Optically Programmable Gate Array

Jose Mumbru, George Panotopoulos, Demetri Psaltis
Department of Electrical Engineering, California Institute of Technology
MC 136-93, Pasadena, CA 91125
Email: {jmumbru, gpano, psaltis}@sunoptics.caltech.edu

Xin An, Fai Mok
Holoplex Inc., 600 S. Lake Ave. Suite 102, Pasadena, CA 91106
Email: {xina, fai}@holoplex.com

Suat Ay, Sandor Barna, Eric R. Fossum
Photobit Corp., 135 N. Los Robles Ave. 7th Floor, Pasadena, CA 91101
Email: {suat, barna, fossum}@photobit.com

ABSTRACT

The Optically Programmable Gate Array (OPGA), an optical version of a conventional FPGA, benefits from a direct parallel interface between an optical memory and a logic circuit. The OPGA utilizes a holographic memory accessed by an array of VCSELs to program its logic. An active pixel sensor array incorporated into the OPGA chip makes it possible to optically address the logic in a very short time allowing for rapid dynamic reconfiguration. Combining spatial and shift multiplexing to store the configuration pages in the memory, the OPGA module can be made compact. The reconfiguration capability of the OPGA can be applied to solve more efficiently problems in pattern recognition and database search.

Keywords: Optical memory, Programmable logic, VCSELs, Active pixel sensor.

1. THE HOLOGRAM TO SILICON CIRCUIT INTERFACE

Holographic memories[1] have 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 it 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.

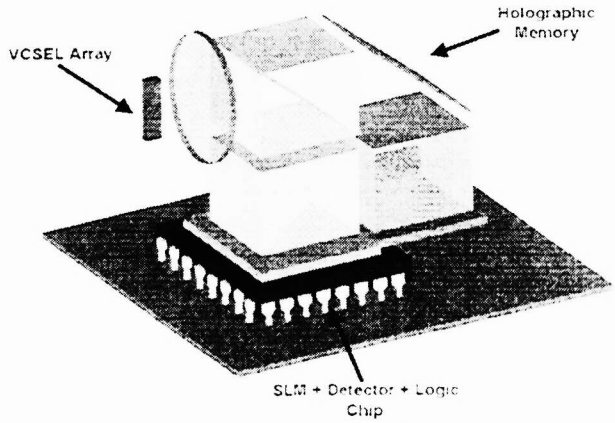


Figure 1. Optical memory-Silicon circuit interface.

In *Optics in Computing 2000*, Roger A. Lessard, Tigran Galstian, Editors,
SPIE Vol. 4089 (2000) 0277-786X/00/$15.00

A specific way to implement such an interface is as presented in **figure 1**.[1] In this case, a direct interface avoids the slow detector-logic and logic-SLM communication bandwidth between chips by simply integrating on the same chip the logic circuitry, an array of photodetectors and the SLM.

2. OPTICALLY PROGRAMMABLE GATE ARRAY (OPGA)

Although our final goal is to have a bi-directional parallel interface between the optical memory and the processor, we have so far looked at the slightly simpler unidirectional case. In this scenario, there exists a computing device, the Field Programmable Gate Array (FPGA), which has enough parallelism built in their hardware as to exchange data efficiently with the optical memory and, consequently, show the benefits of the direct memory-processor interface.

An FPGA consists of an array of Configurable Logic Blocks (CLBs) each one of them able to compute a basic logic function and a mesh of programmable interconnects[2]. The functionality of the FPGA is defined by the configuration data, which is typically stored in an external memory and downloaded into the FPGA chip on demand.

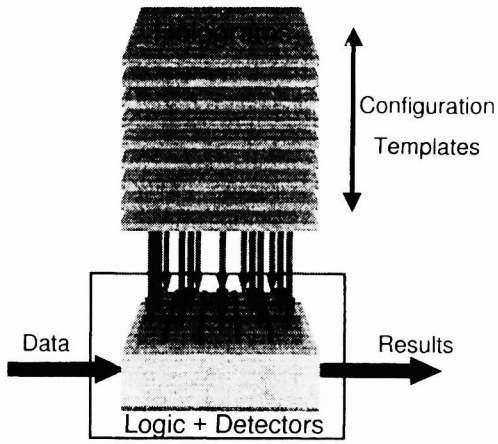


Figure 2. Optically reconfigurable FPGA.

