Middle-East Journal of Scientific Research 22 (6): 822-828, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.22.06.21953

A Temperature-Aware Scheduling with Incremental Binding and Floorplanning for HLS

[1]A. Punitha and [2]M. Joseph

[1]Senior Assistant Professor, Chettinad College of Engineering and Technology, Tamil Nadu, India
[2]Professor, St. Joseph College of Engineering and Technology, Tamil Nadu, India

Abstract: Modern Integrated Circuits' reliability and performance mainly depends upon its temperature and power due to the continuous process scaling. The IC peak temperature depends upon the power density of the IC. Thus, a power-aware High-Level Synthesis technique concentrates on the overall power reduction and is not appropriate for temperature-aware IC design. The temperature-aware design technique requires optimization in all phases of design. This paper, proposes an effective temperature-aware scheduling with incremental binding for the IC design. In the proposed method,the temperature awareness is incorporated in both high-level and physical level synthesis. Comparison of the proposed method with other temperature-aware techniques in experimental analysis shows significant reduction of the peak temperature and power of the Integrated Circuit.

Key words: High-Level Synthesis (HLS) · Temperature · Integrated Circuits (Ics) · Scheduling · Binding

INTRODUCTION

The reliability of the Integrated Circuits (ICs) is mainly affected by its high operating temperature [1]. This requires proper IC cooling techniques which increases the cooling cost, packaging cost and size [2, 3]. Due to this thermal issue, the performance of the Application Specific Integrated Circuits (ASIC) is significantly affected, mainly in portable devices [4]. The International Technology Roadmap for Semiconductors (ITRS) [5] projected that the thermal issues has significant impact on the design of future IC. The feature size decrease by technology scaling increases the transistor densities in a single die [5]. This allows ICs to have billions of transistors which provide high computational capability at the cost of more power dissipation. The high logic density in particular region of an IC, causes more heat dissipation in that region which increases the IC peak temperature, resulting in an uneven thermal hotspots [6]. The peak IC operating temperature has many drawbacks such as it decreases the carrier mobility reducing the speed in transistor switching, interconnect and increase the sub-threshold leakage power due to more carrier concentration [4]. This causes the IC to miss its timing margin leading to functional failure. In order to obtain a reliable and an optimal IC performance, it is inevitable to overcome the thermal issue in every stage of High-level Synthesis(HLS).

High-Level Synthesis: HLS [7, 8, 9] is the process of transforming the behavioral representation of a design into a structural representation (Register Transfer Level (RTL)). Behavioral representation could be in any Hardware Description Languages (HDL) like VHDL, Verilog, etc. During HLS, the constructs specified in the HDL are transformed into hardware entities. Optimal integration of these hardware entities creates many optimization opportunities in the area of VLSI. Many papers talk about the optimization techniques that could be applied for efficient integration of hardware.

HLS is broadly classified into front end and back end. Front end comprises of scanner, parser and intermediate code generator. Their work is similar to a High Level Language (HLL) compiler. Scanner scans the input and parser produces an annotated parse tree that is then elaborated into an intermediate representation. The following tasks are done during elaboration, instantiating modules, evaluating and propagating symbolic constants, checking connectivity of all devices in the circuit and producing a checked consistent design.

Corresponding Author: A. Punitha, Chettinad College of Engineering and Technology, Tamil Nadu, India.

Two types of internal representations generally used are parse trees and graphs. Most approaches use variations of graphs that contain both the data-flow and the control-flow implied by the specification. This graph is called as Control/Data Flow Graph (CDFG), a variant of syntax tree, along with control information. HLS back end comprises of optimizer and synthesizer phases. The synthesizer phases consists of scheduling, allocation and binding. Compiler optimization techniques are applied on the CDFG during optimization phase. This is done to improve speed, silicon area and power. Scheduling assigns operations to clock cycles. The allocation decides the required number of functional units; the scheduling allocates control steps to the operations and the binding unit assign the functional units to the scheduled operations.The floor planning information provides proper placing of functional units with interconnect. Interconnect contributes to the significant portion of total power dissipation [10]. For an effective thermal aware IC design, the temperature optimization has to be done in all HLS phases [11].

