Lehigh University
Lehigh Preserve

Theses and Dissertations

1985

The back-end processor dedicated to data base management /

Kathleen L. Harter
Lehigh University

Follow this and additional works at: https://preserve.lehigh.edu/etd

 Part of the Industrial Engineering Commons

Recommended Citation
Harter, Kathleen L., "The back-end processor dedicated to data base management /" (1985). *Theses and Dissertations*. 4511.
https://preserve.lehigh.edu/etd/4511

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

The Back-end Processor
Dedicated to Database Management

by

Kathleen L. Harter

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1985

Certificate of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 20, 1985
(date)

John C. Wyant
Professor in Charge

S. C. Lowe
Chairman of Department

TABLE OF CONTENTS

ABSTRACT

Information has become a vital corporate resource. Many companies are installing new information management systems in order to increase their competitive position.

Traditionally, a database management system is software based and runs on a host computer under a particular operating system. Within the past several years, specialized database machines have been developed. The database machine, a back-end processor, consists of specially configured hardware and software dedicated to the database management function in conjunction with a host machine to perform data processing functions. This approach offers a solution to the overloaded systems and poor response time which exists with software based DBMS implemented on a general-purpose computer.

Since the back-end processor is completely dedicated to database management, the data management system can be more efficient than its software counterpart. The processor is engineered for high performance on data management functions since many of

these functions are performed in firmware, or microcode, rather than software. Also, information contained on the back-end processor can be accessed by various users - whether its a mainframe, personal computer or minicomputer.

A manufacturing company which utilizes the Intelligent Database Machine (IDM) manufactured by Britton-Lee was used to gain information on how this particular back-end processor performed in the manufacturing environment.

After several test between their old software based database management system known as POLARIS and the new system with the IDM, the IDM proved to be twice as fast. Although the back-end processor is a relatively new technology, it has proven to be a valuable resource for this company. Response time which is critical in today's marketplace has been improved and will continue to improve with the advances made to this special processor.

INTRODUCTION

Information is perhaps the single most important resource in industry today. Many companies are coming to the realization that their information resources can be used as potent strategic weapons. In fact, several firms are installing new information management systems to increase their competitive position.[1] Used strategically, these systems become a new factor in corporate strategic planning.

A database management system (DBMS) is one of the means by which Information Resource Management (IRM) can be implemented. The DBMS must be chosen for its ability to be flexible and process the data quickly and to serve all necessary user needs. Each company wants to install "the system" which will meet their needs such that they gain more competitive advantage.

Traditionally, DBMS are software based and run on the host computer with a particular operating system.

1. "Business is turning data into a potent strategic weapon.", Business Week, August 22, 1983, p. 92.

Three specific types exist: hierarchical, network and relational. Each of these three types has advantages and disadvantages. The needs of the business dictate which or what combination of the three kinds a business installs.

In the past several years, specialized database machines have been developed. The database machine (a back-end processor) consists of specially configured hardware and software dedicated to the database management function in conjunction with a host machine to perform data processing functions. Supporters of this new technology believe it offers a new approach to the old database management problem of overloaded systems and poor response with software based DBMS implemented on general-purpose computers.

An overloaded host system can attach a database machine to its environment thus extending the host computer resources, while performing the data management functions at a substantially lower overall cost and with greater speed.

The back-end database processor may be compared with a front-end processor relative to telecommunications. The front-end processor performs all communication functions leaving the host computer to perform less specialized tasks. This concept has proven highly successful in telecommunication networks.[2] It is anticipated that the back-end database processor will do as well with respect to data base processing.

Problem Definition

With data being such an important asset in industry today, access to it and timely system response become critical. Resource saturation is becoming an increasing problem with the demand placed on data processing functions increasing at an escalating rate.

Since the back-end processor is completely dedicated to database management, the data management system can be more efficient than its software

2. Basu, D., "Data base processors: a new trend.", Infosystems, August 1982, pp. 65-66+.

counterpart which runs on a general purpose computer under a general purpose operating system. The processor is engineered for high performance on data management functions. Many of these functions are performed with firmware, or microcode, rather than software. This is usually a less expensive, more reliable solution than performing the same tasks with software on a general purpose machine.

The dedicated processor provides a high level of efficiency and performance for a relatively low cost by performing the database management functions with a combination of software, hardware and firmware.

Information contained on the back-end processor can be accessed by numerous users - whether its a mainframe, minicomputer or personal computer and no matter who the vendor is. All can share the same data.

History of software database management systems

In 1964 the first significant data base management system (DBMS), I-D-S, was developed by Charles Bachman and associates at GE. It is the forerunner of many of

today's data base management systems since it provided a major source of ideas for the CODASYL DBTG data base proposals.[3]

Since that time, a growing awareness of data as a vital corporate resource has led to the development of numerous database models and systems. The reduction in cost for the underlying hardware is also a factor in this growing market. It is estimated that 50 companies market 54 different DBMS packages and that the sale of DBMS software generates $150 million per year in revenues.[4] In fact, the 1980's is believed to be the decade in which data base will become the dominant data management technology.[3]

Currently, over 15,000 database systems are installed in the United States and are being used directly by hundreds of thousands of people.[5] Most likely, every citizen in the U. S. today has used a

3. "The history of data base management systems.", Data Management, February 1983, pp. 21-23+.
4. Krass, P. and Weiner H., Datamation, October 1981, pp. 153+.
5. Blasgen, Michael W., "Database Systems.", Science, February 12, 1982, pp. 869+.

database - at least indirectly. Writing a check, making an airline reservation, paying a bill or using a credit card all create a database transaction.

Database management systems have supplanted the use of files for several reasons. At a reasonable cost, DBMS provide for data independence which protects the application programs from changes in hardware, operating systems, storage devices and various other system concerns. Data sharing is promoted among applications. Security is enhanced since only authorized individuals, terminals and programs perform specific functions. Integrity of data is preserved since hardware and/or software defects will not make the database inconsistent. DBMS provide ease of use to both system programmers who find it easy to install and maintain, and to application programmers and users. Finally, and perhaps most important, response time and throughput requirements are met.

A number of services are offered by a data base management system. First, it supplies a user's view of the data, or describes the way data appears to the user. This is usually quite different from the way

data are stored in the computer. The DBMS provides the mapping between what the user sees and how the data are actually stored.

