2-Dimensional Matrix Multipliers with with Incoherent Light

Alexander Scotte*
Department of Applied Physics, Cornell University
Ithaca New York 14853, USA

(Dated: March 7, 2022)

In this paper I introduce topics ranging from optical matrix multiplication to Phonic reservoir computing. I then discuss my current progress with the light source in the incoherent light matrix multiplication project.

I. INTRODUCTION TO OPTICAL MATRIX MULTIPLICATION

The light matrix multiplier or "Stanford matrix multiplier" is a device that uses a variety of light sources to conduct a vector dot product. Through the use of a laser array or LED matrix each, N by M multiplication and addition operation is able to be completed in a single cycle lasting 10 nSec. In a fully incoherent matrix vector multiplier source, light is spread vertically to eventually be imaged horizontally by an optical detector such as a Camera or an array of horizontal light sources.[1]

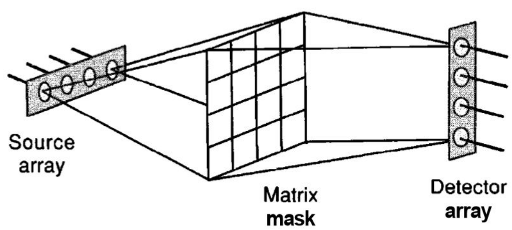

FIG. 1. Parallel matrix-vector multiplier example[1]

The light emitting diode matrix and optical diffraction device both perform linear operations. An optical diffraction device is placed in from of a source to create random optical nodes from the intersection of light. The optical matrix multiplication allows for neural networks to be arranged from the creation of the optical nodes. A device that allows for multiple multiplication layers is related by[2]

$$x_i^{(k+1)} = f(\sum_j A_{ij}^{(k)} x_j^{(k)}) \tag{1}$$

where $\vec{x}$ is related to the number of neural connections $A\vec{x}$ based on the number of inputs into the $k+1$ layer from the k layer.

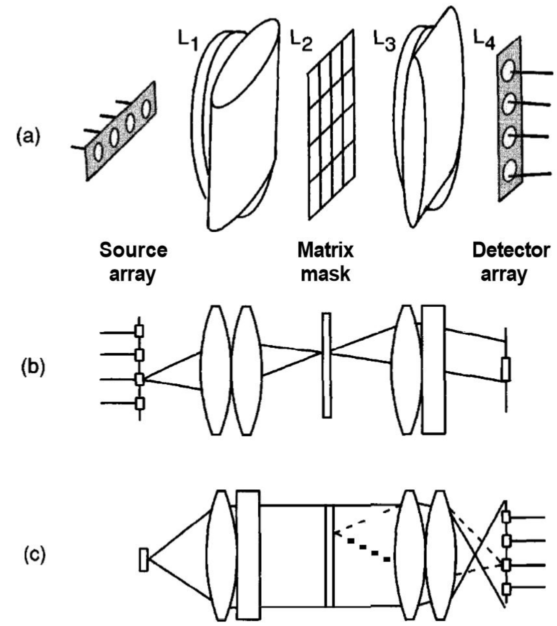

FIG. 2. Parallel matrix-vector multiplier example [1]

A. Applications of A Parallel Incoherent Matrix-Vector Multiplier

Due to the devices ability to perform fast matrix calculations and form optical nodes, a neural network can be created to form an efficient reservoir computer with dimensions (m, n, k) which has an energy consumption that is represented by

$$E_{tot} = (mk + nk)E_{in} + (mn)E_{out} \tag{2}$$

The creation of iterative inversion matrices, Hopfield neural networks, and optical crossbar switch matrices have been constructed with the use of an optical array. The optical matrix multiplier is a useful device that can be used in a variety of fields that require advanced processing.[1]

* McMahon Laboratory, Cornell University, ajs667@cornell.edu .

II. INTRODUCTION TO PHOTONIC RESERVOIR COMPUTING

(The following sections on reservoir computing are based on the findings from [3]) A reservoir computer(RC) is a system consisting of three neural layers that include an input matrix, a reservoir, and a readout node. For the use of a photonic RC the input layer may consist of an array of light emitting diodes which produce a randomized reservoir consisting of a neural network through Parallel Incoherent Matrix-Vector Multiplication. The randomized property of creating optical neurons helps to reduce the reservoir's complexity due to the absence of the need for a re-configurable optical connection link.

