Design of a CCD focal-plane codec preprocessor for lossless image compression

S.E. Kemeny, H. Meadows, and E.R. Fossum

Center for Telecommunications Research
Columbia University
New York, New York 10027

ABSTRACT

The design of a CCD-based codec preprocessor integrated with an areal imager is presented. The imager is a 256x256 buried channel frame transfer device with 15x15 um^2 pixels. Conventionally, imagers deliver pixel data in a raster-scan sequential format. Typical of many image processing tasks, the image coding scheme used in our lossless compression algorithm requires 3x3 pixel neighborhood blocks with the center pixel arriving first. Several techniques for accomplishing this type of neighborhood reconstruction (NR), consisting of both global (row) and local (pixel) reordering operations are described and realized on the focal-plane. The CCD NR circuits are designed to operate at a 30 Hz frame rate with minimal power (< 10 uW) and real-estate (< 250x400 um^2). The codec preprocessor consists of the NR circuitry and subsequent pyramid sampler. Differential output provides off-chip electronics with data suitable for coding and compressed transmission.

1. INTRODUCTION

Focal-plane image processing refers to the on-chip integration of image acquisition and image processing hardware. There is a wide range of image processing tasks which can be implemented on the focal-plane, from simple detector amplifier read-out to more complex tasks such as motion detection or image coding.[1] One important application is in the area of image transmission. Due to the high throughput requirements encountered in lossless image transmission systems (> 10^9 bits/s), lossless image compression is important for many applications. Focal-plane integration of preprocessing hardware can alleviate the performance requirements of the downstream processing circuitry, and so facilitate real-time image transmission. To this end, a CCD Codec Preprocessor (CP) integrated with a 256x256 buried channel CCD frame transfer imager with 15 um^2 pixels has been designed.

Conventional CCD imagers are read out in a raster scan sequential format so that vertical neighbors are separated by a full row of pixels. Typical of many image processing tasks, such as convolution and edge detection, the image coding used in our lossless compression algorithm is performed on 3x3 pixel neighborhood blocks. In order to supply downstream electronics with the appropriate data sequences (3x3 kernels), the image data must be reorganized. This reformatting of image data is referred to as neighborhood reconstruction (NR). Several approaches to NR utilizing CCD circuits have been developed. The codec preprocessor consists of the CCD NR circuitry followed by a sampling

output stage which delivers pixel data in the appropriate sequence for off-chip differential pyramid encoding. Four codec preprocessors implementing a variety of NR schemes have been designed. Two are designed for hybridization to the CCD imager within a standard 68 pin package and two are realized on the focal-plane.

2. CCD CODEC PREPROCESSOR ALGORITHM

The codec preprocessor (CP) provides data for use with a pyramid structured progressive transmission technique for lossless image compression. The coding algorithm has been adapted from a lossless compression transform described by Torbey.[2] The algorithm exploits the spatial (or temporal) correlation between closely spaced pixels of an image by encoding the differences between nearest neighbors to form an improved effective image intensity histogram for off-chip conventional (eg. Huffman) coding. On every 3x3 neighborhood, an off-chip encoder forms the difference between the center pixel and each of its surrounding eight neighbors. Therefore the encoder requires the reconstruction of 3x3 neighborhood blocks with the center pixel arriving first as illustrated in Fig. 1. The organization of the CP is shown in Fig. 2. The NR block reorganizes the imager data into 9-pixel blocks and outputs these local neighborhoods into two streams. One output stream (A) represents the center pixels, while the other (B) is composed of the corresponding peripheral pixels. The two streams are fed into a differential input A/D converter. The resultant digital output code represents the desired differences between the center and peripheral pixels and is in the appropriate order for progressive transmission and image restoration.

