Spring, 2013

Luminous Chessboard

ECE445 – DESIGN REVIEW

TA: MUSTAFA MUKADAM
QIANLIANG LIU, KE MA

UNIVERSITY OF ILLINOIS AT URBANA - CHAMPAIGN | ECE

Table of Contents

1. Introduction 3
    1.1 Title 3
    1.2 Objective 3
        1.2.1 Goal 3
        1.2.2 Functions 3
        1.2.3 Benefits 3
        1.2.4 Features 4
2. Design 5
    2.1 Block Diagram 5
    2.2 Block Description 6
        2.2.1 Overall Summary 6
        2.2.2 Detection Module 7
        2.2.3 Lighting Module 10
        2.2.4 Micro-controller 16
    2.3 Circuit Layout 18
        2.3.1 Micro-controller Circuit Layout 18
        2.3.2 Lighting Module Layout 19
        2.3.3 Detection Module Layout 20
3. Calculations 21
4. Requirements and Verification .... **Error! Bookmark not defined.**
    4.1 Requirements and Verifications 22
    4.2 Tolerance Analysis 27
    4.3 Safety 27
    4.4 Ethics Issues 28
5. Cost and Schedule 30
    5.1 Cost 30
        5.1.1 Labor 30
        5.1.2 Parts 30
        5.1.3 Total cost 30
    5.2 Schedule: 31
6. Reference 32

1. Introduction

1.1 Title

Chess is a popular board game with a long history for which the chessboard and pieces are usually made of wood or plastics. These years with the fast development of the techniques of mobile computer, chess is often played on a touch screen without real chessman pieces. However, we believe the feeling of moving a tangible chessman piece is irreplaceable. So that we are willing to make efforts to combine the benefits of the two by designing a chessboard computer with real board and pieces which can "see" what is happening and can give computed feedback.

There also is a positive expectation of the market because according to our research, none of the existing chess computer actually is able to recognize the pieces. Most of them achieve this function by memorizing the pieces' starting positions and tracing them. This process is very likely to be interrupted by a manual mistake.

1.2 Objective

1.2.1 Goal

The goal of our project is to design an electronic chessboard that can recognize chessman pieces and is equipped with a lighting system showing potential moves with an AI algorithm.

1.2.2 Functions

Once a piece is picked up, the chessboard should light up the available positions it can go, and indicate good moves and bad moves with different colors of LEDs, and afterwards detect the move the player conducts. The AI algorithm will be simple and be able to look one or two steps ahead.

1.2.3 Benefits

- ❖ Direct instructions for rookies
- ❖ Extra fun

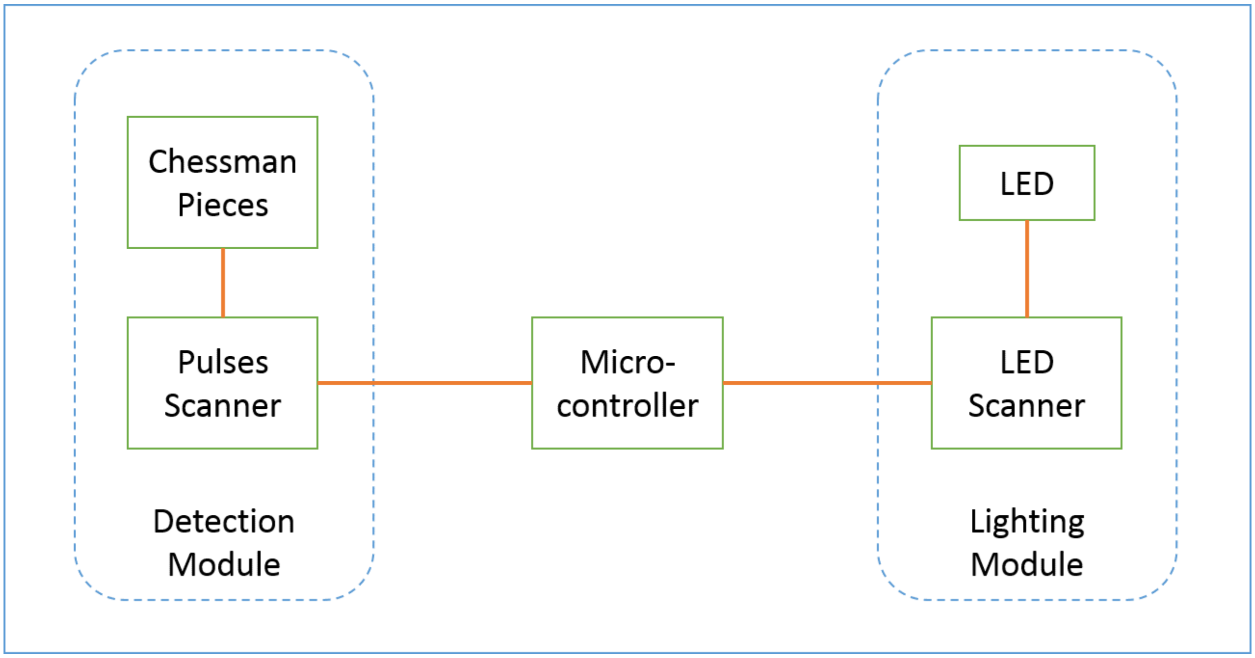
1.2.4 Features

- Recognize pieces
- Warnings for bad moves.
- Glowing.

2. Design

2.1 Block Diagram

The project are made up of 3 parts, the detection module, the lighting module, and the micro-



controller.

