MiniFrame™ Input/Output Expansion Hardware Manual

MiniFrame™ Input/Output Expansion Hardware Manual

CAUTION

This equipment generates, uses, and can radiate radio frequency energy. If this product is not installed and used in accordance with directions in this manual, it may cause interference to radio communications. It has been tested and found to comply with the limits for Class A computing devices, pursuant to Subpart J of Part 15 FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause radio interference; if this occurs the user, at his own expense, is required to make any necessary corrections to stop such interference.

Specifications Subject to Change

First Edition

PREFACE

This manual describes Convergent Technologies MiniFrame Computer System input/output expansion board. It is written primarily for programmers, field service personnel, continuation engineers, and technical trainers. The manual includes a section on installation.

There are three types of input/output expansion boards:

- MiniFrame Ethernet Expansion Board
- MiniFrame Communications Expansion Board (RS-232-C)
- MiniFrame Combination Expansion Board (Ethernet and RS-232-C).

This manual describes all three boards. The manual delineates and presents technical information for all three boards in the architecture and theory of operation sections, as follows:

- Common information: pertains to the interrupt controller and the address decoder and interface, which is located on all three boards.
- RS-232-C information: pertains to RS-232-C circuitry, which is located on the Communications Expansion Board and Combination Expansion Board.
- Ethernet information: pertains to Ethernet circuitry, which is located on the Ethernet Expansion Board and Combination Expansion Board.

Throughout the manual, the following conventions apply:

- Signal name acronyms appear in uppercase.
- Active low signal names end with a minus sign (for example, XCLK-), and active high signal names end with a plus sign (for example, XCLK+).

- Sheet numbers in parentheses following signal names or hardware elements refer to the expansion board schematics (located in Appendix C). The schematics omit the plus signs in active high signal names, and active low signal names have the suffix " ", rather than "-".

The manual contains 4 sections and 1 appendix, as follows:

Section 1 is an overview of the three input/output expansion boards.

Section 2 describes how to install the expansion board into the MiniFrame Computer.

Section 3 describes the boards' functional elements, registers, and interface.

Section 4 describes boards' circuitry and operation.

Appendix A contains the schematics.

CONTENTS

LIST OF TABLES

SECTION 1: OVERVIEW

The MiniFrame Computer System centers around the Motorola MC68Ø1Ø microprocessor (CPU). The system contains five main functional elements:

- o System control
- o Processor control
- o Memory control
- o Input/output
- o Interrupt control.

This manual assumes familiarity with the MiniFrame Hardware Manual, which describes the MiniFrame Computer System at a functional, circuit, and system level.

The MiniFrame input/output expansion board provides added input/output to the MiniFrame Computer System in the form of either an Ethernet local area network, eight RS-232-C communications channels, or both. All three boards includes the following functional elements, with the noted exceptions:

- o Interrupt controller
- o Address decoder and interface
- o RS-232-C control (not on Ethernet Expansion Board)
- o Ethernet control (not on Communications Expansion Board).

The remainder of Section 1 gives brief overviews, with illustrations, of the three boards.

COMMUNICATIONS EXPANSION BOARD OVERVIEW

The Communications Expansion Board provides an interface to peripherals (eg. terminals printers, and modems) with eight RS-232-C input/output channels, in addition to the two channels on the main board. The RS-232-C circuitry centers around four 8274 multi-protocol serial controller chips. Each chip has an A-port and a B-port, for two channels. Each channel has a separate onboard baud rate clock. The CPU can also select an external clock for a channel, supplied by the peripheral connected to that channel.

All interaction between the 8274s and the main board occur via CPU intervention. If an 8274 needs to read or write memory, it interrupts the the CPU. The CPU routes the data between the 8274 and memory by reading one element, then writing the other.

All interrupts from the 8274s have CPU interrupt priority level five. Within level five, the main board 8274 has the higher priority, and all four expansion 8274s have equal priority. The interrupt controller on the expansion board enables one expansion 8274 interrupt line at a time. The main board 8274 interrupt takes precedence over expansion interrupts. The CPU can mask out level expansion interrupts either internally (by an instruction) or externally (by masking the main board programmable interrupt controller).

Figure 1-1 illustrates a Communications Expansion Board.

This and the next two drawings are simple drawings of the three boards. An artist will draw them during production.

Figure 1-1. Communications Expansion Board

ETHERNET EXPANSION BOARD OVERVIEW

The Ethernet Expansion Board supplies the interface to an Ethernet network operating at a transfer rate of 10M bits/second. Ethernet control centers around the AM7990 Local Area Network Controller for Ethernet (LANCE) and the AM7991 Serial Interface Adapter (SIA). The 7990 performs memory management (on board DMA), error reporting, packet handling, and interface functions. The 7991 performs Manchester encoding and decoding of the serial data stream with phase-locked-loop clock recovery.

At initialization, the operating system sets up the Ethernet registers and reserves a portion of memory for communication between the CPU and the Ethernet control. The CPU can read or write the Ethernet registers any time that the 7990 is not doing a packet transmission and not reading or writing memory.

Most Ethernet interrupts are not error related; usually they signify the beginning of a packet transmission or end of a packet reception. Because of this, the CPU does not have to respond to Ethernet interrupts immediately. After the 7990 generates an interrupt, Ethernet input/output operation continues while Ethernet support circuitry (1) fields the interrupt, (2) acknowledges the Ethernet controller, then (3) logs the interrupt in memory and itself interrupts the CPU.

Figure 1-2 illustrates the Ethernet Expansion board.

Figure 1-2. Ethernet Expansion Board.

COMBINATION EXPANSION BOARD OVERVIEW

The Combination Expansion Board combines the circuitry and function of the other two expansion boards. Figure 1-3 illustrates the Combination Expansion Board.

Figure 1-3. Combination Expansion Board.

SECTION 2: INSTALLATION

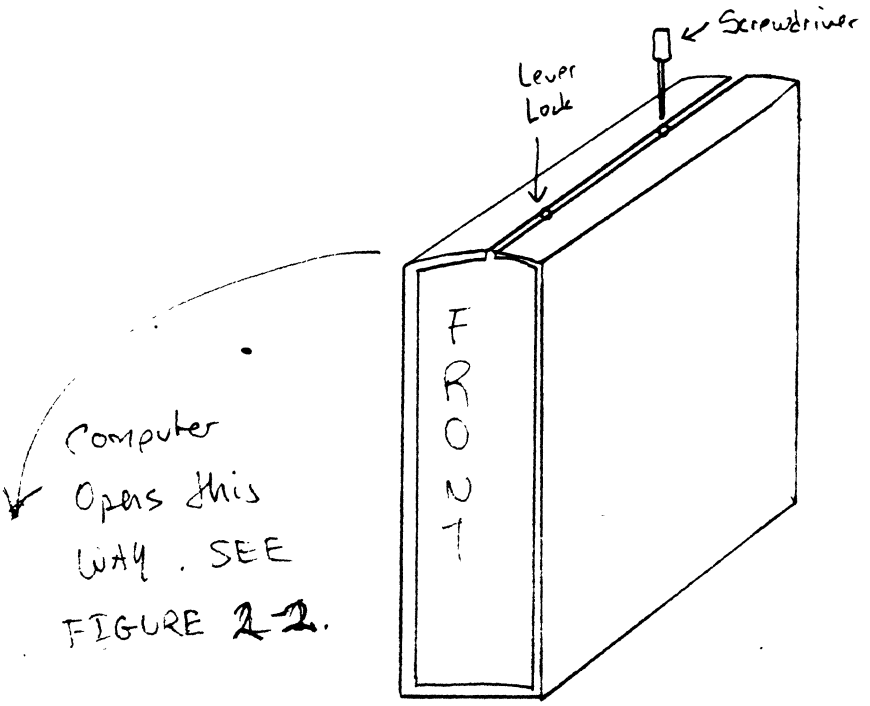
This section describes how to open the MiniFrame Computer cabinet and install or remove expansion boards. Neither of the Ethernet, Communications, or Combination Expansion Boards require configuring. Only qualified service personnel should open the cabinet.

OPENING THE COMPUTER CABINET

To open the computer cabinet:

1. Place the computer on its base on the floor. Facing the front of the cabinet, allow about three feet on the left side for the cabinet to open.

2. Insert a rod shaped tool such as a screwdriver about 1/2 inch into the one of the two small openings on the top of the cabinet. (See Figure 2-1.) Push the tool down aproximately 1/4 inch after you feel resistance. This unlocks one of the two lever locks.



3. While maintaining the downward pressure on the tool, force apart the unlocked half of the cabinet enough to unlock the lever (about 1/2 inch).

4. Repeat steps 2 and 3 to unlock the other lever lock.

5. Swing down the cabinet until it rests on the floor, as illustrated in Figure 2-2. The cabinet contains

   - Power supply
   - Hard disk drive
   - Floppy disk drive
   - Main Processor board
   - Memory Expansion Boards (optional).

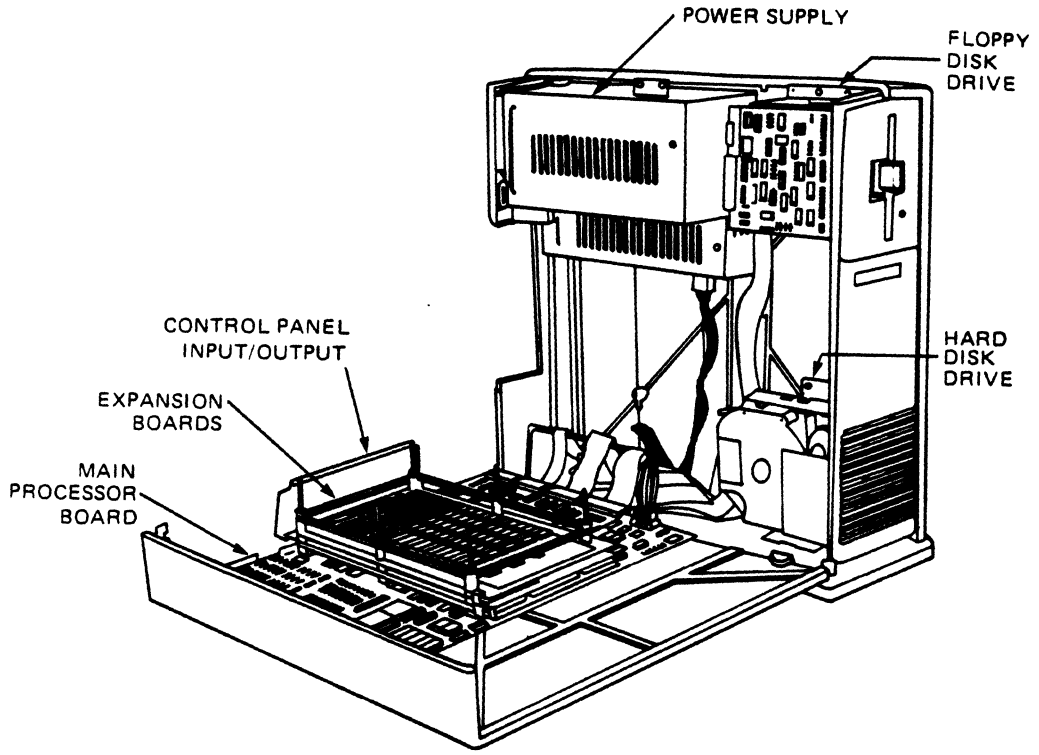


Figure 2-2. Inside the Computer Cabinet

EXPANSION

When installing or removing the expansion board, be aware of the following:

- Expansion boards require no cabling; zero insertion force (ZIF) connectors on each board connect the boards to each other and to the Main Processor board.
- Expansion boards stack on top of the Main Processor board and are supported by a threaded-stud metal structure that contains six studs. Nuts screwed down onto the studs secure the boards. (See Figure 2-2.)
- The Ethernet, Communications, or Combination Expansion Boards must be installed closest to the Main Processor Board, beneath any Memory Expansion Boards.

Installing an Expansion Board

To install an expansion board:

1. Remove and set aside any preexisting Memory Expansion Boards. (See "Removing An Expansion Board".)
2. Open the ZIF connector on the expansion board by pulling the plastic ZIF actuator lever away from the board as far as the lever will go (1/2 inch).
3. Center the board over the threaded-stud and slide the board down until it lies flush with the board underneath it; make sure the ZIF connectors line up.

NOTE

Use care when mating the ZIF connectors, as force can cause damage. Also, the ZIF connector pins must be clean and straight.

3. Install and finger tighten the six nuts that secure the board.
4. Close the ZIF contacts by slowly pushing the plastic ZIF lever towards the board. Keep the lever exactly perpendicular to the side of the board. See Figure 2-3.

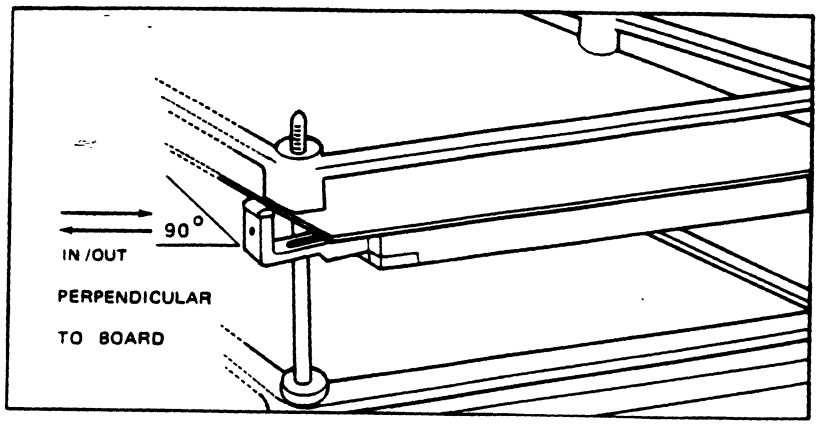


Figure 2-3. The ZIF Actuator Lever

5. When the end of the ZIF lever is flush with the board, the ZIF contacts are closed, and the board is installed.

6. Install any Memory Expansion boards by repeating steps 2 through 6.

7. Close the computer cabinet.

Removing an Expansion Board

To remove an expansion board: (If there are Memory Expansion Boards installed, remove them by this procedure first.)

1. Loosen and remove the six nuts that secure the board to the threaded-stud metal structure.

2. Pull the ZIF actuator lever all the way out (approximately 1/2 inch) from the topmost expansion board. See Figure 2-3.

3. Slowly remove the board by lifting it away from the metal structure.

4. If there are one or more Memory Expansion Boards installed in your computer, repeat steps 2 and 3 until you have removed the expansion board.

5. If you are either replacing the Communications Expansion board or putting Memory Expansion boards back, install the board or boards as detailed above in steps 2 through 6 of "Installing a Communications Expansion Board".

6. Replace and hand tighten the six nuts.

7. Close the computer cabinet.

CLOSING THE COMPUTER CABINET

To close the computer cabinet:

1. Swing the hinged portion of the cabinet up until the lever lock levers on the hinged portion are touching the locks on the fixed portion.

2. Slowly force the cabinet together until the lever locks at the front and back snap closed.

COMPUTER'S INPUT/OUTPUT PANEL

At the rear of the computer is an input/output panel. Figures 2-4, 2-5, and 2-6 illustrate the input/output panels for the three types of Communications Expansion boards.

This and the next two drawings will be drawn by an artist during production. They will be like this with labels on the connectors.

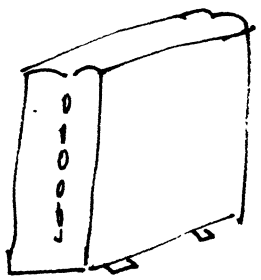


Figure 2-4. Input/Output Panel with Ethernet Board

Figure 2-5. Input/Output Panel with RS-232-C Board

Figure 2-6. Input/Output Panel with Ethernet/RS-232-C Board

SECTION 3: ARCHITECTURE

This section contains three parts:

- Functional description
- Registers
- Interface connectors.

FUNCTIONAL DESCRIPTION

Input/output expansion architecture contain four main functional elements:

- Interrupt controller

  Interrupt arbiter
  Interrupt acknowledge decoder
  Ethernet interrupt vector generator.

- Address decoder and interface

  Address buffer
  Data buffer
  Expansion address decoder.

- RS-232-C control

  Four 8274 controllers
  Eight discrete port registers
  Three 8253 timers.

- Ethernet control

  7990 Ethernet controller
  7991 Serial interface adapter
  Ethernet address decoder
  Ethernet state machine
  DMA state machine
  Ethernet DMA controller.

Table 3-1 describes the buses on the input/output expansion boards.

Table 3-1. Input/Output Expansion Buses

Name	Acronym	Function
Address (Ethernet)	A16-23	The 7990 outputs the upper DMA address to the Ethernet DMA address register on this bus.
Communications Address	CA0-15	Decoded from the logical address (LA) bus when the CPU reads or writes an expansion register.
Data/Address Lines	DAL0-15	During a CPU register read or write, the 7990 outputs register data on this bus. During Ethernet DMA, the 7990 first outputs the memory address on this bus (and on A16-23), then reads or writes the memory data on this bus.
Expansion Data	XD0-15	Three functions: (1) routes register and DMA data between the Ethernet Holding Register and the CPU or memory, (2) routes, via the communications data (CD) bus, the register data byte (lower) or the interrupt vector byte between the 8274s and the CPU, and (3) routes the Ethernet interrupt vector byte from the Ethernet Vector Register) to the CPU.
Logical Address	LA0-15	Address bus from the main board. It goes to the expansion address decoder, the 8274 and 8253 chip address inputs, and the Ethernet address decoder, all via the communications address (CA) bus. It also interfaces the Ethernet DMA Address Register and interrupt status ring pointer to the main board memory control.

Interrupt Controller

The interrupt controller contains:

- o Interrupt arbiter
- o Interrupt acknowledge decoder
- o Ethernet interrupt vector generator.

The interrupt controller handles RS-232-C interrupts from the expansion 8274s and Ethernet interrupts from the Ethernet control. The 8274 on the main board and the expansion interrupts all have CPU interrupt priority level 5. Within level 5, the 8274 on the main board has a higher priority, and the expansion 8274 and Ethernet share a lower priority.

The interrupt arbiter is a PAL implemented mod-five counter. It executes a five window loop, where it successively enables interrupts from each of the expansion 8274s or the Ethernet control, at each clock enabling a different interrupt signal.

When an expansion 8274 or Ethernet interrupts the CPU, the interrupt arbiter sends the expansion board interrupt signal to the main board interrupt control circuitry. The arbiter freezes, then waits for the CPU to respond with a interrupt priority level five interrupt acknowledge cycle.

When the main board's interrupt control receives the expansion interrupt signal, if the CPU has not masked out level five interrupts and if no higher priority interrupt is asserted, the interrupt control generates a CPU interrupt.

When the CPU acknowledges the level five interrupt, the interrupt control on the main board sends an acknowledge pulse to the expansion board. The interrupt acknowledge decoder multiplexes the pulse to the device that originally generated the interrupt.

