Journal of Systems Architecture 60 (2014) 119–139

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

# Multi-objective module partitioning design for dynamic and partial reconfigurable system-on-chip using genetic algorithm

Nithiyanantham Janakiraman *, Palanisamy Nirmal Kumar

*Department of Electronics and Communication Engineering, College of Engineering Guindy, Anna University, Chennai 600025, Tamil Nadu, India*

ARTICLE INFO

*Article history:*
Available online 17 October 2013

*Keywords:*
Multi-objective problem
Module partitioning solution
Genetic algorithm
SoC
FPGA
Dynamic and partial reconfiguration

ABSTRACT

This paper proposes a novel architecture for module partitioning problems in the process of dynamic and partial reconfigurable computing in VLSI design automation. This partitioning issue is deemed as Hypergraph replica. This can be treated by a probabilistic algorithm like the Markov chain through the transition probability matrices due to non-deterministic polynomial complete problems. This proposed technique has two levels of implementation methodology. In the first level, the combination of parallel processing of design elements and efficient pipelining techniques are used. The second level is based on the genetic algorithm optimization system architecture. This proposed methodology uses the hardware/software co-design and co-verification techniques. This architecture was verified by implementation within the MOLEN reconfigurable processor and tested on a Xilinx Virtex-5 based development board. This proposed multi-objective module partitioning design was experimentally evaluated using an ISPD'98 circuit partitioning benchmark suite. The efficiency and throughput were compared with that of the hMETIS recursive bisection partitioning approach. The results indicate that the proposed method can improve throughput and efficiency up to 39 times with only a small amount of increased design space. The proposed architecture style is sketched out and concisely discussed in this manuscript, and the existing results are compared and analyzed.



## 1. Introduction

Genetic algorithm (GA) is a flexible scheduling and optimization technique based on the natural selection process. Genetic Algorithm is being exercised to several rigid scheduling, partitioning and optimization problems. For example, hardware/software co-design, VLSI design automation, layout optimization, fault tolerance, solutions of multifaceted Boolean algebraic equations, dynamic scheduling and partitioning solutions in partial reconfigurable architectures. The examples are vigorous broad-spectrum of scheduling and optimization techniques. But appliance of software-based GA to composite troubles escalates intolerable delays in the scheduling and optimization processes. When the breathing space is large, the above-intolerable delays are factual for every non-trivial applications like reconfigurable computing, dynamic and partial reconfiguration of integrated circuit architecture. These types of complex problems realized during module partitioning in static, dynamic and partial reconfigurable architectures can be overwhelmed through hardware and software based GA.

Here, the objective is to create a novel architecture and to define an optimum method to implement the GA-based solutions for partitioning issues in dynamic and partial reconfigurable systems. These partitioning problems are assumed to be Hypergraph model, because of non-deterministic polynomial (NP) hard problems. NP hard is a class of decision-making problems. Exponential algorithms are not useful to solve these problems. So a probabilistic algorithm can be used through the transition probability matrices like Markov chain. This can be achieved by the proper arrangement of pipelining and parallel processing of the system components with the fine-tuned control entity and hardware/software co-design methods. This effort demonstrates the feasibility of resolving the module-partitioning problem using hardware/software co-design based GA engine.

This is the extension work based on other research in dynamic reconfigurable computing and hardware realization of GA. Enhanced system performance could be achieved by mapping the designed software components to reconfigurable hardware (Xilinx Virtex 5 FPGA). This article has seven sections as follows: Section 1 gives an overview of this article. Section 2 discusses the overview of module partitioning issues, genetic algorithm, reconfigurable computing and literature survey. Section 3 describes the proposed architecture design and its functionalities. Section 4 briefs about the various issues in dynamic partitioning and placement solutions. Section 5 explains the experimental set-up, evaluation methodology, simulation and implementation results of this proposed structure and the performance comparison with previous work.

* Corresponding author. Tel.: +91 9894424665.

1383-7621/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sysarc.2013.10.001

Section 6 concludes the paper and discusses the possibilities of further extension. "References" section lists the reference papers.

## 2. Background

### 2.1. Introduction to partitioning issues

Partitioning of modules and circuits is a task or course of dividing a system structure and functionality into smaller elements [1]. This process may be implemented in physical and/or logical level separation of components. This system partitioning has some limitations like module size, functionality level and inter-connection complexity.

In general, partitioning issues can be classified as constructive and iterative methods. Constructive algorithms are used to determine a partition from the graph description of the circuit or system. But iterative algorithms have aim to improve the quality and the features of an existing solution of partitioning issues. Constructive partitioning method is to be used in this paper. So this proposal can be applied in spectral and clustering methods, network flow computations, placement-based partitioning, mathematical programming and eigenvector methods.

On the other hand, partitioning algorithms can be tagged as deterministic or probabilistic methods. While execution, deterministic algorithms generate the identical solutions each time, whereas, probabilistic algorithms produce different solutions each time, because probabilistic algorithms are based on random numbers. In multi-objective optimization, more than one objective function (or) solution have to be optimized simultaneously in the presence of trade-offs between two or more conflicting objective functions. Most of the realistic problems should have multiple-objectives (i.e. minimization of cost, maximization of performance, maximization of reliability, etc.).

