NO8600101

UNIVERSITY OF OSLO

INSTITUTE OF PHYSICS

REPORT SERIES

ISSN 0332-5571

A FAST ADC SCANNER FOR MULTIPARAMETER NUCLEAR PHYSICS EXPERIMENTS

G. Midttun, F. Ingebretsen, K. Holt, B. Skaali
Institute of Physics, University of Oslo
POB. 1048, Blindern, Oslo 3
Norway

OUP--
Report 83-15 April 5, 1983

A FAST ADC SCANNER FOR MULTIPARAMETER NUCLEAR PHYSICS EXPERIMENTS

G.Midttun, F.Ingebretsen, K.Holt, B.Skaali
Institute of Physics, University of Oslo
POB. 1048, Blindern, Oslo 3
Norway

A fast readout system for multiparameter experiments in nuclear physics is described. The central part of the CAMAC aqcuisition hardware is an ADC scanner module. The scanner incorporates a new arbitration logic and direct memory access for simultaneous transfer of singles and correlated data. Together with specially designed ADC interfaces the system can be set up for any configurations of singles and multiparameter events from 1 up to 15 ADC's in one crate.

CONTENTS

page

1. INTRODUCTION 1

2. HARDWARE CONFIGURATION 2

   System environment of the scanner 2
   Principles of operation 4
   Transfer of data 5
   Control of the computer memory data buffer 6
   Scanner registers 6

3. ARBITRATION AND ADDRESS LOGIC 7

   Arbitration of requests 7
   Calculation of memory address displacement 9
   The memory address pointer 10
   Performance of the address logic 11
   Error detection 12

4. SIMULTANEOUS TRANSFER OF SINGLES AND MULTIPARAMETER DATA 12

5. IMPLEMENTATION IN A DATA ACQUISITION SYSTEM 13

   The SHIVA data acquisition system 14
   SINTRAN-III monitor calls 14

REFERENCES 17

Appendix

page

A. TABLES 29

B. DETAILED DRAWINGS OF THE SCANNER 37

1. INTRODUCTION

Detector systems for nuclear physics experiments with large parameter sets or high data rates require fast readout and transmission hardware with data compression capabilities. In this report we describe a fast ADC scanner system developed for the Oslo University Cyclotron Laboratory. The system is characterized by :

- a very fast scanning logic based on a novel arbitration technique,
- variable event length with zero compression,
- automatic event cancelling by ADC time-out or external veto signal,
- possibility for concurrent singles and multiparameter acquisition.

The scanner is located in a normal station of a CAMAC crate (ref.1) using the CCN10 crate controller for NORD-10/100 computers (ref.2). The module utilizes the autonomous function controller bus (AFC) of the controller to achieve both DMA and CAMAC access.

The complete system consists of an ADC scanner, a set of ADC interfaces (ref.3), an external decision logic and a program system to supervise the modules. The control of the hardware system is incorporated in the program system SHIVA (ref.4). The decision logic determines which parameters that constitute an incoming event. This can be done by conventional gating logic; a flexible system is described in ref. 5. The events are defined by the external decision logic and the position of the ADC interfaces in the CAMAC crate.

The system described in this report is a further development of an ADC scanner system interfaced to a PDP7 computer (ref.5). The new system uses the same arbitration logic as the old system. The new de-

sign includes full compatibility with the CAMAC standard and an interface to a modern 16 bits mini computer. In addition, the new system is more user friendly and incorporates a novel method of controlling the computer memory data buffer.

2. HARDWARE CONFIGURATION

2.1 SYSTEM ENVIRONMENT OF THE SCANNER

The basic design of the scanner is independent of the actual CAMAC system used. However, two requirements must be fulfilled:

- The crate controller must be able to receive dataway cycle requests (ref.6) from other modules in the crate.
- A DMA bus interface to the dataway must be available.

In a NORD-10/100 CAMAC system, the control of autonomous dataway operations as well as DMA transfers are done via the LINK bus (ref.2). The LINK bus is available at the CCN10 crate controller front. The ADC scanner plus ADC interfaces and eventually other DMA or AFC controllers are physically daisy-chained by cables to this bus.

DMA transfers are initiated by a request to the crate controller for a DMA transfer CAMAC cycle (ref.2). During the granted DMA dataway cycle, data are transferred to or from the crate controller. The module that initiated the transfer can be the destination or the source of the data.

A module which is granted an AFC cycle (or DMA) can address other CAMAC modules via CAMAC NAF commands.

Fig. 1 shows a block diagram of the scanner module with the various interconnections in the system. The scanner module initiates a DMA transfer when the arbitration logic detects a request from an ADC interface. The arbitration circuit encodes the request to an address of the NAF file containing the NAF command to be executed in the granted DMA cycle. In this cycle the crate controller decodes the module address N of the selected interface (transferred over the LINK bus). The interface receive the AF part of the command from the scanner NAF file over the dataway. During the cycle the data are transferred to the crate controller. The memory address calculated by the scanner plus the data are transferred over the highway to the host computer.

The ADC interface modules must provide the following signals:

- EVENT READY: ADC busy status daisy-chained between all the modules in the setup.
- CLEAR EVENT: Daisy-chained timeout signal between the modules.
- REQUEST SCANNER: The ADC flag.

A general ADC interface that incorporates the requirements listed above has been developed and is documented in ref. 3. This interface can also operate as an autonomous function controller in parallel to scanner supervised multiparameter transfers. That is, the interface can transfer data on DMA to a host computer or locally in the crate to other CAMAC modules like a display unit, or add-one memories for direct histogramming tasks.

2.2 PRINCIPLES OF OPERATION

The present scanner and interfaces permits any configurations of singles and multiparameter events from 1 up to 15 ADC's in one crate. Multiparameter events can consist of any combination from 1 to N datawords, where N is the number of triggered ADC's. The data is transferred on a first come, first served basis to predefined locations in an event record according to the set of overlapping gate pulses generated by the external decision logic. Incomplete events due to ADC timeout will be cleared automatically. This clear facility can also be activated by external control logic. The clear event facility not only clears the connected ADC's; it also moves the data memory buffer pointers such that the next valid event overwrites the erroneous event record in the host computer memory.

The overlapping gate pulses are strobed in a 16 bits "ADC pattern register" and constitute the identification of the event in progress. The bit position in the ADC pattern register defines the actual ADC numbers. The most significant bit (bit 16) is always set to one for unique software identification of the pattern word. This identification word is normally transferred to the first location in the event record. The automatic transfer of the identification word must be used only when the incoming events have a variable number of parameters. The optional transfer is program selectable.

The basic operations and requirements of the scanner system are then:

1. The NAF register file is loaded with the NAF values of all possible ADC interfaces which can be included in the event.

2. The ADC's to be included in an event are defined by means of a set of overlapping gate pulses to the pattern register. This pattern can be any subset of the ADC's defined in the NAF re-

gister file. The content of the pattern register will be transferred to the identification word if enabled.

3. The end of conversion flags from the ADC's generate request scanner signals. Each signal will generate an AFC cycle of the scanner which causes the corresponding ADC to be read out by a DMA cycle to a predefined location in the event data record in the host computer.

4. Timeout from any of the ADC's will generate a clear event signal which will cancel the event.

2.3 TRANSFER OF DATA

The REQUEST SCANNER (ADC flags) are connected to the high speed scanning circuit, and data from a selected ADC is read and transferred immediately.

Data from the ADC's will be read in random sequence, but the DMA address logic of the scanner generates ordered addresses according to the ADC number and number of involved ADC's in the event. This is necessary since the parameter data words contain no identification. An example is shown in figure 3. The first event, that consists of parameters 1,3,4 and 6 is written into the computer memory in the format shown in the figure. Since parameter 2 is not part of the event, the parameter 3 is found next to parameter 1 and so forth. Details of the arbitration and address logic are given below.

2.4 CONTROL OF THE COMPUTER MEMORY DATA BUFFER

The 16 bits memory address register and the 12 bits word count register control the memory data buffer. When fixed length event records are transferred, the scanner will issue a CAMAC Look At Me (LAM) signal when the word count reaches zero. Since we can transfer variable event records, the scanner support buffer overflow in the data buffer in the host computer. This feature is selected by COST bit 4 as shown in table 2. In this mode two conditions must be fulfilled for a LAM to be issued. Firstly, the word count register must have reached zero. Secondly, when this happens the scanner generate a LAM when the on-going event record transfer terminates.

2.5 SCANNER REGISTERS

The NAF commands for the scanner are listed in table 1. The autonomous action of the scanner is controlled through the following registers:

- COST register: A 16 bits control and status register. The content and the meaning of the bits are given in table 2.

- NAF File: A 16 bits * 15 words register file loaded with the NAF commands for the ADC interfaces. Bits 13-0 contain the NAF command while bits 15 and 14 eneables/disables premature termination of the scanner operation on missing X or Q response from the addressed interface, cf. table 3.

- Word count register: The 12 bits register is loaded with the number of words in the DMA buffer in the host computer. The setting of the register is dependent on the mode of the buffer overflow status (COST bit 4). When all events have the same number of parameters, the word count register should be set to a multiple

of the number of parameters (COST bit 4 = 0). If the event data records consist of a variable number of parameters the word count register should be set to the total bufferlength minus the maximum number of correlated ADC's plus one. The number of locations the event record overflows is available in COST bit 3-0. The number is calculated relative to the location where the word count register reaches zero.

- Memory address register: The 16 bits register is loaded with the memory address of the first location in the DMA buffer in the host computer.

3. ARBITRATION AND ADDRESS LOGIC

3.1 ARBITRATION OF REQUESTS

The novel arbitration logic (ref.5) is based on a first come, first served priority scheme. The idea can be visualized as a query signal travelling around a circle that at different nodes are intercepted by the request signals from the ADC's (fig.2). At each node there will be taken one of two actions:

i) If the request signal and the query is present, generate a "selected" signal and stop the search.

ii) If the request is not present but the query from the preceeding node is present, submit the query to the next node.

Since the logic enters a "stop mode" after a selection, each node should have an entry which permits a generation of the query signal to the

next node when the selection has been recognised by the rest of the system. These requirements can be expressed in the following equations:

$$SLi = GLi \cdot Ei \qquad (1)$$

$$Ei+1 = (\text{not } GLi) \cdot Ei + Gi \qquad (2)$$

where SL is the select signal, GL is the request, E is the query signal and G is the generate query signal. If we interprete G as the carry generate and the GL as the propagate term, eq. 2 is similar to the carry propagation in a binary adder. Therefore a high speed scanner circuit can be build with carry look ahead generators. The condition in eq. 1 is realised with AND gates and the generate term is the feedback from a register which updates the selected signals.

Typical average settling time in the 74S182 carry look ahead circuit is 16.5 ns; worst case being 32.5 ns. By cascading the look ahead chips with the same "scanning technique", the settling time for the whole circuit will be logarithmic to the number of requests.