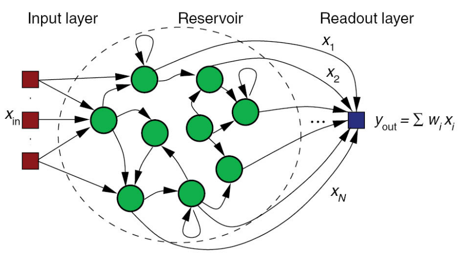

FIG. 3. Reservoir computer layout consisting of an input layer, a randomly arranged reservoir layer, and a readout layer[3]

A. Passive silicon Chip Reservoir Computer

The reservoir computer may be created in variety of ways, including through software and hardware implementation. Physical RC systems have the ability to include higher parallelism, lower power consumption, and larger bandwidth than its software counterpart. However, hardware has a limit on the number of neural connections that may be created due to the modern RC techniques. Current physical systems that include chip silicon reservoir computers have the ability to perform complex tasks at high speeds with relatively low power consumption saturation based on the hyperbolic tangent function. The silicon hardware system uses optical Wave-guides, Splitters, and Combiners to create a linear network of nodes which produces a response that is optoelectrically converted by a detector. Unfortunately, the chip allows for more optical losses and difficulty in measuring nodes in parallel when the system is scaled to include more neurons.

B. Vertical Cavity Surface emitting Laser Reservoir Computer

A VCSEL reservoir computer uses a diffractive element to split the incoming linear laser array into different

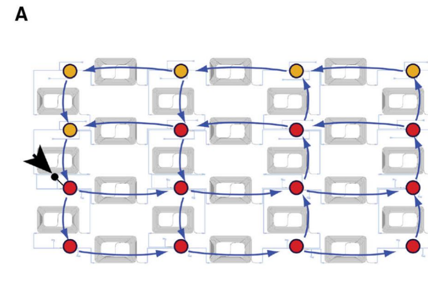

FIG. 4. Passive silicon chip Reservoir [3]

modes which may be modeled by the small angle approximation formula. Different diffractive orders will overlap with their non-mode neighbors creating a a reservoir of coupled optical networks. Large bandwidths are generally reachable with the energy efficient Vertical Cavity Surface emitting laser reservoir computer. The update rate is on the order of 0.5GHz with the current systems running computational tasks. However, the Princeton experiment only uses a Matrix of eight by eight lasers which limits the models update rate. Future devices can be miniaturized to a 1mm squared area to be used in commercial appliances[3].

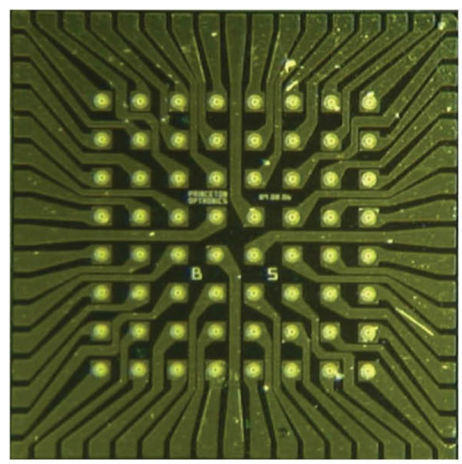

FIG. 5. VSCSEL laser Diodes, Princeton Optronics[3]

C. Applications of a Photonic Reservoir Computer

Photonic Reservoir computers offer a wide range of benefits over conventional computers. These include, but are not limited to faster computational speeds of complex tasks (near that of light), much greater energy efficient systems, optical header recognition systems, and faster control loops. Through the use of optical properties, these advanced computers may be able to function within large processing bandwidths.

III. CURRENT PROGRESS

A. LED matrix Introduction

The current RGB matrix model created by Ada fruit contains 2048 light emitting diodes in a 64 by 32 array. This matrix has the purpose to operate as the test light source of the incoherent matrix multiplier. Each LED's brightness can be individually controlled to change the state of each vector in the Array. Due to the shape of each LED the matrix produces spherical waves that are passed through a diffractive device to allow for matrix multiplication and the creation of nodes for a neural network for a reservoir computer. With the use of all three possible colors there are over 1.429559296 E+9 combinations and each one has a varying level of brightness.

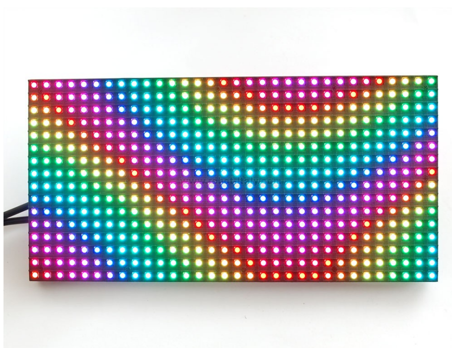

FIG. 6. LED matrix[4]

B. LED matrix Components and technical features

Each that exists can't be driven individually due to the large quantity on the matrix board. To fix this problem the array includes a driver similar to a 74HC595 to control two lines of the array at once. The first and ninth lines are connected with each other and follow a similar pattern to allow for a cleaner image as the board refreshes. While the circuitry does allow for individual control of each LED, it lacks the ability to have pulse width modulation. For a screen to change states the entire screen needs to turn on and off which limits the control of brightness levels for the LEDs.[5]

