VLSI System for SAR Processing

by

Danny Cohen
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291

and

Vance Tyree
Caltech/JPL
Pasadena, CA 91125

ABSTRACT

Synthetic Aperture Radar (SAR) is a radar system that processes the return signal to achieve the effect of having a larger aperture than the one provided by the physical dimensions of its antenna. The processing consists of a weighted summation of regularly spaced samples from the signal history, hence of logic for the arithmetic and storage for the signal history. LSI and VLSI technology offer some beautiful ways to implement this computation in chips in which the storage and logic functions are commingled.

The SAR problem discussed in this paper is based on actual requirements set forth by NASA for a spaceborne application.

The requirements for high resolution and high quality necessitate a data sampling rate of 7.5 MHz. For each data value 1,025 4-bit complex multiply+add operations are needed, which is equivalent to 7.7 GHz complex multiply+add operation rate. Since this rate is much too high for general purpose systems, a special-purpose device was sought.

This paper discusses two architectures based on parallel operation of 1,025 identical cells, each of which is capable of performing arithmetic, storage, and several control operations. The operation rate in each device is only 7.5 MHz, which is quite manageable, especially with the help of a substantial degree of pipelining.

A computational-mathematical analysis is used as a primary tool for evaluating the design and some of its tradeoffs.

Two different approaches are discussed and compared; both are based on having 1,025 identical cells working in parallel, but differ in their dual approaches to the flow of data. The mathematics require a relative motion of the data with respect to some (relatively) constant sets of coefficients. In one approach the coefficients are held stationary in space, and the data flows past them; in the other, the data is held and the coefficients flow past.

The paper discusses the architecture, both approaches, some of the control issues, and most important, some aspects of the methodology of the design.

BACKGROUND

Synthetic Aperture Radar (SAR) is a radar system that uses its own motion and information processing capabilities to achieve an effective (virtual) radar aperture which is much larger than the physical aperture provided by its antenna.

Why should the aperture be made larger? Or, what is wrong with the conventional circular-scan radars such as those carried in the noses of most aircraft?

The answer to this obvious question can be found in [1]. It is copied here verbatim:

> Experience has shown that most of the images made by circular-scan radar systems aboard airplanes are poorly defined.
>
> The poor definition results from a fundamental reason: Most airborne circular-scan radar antennas are rather small, and fine angular resolution can be obtained only with an imaging system that has a large aperture with respect to the wavelength of radiation received. In other words, the resolution of a large-aperture lens or antenna is finer than the resolution of a small-aperture lens or antenna. The limiting angular resolution is proportional to the ratio between the wavelength received and the size of the aperture.

In conventional optics, the larger the aperture, the higher the quality obtained for certain given conditions. Similarly, in the SAR case, the larger the aperture, the higher the resolution, and the less energy required to obtain a desired signal-to-noise ratio.

The basic idea is very simple. The higher quality image of any ground position is computed from all the radar returns (echoes) from it. The multitude of returns is due to the width of the radar beam and to the motion of the platform relative to the planet. The image obtained by storing these reflections and adding them coherently is better than the one obtained from conventional systems.

The reader interested in the theory and the details of SAR technology is advised to read the article on side-looking radars in Scientific American [1] and the less popular and more detailed literature mentioned in references [2] through [6].

THE MATHEMATICAL PROBLEM

The following is a description of the SAR processing problem, substantially simplified for the purpose of this discussion. The result of the computation is a ground texture map of a swath roughly parallel to and considerably to one side of the track followed by the platform. One simplification employed in this description is to neglect the effects of the altitude of the platform. In the description which follows, think of yourself as looking down on a platform which is moving on a railroad track, the X axis, at velocity v. A map of the area to one side of the track is to be constructed.

At the times $t_i = iNT$, for $i=0,1,2,\ldots$ a radar beam is transmitted in the Y-direction. At the times $t_{i,j} = (iN+j)T$, for $i=0,1,2,\ldots$ and for $j=0,1,\ldots,N-1$, both the magnitude and the phase of the return are recorded. Let $D_{i,j}$ denote the data recorded at the time $t_{i,j}$.

The set {i,*} is called the ith vertical column, and the set {*,j} is called the jth horizontal row.

The sampling period, T, is chosen such that the required image pixel spacing along the Y-direction, Py, is achieved. The relation between T and Py is $Py = \frac{1}{2}Tc$, where c is the speed of electromagnetic propagation. The factor of $\frac{1}{2}$ is due to the reflection. Hence the sampling period is $T = \frac{2Py}{c}$ and the sampling rate is $f = \frac{c}{2Py}$.

