THE PRELIMINARY REESARCH OF DATA FLOW MACHINE AND DATA BASE MACHINE

AS THE BASIC ARCHITECTURE OF FIFTH GENERATION COMPUTER SYSTEMS

Hidehiko	TANAKA	University of Tokyo
Makoto	AMAMIYA	Musashino Electrical Communication Laboratory, NTT
Yuzuru	TANAKA	Hokkaido University
Yoshihiko	KADOWAKI	HITACHI Ltd.
Masahiro	YAMAMOTO	Nippon Electric Co. Ltd.
Toshio	SHIMADA	Electrotechnical Laboratory
Yukio	SOHMA	FUJITSU Ltd.
Makoto	TAKIZAWA	Japan Information Processing Development Center
Noriyoshi	ITO	OKI Electric Co. Ltd.
Akikazu	TAKEUCHI	MITSUBISHI Electric Corp.
Masaru	KITSUREGAWA	University of Tokyo
Atsuhiro	GOTO	University of Tokyo

The summary of the preliminary research of data flow machines and database machines, which started in July 1981 and will end in February 1982, is shown in this paper. The objective of this research is to analize the study items of both machines and to show a detailed research plan which will be put into action from April 1982. At the same time, we hope that we can show some concrete architecture of both machines that can be the major components of basic hardware structure of the knowledge information processing system that is the final target of this FGCS project. The requirement specification of both experimental machines that should be developed during the first three years of the FGCS project is touched as well.

1. Introduction

The main objective of the FGCS is to realize the knowledge information processing systems(KIPS). The major components of KIPS are the inference engine and the knowledge base system. Although inference operation can be implemented on conventional von Neumann machines of sequential type, some excellent parallel execution mechanism is required to carry out inference operations of practical size with high speed. The data flow mechanism is one of the most attractive and feasible solutions for this problem. On the other hand, KIPS requires knowledge information of fairly large size. To realize a knowledge base system with such high performance, we need some sophisticated hardware support. The data base machine can be the powerfull means for it. Accordingly, we can conclude that the data flow machine and data base machine are the most promising candidates for the basic architecture of the KIPS.

The FGCS project will run for coming 10 years which can be divided roughly into 3 stages. The research of the data flow machine and data base machine follows the 3 stage-plan. Though we are going to develop at the final stage, practical machines in such sence as usefull for KIPS applications, this preliminary research is one of a preceding research that is going to supply some fundamental data for the coming research project of the first stage. At present, we are working to clarify the research themes to be investigated, and allocating them to the 3 stages. Then, we will select the technological alternatives and show some preliminary design specifications for the data flow machine and the data base machine.

The same kind of research planning work is running simultaneously at the architecture and software working groups. However, their work is to make up the overall plan of 10 years and to make clear the role of data flow machine and data base machine in the environment of the KIPS. Our research group concentrates our effort on the plan of the first stage, and is going to develop some hardware architectures which supports efficient execution of data flow operations or data base operations. How to apply these architectures to KIPS is not within our scope, but within theirs.

This paper shows the role of data flow machine and data base machines in some detail, which we expect for KIPS at this time. Following the discussion, the research plan and some reguirement specifications of both machines are explained with research themes.

2. The role of data flow machine and data base machine in the FGCS environment

As 'inference' is the fundamental operation in KIPS, the corresponding hardware should be of high level to shorten the semantic gap between hardware and software. 'Try and error' and non-deterministic operations are characteristic of inference machines. Accordingly, some powerfull speed-up mechanism through parallel hardware is indispensable for them. As the data flow concept is an excellent methodology to control parallel operations, we can imagine a inference machine of such architecture that the components are functionally distributed and whole system is controlled through data flow concept.

The nucleus of description system in KIPS is composed of the problem solving description language and the knowledge representation language. The problem solving description language can be designed in the logic programming style and/or functional style. As predicate calculus is a high level descriptive language which permits non-deterministic programming, the logic programming is suitable for some knowledge based processing. However, parallel processing is needed to execute with high speed the program written in logic programming languages because the execution of the language is rather complicated. It is shown that the programs written in logic programming languages are closely related to the data flow graph which is one of the machine languages of data flow machine. With functional programming, it is estimated to be very easy to build the program logic, because the program has no history sensitivity. Functional programming can be the execution mechanism for logic programming and relational algebra. On the other hand, the data flow machine is a natural execution mechanism of logic programming. Therefore, data flow machines can be the basis of inference machines.

