

Multi-Label Low Level Cue Algorithm For Object-Level Image Segmentation

M.Arun^{#1}, R.V.Saranya^{*2}, R.K.Yuvasti^{#3}

^{#1} UG scholar, Department of Electronics and Communications Engineering, K.L.N.College of Engineering, Pottapalayam, Tamil Nadu, India.

^{*2} Assistant Professor, Department of Electronics and communication Engineering, K.L.N.College of Engineering, Pottapalayam, Tamil Nadu, India.

^{#3} UG scholar, Department of Electronics and Communications Engineering, K.L.N.College of Engineering, Pottapalayam, Tamil Nadu, India.

ABSTRACT—This paper considers the problem of automatically segmenting an image into a small number of regions that correspond to objects conveying semantics or high-level structure. While such object-level segmentation usually requires additional high-level knowledge or learning process, we explore what low level cues can produce for this purpose. Our idea is to construct a feature vector for each pixel, which elaborately integrates spectral attributes, color Gaussian Mixture Models and geodesic distance, such that it encodes global color and spatial cues as well as global structure information. Then we formulate the Potts variational model in terms of the feature vectors to provide a variational image segmentation algorithm that is performed in the feature space. We also propose a heuristic approach to automatically select the number of segments. The use of feature attributes enables the Potts model to produce regions that are coherent in color and position, comply with global structures corresponding to objects or parts of objects and meanwhile maintain a smooth and accurate boundary. We demonstrate the effectiveness of our algorithm against the state-of-the-art with the dataset from the famous Berkeley benchmark.

KEY WORDS—Image segmentation, low level cues, object segmentation, variational model.

I. INTRODUCTION

Based on the elimination feature of redundant inverters in merging 1-bit flip-flops into multi-bit flip-flops, gives reduction of wired length and this result in reduction of power consumption. With the growing popularity of portable devices, power reduction has become a popular design goal for advanced design application.

In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite-state machine, the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing signal.

Flip-flops can be either simple (transparent or opaque) or clocked (synchronous or edge-triggered); the simple ones are commonly called latches. The word latch is mainly used for storage elements, while clocked

devices are described as flip-flops. A latch is level-sensitive, whereas a flip-flop is edge-sensitive. That is, when a latch is enabled it becomes transparent, while a flip-flop's output only changes on a single type (positive going or negative going) of clock edge. The flip-flops on clock tree accounted for a large proportion of power consumption. Although the power distribution will vary with different ASIC design, reducing power consumption of the flip-flop on clock tree can eliminate total power consumption efficiently.

Multi-bit flip-flop is an effective power-saving implementation methodology by merging single-bit flip-flops in the design. Using multi-bit flip-flops can reduce clock dynamic power and the total flip-flop area effectively.

Due to the popularity of portable electronic products, low power system has attracted more attention in recent years. As technology advances, a systems-on-chip (SOC) design can contain more and more components that lead to a higher power density. This makes power dissipation reach the limits of what packaging, cooling or other infrastructure can support. Reducing the power consumption not only can enhance battery life but also can avoid the overheating problem, which would increase the difficulty of packaging or cooling .

A Single bit flip flop

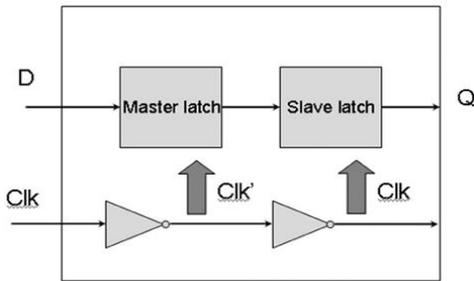


Figure 1.1 Single bit flip flop.

The latches need “Clk” and “Clk’ ” signal to perform operations, such as Figure2 shows. In order to have better delay from Clk-> Q, we will regenerate “Clk” from “Clk’ ”. Hence we will have two inverters in the clock path.duv

1.2 Merging two one-bit flip flops into one two-bit flip flop

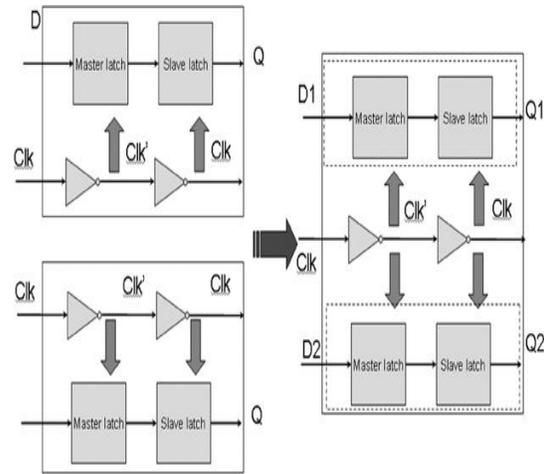


Figure 1.2 Merging two 1-bit flip-flops into one 2-bit flip-flop.

