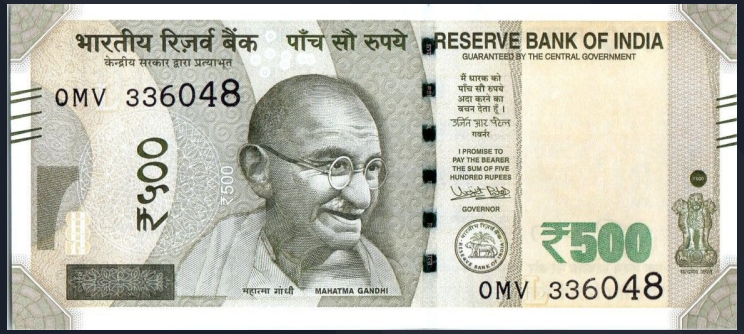
FAKE CURRENCY DETECTOR



DIGITAL SYSTEMS PROJECT

Need of Fake Currency Detector

- Accustomed method of looking through the watermark of the note through the naked eye does not always help to detect the counterfeit money due to methods of advance printing.
- Production of large amount of counterfeit money reduces the value of real money.
- Once in circulation, it becomes difficult to track every fake note until it is deposited in bank.

- RBI Mentions a list of 17 features to look for in the real ₹500 note, and matching every feature will consume a lot of time and resources.
- Matching Every Currency note with their features is an additional load for merchants who trade in cash.
- We wanted to reduce the time of identifying counterfeit currency, that would assist the common people to identify and terminate its circulation.

Image Courtesy

RBI - https://www.rbi.org.in/commonman/English/Scripts/CurrencyNotePosters.aspx#

How we are planning to implement the project

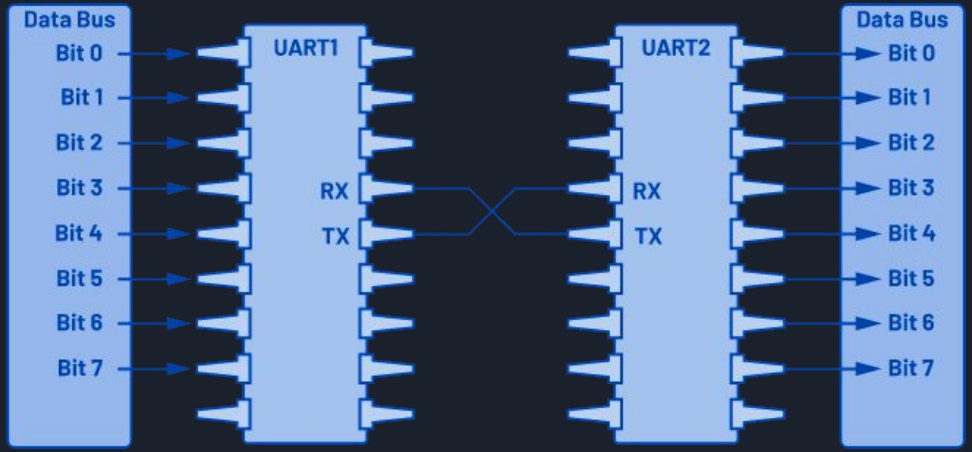
- We decided to compare some important features of real note with the test note. Such features include the security strip, the polygon above the emblem and also the strips at the edges of the note for blind people.
- As we were falling short of time and facing much difficulties we decided to compare only one of the security features as of now, i.e, the security strip. We can extend the same procedure for comparing other features.
- First of all, we will detect the location of the strip by feature extraction in python. Then we will crop out a rectangular part from the image which contains the security strip of the test note.
- We will then convert the image into B and W scale and then in binary form and then compare the intensity of the test and real note security strip through the verilog code.

Group Task Division:

- Aaryan and Pratik were responsible for the code for UART for connecting the FPGA board and the PC.
- Darshi and Vaibhavi were involved in figuring out the actual algorithm for the project. Both also took the inspiration from one of the seniors, Harsh Patel of BTech’18 batch.
- Aaryan and Darshi were also responsible for storing the image files in BRAM.

How UART works?

- UART stands Universal Asynchronous Receiver-Transmitter, is one of the most used device-to-device communication protocols [1] .
- Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end [1] .
- UART works by involving transmitting and receiving serial data. In serial communication, data is transferred bit by bit using a single line or wire [1] .