The two basic approaches to providing users with "logical" views are network and relational. The latter is a high-level, mathematical set-oriented, "non-navigational" view with related procedural and non-procedural query languages. It is used in evolving environments where adaptability and ease of change are primary concerns. On the other hand, the network view is "navigational" in nature and requires the user (usually an application programmer) to be aware of how to move among record occurrences. It is utilized when the application is well structured and efficiency is critical. A DBMS provides the user a means to retrieve, update, insert and delete data from the database. Lastly, the DBMS provides execution control over the database, determining recovery options and level of concurrent access.

History of the back-end processor

Initial interest in back-end DBMS machines rose in 1976, only to fade around 1979 as hardware economics changed and communications overhead was better understood. It was feared that the communications cost would be considerably greater than the benefits gained by the use of this special hardware. In 1981, however, considerable interest and excitement was again created with the introduction of the IDM-500 by Britton-Lee.[6] It was the first specialized data base machine for the mini/maxi environment.

Data base machines operate in an environment including host computers and one form of data storage which is usually disk based. The host can be either a minicomputer, a mainframe, a personal computer, or a combination of all of these. The user interfaces with the host computer which accepts requests from the user

6. "The history of data base management systems.", Data Management, February 1983, pp. 21-23+.

for information and passes the request to the data base machine for processing. After the data base machine examines the database and retrieves the required data, it returns the result to the host.

There are several advantages to a system based on a data base machine. First, as was mentioned earlier, the data on these data base machines can be shared by a variety of host computers. The hosts can be from different vendors, thus having different characteristics, yet no duplication or conversion of data is required in order for the different hosts to access the data. Alternatively, a single host can have several data base machines attached to it, providing additional data management facilities as requirements increase. This capability can extend the life of the host and satisfy data storage and retrieval requirements at a reduced cost rather than having to upgrade the host to a larger system. The modularity this produces translates to future growth by data base sharing through multiple data base machines.

Second, the data base machine provides faster access to the database. Data base machines can perform

database operations much faster than a host computer since they are special-purpose computers engineered for high performance. The data base machine off-loads the host so it can perform other tasks such as accomodating more users and/or programs. Therefore, costs are lower for a given performance level since the data base machine is considerably less expensive than the host.[7] The database machine can also function as a network database server as computer networks become more commonplace and the need arises to provide shared database resources to all computers on the network.

Some database machines support the relational data model. DBMS's based on this model are known not only for power and flexibility, but also for slow performance and heavy use of computer resource. A relational DBMS allows greater flexibility in the manipulation of data without the need for the user to be concerned with how the data is stored or retrieved. The database machine can be optimized to efficiently support a relational DBMS.

7. Epstein, R., "Why database machines?", Datamation, July 1983, pp. 139-140+.

Finally, the back-end DBMS processor provides tighter security mechanisms since the only access to the data base is through the processor. It also provides for improved recoverability since the recovery processes will not impact the entire system.

Although the concept of a database machine was met with some skepticism at first, the initial response appears positive. It is believed that as software techniques and advances in VLSI circuits continue, by microcoding a well defined collection of subroutines which meet high performance and at a modest increase in price, the performance of the data base machine will increase much faster than host-resident software DBMS systems.[8]

8. Epstein, R., "Why database machines?", Datamation, July 1983, pp. 139-140+.

ARCHITECTURE

There are two commercially available database machines: the Britton-Lee IDM (Figure 2.1) and Teradata's DBC/1012 Data Base Computer (Figure 2.2). The latter is intended for use in the IBM mainframe environment while the IDM is for the mini/maxi environment.

Special Hardware

Special hardware exists for the nucleus of a back-end processor system. In the DBC/1012 Data Base Computer manufactured by Teradata, two subsystems are present - Interface Processors (IFPS) and the Interprocessor buses (YNETS). The IFP manages the dialogue between the user session in the host and the DBC/1012 Data Base Computer. In order to achieve reliability and fail-safe operation when a DBC/1012 Data Base Computer has several hosts, at least two IFP's are usually assigned to each host. Dual Interprocessor buses or YNETS connect the IFPs with

another subsystem known as the AMPs or Access Module Processors. To share the communication load, both YNETS operate concurrently. Each YNET is an array of active logic that supplies selection and sorting functions which is a complete contrast to an ordinary bus which contains no logic at all. Each YNET is independent of the other (serving as backup for the other) ensuring continuous operation in the event of a system failure of either. This independence extends to physical partitioning, electrical power, internal clocks and interface logic. The YNET provides communication between single processors, from a single processor to a group of processors, from a group of processors to a single processor and between various processors. Data returned from the AMPs is merged across the YNET to the originating IFP and is automatically sorted into the sequence specified by its request. Thus the YNET acts as a ultra-high-speed sort/merge processor.

Two modules, the Database Processor and the Database Accelerator, makeup the IDM's special hardware. The former translates among different hosts'

data types, executes most of the software in the system, and manages all system resources. It evaluates the IDM configuration when the IDM is turned on and takes advantage of all available modules. If the Database Accelerator (which is an optional feature of an IDM) is present, the Database Processor senses its availability and issues a call for its service. Specifically designed to perform relational database functions, the database accelerator is a very high speed processor. The addition of the Accelerator to the IDM can improve throughput by a factor of 2 to 10 depending on the operation being performed. This processor can initiate disk activity and will search disk pages as they are transferred to memory. It may even complete processing of the page before it is completely read in. Most of the time-consuming work is performed by the Accelerator under the direction of the Database Processor since the Accelerator and the IDM system software are structured this way.

Host-resident software

In the IDM, a host interface module or board exists which accepts commands from the host(s), checks to ensure all information that was sent to and from the host(s) has been transmitted and received correctly. It then coordinates retransmission in case of error and notifies the Database Processor that it is transferring the request into the main memory for processing.

The IDM and its host(s) communicate over an IEEE-488 parallel interface which has a maximum data rate of 170 K bytes/second or over an RS-232 serial interface with programmable baud rates of up to 19,200 baud. This is currently the limitations of the IDM. However, with the addition of a network to the system the baud rate can be increased. Separate parallel host interface modules can be assigned to each host in high transaction rate environments, thus increasing parallelism - two host can then communicate with the IDM simultaneously.

