Lawrence Berkeley National Laboratory
Recent Work

Title

SOME THOUGHTS ON THE DATA ACQUISITION PROBLEM OF HIGH-ENERGY PHYSICS EXPERIMENTS

Permalink

https://escholarship.org/uc/item/8676807g

Author

Kirsten, Frederick A.

Publication Date

1969-11-01

eScholarship.org Powered by the California Digital Library
University of California

For IEEE Transactions on Nuclear Science
Symposium, San Francisco, Ca.,
October 29, 30, 31, 1969

UCRL-19383
Preprint

eyf

SOME THOUGHTS ON THE DATA ACQUISITION PROBLEM OF HIGH-ENERGY PHYSICS EXPERIMENTS

RECEIVED
LAWRENCE
RADIATION LABORATORY
DEC 19 1969
LIBRARY AND
DOCUMENTS SECTION

Frederick A. Kirsten

November 1969

AEC Contract No. W-7405-eng-48

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

UCRL-19383

eyf

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

SOME THOUGHTS ON THE DATA ACQUISITION PROBLEM OF HIGH-ENERGY PHYSICS EXPERIMENTS

Frederick A. Kirsten

Lawrence Radiation Laboratory
University of California
Berkeley, California 94720

Summary

Some of the questions that arise when considering a standard data-bussing system are discussed. The discussion is restricted to applications in high-energy physics instrumentation. It is pointed out that it is necessary to provide for the operation of such systems without on-line computers.

Introduction

This paper contains comments and observations on a segment of the digital data handling problem of physics instrumentation. It concentrates particularly on the process of transferring data from digitizing devices (data sources) to the computer or recording devices. In this paper, that process is referred to as data acquisition--i.e., the acquisition of the digital data by a device that uses or records the data. The paper also concentrates on instrumentation for high-energy physics experiments. These experiments typically are characterized by having many digital data sources, each generating one or more words of data, and by requiring for each event the transfer of many words of data in a relatively fixed order. The experiments are also typified by relatively short lifetimes--of the order of a year or less--after which the system modules are separated and reassembled into another configuration for another experiment.

The acquisition of the data is facilited by use of a standard scheme for "data bussing." But, because of the short lifetimes, oversophistication exacts a penalty. The system should be as simple as is consistent with the required performance, in order to reduce the learning time, operator confusion, system testing time, and effort in design of operating procedures and controls.

The comments and observations are based in part on experience acquired through the design and use of a standard data-bussing system known as the NIDBUS system. It contains many features that are common to other commercial or laboratory-designed systems, and some which are, as far as I know, unique.

The recent appearance of the CAMAC system has placed us at a crossroad. We can continue in our present direction, or make a big change. The data-bussing systems used by various laboratories and commercial firms in the United States share the common feature of being mutually incompatible. Most of these existing data-bussing systems are firmly entrenched in their particular areas of use. No one of the existing systems, except CAMAC, has the necessary widespread acceptance and momentum to make it a practical candidate for alleviating the compatibility problem.

It is no secret, however, that the reception of CAMAC in the United States has been less than overwhelming. The major nuclear laboratories have agreed to use CAMAC, but so far its use has been on a very small scale. The usual attitude of manufacturers toward CAMAC has been, "Let's wait and see what sort of demand develops," while that of the user has been, "Let's wait and see how others make out." I think there are two reasons for this:

1. Recent lack of research dollars has made designers conservative, unwilling to risk part of the budget on new designs.

2. Lack of information in the United States on the relative advantages and disadvantages of standardizing on CAMAC.

Let me expand the latter point. There is no doubt that there are advantages to standardizing on a data transfer system. At present no two manufacturers of nuclear data acquisition equipment are using the same data-bussing standards. There is very little standardization between laboratories. This generally means that every time a new piece of equipment is acquired from outside one's own environment, an interfacing problem arises. Designing these interfaces is really "busy work." They add nothing to the power of the equipment but use up valuable man-hours. A widely adopted standard system could greatly reduce this problem.

However, the use of a standard inevitably involves some compromises. If it is a general-purpose standard, it will probably not be optimum for the special-purpose systems. The design of an individual module may be more expensive if it conforms to the standard than if it were required

simply to satisfy the module's intended function. The communication between modules may be more complicated and less efficient than in a more specialized system.

In attempting to evaluate the validity of such arguments, one finds CAMAC is probably its own worst enemy. The CAMAC specification is relatively complex. This not only complicates the problem of evaluating it, but each feature and specification provides the possibility of an objection being raised.

