International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

Implementation of Low Power and Memory Efficient 2D FIR Filter Architecture

Venkata Krishna Odugu, C Venkata Narasimhulu, K Satya Prasad

Abstract: ***A memory efficient design is analyzed to derive a low power-area-delay two dimensional (2D) Finite Impulse Response (FIR) filter architecture. The parallel processing concept is introduced in the fully direct-form 2D FIR filter. Due to this, memory reuse is carried out, and it reduces the overall storage memory of the FIR filter. The non-separable 2D FIR filter structure is designed and implemented with block size L and filter length N. The high speed and power efficient multipliers and optimized Carry Look Ahead (CLA) adders are used in the arithmetic module of the FIR filter and a pipelined adder unit is used for the final computation of the filter output. The switch level modification in the logic gates is proposed to reduce the area, power and delay of the adders. This proposed architecture is represented in HDL code and validation is carried out in CADENCE environment using NC Simulator and RTL Compiler synthesis tool. The area, power and delay reports are generated and compared with existing memory efficient 2D FIR filter hardware structures. The power is reduced to 44% and delay is reduced by 20% using Modified CLA (MCLA) adders and pipelining in the design.***

Index Terms: 2D-FIR, low power Multiplier, Parallel prefix adder, CLA, and memory reuse.

I. INTRODUCTION

In the two-dimensional signal processing such as image, video processing applications and for bio-medical signal processing [1], 2D digital filters are most frequently used. In the biometric systems, for feature extraction [2] and face recognition purpose [3] 2D filters are desired. The two-dimensional concept can be applied for both FIR and Infinite Impulse Response (IIR) fitters, but 2D FIR filters are more popular than IIR filters in terms of stability and simplicity of the design.

A. Related Work

To implement a memory efficient and less hardware complexity 2D FIR architecture, some investigations are carried out on existing structures. The symmetry 2D filters are discussed in [4]. The hardware metrics analysis and VLSI (Very Large Scale Integration) architectures for several symmetrical IIR and FIR filters are presented.Here, the un-symmetry frequency response is decomposed into sub components after that desired symmetry is obtained. This research paper provides four-fold symmetry IIR and FIR filters with less number of multipliers. In [5], the generalized formulas are defined to derive the new 2D VLSI filter architectures using sub filter blocks with local interconnection framework without any global broadcasting. In this work, FIR filter with quadrant symmetry and IIR filters with separable denominators are realized with the advantage of less number of multipliers. Many systolic architectures are implemented for 2D FIR filters to achieve optimization in an area, power, and delay. Few papers are considered to examine the concept of 2D Filters.In [6], the new systolic transformation technique and modified reordering schemes are accomplished to implement 2D systolic FIR and IIR filters. Due to the combinations of thesetwo techniques, lower quantization error, local broadcast, zero latency, and satisfactory critical paths are achieved. Another systolic transformation based on reordering of delay elements and summations, a new VLSI systolic array FIR and IIR filter structures [7] are realized. In this, a detailed logic gate level structure is presented with low latency, local broadcast with an accepted number of multipliers and delay elements. A bit level VLSI architectures for one dimensional and 2D filter are discussed in [8]. These structures are regular, modular and also compatible with other dedicated systems. In this work, hardware utilization and throughputs are improved with less latency. These structures are good enough foroptimization due to structure modularity and simplicity. These existing structures consist of many delay or storage elements in the data path to overcome the global signal broadcast. Memory complexity is a major issue in existing structures. The memory complexity affects the area occupancy and power consumption of the structure [9]. A memory-centric 2D FIR filter in non-separable and separable models are proposed in [10] with some penalty of power and delay. In this structure, through-put increased L - times when compared with previous works, but the hardware modules also increased to L times. The high number of hardware modules increases the area and power consumption. In this paper, a memory efficient 2D FIR filter with low power-area, and low delay architecture is proposed. In the proposed work, block based input processing is used for the reduction of memory storage and to achieve memory reuse in the fully direct-form 2D FIR filter. In the fully direct-form structure, the registers are placed in an input data path only, whereas in fully transpose-form the registers are placed in the intermediate signal level [10].

Revised Manuscript Received on 22 May 2019.
***** Correspondence Author
Venkata Krishna Odugu*, Electronics & Communication Engineering Department, Research Scholar JNTUK, Kakinada, India.
C. Venkata Narasimhulu, Electronics & Communication Engineering Department, Geethanjali College of Engineering and Technology, Hyderabad, India.
K. Satya Prasad, Rector, Vignan/s Foundation for Science, technology & Research, Guntur, India

The fully direct-form structure is converted into an optimized block-based architecture with memory reuse. The architecture consists of the arithmetic module and memory modules. The arithmetic module consists of a functional unit (FU) and adder block. In the FU, the important module is a multiplier which consumes more power and requires more hardware resources.

In this paper, the optimization is carried out in order to prune the power consumption, delay, and area of the multiplier. Always a compromise is required between the power consumption and speed of the multiplier. In the VLSI design, dynamic power is the major part of the total power consumption. The dynamic power is reduced in the multiplier by the reduction of the switching activities.

In this work, a new multiplier is used named as Bypass Zero Feed Multiplicand Directly (BZ-FMD) multiplier to reduce the power and to improve the speed of the structure. This multiplier is modified version of Bypass Zero, Feed A Directly (BZ-FAD) multiplier [11]. The switching activities and hard ware blocks are reduced in this existing multiplier. Due to this multiplier, the total dynamic power of the proposed FIR filter is reduced.

The performance of the multiplier is further improved using a parallel prefix adder for the fast addition of partial products.This fast parallel prefix adder is based on the high-speed adder logic of CLA. A modification is taken place in the conventional CLA to optimize the addition process with respect to speed, area, and power. These Modified CLA (MCLA)are also used for the final addition of the filter.

The optimized MCLA adders are used instead of normal Ripple Carry Adders (RCA) in the FU and adder block to improve the speed and power reduction of the FIR filter. The pipelined addition process is carried out using MCLAs for the computation of the final addition of the FIR filter. The MCLA adders and pipelining concepts are reducing the power consumption and delay of the entire filter structure.

The Transmission Gate Logic (TGL) is proposed to implement the all logic gates required for the implementation of adders. TGL can reduce the number of transistors each gate and improves the speed of the circuit.

