100BASE-X Repeater Management Module

National Semiconductor
Application Note AN-1106
Mike Heilbron
Brad Kennedy
Bill Lee
Steve Rees
May 1998

Overview

This document describes an example implementation of a management module for a 100BASE-X Ethernet repeater. It assumes a basic familiarity with the DP83850, DP83856 and NS486 devices. Their data sheets can be found on National Semiconductor's web site at www.national.com. This note is not intended to be used as specification for a production product. National recommends that all aspects of this design be reviewed before implementation.

1.0 Introduction

The DP83856 100RIB, in conjunction with the DP83850 100RIC™, can be developed into a system that provides Ethernet statistics on a per-port basis. In order for such a system to respond to commands that are sent over the network, an Ethernet MAC, microprocessor, and driver software must also be implemented into the design, otherwise, statistic information can be retrieved through a serial port connection with minimal software overhead.

The 100RIB can also be used with the DP83858 100RIC8. Throughout this note, any reference to the 100RIC can be thought of as applying to the 100RIC8 as well.

Hardware Features

- Utilizes the DP83856 to record traffic statistics.
- Uses the NS486SXL for managing the statistics and responding to requests.
- Able to transfer the statistics via the network, or through a serial port, without sacrificing an Ethernet port.
- Monitors up to 16 100RICs (maximum of 192 ports).

2.0 Why Have Management?

The cost of a network includes more than just the cost of the network equipment itself, maintenance of the network is also a cost that must be considered. Network management is a useful way to reduce the cost of ownership of a network by giving the network administrator the tools and information needed to keep the network running at its best. This is especially important as networks become more complex and geographically dispersed.

The 100RIC/100RIB combination can give a network manager traffic information that will reveal the network's performance. Statistics such as packet throughput and collision and erred packet rate on a per-port basis can be helpful in predicting problems in a network. By placing one management module using the 100RIB in a stack of up to 16 100RICs, the network manager can monitor the traffic on up to 192 nodes.

3.0 General Design Overview

The management module was originally designed as an evaluation platform to be connected to a 100RIC based repeater evaluation platform. The two evaluation boards were connected using ribbon cables. An alternative design could place the management module on the same board as the repeater functions.

Because the module was intended to be an evaluation platform, test points and other features have been added to this design that would not normally be included in a production design. Also, since component layout did not need to be efficient, clock buffers and other signal enhancing was used. Although the emphasis on this design was

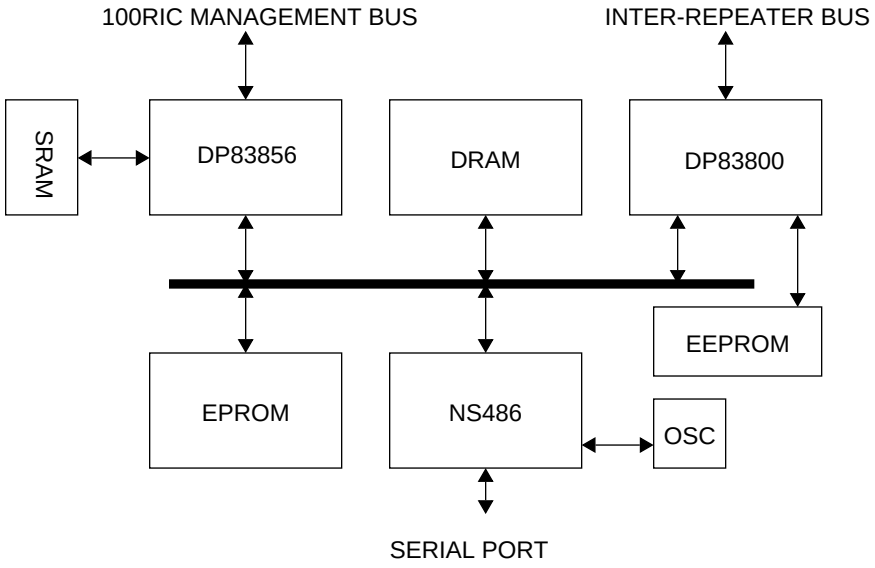


Figure 1. System Block Diagram

device evaluation, it is a good indication of the complexity of a production design. National highly recommends that all aspects of this design be reviewed when considering its use for production.

4.0 DP83856 100RIB

The DP83856 100RIB is a Fast Ethernet repeater information base designed to meet network management requirements. It is used in a system that monitors all of the Ethernet traffic that flows through the repeater stack to which it is attached and keeps track of all of the statistics called out in clause 30.4 in the IEEE 802.3u specification. Most of the statistics are monitored by the 100RIB, others are monitored in the 100RIC and physical layer, and a small set of statistics are calculated in software.

Using the 100RIB reduces the need of a powerful microprocessor and large amounts of software. The statistics are gathered though hardware and stored in dedicated SRAM. This information can then be read through the 100RIC's ISA-like interface. Only one 100RIB is needed for up to 16 100RICs which keeps the total system cost down and simplifies the overall design of a managed repeater.

The statistical information held in the 100RICs and physical layers can be efficiently retrieved by the 100RIB. A single command from the microprocessor will cause the 100RIB to read all of the counters in the 100RIC, and then interrupt the processor when finished. Simple commands will retrieve physical layer information as well.

4.1 Management Bus

The module was designed to be connected to a repeater through ribbon cables. The distance between the boards was kept short, and signal noise was not a concern. Refer to the schematics at the end of this document for more details.

A 74ABT244 buffer was used to strengthen the received signal from the transmit data bus on the repeater platform, 13 devices and the ribbon connector on the repeater share this bus. These signals were not buffered prior to being transmitted over the ribbon cable. If the module were an integral part of the repeater design, this buffer would not be needed.

No buffer was used to strengthen the management data bus. This bus is buffered by a 74ABT16245 on each repeater in the stack. Management signals from each repeater were found to be strong enough to not warrant the buffering recommended in the DP83850 datasheet. This decision is implementation dependent, and every designer should analyze the situation before choosing to leave the buffer out.

A buffer is required for RDIO, which is a bidirectional signal. This design followed the recommendations from the 100RIC data sheet which has two 74ABT125 buffers connected in parallel as a transceiver using RRDIR to control the direction. The "RIB_CNTL" GAL16V8 is used to invert the RRDIR signal to one of the buffers.

The GAL is also used to modify the /M_DV input to the 100RIB. The 100RIB data sheet in the Systems Consideration section explains that noise on the /M_DV line may cause the 100RIB to record improper data. Gating the signal with TX_RDY solves this problem. Refer to the 100RIB data sheet for a thorough explanation and Section 7.1 of this note for the logic equations.

4.2 SRAM

All of the information gathered by the 100RIB is stored in SRAM. Refer to the schematic for illustration. SRAM used in this design must have a read access time of 20 ns or faster.

4.3 ISA Bus

The address and data bus from the 100RIB, as well as most of the control signals, can be connected directly to the buffered ISA bus from the microcontroller. The two control signals that require conditioning are the 100RIB's chip select and read/write signals. The NS486 asserts its read and write commands after it asserts its chip selects. The 100RIB, however, requires the assertion of its chip select to occur no sooner than the assertion its read/write line.

The "RIB_CNTL" GAL is also used to modify the signals generated by the NS486. Refer to the schematic and Section 7.1 for more information on the wiring connection and logic equations.

Figures 2 and 3 show the signal relationship between the commands and the data. The following table defines the signals.

Table 1. 100RIB Signal Definition

Address	Address bus from the CPU.
/IOW	I/O write signal from the CPU.
/IOR	I/O read signal from the CPU.
GR/W	Gated read/write signal from the GAL to the 100RIB
/RIB_CS	Chip select from the CPU.
/GRIB_CS	Gated chip select from the GAL to the 100RIB
/CRDY	CPU ready, not used in this design but shown for reference only.

For register writes, the NS486 was programmed with 4 wait states (refer to Table 2 for other programming suggestions). For register reads, 5 wait states were used. Programmed wait states were used to end the cycle instead of CRDY to simplify the design. The GAL chosen for this design had a propagation delay of 10 ns.

5.0 NS486SXL Microprocessor

The NS486SXL is a highly integrated embedded system controller incorporating an Intel486^{TM}-class 32-bit processor along with other features that make it ideal for this application. Its ISA-like bus connects well to the DP83800 Ethernet MAC and the 100RIB. It also has an on-board UART and DRAM controller. A block diagram of the microcontroller portion of the design is shown in Figure 4.

The following is the initialization sequence used after the board is first powered up. Refer to Section 8 and the NS486SXL data sheet for more details. Table 2 outlines the chip select and wait state information programmed during initialization.

Set the NS486 to 5V I/O mode

Enable the NS486 Cache

Set up the NS486 Bus Interface to enable the internal UART

Set up external Chip Selects

Set up Chip Select and wait state timings

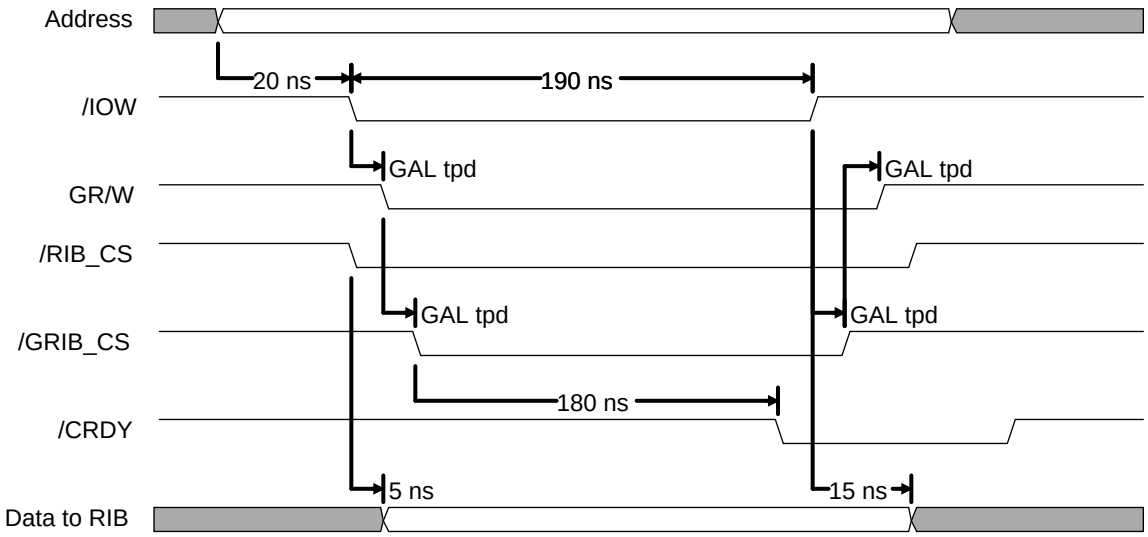


Figure 2. 100RIB Register Write

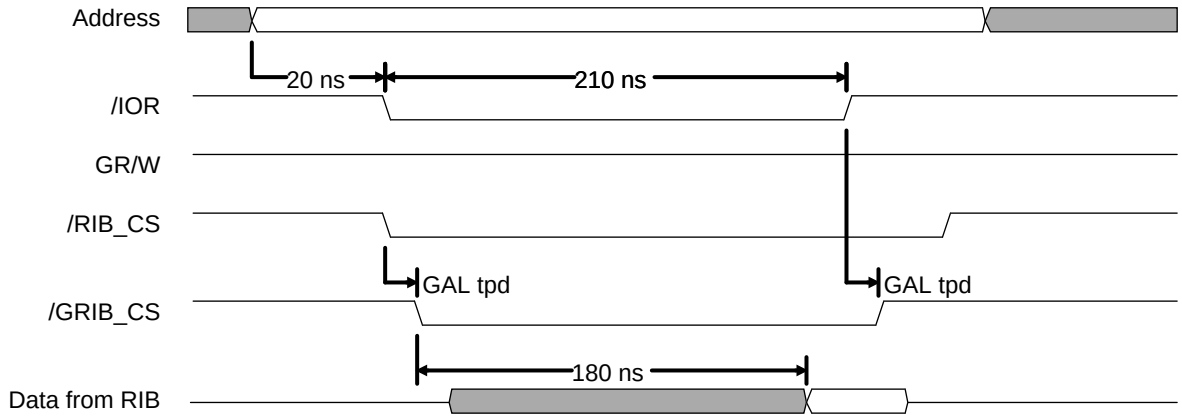


Figure 3. 100RIB Register Read

Map the logical chip selects to the physical signals

Initialize DRAM

Initialize UART

Set up the Global Descriptor Table

Set up Code Segments

Set up the stack

A reference clock of 50.0 MHz is supplied to the NS486 by an oscillator. The NS486 divides this by 2 to provide a 25 MHz system clock to the MAC portion of the design. This clock is buffered by a 74CT2525 because of the long trace lengths it travels to get to the MAC. Inclusion of the buffer is implementation dependent.

