Rapid Design Space Exploration for multi parametric optimization of VLSI designs

Anirban Sengupta, Reza Sedaghat, Zhipeng Zeng
Electrical and Computer Engineering, Ryerson University
Toronto, Canada

Abstract – **Design Space Exploration (DSE) is one of the most important stages in High Level Synthesis designing methodology. This paper presents a novel DSE approach for the current generation of systems with heterogeneous multi parametric optimization objectives. The method introduced in this paper is capable of concurrently resolving multiple conflicting issues encountered during DSE, such as maximization of accuracy needed in the evaluation of design space with minimization in time expended to explore the best architecture. Results of the proposed method for different benchmarks indicated significant acceleration in exploration process compared to another existing approach that is also based on Pareto optimal analysis.**

I. INTRODUCTION

For systems with heterogeneous multi-parametric optimization objectives based on stringent operational constraints, the evaluation and selection of the optimal architecture for system design is one of the most vital steps in the development process. Since the architecture design space consists of innumerable design alternatives with the same functionality, selection of the best configuration for the design requires extensive analysis of the design space. This is the case not only because of the assorted nature of the performance parameters, but also due to the diversity in architecture implementation. Therefore, for the current generation of Very Large Scale Integration (VLSI) and System-on-Chip (SoC) designs requiring optimization of multiple conflicting performance objectives an efficient design space exploration approach that simultaneously optimizes the multiple conflicting performance objectives and the time expended to explore the design space with high precision is essential [1].

Many approaches to DSE exploration have been proposed till now. For instance, authors in [2] utilize the Architecture Configuration Graph (ACG) based on hierarchical factor for architecture evaluation and selection. Moreover in [3], the authors use evolutionary algorithms like Genetic Algorithm (GA) for efficient DSE. They propose a new encoding scheme to improve the efficiency of GA search for design space exploration using chromosome representation that encodes the precedence relationships among the tasks in the input behavioral specification with a topological order-based representation to specify schedule priorities. Furthermore, in [4], the work is based on binary encoding of chromosomes for efficient design space exploration. Authors in [5] also suggest an evolutionary algorithm for successful evaluation of the design for an application specific SoC.

II. THE PROPOSED FRAMEWORK BEHIND DESIGN SPACE EXPLORATION

A. Analysis for Execution time

The term ‘workload’ of a resource can be defined as the time taken (or clock cycles needed) by a resource to finish its assigned operation during scheduling. Hence the total workload (*W*) of all the resources for finishing their respective operations can be represented by (1):

$$W = (N_{R1} \cdot T_{R1} + N_{R2} \cdot T_{R2} + ... + N_{Rn} \cdot T_{Rn}) \cdot D.T_P \quad (1)$$

Where N_{Ri} represents the number of resource ‘Ri’ and ‘T_{Ri}’ represents the number of clock cycles needed by resource ‘*Ri*’ (1<=i<=n) to finish each operation. ‘D’ is the number of times the data elements needs to be processed for all sets of data; ‘T_p’ is the time period of the clock.

From the theory of approximation of differentials the change in ‘workload’ is approximated in (2).

$$dW = D \cdot [(\frac{\partial W}{\partial N_{R1}} \cdot \Delta N_{R1} + \frac{\partial W}{\partial N_{R2}} \cdot \Delta N_{R2} + ... + \frac{\partial W}{\partial N_{Rn}} \Delta N_{Rn}) + \Delta T_p \cdot \frac{\partial W}{\partial T_p}] \quad (2)$$

