A reprint from

Optical Engineering

26(9), 916-922 (September 1987).

ISSN 0091-3286

CHARGE-COUPLED COMPUTING FOR FOCAL PLANE IMAGE PREPROCESSING

Eric R. Fossum

Columbia University
Department of Electrical Engineering
1312 Seeley W. Mudd Building
500 W. 120th Street
New York City, New York 10027

Charge-coupled computing for focal plane image preprocessing

Eric R. Fossum, MEMBER SPIE
Columbia University
Department of Electrical Engineering
1312 Seeley W. Mudd Building
520 W. 120th Street
New York, New York 10027

Abstract. A new class of charge-coupled devices called charge-coupled-computing devices is described. These analog circuits perform arithmetic functions such as addition, subtraction, and magnitude comparison in the charge domain. The circuits are compact and are designed to be insensitive to rail voltages, simplifying their utilization. These devices, in conjunction with input, output, and analog memory circuits, can be combined to form a simple but general-purpose and fully programmable charge-coupled computer. A prototype charge-coupled computer has been fabricated and tested. Prospects for forming a large array of computers (e.g., 1000 to 10,000) on a single chip for spatially parallel image preprocessing are discussed. Such image plane preprocessing of data would find use in real-time mobile robot vision systems, in which low power, lightweight computing is critical for economical viability.

Subject terms: charge-coupled devices; image processing; focal plane arrays; machine vision; autonomous vehicles; signal processing; analog computers; parallel computation.

Optical Engineering 26(9), 916-922 (September 1987).

CONTENTS

1. INTRODUCTION

Real-time vision in machines requires the synthesis of imaging hardware, computing hardware, and image processing software. For mobile robots, the system must be portable. These mobile robots may be used in the future for underwater inspection, agriculture, space exploration, and transportation in addition to obvious defense applications. Thus, the vision system must be lightweight and require low power as well as having the high throughput necessary for achieving real-time operation.

In the image preprocessing portion of the vision system, the tasks performed include smoothing (noise removal), level shifting (fixed-pattern-noise removal), gain adjustment (sensor nonuniformity compensation or adaptive contrast enhancement), sharpening, edge enhancement, thresholding, frame-to-frame subtraction, motion detection, and region growing. These tasks can be performed (although not always optimally) using local neighborhood operations on image array picture elements (pixels). Generally, these preprocessing functions consume the greatest portion of the time required by the vision process.

In the course of effecting the preprocessing tasks, each pixel may undergo as many as 100 to 500 simple arithmetic operations per frame. The frame rate may range from 1 Hz in low speed systems, to 100 Hz in systems operating on a par with human vision, to 1000 to 10,000 Hz in high performance systems. Thus, the number of operations required in a 100 Hz frame rate system with a modest array size of 100×100 elements easily could exceed 1×10^8 arithmetic operations per second. Advances in very high speed integrated circuits (VHSIC) may allow serial computing systems to achieve such high throughput, but a parallel computing approach (single instruction, multiple data) can be used to alleviate the performance requirements. Unfortunately, massively parallel computing systems are presently incompatible with the portability needs of a mobile robot system. Furthermore, the advantages of massively parallel systems often are offset by the problem of loading and unloading the parallel data from a serial data stream at sufficiently high data rates.

In this work, an advanced approach is advocated. It is proposed to perform the image preprocessing functions in parallel on the image plane itself, similar to a biological retina. Since the image data arrive in a parallel manner and are transduced to an electrical form in parallel, it seems natural to perform spatially parallel image preprocessing on the image plane as well. By contrast, conventional imaging hardware transfers the image data off the plane by multiplexing it into a serial data stream (using charge-transfer devices) with little or no modification of the input data. The

Invited Paper CD-112 received Dec. 18, 1986; revised manuscript received Jan. 6, 1987; accepted for publication Jan. 27, 1987; received by Managing Editor June 23, 1987.

serial data bottleneck is one of the most serious problems facing conventional approaches to real-time imaging processing. With processing being performed on the image plane, not only are the throughput requirements of subsequent signal processing alleviated, but the opportunity exists for selectively reading only relevant data off the image plane, thus further reducing serial signal processing and transmission bottlenecks.

The computation on the image plane may be done in a number of ways. For example, conventional or bit-slice digital architectures might be employed. However, the real estate available for such computation is critically small, and the need for a parallel array of analog-to-digital converters will make such an approach particularly unattractive. Thus, a real-estate-efficient *analog* computer would obviate the need for A/D conversion prior to processing.

The degree of parallelism present on the image plane determines the real estate available for each processing element (PE). Ideally, a truly spatially parallel architecture, in which there is one PE for each pixel, would be used. For monolithic devices, the real estate consumed by the PE circuitry takes away from the real estate available for the photon transducer itself. This can be a serious problem if the light levels are low or the frame rate is high. In a hybrid "bumped" detector/readout system, the image resolution is not necessarily hindered by the density of the PE circuitry, but if the number of pixels exceeds the number of processors, then a lower degree of parallelism must be accepted. In future 3-D integrated circuits, the PE density can be increased if the PE real estate can be spread into the Z direction.