The designing of a non-separable 2D FIR filter to avoid the redundancy in computations of the filter is presented in section II. Section III describes the proposed architecture of block based 2D FIR non-separable filter and sub modules implementation. Synthesis results and conclusions are explained in section V and section VI respectively.

B. Background Work

The 2D FIR filters can be represented and implemented in two ways, such as separable and non-separable. The general equations (1) and (2) represents 2D FIR separable and non-separable filters respectively:

$$\boldsymbol{H(z_1,z_2)=\sum_{l=0}^{N-1}\sum_{k=0}^{N-1}h(l,k)z_1^{-l}z_2^{-k}}\ (\mathbf{1})$$

$$H(z_1,z_2)=\sum_{l=0}^{N-1}\sum_{j=0}^{N-1}h_1(i).h_2(j).z_1^{-i}z_2^{-j} \qquad (2)$$

Where, $h(l,k)$ is the coefficient matrix of the non- separable FIR filter and $h(i),h(j)$ are one-dimensional impulse responses of a separable FIR filter.

The basic separable and non-separable 2D FIR filter block diagrams are represented in Fig.1. Memory complexity and hardware logic blocks such as adders and multipliers required for the conventional design of the non- separable filter is high. From (1) the non-separable 2D FIR filter with a size of (N x N) requires (N - 1) shift registers, each shift register size is M, $(N - 1)^2$ registers for the processing of a column of the input image (M x M). The hardware blocks, N^2-1 adders, and N^2 multipliers are required. In this paper, the conventional structure is modified to reduce the complexity.

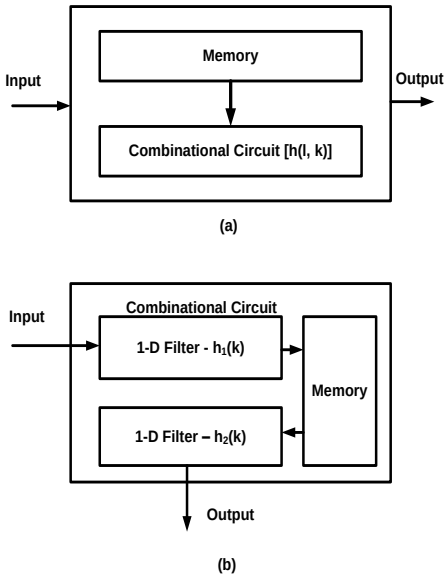


Fig. 1. Conventional (a) separable and (b) non-separable 2D FIR filters.

II. DESIGN OF NON-SEPARABLE 2D FIR FILTER

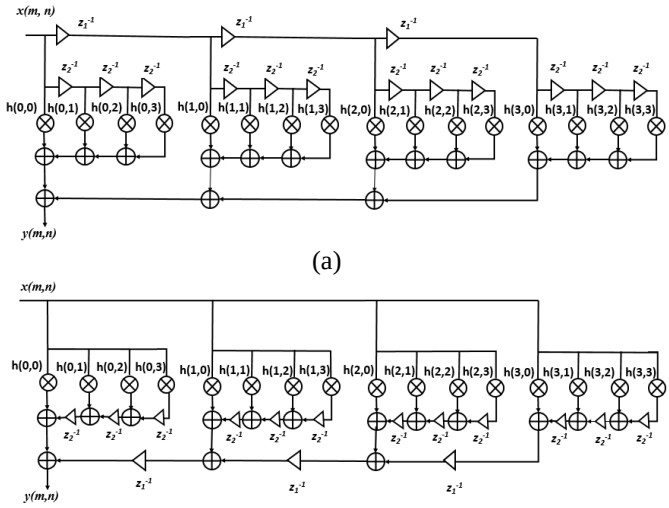
The basic equation (1) of a non-separable FIR filter can be rewritten as follows:

$$H(z_1,z_2)=\sum_{l=0}^{N-1}z_1^{-l}H_l(z_2) \qquad (3)$$

$$H_l(z_2)=\sum_{k=0}^{N-1}h(l,k).z_2^{-k} \qquad (4)$$

The above equations (3) and (4) of 2D FIR non-separable filters can be implemented in a fully direct form structure or fully transpose form structures as shown in Fig. 2. The fully direct form of the 2D FIR filter requires less number of memory elements. The memory of a fully direct-form 2D FIR filter is independent of intermediate signal width bits. All delay elements are placed in the input path of architecture only. This is a useful feature to reduce memory [10]. The fully transform-structure memory depends on intermediate signal widths.

The intermediate signal width is more than the input signal width.The same number of delay elements and arithmetic components are required for these two structures. The word length of the input and intermediate signals are different, so the overall memory of the two structures is different in terms of bits. The Table I shows the estimated memory of two structures, with a filter length of N = 8, the input image size is 512 x 512, the input signal width b = 8, and an intermediate signal width d =16. The fully direct structure requires less memory as per Table I.



(a)

(b)

Fig. 2. (a) Fully Direct form structure and (b) Fully Transpose form structure.

Table.I Comparison of Memory requirement for fully-direct and fully transpose forms

Name of the structure	Shift Register-Words		Total Memory Bits	Memory Bits
	Input signal memory	Intermediate signal memory		M=512, *b =8, N=8, d =16*
Fully- direct form structure	(M+N) (N-1)	0	(M+N) (N-1)b	29120
Fully -transpose form structure	0	(M+N) (N-1)	(M+N) (N-1)d	58240

A. Memory Reuse

The fully direct-form structure is considered for the design of proposed 2D FIR filter architecture. The input data flow of the direct-form structure is analyzed to explore the reusing of memory in the FIR filter. For understanding the memory reuse concept, the redundancy of input samples for the filter length N = 4 as shown in Fig.3. If the mthrow output computation is considered, then the outputs are *{y(m,n), y(m,n+1), y(m,n+2), y(m,n+3)}*. For the 4 x 4 filter, 16 input samples are required with respect to 4 columns and 4 rows of 2D input as shown in Fig.3. The shift registers and Serial-In-Parallel-Out (SIPO) Shift Register Blocks (SRB) are used to give past samples of rows and columns respectively.

The data flow of a fully direct structure represents, the 28 samples out of 64 samples are differentand the remaining 36 samples are redundant. These redundant samples corresponding to the outputs *{y(m,n), y(m,n+1), y(m,n+2), y(m,n+3)}* are highlighted in the Fig.3. The parallel computation or using block-based structure the redundancy can be avoided in direct form structure. For the determination of the particular output, past sample values are required. These past samples can be retrieved from the SRBs in every clock cycle [10].

