BUBBLES FOR RELATIONAL DATABASE

Hsu Chang

IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT: The mechanical disk storage is limited by long initial delay, few inputs/outputs, and serial access, thus necessitating large complex programs in existing database systems in order to map the user's view into the physical storage and to provide different access paths in response to different queries. Bubble devices are capable of set-oriented processing and associative addressing architecture. This paper presents concepts on data structure, storage structure, and access methods for bubble implementation of relational database.

INTRODUCTION

Mechanical disks have provided the first and so far the only economical large files with acceptable access speed for database applications. However, the disks have long initial delays (tens of ms) associated with the mechanical movements of heads from track to track, and along a track. The read/write heads and the mechanisms regulating their motions are complex and expensive, hence are supplied sparingly. The so-called direct access refers to track selection; then the bits must be accessed serially along a very long track. Finally, disks are storage devices only. Within the storage media, there is no data manipulation capability, however simple.

As a consequence(1) large batches are collected (10^4 bytes) per I/O request, thus restricting search speed and flexibility. Only 1 percent of the huge data file can be readily addressed at any time. The data file must rely on sequential organization, which results in extreme performance variation for various query modes.

Database operations such as search, retrieval, statistical computation, and update are inherently associative and set-oriented. Since these capabilities are lacking in disks, complex programs have been constructed to map the user's view as expressed in the above operations into the physical reality of the disk file, and also to provide different access paths to allow different queries to locate data fast.

RELATIONAL DATABASE

Both in terms of database system concepts and the hardwares to facilitate their implementation, there have been significant advances in recent years.

A database can be considered as a structured collection of data which is accessible by concurrent users through computer systems. A relational model of database views data in tabular form, with a set of similar entities arranged in rows (tuples), and their many attributes aligned in columns (domains). A relational database is a collection of such tables. The items in the table may vary with time because of modifications, insertions, and deletions. Data in two or more relations (i.e., tables) can be interrelated through compatible attributes which appear in each of the relations. This allows the user to execute queries which have complex selection criteria. There is extensive and excellent literature on relational database (e.g., see Martin's book(2)), since Codd(3) advanced the concept in 1970.

The main pursuit of this paper is based on the observation that there appears natural correspondence between the tabular form of relational data model and the array-like solid-state storages such as bubbles, as well as between the set-oriented query operations and the group-coordinated bubble manipulations.

It should be mentioned that there have been several proposals, CASSM (4) RAP (5), RARES (6) and STARAN (7), which couple microprocessors to rotating memories for relational database implementation. The present paper attempts to advance the art by implementing search, retrieval, and reordering of data on chip, thus achieving a higher degree of integration in hardware and operation, as well as moving the implementation closer to the high level view implied by the relational model.

A RELATIONAL BUBBLE CHIP

A bubble chip is configured to facilitate relational data base operations. The design is aimed at data selection and rearrangement on the chip, so as to minimize data traffic through the limited number of I/O ports and to facilitate data access in response to different user views. Refer to Fig. 1. The components on the chip are described in this section and the relational database operations are described in the next.

Storage -- Data are stored in many nonvolatile, nondestructive-readout shift registers which are capable of start and stop of data movement, and data reversal; as well as read, write, and text-editing functions(8). The shift registers are arranged in parallel, and each is segmentized into cross-linked loops(9). The rows of registers correspond to tuples, and columns of loops correspond to domains in a relational database.

Data Rearrangement -- The cross-linked loops are capable of four modes of linkage: global shift, detached shift, exchange, and delta exchange. By proper sequences of these operations, various data rearrangements(9,10,11) can be achieved, which include topping, FIFO, FILO, sorting, etc. In the present chip design, the loops will mostly be used to rearrange columns to facilitate associative search and access.

Associative Search -- Associative search (content or context addressing) identifies simultaneously all entries which contain a given attribute or a Boolean combination of attributes. Only the identified entries are accessed through the limited number of output ports. Speed is improved both by simultaneous search and by limiting the I/O facilities for the traffic of pertinent data.