Each 1-bit flip-flop contains two inverters, master-latch and slave-latch. Due to the manufacturing rules, inverters in flip-flops tend to be oversized. As the process technology advances into smaller geometry nodes like 65nm and beyond, the minimum size of clock drivers can drive more than one flip-flop. Merging single-bit flip-flops into one multi-bit flip-flop can avoid duplicate inverters, and lower the total clock dynamic power consumption. The total area contributing to flip-flops can be reduced as well.

II. CLUSTER USING MULTI BIT FLIP FLOPS FOR POWER REDUCTION

We propose an efficient two-phase approach to merge the possible 1-bit flipflops into some multi-bit flip-flops for clock power reduction.

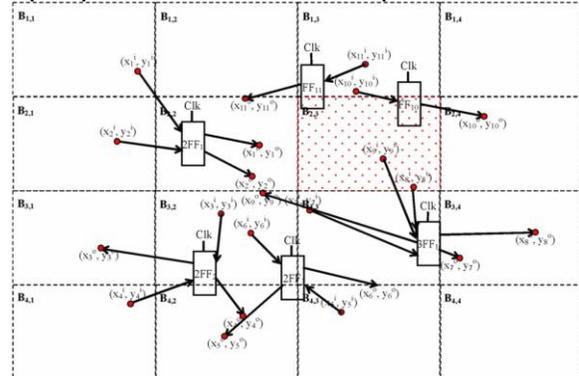


Figure 2 Multi-bit flip flops.

B. Iterative merging process in a merging graph

Two flip flops are merged into a multi-bit flip-flops,two redundant inverters can be eliminated in a merging process.In the meging process of all the 1-bit flip-flops,it is clear that the number of the merging pairs is at most (n-1),where n is the number of the given 1-bit flip-flops. For the construction of constrained multi-bit

flip-flops, an iterative merging process can be proposed. Firstly, all the vertices in a merging graph are sorted in an increasing order according to the degrees of all vertices in the merging graph. The vertices with the smallest degree and their connecting vertices can be selected for the construction of the merging pairs and the original merging graph can be modified by deleting the selected vertices and adding the merged vertices. The edge set in the modified merging graph is empty the iterative merging process will stop. As a result the final merging result represents a set of constrained multi-bit flip-flops.

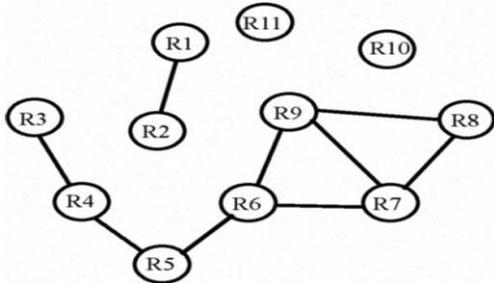


Figure 2.1 Construction of a merging graph for 1-bit flip-flops.
C. Analysis of time complexity

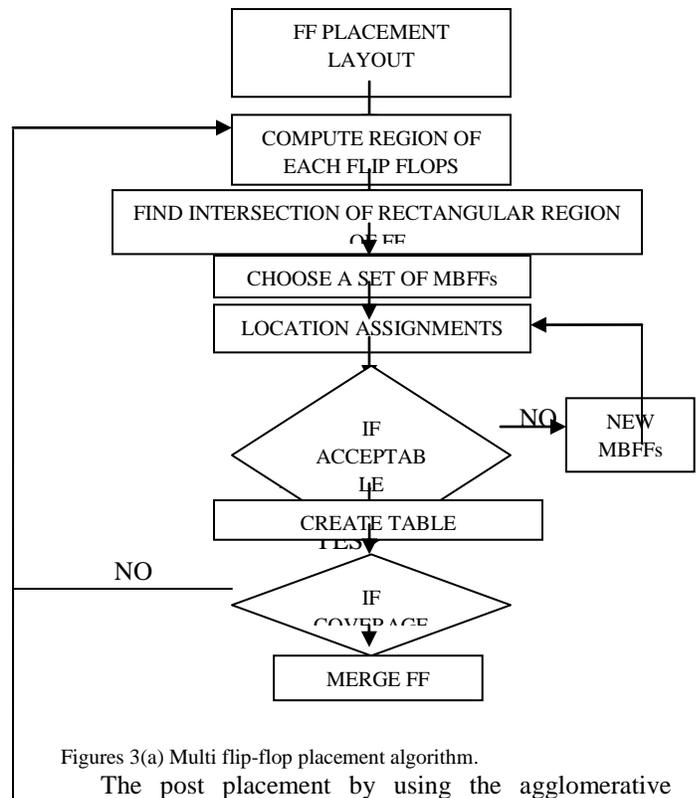
Set of 1-bit flip-flops with the routing net constraints and the bin congestion constraints in a placement plane, our proposed merging process for the construction of constrained multi-bit flip-flops can be divided into two sequential phases: construction of a merging graph for 1-bit flip-flops and iterative merging process in a merging graph. For the analysis of time complexity in the construction of a merging graph, based on the intersecting regions among all the feasible location regions it is clear that the number of vertices in a merging graph, based on the intersecting regions among all the feasible location regions, it is clear that the number of vertices in a graph is in $O(n)$ and the number of edges in a merging graph is in $O(n^2)$, where n is the number of 1-bit flip-flops in a placement plane. Since the number of the maximum merging pairs is in $O(n)$, the time complexity of finding all the merging pairs is in $O(n)$. Hence the time complexity of running the iterative merging process is in $O(n \log n)$. The time complexity of the proposed merging approach for the construction of constrained multi-bit flip-flops is in $O(n^2)$.