Charge-coupled-device structures are well suited for such a system.[1] The analog nature of the image data can be compactly represented in the charge domain, requiring a single electrode for storage. The image data are refreshed at the frame rate, so the dynamic nature of CCD signal representation is generally not a concern. The accuracy of computation in the charge domain, which can be of the order of one part in 256 or better, is more than adequate for most image preprocessing applications. Furthermore, charge-transfer devices already have established themselves as the technology of choice for image data readout. The difficulty lies in the design of the charge-domain circuits.

There recently has been a growing interest in performing image preprocessing on the image plane in the charge domain. Joseph et al. at Honeywell reported on a parallel image processing (PIP) chip that performs charge-domain operations on images.[2] A second version of this chip is still undergoing testing. Beaudet* at Westinghouse recently reported on the design of a general-purpose focal plane charge-domain convolver chip. Sage† at MIT Lincoln Laboratories reported on the use of conventional CCD imagers, operated unconventionally, to perform convolution operations on image data. Related works include serial data stream image processing in the charge domain[3] and charge-domain circuits for signal processing,[4] as reported by Vogelsong and Tiemann.

*P. Beaudet, presented at 1986 IEEE Workshop on Charge-Coupled Devices (Harriman, N.Y.), unpublished.

†J. P. Sage, presented at 1986 IEEE Workshop on Charge-Coupled Devices (Harriman, N.Y.), unpublished.

2. CHARGE-COUPLED-COMPUTING CIRCUITS

There are several issues to be concerned with when designing and implementing arithmetic functions in the charge domain for image plane processing applications. First, the circuit must be as real estate efficient as possible. Second, it can use only simple biasing; that is, all clock voltages must swing between the same two rail voltages, and the circuit operation should be insensitive to these rail voltages. Third, the circuit must have enough dynamic range for the application, so in general small signal approximations cannot be used to describe device behavior. Finally, the circuit must have the required accuracy.

One may note that doing computation in the charge domain essentially limits one to positive signal quantities. However, since image data are essentially a representation of the local photon flux, the input data are always greater than zero, and there are no intrinsically negative signals to be concerned with. Intermediate negative quantities cannot be used easily and should be avoided in the design of algorithms. It also is possible to use a sign bit to represent negative quantities, but this often is cumbersome.

The approach taken in this work is to use three-dimensional charge coupling. In conventional charge-transfer devices, capacitors are coupled laterally through the use of fringing fields generated by bias voltages applied to the electrodes, as shown in Fig. 1(a). The circuits advocated here also use vertical coupling between the charge on the electrode and the charge in the channel. This is illustrated in Fig. 1(b). The approach is similar to that used in floating-diffusion output amplifiers in conventional CCDs except that the discharged electrode controls a CCD potential well instead of a transistor current. The sequence is as follows: an electrode is precharged to a voltage ($-V$) by means of a transistor switch and is then left floating. Signal charge collected by a floating diffusion attached to the electrode by means of a wire (metal, polysilicon, diffusion, etc.) is "subtracted" from the precharged electrode. The resultant change in potential well depth is used to generate or control charge packets.

Such 3-D coupling has several advantages. Primarily, it offers a new set of possibilities for novel circuit design since both electrodes and the semiconductor are used in the charge domain. A second advantage is that charge can be transferred across long distances in a short time through the use of conventional wiring without requiring sequential electrode transfers. Thus, the planar topological constraint normally associated with CCD circuits is alleviated. Third, the precharged gate can be used as a summing node or for current integration.

There are problems introduced by this method as well. For example, if the wiring capacitance is not carefully minimized, charge-transfer efficiency can seriously suffer. Switched capacitor circuit designers face similar difficulties. Another problem is that gate charge subtraction based circuits are susceptible to kTC noise, which may dominate the noise floor.

The simplest arithmetic function implemented in the charge domain is addition. The preferred approach is the use of a summing bucket or accumulator. Charge packets are successively inserted into the summing bucket, and then the sum is transferred when complete. In this circuit, and in

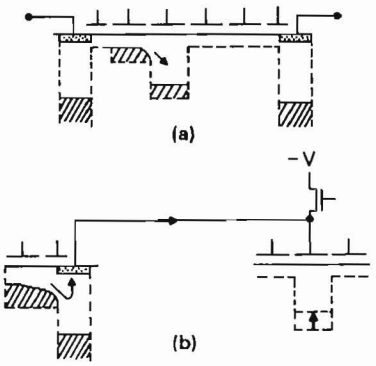


Fig. 1. Illustration of three-dimensional charge coupling. (a) Conventional lateral coupling for charge transfer. (b) 3-D coupling using gate charge subtraction. The circuit operation is described in the text.

