Min-Cost electronic design using tabu search

Marc Sevaux∗ Kenneth Sørensen† Alok Singh‡

∗LESTER, University of South-Brittany
Centre de Recherche, F-56321 Lorient, France
marc.sevaux@univ-ubs.fr

†Centre for Industrial Management, KU Leuven
Celestijnenlaan 300A Bus 2422, B-3001 Heverlee, Belgium
Kenneth.Sorensen@cib.kuleuven.be

‡Dept. of Computer and Information Sciences, University of Hyderabad
Hyderabad - 500046, India
alokcs@uohyd.ernet.in

1 Problem definition

The High-Level Synthesis (HLS) problem consists in transforming a source code (e.g. in the C or VHDL language) so that an electronic component can execute it as many times as required. Our goal is to help the decision maker to design a FPGA (field programmable gate array). This will be done by the use of specific libraries that combine logic programmable components into basic operators (adders, multipliers, etc). A complete introduction to HLS can be found in [1]. GAUT [2] is such a HLS platform and a major algorithmic phase transforms the source code into a large graph of basic operations.

The input is a set of operations (tasks) that have to be scheduled on different resources. For each task $j$, a duration $p_j$, a release date $r_j$, a due date $d_j$, a specific type of resource $t_j$ and a set of preceding tasks are considered. An unlimited set of resources is available to solve the problem. Two kinds of resources can be considered: the single purpose resources that are only of one type (typically adders and multipliers), and the multi-purpose resources (ALU for example). For each kind of resources a specific cost is given. We denote by $s_j$ the starting time of the job $j$. Job $j$ cannot start before its release date ($s_j \geq r_j$) and is completed before its due date ($s_j + p_j \leq d_j$).

2 Proposed methodology

Based on previous work [3], we have extended the approach to deal with new features like precedence constraints, multi-purpose resources and resource costs.

Montreal, Canada, June 25–29, 2007
Initial heuristic  A heuristic will first solve the problem by sequencing the jobs one by one (in a predefined priority order) on the first available resource, respecting the precedence constraints, the release dates and the due dates. When necessary a new resource of the desired type is added. With this heuristic, only single purpose resources are used. The tabu search phase will be in charge of inserting multi-purpose resources.

A two-phase procedure  For solving this problem efficiently, a two-phase procedure is developed. The first phase will try to reduce the number of resources and the second phase will sequence the jobs on the resources.

The first phase will simply consists in scanning potential replacements of several single purpose resources with multi-purpose resources. Since the objective is to reduce the total cost, it may happen that too many resources are removed. In that case, a tabu search procedure will do its best to reschedule the jobs on the other resources.

As presented in [3], starting times of the sequenced jobs are computed only when necessary. We only ensure at each iteration that the current partial schedule respects all the constraints (time windows, precedence and resource type). For each resource, a list of jobs that are sequenced on that resource is kept in memory. This list is the partial order of the jobs on that resource. In addition, a list of “not yet sequenced” jobs is also maintained.

To begin the tabu search procedure, we start with the solution given by the heuristic. In that solution, we look in the system where some resources can be removed (or replaced by others). Jobs initially sequenced on those resources as put in the “not yet sequenced” job list and the tabu search will try to set then on time by making some exchanges. If a valid solution is found, some resources are removed and the tabu search procedure is re-started. Preliminary results show the efficiency of the proposed approach.

3  Extensions

This study is a first moderate step towards the use of efficient heuristic methods for HLS design. To be sure to attract researchers from the electronics community, it is necessary to go further. We will extend our work in several directions: bi-objective optimisation (considering additional objectives like $C_{\text{max}}$ and the total flow of operations) and memory allocation where we will take into account the memory as a limited resource for storing data.

References


Montreal, Canada, June 25–29, 2007