$\mu\mathcal{P}coin$: A Simplified Cryptocurrency Protocol

Herrick Fang and Teerapat (Mek) Jenrungrot
Email: {hfang,mjenrungrot}@hmc.edu

November 1, 2017

1 Introduction

Cryptocurrency, such as Bitcoin, has generated a lot of attention in the recent years. The underlying technology has the potential to disrupt the global economy, though most people still do not have a full understanding of the technology behind it. To make the underlying technology more accessible to the public, we have decided to implement a simplified version of the Bitcoin protocol to demonstrate how people can transact digital currency. The blockchain technology behind the Bitcoin protocol and other cryptocurrencies requires lots of intensive computation, which is used for reaching consensus through a peer-to-peer network adhering to a protocol for validating new blocks. To account for intensive computations related to blockchain technology, we–in this project–will demonstrate the utility of FPGAs in accelerating computations in the cryptocurrency applications.

In this proposal, we will create a prototype system, composed of 2 users in the same network to demonstrate the core functionality of the Bitcoin protocol such as making transactions and generating coins (commonly known as Bitcoin mining). Specifically, two sets of a Raspberry Pi and an FPGA are used to represent two users in the network. The two sets can make transactions and perform proof-of-work (mining process). Another Raspberry Pi will be used to show the list of transactions in the network by displaying blockchain results on a monitor. Note that this last Raspberry Pi does not participate in any transactions making but focuses on illustrating the functionality of our prototype system, so it will be referred to as an observer. Figure 1 depicts the overview of our proposed system.

2 $\mu\mathcal{P}coin$ Protocol

In this protocol, we consider that a single user has the following 5 components:

1. HTTP Server (Apache) - provides an API to manage the blockchain, addresses, transactions, mining requests, and peer connections
2. Miner - gets the list of pending transactions and creates a new block containing the transactions. To make the protocol simple, every block has at most 2 transactions.
3. Node - perform all the data exchange between nodes, including getting new peers, new blocks, and new transactions. The node rebroadcasts every new piece of information it receives unless the information is already in its database of peers, transactions, or blockchain.
4. Operator - handles wallets, addresses, and transaction creation. Note that one operator can have multiple wallets and addresses.
5. Blockchain - contains two pieces of information: the block list and the transaction map. This component verifies and synchronizes the arriving blocks and transactions.

These components are shown in Figure 2. This protocol uses a HTTP interface to control every component. For instance, a user can make a transaction by making a HTTP request on the Raspberry Pi. Once a transaction is made, the information is broadcasted to the entire network. When another Raspberry Pi receives new information, it performs synchronization of the blockchain and the corresponding transactions. Specifically, it checks if the arriving block/transaction has already existed in the data storage. The mining part is invoked by the HTTP request as well.

3 Project Scope

In this section, we describe the scope of our project by explaining the functionality of the simplified Bitcoin protocol and the components that will be implemented and how each component interacts with one another. Based on Figure 1, the system consists of 3 Raspberry Pi, 2 of them (known as users) connected to their own FPGA and another one (known as an observer) connected to an external display.

Our intent is to create to create a replicable model to demonstrate the general process of transacting digital currency. Thus, our project focuses on a prototype implementation of a blockchain network, so users can make valid transactions to others and everyone in the network can receive the transactions correctly.

Deliverable

Raspberry Pi (Users):

1. HTTP Server – Contains the networking protocols for sending and receiving large amounts of data and controlling the other 4 components. We will demonstrate a working HTTP server that can control other components hosted on each Raspberry Pi.
2. Miner – Contains the interface to the FPGA. We will demonstrate that data can be sent through the SPI protocol to aid computations.
3. Node – Creates and participates in a peer-to-peer network. We will demonstrate that a node can find other nodes and request information on the blockchain from them by listening on some HTTP port.
4. Operator – Operates the wallets and addresses. We will demonstrate that an operator can maintain the list of wallets and addresses after making transactions, and that overspending is not allowed.
5. Blockchain – Contains all of the information inside a blockchain. We will demonstrate that the entire blockchain is valid, with correct information through a GUI display of the observer.

FPGA

1. Miner – Implements the SHA-256 Algorithm specified by FIPS 180-4 [3]. We will demonstrate the accuracy and running-time of the SHA-256 implementation.
2. Blockchain – Verifes a hash in case of receiving a new block. We will demonstrate the ability to verify a hash and the running-time of our implementation.