We must realistically face the fact that if a data-bussing standard is to be adopted in the United States, it will almost certainly have to be CAMAC. It exists and has reached a high state of development backed by a lot of thought and a certain amount of experience. To develop an alternate to the same degree as CAMAC already is developed, would take a minimum of a year and a half. This would cut seriously into the expected useful lifetime of such a system. Therefore, we should concern ourselves not with whether or not to adopt CAMAC but with determining in which areas the CAMAC philosophy can be exploited profitably. It is my belief that the exploitation will be most profitable if designers of CAMAC modules and systems will follow the CAMAC specifications as completely as possible. However, it should be realized that, even within the specification, there are some decisions affecting the usability and interchangeability of designs that should be made. For example:

1) Should we try to standardize on a common word size (which may or may not be 24 bits)? If so, what word size; if not, how do we cope with the problem of utilizing modules of differing word size?

Data Systems With and Without Computer Hardware and Software Control

Although one usually thinks of an on-line computer (wishfully?) in connection with a data-bussing system, it should be possible, and is often necessary, to operate without a stored-program computer. Some systems are originally intended to operate without a computer. Others that are intended to work with computers often have periods of time during which they do not or cannot. These include the periods in which the computer is out of service, or is being used for generation of programs, or is simply not yet available. We have found it extremely useful to be able to take some arbitrary part of a data acquisition system and connect it to a simple recording device such as a typewriter or incremental tape deck. As brought out in the following section, the characteristics of the data-bussing system influence the ease of this type of operation.

It should also be remembered that there are two modes of communication with a computer: a) program-controlled and b) via data channel. In the former, the computer program can control the transfer of each individual word. In the latter, the program initiates the transfer of a block of words which then continues under hardware control. Program control obviously offers the greater flexibility, but at the expense of time. The speed of data transfer is always higher in the data channel mode. When data transfers are controlled by a program (software) operating in a computer, the system can be said to be software-controlled. If the transfers are via data channel, or if a computer is not in use, we can refer to the system as being under hardware control. The features of the data-bussing system can be optimized for one type of control or the other.

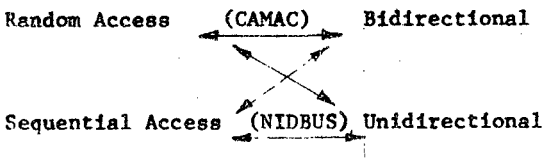
Random or Sequential Access

Table I indicates two more ways in which data-bussing systems can be categorized--by mode of accessing (addressing) the data, and by direction of data travel. The NIDBUS system is basically a sequential access, unidirectional system. CAMAC is basically a bidirectional, random-access system.

TABLE I

Classification of Data-Bussing Systems by:

a) Addressing Mode b) Directionality



The process of accessing or addressing a given data word consists of generating a signal--the word access signal in Fig. 1--that connects the appropriate data register onto a data bus. Two modes of generating the access signals are common in data acquisition systems--random access and sequential access.

In the random mode, any given data word can be accessed in a single cycle of the system. A complete cycle consists of two parts--address generation (in the controlling device) and decoding (in the data modules), and word transfer. This mode is well matched to software-controlled transfers to or from a computer. The program stores the valid addresses (locations of the data words) and issues them one at a time as each additional word is to be accessed. As shown in Fig. 2, address decoding is required for each

data word.

Two forms of address assignment are in use, each having its advantages and disadvantages. In the first, a unique device code is assigned to each module. The module responds to this regardless of its hardware location. The system designer need not specify in advance where each module will be physically located, but must assign addresses to the modules and insure that each module has a unique address. Physical locations may be changed without changing the response of the system. In the second form of assignment, modules are addressed by the location in the hardware--e.g., card-slot number. Locations must be assigned in advance, and cannot be changed unless the address is correspondingly changed. CAMAC combines these two forms. Each card slot of a CAMAC crate has an address, with the "slot pointers" radiating from the control slot. In addition, bussed sub-addresses permit addressing separate sections (or registers) on the selected card.

In sequential access systems, data words are accessible in fixed or semifixed sequence. Figure 3 shows how a shift register may be used as a "word pointer;" it points at or accesses each data word in turn. In the simple form shown, a single shift-register flip-flop constitutes the address "decoding" for each word. All bits of the shift register except the first are initially set to zero. Each Shift Command pulse causes the word pointer thus generated to be shifted along one notch. Because of its simplicity of control, this mode of addressing is especially well adapted to hardware control. The hardware need only generate a single pulse each time another data word is desired. There is no requirement for a memory to store the valid addresses.

