Professional Portfolio

Yin Fung Khong
M.S. Computer Engineering

CliftonStrengths:

Achiever, Analytical, Responsibility, Competition, Discipline

Professional Summary

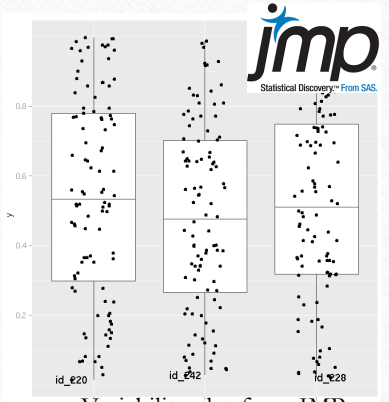
- Education
  - **M.S.** in **Computer Engineering**, *California State University – Northridge (CSUN) 3.94GPA*
- Experience
  - **Software Optimization** Specialist & Project Manager **Lead,** *ISI Language Solutions*
  - **Research Assistant** & **Graduate Assistant,** *California State University - Northridge (CSUN)*
  - **Graduate Intern**, *Intel*
- Skills
  - **Programming:** MATLAB, VBA, Verilog, System Verilog, VHDL, ARM, Java, Python, C, C#
  - **Laboratory Tools:** Oscilloscope, DMM, logic analyzer, waveform generator, soldering iron
  - **Software Tools:** Microsoft Office Suite, JMP, Mondays.com, Notion.so, Xilinx Vivado/HLS, Synopsys, Cadence PSpice

Professional Summary *cont.*

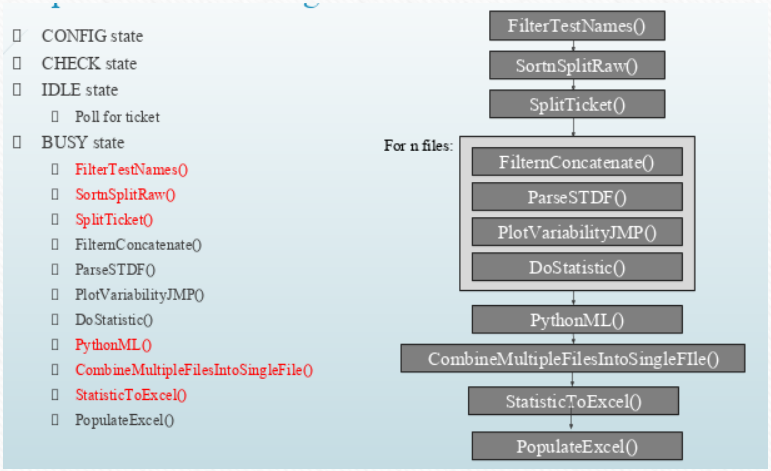
- Research Publications
  - S. Semerjian, Y.F. Khong, S. Mirzaei, **White Blood Cells Classification using Built-In Customizable Trained CNN**, *IEEE International Conference on Systems, Man, and Cybernetics 2020, Toronto, Canada. Apr 2020.*
  - Y.F. Khong and S. Mirzaei, **A Novel Approach for Efficient Implementation of Nucleus Detection and Segmentation Using Correlated Dual Color Space**, *IEEE International Conference on Systems, Man, and Cybernetics, Oct 2019.*
- Courses taken
  - **D**iagnosis and Reliable Design of Digital Systems
  - **D**igital System Design Automation and VHSL Modeling
  - **D**igital Systems Design with Programmable Logic
  - **D**igital Design with Verilog and System Verilog
  - **F**PGA/ASIC Design and Optimization using VHDL
  - **S**ystem on a Chip (SoC) Design
  - **M**icroprocessor Systems

Project Experience

- **Automated Data Analytic Software utilizing C# and JMP**
  - Developed an automation tool in C# which will process the STDF files from an ATE and populate an Excel report for test data analysis.
    - Features including:
      - Decompresses and filters the STDF files according to TIU, Location and Site, LOT name and test name.
      - Plots Variability graphs according to test names by triggering an JMP instance and import the graphs into the Excel report.
      - Calculates parametric values (mean, stdev, LSL, USL ...) of each test name and populate into an Excel report.
  - Reduced manual labor and processing time by **65%**.