In the DBC/1012 Data Base Computer environment, the Teradata Director Program (TDP) resides on the host computer on a dedicated address space in the host and manages the communication between the users of the host(s) and the DBC/1012 Data Base Computer. As users request data either from a batch mode or interactively, the TDP will accept these requests. It in turn creates a request message and communicates it over a block multiplexer channel for processing by an Interface Processor contained in the Data Base Computer itself.

Disk Storage

In the DBC/1012 Data Base Computer a subsystem of access module processors (AMPS) handle most data manipulation functions. An AMP consists of: the YNET Interface, Data Base Manager and the Disk Interface. The AMP responds to the IFP requests received from the YNET by retrieving data from or storing data on its DSU (Disk Storage Unit). The Data Base Manager services insert, delete, retrieve and update operations on a data base as well as rollback and recovery, data base reorganization and logging. The Disk Interface passes

and receives commands to and from the DSUs. The Disk Storage Unit (DSUs) is the data storage medium for the DBC/1012 Data Base Computer. The DSU space is divided conceptually into: system area, primary copy area and fallback copy area. Under normal conditions, the primary copy area is accessed. When the primary copy is temporarily unavailable, the fallback copy is accessed which stores a copy of data which is primary on other AMPs within the system. The system area stores system tables and programs.

User requests for data are processed in parallel; data is accessed simultaneously on all AMPs, greatly increasing system throughput. This is the beauty of back-end processors operating in a parallel multi-processing capability; as DSUs and their AMPs are added, storage capacity is increased and processing becomes faster.

When the database is created, rows of a table are distributed among the AMPs. This is also true of the fallback copy. However the fallback copy of any row is stored on a different AMP from the one where the primary copy exists. This ensures that the stored data

remains available should an AMP fail. In the IDM, memory is available which holds the system information, the relational DBMS software, stored commands, user work space and the most frequently used data pages. The amount of IDM memory can be enlarged for optimum IDM throughput. The IDM memory subsystem consists of a Memory Timing and Control (MTC) module and at least one memory module. Memory modules are available in 1 megabyte capacities. The MTC manages the main memory subsystem and provides single bit error correction and double bit error detection. Controlling one or more memory storage boards, the MTC supplies data at the processing speed of the Database Accelerator.

The IDM Disk Controller moves data between external disks and the main memory of the IDM. Disk drives made by different vendors can be directly connected to the IDM since the disk controller interface is compatible with industry standards. One controller can manage 1-4 disk drives which are dedicated to the IDM and only accessed through the IDM. It is designed to work with the Database Accelerator

which can process data at the speed it is read from the disk.

Relational Database Concept

Both back-end processors previously mentioned support the relational database model. It has been acclaimed as the most flexible and elegant of the data models. Based on a rigorous mathematical structure, its data can be easily described and accessed. End users are put in direct touch with the information stored in the processor. Also, programmers' productivity is increased in the development of application programs. With this model, the data processing professional does not have to navigate his or her way along access paths to reach target data. The relational database model uses what is known as associative addressing which surpasses the traditional positional addressing used in hierarchical and network models.[9]

9. Codd, E. F., "Relational Database: A Practical Foundation for Productivity.", Communications of the ACM, February 1982, pp. 109-117.

In the relational database, every piece of data can be uniquely addressed by means of the relation name, primary key value and the attribute name. Thus, associative addressing allows users to leave it to the system to select appropriate access paths when retrieving data, and determine the details of placement of a new piece of information that is being inserted into a database. Addressing data by value, rather than by position, also boosts the productivity of programmers. (Positions of items in sequence are normally subject to change and a person can find it difficult keeping track of them, especially when the sequences contain many items.)

Software relational DBMS systems running on a host system under control of a general purpose operating system are plagued with performance problems. These systems generally are not fast enough to be commercially viable for transaction processing. With hardware (back-end processor) designed specifically to perform relational DBMS function, however, the situation is changed. The back-end processor offers all the benefits of the relational

data model while satisfying the high performance criteria of DBMS systems.