Need for Temperature Optimization: Elevated temperatures in a silicon device could lead to memory errors, hard disk read write errors, failure of the device and other problems. Manufacturers specify a maximum operating temperature for their products. The life of an electronic device is directly related to the operating temperature. Excessive temperature results in heat. Heat has to be removed from a device to ensure that the device is maintained within its functional and maximum design temperature limits. If heat buildup becomes excessive, the device's temperature might exceed the temperature limits and the device may fail to perform. Hence, temperature has to be controlled to ensure product reliability and performance.

In this paper, a Temperature-aware scheduling with incremental binding for an effective thermal aware IC design is discussed. Section 2 presents the prior work done in HLS. Section 3 presents the scheduling and incremental binding of the proposed method. Section 4 presents the implementation and results of the proposed method in comparison with existing power-aware technique. Section 5 concludes the paper [12-36].

Prior Work: Temperature is not considered in many RTL designs. Temperature variations and hotspots account for over 50% of electronic failures. Temperature of an IC chip depends on the position and time of a module, heat generated in the module, assignment of tasks to a module, relative timing of operations executing in a module. Power and temperature variations can also lead to significant temperature timing uncertainty, thereby reducing performance. High temperatures could be avoided by appropirate temperature modeling and simulation [29]. Task scheduling and resource binding in HLS significantly impacts the power and hence their temperature. An increase in temperature increases sub threshold leakage, leading to further increase in temperature [30].

Yibochen *et al*. minimized the leakage power of the IC during HLS process. Aging effects such as Negative Bias Temperature Instability (NBTI) is considered. Based on the initial threshold voltage, the NBTI varies. The authors utilized this feature to minimize leakage power [31]. The temperature is distributed non-uniformly in an IC. The synthesis methodology should consider the thermal differences between the modules in the floorplan. Vyas *et al*. proposed a HLS methodology that is aware of temperature. The proposed approach takes the estimates from the incremental floorplanner and based on the estimates the power and peak temperature is optimized [30]. Rajarshi *et al*. introduced a resource binding technique. This technique takes into account the temperature effects and reduces the maximum temperature that is reached by a module in a design. Temperature constrained resource minimization and resource constrained temperature minimization techniques are proposed by the authors. Based on the constraints, the maximum observed temperature could be reduced with the area and power penalty [14].

Nan Wang *et al*. introduced a min cut based leakage power aware scheduling. The proposed algorithm is probability based to estimate resource usage effectively that also helps to estimate leakage power [32] and reduce it. Alberto *et al*. found a way to solve the power reduction problem in hetrogenousdatapath. Low-power functional units are introduced in the place of non low-power counterparts that reduced power by 27% [33].

Making early decisions related to temperature will guide us to obtain optimized design [34]. Thermal aware module binding in high level synthesis, discussed in [35] minimizes the peak switched capacitance of the module and the total switched capacitance of modules. Reducing the switched capacitance minimizes the temperature. Number of transistors increase every year as per Moore's Law. Leakage power is associated with the number of transistors.

On the other hand, the compilers could me made thermal aware. Inserting NOPs in the assembly code will make the hottest unit, cool down. Thermal aware instruction binding technique could be employed that would effectively bind the instructions executed in parallel to the coolest possible functional units. Power can also be taken into account during binding. This idea is discussed in [36]. Comparing the previous two methods, temperature aware binding reduces temperature of an IC effectively. The full-chip thermal modeling and analysis for IC during synthesis are proposed in [12, 13]. Thermal optimization techniques in behavioral and physical synthesis using voltage assignment and voltage island generation is designed by Gu *et al*. in TAPHS [15].

The research literature that handles Tempe ration optimization at a higher abstraction level is very minimum. This paper proposes an idea to optimize temperature during HLS by performing temperature aware scheduling and binding at the RTL level.

