《特別寄稿》

A VLSI Architecture for the Real-Time 2-D Digital Signal Processing

실시간 2차원 디지탈 신호처리를 위한 VLSI 구조

권 희 훈
(국립 충주산업대학 전자통신과)

■ 차 례 ■

ABSTRACT

The throughput requirement for many digital signal processing is such that multiple processing units are essential for real-time implementation. Advances in VLSI technology make it feasible to design and implement computer systems consisting of a large number of function units. The research on a very high throughput VLSI architecture for digtal signal processing applications requires the development of an algorithm decomposition scheme which can minimize data communication requirements as well as minimize computational complexity.

The objectives of the research are to investigate computationally efficient algorithms for solution of the class of problems which can be modeled as DLSI(discrete linear shift invariant) systems or adaptive systems, and develop VLSI architectures and associated multiprocessor systems which can be used to implement these algorithms in real-time.

A new VLSI architecture for real-time 2-D digital signal processing applications is proposed in this research. This VLSI architecture extends the concept of having a single processing units in a chip. Because this VLSI architecture has the advantage that the complexity and the number of computations per input does not increase as the size of the input data is increased, it can process very large 2-D date in near real-time.

要 約

다수의 처리 장치가 실시간 실현에 필수적이라는 것이 많은 디지탈 신호처리를 일정한 시간 내에 하기 위한 요구 조건이다. VLSI 기술이 발전함으로 많은 기능 장치로 구성된 컴퓨터 시스템을 설계하고, 실현하는 것이 가능하게 되었다.

일정한 시간내에 높은 처리 능력을 갖음으로서 디지탈 신호처리에 응용할 수 있는 VLSI 구조를 연구하는데 데이타 통신의 요구량과 계산의 복잡성을 최소화 할 수 있는 알고리듬의 개발이 요구된다. 이 문제를 해결하는 방법으로 DLSI 시스템이나 적응 시스템을 모델로 하는 효과적인 알고리듬을 조사하고, 이 알고리듬을 실현할 수 있는 VLSI 구조와 연관된 멀티 프로세서 시스템을 개발하는데 본 연구의 목적이었다.

본 연구에서는 실시간 2차원 신호처리를 할 수 있는 새로운 VLSI 구조를 제안했다. 이 VLSI 구조는 칩 내부에서 단일 처리 장치가 갖는 개념을 다수의 처리 장치를 사용하는 경우로 확장하였다. 이 VLSI 구조는 입력 데이타의 크기가 증가함에 따라서 복잡성과 입력 당 계산의 수가 증가하지 않는다는 장점을 갖기 때문에 매우 큰 2차원 데이타를 실시간에 처리할 수 있다.

I. Introduction

The use of digital signal processing become more and more popular in our everyday life. Digital techniques provide greater precision, repeatability, higher signal-to-noise ratio and greater flexibility than their analog counterparts[1]. The complexity of digital signal processing has risen markedly, following not only theoretical advances in signal processing area but also the rapid advances in integrated circuit technology. The ever-increasing demands for performance, sophistcation, and real-time signal processing strongly support the need for tremendous computational capability.

A solution to the high-speed requirements of signal processing is found in these of special purpose computers[2]. The availability of low cost, high-density, fast VLSI devices make it feasible to build highly concurrent special purpose architectures for signal processing applications. While VLSI holds the promise of high parallelism by offering almost unlimited hardware at very low cost, there are several inherent technical constraints[3]. The fundamental limitation is the high cost of communication, relative to logic and storage. Communication is expensive in terms of chip area, power consumption and time delay both and between chips. Also design complexity, limitation on the number of pins and testability are other constraints on VLSI technology. A high performance for digital signal processing must use a repetitive modular structure and localized interconnections because of the technological constraints on VLSI[4].

In recent years, a great number of digital signal processor(DSP) chips have been developed and used for various applications[5,6,7,8]. They are designed to increase throughput over general purpose microprocessors for digital signal processing computational primitives such as multiplication and accumulation operations. They can be applied to many other systems because the internal instruction data memories are programmable. While these DSP chips provide substantial speed-up for many 1-D signal processing and telecommunication applications compared to standard microprocessors, they are still slow when applied to highly computational and specialized applications such as speech recognition and spectrum estimation. More specialized architectures with multiprocessing capability are desirable for real-time implementation of these signal processing algorithms. Special purpose processors with high efficiency need to be developed through extensive study of algorithms. Such processors will consist of hardware that is optimized for a particular class of algorithms.

Since 1978, when Kung and Leiserson introduced systolic arrays[9], much research has been done and much has been written about the design of algorithms and architectures suitable for such structures[10]. A systolic architecture using a large number of modular and locally interconnected processors for massively pipelined and paralleling, is one of the most promising architectures for high-performance, costeffective digital signal processing systems. However, there are several issues regarding implementation of systolic arrays such as the requirements for global synchronization, inefficient data transfer, and lack of programability[11,12]. As the array size grows larger, the global synchronization requirements are more difficult to implement, and an asynchronous system will give better performance[13].