RS-232-C Interrupt Acknowledge

The interrupt controller receives the first CPU acknowledge pulse and routes it to the 8274 pointed to by the interrupt arbiter. The 8274 prepares the interrupt vector byte. Shortly later, the main board interrupt control sends another interrupt acknowledge pulse. The 8274 outputs the vector byte. The interface/address control routes the byte to the CPU.

Ethernet Interrupt Acknowledge

Ethernet control interrupts the CPU indirectly, via the Ethernet state machine. During the interrupt ackowledge cycle, the 799Ø

Ethernet controller does not supply the vector byte to the CPU. Instead, the interrupt controller enables the Ethernet interrupt vector generator, a set of latches that the CPU loads during initialization. The latches supply the vector byte to the CPU.

Interrupts Disabled by the Main 8274

When the main board 8274 is interrupting the CPU, it flags the expansion board. Unless the expansion board has already interrupted the CPU and has received the first CPU acknowledge pulse, the flag disables expansion interrupts until the CPU completes the interrupt acknowledge cycle to the main board 8274, and the main 8274 removes the flag.

Address Decoder and Interface

The CPU address decoder, on the main board, enables the expansion address decoder when the CPU outputs an expansion register address. The expansion address decoder decodes the register select portion of the CPU address. If the CPU addresses an expansion RS-232-C register, the decoder outputs chip select and read/write signals to the addressed RS-232-C chip.

If the CPU addresses an Ethernet register, the decoder outputs a select signal to the Ethernet decoder. The Ethernet decoder does a further decode and itself outputs the Ethernet register read/write signals.

RS-232-C Control

Four 8274 RS-223C controllers provide two RS-232-C channels apiece, via their A- and B-ports. Three 8253 timers provide three clocks apiece. Each channel thus has its own baud rate generator with one clock is unused. Table 3-2 lists the computer's ten RS-232-C channel assignments, including the two on the main board.

Table 3-2. RS-232-C Channel Assignment

Chip	Port-A	Port-B
Main 8274	Channel Ø	Channel 1
Expansion 8274-1	Channel 2	Channel 3
Expansion 8274-2	Channel 4	Channel 5
Expansion 8274-3	Channel 6	Channel 7
Expansion 8274-4	Channel 8	Channel 9

Associated with each 8274 are two discrete registers, the Port Status Register and the Secondary Transmit/Clock Select Register. The 8274s perform all phases of communications control, including serial data-in/data-out, serial to parallel conversion, parallel to serial conversion, port status update, and CPU interrupt.

All RS-232-C interaction with the main board occurs under program control; when an 8274 needs to transfer data to or from memory, it interrupts the CPU. The CPU decodes the interrupt and itself does the transfer by reading memory and writing the 8274 data register or reading the 8274 data register and writing memory.

The CPU reads and writes the internal 8274 registers, the discrete 8274 registers, and the 8253 registers, via slow cycle accesses. Slow cycle accesses are described in detail in Section 6 of the MiniFrame Hardware Manual. "RS-232-C Register Reads and Writes", in Section 4 of this manual, summarizes RS-232-C register accesses.

NOTE

A 72Ø1 controller performs identical functions as the 8274 and may be substituted for an 8274.

Ethernet Control

Ethernet control requires CPU intervention only to monitor the beginning or end of packet data transfers or to respond to a data transfer error. The control allows the CPU a service latency of up to 256 interrupts. It contains:

- o AM799Ø local area network controller for Ethernet
- o AM7991 serial interface adapter
- o Ethernet coax transceiver
- o Ethernet state machine
- o DMA controller
- o DMA state machine.

The 799Ø and 7991 together form the Ethernet node. The two state machines control interface between the 799Ø and the DMA controller and the main board. The Ethernet state machine controls CPU accesses to the Ethernet registers and the 799Ø side of a 799Ø/CPU interrupt. The DMA state machine handles memory accesses by the 799Ø and the memory side of a 799Ø/CPU interrupt.

Ethernet Node Operation

At any time, the Ethernet node functions, with respect to the local area network, in either transmit mode or receive mode. When idle, the node is in receive mode.

When the CPU shifts the node to transmit mode, the 7990 initiates a memory DMA read cycle. The 7990 prefaces the read data with a preamble and sync patter, then calculates and appends the data with a 32-bit CRC field. The 7990 then serially transmits the data to the 7991. The 7991 takes the data and outputs the Manchester encoded TRANSMIT- and TRANSMIT+, which drive the transceiver cable. The transceiver outputs the signals to the Ethernet coax cable.

When the node is in receive mode, and a carrier is present on the Ethernet coax, the transceiver creates the inputs RECEIVE- and RECEIVE+. The Manchester decoder in the 7991, synchronized by the 7991's phase locked loop, recovers the clock and data and supplies them to the 7990 at TTL levels. While it is receiving data, the 7991 asserts CARRIER PRESENT, to tell the 7990 that clock and data are available. The 7990 calculates a CRC on the incoming data stream and compares it with the CRC appended to the stream. If the CRCs don't compare, the 7990 generates a packet error and interrupts the CPU.

Ethernet Interrupts

When the 7990 asserts its interrupt line, the Ethernet state machine does the following:

1. Fields the 7990 interrupt.
2. Reads and latches the controller's Interrupt Status Register (ISR)
3. Writes a 7990 register to clear the 7990 interrupt flag.
4. Resumes operation.

The DMA state machine logs the interrupt in memory by writing the stored ISR contents into the interrupt status ring, a system defined buffer located in memory. The machine writes the ring location that the ISR pointer, a discrete Ethernet control register, points to.

The DMA state machine increments the ISR pointer and sends an interrupt request to the interrupt control. The interrupt controller alerts the CPU that an interrupt has occurred. The CPU may or may not service the interrupt at that time. The ISR ring is 256 words long, allowing the 7990 to nest 256 interrupts.

The 7990 interrupts to the CPU for:

- Completion of initialization
- Reception of a packet
- Transmission of a packet
- Transmitter timeout
- CRC receive, babbling, missed packet, or memory errors.

Ethernet DMA

The 7990 reads and writes memory as if it were the only system bus master. However, since the system has other bus masters, the 7990 requires a arbitration, control, and data buffer. The DMA state machine and memory address and data interface provide the buffer.

Initially, the 7990 loads the read or write address into the DMA address register latches. Next the 7990 notifies the DMA state machine that it has output the address. The DMA state machine sends an expansion DMA request to the Main Processor board's bus arbiter. When the arbiter honors the request, the machine tells the 7990 that memory is ready for the transfer.

The 7990 outputs the write data or inputs the read data, then negates the DMA request. The DMA state machine ends the transfer by negating the expansion DMA request.

7990 Register Accesses

At system initialization, the CPU sets up the 7990 by doing register writes to the four Control and Status Registers (CSR 0-3). The CPU very rarely reads the 7990 status registers; it does not need to. (As "Ethernet DMA", above, described, the Ethernet state machine automatically logs the Interrupt Status Register contents in memory after the 7990 generates a CPU interrupt.) However, the CPU can read the 7990 registers, but only when the 7990 is not enabled to send or receive data.

To write a register, the CPU gains bus control and outputs the register address. The Main Processor board's CPU address decoder outputs a select signal to the expansion address decoder. The expansion address decoder outputs an address field to the Ethernet address decoder, which then outputs the register write control signals to the 7990. The Main board interface routes the data to the 7990. A register read is similar, except the 7990 routes the data to the CPU.

REGISTERS

The CPU loads an expansion board register by executing a programmed slow cycle input/output instruction. At roughly 300 ns into the slow cycle, the CPU address decoder decodes $C7XXXX and asserts XSEL-. XSEL- enables the secondary address decode logic on the expansion board.

NOTE

Throughout this manual, the character "$" indicates the beginning of a hexidecimal address. Also, when any "X" appears in an address (including binary) the character (or bit) represented by the X can be any acceptable hexidecimal character (or either binary 1 or 0).

As Table 3-3 shows, the expansion address decoder decodes bits 15-12 of the CPU address to select either an 8274 reset, an 8274 controller, an 8253 timers, or an Ethernet register.

Table 3-3. Expansion Address Decode

Hex Address	A15-12	Function
$C7XXXA	XXXX	Reset all 8274 channels A and B.
$C70XXX	0000	Select 8274-1 channels A and B.
$C71XXX	0001	Select 8274-2 channels A and B.
$C72XXX	0010	Select 8274-3 channels A and B.
$C73XXX	0011	Select 8274-4 channels A and B.
$C74XXX	0100	Select 8253-1.
$C75XXX	0101	Select 8253-2.
$C76XXX	0110	Select 8253-3.
$C78XXX	1000	Select Ethernet registers.

The supervisor programs the 8274s to run in interrupt mode, supplying them with internal interrupt vectors for 68010 interrupt processing. The main board 8274, expansion 8274s, and the Ethernet 7990 all have interrupt priority level five. The supervisor uses the interrupt priority input/output feature to prioritize interrupts between the main board 8274 and an expansion interrupt, as follows:

- When the main board 8274 interrupts the CPU, the main 8274 negates its the interrupt priority output (IPO)

signal. The IPO signal disables expansion interrupts until the CPU exectutes a level five interrupt acknowledge cycle, and the 8274 reasserts the IPO signal.

- The main 8274 interrupt may still disable the expansion interrupt controller, even though the expansion board has interrupted the CPU, and the CPU has begun a level five interrupt acknowledge cycle. The main 8274 interrupt prevails if the main board interrupt control hasn't yet generated the first interrupt acknowledge pulse (which the CPU does at roughly 500 nanoseconds into the interrupt acknowledge cycle). The acknowledge pulse disables the 8274 IPO/IPI hardware.

  When IPO disables an expansion interrupt, the expansion interrupt controller holds the interrupt signal asserted. The main board's interrupt controller reprioritizes the interrupt and generates another interrupt priority level five interrupt when applicable.

RS-232-C Registers

There are seven groups of RS-232-C registers. Four groups center around the the four 8274 controller chips. The other three groups are in the three 8253 programmable interval timer chips.

8274 (Including External Discrete) Registers

Tables 3-4 through 3-7 describe the 8274 registers. Registers having 8274 in parenthesis are internal to the 8274 chip. The other registers are discrete support registers. Refer to an 8274 data sheet for a description of the internal 8274 registers.

Table 3-4. 8274-1 Registers

Address	Register	Type
$C70000	Port A data (8274 register)	Read/write
$C70002	Port B data (8274 register)	Read/write
$C70004	Port A command (8274 register)	Write only
$C70006	Port B command (8274 register)	Write only
$C70008	Secondary transmit/clock select	Write only
$C70008	Port status	Read only
$C7X00A	Reset (Initialize 8274)	Write only

Table 3-5. 8274-2 Registers

Address	Register	Type
$C71000	Port A data (8274 register)	Read/write
$C71002	Port B data (8274 register)	Read/write
$C71004	Port A command (8274 register)	Write only
$C71006	Port B command (8274 register)	Write only
$C71008	Secondary transmit/clock select	Write only
$C71008	Port status	Read only
$C7X00A	Reset (Initialize 8274)	Write only

Table 3-6. 8274-3 Registers

Address	Register	Type
$C72000	Port A data (8274 register)	Read/write
$C72002	Port B data (8274 register)	Read/write
$C72004	Port A command (8274 register)	Write only
$C72006	Port B command (8274 register)	Write only
$C72008	Secondary transmit/clock select	Write only
$C72008	Port status	Read only
$C7X00A	Reset (Initialize 8274)	Write only.

Table 3-7. 8274-4 Registers

Address	Register	Type
$C73000	Port A data (8274 register)	Read/write
$C73002	Port B data (8274 register)	Read/write
$C73004	Port A command (8274 register)	Write only
$C73006	Port B command (8274 register)	Write only
$C73008	Secondary transmit/clock select	Write only
$C73008	Port status	Read only
$C7X00A	Reset (Initialize 8274)	Write only

Tables 3-8 and 3-9 describe the Secondary Transmit/Clock Select Register and the Port Status Register, respectively. Figures 3-1 and 3-2 are register summaries.

NOTE: In each channel the Secondary Transmit/Clock Select and Port Status Registers have the same address. The Secondary Transmit/Clock Select Register is addressed by a CPU write only, and the Port Status Register is addressed by a CPU read only.

D7	D6	D5	D4	D3	D2	D1	DØ
X	X	X	X	STDA	STDB	SELA	SELB

Figure 3-1. Secondary Transmit/Clock Select Summary

Table 3-8. Secondary Transmit/Clock Select Register

Bit	Name	Function
7-4	Not used.	These bits are not used.
3	Secondary transmit data A (STDA)	This bit associates with the A port and drives pin 14 of the RS-232-C connector. The CPU can write data to an external source via this bit.
2	Secondary transmit data B (STDB)	This bit associates with the B channel and drives pin 14 of the RS-232-C connector. The CPU can write data to an external source via this bit.
1	Select clock A (SELA)	This bit selects between internal and external clocks for port A. SELA = Ø selects the internal clock. SELA = 1 selects the external clock. When the internal clock is selected, the same clock is supplied to the chip for both transmit and receive clocks.
Ø	Select clock B (SELB)	This bit selects between internal and external clocks for port B. SELB = Ø selects the internal clock. SELB = 1 selects the external clock. When the internal clock is selected, the same clock is supplied to the chip for both transmit and receive clocks.

D7	D6	D5	D4	D3	D2	D1	DØ
RIA	DSRA	SRDA	RIB	DSRB-	SRDB-	FCC+	Ø

Figure 3-2. Port Status Register Summary

Table 3-9. Port Status Register

Bit	Name	Function
7	Ring (RIA) indicator A	This is the ring indicator from channel A.
6	Ring (RIB) indicator B	This is the ring indicator from channel B.
5	Data (DSRA-) set ready A	This is the data set ready indication from channel A. DSRA = Ø indicates data is ready. DSRA = 1 indicates data is not ready.
4	Data (DSRB-) set ready B	This is the data set ready indication from channel B. DSRB = Ø indicates data is ready. DSRB = 1 indicates data is not ready.
3	Secondary received data A (SRDA)	This bit associates with the A channel and is driven from pin 16 of the RS-232-C connector. The CPU can read data from an external source via this bit.
2	Secondary received data B (SRDB)	This bit associates with the B channel and is driven from pin 16 of the RS-232-C connector. The CPU can read data from an external source via this bit.
1-Ø	Wired high.	These signals are wired high. They allow the CPU to determine there is a Communications Expansion Board or Combination Expansion Board in the system.

8253 Timer Registers

The 8253 timer chip has three counters in it. The input to counters Ø, 1, and 2 is a 1,228,8ØØ-Hz clock. The frequency divides to obtain the range of internally generated baud rates, from 19,2ØØ baud to 11Ø baud, using either 16X or 1X clocks. The control word selects the clock divisor. Table 3-1Ø lists the 8253 timer registers. Refer to Table 3-1 for the channel assigments.

Table 3-1Ø. 8253 Timer Baud Rate Registers

Address	Chip	Register	Function
$C74ØØØ	8253-1	Counter Ø	Baud rate for channel 3.
$C74ØØ2	8253-1	Counter 1	Baud rate for channel 4.
$C74ØØ4	8253-1	Counter 2	Baud rate for channel 5.
$C74ØØ6	8253-1	Control word	8253-1 control word.
$C75ØØØ	8253-2	Counter Ø	Baud rate for channel 7.
$C75ØØ2	8253-2	Counter 1	Baud rate for channel 8.
$C75ØØ4	8253-2	Counter 2	Baud rate for channel 6.
$C75ØØ6	8253-2	Control word	8253-2 control word.
$C76ØØØ	8253-3	Counter Ø	Baud rate for channel 9.
$C76ØØ2	8253-3	Counter 1	Baud rate for channel 1Ø.
$C76ØØ4	8253-3	Counter 2	Not used.
$C76ØØ6	8253-3	Control word	8253-3 control word.

Ethernet Registers

Table 3-11 describes the Ethernet registers.

Table 3-11. Ethernet Registers

Address	Function
$C78000	The CPU writes this address to reset the Ethernet hardware, just as a power-on or hard reset does. PAL 270 asserts ERESET-.
$C7800E	The CPU writes this address to load the interrupt status ring base address. PAL 270 asserts LTCHSTADR-, and the 8-latch wide 373 chip latches the address data from the CPU.
$C78010	The CPU writes this address to load the 7990 write only Register Address Port (RAP). PAL 274 controls the 7990 access.
$C78012	The CPU writes this address to load the 7990 write only Register Data Port (RDP). PAL 274 controls the 7990 access.
$C78014	The CPU writes this address to fetch data from the RAP and stores it in the Ethernet output holding register. PAL 274 controls the 7990 access, and PAL 273 causes the Ethernet output holding register to latch the data.
$C78016	The CPU writes this address to fetch data from the RDP and stores it in the Ethernet output holding register. PAL 274 controls the 7990 access, and PAL 273 causes the Ethernet output holding register to latch the data.
$C78018	The CPU reads this address to read the data from the Ethernet output holding register. PAL 273 causes the Ethernet output holding register to output the data.

INTERFACE CONNECTORS

This section contains four tables:

- Tables 3-12 and 3-13 lists the pin and signal assignments of the RS-232-C A-port J3 connector and B-port J4 connector for 8274-1. The pin and signal assignments of the connectors for the other three 8274s are the same; except the connector number changes, as follows:

  8274-2: A-port connector J5, B-port connector J6
  8274-3: A-port connector J7, B-port connector J8
  8274-4: A-port connector J9, B-port connector J10

- Table 3-14 lists the pin and signals assignments of the J20 expansion board/main board connector, which carries the MiniFrame System Bus.

- Table 3-15 lists the pin and signal assignments of the Ethernet connector.

