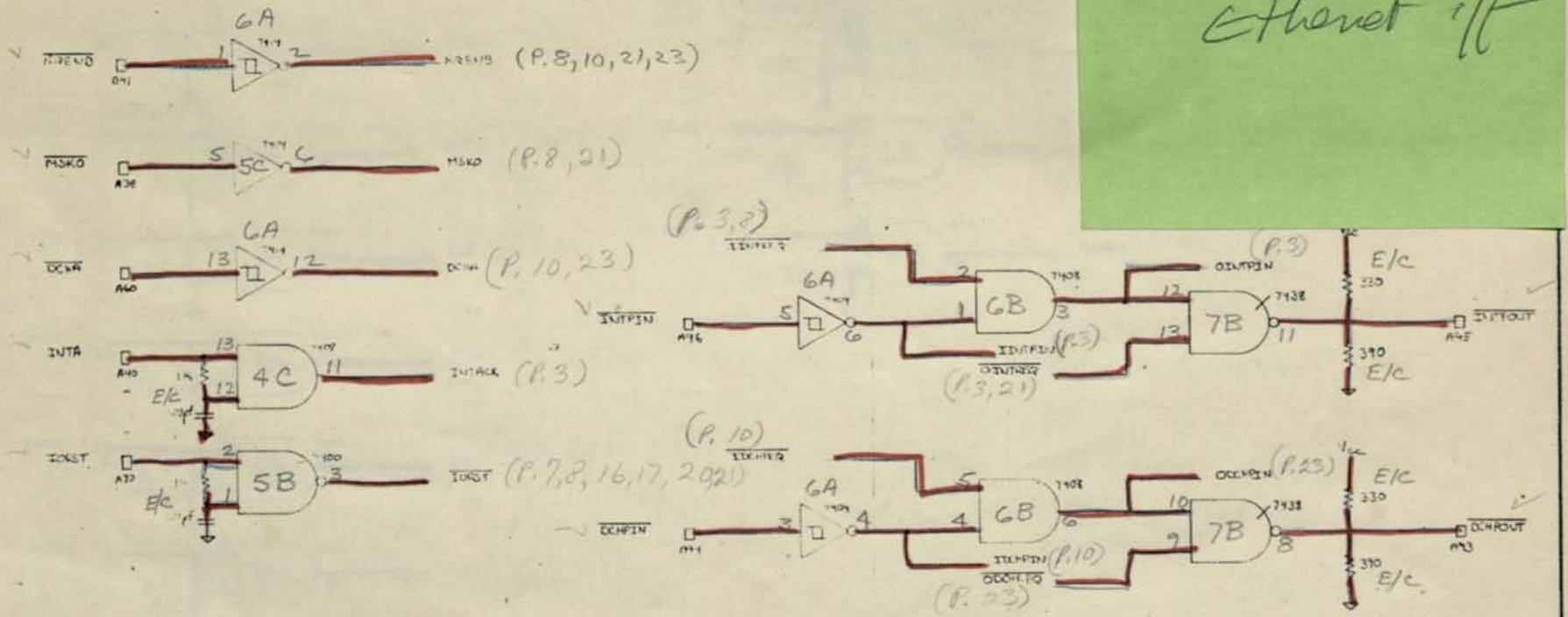
SN1DO SENT TO U.MR

Nova PCB
Nova
Ethernet I/F



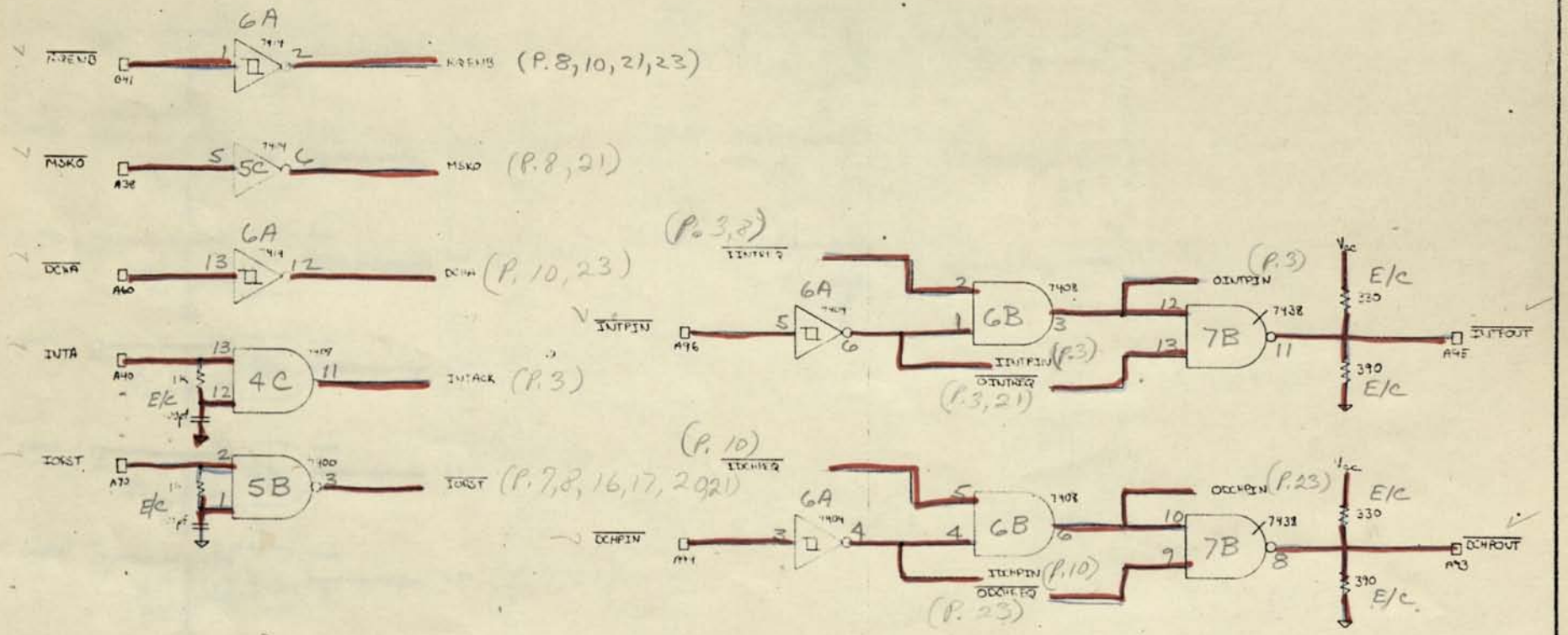
check for 1200 comp.

