A NOVEL PDWC-UCO ALGORITHM BASED OPTIMAL PLACEMENT OF BUFFER IN FPGA ARCHITECTURE

A. SRIDEVI[1*], DR.V.LAKSHMIPRABHA[2]

[1]Associate Professor, Dept. of ECE, SNS College of Technology,Coimbatore-641 035,Tamilnadu, India, Phone No-0422 – 2666264.

[2]Principal, Government College of Technology, Coimbatore, Tamil Nadu, India.

[*]Corresponding author

Abstract**:** *Clock distribution networks consume a significant amount of the whole chip power budget. Therefore, reduction in the power consumption of the clock networks is a significant objective in high-performance Integrated Circuit (IC) designs. This paper presents a novel Particle Distance Weighted Clustering (PDWC)-Unity Clustering Optimization (UCO) algorithm for the optimal placement of clock buffers in the Field Programmable Gate Array (FPGA) architecture. A novel PDWC algorithm is applied for clustering the logical components based on the minimum distance between components. A UCO algorithm is developed to determine the optimal location for the placement of the buffers. This clustering technique reduces the delay rate of the architecture due to the minimum number of logical components. The overall area and power consumption of the FPGA architecture are reduced due to the optimal placement of the buffers and latches. Our proposed PDWC-UCO algorithm achieves lower delay, power consumption, wire length, latency and skew than the existing Flip-Flop (FF) merging and register clustering algorithms.*

Index Terms*—Buffer Placement, Clustering, Field Programmable Gate Array (FPGA), Particle Distance Weighted Clustering (PDWC), Unity Clustering Optimization (UCO)*

1. Introduction

In Very Large Scale Integration (VLSI) applications, reduction in the power dissipation has become a crucial concern in the design of modern IC, especially for the portable devices with strict requirements on power density and battery life [1]. Minimization of the clock power is the significant objective in the development of modern system-on-chips (SoC) [2] to reduce the overall power consumption. There is a need to find novel solutions to reduce power consumption of the clock distribution network and facilitate the improvement in the chip functionality. Various design methodologies such as buffer sizing [3], [4], clock gating [5], register clustering [6], [7], banking [8] and Multi-bit FF (MBFF) replacement [9], [10], [11], [12], [13], [14], [15] are developed to reduce the power consumption of the clock generator and chip area. But, only a small number of FFs are merged due to the static combinational logic cells. Also, the reduction of the wire length and clock sink is limited.

1.1 Motivation

Substantial placement of the buffers results in the increase in the power consumption. Thus, optimal placement of the buffer is a critical task in the design of clock distribution network in VLSI designs [16]. Existing buffer placement algorithms [17] and [18] are developed to achieve a significant reduction in power consumption and wire length. In the work proposed by Reuben et al [19], the buffers are placed at the mesh nodes according to the density of the clock sinks in its vicinity and the proximity of the buffer to the sinks. The clustering-based buffer placement algorithm [16] is proposed to find a group of the clock sinks and assign a single buffer to the cluster. But, these algorithms are highly inefficient for high loads. Also, the existing buffer placement algorithms focus on the reduction of skew with a minimum number of buffers.

In the FPGA architecture, reduction in the size of the components results in the minimization of the overall area, delay and power consumption. K-Map based Boolean expression method and block optimization methods are developed to reduce the size of the FPGA architecture. These are mainly used in the data compression, encoding, and decoding process. Previously, Sridevi and Dr. V. Lakshmiprabha presented clock distribution using H-Tree based design, coconut-tree design [20] and mixed-tree based clock distribution [21] that replaces the parallel register allocation. By using the optimal placement of buffers and latches, the number of register usage is reduced. However, this may lead to the delay incurred by the logical process at each stage of clusters. Hence, there is a need for a novel algorithm for the optimal placement of buffers to reduce the power consumption and wire length of the FPGA architecture.

1.2 Our proposed work

In our proposed work, an enhanced architecture is implemented in the form of optimal clustering technique to overcome the above limitations. This paper presents the optimal replacement of buffers and latches through the optimized distribution of the clusters. Thus, the number of register usage in the FPGA architecture is reduced. This results in the effective management of load across the clock generator. A PDWC algorithm is developed for grouping the logical components into clusters. A UCO algorithm is presented to determine the optimal location for the placement of the buffers. This enables the best placement of the buffers and latches to provide clock distribution to the FPGA architecture. This type of clustering technique reduces the delay rate of the architecture due to the minimum number of logical components. During each clustering condition, the optimal placement of the register enables pairing of the blocks at each stage. Our proposed PDWC-UCO algorithm achieves minimum delay, power

consumption, wire length, latency and skew than the existing FF merging and register clustering algorithms.

1.3 Contributions of our proposed work

The main contributions of our proposed work are

- A novel clustering algorithm and an optimization algorithm are proposed to determine the optimal location for the placement of buffers.
- Optimal placement of the buffers ensures uniform clock distribution to the FPGA architecture, while reducing the wire length.
- The load across the clock generator is managed efficiently.
- Due to the reduction in the total number of buffers and latches, the area and power consumption of the FPGA architecture are reduced.

1.4 Organization

The remaining sections of the paper are unfolded as follows: Section II describes the conventional research works related to the FF merging, register clustering, and clock power optimization techniques. Section III explains the proposed work including FF merging, PDWC algorithm, and UCO algorithm in detail. Section IV illustrates the performance analysis including the wire length, delay and channel width analysis of each benchmark circuit. Section V presents the comparative analysis of the proposed work with the existing FF merging techniques and register clustering algorithms. Section VI illustrates the conclusion and future implementation of the proposed work.

2. Related Works

This section describes the conventional research works related to the FF merging, register clustering, and clock power optimization techniques.

2.1 FF merging and register clustering

Deng et al. [1] formulated a buffer allocation algorithm and register clustering algorithms for satisfying the slew constraints within the clusters. The register clustering algorithms achieved a significant reduction in the power consumption without affecting the skew and maximum latency of the clock tree. Jiang et al. [12] focused on post-placement MBFF clustering and identified necessary partial sequences for FF clustering. Efficient power saving was achieved during post-placement even under timing and placement density constraints. Shyu et al. [13] applied coordinate transformation for identifying the FFs that can be merged and illustrated the creation of a combination table to compute the possible combinations of the FFs. A hierarchical approach was applied for merging FFs. The total wire length was minimized and time complexity was educed, along with the reduction in the clock power. Liu et al. [14] developed an FF merging algorithm based on agglomerative clustering and evaluated the power consumption of the clock tree after FF merging. A framework to determine a preferable location for relocated merged FFs was proposed. The wire length of the clock tree was reduced with minimum power consumption level. Tsai et al. [15] introduced MBFF bonding at placement for reducing the clock power. The proposed approach achieved a significant reduction in the clock power than the post-placement MBFF merging approach.

Deng et al. [22] devised a novel methodology for clustering the registers to reduce the power consumption of the clock trees. Reduction in the power consumption and skew was achieved. Wang et al. [23] proposed an approach for merging and relocating the FFs to reduce the overall size and power consumption of the clock distribution network. Through selective merging and relocation of the FFs based on the timing and placement density limitations, the switching power of the nets connected to the FFs was controlled. The proposed approach achieved a significant reduction in the total switching capacitance of the networks. Lo et al. [24] presented a novel flow for power optimization including MBFF-aware Clock Gate (CG) cloning, FF merging, and optimization of the CG placement. The optimization flow resulted in better dynamic power and minimization of the clock wire length. Lin et al. [25] introduced a novel placement approach with clock-tree aware FF merging during placement of FFs and logic cells. The proposed approach was effective in the reduction of FF power and clock latency. Hsu et al. [26] proposed a novel approach to optimize power consumption in the clock network. The proposed approach was effective in avoiding the crosstalk in the IC while achieving power optimization.