The Optically Programmable Gate Array (OPGA), **figure 2**, results from interfacing a holographic memory[3] to the FPGA. In the OPGA, the computation is still performed electronically, as in the conventional FPGA, but the reconfiguration is brought into the chip optically. The holographic memory can store a large number of configuration templates that can be transferred down to the logic in the FPGA chip in very short time.

The OPGA combines three major components: an array of VCSELs used to readout the holograms; the optical memory that contains the set of configuration templates; and the VLSI chip that contains CMOS logic and photodetectors. Although each one of these components presents a number of issues that need to be discussed[4], in this paper we focus on the silicon chip.

3. VLSI SILICON CHIP DESIGN

The OPGA chip integrates on the same die an array of pixels to detect the reconstructed hologram as well as the logic circuit of a conventional FPGA, as represented in **figure 3**. The detectors must have very small pitch to result in a low area overhead and enough sensitivity to guarantee a reconfiguration time in the range of 1-100μsec.

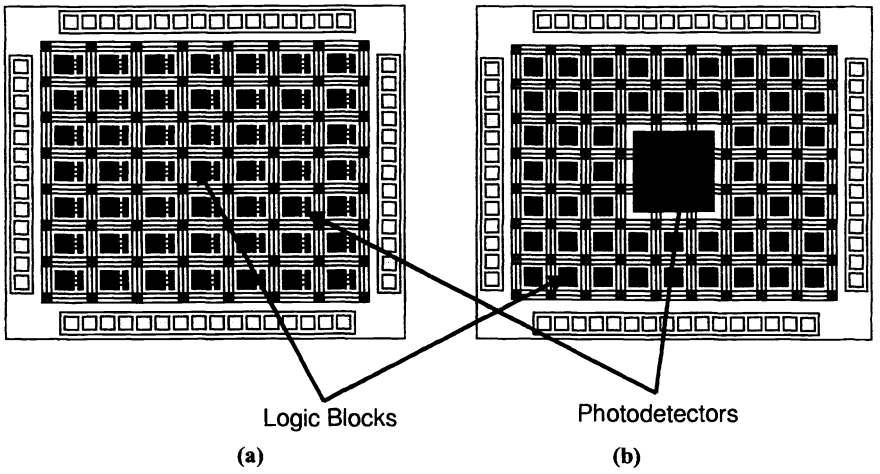


Figure 3. Detector distribution on the OPGA chip: **(a)** Sparse; **(b)** Concentrated.

There are two completely opposite topologies to layout the detectors and the logic. We could sparsely distribute the detectors with the logic, shown in **figure 3a**, so the bits are detected exactly there where they are used to program the logic, or conversely, concentrate all the detectors on a single array and distribute the detected signals across the chip, **figure 3b**.

The second topology makes the optics simpler because the quality of the hologram needs to be more uniform over a smaller area than in the first case. However the first case greatly simplifies the mesh of buses to deliver the detected signals to the logic blocks. Since each strategy has its advantages over the other, they both have been investigated. A chip has been designed integrating the active pixel sensors (APS)[5,6] inside each logic block and programmable interconnect while another design interfaces a global APS array to the logic block array.

3. 1. Differential photodetectors

A passive pixel is interesting mainly because of its smaller size and lower overhead on silicon area compared to the APS[7,8,9]. This was the reason we explored its suitability in the OPGA chip. However, the testing of the very first prototype chip revealed a number of problems that instead motivated the use of the active pixel design. Passive pixels turned out to be very sensitive to blooming, bus charge injection and cross-talk effects. These problems become more dramatic as the operation speed is increased and the pixel pitch decreased. The architecture of the APS is more robust to this kind of problems, so it has been adopted in the next generations of OPGA chips.

The light detected by each APS sensor needs to be converted into a logic value 1 or 0 by comparing it to some threshold. The simplest way to perform such conversion is to set the same threshold to all the photodetectors in the chip. However, a global threshold cannot compensate for spatial variations of intensity across the entire data page. An alternative is to use different threshold levels across the area of the chip. This is not a perfect solution either, even assuming that generating many different bias voltages for the thresholds is not an issue, since this spatial non-uniformity in the reconstructed hologram can change from one holographic page to another.

