



LIBRARY OF THE  
UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN

510.84

Il 6r

no. 812 - 817

cop. 2



## CENTRAL CIRCULATION BOOKSTACKS

The person charging this material is responsible for its renewal or its return to the library from which it was borrowed on or before the **Latest Date** stamped below. **You may be charged a minimum fee of \$75.00 for each lost book.**

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

TO RENEW CALL TELEPHONE CENTER, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

MAR 11 1998

When renewing by phone, write new due date below  
previous due date.

L162



Digitized by the Internet Archive  
in 2013

<http://archive.org/details/graphicsonetermi814artw>





510.84  
IL62  
no. 814  
cop 2

## GRAPHICS ONE TERMINAL DESIGN AND USER'S MANUAL

by

BRUCE A. ARTWICK and ALFRED D. WHALEY

July 1976



DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

The Library of the

DEC 10 1976

University of Illinois  
at Urbana-Champaign





GRAPHICS ONE TERMINAL DESIGN AND USER'S MANUAL

by

BRUCE A. ARTWICK and ALFRED D. WHALEY

July 1976

DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  
URBANA, ILLINOIS 61801

This work was supported in part by the United States Energy Research and Development Administration under contract US ERDA E(11-1) 2383.



## FOREWORD

The research and development described in this report occurred during the period from January 1975 to April 1976 with many delays due to parts availability problems. The graphics system described is believed to be one of the first in a new generation of computer graphic terminals incorporating large bit maps stored in high density random access memories, and using internal high speed micro-computers for character and vector generation.

Credit for participation in the early development stages and for the design of portions of the processor data paths and control structure goes to Lawrence Lopez.

This research was supported in part by the United States Energy Research and Development Administration under contract US ERDA E(11-1) 2383.



## TABLE OF CONTENTS

	Page
INTRODUCTION . . . . .	1
SCREEN DISPLAY OPERATION . . . . .	3
Matrix Generator . . . . .	3
Memory System and Cards . . . . .	5
Memory Address and Timing Control Card . . . . .	7
Address Structure and Registers . . . . .	8
Writing and Reading Screen Memory . . . . .	10
PROCESSOR STRUCTURE . . . . .	11
Data Units . . . . .	12
Arithmetic and logic unit. . . . .	12
Accumulator. . . . .	12
Memory mask multiplexer. . . . .	12
Data Units . . . . .	12
Program counter. . . . .	12
Address register . . . . .	13
PC/address register multiplexer. . . . .	13
Control System . . . . .	14
Operation register . . . . .	14
Sequencer. . . . .	15
Hardware implementation of sequencer . . . . .	15
Sequencer initialization . . . . .	15
Sequencer pause now signal . . . . .	18
PC/address multiplexer control . . . . .	19



Branch decoder . . . . .	19
Clock. . . . .	19
1K RAM/ROM Memory System. . . . .	21
Peripheral Bus Structure. . . . .	21
Design Schematics . . . . .	21
System Performance . . . . .	23
SYSTEM PROGRAMMING . . . . .	24
Background . . . . .	24
Addressing Modes. . . . .	25
Instruction Set . . . . .	27
I/O Device Register Locations . . . . .	29
INTERFACING GUIDE . . . . .	31
Using the Peripheral Bus . . . . .	31
Slow Devices and Memory . . . . .	32
Peribus Lines . . . . .	32
Data Routing System . . . . .	32
Back Panel Wiring . . . . .	33
APPENDIX A . . . . .	34
APPENDIX B . . . . .	49
APPENDIX C . . . . .	59





## INTRODUCTION

The system described in this report was developed out of the need for a low cost computer terminal with extensive graphic capabilities. Preliminary design criteria included simple hardware construction using a micro-computer for flexibility, expandability, character generation, vector generation and other graphics tasks.

Bilevel screen display data is stored in a large random access memory which is read out sequentially to a video display, one bit at a time. Through cycle stealing, memory and thus picture modification is accomplished. The generated dot matrix is of arbitrary dimensions which are not necessarily powers of two therefore advanced counting schemes are used to keep track of line and bit counts.

Due to the system's low cost constraint, a standard television receiver is used as a video monitor by cutting into the video circuitry after the RF tuner. Standard 525 line resolution is used. Small matrix displays (up to 200 dots square) use double line resolution while larger displays use the interlace facilities to increase resolution. Alternate frames must therefore be displayed on the two interlaced fields adding to the complexity of the counting system.

In the development of the terminal the need for a high speed micro-computer for fast character and vector generation became apparent. Although MOS and bipolar processors were currently available, a TTL design was chosen. This design offers the advantages of being inexpensive, more specialized for graphics work and much faster than MOS processors. After development was completed a fairly powerful computer resulted with about the same number of integrated circuits required to

interface to popular microprocessors at about one third the cost.

This report is presented in two sections. The screen display generator will first be presented followed by the microcomputer section.

### SCREEN DISPLAY OPERATION

The screen display unit consists of three parts.

1. The matrix generator
2. The memory control and addressing unit
3. The random access memory system

#### MATRIX GENERATOR

A vertical line counter and a horizontal bit counter along with corresponding count recognizers keep track of the matrix size and position. When a preset horizontal bit count is recognized the bit clock and the video display are gated off. Through the adjustment of the bits per line recognizer and the bit clock rate, display width and density can be respectively adjusted. The left margin of the matrix is determined by delaying the clock and video restart from the horizontal sync pulse.

The vertical counter has two recognizers. The start display recognizer determines which line from the top of the display will be used to start the matrix and also triggers a memory initialization system which prepares display data one line before the display begins. The stop display recognizer turns the video enabling system off when the desired number of lines has been displayed. Although the video is blanked at the bottom of the screen, memory reading continues in order to keep the dynamic memories refreshed. A new memory restart address is issued to the memory counter upon frame initialization one line before the display starts.

The horizontal bit count recognizer watches the horizontal counter and determines when ten, twenty or forty bits have been displayed.

The memory control circuitry can use the resulting signal to prepare new information by loading new data from the random access memory into a large shift register (see memory section). This is the read request load shift register signal.

The matrix generator requires that valid data always be available to be shifted onto the video monitor. It is up to the memory control system to see that this requirement is met and that the memory output looks like the output of a large shift register.

The matrix generation just described is performed by the matrix generator card. Two important subsystems of the matrix generation system are the frame initialization system and the sync signal and video output generator.

A National Semiconductor MM5320 color television video sync generator IC is used to generate all required sync signals and proper input clock waveforms are controlled by a dual one-shot clock. Good waveforms are vital for proper field index generation which is used for the interlacing. A one transistor video mixer has been designed to meet EIA RS 330 standards for video monitor output. The dot matrix is bilevel with no intermediate gray shades employed.

Within the matrix generation sequence there exists a frame initialization step. Before a frame can properly begin, valid data for that frame must be available in the shift register.

A frame starts on line 0. Initialization procedures are performed on line -1 and consist of loading the restart address, reading the memory at that location, loading the memory contents into the latches, and incrementing the memory address register. The memory card design

shows these units and at what times each of these events take place.

Since alternating frames require different frame starting addresses when the interlace facility is being used, a toggle flip flop and two gates have been provided to keep track of alternate fields. Initial synchronization of this flip flop is performed by the field index signal. Figure 1 shows the matrix generator card.

#### MEMORY SYSTEM AND CARDS

Due to the large amount of memory needed to store a dot matrix (160k bits for a 400 square matrix), a 4k dynamic RAM (Intel 2107) was used. The only refresh requirement is that sixty-four rows be read once every millisecond (this memory is organized as 64 rows by 64 columns). This refresh specification is met through the continuous reading of the memory for display purposes.

Calculations showed that fifty nanoseconds per bit access time would be needed to display 400 bits on one forty microsecond line allowing half of the time for memory cycle stealing for external memory access. A memory parallelling scheme was employed to reduce this requirement to 1.5 to 2.0 microseconds depending on the number of RAMs used. For various sized matrices ten, twenty or forty are parallellled leaving 50, 100 and 200 percent safety timing margins respectively.

Latches are used as a high speed buffer memory for RAM data and a card to card chainable shift register is loaded from the latches at the proper time.

Parallel dumping of memory into the latches is performed by simultaneously enabling all the RAMs and strobing in data at the proper time. The RAMs all have a common address. Individual bit access is



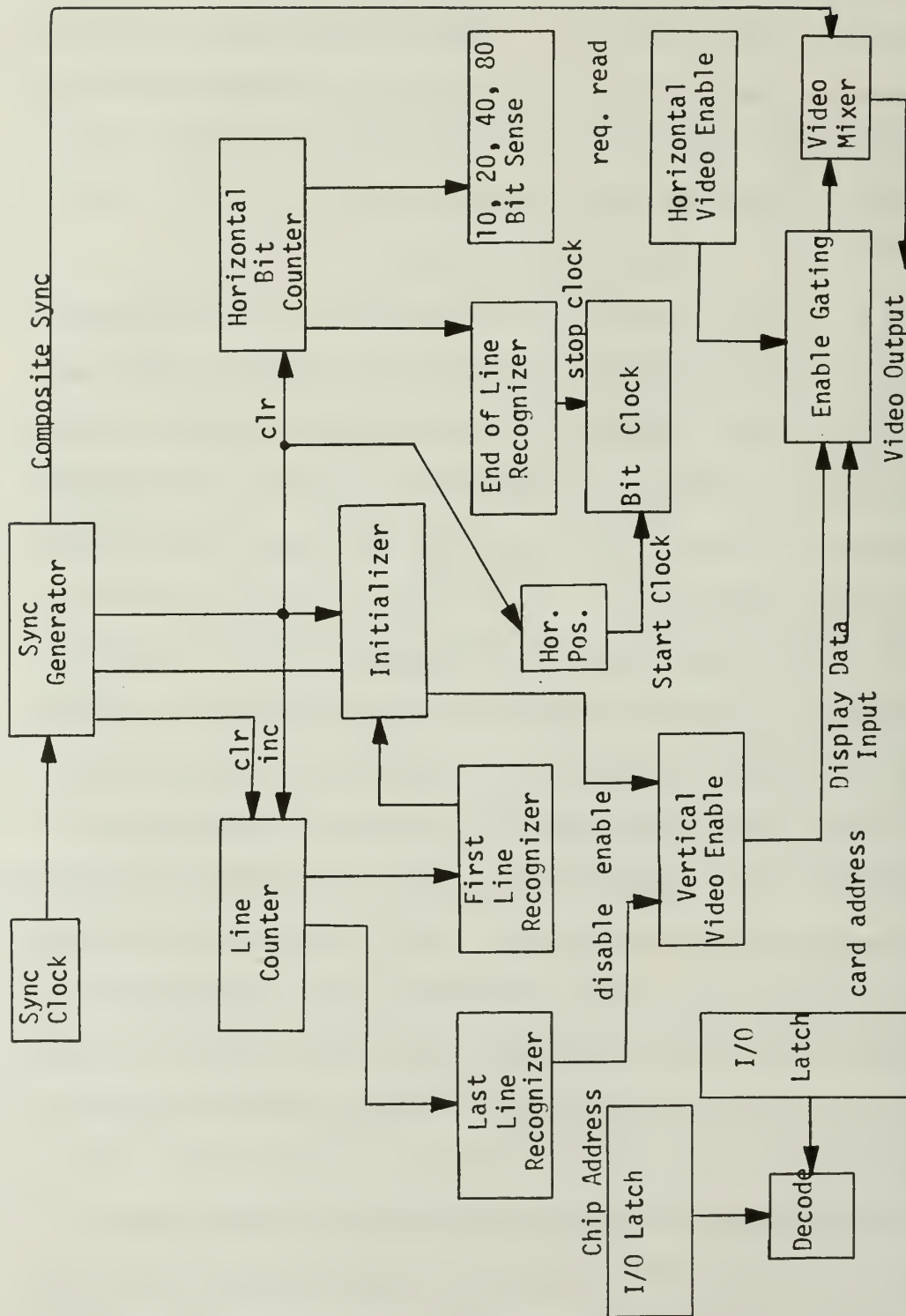


Figure 1. Matrix generator card.

performed by supplying the proper common address and enabling the desired chip which is selected by the chip address decoder. Individual bit reading is accomplished by examining the access data output as only an enabled and thus the selected chip can provide a logic one level.

Individual bit writing can be performed by providing all the RAMs with a write signal. Only the enabled chip will respond, writing the data bit into that RAM.

#### MEMORY ADDRESS AND TIMING CONTROL CARD

The memory control system must:

1. keep track of memory display addresses
2. generate memory timing pulses for read and write cycles
3. interleave display and memory access cycles

Address information supplied to the RAMs has two sources, the memory address counter for sequential display dumping, and the screen data address, for individual bit access by the processor for image modification. The screen data address is buffered for asynchronous loading, allowing the processor to load the buffer at any time yet permitting the display unit to synchronously load the address when needed. The address counter is controlled by the matrix generator. The memory timing sequence determines at which times either address is needed and operates the memory address multiplexer accordingly. The chip and card address selector operates in the same way, however additional decoding of the address is provided to drive the memory card decoders. This circuitry is on the matrix generator board due to spacing problems.