The sequential access method requires the simpler addressing logic and requires less programming in that the individual address codes need not be generated. However, it does not permit the flexibility of the random access scheme. On the other hand, a random access system can be made sequential access if the addresses are generated in sequence. But, in general, not all addresses are represented by hardware. Unless a scheme is used for detecting invalid addresses automatically, some sort of memory device must be used to control the address generation. With CAMAC, the Q response signal provides the possibility for detecting the valid addresses, but the controller must be designed for this mode of operation.

The sequential mode has been important in high-energy physics instrumentation. Several commercial and laboratory systems now in use are based on it. Usually, each high-energy physics event generates many words (tens or hundreds of words) of data in a fixed or semifixed program of data sources. Thus the flexibility of software control is not necessary. Transfers via data channel simplify the programming and reduce the dead-time of the experiment.

Unidirectional and Bidirectional Systems

Another basis of classification is with regard to the possible directions of data travel, e.g., unidirectional or bidirectional. The bidirectional mode obviously provides more capability and is necessary for the general data handling problem. However, unidirectional (source to sink) is normally adequate for data acquisition in high-energy experiments, where sophisticated control of source modules is not (yet) utilized. The reduction in number of bus lines required is significant where modules are cabled together.

The remote control of data source modules is undoubtedly a feature that will be further developed in the future. This implies a need for a flow of digital information from a control center into the data source module--i.e., in the opposite direction from the flow of data. To date, our efforts in this direction have been based on a distinct separation between the system transporting the data words and the system carrying the control words. This is based on two arguments: 1) The resulting control system is simpler. The burden on the operator is reduced, particularly when the controlled data source module is being tested or operated apart from an on-line computer (an inevitable situation). 2) It seems natural to think separately about the control and data functions. In our typical data source modules, the registers that contain the data to be transferred are not the devices that are controlled. This is quite different from, say, a register in a magnetic tape unit which may receive data from, or send data to, a computer.

In our systems, the control words are stored in standardized registers located apart from the data source modules. A simple, manually operated console is used to setup or change the control parameters. The stored control bits are transmitted over cables to the source modules as dc levels. If, in a given instance, control is not needed, the cables are removed. The register modules so released can then be used to control other source modules.

The NIDBUS Solution

For the reasons developed above, the NIDBUS system was made to be unidirectional, and, although basically sequential mode addressing, also contains a rudimentary random access facility. The random access consists of a four bit program number code, which is maintained during the transfer of each "event." Data modules (which may generate many data words) are skipped or read out in response to the program number then in force. This permits a simple, semi-automatic means of controlling the transfer sets

of data. One set may be transferred for each event. Another set may be transferred once for each beam spill, and another for each experimental run. Under software (or hardware) control, the program number is set to determine which set of data is to be transferred. Then, once initiated, the transfer proceeds in sequential mode. Figure 4 shows a block diagram of this addressing scheme.

The Word-Size Problem

A real stumbling block in the path of designing the standard data-bussing system is the data word size problem. The problem is simply that there is no standard word size. For example, the commonly used or available computers have word sizes of 8, 12, 16, 18, or 24 bits. We find magnetic tape units with 6- or 8-bit characters. Typewriters utilize characters of 3 bits for octal, 4 bits for numerical BCD characters, or 6 bits for ASCII characters. On the other hand, devices that generate digital data in high-energy physics include magnetostrictive digitizing scalers of 12 and 14 bits, accumulating scalers of 24 bits and more, ADCs of 6, 8, or more bits, just to name a few. Throughtout the whole spectrum of data acquisition and data recording devices, we find no single, natural, multi-bit data-word size.

In software-controlled systems, the word-size problem is less serious than in hardware-controlled systems. The program can store in a table the size of every word along with its address. When the word is accessed, the program can initiate whatever processes (packing or unpacking) required to cope with the size of the data word. In hardware-controlled systems, one would prefer to have a simple, hardware-actuated means to cope with such problems, or, better yet, to standardize completely on the word size used. Obviously the size standardization is extremely difficult, if not impossible, on a national scale. It is even difficult on a laboratory scale.

