Contract Number NAS7-1152

P- 114

Ultra-Dense Magnetoresistive Mass Memory

Fifth Quarter Report

July 15, 1992 through October 15, 1992

October 30, 1992

For

Chief, Procurement Office
NASA Resident Office-JPL
4800 Oak Grove Drive
Pasadena, Ca 91109
M/S 180-805

Nonvolatile Electronics, Incorporated
12800 Industrial Park Blvd.
Plymouth, Minnesota 55441-3928

This page intentionally blank

TABLE OF CONTENTS

SECTION			PAGE
1	OVERVIEW		1-1
	1.1	Introduction	1-2
	1.2	Expenditures	1-2
	1.3	Fifth Quarter Accomplishments	1-2
	1.4	Goals for Next Quarter	1-3
	1.5	Overview of the Following Sections	1-4
2	WAFER CONTROL SYSTEM DESIGN		2-1
	2.1	Introduction	2-2
	2.2	Stand Alone Reliability 16K Macrocell Tester	2-2
	2.3	Software Operation of the 16K Macrocell tester	2-2
	2.4	Software Design of the 16K Macrocell Tester	2-3
3	WAFER BUS DESIGN		3-1
	3.1	Introduction	3-2
4	Test Chip		4-1
	4.1	Introduction	4-2
	4.2	Test Chip Design	4-2
	4.3	Processing	4-8
	4.4	Conclusion	4-8
5	1-MBIT STATUS REPORT		5-1
	5.1	Introduction	5-2
	5.2	Parasitic Equivalent of Unused Sense Lines in the 1 Megabit array	5-3
	5.3	MRAM Bit Specifications	5-11
	5.4	System Timing	5-12
	5.5	1 Megabit Chip Architecture	5-13
	5.5.1	2K Memory Segment	5-13
	5.6	Conclusion	5-16
	5.7	Appendix	5-16

This page intentionally blank

SECTION 1

OVERVIEW

1.1 Introduction

This report details the progress and accomplishments of Nonvolatile Electronics, Inc., on the design of the wafer scale MRAM mass memory system during the fifth quarter of the project. NVE has made significant progress this quarter on the one megabit design in several different areas. A test chip, which will verify a working GMR bit with the dimensions required by the 1 Meg chip, has been designed, laid out, and is currently being processed in the NVE labs. This test chip will allow electrical specifications, tolerances, and processing issues to be finalized before construction of the actual chip thus providing a greater assurance of success of the final 1 Meg design. A model has been developed to accurately simulate the parasitic effects of unselected sense lines. This model gives NVE the ability to perform accurate simulations of the array electronics and test different design concepts. Much of the circuit design for the 1 Meg chip has been completed and simulated and these designs are included in this report. Progress has been made in the Wafer Scale design area to verify the reliable operation of the 16K macrocell. This is currently being accomplished with the design and construction of two stand alone test systems which will perform life tests and gather data on reliability and wearout mechanisms for analysis.

1.2 Expenditures

During the fifth quarter portion of this program, June 29, 1992 through September 27, 1992, NVE spent $77,489. Cumulative expenditures since the start of the program total $282,539 through September 27, 1992.

1.3 Fifth Quarter Accomplishments

This program's fifth quarter led to important progress in all areas of the ultra-dense mass memory design. What follows is a synopsis of this quarter's accomplishments:

1. Wafer Controller - The wafer controller design is essentially done on paper and is on hold until the 16K macrocell is finalized by Honeywell. This was expected to be accomplished last quarter but was again delayed

due to manufacturing problems. It is hoped that these problems will be resolved next quarter and construction of the controller can commence.

Support efforts for the 16K macrocell continued to see excellent progress in the reliability and test area. Two stand alone test systems were designed and constructed which will allow long term cycling of the chip without operator attention. They will be used to gather data on reliability and wearout mechanisms . The test systems operate continuously and are immune to power failures. Results of the life tests will be reported in future quarterly reports

2. Wafer Bus Design - The wafer bus design has been finalized in concert with the design improvements to NVE's 16K MRAM chip. NVE expects to have the masks for the wafer bus made as soon as wafers with the improved 16K part become available from Honeywell SSEC.

3. Test Chip - It was decided that a test chip incorporating the MRAM bit to be used by the 1 Meg design would be desirable in order to insure success of the project. Having a working GMR bit with the dimensions required by the 1 Meg would allow electrical specifications and tolerances to be finalized with confidence. Also any processing issues affecting the layout of the design would become apparent and could be addressed. The test chip has been designed and laid out and is currently being processed in the NVE lab.

4. One Megabit Macrocell Design - Excellent progress was made this quarter in the design of the 1 Meg chip. The core of the chip consisting of the MR Bits, sense lines, decoders, drivers, etc. has been designed and laid out. Simulations have been implemented on these circuits and a model of the unused sense line parasitics has been constructed. These efforts are necessary to have a good working chip the first time though wafer processing. Decision were made to finalize design rules, the MRAM bit specifications, timing specifications, and chip architecture.

1.4 Goals for Next Quarter

NVE has established the following goals for the end of the sixth quarter of this project:

1.) Acquire a wafer of the improved 16K MRAM chips from Honeywell SSEC, have the wafer bus masks manufactured, and be prepared to do the bus line deposition on this wafer by the start of the seventh quarter of the program.

2.) Finalize the circuit design of the 1 Meg MRAM macrocell, design the remaining support circuits, simulate, and begin final layout modifications. Conduct design reviews on final circuits and layout.

3.) Design the mechanical layout of the Wafer Scale controller and layout the circuit boards

4.) Complete the fabrication of the test chip, test, and verify advanced circuit concepts.

By the end of the sixth quarter, NVE will be nearly finished with the actual design and layout of the one megabit macrocell.

1.5 Overview of the Following Sections

This report is organized into the following four sections: Wafer Control System Design, Wafer Bus Design, Test Chip, and One Megabit Status. The first two sections deal with progress NVE has made on the prototype wafer scale mass memory, or demonstration vehicle. The Wafer Control System Design section details the progress made on the top level system architecture and the Wafer Bus Design section discusses the progress on the busing scheme that NVE will employ to carry power and signals across the wafer

The Test Chip section discusses a test vehicle used to evaluate a wide variety of bit configurations in order to fully characterize an MRAM bit manufactured with GMR materials..

The last section of this report, One Megabit Status, shows the progress NVE has made on the design of a larger macrocell which would be used on the wafer scale finished product. It is interesting to note that one of these larger macrocells would be equivalent to the entire wafer NVE is developing for the demonstration vehicle, and that the finished product, using these larger macrocells and a six inch wafer, would result in approximately 256 megabits of memory on a single wafer.

SECTION 2

WAFER CONTROL SYSTEM DESIGN

2.1 Introduction

A control system for the 16K macrocell MRAM wafer scale memory array has been designed with many important features required for the successful operation of this array and other larger MRAM arrays. This system has been designed to allow the wafer scale memory array to be connected directly to the IBM PC ISA (Industry Standard Architecture) bus with the capability of operating at main memory speeds. Many applications currently exist in the memory card and hard disk replacement market for a device such as this. Construction of the hardware will begin during this quarter provided that there are good working 16K parts from Honeywell.

Two stand alone test systems have been designed and built to test the long term reliability of the 16K macrocells. These test systems will also be used to test the 1Meg cells when they are completed. Since these systems are software controlled, it is a simple matter to modify their operation to test various features and different configurations.

2.2 Stand Alone Reliability 16K Macrocell Tester

The schematic diagram and board layout of the macrocell reliability tester are show in Figures 2-1 and 2-2. The hardware consists of a Dallas Semiconductor DS5000 microprocessor with 32K of battery backed SRAM. This SRAM contains the 16K failure bit map as well as the registers needed to maintain operation in the event of a power failure. Communication with the testers is accomplished via a serial RS232 connection operating at 4800 baud. Operation of the tester can be interrupted by plugging a serial cable into the tester and hitting a carriage return. The status can then be read out and the tester restarted. If power is lost or the tester is unplugged, it will maintain the error bit map and will continue the test when the power is restored.

2.3 Software operation of the 16K Macrocell Tester

The Program is a command driven program, accepting commands from the serial port. The serial port is configured to use 4800 baud, 8 bits, no stop bits, no parity.

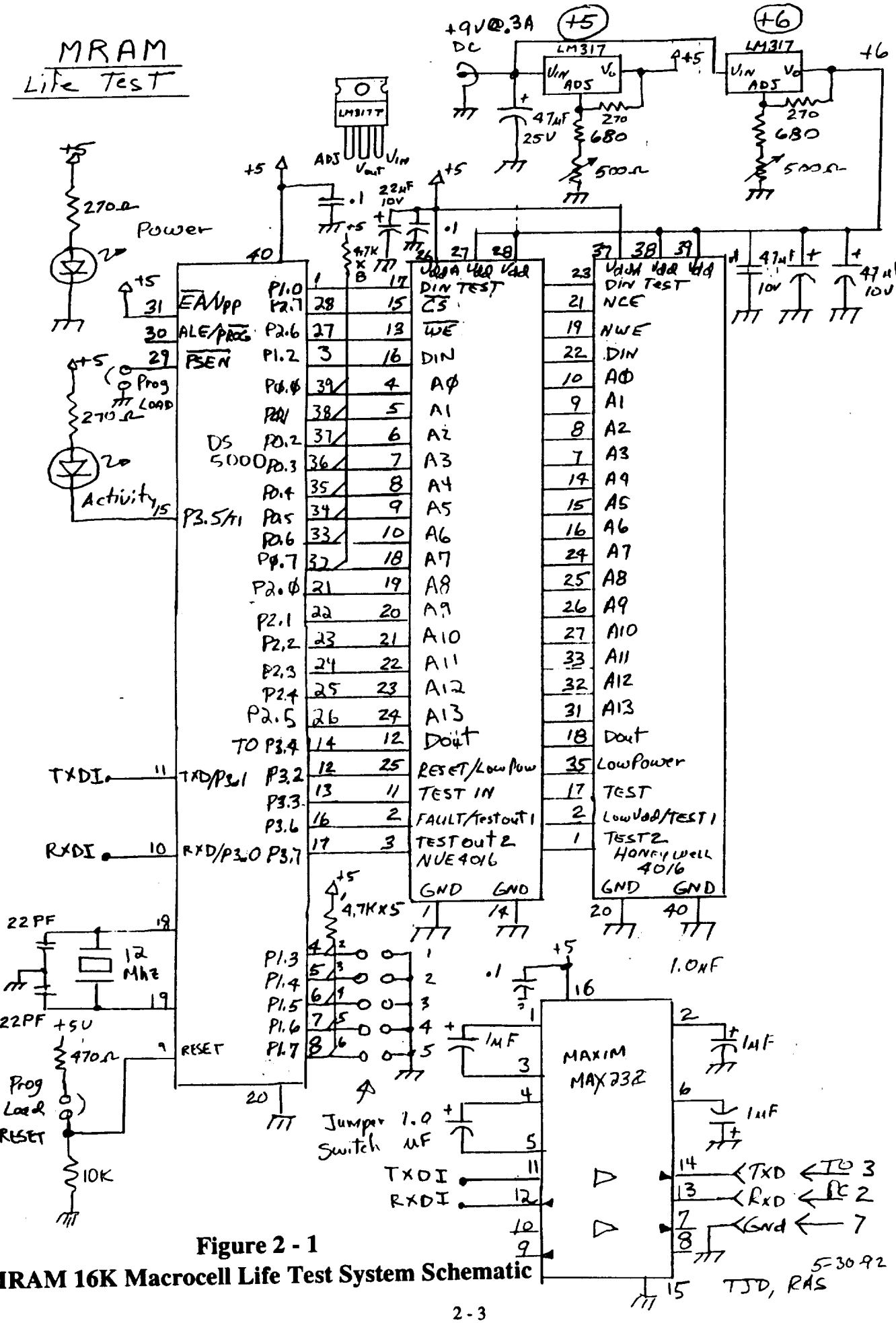


Figure 2 - 1

MRAM 16K Macrocell Life Test System Schematic

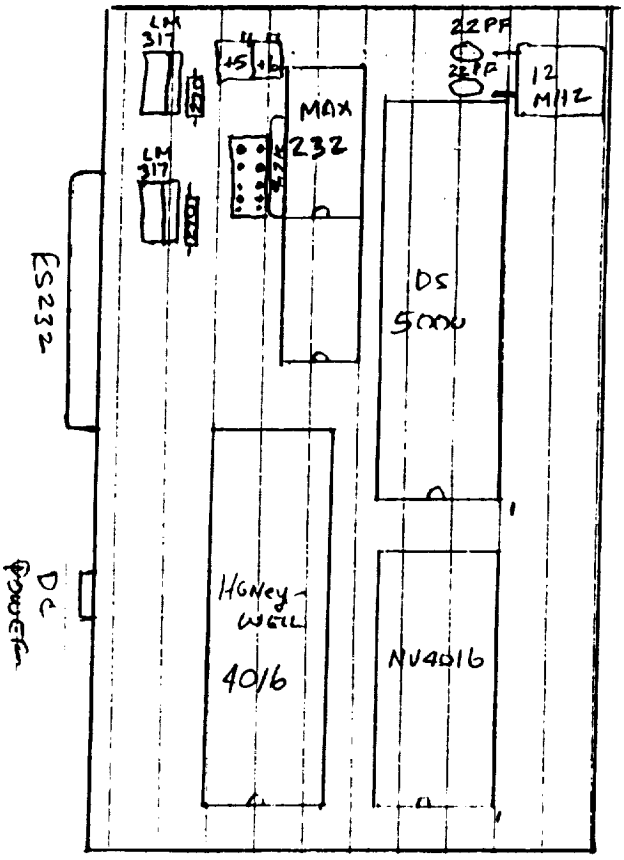


Figure 2 - 2
MRAM 16K Macrocell Life Test System Assembly Drawing

Upon reset, the program initializes the serial port, then continuously polls it, waiting for a command from the master PC. The command is an ASCII byte. Current commands range from "1" to"F". All command letters must be in uppercase. When a command is received, it is executed, and the software waits for the next command. Some commands expect additional input data bytes, and some respond by transmitting data back to the master PC.

The following is a list of all current commands:

Master Command	Returns	Description
1AAAA	D	Read Bit At Hex Address AAAA; Returns 1 or 0
2AAAAD		Write Bit At AAAA to D; No Return
3DDDDDDDDDDDDDDDDDDD (HEX)		Load Shift Register with Data D
4		Activate Shift Register
5		Deactivate Shift Register
6NN	EEEE AAAAAAAA	Do Walking 1-0 Test NN Times (Returns four digit hex number of failing bits and 8 digit loop count). If NN = 00, loop until carriage return.
7AAAA	EEEE	Do Pulse 1-0 Test on AAAA 16,384 Times (Returns four digit number of failures)
8	AAAA	Dump Bad Bit Addresses
9NN	EEEE	Do a Write 1, Read, Write 0, Read on RAM NN Times - Returns number of failing bits
ANND		Write Address NN with data byte D
BNN	D	Read Address NN ; returns data byte D
CNND		Write Address NN with data byte D into 5 locations to allow error correction.
DNN	D	Read Address NN with error correction
E -- <2048 Bytes>		Program 16K with incoming bit data (Bit 7 - Start) 2048 bytes = 16K bits
F -- <2048 Bytes>		Read 16K Contents ; Returns 2048 data bytes
G -- <40 Bytes>		Read Address 01 to 41 with error correction

Ram Memory Usage

R0-R7 used in most routines
Locations 20H TO 32H -- Stores shift register ASCII codes
Locations 4000H TO 7FFFH -- Used in RAM testing for a bad bit map.
The program requires 32K of RAM to operate.
Flag Area -- 3FF0 TO 3FFF

P0.0	Pin 39	A0	P1.0	Pin 1	DIN TEST	P2.0	Pin 21	A8
P0.1	Pin 38	A1	P1.1	Pin 2	Not Used	P2.1	Pin 22	A9
P0.2	Pin 37	A2	P1.2	Pin 3	DIN	P2.2	Pin 23	A10
P0.3	Pin 36	A3	P1.3	Pin 4	Jumper 1	P2.3	Pin 24	A11
P0.4	Pin 35	A4	P1.4	Pin 5	Jumper 2	P2.4	Pin 25	A12
P0.5	Pin 34	A5	P1.5	Pin 6	Jumper 3	P2.5	Pin 26	A13
P0.6	Pin 33	A6	P1.6	Pin 7	Jumper 4	P2.6	Pin 27	/WE
P0.7	Pin 32	A7	P1.7	Pin 8	Jumper 5	P2.7	Pin 28	/CS

P3.0	Pin 10	TXD
P3.1	Pin 11	RXD
P3.2	Pin 12	LOW POWER
P3.3	Pin 13	TEST IN
P3.4	Pin 14	DOUT
P3.5	Pin 15	Activity Light
P3.6	Pin 16	Fault/Test Out 1
P3.7	Pin 17	Test Out 2

2.4 Software Design of the 16K Macrocell Tester

The program for the 16K Macrocell Tester was written in 80C32 assembler language for speed and efficiency. Using modular routines allows ease of reconfiguration. The listing is shown in the following section.