3.2 CALCULATION OF MEMORY ADDRESS DISPLACEMENT

The address logic circuit diagram is shown in fig. 4. The problem of calculating the memory address is twofold. Firstly, a pointer to the first vacant location in the memory where an event record can start, must be updated. Secondly, the displacement from this pointer to the actual address for a parameter must be calculated. This is necessary when we want to transfer the data as they come and achieve zero compression in the event data record.

Since the parameters are stored in ascending order, certain restrictions are put on the number of possible locations where a parameter can be stored. Relative to the identification word parameter 1 can only be stored in one location, i.e. the displacement for parameter 1 will be 1. For parameter 2 the displacement depends on whether or not parameter 1 is a part of the event. If parameter 1 is present the displacement for parameter 2 will be two, one otherwise. Relative to the address pointer updating the vacant address loacations, the displacements must be decreased with one, i.e. the pointer is already displaced with one.

Information on how many parameters constitute an event is contained in the ADC pattern register. The memory address can therefore be calculated in the following way: The memory address of the first vacant location MA(i) is known (start of the event record). The memory address of the location MA(k) for parameter k will then be:

$$MA(k) = MA(i) + \left[\sum_{j=1}^{k} B(j)\right] - 1$$

where B(j) is the j'th bitvalue of the ADC pattern register. The displacement is calculated in two steps. The first step is to extract the number of bits affecting the displacement from the ADC pattern register. This is done with a mask operation. The content of the ADC pattern register and a maskword containing a 1 in all bits from the least significant to the k-1'th bit are logically multiplied. The k-1'th bit position is equivalent to subtracting the 1 in the above equation. The mask word is then 0 for parameter 1, 1 for parameter 2, 3 for parameter 3 and 7 for parameter 4 and so forth. The resultant word contain a number of bits equal to the displacement for parametar k. These bits are summed together in a "N out of M" encoder (ref.7). This sum is then added to the pointer MA(i) to give the correct memory address.

A programmable read only memory contains the different mask words for the different parameter numbers (15 in all). The ADC numbers are the entries to this table. The maskwords are listed in table 6.

3.3 THE MEMORY ADDRESS POINTER

Updating the pointer MA(*i*) is also done in two steps. This part of the circuit contains a latch and a memory address counter. The latch and counter are initialized with the start address of the event data buffer. The latch serves as the pointer MA(*i*). Whenever a transfer takes place, the counter is incremented, and when the transfer of a complete event is ready, the content of the counter is strobed into the latch. The transfer is complete when the event ready status line is true.

This particular scheme makes it fairly easy to correct errors like timeout, missing X and Q responses, or sample a veto signal during accumulation. Whenever such a condition arises the strobe signal for transfer of the counter to the latch is inhibited, and the old content of the latch is clocked into the counter. The same scheme is also applied

to the word count register. In this way the event data buffer will always contain valid and compressed event records.

Due to the fact that the transfer of the identification word is program selectable, the memory address pointer must be adjusted to these possibilities. This is achieved by controlling the carry input to the adder circuit adding the pointer and the displacement, cf. table 6.

3.4 PERFORMANCE OF THE ADDRESS LOGIC

The detailed design of the memory address logic is given in the circuit diagrams in the appendix B. The circuit have several types of gates and medium scale integrated circuits in TTL logic. Some circuits are of high speed type (the arbitration) while others are of low power types to minimize power consumption.

The time used by the scanner module to calculate the memory address of a parameter depends on the position of the parameter in the event data record. The maximum delay in the logic calculated from the maximum values given in the data sheets of the various chips are 312 ns; the minimum 144 ns. These values are calculated from the request scanner input to the output of the memory address adder (the final address). In several test runs the mean average time is measured to be about 200 ns.

This high speed performance means that the scanner module contributes very litle to the total dead time in the acquisition system.

3.5 ERROR DETECTION

The scanner can be enabled to cause premature termination of the data accumulation, i.e. cause a LAM to be generated. Three error types can be detected.

COST bit 7 enables a timeout condition to generate a LAM. Bits 14 and 15 of the NAF register file controls the interpretation of the X and Q responses issued by the ADC interfaces during dataway cycles. Table 3 shows the conditions. The scanner interprets the responses as:

$$X'=(X+\overline{NAF14}) \text{ and } Q'=(Q+\overline{NAF15})$$

When the error detection is disabled, the X and Q responses are interpreted as true.

4. SIMULTANEOUS TRANSFER OF SINGLES AND MULTIPARAMETER DATA

The fact that the scanner and the ADC interface both have the hardware necessary for data transfers, an ADC can run in singles and coincidence mode simultaneously.

In multiparameter experiments, the singles data must be accompanied with a "singles" gate ORed with the coincidence gate. The singles gate and the coincidence gate originates usually from the same discriminator in the spectrometer. There are then two possibilities, either are the singles gate and the coincidence gate puls present at the same time, or

the singles gate occur alone. This is true since the coincidence also represents a singles event. There will be different delay in the electronic chain for the two gate pulses. The coincidence gate goes through the decision logic while the singles gate is used directly. This time skew must be compensated to avoid misinterpretation of the coincidence condition. The two gate pulses must be aligned such that the pulses overlap.

In the ADC interface module, the internal logic will clear the connected ADC only when all channel requests are executed. The timeout clock is only triggered by the ADC scanner controlled channel. When the singles gate causes a timeout no harm is done. The next singles or coincidence gate with true data will cause a transfer.

5. IMPLEMENTATION IN A DATA ACQUISITION SYSTEM

The large amount of raw data in multiparameter experiments usually requires data storage onto magnetic tape for later analysis. If time and available computer memory space permits, the data can be partially processed on-line. Hence a data aqcuisition system must control three data streams;

i) input from the CAMAC acquisition system,

ii) output to MT,

iii) output to the data processing (sorting) task.

The necessary means for synchronizing these parallel tasks must be implemented in the operating system. To avoid waiting time a double buffering scheme is required; the input stream must go to one buffer whilst the output streams are connected to the previously filled buffer.

5.1 THE SHIVA DATA ACQUISITION SYSTEM

The organization of the data streams is schematically shown in fig. 5.

The interrupt driver is activated by a LAM from the scanner module when a buffer has been filled. The driver switches the input stream to the other buffer (if free) and starts a SINTRAN-III direct task program on level 8. This direct task then starts the MT dump and event sorting RT programs if the corresponding output stream has been enabled.

The necessary communication and synchronization are achieved through monitor calls which have been included in the operating system SINTRAN-III. The overall control is performed by the SHIVA data acquisition program system, ref. 4.

5.2 SINTRAN-III MONITOR CALLS

```
MON 173 - DEFINE SCANNER PROGRAMS AND DATA BUFFERS

          SEGMENT CONTAINING BUFFER AREA MUST BE
          FIXC'ED BY MAIN CONTROL PROGRAM

CALLING PARAMETERS
      A = START ADDRESS OF BUFFER AREA
      D = WORD COUNT PARAMETER
          D(0-11)  = BUFFER LENGTH
          D(12-15) = WORD COUNT MODIFIER
                   = 0 ; DISABLE BUFFER OVERFLOW
                   > 0 ; ENABLE BUFFER OVERFLOW,
                         WCR:= BUFFERLENGTH - MODIFIER
      T = RT DESCRIPTION ADDRESS FOR MT TRANSFER PROGRAM,
        = 0 IF MT TRANSFER INHIBITED
      X = RT DESCRIPTION ADDRESS FOR SORTING PROGRAM,
        = 0 IF SORTING INHIBITED

SKIP  RETURN : A = UNCHANGED
ERROR RETURN : A = ERROR CODE (OCTAL)
               150 = ILLEGAL PARAMETER
               151 = SCANNER CONNECTED
```

```
MON 174 - ADC SCANNER CONTROL

    *** - NOTE BUFFER NUMBERING IS 1-2
    *** - SCANNER NAF FILE AND IDENT WORD CONTROL
          - MUST BE LOADED PRIOR TO CONNECT CALL

MODE= 0 : DEFINE CAMAC ADDRESSES AND INITIALIZE
      1 : BUFFER CONTROL FOR MT TRANSFER
      2 : BUFFER CONTROL FOR SORTING
      3 : DISCONNECT SCANNER
      4 : CONNECT SCANNER
      5 : FETCH STATUS


PAR.: T = MODE IN BITS 8-15

MODE  0 :
      A = NAL ADDRESS SCANNER
          (BITS 0-3   = GL, 0-17)
          (BITS 5-8   = SUBADDRESS)
          (BITS 9-13  = STATION)
          (BIT  14    = XERROR)
      D = CRATE NO


MODE  1 :
      A = BUFFER CONTROL PARAMETER
        = -1   : END OF RUN ACKNOWLEDGED
        =  0   : FETCH TRANSFER PARAMETERS
        = 1/2  : BUFFER TO RELEASE AFTER TRANSFER
      D = MT TRANSFER STATUS
        = 0    : OK
        = HARDWARE STATUS IF ERROR

MODE  2 :
      A = BUFFER CONTROL PARAMETER
        =  0   : FETCH SORTING PARAMETERS
        = 1/2  : BUFFER TO RELEASE AFTER SORT

MODE  3 : NONE
MODE  4 : NONE
MODE  5 : NONE
```

```
SKIP RETURN VALUES
MODE  0 - NONE
MODE  1 - A = BUFFER TO TRANSFER (1/2)
            = 0 ALL BUFFERS TRANSFERRED
            =-1 TERMINATION AFTER DISCONNECT
            =-2 MT ERROR DURING CONNECT NOT CLEARED
                          (BY DISCONNECT - CONNECT)
          T = BUFFER START ADDRESS
          D = BUFFER WORD LENGTH
MODE  2 - A = BUFFER TO SORTING (1/2)
            = 0 ALL BUFFERS SORTED
            =-1 TERMINATION AFTER DISCONNECT OR ERROR
          T = BUFFER START ADDRESS
          D = BUFFER WORD LENGTH
MODE  3 - A = SCANNER STATUS WORD BITS 0-7
MODE  4 - A = SCANNER STATUS WORD BITS 0-7
MODE  5 - A = SCANNER STATUS WORD BITS 0-7
          T = NO OF BUFFER TRANSFERRED


ERROR RETURN : A = ERROR CODE (OCTAL)
               150 = BAD PARAMETER(S)
               151 = SCANNER CONNECTED        ,MODE          4
               152 = SCANNER NOT CONNECTED    ,MODE   1 2 3
               153 = SCANNER NOT INITIALIZED  ,MODE          4
               155 = RESOURCES NOT DEFINED    ,MODE          4
```

REFERENCES

[1] CAMAC: A modular instrumentation system for data handling, EUR 4100e (1972)

[2] CAMAC-CC-NORD-10 general information, ND12.007, CAMAC-CC-NORD-10 Hardware, ND12.006, CDMA-NORD-10 general information, ND12.004

[3] G.Midttun, K.Holt, F.Ingebretsen, B.Skaali: A general ADC-CAMAC Interface Module, Institute of Physics, University of Oslo, report 82-10 (1982)

[4] B.Skaali: SHIVA reference manual, Institute of Physics, University of Oslo, report to be published

[5] G.Midttun, F.Ingebretsen, P.J.Johnsen: A Multi-parameter CAMAC compatible ADC Scanner, Nuclear Instruments and Methodes, 159 (1979) p. 567-574

[6] CAMAC: Multiple controllers in a CAMAC crate, EUR 6500e (1978)

[7] Designers manual System 74, TEXAS Instr. 1973, p. 116

Figure capture

Figure 1: Block diagram of the ADC Scanner with various interconnections.