+5 = ~~A3~~, ~~A4~~, ~~A91~~, ~~A98~~, ~~B3~~, ~~B4~~, ~~B97~~, ~~B98~~

GND = ~~A1~~, ~~A2~~, ~~A34~~, ~~A94~~, ~~B99~~, ~~100~~, ~~B1~~, ~~B2~~, ~~B50~~, ~~B93~~, ~~B99~~, ~~B100~~

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY D
DATE:		REVISED 5/1/79
COMMON INTERFACE LOGIC		
		DRAWING NUMBER 1 of 32

SN100 SENT TO U. of R.

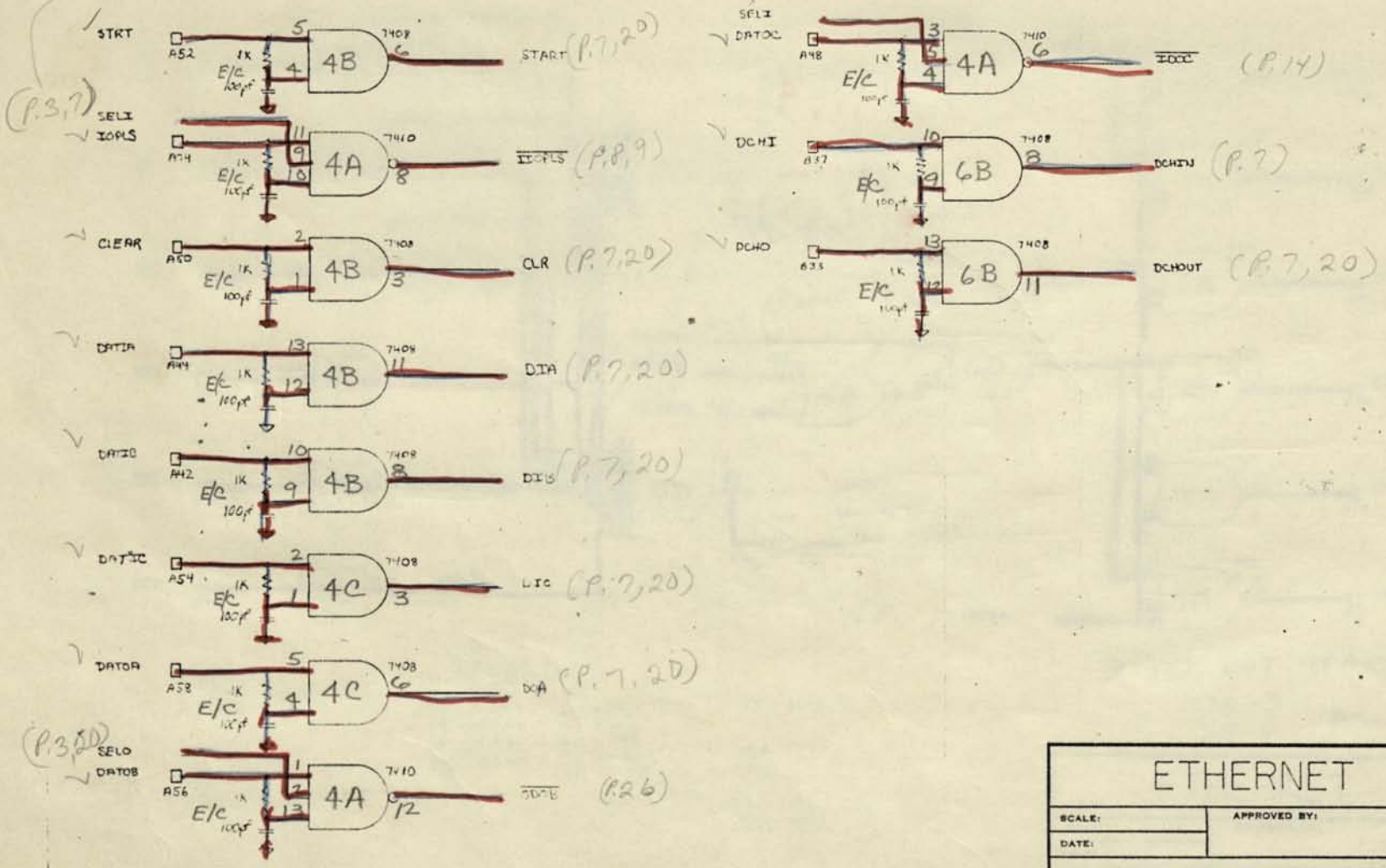


Check for 1200 Comp.