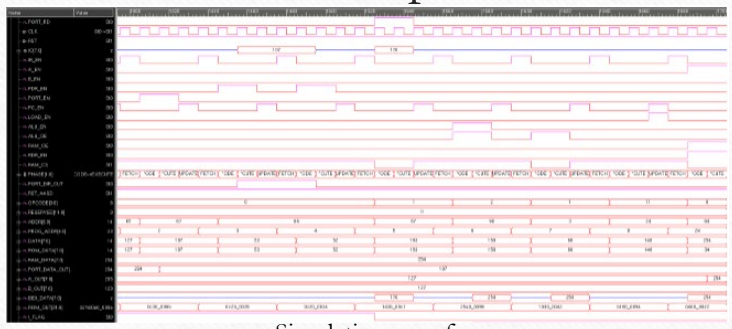
Variability plot from JMP



State diagram of the software tool

Project Experience *cont.*

- **RTL RISC-Y Processor using Verilog and System Verilog**
  - Designed a **fully working and synthesizable RISC-Y Processor** from scratch by creating modules such as Arithmetic Logic Unit, Sequence Counter, MUXs, etc.
  - Created a **testbench to verify the functionalities** of the processor -- proven to be working as expected.
  - Generated a **professional project report** with sufficient snippets of test data and output waveforms as proofs.



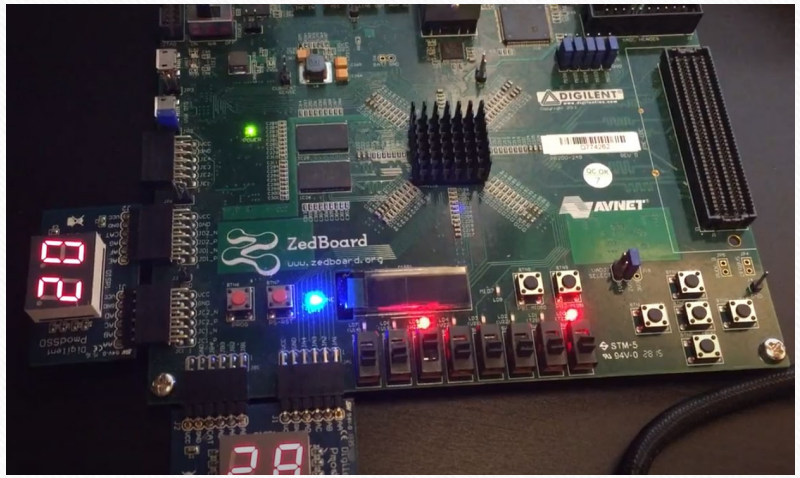
Simulation waveform

```
***Filename: PROCESSOR.sv       Created by: Yin Fung Khong 05-05-2018 ***
*************************************************************************
*** This is top-level module for the RISC-Y PROCESSOR                 ***
*************************************************************************/
`timescale 1 ns / 1 ns

module PROCESSOR(CLK, RST, IO);
      import PKG_PHASE::*;
      parameter OUT_WIDTH = 8;
      parameter ROM_WIDTH = 32;
      parameter DEPTH = 5;
      input CLK, RST;
      inout wire [OUT_WIDTH-1:0] IO;

      wire IR_EN, A_EN, B_EN, PDR_EN, PORT_EN, PORT_RD, PC_EN, LOAD_EN,
ALU_EN, ALU_OE, RAM_OE, RDR_EN, RAM_CS;
      wire CYCLES_PHASE;
      wire PORT_DIR_OUT, RST_AASD;
      wire [3:0] OPCODE;
      wire [11:0] RESERVED;
      wire [6:0] ADDR;
      wire [DEPTH-1:0] PROG_ADDR;
      wire [OUT_WIDTH-1:0] DATA, ROM_DATA, RAM_DATA, PORT_DATA_OUT, A_OUT,
B_OUT;
      wire [OUT_WIDTH-1:0] BIDI_DATA, ALU_DATA;
      wire [ROM_WIDTH-1:0] ROM_OUT;
      wire I_FLAG, OF, SF, ZF, CF;

      //    Sequence controller subsystem
      AASD  MSTR_RST(.RST_AASD(RST_AASD), .CLK(CLK),.RST_IN(RST));
      PHASER      PHASE_GEN(.PHASE(PHASE), .CLK(CLK),   .RST(RST_AASD),
.EN(1'b0));
      SEQ_CTRL CTRL_UNIT(.IR_EN(IR_EN), .A_EN(A_EN), .B_EN(B_EN),
.PDR_EN(PDR_EN), .PORT_EN(PORT_EN), .PORT_RD(PORT_RD), .PC_EN(PC_EN),
.PC_LOAD(LOAD_EN), .ALU_EN(ALU_EN), .ALU_OE(ALU_OE),      .RAM_OE(RAM_OE),
.RDR_EN(RDR_EN), .RAM_CS(RAM_CS), .ADDR(ADDR), .OPCODE(OPCODE),
.PHASE(PHASE),     .OF(OF),   .SF(SF),   .ZF(ZF),   .CF(CF),   .I_FLAG(I_FLAG));
```

