The Development and Application of a Digital Charge Coupled Logic (DCCL) Arithmetic Unit

E. Hyman
TRW
Defense and Space
Systems Group

R. A. Allen
TRW
Defense and Space
Systems Group

ABSTRACT

An arithmetic unit chip is described which combines the inherent low power dissipation of CCD's with the accuracy of 16-bit digital arithmetic. A technique for electrically changing a 32-bit adder to a 32-bit subtractor or an exclusive-OR function is also described.

The arithmetic unit operates as a pipeline system requiring 40 clock cycles to process data; a programming technique for overcoming this limitation is discussed. The paper also includes a description of how the arithmetic unit can be organized to implement an analysis filter for linear prediction encoding of voice.

DCCL ARITHMETIC UNIT

Previous papers [1]-[3] have described in detail how logic and arithmetic functions such as AND, OR, exclusive-OR, complement, half-adders and full-adders are designed in Digital Charge Coupled Logic (DCCL). In this paper we describe how these functions are interconnected with continuous CCD transfer channels to form a DCCL/LSI pipeline arithmetic unit.

In pipeline DCCL arithmetic, where it is required to begin processing a new word each successive clock cycle, the 2's complement number system is ideal for addition or subtraction. However, in order to perform multiplication, it is best to convert the 2's complement notation to signed binary numbers. To illustrate, a 16 x 16 multiplier-adder will be briefly described.

The two 16-bit input words to the array enter the multiplier in parallel and are formed into their partial products by 256 AND gates (as shown in Figure 1). The partial products are added by an array of full-adders and since the carry from the least significant product column propagates through 31 other columns gaining a clock phase delay in each column, the product is skewed a total of 32 clock phases from the least significant to the most significant product.

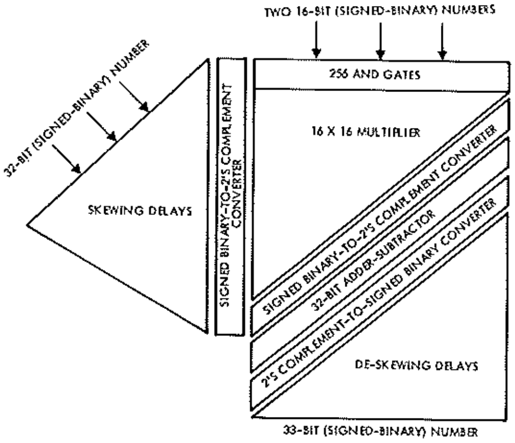
TWO 16-BIT (SIGNED-BINARY) NUMBERS
32-BIT (SIGNED-BINARY) NUMBER
256 AND GATES
16 X 16 MULTIPLIER
SKEWING DELAYS
SIGNED BINARY-TO-2'S COMPLEMENT CONVERTER
SIGNED BINARY-TO-2'S COMPLEMENT CONVERTER
32-BIT ADDER-SUBTRACTOR
2'S COMPLEMENT-TO-SIGNED BINARY CONVERTER
DE-SKEWING DELAYS
33-BIT (SIGNED-BINARY) NUMBER

Figure 1. Block Diagram of the DCCL Arithmetic Unit

The 32-bit product from the multiplier and the 32-bit data word are both converted to 2's complement notation before transferring to the adder/subtractor. In converting negative numbers from signed binary to 2's complement, it is necessary to complement and increment by adding a binary-one to the least significant bit. In a DCCL pipeline system, the incrementing is carried out by parallel interconnected half-adders, with each word-bit being incremented on successive clock cycles as shown on the logic diagram of a 6 x 4 multiplier of Figure 2. The blocks marked CD in Figure 2 are charge duplicating cells that generate two separate charge packages that are of the same binary value and size as their input packet. The skewed product from a multiplier matches the skewed input to the number converter, however, the 32-bit input data arrives synchronously, and results in having to skew the input data word with CCD delay lines.

