Summer 2000

Cellular automata study of the feasibility of a parallel optical computer

Jason Robert Lane

CELLULAR AUTOMATA STUDY OF THE FEASIBILITY OF A PARALLEL OPTICAL COMPUTER

by

JASON ROBERT LANE

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2000

Approved by

[illegible]
Dr. C. H. Wu, Advisor

Steve E. Watkins
Dr. Steve E. Watkins

William F. Parks
Dr. William F. Parks

ABSTRACT

The speed of modern electronic computers has inherent design limitations which will soon be reached. A possible substitute is the optical computer. In this project, the feasibility of the optical computer is explored by designing and building a parallel optical system. This system is used to test the basic principles that would govern a parallel optical computer. The system is based on the principle of cellular automata, which is a simulation technique used to study interactions of objects in a system. The following goals were set. The system would be based on cellular automata. A one-dimensional array with nine cells, or data bits, in the array would be studied. The transition rules governing how the data is modified would be easy to change. Finally, the system would be as fully optical as possible, with electrical counterparts allowed as needed. These goals help steer the direction of the design, and are discussed in this thesis. The design is built, and test data is introduced to determine proper operation of the system.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. C. H. Wu for his help and patience during my completion of this project, without which, this project would not have been possible. I would also like to thank Dr. Watkins and Dr. Parks for providing information and assistance when it was needed.

I would also like to thank my friends, and especially Mike Simpson, whose patience and support were much welcomed during the course of the project.

Finally, I would like to thank my family, who knew how to keep me working on this project when nothing else worked.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1. THE PROBLEM FACING MODERN COMPUTERS

The trend in today's computer development is towards faster and faster personal computers. Processor speeds have increased steadily, if not exponentially, over the past few years. It is now possible to purchase a 1 GHz processor for use in a personal computer, and the trend is not likely to end anytime soon.

However, computers will not just keep getting "faster and faster." There are physical limitations to how fast the modern computer can get. For example, transistor switching time, however short, is inherently limited by the transit time due to resistance and capacitance between the p and n junctions. This in turn is determined by the transistor size. Transistors are formed using a photolithography process, where a flash of light is directed thru a mask to a photoresist layer above a silicon dioxide layer on the substrate, causing the exposed portions of the photoresist to be easily dissolvable. The revealed silicon dioxide layer is then etched away with chemicals.[1] Transistors today are unbelievably small. So small, in fact, that diffraction around the edges of the mask is becoming the limiting factor in how small they can get. The current solution is to use a higher frequency light source (current photolithography processes use ultraviolet or x-ray light sources) to allow more finely detailed masks. This will be the significant factor in determining how 'fast' computers can get in the years to come.[2]

There are other factors, too, such as bus speed being limited by electromagnetic sensitivity, heat dissipation concerns, and speeds of moving parts such as hard drives. All of these factors contribute to the widely held belief that a limit on computer speed will soon be reached, and most would agree that optical components should replace electronic components in the computers of tomorrow.

1.2. THE SOLUTION

So the question seems to be what components should be replaced? A brief scan of scientific journals over the past few years will reveal much research into developing specific solutions to the problems mentioned above. One solution that comes to mind is replacing the busses in a computer with fiber optics [3]. The busses are the slowest parts of a personal computer today, and would benefit the most with being replaced by optical components. Electromagnetic sensitivity would be drastically reduced, and data transfer from CPU to other parts of the system would be able to approach speeds comparable to that of the CPU clock itself. (The current bus speeds available for the home computer range from 100 to 200 MHz, whereas CPU clocks are approaching 1 GHz.)

Another recent development of interest is that of the holographic storage device to replace the compact disk and hard drive [4][5]. This device is basically a stack of many holographic plates that can be both written to and read from using a three-laser system for addressing and data read/write. The storage capability is comparative to today's largest hard drives, with the added bonus that due to the nature of a hologram (specifically, a Fourier transform of the image, not the image itself being stored) the device is very rugged and dependable, able to take abuse which would render a compact disk unreadable.

These are two of the better developments that the author has seen, and it is his hope that they are implemented over the next few years. Yet these developments provide only stopgap solutions to the inherent speed limitations of the electronic computer architecture. So why not build a fully optical computer? A fully optical computer would eliminate the problems associated with a fully electronic computer, such as electromagnetic sensitivity, heat problems, moving parts, and transistor switching time limitations.

Though nice in theory, it is not a simple task to just pull out the motherboard, hard drive, memory chips, etc., and replace them with optical boards, holographic cubes, lenses and mirrors. The obvious reason is that optical device technology is not as well developed as electronic device technology. This means that optical components are much larger and more expensive than electronic components. Another, more important reason is that an optical computer modeled on an electronic computer would not take full advantage of an optical system's benefits. That is, the ability of an optical system to process large amounts of data at the same time.

An electronic computer is a serial system, where data is processed one word at a time. That is, a memory location is addressed, the word, whether 8, 16 or 32 bits in length, is fetched from memory and fed to the central processing unit (CPU), actions as specified by the current instruction are performed on the word, and the results are returned to memory or output to an output device. This process is repeated over and over, depending on the program being run. Though a bulky process, modern electronic computers are designed to perform millions of these instructions per second, and hence run complicated operations quickly.

An optical computer would not rely on speed as much as it would rely on massive parallelism. Thousands of words could be retrieved from memory simultaneously, fed into the CPU, actions according to the instructions given performed on the data all at once, and the results returned to memory or output to an output device. If the memory device is a holographic cube as mentioned above, the read/right operation would be measured in milliseconds, and thus millions of operations per second could be performed with an optical computer. To give an idea of the speed, if an operation is performed on a million data bits in a 1000 x 1000 grid every millisecond, this is comparable to one operation being performed every 10^{-9} seconds, or operating at 100 MHz. Note this is not because of the speed of the system, but because of the massive amount of data being operated on, and thus operations could be performed as quickly as on an electronic computer.

Most optical computers built to date in various laboratories are based on optical models of the electronic computer, with different subsystems modeling electronic counterparts [6]. Thus, these optical computers are basically serial systems, operating on one word at a time. This study is of an optical system based on the cellular automata scheme proposed by Von Neuman [7] and further discussed by Murdocca [8], as a basis for a truly parallel optical system, where large amounts of data are operated on at the same time.