2.2 Clock power optimization Techniques

Lin et al. [27] investigated the utilization of the pulsed-latch and proposed a migration approach for power conservation in a clock tree. The clock tree topology was determined based on the low-cost and maximum flow formulation to maintain the load balance and reduce the wire length between the pulse generators and pulsed latches. Ward et al. [28] proposed a novel latch placement methodology with lower local clock tree capacitance. The optimized placement solutions were generated for various input configurations, and a redundancy removal approach was applied to remove over 99% of the placement templates without any information loss. Finally, a decision tree induction algorithm was applied to select the template during the optimization of the clock. The proposed approach achieved an effective reduction in the clock tree capacitance and total power due to the extreme fast selection of the template. The concept of MBFF construction suffers due to the routability issue that occurs during merging of FFs. Chen et al. [29] introduced an efficient approach for merging 1-bit FFs to reduce clock power, by considering the routability constraints the proposed approach reduced the FF area and ensured a considerable amount of clock power saving.

Hyman et al. [30] presented a novel strategy for clock control and power reduction in VLSI circuits. The peak power was reduced without the need to change the operating frequency and utilize an additional power supply voltages. Riahi Alam et al. [31] proposed a centralized microarchitecture-level clock gating for low power hardware accelerators. The reduction in the power dissipation was achieved based on the number of registers and size of the circuit. Mistry et al. [32] devised a sub-clock power gating technique to reduce the leakage power during the active mode. Through the gating of the combinational logic within the sub-clock and reduction of the virtual supply, effective power reduction was achieved. Teng and Anderson [33] explored the usage of pulsed latches to

replace the FFs for timing optimization in FPGA. The latch-based optimization achieved better performance without the increase in the overall area. Jawahar et al. [34] proposed a novel Self-Gated Resonant (SGR) clocked FF to reduce the dynamic power dissipation. The timing performance of the FF was improved. Shelar [35] introduced a clustering algorithm to minimize the power consumption in a local clock tree. The algorithm achieved 14% reduction of the clock tree power while satisfying the skew or slew constraints.

Applying MBFF replacement for the reduction of the clock power in modern nanometer ICs is a promising technique. Due to the closer interconnecting wires in the MBFFs, it may lead to serious crosstalk. Also, there arises a complex problem during merging of the FFs without considering the placement constraints. To overcome these issues, a novel PDWC-UCO algorithm for the optimal placement of buffers in the FPGA architecture is proposed. Our proposed achieves reduction in the wire length, area and delay rate, while ensuring uniform distribution of clock to all logical components.

3. PDWC-UCO-based Optimal Buffer Placement

This paper presents the optimal replacement of buffers and latches through the optimized distribution of the clusters. For this process, an FPGA architecture is implemented to perform clustering of the logic components. From the Netlist detail of the FPGA architecture, a total number of components usage, connectivity details, register placement information, etc. are analyzed to extract the optimal location for the clock buffer in best clusters. A PDWC algorithm is developed for clustering the logical components, and a UCO algorithm is presented to determine the optimal location for the placement of the buffers. This enables the best placement of the buffers and latches to provide clock distribution to the FPGA architecture. Thus, the number of register usage in the FPGA architecture is reduced. This results in the effective management of load across the clock generator. This type of clustering technique reduces the delay rate of the architecture due to the minimum number of logical components. During each clustering condition, the optimal placement of the register enables pairing of the blocks at each stage. The crossing of the bit sequence from input to output optimizes the delay block due to the link formation in the register. Fig.1 shows the overall flow diagram of our proposed work.

3.1 FF Merging

During clustering, the distance between the logical components is calculated. After the distance computation, the cluster index is obtained, and cluster matrix is generated. The clusters are formed in such a way that the components located closer to each other are grouped into a cluster. The distance between the clusters is estimated, and the neighboring clusters with minimum distance interval are grouped. The buffers are placed proximate to the clusters that are located close to each other. If a cluster is located away from other clusters, that cluster acts a single cluster, and a buffer is placed individually to that cluster. Fig.2 (a) shows an illustration of the clock tree with 1-bit FFs and Fig.2 (b) shows the reduced clock tree after replacing 1-bit FFs with 2-bit FFs. Fig.3 shows an illustration of the FF merging with ten FFs. Fig.3 (a) shows the initial placement of the FFs and clock routing tree before FF merging. Fig.3 (b) illustrates the mapping of the FFs and Fig.3 (c) shows the clustering of the FFs and placement of the buffers based on the minimum distance between the components. Fig.3 (d) depicts the final buffer placement with newly generated 2-bit FFs.