+5 = ~~A3, A4, A97, A98, B3, B4, B97, B98~~

GND = ~~A1, A2~~ A33, ~~A34, A99, 100, B1, B2, B30~~ B92, ~~B99, B100~~

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY D
DATE:		REVISED 5/7/74
COMMON INTERFACE LOGIC		
		DRAWING NUMBER 1 of 32

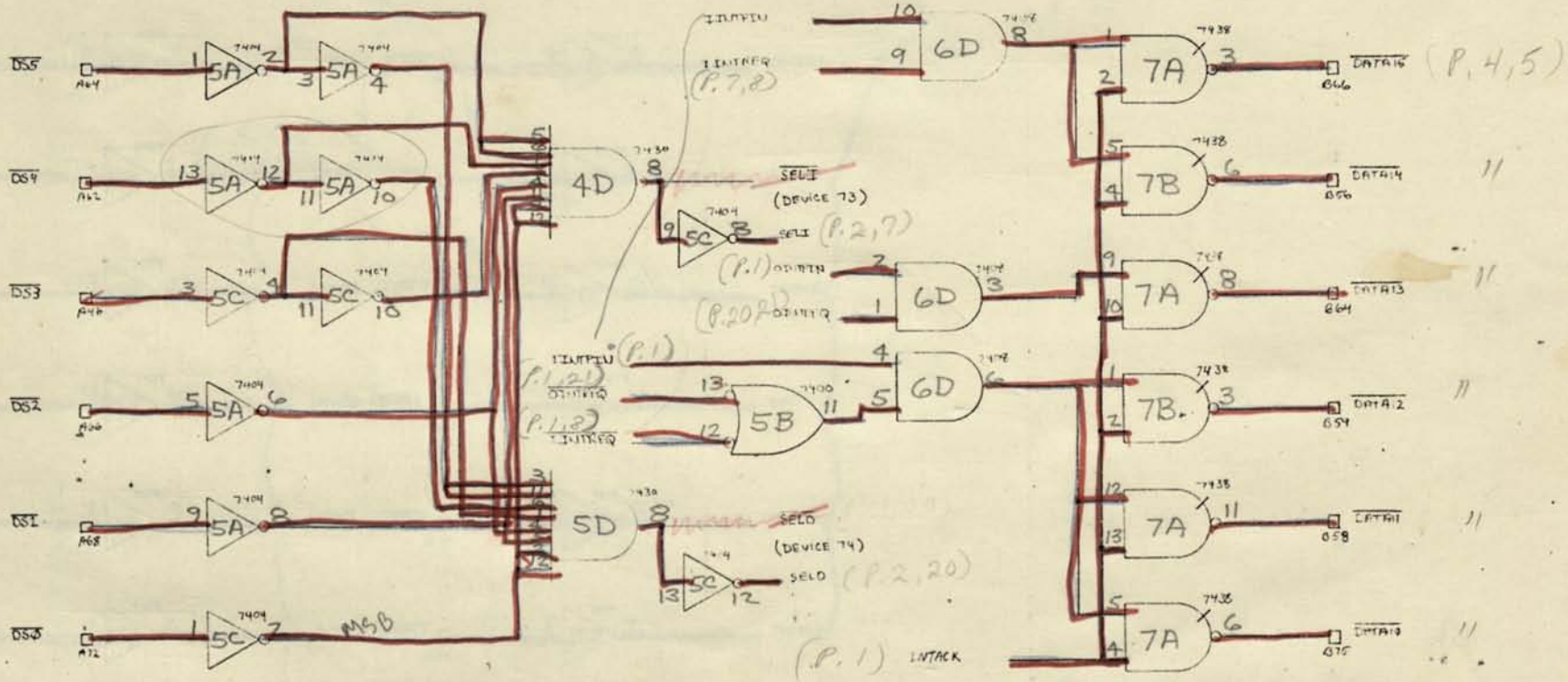


ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED

MORE INTERFACE LOGIC

DRAWING NUMBER 20/32



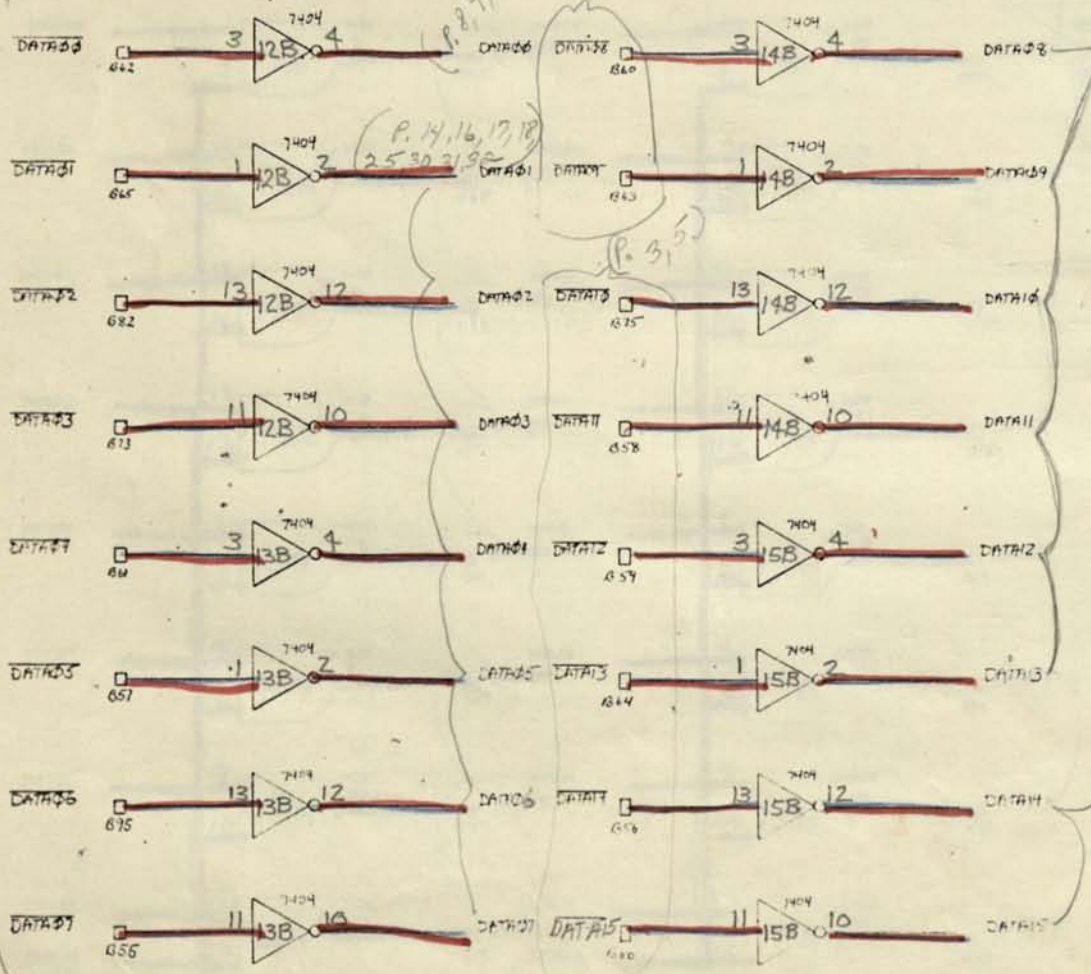
DEVICE CODE SELECT

15 - 4D#4
14 - 4D#11
11 - 5C#1
12 - 5C#7
6 - 4D#12
4 - 5A#8
3 - 5A#9
2 - 5A#5
1 - 5A#6

15 - 7A#3
1 -
14 - 7A#11
4 -
12 -
6 - 7A#6

SN 122 CUT TRACES @
7B-3
7A-1
Device 43 - ICEMAN

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 6/1/74
DEVICE SELECT LOGIC		
		DRAWING NUMBER 3-1/32

