THERMAL AWARE PLACEMENT AND ROUTING USING MULTIPLY SUPPLY VOLTAGE OPTIMIZATION

JEYAN PRINCE CHARLES D VINSLEY S.S
Assistant Professor, Maria College of Engineering and Technology, Attoor, Thiruvattaru, Tamil Nadu,India, +919942863116, jeyanprincecharles@gmail.com
Professor, Lourdes mount college of engineering and technology, Mullangana Vilai,Tamilnadu,India

Abstract: *Power dissipation in the in an IC at a particular spot increase the thermal profile of the spot which leads to the failure of the IC. There are huge research which aims for the thermal reduction and accumulation. The work proposes and algorithm which effectively goes for thermal profile based placement and routing and the different blocks are mapped with different supply voltage so as to reduce the thermal profile which leads to the critical path variation and the conditions are verified to check the timing issues. This increases the white space and increase the routability of the architecture. This proposed algorithm is verified with the few bench mark circuits and the results shows that this leads to reduction n area and power and the same has been extended for the multiplier architecture which also report 60% reduction in area and 9% reduction in power. The implementation of the algorithm is done using matlab and the proposed VLSI architecture was implemented using FPGA and ASIC platform.*

Key words: *Thermal Profile, Floor plan, Multiple supply voltage, Critical path and FPGA.*

1. Introduction

As the scaling of the MOS devices increases the power density of chip increases dramatically. The microchip manufactured with 100nm produces a power dissipation of around 50W/cm^2 and due to scaling if the technology node reduced to 50nm size the power dissipation goes to 100 W/cm^2 [1]. The increase in temperature will have the consequence as follows i) Raises the $I_{leakage}$ of the devices and there by increases the leakage power consumption of the design ii) Decrease in the mobility of the charge carriers and thereby increase the delay of that design iii) It increase the resistance of the wire interconnect and thereby increase the delay. Research shows that for every 10C increase in temperature the driving capability decrease by 4% and the reliability decreases to a great extend [2]. Temperature reduction is one among the design constrain in addition to area, power speed optimization problem. The problem of uneven temperature profile of an IC is being addressed by many researchers. One technique to address this issue is by adjusting the floorplan based on the cost function optimization. The cost function should have the parameters like temperature (max and Min), wire-length, area etc. The traditional floor-planning method is unable to solve the problem efficiently due to power dissipation caused so the floorplan should be a thermal aware floorplan.

Our work proceeds as follows: Section II introduces the thermal analysis on integrated circuits. Section III discuss about the previous work. Section IV proposes the algorithm and its implementation. Section V compares the results of the proposed algorithm with the existing. Section VI discuss about the application of this algorithm in the floorplan for the multiplier architecture and the analysis of the results.

2. Thermal Analysis on Integrated circuits

Heat produces in an IC is due to the inrush current flow in the circuit and this continuous current flow creates a power loss in the form of heat/hotspots. This gets aggravated due to scaling of device which provides less area and more compaction. The accumulation of heat produced leads to the self-destruction of the circuit so it is highly essential that the heat produced gets removed by heat conduction process. The temperature of the IC is determined by the rate at which the heat is being removed, so the heat conduction in an IC is determined by the heat conduction equation (1).

$$-k\nabla^2 T = g \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where g represents power density in volume (W/m^3), k is the thermal conductivity (W/mK). And T is the temperature in kelvin of the IC. Let's discuss the boundary condition of an IC. The heat dissipation follows an adiabatic operation in the vertical sides and the same could be extended for side not attached to heat sink. The side of the chip attached to the heat sink can be considered as convective conduction. Equation 1 has been framed assuming a steady state temperature flow and thereby the thermal conductivity is not a function of temperature. Most of the heat simulators of an IC make these assumptions and approximations in floor-planning. The thermal heat flow that takes place between substrate and surrounding can be modelled by equation (2).

BUPT

$$g = k\frac{\Delta T}{\Delta z} \quad \text{..... (2)}$$

The distance of separation between substrate and heat sink is represented by Δz. ΔT shows the temperature difference between the ambient air and substrate, k represents the thermal conductivity between substrate and the ambient air and g is the power density (W/m^2) of the block under consideration. Equation 2 can be rewritten as:

$$T_{Substrte} = \frac{\Delta z}{k} g + T_{air} \quad \text{..... (3)}$$

The idea of hotspots surrounded by the cooler blocks will lead to the reduction of hotspot temperature faster and also more the inter block spacing (i.e., halo space) will also lead to the heat reduction easy and fast and depicted in the fig.1. Figure 1 illustrate that block temperatures can also be decreased by spreading high power density blocks along the edge of the chip which is congruent with the results of [3]

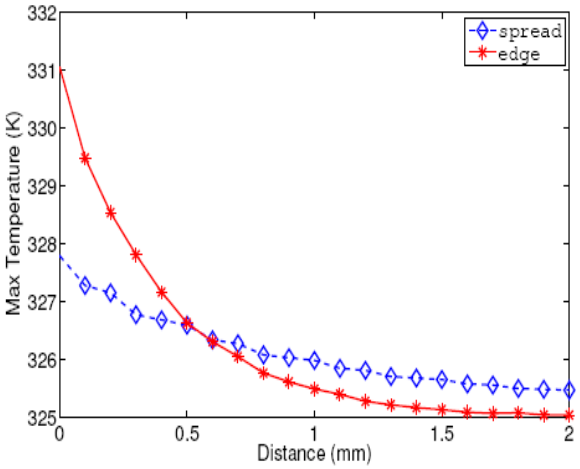


Fig.1 Illustration of block spreading of high heat profile blocks along the edges

3. Previous Work

There are a huge research work reported on the investigation of algorithms for both2-D [4]–[6] and 3-D [7]–[11] floorplans. These works tries to reduce the cost functions without decrease in the chip temperature. Sheldon et al. [12] came up with an algorithm to reduce the heat by white space optimization in the floorplanning. Lixia Qi et al. [13] used simulated annealing for thermal aware routing. Gracia Nirmala Rani et al. [14] used GA for thermal aware floor plan. Ehsan et al. [15] discusses the technique of enabling the power density based on the thermal profile. Chih-han Hsu et al. [16] made a reliability study with rectangle and double signal through silicon vias insertion. David Guilherme et al. [17] developed a tool which provides a technology independent layout generation which takes the thermal profile into consideration.

