A Framework for Fast Design Space Exploration using Fuzzy search for VLSI Computing Architectures

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

Abstract—In High level Synthesis design methodology, the evaluation and selection of the optimal architecture for the system is done through a process called Design Space Exploration (DSE). This paper presents a novel framework for fast DSE using fuzzy search technique for optimizing modular computing architecture for the current generation of multi objective VLSI designs. The proposed method is able to radically reduce the number of architectural variants to be analyzed during design space exploration while simultaneously maintaining the precision required during the exploration process. Significant improvement in speedup during DSE is obtained for different benchmarks, compared to a DSE method with binary search mechanism.

I. INTRODUCTION

Optimizing concurrently multiple performance metrics for the modern generation of complex Very Large Scale Integration (VLSI) and System-on-Chip (SoC) designs requires the selection of an optimal architecture from the large design space, which reflects not only the objective functionality for the target application but also the system requirements specified [1]. Exploring the design space generally involves optimization of conflicting issues: a) speed of the exploration process and b) the precision with which the design points are evaluated. Therefore the selection of the optimal architecture for the system design [2] involves tradeoffs of multiple parameters. As the complexity of the design space increases, heuristic techniques are being widely used to solve the problem of design space exploration. Evolutionary algorithms including Genetic Algorithms have been proposed in [3]. Authors in [4] suggested another approach to DSE based on binary encoding of chromosomes to yield better results. An approach for synthesis of heterogeneous embedded systems using Pareto Front Arithmetic (PFA) was proposed in [1] to explore the giant search spaces. In addition, Pareto optimal analysis was proposed in [5] to efficiently optimize multiple performance parameters for the system design.

II. THE PROPOSED THEORY FOR FUZZY SEARCH TECHNIQUE IN DESIGN SPACE EXPLORATION (DSE)

In the fuzzy set theory, the characteristic function is generalized to a membership function that assigns every element a value e.g. membership value (MV). The membership function \( \mu_F \) of a fuzzy set \( F \) is a function: \( \mu_F : U \rightarrow [0,1] \). In order to present the algorithm, a graphical representation is used, taking into consideration that architectural variants in the design space are already sorted in increasing or decreasing order of magnitude using such DSE approaches shown in [5][6]. The architectural variants of the arranged (sorted) design space are represented in the form of a fuzzy set where each variant has a certain assigned membership value based on the membership function defined below. In the theory presented in this paper, only the variant’s actual minimum and maximum values for the parameter are calculated initially. The MV of the variants between the two extremes will be considered to be directly proportional to the sorted position of the variants in the sorted design space. Hence the MV of the variants of the design space are not calculated from their actual values but are calculated considering that MV of the variants are proportional to their sorted position in the sorted design space (sorted design space is obtained by using approaches like [5][6]). Therefore, the membership value of a variant is calculated using equations (1) or (2), for sorted design space in increasing or decreasing orders respectively:

\[
\tau = \frac{(x - \alpha)}{(\beta - \alpha)} \quad (1)
\]

Or,

\[
\tau = \frac{(x - \beta)}{(\alpha - \beta)} \quad (2)
\]