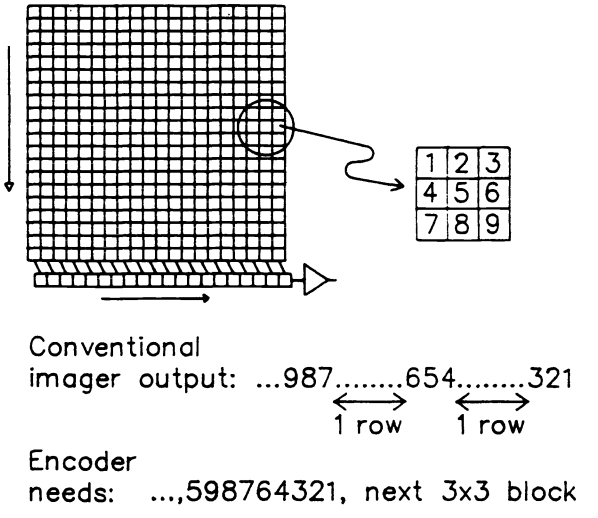


Fig. 1. Pyramid encoder neighborhood reconstruction requirements.

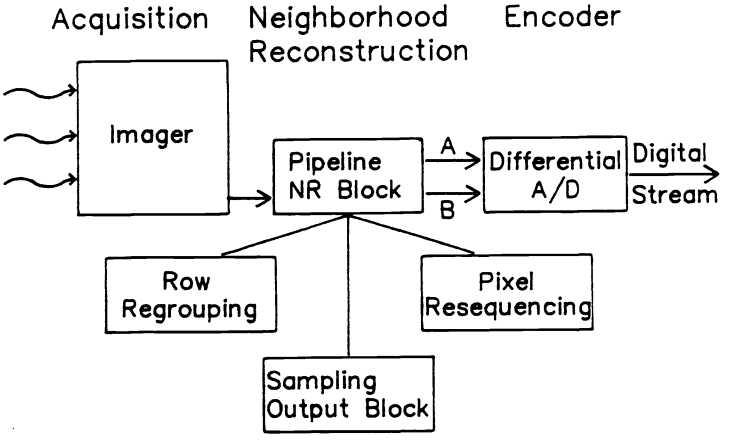


Fig. 2. CCD codec preprocessor block diagram.

3. NEIGHBORHOOD RECONSTRUCTION ARCHITECTURES

Neighborhood reconstruction consists of two main functions: row regrouping and pixel resequencing. Row regrouping refers to the

vertical reformatting of image data to provide simultaneous access to several rows of data (eg. 3 for a 3x3 kernel). Pixel resequencing refers to the reordering of pixels to create a desired pixel sequence within a local neighborhood or row. Several approaches to each of these functions will be discussed.

3.1. Row regrouping

Row regrouping can be accomplished in a variety of ways. On-chip row reconstruction utilizing a CCD delay line to provide a buffer for storage of the desired rows of data has been demonstrated by Hall et al.[3] In their approach three rows of data are sequentially read in to a long serial CCD delay line. The pixel data are then simultaneously sensed non-destructively at the beginning, middle and end of the delay line. Three adjacent pixels are sensed at each location, providing a 3x3 neighborhood block. Although simple to implement, this technique suffers from a large and non-uniform CTE loss introduced by the long shift register. Another technique for row regrouping which utilizes a floating diffusion direct read-out approach has also been reported[4], but suffers from threshold voltage variations and low speed.

3.1.1. Row regrouping: single output method

An improved version of the delay approach to row regrouping is shown in Fig. 3. Three rows of imager data are sequentially written into 3 separate registers, rather than one long one. The registers are then jointly clocked to provide simultaneous read-out of the 3 imager rows.