A very efficient way to be more robust to intensity non-uniformity of the reconstructed data page is differential encoding[10,11]. In this case, each single bit of information required to program the chip is represented by a pair of pixels in the hologram. The differential photodetector has to have two photosensitive areas, referred to as left and right pixels, which need to be matched to the pixel pair in the hologram. The logic 1 is then represented by left pixel on and right pixel off and logic 0 by left pixel off and right pixel on.

This coding scheme makes it unnecessary to set any threshold for the photodetectors. Since the global variation of the incident illumination is reduced, the signal-to-noise ratio is increased and therefore the bit-error rate (BER) is improved. From the optics point of view, this type of data representation is simple and does not increase the system cost.

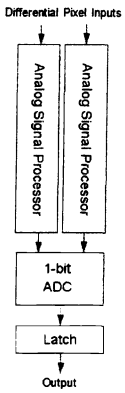


Figure 4. Block diagram of the differential photodetector.

The differential photodetector, **figure 4**, takes the input signals of two APS sensors, and passes them to two independent analog signal processing chains (ASP) that contain the sample and hold circuitry and a first gain stage based on a source follower amplifier. The 1-bit Analog-to-Digital converter (ADC) will produce the logic 1 or 0 to be stored in the latch. The 1-bit ADC is composed of a high-speed amplifier and latch type comparator.

3. 2. Optically programmable logic block

The Optically Configurable Block (OCB), shown in **figure 5**, is basically an optically addressed Look-Up Table (LUT). An OCB has an N-bit block of SRAM to hold the configuration data, as in a conventional LUT, and a 2N-block of pixels connected to a shared differential photodetector to optically load the block of memory. A controller decodes the input signals to access one particular SRAM cell and output its contents, as well as manage the programming of the OCB.

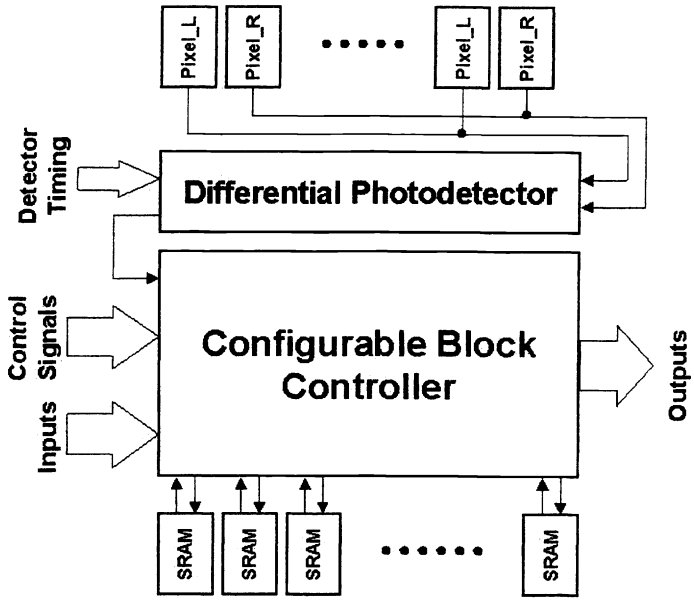
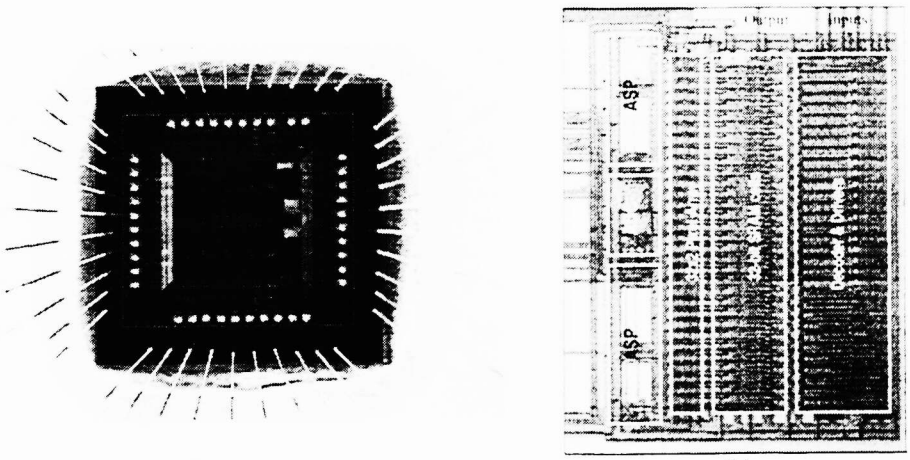


Figure 5. Optically configurable logic block.

