

SYSTOLIC ARCHITECTURES BASED ON BARREL SHIFTERS
FOR REAL-TIME SIGNAL AND IMAGE PROCESSING

P. A. RAMAMOORTHY AND TAU CHEN

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF CINCINNATI
CINCINNATI, OH 45221

ABSTRACT

The throughput in real-time digital signal processing (DSP) applications is limited by both the capability of the processors employed for number-crunching operations and the capacity of a supporting communications link. The systolic architectures eliminate the memory bandwidth problems by allowing multiple computations for each memory access and hence makes possible a high throughput in real-time applications.

In conventional systolic arrays, the computational element includes a multiplier and an accumulator (MAC). The multiplier in the basic cell requires either large chip area if high speed is desired or time consuming if serial architecture is used. The replacement of the multiplier by the barrel shifter has been proposed in this paper. The new basic cell consists of a barrel shifter and an accumulator (BSAC). By the variations of the connection among the basic cells, the throughput data rate can be increased significantly. Also, a large reduction in the number of the gates can be achieved. The results obtained indicate that the BSAC-based systolic arrays can outperform the conventional ones and achieve throughput data rate of the order of 100 MHz or higher.

I. INTRODUCTION

The concept of systolic architecture has developed as a general methodology for mapping high-level computations into hardware structures. This architecture is particularly advantageous from the very large scale integration (VLSI) fabrication for its high degrees of simplicity, regularity and sparing use of memory bandwidth.

Systolic architectures make use of pipelined multiprocessing to obtain high throughput in real-time applications. In a systolic system, once the data is fetched from the memory, it passes through many processing elements to permit completion of computations based on that data. Hence, the system memory bandwidth problem is eliminated and a speed-up in the execution time of compute-bound problems without increasing I/O requirements is achieved. Nevertheless, in systolic array, the compute-bound computations are still limited by the processing capabilities,

i.e., the data flow through the array is limited by the computational time needed for one basic cell.

In conventional systolic arrays, the computational element includes a multiplier, an accumulator and registers for controlled transfer of data from one cell to the other. The multiplier in the basic cell requires either large chip area if high speed is desired or time consuming if serial architecture is used. The replacement of multipliers by barrel shifters in the basic cells are proposed and discussed. It is shown that not only high degrees of simplicity, regularity and concurrency can be preserved as in the conventional systolic arrays, but also a higher speed-up in the throughput data rate can be achieved. Above all, a large reduction in the numbers of gates required to realize a systolic array can be achieved.

II. THE STRUCTURE OF BASIC COMPUTATIONAL ELEMENT

The basic cell, called MAC, in conventional systolic array is depicted in Figure 1a. They perform the operations (or their variations) given below:

$$X_{out} = X_{in}$$

$$Y_{out} = Y_{in} + A \cdot X_{in} \quad (1)$$

By replacing the multiplier in the above cell with a barrel shifter, one can get a new basic cell, called BSAC, shown in Figure 1b. The basic operations of this new cell will be:

$$X_{out} = X_{in}$$

$$Y_{out} = Y_{in} + k_j \cdot 2^{aj} \cdot X_{in} \quad (2)$$

$$K_j = 1 \text{ or } 0 \text{ or } -1$$

$$-13 \leq a \leq 1 \text{ (for 16-bit fixed-point number)}$$

From Equation (1) and Equation (2), one can easily establish the relationship as following:

$$A = \sum_j K_j \cdot 2^{aj} \quad (3)$$

The values of K_j , a_j can be calculated and preloaded. By using the canonical signed digit code (CSDC) we can have a minimal BSAC cell representation for a given value of A . The CSDC is simply a recording algorithm that minimizes the number of nonzero signed digits in a representation involving only +1, 0 and -1.

The barrel-shifter performs a high-speed, or "flash", operation, shifting its entire contents any number of bit positions within one time clock cycle. Hence we can use a number of barrel shifters and accumulators in conjunction with the CSDC to replace multiply-accumulate operations. For each multiplier the different CSDC may have different number of nonzero digits, i.e., different number of basic cells will be needed for each MAC operation. This nonuniformity leads to different architectures that have different properties in terms of tradeoff in precision, timing and cost.

The actual numbers of barrel-shifters and add/subtract operation that result from a CSDC representation for a given value of A vary quite differently. The average number of BSAC units needed for a N -by- N bit multiplication is given by $N/3$ [1]. From the hardware point of view, an over-all reduction of the number of the gates is possible even if we use $N/3$ BSAC units to replace MAC cell. Also, one can selectively reduce the number of BSAC cells used to replace a MAC cell to get further reduction in the number of gates.