Figure 2: Scanning principle; the arbitration circle.

Figure 3: Data structure; an example of event record layout.

Figure 4: Block diagram of the address logic; the memory address latch and counter register and the displacement mask operation.

Figure 5: SHIVA program system; block diagram of the program structure and data stream control.

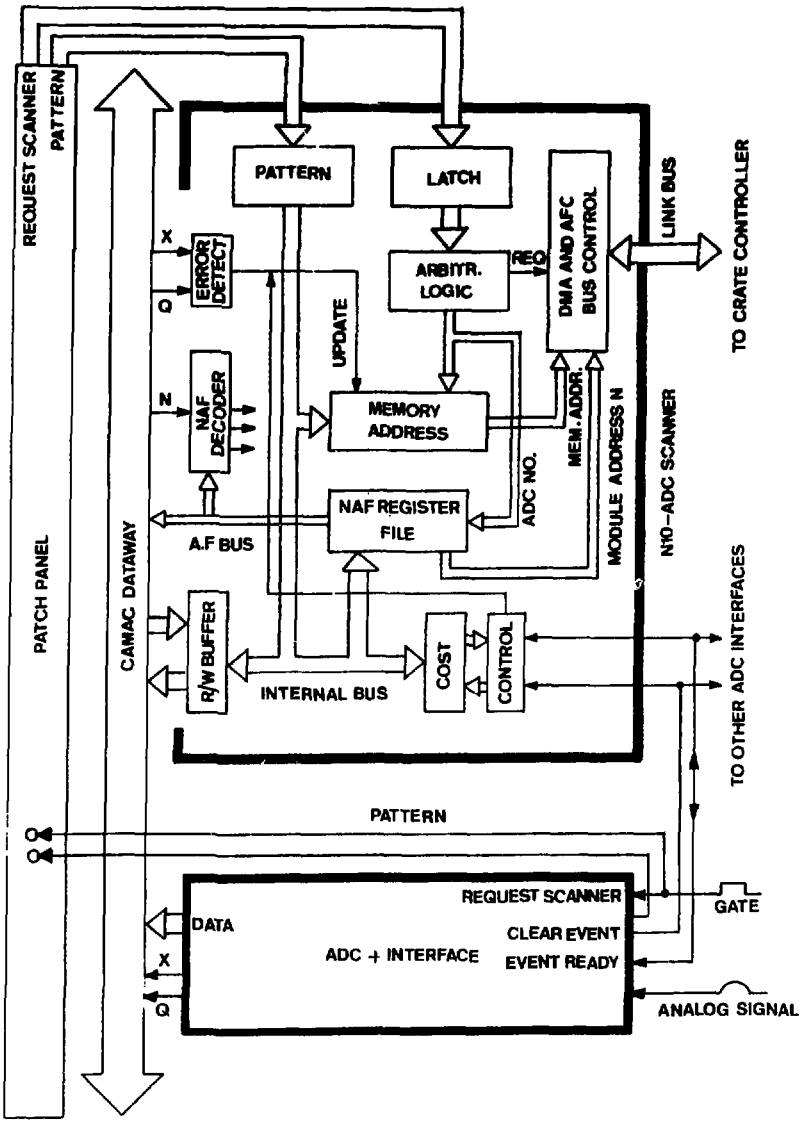

Fig. 1: Block diagram of the ADC Scanner

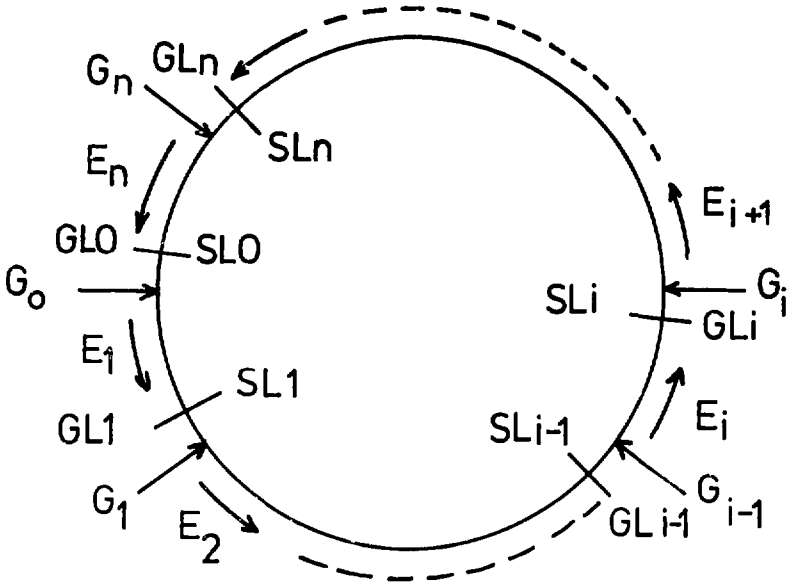

Fig. 2: Scanning principle; the arbitration circle

IDENT FLAG	BIT CODED ADC NUMBERS OR DATAWORDS	
BIT	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	
1	0 0 0 0 0 0 0 0 0 1 0 1 1 0 1	IDENT EVENT(I)
0	DATAWORD PARAMETER 1	
0	DATAWORD PARAMETER 3	
0	DATAWORD PARAMETER 4	
0	DATAWORD PARAMETER 6	
1	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	IDENT EVENT(I+1)
0	DATAWORD PARAMETER 2	
1	0 0 0 0 0 0 0 0 0 1 0 0 0 1 1	IDENT EVENT(I+2)
0	DATAWORD PARAMETER 1	
0	DATAWORD PARAMETER 2	
0	DATAWORD PARAMETER 6	

ADC SCANNER - EVENT DATA STRUCTURE

THE INCLUSION OF IDENT WORDS IS PROGRAM SELECTABLE. AN IDENT WORD IS NOT REQUIRED IF ALL EVENTS HAVE THE SAME NUMBER OF PARAMETERS.

Fig. 3: Data structure, an example of event record layout

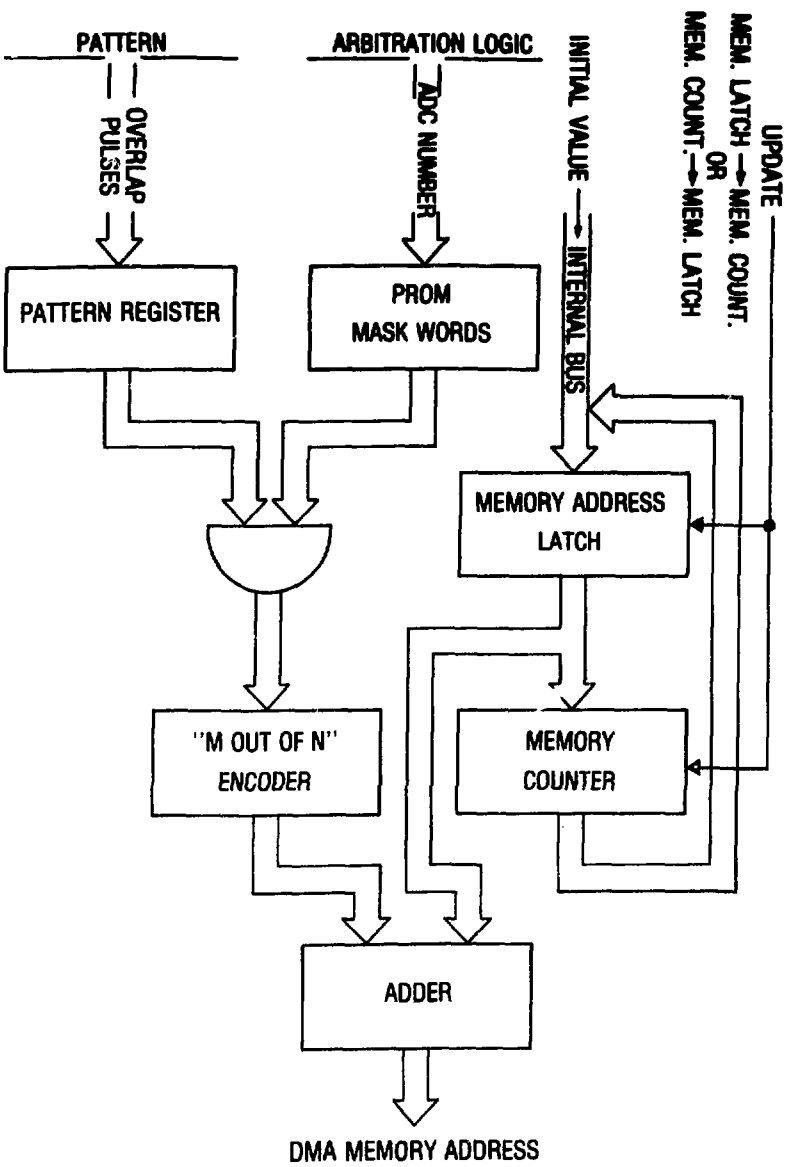

Fig. 4: Block diagram of the address logic, the memory address latch and counter register, and the displacement mask operation