2. CELLULAR AUTOMATA COMPUTING

2.1. INTRODUCTION

Cellular automata formed the basis for the system because no optical computer based on cellular automata was known to be built in a lab setting. Cellular automata is a simulation technique used to study interactions in systems of many objects [9]. Changes in the states, or configurations, of the objects are dependent on the previous states of neighboring objects. The transition functions that govern how the cells are changed are called rules, and the rules depend only on neighboring cells of the automaton. As opposed to a serial system where the state of the objects would be updated one at a time, all objects are updated simultaneously, making cellular automata ideally suited for forming the basis of a highly parallel computer network system. See the Appendix, Table 1 for comparisons between an optical computer.

The most famous example of cellular automata is Conway's Game of Life [10]. First published in 1970, the rules for this survival simulation are simple: Take a 2-D array of cells, with filled cells being live, and empty cells being dead. If two or three live cells surround a filled cell, then it survives to the next generation. If exactly three live cells surround an empty cell, then that cell is born (filled) in the next generation. All other cases result in the cell becoming or remaining dead.

These simple rules allow surprisingly complex, stable patterns to form. As computers have become fast enough to process hundreds of generations in a few seconds, research into Conway's Game of Life has grown exponentially [11]. Patterns have been discovered which duplicate themselves, add two binary numbers, count from one to ten (and output 1, 2, 3, etc.), and even simulate a single cell in Conway's Game of Life [12].

2.2. SUBSYSTEMS IN A CELLULAR AUTOMATA COMPUTER

Three operations are required for computing with cellular automata. These operations are divided into search, control, and replacement [13].

2.2.1. Search. The transition rule activates when certain neighbors for the cell take on specified values. This requires that the entire array of cells be searched to find the neighbors that satisfy these values. Multiple rules require multiple searches. Also, the rules should not be conflicting. Otherwise additional complexity is introduced in the form of rule priorities.

The best method to search the neighbors for a cell is to use interconnections to bring the values from the neighboring cells together. The connections are space invariant since the same operation applies to each cell.

Search also requires logic capability. Logic is the underlying mechanism for manipulating symbols in all forms of computations. Specifically, once the values for the neighboring cells are brought together, logic operations are performed to determine if the resulting values meet the requirements laid down by the transition rules. This can be done several ways. Correlation may be used to search for the specific patterns, *and* operations could be used to search for patterns of 1's, and *or* operations could be used to search for patterns of 0's. These latter two may be implemented by value addition with thresholding.

2.2.2. Control. Control of data flow is critical to cellular automata computing. Control would allow some regions of automata to perform different operations than others. For example, one half of the automata grid would perform *or* functions, while the other half would perform *and* functions. Another instance would be to fix the values of some cells in the array. These cells would act like boundary conditions or obstacles for steering the data along the 2-D array.

2.2.3. Replacement. Finally, the values of the cells must be replaced every generation. This includes clearing the previous values and updating the cells while the information is progressing around the loop. Therefore, synchronization is required for a cellular automata computer. This synchronization can be accomplished with the use of storage or memory devices.

Another part of the cellular automata computer is the ability to insert initial values to begin computations. Although important, it is not a separate component, as the input pattern can be loaded from the memory device in the replacement subsystem.

3. CRITERIA GOVERNING THE DESIGN OF THE SYSTEM

3.1. CRITERIA LISTED

As stated previously, it was decided early on to design a system based on cellular automata. This was due largely to the fact that, to the author's knowledge, no cellular automata computer has been built to date. Also, it was recognized that the general definition of cellular automata allows great flexibility in the design of an optical computer. Specifically, computational power is limited only by the transitional rules defined. In the one-dimensional array case studied during the course of the project, 256 different rule definitions can be applied using nearest three-neighbor comparisons (the cell and its two nearest neighbors). The formula that gives the number of different rule definitions for a given number of neighbors is as follows:

$$\text{number of possible unique rules} = 2^{2^{n}} \qquad \text{eq. 1}$$

Where n is the number of neighbors included in the comparisons.

Obviously, criteria had to be laid down for the design of the system. The first criteria, namely that the system be based on cellular automata, has already been discussed. It was at this point in time that the criteria were set. These criteria are as follows:

1) The system will be based on the principle of cellular automata.
2) The simple case of a one-dimensional array will be studied.
3) The size of the array for the system to be built is set at 9 cells.
4) The transitional rules must be easy to change.

5) The system will be optic in nature. When needed, electrical components or systems with optical counterparts can be used.

3.2. REASONS FOR THE CRITERIA GIVEN

3.2.1. One-Dimensional Array. The simple case of a one-dimensional array was studied because the principles apply to the two-dimensional case as well. Converting a one-dimensional system to two dimensions would be nothing more than duplicating the original system and modifying the interconnections. Therefore, the project focused on one dimension, and studies in a two-dimensional system were left for simulations only.

3.2.2. Nine Cell Array. It was preferred that as many possible input conditions could be viewed at once. Since the project focused on two-neighbor comparisons (the cell itself and its right-hand neighbor), but with the ability to expand to three-neighbor comparisons, an array of cells at least eight in length was deemed adequate. This was to cover all the cases ranging through two inputs of *low-low* to *high-high*. A nine-cell array was decided on to provide a little extra flexibility in choosing initial inputs without making the system too bulky.

3.2.3. The Transition Rules Must Be Easy To Change. To keep the system flexible, it was to be designed so that as little modification as possible was needed to change the transitional rules. In other words, separate systems were not wanted for separate rules. Ideally, the rules could be loaded into the system much like a program is loaded into a computer. If this complicated the system too much, then minor hardware modifications would be allowed.

3.2.4. Optical System With Electrical Counterparts Allowed. Since the goal of the project was to determine the feasibility of an optical computer, it was obvious that optics would be used. The last criterion was a cost criterion. That is, to keep the system as inexpensive as possible. Costs in optical components, especially active optical components, can range into the tens of thousands of dollars, and so electronic counterparts were allowed. For example, storing the output data in an optical storage device may be cost prohibitive. Thus the data could be stored in a conventional electronic equivalent, such as a flip-flop or memory chip. As a consensus, the optical light path would not be interrupted from source to output plane, to stress the inherent speed benefits that come from optical data transfer.

Even with these five criteria limiting the scope of the project, it was discovered that much flexibility was still allowed in the system. All criteria were met in the final design of the system. The evolution of this design is discussed next.

