IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 4, Ver. III (Jul-Aug. 2014), PP 52-60
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Analysis of different legalisation methods for unequal sized recursive Min-cut placement

Kalimidi Yugandhar[1], G. V. Marutheswar[2]
[1](Research Scholar, Department of Electrical Engineering, Sri Venkateswara University College of Engineering, Tirupati, India)
[2](Professor, Department of Electrical Engineering, Sri Venkateswara University College of Engineering, Tirupati, India)

Abstract: *In this paper, different legalisation methods are examined for a new method of min-cut partitioning for VLSI placement. Traditional min-cut placers divide the given circuit into equal sized sub-circuits at each partitioning level. This paper uses a new partitioning method called unequal sized partitioning in which instead of making the sub-circuits size equal, the size of sub-circuits are made unequal. This paper analyses the placement results to minimize the total wire length of unequal sized partitioning for different legalisation schemes. Firstly an introduction to unequal sized recursive partitioning and legalisation schemes are presented. Then these schemes are applied to MCNC benchmark circuits for different partition ratios ranging from 0.1 to 0.9. The results prove that all the legalisation schemes do not give optimal wire lengths for equal sized partitioning. Each circuit has optimal wire length at different partition ratio for a different legalisation method. Finally this paper suggests the need for unequal sized partitioning which improves the wire length significantly as compared to the conventional equal sized partitioning.*

Keywords: *Legalisation, Min-cut algorithm, Partition ratio, Placement, Unequal sized partitioning*

I. Introduction

VLSI placement is an important step in the physical design process. The objective of the VLSI placement is to assign each circuit element to a unique location of the chip. Broadly this process of assigning circuit elements to specific locations can be classified into two types, namely global placement and detailed placement. Simulated Annealing [1], Min-cut [2], Numerical optimization [2], Force-directed [3], and Evolution based [4] are some of the placement algorithms used to solve the VLSI placement problem [5]. Constructive placement algorithms use simple methods to generate the rough initial placements during the global placement. Placement by partitioning or Min-cut algorithm is an important constructive placement algorithm used extensively in the industry.

Min-cut algorithm divides the given circuit into sub-circuits at each level of partitioning and assigns the sub-circuits to specific chip areas. Sometimes, the allotted number of modules to a sub-circuit may exceed its allotted area capacity causing the overlapping of the modules. This is a typical violation of area constraint. The process of removing of this violation and making an overlap free placement is called Legalisation. Many methods were proposed in the literature such as ripple move [6] [7] [8], cell shifting [9], grid wrapping [10] and slot assignment [6] [7] to remove the area congestion. In this paper, an algorithm which shifts the cells from an excess partition to the other partition in accordance with the optimisation requirements is used.

Dragon [11], FengShui [12], NTUPlace2 [13] and Capo [14] use min-cut partitioning algorithm for VLSI placement. They use either recursive balanced bisection or quadrisection [15]. The balanced min-cut bisection at each partitioning level may provide an optimal partitioning, but at the placement level an optimal partitioning may not give an optimal placement. All of this placement tools ignore the unequal sized partitioning and assume that the balanced min-cut partitioning give an optimal partitioning at all times for all the circuits. This paper employs a new method of partitioning called unequal sized partitioning to solve VLSI placement problem [5]. Since the problem of VSLI placement problem is NP complete [16], it cannot be denied that unequal sized partitioning may provide better optimal solution. The objective of this work is to examine the effect of different legalisation schemes for unequal sized recursive partitioning.

This paper is organised as follows. In section 2, a new partitioning approach called unequal sized partitioning is introduced with illustrations. Different possible implementations of unequal sized partitioning are discussed in detail in this section. The section 3, gives a brief introduction of different legalisation schemes

which are employed in the proposed algorithm. In section 4, the hypergraph implementation of circuit is discussed in detail in addition to the benchmark circuits, software and operating system used to carry out the experiment. In section 5, the experimental results are reported which show a significant improvement of placement results with unequal sized partitioning. And finally the last section concludes the work emphasizing the necessity of unequal sized partitioning as an alternative approach to equal sized partitioning.

