FPGA COLOR DETECTION

Dr. Dave's Design Dudes
Jonathan Georgino and Herbert Brittner
December 11, 2008

Objective:

Design a circuit with the ability to accurately determine the color of a test sample, and provide both audio and visual output.

Motivation:

- Industrial Sorting
- Product Quality Control
- Robotics / Automation
- Visually Impaired

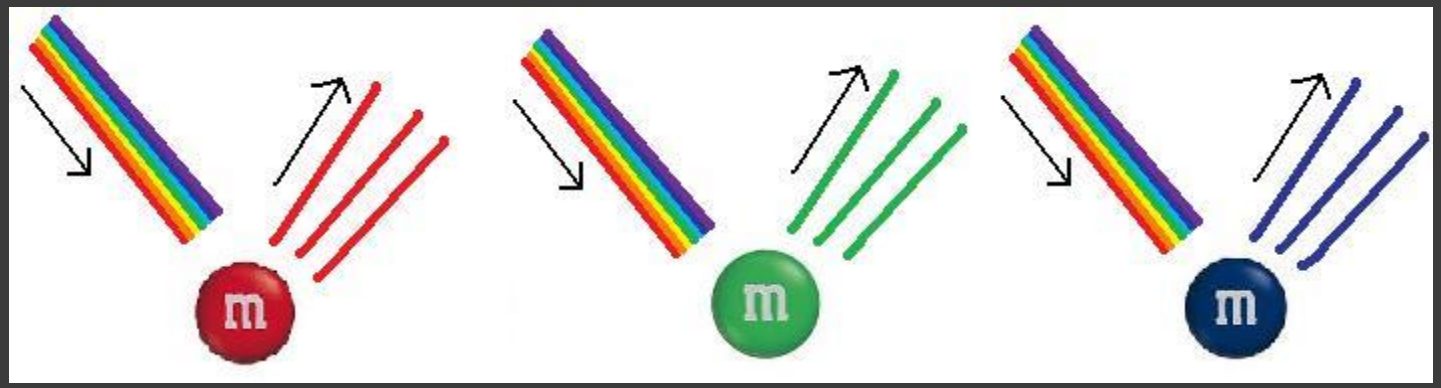
Experiment:

Use VHDL and a Xilinx Spartan 3E FPGA board to determine the color of M&Ms:

- Red
- Orange
- Yellow
- Green
- Blue
- Brown

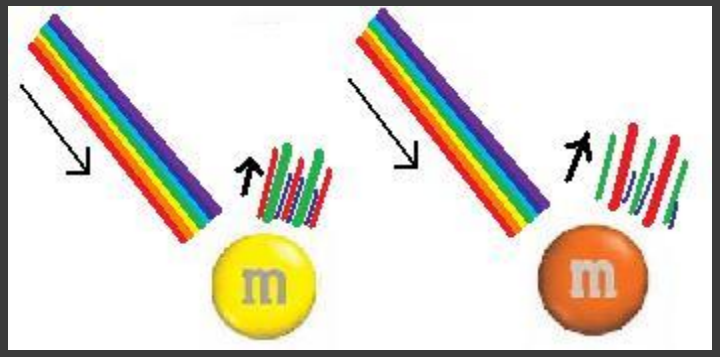
Research:

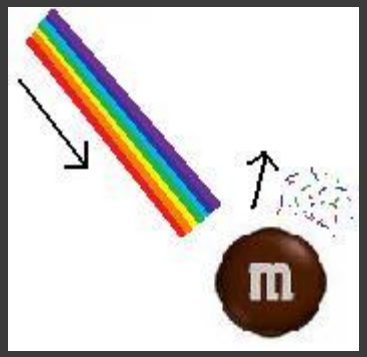
- Sensation of color is based on wavelength of reflected light
- White light consists of light of all wavelengths



Research: (cont.)