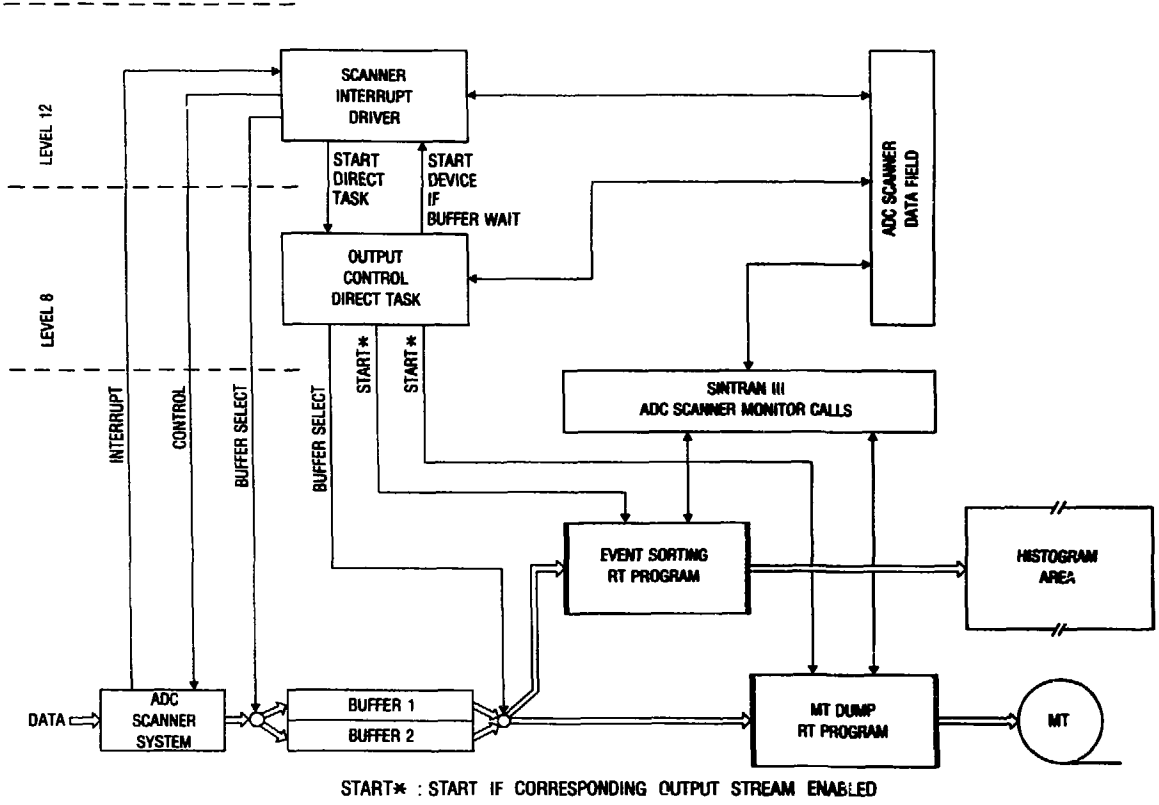

Fig. 5: SHIVA program system, block diagram of the program structure and data stream control

Appendix A

TABLES

Table 1: NAF commands for the scanner

Table 2: COST register

Table 3: Interpretation of the X and Q responses

Table 4: Content of NAF decoder PROM 74S472

Table 5: Content of NAF decoder PROM 74S288

Table 6: MASK PROM table

Table 1: NAF commands for the Scanner

Function	Action
F(0)A(0)	Read word count register
F(0)A(1)	Read memory address register
F(0)A(2)	Read control and status register
F(0)A(3)	Read ADC pattern register
F(1)A(1-15)	Read NAF File
F(8)A(0)	Test LAM, Q=1 if LAM
F(10)A(0)	Clear LAM
F(16)A(0)	Write word count register
F(16)A(1)	Write memory address register
F(17)A(1-15)	Write NAF File
F(24)A(0)	Disable LAM
F(24)A(1)	Disable COST bit 4, buffer overflow
F(24)A(2)	Disable COST bit 6, transfer of ident. word
F(24)A(3)	Disable COST bit 7, timeout interrupt
F(25)A(0)	Execute DMA request, test purpose
F(26)A(0)	Enable LAM
F(26)A(1)	Enable COST bit 4, buffer overflow
F(26)A(2)	Enable COST bit 6, transfer of ident. word
F(26)A(3)	Enable COST bit 7, timeout interrupt
F(27)A(0)	Test COST bit 9, Q=1 if ready
F(27)A(1)	Test COST bit 14, Q=1 if error

Table 2: COST register

Bit 0-3	:	Contains the number of locations the data buffer has overflown.
Bit 4	:	Enables buffer overflow.
Bit 5	:	Indicate that an overflow has occured.
Bit 6	:	Enable transfer of the pattern register to the top of the event records.
Bit 7	:	Enable timeout to generate LAM.
Bit 8	:	Indicate that a timeout has occured.
Bit 9	:	Indicate that the module is ready to operate.
Bit 10	:	Enable LAM.
Bit 11	:	Indicate LAM status
Bit 12	:	X response of last DMA cycle.
Bit 13	:	Q response of last DMA cycle.
Bit 14	:	Indicate X or Q error.
Bit 15	:	Indicate normal termination of the scanner.

Table 3: Interpretation of the X and Q responses

X' and Q'	Comments
X'=Q'=1	Data valid, transfer granted.
X'=1,Q'=0	Command accepted, data invalid.
X'=0,Q'=1	No practical situation.
X'=Q'=0	Data and command invalid.

$X'=(X+\overline{NAF14})$ and $Q'=(Q+\overline{NAF15})$

Table 4: Content of NAF decoder PROM 74S472

Function	Address HEX	Output HEX
F16A0	1FB	00
F16A1	1EB	10
F17A1	1E9	08
F17A2	1D9	08
F17A3	1C9	08
F17A4	0F9	08
F17A5	0E9	08
F17A6	*0D9*	*08*
F17A7	0C9	08
F17A8	1F8	08
F17A9	1E8	08
F17A10	1D8	08
F17A11	1C8	08
F17A12	0F8	08
F17A13	0E8	08
F17A14	0D8	08
F17A15	0C8	08
F10A0	1F3	99
F24A0	17B	85
F24A1	16B	95
F24A2	15B	8D
F24A3	14B	9D
F26A0	173	83
F26A1	163	93
F26A2	153	8B
F26A3	143	9B
F25A0	179	87
F8A0	17F	AA
F0A0	1FF	38
F0A1	1EF	24
F0A2	1DF	34

Table 4 cont.

F0A3	1CF	2C
F1A1	1ED	3C
F1A2	1DD	3C
F1A3	1CD	3C
F1A4	0FD	3C
F1A5	0ED	3C
F1A6	0DD	3C
F1A7	0CD	3C
F1A8	1FC	3C
F1A9	1EC	3C
F1A10	1DC	3C
F1A11	1CC	3C
F1A12	0FC	3C
F1A13	0EC	3C
F1A14	0DC	3C
F1A15	0CC	3C
F27A0	171	A2
F27A1	161	B2
DEFAULT		FF

Table 5: Content of NAF decoder PROM 74S288

Function	Address HEX	Output HEX
FA6A0	0	E0
F16A1	1	E1
F17A1-A15	2	AF
F0A0	3	D2
F0A1	4	D3
F0A2	5	F4
F0A3	6	D5
F1A1-A15	7	96
F27A0	8	F7
F27A1	9	F8
F8A0	A	7F
DEFAULT		FF

Table 6: MASK PROM table

Address HEX		Maskvalue HEX
0	Ident. word	0000
1	Parameter 1	8000
2	Parameter 2	8001
3	Parameter 3	8003
4	Parameter 4	8007
5	Parameter 5	800F
6	parameter 6	801F
7	Parameter 7	803F
8	Parameter 8	807F
9	Parameter 9	80FF
A	Parameter 10	81FF
B	Parameter 11	83FF
C	Parameter 12	87FF
D	Parameter 13	8FFF
E	Parameter 14	9FFF
F	Parameter 15	BFFF

The entry of the mask PROM are the ADC numbers (parameter numbers). The most significant bit of the PROM output controls the carry input of the memory address adder. When the pattern register transfer option is enabled this PROM bit is the carry. For the identification word the bit is zero, otherwise 1. The request for identification word transfer is treated the same way the ADC transfer requests are handled by the scanner logic; hence this "parameter" has the number 0 and the given mask value.