The data $D_{i,j}$ corresponds to the return from the ground position (x,y).

Conventional radar uses $D_{i,j}$ for $F_{i,j}$, the image corresponding to (x,y). However, the SAR system computes $F_{i,j}$ by a coherent adding of m returns from each side of (x,y). Hence

$$F_{i,j} = \sum_{k=-m}^{m} a_{k,j} D_{i-k,j}$$

The number of multiplications required for each point $F_{i,j}$ is 2m+1. Since a new value of $D_{i,j}$ is recorded every time period T, the multiplication rate required is $r = (2m+1)/T = (2m+1)f$.

It is worth mentioning again that this description is an extreme simplification of the real problem. Among the factors omitted for the sake of simplicity and clarity are the effects of the angle between the planet motion and the platform velocity and the distance variation of each surface position from the system as a function of j and k.

In addition, the system is described here as if the platform is at ground level (whereas 800 Km is a typical altitude), as if the $\{t_{i,j}\}$ are uniformly distributed (whereas typically there is an inter-beam waiting period for range gating etc.), and as if the pixel spacing is based only on the range (rather than the slant-range).

These simplifications are made here since they do not change the basic concepts of the system. As a matter of fact, the system which is now being VLSI-implemented without benefit of these simplifications is very similar to the one described in this paper.

The relation between the pixel-spacing and the system resolution depends on many factors (e.g., the value of m). The exact relation is also left out of this paper. It is sufficient, for the purpose of this paper, to assume that the resolution is close to the pixel spacing.

TYPICAL NUMBERS

The following numbers used as examples are taken from the requirements for SEASAT-A, which may be found in reference [10].

Py = 20 meters N = 1,024
f = 7.5 MHz m = 512 points on each side

This implies a multiplication rate of $r = (2m+1)f = 1{,}025 \times 7.5\text{MHz} = 7.7\text{GHz}$

First, the bad news about these numbers. Since both the coefficients $\{a_{i,j}\}$ and data $\{D_{i,j}\}$ are complex quantities, each multiplication requires 4 real multiplications. Therefore, the rate of real multiplications is about 30.75 GHz.

The good news is that both the coefficients and the data are handled with only 4 bits of significance. However, the accumulation is performed in 18-bit complex arithmetics.

DISCUSSION

Let Z be the operator which delays data by the time period T. However, Z does not affect the constant coefficients.

Hence $Z\,D_{i,j} = D_{i,j-1}$ for $j>0$, and $Z\,D_{i,o} = D_{i-1,N-1}$
Similarly $Z^N D_{i,j} = D_{i-1,j}$, but $Z\,b_{i,j} = b_{i,j}$, where the b's are constant coefficients.

By definition, we have

$$F_{i,j} = \sum_{k=-m}^{m} a_{k,j}\, D_{i-k,j} = \sum_{k=-m}^{m} a_{k,j}\, Z^{kN}\, D_{i,j}$$

This is mathematically (i.e., formally) correct. However, it is unrealizable, since it involves negative powers of the operator Z. Since Z means delay, the operator Z^{-1} means prediction. Since we do not know how to build the prediction operator, it must be circumvented.

F can also be represented by

$$Z^{mN} F_{i,j} = \sum_{k=-m}^{m} a_{k,j} Z^{(m+k)N} D_{i,j}$$

Define $h \triangleq k+m$, and obtain

$$Z^{mN} F_{i,j} = \sum_{h=o}^{2m} a_{h-m,j} Z^{hN} D_{i,j}$$

Define $b_{k,j} \triangleq a_{k-m,j}$ and obtain

$$Z^{mN} F_{i,j} = \sum_{k=o}^{2m} b_{k,j} Z^{kN} D_{i,j}$$

This means that by implementing the operations shown on the right-hand side of the above equation, the image function F obtained is delayed by mNT. This is not surprising, since the definition of F requires m neighboring data values on each side. Hence, let $G \triangleq Z^{mN}F$ represent the delayed image function.

APPROACH (A)

The implementation of $G = \sum b_{k,j} Z^{kN} D$ is discussed and analyzed in [7]. It is shown there that an optimal implementation (with respect to a set of design objectives defined in that reference) is realized from the following presentation

$$G_{i,j} = \sum_{k=o}^{2m} Z^{k} \, b_{k,j} \, Z^{k(N-1)} \, D_{i,j}$$

This computation can be implemented by the circuits consisting of 2m+1=1,025 cells as shown in figure 1. Please remember: the Z^k does not affect the b's.