FIGURE 1. TOP LEVEL BLOCK DIAGRAM

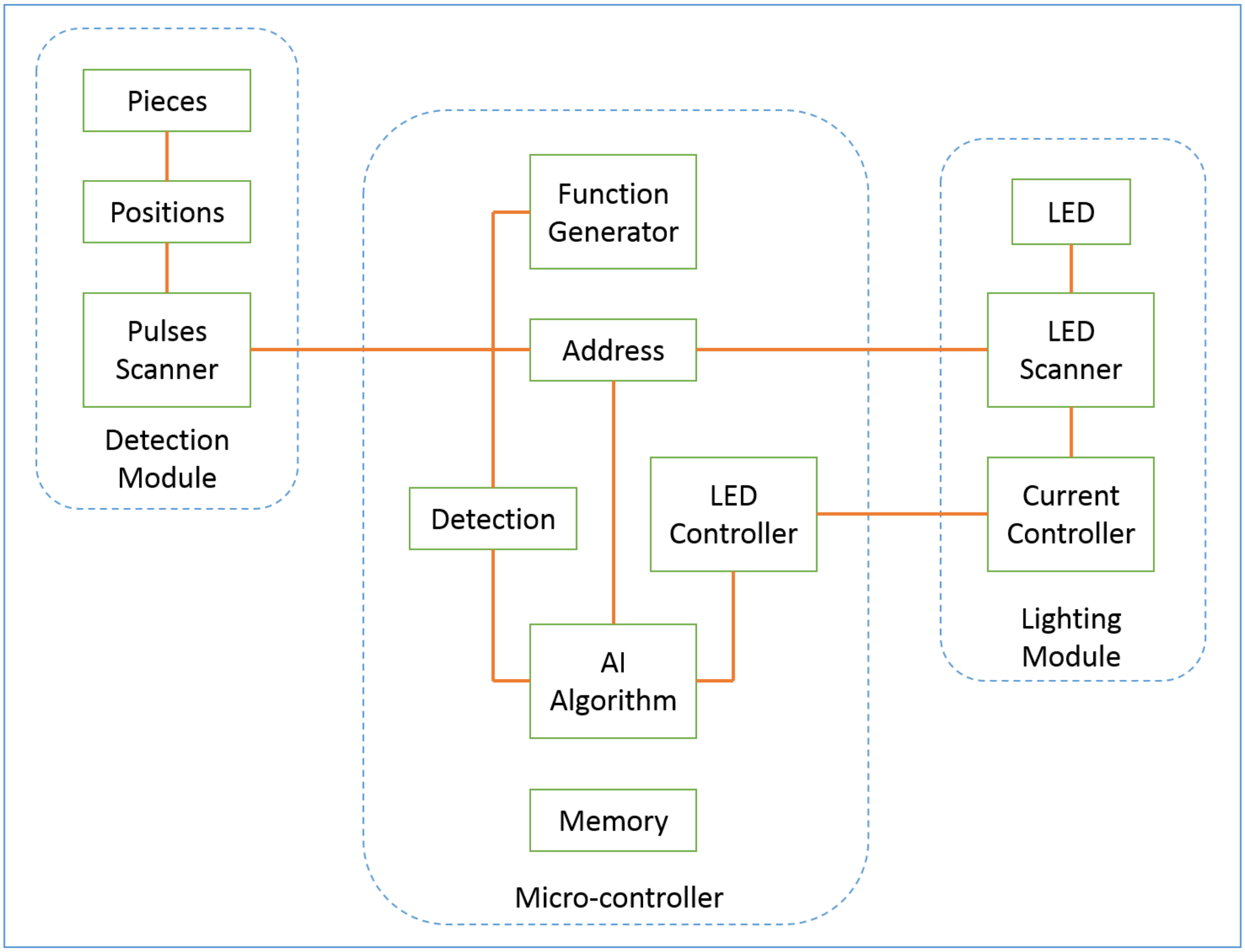


FIGURE 2. DETAILED BLOCK DIAGRAM

2.2 Block Description

2.2.1 Overall Summary

1) Detection Module

This module connects the positions with or without the pieces on to the processor. The pieces are plugged on the chessboard as black boxes.

a. Chessman Piece

A chessman has a circuit which may consist of wires and diodes that can transfer the input pulses waves to the output port to be detected.

b. Pulses Scanner

Since we cannot receive all the feedback pulses at the same time, MUXs are used to scan through the position.

2) Lighting Module

This model light up the chessboard to indicate good and bad moves which is the main function of the design.

a. LEDs

One of the main feature of the chessboard is to indicate good and bad moves with lighting system. The feature may be implemented by installing 64 LEDs under the squares of the chessboard.

b. D/A

The micro-controller output digital signals, while the LEDs are basically operated with currents. This model converts binary control bits to analog currents that turn on or off the lights.

3) Micro-controller

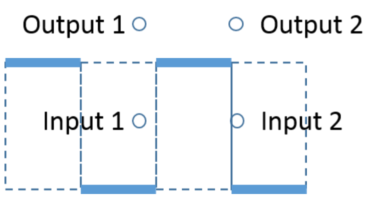
The head of the design. It is in charge of recognizing what is happening on each square with received pulses, and choosing which square is lighting up with what color. The AI will be programmed with C language.

2.2.2 Detection Module

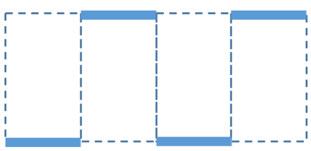
	Inputs	Ouputs
Pieces	A pulse and an inverse pulse	Identifier pulses
Positions	A pulse and an inverse pulse	Piece pulses or 0
Pulse Scanner	Pulses from all the squares; 4-bit scanning address	Pulses from the scanned square

On every position of the chessboard, we put a 4-pin socket on it. Two of the pins are connected to the micro-controller that generates two pulse waves including an inverse one to all the positions. The other two pins are also connected to the micro-controller as the chessboard outputs. Originally, there is no wire connections among the four pins, so the basic outputs of an empty position are always zero, namely, no pulses back.

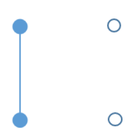
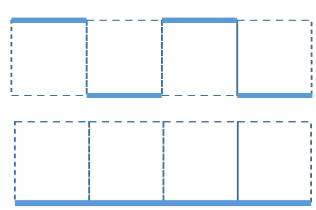
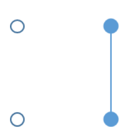
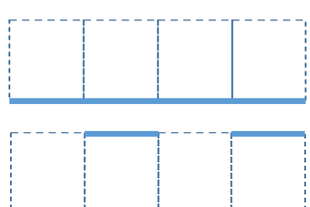
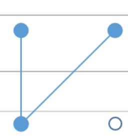
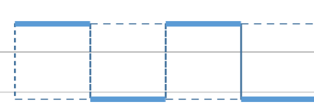
As discussed above, a chessman piece has the same 4-pin socket inside it so that it can be plugged onto the board squares. Inside the chessman, we build a circuit which may consist of wires and diodes that can transfer the input pulses waves to the output port to be detected. Different circuits indicate different types of piece. These pieces act as “black boxes” for the board to recognize.

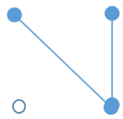
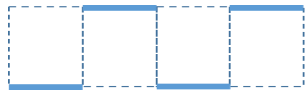
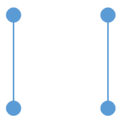
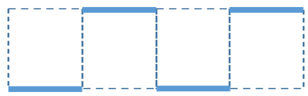
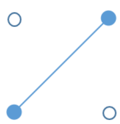
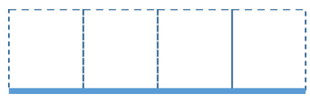
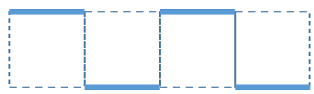
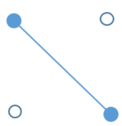
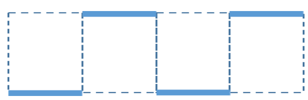
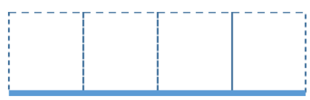
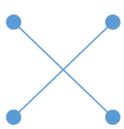
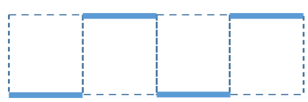
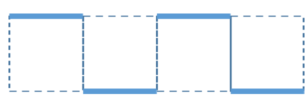
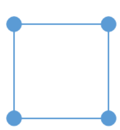
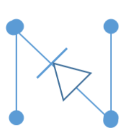
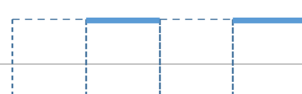