A wavefront array architecture which combines the features of the asynchronous, data-driven properties of data flow machines and regularity, modularity, and local communication properties of systolic arrays was introduced by S.Y.Kung. The wavefront array operates asynchronously and, therefore, does not require global synchronization. The wavefront architecture can provide asynchronous waiting capacity. Consequently, it can cope with timing uncertainties, such as local clocking, random delay in communications, and fluctuations of computation times. The implemen-

tation of the wavefront concept requires the use of buffers and handshaking protocols for data communications between processors. The self timed asynchronous scheme can be costly in terms of extra hardware and delay in each processor associated with the handshaking overhead [11]. This problem need to be addressed in developing the architectural concepts presented in this research.

The wavefront array is applicable to a large class of algorithms that possess recursivity and locality. A wavefront array architecture with well-designed processing elements is very suitable for efficient and reliable design of large scale computing systems. It is especially appealing for VLSI implementation.

In this research program, a special VLSI architecture with properties similar to those of a wavefront array architecture will be developed. This architecture will use an elastic first-in first-out(FIFO) type buffer between processors in order to offset the time delay associated with handshaking overhead. Since the dataflow requirements will be predetermined by a given algorithms classified as discrete linear shift invariant (DLSI) system.

II. Computational Primitive

Many digital signal processing problems can be modeled as DLSI systems, for which the system parameters do not vary with changes in the independent variables(time, space, distance, range, etc.). In addition, many shift variant systems which are usually represented by partial differential equations can be approximated over small intervals of the independent variables as DLSI systems. DLSI systems are typically solved with the use of finite difference equations. Instead of directly solving these equations, a state space representation can be used to minimize the data communication requirements of the system without increasing computational complexity. By properly assigning a state variable to parameter that must be saved for later computation, a set of order difference equations can be devoloped to represent the original algorithm. Then, the state variables and the current output can be computed using a linear combination of the most recent state variables and the current input. Once all the previous state variables have been computed, the current state variables and the current output can be computed in parallel. In addition to allowing parallel computations, this approach reduces the data communication requirements for implementing the algorithm.

2.1. Computational Primitive for 2-D DLSI System

In a previous research, a promising approach to the design of special purpose systems for 2-D digital signal processing was developed [1]. Concentrating on the set of DSP algorithms which can be classified as DLSI systems, a state space model for the DLSI system was developed. Then, all of the computations required to implement the state space model were mapped into a single computational primitive. Essential procedures leading to the computational primitive of DLSI systems are summarized as follows.

A set of finite difference equations is one of the forms commonly used for representing DLSI systems. A general, causal 2-D DLSI system with quarter plane support can be represented by finite difference equations[15] as given by equation (1).

$$g(m,n) = \sum_{j=0}^{I}\sum_{k=0}^{J} a(j,k)f(m-j,n-k) - \sum_{j=0}^{I}\sum_{\substack{k=0 \\ (j,k)\neq(0,0)}}^{J} b(j,k)g(m-j,n-k) \quad (1)$$

The parameters a(j,k) and b(j,k) in above equation are coefficients which determine the characteristics of system. Since the coefficient can take on values, this equation can be represent many 2-D DLSI problems including spatial domain filters, image processing, simulation, control systems, etc. A 2-D DLSI transfer function

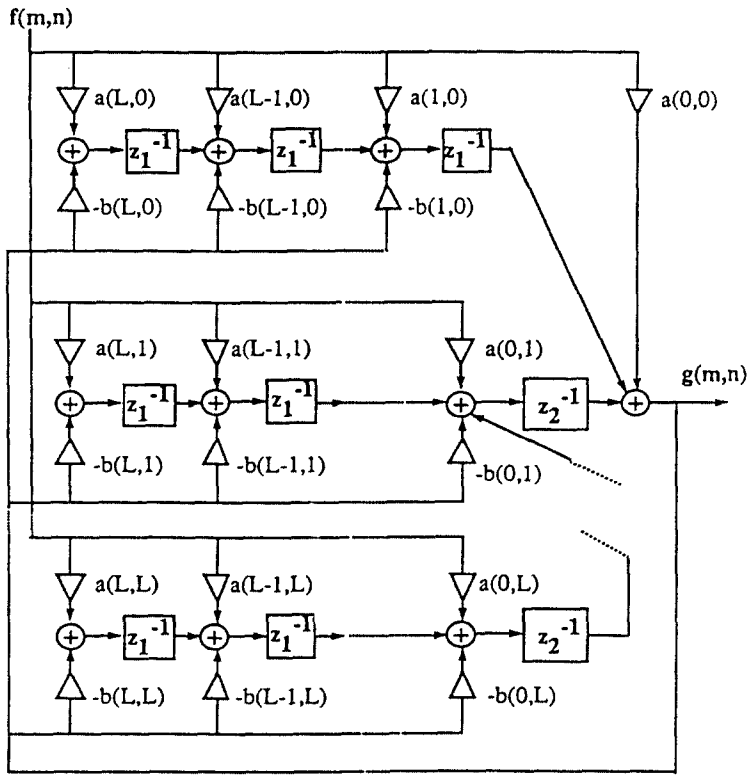

Fig. 1. A block diagram for a general order 2 D DLSI system.