4. DESIGN OF A SIMPLE OPTICAL CELLULAR AUTOMATA SYSTEM

4.1. INITIAL DESIGN

4.1.1. Coding The Data and Comparing Neighbors. The first step in the design process was deciding how to code the data optically. The obvious choice is to simply make filled cells lit for logical *high*, and empty cells dark for logical *lows*. Thus, an initial data pattern would have relative intensity levels of 0 and 1, as shown in Figure 4.1. With a laser beam enlarged and collimated, a shadow mask pattern could be inserted

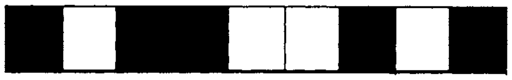

Figure 4.1: Sample input pattern

after collimation and before the output plane, giving the initial data pattern as shown. For right-hand neighbor comparisons, the pattern could be copied either by a beam splitter or by using Texas Instrument's Deformable Mirror Device (DMD) [14]. The copy would be right-shifted one cell, and then both patterns added at the output plane. In the case of the DMD, two mirrors would represent one cell. Each mirror in the DMD has two positions available: Tilted 10° to the left, and 10° to the right. With one mirror tilted 10° to the left, and one mirror tilted 10° to the right, it would serve the same function as a beam splitter, but with the added benefit of programming the array of mirrors into many different arrangements.

4.1.2. Building a Working Model. Since the mirrors of the DMD measure on the order of micrometers, a large-scale model of 18 DMD mirrors was built to the specifications shown in Figure 4.2. Since the system was working with right hand neighbor comparisons, it was not essential to build anything more complicated than a

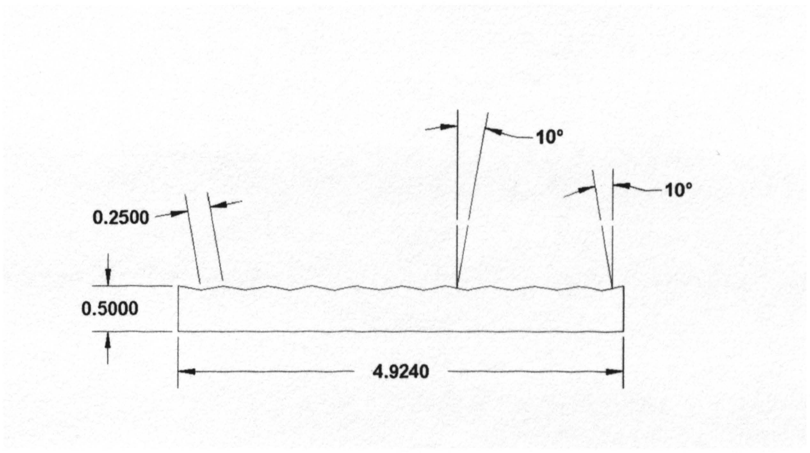

Figure 4.2: Macro scale model of a DMD (Measurements in inches)

static model, with the mirrors alternately tilted 10° in either direction. Once this was done, the design shown in Diagram 1 of the appendix was built in the laboratory.

It was hypothesized that there would be 3 relative intensity levels at the output plane: 0 if both input cells being compared were dark, 1 if one or the other cell was light, and 2 if both cells were light. An array of photo resistors was built as shown in Figure 4.3 to measure the voltage levels and confirm this hypothesis. When two cells were lit, the voltage measurement ranged between 485 and 520 mV. When one cell was lit, this value ranged between 240 and 260 mV. Variations were due to non-uniformity in the light source, since it was not deemed critical to render the light source uniform, only collimated. When no input cells were lit, the voltage across the photocells measured about 10mV. This small voltage was due to 'background noise,' i.e. light from the other

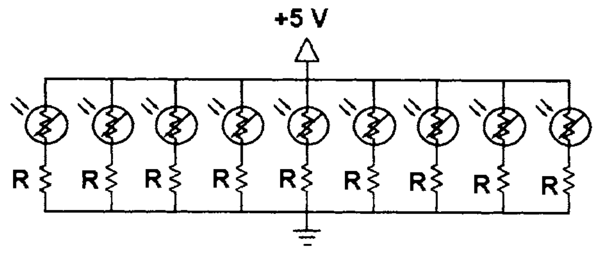

Figure 4.3: Photo Resistor Array R = 10kΩ

room, lit display panels, the computer monitor, etc. These measurements supported the hypothesis that the output would have relative intensity levels of 0, 1, and 2, although there was some variation due to the background noise and non-uniformity of the light source.

4.2. PERFORMING MULTIPLE ITERATIONS

Now that a single iteration had been achieved, the next logical step was to perform multiple iterations. The question now arose as to how to bring back the output back to the input plane in the same format as the input. That is, how to change the 3-level output to a 2-level output.

4.2.1. Thresholding. Obviously, thresholding would be essential for this part of the system. Thresholding is defined here as modifying an optical data bit such that a value above a predefined intensity level would register as a logical *high*. A value below this predefined intensity level, or threshold value, would register as a logical *low*. If necessary, this could be reversed such that a value below the threshold level would register as a logical *high,* and a value above the threshold level would register as a logical *low*. Examples of thresholding are given in Figure 4.4. In the first example, the threshold value is set at a relative intensity level of 0.5, with the result that a logical *or* is performed on the two inputs. In the second example, the threshold value is set at a relative intensity level of 1.5, and the result is a logical *and*. This would require an active

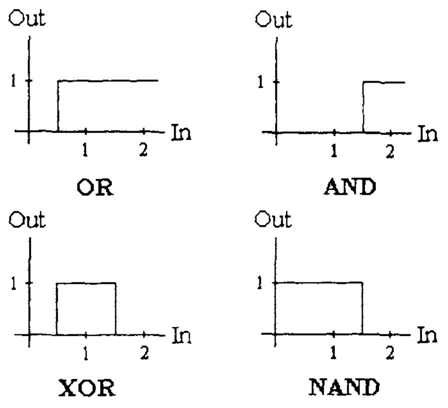

Figure 4.4: Thresholding examples

optical device such as a spatial light modulator (SLM) to perform the thresholding.

4.2.2. Synchronization of the Iterations. To refresh the input array when the output array became available, a device such as a SLM or a LCD display would be needed. Also, a synchronization clock would be required to keep the system from 'running away,' or refreshing the input too fast for the system to be of any use. After all, the optical light path from input to output and back was on the order of tens of centimeters, and light does not take very long at all to travel that distance.

SLM's were the obvious choice, both at the input plane and at the output plane. However, these devices range into the tens of thousands of dollars apiece, so substitutes were sought . An array of photo resistors on-hand from the measurements previously mentioned was the ideal choice for detecting the values of the cells at the output plane. Each value could then be fed into a synchronization device such as a flip-flop, before replacing the previous values at the input plane. A transparent LCD panel was desired for the input plane, but a suitable LCD panel could not be found. The reason was that, outside of custom-made panels, the displays available on the market were of unsuitable size. Since a custom-made panel would be too expensive, the single laser light source and input plane were replaced with an array of high-output LED's. This had the added benefit of allowing the output of each flip-flop to be connected directly to each light source. An LCD panel would require the addition of a controlling unit, adding unnecessary cost and complexity to the system. This substitution allowed the design to remain simple, which was a desired goal of the project.