Appendix B

DETAILED DRAWINGS OF THE SCANNER

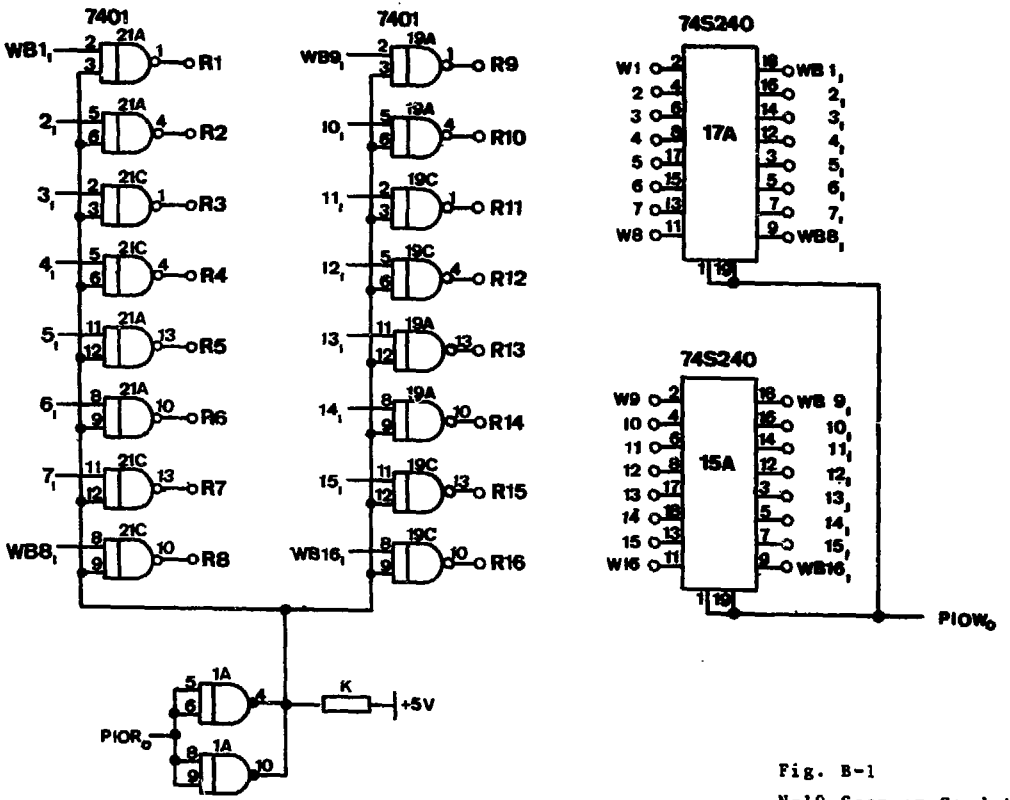

Fig. B-1
N-10 Scanner Card A
Read/Write

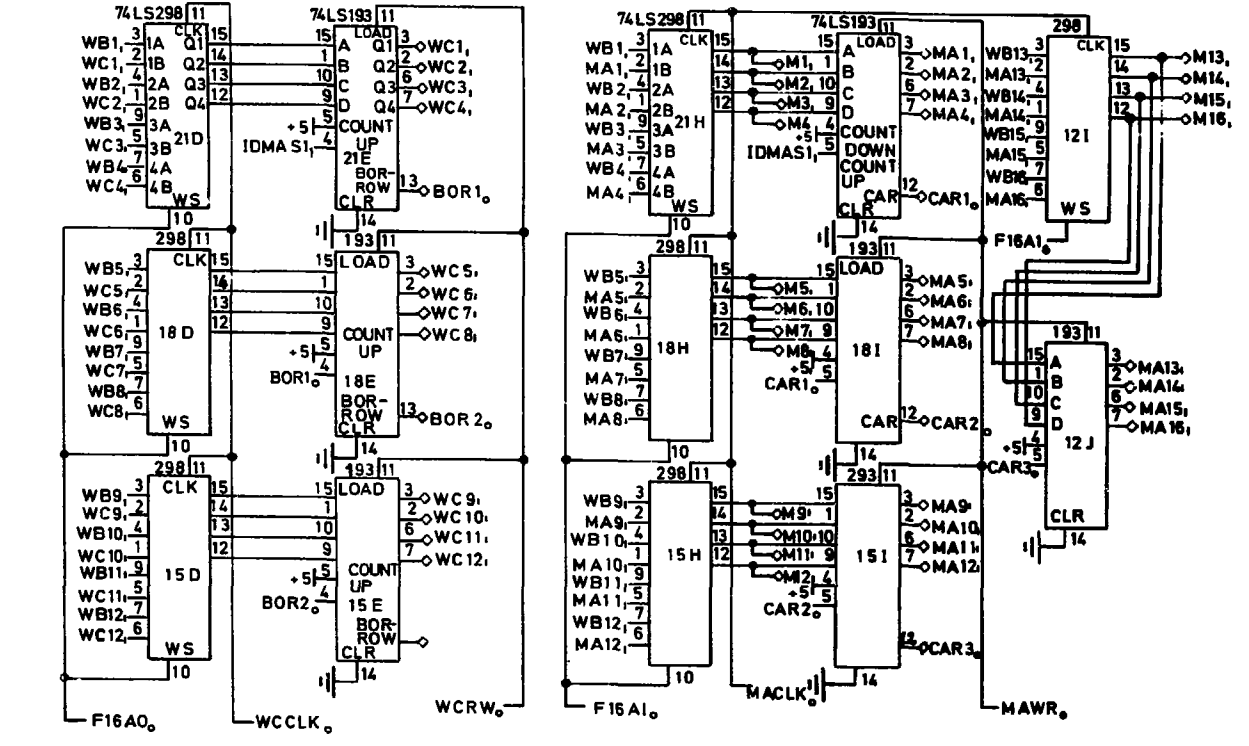

Fig. B-2
N-10 Scanner Card A
Memory Address + Word Count

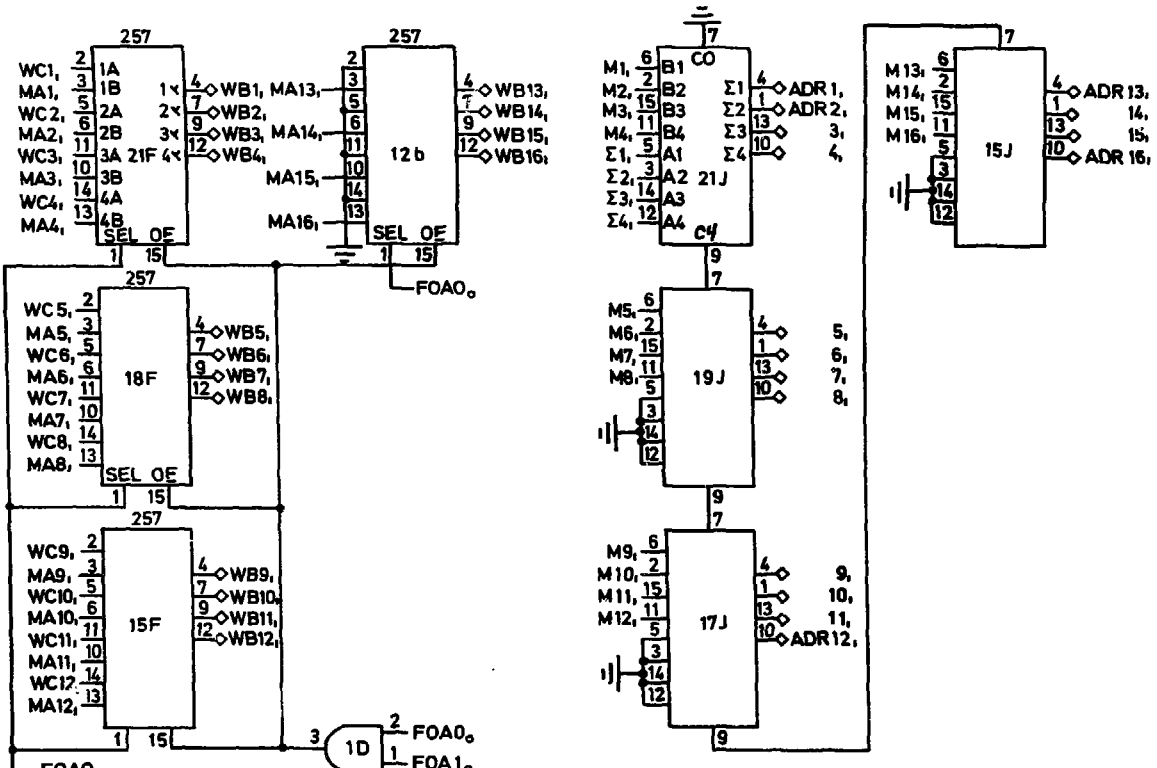

Fig. B-3

N-10 Scanner Card A

Memory Address/Word Count MPX (Read) + Memory Address Adder

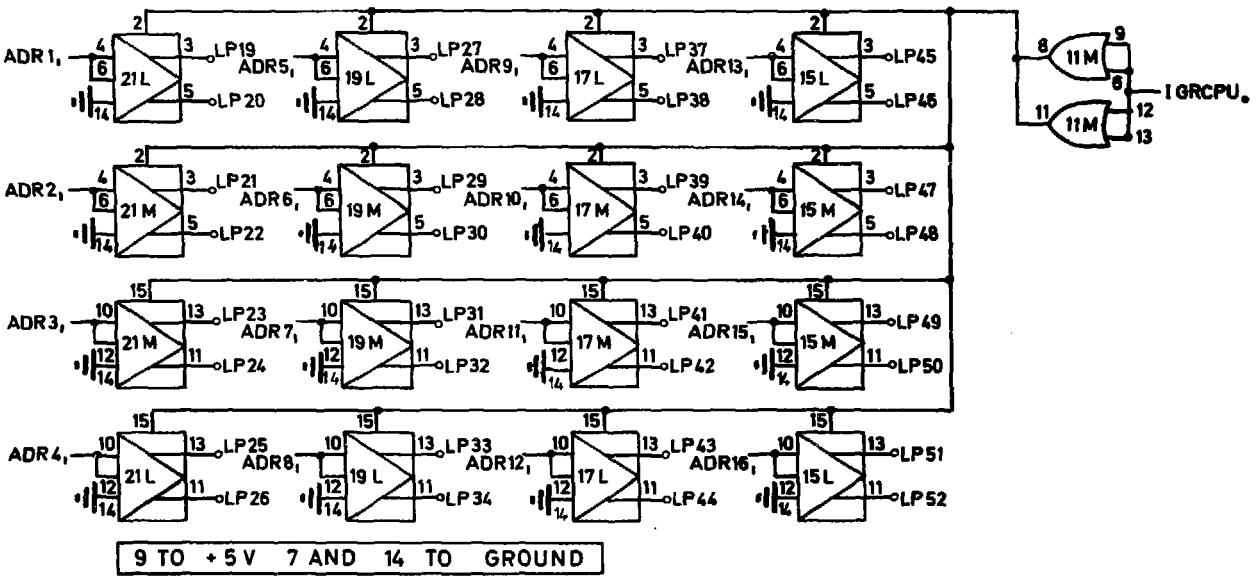

