Optical memory for computing and information processing

Jose Mumbru[a], Gan Zhou[a,b], Xin An[b], Wenhai Liu[a], George Panotopoulos[a], Fai Mok[b], and Demetri Psaltis[a]

[a]Department of Electrical Engineering, California Institute of Technology
MC 136-93, Pasadena, CA 91125
Email: {jmumbru, wliu, gpano, psaltis}@sunoptics.caltech.edu

[b]Holoplex Inc., 600 S. Lake Ave. Suite 102, Pasadena, CA 91106
Email: {gan, xina, fai}@holoplex.com

ABSTRACT

The high data transfer rate achievable in page-oriented optical memories demands for parallel interfaces to logic circuits able to process efficiently the data. The Optically Programmable Gate Array, an enhanced version of a conventional FPGA, utilizes a holographic memory accessed by an array of VCSELs to program its logic. Combining spatial and shift multiplexing to store the configuration pages in the memory, the OPGA module is very compact and has extremely short configuration time allowing for dynamic reconfiguration. The reconfiguration capability of the OPGA can be applied to solve more efficiently problems in pattern recognition and digit classification.

Keywords: Optical memory, Programmable logic, VCSELs, Neural networks.

1. INTERFACING HOLOGRAMS TO SILICON CIRCUITS

Compact optical memory modules[1] posses inherently a high degree of parallelism, since the data that is written into the memory or readout from it is accessed as a page of pixels. Such parallelism results in a large communication bandwidth between the memory and the array of photodetectors during a readout cycle (~10 GByte/sec), or the spatial light modulator (SLM) upon recording (~10 MByte/sec). The use of optical memories in information processing systems makes necessary to consider the interface between the holographic module and the silicon circuitry that processes the data retrieved from the memory and stores the results of the computation in it.

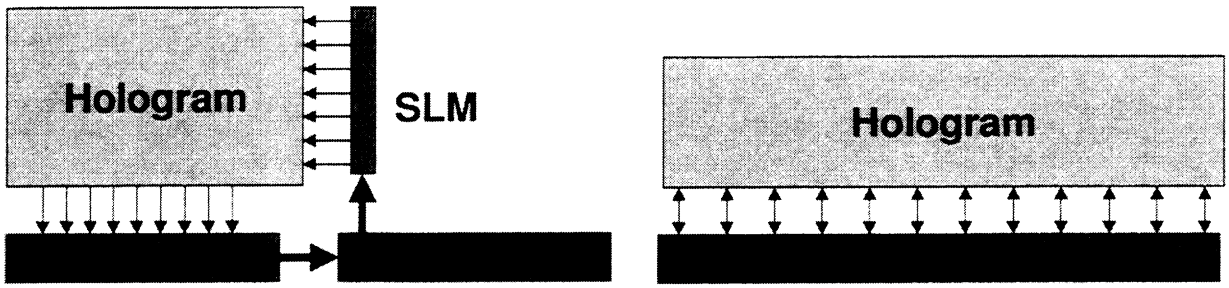


Figure 1. Optical memory-Silicon circuit interface: **(a)** Indirect; **(b)** Direct.

The conventional way to interface both components, holographic memory and silicon circuit, is as shown in **figure 1a**. This architecture suffers from slow detector-logic and logic-SLM communication bandwidth. Even though that the information can be delivered very fast from and to the optical memory, the parallelism is lost in the communication between different chips. Therefore a more effective way to implement such interface is as depicted in **figure 1b**. In this case, a direct interface avoids the slow interchip communication by simply integrating on the same Silicon die the logic circuitry and an array of

Part of the SPIE Conference on Algorithms, Devices, and Systems for Optical Information Processing III • Denver, Colorado • July 1999
SPIE Vol. 3804 • 0277-786X/99/$10.00

photodetectors. However, the question is now to identify which computing devices have enough parallelism built in their hardware as to exchange data efficiently with the optical memory.