corresponding to equation (1) is given by

$$H(z_1,z_2)=\frac{\sum_{j=0}^{L}\sum_{k=0}^{L} a(j,k)z_1^{-j}z_2^{-k}}{1+\sum_{j=0}^{L}\sum_{\substack{k=0\\ j+k>0}}^{L} b(j,k)z_1^{-j}z_2^{-k}} \qquad (2)$$

Note that $H(z_1,z_2)$ describes an input-output relationship between the transform of the input sequence, $F(z_1,z_2)$, and the transform of the output sequence $G(z_1,z_2)$. This relationship can be written as follow:

$$G(Z)=a(0)F(Z)+\sum_{\substack{j=0\\ j+k>0}}^{L}\left[\sum_{k=0}^{L}[a(K)F(Z)-b(K)G(Z)]z_1^{-j}\right]z_2^{-k} \qquad (3)$$

Fig. 1 gives a block diagram representation of the 2-D DLSI system partitioned as a specified by equation (3). The number of vertical delays is the same as the order of the filter in the z_2 variable which is the minimum possible number. A state space representation can be obtained by assigning a horizontal state variable to the input of the horizontal delay blocks (associated with z_1 variable) and by assigning a vertical state variable to each of the vertical delay blocks (associated with the z_2 variable).

The computational primitive for the general order 2-D DLSI system can be obtained from the state equations for the vertical state variables and is given by equation (4).

$$q_l(m,n)=c_k{}^{*}f(m,n)+d_k{}^{*}y(m,n)+r_l(m-1,n)$$

$$+q_{1\ 1}(m,n-1). \quad (4)$$

In the above equation, $q_1(m,n)$ represents the current value of the state variable or the output as appropriate, $r_1(m-1,n)$ represents a previous value of the horizontal state variable, $q_{1\ 1}(m, n-1)$ represents a previous value of the vertical state variable, f(m,n) represent the current input, y(m,n) represents a feedback term which is precomputed to simplify the computations and c_k and d_k are appropriate system coefficients. All the computational requirements for the 2 D DLSI system can be mapped into this computational primitive which requires two multiplications and three additions [11]. Thus, 2 D DLSI systems can be implemented by solving equation (4) repeatedly with proper parameter substitutions for each computational requirement. A special processor can be designed from this computational primitive to implement general order 2-D DLSI systems.

2.2. Modified Computational Primitive

The computational primitive can be modified to improve performance. One variation is arranging a multiplication, and an addition and a multiplication accumulation structure is given by equation (5).

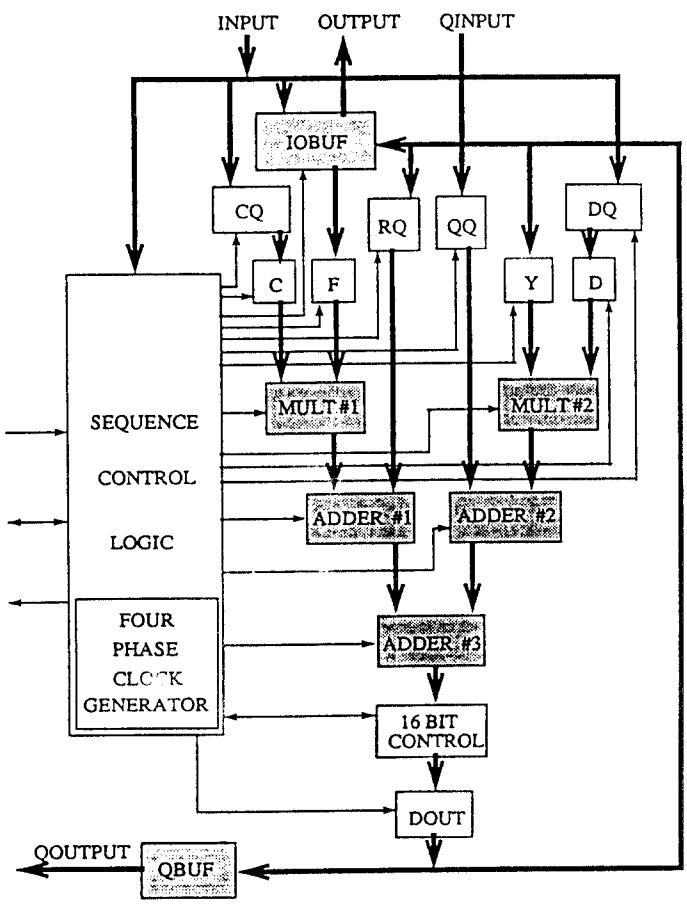

Fig. 2. A block diagram for the previously developed DSP.

