ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Implementation of Multi-Bit flip-flop for Power Reduction in CMOS Technologies

K. Siva Prasad, G. Rajesh, V. Thrimurthulu

Post Graduate student, Department of ECE, CREC, TIRUPATI, A.P., India

Associate Professor, Department of ECE, CREC, TIRUPATI, A.P., India

Head of the Department, Department of ECE, CREC, TIRUPATI, A.P., India

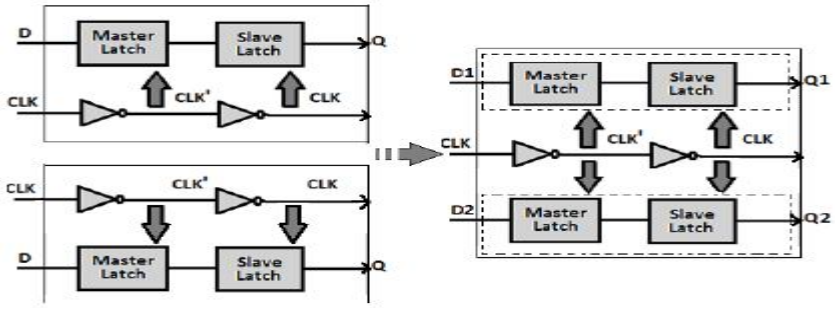
ABSTRACT: Nowadays Power has become a major concern in low power VLSI design. Achieving low power consumption is a tedious one in IC fabrication industries. In modern integrated circuits, clocking is the most dominating power consuming element. Hence this paper describes a method for reducing the power consumption by replacing some flip-flops with fewer multi-bit flip-flops. We perform a co-ordinate transformation to identify those flip-flops that can be merged and their legal regions. Besides, we show how to build a combination table to enumerate possible combinations of flip-flops provided by a library. Finally, we use a hierarchical way to merge flip-flops. Besides power reduction, the objective of minimizing the total wire length is also considered. This algorithm can reduce clock power and the running time is very short.

KEYWORDS: Low power, merging, multi-bit flip-flop, replacement, wire length, clock power reduction.

I. INTRODUCTION

In recent years the low power systems are more attracted. As technology progressing, a systems-on-a-chip design consists of increased number of components which is leading to a higher power density. This complexity limits the power dissipation, cooling or other infrastructure can support. Overheating affects the battery power, which also causes the difficulty of packaging or cooling. Therefore, the consideration of power consumption in complex SOCs has become a big challenge to designers. Moreover, in modern VLSI designs, power consumed by clocking has taken a major part of the whole design especially for those designs using deeply scaled CMOS technologies. As a result, several methodologies have been proposed to reduce the power consumption of clocking.

Besides, for a design with power consumption, the smaller flip-flops are replaced by larger multi-bit flip-flops, so that the device variations in the corresponding circuit can be reduced effectively.



(a) (b)

Fig.1.Example of merging two 1-bit flip-flops into one 2-bit flip-flop.(a) Two 1-bit flip-flops (before merging). (b) 2-bit flip-flop (after merging).

Fig. 1 shows the block diagrams of 1- and 2-bit flip-flops. If we replace the two 1-bit flip-flops as shown in

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Fig. 1(a) wiht the 2-bit flip-flop as shown in Fig. 1(b), the total power consumption can be reduced because the two 1-bit flip-flops can share the same clock buffer.

After this replacement, the locations of the flip-flops and the wirelengths of nets connecting pins to a flip-flops will be changed. To avoid violating the timing constraints, we restrict the wire lengths of nets, which cannot be longer than specified values after this process. Besides, to guarantee that a new flip-flop can be placed within the desired region, we also need to consider the area capacity of the region. As shown in Fig. 2(a), after the two 1-bit flip-flops f_1 and f_2 are replaced by the 2-bit flip-flop f_3, the wire lengths of nets net_1, net_2, net_3, and net_4 are changed. To avoid the timing violation caused by the replacement, the Manhattan distance of new nets net_1, net_2, net_3, and net_4 cannot be longer than the specified values.

