USING A RELATIONAL DATABASE MANAGEMENT SYSTEM FOR COMPUTER AIDED DESIGN DATA

by

Antonin Guttman and Michael Stonebraker

Memorandum No. UCB/ERL M82/37

25 March 1982

ELECTRONICS RESEARCH LABORATORY

This work was supported by the National Science Foundation under grant ECS 8007683 and by the Air Force Office of Scientific Research under grant 78-3596.

USING A RELATIONAL DATABASE MANAGEMENT SYSTEM FOR COMPUTER AIDED DESIGN DATA

Antonin Guttman
Michael Stonebraker

University of California
Berkeley

1. Introduction

There has been considerable interest in using relational database systems to manage data for computer aided design (CAD) systems. However, concerns have been expressed that database systems generally have been designed for other uses and might not be well suited to CAD. In addition, there is concern that relational DBMS's are too slow. The purpose of this paper is to report our experience with implementing a CAD application in a relational DBMS and comparing its performance with a non-database CAD package.

In Section 2 we discuss the database schema that was used and describe the structure of the special purpose CAD package. Then in Section 3 we indicate the experiments performed and give the results. Section 4 contains a discussion of the results. Lastly, Section 5 indicates areas where a relational DBMS was found to be inadequate as a support tool for CAD applications.

2. The Database Schema.

The application package benchmarked was KIC, a graphics editor for integrated circuit designs developed at Berkeley [2]. A KIC database consists of a collection of circuit "cells". Each cell can contain mask geometry and subcell references. A complete circuit design expands into a tree, with a single cell at the root and other cells, used as subcells, for the non-root nodes. Mask geometry can be associated with each node in the tree. During editing KIC stores a circuit in virtual memory on a VAX 11/780 computer.

Our INGRES schema reflects the above structure and has five main relations:

cell master (name, author, master id, defined)

cell ref (parent, child, cell ref id, t11-t32)

box (owner, use, x1, x2, y1, y2)

wire (owner, use, wire id, width, x1, y1, x2, y2) polygon (owner, use, polygon id, vertnum, x, y)

In the cell master relation, name is the textual name given to the cell and author is the name of the person who designed it. Master id is a unique identification number assigned to each cell. It is used for unambiguous references to the cell within the database.

The cell ref relation describes subcell references. For example, if the cell "cpu" contains "register" as a part, then the cell ref relation contains a tuple in which parent is the identifier of "cpu" and child is the identifier of "register". T11 through t32 are a 3 X 2 matrix specifying the location, orientation and scale of the subcell with respect to the parent. This representation of a spatial transform is the one generally used in computer graphics [3].

The box relation describes mask rectangles. Owner is the identifier of the cell of which the box is a part. Use specifies the mask layer; e.g. metal or diffusion. X1 and x2 are the x-coordinates of the left and right sides of the box while y1 and y2 are the y-coordinates of the top and bottom.

A "wire" is a connected path of lines. Each tuple in the wire relation describes one line segment, giving the coordinates of its centerline (x1, y1, x2, y2) and its width. Wire id is a unique identifier for a particular wire. Owner and use mean the same as for the box relation.

A "polygon" is a solid shape with any number of vertices. One vertex is stored in each tuple of the polygon relation. X and y are the coordinates of the vertex, and vertnum orders the vertices (tuples) within one polygon.

3. The Experiment

For our test data we used circuit cells from two design projects at Berkeley. GORDA is a prototype layout for a switched capacitor filter [5], and dec0-1 is a part of the RISC VLSI computer [6,7]. The design descriptions were translated from KIC files and loaded into the INGRES database.

We programmed three representative CAD retrieval operations and measured the speed of our test programs compared to KIC performing the same operations. The test programs were written in C and made calls to the INGRES DBMS [4].

Top-level geometry retrieval. The first program retrieved the geometry data associated with a given circuit cell, not including geometry belonging to subcells. This is the first step in most CAD editing sessions.

Geometry retrieval with tree expansion. The second program retrieved geometry for all cells referenced in a fully expanded design tree. Geometry for lower level cells was transformed to its proper position relative to the root cell. This operation produces a plot of an entire design.