$$q_l(m,n)=[c_k*f(m,n)+r_j(m-1,n)]+[d_k*y(m,n)+q_{l-1}(m,n-1)]. \quad (5)$$

This is a more generalized computational primitive since inner product type of operations, such as matrix-vector multiplication can easility be mapped into this structure. Moreover, this arrangement gives additional regulariry for VLSI implementation. A DSP architecture that implements 2-D DLSI systems using the computational primitive given by equation (5) was developed. Fig.2 is a block diagram for the single chip DSP developed in the previous research [16]. The DSP can realized a L^{th} order 2-D DLSI system in real-time using $(L+1)^2+1$ processors.

Ⅲ. Proposed Processor

Similar approach can be used to the development of algorithm decomposition procedures and corresponding VLSI architectures for the other classes of digital signal processing algorithms. This research program uses VLSI as a medium to solve some of the special architectural problems associated with DSP. The ability to fabricate many devices on a single chip provides the opportunity to design complete systems on a chip. The feasiblity of using the state-of-the-art VLSI technology to simplify overall system design will be considered.

The procedure for this research program is as follow:

(1)Investigate or choose the algorithm for a given DSP application.

(2)Develop a computational structure for the algorithm.

(3)Decompose the computational structure into computational primitives.

(4)Develop a macro-signal flow graph(MSFG) for the system with each node in the MSFG having the capability to implement one or more of the computational primitives and handle the required data communications.

(5)Develop a behavioral simulator for the system to determine system performance.

(6)Iterate by going back to previous steps as necessary to improve performance.

(7)Develop VLSI architecture for real-time implementation. Consider the feasibility of using commercial DSP chips. Develop a functional simulator to verify functionality of the system.

(8)Study the feasibility of designing a prototype VLSI chip and an example system to demonstrate architectural features.

3.1. Processor Architecture

The use of computational primitive to implement a 2-D DLSI system results in a regular and modular structure which is a very desirable feature for VLSI implementation. By further specializing the processor architecture, a very high throughput single chip DSP can be developed to implement the 2-D DLSI system in realtime. Based upon previous research, a new processor architecture which cntains 9 multiplier/accumulators in a single chip is proposed. The new DSP architecture will have much simple control scheme and more flexibility than the previous one. For example, single processor system will implement a second order 2-D DLSI system in real-time. It will also be cascadable to implement higher order systems. A brief description of the new DSP architecture is given to demonstrate the architectural concepts. For actual VLSI implementation, further research should be conducted.

3.2. Arithmetic Block

Fig. 3 is a block diagram for an arithmetic unit of the processor. A multiplier and two adder are pipelined to form an arithmetic unit. A high speed 16-bit by 16-bit multiplier forms the first stage of the pipeline. The multiplier loads two 16 bit by integers and generates a 32-bit result at every cycle. Two parallel adders form the second stage of the pipeline. They add proper state variables to the multiplier outputs, accumulate

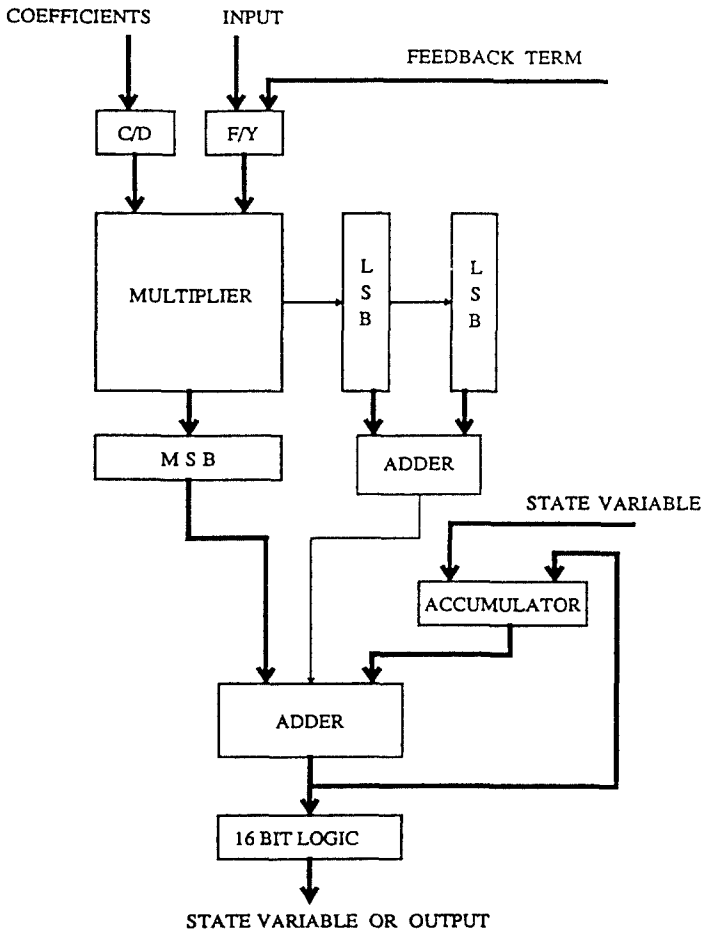

Fig. 3. A block diagram for an arithmetic unit of proposed processor.

the result, and generate a state variable at every cycle.

