VORAGO TECHNOLOGIES

AMD FPGA Reconfiguration Using VA416x0

June 2024 Version 0.7

VA41620/VA41630

Abstract

This application note demonstrates a methodology for using a VA416x0 Rad-Hard ARM® Cortex®-M4 microcontroller to reconfigure an AMD Kintex® Ultrascale™ XQRKU060 FPGA in flight.

Having the ability to reconfigure an FPGA in flight offers many benefits. First, this adds reliability to the system, a corrupted FPGA bitstream (configuration file) can be re-uploaded, or a backup configuration can be used instead. Second, this adds flexibility to the system and to the development timeline. Features can be added or completed, bugs corrected after launch of the spacecraft, and multiple mission modes can be supported. Finally, this adds radiation hardness to the system. The VA416x0 can be used as a hardened backbone to the FPGA, protecting against bitstream corruption, monitoring of the FPGA's status and reconfiguring as necessary.

Table of Contents

1 Overview of functionality

The demonstration system contains three major hardware components, and three software components. The hardware consists of a PC (acting as the source of the configuration bitstream files), the VA416x0 Evaluation Kit board, and the Alpha Data® XQRKU060 FPGA development board. Major software components include terminal software running on the PC (TeraTerm), the VA416x0 example firmware (source and binaries available upon request), and the FPGA bitstream files (example XQRKU060 'blinky' binaries included with the firmware).

On powerup of the FPGA, it is in an unconfigured state and is waiting to receive a bitstream. On powerup of the VA416x0, it awaits an 'upload' command and subsequent transfer of a

bitstream file over a UART interface. The VA416x0 then writes the received bitstream to the FPGA using the Slave Serial interface and the FPGA begins running until another reconfiguration is requested.

2 Required hardware

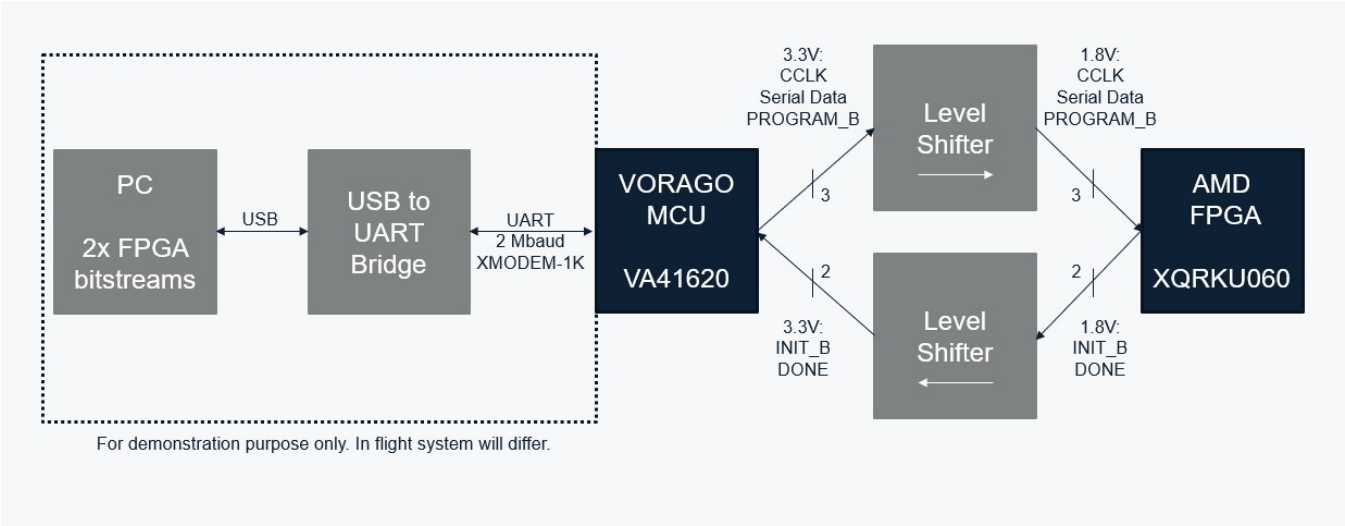
1. Alpha Data® XQRKU060 FPGA development board, ADM_SDEV_BASE
2. ATX power supply to power the FPGA development board, >= 20A (100W) on +5V rail
3. VORAGO Technologies VA41620 or VA41630 EVK board, PEB1-VA41620 or PEB1-VA41630.
4. Xilinx® HW-FMC-105-DEBUG breakout board (Rev. B) – configured for 1.8V I/O
5. Unidirectional 3.3V to 1.8V level shifter (3 pins) – SN74LXC8T245 used in this example.
6. Unidirectional 1.8V to 3.3V level shifter (2 pins) - SN74LXC8T245 used in this example.
7. USB to UART adapter – Silicon Labs CP2102 breakout board used in this example and recommended for fastest transfer rate.
8. Jumper wires to connect the VA416x0 board to level shifters and to the FMC breakout board.
9. Windows PC and 2x USB cables (micro to power VA416x0 board), CP2102 board uses mini-USB.

