# Error Correcting Product Accumulator Demonstration Setup Guide

Courtesy Maitrix, LLC

October 2021

# Table of Contents

# Error Correcting Product Accumulator Demonstration Guide

## Summary:

Maitrix has released a fully functional demonstration of its highly innovative error correcting arithmetic. The demonstration includes two versions; 1) a precompiled version which includes a bit file for programming the Arria-V FPGA and a hex file for downloading the sample demonstration code to the NIOS-II processor, and 2) a version including full source code in Verilog which can be compiled and fully analyzed. Please contact Maitrix for the source code version.

Note that this demonstration uses the Arria-V GX development card, which is now obsolete. These boards might be found on eBay. Maitrix is working to port the application to the Arria-10 development card soon.

## Components Needed:

The following hardware items are required to run the error correcting product accumulator demo:

1. Altera Arria-V Starter Kit (development board): DK-START-5AGXB3N
2. Terasic Altera Communication Card: part #P0078
3. Function generator with 3MHz to 10 MHz frequency adjustment
4. BNC cable with BNC to SMA adapter
5. Serial RS-232 communication cable (DB-9 female with cross connection)
6. USB to RS-232 converter cable to add comm port to PC
7. PC with Comm port & terminal software configured for VT102 emulation

## Software to run the pre-compiled Demo:

1. Quartus Web Edition 18.1 or greater (free download)

## Software to compile the Verilog source version:

1. Quartus Standard Edition 18.1 or greater with Arria-V devices installed (requires license)

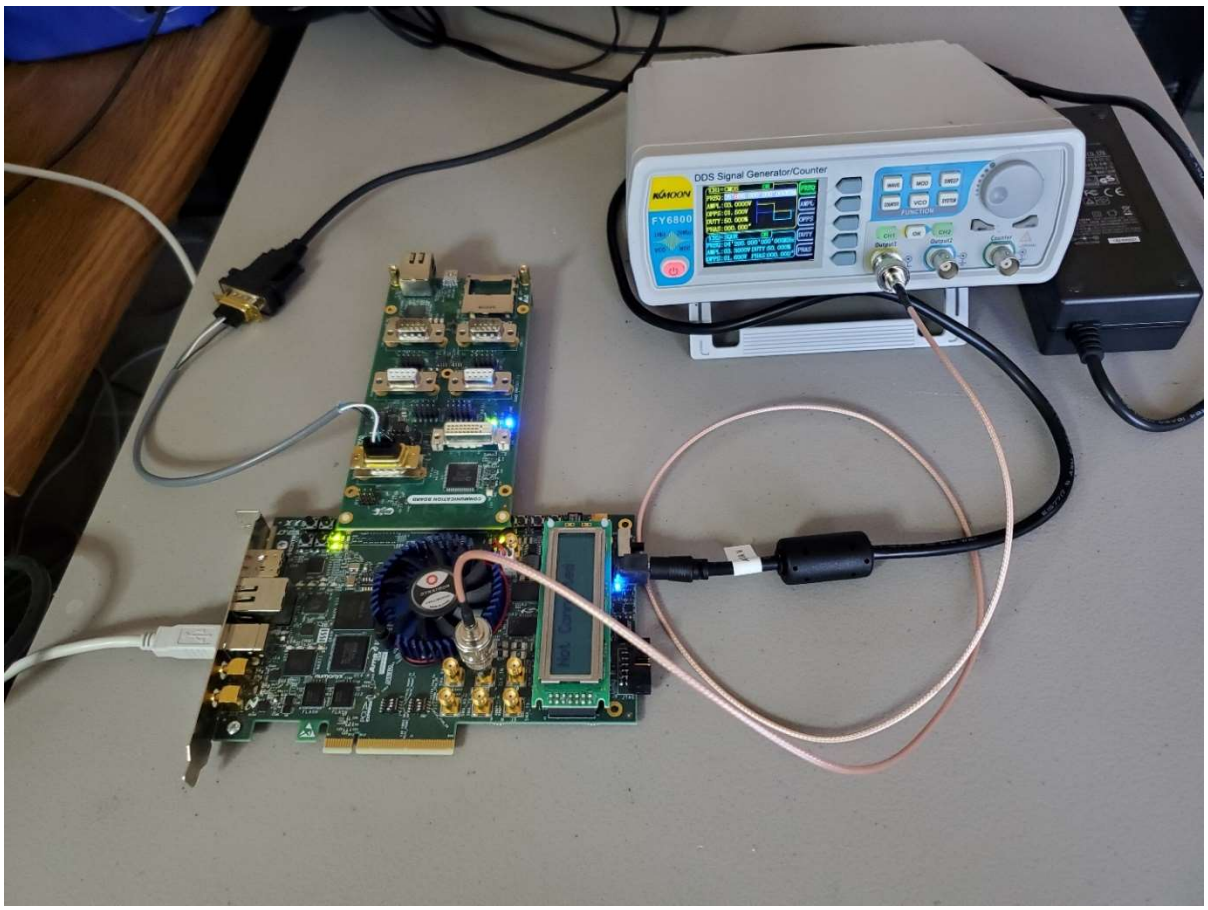
## What is included in the Maitrix zip file:

1. FPGA configuration file: demo_freqcnt375.sof
2. A5_ec_demo_1.elf
3. Run_this_app.bat
4. Quartus 18.1 project directories including Verilog source files
5. Nios Software project files
6. Maitrix end-users License notice

# Step by Step Instructions for running pre-compiled demonstration:

## Setting up the hardware:

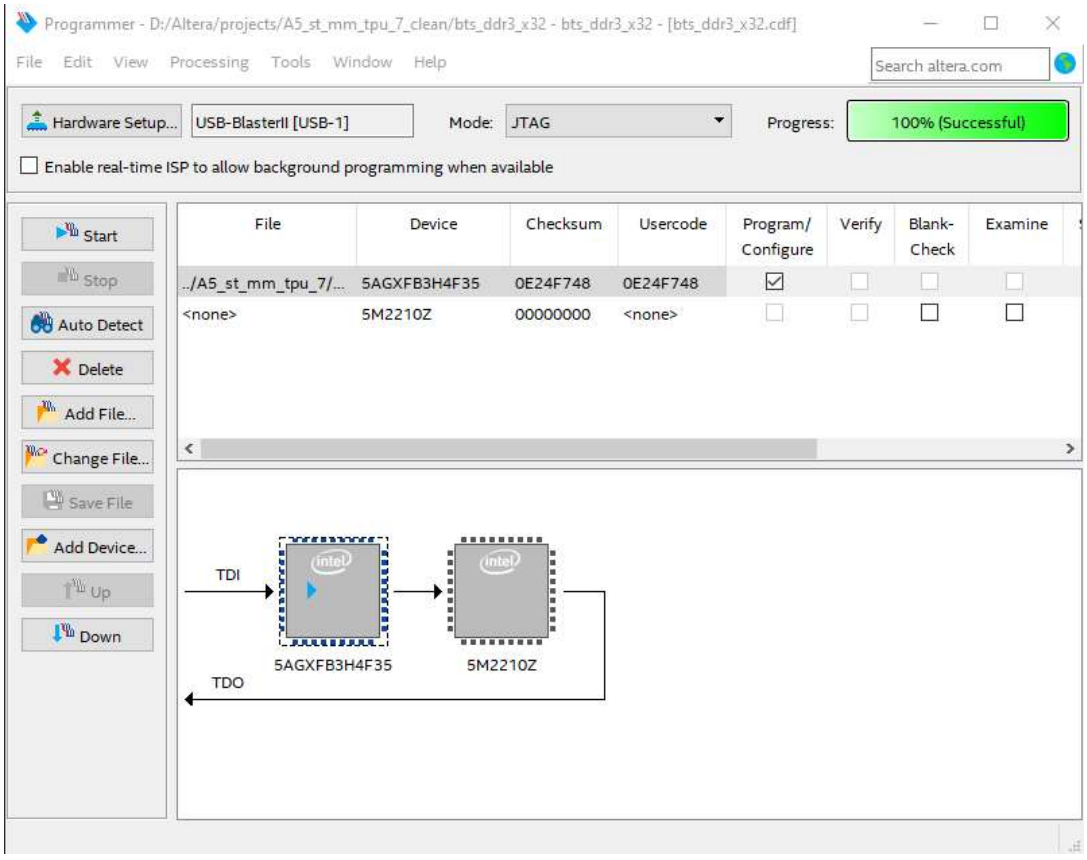
1. Connect the Terasic HSMC communications card to the Altera Arria-V development board HSMC port.
2. Connect a serial cable to the RS-232 (0) port DB-9 connector on the HSMC communications card. See cable drawing in Appendix A to make your own RS-232 serial cable. Connect the serial cable to a suitable RS-232 port on your PC. You may require a USB to RS-232 converter cable to install an RS-232 port on your PC.
3. Connect the BNC to SMA adapter to “clk_in” SMA (J7) connector on the Arria-V development board.
4. Calibrate the frequency generator output to achieve a voltage output of 0-3.3 volts. Make sure offset is set correctly. Pre-adjust the frequency generator output to 4 MHz.
5. Connect a BNC cable from the frequency generator output to the BNC to SMA adapter installed on the development board from step 3.
6. Connect the Arria-V power supply and power on the development board and your PC. Your finished set-up should look like:



## Configuring the FPGA Hardware for pre-compiled demo:

1. Un-zip the Maitrix supplied demo files to a suitable location on your PC. For example, C:\Maitrix\EC_demo.
2. Make sure Quartus is properly loaded on your PC and the USB cable driver is installed.
3. To program the Arria-V FPGA, connect the USB cable supplied in the Arria-V development kit to your PC and to the Arria-V development board.
4. Run the Quartus development software.
5. Launch the device Programmer by clicking on **Tools->Programmer**.
6. Make sure the correct Hardware setup is shown on the upper left box of the Programmer application. If not, press the Hardware setup button and troubleshoot your connection.
7. Click on the Auto Detect button to detect the FPGA device on the Arria-V FPGA card.
8. The device may need to be manually adjusted to be as shown below.
9. Click the **Add File** button and navigate to the directory where you extracted the demo files and click on the **demo_freqcnt_1.sof** file.

   The programmer screen should look like:




10. Click on the Start button to download the FPGA **demo_freqcnt_1.sof** file to the FPGA. The FPGA device should be configured once the programmer software progress bar has completed. It may take several attempts to get the FPGA programmed. The user may also use the NIOS command shell, change the directory to the demo directory, and invoke the command: **nios2-configure-sof**

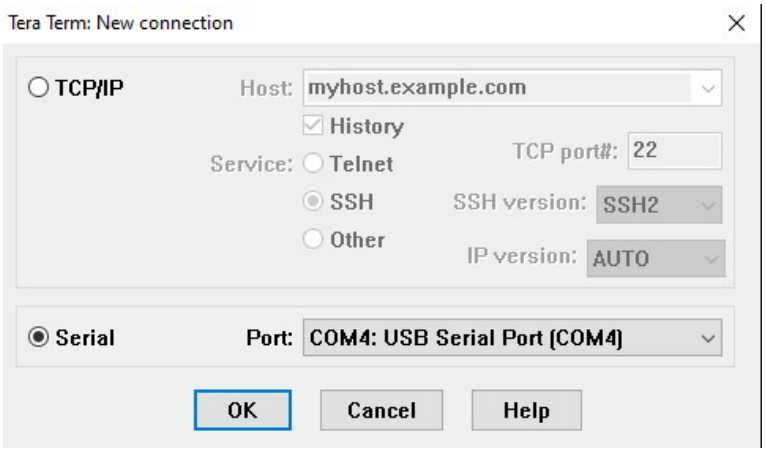
## Downloading NIOS software for pre-compiled demo:

Downloading the software to the NIOS processor is performed after the FPGA is configured in the preceding step.