4.2.3. Designing the Thresholding Circuit. To provide enough voltage to drive the flip-flops, the output of the photo resistors needed to be latched to near +5 Volts for a logical *high* or near 0 Volts for a logical *low*. This was accomplished using the circuit design in Figure 4.5. Since an *or* function was chosen as the initial test operation, it was

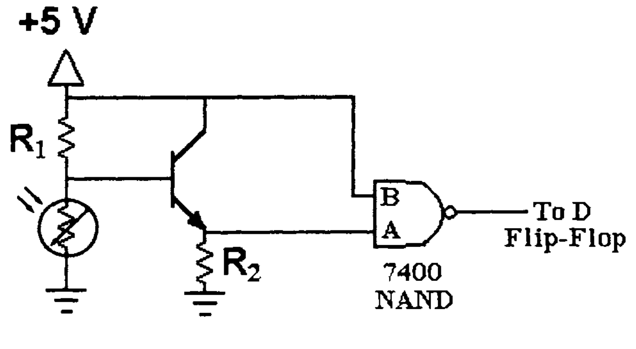

Figure 4.5: Thresholding circuit

desired that the output of the flip-flop was latched *high* when a relative intensity level of 1 or 2 was incident upon the photo resistor. A simple voltage divider was chosen to drive the base of the pnp transistor, with the value of R chosen such that the transistor was turned off when the resistance of the photocell dropped due to the appropriate amount of light falling upon the photocell. Thus, when an intensity level of 1 or 2 was incident upon the photo resistor, the transistor would be turned off, providing a logical *low* to input A of the *nand* gate. Since input B was latched to 5 Volts, this would provide a logical *high* output to be fed to the D-type flip-flop. The appropriate LED would then be turned on at the next rising clock edge.

When an intensity level of 0 was incident upon the photo resistor, the transistor would be turned on, providing a logical *high* voltage to input A of the *nand* gate. Since input B was latched to 5 Volts, this would provide a logical *low* output to be fed to the flip-flop. The appropriate LED would then be turned off on the next rising clock edge. The circuit would thus allow a logical *or* function to be performed.

4.2.4. Building the Cells. Initially, a single cell was built for testing voltages at various points, choosing a value for R, and aligning the LED with the photocell. The resistance value for R was expected to be different for each cell, as this would change due to manufacturing differences in LED's, photo resistors, and also slight differences in alignment between LED and photo resistor.

Once testing of the single cell and voltage measurements taken at various points was accomplished, additional cells and circuitry were added to the system. It was discovered that the resistance value R used in the first cell was adequate for use with the additional cells. As right-hand neighbor comparisons were desired, alignment of the LED's were needed such that light from each would fall upon the appropriate photo resistor and the photo resistor to its left. To limit the amount of light from each LED reaching the output plane, the LED's were placed in 0.9 cm holes drilled in a project box measuring 20.3 x 7.7 x 2.6 cm with additional holes drilled directly opposite the LED's as shown in Figure 4.6. These holes, acting as aperture windows, limited the incident

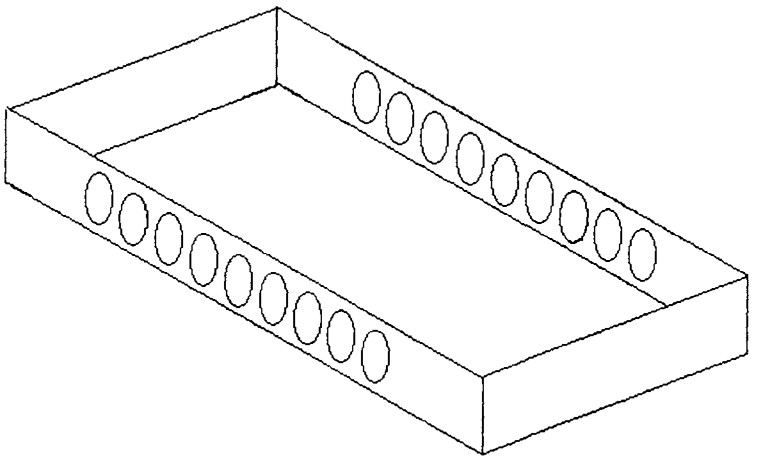

Figure 4.6: Line drawing of project box used to hold LED's with holes cut as shown

light falling on the output plane to the cell's photo resistor and its left- and right-hand neighbors. To limit the amount of light further, wooden slats measuring 14 cm in length were placed against the project box between the apertures and the output plane. Also, wooden slats were placed inside the project box in between the holes. Thus, with little decrease in intensity, the light coming from a cell's LED was limited to falling upon that cell's photo resistor and it's left-hand neighbor, allowing the desired addition of a cell's value with its right-hand neighbor's value at the output plane.

4.2.5. Testing the System. When all nine cells were built, testing began with initial patterns. At this point in the design, the boundary condition of the first cell in the array remaining in its initial state throughout the iterations was imposed. An example is shown in Figure 4.7. Thus, if the first cell in the initial pattern was *high*, then it would remain *high* throughout the iterations. As was expected with an *or* function, if a cell was initially *high*, then all cells to the left of it (as looking at the output plane from the input plane) would eventually reach a stable state of *high*. Thus, for example, if the first cell was *low*, and the second cell was *high*, then cells three thru nine would eventually turn *high*, irregardless of their initial state.

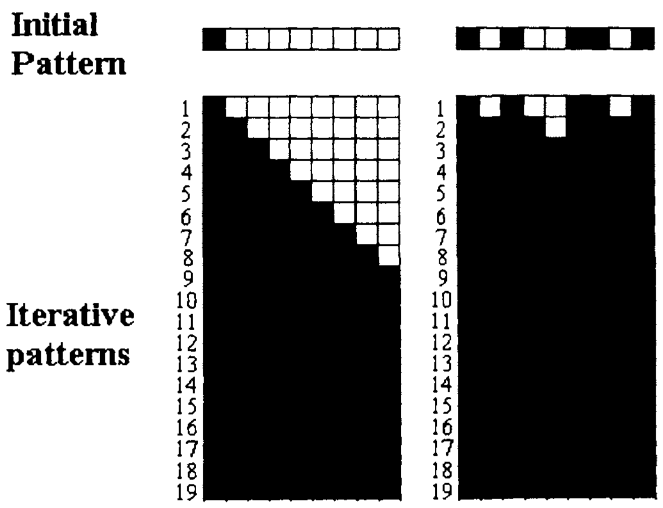

Figure 4.7: Iterations for system with boundary conditions in place

At this point, the boundary condition was removed by placing mirrors on each side of the bank of photo resistors and aligning them such that light from the last LED was incident upon the first photo resistor, in addition to falling upon the last photo resistor in the array. Thus, if cell nine was *high*, then cell one would become *high* on the next transition. This was the final design as desired for performing a logical *or* operation. Testing various input patterns revealed that eventually all cell values would become logical *high*, as expected. Figure 4.7 gives an example of two initial input patterns tested, plus the iteration patterns shown up to the point where a stable output was reached. By stable output it is meant that further iterations cause no change in the output pattern.

