International Journal of Innovative Research in Engineering
Special Issue, Volume 4, Issue 3(May-June 2023), PP: 82-87.
Innovations in Computing, Materials and Communication Technologies (ICICMCT'23) **ISSN No: 2582-8746**

Pulsed Latches Based Parallel Pipelined Architecture and Algorithm for MatrixTransposition

Shaheela A M[1], Anitta Antony[2]
[1] Student, M-tech in VLSI Design, ECE, IES College of Engineering, Chittilappilly, Kerala, India.
[2]Assistant Professor, Department of ECE, IES College of Engineering, Chittilappilly, Kerala, India.

How to cite this paper:
Shaheela A M[1], Anitta Antony[2],"Pulsed Latches Based Parallel Pipelined Architecture and Algorithm for Matrix Transposition", IJIRE-V4I03-82-87.

Abstract: *This paper proposes a new algorithm and architecture for continuous flow matrix transposition using pulsed latches. The algorithm supports P-parallel matrix transposition. The hardware architecture reaches the theoretical minimum in terms of memoryand latency. Low power consumption and area efficiency is the most advantage of using pulsed latches. Instead of using conventional single pulsed clock signals, here we use multiple non overlap delayed pulsed clock signals in order to reduce the timing problems between pulsed latches. Compared with the state-of-the-art architecture, the proposed architecture supports matrices whose rows and columns are integer multiples of P. Here P can be arbitrary, includingbut not limited to power-of-two-integers. More over our result provide additional insight into continuous flow non-square matrix transposition.*

Key Word: *Continuous flow; Matrix transposition; State-of-art architecture;*

I.INTRODUCTION

A matrix is a square or rectangular array of numbers or functions that arranged in a fixed number of rows and columns. In Linear algebra, the transpose of a matrix is one of the most commonly used methods in matrix transformations.For a given matrix, the transpose of a matrix is obtained by interchanging rows into columns and columns to rows. The matrix transpose s denoted by the letter T. The application of matrix transposition is extravagant these days. In image signalprocessing, matrix transposition is used for adding brightness or applying filters or image rotation. As matrices are the fundamental concepts in AI, matrix transpose leads a major role in machine learning. As a key component in matrix-basedalgorithms, it plays an important role in deep learning applications, including computer vision and nature language processing. Specifically, matrix transpose operations are often implemented in fully connected layers, convolutional layers,and self-attention layers. To reduce the area and power consumption flip-flops are replaced with pulsed latches. This paperuses shift registers using pulsed latches instead of using master slave flip flops. Such a shift register using pulsed latches reduces the power consumption and area. The timing problem is solved by using multiple non-overlap delayed pulsed clocksignals instead of the conventional single pulsed clock signal. The shift register used additional temporary storage latch.

II.LITERATURE SURVEY

The crucial role of matrix transpose in SAR (Synthetic Aperture Radar) imaging technique and introduced two new matrix transpose methods and implemented. Two algorithms namely Range-Doppler algorithm (RD) and Chirp-Scaling algorithm (CS) has introduced. Range-Doppler (RD) algorithm is the most effective and commonly used algorithm in SAR imaging technique. It splits two dimensional images to one dimensional processing. The main phases of RD algorithm is Preprocessing, Range compression, Matrix transpose and Azimuth compression. To the improvement of RD algorithm introduced CS (Chirp-Scaling) algorithm is introduced. CS algorithm maintains a good phase-precision by using a phase factor namely Chirp-Scaling factor. Chirp-Scaling factor is used to change the phase shift property of migration in order toavoid the interpolation operation. CS algorithm requires a Preprocessing, Azimuth FFT, Range FFT, Range IFFT and Azimuth IFF.[1]

The bit reversal on series of data and took simple and optimum circuits. Simple in the sense it consisted of only buffers and multiplexers and optimum in two senses: first it uses only minimum number of registers that are necessary for calculating bit reversal and the second one is minimum latency. The circuits is made optimum in two senses: minimum latency and minimum storage elements. The output frequencies of the FFT are sorted out in pipelined architecture as the circuits are optimum for that functioning. Optimum circuits have been found out.[17]