Knowledge can be considered to be general data of unfixed form and of complex structure which are described by the knowledge representation language. The knowledge base system is required to be highly general so as to handle any kind of knowledge data. The leading candidate for the basic structure of this system is the data base system which supports relational model. The main component of data base system is the data base machine which supports relational algebra. Another language interface of relational data base is the relational calculus, which is closely related to logic programming and can be mapped easily to the corresponding statements of relational algebra. Although the knowledge representation itself is a very important problem to be investigated with the progress of FGCS project, the other distinct characteristic of knowledge base is that the volume of knowledge data can be very large. Knowledge is composed of individual facts, data which represent some procedures, and so on. So, the volume will become very large, and it can be a very difficult problem to maintain the storage efficiency and processing efficiency to a high level. One of the method to meet this situation is to separate the individual facts in knowledge base and to store them in the data base machine. Existing data bases can be also utilized through the knowledge base system in such organization. Accordingly, data base machine can be one of the important components of knowledge base system.

As mentioned above, data flow machine and data base machine are the basic architectures of the FGCS. These machines are closely related each other, as data flow concept can be used to realize the data base machine, and on the other hand, the structure memory which is a major component of data flow machine can be built using the data base machine, but we are sure that both will play the important two distinct roles of the KIPS as the inference machine and knowledge base system.

3. The research schedule of this year

Up to this time, the research work of this year is scheduled to take the following steps:

(1) Evaluation of existing data flow machines and data base machines.
(2) Extraction and categorization of research themes.
(3) Allocation of these themes among 3 stages of the project with the expected performance and required system size.
(4) Analysis of the themes allocated to the first stage.
(5) Requirement specification of data flow machine and data base machine of the first stage.
(6) Design of gross architectures of both machines by selecting a few candidates.
(7) Division of machines into several components and their conceptual design.

(8) Estimation of the performance of both machines.
(9) Clarification of mutual relation between data flow machine and data base machine.
(10) Software system design for the evaluation and test of the machines.
(11) Estimation of the manpower and the amount of money which will be needed to develope the experimental machines.

As for data flow machines, feasibility study is the main theme of the first stage of the project. So, what kinds of research theme should be listed up for it is considered at the step (2) through (5). We are going to design the first experimental data flow machine which supports functional programming. For data base machine, we have already many experimental machines in the world. Therefore the step (1) should be stressed to design the next generation data base machines, and the special characteristics which is required particularly for the KIPS applications should be clarified. We are going to design the first experimental data base machine which supports relational algebra as the basic operations.

4. Research of a data flow machine

4.1 Study items of data flow machines

Followings are the study items of data flow machines, that should be investigated prior to the real design. The items are grouped largely into 3 parts, hardware related items, software related items and system development related items.

[a] Hardware related items

(1) System configuration

First of all, the general structure of data flow machine should be clarified. What kinds of components should be identified and how they should be interconnected are the primary subjects. Up to this time, we identified five major components: activity memory, activity controller, processing elements, structure memory and interconnecting networks. The activity memory holds the data flow graph and the data which flow on the graph. In order to provide a very large memory space for this activity memory, we think that it will be necessary to have some 'virtual memory' mechanism and some 'cache memory' which holds the 'working set' of the data flow graph. The activity controller detects the fired activities, makes up the operations which are ready to be executed and sends them to processing elements. Processing elements receive the operations from the activity controlller, process them by accessing the structure memory and return the results to the activity memory. The structure memory holds the structure data and executes the basic operations for the structures. It shold be general enough to support all kinds of structures. As many numbers of processing elements access to the structure memory simultaneously, the memory should have very large bandwidth through dividing it into many memory banks. To make the structure sharing possible, some reference count mechanism or copying feature should be supported. The garbage controller will be one of the important components to realize the structure memory, as well.

