

Hardware Realization of 2-D General Model State Space Systems

Prabhat Chandra Shrivastava¹, Prashant Kumar², Manish Tiwari³

Department of Electronics & Communication Engineering,
Motilal Neharu National Institute of Technology, Allahabad, Pin- 211004, India

¹prabhatphd@gmail.com, ²pkumar.mnnit@gmail.com, ³manishtiwari_me@yahoo.com

Abstract— In this paper, a hardware oriented realization, for the real time two dimensional (2-D) state space general model systems, is proposed. The proposed design of 2-D systolic array utilizes processing elements (PEs) which is simpler in use. The number of PEs is made equal to the multiplication of both the number of rows and column of the considered architecture. The proposed architecture provides excellent performance in terms of speed, efficiency and accuracy. Further, different designs of PEs are also proposed for state space feedback controller and unified structure. The unified structure can be work as state space systems as well as feedback controller after applying some control signals. It is shown that the area of unified structure is slightly more than that of structure-I and II taken separately. The ASIC synthesis results shows the proposed unified structure for system/controller of order 7 has taken approx 22% more area and 50% less area than state space system and controller with feedback respectively. Finally, the proposed architecture is implemented and analyzed using Verilog-HDL and Synopsis Design Compiler with 90nm TSMC target libraries.

Keyword - 2-D system, FM model, General model, state space filter realization, Synopsis Design Compiler, Verilog.

I. INTRODUCTION

The recent past have seen a continuously going research interests in the area of two-dimensional (2-D) discrete systems, due to their theoretical and practical viability in many application areas such as weather signal, seismographic data processing signal, mechanical vibration signal, remote sensing and telemetry signals, radio astronomy signal, biomedical system, navigation signals, gas absorption, water stream heating, radar and mobile signals etc. [1-3]. In a 2-D discrete system, information propagates in two independent directions, thereby making the system dynamics a function of two independent integer variables. The state-space methods have long outlived the other conventional methods to represent and analyze the practical systems. Motivated by this, many authors have attempted to describe the 2-D filter behavior in terms of linear state-space models [4-8]. In this paper, we consider the 2-D General state-space model [9] because many other popular state-space models can be represented as a special case of the General model.

Rapid developments in VLSI design and computing technology have attracted many researchers [10-13] to study and analyze the hardware architectures for 2-D state-space filtering. In [10], high speed VLSI architecture is proposed for the realization of 2-D recursive filters. The Local state-space realization of the generalized 2-D filter transfer function is obtained by mapping its signal flow graph to the 2-D state-space model. The 2-D recursive digital filter represented by a 2-D block state-space model is considered in [11] and a pipelined SIMD architecture is developed for the purpose of 2-D block processing. This type of architecture is particularly useful when similar operations on a large set of data are to be performed. Studies in [10] is further carried out in [12] for high speed architectures of 2-D state space recursive filter. An advanced state update architecture based on 2-D systolic array is proposed and it is shown that adjustable throughput rates for different requirements can be obtained due to flexible and modular global speedup architecture. Design of fine grain VLSI array processor for real time 2-D state-space systems is considered in [13]. The signal processing architecture in [13] consist of a linear systolic array which is obtained by exploiting the inherent concurrency in the 2-D state-space systems. At this point, it is worth mentioning that all the hardware architectures reported so far in the literature are based on the Roesser model due to its simplicity and similarity with the 1-D state-space model. However, designing a state-space architecture based on the 2-D General model, which is a super set of various other popular models e.g. Fornasini-Marchesini (FM) First 2-D state-space Model [6] and FM second 2-D state-space Model [8], is much more complex and tedious than the Roesser model. Here, the authors also feel to mentions that a processing architecture based upon the General model may be very useful since it can cover a large variety of signal processing and control applications. Motivated by this, we consider the problem of designing a processing architecture described by the 2-D General state-space model.

Notations: Throughout this paper, following notation are taken: R^n and $R^{n \times n}$ denotes real vector space of dimension $n \times 1$ and $n \times n$ respectively.

II. PRELIMINARIES

The basic preliminaries and definitions are defined in this section.

A. The 2-D General model Filter:

The state space representation of 2-D discrete linear shift invariant General model [9] can be defined by

$$x(i+1, j+1) = A_1 x(i, j+1) + A_2 x(i+1, j) + A_3 x(i, j) + B_1 u(i, j+1) + B_2 u(i+1, j) + B_3 u(i, j) \quad (1.1a)$$

$$y(i, j) = C x(i, j) + D u(i, j) \quad (1.1b)$$

$$i \geq 0, j \geq 0 \quad (1.1c)$$

where $x(i, j) \in R^{n \times 1}$ is state vector, $A_1, A_2, A_3 \in R^{n \times n}$ are the known system matrices, $B_1, B_2, B_3 \in R^{n \times m}$ are the known input matrices, $u(i, j) \in R^{m \times 1}$ is the input vector, $y(i, j)$ is a scalar output, $C \in R^{1 \times n}$, and $D \in R^{1 \times m}$. For simplicity, we have taken as $m = 1$ and $n = 2$ throughout the manuscript.

