Retraction

Retraction: A Comparative Study on minimum skew Clock tree distribution algorithms for high-speed Digital Integrated Circuits (J. Phys.: Conf. Ser. 1916 012125)

Published 23 February 2022

This article (and all articles in the proceedings volume relating to the same conference) has been retracted by IOP Publishing following an extensive investigation in line with the COPE guidelines. This investigation has uncovered evidence of systematic manipulation of the publication process and considerable citation manipulation.

IOP Publishing respectfully requests that readers consider all work within this volume potentially unreliable, as the volume has not been through a credible peer review process.

IOP Publishing regrets that our usual quality checks did not identify these issues before publication, and have since put additional measures in place to try to prevent these issues from reoccurring. IOP Publishing wishes to credit anonymous whistleblowers and the Problematic Paper Screener [1] for bringing some of the above issues to our attention, prompting us to investigate further.

[1] Cabanac G, Labbé C and Magazinov A 2021 arXiv:2107.06751v1

Retraction published: 23 February 2022
A Comparative Study on minimum skew Clock tree distribution algorithms for high-speed Digital Integrated Circuits

Dominic Mathew¹, Sophia S¹ and Kiran Kumar Pasupuleti²
¹Department of Electronics and Communication, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.
²LeadSoc Technologies, Bengaluru, Karnataka, India.
Email: 19epae001@skcet.ac.in

Abstract. In this paper an extensive analysis on clock tree synthesis algorithms are carried out with conventional cts, Mesh based cts, Meshless Multisource and flexible H-tree multisource cts with mesh. The algorithms are analysed with industry standard placement benchmarks. Latency, clock power, wirelength, number of buffers, and clock skew are observed. The clock distribution algorithm which provides a single digit least skew is picked, the pros and cons of that algorithm is elaborately discussed.

1. Introduction
Application Specific Integrated Circuits have an enormous number of applications in this digital era. Clock tree synthesis performs an essential aspect in the implementation of high speed digital integrated circuits. Clock tree distribution is preeminent in deciding the excellence of outcome, where the floor planned, power planned and globally placed design is being routed for clock nets only to avoid the timing violations [1-4].

A common question crop up in the clock tree synthesis phase is what is outstanding clock tree distribution algorithm and development approaches to be done during which results is a minimum skew clock tree network with optimal power and latency.

In this study a comprehensive analysis is made on various benchmarks with cadence 45nm Process development kit under clock tree synthesis stage [5-8].

2. Benchmarks

| Snapshot | Density (%) | Total instances | FFs | Total area (um^2) |
|----------|-------------|-----------------|-----|------------------|
| B19      | 73.47       | 64173           | 5,513 | 137468          |
| JPEG     | 70.19       | 204891          | 37,411 | 573034         |
| VGA      | 73.41       | 31501           | 17,051 | 155976         |
| Leon3mp  | 58.11       | 363451          | 108,721 | 1337844     |
| VGAblock | 64.64       | 31501           | 17,051 | 155976         |
| VGAar    | 73.41       | 31501           | 17,051 | 155976         |
| RAM      | 75.02       | 35020           | 29,128 | 1419855       |
In this study 7 benchmarks are taken into consideration in table 1. The benchmarks are obtained from the various sources like ISPD2012, IWS2005 and open cores in which leon3mp is the biggest of all followed by jpeg, b19 and VGA. The RTls of the benchmarks are synthesized without any unsolved references and lint errors [9-11]. All the designs are synthesis in single mode single corner format with an operating voltage of 1.1 v. All the designs are constrained with 1ns time period and 60ps maximum transition. The synthesized netlist, physical libraries, timing libraries, parasitic information and constraints are given as the input for placement and route tool in which the global power and ground nets are defined. Leon3mp, JPEG, B19 and RAM are floorplanned with 1:1 aspect ratio.

Whereas VGA design is floorplanned in three different fashion. VgaBlockage is floorplanned with two placement blockages as shown in the Figure 1 at the top right corner and the mid of the bottom edge, this purposefully done to check the adaptability of the flexible H-tree and skew offered by the technique. VgaAR is floorplanned with 1:2 aspect ratio and one more design VGA is floorplanned with 1:1 ratio like the other four. After floorplanning the power & ground rings, strips and rails are routed. The design is globally placed and optimized, the post placement checks are performed and proceeded for clock tree synthesis.