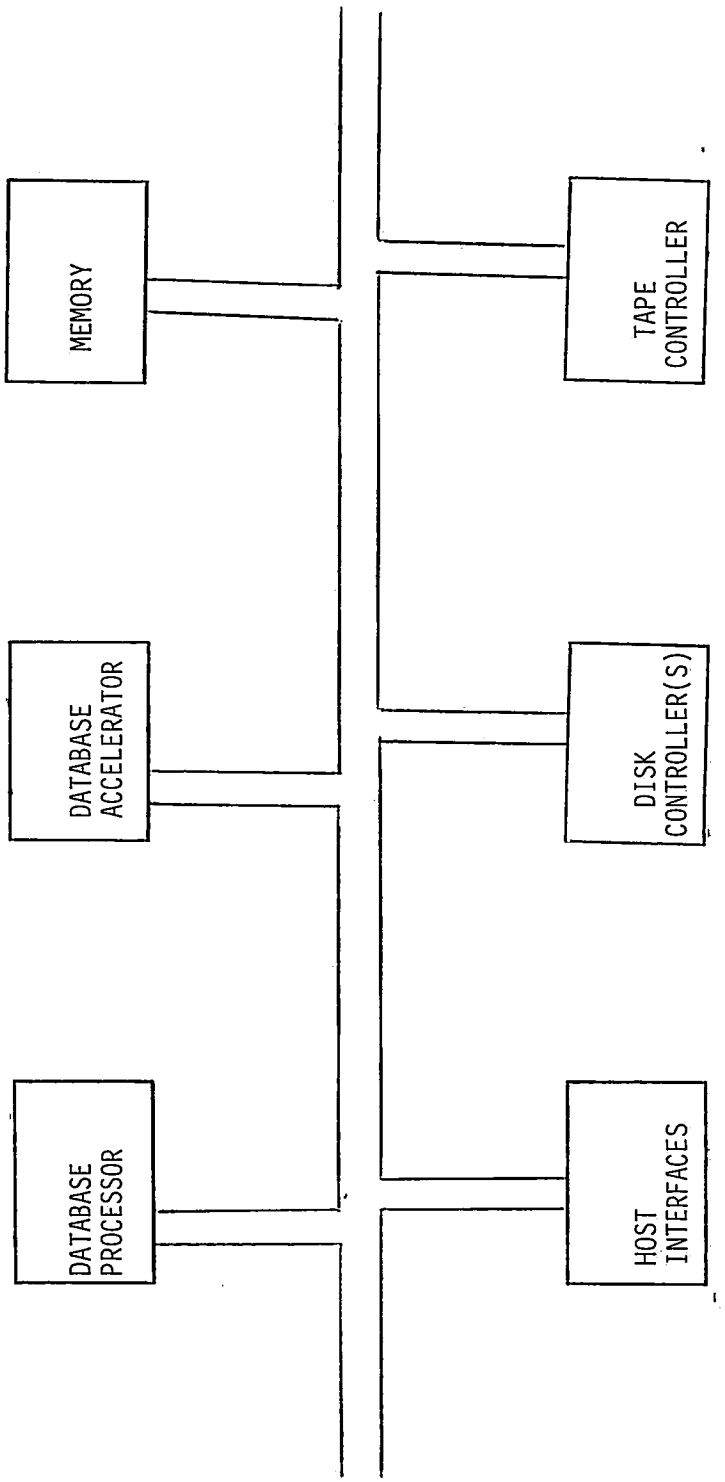

Figure 2.1 - Intelligent Database Machine

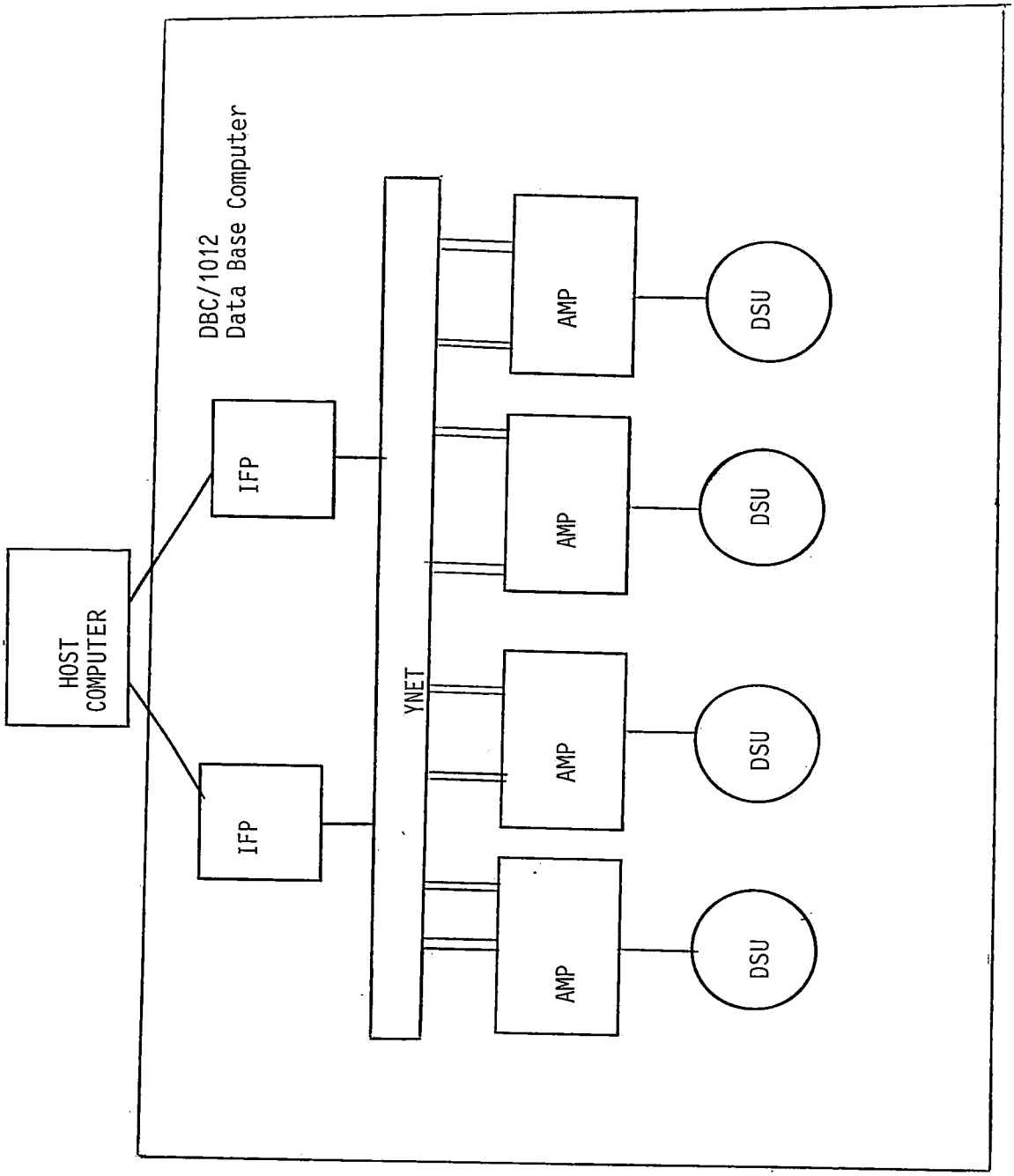

Figure 2.2 - DBC/1012 Data Base Computer

SYSTEM FACILITIES

Numerous facilities exist for controlling the operation and administration of the back-end processor. These are data protection and space allocation. Data protection consists of concurrency control, transaction log, transaction management and recovery.

Data Protection

Data protection guarantees the integrity of a data base during concurrent access by users, following abnormal termination of an application program, and after the failure of a host system or the back-end processor. It is accomplished by several means.

Concurrency Control

The consistency of shared data stored in the data base is not violated by users who are concurrently trying to change the same data. In addition, a facility must ensure the user he will never see inconsistent data where it is important to him.

The concurrency control is implemented by "locking" the shared data. A user who is holding a lock is ensured that the data remains consistent and any other request for access is held until the lock is released.

The system automatically performs the locking function. During the processing of a request, locks are acquired and upon completion, they are released. Three levels of locks and three types of locks are available. The level of locks are database, table and table row. The types of locks are exclusive, write, read and access. Exclusive locks are the most restrictive and apply only to a data base or table. With this type of lock all other users are locked out, however, it is rarely used. Write locks allow a single user to modify data while locking out all others except readers not concerned with data consistency. No new read locks are allowed until a write lock is released. A read lock ensures consistency during read operations. No modification of a table is permitted while several users hold read locks on that table. When a user in not concerned with data consistency, he specifies an

access lock. This allows a greater number of users to share the same data concurrently.