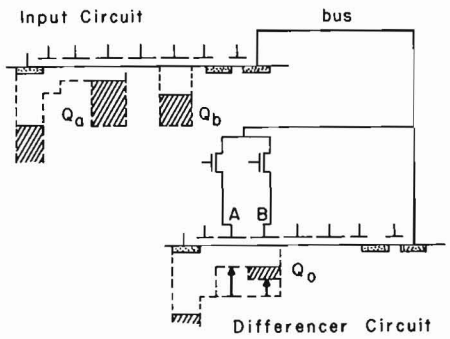


Fig. 2. Linear charge-packet-differencing circuit. Charge packets Q_a and Q_b are subtracted from the gate charge on electrodes A and B, respectively. The operation is discussed in detail in Ref. 6.

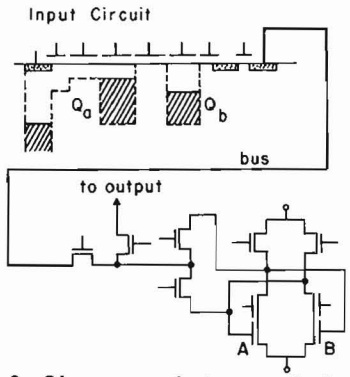


Fig. 3. Charge-packet-magnitude comparator. Charge packets Q_a and Q_b are used to preset node voltages A and B, respectively, of balanced flip-flop.

succeeding circuits, the dynamic range of the operation is determined by the size of the electrode area and the voltages applied. However, since all buckets swing between the same clock rail voltages, electrode area is the only real variable. A summing bucket therefore, by its nature, should be larger than average. Increasing the electrode lengths can impair circuit speed since the charge transfer time scales as L^n, where n is 2 or larger. Hence, there is a trade-off between dynamic range and speed of operation. The accuracy of the accumulator is limited by the input transfer efficiency, which depends on clock width. In general, accuracy is also traded for speed.

A simple function, although difficult to implement accurately, is time-invariant signal attenuation, wherein the output charge packet is a fixed fraction of the input charge packet. Previous workers have tried various approaches, but the most successful is that reported by Bencuya and Steckl,[5] which uses a channel stop barrier to divide a bucket in the transverse direction (parallel to charge flow). Partition noise is minimized by this technique, and the accuracy is determined by photolithography. Repeated application of charge-packet splitting and summation can be used to effect variable attenuation[4] of charge packets.

A more difficult function to implement compactly is charge-packet differencing. Fossum and Barker[6] reported on an intrinsically linear and compact charge-packet-differencing circuit that generates an output charge packet equal to the difference of two input charge packets, such that $Q_o = Q_a - Q_b$ for $Q_a > Q_B$ and $Q_o = 0$ otherwise. The circuit is shown schematically in Fig. 2. The advantage of this scheme is that the output charge packet can be regenerated many times. For example, if $Q_b = 0$, then the circuit can be used as a charge-packet-copying circuit, producing multiple copies of an input charge packet. This property also can be used to serve as short-term memory for frame-to-frame operations.

The absolute value of the difference can be implemented using the differencing circuit just discussed if it is operated twice. First, Q_a and Q_b are used to generate an output packet that is stored in the output summing bucket. Then, Q_a and Q_b are reloaded onto the Q_b and Q_a inputs, respectively, and a second output packet is summed with the first. Since only one of the two output packets can be nonzero, the result is the absolute value of the difference.

Fixed gain or attenuation also can be implemented using the differencing circuit if the areas of the A and B electrodes are not equal. However, as discussed, circuit speed will suffer if the electrode width becomes too large.

For several applications, it would be useful to compare the magnitude of two charge packets. The output of such a magnitude-comparator circuit is used to conditionally generate a charge packet. For example, the magnitude comparator could gate a second differencing circuit whose output would be zero if the circuit were disabled by the magnitude comparator. Colbeth et al.[7] recently developed a comparator that has a large dynamic range and the sensitivity and speed required for charge-coupled computing. This circuit, shown schematically in Fig. 3, is basically a flip-flop whose nodes A and B are precharged according to the size of charge packets Q_a and Q_b, respectively. When the flip-flop is enabled, its positive feedback swings it into one of two stable states. The final state depends on the precharged node voltages, and the stable node voltages can be used to drive other circuits. In the case of the differencing circuit, the node voltage can be used to selectively gate the fill cycle, as illustrated in Fig. 4.

Two arithmetic functions whose implementation in the charge domain are not yet fully developed are a charge-domain multiplier and a logarithmic compressor. In the former, an output charge packet is generated such that $Q_o = Q_aQ_b/Q_{ref}$, where Q_{ref} is a reference size and may be externally controlled. A scheme for implementing this in the charge domain by adapting the circuit of Yamasaki and Ando[8] is being investigated. A logarithmic compression circuit has limited use in a computing circuit (although charge-packet multiplication would then become easy) but can be implemented in the photon transducer fairly easily, by using either photocapacitive transduction or an open circuit solar cell type detector.

