Resource Constrained Scheduling using Behavioral Network Graph

Dr.M.Sangeetha[1*],Dr. R. Vani[2] and T. Vijayan[3]
{sang_gok@yahoo.com[1], rvani18@gmail.com[2], tvij16@gmail.com[3]}

Professor, Department of Electronics and Communication Engineering, Bharath Institute of Higher Education and research, Chennai, India[1]
Professor, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Chennai[2]
Asst. Professor, Department of Electronics and Communication Engineering, Bharath Institute of Higher Education and research, Chennai[3]

Abstract. The behavioral modeling in HDL language is converted into Control Flow Graph (CFG). The resources in the HDL program is marked in the Control Flow Graph. Each node in the Control Flow Graph is replaced with state value node. Equivalent Behavioral Network Graph is generated for the Control Flow Graph. The unscheduled Behavioral Network Graph is scheduled according to the resources available. The scheduling is performed for any number of resources. Theoretically it is proved using Finite State Machine in this paper.

Keywords: Estimator, Behavioral network graph, Control Flow Graph, hardware estimation.

1 Introduction

The HDL program in functional level is simulated for functionality verification. The estimation is performed for the behavioral description. The estimation[1][2][3][4] will be accurate only if same intermediate format is used for High Level Synthesis and Logic Synthesis. The Behavioral Network Graph is used to represent the logic synthesis. The generated hardware is an optimized one. The state value node(STN) is used to represent each node. State cut variable SCi is associated with each state value node (STN). The algorithm is represented in [5][6]. The unscheduled control BNG is obtained. This unscheduled graph represents the worst case. It is scheduled according to the availability of resource. The behavioral description is transformed into Control Flow Graph(CFG). The nodes in the Control Flow Graph is replaced with state value node. The state value node consists of register, multiplexer and wire. The '1' in state cut variable is realized as register and '0' in state cut is replaced with wire. The number of register is estimated according to the state cut. If one adder is available, the state cut is placed every one adder.

The AND gate is estimated according to the fork nodes in the Control Flow Graph. OR gate is estimated according to the join nodes in the graph.

The designed estimator calculates register, AND gate and OR gate.

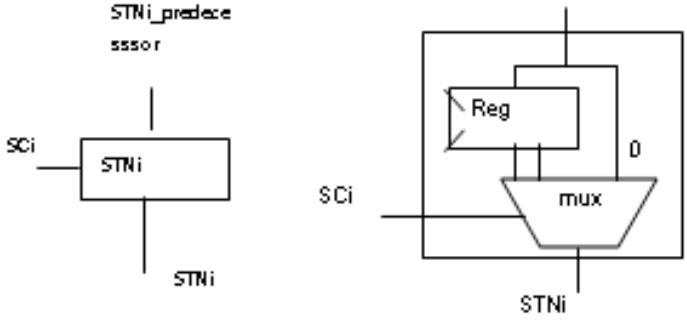


Figure 1 State Value Node

I3CAC 2021, June 07-08, Chennai, India

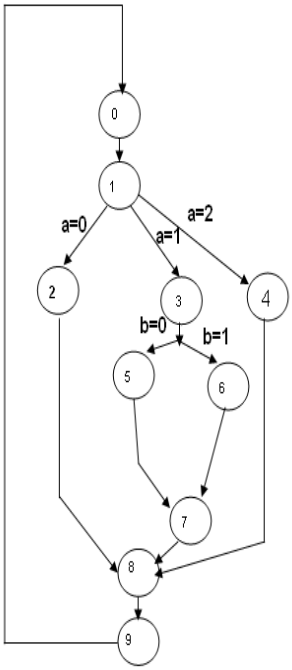
DOI 10.4108/eai.7-6-2021.2308871

2 Previous Work

For Resource constrained scheduling and timing constrained scheduling Gang Wang et al used ant colony optimization techniques. The scheduling is done in behavioural description. The Data Flow Graph is used for each basic block. The scheduling performed is timing constrained scheduling(TCS) or resource constrained scheduling(RCS). With given deadline the number of resources is calculated. Presynthesis optimization is performed by Rafael Ruiz-Sautua et al does efficient High Level Synthesis of DFG by addition, multiplication and logic operation. Homogenous specification is used in HLS algorithm. Different type, width and format is used in every cycle. Bergamaschi used Behavioral Network Graph for High level Synthesis internal representation. Estimates the control construct from behavioral description. The obtained final behavioural network graph is the complete RTL design. Stitt and vahid has done traditional hardware/software partitioning. Control Flow analysis and data flow analysis is used for optimization. Functional partitioning is performed in hardware/software partitioning and hardware/hardware partitioning by Vahid. Multitasking overhead is obtained when software part is partitioned, heuristic algorithm is developed. Wonyong Sung et al presented optimization technique which reduces the code and data size, Optimization chances are increased, large grain is formed from fine grain cluster for nodes.