4 Conclusion

Our project is two-fold. First, we seek to implement a protocol that encompasses the scope of the cryptocurrency world, focusing on what a large overview of the blockchain protocol underlying crytocurrency protocol and its implementation. Since our project is a re-creation of an existing technology that is not commonly understood, we also believe that our project can serve as a basis for illustration of the application of blockchain technology. Thus, associated with our implementation, we can also describe possible use cases to extend our project and how the blockchain technology is leveraged for other cases besides digital currency. Second, our prototype can serve as a primer for people to understand why FPGAs are more suited to the task of reducing the intensive computations on the main processor in the application of blockchain technology.

Appendix A: SHA-256 Algorithm

4.1

References

[1] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies.* O'Reilly Media, Inc., 2017.

[2] conradoqg. naivecoin. https://github.com/conradoqg/naivecoin, 2017.

[3] PUB FIPS. 180-4. *Secure hash standard (SHS)*, 2015.

[4] Internet Research Task Force (IRTF). Edwards-curve digital signature algorithm (eddsa). 2017.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

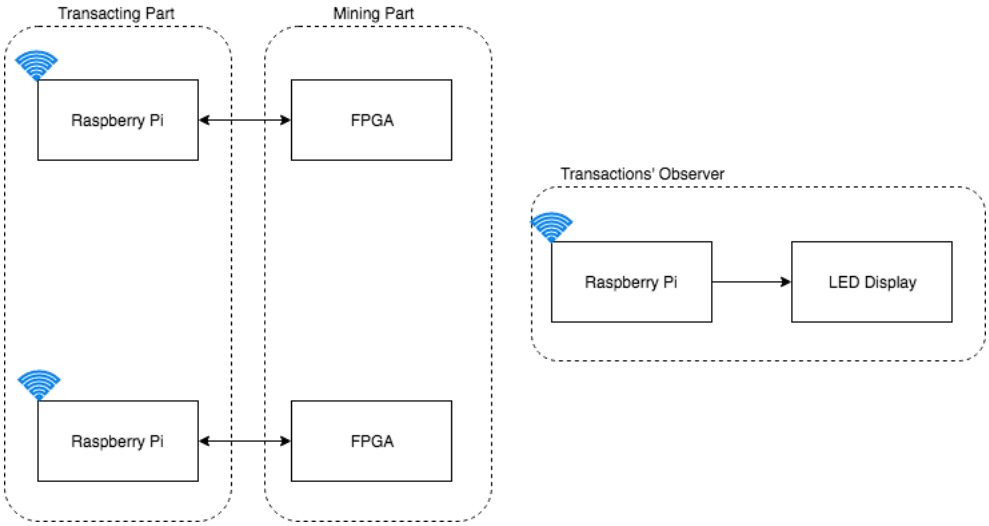

Figure 1: Overview system

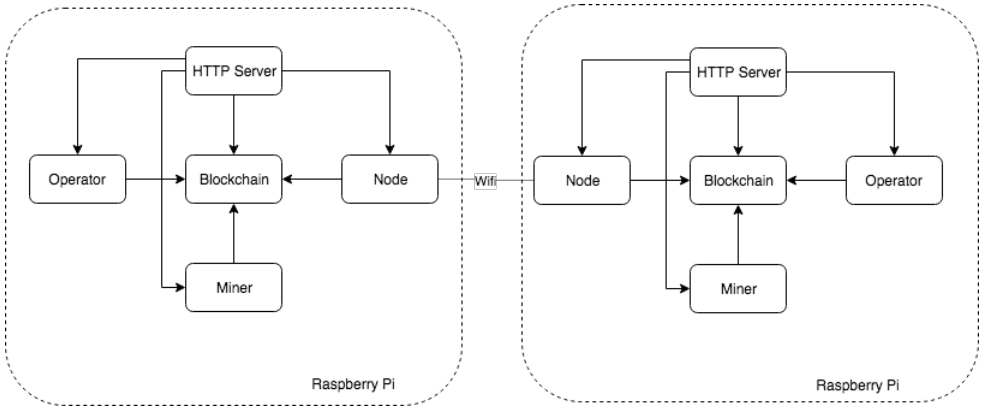

Figure 2: Illustration of connections between 2 Raspberry Pi (Users) based on the 5-component system