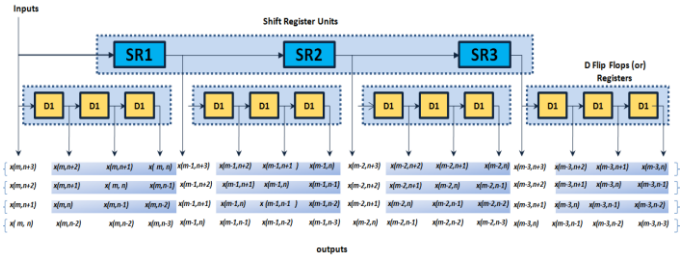


Fig.3. Data flow in the fully direct form structure for N=4 with four outputs *{ y(m,n), y(m,n+1), y(m,n+2), y(m,n+3)}*, [10].

B. Block formulation of the Non-separable 2D FIRFilter

In this section, the block-based non-separable filter design equations are derived. The block size L, input samples are processed and generate the L output samples in every clock cycle.The output equation of non-separable FIR filter for mthrow is given by,

$$y_{m,k} = \sum_{i=0}^{N-1} V_{i,k} \tag{5}$$

Where $y_{m,k}$ is the final output the filter represented by,

$$y_{m,k} = [y(m,kL) \quad y(m,kL-1) \quad . \quad . \quad y(m,kL-L+1)]^T \tag{6}$$

and $V_{i,k}$ is an intermediate vector defined as,

$$V_{i,k} = [v(i,kL) \quad v(i,kL-1) \quad . \quad . \quad v(i,kL-L+1)]^T \tag{7}$$

$V_{i,k}$ is the product of an impulse response matrix with input matrix A_k^{m-i} as given by,

$$V_{i,k} = A_k^{m-i}.h_i \,(8) \tag{8}$$

Where, A_k^{m-i} is part of theinput matrix is given by (9), from *(m-i)*throw of the image matrix of size 512 x 512.

$$A_k^{m-i} = \begin{bmatrix} x(m-i,k) & x(m-i,kL-1) & \ldots. & x(m-i,kL-N+1) \\ x(m-i,kL-1) & x(m-i,kL-2) & \ldots. & x(m-i,kL-N) \\ \vdots & \vdots & & \vdots \\ x(m-i,kL-L+1) & x(m-i,kL-L) & \ldots. & x(m-i,kL-N-L+2) \end{bmatrix} \tag{9}$$

The impulse response matrix is given by equation (10),

$$h_i = [h(i,0) \quad h(i,1) \quad . \quad . \quad . \quad h(i,N-1)]^T \tag{10}$$

The internal vectors of matrix $V_{i,k}$ are the inner product of impulse response h_i and S_k^{m-i} is the *l-th*row of A_k^{m-i} is given by equation (11),

$$v(i, kL-1) = S_{k,l}^{m-i} \cdot h_i \quad (11)$$

III. IMPLEMENTATION OF PROPOSED 2D FIR FILTER ARCHITECTURE

The proposed block-based FIR filter is implemented in a systematic architecture to avoid the redundancy in the data flow of the filter. To avoid redundancy samples, memory reuse concept is carried out and it reduces the overall storage memory with respect to the input data path. The architecture is designed with a block of L = 4 and length of the filter N = 8.

The non-separable block-based 2D FIR filter architecture is presented in the Fig.4. The equations (5) and (8) are converted into a fully direct-form structure with L = 4 parallel inputs. This architecture is comprised of two main modules, such as memory module and the arithmetic module.

Memory module consists of an array of 28 shift registers of P =128 words and 8 input register units (IRU), where P = M/L = 512/4 = 128. From 28 shift registers, every 4 registers are grouped and named as SRB. For this architecture, 7-SRBs are required i.e. SRB1, SRB2, ….SRB7.

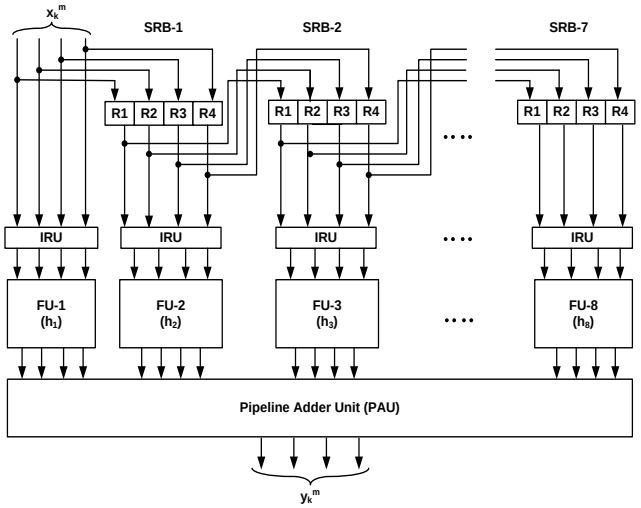


Fig. 4. Block-based non-separable 2D FIR filter architecture.

The block of 4 inputs are applied to the filter and it computes 4 outputs in every clock cycle. Like this, all the inputs from the input image matrix of 512 x 512 are applied block by block in serial order and generate corresponding outputs [15]. Due to the block based concept, the entire image can be completed in MP = 512 x 128 clock cycles instead of 512 x 512 cycles. The delay is reduced and throughput increased. In every clock cycle, N -1 = 7 input blocks are corresponding to N -1 = 7 consecutive input rows are obtained from the SRB unit.

The 7-past input blocks and current input block samples are applied to 8-IRUs. The internal register arrangement corresponding of redundancy avoiding logic is shown in Fig.5 for L = 4 and N = 8. It consists of (N -1) = 7 registers or D-Flip Flops to produce 8-point input vectors. The 8-point input vector is a combination of past and current input samples.

The 8 - IRUs generate the matrix of $[A_k]$ is shown in (9) of size 4 x 8 because L= 4 and N = 8; The first IRU receives the input samples from the current block and generates the 4 x 8 matrix, which is applied to the Functional Unit (FU). Similarly, 7 IRUs generates the 4 x 8 matrices and passed to the corresponding 7-FUs.