3. Cost Estimation

The Control Flow Graph is obtained from the behavioral description. The nodes which have resources are taken into consideration in CFG. Each node in Control Flow graph is represented with State Value Node (STN). The fork node is replaced with two inputs AND gate with a State Value Node and join nodes with OR gate and State Value Node. A simple nodes with IF within Case construct is shown in Figure 2. The number of nodes taken is 10. The node 1 represents the CASE statement, its successor are 2,3,4. Node 3 represents the IF statement and its successor are 5 and 6. Node 7 is end for the IF statement and node 8 is end of CASE statement. The condition for CASE statement is ‘a=0’, ‘a=1’ and ‘a=2’. The condition for IF statement is ‘b=0’ and ‘b=1’. The SCn represents the state cut variable. Each node has state cut variable. The state cut variable determines the realization of register or wire. If value in state cut variable is ‘0’ the state value node is realized as wire. If the value in state cut variable is ‘1’ the state value node is realized as register. The value of state cut variable depends on the number resources available. If resource available is one, state cut is placed after single resource.



IF within Case construct	
node	successor
0	1
1	2,3,4
2	8
3	5,6
4	8
5	7
6	7
7	8
8	9
9	0

Figure 2 CFG for IF within Case Construct

Each node in Control Flow Graph in Figure 2 is replaced with State Value Node, the fork node and join node is placed in control BNG according to the algorithm discussed in [5][6].

4.Matrix representation of Control Flow Graph

The consecutive nodes are identified in two dimensional matrixes. The input to matrix is Control Flow Graph. The nodes are represented in rows of the matrix and successors with column of the matrix.

Table 1: Matrix representation of CFG

	0	1	2	3	4	5	6	7	8	9	10
0		1									
1			1	1	1						
2									1		
3						1	1				
4									1		
5								1			
6								1			
7									1		
8										1	
9											1
10	1										

Table 2: Expanded Matrix Representation for Estimation of Gates

	0	1	2	3	4	5	6	7a	7b	8a	8b	8c	9	10
0		1												
1			&	&	&									
2										1				
3						&	&							
4												1		
5								1						
6									1					
7											+			
8														
9a													+	+
9b														+
10	1													

The successor is checked if a node has successor it is represented in matrix as ‘1’ and remain columns will be blank.The matrix representation of CFG is shown as in Table 1 for the Figure 3. Figure 2 represents the node representation. The successors in node 1 is three ‘1’which is replaced with AND Gate. For node3 it has two successors which is replaced with AND Gate. The node which has more than two ‘1’s in the column, the columns are expanded according to the number of ‘1’s and replaced with ‘+’ which shows number of inputs to OR gate as shown in table 3.

The ‘&’ symbol is used to represent AND Gate and ‘+’ the number of inputs to OR gate. We can estimate the inputs to OR gate and also number of two input OR gate. Figure 3 represents the unscheduled BNG[5][6].

5 Algorithm for generation of Control BNG

Each node in the Control Flow graph is replaced with state value node. The AND gate is placed in fork nodes and OR gate is placed instead of join nodes. The Control BNG is generated using the following steps.

1. SCi variable is associatedwith each node
2. Nodes in CFG is traversed through entire graph. A node with single predecessor and single successor is replaced with state value node. State cut variable is associated with each state value node STNi.
3. STNij is formed from join nodes for each predecessor edge, all state value node STNij is connected to single OR gate. STNi is the output of OR gate.
4. Fork nodes(multiple successor edges), a state value node is created which is then connected to the AND gate. Two input AND gate is use. One input will be from fork node(STNi) and other input will be its corresponding successor edge. STNij is the output of each AND Gate.
5. State value node is connected as in Control Flow Graph. For join nodes extra state value node is created. State cuts are used in all state value node.

The generated Behavioral Network Graph represents all possible schedule. Since it has all possible schedule, the state cut can be placed at any position which is realized as register. Based on available hardware the state cut is placed.

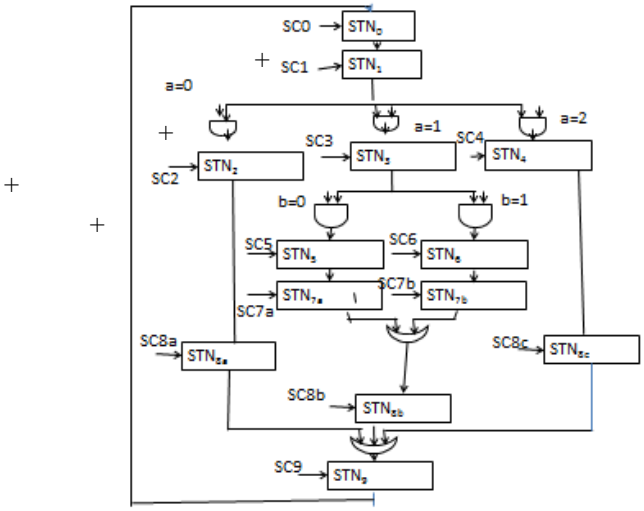


Figure 3 Unscheduled Controls BNG

The four adders are available in the Unscheduled BNG. So four state cut has to be placed. The theoretical representation for single adder is shown in the Figure 4.