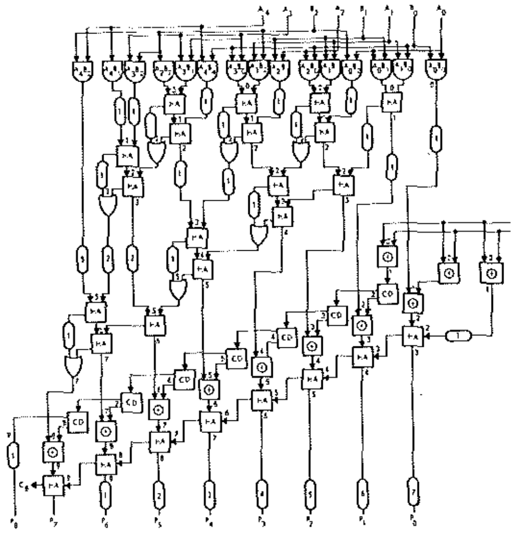

Figure 2. The logic diagram of the 6 X 6 multiplier and signed binary to 2's complement converter. This diagram is directly expandable to a 16 x 16 multiplier array.

The two 32-bit 2's complement numbers enter the array skewed but with the two least significant bits and each subsequent pair of bits synchronously. One of the words; (the subtrahend in the case of the subtract mode) is transferred into one of the input ports of each of the 32 exclusive-OR gates. A control line, K, determines the binary value of the input to the other input port of the exclusive-OR gates as shown in Figure 3.

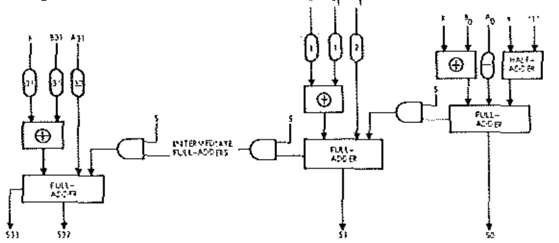

Figure 3. Logic Diagram of DCCL 32-Bit Adder/ Subtractor/Exclusive-OR Array

When control line K is switched so that a binary zero is entered into the exclusive-OR gates, the data is transferred through the exclusive-OR without being changed. However, when the control line K is switched so that a binary one is entered into the exclusive-OR gates, the data is complemented. The control line K, is also transferred into one input terminal of a half-adder and a binary one input packet is transferred into the other half-adder input during each clock period. The half-adder will generate a binary-one charge packet from its carry-out port each clock period when K = 1 and a binary-zero charge packet from the carry-out port when K = 0.

The carry-out port of the half-adder is connected to the carry-in port of the least-significant-bit full-adder in the array. Thus when K = 1, the input data (the subtrahend) is complemented and the difference is incremented by one.

The output from the 32 exclusive-OR gates and the other 32-bit input data are transferred into two of the three inputs of each of the 32 full-adders as shown in Figure 3. The carry-out port of each full-adder is connected to the carry-in port of the next more significant full-adder. Each pair of bits are added (or subtracted) in turn, at each subsequent clock phase. Therefore the most significant pair are acted upon, 32 clock phases after the least significant pair.

In order to be able to switch the function of the array from add/subtract to exclusive-OR, two input AND gates are inserted between the carry-out and carry-in ports of each full-adder. A control line S controls the value of the value of the binary input to the other input port of the AND gates. When S is at the binary zero value, the carry channel is inhibited, and the array output is the exclusive-OR. When S is at the binary-one value, the carry-bits are enabled and the array output is the add/subtract. There are no clock phase delays through a DCCL AND gate, so this extra functional capability does not require more skewing.

The skewed 2's complement output of the adder array is then changed back to the signed binary to 2's complement converter described above.

The output from the number converter is still in a skewed format and required an array of shift registers in order to synchronize the output data.

APPLICATIONS

In order to match the performance of the CCD chip presented in this paper, at an operating speed of 5MHz, a general purpose computer requires a 200ns instruction cycle time. Within this 200ns, a 16 x 16 multiply, followed by a 32-bit add must be performed. To prevent the necessity of a faster operating speed, the computer must be fully pipelined, instruction access and data access must occur simultaneously with arithmetic processing. The existing ALU technology for implementation of a multiplier/adder at this data rate requires a power dissipation in excess of 20W, whereas the equivalent throughput is obtained by the CCD processor presented in this paper at a power dissipation of approximately 1W. Consequently, spacecraft and other low power, high density applications lend themselves to the use of this chip.

