Array
FW	Firmware
GHRD	Golden Hardware Reference Design
GPIO	General Purpose Input/Output
GSRD	Golden System Reference Design
LPDDR4	Low Power Double Data Rate Generation 4
MAC	Media Access Controller
MC	Memory Controller
PC	Personal computer
QSPI	Quad Serial Peripheral Interface
RISC-V	Reduced Instruction Set Computer-V
ROM	Read-Only Memory
TSE	Tri-Speed Ethernet
SGDMA	Scatter-Gather Direct Memory Access
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter

1. Introduction

The Golden System Reference Design (GSRD) is a collection of software and hardware components for developing a wide variety of end-user applications using the Lattice CertusPro™-NX FPGA. The GSRD architecture provides a base level FPGA system design that allows you to add or remove any additional soft-IPs or custom IPs, without having to build the entire FPGA system from scratch. You can focus on integration and testing your own developed blocks and scale your application accordingly. The GSRD demo also showcases various complex soft-IP features, systems, and tool capabilities to help you understand Lattice FPGA devices better.

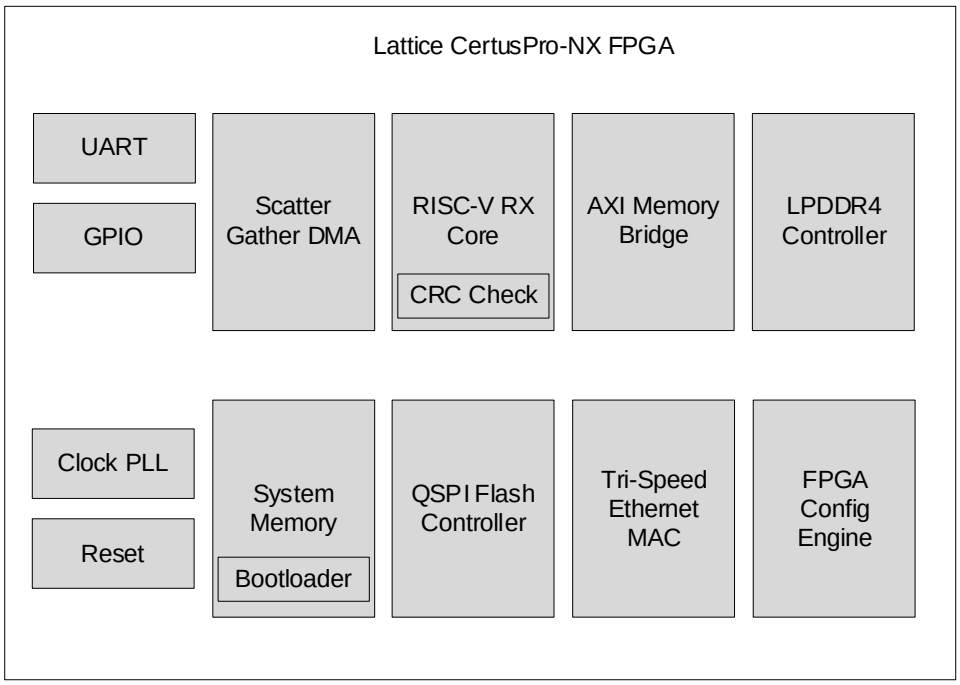

Figure 1.1. GSRD Top Level Block Diagram

The GSRD includes the following components:

- Golden Hardware Reference Design – Hardware Components:
  - RISC-V RX CPU
  - QSPI Flash Controller
  - Scatter-Gather DMA
  - LPDDR4 MC

    Note: The main building blocks are listed above. For more details, refer to the GSRD Reference Design User Guide (FPGA-RD-02283).
- Golden System Reference Design – Software Components:
  - Bootloader
  - Lattice BSP IP Drivers
  - GPIO
  - Tri-Speed Ethernet MAC
  - SGDMA
  - UART
  - QSPI
- FreeRTOS and its Drivers
- FreeRTOS Sample applications

The GSRD is enabled by the RISC-V core based FreeRTOS and Lattice FPGA IP cores. The FPGA IP device drivers (or Board Support Package) play a crucial role in facilitating the boot process of an RISC-V based FreeRTOS system. These BSP drivers, which are written in C language, serve as the interface between the hardware components of the FPGA and the software to enable seamless communication and control.

During the boot sequence, the FPGA IP device drivers initialize and configure the FPGA peripherals, ensuring proper interaction with the RISC-V processor. The initialization includes setting up memory controllers, configuring communication interfaces, and initializing essential system components. The drivers provide a standardized abstraction layer to enhance the portability of FreeRTOS on RISC-V architectures across various FPGA implementations for an efficient boot-up experience.

The list below shows the device drivers employed in the GSRD:

- GPIO (General Purpose Input/Output)
  - GPIO drivers control the input and output operations of digital pins, allowing the FPGA to interact with external devices or sensors.
- Tri-Speed Ethernet Media Access Controller (TSE MAC)
  - The TSEMAC IP core is a complex core containing all necessary logic and interfacing and clocking infrastructures necessary to integrate an external industry-standard Ethernet PHY with an internal processor efficiently with minimal overhead.
- SGDMA (Scatter-Gather Direct Memory Access) Controller
  - This driver manages the data transfer between different memory locations, enhancing efficiency by allowing non-contiguous data to be transferred without involving the CPU.
- UART (Universal Asynchronous Receiver-Transmitter)
  - UART drivers handle serial communication, enabling the FPGA to communicate with devices such as sensors, displays, or other microcontrollers.
- QSPI Flash Controller
  - This driver controls the Quad Serial Peripheral Interface Flash memory, managing read and write operations to non-volatile storage, commonly used for program storage in embedded systems.

Notes:

- For more details, refer to the respective driver API documents listed in the References section.
- For more details on the GSRD reference design, refer to the GSRD Reference Design User Guide (FPGA-RD-02283).

1.1. GSRD Flow

This user guide demonstrates how to successfully program the FPGA and FW files to showcase the GSRD data flow.

- The system provides similar functionality both in Bare Metal and FreeRTOS mode. For the purposes of the demo, the Golden and Primary Firmware binaries and the FPGA bitstream images are stored in the external SPI Flash.
- The initial bootloader code is a part of the internal system memory ROM and is a part of the FPGA bitstream stored in flash. This bootloader has the functionality to check the CRC by the RISC-V Firmware. Upon power-on, the bootloader configures the peripherals and the GSRD building blocks such as UART, GPIO, I^2C, SGDMA, TSE, and QSPI Flash Controller.
- The bootloader code loads the bitstream from flash to program the SRAM FPGA and fetches the Primary FW via the QSPI interface to the FPGA.
- As the FW needs to be executed from the external LPDDR4, the RISC-V module receives the incoming FW from QSPI and routes it to the external LPDDR4.
- The FW is stored at the beginning of LPDDR4 Memory. After the FW is copied, the RISC-V module then fetches the data back from LPDDR4 and calculates the CRC.
- This FW CRC is compared with the original CRC that is a part of flash.
  - If the data is consistent with the CRC, the RISC-V module executes the FW from LPDDR4.
  - If the CRC fails for either the Primary FPGA bitstream or Primary FW, the GSRD architecture triggers the switching to load the Golden FW and Bitstream from the SPI Flash.
- Upon the execution of the correct Primary or Golden FW from LPDDR4, the building blocks mentioned earlier are up and running with their associated drivers. For example, if any Ethernet data is expected to arrive, the RISC-V module sets the SGDMA IP with the respective parameters such as addresses, data-length, and others to successfully route the incoming Ethernet packets.

- When the Ethernet frame is received, the data is forwarded to the SGDMA block and the SGDMA transfers this data to the respective endpoints based on the parameters set. The endpoint in this case is the main memory LPDDR4 which is accessed by a Multi-Port Memory Bridge and LPDDR4 Memory Controller.
- You can also store the data in the system memory. Hence, to load the data, the data-path used for the incoming transaction is TSE_MAC, SGDMA, MPMC, and LPDDR4 MC.
- For outgoing data, you can transfer data from LPDDR4 to TSE MAC to the Host PC/device outside of the FPGA.
- When no data transfer occurs over Ethernet and SGDMA, the RISC-V CPU continues running its tasks in the usual manner based on the loaded FW execution.

Figure 1.2 shows the boot flow of the Primary and Golden images.

Note: For more details, refer to the GSRD Reference Design User Guide (FPGA-RD-02283).

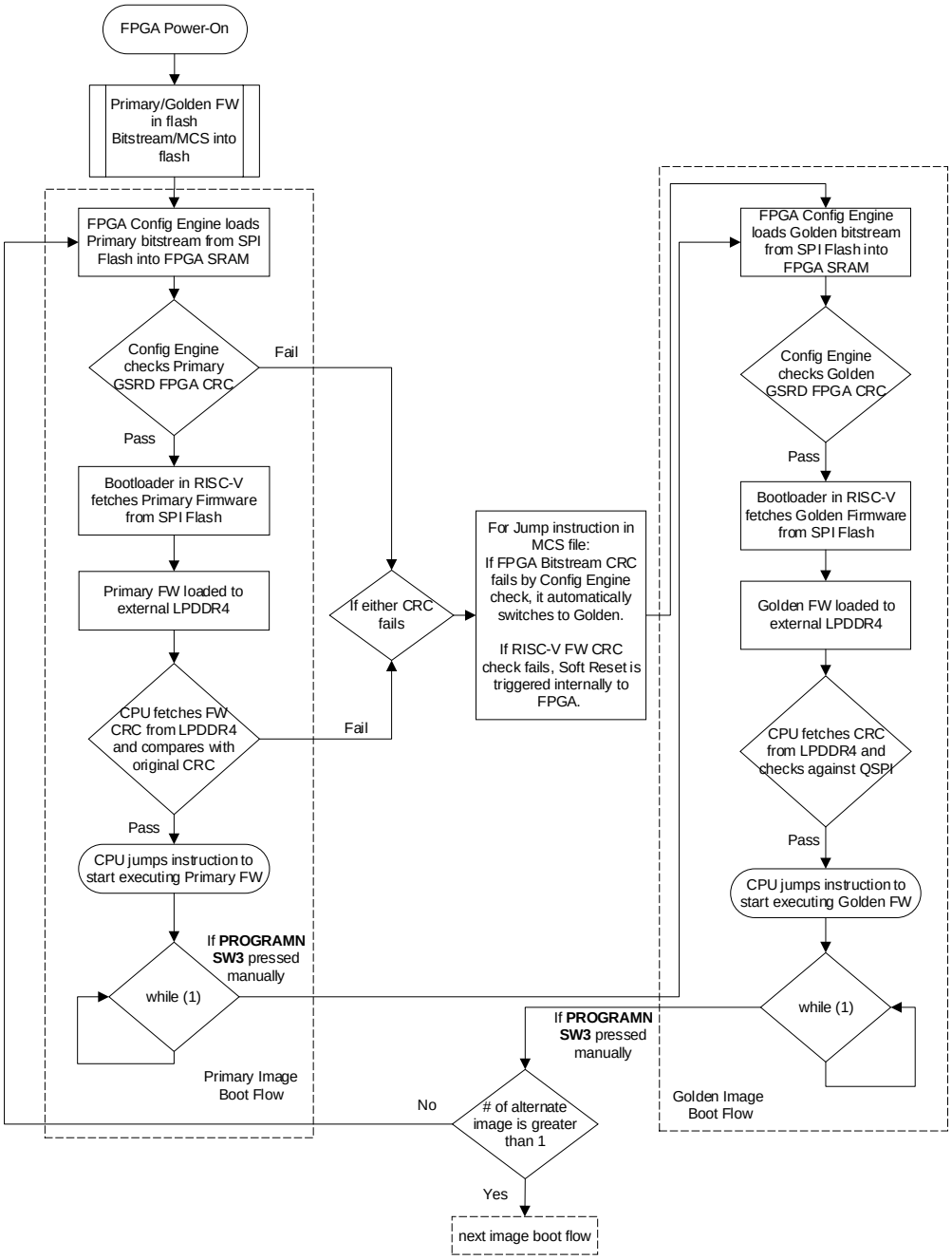

Figure 1.2. GSRD FPGA and Firmware Booting Diagram

2. Hardware and Software Requirements

2.1. Hardware Requirements

This demonstration requires the following hardware components:

- Lattice CertusPro-NX Versa Evaluation Board
- USB Type-A (UART) cable for programming the bitstream and propel terminal prints
- Electrical 1G SFP(s) Model FS SFP-GB-GE-T for Ethernet connection on the CertusPro-NX board (inserted at the J15 port of the CertusPro-NX Versa Board)
- Ethernet cable to connect the CertusPro-NX board to the Host PC
- 12 V power adapter for board power

2.2. Hardware Setup

Figure 2.1 shows the hardware connections.