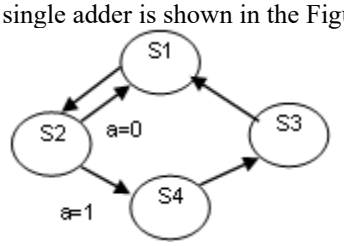


Figure 4 FSM for single adder/Subtractor

The estimator estimates four register, two '2' input AND gate and one '2' input OR gate as shown in Figure 5. The resources are shown as in unscheduled Control BNG as in Figure 3. The state cut is placed according to the hardware resources available. When state cut is placed it is realized as register else it is realized as wire. The final BNG is as shown in Figure 5.

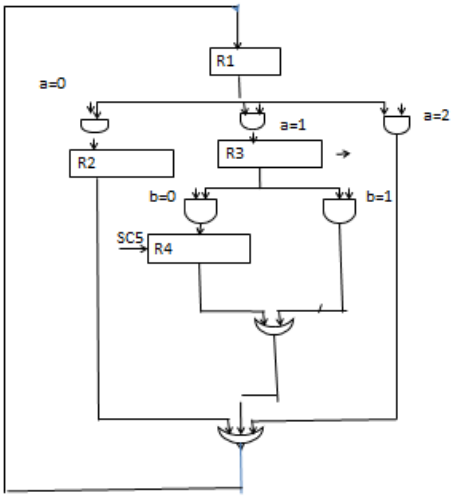


Figure 5 FSM implementation using BNG for single adder/Subtractor

The state cut is placed for every two adders. The FSM representation is shown in below Figure 6. The estimated registers are '3' and two '2' input AND gate and one '2' input OR gate.

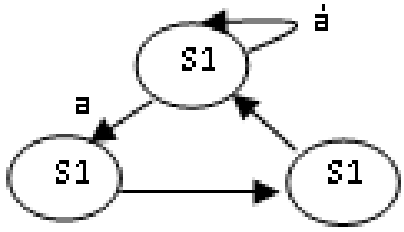


Figure 6 FSM for two adder/ Subtractors

6 Result and Discussion

The number of nodes is given as an input in the designed estimator. The successor for each node is given as an input as shown in Figure 7. Node 1represents the CASE statement and Node 3 represents the IF Construct. An example of IF within CASE statements is fed as an input as shown in Figure 2. The unscheduled Control Behavioral Network Graph is generated. The estimator needs the nodes with resources and requires number of resources available. The state cut is placed according to the number of resources available. Four state cut is placed which results in registers as shown in Figure 5. 5 two input AND gate, two input OR gate and three input OR gate is resulted. The final RTL is obtained as shown in Figure 5. The final Control BNG is obtained as shown in Figure 8.