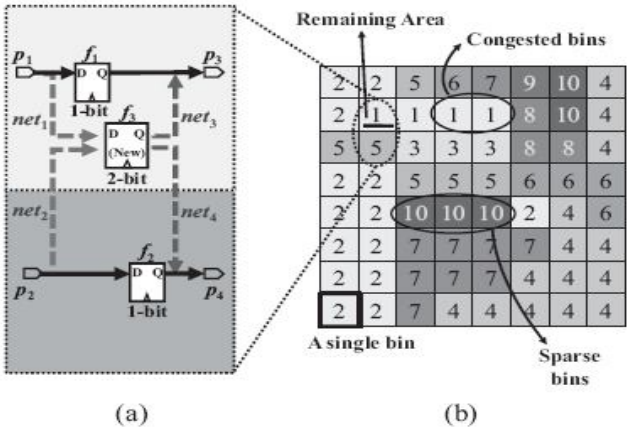


Fig. 2. (a) The wire length variation with flip-flops combination.(b) The Changes the density with flip-flop combination

In Fig. 2(b), we divide the whole placement region into several bins, and each bin has a defined area capacity denoting the remaining area that additional cells can be placed within it.Suppose the area of f_3 is 7 and f_3 is assigned to be placed in the same bin as f_1. We cannot place f_3 in that bin since the remaining area of the bin is smaller than the area of f_3. In addition to the considerations mentioned in the above, we also need to check whether the cell library provides the type of the new flip-flop. For example, we have to check the availability of a 3-bit flip-flop in the cell library when we desire to replace 1- and 2-bit flip-flops by a 3-bit flip-flop.

II. RELATED WORK

The problem we are facing in usage of multi-bit flip-flops to reduce power consumption is in the post-placement stage. They use the graph-based approach to deal with this problem. In a graph, each node represents a flip-flop. If two flip-flops can be replaced by a new flip-flop without violating timing and capacity constraints, they build an edge between the corresponding nodes. After the graph is built, the problem of replacement of flip-flops can be solved by finding an m -clique in the graph. The flip-flops corresponding to the nodes in an m -clique can be replaced by an m -bit flip-flop. They use the branch-and-bound and backtracking algorithm to find all m -cliques in a graph. Because one node (flip-flop) may belong to several m -cliques (m –bit flip-flop), they use greedy heuristic algorithm to find the maximum independent set of cliques, which every node only belongs to one clique, while finding m -cliques groups. However, if some nodes correspond to k -bit flip-flops that k >1, the bit width summation of flip-flops corresponding to nodes in an m-clique, j , may not equal m .If the type of a j -bit flip-flop is not supported by the library, it may be time-wasting in finding impossible combinations of flip-flops.

III. OUR MAIN CONTRIBUTION FOR THE PAPER

In dealing with the difficulty of the problem described above effectively, we have to repeatedly search a set of flip-flops in a particular location that can be replaced by a new multi-bit flip-flop until none can be done. However

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

as the number of flip-flops in a chip increases in the circuit, the complexity would increase exponentially, which makes the method practically difficult. To handle this problem more effectively and get better results, we have used the following approaches.

1) Facilitating the identification of mergeable flip-flops by transforming the coordinate system of cells. This method also helps to reduce the memory used to record the feasible placement region.

2) To avoid wasting time in finding impossible combinations of flip-flops, we first build a combination table before actually merging two flip-flops, which contains only the bits that are used in the Integrated Circuit (like 1-,2-, 3-bit etc) according to it's library, avoiding the combinations that are not available in the library.

3) We do partition of the chip into several sub-regions and perform replacement in each sub-region to reduce the complexity.

IV. PROPOSED METHOD

The following flow chart is divided into three stages. First we have to identify a **legal placement region** for each flip-flop f_i . Then the feasible placement region of a flip-flop associated with different pins are found based on the timing constraints defined on the pins. The legal placement region of the flip-flop f_i can be obtained by the overlapped area of these regions. However, because these regions are in the diamond shape, it is not easy to identify the overlapped area. Therefore, the overlapped area can be identified more easily if we can transform the coordinate system of cells to get rectangular regions. In the second stage, we would like to build a **combination table,** which defines all possible combinations of flip-flops in order to get a new multi-bit flip-flop provided by the library. The flip-flops can be merged with the help of the table. After the legal placement regions of flip-flops are found and the combination table is built, we can use them to **merge flip-flops**.