Input and output functions also must be implemented as circuits. In particular, to execute 3×3 kernel operations, a given PE must be able to obtain data from the local neighborhood on the image plane. A simple charge-transfer-device structure could be used here, but since the same

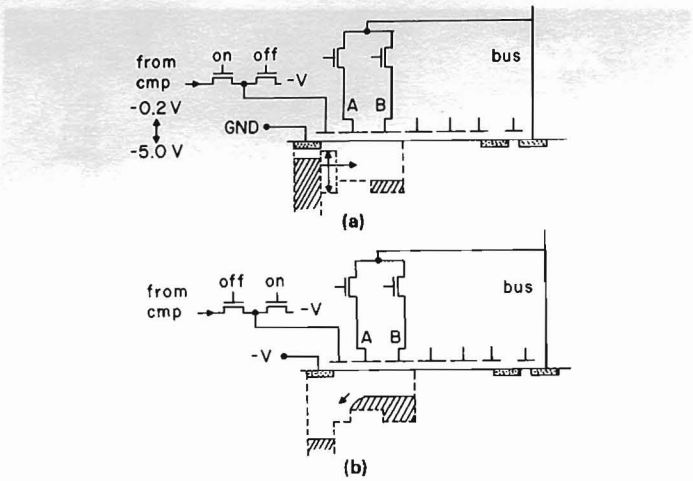


Fig. 4. Gated charge-packet differencer. The input transfer gate is used to conditionally block the fill cycle of the differencer.

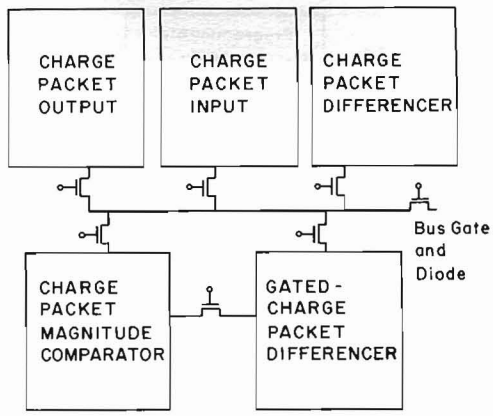


Fig. 5. Block diagram of architecture of the prototype charge-coupled computer.

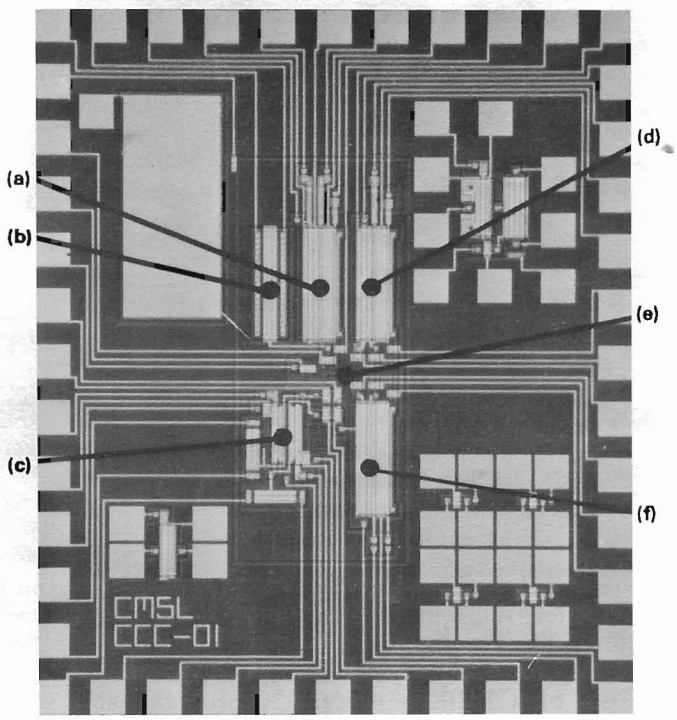


Fig. 6. Photograph of fabricated prototype charge-coupled computer. Wiring is 10 μm in width. Chip size is 2.2 x 2.6 mm. (a) Input. (b) Output. (c) Magnitude comparator. (d) Regular differencer. (e) Bus. (f) Gated differencer.

charge packet frequently must be shared with many neighbors, a replicator-based transceiver structure is suggested. Data to be transmitted are loaded into the replicator and are transmitted by replicating the charge packet and transferring it using the 3-D charge-coupling technique described.

3. PROTOTYPE CHARGE-COUPLED COMPUTER

To experimentally investigate the charge-coupled-computing schemes described, a simple charge-coupled computer was designed, fabricated, and tested. The fabrication process used at Columbia to build the devices is primitive by modern standards but nevertheless allows exploration of the relevant concepts. P-channel diffused junctions, diffused channel stops, and single-level aluminum electrodes separated by open submicrometer gaps were employed, leading to a 10 μm design rule. An internal gettering cycle was used to reduce dark current to the nA/cm^2 level. The process results in relatively poor charge-transfer efficiency due to 20 mV barriers in the open gaps. Source and drain to gate capacitance is another problem with this process, especially in view of the 3-D coupled circuits used.