The NIDBUS system has three features which help to alleviate the word-size problem. It uses two types of word-size signals and has a standardized procedure for packing small words into a single word. The first type of word signal is a single bit indicating whether the current data word is of 12-bit or 18-bit size. These two sizes came about because we were using exclusively 12- and 18-bit computers. This idea can be expanded to indicate a larger number of possible word sizes by increasing the number of bus lines assigned to carry the word-size information. The word-size signals are generated by the modules producing the data. They are interpreted by the recording devices. For example, a 6-bit tape unit unpacks a 12-bit word into two characters, or an 18-bit word into three characters. The typewriter unpacks a 12-bit word into four octal characters or an 18-bit word into six octal characters, etc. The module that interfaces such a system to a 12-bit computer data channel will automatically respond so as to cause a single memory storage cycle for each 12-bit word, and will automatically pack two 18-bit words into three 12-bit words in the computer's memory. In other words, the hardware-controlled devices can be designed to respond automatically and simply to the word-size indications.

The second NIDBUS word-size indicator permits handling pieces of data that occupy more than 18 bits. For example, an accumulating scaler of 7 decimal (BCD) digits generates a 28-bit datum. This will be transmitted via three 12-bit words or two 18-bit words. A 1-bit formatting signal indicates to the recording device that the two or three words are to be joined together to form a single datum. This can control the typewriter so as to automatically type the 7 decimal digits as a single data word.

A standard piece of NIDBUS hardware, the data-assembly box, is used to cope with odd sizes. Figure 5 shows a block diagram of this device. A standard connector is used to input data from "static" data sources. Static data sources do not have the wordpointer logic built into them. During the readout time, they present static data levels to the data-assembly box, which contains the necessary data gates and scanning logic. As shown in Fig. 5, the incoming data is connected to the data gates through "cross-connect" blocks. These consist of taper pin receptacles. Taper pin jumpers are used to connect the incoming data bits to the appropriate data gate. In this way, words may be packed, unpacked, or the bits unscrambled, as required in a given application.

The NIDBUS system also utilizes a third bit to indicate whether the data word is in binary BCD format. This information is generally not required for computer applications inasmuch as the computer program must already be aware of the format of the incoming data. However, a "stupid" device such as a typewriter control can automatically switch its mode from binary to BCD according to the state of this signal.

These three bits (word-size bits and binary/BCD bit) may be considered as descriptor bits of the data word. They must be transmitted and be available to the recording device at the same time as the data word. The CAMAC specification does not provide for any of these signals. However, it would not violate the specification to consider, as with NIDBUS, that the largest CAMAC word to be transmitted is 16 or 18 bits, and to dedicate the remaining bits of the 24-bit CAMAC word as descriptor bits. If this scheme is to be used, it needs to be agreed upon by all designers and users, otherwise compatibility is reduced. If such agreement cannot be reached, CAMAC designers must devise other means of solving the problems mentioned.

Noise Considerations

When one compares a NIM module and a CAMAC module, it is immediately apparent that there is

a distinct difference in the amount of noise protection provided. The NIM specification includes the requirement for a complete metal skin around the module. In most modules this includes a metal hood over the power connector, which engages a similar hood on the power distribution hardware. The CAMAC module design includes a much lower level of shielding capability. Complete shielding is difficult to accomplish with a card-edge connector at the rear.

The overall shielding of the NIM module tends to greatly reduce the susceptibility of the circuitry within from responding to external ground current noise as well as inductively or capacitively coupled noise signals. The NIM user can consider each NIM module as a well-shielded entity. He can connect cables, which may be part of a ground loop, to front or rear panels with relatively small danger of noise pickup by the circuits inside. The CAMAC user may need to exercise more caution. Since the modules are not individually as well shielded, he may need to take measures to insure that the rack that the CAMAC crate is mounted in be considered as providing the noise shielding. He may need to use extra grounding panels to intercept currents due to ground loops before they reach the CAMAC panels. This will be especially important in systems used with spark chambers.

Is Hardware Standardization Practicable?

The CAMAC specification covers both hardware and electronic aspects. It seems obvious that the electronic voltage levels, control signals, sequences, etc., need to be well-standardized. However, it is not so obvious that the hardware should be or can be so strictly controlled. Many configurations of source modules (and other classes of modules, as well) simply are not well adapted to the CAMAC design. Some devices require more shielding than CAMAC can offer. Some subsystems, such as magnetostrictive spark chamber digitizers, use large numbers of scaler cards of similar design. These can quite possibly be made more cheaply in other than CAMAC form. In the NIDBUS system, which has only electronic and cabling standards, we have ended up with hardware modules of all shapes and sizes.

Recently, the CAMAC committees have been working on the "vertical highway"-- the mechanism by which crate controllers are interconnected. I think it would be sensible to extend this idea by allowing "crate-controllers" to reside in non-CAMAC modules. The magnetostrictive digitizer mentioned above would be a good candidate for such treatment.