Temperature Aware Scheduling and Iterative Binding: In HLS, the back end uses the intermediate representation such as Control Data Flow Graph (CDFG) to generate the RTL net list. The Control Flow Graph (CFG) represents the control flow between the basic block, whereas the Data Flow Graph (DFG) gives the dependency between the operations within the basic block. A basic block is a sequence of consecutive operations in which flow of control enters at the beginning and leaves at the end without halt or jump except at the end. Figure 1 gives an example Basic Block.

The scheduler partitions the CDFG into control steps and schedules the operations to the control step which represents a scheduled CDFG.The binding assigns the available functional units for the operations in the control step of the scheduled CDFG. In order to reduce the thermal effects of the IC, the proposed method incorporates temperature awareness in the scheduling, binding and floorplanning as discussed below.

Fig 1: An example Basic Block.

Temperature Aware Scheduling: Present scheduling algorithms are either resource constrained or time constrained. For each operation, the scheduler assigns the operations within the schedule interval based on the scheduling constraints. The schedule interval gives the time interval between the As-Soon-As-Possible (ASAP) [21, 22] and As-Late-As-Possible (ALAP) schedules [23]; the ASAP schedules uses the earliest control step for the operations and the ALAP schedules uses the longest control step for the operations as long as it satisfy the scheduling constraints. Eventhough, scheduling algorithm such as Force-Directed Scheduling (FDS) algorithm [24] handles even distribution of operations among the control steps, the continuous switching activity in a functional unit results in increase in the power density. This leads to thermal variation across the IC with subsequent thermal hotspots creation. In our proposed method, in addition to the even distribution of operations, the scheduling follows two methodologies that reducesthe thermal hotspot creation. The scheduler first considers the operation adjacency upto three control steps. In order to avoid the continuous switching activity of functional units, the sameoperation type used in three adjacent control steps is moved to the successive control steps within the schedule interval. Let 0_i bean operation in the control step C_i and the schedule S has resource constraints r_i. The schedule interval is given by t_{sch}. The scheduler allocates the operation evenly within the schedule interval based on its constraints. The operation adjacency is included as a constraint into the scheduling algorithm.

The operation scheduling $S(0_i)$ with adjacency constraint is given as

$$S(0_i) \mid adj_{neg}(C_i \, to \, C_{i-2}) \tag{1}$$

Where adj_{neg} represents the lack of adjacency in those control steps as long as it satisfy the data dependency based on operator precedence.

Secondly, the scheduler exploits the parallelism between the basic blocks to avoid thermal hotspot creation. Due to the sequential execution of operations in the CFG, the parallelism can be exploited only within the basic block allowing resource sharing within it. The CDFG branch creates concurrent and mutually exclusive basic blocks; which allows parallel execution and resource sharing between the basic blocks. The scheduler moves some operation from the basic block having high operational density to the other basic block. While moving an operation from high operational density basic block, the operation adjacency is also considered.

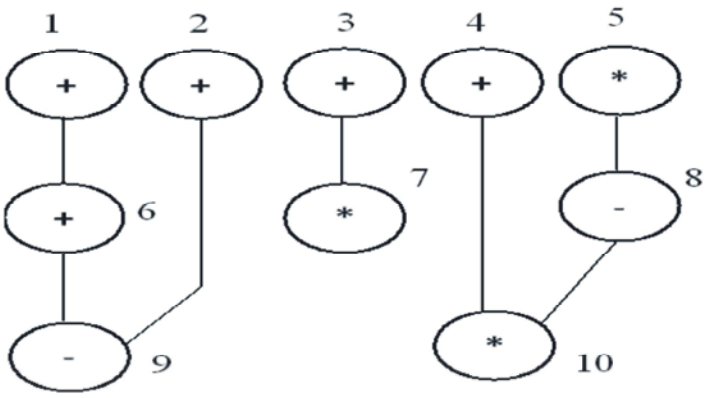

Fig 2: Scheduled Data Flow Graph (DFG).

