

Designing and Testing The Optical Mouse

Richard F. Lyon, Fairchild Advanced R&D¹
Martin P. Haerberli, Xerox Palo Alto Research Center

A mouse is a pointing device used with interactive computer systems (see Figure 1). The user moves the mouse on a work surface, usually next to the keyboard, and the movement of the mouse is translated into cursor motion on the screen. Mice have recently become available on the office-products market as a part of the Xerox "Star" (8010 Professional Workstation), as well as in other markets, as part of the Xerox D1100 Scientific Workstation, the Symbolics and LMI Lisp machines, the SUN workstation, among others.

Electro-mechanical mice were first developed in the 1960s at Stanford Research Institute, and are described in Newman and Sproull (1973), and Englebart *et al.* (1967). The original mouse used a pair of wheels turning potentiometer shafts to encode X and Y motion into analog signals.

The mouse was redesigned at Xerox to use ball-bearings as wheels, and to use optical shaft encoders to generate a two-bit quadrature signalling code (see Figure 2). The mouse was again redesigned to use a ball instead of two wheels, eliminating the drag of side-slipping wheels.

In Xerox research, the mouse has been in popular use for over eight years, and has been preferred over other pointing devices (Card *et al.* 1977). However, it has not been outstandingly reliable. The balls or wheels can get dirty, and slip on the pad rather than roll; or the commutators can get dirty, and skip. This is a major maintenance problem with workstations that use

¹Formerly with Xerox Palo Alto Research Center.



FIGURE 1. Optical mouse.

K.O. Beckman

the mouse in an uncontrolled environment. Another disadvantage of the electro-mechanical mouse is that it's expensive; it contains many precision-machined parts, and requires many intricate assembly steps.

In June 1980 we started work on an optical replacement for the electro-mechanical mouse. The design was submitted for implementation in late July, and chips were returned and successfully tested in December 1980. With the help of several interested colleagues, this chip was incorporated into the standard mouse housing.

These optical-mouse prototypes confirmed the validity of the concept. The next step was to introduce several improvements which would lead to a fully functional, manufacturable device. There is no good way to test the chip in volume production. The sensor spacing and the optics in the prototype-mouse housings we built did not let us to generate quadrature signals at the same rate (pulses per inch) at which they are generated by the PARC mechanical mice. We believed that the photosensitivity and regularity of the chip could be improved by separating the logic from the sensors. We began to make these

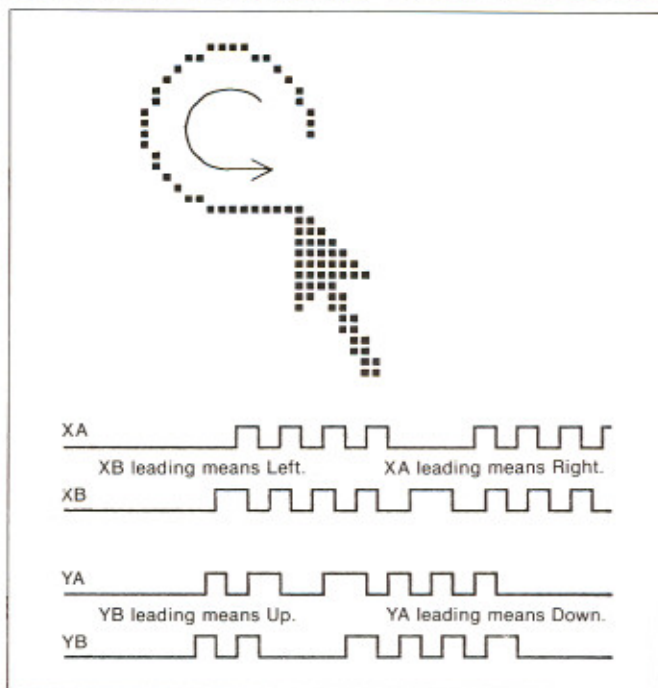


FIGURE 2. Quadrature encoding of pointer motion on a bit-map display system.

improvements in February 1981. By June, the design was ready for implementation; parts were returned and tested in July.

In short, a mouse with no moving parts was built using innovations in electro-optics, circuits, geometric combinatorics, and algorithms, all combined in a single custom-nMOS integrated circuit (patent pending). The chip uses several interesting layout, circuit, and timing styles that are widely applicable. (Lyon 1981[-1], Lyon 1981[-2])

Design Goals

The primary design goal was to develop a device which is much more reliable than current electro-mechanical mice. Such a replacement for an electro-mechanical mouse **must**:

- Be plug-compatible with existing mechanical mice.
- Have no moving parts (not counting button switches, if any).
- Work under a wide range of process parameters.

It could also include on-board switch debouncers, because we had to provide them anyway, and the silicon area required was cheap. Optical mice require a special patterned mouse pad, but this restriction is not too severe, because most users of ball mice use a special pad anyway to increase the friction on the ball.