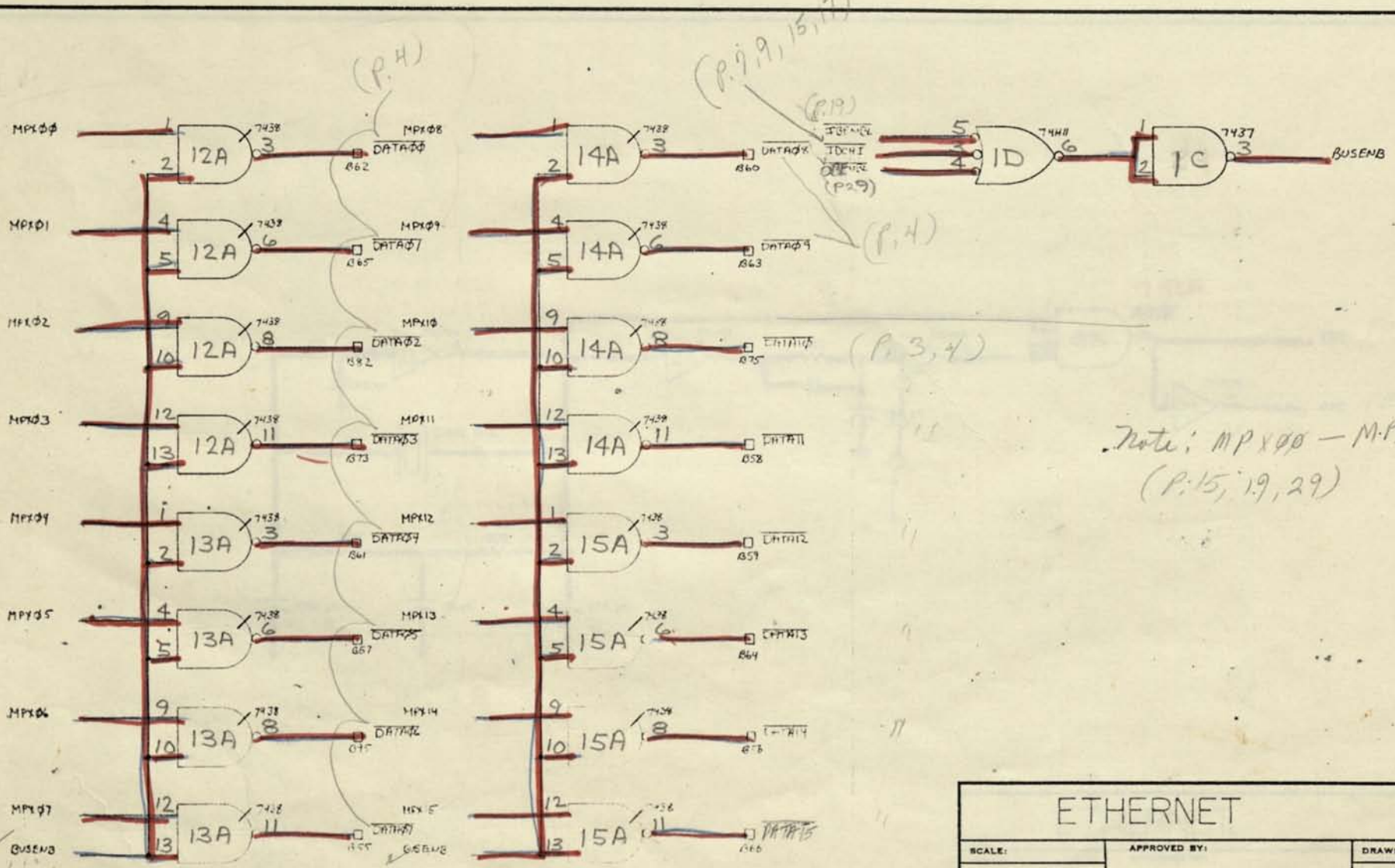


ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED

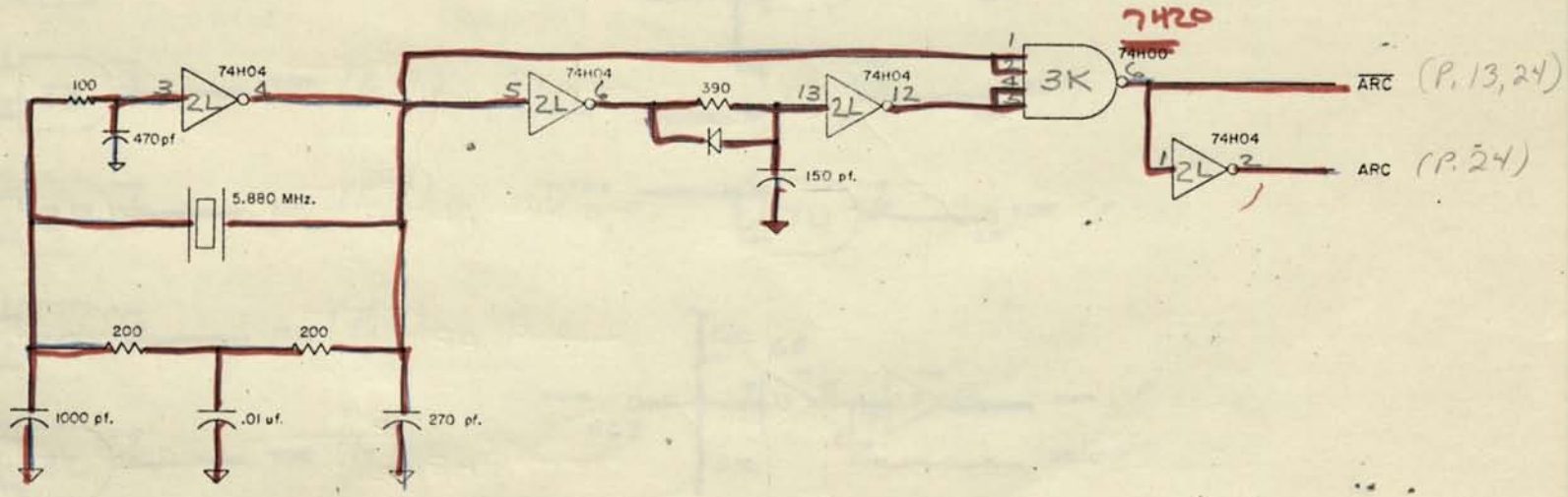
NOVA BUS RECEIVERS

DRAWING NUMBER 4 of 32