The bubble associative devices(12,13) have three distinct advantages. First, since the shift register circulates its entire content to the associative search station, there is minimum per-bit overhead resulting from the associative search logic. By contrast, when random-access memory cell is provided with associative search capability on a per cell basis, each cell and consequently the entire chip size are made four or five times larger. Second, since the search device has memory capability, one cell suffices for any length of search attribute. Third, the search attribute is conveyed by magnetic field gradient associated with a conductor current. Hence by varying pulse pattern, Boolean functions such as OR, NEGATION, and NOR can be executed. In addition, the use of two consecutive search cells allows executions such as AND, NAND, A ≤ ≤ B, etc.

Accesses -- Any of the shift registers can be accessed randomly via the decoder (14) if both the associative search gate remains open and the proper decoder control lines are activated. The random access of a shift register allows the write-in, read-out, insertion, deletion and rearrangement of data for the shift register in the buffer loops and the decoder without affecting the other shift registers on the chip.

When the selected domains are already propagated through the associative-search gates into the buffer loops, the loops can be converted from parallel to serial connection (15) so that the selected data in each column of buffer loops can be accessed via one I/O port.

Indicator -- The occupancy-indicator shift register is at one end of and orthogonal to all the storage shift registers. Each shift register, when written-in, contains an occupancy bubble, which can be replicated into the indicator shift register. The indicator register, being a much shorter register, and without having to go through the buffer register, can quickly indicate the occupancy status of all the shift registers, and where new data entries can be entered.

CHIP OPERATIONS

Before we proceed to discuss the data base operations possible on bubble chips, let us mention the concept of relational completeness. Codd (16) has formulated a theoretical basis to determine how complete a selection capability is provided by any data sublangugage, which is intended to interrogate and update a data base. He has shown that the basic capabilities required consist of only the traditional set operations (Cartesian product, union, intersection, difference) and relational operations (projection, join, restriction). In the following, we shall illustrate how the above operations are performed on the bubble chips for relational data base.

To perform any of the four set operations, there must be two relations (or tables) on two separate chips, and the results are assembled on a third chip. These operations are performed on one or more common or compatible attributes of the original tables. However, the new table may obtain other attributes from the original tables. Essentially all data base operations are performed on the bubble chips. Semiconductor devices are only needed as drivers for write, read, and associative search.

To perform the Cartesian product operation, the items of a selected column are read out row by row from the first table (or chip). After each item has been read out, the first chip is arrested; and from the second chip, an entire column is read out row by row and each matched with the item from the first chip. Then the matched pairs are entered into a new table. The process is repeated until the column of the first chip is completely read out. The Cartesian product of two sets C, D is denoted C x D, and is defined by

$$C \times D = \ (c,d) : c \varepsilon C \wedge d \varepsilon D$$

To perform the union operation, two compatible relations (or tables) from two separate chips are consecutively read out and written into a third chip to form a new table. The union of two sets C and D is denoted by $C \cup D$, and is defined by

$$C \cup D = \ c : c \varepsilon C \vee c \varepsilon D$$

To perform the intersection operation, the items of a selected column are read out row by row from the first table. While each item is being read out, the bubble signal is transformed into electrical signal, which is used in turn to perform associative search on a selected column in the second chip. If there is a match, the item is written into the third table. The above process is repeated when all rows of the first chip have been read out and matches attempted with the second chip. The intersection of two sets C and D is denoted by $C \cap D$, and is defined by

$$C \cap D = \{c : c \varepsilon C \wedge c \varepsilon D\}$$

To perform the difference operation, the procedure is similar to that of the intersection operation. The only difference is in that only at mismatch rather than match, the item is used in the third (new) table. The difference of two sets C and D is denoted by C -D, and is defined by

$$C - D = \{c : c \varepsilon C \wedge c \notin D\}$$

We shall now proceed to describe the relational operations (projection, join, restriction). The projection operation manipulates a relation by dropping columns from a relation and then eliminating the resulting redundancy in the tuples (rows). The operation is executable by the following steps: (A) Move the undesired column(s) to the far left, leaving the desired columns adjacent to the access circuitry. (B) Read out all the tuples successively via the decoder. The read-out tuple is entered into a new table. (C) After each readout, the just read tuple can be transformed from a bubble signal into an electrical signal, which is then used to perform an associative search on the original table. All the matched shift registers will be sealed off after the search. (D) The above process is repeated until the entire original table has been read out. Note that the sealed-off tuples are never read out.