The initial conditions associated with the system are that there exist two positive integers and such that,

$$x(i, 0) = 0, i \geq r_1, x(0, j) = 0, j \geq r_2 \quad (1.2)$$

The equilibrium $x(i, j) = 0$ of above systems is said to be globally asymptotically stable if

$$\lim_{i \rightarrow \infty \text{ and/or } j \rightarrow \infty} x(i, j) = \lim_{i+j \rightarrow \infty} x(i, j) = \mathbf{0} \quad (1.3)$$

Transfer function for the General model is given as:

$$H(z_1, z_2) = C(z_1 z_2 I - z_2 A_1 - z_1 A_2 - A_3)^{-1} (z_2 B_1 - z_1 B_2 - B_3) + D \quad (1.4)$$

In (1.1), if $B_1 = B_2 = 0$ and $B_3 = B$, then (1.1) become the Fornasini-Marchesini (FM) First 2-D state-space model [6]. Similarly, When $A_3 = B_3 = 0$, then equation (1.1) become the FM second state-space model [8].

The block diagram of the 2-D General Model state space system is shown in figure 1. The block contains multiplier, adder and shifter circuits which will be discussed in next section.

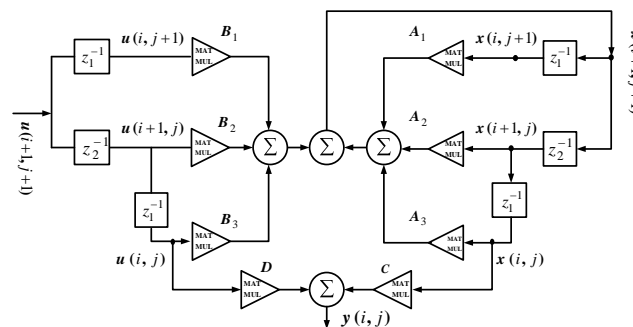


Fig.1: Block diagram 2-D General Model State Space System

B. Matrix Multiplication Unit (MTMU)

In 2-D state-space systems, requirement is to process massive amounts of data at high speeds along with accuracy. Matrix multiplication plays fundamental role in various engineering applications and scientific computations, like digital signal processing (DSP), digital image processing. Matrix multiplication units (MTMU) are more important part of proposed structures. The key component in the matrix multiplication is Multiplier-Accumulator or the MAC unit. The speed of operation and efficiency of the system depends on the performance and the number of MACs available in the design. The proposed architecture has required two types of MTMUs, 2×2 to 2×1 and 2×1 to 1×2 . Here, we consider two simple methods for 2×2 to 2×1 matrix multiplication units which are shown in Figure 2(a) and Figure 2(b).

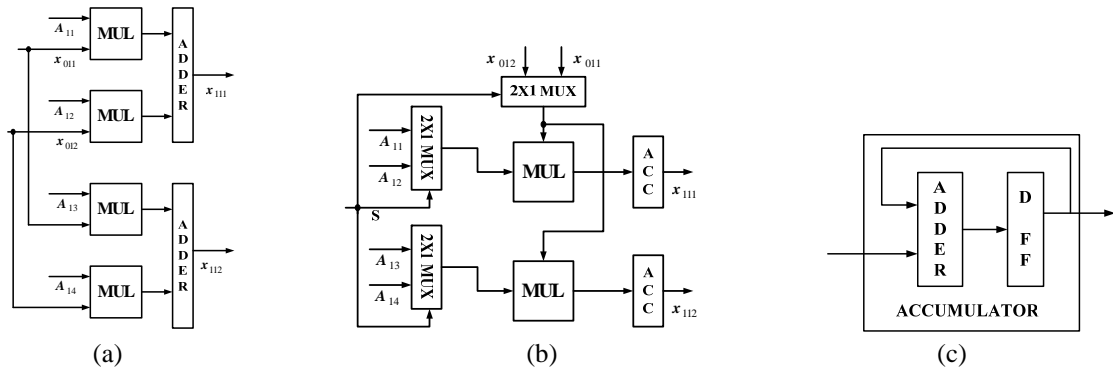


Fig 2: Matrix multiplication units (MTMU) (a) Method-I (4-multiplier) (b) Method-II (2-Multiplier) (c) Accumulator Unit (ACC).

Method-I in Fig 2(a), uses four multiplier and two adder circuits for operation. Multipliers are the most important part of any MTMU, because they take most of the area and time for the operation. Method-II, as shown in Fig. 2(b) utilizes lesser number of multiplier circuits for same operation at the cost of some small additional circuits which are 2X1 multiplexers and accumulator. Due to simplicity and speed, method-I is very suitable for proposed design. Finally, Fig. 2(c) shows a conventional MAC unit.