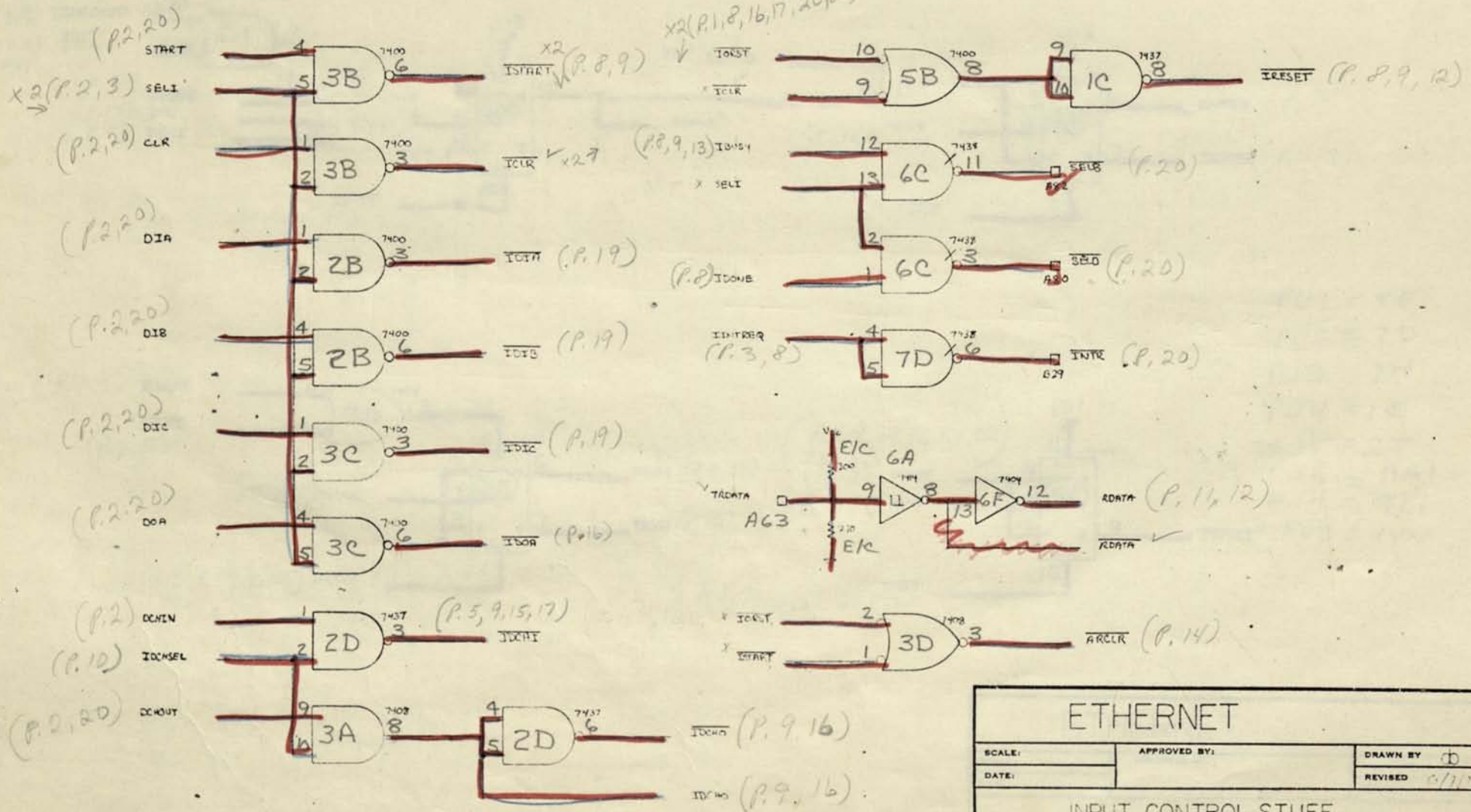


Note: MPXØØ — MPX15
(P.15, 19, 29)

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY QC
DATE:		REVISED 6/7/74
NOVA BUS DRIVERS		
		DRAWING NUMBER 5-1132