We identified 4 kinds of interconnecting metworks. First is the arbitration network to connect the activity memory to processing elements. Second is the distribution network to connect the processing elements to the activity memory. Third is the mutual interconnecting network of processing elements. Fourth is the internal network of each processing element which will contain several operation units. These networks should be designed to make the data passing very rapid among many activity memories and structure memory. Fig.1 shows an example of a configuration of data flow machine.

(2) Control mechanism

There are two ways of activity control principle. One is the data-driven system and the other is the demand-driven system(Fig.2). To execute some compiled code of data flow languages, the former will be used. When some interpreter is used to interprete a program, the latter is used usually. At the first stage of the program interpretation using demand-driven control, requests of a operation(say 'A') is issued. At the second stage, some result data are returned from operation 'A' to the previous operation(say 'B'). As operation 'B' is drived by this resulted data, this stage of execution can be said to be data-driven.

Besides the principal method above, another efficient control mechanism such as firable node detection, lazy evaluation, processor allocation, and token labelling are required. The

detection mechanism of fired activities is very important to make the throughput very high. There are several alternatives such as associative memory method, hashing method and so on. We can introduce the concept of lazy evaluation to improve the parallelism by permitting the partial start of execution of functions before getting all the operands needed. The allocation mechanism of firable activities to processing elements is also important to keep the utilization level of processing elements high. One method is to use the unit of procedure-code as the allocation unit for each processing element. Accordingly, the tree structure of procedure call reflects on the loading pattern of program code. To control iterative, recursive and pipeline operations, tokens which flow on the data flow graph should be identified uniquely by some identifier (colored token). When tokens are dynamically generated, this token labelling can be serious overhead and may need a large name space for the identifire. Some hardware support will be necessary to cope with it.

(3) Basic operations

The basic operation set should be clarified as the machine language of data flow machine. It will contain system control instructions such as start/stop, structured data manipulation instructions, interative , recursive or conditional control instructions, and arithmetic and logical instructions. We think it had better to include a few instructions that are unification oriented for the purpose of logic programming. The representation format of activity is also a design item. Fig.3 shows two types of formats. a) shows a mixed format of program code and token data. b) shows a separate format in which token data is stored separately from program code. We can use the unit of procedure-processing as the unit of data-driven control. In this case, the execution of each procedure is left free, so that it can run also in sequential style of the conventional von Neumann machines and the procedure can be regarded as a basic operation.

All the items listed above is concerned with the internal structure of processing elements.

(4) Input/Output mechanism

One of the main feature of data flow machine is its fitness for the functional programming. Each operation can be described independently of other operations, and interacts each other only through the availability of the operand data. Input and Output operations may break this functionality. The concept of these operations had better be handled smartly preserving this functionality. For example, one means is to use the concept of 'stream'. Though file is a very important concept in conventional machines, it has never been treated in the research of data flow machines. Some hardware mechanism such as channel that handles the stream should also be considered for the processing elements, structure memory and activity memory.

(5) Implementation technology

The structure of data flow machines should have the full expandability in terms of its performance and capacity. As the machine of the stage II and III will be implemented by VLSI technology, the modulality of the structure should be considered from the beginning of its research. System reliability is also an indispensable item. Retry mechanism for data flow operations, error isolation mechanism, reallocation of activity and so on are the major study items for this theme.

[B] Software related items

(1) High level languages

To develop a data flow machine, we need some data flow oriented high level languages. These languages will be functional type, that has a great merit of making the programming easy in the sense that operations have no history sensitivity. But, this is not all that we should consider for the characteristics of programming languages. How the programming languages that is designed from the programmers point of view can be compiled into very efficient codes in terms of data flow graph is the other important theme which should be solved in future. This can be the key point that decides whether data flow machines can be general purpose systems of wide use or not. Some basic language constructs such as if-then-else, repeat-until and while-do have been investigated. But, other items such as scope rule of variables, separate compilation, task generation, data type, data abstraction and so on are left uninvestigated.

(2) Operating systems

What kinds of operating systems should be designed for data flow

machine is entirely unknown at present. Whether each function of operating system implemented on the present conventional machine should be implemented on the operating system for data flow machine or not, and how it can be implemented if it should be, will be clarified as the research progresses. Especially, some history sensitive processing will be required for file handling and data base utilization. Nondeterministic processing will also be requested, because nondeterminicity is essential for resource management that is a major function of operating system. What are the corresponding concepts to job, job abortion, process, interruption, multiprogramming, program status word, and so on ? How can all status information of the machine be dumped out at the check points for the recovery ? What makes the problem complicated seems to be the fact that many processing elements of the data flow machine run asynchronously. But, we think this problem is not so difficult essentially. For example, in order to stop the machine at some point, it is all to be done that the allocation controller of activity (even if there are many controllers, they can stop independently) is forced to stop.