Retrieval by location. The third program retrieved top-level geometry that fell within a small area in the middle of a cell. This operation would be used to "window" a circuit.

The tables below show the results of the tests. Geometry Tuples refers to the number of tuples representing geometry retrieved from the INGRES database during each operation. CPU Time and Elapsed Time were reported by the operating system. The msec/Tuple figures are time divided by the number of INGRES tuples. The Relative Time rows give the slowdown factor of our test programs relative to KIC.

Circuit	GORDA	
Geometry Tuples	592	
	KIC	INGRES
CPU Seconds	7.5	24.0
CPU msec/Tuple	13	41
Relative CPU Time	1	3.2
Elapsed Seconds	7.5	41
Elapsed msec/Tuple	13	69
Relative Elapsed Time	1	5.5

Test 1: Top Level Retrieval

Circuit	GORDA	
Geometry Tuples	12,779	
	KIC	INGRES
CPU Seconds	128.3	443.5
CPU msec/Tuple	10	35
Relative CPU Time	1	3.5
Elapsed Seconds	128.3	649
Elapsed msec/Tuple	10	51
Relative Elapsed Time	1	5.1

Test 2: Retrieval with Tree Expansion

Circuit	dec0-1	
Geometry Tuples	448	
	KIC	INGRES
Geometries in Window	87	85
CPU Seconds	.75	14.5
CPU msec/Tuple	9	171
Relative CPU Time	1	20
Elapsed Seconds	.75	33
Elapsed msec/Tuple	9	388
Relative Elapsed Time	1	45

Test 3: Retrieval by Location

4. Discussion of Results

In Tests 1 and 2, the factor of three increase in CPU time for INGRES can be attributed primarily to the fact that INGRES is a general-purpose DBMS while KIC includes only functions that it requires. Any DBMS used in place of special purpose code will show some decrease in performance. This cost must be balanced against the advantages of using a DBMS, e.g. simplification of application software, data independence, etc.

The elapsed time measurements show a larger performance difference, about a factor of five. This is mainly because INGRES geometry data is disk resident. KIC geometry data resides in virtual memory, and since the tests were run on a dedicated machine with ample main memory, KIC's data was actually in real memory.

KIC has a spatial bin structure that allows it to quickly isolate geometry in the window for Test 3. INGRES has no such access method and must perform a sequential search.

The new features suggested in the next section may narrow the performance gap. In addition, changing INGRES to manage a virtual memory database would dramatically improve performance.

5. New Features

During our experimentation we identified four features that should be added to a relational DBMS to facilitate CAD applications. We discuss them in turn.

5.1. Ragged relations

It would have been convenient for a field in a relation to repeat a variable number of times. In our database, for example, this would have allowed us to store data for a varying number of "boxes" in the cell master relation, instead of in a separate box relation. The cell master relation could have been defined by

```
    cell master (master id, name, author,
                 defined, &(use, x1, x2, y1, y2)).
```

where the notation "&(...)" means that the group of five fields describing a box is repeated, once for each box. The wire, polygon and cell reference relations could be coalesced with the cell master relation in a similar way.

This revised schema might be more natural for a user to understand. In addition, it would speed access to the geometry for a particular cell, since the geometry would be in one tuple rather than distributed across three relations.

We propose this extension under the name "ragged relations". One way to support ragged relations would be to first provide ordered relations, which has been suggested by Stonebraker and others [8,11], and then allow relations to nest, so that a field of a relation could be an ordered relation. We are investigating this idea.

A database with ragged relations or nested relations is not in first normal form, and reflects the hierarchical nature of the data. A language that provides just the standard relational data manipulation operations must be augmented before it can be used with ragged relations or nested relations. We are investigating such an augmentation.

5.2. Transitive Closure

Our second test program was required to expand subcells which could nest a variable number of times. This is an operation similar to a transitive closure and does not correspond to a single INGRES query. We have extended the syntax of QUEL with a * operator to facilitate such tasks. For example:

```
    range of r is cell ref
    range of t is tree
    retrieve * into tree (cell = r.child)
               where r.parent = ROOT
               or r.parent = t.cell
```

Here the tuple variable t ranges over the tree relation, which is the result of the query. The retrieve adds tuples to tree, and is repeated (with t ranging over the new tuples) until there are no new additions.