In equation (1) and (2), ‘\( x \)’ is the position of the variant; ‘\( \tau \)’ presents the membership value (in the fuzzy set) of the variant in the sorted design space. ‘\( \alpha \)’ and ‘\( \beta \)’ are the first and the last element in the same sorted design space, which means ‘\( \alpha \)’ is equal to 1 and ‘\( \beta \)’ is equal to the total number of variants in the sorted design space. In the figures (Fig. 1 and Fig. 2), the x-axes refer to the architectural variants of the design space and the y-axes refer to the actual values and its membership values. ‘\( \tau_{\text{Min}} \)’ is the membership value of the border variant for the parameter in the architecture space. ‘\( \tau_{\text{Max}} \)’ is the membership value for the variant under test and \( V_{\text{Variant}} \) is its respective value. Similarly, ‘\( \tau_{\text{Min}} \)’ and ‘\( \tau_{\text{Max}} \)’ are the membership values for the minimum and maximum variants in the architecture space, while ‘Max’ and ‘Min’ are its respective values. The algorithm has been developed to search for the border variant with its given actual value. The trend line shown in Fig1 represents the increase in membership values of each variant in the design space for area/power.
parameters, while the trend line shown in Fig2 represents the decrease in membership values of each variant in the design space for execution time parameter. The membership values in the presented theory are not calculated based on the actual values of each variant but rather on its position (using equation (1) or (2)) in the arranged design space obtained by using DSE approach in [5][6]. In Fig.1 the increase in membership value for area (actual area increase) is approximated by the straight line (OR) corresponding to the actual border value (VBorder) searching for. ‘V1’ indicates the initial variant obtained corresponding to the calculated initial membership value (τini). ‘P’ is a point in the straight line corresponding to the actual membership value (τv) and the actual variant value (VVariant) of variant ‘V1’. For example, the variant value (VVariant) calculated is less than the value for (VBorder), then the search should be performed between points ‘P’ and point ‘R’. Next a second straight line (PR) is used to approximate the increase in membership values for area/power parameter where in this straight line (PR), point ‘N’ corresponds to the actual border value searching for. Using the similarity between the triangles ΔPNQ and ΔPRS the function described in (3) can be derived:

\[
\frac{\tau_{\text{Max}} - \tau_{v}}{\tau_{b} - \tau_{v}} = \frac{\text{Max} - V_{\text{Variant}}}{(V_{\text{Border}} - V_{\text{Variant}})} \tag{3}
\]

Similar analysis has been made for execution time with decreasing trend line, when the calculated variant value (VVariant) is more than the border value (VBorder). The decrease in trend line execution time has been shown in Fig.2 where ‘M’ refers to the point on that line corresponding to the actual border value (VBorder) searching for. ‘V1’ indicates the initial variant obtained corresponding to the calculated initial membership value (τini). ‘P’ is a point in the straight line corresponding to the actual membership value (τv), and the actual variant value (VVariant) of variant ‘V1’. For example, if this calculated variant value is more than the border value (VBorder), then the search should be performed between points ‘P’ and point ‘O’. Now, a next second straight line is used to approximate the decrease in membership values for execution time. Here, ‘N’ is a point corresponding to the actual border value looking for (VBorder). Using similarity between the triangles ΔMPN and ΔRPO, the following function (4) is derived:

\[
\frac{\tau_{\text{Min}} - \tau_{v}}{\tau_{b} - \tau_{v}} = \frac{\text{Min} - V_{\text{Variant}}}{(V_{\text{Border}} - V_{\text{Variant}})} \tag{4}
\]

Similar analysis has been made for area/power with increasing trend line when the calculated variant value (VVariant) is more than the border value (VBorder). The proposed algorithm is described as follow.

**Algorithm - Searching for the border variant (Border)**

1. Define the Universe of discourse. (The fuzzy set)
2. Identify and define the Linguistic variables
3. Assign the membership values (τ) based on (1) or (2) for each variant in the universe of discourse based on trend line for that parameter (increasing or decreasing).

4. Calculate the initial membership value (τini) based on the function: τini = (VBorder - Min)/(Max - Min) ; where τ is the initial membership value corresponding to border variant (VBorder) ‘Min’ and ‘Max’ are the minimum and maximum value of the variants for that parameter.
5. Look for the variant (V) closest to “τini” in the fuzzy set.
6. Calculate the value of the variant ‘V’ (VVariant).
7. If Vvariant < VBorder then goto step 8, else goto step 10.
8. Solve the membership value (τb) based on equation (3)
10. Solve the membership value (τb) based on function (4)
11. Look for the variant ‘V’ which has the closest membership value to “τb” calculated in step 8 or step 10.
12. If variant ‘V’ has already been checked , then {If Vvariant < VBorder then look for the unchecked variant with the next higher membership value in the set, and jump to step 13. Else if Vvariant > VBorder then look for the unchecked variant with the next lower membership value in the set, and jump to step 13} Else variant ‘V’ has not been checked then go to step 13
13. Calculate the Vvariant.
14. If the ‘Border’ is still not found then repeat step 7.