III. PROPOSED ALGORITHM

We propose agglomerative clustering algorithm to find the nearest clustering of flip flops for merging the flip flops. The algorithm forms clusters in a bottom-up manner, as follows:

1. Among all current clusters, pick the two clusters with the smallest distance.

2. Replace these two clusters with a new cluster, formed by merging the two original ones. This algorithm finds the clusters of flip flop and finally combine FFs to reduce the wire length.



Figures 3(a) Multi flip-flop placement algorithm.

The post placement by using the agglomerative clustering algorithm. Inter connecting wirelength such that both placement density and timing slack constraints are satisfied. The power optimization with multi flip flops. The power optimization flow to gather with the flip flop grouping. The multi flip flop placement algorithm called as agglomerative clustering algorithm. The table is create for the clustering algorithm. The total number of flip flop is coverage. Finally merge the flip flop.

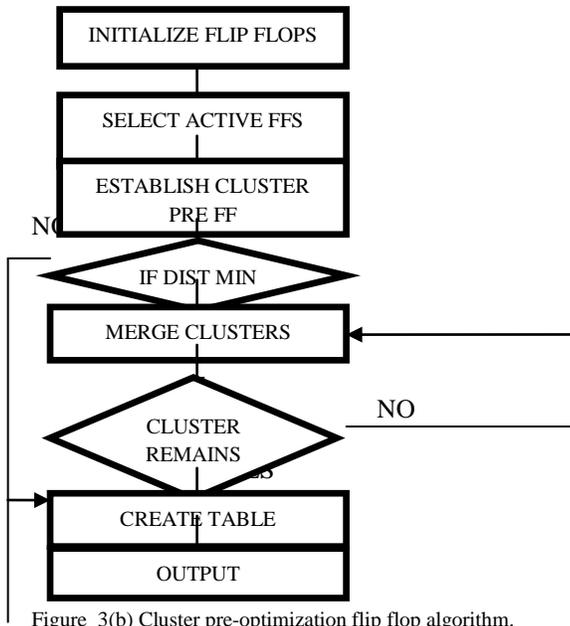


Figure 3(b) Cluster pre-optimization flip flop algorithm.

Initially put each article in its own cluster in the flip flops. Then flip flop selective to the active flip flop. The pre-optimization with flip flop in a establish cluster. If it is a distance minimum to the agglomerative clustering algorithm. And we can merge to the cluster. And then remaining cluster to the table format. Finally we get the output.

IV. IMPLEMENTATION

A. Modules

1. Construct Delay Flip flops,
2. Check distance,
3. Form cluster Flip Flops,
4. Create table,
5. Merge flip flop.

B. Module description

1) Construct Delay Flip Flops

A master-slave D flip-flop is created by connecting two gated D latches in series, and inverting the enable input to one of them. This flip-flops gets single bit as input and generate output as Q and Q'. At a time any one of Delay flip-flop was activated according to clock pulse generation. Each FlipFlop handle single bit input for separated clock generator.

2) Check Distance

Normally, flip flops in a logic circuit was handling single bit input with separated clock generation. This may consume more power. This can reduce by merging of flip flops. In this module, they find the distance between the neighboring flip flops which gets the distance between each flip flops.