A pipelined algorithm and architecture for finding out the transpose of N x N matrix. It uses registers and gives minimum memory and latency. The circuit has a simple control strategy and is a identical cascaded basic circuit. It can be only applicable for square matrix.Matrix transposition is a simple operation that can be expressed as

$$A_{i,j} = A_{j,i} \text{ for all, } i, j = 0, 1, 2, ..., N - 1.$$

Consider an N × N matrix and its transposed matrix. The matrix transposition can be found out by reversing the elements on every diagonal from the bottom left to the top right. We can employ a series of steps and permutations among elements on each diagonal to achieve this reversal. Based on the above considerations, we present a pipelined matrix transposition algorithm. Modular pipelined architecture is introduced and is achieved by series of permutations namely Basic permutation circuit and Cascade transposition architecture. In basic permutation circuit, the bit reversal is done by swapping the position of two elements with a fixed interval. And in cascaded transposition architecture composed of N − 1cascade basic permutation circuits with a shift register (SR) of length N−1. It includes only identical circuits. [8]

A unique pipelined algorithm for transposing non-square matrices and describe the corresponding architecture forthis algorithm. This architecture composed of series of cascaded basic circuit and is controlled by control strategy. The proposed architecture and algorithm. Besides, the proposed calculation and engineering could be handily reached out to N- equal executions for lattice interpretation. This architecture upholds lattices whose lines and segments are number products; it is essentially utilized for radix-2s butterfly calculations utilizing lattice interpretations. Trial results show that the proposed single-way design can diminish the calculation cycles and circuit region by an element of 9.18% and 5.87%, respectively, for a 32×16 lattice interpretation calculation, looked at with those of an as of late proposed best in class engineering for gridinterpretation. Matrix transposition is a fundamental operation that can be expressed as

$$A_{i,j} = (A^T)_{j,i}, \text{ where } i = 0, 1, 2,...,NR - 1 \text{ and } j = 0, 1, 2,...,NC - 1.$$

Two methods are taken into consideration Basic serial exchange circuit and cascade transposition architecture. In basic serial exchange circuit Garrido et al proposed a piece inversion circuit that can trade the places of two components with a decent stretch between them, where S and S¯ represent the control flags that can be utilized to change the status of the two multiplexers in the circuit to one or the other pass-by or trade modes; what's more, L alludes to the length of the cushion utilized for component trading.

If a matrix has NR × NC number of rows and columns respectively, the transposition architecture composed of N – 1 cascaded permutation. The proposed NR × NC lattice interpretation engineering is made out of N − 1 fountain fundamental change circuits with a shift register (SR) of length N −1 and (K −1)(N −1) overflow fundamental change circuits with a SRof length N. Altogether, the proposed overflow interpretation architecture is created of K(N − 1) overflow fundamental change circuits. It is extended up to N parallel matrix and also supports continuous data, and is suitable for realizing pipelined multi-dimensional FHT and FFT operations. Image processing and transposition of non-square matrix is found out here.[12]

III.EXISTING SYSTEM

The existing system is to find the matrix transposition in a parallel and pipelined manner using registers consisting of flip flops. The transpose of any matrix such as square or non-square matrix can be found out using this. It is a basic pipelined algorithm and can be used to find p parallel matrix transposition. The transposition is found in 3 steps. Step A B & C.

This architecture is meant for P-parallel matrix transposition. The matrix transposition is calculated by a set of basic permutations along with a circuit and a set of algorithms in a single path way. Cascaded P-parallel architecture and iscontrolled by control strategy. Memory and latency is also take into consideration. Cascaded P-architecture involves two PEs. PE_1 and PE_{II} . The proposed architecture composed of shift registers cascaded serially to perform p-parallelpermutations. The function of PE_1 is to exchange data points between different input ports and it completes step B. Step A & C is completed by PE_{II} architecture where, data points are exchanged same input ports but in different clock cycles.

