Image Processing and Conversion

Varsha Athreya, Melissa Regalado,
Punnisa Amornsirikul, Conor Ruane

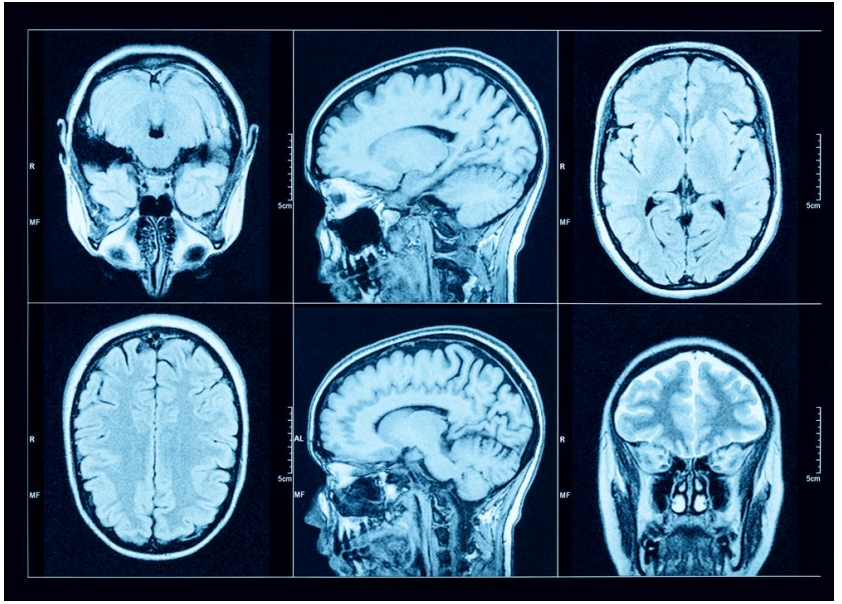
ENG EC 311

Goal

- Convert images using UART on FPGAs
- Transformations controlled by switches:
  - Darken
  - Blur
  - Rotate
  - Invert