Although the shift register nature of CCD's does not allow efficient programming of arbitrary algorithms, as would occur in a general purpose computer, certain processes with wide applications are readily implementable in CCD. Furthermore, programming and scheduling techniques allow efficient implementation of general purpose algorithms. Some of these algorithms and techniques are discussed in the following paragraphs.

The pipeline nature of the CCD ALU chip allows efficient computational through-put for those tasks allowing many multiplies or adds in parallel, where one arithmetic operation is not a function of succeeding operations. Examples of such functions are:

Correlation
Discrete Fourier Transform
Delay Line

Incorporating into the CCD chip the additional facility of latching the 32-bit output from the adder and adding this latched value, rather than the other 32-bit input value, to the output of the multiplier facilitates correlation. Here,

$$\emptyset_j = \sum_{i = j+1}^{k} X_i X_{i-j} \qquad (1)$$

is calculated by loading X_{j+1} and X_1 into the multiply portion of the chip, followed by X_{j+2} and X_2, etc. The complete correlation is accomplished in (k - j) clocks in addition to the chip delay.

A network of 61 correlators depicted in Figure 4 is required for pitch extraction in the process of linear prediction for voice data bandwidth reduction. This correlation network consists of 61 independent calculations of a 16 x 16 multiply preceeding 61 calculations of a 32-bit add to partial accumulation. Even without the additional facility mentioned above, the three operands (two 16-bit multiplicands and the 32-bit partial accumulation for each of the 61 correlations) are loaded into the chip at each clock and the updates occur at the maximum chip through-put rate.

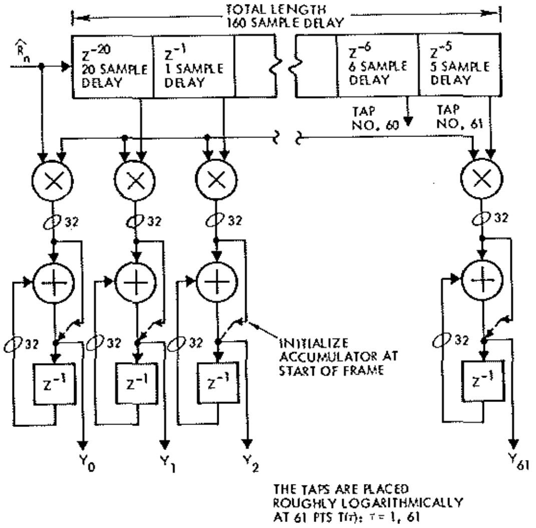

Figure 4. ACF Network

A study on the CCD implementation of the auto-correlation function resulted in the conception of a stand alone processor, depicted in Figure 5. The dashed line of Figure 5 contains that portion of the processor implementable by the CCD ALU considered in this paper. The circulating shift register/MUX complex, which is CCD realizable, allows computation of correlations of arbitrary lags in a straight forward manner via equation (1). This stand alone processor eliminates the need for address generation, as would occur when computing many partial correlation updates as each data sample arrives. It also reduced the processing memory bandwidth, since the shift registers maintain all the needed sampled data. Furthermore, this processor can be implemented in parallel with a general purpose computer. This method is consistent with the correlation techniques on the existing TRW voice processors.