**III. DETERMINATION OF THE OPTIMAL ARCHITECTURE BASED ON THE GIVEN SYSTEM SPECIFICATIONS**

**A. Problem formulation and system specifications**

For the demonstration of the proposed DSE search technique, digital IIR filter is selected as the benchmark application. The assumed real system specifications are described in Table I. The function of a second order digital IIR Chebyshev filter is given as follows [7]:

\[
y(n) = 0.041x(n) + 0.082x(n-1) + 0.041x(n-2) - 0.6743y(n-2) + 1.4418y(n-1)
\tag{6}
\]
B. Arrangement of the Architecture Design Space in increasing order for Area and the use of the fuzzy search technique to find the border variant for the area

The approach in [5] [6] is used in this paper, to arrange the architecture design space in increasing order for 'area'. The approach calculates the 'criterion for hierarchical arrangement' for creating a hierarchical tree that represents the architecture design space in increasing order for area. Using that approach for DSE, the hierarchical tree obtained is shown in Fig. 3. The nodes at the last level denote the variant numbers after arrangement in increasing orders of magnitude for area. The arrangement of the design space in increasing order helps in pruning the design space in order to obtain the border variant for area. After arrangement of the hierarchical tree, the universe of discourse set is constructed based on the arrangement of the variants in the design space in increasing order in the hierarchical tree (In other words, the set reflects the arranged design space in the tree, as area shown in Fig.3). After defining the universe of discourse set for area, the algorithm described in Section II is followed to obtain the border variant for area. Therefore, the universe of discourse set is represented as \( \text{(Large area (µL))} \):

\[
\{ V5, V8, V16, V25 \}
\]

where 'Large area' is a linguistic variable for the set defined above. Then according to step 4 of the fuzzy search algorithm, the initial membership value \( (\tau_{\text{ini}}) \) is obtained based on the calculated Min and the Max value of area, as well as using the specification of \( V_{\text{Border}} = 160 \text{ a.u} \) specified in Table I. Min = 83 a.u. and Max = 245 a.u. as the minimum and maximum values of the variants with minimum and maximum resources respectively, were calculated using (7):

\[
A = (N_{R_1} \cdot K_{R_1} + N_{R_2} \cdot K_{R_2} + \ldots + N_{R_N} \cdot K_{R_N}) + A(R_{\text{clk}})
\]

where '\( N_{R_i} \)' represents the maximum number of resource '\( R_i \)' and '\( K_{R_i} \)' represents the area occupied per unit resource '\( R_i \) (1 <= i <= n)' while '\( A(R_{\text{clk}}) \)' represents the area of the clock oscillator. As shown in Table II, the fuzzy search technique finds the border variant in just two iterations by analyzing the area of the variants according to equation (7). The border variant obtained is variant 11 (marked in bold in Fig.3 and Table II). This value indicates the last variant in the arranged design space which satisfies the constraint for area (\( V_{\text{Border}} \)).

C. Arrangement of the design space in decreasing order and the use of the fuzzy technique to find the border variant for the execution time parameter

![Figure 3. ACG representing arranged design space in increasing order from left to right extreme for area](image)

TABLE I. SYSTEM SPECIFICATIONS

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Maximum hardware area of resources: 160 area units (a.u)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2) Maximum time of execution: 200 µs (for 1000 sets of data)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) Power consumption: Minimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) Maximum resources available for the system design: a) 3 Adder/Subtractor units. b) 3Multiplier units c) 3clock frequency: 24 MHz, 100 MHz and 400 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5) No. of clock cycles needed for multiplier and Adder/subtractor to finish each operation: 4 cc and 2cc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6) Area occupied by each adder/subtractor and multiplier: 12 area units (a.u.) and 65 a.u. on the chip (e.g. 12 CLB on FPGA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7) Area occupied by the 24MHz 100MHz and 400 MHz clock oscillator: 6 area units, 10 area units and 14 area units.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8) Power consumed at 50 MHz, 100 and 400 MHz: 10milliwatt (mW)/a.u. and 32mW/a.u and 100mW/a.u.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As mentioned in the previous section, the arrangement of the design space in decreasing order helps to obtain the border variant for execution time. Hence the design space is arranged in decreasing order for 'execution time' using the approach [5] [6] in the similar method as the arranged design space was obtained for area. The next step is to apply fuzzy search on the arranged design space to obtain the border variant for execution time. Before applying fuzzy search, the universe of discourse set for time of execution parameter is defined as:

small time of execution set (µ S) = \{ \tau_{F1}, \tau_{F2}, \tau_{F3}, \tau_{F4}, \tau_{F5}, \tau_{F6}, \tau_{F7}, \tau_{F8}, \tau_{F9}, \tau_{F10}, \tau_{F11}, \tau_{F12}, \tau_{F13}, \tau_{F14}, \tau_{F15}, \tau_{F16}, \tau_{F17}, \tau_{F18}, \tau_{F19}, \tau_{F20}, \tau_{F21}, \tau_{F22}, \tau_{F23}, \tau_{F24}, \tau_{F25}, \tau_{F26}, \tau_{F27} \}

where 'small time of execution' is a linguistic variable for the set defined above. (Note that the universe of discourse set for execution time is constructed based on the arrangement of the variants in the design space in decreasing order, similar to the way the set was constructed for area). Next according to step 4 of the fuzzy search algorithm, the initial membership value \( (\tau_{\text{ini}}) \) is obtained based on the calculated Min and Max value of execution time, as well as using the specification of \( V_{\text{Border}} = 200 \mu s \) specified in Table I. The values of Min = 20.01Ìs and Max = 833.41Ìs as the minimum and maximum values of the variants with minimum and maximum resources respectively, were calculated using (8).

\[
\tau_{\text{exe}} = [L + (N-1) \cdot T_c] \cdot T_p
\]

where '\( L \)' represents latency of operation, '\( T_c \)' represents the cycle time, '\( N \)' denotes the number of times the set of data elements needs to be processed and '\( T_p \)' is the time period of the clock. After obtaining the initial membership value the steps of the fuzzy search algorithm are followed to obtain the border variant (variant 5) in just five iterations (see Table III) after evaluation of their respective execution times (using (8)). The variants in the Pareto optimal set obtained after finding the border variant for both parameters are V5, V7, V16, V25 and V8. Now as per the specifications the variant should have minimum power, hence after analysis it was found that variant 'V5' (Area = 152 a.u, T_{exe} = 100.04 µs, Power consumption...
The proposed approach for DSE applied on the benchmarks, yielded superior results in terms of speedup compared to the current DSE approach using binary search technique in [5]. The fuzzy search for DSE introduced in this paper is capable to search the design space faster because of its capability to derive precise results from imprecise (or unknown) complex conditions such as searching a huge design space fast, where the actual values of the variants are unknown. The results of the proposed approach when compared with DSE approach in [5][6] that uses binary search and also with exhaustive search shows considerable speedup of up to 21% and 81% respectively, for a specific benchmark as reflected in Table IV.

V. CONCLUSION

This paper proposed a novel fuzzy search method for rapid design space exploration for VLSI computing architectures. For benchmark applications, the proposed method showed significant improvement in speedup and acceleration when compared to current DSE approach that uses binary search.

### Table II The Variants Obtained for Execution Time After Applying Fuzzy Search on the Arranged Design Space

<table>
<thead>
<tr>
<th>Equations for obtaining the calculated membership values</th>
<th>Calculated membership values(τ)</th>
<th>Corresponding variant in the set according to the calculated ‘τ’</th>
<th>Area</th>
<th>Decision based on the V_{border}</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau = \frac{160 - 83}{245 - 83} )</td>
<td>( \tau_{ui} = 0.475 )</td>
<td>0.462/V11</td>
<td>( A^{14} = 2<em>12 + 2</em>65 + 6 = 160 ) a.u.</td>
<td>( A^{14} &lt; ) V_{border} (search right)</td>
</tr>
<tr>
<td>( 1 - \frac{0.462}{245 - 83} )</td>
<td>( \tau_n = 0.462 )</td>
<td>0.500/V14</td>
<td>( A^{14} = 2<em>12 + 2</em>65 + 10 = 164 ) a.u.</td>
<td>Stop</td>
</tr>
</tbody>
</table>

