

## Playing Card Reader

Nick Hougardy - [nickhoug@mit.edu](mailto:nickhoug@mit.edu)

Project Abstract

10/18/2022

Over the summer, I was building a “Smart Poker Table” that had significant hardware problems and time delays, culminating in the project not getting done. In short, this table (would’ve) calculated the odds of each player winning the hand with their two pocket cards in an 8 handed game of Texas Hold’em. Based on some state machine logic, another hardware piece would then capture the flop, turn, and river, which are the remaining 5 cards to come out onto the board that all players use alongside their own two cards. Along each step, the hardware would recalculate the odds given the new information on the board and using the remaining players in the hand. The logic and state machines for this project are solid, but the hardest part I had issues with was getting the hardware I was using to recognize a card and record it, which is where I think the FPGA would soar. Beforehand, I was using a Raspberry Pi 3 along with the camera Pi module and I wasn’t getting anywhere close to the results I needed. That, and the camera output was something along the lines of 7 frames per second.

There are several milestones for this project that I think would make great benchmarks. The logical progression as I see it is the main goal should be getting the FPGA to read a card. By this, I mean inputting a card through the camera module and outputting a string to a Monitor or byte to some of the pins on the board. For instance, if there are 52 cards, I can assign each card a 6-bit number ( $\log_2(52) = 6$ ). So, the Ace of Spades might output a unique value of “101010” to some pins (or the LED bank) or I can just write text to a monitor. Again, if I can recognize a card, I think that is a very reasonable goal for this project. A more impressive goal could be getting the FPGA to recognize multiple cards in the same frame with no cap on the number of cards. So, if 7 cards are in frame, it will recognize 7 cards exactly. This, in my opinion, is much harder, but doable.

A much larger stretch goal, which I almost don’t even want to mention because of the complexity, would be getting the entire table to work. 9 camera modules (1 for all 8 players and 1 for the community cards) all working in sync. I don’t know if this would be 9 FPGAs, or if there is some way to hook up 9 cameras to the same FPGA. Maybe using I2C and giving each camera a specific hex address? At any rate, this would also mean getting the odds calculating component working in addition to the card reader, which I have no idea where to even start with that. There are some mathematical shortcuts that will get you close to the approximate percentage of winning, but to be accurate within 1-2%, doing winning combinations over total combinations would be the only mathematical way to do it, which would require a lot of combinational power. I think if I got to have heads up working, meaning just two players and the community cards, that would be a resounding success and this would be something I’d be extremely proud of.