The architecture of the simple charge-coupled computer is shown in Fig. 5. It consists of input and output structures, a charge-packet differencer, a charge-packet-magnitude comparator, and a second charge-packet differencer gated by the output of the magnitude comparator. The circuits are connected together by means of a central diffused-junction bus through which charge to be exchanged between circuits is passed. MOSFET switches control the connection of various circuits to the bus at the appropriate time. A MOSFET switch connects the bus to an input diode to allow the precharging of electrodes to the voltage applied to the input diode. For example, the output amplifier reset voltage is applied in this manner.

The computer is general purpose and is programmed by applying a particular sequence of clocking signals. Figure 6 is a photograph of the fabricated computer.

Testing of the charge-coupled computer is under way, and the functionality of the charge-coupled-computing circuits has been verified. A block diagram of the test station is shown in Fig. 7. Digital timing is provided by a Pulse Instruments model PI-5800A pattern generator, which is remotely controlled by an IBM PC/AT. The timing signals are level shifted by a bank of Pulse Instruments drivers, models PI-451A and PI-454, and applied to the chip under test, which is inside a shielded box. All connections are coaxial to the printed circuit card on which the chip socket is located so as to minimize noise pickup and clock coupling. DC voltages are supplied through 10-turn potentiometers from a common power supply. Three on-chip source-follower output amplifiers with external resistors are used to monitor the performance of the circuits. One amplifier is connected to the central bus, while the other two monitor

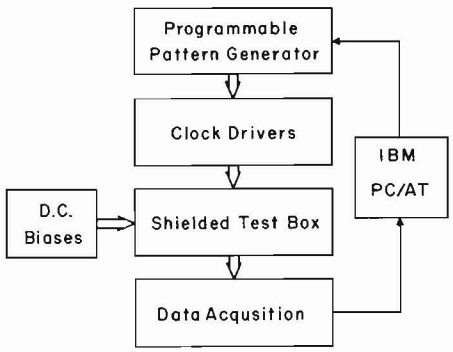


Fig. 7. Block diagram of test station.

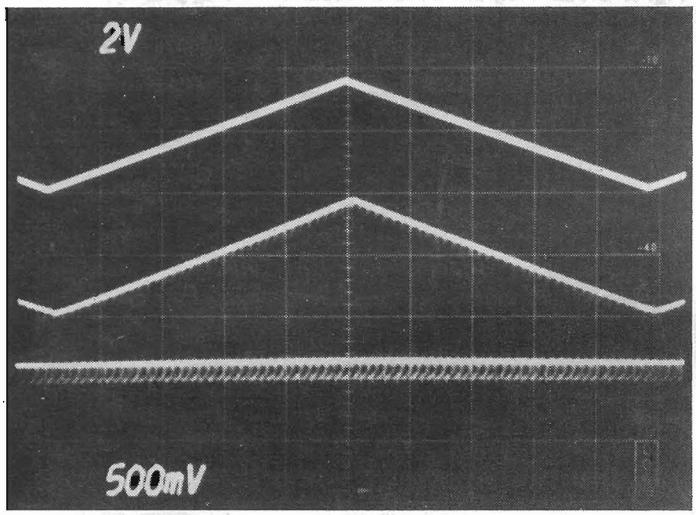


Fig. 8. Input/output transfer characteristic of charge-coupled computer. Top line: input. Middle line: output. Clock width = 500 ns. Scale is 1 ms/division.

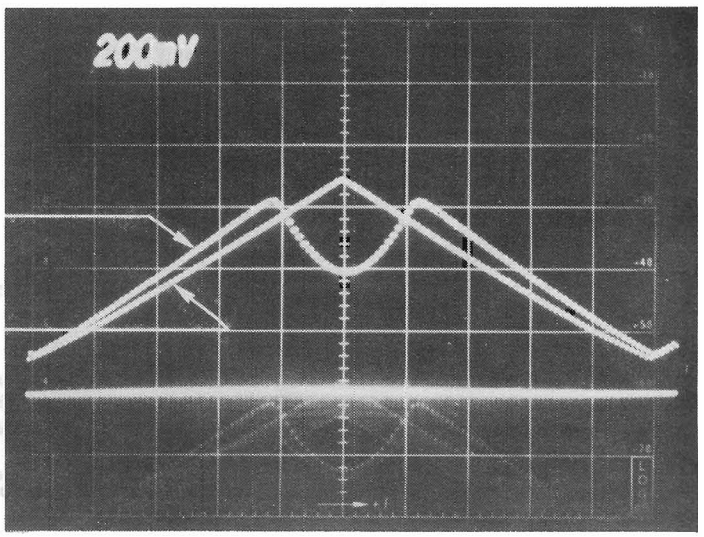


Fig. 9. Replicator-mode characteristic of charge-packet differencer. Top arrow: replicated packet. Bottom arrow: original packet. Clock width = 400 ns. Scale is 1 ms/division.