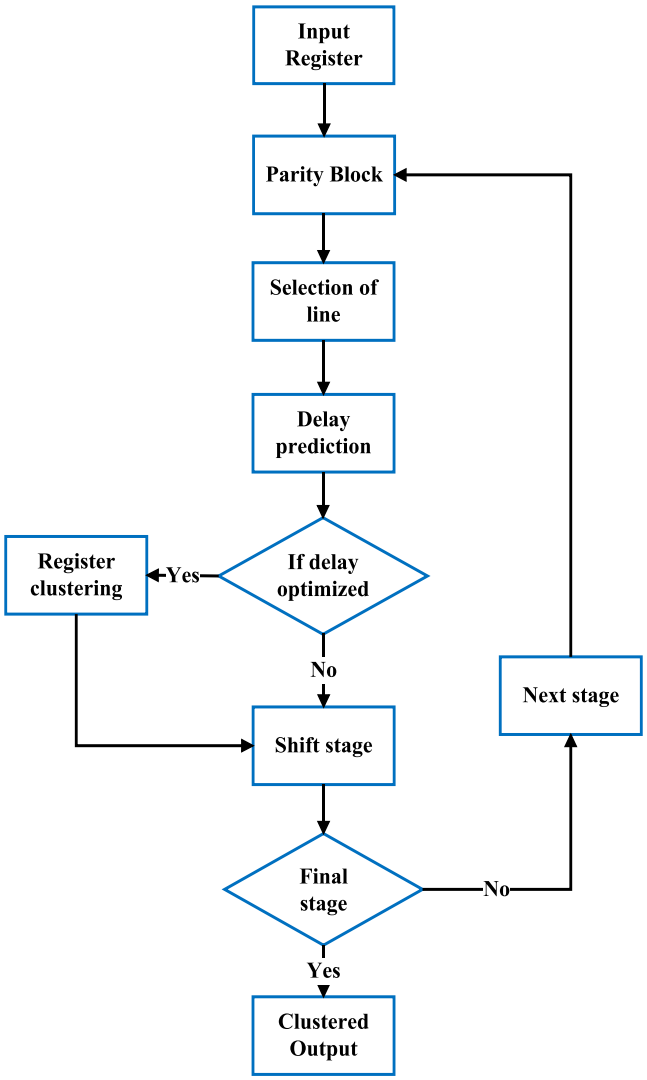


Fig.1 Overall flow diagram of our proposed work

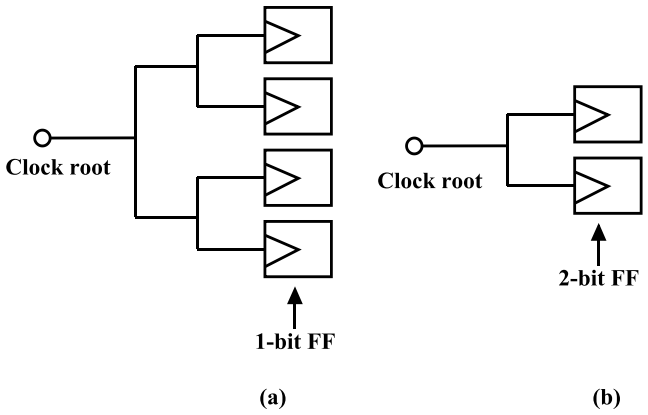
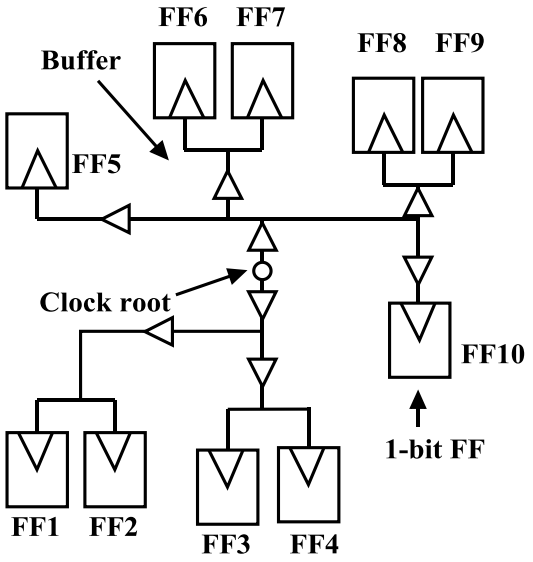
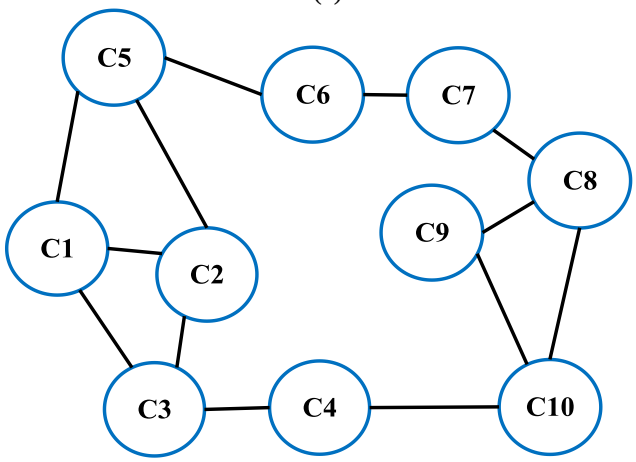


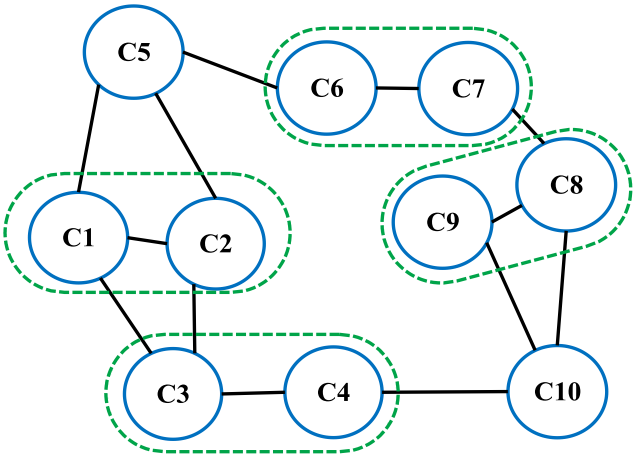
Fig.2 (a) Illustration of the clock tree with 1-bit FFs and (b) Reduced clock tree after replacing 1-bit FFs with 2-bit FFs



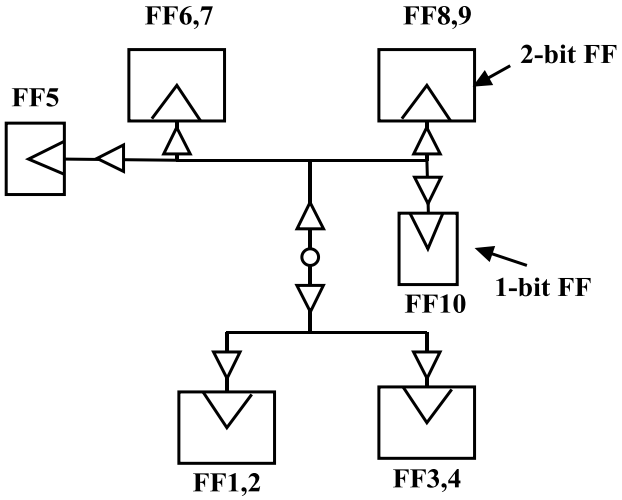
(a)



(b)



(c)



(d)

Fig.3 (a) Initial placement of the FFs and clock routing tree before FF merging (b) Mapping of the FFs (c) Clustering of the FFs and placement of buffers based on the minimum distance between the components and (d) Final buffer placement with newly generated 2-bit FFs

3.2 PDWC Algorithm

A novel PDWC algorithm is developed for clustering the logical components based on the minimum distance between each logical component in the architecture. Each cluster includes 'N' number of FFs that can handle 'N' number of bits. After clustering, the Cluster Head (CH) is selected such that the distance between the CH to the rest of the components is minimum. Then, the buffer is located at the center of the cluster to reduce the wire length. The distance between each component in the component matrix 'M' is estimated by using the following equation

$$RV = \sqrt{\sum(M_R)^2 + \sum(M_C)^2 - 2(M_R \times M_C)} \quad (1)$$

The minimum distance vector and corresponding index of the vector are estimated as given below

$$[V, idx] = Min\left(RV_{R,(1\ to\ R-1)}\right) \quad (2)$$

The cluster weight and column index are updated based on the minimum distance vector and index.

$$T(R, C) = V \quad (3)$$

$$Ct = C(C != idx) \quad (4)$$

The maximum value is obtained from the selected column index and temporary column of the matrix.

$$T(R, Ct) = Max([T(R, Ct), T(idx, Ct)]) \quad (5)$$

The transverse value of the temporary cluster is obtained as

$$T(C, R) = Y(R, C)^{-1} \quad (6)$$

The maximum transverse value at the row and column of the component matrix is defined as 'i' and 'j'. The maximum distance index is extracted based on the size of the transverse value of the cluster, 'i' and 'j'.

$$L = \{i(j), K(j)\} \quad (7)$$

The minimum value of the cluster weight is estimated as

$$[y, j] = \min(T(L, K)) \quad (8)$$

The cluster index is updated, and the matching index is also updated as an empty set. Finally, the cluster index is extracted, and cluster matrix is generated based on the index. Fig.4 shows the initial placement of the logical components and Fig.5 shows

the scattered plot of initial cluster. Fig.6 shows the cluster formation with the indexed cluster matrix.