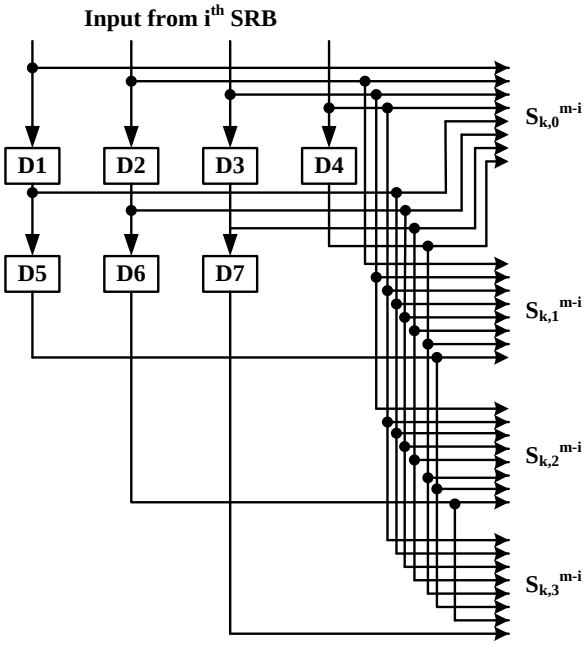


Fig. 5. The internal structure of IRU.

A. Arithmetic module

The first important block in the arithmetic module is FU. Here, N = 8 FUs are required to multiply the input vectors and the filter coefficients [h_i]. FU receives 4 input vectors from each of 8 IRUs. The $(i + 1)^{th}$ FU receives from $(i +1)^{th}$ IRU and computes the inner product of the input vector and $(i +1)^{th}$ row of impulse response matrix. The output matrix of FUs is called as [V_i].

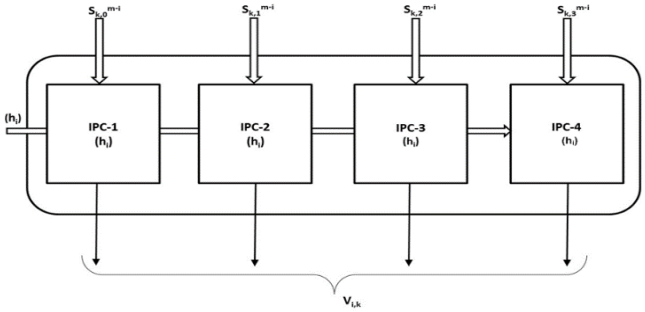


Fig. 6. The Internal block diagram of FU.

The internal structure of FU is shown in Fig.6. FU comprises 4 inner product cells (IPC). IPC is used to multiply the input vectors and corresponding elements of the impulse response matrix. The internal combinational logic of IPC is shown in Fig.7. It produces an 8-point inner product and adds 8 partial products together using an adder circuit.

B. Multiplier Implementation

Each IPC requires N multipliers and N-1 adders to produce the product of input samples and coefficients of the filter. The optimized multiplier and adder circuits are described in this section.

The basic multiplication operation is consists of two steps, partial product generation and addition of partial products. The dynamic power consumption of multiplier or any circuit depends on the switching activities. Equation (12) represents the general dynamic power in the VLSI circuits.

$$P_{Dynamic} = \alpha . C_L . f_{clk} V_{DD}^2 \quad (12)$$

Where C_L is Load capacitance, f_{clk} is clock frequency, V_{DD} is the power supply and α represents switching activity. The number of switching activities represents the power consumption of the circuit.

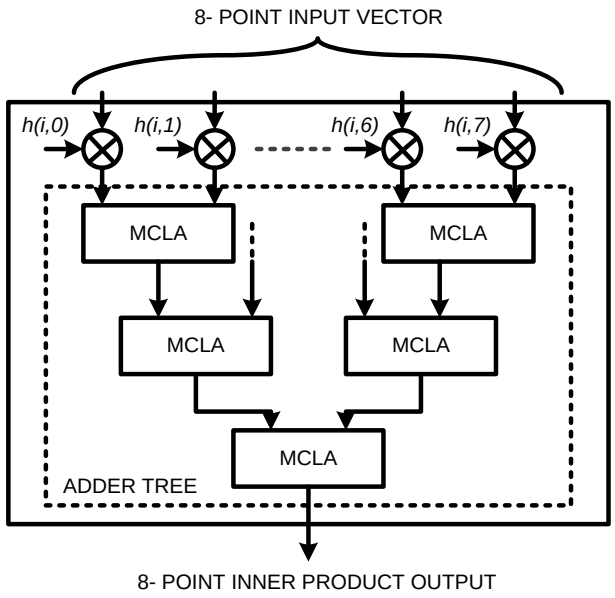


Fig. 7. Internal combinational logic diagram of IPC.

The conventional shift-add multiplier is modified and a new multiplier is developed based on Bypass Zero, Feed A Directly (BZ-FAD) is proposed in [11]. The BZ-FAD multiplier switching activities depends on (i) shifting the bits of multiplier (ii) switching of partial product bits (iii) adder switching activities and (iv) multiplexer switching activities for final addition. For further reduction in delay and to reduce the number of hardware blocks, the BZ-FAD is modified as Bypass Zero, Feed Multiplicand Directly (BZ-FMD) [12].

BZ-FMD multiplier architecture is shown in Fig. 8. It consists of an adder, multiplexer, product register, feed-register, and a controller.In order to reduce delay and area, some components are removed in the BZ-FAD multiplier. Ring counter is replaced with binary counter to reduce the complexity and are placed in the control block.

Initially, the controller checks the multiplier 0^{th} position bit to '0' or '1'. The controller consists of a synchronous binary counter instead of the ring counter, which increments for every clock cycle and selects each bit and checks. The adder, multiplexer, and feed register blocks are completely controlled by the controlling block. The control block controls the feed register data which feeds to adder block or multiplexer.

The multiplier bit decides the output of the multiplexer either it should be feeder register value of a previous partial product or adder output. The adder processing is skipped for the multiplier bit as '0' and feeder register feeds the previous partial product value directly to the MUX. Otherwise, the sum of multiplicand value and the previous partial product is directly given to MUX. The switching activity required for the zero bit addition is eliminated and directly the multiplicand fed to MUX.