References

1) A description of the NIDBUS system has not been published but is available in: F. Kirsten, "The NIDBUS Data Bussing System," Engineering Note EET-1235, Lawrence Radiation Laboratory, Berkeley, California, Dec. 1967

2) The CAMAC system is described in many published papers such as: I. N. Hooton and R. C. M. Barnes, "A Standardized Data Highway for On-Line Computer Applications," AFIPS Conference Proc., Vol. 33, Pt II (Fall Joint Computer Conf., 1968) pp. 1077-1087, Thompson Book Co., Washington, D.C.

Captions

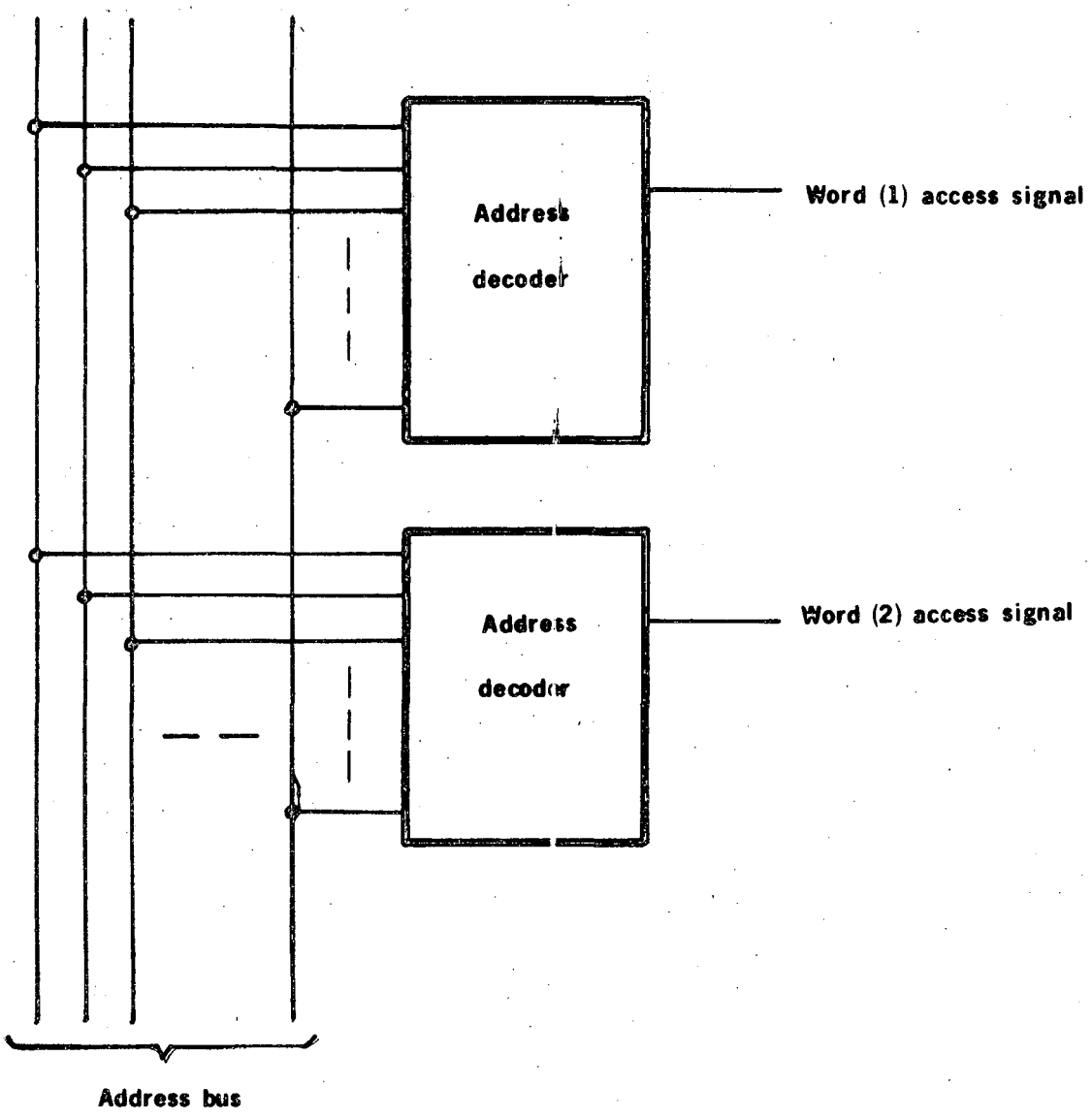
Figure 1. The basic logic involved in accessing a data word. The word is contained in the data register. Under control of the word access signal, the data gates connect the word onto the multi-bit data bus.