PDWC Algorithm

Input: Component Matrix, 'M'
Output: Cluster Matrix, 'Y'
Procedure
Step 1: For (R = 2 to Row Size(Y))
Step 2: C = 1 to (R-1);
Step 3: Estimate the distance between each particle/component in given matrix using eqn (1).
Step 4: Estimate the minimum vector and corresponding index using eqn (2).
Step 5: Update Cluster weight using eqn (3).
Step 6: Update Column index using eqn (4).
Step 7: Get maximum value from selected index and temporary column of the matrix using eqn (5).
Step 8: Get the transverse value of temporary cluster using eqn (6).
Step 9: End loop 'R'
Step 10: K = size (T); N = length (T);
Step 11: [i, j] = max (T); //Maximum Value of T at Row and Column as, 'i', 'j' respectively.
Step 12: Extract the maximum distance index using eqn (7).
Step 13: Estimate the minimum value of cluster weight using eqn (8).
Step 14: L = {L, K (j)}; //Update Cluster index
Step 15: K (K==K (j)) = {}; //Update matching index as empty set.
Step 16: Repeat step 11 to 15 for R = {3, 4 … N-1}
Step 17: L = {L, K};
Step 18: RI (L (R)) = r; //Extract cluster index ∀ R = {1, 2 … N}
Step 19: Y=T (L, L);

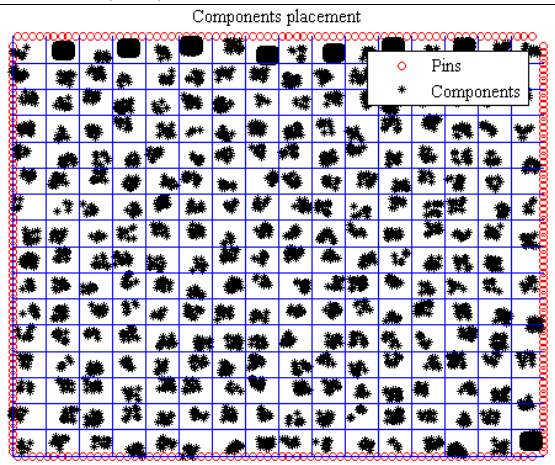


Fig.4 Initial placement of logical components

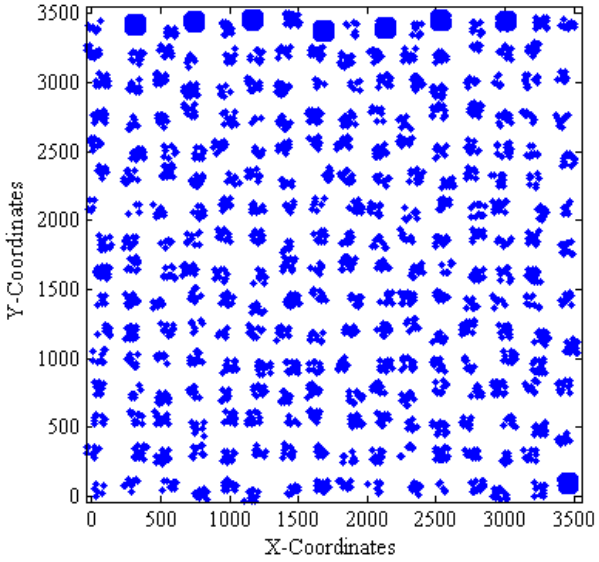


Fig.5 Scattered plot of initial cluster

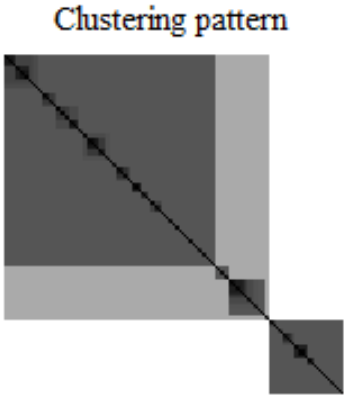


Fig.6 Cluster Formation with indexed cluster matrix

3.3 UCO Algorithm

The UCO algorithm determines the optimal location for placing the buffers in such a way to minimize the wire length between the components. The CH is selected based on the centroid of the cluster. The 'X' and 'Y' coordinates are estimated for buffer placement, and position of the centroid is updated. The 'X' and 'Y' coordinates are updated to the new position of the cluster. The clusters are updated until the iteration size is reached. Finally, the best fitness value is extracted to determine the optimal location for buffer placement.

The objective function for the cluster matrix is calculated as

$$C_i = \left(1 - \sqrt{\frac{Y_i}{1+\frac{n}{(n-1)\times\sum Y}}}\right) \times \left(1 + \frac{n}{(n-1)\times\sum Y}\right) \quad (9)$$

Where, 'n' is the size of the cluster matrix. The cost function is updated as

$$Cost_i = \min(C) - \alpha \times (\max(C) - \min(C)) \quad (10)$$

Where, α is the distance constant. The best CH is selected by using the following equation

$$H = C_i \left(\frac{Y}{\sum e^{-\frac{\beta}{N}}}\right) \quad (11)$$

Where β is the centroid constant.

The 'X' and 'Y' coordinates for buffer placement are estimated by using the equations given below

$$X_{Co-ordinate}(n) = x(n-1) + \left(\left(H^{-\frac{1}{\alpha}}\right)\right) \quad (12)$$

$$Y_{Co-ordinate}(n) = y(n-1) + \left(\left(H^{-\frac{1}{\alpha}}\right)\right) \quad (13)$$

The position of the cluster centroid is updated using the following equation

$$C = H(m) * \left(Radius * \left(Var_{high} - Var_{low}\right)\right) \quad (14)$$

Where, $Radius_{Update}(l) = (-1)^{Rand_{0,1}} \times Radius_{Pre} \times \left(\frac{2}{Radius_{Num}-1}\right) + Radius_{Pre}$

The set of CHs at the best location is determined based on the cost function and best CHs. If the position of the previous CH is better than the present CH, the previous CH is updated. Otherwise, the 'X' and 'Y' coordinates are updated to the new position. The clusters are updated until the iteration size is reached. Finally, the best fitness value is obtained for the placement of the buffers at the optimal location. This reduces the wire length and power consumption of the architecture. Fig.7 shows the fitness plot for the optimization process.

UCO Algorithm

Input: Cluster matrix 'Y'.
Output: Best Placement, 'P'.
Step 1: Initialize particles as providing clustering function, Y
Step 2: Initial objective function calculation for cluster matrix using eqn (9)
Step 3: Update cost function using eqn (10).
Step 4: Select Best Cluster head using eqn (11).
Step 5: Estimate particle placement based on 'X' and 'Y' coordinates calculated using eqns (12, 13)
Step 6: Update cluster centroid position using eqn (14)
$C(x) = \{Cost, H\}$ // C(x) is the set of cluster head at best location.
If $(C(x) > C(x+1))$
Update $C(x)$;
Update $X_{Co-ordinate}$ and $Y_{Co-ordinate}$ to New Position
End
Step 6: Update Clusters, until iteration size is reached.
Step 7: Extract Best Fitness value output $'BF'$.
Step 8: $P = Y\left(BF > avg(BF)\right)$