The switching activities regarding shifting of the partial product is also reduced in this multiplier structure. In this multiplier, the higher half partial product bits only shifted right after processing and lower half partial product bits are remained and stored directly in the product register. Conventionally, the entire partial product is shifted right, whereas in proposed multiplier only half of the product bits are shifted. The switching activities for partial product shifting are reduced to 50% in the proposed multiplier.

In the designed multiplier, the switching activities corresponding to shifting of multiplier bits, addition and shifting of partial products are reduced. The reduction in switching activities decreases the major part of the dynamic power consumption. An 8-bit multiplier is implemented in this paper for the multiplication of filter coefficients and input samples.

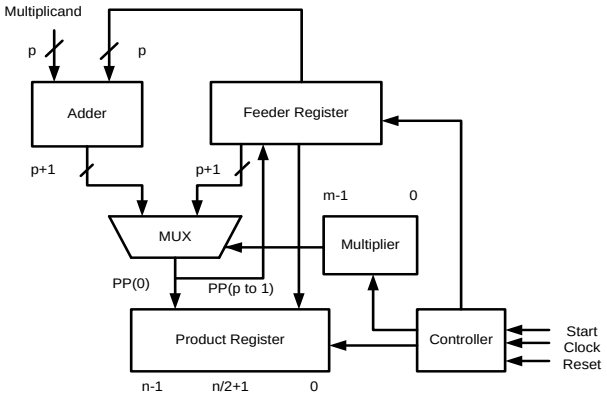


Fig.8. Architecture of multiplier

C. Parallel Prefix adder based on Modified CLA (MCLA)

The multiplication process is a combination of shifting and summation. Many adders are required for higher bit multiplication. Hence, the performance of the multiplier depends on adder also. For the N-tap filter implementation, N number of adders required for the summing of individual tap outputs and to get final filter output. In order to reduce power consumption and delay, the adder optimization is needed. In this section, a high-speed Carry Look Ahead (CLA) adder with necessary modification is described. The CLA equation is modified for the saving of the hardware andfurther to improve the performance and power saving. In this paper, the 8-bit and 16-bit parallel prefix MCLA adders are presented. The 8-bit MCLAs are used for the addition of partial products in the multiplier and 16-bit MCLA adders are used in adder tree.

In this Modified CLA (MCLA), a modified carry M_i is determined in the place of conventional carry c_i. After the calculation of propagating and generate terms, a parallel prefix addition concept is used to reduce the time [13].

The modified carry M_i and sum$S_i$of the MCLA is given by equations (13) and (14) respectively.

$$M_i = g_i + g_{i-1} + p_{i-1}.g_{i-2} + p_{i-1}.p_{i-2}.g_{i-3} + \cdots + p_{i-1}.p_{i-2} \ldots . p_1.g_0 \quad (13)$$

Where g_i and p_i are generating and propagate terms in equation (13). This equation is modified to improve the efficiency of the adder. The real carry is given by equation

$$c_i = M_i.p_i \quad (14)$$

The modified carry for even and odd bit positions are different. The M_i for even and M_{i+1} for odd given by equations (15) & (16),

$$M_i = (G_i^*, P_{i-1}^*) \odot (G_{i-2}^*, P_{i-3}^*) \odot \ldots \ldots \ldots \odot (G_0^*, P_{-1}^*) \quad (15)$$

$$M_{i+1} = (G_{i+1}^*, P_i^*) \odot (G_{i-1}^*, P_{i-2}^*) \odot \ldots \ldots \ldots \odot (G_1^*, P_0^*) \quad (16)$$

After the computation of above-modified carries for even and odd bit positions, the real carry is determined. Using an equation (14).The sum is calculated using the equation (17).

$$S_i = d_i \oplus (p_{i-1}.M_{i-1}) \quad (17)$$

The above carry calculation methods are only used for the lower half of the bits. The upper half of the bits carry is parallel computed as follows. In this method, the modified carry determined using intermediate propagate termand intermediate generate term as given in the equation (18).

$$c_i = \left(G_{i:k} + P_{i-1:k-1}.G_{k-1:j+1}\right).p_i \quad (18)$$

If we consider 16-bit adder, then the carry of the 8th bit is defined as equation (19)

$$M_8 = (G_{8:7}, P_{7:6}) \odot (G_{6:3}, P_{5:2}) \odot (G_{2:-1}, P_{1:-2}) = \left(G_{8:7} + P_{7:6}.G_{6:-1}, P_{7:6}.P_{5:-2}\right) \quad (19)$$

The remaining upper half bits carries are determined using the equations from (20) to equation (27).

$$c_8 = (G_{9:8} + P_{7:6}.G_{6:-1}).p_8 \quad (20)$$

$$c_9 = (G_{9:8} + P_{8:7}.G_{7:0}).p_9 \quad (21)$$

$$c_{10} = (G1_{10:7} + P_{9:6}.G_{6:-1}).p_{10} \quad (22)$$

$$c_{11} = (G_{11:8} + P_{10:7}.G_{7:0}).p_{11} \quad (23)$$

$$c_{12} = (G_{12:7} + P_{11:6}.G_{6:-1}).p_{12} \quad (24)$$

$$c_{13} = (G_{13:8} + P_{12:7}.G_{7:0}).p_{13} \quad (25)$$

$$c_{14} = (G_{14:7} + P_{13:6}.G_{6:-1}).p_{14} \quad (26)$$

$$c_{15} = (G_{15:8} + P_{14:7}.G_{7:0}).p_{15} \quad (27)$$

The above modified CLA equations are considered [13] and an 8-bit and 16-bit parallel prefix adders are implemented and shown in the Fig.9 and Fig.10 respectively.

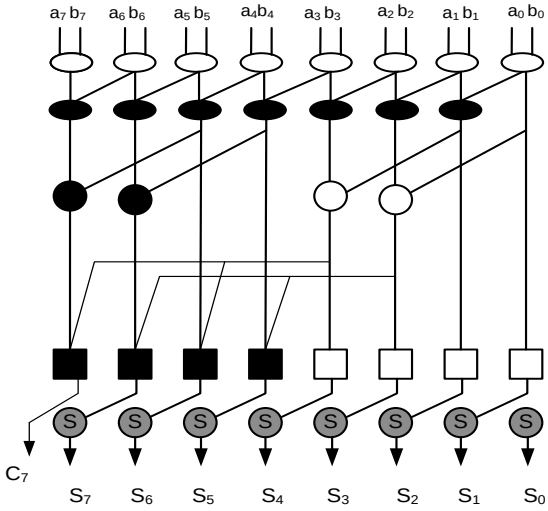


Fig.9. The 8-bit parallel prefix MCLA adder for multiplication

The logic cells required for the parallel prefix summation of 8-bit and 16-bit adder structures are shown in the Fig.11. All the logic cells are implemented by AND, OR and XOR logic gates.