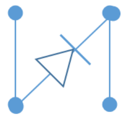
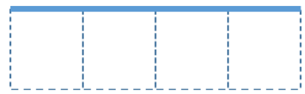
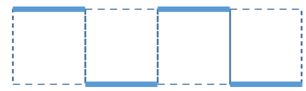
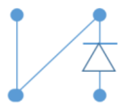
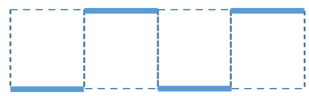
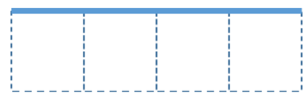
Input 1 Pulse:



Input 2 Pulse:

Chessman Type	Connections	Outputs
1		1. 2. 
2		1. 2. 
3		1. 

		2. 
4		1.  2. 
5		1.  2. 
6		1.  2. 
7		1.  2. 
8		1.  2. 
9		1.  2. 
10		1.  2. 

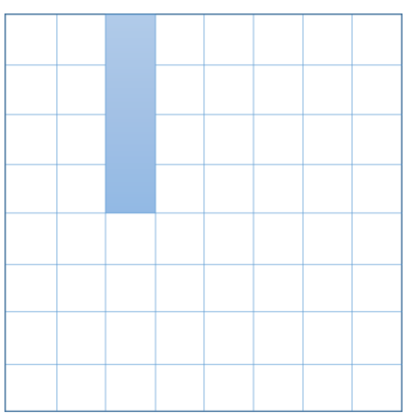
11		1.  2. 
12		1.  2. 

From above, we know that there are 2 feedback pulses for each square, 128 in total.

2 pulse/square * 64 squares = 128 pulses

We will use 8 16-to-1 MUXs (MC14067BCP) in parallel to reduce the 128 lines into 8. With 8 pulses conveyed into the micro-controller, 4 squares are recognized.

Since we use the same select bits for the detection scanning MUXs and the LED column scanning MUX, the 4 squares we detect are in a column, the upper of the lower half, as illustrated below.



2.2.3 Lighting Module

	Inputs	Ouptuts

LEDs	Operation currents	N/A
LED Scanner	Operation currents; 4-bit scanning address	Operation currents
Current Controller	Digital control signals	Analog currents

The LEDs we use are bipolar green/red LEDs (HLMP-4000). Each of them has two diodes which are controlled by 2 individual bits. 4 possible states are produced:

Bits	Color
00	Off
01	Green
10	Red
11	Yellow (Green + Red)

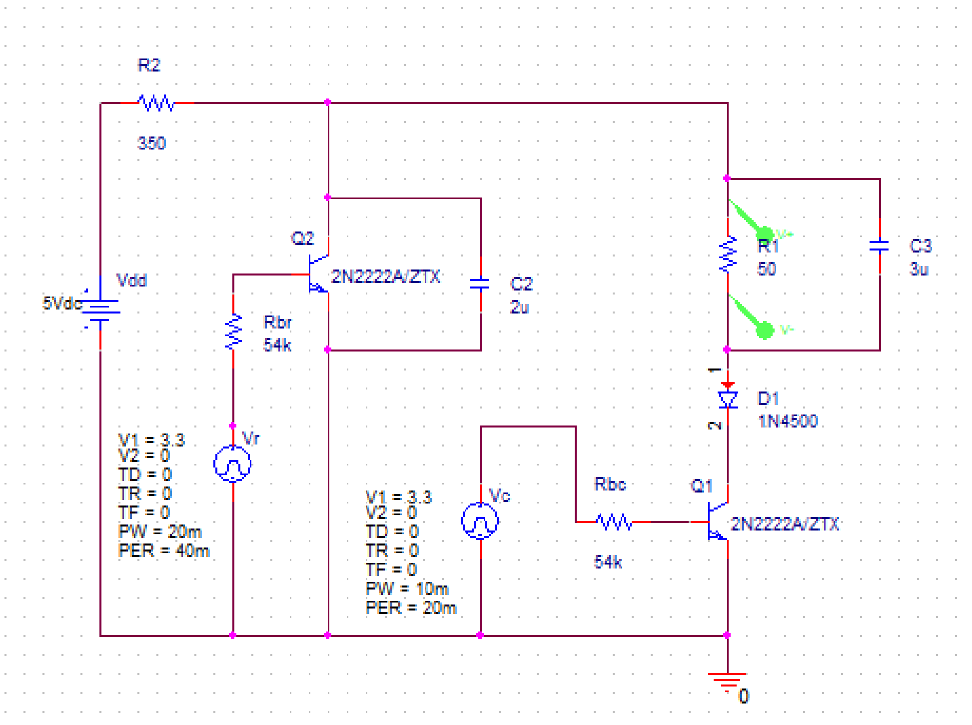
The board is 8 by 8, which means the control bits should be 16 by 8.

16 * 8 = 128 Diodes.

Each of the diode has a current that is controlled by 2 bits, a row bit and a column bit.

To make this possible, three steps are taken.

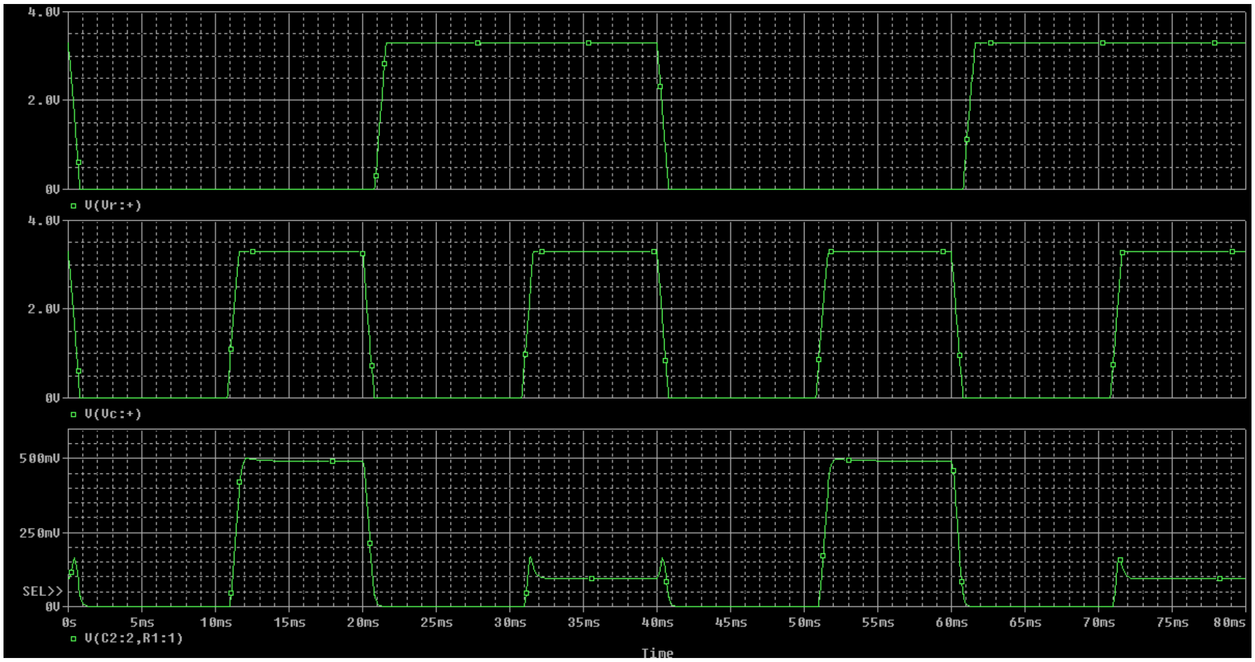
To start off, we need to convert the 2 digital bits which basically are two small voltages into a current that the LED diode will be operating on to be on or off. A simple circuit like following is