Table 3-13. 8274 A-Port Connector (J-odd) Pin Assigments

Pin	Signal	Input/Output	Pin	Signal	Input/Output
2	TXDA	Output	14	STDA	Output
3	RXDA	Input	15	TXCA	Input
4	RTSA	Output	16	SRDA	Input
5	CTSA	Input	17	RXCA	Input
6	DSRA	Input	20	DTRA	Output
7	GND		22	RIA	Input
8	CDA	Input	24	CLK OUT A	Output

Table 3-14. 8274 B-Port Connector (J-even) Pin Assigments

Pin	Signal	Input/Output	Pin	Signal	Input/Output
2	TXDB	Output	14	STDB	Output
3	RXDB	Input	15	TXCB	Input
4	RTSB	Output	16	SRDB	Input
5	CTSB	Input	17	RXCB	Input
6	DSRB	Input	20	DTRB	Output
7	GND		22	RIB	Input
8	CDB	Input	24	CLK OUT B	Output

Table 3-14. MiniFrame System Bus J2Ø Connector

Pin	Signal	Pin	Signal	Pin	Signal
1	+5 Volts	35	Ground	69	LA3
2	+5 Volts	36	T17Ø+	7Ø	LA4
3	Ground	37	XGNT-	71	LA5
4	XD8	38	CASACK-	72	Ground
5	XD9	39	XDMAREQ-	73	LA6
6	XD1Ø	4Ø	Ground	74	LA7
7	XD11	41	XIN-	75	Ground
8	Ground	42	PS+	76	LA8
9	XD12	43	Ground	77	LA9
1Ø	XD13	44	ROE-	78	LA1Ø
11	Ground	45	XDØ	79	LA11
12	XD14	46	XD1	8Ø	Ground
13	XD15	47	XD2	81	LA12
14	INTA8274-	48	Ground	82	LA13
15	IRQ8274B-	49	XD3	83	Ground
16	Ground	5Ø	XD4	84	LA14
17	XIRQ-	51	Ground	85	LA16
18	XINTA-	52	XD5	86	LA15
19	Ground	53	XD6	87	LA17
2Ø	LWT-	54	XD7	88	Ground
21	NMI-	55	MEIPE-	89	LA18
22	LMA2Ø	56	Ground	9Ø	LA19
23	LMA19	57	IPO-	91	Ground
24	Ground	58	CASINH-	92	LA2Ø
25	LMA18	59	Ground	93	LA2Ø
26	LMA17	6Ø	LLDS-	94	LA2
27	Ground	61	LUDS-	95	+5 VoLts
28	LMA16	62	XSEL-	96	Ground
29	LMA15	63	REFRAS-	97	+5 Volts
3Ø	LMA14	64	Ground	98	LA1
31	LMA13	65	ENCAS-	99	+12 Volts
32	Ground	66	RAS-	1ØØ	-12 Volts
33	LMA12	67	Ground		
34	RESET-	68	PCLK+		

Table 3-15. Ethernet J2 Connector

Pin	Signal	Pin	Signal	Pin	Signal
1	GND	6	GND	11	Not used
2	COL+	7	Not used	12	RCVR-
3	TRANS+	8	Not used	13	PLUS12
4	Not used	9	COL-	14	Not used
5	RCVR+	1Ø	TRANS-	15	Not used

SECTION 4: THEORY OF OPERATION

The theory of operation describes the input/output expansion boards' circuitry and operation.

NOTE

The Combination Expansion Board has 16 sheets of schematics, the Communications Expansion Board has 12 sheets of schematics, and the Ethernet Expansioin Board has 8 sheets of schematics. Sheets 1-4 of all three sets show the address decoder and interface and the interrupt controller. Sheets 5-12 of both the Communications Expansion Board's schematics and the Combinations Expansion Board's schematics show the RS-232-C control. Sheets 13-16 of the Combinations Expansion Board's schematics and sheets 5-8 of the Ethernet Expansion Board's schematics show the Ethernet control. In this manual, Ethernet hardware references list the Combination Expansion Board's schematics sheet number first, followed by a semi-colon, then the Ethernet Expansion Board's schematics sheet number (for example: sheet 13; 5). Appendix A includes all three sets of schematics.

INTERRUPT CONTROLLER (SHEET 3)

The interrupt controller centers around the

- Interrupt arbiter (PAL 60)
- Interrupt acknowledge decoder (PAL 61).

The interrupt arbiter successively enables one each of the 8274s interrupt signals (SINT1-, SINT2-, SINT3-, and SINT4-) and the 7990 interrupt signal (INTE-). At each XCLK-, if the interrupt signal that is enabled is not asserted, the arbiter increments a binary field (Y1-3), which opens the next window in the loop.

The loop sequence, illustrated in Figure 4-1, is as follows:

- State IØ (Y1-3 = ØØØ); looks at SINT1-
- State I1 (Y1-3 = ØØ1); looks at SINT2-
- State I3 (Y1-3 = Ø11); looks at SINT3-
- State I2 (Y1-3 = Ø1Ø); looks at SINT4-
- State I4 (Y1-3 = 1ØØ); looks at INTE-
- Back to state IØ.

NOTE

The interrupt arbiters on the RS-232-C board and the Ethernet board execute the above loop, but with some inputs inactive. On the RS-232-C board, the arbiter is idle every fifth clock and active otherwise; while on the Ethernet board, the arbiter is active every fifth clock and idle otherwise.

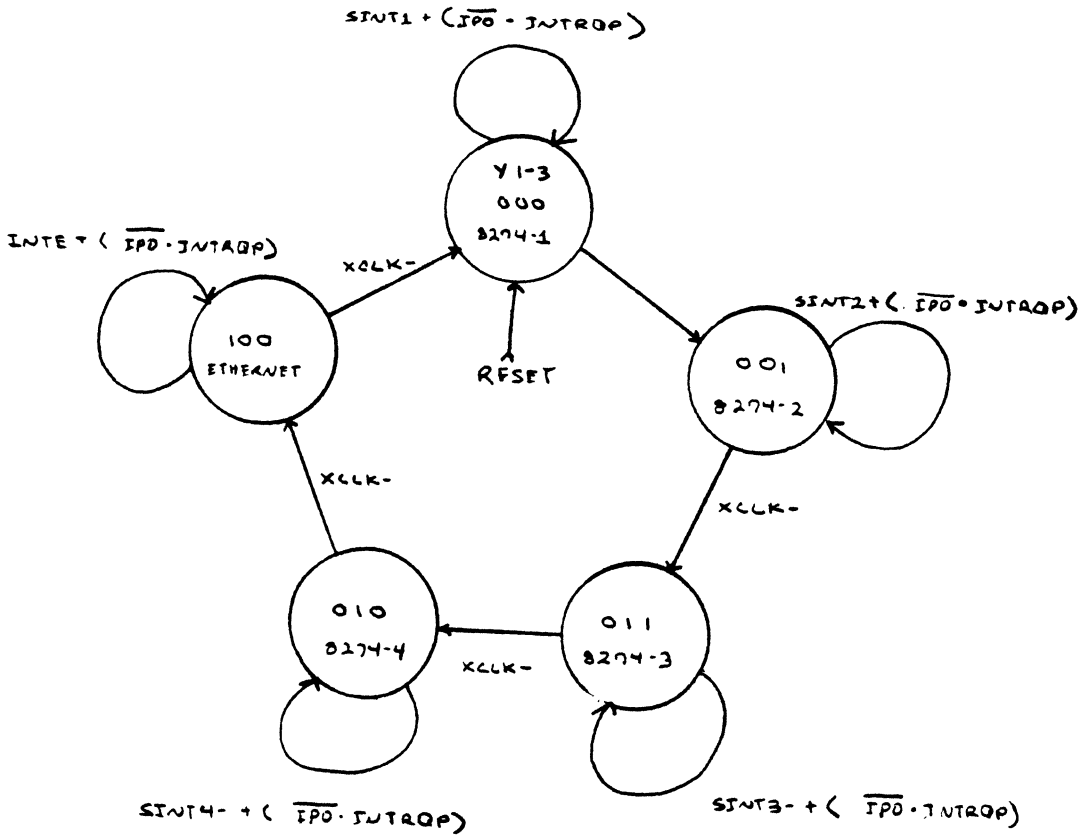


Figure 4-1. Interrupt Controller State Sequencing Diagram

When a device asserts its interrupt signal, the interrupt arbiter asserts INTRQP- and freezes the Y field. INTRQP- generates INTRQ. INTRQ sends 8274IRQ- to the interrupt control on the main board. If main interrupt control is enabled for priority level five interrupts, and no higher priority interrupt is active, the interrupt control sends the level five interrupt field to the CPU.

When the CPU returns a level five interrupt ackowledge (it may not happen immediately, as program control can internally mask out the interrupt), the main interrupt acknowledge decoder asserts INTA8274-. INTA8274- goes to the main 8274 and to the expansion interrupt acknowledge decoder (PAL 61). Descriptions of the RS-232-C interrupt acknowledge cycle and the Ethernet interrupt acknowledge cycle follow.

RS-232-C Interrupt Acknowledge

When the main board asserts INTA8274- (250 nanoseconds into the interrupt acknowledge cycle), the interrupt arbiter asserts VEC8274-. VEC8274- and the asserted ACK- signal together activate the appropriate 8274 INTA- chip acknowledge signal.

NOTE

Before the INTA- signal pulses, the main 8274 can break into the the expansion 8274 interrupt cycle by asserting its CPU interrupt line and negating its IPO- line. IPO- is ORred to the IPI- input of the expansion 8274 and causes them to disable their interrupt lines and ignore any acknowledge pulses. IPO- keeps the arbiter frozen by keeping INTRQP- asserted. When the CPU has serviced the main 8274, the main 8274 reasserts IPO-, which allows the expansion 8274s to again assert their interrupt lines. The interrupt arbiter reasserts 8274IRQ- and waits for the CPU to execute another level five interrupt acknowledge cycle. When INTA8274- pulses, it disables the 8274 internal IPO- hardware, thus guarenteeing the completion of the expansion acknowledge cycle.

At 500 nanoseconds, the main board negates INTA8274-. The interrupt arbiter negates VEC8274-, and the expansion INTA- signal (that corresponds to Y1-3) goes high. One clock later CTRLINTA- causes the interrupt arbiter to reassert VEC8274-. VEC8274- modifies the INTA8274- waveform by pulsing the INTA-

signal low after 100 nanoseconds rather than after 250 nanoseconds. (See Figure 4-2.) The faster pulse gives the interrupting chip time to output the vector byte.

At 750 nanoseconds, the main board reasserts INTA8274-. CTRLINTA- and INTA8274- generate INTAOE-. The interrupt acknowledge PAL asserts DATOE- and XIN-, which enables the data interface. The interface routes the vector byte to the main board. Figure 4-2 illustrates the RS-232-C interrupt cycle timing.

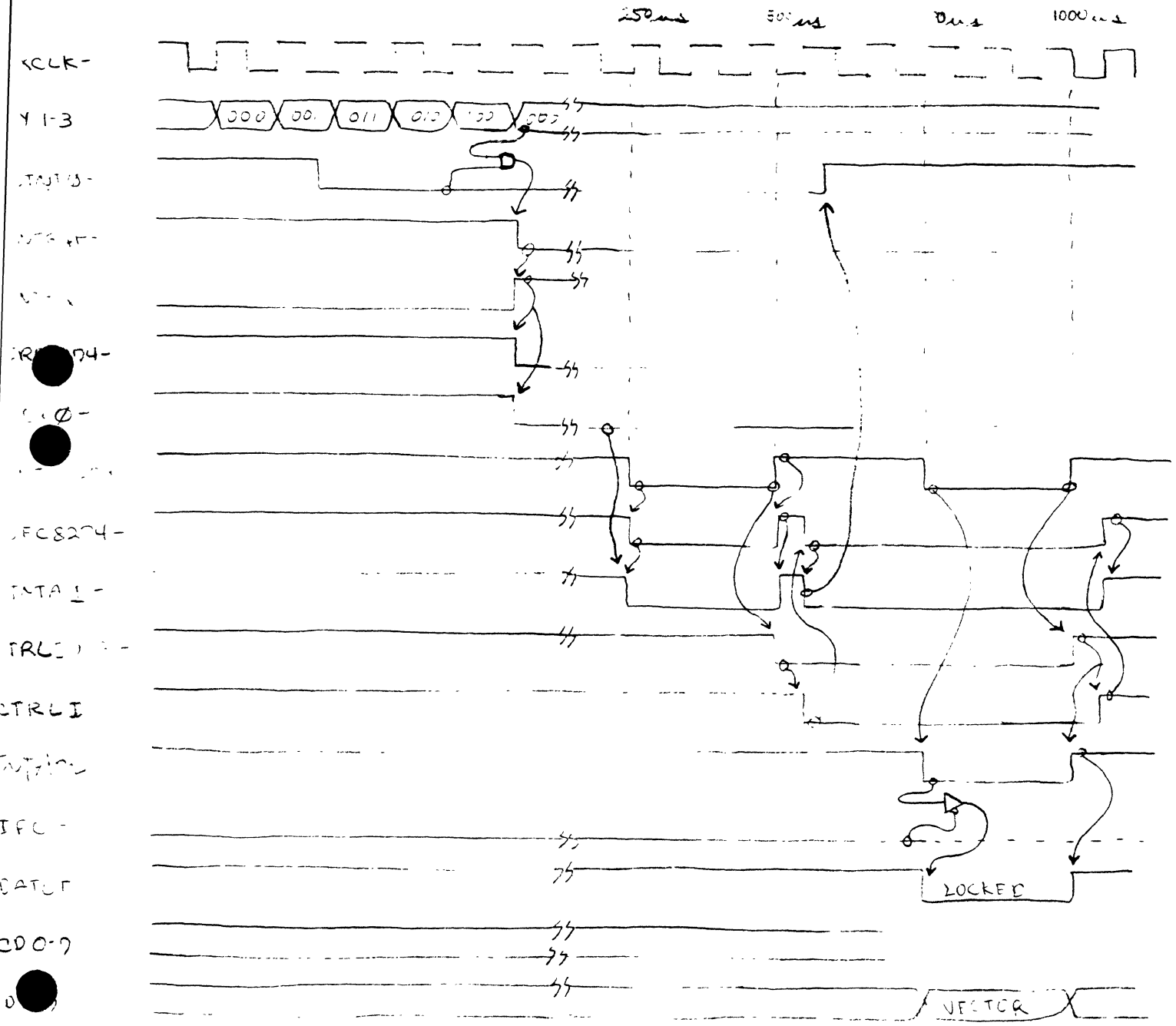


Figure 4-2. RS-232-C Interrupt Cycle Timing

Ethernet Interrupt Acknowledge

During an Ethernet interrupt acknowledge cycle, the 799Ø is not directly involved in the acknowledge portion of the interrupt cycle. Instead, the Ethernet state machine acknowledges the 799Ø after it receives the acknowledge. The 799Ø thus continues input/output operation.

At roughly 25Ø nanoseconds into the cycle, the main board interrupt control asserts INTA8274-. At 5ØØ nanoseconds the main board negates INTA8274-. The positive going edge of INTA8274- clocks a latch, which asserts CTRLINTA-.

At 75Ø nanoseconds the main board interrupt control reasserts INTA8274-. CTRLINTA- and INTA8274- drive INTAOE- low. INTAOE- and ACKE- generate VECTOROE-, which enables the Ethernet interrupt vector latches. The latches output the Ethernet vector byte. (The CPU loads the latches during system initialization.)

NOTE

Before the main board pulses INTA8274-, the main 8274 can cut into the Ethernet interrupt acknowledge by asserting its CPU interrupt signal and negating its IPO- line. IPO- disables ACKE-, which stops the expansion interrupt controller from outputting the Ethernet vector byte. IPO- also keeps the arbiter frozen by holding INTRQP- asserted. When the CPU has serviced the main 8274, the main 8274 reasserts IPO-. The interrupt arbiter reasserts 8274IRQ- and waits for the CPU to execute another level five interrupt acknowledge cycle. When INTA8274- pulses, it disables the 8274 internal IPO- hardware, thus guarenteeing the completion of the expansion acknowledge cycle.

1ØØØ nanoseconds into the Ethernet acknowledge cycle, INTAOE- clocks the Ethernet acknowledge flip/flop. The flip/flop outputs INTECLR-. INTECLR- and ACKE- send INTAE- to the Ethernet control. At the next clock, the Ethernet state machine negates INTE-, which negates INTRQP-, INTRQ, 8274IRQ-, and ACKE-. INTRQ clears INTCLR-. ACKE- clears INTAE-. Figure 4-3 illustrates the timing relationships between the interrupt signals.

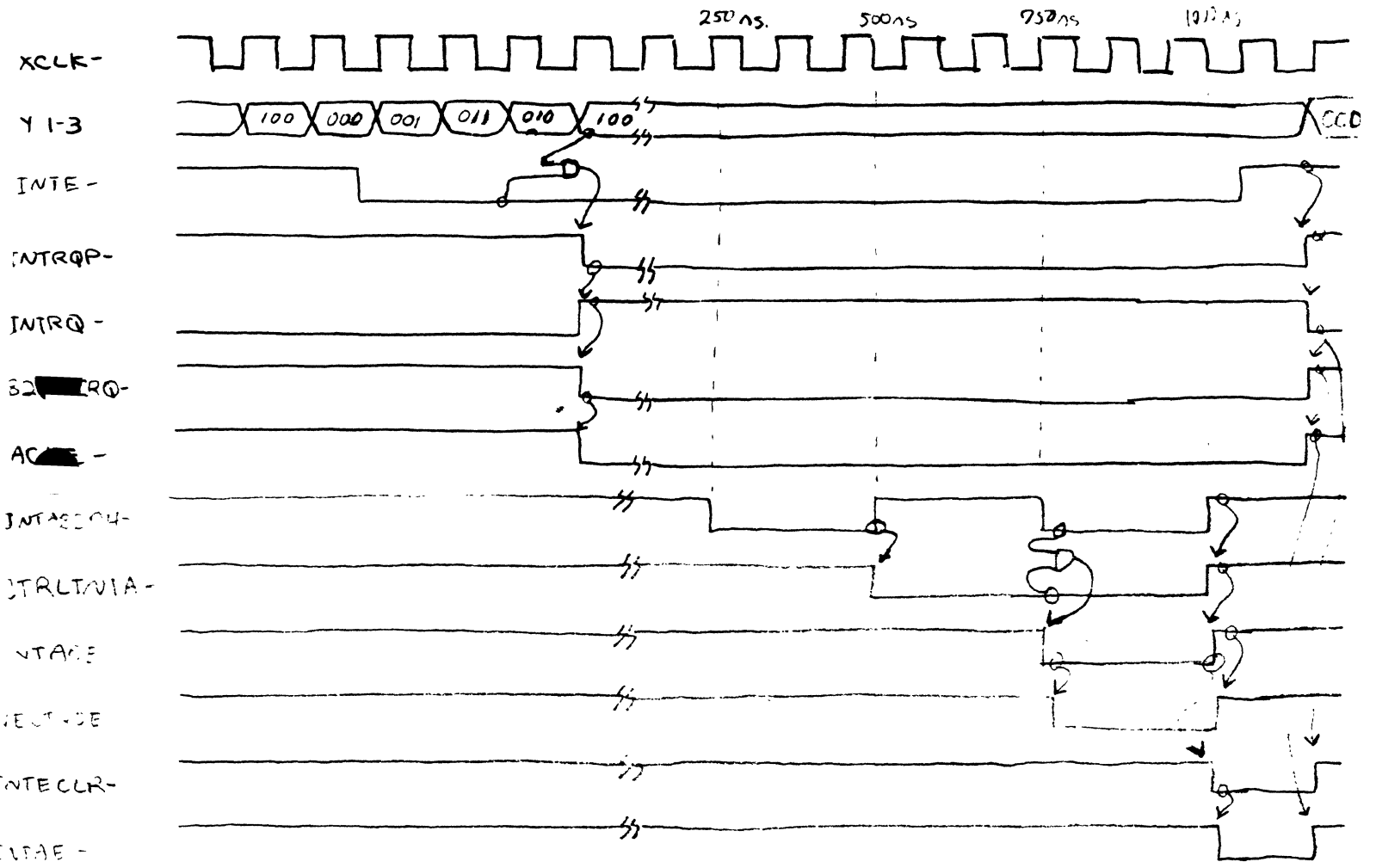


Figure 4-3. Ethernet Interrupt Cycle Timing

INTERFACE/ADDRESS DECODE (SHEETS 2 AND 4)

The interface/address decode receives and decodes a portion of the logical address and read/write control signals from the main board. It also routes data between the main board and the Ethernet control or RS-232-C control. It contains

- o Control buffer (sheet 2)
- o Address buffer (sheet 4)
- o Expansion address decoder (sheet 4)
- o Communications data buffer (sheet 4).