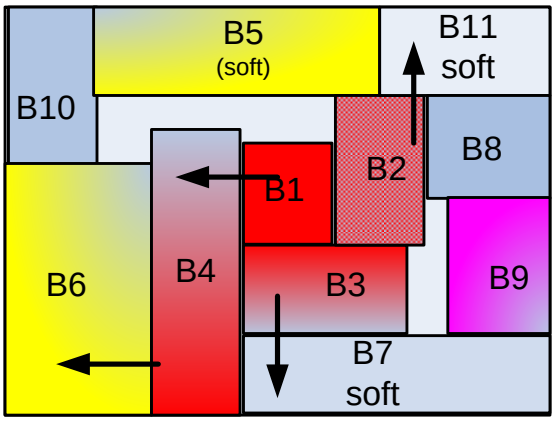
All reported works came up with the evolutionary technique of shuffling the design blocks so as to reduce the thermal profile of the design. This work aims for not only reducing the thermal hotspots but also resynthesize the design for reducing the power and scattering the thermal profile equally. The following are the objectives for the proposed work

1) Creating a multi-supply voltage islands for reduction of power dissipation
2) Allocation of high heat profile blocks at the edges surrounded by cool profile blocks
3) Dynamically allotting the white space around the hotspots to reduce the heat produced
4) Swapping, Rotation and altering the dimensions of the soft blocks which could lead to the reduction of wire length and hotspots in ICs

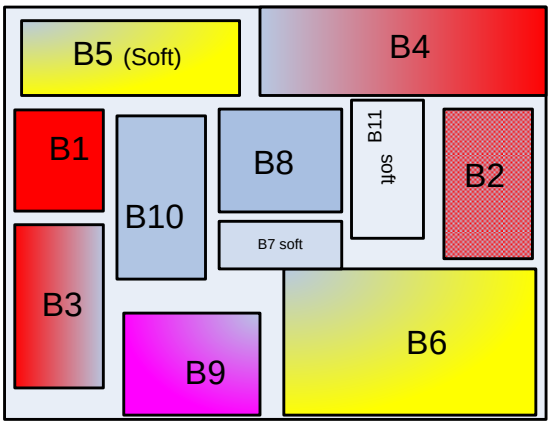
4. Proposed Algorithm

1)Identify the large power density blocks (Hottest block) (B1, B2….Bn) and rate them accordingly as (R1, R2…Rn). Where 'n' represents the number of blocks.
2)Arrange the high power density blocks (R1 to Rn) along the edges of the chip. And these arrest the thermal diffusion to lateral blocks.
3) Identify the high switching activity block and reduce the supply voltage of the block.
4)If (Timing of the power reduction blocks Bi) <Critical Timing. Then No issue proceed for the next block for voltage reduction.
5)If (Timing of the power reduction blocks Bi) >Critical Timing. Then increase the supply voltage of the block till the critical timing of the block is being met.
6) Now multiple blocks will work with multiple supply voltage which leads to the thermal heat reduction with reduction in speed.
7)Dynamically allocate the white space based on the thermal profile of the block. If more the thermal profile more will be the white space allocation surrounding it.
8) Identify the soft block that has the thermal profile >ε which is the threshold heat value of the block.
9)Now the block dimension Bi can be altered so as to fit with larger area so that the heat profile of the block Bi goes down.
10)Placing of white space around the block Bi increases the routability and reduces the connection in the routing.

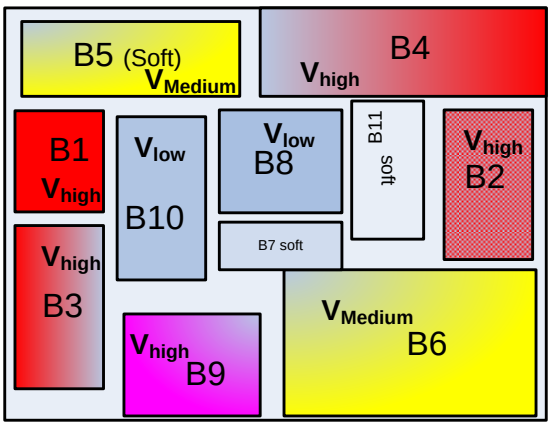
BUPT



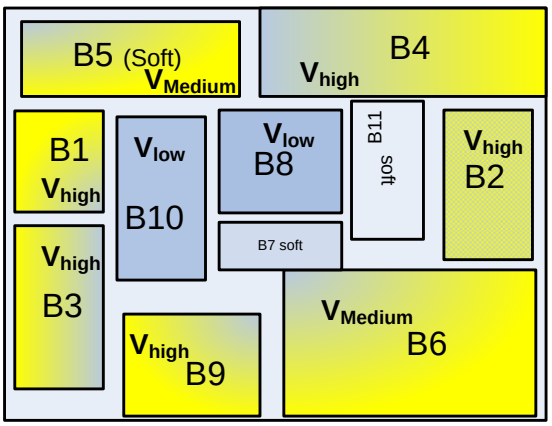
(a)



(b) All blocks works with same supply voltage



(c)