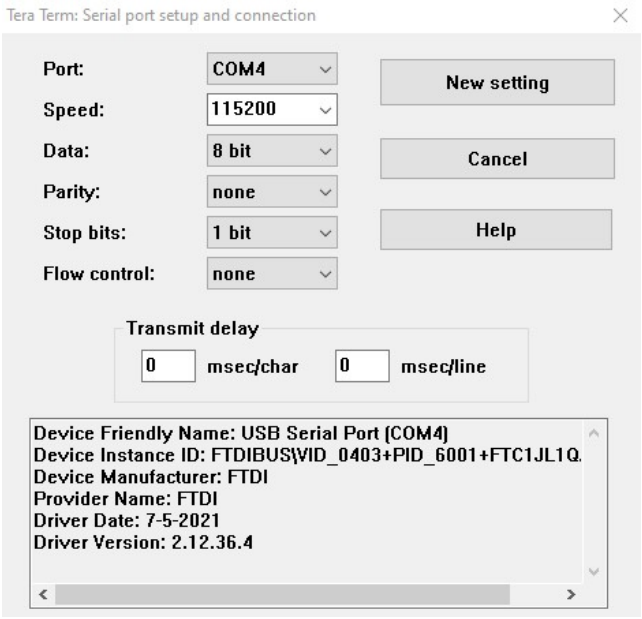
1. Run a NIOS command shell program located in the Quartus program files group under your windows start menu.
2. In a NIOS-II command shell, type the CD command to change to the directory where you extracted the demo files. For example: CD C:/maitrix/EC_demo
3. In a NIOS-II command shell, type the following command to download the *A5_ec_demo_1.elf* demo file to the NIOS2 processor: **nios2-download -g A5_ec_demo_1.elf**
4. To reset the processor and run the code, type this command: **nios2-download -g**
5. The NIOS demo program should be loaded into the FPGA if the command in step 3 is successful. It should start to run automatically.

## Configuring the PC for Terminal Emulation

1. Any terminal program with VT102 terminal emulation can be used. These instructions cover downloading and using **TeraTerm** terminal emulator software.
2. Your PC must have an RS-232 serial port. If not, then a USB to serial port adapter is recommended. Make sure your USB serial port is compatible with your version of Windows.
3. An RS-232 NULL modem serial cable is required. We used a homemade NULL Modem RS-232 cable with female-pin RS-232 connectors at both ends. This cable will connect to the HSMC communications daughter card at the DB-9 connector designated as **RS-232 (0)**.
4. Start the TeraTerm application. You should see a dialog box similar as shown:

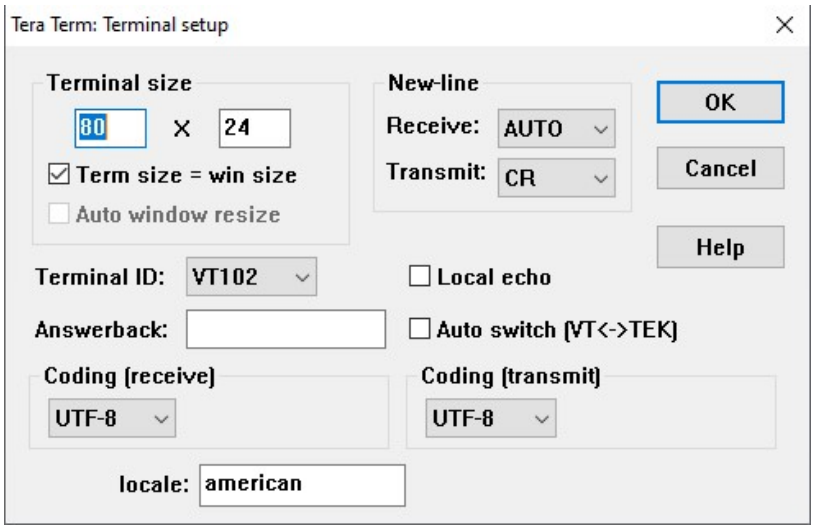



5. Configure the serial port to the available serial port on your PC. In the example above, **COM4: USB Serial Port** is used. Use the Device Manager under Windows if you have problems with your serial port.