Several counters control the control strategy of above architecture. The control signals are denoted by S_n where, n is the number of stages in step A,B & C. The control signals are generated by three counters, called C_0,C_1 and C_2. The control strategies in each steps are: In Step A, n ∈ [0, $(n_c - 1)(P - 1) - 1$], S_n as follows:

$$S_n = \ 1\ LOC_1 \leq C0 \leq LOC_1 + (offset_1 - 1{,}0 \text{ else.}$$

In Step B, n ∈ [0, P − 2], S_n as follows:

$$S_n = 1\ 1 \leq C1 \leq P - n - 1{,}0 \text{ else.}$$

In Step C, n ∈ [0, $((n_r - 1)(\ (n_c P - 1) - 1$], S_n as follows:

$$S_n = \ 1\ LOC_2 \leq C2 \leq LOC_2 + offset_2 - \ -1{,}0 \text{ else.}$$

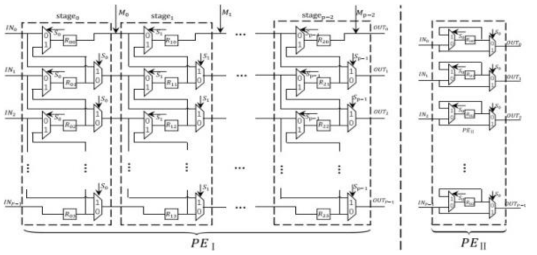


Fig 3.1 P-parallel Pes in algorithm 1

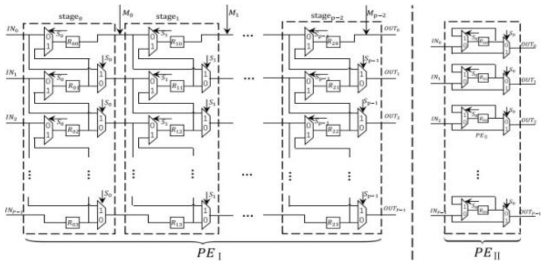


Fig 3.2 P-parallel architecture block diagram

Considering a N_R X N_C matrix, the number of $stage_1$ in step A is (n_C -1) X (P – 1), the number of $stage_2$ in stepB is

P – 1, the number of $stage_3$ in step C is $(n_C \text{ P} - 1) \times (n_r - 1)$; so, the total size of the memory is as follows:

$$D = (n_c - 1) \times (P - 1) \times P + (P - 1) \times P + (n_c\ P - 1) \times (n_r - 1) \times P,$$
$$= (n_c P - 1)(n_r P - 1) + P - 1,$$
$$= (N_C - 1)(N_R - 1) + P - 1$$

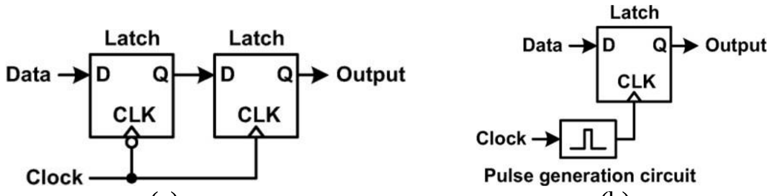
and the total latency is as follows:

$$L = D/P\ , = [(N_C - 1)(N_R - 1) + P - 1]/P\ .$$

This architecture achieves a minimum latency and memory. This proposed architecture uses more multiplexers than others to support P-parallel matrix algorithm for any square and non-square matrix transposition. The control strategyused here is somehow easy to implement and execute.

IV. PROPOSED ARCHITECTURE

A low power and area efficient pulsed latches is proposed in this paper instead of flip flops that takes a larger areaand consumes high power. By using pulsed latches timing problems may happen in the entire circuitry. So, in order to avoidthat uses multiple non-overlapped delayed pulse clock signals instead of using conventional single pulsed clock signals. Additional temporary storage latches are included. Master slave flip flop contacting two latched is being replaced by a pulsed latch consisting of pulsed clock signals as shown in Fig 4.1. By using this method, the required area and power consumption become half and it became low power and area efficient design procedure. Pulsed latches have timing problems inbuilt so, it cannot be used directly in shift registers.



(a) (b)

Fig 4.1 (a) Master slave flip flop (b) pulsed latches.