Now applying partial derivative to the (1) with respect to N_{R1}.....N_{Rn} and T_p will produce the following set of equations:

$$\frac{\partial W}{\partial N_{Rn}} = \frac{\partial[(N_{R1} \cdot T_{R1} + N_{R2} \cdot T_{R2} + ... + N_{Rn} \cdot T_{Rn}) \cdot T_p \cdot D]}{\partial N_{Rn}}$$

$$= T_{Rn} T_p \cdot D \quad (3)$$

$$\frac{\partial W}{\partial T_p} = \frac{\partial[(N_{R1} \cdot T_{R1} + N_{R2} + ... + N_{Rn} \cdot T_{Rn}) \cdot T_p \cdot D]}{\partial T_p}$$

$$= (N_{R1.} \cdot T_{R1} + N_{R2} \cdot T_{R2} + ... + N_{Rn} \cdot T_{Rn}) \cdot D \quad (4)$$

This work is supported by the Optimization Problem Research and Applications Laboratory (OPRAL), Ryerson University, NSERC CRSNG, Ontario Innovation Trust, Sun Microsystems, Motorola and Canadian Microelectronics Corporation (CMC).

Now substituting (3) and (4) in (2) yields (5):

$$dW = \Delta N_{R1} \cdot T_{R1} \cdot T_p \cdot D + \Delta N_{R2} \cdot T_{R2} \cdot T_p \cdot D + .. + \Delta N_{Rn} \cdot T_{Rn} T_p \cdot D$$
$$+ D \cdot \Delta T_p \cdot (T_{R1} \cdot N_{R1} + T_{R2} \cdot N_{R2} + ... + T_{Rn} \cdot N_{Rn}) \quad (5)$$

Equation (5) reflects the change in total workload with the change in the number of all the resources and the clock period (clock frequency).

$\Delta N_{Rn} \cdot T_{Rn} \cdot T_p \cdot D$ = The change of '*W*' contributed by the change in the number of resource Rn. And,

$\Delta T_p \cdot (T_{R1} \cdot N_{R1} + T_{R2} \cdot N_{R2} + ... + T_{Rn} \cdot N_{Rn}) \cdot D$ = The change of '*W*' contributed by the change in clock period (frequency).

Considering constraint on the number of resources, the increase in total workload (W) will cause an increase in total execution time. Therefore, the more the workload increases, the more the execution time increases under resource constraints. Hence the change in number of a resource (e.g. change in adder from one to three) that contributes to the change in total workload the most, also contribute to the change in total execution time the most. So based on above analysis, the PF for execution time parameter is defined as:

$$PF(Rn) = \Delta N_{Rn} \cdot T_{Rn} \cdot (T_p)^{\max} \;/\; N_{Rn} \quad (6)$$

$$PF(Rclk) = \frac{N_{R1} \cdot T_{R1} + N_{R2} \cdot T_{R2} + .. + N_{Rn} \cdot T_{Rn}}{N_{Rclk}} \cdot (\Delta T_p) \quad (7)$$

'D' is ignored in the expression for PF because it does not contribute to the change in Priority Order (PO) sequence described later in the paper. The factors defined above reflect the average change in execution time (T_{exe}) with the change in number of a resource (change in adder from one to three) at maximum clock period as well as reflects the average change in execution time (T_{exe}) with the change in clock frequency. In the expression for PF in (6), minimum clock frequency is considered because at this frequency the clock period is the maximum. Hence, the change in number of a specific resource at maximum clock period will influence the change in execution time the most, compared to the change in execution time at other clock periods.

B. *Analysis for power consumption*

For a system with 'n' functional resources the total power consumption (*P*) of the resources can be represented by the following equations.

$$P = \sum_{i=1}^{n} (N_{Ri} \cdot K_{Ri}) \cdot p_c \quad (8)$$

$$P = (N_{R1} \cdot K_{R1} + N_{R2} \cdot K_{R2} + .. + N_{Rn} \cdot K_{Rn}).p_c \quad (9)$$