2. FIELD PROGRAMMABLE GATE ARRAYS

A Field Programmable Gate Array (FPGA) is also a device with an intrinsic high level of parallelism implemented in its hardware. Thus, it is natural to think of a way to combine this device with an optical memory. An FPGA consists of an array of Configurable Logic Blocks (CLBs) each one of them able to compute a basic logic function. Although different architectures for an FPGA exist, one of the more widely used is the symmetric array[2], **figure 2**. In this case, the CLBs are overlaid in a two-dimensional arrangement and interleaved with vertical and horizontal buses used to establish connectivity among them. Connections between segments in two different buses can also be performed by means of programmable interconnects in switching matrices. Finally, on the periphery of the chip, there are some input/output cells.

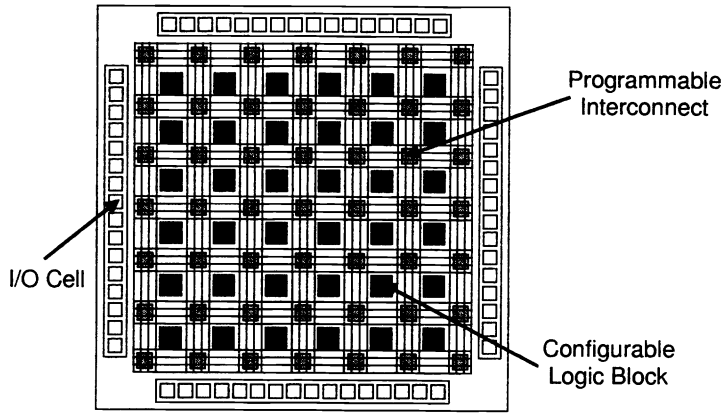


Figure 2. Architecture of a typical FPGA.

These devices have gained popularity due to the fact that they are between a software–oriented solution, like a microprocessor running a program stored in memory, and a hardware-oriented solution, like a specific circuit or ASIC. The FPGA based solution is faster than a microprocessor, speedups of several orders of magnitude have been achieved for some applications, and more flexible than an ASIC. FPGAs contain some hardware resources that can be programmed by the user to implement some given task and, by changing that configuration, the same hardware can be used to perform something totally different.

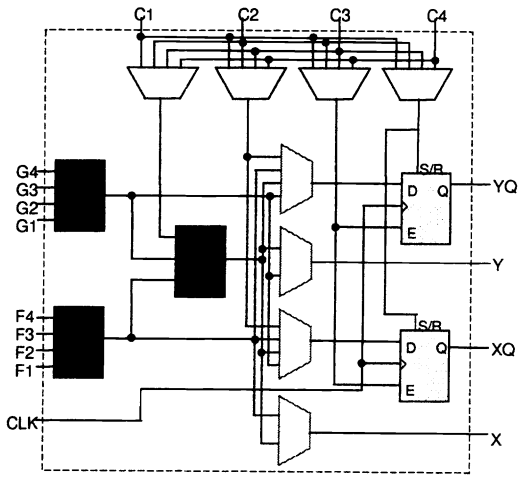


Figure 3. Schematic of a CLB.

The configuration data of the FPGA is stored in an external memory and downloaded into the FPGA chip on demand. Although the size of these devices, in logic gates, can vary among different models and manufacturers, they can contain the order of 10^5 logic gates, which means that the configuration data page can be as large as 1Mbit of information. In terms of Silicon area, the FPGA dies are relatively large, they can be 2cm by 2cm.

2. 1. CLB architecture

The basic building units of the FPGA are the CLBs. Despite the fact that there are CLBs based on other kind of logic blocks, like multiplexors or OR-AND arrays, the use of Look-Up Tables (LUTs) to synthesize logic functions is considered to enjoy of much higher flexibility.[1] A LUT can be seen as a small bank of memory where the inputs encode the address of a position in this memory, which stores the result of a pre-programmed logic function of the inputs. By changing the bits stored in the LUT, the function computed by this is altered.

The schematic of a LUT-based CLB is shown in **figure 3**. In this case, two sets of inputs, on the left-hand-side, feed two independent 4-input LUTs. A third LUT has the ability of combining the results of the LUTs from the previous stage, increasing the functionality of the CLB to implement more complex logic functions. The results of the LUTs are routed through a series of multiplexors governed by some control signals that come from the top of the CLB. The two outputs of the CLB are on the right-hand-side and they can be buffered if necessary by means of flip-flops. These registers allow implementing sequential logic in the CLB.