The Manhattan distance function computes the distance that would be traveled to get from one data

point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

The formula for this distance between a point X=(X1, X2, etc.) and a point Y=(Y1, Y2, etc.) is:

$$d = \sum_{i=1}^n |x_i - y_i|$$

Where n is the number of variables, and Xi and Yi are the values of the ith variable, at points X and Y respectively.

The difference between Manhattan distance and Euclidean distance:

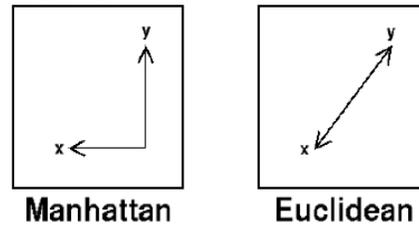


Figure 4.2.2 Check distance.

3) Form Cluster Flip Flops

We form flip flops in clusters according to the distance information. This clustering represents what are the flip flops can be made to merge and form as a multi bit flip flops. This clustering was found by agglomerative clustering algorithm to form clustered flip flop.

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

The hierarchy within the final cluster has the following properties:

1. Clusters generated in early stages are nested in those generated in later stages.
2. Clusters with different sizes in the tree can be valuable for discovery.

4) Create Table

We construct a table called as combinational table which contains cluster information of each flip flop and distances between them. From this, the decision was taken to merge FFs. This table was update according to the application usage. Then from this information, we merge FF at a mergable range of clusters.

5) Merge Flip Flops

We finally merge flip flops according to the acceptable range from combinational table. This formation made the individual FFs in to multi-bit flip-flops which contains common clock for FFs and form multi bit FlipFlops.

V. PROBLEM DESCRIPTIONS

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

In this work, we implement Flip-Flop clustering by using agglomerative clustering. Here we extract Flip-Flop wire length by finding distance between each Flip-Flop and create a table and arrange distance information of Flip-Flops. Then from that we extract clusters for acceptable rate. Then we form merging of Flip-Flops which are form in clusters. By this we reduce the wire-length of each Flip-Flops.

VI. EXPERIMENTAL RESULTS

We implemented our algorithms in the C++ programming language, and performed three sets of experiments on a 2.66GHz Intel Core i7 PC under the Linux operating system. First of all, we compared the flip-flop power consumption, wirelength. Finally, we merging to the flip-flop in the power reduction and wirelength reduction.

TOTAL NUMBER OF FLIP FLOP MERGED = 121

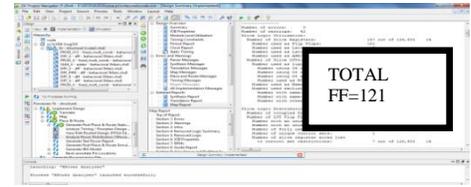


Figure. 6. Total number of flip flop merged in this model.

METHODS	AREA	SPEED	POWER
EXISTING SYSTEM	FFs(264)	195.32(MHZ)	43(mwatt)
PROPOSED SYSTEM	FFs(148)	233.05(MHZ)	36(mwatt)

TABULATION FOR TOTAL POWER CONSUMPTION.

TOTAL POWER CONSUMPTION=36mW		
Power		
V _{CCINT}	1.000V	0.000W
V _{CCAUX}	1.800V	0.000W
V _{CCADC}	1.800V	0.036W

A. Power Reduction

With the growing popularity of portable devices, power reduction has become a popular design goal for advanced design application, whether mobile or not. Reducing power consumption in chips enables better, cheaper products to be designed and power-related chip failures to be minimized. As a result, how to minimize power consumption has become an important design goal that every chip designer must take care.

Several lower power design techniques have played an important role in the design flow. Clock gating methodology is used for the register bank to replace the multiplexers and it can avoid the operation of reloading the same data value. The clock gating technique could reduce the dynamic power consumption efficiently. The multi-Vth concept is aimed at using multi-Vth cell with satisfying performance to reduce leakage consumption, and replace lower Vth (LVT) cells by high Vth (HVT) ones, if there is room for slack. Multiple Supply Multiple

Voltage (MSMV) supplies of different voltages are used for core logic, base on satisfy performance or functional requirement to adjust operating voltage for each domain, even shut off this domain.

We can see that the flip-flops on clock tree accounted for a large proportion of power consumption. Although the power distribution will vary with different ASIC design, reducing power consumption of the flip-flop on clock tree can eliminate total power consumption efficiently.

TABLE VI
COMPARISON OF VARIOUS CONSUMPTION.

