A BALANCED MULTILEVEL DECOMPOSITION METHOD

Henry SELVARAJ, Tadeusz ŁUBA,

Warsaw University of Technology, Institute of Telecommunications,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract

A general decomposition concept is presented in this paper. The main strategy behind the presented Multilevel Decomposition Method (MDM) is striking a balance between serial decomposition and parallel decomposition. Such a strategy, on one hand, is capable of detecting and utilizing the fact that a group of outputs may depend on the same set of input variables and on the other hand eliminates the redundant variables of different outputs. The method is applicable to a variety of Field Programmable Gate Arrays. The results prove that the method is efficient and does not suffer from its generality.

MDM Method

A general and complete decomposition method has been developed for Boolean functions specified using minterms and cubes. The method is called general because it is applicable to a variety of Field Programmable Gate Arrays. In fact, the method is applicable to any device with fixed number of inputs and outputs. The method is called complete as it is capable of producing a DAG for any given function, where a node represents a Logic Block with user defined number of inputs and outputs.

Based on the general decomposition algorithm, a logic synthesis system called **DEMAIN** has been developed for both HP-UX and MS-DOS operating systems.

Its superiority, with respect to the number of Logic Blocks, over both commercial and academic tools has been proved through experiments on MCNC benchmarks. An example design using ALTERA's FPGA was also performed. For most experiments with real designs, the decomposition process has taken a few minutes on a HP Workstation. In this system, the functional decomposition is always carried out at the very beginning of the design process, when the existing don't care conditions can be effectively exploited to minimize the complexity of the resulting components. The developed results demonstrate that logic synthesis based on functional decomposition is usually much more efficient than the conventional approach in which technology-independent minimization is followed by technology mapping.

The system is very general and the user can arbitrarily specify the maximum number of inputs and maximum number of outputs for all components of a function to be decomposed. Alternatively, the designer can interactively control the decomposition process by selecting, for each iteration of the decomposition, the component of the partially decomposed function to be dealt with and the type of decomposition (parallel or serial). The maximum acceptable number of inputs, outputs for the resulting subfunctions can also be specified. This way, the decomposition procedure allows the designer to examine several alternative solutions, and select the one which is most suitable for a given project. The system can therefore form a basis for the development of a general decomposition-based synthesis tool which would accept a set of design constraints and decompose a given function so that to meet those constraints.

Though the presented Multilevel Decomposition Method is applied only for binary valued Boolean functions in this work, with some modifications it is possible to adapt the same for multiple-valued functions also [2]. Moreover, further research can be directed towards a more effective coding process. At present, the process simply tries to retain and introduce as many "don't cares" as possible without giving any consideration to the consequences on further decomposition. In conjunction with a more appropriate evaluation of the variables, the coding process can possibly give a direction for further decomposition.

References

[1] T.Łuba, M.Markowski, B.Zbierzchowski, "Logic Decomposition for Programmable Gate Arrays", Proc. Euro-ASIC'92, pp.19-24, 1992.

[2] T.Łuba et al., "An Implementation of Decomposition Algorithm and its Application in Information Systems Analysis and Logic Synthesis". In W.Ziarko (Ed.) Rough Sets, Fuzzy Sets and Knowledge Discovery. Workshops in Computing Series. Springer-Verlag London 1994.

[3] H.Selvaraj, "FPGA-based Logic Synthesis". Technical Report. Warsaw University of Technology, 1994.