The control buffer receives read/write, interrupt, and clock signals from the main board and routes them to the rest of the expansion board for register accesses. The address buffer receives the register address, via the logical address bus (LA1-4 and LA12-15) on the main board, and routes it to the communciations address bus (CA1-4 and CA12-15).

The address decoder, when enabled by CSEL+ (from the control buffer), decodes CA1-4 and CA12-15. It outputs chip select signals to the RS-232-C circuitry and an Ethernet select signal to the Ethernet address decoder (sheet 15). The communications data buffer routes the lower byte of data to or from the RS-232-C register or Ethernet interrupt vector latches.

Table 4-1 lists the decode of the expansion address.

Table 4-1. Expansion Address Decode

Hex Address	A15-12	Function
$C7XXXA	XXXX	Reset all 8274 channels A and B.
$C70XXX	0000	Select 8274-1 channels A and B.
$C71XXX	0001	Select 8274-2 channels A and B.
$C72XXX	0010	Select 8274-3 channels A and B.
$C73XXX	0011	Select 8274-4 channels A and B.
$C74XXX	0100	Select 8253-1.
$C75XXX	0101	Select 8253-2.
$C76XXX	0110	Select 8253-3.
$C78XXX	1000	Select Ethernet registers.

X= Don't Care.

RS-232-C CONTROL (SHEETS 5-12)

RS-232-C interaction with the main board occurs via RS-232-C interrupts to the CPU and via CPU reads or writes of the RS-232-C registers.

The RS-232-C circuitry contains

- Four 8274 chips (sheets 5-8), each controlling two channels
- Four 8-bit Port Status Registers (Sheets 5-8), one per 8274
- Four Secondary Transmit/Clock Select Registers (PAL 264, sheet 9; PAL 266, sheet 10; PAL 267, sheet 11; and PAL 265, sheets 12).
- Three 8253 programmable interval timers (sheets 9-11) each with three outputs, giving individual baud rates per channel
- Channel interface circuitry.

The hardware is described here in the context of overall RS-232-C operation. Section 6 in the MiniFrame Hardware Manual provides timing diagrams for the CPU slow cycle register accesses.

The communications data bus, CD7-Ø, buffered off the external data bus (XD7-Ø sheet 4) from the main board, connects the 8274s and 8253s to the CPU. During a register read or write, after the CPU outputs the register address, the Main board CPU address decoder asserts XSEL-, and the expansion board buffers XSEL- to CSEL+.

CSEL+ enables the address decoders (sheet 4) which output either the read/write signal for the addressed discrete register or the chip select and read or write signal for the addressed 8274 or 8253. CA1-2 address a specific register within the chip. Refer to "RS-232-C Registers", in Section 3, for the RS-232-C register address decode.

8274 or 8253 register read or write timing is as follows:

1. The main address decoder asserts XSEL- at 310 nanoseconds.
2. The expansion address decode asserts the applicable CS- at 320 nanoseconds.
3. The selected chip outputs the read data (if applicable) at 550 nanoseconds (8274) or 600 nanoseconds (8253).
4. The address decoder negates the select at about 870 nanoseconds. (This occurs from AT5 on the main board becoming active and negating XSEL- at roughly 860 nanoseconds.)
5. The CPU clocks in the read data (if applicable) at 900 nanoseconds.

6. The CPU negates the write line (if applicable) at about 900 nanoseconds, and the CPU removes the write data at 950 nanoseconds. In between, the chip latches the data.

Due to the 300 nanosecond (8274) and 1000 nanosecond (8253) recovery times, the CPU cannot access the same chip in successive cycles. The Port Status Register read timing and Secondary Transmit/Clock Select Register write timing is the same.

The Secondary Transmit/Clock Select Registers receive only the lower four data bits and decode them as follows:

- Bit 3 is the secondary transmit data for port A, output as SCISTDXA-
- Bit 2 is the secondary transmit data for port B, output as SCISTDXB-
- Bit 1 selects the port A clock. High selects the external clock, CTXCA, and low selects the 8253 clock, TIMROUTX.
- Bit 0 does a similar clock selection for port B.

Refer to Section 4 for a description of the Port Status Register. Refer to "Interrupt Controller" for a description of 8274 interrupts.

ETHERNET CONTROL (SHEETS 13-16; 5-8)

This subsection describes the Ethernet circuitry and the four types of Ethernet Operations:

- CPU initiated Ethernet register read or write
- CPU interrupt
- Ethernet DMA read and write
- Serial data in and out.

Ethernet Circuitry

The Ethernet circuitry contains

- AM7990 local area network controller (sheet 13; 5)
- AM7991 Serial data interface adapter (sheet 13; 5)
- Ethernet state machine, PAL 71 (sheet 14; 6)
- DMA state machine, PAL 72 (sheet 14; 6)
- Ethernet DMA Address Count Register (sheet 16; 8)
- Ethernet output holding register (sheet 16; 8)
- Ethernet data input latches (sheet 16; 8)
- Interrupt status ring pointer (sheet 16; 8).

This manual does not describe the 7990 family chips. Refer to the AM7990 Family Ethernet Node Specification by Advanced Micro Devices. The state machines are described in the context of the four Ethernet operations.

Ethernet Data Holding Registers

The Ethernet output holding register holds outgoing data from the Register Address Port (RAP) and Register Data Port (RDP), located in the 7990. The Ethernet control fetches and loads the holding register as requested by the CPU. The CPU then reads the holding register to obtain the data.

The holding register also acts as a data buffer between the 7990 and memory during Ethernet DMA. Data for memory comes from the 7990's register data port either during the fetch and hold portion of a CPU interrupt (for the Interrupt Status Register) or during DMA. The Ethernet data input latches perform a similar function for incoming DMA and CPU register write data.

Ethernet DMA Address Register

The 7990 loads the Ethernet DMA address register before triggering the DMA state machine for a DMA operation. During the operation, the register supplies the DMA address to the memory control on the main board.

Interrupt Status Ring Pointer

The interrupt status ring pointer is a 21-bit address register which points to the current open location in the 256-word wide interrupt status ring (located in memory). At initialization, the CPU loads the pointer with the starting address of the ring. The upper 14 bits are fixed. The lower 8 bits are a counter.

When the 7990 controller interrupts the CPU, the Ethernet circuitry writes the contents of Interrupt Status Register (in the 7990) to the memory location pointed to by the ring pointer. Then the Ethernet circuitry increments the pointer.

When the lower 8-bit counter overflows, it wraps around and begins at zero. The upper 14 bits are unchanged. The CPU thus has a latency of 256 interrupts before the Ethernet control begins to overwrite valid data in the interrupt status ring.

CPU Initiated Ethernet Register Read Or Write

The CPU initializes and communicates with the Ethernet hardware by reading or writing the following addresses:

- o $C78000 - a write to this address resets the Ethernet hardware.
- o $C7800E - a write to this address latches the Interrupt Status Ring base address.
- o $C78010 - a write to this address loads the 7990 Register Address Port (RAP)
- o $C78012 - a write to this address loads the 7990 Register Data Port (RDP)
- o $C78014 - a write to this address causes the Ethernet control to fetch data from the 7990 RAP and store it in the Ethernet holding register.
- o $C78016 - a write to this address causes the Ethernet control to fetch data from the 7990 RDP and store it in the Ethernet holding register. Note that this overwrites the previous holding register contents.
- o $C78018 - a read to this address reads the contents of the Ethernet holding register.

Table 4-2 is an address decode summary of address bits CA4-1.

Table 4-2. Ethernet Address Decode Bits CA4-1

CA4-1	Function And Type
0000	Reset Ethernet; write only, data independent.
0111	Latch interrupt status ring base; write only with data.
1000	Load Register Address Port (RAP); write only with data.
1001	Load Register Data Port (RDP); write only with data.
1010	Fetch and hold RAP; write only, data independent.
1011	Fetch and hold RDP; write only, data independent.
1100	Read Ethernet holding register; read only with data.

At the beginning of a register access, the CPU outputs the register address. The main board's CPU address decoder asserts CSEL+. The expansion address decoder (sheet 4) asserts ESEL-, which goes to the Ethernet state machine (PAL 71, sheet 14).

The Ethernet machine, decoding CLWT and CA4-1, outputs EØ-2 and goes to state E1, state E2, or state E3. EØ-2 goes to the state machine decoder (PAL 73), the 799Ø control PAL (PAL 74, sheet 15; 7), and the Ethernet interface control PAL (PAL 7Ø, sheet 15; 7). The remainder of this section describes the following three types of register accesses from the point where the Ethernet state machine leaves state EØ and goes to state E1, E2, or E3:

- State E3; CPU writes the RAP or RDP.
- State E2; RAP or RDP fetch and hold (and interrupt).
- State E1; CPU reads the Ethernet output holding register.

Figure 4-4 is a state sequencing diagram of the Ethernet state machine.

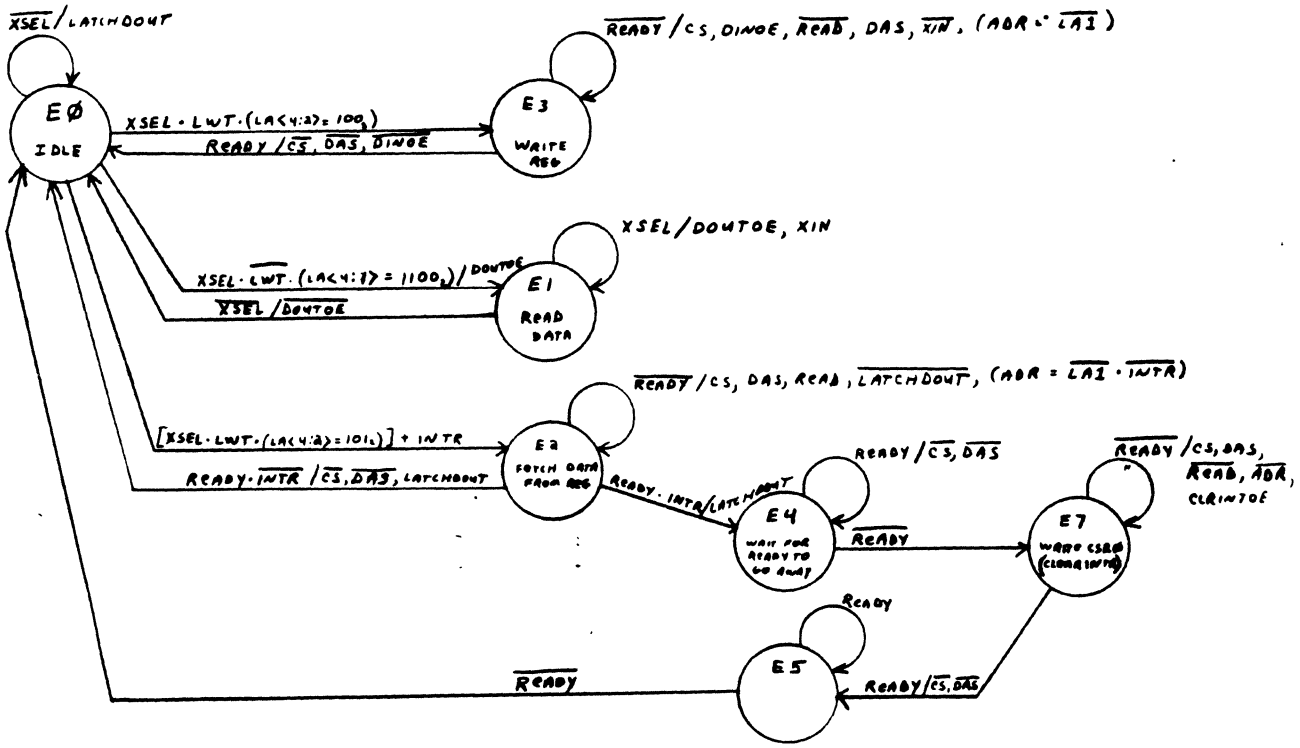


Figure 4-4. Ethernet State Machine State Sequencing Diagram

CPU Write to the RAP or RDP

The Ethernet state machine decodes CLWT (high) and CA4-2 = 1ØØ, asserts EØ and E1, and goes to state E3. The state machine decoder (PAL 73) asserts DINOE-, which enables the data from the main board through the data input latches (sheet 16) to DALØ-15.

The 7990 control PAL asserts CS-, READ (low), and DAS-, and either asserts ADR (CA1-0 = 00) or negates ADR (CA1-0 = 10). The 7990 controller asserts READY- and strobes in the data on DAL0-15. The Ethernet state machine returns to state E0 at the next clock.

7990 RAP or RDP Fetch and Hold

The Ethernet state machine decodes CLWT (high) and CA4-2 = 101, asserts E1, and goes to state E2. The state machine decoder asserts LATCHDOUT-, which enables the Ethernet output holding register.

The 7990 control PAL asserts CS-, READ (E0 is low), DAS-, and decodes CA1 to either assert ADR (CA1 = 0) or negate ADR (CA1 = 1). The 7990 fetches the data and outputs it (roughly at either 600 or 1200 nanoseconds, depending on the whether the 7990 is recovering from a previous RDP write), then asserts READY-. READY- returns the Ethernet state machine to state E0 and causes the state machine decoder to negate LATCHDOUT-. The holding register latches the data.

A CPU interrupt also starts at state E2. Then the Ethernet machine fetches the Interrupt Status Register contents and places them in the Ethernet output holding register.

CPU Read of the Ethernet Output Holding Register

The Ethernet machine decodes CLWT (low) and CA4-1 = 1100 and goes to state E2. The state machine decoder asserts DOUTOE- and XIN-. DOUTOE- enables the contents of the holding register onto the expansion data bus (XD0-15). XIN- routes the data onto the main board's logical data bus.

At the end of the transfer (roughly 870 nanoseconds), the address decode negates ESEL-. The CPU clocks in the read data (900 nanoseconds), and, at the next clock (950 nanoseconds), the Ethernet state machine returns to state E0.

Ethernet DMA Read and Write

At the beginning of the DMA operation, the 7990 asserts HOLD- (to request control of the bus), and the DMA state machine returns the bus acknowledge, HLDA-. The 7990 outputs the read or write address on A16-23 and DAL0-15, asserts or negates READ-, and asserts ALE+. The DMA address counter latches the address.

At the next clock, the 7990 control PAL asserts SYNCREAD-. HOLD- and HLDA- stay asserted. Shortly later the 7990 asserts DAS-.

If it is doing a memory write, the 7990 also outputs the write data onto DAL0-15. DAS- and HLDA- cause the DMA state machine to assert XDMAREQ- and go to state 1. The Ethernet output holding register, enabled by LATCHDOUT- (which the state machine decoder has negated), latches the contents of DAL0-15.

The DMA state machine waits in state D1 for the main board bus arbiter to return XGNT-. At the clock after the arbiter asserts XGNT-, the machine goes to state D2 (150 nanoseconds into the transfer). Simultaneously, the state machine decoder asserts DOUTOE- and ADROE- and asserts or negates LWT- and XIN- (depending on SYNCREAD-). The interface hardware outputs the address and the write data (if applicable) to the main board.

At the next clock (250 nanoseconds), the machine goes to state D3. At the following clock (350 nanoseconds), the machine goes to state D4, where it asserts READY- and removes XDMAREQ-. If the 7990 is reading memory, the state machine decoder, enabled by state D4, asserts DINOE-. DINOE- routes the read data to DAL0-15. READY- tells the 7990 that the transfer is almost over.

If the 7990 is doing a read, it latches the read data. Later the 7990 removes DAS-; the DMA state machine removes READY-. At the next clock, the DMA state machine returns to state D0. When the 7990 removes HOLD-, the DMA state machine removes HLDA-. Figure 4-5 is a state sequencing diagram of the DMA state machine.

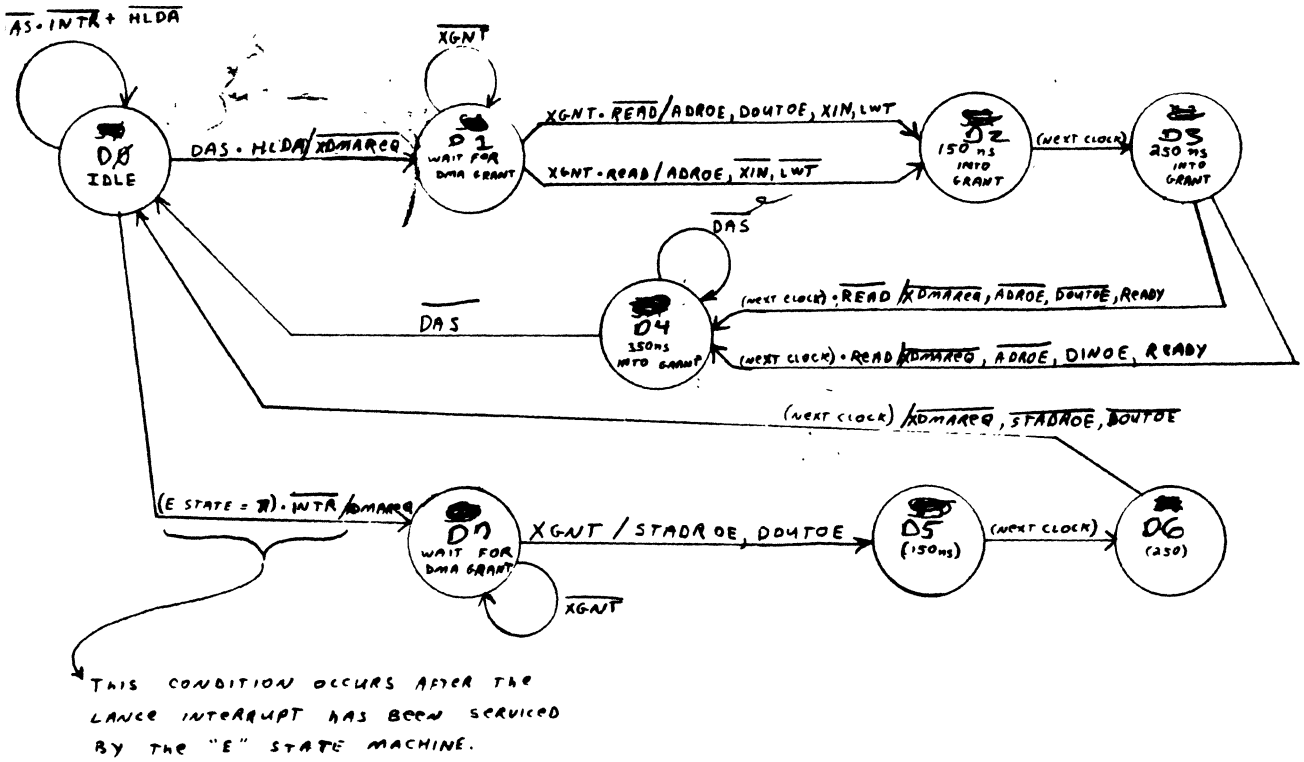


Figure 4-5. DMA State Machine State Sequencing Diagram

Ethernet CPU Interrupt

The interrupt has two phases. First the 799Ø asserts the interrupt signal, and causes the Ethernet state machine to go to state E2. The Ethernet state machine fetches the contents of the Register Data Port in the 799Ø and places them in the Ethernet output holding register. Second, the DMA state machine writes the holding register contents to the memory location pointed to by the Interrupt Status Ring counter, then increments the counter.