C. Processor Units (PEs)

In this section, we consider the architecture of processing element (PE) for the 2-D state-space system defined by (1.1). The basic function of a PE is to calculate the next state vector $x(i+1, j+1)$, for which it needs multiple inputs. A simplified block diagram of a basic PE with state inputs is shown in Figure 3(a). To calculate next state vector $x(i+1, j+1)$ in accordance with (1.1), a PE has to perform inner matrix multiplications and matrix addition operations. This is shown in Figure 3(b). Thus, design problem is twofold: first designing of efficient matrix multiplication units (MTMU) and second is matrix additions unit (MAU). The design of MMTUs is further complicated by the fact that a variety of matrix multiplication operation, due to the different size of matrices in (1.1), is required to calculate the next state. As mentioned in (1.1) we assume $m = 1$ and $n = 2$, then we require an MMTU for performing multiplication of matrices with size 2×2 to 2×1 and another MMTU to multiply the matrices of size 2×1 to 1×2 . In Figure 3(b), blocks identified as MAT MUL-1, MAT MUL-2 and MAT MUL-3 (in blue color) requires first type of MMTU while the blocks identified as MAT MUL-4, MAT MUL-5 and MAT MUL-6 (in green color) needs second type of MMTU. The output of both the MMTUs is added in an MAU to generate the next state vector $x(i+1, j+1)$, as shown in Figure 3(b). The values of state vectors $x(i, j+1)$, $x(i+1, j)$, $x(i, j)$ will be updated after each processing stage so that calculations at the next processing stage can be done correctly. Usually a systolic array technique, discussed in the following section of the manuscript, is employed for this purpose.

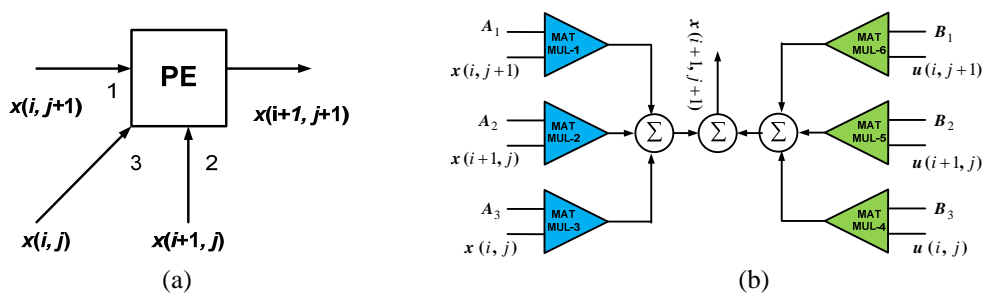


Fig. 3: (a) Block diagram of PE. (b) Internal architecture of processor elements (PE).

III. PROPOSED ARCHITECTURE

A simplified architecture for the 2-D state-space system described by the General model (1.1) is shown in Figure 4. The whole architecture is divided three different parts namely: external input-data distributor, the General model state-space system which is a systolic array of PEs (Figure 5) and a memory management unit (MMU) for storing next state vectors. Each of these parts is now discussed in detail in following subsections.

A. External Input-Data Distributer:

The external input-data distributor (EIDD) hosts arrays of 8-bit shift registers. The task of EIDD is to receive and store the known matrices $A_1, A_2, A_3, B_1, B_2, B_3, C$ and D in linear manner, then to receive and store the initial values of state vectors $x(i, j+1), x(i+1, j)$ and $x(i, j)$, and finally to receive and store the values of input vectors $u(i, j+1), u(i+1, j)$ and $u(i, j)$, such that $0 \leq i \leq 7$ and $0 \leq j \leq 7$. The values of input vectors will be required for state space filter only. The values of A and B is connected directly to all the processing stages through EIDD. The values of C and D are reserved to generate output $y(i, j)$ according to (1.1b), as and when required.

B. Systolic Array of PEs:

The second part of proposed architecture is linear systolic arrays of PEs. Systolic array provides an attractive solution to mapping signal processing algorithms onto very large scale integrated circuit (VLSI) hardware. The linear systolic array contains 7×7 PEs, each of which is combination of adder and multipliers as shown in Figure 3(b). The number of PEs used in this architecture is equal to the multiplication of number of rows to column in given systolic array. Due to its simple structure, localized communication between PEs, the systolic array of PEs maybe increase up to $N \times N$ where N represents number of row/column in systolic array. Furthermore, the communications between PEs and computations in PEs can be done at the same time without interrupting the data flow. Since the systolic array of PEs should be initialized to zero, the output of the PE should be connected to the corresponding entries directly. Neither delays nor other circuits are required to be connected between two corresponding PEs as shown in Figure 5. Only the computed value of next state vector $x(i+1, j+1)$ will move systolically from cell to cell while other inputs remains fixed for all PEs. Due to the recursive property of proposed architecture, the state value at $(i+1, j+1)$ cannot be processed until the instances $(i, j+1), (i+1, j)$ and (i, j) are available. E.g. $x(1,1)$ depends upon $x(0,1), x(1,0)$ and $x(0,0)$; $x(2,1)$ depends upon $x(1,1), x(2,0)$ and $x(1,0)$; $x(1,2)$ depends upon $x(0,2), x(1,1)$ and $x(0,1)$ etc.