C. Matrix statistics

The lack of pulse width modulations creates problems when trying to achieve full LED brightness for the entire matrix. In order to achieve one hundred percent color, the board must be running at high speeds with the best results occuring at around 50MHz. However, certain colors require increasing amounts of current and

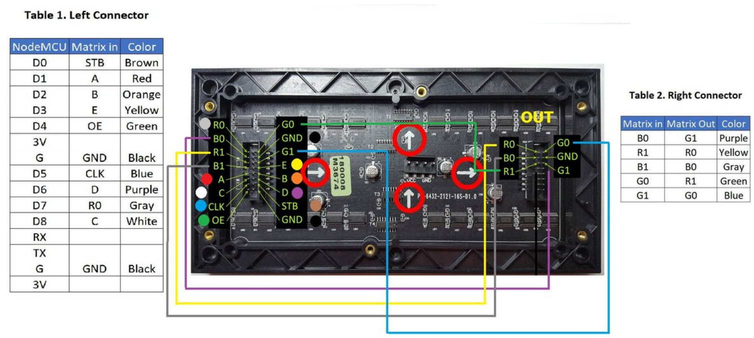

FIG. 7. LED matrix components[4]

full RGB(white) requires the most. Current would usually remain constant for each LED color, but during the test, clock speeds are increasing and they required less frequency changes at higher values.

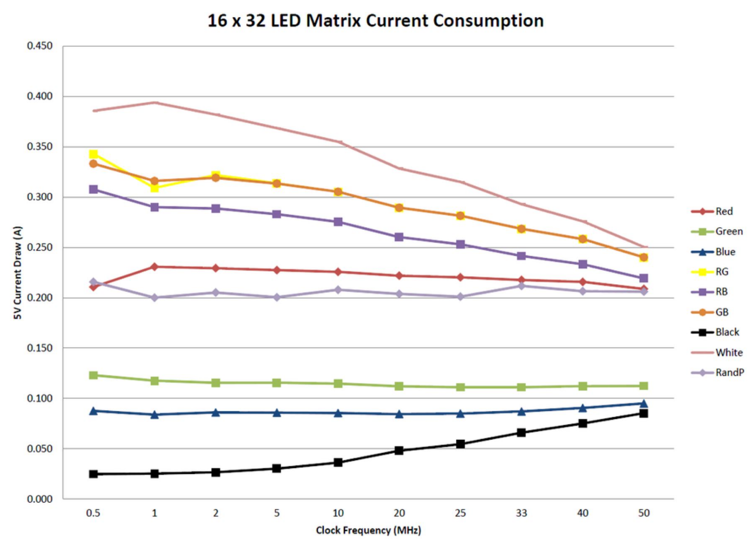

FIG. 8. LED matrix refresh rate vs current consumption by by rhb.me (CC By-NC-SA)[6]

D. Matrix Control System

In order to achieve the high clock speeds needed to power the 64 by 32 LED matrix the raspberry pi 3 was introduced to power the system. With a processing speed of about 1.20 GHz and a 64-bit quad core processor, the clock speed was well beyond the 50 MHz recommendation. The microprocessor runs on a form of linux called Raspbian where python script is its main language.

In practice, running the matrix with the Raspberry Pi creates a problem where, then LED near the triggered sources will light up(known as ghosting) due to the inefficient spread of current throughout the control circuit. However, ghosting tends to lessen when the entire board is being used instead of small sections of LEDs. Since every LED purposely has current flowing through its diode connections, the bias current that is emitted by each pin on the LED controller has less of an effect on the state of each light source. For example, if a small section of the matrix is being used, the LED controllers which are responsible for controlling those specific areas may allow for a small bias current to flow through specific non-active LEDs. This allows for red light to appear in supposedly non-active LED's since the bias current delivers enough energy to bypass the red light voltage threshold. The controller is not an ideal circuit element that can block specific nodes. In comparison, a modern LED controller acts as a non-ideal op-amp that cannot block bias current from flowing into the positive and negative input terminals.