The Imager and Motion Tracker: A Smart Digital Sensor

The optical mouse combines two novel concepts to make a one-chip imaging and tracking system:

- The first concept is a simple "mostly digital" circuit that produces digital-image (bit-map) snapshots of bright features in a dark field, using self-timed circuit techniques and mutually inhibiting light sensors.
- The second concept is a tracking algorithm, which uses the digital imager to form pictures of an easy-to-track contrasting pattern, and a digital machine that uses images of that pattern as input and tracks relative image motion.

The notion of a digital imager and tracker applies equally well to linear or two-dimensional sense arrays.

The Digital Imager

Because it was easily available at Xerox, the nMOS integrated circuit technology was chosen to implement the optical mouse chip. (Other technologies, such as pMOS, CMOS, or bipolar, could have been used instead.) In nMOS, when light strikes the circuit side of a chip, the photons are converted to hole/electron pairs with reasonable quantum efficiency (see Figure 3). The holes are generally attracted to the negative-biased p-type silicon substrate, whereas the electrons are attracted to n-type diffused source/drain regions and channel regions (Séquin and Tompssett 1975). Thus, light is detected by collecting negative charge (electrons). If a node is isolated by a turned-off transistor, it is said to be a "dynamic node." A dynamic node which has been charged to a positive voltage will "leak" to a lower voltage as light is received. An imager is simply an array of subcircuits, with a dynamic node in each, which can watch the declining voltages and make a sensible bit-map image from them.

Therefore, each imager pixel (sub-circuit or cell) contains a dynamic node, a transistor to "reset" it high and then isolate it, and an "inverter" circuit to sense the voltage of the node and to

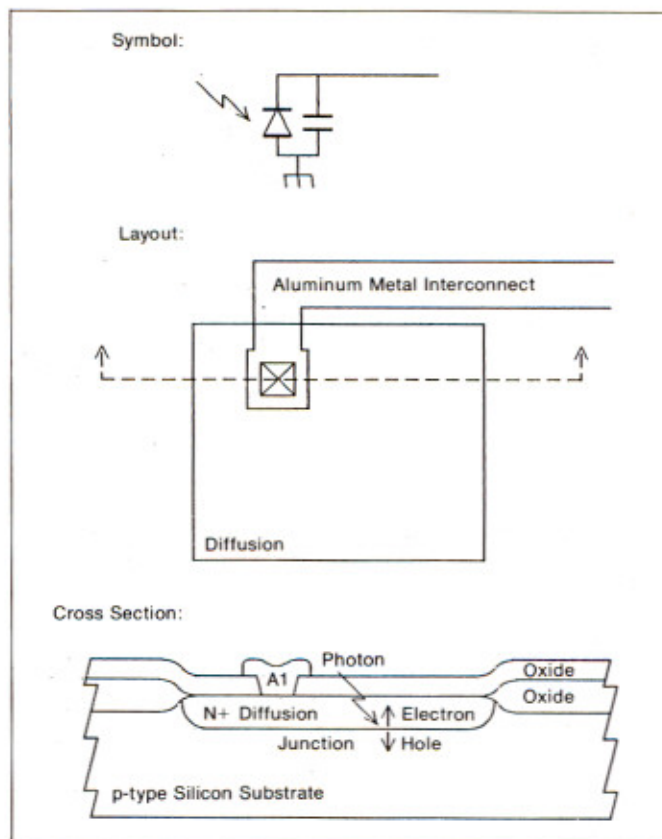


FIGURE 3. An nMOS photo-diode.

communicate it to other circuits. The output voltage from the inverters will start LOW when the array is reset, then go toward HIGH as the corresponding dynamic nodes go LOW due to light. Figure 4 is a schematic diagram of this simple "analog" imager cell.

Initially, such an array of analog imagers has a digital all-LOW output. Then, it has an interesting analog image for a while; but it ends up in the digital all-HIGH state, until it is reset. Both of its digital states are uninteresting. We needed a reliable way to get an interesting digital bit-map image. One way to do this was to implement a form of "inhibition" between cells, so that after some cell outputs have gone HIGH, all others are held LOW, and the picture is stable from that time. This is somewhat analogous to the lateral inhibition in the retina of most biological vision systems (von Bekesy 1967). It has the desirable effect of producing sensible images, almost independent of the light level. Such digital sensor-arrays can be built in a self-timed loop of logic that recognizes stable images, latches them, resets, and starts over, at a rate roughly proportional to the light intensity.

The simplest imager with *mutual inhibition* is the two-pixel system shown in Figure 5. Each pixel circuit is essentially a NOR-gate. One input of each is the light-sensitive dynamic node, and the second input is the output of the other cell. The initial reset state is 00, with outputs being pulled LOW by the NOR inputs that are connected to the initially HIGH dynamic nodes. The final state can be either 01 or 10, because 00 will decay with time, and 11 cannot result from cross-coupled NOR gates.