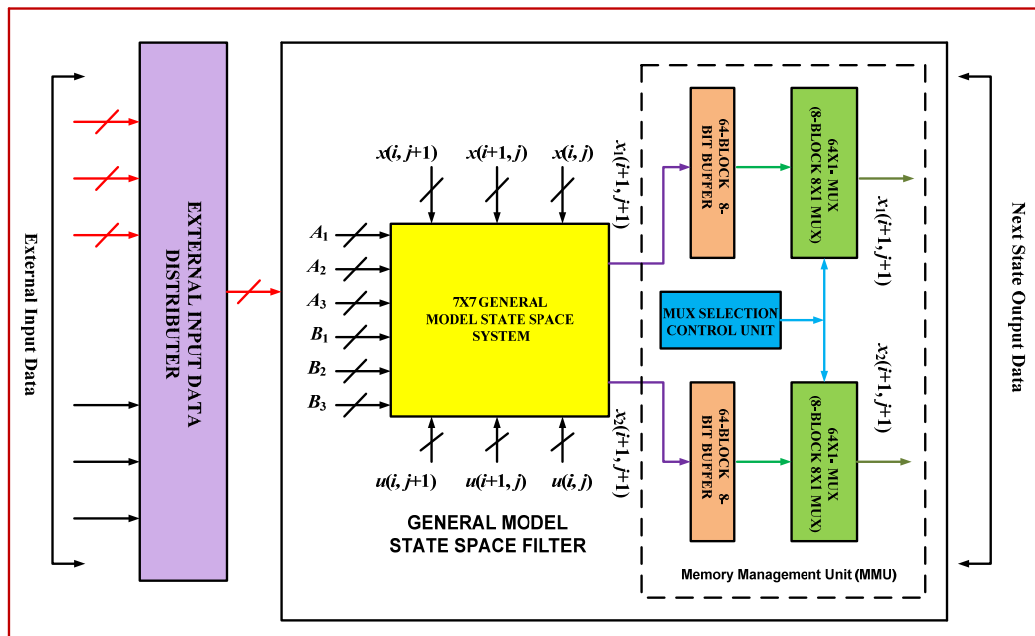


Fig. 4: Proposed architecture for the 2-D General model state-space system

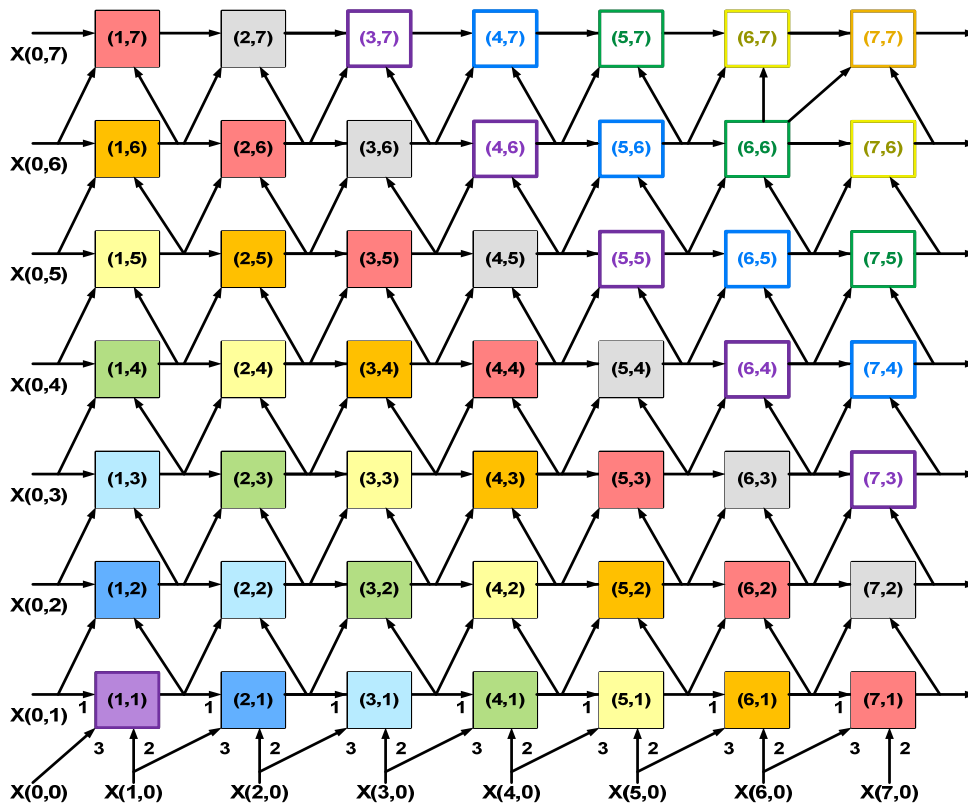


Fig. 5: Systolic arrays of processor units (7x7)