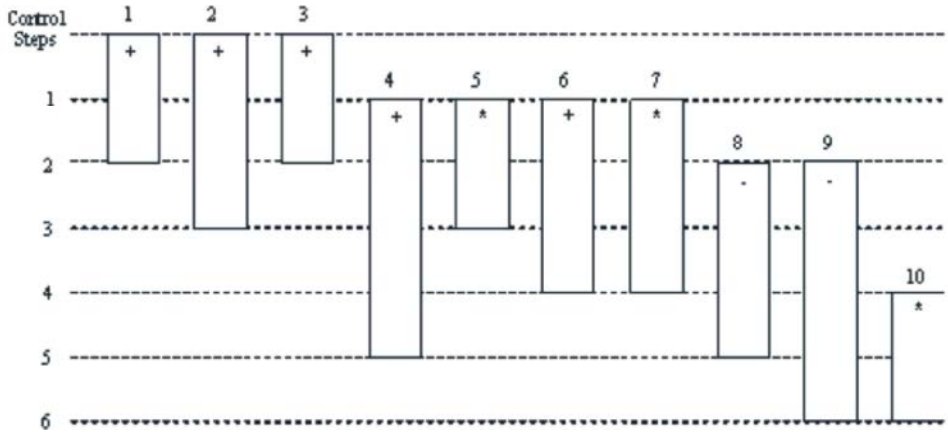

Fig 3: Operation schedule interval

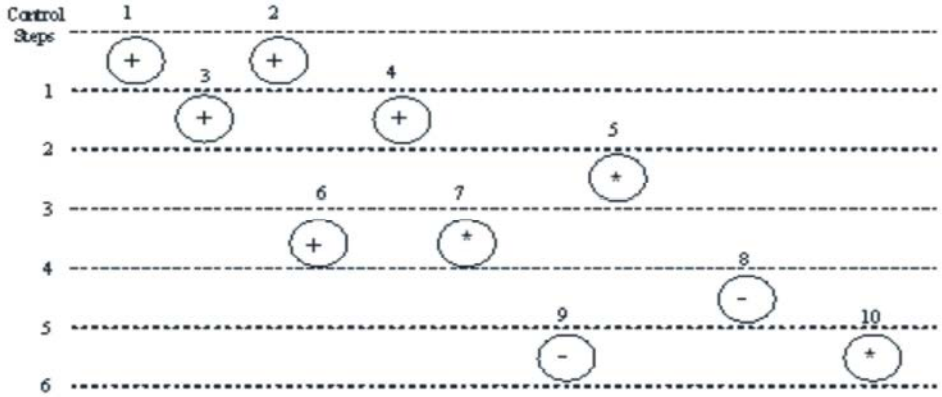

Fig 4: Proposed Temperature-aware Schedule

The scheduled DFG given in Figure 2 has resource constraints of two adder units along with multiplier and subtraction unit of each.

The schedule interval for each operation in the control steps is given in Figure 3. The proposed temperature-aware scheduling strategy schedules the operation in the control step which avoids the heat buildup in particular functional unit. Figure 4 shows the operation schedule according to resource constraint, in which the same functional operation is not scheduled more than two consecutive control steps to avoid heat buildup.

Incremental Binding and Floor Planning: The binding operation assigns the functional unit to the operations in the scheduled CDFG. The floor planning creates the

RTL netlist which consists of Data Path Unitwith interconnects. The data exchange between the neighboring basic blocks in the floorplan results in reduced data communication cost in terms of latency and power; whereas in other cases of data exchange the data communication cost is increased. The data transfer between the basic blocks, where it is generated and received is determined using Static Single Assignment (SSA) [25]. In SSA, the data variable which is exchanged between the basic blocks with different definitions is renamed into distinct variables for each definition. The proposed floorplanner uses the SSA and places the basic blocks with data exchange in the neighboring location, to avoid long interconnect between them. In the incremental binding and floorplanning, thermal analysis is performed in the initial floorplan. The rebinding algorithm uses the thermal analysis result to identify the functional unit with high temperature. It moves some of the operations from the high temperature functional unit to the functional unit which satisfies both the constraints of minimum temperature and less distance. The binding move for an operation between functional units has to satisfy data dependency in the DFG. This rebinding is followed by floorplanning method discussed above to reduce the data communication cost for the data exchange between the basic blocks. In the re-floorplanning, the initial physical locality is not altered in order to make the rebinding feasible.