3 Required software

1. AN1320 software package (zip file) – contains VA416x0 AN1320 example firmware, and 2x example ‘blinky’ XQRKU060 bitstream files. Firmware source and binaries are available upon request: https://www.voragotech.com/contact-us
2. TeraTerm terminal software, version 4.106 or later
3. VA416x0 firmware development toolchain – Keil MDK-ARM, IAR EWARM, or VS Code / CMake / GCC arm-none-eabi 10.3.1 or 11.3.1 supported
4. SEGGER J-link software package, latest version (to download firmware to the VA416x0 board)
5. CP2102 or other USB-UART adapter driver (depends on UART adapter used)

4 Detail of functionality

4.1 System Block Diagram



4.2 FPGA hardware

AMD FPGAs are highly configurable programmable logic devices. The binary file that describes the connections and functionality of the device is called a configuration file or bitstream. This file is generated by the FPGA design tools. On powerup, the FPGA is in and empty or unconfigured state. In order for the FPGA to present the desired functionality, the bitstream file must be loaded into the FPGA through the configuration interface. The FPGA has three ‘mode pins’ that set the operational mode of the FPGA’s configuration interface. Some of these configuration modes are Master types, where the FPGA reads a bitstream from an external device, such as a flash memory chip. Other configuration modes are Slave types, where the bitstream is written to the FPGA by an external device such as a microcontroller. For the purposes of this application note, the FPGA must be set to Slave Serial mode (111b). This mode has three input pins, CCLK, Serial Data, and PROGRAM_B, and two output pins, INIT_B and DONE.

To begin configuration of the FPGA, the PROGRAM_B pin is first driven low. The falling edge of PROGRAM_B clears the FPGA’s current configuration and sets the DONE pin low, and the rising edge starts a new configuration process. When the FPGA begins clearing its internal memory either due to a power-on reset or a falling edge of PROGRAM_B, the INIT_B pin is pulled low by the FPGA. INIT_B is released once the memory clear has completed and signals that the configuration process can continue. Upon the rising edge of INIT_B, an external MCU can drive PROGRAM_B high and begin sending the bitstream file to the FPGA by driving the Serial Clock (CCLK) and Serial Data pins. Once the entire bitstream file has been uploaded, the DONE signal is pulled high by the FPGA to indicate completion of the configuration process.

4.3 VA416x0 hardware

The VORAGO VA416x0 EVK board consists of the VA41620 or VA41630 MCU, clock source, voltage regulators, a Segger J-link OB USB debug interface, and a GPIO breakout board for connection to external hardware. Two GPIO pins (UART0 TX, RX) are used for the PC side connection to send the bitstream files to the MCU, and five GPIO pins connect to the FPGA's configuration interface through 3.3V (MCU side) to 1.8V (FPGA side) level shifters.

4.4 Breakout board and level shifters

The Alpha Data® XQRKU060 FPGA development board provides multiple FMC connectors for IO connectivity. To access the configuration pins used in this application note, a Xilinx® HW-FMC-105-DEBUG breakout board is attached to the FMC port labeled 'CONFIG' on the FPGA board. This breakout board is set up to use 1.8V logic levels. Because the VORAGO VA416x0 is a 3.3V IO device, level shifters are necessary to use between the VA416x0 board and the FMC breakout board. Any unidirectional 1.8V->3.3V / 3.3V->1.8V level shifters will be suitable for use in this application, SN74LXC8T245 is used in this example. There are three signals to be level shifted from the 3.3V to the 1.8V domain (CCLK, Serial Data and PROGRAM_B), and two signals to be shifted from the 1.8V to 3.3V domain (INIT_B and DONE).

4.5 PC connection / software

The bitstream files for configuring the XQRKU060 can be quite large, from a few megabytes up to ~24MB or so. Because there isn't sufficient memory on the VA416x0 EVK board, a Windows PC is used as the source of the bitstream files to upload. This example uses a Silicon Labs CP2102 USB-UART bridge to connect the PC to the VA416x0 and transfer the bitstream files using the XMODEM-1K protocol (with CRC checksum) at 2Mbaud. The terminal software TeraTerm is used to establish the UART connection to the MCU, initiate the bitstream upload with the '$upload' command, and transfer the bitstream file via XMODEM-1K.