Figure 2. The address decoding scheme for the random addressing mode of operation.

Figure 3. A basic word accessing (addressing) logic scheme for the sequential accessing mode of operation.

Figure 4. This is the basic accessing logic used in the NIDBUS system. It combines the sequential access logic with a program-number control. The program number carried on the program number bus controls whether the module is read out or skipped.

Figure 5. A block diagram of a word assembly box. Data is received from "static" data sources on the left, packed, unpacked or scrambled in the cross-connect area, and connected onto the data bus through the data gates under control of the scan control logic.



XBL 6910-5932

Fig. 1

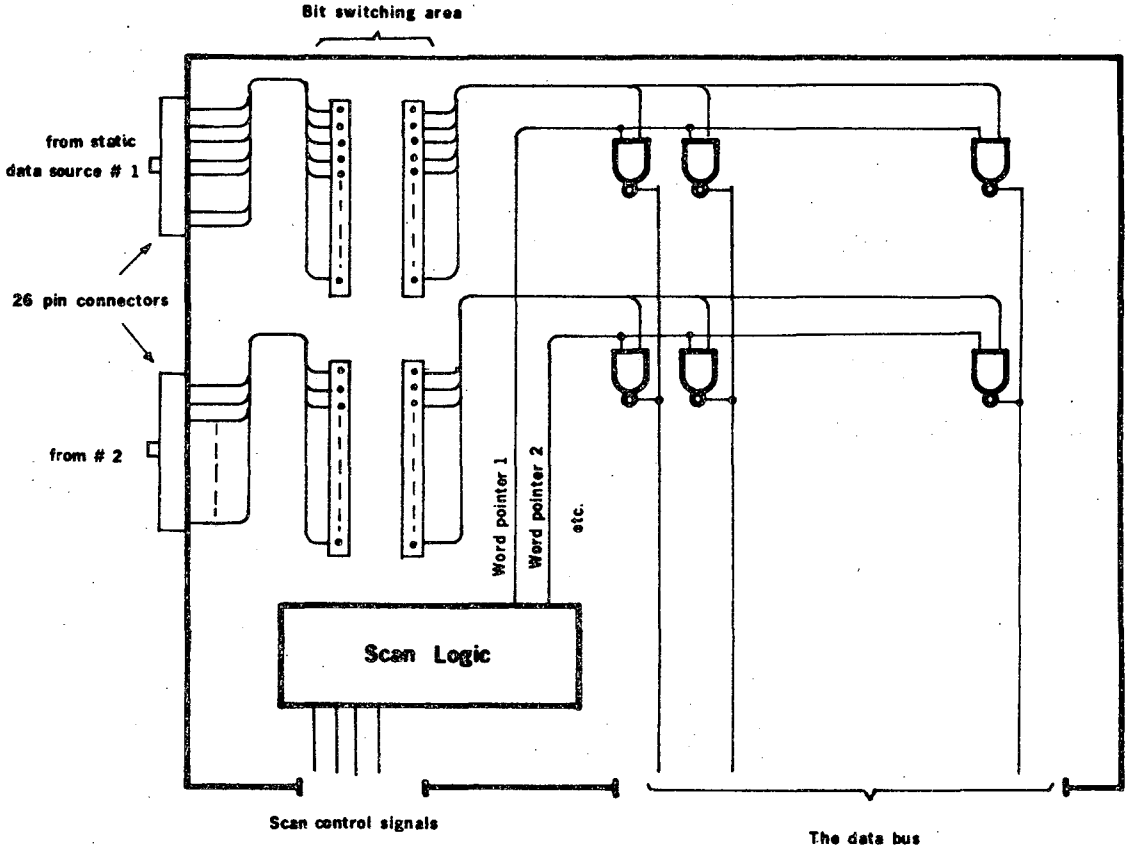
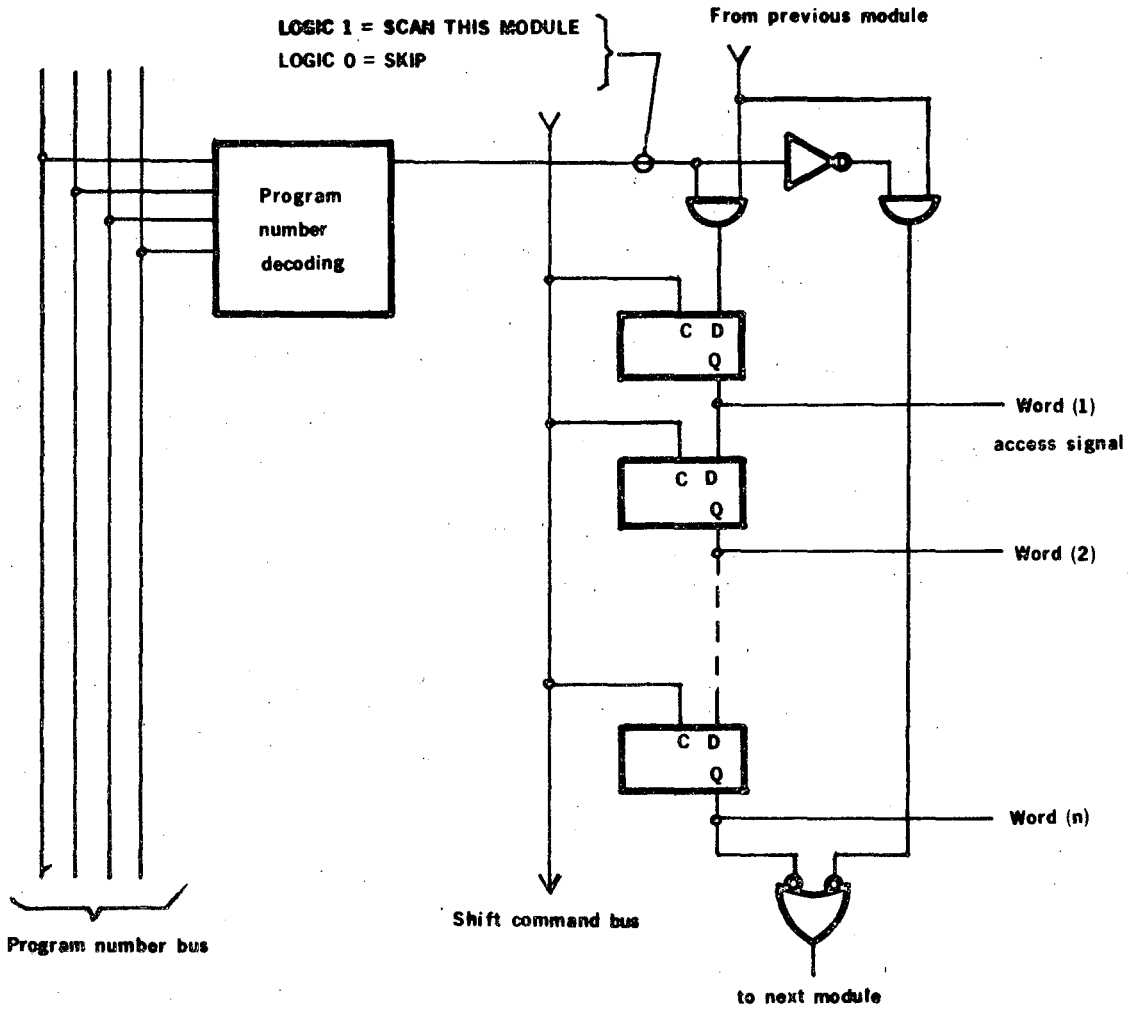