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 1/11/4
ARC		
		DRAWING NUMBER 6 of 32

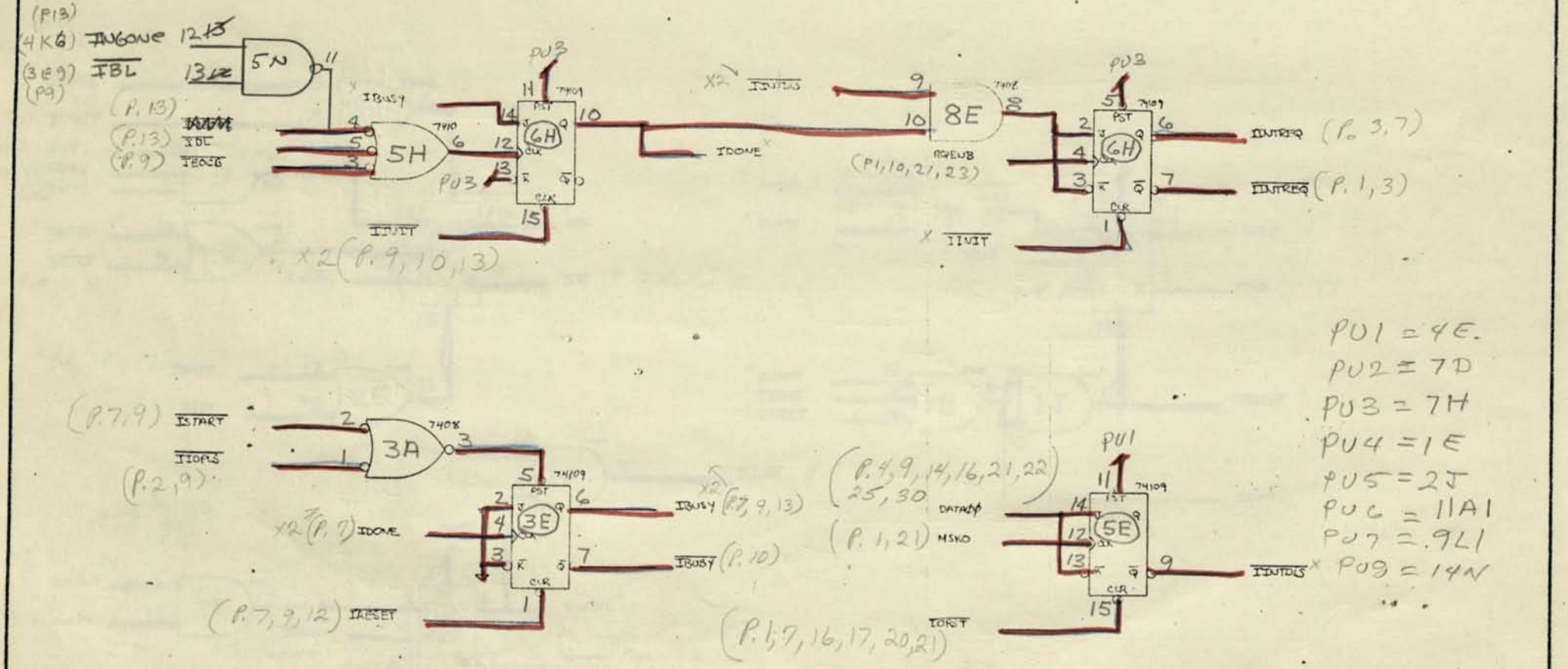


ETHERNET

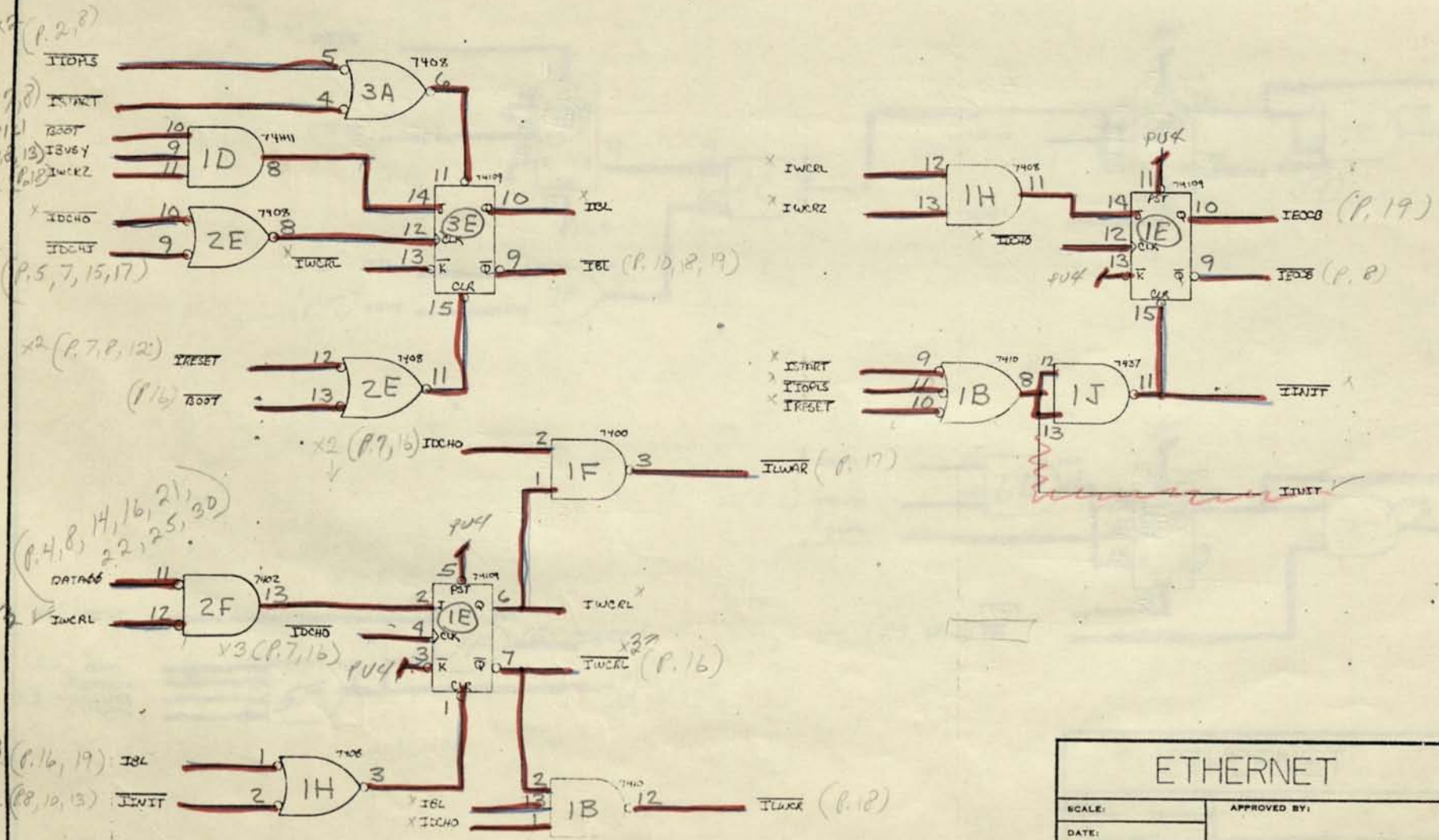
SCALE:	APPROVED BY:	DRAWN BY D
DATE:		REVISED 6/7/74

INPUT CONTROL STUFF

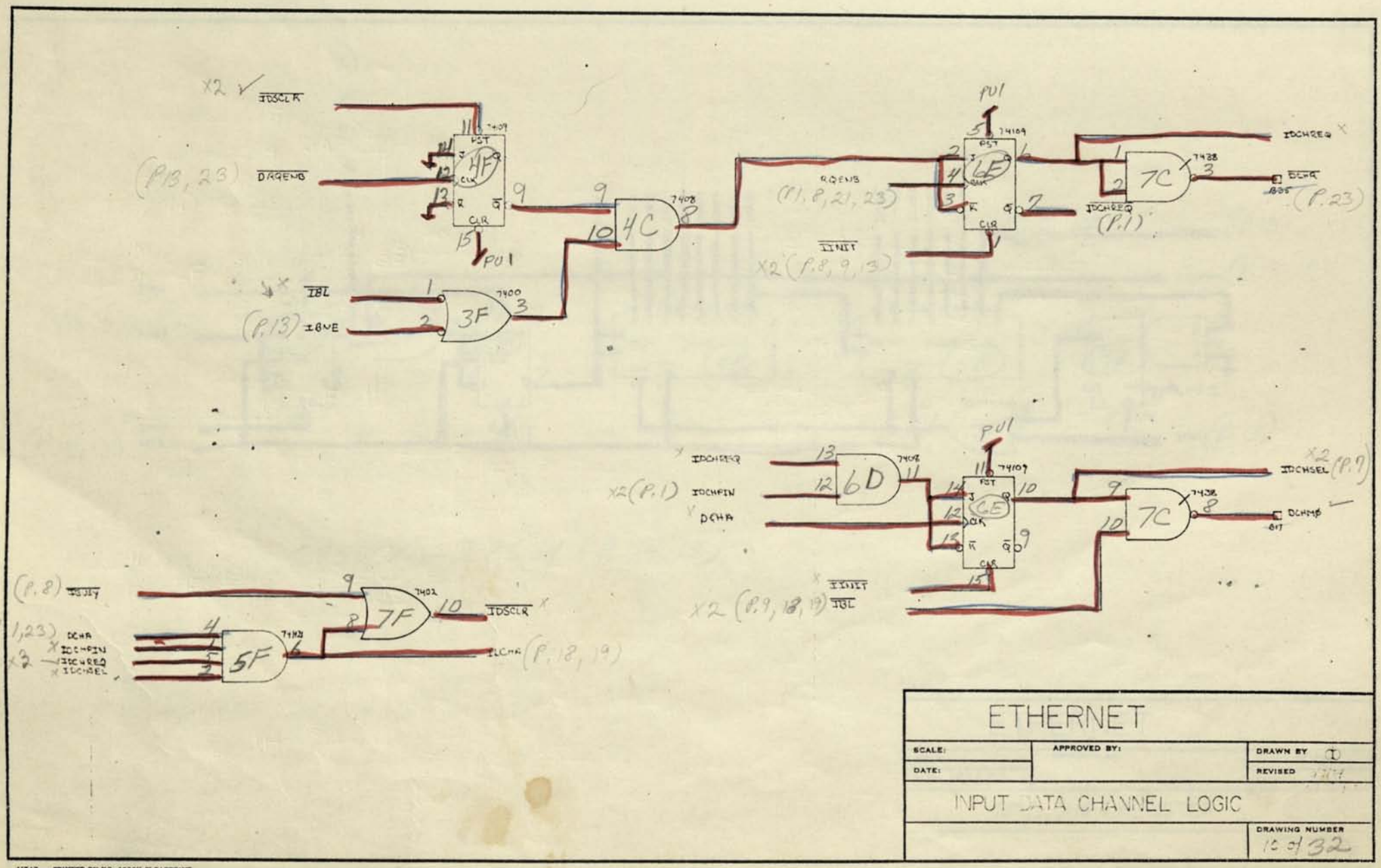
