Group: Graham Gobieski, Kevin Kwan, Shang Liu, Ziyu Zhu
UNIs: gsg2120, kjk2150, zz2374, sl3881
Embedded System, Spring 2016

Proposal: High Frequency FOREX Trading

High frequency trading (HFT) was made legal in 1999 and has since driven stock market volume and become an important source of liquidity for the market. Initially, high frequency trading firms operated on the time scale of seconds, but as technology has improved, so has the time required to execute a trade; now trading firms must compete on milli- or microsecond time scales. This has led to many firms turning to specialized hardware including graphical processing units and field programmable gate arrays (FPGAs).

In its most simple form, high frequency trading relies on simple arbitrage situations that occur hundreds or thousands of times per second in markets. By recognizing and acting on an arbitrage situation in a fraction of second, a trading firm can make the difference between the purchase price and the price when sold.

In this project, we specifically focus on a particular sort of arbitrage that occurs in foreign exchange currency markets (FOREX). Known as currency arbitrage, this sort of arbitrage arises when a trader converts his or her money to some sort of other currency and after several more conversions converts back to the original currency with a profit. For example, imagine that one purchases one euro at the current USD/Euro exchange rate, then the trader converts that euro to a pound, and finally converts the pound back to dollars. If the conversion rate between the dollar and euro is less than rate between the pound and the dollar, the trader will make up the difference as long as the middle conversion rate does not mitigate the profit.

Such currency arbitrage can be detected via standard graph algorithms. Imagine a weighted directed graph in which the nodes represent different currencies and the directed connections between nodes represent conversion rates. For example the dollar node would be connected to the euro node via an arrow with a weight proportional to the conversion rate between those two currencies. Then to detect arbitrage it is as simple as detecting a cycle that fits a few parameters; mainly that the negated sum of the logs of the weights is negative. The Bellman-Ford algorithm can be used to identify such cycles.

The project has two distinct programming parts: first, we will write a simulator that will send simulated FOREX data to the FPGA and second, we will write an algorithm on the FPGA to identify currency arbitrage cycles. In addition, we will write two device drives – a driver for a VGA display so the we can visualize data, and a driver for a usb device that will take in the raw simulated FOREX data. Please see the image below for further explanation of the architecture of the project.

Group: Graham Gobieski, Kevin Kwan, Shang Liu, Ziyu Zhu
UNIs: gsg2120, kjk2150, zz2374, sl3881
Embedded System, Spring 2016

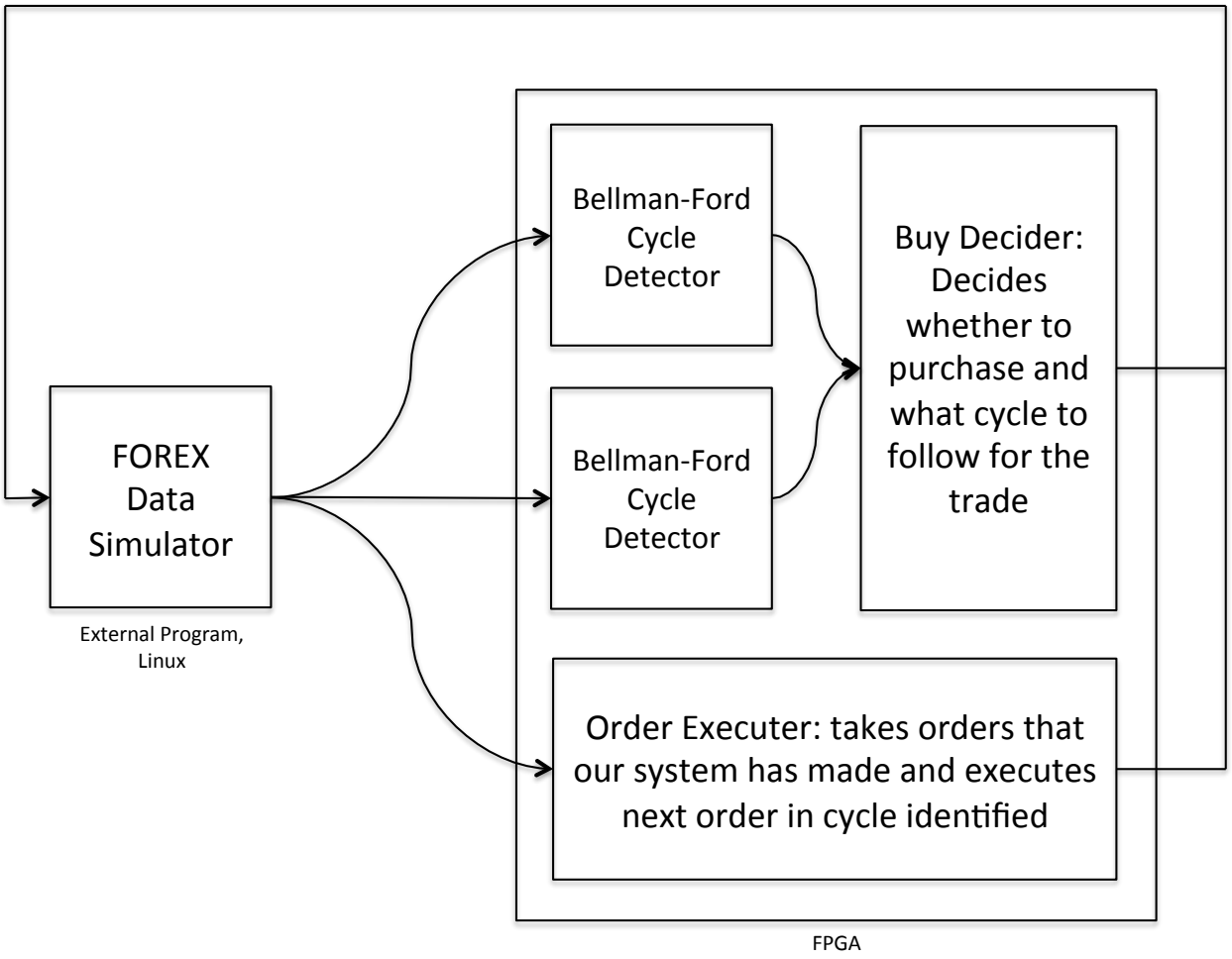

Figure 1: Basic architecture of the High Frequency FOREX Trading Platform proposed.