The output of the pipeline is properly rounded into 16-bit integers by the 16-bit control logic and latched in the results register. The 16-bit control logic implements a saturated arithmetic function and generates overflow status signals for overall scaling operations. A state variable or an output is latched into the results register at every cycle. Thus, each arithmetic unit implements a state equation in a single cycle. Nine of the above arithmetic units are arranged according to the computational requirements in order to implement a second order 2-D DLSI in a single chip. For wider applications, reconfigurability will be considered among arithmetic units in this research.

3.3. Data Block

For maximum throughput, the arithmetic pipeline should be fed continously with appropriate data. Input and output data are kept in the off-chip buffer memory IBUF and OBUF respectively, with one row of data as a block. During the

data processing operations, the current input data is loaded from IBUF into the current input register F in a sequential manner. It will be used to compute the current output and all of the current state variables in parallel. The current output data can be stored in OBUF at every cycle. IBUF and OBUF are used as elastic buffers between the system controller and the processor for efficient asynchronous processor operations. Data transmission can be accomplished without processor interruption by making IBUF large enough to hold one row of data. The processed output data can be written into OBUF at the processor's own speed by using a FIFO as OBUF between the processor and the system controller.

Once input data is available in IBUF, the processor starts its data processing operation and continues its operation independently of the system until it has processed up to one row of data. If the input data is not available for some reason, the processor will suspend its operation and wait upon the input data. If the input buffer is not available during the data output operation, the processor will suspend its operation and wait until the output buffer becomes available. Thus, the processor can wait for valid data to arrive without global synchronization requirements by employing an elastic buffer memory and hand shaking signals.

System coefficients are stored in the coefficient registers C and D during the processor initialization step. An important architectural consideration in this research is that the system coefficients need to be scalable during the processor operation and automatic system gain control need to be achieved between frames of data. The overall scale factor for processing of a new frame may be determined by scale control logic of the system controller. Then, the proposed DSP will achieve wider dynamic range than most other fixed point processors. Another important architectural consideration will be that the system coefficients are allowed to be changed during the data processing operation so that it may implement an adaptive system.

A state variable represents information that must be stored for later use. For the 2-D DLSI system, there are two kinds of state variables : the horizontal state variables and the vertical state variables. The current output is computed as a linear combination of the most recent vertical state variable in a vertical direction, the most recent horizontal state variable in the horizontal direction and the current input. A horizontal state variable requires a column delay and a vertical state variable requires a row delay. In the row by row implementation scheme, a column delay means a storage of an entire row data. The horizontal state variables are stored in the horizontal state variable registers and are used for the computation upon arrival of the next data. In the proposed processor, the horizontal variables in parallel using the current input, the current output and the previous value of the horizontal state variables.

The vertical state variables require off-chip data communication for an efficient implementation of row delays. Thus, it is important to use minimum number of vertical state variables for given algorithm. The variables are stored in the vertical state variable queue, QBUF, until they used for computation on the next row of data. QBUF is a FIFO type elastic buffer and is used as a data communication buffer. The processor will continously write the vertical state variables into QBUF at its speed until QBUF becomes full and will continuously read the vertical state variables from QBUF at its own speed until QBUF becomes empty. By allowing QBUF to be able to hold all the vertical state variables of one row of data, the processor can operate at full speed. QBUF requires and independent read and write capability and may be implemented by a multiport RAM or a FIFO chip.

3.4. Sequence Control Block

The sequence control logic circuitry generates control signals so that proper data are continu-

ously available to each functional unit at every cycle. All the internal and external control signals are generated in this block. Since the processor is designed to have good processor to processor communication capability for an efficient asyn chronous operation, it has handshaking signals in all four directions. The processor has an internal non-overlapping four phase clock generator. Thus, four different operations can be performed in a single cycle. For wider applications, a micro programming approach can be used for actual im plementation of this block.

3.5. Throughput Consideration

Each arithmetic unit of the processor is designed to have a throughput of two multipli cation/addition operations in a single cycle. As an example, we will look at a second order 2-D IIR digital filter, which can be represented by equation (1) with L=2 as follows:

$$g(m,n)=\sum_{j=0}^{2}\sum_{k=0}^{2} a(j,k)f(m-j,n-k) - \sum_{j=0}^{2}\sum_{\substack{k=0 \\ (j,k)\neq(0,0)}}^{2} b(j,k)g(m-j,n-k). \quad (6)$$

Direct implementation of equation (6) will require 17 multiplications and 16 additions per input for the second order system. Direct implementation requires past values of the input and the output for the computation of the current output. This scheme results in a high data communication re quirement which is undesirable for VLSI im plementation.

