Abstract

The design of electronic circuits, especially of VLSI components, heavily relies on extensive computer support. CAD tools for this purpose incorporate high algorithmical power. However, they lack integration and reliable common data interfaces. Recent research efforts have shown that a central database is an indispensable basis for any CAD system.

As existing database management systems (DBMS) do not prove suitable for the CAD/CAM area, specially tailored systems have to be developed. The DAMASCUS system presented in this paper is an approach to DBMS support for VLSI design. We discuss its central concepts, including a multilevel architecture, consistency control, and transaction management. Special emphasis is placed upon its data modelling facilities.

1. Introduction

In order to be economically and intellectually feasible, the design of electronic circuits, especially VLSI components, requires extensive computer support. In fact, a variety of tools and systems for various design phases have been developed that improve the quality and efficiency of the design process by incorporating high algorithmic power. Though the development of tools will remain a central point of research, another aspect has recently emerged, namely the demand for the integration of tools by a common data management and by standardized interfaces. Up to now most CAD tools work isolated from each other, with every tool using or generating its individual files and requiring special data formats. Transformation programs have to be used to enable communication between the tools.

Apart from data redundancy with all its drawbacks, the file system approach lacks a lot of capabilities that are today common for data management in business and administration types of applications. While these facilities are very desirable for the CAD/CAM environment, too, existing database management systems (DBMS) are rather inadequate for this area.

This paper points out the special data management requirements of CAD for VLSI. Afterwards we explain the shortcomings of commercially available systems with respect to these requirements. The principal solutions to overcome these shortcomings will be presented. We argue that the only long-term solution is the development of new, specialized database management components for design systems.

The DAMASCUS system, a prototype of which is currently developed at FZI Karlsruhe, is presented as an approach to a DBMS specifically tailored for VLSI design. The central concepts including the system architecture and the data modelling facilities are presented. Furthermore solutions for consistency checking, transaction management as well as some performance issues are sketched.
2. Advantages of DBMS support for CAD systems

The main advantage offered by a DBMS in comparison to the file system approach is the integration of data and tools [Dat86]. In most of today's CAD systems each tool reads and writes its data using separate files and individual formats (fig. 1). Transformation programs are required to mediate between the different tools. When existing tools are changed or new ones are created, the set of transformation programs has to be modified or extended. Furthermore, the separate storage of data for each tool introduces redundancy and the problem of inconsistencies between the different copies. The integrated management of all data together with a unified interface (fig. 2) is therefore clearly desirable.

Fig. 1: Data management in CAD systems (today)

Apart from the integration aspect, the database approach offers a number of properties that are useful for CAD data management:

- **Homogeneous standardized interface**: All users and application programs are provided with a stable common data interface. Thus tool construction becomes easier and more comfortable. Even if a program still requires its individual format, the DBMS view mechanism supports the necessary adaptation.

- **Data independence**: Database systems offer independence on the physical as well as the logical level. Physically an application may ignore the storage structures and storage media actually used. If for example the storage structures are changed to improve access performance, the application programs need not be modified. On the other hand, information about the logical structure of the data is kept in the DBMS, while each application program may use its individual view of the data. Thus structures for new purposes may be added without modifying any of the old applications.