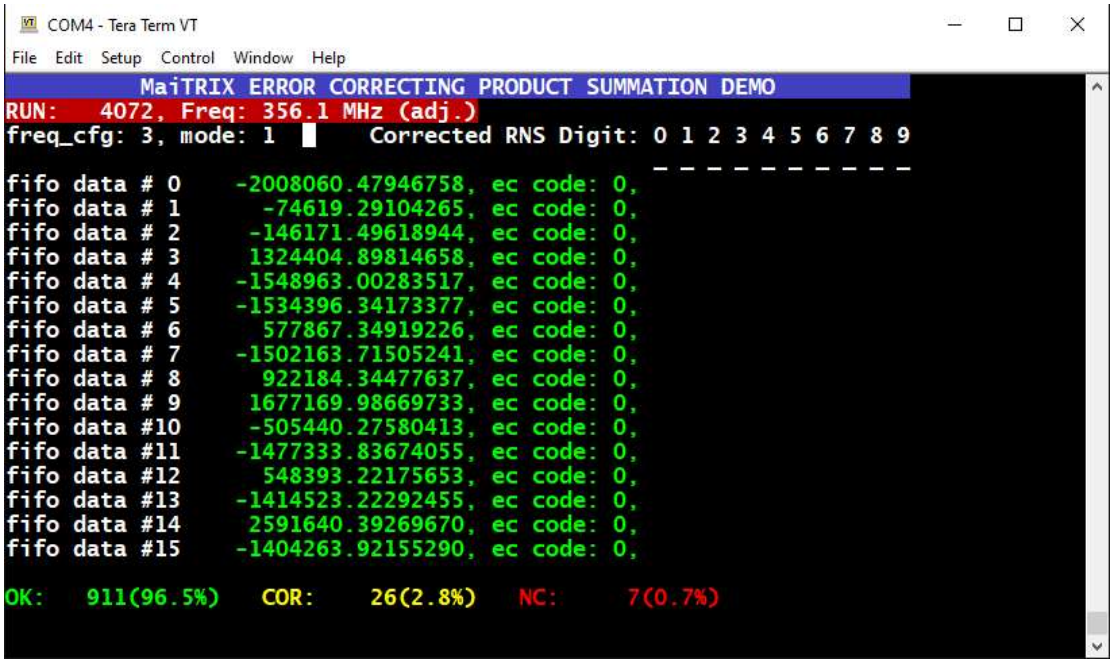


6. Configure the serial port for **115200 baud, 8 bits, no parity and 1 stop bit** as shown in the dialog box above. Press the new setting button to continue.

7. Using the **Setup** menu in TeraTerm, access the **Terminal Setup** option. Select the Terminal ID: **VT102** and select **AUTO** for the receive new-line. These setting should look like:




8. The user may also adjust font size and other appearance settings as desired. Press ‘c’ character in TeraTerm to force NIOS2 to clear and refresh the screen. The following terminal screen shot with the error correcting demo running looks like:



## Using the Error Correcting Product Summation Demonstration

The demonstration performs a total of 16 product summations. Each product summation consists of sixteen 64-bit multiplies and sixteen 128-bit additions. The arithmetic is entirely performed in a signed RNS fixed-point format. The final answer is formatted as a signed 32.32 format fixed-point fraction.

An error correction unit is connected to a results FIFO and allows the demonstration to detect if a digit error in the final product summation has occurred. If a single digit error has been detected, the digit is corrected in hardware. These are shown as results in Yellow. If two or more digits are in error, there is a high probability the error is detected, but it cannot be corrected. This result is incorrect and is shown in RED. Final product summations that are correct are shown in Green.