Ethernet State Machine Fetching the RDP

When the 799Ø asserts INTR-, the 799Ø control PAL synchronizes it and sends SYNCINTR- to the Ethernet state machine and the DMA state machine. The Ethernet state machine outputs INTE- to the interrupt controller and goes to state E2. In state E2, the Ethernet state machine does a fetch and hold of the 799Ø's Register Data Port. (See "799Ø Register Fetch And Hold", above.). When the 799Ø returns READY-, instead of returning to state EØ, the Ethernet machine goes to state E4.

When the 799Ø removes READY-, the Ethernet state machine asserts the required chip control signals to write the 799Ø Control and Status Register #Ø (CSRØ). The CSRØ write data comes from the clear interrupt latches, which the state machine enables by CLRINTOE-. The CSRØ write clears the interrupt. When the 799Ø asserts READY- the machine goes to state E5. When the 799Ø removes READY-, the machine returns to state EØ.

The Ethernet state machine keeps INTR- asserted until the interrupt controller returns the acknowledge INTAE-. The delay before receiving INTAE- is indeterminate, and the 799Ø can reassert INTR- before the CPU honors the first interrupt with no loss in service. In fact, 799Ø can interrupt the CPU many times before the CPU must respond, due to the ISR ring being 256 words long. At the clock after the CPU returns INTAE-, the Ethernet state machine removes INTR-.

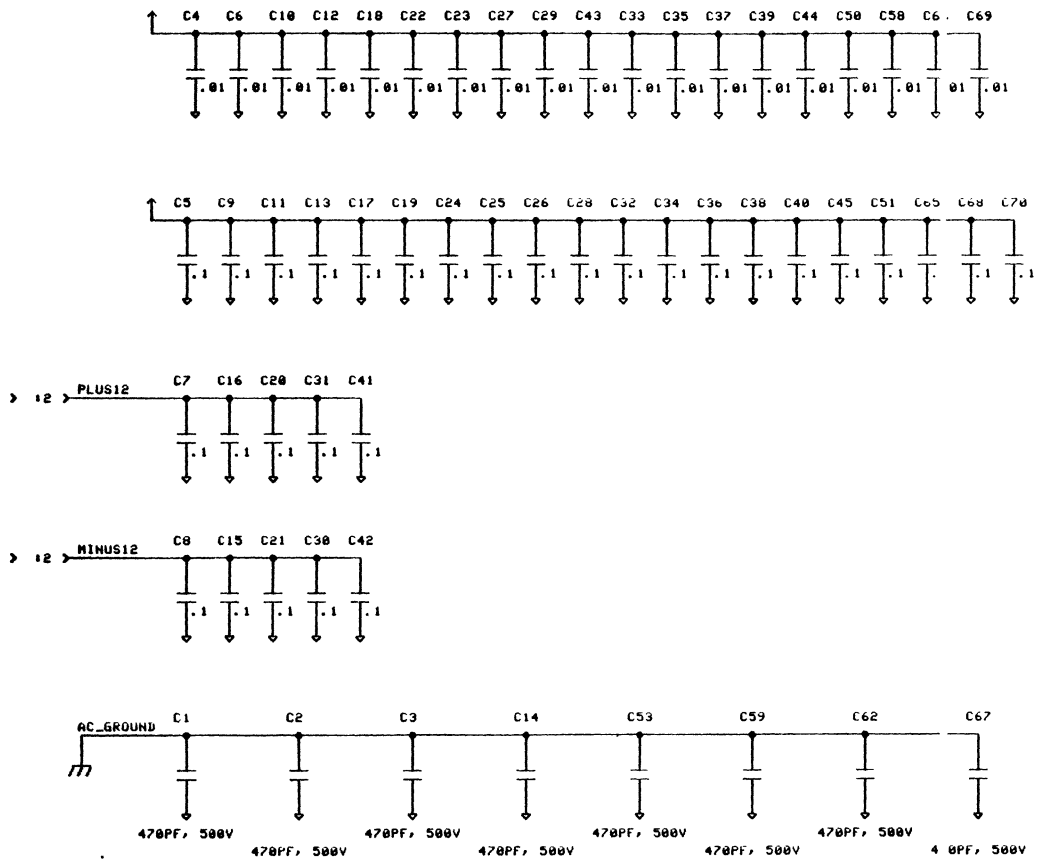
Ethernet Output Holding Register to Memory

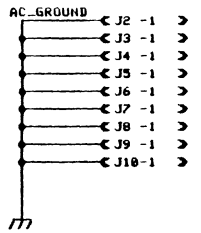
SYNCINTR- and the signal E2 cause the DMA state machine to assert XDMAREQ- and to go to state E7. In state E7, the machine is not dependent on E2 and asserts XDMAREQ- until the bus arbiter returns XGNT-. When the bus arbiter returns XGNT-, the state machine decoder asserts STADROE-, DOUTOUE-, XIN-, and LWT-. The Ethernet control PAL (PAL 273, sheet 15; 7) prepares the clock to the ISR ring address counter by asserting STADCLK-.

The ISR counter (sheet 16; 8) sends the main board memory control the memory address of the next open position in the interrupt status register ring. The holding register places the Interrupt Status Register contents (loaded by the Ethernet machine) onto XDØ-15.

At the next clock, the DMA state machine goes to state D5. The main board memory control loads the ISR contents into the ISR ring. The next clock sends the DMA state machine back to state E6 and causes it to remove XDMAREQ-.

At the next clock, the DMA state machine goes to state DØ, while the bus arbiter removes XGNT-. The state machine decoder negates STADROE-. The Ethernet control PAL negates STADCLK, causing the ISR ring address counter to increment and point to the next open position in the Interrupt Status Ring.





REVISIONS					
ZONE	REV	DESCRIPTION	DATE	CHK	APPROVED

NOTES:

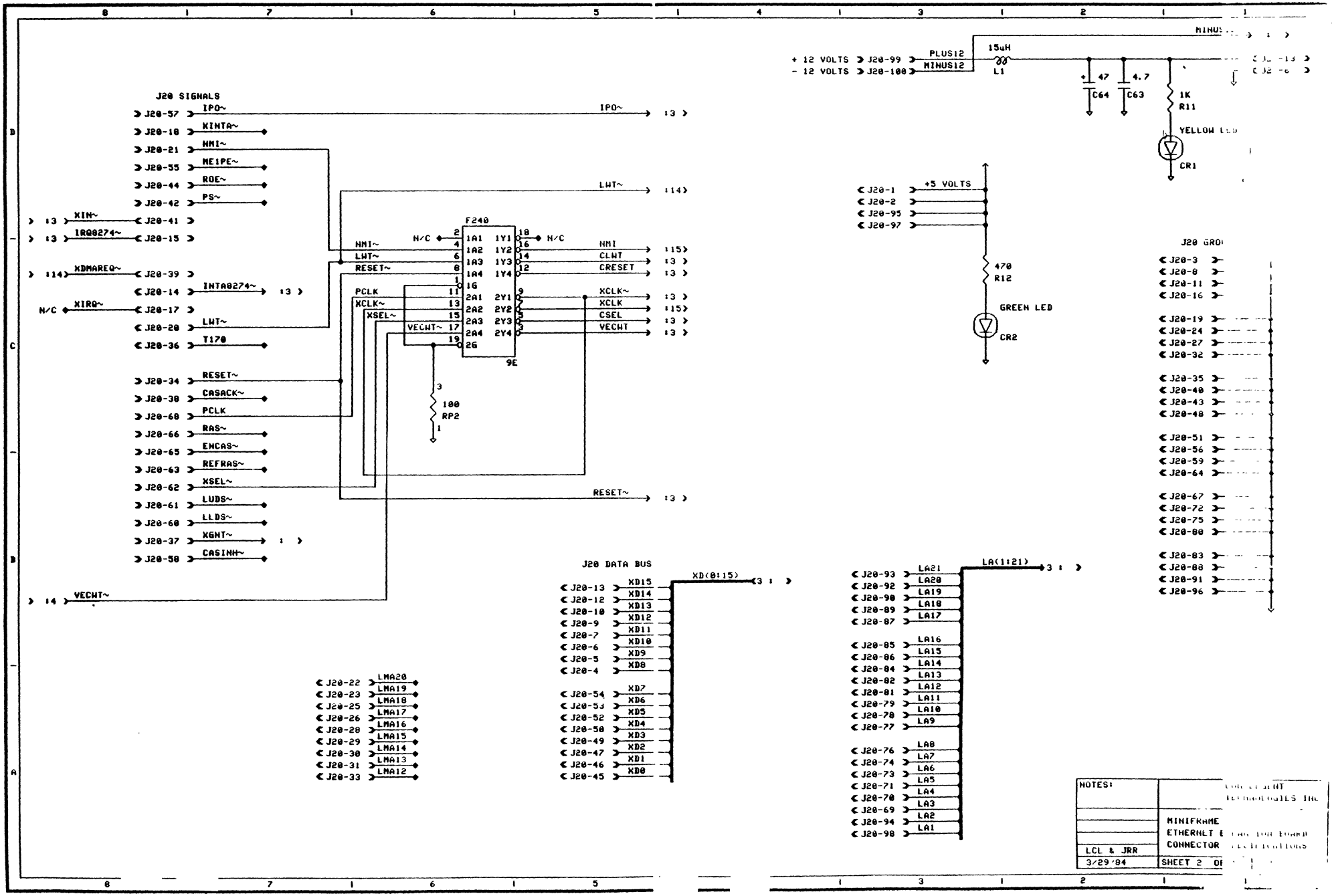
APPROVALS	DATE
DRAWN	
CHECKED	
APVD	

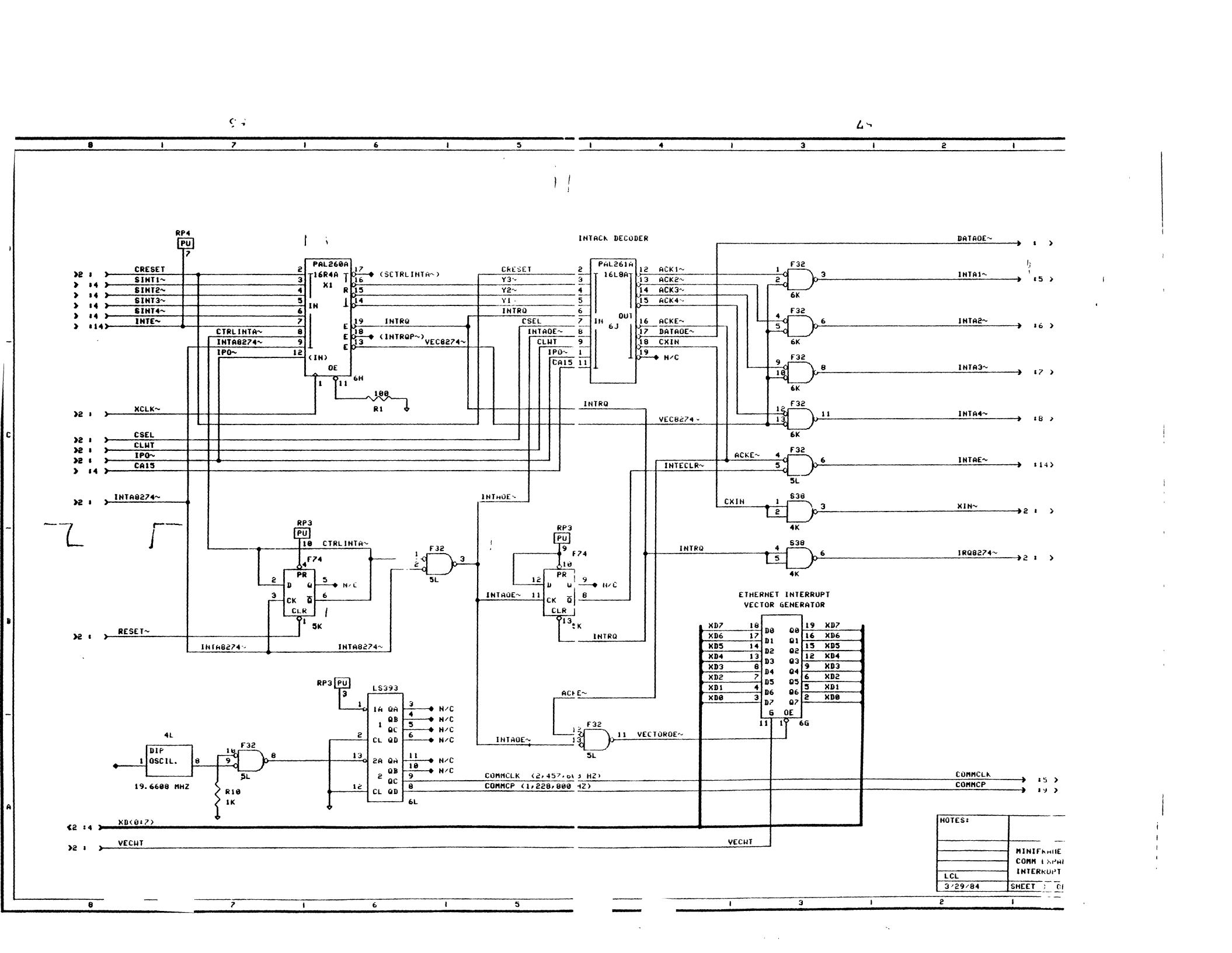
CONTINENTAL TECHNOLOGIES INC.

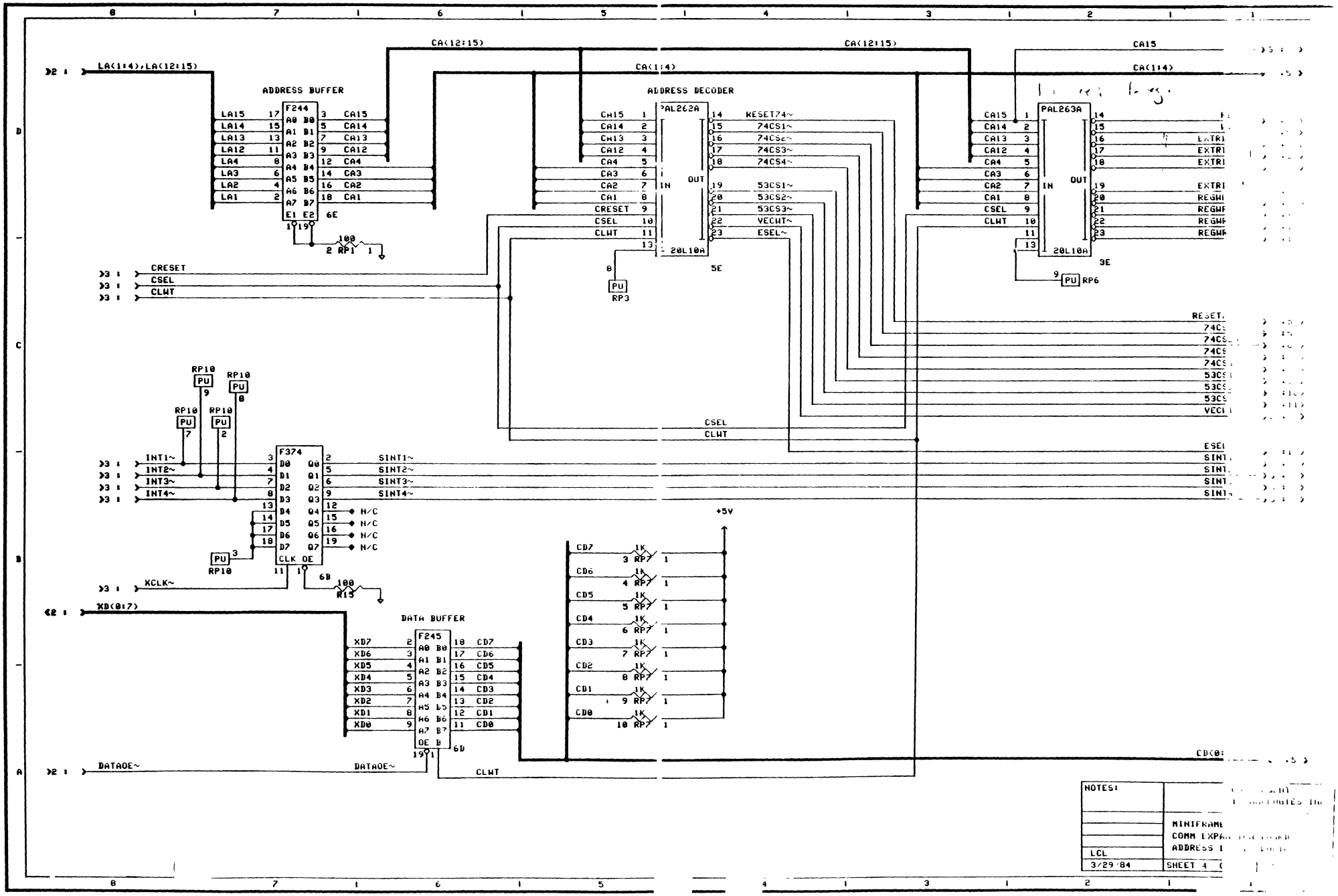
MINIFRAME
COMM EXPANSION BOARD

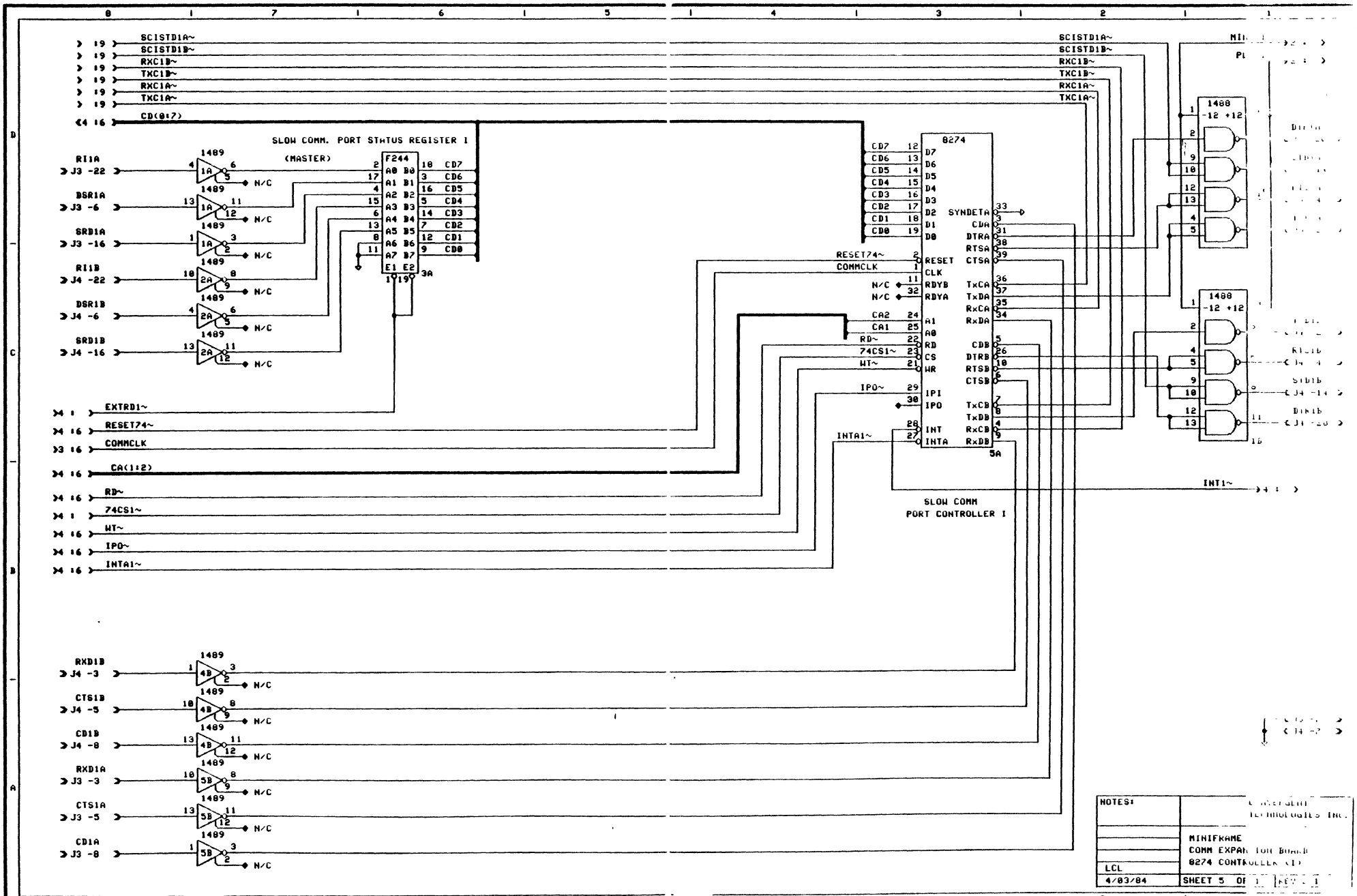
DWG. NO.