Fig. B-4
N-10 Scanner Card A
Diff. Line Drivers Memory Address

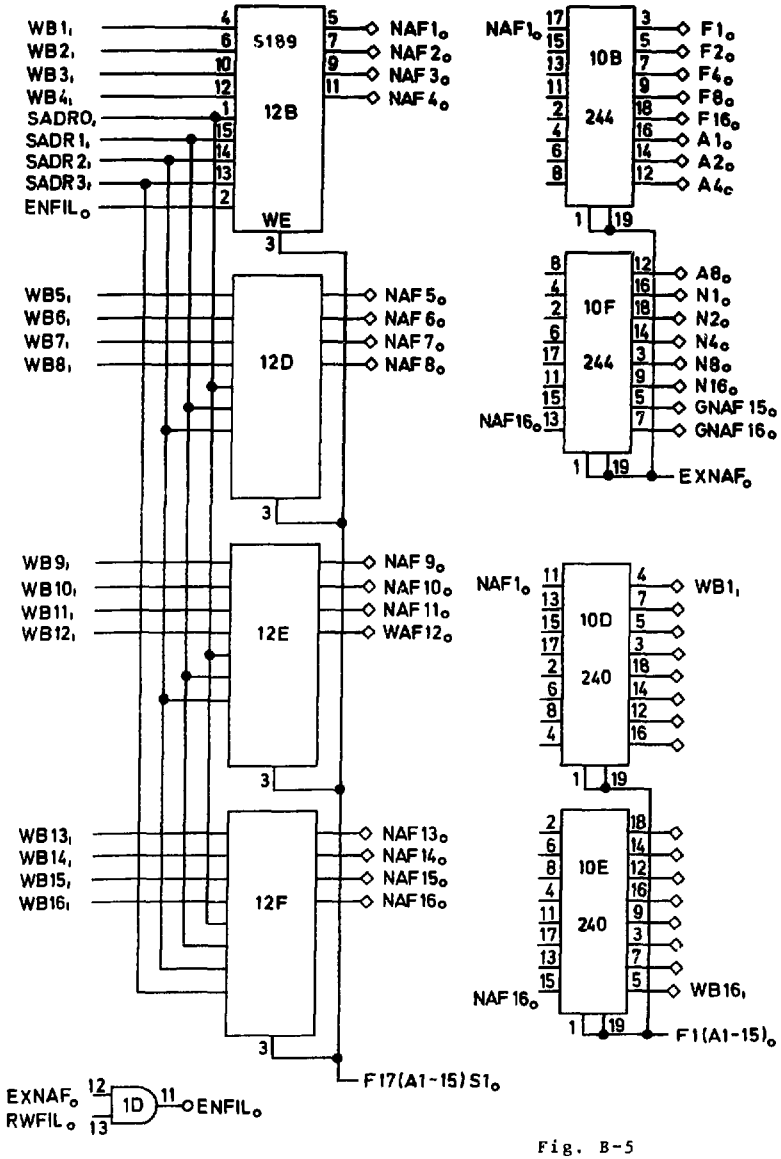

Fig. B-5
N-10 Scanner Card A
NAF-file + Buffers

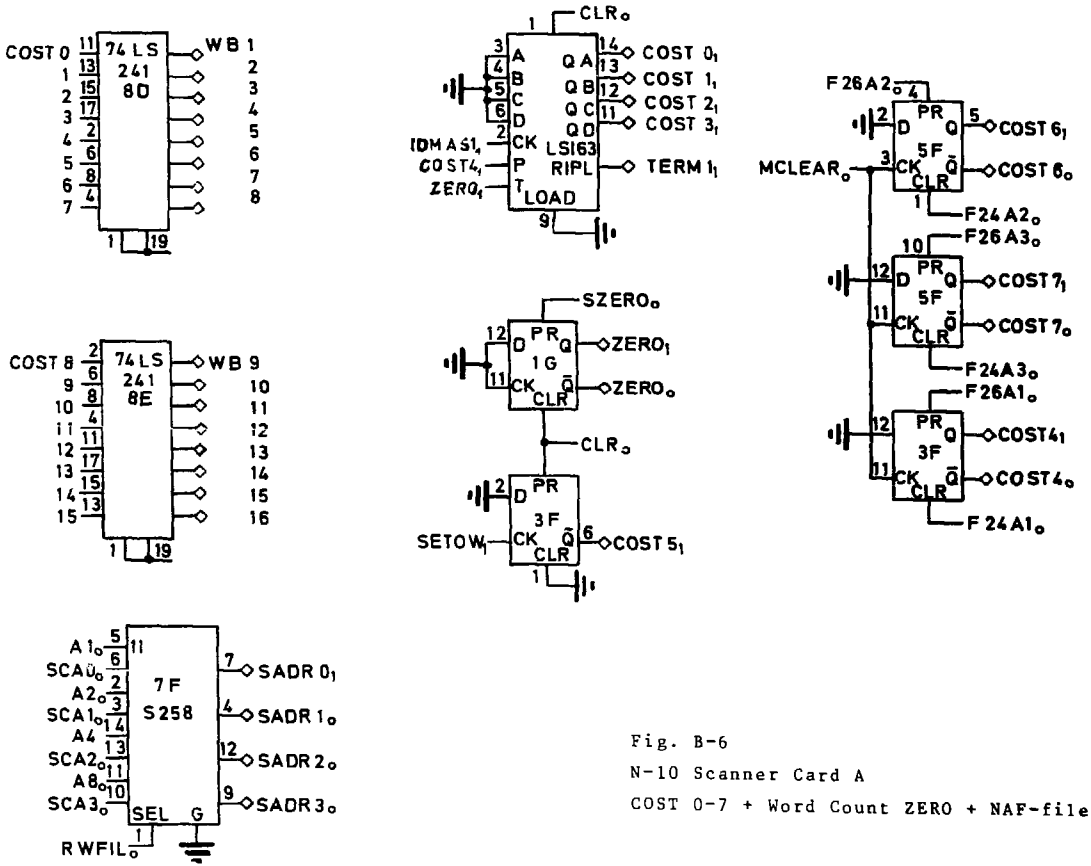

Fig. B-6

N-10 Scanner Card A

COST 0-7 + Word Count ZERO + NAF-file Address

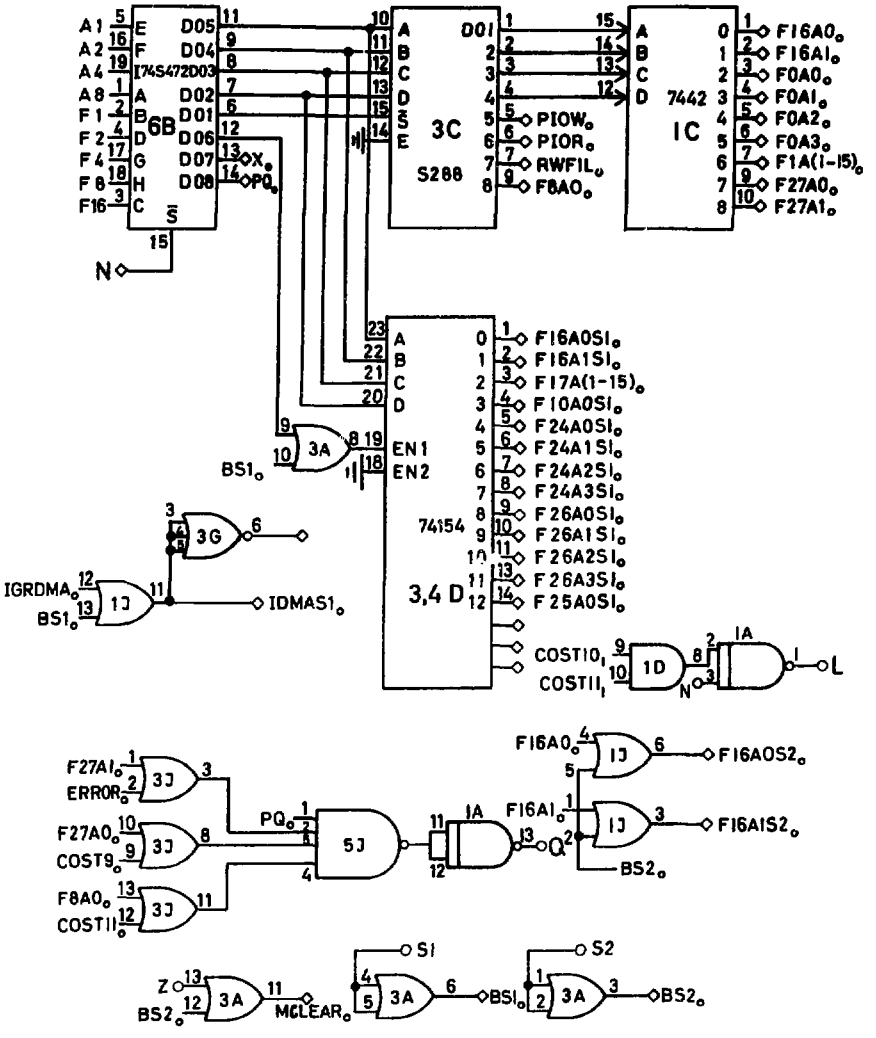

Fig. B-7
N-10 Scanner Card A
NAF Decoder

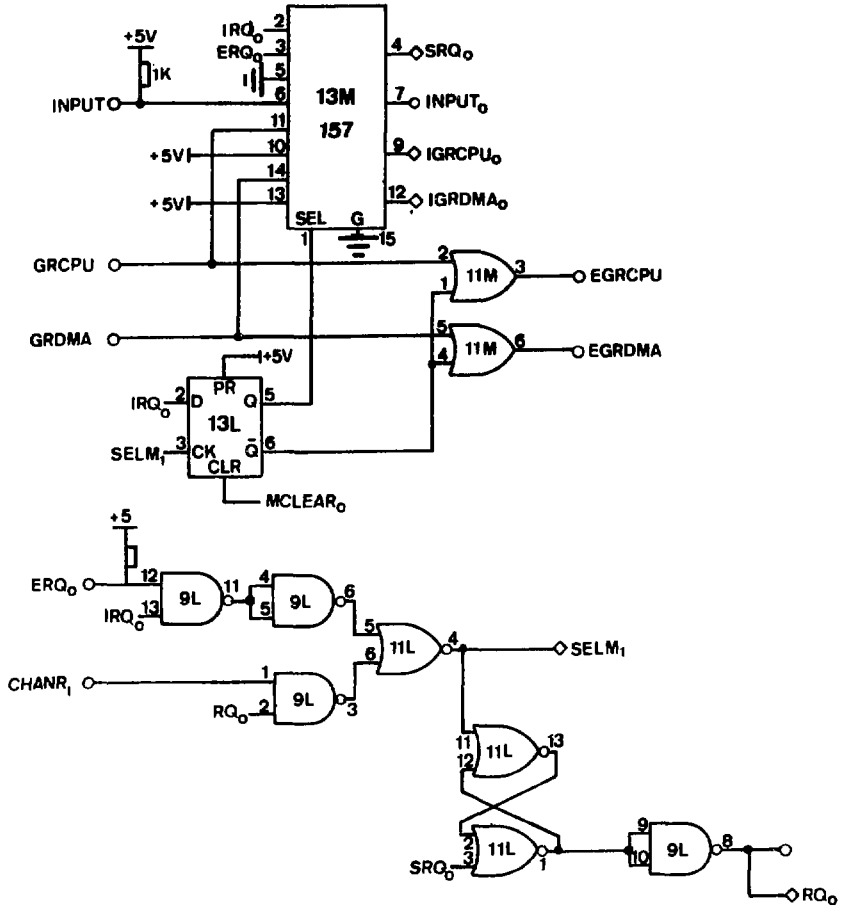

Fig. B-8