Transaction Log

A sequential record of changes made to a data base is a transaction log. One record of the log consists of the user identification, time stamp, and a copy of the data prior to modification. The log is used to "roll back" or restore a data base to its state before the transaction was started if the transaction is not completed or aborted.

Transaction Management

Transactions are managed so as to maintain consistent data without necessarily locking resources. Either all transactions, whether implicit or explicit, are performed or none are performed. The entire request will be aborted if for some reason the entire request can not be performed successfully.

Recovery

When a transaction is aborted for any reason, recovery takes place which will, using the transaction

log, restore the database(s) to their pre-failure condition. Then, using the logged entries, the database will be updated with the completed transactions.

Space Allocation and Access Control

The DBC/1012 Data Base Computer initially contains one database named "DBC" which the system administrator manages. All other organizations databases are formed from this disk space by the administrator.

The administrator's first concern is with security, so (s)he will typically create an administrator's data base from the DBC with all the remaining disk space not needed for the system tables. It is from the administrator's database that allocation is made for all other organization's databases. When the administrator allocates space for an organization, he also creates one person as a supervisory user who then allocates space to create the subordinate databases his organization may want. This hierarchical ownership structure gives the owner of a database

complete control over the security of owned data. The owner is free to dump the database and/or control access to its data by granting or revoking access privileges to it.

USER FACILITIES

Data Dictionary

A data dictionary is a service provided by a data management system to its users. The dictionary allows the users to interactively define the data schema and to look up that schema once defined. Through the use of the IDM system relations and stored commands, the data dictionary functions are performed.

Several system relations exist in each database. Three relations of particular interest for performing data dictionary functions are: relation, attribute and description. The first, or 'relation' relation contains a list of all relations in the database which include the IDM-assigned relation id, the name of the relation, the user id of the owner of the relation, the number of tuples, and other commands the IDM needs to process commands on the relation. The 'attribute' relation contains information about each attribute of each relation in the database. The 'attribute' type, name, relation id of the relation that the attribute

belongs to, and the IDM-assigned attribute id are kept in the 'attribute' relation. The 'descriptions' relation associates one or more descriptions with a relation id/attribute id pair. Should the attribute id be 0, the description is associated with the relation id only.

When a relation is created by a user, the 'relation' and 'attribute' relations are automatically updated. After creation, the user can add a tuple to the 'descriptions' relation to store information about the relation. When the relation is destroyed, the IDM will automatically delete the associated tuples in the three relations.

Stored commands which users define can facilitiate access to the data dictionary. The user defines a subroutine of commands which s(he) will probably use frequently and then can access them with a single command.

The database is initially empty except for system relations that the IDM automatically maintains as part of the integrated data dictionary. This dictionary

contains current information about: the name of each relation in the database and its description; each attribute of a given relation, its type and the description of the attribute; all database users and groups of users with their access privileges; a catalog of all clustered and nonclustered indices associated with any relation; and any changes made to logged relations.

IDM's relational DBMS command set

Some of the basic functions supported by the IDM command set are as follows:

Create/destroy database
Create/destroy relation
Append data to a relation
Retrieve data from a relation or relations
Permit/Deny access to data in a relation
Modify or delete data in a relation
Create/destroy indices on data
Create/destroy view
Define stored command

Begin/End transaction

Audit changes on a relation

Create/Read/Write/Destroy random access files

Initially (after a logical analysis of the data relationships has been performed) the DBA can start to create the database and establish the relations. If the user does not specify how much space should be allocated for the database when he issues a 'create database' command, the IDM will assign the first available space. (A special database is maintained by the IDM to hold information about each of the databases, the IDM's environment, user-specified parameters, and the IDM relational DBMS software.) The user may retrieve information in the exact same way he would access data from his own relations.

Next, the database must be opened by the user who wishes to issue commands to that database. The purpose of the OPEN command is to assign a process ID through the IDM to the user so that all work done for that user can be identified uniquely as his. The user proceeds to issue a 'create relation' command to establish a relation in the opened database. If all changes to

this relation were to be logged, the 'logging' option would have to be specified on the 'create relation' command. 'Logging' permits the user to obtain an audit trail of exactly what modifications were made to the relation and who made the changes. It is a critical feature in the IDM's crash/recovery procedure. The result of a 'create relation' command is an empty relation. Rows or tuples are then added. The IDM system relations are automatically updated to indicate the existence of the new relation.

The user may take advantage of the bulk load facility, COPY IN, which is used for the inital load of a database or transferring data between the IDM and the host, between two IDM's or between different databases on the same IDM. The other option for the user is to use the APPEND command in which a tuple at a time is loaded into the database just created. When issuing one of these commands, if information for one of the attributes of the relation is missing, zeros will be used in numeric fields while spaces are inserted in the character fields.

Once the data is loaded, the user can retrieve the data by utilizing the RETRIEVE command. This command supports the three basic operators of the relational data model: select, join and project. These operators can be combined with Boolean and comparison operators and aggregation for a precise description of the information desired.

Data in relations can be updated by the REPLACE, DELETE, and DESTROY commands supported by the IDM. The DELETE command deletes only one tuple of the relation whereas the DESTROY command will delete the entire relation.

Tuning the IDM database

Facilities exist to help the DBA optimize the database for the typical types of access and updates. The special functions are: IDM INDEX, VIEW, and STORED COMMAND.

Indices

Two types of indices, the clustered and the non-clustered, offer means of accessing data in relations quickly through the use of pointers. The clustered index causes the data to be maintained in physical sort order on the clustered index field. There can only be one clustered index per relation. To access the relation by other attributes, non-clustered indices may be created. Each entry of the non-clustered index contains the value of the data item for which the index was built and a pointer to the disk location that contains the tuple with that value. When ever the access patterns change, the indices may be destroyed and new ones built.

Views

The VIEW capability of the IDM virtually eliminates the need to duplicate data. The database can be tuned by the DBA through creating VIEWS which conform to the record-layouts needed by various application programs. Views are virtual tables that combine commonly related data from one or more

relations. Views can also be used to restrict access to sensitive data since the user does not have to be aware that he is only 'viewing' data which has been materialized from underlying relations. All changes to the underlying relations will be reflected in the VIEW automatically by the IDM. Views can be subsets of a relation or a combination of one or more relations. Use of VIEWS simplifies accessing information from the database for the non-dp professional user. Additionally, it makes programs more independent of the data in the database as well as simplifying the programming task.