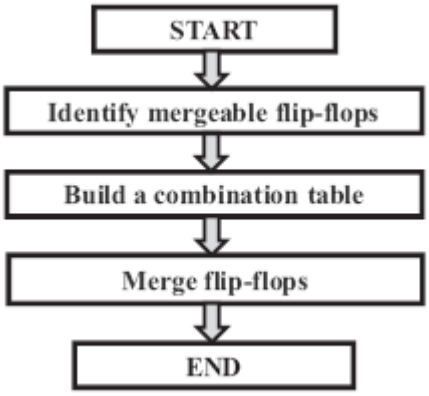


Fig. 3. Flow chart of our algorithm

To increase the speed of the program, we divide a chip into several bins and merge flip-flops in a local bin. The flip-flops in different bins can also be mergeable. These bins are combined into a larger bin and repeat this step until no flip-flop can be merged anymore. In this section, we would detail each stage of our method. In the first subsection, we show a simple formula to transform the original coordination system into a new one so that a legal placement region for each flip-flop can be identified more easily. The second subsection presents the flow of building the combination table. Finally, the replacements of flip-flops will be described in the last subsection.

A. Identification of mergeable flip-flops

We have to Facilitate the identification of mergeable flip-flops by transforming the coordinate system of cells. In this way, the memory used to record the feasible placement region can also be reduced. We have shown that

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

the shape of a feasible placement region associated with one pin p_i connecting to a flip-flop f_i would be diamond in Section II. Since there may exist several pins connecting to $\boldsymbol{f}_i$, the legal placement region of $\boldsymbol{f}_i$ are the overlapping area of several regions. As shown in Fig. 4(a), there are two pins p_1 and p_2 connecting to a flip-flop f_1, and the feasible placement regions for the two pins are enclosed by dotted lines, which are denoted by $R_p(p_1)$ and $R_p(p_2)$, respectively. Thus, the legal placement region $R(f_1)$ for f_1 is the overlapping part of these regions. In Fig. 4(b), $R(f_1)$ and $R(f_2)$ represent the legal placement regions of f_1 and f_2. Because $R(f_1)$ and $R(f_2)$ overlap, we can replace f_1 and f_2 by a new flip-flop f_3 without violating the timing constraint, as shown in Fig. 4(c).

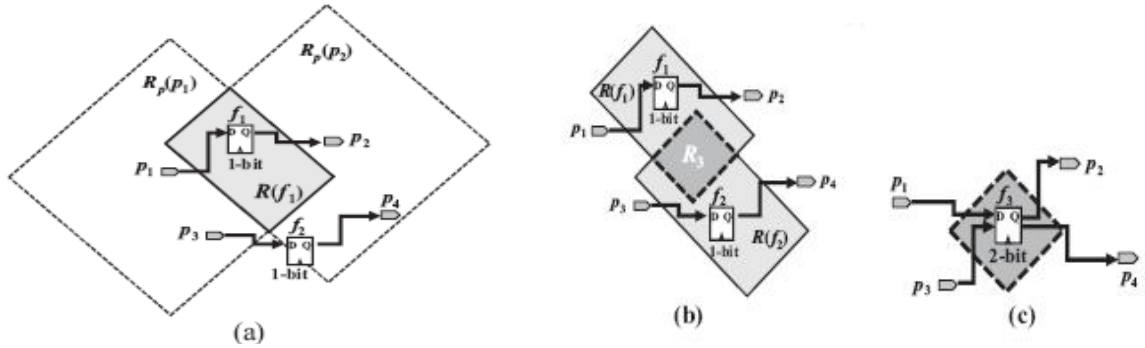


Fig. 4. (a) Feasible regions $R_p(p_1)$ and $R_p(p_2)$ for pins p_1 and p_2 which are enclosed by dotted lines, and the legal region $R(f_1)$ for f_1 which is enclosed by solid lines. (b) Legal placement regions $R(f_1)$ and $R(f_2)$ for f_1 and f_2, and the feasible area R_3 which is the overlap region of $R(f_1)$ and $R(f_2)$.(c) New flip-flop f_3 that can be used to replace f_1 and f_2 without violating timing constraints for all pins p_1, p_2, p_3, and p_4.

The diamond shaped regions are very difficult to identify in the circuit. Moreover, four coordinates are required to record an overlapping region [see Fig. 4(a)]. Thus, if we can rotate each segment 45°, the shapes of all regions would become rectangular, which makes identification of overlapping regions become very simple. For example, the legal placement region, enclosed by dotted lines in Fig. 4(a), can be identified more easily if we change its original coordinate system [see Fig. 4(b)]. In such condition, we only need two coordinates, which are the left-bottom corner and right-top corner of a rectangle, as shown in Fig. 4(b), to record the overlapped area instead of using four coordinates.