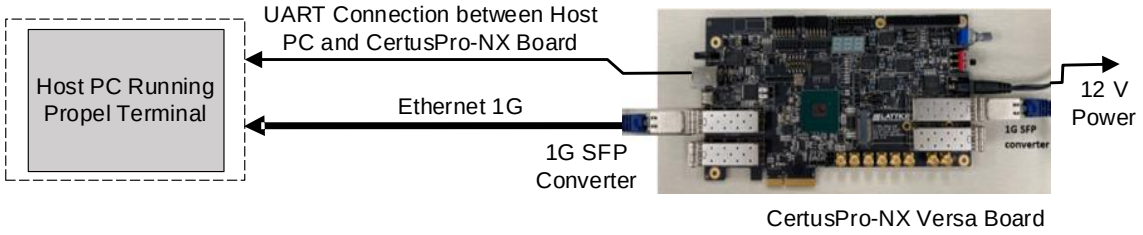

Figure 2.1. Hardware Setup for GSRD

2.3. Hardware System Readiness

To set up the system:

1. Connect the hardware as shown in Figure 2.1.
2. Program the Lattice CertusPro-NX Versa Evaluation Board as follows:
   a. For multi-boot testing – Program only the FW (*.bin*) and bitstream (*.mcs*) file.
   b. For standalone Primary/Golden testing – Program only the FW (*.bin*) and their respective bitstreams (*.bit*) files.
3. Power cycle the board after programming the FW binaries (*.bin*) file(s).
4. Power cycle the board after programming the *.mcs* file, Reset the board to check the UART output. To reset, press the **SW3** button on the CertusPro-NX Versa Evaluation Board.
5. Configure the Ethernet settings based on the steps described in the Running the GSRD/GHRD Demo section.
6. Power up the system based on the following sequence:
   a. Power on the Host PC and the CertusPro-NX Versa Board.
   b. Program the binary (*.bin*) file of the Primary and Golden system into the SPI Flash.
   c. Program the *.mcs* file into the SPI Flash.
   d. Run the Lattice Propel™ terminal.

   Note: You can run the Propel terminal before programming the standalone bitstreams (*.bit*) in SRAM fast configuration mode for output to show instantly on the Propel Terminal window without needing to apply a push-button reset (**SW3**).

Figure 2.2. Reset Button SW3 of CertusPro NX Versa Board

2.4. Software Requirements

- Lattice Radiant™ software version 2023.2 or later
- Lattice Radiant Programmer version 2023.2 or later
- Lattice Propel 2023.2 – Contains both the SDK and the Propel Builder

3. GSRD Demo Package

For this demonstration, there are a total of five programming files provided with their corresponding flash addressees to run a successful GSRD. These files are in the GSRD demo *GSRD_Executables.zip* folder.

Table 3.1. GSRD Demo Package

File Description	File Name	Starting Address in Flash
Primary FW Image	Primary_AppCrc.bin	0x020A 0000
Primary FPGA Image	soc_golden_system_impl_1.bit	0x0000 0000
Golden FW Image	Golden_AppCrc.bin	0x0200 0000
Golden FPGA Image	soc_golden_system_impl_1.bit	0x0000 0000
MCS File	GSRD_Primary_Golden.mcs	0x0000 0000

4. Running the GSRD/GHRD Demo

4.1. Opening the UART Terminal

Note: Open this terminal only after the FW (*.bin*), *.bit* and *.mcs* files are programmed. Otherwise, skip to the Programming the Standalone Primary/Golden GSRD Bitstream and Firmware section.

1. Connect the CertusPro-NX Versa board to the PC/laptop using a UART cable.
2. Connect the Ethernet cable between the CertusPro-NX Versa board with SFP and the Ethernet port on the PC/laptop as shown in the Hardware Setup section.
3. Open the Propel SDK 2023.2 tool.
4. Double-click the terminal button as shown in Figure 4.1.

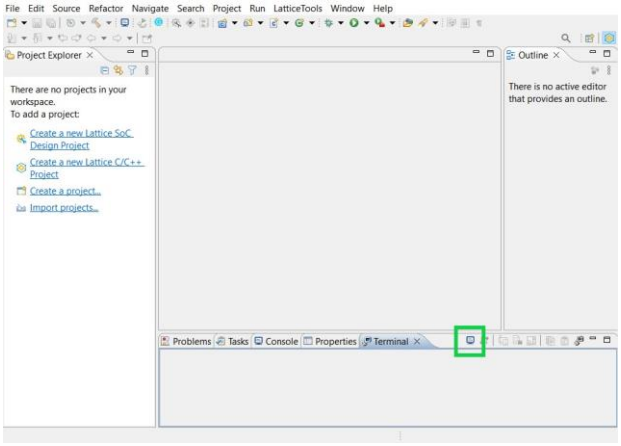

Figure 4.1. Open Terminal

The terminal opens as shown in Figure 4.2.

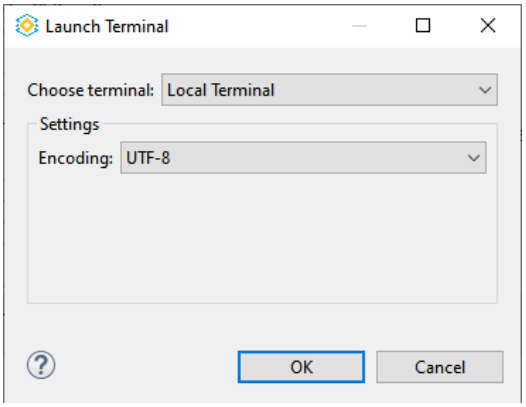

Figure 4.2. Propel Terminal Selection Window

5. **Choose terminal: Serial terminal** as shown in Figure 4.4.
6. In the **Serial port** dropdown list, select the last *COM* in the list as shown in Figure 4.3.
   Note: This detail can be found under the *Ports (COM & LPT)* section in your local PC, under *Device Manager*. Your *COM* can be different. If a USB port does not work, try a different USB port.

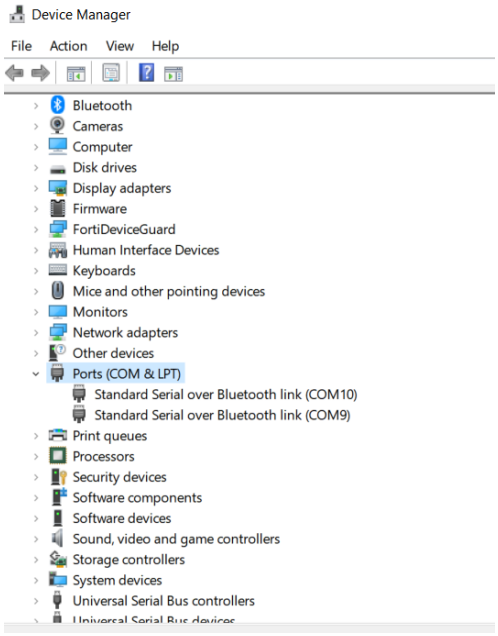

Figure 4.3. Propel Terminal Selection Window

7. Set **Baud rate:** 115200.
8. Click **OK**.

   Note: Apply these settings before programming the bitstream in SRAM - fast configuration settings to see the output on the Propel terminal.

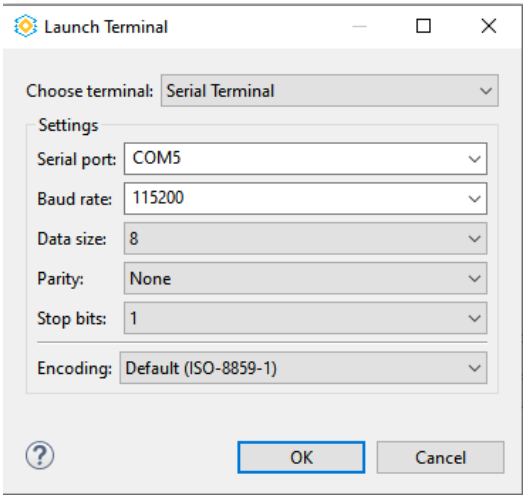

Figure 4.4. Propel Terminal: Com and Baud Rate Selection

9. If you are not receiving Ethernet packets, apply the setting as shown below:
   a. Press the **Windows + R** button simultaneously.
   b. Type the command: *ncpa.cpl* and press **OK**.

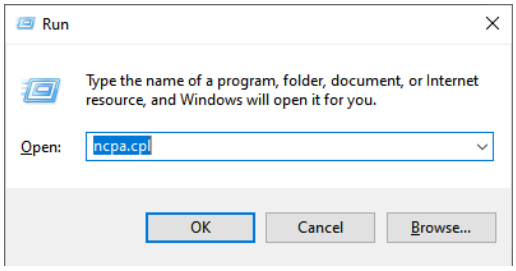

Figure 4.5. Ethernet Setting Command: ncpa.cpl

c. Right-click on **Ethernet** and click **Disable**.

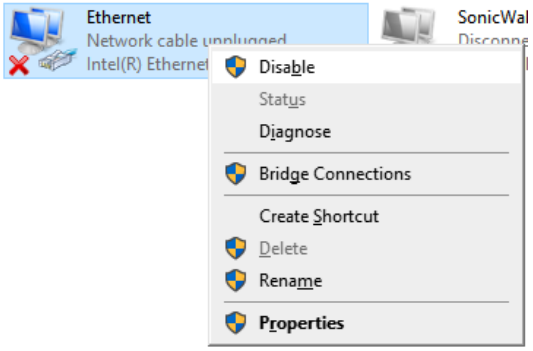

Figure 4.6. Ethernet Setting: Disable

d. Right-click again and click **Enable**.

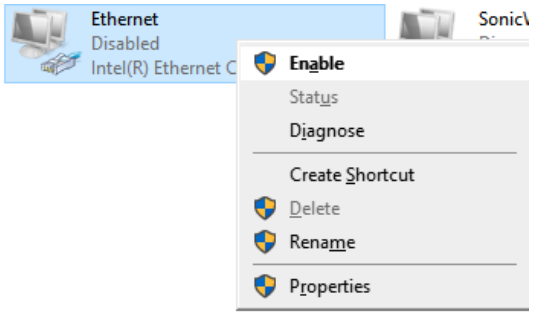

Figure 4.7. Ethernet Setting: Enable

4.2. Programming the Standalone Primary/Golden GSRD Bitstream and Firmware

1. Program the SPI Flash on the CertusPro-NX Versa board. Refer to Appendix A. Programming the GSRD Primary/Golden Bitstream, Firmware Binaries and MCS File for more details.
2. Program the standalone Primary GSRD into SPI Flash. Refer to A.1. Primary GSRD Programming Flow for more details.
   a. Erase all.
   b. Program the Primary Firmware CRC binary (*.bin*) file at the start address as described in the GSRD Demo Package section.
   c. Program the Primary FPGA bitstream (*.bit*) file as described in the GSRD Demo Package section.
   d. Power cycle the CertusPro-NX Versa board.
   e. Press the reset button (**SW3**) on CertusPro-NX Versa board.
   f. Check the LED Status on the board:

- **D63**: Linkup LED
- **D64**: LPDDR4 initialization LED
- **D67**: PLL lock LED
- **D65**: Running animation LED
- **D66**: Running animation LED
- **D104**: Running animation LED
- **D105**: Running animation LED