- Primary Colors (Red, Green, Blue) can be combined to form all other colors
- Black is the absence of light





Sensor Selection:

- Original Design

Output: Voltage Differential

Advantage: Cost ($5.00)

Disadvantages: Design time
Poor accuracy
Large size

- Pre-made IC

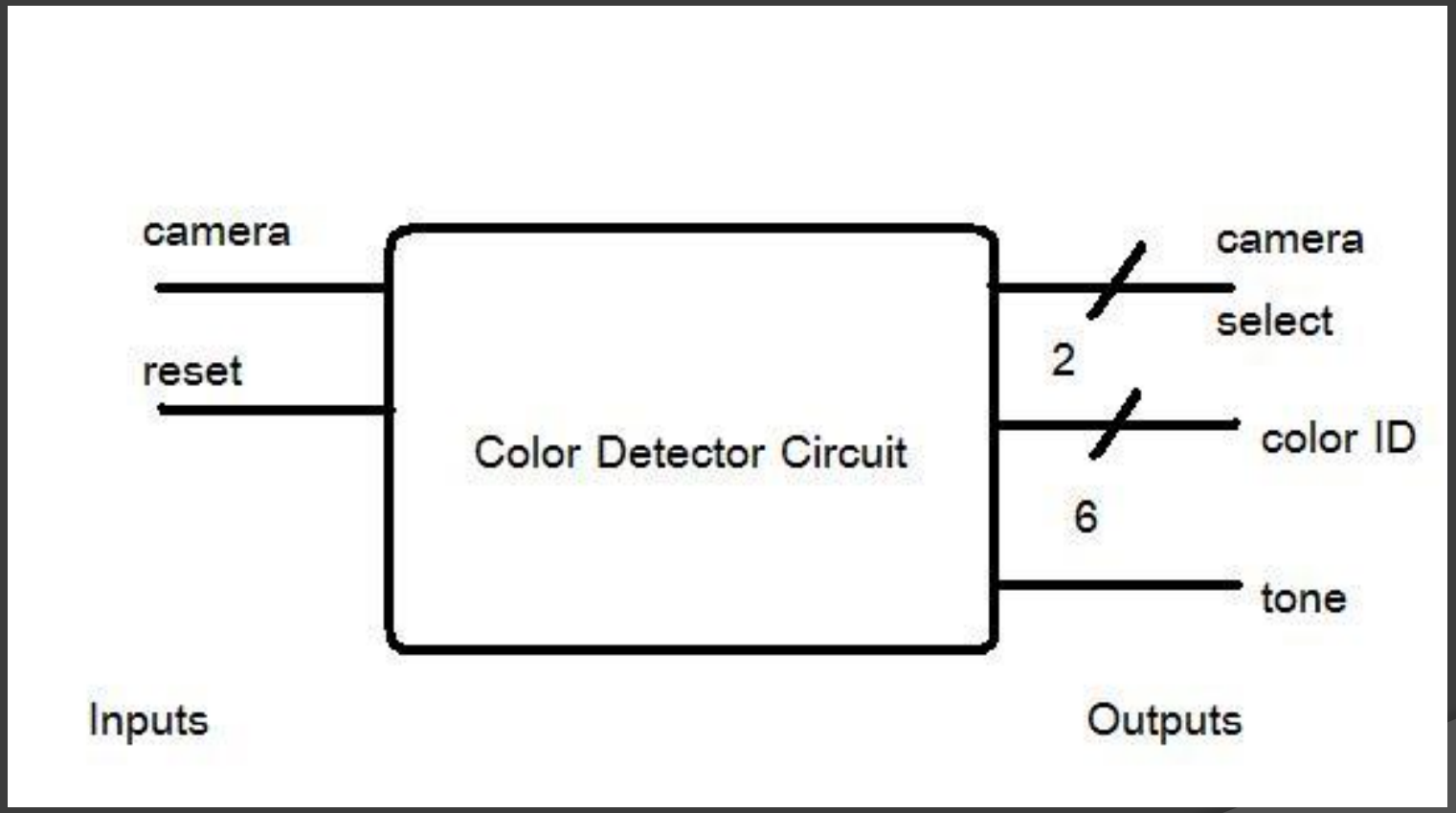
Output: Square wave signal

Advantages: Small size
High accuracy
No design time

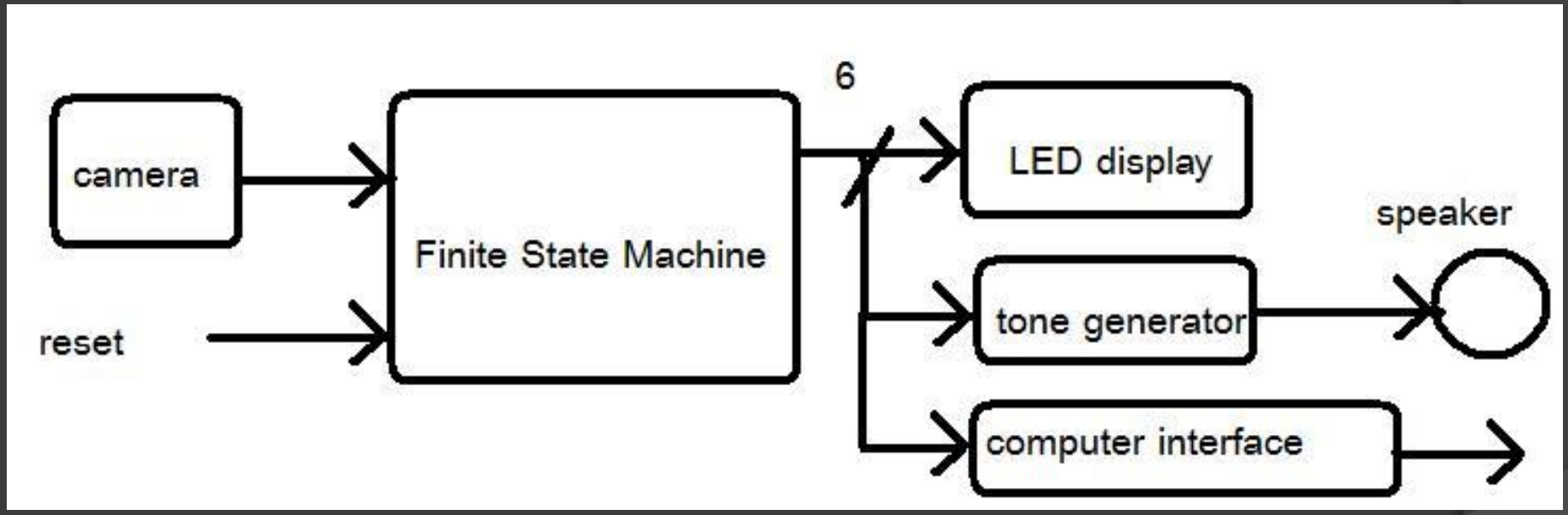
Disadvantages: Cost ($35.00)

Our Choice!