Bit switching area
from static
data source # 1
26 pin connectors
from # 2
Word pointer 1
Word pointer 2
etc.
Scan Logic
Scan control signals
The data bus
XBL 6910-5933

Fig. 2



XBL 6910-5934

Fig. 3

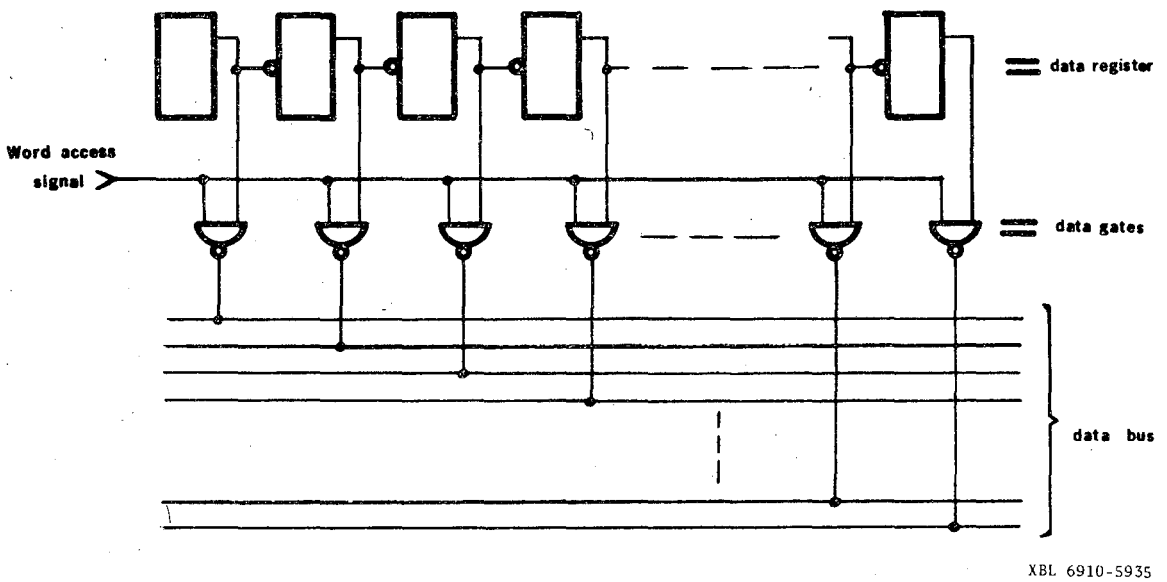
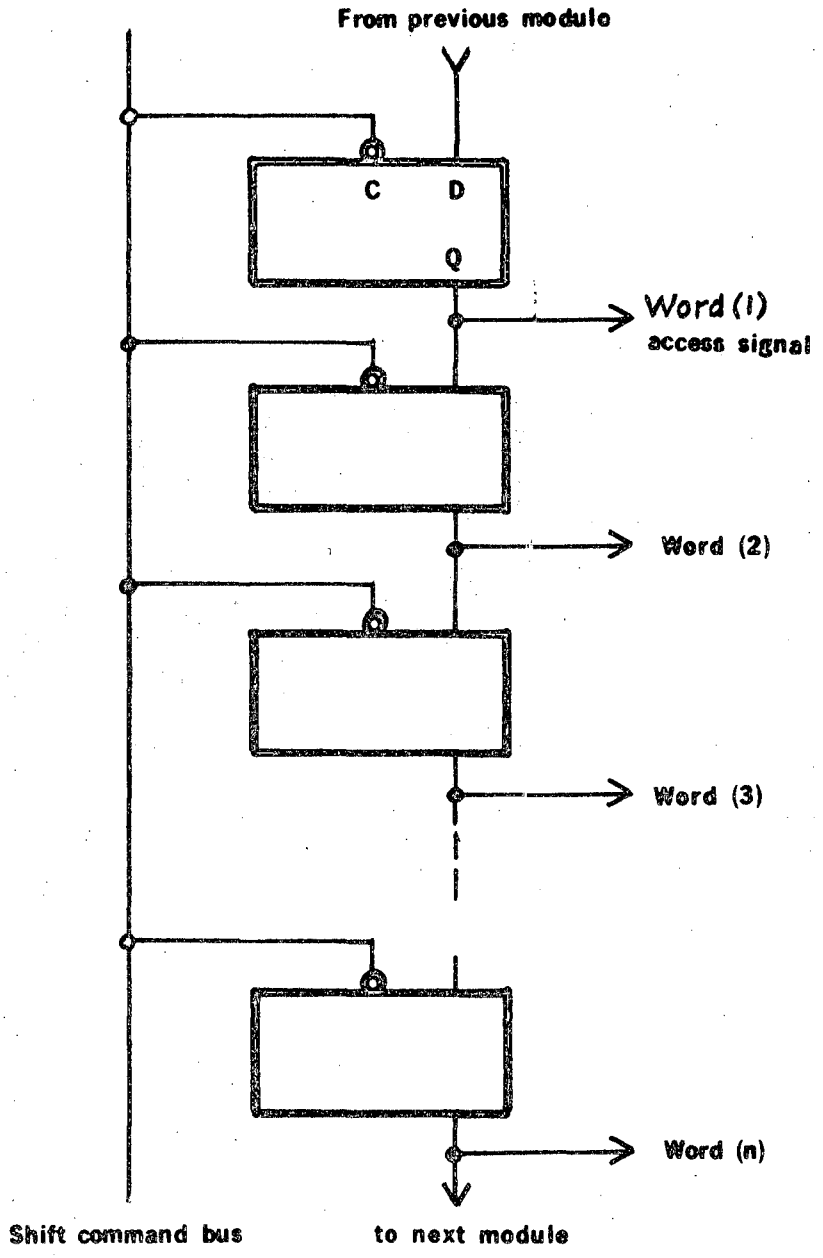

data register
Word access signal
data gates
data bus
XBL 6910-5935

Fig. 4



XBL 6910-5936

Fig. 5

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720