2.3.1 Software Listing For MRAM Life Tester

```
The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  1
                                06-18-92


;-------------------------------------------
;8051 Assembly Test Program for THE NV4016 Chip
;August 21 1991
;Tim Dupuis
;Updated June 18 1992 RAS
;-------------------------------------------

;-------------------------------------------
;Basic Structure:
;The Program is a command driven program, accepting commands
;from the serial port.  The serial port is configured to  use
;2400 baud, 8 bits, no stop bits, no parity.  It is capable
;of running at 4800 baud, but was purposely slowed down to ease
;communication with the slow Compaq Portable PC.
;
;Upon reset, the program initializes the serial port, then
;continuously polls it, waiting for a command from the master PC.
;The command is a ASCII byte.  Current commands range from "1" to
;"F".  All command letters must be in uppercase.  When a command
;is received, it is executed, and the software waits for the next
;command.  Some commands expect additional input data bytes, and some
;respond by transmitting data back to the master PC.
;
;The following is a list of all current commands:
;Master Command      Returns
;1AAAA          D     Read Bit At Hex Address AAAA; Returns 1 or 0
;2AAAAD               Write Bit At AAAA to D; No Return
;3DDDDDDDDDDDDDDDDDDD      Load Shift Register with Data D (HEX)
;4                    Activate Shift Register
;5                    Deactivate Shift Register
;6NN            EEEE  Do Walking 1-0 Test NN Times (Returns
;                     four digit hex number of failing bits)
;7AAAA          EEEE  Do Pulse 1-0 Test on AAAA 16,384 Times
;                     (Returns four digit number of failures)
;8              AAAA  Dump Bad Bit Addresses
;9NN            EEEE  Do a Write 1, Read, Write 0, Read on RAM NN
;                     Times - Returns number of failing bits
;ANND                 Write Address NN with data byte D
;BNN            D     Read Address NN ; returns data byte D
;CNND                 Write Address NN with data byte D into 5
;                     locations to allow error correction.
;DNN            D     Read Address NN with error correction
;E -- <2048 Bytes>          Program 16K with incoming bit data
;                                (Bit 7 - Start) 2048 bytes = 16K bits
;F -- <2048 Bytes>          Read 16K Contents ; Returns 2048
;                     data bytes
;G -- <40 Bytes>            Read Address 01 to 41
;                     with error correction
;
;
;-------------------------------------------
; RAM MEMORY USAGE
; R0-R7 USED IN MOST ROUTINES
; LOCATIONS 20H TO 32H -- STORES SHIFT REGISTER ASCII CODE
; LOCATIONS 4000H TO 7FFFH -- USED IN RAM TESTING PROGRAMS TO

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  2
                                06-18-92

;                    BE A BAD BIT MAP.
; PROGRAM REQUIRES 32K BYTES OF RAM TO OPERATE.
; FLAG AREA -- 3FF0 TO 3FFF
;-------------------------------------------
;
;                              Port layout
;
; P0.0     Pin 39    A0 I P1.0 Pin 1  DIN TEST I P2.0      Pin 21    A8
; P0.1     Pin 38    A1 I P1.1 Pin 2  Not Used I P2.1      Pin 22    A9
; P0.2     Pin 37    A2 I P1.2 Pin 3  DIN      I P2.2      Pin 23    A10
; P0.3     Pin 36    A3 I P1.3 Pin 4  Jumper 1 I P2.3      Pin 24    A11
; P0.4     Pin 35    A4 I P1.4 Pin 5  Jumper 2 I P2.4      Pin 25    A12
; P0.5     Pin 34    A5 I P1.5 Pin 6  Jumper 3 I P2.5      Pin 26    A13
; P0.6     Pin 33    A6 I P1.6 Pin 7  Jumper 4 I P2.6      Pin 27    /WE
; P0.7     Pin 32    A7 I P1.7 Pin 8  Jumper 5 I P2.7      Pin 28    /CS
;
; P3.0     Pin 10    TXD
; P3.1     Pin 11    RXD
; P3.2     Pin 12    LOW POWER
; P3.3     Pin 13    TEST IN
; P3.4     Pin 14    DOUT
; P3.5     Pin 15    Activity Light
; P3.6     Pin 16    Fault/Test Out 1
; P3.7     Pin 17    Test Out 2
;-------------------------------------------

001B =    ESC   EQU   01BH          ;ESCAPE CODE
000D =    CR    EQU   00DH          ;CARRAGE RETURN
000A =    LF    EQU   00AH          ;LINE FEED
0020 =    SPACE EQU   020H          ;SPACE
0009 =    R10   EQU   009H          ;R0 - BANK 1
000A =    R11   EQU   00AH          ;R1 - BANK 1
000B =    R12   EQU   00BH          ;R2 - BANK 1
000C =    R13   EQU   00CH          ;R3 - BANK 1
000D =    R14   EQU   00DH          ;R4 - BANK 1
000E =    R15   EQU   00EH          ;R5 - BANK 1
000F =    R16   EQU   00FH          ;R6 - BANK 1
0010 =    R17   EQU   010H          ;R7 - BANK 1
3FF0 =    FLAG  EQU   03FF0H        ;NONVOLATILE PROGRAM FLAGS
3FF1 =    R1_ERR      EQU   03FF1H        ;ERROR REGISTER
3FF2 =    R2_ERR      EQU   03FF2H        ;ERROR REGISTER
3FF3 =    R3_ERR      EQU   03FF3H        ;ERROR REGISTER
3FF4 =    R4_ERR      EQU   03FF4H        ;ERROR REGISTER

0030              ORG   30H
          START:          ;Initialization
0030 75D000       MOV     PSW,#00H        ; SET BANK 0
0033 8500A8       MOV   IE,0        ;CLEAR INTERRUPT
0036 758922       MOV   TMOD,#22H   ;COUNTER1,MODE2_8BITS,AUTO RELOAD
0039 758DF3       MOV   TH1,#0F3H   ;TIMER OVERFLOW REGISTERS USED TO
003C 758BF3       MOV   TL1,#0F3H   ; SET BAUD RATE = 2400 BAUD TH1,TL1 TO
CODE #0E6H
                                    ; FOR 4800 SET TH1,TL1 TO CODE #0F3H
003F 758780       MOV   PCON,#80H   ; SET SMOD BIT=1 TO SET BAUD RATE
0042 759850       MOV   SCON,#50H   ; MODE 1 BAUD LENGTH=10BITS
                                    ; START BIT=0

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  3
                                06-18-92

                                    ; 8 DATA BITS (LSB FIRST)
                                    ; STOP BIT=1
                                    ; RECEIVE SHIFT REGISTER ENABLED
0045 758840       MOV   TCON,#40H   ; TIMER 1 OPERATION ENABLED
                                    ; 4800 BPS SHOULD BE WORKING
0048 7440         MOV   A,#40H
004A F581         MOV   SP,A
004C 7590FF       MOV   P1,#0FFH    ; Din Test must be high (P1.0)
004F 7580FF       MOV   P0,#0FFH
0052 75A0FF       MOV   P2,#0FFH
0055 75B0D3       MOV   P3,#0D3H    ; Test In (P3.3) & Low Power (P3.2) must be low

0058 7AFF         MOV   R2,#0FFH    ; TIMEOUT DELAY
005A 00     WLOOP: NOP
005B 00           NOP
005C DAFC         DJNZ  R2,WLOOP

          ;-------------------------------------------
          ;MAIN LOOP - MONITOR SERIAL PORT FOR A COMMAND
          ;-------------------------------------------
          ;
          ;---------- Sign on ----------
          ;
005E 903FF0       MOV     DPTR,#FLAG      ;GET FLAG BYTE
0061 E0           MOVX    A,@DPTR
0062 5401         ANL     A,#01H          ;ONLY INTERESTED IN POWER FAIL-
CONTINUE BIT
0064 7002         JNZ     SKIPALL         ;IF SET GO TO WALKING 1-0'S
0066 016B         AJMP    GOON

0068 020807   SKIPALL:LJMP SA62            ;MAKE THE LONG JUMP
          ;----------
006B 120E51   GOON:       LCALL   SEND_ST         ;SEND THE MESSAGE
006E 0D 0A          DB      CR,LF
0070 4E 6F 76       DB      'Novolatile Electronics, Inc 16K MRAM Test',CR,LF
```

```
0073 6F 6C 61 74 69 6C 65 20 45 6C
007D 65 63 74 72 6F 6E 69 63 73 2C
0087 20 49 6E 63 20 31 36 4B 20 4D
0091 52 41 4D 20 54 65 73 74 0D 0A
0098 20 20 20       DB      '       Version 06-18-92 Rev A',CR,LF,LF
009E 20 20 20 20 20 20 20 20 20 56
00A8 65 72 73 69 6F 6E 20 30 36 2D
00B2 31 38 2D 39 32 20 52 65 76 20
00BC 41 0D 0A 0A
00C0 20 20 20       DB      '               Commands',CR,LF
00C3 20 20 20 20 20 20 20 20 20 20
00CD 20 20 20 20 20 20 43 6F 6D 6D
00D7 61 6E 64 73 0D 0A
00DD 2D 2D 2D       DB      '--------------------------------------------',CR,LF
00E0 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
00EA 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
00F4 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
00FE 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
0108 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
0112 2D 2D 2D 2D 2D 2D 0D 0A
011A 20 31 41       DB      ' 1AAAA  (Returns D)  Read Hex Address AAAA; Returns 1 or
0',CR,LF

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  4
                                                    06-18-92
011D 41 41 41 20 20 28 52 65 74 75
0127 72 6E 73 20 44 29 20 20 52 65
0131 61 64 20 48 65 78 20 41 64 64
013B 72 65 73 73 20 41 41 41 41 3B
0145 20 52 65 74 75 72 6E 73 20 31
014F 20 6F 72 20 30 0D 0A
0156 20 32 41       DB      ' 2AAAAD          Write Hex Address AAAA with data D',CR,LF
0159 41 41 41 44 20 20 20 20 20 20
0163 20 20 20 20 20 20 20 20 57 72
016D 69 74 65 20 48 65 78 20 41 64
0177 64 72 65 73 73 20 41 41 41 41
0181 20 77 69 74 68 20 64 61 74 61
018B 20 44 0D 0A
018F 20 33 44       DB      ' 3DDDDDDDDDDDDDDDDDDD Load MRAM Shift Reg. with hex
data D',CR,LF
0192 44 44 44 44 44 44 44 44 44 44
019C 44 44 44 44 44 44 44 44 20 4C
01A6 6F 61 64 20 4D 52 41 4D 20 53
01B0 68 69 66 74 20 52 65 67 2E 20
01BA 77 69 74 68 20 68 65 78 20 64
01C4 61 74 61 20 44 0D 0A
01CB 20 34 20       DB      ' 4               Activate shift register',CR,LF
01CE 20 20 20 20 20 20 20 20 20 20
01D8 20 20 20 20 20 20 20 20 41 63
01E2 74 69 76 61 74 65 20 73 68 69
01EC 66 74 20 72 65 67 69 73 74 65
01F6 72 0D 0A
01F9 20 35 20       DB      ' 5               Deactivate shift register',CR,LF
01FC 20 20 20 20 20 20 20 20 20 20
0206 20 20 20 20 20 20 20 20 44 65
0210 61 63 74 69 76 61 74 65 20 73
021A 68 69 66 74 20 72 65 67 69 73
0224 74 65 72 0D 0A
0229 20 36 4E       DB      ' 6NN (Returns EEEE)  Walking 1-0 test NN times. If NN
=00,',CR,LF
022C 4E 20 28 52 65 74 75 72 6E 73
0236 20 45 45 45 45 29 20 20 57 61
0240 6C 6B 69 6E 67 20 31 2D 30 20
024A 74 65 73 74 20 4E 4E 20 74 69
0254 6D 65 73 2E 20 49 66 20 4E 4E
025E 20 3D 30 30 2C 0D 0A
0265 20 20 20       DB      '               Test until CTRN is received from key board',CR,LF
0268 20 20 20 20 20 20 20 20 20 20
0272 20 20 20 20 20 20 20 20 54 65
027C 73 74 20 75 6E 74 69 6C 20 43
0286 54 52 4E 20 69 73 20 72 65 63
0290 65 69 76 65 64 20 66 72 6F 6D
029A 20 6B 65 79 20 62 6F 61 72 64
02A4 0D 0A
02A6 20 20 20       DB      '               Returns number of failures up to 4000 hex.',CR,LF
02A9 20 20 20 20 20 20 20 20 20 20
02B3 20 20 20 20 20 20 20 20 52 65
02BD 74 75 72 6E 73 20 6E 75 6D 62
02C7 65 72 20 6F 66 20 66 61 69 6C
02D1 75 72 65 73 20 75 70 20 74 6F
02DB 20 34 30 30 30 20 68 65 78 2E
02E5 0D 0A

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  5
                                                    06-18-92
02E7 20 37 41       DB      ' 7AAAA(Returns EEEE) Pulse 1-0 test at address AAAA 16,384
times.',CR,LF
02EA 41 41 41 28 52 65 74 75 72 6E
02F4 73 20 45 45 45 45 29 20 50 75
02FE 6C 73 65 20 31 2D 30 20 74 65
0308 73 74 20 61 74 20 61 64 64 72
0312 65 73 73 20 41 41 41 41 20 31
031C 36 2C 33 38 34 20 74 69 6D 65
0326 73 2E 0D 0A
032A 20 20 20       DB      '               Returns number of failures up to 4000 hex.',CR,LF
032D 20 20 20 20 20 20 20 20 20 20
0337 20 20 20 20 20 20 20 20 52 65
0341 74 75 72 6E 73 20 6E 75 6D 62
034B 65 72 20 6F 66 20 66 61 69 6C
0355 75 72 65 73 20 75 70 20 74 6F
035F 20 34 30 30 30 20 68 65 78 2E
0369 0D 0A
036B 20 38 20       DB      ' 8 (Returns AAAA)  Dump bad bit addresses',CR,LF
036E 28 52 65 74 75 72 6E 73 20 41
0378 41 41 41 29 20 20 20 20 44 75
0382 6D 70 20 62 61 64 20 62 69 74
038C 20 61 64 64 72 65 73 73 65 73
0396 0D 0A
0398 20 39 4E       DB      ' 9NN (Returns EEEE)  Write 1s, Read, Write 0s, read NN
times',CR,LF
039B 4E 20 28 52 65 74 75 72 6E 73
03A5 20 45 45 45 45 29 20 20 57 72
03AF 69 74 65 20 31 73 2C 20 52 65
03B9 61 64 2C 20 57 72 69 74 65 20
03C3 30 73 2C 20 72 65 61 64 20 4E
03CD 4E 20 74 69 6D 65 73 0D 0A
03D6 20 20 20       DB      '               Returns number of failures up to 4000
hex.',CR,LF,ESC
03D9 20 20 20 20 20 20 20 20 20 20
03E3 20 20 20 20 20 20 20 20 52 65
03ED 74 75 72 6E 73 20 6E 75 6D 62
03F7 65 72 20 6F 66 20 66 61 69 6C
0401 75 72 65 73 20 75 70 20 74 6F
040B 20 34 30 30 30 20 68 65 78 2E
0415 0D 0A 1B

0418 120CB6         LCALL   GETCH             ;PAUSE TO READ SCREEN
041B 120E51         LCALL   SEND_ST           ;SEND THE MESSAGE

041E 20 41 4E       DB      ' ANND            Write address NN with data D',CR,LF
0421 4E 44 20 20 20 20 20 20 20
042B 20 20 20 20 20 20 20 20 57 72
0435 69 74 65 20 61 64 64 72 65 73
043F 73 20 4E 4E 20 77 69 74 68 20
0449 64 61 74 61 20 44 0D 0A
0451 20 42 4E       DB      ' BNN (Returns D)   Read address NN, return data D',CR,LF
0454 4E 20 28 52 65 74 75 72 6E 73
045E 20 44 29 20 20 20 20 20 52 65
0468 61 64 20 61 64 64 72 65 73 73
0472 20 4E 4E 2C 20 72 65 74 75 72
047C 6E 20 64 61 74 61 20 44 0D 0A
0486 20 43 4E       DB      ' CNND            Write address NN with data D into 5
locations',CR,LF
0489 4E 44 20 20 20 20 20 20 20 20

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page  6
                                                    06-18-92
0493 20 20 20 20 20 20 20 20 57 72
049D 69 74 65 20 61 64 64 72 65 73
04A7 73 20 4E 4E 20 77 69 74 68 20
04B1 64 61 74 61 20 44 20 69 6E 74
04BB 6F 20 35 20 6C 6F 63 61 74 69
04C5 6F 6E 73 0D 0A
04CA 20 20 20       DB      '               to provide error correction',CR,LF
04CD 20 20 20 20 20 20 20 20 20 20
04D7 20 20 20 20 20 20 20 20 74 6F
04E1 20 70 72 6F 76 69 64 65 20 65
04EB 72 72 6F 72 20 63 6F 72 72 65
04F5 63 74 69 6F 6E 0D 0A
04FC 20 44 4E       DB      ' DNN (Returns D)   Read address NN with error
correction',CR,LF
04FF 4E 20 28 52 65 74 75 72 6E 73
0509 20 44 29 20 20 20 20 20 52 65
0513 61 64 20 61 64 64 72 65 73 73
051D 20 4E 4E 20 77 69 74 68 20 65
0527 72 72 6F 72 20 63 6F 72 72 65
0531 63 74 69 6F 6E 0D 0A
```

```
0538 20 45 2D       DB      ' E--<2048 Bytes>   Program 16K with serial in 2048 byte
stream',CR,LF
0538 2D 3C 32 30 34 38 20 42 79 74
0545 65 73 3E 20 20 20 20 20 50 72
054F 6F 67 72 61 6D 20 31 36 4B 20
0559 77 69 74 68 20 73 65 72 69 61
0563 6C 20 69 6E 20 32 30 34 38 20
056D 62 79 74 65 20 73 74 72 65 61
0577 6D 0D 0A
057A 20 20 20       DB      '               (Bit 7 = start, 2048 Bytes = 16K Bits',CR,LF
057D 20 20 20 20 20 20 20 20 20 20
0587 20 20 20 20 20 20 20 20 28 42
0591 69 74 20 37 20 3D 20 73 74 61
059B 72 74 2C 20 32 30 34 38 20 42
05A5 79 74 65 73 20 3D 20 31 36 4B
05AF 20 42 69 74 73 0D 0A
05B6 20 46 2D       DB      ' F--<2048 Bytes>   Read 16K contents, send serial 2048
Bytes',CR,LF
05B9 2D 3C 32 30 34 38 20 42 79 74
05C3 65 73 3E 20 20 20 20 20 52 65
05CD 61 64 20 31 36 4B 20 63 6F 6E
05D7 74 65 6E 74 73 2C 20 73 65 6E
05E1 64 20 73 65 72 69 61 6C 20 32
05EB 30 34 38 20 42 79 74 65 73 0D
05F5 0A
05F6 20 47 2D       DB      ' G--<40 Bytes>     Read address 01 to 41 with error correction'
05F9 2D 3C 34 30 20 42 79 74 65 73
0603 3E 20 20 20 20 20 20 20 52 65
060D 61 64 20 61 64 64 72 65 73 73
0617 20 30 31 20 74 6F 20 34 31 20
0621 77 69 74 68 20 65 72 72 6F 72
062B 20 63 6F 72 72 65 63 74 69 6F
0635 6E
0636 0D 0A 1B       DB      CR,LF,ESC

        MLOOP:
0639 120E3C     LCALL   CRLF
        MLOOP1:
```



```
063C 120CC8     LCALL GETECHO
063F B43103     CJNE  A,#'1',NO1    ;CHECK IF RECEIVED CHARACTER IS
0642 0206A1     LJMP  ISA1          ;A COMMAND
0645 B43203  NO1: CJNE  A,#'2',NO2
0648 0206D6     LJMP  ISA2
064B B43303  NO2: CJNE  A,#'3',NO3
064E 02070B     LJMP  ISA3
0651 B43403  NO3: CJNE  A,#'4',NO4
0654 02071B     LJMP  ISA4
0657 B43503  NO4: CJNE  A,#'5',NO5
065A 020720     LJMP  ISA5
065D B43603  NO5: CJNE  A,#'6',NO6
0660 020725     LJMP  ISA6
0663 B43703  NO6: CJNE  A,#'7',NO7
0666 020920     LJMP  ISA7
0669 B43803  NO7: CJNE  A,#'8',NO8
066C 020993     LJMP  ISA8
066F B43903  NO8: CJNE  A,#'9',NO9
0672 0209C5     LJMP  ISA9
0675 B44103  NO9: CJNE  A,#'A',NOA
0678 020A27     LJMP  ISAA
067B B44203  NOA: CJNE  A,#'B',NOB
067E 020A4B     LJMP  ISAB
0681 B44303  NOB: CJNE  A,#'C',NOC
0684 020A6D     LJMP  ISAC
0687 B44403  NOC: CJNE  A,#'D',NOD
068A 020AB9     LJMP  ISAD
068D B44503  NOD: CJNE  A,#'E',NOE
0690 020AD6     LJMP  ISAE
0693 B44603  NOE: CJNE  A,#'F',NOF
0696 020AF4     LJMP  ISAF
0699 B44703  NOF: CJNE  A,#'G',NOG
069C 020B15     LJMP  ISAG
069F C13C    NOG: AJMP  MLOOP1


        ;-------------------------------
        ;COMMAND 1 - READ A BIT AT ADDRESS AAAA
        ;
        ;-------------------------------
        ISA1:
06A1 120CC8     LCALL GETECHO
06A4 FB         MOV   R3,A          ;ADDRESS BYTE 1
06A5 120CC8     LCALL GETECHO
06A8 FC         MOV   R4,A
06A9 120CC8     LCALL GETECHO
06AC FD         MOV   R5,A
06AD 120CC8     LCALL GETECHO
06B0 FE         MOV   R6,A
06B1 120E3C     LCALL   CRLF
06B4 EE         MOV     A,R6
06B5 120DDA     LCALL XVERT         ;CONVERT HEX ADDRESS INTO BINARY FORM
06B8 F8         MOV   R0,A
06B9 ED         MOV   A,R5
```



```
06BA 120DDA     LCALL XVERT
06BD C4         SWAP  A
06BE 4200       ORL   R0,A
06C0 EC         MOV   A,R4
06C1 120DDA     LCALL XVERT
06C4 F9         MOV   R1,A
06C5 EB         MOV   A,R3
06C6 120DDA     LCALL XVERT
06C9 C4         SWAP  A
06CA 4201       ORL   R1,A
06CC 120CCF     LCALL READ          ;READ BIT AT ADDRESS
06CF 2430       ADD   A,#'0'
06D1 120CBE     LCALL PUTCH         ;WRITE DATA IN ASCII FORM TO SERIAL PORT
06D4 C139       AJMP  MLOOP

        ;-------------------------------
        ;COMMAND 2 - WRITE A BIT AT ADDRESS AAAA TO D
        ;-------------------------------
        ISA2:
06D6 120CC8     LCALL GETECHO
06D9 FB         MOV   R3,A
06DA 120CC8     LCALL GETECHO
06DD FC         MOV   R4,A
06DE 120CC8     LCALL GETECHO
06E1 FD         MOV   R5,A
06E2 120CC8     LCALL GETECHO
06E5 FE         MOV   R6,A
06E6 120E3C     LCALL   CRLF
06E9 EE         MOV     A,R6
06EA 120DDA     LCALL XVERT
06ED F8         MOV   R0,A
06EE ED         MOV   A,R5
06EF 120DDA     LCALL XVERT
06F2 C4         SWAP  A
06F3 4200       ORL   R0,A
06F5 EC         MOV   A,R4
06F6 120DDA     LCALL XVERT
06F9 F9         MOV   R1,A
06FA EB         MOV   A,R3
06FB 120DDA     LCALL XVERT
06FE C4         SWAP  A
06FF 4201       ORL   R1,A
0701 120CC8     LCALL GETECHO
0704 9430       SUBB  A,#'0'        ;A CONTAINS DATA NOW
0706 120CEE     LCALL WRITE
0709 C139       AJMP  MLOOP

        ;-------------------------------
        ;COMMAND 3 - LOAD THE SHIFT REGISTER
        ;-------------------------------
070B 7A13    ISA3: MOV   R2,#19
070D 7932          MOV   R1,#32H
        ISA31:
070F 120CC8     LCALL GETECHO
0712 F7         MOV   @R1,A         ;STORE CHARACTER IN RAM
```



```
0713 19         DEC   R1
0714 DAF9       DJNZ  R2,ISA31
0716 120D9D     LCALL SHIFT
0719 C139       AJMP  MLOOP

        ;-------------------------------
```

```
        ;COMMAND 4 - ACTIVATE SHIFT REGISTER
        ;-------------------------------------
071B 120D69   ISA4:  LCALL  ACTIV
071E C139            AJMP   MLOOP

        ;-------------------------------------
        ;COMMAND 5 - DEACTIVATE SHIFT REGISTER
        ;-------------------------------------
0720 120D84   ISA5:  LCALL  DACTIV
0723 C139            AJMP   MLOOP

        ;-------------------------------------
        ;COMMAND 6 - WALKING 1-0 TEST PATTERN
        ;-------------------------------------

        ISA6:
0725 903FF0          MOV    DPTR,#FLAG      ;GET FLAG BYTE
0728 E0              MOVX   A,@DPTR
0729 4401            ORL    A,#01H          ;ONLY INTERESTED IN POWER FAIL-
CONTINUE BIT
072B F0              MOVX   @DPTR,A         ;SET IT FOR WALKING 1-0'S
        ;-----------

072C 120CC8          LCALL  GETECHO
072F FB              MOV    R3,A
0730 120CC8          LCALL  GETECHO
0733 FC              MOV    R4,A
0734 120E3C          LCALL  CRLF
0737 EC              MOV    A,R4
0738 120DDA          LCALL  XVERT
073B F9              MOV    R1,A
073C EB              MOV    A,R3
073D 120DDA          LCALL  XVERT
0740 C4              SWAP   A
0741 4201            ORL    R1,A
0743 E9              MOV    A,R1
0744 FF              MOV    R7,A        ;R7 CONTAINS NUMBER OF CYCLES TO TEST
        ;---------- Ask if bad bit buffer should be cleared -----

0745 120E51          LCALL    SEND_ST
0748 0D 0A           DB       CR,LF
074A 44 6F 20        DB       'Do you want to clear the bad bit buffer - Y or N ',ESC
074D 79 6F 75 20 77 61 6E 74 20 74
0757 6F 20 63 6C 65 61 72 20 74 68
0761 65 20 62 61 64 20 62 69 74 20
076B 62 75 66 66 65 72 20 2D 20 59
0775 20 6F 72 20 4E 20 1B
077C 120CC8          LCALL    GETECHO
077F 120E3C          LCALL    CRLF
0782 B45939          CJNE     A,#'Y',    SA611    ;IF 'N', DON'T CLEAR IT

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page 10
                                    06-18-92


0785 120E51          LCALL    SEND_ST
0788 54 68 65        DB       'The bad bit buffer has been cleared ',ESC
078B 20 62 61 64 20 62 69 74 20 62
0795 75 66 66 65 72 20 68 61 73 20
079F 62 65 65 6E 20 63 6C 65 61 72
07A9 65 64 20 1B
07AD 120E3C          LCALL    CRLF
07B0 7940            MOV    R1,#040H      ;CLEAR THE BAD BIT BUFFER IN RAM
07B2 E4              CLR    A
07B3 904000          MOV    DPTR,#4000H
07B6 7800      SA60: MOV    R0,#00H
07B8 F0        SA61: MOVX   @DPTR,A
07B9 A3              INC    DPTR
07BA D8FC            DJNZ   R0,SA61
07BC D9F8            DJNZ   R1,SA60

        SA611:
07BE 120E51          LCALL    SEND_ST
07C1 0D 0A           DB       CR,LF
07C3 44 6F 20        DB       'Do you want to clear the Loop Counter - Y or N ',ESC
07C6 79 6F 75 20 77 61 6E 74 20 74
07D0 6F 20 63 6C 65 61 72 20 74 68
07DA 65 20 4C 6F 6F 70 20 43 6F 75
07E4 6E 74 65 72 20 2D 20 59 20 6F
07EE 72 20 4E 20 1B
07F3 120CC8          LCALL    GETECHO
07F6 120E3C          LCALL    CRLF
07F9 B4590B          CJNE     A,#'Y',    SA62     ;IF 'N', DON'T CLEAR IT
07FC 903FF1          MOV      DPTR,#R1_ERR
07FF E4              CLR    A
0800 F0              MOVX   @DPTR,A
0801 A3              INC      DPTR
0802 F0              MOVX   @DPTR,A
0803 A3              INC      DPTR
0804 F0              MOVX   @DPTR,A
0805 A3              INC      DPTR
0806 F0              MOVX   @DPTR,A

        SA62:
0807 120C24          LCALL  WALK10       ;EXECUTE WALKING TEST R7 TIMES

        ;---------- Increment the loop Counter ----------

080A 903FF1          MOV      DPTR,#R1_ERR
080D C3              CLR    C
080E E0              MOVX   A,@DPTR
080F 2401            ADD      A,#01H              ;BUMP THE 32 BIT LOOP COUNT
0811 F0              MOVX   @DPTR,A
0812 A3              INC      DPTR
0813 E0              MOVX   A,@DPTR
0814 3400            ADDC   A,#00H
0816 F0              MOVX   @DPTR,A
0817 A3              INC      DPTR
0818 E0              MOVX   A,@DPTR

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page 11
                                    06-18-92


0819 3400            ADDC   A,#00H
081B F0              MOVX   @DPTR,A
081C A3              INC      DPTR
081D E0              MOVX   A,@DPTR
081E 3400            ADDC   A,#00H
0820 F0              MOVX   @DPTR,A

        ;---------- Flash Activity Light ----------
0821 E5B0            MOV      A,P3
0823 F4              CPL      A                   ;FLIP LED STATUS BIT
0824 5420            ANL      A,#020H             ;GET LED STATUS BIT
0826 C2B5            CLR      P3.5                ;CLEAR IT OUT
0828 42B0            ORL      P3,A                ;SET COMPLEMENT VALUE
        ;
082A EF              MOV      A,R7
082B 7005            JNZ      SA622               ;IF COUNT = 0 DO IT UNTIL 'CTRN'
082D 3098D7          JNB      RI,SA62
0830 8002            SJMP     SA623               ;ELSE LOOP UNTIL R7 GOES TO ZERO

0832 DFD3      SA622:        DJNZ   R7,SA62
                                  ;COUNT UP NUMBER OF FAILED BITS
        SA623:
0834 903FF0          MOV      DPTR,#FLAG          ;GET FLAG BYTE
0837 E0              MOVX   A,@DPTR
0838 54FE            ANL      A,#0FEH             ;ONLY INTERESTED IN POWER FAIL-
CONTINUE BIT
083A F0              MOVX   @DPTR,A             ;CLEAR IT TO EXIT WALKING 1-0'S
        ;-----------
083B 7800            MOV    R0,#00
083D 7900            MOV    R1,#00      ;CLEAR ERROR REGISTER - 16 BIT

083F 904000          MOV    DPTR,#4000H
0842 7A40            MOV    R2,#040H
0844 7B00      SA64: MOV    R3,#00H
0846 E0        SA63: MOVX   A,@DPTR
0847 C3              CLR    C
0848 28              ADD    A,R0
0849 F8              MOV    R0,A
084A E4              CLR    A
084B 39              ADDC   A,R1
084C F9              MOV    R1,A
084D A3              INC    DPTR
084E DBF6            DJNZ   R3,SA63
0850 DAF2            DJNZ   R2,SA64     ;R1,R0 CONTAIN THE COUNT
        ;
        ;---------- Send error count string ----------
        ;
0852 120E51          LCALL    SEND_ST             ;SEND THE MESSAGE
0855 0D 0A           DB       CR,LF
0857 4E 75 6D        DB       'Number of Failed Bits With Walking 1-0 Pattern = ',ESC
085A 62 65 72 20 6F 66 20 46 61 69
0864 6C 65 64 20 42 69 74 73 20 57
086E 69 74 68 20 57 61 6C 6B 69 6E
0878 67 20 31 2D 30 20 50 61 74 74
```

```
0882 65 72 6E 20 3D 20 1B

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page 12
                                    06-18-92


0889 AA00        MOV     R2,R0              ;SAVE LOWER ERROR COUNT R0
088B E9          MOV     A,R1
088C 120E03      LCALL   RVERT
088F E9          MOV     A,R1
0890 120CBE      LCALL   PUTCH
0893 E8          MOV     A,R0
0894 120CBE      LCALL   PUTCH
0897 EA          MOV     A,R2
0898 120E03      LCALL   RVERT
089B E9          MOV     A,R1
089C 120CBE      LCALL   PUTCH
089F E8          MOV     A,R0
08A0 120CBE      LCALL   PUTCH
                 ;
                 ;---------- Send Loop count string ----------
                 ;
08A3 120E51      LCALL   SEND_ST            ;SEND THE MESSAGE
08A6 0D 0A       DB      CR,LF
08A8 4E 75 6D    DB      'Number of Loops Completed With Walking 1-0 Pattern = ',ESC
08AB 62 65 72 20 6F 66 20 4C 6F 6F
08B5 70 73 20 43 6F 6D 70 6C 65 74
08BF 65 64 20 57 69 74 68 20 57 61
08C9 6C 6B 69 6E 67 20 31 2D 30 20
08D3 50 61 74 74 65 72 6E 20 3D 20
08DD 1B

08DE 903FF4      MOV     DPTR,#R4_ERR       ;SEND MOST SIGNIFICANT DIGIT FIRST
08E1 E0          MOVX    A,@DPTR
08E2 120E03      LCALL   RVERT
08E5 E9          MOV     A,R1
08E6 120CBE      LCALL   PUTCH
08E9 E8          MOV     A,R0
08EA 120CBE      LCALL   PUTCH
08ED 903FF3      MOV     DPTR,#R3_ERR       ;SEND NEXT SIGNIFICANT DIGIT FIRST
08F0 E0          MOVX    A,@DPTR
08F1 120E03      LCALL   RVERT
08F4 E9          MOV     A,R1
08F5 120CBE      LCALL   PUTCH
08F8 E8          MOV     A,R0
08F9 120CBE      LCALL   PUTCH

08FC 120E4B      LCALL   PUTSPACE           ;SPACE
08FF 903FF2      MOV     DPTR,#R2_ERR       ;SEND NEXT SIGNIFICANT DIGIT FIRST
0902 E0          MOVX    A,@DPTR
0903 120E03      LCALL   RVERT
0906 E9          MOV     A,R1
0907 120CBE      LCALL   PUTCH
090A E8          MOV     A,R0
090B 120CBE      LCALL   PUTCH
090E 903FF1      MOV     DPTR,#R1_ERR       ;SEND LEAST SIGNIFICANT DIGIT FIRST
0911 E0          MOVX    A,@DPTR
0912 120E03      LCALL   RVERT
0915 E9          MOV     A,R1
0916 120CBE      LCALL   PUTCH
0919 E8          MOV     A,R0

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page 13
                                    06-18-92


091A 120CBE      LCALL   PUTCH
091D 020639      LJMP    MLOOP      ;GO TO OPERATOR INPUT

                 ;-----------------------------------
                 ;COMMAND 7 - TEST BIT AT ADDRESS AAAA, 65,536 TIMES
                 ;-----------------------------------
                 ISA7:
0920 120CC8      LCALL   GETECHO
0923 FB          MOV     R3,A
0924 120CC8      LCALL   GETECHO
0927 FC          MOV     R4,A
0928 120CC8      LCALL   GETECHO
092B FD          MOV     R5,A
092C 120CC8      LCALL   GETECHO
092F FE          MOV     R6,A
0930 120E3C      LCALL   CRLF
0933 EE          MOV     A,R6
0934 120DDA      LCALL   XVERT
0937 F8          MOV     R0,A
0938 ED          MOV     A,R5
0939 120DDA      LCALL   XVERT
093C C4          SWAP    A
093D 4200        ORL     R0,A
093F EC          MOV     A,R4
0940 120DDA      LCALL   XVERT
0943 F9          MOV     R1,A
0944 EB          MOV     A,R3
0945 120DDA      LCALL   XVERT
0948 C4          SWAP    A
0949 4201        ORL     R1,A       ;R0,R1 CONTAIN THE ADDRESS
094B 7D00        MOV     R5,#00
094D 7E00        MOV     R6,#00
094F 7B28        MOV     R3,#40
0951 7C00    SA70: MOV   R4,#00
0953 7401    SA72: MOV   A,#01
0955 120CEE      LCALL   WRITE
0958 120CCF      LCALL   READ
095B 30E02A      JNB     0E0H,SA71
095E 7400        MOV     A,#00
0960 120CEE      LCALL   WRITE
0963 120CCF      LCALL   READ
0966 20E01F      JB      0E0H,SA71
0969 DCE8    SA73: DJNZ  R4,SA72
096B DBE4        DJNZ    R3,SA70
096D EE          MOV     A,R6
096E 120E03      LCALL   RVERT
0971 E9          MOV     A,R1
0972 120CBE      LCALL   PUTCH
0975 E8          MOV     A,R0
0976 120CBE      LCALL   PUTCH
0979 ED          MOV     A,R5
097A 120E03      LCALL   RVERT
097D E9          MOV     A,R1
097E 120CBE      LCALL   PUTCH
0981 E8          MOV     A,R0

The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04  Page 14
                                    06-18-92


0982 120CBE      LCALL   PUTCH
0985 020639      LJMP    MLOOP      ;GO TO OPERATOR INPUT

0988 C3      SA71: CLR   C
0989 7401        MOV     A,#01
098B 2D          ADD     A,R5
098C FD          MOV     R5,A
098D 7400        MOV     A,#00
098F 3E          ADDC    A,R6
0990 FE          MOV     R6,A
0991 80D6        SJMP    SA73


                 ;-----------------------------------
                 ;COMMAND 8 - DUMP ADDRESSES OF BAD BITS
                 ;-----------------------------------
0993 904000  ISA8: MOV   DPTR,#4000H
0996 7940        MOV     R1,#40H
0998 7800    SA80: MOV   R0,#00H
099A E0      SA82: MOVX  A,@DPTR
099B 20E008      JB      0E0H,SA81
099E A3      SA83: INC   DPTR
099F D8F9        DJNZ    R0,SA82
09A1 D9F5        DJNZ    R1,SA80
09A3 020639      LJMP    MLOOP      ;GO TO OPERATOR INPUT

09A6 E8      SA81: MOV   A,R0
09A7 FB          MOV     R3,A
09A8 E9          MOV     A,R1
09A9 120E03      LCALL   RVERT
09AC E9          MOV     A,R1
09AD 120CBE      LCALL   PUTCH
09B0 E8          MOV     A,R0
09B1 120CBE      LCALL   PUTCH
09B4 EB          MOV     A,R3
09B5 120E03      LCALL   RVERT
09B8 E9          MOV     A,R1
09B9 120CBE      LCALL   PUTCH
09BC E8          MOV     A,R0
09BD 120CBE      LCALL   PUTCH
09C0 120E3C      LCALL   CRLF
```

```
09C3 80D9          SJMP   SA83
          ;-------------------------------------
          ;COMMAND 9 - 1-0 TEST PATTERN
          ;-------------------------------------

          ISA9:
09C5 120CC8        LCALL  GETECHO
09C8 FB            MOV    R3,A
09C9 120CC8        LCALL  GETECHO
09CC FC            MOV    R4,A
09CD 120E3C        LCALL    CRLF
09D0 EC            MOV    A,R4
09D1 120DDA        LCALL  XVERT
```



```
09D4 F9            MOV    R1,A
09D5 EB            MOV    A,R3
09D6 120DDA        LCALL  XVERT
09D9 C4            SWAP   A
09DA 4201          ORL    R1,A
09DC E9            MOV    A,R1
09DD FF            MOV    R7,A        ;R7 CONTAINS NUMBER OF CYCLES TO TEST

09DE 7940          MOV    R1,#040H    ;CLEAR THE BAD BIT BUFFER IN RAM
09E0 7400          MOV    A,#00
09E2 904000        MOV    DPTR,#4000H
09E5 7800     SA90: MOV   R0,#00H
09E7 F0       SA91: MOVX  @DPTR,A
09E8 A3            INC    DPTR
09E9 D8FC          DJNZ   R0,SA91
09EB D9F8          DJNZ   R1,SA90

09ED 120BFC   SA92: LCALL  TEST10      ;EXECUTE WALKING TEST R7 TIMES
09F0 DFFB          DJNZ   R7,SA92

09F2 7800          MOV    R0,#00
09F4 7900          MOV    R1,#00      ;COUNT UP NUMBER OF FAILED BITS
09F6 904000        MOV    DPTR,#4000H
09F9 7A40          MOV    R2,#040H
09FB 7B00     SA94: MOV   R3,#00H
09FD E0       SA93: MOVX  A,@DPTR
09FE C3            CLR    C
09FF 28            ADD    A,R0
0A00 F8            MOV    R0,A
0A01 7400          MOV    A,#00
0A03 39            ADDC   A,R1
0A04 F9            MOV    R1,A
0A05 A3            INC    DPTR
0A06 DBF5          DJNZ   R3,SA93
0A08 DAF1          DJNZ   R2,SA94      ;R1,R0 CONTAIN THE COUNT

0A0A E8            MOV    A,R0
0A0B FB            MOV    R3,A
0A0C E9            MOV    A,R1
0A0D 120E03        LCALL  RVERT
0A10 E9            MOV    A,R1
0A11 120CBE        LCALL  PUTCH
0A14 E8            MOV    A,R0
0A15 120CBE        LCALL  PUTCH
0A18 EB            MOV    A,R3
0A19 120E03        LCALL  RVERT
0A1C E9            MOV    A,R1
0A1D 120CBE        LCALL  PUTCH
0A20 E8            MOV    A,R0
0A21 120CBE        LCALL  PUTCH
0A24 020639        LJMP   MLOOP       ;GO TO OPERATOR INPUT

          ;-------------------------------------
          ; ISAA ROUTINE - WRITE BYTE AT ADDRESS NN DATA D
          ;-------------------------------------
```



```
          ISAA:
0A27 120CC8        LCALL  GETECHO
0A2A FB            MOV    R3,A
0A2B 120CC8        LCALL  GETECHO
0A2E FC            MOV    R4,A
0A2F 120CC8        LCALL  GETECHO
0A32 FD            MOV    R5,A
0A33 120E3C        LCALL    CRLF
0A36 EC            MOV    A,R4
0A37 120DDA        LCALL  XVERT
0A3A F8            MOV    R0,A
0A3B EB            MOV    A,R3
0A3C 120DDA        LCALL  XVERT
0A3F C4            SWAP   A
0A40 4200          ORL    R0,A        ;A CONTAINS ADDRESS 0 TO 255
0A42 ED            MOV    A,R5        ;A CONTAINS DATA BYTE TO WRITE
0A43 7B00          MOV    R3,#00      ;WRITE BYTE #00
0A45 120B9C        LCALL  WBYTE1        ;WRITE BYTE 1X ROUTINE
0A48 020639        LJMP   MLOOP       ;GO TO OPERATOR INPUT

          ;-------------------------------------
          ; ISAB ROUTINE - READ BYTE AT ADDRESS NN DATA D
          ;-------------------------------------
          ISAB:
0A4B 120CC8        LCALL  GETECHO
0A4E FB            MOV    R3,A
0A4F 120CC8        LCALL  GETECHO
0A52 FC            MOV    R4,A
0A53 120E3C        LCALL    CRLF
0A56 EC            MOV    A,R4
0A57 120DDA        LCALL  XVERT
0A5A F8            MOV    R0,A
0A5B EB            MOV    A,R3
0A5C 120DDA        LCALL  XVERT
0A5F C4            SWAP   A
0A60 4200          ORL    R0,A        ;A CONTAINS ADDRESS 0 TO 255
0A62 7B00          MOV    R3,#00
0A64 120BCA        LCALL  RBYTE1        ;WRITE BYTE 1X ROUTINE

0A67 120CBE        LCALL  PUTCH
0A6A 020639        LJMP   MLOOP       ;GO TO OPERATOR INPUT

          ;-------------------------------------
          ; ISAD ROUTINE - WRITE BYTE AT ADDRESS NN DATA D 5 TIMES
          ;-------------------------------------
          ISAC:
0A6D 120CC8        LCALL  GETECHO
0A70 FB            MOV    R3,A
0A71 120CC8        LCALL  GETECHO
0A74 FC            MOV    R4,A
0A75 120CC8        LCALL  GETECHO
0A78 FD            MOV    R5,A
0A79 120E3C        LCALL    CRLF
0A7C EC            MOV    A,R4
0A7D 120DDA        LCALL  XVERT
```



```
0A80 F8            MOV    R0,A
0A81 EB            MOV    A,R3
0A82 120DDA        LCALL  XVERT
0A85 C4            SWAP   A
0A86 4200          ORL    R0,A        ;A CONTAINS ADDRESS 0 TO 255
0A88 ED            MOV    A,R5        ;A CONTAINS DATA BYTE TO WRITE
0A89 7B00          MOV    R3,#00H
0A8B F560          MOV    60H,A       ;R6 CONTAINS DATA
0A8D 8861          MOV    61H,R0      ;R7 CONTAINS ADDRESS
0A8F 120B9C        LCALL  WBYTE1        ;WRITE BYTE 1X ROUTINE
0A92 7B08          MOV    R3,#08H
0A94 E560          MOV    A,60H
0A96 A861          MOV    R0,61H
0A98 120B9C        LCALL  WBYTE1
0A9B 7B10          MOV    R3,#10H
0A9D E560          MOV    A,60H
0A9F A861          MOV    R0,61H
0AA1 120B9C        LCALL  WBYTE1
0AA4 7B18          MOV    R3,#18H
0AA6 E560          MOV    A,60H
0AA8 A861          MOV    R0,61H
0AAA 120B9C        LCALL  WBYTE1
0AAD 7B20          MOV    R3,#20H
0AAF E560          MOV    A,60H
0AB1 A861          MOV    R0,61H
0AB3 120B9C        LCALL  WBYTE1      ;IT'S WRITTEN 5 TIMES
0AB6 020639        LJMP   MLOOP       ;GO TO OPERATOR INPUT

          ;-------------------------------------
          ; ISAD ROUTINE - READ BYTE AT ADDRESS NN DATA D 5 TIMES
```

```
        ;-------------------------------
        ISAD:
0AB9 120CC8     LCALL  GETECHO
0ABC FB         MOV    R3,A
0ABD 120CC8     LCALL  GETECHO
0AC0 FC         MOV    R4,A
0AC1 120E3C     LCALL   CRLF
0AC4 EC         MOV    A,R4
0AC5 120DDA     LCALL  XVERT
0AC8 F8         MOV    R0,A
0AC9 EB         MOV    A,R3
0ACA 120DDA     LCALL  XVERT
0ACD C4         SWAP   A
0ACE 4200       ORL    R0,A        ;R0 CONTAINS ADDRESS 0 TO 255
0AD0 120B37     LCALL  RDRBY
0AD3 020639     LJMP   MLOOP       ;GO TO OPERATOR INPUT


        ;-------------------------------
        ; ISAE ROUTINE - PROGRAM ENTIRE MRAM WITH INCOMING DATA
        ;-------------------------------
0AD6 7940    ISAE:  MOV   R1,#40H
0AD8 7800    ISAE1: MOV   R0,#00      ;SET UP TO RECEIVE 2048 BYTES
        ISAE3:
0ADA 120CC8     LCALL  GETECHO
```



```
0ADD FB         MOV    R3,A        ;STORE IN R3
0ADE 7A08       MOV    R2,#8       ;ROTATE ALL 8 BITS
0AE0 EB     ISAE2: MOV    A,R3
0AE1 5401       ANL    A,#1
0AE3 120D3A     LCALL  LWRITE      ;WRITE THE BIT
0AE6 EB         MOV    A,R3
0AE7 13         RRC    A
0AE8 FB         MOV    R3,A        ;ROTATE 1 BIT
0AE9 18         DEC    R0          ;DECREMENT ADDRESS COUNT
0AEA DAF4       DJNZ   R2,ISAE2    ;WRITE ALL 8 BITS
0AEC 08         INC    R0          ;DUMMY INCREMENT TO FIX DJNZ INSTRUCTION
0AED D8EB       DJNZ   R0,ISAE3
0AEF D9E7       DJNZ   R1,ISAE1
0AF1 020639     LJMP   MLOOP       ;GO TO OPERATOR INPUT

        ;-------------------------------
        ; ISAF - READ ENTIRE MRAM AND DUMP TO SERIAL PORT
        ;-------------------------------
0AF4 7940    ISAF:  MOV   R1,#40H
0AF6 7800    ISAF1: MOV   R0,#00      ;SET UP TO RECEIVE 2048 BYTES
0AF8 7A08    ISAF3: MOV   R2,#8       ;ROTATE ALL 8 BITS
0AFA 7B00           MOV   R3,#00
0AFC 120D19  ISAF2: LCALL  LREAD      ;READ MRAM LOCATION

0AFF 5401       ANL    A,#1
0B01 C3         CLR    C
0B02 13         RRC    A           ;CARRY FLAG CONTAINS DATA
0B03 EB         MOV    A,R3
0B04 13         RRC    A
0B05 FB         MOV    R3,A        ;ROTATE BIT INTO R3 REGISTER
0B06 18         DEC    R0          ;DECREMENT ADDRESS COUNT
0B07 DAF3       DJNZ   R2,ISAF2    ;READ ALL 8 BITS

0B09 EB         MOV    A,R3
0B0A 120CBE     LCALL  PUTCH       ;WRITE THE CHARACTER
0B0D 08         INC    R0          ;DUMMY INCREMENT TO FIX DJNZ INSTRUCTION
0B0E D8E8       DJNZ   R0,ISAF3
0B10 D9E4       DJNZ   R1,ISAF1
0B12 020639     LJMP   MLOOP       ;GO TO OPERATOR INPUT

        ;-------------------------------
        ; ISAG - READ ADDRESS 01 TO 41 (40 BYTES WITH ERROR CORRECTION)
        ;-------------------------------
0B15 904100  ISAG:  MOV   DPTR,#4100H
0B18 7428           MOV   A,#40
0B1A F0             MOVX  @DPTR,A
0B1B F8             MOV   R0,A
0B1C 7400           MOV   A,#00
0B1E A3             INC   DPTR
0B1F F0             MOVX  @DPTR,A
0B20 904100  ISAG1: MOV   DPTR,#4100H
0B23 E8             MOV   A,R0
0B24 F0             MOVX  @DPTR,A
0B25 A3             INC   DPTR
0B26 E0             MOVX  A,@DPTR
```



```
0B27 04             INC   A
0B28 F0             MOVX  @DPTR,A
0B29 F8             MOV   R0,A
0B2A 120B37         LCALL  RDRBY
0B2D 904100         MOV   DPTR,#4100H
0B30 E0             MOVX  A,@DPTR
0B31 F8             MOV   R0,A
0B32 D8EC           DJNZ  R0,ISAG1
0B34 020639         LJMP  MLOOP       ;GO TO OPERATOR INPUT

        ;-------------------------------
        ; READ ERROR BYTE - R0 CONTAINS ADDRESS
        ;-------------------------------
0B37 AE00    RDRBY: MOV   R6,R0       ;R6 CONTAINS ADDRESS
0B39 7B00           MOV   R3,#00H
0B3B 904000         MOV   DPTR,#4000H
0B3E 120BCA         LCALL  RBYTE1     ;READ BYTE 1X ROUTINE
0B41 F0             MOVX  @DPTR,A
0B42 A3             INC   DPTR
0B43 7B08           MOV   R3,#08H
0B45 A806           MOV   R0,R6
0B47 120BCA         LCALL  RBYTE1
0B4A F0             MOVX  @DPTR,A
0B4B A3             INC   DPTR
0B4C 7B10           MOV   R3,#10H
0B4E A806           MOV   R0,R6
0B50 120BCA         LCALL  RBYTE1
0B53 F0             MOVX  @DPTR,A
0B54 A3             INC   DPTR
0B55 7B18           MOV   R3,#18H
0B57 A806           MOV   R0,R6
0B59 120BCA         LCALL  RBYTE1
0B5C F0             MOVX  @DPTR,A
0B5D A3             INC   DPTR
0B5E 7B20           MOV   R3,#20H
0B60 A806           MOV   R0,R6
0B62 120BCA         LCALL  RBYTE1
0B65 F0             MOVX  @DPTR,A

                                      ;ALL 5 BYTES READ - LAST ONE IS IN A
0B66 7900           MOV   R1,#00
0B68 7808           MOV   R0,#08
0B6A 904000  BAD3:  MOV   DPTR,#4000H
0B6D 7A05           MOV   R2,#05      ;BIT COUNT
0B6F 7B00           MOV   R3,#00
0B71 EE             MOV   A,R6
0B72 E0      BAD1:  MOVX  A,@DPTR
0B73 5401           ANL   A,#1        ;CHECK BIT 1
0B75 2B             ADD   A,R3
0B76 FB             MOV   R3,A
0B77 E0             MOVX  A,@DPTR
0B78 13             RRC   A
0B79 F0             MOVX  @DPTR,A
0B7A A3             INC   DPTR
0B7B DAF5           DJNZ  R2,BAD1
```



```
0B7D BB0502         CJNE  R3,#5,ISNO5
0B80 618F           AJMP  ISAA1
0B82 BB0402  ISNO5: CJNE  R3,#4,ISNO4
0B85 618F           AJMP  ISAA1
0B87 BB0302  ISNO4: CJNE  R3,#3,ISNO3
0B8A 618F           AJMP  ISAA1
0B8C C3      ISNO3: CLR   C
0B8D 6192           AJMP  BAD2
0B8F 7401    ISAA1: MOV   A,#1
0B91 13             RRC   A
0B92 E9      BAD2:  MOV   A,R1
0B93 13             RRC   A
0B94 F9             MOV   R1,A        ;ROTATE BIT INTO R1
0B95 D8D3           DJNZ  R0,BAD3
0B97 E9             MOV   A,R1
0B98 120CBE         LCALL  PUTCH
0B9B 22             RET
```

```
        ;-------------------------------------
        ; WBYTE 1 ROUTINE
        ;-------------------------------------
0B9C AD00   WBYTE1: MOV   R5,R0
0B9E FC             MOV   R4,A
0B9F 7900           MOV   R1,#00
0BA1 E8             MOV   A,R0
0BA2 C3             CLR   C
0BA3 33             RLC   A
0BA4 F8             MOV   R0,A
0BA5 E9             MOV   A,R1
0BA6 33             RLC   A
0BA7 F9             MOV   R1,A
0BA8 E8             MOV   A,R0
0BA9 33             RLC   A
0BAA F8             MOV   R0,A
0BAB E9             MOV   A,R1
0BAC 33             RLC   A
0BAD F9             MOV   R1,A
0BAE E8             MOV   A,R0
0BAF 33             RLC   A
0BB0 F8             MOV   R0,A
0BB1 E9             MOV   A,R1
0BB2 33             RLC   A         ;R0,R1 CONTAIN BIT ADDRESS
0BB3 4B             ORL   A,R3      ;OR WITH UPPER ADDRES FOR BYTE # IN R3
0BB4 F9             MOV   R1,A
0BB5 7A08           MOV   R2,#8
0BB7 EC     WBYLP:  MOV   A,R4
0BB8 5401           ANL   A,#1
0BBA 120CEE         LCALL WRITE
0BBD EC             MOV   A,R4
0BBE C3             CLR   C
0BBF 13             RRC   A
0BC0 FC             MOV   R4,A
0BC1 C3             CLR   C
```



```
0BC2 08             INC   R0
0BC3 E9             MOV   A,R1
0BC4 3400           ADDC  A,#0
0BC6 F9             MOV   R1,A
0BC7 DAEE           DJNZ  R2,WBYLP
0BC9 22             RET

        ;-------------------------------------
        ; RBYTE 1 ROUTINE
        ;-------------------------------------
0BCA AD00   RBYTE1: MOV   R5,R0
0BCC 7C00           MOV   R4,#00
0BCE 7900           MOV   R1,#00
0BD0 E8             MOV   A,R0
0BD1 33             RLC   A
0BD2 F8             MOV   R0,A
0BD3 E9             MOV   A,R1
0BD4 33             RLC   A
0BD5 F9             MOV   R1,A
0BD6 E8             MOV   A,R0
0BD7 33             RLC   A
0BD8 F8             MOV   R0,A
0BD9 E9             MOV   A,R1
0BDA 33             RLC   A
0BDB F9             MOV   R1,A
0BDC E8             MOV   A,R0
0BDD 33             RLC   A
0BDE F8             MOV   R0,A
0BDF E9             MOV   A,R1
0BE0 33             RLC   A         ;R0,R1 CONTAIN BIT ADDRESS
0BE1 4B             ORL   A,R3      ;R3 CONTAINS UPPER ADDRESS BITS
0BE2 F9             MOV   R1,A

0BE3 7A08           MOV   R2,#8
0BE5 120CCF RBYLP:  LCALL READ
0BE8 13             RRC   A
0BE9 13             RRC   A         ; BIT 7 OF A CONTAINS DATA BIT
0BEA 5480           ANL   A,#80H
0BEC FB             MOV   R3,A
0BED EC             MOV   A,R4
0BEE C3             CLR   C
0BEF 13             RRC   A
0BF0 4B             ORL   A,R3
0BF1 FC             MOV   R4,A
0BF2 C3             CLR   C

0BF3 08             INC   R0
0BF4 E9             MOV   A,R1
0BF5 3400           ADDC  A,#0
0BF7 F9             MOV   R1,A
0BF8 DAEB           DJNZ  R2,RBYLP
0BFA EC             MOV   A,R4
0BFB 22             RET
```



```
        ;-------------------------------------
        ;TEST 1-0 ROUTINE  WRITE 1, READ , WRITE 0, READ
        ;-------------------------------------
0BFC 904000 TEST10: MOV   DPTR,#4000H
0BFF 7940           MOV   R1,#040H
0C01 7800   TT100:  MOV   R0,#00H
0C03 7401   TT103:  MOV   A,#01
0C05 120D3A         LCALL LWRITE
0C08 120D19         LCALL LREAD
0C0B 30E011         JNB   0E0H,TT101
0C0E 7400           MOV   A,#00
0C10 120D3A         LCALL LWRITE
0C13 120D19         LCALL LREAD
0C16 20E006         JB    0E0H,TT101
0C19 A3     TT102:  INC   DPTR
0C1A D8E7           DJNZ  R0,TT103
0C1C D9E3           DJNZ  R1,TT100
0C1E 22             RET
0C1F 7401   TT101:  MOV   A,#01
0C21 F0             MOVX  @DPTR,A
0C22 80F5           SJMP  TT102

        ;-------------------------------------
        ;WALKING 1 - 0 TEST ROUTINE
        ;-------------------------------------
0C24 904000 WALK10: MOV   DPTR,#4000H
0C27 7400           MOV   A,#0
0C29 120CA8         LCALL FILL
0C2C 7940           MOV   R1,#040H
0C2E 7800   W100:   MOV   R0,#00H
0C30 120D19 W101:   LCALL LREAD
0C33 20E011         JB    0E0H,W102
0C36 7401           MOV   A,#01
0C38 120D3A         LCALL LWRITE
0C3B 120D19         LCALL LREAD
0C3E 30E006         JNB   0E0H,W102
0C41 A3     W103:   INC   DPTR
0C42 D8EC           DJNZ  R0,W101
0C44 D9E8           DJNZ  R1,W100
            ;       SJMP  W104
0C46 22             RET
0C47 7401   W102:   MOV   A,#1
0C49 F0             MOVX  @DPTR,A
0C4A 80F5           SJMP  W103

0C4C 904000 W104:   MOV   DPTR,#4000H
0C4F 7940           MOV   R1,#040H
0C51 7800   W105:   MOV   R0,#00H
0C53 120D19 W107:   LCALL LREAD
0C56 30E007         JNB   0E0H,W106
0C59 A3     W109:   INC   DPTR
0C5A D8F7           DJNZ  R0,W107
0C5C D9F3           DJNZ  R1,W105
0C5E 8005           SJMP  W108
0C60 7401   W106:   MOV   A,#1
```



```
0C62 F0             MOVX  @DPTR,A
0C63 80F4           SJMP  W109

0C65 904000 W108:   MOV   DPTR,#4000H
0C68 7401           MOV   A,#1
0C6A 120CA8         LCALL FILL
0C6D 7940           MOV   R1,#040H
0C6F 7800   W100A:  MOV   R0,#00H
```

```
0C71 120D19  W101A: LCALL  LREAD
0C74 30E012        JNB    0E0H,W102A
0C77 7400          MOV    A,#0
0C79 120D3A        LCALL  LWRITE
0C7C 120D19        LCALL  LREAD
0C7F 20E007        JB     0E0H,W102A
0C82 A3      W103A: INC    DPTR
0C83 D8EC          DJNZ   R0,W101A
0C85 D9E8          DJNZ   R1,W100A
0C87 8005          SJMP   W104A
0C89 7401    W102A: MOV    A,#1
0C8B F0            MOVX   @DPTR,A
0C8C 80F4          SJMP   W103A

0C8E 904000  W104A: MOV    DPTR,#4000H
0C91 7940          MOV    R1,#040H
0C93 7800    W105A: MOV    R0,#00H
0C95 120D19  W107A: LCALL  LREAD
0C98 20E007        JB     0E0H,W106A
0C9B A3      W109A: INC    DPTR
0C9C D8F7          DJNZ   R0,W107A
0C9E D9F3          DJNZ   R1,W105A
0CA0 8005          SJMP   W108A
0CA2 7401    W106A: MOV    A,#1
0CA4 F0            MOVX   @DPTR,A
0CA5 80F4          SJMP   W109A
0CA7 22      W108A: RET


        ;-------------------------------------
        ;FILL RAM ROUTINE
        ;-------------------------------------
        FILL:
0CA8 FC            MOV    R4,A
0CA9 7940          MOV    R1,#040H
0CAB 7800    FILL0: MOV    R0,#00H
0CAD EC      FILL1: MOV    A,R4
0CAE 120D3A        LCALL  LWRITE
0CB1 D8FA          DJNZ   R0,FILL1
0CB3 D9F6          DJNZ   R1,FILL0
0CB5 22            RET

        ;-------------------------------------
        ;GETCH SUBROUTINE
        ;-------------------------------------
0CB6 3098FD  GETCH: JNB    RI,GETCH        ;wait for a keyboard response
0CB9 C298          CLR    RI
```



```
0CBB E599          MOV    A,SBUF          ;read the serial buffer
0CBD 22            RET

        ;-------------------------------------
        ;PUTCH SUBROUTINE
        ;-------------------------------------
0CBE C299    PUTCH: CLR    TI

0CC0 F599    XF1:   MOV    SBUF,A
0CC2 3099FD  XF2:   JNB    TI,XF2          ; WAIT FOR BUFFER TO CLEAR
0CC5 C299          CLR    TI              ; CLR FLAG
0CC7 22            RET
        ;-------------------------------------
        ;GETECHO SUBROUTINE - GET AND ECHO CHARACTER
        ;-------------------------------------
        GETECHO:
0CC8 120CB6        LCALL  GETCH
0CCB 120CBE        LCALL  PUTCH
0CCE 22            RET
        ;-------------------------------------
        ;READ MRAM LOCATION SUBROUTINE
        ;R0 CONTAINS LOWER 8 BITS OF ADDRESS
        ;R1 CONTAINS UPPER 6 BITS OF ADDRESS
        ;DATA IS RETURNED IN A
        ;-------------------------------------
0CCF E8      READ:  MOV    A,R0
0CD0 F580          MOV    P0,A
0CD2 E9            MOV    A,R1
0CD3 44C0          ORL    A,#0C0H
0CD5 F5A0          MOV    P2,A
0CD7 53A07F        ANL    P2,#7FH
0CDA 00            NOP
0CDB 00            NOP
0CDC 00            NOP
0CDD 00            NOP
0CDE 00            NOP
0CDF 20B406        JB     T0,RD1A
0CE2 7400          MOV    A,#0
0CE4 43A0C0        ORL    P2,#0C0H
0CE7 22            RET
0CE8 7401    RD1A: MOV    A,#1
0CEA 43A0C0        ORL    P2,#0C0H
0CED 22            RET

        ;-------------------------------------
        ;WRITE MRAM LOCATION SUBROUTINE
        ;R0 CONTAINS LOWER 8 BITS OF ADDRESS
        ;R1 CONTAINS UPPER 6 BITS OF ADDRESS
        ;DATA IS IN 0E0H
        ;-------------------------------------
0CEE 20E014  WRITE: JB     0E0H,W1
0CF1 5390FB        ANL    P1,#0FBH
0CF4 E8            MOV    A,R0
0CF5 F580          MOV    P0,A
0CF7 E9            MOV    A,R1
```



```
0CF8 54BF          ANL    A,#0BFH
0CFA 4480          ORL    A,#080H
0CFC F5A0          MOV    P2,A
0CFE 53A07F        ANL    P2,#7FH
0D01 43A080        ORL    P2,#080H
0D04 22            RET
0D05 439004  W1:   ORL    P1,#04H
0D08 E8            MOV    A,R0
0D09 F580          MOV    P0,A
0D0B E9            MOV    A,R1
0D0C 54BF          ANL    A,#0BFH
0D0E 4480          ORL    A,#080H
0D10 F5A0          MOV    P2,A
0D12 53A07F        ANL    P2,#7FH
0D15 43A080        ORL    P2,#080H
0D18 22            RET

        ;-------------------------------------
        ;LREAD MRAM LOCATION SUBROUTINE
        ;R0 CONTAINS LOWER 8 BITS OF ADDRESS + 1
        ;R1 CONTAINS UPPER 6 BITS OF ADDRESS + 1
        ;DATA IS RETURNED IN A
        ;-------------------------------------
        LREAD:
0D19 E8            MOV    A,R0
0D1A 14            DEC    A
0D1B F580          MOV    P0,A
0D1D E9            MOV    A,R1
0D1E 14            DEC    A
0D1F 44C0          ORL    A,#0C0H
0D21 F5A0          MOV    P2,A
0D23 53A07F        ANL    P2,#7FH
0D26 00            NOP
0D27 00            NOP
0D28 00            NOP
0D29 00            NOP
0D2A 00            NOP
0D2B 20B406        JB     T0,LRA1
0D2E 7400          MOV    A,#0
0D30 43A0C0        ORL    P2,#0C0H
0D33 22            RET
0D34 7401    LRA1: MOV    A,#1
0D36 43A0C0        ORL    P2,#0C0H
0D39 22            RET

        ;-------------------------------------
        ;LWRITE MRAM LOCATION SUBROUTINE
        ;R0 CONTAINS LOWER 8 BITS OF ADDRESS + 1
        ;R1 CONTAINS UPPER 6 BITS OF ADDRESS + 1
        ;DATA IS IN 0E0H
        ;-------------------------------------
0D3A 20E016  LWRITE: JB     0E0H,LW1
0D3D 5390FB        ANL    P1,#0FBH
0D40 E8            MOV    A,R0
0D41 14            DEC    A
```

```
0D42 F580          MOV   P0,A
0D44 E9            MOV   A,R1
0D45 14            DEC   A
0D46 54BF          ANL   A,#0BFH
0D48 4480          ORL   A,#080H
0D4A F5A0          MOV   P2,A
0D4C 53A07F        ANL   P2,#7FH
0D4F 43A080        ORL   P2,#080H
0D52 22            RET
0D53 439004   LW1:  ORL   P1,#04H
0D56 E8            MOV   A,R0
0D57 14            DEC   A
0D58 F580          MOV   P0,A
0D5A E9            MOV   A,R1
0D5B 14            DEC   A
0D5C 54BF          ANL   A,#0BFH
0D5E 4480          ORL   A,#080H
0D60 F5A0          MOV   P2,A
0D62 53A07F        ANL   P2,#7FH
0D65 43A080        ORL   P2,#080H
0D68 22            RET

        ;-------------------------------------
        ;ACTIVATE SHIFT REGISTER ROUTINE
        ;-------------------------------------
0D69 759003   ACTIV: MOV   P1,#3      ;BRING LOWPOW AND TEST HIGH
0D6C 759001        MOV   P1,#1      ;LEAVE TEST HIGH
0D6F 120D79        LCALL DELAY1
0D72 759000        MOV   P1,#0
0D75 120D9D        LCALL SHIFT
0D78 22            RET

        ;-------------------------------------
        ; DELAY ROUTINE
        ;-------------------------------------
0D79 7C0A     DELAY1: MOV   R4,#10
0D7B 7B00     DL1:  MOV   R3,#00
0D7D EB       DL2:  MOV   A,R3
0D7E EB             MOV   A,R3
0D7F DBFC           DJNZ  R3,DL2
0D81 DCF8           DJNZ  R4,DL1
0D83 22             RET

        ;-------------------------------------
        ;DE-ACTIVATE SHIFT REGISTER ROUTINE
        ;-------------------------------------
0D84 759002   DACTIV: MOV   P1,#2      ;BRING LOWPOW AND TEST HIGH
0D87 759000        MOV   P1,#0      ;LEAVE TEST HIGH.
        ; TRANSMIT 24 'W' S TO CONSOLE TO GIVE A 50ms Delay
0D8A A818          MOV   R0,24
0D8C 7477     DACT1: MOV   A,#'w'
0D8E C299          CLR   TI
0D90 F599          MOV   SBUF,A
```



```
0D92 3099FD   DXF2:  JNB   TI,DXF2     ; WAIT FOR BUFFER TO CLEAR
0D95 C299          CLR   TI         ; CLR FLAG
0D97 D8F3          DJNZ  R0,DACT1
0D99 759000        MOV   P1,#0
0D9C 22            RET

        ;-------------------------------------
        ;PROGRAM SHIFT REGISTER ROUTINE
        ;DATA IS IN HEX FORMAT AT LOCATIONS 20H TO 33H
        ;-------------------------------------
0D9D 758037   SHIFT: MOV   P0,#37H
0DA0 75A0DF        MOV   P2,#0DFH    ;SET UP ADDRESS BUS
0DA3 439001        ORL   P1,#1       ;MAKE TEST HIGH
0DA6 53A07F        ANL   P2,#7FH     ;CE GOES LOW
0DA9 5390FE        ANL   P1,#0FEH    ;TEST GOES LOW

        ; START THE SHIFTING PROCESS
0DAC 7913          MOV   R1,#19      ;19 HEX DIGITS
0DAE 7820          MOV   R0,#20H     ;R2 IS POINTER TO HEX DIGIT
0DB0 E6     SHFTDO: MOV   A,@R0      ;A CONTAINS ASCII DIGIT
0DB1 120DDA        LCALL XVERT       ;CONVERT TO BINARY
0DB4 C4            SWAP  A
        ;EXECUTE THE SHIFTING FUNCTION 4 TIMES
0DB5 7B04          MOV   R3,#04H
0DB7 53A0FB   SHFT1: ANL   P2,#0FBH    ;SET CLK TO LOW
0DBA 33            RLC   A           ;ROTATE A RIGHT THROUGH THE CARRY
0DBB 4005          JC    SHFT11
0DBD 53A0FE        ANL   P2,#0FEH    ;SET DATA TO A 0
0DC0 8003          SJMP  SHFT10
0DC2 43A001   SHFT11: ORL   P2,#01H    ;SET DATA TO A 1
0DC5 43A004   SHFT10: ORL   P2,#04H    ;RISING EDGE OF CLK SHIFTS IN DATA
0DC8 DBED          DJNZ  R3,SHFT1    ;DO IT 4 TIMES

0DCA 08            INC   R0
0DCB D9E3          DJNZ  R1,SHFTDO
0DCD 43A007        ORL   P2,#07H     ;BRING ALL CLK,DAT,LAT HIGH
0DD0 53A0FD        ANL   P2,#0FDH    ;BRING LAT LOW
0DD3 43A007        ORL   P2,#07H     ;BRING LAT HIGH
0DD6 75A0FF        MOV   P2,#0FFH    ;END THE CYCLE
0DD9 22            RET

        ;-------------------------------------
        ;XVERT SUBROUTINE - CONVERTS ASCII HEX TO BINARY
        ;-------------------------------------
0DDA B44102   XVERT: CJNE  A,#'A',SHA
0DDD 801E          SJMP  SHL1
0DDF B44202   SHA:  CJNE  A,#'B',SHB
0DE2 8019          SJMP  SHL1
0DE4 B44302   SHB:  CJNE  A,#'C',SHC
0DE7 8014          SJMP  SHL1
0DE9 B44402   SHC:  CJNE  A,#'D',SHD
0DEC 800F          SJMP  SHL1
0DEE B44502   SHD:  CJNE  A,#'E',SHE
0DF1 800A          SJMP  SHL1
0DF3 B44602   SHE:  CJNE  A,#'F',SHF
```



```
0DF6 8005          SJMP  SHL1
0DF8 C3       SHF:  CLR   C
0DF9 9430          SUBB  A,#'0'      ;IS A NUMBER
0DFB 8005          SJMP  SHG
0DFD C3       SHL1: CLR   C
0DFE 9441          SUBB  A,#'A'      ;IS A LETTER
0E00 240A          ADD   A,#0AH
0E02 22       SHG:  RET             ;A CONTAINS THE NEXT 4 BITS IN LSB'S

        ;-------------------------------------
        ;RVERT SUBROUTINE - CONVERTS BINARY TO ASCII HEX
        ;-------------------------------------
0E03 F5F0     RVERT: MOV   B,A
0E05 540F          ANL   A,#0FH
0E07 120E15        LCALL RVERT1
0E0A F8            MOV   R0,A
0E0B E5F0          MOV   A,B
0E0D C4            SWAP  A
0E0E 540F          ANL   A,#0FH
0E10 120E15        LCALL RVERT1
0E13 F9            MOV   R1,A
0E14 22            RET

0E15 B40A02   RVERT1: CJNE  A,#0AH,RHA
0E18 801D          SJMP  RHL1
0E1A B40B02   RHA:  CJNE  A,#0BH,RHB
0E1D 8018          SJMP  RHL1
0E1F B40C02   RHB:  CJNE  A,#0CH,RHC
0E22 8013          SJMP  RHL1
0E24 B40D02   RHC:  CJNE  A,#0DH,RHD
0E27 800E          SJMP  RHL1
0E29 B40E02   RHD:  CJNE  A,#0EH,RHE
0E2C 8009          SJMP  RHL1
0E2E B40F02   RHE:  CJNE  A,#0FH,RHF
0E31 8004          SJMP  RHL1
0E33 2430     RHF:  ADD   A,#'0'      ;IS A NUMBER
0E35 8004          SJMP  RHG
0E37 2441     RHL1: ADD   A,#'A'      ;IS A LETTER
0E39 940A          SUBB  A,#0AH
0E3B 22       RHG:  RET             ;A CONTAINS THE NEXT 4 BITS IN LSB'S
        ;-------------------------------------
        ;CRLF SUBROUTINE - SENDS CARRIDGE RETURN LINEFEED
        ;-------------------------------------
```

```
        CRLF:
0E3C C0E0       PUSH    ACC
0E3E 740D       MOV     A,#0DH          ;SEND CARRAGE RETURN
0E40 120CBE     LCALL   PUTCH           ;ECHO CHARACTER
0E43 740A       MOV     A,#0AH          ;SEND LINE FEED
0E45 120CBE     LCALL   PUTCH           ;ECHO CHARACTER
0E48 D0E0       POP     ACC
0E4A 22         RET
        ;---------------------------------------
        ;SPACE SUBROUTINE - SENDS SPACE
        ;---------------------------------------
```



```
        PUTSPACE:
0E4B 7420       MOV     A,#SPACE        ;SEND SPACE
0E4D 120CBE     LCALL   PUTCH
0E50 22         RET
        ;---------------------------------------
        ; ---- Send a string to serial xmitter ----------
        ;---------------------------------------
        SEND_ST:
0E51 8583F0     MOV     B,DPH           ;SAVE THE DATA POINTER
0E54 E582       MOV     A,DPL
0E56 D083       POP     DPH             ;LOAD DPTR WITH FIRST CHARACTER
0E58 D082       POP     DPL
0E5A C0F0       PUSH    B
0E5C C0E0       PUSH    ACC             ;SAVE DATA POINTER REGISTERS
0E5E E4         CLR     A               ;ZERO OFFSET
0E5F 93         MOVC    A,@A+DPTR       ;FETCH FIRST CHARACTER OF STRING
        SEND_IT:
0E60 120CBE     LCALL   PUTCH           ;SEND IT
0E63 A3         INC     DPTR            ;BUMP THE POINTER
0E64 E4         CLR     A
0E65 93         MOVC    A,@A+DPTR       ;GET THE NEXT CHARACTER TO SEND
0E66 B41BF7     CJNE    A,#ESC,SEND_IT  ;LOOP UNTIL ESC IS FOUND
0E69 8583F0     MOV     B,DPH           ;SAVE RETURN ADDRESS
0E6C E582       MOV     A,DPL
0E6E D082       POP     DPL             ;RESTORE DATA POINTER REGISTER
0E70 D083       POP     DPH
0E72 C0E0       PUSH    ACC             ;SAVE RETURN ADDRESS
0E74 C0F0       PUSH    B
0E76 7401       MOV     A,#1
0E78 22         RET                     ;RETURN TO CODE AFTER ESC
        ;---------------------------------------


0030            END START
```


;%T Symbol Name Type Value

ACTIV . . . . . L 0D69
BAD1. . . . . L 0B72
BAD2. . . . . L 0B92
BAD3. . . . . L 0B6A
CR. . . . . I 000D
CRLF. . . . . L 0E3C
DACT1 . . . . . L 0D8C
DACTIV. . . . . L 0D84
DELAY1. . . . . L 0D79
DL1 . . . . . L 0D7B
DL2 . . . . . L 0D7D
DXF2. . . . . L 0D92
ESC . . . . . I 001B
FILL. . . . . L 0CA8
FILL0 . . . . . L 0CAB
FILL1 . . . . . L 0CAD
FLAG. . . . . I 3FF0
GETCH . . . . . L 0CB6
GETECHO . . . . . L 0CC8
GOON. . . . . L 006B
ISA1. . . . . L 06A1
ISA2. . . . . L 06D6
ISA3. . . . . L 070B
ISA31 . . . . . L 070F
ISA4. . . . . L 071B
ISA5. . . . . L 0720
ISA6. . . . . L 0725
ISA7. . . . . L 0920
ISA8. . . . . L 0993
ISA9. . . . . L 09C5
ISAA. . . . . L 0A27
ISAA1 . . . . . L 088F
ISAB. . . . . L 0A4B
ISAC. . . . . L 0A6D
ISAD. . . . . L 0AB9
ISAE. . . . . L 0AD6
ISAE1 . . . . . L 0AD8
ISAE2 . . . . . L 0AE0
ISAE3 . . . . . L 0ADA
ISAF. . . . . L 0AF4
ISAF1 . . . . . L 0AF6
ISAF2 . . . . . L 0AFC
ISAF3 . . . . . L 0AF8
ISAG. . . . . L 0B15
ISAG1 . . . . . L 0B20
ISNO3 . . . . . L 0B8C
ISNO4 . . . . . L 0B87
ISNO5 . . . . . L 0B82
LF. . . . . I 000A
LRA1. . . . . L 0D34
LREAD . . . . . L 0D19
LW1 . . . . . L 0D53

LWRITE. . . . . L 0D3A
MLOOP . . . . . L 0639
MLOOP1. . . . . L 063C
NO1 . . . . . L 0645
NO2 . . . . . L 064B
NO3 . . . . . L 0651
NO4 . . . . . L 0657
NO5 . . . . . L 065D
NO6 . . . . . L 0663
NO7 . . . . . L 0669
NO8 . . . . . L 066F
NO9 . . . . . L 0675
NOA . . . . . L 067B
NOB . . . . . L 0681
NOC . . . . . L 0687
NOD . . . . . L 068D
NOE . . . . . L 0693
NOF . . . . . L 0699
NOG . . . . . L 069F
PUTCH . . . . . L 0CBE
PUTSPACE. . . . . L 0E4B
R10 . . . . . I 0009
R11 . . . . . I 000A
R12 . . . . . I 000B
R13 . . . . . I 000C
R14 . . . . . I 000D
R15 . . . . . I 000E
R16 . . . . . I 000F
R17 . . . . . I 0010
R1_ERR. . . . . I 3FF1
R2_ERR. . . . . I 3FF2
R3_ERR. . . . . I 3FF3
R4_ERR. . . . . I 3FF4
RBYLP . . . . . L 0BE5
RBYTE1. . . . . L 0BCA
RD1A. . . . . L 0CE8
RDRBY . . . . . L 0B37
READ. . . . . L 0CCF
RHA . . . . . L 0E1A
RHB . . . . . L 0E1F
RHC . . . . . L 0E24
RHD . . . . . L 0E29
RHE . . . . . L 0E2E
RHF . . . . . L 0E33
RHG . . . . . L 0E3B
RHL1. . . . . L 0E37
RVERT . . . . . L 0E03
RVERT1. . . . . L 0E15
SA60. . . . . L 07B6
SA61. . . . . L 07B8
SA611 . . . . . L 07BE
SA62. . . . . L 0807
SA622 . . . . . L 0832
SA623 . . . . . L 0834

SA83..... L 0846

SA84..... L 0844
SA70..... L 0951
SA71..... L 0968
SA72..... L 0953
SA73..... L 0969
SA80..... L 0998
SA81..... L 09A6
SA82..... L 099A
SA83..... L 099E
SA90..... L 09E5
SA91..... L 09E7
SA92..... L 09ED
SA93..... L 09FD
SA94..... L 09FB
SEND..... U 0000
SEND_IT..... L 0E60
SEND_ST..... L 0E51
SHA..... L 0DDF
SHB..... L 0DE4
SHC..... L 0DE9
SHD..... L 0DEE
SHE..... L 0DF3
SHF..... L 0DF8
SHFT1..... L 0DB7
SHFT10..... L 0DC5
SHFT11..... L 0DC2
SHFTDO..... L 0DB0
SHG..... L 0E02
SHIFT..... L 0D9D
SHL1..... L 0DFD
SKIPALL..... L 0068
SPACE..... I 0020
START..... L 0030
TEST10..... L 0BFC
TT100..... L 0C01
TT101..... L 0C1F
TT102..... L 0C19
TT103..... L 0C03
W1..... L 0D05
W100..... L 0C2E
W100A..... L 0C6F
W101..... L 0C30
W101A..... L 0C71
W102..... L 0C47
W102A..... L 0C89
W103..... L 0C41
W103A..... L 0C82
W104..... L 0C4C
W104A..... L 0C8E
W105..... L 0C51
W105A..... L 0C93
W106..... L 0C60
W106A..... L 0CA2
W107..... L 0C53
W107A..... L 0C95

W108..... L 0C65
W108A..... L 0CA7
W109..... L 0C59
W109A..... L 0C9B
WALK10..... L 0C24
WBYLP..... L 0BB7
WBYTE1..... L 0B9C
WLOOP..... L 005A
WRITE..... L 0CEE
XF1..... L 0CC0
XF2..... L 0CC2
XVERT..... L 0DDA

;%Z

00 Errors (0000)

SECTION 3

WAFER BUS DESIGN

3.1 Introduction

The wafer bus design was essentially complete as of the end of the third quarter of this program.

Changes were made to the 16K chip's internal signals which did not affect the chip size as explained in detail in the fourth quarter report. As a result, no additions were required to the die area and the wafer bus design remained as is.

NVE plans to wait for Honeywell SSEC to produce some wafers with good 16K parts before ordering the manufacture of the wafer bus masks . Honeywell had planned to have 16K wafers with the new timing sequence through their process line by the end of August. However, due to process problems, this did not occur. The current plan is to have parts by the ends of November. At this time, NVE will procure and test some of these wafers, and upon verifying proper operation of the parts, will order the wafer bus masks.

SECTION 4

TEST CHIP

4.1 Introduction

NVE decided early on in this program that a test chip incorporating the MRAM bit required by the 1 Meg design would be desirable. Having a working GMR bit with the dimensions required by the 1 Meg would allow electrical specifications and tolerances to be finalized with confidence; in addition, any processing issues affecting the layout of the design would become apparent and could be addressed.

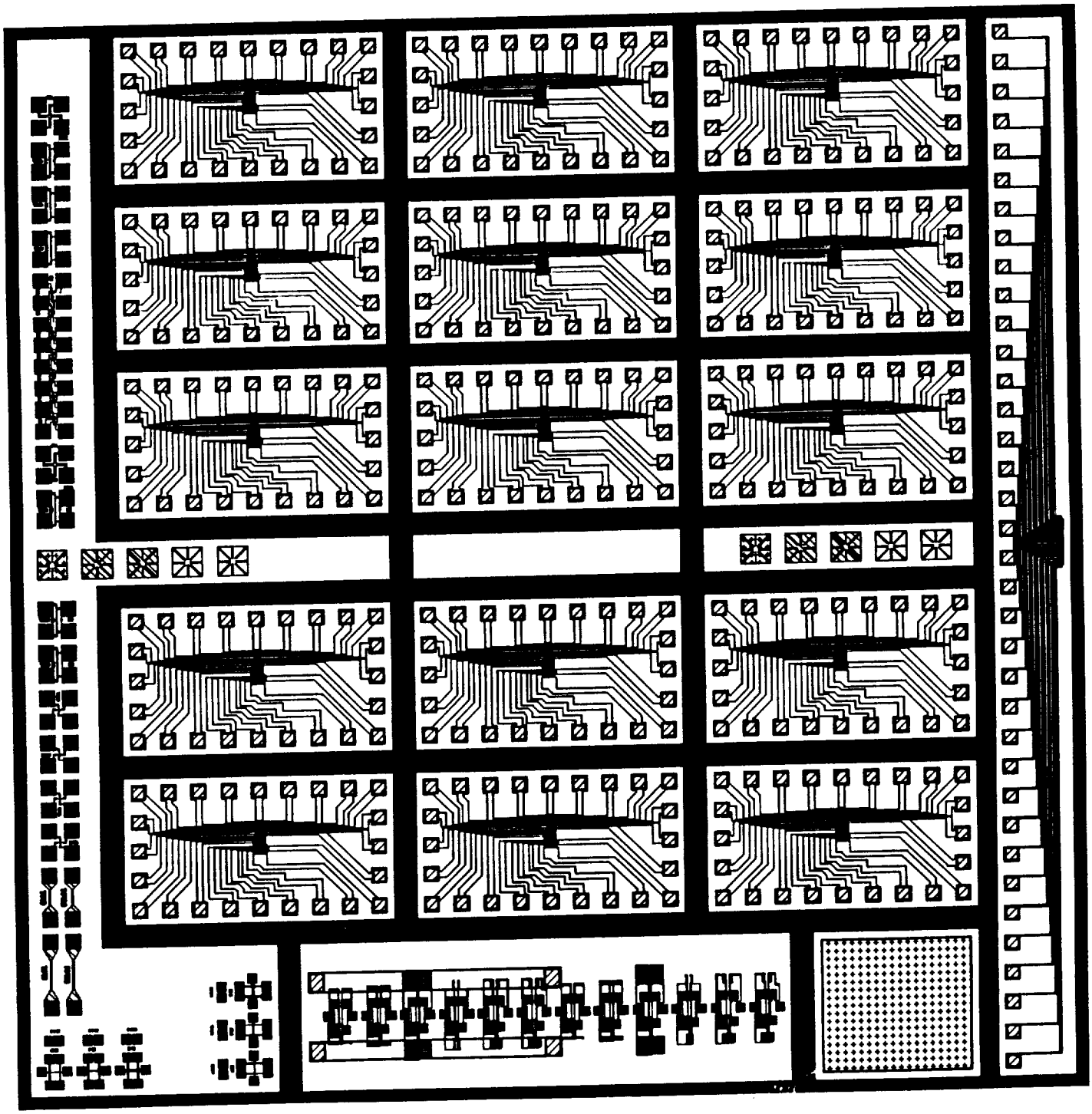
This test chip was designed and laid out during the fifth quarter. NVE has had the masks manufactured, and the first batch of test chips is currently in processing at NVE's lab. During the sixth quarter, NVE plans to test the chips and make whatever design and process modifications are required in order to produce a working bit using GMR materials.