C. Memory Management Unit For State Vectors:

The last part of proposed architecture is a memory management unit (MMU) for the output of next state vectors. This unit contains two arrays of 64 8-bit registers. These registers receive the next state vectors from systolic array linearly and store them for future use such as generating the output or displaying the calculated values etc.

The proposed structure as shown in Figure (4) and Figure (5) may well be used for a variety of filtering, data processing and control application. To illustrate, we consider the case of 2-D discrete state-space systems with state feedback controllers which presented in the next section.

IV. A SPECIAL CASE: 2-D STATE-SPACE SYSTEM WITH FEEDBACK CONTROLLERS AGE STYLE

In this section, we consider the realization of feedback control gain to stabilize the 2-D state-space system realized through the architecture proposed in Section III. The state feedback control law to stabilize the state-space system (1.1) may be given as:

$$U = K X \tag{2.1a}$$

Where

$$U = \begin{bmatrix} u(i, j + 1) \\ u(i + 1, j) \\ u(i, j) \end{bmatrix} \tag{2.1b}$$

$$K = [k_1 \quad k_2 \quad k_3] \tag{2.1c}$$

and

$$X = \begin{bmatrix} x(i, j + 1) \\ x(i + 1, j) \\ x(i, j) \end{bmatrix} \tag{2.1d}$$

Thus, (2.1a) becomes

$$\begin{bmatrix} \mathbf{u}(i, j+1) \\ \mathbf{u}(i+1, j) \\ \mathbf{u}(i, j) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}(i, j+1) \\ \mathbf{x}(i+1, j) \\ \mathbf{x}(i, j) \end{bmatrix} \quad (2.1e)$$

Applying (2.1e) into (1.1a) we get,

$$\mathbf{x}(i+1, j+1) = \mathbf{A}_1 \mathbf{x}(i, j+1) + \mathbf{A}_2 \mathbf{x}(i+1, j) + \mathbf{A}_3 \mathbf{x}(i, j) + \mathbf{B}_1 \mathbf{k}_1 \mathbf{x}(i, j+1) + \mathbf{B}_2 \mathbf{k}_2 \mathbf{x}(i+1, j) + \mathbf{B}_3 \mathbf{k}_3 \mathbf{x}(i, j) \quad (2.2a)$$

and finally we get

$$\mathbf{x}(i+1, j+1) = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{k}_1) \mathbf{x}(i, j+1) + (\mathbf{A}_2 + \mathbf{B}_2 \mathbf{k}_2) \mathbf{x}(i+1, j) + (\mathbf{A}_3 + \mathbf{B}_3 \mathbf{k}_3) \mathbf{x}(i, j) \quad (2.2b)$$

Equation (2.2b) provides a simple way of realizing the controller gains to stabilize the 2-D state-space system represented by the architecture proposed by Figure (4) and Figure (5). Processing element (PE) for a 2-D state-space system with controller gain may be represented by following architecture.

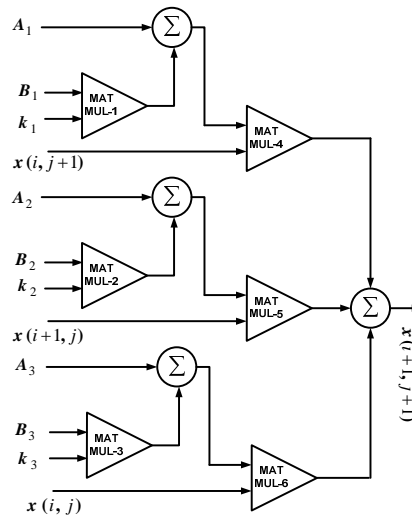


Fig. 6: Processing elements (PE) 2-D system with feedback controller.

It is worth noting that the PE shown in Figure (6) is a simple extension of the PE shown in Figure (3b), where input to various MMTUs in a PE is redefined. Hence, the architecture proposed in Figure (4) and Figure (5) remains valid.

A simple inspection of the PE proposed in Figure (6) reveals that the dedicated architecture which can be used only for the control applications. Whereas, it is desirable that depending upon the situations the hardware architecture may be used for control applications or any other data processing applications.