The presence of the initial state (READY) is sensed by a NOR gate whose threshold is much lower than the threshold of the pixel NOR gates. READY can be used to tell the timing logic

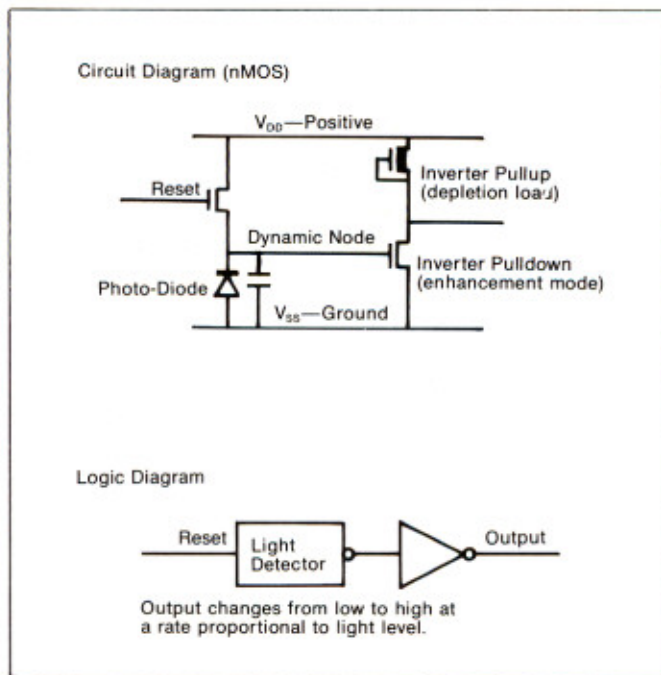


FIGURE 4. Simple "analog" imager cell.

driving RESET that the imager has been sufficiently reset. The existence of a final state (DONE) can be sensed by an OR gate whose logic threshold is higher than the thresholds of the pixel NOR gates. Intermediate and metastable states will have both output voltages near the NOR gate thresholds, but clearly below the OR gate threshold. Therefore, this two-pixel digital imager compares the light level at two points, and indicates when it has made a decision. (However, there is no bound on how long it might take to make a decision, even in bright light.)

The set of possible stable images can be enumerated for any pattern of inhibition and for any shape and size image array. Using a four-by-four sensor array, with inhibition of cells up to 2.9 or more pixels away, it is easy to formulate a simple and reliable tracking algorithm that follows spots in a hexagonal array, and represents their motion with a quadrature code.

In the mutual-inhibition detector array, the cells race to see which can be the first within a neighborhood to get enough light and inhibit the others.

The inhibition network is defined by choosing an inhibition neighborhood for each cell. Generally, we choose neighborhoods symmetrically, such that if A inhibits B, then B inhibits A; we say A "is coupled with" B, reflecting the cross-coupled NOR structure. For example, each pixel may be connected with its eight neighbors in a square grid, resulting in nine-input NOR gates. In many cases, the inhibition neighborhood of some cells will be all the other cells in the array. (CELL-DONE signals from such cells will be redundant, but may be implemented just for the convenience of layout regularity.)

Trackers

The simplest application that illustrates the digital imager/tracker is a linear-motion sensor. Such a sensor can be built from a row of imager cells looking at narrow white lines, about one-third white (approximately orthogonal to the row of imager cells), on a dark background. If four imager cells are used, and if we arrange for each cell to inhibit cells up to two steps away (say, cells at distances of less than 2.5), then we get a set of three

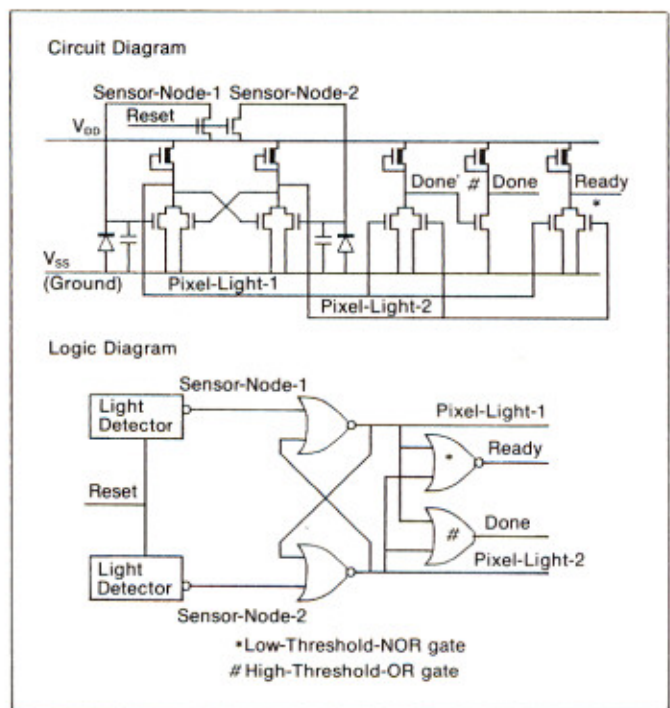


FIGURE 5. Two-pixel digital imager.