4.2 Test Chip Design

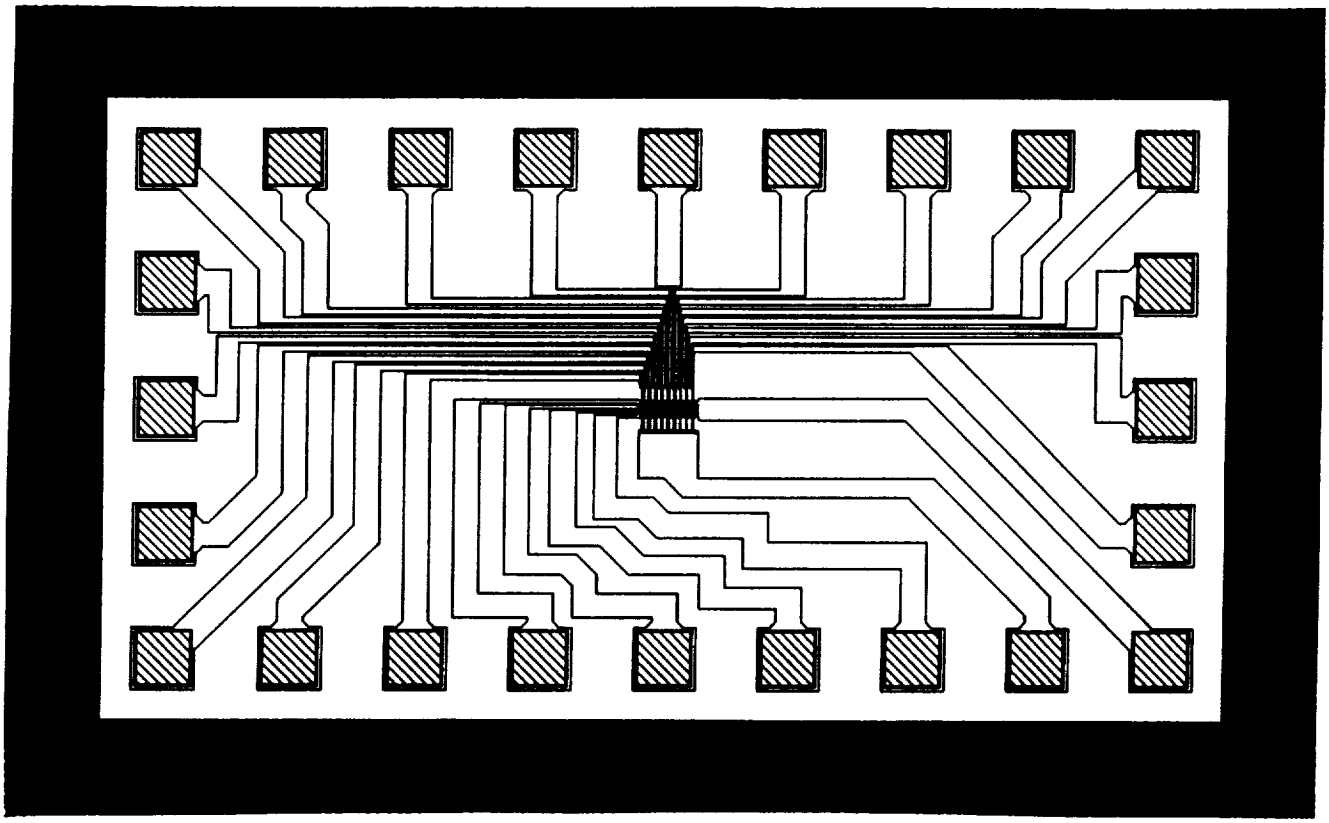
The test chip was designed to incorporate a wide variety of bit configurations, in order to fully characterize an MRAM bit manufactured with GMR materials. Some bits were designed as near duplicates of the ones used by NVE on the 16K MRAM chip, some were designed to be exactly or almost exactly the size required by the 1 Meg bit specification, and some were designed to test the fundamental limits of the materials by being very small. NVE also included some bits that do not have a taper at either end. The taper is normally included to trap magnetic domain walls and prevent them from travelling to the next bit on the string and upsetting it. The bits with no taper could be designed because the GMR material in these bits is not connected together along the bit string; thus the magnetic domains cannot leave the bit. If these particular bits work as designed, they would lead to a large increase in array density, because the taper used on all bits up to this point is a fundamental limit to the density of the array.

The chip also incorporates various testing structures for checking contact and GMR material resistance, mask alignment, metal structure integrity, etc. The following pages contain plots of the entire chip and all the MRAM arrays found on it.

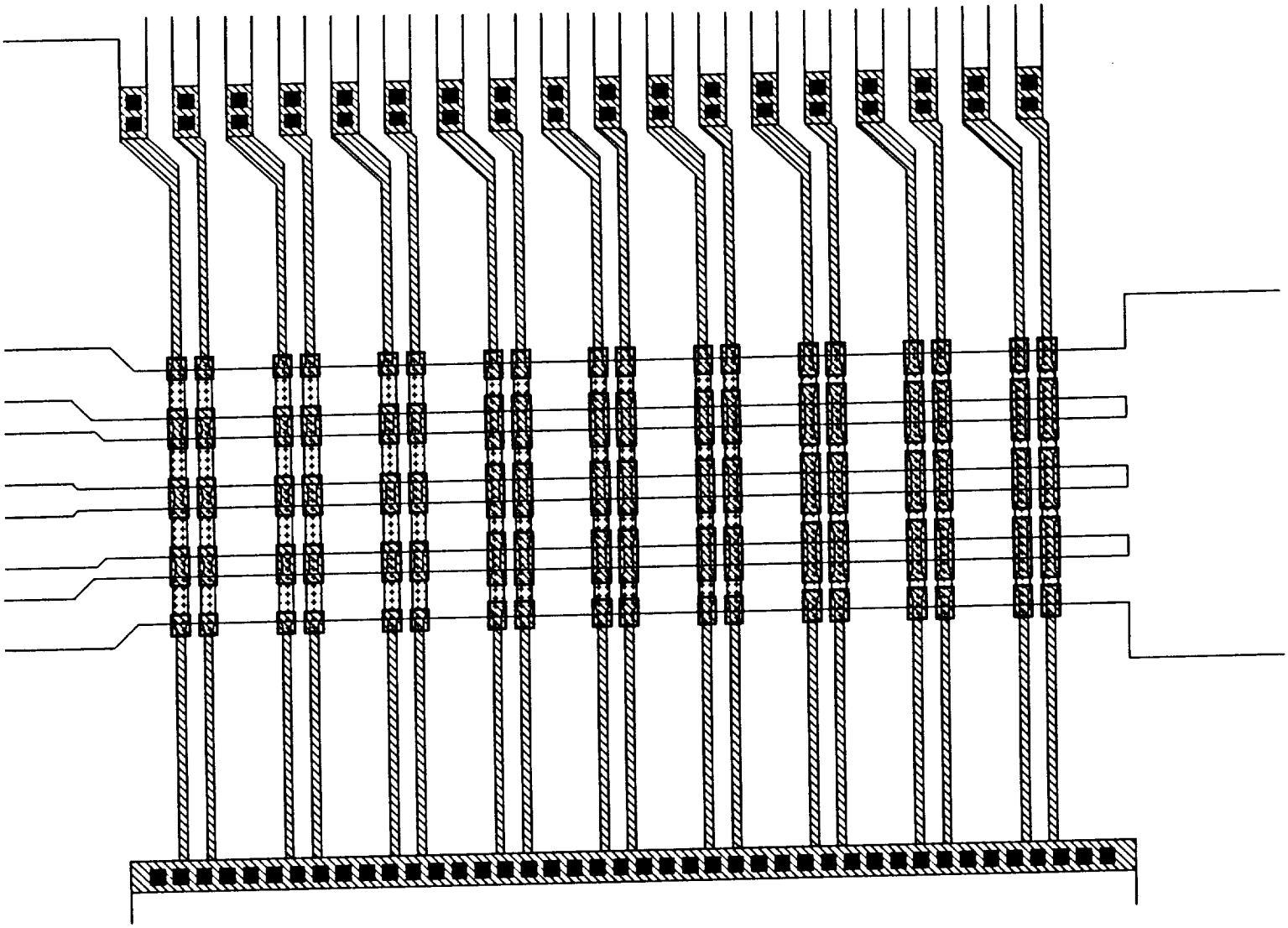
The plot below shows the entire test chip. Material test structures occupy the extreme right and left sides of the chip, as well as the bottom row. The fifteen blocks with bonding pads are the test arrays of different MRAM bits.



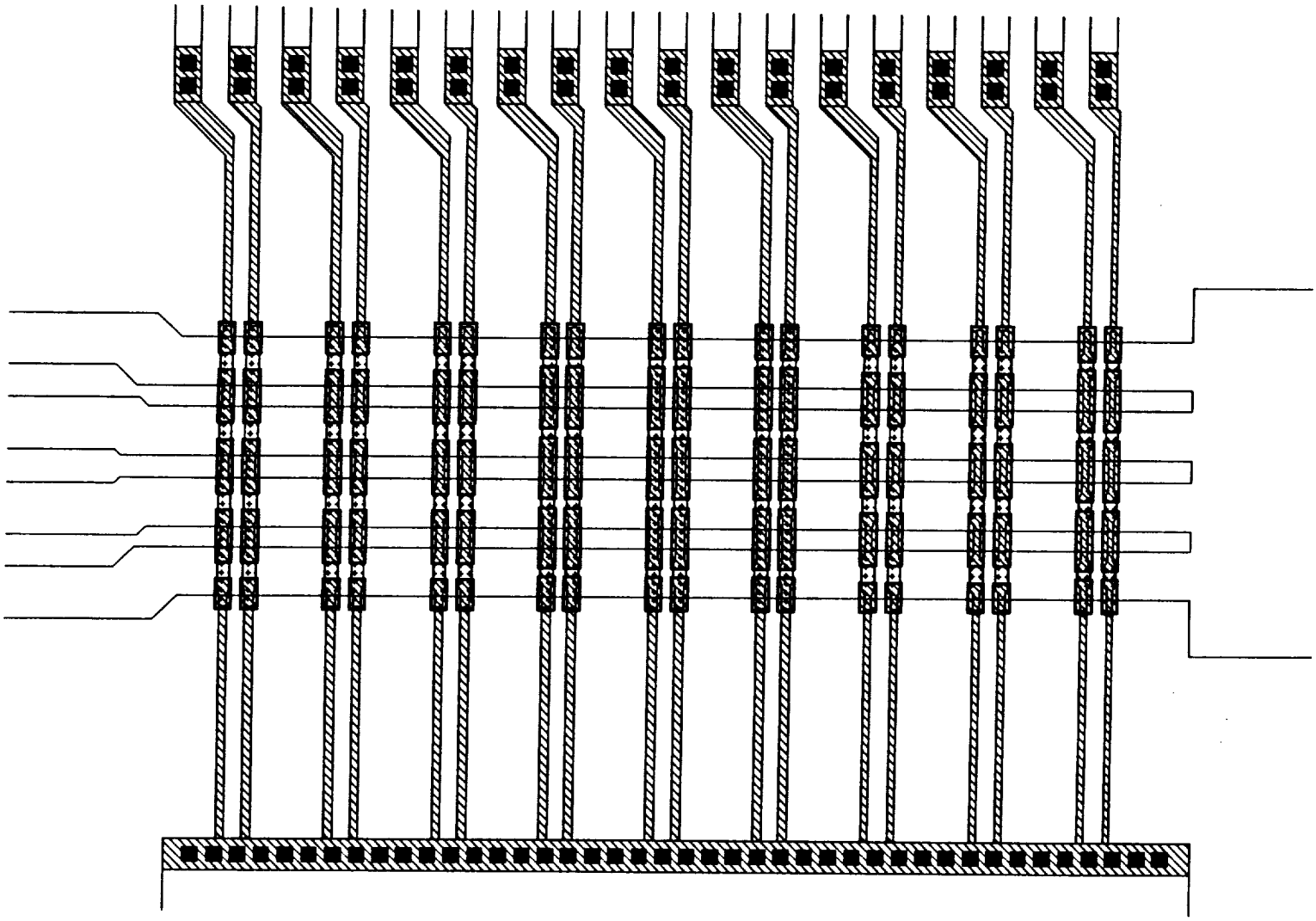
The plot below is a closer look at one of the MRAM test arrays. The dense area in the middle is the array of bits. The I/O pads(for bonding or probing) are clearly visible at the periphery of the array. The pad arrangement matches a standard probe card NVE uses to test die. The dark area around the array is essentially a "street" within the die, where all deposited layers have been stripped away. The die can be cut along these streets, so that individual test arrays can be cut out of the die and packaged for easier handling and testing.



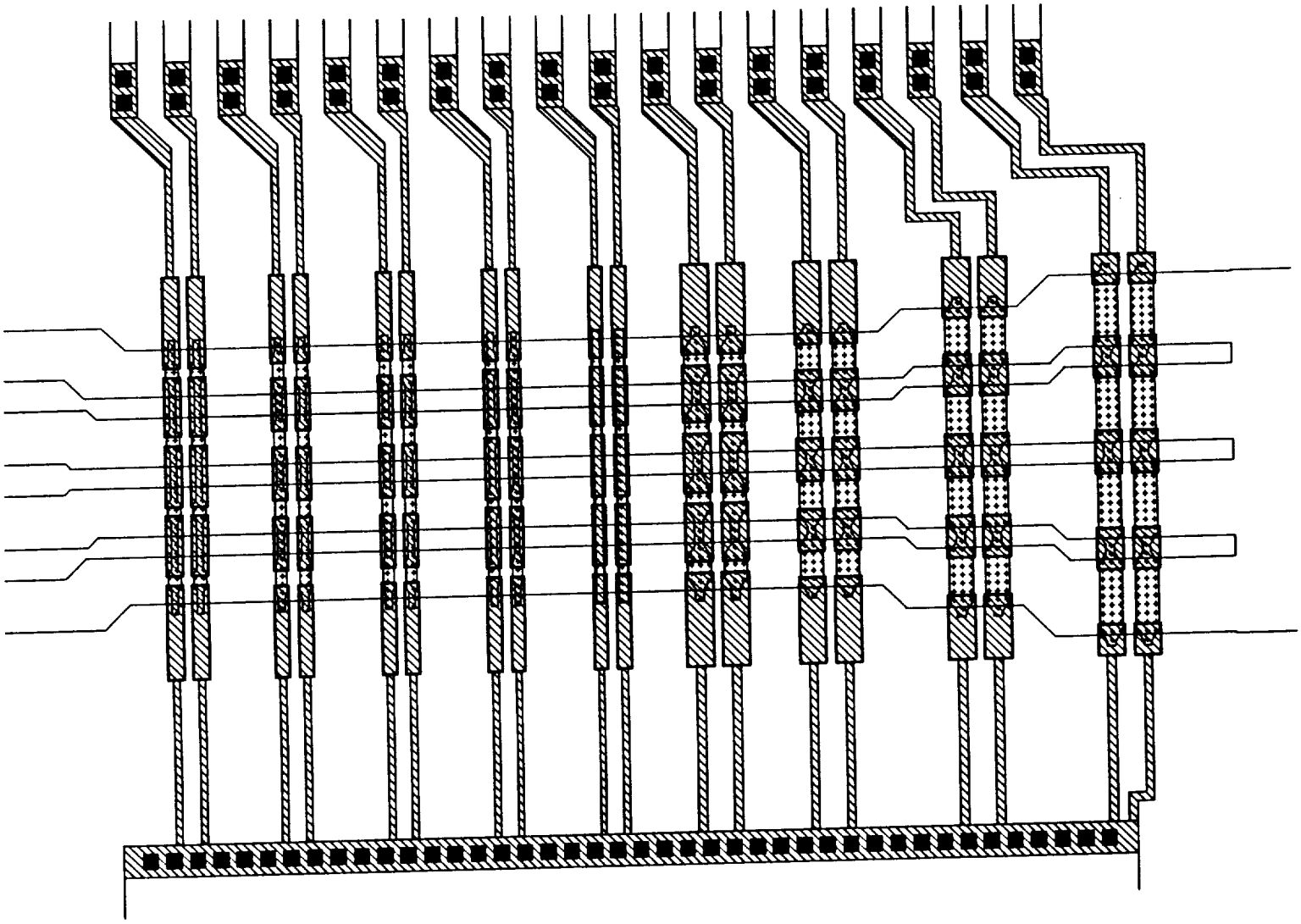
The plot below is a close up of one of the three different MRAM bit arrays designed on this test chip. The sense lines run vertically, and the word lines run horizontally. The MRAM bits shown here are all very similar or the same as the one specified for the 1 Meg. There are nine different bits in this array, and eight copies of each. The dimensions varied in these bits are the length of the taper and the overall length of the bit.



The plot below shows another of the MRAM bit arrays. This array features bits that are the same style as the one specified for the 1 Meg, but the sizes are considerably smaller. The features varied in this array are the neck size, taper length, and overall bit length.



The plot below shows the last of the three MRAM bit arrays. The bits on the left side of this array are very small tapered bits. The bits on the right side of the array are variations on the bit used by NVE in the 16K MRAM. The bits in the very center column are experimental non-tapered bits.



4.3 Processing

A new manufacturing process was developed for use on this test chip. In working closely with Honeywell SSEC on the 16K MRAM chip, NVE has found that the existing process produces a high failure rate of the contacts to the surrounding circuitry. NVE's new process avoids this problem, and in addition allows the use of more industry standard processing steps.

In the existing process, the magnetic sandwich which forms the bit is deposited on the wafer; next, a much thicker layer of Metal-1 is deposited on top of this sandwich. Where a bit is desired, this Metal-1 is etched away, leaving only the magnetic sandwich; the remaining Metal-1 forms a shorting bar, or contact, between the bit and the next bit on the sense line. In the new process, the magnetic sandwich is deposited, and then covered with a passivation layer. This layer is then opened up over the ends of the bits, exposing the magnetic sandwich. Metal-1 is then deposited over these openings, making contact to the magnetic layer. The new process results in less of a step for the contact to the surrounding circuitry, allowing higher yield.

This new process is currently being used in the manufacture of the first lot of test chips. NVE will have more information available about the feasibility of this process as the test chips are produced.

4.4 Conclusion

By the middle of the sixth quarter, NVE hopes to have working GMR bits demonstrated on the test chip, complete with electrical parameter and processing parameter standards. With this information final simulations of the 1 Meg circuitry can be performed, and any modifications required can be made. Also, NVE's new process can be evaluated with an eye towards production of the 1 Meg chip.

This page intentionally blank

SECTION 5

1 MEGABIT

STATUS REPORT

5.1 Introduction

The fifth quarter of this program resulted in a great deal of progress in the design of the memory array, sense amplifier, and drive electronics. Several different sense amplifier designs were evaluated and the best one for this task was selected. Some of the criteria used in its selection were simplicity, wide bandwidth, insensitivity to wide power supply variations, stability, impact on the overall chip size, SNR, noise rejection, and scale ability to 3 volts. Once the sense amplifier was selected and designed, the inner section or segment of the array was laid out. Several innovations have occurred during this design which have resulted in chip size reduction and noise reduction. The concept of using an unselected sense line for a reference during a read instead of extra dummy sense lines has resulted in a chip area savings. Crossing sense select lines half way up the array column with adjacent columns has resulted in transforming the word line to sense line coupling from a differential sense noise signal to a common mode sense noise signal at one half the amplitude. This has resulted in an increase in speed since it is not necessary to wait for the noise to settle.

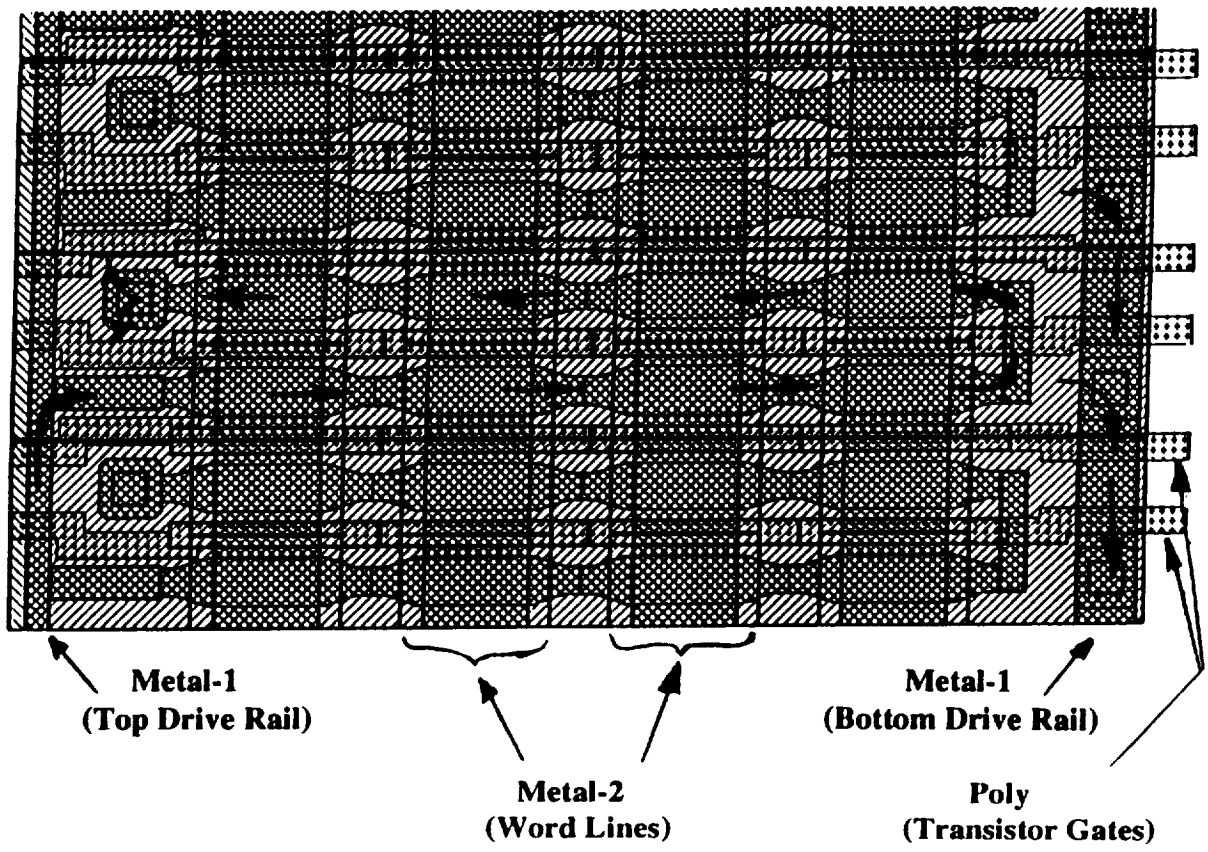
With the design improvements described above, it was possible to lay out the system timing. It was determined that the read cycle would take 250 nanoseconds and the write cycle would be 100 nanoseconds. One of the areas of concern is the accurate representation of the parasitic effects of unselected sense lines during simulations. This concern has been addressed and has resulted in a very accurate model of these parasitic effects which can be used in simulations of the drive and sense circuitry. This model includes all coupling and stray capacitances as well as element and interconnected resistances.

The memory array, drivers, decoders, sense preamplifier, and buffers along with the sense and word drivers have been designed and laid out. The initial design of the chip is two thirds complete, and LVS checks have been initiated.

5.2 Parasitic Equivalent of Unused Sense Lines in the 1 Megabit Array

The operation of the memory array in each segment of NVE's 1 megabit memory chip is influenced to a large extent by the parasitic load of unused sense lines. When a sense line is turned on in a given segment of the array, the parasitic resistances and capacitances of the sense lines which are not selected affects the rise time, or settling time of the array. Each current "driver" supplies sense line current to one sense line which is turned on and 128 which are turned off. The purpose of this investigation was to determine a worst case loading for a given current driver.

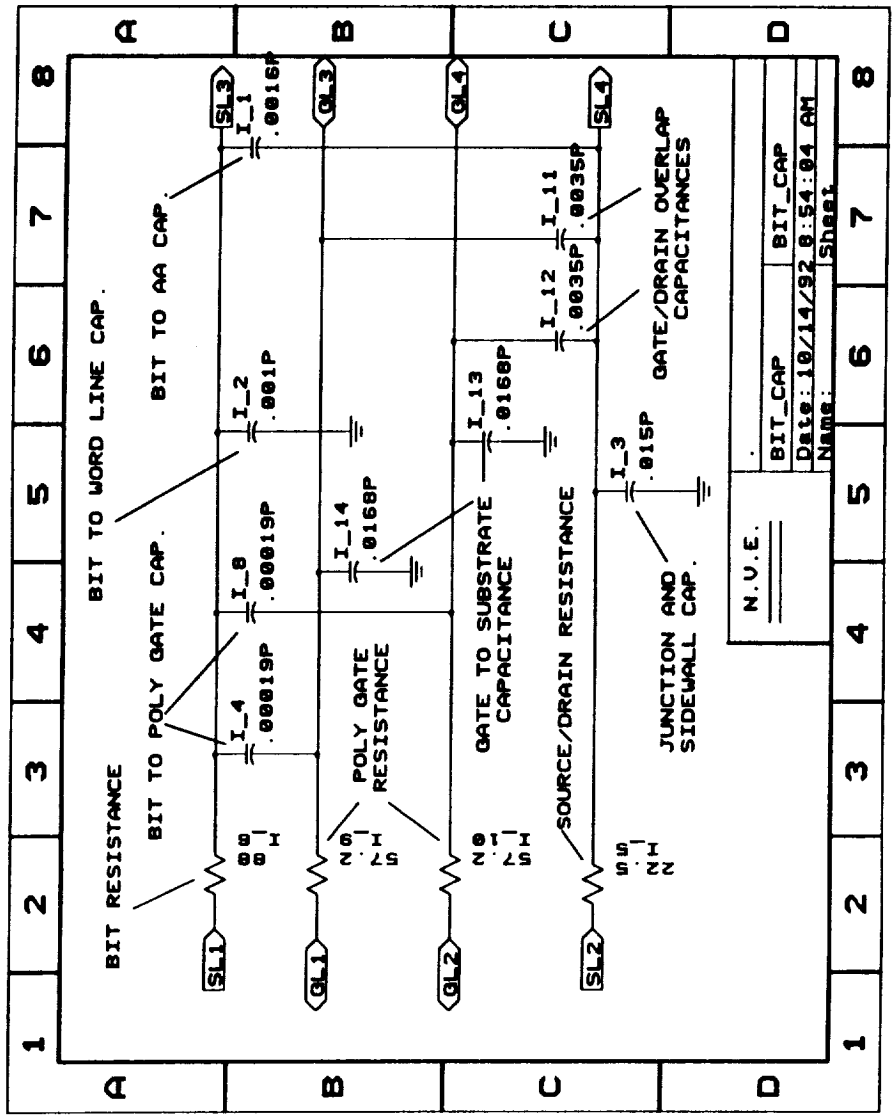
The 1 megabit memory array was laid out by NVE to be as dense as possible. It uses buried gate transistors as switches to turn sense lines on and off. A section of the memory array is shown below; a current path through one of the sense lines is drawn on the array:

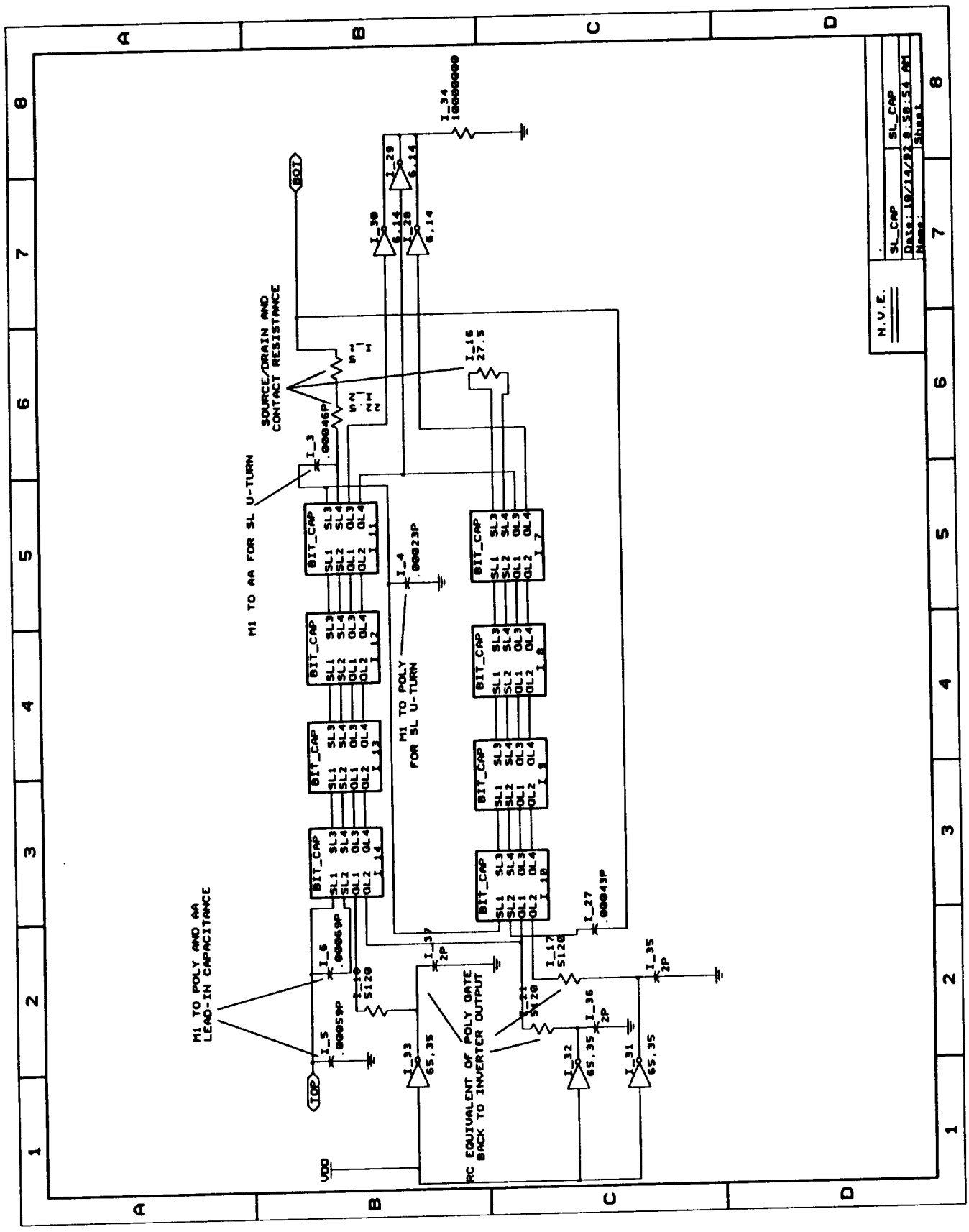


Note that when the two poly lines on either side of the bit string contact are driven to the supply voltage, two n-channel transistors are turned on in parallel to allow the sense line current to flow through the sense line and out the bottom drive rail(the metal-1 line on the left was arbitrarily picked as the top drive rail, and the metal-1 line on the right is the bottom drive rail). The parasitics seen by each driver are the result of the current charging the capacitors in the unused sense lines through the unused sense line resistances.

These parasitics must be modeled in two current directions, because operation of the chip requires sense line current in the forward direction(top drive rail to bottom drive rail) and the reverse direction(bottom drive rail to top drive rail) during a read operation. Since the array is not symmetrical left to right, the parasitics seen by the drivers will differ depending on the current direction.

An electrical model of an individual bit was developed which took into account all the parasitic resistances and capacitances. This model was developed using worst case parameters and guidelines from the ATMEL 0.8 micron CMOS process, which NVE will use to manufacture the 1 megabit chip. This bit model was then used to form an eight bit sense line, and the peripheral resistances and capacitances needed to complete one parasitic sense line were added to the eight bits to complete the model. The schematics for the bit and the sense line are shown on the following pages.


BIT RESISTANCE
BIT TO POLY GATE CAP.
BIT TO WORD LINE CAP.
BIT TO AA CAP.
POLY GATE RESISTANCE
GATE TO SUBSTRATE CAPACITANCE
SOURCE/DRAIN RESISTANCE
JUNCTION AND SIDEWALL CAP.
GATE/DRAIN OVERLAP CAPACITANCES
I_1 .0016P
I_2 .001P
I_3 .015P
I_4 .00019P
I_8 .00019P
I_11 .0035P
I_12 .0035P
I_13 .0168P
I_14 .0168P
SL1
SL2
SL3
SL4
GL1
GL2
GL3
GL4
N.U.E.
BIT_CAP
BIT_CAP
Date: 10/14/92 8:54:04 AM
Sheet


SL_CAP
Date: 10/14/92 8:58:54 AM
Sheet
Name:
N.V.E.
SOURCE/DRAIN AND CONTACT RESISTANCE
M1 TO AA FOR SL U-TURN
M1 TO POLY FOR SL U-TURN
M1 TO POLY AND AA LEAD-IN CAPACITANCE
RC EQUIVALENT OF POLY GATE BACK TO INVERTER OUTPUT
BOT
TOP
VDD
BIT_CAP

These schematics were then transferred into HSPICE decks for electrical analysis. The approach taken was to run a voltage source into one end of the parasitic sense line, and observe the behavior of the current. An identical voltage source was used to power a simple RC combination at the same time. The values of the resistor and the capacitor in the RC combination were varied until the behavior of the current through the parasitic sense line and the RC combination was as close to identical as possible.

The HSPICE decks used for the simulations in the forward and reverse current directions are shown below:

```
SENSE LINE CAPACITANCE HSPICE SIMULATION - FORWARD SL CURRENT
VDD VDD GND DC 5
VSS VSS GND DC 0

VTOP TOP GND PWL(0N 0 10N 0 11N 2.2 30N 2.2)
VBOT BOT GND DC 0

VCHARGE CHARGE GND PWL(0N 0 10N 0 11N 2.2 30N 2.2)
RCHARGE CHARGE CAPV REQ
CCAP CAPV GND CEQ

.PARAM CEQ=.105P  REQ=610
.TEMP 25

.INCLUDE 'C:\NVE\1MEG\SPI_SCH\SL_CAP.SPI'
.INCLUDE 'C:\NVE\1MEG\SPICE\ATMEL-8.NPR'

.TRAN 1N 30N *SWEEP CEQ .106P .103P .001P

.OPTIONS RELTOL=.0003 LVLTIM=2 CHGTOL=1E-15 POST=1 SCALE=1U

.END
```

```
SENSE LINE CAPACITANCE HSPICE SIMULATION - REVERSE SL CURRENT
VDD VDD GND DC 5
VSS VSS GND DC 0

VBOT BOT GND PWL(0N 0 10N 0 11N 2.2 30N 2.2)
VTOP TOP GND DC 0

VCHARGE CHARGE GND PWL(0N 0 10N 0 11N 2.2 30N 2.2)
```

```
RCHARGE CHARGE CAPV REQ
CCAP CAPV GND CEQ

.PARAM CEQ=.097P  REQ=43
.TEMP 25

.INCLUDE 'C:\NVE\1MEG\SPI_SCH\SL_CAP.SPI'
.INCLUDE 'C:\NVE\1MEG\SPICE\ATMEL-8.NPR'

.TRAN 1N 30N *SWEEP REQ 55 35 5

.OPTIONS RELTOL=.0003 LVLTIM=2 CHGTOL=1E-15 POST=1 SCALE=1U

.END
```

Plots of the current through the parasitic array and the RC combination, in both current directions, are shown at the end of this section.

The final values for the equivalent resistance and capacitance of one sense line for both current directions, as can be seen in the decks, are:

Forward Sense Line Current: Equivalent Resistance = 610 Ohms
Equivalent Capacitance = .105 pF

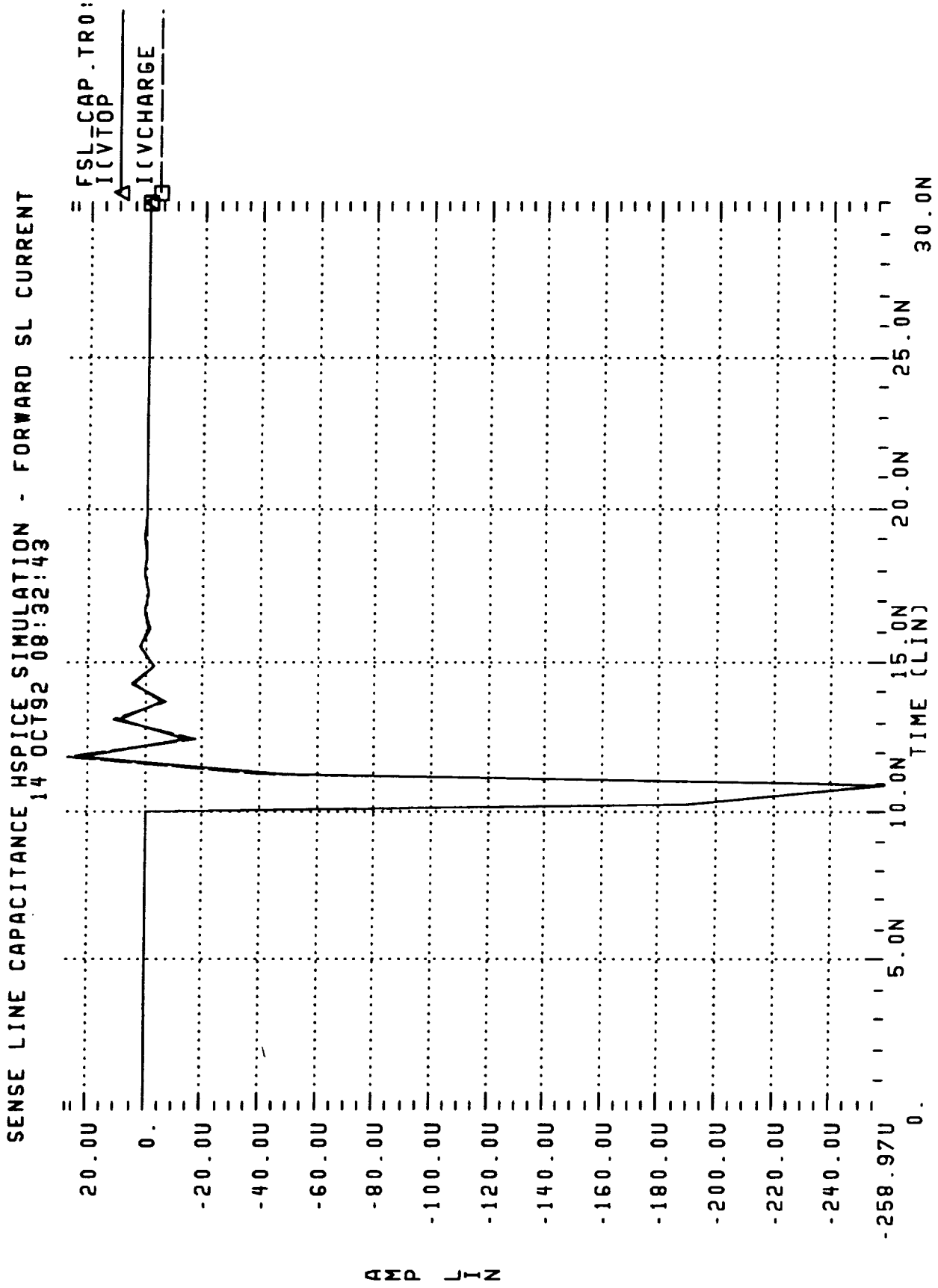
Reverse Sense Line Current: Equivalent Resistance = 43 Ohms
Equivalent Capacitance = .097 pF

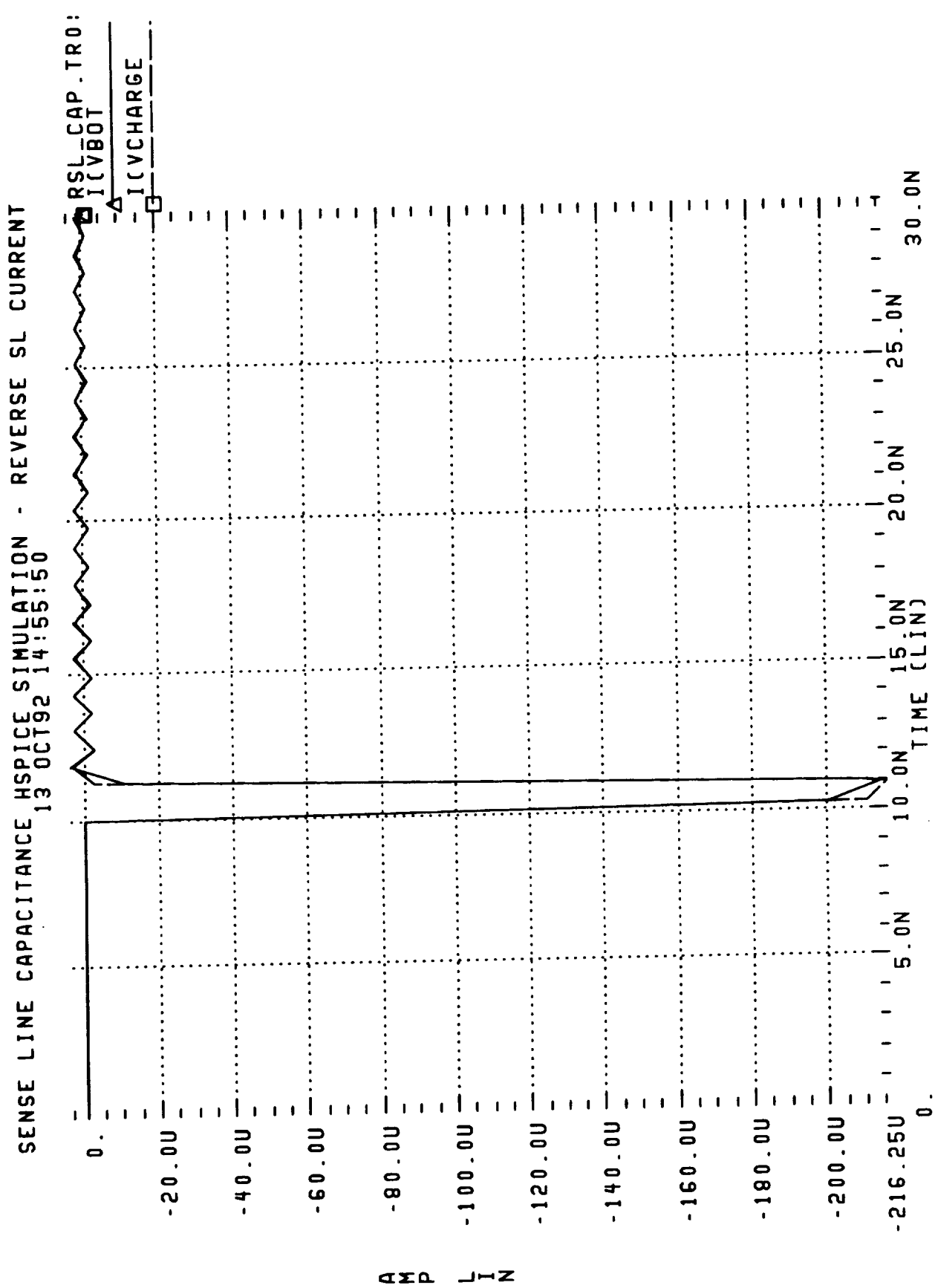
Each driver sees 128 of these equivalent RC loads in parallel when it tries to drive a sense line, so the total parasitic load is equal to the parallel combination of 128 of these elements. These total loads are as follows:

Forward Sense Line Current: Equivalent Resistance = 4.77 Ohms
Equivalent Capacitance = 13.44 pF

Reverse Sense Line Current: Equivalent Resistance = .336 Ohms
Equivalent Capacitance = 12.42 pF

These values will be used for all electrical simulations involving the sense line driver circuitry, and also for simulations which quantify the word line coupling effect.

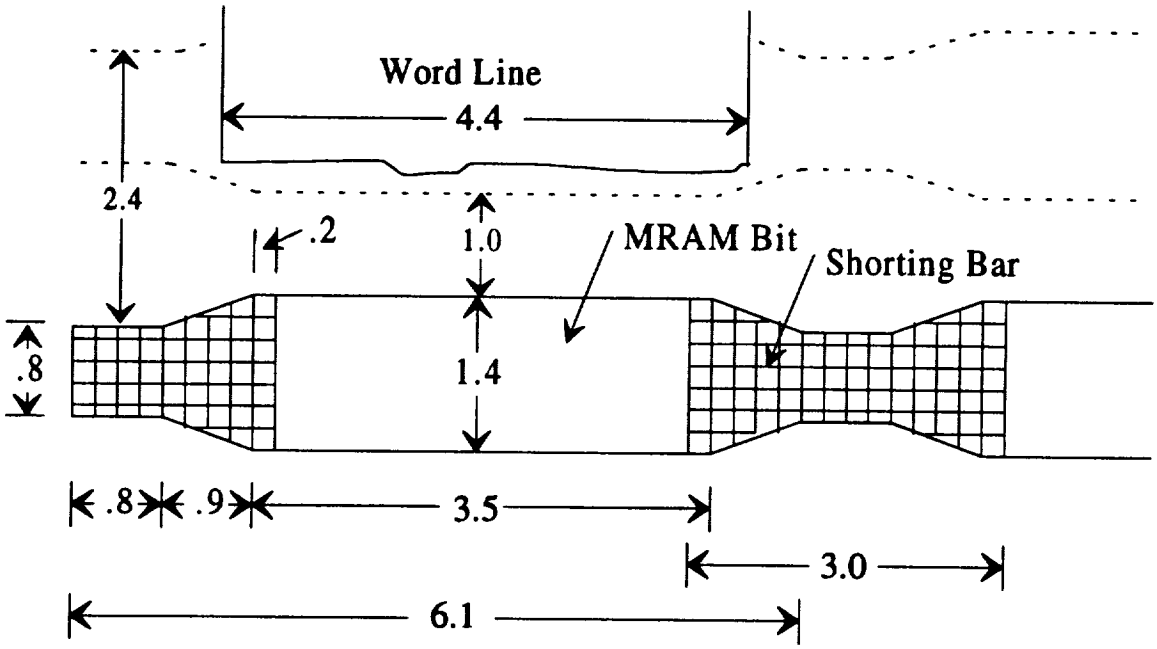

SENSE LINE CAPACITANCE HSPICE SIMULATION - FORWARD SL CURRENT
14 OCT92 08:32:43
FSL_CAP.TR0:
I(VTOP
I(VCHARGE
20.0U
0.
-20.0U
-40.0U
-60.0U
-80.0U
-100.0U
-120.0U
-140.0U
-160.0U
-180.0U
-200.0U
-220.0U
-240.0U
-258.97U
AMP LIN
0.
5.0N
10.0N
15.0N
20.0N
25.0N
30.0N
TIME (LIN)


SENSE LINE CAPACITANCE HSPICE SIMULATION - REVERSE SL CURRENT
13 OCT92 14:55:50
RSL_CAP.TR0:
I(VBOT
I(VCHARGE
0.
-20.0U
-40.0U
-60.0U
-80.0U
-100.0U
-120.0U
-140.0U
-160.0U
-180.0U
-200.0U
-216.25U
0.
5.0N
10.0N
15.0N
20.0N
25.0N
30.0N
TIME (LIN)
AMP LIN

5.3 MRAM Bit Specification

During the design of the array, it was found that the bit could be made a slightly smaller while still retaining an acceptable signal level. This results in a significant savings in chip area. This change is shown here.

1 Megabit MRAM Bit Specification



*All Dimensions in Microns
*Sense Line Current = 2.5 mA
*Bit Resistance = 80 Ohms
*Nominal Signal Size = 3 mV

Note that this cell has tapered ends, uses about 2.5 squares of GMR material for resistance, and has a width of 1.4 microns. Because the magnetic films are thinner than in the 16K cell (about 50 Angstroms instead of 150 Angstroms), the curling of the edge spins is limited to about 0.25 microns from the edge as compared to about 0.4 microns for the current 16K cell. The shorting bars are tapered with a 1:3 slope in a similar fashion to the current cell. The word line overlaps the bit ends to ensure a larger word field at the end of the cells.

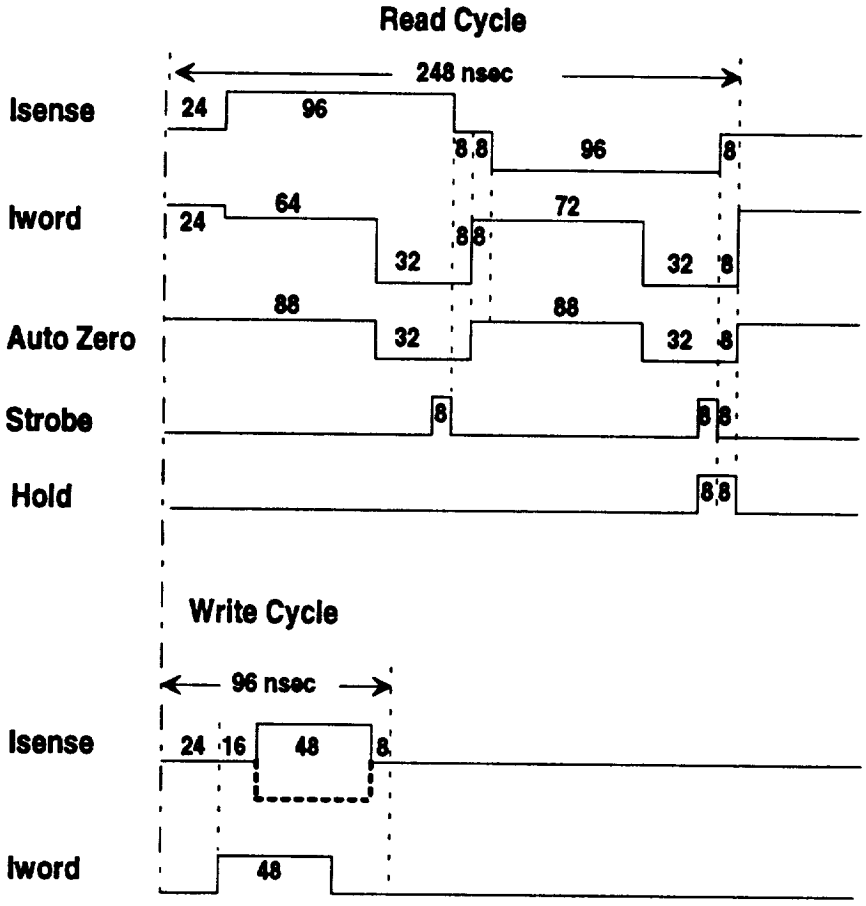
The resistance of the cell is 80 Ohms, about 61.5 Ohms due to series cell resistance and 18.5 Ohms due to contacts. The expected signal size is about 3 mV, which at 2.5 mA and 6% usable magneto resistance represents 22 percent of the total voltage drop across the cell. This is a typical signal, and "worst-casing" to allow for signal distributions will be accounted for in the signal/noise calculations being made in circuit design.

5.4 System Timing

The read cycle is initiated by word (negative) and sense currents (four positive currents for four bits, or a nibble) coming on together after decoding of sense and word lines. The initial value of the word current will be about 8 mA while the sense system is auto zeroed. The sense current is 2.43 mA per sense line or 9.75 mA per nibble. After auto zeroing, the word read current is increased to 30 mA. The sense current is turned off first followed by the word current.

The write cycle is initiated by turning on a word (positive) and sense (positive or negative, depending on data) currents. The writing values of sense current are plus or minus 2.0 mA, and the writing values for word current is 20 mA. Fall times for the write signals are not critical, but it is somewhat preferable for the sense current to fall last. About 120 ns recovery time is estimated after fall of the sense current.