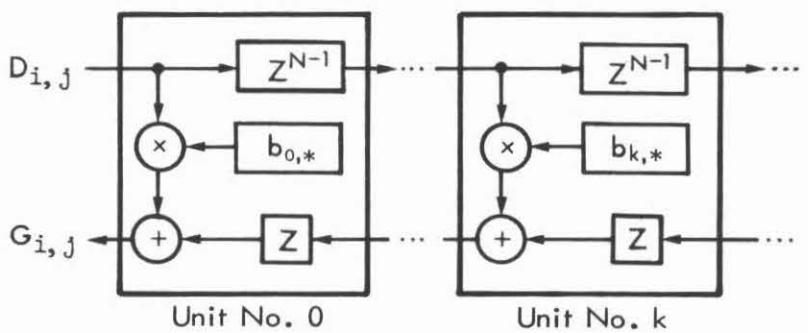

Figure 1: The cells for approach (A).

This implementation is systolic[1] and is relatively easy to implement in VLSI. The complex multiplication rate of each cell is only

$$r' = r/1,025 = 7.5 \text{ MHz}$$

---

[1] H.T. Kung and C.E. Leiserson, in [8], write, "A systolic system is a network of processors which rhythmically compute and pass data through the system. Physiologists use the word "systol" to refer to the rhythmically recurrent contraction of the heart and arteries which pulses blood through the body. In a systolic computing system, the function of a processor is analogous to that of the heart. Every processor regularly pumps data in and out, each time performing some short computation, so that a regular flow of data is kept up in the network."

which for only 4-bit real terms is well within the performance range of off-the-shelf, commercially available LSI multipliers [9].

It is possible to use pipeline multipliers because the data can easily be arranged such that it is available serially, with the least significant bits leading, and because the delay introduced by the pipeline is insignificant.

This approach allows the application of even slower circuits, which may be beneficial for power and size considerations.

Figure 2 shows how the coefficients, $\{b_{k,j}\}$, can be stored in sequential memory (circular shift registers) rather than in random access memory. This also may be beneficial for power and size considerations.

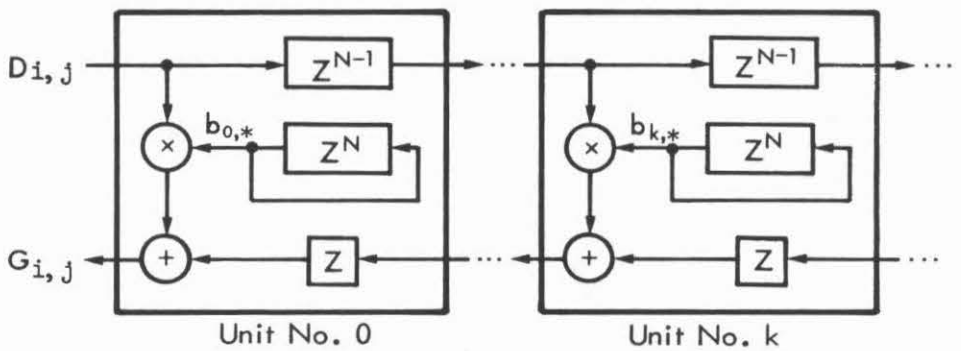

Figure 2: Approach (A), with circular shift registers for the b's.

In summary, 2m+1 cells, arranged in a linear sequence, are used. The kth cell (for k=0,1,2...2m) computes the contribution of the (m-k)th column ahead/ago, to the image of the current position column. This cell stores 2N complex data quantities: N coefficients, $\{b_{k,*}\}$, N-1 input values, and 1 partial sum.

Each device performs one complex multiplication and one complex addition at the sampling rate, f.

APPROACH (B)

This approach is dual, in a way, to the previous one. Whereas in (A) each cell computes a single selected phase of all the image columns $\{G_{i,*}\}$, in (B) each cell computes all the phases of selected image columns only. As before, there are 2m+1 devices, numbered k=0,1,...2m.

From now on all the first indices (such as the "i" in $b_{i,j}$) are computed in modulo (2m+1) arithmetic.

In this approach the kth cell computes the image columns $G_{i,*}$ for all values of i, such that $i \equiv k \pmod{2m+1}$.

Hence, if the entire image is considered as a series of "frames", each composed of 2m+1 vertical image columns, then each cell produces all the images which belong to a certain column in all the frames.