V. UNIFIED STRUCTURE FOR DATA PROCESSING AND CONTROL:

We have mentioned in the previous section that the proposed structure with PEs in Fig. (6) is suitable only for control application. Thus, in this section we consider a unified structure which is suitable for a variety of data processing and control applications. A comparison between the architecture of PEs of Figure (3b) and Figure (6) shows that difference lies in the inputs given to various MMTUs. In order to solve the issue in a simplified manner we introduce a multiplexer which selects input that is feeded to MAT MUL-4, MAT MUL-5 and MAT MUL-6 blocks of Figure (3b). This is shown in Figure (7). A control signal namely, controller/filter, is provided in the structure for selecting the input data either for control applications or for signal processing applications. When this control signal is LOW the structure effectively works as the one proposed in Figure (6) and when this control signal is HIGH the structure works as the one proposed in Figure (3b). Even though, efficient realization for the structure proposed in Figure (7) may be obtained, the authors have chosen the above structure to maintain simplicity in realization. However, readers are encouraged to explore various other realizations which are simple as well as efficient.

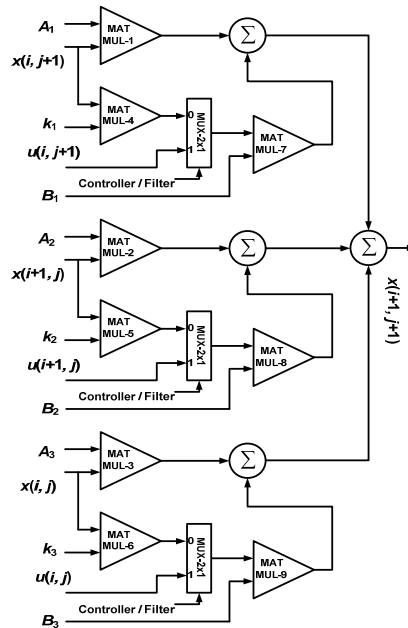


Fig 7: Unified Structure of processing elements for 2-D state-space system.

VI. PERFORMANCE COMPARISON AND VERIFICATION OF PROPOSED STRUCTURES:

In this section we have calculated theoretically the hardware and time complexity count of different types of digital circuits which is used in designing of state-space filters. The theoretical hardware analyses of general model state-space system (GMSSS), Controller and GMSSS along with feedback controller are shown in Table-I.

A. Data flow and timing analysis:

The timing operations of proposed design have divided in to three parts such as external input-data distributor, systolic array of PEs and MMU for state vectors. Both EIDD units and MMU for next state vectors are fixed for all three structures. The most important part of proposed design is linear systolic array. The propagation delay of this part depends on time taken by all processing elements together. Here, we have used 49 PEs. The outputs of PE₁₁ in first diagonal (violet color) of systolic array will be broadcasted to PEs of second diagonal (indium color) and these will be further broadcasted to third diagonal (green color). The process will be continued up-to last diagonal. All processing will be done in recursive. The time taken by particular diagonal is fixed and equal to propagation delay of PE. The number of diagonal in this architecture is thirteen as shown in figure 3(b). Let T_{PE} and T_{SA} be the propagation delays of one PE and systolic arrays respectively, then total processing time taken by systolic arrays is thirteen T_{PE} i.e. T_{SA} = (2N - 1) T_{PE}. Here, we have discussed two types of 2x1 to 1x2 matrix multiplication. Each method has particular performance for area and timing. The propagation delays of PE for method-I is given by T_{PE} = 2T_{8MUL}+72T_{2FA} while method-II gives T_{PE} = 3T_{8MUL}+88T_{2FA}+2T_{8MUX}+2T_{16DFF}, where T_{8MUL}, T_{2FA}, T_{8MUX} and T_{16DFF} are delays of 8-bits multipliers, 2-bits full adder, 8-bits 2:1 MUX and 16-bits D flip-flops respectively. The propagation delay of method-I are approximately 50% less than that of method-II at the cost of area. We have used many methods of matrix multiplication with specific properties and this is other area of research. Due to simplicity of method-I, we have taken method-I throughout paper. The time taken by proposed state-space filter is given by

$$T_{SSS} = T_{TEIDD} + T_{SA} + T_{TMMU} \tag{3.1}$$

Where T_{SSS}, T_{TEIDD}, T_{SA} and T_{TMMU} are time taken by proposed state space system, external input data distributor, linear systolic arrays and MMU for next state vectors respectively. The values of T_{SSS} may be varies according to which type of architecture we have taken.

$$T_{SSS} = T_{TEIDD} + 13(2T_{8MUL}+72T_{2FA}) + T_{MMTU} \tag{3.2}$$

Similarly, the times taken by Controller and GMSSS along with feedback controller are defined as T_{CONT} and T_{UNF} respectively and which is given below.