The reset signal supplied to the NS486 (into PWGOOD) comes from the reset circuitry on the repeater board. When the module was laid out, this signal had to travel over 12 inches (including the ribbon cable). A RC termination to ground helped to sharpen the rise time of this signal.

5.1 ISA Bus

The data sheet for the NS486SXL recommends buffering the ISA address and data bus. As recommended, the reference design placed the buffering after the dynamic RAM. The NS486SXL is the only bus master in this system, and therefore the address bus needs only to be active in one direction. The data bus, however, needs to be bidirectional. By placing the buffer for the data bus between the EPROM and all of the I/O devices, the direction of the buffer will only be dependent on I/O device selections. Fairchild Semiconductor's 74F245 bidirectional buffers were used in this design for the data bus; 74F244 line drivers were used for the address bus.

5.2 Memory

The EPROM contains all of the operational code for the NS486. Fairchild's NM27P210, a 64K x 16 EPROM, was used for this design. The amount of space required to hold the software is implementation dependent.

DRAM SIMMs were used to speed up the implementation of this design. In addition to allocating space for all of the statistics, additional space is required to hold management

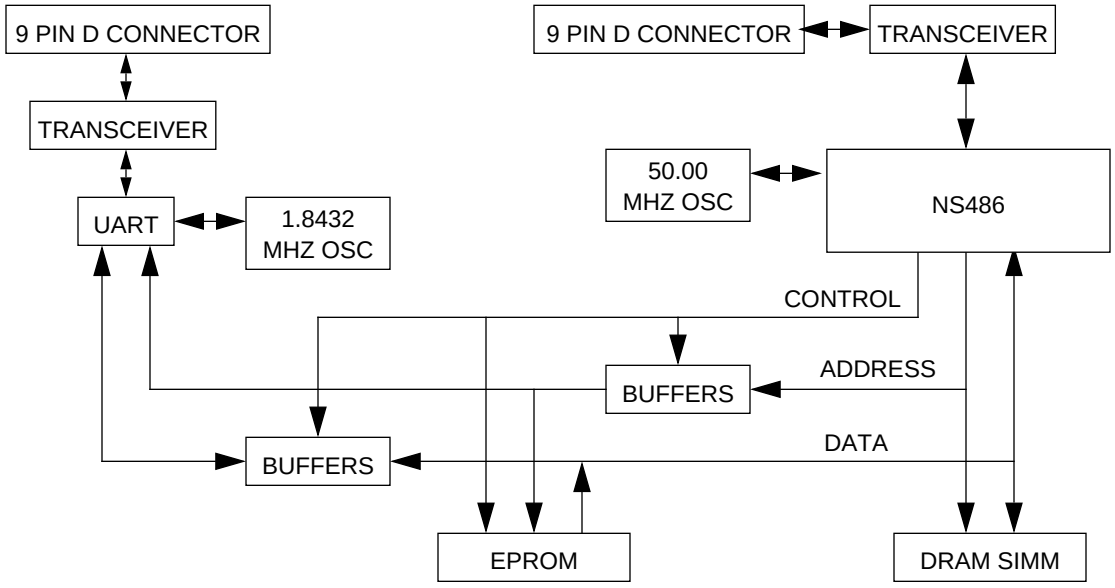


Figure 4. NS486 Circuit Block Diagram

requests received through the MAC and the formation of the responses. The NS486 has an on-board DRAM controller that was connected to the DRAM as recommended by the NS486SXL data sheet.

5.3 Internal and External UART

The CPU's internal UART may be used to gather statistics from the management module. A serial port from another computer may be connected to this port, and communication software will have to be written. In this way, statistics may be read locally without having to rely on the Ethernet MAC circuitry.

An external UART is used in this design to verify the code in the EPROM. National's PC16550D was used for this design. Many debuggers allow for monitoring the code as it runs. A second UART is useful when trying to debug the transfer of information through the CPU's internal UART. Once the code is fully operational, this UART is not needed.

6.0 DP83800 10/100 Mb/s MAC

The DP83800 is a 10/100 Mb/s Ethernet MAC designed specifically for the ISA bus. It is designed for programmed I/O operation that provides a simple host interface. In this design, programmable logic provides the glue between the MAC's MII and the 100RIC's inter-repeater bus.

In order to get the statistics from the management module over the network, a MAC is required to receive management commands (e.g. SNMP) and transmit the information. The simplest implementation would be to connect the MAC's MII to one of the repeater ports. To avoid using up a port, the MAC was designed to transmit packets over the inter-repeater bus and receive packets from the shared MII transmit bus from the local 100RIC. Packet information is stored in the DRAM.

6.1 Inter-repeater Bus

GAL22V10s are used to translate the MII signals from the MAC to signals emulating another repeater on the inter-repeater (IR) bus. In addition to transmit data, the logic also sends out a repeater ID and management data. The logic equations can be found in Section 7.0. Note that, just like any other repeater added to the stack, the management module repeater ID must be unique.

Table 2. 100RIB Signal Definition

Device	Type	Chip Select	Interrupt	Number of Wait States
100RIB	I/O	CS1	IRQ1	Write: 4 wait states Read: 3 wait states
DP83800	I/O	N/A	IRQ3	1 program command delay, 3 wait states
Internal UART	I/O	N/A	N/A	N/A
External UART	I/O	CS4	IRQ4	1 program command delay, Write: 2 wait states Read: 3 wait states
EPROM	MEM	CS0	N/A	N/A
DRAM	DRAM	N/A	N/A	N/A

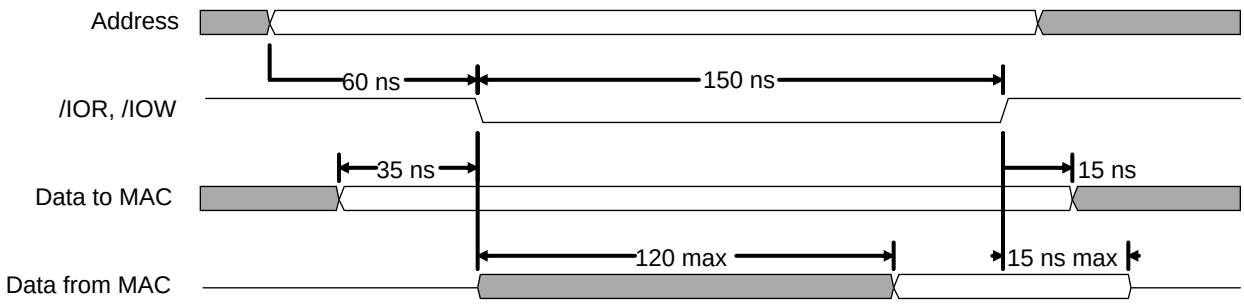


Figure 5. MAC Register Read and Write

The GAL labeled “IRV” mimics the 100RIC’s forcing and sensing of the IR vector. It has as its inputs the module’s ID and the IR vector bus. At the start of a MAC transmission this GAL will force the MAC’s vector ID onto the bus. If the vector sensed on the bus equals this ID, the GAL will assert the “vector equal” (/VECT_EQ) signal.

The GAL labeled “MDATA” mimics the 100RIC’s generation of the management data onto the inter-repeater bus. At the fourth clock cycle after the assertion of the MAC’s TXEN the lower 4 bits of the MAC’s vector ID are forced onto the IR management bus. On the fifth cycle the most significant bit is forced onto the bus. On the sixth, a port ID of 0h is forced. During this 3-cycle activity, /M_DV is asserted. MDATA also generates a 3-bit counter to keep track of the clock cycles. The most significant bit and a 1-clock-delayed version are fed to the “CONTROL” GAL.

The GAL labeled “CONTROL” generates all of the control signals to the IR bus and the collision signal to the MAC. COL is asserted during an IR bus collision or a vector ID mismatch during transmission. Transmit buffers are enabled during the time the MAC’s TXEN is asserted (and for 4 cycles after TXEN deasserts) allowing the MAC to drive the IR bus.

6.2 ISA Bus

The DP83800’s address and data busses are connected to the buffered busses from the NS486. The control signals do not need to be buffered. As noted in Table 2, the NS486 is programmed to provide 1 programmed command delay and 3 wait states when accessing the MAC’s registers. Figure 5 shows the signal timing relationships.

6.3 Serial EEPROM

The Ethernet address for the MAC is stored in a serial EEPROM connected directly to the MAC. Fairchild’s NM93C46LZ, a 1024-bit serial MICROWIRE™ EEPROM, was used in this design. Refer to the schematic for illustration.

7.0 PAL Equations

This section lists sample code for the four GAL devices used in the management module design.

7.1 RIB_CNTL

```
{This device will modify the R/W and /CS signals from the CPU to ensure that the assertion of /CS is after the assertion of the /Write pulse, and the deassertion occurs before the deassertion of the /Write pulse. It also ensures that /M_DV is asserted only when TX_RDY is also asserted. This avoids the possibility of not counting a packet after random invalid management bus activity (please see Section 6.4 of the DP83856 datasheet for more details on this issue.)


Inputs:
         /rib_cs......Chip select line from the CPU
         /iow.........I/O Write line from the CPU
         /ior.........I/O Read line from the CPU
         rrdir.........Serial MII register interface direction signal from the 100RIB
         tx_rdy.......Transmit Data Ready signal from the 100RIC
         m_dv_.........Management Data Valid signal from the 100RIC

Outputs:
         gr_w.........read/write strobe connected to the 100RIB. Read is active high
         /gcs.........Chip select connected to the 100RIB
         /b_rrdir.....inverted rrdir signal for rdio buffering
         m_dv_mod_.....TX_RDY gated /M_DV signal
}



CHIP rib_cntl GAL16V8
 /rib_cs=2 /iow=3 /ior=4 m_dv_=6 tx_rdy=7
 rrdir=9 gr_w=12 /gcs=13 m_dv_mod_=18
 /b_rrdir=19

EQUATIONS
b_rrdir = rrdir
m_dv_mod_ = m_dv_ + /tx_rdy
gr_w = /((iow * rib_cs) + (/gr_w * gcs))
gcs = (rib_cs * ior) + (iow * /gr_w)
```