The basic module of the system, according to this approach, is an accumulator as shown in figure 3.

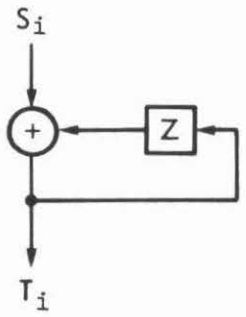

Figure 3: The basic accumulator.

It is easy to see that

$$T_i = S_i + T_{i-1} = S_i + Z\ T_i$$

$$T_i - Z\ T_i = (I-Z)\ T_i = S_i$$

$$T_i = (I-Z)^{-1} S_i = \sum_{k=o}^{\infty} Z^k\ S_i$$

This holds, obviously, if $Z^k=0$ for some $k>0$.

This shows, not very surprisingly, that each T_i is the sum of some of the previous input values $\{S_j \mid k<j\leqslant i\}$.

If the input sequence is $\{S_{i,j}\}$ for $i=0,1\ldots$ and $j=0,1\ldots(N-1)$, and if the Z is replaced by Z^N (a shift register of length N), then

$$T_{i,j} = \sum_{k=o}^{\infty} Z^{kN} S_{i,j}$$

Hence, the column $T_{i,*}$ is the sum of some of the previous columns.

In order to use this accumulator for the SAR processor the $\{S_{i,j}\}$ should be the products of the input values $\{D_{i,j}\}$ by the coefficients, and the summation should include only 2m+1 terms.

In order to limit the range of the summation the cell is modified such that the input to the accumulators-column, Z^N, is cleared at all times (i,j) whenever $i\equiv k \pmod{2m+1}$. Hence, this occurs for N successive cycles, every N(2m+1) cycles, or for one column-period every frame-period.

The modified cell is shown in figure 4. Let $G^{(k)}_{i,j}$ be the output of the adder at $t_{i,j}$, in which the raw data $D_{i,j}$ is multiplied by the coefficient $b_{u,v}$, for some yet undetermined values of u and v.

One can verify that

$$G^{(k)}_{i,j} = \sum_{h=o}^{\beta} b_{u,v}\, D_{i-h,j} \quad \textbf{where } \beta\equiv i-k-1 \pmod{2m+1} \textbf{ and } 0\leqslant\beta\leqslant 2m$$

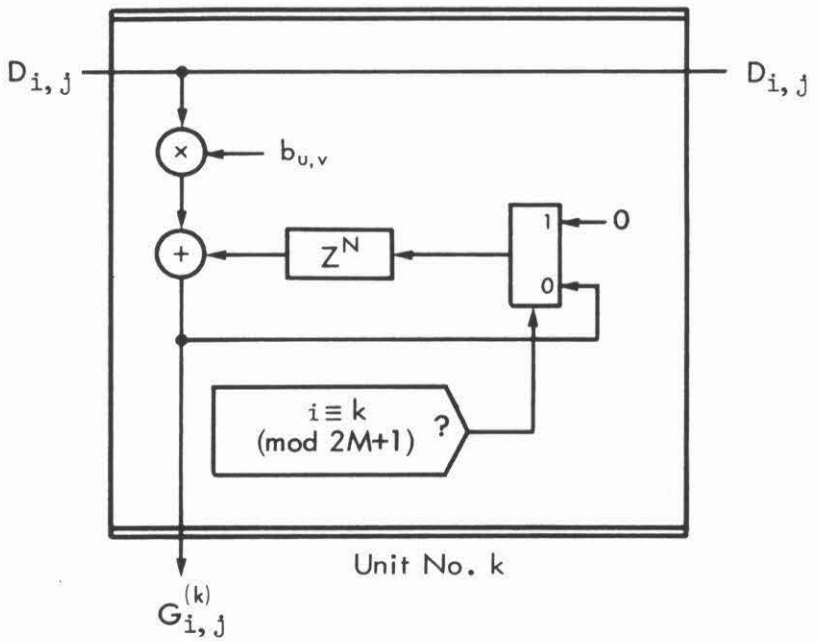

Figure 4: The modified cell, for the kth phase.