(3) Activity allocation

There are two problems regarding activity allocation. One is the activity allocation to memory. The other is to processing elements. As we will have to divide the activity memory into many units primarily for reasons of throughput improvement, we will face the problem of allocation of data flow subgraph to each unit. This allocation strategy has influence upon the quantity of inter-module communication and the utilization factor of each processing element. Accordingly, this should be done so as to maximize the processing efficiency. A very simple algorithm is to handle each procedure as the allocation unit and allocate procedures called by a procedure to the near modules in the sense of communication overhead. This allocation is performed at the time of program loading, so it is static. On the other hand, allocation of activities to processing elements is dynamic and should be scheduled with the intra/inter communication overhead taken into account.

(4) Parallel algorithm

Data flow machine can realize the maximum parallelism expressed in the program. However, it cannot do beyond the maximum. This maximum depends on, not only the description style of the program, but the parallel algorithm itself. The former is related to the development of programming languages. The latter to the characteristics of the problem which is going to be described in some language. Inference applications are guessed to have very large parallelism. However, this precise analysis is left to the future research.

[C] Items for system development

(1) System description and simulation languages

To fix all design parameters and evaluate the performance of the experimental system, it is necessary to describe the system formally and simulate it with various parameters. Accordingly, we need some system description languages and simulation languages, which are suitable to describe asynchronous parallel operations.

(2) Debug and maintenance facility

We need some debugging aid to ease the debugging of the complicated asynchronous multiprocessor, such as hardware monitor and software conventions. Powerful service processor will also be needed to get the status of the experimental machine. The system maintenance method should be re-examined to match with the new architecture as well.

(3) Connection to other machines

At the first stage of the development, some host machine will be needed to supplement the function of the experimental data flow machine. Software development, compilation, linking, loading and so on will be executed on the host machine. So, input/output interfaces of the machine should be general to support these facility.

(4) Matching to conventional machines

When the computer system shifts from the conventional von Neumann machine to the new data flow machine, program mobility can be a very important factor to keep alive the enormous amount of software property. We will be requested to develop the practical way of this shifting.

4.2 Allocation of the items to 3 stages

As the research of data flow

machine is a system development, we cannot ignore any items listed above. However, all items cannot be solved simultaneously. We assigned to each stage the items which should be put stress to be investegated, as follows.

(1) first stage

The objective of the first stage is to show the basic feasibility of data flow machine and give the basic structure for it. All the items except the ones allocated to stage II and III should be investegated parallel. As for system configuration, the basic structure should be investegated at stage I, but the extensions such as virtual memory support is postponed to the stage II and later. Regarding control mechanism, we are going to develop hardware support mechanism for firable node detection, lazy evaluation, processor allocation, token labeling and so on. As for input/output mechanism, only the fundamental input/output interface is implemented on the machine and the full function will be developed at the stage II. Accordingly, the file concept will not be implemented during the stage I. The implementation technology of first stage is based on the present technology, though the modularity of system configuration should be considered for the later development. The high level language which will be used for stage I machine is based on the languages which had been proposed for data flow machine and functional programming. Only the minimum function will be implemented for the requirements of operating system. As for parallel algorithm, ever-continuing research efforts is needed. Software simulation will be needed to fix parameters at stage I, so we will develop some system description and simulation languages that are very good to describe asynchronous behaviour. The item of matching to conventional machines will not be necessary to be taken into account at the stage I.

(2) Second stage

The objective of the second stage is to test the implementation technology of VLSI and to develop the practical technologies of data flow machines. The hardware will be implemented by VLSI. Secondary memory concept will be realized in very natural way. The reliability issue and operating system design should also be investigated. As for high level languages, a few new languages will be developed for general purpose and for the inference processing. Personal data flow machine will be developed at stage II as well. On the stage II machine, various experiments will be done to test the effects of parallel execution of inference applications.