(d)

Fig: 2 Pictorial representation of the proposed algorithm

4.1 Development of Cost Function

The cost function for thermal aware placement and routing includes different parameters of interest

Cost Function = α (switching activity of Bi) + β (Power density) + γ (Proximity factor P_B)…….. (4)

Where ‘α’ represents the switching activity of each block based on the signal statics of the input. Here α, β and γare coefficient whose value range are less than ‘1’ (0< α, β and γ < 1).

$$P_B = \frac{1}{n} \sum_{i,j}^{n} \frac{p_i + p_j}{d_{ij}^2} \qquad \text{……(5)}$$

where p_i in equation (5) represents the power dissipation in block i and the Euclidean distance between the block i and block j edges is shown by d_{ij}, and n represents the number of blocks in the floor-plan that have a powerdensity greater than the average power density including the standard deviation.

$$T_{pd} = \frac{C_{Charge} V_0}{k(V_0 - V_t)^2} \qquad \text{……(6)}$$

$$P = C_{total} V_0^2 f \qquad \text{……(7)}$$

The variation in the supply voltage will affect the delay and power by equation (6) and (7). These are the computation of propagation delay of CMOS circuits and the other for computing the powerconsumption. T_{pd} represent the propagation delay which takes place because of charging and discharging of stray capacitances in the critical path. The power consumption of a CMOS circuit can be estimated using (7).

BUPT

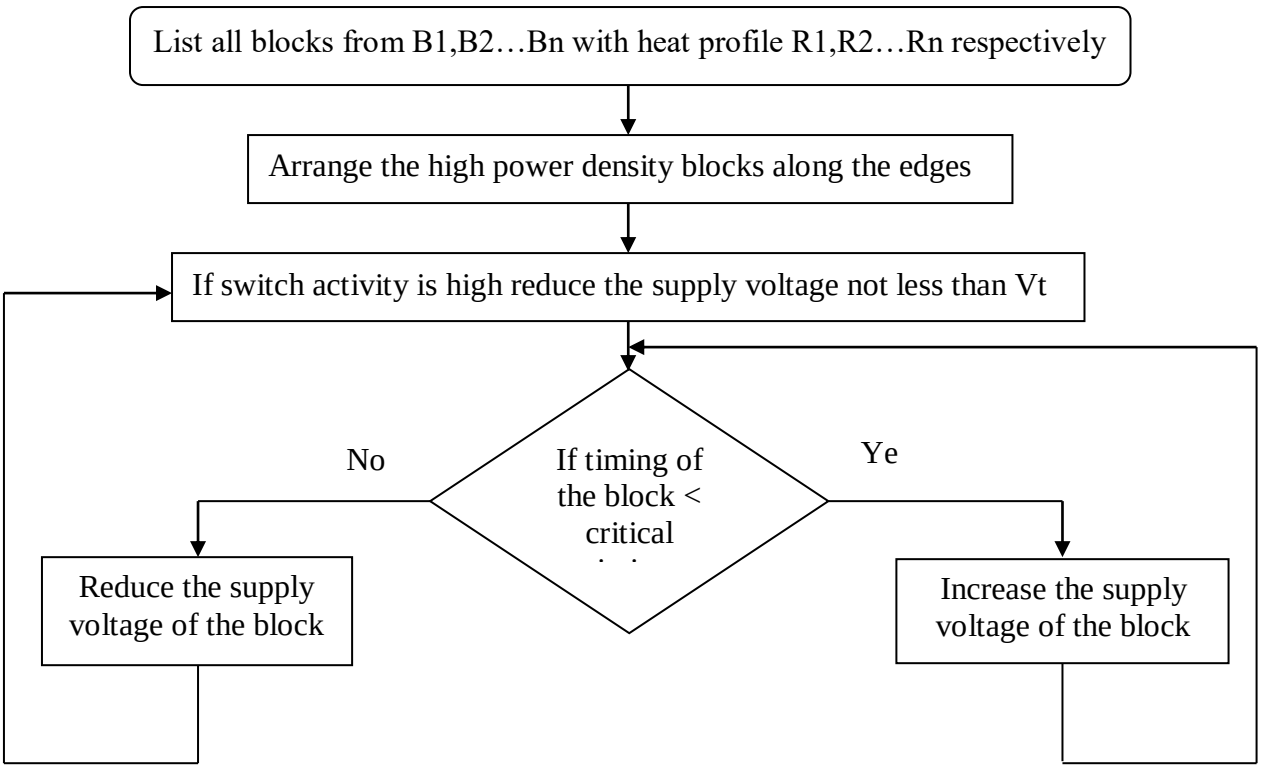


Fig: 3 Flow chart of the proposed algorithm

V. Simulation results and Comparison

The algorithm was implemented using matlab and the algorithm is tested with 10 blocks with 3 soft blocks and the remaining is fixed blocks. The temperature profile of each blocks are mapped with the specification of the blocks and they are floor planned according to the algorithm. The selection of the block was modelled with Verilog code and the same proposed architectures are synthesized in FPGA and ASIC platform. Fig.4 shows the matlab simulated results and Fig.5 shows the FPGA synthesized results of the algorithm architecture. Fig.6 shows the RTL synthesized view of the proposed architecture. The proposed algorithm is tested with two benchmark circuit ami33 and ami49 and the results are compared with previous existing algorithm and they are tabulated in Table.1, 2 and 3. The same architecture is also is implemented in the ASIC platform and the same is shown in fig.7.