Level-0 Design

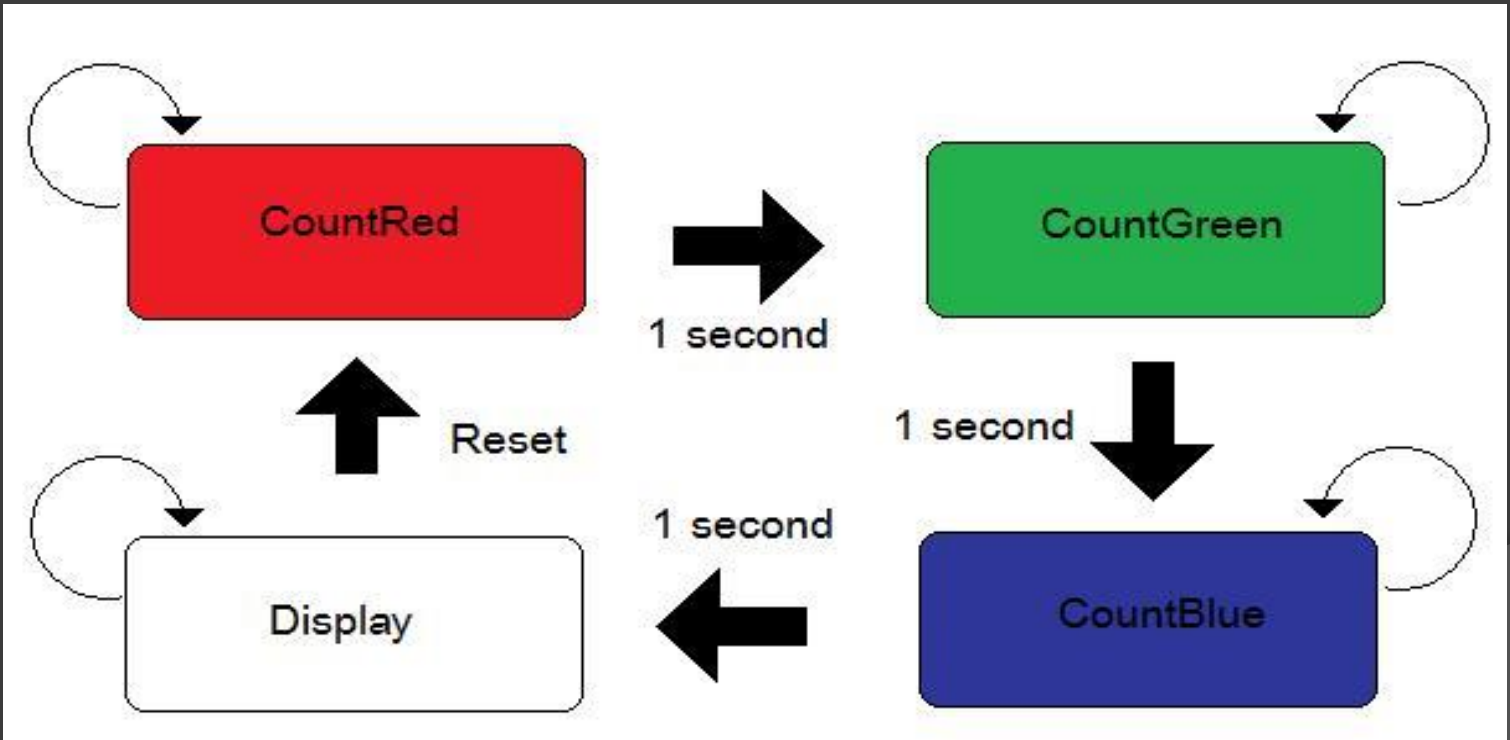


Level-1 Design



Finite State Machine:

- Data collected in states CountRed, CountGreen, CountBlue
- Data analyzed in state Display



Data Interpretation:

- Output from sensor is a square wave with a frequency corresponding to light intensity
- Measured using Oscilloscope in ideal conditions

	Red Filter	Green Filter	Blue Filter
Red	4.0 KHz	1.3 KHz	1.2 KHz
Orange	6.6 KHz	2.7 KHz	1.5 KHz
Yellow	7.6 KHz	12.5 KHz	4.0 KHz
Green	1.1 KHz	5.9 KHz	2.7 KHz
Blue	0.9 KHz	3.4 KHz	12.5 KHz
Brown	0.9 KHz	1.1 KHz	1.1 KHz

Data Interpretation: (cont.)

- Lighting conditions may not always be ideal
- Compare values relative to each other, not a reference table
- Easy for primary colors, difficult for secondary