N-10 Scanner Card A
LINK-BUS Control

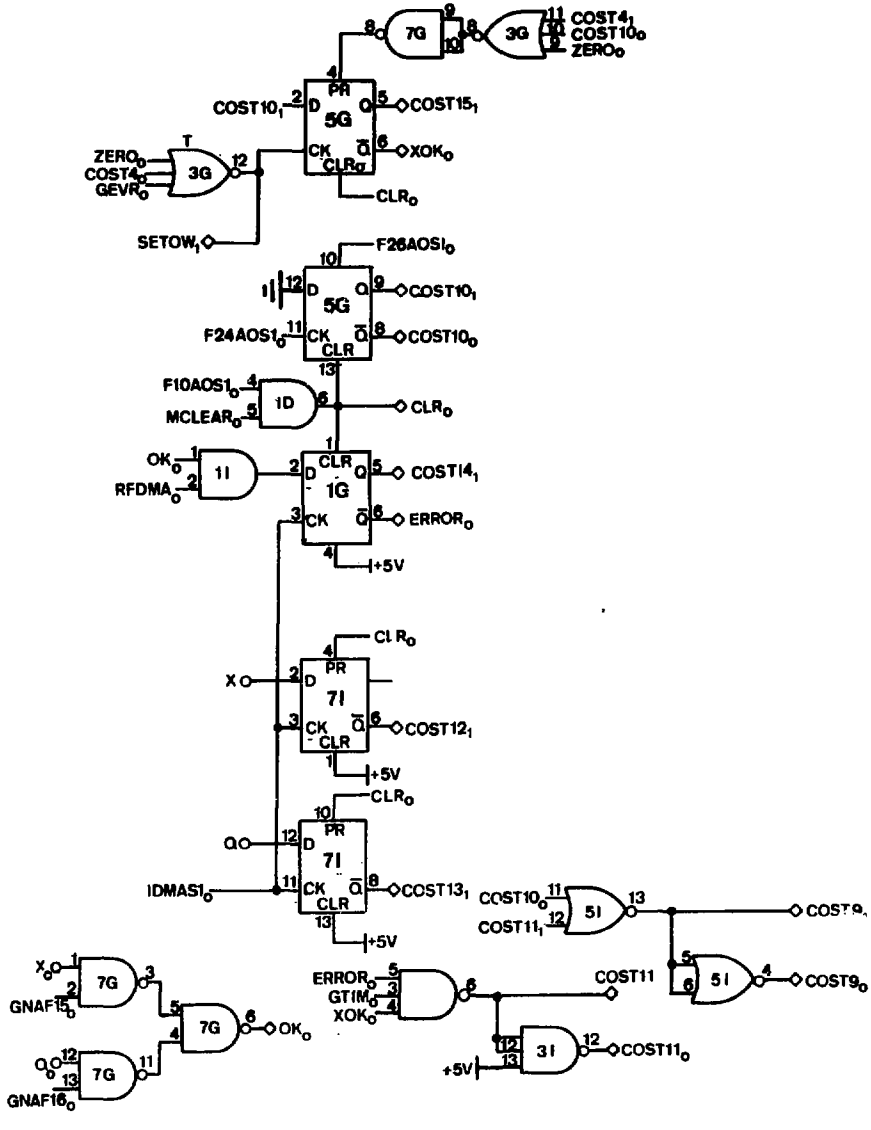

Fig. B-9
N-10 Scanner Card A
COST 9-15

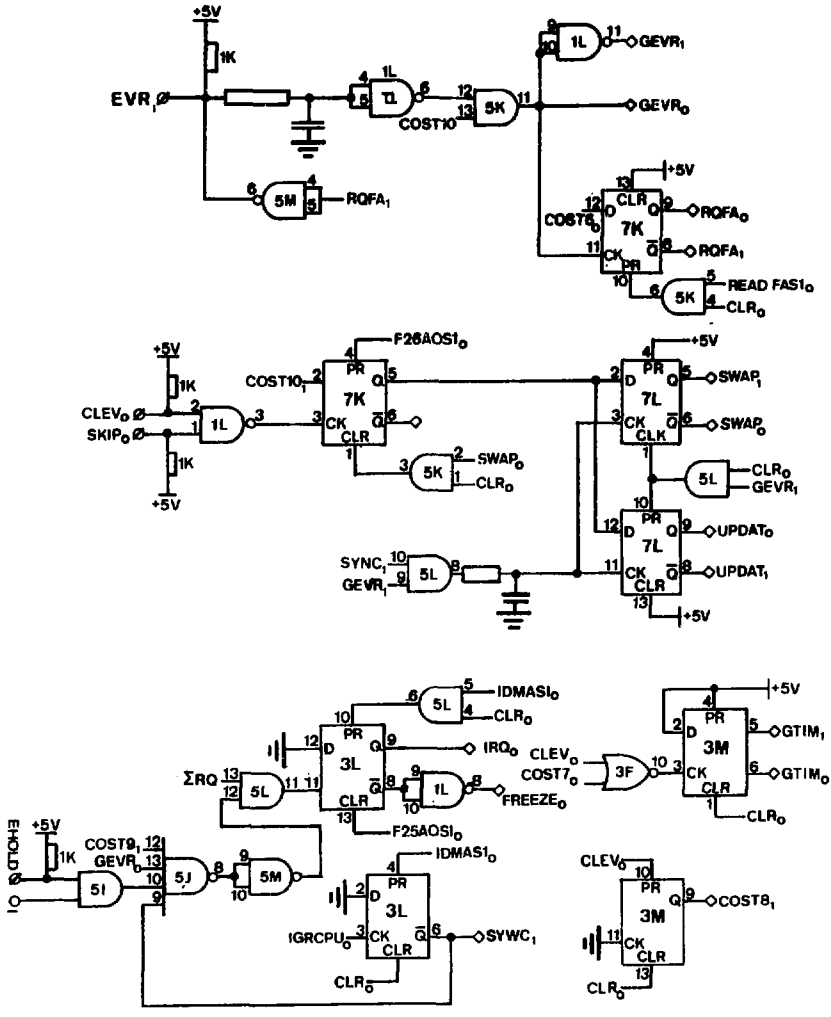

Fig. B-10
N-10 Scanner Card A
Event Control

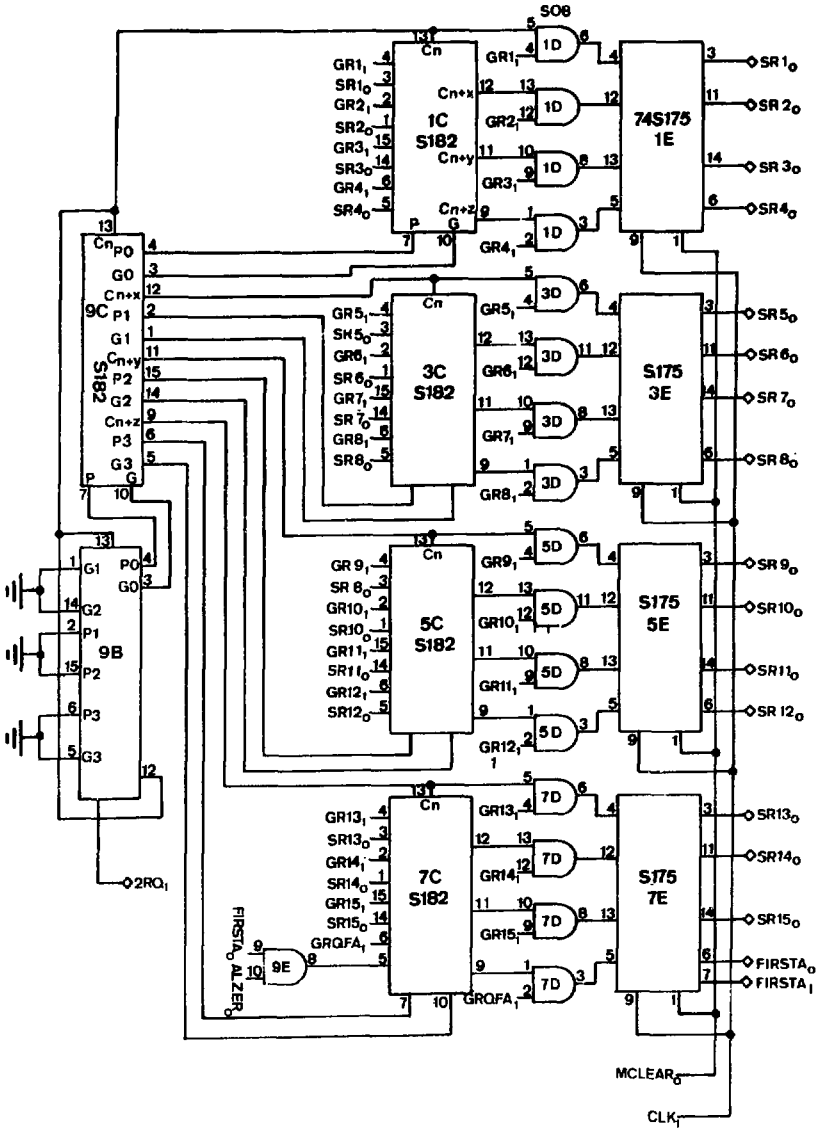

Fig. B-11

N-10 Scanner Card B

Arbitration Logic

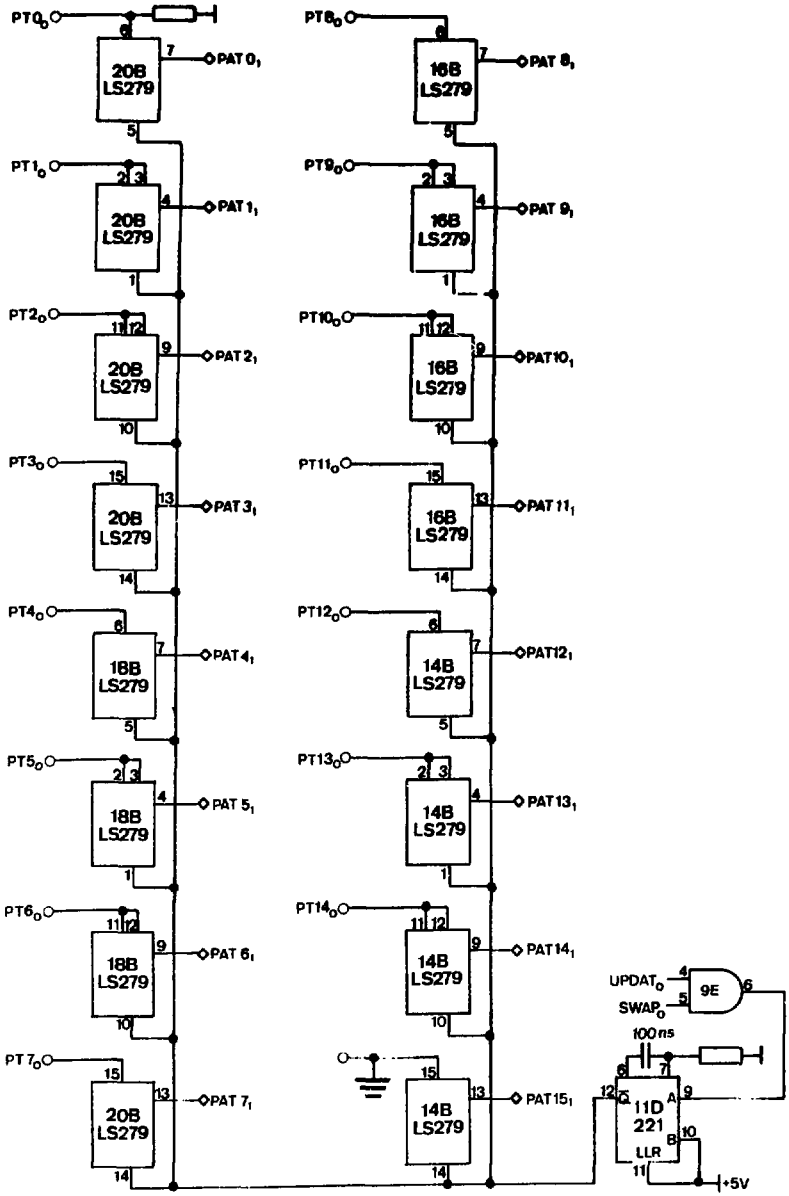

Fig. B-12
N-10 Scanner Card B
Pattern Register