3. Methodology
Four different clock tree synthesis algorithms are taken into consideration for the study. All the clock tree distribution techniques specification are loaded prior to place and route tool. All the benchmarks are provided with a specification of 25ps target skew, 60ps maximum transition, metal layer 7-6 is assigned for top tree, metal layer 6-5 is assigned for trunk and metal layer 5-4 is assigned for leaf. And most importantly all the clock routes are shielded both the sides with VSS net to avoid signal integrity issues.

3.1. Conventional CTS
This is the most basic clock tree synthesis algorithm used in a physical design flow in figure 2. In this algorithm the clock source is routed to the clock sink pins in limitless levels. The main objective of this clock tree algorithm is to reduce the clock signal transition delay and to minimize the fanout and output load. The long nets are divided into many subsections to minimize the resistance and capacitance of the
nets which successively curtail the transition delay. The destination paths are shared by using clock buffers in multiple levels in order to avoid a single net driving too many sinks which again leads to more transition delay.

3.2. Clock Mesh
In a clock mesh based clock tree distribution algorithm a clock grid is designed in the clock trunk region mostly with metal layer 6. The top tree is routed to the mesh and the mesh has a matrix of clock buffer drivers. The mesh grid size and buffer drivers are directly proportional to the volume of the design and the estimate of sequential instances.

In this the smaller benchmarks such as VGA and RAM are provided with 32*32 clock buffers whereas in bigger designs such as JPEG and B19 40*40 buffer drivers are used. Leon3mp being the biggest of all the benchmarks requires 64*64 buffer matrix mesh drivers in order meet the clock skew requirement.

3.3. Flexible-Htree

Figure 3. Layout of Flexible-Htree on Vga, VgaAr and VgaBlockage

In this algorithm the top tree is in adaptable h-tree fashion where the endpoints of the top tree are balanced to have the same insertion delay. This algorithm is a mesh less multi-tap cts technique where the endpoints of the top h-Tree is tapped to the sinks directly in figure 3.

Figure 4. Layout of various sections of adaptable H-tree of VgaBlockage
The adaptability of the H-tree can be very much observed in the designs with blockages. The H-tree modifies its net structure according to the presence of the placement and route blockages which is very much observed in the benchmark VgaBlockage is shown in the Figure 4. The advantage of this design is it used less routing resources due to the absence of the mesh drivers.

3.4. Flexible-Htree Multi-Source CTS

![Figure 5. Layout of Flexible-Htree Multi-Source CTS on Leon3mp, VgaAr and JPEG](image)

This is the most advanced algorithm in which the top tree flexible H-tree and the trunk is mesh structure. Matrix of mesh drivers are assigned for each design as same as the clock buffer assignment in the mesh based clock tree technique. In the vacant region where the standard cells are not placed were excluded from the buffer placement which can be seen in the Figure 5-7. This algorithm takes too much routing resources and consumes more power.

4. Results

4.1. Max Latency

The maximum latency of the design is the maximum delay taken for the clock signal to reflect the transition of the signal from the source clock pin of design to the sink pins of the sequential instances. The Latency is directly proportional to power and inversely proportional to skew. The Latency of the simple conventional clock distribution technique is less when compared with all the other techniques. Since the Mesh Structure uses large amounts of routes it takes more time for the signal to reach the destination, it takes 20 to 50% more delay for the mesh technique than the conventional clock tree distribution technique. The adaptive H-Tree technique lags more than mesh structure since it takes long routes to finish the top H-tree Topology and again it takes much more long routes to tap the top tree to the sink pins. There it takes around 23 to 51% more max latency than the previous two cases.

The Adaptive h-tree with mesh drivers provides the highest latency than all other techniques. The long routes of the top H-tree, large grid routes for the mesh drives and routes from the mesh drivers to the sink pins makes adaptive h-tree with mesh drivers to have 40 to 68% more arrival time than the other techniques.
The Latency Vs Power graph clearly shows that the latency and power are directly proportional quantities. The power of the adaptive H-tree with mesh drivers is the highest of all due to the usage of the maximum number of buffers and highest wire length for clock tree distribution in figure 8 and 9.