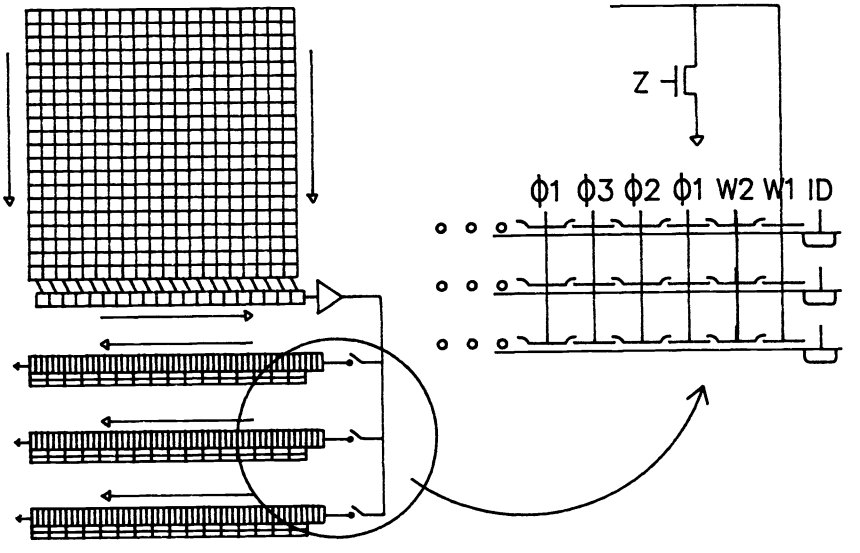


Fig. 3. Row-regrouping: sequential serial pixel output.

Writing of the data is accomplished by means of the potential equilibration or "fill and spill" technique.[5] Each pixel charge packet is regenerated in one of the serial shift registers during every pixel read-out cycle. In this approach, as in conventional areal

imagers, the pixels are sequentially read-out of the CCD multiplexer and converted to the voltage domain via a source-follower amplifier. This signal voltage is used to set the voltage level on metering well 1 (W1). Subsequently, the input diode is momentarily forward biased initiating the fill of the fill and spill sequence. Lastly, the excess charge spills back over the barrier of W1 leaving the signal charge Qs = $ACox(V_{w2} - V_{w1})$ under W2. The regenerated signal charge packet is then transferred through the shift register via standard 3-phase clocking. By keeping two of the three input diodes reverse biased while the third is forward biased, the appropriate shift register is selected. Although the source-follower amplifier inverts the signal and introduces a less than unity gain, the fill and spill stage produces compensating positive gain and inversion. By increasing the area of the fill and spill electrodes (W1 and W2) relative to the pixel electrodes, the signal charge (Qs) is increased. Inversion is accomplished by applying the signal voltage to W1 rather than W2. After the first row is written, the charge packets are shifted into a parallel register for storage during the writing of the second and third rows. After all three rows are written, the packets are transferred back into the serial register for simultaneous read-out. This scheme can be expanded to provide access to any number of rows. The only requirement is that the source-follower amplifier stage be scaled to drive an increased load.

Although this NR approach suffers from delay, non-uniformity introduced by the reordering process is minimized. Each pixel undergoes the same number of transfers and passes through the same output amplifier. Since this type of NR receives pixel data from conventional raster scan imager output, it can be implemented on or off the focal-plane. This architecture was adopted for the two CP chips which are implemented off the focal-plane.

3.1.2. Row regrouping: triple output method

A second approach to row regrouping which is not based on delay and does not require any charge/voltage conversion is illustrated in Fig. 4. In this architecture, a non-conventional lay-out employing a triple-poly process is used to simultaneously read-out three rows of the buried-channel frame-transfer imager array. A parallel-serial-parallel structure (sp^3) is used to load three rows of imager data into the three phase serial registers. Two architectures for implementing this structure have been designed.

In the first, an adaptation of conventional parallel to serial transfer architecture is implemented. The serial registers are used in two modes: serial or parallel. In the former, they are clocked conventionally to serially transfer image packets out of the array. In the latter, two of the three phase electrodes are operated as one parallel gate while the third phase electrode acts as a channel stop. Since this parallel transfer is across the length, rather than width of a CCD electrode, the transfer time will be diffusion limited for most pixel sizes (> 12 um). As long as this time is less than the read-out rate of the serial multiplexer, it will not impose any speed constraints to the NR block. For the current design, assuming

diffusion limited transfer, the parallel transfer time constant is expected to be less than 50 ns. The read-out rate is 1 us/pixel, allowing more than 20 time constants for the diffusion limited exponential decay. This architecture provides the row regrouping for the two CPs which are implemented on the focal-plane.