$$T_{CONT} = T_{TEIDD} + 13(2T_{8MUL}+88T_{2FA}) + T_{MMTU} \tag{3.3}$$

$$T_{UNF} = T_{TEIDD} + 13(2T_{8MUL}+72T_{2FA}) + T_{MMTU} \tag{3.4}$$

B. Area Complexity Analysis:

The area of proposed architecture is mainly due to area of systolic array and the associated EIDD and MMU for next state vectors. Therefore the area required for proposed architecture A_{SSS} can be expressed as:

$$A_{SSS} = A_{EIDD} + A_{SA} + A_{MMU} \tag{4.1a}$$

where A_{EIDD} , A_{SA} and A_{MMU} are the area required for a external input data distributor, linear systolic array and Memory Management Unit for next state vector respectively. From figure 3(a), A_{EIDD} , A_{SA} and A_{MMU} can be expressed as:

$$A_{EIDD} = 3 \times 31 A_{8DMUX} = 93 A_{8DMUX} \tag{4.1b}$$

$$A_{SA} = 49 A_{PE} = 49(18 A_{8MUL} + 272 A_{2FA}) \tag{4.1c}$$

$$A_{MMU} = 126 A_{8MUX} + 128 A_{8DFF} \tag{4.1d}$$

where A_{8DMUX} , A_{8MUL} , A_{8DFF} , A_{8MUX} and A_{2FA} are area of 8-bits demultiplexers, multiplier, D-flip flop, multiplexer and 2-bits full adder respectively. The area of DMUX and MUX are approximately same. Now equation (4.1a) can be written as:

$$A_{SSS} = 219 A_{8MUX} + 49(18 A_{8MUL} + 272 A_{2FA}) + 128 A_{8DFF} \tag{4.1e}$$

Similarly, the areas taken by Controller and GMSSS along with feedback controller are defined by A_{CONT} and A_{UNF} respectively and which is given below.

$$A_{CONT} = 219 A_{8MUX} + 49(30 A_{8MUL} + 448 A_{2FA}) + 128 A_{8DFF} \tag{4.2}$$

$$A_{UNF} = 219 A_{8MUX} + 49(24 A_{8MUL} + 348 A_{2FA} + 3 A_{8MUX}) + 128 A_{8DFF} \tag{4.3}$$

The details of theoretically timing and area calculation are given below:

TABLE-I

Type of Architecture	Timing (per unit)	Area (per unit)
State Space System	$T_{TEIDD} + 13(2T_{8MUL} + 72T_{2FA}) + T_{MMTU}$	$219A_{8MUX} + 49(18A_{8MUL} + 272A_{2FA}) + 128A_{8DFF}$
Controller Structure	$T_{TEIDD} + 13(2T_{8MUL} + 88T_{2FA}) + T_{MMTU}$	$219A_{8MUX} + 49(30A_{8MUL} + 448A_{2FA}) + 128A_{8DFF}$
Unified Structure	$T_{TEIDD} + 13(3T_{8MUL} + 72T_{2FA} + T_{MUX}) + T_{MMTU}$	$219A_{8MUX} + 49(24A_{8MUL} + 348A_{2FA} + 3A_{8MUX}) + 128A_{8DFF}$

C. Synthesis Result of Proposed Architecture:

We have coded the proposed design and processing elements in Verilog HDL. Further, designs are synthesized by Synopsis Design Compiler using 90-nm standard CMOS library. The word length of input samples and weights are taken to be 8 bits. The details of synthesis results of processing elements and proposed designs are depicted in TABLE-II and TABLE-III respectively.

From the TABLE-II, it can be observed that, the area of PEs for controller take more area, power and DAT (data arrival time) than that of PE of unified structure, while unified structure are working for both state space system as well as controller with feedback. The main reason behind this is that, the PE of controller used twelve 8X8 and twelve 8X16 multiplier respectively together and the PE of unified structure used twenty four 8X8 multiplier.

TABLE-III shows the synthesis report of proposed structures at 10MHz frequency. From the results as shown in TABLEs, the unified structure is more efficient as compare to other structures.

TABLE II (SYNTHESIS REPORT OF PROCESSING ELEMENT)

Type of Architecture	State Space System	Controller Structure	Unified Structure
Area (μm^2)	48439.28	92268.64	63692.63
Power (mw)	27.70	67.37	36.03
DAT (ns)	1.41	2.21	2.51

TABLE III (SYNTHESIS REPORT OF PROPOSED STRUCTURE)

Type of Architecture	State Space System	Controller Structure	Unified Structure
Area (μm^2)	2421950.37	4543339.67	3185925.66
Power (mw)	140.31	150.44	163.68

VII. CONCLUSION

In this paper the hardware realization problem of two dimensional linear general model state-spaces system and controller have been proposed. The realization of this architecture has been approached from a system theoretic point of view as well as hardware. Using this proposed scheme, we have derived a parallel architecture for the implementation of linear systolic arrays with diagonal scanning methods has included. By using a linear systolic array instead of single processor element, the time and throughput rate can be increased significantly. In this paper, we have presented two other architectures such as controller for 2-D system and unified structure which can be used either state space system or controller using some controlled signals.