Stored Commands

When several commands are always executed together, they can be sent to the IDM as a command group where they are preprocessed and stored for subsequent execution. All the necessary preprocessing is done by the IDM when the command group is stored. This is a particularly powerful feature. After the command group has been interpreted and sent to the IDM for storage, the host will never have to interpret the group of commands again. Instead, the host simply

sends the command name to the IDM with the appropriate parameters, thus reducing the amount of information that must be transmitted to the IDM. Also, the time to execute the command set is significantly reduced since the preprocessing step is eliminated as a side benefit of using stored commands. Application programs running on the host are smaller, more efficient and easier to maintain.

Securing the database

Two protection commands exist to protect data from unauthorized access: PERMIT and DENY. Activities which can be permitted or denied are read, execute, write, create index, create database and create relation. Keywords following the two protection commands specify what is being permitted/denied and to whom it is permitted/denied. Based upon user identifier, a user is denied/permitted access to a relation. The protection can be on an attribute or relation basis.

BACK-UP AND RECOVERY

The IDM transaction log is an essential part of the IDM's means of protecting data against loss and/or unauthorized access. Data could be potentially lost in one of two ways: hard failures or head crashes, and soft failures or power failure.

Each command, unless it is one of a group of commands between the BEGIN and END commands, is a transaction. These transactions always appear to have run to completion or never to have been started. In order to maintain consistency in the database, the IDM must keep track of all changes to the database. Then, using before and after images, the IDM can undo any partially completed transactions and ensure all finished transactions are properly recorded in the database. Just as the IDM knows the END command signals that the transactions should be committed, it also knows when the ABORT command is issued it must use its transaction log to automatically backout all changes made to the database since the BEGIN command was executed.

Enough information to reconstruct the change and/or back it out is recorded automatically in the IDM transaction log each time an update to a relation occurs. Before the transaction is committed from IDM memory to the database, the IDM transaction log is written from IDM memory to disk. After the actual transaction is committed to disk, a 'done' token is written to the transaction log.

After a soft crash in which the transaction log is left intact and the IDM is rebooted, a recovery program is run against the database. The program, by looking into the IDM transaction logs can detect those transactions which were only partially completed. Such transactions are backed out. Check points noted in the transaction log help to speed up this procedure. A check point forces all data blocks which have been modified to be written out to disk. Thus, the recovery program can proceed from the log's most recent check point instead of the beginning of the transaction log.

Transactions can be left in one of four states after a soft crash:

1) Transactions completed and modified data written out to disk before crash. Everything okay.
2) Transaction not completed and data never written out. Everything okay.
3) Transactions completed, database changes not written out. By seeing the 'done' token in the transaction log, the IDM will update the modified database data information stored in the log.
4) Transactions not completed, some of database changes written out. Here, IDM will not find the 'done' token, therefore, it uses the before images of the partially completed transaction log to restore data to its original state.

IDM backup and recovery procedures must be followed to protect against hard disk crashes when the transaction logs are not left intact.

The IDM performs backup and recovery by database which quarantees the backup of each database is

consistent. This is unlike most other systems which do it by disk device.

The IDM crash/recovery procedure requires occasional backup of the complete database and frequent backups of the transaction log.

Backup Procedures

The transaction log is updated each time a change is made to a logged relation which was specified at creation time of the relation. The following procedure for handling the possibility of catastrophic loss of data is to :

1) dump the entire database occasionally
2) dump the transaction log frequently
3) in case of data loss, load the last database dump
4) load and roll the transaction logs forward

All relations important to the application must be logged for the success of the above procedure. Normally, a full database dump should be obtained shortly after the initial load of the database.

Additional full database dumps should be performed at regular intervals. Between the full dumps, periodic dumps of the transaction log should be taken. The frequency is determined by how valuable the data is and how dynamic the databases are. When a database needs to be restored, the DBA needs to LOAD the most recent full dump and then LOAD and ROLLFORWARD all subsequent transaction log dumps. Transaction log dumps can occur while users are actively updating the database while full dumps require the database to be offline until the dump is complete. At the completion of these dumps, the IDM will automatically truncate the transaction log and begin logging all over. It is important for users to know what objects are logged. This is noted on the create relation command. Only logged objects can take advantage of the back-up procedures. By viewing the RELATION relation, a user can obtain this information.

Recovery Procedures

To recover a database which may be on a disk pack rendered unusable by a disk crash, the DBA would first recreate the destroyed database. Next, he loads the

most recent full database dump. Then he loads and rolls forward any subsequent dumps of the transaction logs. When this is completed, the database is ready to use. It is important to note that any updates that were made to the database after the last transaction log dump and before the crash will be lost. The DBA needs to define procedures to provide for re-entry of data which was unavoidably lost between the last transaction log backup and the crash.

COMMUNICATIONS

Several interfaces exist in the communication between the host and the IDM channel. First is the interface between the host user program and the host operating system. Next is the interface between the host operating system and the host communication device. Last is the interface between the host communication device and the IDM channel. The host communication device is the handler program and the hardware of the I/O device that is directly communicating with the IDM.

A set of commands for communicating between host and channel are supported by the IDM channel. The host issues a subset of these commands primarily to write to the IDM and to read from the IDM and to cancel IDM requests.

Host-to-IDM Channel Communication Commands

A protocol for sending IDM commands and requesting results is supported by the IDM. This allows for

flexibility in host operating systems or host interface programs. Three separate communication commands support each of the operations of writing to the IDM and reading from the IDM. The host can issue any of the commands to read from or write to the IDM.

The different READ and WRITE communication commands exist so that the host may decide what course of action to take if its request can not be serviced immediately by the IDM. The host operating system has three choices: it can wait until the request can be granted, not wait until the request can be granted and try again at a later time, or it can not wait and have the IDM call the host when the host should reissue the request. The third option would most likely be adopted by a host running a timesharing operating system which would allow the host to be efficient in operation. In a stand alone environment, however, the host would typically choose to wait until the request can be granted. The second alternative would most likely be utilized by a host who did not want to wait but also one that does not wish to support calls from the IDM indicating the request should be re-issued. Usually

host characteristics determine the appropriate WRITE and READ commands.