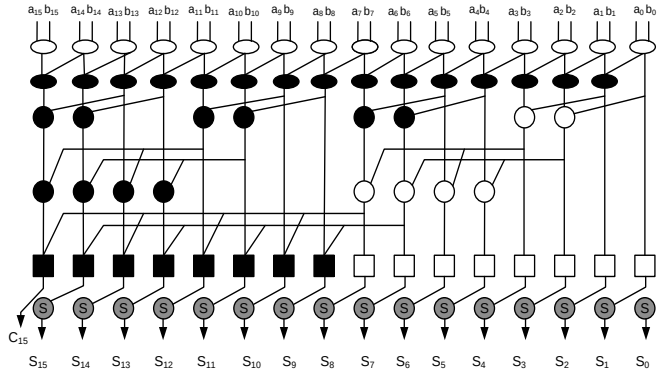


Fig.10. The 16-bit Parallel Prefix MCLA adder for filter outputs summation.

The implementation of each logic cell at transistor level is focused. The conventional CMOS logic is replaced with Transmission Gate Logic (TGL). The TGL logic reduces the number of transistors for the implementation of gates. The TGL AND/OR gates are shown in the Fig.12. The CMOS logic gates requires 4 transistors and whereas TGL gates can be implemented using 3 transistors only. Hence, total number of transistors for adder implementation is reduced thereby decreasing the switching activity and power dissipation. An 8-bit adder requires totally48 AND/OR gates. Each gate is constructed by 4 transistors by conventional CMOS logic then totally 192 transistors are required. The TGL AND/OR gates used total of 144 transistors only. The area saving is 25% in one 8-bit adder.

This optimization can increase the area saving for higher bit adders. If transistors count reduced then power consumption also reduced in each adder and multiplier. The overall 2D FIR filter area and power saving is improved by TGL gates.

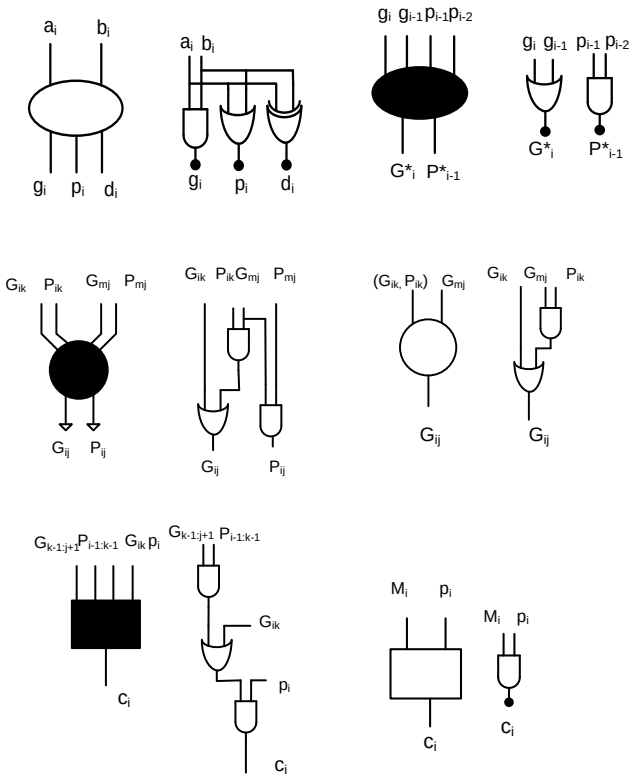
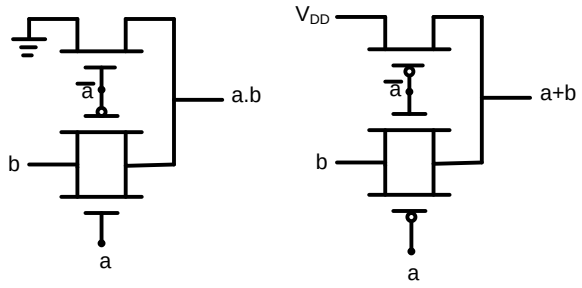


Fig.11. Internal logic cells are used in MCLA adder



(a) (b)

Fig. 12. (a) TGL-AND gate (b) TGL-OR gate.

The adder block consists of a special type of adder, i.e. MCLA. The MCLA adders are arranged in tree form is called asAdder Tree (AT) and used to compute the addition of IPC.

The final adder block of the FIR filter to produce an output of the filter which is the second block in the arithmetic module is designed as pipeline adder is called Pipeline Adder Unit (PAU).The PAU consists of D-FFs and MCLAs to do the final addition of the vectors from the FUs. The internal view ofthe PAU is presented in Fig. 13. The input data path is optimized by parallel processing and final output computation is pipelined. Pipelining and parallel processing are used to reduce the power consumption and reduce the critical path delay of the VLSI architectures [14, 15].

The output per one input block of 4 samples takes one clock cycle. Each clock cycle is defined using a minimum clock period, which depends on all arithmetic blocks delay. The clock period for this architecture is $T = T_M + T_{PAU} + T_{MCLA}(2\log_2 N - 1)$, where T_M is one multiplier computation time, T_{PAU} is the time required for the PAU and T_{MCLA} is a delay of the MCLA adder in the adder tree. The single row of the input image takes 128 clock cycles and the entire image matrix can be completed in 128M cycles, where M = image matrix size = 512.