designed to achieve it.

We use two pulse voltage source repeating the logical HIGH and logical LOW to see the performance of the diode current.

The way we measure the current is to measure the voltage drop across a 50Ω resistor in series with the diode.



In the simulation, the first and the second plots reflect the voltage of row control bit and the column control bit.

As shown, for a diode, when the row control bit is LOW and the column control bit is high, the current through is 500mV/50Ω = 10mA. This is a current that can light it up.

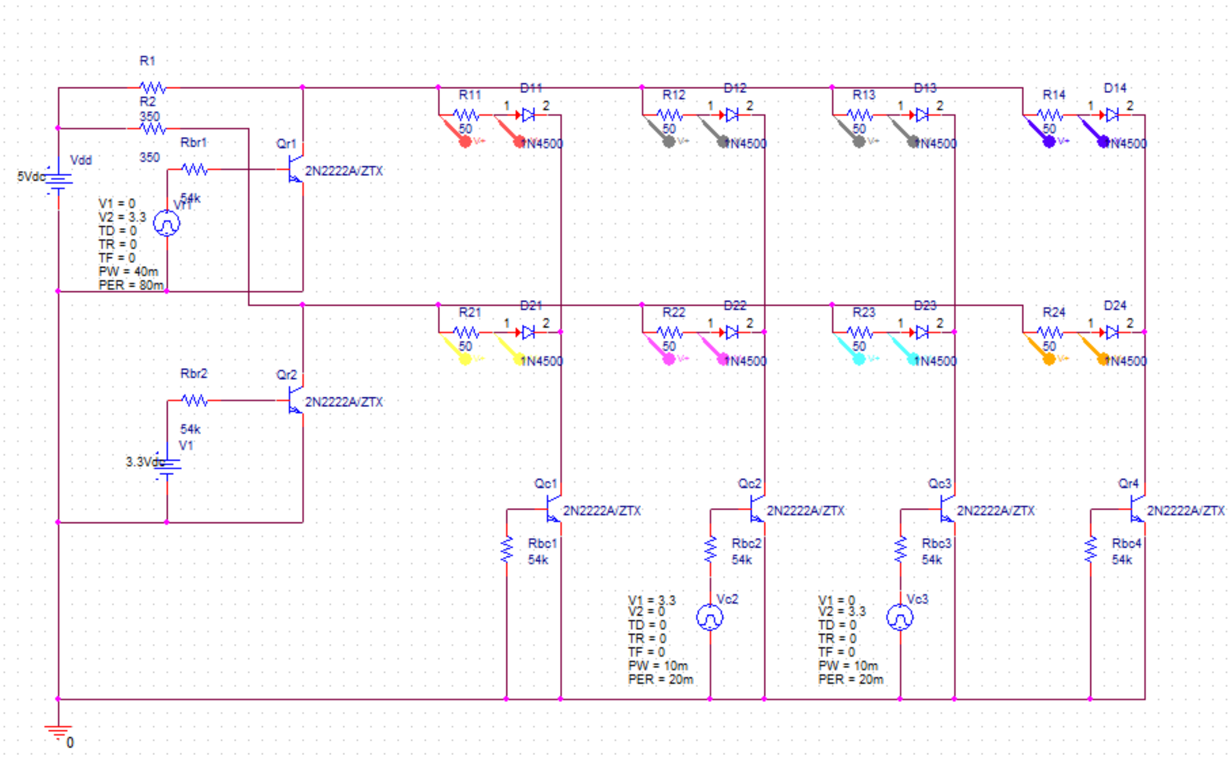
Other than this state, including the glitches, the current is no bigger than 150mA/50Ω = 3mA. The LED is off.

Secondly, it is unnecessary to build 128 current controlling circuit. Instead, we put the diodes in row and in column and scan them row by row. When a diode is focused, all the others will be off.

In addition, although we scan through the diodes, we still need 8 row bits and 16 column bits which may be more than our processor can hold. So just like the detection system, we will use a 1 to 16 DEMUX (MC14067BCP) with the 4 select bits. By doing this, only one of the 16 column is simulated each time. We scan the diodes not row by row, but one by one. This means a lighting

LED is twinkling 8*16 = 128 times slower than the micro-controller operation frequency. But actually this still is fast enough to cheat on our human eyes.