To learn more about the error correcting demonstration, read the project report entitled:

*Error correcting product accumulator pipeline analysis*
*Which can be found at:*
Publication – MaiTRIX

### Generating Errors:

The error correcting demonstration includes the ability to induce real world errors into the arithmetic pipeline. This is performed by increasing the operating frequency of the product summation unit until errors begin to appear. Increasing the operating frequency of the product summation unit can be performed using a frequency generator connected to the Aria-V development board as shown in the setup, and setting the clock source to be adjustable by selecting Freq_cfg = 3.

### Supported Hotkeys

There are several hotkeys supported by the *A5_ec_demo_1.elf* application. All hotkey characters are lowercase only. Remember to click into the terminal emulator PC application before hitting the hot key on the keyboard.

The following hotkeys are supported by the demonstration:

| Hotkey | Function | Notes |
|---|---|---|
| c | Clear Screen | Clears the terminal screen and allows re-sync with terminal |
| d | Numeric Format | Toggles display between Decimal and Hexadecimal |
| r | Clear Counters | Clears the OK, COR and NC counters. |
| s | Pause | Toggles between start and stop; used to examine results |
| m | Switch mode | Up to 4 data modes are available; see below |
| f | Change clock source | Three different clock sources are available; see below |
| v | Change value | Used for static values mode to change the static values |

### Clear Screen (c)

This hotkey instructs the NIOS2 to clear the terminal by issuing a clear screen command. This is useful if the screen is cluttered, and the application has lost synchronization with the terminal.

### Numeric Format (d)

Each time this hotkey is hit, the demonstration switches the numeric value format displayed from decimal to hexadecimal and from hexadecimal to decimal radix format.

### Clear Counters (r)

Hitting this hotkey clears the OK, COR and NC counters. The OK counter counts all the product summations that are OK (no error). The COR counter counts the number of product summations that were corrected (corrected error), and the NC counter counts the number of product summations that are not corrected (un-corrected error). A "runs" counter is also shown in the RED bar area; this counts total product summations but is not cleared.

### Pause (s)

Hitting this hotkey will pause the product summation process. Hitting the hotkey again will resume the product summation process.

### Mode (m)

Hitting this hotkey will switch the mode of the demonstration. There are three modes. The current mode is shown at the top left of the terminal screen. The following modes are available:

| Mode | Description | Notes |
|---|---|---|
| 0 | Data read from LUT and is re-run but offset by 1 | Tests the LUT and how the errors change with LUT walk |
| 1 | New random data is processed each run | Random data test |
| 2 | A fixed data set is run. | Used to demonstrate the repeatability of errors |
| 3 | A fixed single value is processed | Used to demonstrate the repeatability of errors |

### Clock source (f)

When the application starts, the product summation clock source is switched to an internal PLL and set to 375 Mhz. The pipeline operating frequency is seen at the top of the screen in RED. The FPGA product summation hardware is rated to only 250 MHz. Therefore, over-clocking occurs and so arithmetic errors occur. However, the clock source may be changed by hitting the 'f' hotkey. The frequency configuration (**freq_cfg:**) chosen is also shown top left of screen. The following frequency settings are available:

| Freq_cfg: | Frequency Mode | Notes |
|---|---|---|
| 1 | 50 MHz fixed (stable) | Stable low frequency clock |
| 2 | 375 MHz fixed (over-clock) | Fixed internal overclock |
| 3 | Adjustable frequency | Requires a function generator |

### Change Value (v)

Hitting this hotkey changes the values processed when using static value modes, such as mode 2 & 3. Therefore, the data dependency of the arithmetic errors is more obvious and can be experimented with.