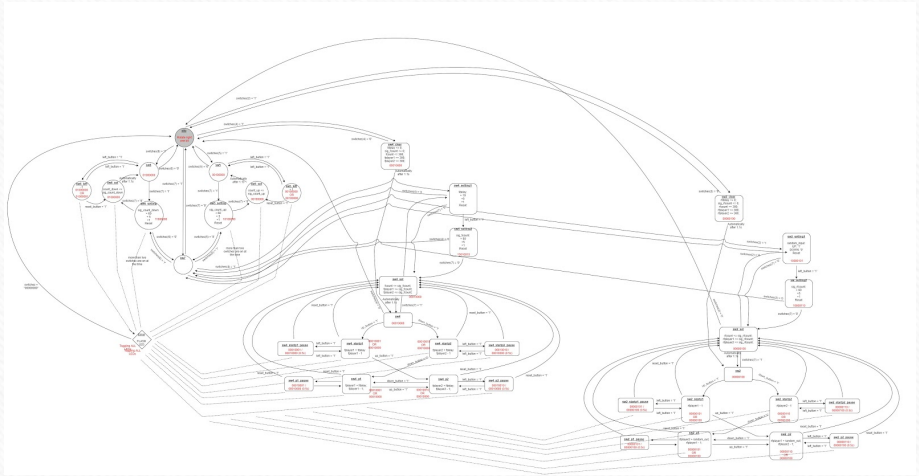
Snippet of the RISC-Y Processor Verilog code

Project Experience *cont.*

- **RTL FPGA Multi Clock and Timers using Zedboard SoC**
  - Designed and implemented multiple chess clocks and timers targeting FPGA in VHDL.
  - Proven knowledge and application of **Finite State Machine (FSM) and Linear-Feedback Shift Register (LFSR)** in hardware design.
  - **Verified the functionalities** of the design by creating testbenches and simulation.

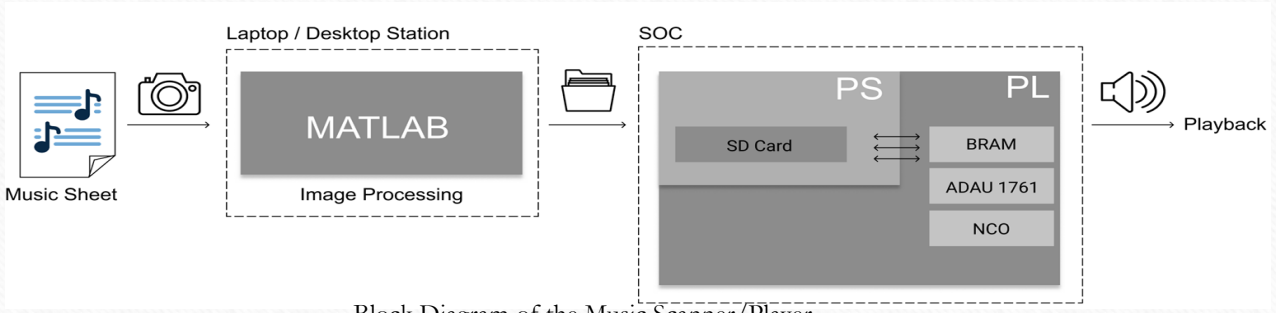


Zedboard running the chess clock timer.

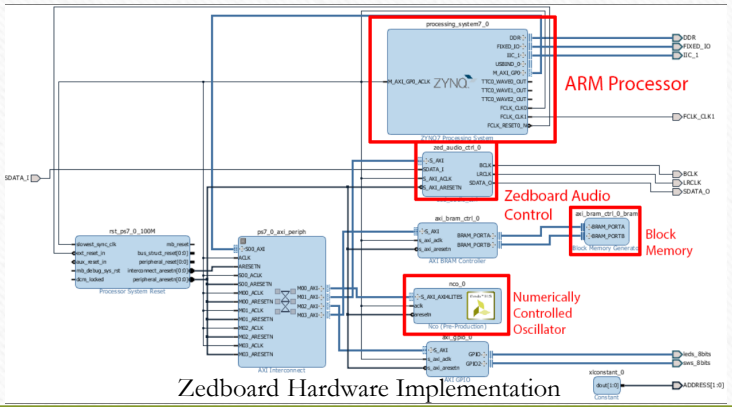

Complete FSM diagram

Project Experience *cont.*

- **Music Scanner-Player**
  - Implemented **Optical Music Recognition system** utilizing MATLAB, streamlined image capturing processing with a DSLR camera, extract the musical information from the musical notes and store into Xilinx SoC Zedboard.
  - Reproduced the melody of the music via ADAU1761 utilizing I2C communication protocol and numerically controlled oscillator.
  - Awarded **second-runner up project** in Senior Design Showcase and the **Best Overall Presentation Award**.