The state space implementation of equation (6) will also require 17 multiplications and 16 additions per input for the second order system. However, only the current input is necessary along with the previous values of the state variables for the computations. This scheme results in a low and simple data communication require ment which is desirable for VLSI im plementation.

The proposed single chip processor having 9 arithmetic units will take a single cycle for the second order case to compute all the state varables and the output for each input. The pro cessor requires no extra time for data input or data output operations. By matching the input sampling time with the processor cycle time, a real time 2-D IIR digital filter can be implem ented.

Mostly commercially available DSP chips have special hardware for a single cycle multiplication and accumulation. They provide substantial speed-up as compared to standard microproces sors. However, there is some overhead associated with the single cycle multiplication instruction. In most cases, the data preparation operations prior to multiplication require more time than the multiplication itself.

The TMS DSP chip from Texas Instruments is one of the most widely used DSP chip. In pre vious research, a second order 2-D IIR low pass digital filter was implemented using the TMS 32010 instruction set. The state space implemen tation technique in order to reduce the data com munication requirements can be used. The main body of the program which implements equation (6) using the state space techniques used 214 instructions excluding initialization steps. Thus, it will take more than 200 times as many pro cessor cycles to implement a second order 2-D IIR digital filter using a TMS32010 DSP chip as compared to using the proposed single chip pro cessor. Because of the specific computational structure, an advanced DSP chip, such as, TMS320C25 will not much reduce the number of insrructions for given example. The proposed DSP would be at least 50 times faster than the TMS320C25.

3.6. Consideration for Prototype Chip Design

Design, fabrication and test of a VLSI chip is a quite complicated and expensive procedure. Ad vanced CAD tool are essential for the design of VLSI chips in a reasonable time. In this research, a datailed simulation will be performed on the

proposed architecture. A SUN 3/150C workstation will be used for design and simulation of the processor architecture. Publically available VLSI design tool and design data base will be extensively used for the design of prototype chip. Although the actual VLSI implementation is not a part of this proposed research, a feasibility will be continuously considered throughout the architecture development.

The arithmetic block and the data block need to be custom designed for fast operation and a reasonable chip area. The sequence control block will be designed using the standard cell library and automatic placement and routings. The chip will be designed to run with a 40 MHZ system lock. Since the processor will use a four phase clock internally, the overall throughput of the processor will be one output per 100 nsec. The chip will contain over a hundred thousand transistors. Either 1.25 ㎛ MCNC double metal CMOS technology or MOSIS schalable double metal CMOS technology or both may be used for the prototype chip design.

Ⅳ. System Configuration for 2-D Digital Signal Processing

Using the proposed sigle chip DSP, a real-time 2-D digital signal processing system can be implemented. Since the DSP has been designed to perform all the necessary computations in a single chip, the system configuration has great flexibility. Behavioral simulation and verification for these system configurations will be one of the major works in this research.

4.1. A Single Processor System for Second Order System

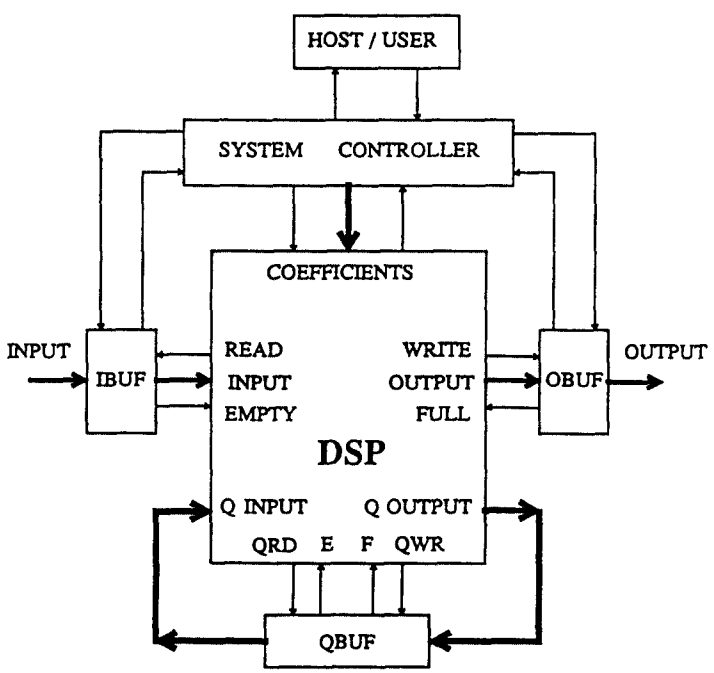

Fig. 4. A block diagram of a single processor system for 2-D digital signal processing

Fig. 4 gives a block diagram of a single processor system for 2-D digital signal processing. The system consists of a system input buffer, system output buffer, and a system controller. The system controller interacts with a host or user and controls the overall operation of the system.