Let s_{ik} denote the operation schedule in the scheduled DFG and it is fixed. For these scheduled DFG, b_{ij} gives the binding of available functional units to every operation in all the control steps.

The binding is feasible only if it satisfies the following constraints.

- $\sum_{j=1}^{r_{\max}} b_{ij} = 1$

for $i=1,2,\ldots\ 0_{max}$

- $\sum_{i=1}^{0_{\max}} b_{ij} . s_{ik} \leq 1$

For $j=1,2,\ldots\ r_{max}$ *and* $k=1,2,\ldots\ c_{max}$

Here, i is the operation in the control steps with maximum operations 0_{max} in the scheduled DFG.

k is the number of control steps with maximum control steps c_{max} in the scheduled DFG.

j is the resource numbers available for a particular operation type with maximum operations r_{max}.

The rebinding moves the operations from the high temperature functional unit to the possible functional unit based on its locality and temperature constraints along with the data dependency. The rebinding assigns weight W to the possible resources based on the cost function derived from the locality and temperature. The cost function determines the locality and temperature form the initial floorplanning and thermal analysis. The functional unit with minimum temperature and interconnect distance has minimum cost function, which is considered for the binding move.The binding move from one functional unit to another functional unit with k possible.

neighboring blocks is given by

$$b(r_{j1}) \rightarrow b(r_{j2}) \mid r_{j2} = min\ (W(r_j))$$

$$W(r_j) = cost\ (d_i, T_i),\ where\ j = 1,2,\ldots\ k$$

RESULTS

This section discusses the efficiency of the proposed temperature-aware technique with power-aware technique. The media benchmark circuits [26] are used to evaluate the proposed temperature-aware technique are HAL (second-order differential equation solver), FIR1 (Finite Impulse Response filter), Jacobi (iterative fourth-order linear system), ARF (Auto-Regression Filter), IIR77 (Infinite Impulse Response Filter) and EWF (Elliptic Wave Filter). The benchmark circuits have varying number of scheduling nodes in its CDFG. The thermal analysis is done using an architectural-level Thermal modeling tool ISAC [27]. The ISAC tool uses the floorplan information with its power dissipation estimates as power traces and provides the thermal distribution and peak temperature of the floor plan. Both power-aware and temperature-aware techniques are of resource constrained type (i.e. fixed number of resources). The IC peak temperature and power mainly depends on its power density and is not evenly distributed. Since the power-aware technique [28] does not concentrate on power density reduction, there is no significant reduction in the IC peak temperature and power. In the proposed temperature-aware technique, the scheduling mainly concentrates on the power density reduction of the functional unit by avoiding continuous switching activity, achieves significant reduction in IC Peak temperature and power. Further the incremental binding and floorplanning reduces the temperature of a high temperature functional module and the interconnect length achieves significant

Table 1: Comparison of IC peak temperature.

Bench Mark	Power Aware [19] (°C)	Proposed (°C)
HAL	75.2	67.5
FIRI	79.5	69.1
Jacobi	57.2	51.4
ARF	89.6	81.2
IIR77	82.5	73.4
EWF	83.4	74.9

Table 2: Comparison of IC peak power.

Bench Mark	Power Aware [28](°C)	Proposed (°C)(*mW*)
HAL	161.7	156.8
FIRI	165.3	160.1
Jacobi	153	147.2
ARF	194.1	188.5
IIR77	177.3	171.4
EWF	184	178.2

IC peak temperature and power reduction. Table 1 gives the IC peak temperature comparison of the proposed method with power-aware technique for various benchmark circuits [29].