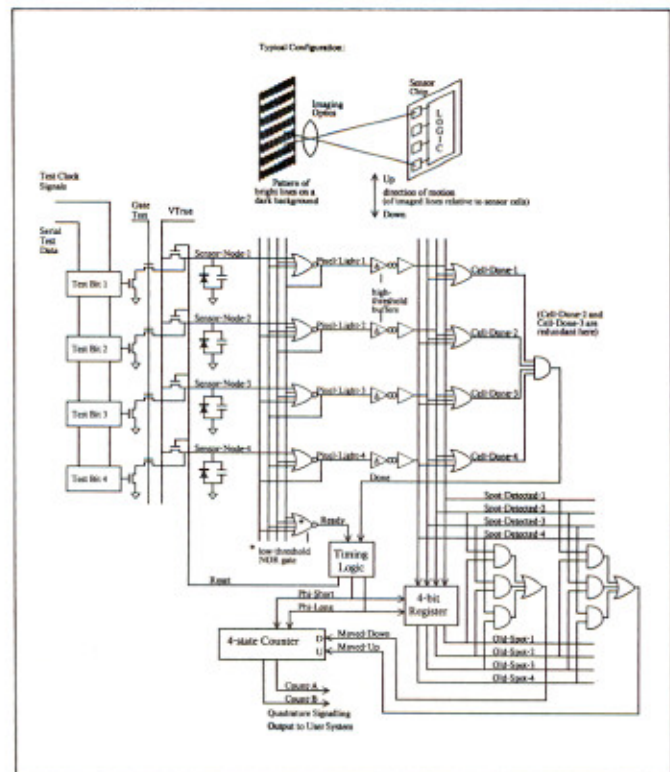


FIGURE 6. Linear motion detector including generalized done/detect, and testing circuitry.

stable images:

1 0 0 1 0 1 0 0 0 0 1 0 (radius 2.5 inhibition)

If the white-line spacing (imaged onto the chip) is about three cell widths, then these images correspond obviously to positions of the bright lines relative to the cells (1=bright). Figure 6 shows a simple digital machine (on the same chip) that

Inhibition Neighborhoods	Inhibition Radius	Stable Images and how many of each	Total Images
	<1	1	1
	1.1	42	42
	1.5	79	79
	2.1	43	43
	2.3	25	25
	2.9	21	21
	* 3.0s	30	30
	3.1	26	26
	3.2	14	14
	3.7	14	14
	4.3	16	16

* "3.0 special" was chosen for the optical mouse, its images are simply characterized as either a central dot or a pair of dots on opposite edges but not sharing an edge.

FIGURE 7. Inhibition neighborhoods and stable images for a four-by-four sensor array.

would compare the current image with the previous image, and output a signal that says *moved up* or *moved down*. Thus, we have a relative motion sensor for one dimension of travel.

The advantage over previous optical quadrature detectors is that no fixed threshold or specific light-level is needed. The sensors will cycle at a rate depending on the light level, and latched outputs will be made available to the host system.

How can we characterize inhibition patterns? One simple method, used in the previous example, is by the "radius of inhibition," or the radius of the circle around a given sensor which contains the inhibited neighbors. Figure 7 shows a taxonomy of stable patterns generated by different inhibition radii, for two-dimensional sensor arrays.

The general tracking concept is to use a hexagonal array of white dots (which looks like dots of constant spacing but no particular orientation, when seen through a small window at an arbitrary angle), and to pick a dot-spacing such that bit-map images can be easily associated with the dot array, and such that movement can be detected easily by comparing successive snapshots. The white-dot spacing should be slightly more than the inhibition radius. Using "3.0 special" inhibition with a four-by-four sensor array, we recommend a dot-spacing of about 3.4 pixels, because that is the average distance between dots in the stable images with two dots. Thus, the dots in the stable images correspond obviously to the positions of one or more dots in the hexagonal dot-array (see Figure 8).

Counting all rotations and mirrorings, there are 30 distinct stable images for the "3.0 special" inhibition. Of the 900 combinations of two successive stable images, most have an obvious interpretation in terms of movement of the white dots

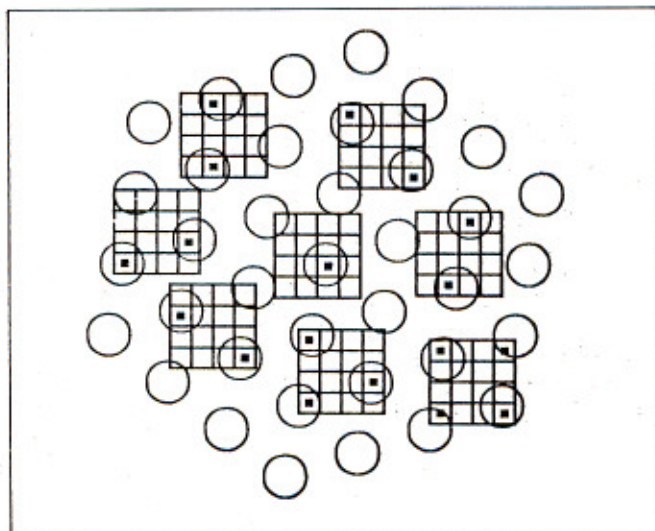


FIGURE 8. Various positions of 4x4 imagers with respect to a hexagonal dot array, showing ways to see all the possible stable images for inhibition with radius 2.9 or more.