A shift register is proposed which is divided into M sub shift registers in order to reduce the number of delayedpulsed clock signals. This delayed pulsed clock signals are generated by delayed pulsed clock generator. For better understanding, a 4-bit sub shift register consists of 5 latches in which the 4 latches store 4-bit data and last latch store 1-bittemporary data. Here 5 non-overlap delayed pulsed clock signals are generated

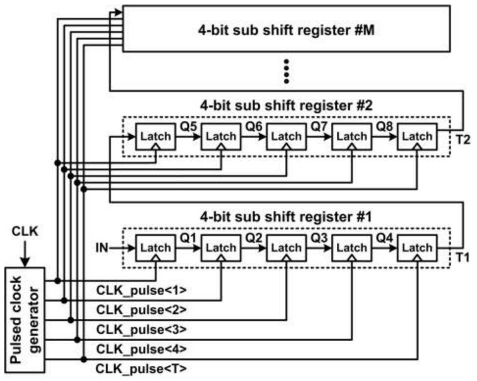


Fig 4.2 schematic of proposed shift register

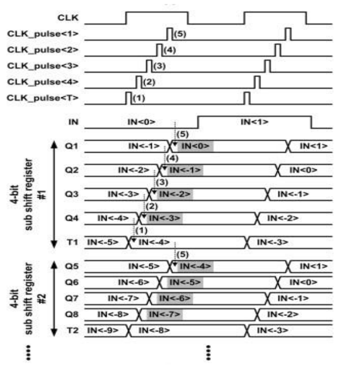


Fig 4.3 waveform of proposed shift register

The proposed shift register used delayed pulsed clock generator which uses an extra AND gate than conventional delayed pulsed clock generator. But in conventional delayed pulsed clock generator, the width of the clock pulse is larger than the summation of raising and falling times in delay circuits to keep the shape. However, the clock width is comparatively shorter in delayed pulsed clock generator since the sharp pulse clock signal is generated by the AND gate and two delayed signals. The power and area optimization goes hand in hand. The power is mainly consumed by latches for data transition and clock loading and the total area of utilization also depends on clock loading of latches.

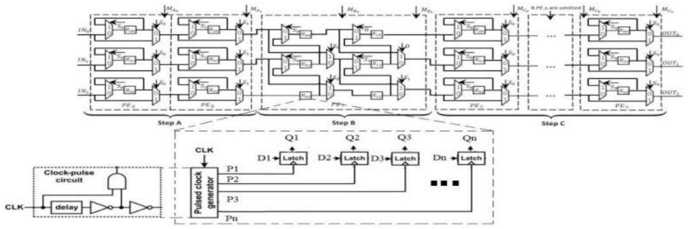


Fig 4.4 Proposed matrix transposition model using pulsed latches and delayed pulsed clock generator.

V.EXPIREMENTAL RESULTS

The proposed matrix transposition architecture is stimulated on the ModelSim software tool. The existing and proposed architectures are evaluated in the FPGA Xilinx 14.4 ISE tool. It is analyzed ina system consisting of Intel i5 processor with 8 GB of RAM and 500 GB harddisk.

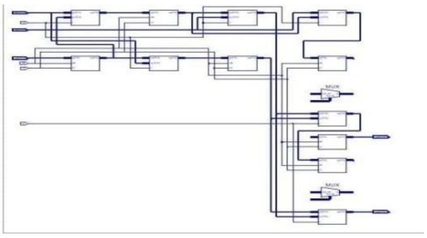


Fig 5.1 RTL view of PE1 algorithm.

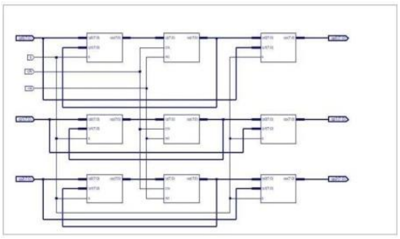


Fig 5.2 RTL view of PE II algorithm.