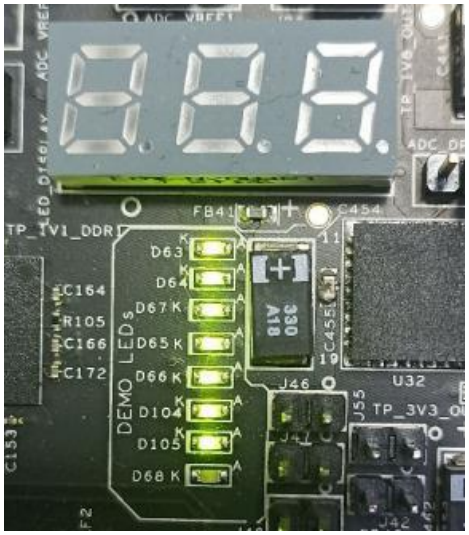

Figure 4.8. LED Status of Primary Image on the CertusPro-NX Versa Board

3. Open and run the Propel terminal to check the UART output. Refer to the Opening the UART Terminal section for more details. The output status is as follows:
   - DDR Status:
     30021 – DDR initial interrupt value status
     0 – Polling DDR INIT Done signal
     30027 – DDR in self-refresh mode
     3001f – Polling DDR status; DDR ready to be initialized
   - LPDDR4 CRC:
     3e26 – CRC calculated for FW data copied from the flash to LPDDR4
   - FW CRC:
     3e26 – Original copied FW CRC
   - SGDMA packet received:
     Shows the first Ethernet packet
   - S2MM Status:
     Shows the S2MM status 80000040 (SGDMA transaction completed with packet length 0x40)
   - List of tasks running:
     Indicates that the firmware is running and executing correctly

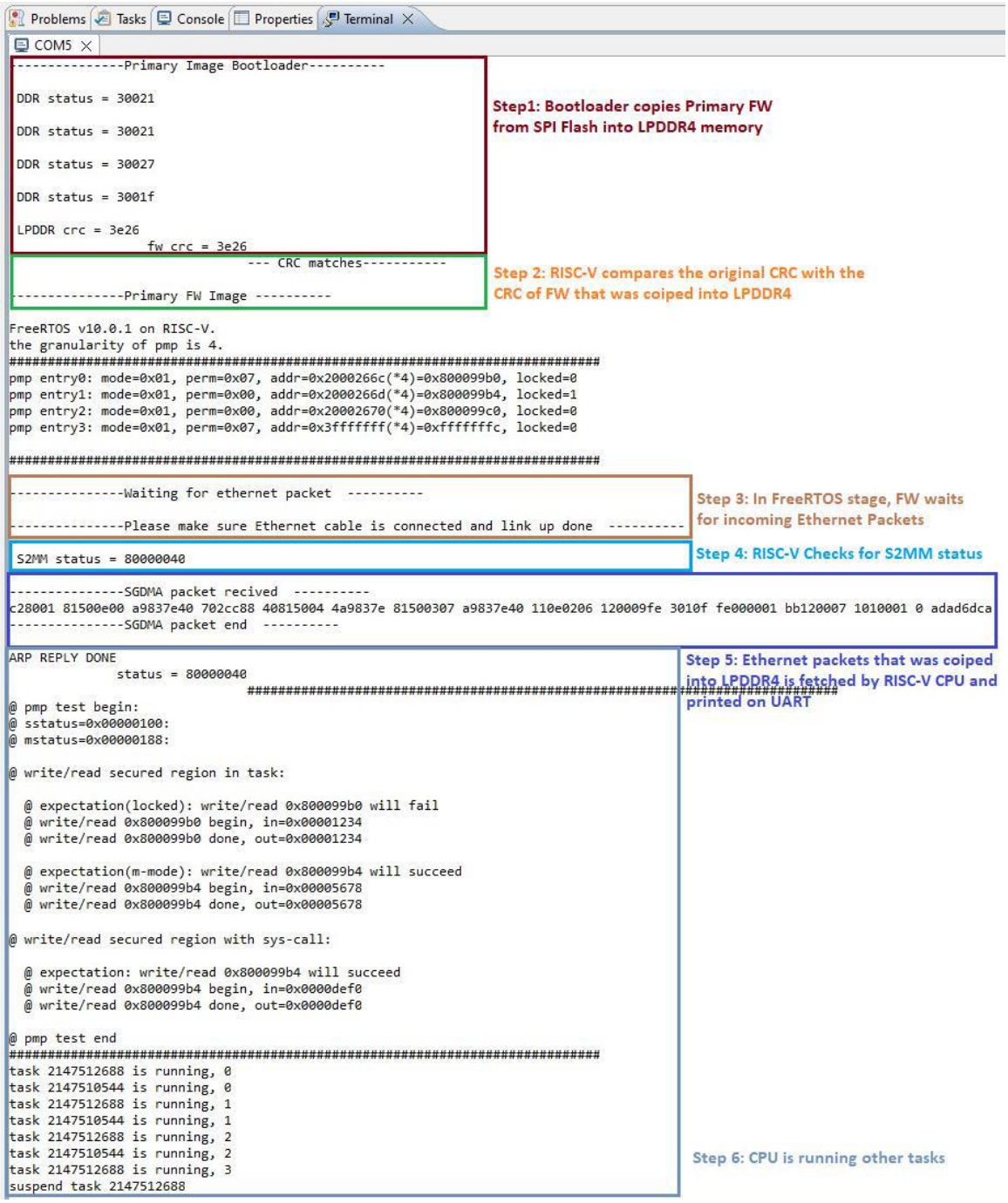

Figure 4.9. Primary Image Display Output on the Propel Terminal

4. To program the Standalone Golden GSRD into the SPI Flash, refer to A.2. Golden GSRD Programming Flow.
   a. Erase all.
   b. Program the Golden Firmware CRC binary (*.bin*) file at the start address as described in the GSRD Demo Package section.
   c. Program the Golden FPGA bitstream (*.bit*) file as described in the GSRD Demo Package section.
   d. Power cycle the CertusPro-NX Versa board.
   e. Press the reset button (**SW3**) on CertusPro-NX Versa board.
   f. Check the LED Status on the board:

- **D63**: Linkup LED
- **D64**: LPDDR4 initialization LED
- **D67**: PLL lock LED
- **D65**: Running animation LED
- **D66**: Running animation LED
- **D104**: Running animation LED

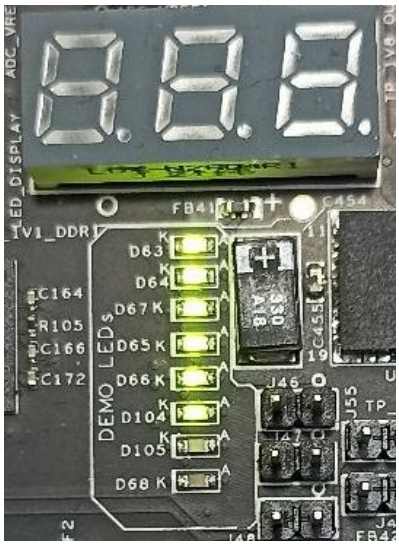

Figure 4.10. LED Status of Golden Image on the CertusPro-NX Versa Board

5. Open and run the Propel terminal to check the UART output. Refer to the Opening the UART Terminal section for more details. The output status is as follows:
   - DDR Status:
     30021 – DDR initial interrupt value status
     0 – Polling DDR INIT Done signal
     30027 – DDR in self-refresh mode
     3001f – Polling DDR status; DDR ready to be initialized
   - LPDDR4 CRC:
     bd15 – CRC calculated for FW data copied from the flash to LPDDR4
   - FW CRC:
     bd15 – Original copied FW CRC
   - SGDMA packet received:
     Shows the first Ethernet packet
   - S2MM Status:
     Shows the S2MM status 80000040 (SGDMA transaction completed with packet length 0x40)
   - List of tasks running:
     Indicates that the firmware is running and executing correctly

Figure 4.11. Golden Image Display Output on the Propel Terminal

4.3. Programming the GSRD Primary Firmware, Golden Firmware and MCS File

4.3.1. Manual Booting the Primary CRC Binary and Golden CRC Binary Firmware Files

1. To program the flash on the CertusPro-NX versa board, refer to A.1. Primary GSRD Programming Flow.
2. To program the FW (*.bin*) and Bitstream (*.bit/.mcs*) files in the flash:
   a. Erase All – 0x00000000 to 0x003FF0000
   b. Program the Primary CRC (*.bin*) firmware file
   c. Program the Golden CRC (*.bin*) firmware file
   d. Program the multi-boot nitstream (*.mcs*) file

      Note: Refer to A.1. Primary GSRD Programming Flow for more details.
   e. After programming, open the Propel terminal to review the UART output. Refer to Opening the UART Terminal for more details.
   f. Display the output of Primary and Golden images.
   g. Press the **SW3** reset button on the CertusPro-NX Versa board to refresh the output.
   h. Press the **SW2 PROGRAMN** button on the CertusPro-NX Versa board to switch from the Primary image output to the Golden image output.

Figure 4.12. Primary Image Display Output on the Propel Terminal

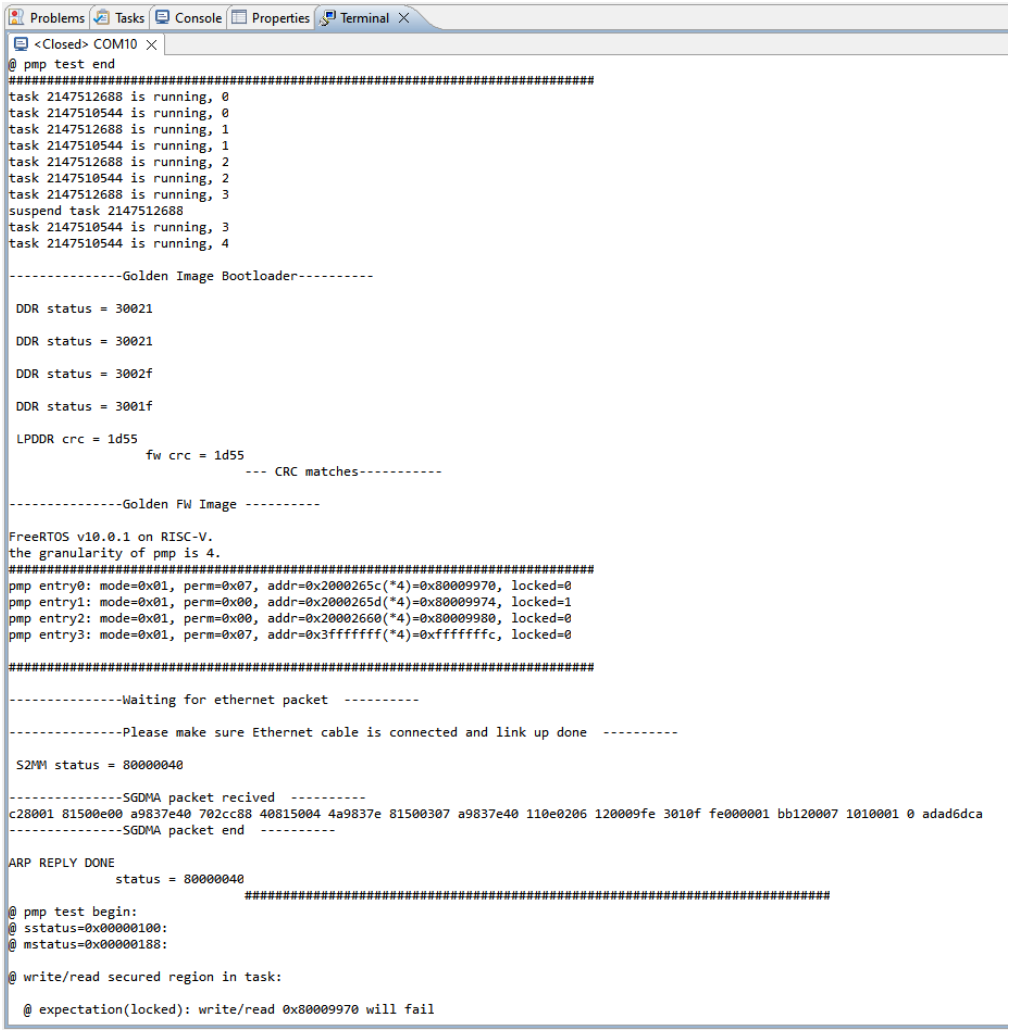

Figure 4.13. Display Output on the Propel Terminal of the Primary Image Switching to Golden Image

4.3.2. Manual Booting of the Primary CRC Binary and Golden (without CRC) Binary Firmware Files

1. To program the flash on the CPNX board, refer the GSRD user Demo guide section Appendix A.
2. To program the FW (*.bin*) and Bitstream (*.bit/.mcs*) files in flash:
   a. Erase All – 0x00000000 to 0x003FF0000
   b. Program the Primary CRC (*.bin*) firmware file
   c. Program the Golden without CRC (*.bin*) firmware file
   d. Program the Multi-boot Bitstream (*.mcs*) file

      Note: Refer to A.1. Primary GSRD Programming Flow for more details.
   e. After programming, run the Propel terminal to check the UART output. Refer to Opening the UART Terminal for more details.
   f. Review the the output of the Primary image and Golden image.
   g. Press the **SW3** reset button on the CertusPro-NX Versa board to refresh the output.
   h. Press the **SW2 PROGRAMN** button on the CertusPro-NX Versa board to switch from the Primary image output to the Golden image output.