When a host computer (or user) wants to process 2-D data, it (he) will give the system controller the command to start operation. Then, the system controller receives the necessary information about the process from the host or user. When the system controller becomes ready to start data processing operations, it will initiate the DSP by providing the system coefficients to the processor. Then it controls the system input buffer and the system output buffer for the data transmission to and from the DSP, one row of data as a block. The system input buffer and the system output buffer may be implemented by FIFOs to make the system control easier. QBUF holds the vertical state variables for an entire row of data and provides them for the processing of the next row of data. QBUF is controlled by the processor and may be considered as a part of the processor.

4.2. A Cascaded Configuration for Higher Order System

The procesor can be cascaded to implement higher order 2-D DLSI system. Each processor implements a separate transfer function and the processors are cacaded in a pipeline fashion to form the overall system. Fig. 5 is a linear array for the real time implementation of $2L^{th}$ order 2-D DLSI system with each processor implementing a second order section. In this array, each processor implements a real-time second order 2-D DLSI system and is cascaded with the next processor which also implements a real-time second order 2-D DLSI system with different system coefficients.

Once all the processors are initialized by the system controller, then the first processor starts operation when the input data are available and sends the result to the second processor. The second processor start operation as soon as the data are available and sends the result to the third

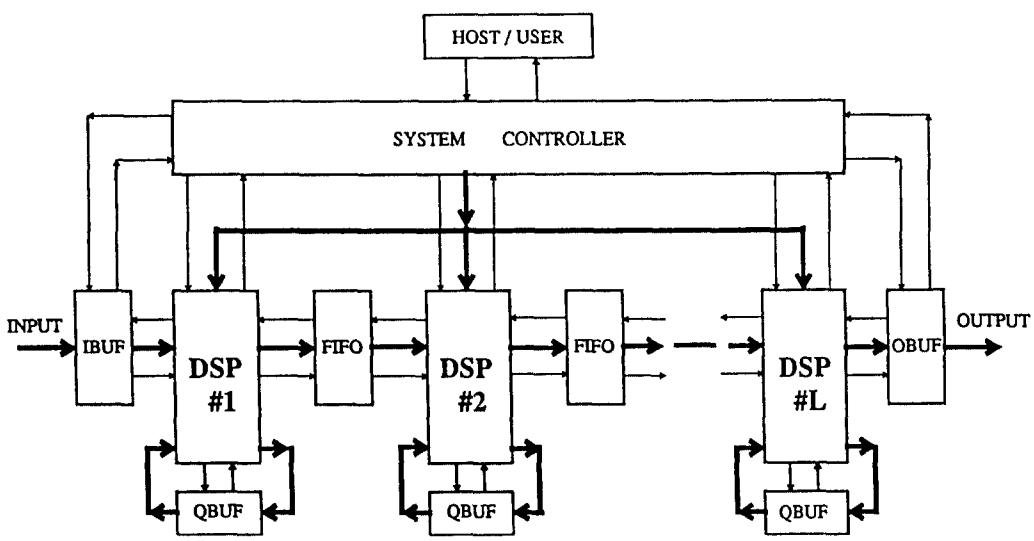

Fig. 5. A linear array for the real-time implementation of a $2L^{th}$ order 2-D DLSI system.

processor in the array. Data processing is continued in this pipeline fashion until the last processor (in this case L^{th} processor) finishes and sends the result to the system output buffer.

Each processor starts operation whenever data and vertical state variables are available and generates an output at every cycle. Since the data is processed sequentially, each processor must wait for the result from the previous processor. This wavefront protocol is handlend by each processor with the use of an elastic buffer memory and handshaking signals which are generated on chip for asynchronous operation.

An important advantage of using a cascaded configuration for the higher orde 2-D DLSI system is that the number of processors increases only in the order of L. For example, a $2L^{th}$ order filter implemented by cascading L second order sectons in a pipeline fashion, requires only L processors. Thus, a very high order 2-D DLSI system can be implemented in real-time with some initial latency deponding upon the number of cascaded stages.

V. Summary

In this research, a VLSI architecture for 2-D digital signal processing is presented. This architecture uses a specially designed high performance DSP as a processing unit in a real-time system. The DSP architecture is based upon a computational primitive developed by algorithm decomposition techniques. The overall system has a simple interconnection network and a simple control scheme. Each processor in this architecture operates asynchronously without global synchronization requirement thus providing very efficient operation.

The concepts developed in this research program can be applied to many other problems with intensive computational requirements. Starting from the algorithm analysis, then developing the architecture for the class of problems, the resulting architecture performs very well. Considering the difficulty and inefficiency in algorithm mapping procedures to existing multiprocessor system, this is an obvious and practical approach for most real-time signal processing problems. Furthermore, all the byproducts of this proposed research can be used to the development of advanced architecture for more complicated and computationally problem.