(3) Third stage

The real operating system will be implemented on the third stage machine. This machine will not only realize the connection between data base system and data flow machine, but be implemented with large scale to show the feasibility of large parallelism. This machine will be the basis of the KIPS.

4.3 Requirement specification of the stage I data flow machine

Followings are the requirement specification of stage I machine, that is set up at present.

- Universal functional unit with microprogram control (writable control storage).
- Hardware support for inference operations such as unification.
- Non-numerical processing oriented operation set such as character handling and unification for inference opration.
- 4 functional units for a processing element with an activity allocator.
- 16 processing elements per a system 4 sets of switching networks (16 x 16 packet switch).
- Activity memory of 16 banks, each of which capacity is 1 MB (16 MB in total).
- 4 M words of structure memory, with 10 ~ 16 bytes for a word that corresponds to an elementary cell for general structures.
- Parallel garbage collector for each structure memory bank.
- Data flow oriented high level language with the compiler.
- One processing element acts as an input/output controller.

5. Research of a data base machine

5.1 Study items of data base machine

(1) Specification of functions in data base machine and its host interface

We can design two types of data base machines. One is a stand-alone system that offers data base services to users who are connected locally or through communication lines. The

other is a host attached functional unit that is specialized in handling the data base system functions. Former can be thought to be made of two units, a host function handling unit and a data base function handling unit. This latter unit is the same as the host attached functional unit conceptually. We call this unit a data base machine. The data base machine should be equipped with a well defined interface so that the functions allocated to this machine are well isolated from the host and fully general to be used for any kind of applications. What kinds of functions of data base management system should be assigned to data base machine ? We can distinguish several functions such as data manipulation, data compression and encoding, secondary memory management, mass memory control, query analysis and optimization, integrity control, concurrency control, security control, and recovery control. As we have a plan to use this machine as the basis of knowledge base system, this functional separation is one of the important research items.

(2) Processing algorithm

The relational model that is based on a fine theoretical background is so general that any kinds of data can be expressed by the model. Accordingly, the data base machine that supports relational model is a good target for our objectives. Whether the navigational processing can be supported or not is a study item to be clarified in future, because other data model support may be required. Though it is also a study item to define a set of premitive operations, the first good candidate is the set oriented operations such as ones defined in relational algebra (union, intersection, defference, Cartesian product, selection, projection, join and division). The processing algorithm of these operations is a study item as query processing. However, so far as update operations are concerned, no good set operations have been proposed. When we want to update a shared database with real time, we are forced to face a very diffcult problem of concurrency control. We need some smart locking algorithm for it to preserve data base integrity. At the implementation of these operations, it will be necessary to manage the intermediate result relations that are produced under the execution of operations.

(3) Hardware architecture

Data base processing algorithms are realized by the corporation among software, firmware and hardware. Division of functions among these is an important disign item which trades performance and flexibility with cost. Hardware architecture is a mean to realize the parallelism of processing. We can introduce parallelism in the three forms. First is the parallelism of using many processing elements and memory modules that are connected each other through connecting circuits such as packet switches. Second is the internal parallelism of a processing element that can be made of many functional units such as sorter. Third is the internal parallelism of a memory module made of many memory cells and several functional units such as search modules. First one is closely related to the concurrent multi-user support and the total system throughput. Second one is the main factor to limit the speed of a single primitive operation, and is realized by parallel processing and pipeline processing among many units. Third one acts substantially to reduce the data volume that flows within the network of processing elements. The configuration of the hardware system and the control mechanism is an major study item of data base machine.

(4) Large capacity handling feature

Up to this time, there proposed many data base machine architectures. But, almost all of them don't support large capacity of data that exceeds several hudreds mega bytes. As the coming data base system will be of very large size, it is important to have a large capacity handling mechanism. Examples are the pagenation that supports working set concept for data base, and the staging network that loads necessary relations to working memory modules from mass memory such as mass storage system.