The only difference between the OCB and a standard configurable block found in any commercially available FPGA is that it requires extra Silicon area to implement the optical detectors and the wiring required for the timing and control signals of the photodetector.

4. PIXEL LEVEL OPGA CHIP TESTING

A first test chip using APS sensors was fabricated using 0.35μm double-poly, triple-metal (DPTM) CMOS process, **figure 6a**. The chip contains 12 non-interconnected optical blocks. The chip will be used to test the response of the 5μm x 5μm pixel-size differential detectors under uniform illumination.



(a) (b)

Figure 6. (a) First generation APS OPGA chip; **(b)** Optically Configurable Block.

The optically configurable blocks of this chip were designed to implement a 5-input LUT. The picture of one of the block is presented in **figure 6b**. The block consists of a 32x2 array of photodiode pixels, a differential photodetector (ASP chains and 1-bit ADC), a 32 SRAM-cell bank, and the controller composed of a decoder and drivers. The area of the OCB is 180μm x 200μm, out of which about the 30% is due to the optoelectronic circuitry.

In normal operation, the configuration data stored in the SRAM cells is accessed by the inputs through the decoder. During configuration mode, the data is loaded serially into the SRAM cells. The decoder cycles through each one of the pixel pairs to detect its value and stored it into the corresponding SRAM cell. Therefore the full block configuration requires 32 readout cycles. Since one readout cycle takes as short time as 500nsec, the programming of the full block can be done in just 16μsec.

4. 1. Photodetector response

The differential response of the photodetector has been studied. Since uniform illumination is applied to the entire chip, a good way to model spatial intensity variance is to block totally, in some cases, or partially, in others, the pixel openings of the LUT. By reducing the exposed area of the photodiode, we can simulate local differences of intensity and single pixels being off. **Figure 7** is a photograph of the pattern that covered the pixel windows of one of the logic blocks. The gradual covered pixel pairs allow us to test the sensitivity of the comparator in the 1-bit ADC, which was able to correctly discriminate logic 1 from logic 0 in all the cases. From the electronic point of view, the comparator was designed to discriminate differential input signal in the range of 1-2mV.

Figure 7. Window openings on the pixel array of the LUT.

The photon threshold level of the photodetector for single pixels has been measured for different integration times in the range from 6μsec up to 3msec. The threshold level is measured as the number of incident photons that need to be collected in order to have enough signal electrons to detect "pixel on" in the presence of noise due to the pixel dark current. The

experiment has been carried out in both the green (550nm) and red (650nm) wavelengths. **Figure 8** plots the averaged minimum number of photons as a function of the detector integration time. As it can be seen from the plot, the shorter the integration time the fewer photons are needed since there is less time for the noise to build up. For an integration time of 6μsec, about 800 photons are required for green light (lower curve) and 2500 photons are necessary if red light is used (upper curve). As the wavelength is increased, the absorption depth of the incident photons is also increased. Therefore the collection of photons in the red is less efficient than in the green, since there are photons that pass through the 0.3μm diffusion layer and reach the substrate. This is consistent with higher photon thresholds.