II. Unequal sized recursive partitioning

Traditional min-cut placers divide the given circuit recursively into equal sized partitions till each

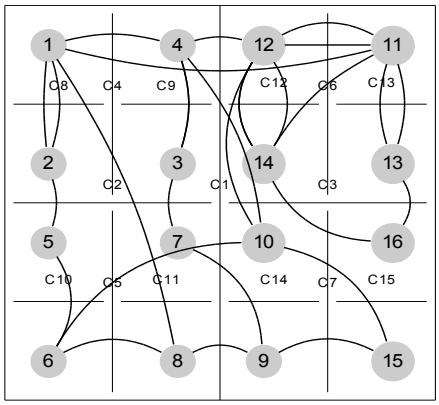

Fig. 1 Conventional equal sized partitioning

module is mapped to a particular location of the chip. The Fig. 1 shows a typical example of min-cut placement by equal sized partitioning. The circuit is partitioned with equal number of modules and equal areas with minimum net cut at each level of partitioning. The cutline C1 divides the modules into two sets of eight modules each with the number of interconnections between them as 6. This is the first level of partitioning. In the second of level of partitioning the horizontal cut-lines C2 and C3 divide the blocks obtained in the previous level into blocks of four modules each. The process of division is continued to these sub-blocks further till each module is mapped to unique location.

Contrary to the traditional method of equal sized partitioning, the modules are divided into unequal sized partitions and mapped to unequal sized chip areas as shown in Fig. 2 for unequal sized recursive partitioning. The number of modules allotted to the block B1 is shown as N(B)×partition ratio. If partition ratio is defined as 0.4, then the left smaller block B1 gets $0.4\times N(B)$, while the right bigger block gets $0.6\times$ N(B). In the next partitioning level, the block B2 is considered for partitioning leaving its complementary block B1. The algorithm recursively apply the same partition ratio to this block B2 and divides into B21 and B22, where B21 gets 0.4$\times$ N(B2) modules and B22 gets $0.6\times N(B2)$ modules. This process is repeated recursively till all the modules are allotted to unique locations of the chip.

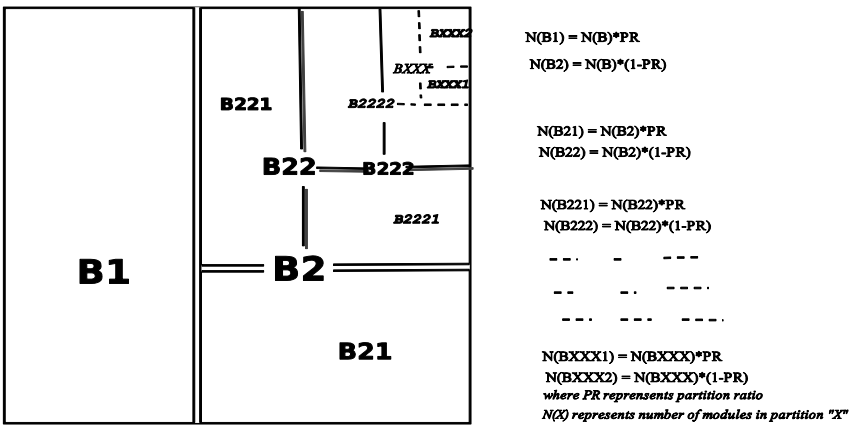

Fig. 2 Unequal sized recursive partitioning

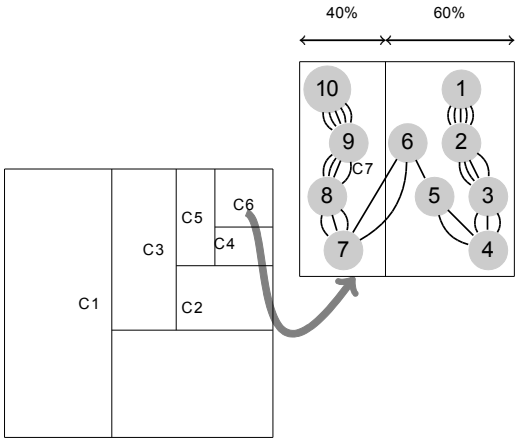

Fig. 3 A chip layout partitioned alternatively with vertical and horizontal cutlines with partition ratio of 0.4

The Fig. 3 represents a typical layout region with alternate vertical and horizontal cut-lines for unequal sized partitioning. Each of the cut-lines divides the partition area repeatedly into two partitions of areas of 40% and 60%. At each partitioning step, the total number of modules are divided into partitioning sizes of 40% and 60% and the modules are assigned to the respective partitioning areas. From this figure, it can be observed that we reached a stage where circuit to be partitioned have total number of ten modules. The vertical cut line C7 cuts the partitioning area into two partitions of areas of 40% and 60% of total area. The modules are divided into two partitions of four modules and six modules each such that the cut size is two nets.