Hence, at the time $t_{i,j}$ the $G^{(k)}$ is the weighted sum of the previous i-k (mod 2m+1) columns. In order to have $G^{(k)} = G$, we consider it only when $\beta = 2m$, i.e., when $i \equiv k \pmod{2m+1}$, and get

$$G^{(k)}_{i,j} = \sum_{h=o}^{2m} b_{u,v} D_{i-h,j}$$

If u and v are chosen to be $u = k-i+h \pmod{2m+1}$ and $v = j$, then

$$G^{(k)}_{i,j} = \sum_{h=o}^{2m} b_{h,j} D_{k-h,j} = G_{i,j} \quad \textbf{when } i \equiv k \textbf{ (mod 2m+1)}$$

We have

$$G_{i,j}^{(k)} = \sum_{h=o}^{\beta} b_{k-i+h,j}\, D_{i-h,j} =$$

$$= \sum_{h=o}^{\beta} b_{k-i+h,j}\, Z^{hN}\, D_{i,j} = \sum_{h=o}^{\beta} Z^{hN}\, b_{k-i+h,j}\, D_{i,j}$$

Introduce Z_b, the delay operator which operates only on the coefficients (the b's) similarly to the operator Z, which operates only on data

$$G_{i,j}^{(k)} = \sum_{h=o}^{\beta} Z^{hN}\, Z_b^{-(k+h)N}\, b_{-i,j}\, D_{i,j} = Z_b^{-kN} \sum_{h=o}^{\beta} Z^{hN}\, Z_b^{-hN}\, b_{-i,j}\, D_{i,j}$$

This expression requires some interpretation: since the $\{b_{i,j}\}$ are <u>known</u> coefficients, negative powers of the delay are allowed. Since the first index is computed modulo 2m+1, a single circular shift register of length N(2m+1) can be used to store all the $\{b_{i,j}\}$.

Since the "nominal" b in the above expression is $b_{-i,j}$ the $\{b_{i,j}\}$ are arranged such that each $b_{i,j}$ is followed by $b_{i,j+1}$ (like the data, $D_{i,j}$), but the entire column $b_{i,*}$ is followed by the column $b_{i-1,*}$ (unlike the data).

Since the kth cell requires the phase -kN (as suggested by the term in front of the $\sum$-sign), each cell taps the circular shift register N units apart. Hence each cell contains a shift register of length N for the data (Z^N) and a shift register, also of length N, for the coefficients (Z_b^N).

However, only when $i \equiv k \pmod{2m+1}$ is the output $G^{(k)}_{i,j}$ a valid value for the required $G_{i,j}$. Therefore, the kth cell is allowed to announce its output only then. This permission-to-announce is implemented with a tri-state driver. The kth device therefore has the structure shown in figure 5.

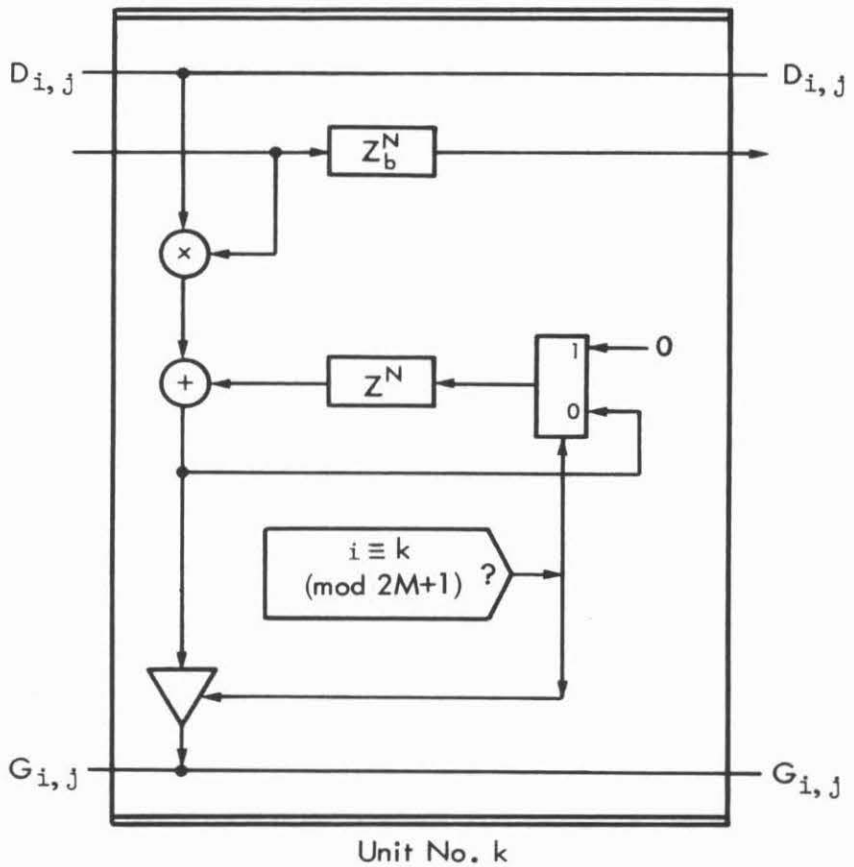

Figure 5: The kth cell for approach (B).