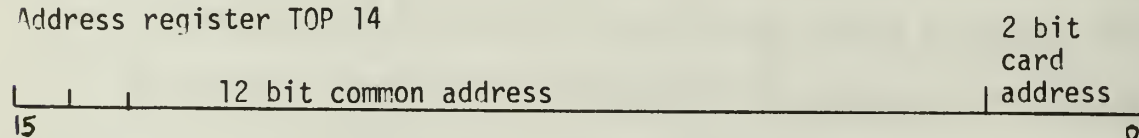
Upon receiving a read request signal from the matrix generator the memory timing system generates pulses to produce a read / read

modify write sequence. The display address is read and data is loaded into the memory cards' latches. After the read is completed, the address from the screen data address register is multiplexed to the memory and a read modify write cycle is performed. A write flip flop can be externally set by the controlling processor and the next read modify write cycle will write the contents of the input data into the addressed location. A read will also be performed at this address whether requested or not and the data is available for processor use. Proper delays and pulse durations as specified by Intel have been met. Figure 2 shows the memory control and timing card.

#### ADDRESS STRUCTURE AND REGISTERS

The display unit is designed to be driven by a sixteen bit data bus and therefore appears to be three registers. The two address registers can be loaded by strobing the load bottom 4 or load top 14 lines. The set write flag line is also available to the processor to perform accesses.

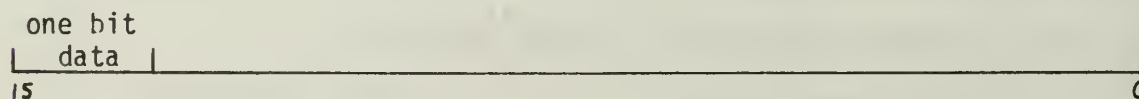
Address register TOP 14



Address register BOTTOM 4



Screen Data



The memory system design results in the following means of



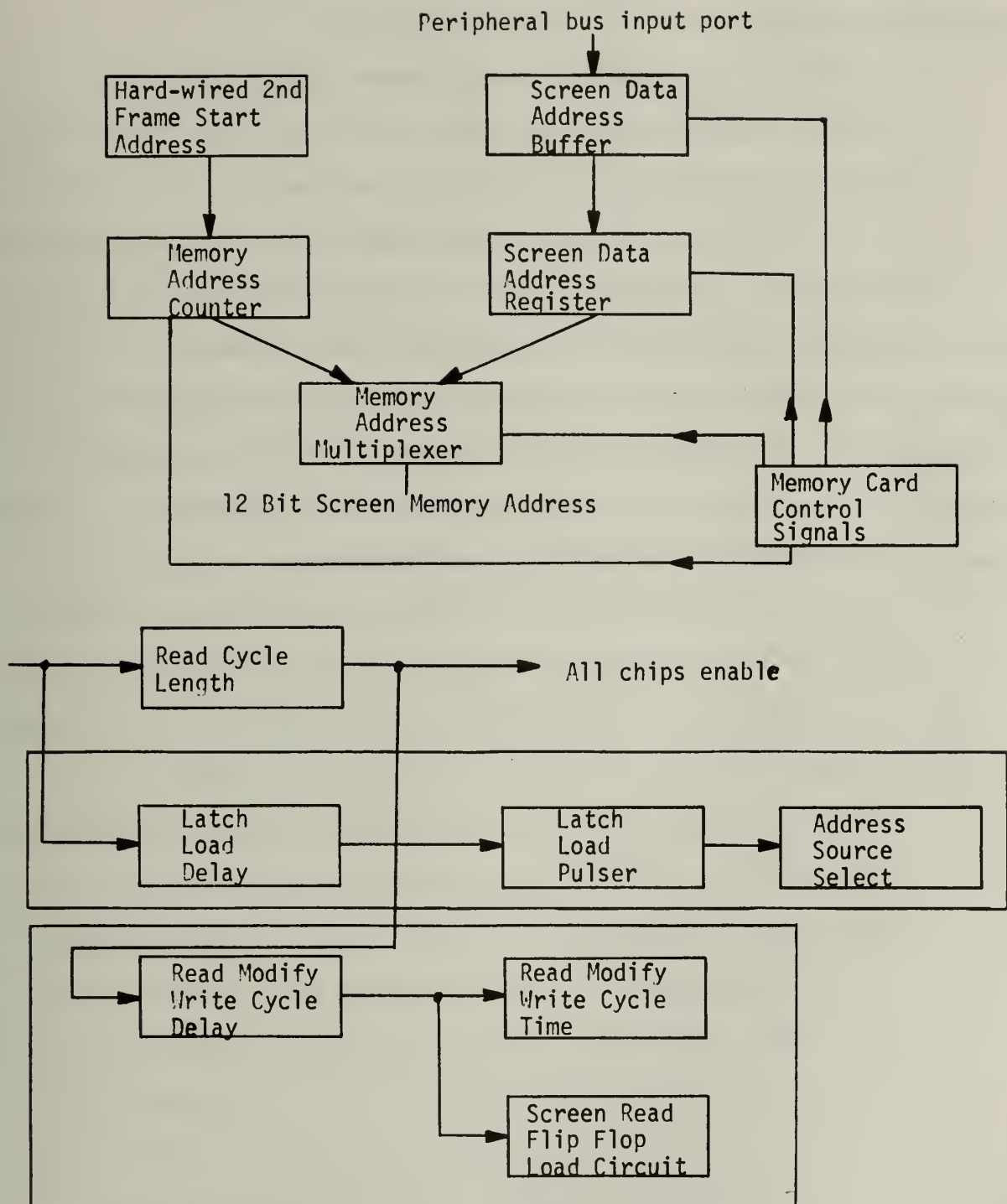


Figure 2. Memory timing and control card.

addressing individual bits.

1. A 2 bit card address (for up to 4 memory cards)
2. A 12 bit common address (for the 4K RAMs)
3. A 4 bit chip address (10 memory chips per card)

#### WRITING AND READING SCREEN MEMORY

Although the screen display unit can take data bits to be written at an average rate of one bit per three microseconds, there are periods of up to twenty or thirty microseconds where the display clock is stopped (right and left screen margins) where no data can be written. The write flag can be used as a screen busy flag for the controlling processor so writing during these periods can be avoided.

## PROCESSOR STRUCTURE

The two constraints the processor is based upon are simple, highly efficient design and high speed operation. Data paths and control schemes were designed accordingly. A single accumulator sixteen bit parallel processor design was used. Sixteen bits provide one indirect bit, a five bit opcode and a ten bit address field. This allows for a 1k range of direct memory addressing and sixteen bit arithmetic which was thought to be enough for a graphics terminal. Through indirect addressing a full 64K of memory or device registers can be accessed.

Arithmetic operations are based on the versatile functions of the 74181 arithmetic and logic unit. Through the use of an extended opcode field on nonmemory reference (generic) instructions where the address field is not used, a large instruction set is possible.

The processor unit is built on two nine by five inch circuit boards which are screwed together to form an 88 pin two level plug-in module. The processor is contained on one card while the clock and 1K of read only and random access memory are on the other. Twenty connections are made between the cards with plug-in flexible cables.

The processor module can be divided into 4 basic units.

1. Data units
2. Address units
3. Control system
4. Memory system

These will now be explained.

## DATA UNITS

Three units form the basis for data flow:

ARITHMETIC AND LOGIC UNIT. Four 74181 ALU integrated circuits are used to perform arithmetic, logic and multiplexing operations between memory data and accumulator data with the result going to the accumulator

ACCUMULATOR. Two 74198 eight bit bidirectional shift registers comprise the accumulator. All shifts and rotates are done in the accumulator.

The output from the ALU can be read into the accumulator and the output can be strobed into memory or used by the ALU as an operand.

MEMORY MASK MULTIPLEXER. On immediate mode instructions, the lower ten bits from the address field are used as data. The six bit memory mask multiplexer provides either a 1 or 0 to all six top ALU memory inputs on these operations. Bit nine determines if the top six bits will be 0 or 1, thereby acting as a sign propagation bit.

The established data paths form four classes of data manipulating instructions.

1. No load-no store: operates only on accumulator (e.g. complement)
2. load-no store: loads data from memory and may operate on it (e.g. add)
3. no load-store: stores data to memory which may have been operated on (e.g. complement and store)
4. load-store: memory and accumulator contents are operated on and stored (e.g. add and store)

## ADDRESS UNITS

PROGRAM COUNTER. Four 74161 high speed look ahead counters are used to

keep track of the program location. Every instruction performed supplies a clock pulse to these synchronous counters. Normally, the count is incremented but if a branch operation is in effect this clock pulse causes the PC to be loaded with the effective address if the proper conditions (+,-,zero or hardware flag) are met.

ADDRESS REGISTER. Two 74174 and one 74175 six and four bit D latches form this register. An instruction fetch causes the instruction to be loaded and the top six bits cleared.

case 1. If this is a memory reference instruction the address can be multiplexed to the memory and the memory contents used as data.

case 2. If this is a branch instruction the contents can be loaded into the PC for the branch address.

On indirect instructions, normal referencing is used (case 1 above) but all sixteen bits of the memory contents are loaded into the address register. On the operational phase of the indirect instruction, case 1 or 2 again apply but this time a full sixteen bits are available to address locations outside of the lower 1K of memory.

PC/ADDRESS REGISTER MULTIPLEXER. Four 74157 two to one four channel multiplexers determine which address the memory will use; the PC or the address register. The control unit section tells when each is appropriate.

## CONTROL SYSTEM

The operation register, sequencer, branch detector, and clock comprise the control unit.

OPERATION REGISTER. This is an 8 bit D latch (one 74174 and one 7474) which is loaded during the instruction fetch. Seven bits are used to store the opcode of the instruction and one bit stores the value of an external input to the processor (the external flag). The external flag can be tested by an instruction and is useful for branching on certain external conditions which is further covered in the branch detector section. Further decoding of the opcode is needed for instruction execution. Two 8223 programmable read only memories derive the necessary 13 bit ALU and sequencer gating control codes.

## ROM 1 - ALU Function Selector

Bit	Signal	Function
1	Set 2 indicator	Activates opcode extensions
2	S3	
3	S2	ALU function control
4	S1	
5	S0	
6	Cn	
7	Mode	
8	Immediate mode indicator	

## ROM 2- Sequence Control ROM

1	Branch
2	No Read
3	Read
4	Write
5	Branch On External Flag

An indirect instruction temporarily disables ROM 2 to hold off execution of that instruction until the indirect address has been accessed. After the address has been accessed, the indirect bit latch (part of the operation register) is cleared preventing further indirecting of the instruction. Figure 4 shows the operation register and ROM



decoding.

SEQUENCER. Synchronous control points are used to provide control sequences for the processor. Clocked flip flops, each representing a state, are chained together to form control chains. The opcode of the instruction to be performed determines the control sequence to be used. The state chain structure is shown in figure 3.

HARDWARE IMPLIMENTATION OF SEQUENCER. A string of D flip flops is used to impliment the flow chart. Through proper initialization, one bit can be routed through the timing chains with derived pulses being used for timing. Efficient design was used in making full use of flip flop features. Instead of gates being used to channel bits from input to output, the overriding clear inputs are held low or high to accomplish the same thing. This has the added benefit of clearing out any erroneous bits which may have been introduced in the timing chain. Figure 5 shows how the D latch clear feature is effectively used.

The opcode, via the control ROM and extension bit decoder, determines which flip flops will be held clear and where the bit will flow.

The timing diagram with the design schematics shows which pulses are used to load the data units and memory at what times. NR means no read, NW means no write, R means read and NRNW represents no read, no write. The NRNWx2 is a useful sequence which loads the accumulator twice. This allows for double shifts and rotates which are code and time saving.