7.0 PAL Equations (Continued)

7.2 MDATA

```
{This module is used to generate inter-RIC™ management bus signals for the generic 100M
MAC. The management bus signals are similar to those generated by the DP83850 and consist
of a framing signal (/M_DV) and nibble wide data output on MD0 through MD3. Three nibbles
(the MAC's IR_BUS Vector ID (MSB and LSB) and port number) are generated for each
packet.)

Note that this GAL needs to be a 22V10-15L. The speed rating is important for IR bus tim-
ing.


Inputs:
         LC.............Local clock (25Mhz).
         TXEN...........Transmit enable from the MAC.
         DEV_ID<4:0>....Unique ID for IR-Vector arbitration.

Outputs:
         MD<3:0>........Mgmt data to mgmt bus.
         M_DV_ .........Active low mgmt data valid.
         out<3:0>.......Counter outputs (out2 and out3 used by CONTROL GAL)
}

CHIP  mdata GAL22V10
LC=1 TXEN=3 DEV_ID0=6 DEV_ID1=7 DEV_ID2=8
DEV_ID3=9 DEV_ID4=10 MD0=15 MD1=16 MD2=17
MD3=18 M_DV_=19 out3=23
out0=20 out1=21 out2=22


EQUATIONS
{Generate the counts and decodes that will be used generate Management Bus signals. The
out(2:0) signals create a counter that counts from 0h to 7h when the TXEN line goes
active. Once 7h has been reached the out(2:0) signals remain at 7h until TXEN is deas-
serted, at which time they step down to 0h and remain there until the next TXEN asser-
tion. out3 is an extension of out2.}


out0 := (/out0 + (out0 * out1 * out2)) * TXEN
out1 := (out0 * /out1 + /out0 * out1 + out0 * out1 * out2) * TXEN + out0 * /TXEN
out2 := (out0 * out1 * /out2 + out2 * /out1 + out2 * /out0 + out0 * out1 * out2) * TXEN
         + out1 * /TXEN
out3 := out2


{Generation of Management Bus (M_Bus) signals; When the count generated by the out(2:0)
signals reaches 4, the lower RID bits are driven onto the M_Bus. At a count of 5, the
upper RID bit is driven onto the M_Bus. Finally at a count of 6, the port ID (which in
this design is hard-coded to 0h) is driven out onto the M_Bus. The /M_DV signal is
asserted while these three nibbles are output (i.e. during the 4th, 5th, and 6th
counts).}


MD0 := (TXEN * DEV_ID0 * /out0 * /out1 * out2) + (TXEN * DEV_ID4 * out0 * /out1 * out2)
MD1 := TXEN * DEV_ID1 * /out0 * /out1 * out2
MD2 := TXEN * DEV_ID2 * /out0 * /out1 * out2
MD3 := TXEN * DEV_ID3 * /out0 * /out1 * out2
M_DV_ := (/TXEN + /out2) + (out0 * out1)
```

7.0 PAL Equations (Continued)

7.3 CONTROL

```
{This module is used to generate the IR_ACTIVE and IR_COL_OUT signals for the MAC on the
interRIC bus. It also controls the transmit buffers to the interRIC and management buses
and placed the MAC's signals to TRI-STATE mode to avoid contention when it is not in con-
trol of these buses. Additionally, the MAC is notified of collisions through the COL sig-
nal.)


Inputs:
         TXEN............Transmit enable from the MAC.
         VECT_EQ_........Vector equal signal from IRV_PLA.
         IR_COL_IN_......Collision signal from the IR bus.
         RESET...........Active high reset signal
         out[3:2]........Outputs from the state machine in MDATA.

Outputs:
         COL.............Collision to the MAC.
         TXD_EN_.........Transmit enable - for Transmit buffers to IR/Mgmt bus.
         IR_COL_OUT_.....Collision signal to IR bus.
         IR_ACTIVE_EN_...Asserted low during transmission.
}

CHIP  control GAL16V8
 RESET=2 TXEN=3 IR_COL_IN_=4 VECT_EQ_=5
 out2=6 out3=7 TXD_EN_=14 IR_COL_OUT_=15
 COL=16 IR_ACTIVE_EN_=17


EQUATIONS
{Transmit buffers are enabled during the time TXEN is asserted (and for 4 cycles after
TXEN deasserts) if COL and RESET are both deasserted.}


TXD_EN_ = COL + (/TXEN * /out3) + RESET
IR_ACTIVE_EN_ = /TXEN


{Collision is asserted during a transmission if the management module receives a colli-
sion notification from any of the other 100RICs in the stack (via /IR_COL_IN_) or if the
vector on the interRIC bus does not equal the DEV ID. Additionally, the management module
notifies other 100RICs on the stack of collisions it detects by asserting the
/IR_COL_OUT_ signal.}


IR_COL_OUT_ = /TXEN + /VECT_EQ_ + /out2
COL = /IR_COL_IN_ + (TXEN * VECT_EQ_ * out2)
```

7.0 PAL Equations (Continued)

7.4 IRV

```
{This module is used to interface a generic 100 Mb/s MAC to the Inter-Repeater and Man-
agement buses). It generates IR_Vectors for the 100 Mb/s MAC and provides functionality
to compare the MAC's IR_Vectors to the vectors on the interRIC bus (used in collision
detection). The module also generates carrier sense for the MAC.


Inputs:
         TXEN.............Transmit enable from the MAC.
         DEV_ID<4:0>......MAC's Unique ID for IR-Vector arbitration.
         IRV_IN<4:0>......Vector inputs from IR bus.
         IR_ACTIVE_IN_....Network activity notification from IR bus.

Outputs:
         IRVOUT<4:0>......Vector ID enables for driving IR bus.
         VECT_EQ_.........IR bus vectors & MAC DEV_ID equal signal.
         CRS..............Carrier sense to the MAC.
}

CHIP  irv GAL22V10
 DEV_ID4=1 DEV_ID3=2 DEV_ID2=3 DEV_ID1=4
 DEV_ID0=5 IRV_IN4=6 IRV_IN3=7 IRV_IN2=8
 IRV_IN1=9 IRV_IN0=10 TXEN=11
 IR_ACTIVE_IN_=13 VECT_EQ_=14 CRS=15
 IRVOUT4=16 IRVOUT3=17 IRVOUT2=18 IRVOUT1=19
 IRVOUT0=20
 int_vect_eq_1_=21 int_vect_eq_2_=22

EQUATIONS
IRVOUT4 = DEV_ID4 + /TXEN
IRVOUT3 = DEV_ID3 + /TXEN
IRVOUT2 = DEV_ID2 + /TXEN
IRVOUT1 = DEV_ID1 + /TXEN
IRVOUT0 = DEV_ID0 + /TXEN
VECT_EQ_ = int_vect_eq_1_ + int_vect_eq_2_
CRS = TXEN + /IR_ACTIVE_IN_
int_vect_eq_1_ = (DEV_ID0 * /IRV_IN0 + /DEV_ID0 * IRV_IN0) + (DEV_ID1 * /IRV_IN1 +
                 /DEV_ID1 * IRV_IN1)
int_vect_eq_2_ = (DEV_ID2 * /IRV_IN2 + /DEV_ID2 * IRV_IN2) + (DEV_ID3 * /IRV_IN3 +
                 /DEV_ID3 * IRV_IN3) + (DEV_ID4 * /IRV_IN4 + /DEV_ID4 * IRV_IN4)
```

8.0 Initialization Code

8.1 NS486 Initialization Code

```
;***************************
; Internal and External Chip Select setup
; for the 100RIB, MAC and External UART
;***************************

InitBIU  proc near

; --- Disable all CSs
         out BIU_CS_EN,000h

; --- Setup 100RIB as CS1, width 0ffh
         out BIU_CSBAR1_0, RIB_BASE
         out BIU_CSBAR1_2, 00000h
         out BIU_CSARR1_0, 000ffh
         out BIU_CSARR1_2, 00000h

; --- Setup External UART as CS2, width 07h
         out BIU_CSBAR2_0, UART_BASE
         out BIU_CSBAR2_2, 00000h
         out BIU_CSARR2_0, 00007h
         out BIU_CSARR2_2, 00000h

; --- Setup MAC as CS3, width 01fh
         out BIU_CSBAR3_0, 00300h
         out BIU_CSBAR3_2, 00000h
         out BIU_CSARR3_0, 0001fh
         out BIU_CSARR3_2, 00000h

; --- Select which are memory
         out BIU_CS_TYPE, 000h

; --- RIB is 16-bit, UART is 8-bit, MAC is 16-bit
         out BIU_P16LCSR, 005h

; --- Set-up timing for RIB chip select
         out BIU_CSATR1, 00Ch

; --- Set-up timing for MAC chip select
         out BIU_CSATR2, 00Ch

; --- Map the logical chip selects to the physical signals
         out BIU_ECSSR1, 001h ;RIB
         out BIU_ECSSR4, 002h ;External UART

;out    BIU_ECSSR3, 003h;MAC (Not Used)
```

8.0 Initialization Code (Continued)

```
; --- Set all CSs as cacheable ---
         out BIU_CCSR, 0FFh

; --- Enable the BIU chip selects ---
         out BIU_CS_EN, 05h   ;Enable RIB and MAC
         ret
InitBIU         endp

;**************************
; Verification routines
; The following I/O reads can be performed
; to verify that each of the components has
; been properly enabled on the Management  board.
;**************************

Verify  proc near

; --- Check for MAC - should return 2201h if present

CheckMAC:
         in MAC_ID_REG
         cmp ax,2201h
         je CheckRIB
         << MAC NOT PRESENT >>

; --- Check for 100RIB - should return 0001h if present

CheckRIB:
         in RIB_ID_REG
         cmp ax,0001h
         je Done
         << 100RIB NOT PRESENT >>
Done:
         ret
Verify endp

;**************************
; External UART initialization code.
;**************************

UartInit  proc near

; --- set DLAB, no parity, 1 stop, 8 bit word
         out UART_LCR,083h

; --- set baud rate LSB (9600baud)
         out UART_DLL,12h
```

8.0 Initialization Code (Continued)

```
; --- set baud rate MSB (9600baud)
         out UART_DLM,00h

; --- clear DLAB, no parity, 1 stop, 8 bit word
         out UART_LCR,003h

; --- enable FIFOs (if 16550) Trigger = 1 byte
         outpb (base + UART_FCR),007h

; --- disable all interrupts (use polling)
         outpb (base + UART_IER),000h

UartInit  endp
```