'K_{Ri}' represents the area occupied per unit resource Ri and 'p_c' denote the power consumed per area unit resource at a particular frequency of operation. Now using the theory of approximation of differentials the change in power consumption can be formulated as shown in (10):

$$dP = (\frac{\partial P}{\partial N_{R1}} \cdot \Delta N_{R1} + \frac{\partial P}{\partial N_{R2}} \cdot \Delta N_{R2} + ... + \frac{\partial P}{\partial N_{Rn}} \Delta N_{Rn}) + \Delta p_c \cdot \frac{\partial P}{\partial P_c} \quad (10)$$

Applying partial derivative to (9) will produce the following equations:

$$\frac{\partial P}{\partial N_{Rn}} = \frac{\partial[(N_{R1} \cdot K_{R1} + N_{R2} \cdot K_{R2} + ... + N_{Rn} \cdot K_{Rn}) \cdot p_c]}{\partial N_{Rn}}$$

$$= K_{Rn} \cdot p_c \quad (11)$$

$$\frac{\partial P}{\partial p_c} = \frac{\partial[(N_{R1} \cdot K_{R1} + N_{R2} \cdot K_{R2} + ... + N_{Rn} \cdot K_{Rn}) \cdot p_c]}{\partial p_c}$$

$$= N_{R1} \cdot K_{R1} + N_{R2} \cdot K_{R2} + ... + N_{Rn} \cdot K_{Rn} \quad (12)$$

Substituting (11) and (12) in (10) yields (13) below:

$$dP = (\Delta N_{R1} \cdot K_{R1} \cdot p_c + \Delta N_{R2} \cdot K_{R2} \cdot p_c + ... + \Delta N_{Rn} \cdot K_{Rn} p_c)$$
$$+ \Delta p_c \cdot (K_{R1} \cdot N_{R1} + K_{R2} \cdot N_{R2} + ... + K_{Rn} \cdot N_{Rn}) \quad (13)$$

Equation (13) represents the change in total power consumption with the change in the number of all resources and the clock period (clock frequency).

$\Delta N_{Rn} \cdot K_{Rn} \cdot p_c$ = The change of '*P*' contributed by the change in the number of resource Rn; and

$\Delta p_c \cdot (K_{R1} \cdot N_{R1} + K_{R2} \cdot N_{R2} + ... + K_{Rn} \cdot N_{Rn})$ = The change of '*P*' contributed by the change in clock period. Let's define the PF for power consumption as shown below:

$$PF(Rn) = \Delta N_{Rn} \cdot K_{Rn} \cdot (p_c)^{\max} \;/\; N_{Rn} \quad (14)$$

$$PF(Rclk) = \frac{N_{R1} \cdot T_{R1} + N_{R2} \cdot T_{R2} + .. + N_{Rn} \cdot T_{Rn}}{N_{R_{clk}}} \cdot (\Delta p_c) \quad (15)$$

The priority factors defined in (14) and (15) reflect the average change in the total power consumption of the system with the change in number of a resource (e.g. change in adder from one to three) at maximum clock period. In the above equations, the maximum clock frequency was considered because at this frequency the total power consumption is the maximum. Thus the change in the number of a specific resource at maximum clock frequency will influence the change in the total power consumption (*P*) the most, compared to the change at other clock frequencies. The priority factor helps to arrange the architectural variants of the design space in increasing or decreasing order of magnitude depending on the parameter of optimization.

III. PROPOSED METHOD OF DESIGN SPACE EXPLORATION FOR DETERMINATION OF THE OPTIMAL VARIANT OF ARCHITECTURE

A. System specifications

The case study of a selected benchmark has been provided for demonstration of the proposed method based on some real system specifications (as shown in Table I). The function of the selected second order digital IIR Butterworth filter [6] benchmark is given in (16).

$$y(n) = 0.167x(n) + 0.5x(n-1) + 0.5x(n-2) + 0.167x(n-3) - 0.33y(n-2) \qquad (16)$$