3. OPTICALLY PROGRAMMABLE GATE ARRAY (OPGA)

Based on this FPGA architecture, the OPGA is a device where the computation is still performed by programmable logic blocks and interconnects as in the conventional FPGA, but where the reconfiguration is brought into the chip optically. This optical reconfiguration capability results from interfacing an optical or holographic memory with a Silicon chip where, in addition to the logic resources, an array of photodetectors has been incorporated, as is illustrated by **figure 4**. The holographic memory can store a large number of configuration templates that can be transferred down to the logic in the FPGA chip in a page-oriented mode. By taking the reconfiguration circuitry out of the FPGA chip, the OPGA can achieve a larger logic density, i.e. more CLBs can be implemented, than in the conventional device.

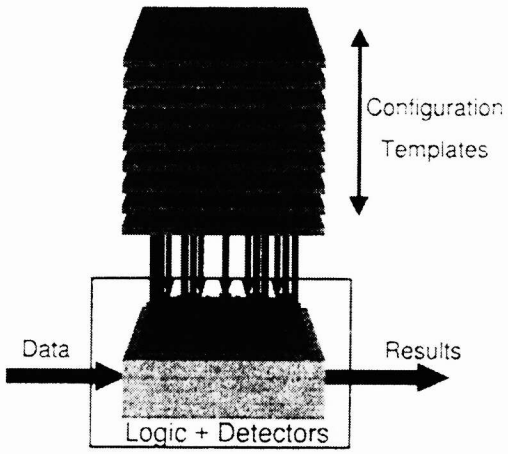


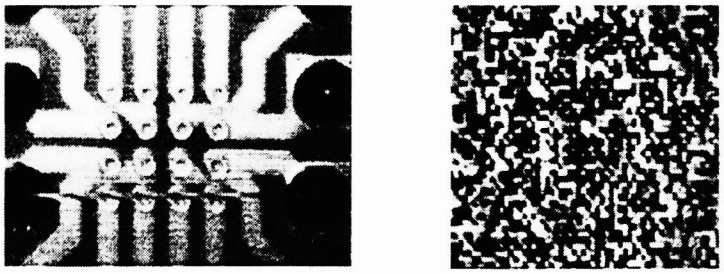
Figure 4. Interface between optical memory and FPGA.

The OPGA is basically the integration of three main components or technologies: an array of VCSELs used to retrieve the templates stored in the memory; the optical memory that contains a large set of configuration contexts; and the VLSI chip that combines CMOS logic and photodetectors. Each one of these components presents a number of issues that need to be discussed in the following subsections.

In its initial implementation, the OPGA module is intended to operate as a Holographic Read-Only Memory (HROM), where a priori and for a given application, the user will decide the library of different configuration templates that needs to be stored in the memory. This alleviates the OPGA module from all the Optics and Optoelectronics required to write in the memory, like spatial light modulator, and makes it very compact. However, posterior OPGA designs will contemplate both read and write capabilities, which will suppose an increase in the computational flexibility of the device.

3. 1. VCSEL arrays

An array of VCSELs is used as light sources for the OPGA module to selectively retrieve one data page of the optical memory at a time. Experiments have been performed using two different types of arrays, a 4x4 symmetric array with 50μm pitch (**figure 5a**) and a 5x1 array with 140μm pitch, both provided by Honeywell. These arrays were tested and characterized to verify if this type of laser diodes is suitable for holographic recording. The elements in the array work in the red region of the spectrum between 675 and 680nm with a dispersion of values for the wavelength of just 0.2nm across the entire array. The VCSELs operate in single mode and the output power is on the range of hundreds of μWatts, from 50μW for the worst element up to almost 300μW for the best one. These elements present very good coherence length, better than a meter, and a small divergence angle. When switching on the VCSELs, the risetime is less than 100ps, which allows switching speeds over 1GHz.



(a) (b)

Figure 5. (a) 4x4 VCSEL Array; **(b)** Hologram reconstructed using the VCSELs.

Their stability over time of the wavelength has also been studied. The wavelength of the light emitted by the VCSELs has been monitored over a period of three hours. In **figure 6**, it is shown the measurements for three different elements in the array. As it can be observed, the behavior is very flat throughout the whole experiment. The dashed lines correspond to the absolute minimum and maximum value of the wavelength measured across the entire array, revealing a high uniformity.