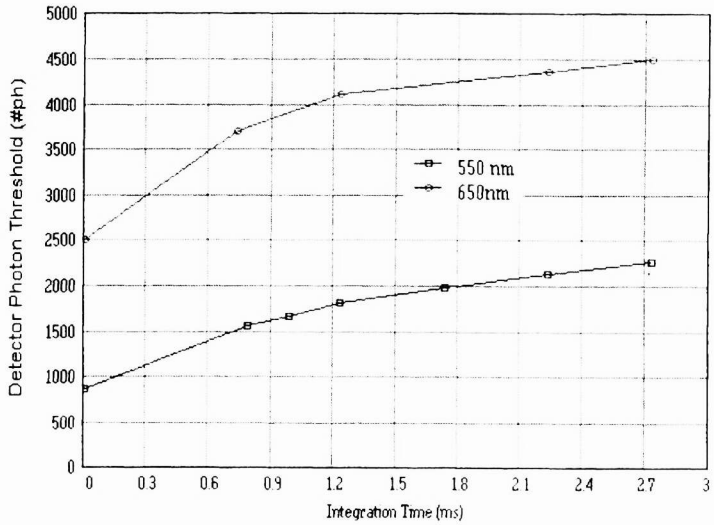


Figure 8. Measured photon threshold level as a function of the integration time.

5. FULL OPGA CHIP ELECTRONIC TESTING

The next generation OPGA chip, also 0.35μm DPTM CMOS process, mimics a small-scale FPGA by having a 2 x 2 array of logic blocks, based on a 5-input LUT, and interconnection matrices to fully interconnect the logic blocks among them and with the I/O buses.

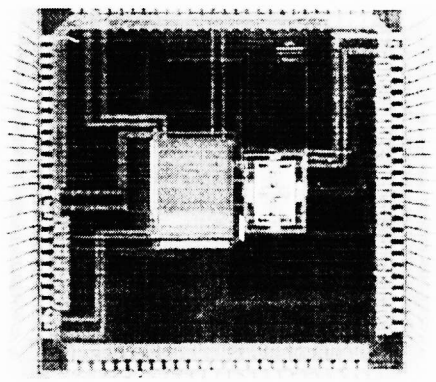


Figure 9. Full OPGA chip: APS array and fully connected logic array.

In the full OPGA chip, the strategy adopted has been to concentrate all the photodetectors in one block separated from the logic array. The chip, **figure 9**, combines a 64 x 32 array of differential APS sensors (big block on the left in the picture) and the logic array containing the logic blocks and interconnecting resources (small block on the right in the picture). The APS pixel size is 15μm, which matches the size of the pixel of our Kopin SLM that will be used to record the holograms in the optical memory.

The programmable logic array, **figure 10**, contains, in addition to the four logic blocks, an interconnection network based on five crossbar switching matrices, and four connection matrices. The crossbar switching matrices connect segments of bus to other segments, while the connection matrices establish connectivity between segments of bus and the inputs of the logic blocks.

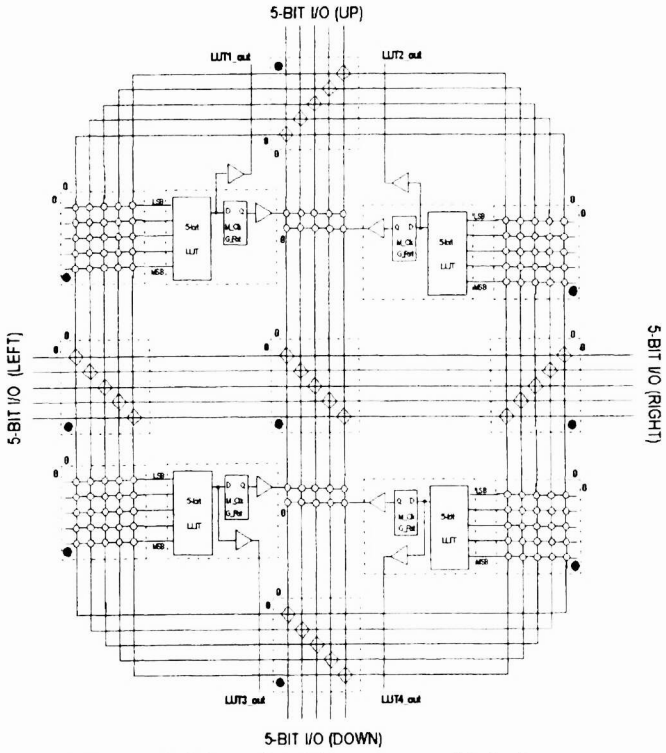


Figure 10. Schematic of the programmable logic array.