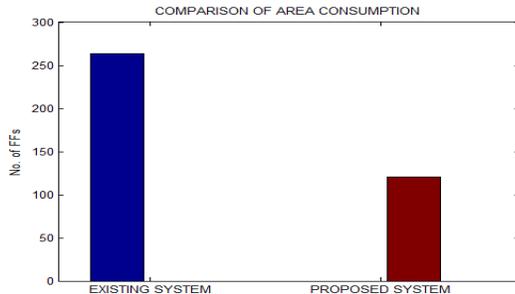


Figure 6.1(a) Comparison of Area

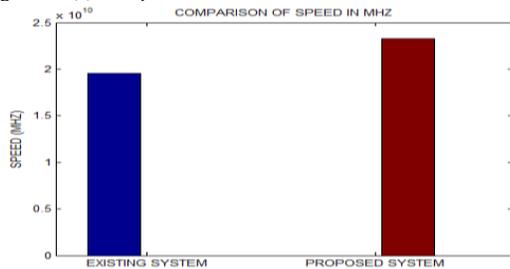


Figure 6.1(b) Comparison of Speed.

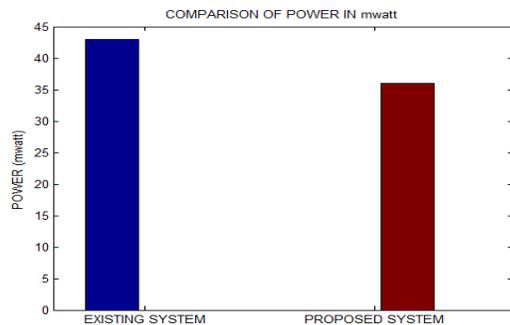


Figure 6.1 (c) Comparison of power.

VII. CONCLUSION

We proposed merging of flip flops to reduce power reduction of single bit Flip Flop to form multi bit Flip Flop. This merging was found by acceptable distance range each neighboring Flip Flops. This was done by

forming clusters of flip flops by using agglomerative clustering algorithm.

VII.ACKNOWLEDGMENT

The authors would like to thank Dr.Y.-W. Tsai and S.-F. Chen from Faraday Technology Corporation, Hsinchu, Taiwan, for providing the measured data including area and power consumption of the multi-bit flip-flops, and the benchmark circuits. They would also like to thank Prof.W.-K. Mak's team of National Tsing Hua University, Hsinchu, for providing the benchmark circuits.

REFERENCES

- [1] C. Bron and J. Kerbosch (1973), "Algorithm 457: Finding all cliques of an undirected graph", ACM Commun., vol. 16, no. 9, pp. 575–577.
- [2] CAD Contest of Taiwan [Online]. Available: http://cad_contest.cs.nctu.edu.tw/cad11.
- [3] D. Duarte, V. Narayanan, and M. J. Irwin (2002), "Impact of technology scaling in the clock power in Processing" IEEE VLSI Comput. Soc. Annu. Symp., Pittsburgh, PA, pp. 52–57.
- [4] Faraday Technology Corporation [Online]. Available: <http://www.faraday-tech.com/index.html>.
- [5] H. Kawaguchi and T. Sakurai (1997), "A reduced clock-swing flip-flop (RCSFF) for 63% clock power reduction", in VLSI Circuits Dig. Tech. Papers Symp., pp. 97–98.
- [6] Jin-Tai Van, Zhi-Wei Chen(2010), "construction of constrained multi-bit flip-flops for clock power reduction".
- [7] Mark Po-Hung Lin, Chih-Cheng Hsu, and Yao-Tsung Chang(2011), "Recent research in clock power saving with multi-bit flip flops".
- [8] P. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon(1998), "High-performance microprocessor design", IEEE J. Solid-State Circuits, vol. 33, no. 5, pp. 676–686.
- [9] R. Lu, G. Zhong, C. Koh, and K. Chao, "Flip-flop and repeater insertion for early interconnect planning(2002)," Design Automation and Test in Europe, pp.690-695.
- [10] W. Hou, D. Liu, and P.-H. Ho (2009), "Automatic register banking for lowpower clock trees", in Processing Quality Electron. Design, San Jose, CA, pp. 647–652.
- [11] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, and Q. Wang (2005), "Power-aware placement", in Processing Design Autom. Conf., pp. 795–800.
- [12] Y.-T. Chang, C.-C. Hsu, P.-H. Lin, Y.-W. Tsai, and S.-F. Chen (2010), "Post-placement power optimization with multi-bit flip-flops", in Processing IEEE/ACM Comput.-Aided Design Int. Conf., San Jose, CA, pp. 218–223.
- [13] Zhi-Wei Chen, Jin-Tai Yan(2012), "Utilization of multi-bit flip-flops for clock power reduction", in College. of Engineering., Chung-Hua University., Hsinchu, Taiwan, R.O.C.