Table 2. Comparison table of all the clock tree algorithms on benchmarks

| Benchmarks | CTS | Max Latency ps | Skew ps | Power mW | Total instances | Total area µm² | Wire length µm |
|------------|-----|----------------|---------|----------|-----------------|----------------|----------------|
| B19        | Conv| 97             | 17      | 7.285    | 75643          | 157453         | 1230813        |
|            | Mesh| 137            | 25      | 39.46    | 77022          | 162444         | 1271021        |
|            | FHtree| 143      | 10      | 16.94    | 75884          | 159507         | 1245114        |
|            | FHMu l| 227     | 3       | 36.48    | 77098          | 167272         | 1269725        |
| JPEG       | Conv| 172            | 24      | 48.43    | 224374         | 606740         | 3197130        |
|            | Mesh| 169            | 27      | 64.19    | 224664         | 609587         | 3230286        |
|            | FHtree| 208    | 26      | 61.38    | 224670         | 609568         | 3209380        |
|            | FHMu l| 379   | 6       | 122      | 226832         | 628276         | 3262050        |
| VGA        | Conv| 104            | 20      | 20.53    | 39024          | 169629         | 1039587        |
|            | Mesh| 131            | 19      | 54.41    | 40034          | 177473         | 1059371        |
|            | FHtree| 120    | 12      | 27.71    | 39157          | 171087         | 1045812        |
|            | FHMu l| 244  | 4       | 56.73    | 40230          | 179852         | 1061998        |
| Leon3      | Conv| 229            | 21      | 121.9    | 450919         | 1499232        | 13145968       |
|            | Mesh| 257            | 26      | 156.9    | 451444         | 1499573        | 13180006       |
|            | FHtree| 312    | 33      | 151.2    | 451535         | 1498559        | 13186429       |
|            | FHMu l| 506  | 9       | 251      | 454618         | 1525074        | 13306446       |
| Vga Block  | Conv| 129            | 23      | 21.05    | 39531          | 170640         | 1116176        |
|            | Mesh| 134            | 18      | 51.42    | 40768          | 181179         | 1136827        |
|            | FHtree| 163   | 25      | 28.14    | 39589          | 171159         | 1119619        |
|            | FHMu l| 269  | 4       | 56.2     | 40809          | 181481         | 1136877        |
| VgaAr      | Conv| 81             | 12      | 17.3     | 38850          | 169907         | 1029925        |
|            | Mesh| 136            | 22      | 41       | 39493          | 174288         | 1053948        |
|            | FHtree| 134  | 12      | 26.88    | 38971          | 170993         | 1036615        |
|            | FHMu l| 223  | 4       | 53.21    | 40047          | 178712         | 1063086        |
| Ram        | Conv| 108            | 23      | 18.39    | 35020          | 158743         | 1419855        |
|            | Mesh| 133            | 24      | 52       | 35902          | 166563         | 1435238        |
|            | FHtree| 145  | 9       | 89.11    | 35121          | 160259         | 1425846        |
|            | FHMu l| 229  | 3       | 132.7    | 39073          | 192503         | 1499342        |
As discussed in the previous section the clock tree distribution techniques which takes the highest amount of latency with long wire length and highest amount clock buffer drivers consumes more clock power. So the adaptive h-tree with mesh drivers consumes up to 213% more clock power than the conventional technique. Whereas the Mesh structure and H-tree technique consumes 67% to 98% more clock power than conventional clock tree. The clock power consumption can be optimized using high voltage threshold cells & clock buffers, providing the toggle rate information and control the switching activity using clock gating cells. But the optimizations are not performed in this work to be aware of the actual clock power metrics of the individual clock tree distribution techniques.