III. FINITE IMPULSE RESPONSE FILTERS' IMPLEMENTATION

Based on the way the inputs can be fed into the basic cells, we can obtain two types structures. For simplicity, we use an example of a second-order FIR filter to demonstrate the idea. Consider the difference equation of the filter is given as below:

$$Y(n) = A(0)*X(n) + A(1)*X(n-1) + A(2)*X(n-2)$$

where $A(i)$ ($i=0,1,2$) are the known filter's coefficients. Suppose for a given precision the number of CSDC expression for $A(0)$, $A(1)$, $A(2)$ are 2, 4, and 3, respectively. Then we can obtain Type I structure, called the cascaded architecture, in which the signal inputs are serially loaded into each basic cells that are necessary for one multiplication. In such a connection, different number of BSAC's for different multiplier values can be used. In Figure 2, we have given the block diagram and the timing sequence for a second-order filter example.

Note that, in Figure 2, we have made some deviation from exact cascading. In Figure 2, the input to the first BSAC unit corresponding to a multiplier coefficient should be the output from the last but one BSAC unit rather than the last unit corresponding to the previous multiplier. From the timing diagram, one can see the

necessity for this deviation. The deviation of the connections among cells guarantee that the throughput is governed by the delay in just one BSAC cell.

Now, the assumption in the conventional systolic arrays that the data rate which is sent to the systolic array is not higher than the computation rate of a basic cell still holds, but that depends on the time delay required in each BSAC computational element, and not the time needed for a whole MAC operation, which can be quite small.

Type II is called parallel architecture in which the inputs enter the basic cells corresponding to one multiplication simultaneously. The number of BSAC basic cells for each multiplication should be the same. Using the same example above, one can synthesize the parallel structure shown in Figure 3. The data flow is quite regular without any branching. However some extra levels of addition are necessary at the output end.

Because the structure need same number of BSAC for each coefficient, we introduce some redundancy to get the desired minimum precision for each coefficient or use less BSAC units resulting in some kind of truncation errors. The trade-off involved is demonstrated in details by way of an example given below.

The example is an 121-order highpass FIR filter whose frequency response is shown in Figure 4a. From the experimental results, Figure 4(b) to 4(d), one can conclude that three BSAC operations for one multiplier in the parallel architecture is quite enough. For the cascaded architecture, one can reduce the number of BSAC operations for one multiplier by eliminating the BSAC operation needed to represent the least significant bits. From figure 4(e) to 4(h), it can be noticed that an average of only 1.3 BSAC operations are needed for each multiplier coefficient to approximate the desired filter specification. The reduction of the gates number for 16-by-16 bit operation can be calculated as follows:

With MAC: $3,168 + 420 = 3,588$ gates

With BSAC

in cascaded structure
 $1.31 * (256 + 420) = 886$ gates

in parallel structure
 $3 * (256 + 420) = 2,028$ gates

where 3,168 is the number of gates needed for 16-by-16 bit multiplier, 420 is the number of gates needed for 32-bit adder and 256 is the number of the gates needed for 16-by-16 bit (32-bit output) barrel shifter. The reduction of the number of the gates can be seen from the above calculations. That along with the fact that BSAC units have a simple structure makes it highly

preferable over MAC based systolic arrays for VLSI implementation.

IV. COMMENTS ON IIR FILTER IMPLEMENTATION

The corresponding rational system function for digital filters has the form

$$H(z) = \frac{\sum_{k=0}^M a(k) z^{-k}}{1 - \sum_{k=1}^N b(k) z^{-k}} = \frac{Y(z)}{X(z)}$$

The input and output are related by the difference equation

$$y(n) = \sum_{k=0}^M a(k) \cdot x(n-k) + \sum_{k=1}^N b(k) \cdot y(n-k)$$

Using the conventional MAC systolic architecture to implement IIR filters, data flows through many processing elements in a rhythmic fashion, and the output can be generated in every other time cycle. The reason for two time cycle requirement is because of the limitation for calculating the sum of the previous outputs' products. In order to gain higher throughput rate, one has to speed up the summing operation or to merge it into some other operation. That is why one can obtain numerous variations in the design to achieve better results.