with respect to the imager; those that do not must be handled by the tracking algorithm, but will probably not occur often. Clearly, we can build a finite-size machine that takes a stable image as input, examines its current state (the previous input), and produces signals to control X and Y quadrature counters. For details, see Lyon (1981[-1], 1981[-2].)

The Self-Timed Action of The Imaging-Tracking System

Timing logic unites the imager and the synchronous finite-state tracker machine in a self-timed system.

The timing logic generates two-phase, non-overlapping clock signals to run the digital logic, so that each cycle is synchronized to the reset-done cycle of the imager. This same clock runs an up-down counter controlled by the tracker PLA for each of X and Y, to generate quadrature signals which can be sent off the chip. So we had three things to design, whose parts are not interesting in themselves: DONE and READY detectors, a clock-and-reset signal-generation circuit, and an up-down counter with quadrature outputs. (These parts, and the logic-level and timing details of the clocking circuit, as shown in Figure 9.)

Clocks are generated through a delay-independent (self-timed) handshake with the imager array, and it is assumed that the digital logic is fast enough to keep up with the imager. The generated clocks are called "Phi-long" and "Phi-short," to indicate which one is of unbounded length. Phi-long should be used to enable quasi-static feedback, so as to keep the logic alive and insensitive to light while waiting for the imager.

This technique doesn't care how slow the imager is; everything is willing to wait till there is a solid digital answer. The imager should receive enough light to cycle faster than once every few hundred microseconds, on the average, so that it will get image samples often enough to track mouse motion of several thousand steps per second.

The First Mouse-Chip Layout

The first optical mouse was 3.468mm by 4.360mm (15.12mm²) in a typical nMOS process ($\lambda=2.5\mu$ microns, or 5-micron lines). A photo and floorplan are included in Lyon

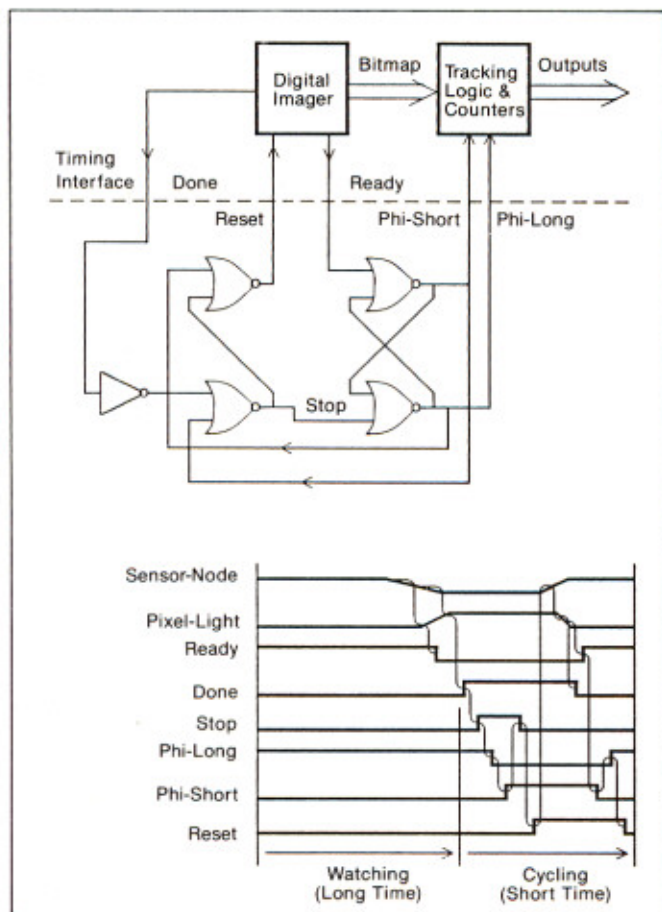


FIGURE 9. Imager and logic tied together by self-timed clock circuit, with timing waveform diagram.

(1981[-1]).

There is a single layout for a programmable sensor and a logic cell. This layout can be customized for each position in the array, to implement any inhibition pattern and the tracking algorithm described above. The logic to detect a stable image is also partly programmable, and distributed. Figure 10 shows the cell layout with programming for the upper left position.

The layout used in this first version of the chip treated a sensor cell with its logic and memory as a low-level cell, and built the array by selectively programming the cells in different positions. This approach requires large amounts of wiring area, because every cell has to have access to every other cell's PIXEL-LIGHT line.