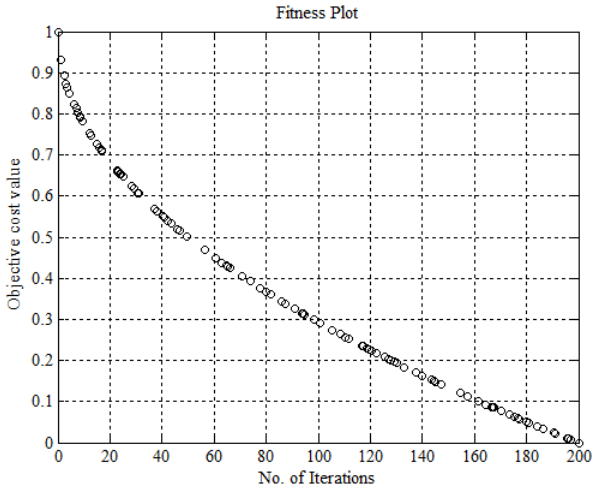


Fig.7 Fitness plot for the optimization process

4. Performance Analysis

This section presents the performance analysis of the proposed PDWC-UCO algorithm. Fig.8 shows the wire length of each benchmark circuit. Fig.9 shows the delay and Fig.10 illustrates the channel width of each benchmark circuit. The Placement Example with Known Optimal wire length (PEKO) and IBM benchmark suites [36] and [37] are used for the testing purpose in our proposed work. The PEKO suite is developed at the University of California, Los Angeles (UCLA) VLSI Computer Aided Design (CAD) laboratory. For performance analysis, ten circuits are considered from the IBM benchmark suite [37] that are derived from ISPD-98 IBM circuit benchmarks. Our proposed algorithm achieves a significant reduction in the wire length, delay in nanoseconds and channel width in nanometer.

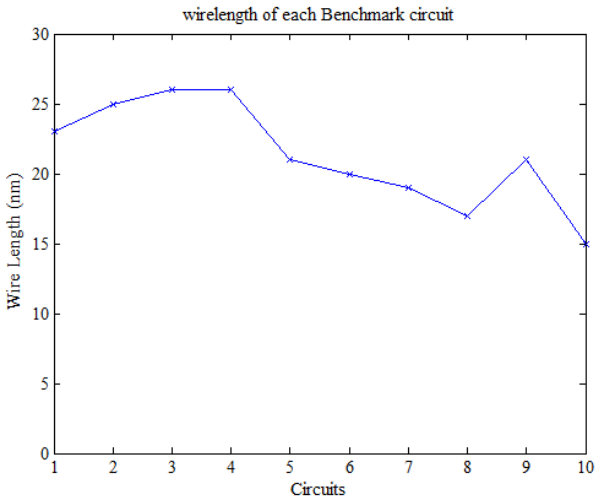


Fig.8 Wire length of each benchmark circuit

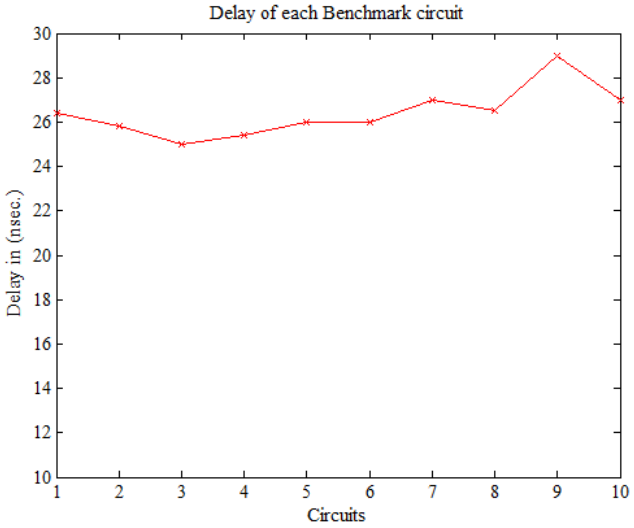


Fig.9 Delay of each benchmark circuit

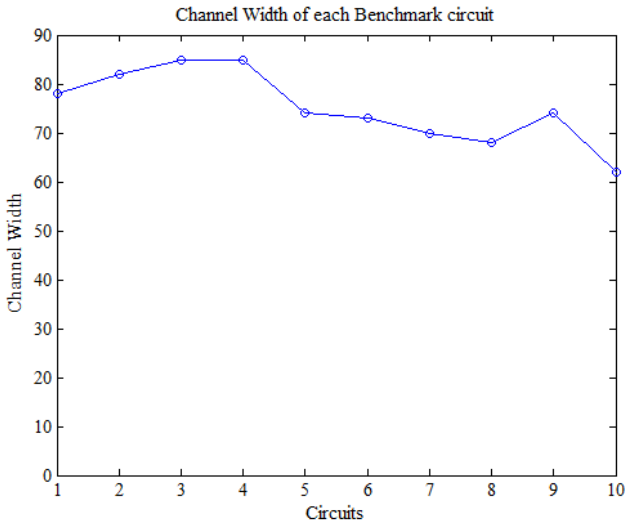


Fig.10 Channel width of each benchmark circuit

5. Comparative Analysis

This section illustrates the comparative analysis of the proposed PDWC-UCO algorithm with the traditional placement flow without applying MBFFs, clock-tree aware and clock-tree unaware Multi-Bit FF (MBFF) generation [38], post-placement MBFF generation [10], ICCAD [39] and FF-bond [15] techniques. The IWLS-2005 benchmark suite [40] is used for the comparative analysis. Table I illustrates the comparison of the FF usage for memory control (mem-ctrl) circuit including the number of 1-bit, 2-bit, 4-bit FFs, clock sinks, FF power ratio in %, Wire Length (WL) in nanometer (nm), total Clock Wire Length (CWL) in nm, clock latency ($\mathbf{D_{clk}}$) in picoseconds, Worst Negative Slack (WNS) in nanoseconds (ns) and Total Negative Slack (TNS) in ns. Due to the placement of the buffers and latches in the optimal location, our proposed PDWC-UCO algorithm achieves a lower number of 1-bit, 2-bit, 4-bit FFs, FF power ratio, WL, CWL, clock latency and higher WNS and TNS than the existing techniques.

Table II illustrates the comparative analysis of the benchmark circuits for the CLK-Adjustable Delay Buffer (ADB) continuous delay [41], ADB-pullup, ADB-pullup-Buffer Sizing (BS) [42] and proposed PDWC-ACO algorithm. The ISCAS'95 [42], ITC'99 benchmarks [43] and ISPD'09 [37] are used for the comparative analysis. The ADB-pullup uses consistently less number of ADBs when compared to the CLK-ADB. The ADB-pullup-BS requires less number of ADBs than the ADB-pullup and CLK-ADB. The proposed PDWC-UCO algorithm requires lower area and ADBs than the CLK-ADB, ADB-pullup, and ADB-pullup-BS. Table III shows the comparison of the benchmark circuits for the proposed PDWC-UCO algorithm and CLK-ADB-RD, ADB-pullup-Q and ADB-pullup-QBS [42]. The ADB-pullup-Q requires less number of ADBs than the CLK-ADS-RD and ADB-pullup-QBS uses a fewer number of ADBs than the ADB-pullup-Q. The proposed PDWC-UCO algorithm requires less number of ADBs and lower area than the CLK-ADS-RD, ADB-pullup-Q, and ADB-pullup-QBS. Due to the clustering of the logical components and optimal placement of buffers, our proposed work requires a fewer number of ADBs and less area.