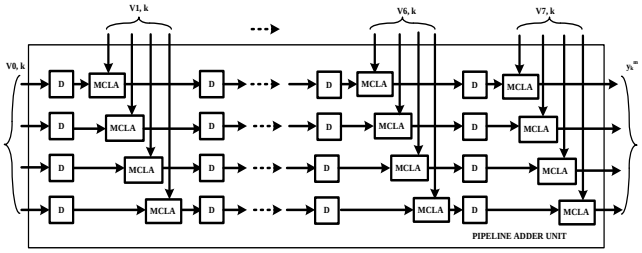


Fig.13. Pipeline Adder Unit (PAU) of 2D FIR structure

IV. SYNTHESIS RESULTS

The block based non-separable structure comprises of $LN^2 = 4 \times 64 = 256$ multipliers and $L(N^2 - 1) = 4 \times 63 = 252$MCLAs and $[(M + N)(N - 1)] = 512 + 8 \times 7 = 28672$ registers. This structure generates 4 outputs per each clock cycle. The memory reuse efficiency of non-separable structure is $L - 1 = 3$, and memory bandwidth per output (MBWPO) is $(L + N)(N - 1) / L = 21$. The MBWPO is 3 times less than the existing structures [10].

The proposed 2D FIR filter design is coded in HDL for block size L = 2 and 4 and filter sizes N = 4 and 8. The code is simulated in the 'NCSim' simulator from CADENCE tools. The simulated HDL code is synthesized using Encounter RTL compiler in the TSMC 90nm CMOS technology library from CADENCE tools. The generic building blocks library of the TSMC 90nm CMOS library is used for the synthesis of architecture and D-FFs are used as registers and shift registers. The input sample signal width is considered as $b = 8$ bits and the intermediate signal width is d =16 bits.

The proposed non-separable 2D FIR filter design with a block size of L = 2 and 4 and filter lengths N = 4 and 8 is compared with existing 2D FIR filter architectures in Table II. For the comparison purpose N=4 with block size L = 4 FIR filter is also implemented. Fig.14 represents the power comparison plot of the proposed design with existing 2D FIR filter architectures.

Table II. Comparison of the area and power parameters of different non-separable FIR filter architectures (L = 4)

Structure	Length of the filter (N)	Area (μm^2)	Power (mW)		
			Static	Dynamic	Total
Proposed Non-separable	4	34166	0.3424	3.0834	3.4258
	8	55186	0.5766	4.3634	4.9401
Khoo [5]	4	1009878	3.9107	4.9441	8.8548
Mohanty*et. al* [10]	4	791361	2.8016	3.2918	6.0934

The total power for filter length N=4 is reduced by 44% comparatively structure [10], and even the length of the filter is increased to 8, the total power saving is 20% compared [10]. The proposed design power reduction is 62% compared with the structure of [5]. The area saving is more than the existing architectures as shown in Table II. The Table III shows the comparison between the Non separable FIR filter power with block sizes L = 2 and 4 for different filter orders N = 4 and 8.

Table. III Comparison of power for block size L=2 and 4

Structure	Length of the filter (N)	Input Block Size	Area (μm^2)	Power (mW)		
				Static	Dynamic	Total
Proposed Non-separable FIR filter	4	2	20182	0.2032	1.6233	1.8265
		4	34166	0.3424	3.0834	3.4258
	8	2	28602	0.2342	2.4869	2.7211
		4	55186	0.5766	4.3634	4.9401

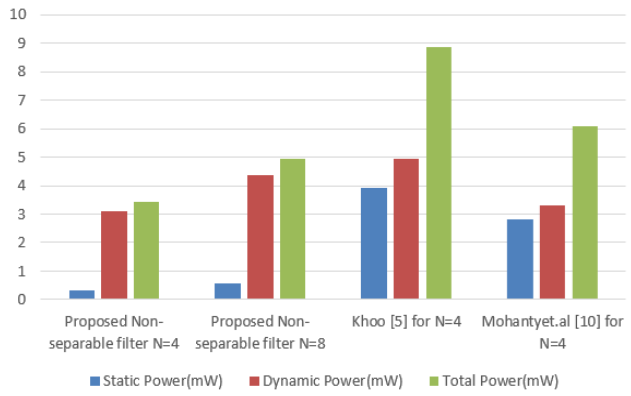


Fig.14. Graphical Comparison of power between proposed and existing structures.

The proposed design is investigated with MCLA adders used in multiplier and in the adder block and the same design with normal CLA adders in RTL compiler synthesis tool. The estimated synthesis results are shown in Table IV. The total power is reduced by 52% with MCLA adders with the area penalty of 2% only. The graphical comparison of power, area, and delay for the proposed design with MCLA and CLA is represented in Fig.15 and Fig.16 respectively.

Table.IV. Comparison of the area, power and delay parameters with MCLA and with CLA in 2D FIR filter.

Design	Length of the filter (N)	Power (mW)			Area (μm^2)	No. of Cells	Delay (ps)
		Static	Dynamic	Total Power			
Proposed filter (with MCLA)	4	0.3424	3.0834	3.4258	34166	5770	783
	8	0.576	4.3634	4.9401	55186	10793	887
Proposed filter (with CLA)	4	0.4869	5.7487	6.2356	38546	6206	956
	8	0.399	9.8979	10.297	58574	12986	1109

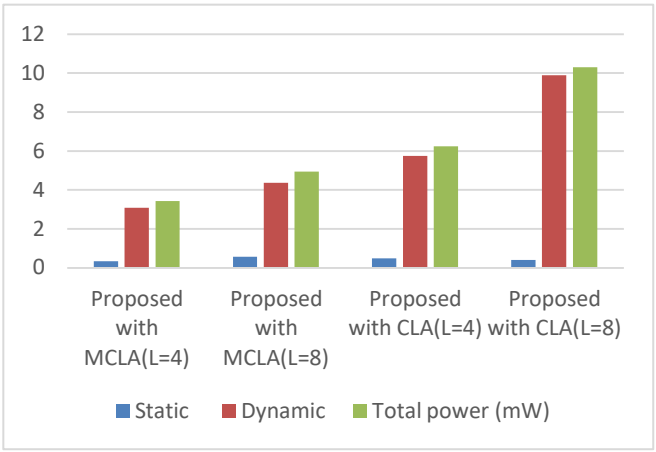


Fig.15. Comparison graph of power for the proposed design with MCLA and with CLA

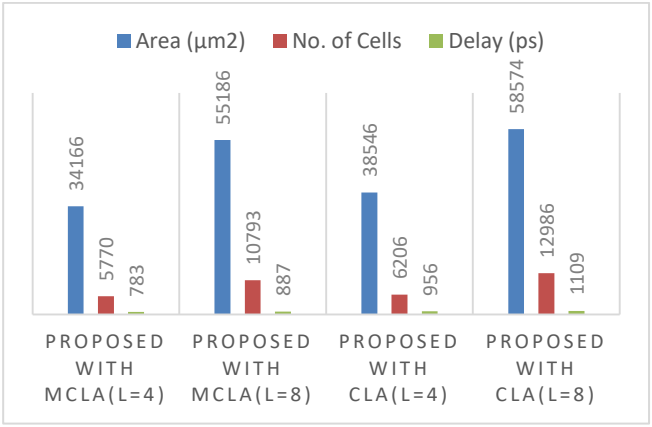


Fig.16. Comparison graph of area and delay for the proposed design with MCLA and with CLA for N=8