4.3. FINAL DESIGN

The final stage of the project was to modify the system so that either a stable output could be reached where some cells were *high* and some cells were *low*, or an unstable output could be reached. The system is shown in Figure 4.8. By unstable output it is meant that the output array does not settle down into any pattern. That is, the output continuously changes for infinite iterations. It was hypothesized that an *xor* or an *xnor* operation would provide this, and so an arbitrary decision to perform an *xor* operation was made. This meant that the cell value would be latched *high* on the next iteration if its value or its right-hand neighbor's value was *high*, and the cell value would be latched *low* if both the cells were either *high* or *low*.

To accomplish this, two thresholding operations needed to be performed simultaneously: One thresholding operation would check if the relative value of the intensity level at the photo cell was either 0 or 1, and one thresholding operation to check if the intensity level was 1 or 2. The design for this system is shown in Diagram 2 of the appendix. As the voltage divider/transistor setup already performed the first of the two desired thresholding operations, additional circuitry was only required to perform the second operation and then compare the two thresholding operations. This required the addition of a second photo resistor for each cell at the output plane, connected in a voltage divider setup to the base of an npn transistor in a similar fashion as before. In this case, the transistor is turned on if the relative intensity level at the photo resistor is 1 or 2. Since it was desired that the cell's LED would be turned on at the next transition if and only if both transistors were turned on (meaning an intensity level of 1 was detected at the output plane), both transistor outputs were fed into a logical *nor* gate. The output of the *nor* gate was then fed to the D-type flip-flop which drove the LED's.

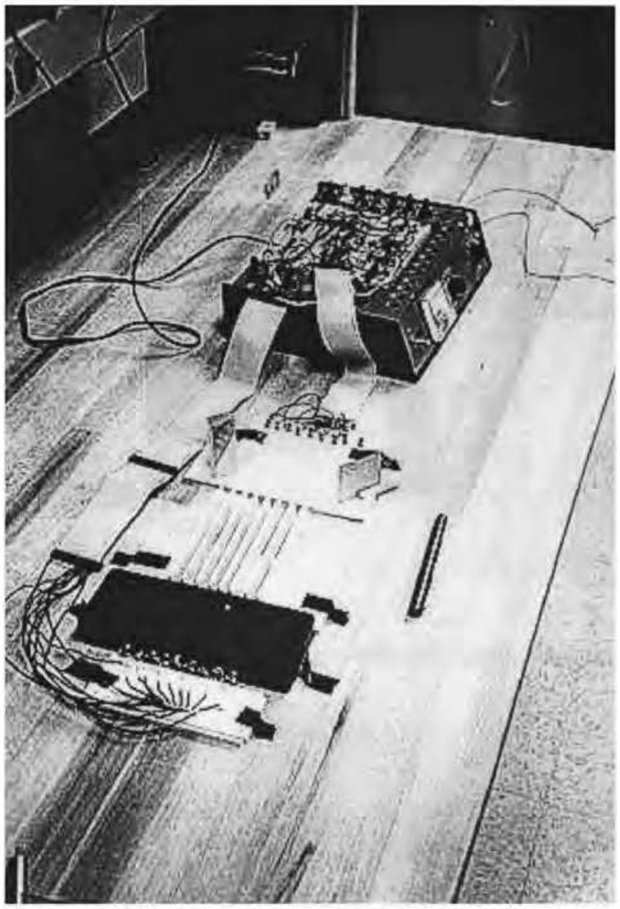

Figure 4.8: Final system

This system proved much more sensitive to variations in the amount of light incident upon the photo cells than the previous system. What this meant was that there was little variation in what the value of R_2 could be, and so each cell ended up with a different value for R_2. Also, as opposed to the previous system, slight movement of either the LED's or the breadboard on which the array of photo resistors were placed caused the system to malfunction, and so care was taken to avoid misalignment of these two subsystems. Figure 4.8 shows the final system as set up in the lab. Figure 4.9 shows the LED box and the dividers used to keep light on its intended path.

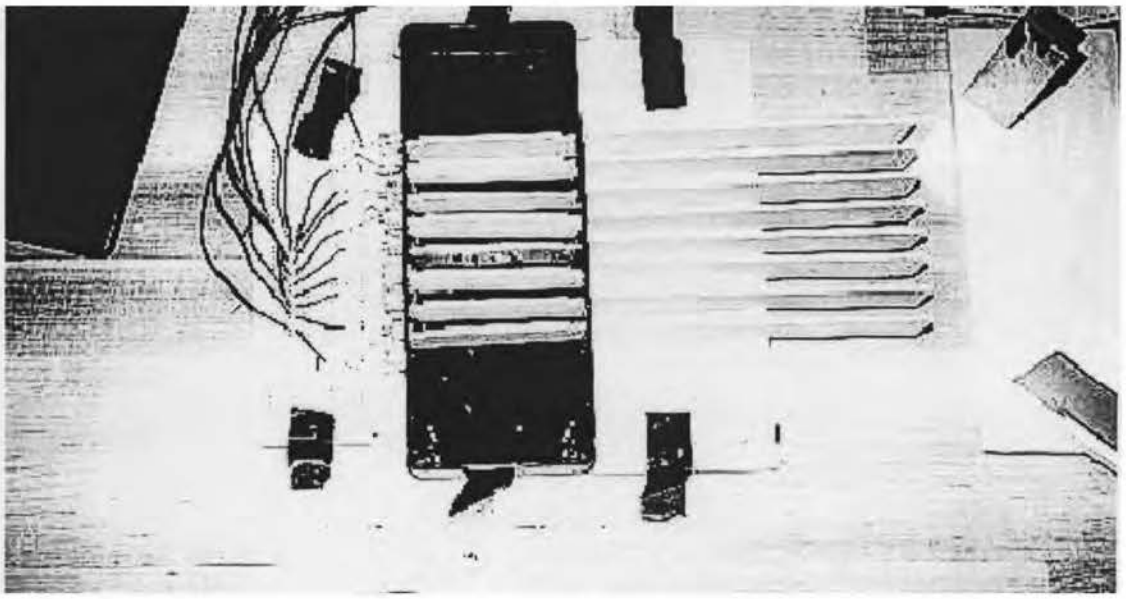

Figure 4.9: A second view with dividers shown

5. TEST PATTERN RESULTS

5.1. WITH A BOUNDARY CONDITION

Many input patterns were tested on the final design. Figure 5.1 shows one example. The clock was set at 1 Hz to facilitate recording of the iterative patterns. Three examples are given, the first being with a boundary condition as described below, and the