Another notable layout feature is the regular structure used for the "random" timing logic. It is essentially just like one plane of a PLA, except that it can also be programmed with contacts between the lines running orthogonally through it. With a bit of optimization, this becomes topologically identical to PL-style gate-matrix layout.

The benefits of this approach were clear: chip-design time was very short (about 1 person-month); standard switch-level simulation could be used to verify the correctness of the circuits; the first implementation worked; several orders of magnitude of light-level variation were tolerated, and the techniques developed were very resistant to process parameters and temperature.

The idea of using lateral inhibition to make a digital imager was conceived in June 1980. We developed the rest of the

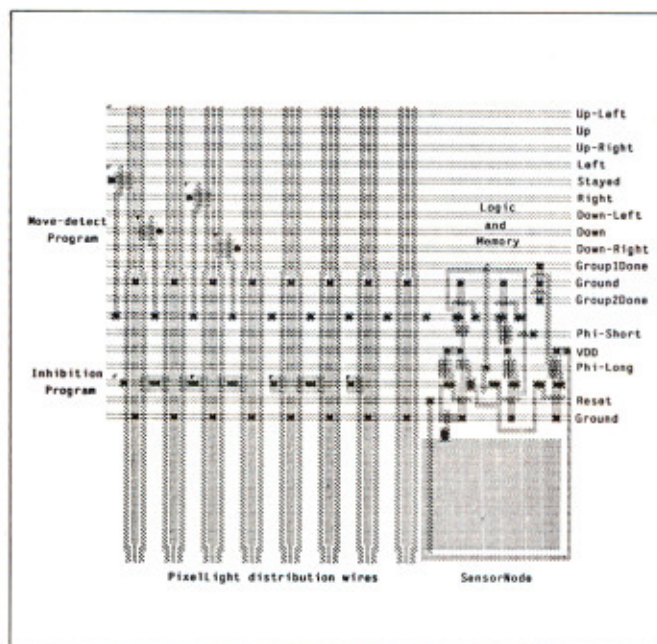


FIGURE 10. The layout of the upper-left optical mouse cell.

techniques discussed here while writing up the inhibition idea, in June and July 1980. A chip design was done quickly in the latter part of July, and was debugged by hand crosschecking of the layout against design sketches. After the chip was implemented, our tools for design-rule checking, circuit-extraction, and simulation became more available, and the design was verified as correct (except for some non-fatal design-rule violations).

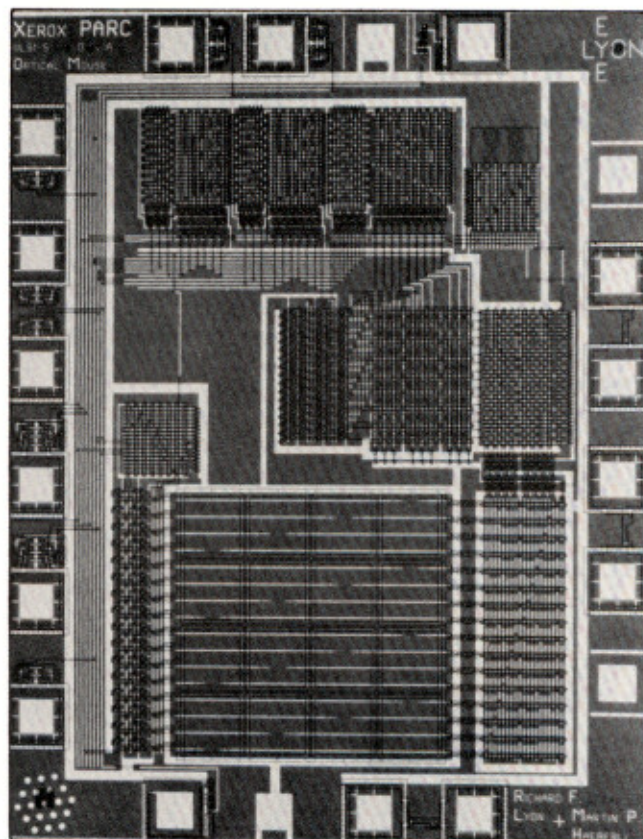
At first, the area penalty due to distributing the logic through the sensor array was not seen as a problem. Then we realized that the wiring in the sensor array required us to place the sensors far apart, which meant a greater magnification in the optical system. The light-sensitivity of the sensor nodes is proportional to the flux, and the flux decreases with the square of the magnification. To improve the light sensitivity, and to adjust the sensor spacing to generate the correct number of motion steps per inch of mouse motion, we switched to a new layout style, using a densely packed array of $N+1$ diffused areas as sensor nodes, with all the logic in compact regular structures outside of it. (This new layout is discussed further below.)

Testing

Finished optical mice had to be tested, to make sure that they were correctly built, to cull out those which suffered from fabrication errors, and to gather yield data.