The Fig. 4 represents different unequal partitioning strategies for a circuit having ten modules. Fig. 4(a) shows conventional partitioning for VLSI cell placement where partition areas and sizes are equal sized. The Fig. 4(b) represents a typical sub-circuit of unequal partition sized placement. Here the sub-circuit area is divided into two partitions such that the left one has 40% of the total number of modules of the sub-circuit and

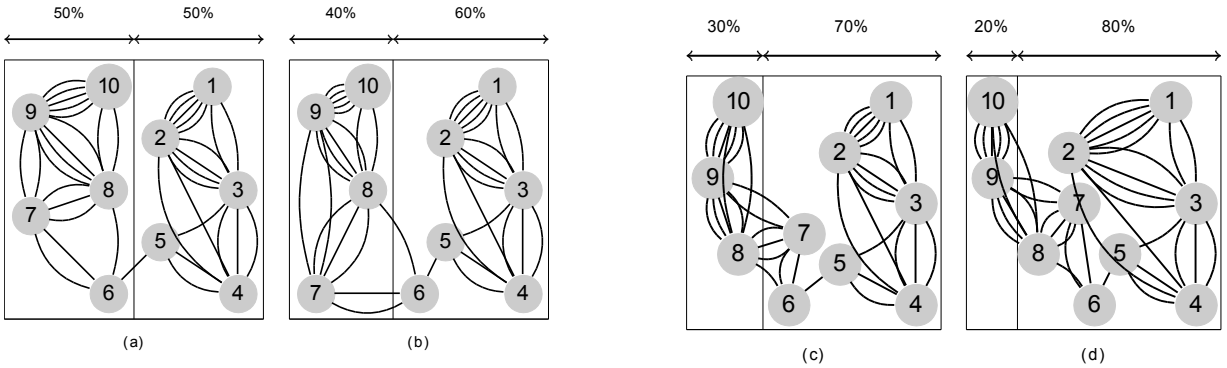

Fig. 4 A circuit with ten modules partitioned with different partition ratios of 0.5, 0.4, 0.3 and 0.2

the right one accommodates the remaining 60% of the modules. The cut line is placed at 0.4 times the length of x-axis of the sub-circuit from left most bottom corner. Here the cut size is two nets. And for a horizontal cut the cutline is placed at 0.4 times the length of the y-axis of the sub-circuit from the left most bottom corner. The Fig. 4(c),(d) represent 30-70 partition and 20-80 partition, and the process of placement is same as above with variation in partition ratios.

III. Legalization

The Fig. 5 shows a typical placement violation after an unequal sized min-cut partitioning step. The maximum capacity of the bottom block is 14 modules, but the placement algorithm allotted 18 modules with an excess of four modules. This kind of situations usually happens after global placement, where the primary objective of the placement algorithm during global placement is to minimize the wire length without considering congestion. It is necessary to remove this placement violation before proceeding to the next level of partitioning. A separate procedure is incorporated in the algorithm to remove this violation to make an overlap free placement.

The process of legalisation usually takes place in two steps

1. Selection of module from overlapped partition
2. Shifting of selected module to the other partition

The selection of module for shifting is to be done keeping the optimisation in mind. For some of the circuits, minimization of wire length is given higher priority. For such circuits, the module from the overlapped partition which can reduce the wire length drastically is selected for shifting to other partition. For those circuits whose objective is the minimization of congestion, the module which has high fan-out is considered for shifting

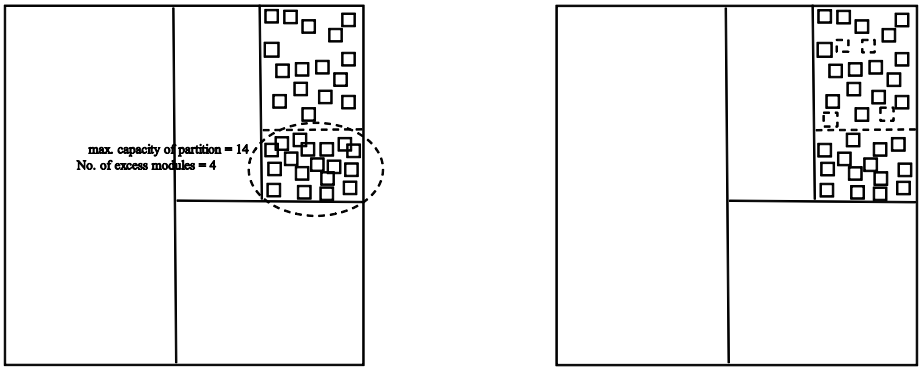

Fig. 5 An illustration of placement constraint

to other partition. Similar is the case with timing critical circuits, where the selection of module is done keeping in mind the change in delay caused by shifting.