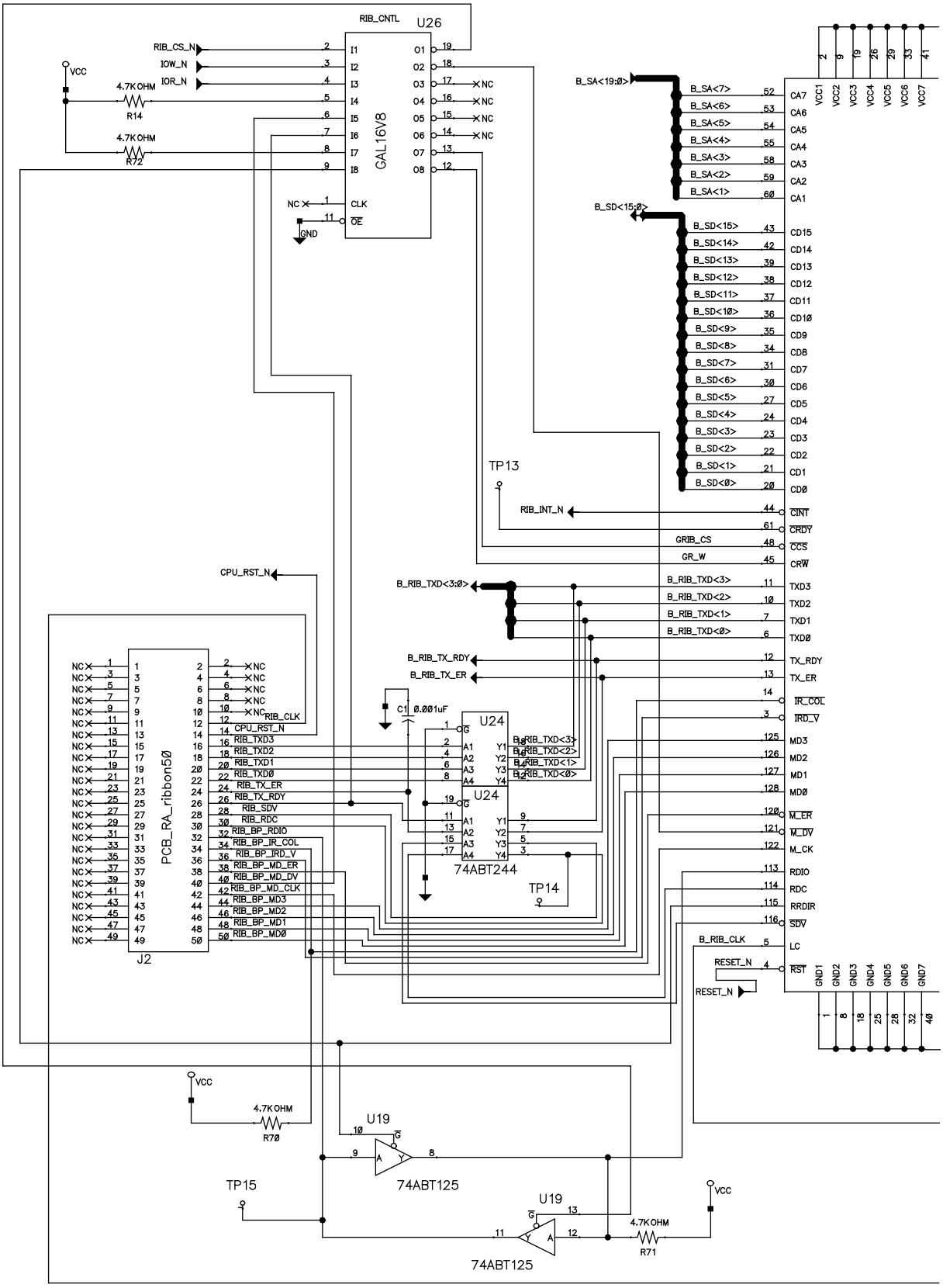
8.2 MAC initialization Code

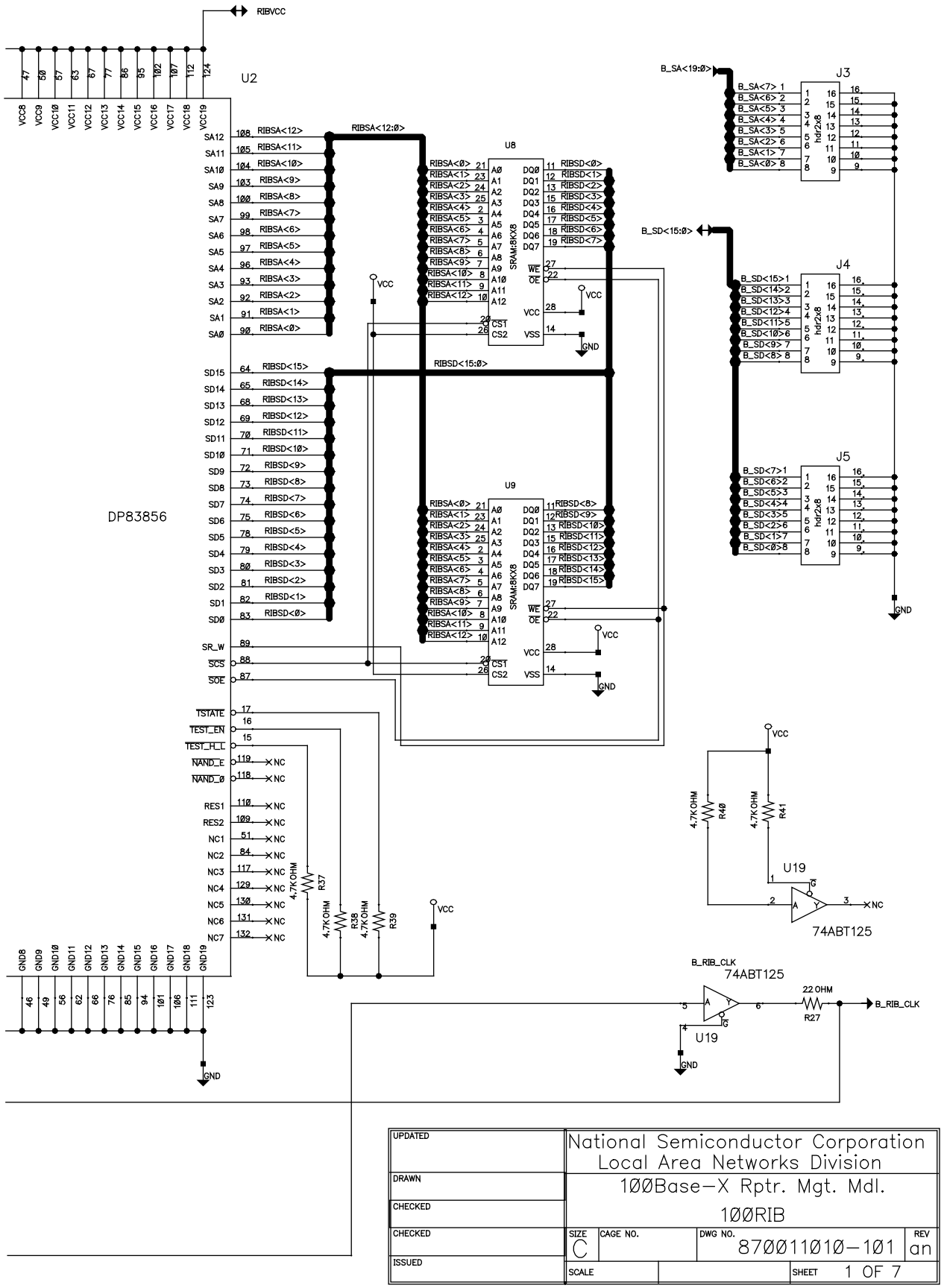
```
;**************************
; NOTE: MAC register reads and writes have been
; simplified - read/write routines with a critical
; section should be used if interrupts are to be
; implemented.
;**************************

InitMAC proc near
; --- Check for MAC - ID Register should return 2201h, otherwise exit
         in MAC_ID_REG
         cmp ax,2201h
         jne mac_init_exit

; --- Perform MAC software reset
         out MAC_COMMAND,4008h

; === TRANSMIT SETUP ===

; --- Setup TCR for blind transmit, auto padding
         out MAC_TX_CONFIG,0040h

; --- Setup Tx threshold to maximum
         out MAC_TX_FREE_THRESH,08000h

; --- Set minimum TX threshold to maximum
         out MAC_TX_MIN_THRESH,00040h

; --- Set Send Retries Register for 16 retries
         out MAC_TX_RETRY,0000h

; === RECEIVE SETUP ===
```

8.0 Initialization Code (Continued)

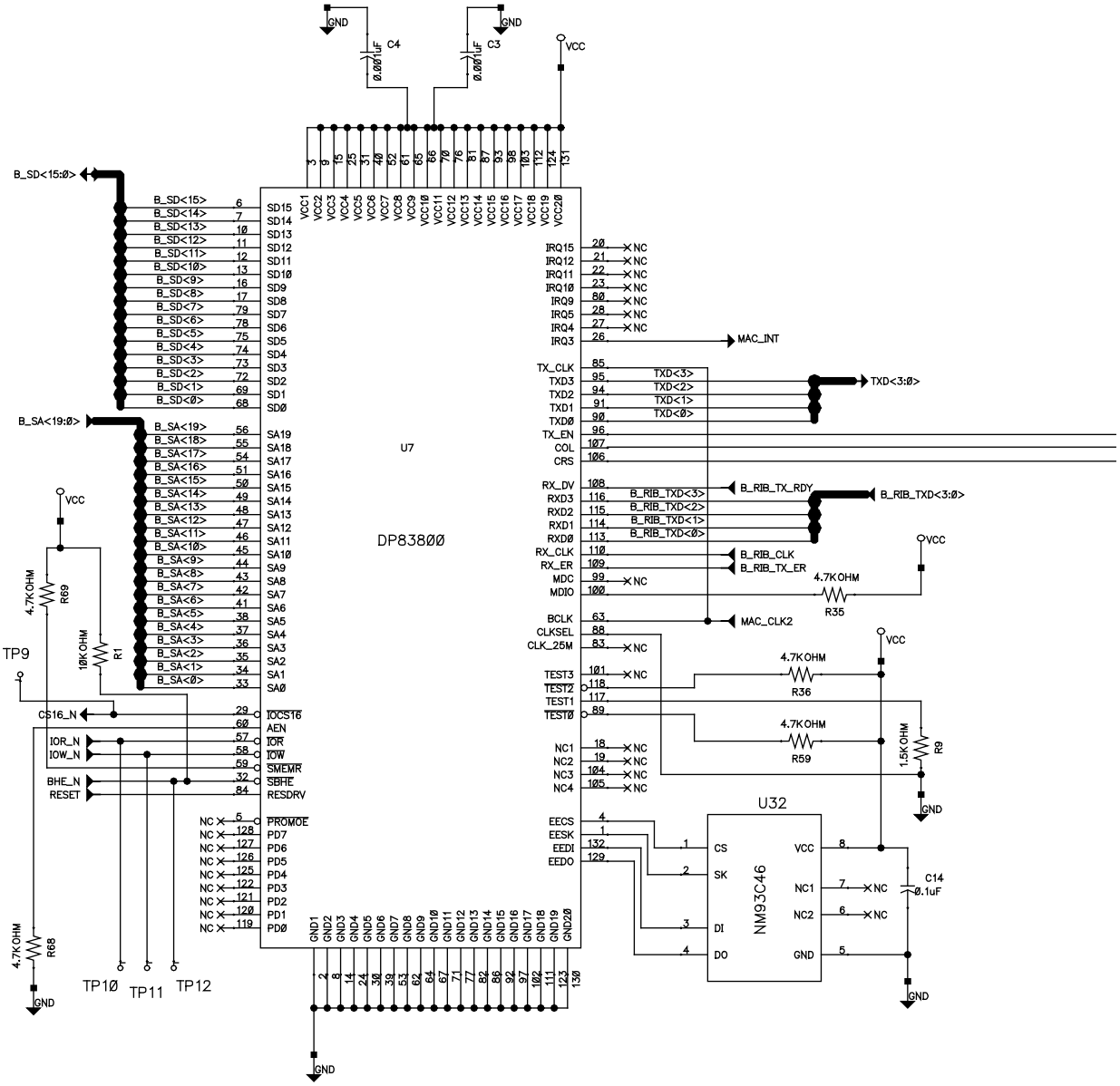
```
; --- Setup Rx threshold
          out MAC_RX_THRESH,0000h

; --- Setup MCR for 32-bit access
          out MAC_MODE_CONFIG,0001h

; === INTERRUPT SETUP ===

; --- Setup Interrupt Mask Register
          out MAC_INT_MASK,0009h

; --- Clear the Interrupt Vector Register - wait

ClearIVR:
          in MAC_INT_VECT
          cmp ax,0
          jnz ClearIVR

; --- Disable physical interrupt generation
          out MAC_INT_ENABLE,0000h

; === MIB SETUP ===

; --- Clear and enable MIB counters
          out MAC_MIB_CONTROL,0F100h

; === CAM SETUP ===

; --- Disable CAM
          out MAC_CAM_CONTROL,0080h

; --- Put Ethernet address into CAM register
          out MAC_CAM_DATA,[SRC_STORE]
          out MAC_CAM_DATA,[SRC_STORE+2]
          out MAC_CAM_DATA,[SRC_STORE+4]

; --- Set CAM Mask
          out MAC_CAM_MASK,0001h

; --- Enable CAM
          out MAC_CAM_CONTROL,08000h

; --- Set receive enable bits in Command Register
          out MAC_COMMAND,1208h

mac_init_exit:
          call ControlPkt
          ret

InitMAC endp
```