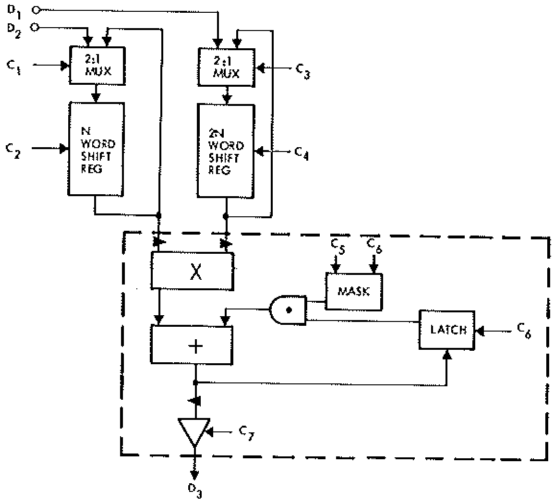

Figure 5. Stand Alone ACF Processor

A critical factor in the use of the CCD chip is the number of shifts required to generate an arithmetic result, once the operands have been clocked into the chip. For an iterative algorithm, one in which each computation requires the successful completion of a previous computation, the through-put is a fraction of the case for the more pipelined algorithms mentioned above. For the ALU chip considered, 40 clocks are required to generate the result, once the operands are made available. Hence, for an iteration on a previous result, such as Newton-Raphson method for square root or divide, the efficiency of the chip is 1/40th the efficiency for, say, correlation.

The pipeline restruction can be overcome however, if the computational requirements is for a number of processes to occur in parallel. A practical example is the implementation of the analysis filter for linear prediction encoding of voice. The computational requirements for this application are depicted in Figure 6. Ten complete analyzer filters must be implemented between voice samples, where the residue from one filter constitutes the input to the next filter in line. Since only 15 to 16 complete iterations through the CCD ALU are feasible between voice samples, the calculation of the complete 10-stage analyzer filter is impossible in a straight forward manner.

However, the computation can be accomplished in the following manner:

> Analyzer Stage 1 is performed on voice sample S_i
>
> Analyzer Stage 2 is performed on voice sample S_{i-1}
>
> Analyzer Stage 10 is performed on voice sample S_{i-9}

with the complete 10 stage filter being accomplished on voice sample S_{i-9}.

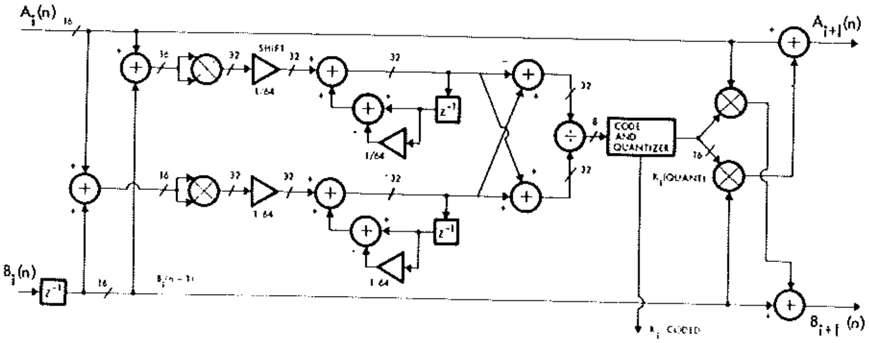

Figure 6. Applicable Voice Algorithms

ADDRESSING CONSIDERATIONS

Incorporation of this CCD ALU chip into a general purpose CPU requires some addressing considerations. The output of the CCD ALU may be required for re-insertion into the chip or the output may be placed into data memory. Since operands are loaded 40 instructions prior to the occurence of the arithmetic result, the intended store address may not be known at the time that the result is available. Although the intended use of the data may be known at the time the operands are placed into the chip, it is only in the case of highly deterministic programs that instructions are known 40 stages in advance. Here, the addressing problem is alleviated with the incorporation of an address shift register of length 40. The storage address is calculated simultaneous with the retrieval of operands and this store address is placed into this shift register. The store address is available from the shift register at the same time that the arithmetic result became available from the CCD ALU.

This work was supported in part by the Naval Electronics System Command and Directed by the Naval Research Laboratory under Contract N00014-74-C-0068.

REFERENCES

[1] T. A. Zimmerman and D. F. Barbe, "A New Role for Charge-Coupled Devices: Digital Signal Processing," Electronics, pp 97-103, March 1977.

[2] C. S. Miller, C. F. Motley, and R. A. Allen, "Digital Charge Coupled Device Technology and Digital Filters Application," EASTCON-77 Proceedings, September 1977

[3] T. A. Zimmerman, R. A. Allen, R. W. Jacobs, Jr., "Digital Charge Coupled Logic (DCCL)," IEEE Journal of Solid-State Circuits, pp 473-384, October 1977