The modules which are loosely connected or in other words those modules having least number of connections to other modules in that partition is considered for shifting, so that increase in total wire length of the circuit is not significant. Some legalisation method selects those modules which are strongly connected or having high number of connections in other partition is considered for shifting, which also results a slight increase in overall wire length. Still some legalisation techniques use an hybrid of the above mentioned two techniques to reduce the overall wire length.

```
Procedure{Legalisation } {}
        If {Legalisation==Leastcon}
                Initialise Leastcon=MAX
                For{Each gate in this partition}
                        Check the fanout value
                        If {fanout<Leastcon}
                            Leastcon = fanout
                            Flag this gate
                        EndIf
                EndFor
        Else
        If {Legalisation==Mostlen}
                Initialise Mostlen=MIN
                For{Each gate in this partition}
                        Check the wire length by moving this gate to the other partition
                        If {wirelength > Mostlen}
                            Mostlen = wirelength
                            Flag this gate
                        EndIf
                EndFor
```

```
        Else
                Initialise Leastlen=MAX
                For{Each gate in this partition}
                        Check the wire length by moving this gate to the other partition
                        If {wirelength < Leastlen}
                                Leastlen = wirelength
                                Flag this gate
                        EndIf
                EndFor
        EndIf
        EndIf
EndProcedure
```

Fig. 6 Legalisation algorithm

The algorithm shown in Fig. 6 Checks the maximum capacity of the partition at each partitioning level to ensure that there is no placement violation. The maximum capacity of any partition is defined by its total area times the number of gates placed per site. A violation is reported if the partitioning step yields a result that has more gates in a partition than its maximum capacity.

Different strategies are implemented in the Algorithm to look after the legalization. Each strategy selects a module in the current partition that will be moved to the adjacent partition to fit the capacity constraints. The strategies are:

1) **Least Connected:** This strategy selects the module that is least connected to other gates and pads. This strategy is efficient especially for larger benchmarks.
2) **Least Decrease in Wire length:** This strategy selects the module that results in the least decrease in wire length when shifted from the current partition to the adjacent one.
3) **Most Decrease in Wire length:** This strategy selects the module that results in most decrease in wire length when shifted from the current partition to the adjacent one.

IV. Implementation details

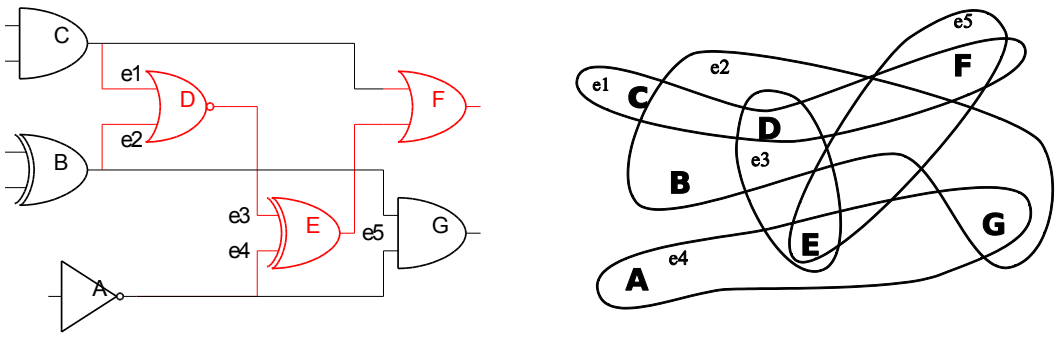

Fig. 7 A simple circuit with seven logic gates and its hypergraph

In VLSI CAD, the circuits are represented using hypergraphs. A hypergraph is represented by G = (V, E), where V is a set of circuit gates and E is a set of signal nets. A hyperedge e E connects any number of vertices in V. The Fig. 7 shows a circuit with seven logic gates and five edges and its hyper graph. The gates are represented as a set vertices V ={A,B,C,D,E,F} with each vertex v ∈ V having a size s(v) and the interconnection wires are represented as a set of hyper-edges E= {e1,e2,e3,e4,e5} with each hyper-edge e ∈ E having a weight w(e).