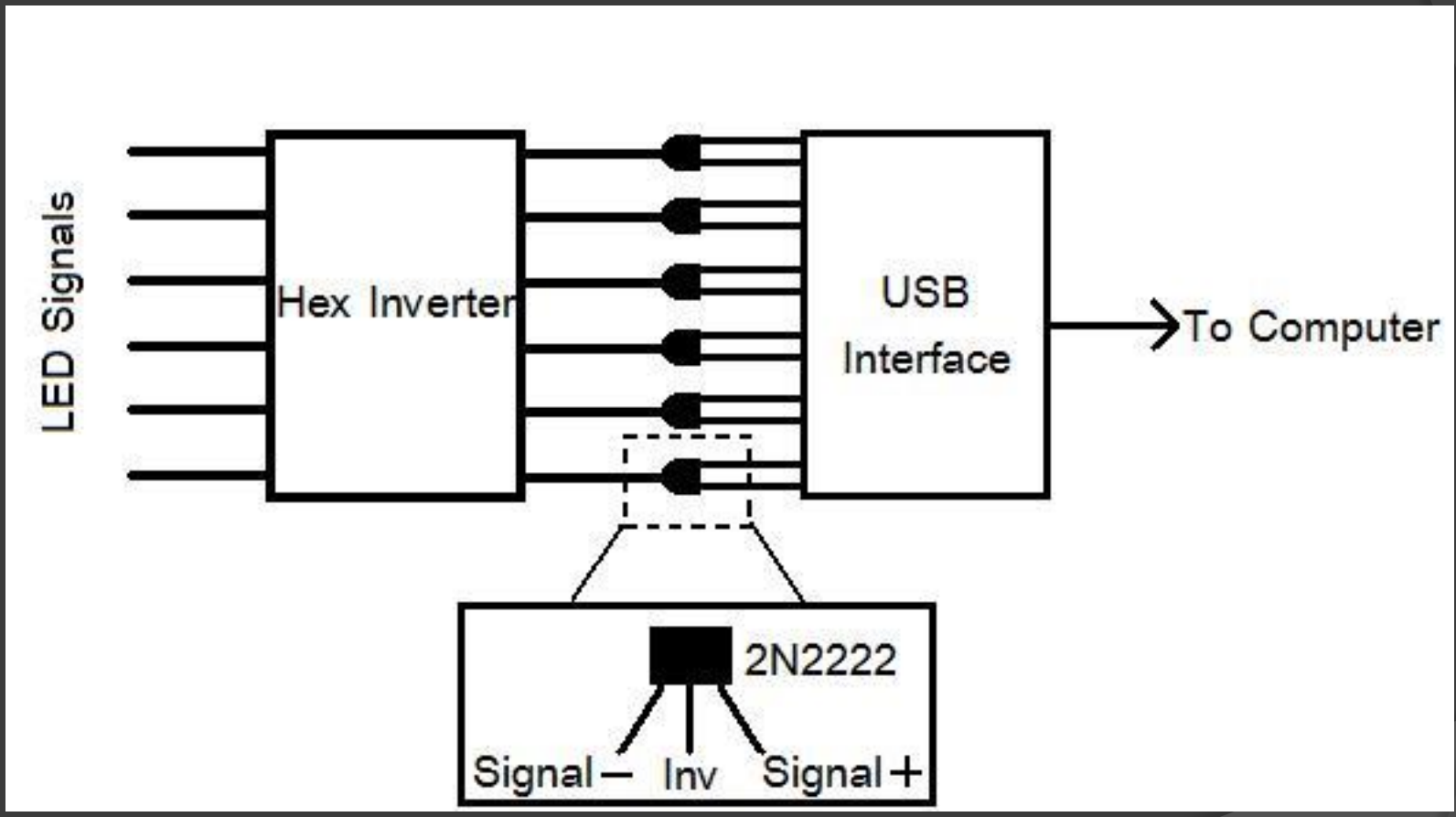
Ex: if(green > red and green > blue and blue>red) then Green

Computer Interface:

- Keeps statistics
- Voice identifies each color
- Easily expandable to fulfill any additional data requirements

- Not part of original design (added 3 days ago)
- Built as rapid prototype
- Software written in Visual Basic