Some database operations involving transitive closure are possible in Query-By-Example [12] and in the ORACLE system [13].

5.3. Access by spatial location

Many CAD programs, like our third test program, need to retrieve design data according to its spatial location. This test would have run much faster in INGRES if data could be classified according to a system of spatial "bins", i.e. by approximate location.

We are considering adding a spatial bin mechanism to INGRES as an extension of the secondary index facility. A two-dimensional array of bins is defined by a grid. A bin index is a file containing, for each bin, pointers to all the objects that might overlap that bin. Geometries in a particular area can be found quickly by looking in the index under the appropriate bins. For example, a bin index could be defined by:

```
    index on box is boxbins
            (min(x1,x2) through max(x1,x2) by 10,
             min(y1,y2) through max(y1,y2) by 10).
```

This bin index is based on a grid of 10 X 10 squares. The min-max expressions specify the size of each box and define the set of bins it could overlap.

5.4. Unique identifiers

Codd and others [9,10] have suggested the usefulness of unique identifiers for database objects. They should be generated automatically by the database system and handled internally as a special type. In our application we were forced to manage our own unique identifiers for cells, wires, etc.

6. Conclusions

INGRES performs typical computer aided design retrieval operations considerably slower than special purpose programs. We have suggested a number of enhancements that could be made to a relational database system that would improve performance and make the system easier to use for CAD applications. We are continuing to look for additional mechanisms along the same lines.

7. Acknowledgements

This work was sponsored by NSF grant ECS-8007683-Wong-2/82 and by AFOSR grant 78-3596-Stonebraker/Wong-6/82.

8. References

[1] Held, G.D., M.R. Stonebraker and E.Wong "INGRES: A Relational Data Base System," Proc. AFIPS 1975 NCC, Vol. 44, AFIPS Press, Montvale, N.J.

[2] Keller, K. Kic, A Graphics Editor for Integrated Circuits, Masters Thesis, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, CA. June 1981.

[3] Newman, W.M. and R.F. Sproull "Principles of Interactive Computer Graphics," McGraw-Hill, N.Y. 1979.

[4] Woodfill, J. et al. "INGRES Version 6.2 Reference Manual," Memo No. UCB/ERL M79/43, Electronics Research Laboratory, University of California, Berkeley, CA. July 1979.

[5] The circuit cell GORDA is a layout prototype of a switched capacitor filter made by Jesus Guinea at Electronics Research Labs, University of California, Berkeley, CA.

[6] Patterson, D.A. and C.H. Sequin "RISC I: A Reduced Instruction Set VLSI Computer," Proc. Eighth International Symposium on Computer

Architecture, May 1981, pp. 443-457.

[7] Fitzpatrick, D.T. et al. "A RISCy Approach to VLSI," VLSI Design, Fourth Quarter (October 1981), pp. 14-20. (Also appears in Computer Architecture News, March 1982.)

[8] Stonebraker, M. and J. Kalash "TIMBER: A Sophisticated Relation Brouser," Memo No. UCB/ERL M81/94, Electronics Research Laboratory, University of California, Berkeley, CA. December 1981.

[9] Codd, E.F. "Extending the Database Relational Model to Capture More Meaning," ACM Transactions on Database Systems, Vol. 4, No. 4, December 1979, pp. 397-434.

[10] Lorie, R.A. "Issues in Database for Design Applications," IBM Research Report RJ3176 (38928) 7/10/81.

[11] Lorie, R.A., R. Casajuana, and J.L. Becerril "GSYSR: A Relational Database Interface for Graphics," IBM Research Report RJ2511 (32941) 4/24/79.

[12] Zloof, M. "Query-By-Example: Operations on the Transitive Closure," IBM Research Report RC 5526 (Revised) (#24020) 10/29/76

[13] "ORACLE SQL Language - Reference Guide," Copyright by Relational Software Incorporated, October 1980.