(5) Auxiliary functions

Multiuser support function that permits simultaneous execution of many user queries and/or updates is essential to improve the system throuput sufficiently enough to be used as the knowledge base system for inference machines. Though conventional data base systems can handle the formatted data only, it may be required to handle unformatted data for the data base system to be used as the knowledge base. Some encoding mechanism may be included in the architecture. Security and integrity

support functions are also important study items. As the mechanism to support these functions has not always been clarified perfectly, it will be required to investigate the functions of data base management system and to establish the methodology prior to the design of an experimental system of full scale (the machine of the stage II). These functions may require some hardware support mechanism.

(6) Distributed data base facility

The future data base systems will be connected each other by communication lines and make up a total integrated system. Accordingly, data base systems will not be stand-alone, but be mutually related. This distributed data base facility will be very important in future. The main function of the distributed data base is to integrate the many data base systems and to make up a large virtual data system that can be used as if it were a single system from users point of view. To achieve this objective, we can identify many difficult problems such as data model homogenization, realization of data-location-transparency, query processing that extends over many locations, concurrency control of update operations, commitment control against failures, and so on.

(7) Implementation technology

As the data base machine will be implemented by VLSI devices, the hardware structure had better be modular. For memory devices, we have RAM, magnetic bubble memory, CCD, magnetic disk, magnetic tape and optical disk at present. How to combine these devices and to make up memory hierarchy for data base processing is one of the design points. Device technology such as GaAs, HEMT and Josephson Junction device will also be related to this design.

(8) RASIS

As the system size grows, the problem of reliability becomes severer because of its great influence to the users in case of system failure. This is the same for data base machine. Because of its large capacity, it may require great overhead to collect the check point information and the update journal records. Moreover, the consistency maintenance among data will saddle us with very difficult problems especially for distributed data base systems. We need a smart methodology that enables us to limit the region of maintenance within a few systems for RASIS(Reliability, Availability, Serviceability, Integrity and Security).

5.2 Allocation of the items to 3 stages

We can divide the development steps of data base machines into 3 stages. At present, we allocate the each item listed above to the 3 stages as follows.

(1) First stage

The major objective of the first stage is to show the feasibility of a relational data base machine and to give the fundamental hardware architecture for it. We will take the relational algebra as the host interface, design the memory modules which are equipped internally with search mechanism and clarify the architecture of processing elements which contain some special functional units such as sorter. We should investigate the mechanism of large capacity handling feature through pagenation or staging from some mass memory to the working memory that corresponds to the secondary memory such as magnetic disk. We think that the working memory will be made of magnetic bubble memory or MOS RAM in place of magnetic disk. RAS support is also a distinguishable study item that should be treated all over the three stages. Though the distributed data base support is very important, its research will be executed independently of the data base machine development at this stage, because the reseach is essentially a software development. Multi-user support feature will also be considered and have influence upon the architecture design. Together with the development of an experimental machine, the research of data base system itself should be done by implementing many kinds of auxiliary functions on conventional computers to develop the methodology to maintain integrity and security at the multi-user environment. Fig.4 shows a general configuration of a data base machine.

(2) Second stage

A machine of the second stage will be designed to show the high performance of a large scale data base machine and be made using VLSI chips. A number of memory modules and processing elements will be connected through very high speed connecting networks. Handling mechanisms of

unformatted data will be designed and implemented on the machine. Various experiments will be done on this machine to test the technology of implementing knowledge base on the data base machine. The architecture of the machine will reflect the result of the software research of the first stage with regard to the data base management functions.

(3) Third stage

At stage III, the technology to realize high performance data base machine and to represent knowledge data on the data base machine will be joined to design a knowledge base oriented data base machine. The distributed data base technology will also be implemented for the stand-alone data base machine and grow as the distributed knowledge base system technology. At the time, the integrity maintenance technology will play an important role in consistency support of knowledge data. The machine of the stage III will be used to make up the knowledge information processing system as the fundamental mechanism of knowledge base system.