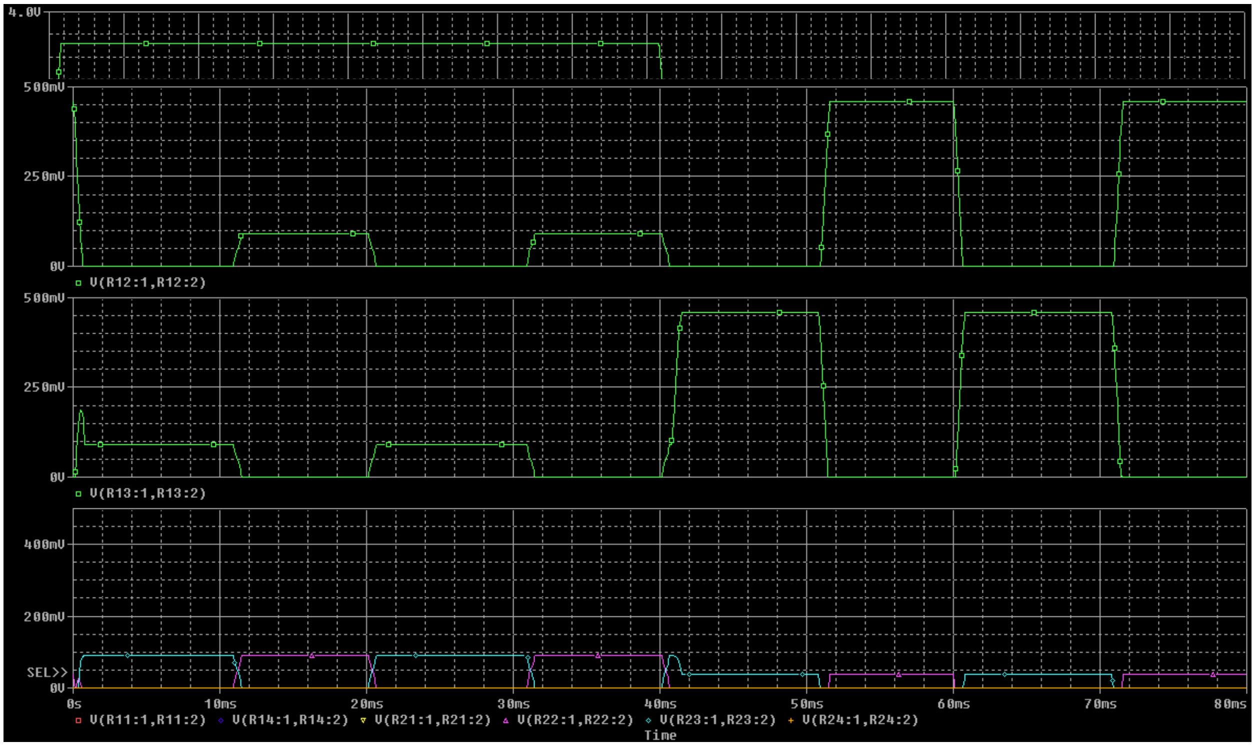
Here is a complete control circuit for a 2*2 board (4*2 diodes). Same thing can be expanded to



an 8*8 chessboard.

To simulate this, we put the first row bit always LOW, and the second always HIGH, which means only the first row can be on.

For the column control bits, we connect column 1 and 4 to the ground to make them off. And we use two inverse pulse sources the scan the diodes at row 1 column 2 and row 1 column 3.



Plot 1 – Row 1, Plot 2 - column 2, Plot 3 – column 3.

Plot 1 – Voltage across R12. Supposed to be on at time 50 – 60ms, 70 – 80ms.

Plot 2 – Voltage across R13. Supposed to be on at time 40 – 50ms, 60 – 70ms.

Plot 3 – Voltages across other resistors. Supposed to be always off.

2.2.4 Micro-controller

Function generator:

- No input
- Output: 30MHz square wave and it's inverting signal

Function generator is connected on the PCB board. When a 4-pin socket is put in the chessboard, the signals from function generator will go through the wire on the socket back to processor. In order to differentiate kinds of chessman, the path on the socket is different for each kind of chessman.

Address

- No input
- Output: 4 bits signal

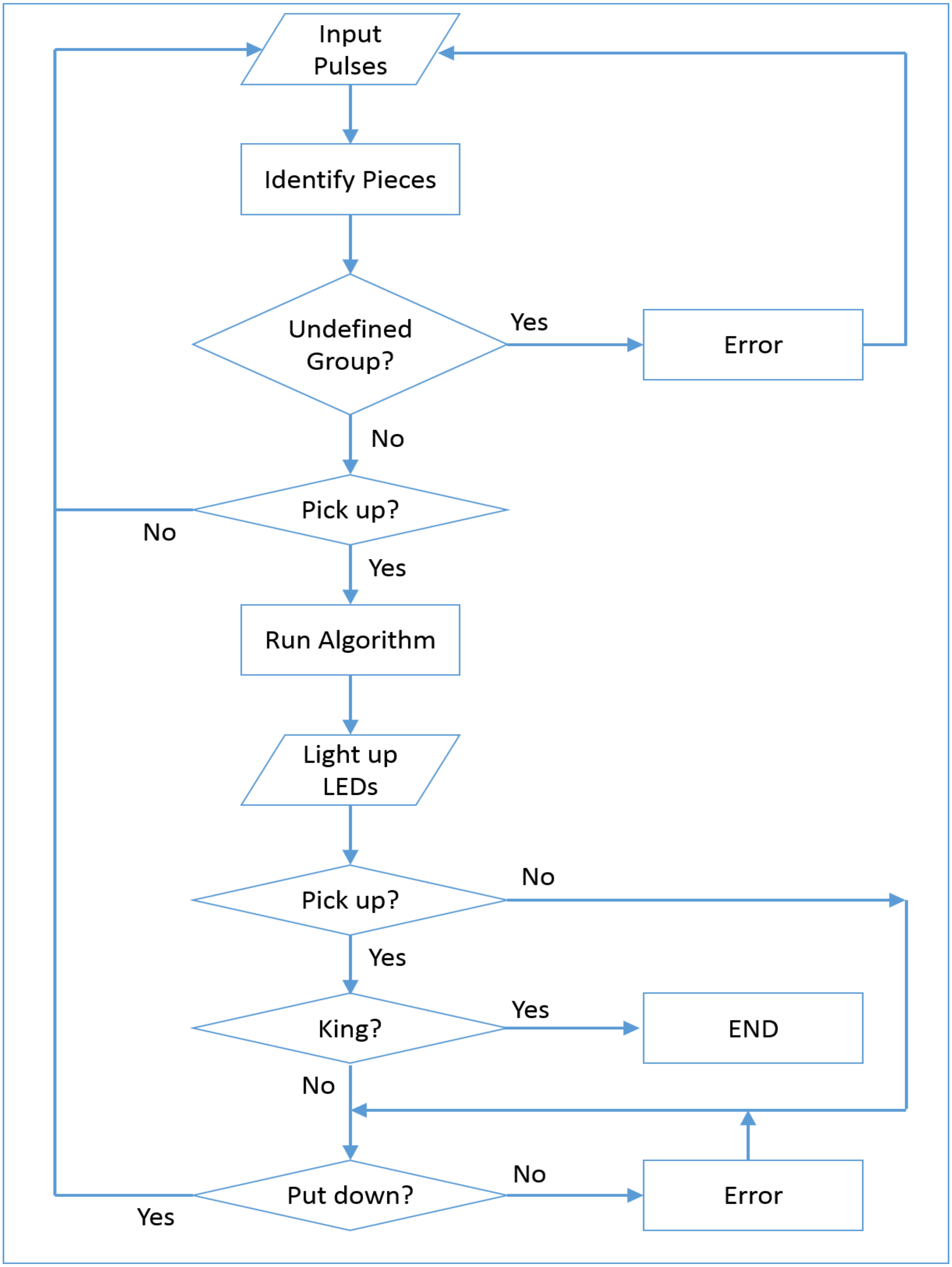
The 4 bits signal from the address will keep counting up to 1111 then back to 0000 in order to get all the information from the MUX so that the information of every signal square on the chessboard will send to AI.