B. Building a Combination Table

The new flip-flop f_i provided by the library $\boldsymbol{L}$ is chosen initially to replace with summation of the flip-flops(note that the bit width of f_i should equal to the summation of bit widths of these flip-flops), when the feasible regions of these flip-flops overlap. In this paper, we will build a combination table, which records all possible combinations of flip-flops to get feasible flip-flops before replacements. Thus, we can gradually replace flip-flops according to the order of the combinations of flip-flops in this table. Since only one combination of flip-flops needs to be considered in each time, the search time can be reduced greatly.

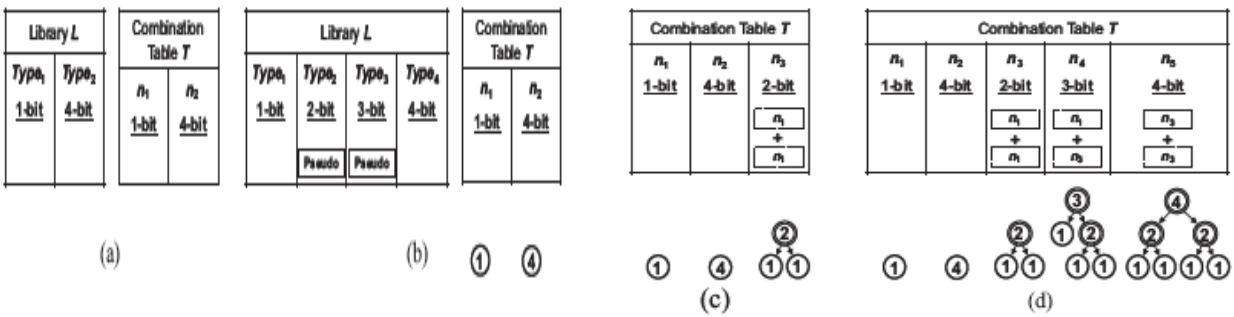


Fig. 5. Example of building the combination table. (a) Initialize the library L and the combination table T. (b)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Pseudo types are added into L, and the corresponding binary tree is also build for each combination in T. (c) New combination n_3 is obtained from combining two n_1s. (d) New combination n_4 is obtained from combining n_1 and n_3, and the combination n_5 is obtained from combining two n_3s.

For simplification, we use a binary tree to represent one combination. Each node in the tree denotes one type of a flip-flop in ***L***. The types of flip-flops denoted by leaves will constitute the type of the flip-flop in the root. For each node, the bit width of the corresponding flip-flop equals to the bit width summation of flip-flops denoted by its left and right child. Let n_i denote one combination in ***T*** , and $b(n_i)$ denote its bit width. In the Beginning, we initialize a combination n_i for each kind of flip-flops in ***L*** . Then, in order to represent all combinations by using a binary tree, we may add **pseudo types**, which denote those flip-flops that are not provided by the library.

For example, assume that a library only supports two kinds of flip-flops whose bit widths are 1 and 4, respectively. In order to use a binary tree to denote a combination whose bit width is 4, there must exist flip-flops whose bit widths are 2 and 3 in ***L***. Thus, we have to create two pseudo types of flip-flops with 2- and 3-bit if ***L*** does not provide these flip-flops. Function **InsertPseudoType** in algorithm 1 shows how to create pseudo types. Let b_{max} and b_{min} denote the maximum and minimum bit width of flip-flops in ***L***. In **InsertPseudoType**, it inserts all flip-flops whose bit widths are larger than b_{min} and smaller than b_{max} into ***L*** if they are not provided by ***L*** originally. After this procedure, all combinations in ***L*** are sorted according to their bit widths in the ascending order. At present, all combinations are represented by binary trees with 0-level.