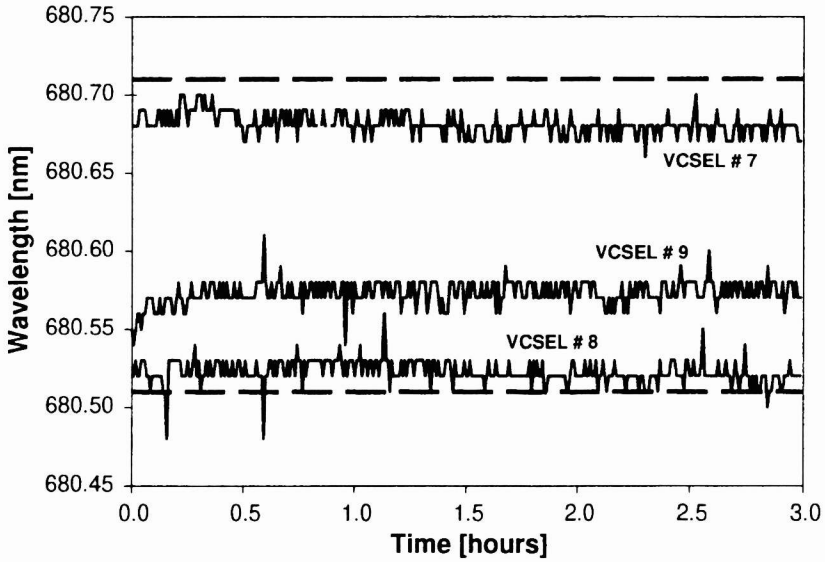


Figure 6. Wavelength fluctuation in time for the VCSELs in the array.

The fluctuation in wavelength has been found to be smaller than 0.016%. This very good stability, even without any thermal control of the VCSEL, makes the VCSELs adequate to record and readout holograms (**figure 5b**) and, therefore, a good choice for the OPGA system.

3. 2. VLSI Silicon chip design

The OPGA chip contains in addition to the logic circuit, as in a conventional FPGA, the array of photodetectors. The detectors must have very small pitch to result in a low area overhead and enough sensitivity to guarantee short integration time. There are basically two topologies to incorporate the photodetectors to the existing logic of the FPGA: sparsely overlay the detectors across the whole chip interleaving them with the logic as in **figure 7a**, or conversely, pack all the detectors in a single large array on a specific region of the chip (**figure 7b**).

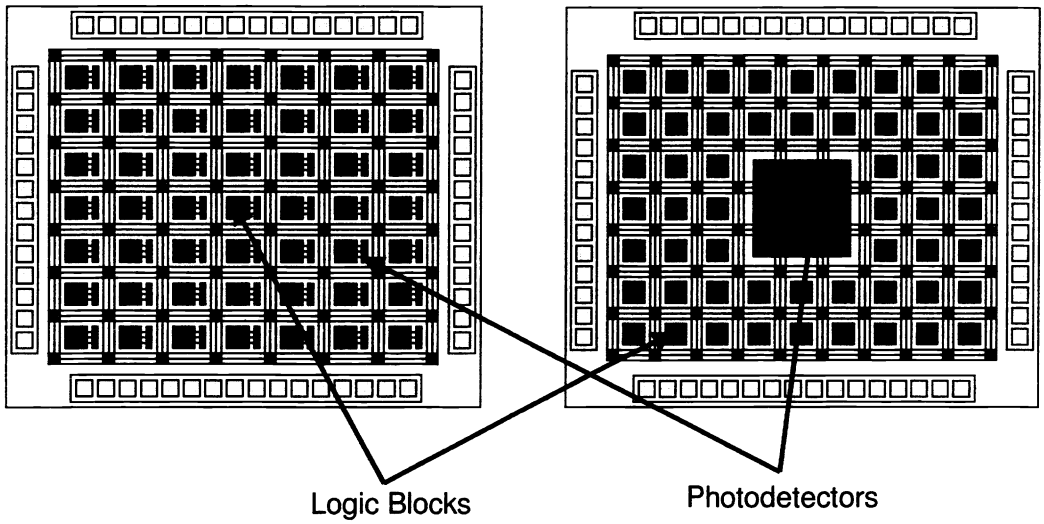


Figure 7. Detector distribution on the OPGA chip: **(a)** Sparse; **(b)** Concentrated.