Assume an m-ary relation R contains n tuples, r (i.e., a table of m columns and n rows). Let A = (j1, j2,..., jk) , a subset, select the columns from the set (1, 2,...m), then the projection of R on A is defined by

$$R[A] = \{ r[A] : r \varepsilon R \}$$

The join operation forms a new relation out of two separate relations, which does not use all the attributes of these relations. The join criterion can be any of the relations $=, \neq, <, \leq, >$ and $\geq$, which is applied to two compatible attributes from the two original relations. The restriction operation is a special case of join when the two original relations are identical.

The operation is executable by the combination of associative search (content of context addressing) and Cartesian product, and consists of the following steps: (A) In each original table move the compatible columns close to the associative search devices, followed by other columns to be used in the new table. (B) Read out from the compatible column of one table, via the decoder, one row at a time from the top. The row just read out is temporarily stored in the third chip. (C) While reading out, the relevant attribute, in its electrical signal form, is used in the second chip to perform an associative search. (D) If the search yields tuples in the second table, the tuples are read out and written into the third table to be joined with the temporarily stored tuple from the first table by Cartesian product. The resulting tuples are stored permanently. However, if the search yields no tuple in the second table, the temporarily stored tuple in the third table is erased. (E) Steps B through D are repeated until the first table has been read out completely.

Let θ denote any of the relations $=, \neq, <, \leq, >$ and $\geq$. The θ-join of relation R on domain A with relation S on domain B is defined by

$$R[A\theta B]S \doteq \{(\overset{\frown}{rs}) : r \in R \wedge s \in S \wedge (r[A]\theta\ S[B])\}$$

CONCLUSIONS

A. This paper serves as an existence-theorem proof that a bubble chip dedicated to relational data base operations can be designed. The selection capability to interrogate and update a data base has been demonstrated to be complete. The capability includes Cartesian product, union, intersection, difference, projection, join and restriction.

B. The hardware design of the chip has a remarkable resemblance to the relational data model. The data items are arranged in a two-dimensional array. The columns are interchangeable in position by the operation of the cross-linked loops, and the rows are equally accessible via the decoder. There are no built-in data linkages such as indices or pointers. On the contrary, during query, relevant data items are quickly identified by associative search and easily accessed via the decoder. While no user's view is imposed on the storage of data, all users can formulate queries and access the data with equal convenience and speed. The on-chip search- and identify- capability preempts the need to move irrelevant data through the input/output ports. The start-stop capability eliminates idle times between accesses.

C. While the term "data base" carries the connotation of a huge collection of data (e.g., greater than 10^{10} bits), the modular solid-state chips such as bubbles may well prove useful at more modest capacity. The rather small number of essential operators lend themselves to keyboard operations, and the simple device functioning needs very limited logic and programming. It is conceivable that a calculator-size query machine could be constructed. Of course, the concepts and hardwares discussed here are equally useful for large data base if bubbles indeed realize the high-density (hence low-cost) potential; and amenable to high-level language (e.g., Query-by-Example, (17)) implementation.

D. The key components in the bubble chip are the associative search devices. Up to the present, there have only been sporadic experimental studies of bubble-bubble interaction devices. Although conceptually feasible, such devices have not received the attention accorded to the conventional bubble devices (e.g., major-minor loops). It should also be noted that the hardware requirements and chip configuration for a relational data base, as delineated in this paper for bubbles, can by and large be applied to semiconductor devices which have always offered logic and switch capabilities.