5.3 Requirement specification of the stage I data base machine

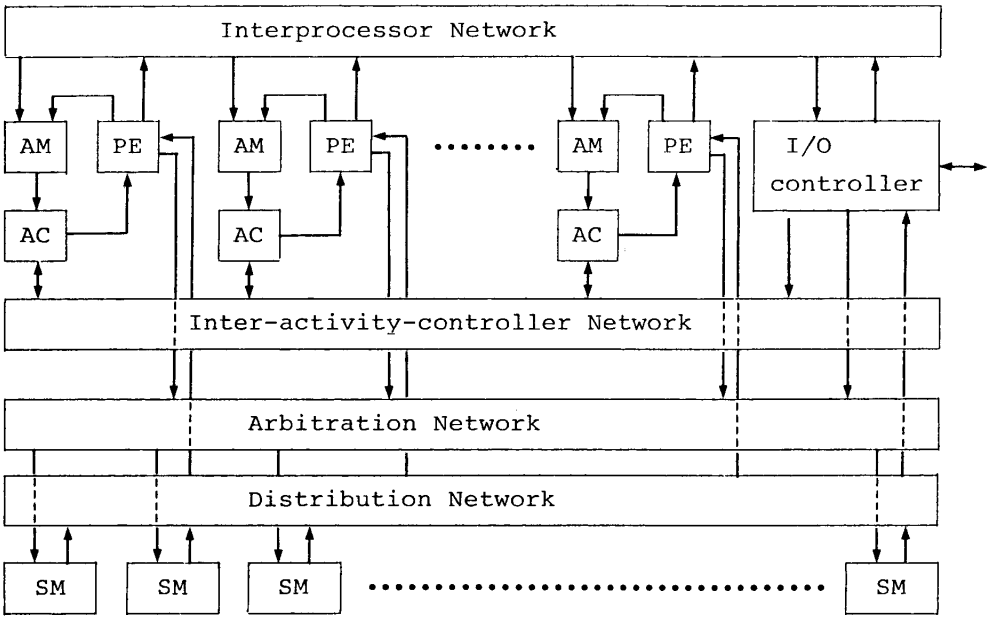
Followings are the requirement specification of the first experimental data base machine, that is imagined at present.

- Relational algebra language support as the host interface.
- Memory modules (working memory), each of which size is 4 MB and the number of modules is 32 (128 MB in total), with search and format transformation mechanism included.
- Processing elements: each element has a local memory of 256kB and special hardwares for set operations. 16 elements in total.
- 32X16 switching network: high speed packet switching network
- Mass memory staging mechanism that supports 20 GB of data with the data transfer rate of 50 MB/sec.
- Performance: 10 times faster than the processing on the conventional large scale machine.

6. Conclusion

The preliminary research of data flow machine and data base machine started in July of this year. As it passed only one and a half months since the beginning (this paper is written in August), the content of this paper is only the preliminary results of this 'preliminary research'. Accordingly, it includes some personal view regarding the schedule of the 10 years plan and the specification of the first experimental machines. It may be changed as the research progresses, and will be fixed at the end of this fiscal year (February 1982).

In this paper, we identified the study items of the data flow machine and the data base machine, and showed an example of the 10 years schedule in terms of the items. During the first three years, an experimental data flow machine will be developed so as to show the feasibility and the basic structure. Regarding the data base machine, an experimental system will also be developed to investigate the data operation mechanisms that allow great parallelism by expanding the set operations on many functional units. We hope that this preliminary research will give an explicit and concrete plan that will lead to a promising result.



AM : activity memory module
AC : activity controller
PE : processing element
SM : structure memory module