Motivation

- Computer vision
- Photo editing
- Medical imaging



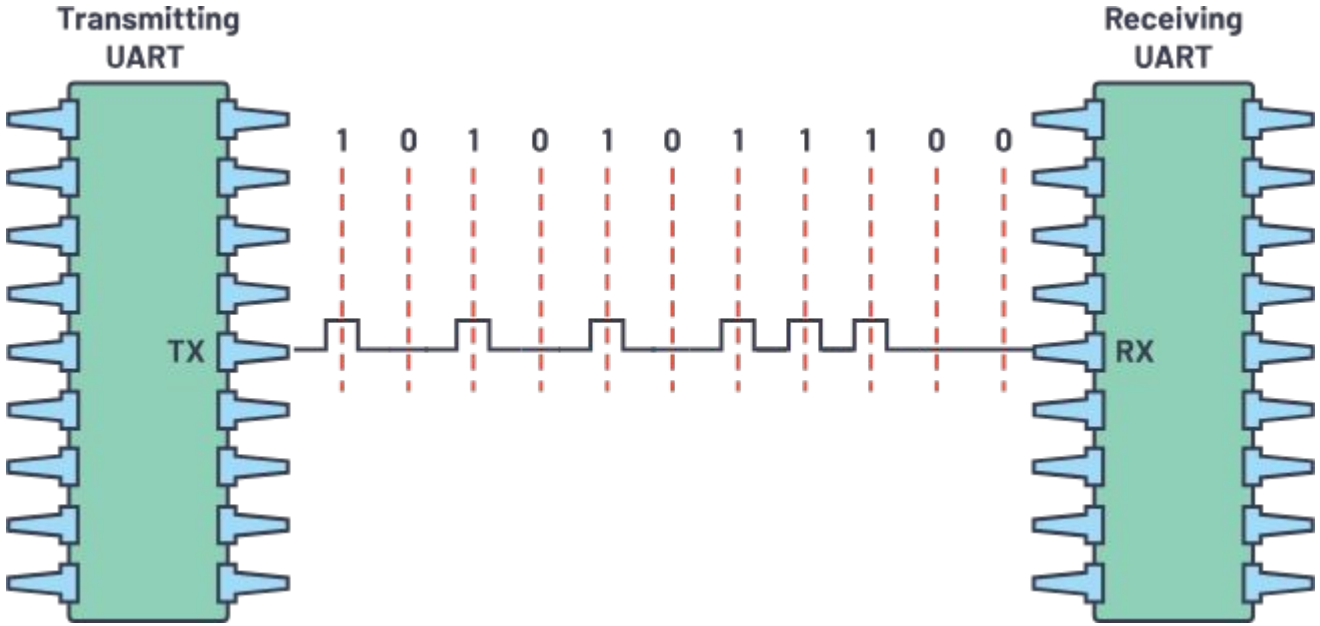
Functionality

- Receive image pixels in UART from MATLAB
- Perform operation in separate module
- Transmit image from UART back to MATLAB
- Display original and transformed image using VGA eventually

Specification

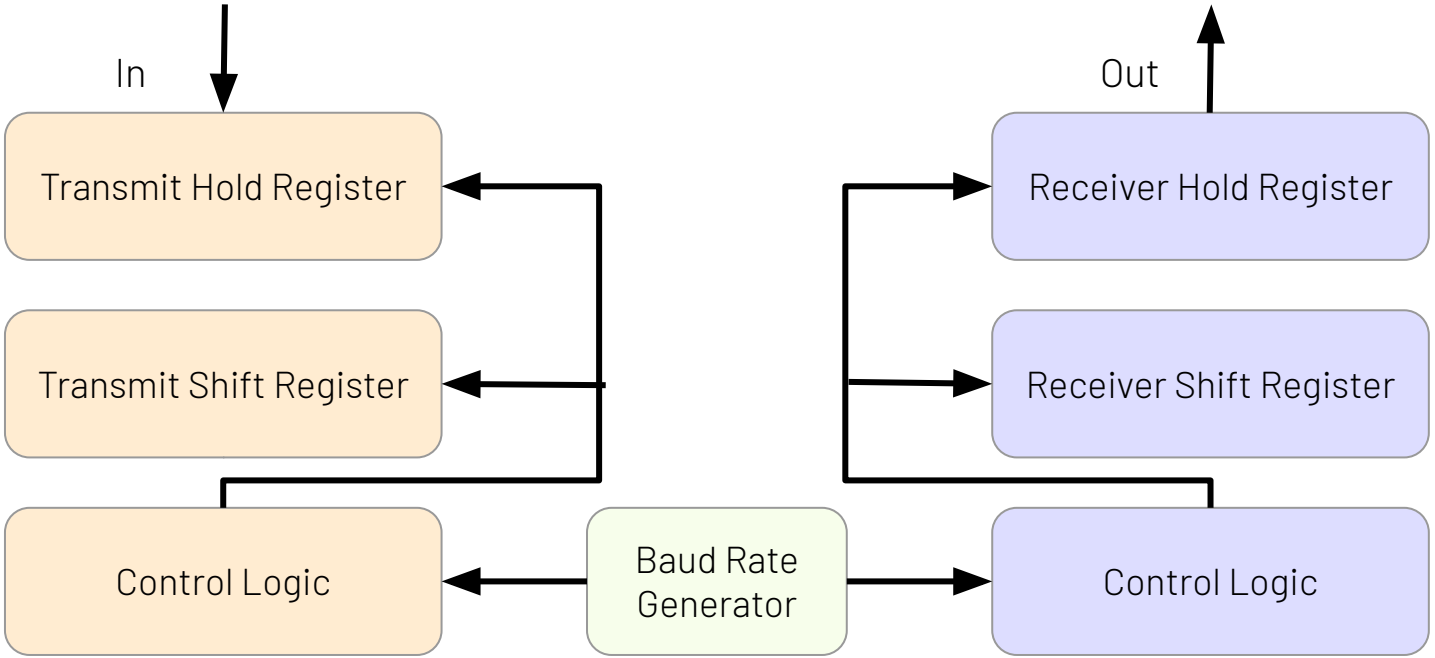
- Receive/transmit 8 bits at a time
- 9600 Baud Rate
- Utilize UART and FPGA
  - (Possibly given time, VGA)

