requires a very slow cooling schedule, which can make SA prohibitively slow. Therefore, an efficient algorithm for the placement problem is desirable.

A placement is acceptable if 100% routing can be achieved within a given area. This is not an easy task for FPGA architectures with very limited routing resources, like the *Atmel 6000* circuits [3]. A good placement will not only put connected blocks together, but will also ensure that logic elements are not placed too closely in order to ensure the routability of the circuit. Our first contribution is an algorithm which not only reduces the cut size and balances the size of the two portions, but also evenly distributes the connections among them. Our second contribution is another sequence of slicing lines which is more adequate for FPGA circuits than the traditional methods. This sequence reduces the maximum cut size and the total wirelength of the circuit.

The rest of this paper is organized as follows. In the next section, we define the placement problem. In Section 3, we briefly describe some related work. In Section 4, we describe the placement algorithm for the FPGA circuits. Experimental results are presented in Section 5. Finally, conclusions are made in Section 6.

2. The Placement Problem

Given a collection of cells or modules with ports on the boundaries and a collection of nets (which are sets of ports that are to be connected together), the process of *placement* consists of finding suitable physical locations for each cell on the entire layout. The locations are suitable if they minimize given objective functions, subject to certain constraints imposed by the designer, the implementation process, or layout style. The cells may be standard-cells, macro-cells, FPGA logic blocks, etc.

More formally, the placement problem can be defined as follows. Given a set of m modules, $M = \{M_1, M_2, \ldots, M_m\}$, a set of n nets $N = \{N_1, N_2, \ldots, N_n\}$, and a set of p primary input pins and primary output pins $R = \{R_1, R_2, \ldots, R_p\}$, we associate with each module $M_i \in M$ a set of nets N_{M_i}, where $N_{M_i} \subseteq N$. Similarly, we associate with each net $N_i \in N$ a set of modules M_{N_i}, where $M_{N_i} = \{M_j \mid N_i \in N_{M_j}\}$. We are also given a set of locations $L = \{L_1, L_2, \ldots, L_k\}$, where $k \geq n$. The placement problem is to assign each $M_i \in M$ to a unique location L_j such that some objective function is optimized. Usually each module is considered to be a point, and if M_i is assigned to location L_j then its position is defined by the coordinate values (x_j, y_j). Sometimes a subset of the modules in M are fixed, i.e., pre-assigned to locations, and only the remaining modules can be assigned to the remaining unassigned locations.

Depending on the technology used, different physical placement constraints exist. For gate-array technology, all modules have the same shape and size and are to be placed into predetermined locations on the placement area. For macro-cell technology, modules have different shapes and sizes, and the dimensions $w_i \times h_i$ of M_i for all the modules are given in the circuit specification. The placement area has dimensions $W \times H$ and is given in the circuit specification.

For performance driven placement, timing specifications are also given. Timing specifications of a circuit include signal arrival times at the primary inputs, the required signal arrival times at the primary outputs, internal delay d_i of a module M_i, and the maximum allowable signal skew C_i at module M_i for all the modules.