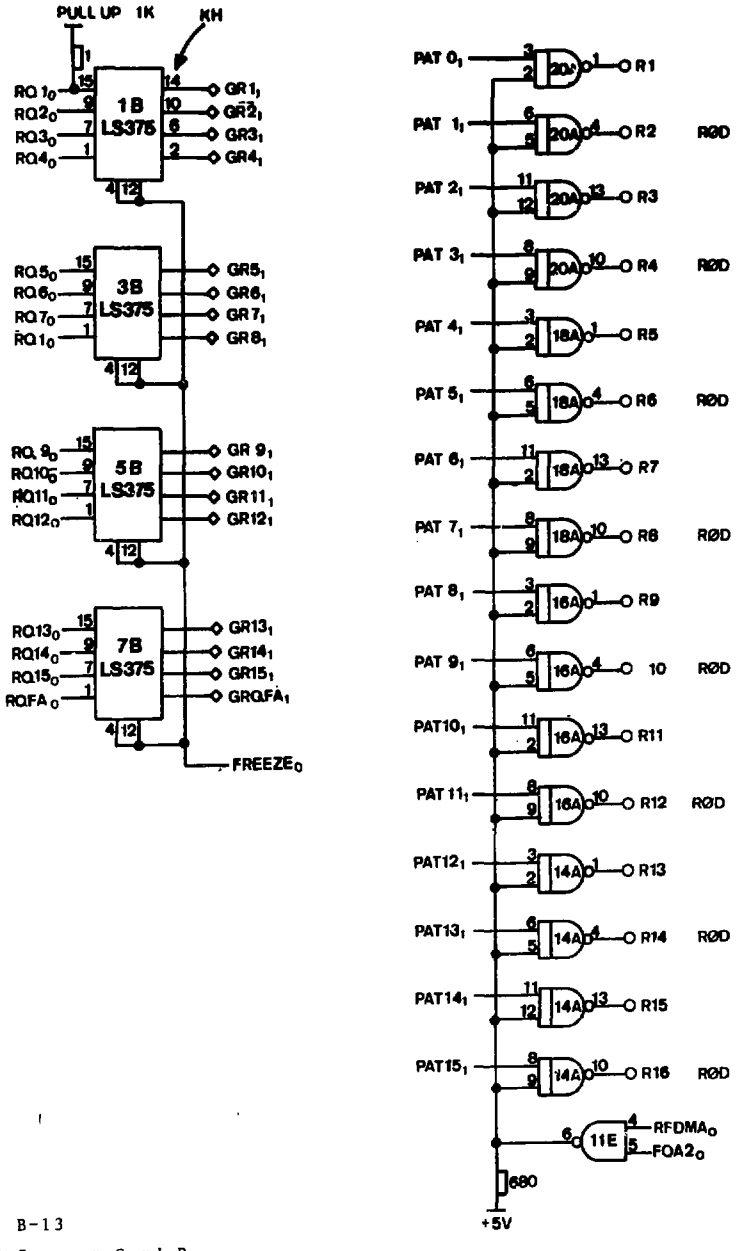

Fig. B-13

N-10 Scanner Card B

ADC Request Latches + Pattern Read Buffer

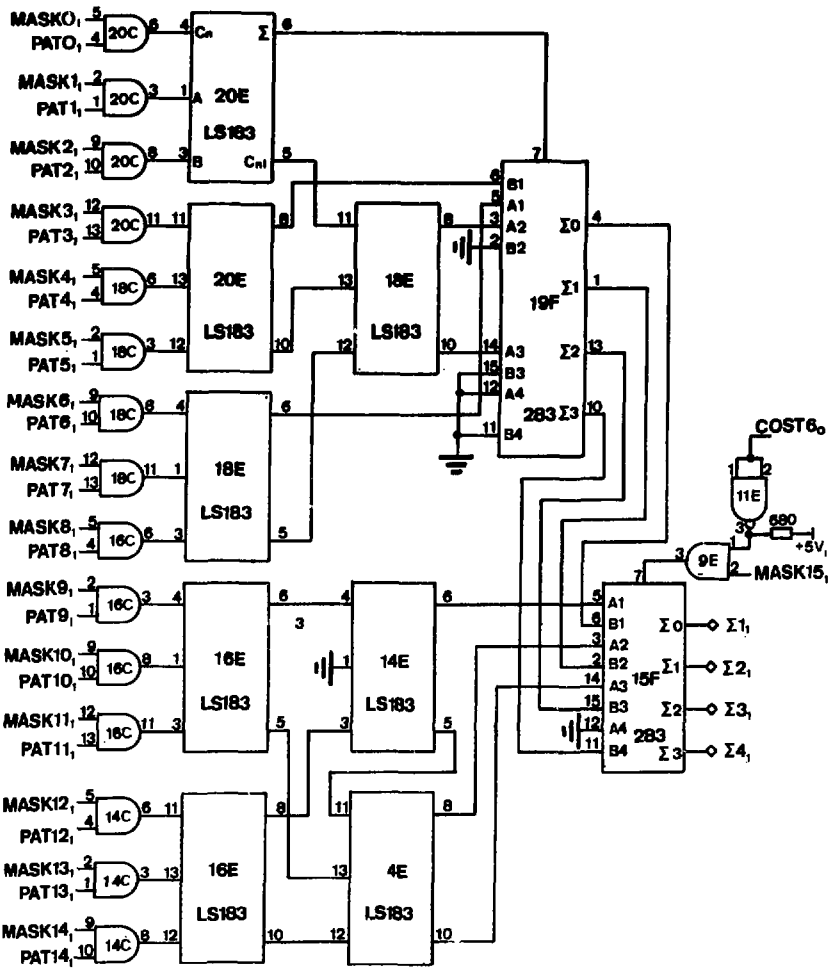

Fig. B-14
N-10 Scanner Card B
Displacement Vector

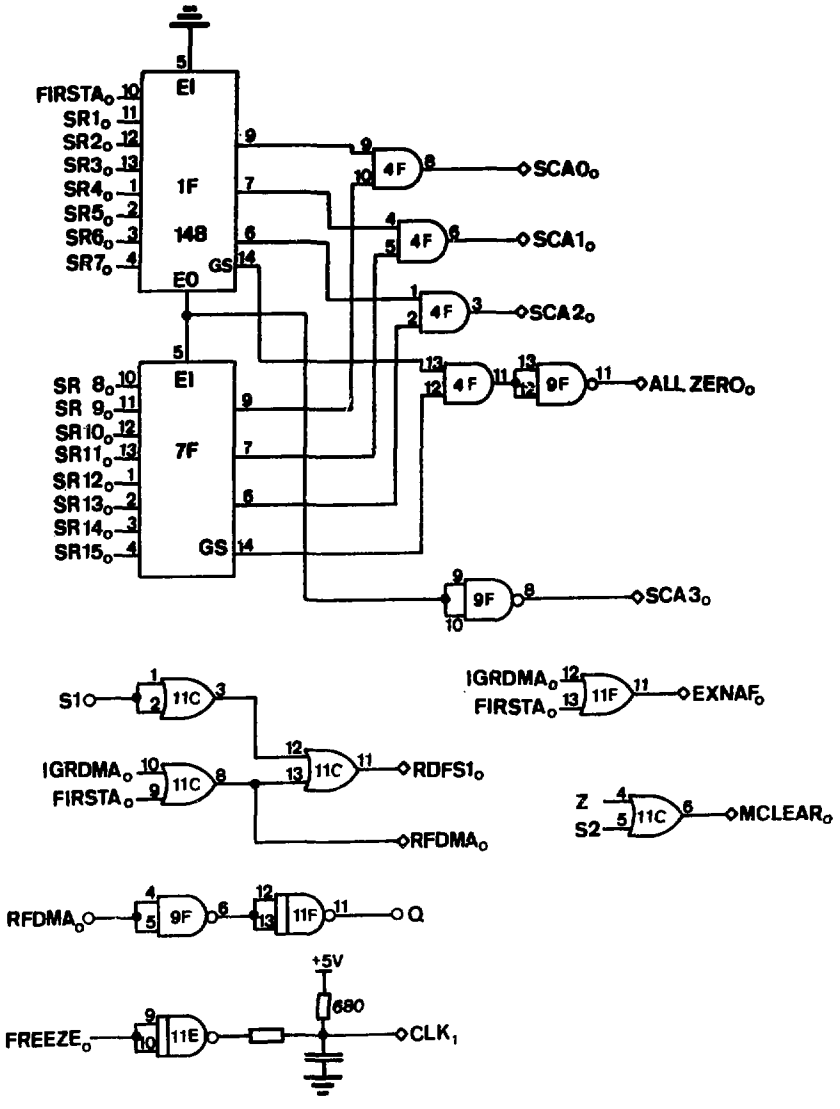

Fig. B-15

N-10 Scanner Card B

ADC Number Encoder + Misc. Control

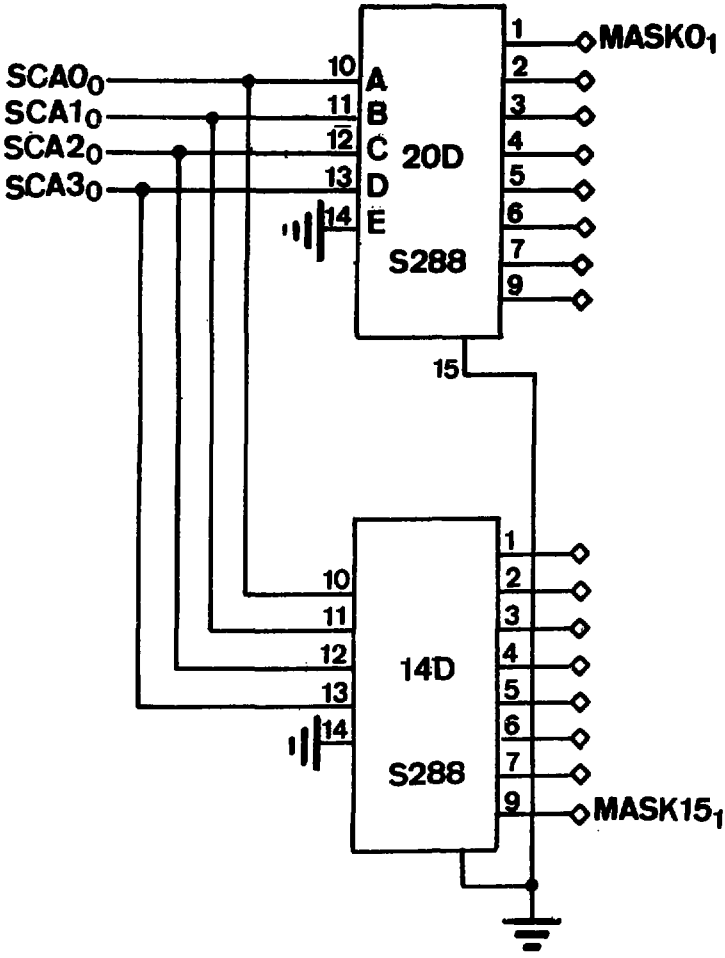

Fig. B-16
N-10 Scanner Card B
MASK PROMs for Displacement Vector