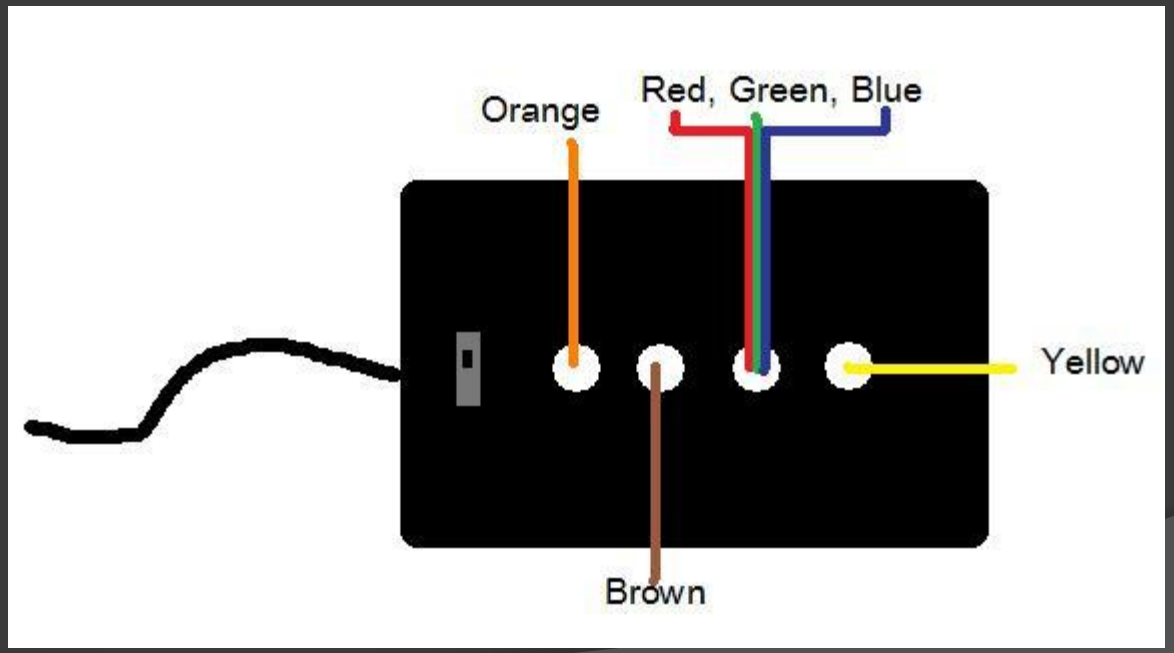
Computer Interface: (cont.)



Hex Inverter necessary due to LED Signal's implementation of inverted logic.

Output:

- Inverted logic to turn on LEDs
- Easily connect output to sorting mechanism

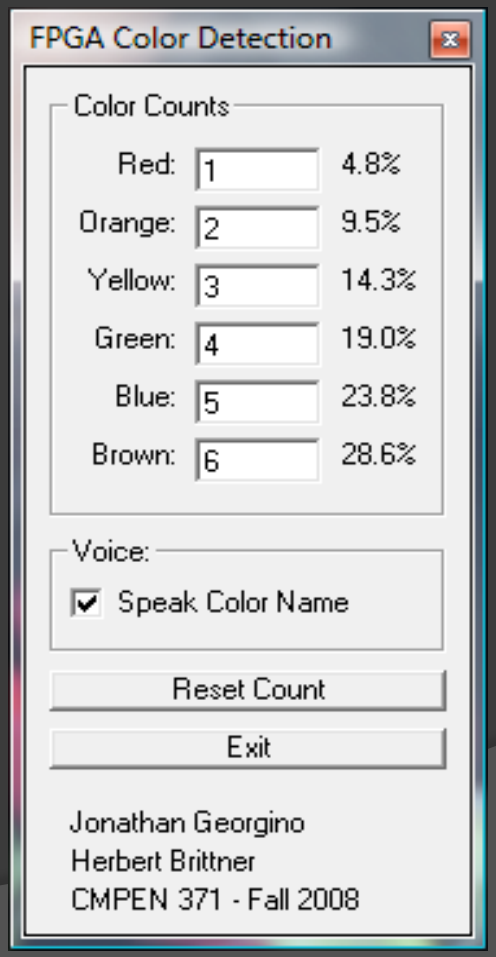


Output: (cont.)

- Speaker

Color	Frequency
Red	6.0 KHz
Orange	5.0 KHz
Yellow	4.0 KHz
Green	3.0 KHz
Blue	2.0 KHz
Brown	1.0 KHz

- Computer



End Result:

- Fully functional identification of six colors
- LED display
- Audio tone output for each color
- Voice announces each color through Computer Interface
- Statistics done through Computer Interface
- Accuracy rate of 100%
- Easy to interface

Improvements:

- Decrease sample time
- Make orange and brown LEDs for display
- Crimp cables to connectors for easier interfacing
- Computer software could be improved