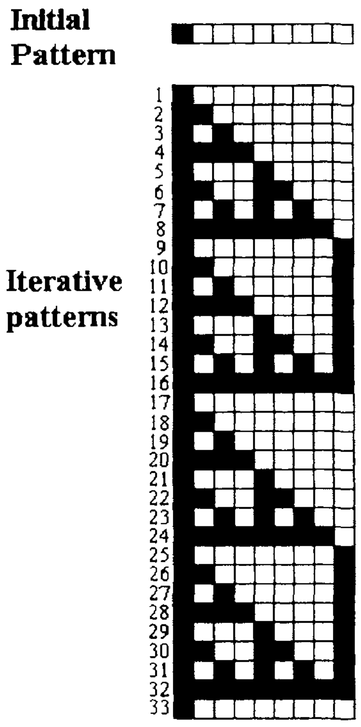

Figure 5.1: Initial pattern tested

next two with that boundary condition removed.

The initial pattern for an iteration was loaded in by simply cutting holes in a cardboard sheet and placing this sheet at the end of the wooden slats. Light by means of a laser pointer was then introduced to the appropriate photo resistor, allowing their respective LED's to turn on. Once this was accomplished, the pattern was automatically loaded into the system, whereas light from those LED's fell on their photo resistors, which then fed back to the input plane, keeping the pattern in place. The cardboard sheet would then be removed and iterations begun.

For adjustment purposes, the boundary condition was applied where the first cell was allowed to keep its current state. After all cells were operational, the first input pattern was introduced as shown in the initial pattern in Figure 5.1. Cell 1 is shown on the left in these figures. The iterations were copied down as they occurred, and these iterations are also shown. In the figure, 33 iterations are shown, with the initial pattern at top, and the 33rd iteration at bottom. Notice that the pattern repeats after 16 iterations, and also note the fractal nature of the pattern formed, as noted in. [15]

5.2. WITH BOUNDARY CONDITION REMOVED

The rest of the test patterns were conducted with the boundary condition removed. That is, the state of the ninth cell affected the state of the first cell. For comparison, the next test pattern was the same as the first, and the iterations are shown in the same format in Figure 5.2. In this case, the initial pattern is not repeated. However, the second iteration is, with the exception that the pattern is left shifted one cell every eighth iteration. Although only 33 iterations are shown, the pattern does repeat itself after 65 iterations. Again, it was surprising to discover the fractal nature of the patterns formed. Although the project does not touch on this result, it could form the basis for future research. Figure 5.3 shows the system in operation. In this photo, cell 1 is at the bottom. Note the two mirrors used to transfer light from the last LED to the photo resistor for cell 1.

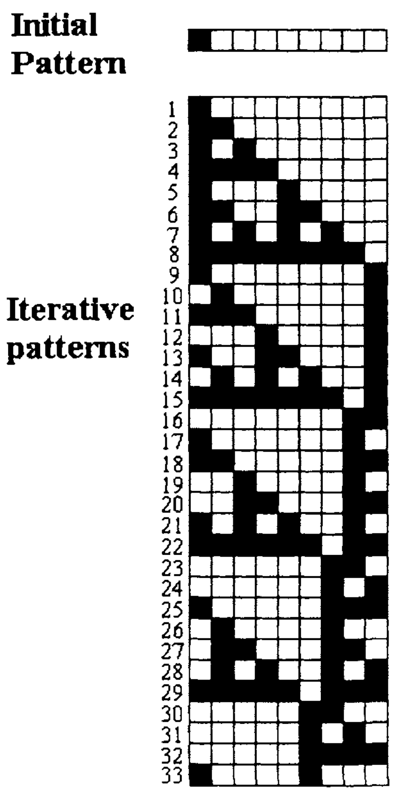

Figure 5.2: First pattern with boundary condition removed.

This pattern was chosen because eventually, every possible input state occurs for every cell sometime during the iterations. For example, cell 2 would be *high* or *low* for each of cell 3's two possible input states. This can be seen in iterations 1 through 4 in the previous figure. Thus, it was easy to spot errors in the operation of the system so that adjustments could be performed quickly.

The next pattern is shown in Figure 5.4, with its following iterations given. This iteration is interesting because the pattern appears to move backwards across the cells. In other words, the pattern is shifted to the left, although the system is designed so that state of a cell affects the state of the cell to its right. Also, the pattern repeats itself every fourth iteration, although the initial pattern is not repeated.

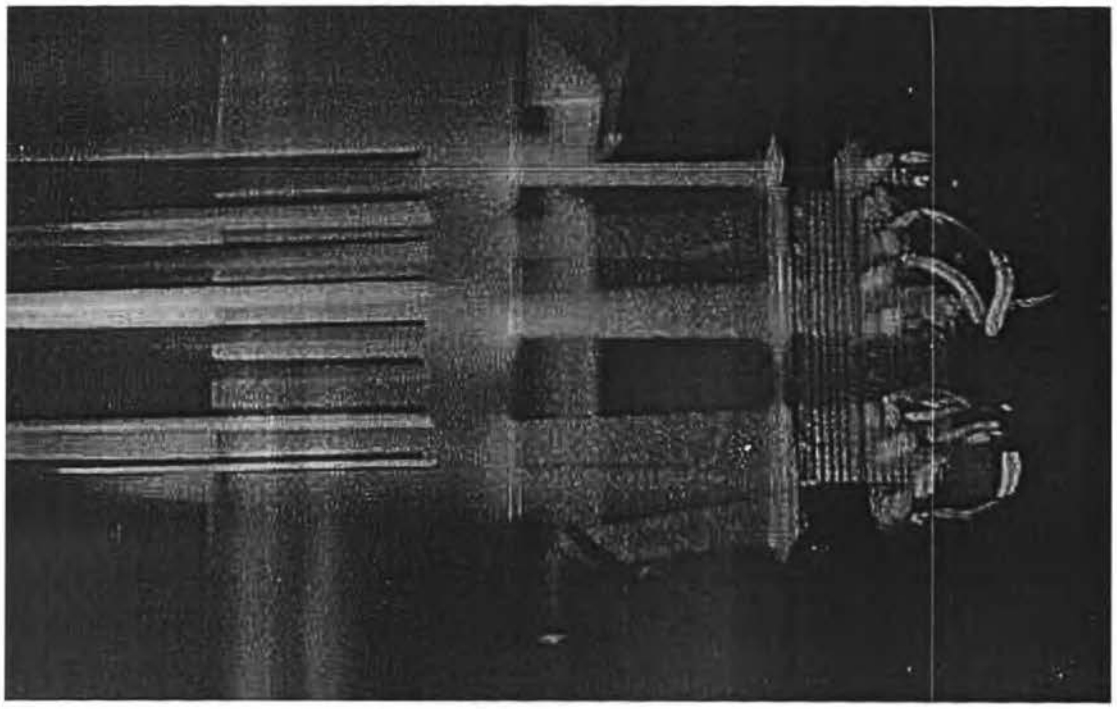

Figure 5.3: System in operation, showing light paths incident on output plane.