In the benchmark circuits,the FIR1 achieves the maximum IC peak temperature reduction of 13.08% and the ARF circuit achieves the minimum IC peak temperature reduction of 9.37%. The IC peak temperature reduction of the proposed temperature-aware technique varies across the benchmark circuits due to the number of scheduling nodes in the CDFG. Since the IC peak power consumption is based on its power density, the proposed temperature-aware techniques reduce the peak power consumption also. Table 2 gives the IC peak power comparison of the proposed method with power-aware technique for various benchmark circuits. Thus the proposed temperature-aware technique achieves maximum reduction of IC peak power for Jacobi with 3.79% and minimum reduction of IC peak power for ARF with 2.88%.

This helps to reduce the overall power consumption and the leakage power. Since the IC peak temperature also depends on the thermal conductivity across the spatial domains, the benchmark circuit having peak temperature and peak power consumption are not identical [30].

CONCLUSION

In this paper, a temperature-aware scheduling with incremental binding and floorplanning for HLS is proposed to reduce the IC peak temperature and power. This temperature-aware technique focuses on the reduction of power density to achieve a minimum IC peak temperature. Various benchmark circuits are used to analyze the efficiency of the proposed method. Compared to the power-aware technique, the proposed method achieves maximum temperature improvement of 13.08% for FIR1 and maximum power improvement of 9.78% for Jacobi.

REFERENCE

1. Pedram, M. and S. Nazarian, 2006.Thermal modeling, analysis and management in VLSI circuits: Principles and methods,IEEE Proceedings, 94(8): 1487-1501.
2. Sylvester, D. and H. Kaul, 2001. Power Driven Challenges in Nanometer Design,IEEE Design and Test of Computers, 13(6): 12-21.
3. Borkar, S., 2002. Design Challenges of Technology Scaling,IEEE Micro, 9: 23-29.
4. Yeh, L.T. and R.C. Chu, 2002. Thermal Management of Microelectronic Equipment: Heat Transfer Theory, Analysis Methods and Design Practices. New York, NY: ASME Press.
5. ITRS International Technology Roadmap for Semiconductors. http://public.itrs.net, 2006.
6. Skadron, K., M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, 2003. Temperature-aware microarchitecture, in Proc. Int. Symp.Comput. Arch., pp: 2-13.
7. Camposano, R. and W. Wolf, 1991. High Level VLSI Synthesis. Kluwer, MA.
8. Gajski, D., N. Dutt, A. Wu and S. Lin, 1992. High-Level Synthesis: Introduction to Chip and System Design. Kluwer, MA.
9. Raghunathan, N.K. Jha and S. Dey, 1998. High-Level Power Analysis and Optimization. Kluwer, MA.
10. Cong, J. and Z. Pan, 2001. Interconnect performance estimation models for design planning, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, pp: 739-752.
11. Prabhakaran, P. and P. Banerjee, 1998. Simultaneous scheduling, binding and floorplanning in high-level synthesis, in Proc. Int. Conf. VLSIDes, pp: 428-434.
12. Li, P., 2004. Efficient full-chip thermal modeling and analysis, in Proc. Int. Conf. Computer-Aided Design, pp: 319-326.
13. Guo, X., 2004. The creation of compact thermal models of electronic components using model reduction, in Proc. Semiconductor Thermal Measurement & Management Symp, pp: 104-110.
14. Mukherjee, R., S. O renci Memik and G. Memik, 2005. Temperature-aware resource allocation and binding in high-level synthesis, in Proc. Design Automation Conf.