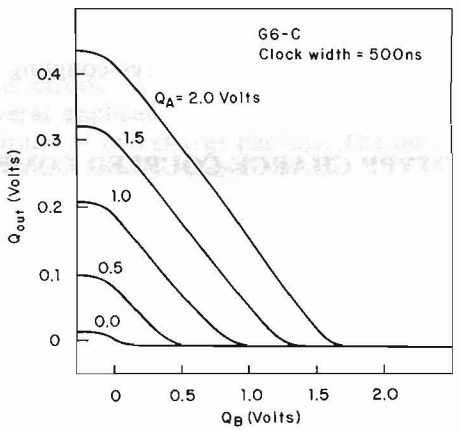


Fig. 10. Transfer characteristics of charge-packet differencer. Note zero reference points. Charge measured by input and output voltages.

the transient performance of the charge-packet-magnitude comparator, following the A and B node voltages. Data are captured using a dual differential sample-and-hold circuit and are either manually or automatically recorded through the use of a Keithley model 230 programmable voltage source and a Keithley model 617 electrometer. Despite reasonable precautions, the test station noise is dominated by residual 60 Hz pickup, corresponding to 10 mV peak-to-peak as seen on the output amplifier.

The simplest operation to verify on the charge-coupled computer is input and output. Figure 8 shows the input/output transfer characteristic with the input operated in a surface-equilibration mode. Charge is transferred to the output amplifier using "drop-and-push" clocking. From the geometry, a transfer time of 200 ns is expected, and the system clock rate was set accordingly. A rise rate of 0.5 V/ns was used on all push clocks. The output amplifier is reset through use of the bus gate and diode. The input circuit delivers 43 pC/V at the output as measured by monitoring the dc current flow through the input diode, and the output circuit delivers 0.65 V at the output per volt at its input, as measured by dc use of the bus gate and diode. Thus, the effective output amplifier capacitance is 28 pF, which agrees with calculations based on the layout. The parasitic bus capacitance is dominated by source and drain to gate capacitance for all bus switches and was estimated to be 1.1 pF. Although the charge-transfer loss due to this parasitic capacitance is expected to be large in this prototype computer, a modern self-aligned process would eliminate nearly all of the parasitic capacitance.

The charge-packet differencer was tested both in replicator mode ($Q_b = 0$) and as a differencer. These data are shown in the oscilloscope photograph in Fig. 9 and in Fig. 10. The differencer linearity and gain are strongly affected by the large bus parasitic capacitance. Nevertheless, it works surprisingly well. The differencer output was generated and summed two times in the differencer summing bucket prior to transfer to the output amplifier. However, to get optimal performance from the differencer, the operating voltages were 25 V on the input switches and 12 V on the transfer line.

The charge-packet-magnitude comparator was tested by again forming two charge packets Q_a and Q_b in the input circuit and sequentially transferring them to the comparator nodes. The clocking voltages for the comparator are shown

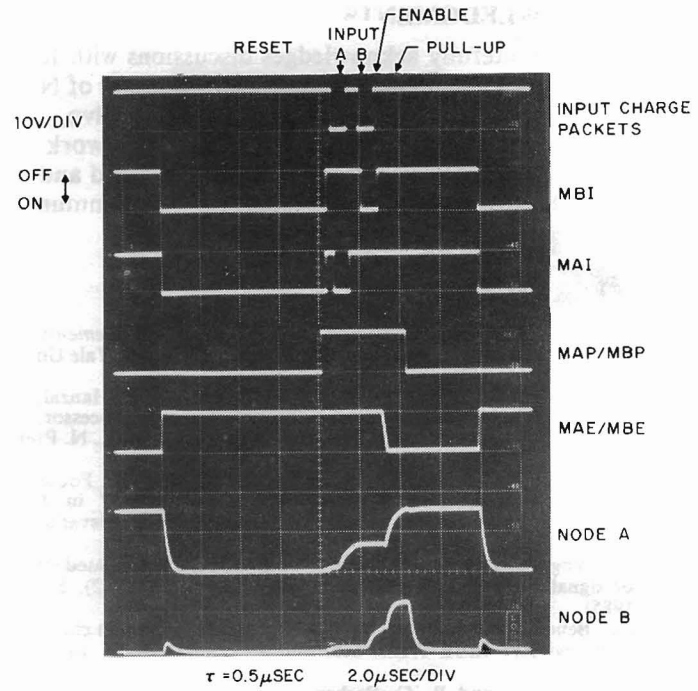


Fig. 11. Clocking waveforms and output of node voltage source-follower output amplifiers for charge-packet-magnitude comparator.

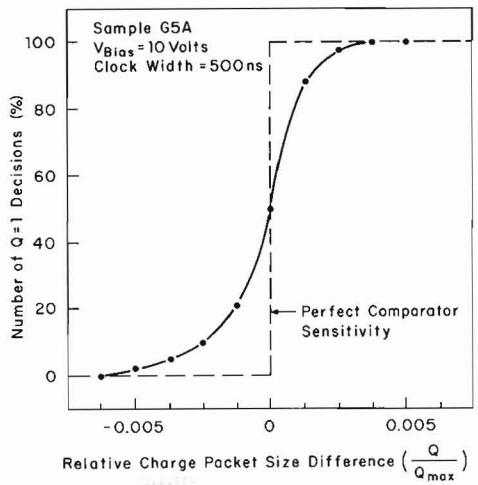


Fig. 12. Sensitivity of charge-packet-magnitude comparator as a function of relative charge-packet-size difference.