The objective of the min-cut algorithm is to partition the given circuit into two sub-circuits such that the interconnections between the two partitions are minimum. This task of partitioning can be easily implemented using the hypergraph partitioning. The Half-Perimeter Wire length (HPWL) net model is used in

the algorithm wire length estimation. This method calculates the perimeter of the bounding box of the modules under consideration and approximates the wire length by the half-perimeter. HPWL is most effective for 2-pin and 3-pin nets and since most of the nets in any circuit are either 2 or 3 pin nets this method is widely adopted in the industry. The algorithm shown in Fig. 6 is implemented in C++. Eclipse, an integrated development environment (IDE) is used to develop and carry the experiment. The operating system Ubuntu 12.04.4 LTS is used to run Eclipse. The hardware used include a processor of 4x Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz with memory of 4041MB. MCNC benchmark circuits *fract, primary1, industry1* and *biomed* are used to test our hypothesis. The Table 1 shows the characteristics of the benchmark circuits used.

Table 1 Characteristics of MCNC Benchmarks

Circuit	#Cells	#Nets	#Pins	#Rows
fract	125	163	454	6
Primary1	752	1266	3303	16
Industry1	2271	2479	8024	64
Biomed	6417	5742	26947	46

V. Experimental results

In **Error! Reference source not found.** Table 2 the total wire lengths of MCNC benchmark circuit *fract* for various partition ratios are compared for different legalisation methods. It clearly shows that the three different legalisation methods yield optimal wire lengths at different partition ratios. The Table 3 compares the wire lengths for benchmark *primary1*, the results show that optimal wire lengths do not vary with the legalisation scheme and partition ratios. The Table 4 compares the wire lengths for benchmark *industry1*, the results show that optimal wire lengths vary with the legalisation scheme and partition ratios. Similar is the case with *biomed*. Also from the tables, it can be observed that the improvement in wire length is significant with all of the benchmark circuits with an average improvement of **1.26%.**

Table 2. Comparison of legalisation techniques for "fract"

PARTITION RATIO	FRACT		
	LEASTCON	LEASTLEN	MOSTLEN
0.1	1126.76	1175.75	1080.22
0.2	989.59	1023.88	960.19
0.3	989.59	1023.88	960.19
0.4	1023.88	1072.87	1045.93
0.5	**967.54**	**979.79**	940.61
0.6	967.54	979.79	940.61
0.7	994.49	1033.68	**896.51**
0.8	1011.63	1072.87	965.09
0.9	1011.63	1072.87	965.09

Table 3 Comparison of legalisation techniques for "Primary1"

PARTITION RATIO	PRIMARY1		
	LEASTCON	LEASTLEN	MOSTLEN
0.1	12709.19	14540.15	11017.37
0.2	12519.45	13357.46	11093.27
0.3	11485.39	12538.43	10059.29
0.4	10818.15	12089.38	9992.79
0.5	**9901.09**	**10445.01**	**9246.49**
0.6	12127.33	12231.68	10008.69
0.7	10950.96	11928.11	9546.91
0.8	11197.62	11874.35	9382.47
0.9	11671.96	13180.37	9733.49

Table 4. Comparison of legalisation techniques for "Industry1"

PARTITION RATIO	INDUSTRY1		
	LEASTCON	LEASTLEN	MOSTLEN
0.1	58227.01	65392.73	51238.38
0.2	50662.85	57853.86	44600.76
0.3	40701.67	47301.34	38004.25
0.4	38437.48	40135.62	36397.81
0.5	36644.47	37292.74	35977.23
0.6	36087.91	**37175.73**	**34668.04**
0.7	**35623.05**	44739.91	38143.39
0.8	11197.62	11874.35	9382.47
0.9	11671.96	13180.37	9733.49

Table 5. Comparison of legalisation techniques for "biomed"

PARTITION RATIO	BIOMED		
	LEASTCON	LEASTLEN	MOSTLEN
0.1	174566.37	205029.67	147128.12
0.2	129010.23	153902.95	114007.53
0.3	93787.59	119473.54	86107.77
0.4	77436.42	82952.91	76336.73
0.5	70726.49	**71656.72**	70351.51
0.6	**69792.65**	77313.83	**69994.56**
0.7	79520.43	115583.15	76827.08
0.8	111761.27	158388.26	100594.88
0.9	148436.94	169770.98	119304.08