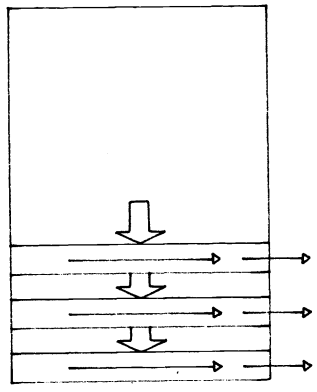


Fig. 4. Row regrouping: parallel read-out of three rows.

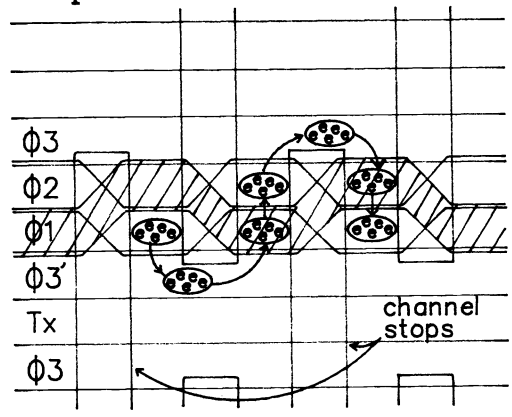
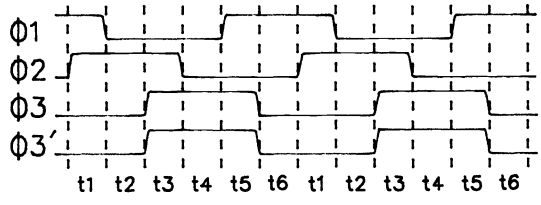
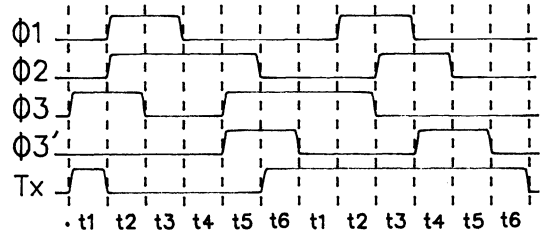


Fig. 5. Schematic lay-out of twist architecture illustrating serial transfer of charge packets.



(a)



(b)

Fig. 6. Twist architecture timing diagrams. a) Serial mode operation. b) Parallel mode operation.

The second parallel-serial-parallel architecture uses a novel triple-poly lay-out, without metal bus lines. In this technique a twist architecture shown schematically in Fig. 5 is used to serially transfer charge. The charge packets follow a sinusoidal pattern through the register by conventional 3-phase clocking (Fig. 6a). By clocking the electrodes in an alternate pattern (Fig. 6b), the register can operate as a set of parallel transfer gates. As shown in the timing diagrams, two clock cycles are required for transfer of one charge packet. This drawback can be overcome by operating the serial registers at twice the nominal frequency. There is no CTE penalty for the higher speed operation since the charge packets do not encounter long transfer lengths (> 7 um), but there is a subsequent increase in power dissipation. The major advantage of this approach is the elimination of metal clock lines. This leads to decreased real-estate consumption and more importantly, decouples pixel size from the design rule constraints of metal to poly contacts. Minimum contact size generally determines minimum device geometry. Therefore, much smaller

pixels (implying higher resolution) can be accommodated with this technique than would be expected within a given set of design rules.

Both parallel-serial-parallel read-out structures can be expanded to simultaneously read-out any number of rows. No scaling adjustments are necessary. The only requirement is that the time needed for processing one pixel be less than the time to transfer the image packets across the serial register. An additional chip, integrating both sp^3 row regrouping techniques has been designed to quantitatively compare the two architectures.