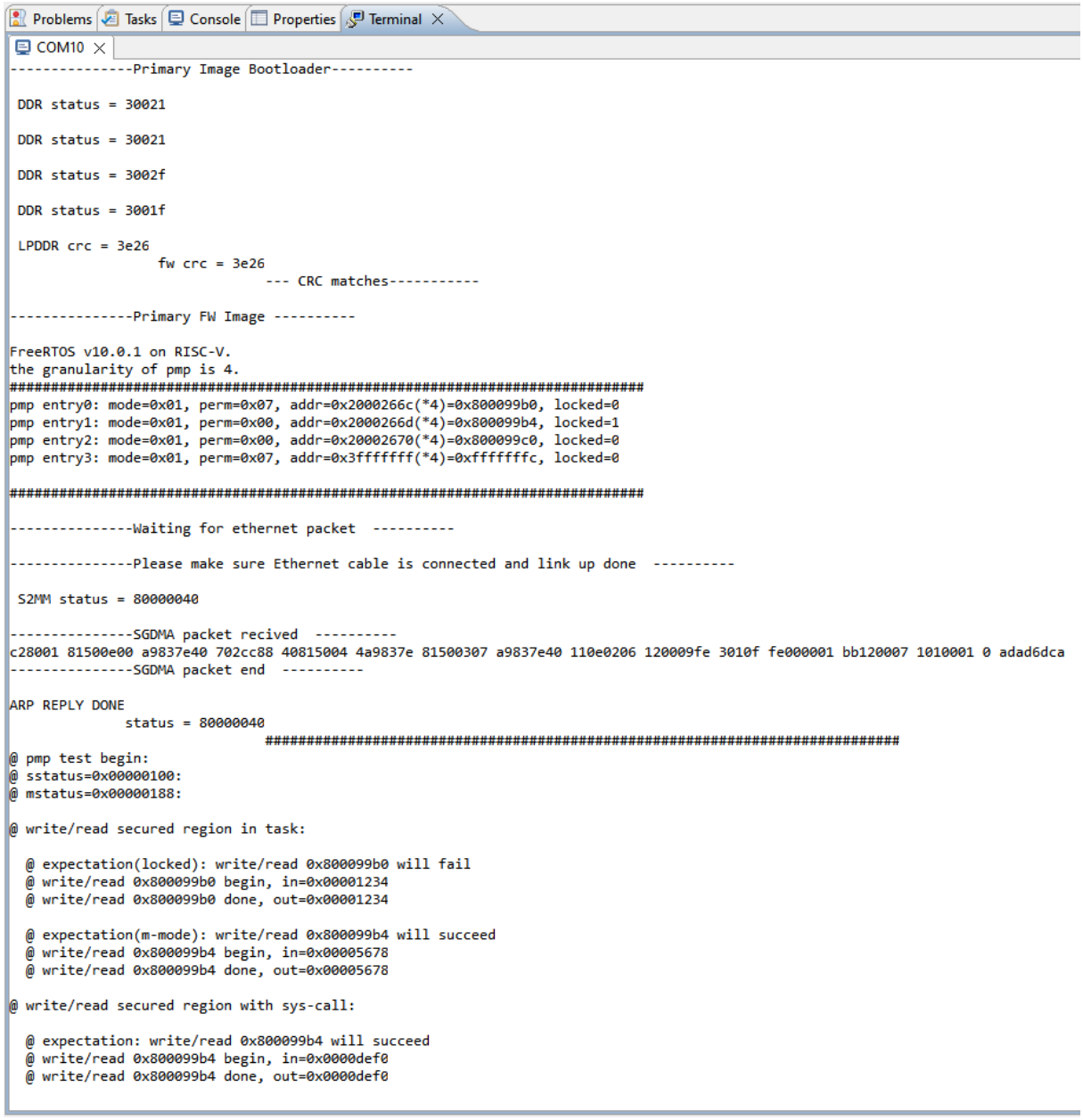

Figure 4.14. Primary Image Display Output on the Propel Terminal

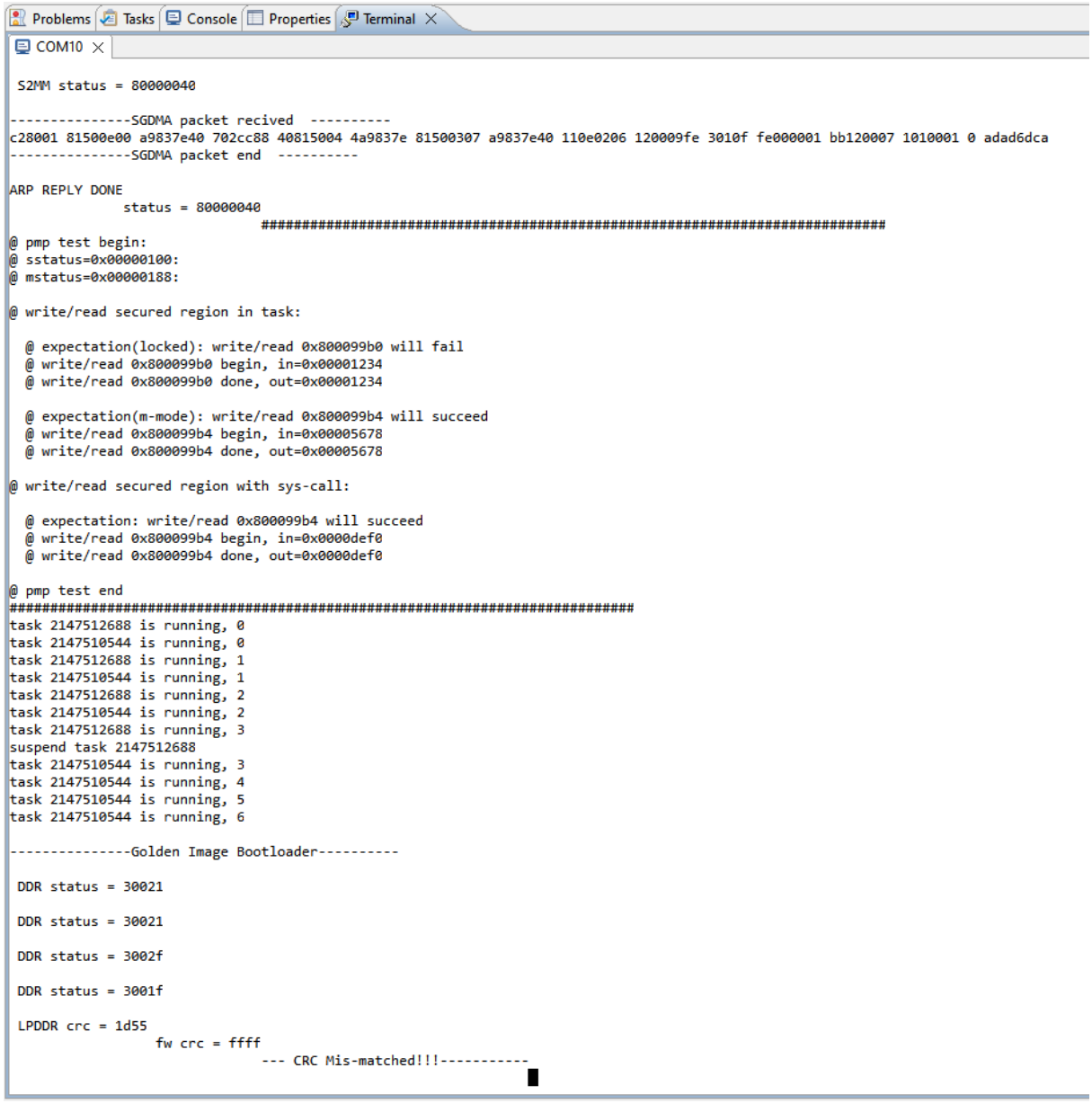

Figure 4.15. Primary and Golden Mismatch Images Display Output on the Propel Terminal

4.3.3. Automatic Booting of the Primary (without CRC) Binary and Golden CRC Binary Firmware Files

1. Program the flash on the CertusPro-NX Versa board. Refer to Appendix A. Programming the GSRD Primary/Golden Bitstream, Firmware Binaries and MCS File for more details.
2. Program the FW (*.bin*) and bitstream (*.bit/.mcs*) files in the flash.
   a. Erase All – 0x00000000 to 0x003FF0000
   b. Program the Primary image without the CRC (*.bin*) firmware file
   c. Program the Golden image with the CRC (*.bin*) firmware file
   d. Program the multi-boot nitstream (*.mcs*) file
3. After programming, run the propel terminal to check the UART output. Refer to the Opening the UART Terminal section for more details.

   Note: The terminal displays the output of the Primary and Golden images. First, the Golden bootloader boots as it fails the Primary CRC check during power-on and switces to the Golden bootloader before the UART terminal launches.

4. Press the **SW3** reset button on the CertusPro-NX Versa board to refrest the output.
5. Press the **SW2 PROGRAMN** button on the CertusPro-NX Versa board to switch from the Primary to Golden Image output.