4.6 VA416x0 firmware overview

The AN1320 firmware for the VA416x0 consists of the following components:

1. VA416xx_hal components – timer, clkgen, ioconfig peripheral drivers, etc. (common/drivers)
2. CMSIS – ARM components, system_va416xx.c/.h, core_cm4.h, etc. (common/mcu)
3. main.c – main – initializes the MCU, and starts the command processor
4. board.c – contains the GPIO configuration
5. board.h – contains the FW version, FW configuration (including the UART baud rate)
6. cmd_interface.c/.h – the UART command processor
7. crc.c/.h – CRC engine (part of the XMODEM protocol)
8. hardFault_handler.c – fault handler
9. uart.c/.h – UART driver

10. startup_va416xx.s – startup / vector table file
11. xmodem.c/.h – XMODEM protocol / bitstream receive handling
12. xqrku060.c/.h – handles signaling to / from the XQRKU060 FPGA

5 Software setup process

1. Install the VA416x0 firmware development toolchain (Keil, IAR, or GCC / CMake / VS Code) using the VA416x0 Evaluation Kit user's guide. A copy of the user's guide is included in the AN1320 software zip.
2. Install TeraTerm V4.106 or later.
3. Install the CP2102 USB-UART converter driver, or other UART adapter driver.
4. Open the AN1320 firmware project, build and download the firmware to the VA416x0 EVK board. For VS Code/CMake/GCC, the build commands are: 'make init', then 'make fpga'. Make sure to choose the download/debug option 'JLink: fpga run in FRAM' to write the firmware into persistent memory.

6 Hardware connection process

6.1 FPGA board setup

1. Ensure the FPGA board is set up in configuration mode 111b (slave serial mode). This is done by installing 0 ohm resistors on R221, R222, and R223, and removing any resistors from the pads directly to the right of R221, R222, and R223 (back side of the FPGA board).
2. Make sure the power switch SW3 on the FPGA board is in the 'off' position.
3. Connect the Xilinx HW-FMC-105 breakout board to the FMC port labeled 'CONFIG'
4. Connect the ATX power supply 24-pin cable to connector J5 on the FPGA board.
5. Connect the mains power cable from an outlet to the ATX power supply.

6.2 PC to VA416x0 EVK connections

1. Using a jumper wire, connect J7 pin 3 (GND) on the VA416x0 EVK Core Board to 'GND' on the CP2012 breakout board.
2. Using a jumper wire, connect J7 pin 1 (MCU TX) on the VA416x0 EVK Core Board to 'RXI' on the CP2102 breakout board.
3. Using a jumper wire, connect J7 pin 2 (MCU RX) on the VA416x0 EVK Core Board to 'TXO' on the CP2102 breakout board.
4. Using a USB cable, connect the CP2102 USB-UART adapter to the PC.

6.3 VA416x0 EVK to level shifter connections

For this section and section 6.4, 'Level shifter 1' (or LS1 for short) refers to the 3.3V (input) to 1.8V (output) level shifter (CCLK, DATA, PROGRAM_B). Level shifter 2 (or LS2 for short) refers to the 1.8V (input) to 3.3V (output) level shifter for the signals INIT_B and DONE. Both

level shifter ICs in this example are TI SN74LXC8T245. For other level shifters, some of the connections may differ.

1. Using jumper wires, connect +3.3V power on the VA416x0 EVK board (J39) to LS1 pins 1 and 2 (VCCA and DIR), and LS2 pin 1 (VCCA).
2. Using jumper wires, connect GND on the VA416x0 EVK board (J39) to LS1 pin 11 (GND), LS2 pin 2 (DIR), and LS2 pin 11 (GND).
3. Using a jumper wire, connect VA416x0 GPIO pin PF8 to LS1 pin 3 (A1).
4. Using a jumper wire, connect VA416x0 GPIO pin PF5 to LS1 pin 4 (A2).
5. Using a jumper wire, connect VA416x0 GPIO pin PF3 to LS1 pin 5 (A3).
6. Using a jumper wire, connect VA416x0 GPIO pin PF7 to LS2 pin 3 (A1).
7. Using a jumper wire, connect VA416x0 GPIO pin PF6 to LS2 pin 4 (A2).