When issuing a WRITE command, the operating system must also supply the database instantiation number, host user id, process id and the number of bytes of data. If the IDM is currently able to accept the WRITE command from the host, it will respond YES, otherwise NO will be sent to the host. When the response is YES, the IDM along with the YES response sends a count indicating the number of bytes of data it can write. The count is always less than or equal to the count sent by the host. The host will then send the data to the IDM. If the response was NO, the IDM will send an error code indicating why the request can not be granted.

The READ communication command enables the host to request results from the IDM. As in the WRITE command, the four parameters must be used. After the host issues a READ command, the IDM responds with a YES or NO. If results are available, the response is YES, otherwise it is NO. Along with the YES response, the IDM also sends a count indicating the number of bytes

to read. The count is less than or equal to the count sent by the host. The data is then sent by the IDM. The response will be NO if results are not available or an error has occurred in which case an error code will also be sent.

The CANCEL command is used to cancel the current IDM command issued by the host program. The database instantiation number and the process id of the program that issued the command must be provided. The current command is aborted and the results are flushed. To ensure all buffers containing data for this database instantiation number are flushed within the host system and IDM, the CANCEL command is succeeded by an acknowledgement which the host must read. In effect, the host issues a CANCEL command to the IDM and the IDM immediately responds with a YES unless an error has occurred. The host will then have to issue a READ command on the same database instantiation number. The IDM responds with a NO accompanied by a special error code indicating the CANCEL was acknowledged.

The HELLO command is used by the host to tell the IDM it is still on-line. The host must communicate

with the IDM at least every 'x' minutes or the IDM will assume the host is down. The 'x' minutes is defined by the host.

IDM Channel-to-Host Communication Commands

Three asynchronous communication commands are sent to the host by the IDM when the host issues a READ or WRITE in which it will wait for the IDM to respond if the initial request could not be fulfilled. The RESULTS command which supplies the process id and database instantiation number of the requestor tells the host that requested results are available. In return the host will send the respective READCALL command.

The command RAVAIL informs the host that the IDM can now accept READCALL and WRCALL requests. The WCONTINUE command informs the host that a particular write request can continue. The WAVAIL tells the host the IDM is now available for writing.

SPECIAL PURPOSE IN MANUFACTURING ENVIRONMENT

Application

In this section, information will be used which has been obtained from a manufacturing company which requests to remain anonymous in this document.

At this company, seven IDM500's which have the database accelerator and use the RS232 serial interface are used for three distinct systems which are: Process Control, MAGNETS, and 411. The host computers which access the IDM are the 3B20S and the PC6300. The 411 system is an on-line phone book which lists plant personnel by phone number and their location in the plant. The MAGNETS system is a production system which aids in the tracking of magnets as they are mounted onto finished semiconductors. The main system utilizing the IDM is Process Control (PC) or a lot tracking/lot routing system.

The PC system is an information system which provides on-line control for Very Large Scale

Integration (VLSI) manufacture. The goals of the system are to enhance productivity or increase shop throughput, maximize yield or improve device yields, provide a base for the development of future semiconductor manufacturing techniques, reduce investment, and reduce cost.

Currently, PC software runs on the AT&T 3B20S minicomputer and the Britton-Lee Intelligent Database Machine (IDM). The 3B20 holds the UNIX operating system and the PC software. The IDM stores the information about how a product should be manufactured and where a product is located in its manufacturing process.

The product/process engineer creates a recipe or routing for which a product is to be created. This routing is stored on the IDM. Shop operators, using terminals in the manufacturing area, step the product through the various process steps outlined previously by the engineer in the recipe or routing until manufacture is completed. As products are manufactured, information about them is copied from the

IDM by the 3B20 which then uses this information to interface with other computer systems.

For one area of the shop, the system configuration consists of two 3B20 minicomputers tied to one IDM. The IDM has two mirrored disks and one tape drive while each of the 3B20 computers have several disk drives and are connected to various printers and terminals throughout the plant. Normally, one host can have several IDM's connected to it, however, at this company this is not the case since PC would not be able to run that way.

In the PC system, anywhere from 1 to 10 databases per IDM exists with approximately 60 relations per database and 12 attributes per relation. Anywhere from 0 to +30,000 tuples/relation can occur. The IDM memory can be up to 6 Megabytes.

User Functions

The users of the PC systems, mainly shop operators, enter his/her bar code which the system recognizes upon login. The operator is now able to

perform numerous functions based on the security level assigned to him/her. Some of the functions consist of:

- hold and release lots placed in queue
- split or merge lots
- acquire help with the PC commands
- change priority levels assigned to lots
- perform editing functions to different data bases for each lot
- enter data collected for a lot

In order to complete one manufacturing process step, a series of five commands better known as the "Circle of Life" are performed. Approximately 100 circles of life are performed every hour. The first of these commands allows the user to view and select lots for processing on a facility. The next command displays process instructions allowing the user to begin processing work. Next the user is instructed as to whether any special data is required and is helped in recording this data. The next command records the completion of a process and the final command permits the user to document the lot yield following the process step.

As the operator processes a lot through its routing, the information is stored on the IDM. This information is retrieved several times as the product moves through the manufacturing environment. As users request information about a particular product, the 3B20's will ask the IDM to display information about the product on the users terminal. Should the IDM go out of service, then the entire associated shop area will be unable to access the PC system. When this occurs, the shop operators will record the data on cleanroom paper for later entry into the system by the system trainers. Production time is not lost, however, there is one problem. Lots no longer have the distinction of being in a hold status. Therefore, operators may start working on a lot which really should not be worked on.

The PC system does utilize numerous stored commands. There is also a data dictionary in the system relations supported by the IDM. The dictionary is used by the query language on every database when one performs a retrieve, replace, append, etc. to make sure fields are the correct size and type.

Daily Procedures

Once a day, database backups are performed on each of the production databases. Transaction log dumps are not taken since it is believed that they take too long to "rollforward".

So, what happens if a crash occurs after the database backups are performed? There is a 24 hour period in which the system is vulnerable. At this company, mirrored disks are utilized. Mirrored disks are identical copies of disks. The reliability of one disk is 99.96. If one disk goes out of service (which has happened at this company) the shop can continue to operate on the one good disk. Should both disks crash, the current status of each lot is maintained on a file in the 3B20 and updated once an hour. When the disks are back in working condition, the system trainers will manually, using the listing showing the current status of each lot, perform a command on each lot which will update the system and processing will continue from this point.