Block Diagram

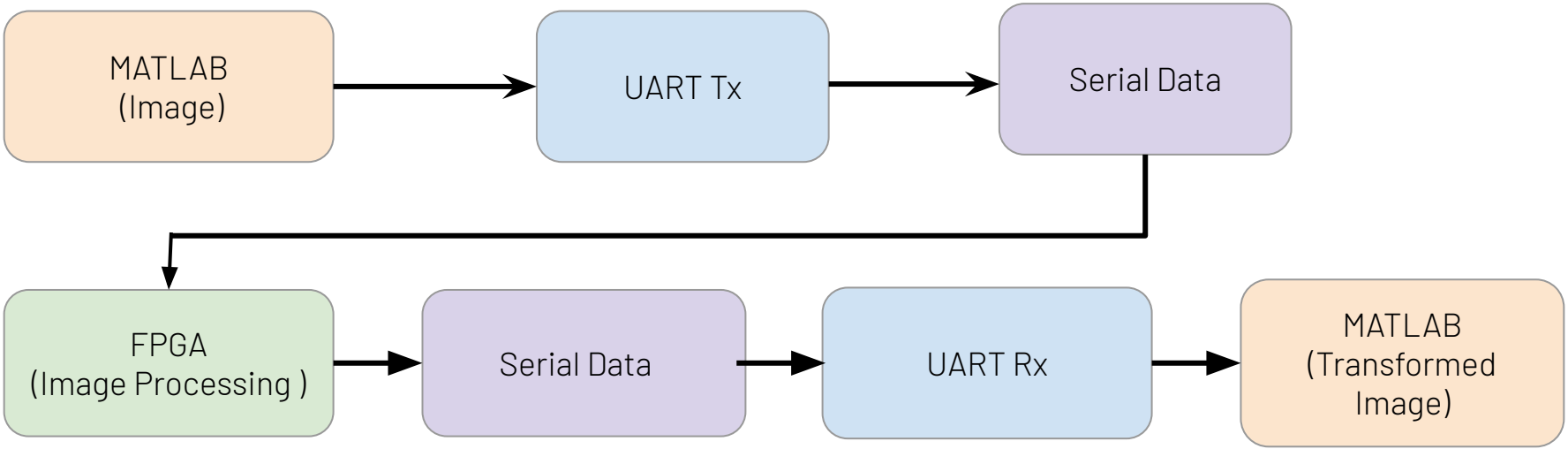


https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html

Block Diagram



Block Diagram

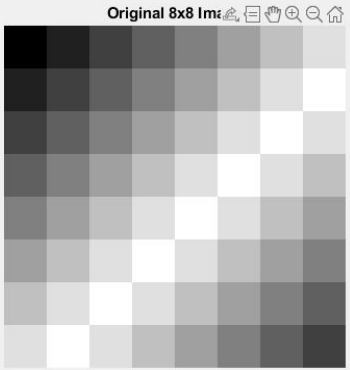


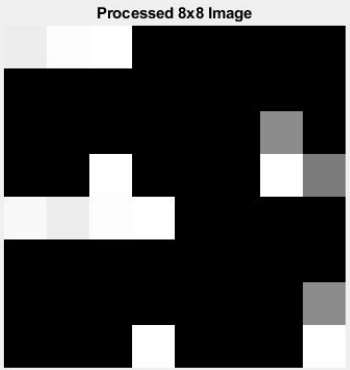
Code Snippet (MATLAB)

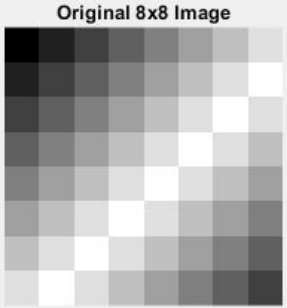
```
clear all;

% UART communication
uart = serialport('COM4', 9600);
uart.Timeout = 30;
flush(uart); % Clear UART buffers
```