From the electronics point of view, it is more convenient the first one, so each pixel is detected exactly where it is needed to program the logic element. This makes unnecessary to distribute the detected signals all across the chip. However from the optics side, to have detectors spread over the entire chip means that the quality of the reconstructed hologram must be much more uniform over a large area. Therefore, the second topology makes the optics simpler because the hologram needs to be uniform in a much smaller region. However, this comes at the price of having to implement a more complex mesh of buses to deliver the detected signals to the logic blocks. Therefore, the first topology has been adopted for the OPGA chip. The detectors are implemented in small arrays in the CLBs and interconnect boxes.

3. 3. Holographic memory

The technique used to store and multiplex the holograms in the optical memory determines the architecture of the entire module. For this reason, it is not possible to discuss on the holographic memory without giving a more global view to the system that encompasses both the VCSEL array and the array of photodetectors in the chip.

Mainly due to the limitation in power per VCSEL, we have developed a technique to multiplex the holograms where we could still have short reconfiguration times, in the range of tens of μs, but with a not very demanding requirement on the power per VCSEL. This technique combines both spatial and shift multiplexing.[3] Upon recording, **figure 8**, a lens focuses the beam that impinges the SLM down to a small spot on the recording medium. By changing the angle of incidence of the beam on the lens, the signal spot focuses on a different location in the material, which is partially overlapping with the previous ones. The pages of data are recorded in these partially overlapping circles that span a stripe on the optical material.

To achieve Bragg mismatch among holograms, a converging reference beam needs to be shifted accordingly to illuminate the corresponding signal spot.

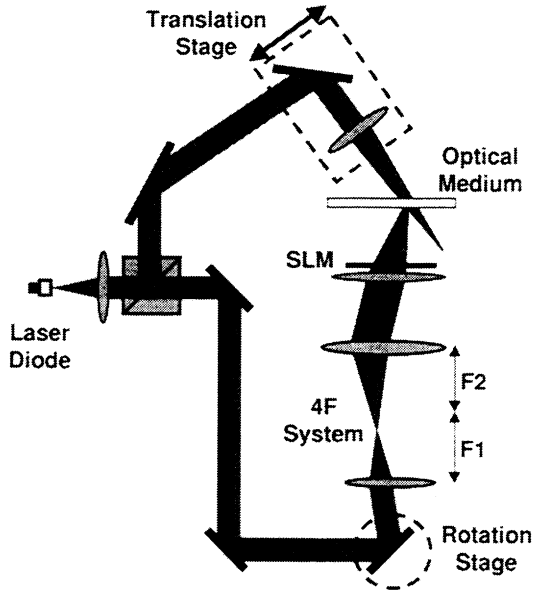


Figure 8. Recording setup combining spatial & shift multiplexing.

In the recording setup (**figure 8**), a laser diode with enough coherence length can be used instead of the VCSEL array. The beam emitted by the diode is collimated and splitted into the signal and reference arm. The signal beam passes through a rotation stage and a 4-F system that changes its angle before it illuminates the SLM. The reference beam is focused by a lens mounted on a mechanical scanner used to translate the beam beyond the shift-selectivity of the optical medium.

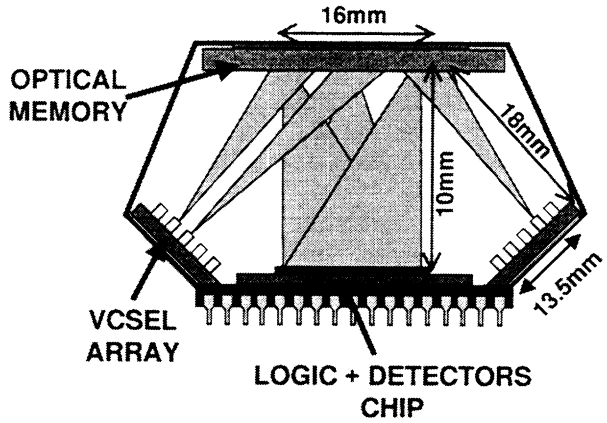


Figure 9. OPGA reader.