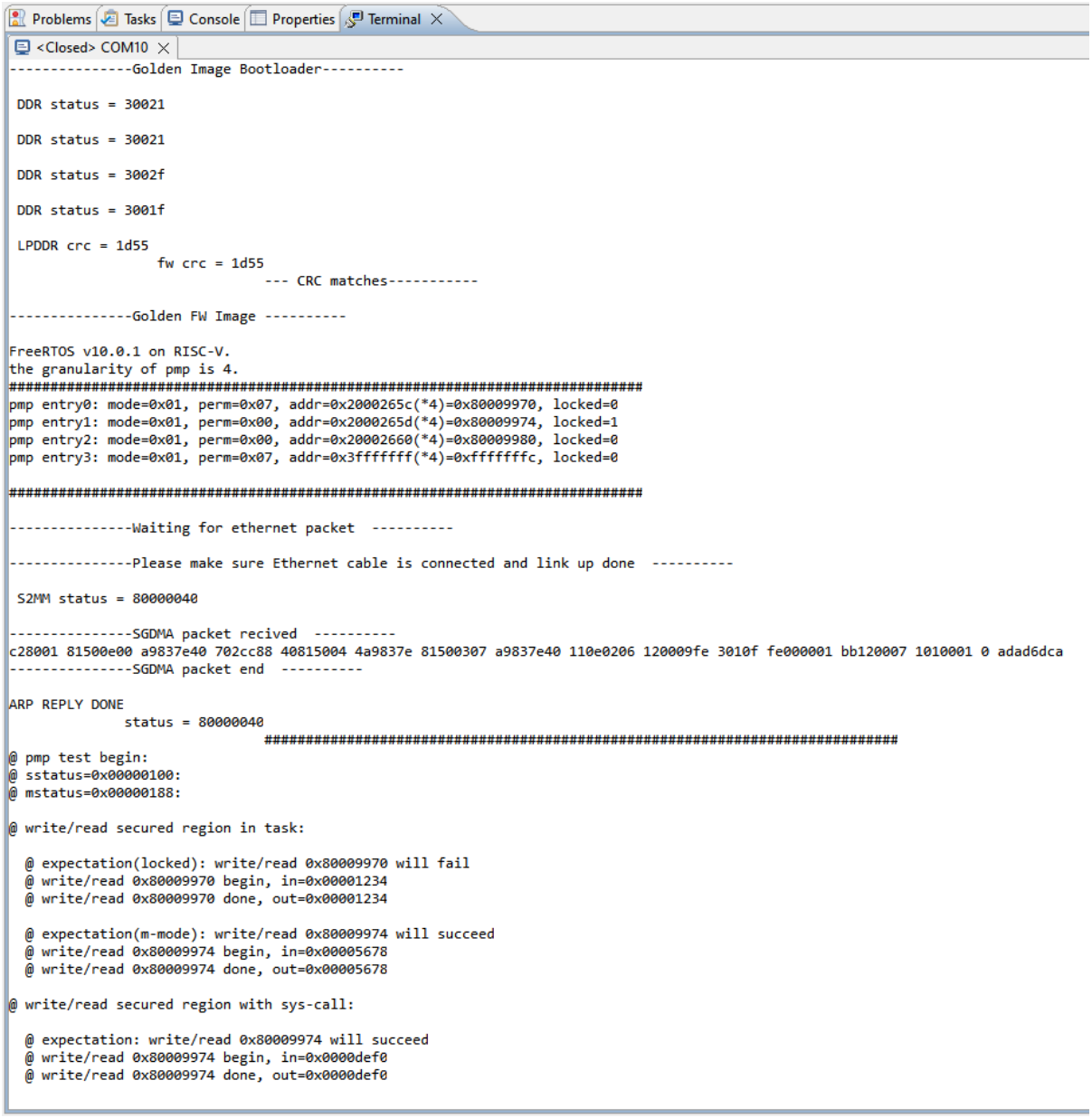

Figure 4.16. Golden Image Display Output on the Propel Terminal

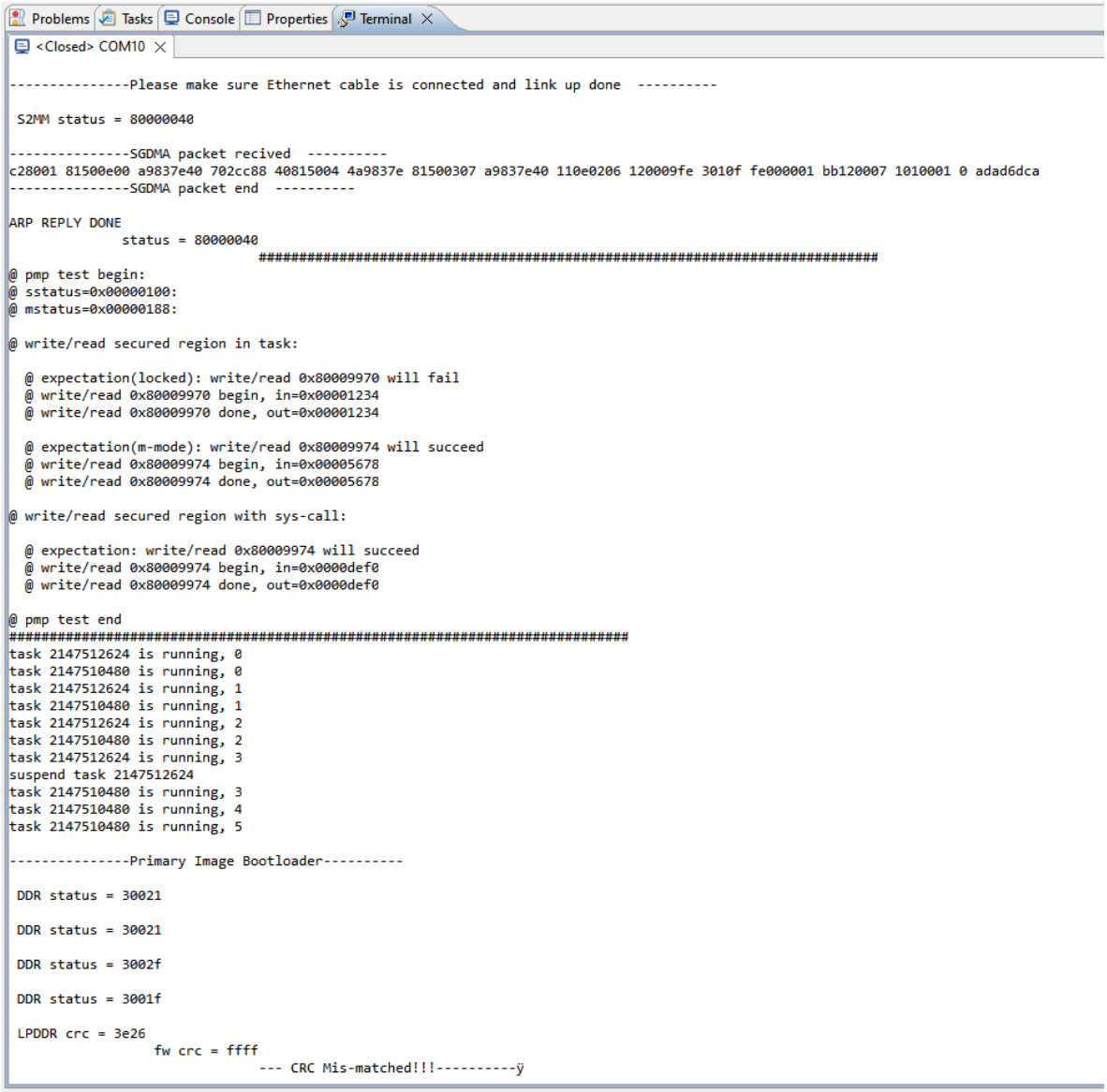

Figure 4.17. Primary and Golden Mismatch Images Display Output with the Multi-Boot Feature Enabled

Appendix A. Programming the GSRD Primary/Golden Bitstream, Firmware Binaries and MCS File

Ensure that following jumpers are connected:

- Pin 1 and 2 of J32 and J33 are shorted to select UART
- Pin 1 and 2 of J58 are shorted to select 3.3 V as Flash I/O

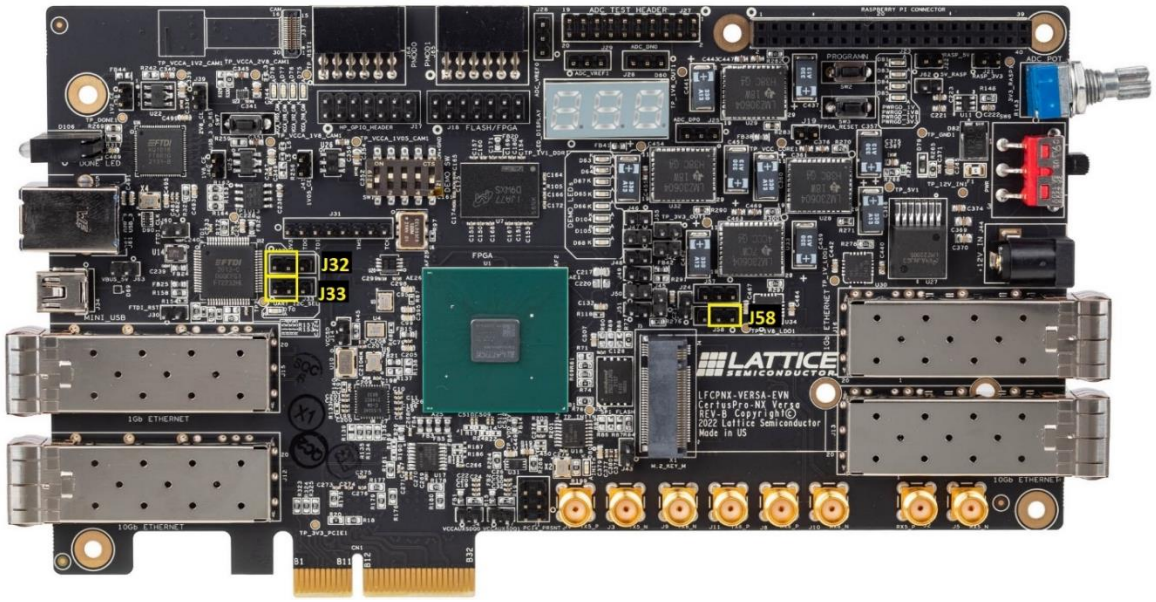

Figure A.1. CertusPro-NX Board Jumper Connection

Note: To generate the bitstream and binary file, refer to the GSRD Reference Design User Guide (FPGA-RD-02283).

To program the SPI Flash in the Radiant Programmer:

1. Connect the CertusPro-NX Versa board to the PC/laptop using a USB cable.
2. Power-on the CertusPro-NX Versa board.
3. Start Radiant Programmer. In the Getting Started dialog box, select **Create a new blank project.** Browse to the **Project Location** in your local machine.