ACKNOWLEDGMENT

The author would like to thank Moshe Zloof for introducing him to the subject of relational database, also Peter deJong and C.K. Wong for stimulating discussions.

REFERENCES

1. U.O. Gagliardi, "Effect of Electro-Mechanical Magnetic Memories on the Architecture of Data Management Systems," Proceedings of the Symposium on Advanced Memory Concepts (Ed. S.W. Miller and U.O. Gagliardi), Stanford Research Institute, pp. V-163 to V-191, June 1976.

2. J. Martin, "Computer Data-Base Organization" (book), Chaps. 13-14, Prentice Hall, (1975).

3. E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", Comm. ACM vol. 13, No. 6, 377-87, June 1970.

4. G.P. Copeland, G.J. Lipovski, and S.Y.W. Su, "The Architecture of CASSM: A Cellular System for Non-numeric Processing", Proc. 1st Annual Symp. on Computer Architecture, pp. 121-8, Dec. 1973.

5. E.A. Ozkarahan, S.A. Schuster, and K.C. Smith, "RAP-An Associative Processor for Data Base Management", Proc. AFIPS 1976 NCC, Vol. 44, pp. 379-88.

6. C.S. Lin, D.C.P. Smith, and J.M. Smith, "The Design of a Rotating Associative Memory for Relational Database Applications", ACM Trans. on Database Systems, Vol. 1, No. 1, pp. 53-65, March 1976.

7. P.B. Berra, "Some Problems in Associative Processor Applications to Database Management", Proc AFIPs 1974 NCC, Vol. 43, pp. 1-5, 1974.

8. S.Y. Lee, T.C. Chen, H. Chang, and C. Tung, "Text Editing with Magnetic Bubbles", Proc. Compcon Fall 1974, pp. 69-72.

9. C. Tung, T.C. Chen, and H. Chang, "Bubble Ladder for Information Processing", IEEE Trans. on Mag., Vol. 11, No. 5, pp. 1163-5, September 1975.

10. H. Chang and M.S. Cohen, "Joinable Magnetic Bubble Loops Using Current-Controlled Switches", IBM Tech, Disclosure Bulletin, Vol. 18, No. 11, pp. 3856-8, April 1976.

11. T.C. Chen, C. Tung, V.Y. Lum, and K.P. Eswaren, "Efficient Sorting Using Uniform Bubble Ladders", Internatl. Conf. on Magnetic Bubbles (Eindhoven, September 13-15, 1976).

12. S.Y. Lee and H. Chang, "Associative-Search Bubble Devices for Content-Addressable Memories", Compcon Fall 1975, Paper 8.2.

13. S.Y. Lee and H. Chang, "Associative-Search Bubble Devices for Content-Addressable Memory and Array Logic," to be issued as an IBM Research Report.

14. H. Chang, J. Fox, D. Lu, and L. L. Rosier, "A Self-Contained Magnetic Bubble Domain Memory Chip," IEEE Trans. on Magnetics, vol. MAG-8, no. 2, pp. 214-222, June 1972.

15. S. Y. Lee and H. Chang, "An All-Bubble Text-Editing System," IEEE Trans. on Magnetics, Vol. MAG-10, pp. 746-749, Sept. 1974.

16. E.F. Codd, "Relational Completeness of Data Base Sublanguages," in Courant Computer Science Symposia, Vol. 6: "Database Systems," edited by R. Rustin, Prentice Hall, 1972.

17. M.M. Zloof, "Query-by-Example: A Data Base Management Language, IBM Research Report, 1977.

POSTSCRIPT

This manuscript was prepared in January 1977. Between its preparation and its release for publication, advances have been made in several related areas. These advances do not directly affect either the substance or the conclusions of the manuscript to require updating or revision. However, a brief review of these related areas is presented in this postscript to provide a better perspective for the paper.