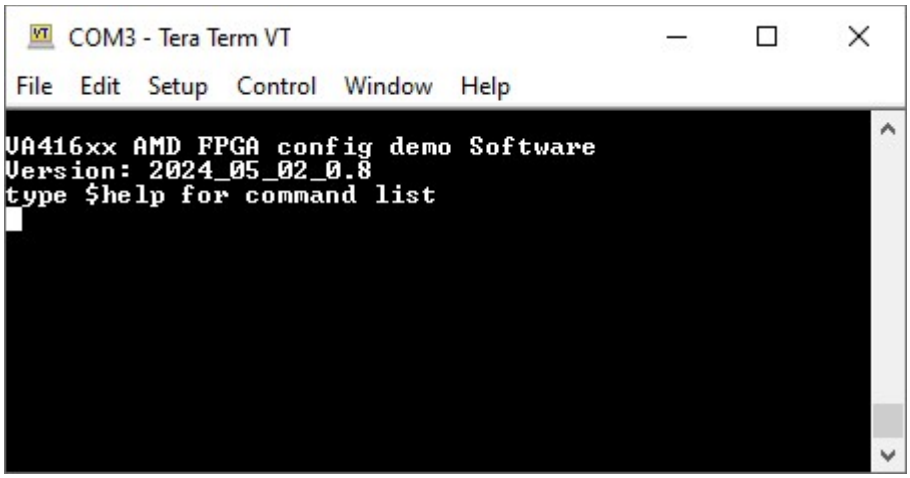
6.4 Level shifter to FMC-105 breakout board connections

1. Using a jumper wire, connect LS1 pin 24 (VCCB) to FMC-105 J6 pin 6 (VADJ 1.8V).
2. Using a jumper wire, connect LS2 pin 24 (VCCB) to FMC-105 J6 pin 5 (VADJ 1.8V).
3. Using a jumper wire, connect LS1 pin 13 (GND) to FMC-105 J16 pin 3 (GND).
4. Using a jumper wire, connect LS2 pin 13 (GND) to FMC-105 J16 pin 4 (GND).
5. Using a jumper wire, connect LS1 pin 21 (B1) to FMC-105 J2 pin 9 (PROGRAM_B).
6. Using a jumper wire, connect LS1 pin 20 (B2) to FMC-105 J23 pin 1 (Serial Data).
7. Using a jumper wire, connect LS1 pin 19 (B3) to FMC-105 J2 pin 13 (Serial Clock).
8. Using a jumper wire, connect LS2 pin 21 (B1) to FMC-105 J2 pin 11 (INIT_B).
9. Using a jumper wire, connect LS2 pin 20 (B2) to FMC-105 J2 pin 17 (DONE).
10. Using a jumper wire, connect FMC-105 J23 pin 10 (FMC_HA23_P) to FMC-105 J19 pin 2 (GND).

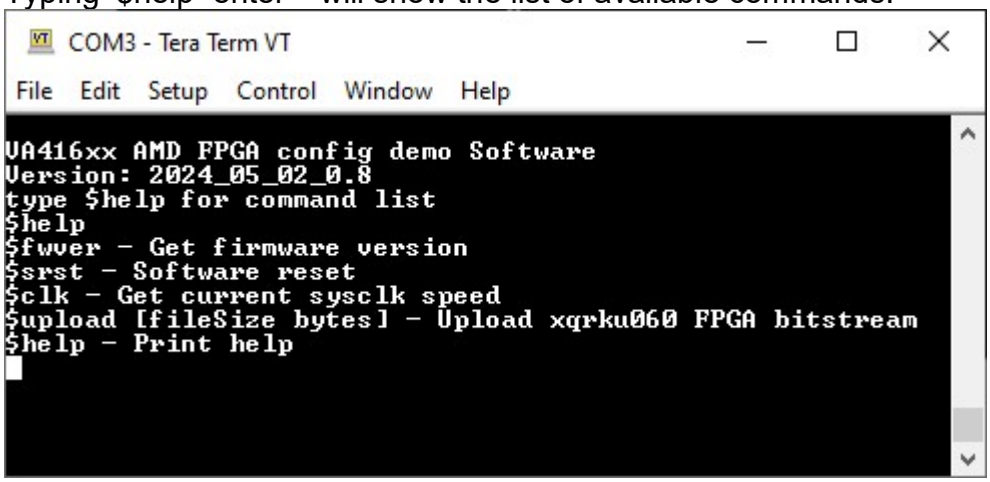
7 Uploading a bitstream

Once the hardware has been set up and the AN1320 firmware has been flashed to the VA416x0, the following procedure can be used to transfer a bitstream to the XQRKU060 FPGA:

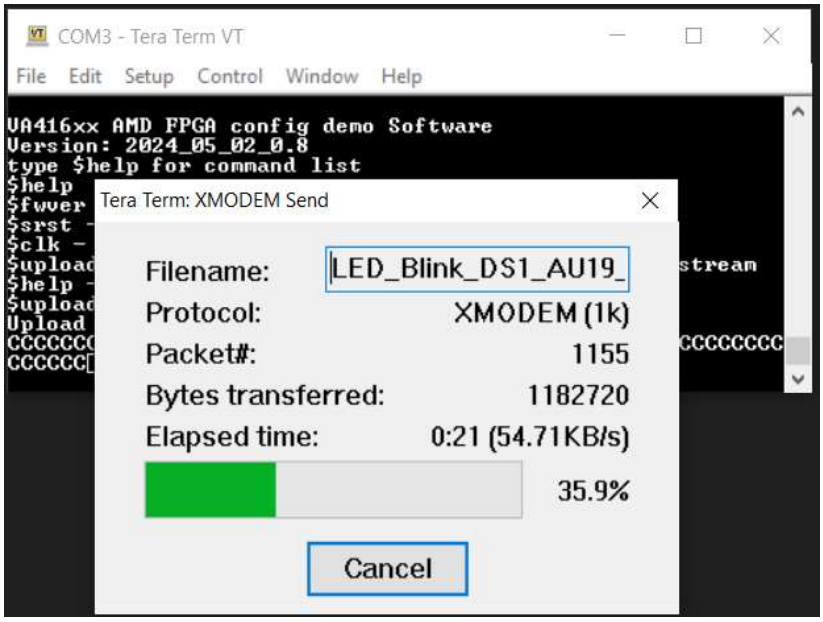
1. Power on the VA416x0 board by plugging in the micro-USB cable to J17, or supplying 5V to the DC jack J16.
2. Power on the FPGA board by turning on the switch on the ATX power supply, and moving the power switch SW3 on the FPGA board to the ‘on’ position.
3. Connect the USB-UART adapter to a USB port on the PC, make note of the COM port number that it enumerates as (can be seen in Windows Device Manager)
4. Open TeraTerm, and in the New Connection window select ‘Serial’, and select the COM port of the USB-UART adapter.
5. Click ‘Setup’, then ‘Serial port…’ and under ‘Speed’, change the baud rate to 2000000.
6. Press and release the reset button on the VA416x0 EVK board, and you should see the following text:



7. Typing '$help<enter>' will show the list of available commands:



8. To upload a bitstream file, first enter the command '$upload [fileSize]', where [fileSize] is the size of the bitstream file in bytes. For the LED_blink_DS1_AU19.bit example bitstream, the command would be '$upload 3295265'. For LED_blink_DS4_AT18.bit, the command would be '$upload 3295733'.
9. The AN1320 firmware is now waiting for an XMODEM transfer of the bitstream file. To send the file, in TeraTerm click 'File', then 'Transfer', then 'XMODEM', then 'Send'. Make sure the '1K' option is selected. Select the desired bitstream file, then click 'Open'. The transfer will begin.



10. Once the transfer has completed, the ‘Done’ LED D4 should be illuminated on the FPGA board, and if writing one of the example LED ‘blinky’ bitstreams, either LED DS1 or DS4 should be blinking on the Xilinx FMC-105 breakout board to indicate successful configuration of the FPGA.
11. To upload a different bitstream, start again at step 8.

8 Conclusion

This application note shows how to use the VORAGO Technologies VA416x0 microprocessor to configure / reconfigure an AMD XQRKU060 FPGA, and examines the hardware setup, software setup, and operation of the example application. This can be used to add reliability, flexibility, and radiation hardness to an FPGA based spacecraft subsystem.

9 Other Resources

UG570: AMD UltraScale™ Architecture Configuration : https://docs.amd.com/v/u/en-US/ug570-ultrascale-configuration

Alpha Data® ADM-SDEV-BASE/XCKU060 User Manual: https://www.alpha-data.com/xml/user_manuals/adm-sdev-base%20user%20manual_v1_7.pdf

VORAGO Application Notes: https://www.voragotech.com/documentation

VORAGO ARM® Cortex®-M4 Products: https://www.voragotech.com/arm-cortex-m4-family

Revision log:

May 14, 2024 – Initial template created (V0.1)
May 15, 2024 – Abstract, sections 1, 2, and 3, and 9 completed (V0.2)
June 3, 2024 – Completed sections 4.1, 4.2, 4.3, 4.4 (V0.3)
June 4, 2024 – Completed sections 4.5, 4.6, 5 (V0.4)
June 5, 2024 – Completed sections 6.1 6.2, 7, 8 (V0.5)
June 5, 2024 – Completed sections 6.3, 6.4 (V0.6)
June 6, 2024 – Edited sections 3 and 6, added support for Keil and IAR firmware toolchains (V0.7)