In summary, 2m+1 cells, arranged in a linear sequence, are used. The kth cell (for k=0,1,2...2m) performs all the computation for all the image columns $G_{i,*}$ such that $i \equiv k \pmod{2m+1}$. This cell stores 2N complex quantities: N coefficients, $\{b_{i,j}\}$, and N partial sums.

Each cell performs one complex multiplication and one complex addition at the sampling rate, f.

The linear sequence arrangement is needed only for the coefficients. However, any other arrangement (e.g., tree) can be used for the input data $\{D_{i,j}\}$ and the output image, $\{G_{i,j}\}$.

ABOUT THE IMAGE PIXEL SPACING

The horizontal pixel spacing, Px, of the system is Px=NTv, whereas the vertical pixel spacing, Py, is $Py = \frac{1}{2}Tc$. The ratio, R, between these pixel spacings is

$$R = \frac{Py}{Px} = \frac{Tc}{2NTv} = \frac{c}{2Nv} .$$

For the numbers used here, and for v = 36,000 Km/h = 10^4 m/sec (which is about 22,500 mph) we get

$$R = \frac{c}{2Nv} = \frac{300,000,000}{2\times1,025\times10,000} \approx 15$$

This suggests that it is possible to reduce the output in the X-direction by a factor P (which does not exceed R) without degrading the image quality.

Obviously, it is desirable to take advantage of the reduced requirement for output, and to reduce the computation accordingly.

This advantage can be achieved by computing only every Pth image column, $\{G_{i,*}\}$, e.g., for $i \equiv 0 \pmod P$.

COMPARISON OF THE TWO APPROACHES

The single most important issue of the architecture of the SAR processor is the dynamics of the data flow (shall we call it data-dynamics?). It is clear that the data and the coefficients must flow past each other.

The first approach keeps all the coefficients for each image-column, $\{b_{k,*}\}$, always in the same device and keeps all the data flowing past them. Hence, relatively stationary coefficients with dynamic data.

The second approach keeps all the data required for computing a certain image-column in the same device and keeps all the coefficients flowing past them. Hence, relatively stationary data with dynamic coefficients.

Approach (A) is completely laminar flow systolic system, composed of 2m+1 devices, each of which is systolic, too. Approach (B) is a periodic-laminar flow systolic system, composed also of 2m+1 devices which are systolic only periodically.

The operation control is not discussed here for either approach. In both cases it is simple and straightforward if both broadcast and daisy-chained connections are used. The problem of assigning an individual identity (e.g., the value of k) to each device, dynamically, in spite of identical hardware can be solved in any of several ways, for either approach.

Generally, the operation control is similar (in complexity, connectivity etc.) for both approaches.

Dynamic data allows simpler access to the resulting $G_{i,j}$, because it is always produced by the same device. One may notice that approach (A) does not require the tri-state output drivers which (B) requires. This is an advantage of (A) over (B).

Dynamic coefficients, as in approach (B), significantly simplify the process of changing the coefficients, $\{b_{i,j}\}$, (when needed due to changes of the flight parameters) because all the coefficients circulate through a single

loop, which allows for easy external injection of the new set of coefficients. This is not just a simplification, but also a simple way to achieve a synchronous and an instantaneous change. Therefore, this is a significant advantage of (B) over (A).

In (A) an arithmetic malfunctioning (multiplication or addition) in a single cell, affects all the image points, whereas in (B) it affects only the columns computed by this particular cell. Hence, (B) is much more robust than (A).

Suppose that $\{G_{i,j}\}$ is to be computed only for a certain subset of columns, for example only for one column in P (i.e., i=nP). How does this affect the architecture? In approach (A) no saving of processing hardware can be achieved without major modifications of the architecture, whereas in (B) only the devices corresponding to these columns have to be implemented. However, the Z_b^N of each eliminated cell should be included in the system such that a total $Z_b^{N(2m+1)}$ circular shift register is maintained.

The last three considerations are overwhelming reasons to prefer approach (B) over (A). Therefore, the system which is currently being implemented in VLSI is based on (B).