Table I Comparison of FF usage for mem-ctrl circuit

Methods	Number of 1-bit FFs	Number of 2-bit FFs	Number of 4-bit FFs	Number of clock sinks	FF Pwr Ratio (%)	WL × 10^9 (nm)	CWL × 10^7 (nm)	D_{clk} × 10^5 (ps)	WNS (ns)	TNS (ns)
No MBFF Gen.	1083	0	0	1083	100	1.11	6.89	13.35	7.86	1756.8
Post Placement MBFF Gen.	255	352	31	638	88.38	1.19	5.17	8.41	7.78	1749.5
FF-Bond	3	78	231	312	79.21	1.33	3.44	6.52	8.64	1919
ICCAD	15	22	256	293	78.63	1.29	3.09	4.89	8.52	1845.1
Clock-Tree-Unaware MBFF Gen.	1	7	267	275	78.12	1.33	2.93	5.44	8.77	1866.3

Clock-Tree-aware MBFF Gen.	3	6	267	276	78.15	1.34	2.91	3.95	8.88	1950.4
PDWC-UCO	2	4	274	264	77.82	1.27	2.79	3.82	9.08	1992.5

Table II Comparison of Benchmark circuits in Continuous delay

Benchmark Circuit	FFs/Buffs	Skew/ Latency (ps)	Skew bound (ps)	CLK-ADB		ADB-Pullup		ADB-Pullup-BS		PDWC-UCO	
				ADBs	Area	ADBs	Area	ADBs	Area	ADBs	Area
s35932	1728/97	264.1/545.1	30	27	156.1	25	151.5	20	135.4	19	119.3
			40	25	147	23	140	19	126.9	17	113.8
			50	25	144.5	23	137.8	19	124.7	17	111.6
s38417	1564/89	387.1/612.1	30	31	212.1	27	196.1	22	180.6	20	165.1
			40	28	197.9	25	184.6	20	169	19	153.4
			50	26	186.5	23	173.1	18	164.4	18	155.7
s38584	1168/66	299.8/552.8	30	22	138.3	20	127.3	13	123.2	12	119.1
			40	18	118.9	16	107.3	11	107.4	11	107.3
			50	18	118.8	16	105.7	11	104.6	10	103.5

Table III Comparison of Benchmark Circuit in discrete delay

Benchmark Circuit	FFs/Buffs	Skew/ Latency (ps)	Skew bound (ps)	CLK-ADB-RD		ADB-Pullup-Q		ADB-Pullup-QBS		PDWC-UCO	
				ADBs	Area	ADBs	Area	ADBs	Area	ADBs	Area
s35932	1728/97	264.1/545.1	30	42	228.1	29	171.5	25	158.7	23	145.9
			40	26	151.7	24	146	19	129.1	18	112.2
			50	25	145.2	23	139.1	19	125.9	19	112.7
s38417	1564/89	387.1/612.1	30	36	235.5	28	202.3	23	186.5	22	170.7
			40	31	211.8	27	195.4	20	171.3	20	147.2
			50	29	200.8	25	183.9	18	168.1	18	152.3
s38584	1168/66	299.8/552.8	30	22	138.2	21	132.8	17	133.8	15	132.8
			40	21	133.4	20	126.5	16	125.1	14	123.7
			50	18	118.9	16	106.3	11	105.5	10	104.7

The proposed PDWC-UCO algorithm is compared with Clock Tree Synthesis (CTS) [44], Shelar's register clustering algorithm [45], K-means based Register clustering algorithm (KMR), K-splitting based Register clustering algorithm (KSR) and Greedy Search based Register clustering algorithm (GSR) [1] by using the ISPD 2009 [46] and ISPD 2010 [47] benchmark circuits. Table III shows the comparison of results on the CTS flow with and without Register Clustering Algorithms on ISPD 2010 Benchmarks (10f01) for 1107 registers. The benchmarks are obtained from the ISPD-98 IBM circuit benchmarks. The netlists are converted into the Bookshelf placement format, and the placement-related information is added. Our novel PDWC algorithm enables efficient clustering of the logical components based on the minimum distance between the components. Hence, our proposed PDWC-UCO algorithm achieves a significant power reduction than the existing algorithms. Fig.11 shows the skew analysis for the proposed PDWC-UCO and existing clustering algorithms. Our proposed PDWC-UCO algorithm achieves minimum skew than the existing algorithms. Fig.12 illustrates the power analysis and Fig.13 maximum latency plot for the proposed PDWC-UCO and existing clustering algorithms. The proposed algorithm requires lower power consumption and latency than the existing algorithms.

Table III Comparison of Results on CTS Flow with and without Register Clustering Algorithms on ISPD 2010 Benchmarks (10f01) for 1107 Registers

Methods	Skew (ps)	Power (fF)	Max. Latency (ps)
CTS	45.22	194024	798.97
Shelar	79.19	151773	791.25
KMR	30.67	153885	756.28
KSR	46.81	146083	779.67
GSR	34.67	142847	763.59
PDWC-UCO	30.128	135217	754.48

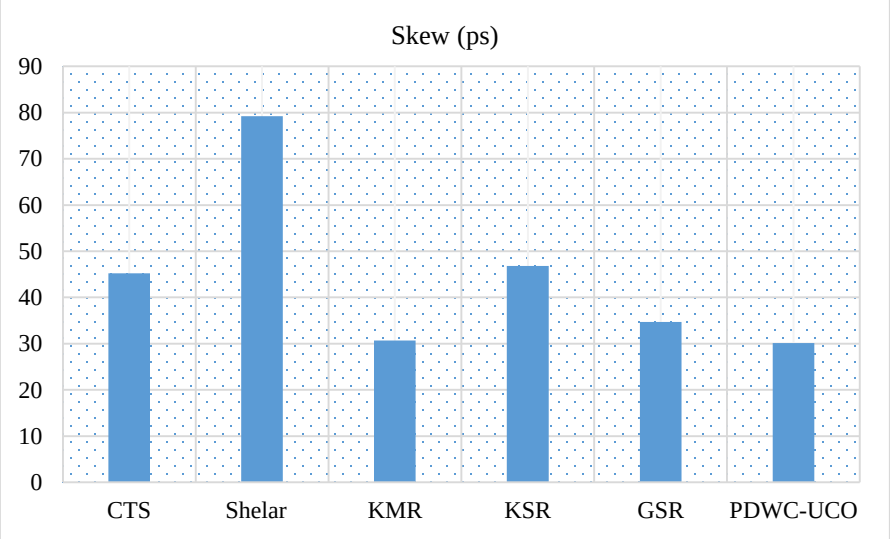


Fig.11 Skew analysis for the proposed PDWC-UCO and existing clustering algorithms