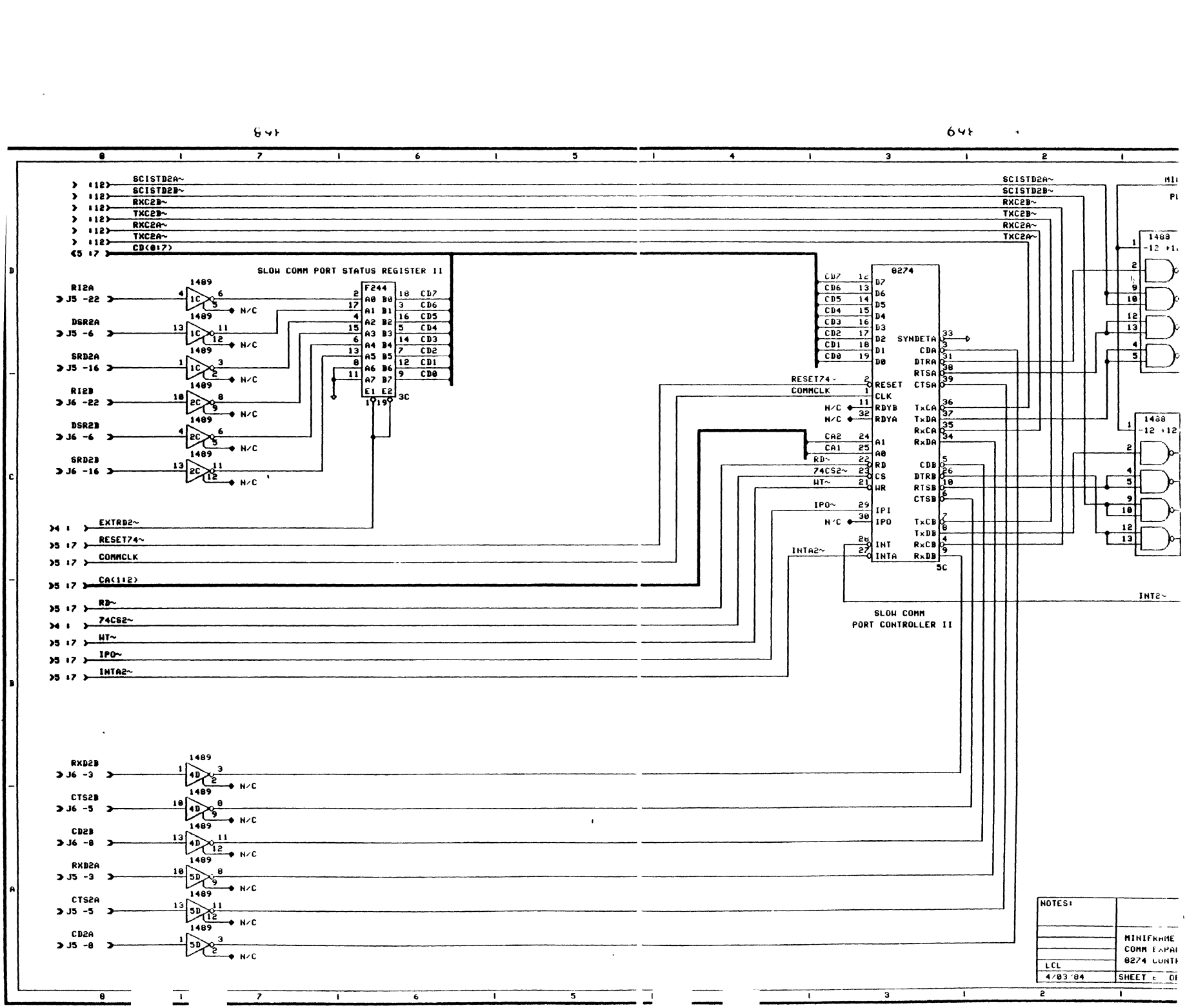
SHEET 1 OF 1 REV 2.0

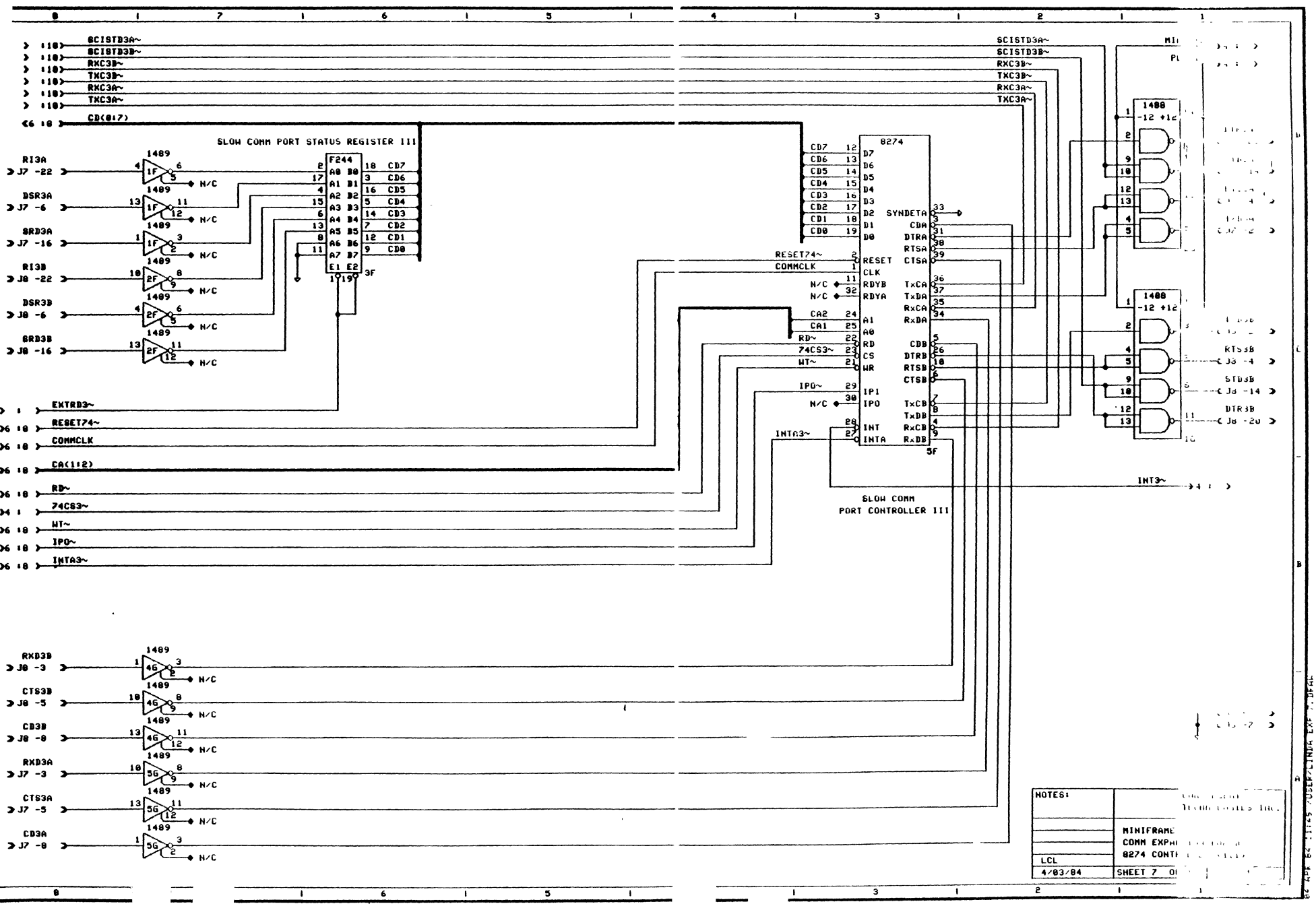

J20 SIGNALS
J20-57 IPO~
J20-18 XINTA~
J20-21 NMI~
J20-55 ME1PE~
J20-44 ROE~
J20-42 PS~
XIN~ J20-41
IRQ8274~ J20-15
XDMAREQ~ J20-39
INTA8274~ J20-14
XIRQ~ J20-17
LWT~ J20-20
T170 J20-36
RESET~ J20-34
CASACK~ J20-38
PCLK J20-68
RAS~ J20-66
EMCAS~ J20-65
REFRAS~ J20-63
XSEL~ J20-62
LUDS~ J20-61
LLDS~ J20-60
XGNT~ J20-37
CASINH~ J20-58
VECHT~ J20-50
F240
9E
N/C
NMI~
LWT~
RESET~
PCLK
XCLK~
XSEL~
VECHT~
1A1 1A2 1A3 1A4 2A1 2A2 2A3 2A4
1Y1 1Y2 1Y3 1Y4 2Y1 2Y2 2Y3 2Y4
1G 2G
NMI
CLWT
CRESET
XCLK~
XCLK
CSEL
VECHT
RESET~
RP2
100
J20 DATA BUS
XD(0:15)
J20-13 XD15
J20-12 XD14
J20-11 XD13
J20-10 XD12
J20-9 XD11
J20-7 XD10
J20-6 XD9
J20-5 XD8
J20-4 XD7
J20-54 XD6
J20-53 XD5
J20-52 XD4
J20-50 XD3
J20-49 XD2
J20-47 XD1
J20-46 XD0
J20-45
J20-22 LMA20
J20-23 LMA19
J20-25 LMA18
J20-26 LMA17
J20-28 LMA16
J20-29 LMA15
J20-30 LMA14
J20-31 LMA13
J20-33 LMA12
+ 12 VOLTS J20-99 PLUS12
- 12 VOLTS J20-100 MINUS12
15uH L1
47 C64
4.7 C63
1K R11
YELLOW LED CR1
+5 VOLTS
J20-1
J20-2
J20-95
J20-97
470 R12
GREEN LED CR2
J20 GRO
J20-3
J20-8
J20-11
J20-16
J20-19
J20-24
J20-27
J20-32
J20-35
J20-40
J20-43
J20-48
J20-51
J20-56
J20-59
J20-64
J20-67
J20-72
J20-75
J20-80
J20-83
J20-88
J20-91
J20-96
LA(17:21)
J20-93 LA21
J20-92 LA20
J20-90 LA19
J20-89 LA18
J20-87 LA17
J20-85 LA16
J20-86 LA15
J20-84 LA14
J20-82 LA13
J20-81 LA12
J20-79 LA11
J20-78 LA10
J20-77 LA9
J20-76 LA8
J20-74 LA7
J20-73 LA6
J20-71 LA5
J20-70 LA4
J20-69 LA3
J20-94 LA2
J20-98 LA1
NOTES:
MINIFRAME
ETHERNET
CONNECTOR
LCL & JRR
3/29/84
SHEET 2

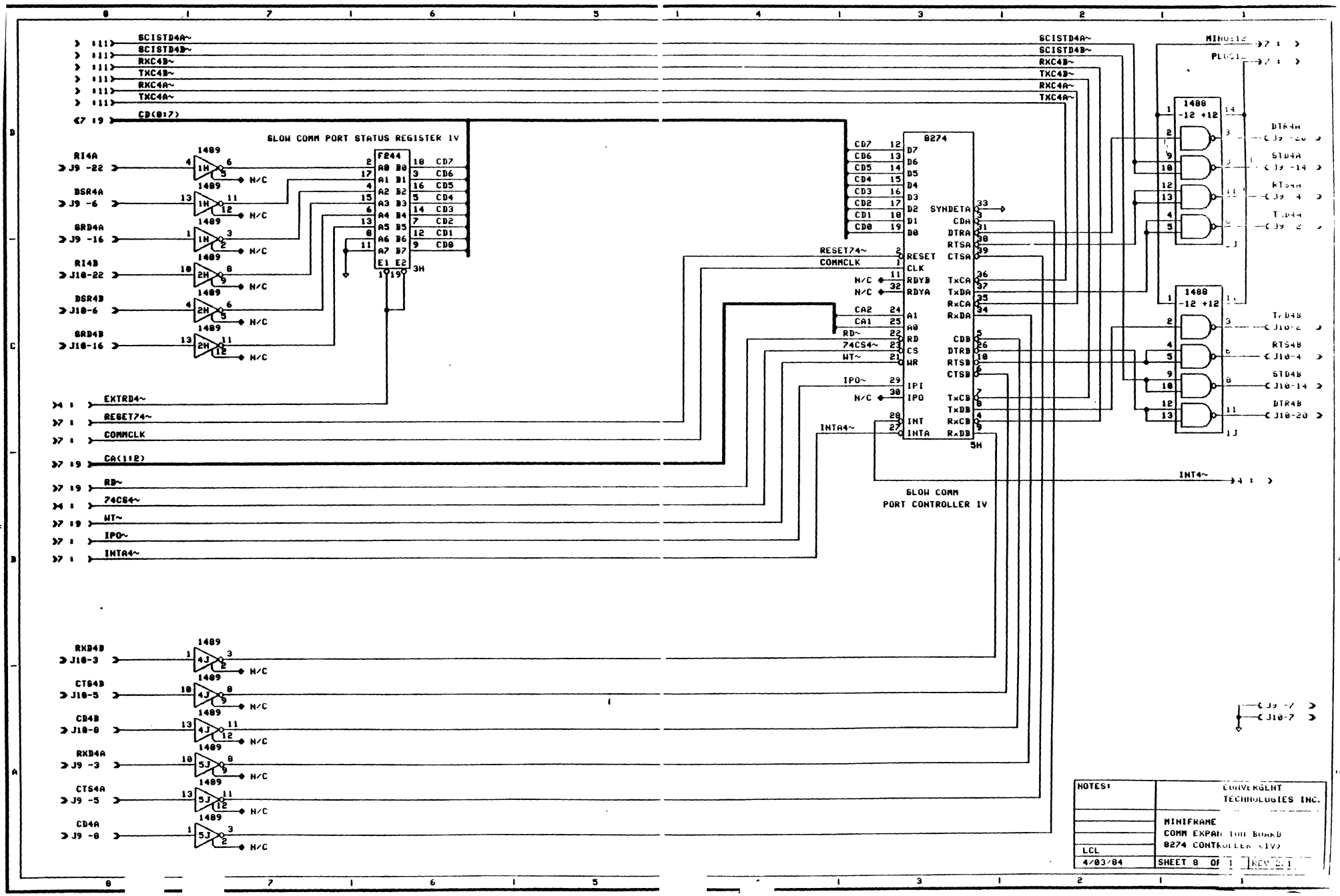

RP4
PU
PAL260A
16R4A
X1
CRESET
SINT1~
SINT2~
SINT3~
SINT4~
INTE~
CTRLINTA~
INTA8274~
IPO~
(IN)
OE
6H
100
R1
(SCTRLINTA~)
INTRQ
(INTRQP~)
VEC8274~
XCLK~
CSEL
CLWT
IPO~
CA15
INTA8274~
RESET~
INTACK DECODER
PAL261A
16L8A
IN
6J
OUT
ACK1~
ACK2~
ACK3~
ACK4~
ACKE~
DATAOE~
CXIN
N/C
Y3~
Y2~
Y1~
INTRQ
CSEL
INTAOE~
CLWT
IPO~
CA15
F32
6K
INTA1~
INTA2~
INTA3~
INTA4~
INTAE~
5L
S38
4K
XIN~
IRQ8274~
ACKE~
INTECLR~
CXIN
INTRQ
VEC8274~
RP3
F74
PR
CK
CLR
5K
INTAOE~
INTRQ
LS393
1A QA
1 QB
QC
CL QD
2A QA
2 QB
QC
CL QD
6L
N/C
4L
DIP
OSCIL.
19.6608 MHZ
R10
1K
COMMCLK (2,457,600 HZ)
COMMCP (1,228,800 HZ)
COMMCLK
COMMCP
ETHERNET INTERRUPT
VECTOR GENERATOR
XD7
XD6
XD5
XD4
XD3
XD2
XD1
XD0
D0
D1
D2
D3
D4
D5
D6
D7
G
OE
6G
VECTOROE~
XD(0:7)
VECHT
NOTES:
MINIFRAME
COMM EXPAN
INTERRUPT
LCL
3/29/84
SHEET