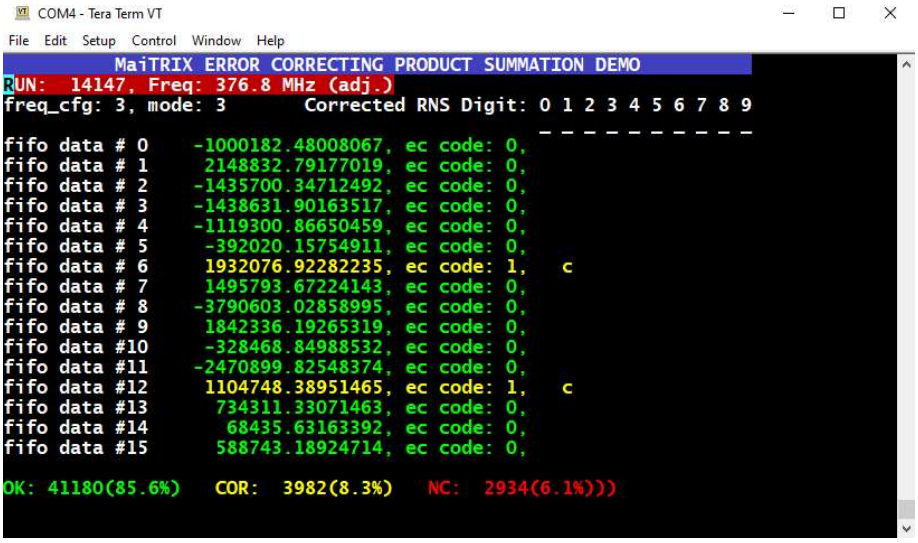
## Types of Errors:

This section shows some of the different error conditions that may be studied using the demo.

### Single Digit Errors:

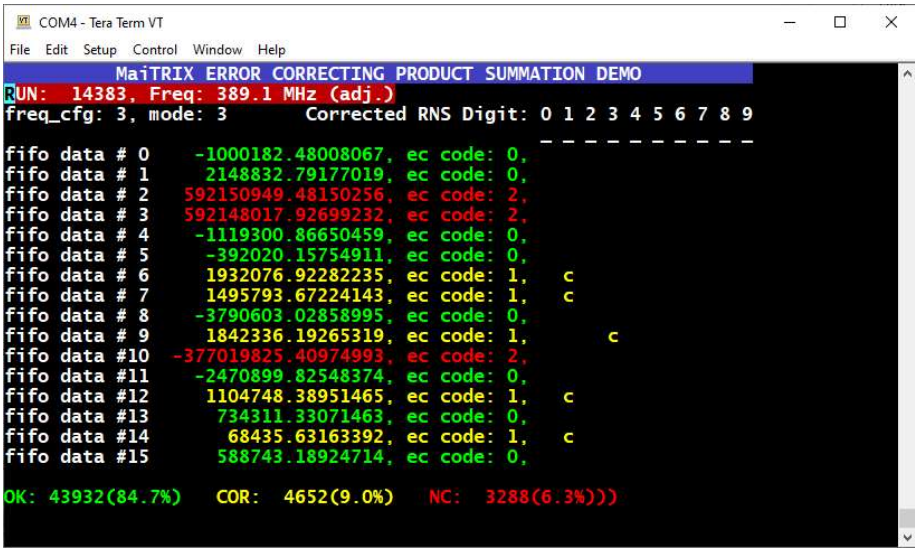
Single digit errors are detected, and the resulting product summation is corrected. These results are shown in YELLOW. The digit that is in error is shown as the a 'c' character printed under the digit column of the error digit. There are 10 total digits in the RNS arithmetic representation.

The example below shows two product summations in error, but these summations were corrected. The digit in error is shown with a 'c' character under the digit column "1". Note two product summations were corrected. Both digit errors occurred in the same digit position. This is common since the FPGA circuit will route such that one digit is slower than all others:




### Two or more digits in error:

Two or more digits in error can be detected, however their position is unknown. They are shown in RED. Furthermore, the value in RED cannot be corrected and is in error:



## References:

HSMC communications card:

Terasic - Daughter Cards - HSMC Communication Card

Arria-V development board (now obsolete!)

(these boards may be found on EBay)

Intel Arria V GX FPGA Starter Kit

www.Maitrix.com