SEQUENCER INITIALIZATION. Initialization is performed by setting the initialization SR flip flop. The S0 state flip flop is preset and the

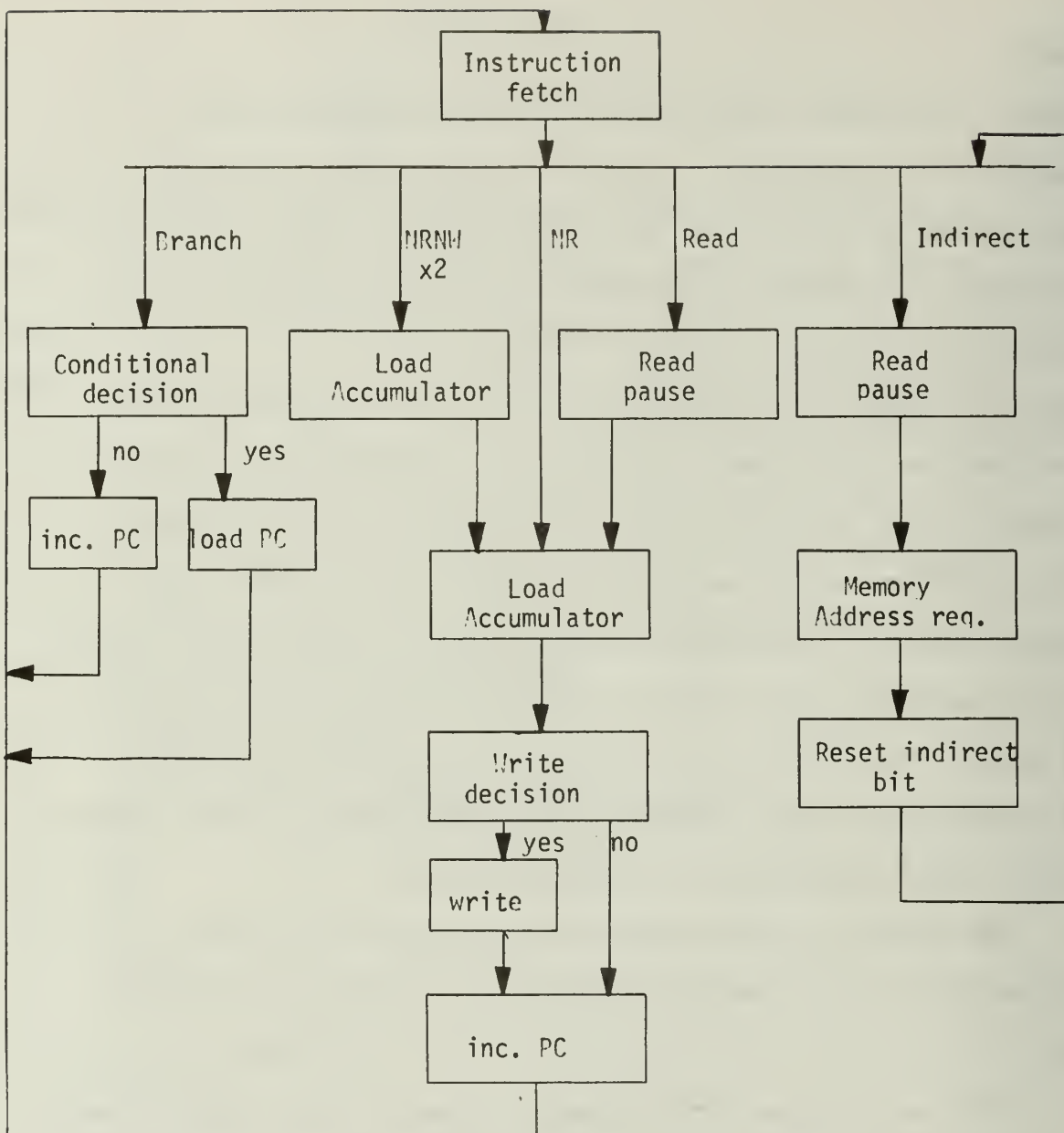


Figure 3. Flow chart of state chain functions.



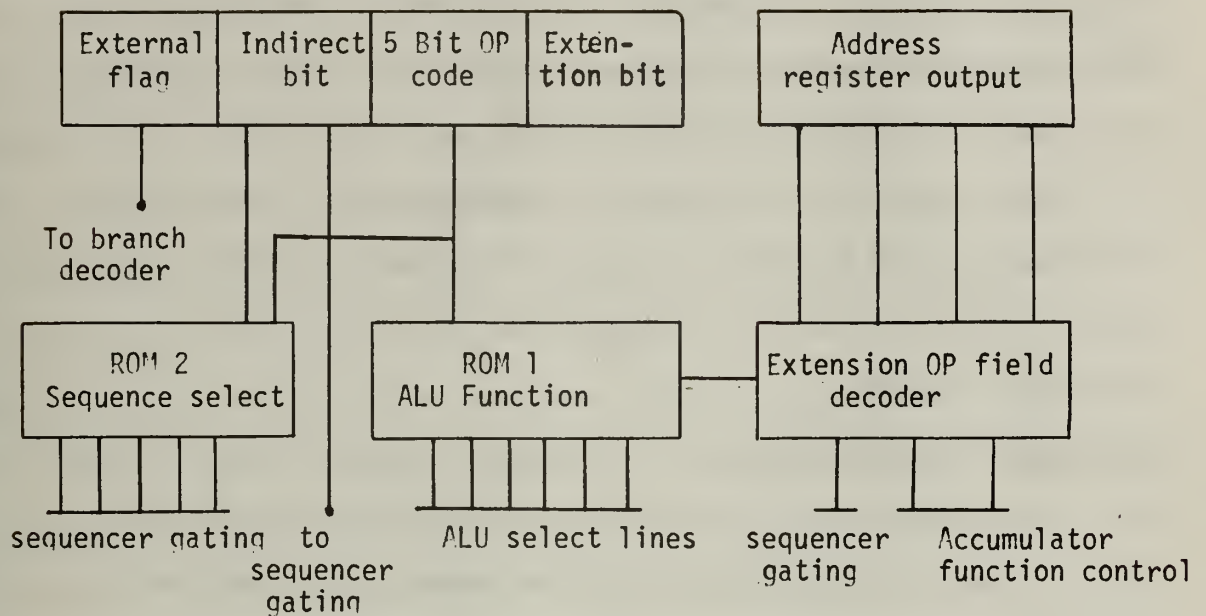


Figure 4. Operation decoding and sequence control.

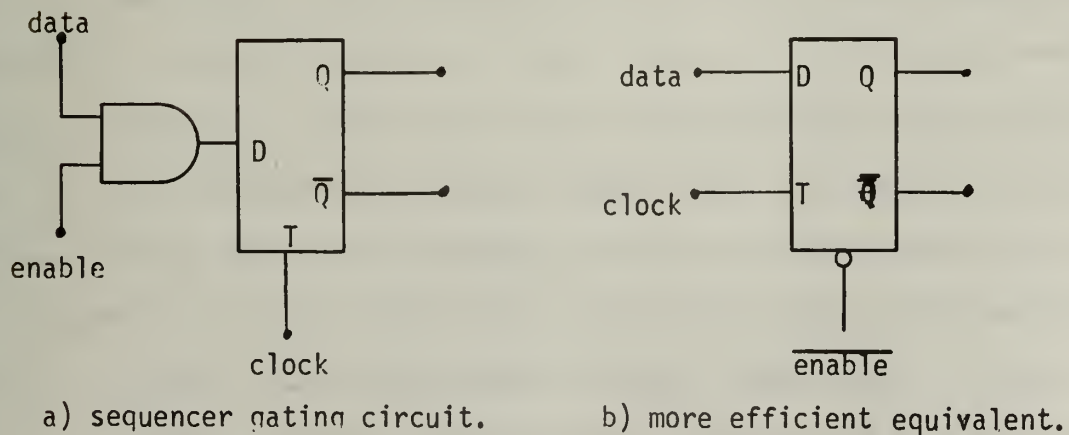


Figure 5. Efficient use of flip flops.

S1 flip flop is cleared. The running clock propagates '0's from S1 throughout the timing chains. The PC is cleared and the valid data at memory location 0 settles. When the initialization flip flop is reset, execution at 0 begins on the first leading edge of the clock. The initialization flip flop is provided for debouncing and a standard SPDT switch can be used to trigger it. If a fast device is used to initialize the system, 600 ns must be allowed for memory settling time in an initialization pulse.

SEQUENCE PAUSE NOW SIGNAL. This processor is designed to run extremely fast so when slow memory (access time of more than 35 ns) is used, the processor must halt operations until the memory data becomes valid. Since fast device registers and PROMS as well as slow MOS RAMs are used, responsibility to stop operation of the processor has been delegated to the memory being addressed. When the processor knows a memory read pause will be required if slow memory is used, it sends out a pulse called the Pause Now signal. If the slow memory being addressed sees this, it can pull down (hold at logic 0) the clock enable line. This stops the processor clock for as long as is needed to access the memory. A one-shot in the memory unit is ideal for this application. When data becomes valid, releasing the clock enable line resumes processing. It should be noted that no pause now signal is given on the write signal. It's up to the memory being used to pause if it needs extra write time. Forty ns after the clock enable line is released, the memory address changes (only on the write cycle) in preparation for the next instruction fetch. Care should be taken in memory design to insure that address hold requirements are met.

PC/ADDRESS MULTIPLEXER CONTROL. During the fetch cycle, the PC is multiplexed to the memory address to fetch the instruction at that address. After the operation register and address register are loaded the address register takes over the memory address. On immediate mode instructions, however, the PC controls the memory address throughout the operation so the data can be used (immediate mode instructions contain an opcode and data field).

BRANCH DECODER. Every instruction performed causes a clock pulse to go to the PC clock input. The branch instruction is no exception, as the PC can not only be incremented but loaded as well. If the proper conditions are met, the data is loaded from the address register and would be the branch address specified by the instruction.

The lowest four bits of the opcode specify the branch conditions and the accumulator high bit (for accumulator zero) and the external flag provide the comparison. An and-or gate decodes the result of the comparison and activates the PC load function which will synchronously be performed instead of an increment. Figure 6 shows branch decoder operation.

CLOCK. A frequency dividing flip flop is used to buffer the clock and to let the clock be stopped at either a high or low level. At high clock frequencies the clock stopping system may not have time to stop the clock before a new clock pulse comes out of the master clock. Since the sequencer states are activated only on leading edges this small pulse will have no effect because it will cause a trailing edge in the final output. This acts as a safety factor.

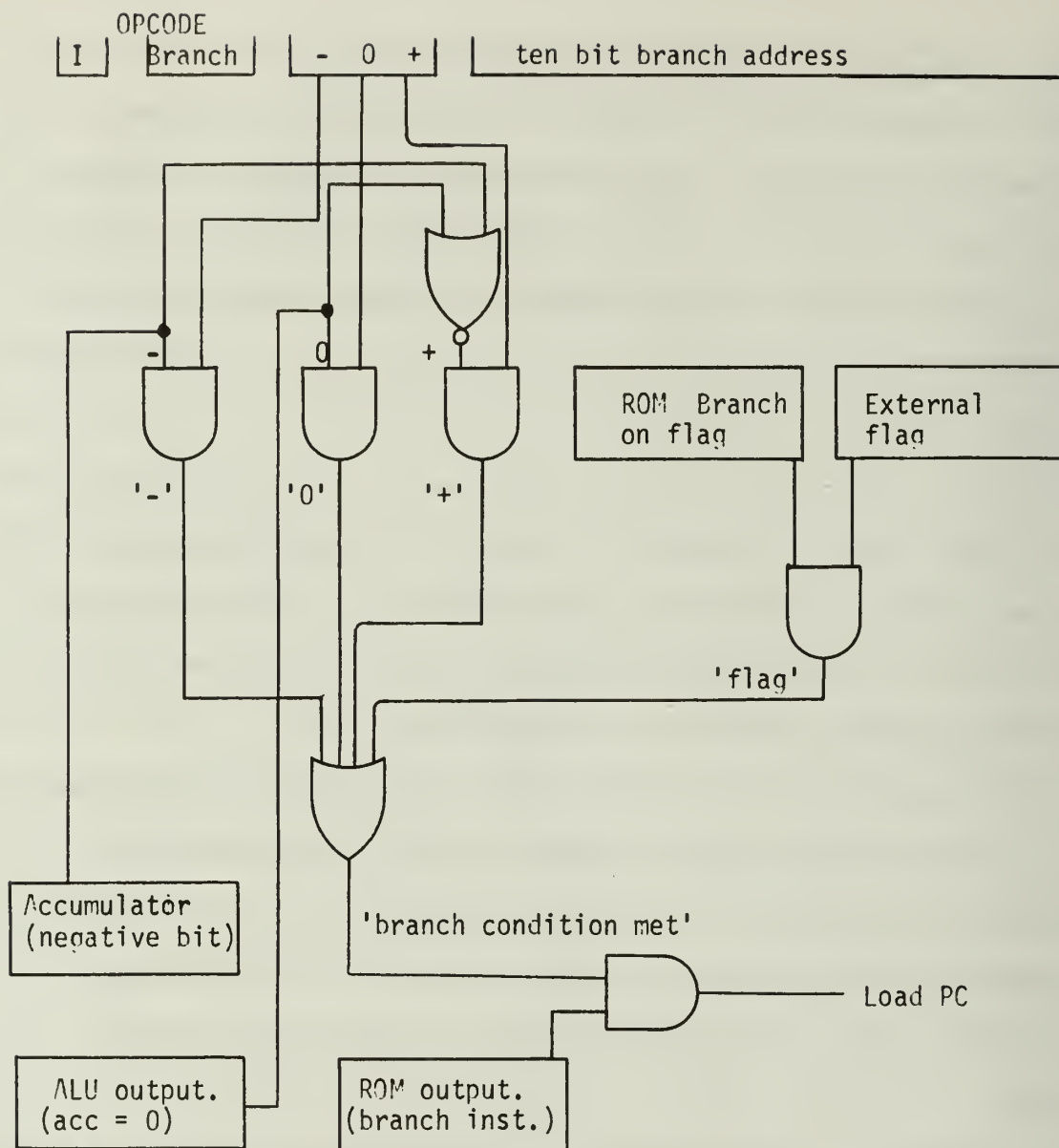


Figure 6. Branch decoding network.