15. Gu, P., Y. Yang, J. Wang, R.P. Dick and L. Shang, 2006. TAPHS: Thermal-aware unified physical-level and high-level synthesis, in Proc. Asia & South Pacific Design Automation Conf, pp: 879-885.
16. Lim, P. and T. Kim, 2006. Thermal-aware high-level synthesis based on network flow method, in Proc. Int. Conf. Hardware/Software Codesign and System Synthesis.
17. Tsai, C. and S. Kang, 1999. Standard Cell Placement for Even On-Chip Thermal Distribution. International Symposium on Physical Design.
18. Goplen, B. and S. Sapatnekar, 2003. Efficient thermal placement of standard cells in 3D ICs using a force directed approach, in Proc.Int. Conf. Computer-Aided Design, pp: 86-89.
19. Cong, J., J. Wei and Y. Zhang, 2014. A thermal-driven floorplanning algorithm for 3D ICs, in Proc. Int. Conf. Computer-Aided Design, pp: 306-313.
20. Shang, L., L.S. Peh, A. Kumar and N.K. Jha, 2004. Thermal modeling, characterization and management of on-chip networks, in Proc. Int. Symp. Microarch, pp: 67-78.
21. Tseng, C. and D.P. Siewoirek, 1986. Automated Synthesis of data paths in digital systems. IEEE Trans. Computer Aided Design, CAD, 5: 379-295.
22. Genotys, C.H. and M.I. Elmastry, 1987. A VLSI methodology with testability constraints, in Proc 1987 Canadian Conf. VLSI (Winnipeg).
23. Kung, S.Y., H.J. Whitehouse and T. Kailath, 1985. VLSI and Modern Signal Processing. Englewood Cliffsm NJ: Prentice Hall, pp: 258-264.
24. Paulin, G., Pierre and John P. Knight, 1989. Force Directed Scheduling for the behavioral synthesis of ASIC's, IEEE Trans. Computer Aided Design, 8: 661-679.
25. Cytron, R., J. Ferrante, B.K. Rosen, M.N. Wegman and F.K. Zadeick, 1989. An efficient method of computing static single assignment form, Symposium on Principles of Programming Languages.
26. Lee, C., M. Potkonjakm and W.H. Mangione-Smith, 1997. Media Benc h: Atool for evaluating and synthesizing multimedia and communicatonssystems, in Proc. Int. Symp. Microarch., pp: 330-335.
27. Raghunathan and N.K. Jha, 1997. SCALP: An iterative-improvementbasedlow-power data path synthesis system,IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 16(11): 1260-1277.
28. Mohanty, S.P., N. Ranganathan and S.K. Chappidi, 2005. Simultaneous peak and average power minimization during datapath scheduling, IEEE Trans Circuits and Systems I, pp: 1157-1165.
29. Jeng-Liang Tsai, C.C.P. Chen., Guoqiang Chen, B. Goplen, Haifeng Qian, Yong Zhan, Sung-Mo Kang, M.D.F. Wong, and S.S. Sapatnekar, 2006. Temperature-Aware Placement for SOCs, Proceedings of the IEEE, 94(8): 1502-1518.
30. Krishnan Vyas and Srinivas Katkoori, 2009. Simultaneous Peak Temperature and Average Power Minimization during Behavioral Synthesis, IEEE, pp: 419-424.
31. Chen Yibo, Yuan Xie, Yu Wang and Andres Takach, 2010. Minimizing Leakage Power in Aging-Bounded High-level Synthesis with Design Time-Vth Assignment, IEEE, pp: 689-694.
32. Nan Wang, Song Chen and Takeshi Yoshimura, 2013. Min-Cut Based Leakage Power Aware Scheduling in High-Level Synthesis, IEEE, pp: 164-170.
33. Alberto A. Del Barrio, Seda Ogrenci Memik, Marý´a C. Molina, Jose M. Mendýas and Roman Hermida, 2013. A fragmentation aware High-Level Synthesis ?ow for low power heterogenousdatapaths, INTEGRATION, the VLSI journal, pp: 119-130.
34. Joseph, M., 2007. NarasimhaB.Bhat and K.Chandrasekaran, Technology driven High Level Synthesis, 15th International Conference on Advanced Computing and Communications, IEEE, pp: 485-490.
35. Lim Pilok and Taewhan Kim, 2006. Thermal-Aware High-level Synthesis based on Network Flow Method, CODES+ISSS '06, ACM, pp: 124-129.
36. Carrion Benjamin and Schafer Yongho Lee, 2007. Taewhan Kim, Temperature-Aware Compilation for VLIW Processors, 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications RTCSA.