AI

- Input: array of bits that contains the feedback from all the squares
- Output: reference bits for recognizing piece and 9 bits for LED control

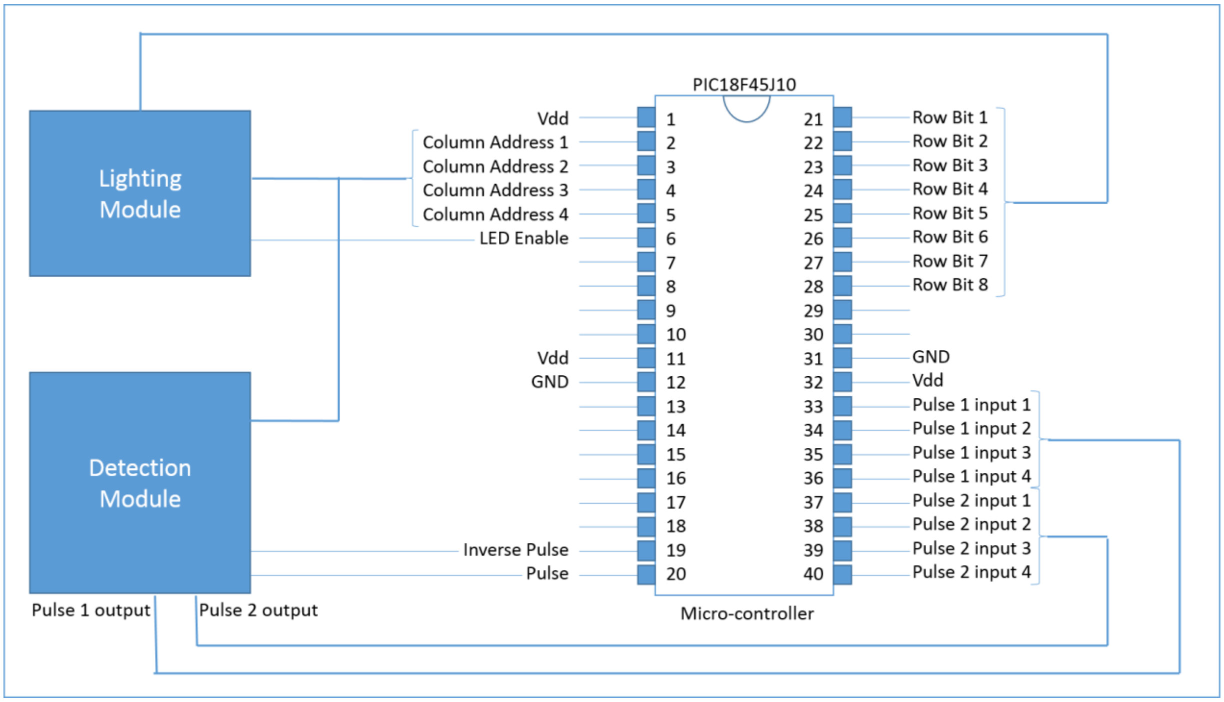
The AI part will organize the digital signal from mux into groups of 4, then compare them with all 12 different 4 bits reference signal in order to recognize the position and the type of chessman on each square. When a piece is picked up, the change of signal will trigger the AI to recall the type on the changing position and search the move for that kind of chessman. Then the Ai will send a 9-bits signal to the LED to tell it which positions should be lighted up.