The class of adaptive systems is another important class for digital signal processing applications. In this class of systems, the system coefficients are altered in some way based upon contact with its environment. By allowing change of the system coefficients during data processing operations, the proposed architecture can be further extend to handle adaptive signal processing algorithms. Similar approach can be used to the development of algorithm decomposition procedures and corresponding VLSI architectures for class of adaptive systems. Adaptive systems find application in such areas as communications, radar, sonar, seismology, navigation and mechanical design.

Reference

1. J. Allen, "Computer architecture for digital signal processing," Proceedings of the IEEE, vol.73, no.5, pp.852-872, May, 1985.
2. S. Y. Kung, H. J. Whitehouse, and T. Kailath, Eds., VLSI and Modern Signal Processing, Englewood Cliffs, NJ : Prentice-Hall Inc., 1985.
3. C. L. Seitz, "Concurrent VLSI architectures," IEEE Trans. on Computers, vol. C-33, no.12, pp.1247-1265, December, 1984.
4. H. T. Kung, "Why systolic architectures?," Computer, vol.15, no.1, pp.37-46, Jan., 1982.
5. K. S. Lin, G. A. Frantz, and R. Simar Jr., "The TMS320 family of digital signal processors," Proceedings of the IEEE, vol.75, no.9, pp.1143-1159, September, 1987.
6. W. P. Hays, et al., "A 32-bit VLSI digital signal processor," IEEE Journal of Solid-State Circuits, vol.SC-20, no.5, pp.998-1004, October,

1985.

7. K. L. Kloker, "The architecture and applications of the Motorola DSP 56000 digital signal processor family," Proc. ICASSP, pp.523-526, 1987.
8. A. Aliphas and J. A. Feldman, "The versatility of digital signal processing chips," IEEE Spectrum, vol.24, no.40-45, June, 1987.
9. H. T. Kung and C. E. Leiserson, "Systolic arrays for VLSI," in Spare Matrix Proc. 1978, I. S. Duff and G. W. Stewart, Eds., Philadelphia, PA : Soc. Industrial Appl. Math., pp.256-282, 1979.
10. J. A. B. Fortes and B. W. Wah, "Systolic array-from concept to implementation," Computer, vol.29, no.7, pp.12-17, July, 1987.
11. S. Y. Kung, K. S. Arun, R. J. Gal-Ezer, and D. V. Bhaskar Rao, "Wavefront array processor : language, arichitecture, and applications," IEEE Trans. on Computers, vol. C-31, no.11, pp.1054-1066, November, 1982.
12. S. Y. Kung, "On supercomputing with systolic/wavefront array processors," Proceedings of the IEEE, vol.7, no.7, pp.867-884, July, 1984.
13. A. L. Fisher and H. T. Kung, "Synchronizing large VLSI processor arrays," IEEE Trans. on Computers, vol.C 34, no.8, pp.734-740, August, 1985.
14. J. H. Kim and W. E. Alexander, "A multiprocessor architecture for twodimensional digital filters," IEEE Trans. on Computer, vol. C-36, no.7, pp.876-884, July, 1987.
15. D. Dudgeon and R. Merseau, Multidimensional Digital Signal Processing, Englewood Cliffs, NJ : Prentice-Hall Inc., 1984.
16. S. M. Park, W. E. Alexander, and J. H. Kim, "A novel VLSI architecture for the real-time implementation of 2-D signal processing systems," Proceedings of the 1988 International Conference on Computer Design, October, 1988.
17. S. M. Park, W. E. Alexander, and J. H. Kim, "A multiprocessor system for realtime implementation of spatial domain digital filters," Proceedings of the 22nd Asilomar Conference on Signals, Systems and Computers, November, 1988.
18. Alan V. Oppenheim, Ronald W. Schafer, Discretetime Signal Processing, Englewood Cliffs, NJ : Prentice-Hall, 1989.
19. Kun-Shan Lin, Digital Signal Processing Applications with the TMS320 family, Englewood Cliffs, NJ : Prentice-Hall, 1987.

권 희 훈

- 1952년 3월 11일생
- 1976년 2월 : 동아대학교 공과대학 전자공학과 졸업(공학사)
- 1979년 2월 : 동아대학교 대학원 전자공학과 졸업(공학석사)
- 1982년 5월 : Southern Illinois University, School of Technical Careers, Dept. of Electrical and Electronic Technology 수료
- 1988년 8월 : 동아대학교 대학원 전자공학과 박사과정 수료(공학박사)
- 1973년 9월 30일~1979년 1월 30일 : 국방부 조병창(현 대우정밀) 기술연구관
- 1978년 8월 20일~1979년 1월 30일 : 부산 동의전문대학 전자과 전임강사
- 1991년 12월~1993년 1월 : Old Dominion University, Dept. of Electrical and Computer Engineering, Visiting Professor(Post Doc.)
- 1979년 3월 1일~현재 : 국립 충주산업대학 전자통신과 교수