The relational data base chip was motivated by the desire to integrate not only data storage but also data manipulation and associative search on the same chip in order to speed up processing by parallel search and the elimination of all output except the final qualified data. While the advantages are obvious, the usual questions directed to a new approach can be raised here: Is it technically feasible? Are the functions realizable by existing devices? Is it economically feasible? These questions are illuminated, although not completely answered, by the advances in related areas.

Technical feasibility. The key element is the associative search device, which is described in conceptual details in Refs. 1 and 2. Its physical embodiment depends on the controllable steering of a moving bubble by a latched bubble. Such bubble-bubble interaction phenomena have been studied in related logic devices (Refs. 3-6). Our conviction is that when properly designed the bubble-bubble interaction devices could offer operating margins comparable to conventional discrete bubble (memory) devices. Many unconventional devices are surveyed in Refs. 7 and 8 and analytically treated in Ref. 8.

Realization in existing devices. Available devices include both logic per track (i.e. a fixed head disk file plus microprocessors) and major-minor loop array (a conventional bubble memory). A disk cannot avail itself of logic until the data is already ouput through the read head. A major/minor loop array does not have on-chip logic to perform associative search. However, data can be arranged on a major/minor loop chip to facilitate associative search off chip. Moreover, off-chip marker loops can be used to eliminate unnecessary traversing of data. Such an approach was discussed in Ref. 9 (a sketch) and Ref. 10 (detailed description of instruction set and query examples). It has the advantage of utilizing existing chips, but it is incapable of parallel on-chip search.

Ref. 11 advances another interesting idea. Since both the bubble chip and the Query-by-Example language mimic relational data model, the natural affinity makes the interpretation of QBE in bubble hardware languages easy.

Economic feasibility Bubbles are only available in small modules (64 to 256 kb/chip) and high cost/bit (comparable to floppy disks but higher than that for fixed-head or moving head files). A technology forecast is done in Ref. 7, which predicts that bubbles will compete with FHF in the early 1980's, and may even challenge MHF later.

1. S.Y. Lee and H. Chang, "Associative Search Bubble Devices for Content Addressable Memories", CompCon Fall '75, Digest of Papers, pp. 91-92, 1975.

2. S.Y. Lee and H. Chang, "Associative Search Bubble Devices for Content Addressible Memoreis and Array Logic", to be issued as an IBM Research Report, 1978.

3. H. Ishihara, N. Seki, and I. Wakabayashi, "A Premalloy Overlay and Conductor Bubble Logic Gate", Digest of Papers, the 7th Japan Applied Magnetics Conf., 5aA-8, 1975.

4. H. Ishihara, N. Seki, and I. Wakabayashi, "A Scheme of Magnetic Bubble Content Addressable Memory", Digest of Papers, the 7th Japan Applied Magnetics Conference, 5aA-9, 1975.

5. H. Ishihara and T. Hoshino, "An Interaction and Distribution Separated Bubble Logic Gate", Digest of Papers, the 8th Japan Applied Magnetics Conference, 20aA-1, 1976.

6. H. Ishihara, et al., "A Bubble Logic Circuit", Japan National Electro Communication Conference, page 1-244, 1978.

7. H. Chang, "Magnetic Bubble Memory Technology", (book) to be published by Marcel Dekker, Sept. 1978.

8. T.C. Chen and H. Chang, "Magnetic Bubble Memory and Logic", Advances in Computers, vol. 17, Academic Press, to be published in 1978.

9. H. Chang and A. Nigam, "Major-Minor Loop Chips Adapted for Associative Search in Relational Data Base", to be published in IEEE Trans. on Magnetics, 1978.

10. H. Chang, "Magnetic Bubbles and Relational Data Base", to be presented at the 4th International Conference on Very Large Data Bases, Sept. 13-15, 1978.

