

A Projection-based Decomposition in EHW Method for Design of Relatively Large Circuits

Yanyun Tao
School of urban rail transportation,
Soochow University,
Suzhou, 215137, China
taoyanyun@suda.edu.cn

Yuzhen Zhang*
(Corresponding author)
The First Affiliated Hospital of
Soochow University
Suzhou, 215006, China
Yuzhen_Tao@163.com

Lijun Zhang, Chao Gu
School of urban rail transportation,
Soochow University,
Suzhou, 215137, China
tyy8219@163.com

ABSTRACT

Scalability is the most important issue and not well-addressed in EHW field by far. To solve scalability, this paper proposes a novel EHW system called PD-ES, which integrates a Projection-based Decomposition (PD) and Evolutionary Strategy (ES). PD gradually decomposes a Boolean function by adaptively projecting it onto the property of variables, which makes the complexity and number of sub logic blocks minimized. The gate-level approach-CGP including ES searches complete solutions for these blocks. By employing PD into EHW system, the number of logic gates used for evolving and assembling the sub blocks decreases largely, and the scalability can be improved consequently. The MCNC circuits and n -parity circuits are used to prove the ability of PD-ES in solving scalability. The results illustrate that PD-ES is superior to 3SD-ES and fixed decomposition in evolving large circuits in terms of complexity reduction. Additionally, PD-ES makes success evolution in design of larger n -even-parity circuits as SDR has done.

Categories and Subject Descriptors

B. Hardware- B.6 [LOGIC DESIGN]: B.6.3 Design Aids- Automatic synthesis

Keywords

Projection; decomposition; Scalability; EHW; circuit evolution;

1. INTRODUCTION

Evolvable Hardware (EHW) is an adaptive technique of searching a solution for hardware realization. Scalability is the most important and real issue of EHW for a long time that needs to be urgently solved. By far, the main concepts for addressing scalability can be divided into two types: Function-level Evolution[1-2] and Evolutionary Algorithms (EAs) with decomposition strategies[3-6]. Function-level approaches, such as ADFs-GP, ECGP [1], and TSA [2], has been studied and applied to solve modeling of complex system. However, function-level approaches require human intervenes for appropriate modules or functions selection, so that the performance of evolution strongly depends on the selected functions or building blocks. TSA and ECGP can overcome this problem by automatically finding beneficial modules in the evolution process without human intervene. By using decomposition strategies, several approaches intended to tackle scalability have been proposed. The first proposed approach with decomposition to scalability is divide-and-conquer. Bi-directional Incremental

Evolution (BIE) [3] and Generalized Disjunction Decomposition (GDD) [4] are both based on the opinion of divide-and-conquer, etc. As for the parity circuits, the complex circuit evolved up to now is the 17-parity by GDD. Stepwise dimension reduction (SDR) is proposed to solve design of large circuits by reducing dimension of input-output combinations [5]. 3SD-ES aims to solve the scalability in evolutionary design of sequential circuits by input-decomposition, state decomposition and output-decomposition [6].

To our minds, decomposition is a more efficient strategy for scalability, because, it makes the generation of modules (or sub logic blocks) independent on EAs so that it is generalized to different EAs. The main problem of most decomposition strategies used in EHW is that the decomposition is mainly based on Shannon theory without any optimization and large extra gates are needed to assemble the complete circuit. For optimizing the decomposition and obtaining less-complex sub logic blocks, we propose a projection-based decomposition, namely PD, together with Cartesian Genetic Programming (CGP) to evolve and minimize the total complexity of relatively large circuits.

2. PROJECTION DECOMPOSITION

This study proposes a novel framework called Projection Decomposition (PD), including **Shannon projection**, **EXORs projection**, and projection with **remainder** to reduce the complexity of sub logic blocks. A complex Boolean function can be mapped into simple sub functions by projecting it on various properties of variables.