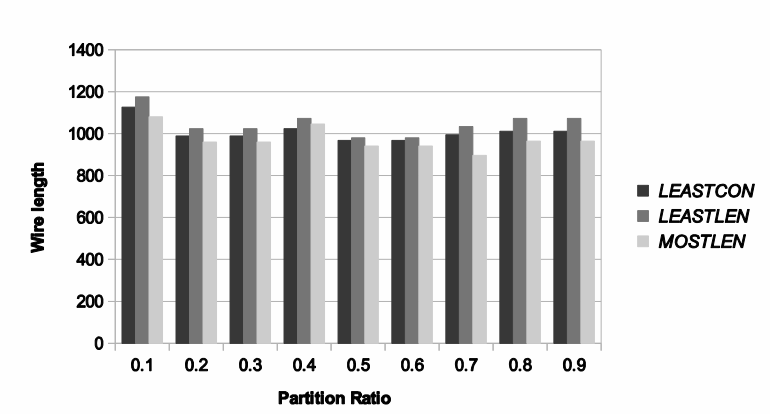

Fig. 8 Wire length variation for different partition ratios for "fract"

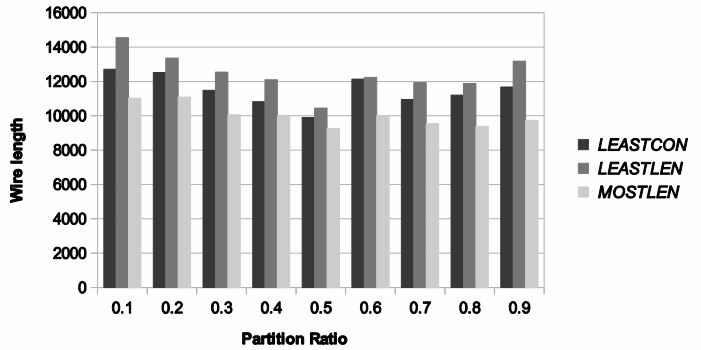

Fig. 9 Wire length variation for different partition ratios for "primary1"

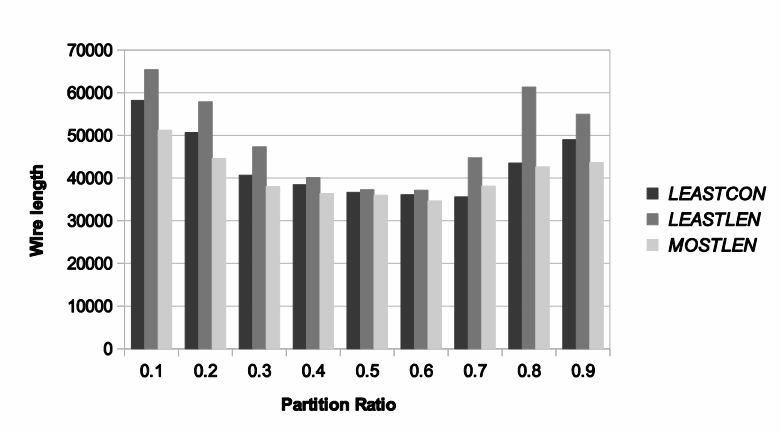

Fig. 10 Wire length variation for different partition ratios for "industry1"

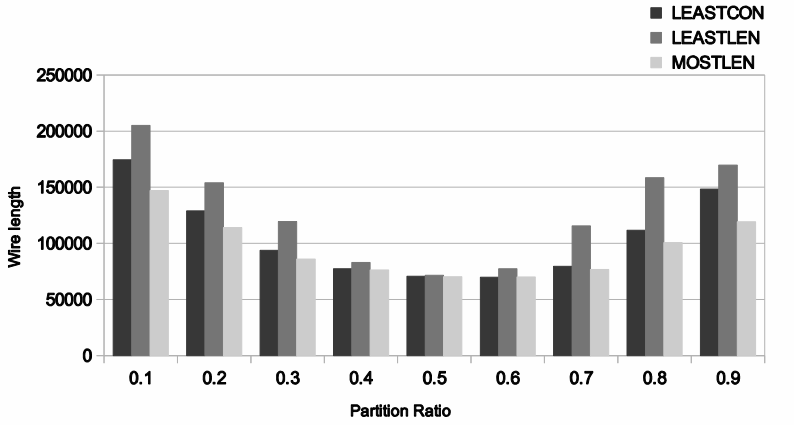

Fig. 11 Wire length variation for different partition ratios for "biomed"