V. CONCLUSION

A systematic design is implemented to achieve low power, area and delay memory efficient 2D FIR non-separable filter architecture. The concurrent evaluation of output is achieved using block-based concept, so that throughput is increased to L times. The storage memory is reduced using memory reuse in fully direct form structure. The proposed filter design is implemented using low power BZ-FMD multiplier with parallel prefix adders. The MCLA addition concept is used to implement parallel prefix adder. Adder cells are optimized in terms of number of transistors to reduce the area, power and delay using TGL gates. The parallel processing and pipelining techniques in the final addition of the filter are used to reduce power consumption and critical path of the entire filter. The design is validated in two ways, with MCLA with TGL gates and with CLA adders. The power and area results of the non-separable 2D FIR filter with MCLA are better than the existing FIR structures. The input image is considered as 512 X 512, the input block size is L = 2 and 4, and filter length is N = 4 and 8 for the implementation of the architectures, and design is elaborated and synthesized in RTL Compiler tools from TSMC 90nm CMOS library.

The 2D FIR Filter structure with proposed MCLA adders consumed 50% less power than the normal 2D FIR filter with 20% reduced delay. The experimental results show that the proposed architecture is better than the existing memory efficient architectures in terms of area, delay, and power.

REFERENCES

1. H. Mohammadzade, L. T. Bruton, "A simultaneous div-curl 2D Clifford Fourier Transform filter for enhancing vortices, sinks, and sources in sampled 2D vector field images," in Proc. IEEE International Symposium on Circuits and Systems, May. 2007, pp. 821- 824.
2. T. Barbu, "Gabor filter based face recognition technique," in *Proc. Rmanian Acad. Ser. A*, 2010, vol. 11, no. 3/2010, pp. 277–283.
3. S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1160–1167, Oct. 2002.
4. P. Y. Chen, l. D. Van, H. C. Reddy, and C. T. Lin, "A new VLSI 2-D four-fold-rotational-symmetry filter architecture design", in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2009, pp. 93-96.
5. I. H. Khoo, H. C. Reddy, L. D. Van, and C. T. Lin, "Generalized formulation of 2-D filter structures without global broadcast for VLSI implementation", in Proc., *IEEE MWSCAS, Seattle, WA, USA*, Aug. 2010, pp. 426-529.
6. L. D. Van, "A new 2-D systolic digital filters architecture without-global broadcast", IEEE Trans. Very Large Scale Integr. Syst., vol. 10, no. 4, pp. 477-486, Aug. 2002.
7. Van, Lan-Da, et al. "A new VLSI architecture without global broadcast for 2-D digital filters." 2000 *IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*. Vol. 1. IEEE, 2000.
8. Mohanty, B. K., and P. K. Meher. "High throughput and low-latency implementation of a bit-level systolic architecture for 1D and 2D digital filters." *IEE Proceedings-Computers and Digital Techniques* 146.2 (1999): 91-99.
9. B. K. Mohanty and P.K. Meher, "A high-performance FIR Filter Architecture for Fixed and Reconfigurable Applications", IEEE Trans. On VLSI Systems, vol. 24, issue 2, pp. 444-452, 2016.
10. B. K. Mohanty and P.K. Meher, and A. Amira, "Memory Footprint Reduction for Power-Efficient Realization of 2-D Finite Impulse Response Filters", in *IEEE Trans Circuits Syst. I*, vol. 61, no. 1, Jan. 2014.
11. M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, BZ-FAD: A low-power low area multiplier based on shift-and-add architecture, IEEE Trans. Very Large Scale Integration (VLSI) Systems. (2009) 302–306.
12. Pinto, Rohan, and Kumara Shama. "Low-Power Modified Shift-Add Multiplier Design Using Parallel Prefix Adder." *Journal of Circuits, Systems and Computers* 28.02 (2019): 1950019.
13. Poornima N and V S KanchanaBhaaskaran, Area efficient hybrid parallel prefix adders, J. Procedia Materials Science. 10 (2015) 371–380.
14. A. P. Vinod and E.M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers" IEEE Trans. Wireless Commun., vol. 7, no. 5, pp. 1669-1675, Jul. 2006.
15. O. Venkata Krishna, C. VenkataNarasimhulu and K. Satya Prasad "Design and Implementation of Block Based Transpose Form FIR Filter" in IJCA, Issue 8 Volume 1, Jan-Feb. 2018.

AUTHORS PROFILE

Mr. Venkata Krishna Odugu received B.Tech degree in Electronics and Communication Engineering from Acharya Nagarjuna University, 2004 and Master of Technology with specialization VLSI System Design from JNTU Hyderabad in 2009 and pursuing Ph.D. fromJNTU Kakinada in the area of VLSI Signal Processing. Interested areas are VLSI Design and VLSI Signal Processing etc.

Dr. C. Venkata Narasimhulu received B.Techdegree in Electronics and Communication Engineering from S V University, Tirupathi in 1995 and Master of Technology in Instrumentation & Control Systems from REC, Calicut in 2000 and Ph.D. from JNTU, Kakinada in 2013 in the area of signal Processing. He has published more than 25 technical papers in national and international Journals and Conferences. His interested areas are Signal Processing, Image processing and medical image processing etc.

Dr. K. Satya Prasad received B Tech. degree in Electronics and Communication Engineering from JNTU college of Engineering, Anantapur, in 1977 and M. E. degree in Communication Systems from Guindy college of Engg. , Madras University, in 1979 and Ph.D from Indian Institute of Technology, Madras in 1989. He has published more than 139 technical papers in different National & International conferences and Journals and Authored one Text book. His areas of Research include Communications Signal Processing, Image Processing, Speech Processing, Neural Networks & Ad-hoc wireless networks etc.