4.3. Skew
The skew is one of the key parameters which decides the performance of the ASIC implementation. The shift in the skew can cause setup or hold violation which is one of the major issues in digital design. So all the clock tree distribution techniques are designed to obtain a skew value which is close to zero. The skew value for the conventional clock tree, mesh and adaptable h-tree is the range of 12 to 33ps which is a very large value.
This may lead to timing violations in the design. Whereas the adaptable H-tree with mesh drivers provides the skew of 3 to 9 ps which can be optimized further to bring down the skew close to zero. From the TABLE 2 we can infer that skew is high for larger designs like Leon3mp and low for the designs which has less instances like RAM. The skew for Vga, VgaAR and vgaBlockage which is designed using adaptive H-tree mesh drivers technique is same i.e 4ps. Which clearly tells that the top tree adapts itself depending on the blockages with minimal changes in the latency.

From the Figure 10 and 11 we can infer that the lowest skew yielding technique consumes the highest clock power. The latency and power are directly proportional metrics which is very much evident from the Figure 12 and Figure 13 by the similarities of the plots.

5. Conclusion
From the Extensive analysis of the four clock tree distribution techniques with the standard benchmarks for the digital design and implementation. The Flexible H-tree with Multi-source clock tree mesh drivers provides the least Skew with the trade off in the insertion day, clock power consumption and Buffer area even with the blockages and High aspect ratio. Therefore this clock tree distribution technique be can used in high speed designs.

Reference
[1] K. Han, A. B. Kahng and J. Li, "Optimal Generalized H-Tree Topology and Buffering for High Performance and Low-Power Clock Distribution," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 2, pp. 478-491, Feb. 2020, doi: 10.1109/TCAD.2018.2889756.
[2] S. N. Shahsavani and M. Pedram, "A Minimum-Skew Clock Tree Synthesis Algorithm for Single Flux Quantum Logic Circuits," in IEEE Transactions on Applied Superconductivity, vol. 29, no. 8, pp. 1-13, Dec. 2019, Art no. 1303513, doi: 10.1109/TASC.2019.2943930.
[3] A. D. Mehta, Yao-Ping Chen, N. Menezes, D. F. Wong and L. T. Pilegik, "Clustering and load balancing for buffered clock tree synthesis," Proceedings International Conference on Computer Design VLSI in Computers and Processors, Austin, TX, USA, 1997, pp. 217-223, doi: 10.1109/ICCD.1997.628871.
[4] P. J. Restle et al., "A clock distribution network for microprocessors," in IEEE Journal of Solid-State Circuits, vol. 36, no. 5, pp. 792-799, May 2001, doi: 10.1109/4.918917.
[5] R. Samanta, I. Hu and P. Li, "Discrete Buffer and Wire Sizing for Link-Based Non-Tree Clock Networks," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 7, pp. 1025-1035, July 2010, doi: 10.1109/TVLSI.2009.2019088.
[6] H. Seo, J. Kim, M. Kang and T. Kim, "Synthesis for power-aware clock spines," 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 126-131, doi: 10.1109/ICCAD.2015.7372559.
[7] Y. Kim and T. Kim, "Algorithm for synthesis and exploration of clock spines," 2017 22nd Asia
and South Pacific Design Automation Conference (ASP-DAC), Chiba, Japan, 2017, pp. 263-268, doi: 10.1109/ASPDAC.2017.7858330.

[8] S, D., & H, A. (2019). AODV Route Discovery and Route Maintenance in MANETs. 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). doi:10.1109/icaccs.2019.8728456

[9] H. Anandakumar and K. Umamaheswari, An Efficient Optimized Handover in Cognitive Radio Networks using Cooperative Spectrum Sensing, Intelligent Automation & Soft Computing, pp. 1–8, Sep. 2017. doi:10.1080/10798587.2017.1364931

[10] E. K. Teh, M. A. M. Zawawi, M. F. P. Mohamed and N. A. M. Isa, "Practical Full Chip Clock Distribution Design With a Flexible Topology and Hybrid Metaheuristic Technique," in IEEE Access, vol. 9, pp. 14816-14835, 2021, doi: 10.1109/ACCESS.2021.3053052.

[11] V. G. Srivatsa, A. P. Chavan and D. Mourya, "Design of Low power & High Performance Multi Source H-Tree Clock Distribution Network," 2020 IEEE VLSI DEVICE CIRCUIT AND SYSTEM (VLSI DCS), Kolkata, India, 2020, pp. 468-473, doi: 10.1109/VLSIDCS47293.2020.9179954.