The Fig. 8 to Fig. 11 shows the variation of wire lengths for various partition ratios for MCNC benchmarks. From this figures, it can be inferred that each of the circuit behave differently without any uniform variation to different partition ratios. The experimental results show that some of the circuits yield optimal wire lengths for equal sized partitioning while other circuits yield optimal wire length for unequal sized partitioning.

VI. Conclusion

This paper examines the behavior of the circuits for different legalisation schemes for unequal sized recursive partitioning. Firstly modeling of unequal sized partitioning is discussed and then these models are applied to MCNC benchmark circuits. These results show that equal sized partitioning do not always give optimal placement results. The minimal wire length can occur at any partition ratio and purely a characteristic of the circuit topology. These results necessitate checking all the partition ratios to find which partitioning strategy gives optimal solution.

References

[1] Scott. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies.," Journal of statistical physics 34.5-6 , pp. 975-986, 1984.

[2] Khushro, and Pinaki Mazumder. Shahookar, "VLSI cell placement techniques.," ACM Computing Surveys (CSUR) 23.2, pp. 143-220, 1991.

[3] Maurice, and Jerome M. Kurtzberg. Hanan, "Placement techniques.," Design automation of digital systems 1, pp. 213-282, 1972.

[4] R-M., and Prithviraj Banerjee. Kling, "ESP: A new standard cell placement package using simulated evolution.," In Proceedings of the 24th ACM/IEEE Design Automation Conference, pp. 60-66, 1987.

[5] Melvin A Breuer, "A class of min-cut placement algorithms," In Proceedings of the 14th Design Automation Conference, pp. pp. 284-290, 1977.

[6] Tony F., Jason Cong, Tianming Kong, and Joseph R. Shinnerl. Chan, "Multilevel optimization for large-scale circuit placement.," In Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design, pp. 171-176, 2000.

[7] Tony F., Jason Cong, Tim Kong, Joseph R. Shinnerl, and Kenton Sze. Chan, "An enhanced multilevel algorithm for circuit placement.," In Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design, p. 299, 2003.

[8] Sung Woo, and John Lillis Hur, "Mongrel: hybrid techniques for standard cell placement.," Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design., 2000.
[9] Natarajan, and Chris CN Chu. Viswanathan, "FastPlace: Efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 24.5 , pp. 722-733, 2005.
[10] Zhong, James D. Ma, Suzanne M. Fowler, and Rob A. Rutenbar. Xiu, "Large-scale placement by grid-warping.," In Proceedings of the 41st annual Design Automation Conference, pp. 351-356, 2004.
[11] Maogang, Xiaojian Yang, and Majid Sarrafzadeh. Wang, "Dragon2000: standard-cell placement tool for large industry circuits," In Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design, pp. 260-263, 2000.
[12] Ameya R., Satoshi Ono, and Patrick H. Madden. Agnihotri, "Recursive bisection placement: feng shui 5.0 implementation details.," Proceedings of the 2005 international symposium on Physical design. ACM, 2005.
[13] Zhe-Wei, Tung-Chieh Cheny, Tien-Chang Hsuy, Hsin-Chen Chenz, and Yao-Wen Changyz. Jiang, "NTUplace2: A hybrid placer using partitioning and analytical techniques.," In Proceedings of the 2006 international symposium on Physical design, pp. 215-217, 2006.
[14] Jarrod A., David A. Papa, Saurabh N. Adya, Hayward H. Chan, Aaron N. Ng, James F. Lu, and Igor L. Markov. Roy, "Capo: robust and scalable open-source min-cut floorplacer.," In Proceedings of the 2005 international symposium on Physical design, pp. 224-226, 2005.
[15] Dennis J-H., and Andrew B. Kahng Huang, "Partitioning-based standard-cell global placement with an exact objective.," In Proceedings of the 1997 international symposium on Physical design, pp. 18-25, 1997.
[16] Wilm E. Donath, "Complexity theory and design automation.," In Proceedings of the 17th Design Automation Conference, pp. 412-419, 1980.