

AION FPGA Communication

AION FPGA Communication

Earl Mai

emai@epicblockchain.io

July 24th, 2018

Revision 0.1

AION FPGA Communication

Scope	3
Universal Asynchronous Receive Transmit (UART)	4
Send	4
Receive	4
Temperature	5
Software	5

AION FPGA Communication

Scope

This document describes the way a user would interface with the FPGA platform. This is meant to aid in creating mining software/communication with the design.

Universal Asynchronous Receive Transmit (UART)

A UART design based on the works of “Open-Source-FPGA-Bitcoin-Miner” has been used to allow the design to communicate with the host. The UART is currently not configurable (needs register control) but can be built to meet the desired communication speeds. The current implementation is set to run at 115200bps and can run as fast as your serial port can handle.

Data size: 8bits
Parity: None
Stop bits: 1
Baud: 115200bps

Data is always sent in ASCII encoded text format representing hexadecimal numbers. The stream of input is converted when received by the design. This also holds true when data is sent back to the host. Therefore the valid input characters are [0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F].

Send

The design expects a 480bits of data which consists of a blockheader hash and nonce. When a carriage return/line feed is sent, the values are flopped, the design resets and the core begins the Equihash algorithm.

bytes	Description
32	H(Blockheader)
28	Nonce

Receive (Solution)

When a solution is found, the UART will transmit the solution with header, partial nonce and pairs which are defined in detail in the table below.

bits	Description
32	Header (0xDEADBEEF)
32	Nonce[31:22]
32	Pair 0
32	Pair 1
32	Pair 2

AION FPGA Communication

32	Pair ...
32	Pair 510
32	Pair 511

Receive (Temperature)

The current bitstream uses the XADC sensor from Xilinx to report the current FPGA temperature every 4 seconds. This data is transmitted across the UART 16bits at a time and can easily be differentiated from the solution data by it's size.

$$\text{Temperature (C)} = \text{Value} * 503.975 / 65536 - 273.15$$

Software

There is various software available that support serial communication. From Putty and the original Tera Term have been used extensively over the years but we recommend using any tool that supports more than the typical serial COM1-COM4. It so happens our in-house machine always fixes itself to COM12 making a lot of legacy software useless.

- <https://ttssh2.osdn.jp/> (Tera Term Pro)
- <https://www.putty.org/> (Putty)