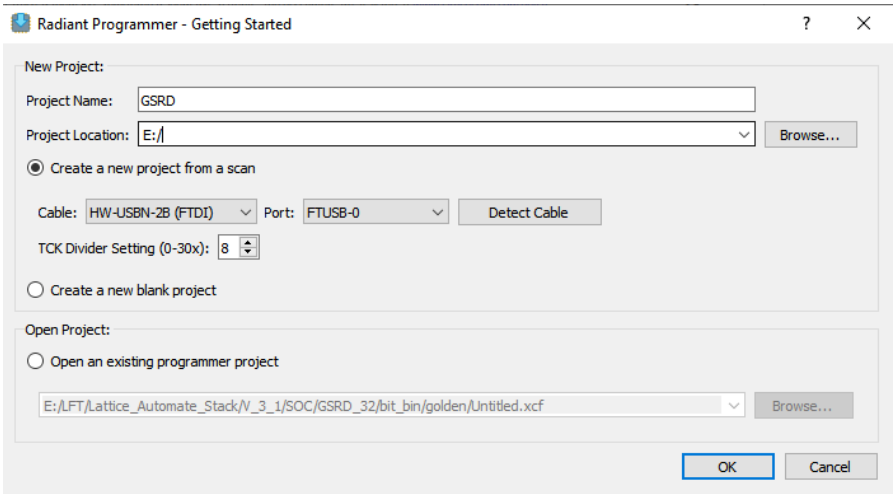

Figure A.2. Radiant Programmer – Default Screen

4. Click **OK**.

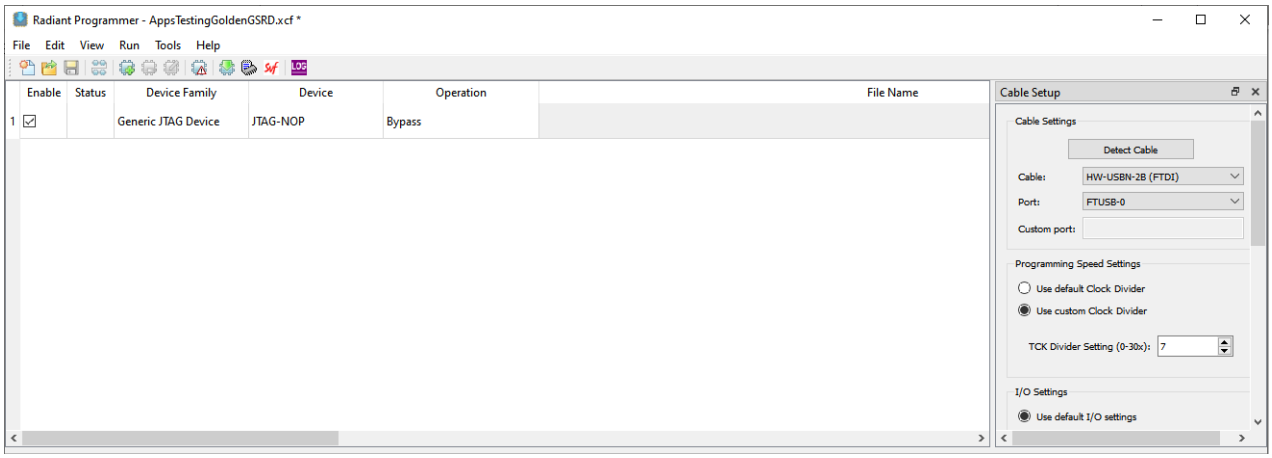

Figure A.3. Default Radiant Programmer Main Interface

5. Click on the **Scan Device** icon as shown below.

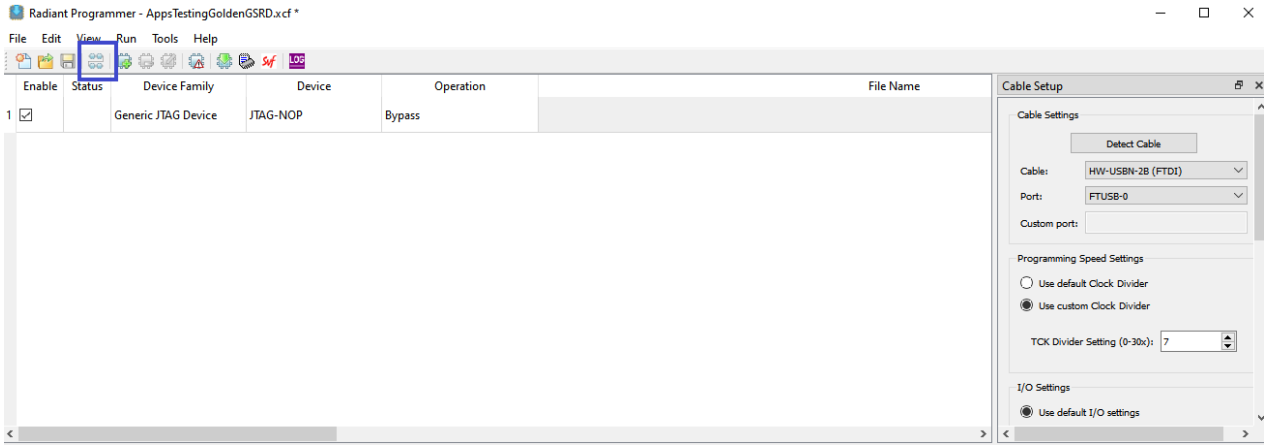

Figure A.4. Scan Device

6. Click on **Use Current Value**.

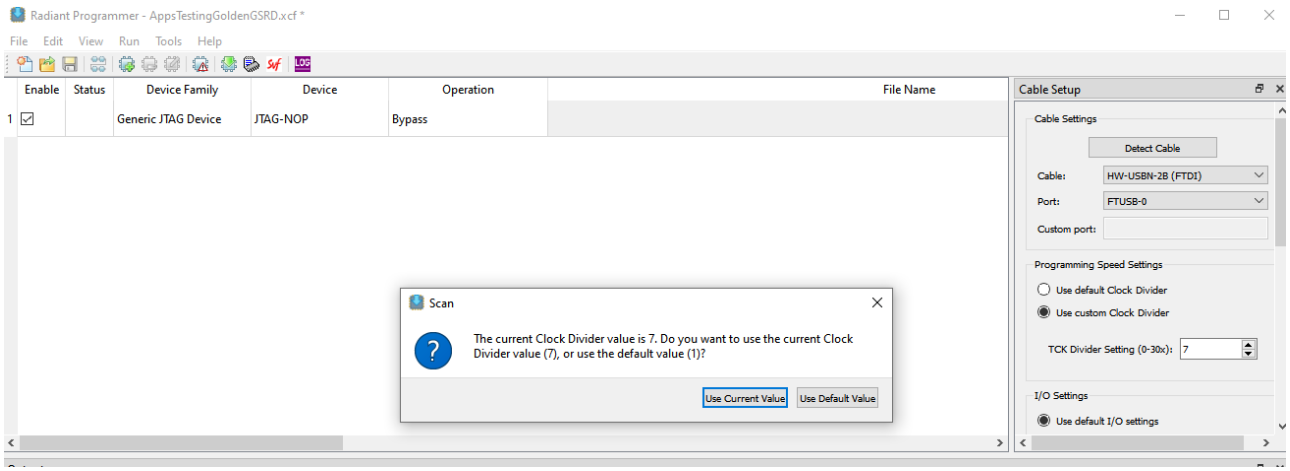

Figure A.5. Use Current Value

7. Click on **Yes.**

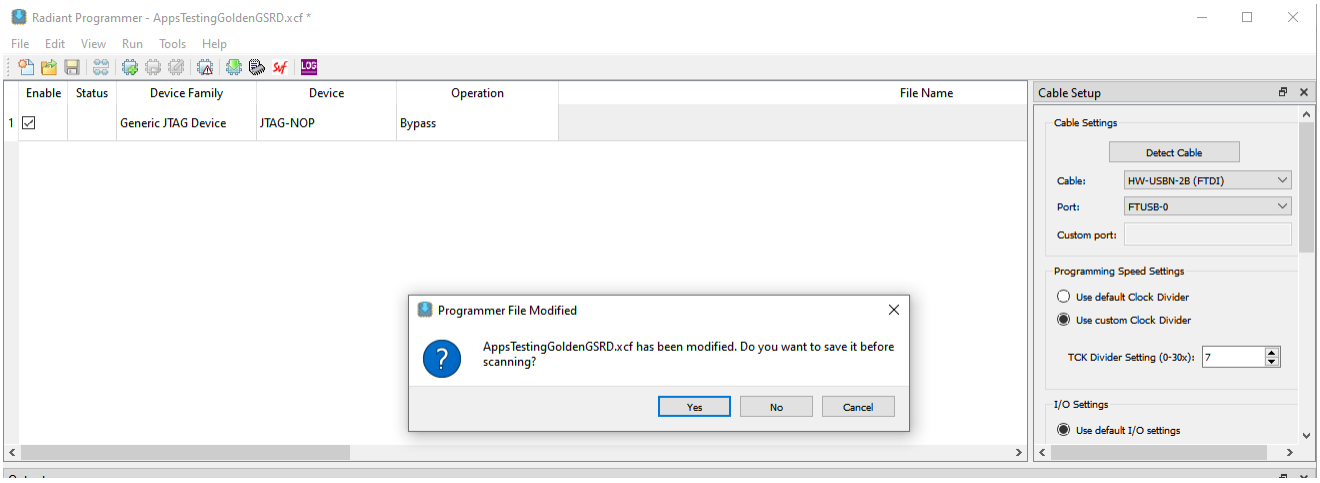

Figure A.6. Click Yes

8. Ensure the correct **Device Family** (LFCPNX) and **Device** (LFCPNX-100) are selected as shown below. If not, select the **Device** manually.

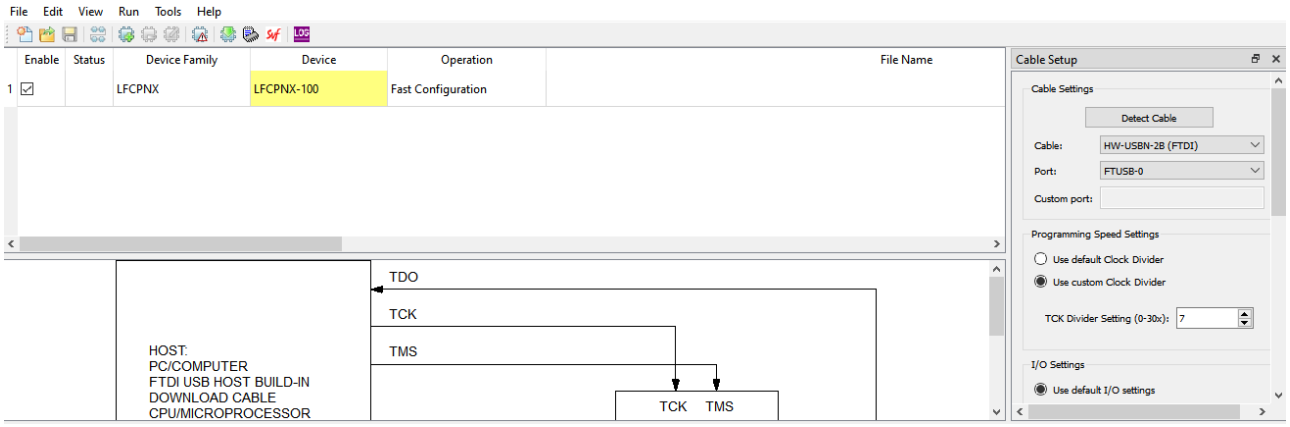

Figure A.7. Select Correct Device Family and Device – LFCPNX and LFCPNX-100

9. Double-click on the **Operation** tab or right-click and select **Device Properties**.

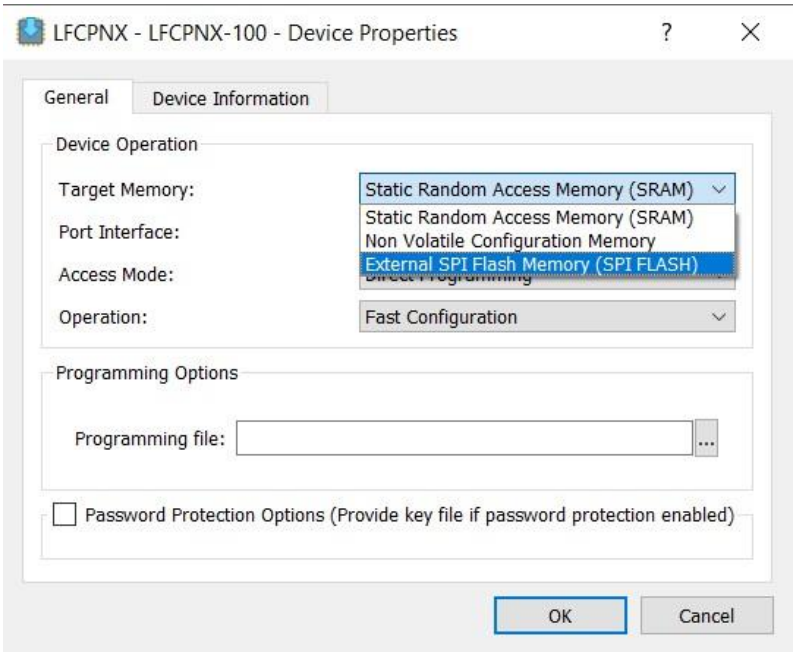

Figure A.8. Radiant Programmer – Device Operation

10. Before programming, you need to erase the entire SPI flash memory by applying the settings below.
    - Under **Device Operation**, select the following options:
        - **Target Memory**: External SPI Flash Memory (SPI FLASH)
        - **Port Interface** : JTAG2SPI
        - **Access Mode**: Direct Programming
        - **Operation**: Erase All
        - **Start addresses (Hex):** 0x00000000
        - **End address (Hex):** 0x03FF0000
    - Under **SPI Flash Options**, select the following options:
        - **Family**: SPI Serial Flash
        - **Vendor**: Macronix
        - **Device**: MX25L51245G
        - **Package**: 8-land WSON

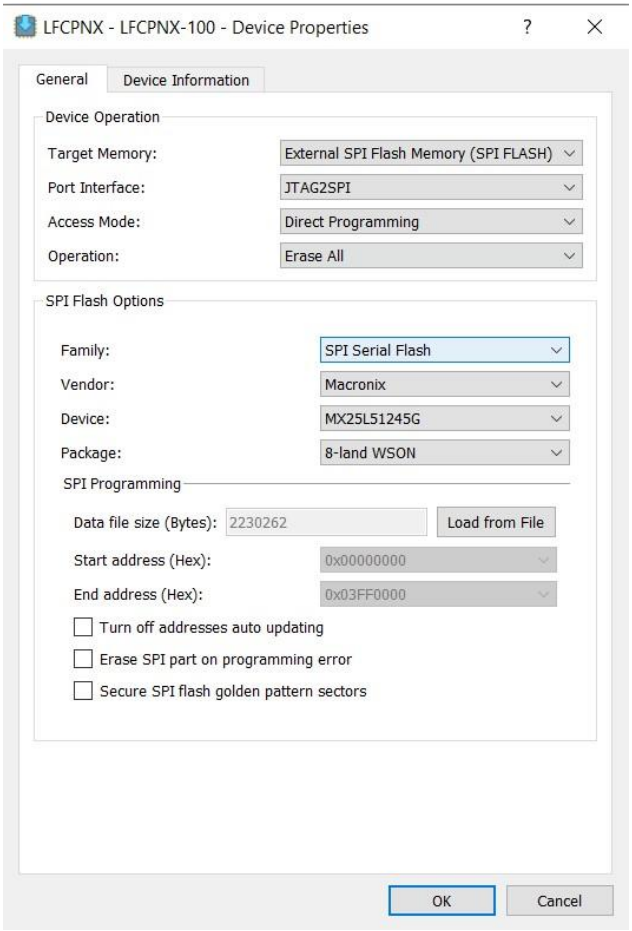

Figure A.9. Radiant Programmer – Erase All

11. Click **OK** and click the **Program Device** icon or the menu item, **Run >Program Device**. This erases the flash memory.
12. Power cycle the CertusPro-NX Versa Board.
13. Erase the FGPA SRAM.