VLSI IMPLEMENTATION

The control signalling of the system can be based on the application of two regimes of communication in the system: one is a "hop-by-hop" communication, like a daisy chain, between the cells, and the other is based on simultaneous broadcasting to all the units.

In order to reduce drastically the total power consumption of the system, a technology is being developed to allow these devices (each being a VLSI chip, e.g., a "lot" of silicon), to be interconnected without having to cut the wafer. This necessitates checking each device on the wafer. Printing metal connections on the wafer makes it possible to connect the good ones and skip the bad ones.

There are many other important details: the organization of a linear array on a round wafer, loading the coefficients into different memories (for the former approach, only), dynamically assigning identification to the devices, increasing system robustness, and so forth.

These details are very important, and have conceptually simple solutions. We did not find it necessary to include them here.

Jet Propulsion Laboratory was funded in FY77 to develop and demonstrate real-time SAR processor technology that would enable on-board spacecraft SAR processing. A custom VLSI implementation of approach (B) described in this paper will be an important factor in enabling an on-board SAR processor. This VLSI device, the Azimuth Correlator Device (ACD), is being designed and fabricated at TRW, Inc., and contains all of the functional elements of approach (B) (figure 5) with some additional circuitry to perform control functions and to correct far migration of image elements through the processed aperture.

The range migration results from a simple geometric relationship between the SAR instrument and an image element on the surface of the earth (or other planet) that results in a change in surface element range as the surface element occupies different azimuth positions within the area illuminated by the radar antenna.

A functional block diagram of the ACD (figure 6) shows the Range Migration Compensation (RMC), which is divided into coarse and fine components (RMC-C and RMC-F). The Azimuth Reference Function (ARF) coefficients and the RMC coefficients are stored in shift register memory, which also serves as the delay operator in the ACD. The ACD contains a complex multiplier, a complex adder, shift register memory to store coefficients and partially processed image elements, and control and timing circuitry.

A more detailed description of this VLSI device is contained in [11]. It is expected that the first lot of ACD chips will be available for testing in the third quarter of 1979.