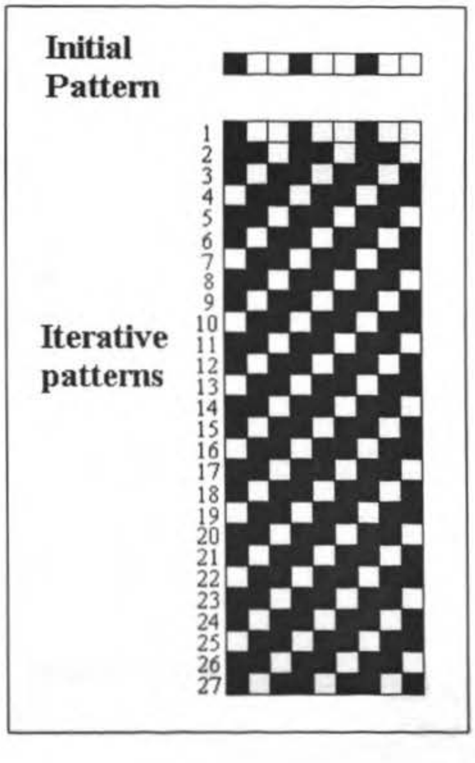

Figure 5.4: Final test pattern

6. FURTHER RESEARCH

These final tests concluded the actual work done for the research project. However, it is obvious that the project could form the basis for future research. So the question arises as to where the research could lead. The obvious next step would be to develop the rules for right-hand neighbor comparisons further. For example, the system does not differentiate between when light from the LED or the neighboring LED falls on the photocell. This would allow a rule such as the output is on when the LED is on and its neighbor is off.

After this, both-neighbor, or 3-cell, comparisons could be considered. The right-hand neighbor comparison as studied in this project is a special case of both-neighbor comparisons, and rules can be formed for both-neighbor comparisons to model the right-neighbor rules as studied above (in addition to others for 3 cells). Beyond that, 2-dimensional systems could be developed, with rules for comparing nearest 4 neighbors, nearest 8 neighbors and so on. The goal could be to model Conway's Game of Life in an optical cellular automata system.

In the meantime, the hardware could be developed further. As mentioned previously, all the electrical components used in the project have active and/or passive optical counterparts. This was a goal of the project since the criteria were set, with the underlying idea that these electrical components could eventually be replaced. Also, the size of the system could be drastically reduced, such that, for example, a 1000x1000 array of cells could fit in the same area as the system built. Also, a subsystem could be added to provide easy loading of input patterns and rules used to govern the cellular automata.

7. CONCLUSIONS

This project was begun with the idea that an optical computer would be an ideal replacement for the modern electronic computer. The goal of the project was to determine the feasibility of an optical computer based on the principle of cellular automata. As it was desired to design a parallel system, cellular automata was chosen due to its inherent parallel comparison of many objects in a system. The second step in determining the feasibility of an optical computer was to design a system based on cellular automata. Criteria were set down to facilitate the design of this system. Finally, the system was built and tested. Due to the relative simplicity of the system built, and the large amount of flexibility in the design (specifically due to the ability to determine and change the rules on which cell comparisons are based), the conclusion is that a parallel optical computer based on cellular automata is not only feasible, but would be relatively uncomplicated and easy to build. An unexpected benefit was how much further this project can be taken. The comparisons can be enlarged to encompass both nearest neighbors in a 1-dimensional array, allowing 256 different rules to be performed. The 1-dimensional array could be expanded to two dimensions, the size of the system could be reduced, and the electronic components used in the system built can be replaced with optical counterparts. Obviously, this research project could form the foundation for research into optical computing for many years to come.

APPENDIX

These are the two designs used during the course of the project. Figure A.1 is a fully optical design to test voltage measurements, alignment, etc, and only performs one iteration of cellular automat. Figure A.2 is the circuit diagram used in the final design of the project.