AI Flow chart



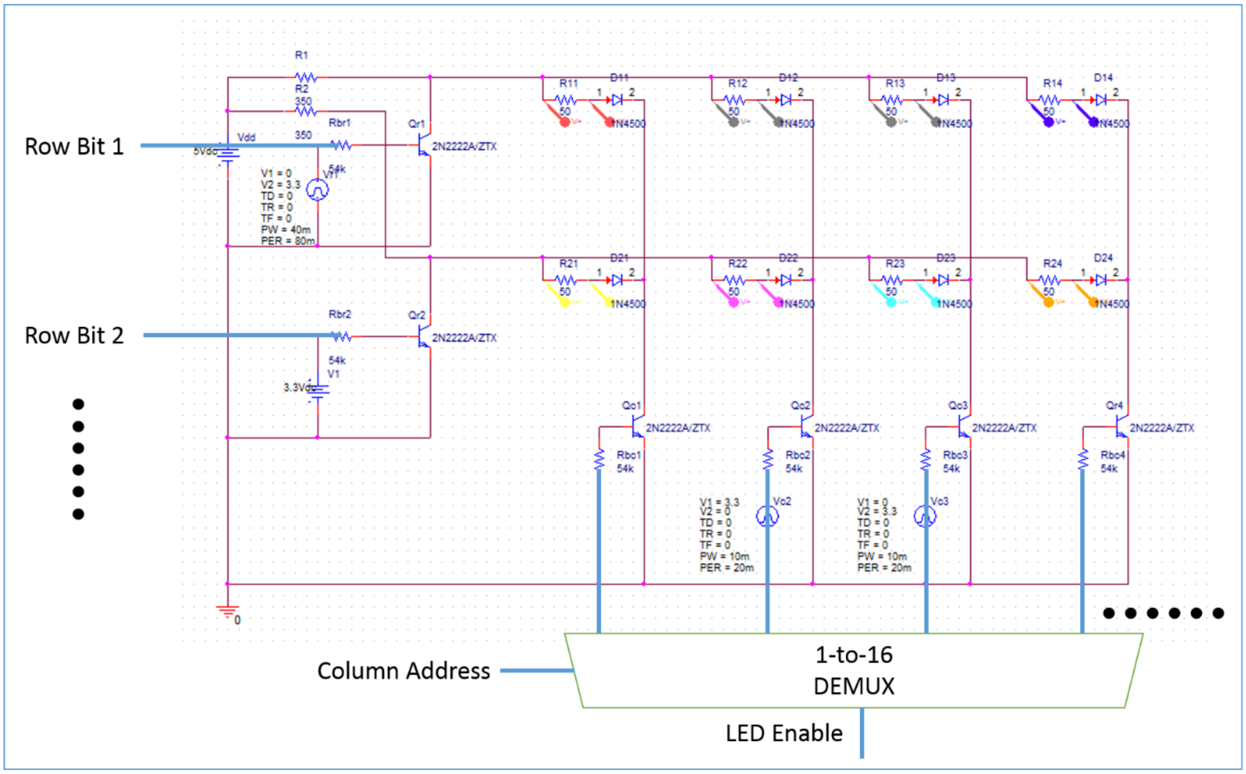
2.3 Circuit Layout

2.3.1 Micro-controller Circuit Layout

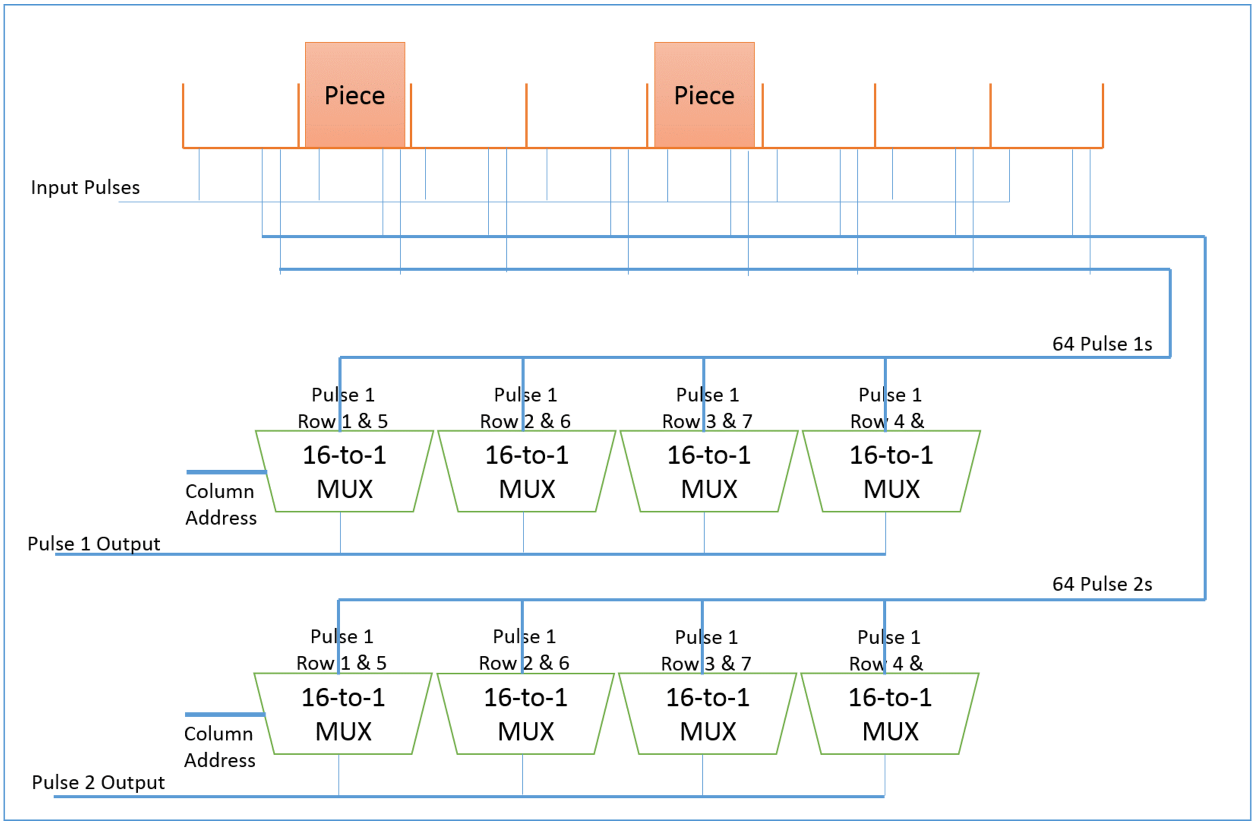


2.3.2 Lighting Module Layout

For the lighting module, we didn't draw out the whole 16*8 diodes circuit. We just use the signals from the micro-controller to replace the AC source on a 4*2 diodes circuit. The 16*8 circuit is exactly the same.



2.3.3 Detection Module Layout



3. Calculations

LED operating current: 10mA

Ic = 10mA

R1=

From Data Sheet of 2N2222A transistor,

VOUT=3.3V

When Ic=10mA, Ib=0.05mA,

RC=460Ω, in case the current won't be large enough to light up LED if VCE is large, we will use

RC=430Ω

Rb=54kΩ

Estimate power consumption:

$P_{signal} = Ic^2 Rc + Ib^2 Rb$ = 0.043W

$P_{tot} \approx 128 * Ic^2 * Rc + 22 * Ib^2 * Rb =$ 5.506W For light up the whole chessboard.

128--#of LEDs

22-- # of transistors

4. Requirements and Verification

4.1 Requirements and Verifications

4.1.1 Detection module

Requirements	Verifications
Piece: 1. The socket could conduct current from 2 input pins to 2 output pins 2. Different connections on the sockets can have different ac signal responses.	1. Connect the output pins and inputs pins, then connect socket with resistor then apply DC power across it, use multi-meter test for current. Desired output: $I \le \frac{V}{R}$ 2. Using 2 function generators to generate 2 identical signals with phase difference of 180. Then use oscilloscope to test 2 outputs from the chessman, and see if different kinds of chessman has different signal combinations.
Pulse scanner: 1. MUX can output from 16 different input pins. 2. MUX can switch output when its select bits are changing continuously.	1. Connect MUX with a 5V DC power supply and test output voltage with multi-meter for different selected input pin. Desired output: 5V from selected input, 0 from others 2. Connect the select bits of MUX with the outputs of a 4-bits counter. Then connect the inputs of MUX to 5V power supply. Manually trigger the counter to count up and switch the

3. MUX can switch outputs when its select bits are controlled by PIC processor.	power supply to different inputs, and then use multi-meter to test the voltage of the output of MUX. 3. Write a counter up code in PIC processor; then connect the outputs of processor to the select pins of MUX. Apply a high voltage to a specific input pin. Connect the output of the MUX to oscilloscope, and observe if the result matches the input series.
Position: 1. No current should be conduct on an empty socket when no pieces are put on a square.	1. Apply a 5V power supply on the input sides, then use multi-meter to test the current on the output pin and observe if the current is zero.

4.1.2 Microcontroller

Requirements	Verifications
Function generator: 1. Processor needs to be able to output voltage when it's powered by voltage supply. 2. Processor needs to be able to generate a pulse signal.	1. Write a code in the processor that can output '1' to all the outputs pins. Then use multi-meter to test the output voltage. 2. Write a code in the processor that can output a square wave to all the outputs

3. Processor needs to be able to generate 2 identical square waves with 180 phase difference to 2 output pins at the same time.	pins. Then use oscilloscope to test the output signal wave. 3. Connecting the PIC chip with oscilloscope and test the outputs from processor and check if 2 channels have 2 identical signals with phase difference of 180.
Address: 1. Processor needs to be able to generate 4 bits signal that can count up to '1111' and back to '0000' continuously.	1. Connect the 4 outputs pins that generate the select bits for MUX to the oscilloscope and test the period of the wave of each bits. And check if the wave fits the wave of a 4-bit counter.
Detection: 1. Processor needs to be able to receive inputs. 2. Processor needs to be able to use the program to compare the inputs to the reference in order to identify the kind of the chess piece.	1. Write a simple code into the processor. Then apply the voltage to different input pins and use multi-meter to check if processor outputs correctly. 2. Using port board of PIC chip to connect the chip with computer. Then enter different combinations of inputs correspond to each kind of chess piece. And check if correct different outputs are showing from detection function corresponds to different

	inputs.
AI: 1. The processor needs to be able to generate correct series of bits that contains the signal to the light up LED.	1. Using port board of PIC chip to connect the chip with computer. Then enter different combinations of inputs correspond to each kind of chess piece. And check if correct different outputs are showing from AI function corresponds to different inputs.
LED control: 1. The processor needs to be able to keep generating signals to 9 different output pins corresponding to the location of the LEDs that should be lighted up. 2. The output voltage should be about $3V \mp 0.3V$	1. Using port board of PIC chip to connect the chip with computer. Then enter different combinations of inputs correspond to each kind of chess piece. And check if correct different outputs are showing from LED function corresponds to different inputs. 2. Power up the PIC chip with 5V power supply, and use multi-meter to test the output voltage of the chip.