The internal timing diagrams used for reading and writing are shown on the next page. These timings have been revised because of design improvements and innovations resulting in much faster read - write times. The preliminary product specification has been completed and is contained in the appendix. The pin assignments for the part are shown in the specification. Since it is organized as a 256K x 4 MRAM, it will require 32 pins. If the test features were removed from the part as well as the low power feature, the pin count could be reduced to 28 pins. However, due to it's present size, it would not fit in a 28 pin, 300 mil package. It was decided to provide test features on the 1Meg similar to the test features on the 16k part. This would allow the use of the 16K test equipment for evaluation of the 1 Megabit part.

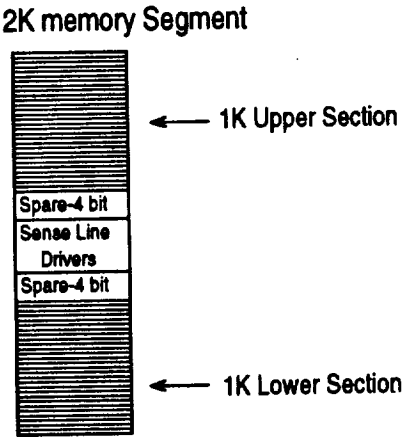


5.5 1 Megabit Chip Architecture

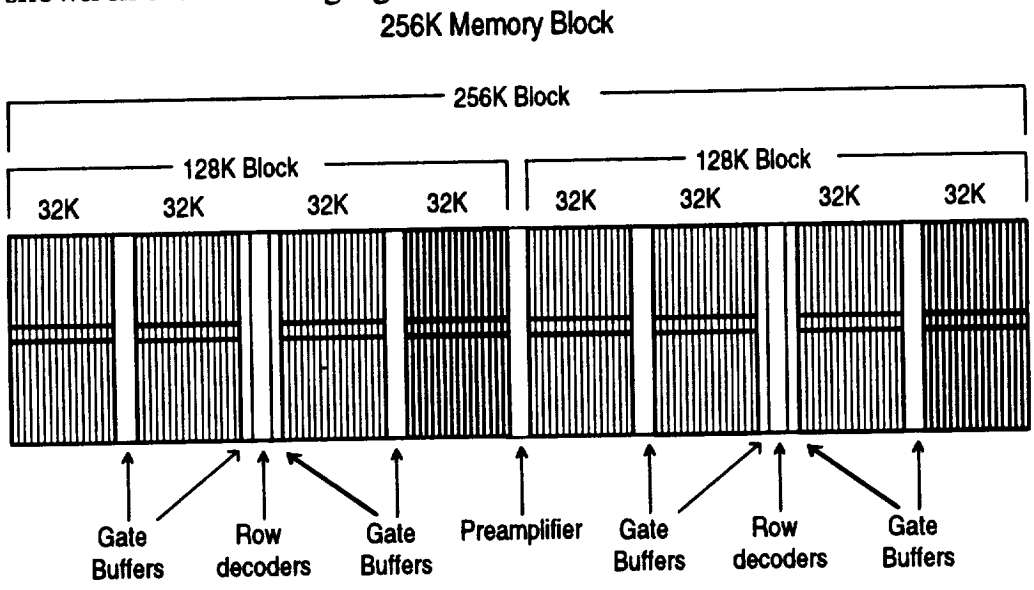
Given that the 1 Megabit MRAM chip should be as dense as possible, and that the memory array will occupy the largest percentage of the chip area, the basic architecture of the chip was designed around the densest possible memory array. During the design process, it was determined that to achieve maximum density the flexibility of going from dual redundant to non redundant with a simple metal mask change was not feasible without penalty to die size. Therefore, it was decided to eliminate this feature.

5.5.1 2K Memory Segment

The basic 2K memory segment is built from 4 bit sense lines. Each segment consists of a 1K upper and a 1K lower section which are separated by a spare 4 bit sense line for each section and the sense line drivers. Each 1K block contains 256 - 4 bit sense lines. The segment lay out is show in the following figure.

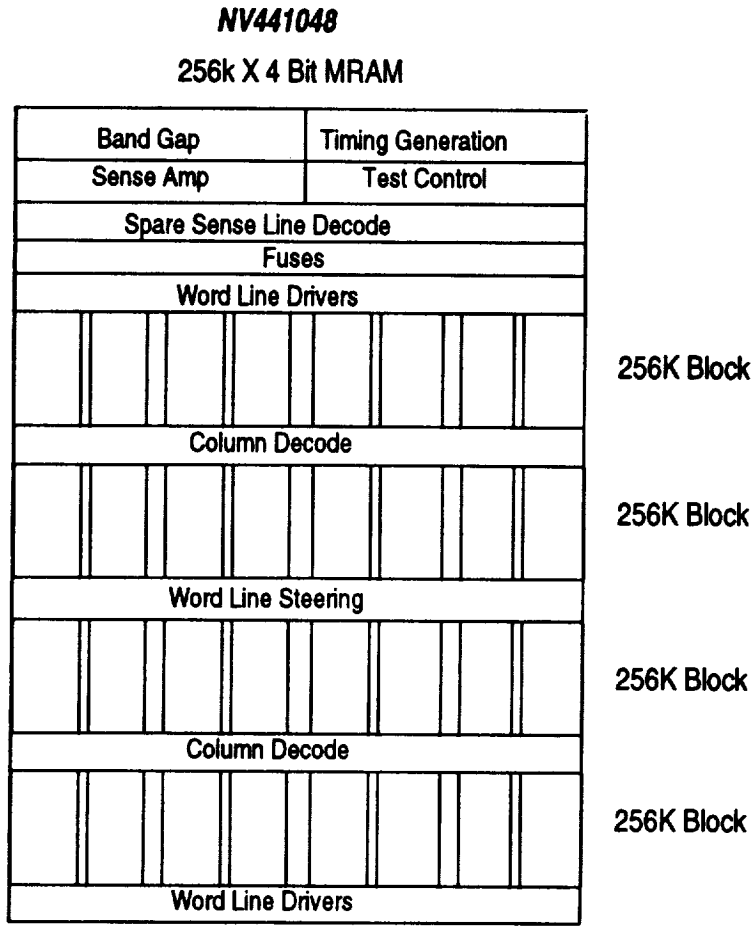


The array is formed in a modular fashion by combining 16 of the 2K segments to form a 32K block. These blocks are separated by gate line buffers. It is necessary to rebuffer the gate lines every 16 segments and to locate their respective decoder in the middle of each 128K block in order to keep propagation delays to a minimum. The preamplifier for 2 of the 128K blocks is located in the middle of them to minimize propagation delays. This is shown in the following figure.



The 256K blocks are next combined to form an array resulting in a 256K x 4 bit architecture. The column decode for the word lines is located between two lower 256K blocks and two upper 256K blocks, again to minimize propagation delay and IR drop. Each 256K block has a set of word line drivers associated with it. The word line steering circuitry is located in the middle of the four - 256K blocks. The rest of the miscellaneous circuitry such

as the Band Gap, Sense Amplifier, Test Control, Timing generators, Spare Line Decoders, and Fuses are located at the top of the chip. This is shown in the following diagram.



The schematic diagrams for the 256K blocks are contained in the appendix along with the definitions for the signal names used.

Much of the peripheral support circuitry can be used from NVE's 16K MRAM part. These circuits include the Bandgap, I/O buffers, self-test, and trim circuitry. A particularly important side benefit of this approach will be the ability to test the 1 Megabit part in the same manner as the 16K, using the same test equipment. NVE has developed test equipment which allows the output of each individual MRAM bit on the chip to be observed. A bad bit can be readily identified, and then visually observed under a Scanning Electron Microscope to determine why it has failed. These failure analysis techniques have been developed to a high degree, and have been instrumental in taking NVE's 16K part from prototype to production status.

These same failure analysis techniques can be applied to the 1 Megabit once it is produced, because it will contain the same self-test system as the 16K.

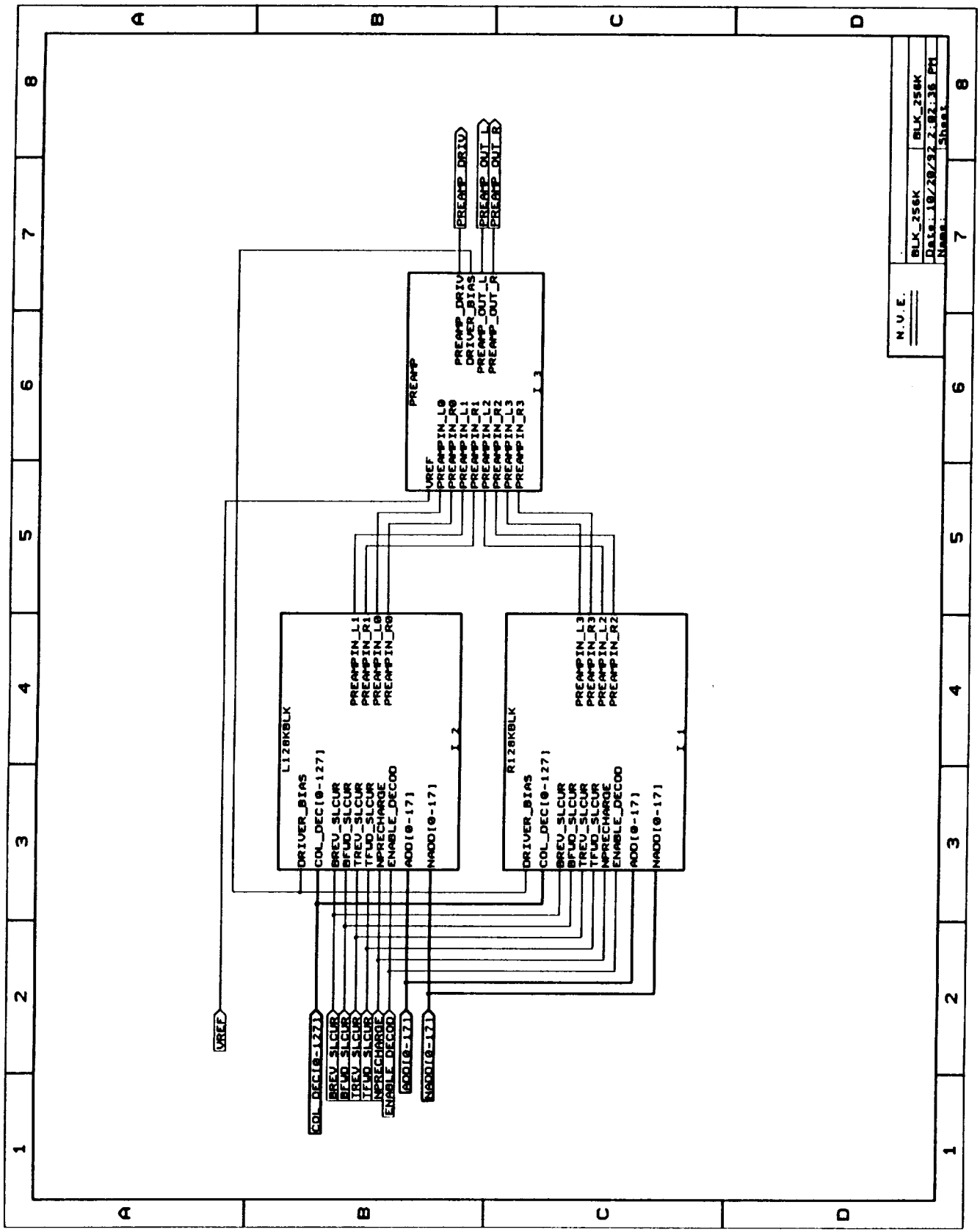
The sense amplifiers used in the 1 Megabit MRAM will be the only analog circuits on the chip which will be significantly different than those on the 16K. NVE has investigated four different sense amp designs for possible use in the 1 Meg and has selected the best one for the task as discussed earlier in this report.

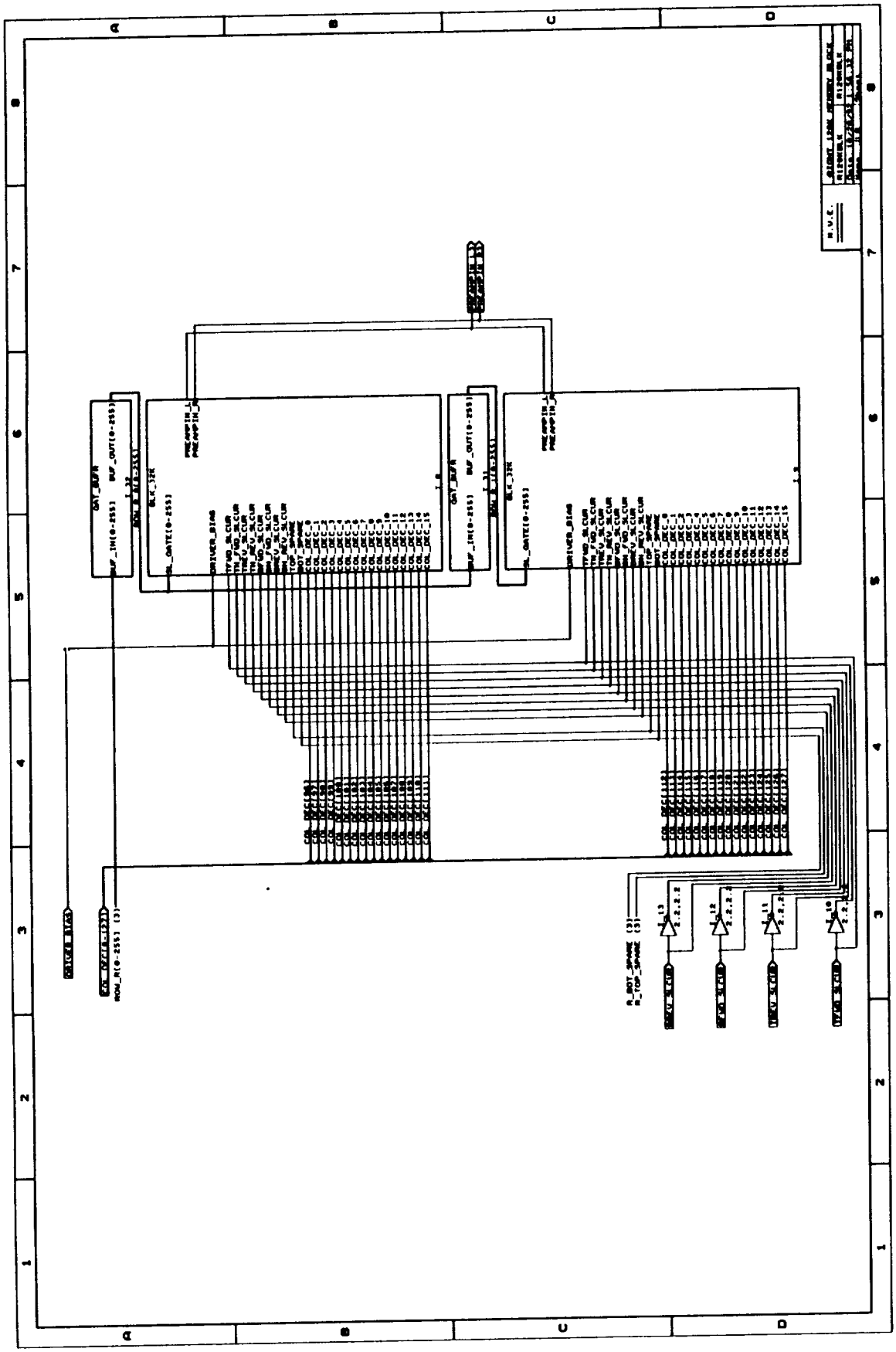
5.6 Conclusion

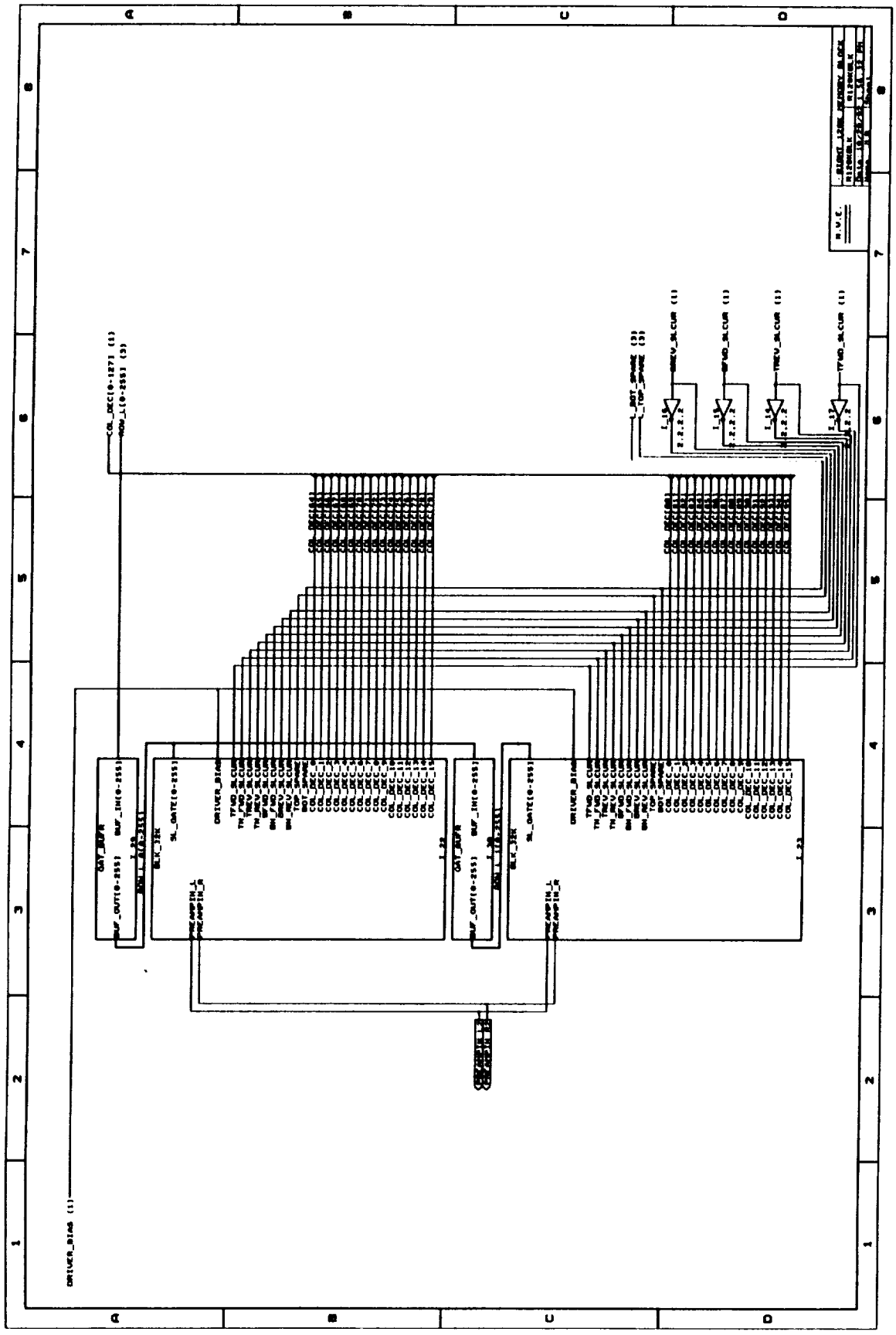
Significant progress was made this quarter on the 1 Megabit design effort and a large portion of the chip has been laid out. Many innovations during this quarter have resulted in a dense chip as well as circuit designs to improve performance and density. By the end of the next quarter, NVE plans to have completed the design and layout of the chip.

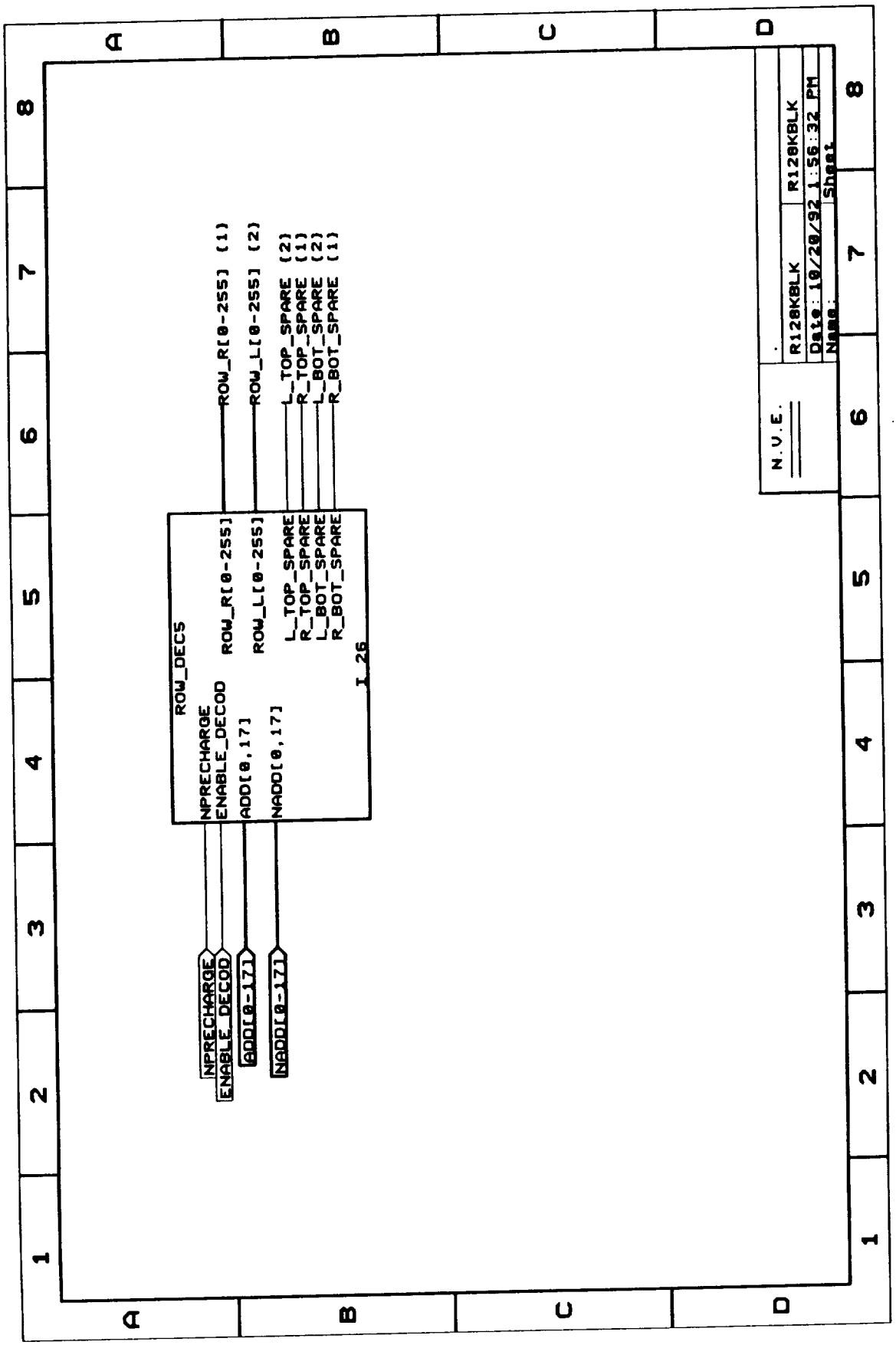
5.7 Appendix

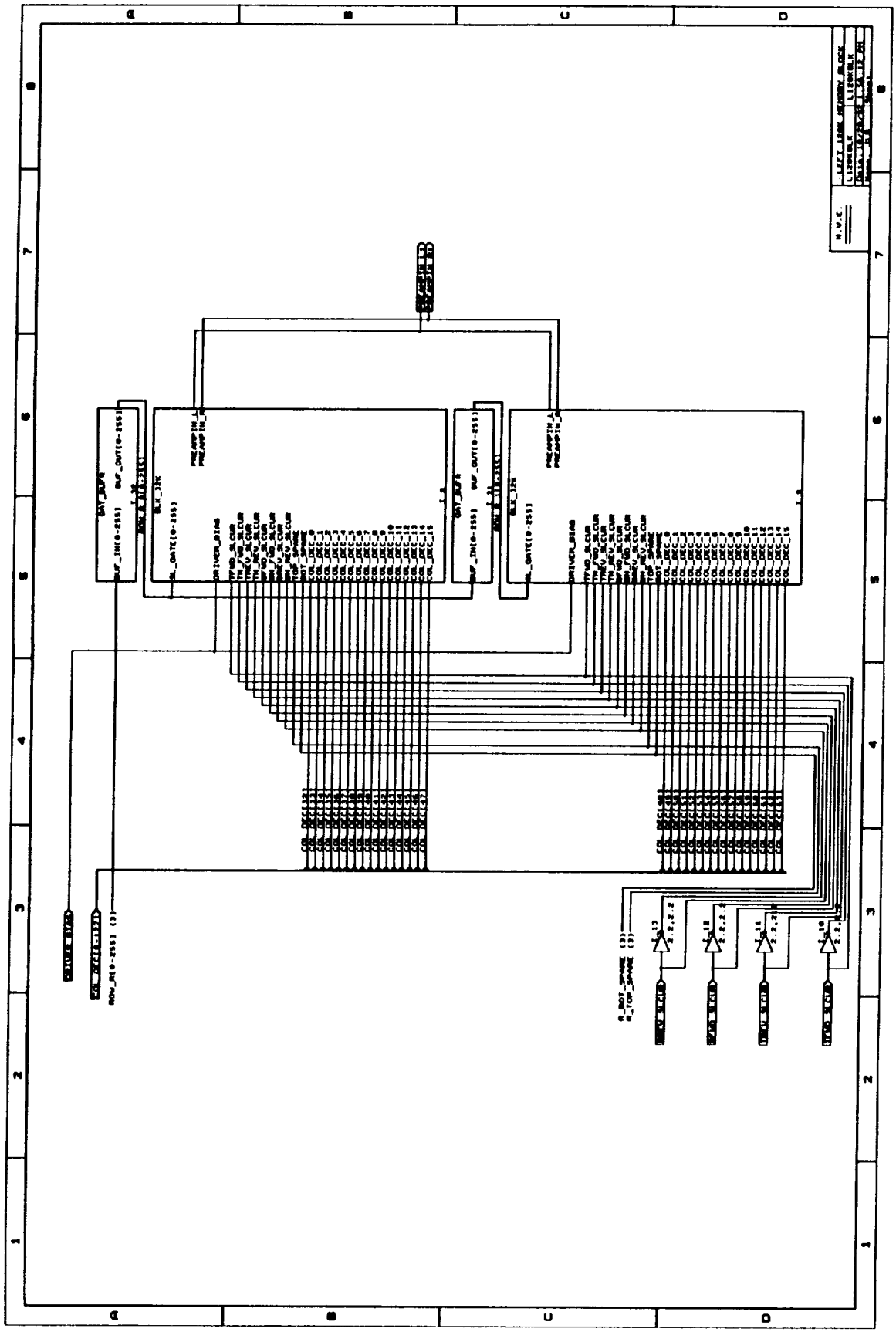
The appendix to this section contains the schematics for the 256K block, the definitions for the signal names used, and the NVE MRAM specifications.

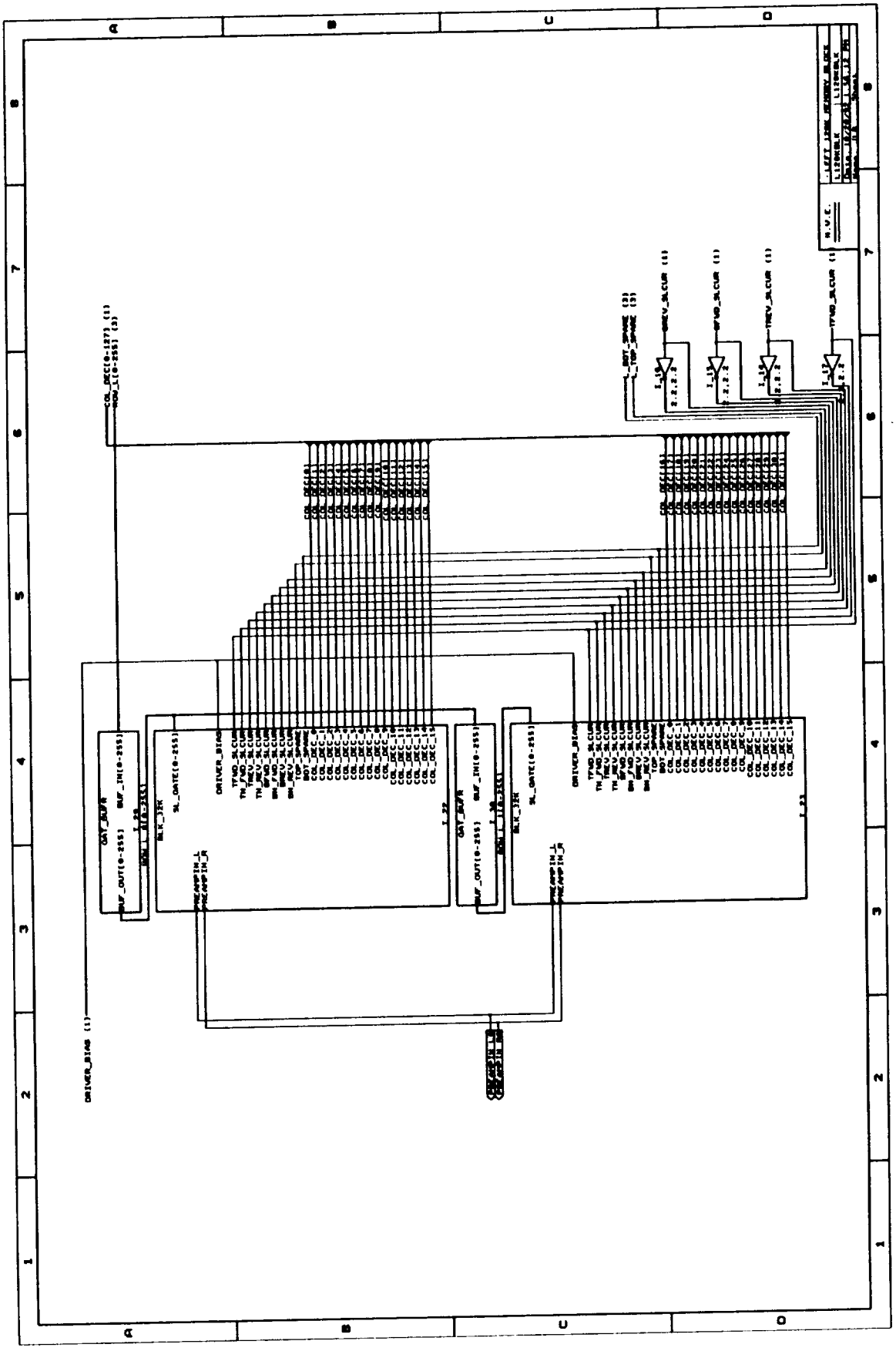

A
B
C
D
1
2
3
4
5
6
7
8
PREAMP_DRIV
PREAMP_OUT_L
PREAMP_OUT_R
PREAMP
I3
VREF
PREAMPIN_L0
PREAMPIN_R0
PREAMPIN_L1
PREAMPIN_R1
PREAMPIN_L2
PREAMPIN_R2
PREAMPIN_L3
PREAMPIN_R3
DRIVER_BIAS
L128KBLK
I2
COL_DEC[0-127]
BREV_SLCUR
BFWD_SLCUR
TREV_SLCUR
TFWD_SLCUR
NPRECHARGE
ENABLE_DECOD
ADD[0-17]
NADD[0-17]
R128KBLK
I1
N.V.E.
BLK_256K
Date:10/20/92 2:02:36 PM
Sheet

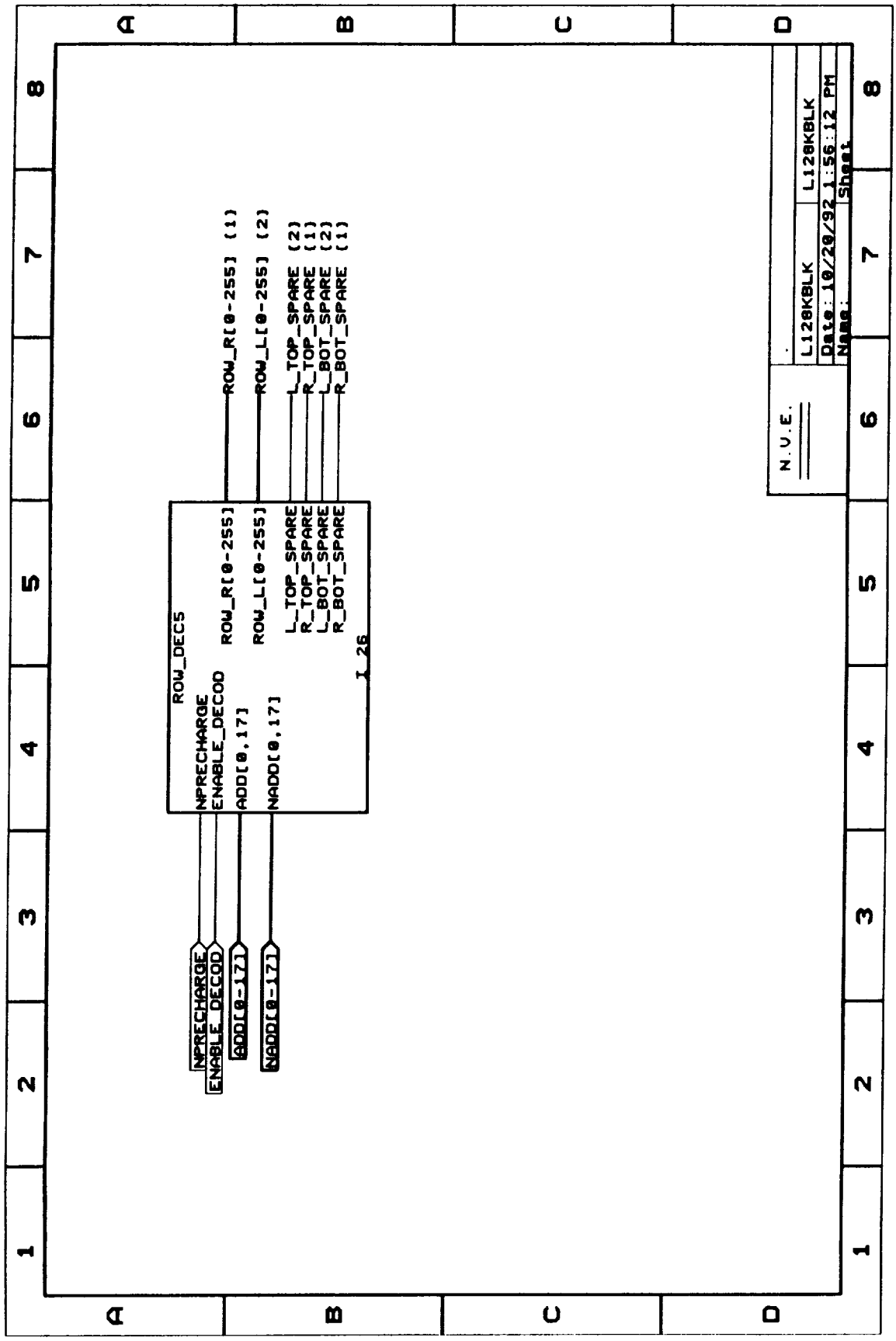


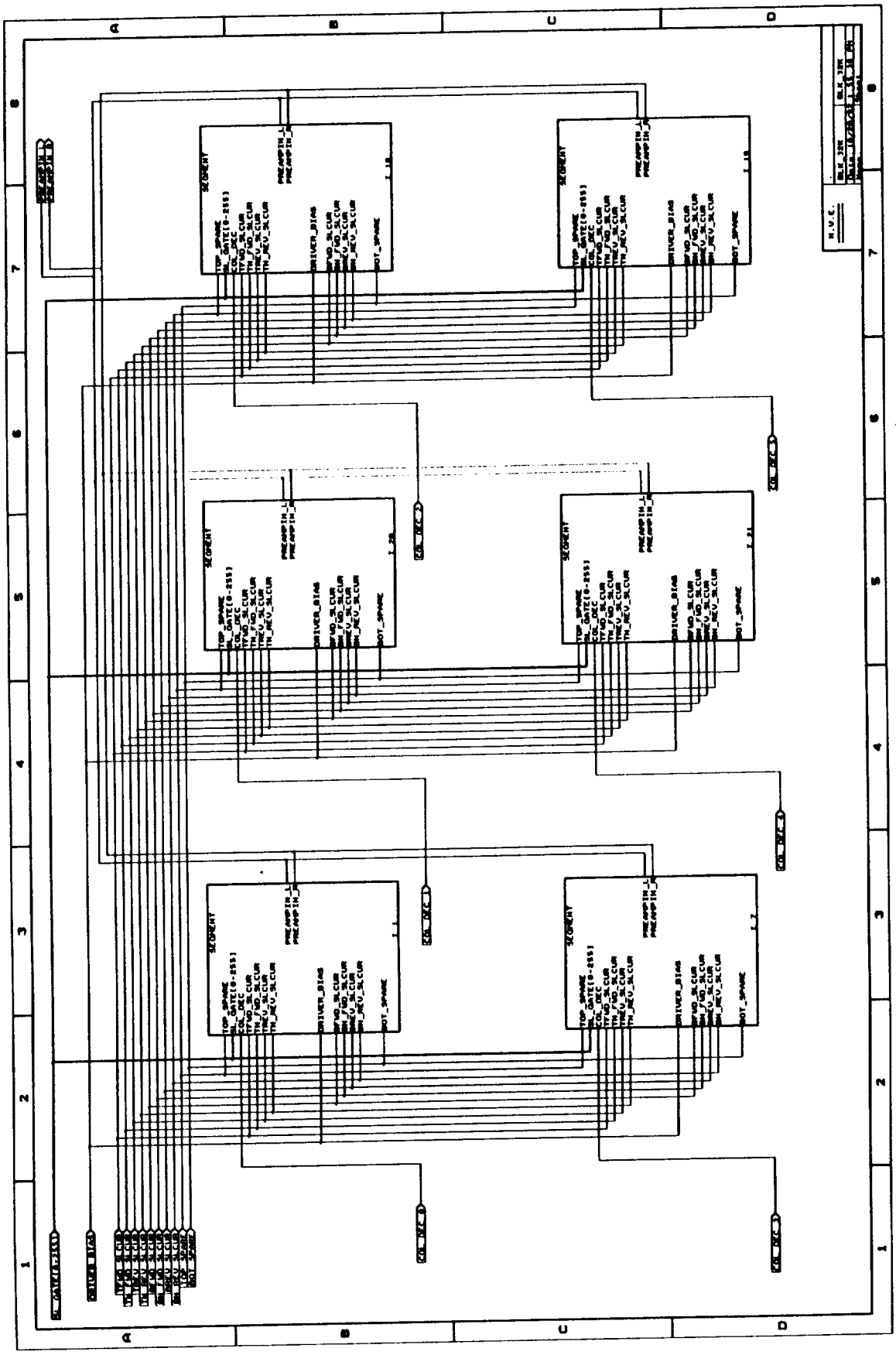


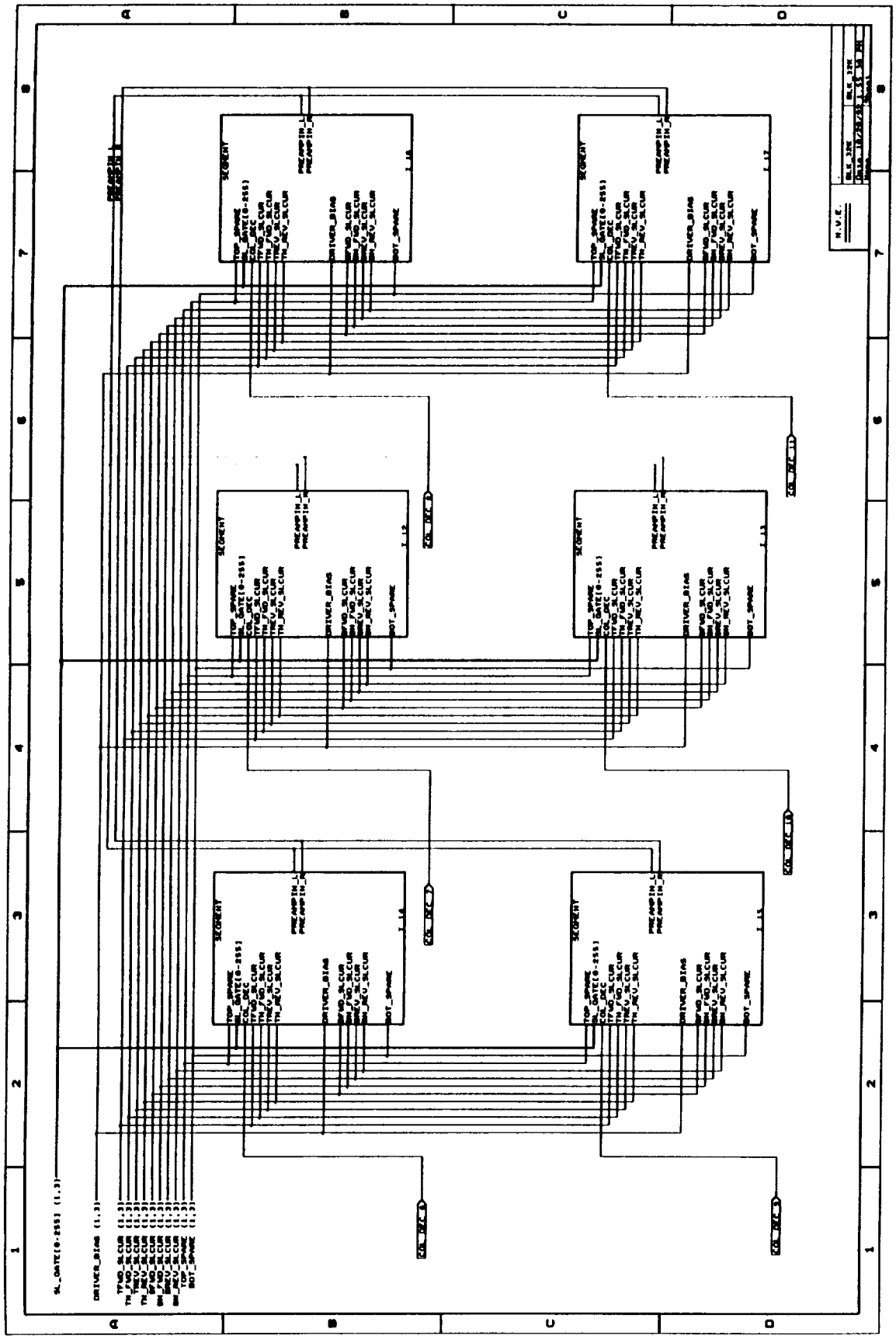

A
B
C
D
1
2
3
4
5
6
7
8
ROW_DEC5
NPRECHARGE
ENABLE_DECOD
ADD[0,17]
NADD[0,17]
ROW_R[0-255]
ROW_L[0-255]
L_TOP_SPARE
R_TOP_SPARE
L_BOT_SPARE
R_BOT_SPARE
J_26
ROW_R[0-255] (1)
ROW_L[0-255] (2)
L_TOP_SPARE (2)
R_TOP_SPARE (1)
L_BOT_SPARE (2)
R_BOT_SPARE (1)
NPRECHARGE
ENABLE_DECOD
ADD[0-17]
NADD[0-17]
N.V.E.
R128KBLK
R128KBLK
Date: 10/20/92 1:56:32 PM
Name:
Sheet

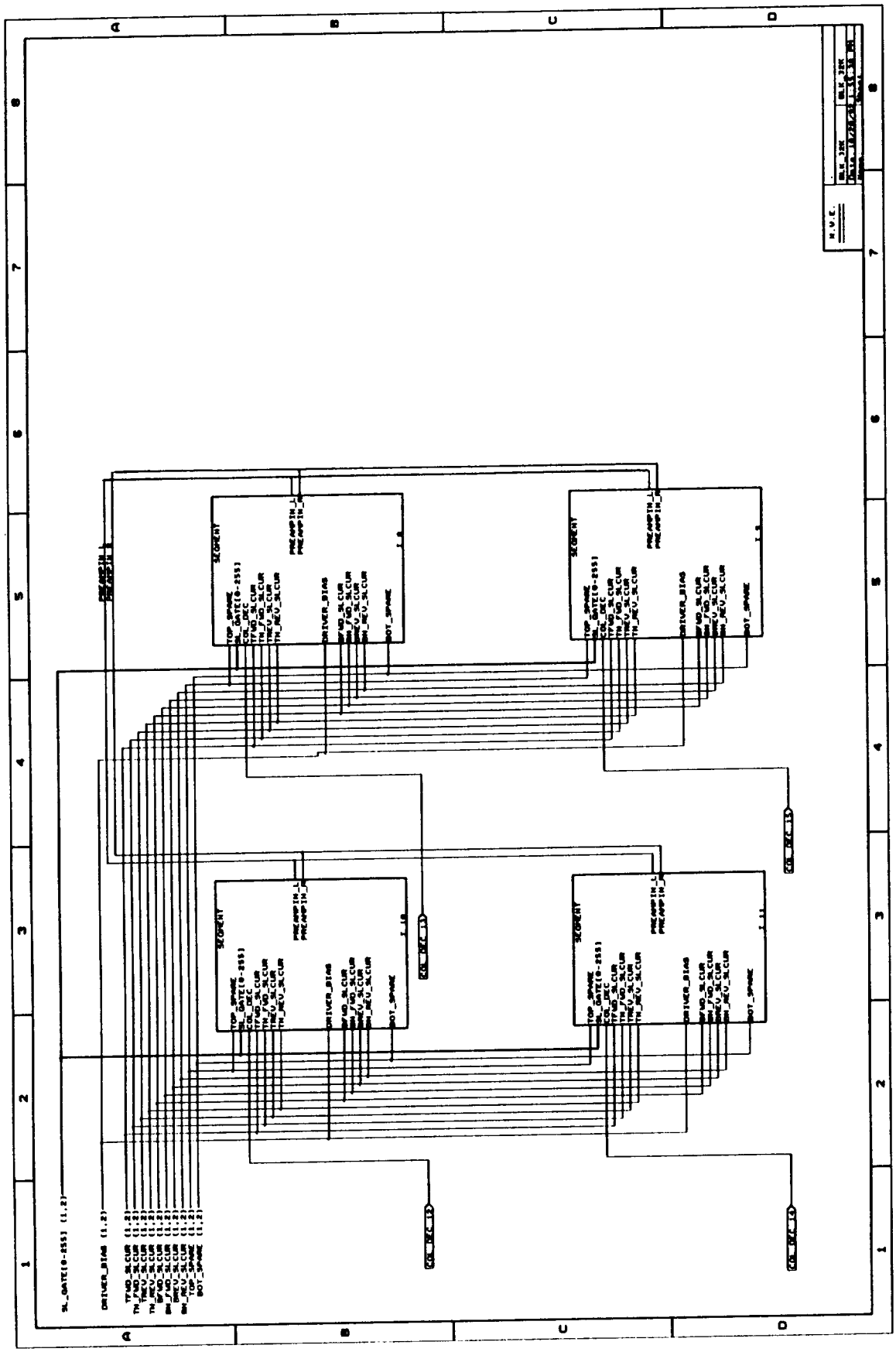


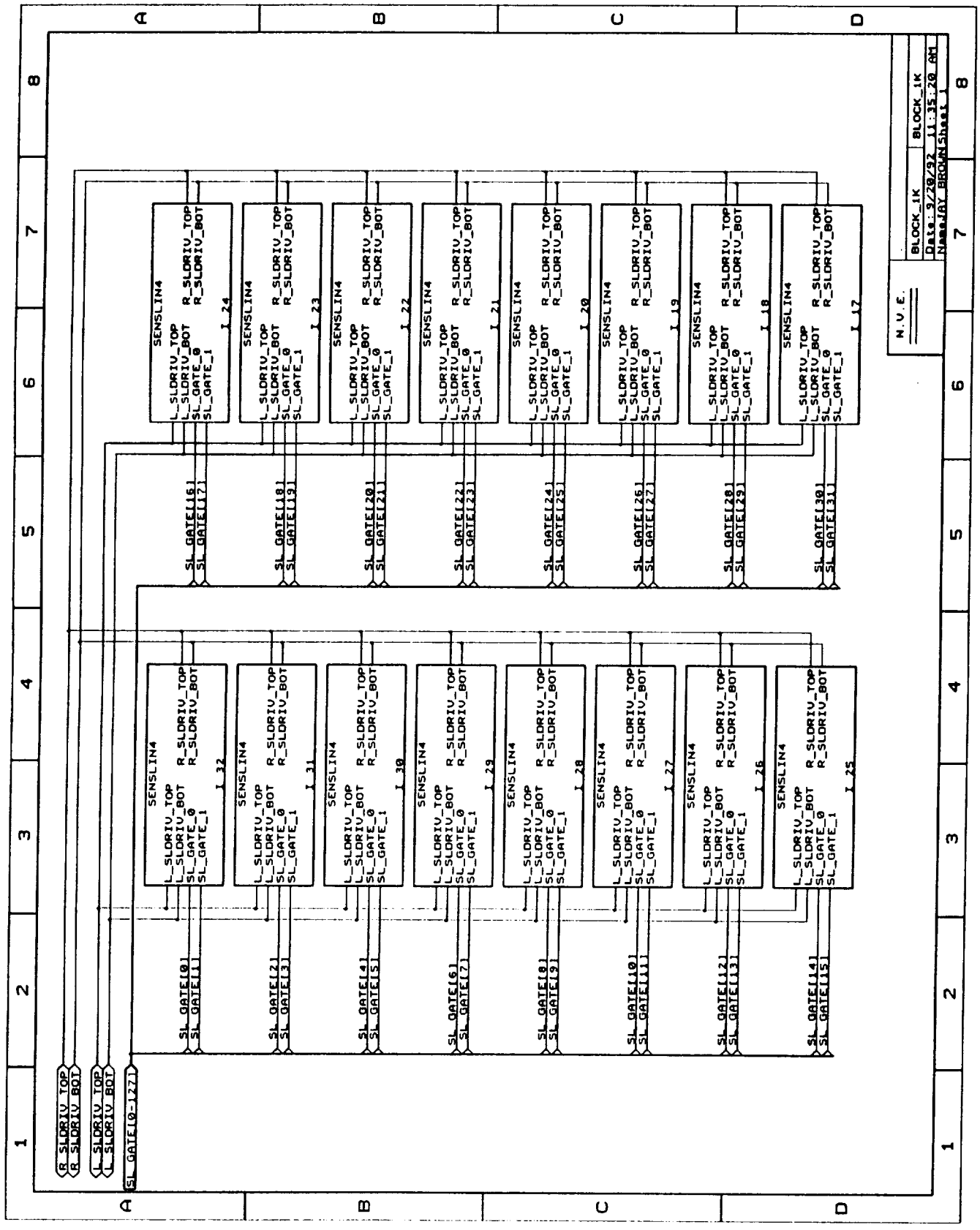


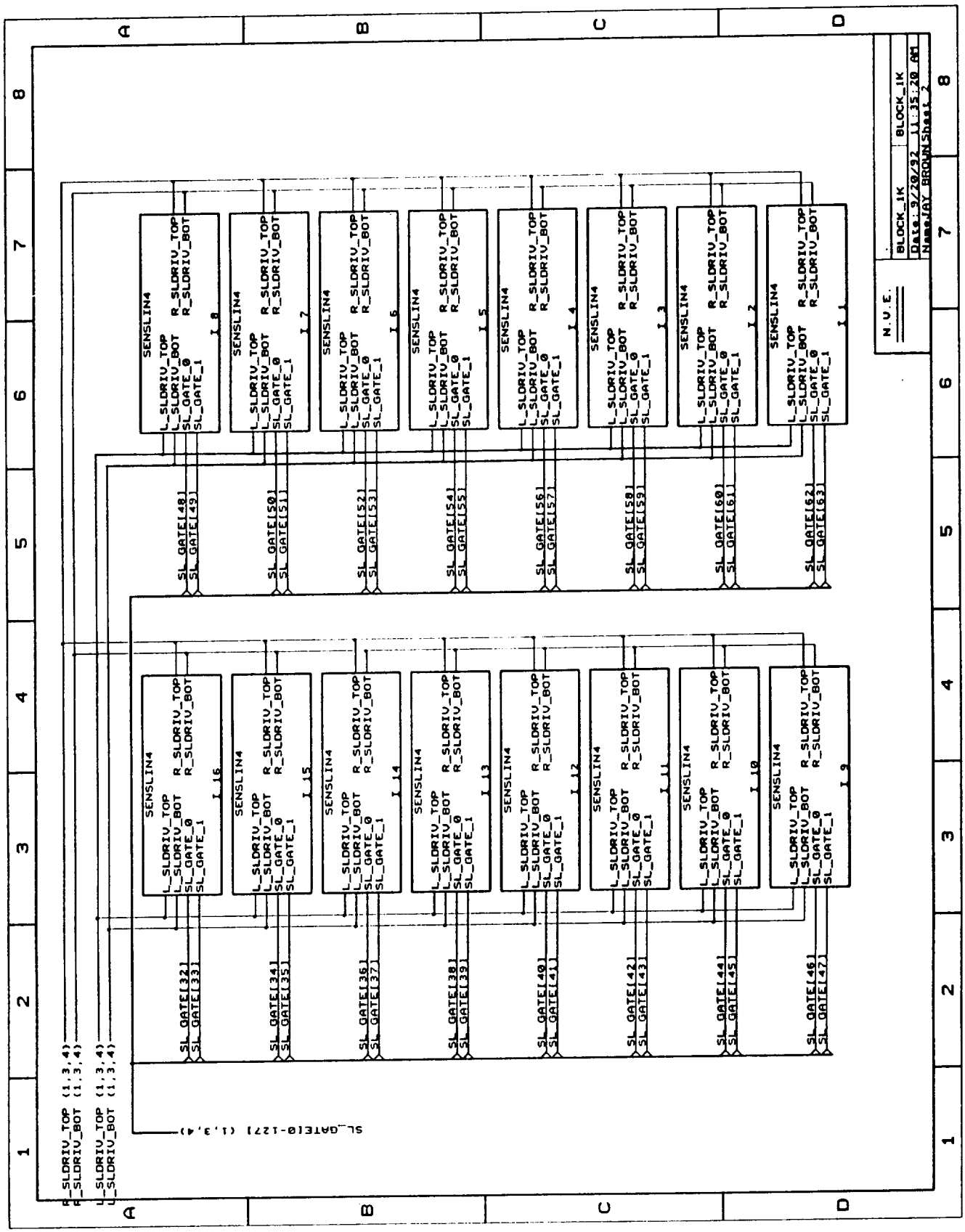

ROW_DEC5
NPRECHARGE
ENABLE_DECOD
ADD[0,17]
NADD[0,17]
ROW_R[0-255]
ROW_L[0-255]
L_TOP_SPARE
R_TOP_SPARE
L_BOT_SPARE
R_BOT_SPARE
I_26
ROW_R[0-255] (1)
ROW_L[0-255] (2)
L_TOP_SPARE (2)
R_TOP_SPARE (1)
L_BOT_SPARE (2)
R_BOT_SPARE (1)
NPRECHARGE
ENABLE_DECOD
ADD[0-17]
NADD[0-17]
N.V.E.
L128KBLK
L128KBLK
Date: 10/20/92 1:56:12 PM
Name:
Sheet

