DDB: An Object Oriented Design Data Manager for VLSI CAD

Anoop Singhal, Robert M. Arlein and Chi-Yuan Lo

AT&T Bell Laboratories
Murray Hill, NJ 07974

Abstract

In this paper we present an object oriented data model for VLSI/CAD data. A design data manager (DDB) based on such a model has been implemented under the UNIX/C++ environment. It has been used by a set of diverse VLSI/CAD applications of our organization. Benchmarks have shown it to perform better as compared to commercial object oriented database systems. In conjunction with the ease of data access, the data manger served to improve software productivity and a modular program architecture for our CAD system.

1. Introduction

A design data manager is emerging as an important component of an integrated VLSI CAD system. Our earlier papers [MeSi87] [SiPa89] presented data models for design versions, design configurations and circuit description data. In this paper we present our experience in applying an object oriented paradigm to design and implement a data manager (DDB) for VLSI/CAD. Section 2 discusses an object oriented data model for VLSI design data. Section 3 discusses the system architecture and finally section 4 provides conclusions.

2. An Object Oriented Data Model for Design Data

Our data model is based on the work of EDIF [IMG92] and CFI [CFI93]. Designs are hierarchical i.e. a design is built from other designs recursively until a subdesign is reached which is composed of primitive elements. The entity *cell* is used to represent a unit of hierarchical design. A cell may have different *views* such as behaviour, netlist or layout. The inclusion of smaller designs of which a view is composed of is represented by the *instance* entity. Therefore, an instance can be regarded as a relationship between two views: the view using (owning) the instance and the view describing the instance. For example, in the netlist view of half adder an instance of the netlist view of XOR represents the fact that the xor gate is used as a component in the design of half adder.

Having provided a representation of the hierarchical structure of designs we discuss some more entities that are required to represent the connectivity of instances within a view. The connectors of a design are represented by the *port* entity. A port has a name and a direction of signal flow which is either input or output or bidirectional. The pins of an instance are

SIGMOD /5/93/Washington, DC,USA

represented by the entity *portinstance*. The portinstance belonging to an instance corresponds to the port in the view that is referenced by the instance object. The concept of a *net* entity is used to represent the set of connections among portinstances and ports of a view. Finally, the entity *library* is used to model the concept of a cell library. This model is described in EXPRESS [EXP91] as follows.

```
ENTITY lib;
 name   : name_def;
 cells  : SET OF cell;
END ENTITY;

ENTITY view;
 name            : name_def;
 ports           : SET OF port;
 nets            : SET OF net;
 instances       : SET OF instance;
END_ENTITY;

ENTITY net;
 name            : name_def;
 joined_ports    : SET OF port;
 joined_pinsts   : SET OF pinst;
END_ENTITY;

ENTITY pinst;
 describer       : port;
 owner           : instance;
UNIQUE
 describer, owner;
WHERE
 r1: describer IN owner.describer.ports
END_ENTITY;

ENTITY cell;
 name   : name_def;
 views  : SET of view;
END_ENTITY;

ENTITY port;
 name   : name_def;
 dir    : input, output etc.;
END_ENTITY;

ENTITY instance;
 name     : name_def;
 describer : view;
 owner    : view;
 pinsts   : SET OF pinst;
END_ENTITY;
```

Figure 1: The Data Model in EXPRESS

This model can be translated into a set of C++ class definitions. Basically, each entity in the EXPRESS model becomes a C++ class definition.

3. *System Architecture*

Figure 2 shows an architecture diagram of the system. The Hierarchical Search Manager (HS) [PaSi90] provides a set of functions for CAD applications to efficiently search the design data space. The Database Language (DBLANG) [ArVa92] provides a programmable extension language for end users to efficiently query/modify the design data.

4. *Conclusion*

This paper presented an object oriented data model for VLSI CAD data. This data model was used to implement a design data manager (DDB) in the UNIX/C++ environment. We believe that this system will serve to improve software productivity and a modular architecture for our CAD system. The work described in this paper will be extended in several directions. We plan to use a remote procedure call (RPC) mechanism for applications to communicate with the data manager from remote machines using a client/server paradigm. Mechanisms for change notification, data consistency and incremental changes will also be implemented.