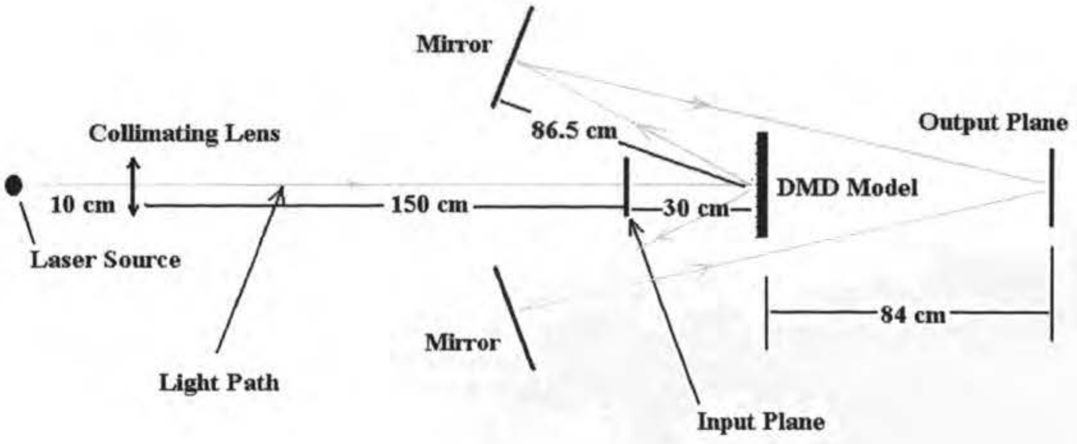

Figure A.1: Initial Design

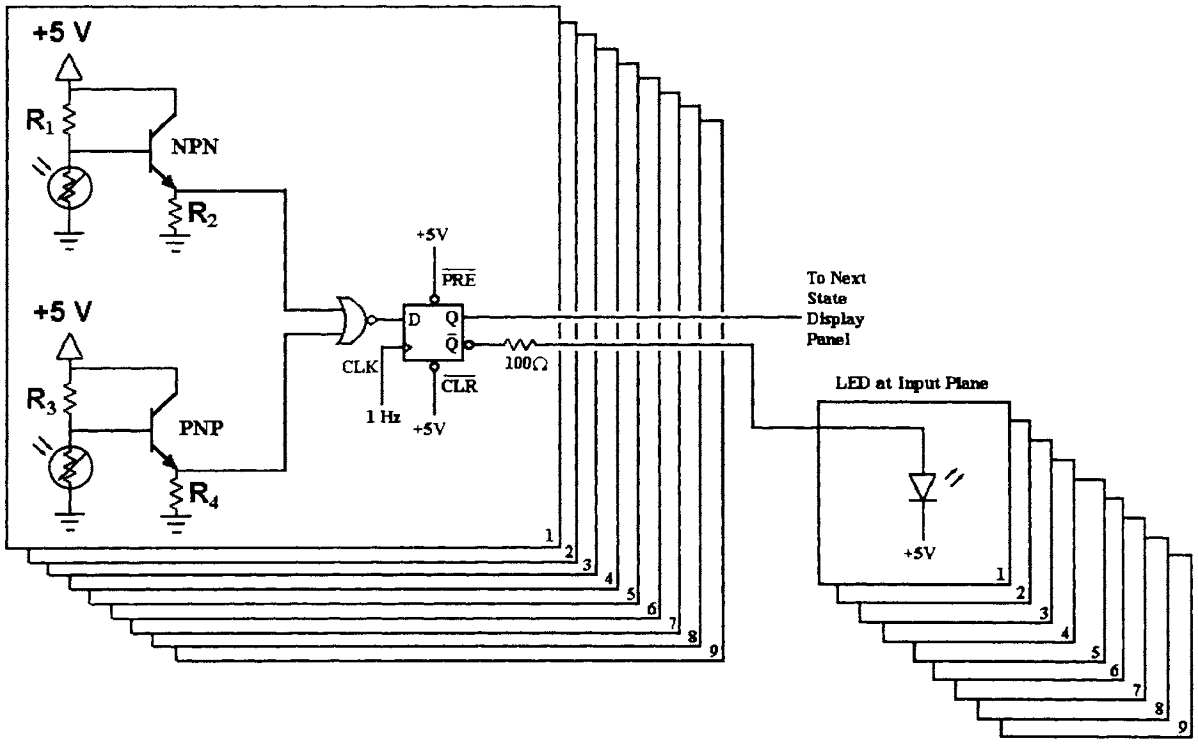

Figure A.2: Circuit Diagram for Final Design

The above diagram shows the feedback circuit used in the final system. The output of the top voltage divider and NPN transistor latches high when the light incident upon the photo cell is a relative intensity of 0 or 1. The output of the bottom voltage divider and PNP transistor latches high when the light incident upon its photo cell is a relative intensity of 1 or 2. The *nor* gate that the two outputs are fed to checks for when both inputs are latched high, thus indicating a relative intensity level of 1. Thus, the circuit performs a *xor* thresholding operation.

Table 1: Comparison of an electronic versus an optical computer.

Electronic	Optical
Serial system	Parallel system
Millions of operations per second	Thousands of operations per second
Mature technology	New technology
Physical limit on speed of today's design	Amount of calculations not dependant on speed of system

BIBLIOGRAPHY

1. S. M. Sze, *Modern Semiconductor Devices,* (John Wiley & Sons, Inc. 1998), 348.
2. Clay Courtney, John D'Amelio, *MacCentral: Under the Hood,* (1997). Available WWW: http://www.maccentral.com/features/110797_hood.shtml.
3. T. Jannson, G. Xu, J. M. Bartha, "Physical Optics Corp. Optical Computer Motherboards," *Precision Plastic Optics for Optical Storage, Displays, Imaging, and Communications: 1997,* Proc. SPIE. **3135**, 206-218 (1997).
4. Mark A. Neifeld and Wu-Chun Chou, ``Information Theoretic Limits to the Capacity of Volume Holographic Optical Memory," *Applied Optics*, **36**(2), 514-517 (1997).
5. Geoffrey W. Burr, Wu-Chun Chou, Mark A. Neifeld, H. Coufal, J.A. Hoffnagle, and C.M. Jefferson, ``Experimental Evaluation of User Capacity in Holographic Data Storage Systems," *Applied Optics,* **37**(23) 5431-5443 (1998).
6. L. J. Irakliotis, S. A. Feld, F. R. Beyette Jr., P. A. Mitkas, and C. W. Wilmsen, "Optoelectronic parallel processing with surface emitting lasers and free-space interconnects," *Journal of Lightwave Technology,* **13**(6), 1074-1084 (1995).
7. J. Von Neuman, editor. *Theory of Self-Reproducing Automata.* (University of Illinois Press, Urbana, IL, 1966).
8. M. J. Murdocca. "Digital optical computing with one-rule cellular automata." *Applied Optics*, **26**, 682-688 (1987).
9. Alastair D. McAulay, *Optical Computer Architectures*, (John Wiley & Sons, Inc. 1991), 416.
10. Paul Callahan, *Paul's Page of Conway's Life Miscellany*, (1998). Available WWW: http://www.cs.jhu.edu/~callahan/lifepage.html.
11. Mitchel Resnik and Brian Silverman, *Exploring Emergence,* (1996), Available WWW: http://lcs.www.media.mit.edu/groups/el/projects/emergence/index.html.
12. Stephen A. Silver, *Life Lexicon,* (1998), Available WWW: http://www.cs.jhu.edu/~callahan/lexiconf.htm.
13. McAulay, as in [7], page 422.
14. Larry J. Hornbeck, Texas Instruments. "Digital Light Processing for High-Brightness, High-Resolution Applications." *Electronic Imaging, EI '97*, (1997).

15. A. Ehrencrona, *Cellular Automata,* (2000). Available WWW: http://cgi.student.nada.kth.se/cgi-bin/d95-aeh/get/lifeeng#what.

VITA

Jason Robert Lane was born January 17, 1974 in Mountain Home, Arkansas. Graduating from Mountain Home High School in 1992, Mr. Lane went on to college at the University of Missouri, Rolla and received his Bachelor's Degree in Electrical Engineering in December of 1997. Upon completion of the requirements for a Master's Degree in Electrical Engineering at UMR, Mr. Lane plans to work at the Naval Air Warfare Center, Weapons Testing Division in China Lake, California.