The two signals of each UART device are named as:

- Transmitter (Tx)
- Receiver (Rx)

Steps of data transfer through UART module [1] :

1. The transmitting UART receives data in parallel from the data bus.
2. The transmitting UART samples the data frame to minimize error in transferring data.
3. The entire packet is sent serially starting from start bit to stop bit from the transmitting UART to the receiving UART. The receiving UART samples the data line at the preconfigured baud rate.
4. The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.

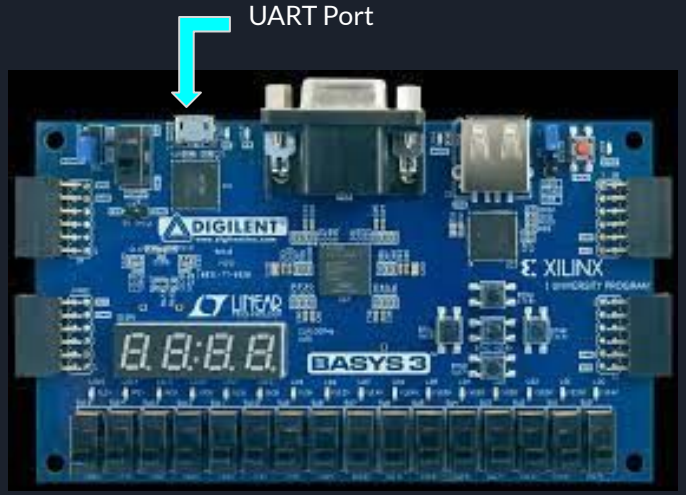


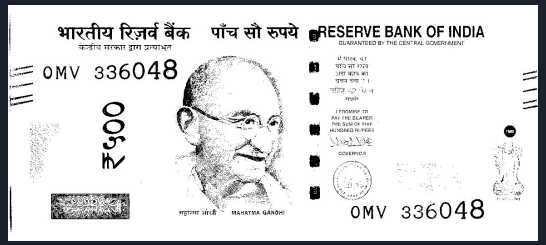
Image Courtesy

Diligent - https://digilent.com/reference/_media/basys3:basys3_rm.pdf

How do we extract the security strip of the note?

- In the initial stage, we just decided to compare the features using the location of the strip in both the notes. Which we assumed to be the same.
- But we discovered that the location of the strip is not the same in each note. So we decided to first extract the location of the stip in the test note.
- We will basically detect the position where the intensity of the image increases suddenly (strip of the image is dark in B and W scale) by comparing and then save it for using it later in the verilog code.
- We cropped out a rectangular part so that we do not need to store the whole note image in binary text.

Security Strip Detection



BW Scale Currency Note

Rect. Region with Sec Strip

Sec Strip Cropped after detecting the location of strip

Binary Text File with 0s and 1s

How does the verilog code works?

- First of all we are storing the binary forms of both (real and test images) in BRAM. Hence, we are creating two BRAMs.
- We will then use the extracted location of the strip of test image and the real strip image location (which is fixed)for comparing the intensities at those locations.
- We also set a threshold difference value, i.e, if the intensity difference is more than the threshold value then the test image is fake otherwise real.
- We are comparing a total of three columns of the security strip.

Final FPGA implementation:

- For the final FPGA implementation we synthesised the code written on Verilog, such that the two images are stored in the block memory of the basys 3 board.
- It has a maximum capacity of 18000 bits which is enough for storing two security strips, of about 11000 bits each
- We process the code on verilog and send the output from the board to the PC through the UART Transmission module.
- The total run time is of about 11000 clock cycles enough to give the output in about a second
- The output is whether the note is real or fake.

Conclusion

We learnt various new concepts in the project:

- How UART works and its use, i.e, how to transmit and receive large number of bits across the PC and the FPGA.
- We also learnt that we the capacity of the FPGA board to store the data is not much so we need to store it in the BRAM.
- BRAM stores multiple values as cluster of arrays and gives output per clock cycle
- How to extract the features from the image based on the intensity difference.
- It is important to keep in mind the duration of the clock cycle as only limited operations can be performed per clock cycle.

References:

1. E. Peňa and M. G. Legaspi, “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,” Analog.com. https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html (accessed 19 Nov. 2022)
2. Image link: https://upload.wikimedia.org/wikipedia/commons/2/2e/India_new_500_INR%2C_MG_series%2C_2016%2C_obverse.jpg
3. Inspiration code: https://github.com/Harshp1802/Fake-Currency-Detector
4.