During readout, the system becomes very compact (**figure 9**) because of two reasons. First, we use reflection geometry for recording, so upon readout the reading beam from the VCSEL and the array of photodetectors are both located on the same side of the material. Secondly, phase-conjugate readout makes unnecessary the use of any extra component. Placing the VCSEL array at the plane where the converging reference beams used for recording focus, each VCSEL illuminates one of the spots in the memory and all the reconstructed images back-propagate to the plane of the SLM where the photodetector array is upon readout.

As benefit from this architecture, we obtain larger values in the diffraction efficiency per hologram, which scales not as the total number of stored holograms but the number of overlapping ones at any location. A simple system design calculation

helps to illustrate the fact that the power required per VCSEL is compatible with the levels that we have in our VCSEL arrays. Assume that we design the memory to store 100 configuration pages, each page is 1000x1000 pixels with a pitch of 5μm, and we use an optical material 200μm thick with M/5. If the number of overlapping holograms is set to 20, then the diffraction efficiency per hologram is as high as 6.25%. Therefore if we consider the photon-budget at the photodetectors and impose 1000 photons to be detected in order to have an acceptable Signal-to-Noise Ratio (SNR), we can parameterize the power per VCSEL as a function of the integration time of the detectors. If the integration time is set to be just 1μs, each VCSEL must output 6.4mW. If a longer integration time is allowed the power required per VCSEL falls into the range of values of the present VCSEL array (320μW provides in 20μs enough photoelectrons for a good SNR).

Another advantage of this architecture is the small area used on the recording medium to store all the holograms. If the lens that focuses the signal beam has a focal length of 10mm, the signal spot size on the material is just 2.7mm in diameter and 100 holograms can be stored on a stripe 2.7mm wide by 16mm long. Given the small dimensions of the area where the pages are recorded, the holograms are much less sensitive to any non-uniformity on the medium and, consequently, the quality of the reconstructed images is higher.

3. 4. Optical materials

Once decided the mechanism to store the configuration templates in the optical memory, we need to consider which optical media are appropriate for the OPGA system. On the materials side, there are mainly three options: Du Pont photopolymer, Polaroid film and $LiNbO_3$.

In the experiments that have been carried out, red-sensitive Du Pont photopolymer, the HRF-700 series, has been used. This material presents very good dynamic range, M/# can be as large as 5, in proportion to its thickness, between 10μm and 100μm and it also has high sensitivity. For the sake of comparison among the different materials, and due to the non-linearity of the sensitivity with thickness for polymers, a better way to express the recording speed is the exposure energy required to achieve 1% diffraction efficiency. For the Du Pont such energy is about $5mJ/cm^2$. However, the main drawback of this polymer is its poor optical quality due to non-uniformity in the material, which distorts the reconstructed images. This problem becomes more important as the pixel size is reduced, even if phase-conjugate readout is used.

Another alternative is the Polaroid film, which presents properties superior to Du Pont photopolymer, M/10 and $150mJ/cm^2$ exposure energy for a 200μm thick sample and at the same time higher optical quality. The red sensitive Polaroid material is still under development and availability is the main issue. However, the most solid choice seems to be iron doped $LiNbO_3$, which is red-sensitive and presents acceptable M/# (M/4.3 for an 8mm thick 0.02%wt Fe sample). Furthermore, another advantage of $LiNbO_3$ over the other two materials is that it can be used for a re-writable memory.

3. 5. Module packaging

Figure 10. OPGA module package.

The last topic to be discussed is the one regarding the integration of the three major components, VCSELs, optical memory and CMOS chip, on a package. The main goal is that the OPGA module needs to be small enough to be mounted on a board in a computer. The main constrain is in the height of the module, and this depends only on the focal length of the lens used before the SLM. As already discussed, this distance can be made as little as 1cm. The resulting architecture is very compact due to the lensless readout and to the small size of the area of recording medium used to store the holograms. The package

shown in **figure 10** houses the optical memory on the top rectangular window. The VCSEL arrays, integrated on both sides, retrieve the holograms detected on the chip located on the bottom of the package. The package also needs to be robust to ensure the alignment between all the components. It is important to preserve the one-to-one correspondence between the pixels in the hologram and the photodetectors on the chip and also to avoid any change on the areas illuminated by the VCSELs on the optical material.