### Table III The Variants Obtained for Execution Time After Applying Fuzzy Search on the Arranged Design Space

<table>
<thead>
<tr>
<th>Equations for obtaining the calculated membership values</th>
<th>Calculated membership values(τ)</th>
<th>Corresponding variant in the set according to the calculated ‘τ’</th>
<th>Execution time</th>
<th>Decision based on the V_{border}</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau = \frac{200 - 20.01}{833.41 - 20.01} )</td>
<td>( \tau_{ui} = 0.2213 )</td>
<td>0.231/V25</td>
<td>( T_{ext} = (22 \tau (1000-1)*20) * 0.0025 = 50.005 ) µs</td>
<td>( T_{ext} &lt; ) V_{border} (search left)</td>
</tr>
<tr>
<td>( 1 - \frac{0.231}{833.41 - 50.005} )</td>
<td>( \tau_n = 0.378 )</td>
<td>0.385/V15</td>
<td>( T_{ext} = (12 \tau (1000-1)*8) * 0.01 = 80.04 ) µs</td>
<td>( T_{ext} &lt; ) V_{border} (search left)</td>
</tr>
<tr>
<td>( 1 - \frac{0.385}{833.41 - 80.04} )</td>
<td>( \tau_n = 0.483 )</td>
<td>0.500/V14</td>
<td>( T_{ext} = (14 \tau (1000-1)*10) * 0.01 = 100.04 ) µs</td>
<td>( T_{ext} &lt; ) V_{border} (search left)</td>
</tr>
<tr>
<td>( 1 - \frac{0.500}{833.41 - 100.04} )</td>
<td>( \tau_n = 0.568 )</td>
<td>0.577/V22</td>
<td>( T_{ext} = (22 \tau (1000-1)*20) * 0.01 = 200.02 ) µs</td>
<td>( T_{ext} &lt; ) V_{border} (search right)</td>
</tr>
<tr>
<td>( 0 - \frac{0.577}{200 - 200.02} )</td>
<td>( \tau_n = 0.577 )</td>
<td>0.538/V5</td>
<td>(Since V22 has been checked so according to our algorithm check V5)</td>
<td>Stop</td>
</tr>
</tbody>
</table>

### Table IV Results of the Proposed Approach Compared with Approach in [5] That Uses Binary Search

<table>
<thead>
<tr>
<th>Applications</th>
<th>Number of Architecture variants in the Design space</th>
<th>Total possible architecture variants for two parameters</th>
<th>Proposed approach with ACG method and Fuzzy Search technique (Number of variants analyzed)</th>
<th>ACG method with the Binary Search technique [5] (Number of variants analyzed)</th>
<th>Speed up using proposed approach compared with [5]</th>
<th>Speedup compared to exhaustive search</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIR Digital Chebyshev Filter</td>
<td>27</td>
<td>54</td>
<td>7 9 16</td>
<td>9 9 18</td>
<td>11.11%</td>
<td>70.37%</td>
</tr>
<tr>
<td>IIR Digital Filter 1</td>
<td>32</td>
<td>64</td>
<td>7 8 15</td>
<td>9 8 17</td>
<td>11.76%</td>
<td>76.56%</td>
</tr>
<tr>
<td>IIR Digital Filter 2</td>
<td>36</td>
<td>72</td>
<td>7 9 16</td>
<td>11 11 22</td>
<td>27.27%</td>
<td>77.78%</td>
</tr>
<tr>
<td>IIR Digital Filter 3</td>
<td>40</td>
<td>80</td>
<td>7 8 15</td>
<td>10 9 19</td>
<td>21.05%</td>
<td>81.25%</td>
</tr>
<tr>
<td>Discrete Wavelet Transformation</td>
<td>216</td>
<td>432</td>
<td>9 13 22</td>
<td>13 13 26</td>
<td>15.38%</td>
<td>94.90%</td>
</tr>
</tbody>
</table>