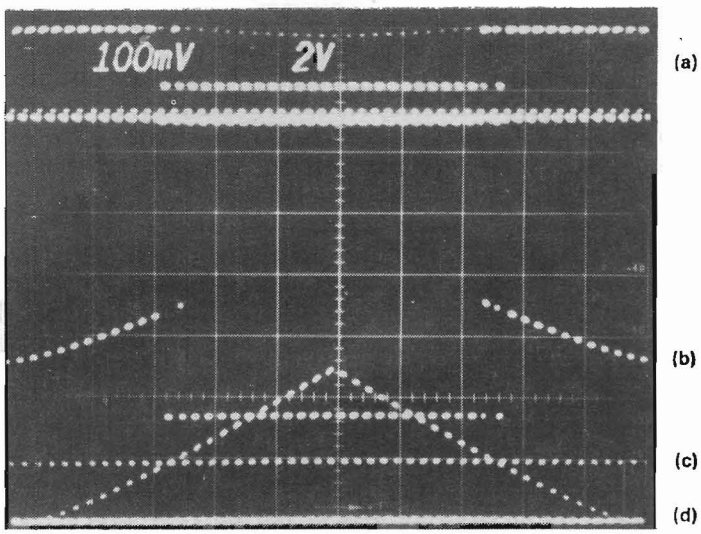


Fig. 13. Output of gated charge-packet differencer as controlled by charge-packet-magnitude comparator. (a) Comparator output. (b) Output. (c) Input A. (d) Input B.

in Fig. 11. The two critical parameters for characterizing the comparator (other than speed, which is sufficient for this computer) are offset and sensitivity. The offset for the comparator depends on alignment accuracy in this layout and was 150 fC for the prototype computer tested. The sensitivity arises due to noise. In this case, the node voltages develop a randomly signed difference for the same size of input charge packet. The sensitivity can be obtained by measuring the frequency of logic ones generated by the comparator as a function of the relative charge-packet size difference. This is shown in Fig. 12. The sensitivity corresponds to a dynamic range of better than 1 part in 100 and is limited by residual 60 Hz pickup.

The gated charge-packet differencer was tested by generating charge packets Q_a and Q_b in the input structure and sending them to the comparator. The charge packet Q_b was replicated by the differencer only if Q_a was larger than Q_b. The output is shown in Fig. 13.

With the functionality of the prototype computer established, measurements are now under way to test the computer's performance when executing more complex programs. These results will be reported when they become available.

4. PROSPECTS FOR SPATIALLY PARALLEL IMAGE PLANE PREPROCESSING

The feasibility of an array of charge-coupled computers is now addressed. There are several issues, as was discussed in the introduction. First is the matter of packing density. The prototype device has nominal dimensions of 600 μm × 1500 μm using a 10 μm design rule. Using a 1.5 μm design rule, allowing for additional real estate for a phototransducer and nearest-neighbor I/O, and with improved design, it is reasonable to expect a packing density of approximately 40 to 50 PE/cm, or between 1600 and 2500 PE/cm^2. Certainly, an array size of 32 × 32 should be obtainable and would give adequate resolution in a number of applications.

A second consideration is processing throughput. With a 1.5 μm design rule, 20 ns clock widths would be more than adequate for good charge-transfer efficiency. Each arithmetic function may take approximately 25 clock cycles, or 500 ns. If 500 operations are required per pixel, the total computation time is 0.25 ms. Thus, for complex image preprocessing tasks, a 4000 Hz frame rate should readily be obtainable. Since this is faster than most applications would require, one might suggest that a more optimal design is to create one charge-coupled computer for every four pixels, thus allowing a resolution on the order of 6000 to 10,000 pixels/cm^2. Such a compromise may make image plane processing attractive for staring IR focal plane array applications, depending on image contrast and dynamic range requirements.

Power density is another issue to be considered. This will depend on the average amount of charge transferred during each clock cycle. A bucket with 10^7 carriers, which is, on average, half full, will dissipate approximately 8 pJ per transfer if the bucket potential changes by 10 V. If each arithmetic function requires 25 transfers and if 500 operations are performed, then 100 nJ are dissipated in the semiconductor to process one pixel. For 5000 pixels per frame, 1 mJ is consumed per frame. Thus, at 100 Hz, 50 mW of power is dissipated, and power consumption will not be a problem in most applications. It should not be overlooked that 5000 pixels are being preprocessed at 100 Hz, corresponding to 250×10^6 operations per second, at a cost of 50 mW.