C. Merging of Flip-Flops

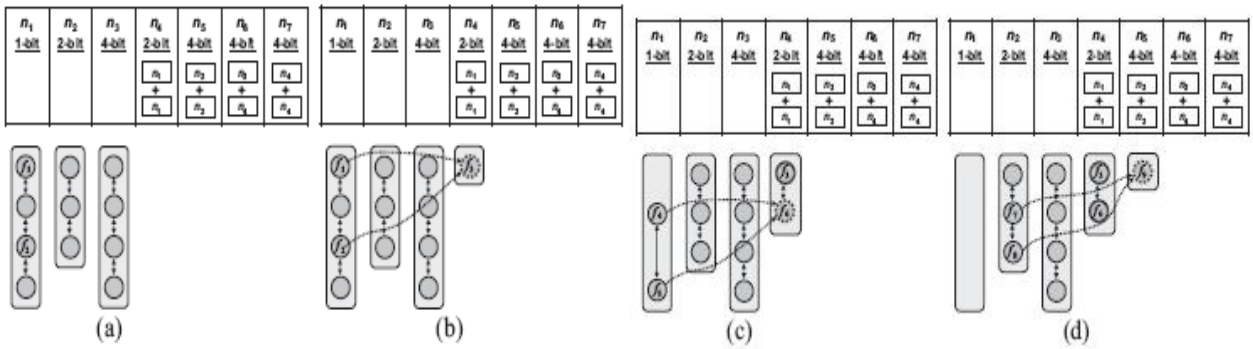
How to use the combination table to combine flip-flops is discussed in this section. To reduce the complexity, we first divide the whole placement region into several sub regions, and use the combination table to replace flip-flops in each sub region. Then, several subregions are combined into a larger sub region and the flip-flops are replaced again so that those flip-flops in the neighbouring sub regions can be replaced further. Finally, those flip-flops with pseudo types are deleted in the last stage because they are not provided by the supported library.

1) Partition of the Region: To speed up our problem, we divide the whole chip into several sub regions. By suitable partition, the computation complexity of merging flip-flops can be reduced significantly. We divide the region into several sub regions, and each sub region contains six bins, where a bin is the smallest unit of a sub region.

2) Replacement of Flip-flops in Each Sub region: Before illustrating our procedure to merge flip-flops, we first give an equation to measure the quality if two flip-flops are going to be replaced by a new flip-flop as follows:

$$\text{cost} = \text{routing\_length} - \alpha \times \sqrt{\text{available\_area}}$$

The routing length denotes the total routing length between the new flip-flop and the pins connected to it, and avail-able area represents the available area in the feasible region for placing the new flip-flop.α is a weighting factor. The cost function includes the term routing length to favor a replacement that induces shorter wire length. Besides, if the region has larger available space to place a new flip-flop, it implies that it has higher opportunities to combine with other flip-flops in the future and more power reduction. Thus, we will give it a smaller cost. Once the flip-flops cannot be merged to a higher-bit type, we ignore the *available area* in the cost function, and hence α is set to 0.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

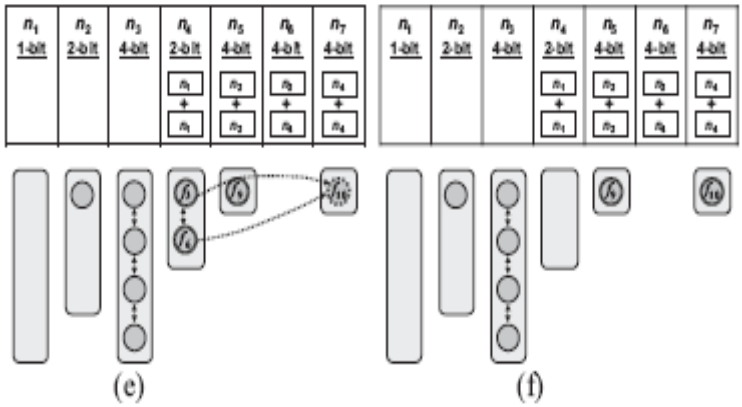


Fig. 6. Example of replacements of flip-flops. (a) Sets of flip-flops before merging. (b) Two 1-bit flip-flops, *f*1 and *f*2, are replaced by the 2-bit flip-flop *f*3. (c) Two 1-bit flip-flops, *f*4 and *f*5, are replaced by the 2-bit flip-flop *f*6. (d) Two 2-bit flip-flops, *f*7 and *f*8, are replaced by the 4-bit flip-flop *f*9. (e) Two 2-bit flip-flops, *f*3 and *f*6, are replaced by the 4-bit flip-flop *f*10. (f) Sets of flip-flops after merging.

After a combination has been built, we will do the replacements of flip-flops according to the combination table. First, we link flip-flops below the combinations corresponding to their types in the library. Then, for each combination ***n*** in ***T,*** we serially merge the flip-flops linked below the left child and the right child of ***n*** from leaves to root.