3.2. Pixel resequencing

Once the desired rows have been made available, it is often necessary to change the order of the pixels within one or more rows. This local reordering of pixels, termed pixel resequencing, is the second major task of neighborhood reconstruction. For instance, the lossless codec preprocessor computes the difference between the center pixel and each of its eight neighbors, requiring that the center pixel be read out first. Two approaches to this particular pixel resequencing will be described, although the same strategies can be applied to a wide variety of reformatting tasks. The two techniques are implemented in both the on and off-chip CPs.

3.2.1. Pixel resequencing: delay technique

In the first approach, illustrated in Fig. 7, the pixels are reordered by use of delay. Six, three and nine pixels are appended to the end of each of the three serial output registers respectively. As the image packets are simultaneously clocked through the appended registers, the pyramid sampling output block receives the center row first, followed by the top and then bottom row. The output block consists of the three source-follower output amplifiers (A,B,C), the sampling capacitors and sampling switches. The output stream, upon conversion to the voltage domain by amp A, is routed to one of two storage capacitors, each of which is associated with a source-follower output amplifier (B and C). The center pixel, which is the second to arrive at amp A is routed to amp C, while the eight peripheral pixels are routed to amp B. By sampling amp C (center pixel), before amp B, the order of the first and second pixel are effectively interchanged. This delayed sampling does however introduce an extra one pixel delay to the 9-pixel neighborhood read-out.

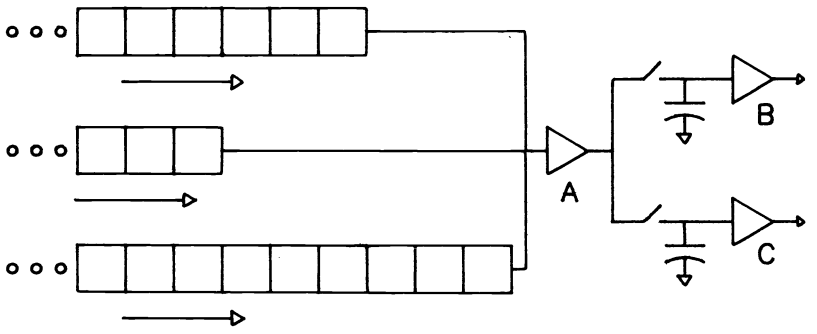


Fig. 7. Pixel resequencing by use of 3, 6 and 9 pixel delay.

3.2.2. Pixel resequencing: wire transfer technique

The second approach to pixel resequencing uses the technique of wire transfer, recently demonstrated by Fossum.[6] Combining elements of both bucket brigade and CCD devices, charge packets are transferred across metal wires facilitating the crossing of signal paths. Conventional serial to parallel and parallel to serial registers provide a parallel structure for the wire transfer re-routing. Charge packets in a serial multiplexer are loaded into a parallel register. They are then routed via wire transfer to the desired parallel inputs. The reordered packets are then transferred into a serial shift register which is read out conventionally. This technique can be readily applied to the CCD CP pixel sequencing problem and is shown schematically in Fig. 8. In this case, a three pixel serial to parallel register is appended to each of the three serial output row registers. The nine parallel pixels are wired to a nine port parallel-in serial-out shift register. The center pixel is wired to the first of the nine inputs, providing the desired resequencing. The data is then serially transferred into the output block described previously. In this case, the first packet in the nine pixel block (which corresponds to the center pixel) is sampled upon arrival, so that no delay is incurred upon read-out of the local neighborhood.