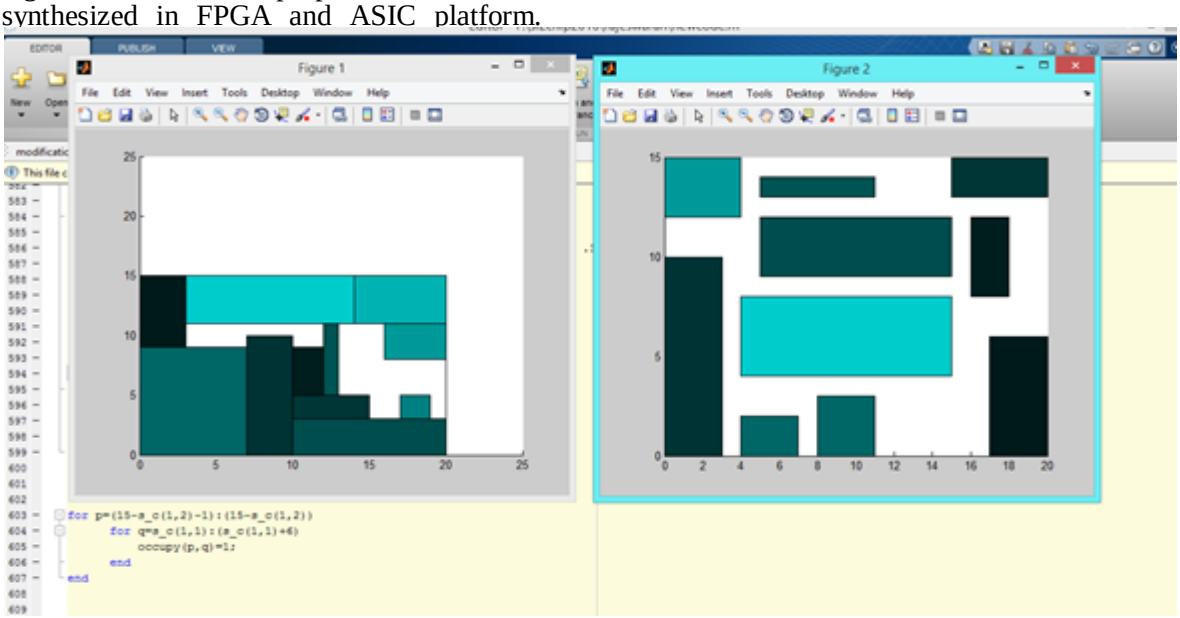


Fig: 4 Matlab simulation results of the proposed algorithm

BUPT

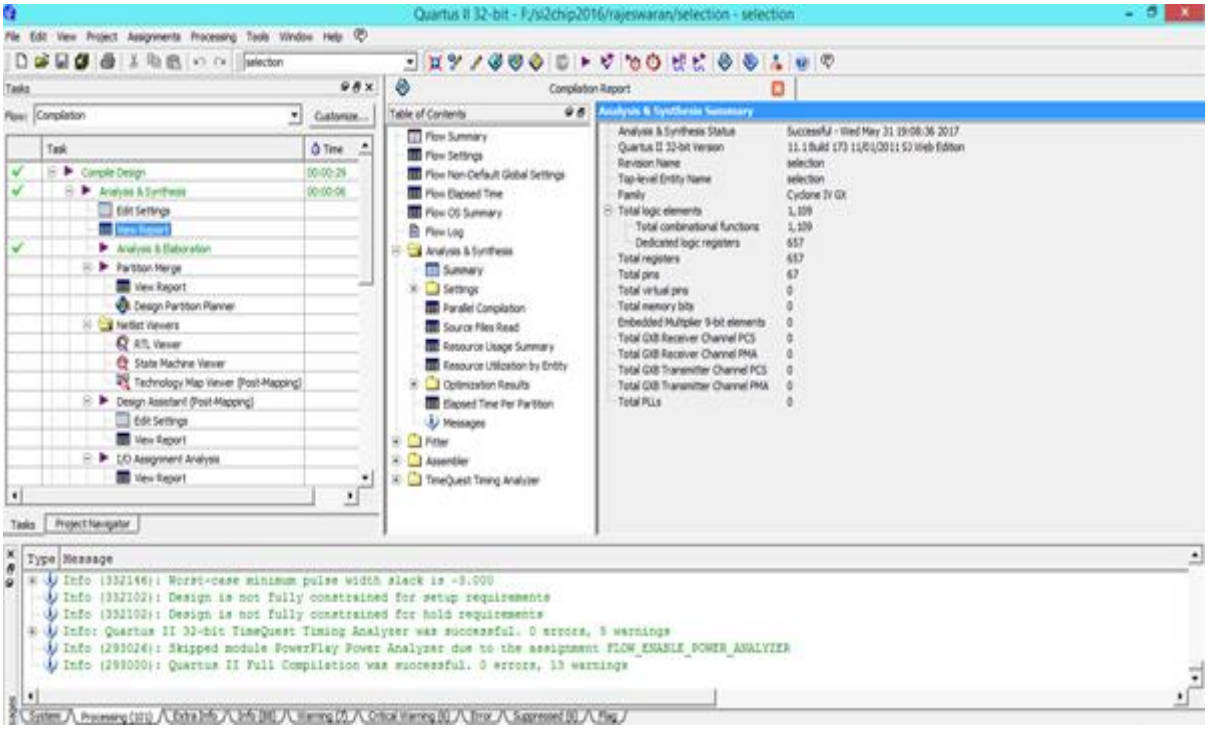


Fig:5 FPGA resource utilization results

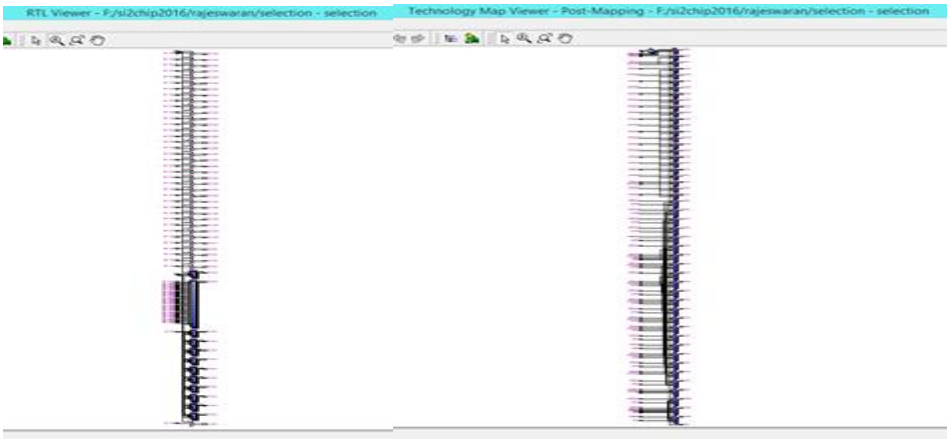


Fig. (a) Fig. (b)

Fig: 6(a) (b) RTL and Technology mapping of the proposed VLSI architecture

Table:1 Area optimized results for the proposed algorithm

Circuit	Design[12]	Design[14]	Proposed algorithm
	Area(mm^2)	Area(mm^2)	Area(mm^2)
Ami33	1.24	1.27	1.22
Ami49	38.7	38.86	38.5

Table:2 Timing results for the proposed algorithm

Circuit	Design[12]	Design[13]	Proposed algorithm
	Time(s)	Time(s)	Time(s)
Ami33	33	78.8	45
Ami49	22	23.86	24.05

Table:3 Temperature variation results for the proposed algorithm

Circuit	Design[12]		Design[13]		Proposed algorithm	
	Temp(C)		Temp(C)		Temp(C)	
	Before	After	Before	After	Before	After
Ami33	112.275	79.629	358.54	357.87	125.37	89.457
Ami49	129.263	79.538	455.47	444.89	138.357	93.672

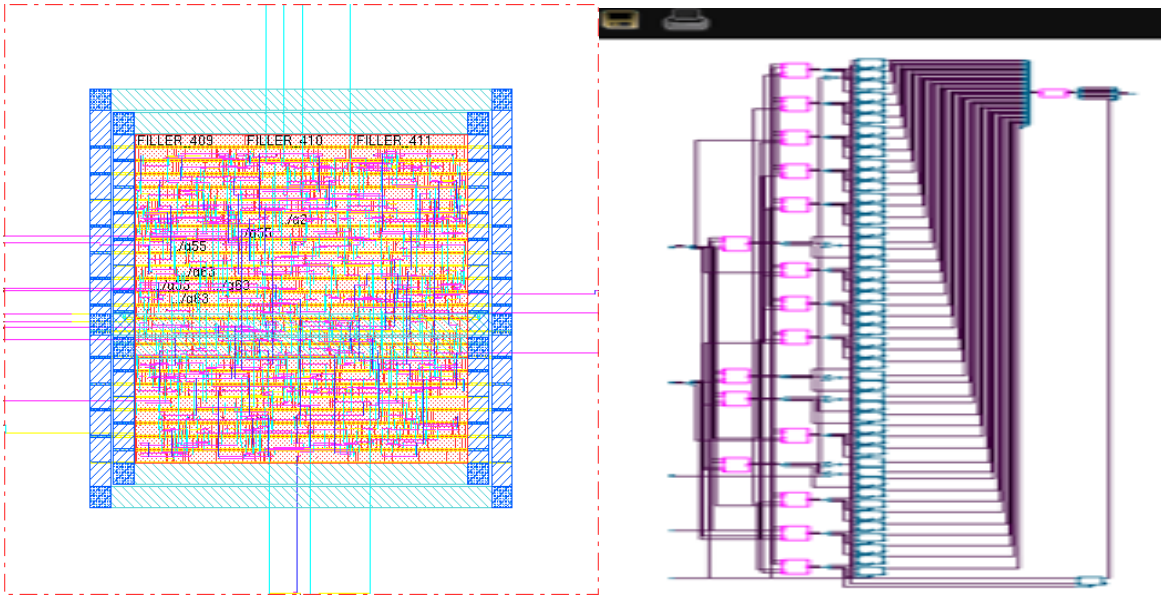


Fig: 7. ASIC implementation of proposed VLSI architecture

VI. Multiplier design for Signal processing using the proposed algorithm

The conventional practice of multiplication in digital signal processing systems involves interconnecting arithmetic elements such as adders, subtractors, multipliers and memories to perform the desired algorithms. Array multiplication algorithms such as Wallace tree [20] and Dadda's which are optimum suggests that when sum of products are to performed, a saving in hardware and improvement in speed results if the multiplication computation are not completed and the and added as is conventional.

In the merged arithmetic scheme [18] [19] to multiply two vectors of length N. We require N multiplication and N-1 addition to perform this operation. The concept of the merged arithmetic technique is instead of multiplying and adding the elements every time in the vector. We simply form a composite bit product matrix from the partial products of all the elements in vector. Then we reduce the composite bit product matrix.

The algorithm for the merged arithmeticis very simple and is shown in fig.8 Merged arithmetic is applied to inner product computation

BUPT

by removing the boundary between the multiplier and adder in of MAC (Multiplication and Accumulation) operation. Fig. 9 illustrates arithmetic scheme for L=7 and M=N=4 where L is inner product length and M and N are the word lengths of the operand vectors.