We were restricted to testing our chip at the pins (it is difficult to reach into the chip and twiddle internal signals, even optically.) One approach might be just to plug each chip into an optical-mouse housing, connect it, and wave it over a mouse pad. This technique works fine for small quantities; in fact, we used it for the first few chips we built. But this process is very labor-intensive, unsuited for volume production, and says nothing useful about test coverage.

Suppose that the chips are tested in the dark. Then, if we can only simulate the effect of photons hitting the sensor array in predetermined ways, we can force the chip into any state we like. All we need is a way to pull selected sensor cells to ground, as if light had hit them, while leaving others intact.



Meigar Photography



FIGURE 11. a) Photo of the improved optical mouse chip. b) Floorplan of the improved optical mouse chip.

In the revised design, a semi-static serial shift register was constructed next to the sensor array. During testing, a test pattern was shifted into this test register. Once the test register was loaded, a test control signal, GATE TEST, was pulsed. Any 1 in the test register caused the corresponding sensor to be pulled down; a 0 left that sensor cell intact.

The chip was completely tested by repeatedly loading the test register and strobing it to the sensor array, while monitoring the X and Y quadrature outputs. The problem then became one of devising the correct sequence of test patterns, in order to maximize test coverage, and of predicting what the correct X and Y outputs were in response to the test sequence.

The sequence of test patterns we used was merely all possible pairs of patterns, in succession. Because the inhibition pattern we used has 30 stable images, there were 30×30 , or 900, steps in the test sequence. If the chip behaved as expected through these 900 steps, and if its debouncers worked, then it was declared "good."

We tried to predict the X and Y quadrature response to a given input sequence by applying the test stimuli in parallel to the chip under test and to a logic simulation of the mouse. But this technique suffered because most logic simulators were too slow, and because thermal noise injected new images into the chip faster than the tester could. Therefore, we abandoned this approach, and implemented a simple simulator for the optical mouse chip in Mesa (Mitchell *et al.*). The test program could then stimulate the real chip and the simulated chip at once, and compare the results in real time.

Each chip was tested completely on the wafer, so that no effort was wasted packaging bad chips. The chips were retested after packaging, to detect parts damaged in that process. Our

testing approach is similar to a number of methods described by Frank and Sproull (1981).

Verification of New Designs

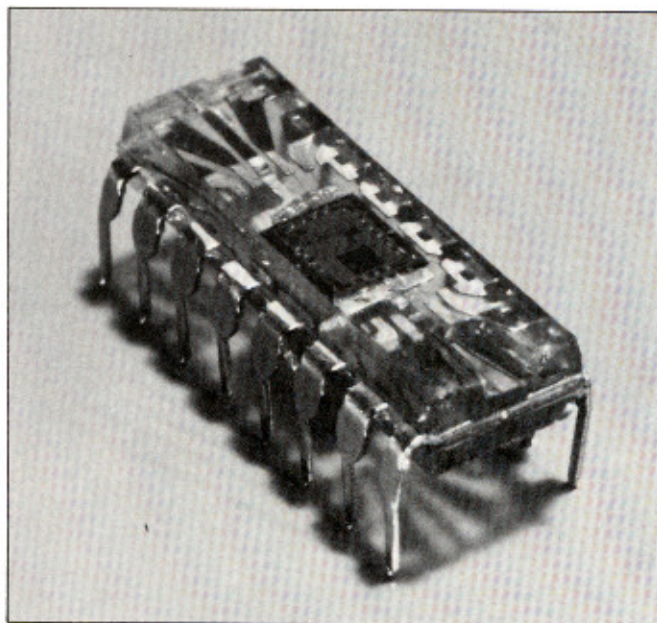
For the optical mouse, we already had a Mesa simulation and a test-sequence generator. The test-sequence generator was connected to a logic simulator driven by a circuit description extracted from the chip geometry. In parallel, it drove the Mesa simulation of the mouse. Wherever there was an error in the geometry, the Mesa simulation and the logic simulation disagreed, and we could solve the problem *before* the chip was implemented.

The interface to the logic simulation and the Mesa simulation of the optical mouse was the same as the interface to chips under test: through simulated pads at the edge of the chip. Therefore, we were verifying the test circuitry as well.

The Improved Mouse Chip Layout

The second version of the optical mouse took much from the first: the tracker and counter PLAs, the timing logic, and the switch debouncers were identical. The sensor array was rearranged to remove the inhibition logic and the tracking logic, which were laid out separately, in a regular pattern. Testing circuitry was also added previously described. (A floorplan and a photo of the new chip are shown in Figure 11.) This chip is 3.2mm by 4.07mm (13.02mm²) at $\lambda = 2.5\mu$, or about 14% smaller than the previous version.

The inhibition and done/detect logic were laid out using a version of the regular structure for random logic. The done/detect logic was just a copy of the inhibition program, laid out on a different pitch. The previously distributed logic for



Meigar Photography

FIGURE 12. The optical mouse chip in a clear injection-moulded plastic package.

feeding the tracker with motion signals was redesigned as a programmed bit convolver, similar to a barrel shifter.