The ASIC synthesis results shows the proposed unified structure for system/controller of order 7 has taken approx 22% more area and 50% less area than state space system and controller with feedback respectively. Finally, the proposed architecture is implemented and analysed using Verilog-HDL and Synopsis Design Compiler with 90nm TSMC target libraries.

REFERENCES

- [1] N. K. Bose, "Applied Multidimensional System Theory," Van Nostrand Reinhold, New York, 1982. T. Kaczorek, "Two-Dimensional Linear Systems," Springer- Verlag, Berlin, 1985.
- [2] W.-S. Lu and A. Antoniou, "Two-Dimensional Digital Filters," Marcel Dekker, Electrical Engineering and Elec-tronics, Vol. 80, New York, 1992.
- [3] R. N. Bracewell, "Two-Dimensional Imaging," Prentice- Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, 1995.
- [4] Roesser R. P., "A discrete state-space model for linear image processing," IEEE Trans. Automat. Control, vol. 20, pp. 1-10, 1975.
- [5] E. Fornasini and G. Marchesini, "State-space realization theory of two-dimensional filters", IEEE Transactions on Automatic Control, Year: 1976, Volume: 21, Issue: 4, Pages: 484 – 492, DOI: 10.1109/TAC. 1976.1101305.
- [6] Manish Tiwari and Amit Dhawan, "A survey on stability of 2-D discrete systems described by Fornasini-Marchesini first model", Power, Control and Embedded Systems (ICPES), 2010 International Conference on Year: 2010, Pages: 1 - 4, DOI: 10.1109/ ICPES. 2010.5698674.
- [7] E. Fornasini, G. Marchesini, "Doubly indexed dynamical systems: state-space models and structural properties," Math. Syst. Theory, vol. 12, pp. 59-72, 1978.
- [8] Manish Tiwari and Amit Dhawan, "A survey on stability of 2-D discrete systems described by Fornasini-Marchesini second model", Circuits and Systems, Vol. 3 No. 1, 2012, pp. 17-22. doi: 10.4236/cs. 2012.31003.
- [9] JERZY E. KUREK, "The General State-Space Model for a Two-Dimensional Linear Digital System", IEEE Transactions on Automatic Control, Year: 1985, Volume: 30, Issue: 6, Pages: 600 - 602, DOI: 10.1109/TAC. 1985.1103998.
- [10] J. Y. Zhang; W. Steenaart, " VLSI implementation of high speed two-dimensional state-space recursive filtering", Circuits and Systems, 1989., IEEE International Symposium on year 1989, Pages: 1099 - 1102 vol.2, DOI: 10.1109/ISCAS.1989.100544.
- [11] M. R. AZIMI-SADJADI; A. R. ROSTAMPOUR, "Parallel and Pipeline Architectures for 2-D Block Processing", IEEE Transactions on Circuits and Systems, Year: 1989, Volume: 36, Issue: 3, Pages: 443 - 448, DOI: 10.1109/31.17593.
- [12] J. Y. Zhang; W. Steenaart, High speed architectures for two-dimensional state-space recursive filtering, IEEE Transactions on Circuits and Systems, Year: 1990, Volume: 37, Issue: 6, Pages: 831- 836 DOI: 10.1109/ 31.55044.
- [13] Y. Iwata; M. Kawamata; T. Higuchi, "Design of fine grain VLSI array processor for real time 2D digital filtering", Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on Year: 1993,Pages: 1559 - 1562 vol. 3, DOI: 10.1109/ ISCAS. 1993 .394034.

AUTHOR PROFILE



Prabhat Chandra Shrivastava received the first class degrees of B.E. (Honours) in Electronics & Instrumentation Engineering from Rajiv Gandhi Technical University, Bhopal in 2008 and M. Tech in Electronics & Communication Engineering from Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India in 2010. Currently, he is a Senior Research Fellow in Electronics and Communication Engineering, MNNIT Allahabad, India. His research interests include the digital signal processing, Digital filter, State Space Filtering architecture design.



Prashant Kumar received the first class degrees of B. Tech in Electronics & Communication Engineering from SMVDU, Jammu & Kashmir, in 2012 and M. Tech in Electronics & Communication Engineering from Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India in 2015. Currently, he is a Junior Research Fellow in Electronics and Communication Engineering, MNNIT Allahabad, India. His research interests include the digital signal processing and VLSI architecture design.



Manish Tiwari received the first class degrees of B. E. in Electronics Engineering in 1996 and M. E. in Electronics and Controls from Birla Institute of Technology Pilani, India in 1999. Currently, he is an Assistant Professor in Department of Electronics and Communication Engineering, MNNIT Allahabad, India. His research interests include microprocessor based system design, embedded system, digital filter architecture design, multidimensional system design and signal processing.