For example, given a library containing three types of flip-flops (1-, 2-, and 4-bit), we first build a combination table ***T*** as shown in Fig. 6(a). In the beginning, the flip-flops with various types are, respectively, linked below n_1, n_2, and n_3 in ***T*** according to their types. Suppose we want to form a flip-flop in n_4, which needs two 1-bit flip-flops according to the combination table. Each pair of flip-flops in n_1 are selected and checked to see if they can be combined (note that they also have to satisfy the timing and capacity constraints described in Section II). If there are several possible choices, the pair with the smallest cost value is chosen to break the tie. In Fig. 6(a), f_1 and f_2 are chosen because their combination gains the smallest cost. Thus, we add a new node f_3 in the list below n_4, and then delete f_1 and f_2 from their original list [see Fig. 6(b)]. Similarly, f_4 and f_5 are combined to obtain a new flip-flop f_6, and the result is shown in Fig. 6(c). After all flip-flops in the combinations of 1-level trees (n_4 and n_5) are obtained as shown in Fig. 6(d), there exist some flip-flops in the lists below n_2 and n_4, and we will merge them to get flip-flops in n_6 and n_7, respectively. Suppose there is no overlap region between the couple of flip-flops in n_2 and n_4. It fails to form a 4-bit flip-flop in n_6. Since the 2-bit flip-flops f_3 and f_6 are mergeable, we can combine them to obtain a 4-bit flip-flop f_{10} in n_7.

IV. EXPERIMENTAL RESULTS

In this work we are evaluating the performance of the replacing flip-flops with fewer multi bit flip-flops for low power consumption. These multi bit flip-flop can be implemented using Verilog coding. In order to get the power report and delay report we are synthesizing the merging using Xilinx and modelsim. Simulation results for the Multi bit flip-flops are shown in the following figure.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

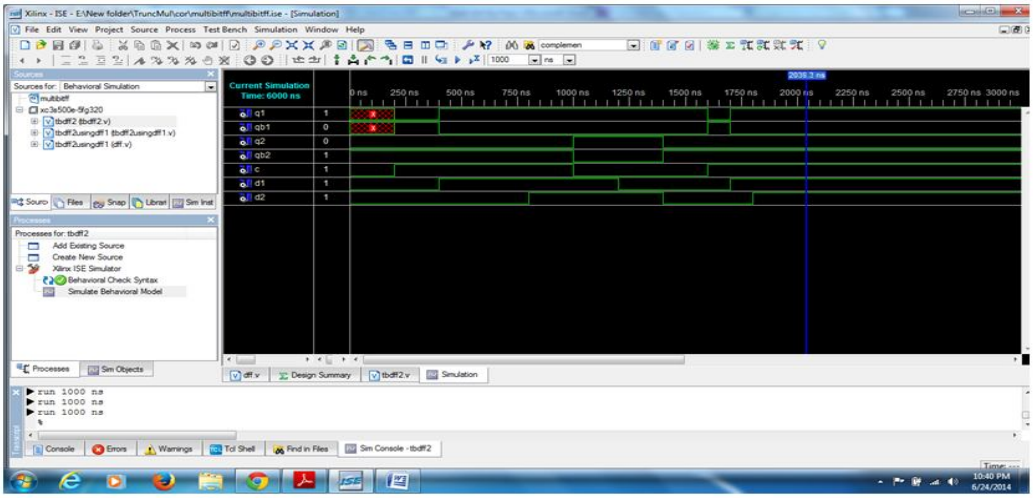


FIGURE 7. SIMULATION RESULT FOR THE MULTIBIT FLIP-FLOPS

VI. CONCLUSION

The algorithm of flip-flop replacement has proposed for power reduction in digital integrated circuit design in this paper. The flip-flop replacement procedure mainly depends on the combination table, that records the relationships among the flip-flop types. The concept of pseudo type helps to enumerate all possible combinations in the combination table. The power reduction and minimizing the total wire length are the cost effectives are considered to the cost function. Through the proposed algorithm we can achieve a balance between power reduction and wire length reduction. Our algorithm can maintain a reasonably good performance of power and wire length reduction even in the integrated Circuit having larger number of flip-flops.

VII. ACKNOWLEDGEMENT

I would like to thank Our Chairman Dr. Sucharitha, the Principal Dr.Mallikarjuniah, Dr. V. Thrimurthulu, Head of the Department, Chadalawada Ramanamma Engineering College and my special thanks goes to Mr. G. Rajesh, Associate professor in CREC for guiding me to bring this paper successfully.