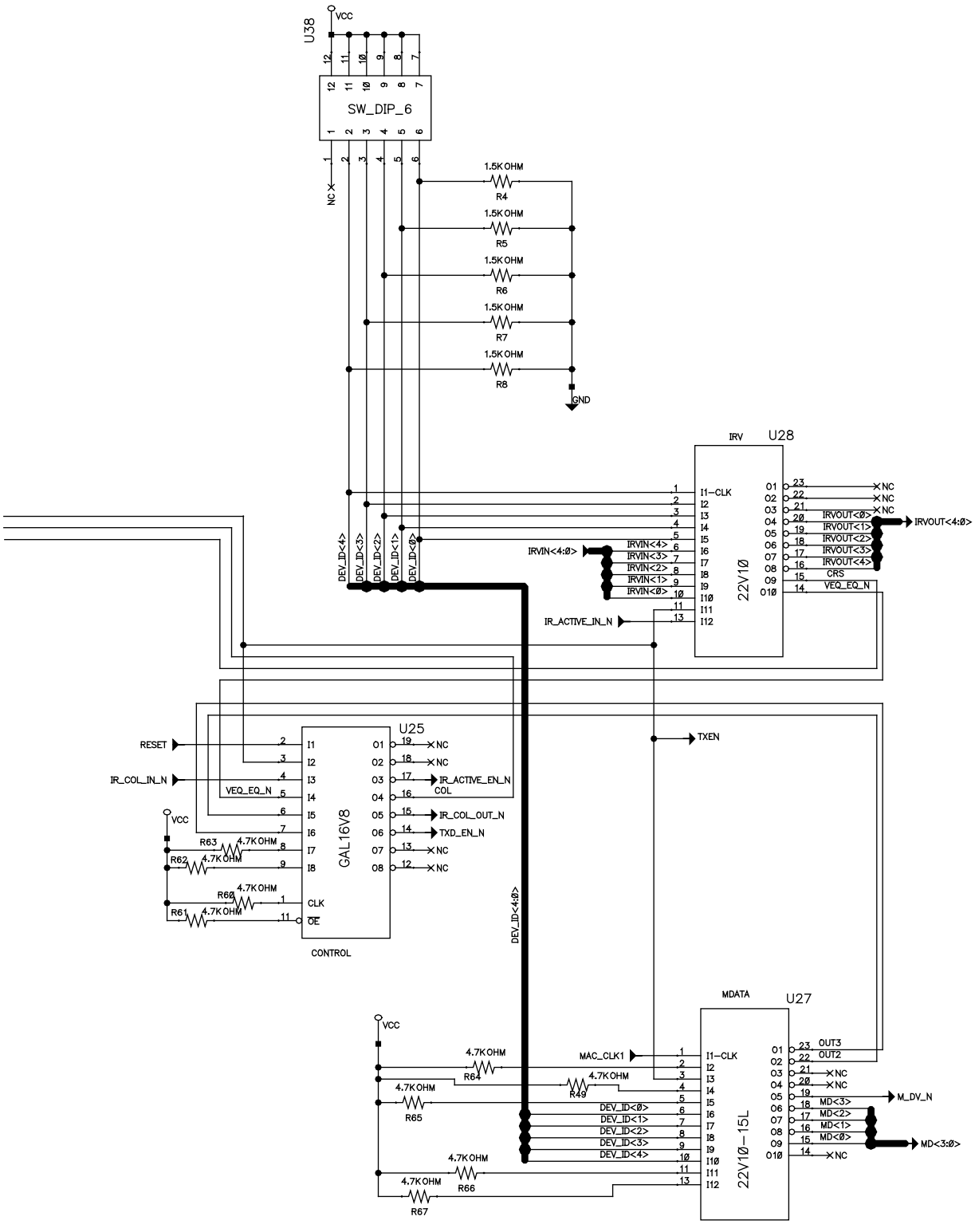
9.0 Schematics





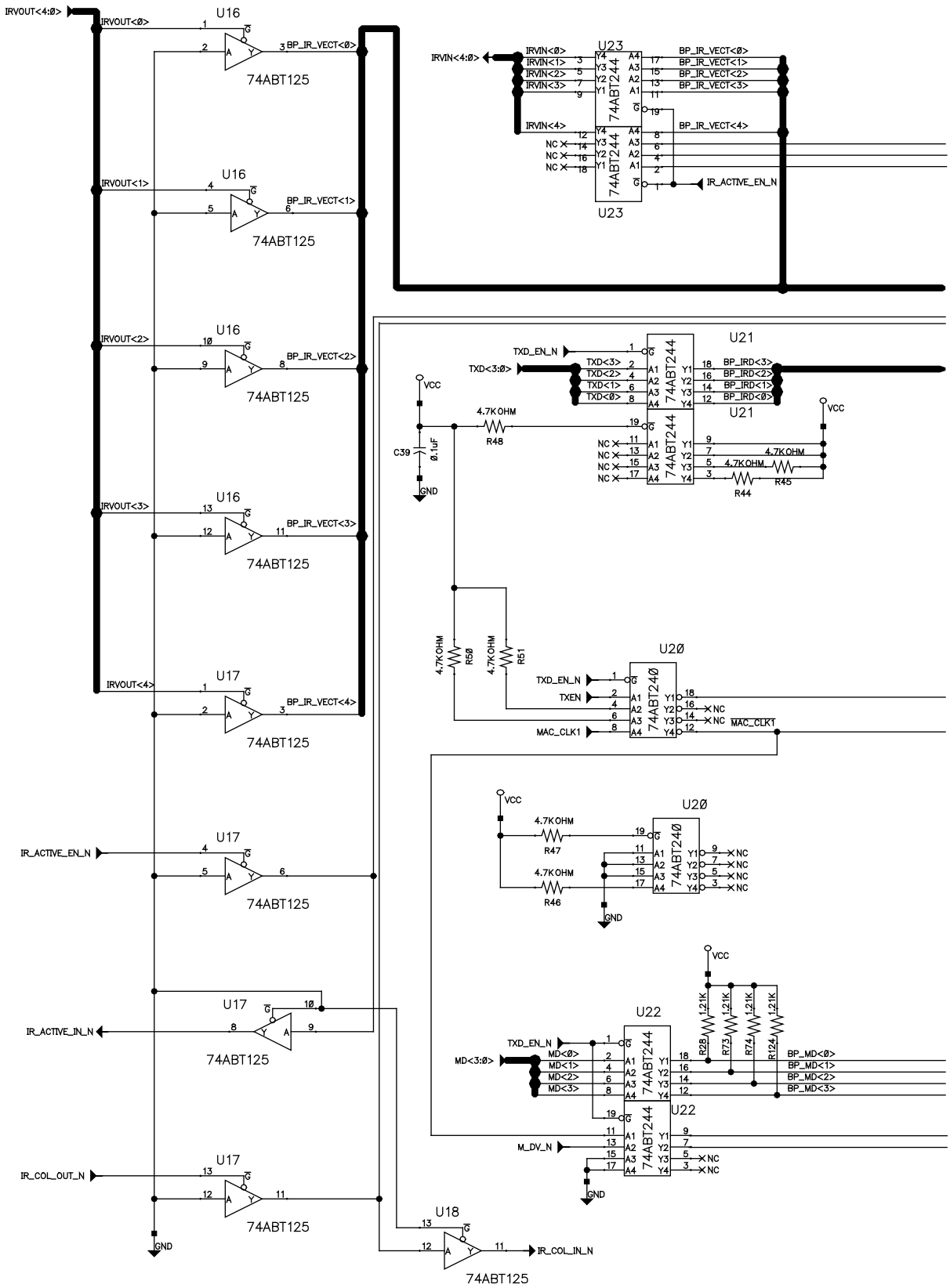
UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base–X Rptr. Mgt. Mdl.			
CHECKED	100RIB			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010–101	REV an
ISSUED	SCALE		SHEET 1 OF 7	


U7
DP83800
U32
NM93C46
B_SD<15:0>
B_SA<19:0>
MAC_INT
TXD<3:0>
B_RIB_TX_RDY
B_RIB_TXD<3:0>
B_RIB_CLK
B_RIB_TX_ER
MAC_CLK2
4.7KOHM
10KOHM
TP9
TP10
TP11
TP12
VCC
GND
C4
C3
0.001uF
C14
0.1uF
1.5K OHM
R1
R9
R35
R36
R59
R68
R69

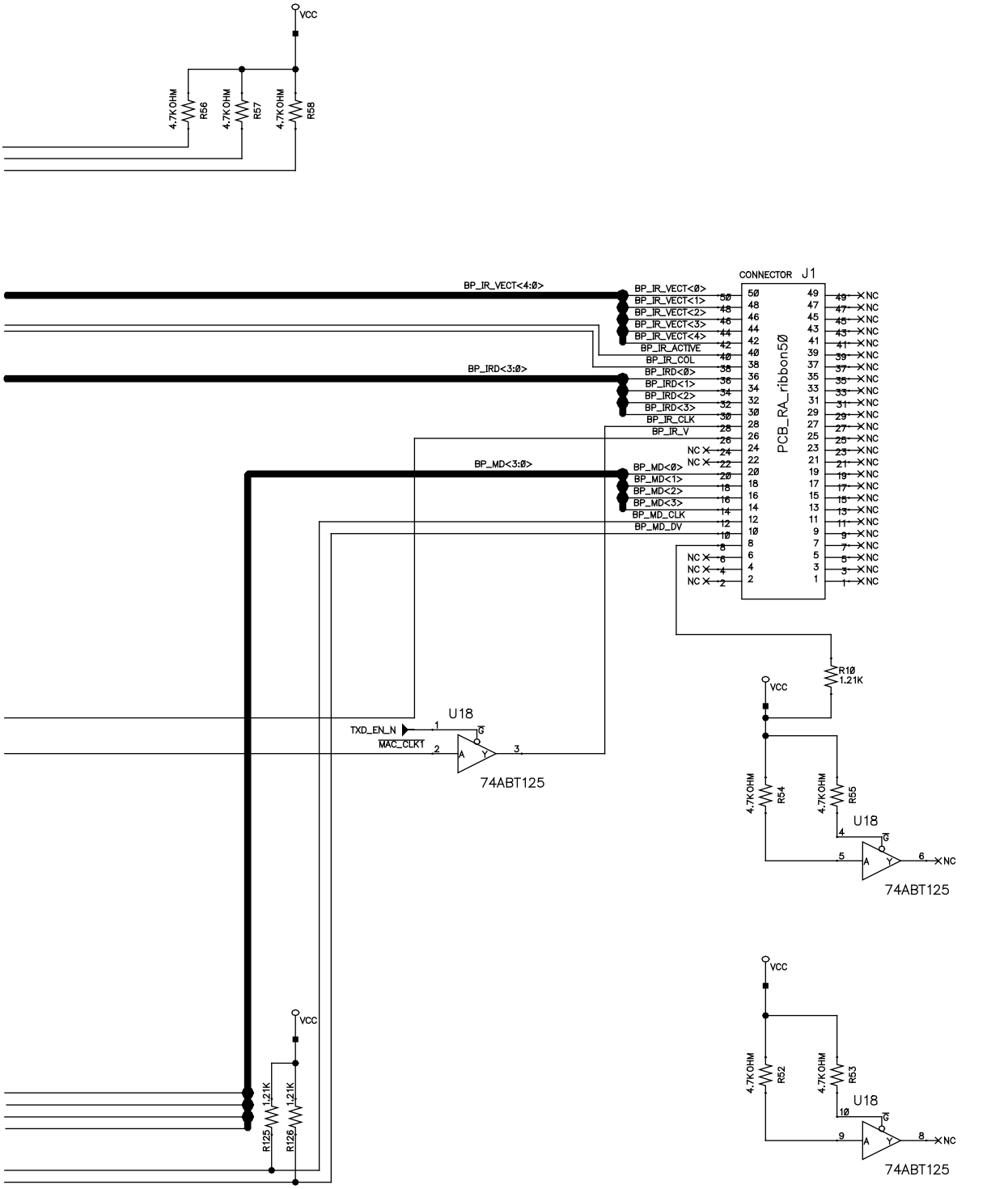


UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base–X Rptr. Mgt. Mdl.			
CHECKED	100 Mb/s MAC			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010–101	REV an
ISSUED	SCALE		SHEET 2 OF 7	