## 1K RAM/ROM MEMORY SYSTEM

The second card in the processor module contains 1K of 500 ns access time random access memory for program storage which is overlapped from location 0 to 32 with a 32 word bootstrap loading program stored in two 8223 read only memories. A two to one multiplexer is used to form two data input busses to the processor. One is open collector and the other is tristate. Only the tristate bus, however, is available for devices external to the processor card. The read only memory is an open collector device on the other bus. A decoder is used to control the multiplexer and the devices on the bus depending on which device is being addressed.

Two one-shots are associated with the RAMs and stop the processor clock upon receipt of a pause now signal. Writing into memory also causes a pause but of different length (for the 300 ns write cycle). The one-shots have been set up to activate the pauses whenever anything in the lower 4K of memory is addressed. This is useful for memory extension as new delay circuitry is not needed for addresses within this range.

## PERIPHERAL BUS STRUCTURE

Data flow to and from other devices external to the main processing unit is done on the peripheral bus. This bus was a result of the processor design and turned out to be a convenient bus to use. Figures 7 and 8 show the peripheral bus structure and signals. More is said about the peripheral bus in the interfacing section.

## DESIGN SCHEMATICS

Hardware simplicity was the goal of the design and thus it does

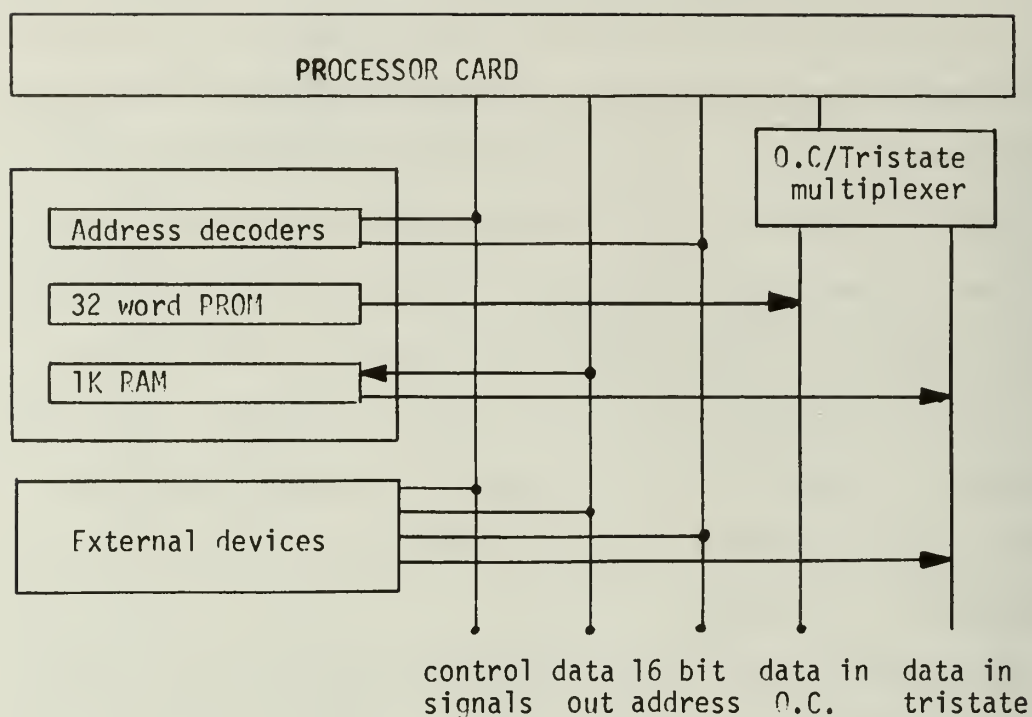


Figure 7. Peripheral bus structure.

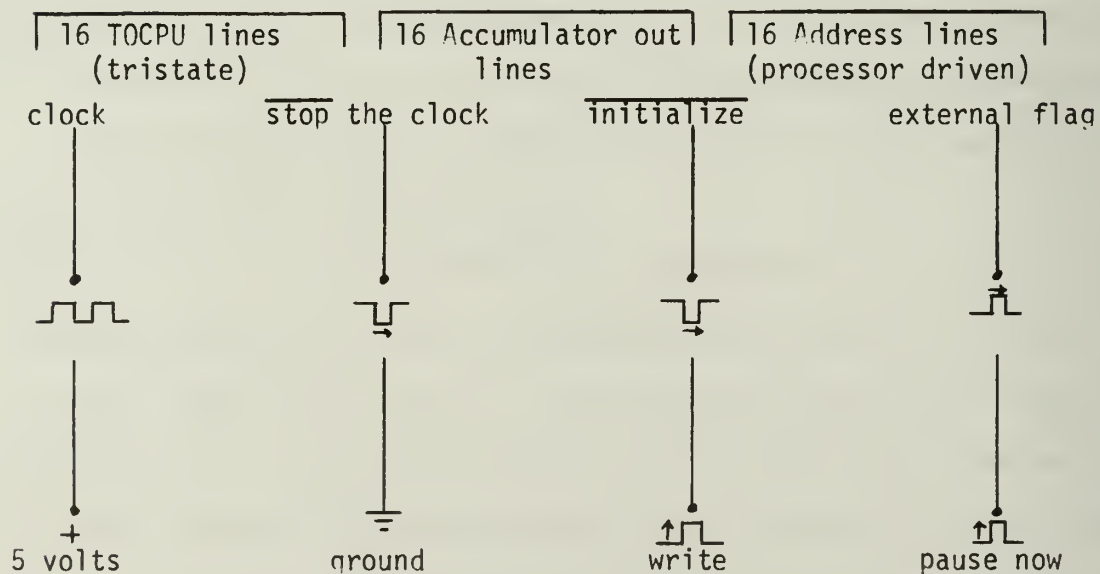


Figure 8. Peripheral bus lines.



not have many gates or flip flops relative to the many functions performed. What it does have, however, may be hard to understand without a close study in some cases because some gates perform many functions (in particular the set 2 decoder, the indirect ROM 2 enable system and the MRMW 2 level gating system). The rest is straight forward. The timing flow diagram and the symbolism on the design drawings show which edges and levels do what.

### SYSTEM PERFORMANCE

The micro computer has incorporated many schottky and high speed parts to obtain high speed operation. Standard clock speed is 20 MHZ which is immediately divided by the clock flip flop to provide a micro cycle time of 80 NS. The processor has been tested to a higher speed.

### SPECIFICATIONS

Instruction Set	39 total (65 with variations)
Instruction Time	740-1660 NS including memory cycles
Addressing Modes	Immediate, deferred, data, direct
Power Requirements	5 volts, 2.0 amps
Clock Frequency	25 MHZ
Microcycle Time	80 NS
Logic	36 TTL MSI and SSI ICs
Arithmetic Unit	74181 ALUs

## SYSTEM PROGRAMMING

A guide to programming the processor with relation to the terminal system will now be presented.

### BACKGROUND

The processor is a dynamic machine; not in the sense of needing refreshing but in that it has no stop instruction and once initialized, is always running.

Initialization clears the PC and starts execution at address zero. In the graphics version of the processor, thirty-two words of read only memory act as a bootstrap program to load the main program from another source (the controlling computer). Once loaded, execution is diverted to the main program stored in the one K of random access memory which was loaded by the bootstrap. There is no way to switch a program into the processor with this arrangement. A larger computer or some sort of buffer memory must be used.

Much thought has gone into the instruction set to make instructions as powerful and useful as possible. One instruction programmers will miss is the subroutine call. This would have taken more hardware to implement and was felt to be not worth the extra cost. Only simulated subroutines are possible.



## ADDRESSING MODES

The processor has a sixteen bit operational field. The bottom one K of memory can be addressed directly while deferred addressing is available for high memory addressing (up to sixty-four K).

DIRECT AND INDIRECT MEMORY ACCESS MODE.

I	Op	Op	Op	Op	Op	A	A	A	A	A	A	A	A	A	A
---	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Indirect      5 bit opcode      10 bit address

This is used for every memory access instruction. The indirect bit signifies indirect addressing and the contents of memory will be used as the operand.

EXTENDED INSTRUCTION MODE. The shifts and rotates all have the same five bit op codes. Four extension bits determine the type of shift or rotate

Op	1	0	0	0	1	Ex	Ex	Ex	Ex	N	N	N	N	N	N
----	---	---	---	---	---	----	----	----	----	---	---	---	---	---	---

Opcode                      Opcode extension      Not used

The indirect bit can be set but since this is a generic instruction, direct or indirect addressing has no meaning. An indirect shift however takes 2 clock pulses longer than a direct shift due to the extra memory fetch.

IMMEDIATE MODE INSTRUCTIONS. The indirect add, subtract, load and logical and, all use the ten bit address field for data. The top seven bits of the instruction assume the value of the propagation bit. The add instruction can be used to add small negative or positive constants to the accumulator. By making the propagation bit zero an add results

and a propagated one causes an add of a negative or a subtract. The SUBM subtracts the accumulator from a small constant. This is quite useful when a negate is desired ( $0-ACC=-ACC$ ). Indirect immediate mode instructions use the value of the contents of the opcode plus the constant. This is probably a useless mode for this instruction.

BRANCH INSTRUCTIONS. The direct addressing mode is used for branch instructions. Indirect branches to high addresses are possible also and execution will continue in high memory unless indirect mode is again used. More hardware would have been necessary for paging. All 7 combinations of plus minus and zero are available as branch conditions. Note that a branch on plus, minus or zero is an unconditional branch.

READ AND WRITE INSTRUCTIONS. These use the same address for read and write operations. This is useful for modifying memory but it should be remembered that the accumulator also assumes the memory value and is modified, thus losing its old contents. The memory increment or decrement (INCS and DECS instruction) should be useful for pointers since it is loaded, incremented and stored in one operation.

GENERIC INSTRUCTIONS. Instructions such as COM (compliment accumulator) which only modify the accumulator use the same format as the direct or indirect memory access instructions but the ten bit address has no meaning.

DATA MODE. The whole sixteen bit field can be used to store data as long as it is never executed. The indirect bit has no significance thus only one level of indirect addressing is possible. This was chosen over multilevel indirecting due to the need for sixteen bit arithmetic.

INSTRUCTION SET. The following instruction list gives opcodes in both octal and binary. The opcode and the address share an octal digit to include the tenth address bit.

Operation	Opcode	Address	Instruction
Store 1774	054	1774	056774

The cross assemblers for this processor assemble opcodes automatically and this makes things much easier.

## PROCESSOR INSTRUCTION SET

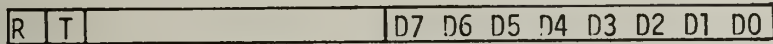
INSTRUCTION	MNU	CYCLES	TIME	OPCODE		OCTAL
MEMORY ACCESS INSTRUCTIONS						
ACC→MEM	ST	4	1160	ns 110110	address	154
MEM→ACC	L	4	1160	101000		120
MEM^ACC→ACC	AND	4	1160	101001		122
∨	OR	4	1160	101010		124
⊕	XOR	4	1160	101011		126
+	ADD	4	1160	101100		130
-	SUB	4	1160	101101		132
-1→ACC→MEM	NOS	4	1160	110011		146
0→ACC→MEM	CLRS	4	1160	110100		150
ACC→ACC→MEM	CAMS	4	1160	110010		144
MEM→ACC→MEM	COMS	5	1660	101110		134
∧	ANDS	5	1660	111001		162
∨	ORS	5	1660	111010		164
⊕	XORS	5	1660	111011		166
+	ADDS	5	1660	111001		170
-	SUBS	5	1660	111101		172
MEM+1→MEM→ACC	INCS	5	1660	111110		174
MEM-1→MEM→ACC	DECS	5	1660	111000		160
ACC→ACC	COM	4	740	001111	arbitrary	036
IMMEDIATE MODE						
ACC+CONST→ACC	ADDM	4	740	011111P	constant	076
ACC^CONST→ACC	ANDM	4	740	010000P		040
CONST→ACC	LOADM	4	740	010101P		052
CONST-ACC→ACC	SUBM	4	740	010111P		056
SHIFTS AND ROTATES						
SHIFT RIGHT	RS	4	740	0100011100	arb. 6	0434
SHIFT LEFT	LS	4	740	0100011000		0430
ROTATE RIGHT	ROR	4	740	0100011110		0436
ROTATE LEFT	ROL	4	740	0100011010		0432
SHIFT RIGHT x2	RS2	5	820	0100011101		0435
SHIFT LEFT x2	LS2	5	820	0100011001		0431
ROT. RIGHT x2	ROR2	5	820	0100011111		0437
ROT. LEFT x2	ROL2	5	820	0100011011		0433
BRANCH INSTRUCTIONS						
BR. ON FLAG	BF	4	740	100000	address	100
BR. + ACC	BP	4	740	100001		102
BR. - ACC	BM	4	740	100010		110
BR. + or -	BPM	4	740	100011		112
BR. ZERO ACC	BZ	4	740	100100		104
BR. + or ZERO	BPZ	4	740	100101		106
BR. - or ZERO	BMZ	4	740	100110		114
BRANCH	B	4	740	100111		116

NOTE: I=Indirect Bit P=Propagation Bit

## I/O DEVICE REGISTER LOCATIONS

## LOCATION

177440 / UART RECEIVE REGISTER



```
UART rec. done
Trans req empty
```

UART receive reg. data

note: Writing in this register clears the receive done flag.