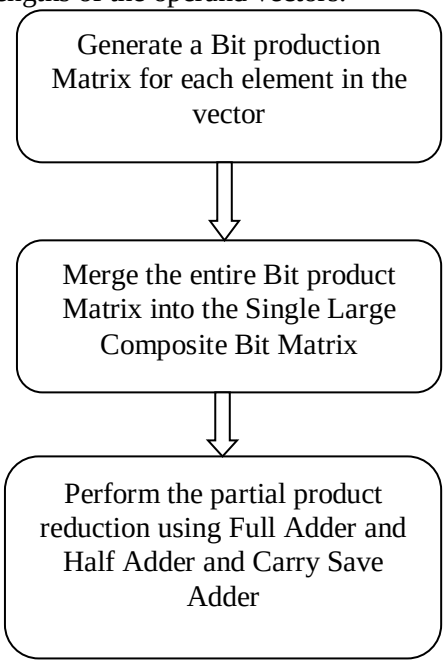


Fig. 8. Algorithm for merged arithmetic scheme

6.1 Unsigned Vectors

Consider, A=[A_0 A_1 A_2 ..... A_{L-1}] and B=[B_0 B_1 B_2 ..... B_{L-1}] between integer vectors of length L. The array product P=A × B canbeexpressed as,

$$P = \sum_{k=0}^{L-1} A_K . B_K = A_0 . B_0 + A_1 . B_1 + \dots\dots A_{L-1} . B_{L-1} \quad \text{---------- (8)}$$

For the sake of simplicity, take the unsigned vectors A_k and B_k first. Now we decompose the array productat the bit level, let A(*i*) and B(*j*) denote the i[th] and j[th] bits of A_k and B_k, respectively. The product P can be obtained as,

$$P = \sum_{k=0}^{L-1} \left[\left(\sum_{i=0}^{M-1} A_k(i) * 2^i \right) * \left(\sum_{j=0}^{N-1} B_k(j) * 2^j \right) \right] \quad \text{------ (9)}$$

where M and N are the word lengths of $A_k \epsilon A$ and $B_k \epsilon B$ respectively and the summation term inside the square brackets denotes the partial product bit matrix of $A_k * B_k$.By rearranging the sequences of L repeated accumulations and the M×N in equation (9), we have

$$P = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\sum_{k=0}^{L-1} A_k(i) . B_k(j) \right] . 2^{i+j}$$

$$P = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C(i,j) . 2^{i+j} \quad \text{-------- (10)}$$

Where $C_k(i,j) = \sum_{k=0}^{L-1} A_k(i) . B_k(j)$ and hence $0 \le C_k(i,j) \le L-1$.

The ripple counter will accumulate the number of 1's generating at i[th] and j[th] partial product bit positions of $C(i,j)$.The expression given in equation (10) can be realized by the semi-serial and semi-parallel architecture introduced in[21]. This version of architecture for unsigned integer array multiplication for M=N=4 and L=7 is shown in fig.(10). The architecture is having M×N number of processing element (PE), pipeline register, and the reduction tree i.e, Dadda tree or Wallace tree [22].

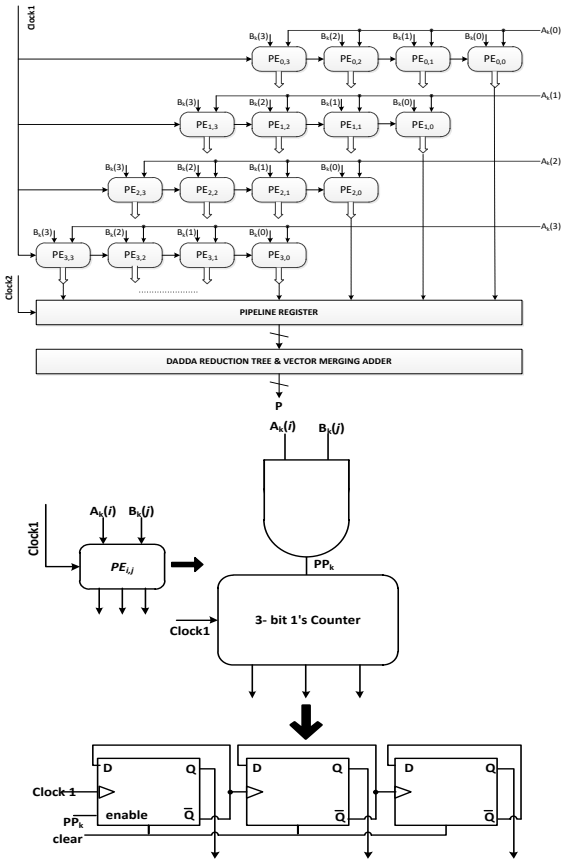


Fig.9 Array multiplier architecture for unsigned integer array

Each processing element has an AND gate and a binary ripple counter of width, $w = \lfloor \log_2 L \rfloor + 1$. Binary counters in all the PE's driven parallel by a common clock, *clock1*.At every clock cycle, a new set of operands from the array A and B is loaded to the

BUPT

processing elements. Ateach PE, L partial product bits will be generated in as many number of clock cycles [25]. The output of AND gateworks as the enable bit of the counter. If the AND gate output is 1 then counter will get incremented by 1 otherwise remains idle. L clock cycles will be required to accumulate all the partial product bits of C(i,j) which is equal to the length of th e integer array. The binary bits produced by the counter will be stored in the pipeline register. The pipeline register is clocked by a separate clock i.e, *clock2=L×clock1*.Therefore, the output of the pipeline register is fed into adder treat the positive edge of the clock2 to obtain the final product *P[26]*.

In this architecture, ateach PE counter has reduced the L vertical bits at bit position (i,j) to $\lfloor \log_2 L \rfloor + 1$ horizontal bits, therefore the complexity of the adder tree has been reduced logarithmically.