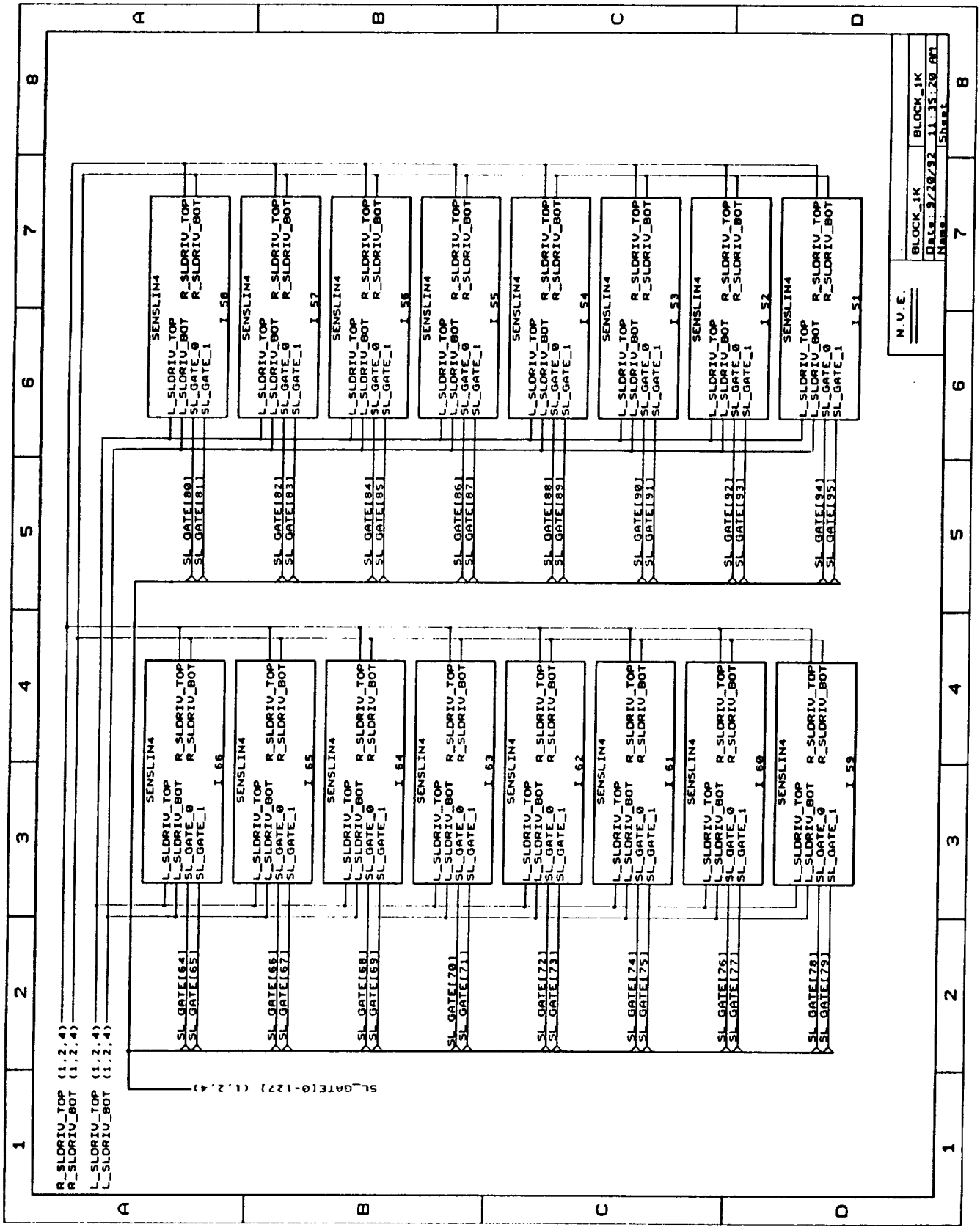


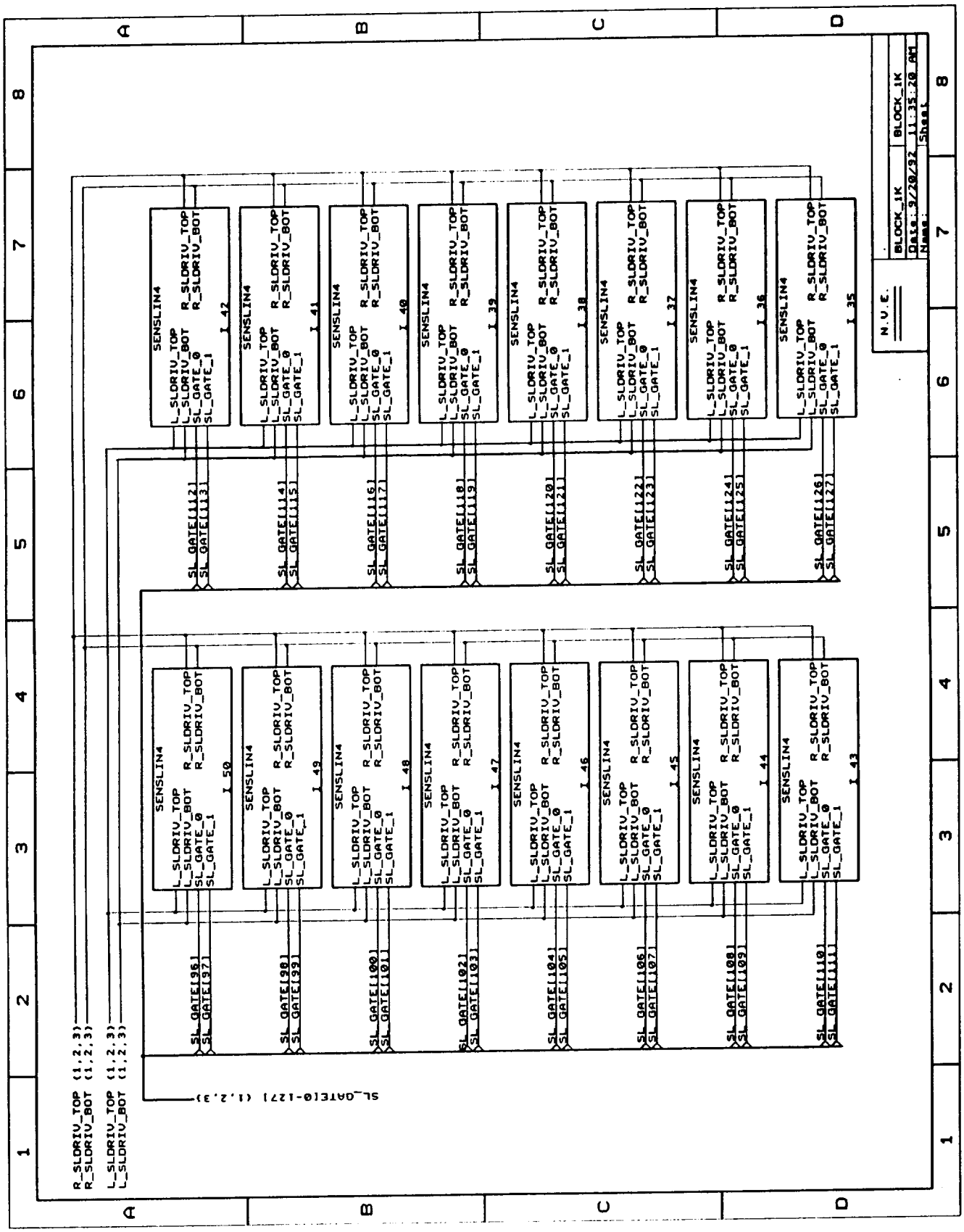


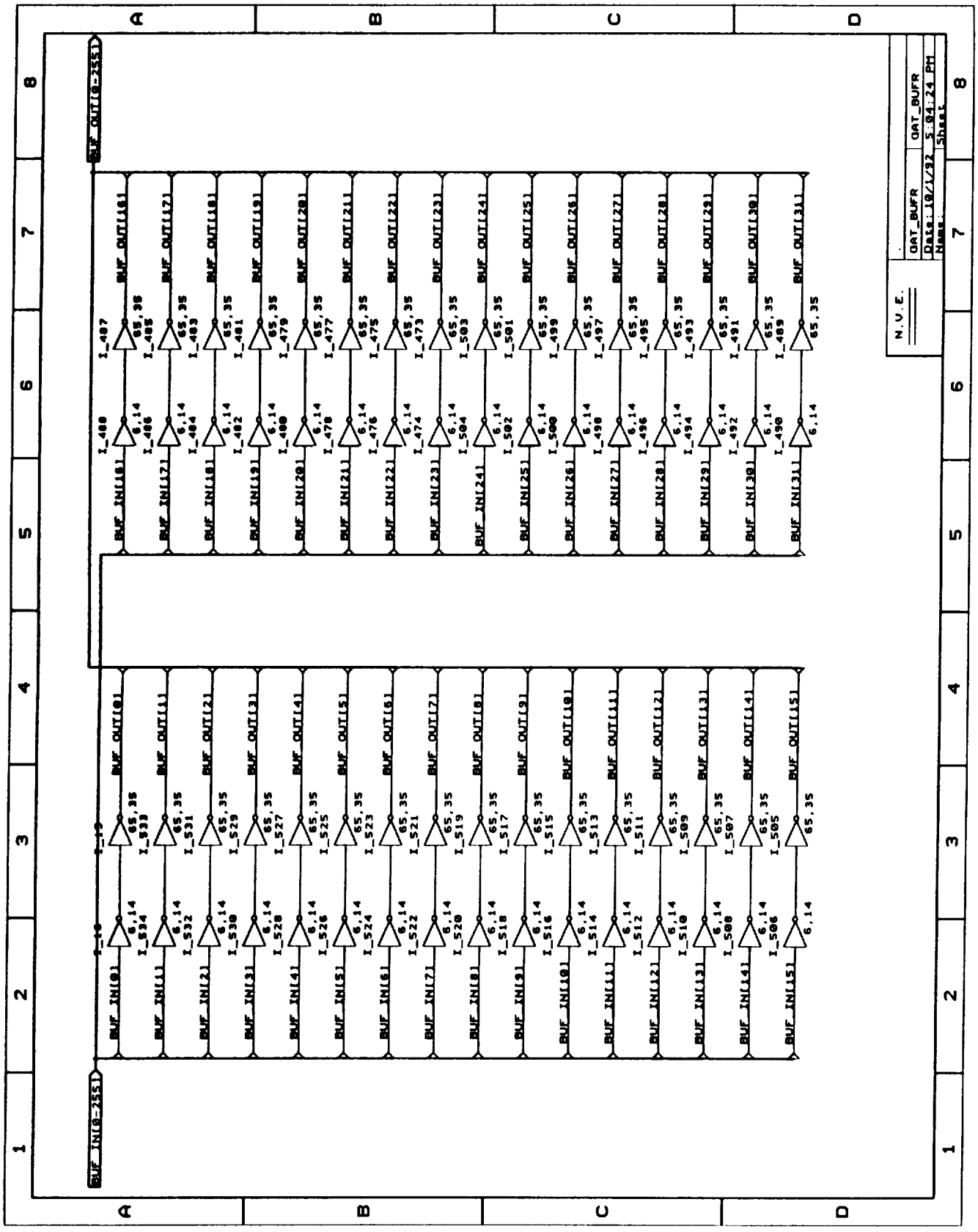


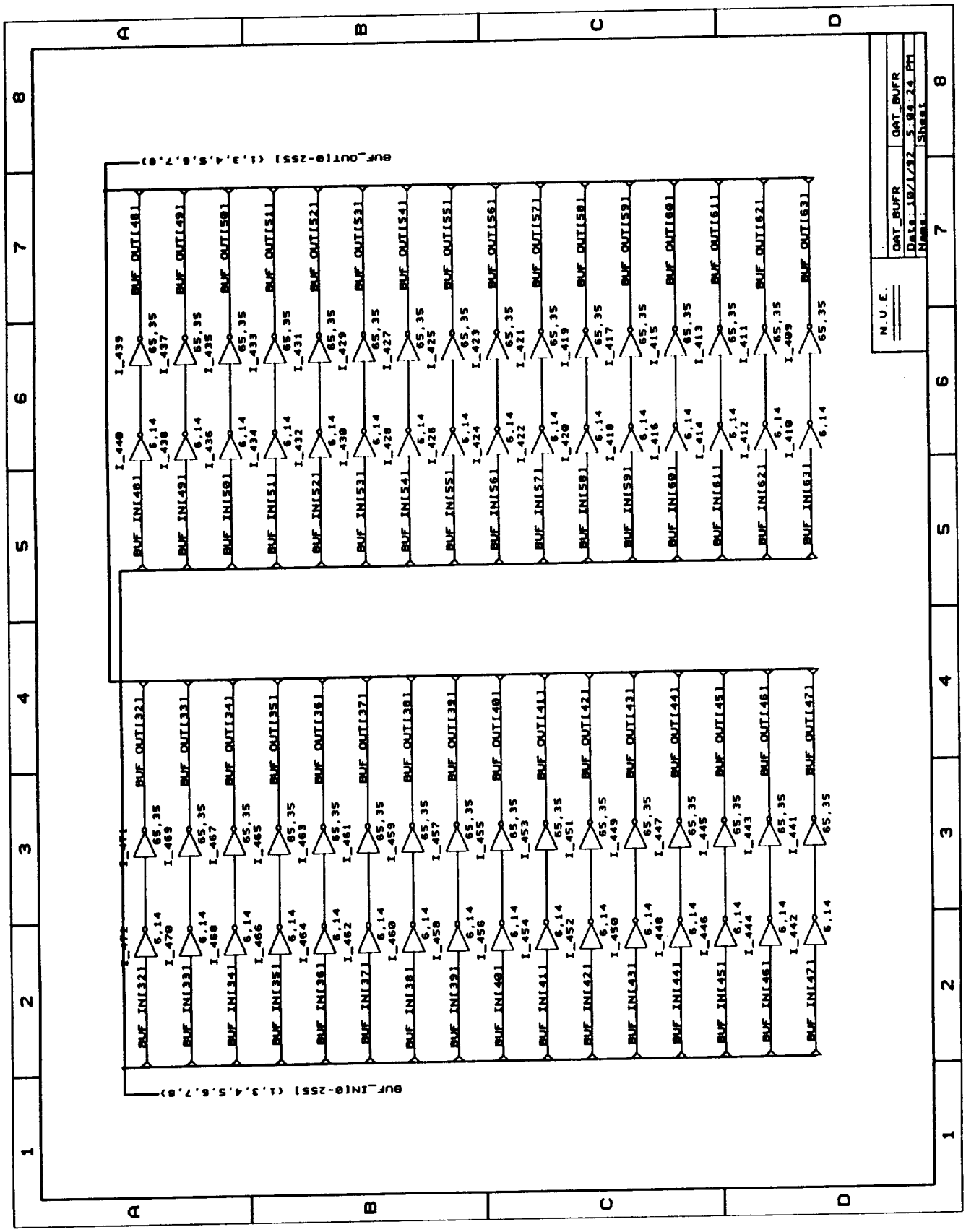

R_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_TOP
L_SLDRIV_BOT
SL_GATE[0-127]
SENSLIN4
L_SLDRIV_TOP
L_SLDRIV_BOT
SL_GATE_0
SL_GATE_1
R_SLDRIV_TOP
R_SLDRIV_BOT
I_24
I_23
I_22
I_21
I_20
I_19
I_18
I_17
I_32
I_31
I_30
I_29
I_28
I_27
I_26
I_25
SL_GATE[16]
SL_GATE[17]
SL_GATE[18]
SL_GATE[19]
SL_GATE[20]
SL_GATE[21]
SL_GATE[22]
SL_GATE[23]
SL_GATE[24]
SL_GATE[25]
SL_GATE[26]
SL_GATE[27]
SL_GATE[28]
SL_GATE[29]
SL_GATE[30]
SL_GATE[31]
SL_GATE[0]
SL_GATE[1]
SL_GATE[2]
SL_GATE[3]
SL_GATE[4]
SL_GATE[5]
SL_GATE[6]
SL_GATE[7]
SL_GATE[8]
SL_GATE[9]
SL_GATE[10]
SL_GATE[11]
SL_GATE[12]
SL_GATE[13]
SL_GATE[14]
SL_GATE[15]
N.V.E.
BLOCK_1K
BLOCK_1K
Date:3/20/92 11:35:20 AM
Name:JAY BROWNSheet 1

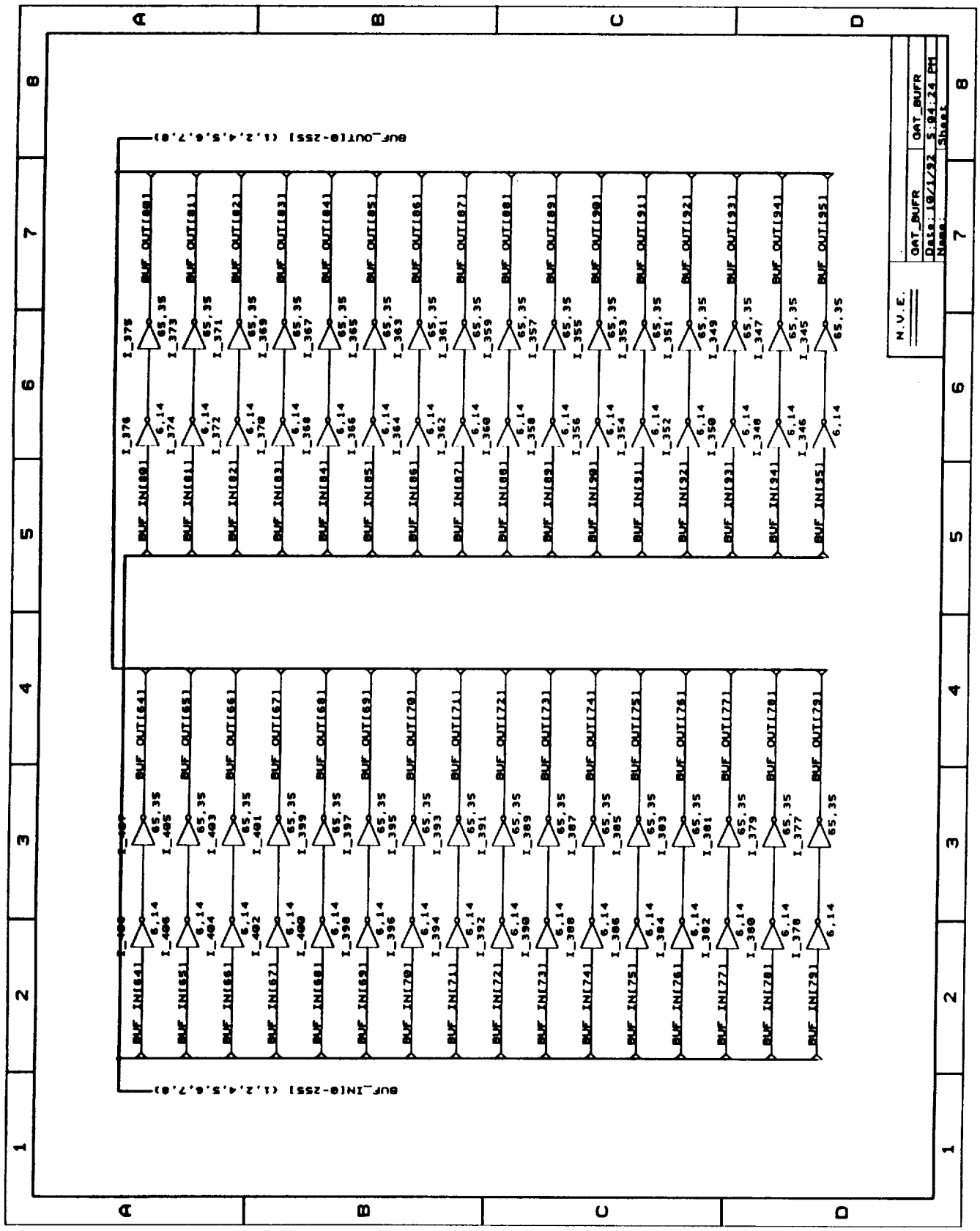

R_SLDRIV_TOP (1,3,4)
R_SLDRIV_BOT (1,3,4)
L_SLDRIV_TOP (1,3,4)
L_SLDRIV_BOT (1,3,4)
SL_GATE0-127 (1,3,4)
SENSLIN4
L_SLDRIV_TOP
L_SLDRIV_BOT
SL_GATE_0
SL_GATE_1
R_SLDRIV_TOP
R_SLDRIV_BOT
I_16
I_15
I_14
I_13
I_12
I_11
I_10
I_9
I_8
I_7
I_6
I_5
I_4
I_3
I_2
I_1
SL_GATE[32]
SL_GATE[33]
SL_GATE[34]
SL_GATE[35]
SL_GATE[36]
SL_GATE[37]
SL_GATE[38]
SL_GATE[39]
SL_GATE[40]
SL_GATE[41]
SL_GATE[42]
SL_GATE[43]
SL_GATE[44]
SL_GATE[45]
SL_GATE[46]
SL_GATE[47]
SL_GATE[48]
SL_GATE[49]
SL_GATE[50]
SL_GATE[51]
SL_GATE[52]
SL_GATE[53]
SL_GATE[54]
SL_GATE[55]
SL_GATE[56]
SL_GATE[57]
SL_GATE[58]
SL_GATE[59]
SL_GATE[60]
SL_GATE[61]
SL_GATE[62]
SL_GATE[63]
N.U.E.
BLOCK_1K
BLOCK_1K
Date:9/29/92 11:35:20 AM
Name:JAY BROWN Sheet 2

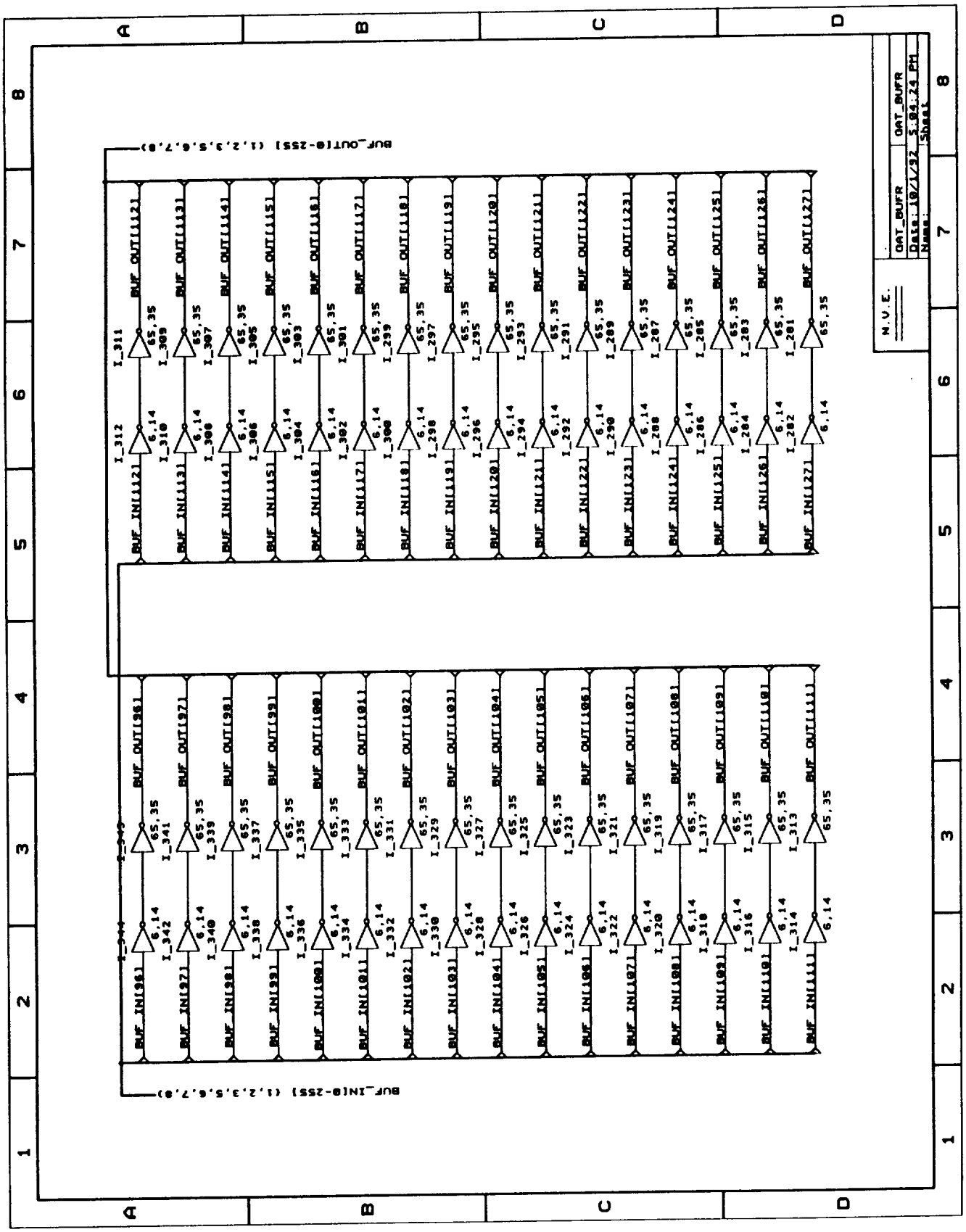

N.V.E.
BLOCK_1K
BLOCK_1K
Date: 9/20/92 11:35:20 AM
Sheet
SENSLIN4
L_SLDRIV_TOP
L_SLDRIV_BOT
SL_GATE_0
SL_GATE_1
R_SLDRIV_TOP
R_SLDRIV_BOT
I_58
I_57
I_56
I_55
I_54
I_53
I_52
I_51
I_66
I_65
I_64
I_63
I_62
I_61
I_60
I_59
SL_GATE[80]
SL_GATE[81]
SL_GATE[82]
SL_GATE[83]
SL_GATE[84]
SL_GATE[85]
SL_GATE[86]
SL_GATE[87]
SL_GATE[88]
SL_GATE[89]
SL_GATE[90]
SL_GATE[91]
SL_GATE[92]
SL_GATE[93]
SL_GATE[94]
SL_GATE[95]
SL_GATE[64]
SL_GATE[65]
SL_GATE[66]
SL_GATE[67]
SL_GATE[68]
SL_GATE[69]
SL_GATE[70]
SL_GATE[71]
SL_GATE[72]
SL_GATE[73]
SL_GATE[74]
SL_GATE[75]
SL_GATE[76]
SL_GATE[77]
SL_GATE[78]
SL_GATE[79]
SL_GATE[0-127] (1,2,4)
R_SLDRIV_TOP (1,2,4)
R_SLDRIV_BOT (1,2,4)
L_SLDRIV_TOP (1,2,4)
L_SLDRIV_BOT (1,2,4)

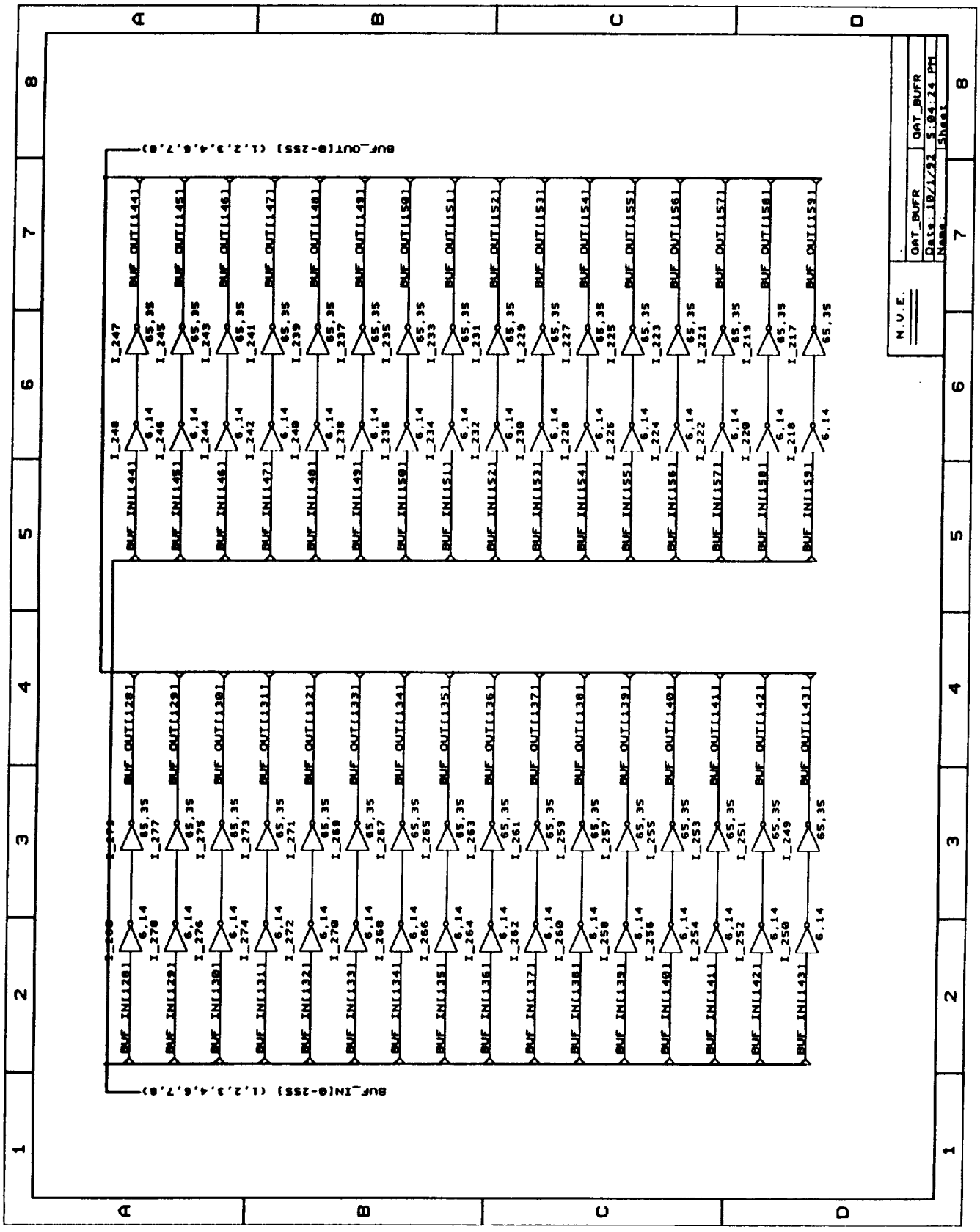

R_SLDRIV_TOP (1,2,3)
R_SLDRIV_BOT (1,2,3)
L_SLDRIV_TOP (1,2,3)
L_SLDRIV_BOT (1,2,3)
SL_GATE[0-127] (1,2,3)
SENSLIN4
L_SLDRIV_TOP
L_SLDRIV_BOT
SL_GATE_0
SL_GATE_1
R_SLDRIV_TOP
R_SLDRIV_BOT
I_50
I_49
I_48
I_47
I_46
I_45
I_44
I_43
I_42
I_41
I_40
I_39
I_38
I_37
I_36
I_35
N.V.E.
BLOCK_1K
BLOCK_1K
Date: 9/20/92
11:35:20 AM
Sheet

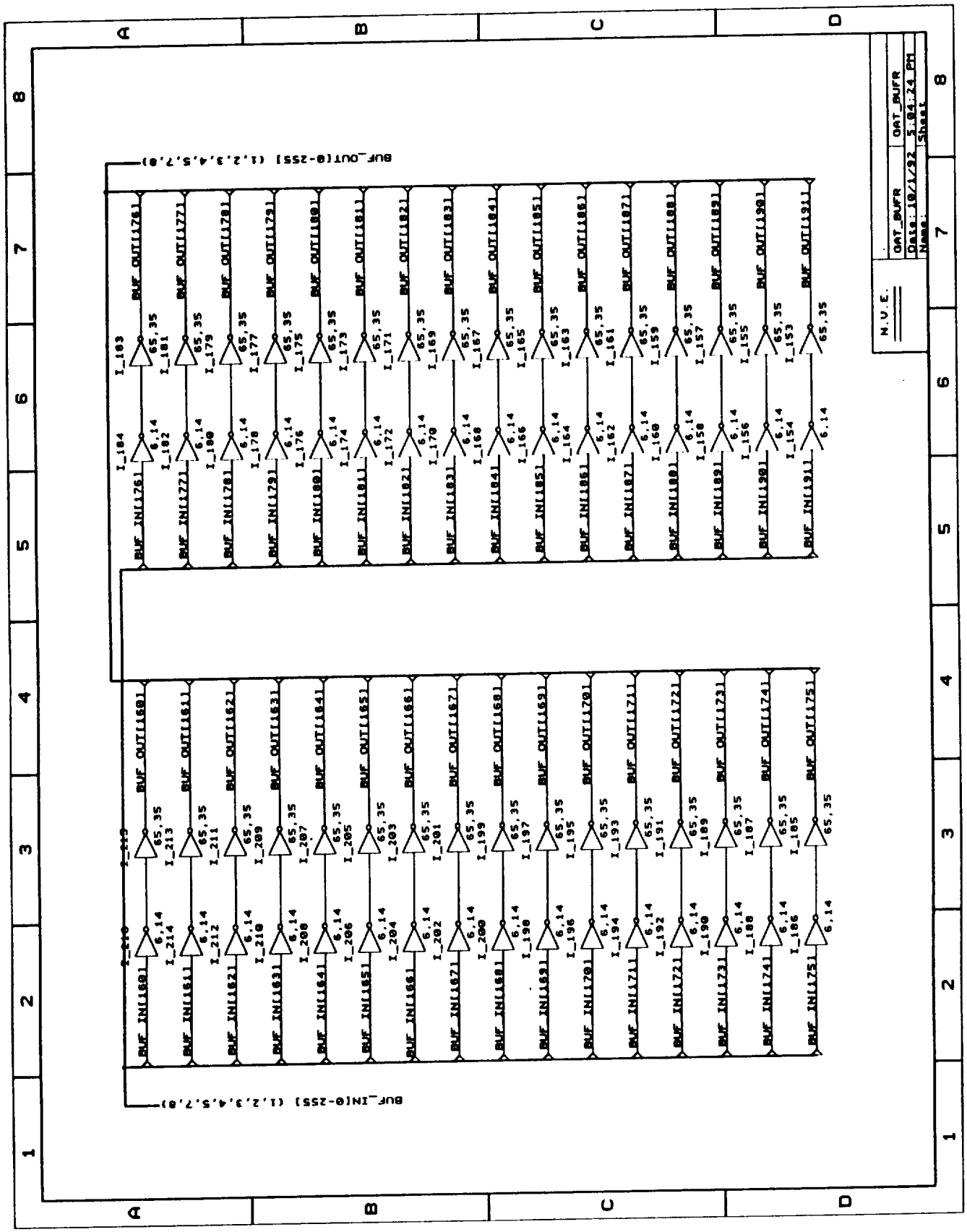

BUF_IN[0-255]
BUF_OUT[0-255]
N.V.E.
DAT_BUFR
Date: 10/1/92 5:04:24 PM
Name:
Sheet
DAT_BUFR

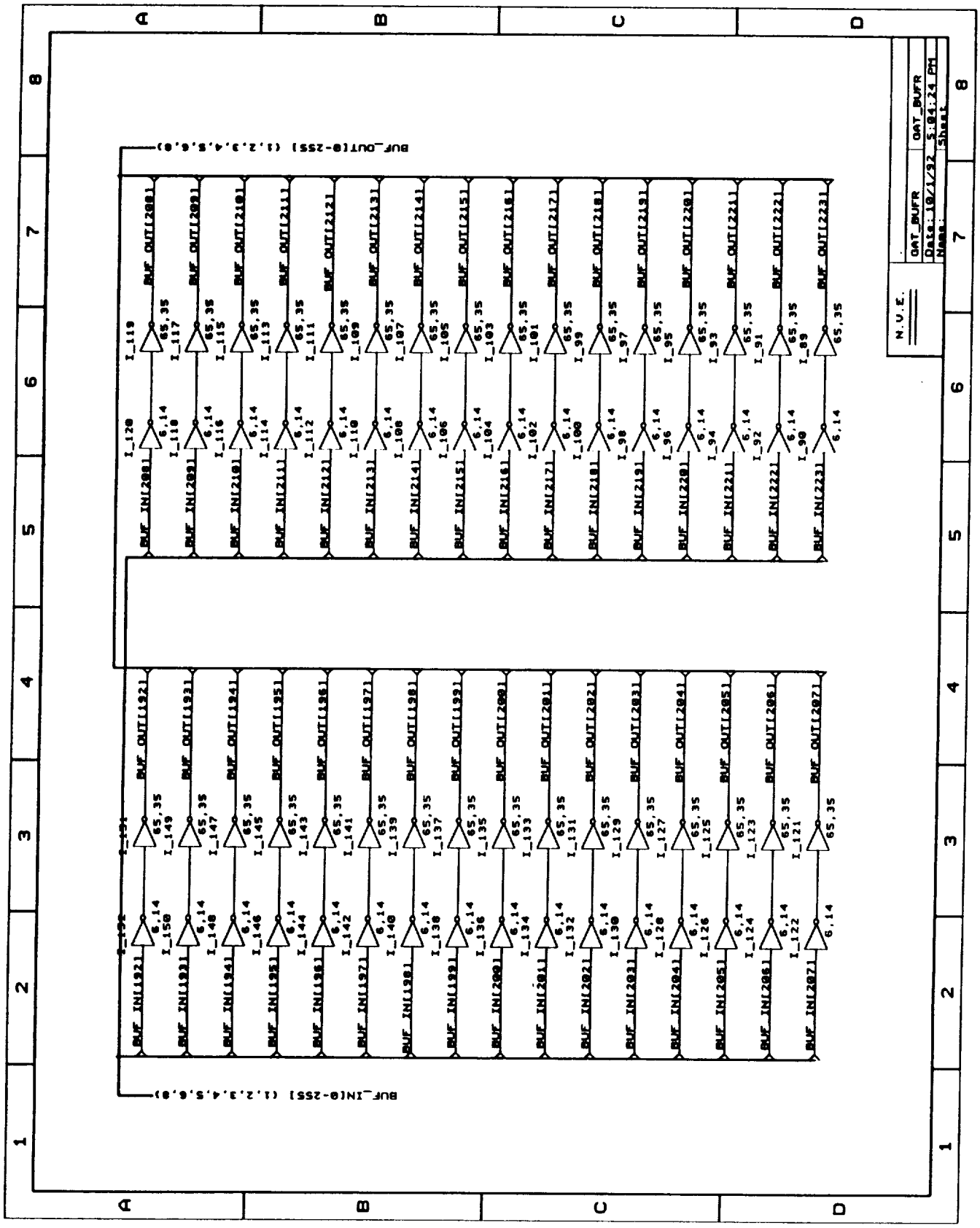

N.V.E.
DAT_BUFR
DAT_BUFR
Date: 10/1/92 5:04:24 PM
Name:
Sheet
BUF_OUT[0-255] (1,3,4,5,6,7,8)
BUF_IN[0-255] (1,3,4,5,6,7,8)

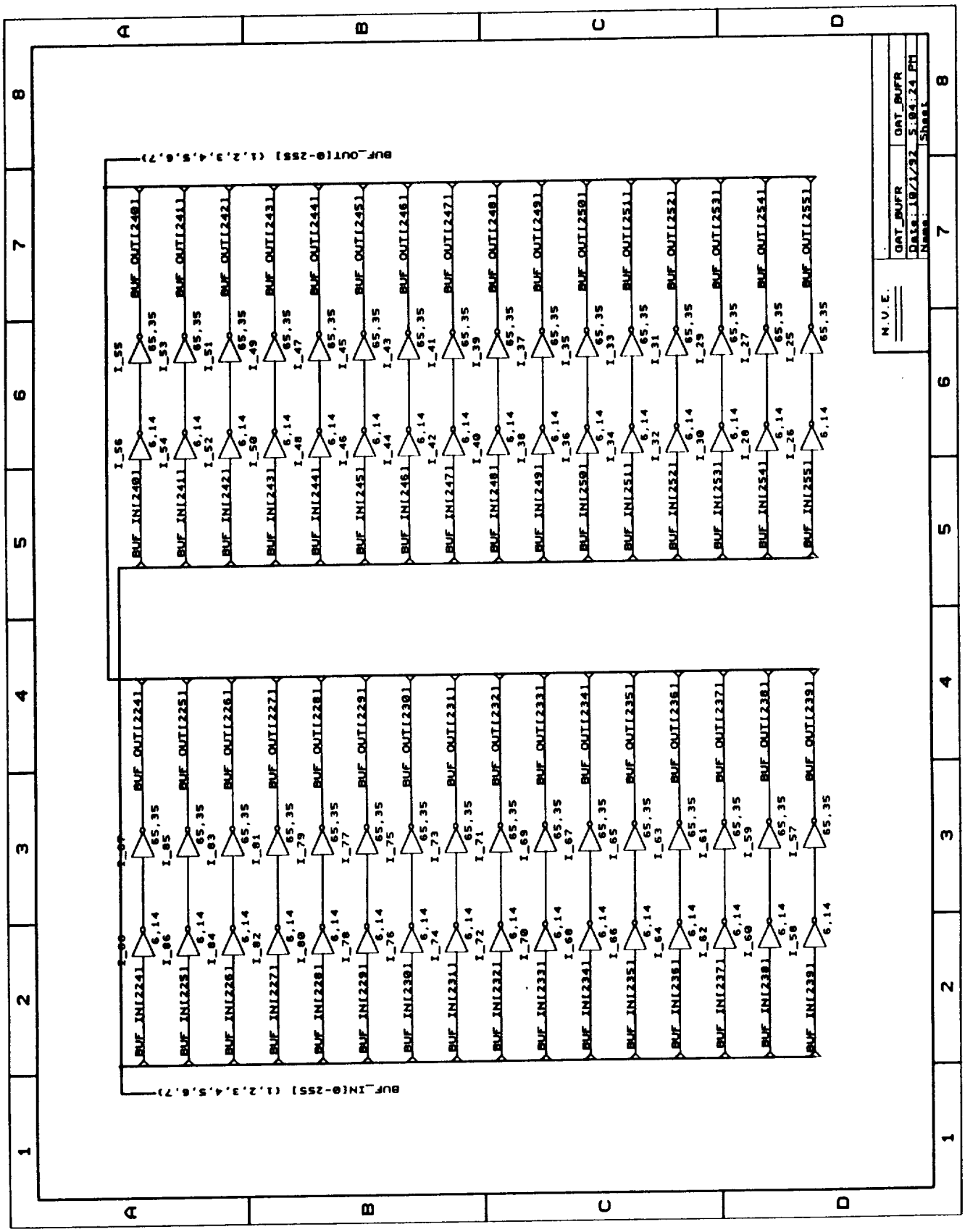

N.V.E.
OAT_BUFR
OAT_BUFR
Date: 10/1/92 5:04:24 PM
Sheet
BUF_OUT[0-255] (1,2,4,5,6,7,8)
BUF_IN[0-255] (1,2,4,5,6,7,8)

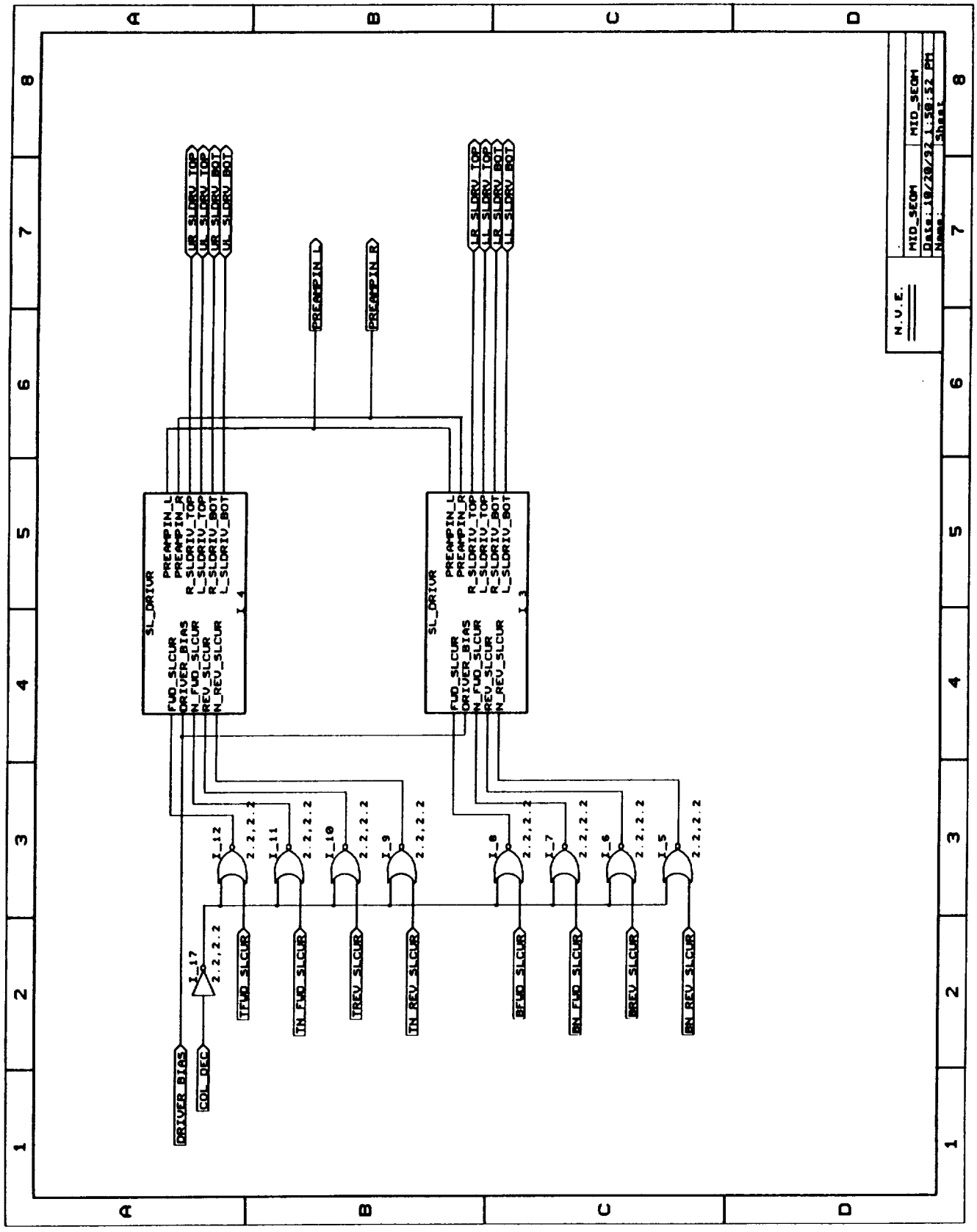

A
B
C
D
1
2
3
4
5
6
7
8
N.V.E.
DAT_BUFR
Date: 10/1/92 5:04:24 PM
Sheet
BUF_OUT[0-255] (1,2,3,5,6,7,8)
BUF_IN[0-255] (1,2,3,5,6,7,8)