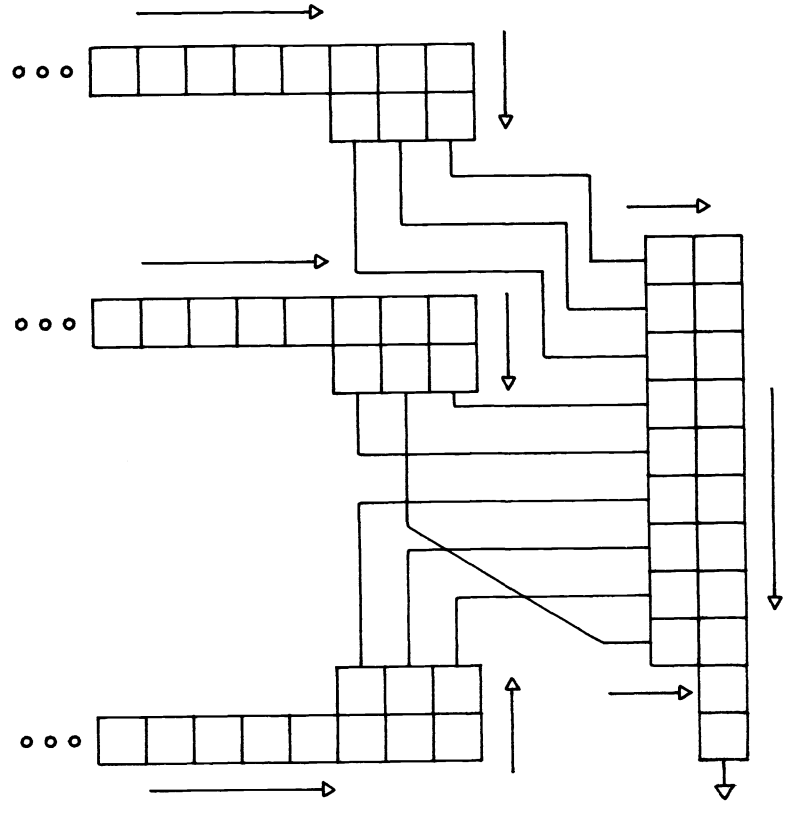


Fig. 8. Wire transfer pixel resequencing for CCD codec preprocessor (nine pixels loaded in parallel).

4. CONCLUSION

Four CCD codec processor chips have been designed. They implement different approaches to neighborhood reconstruction involving both row regrouping and pixel resequencing. Two are designed for hybridization to a separate 256X256 buried channel imager array, while two are integrated with the imager. They are designed to operate at a 30 Hz frame rate with power dissipation less than 10 uW. The hybrid chips measure 2.5x5.5 mm^2, while real-estate consumed by the CPs on the focal-plane chips is approximately 250x400 um^2. After reorganizing the image data into local 3x3 neighborhood blocks the CPs provide differential output suitable for lossless coding and compression to off-chip electronics.

5. ACKNOWLEDGMENTS

The authors would like to thank E-S. Eid of Columbia University for fruitful discussion during the design process. Useful discussion concerning the fabrication of the SP^3 read-out structure with R. Bredthauer and M. Costa of Ford Aerospace is also gratefully acknowledged. This work has been supported, in part, by Ford Aerospace and the NSF Center for Telecommunications Research at Columbia University.

5. REFERENCES

1. E.R. Fossum, "Architectures for focal-plane image processing," Opt. Eng. vol. 28 (8) pp. 865-871 (1989).
2. H.H. Torbey and H. Meadows, "System for lossless hierarchal image coding/decoding," CTR Tech. Rep. No. 94, Columbia University (1988).
3. J. E. Hall, J.F. Breitzmann, M.M. Blouke, and J.T. Carlo, "A multiple output CCD imager for image processing applications," in Proc. 1978 IEEE Int. Electron Devices Meeting, pp. 415-418 (1978).
4. J. P. Vermeiren and G. J. Declerck, "A novel CCD output structure for picture processing applications," in Proc. 1985 IEEE Custom Integrated Circuits Conf., pp. 128-131 (1985).
5. M.F. Tompsett, "Surface potential equilibration method of setting charge in CCD's," IEEE Trans. Electron Devices, vol. ED-22, pp. 305-309 (1975).
6. E.R. Fossum, "Wire transfer of charge packets for on-chip CCD signal processing," see this proceedings, 1242-27 (1990).