B. Calculation of the priority factor for each available resource for execution time parameter

For resource adder/subtractor (R1), multiplier (R2), clock oscillator (R_{clk}):

$$PF(R1) = \Delta N_{R1} \cdot T_{R1} \cdot (T_p)^{\max} / N_{R1} = \frac{(3-1)\cdot 2}{3} \cdot (0.02) = 0.026$$

$$PF(R2) = \Delta N_{R2} \cdot T_{R2} \cdot (T_p)^{\max} / N_{R2} = \frac{(4-1)\cdot 4}{4} \cdot (0.02) = 0.06$$

$$PF(R_{clk}) = N_{R1} \cdot T_{R1} + N_{R2} \cdot T_{R2} \cdot (\Delta T_p) \ / \ N_{Rclk} = (3\cdot 2 + 4\cdot 4)\cdot(0.02 - 0.005) \ / \ 2 = 0.165$$

According to the above analysis the change in number of adder/subtractor affects the change in execution time the least, while the change in clock frequency from 50 MHz to 200 MHz affects the change in execution time the most. The minimum for adder/subtractor and multiplier is one in above equations because the digital IIR Butterworth filter function at least requires one adder and one multiplier to successfully accomplish the functioning of the task.

C. Arrangement of the resources in Priority Order based on the calculation of Priority Factor for execution time

According to the priority factors calculated, the next step is to arrange the resources priority order (PO). The priority order is arranged so that the resource with the lowest priority factor is assigned the highest priority order while the resource with the highest priority factor is assigned the lowest priority order. The priority order of the resource increases with the decrease in priority factor of the resource. Therefore the following PO of the resources is achieved for arranging the design variants in decreasing order for execution time.

PO (R1) > PO (R2) > PO (Rclk)

D. Arrange the design space in decreasing order for execution time according to the priority order

This section presents an algorithm for arranging the random design space in an organized decreasing order for the execution time parameter. The algorithm is based on priority order sequencing as described in the last section (Section III.C). The algorithm presented in Fig. 2 clearly describes the required steps in order to properly arrange the design variants. The PO obtained for execution time was *PO (R1) > PO (R2) > PO (Rclk).* After using the model of the proposed algorithm the arranged design space for execution time is obtained and is shown in Fig.3 (see next page). After the variants were organized in decreasing order the binary search algorithm is applied to obtain the border variant for the execution time parameter. Results of binary search on the design space shown in Fig.3 yielded 'variant V13' (marked in bold) as the border variant for the execution time parameter. This signifies that variant V13 is the first variant in the design space that satisfies the constraint for execution time specified (Table I).

TABLE I. SYSTEM SPECIFICATIONS AND CONSTRAINTS

1)	Maximum power consumption**:** 8 watts (W)
2)	Maximum time of execution: 140 µs (for D =1000 sets of data)
3)	Hardware area of resources**:** Minimum
4)	Maximum resources available for the system design: a) 3 Adder/subtractor units. b) 4 Multiplier units c) 2 clock frequency oscillators: 50 MHz and 200 MHz
5)	No. of clock cycles needed for multiplier and adder/subtractor to finish each operation: 4 cc and 2cc
6)	Area occupied by each adder/subtractor and multiplier: 20 area units (a.u) and 100a.u. on the chip (e.g. 20 CLBs on FPGA)
7)	Area occupied by the 50MHz and 200MHz clock oscillator: 4 area units and 10 area units
8)	Power consumed at 50 and 200MHz:10mW/a.u. and 40mW/a.u.

E. Arrange the design space according to increasing order

This section discusses about arranging the random design space in an increasing order for the power consumption parameter. Similar to the execution time parameter, the PF for all the available resources was calculated to determine how much a change in each resource affects power consumption. Once the PF was determined then the PO was obtained following the procedure described in section D. The PO sequence helped to obtain the arranged design space for power consumption using the algorithm in Fig. 2. Binary search was then applied on the arranged design space for power consumption. Binary search yielded variant V15 as the border variant that meets the constraint imposed for execution time. Since the steps to obtain the border variant for power are exactly the same as those to obtain the border variant for execution time (steps from section B to section D) they have not been shown in the paper. After determining the border variant for power consumption, we see that the Pareto optimal set contains the following variants V13, V14 and V15 (see Fig.3 next page). According to the specification the architecture with the minimum area should be found. Analysis revealed that 'V13' is the only variant which satisfies all the operating constraints the optimization requirements.