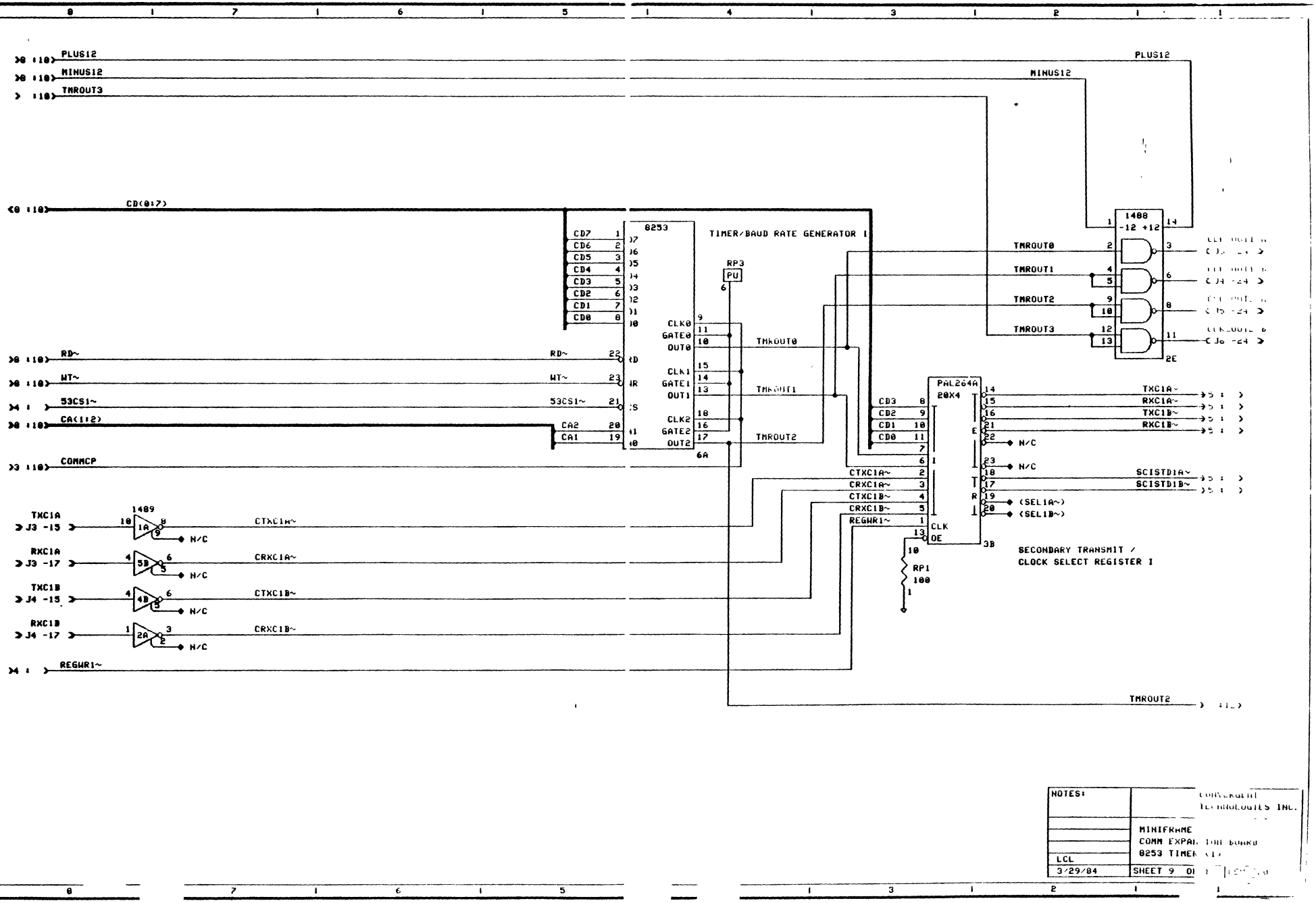

LA(1:4),LA(12:15)
ADDRESS BUFFER
F244
LA15 17
LA14 15
LA13 13
LA12 11
LA4 8
LA3 6
LA2 4
LA1 2
A0 B0
A1 B1
A2 B2
A3 B3
A4 B4
A5 B5
A6 B6
A7 B7
E1 E2
1 19
3 CA15
5 CA14
7 CA13
9 CA12
12 CA4
14 CA3
16 CA2
18 CA1
6E
100
2 RP1 1
CA(12:15)
CA(1:4)
ADDRESS DECODER
PAL262A
CA15 1
CA14 2
CA13 3
CA12 4
CA4 5
CA3 6
CA2 7
CA1 8
CRESET 9
CSEL 10
CLWT 11
13
IN
OUT
14 RESET74~
15 74CS1~
16 74CS2~
17 74CS3~
18 74CS4~
19 53CS1~
20 53CS2~
21 53CS3~
22 VECWT~
23 ESEL~
20L10A
5E
8
PU
RP3
PAL263A
CA15 1
CA14 2
CA13 3
CA12 4
CA4 5
CA3 6
CA2 7
CA1 8
CSEL 9
CLWT 10
11
13
IN
OUT
14
15
16 EXTRI
17 EXTRI
18 EXTRI
19 EXTRI
20 REGWI
21 REGWR
22 REGWR
23 REGWR
20L10A
3E
9
PU
RP6
CRESET
CSEL
CLWT
RESET/
74CS
74CS
74CS
74CS
53CS
53CS
53CS
VECI
CSEL
CLWT
ESEL
SINT
SINT
SINT
SINT
RP10
PU
9
8
7
2
INT1~
INT2~
INT3~
INT4~
F374
D0 Q0
D1 Q1
D2 Q2
D3 Q3
D4 Q4
D5 Q5
D6 Q6
D7 Q7
CLK OE
3 2
4 5
7 6
8 9
SINT1~
SINT2~
SINT3~
SINT4~
13 12 N/C
14 15 N/C
17 16 N/C
18 19 N/C
11 1
6B
100
R13
XCLK~
XD(0:7)
DATA BUFFER
F245
XD7 2
XD6 3
XD5 4
XD4 5
XD3 6
XD2 7
XD1 8
XD0 9
A0 B0
A1 B1
A2 B2
A3 B3
A4 B4
A5 B5
A6 B6
A7 B7
OE B
19 1
18 CD7
17 CD6
16 CD5
15 CD4
14 CD3
13 CD2
12 CD1
11 CD0
6D
+5V
CD7 1K 3 RP7 1
CD6 1K 4 RP7 1
CD5 1K 5 RP7 1
CD4 1K 6 RP7 1
CD3 1K 7 RP7 1
CD2 1K 8 RP7 1
CD1 1K 9 RP7 1
CD0 1K 10 RP7 1
CD(0:
DATAOE~
DATAOE~
CLWT
NOTES:
MINIFRAME
COMM EXPA
ADDRESS I
LCL
3/29/84
SHEET 4


SCISTD1A~
SCISTD1B~
RXC1B~
TXC1B~
RXC1A~
TXC1A~
CD(0:7)
SLOW COMM. PORT STATUS REGISTER I
(MASTER)
F244
RI1A
J3 -22
DSR1A
J3 -6
SRD1A
J3 -16
RI1B
J4 -22
DSR1B
J4 -6
SRD1B
J4 -16
N/C
3A
EXTRD1~
RESET74~
COMMCLK
CA(1:2)
RD~
74CS1~
WT~
IPO~
INTA1~
8274
SYNDETA
CDA
DTRA
RTSA
CTSA
TxCA
TxDA
RxCA
RxDA
CDB
DTRB
RTSB
CTSB
TxCB
TxDB
RxCB
RxDB
5A
SLOW COMM
PORT CONTROLLER I
INT1~
1488
-12 +12
RXD1B
J4 -3
CTS1B
J4 -5
CD1B
J4 -8
RXD1A
J3 -3
CTS1A
J3 -5
CD1A
J3 -8
NOTES:
MINIFRAME
COMM EXPANSION BOARD
8274 CONTROLLER (I)
LCL
4/03/84
SHEET 5 OF 11
REV - 1

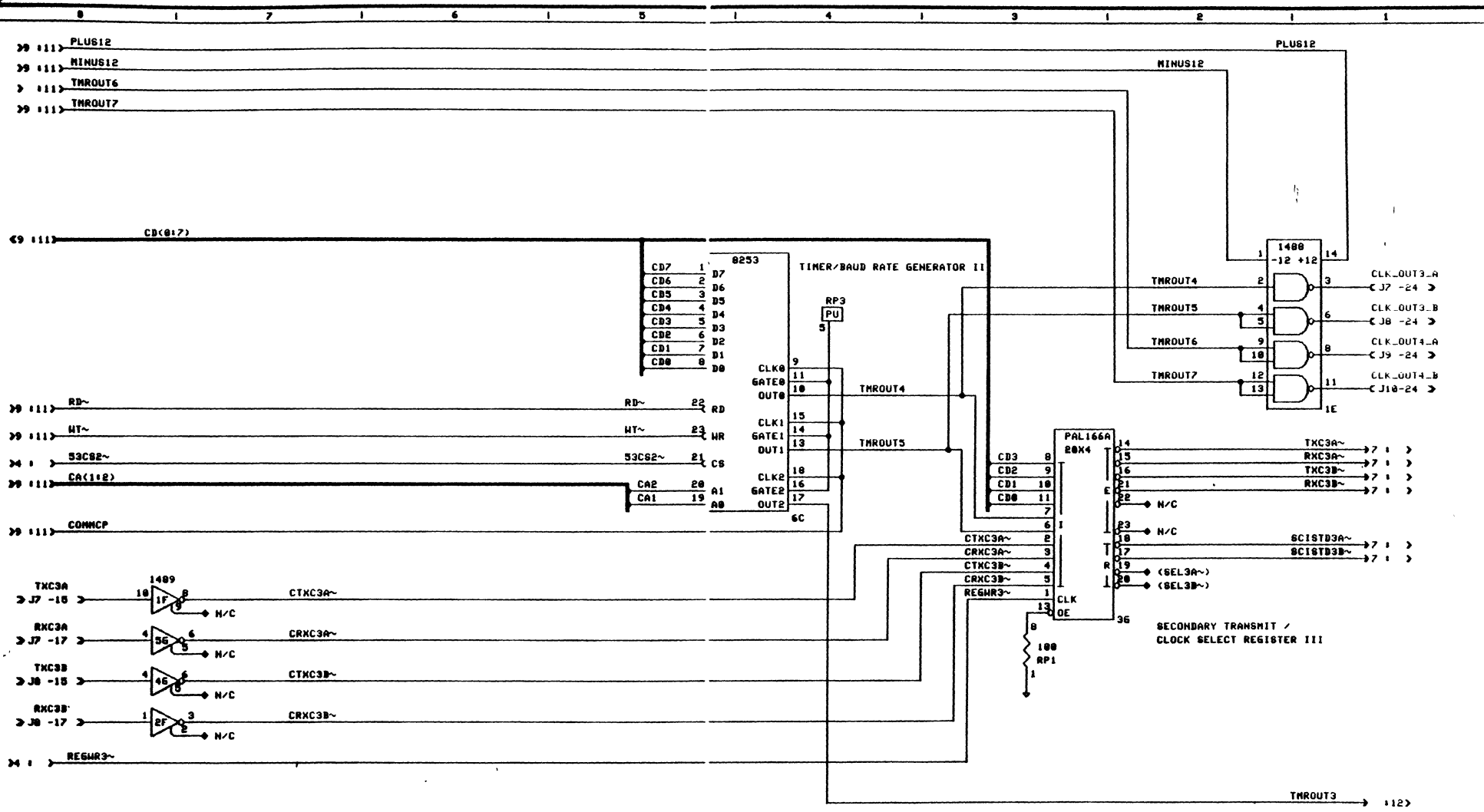

SCISTD2A~
SCISTD2B~
RXC2B~
TXC2B~
RXC2A~
TXC2A~
CD(0:7)
SLOW COMM PORT STATUS REGISTER II
RI2A
J5 -22
DSR2A
J5 -6
SRD2A
J5 -16
RI2B
J6 -22
DSR2B
J6 -6
SRD2B
J6 -16
N/C
F244
EXTRD2~
RESET74~
COMMCLK
CA(1:2)
RD~
74CS2~
WT~
IPO~
INTA2~
8274
SYNDETA
CDA
DTRA
RTSA
CTSA
TxCA
TxDA
RxCA
RxDA
CDB
DTRB
RTSB
CTSB
TxCB
TxDB
RxCB
RxDB
SLOW COMM
PORT CONTROLLER II
INT2~
RXD2B
J6 -3
CTS2B
J6 -5
CD2B
J6 -8
RXD2A
J5 -3
CTS2A
J5 -5
CD2A
J5 -8
NOTES:
MINIFRAME
COMM EXPAI
8274 CONTI
LCL
4/03/84
SHEET

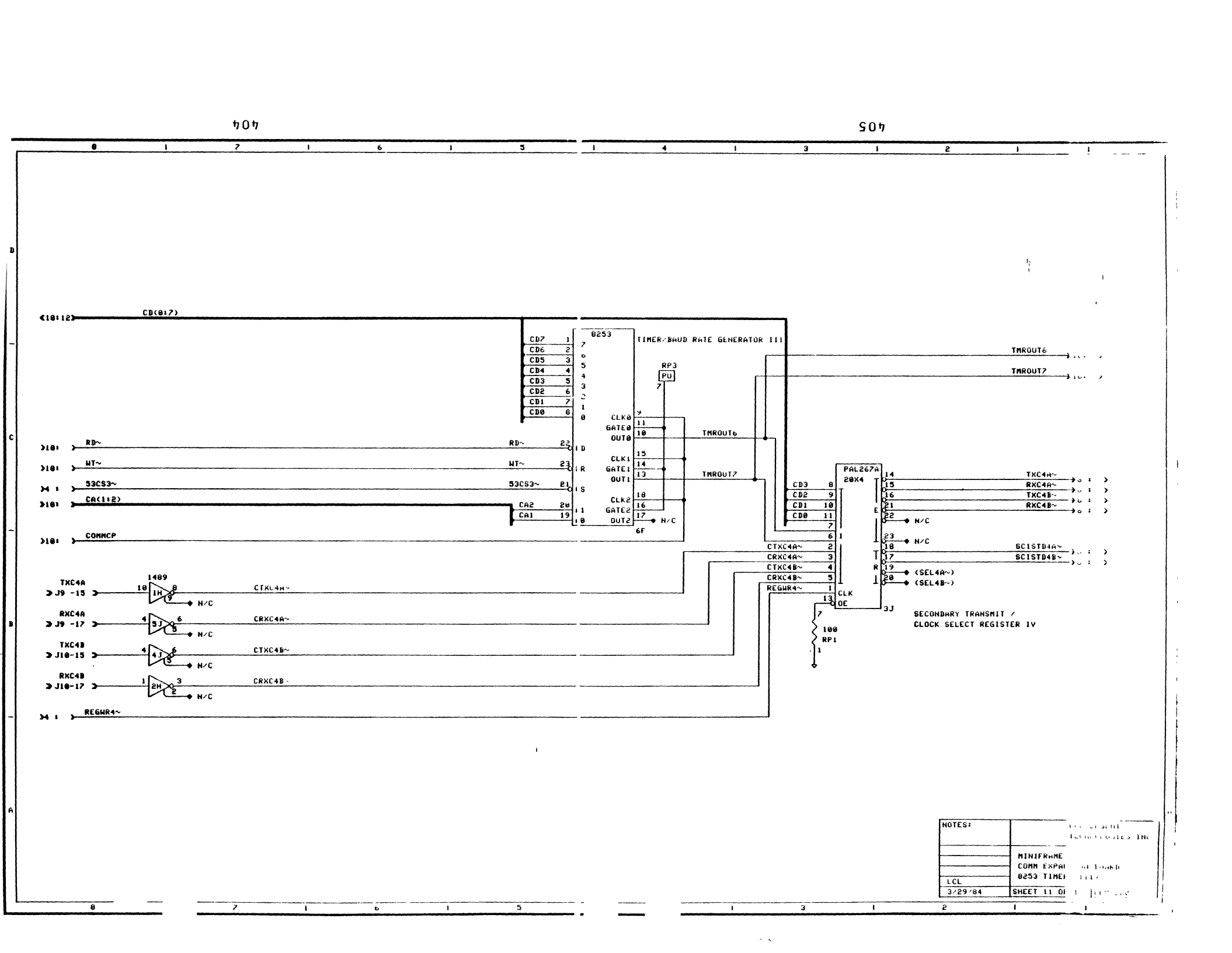

SCISTD3A~
SCISTD3B~
RXC3B~
TXC3B~
RXC3A~
TXC3A~
CD(0:7)
SLOW COMM PORT STATUS REGISTER III
1489
F244
RI3A
J7 -22
DSR3A
J7 -6
SRD3A
J7 -16
RI3B
J8 -22
DSR3B
J8 -6
SRD3B
J8 -16
N/C
EXTRD3~
RESET74~
COMMCLK
CA(1:2)
RD~
74CS3~
WT~
IPO~
INTA3~
8274
SYNDETA
RESET74~
COMMCLK
CA2
CA1
RD~
74CS3~
WT~
IPO~
INTA3~
INT3~
SLOW COMM
PORT CONTROLLER III
RXD3B
J8 -3
CTS3B
J8 -5
CD3B
J8 -8
RXD3A
J7 -3
CTS3A
J7 -5
CD3A
J7 -8
1488
-12 +12
RTS3B
J8 -4
STD3B
J8 -14
DTR3B
J8 -20
NOTES:
MINIFRAME
COMM EXPA
8274 CONT
LCL
4/03/84
SHEET 7 OF