4. RECONFIGURABLE COMPUTING

Reconfigurable processors make possible to use more efficiently their resources by adjusting themselves depending on the characteristics of the input or on non-satisfactory previous results to better implement the target task.

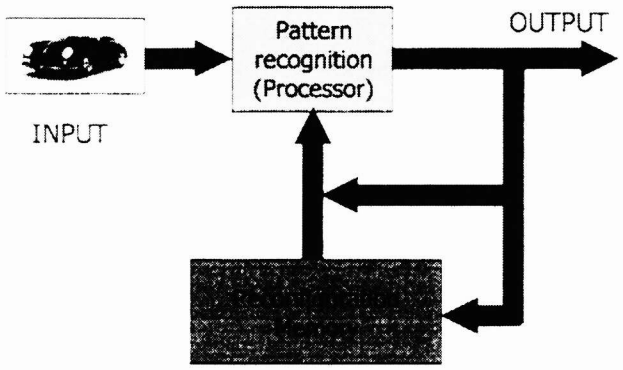


Figure 11. Reconfigurable processor.

Given an application, like pattern recognition, **figure 11**, the reconfigurable processor can be customized to deal with a specific class of objects. It can adapt itself in order to be robust to changes of orientation or illumination of the input object. By reprogramming, the same hardware can be time-multiplexed to carry out sequentially several tasks on the same input, or perform different task to different parts of the same input image. Reconfiguration also possibilities to implement learning by allowing the processor evolve in a controlled manner to learn the function that need to be computed.

5. APPLICATION: CLASSIFICATION OF DIGITS

In this section we present an application of OPGAs in pattern classification using neural networks. When implementing a neural network based classifier the number of units in each hidden layer required to achieve a certain correct classification rate specification depends on the number of output units (i.e. the number of classes). Therefore in a context of limited hardware resources there are problems which cannot be solved using such an approach. In what follows we will use a toy problem to demonstrate the limitations of the aforementioned approach and we will present a strategy based on a tree classifier which overcome these limitations using the flexibility provided by OPGAs.

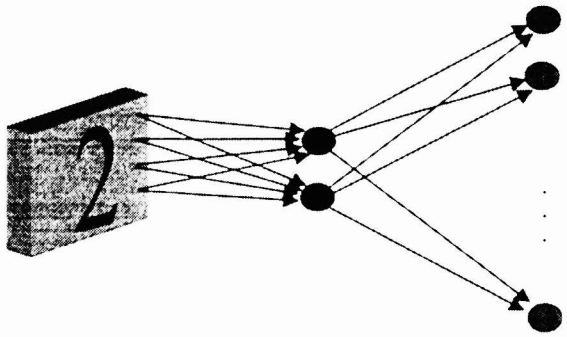


Figure 12. Neural network architecture.

In this problem, we are presented with an 8 by 8 image of a handwritten digit. Our system will classify the digit in one of 10 categories. The database of handwritten digits used in the simulation is composed of a training set of 3823 digits and a test set of 1797 digits. The digits are almost uniformly spread across the 10 categories.

All neural networks used have 64 input units and 2 hidden units, as represented in **figure 12**. The number of output units varies depending on the exact use of the net in the tree. The networks are trained using back propagation with momentum, based on a random initialization of the weights. We use 500 training iterations.

5. 1. Classic neural classifier

To show the limitations in a classic classifier, we first try to classify digits from an increasing number of classes using just one neural network. The correct classification performance obtained is summarized in **figure 13**.

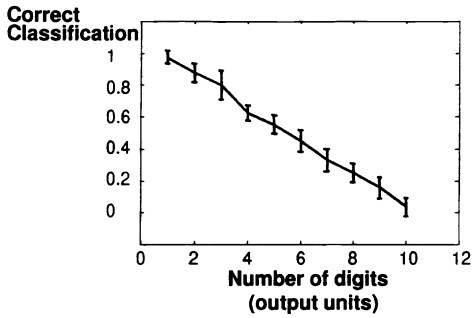


Figure 13. Performance of a single neural network classifier.