Noise in charge-coupled-computing circuits should not be a problem since the number of noise-generated carriers is expected to be smaller than the number of error carriers introduced by circuit inaccuracy. (The impact of predictable circuit inaccuracy is an interesting but unexplored issue.*) The noise floor can be expected to be dominated by kTC noise for 3-D coupled circuits with otherwise good charge-transfer efficiency. If the minimum resolvable charge packet Q_{min} is defined as $AC_{ox}V/r$, where A is the electrode area, C_{ox} is the specific oxide capacitance, V is the bucket depth in volts, and r is the resolution of the circuit, then the minimum resolvable charge packet is equal to the root-mean-square number of noise carriers, $(kTC/q^2)^{1/2}$, when the electrode area is reduced to a critical area A_{cr}, given by $A_{cr} = kTr^2/C_{ox}V^2$. This critical area is significantly smaller than those envisioned in scaled charge-coupled-computing circuits. For example, with $r = 256$, a 100 Å oxide, and a 1 V bucket, the critical electrode size is approximately 0.25 μm $\times$ 0.25 μm and Q_{min}/q is 7 carriers. During operation, kTC noise will be additive, but the number of operations required to build up noise equal to Q_{min} is equal to A/A_{cr} and is larger than the number of operations per pixel per frame anticipated for charge-coupled-computing applications. For example, for an electrode area of 100 μm^2, $r = 256$, a 250 Å oxide, and a 5 V bucket, Q_{min}/q is 16,800 carriers and the critical number of operations is 12,700.

Encouraged by this analysis, we are currently designing and fabricating a modest array of charge-coupled computers for image plane preprocessing experiments.

5. CONCLUSIONS

A new class of CCDs that perform arithmetic and logic functions in the analog charge domain has been described. A prototype charge-coupled computer employing these circuits has been designed, fabricated, and tested. The results of experimental and theoretical studies indicate that it is feasible to put an array of these simple computers on the image plane to perform image preprocessing functions in a spatially parallel way.

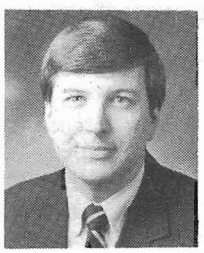
*J. Joseph, private communication (1985).

6. ACKNOWLEDGMENTS

The author gratefully acknowledges discussions with R. C. Barker and R. E. Colbeth. The technical assistance of N. A. Doudoumopoulos, S. E. Kemeny, A. Montalvo, and S. Seshadri is also gratefully acknowledged. This work was supported by an IBM Faculty Development Award and the National Science Foundation Center for Telecommunications Research at Columbia University.

7. REFERENCES

1. E. R. Fossum, *Charge-Coupled Analog Computing Elements and Their Application To Smart Image Sensors,* Ph.D. thesis, Yale University (1984).
2. J. D. Joseph, P. C. T. Roberts, J. A. Hoschette, B. R. Hanzal, and J. C. Schwanebeck, "A CCD-based parallel analog processor," in *State-of-the-Art Imaging Arrays and Their Applications,* K. N. Prettyjohns, ed., Proc. SPIE 501, 238-241 (1984).
3. G. R. Nudd, P. A. Nygaard, G. D. Thurmond, and S. D. Fouse, "A CCD image processor for smart sensor application," in *Image Understanding Systems and Industrial Applications,* R. Nevatia, ed., Proc. SPIE 155, 15-22 (1978).
4. T. L. Vogelsong and J. J. Tieman, "Charge domain integrated circuits for signal processing," IEEE J. Solid-State Circ. SC-20(2), 562-570 (1985).
5. S. S. Bencuya and A. J. Steckl, "Charge packet splitting in charge domain devices," IEEE Trans. Electron Devices ED-31(10), 1494-1501 (1984).
6. E. R. Fossum and R. C. Barker, "A linear and compact charge-coupled charge packet differencer replicator," IEEE Trans. Electron Devices ED-31(12), 1784-1789 (1984).
7. R. E. Colbeth, S. E. Kemeny, N. A. Doudoumopoulos, A. Montalvo, and E. R. Fossum, "Charge packet magnitude comparator for charge-coupled computing applications," submitted to IEEE Trans. Electron Devices.
8. H. Yamasaki and T. Ando, "Optical multiplication in solid-state imaging devices with an inherent MNOS memory gate," IEEE Electron Device Lett. EDL-6(2), 88-90 (1985).



Eric R. Fossum received the BS degree with honors in physics and engineering from Trinity College in Hartford, Conn., in 1979. He received the MS and Ph.D. degrees from Yale University in electrical engineering in 1980 and 1984, respectively. While engaged in graduate study, he spent three summers with the Hughes Aircraft Missile Systems Group in Canoga Park, Calif., working on various problems related to focal plane array detector and readout structures. Since 1984, he has been an assistant professor of electrical engineering at Columbia University. He and his students are currently engaged in research on silicon charge-coupled-computing devices, GaAs charge-transfer devices, GaAs surface passivation, low energy ion beam processing of semiconductors, and connecting optical fibers directly to integrated circuits.

Professor Fossum recently organized the IEEE Workshop on Charge-Coupled Devices. In 1984, he received an IBM Faculty Development Award, and in 1986 he received the National Science Foundation Presidential Young Investigator Award.