Fig 5.3 RTL view of 9 X 6 matrix

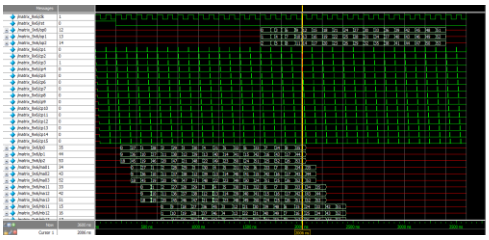
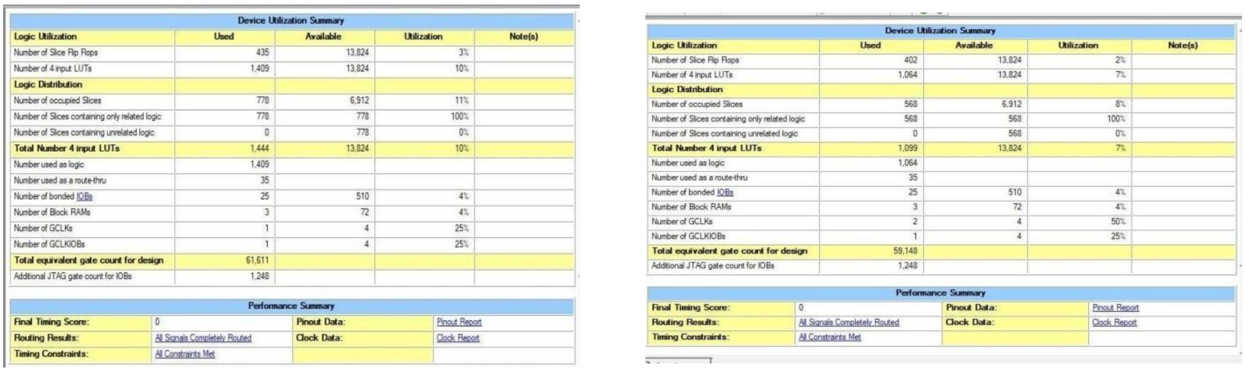


Fig 5.4 Output timing diagram of 9 X 6 matrix

VI.COMPARATIVE ANALYSIS

The proposed system comes with an effective area, low power consumption and the delay element is also low. Comparing the existing and proposed system in terms of area, power and delay



Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	435	13,824	3%	
Number of 4 input LUTs	1,409	13,824	10%	
Logic Distribution				
Number of occupied Slices	778	6,912	11%	
Number of Slices containing only related logic	778	778	100%	
Number of Slices containing unrelated logic	0	778	0%	
Total Number 4 input LUTs	1,444	13,824	10%	
Number used as logic	1,409			
Number used as a route-thru	35			
Number of bonded IOBs	25	510	4%	
Number of Block RAMs	3	72	4%	
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	61,611			
Additional JTAG gate count for IOBs	1,248			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

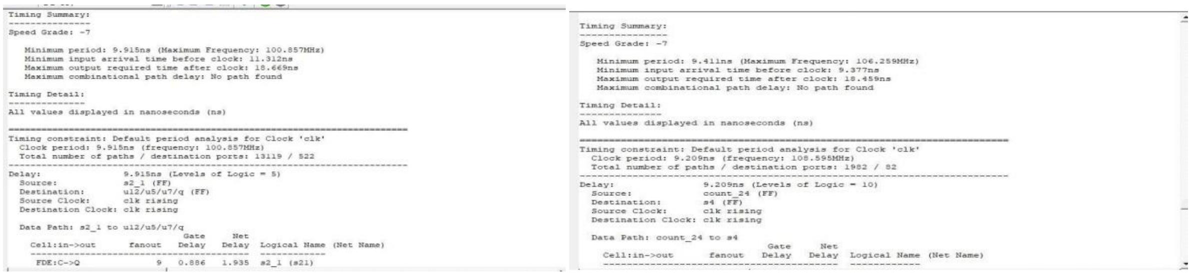
(a)

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	402	13,824	2%	
Number of 4 input LUTs	1,064	13,824	7%	
Logic Distribution				
Number of occupied Slices	568	6,912	8%	
Number of Slices containing only related logic	568	568	100%	
Number of Slices containing unrelated logic	0	568	0%	
Total Number 4 input LUTs	1,099	13,824	7%	
Number used as logic	1,064			
Number used as a route-thru	35			
Number of bonded IOBs	25	510	4%	
Number of Block RAMs	3	72	4%	
Number of GCLKs	2	4	50%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	59,148			
Additional JTAG gate count for IOBs	1,248			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

(b)

Fig 6.1 Area utilization summary (a) existing system (b) proposed system.