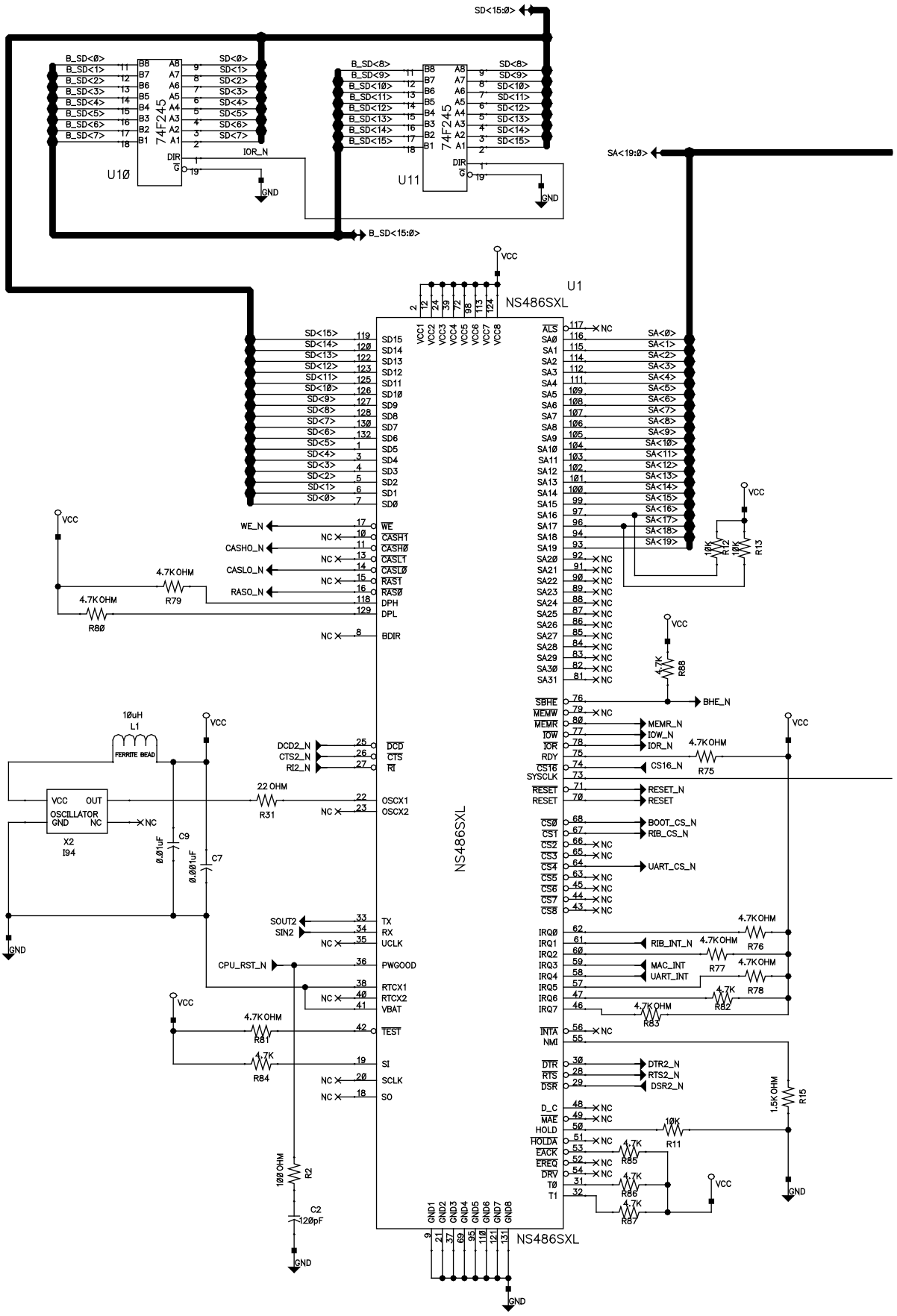
9.0 Schematics (Continued)



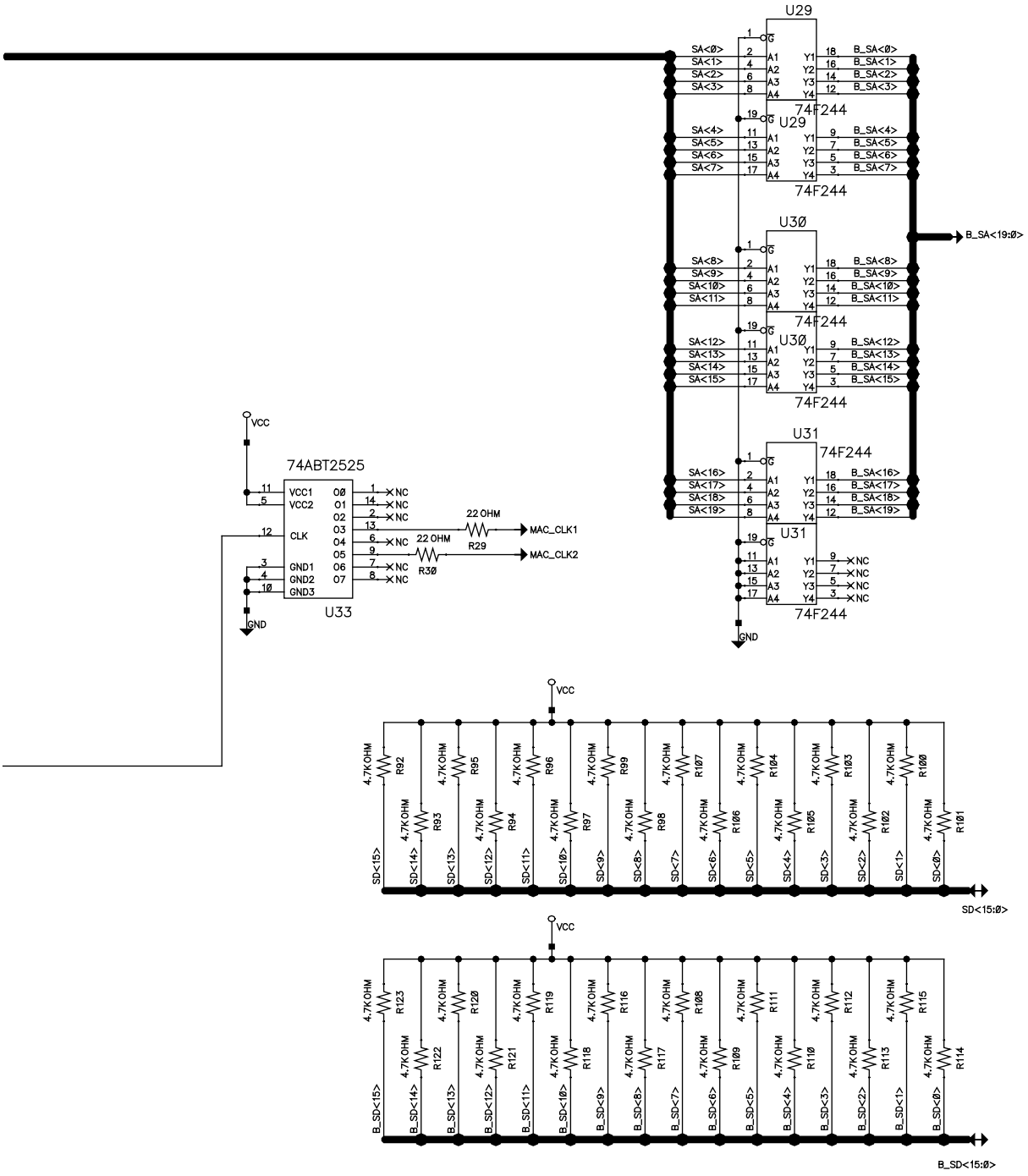
9.0 Schematics (Continued)



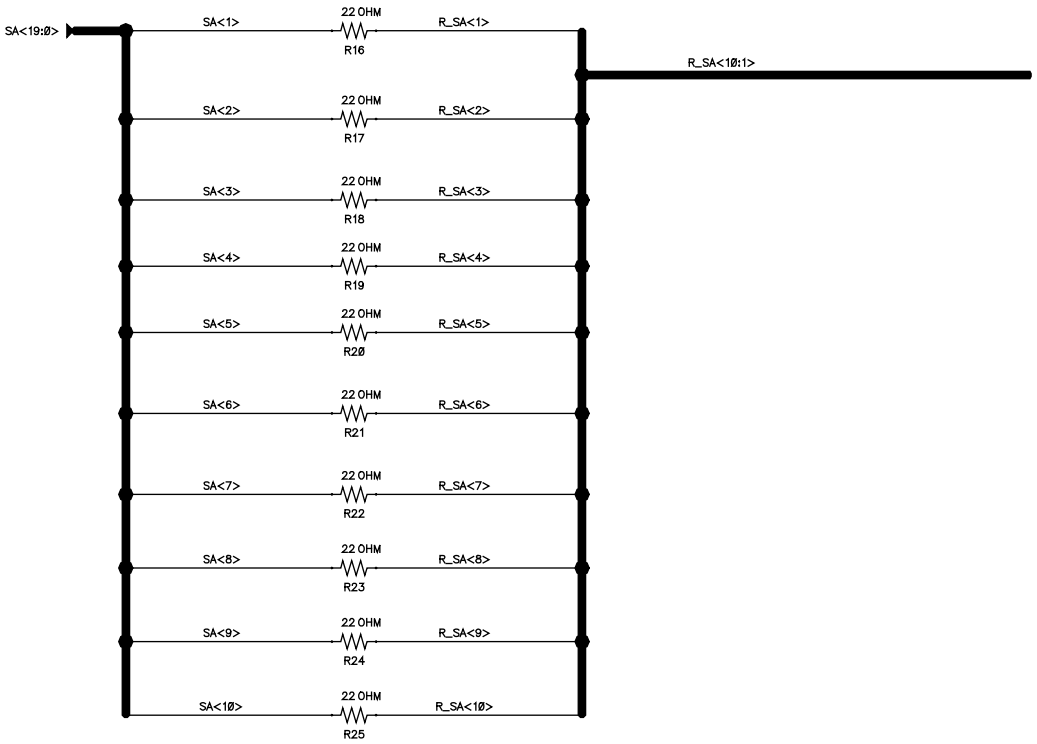
UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base–X Rptr. Mgt. Mdl.			
CHECKED	Inter RIC Bus			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010–101	REV an
ISSUED	SCALE		SHEET 3 OF 7	

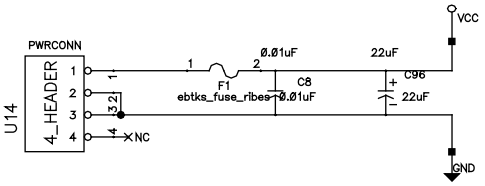

SD<15:0>
B_SD<15:0>
SA<19:0>
U1
NS486SXL
74F245
U10
U11
IOR_N
GND
VCC
WE_N
CASHO_N
CASLO_N
RASO_N
4.7K OHM
R79
R80
BDIR
10uH
L1
FERRITE BEAD
OSCILLATOR
X2
I94
22 OHM
R31
C9
0.01uF
C7
0.001uF
CPU_RST_N
100 OHM
R2
C2
120pF
4.7K OHM
R81
4.7K
R84
R88
BHE_N
MEMR_N
IOW_N
IOR_N
R75
CS16_N
RESET_N
BOOT_CS_N
RIB_CS_N
UART_CS_N
RIB_INT_N
MAC_INT
UART_INT
R76
R77
R78
R82
R83
DTR2_N
RTS2_N
DSR2_N
1.5K OHM
R15
10K
R11
4.7K
R85
R86
R87
R12
R13