6.2 SignedVectors

The signed integers are always represented in the 2s complement form. Therefore representation of signed vectors A_k and B_k isgiven by:

$$A'_K = -A'_K(M-1).2^{M-1} + \sum_{i=0}^{M-2} A'_K(i).2^i$$

------- (11)

$$B'_K = -B'_K(N-1).2^{N-1} + \sum_{i=0}^{N-2} B'_K(j).2^j$$

------(12)

Here, the array product computation is based on the modified Baugh-wooley approach [23], the inner-product $P' = A' \times B'$ is computed as

$$P' = \sum_{k=0}^{L-1} A'_K \cdot B'_K = \sum_{k=0}^{L-1} \Big[A'_K(M-1)B'_K(N-1) \cdot 2^{M+N-2} - \sum_{j=M-1}^{M+N-3} 2^j + \sum_{i=0}^{M-2}\sum_{j=0}^{N-2} A'_K(i)B'_K(j) \cdot 2^{i+j} - \sum_{i=N-1}^{M+N-3} 2^i + \sum_{j=0}^{N-2} (1 - A'_K(M-1) \cdot B'_K(j))2^{j+N-1} + \sum_{i=0}^{M-2} (1 - B'_K(N-1) \cdot A'_K(i))2^{i+N-1} \Big]$$

---------- (13)

Using the identity, $\sum_{i=j}^{n-1} 2^i = 2^n - 2^j$, the constraints in the above equation can be grouped into:

$$-\sum_{j=M-1}^{M+N-3} 2^j - \sum_{i=N-1}^{M+N-3} 2^i = -2^{M+N-1} + 2^{M-1} + 2^{N-1} \quad (14)$$

By substituting (13) into (14), and swapping the accumulation order of L,M and N in (15), weobtain

$$P' = C_k \cdot 2^{M+N-2} + \sum_{i=0}^{M-2}\sum_{j=0}^{N-2} C'_k(i,j) \cdot 2^{i+j} + \sum_{j=0}^{N-2} \sum_{i=0}^{M-2} + \varepsilon \quad -- \quad (15)$$

where,

$$C_k = \sum_{k=0}^{L-1} [A'_k(M-1)B'_k(N-1)]$$

$$C'_k(i,j) = \sum_{k=0}^{L-1} A'_k(i)B'_K(j)$$

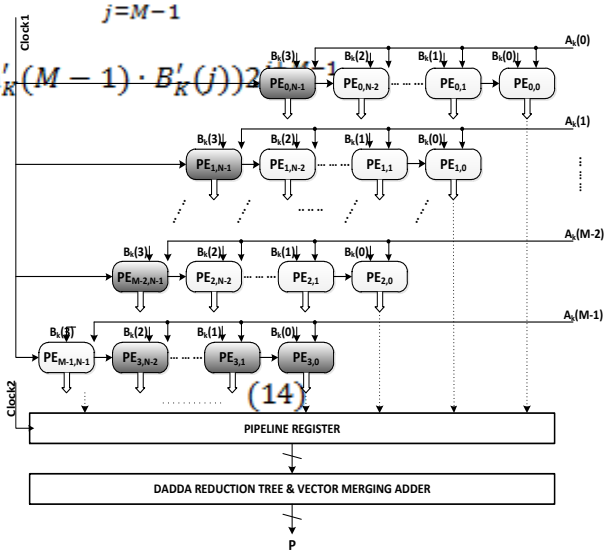
$$C''_k(j) = \sum_{k=0}^{L-1} \overline{A'_k(M-1) \cdot B'_k(j)}$$

$$C'''_k(i) = \sum_{k=0}^{L-1} \overline{B'_k(N-1) \cdot A'_k(i)}$$

$$\varepsilon = L \times (-2^{M+N-1} + 2^{M-1} + 2^{N-1})$$

---------- (16)

The partial products bit generation and accumulation for signed integer array multiplier [24] can be explained with the help of fig.4 for L = 7 and N = M = 4, here two types of PE's are used. To realize $C_k$and $C'_k(i,j)$, the PE implemented for unsigned is used and to realize the $C_k(i,j)$ and $C'''_k(i)$ the AND gate in the PE is replaced by the NAND gate which is shown as a darker PE in the architecture. The error constant ε is calculated separately using the equation (9) and then added in the output of the reduction tree to get the final array product output.



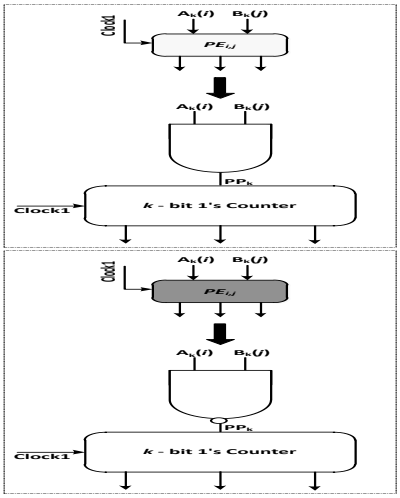


Fig. 10 Array multiplier architecture for signed integer array

6.3 Comparison of synthesized results with existing algorithm

The advantage of the proposed algorithm on the arithmetic scheme is demonstrated by synthesizing both the architectures in TCMS 180nm CMOS process with integer size of M, N bits and integer array of length L=7with M=N=4. The merged arithmetic technique requires a total time of 2×M×L bits to latch the overall set of serial inputs in L cycles for computations. The array architecture isalsoimplementedwith 2×M input pads for the fair comparison. The results of synthesized area and delay of both the architectures are compared respectively in the Table.4. In the table it is shown that the architecture based on thermal floorplan based architecture occupies less silicon area and consumes less powerthan the existing architecture.