The following example demonstrates the variations among the basic cells. Suppose

$$y(n) = \sum_{k=0}^2 a(k) \cdot x(n-k) + \sum_{k=1}^2 b(k) \cdot y(n-k)$$

where all the coefficients, $a(k)$ and $b(k)$, are implemented as two BSAC operations. One can have the best variation as shown in Figure 5. Other variations can be obtained in several ways. One of them will be by adding pairs of poles and zeros to achieve the desired precision in the response. Of course one should pay attention to the problem of the stability. Also, when one use the new systolic array in the IIR filter implementation, the branching variation, timing and data flow control can be varied independently.

V. ANALYSIS FOR CASCADED ARCHITECTURE OF DFT IMPLEMENTATION

The DFT implementation using systolic array is as shown in Figure 6 (for a DFT of size 4). Here $W_N = \exp(-j2\pi/N)$ is the twiddle factor.

By using two different assignments for the coefficient W_N^{nk} (i.e. the number of BSAC cells corresponding to each coefficient W_N^{nk}),
 1) uniformly and 2) nonuniformly (equivalent to

parallel and cascaded architecture), the error analysis for a 256-point DFT implementation is shown in Figure 7. Obviously, the cascaded structure is less sensitive to the parallel one for a given number of BSAC cells for each

coefficient W_N^{nk} .

The following example for a 256-point DFT testing is given by a sinusoidal input signal $x(n)=\cos(2n\pi/32)$ (where $n=0$ to 255). The result, shown in Figure 8, indicates that less than three BS for each coefficient can achieve the acceptable precision.

VI. CONCLUSIONS

The use of barrel shifter/accumulator as computational elements of basic cells of systolic arrays is explored in this paper. One can use cascaded or parallel structure with BSAC units to implement different digital processing algorithms. The cascaded structure is found to be better than the parallel one in the precision that is needed for coefficients for FIR filter and DFT implementation. But the parallel structure can be efficient in IIR implementation. Either of the new systolic architectures can lead to a significant reduction in the number of the gates, and also very simple and modular architecture. That is highly preferred for VLSI fabrication. More importantly, the throughput rate can be increased up to 100 MHz or higher in the new systolic arrays.

REFERENCES

1. P.A. Ramamoorthy and T. Chen, "Systolic Architecture Based on Barrel shifter for Digital Filtering", Proc. IEEE Intl. Conf. on Computers, Systems & Signal Processing, Bangalore, India, December 1984.
2. E.E. Swartzlander, Jr., L.S. Lome and G. Hollnor, "Digital Signal Processing with VLSI Technology", Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, pp. 951-954, April 1983
3. H.T. Kung, "Why Systolic Architectures", IEEE Computer, pp. 37-46, January 1982.
4. H.T. Kung, "Special Purpose Devices for Signal and Image Processing: an opportunity in Very Large Scale Integration", proc. Real-Time Signal Processing III, SPIE, pp. 76-84, 1980.
5. C. Mead and L. Conway, Introduction to VLSI Systems, Addison-Wesley Publishing Company, 1980.

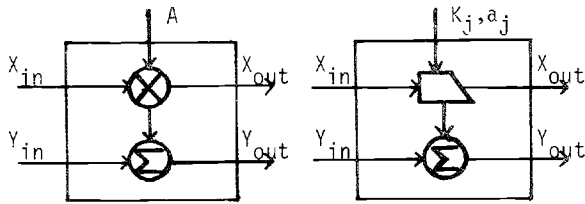


Figure 1. The basic cell in the systolic arrays (a) MAC (b) BSAC structure

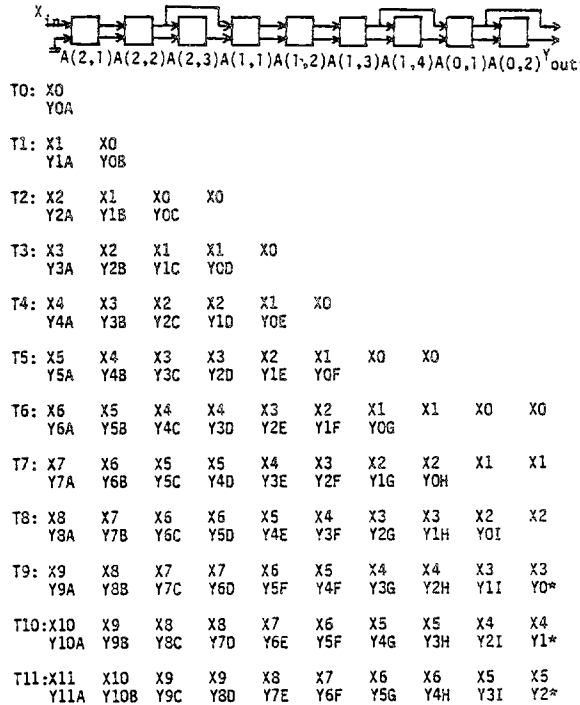


Figure 2. Cascaded structure for a second-order filter example and the timing diagram