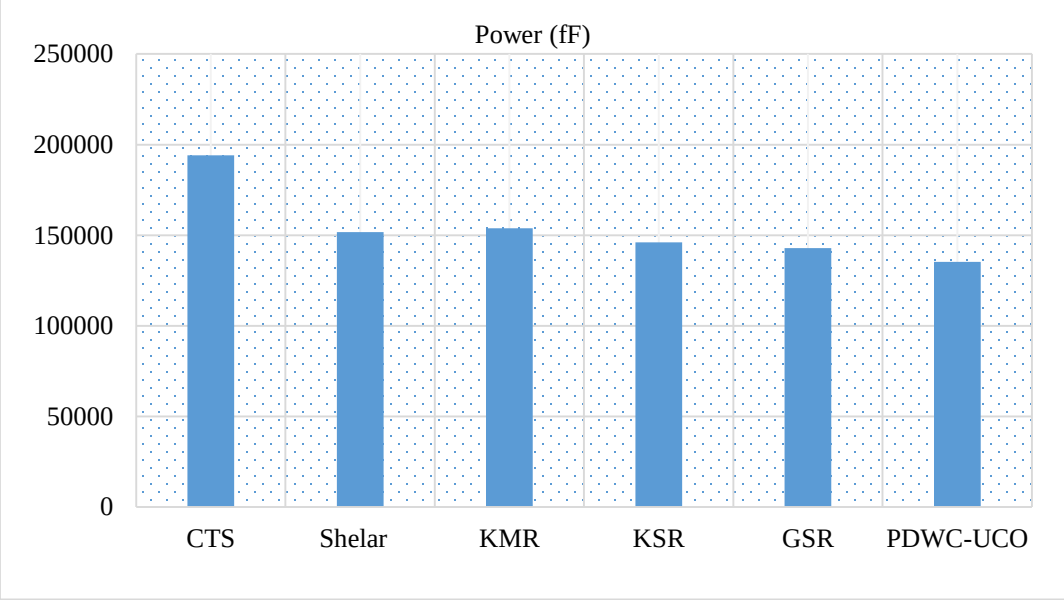


Fig.12 Power analysis for the proposed PDWC-UCO and existing clustering algorithms

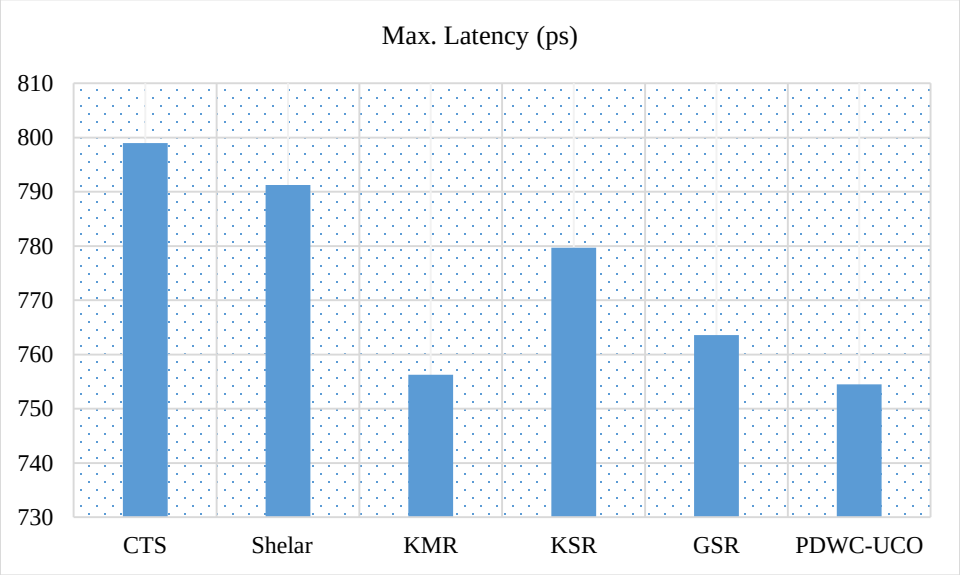


Fig.13 Maximum Latency plot for the proposed PDWC-UCO and existing clustering algorithms

Table IV shows the comparison of the results on CTS flow with and without register clustering algorithms on ISPD 2010 Benchmarks (10f02) for 2249 Registers. Our proposed PDWC-UCO algorithm achieves lower skew, power, and latency than the existing algorithms.

Table IV Comparison of Results on CTS Flow with and without Register Clustering Algorithms on ISPD 2010 Benchmarks (10f02) for 2249 Registers

Methods	Skew (ps)	Power (fF)	Max. Latency (ps)
CTS	36.62	399411	941.17
Shelar	195.11	295862	1088.21
KMR	37.77	287330	901.55
KSR	47.19	272822	924.48
GSR	50.15	267875	963.18
PDWC-UCO	37.13	226818	901.05

6. Conclusion and Future Work

This paper presented a novel PDWC-UCO algorithm for the clustering of logical components and placement of the buffers in the optimal locations. This enables the best placement of the buffers and latches to provide equalized clock distribution to the FPGA architecture. This results in the effective management of load across the clock generator and a significant reduction in the number of register usage in the architecture. This type of clustering technique reduces the delay rate, size and power consumption of the architecture due to the minimum number of logical components. During each clustering condition, the optimal placement of the register enables pairing of the blocks at each stage. Our proposed work achieves an effective reduction in the overall wire length of the FPGA architecture. From the performance and comparative analysis, it is observed that the proposed PDWC-UCO algorithm achieves minimum delay, power consumption, wire length, latency and skew than the existing FF merging and register clustering algorithms. In future, we implement this optimal placement of buffers and clustering of FFs in the application of various FPGA architectures to reduce the wire length, delay, power and area consumption.

References

[1] C. Deng, Y.-C. Cai, and Q. Zhou, "Register Clustering Methodology for Low Power Clock Tree Synthesis," *Journal of Computer Science and Technology*, vol. 30, pp. 391-403, 2015.

[2] M. P.-H. Lin, C.-C. Hsu, and Y.-C. Chen, "Clock-Tree Aware Multibit Flip-Flop Generation During Placement for Power Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 34, pp. 280-292, 2015.

[3] K. Wang and M. Marek-Sadowska, "Buffer sizing for clock power minimization subject to general skew constraints," in *Proceedings of the 41st annual Design Automation Conference*, 2004, pp. 159-164.

[4] J. G. Xi and W. W. Dai, "Buffer insertion and sizing under process variations for low power clock distribution," in *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference*, 1995, pp. 491-496.

[5] M. Donno, A. Ivaldi, L. Benini, and E. Macii, "Clock-tree power optimization based on RTL clock-gating," in *Proceedings Design Automation Conference*, 2003, pp. 622-627.

[6] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, and Q. Wang, "Power-aware placement," in *Proceedings of the 42nd annual Design Automation Conference*, 2005, pp. 795-800.

[7] Y. Lu, C. Sze, X. Hong, Q. Zhou, Y. Cai, L. Huang, *et al.*, "Navigating registers in placement for clock network minimization," in *Proceedings. 42nd Design Automation Conference*, 2005, pp. 176-181.

[8] W. Hou, D. Liu, and P.-H. Ho, "Automatic register banking for low-power clock trees," 2009.

[9] Z.-W. Chen and J.-T. Yan, "Routability-constrained multi-bit flip-flop construction for clock power reduction," *Integration, the VLSI Journal*, vol. 46, pp. 290-300, 2013.

[10] M. P.-H. Lin, C.-C. Hsu, and Y.-T. Chang, "Post-placement power optimization with multi-bit flip-flops," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 30, pp. 1870-1882, 2011.

[11] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, and W.-K. Mak, "Power-driven flip-flop merging and relocation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 31, pp. 180-191, 2012.

[12] I. H.-R. Jiang, C.-L. Chang, and Y.-M. Yang, "INTEGRA: Fast multibit flip-flop clustering for clock power saving," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 31, pp. 192-204, 2012.

