INTEGRATION, the VLSI journal 52 (2016) 335–346

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

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

Multi-supply voltage (MSV) driven SoC floorplanning for fast design convergence

Zhufei Chu, Yinshui Xia*, Lunyao Wang

School of Information Science and Engineering, Ningbo University, Ningbo, China

ARTICLE INFO

Available online 25 September 2015

Keywords:
Multi-supply voltage (MSV)
Floorplanning
Level shifter
System-on-a-chip (SoC)
Physical design

ABSTRACT

With the ever-increasing power demands of consumer electronics and portable devices, multi-supply voltage (MSV) technique is supposed as one of the direct and effective ways for power optimization in SoC designs. To realize MSV implementation, procedures such as voltage assignment, voltage island partitioning and level shifters (LSs) placement should be considered simultaneously during the floorplanning stage. Although many works addressed the MSV-driven design problem, few of them actually took account of LS placement, which makes the generated results may limit the potential applications. Furthermore, existing design frameworks are often very computationally expensive, and it is not beneficial to shorten the time to market. In this paper, we present an MSV-driven SoC floorplanning framework for fast design convergence. Several techniques are proposed and integrated into an efficient and flexible non-randomized floorplanning algorithm. Firstly, to reserve the desired deadspace for the placement of LSs, the netlist is modified by assigning *virtual LSs* in the nets. Secondly, a heuristic based voltage assignment method is presented for accuracy and execution time trade-off. Thirdly, different from previous works which do voltage assignment without physical information feedback, an inner loop is built between voltage assignment and LS placement under the constraints of both timing and physical layout. Experimental results on Gigascale Systems Research Center (GSRC) benchmark suites indicate the proposed approach can improve power saving by 12%, CPU time by 48% with 4% area increase.

1. Introduction

With the tremendous progress of technology scaling, increasing numbers of transistors or intellectual property (IP) cores are integrated into a single chip to enable centralized functions, especially for those portable electronic systems. The accompanied higher power density will cause serious thermal problem, which affects the chip performance and battery life. It is reported that the power demands of microprocessors double every 36 months, whereas the performance of batteries is doubling almost every ten years [1]. This widening power gap makes low power system-on-a-chip (SoC) design become an urgent requirement.

Dynamic power is proportional to the square of the supply voltage. By assigning lower supply voltage for non-critical modules to balance the requirements of performance and power [2], multi-supply voltage (MSV) technique is an increasingly attractive design option for system-on-a-chip (SoC), and is widely used by industry for its effectiveness. The principle behind this is the trade-off between timing slacks and power saving to guarantee the chip performance. However, the MSV technique also brings several design challenges for physical design.

First, given the clock cycle time, the netlist of modules, and several different voltage choices, *voltage assignment* is to assign each module with a supply voltage to optimize total power consumption under timing constraints. Chang and Pedram proved that the voltage assignment problem was an NP-hard problem [3], even each module has at most two choices. To obtain an optimal or sub-optimal solution may consume huge amount of time when the design contains large number of modules. Therefore, an efficient voltage assignment algorithm, which should take into account both accuracy and CPU time, is required to consider MSV at the floorplanning stage.

Second, to ensure the right functionality, level shifters (LSs) should be inserted when low voltage modules driving high voltage modules. Since the LSs have area and timing overhead, an effective LS planning (LSP) algorithm is desirable to optimize power consumption under both the timing and physical constraints. There are several challenges in LSP. First of all, as the required number of LSs can only be determined after the voltage assignment stage, the final LSs requirement is unknown in advance. Hence, to allocate "just enough" deadspace resource is a tough task. Moreover, the

* Corresponding author.
E-mail addresses: chuzhufei@nbu.edu.cn (Z. Chu), xiayinshui@nbu.edu.cn (Y. Xia), wanglunyao@nbu.edu.cn (L. Wang).

http://dx.doi.org/10.1016/j.vlsi.2015.09.002
0167-9260/

placement of LSs has a significant impact on timing. If the LSs are placed at improper locations, the timing closure may hard to satisfy or power optimization quality is degraded.

Third, the MSV-driven floorplanning is much more complex than traditional floorplanning since voltage assignment, LSs placement, voltage island (VI) partitioning should be considered simultaneously to achieve multi-objective optimization. The complexity of floorplanning means that systems may take days or hours to run on designs, which severely hurts design productivity. To make fast design convergence feasible, we need not only an efficient and flexible floorplanning algorithm but also efficient and effective voltage assignment and LSP algorithm.