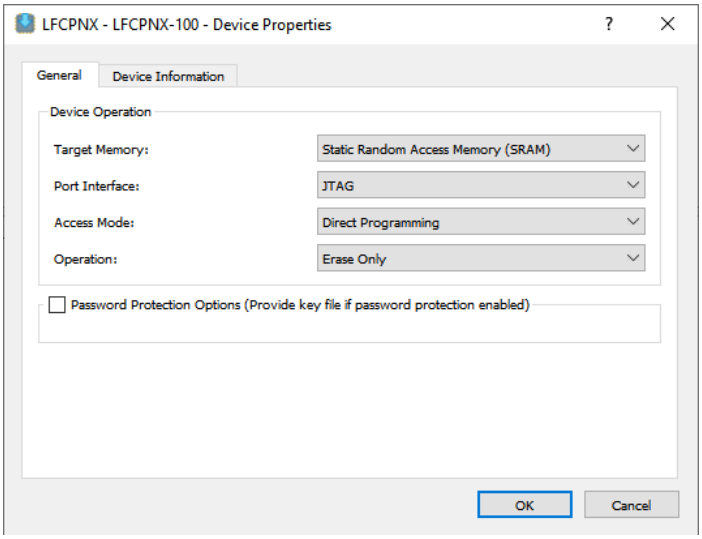

Figure A.10. Radiant Programmer – Erase Only

14. Skip steps 16 to 20 if you have performed the steps before. Otherwise, perform a one time step required by JTAG to configure NV Register 1 to modify the default addressing mode from 24 bits to 32 bits.

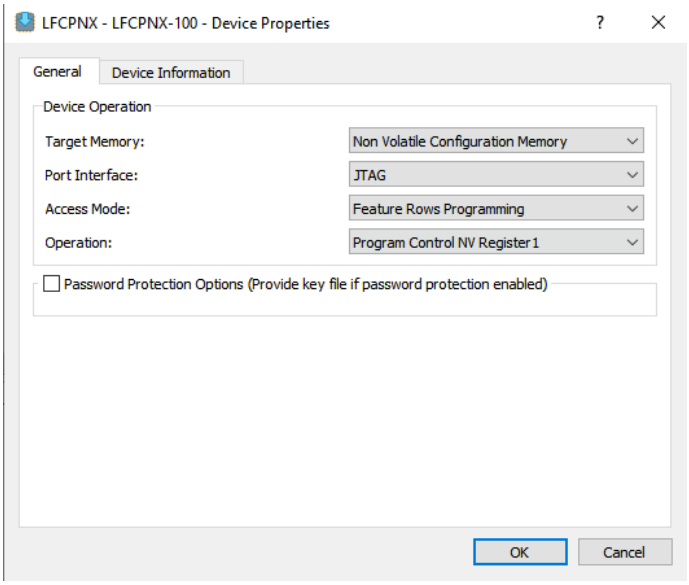

Figure A.11. Radiant Programmer – Configure NV Register 1 Setting

15. Click **OK** and click the **Program Device** icon or go to the menu item, **Run >Program Device**.
16. Change bit 0 to 1 as highlighted below.

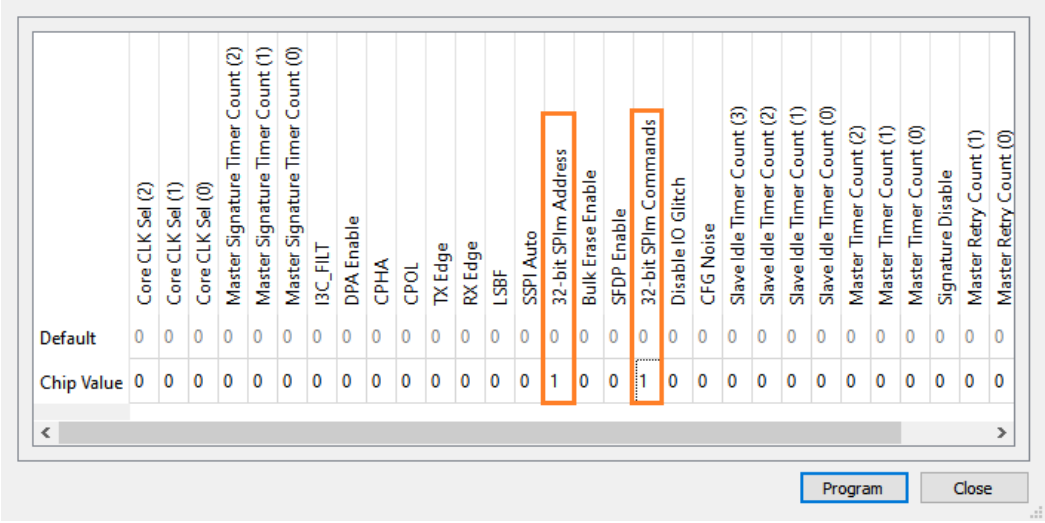

Figure A.12. Radiant Programmer – Configure NV Register 1 Setting (Update Bit Value)

Note: Update the value of the two highlighted fields. NV Register 1 is an OTP (One-Time-Programmable) Register.

17. After changing the bit, click **Program**.
18. Power cycle the CertusPro-NX Versa Board.

A.1. Primary GSRD Programming Flow

A.1.1. Programming the Primary_App Firmware File (.bin)

1. To program the firmware, select the options as shown below.
   - Under **Programming Options**, select the *Primary_AppCrc.bin* binary file.
   - Under **Device Operation**, select the following options:
     - **Target Memory**: External SPI Flash Memory (SPI FLASH)
     - **Port Interface** : JTAG2SPI
     - **Access Mode**: Direct Programming
     - **Operation**: Erase, Program, Verify
   - Under **SPI Flash Options**, select the following options:
     - **Family**: SPI Serial Flash
     - **Vendor**: Macronix
     - **Device**: MX25L51245G
     - **Package**: 8-land WSON
   - Ensure that the following addresses are correct:
     - **Start Address (Hex): 0x020A0000**
     - **End Address (Hex): 0x020D0000**

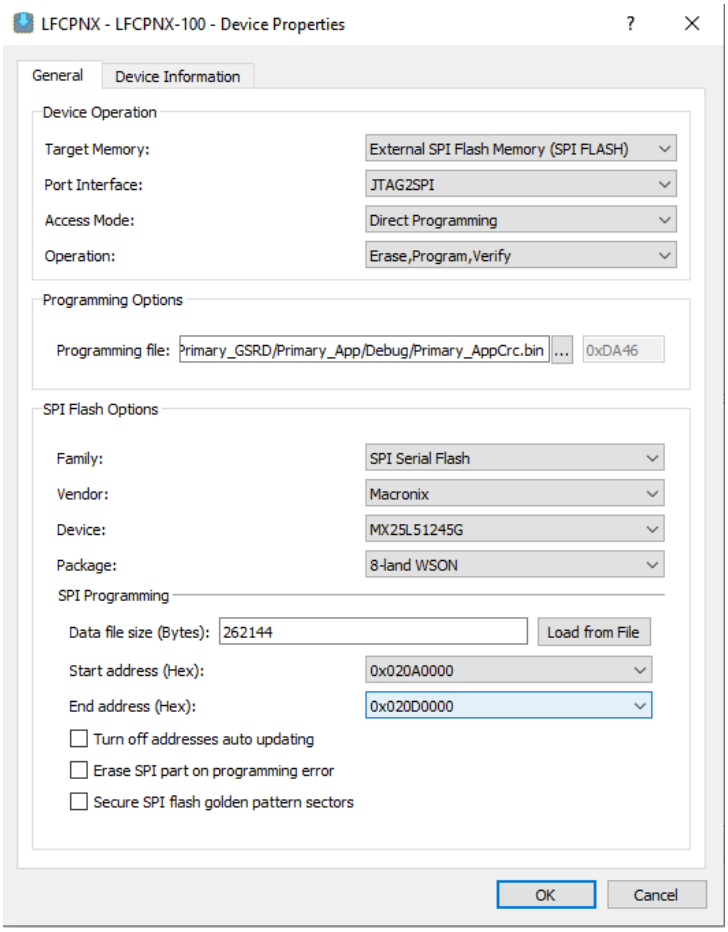

Figure A.13. Radiant Programmer - Binary Flashing Settings

2. Click the **Program Device** icon  or the go to menu item, **Run >Program Device**.
3. Power cycle the CertusPro NX Versa Board.

A.1.2. Programming the Primary GSRD Bitstream File (.bit)

Note: The steps below are only valid for standalone projects. You do not need to program the bitstream while programming the MCS file.

After programming the Flash, power cycle the board and apply the settings below.

1. To program the bitstream, select the options as shown below.
   - Under **Programming Options**, select the *soc_golden_system_impl_1.bit* bitstream file.
   - Under **Device Operation**, select the following options:
     - **Target Memory**: External SPI Flash Memory (SPI FLASH)
     - **Port Interface** : JTAG2SPI
     - **Access Mode**: Direct Programming
     - **Operation**: Erase, Program, Verify
   - Under **SPI Flash Options**, select the following options:
     - **Family**: SPI Serial Flash
     - **Vendor**: Macronix
     - **Device**: MX25L51245G
     - **Package**: 8-land WSON
   - Click Load from File to update the Data file size (Bytes) value.
   - Ensure that the following addresses, which are selected automatically, are correct:
     - **Start Address (Hex):** 0x00000000
     - **End Address (Hex):** 0x00240000

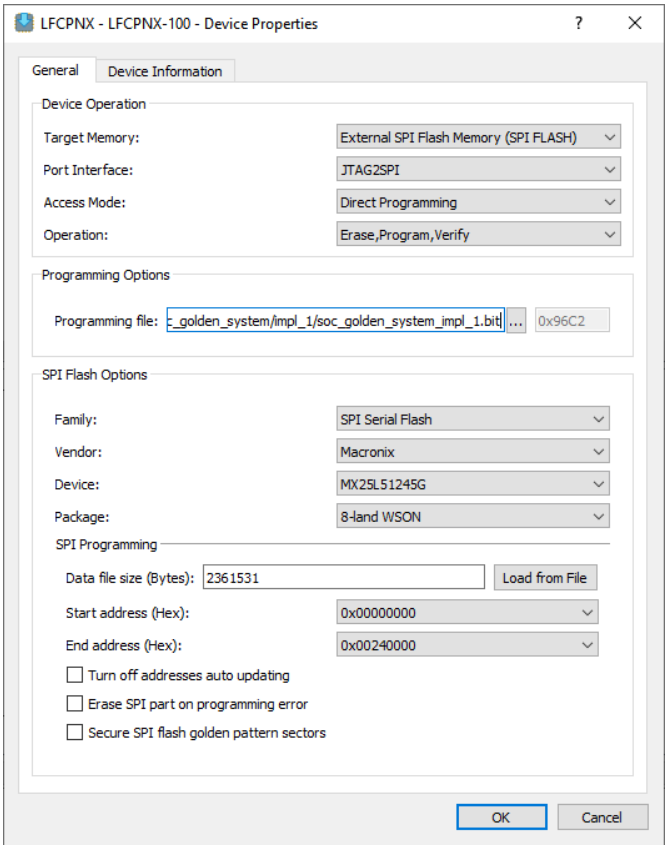

Figure A.14. Radiant Programmer - Binary Flashing Settings

2. Click the **Program Device** icon or go to the menu item, **Run >Program Device**.
3. Power cycle the CertusPro-NX Versa Board.