4.1.3 Lighting module

Requirements	Verifications
LED: 1. LEDs work properly under DC power supply. 2. LEDs need to be able to work when it's powered by ac source.	1. Connect LED with power supply one by one and see if LEDs light up. 2. Connect the LED circuit with function generator. And look at LEDs, see if it's

3. LEDs need to be able to flash fast enough that look like a constant on state when powered by ac source.	working. 3. Connect the LED circuit with function generator. Increase the frequency until LEDs look like constantly on.
LED scanner: 1. The Demux should be able to output at different output pins when select bits are changing. 2. The Demux needs to be able to work when the select bits are controlled by processor.	1. Connect the select bits of MUX with the outputs of a 4-bits counter. Then connect the inputs of MUX to 5V power supply. Manually trigger the counter to count up and switch the multi-meter to different outputs. 2. Use the same counter up code in PIC processor; then connect the outputs of processor to the select pins of MUX. Apply a high voltage to the input pin. Use oscilloscope to test the voltage of one specific pin and observe if the duty cycle of the voltage wave of that pin is $\frac{1}{16}$.
LED current control: 1. The current through LEDs should always within 10mA$\mp$2mA	1. Connect 1 LED circuit on the bread board, then use multi-meter to test the current through LED when transistors are powered by 5V from the kit box. And connect BJTs' gates with signal

	generator, and set voltage to 3.3 peak ac.

4.2 Tolerance Analysis

The most important part in our project is the LED. In order to make our project work, it has to correctly receive the signals from the processor all the time. Also, LED circuit needs to reduce the current down to the safe working level of LEDs. We will use transistors for each LED, and calculate the desired parameters of transistors. After calculating, we will use bread board to test how well LEDs circuit performs under 5V voltage supply and two 3.3V dc input voltage. For the testing circuit, we will use 4 LEDs connected with 4 transistors.

Then, instead of using 3.3V dc input, we will connect the input of transistors to the output of our PIC processor, and observe if the output from processor can drive LEDs. The testing software will be simpler than the one we will use for our project. Basically, we will use the LED control function we wrote to keep generating signals to the demux and the inputs of two row transistors. Also, the address function will only need 1 bit for test a 2X2 size LED circuit. Instead of receiving the data from AI function, LED control function will receive inputs we gave by using power supply. If the LED circuit works perfectly, we should be able to light up any LEDs that we want to light up by changing the inputs of PIC processor.

4.3 Safety

Manufacturing: LEDs are easy to heat up when the supply current is too high for them. Using resistors to lower the current the whole time is necessary for making and testing LED circuits.

Using: Since our signals go through every chess piece, we need to make sure that users won't get electric shock from the chess piece. Therefore, we need to insulate the chess piece perfectly. But for the prototype, we will only use extra sockets to make each piece bigger so that users can grab the piece easier. Also, the wire inside the piece will heated up. Therefore, this chessboard can't be played continuously for long time.

4.4 Ethics Issues

The purpose this project is to build an electronic chessboard that can recognize chessman pieces and is equipped with a lighting system showing potential moves with an AI algorithm. We commit ourselves to the highest ethical and professional conduct and agree with the first code of the IEEE Code of Ethics:

1. To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

    We will ensure the project will be safe for the public.

3. To be honest and realistic in stating claims or estimates based on available data;

    We will make sure the data is correct and realistic and only build the project base on them.

5. To improve the understanding of technology, its appropriate application, and potential consequences;

    Our project will be used for teaching aids.

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

    We will work as a team, trust and be honest to each other.

9. To avoid injuring others, their property, reputation, or employment by false or malicious action;

We will ensure our project won’t be harmful in any level.

10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

We will support and make sure that none of us violates this code of ethics.

5. Cost and Schedule

5.1 Cost

5.1.1 Labor

Members	Payment/hour	Working hours	#weeks	Total (2.5)
Qianliang Liu	$30	12/week	10	$9000
Ke Ma	$30	12/week	10	$9000

5.1.2 Parts

Name	Price/Unit ($)	Quantity	Cost
4-pin socket	0.07	64	8.96
Multi-color RGB LED (HLMP4000)	1.59	64	18.5
PIC18F45J10 microcontroller	2.27	1	2.27
PCB	0		
Miscellaneous (resistors, capacitors, mux, wire, diode)	10		10
BJT transistor (MPS2222A)	0.174	32	5.568
MC14067B-D 16:1 mux/demux	1.29	9	11.61

5.1.3 Total cost

Labor($18000) +Parts($52.43)=$18052.43

5.2 Schedule:

Week	Assignment	Member
2/11	Researching on parts	Ke Ma
	Circuit layout design	Qianliang Liu
	Mock design review	Qianliang Liu
2/18	Sign up for Design review	Team
	PCB design	Qianliang Liu
2/25	Calculating parameters for the LED circuit	Qianliang Liu
	Testing LED circuit on bread board	Ke Ma
	Other parts testing	Qianliang Liu
3/4	Order PCB board for the first trail	Ke Ma
	Soldering parts	Qianliang Liu
3/11	Testing power	Qianliang Liu
	Testing LEDs	Ke Ma
3/25	Programming into the PIC chip Main	Ke Ma
	Programming functions	Qianliang Liu
4/1	Debugging on computer	Ke Ma
	Testing chessboard function	Qianliang Liu
4/8	Final test and verify hardware	Qianliang Liu
	Final test and verify software	Ke Ma
4/15	Demo	Qianliang Liu/ Ke Ma

6. Reference

IEEE Code of Ethics [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

HLMP-40xx, HLMP-08xx T-1 3/4, 2 mm x 5 mm Rectangular Bicolor LED Lamps Data Sheet, Avago Technologies, e. Obsoletes 5989-4264E AV02-1552EN - September 23, 2008, Web

PIC18F45J10 Family Data Sheet, Microchip Technology Incorporated, 2009,Web

MPS2222, MPS2222A NPN transistor data sheet, Semiconductor Components Industries, March, 2007 − Rev. 3, Web

MC14067B Analog Multiplexers /Demultiplexers data sheet, Semiconductor Components Industries, LLC, 2011,June, 2011 − Rev. 7 web