Shannon projection for a Boolean logic function F is based on the theory of Shannon expansion or decomposition. Let $X = \{x_0, x_1, \dots, x_{n-1}\}$ be n variables and $F_{Shan}(X)$ be the logic function obtained by Shannon projection can be denoted as (1).

$$F_{Shan}(X) = x_i \cdot \varphi_{x_i=1} + \bar{x}_i \cdot \varphi_{x_i=0} \quad (1)$$

Where $\varphi_{x_i=1}$ and $\varphi_{x_i=0}$ are the projections obtained by projecting F on $(x_i=1)$ and $(x_i=0)$. The projections, $\varphi_{x_i=1}$ and $\varphi_{x_i=0}$, are actual two sub functions of F . By using Shannon projection, F can be decomposed into two relatively simple functions. Obviously, it is easier for an EHW method to evolve sub functions than directly evolve the original function F .

EXORs projection is based on the concept of Generalized EXORs-Projected Sum of products (GEP-SOP). The function obtained by EXORs projection on a pair of variables (x_i, x_j) can be denoted as (2).

$$F_{EXOR}(X) = (x_i \oplus x_j) \cdot \varphi_{\oplus} + (x_i \oplus \bar{x}_j) \cdot \varphi_{\ominus} \quad (2)$$

Where φ_{\oplus} and φ_{\ominus} are two projections by projecting the Boolean function F onto $(x_i \neq x_j)$ and $(x_i = x_j)$. The difference between Shannon projection and EXORs projection is the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.

http://dx.doi.org/10.1145/2598394.2598411

property of variables used for projection. In EXORs projection, the pair (x_i, x_j) appearing in the largest number of products is selected for projection. If a Boolean function covers more adjacent minterms, Shannon projection is more useful than EXOR projection in terms of complexity reduction. If a Boolean function covers more minterms, which differ from each other by 2 literals, EXORs projection performs better.

Remainder is a set of the low-cost crossing cubes of Boolean function. In some cases, it is more appropriate to project the Boolean function excluding some cubes (products) crossing the projections. These crossing cubes usually have a low-cost (means low complexity) and appear in both projections. A Boolean function produced by not projecting these crossing cubes can obtain a lower complexity. Therefore, these low complex cubes crossing projections are not projected and are put into the remainder.

3. PD-ES

In this study, a gate-level EHW method, Cartesian Genetic Programming (CGP), is used to evolve complete circuits. CGP usually employs Evolutionary Strategy (ES) as its search method for finding solution. According to the previous researches, a gate-level EHW method can successfully achieve solutions in small or middle-scale circuits, but not well-performed in relatively large circuits, e.g. the circuits with more than 10 inputs. Therefore, in order to evolve relatively large circuits, it is necessary to integrate the search method-ES with PD. The integrated EHW system is referred as to PD-ES.

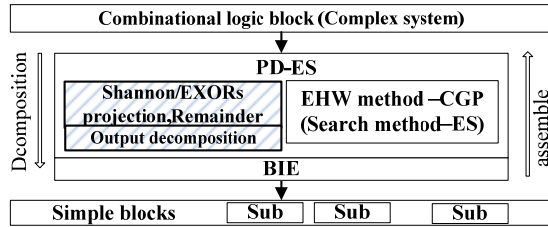


Fig.1 process of PD-ES for evolution of complex circuit

Fig.1 depicts the PD-ES in decomposing and evolving complex circuits. By employment of BIE, PD for decomposition and CGP for evolution are carried out at the same pace. When the fitness improvement of CGP evolution of logic blocks has no progress, PD gradually decomposes them into simpler “Sub” logic blocks, which is enough for CGP to achieve. Additionally, it is necessary for complex circuits with multi-output to use output-decomposition. Unlike PD, the output-decomposition will not generate any extra gate, and it is a complementary to PD. Finally, the evolved “Sub” blocks are assembled to a complete circuit by logic connections.