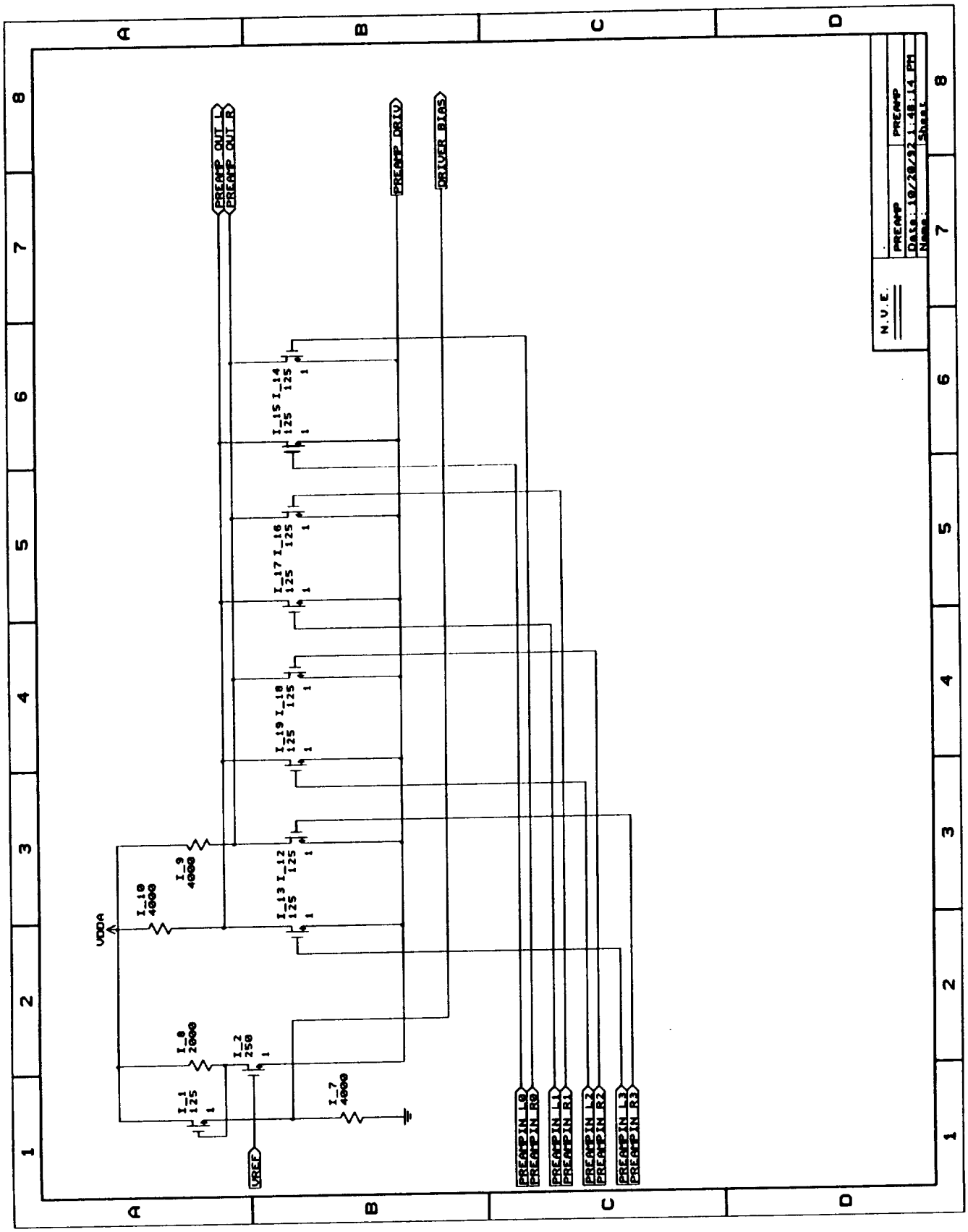

N.V.E.
GAT_BUFR
GAT_BUFR
Date: 10/1/92 5:04:24 PM
Name:
Sheet
BUF_OUT[0-255] (1,2,3,4,6,7,8)
BUF_IN[0-255] (1,2,3,4,6,7,8)

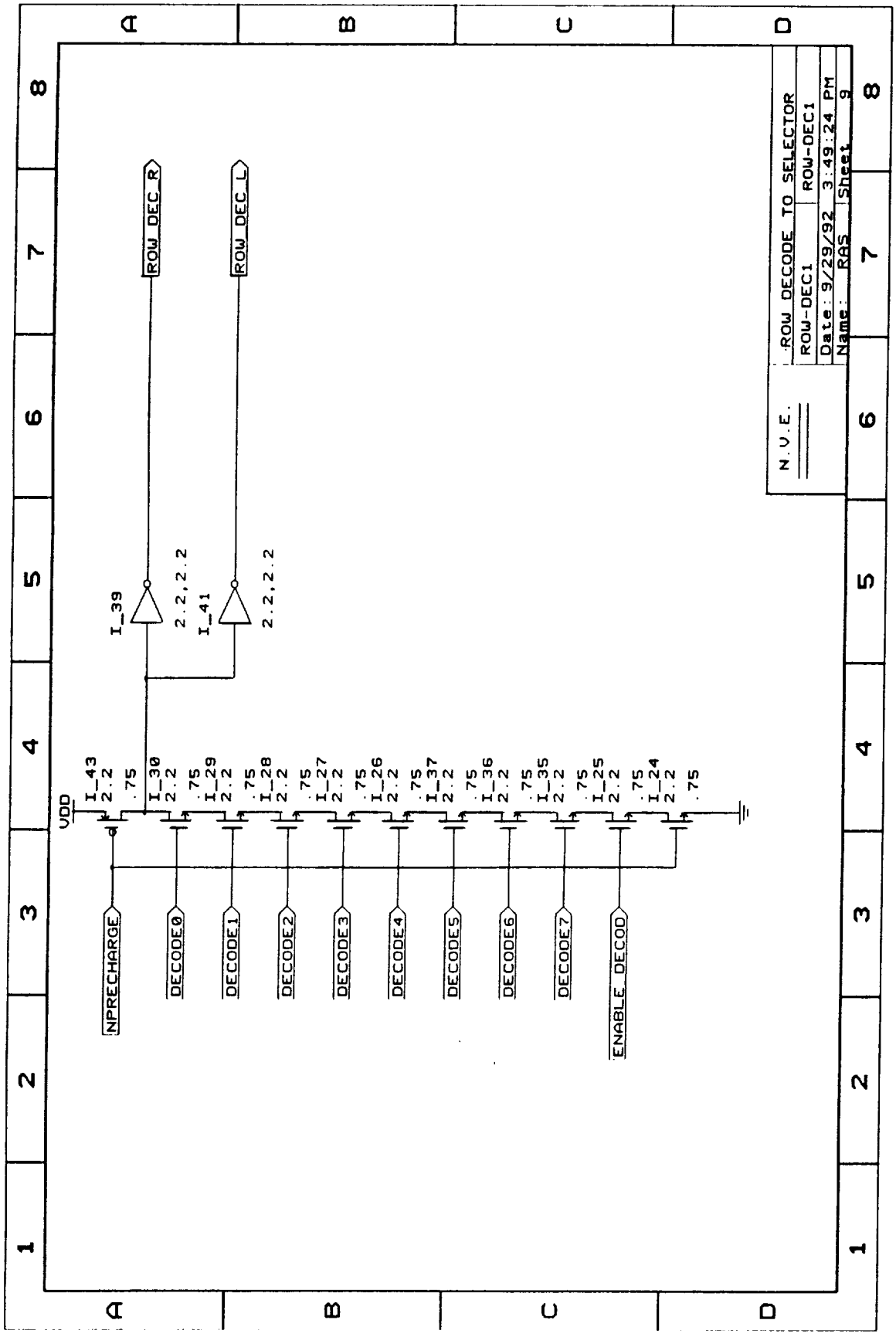

N.U.E.
DAT_BUFR
DAT_BUFR
Date: 10/1/92 5:04:24 PM
Name:
Sheet
BUF_OUT[0-255] (1,2,3,4,5,7,8)
BUF_IN[0-255] (1,2,3,4,5,7,8)

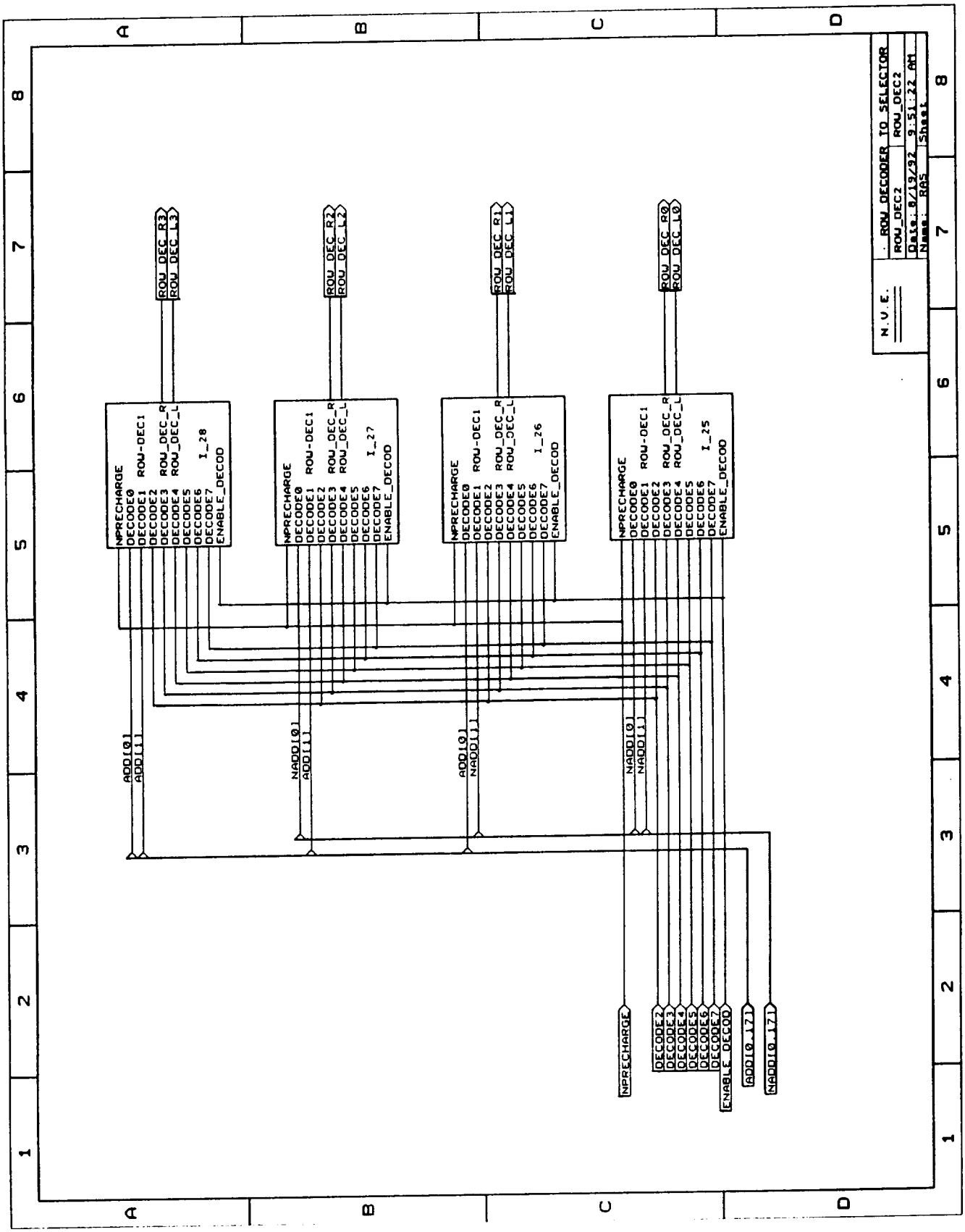

BUF_OUT[0-255] (1,2,3,4,5,6,8)
BUF_IN[0-255] (1,2,3,4,5,6,8)
N.U.E.
OAT_BUFR
Date: 10/1/92 5:04:24 PM
Sheet
A
B
C
D
1
2
3
4
5
6
7
8

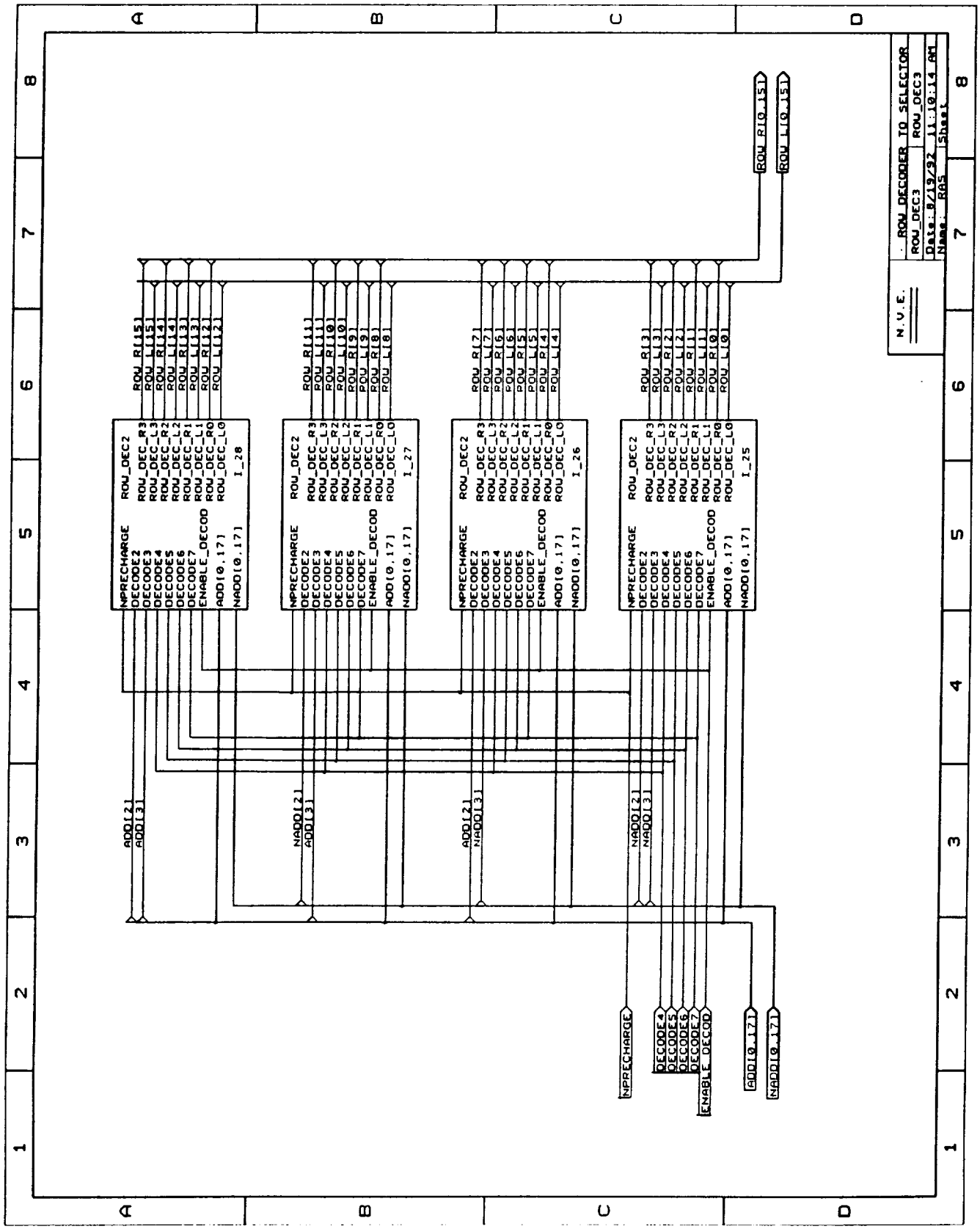

N.V.E.
GAT_BUFR
GAT_BUFR
Date: 10/21/92 5:04:24 PM
Name:
Sheet
BUF_IN[0-255] (1,2,3,4,5,6,7)
BUF_OUT[0-255] (1,2,3,4,5,6,7)

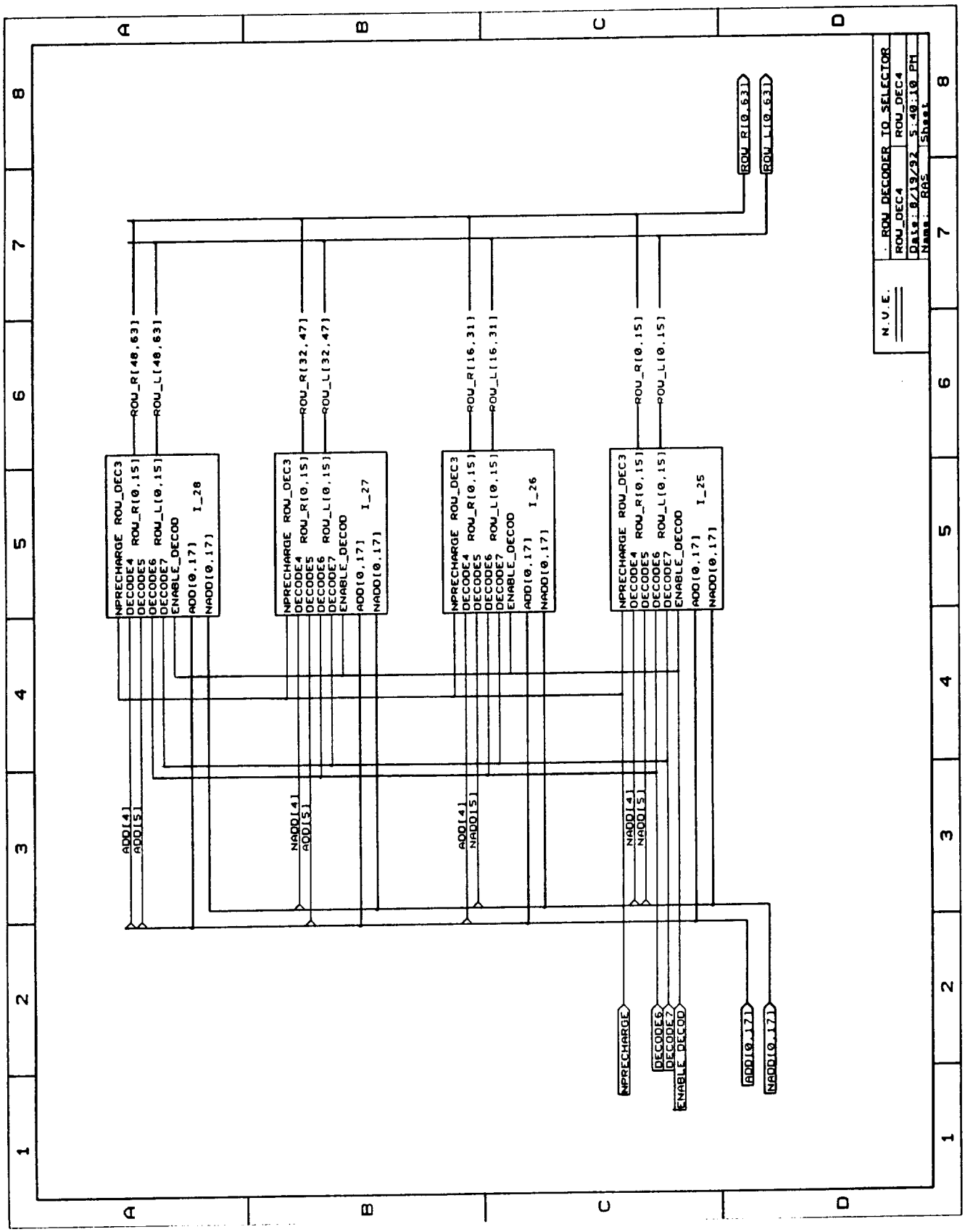

A
B
C
D
1
2
3
4
5
6
7
8
N.U.E.
MID_SEQM
MID_SEQM
Date: 10/20/92 L:58:52 PM
Name:
Sheet
UR_SLDRV_TOP
UL_SLDRV_TOP
UR_SLDRV_BOT
UL_SLDRV_BOT
PREAMPIN_L
PREAMPIN_R
LR_SLDRV_TOP
LL_SLDRV_TOP
LR_SLDRV_BOT
LL_SLDRV_BOT
SL_DRIVR
PREAMPIN_L
PREAMPIN_R
R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
FWD_SLCUR
DRIVER_BIAS
N_FWD_SLCUR
REV_SLCUR
N_REV_SLCUR
I_4
I_3
I_12
I_11
I_10
I_9
I_8
I_7
I_6
I_5
I_17
2.2,2.2
TFWD_SLCUR
TN_FWD_SLCUR
TREV_SLCUR
TN_REV_SLCUR
BFWD_SLCUR
BN_FWD_SLCUR
BREV_SLCUR
BN_REV_SLCUR
DRIVER_BIAS
COIL_DEC

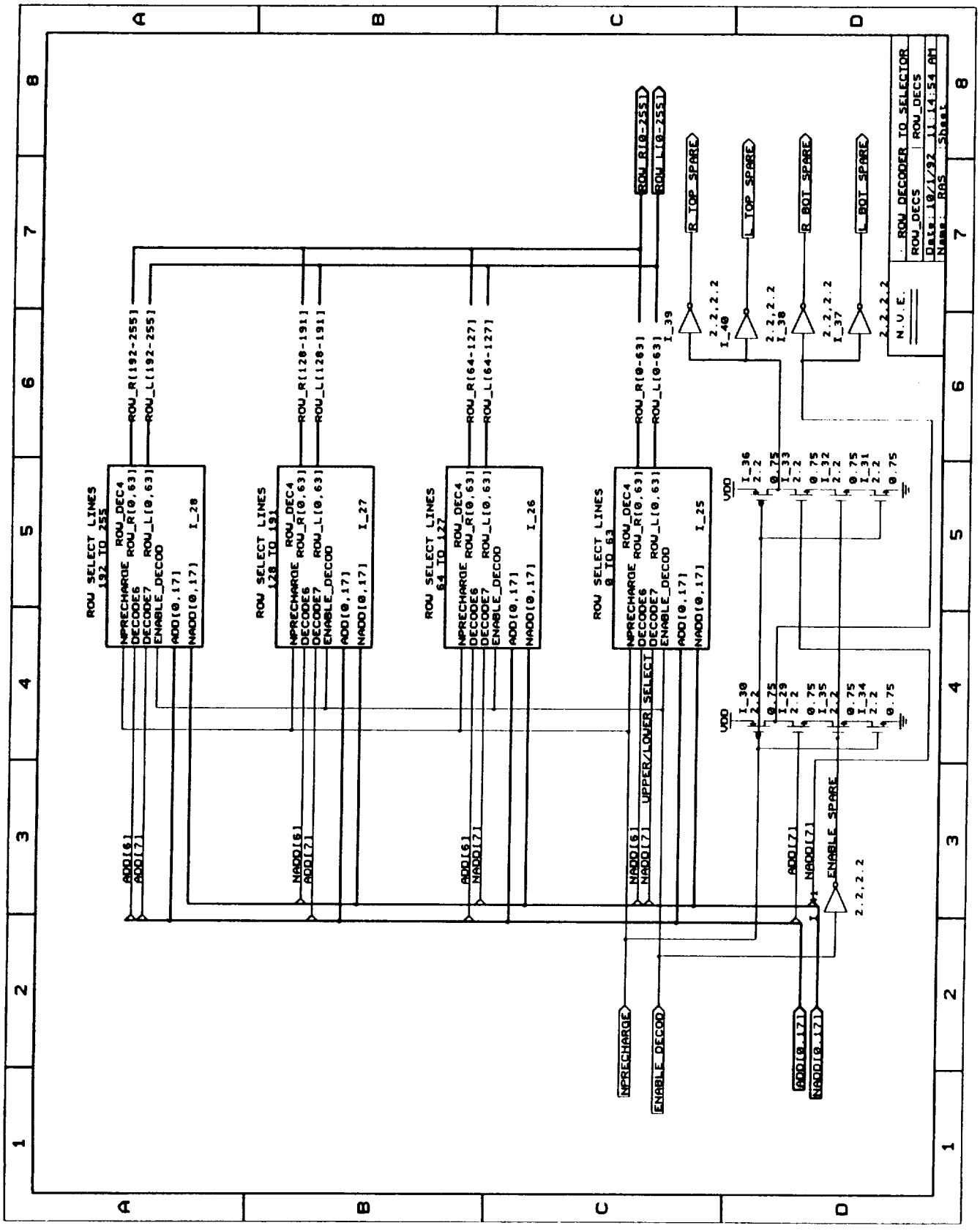

PREAMP OUT L
PREAMP OUT R
PREAMP DRIV
DRIVER BIAS
VDDA
VREF
PREAMP IN L0
PREAMP IN R0
PREAMP IN L1
PREAMP IN R1
PREAMP IN L2
PREAMP IN R2
PREAMP IN L3
PREAMP IN R3
N.V.E.
PREAMP
PREAMP
Date: 10/28/92 1:48:14 PM
Sheet
Name:

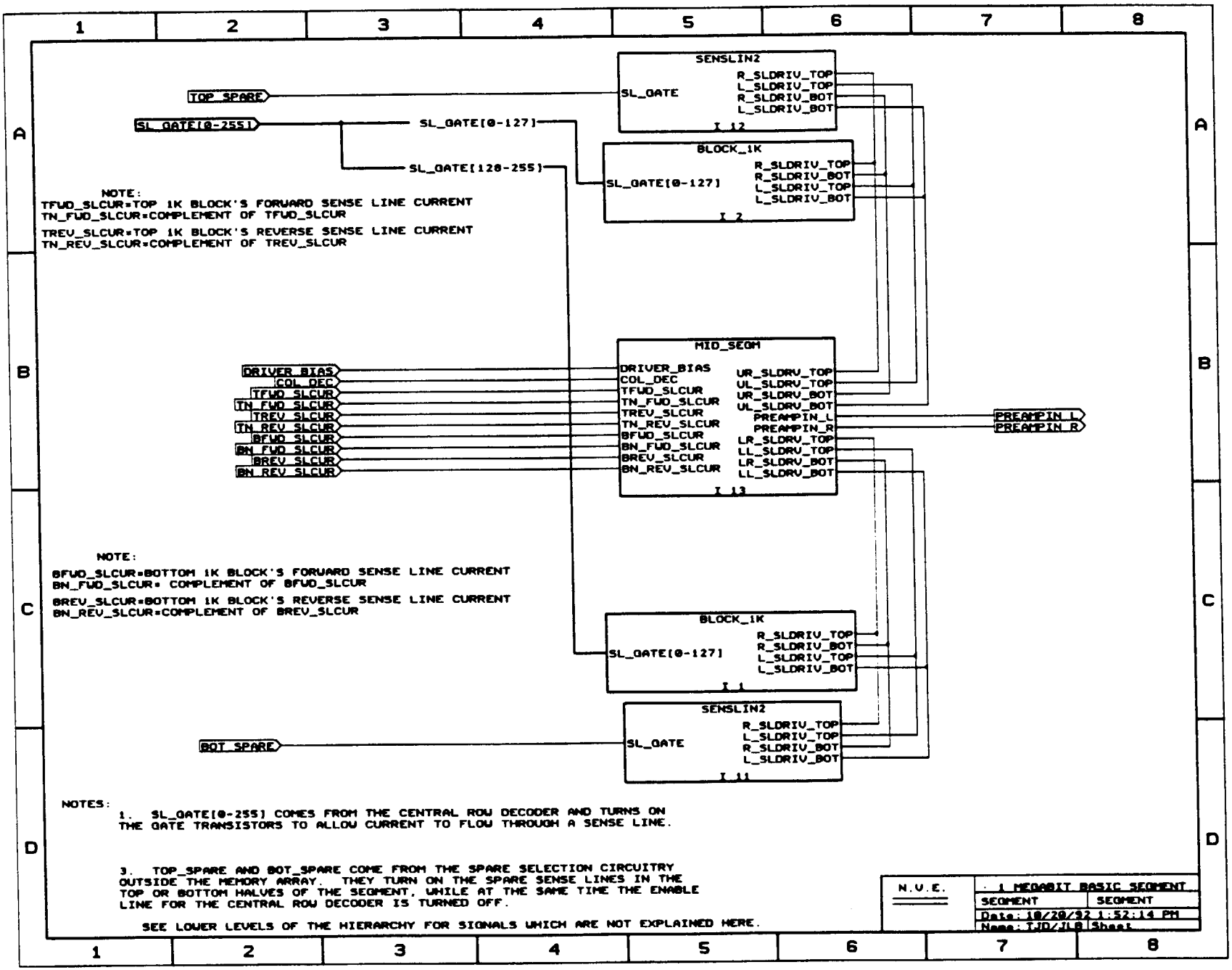

ROW DECODE TO SELECTOR
N.V.E.
ROW-DEC1
ROW-DEC1
Date: 9/29/92 3:49:24 PM
Name: RAS Sheet 9
ROW_DEC_R
ROW_DEC_L
I_39
2.2,2.2.2
I_41
2.2,2.2.2
VDD
I_43 2.2 .75
I_30 2.2 .75
I_29 2.2 .75
I_28 2.2 .75
I_27 2.2 .75
I_26 2.2 .75
I_37 2.2 .75
I_36 2.2 .75
I_35 2.2 .75
I_25 2.2 .75
I_24 2.2 .75
NPRECHARGE
DECODE0
DECODE1
DECODE2
DECODE3
DECODE4
DECODE5
DECODE6
DECODE7
ENABLE_DECOD

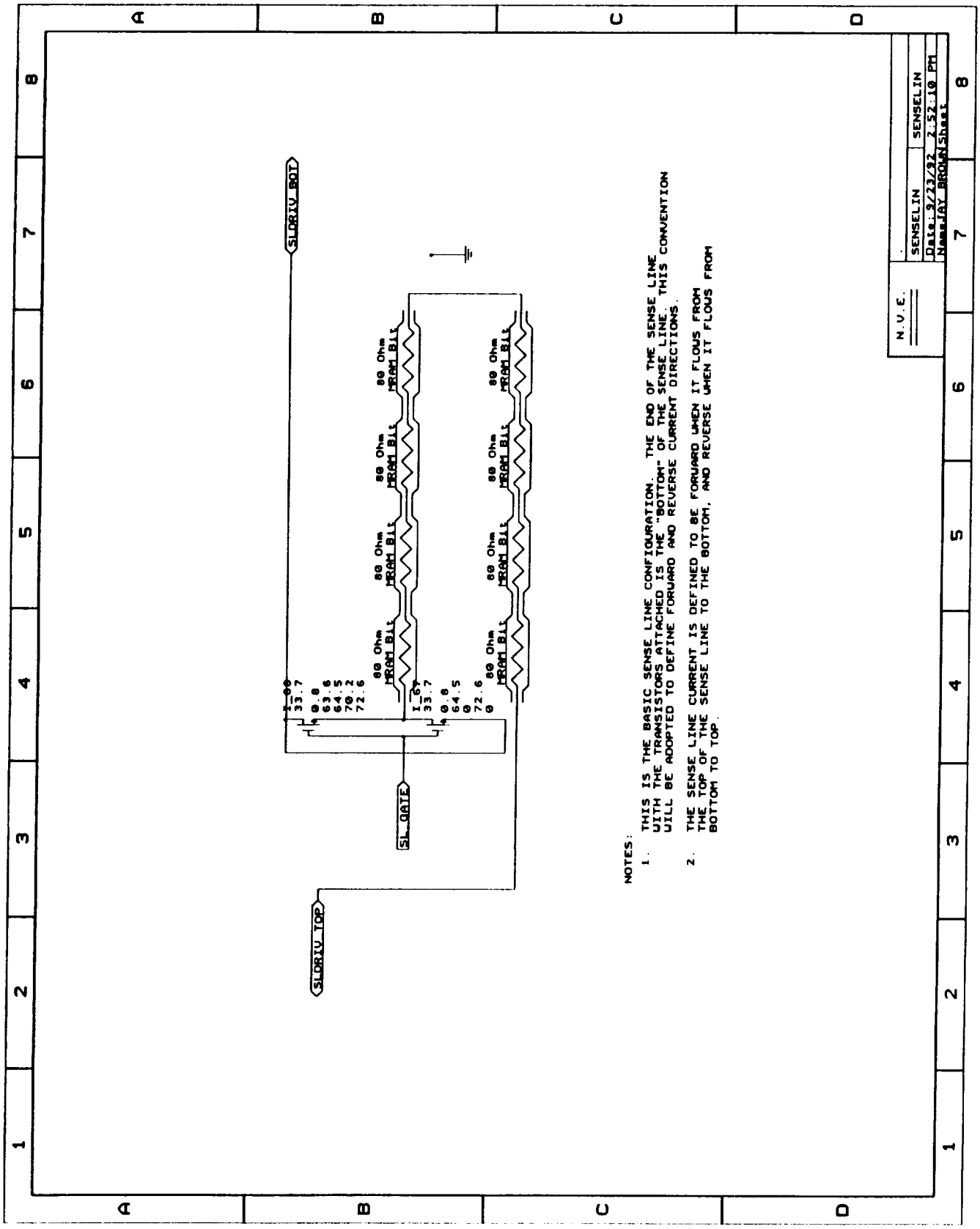

ROW_DECODER_TO_SELECTOR
ROW_DEC2
ROW_DEC2
Date: 8/19/92 9:51:22 AM
Name: RAS Sheet
N.V.E.
ROW_DEC_R3
ROW_DEC_L3
ROW_DEC_R2
ROW_DEC_L2
ROW_DEC_R1
ROW_DEC_L1
ROW_DEC_R0
ROW_DEC_L0
NPRECHARGE
DECODE0
DECODE1
DECODE2
DECODE3
DECODE4
DECODE5
DECODE6
DECODE7
ENABLE_DECOD
ROW-DEC1
ROW_DEC_R
ROW_DEC_L
I_28
I_27
I_26
I_25
ADD[0]
ADD[1]
NADD[0]
NADD[1]
NPRECHARGE
DECODE7
DECODE6
DECODE5
DECODE4
DECODE3
DECODE2
ENABLE_DECOD
ADD10_17]
NADD10_17]

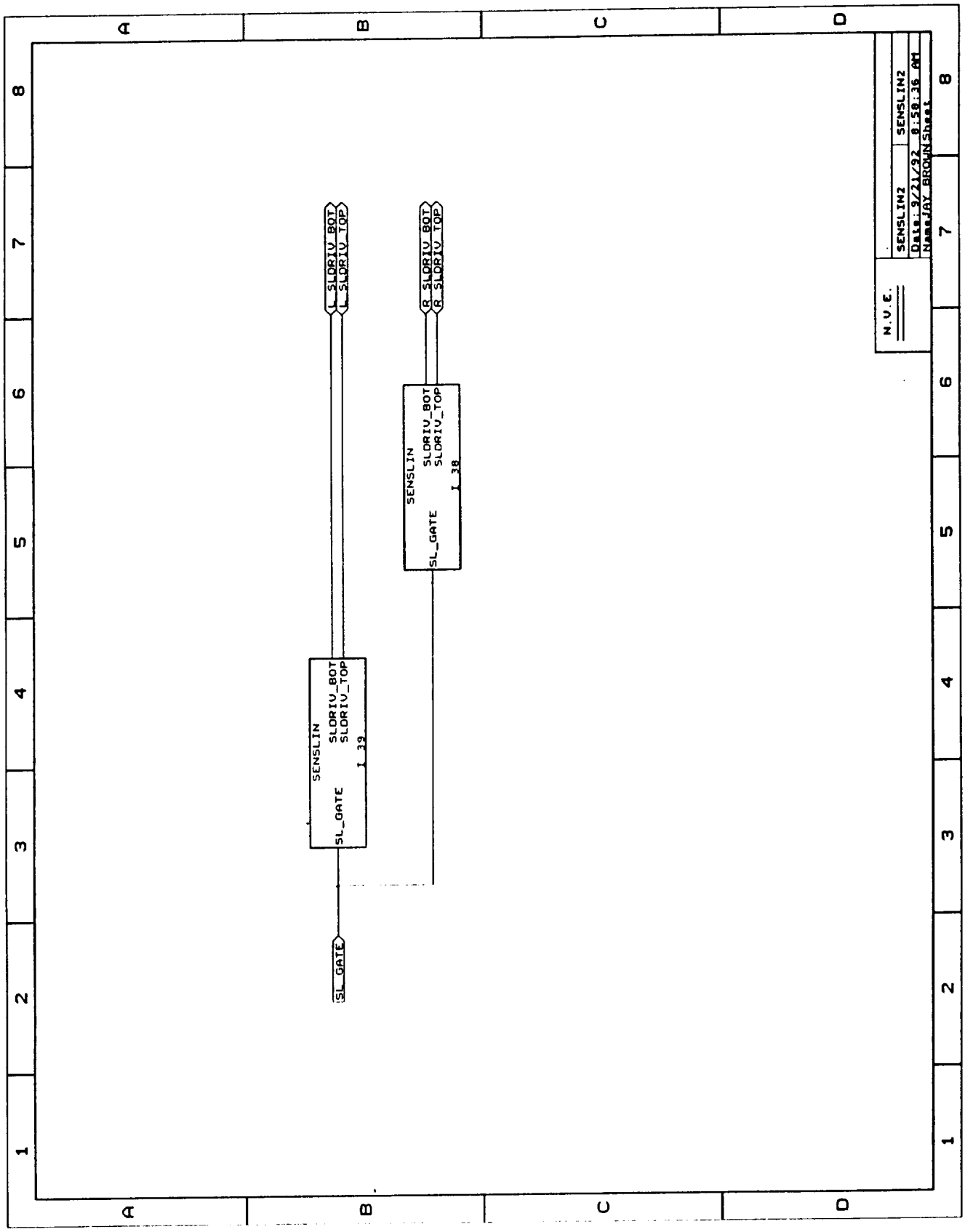

ROW_DECODER_IO_SELECTOR
ROW_DEC3
Date: 8/19/92 11:10:14 AM
Sheet
N.V.E.
ROW_DEC2
NPRECHARGE
DECODE2
DECODE3
DECODE4
DECODE5
DECODE6
DECODE7
ENABLE_DECOD
ADD[0..17]
NADD[0..17]
ROW_DEC_R3
ROW_DEC_L3
ROW_DEC_R2
ROW_DEC_L2
ROW_DEC_R1
ROW_DEC_L1
ROW_DEC_R0
ROW_DEC_L0
I_28
I_27
I_26
I_25
ROW_R[15]
ROW_L[15]
ROW_R[14]
ROW_L[14]
ROW_R[13]
ROW_L[13]
ROW_R[12]
ROW_L[12]
ROW_R[11]
ROW_L[11]
ROW_R[10]
ROW_L[10]
ROW_R[9]
ROW_L[9]
ROW_R[8]
ROW_L[8]
ROW_R[7]
ROW_L[7]
ROW_R[6]
ROW_L[6]
ROW_R[5]
ROW_L[5]
ROW_R[4]
ROW_L[4]
ROW_R[3]
ROW_L[3]
ROW_R[2]
ROW_L[2]
ROW_R[1]
ROW_L[1]
ROW_R[0]
ROW_L[0]
ADD[2]
ADD[3]
NADD[2]
NADD[3]
NPRECHARGE
DECODE4
DECODE5
DECODE6
DECODE7
ENABLE_DECOD
ADD[0..17]
NADD[0..17]

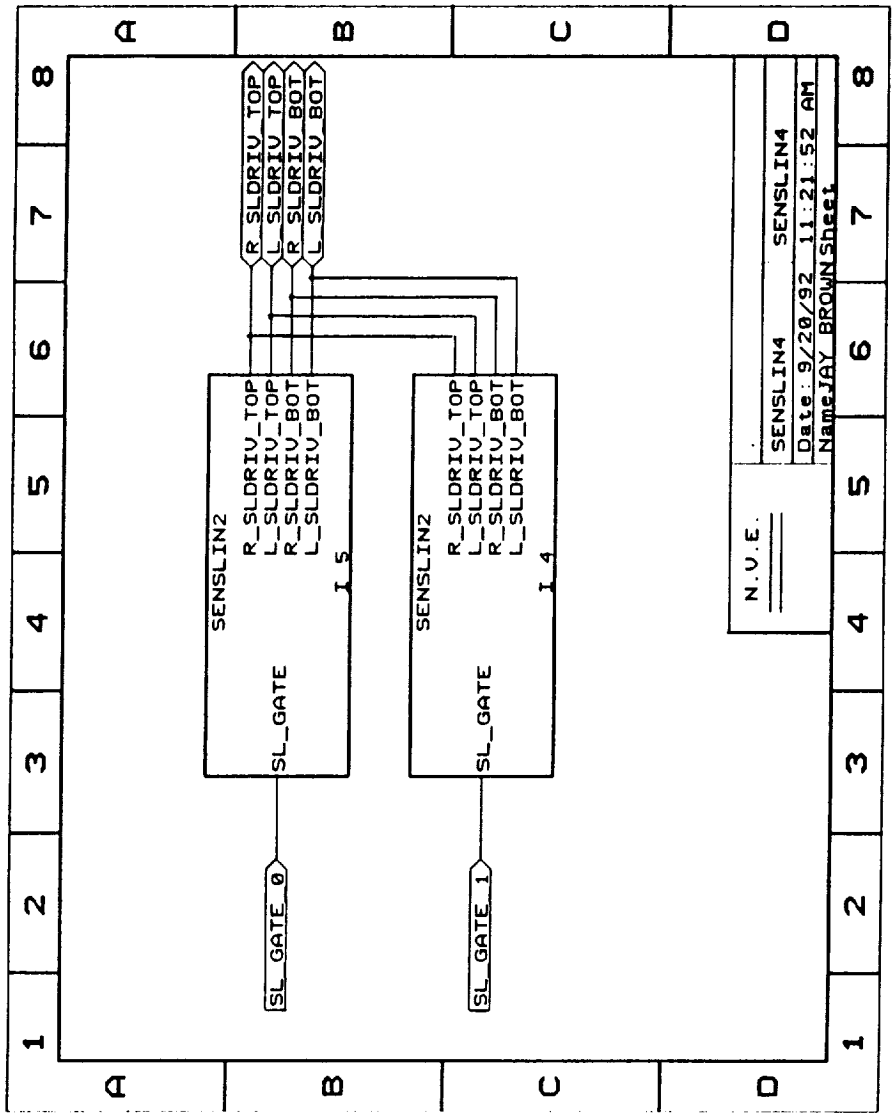

ROW_DECODER_TO_SELECTOR
ROW_DEC4
ROW_DEC4
Date:8/19/92 5:40:10 PM
Name: RAS Sheet
N.V.E.
ROW_R[0,63]
ROW_L[0,63]
ROW_R[48,63]
ROW_L[48,63]
ROW_R[32,47]
ROW_L[32,47]
ROW_R[16,31]
ROW_L[16,31]
ROW_R[0,15]
ROW_L[0,15]
NPRECHARGE ROW_DEC3
DECODE4 ROW_R[0,15]
DECODE5
DECODE6 ROW_L[0,15]
DECODE7
ENABLE_DECOD
ADD[0,17] I_28
NADD[0,17]
NPRECHARGE ROW_DEC3
DECODE4 ROW_R[0,15]
DECODE5
DECODE6 ROW_L[0,15]
DECODE7
ENABLE_DECOD
ADD[0,17] I_27
NADD[0,17]
NPRECHARGE ROW_DEC3
DECODE4 ROW_R[0,15]
DECODE5
DECODE6 ROW_L[0,15]
DECODE7
ENABLE_DECOD
ADD[0,17] I_26
NADD[0,17]
NPRECHARGE ROW_DEC3
DECODE4 ROW_R[0,15]
DECODE5
DECODE6 ROW_L[0,15]
DECODE7
ENABLE_DECOD
ADD[0,17] I_25
NADD[0,17]
ADD[4]
ADD[5]
NADD[4]
NADD[5]
ADD[4]
NADD[5]
NADD[4]
NADD[5]
NPRECHARGE
DECODE6
DECODE7
ENABLE_DECOD
ADD[0,17]
NADD[0,17]

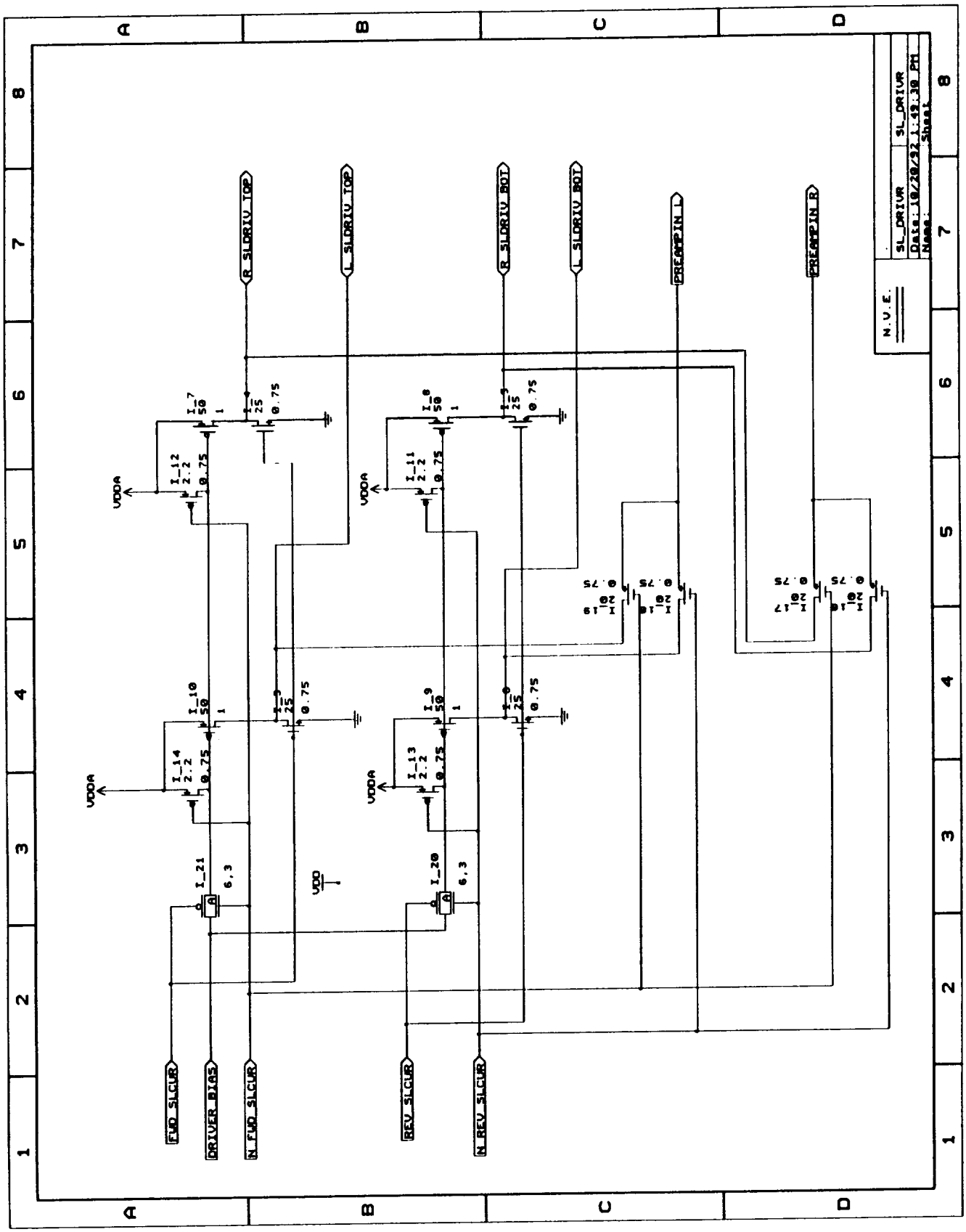

ROW DECODER TO SELECTOR
ROW_DECS
ROW_DECS
Date: 10/1/92 11:14:54 AM
Sheet
ROW SELECT LINES 192 TO 255
ROW SELECT LINES 128 TO 191
ROW SELECT LINES 64 TO 127
ROW SELECT LINES 0 TO 63
NPRECHARGE
DECODE6
DECODE7
ENABLE_DECOD
ADD[0,17]
NADD[0,17]
ROW_DEC4
ROW_R[0,63]
ROW_L[0,63]
ROW_R[192-255]
ROW_L[192-255]
ROW_R[128-191]
ROW_L[128-191]
ROW_R[64-127]
ROW_L[64-127]
ROW_R[0-63]
ROW_L[0-63]
I_28
I_27
I_26
I_25
I_39
I_40
I_38
I_37
R_TOP_SPARE
L_TOP_SPARE
R_BOT_SPARE
L_BOT_SPARE
2.2,2.2
N.U.E.
VDD
I_36
I_33
I_32
I_31
I_30
I_29
I_35
I_34
0.75
2.2
ADD[6]
ADD[7]
NADD[6]
NADD[7]
UPPER/LOWER SELECT
ENABLE_SPARE
2.2,2.2
NPRECHARGE
ENABLE_DECOD
ADD[0,17]
NADD[0,17]

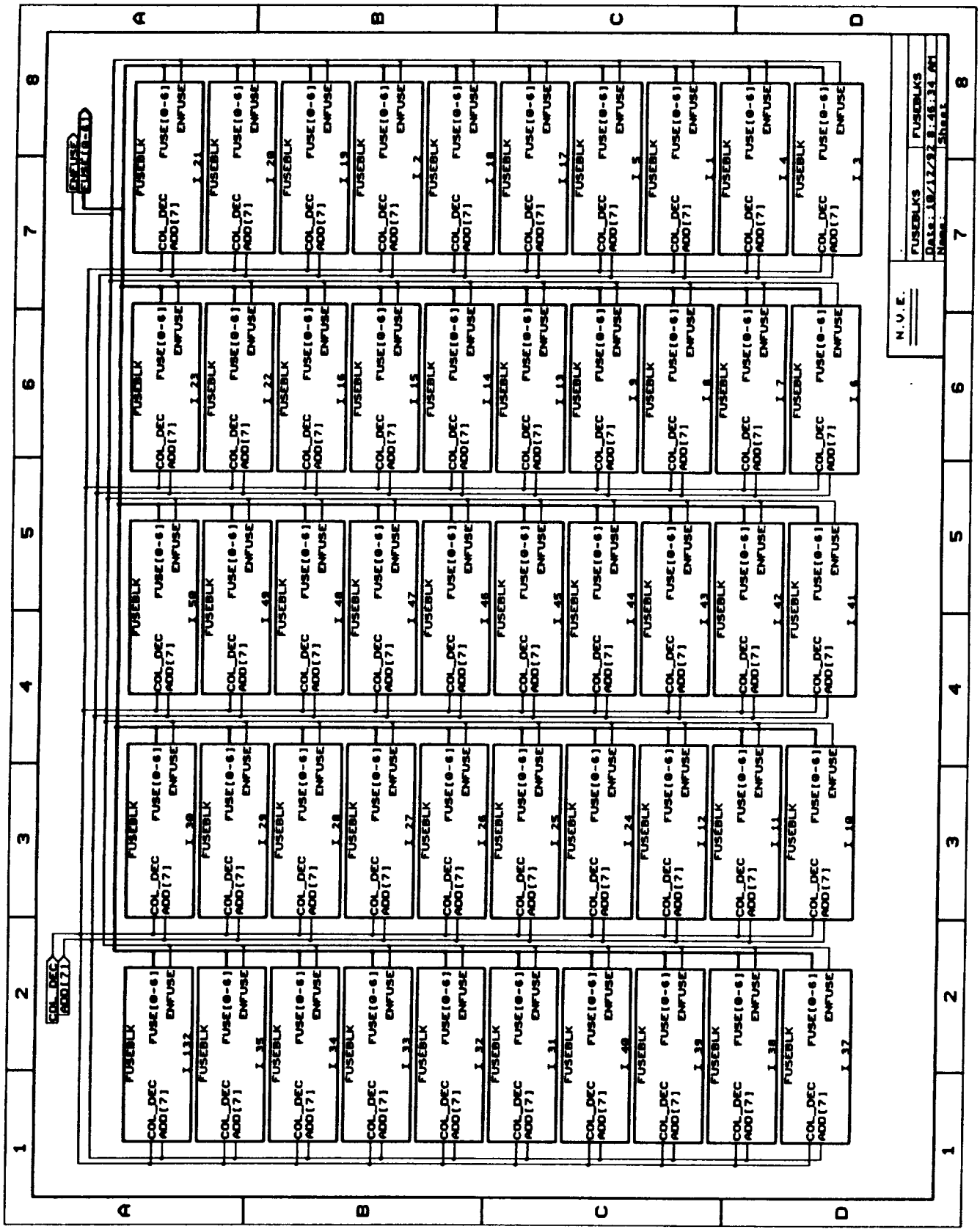

SENSLIN2
TOP_SPARE
SL_GATE
R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
I_12
SL_GATE[0-255]
SL_GATE[0-127]
SL_GATE[128-255]
BLOCK_1K
SL_GATE[0-127]
I_2
NOTE:
TFWD_SLCUR=TOP 1K BLOCK'S FORWARD SENSE LINE CURRENT
TN_FWD_SLCUR=COMPLEMENT OF TFWD_SLCUR
TREV_SLCUR=TOP 1K BLOCK'S REVERSE SENSE LINE CURRENT
TN_REV_SLCUR=COMPLEMENT OF TREV_SLCUR
MID_SEGM
DRIVER_BIAS
COL_DEC
TFWD_SLCUR
TN_FWD_SLCUR
TREV_SLCUR
TN_REV_SLCUR
BFWD_SLCUR
BN_FWD_SLCUR
BREV_SLCUR
BN_REV_SLCUR
UR_SLDRV_TOP
UL_SLDRV_TOP
UR_SLDRV_BOT
UL_SLDRV_BOT
PREAMPIN_L
PREAMPIN_R
LR_SLDRV_TOP
LL_SLDRV_TOP
LR_SLDRV_BOT
LL_SLDRV_BOT
I_13
PREAMPIN_L
PREAMPIN_R
NOTE:
BFWD_SLCUR=BOTTOM 1K BLOCK'S FORWARD SENSE LINE CURRENT
BN_FWD_SLCUR= COMPLEMENT OF BFWD_SLCUR
BREV_SLCUR=BOTTOM 1K BLOCK'S REVERSE SENSE LINE CURRENT
BN_REV_SLCUR=COMPLEMENT OF BREV_SLCUR
BLOCK_1K
SL_GATE[0-127]
I_1
SENSLIN2
BOT_SPARE
SL_GATE
I_11
NOTES:
1. SL_GATE[0-255] COMES FROM THE CENTRAL ROW DECODER AND TURNS ON THE GATE TRANSISTORS TO ALLOW CURRENT TO FLOW THROUGH A SENSE LINE.
3. TOP_SPARE AND BOT_SPARE COME FROM THE SPARE SELECTION CIRCUITRY OUTSIDE THE MEMORY ARRAY. THEY TURN ON THE SPARE SENSE LINES IN THE TOP OR BOTTOM HALVES OF THE SEGMENT, WHILE AT THE SAME TIME THE ENABLE LINE FOR THE CENTRAL ROW DECODER IS TURNED OFF.
SEE LOWER LEVELS OF THE HIERARCHY FOR SIGNALS WHICH ARE NOT EXPLAINED HERE.
N.V.E.
1 MEGABIT BASIC SEGMENT
SEGMENT
SEGMENT
Date: 10/20/92 1:52:14 PM
Name: TJD/JLB Sheet

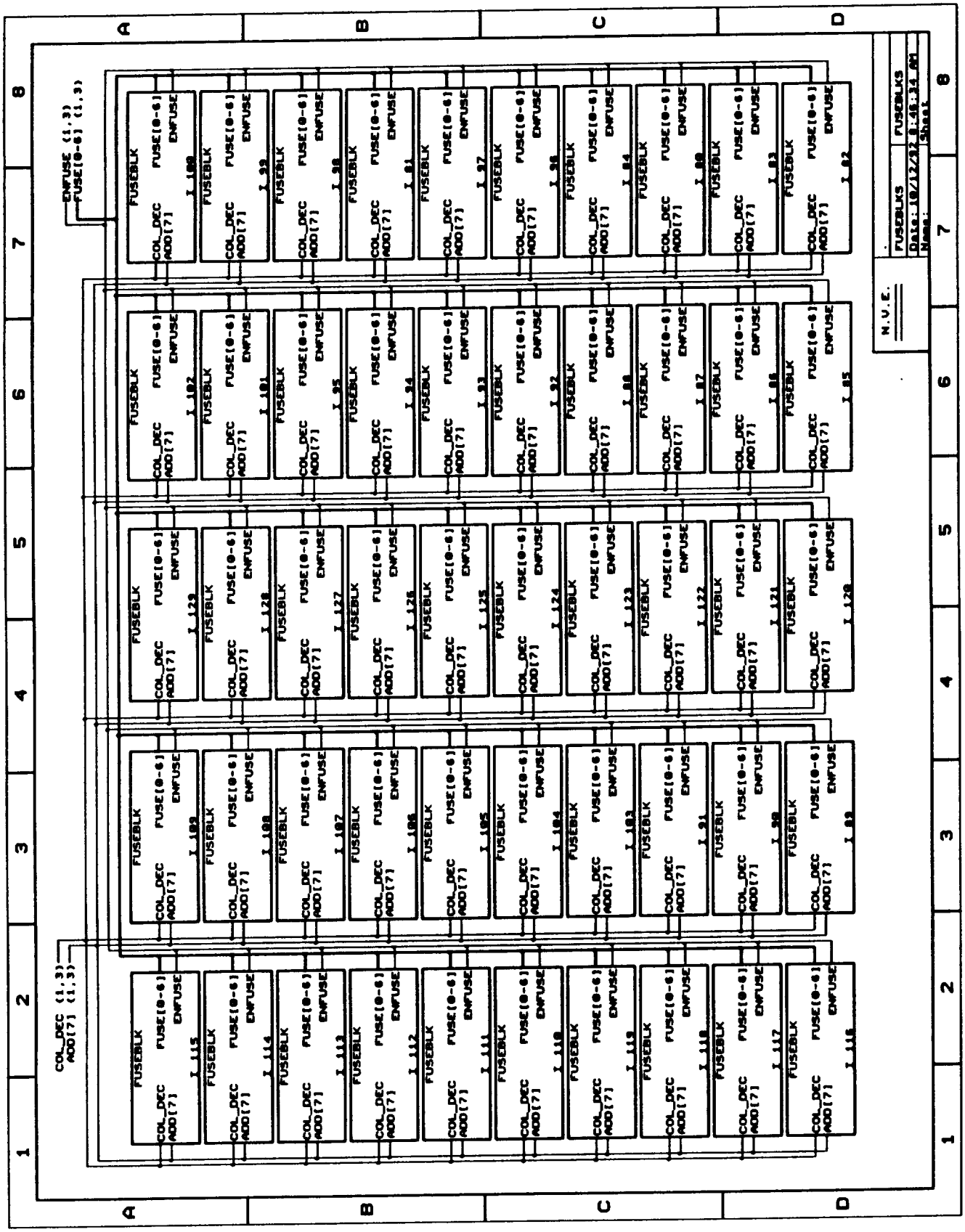

SLDRIV_BOT
SL_GATE
SLDRIV_TOP
80 Ohm MRAM Bit
0
33.7
0.8
63.6
64.5
70.2
72.6
N.V.E.
SENSELIN
SENSELIN
Date: 9/23/92 2:52:10 PM
Name: JAY BROWN Sheet
NOTES:
1. THIS IS THE BASIC SENSE LINE CONFIGURATION. THE END OF THE SENSE LINE WITH THE TRANSISTORS ATTACHED IS THE "BOTTOM" OF THE SENSE LINE. THIS CONVENTION WILL BE ADOPTED TO DEFINE FORWARD AND REVERSE CURRENT DIRECTIONS.
2. THE SENSE LINE CURRENT IS DEFINED TO BE FORWARD WHEN IT FLOWS FROM THE TOP OF THE SENSE LINE TO THE BOTTOM, AND REVERSE WHEN IT FLOWS FROM BOTTOM TO TOP.