[13] Y.-T. Shyu, J.-M. Lin, C.-P. Huang, C.-W. Lin, Y.-Z. Lin, and S.-J. Chang, "Effective and efficient approach for power reduction by using multi-bit flip-flops," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. 21, pp. 624-635, 2013.

[14] S. S.-Y. Liu, W.-T. Lo, C.-J. Lee, and H.-M. Chen, "Agglomerative-based flip-flop merging and relocation for signal wirelength and clock tree optimization," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, p. 40, 2013.

[15] C.-C. Tsai, Y. Shi, G. Luo, and I. H.-R. Jiang, "FF-bond: multi-bit flip-flop bonding at placement," in *Proceedings of the 2013 ACM international symposium on International symposium on physical design*, 2013, pp. 147-153.

[16] J. Reuben, H. M. Kittur, and M. Shoaib, "A buffer placement algorithm to overcome short-circuit power dissipation in mesh based clock distribution network," *An International Journal Engineering Science and Technology* vol. 18, pp. 135-140, 2015.

[17] A. Rajaram and D. Z. Pan, "Meshworks: a comprehensive framework for optimized clock mesh network synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 29, pp. 1945-1958, 2010.

[18] M. R. Guthaus, X. Hu, G. Wilke, G. Flach, and R. Reis, "High-performance clock mesh optimization," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 17, p. 33, 2012.

[19] J. Reuben, M. Zackriya. V, S. Nashit, and H. M. Kittur, "Capacitance driven clock mesh synthesis to minimize skew and power dissipation," *IEICE Electronics Express*, vol. 10, pp. 20130850-20130850, 2013.

[20] A. Sridevi and D. V. Lakshmiprabha, "A Novel Coconut-Tree based Clock Distribution Network for High-Speed FFT architecture," *Australian Journal of Basic and Applied Sciences*, vol. 9, pp. 69-78, 2015.

[21] A.Sridevi and D. V. L. Prabha, "Firefly Optimization and Mixed Tree Based Clock Distribution Network for an Optimal Energy FFT Architecture," *International Journal of Applied Engineering Research*, vol. 10, pp. 38016-38024, 2015.

[22] C. Deng, Y. Cai, and Q. Zhou, "Fast synthesis of low power clock trees based on register clustering," in *16th International Symposium on Quality Electronic Design (ISQED)*, 2015, pp. 303-309.

[23] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, and W.-K. Mak, "Power-driven flip-flop merging and relocation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, pp. 180-191, 2012.

[24] S.-C. Lo, C.-C. Hsu, and M. P.-H. Lin, "Power optimization for clock network with clock gate cloning and flip-flop merging," in *Proceedings of the 2014 on International symposium on physical design*, 2014, pp. 77-84.

[25] M. P.-H. Lin, C.-C. Hsu, and Y.-C. Chen, "Clock-Tree Aware Multibit Flip-Flop Generation During Placement for Power Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 34, pp. 280-292, 2015.

[26] C.-C. Hsu, M. P.-H. Lin, and Y.-T. Chang, "Crosstalk-aware multi-bit flip-flop generation for power optimization," *Integration, the VLSI Journal*, vol. 48, pp. 146-157, 2015.

[27] H.-T. Lin, Y.-L. Chuang, Z.-H. Yang, and T.-Y. Ho, "Pulsed-latch utilization for clock-tree power optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. 22, pp. 721-733, 2014.

[28] S. I. Ward, N. Viswanathan, N. Y. Zhou, C. C. Sze, Z. Li, C. J. Alpert, *et al.*, "Clock power minimization using structured latch templates and decision tree induction," in *Proceedings of the International Conference on Computer-Aided Design*, 2013, pp. 599-606.

[29] Z.-W. Chen and J.-T. Yan, "Routability-constrained multi-bit flip-flop construction for clock power reduction," *the VLSI Journal Integration,*, vol. 46, pp. 290-300, 2013.

[30] R. Hyman, N. Ranganathan, T. Bingel, and D. Tran Vo, "A clock control strategy for peak power and RMS current reduction using path clustering," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. 21, pp. 259-269, 2013.

[31] M. Riahi Alam, M. Ersali Salehi Nasab, and S. M. Fakhraie, "Power Efficient High-Level Synthesis by Centralized and Fine-Grained Clock Gating," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 34, pp. 1954-1963, 2015.

[32] J. N. Mistry, J. Myers, B. M. Al-Hashimi, D. Flynn, J. Biggs, and G. V. Merrett, "Active Mode Subclock Power Gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. 22, pp. 1898-1908, 2014.

[33] B. Teng and J. H. Anderson, "Latch-based performance optimization for field-programmable gate arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 32, pp. 667-680, 2013.

[34] J. J. D. Jawahar, S. M. S. Murthy, and K. B. V. Somasundaram, "Self-gated resonant-clocked flip-flop optimised for power efficiency and signal integrity," *IET Circuits, Devices & Systems*, vol. 10, pp. 94-103, 2016.

[35] R. S. Shelar, "A fast and near-optimal clustering algorithm for low-power clock tree synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 31, pp. 1781-1786, 2012.

[36] C.-C. C. a. M. Xie. (2002-2003). *PEKO Suite*. Available: http://cadlab.cs.ucla.edu/~pubbench/placement/dw.htm

[37] S. Adya and I. Markov. *ISPD02 IBM-MS Mixed-size Placement Benchmarks*. Available: http://vlsicad.eecs.umich.edu/BK/ISPD02bench/

[38] M. P.-H. Lin, C.-C. Hsu, and Y.-C. Chen, "Clock-Tree Aware Multibit Flip-Flop Generation During Placement for Power Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 280-292, 2015.

[39] C.-C. Hsu, Y.-C. Chen, and M. P.-H. Lin, "In-placement clock-tree aware multi-bit flip-flop generation for power optimization," in *Proceedings of the International Conference on Computer-Aided Design*, 2013, pp. 592-598.

[40] (2005). *IWLS 2005 Benchmarks*. Available: http://iwls.org/iwls2005/benchmarks.html

[41] K.-H. Lim and T. Kim, "An optimal algorithm for allocation, placement, and delay assignment of adjustable delay buffers for clock skew minimization in multi-voltage mode designs," in *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, 2011, pp. 503-508.

[42] J. Kim, D. Joo, and T. Kim, "Optimal utilization of adjustable delay clock buffers for timing correction in designs with multiple power modes," *the VLSI Journal Integration,*, vol. 52, pp. 91-101, 2016.

[43] S. Davidson. *ITC'99 Benchmark*. Available: http://www.cerc.utexas.edu/itc99-benchmarks/bench.html

[44] F. Niu, Q. Zhou, H. Yao, Y. Cai, J. Yang, and C. N. Sze, "Obstacle-avoiding and slew-constrained buffered clock tree synthesis for skew optimization," in *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*, 2011, pp. 199-204.

[45] R. S. Shelar, "An efficent clustering algorithm for low power clock tree synthesis," in *Proceedings of the 2007 international symposium on Physical design*, 2007, pp. 181-188.

[46] I. C. o. E. D. A. (CEDA). (2009). *International Symposium on Physical Design (ISPD) 2009 Clock Network Synthesis Contest*. Available: http://ispd.cc/contests/09/ispd09cts.html

[47] A. S. a. I. Corporation. (2010). *International Symposium on Physical Design (ISPD) 2010 High-performance Clock Network Synthesis Contest*. Available: http://archive.sigda.org/ispd/contests/10/ispd10cns.html