Resource Binding for High Performance LSI with Enhanced Plst-Silicon Skew Tunability

Yosuke Haruta (0910050)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 6, 2012

Keywords: timing-adjustability, programable delay elements, resource binding, High level synthesis, delay variation.

With the progress of fabrication-process technology, the variation of signal transmission delay due to variations in process parameters becomes larger, and the yield degradation due to timing errors comes to be a serious problem in recent large scale integration (LSI). To overcome this problem, various pre-silicon and post-silicon approaches have been proposed. One of the typical pre-silicon approaches is the statistical static timing analysis (SSTA) and its application to design optimization. The conventional worst-case timing analysis is based on a simple accumulation of worst-case gate delays, and it results in too much timing margin to enjoy the merit of high speed devices in current LSIs. On the other hand, SSTA provides us a statistical information about delay distribution, and it allows us to make more aggressive decisions based on the trade-off between performance and yield. Post-silicon approaches include V_{th} tuning for transistors, clock skew tuning (post-silicon skew tuning: PSST), etc. In V_{th} tuning, one tries to tune transistor's characteristic, and compensate varied delays somewhat directly. On the other hand, PSST concentrates on satisfying timing constraints for correct operation.

In PSST technique, programmable delay elements (PDEs) are inserted on a clock distribution tree as a part of LSI circuit, and clock timing skew at each flip-flop is tuned chip by chip through these PDEs in accordance

with the actual delays of each LSI chip. Originally intentional skew has been introduced in pre-silicon design for improving the speed performance of a sequential circuit higher than the conservative speed limit given by the largest path delay between flip-flops. PSST also succeeds to the ability of the original intentional skew, and makes it possible to bring out the best performance inherent in each individual chip.

In this research, we focus on PSST technique. The yield of LSI chip with PSST mechanism depends on the probability of success in timing-skew tuning through PDEs, which we call *skew tuning success rate*. Our objective in this research is to develop a new design methodology and algorithm for a LSI with maximized skew tuning success rate. High level synthesis affects topological structure and the timing behavior of LSI circuit, so we focus especially on high level synthesis for our objective.

Resource binding is a task to assign operations and variables to functional units and registers, respectively, and is one of the major sub-tasks in the high level synthesis. Usually it tasks an application algorithm to be implemented and its operation schedule as an input description. The key issue in this resource binding is to assign operations (variables) to the minimum or specified number of functional units (registers) while lifetimes of operations (variables) assigned to the same functional unit (register) do not conflict each other.

A popular functional unit and register binding algorithm is the one called *left edge* algorithm. *Left edge* can provide a solution with the minimum numbers of functional units and registers for an acyclic application algorithm. However, timing skew is not considered in this algorithm. In recent years, *minTc* and *color* have been proposed for functional unit and register binding considering timing skew. *MinTc* assumes fixed intentional skew, but not post-silicon skew tuning. *Color* is only one which assumes PSST up to now. However, it does not evaluate *skew tuning success rate* directly, but it uses only a roughly approximated design objective.

In this paper, we discuss functional unit and register binding in which the *skew tuning success rate* is directly utilized for decision making. In principle, there is a naive solution; brute force enumeration of binding solutions. But it is unrealistic because of its exponential order solution space.

Our first proposal named *pair-wise merge* begins with a trivial resource binding in which every operation (every variable) is bound to different functional unit (different register). After that, we repeat the pair-wise merge of two functional units (two registers) until the resultant number of functional units (registers) reachs to the prescribed number, or the candidate of pair-wise merge is no longer found. Every time when we apply *pair-wise merge*, first we enumerate possible pair-wise merges based on lifetime collision check, then we evaluate *skew tuning success rate* for each candidate of pair-wise merge, and finally we choose the best pair-wise merge which achieves largest skew tuning success rate. The algorithm is very simple, but has many draw backs; the minimum (or specified) number of resource is not guaranteed, and the resultant *skew tuning success rate* tends to decrease suddenly and rapidly as iteration goes on.

Our second proposal named *parallel left edge* guarantees the minimum (or the specified) number of resource while the decisions made during resource binding are based on the evaluation of the *skew tuning success rate*. Conventional *left edge* algorithm always considers a single resource at a time, and the assignment is determined for one resource to another. Our *parallel left edge* algorithm proceeds at one control step to another, from the first control step to the last. At each control step s, operations (or variables) starting at s are going to be assigned to functional units (or registers) which are idle (not occupied by any operation (variable) which starts at control step $s-1$ or earlier) at control step s. To do this, first we evaluate *skew tuning success rate* for each possible binding of one candidate operation (variable) to one functional unit (register). Next, we construct a complete bipartite graph with partite sets X_s: the set of candidate operations (variables) and Y_s: the set of idle functional units (registers) with evaluated *skew tuning success rates* as edge weights. That is, skew tuning success rate evaluated for the case that an operation $x_i \in X_s$ is assigned to $y_j \in Y_s$ is associated with an edge (x_i, y_j) as its edge weight. Then we compute the maximum matching (a perfect matching from X_s) which maximizes the minimum edge weight in the matching, and we adopt resource binding corresponding to matching edges.

Our resource binding algorithm *parallel left edge* as well as previous methods *left edge*, *minTc* and *color* are applied to several benchmark circuits,

and it is found that our *parallel left edge* always generates better solutions than the conventional *left edge* does, but worse solutions than *minTc* and *color* except for some instances. Those worse solutions are thought to come from the poor ability of our *parallel left edge* in finding globally better solutions. On the other hand, for some instances, our *parallel left edge* can generate better solutions than *minTc* and *color*, which indicates the importance of decision making based on the exact evaluation of *skew tuning success rate*.

The development of a better heuristics but still guided by the exact evaluation of *skew tuning success rate* is our next research target. Operation schedule considering *skew tuning success rate* and the incorporation of a statistical timing analysis for improving the estimation of *skew tuning success rate* remain as future problems.