We see that the performance decreases rapidly as the number of classes increases, and for ten different classes the performance is only about 10%. To overcome this problem we should either use a neural network with more hidden units (if the hardware resources allow such a choice) or an alternative strategy, like a tree search.

5. 2. Tree search

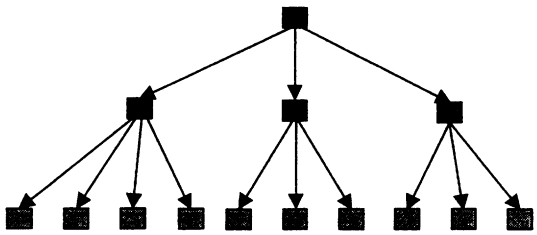


Figure 14. Tree search architecture.

At each node we use a neural net to classify the current digit in one of the subcategories of that node. The OPGA reconfigures itself to implement the neural network in the next node, as dictated by the classification in the current node. At the root node we divide the digits in three categories, namely {1,2,3,8}, {0,5,9} and {4,6,7}. This grouping was made as an attempt to keep maximally correlated digits within the same group, in order to facilitate the classification task of the first layer. At the next level the current digit is classified in one of 3 or 4 categories and again this classification dictates the next classifier to be implemented. The networks of the final layer are used to confirm or overturn the previous classifications. They are trained to recognize only one digit, and a result their correct classification rate is very high (typically above 99%). If the neural network at the leaf gives a positive result (it's output exceeds a certain threshold) then we assume that the classification is correct, and the tree outputs this result. If not we move back to the previous level and follow the path dictated by the next largest outcome of that node. If all leaves of a node of the second level turn out to overturn the classification then

we move back to the root and again follow the path dictated by the next largest outcome. If all output nodes overturn their respective hypotheses we conclude that the input is not a digit.

In order to evaluate this strategy we simulate it. The results of our simulations can be summarized as follows:

Correct classification for the training set	0.97253
Correct classification for the test set	0.91096
Average number of reconfigurations for the training set	3.5124
Average number of reconfigurations for the test set	3.9104

The tree search strategy allows us to decrease the number of steps at the expense of larger memory requirements (order of $n \log n$, where n the number of classes), but this should not be a problem since in OPGAs the reconfiguration storage space is ample.

Using a tree classifier we progressively formulate a hypothesis. At each level of the tree a new component is added to the hypothesis. The current node determines what the next question to be asked is. If a complete hypothesis is rejected we can backtrack and follow a new path, leading to a different complete hypothesis (the rejection is fed back to the previous layer). In more complex classification problems we could use preliminary tests that are more intuitively appealing and use a variety of classifiers in subsequent stages, each of them fine-tuned to make a classification based on the conclusions drawn up to that point.

6. CONCLUSSIONS

In this paper we discuss a suitable way to interface optical memories to silicon circuits that perform some computation. The high level of parallelism in both optical memories and FPGAs makes natural to interface both elements achieving much lower configuration times.

An architecture for the OPGA module has been presented. The module can be made very compact because of the technique to multiplex the holograms in the memory. The characterization of the VCSEL array revealed that VCSELs are suitable for holographic recording and a good choice for the OPGA.

We have applied the OPGA to digit classification. The flexibility of the OPGA allows us to tradeoff between the required area, time and performance. Based on simulation results we further conclude that in order to perform classification using a limited amount of area in a reasonable time the tree search strategy should be preferred, provided that reconfiguration memory is not a scarce asset, as in the case for OPGAs.

ACKNOWLEDGEMENTS

The authors want to acknowledge Arrigo Benedetti for helpful discussions on FPGA in computation. The research is funded by DARPA through Contract F30601-98-1-0199 and by the National Science Foundation Engineering Research Center grant to Caltech.

REFERENCES

1. J. J. P. Drolet, E. Chuang, G. Barbastathis, and D. Psaltis, "Compact integrated dynamic holographic memory with refreshed holograms", Optics letters, Vol. 22, pp. 552-554, 1997.

2. S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, Norwell, 1992.

3. G. Barbastathis, M. Levene, and D. Psaltis, "Shift multiplexing with spherical reference waves", Applied Optics, Vol. 35 No. 14, pp. 2403-2417, 1996.