A.2. Golden GSRD Programming Flow

A.2.1. Programming the Golden_App Firmware File (.bin)

1. To program the firmware, select the options as shown below.
   - Under **Programming Options**, select the *Golden_AppCrc.bin* binary file.
   - Under **Device Operation**, select the following options:
     - **Target Memory**: External SPI Flash Memory (SPI FLASH)
     - **Port Interface** : JTAG2SPI
     - **Access Mode**: Direct Programming
     - **Operation**: Erase, Program, Verify
   - Under **SPI Flash Options**, select the following options:
     - **Family**: SPI Serial Flash
     - **Vendor**: Macronix
     - **Device**: MX25L51245G
     - **Package**: 8-land WSON
   - Ensure that the following addresses are correct:
     - **Start Address (Hex):** 0x02000000
     - **End Address (Hex):** 0x02030000

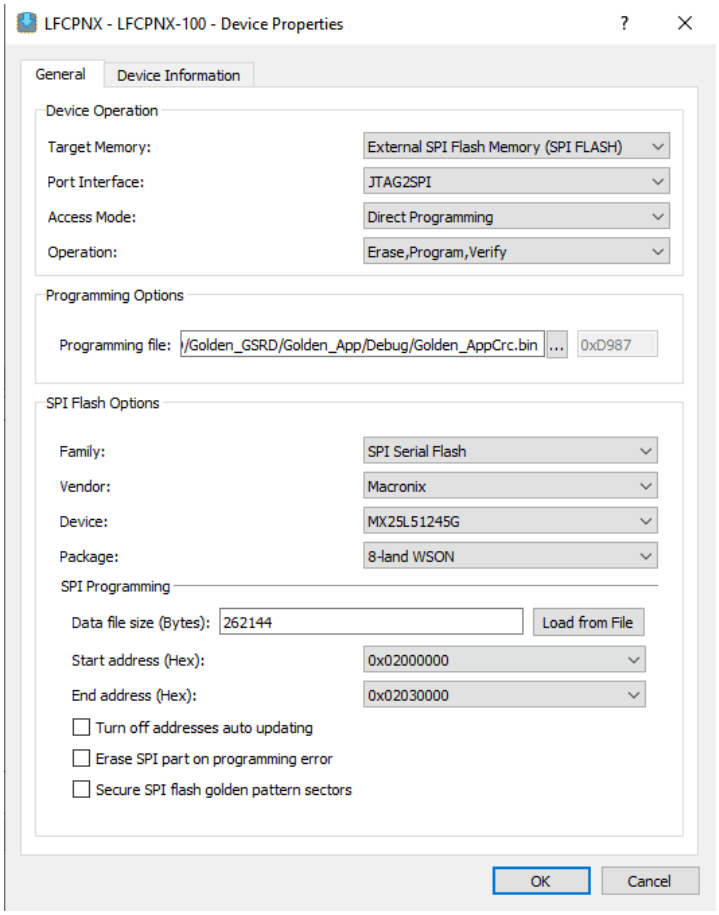

Figure A.15. Radiant Programmer - Binary Flashing Settings

2. Click the Program Device icon  or go to the menu item, **Run > Program Device**.
3. Power cycle the CertusPro NX Versa Board.

A.2.2. Programming Golden GSRD Bitstream File (.bit)

Note: The steps below are only valid for standalone projects. You do not need to program the bitstream while programming the MCS file.

After programming the Flash, power cycle the board and apply the settings below.

1. To program the bitstream, select the options as shown below.
   - Under **Programming Options**, select the *soc_golden_system_impl_1.bit* bitstream file.
   - Under **Device Operation**, select the following options:
     - **Target Memory**: External SPI Flash Memory (SPI FLASH)
     - **Port Interface** : JTAG2SPI
     - **Access Mode**: Direct Programming
     - **Operation**: Erase, Program, Verify
   - Under **SPI Flash Options**, select the following options:
     - **Family**: SPI Serial Flash
     - **Vendor**: Macronix
     - **Device**: MX25L51245G
     - **Package**: 8-land WSON
   - Click Load from File to update the Data file size (Bytes) value.
   - Ensure that the following addresses, which are selected automatically, are correct:
     - **Start Address (Hex)**: 0x00000000
     - **End Address (Hex)**: 0x00240000

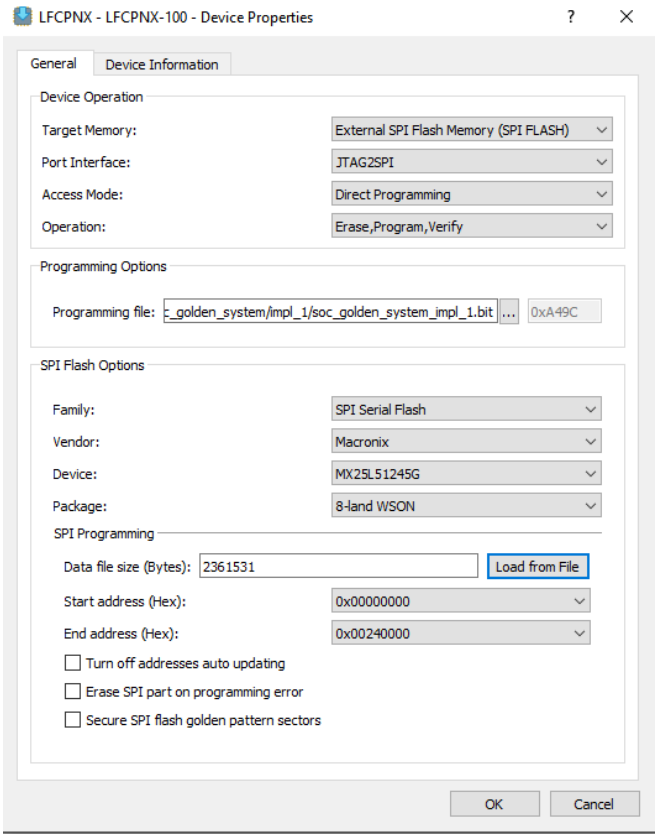

Figure A.16. Radiant Programmer - Binary Flashing Settings

2. Click the **Program Device** icon  or go to the menu item, **Run > Program Device**.
3. Power cycle the CertusPro-NX Versa board.

A.3. MCS Programming Flow

1. To program the MCS file in the SPI Flash in the Radiant Programmer tool, select the options as shown below.

   Ensure that the Primary and Golden firmware (*.bin*) files are programmed at the correct locations in the SPI flash as described in A.1.1. Programming the Primary_App Firmware File (.bin) and A.2.1. Programming the Golden_App Firmware File (.bin). Do not program the respective bitstream (*.bit*) files.

   - Under **Programming Options**, select the *soc_golden_system_impl_1.bit* bitstream file.
   - Under **Device Operation**, select the following options:
     - **Target Memory**: External SPI Flash Memory (SPI FLASH)
     - **Port Interface** : JTAG2SPI
     - **Access Mode**: Direct Programming
     - **Operation**: Erase, Program, Verify
   - Under **SPI Flash Options**, select the following options:
     - **Family**: SPI Serial Flash
     - **Vendor**: Macronix
     - **Device**: MX25L51245G
     - **Package**: 8-land WSON
   - Click Load from File to update the Data file size (Bytes) value.
   - Ensure that the following addresses, which are selected automatically, are correct:
     - **Start Address (Hex):** 0x00000000
     - **End Address (Hex):** 0x007F0000

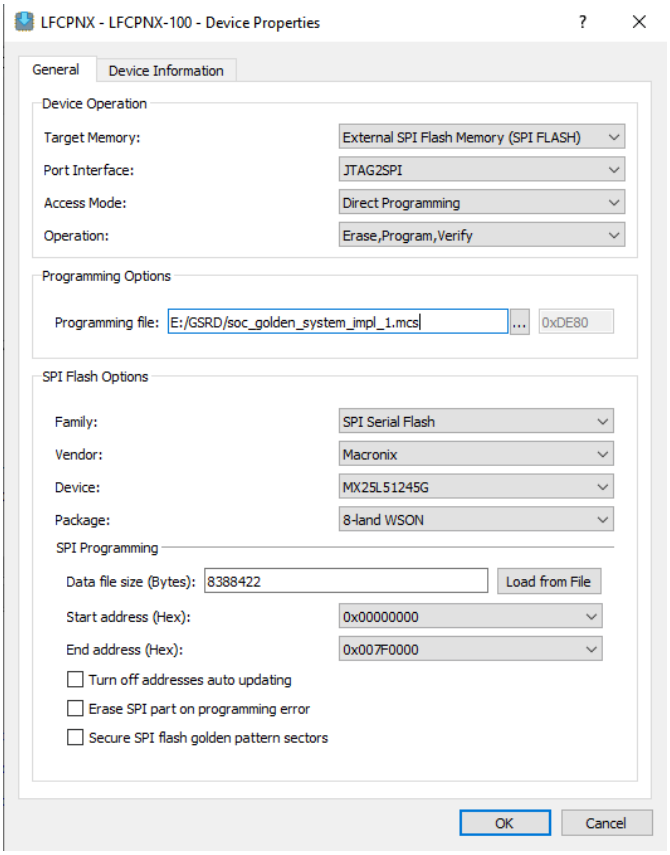

Figure A.17. Radiant Programmer - MCS Flash Settings

2. Click the **Program Device** icon  or go to the menu item, **Run > Program Device**.
3. Power cycle the CertusPro-NX Versa board.

References

- GSRD Reference Design User Guide (FPGA-RD-02283)
- SGDMA Driver API Reference (FPGA-TN-02340)
- QSPI Driver API Reference (FPGA-TN-02339)
- TSEMAC Driver API Reference (FPGA-TN-02341)
- CertusPro-NX web page
- Lattice Solutions IP Cores web page
- Lattice Radiant Software FPGA web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/ Support/AnswerDatabase.

Revision History

Revision 1.1, February 2024

Section	Change Summary
Acronyms in This Document	Changed GSRD description from Golden Software Reference Design to *Golden System Reference Design*.
Introduction	• Changed GSRD description in the bullet point to *Golden System Reference Design*. • Updated Figure 1.2. GSRD FPGA and Firmware Booting Diagram.
Hardware and Software Requirements	• Updated steps in Hardware System Readiness section. • Added Figure 2.2. Reset Button SW3 of CertusPro NX Versa Board. • Minor edit to Software Requirements section.
GSRD Demo Package	Updated the GSRD demo folder and the starting addresses in the flash.
Running the GSRD/GHRD Demo	• Removed the *Programming the GSRD MCS in the External SPI Flash* section. • Moved *Opening the UART Terminal* section up as section 4.1 and made minor edits to the steps within the section. • Added Programming the Standalone Primary/Golden GSRD Bitstream and Firmware section. • Added Programming the GSRD Primary Firmware, Golden Firmware and MCS File section. • Removed the *Terminal Output After MCS Programming* section.

Appendix A	• Renamed this section from *Programming the Standalone GSRD Primary/Golden Bitstream and Firmware in SPI Flash* to *Programming the GSRD Primary/Golden Bitstream, Firmware Binaries and MCS File* and updated the steps in this section.
	• Added the following subsections:
	  • A.1. Primary GSRD Programming Flow
	  • A.1.1. Programming the Primary_App Firmware File (.bin)
	  • A.1.2. Programming the Primary GSRD Bitstream File (.bit)
	  • A.2. Golden GSRD Programming Flow
	  • A.2.1. Programming the Golden_App Firmware File (.bin)
	  • A.2.2. Programming Golden GSRD Bitstream File (.bit)
	  • A.3. MCS Programming Flow

Revision 1.0, December 2023

Section	Change Summary
All	Initial release.

www.latticesemi.com