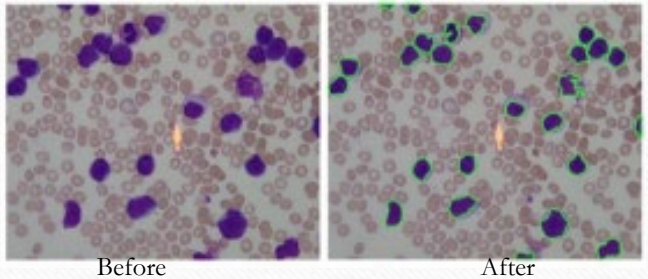
Block Diagram of the Music Scanner/Player



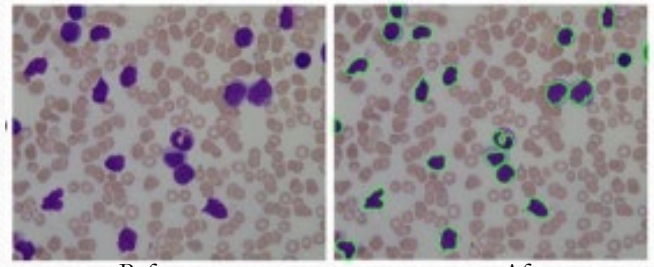
Zedboard Hardware Implementation

Project Experience *cont.*

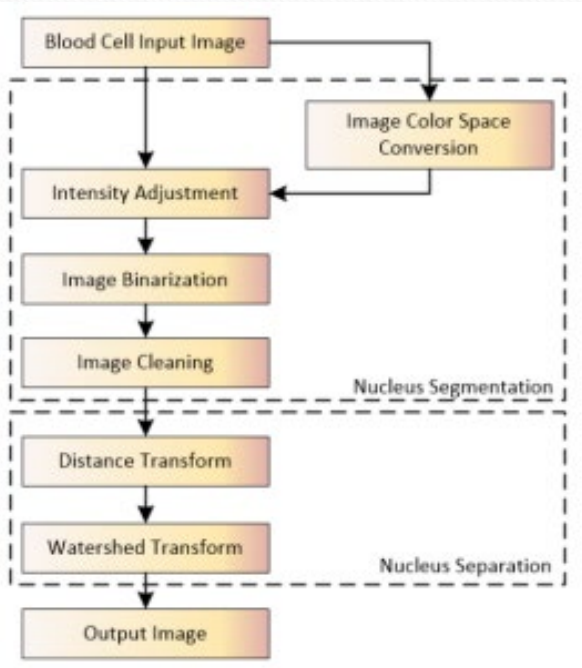
- **A Novel Approach for Efficient Implementation of Nucleus Detection and Segmentation Using Correlated Dual Color Space**
  - Introduced a more **efficient and accurate algorithm** in segment white blood cells from the image background.
  - Achieves a **98.99% accuracy** in segmenting the blood cells while maintaining virtually congruent to the original image.
  - **Constant processing performance** for the same image resolutio regardless of the number or shape of the white blood cells.