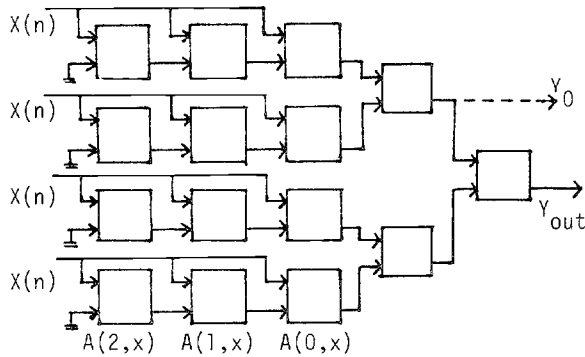


Figure 3. Parallel structure (Y_{out} with some redundancy in the architecture; Y_0 : resulting in some kind of truncation errors)

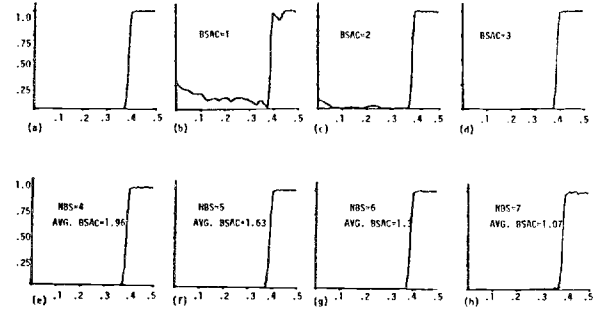


Figure 4. Frequency response: (a)original (b)-(d)parallel structure with different BSAC units for one multiplier, (e)-(h)cascaded structure with reduction of BSAC units.

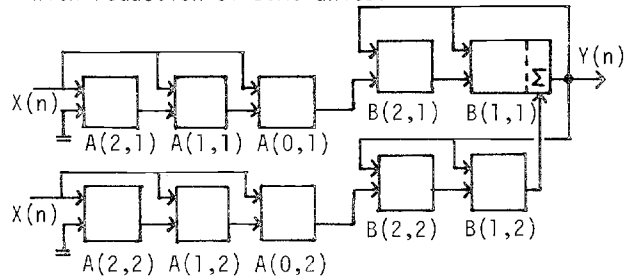


Figure 5. The new systolic array in the IIR filter implementation

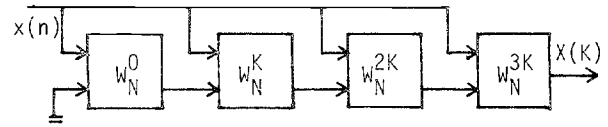


Figure 6. The systolic array for a 4-point DFT implementation

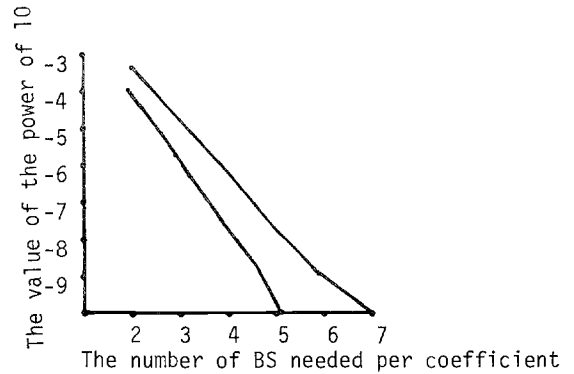


Figure 7. The error analysis for 256-point DFT coefficient with different architectures

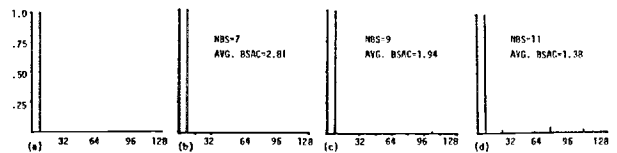


Figure 8. (a)original DFT, and cascaded architecture with BS= (b)2.81 (c)1.94 and (d)1.38 for each coefficient.