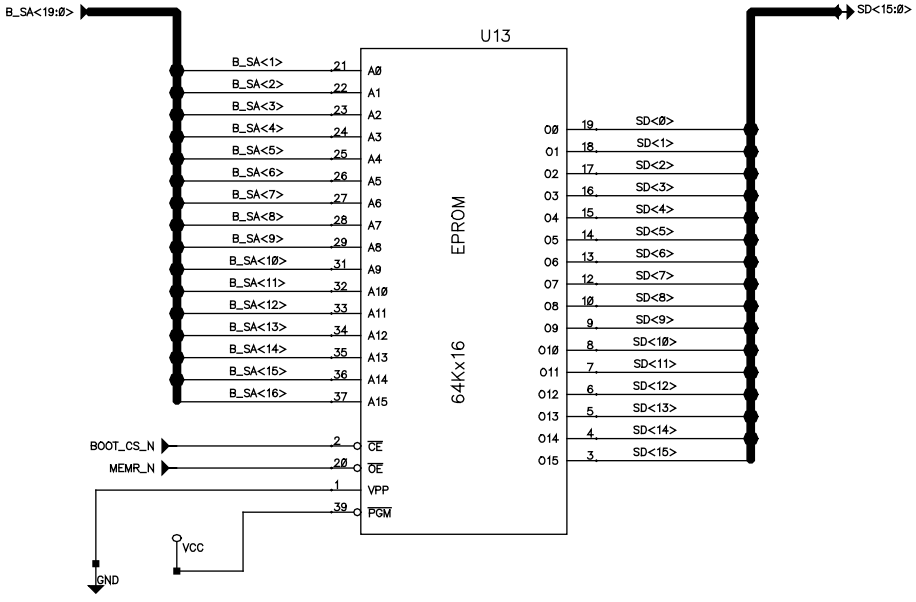
9.0 Schematics (Continued)

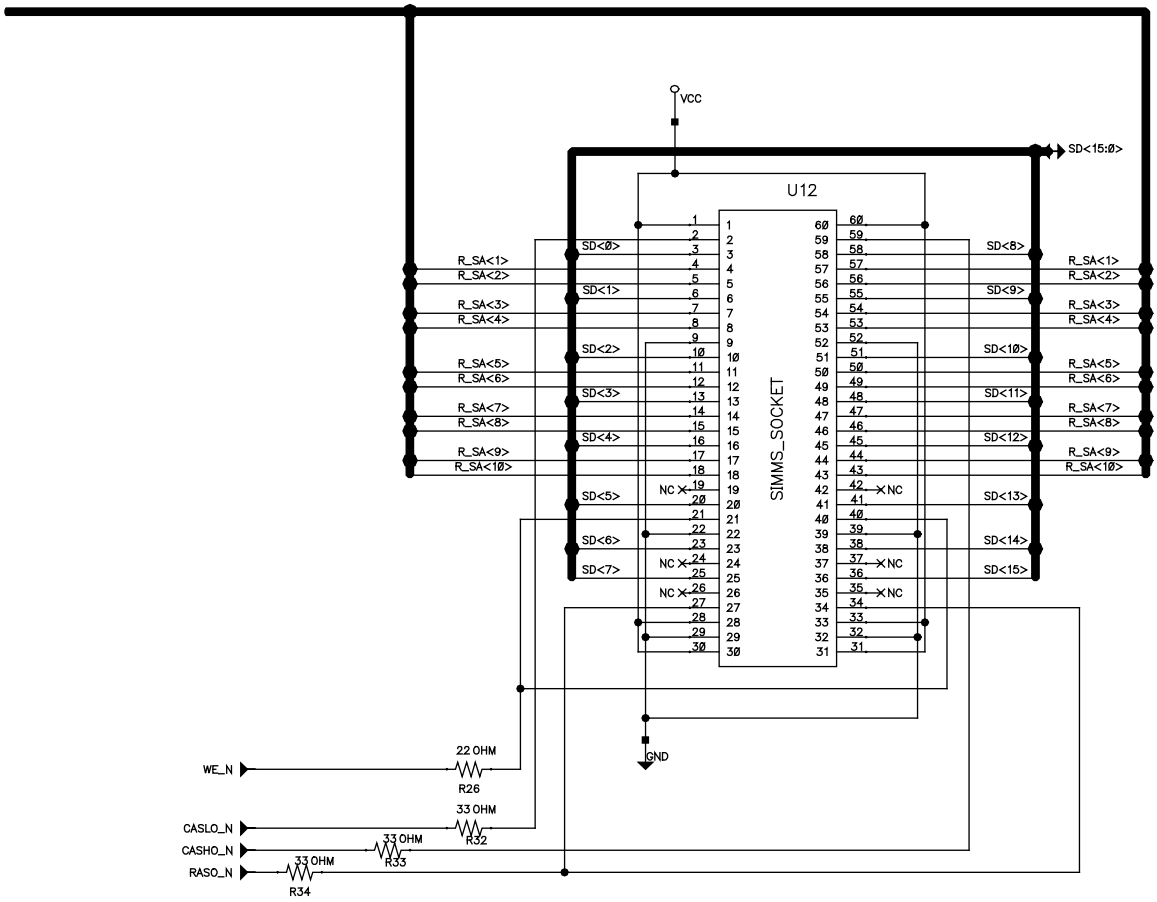


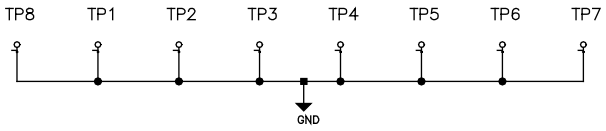
UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base-X Rptr. Mgt. Mdl.			
CHECKED	CPU			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010-101	REV an
ISSUED	SCALE		SHEET 4 OF 7	


SA<19:0>
SA<1>
22 OHM
R16
R_SA<1>
R_SA<10:1>
SA<2>
R17
R_SA<2>
SA<3>
R18
R_SA<3>
SA<4>
R19
R_SA<4>
SA<5>
R20
R_SA<5>
SA<6>
R21
R_SA<6>
SA<7>
R22
R_SA<7>
SA<8>
R23
R_SA<8>
SA<9>
R24
R_SA<9>
SA<10>
R25
R_SA<10>

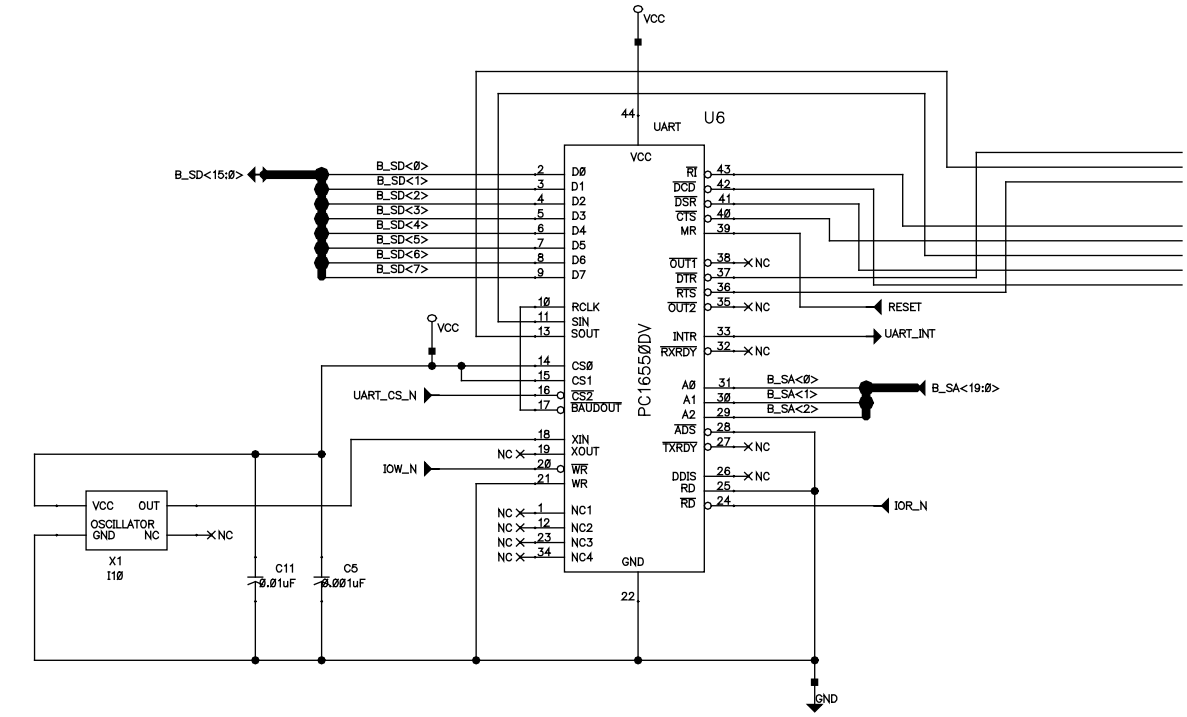

VCC
PWRCONN
U14
4_HEADER
0.01uF
22uF
F1
ebtks_fuse_ribes
C8
0.01uF
C96
22uF
NC
GND

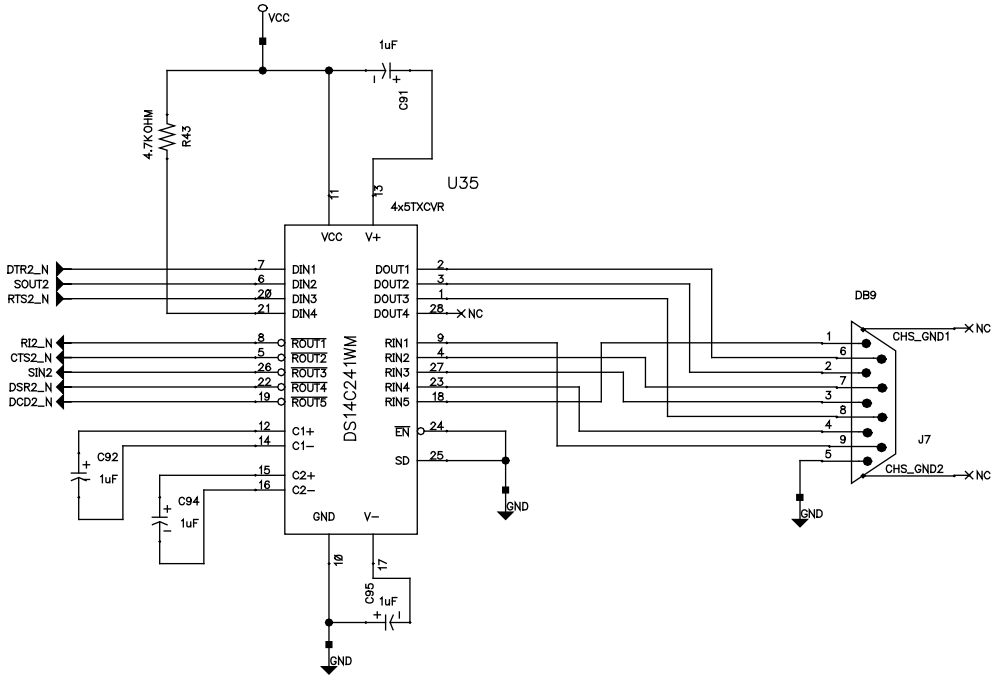

B_SA<19:0>
U13
EPROM
64Kx16
B_SA<1>
B_SA<2>
B_SA<3>
B_SA<4>
B_SA<5>
B_SA<6>
B_SA<7>
B_SA<8>
B_SA<9>
B_SA<10>
B_SA<11>
B_SA<12>
B_SA<13>
B_SA<14>
B_SA<15>
B_SA<16>
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
BOOT_CS_N
MEMR_N
OE
VPP
PGM
VCC
GND
O0
O1
O2
O3
O4
O5
O6
O7
O8
O9
O10
O11
O12
O13
O14
O15
SD<15:0>
SD<0>
SD<1>
SD<2>
SD<3>
SD<4>
SD<5>
SD<6>
SD<7>
SD<8>
SD<9>
SD<10>
SD<11>
SD<12>
SD<13>
SD<14>
SD<15>