177500 / KEYBOARD



Key pressed

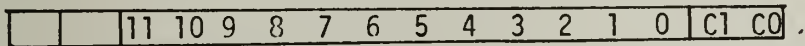
## Keyboard data

## 177540 / SCREEN DATA



Display screen data at preloaded screen address.

## 177404 / SCREEN DISPLAY ADDRESS TOP 14 BITS



12 bit common address

Card address

## 177405 / TOP 14

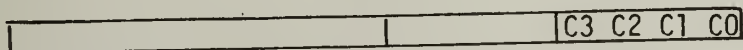
Same as 177404 but sets screen address and writes a '0' on screen.

177425 / TOP 14

Same as 177404 but sets screen address and writes a '1' on screen.

## 177410 / SCREEN DISPLAY ADDRESS BOTTOM 4 BITS

Sets screen address (bottom 4 bits)



Chip address

177411 Bottom 4 BITS

Same as 177410 but sets screen address and writes '0' on screen.

177431 / BOTTOM 4

Same as 177410 but sets screen address and writes '1' on screen.

177402 / UART TRANSMIT REGISTER

	D7	D6	D5	D4	D3	D2	D1	D0
--	----	----	----	----	----	----	----	----

Data to be sent.

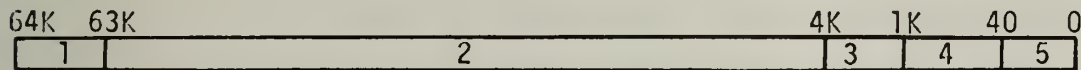
000000-00037      BOOTSTRAP LOADING ROUTINE

177775-177777      RESERVED FOR LOADER



## INTERFACING GUIDE

USING THE PERIPHERAL BUS. Sixty-four K of memory and device registers are accessible through indirect addressing and devices on the peripheral bus should be designed to recognize and respond to an address within this field. The graphics version of the processor has a 4k slow peripherals and memory address field with built in 500 ns delays. Device registers are conventionally located in the upper one K of memory. This allows for memory expansion without register relocation in software.



- 1) fast device registers
- 2) expansion memory
- 3) slow memory and devices
- 4) main memory
- 5) ROM bootstrap

Figure 9. Basic memory map.

A typical device on the peripheral bus consists of an address recognizer, a register and a tristate gate network if the device can write on the bus. Since common write and Pause Now lines are used for all peripherals, design must be such that devices perform these operations only when addressed. Extreme caution is exercised to make sure two sets of tristate gates are not driving the bus at the same time as gate failure could result.

SLOW DEVICES AND MEMORY. If a slow device is to be located in the upper 60K addresses, the Pause Now and Stop the Clock features must be used. Upon receiving a write or Pause Now signal, the device must pull down the Stop the Clock line as quickly as possible and for the duration of the data settling time keep it down. The clock must, therefore, be stopped within forty ns after the Pause Now or a new leading edge will result causing next state execution.

74121 one-shots have an 80 ns propagation delay making them unsuitable for this application. When properly used, a 74122 can give a low value of 23 ns and is more useful.

PERIBUS LINES. Address and accumulator output lines are TTL level driven by 74198 and 74157 integrated circuits within the processor. Loading rules for these ICs must be observed. Data to the processor (Tocpu) is on a tristate bus. The proper transmitters and receivers (8t34) are used. Stop the Clock line is open collector driven and has a 360 ohm pull-up resistor. The gate used to sink the current is properly rated. The write line indicates that the processor is trying to write into a memory location. Data is valid throughout its duration. Initialize is low when initializing. The external flag input can be used for whatever condition is desired as a branch on flag point. The branch is performed on a logic one at this input.

DATA ROUTING SYSTEM. In applications where many device registers are used, instead of having address recognizers and timing circuitry for each register it is more efficient to have a recognizer which recognizes a block of registers (the device registers) using the top portion of the address, and recognize a register within that block using

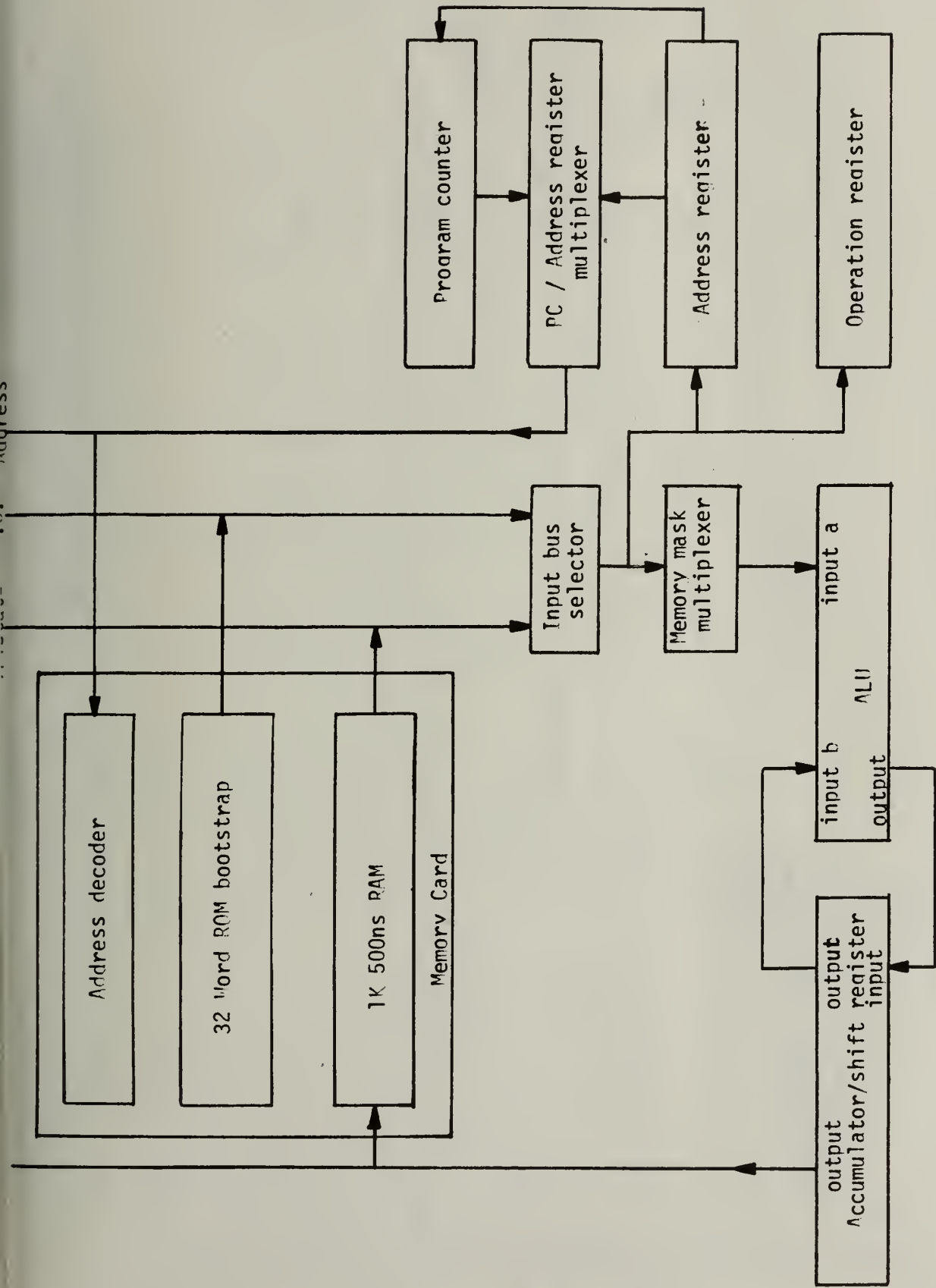


the low part of the address (using one of 8 bits for a 'switch'). A data routing card, therefore, can consist of an address recognizer, delays and gating. Another advantage of the data routing card is that a decoder can be used to select devices to drive the tristate bus. Bus conflicts are avoided as the decoder never selects two drivers at once.

BACK PANEL WIRING. To make the processor operational it is essential that a few of the back plane pins for the two card plug-in module be connected, as more than the twenty interconnecting wires are needed to electrically join the two cards. The essential list is given below. In the terminal wiring list, complete interconnections between data router cards and other devices are shown.

Signal	CPU Card	Memory card
Pause now	e	e
Write	f	5
A8	p	p
A9	n	n
A10	m	m
A11	l	l
A12	k	k
A13	j	j
A14	h	h
+5v	abc2	abc
Gnd	rvwxyz	12467

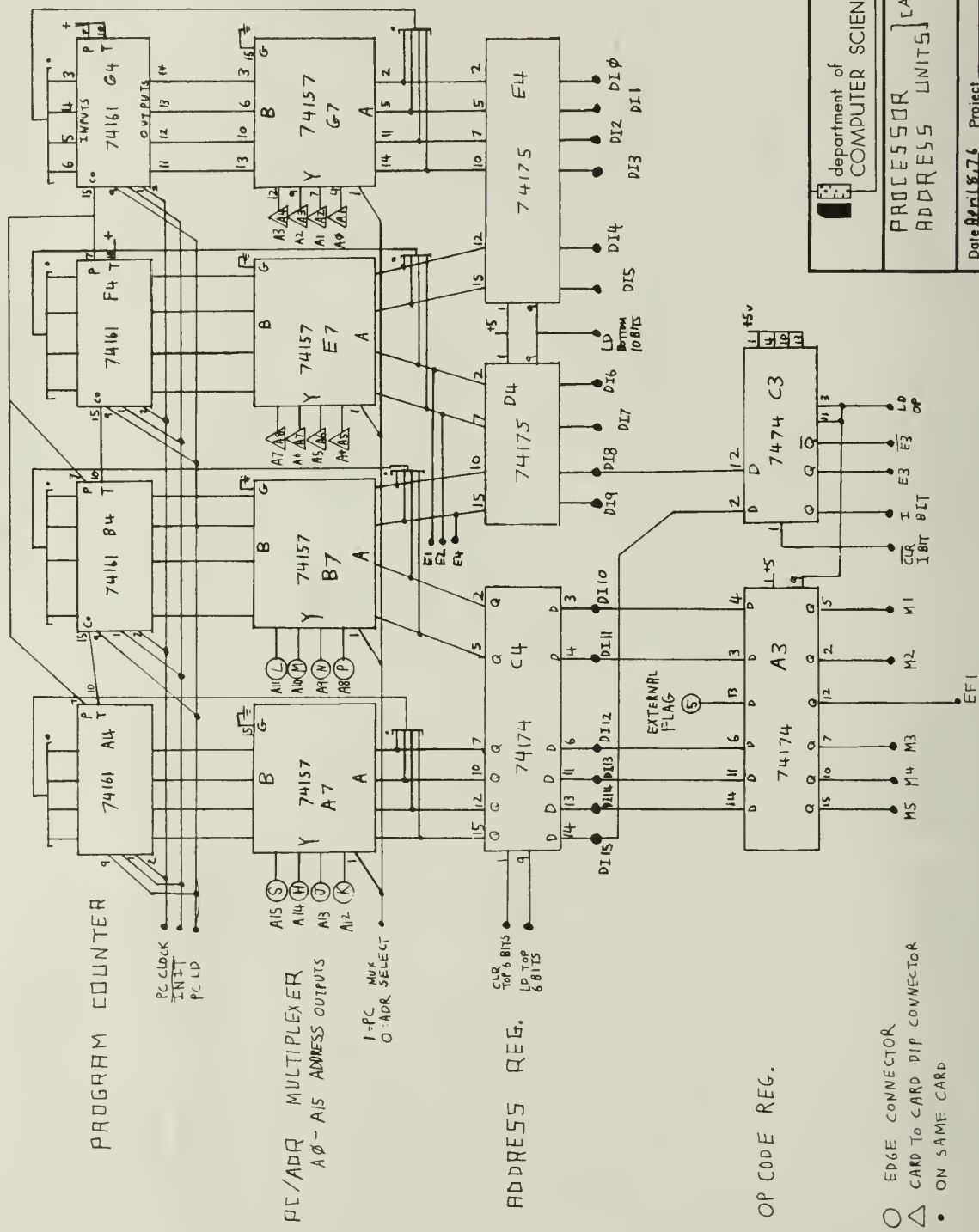
APPENDIX A  
DESIGN SCHEMATICS



Processor address and data unit structure.







OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE REG.

ADDRESS REG.

PC/ADR MULTIPLEXER

PROGRAM COUNTER

OP CODE



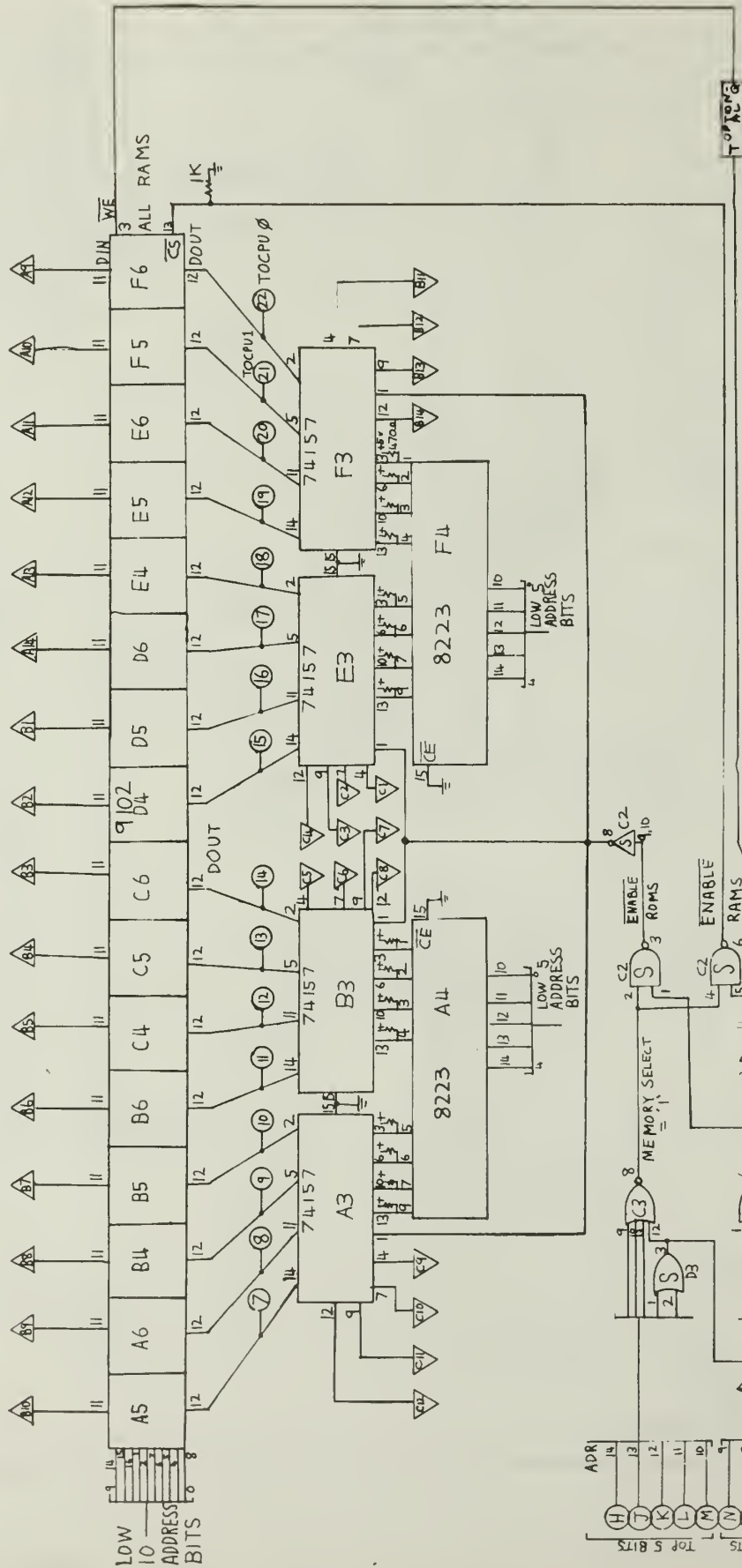




⑤ = Card Edge Connector  
 ▲ = Dip Socket Connector

+5v = 10  
 GND = 9

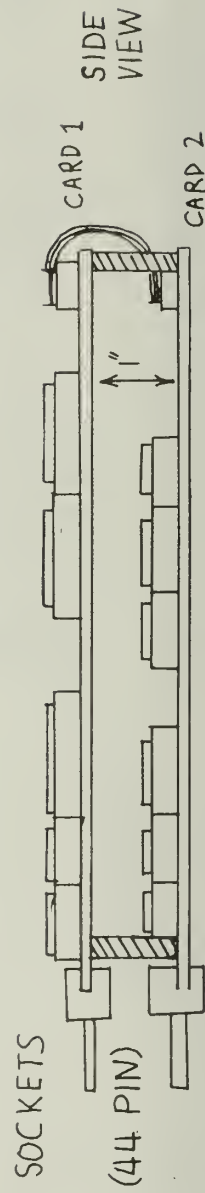
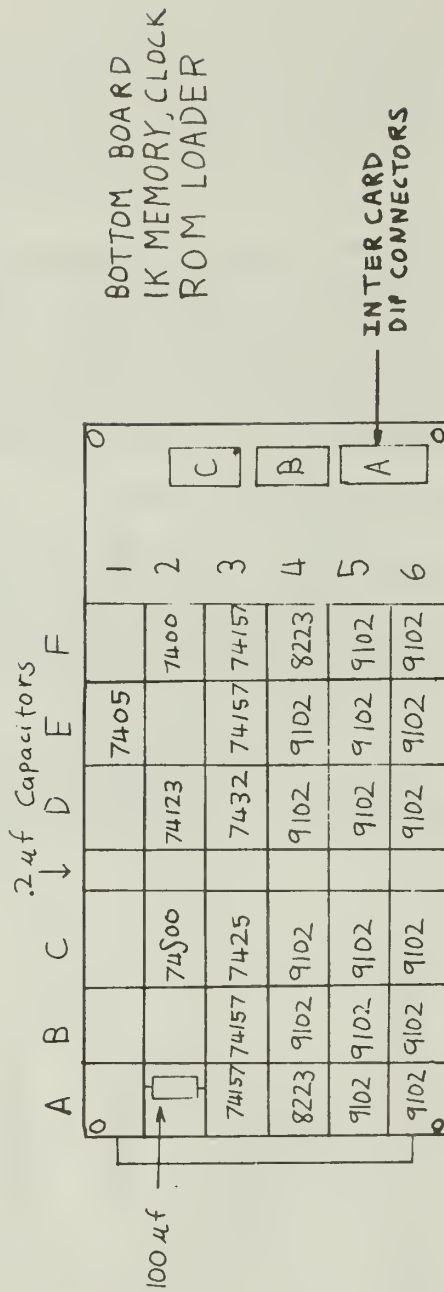
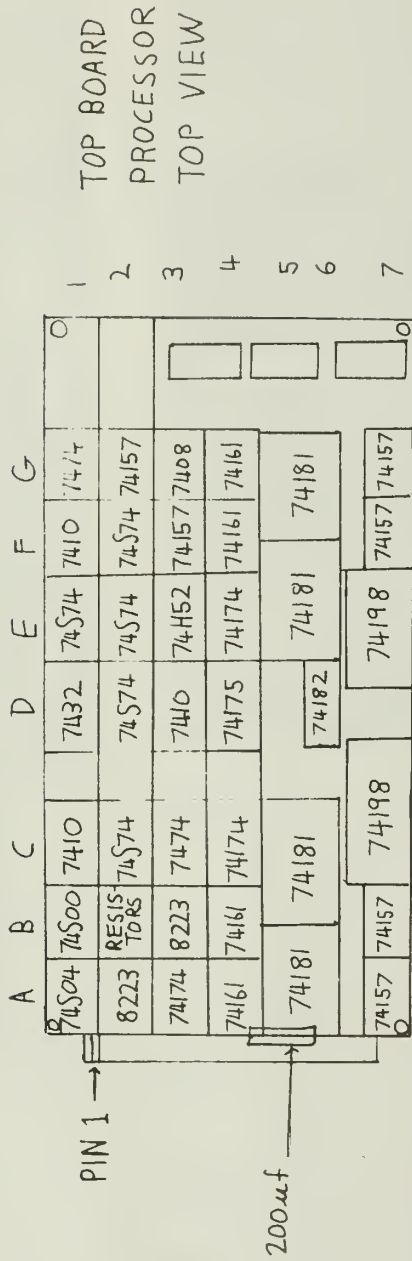
1K RAMS AM9102A



⑤ ③ ② ① ⑦ ⑥ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳ ㉑ ㉒ ㉓ ㉔ ㉕ ㉖ ㉗ ㉘ ㉙ ㉚ ㉛ ㉜ ㉝ ㉞ ㉟ ㊱ ㊲ ㊳ ㊴ ㊵ ㊶ ㊷ ㊸ ㊹ ㊺ ㊻ ㊼ ㊽ ㊾ ㊿

DEPARTMENT OF COMPUTER SCIENCE University of Illinois		TITLE PROCESSOR MEMORY CLOCK CARD2	
Fea.	Designed by B. H. Wick	Date	March 12, 76
Issue	Change order	Approved by	Reference
Revision	Revisions	Sheet 5 of	Drawing No.

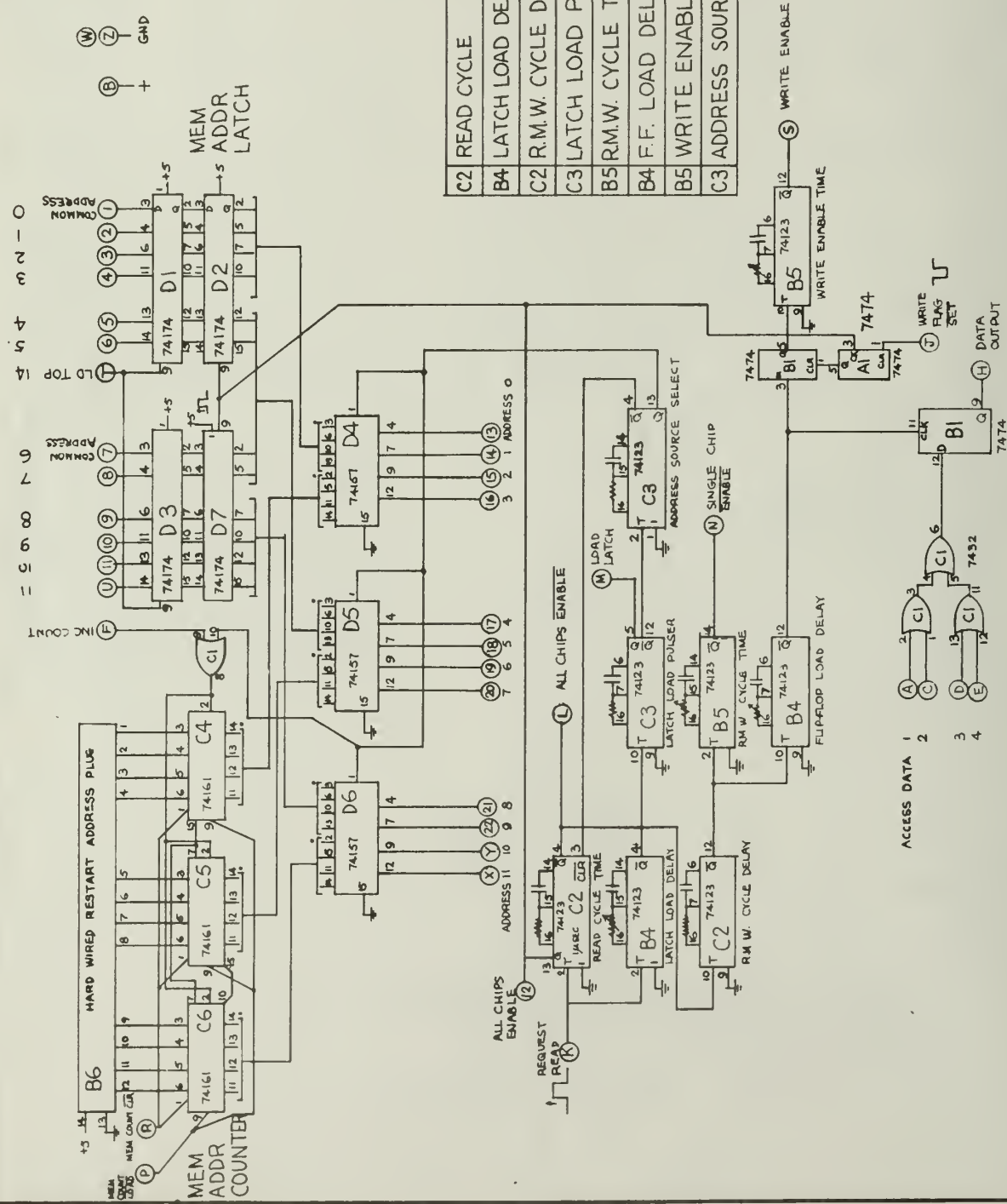




PROCESSOR LAYOUT







	1	2	3	4	5	6	7
A	74174	74174	74174	74174	74174	74174	74174
B	7432	74174	74123	74161	74123	74123	74161
C							
D							

C2	READ CYCLE	1μS	10K	270pf	FIXED
B4	LATCH LOAD DELAY	500nS	50K	51pf	VARIABLE
C2	R.M.W. CYCLE DELAY	250nS	9.1K	51pf	FIXED
C3	LATCH LOAD PULSER	40nS	4.7K	0pf	FIXED
B5	R.M.W. CYCLE TIME	1.1μS	50K	270pf	VARIABLE
B4	F.F. LOAD DELAY	500nS	50K	51pf	VARIABLE
B5	WRITE ENABLE	250nS	50K	30pf	VARIABLE
C3	ADDRESS SOURCE	1μS	10K	270pf	FIXED

ACCESS DATA 1 2 3 4

DEPARTMENT OF COMPUTER SCIENCE University of Illinois		TITLE SCREEN DISPLAY - CARD 2 SMALL CARD -	
Drawn by <b>BARTWICK KEN CHAN</b>	Date	Supervisor's sig	
Approved by	Date	Reference	
Issue Change order Approved by Date		REVISIONS	
Sheet 2 of 4		Drawing No.	