DRAWING NUMBER 7432



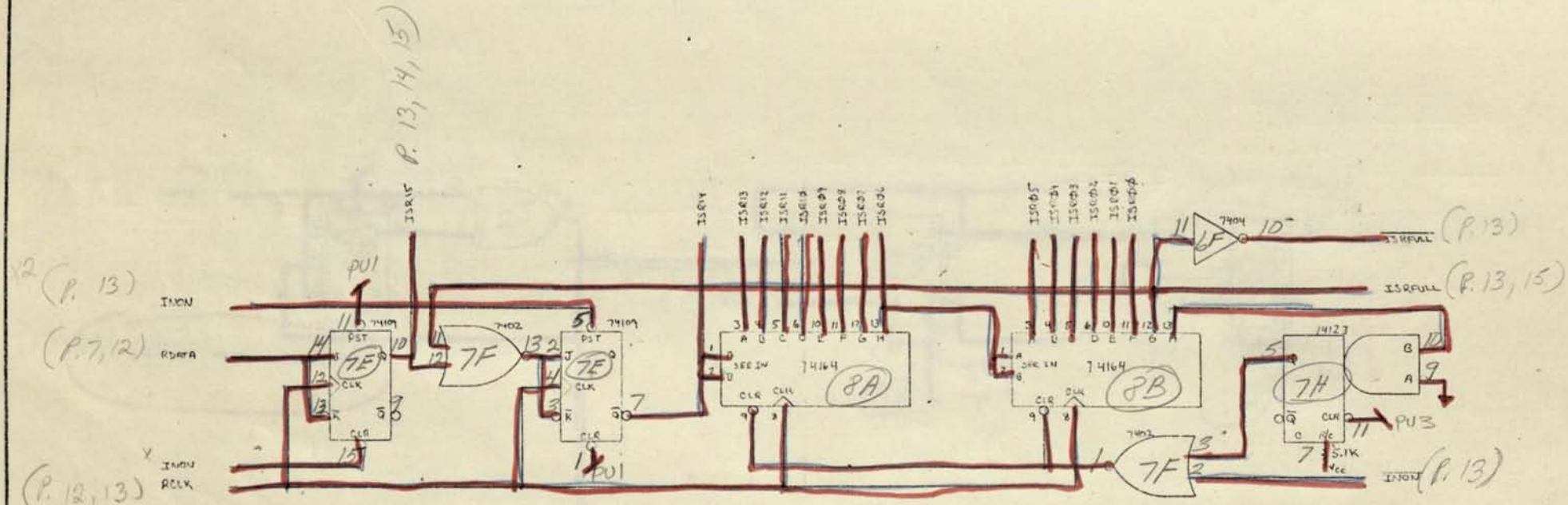
ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED
INPUT INTERRUPT LOGIC		
		DRAWING NUMBER 8932



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/7/74
INPUT BLOCK LINK LOGIC		
		DRAWING NUMBER 9932