5. 1. Interconnection network delay

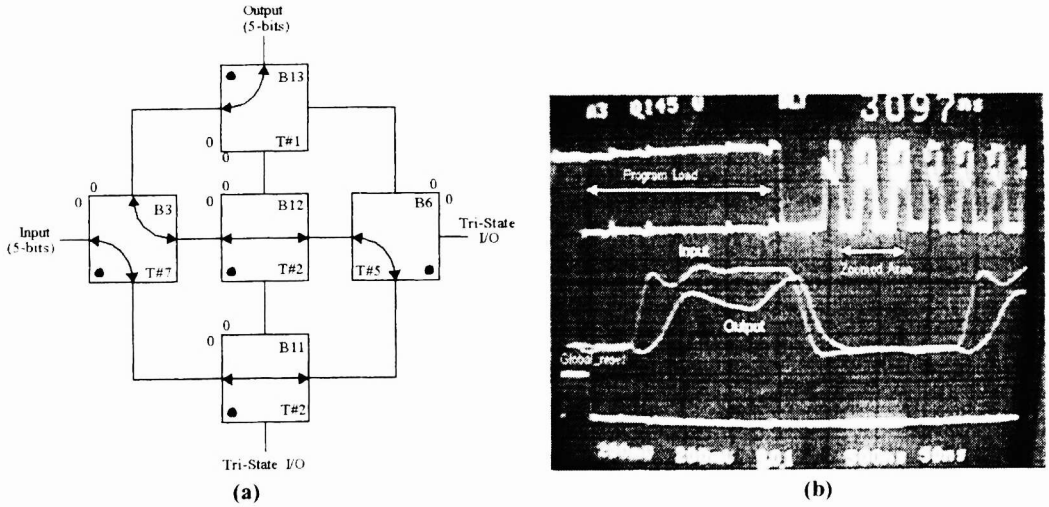
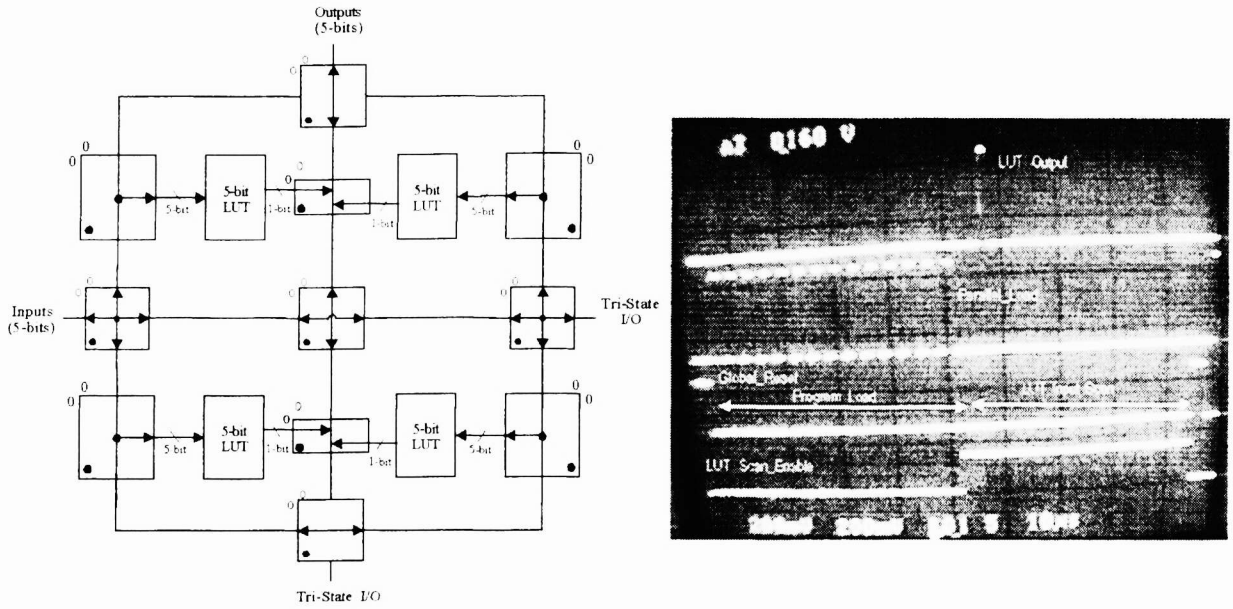