Fig.1 A configuration of a data flow machine

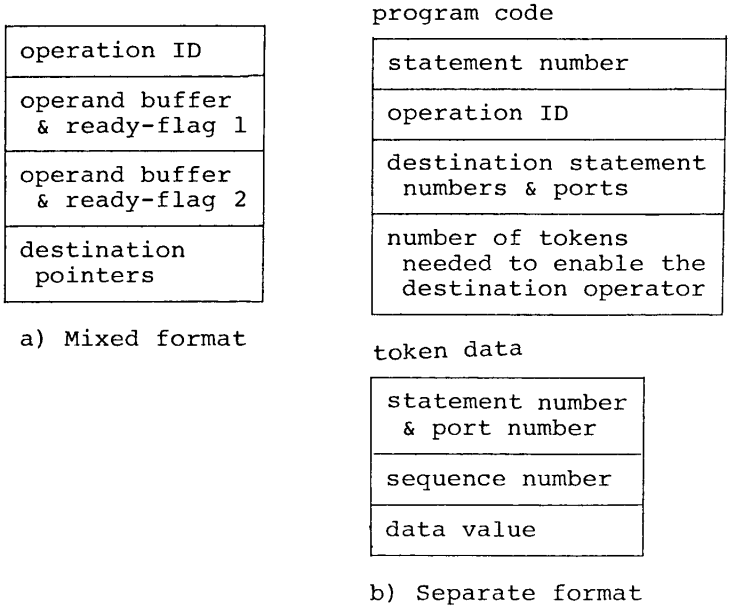
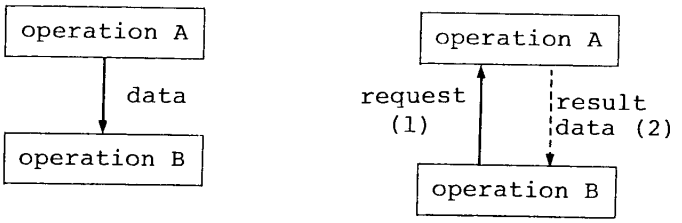
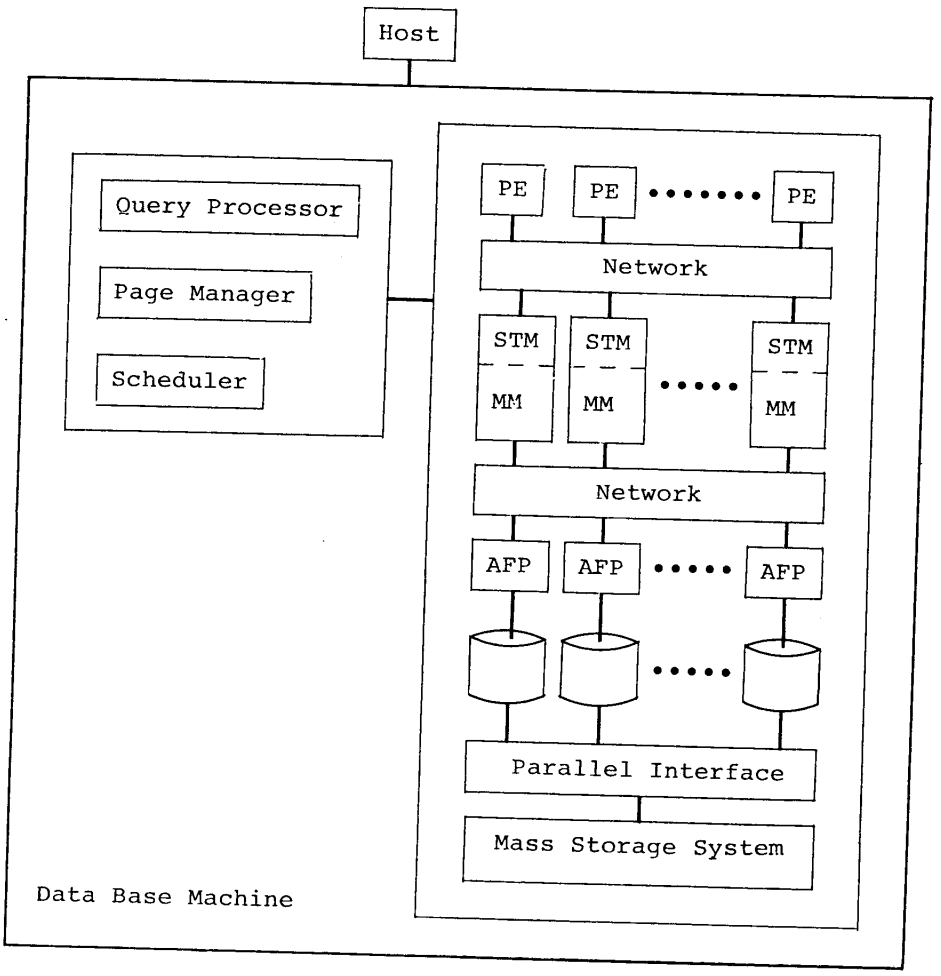


Fig.3 An example of the program and data format



a) Data-driven b) Demand-driven

Fig.2 The 2 ways of activity control principle



PE : processing element
STM : search & format transformation mechanism
MM : memory module
AFP : associative file processor

Fig.4 A configuration of a data base machine