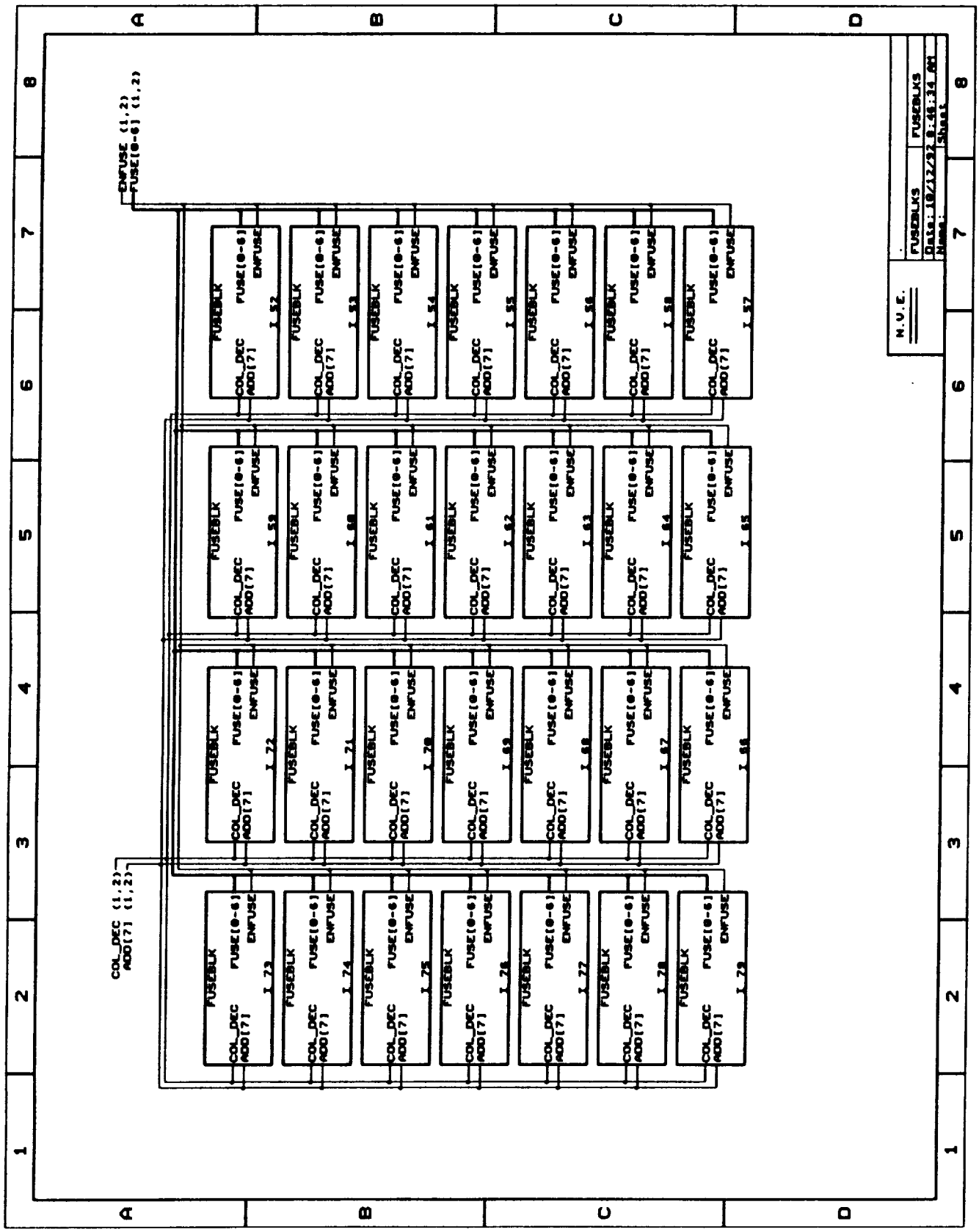

A
B
C
D
1
2
3
4
5
6
7
8
L_SLORIV_BOT
L_SLORIV_TOP
R_SLORIV_BOT
R_SLORIV_TOP
SENSLIN
SLORIV_BOT
SLORIV_TOP
SL_GATE
I_39
SENSLIN
SLORIV_BOT
SLORIV_TOP
SL_GATE
I_38
SL_GATE
N.V.E.
SENSLIN2
SENSLIN2
Date:9/21/92 8:50:36 AM
Name:JAY BROWN Sheet

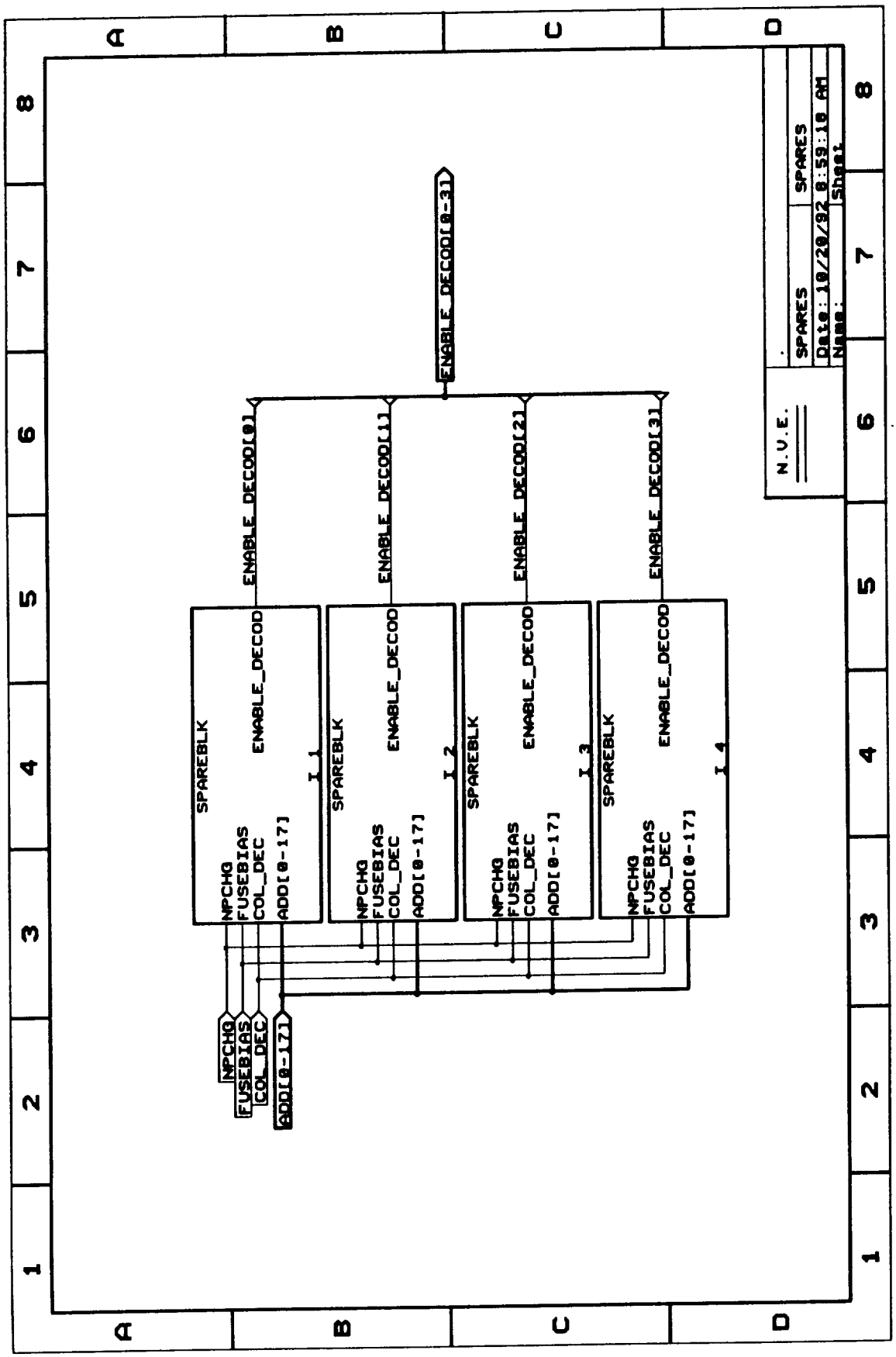

R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
SENSLIN2
R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
I 5
SL_GATE
SL_GATE_0
SENSLIN2
R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
I 4
SL_GATE
SL_GATE_1
N.V.E.
SENSLIN4
SENSLIN4
Date:9/20/92 11:21:52 AM
NameJAY BROWN Sheet

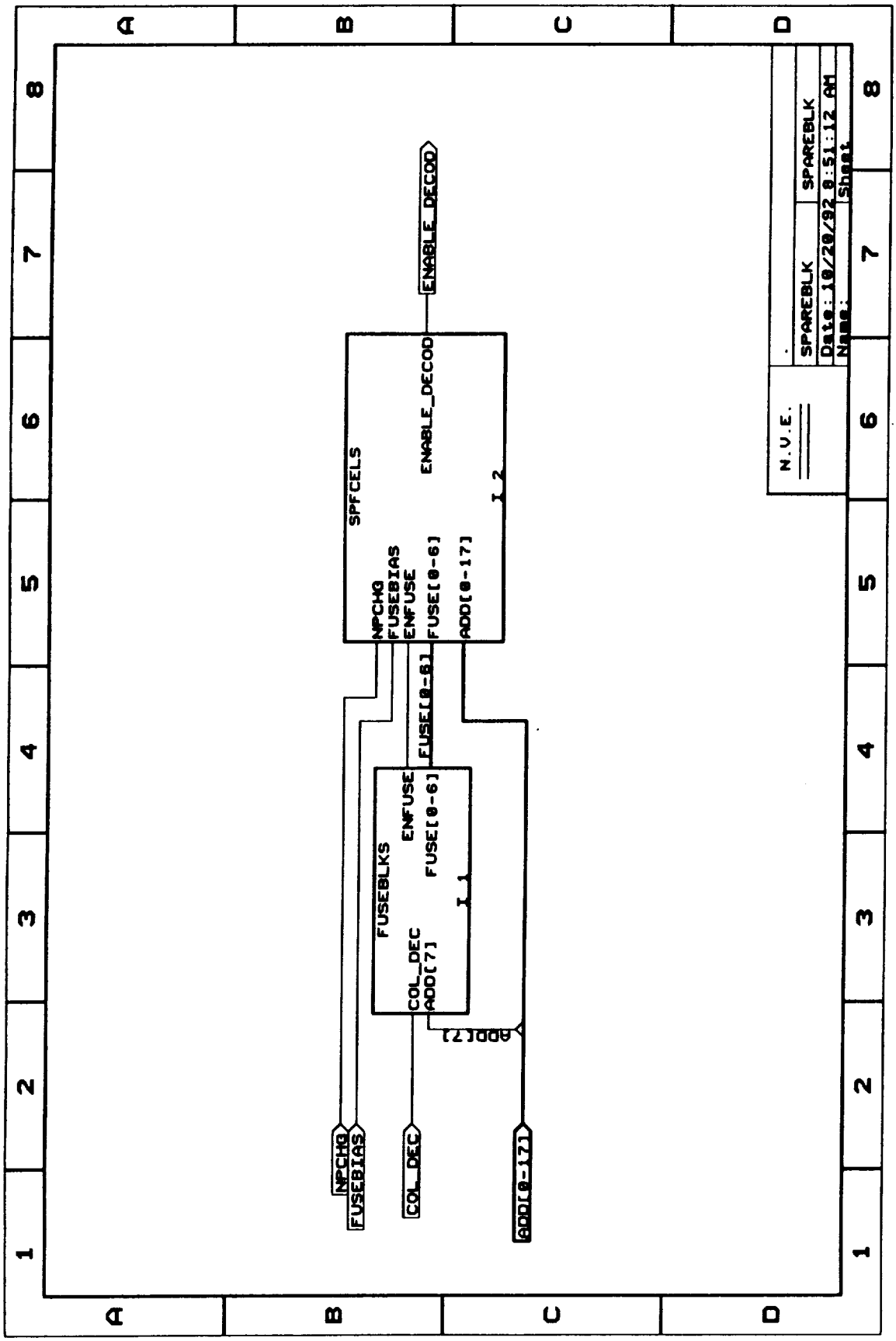

R_SLDRIV_TOP
L_SLDRIV_TOP
R_SLDRIV_BOT
L_SLDRIV_BOT
PREAMPIN_L
PREAMPIN_R
VDDA
VDD
FWD_SLCUR
DRIVER_BIAS
N_FWD_SLCUR
REV_SLCUR
N_REV_SLCUR
N.V.E.
SL_DRIVR
SL_DRIVR
Date: 10/20/92 1:49:30 PM
Sheet

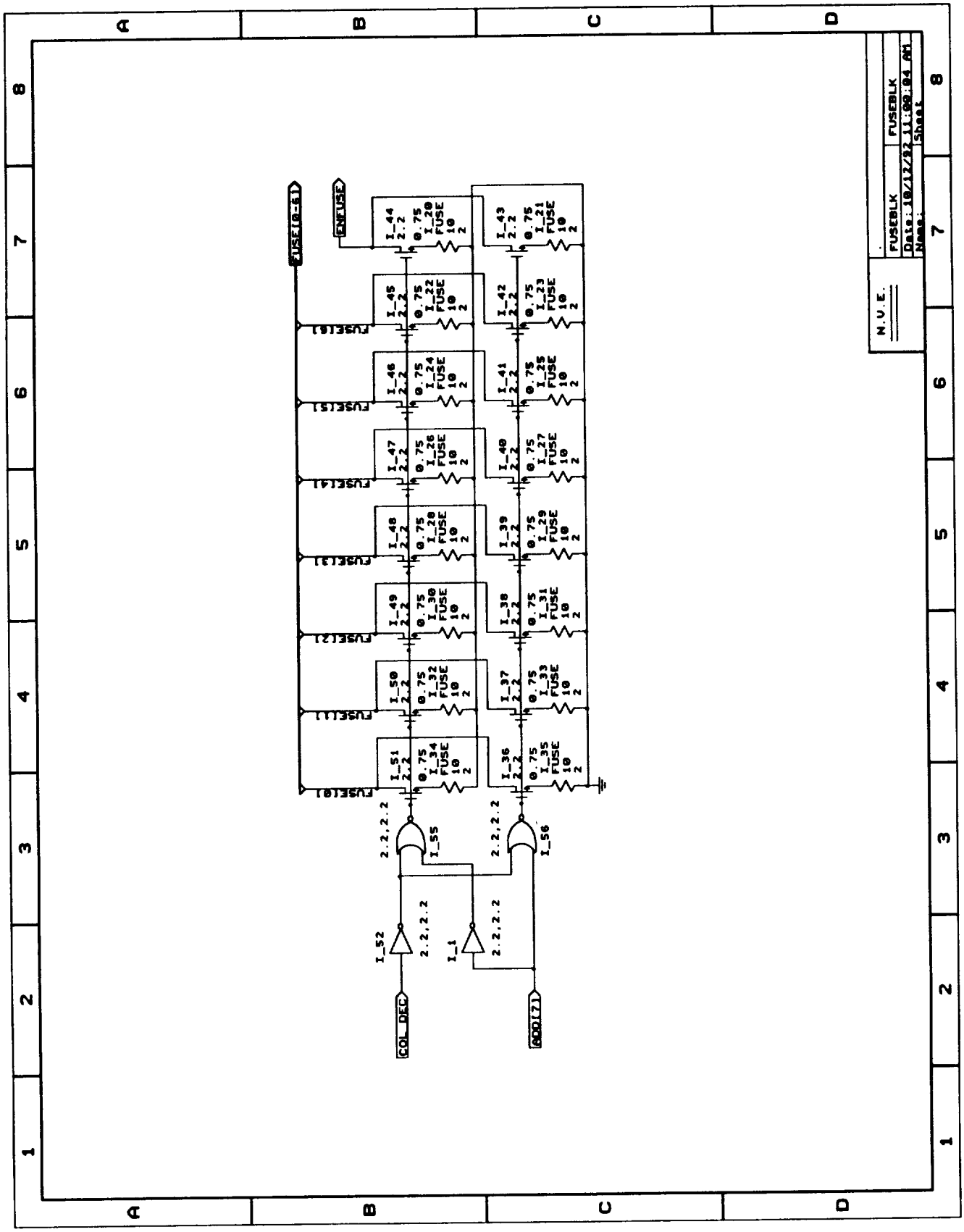

FUSEBLK
COL_DEC
ADD[7]
FUSE[0-6]
ENFUSE
N.V.E.
FUSEBLK3
Date: 10/12/92 8:46:34 AM
Sheet

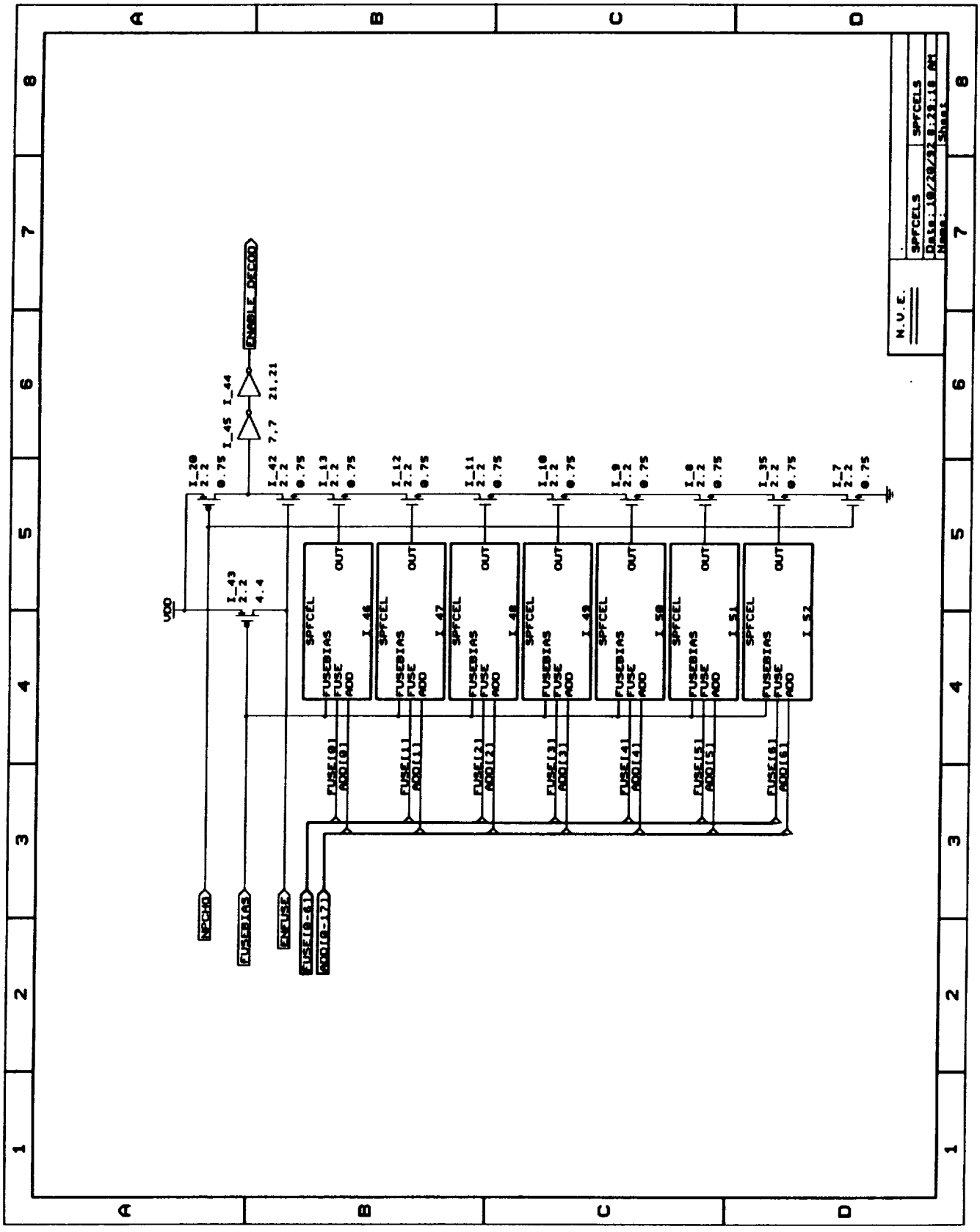

A
B
C
D
1
2
3
4
5
6
7
8
ENFUSE [1:3]
FUSE[0-6] [1:3]
COL_DEC [1:3]
ADD[7] [1:3]
FUSEBLK
COL_DEC
ADD[7]
FUSE[0-6]
ENFUSE
N.V.E.
FUSEBLKS
FUSEBLKS
Date: 10/12/92 8:46:34 AM
Sheet
Name:

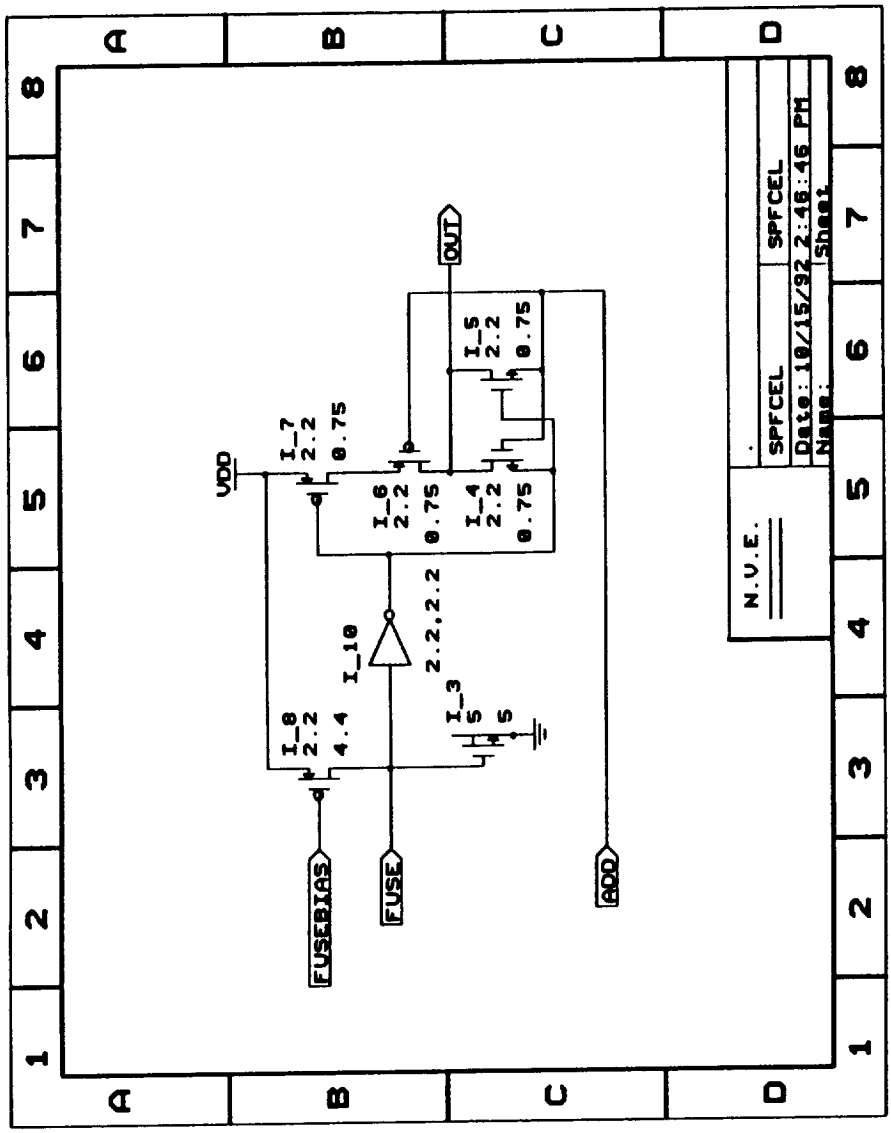

ENFUSE (1,2)
FUSE[0-6] (1,2)
COL_DEC (1,2)
ADD[7] (1,2)
FUSEBLK
COL_DEC
ADD[7]
FUSE[0-6]
ENFUSE
N.U.E.
FUSEBLKS
Date: 10/17/92 8:46:34 AM
Sheet

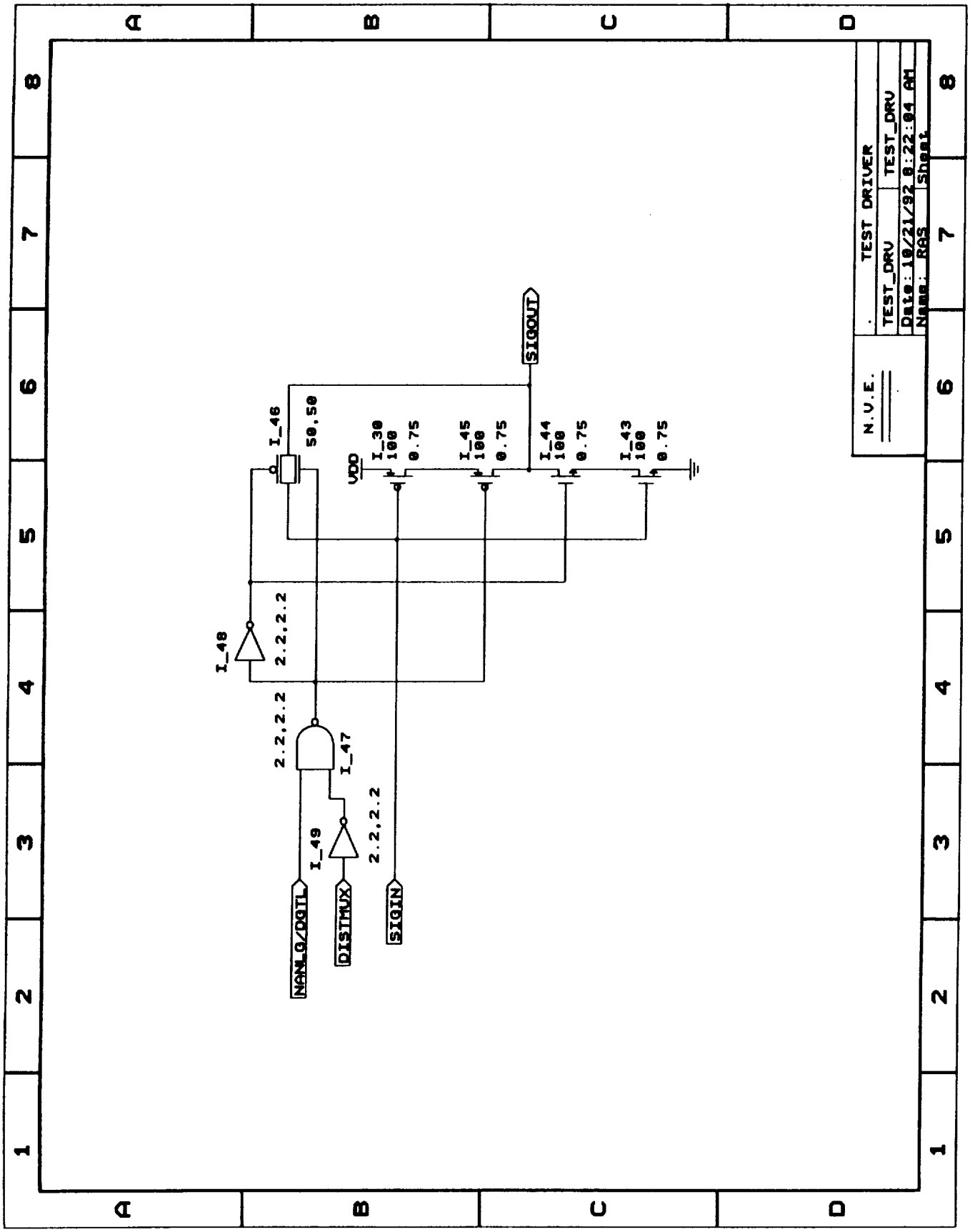

A
B
C
D
1
2
3
4
5
6
7
8
ENABLE_DECOD[0-31]
ENABLE_DECOD[0]
ENABLE_DECOD[1]
ENABLE_DECOD[2]
ENABLE_DECOD[3]
SPAREBLK
ENABLE_DECOD
NPCHG
FUSEBIAS
COL_DEC
ADD[0-17]
I_1
I_2
I_3
I_4
N.V.E.
SPARES
SPARES
Date: 10/20/92 0:59:10 AM
Name:
Sheet

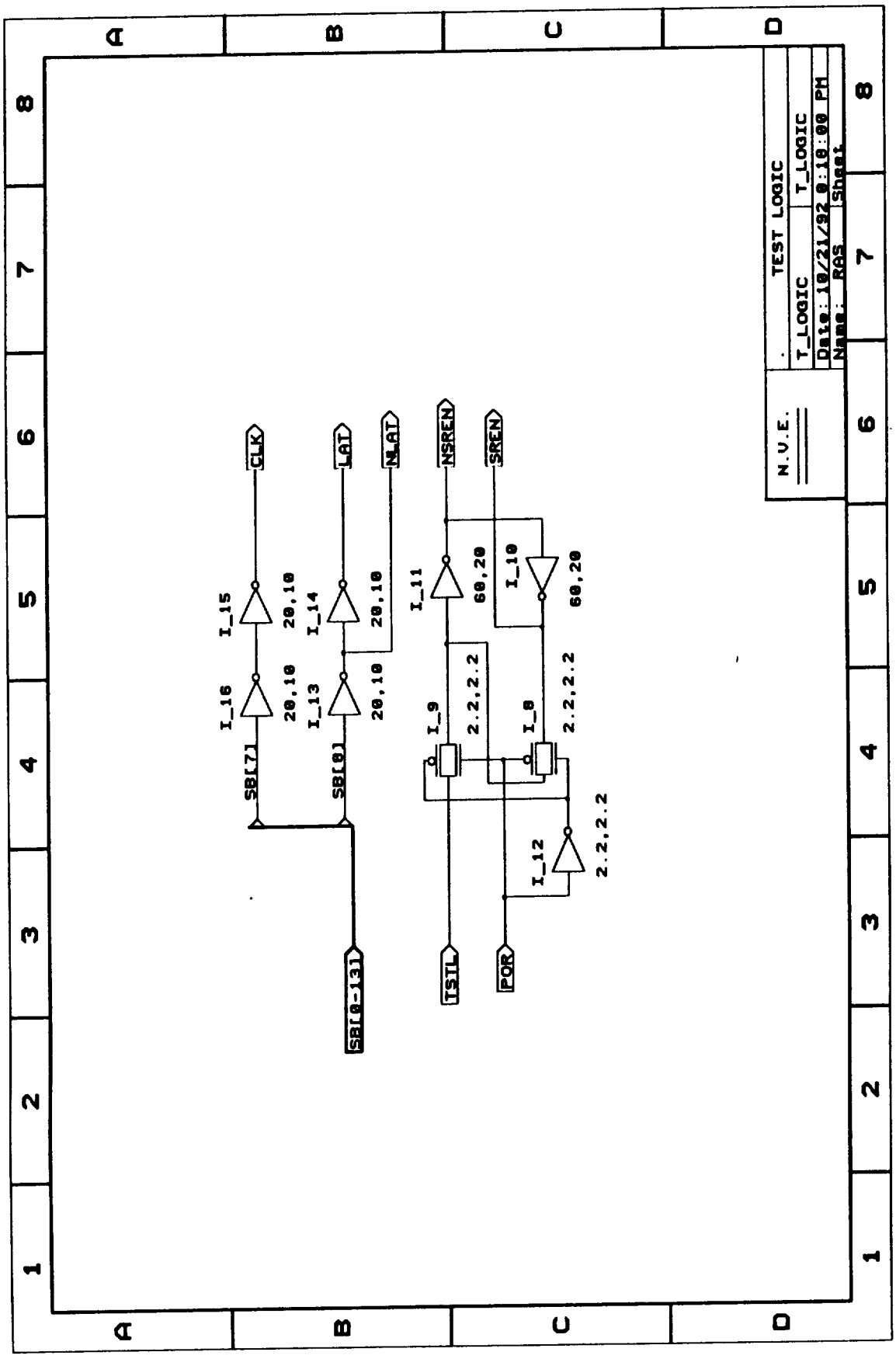

A
B
C
D
1
2
3
4
5
6
7
8
ENABLE_DECOD
SPFCELS
NPCHG
FUSEBIAS
ENFUSE
FUSE[0-6]
ADD[0-17]
ENABLE_DECOD
X2
FUSE[0-6]
FUSEBLKS
ENFUSE
FUSE[0-6]
COL_DEC
ADD[7]
X1
ADD[2]
NPCHG
FUSEBIAS
COL_DEC
ADD[0-17]
N.V.E.
SPAREBLK
SPAREBLK
Date: 10/20/92 8:51:12 AM
Name:
Sheet

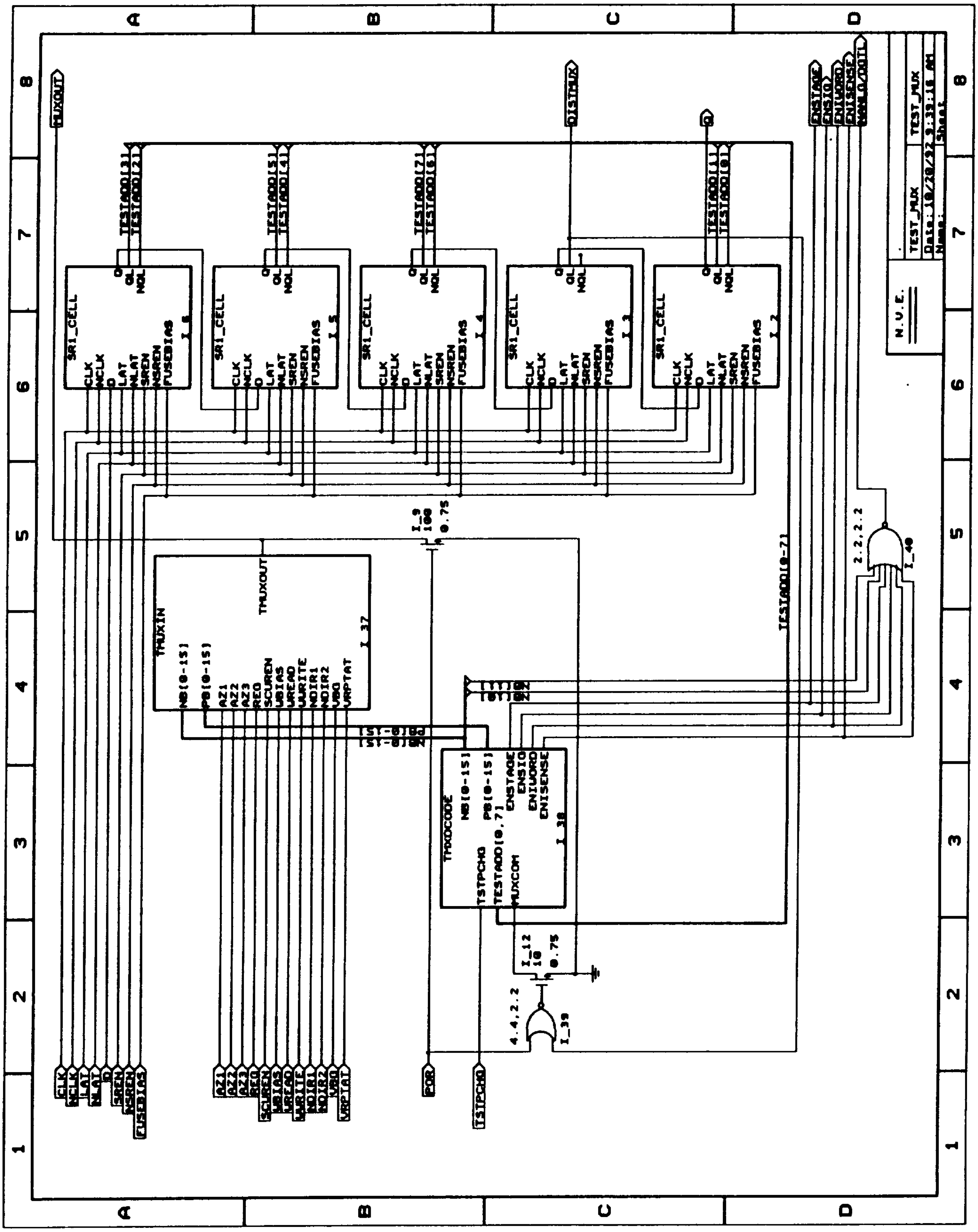

FUSEBLK
FUSEBLK
N.V.E.
FUSE[0-6]
ENFUSE
COL_DEC
ADD[7]
I_52
I_1
I_55
I_56
FUSE[0]
FUSE[1]
FUSE[2]
FUSE[3]
FUSE[4]
FUSE[5]
FUSE[6]

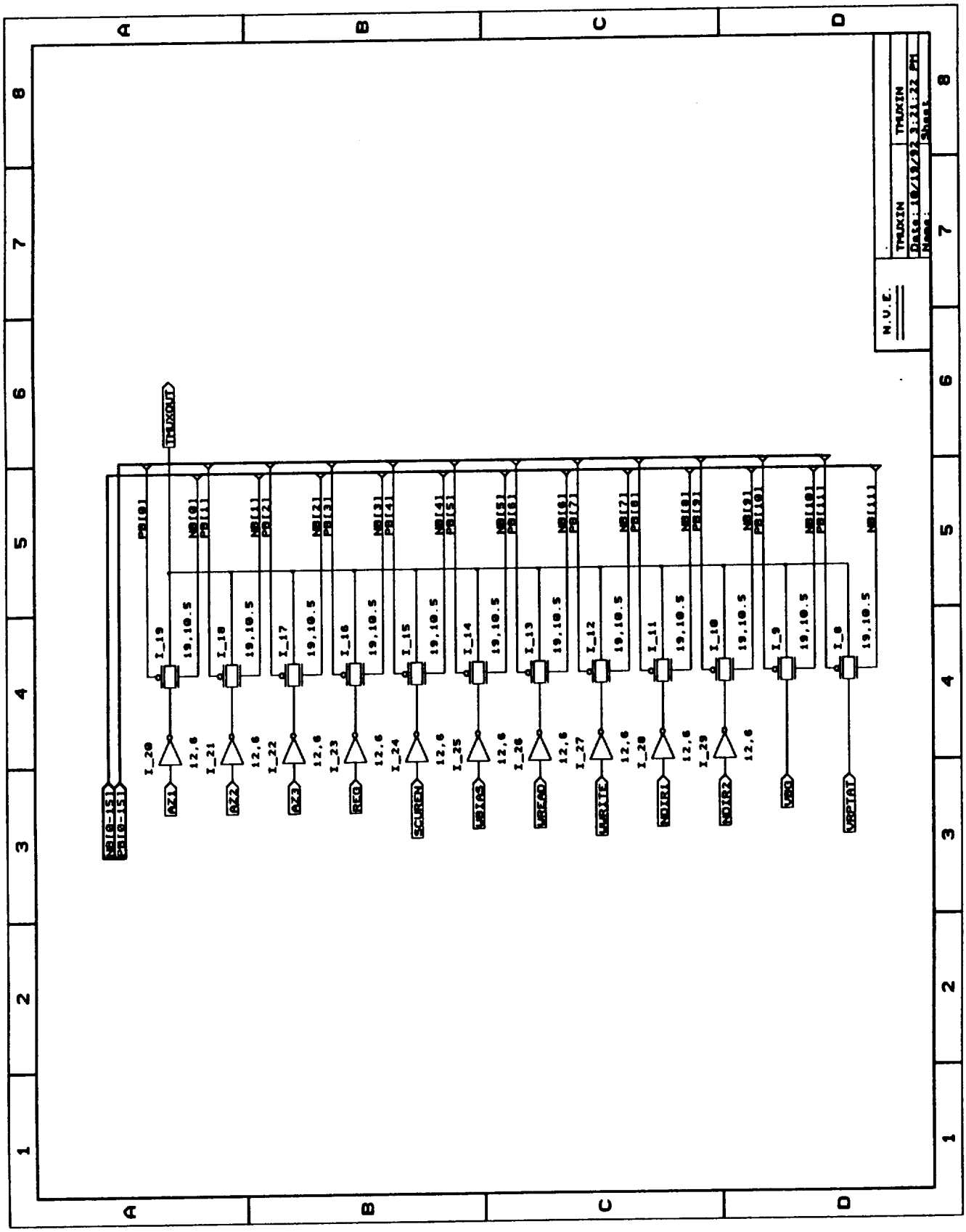

ENABLE_DECOD
I_44
I_45
I_42
I_43
I_20
I_13
I_12
I_11
I_10
I_9
I_8
I_35
I_7
VDD
SPFCEL
FUSEBIAS
FUSE
ADD
OUT
I_46
I_47
I_48
I_49
I_50
I_51
I_52
FUSE[0]
ADD[0]
FUSE[1]
ADD[1]
FUSE[2]
ADD[2]
FUSE[3]
ADD[3]
FUSE[4]
ADD[4]
FUSE[5]
ADD[5]
FUSE[6]
ADD[6]
NPCHG
FUSEBIAS
ENFUSE
FUSE[0-6]
ADD[0-17]
N.V.E.
SPFCELS

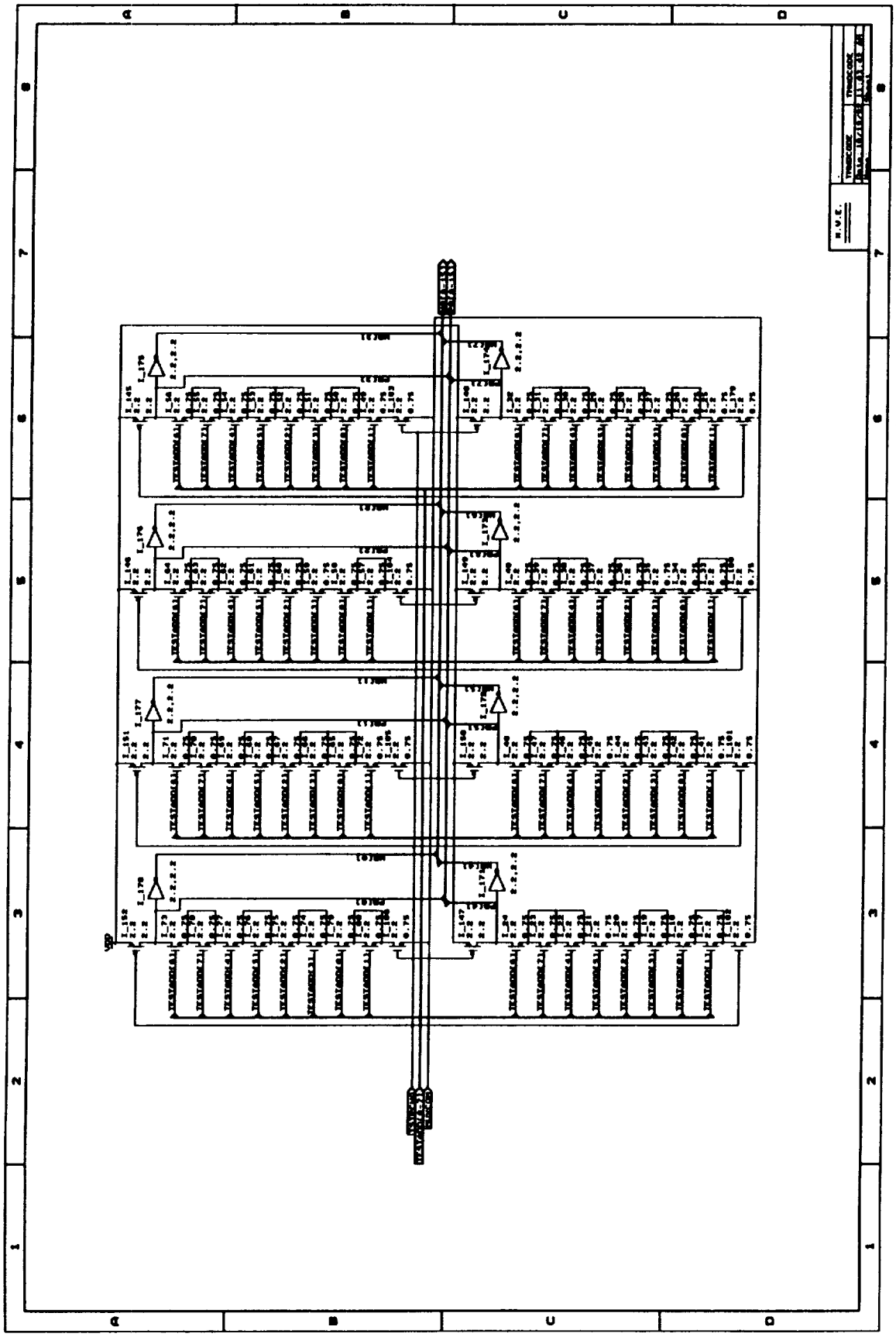

A
B
C
D
1
2
3
4
5
6
7
8
VDD
OUT
FUSEBIAS
FUSE
ADD
I_7 2.2 0.75
I_6 2.2 0.75
I_5 2.2 0.75
I_4 2.2 0.75
I_10 2.2,2.2,2
I_8 2.2 4.4
I_3 5 5
N.V.E.
SPFCEL
SPFCEL
Date:10/15/92 2:46:46 PM
Name: Sheet