Figure 11. Routing resources test: **(a)** Interconnection network programmed into the chip to route the test signal; **(b)** The experimental scopetrace shows the initialization of the chip (Global_reset), the configuration data being sent into the chip (Program Load). A square wave input signal is then routed through the chip (Input) and the output monitored (Output)

To test the routing resources of the chip, the interconnection path depicted in **figure 11a** is programmed into the chip. In this case, only the crossbar matrices need to be configured. The input signal will be received at the left box, and routed through the bottom, right, central and left boxes and finally delivered to the top box. The test signal, a 30Mhz square wave, will be routed throughout the chip following the described interconnection pattern. The chip is first reset (label Global_reset in **figure 11b**) After that, during approximately the first two microseconds, the configuration data for the five crossbar matrices is loaded. This corresponds to the fast oscillation in the upper part of the scopetrace and labeled as Program Load. Finally the test signal is routed through the chip and the its output is monitored (the 6 cycles of the square wave visible on the right of the scopetrace). The middle part of the scopetrace is a zoomed in version that shows that the output follows the input signal. The input-to-output delay has been measured to be 40nsec.

5. 2. Logic block test

To test the logic blocks, a given test configuration is loaded into the four logic blocks (in this example was 000005hex). The interconnection network is set to route the input signals from the left switching matrix to all the logic blocks and the output of the logic blocks is routed out of the chip through the top switching matrix (**figure 12a**).



(a) **(b)**

Figure 12. Logic block test: **(a)** The inputs of the 4 logic blocks are connected to the left crossbar matrix and the 4 outputs are routed out through the top matrix. **(b)** In the scopetrace, the chip is reset (Global_reset) before the logic block and interconnects are programmed (Parallel_Load). After configuration, the test input cycles through all the 5-bit states (during LUT Scan_Enable, but not shown). The LUT outputs logic 1 for only one state (5hex) and logic 0 for all the others (LUT Output)

In the experiment, **figure 12b**, the chip is first globally reset, then all the logic blocks and interconnects are programmed (during Program Load time). Finally, the input signal scans all the 32 possible states of the LUTs (labeled LUT Input Scan) to observe if its contents agrees with the programmed one. From the experiment the input-to-output delay was measured to be 20nsec, this included the routing delay and the LUT propagation delay.

ACKNOWLEDGEMENTS

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. D. Psaltis, and F. Mok, "Holographic Memories", Scientific American, Vol. 273, No. 5, pp. 70-76, 1995.

4. J. Mumbru, G. Zhou, S. Ay, X. An, G. Panotopoulos, F. Mok, and D. Psaltis, "Optically Reconfigurable Processors", *A Critical Review 1999 Euro-American Workshop on Optoelectronic Information Processing*, P. Refregier, B. Javidi, Vol. CR74, pp. 265-288, SPIE, 1999.

5. S. Mendis, S. E. Kemeny, and E. R. Fossum, "CMOS active pixel image sensor", IEEE Trans. Electron Devices, Vol. 41, No. 3, pp. 452-453, 1994.

6. E. R. Fossum, "Active pixel sensors-Are CCD's dinosaurs?", *CCD's and Optical Sensors III*, Vol. 1900, pp. 2-14, SPIE, 1993.

7. G. P. Weckler, "Operation of p-n junction photodetectors in a photon flux integration mode," IEEE J. Solid-State Circuits, Vol. SC-2, pp. 65-73, 1967.

8. R. Dyck and G. Weckler, "Integrated arrays of silicon photodetectors for image sensing," IEEE Trans. Electron Devices, Vol. ED-15(4), pp. 196-201, 1968.

9. P. Noble, "Self-scanned silicon image detector arrays," IEEE Trans. Electron Devices, vol. ED-15(4), pp. 202-209, 1968.

10. J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Volume Holographic Storage and Retrieval of Digital Data", Science, Aug. 1994, pp. 749-752.

11. D. Psaltis, G. W. Burr, "Holographic Data Storage", Computer, Feb. 1998, pp. 52-60.