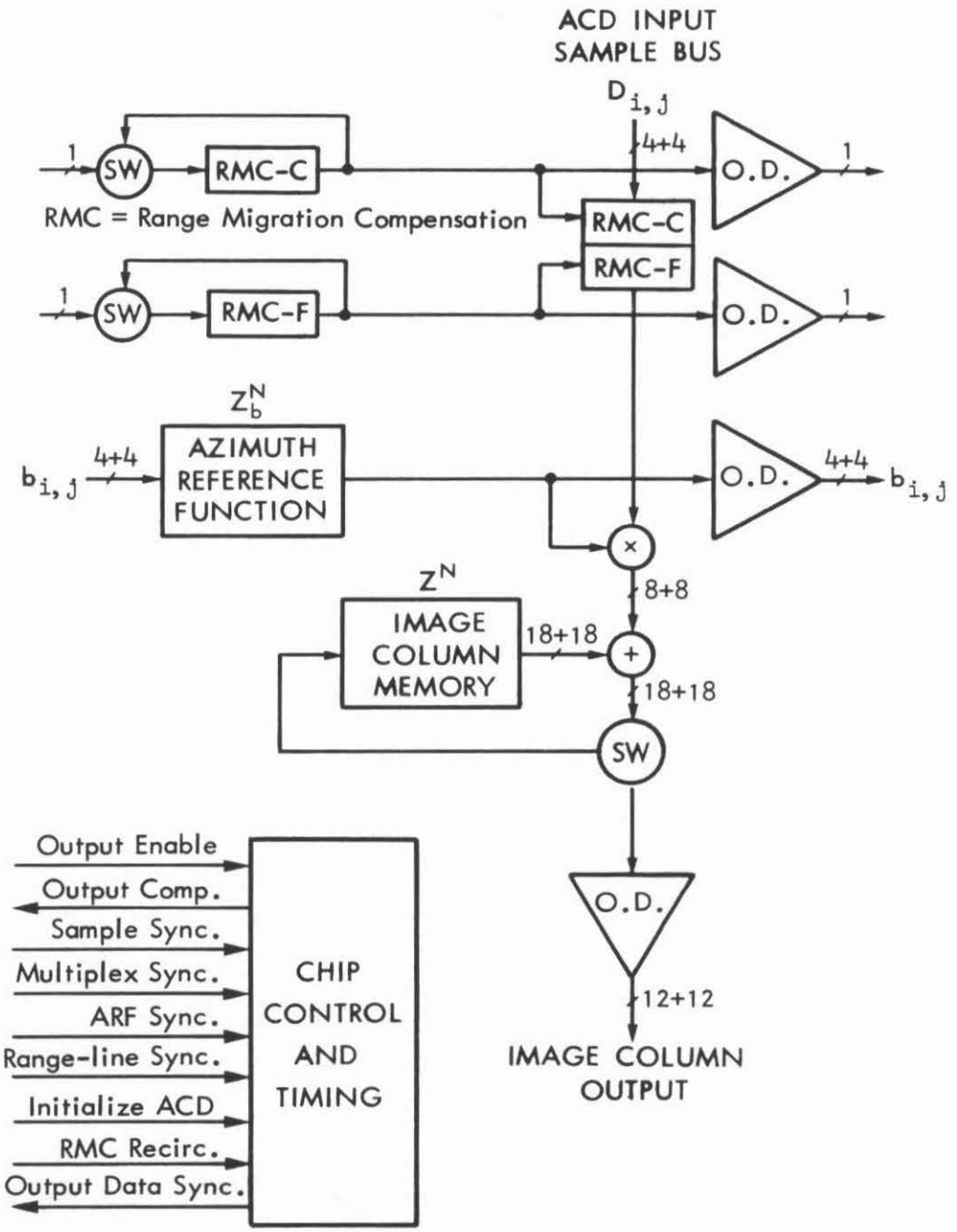

Figure 6: A block diagram of the Azimuth Correlator Device.

CONCLUSION

SAR processing requires a high rate of arithmetic operations and a substantial amount of data storage.

It is possible to implement a SAR processing system based on a multitude of identical cells, all working in parallel, at full capacity, such that they share uniformly the required system operating rate.

In addition, it is possible to organize such a system with only a small number of interconnections.

Therefore, VLSI is most suitable for SAR processing.

ACKNOWLEDGMENTS

The authors would like to extend their appreciation to C. Wu, R.G. Pierson, W.E. Arens and R. Lipes for their contributions to the development of the SAR processor studies that lead to the conception of Approaches (A) and (B). The generalized mathematical representation of a SAR azimuth processor described in this paper was an outgrowth of these earlier studies.

BIBLIOGRAPHY

[1] Jensen, H., Graham, L.C., Porcello, L.J. and Leith, E.N., "Side-looking Airborne Radar", Scientific American, October 1977, 84-95.

[2] Cutrona L.J., Leith, E.N., Porcello, L.J. and Vivian, W.E., "On the Application of Coherent Optical Processing Techniques to SAR", in Proceedings of the IEEE, Vol. 54, No. 8, August 1966, 1026-1032.

[3] Brown, W.M. and Porcello, L.J., "An Introduction to Synthetic Aperture Radar", IEEE Spectrum,, Vol. 6, No. 9, September 1969, 52-62.

[4] Synthetic Aperture Radar, ed. John Kovaly, Artech House, 1976.

[5] Synthetic Aperture Radars, Theory and Design, by Robert O. Harger, Academic Press, 1970.

[6] Proceedings of the Synthetic Aperture Radar Technology Conference, March 8-10, 1978, Las Cruces, New Mexico. This issue can be obtained from: SARTC Publications Committee, Physical Sciences Laboratory, Box 3-PSL, Las Cruces, New Mexico, 88003.

[7] Cohen D., "Mathematical Approach to Iterative Computation Networks", Proceedings of the 4th Symposium on Computer Arithmetics, October 1978, Santa Monica, California, pp. 226-238, IEEE Catalog No. 78CH1412-6C. Also available as USC/Information Sciences Institute Research Report RR-78-73.

[8] Kung, H.T. and Leiserson, C.E., "Systolic Arrays (for VLSI)" to appear in Introduction to VLSI Systems by C.A. Mead and L.A. Conway, Addison-Wesley, 1979.

[9] Data sheets for the LSI multipliers MPY-16A, MPY-16AJ and TDC1010J, TRW LSI products, Redondo Beach, California, 1978.

[10] Rolando, J., "The SEASAT-A Synthetic Aperture Radar Design and Implementation", Proceedings of the Synthetic Aperture Radar Technology Conference, March 8-10, 1978, Las Cruces, New Mexico.

[11] Tyree, V., "Custom Large Scale Integration Circuit for Spaceborne SAR Processors", Proceeding of the Synthetic Aperture Radar Technology Conference, March 8-10, 1978, Las Cruces, New Mexico.