IV. ANALYSIS AND RESULTS

The proposed DSE approach uses binary search after arrangement of the design space using the priority factor method. The search of the optimal architecture requires only $\log_2 \prod_{i=1}^{n} v_{Ri}$ where 'n' = number of type of resources and 'v_{Ri}' is the number of variants of resource 'Ri'. In contrast, the exhaustive search checks for $\prod_{i=1}^{n} v_{Ri}$ architectural variants during optimal architecture selection while satisfying all operating constraints. In the design space exploration approach presented here three performance parameters have been used for optimization, where the execution time and

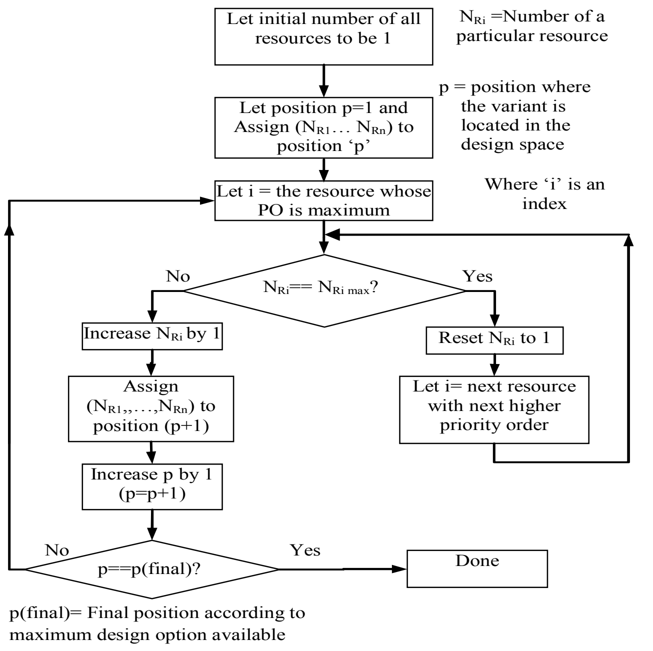
Let initial number of all resources to be 1
N_{Ri} =Number of a particular resource
p = position where the variant is located in the design space
Let position p=1 and Assign (N_{R1}… N_{Rn}) to position 'p'
Where 'i' is an index
Let i = the resource whose PO is maximum
No
N_{Ri}== $N_{Ri\ max}$?
Yes
Increase N_{Ri} by 1
Reset N_{Ri} to 1
Assign (N_{R1},,…,N_{Rn}) to position (p+1)
Let i= next resource with next higher priority order
Increase p by 1 (p=p+1)
No
p==p(final)?
Yes
Done
p(final)= Final position according to maximum design option available

