The term logic synthesis refers to all activities in the design of digital hardware which are related to the implementation of switching functions and sequential machines. Its main aim is the transformation of a (symbolic) behavioral description of a digital system module into a network of (binary) logic building blocks, in such a way that this network realizes the module's behavior, satisfies specified constraints, and optimizes stated objectives. In a broad sense, it includes specification of sequential machines and switching functions, transformations between various specifications and representations (e.g. Mealy/Moore conversion, truth table/BDD conversion), transformations of behavioral specifications into implementations being some networks of logic building blocks (e.g. two- and multi-level synthesis, structural decompositions), transformations of logic networks (e.g. retiming, technology mapping, FPGA/PLD fitting), test pattern generation, simulation and formal correctness verification for switching functions, sequential machines and logic networks and so forth.

Logic synthesis is located on the trajectory of digital system design between the subsystem (register transfer level) design and layout synthesis. Layout synthesis interfaces it with the actual physical design.

The theoretical roots of logic synthesis can be found in the works of Georges Boole which date from around 1850, but just a century later Boolean algebra was applied for the first time in the design practice in the form of the switching network theory for relay networks. As an engineering discipline, logic synthesis started perhaps in seventeenth century, with the development of the first calculation machines by Pascal and Leibnitz and the first mechanical digital controllers for controlling looms by Falcon, around 1720. A real revolution in logic design, however, came with the usage of semiconductor technology for implementing switching functions and sequential machines as integrated circuits. During the 1960's, it became clear that circuits of the required complexity could not be designed manually, and subsequently the first logic synthesis tools emerged.
Modern microelectronic technology provides opportunities to build digital circuits of huge complexity and provides a large variety of logic building blocks. A rapidly growing interest in programmable devices has also been observed, as a result of their very attractive features. Although logic designers have been building circuits for years, they have realized that advances in microelectronic technology are outstripping their abilities to make effective and efficient use of the created opportunities. This has created a strong stimulus for further development of digital circuit theory, logic design methods, and logic synthesis tools. The three papers in this session present some new results in methodology and tools for logic and layout synthesis.

The first paper, *Results Given by a New Evaluation System for Placement and Routing Heuristics*, by R. Rauscher, D. Klawan, and H-J. Bandelt, addresses the problem of layout synthesis, which forms an interface between the actual logic synthesis and the physical design. The goal of the project reported in this paper was to develop and implement a flexible software tool (ESPRO) for easy implementation, evaluation, and comparison of various placement and routing heuristics. The authors used the open system concept. Their tool consists of a set of stand-alone programs that communicate through standard ESPRO files and can be used and combined in an interactive mode.

The second paper, *A System for Heuristic Modifications on PLA-Specifications*, by R. Rauscher and A. Krause, is an extension and refinement of the paper entitled *A System for PLA Minimization Based on Slight Behaviour Modifications*, by R. Rauscher, from EUROMICRO '95. It shows that simultaneous design of functional specifications and their logic level implementations extends the search of the design space, and leads to more efficient solutions. It presents a system for discretization of continuous functional specifications by scaling, rounding to integers and binary coding for a class of problems that involve PLAs as an implementation technique.

The final paper, *Retiming for circuits with Enable Registers*, by H-G. Martin, considers the problem of performance-driven optimization of sequential logic networks by a certain type of local transformations known as retiming. Known retiming techniques assume one clock cycle between registers, but enable registers (contained for example by circuits that come from high level synthesis) introduce also paths with more than one clock cycle. Taking it into account during retiming leads to better results. The presented retiming algorithm also performs a register relocation for circuits containing both the enable registers and D flip-flops.