Response Time

Prior to the IDM installation, the database management system known as POLARIS was used in the Process Control system. POLARIS was a relational database management system which ran on the 3B20. IDM was chosen to replace this database management system in order to improve response time and also to take the load off of the 3B20 minicomputer. It has proven to be successful in both respects. Response time studies show that with the IDM, response has improved substantially. In fact, it is twice as fast as it was with POLARIS.

Advantages and Disadvantages

The major benefit of the IDM at this company is that being a back-end processor, it takes the task of database management off the 3B20, thus providing response time improvements. The IDM's major drawback is that it is the weak link in the Process Control system. If it should go out of service, the shop area supported by it will be down until the IDM can be

brought back on-line. (However, this company keeps a full set of spare boards to be used to remedy the problem quickly. Also, if there is a total IDM failure, the spare IDM will be utilized.)

CONCLUSION

The intelligent back-end database processor although a relatively new technology, has proven to be a valuable resource, at least at this company. Response time, which is critical in the competitive marketplace today, has been improved with the use of the back-end processor. This has helped to improve the throughput of the shop.

Although the IDM has some problems which Britton-Lee continues to work out, the benefit it has provided in the manufacturing environment is significant enough to warrant its continuing use.

BIBLIOGRAPHY

Articles

Abbott, J. L., "A comparison of five database management programs.", Byte, May 1983, pp. 220+.

Basu, D., "Data base processors: a new trend.", Infosystems, August 1982, pp. 65-66+.

Basu, D., "Touching the bases in relational systems.", Data Management, February 1983, pp. 28-29.

Blasgen, Michael W., "Data base systems.", Science, February 12, 1982, pp. 869+.

Bridges, T., "Data base machines-what and why.", Data Management, November 1982, pp 14-16.

Britton, D., "Fitting the tool to the job is the path to the highest performance.", Electronic Design, January 26, 1984.

Brousell, D. R., "IBM: relational DBMS in tests for MVS shops.", Electronic News, June 13, 1983, pp. 26+.

Carlyle, R. E., "The PC makes friends.", Datamation, May 15, 1984.

Codd, E. F., "Relational Database: A Practical Foundation for Productivity.", Communications of the ACM, February 1982, pp. 109-117.

Curtice, R. M. and Jones, P. E., "Database: the bedrock of business.", Datamation, June 15, 1984, pp. 163-164+.

Emmett, R. and Verity, J. W., "MIPs, MIPs, MIPs [new relational data base trio from IBM will sell more hardware].", Datamation, July 1983, pp 72+.

Epstein, R., "Why database machines?", Datamation, July 1983, pp. 139-140+.

Epstein, R. and Hawthorn, P., "Aid in the 80's.", Datamation, pp. 154-156.

Fireworker, R. B. and Konzet, J. D., "Trends in data base management systems.", System Management, May 1982, pp. 30-33.

Fisher, S., "Database engines.", Interact, September 1984, pp. 26-28.

Greenwald, J., "Which data base system is best?", Venture, February 1984, pp. 36.

Harvey, R. E., "Industrial data base: the key to productivity in the factory of the future.", Iron Age, October 21, 1983, pp. 74+.

Jacobson, E. and Crystal, M. I., "Fred, a front end for databases.", Online, September 1982, pp. 27-29.

Jones, K., "Laymen are target users of uc database machine.", Mini-Micro System, July 1982, pp. 83.

Krass, P. and Weiner H., Datamation, October 1981, pp. 153+.

Leigh, W., "Natural language for database access.", Journal of System Management, November 1983, pp. 22-24.

Lettieri, L., "Intel aims database machine at small systems.", Mini-Micro System, July 1982, pp. 18-19.

McKendrick, M. D., "Dedicated machine offers host-independent data-base management.", Computer Technology Review, Winter 1982, pp 11-17.

Morgan, K., "Database processor offers small-system options.", Mini-Micro System, October 1982, pp. 237-238+.

Rauzino, V. C., "The looming battle between data base machines and software data base management systems.", Computerworld, 1981.

Sandberg, G., "A primer on relational data base concepts.", IBM Systems Journal, 1981, pp. 109-117.

Sharpe, W. F., "Relational data base management systems.", Financial Analyst Journal, January/February 1984.

Sweet, F., "What, if anything, is a relational database?", Datamation, July 15, 1984, pp. 118-120+.

"A budding mass market for data bases.", Business Week, January 17, 1983, pp. 128+.

"Britton-Lee aims database machine at small systems.", Mini-Micro Systems, February 1982.

"Business is turning data into a potent strategic weapon.", Business Week, August 22, 1983, p. 92.

"Data bases plugging into the world.", Esquire, September 1983, pp. 271+.

"New development in micro data-base management.", Office, May 1984.

"Relational DBMS provides high performance.", Infosystems, June 1984, pp. 14-15.

"Relational database systems are here!", EDP Analyzer, October 1982, pp. 1-12+.

"The history of data base management systems.", Data Management, February 1983, pp 21-23+.

"Working with databases.", Radio-Election, April 1984, pp. 76+.

Reports

Kent, W., A Simple Guide to Five Normal Forms in Relational Database Theory.

Naval Data Automation Command, Washington D. C., Review of Available Database Machine Technology.

Other Sources

DBC/1012 Data Base Computer Concepts and Facilities. Teradata Corporation, 1984.

Design Decisions for the Intelligent Database Machine. Epstein, R. and Hawthorn, P.

IDM 500 Series. Britton-Lee, Inc.

Intelligent Database Machine. Britton-Lee, Inc.

BIOGRAPHY

Kathleen Harter was born in Reading, Pa. on March 1, 1957 to Lawrence and Louise Seaman of Reading, Pa. After graduating from Reading Central Catholic High School in June, 1975, Kathy entered Lock Haven State College where she majored in Math/Computer Science. In May, 1979 she received a Bachelor of Science degree from Lock Haven State College and graduated cum laude.

Upon graduating from Lock Haven, Kathy joined Western Electric at Cedar Crest, Allentown, Pa. as an Information Systems Designer. Over the years she advanced to the title of Information Systems Staff Member. In October 1983, she transferred to the Reading Works as a member of the Shop Flow team. She is also currently on the board of directors for the Schuylkill Valley Chapter of the Data Processing Management Association.

After a one year interval away from college, she entered Lehigh University in the Fall of 1980 as a part-time candidate for a Master of Science degree in the Industrial Engineering program. Kathy's particular

area of interest is information systems. This endeavor continues to the present.