REFERENCES

[1] W. Hou, D. Liu, and P.-H. Ho, "Automatic register banking for low-power clock trees," in *Proc. Quality Electron. Design*, San Jose, CA, Mar. 2009, pp. 647–652.
[2] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, and Q. Wang, "Power-aware placement," in *Proc. Design Autom. Conf.*, Jun. 2005, pp. 795–800.
[3] P. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, May 1998.
[4] D. Duarte, V. Narayanan, and M. J. Irwin, "Impact of technology scaling in the clock power," in *Proc. IEEE VLSI Comput. Soc. Annu. Symp.*, Pittsburgh, PA, Apr. 2002, pp. 52–57.
[5] H. Kawagachi and T. Sakurai, "A reduced clock-swing flip-flop (RCSFF) for 63% clock power reduction," in *VLSI Circuits Dig. Tech. Papers Symp.*, Jun. 1997, pp. 97–98.
[6] P. Godowsky, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, May 1998.
[7] W. Hou, D. Liu, and P.-H. Ho, "Automatic register banking for low-power clock trees," in *Proc. Quality Electron. Design*, San Jose, CA, Mar. 2009, pp. 647–652.
[8] D. Duarte, V. Narayanan, and M. J. Irwin, "Impact of technology scaling in the clock power," in *Proc. IEEE VLSI Comput. Soc. Annu. Symp.*, Pittsburgh, PA, Apr. 2002, pp. 52–57.
[9] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, and Q. Wang, "Power-aware placement," in *Proc. Design Autom. Conf.*, Jun. 2005, pp. 795–800
[10] H. Kawagachi and T. Sakurai, "A reduced clock-swing flip-flop (RCSFF) for 63% clock power reduction," in *VLSI Circuits Dig. Tech. Papers Symp.*, Jun. 1997, pp. 97–98.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

BIOGRAPHY

K.SIVAPRASAD
Post Graduate Student
CREC, Tirupati, India.

He received B.Tech., degree in Electronics and Instrumentation Engineering from JNTUA University, Andhra Pradesh, India in the year of 2011, Currently doing M.Tech., in Chadalawada Ramanamma Engineering college, from JNTUA University, Andhra Pradesh, India. His research interests include VLSI and Nano technologies. He is also a member of Indian Society of Electrical and electronics Engineers.

GUNDLAPALLE RAJESH

Associate Professor
Dept. Of Electronics & Communication Engineering
Chadalawada Ramanamma Engineering College, Tirupathi
Gundlapalli rajesh is currently working as associate professor in the department of electronics and communication engineering atgineering chadalawada ramanamma engineering college, near tirupati, India. He is nearly 10 years of teaching experience. His extencive education includes B.Tech., from Jawahar Lal Technological University, Hyderabad, India. In additional to this, He is making research in the field Medical Image Processing.

Dr.V.THRIMURTHULU
M.E.,Ph.D.,MIETE.,MISTE.
Professor & Head of ECE Department

He received his Graduation in Electronics & Communication Engineering AMIETE in 1994 from Institute of Electronics & Telecommunication Engineering, New Delhi, Post Graduation in Engineering M.E specialization in Microwaves and Radar Engineering in the year Feb, 2003, from University College of Engineering, Osmania University, Hyderabad., and his Doctorate in Philosophy Ph.D. from Central University, in the year 2012. He has done his research work on Ad-Hoc Networks.
He has published eight papers in various International Journals, presented three papers in International Conferences, two papers presented in National conferences. He has attended 4 International workshops, many seminars. His areas of interest are Communication Systems, Cellular and Mobile Communication, Wireless Communication, Antenna designing and Microwave Devices. He is a Life member of Institute of Electronics & Telecommunication Engineers (MIETE), Indian Society for Technical Education (ISTE), and The International Association of Engineers (IAENG).
He has total 22 years of Academics & Industrial Experience. He has 12 years teaching experience in various Engineering colleges in Hyerabad, A.P. He has been worked as a Professor in ECE Department in K L University. He has worked as a Professor in ECE department in Swathi Institute of Technology & Sciences, Hyderabad. in the year 2011to 2013. He has worked an Associate Professor in the Department of Electronics and Communication Engineering, KG Reddy College of Engineering & Technology,Chilkur(v), Moinabad(Mon), RangaReddy(Dist),A.P. from 2010-11. He had an 11 years Industrial experience as a System Integrator, Customer Support Engineer, and TRC Manager in various Service organizations in Hyderabad, A.P.