DATA TO LD\_XMIT  
BE WRITTEN REG.

DEVICE REGISTER  
BLOCK CODE '177'

DRIVE BUS  
CODE

LD LD SET  
TOP BOT  
8 10  
WRITE  
FLAG

ALLCHIP ENABLE

EXTERNAL  
FLAG 1

'0' = READY  
FOR  
WRITE

'1' = BUSY

A 8  
A 15  
A 14  
A 13  
A 12  
A 11  
A 10  
A 9

A 2

LOAD TOP 14 BITS

WRITE

LOAD BOT. 4 BITS

A 3

A 4

WRITE  
DATA

WRITE SET

A 0

TO CPU 15

DATA OUTPUT

LOAD UART

A 1

A 7 A 6 A 5

(M) (N) (P)

(7) (8) +5

(29) (14) (2)

KEY BD  
DRIVE  
BUS

UART  
DRIVE  
BUS

	D	C	B	A	
TOP OF CARD	7430				1
	74H11				2
	7410				3
	7402	7474			4
	7442				5
	8T09				6
	7474				7

DEPARTMENT OF COMPUTER SCIENCE  
University of Illinois

TITLE  
TERMINAL DATA  
ROUTER CARD

For: \_\_\_\_\_  
Drawn by: B. ARTWICK  
Date: \_\_\_\_\_  
Approved by: \_\_\_\_\_  
Date: \_\_\_\_\_  
Reference: \_\_\_\_\_

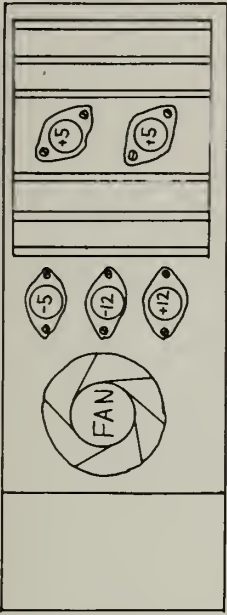
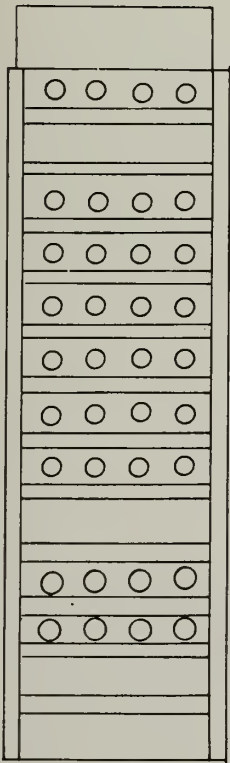
Sheet \_\_\_\_\_ of \_\_\_\_\_ Drawing No. \_\_\_\_\_





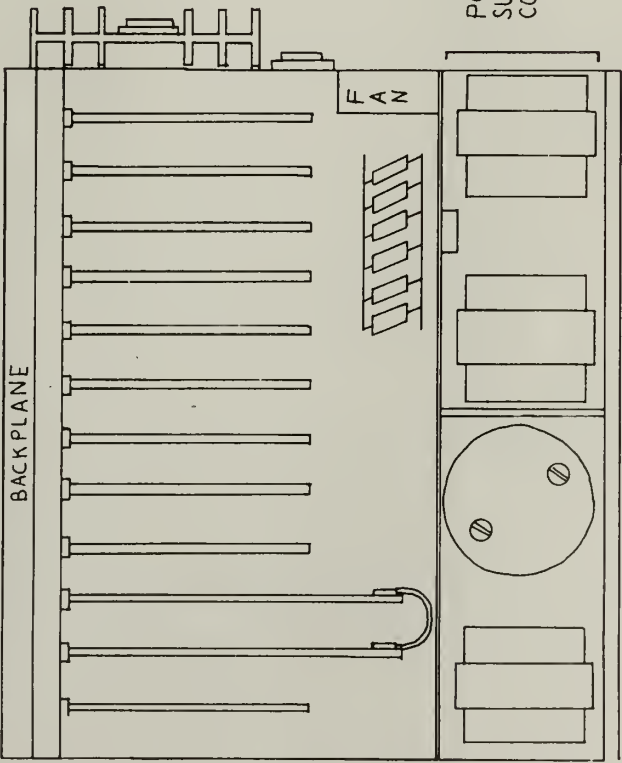


# BACKPANEL VENTILATION HOLES

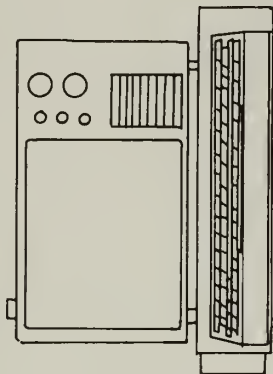


POWER SUPPLY PASS TRANSISTORS

UART CARD  
PROCESSOR  
PROCESSOR  
MEMORY  
DATA  
ROUTER  
MATRIX  
GENERATOR  
MEMORY  
TIMING  
SCREEN  
MEMORY  
40K  
40K  
40K  
40K  
EXPANSION  
POWER  
SUPPLY  
CARD



POWER  
SUPPLY  
COMPONENTS



GRAPHICS TERMINAL

DEPARTMENT of COMPUTER SCIENCE		TITLE	
University of Illinois		GRAPHICS TERMINAL	
For	Drawn by	Date	Submitted day
	B. ARTWICK	APRIL 7, 76	
Approved by	Date	Reference	Sheet of
			Drawing No.

REVISIONS			
Issue	Change order	Approved by	Date



APPENDIX B  
BACK-PANEL WIRING AND INTERCONNECTIONS

## PROCESSOR INTERCARD DIP CONNECTORS

A1 - Address	0	B1 - Accumout	6	C1 - Processor Input	4
A2 -	1	B2 -	7	C2 -	5
A3 -	2	B3 -	8	C3 -	6
A4 -	3	B4 -	9	C4 -	7
A5 -	4	B5 -	10	C5 -	8
A6 -	5	B6 -	11	C6 -	9
A7 -	6	B7 -	12	C7 -	10
A8 -	7	B8 -	13	C8 -	11
A9 - Accumout	0	B9 -	14	C9 -	12
A10-	1	B10-	15	C10-	13
A11-	2	B11- Processor Input	0	C11-	14
A12-	3	B12-	1	C12-	15
A13-	4	B13-	2	C13-	Not used
A14-	5	B14-	3	C14-	Fast Clock

Note: Although no microconsole has been built, the DIP connectors would be an excellent point to obtain signals for such a device.

## TERMINAL CINCH PLUG

This plug is wired to accomodate a keyboard, communications line, and an initialize/run switch.

Pin	Pin
1 - Ground	13 - Run
2 - Keyboard bit 1	14 - Ground
3 - 2	15 - Data into terminal
4 - 3	16 - Data from terminal
5 - 4	17 - Not used
6 - 5	18 -
7 - 6	19 -
8 - 7	20 -
9 - 8	21 -
10- 9	22 -
11- +5 Volts	23 -
12- Initialize	24 -
	25 -

# PROCESSOR CPU CARD PIN ASSIGNMENTS

Pin	Signal	Pin	Signal
1	tocpu7	a	+5
2	tocpu6	b	+5
3	tocpu5	c	+5
4	tocpu4	d	r
5	tocpu5	e	pause now
6	tocpu4	f	write
7	tocpu15	h	a14
8	tocpu14	j	a13
9	option bit	k	a12
10	kb3	l	a11
11	kb2	m	a10
12	kb1	n	a9
13	kb0	p	a8
14	init	r	ground
15	rec. drive	s	a15
16	kbd. drive	t	
17	uart output	u	ground
18	clock check	v	ground
19	write	w	ground
20	-12 volts	x	ground
21	ground	y	ground
22	ground	z	ground

## UART CARD PIN ASSIGNMENTS

Pin	Signal	Pin	Signal
1	tocpu7	a	+5
2	tocpu6	b	+5
3	tocpu5	c	tocpu3
4	tocpu4	d	tocpu2
5	tocpu5	e	tocpu1
6	tocpu4	f	tocpu0
7	option bit	h	kp
8	kb3	j	kb7
9	kb2	k	kb6
10	kb1	l	kb5
11	kb0	m	kb4
12	init	n	load uart
13	rec. drive	p	uart data in
14	kbd. drive	r	ac0
15	uart output	s	ac1
16	clock check	t	ac2
17	write	u	ac3
18	-12 volts	v	ac4
19	ground	w	ac5
20	ground	x	ac6
21		y	ac7
22		z	ground



# PROCESSOR MEMORY CARD PIN ASSIGNMENTS

1	gnd	a	+5
2	gnd	b	+5
3	stop clock	c	+5
4	gnd	d	pause now
5	write	e	
6	gnd	f	
7	tocpu15	h	a14
8	tocpu14	j	a13
9	tocpu13	k	a12
10	tocpu12	l	a11
11	tocpu11	m	a10
12	tocpu10	n	a9
13	tocpu9	p	a8
14	tocpu8	r	a7
15	tocpu7	s	a6
16	tocpu6	t	a5
17	tocpu5	u	a4
18	tocpu4	v	a3
19	tocpu3	w	a2
20	tocpu2	x	a1
21	tocpu1	y	a0
22	tocpu0	z	gnd

# TERMINAL DATA ROUTER CARD PIN ASSIGNMENTS

1	a2	a	a8
2	load top 14 bits	b	+5
3	load bottom 4	c	a15
4	a3	d	a14
5	a4	e	a13
6	write data	f	a12
7	write set	h	a11
8	a0	j	a10
9	write	k	a9
10	tocpu15	l	load xmit reg.
11	data output	m	a7
12	a1	n	a6
13		p	a5
14	gnd	r	kbd drive bus
15	ld buf	s	receive drive bus
16		t	+5
17		u	
18		v	
19		w	
20		x	
21		y	
22	gnd	z	gnd

# MATRIX GENERATOR CARD PIN ASSIGNMENTS

## MEMORY CONTROL CARD PIN ASSIGNMENTS

1	card address 0 (ac0)	a	card 4 c	1	common adr. 0 (ac2)	a	access data 1
2	card address 1 (ac1)	b	+5	2	common adr. 1 (ac3)	b	+5
3		c	card 4 d	3	common adr. 2 (ac4)	c	access data 2
4		d	card 3 c	4	common adr. 3 (ac5)	d	access data 3
5	single chip enable	e	card 3 d	5	common adr. 4 (ac6)	e	access data 4
6	chip address d (ac3)	f	card 2 c	6	common adr. 5 (ac7)	f	inc count
7	chip address c (ac2)	h	card 2 d	7	common adr. 6 (ac8)	h	data output
8	chip address b (ac1)	j	card 1 c	8	common adr. 7 (ac9)	j	write flag set
9	chip address a (ac0)	k	card 1 d	9	common adr. 8 (ac10)	k	request read
10	a	l	load bottom 4	10	common adr. 9 (ac11)	l	all chips enable
11	b	m	-12 volts	11	common adr. 10 (ac12)	m	load latch
12	bit clock	n	load top 14	12	sync load	n	single chip enable
13	video output	p	gnd	13	address 0	p	mem count load
14	request read, load sr	r	sync load	14	address 1	r	mem count clr
15	mem count clr	s	h enable (stop clock)	15	address 2	s	write enable
16	mem count load	t	gnd	16	address 3	t	ld top 14
17	inc count	u		17	address 4	u	common adr. 11 (ac13)
18	display data	v		18	address 5	v	ld buf
19	clr sr	w	+5	19	address 6	w	gnd
20	+5	x		20	address 7	x	address 11
21		y		21	address 8	y	address 10
22	gnd	z	gnd	22	address 7	z	gnd

## VOLTAGE REGULATOR CARD PIN ASSIGNMENTS

1	gnd	a	gnd
2	gnd	b	gnd
3	gnd	c	gnd
4	gnd	d	gnd
5	gnd	e	gnd
6	gnd	f	gnd
7	gnd	h	gnd
8	gnd	j	gnd
9	gnd	k	gnd
10	gnd	l	gnd
11	-12	m	gnd
12	-12	n	+5
13	-5	p	+5
14	-5	r	+5
15	+12	s	+5
16	+12	t	+5
17	+12	u	+5
18	+12	v	+5
19	+12	w	+5
20	+12	x	+5
21	+12	y	+5
22	+12	z	+5

## SCREEN MEMORY CARD PIN ASSIGNMENTS

1	write data	a	+5
2	address 0	b	-5
3		c	+12
4	address 1	d	all chips enable
5	address 2	e	a
6		f	b
7	address 3	h	c
8	address 4	j	d
9		k	gnd
10	address 5	l	clr sr
11	address 6	m	load sr
12	+5	n	bit clock
13	address 7	p	data chain in 1
14	address 8	r	load latch
15		s	gnd
16	address 9	t	
17	address 10	u	
18		v	
19	address 11	w	
20	+5	x	access data 1
21	write enable	y	display data
22		z	gnd

GRAPHICS TERMINAL. The basic terminal system consists of the following.