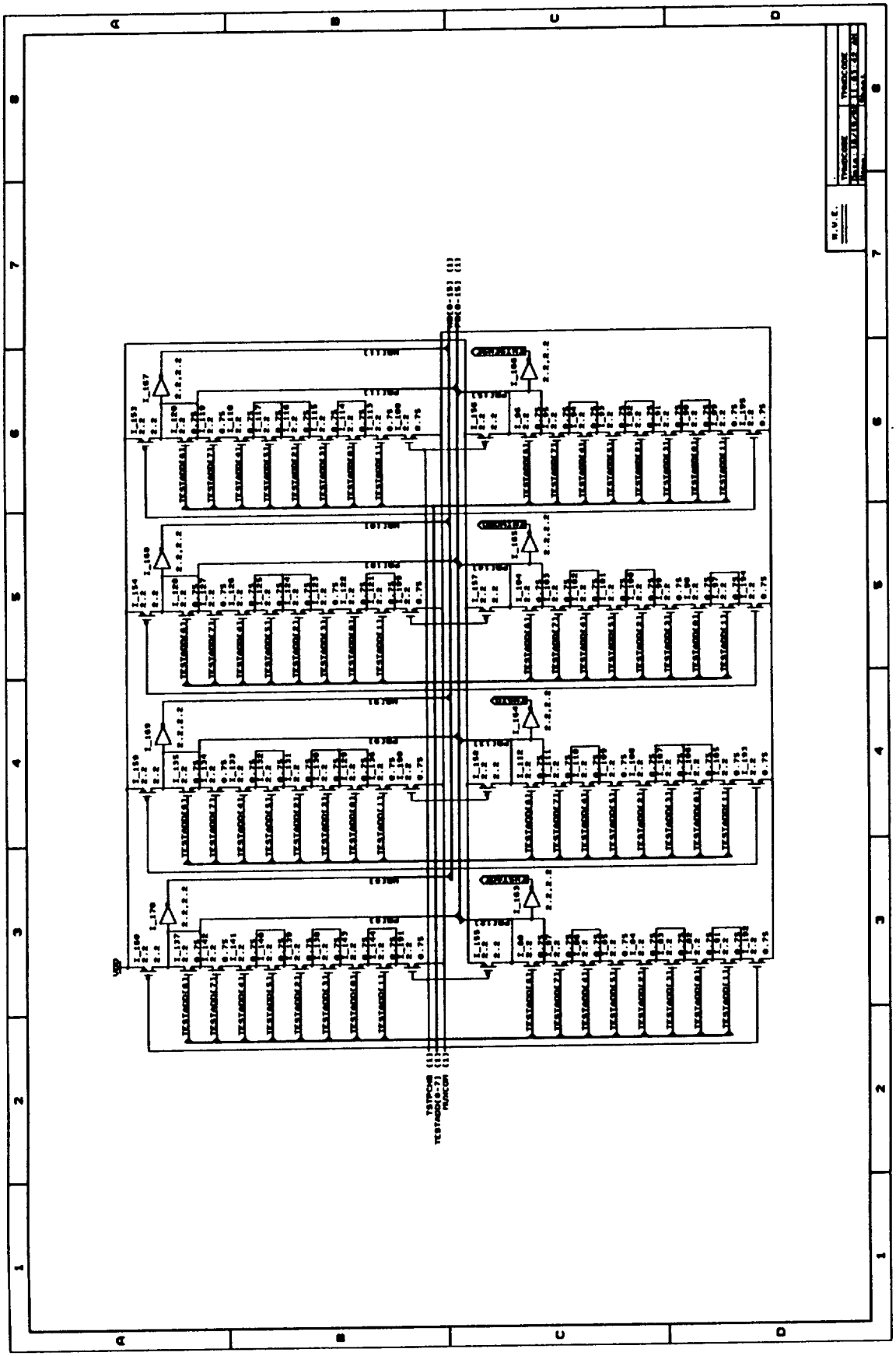

A
B
C
D
1
2
3
4
5
6
7
8
TEST DRIVER
TEST_DRV
TEST_DRV
Date: 10/21/92 8:22:04 AM
Name: RAS
Sheet
N.V.E.
SIGOUT
I_46
50,50
VDD
I_30
100
0.75
I_45
100
0.75
I_44
100
0.75
I_43
100
0.75
I_48
2.2,2.2.2
2.2,2.2.2
I_47
I_49
2.2,2.2.2
NAND/DGTL
DISIMUX
SIGIN

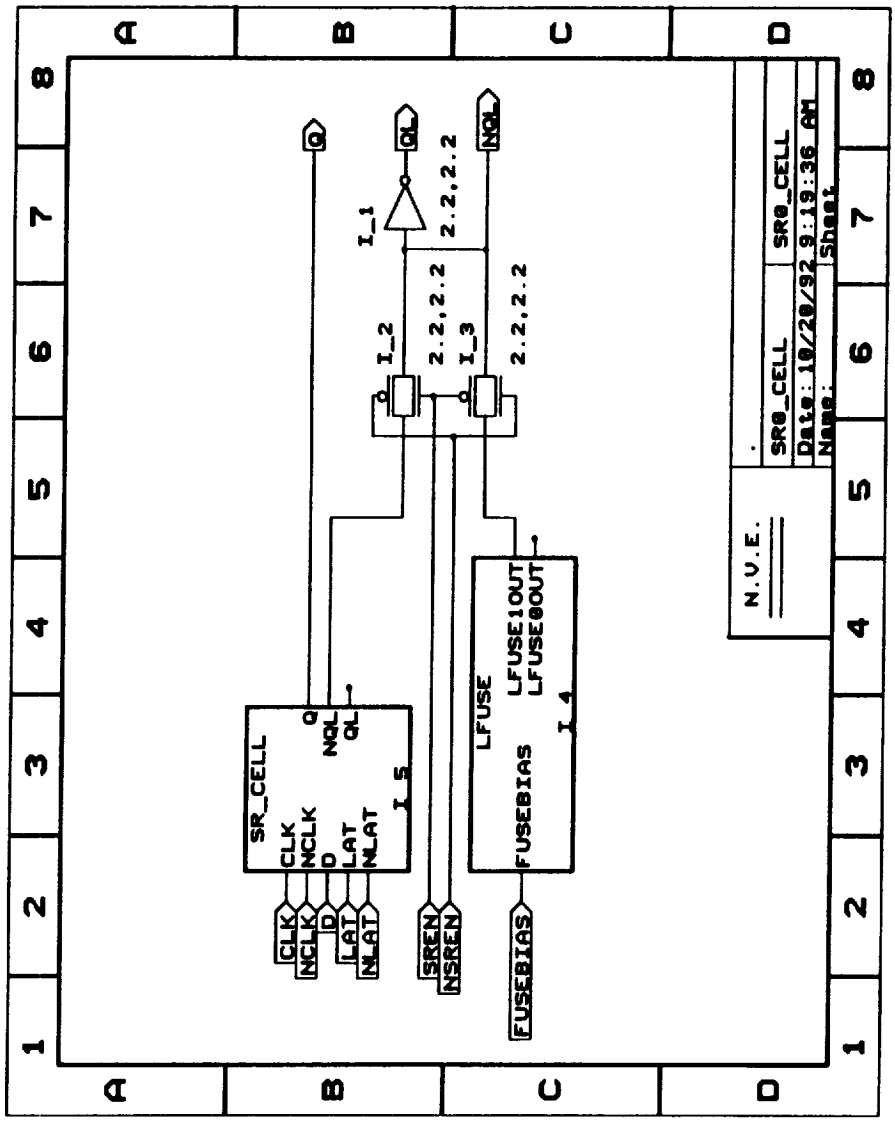

TEST LOGIC
T_LOGIC
T_LOGIC
Date: 10/21/92 0:10:00 PM
Name: RDS Sheet
N.V.E.
CLK
LAT
NLAT
NSREN
SREN
I_15
I_16
I_14
I_13
I_11
I_10
I_9
I_8
I_12
20,10
60,20
2.2,2.2.2
SB[7]
SB[0]
SB[0-13]
TSTL
POR

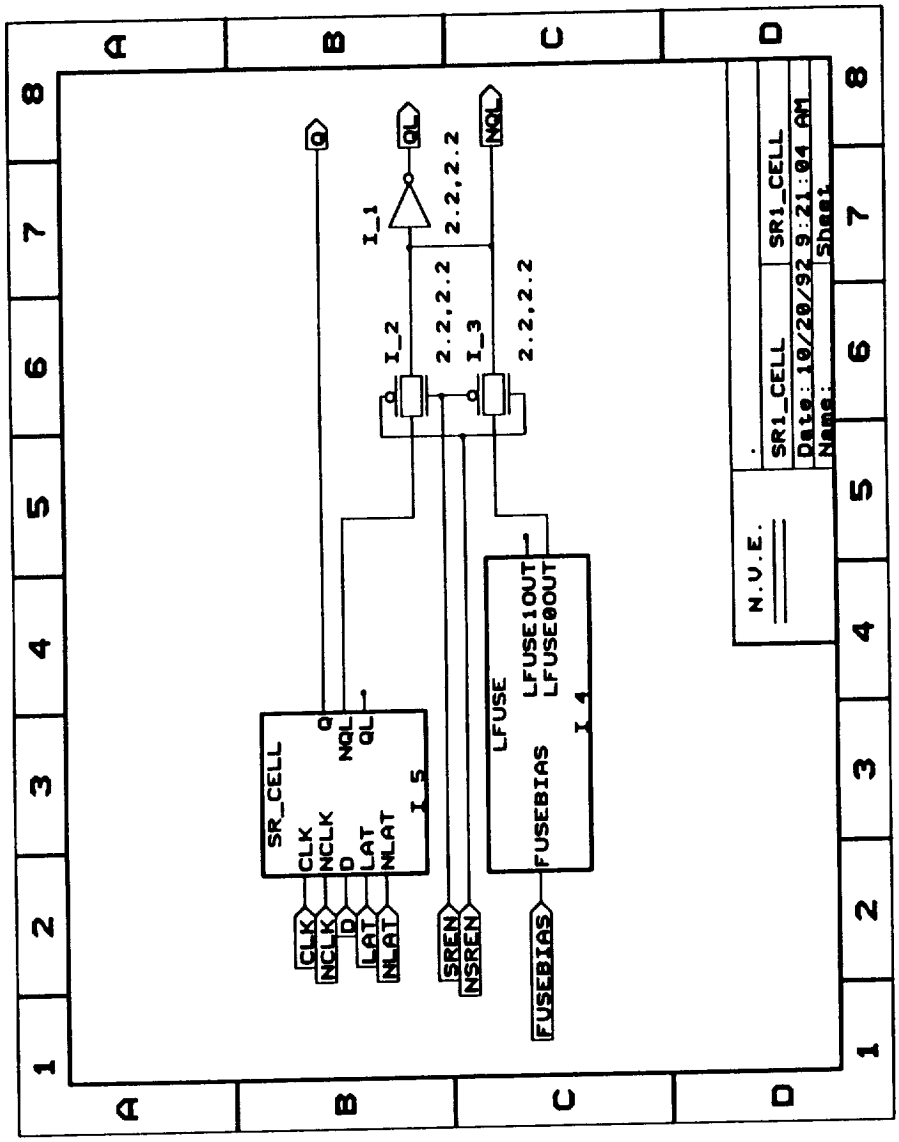

TMUXOUT
DISTMUX
SR1_CELL
CLK
NCLK
D
LAT
NLAT
SREN
NSREN
FUSEBIAS
Q
QL
NQL
TESTADD[0-7]
TMUXIN
NB[0-15]
PB[0-15]
AZ1
AZ2
AZ3
REG
SCUREN
WBIAS
WREAD
WWRITE
NDIR1
NDIR2
VBG
VRPTAT
TMXDCODE
TSTPCHG
TESTADD[0,7]
MUXCOM
ENSTAGE
ENSIO
ENIWORD
ENISENSE
POR
N.V.E.
TEST_MUX

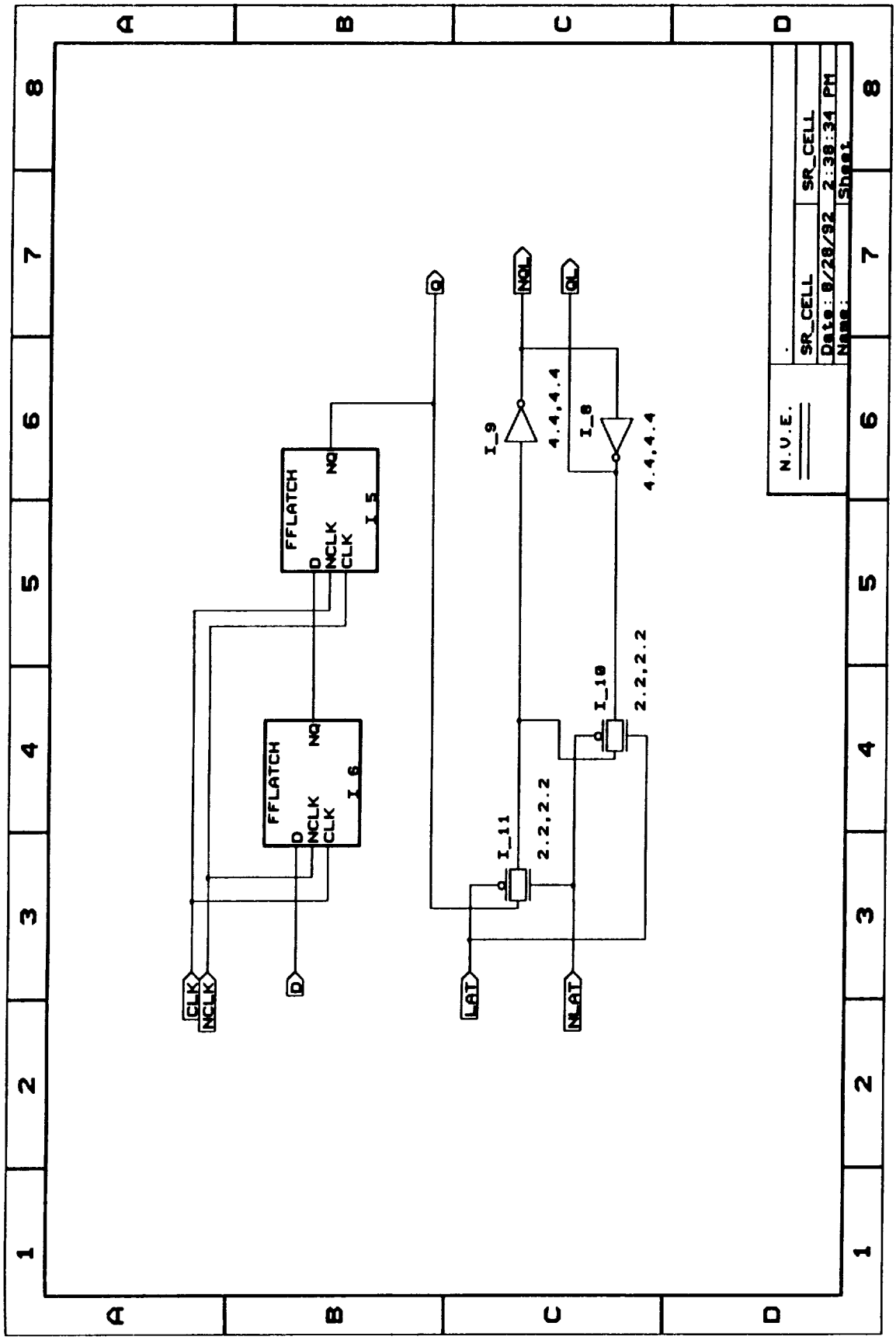

A
B
C
D
1
2
3
4
5
6
7
8
N.U.E.
TMUXIN
TMUXIN
Date: 10/19/92 3:31:22 PM
Sheet
TMUXOUT
PB[0]
NB[0]
PB[1]
NB[1]
PB[2]
NB[2]
PB[3]
NB[3]
PB[4]
NB[4]
PB[5]
NB[5]
PB[6]
NB[6]
PB[7]
NB[7]
PB[8]
NB[8]
PB[9]
NB[9]
PB[10]
NB[10]
PB[11]
NB[11]
I_19
I_18
I_17
I_16
I_15
I_14
I_13
I_12
I_11
I_10
I_9
I_8
19,10.5
I_20
I_21
I_22
I_23
I_24
I_25
I_26
I_27
I_28
I_29
12,6
AZ1
AZ2
AZ3
REG
SCLRGN
VBIAS
VREAD
VWRITE
VDIR1
VDIR2
VBO
VRPTAT
NB[10-15]
PB[10-15]

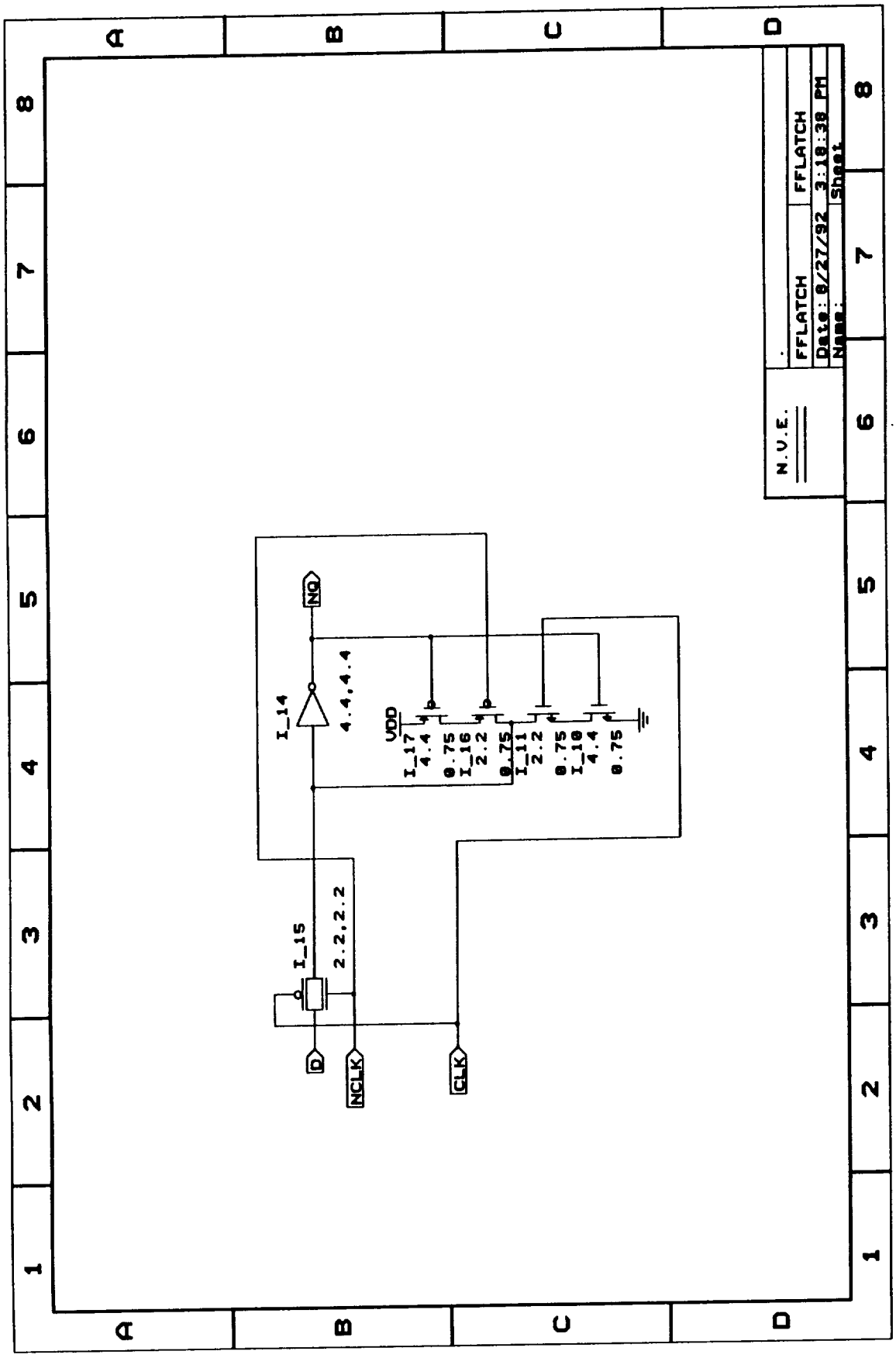


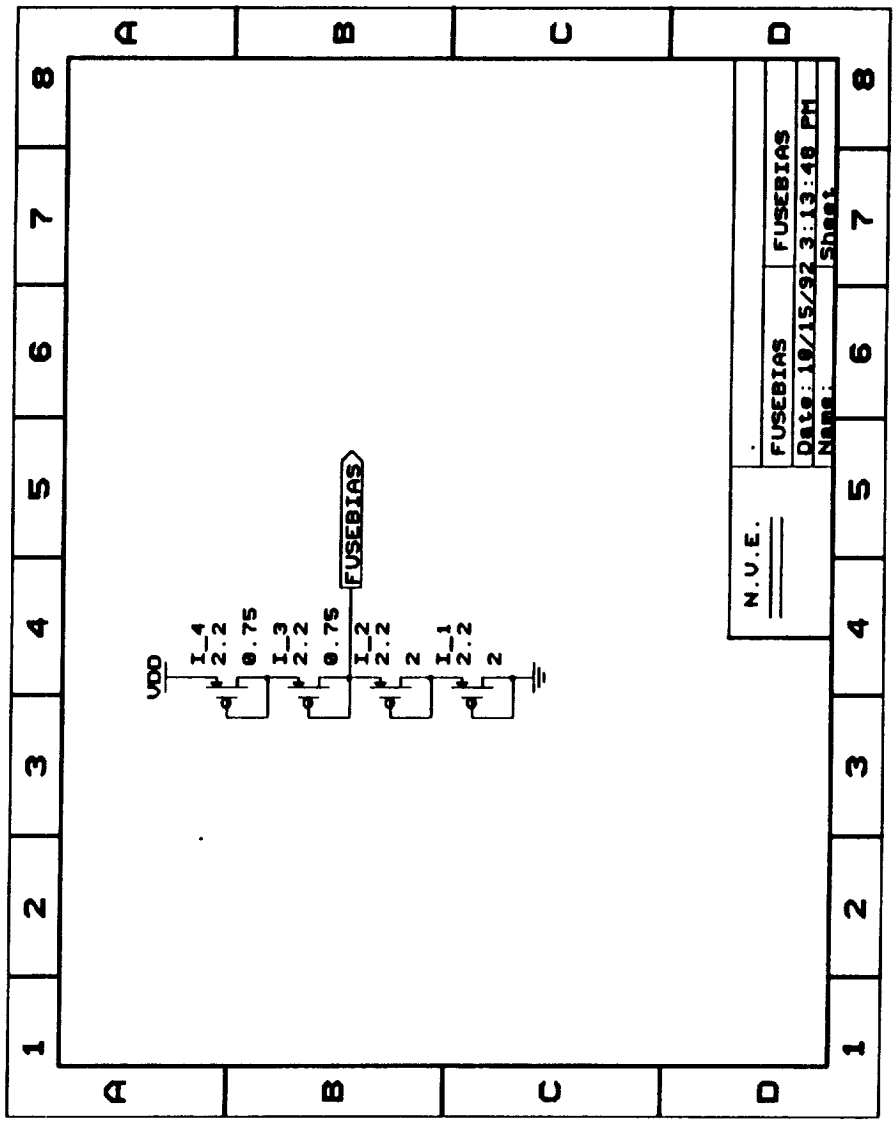



A
B
C
D
1
2
3
4
5
6
7
8
Q
QL
NQL
I_1
2.2,2.2.2
I_2
I_3
SR_CELL
CLK
NCLK
D
LAT
NLAT
I_5
LFUSE
LFUSE1OUT
LFUSE0OUT
FUSEBIAS
I_4
SREN
NSREN
N.U.E.
SR0_CELL
Date: 10/20/92 9:19:36 AM
Name:
Sheet


A
B
C
D
1
2
3
4
5
6
7
8
Q
QL
NQL
I_1
2.2,2.2.2
I_2
I_3
SR_CELL
CLK
NCLK
D
LAT
NLAT
I_5
LFUSE
LFUSE1OUT
LFUSE0OUT
FUSEBIAS
I_4
SREN
NSREN
N.V.E.
SR1_CELL
Date: 10/20/92 9:21:04 AM
Name:
Sheet


A
B
C
D
1
2
3
4
5
6
7
8
FFLATCH
D
NCLK
CLK
NQ
I_5
I_6
CLK
NCLK
D
Q
NQL
QL
I_9
I_8
4.4,4.4
4.4,4.4
I_10
2.2,2.2
I_11
2.2,2.2
LAT
NLAT
N.U.E.
SR_CELL
SR_CELL
Date: 8/28/92 2:38:34 PM
Name:
Sheet:


A
B
C
D
8
7
6
5
4
3
2
1
N.V.E.
FFLATCH
FFLATCH
Date: 8/27/92 3:18:38 PM
Sheet
Name:
NQ
I_14
4.4,4.4
VDD
I_17
4.4
0.75
I_16
2.2
0.75
I_11
2.2
0.75
I_10
4.4
0.75
I_15
2.2,2.2
D
NCLK
CLK


A
B
C
D
1
2
3
4
5
6
7
8
FUSEBIAS
I_4
2.2
0.75
I_3
2.2
0.75
I_2
2.2
2
I_1
2.2
2
VDD
N.V.E.
FUSEBIAS
FUSEBIAS
Date: 10/15/92 3:13:48 PM
Name:
Sheet

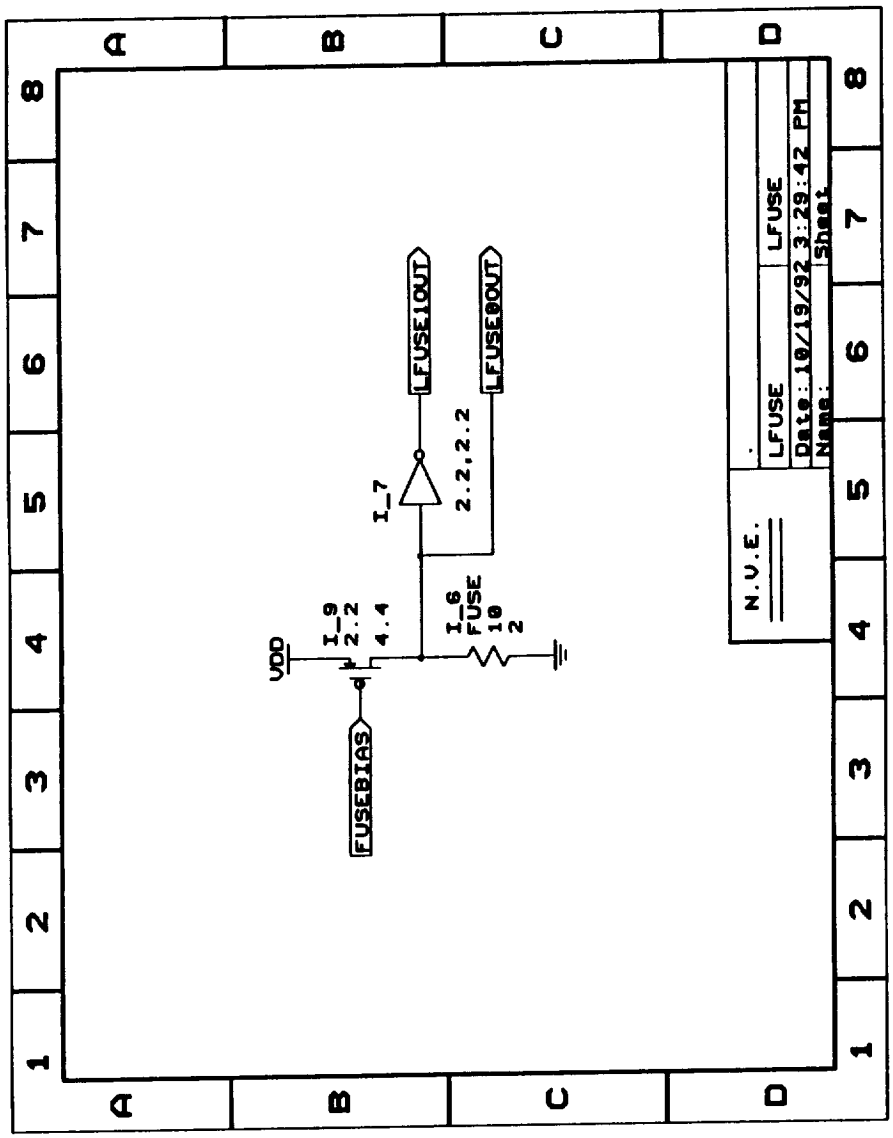

A
B
C
D
1
2
3
4
5
6
7
8
LFUSE1OUT
LFUSE0OUT
I_7
2.2,2.2.2
I_9
2.2
4.4
I_6
FUSE
10
2
VDD
FUSEBIAS
N.U.E.
LFUSE
LFUSE
Date: 10/19/92 3:29:42 PM
Name:
Sheet

TABLE OF ABBREVIATIONS
USED IN THE 1 MEGABIT DESIGN

Update : 10-1-92

ADD[0-17]	Address 0 through 17; latched addresses from the I/O pins; generated by the ALATCH schematic.
BFWD_SLCUR	Bottom Forward Sense Line Current; signal to turn on FWD_SLCUR in top 1K block of a segment.
BLOCK_1K	1024 Bit Memory Array; used in SEGMENT schematics and layout.
BN_FWD_SLCUR	Bottom Not Forward Sense Line Current; signal to turn on N_FWD_SLCUR in top 1K block of a segment.
BN_REV_SLCUR	Bottom Not Reverse Sense Line Current; signal to turn on N_REV_SLCUR in top 1K block of a segment.
BOT_SPARE	Bottom Spare; signal to turn on sense line gate transistors on the segment's bottom 1K block spare sense line.
BREV_SLCUR	Bottom Reverse Sense Line Current; signal to turn on REV_SLCUR in top 1K block of a segment.
BUF_IN[0-255]	Buffer In 0 through 255; input to Gate Line Buffer
BUF_OUT[0-255]	Buffer Out 0 through 255; output from Gate Line Buffer
DECODE0	Decode 0; ADD[0] or NADD[0] input to row decoder.
DECODE1	Decode 1; ADD[1] or NADD[1] input to row decoder.
DECODE2	Decode 2; ADD[2] or NADD[2] input to row decoder.
DECODE3	Decode 3; ADD[3] or NADD[3] input to row decoder.
DECODE4	Decode 4; ADD[4] or NADD[4] input to row decoder.
DECODE5	Decode 5; ADD[5] or NADD[5] input to row decoder.
DECODE6	Decode 6; ADD[6] or NADD[6] input to row decoder.
DECODE7	Decode 7; ADD[7] or NADD[7] input to row decoder.
DRVRDECOD_0	Driver Decode 0; supplies power to the SL_DRVR circuit in segment 0 of BLK_32K, if that segment is selected by the column decoder.
DRVRDECOD_1	Driver Decode 1; same as above for segment 1.
DRVRDECOD_2	Driver Decode 2; same as above for segment 2.
DRVRDECOD_3	Driver Decode 3; same as above for segment 3.
DRVRDECOD_4	Driver Decode 4; same as above for segment 4.
DRVRDECOD_5	Driver Decode 5; same as above for segment 5.
DRVRDECOD_6	Driver Decode 6; same as above for segment 6.
DRVRDECOD_7	Driver Decode 7; same as above for segment 7.
DRVRDECOD_8	Driver Decode 8; same as above for segment 8.
DRVRDECOD_9	Driver Decode 9; same as above for segment 9.
DRVRDECOD_10	Driver Decode 10; same as above for segment 10.
DRVRDECOD_11	Driver Decode 11; same as above for segment 11.
DRVRDECOD_12	Driver Decode 12; same as above for segment 12.
DRVRDECOD_13	Driver Decode 13; same as above for segment 13.

DRVRDECOD_14	Driver Decode 14; same as above for segment 14.
DRVRDECOD_15	Driver Decode 15; same as above for segment 15.
ENABLE_DECOD	Enable Decoder; enables the operation of the precharged NAND gate in the ROW_DEC1 schematic.
ENABLE_SPARE	Enable Spare Sense Line; complement of ENABLE_DECOD; enables the spare sense line selection NAND gates in the ROW_DEC5 schematic.
FWD_SLCUR	Forward Sense Line Current; signal connects drive currents from the SL_DRVR to R_SLDRIV_TOP and L_SLDRIV_TOP.
GATBUF_0	Gate Buffer 0; schematic of first set of buffers for the row decoder; located between the row decoder and the first BLK_32K.
GATBUF_1	Gate Buffer 1; schematic of second set of buffers for the row decoder; located between the first and second BLK_32K.
L_BOT_SPARE	Left Bottom Spare Select; drives the BOT_SPARE lines on the left side of the row decoder (ROW_DEC5).
L_DRVR	Left Sense Line Driver; schematic that contains the p-channel transistors that drive the sense lines on the left half of a segment; contained in the SL_DRVR schematic.
L_SLDRIV_BOT	Left Sense Line Bottom Drive Rail; indicates the sense line bottom drive rail (SLDRIV_BOT) on the left half of a 1K block (BLOCK_1K); used in the SENSLIN2, SENSLIN4, and BLOCK_1K schematics.
L_SLDRIV_TOP	Left Sense Line Top Drive Rail; indicates the sense line top drive rail (SLDRIV_TOP) on the left half of a 1K block (BLOCK_1K); used in the SENSLIN2, SENSLIN4, and BLOCK_1K schematics.
L_TOP_SPARE	Left Top Spare Select; drives the TOP_SPARE lines on the left side of the row decoder (ROW_DEC5).
LEFT_BUFFR	Left Buffer Current; proportional to the left sense line voltage during a read operation; tapped by the preamplifier for sense line signals.
LEFT_DRIV	Left Drive Current; drives the sense line rails on the left side of a segment, and cross couples to the gates of the p-channels in the R_DRVR schematic; generated in the L_DRVR schematic.
N_FWD_SLCUR	Not Forward Sense Line Current; complement of FWD_SLCUR; Pulls R_SLDRIV_TOP and L_SLDRIV_TOP to ground.
N_REV_SLCUR	Not Reverse Sense Line Current; complement of REV_SLCUR; Pulls R_SLDRIV_BOT and L_SLDRIV_BOT to ground.
NADD[0-17]	Not Address 0 through 17; latched complement of addresses 0 through 17 (ADD[0-17]), generated in the ALATCH schematic.
NPRECHARGE	Not Precharge; used to pull the output of a precharged NAND gate high during the precharge cycle; found in many schematics

	throughout the chip.
R_BOT_SPARE	Right Bottom Spare Select; drives the BOT_SPARE lines on the right half of the row decoder (ROW_DEC5).
R_DRVR	Right Sense Line Driver; schematic that contains the p-channel transistors that drive the sense lines on the right half of a segment; contained in the SL_DRVR schematic.
R_SLDRIV_BOT	Right Sense Line Bottom Drive Rail; indicates the sense line bottom drive rail (SLDRIV_BOT) on the right half of a 1K block (BLOCK_1K); used in the SENSLIN2, SENSLIN4, and BLOCK_1K schematics.
R_SLDRIV_TOP	Right Sense Line Top Drive Rail; indicates the sense line top drive rail (SLDRIV_TOP) on the right half of a 1K block (BLOCK_1K); used in the SENSLIN2, SENSLIN4, and BLOCK_1K schematics.
R_TOP_SPARE	Right Top Spare Select; drives the TOP_SPARE lines on the right half of the row decoder (ROW_DEC5).
REV_SLCUR	Reverse Sense Line Current; signal connects current from the SL_DRVR to R_SLDRIV_BOT and L_SLDRIV_BOT.
RIGHT_BUFFR	Right Buffer Current; proportional to the right sense line voltage during a read operation; tapped by the preamplifier for sense line signals.
RIGHT_DRIV	Right Drive Current; drives the sense line rails on the right side of a segment, and cross couples to the gates of the p-channels in the L_DRVR schematic; generated in the R_DRVR schematic.
ROW_DEC1	Row Decoder Level 1; schematic that decodes address lines 0 through 7, and drives a row of sense line gate transistors if selected.
ROW_DEC2	Row Decoder Level 2; schematic that decodes address lines 0 through 7, and drives four rows of sense line gate transistors if selected; calls 4 ROW_DEC1 schematics.
ROW_DEC3	Row Decoder Level 3; schematic that decodes address lines 0 through 7, and drives 16 rows of sense line gate transistors if selected; calls 4 ROW_DEC2 schematics.
ROW_DEC4	Row Decoder Level 4; schematic that decodes address lines 0 through 7, and drives 64 rows of sense line gate transistors if selected; calls 4 ROW_DEC3 schematics.
ROW_DEC5	Top Level Row Decoder; schematic that decodes address lines 0 through 7, and drives 256 rows of sense line gate transistors if selected; calls 4 ROW_DEC4 schematics.
ROW_L[0-255]	Left Row Selector Lines 0 to 255; bus turns on sense line gate transistors on the left half of the row decoder; found in ROW_DEC5 schematic.
ROW_L_0[0-255]	Left Row Selector Lines 0 to 255, after they have passed through

	GATBUF_0.
ROW_L_1[0-255]	Left Row Selector Lines 0 to 255, after they have passed through GATBUF_1.
ROW_R[0-255]	Right Row Selector Lines 0 to 255; bus turns on sense line gate transistors on the right half of the row decoder; found in ROW_DEC5 schematic.
ROW_R_0[0-255]	Right Row Selector Lines 0 to 255, after they have passed through GATBUF_0.
ROW_R_1[0-255]	Right Row Selector Lines 0 to 255, after they have passed through GATBUF_1.
SEGMENT	Segment; schematic which contains 2 BLOCK_1Ks, 2 SL_IMUXs, and 1 SL_DRVR circuit; replicated 512 times to create a 1 Megabit memory array.
SENSELIN	Sense Line; used in schematics and layout for an 8 MRAM bit, or 4 dual redundant logical bit, sense line.
SENSLIN2	Block of Two Sense Lines; used in schematics.
SENSLIN4	Block of Four Sense Lines; used in schematics.
SL_DRVR	Sense Line Driver; schematic supplies sense line current for a segment, and provides taps for use by the preamplifier during a read operation.
SL_GATE	Sense Line Gate; turns on gate transistors in the sense line; used in the SENSELIN schematic.
SL_GATE_0	Sense Line Gate 0; turns on the gate transistors in the top SENSLIN2 block inside the SENSLIN4 block.
SL_GATE_1	Sense Line Gate 1; turns on the gate transistors in the bottom SENSLIN2 block inside the SENSLIN4 block.
SL_GATE[0-127]	Sense Line Gate 0 through 127; bus of the 128 signals which turn on the sense line gate transistors in a 1K block (BLOCK_1K).
SL_GATE[0-255]	Sense Line Gate 0 through 255; bus of the 255 signals which turn on the sense line gate transistors in a SEGMENT.
SL_IMUX	Sense Line Current Mux; schematic; steers the current from the sense line driver to the left and right halves of a 1K block, in either the forward or reverse sense current direction.
SLCT_DRIVR	Select Driver; supplies power to the transistors in the L_DRVR and R_DRVR schematics; this power supply is steered to one of 128 columns of 4 segments through the column decoder.
SLDRIV_BOT	Sense Line Bottom Drive Rail; connected to the drains of the gate transistors in the SENSELIN schematic; while sense line current is in the forward direction, it comes out of this rail, and when sense line current is in the reverse direction, it goes into this rail.
SLDRIV_TOP	Sense Line Top Drive Rail; connected to the first MRAM bit in the sense line string in the SENSELIN schematic; when sense line current is in the forward direction, it goes into this rail,

	and when sense line current is in the reverse direction, it comes out of this rail.
TFWD_SLCUR	Top Forward Sense Line Current; signal to turn on FWD_SLCUR in top 1K block of a segment.
TN_FWD_SLCUR	Top Not Forward Sense Line Current; signal to turn on N_FWD_SLCUR in top 1K block of a segment.
TN_REV_SLCUR	Top Not Reverse Sense Line Current; signal to turn on N_REV_SLCUR in top 1K block of a segment.
TOP_SPARE	Top Spare; signal to turn on sense line gate transistors on the segment's top 1K block spare sense line.
TREV_SLCUR	Top Reverse Sense Line Current; signal to turn on REV_SLCUR in top 1K block of a segment.
TRIM_1XXX	Bit 3 Trim Signal; controls the operation of 8 p-channel transistors in the L_DRVR and R_DRVR schematics for sense line current control.
TRIM_X1XX	Bit 2 Trim Signal; controls the operation of 4 p-channel transistors in the L_DRVR and R_DRVR schematics for sense line current control.
TRIM_XX1X	Bit 2 Trim Signal; controls the operation of 2 p-channel transistors in the L_DRVR and R_DRVR schematics for sense line current control.
TRIM_XXX1	Bit 2 Trim Signal; controls the operation of 1 p-channel transistor in the L_DRVR and R_DRVR schematics for sense line current control.

NVE NV441048

MRAM 256k x 4 MRAM

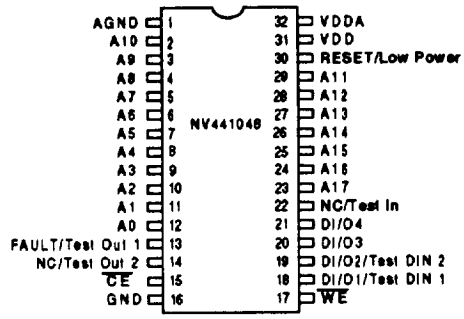
FEATURES

- High speed write: 80ns
- High-performance, low power, CMOS double-metal process
- Single +5V ±10% power supply
- Simple /CE operation
- All inputs and outputs are TTL compatible
- Unlimited read - write operational life
- Unlimited data retention with loss of power
- Data protected during loss of power

OPTIONS	MARKING
• Timing	
250ns access	-250
• Packages	
Plastic DIP (300 mil)	None
Plastic JLCC	JC
• Temperature	
Commercial (0° to +70°C)	None

PIN ASSIGNMENTS (Top View)

32-Pin DIP



GENERAL DESCRIPTION

The NVE MRAM family employs high-speed, low-power CMOS designs using a Permalloy memory element featuring high density and nonvolatile data storage. NVE MRAMs are fabricated using double-layer metal, double-layer polysilicon technology. For flexibility in high-speed memory applications, NVE offers chip enable (/CE) capability which places the outputs in a High-Z state when not selected.

Writing to this device is accomplished when write enable (/WE) and chip enable (/CE) are both low. The falling edge of either /WE or /CE will latch the address and initiate the cycle. During the cycle, all address lines must remain unchanged along with /CE. Reading is accomplished when /WE remains high while /CE goes low. Again address and /CE must remain stable during the cycle.

The device offers a reduced power standby mode when the RESET/Low Power input is brought to high level. In this mode all circuitry is disabled and the chip cannot be accessed. The chip is re-enabled by bringing the RESET/Low Power input back to a low level. This will trigger the internal Power On Reset circuitry, and the chip will generate a FAULT signal during the power on time. At the end of this time the chip can be accessed normally.

Data is protected during the loss of power by internal lock out circuitry which will prevent read or write operations from occurring when the supply voltage has fallen below 4.5 volts. The FAULT signal will go low and remain in this state until the voltage exceeds 4.5 volts and the power on reset time has elapsed.

ABSOLUTE MAXIMUM RATINGS

Voltage on Vdd Supply Relative to Vss ....-1V to +7V
Operating Temperature ..... 0° to 70°C
Storage Temperature (Plastic) .....-55°C to +150°C
Storage Temperature (Ceramic) .....-65°C to +150°C
Short Circuit Output Current..... 50mA
Power Dissipation ..... 1 W
Static Discharge Voltage ..... >2001V
(Per MIL-STD-883 Method 3015.2)
Latch-up Current ..... >200mA

*Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

ELECTRICAL CHARACTERISTICS AND DC OPERATING CONDITIONS

(0°C ≤ T_A ≤ 70°C; Vdd = 5V ± 10%)

DESCRIPTION	CONDITIONS	SYMBOL	MIN	MAX	UNITS	NOTES
Output High Voltage	Vdd (5V) = Min Ioh = 4.0mA	Voh	2.4		V	1
Output Low Voltage	Vdd (5V) = Min Iol = 8 mA	Vol		0.4	V	1
Output Leakage Current	/CS = Voh 0V ≤ Vout ≤ Vdd	Iol	-1.0	+1.0	µA	
Input High Voltage		Vih	2.2	Vdd + 1	V	1
Input Low Voltage		Vil	-.05	0.8	V	1, 2
Input Leakage Current	0V ≤ Vout ≤ Vdd	Iil	-1.0	+1.0	µA	
Operating Current	/CE ≤ Vil; Vdd = Max f = MAX = 1/t_{RC} Outputs Open	Idd	70	100	mA	3, 13
Standby Current	/CE ≥ Vih; Vdd = MAX f = MAX = 1/t_{RC} Outputs Open	Isb	5	14	mA	
Low Power Mode Current	/CE ≥ Vih; Vdd = MAX f = 0; Low Power ≥ Vih	Ilp		10	µA	
Read\Write Protect Voltage	Vdd = 5.0 Volts	Vpv	4.25	5.70	V	13

CAPACITANCE

DESCRIPTION	CONDITIONS	SYMBOL	MAX	UNITS	NOTES
Input Capacitance	T_A = 25°C; f = 1MHZ; Vdd = 5V	Ci	8	pF	4
Output Capacitance	T_A = 25°C; f = 1MHZ; Vdd = 5V	Co	8	pF	4

ELECTRICAL CHARACTERISTICS AND AC OPERATING CONDITIONS

(0°C ≤ T_A ≤ 70°C; Vdd = 5V ± 10%)

DESCRIPTION	SYMBOL	MIN	MAX	UNITS	NOTES
READ Cycle					
Read Cycle Time	tRC	250		ns	11
Read Data Access Time	tAC	225		ns	
Read Chip Enable Cycle Time	tRCE	250		ns	
Chip Enable (Low) to Output in Low-Z	tLZCE		5	ns	6,7
Chip Disable (High) to Output in High-Z	tHZCE		25	ns	6,7
WRITE Cycle					
Write Cycle Time	tWC	100		ns	
Write Chip Enable Cycle Time	tWCE	80		ns	
Address Setup Time	tAS	10		ns	
Address Hold Time	tAH	20		ns	
Chip Precharge Time	tP	50		ns	

POWER-DOWN/POWER-UP TIMING

DESCRIPTION	CONDITIONS	SYMBOL	MIN	MAX	UNITS	NOTES
Vdd slew from 0 to within specification	T_A = 25°C	tR	0		µs	1
Vdd slew form in specification to 0	T_A = 25°C	tF	300		µs	1
Power Up Recovery Time	T_A = 25°C	tRC	0.65	2.2	ms	1

AC TEST CONDITIONS

Input pulse levels	Vdd to 3.0V
Input rise and fall times	5ns
Input timing reference levels	1.5V
Output reference levels	1.5V
Output load	See Figures 1 and 2

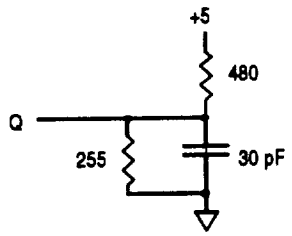


Fig. 1 OUTPUT LOAD EQUIVALENT

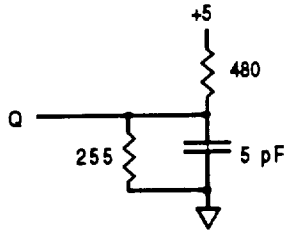
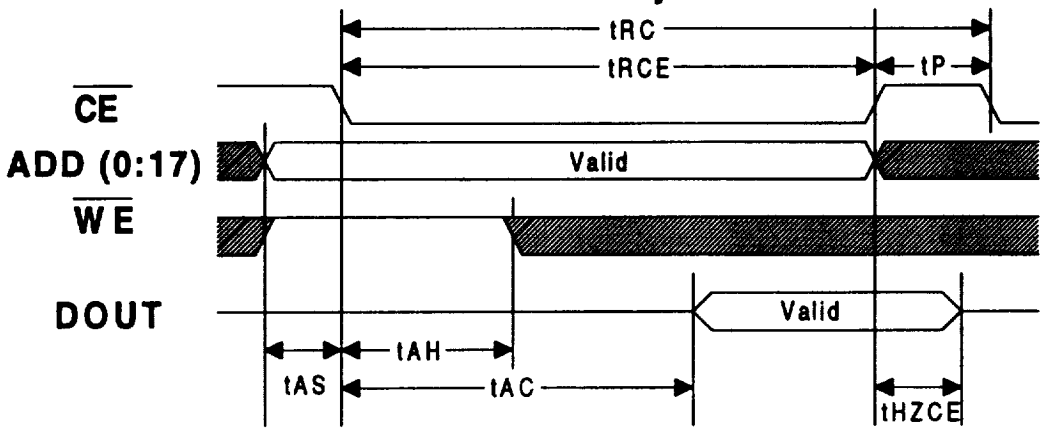


Fig. 2 OUTPUT LOAD EQUIVALENT

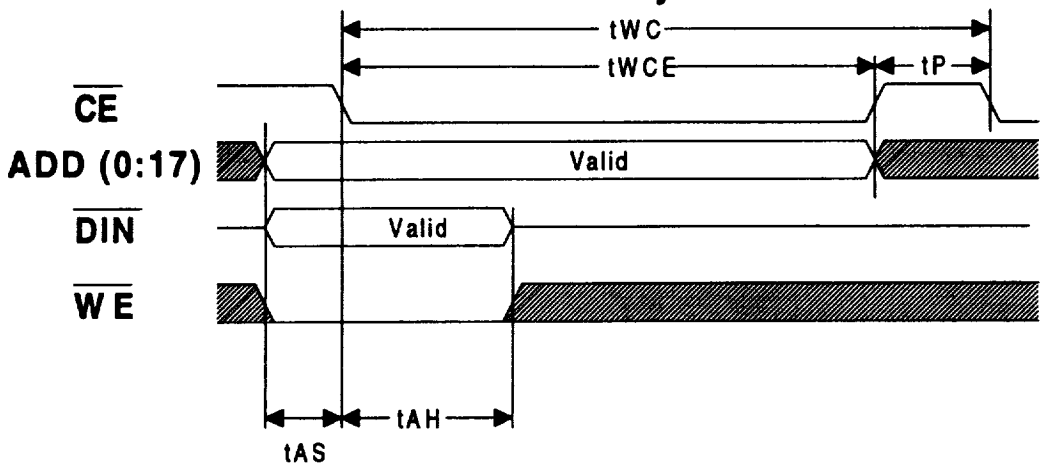
NOTES

1. All voltages referenced to Vdd (GND).
2. -3V for pulse width < 20ns.
3. Icc is dependent on output loading and cycle rates.
4. This parameter is sampled
5. Test conditions as specified with the output loading as shown in Fig. 1 unless otherwise noted.
6. t_{HZCE}, t_{HZOE}, and t_{HZWE} are specified with CL = 5 pf as in Fig. 2. Transition is measured ±500mV from steady state voltage.
7. At any given temperature and voltage condition, t_{HZCE} is less than t_{LZCE} and t_{HZWE} is less than t_{LZCE}.
8. /WE is HIGH for READ cycle.
9. Device is continuously selected. Chip enable is held in it's active state.
10. Address valid prior to or coincident with latest occurring chip enable.
11. t_{RC} = Read Cycle Time.
12. Chip enable (/CE) and write enable (/WE) can initiate but not terminate a write cycle.
13. Typical values are measured at 5V, 25°C and 350ns cycle time.

Read Cycle



Write Cycle



This page intentionally blank

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
---		

4. Title and Subtitle

ULTRA- DENSE MAGNETORESISTIVE MASS MEMORY

5. Report Date

10-30-92

6. Performing Organization Code

7. Author(s)

J. M. Daughton, R. Sinclair, T. Dupuis, J. Brown

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

Nonvolatile Electronics Inc.
12800 Industrial Park Boulevard
Suite 110
Plymouth, MN 55441-3928

11. Contract or Grant No.

NASA7-1152

12. Sponsoring Agency Name and Address

NASA Resident Office - JPL
4800 Oak Grove Drive
Pasadena. CA 91109

13. Type of Report and Period Covered

Fifth Quarter Report
July 15, 1992 to October 15, 1992

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Two stand alone test systems were constructed to test the reliability and determine long term degradation of the 16k macrocells which will be used in the wafer scale demonstration unit. . These testers are programmable and capable of operating without a host computer and can recover in the event of a power loss. A test chip was designed which incorporated the MRAM bit required by the 1 Meg design to characterize the electrical performance and resolve processing issues affecting the layout and design thus providing more assurance of success. A model for the parasitic loading of unused sense lines contained in the 1 Meg MRAM was also developed and will be used to accurately simulate it's operation. This will allow the accurate simulation and measurement of drive and sense currents. The timing of the 1 Meg MRAM chip was finalized and a product specification was generated defining the electrical and mechanical characteristics. The design of the array drivers, decoders, sense preamplifier, and buffers along with the sense and word lines has been completed and laid out. The core of the chip is two thirds complete and LVS checks have been initiated.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Mass Memory, Wafer Scale Integration Magnetoresistive	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
None	None	118	$492,143

NASA FORM 1626 OCT 86