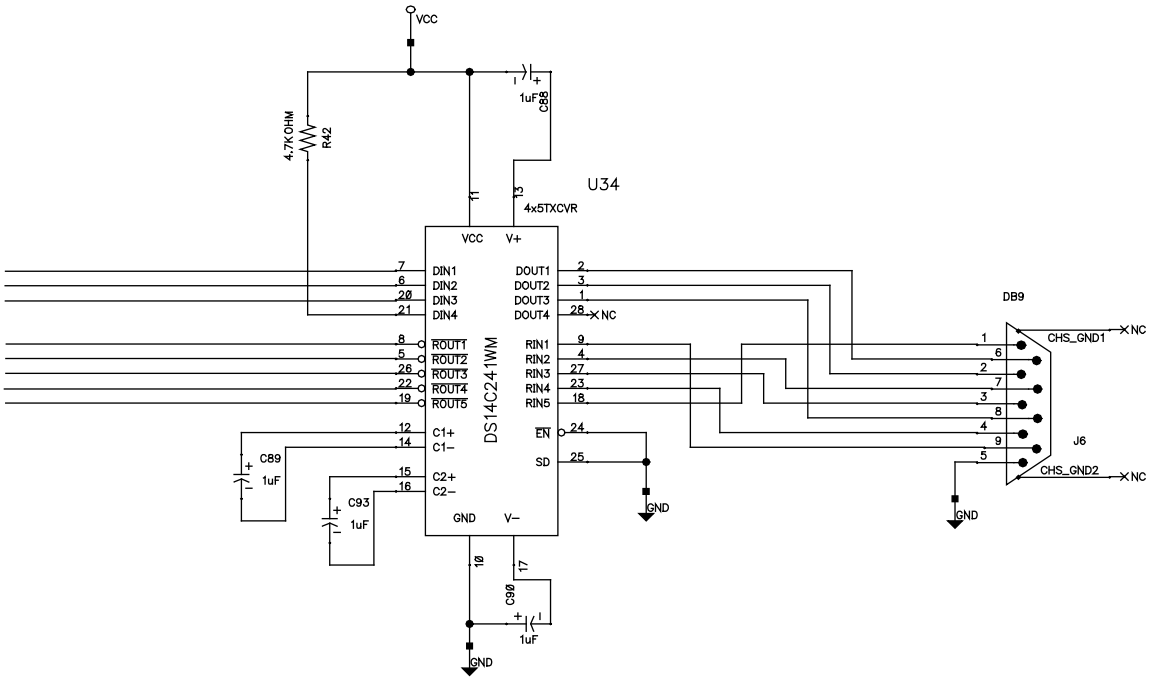


UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base–X Rptr. Mgt. Mdl.			
CHECKED	Memory and Power			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010–101	REV an
ISSUED	SCALE		SHEET 5 OF 7	


VCC
U6
UART
PC16550DV
B_SD<15:0>
B_SD<0>
B_SD<1>
B_SD<2>
B_SD<3>
B_SD<4>
B_SD<5>
B_SD<6>
B_SD<7>
D0
D1
D2
D3
D4
D5
D6
D7
RCLK
SIN
SOUT
CS0
CS1
CS2
BAUDOUT
XIN
XOUT
WR
WR
NC1
NC2
NC3
NC4
RI
DCD
DSR
CTS
MR
OUT1
DTR
RTS
OUT2
INTR
RXRDY
A0
A1
A2
ADS
TXRDY
DDIS
RD
RD
GND
RESET
UART_INT
B_SA<0>
B_SA<1>
B_SA<2>
B_SA<19:0>
IOR_N
IOW_N
UART_CS_N
NC
X1
I10
OSCILLATOR
C11
C5
0.01uF

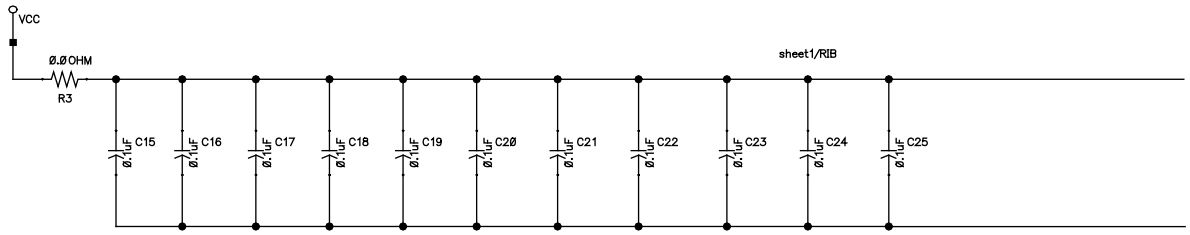

VCC
1uF
C91
4.7KOHM
R43
U35
4x5TXCVR
DS14C241WM
VCC
V+
GND
V-
DIN1
DIN2
DIN3
DIN4
DOUT1
DOUT2
DOUT3
DOUT4
ROUT1
ROUT2
ROUT3
ROUT4
ROUT5
RIN1
RIN2
RIN3
RIN4
RIN5
EN
SD
C1+
C1-
C2+
C2-
C92
C94
C95
DTR2_N
SOUT2
RTS2_N
RI2_N
CTS2_N
SIN2
DSR2_N
DCD2_N
DB9
J7
CHS_GND1
CHS_GND2
NC

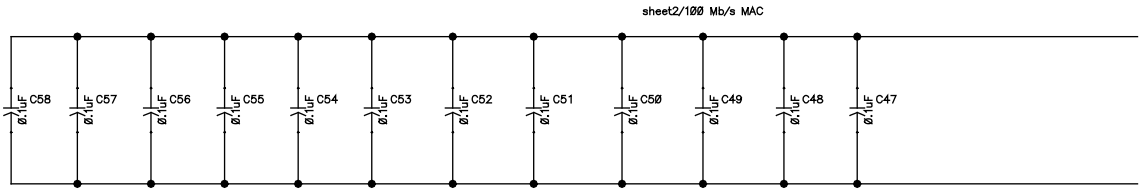
9.0 Schematics (Continued)

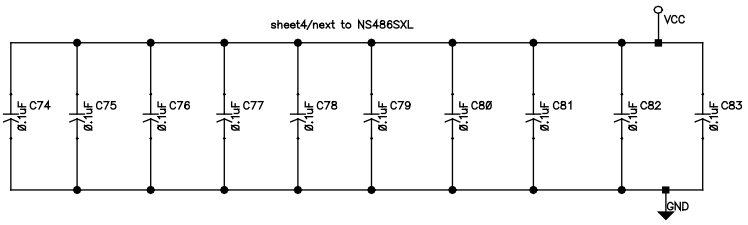


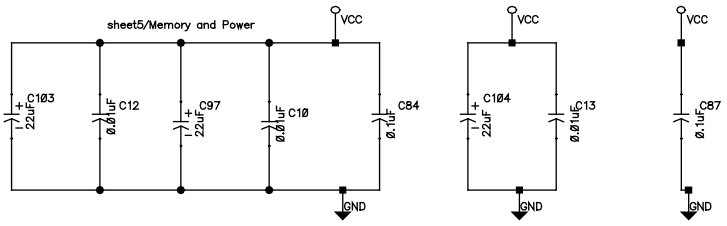
UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	100Base−X Rptr. Mgt. Mdl.			
CHECKED	Serial Ports			
CHECKED	SIZE C	CAGE NO.	DWG NO. 870011010−101	REV an
ISSUED	SCALE		SHEET 6 OF 7	

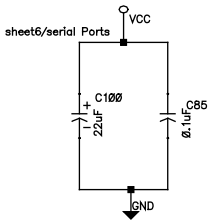
9.0 Schematics (Continued)



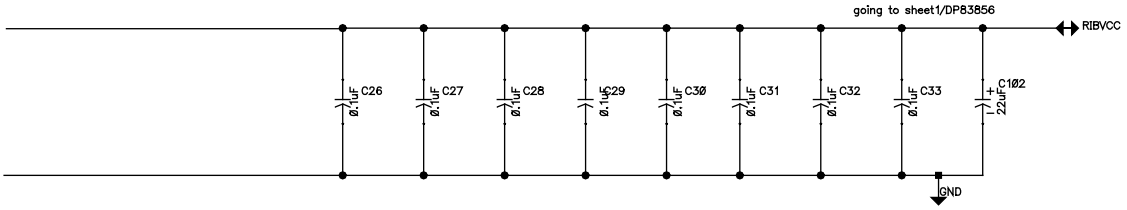


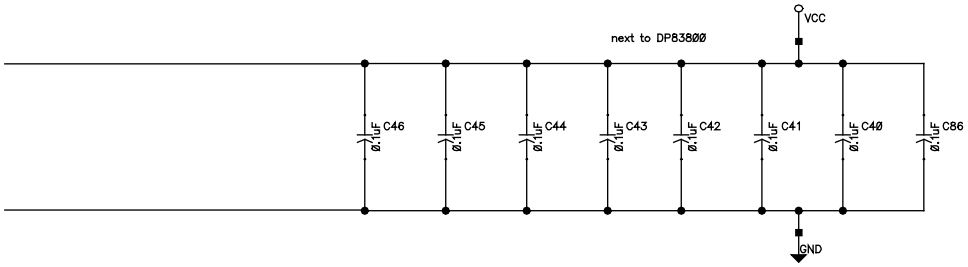


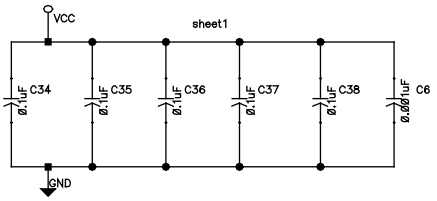


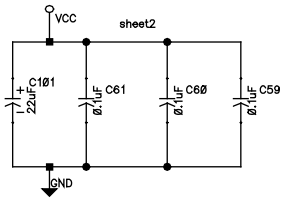


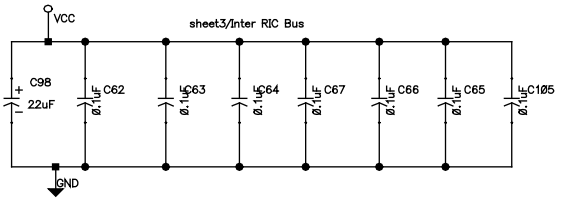
9.0 Schematics (Continued)

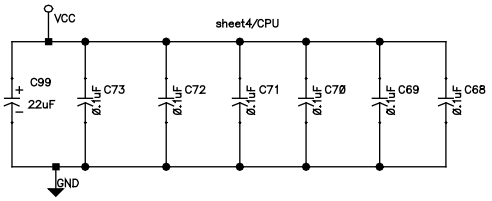












UPDATED	National Semiconductor Corporation Local Area Networks Division			
DRAWN	1ØØBase−X Rptr. Mgt. Mdl.			
CHECKED	Supply Decoupling			
CHECKED	SIZE C	CAGE NO.	DWG NO. 87ØØ11Ø1Ø−1Ø1	REV an
ISSUED	SCALE		SHEET 7 OF 7	

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-254-4466
Fax: 65-250-4466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179

www.national.com