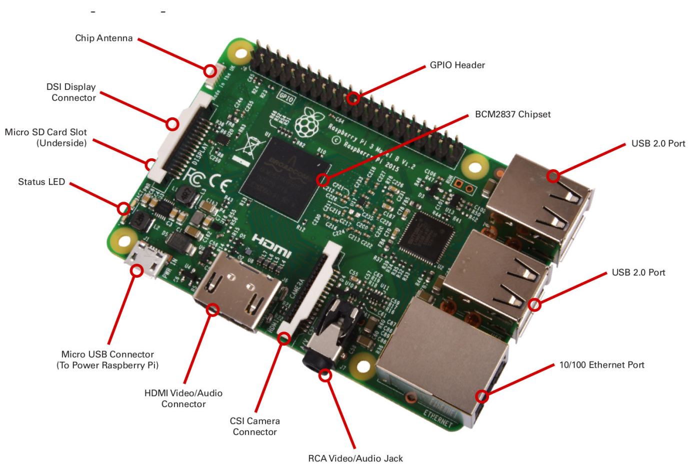

FIG. 9. Raspberry Pi information[5]

E. Current GITHUB repositories and data

In order to test the Raspberry Pi's communication method with the LED matrix, two online GitHub libraries were used. The repository (https://github.com/adafruit/rpi-fb-matrix)[7] was mainly used to prove the matrix's functionality. This library allowed for the display of numerical data on the LED array, however it presented a major ghosting problem without the ability to solve the issue, due to the lack of basic LED control commands in the library. The data being represented on the screen presented basic settings such as the number of matrices connected in parallel(only one at present) and number of LED sections being controlled(four on one array). Further tests have proven that this library is unreliable since the ghosting problem increases with each use.

A second repository (https://github.com/hzeller/rpi-rgb-led-matrix)[8] was primarily used to prove the array's ability to refresh and present different displays. The example files allowed for the presentation of a rotating cube with a variety of colors at a refresh rate of about 150HZ. The hzeller library offers the ability to limit ghosting effects, and allow for pulse width modulation. Testing proves that the higher pwm (the –led-pwm-lsb-nanoseconds[around 300]) command and increased LED occupation of the matrix will result in less ghosting. However, increasing control of the LED population allows for a decrease in refresh rate from about 300Hz to 150Hz(at full board control). An increase in pwm also decreased the brightness quality of each LED being used.

IV. LOOKING FORWARD

A. Raspberry PI and LED Matrix communication protocols

In the current model an Ada-fruit matrix bonnet connects to the Raspberry Pi's GPIO pins so that the matrix is controlled through SPI protocol by the microprocessor. At the same time, the Raspberry PI is currently controlled through an external GITHUB repository API protocol.

In order to reach the required speeds to prove the matrix's practicality, an external library will have to be created with the an increased refresh rate as its primary goal. The use of the API protocol will be used to control and easily download a library to the Raspberry Pi.

B. Refresh rate testing

The GITHUB repositories being used to control the LED array have a command that displays the screens refresh rate. However, this may not accurately represent the physical system. In the future I plan on creating an experiment involving a light sensitive camera and the LED matrix. Software created by a fellow team mate analyzed the refresh rate presented in the camera's video. This data will provide a more accurate representation of matrix's abilities and help us to see its reliability.

C. Final overview

The current goal of my project is to provide a reliable and fast optical array for the matrix multiplication with incoherent light reservoir computer experiment. The optical device needs to have a faster refresh rate to prove better than an OLED screen provided by a common smartphone. Due to the screens common use in electronic hobbyist projects, the array needs to prove it has the ability to be easily reprogrammed and updated unlike a smartphone. This project will allow for a practical answer to the problem of how to create a reliable energy efficient light source.

ACKNOWLEDGMENTS

I would like to thank professor Peter McMahon for this opportunity to help contribute to optical computing research and advance my own knowledge of physics. I would also like to thank Tianyu Wang for allowing me to help contribute to his incoherent light matrix multiplication research project. Finally, I would like to acknowledge Chengji Liu, Joseph Zullo, and Yimeng Zeng for helping on this project to help determine the best device to control the LED matrix and code other instruments.

[1] J. W. Goodman, *Introduction to Fourier Optics* (The McGraw-Hill Companies, 1996).

[2] R. Hamerly, L. Bernstien, A. Sludds, M. Soljacic, and D. England, Large-scale optical neural networks based on photoelectric multiplication, Physical Review (2019).

[3] G. V. der Sande, D. Brunner, and M. C.Soriano, Advances in photonic reservior computing, Nanophotonics **0132**, 561 (2017).

[4] Arduino backpack board for rgb matrix panel.

[5] P. Burgess, Harmonics elimination in three phase cascade h-bridge multilevel inverter using virtual stage pwm.

[6] A. Quedan, How the matrix works.

[7] T. DiCola, rpi-fb-matrix.

[8] H. Zeller, hzeller/rpi-rgb-led-matrix.

[9] X. Lin, Y. Rivenson, N. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozean, All-optical machine learning using diffractive deep neural networks, Science. **361**, 1004 (2018).

[10] S. Popoff, G. Lerosey, R.Carminati, M. Fink, A. Boccara, and S. Gigan, Measuring the transmission matrix in optics: An approach to the study and control of light propagation in disordered media, Physical Review Letters (2010).