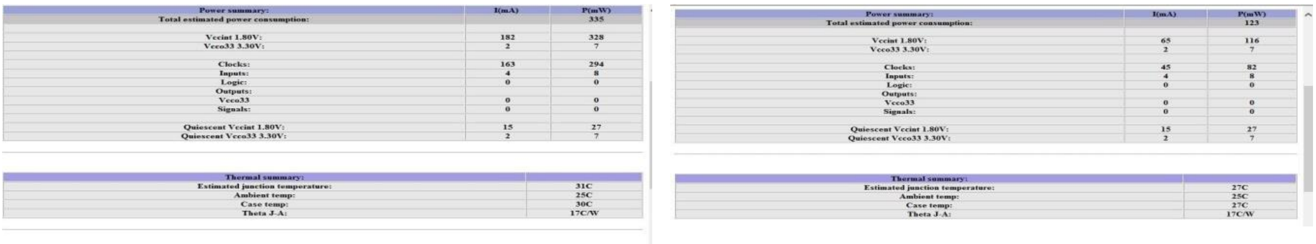
```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 9.915ns (Maximum Frequency: 100.857MHz)
   Minimum input arrival time before clock: 11.312ns
   Maximum output required time after clock: 18.669ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 9.915ns (frequency: 100.857MHz)
  Total number of paths / destination ports: 13119 / 522
-------------------------------------------------------------------------
Delay:               9.915ns (Levels of Logic = 5)
  Source:            s2_1 (FF)
  Destination:       u12/u5/u7/q (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: s2_1 to u12/u5/u7/q
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDE:C->Q              9   0.886   1.935  s2_1 (s21)
```

(a)