From the comparison table it is evident that the area is decreased around 68.45% when compared with the existing architecture. Power is also decreased by 9% in the wallace reduction tree counter based technique when compared with the remaining two architectures. Dadda reduction tree based approach takes least time among the two techniques for computation of the final output i.e, 33% less time as compared to existing algorithm.

Table.4. Comparison between proposed and conventional architecture for multiplier design

Parameters	Wallace Tree architecture		Dadda Tree architecture	
	Proposed	Conventional	Proposed	Conventional
Area (µm2)	9886	10927	9756	10617
Power (µW)	4113.183	4234	4478	4609
Timing (nSec)	3.315	3.8	2.37	2.49

References

[1] W. Huang, M.R. Stan, K. Skadron, S. Ghosh , S. Velusamy (2004). Compact Thermal Modeling for Temperature-Aware Design. *Proc.DAC*, San Diego, CA, USA, pp. 878-883.

[2] J. Srinivasan, S.V. Adve, P. Bose, J.A. Rivers (2004). The Impact of Technology Scaling on Lifetime Reliability. *Proc. DSN*, Florence, Italy, pp 177-186.

[3] J. Lee (2006). General Thermal Force Model with Experimental Studies. *Trans.on Packaging*, vol. 29, no. 1, pp. 20–29.

[4] W.L. Hung et al. (2005). Thermal-Aware Floorplanning Using Genetic Algorithms. *ISQED*, pp. 634–639.

[5] A. Gupta et al. (2007). Leaf: A System Level Leakage-Aware Floorplanner For Socs. *ASP-DAC*, pp. 274–279.

[6] J.-L. Tsai et al. (2006). Temperature-Aware Placement for Socs. *Proc. of the IEEE*, vol.94, no.8, pp. 1502–1518.

[7] J. Cong, J.Wei, and Y. Zhang (2004). A Thermal-Driven Floorplanning Algorithm for 3D ICs. *ICCAD*, pp. 306–313.

[8] W.-L. Hung et al. (2006). Interconnect and Thermal-Aware Floorplanning for 3D Microprocessors. *ISQED*, pp. 98–104.

[9] P. Zhou et al. (2007). 3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three-Dimensional Integrated Circuits. *ICCAD*, 2007, pp.590–597.

[10] B. Goplen and S. Sapatnekar (2003). Efficient Thermal Placement of Standardcells in 3d ICs Using a Force Directed Approach. *ICCAD*, p. 86.

[11] X. Li et al. (2007). Thermal-Aware Incremental Floorplanning for 3D ICs. *ASICON*, pp.1092–1095.

BUPT

[12] Sheldon Logan, Matthew R. Guthaus. Fast Thermal-Aware Floorplanning Using White-Space Optimization. Santa Cruz, CA 95064.
[13] Lixia Qi Yinshui Xia Lunyao Wang (2011). Simulated Annealing Based Thermal-Aware Floorplanning. *IEEE*, pp-463-466.
[14] GraciaNirmalaRani.D, Rajaram.SNivethitha and AthiraSudarsan (2015). Thermal Aware Modem VLSI Floorplanning. *IEEE*.
[15] Ehsan K. Ardestani, Amirkoushyar Ziabari, Ali Shakouri, and Jose Renau (2012). Enabling Power Density and Thermal-Aware Floorplanning. *28th IEEE SEMI-THERM Symposium*, Santa Cruz, CA, 95064.
[16] Chih-han Hsu, Shanq-Jang Ruan, Ying-Jung Chen and Tsang-Chi Kan (2013). Reliability Consideration with Rectangle and Double-Signal Through Silicon Vias Insertion in 3D Thermal–Aware Floor Planning. *14th Int'l Symposium on Quality Electronic Design. IEEE.*
[17] David Guilherme, João Pereira, NunoHorta, Jorge Guilherme (2013). Thermal-Aware Floor Planning and Layout Generation of MOSFET Power Stages. *This work was supported by FCT project PEst-OE/EEI/LA0008. and Instituto de Telecomunicações*.
[18] C. Vinoth, V.S. KanchanaBhaaskaran, B. Brinda, S. Sakthikumaran, V. Kavinilavu, B. Bhaskar, M. Kanagasabapathy and B. Sarath (2011). A NovelLow Power and High Speed Wallace Tree Multiplier for RISC Processor. *IEEE trans.* 978-1-4244-8679-3/11.
[19] Marco Castellano, Paola Baldrighi, Carla Vacchi and Mauro Natuzzi (2008). Algorithm and Architecture for High Speed Merged Arithmetic FIR Filter Generation. *IEEE trans*. 978-1-4244-1688-2/08.
[20] Earl. A. Swartzlander (1978). Merged Arithmetic for Signal Processing. *IEEE trans* CH1412-6/78/0000-0239.
[21] S. A. White (1989). Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review. *IEEE Signal Process*. *Mag*., vol. 6, no.3, pp. 4–19.
[22] B. Parhami (2000). Computer Arithmetic: Algorithms and Hardware Designs. New York: Oxford Univ. Press.
[23] M. R. Meher, C. C. Jong, C. H. Chang, and J. Y. S. Low (2010). A Novel Counter-Based Low Complexity Inner Product Architecture for High Speed Inputs. *Proc. Int. Symp. Circuits Syst. ISCAS*. Paris, France, pp. 705–708.
[24] K. A. Feiste and E. E. Swartzlander, Jr (1997). Merged Arithmetic Revisited. *Proc. IEEE Int. Workshop Signal Process. Syst. (SIPS'97)*, Leicester, U.K., pp. 212–221.
[25] Manas Ranjan Meher, Member, IEEE, Ching Chuen Jong, and Chip-Hong Chang (2012). An Area and Energy Efficient Inner-Product Processor for Serial-Link Bus Architecture. *IEEE trans on circuits and systems--I: regular papers*, vol. 59, no. 12.