1. UART communication card
2. Processor card
3. 1K processor memory
4. Data routing card
5. Screen display matrix generator
6. Screen display memory timing
- 7 to 9. 1 to 4 cards of screen memory
10. Power supply driver card

The wiring list on the following pages follows the numbering conventions above. Extra cards may be included between cards 9 and 10 if more peripherals or memory are desired.

## 22 pin socket for card:

signal	1	2	3	4	5	6	7	8	9	10
ac0	r	22			1,9					
ac1	s	21			2,8					
ac2	t	20			7	1				
ac3	u	19			6	2				
ac4	v	18				3				
ac5	w	17				4				
ac6	x	16				5				
ac7	y	15				6				
ac8		14				7				
ac9		13				8				
ac10		12				9				
ac11		11				10				
ac12		10				11				
ac13		9				u				
ac14		8								
ac15		7								
kp	h									
kb0	12									
kb1	11									
kb2	10									
kb3	9									
kb4	m									
kb5	l									
kb6	k									
kb7	j									
clock check	18									
+5	a,b	a,b	a,b	b,t	b,w	v,b	13,21	13,21	13,21	13,21
+5		c,2	c		20		b	b	b	b
+12							d	d	d	d
-5							c	c	c	c
-12	20				m					
s		l								
r		d								
pause now		e	e							
write	19	f	5	9						
stop clock			3		s					
load top 14				2	l	t				
load bot. 4				3	n					
write data				6			1	1	1	1
write set				7		j				
a0			y	8						
a1			x	12						
a2			w	1						
a3			v	4						
a4			u	5						
a5			t	p						

signal	1	2	3	4	5	6	7	8	9	10
a6			s	n						
a7			r	m						
a8		p	p	a						
a9		n	n	k						
a10		m	m	j						
a11		l	l	h						
a12		k	k	f						
a13		j	j	e						
a14		h	h	d						
a15				c						
tocpu0	f		22							
tocpu1	e		21							
tocpu2	d		20							
tocpu3	c		19							
tocpu4	4		18							
tocpu5	3		19							
tocpu6	2		16							
tocpu7	1		15							
tocpu8			14							
tocpu9			13							
tocpu10			12							
tocpu11			11							
tocpu12			10							
tocpu13			9							
tocpu14	6		8							
tocpu15	5		7	10						
ground	21,22	r,v	1,2	14,22	22,p	w,z	l,t	l,t	l,t	l,t
ground	z	w,x	4,6	z	z		z	z	z	z
ground		y,z	z							
option bit	8									
init	13	3								
receive drive										
bus	15			s						
keyboard										
drive bus	16			r						
load uart	n			l						
uart data in	p									
data output				11		h				
single chip										
enable					5	n				
a					10		f	f	f	f
b					11		h	h	h	h
bit clock					12		p	p	p	p



signal	1	2	3	4	5	6	7	8	9	10
video output					13					
request read										
and load sr					14	k	n	n	n	n
mem count clr					15	r				
mem count load					16	p				
inc count					17	f				
display data					18		y			
clr sr					19		m	m	m	m
card 1 c					j		j			
card 1 d					k		k			
card 2 c					f			j		
card 2 d					h			k		
card 3 c					d				j	
card 3 d					e				k	
card 4 c					a					j
card 4 d					c					k
address0						13	3	3	3	3
address1						14	5	5	5	5
address2						15	6	6	6	6
address3						16	8	8	8	8
address4						17	9	9	9	9
address5						18	11	11	11	11
address6						19	12	12	12	12
address7						20	14	14	14	14
address8						21	15	15	15	15
address9						22	17	17	17	17
address10						y	18	18	18	18
address11						x	20	20	20	20
access data 1						a	x			
access data 2						c		x		
access data 3						d			x	
access data 4						e				x
all chips enabled						l	e	e	e	e
load latch						m	s	s	s	s
write enable						s	22	22	22	22
data chain in 1							r	y		
data chain in 2								r	y	
data chain in 3									r	y
sync load					r	12				

APPENDIX C  
CONSTRUCTION TECHNIQUES  
AND  
ADDENDUM SHEETS

CONSTRUCTION TECHNIQUES. Prototype development has shown that the following methods produce the most reliable processor.

- 1) All sockets should be firmly soldered, screwed or glued to the circuit cards.
- 2) Adequate decoupling should be provided by using one or more .2 uf capacitors per each three integrated circuits.
- 3) A 200 uf filter capacitor should be provided on the processor board for line regulation.
- 4) 1K pull-up resistors are needed on the decoding ROM. Vertical, dual inline mounting was found to be the best method of mounting these resistors.
- 5) The ROM codes are shown in the design schematics. Proper ROM programming techniques should be used.
- 6) Many ground and positive voltage pins are allocated on the processor and memory cards to assure uniform power distribution.
- 7) DIP plugs and sockets provide reliable intercard connections.
- 8) All leads going to MOS inputs must be properly loaded with 470 ohm resistors to prevent ringing.
- 9) All unused inputs should be clamped to positive or ground to prevent oscillations.

## RECOMMENDED DEBUGGING TECHNIQUES

The most efficient way to get the processor operational is to build one completely and step instructions through their cycles with a manual clock. Instructions can be fed into the processor through the intercard dip connectors and the accumulator may be watched. The initialization flip flop and everything it should initialize should be checked first.

It was found that monitoring the whole accumulator with LEDs was helpful. For sequencer checkout, wire wrapping wires to the state outputs and driving labeled LEDs is most efficient. The whole PC and address register may be watched with LEDs but it was found that the lower four address bits of the address multiplexer were good enough, especially in later debugging stages.

If careful wiring practices are observed, debugging should not take long. If the whole system seems dead, check for:

- 1) Power and polarity wiring at the ICs
- 2) All enable, presets and clears at proper levels
- 3) All ICs in proper direction, location and all pins in sockets.

Case 2 is the usual problem. If addressing seems to do strange things, the lookahead connections on the counters should be checked.

## LOADING PROGRAMS UNDER BOOTSTRAP

## VERSIONS ONE AND TWO

Programs are loaded into the processor memory from an external source such as the controlling computer. A bootstrap program is stored in the lower 32 words of memory to perform this initial program loading.

Upon processor initialization, the program counter of the processor is set to location zero and the bootstrap program performs the initial program load. Versions one and two first send a character (a ! character) to the controlling processor which signal a request to be loaded. The controlling computer then sends the program to be loaded as a string of characters. The only difference between bootstrap version one and two is in the way words are composed from the eight bit characters in the input string. Version one assumes all 8 bits are valid and packs two characters per word while version 2 packs three characters per word so a 6 bit loading format can be used.

The first two words received (4 or 6 characters) specify the program's length and address to be loaded at. The remaining words are loaded into processor memory until the program length is reached. The last program instruction is then executed. The last instruction, therefore, acts as a "jump to program start point" instruction. Program loading must not write in the ROM area (address 0 to 31) and the length word (word number one in the loading process) is the total of all words loaded, including the address and length word. The address word (word two of the load) is the load start point minus one.

The following program is bootstrap version two.

```

        org 000000
*   cheap 1 bootstrap loader
        loadm 0005
        st i atrans
        l     adr
        st    ptr
nxtwrđ  loadm 1775
        st    bytct
nxtbyt  l     temp
        ls2
        ls2
        ls2
        st    temp
wait    l i   arcvr
        bpz   wait
        st i  arcvr
        andm 0077
        or s  temp
        incs  bytct
        bm    nxtbyt
        l     temp
        st i  ptr
        decs  ctr
        bmz i ptr
        incs  ptr
        b     nxtwrđ
fill    000000
fil2    000000
fil3    000000
fil4    0
fil5    0
adr      001776
arcvr    177440
atrans   177402
temp     000000
ctr       000000
ptr       000000
bytct    000000
end

```

Figure C-1. Bootstrap version 2.



## POWER AND GROUNDING

To insure a reliable system it is most important that good power bussing and decoupling techniques be followed. Multiple ground and power pins on circuit cards, many small decoupling capacitors, heavy power backplane wiring and heavy ground planes on the cards are all important. The 40K RAM cards were found to be extremely sensitive to poor grounding conditions.

Noisy ground levels and uneven power supply levels produce strange, data sensitive, intermittent problems which are extremely difficult to narrow down to their real cause. For a high speed device such as this terminal, excellent grounding is a must.

U. S. ATOMIC ENERGY COMMISSION  
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR  
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

( See Instructions on Reverse Side )

1. AEC REPORT NO.

C00-2383-0033

2. TITLE

GRAPHICS ONE TERMINAL DESIGN AND USER'S MANUAL

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report

☐ b. Conference paper not to be published in a journal:

Title of conference \_\_\_\_\_

Date of conference \_\_\_\_\_

Exact location of conference \_\_\_\_\_

Sponsoring organization \_\_\_\_\_

☐ c. Other (Specify) \_\_\_\_\_

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.

☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.

☐ c. Make no announcement or distribution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

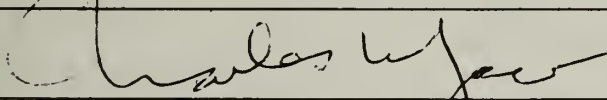
C. W. Gear

Professor and Principal Investigator

Organization

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, IL 61801

Signature



Date

July 1976

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

☐ a. AEC patent clearance has been granted by responsible AEC patent group.

☐ b. Report has been sent to responsible AEC patent group for clearance.

☐ c. Patent clearance not required.



<b>BIBLIOGRAPHIC DATA SHEET</b>	1. Report No. UIUCDCS-R-76-814	2.	3. Recipient's Accession No.
4. Title and Subtitle  GRAPHICS ONE TERMINAL DESIGN AND USER'S MANUAL		5. Report Date July 1976	
7. Author(s) BRUCE A. ARTWICK AND ALFRED D. WHALEY		6.	
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801		8. Performing Organization Rept. No. UIUCDCS-R-76-814	
		10. Project/Task/Work Unit No. C00-2383-0033	
		11. Contract/Grant No. US ERDA E(11-1) 2383	
12. Sponsoring Organization Name and Address United States Energy Research and Development Administration 9800 South Cass Avenue Argonne, IL 60439		13. Type of Report & Period Covered	
15. Supplementary Notes		14.	
16. Abstracts  Described is a graphics terminal using raster scan on an ordinary TV monitor. The scan is done out of a bank of 4K dynamic rams which are set or reset asynchronously by an independent microprocessor. Special features are the microprocessor built out of ordinary 7400 series parts which does all graphics, text and communications and the method for storing the raster image in rams.			
17. Key Words and Document Analysis. 17a. Descriptors  Graphics terminal                      microprocessor microprogramming                  computer graphics raster scan                            programmable terminal programmable terminal			
17b. Identifiers/Open-Ended Terms			
17c. COSATI Field/Group			
18. Availability Statement  Unlimited		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 71
		20. Security Class (This Page) UNCLASSIFIED	22. Price



DEC 15 1976

















UNIVERSITY OF ILLINOIS-URBANA



3 0112 039572828