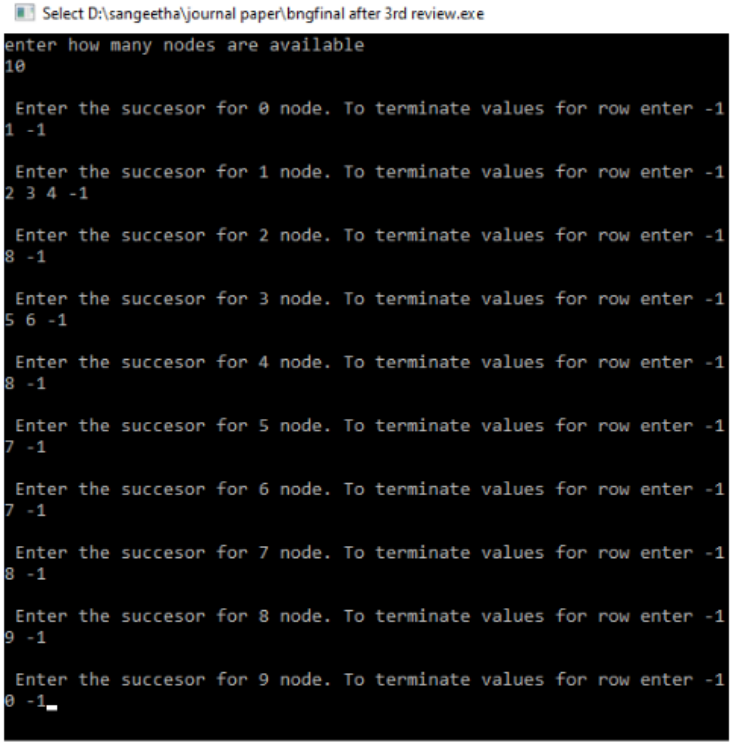


Figure 7 CFG for IF within Case Construct

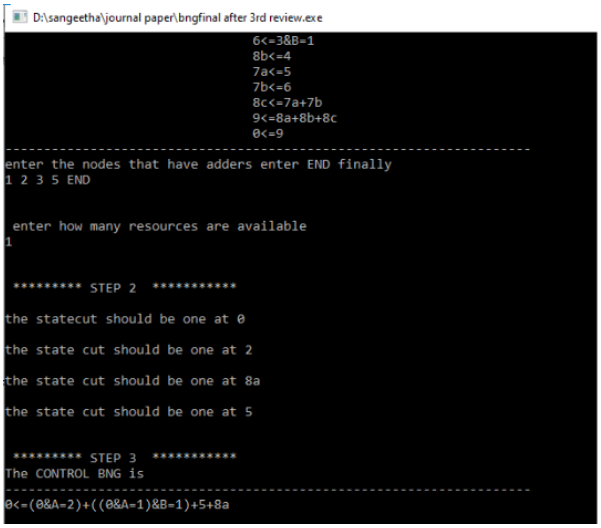


Figure 8 Control BNG

7 Conclusion

Hardware estimation is done using the Control Flow Graph. The concept state value node is used to obtain Unscheduled Behavioral Network Graph from Control Flow Graph which is given as an input in Figure 7. The Scheduling is performed in the unscheduled Behavioral Network Graph. The final gate level representation for the Control Flow Graph is obtained as in Figure 8. The scheduling is done for single resource availability. The estimator is verified theoretically using Finite State Machine. The number of two input AND gate, number of OR Gate. The designed estimator is verified using Finite state Machine.

Conflicts of interest

The authors should declare no conflicts of interest.

References

[1] R. A. Bergamaschi and S. Raje,Control flow versus data-flow-based scheduling: Combining both approaches in an adaptive scheduling system, IEEE Trans. Very Large Scale Integration Syst.,1997; 5(1):82-100.

[2] S. Davison, D. Landskov, B. Shriver, and P. Mallet, Some experiments in local microcode compaction for horizontal machines,IEEE Trans. Comput., 1981; C (30), pp. 460–477.

[3] A. C. Parker, J. T. Pizarro, and M. Mlinar, MAHA: A program for datapath synthesis, in Proc. 23rd ACM/IEEE Design Automat. Conf., June 1986, pp. 461–466.

[4] B. M. Pangrle and D. D. Gajski, State synthesis and connectivity binding for microarchitecture compilation, in Proc. IEEE Int. Conf. Computer-Aided Design, ACM/IEEE, Nov. 1986, pp. 210–213.

[5] P. G. Paulin and J. P. Knight,Force-directed scheduling for the behavioral synthesis of ASIC's,IEEE Trans. Computer-Aided Design, 1989; 8, pp. 661–679.

[6] M.Sangeetha, J.Raja Paul Perinbam :Estimation of control logic for binary synthesis, 2008 International Conference on Signal Processing, Communications and Networking,Chennai,India. DOI: 10.1109/ICSCN.2008.4447237.

[7] Sangeetha M., Tharini C., RajapaulPerinbam J: Equivalent Design Representation and Transformation for High Level Synthesis, online proceedings of 2nd IEEE RTAS Workshop on Model-Driven Embedded Systems (MoDES '04) conference conducted in Ontario, Toronto, Canada, May 2004.

[8] M. Xu, F. J. Kurdah: Area and Timing Estimation for Lookup Table Based FPGAs, European Design and Test Conference, 1996.

[9] J. Gerlach W. Rosenstiel: A Scalable Methodology for Cost Estimation in a Transformational HighLevel Design Space Exploration Environment. Design Automation and Test in Europe, Paris, February 1998.

[10] A. Srinivasan, G. D. Huber and D.P. LaPotin: Accurate Area and Delay Estimation from RTL Descriptioons, IEEE transactions on VLSI systems, 1998;6(1), pp. 168-172.

[11] F. Ferrandi, F. Fummi, E. Macii, M Poncino, D Sciuto, Power Estimation of Behavioural Descriptions Design Automation and Test in Europe, Paris, February 1998.

[12] R. A. Bergamaschi,Bridging the domains of high level and logic synthesis" in Proc. Trans. Computer-Aided Design, 2002;21, pp.582–596.