4. EXPERIMENT AND CONCLUSION

Previous EHW approaches, such as 3SD-ES[6] and SDR[5], are selected for comparison with PD-ES. The experimental circuits in the study are the ones often used in previous researches. They are circuits from Microelectronics Center of North Carolina (MCNC) library, and even n -bit parity circuits from 16-parity to 20-parity. The MCNC circuits usually used in design community are not popular within the EHW community because of the complexities of those circuits. Table.1 and.2 give the performance of PD-ES in these cases. #Gate is the number of gates, Ave Gen and Ave time indicate the average evolution generations and average time, and Best time indicates the best time of evolving the circuit.

In MCNC circuits, the number of gates used by PD-ES is much less than 3SD-ES. This is because the number of sub logic blocks decomposed by our method is much less than 3SD-ES. The total complexity of circuit evolved by PD-ES is significantly reduced. The average generation obtained by PD-ES is more than 3SD-ES in some cases. This is because the number of sub logic blocks decomposed is small so that the evolution time for sub logic blocks increases. Unlike 3SD-ES, the average generations obtained by PD-ES have a small difference in different circuits, because the decomposition in PD-ES is based on the complexity so that the decomposed sub logic blocks have a similar complexity and need similar evolution time. In even parity circuits, PD-ES finds a solution with 7,139 logic gates of 20-parity circuit for 2,636,833 generations. Maybe, SDR can also successfully find a solution for 20-parity circuit, but it has not been reported by far. The total evolution time obtained by PD-ES is superior to SDR in 18 and 19-parity circuits in terms of Ave time. The results demonstrate that PD-ES has the ability to perform better in terms of complexity reduction.

Table.1 Achievable performance of PD-ES in MCNC circuits

Circuit	3SD-ES[6]		PD-ES	
	#Gate	Ave Gen	#Gate	Ave Gen
Bbsse	2,610	25,547	596	1,995,989
Cse	2,562	84,146	760	1,525,126
Tbk	4,402	1,182,301	1,271	1,337,316
Ex1	18,496	136,961	7,155	2,740,719
Styr	17,385	337,287	11,520	2,254,123
Planet	15,252	716,784	17,976	2,990,023

Table.2 Achievable performance of PD-ES in n -parity circuits

Circuit	SDR[5]		PD-ES	
	Best time	Ave time	Best time	Ave time
16-parity	77.454	566.247	1,315.20	1,961.21
17-parity	150	1,665	3,004.95	3,310.66
18-parity	700.54	5,454	4,507.09	5,071.08
19-parity	1,455	13,801	9,297.92	9,425.89
20-parity	N/A		40,221.34	53,104.18

5. ACKNOWLEDGMENTS

This research is supported by the Natural Science Funds of the Jiangsu Education of China (Grant No. 13KJB520023) and the National Natural Science Funds of China (Grant No. 61272105).

6. REFERENCES

- [1] J.A. Walker, J.F. Miller, Investigating the performance of module acquisition in cartesian genetic programming, in *GECCO2005*, Washington, pp:1649–1656. 2005
- [2] YY. Tao, YZ. Zhang, J. Cao, A module-level three-stage approach to the evolutionary design of sequential logic circuits. *Genet Program Evolvable Mach.* 14(2): 1-29. 2013.
- [3] T. Kalganova, An extrinsic function-level evolvable hardware approach, in *EuroGP2000*, LNCS, Vol.1802:60-75. Springer-Verlag. 2000.
- [4] E. Stomeo, T. Kalganova, C. Lambert, Generalized disjunction decomposition for evolvable hardware. *IEEE Trans. Syst. Man Cybern. Part B.* Vol.36 (5):1024–1043. 2006.
- [5] ZF Li, WJ Luo, XF Wang, A Stepwise Dimension Reduction Approach to Evolutionary Design of Relative Large Combinational Logic Circuits. in *ICES 2008*, LNCS Vol. 5216, pp: 47–58, 2008.
- [6] HJ. Liang, WJ Luo, XF. Wang .A three-step decomposition method for the evolutionary design of sequential logic circuits. *Genet Program Evolvable Mach.* Vol.10:231–262. 2009.