This paper presents a methodology for MSV driven SoC floorplanning with emphasizes on fast design convergence. To achieve such a goal, several techniques, which aim to reduce design iterations or CPU time, are proposed and integrated into a non-randomized-searching-engine based floorplanning algorithm to produce high-quality solutions. The proposed method consider voltage assignment, LSP and VI partitioning simultaneously under both timing and physical constraints during floorplanning. The contributions of our work can be summarized as follows:

- Voltage assignment is an imperative step for the subsequent VI partitioning and LSP. Due to its NP-hard characteristic, to obtain an optimal solution by integrate liner programming (ILP) formulation [4] or value-oriented branch-and-bound (VOBB) algorithm [5,6] may not achievable in a reasonable amount of time. As a result, the existing optimal voltage assignment method is not suitable for multi-objective co-optimization. Against this issue, we propose a heuristic-based voltage assignment algorithm for the seek of accuracy and CPU time balance.
- Since the LSs should be placed in the deadspace of the floorplan, apart from the inherent deadspace generated by the floorplanner, we propose a heuristic method to assign virtual LSs in the netlist to reserve the required deadspace.
- To satisfy both timing and physical constraints, the voltage assignment and LS placement are operated iteratively until no constraint violations exist, which is different from previous works that did voltage assignment without the feedback of LS's physical information.
- The proposed voltage assignment and LSP algorithms are integrated into a highly efficient floorplanning algorithm for fast design convergence, which is beneficial for time-to-market reduction and design productivity improvement.

The remainder of this paper is organized as follows. We first give a brief review of related works in Section 2. Then, the preliminaries, problem formulation, and discussions on the previous works are described in Section 3. The main approach including algorithm overview, details of voltage assignment, LSP, and floorplanning implementation details are demonstrated in Section 4. The experimental comparisons are shown in Section 5. Finally, a summary of the research findings is concluded.

2. Related works

To address the MSV-driven design challenges, considerable literature works are presented to solve voltage assignment, MSV-driven floorplanning, and VI partitioning issues. Next, we give a brief review of related works which will be discussed by classifications of MSV implementation phases, timing constraints modeling, and LSs placement.

According to MSV implementation phases, current researches mainly focused on co-optimization during floorplanning stage [6–11] and post-optimization at the post-floorplanning stage [12–15]. Given optimized floorplan solution which generated by available floorplanner, the post-optimization can be easily integrated into current design flow to further optimize power and other parameters. But the resulted solution quality varies due to the given floorplan. In contrast, the co-optimization has larger solution space than the post-optimization. Consequently, the quality of resulted solutions can be significantly improved.

In terms of timing constraints modeling, the literature works generally adopt legal-voltage-levels-set [7–9,16–19] or delay-power (DP) curve [6,20,21] models. The former one define each module has a set with several discrete legal voltage levels, on which the module can satisfy the performance requirements. For instance, the legal voltage levels of module sb_1 are 1.0 V, 1.1 V, and 1.5 V; while the module sb_2 may work on 1.1 V, 1.2 V, and 1.5 V. By finding lowest common voltage level among those modules with adjacent physical locations, this group of works generate VIs with contiguous regions to optimize power saving and power-network-resources (PNRs). For DP curve model, the power-delay trade-off in each module is represented by a curve. It is consistent with the facts that modules working on higher supply voltage have relatively lower timing delay. To maximize power saving under timing constraints, the DP-curve modeling try to lower supply voltage of modules as many as possible by utilizing timing slacks. The regions of the resulted VIs may be not contiguous.

Although many works described above addressed the MSV-driven design problem and considered the impact of LSs. But few of them actually consider the LS placement, which makes the generated results may limit the potential applications. Yu et al. [20] first considers the LS placement problem during floorplanning. For each candidate solution, the voltage assignment and level shifter placement are carried out. They restrict the LSs must be placed in the deadspace which inherent generated by the floorplan. Hence, it may cause extra wirelength and timing overhead if there are no proper spaces for budget. Zhang et al. [22] proposed an LS floorplanning algorithm for a given multi-voltage design. Based on sequence pair representation, the LSs are greedily pre-placed in the deadspace. Then, a floorplanning algorithm is used to re-optimize the locations of LSs. Lin et al. [23] proposed two ways, named LS channel and LS island, to allocate regions for LSs during floorplanning and then a two-stage approach is adopted to place LSs. However, both [22,23] place LSs under the assumption that the operation voltage of each module is determined and the LS impact on timing is not considered. In contrast, Lee et al. [24] proposed a three-stage algorithm to handle the voltage assignment problem under timing constraints. They first do voltage assignment in netlist level by dynamic programming heuristic. Then the netlist is updated by inserting LSs in the required nets. At last, a PNR aware floorplanning algorithm is applied to place the modules and LSs in the netlist. The main drawback is the voltage assignment which is done without the feedback of physical information, which may prevent better solutions.

Based on above discussions, it can be seen that voltage assignment, non-contiguous regions VI partitioning, and LSP are the main obstacles for fast design convergence. In our work, we investigate voltage assignment and physical constraints of LS that account for design convergence and power consumption. We first develop probabilistic based virtual LS insertion method to reserve the required place for LSs. Then, taking advantages of enough deadspace, the voltage assignment and LS placement are operated iteratively to make both timing and physical constraints satisfied. Moreover, a heuristic based voltage assignment method is proposed to trade-off accuracy and CPU time.

3. Preliminaries

In this section, the preliminaries on LSs, our problem formulation, and limitations of the previous works are demonstrated.