Packaging

Because the finished optical mouse had to fit inside a mouse housing, we selected a 16-pin package. One of the ways to get light onto a chip in such a package is to use a ceramic package with a quartz window. This is quite expensive. Instead, we packaged the chip in a clear injection-molded plastic package. (See Figure 12.)

The mouse housing was adapted from the standard PARC ball mouse housing; the top half and buttons stayed the same. The innards of the mouse were completely new. A PC board to mount the switches, lamps, and chip was laid out from a vector-board prototype, and an optical assembly was designed to direct the light from the lamps to the paper and thence to the chip.

Summary

The optical mouse embodies several ideas that are not obvious extensions of standard digital or analog design practices, but which contribute to the design of robust analog-to-digital sensors. Using the lateral-inhibition concept, sensor cells that are trivial and useless by themselves become powerful together.

Fortunately for this project, the nMOS technology we know and love for logic is also well suited for sensing photons; once the ideas and algorithms were firm, the chip design was relatively routine, and quick-turnaround implementation was available through the standard path.

Whatever chip one is designing, it pays to consider beforehand how to test the chip and verify the design. In the case of the optical mouse, a very small investment in chip area and complexity brought great benefits in testability.

The optical mouse is just one example of an application for smart digital sensors, combined with several stages of logic. Others can be imagined, for character recognizers, edge detectors, light-controlled oscillators, etc.

A complete optical mouse has been in use for many months, with only one minor problem: when, having become accustomed to the optical mouse, one is forced to use a workstation with an electro-mechanical mouse, one finds that the performance of the latter is annoyingly erratic.

References

- Card, S.K., W.K. English, and B. Burr. April 1977. "Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT." *Ergonomics*, Vol. 21, No. 8
- Englebart, D.C.; W.K. English and M.L. Berman. 1967. "Display-selection techniques for text manipulation." *IEEE Transactions on Human Factors*, HFE-8, 1, and 5.
- Frank, E.H. and R.F. Sproull. February 1981. *Testing and Debugging Integrated Circuits*. Carnegie-Mellon University, Computer Science Report, CMU-CS-81-105.
- Lyon, R.F. 1981 [-1]. *The Optical Mouse, and an Architectural Methodology for Smart Digital Sensors*, Xerox PARC Technical Report VLSI-81-1.
- Lyon, R.F. October 1981[-2]. "The Optical Mouse, and an Architectural Methodology for Smart Digital Sensors." *CMU Conference on VLSI Systems and Computations*, Conference Proceedings, Pittsburgh, Pennsylvania. H.T. Kung, R.F. Sproull, G. Steele, editors, Computer Science Press.
- Mead, R.A. and L.A. Conway. 1980. *Introduction to VLSI Systems*. Reading: Addison-Wesley.
- Mitchell, J.G. W. Maybury, and R. Sweet. April 1979. *MESA Language Manual*. Xerox PARC report, CSL 79-3.
- Newman, W.M. and R.F. Sproull. 1973. *Principles of Interactive Computer Graphics*. McGraw-Hill.
- Séquin, C.H. and M.F. Tompsett. 1975. *Charge Transfer Devices*. New York: Academic Press, Inc.
- von Bekesy, G. 1967. *Sensory Inhibition*. Princeton University Press.

About the Authors

Martin P. Haerberli studied computer science and electrical engineering at the University of Wisconsin from 1971 to 1974 before joining Techmation N.V., in Amsterdam, The Netherlands, from 1974 to 1976. At Techmation, he was a senior systems programmer, responsible for developing, installing, and supporting operating systems and related software for all of Techmation's European sites. He then joined Bolt Beranek and Newman in Cambridge, Massachusetts, where he maintained and developed multiprocessor communications and systems software for in-house and ARPA networks. In 1978 he joined Xerox in Palo Alto, where he developed and installed advanced prototype office systems. Since January 1981, he has been a member of the research staff in the VLSI Systems Design Area at Xerox PARC, where he has been active in implementation systems and VLSI design. His current interests include designing for testability, self-timed systems, and wafer-scale systems architecture.



Dick Lyon received his BS in engineering and applied science from California Institute of Technology in 1974, and his MS in electrical engineering from Stanford University in 1975. He enjoyed a variety of experiences in digital system design, communication and information theory, analog and digital signal-processing, and real-world hardware development in jobs at Bell Labs, Jet Propulsion Labs, and Stanford Telecommunications, Inc. In 1977, he went to work on speech recognition based on custom VLSI architectures, in the VLSI System Design group at Xerox Palo Alto Research Center. Among other VLSI-related tasks there, he did the optical-mouse work described in this article. Since August 1981, Dick has been with the Artificial Intelligence Research Laboratory of Fairchild Advanced R&D as project manager for speech, where he continues to work on improving man-machine communication.