REFERENCES

[MeSi87] Mehmood Z., Singhal A., et al., "A Design Data Management System for CAD", Proc. of ICCAD Conference, Nov. 1987, pp 220-223.

[SiPa89] Singhal A., Parikh N., Dutt D., Lo C.Y., "A Data Model and Architecture for VLSI/CAD databases", Proc. of ICCAD Conference, Nov. 1989, pp 276 - 279.

[PaSi90] Parikh N., Lo C.Y., Singhal A., and Wu K.W. "HS: A Hierarchical Search Package for CAD data", Proc. of ICCAD Conference, Nov. 1990, pp 478 - 481. Also in IEEE Transactions on CAD, Jan. '93, pp 1-5.

[CFI93] "A Design Representation Information Model and Programming Interface Release 1.0", CAD Framework Initiative, January, 1993.

[IMG92] "EDIF Information Model, version 0.3-6", EDIF European TSC Information Modelling Working Group, Feb. '92.

[EXP91] "Express Language Reference Manual", ISO TC184/SC4/WG5 Document 118, Document N14, March 1991.

[ArVa92] "Coping with EDIF Dialects using DBLANG", EDIF WORLD '92. pp 85 - 94.

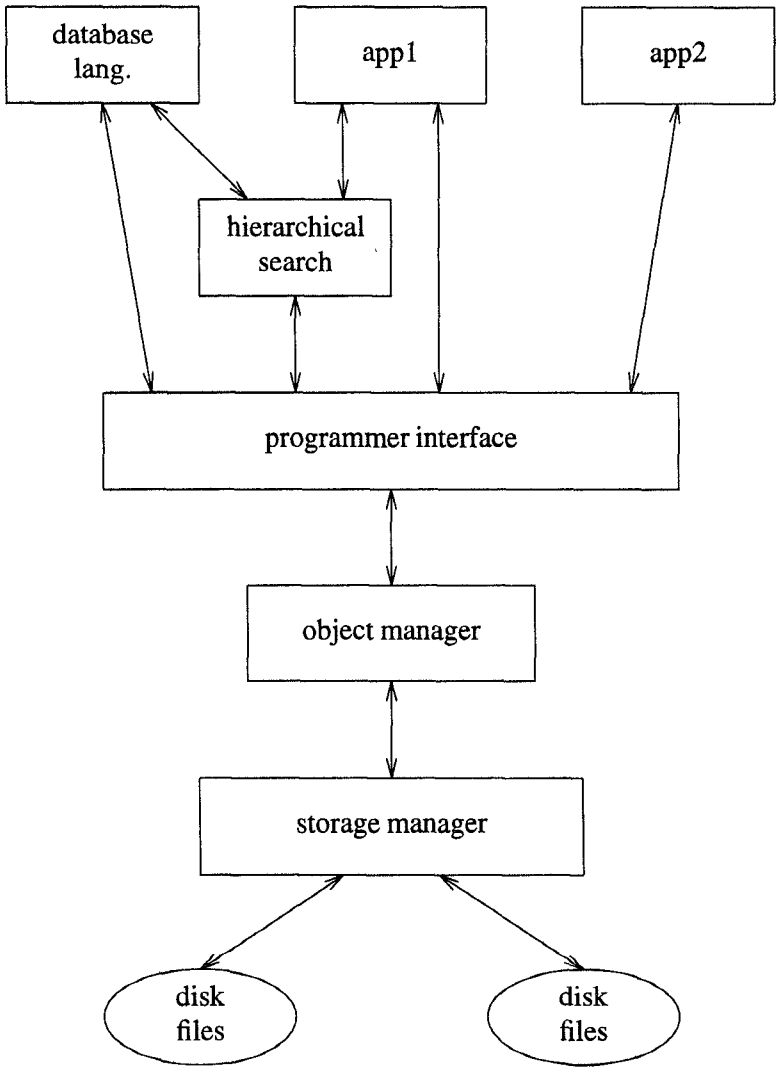

Figure 2 "System Architecture"