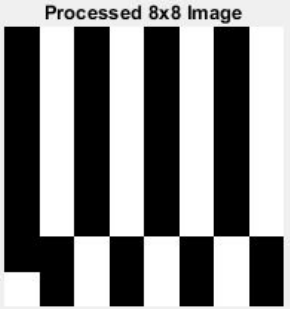
```
% Send the 8x8 image via UART
disp('Sending 8x8 image data...');
write(uart, data, 'uint8'); % Send as uint8

% Receive the processed 8x8 image data
disp('Waiting to receive processed data...');
processed_data = read(uart, numel(data), 'uint8'); % Read the same number of elements
```

Test Examples









Troubleshooting

- Clock Divider was implemented for the Verilog Code
- Timeout was increased
- Buffer size decreased from 1024 to 64 (for an 8 x 8 image)

Code Snippet (Verilog)

```
module UART_RX(
    input clk,
    input rst,
    input rx,
    output reg [7:0] rx_data,
    output reg rx_done
);
```

```
module UART_TX(
    input clk,
    input rst,
    input tx_start,
    input [7:0] tx_data,
    output reg tx,
    output reg tx_busy
);
```

```
// Internal Memory
reg [7:0] image_memory [0:63];
reg [9:0] mem_address;
```

Verilog Snipped

```
case (state)
    IDLE: begin
        mem_address <= 0;
        tx_start <= 0;
        if (rx_done) state <= RECEIVE;
    end
    RECEIVE: begin
        if (rx_done) begin
            image_memory[mem_address] <= rx_data;
            mem_address <= mem_address + 1;
            if (mem_address == 1023) state <= PROCESS;
        end
    end
    PROCESS: begin
        for (i = 0; i < 1024; i = i + 1) begin
           image_memory[i] <= 255 - image_memory[i]; // Example: Invert pixel

        end
        state <= TRANSMIT;
    end
    TRANSMIT: begin
        if (!tx_busy) begin
            tx_data <= image_memory[mem_address];
            tx_start <= 1;
            mem_address <= mem_address + 1;
            if (mem_address == 1023) state <= IDLE;
        end else begin
            tx_start <= 0;
        end
    end
```

Successes

- Designed custom UART to receive and transmit image data from MATLAB
- Transforming (incorrectly) the received image
- Sending image back to MATLAB
- Overall combining MATLAB and UART functionality

Failures

- Trade off between FPGA inclusion and simplicity of implementation
- Missing functionality of multiple conversions
- Clock Divider adjustment for more accurate value
- Optimize MATLab Portion

Questions?

THANK YOU

Your group will be given **10 minutes** to present to the class the following information relating to your project. This material should take the form of PowerPoint (.ppt or .pptx) slides that will be **put in your GitHub repository** by **12:00pm ET** the day of your presentation.

1. *Project title, project members* – this should be a single slide
2. *Goal/Motivation* –what you are doing and why. This slide should include 1 concrete example of how someone could actually use your design in real life.
3. *Short Functionality* – one slide max recapping what you are doing. What was your design supposed to do?
4. *Short Specification* – one slide max recapping the specification of the design. What were the requirements? Constraints?
5. *Detailed Block Diagrams* – provide the real block diagrams for your design.
6. *Code Snippet* – provide 1 or 2 of your best code snippets and discuss. What was unique about it? Challenging? Innovative?
7. *Successes* – discuss how your project was successful and why.
8. *Failures* – discuss how your project did not work out as you planned. Provide examples of what you would do differently.

01

Background

AGENDA

01 BACKGROUND
- About the Team
- Market Trends
- Competitive Landscape
- SWOT Analysis

02 OPPORTUNITY
- Market Size
- Differentiators

03 VISION
- Vision
- Strategy Canvas
- Goals

04 ACTION PLAN
- KPIs/Objectives
- Action Plan
- Timeline
- Revenue

05 CLOSING
- Questions
- Thank You