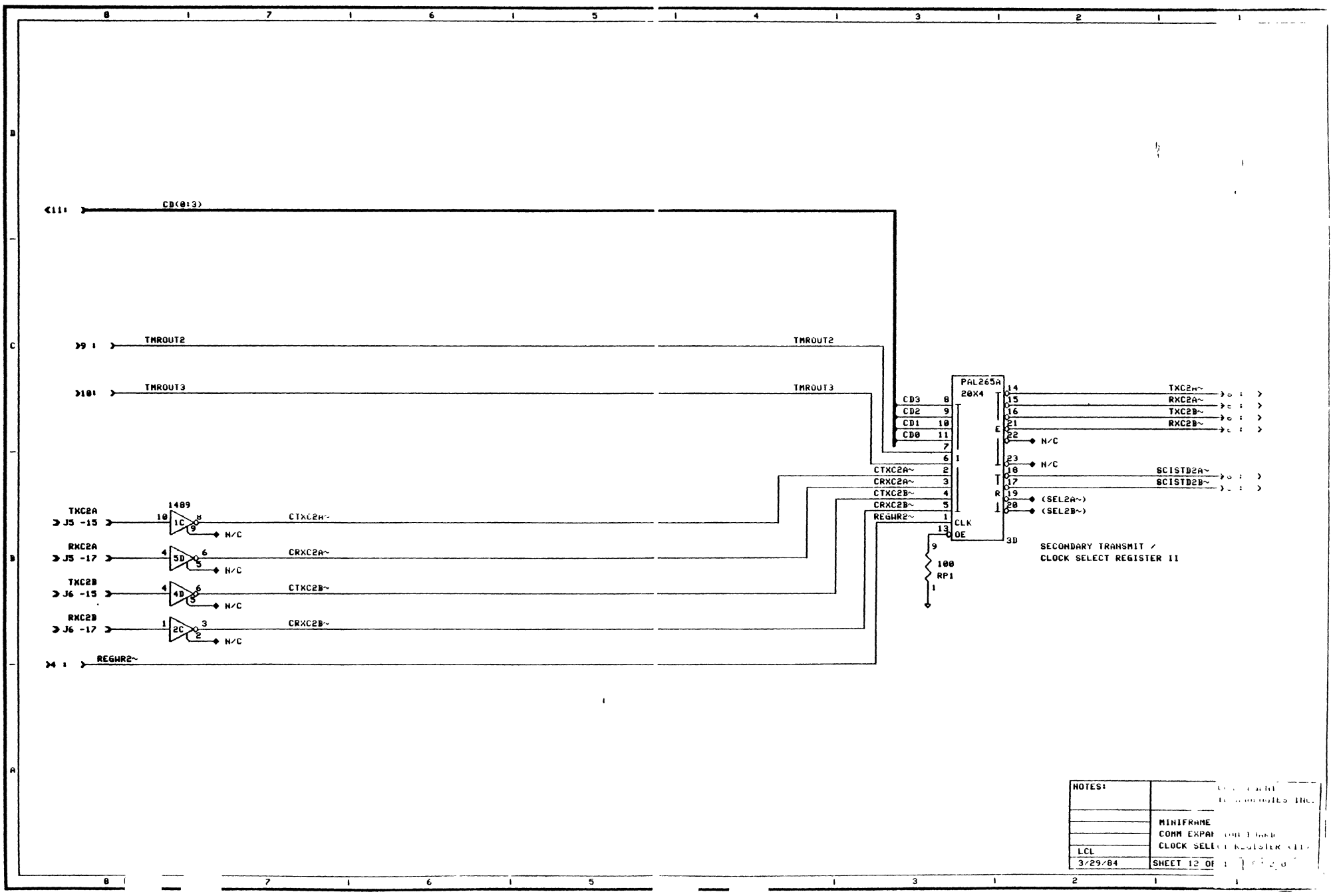

SCISTD4A~
SCISTD4B~
RXC4B~
TXC4B~
RXC4A~
TXC4A~
CD(0:7)
SLOW COMM PORT STATUS REGISTER IV
RI4A
J9 -22
DSR4A
J9 -6
SRD4A
J9 -16
RI4B
J10-22
DSR4B
J10-6
SRD4B
J10-16
N/C
1489
FF44
3H
EXTRD4~
RESET74~
COMMCLK
CA(1:2)
RD~
74CS4~
WT~
IPO~
INTA4~
8274
SYNDETA
CDA
DTRA
RTSA
CTSA
TxCA
TxDA
RxCA
RxDA
CDB
DTRB
RTSB
CTSB
TxCB
TxDB
RxCB
RxDB
INT
IPI
IPO
RESET
CLK
RDYB
RDYA
A1
A0
RD
CS
WR
INTA
5H
SLOW COMM
PORT CONTROLLER IV
INT4~
1488
-12 +12
DTR4A
STD4A
RTS4A
T.D4A
T.D4B
RTS4B
STD4B
DTR4B
RXD4B
J10-3
CTS4B
J10-5
CD4B
J10-8
RXD4A
J9 -3
CTS4A
J9 -5
CD4A
J9 -8
J9 -7
J10-7
NOTES:
CONVERGENT
TECHNOLOGIES INC.
MINIFRAME
COMM EXPANSION BOARD
8274 CONTROLLER (IV)
LCL
4/03/84
SHEET 8 OF 11
REV 2.1

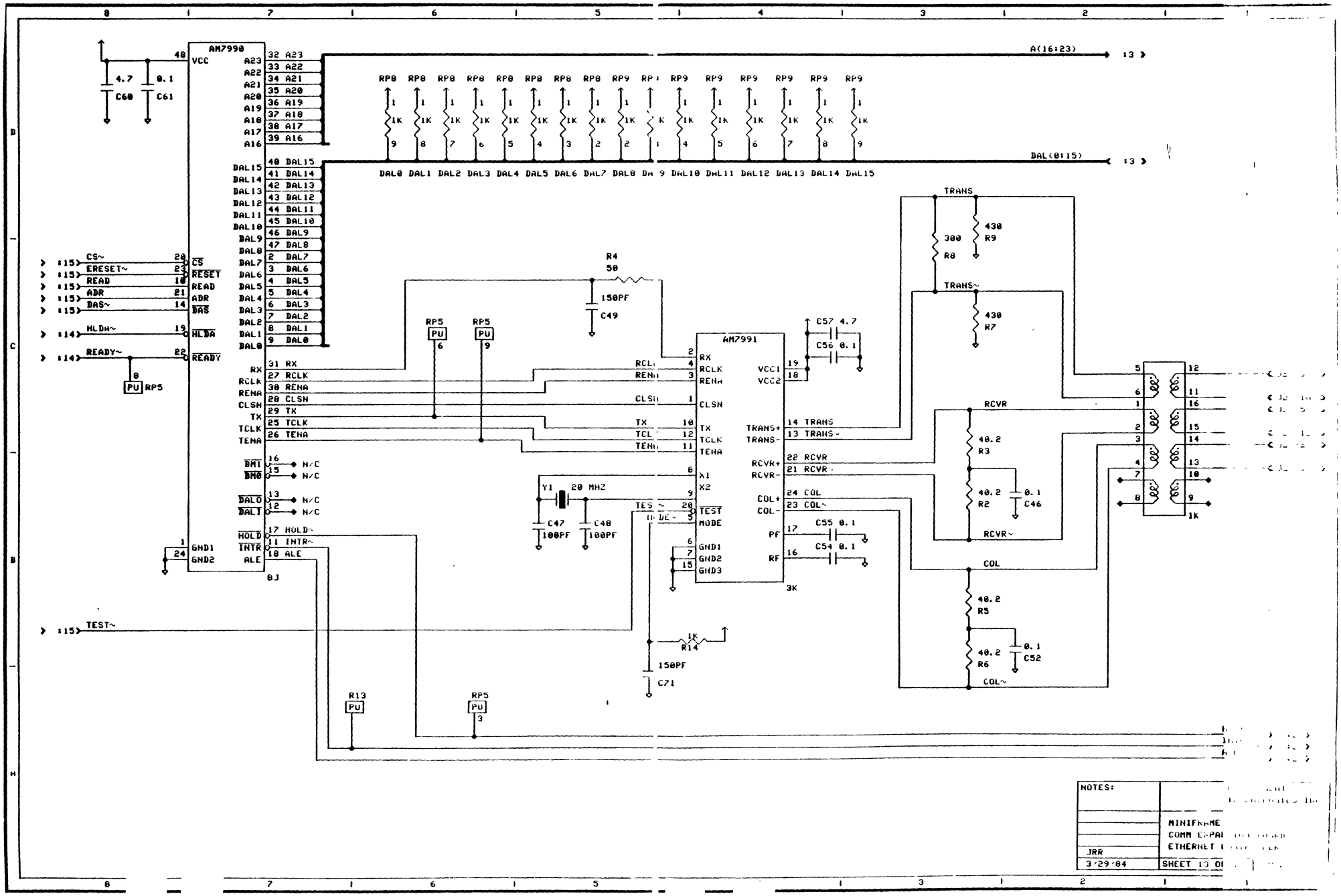


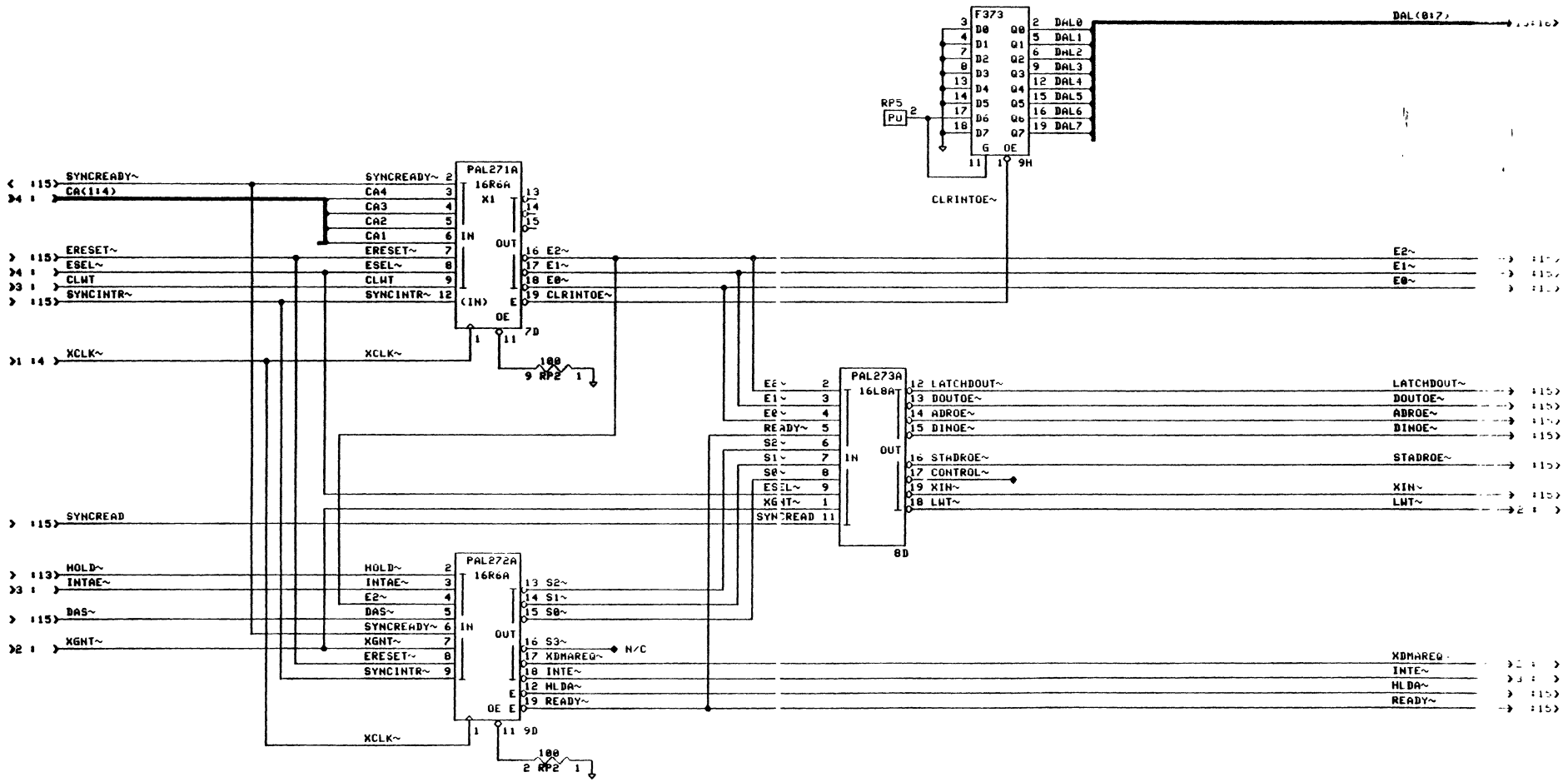
NOTES:	CONVERGENT TECHNOLOGIES INC.
	MINIFRAME
	COMM EXPAN. IOU BOARD
	8253 TIMER (1)
LCL	
3/29/84	SHEET 9 OF


PLUS12
MINUS12
TMROUT6
TMROUT7
CD(0:7)
RD~
WT~
53C82~
CA(1:2)
COMMCP
TXC3A
J7 -15
RXC3A
J7 -17
TXC3B
J8 -15
RXC3B
J8 -17
REGWR3~
1489
1F
2F
N/C
CTXC3A~
CRXC3A~
CTXC3B~
CRXC3B~
8253
TIMER/BAUD RATE GENERATOR II
CD7
CD6
CD5
CD4
CD3
CD2
CD1
CD0
D7
D6
D5
D4
D3
D2
D1
D0
RD
WR
CS
A1
A0
CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2
6C
RP3
PU
TMROUT4
TMROUT5
CA2
CA1
1488
-12 +12
CLK_OUT3_A
J7 -24
CLK_OUT3_B
J8 -24
CLK_OUT4_A
J9 -24
CLK_OUT4_B
J10-24
1E
PAL16A
20X4
I
R
E
CLK
OE
SCISTD3A~
SCISTD3B~
(SEL3A~)
(SEL3B~)
100
RP1
SECONDARY TRANSMIT /
CLOCK SELECT REGISTER III
TMROUT3
NOTES:
CONVERGENT
TECHNOLOGIES INC.
MINIFRAME
COMM EXPANSION BOARD
8253 TIMER (II)
LCL
3/29/84
SHEET 10 OF 16
REV 2.0


CD(0:7)
8253
TIMER/BAUD RATE GENERATOR III
RP3
PU
RD~
WT~
53CS3~
CA(1:2)
COMMCP
TMROUT6
TMROUT7
PAL267A
20X4
TXC4A~
RXC4A~
TXC4B~
RXC4B~
SCISTD4A~
SCISTD4B~
(SEL4A~)
(SEL4B~)
N/C
CLK
OE
100
RP1
TXC4A
J9 -15
RXC4A
J9 -17
TXC4B
J10-15
RXC4B
J10-17
REGWR4~
1489
SECONDARY TRANSMIT / CLOCK SELECT REGISTER IV
MINIFRAME
COMM EXPAN
8253 TIMER
LCL
3/29/84
SHEET 11 OF

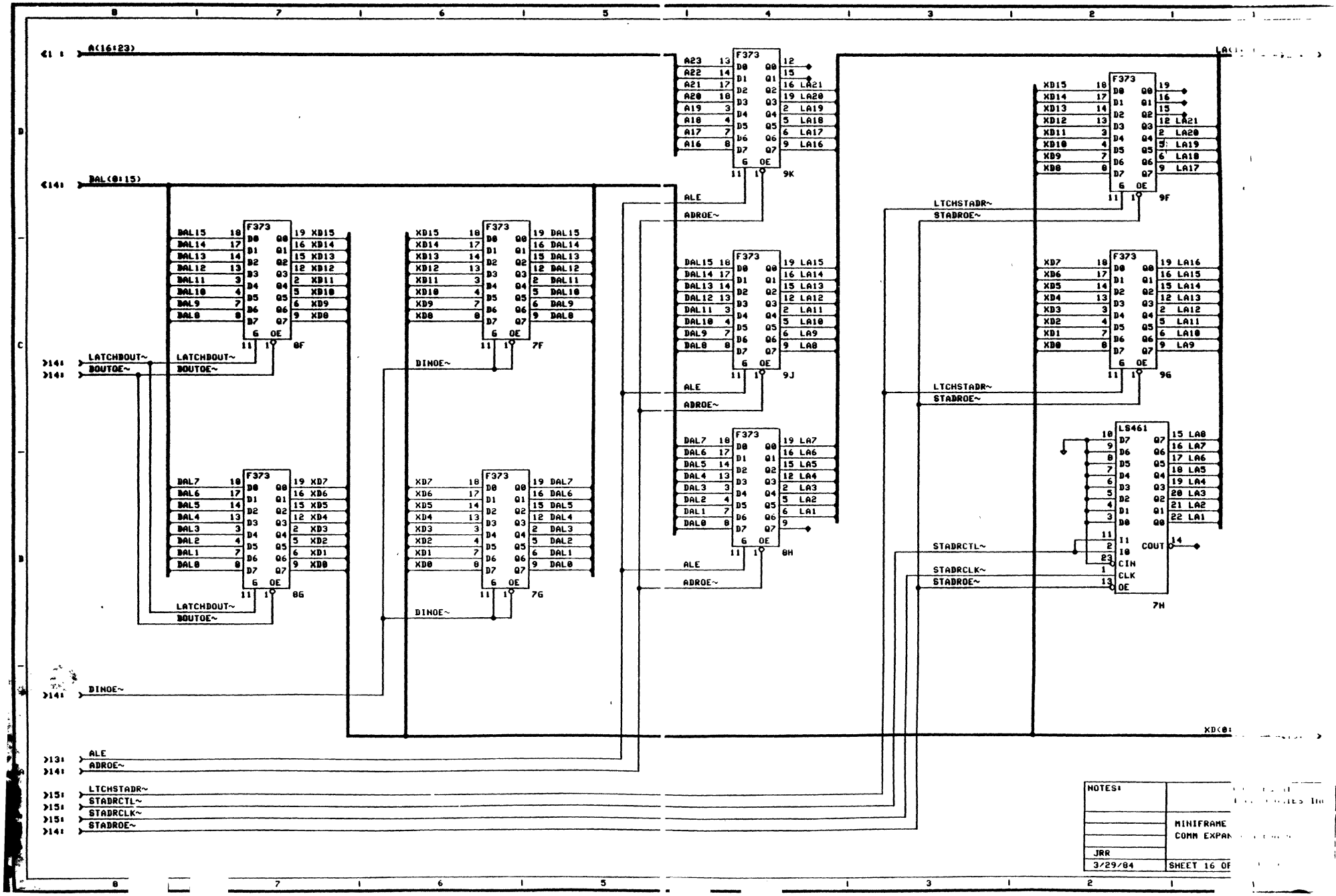

CD(0:3)
TMROUT2
TMROUT3
REGWR2~
TXC2A
J5 -15
RXC2A
J5 -17
TXC2B
J6 -15
RXC2B
J6 -17
1489
N/C
CTXC2A~
CRXC2A~
CTXC2B~
CRXC2B~
PAL265A
20X4
CD3
CD2
CD1
CD0
CLK
OE
100
RP1
TXC2A~
RXC2A~
TXC2B~
RXC2B~
(SEL2A~)
(SEL2B~)
SCISTD2A~
SCISTD2B~
SECONDARY TRANSMIT /
CLOCK SELECT REGISTER II
NOTES:
MINIFRAME
LCL
3/29/84


AM7990
AM7991
A(16:23)
DAL(0:15)
TRANS
TRANS~
RCVR
RCVR~
COL
COL~
TEST~
CS~
ERESET~
READ
ADR
DAS~
HLDA~
READY~
Y1 20 MHZ
NOTES:
MINIFRAME
COMM EXPAN
ETHERNET
JRR
3/29/84
SHEET 10 OF



NOTES:	
	MINIFRAME ETHERNET E[illegible] BOARD
JRR	
3/29/84	SHEET 14 OF [illegible]

NOTES:	CONTINENTAL TECHNOLOGIES INC
	MINIFRAME
	ETHERNET E
JRR	
3/29/84	SHEET 15 OF


A(16:23)
DAL(0:15)
LATCHBOUT~
BOUTOE~
DINOE~
ALE
ADROE~
LTCHSTADR~
STADRCTL~
STADRCLK~
STADROE~
XD(0:
F373
LS461
8F
7F
9K
9J
9F
9G
8H
8G
7G
7H
MINIFRAME
COMM EXPAN
JRR
3/29/84
SHEET 16 OF