The improvement of a particular objective function value has impairment with some of other objective function values. Hence, it has a (possibly infinite number of) Pareto optimal solutions (or) non-dominated solutions. To generate such Pareto optimal solutions, the evolutionary algorithms are popular approaches. The genetic algorithm (GA) is one of the popular meta-heuristic evolutionary algorithms and it is well-suited for this class of multi-objective optimization problems [2]. This paper uses the genetic algorithm method to solve the various partitioning issues. Genetic algorithm is based on the random selection or numbers. So it comes under the probabilistic method.

Partitioning is a significant solution to obtain the efficient place and route with near-optimal layouts. Efficient place and route are necessary to increase the speed of operation and to reduce the routing congestion and wire-length. These are essential to minimize the number of cuts and to trim down the deviation of assigned elements (fan-in, fan-out, intermediate outputs and logical gates) to every partition. In the VLSI physical design automation, the module level partitioning is considered as Multi Chip Module (MCM) partitioning [3].

Most of the previous researchers have not concentrated on genetic algorithm-based hardware/software co-design approach along with the combination of pipelining and parallel processing techniques to solve the Hypergraph model of multi-objective partitioning issues. This paper deals this angle to solve the multi-objective module partitioning problems.

### 2.2. Problem definition

Let a module M be noted by a netlist or Hypergraph G = (V,E), where V represents a set of vertices or nodes and E represents a set of edges or hyperedges of the module. Each hyperedge is connected to two or more nodes by a net. This paper indicates 'n' to denote the number of nodes in V and 'e' to denote the number of hyperedges in E. The capability of a node and a net are indicated by $q_n$ (average number of nets connected to a node) and $q_e$ (average number of nodes connected to a net) respectively. The average number of neighbors contained by a node is calculated by $a = q_n(q_e - 1)$. If two nodes (u and u) are connected by a common net, then only these nodes are said to be neighbor to one another.

A k-way partitioning of G is a set of subsets of V, $H^k = \{V_1, V_2, \ldots V_k\}$. So each node u is an element of V [$u \in V$] and it belongs to exactly one $V_i$.

Let $r_1$ and $r_2$ be two floating point numbers which has a value between 0 and 1. This proposed method fixes the values of $r_1$ and $r_2$ to balance with each other like $r_1 = 1/k$ and $r_2 = 1/k$. So the balanced k-partition can be obtained like $r_1 = |V_i| = r_2$ for each subset $V_i$ of $H^k$. This paper considers the various problem sizes like $k = 4, 8, 16, 32$ and 64, i.e. $k = 64$ and $r_1 = 1 - r_2$. If $k = 65$ then $r_1$ and $r_2$ has no relationship, except the condition $r_1 = r_2$. For the proper balance condition, this proposal has the assumption that all nodes have to be in same unit size.

The k-way partitioning cut-set is defined as

$$E_{cut-set} = \{n_t \in E | \exists_u \in n_t \text{ s.t. } u \in V_{i,j} i \neq j\} \quad (1)$$

here $n_t$ represents a net 't' which can connect a set of nodes. 's.t.' indicates the general mathematical short form of 'such that'. This $E_{cut-set}$ is the set of nets that connect each node to fit into different subsets $\{H^k\}$. The cost value of the cut-set of $\{H^k\}$ is defined as,

$$\text{cost}\left(H^k\right) = \sum_{i=1}^{k} M(n_t), \text{where } n_t \in E_{cut-set} \quad (2)$$

It is used to calculate the cost value of each subset of $\{H^k\}$ and $M(n_t)$ represents the module of net $(n_t)$ which depends on the optimization criterion of partitioning a module. Assume the Hypergraph G contains 'n' nodes and these nodes are being partitioned into $\{V_1, V_2\}$.

### 2.3. Objectives of this partitioning problem

This problem is the multi-objective optimization problem. It deals with the following three objectives.(i) Direct cost minimization:(a) Minimize the cost (C) of each cut in the k-way partitioning:'C' is defined as,

$$C(V_1, V_2, V_3, \ldots, V_k) = \sum_{e_i \in E(V_1, V_2, V_3, \ldots, V_k)} C_i \quad (3)$$

(b) Minimize the cost (C) of sum of all subsets: Subsets can be defined as,

$$\sum_{I=1}^{k} C(V_i) = \sum_{i=1}^{k} \sum_{e_j \in E(V_i)} C_j \quad (4)$$

(c) An identical probability value (f) is used to build the nets which can connect two modules. The expected cost value E (C) can be defined as,

$$E(C(V_1, V_2, V_3, \ldots, V_k)) = f \times \sum_{i=j+1}^{k} \sum_{j=1}^{k-1} |V_i| \times |V_j|] \quad (5)$$

The expected value of cut ratio $(E(R_C))$ for all cuts is having the same value, because of constant probability 'f' value.

**Note:** In the practical implementation of this system uses the genetic algorithm to generate this identical probability value (f).
(ii) Cut minimization(a) Ratio cut:

Each subset $V_i$ has an upper limit $S_U$ and a lower limit $S_L$ values, i.e. $S_L \leqslant S(V_i) \leqslant S_U$. If some of the subset does not have this bound limitation, then the cut ratio $(R_C)$ may be defined as,