Figure 2. Flow chart model of the proposed algorithm

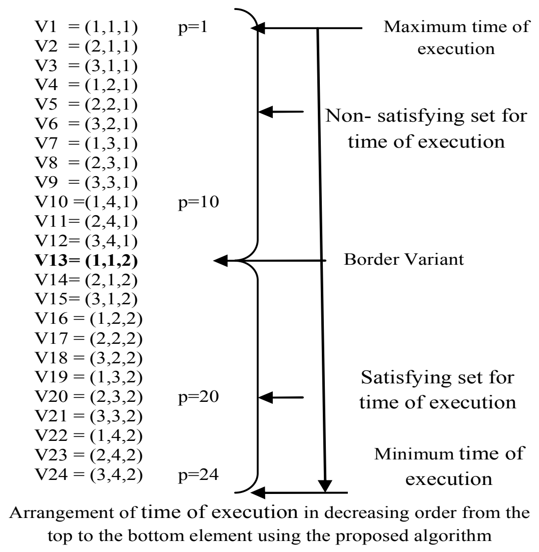

Figure 3. The arranged design space for execution time

power are the parametric constraints and area is the optimization parameter. Hence the searching has to be repeated for both parameters to determine the border variant. Therefore the total number of architecture evaluations using exhaustive search is given as: $M * \prod_{i=1}^{n} v_{Ri}$ and the total number of architecture evaluations using the proposed method is given as: $M * \log_2 \prod_{i=1}^{n} v_{Ri}$. 'M' denotes each performance parameter. Here the value of 'M' is two because there are two performance parametric constraints. The proposed approach was applied on various benchmarks to check the acceleration obtained through this DSE method. Results indicated massive acceleration in the speedup compared to the exhaustive approach. The results of this speedup using the proposed design space exploration framework for the benchmarks are illustrated in Table II. The proposed method was also compared with the current approach in [2] [7] which are also based on Pareto optimal analysis but, unlike our approach, use hierarchical tree arrangement. The result of this comparative study is also shown in Table II.

V. CONCLUSIONS

A novel approach for rapid DSE using Priority Factor method was presented in this paper. For the different benchmarks, the proposed approach for DSE was able to provide increased acceleration when compared to the existing DSE approach based on Pareto optimal analysis that uses hierarchical tree arrangement.

REFERENCES

[1] Christian Haubelt, Jurgen Teich,"Accelerating Design Space Exploration Using Pareto-Front Arithmetic's", In Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'03), Japan, 2003.

[2]Kirischian, L., Geurkov, V., Kirischian, V. and Terterian, I. (2006)'Multi-parametric optimisation of the modular computer architecture', Int. J.Technology, Policy and Management, Vol. 6, No. 3, pp.327–346.

[3] Vyas Krishnan and Srinivas Katkoori, "A Genetic Algorithm for the Design Space Exploration of Datapaths During High-Level Synthesis, IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, June 2006.

[4] E. Torbey and J. Knight, "Performing scheduling and storage optimization simultaneously using genetic algorithms," in Proc. IEEE Midwest Symp. Circuits Systems, 1998, pp. 284–287.

[5] Giuseppe Ascia, Vincenzo Catania, Alessandro G. Di Nuovo, Maurizio Palesi, Davide Patti, "Efficient design space exploration for application specific systems-on-a-chip" Journal of Systems Architecture 53 (2007) pages: 733–750.

[6] S Salivahanan, A Vallavaraj and C Gnanapriya, "Digital Signal Processing", Tata McGraw-Hill Publishing Company Limited, 2006, pp. 439-444.

[7] Kirischian, L. (2000) 'Optimization of parallel task execution on the adaptive reconfigurable group organized computing system', Proc. of International Conference PARELEC 2000, Canada, pp.150–154

TABLE II. The results of the proposed DSE approach for the Benchmarks

Benchmark	Type of resources	Number of variants of each resource	Total possible architecture for exhausted search	Evaluated architectures using the proposed DSE	Speedup obtained	Log_{10}(speedup)	Total Architecture evaluations using Hierarchical arrangement	Speed up of the proposed approach compared to the approach in [2][7] (%)
Edge Detector 1	7	8	$4.2\text{x}10^{6}$	42	$9.9\ \text{x}10^{4}$	4.995635	60	30
Edge Detector 2	14	8	$8.8\ \text{x}10^{12}$	84	$1.05\ \text{x}10^{11}$	11.02119	116	27.58621
2D Image Combiner	32	16	$6.81\ \text{x}10^{38}$	256	$2.66\ \text{x}10^{36}$	36.42488	324	20.98765
3D Image Combiner	32	32	$2.92\ \text{x}10^{48}$	320	$9.13\ \text{x}10^{45}$	45.96047	388	17.52577