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 9.411ns (Maximum Frequency: 106.259MHz)
   Minimum input arrival time before clock: 9.377ns
   Maximum output required time after clock: 18.459ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 9.209ns (frequency: 108.595MHz)
  Total number of paths / destination ports: 1982 / 82
-------------------------------------------------------------------------
Delay:               9.209ns (Levels of Logic = 10)
  Source:            count_24 (FF)
  Destination:       s4 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: count_24 to s4
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
```

(b)

Fig 6.2 Delay analysis of (a) existing system (b) proposed system.



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		335
Vccint 1.80V:	182	328
Vcco33 3.30V:	2	7
Clocks:	163	294
Inputs:	4	8
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	31C
Ambient temp:	25C
Case temp:	30C
Theta J-A:	17C/W

(a)

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		123
Vccint 1.80V:	65	116
Vcco33 3.30V:	2	7
Clocks:	45	82
Inputs:	4	8
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	27C
Ambient temp:	25C
Case temp:	27C
Theta J-A:	17C/W

(b)

Fig 6.3 Power Comparison of (a) Existing system (b) proposed system.

VII.CONCLUSION

Transpose of a matrix is performed in a circuit consisting of pulsed latches in which the clock pulse is generated by delayed pulsed clock generator. The newly proposed circuit gives a low power and area efficient design. The delay problems occurring in the pulsed latches normally is kick stopped by using delayed pulsed clock generator instead of conventional clock pulse generator. Multiple non-overlapped delayed pulsed clock generators are used. The latches are grouped for generating a small number of pulsed clock to several sub shift registers and by using additional latches for temporary data storage.

The transposition of continuous flow matrix is found out. Here, P-parallel architecture is used to find the transposeof any row and column matrix. The architecture gives a theoretical minimum in terms of memory and latency.

References

1. *M. Bian, F. Bi, and F. Liu, "Matrix transpose methods for SAR imaging system," in Proc. IEEE 10th Int. Conf. SignalProcess., Beijing, China, Oct. 2010, pp. 2176–2179.*
2. *D. Im, D. Han, S. Choi, S. Kang, and H.-J. Yoo, "DT-CNN: An energyefficient dilated and transposed convolutional neural network processor for region of interest based image segmentation," IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 67, no. 10, pp. 3471–3483, Oct. 2020.*
3. *F. Mahmood, M. Toots, L.-G. Öfverstedt, and U. Skoglund, "2D discrete Fourier transform with simultaneous edge artifact removal for realtime applications," in Proc. Int. Conf. Field Program. Technol. (FPT), Queenstown, New Zealand, 2015, pp. 236–239.*
4. *S. Murugan and K. Jayakumar, "A DSP based real-time 3D FFT system for analysis of dynamic parameters," in Proc. IEEE Int. Conf. Adv. Commun. Control Comput. Technol., Ramanathapuram, India, 2014, pp. 1489–1492*
5. *H. Song, "The application of computer vision in responding to the emergencies of autonomous driving," in Proc. Int.Conf. Comput. Vis. Image Deep Learn. (CVIDL), Chongqing, China, 2020, pp. 1–5.*
6. *A. Barnard, "The nursing profession: Implications for AI and natural language processing," in Proc. Int. Conf. NaturalLang. Process. Knowl. Eng., Beijing, China, 2007, pp. 497–501.*
7. *K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput.Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.*
8. *Y. Wang, Z. Ma, and F. Yu, "Pipelined algorithm and modular architecture for matrix transposition," IEEE Trans.Circuits Syst. II, Exp. Briefs, vol. 66, no. 4, pp. 652–656, Apr. 2019.*
9. *M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bitdimension permutations," IEEE Trans. Very LargeScale Integr. (VLSI) Syst., vol. 27, no. 5, pp. 1148–1160, May 2019.*
10. *M. Garrido and P. Pirsch, "Continuous-flow matrix transposition using memories," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3035–3046, Sep. 2020.*
11. *M. V. Noskov and V. S. Tutatchikov, "Modification of a two-dimensional fast Fourier transform algorithm by theanalog of the Cooley–Tukey algorithm for a rectangular signal," Pattern Recognit. Image Anal., vol. 25, no. 1, pp. 81–83, Jan. 2015.*
12. *B. Zhang, Z. Ma, and F. Yu, "A novel pipelined algorithm and modular architecture for non-square matrix transposition," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 68, no. 4, pp. 1423–1427, Apr. 2021.*
13. *S. Yang, Z. Quan, M. Nie, and W. Yang, "TransPose: Keypoint localization via transformer, " in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 11802–11812.*
14. *M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bit reversal," IEEE Trans. Circuits Syst. II, Exp.Briefs, vol. 58, no. 10, pp. 657–661, Oct. 2011.*
15. *C. Cheng and F. Yu, "An optimum architecture for continuous-flow parallel bit reversal," IEEE Signal Process.Lett., vol. 22, no. 12, pp. 2334–2338, Dec. 2015.*
16. *T. Järvinen, P. Salmela, H. Sorokin, and J. Takala, "Stride permutation networks for array processors," J. VLSI Signal Process. Syst.*

Signal Image Video Technol., vol. 49, no. 1, pp. 51–71, 2007.

17. *M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bit reversal," IEEE Trans. Circuits Syst. II, Exp.Briefs, vol. 58, no. 10, pp. 657–661, Oct. 2011.*
18. *L. Wei and J. Mellor-Crummey, "Autotuning tensor transposition," in Proc. IEEE Int. Parallel Distrib. Process.Symp. Workshops (IPDPSW), 2014, pp. 342–351.*
19. *J. O. Eklundh, "A fast computer method for matrix transposing," IEEE Trans. Comput., vol. C-21, no. 7, pp. 801–803, Jul. 1972*
20. *S. Panchanathan, "Universal architecture for matrix transposition," IEE Proc. E Comput. Digit. Tech., vol. 139,no. 5, pp. 387–392, Sep. 1992.*
21. *M. R. Portnoff, "An efficient method for transposing large matrices and its application to separable processing of two-dimensional signals," IEEE Trans. Image Process., vol. 2, no. 1, pp. 122–124, Jan. 1993*