Before After



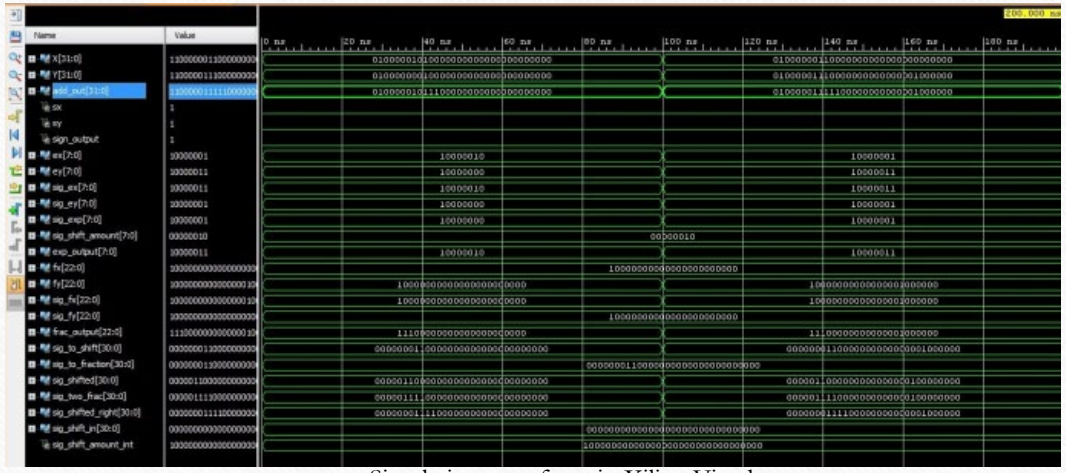
Before After



Algorithm's flowchart for WBC segmentation

Project Experience *cont.*

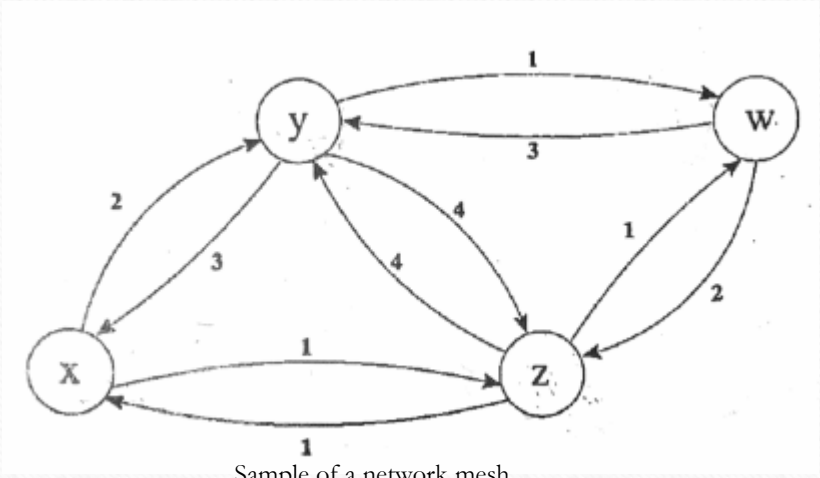
- **32-Bit Binary Floating Point Adder using IEEE 754 Single Precision Format**
  - **Converted a high level logic diagram of a floating point single precision adder** based on IEEE 754 floating point standard into VHDL code in Xilinx Vivado.
  - The adders takes two 32 bits numbers and produce an output with format as below:
    - [**Sign** *(1bit)*][**Exponent** *(8bits)*][**Mantissa** *(23bits)*]
  - Produces an output of 'U' or undefined error handling if the difference between the two exponents exceeds a 23-bit shift in the mantissa.



Simulation waveform in Xilinx Vivado

Project Experience *cont.*

- **Distance Vector Routing in a Remote Messenger App using Java**
  - Implemented a chat messenger that uses Distance Vector Routing Protocol to determine the best route between nodes in the network to transfer the data between hosts.
  - The project was implemented using Java OOP in Visual Studio.



Sample of a network mesh

```java
public class PA2Main {
    static int time;
    public static List<SocketChannel> openChannels = new ArrayList<>();
    public static Selector read;
    public static Selector write;
    static String myIP = "";
    static int myID = Integer.MIN_VALUE+2;
    public static Node myNode = null;
    public static List<Node> nodes = new ArrayList<Node>();
    public static List<String> routingTableMessage = new ArrayList<String>();
    public static Map<Node,Integer> routingTable = new HashMap<Node,Integer>();
    public static Set<Node> neighbors = new HashSet<Node>();
    public static int numberOfPacketsReceived = 0;
    public static Map<Node,Node> nextHop = new HashMap<Node,Node>();
    public static void main(String[] args) throws IOException{

        read = Selector.open();
        write = Selector.open();
        Server server = new Server(4444);
        server.start();
        Client client = new Client();
        client.start();
        myIP = getMyLanIP();

        Timer timer = new Timer();
        Scanner in = new Scanner(System.in);
        boolean run = true;
        boolean serverCommandInput = false;
        while(run) {
            System.out.println("\n");
            System.out.println("User Menu:");
```

Snippet of the Remote Messenger App in Java

THANK YOU!

Look forward to hearing from you! Let's connect!

Cell: (206) 434-2327

Email: yinfung96@gmail.com

LinkedIn: https://www.linkedin.com/in/yinfungkhong/