11. H. Chang, "Interpretation of Query-by-Example Language in Bubble Hardware Language", to be submitted in '78 ACM.

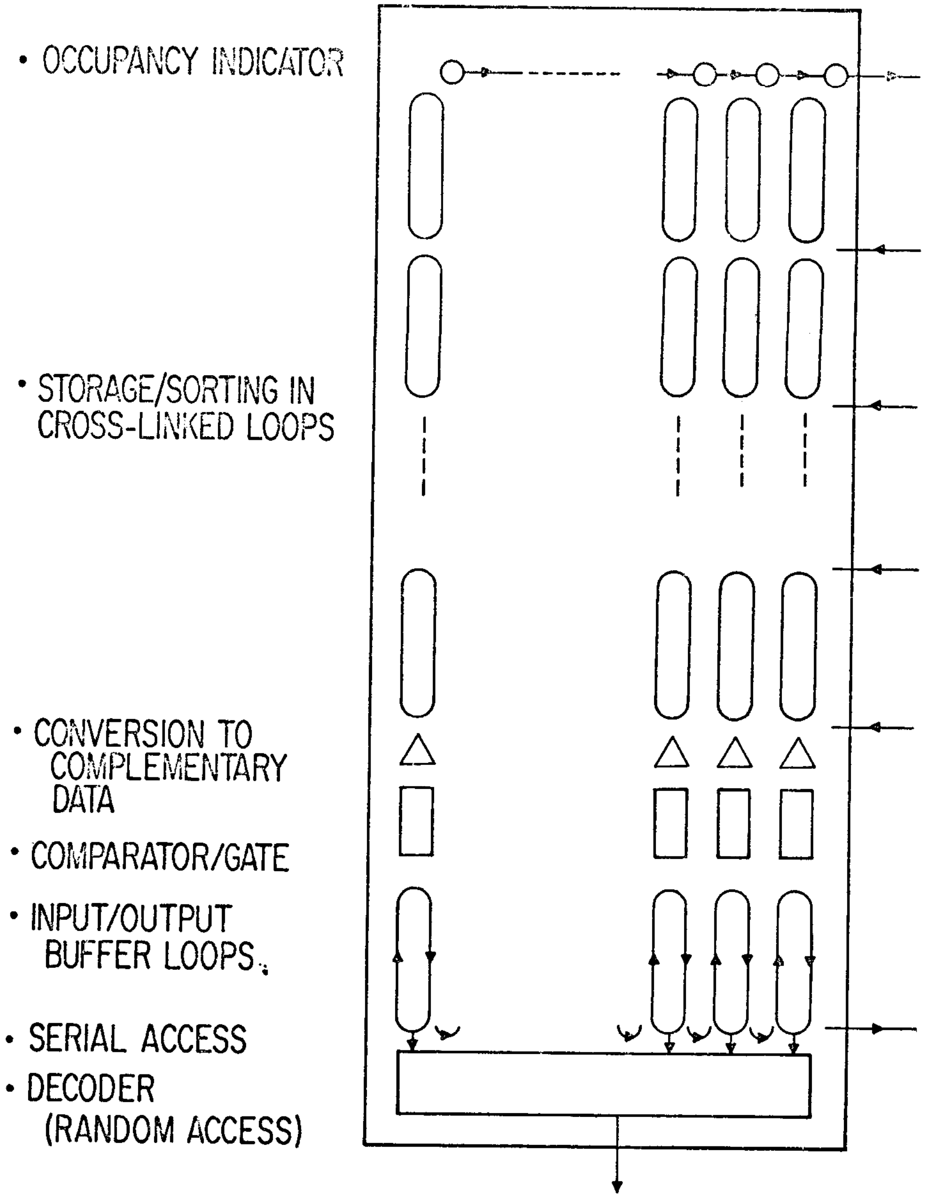
• OCCUPANCY INDICATOR
• STORAGE/SORTING IN CROSS-LINKED LOOPS
• CONVERSION TO COMPLEMENTARY DATA
• COMPARATOR/GATE
• INPUT/OUTPUT BUFFER LOOPS
• SERIAL ACCESS
• DECODER (RANDOM ACCESS)

Figure 1.