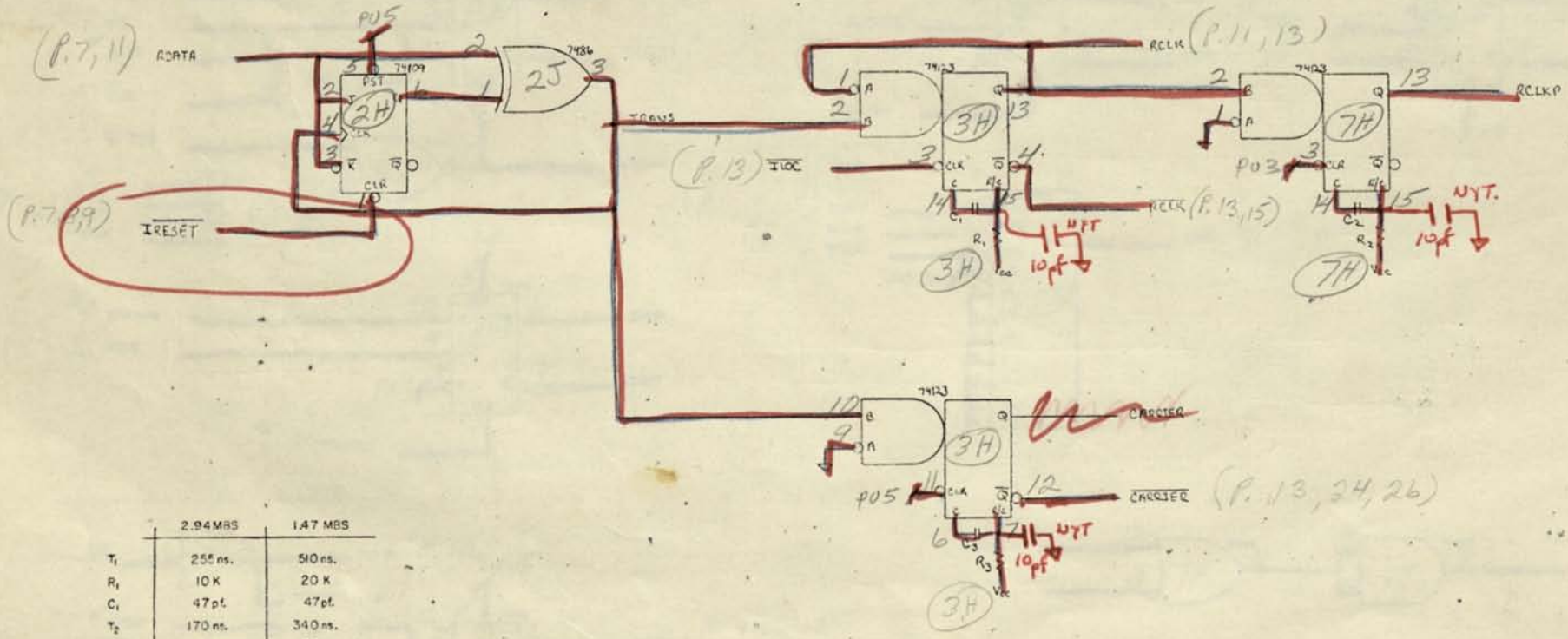


ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED
INPUT DATA CHANNEL LOGIC		
		DRAWING NUMBER 10 of 32



IRS00 — IRS14 (P. 14, 15)

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY db
DATE: 1/2/74		REVISED 6/3/74
INPUT SHIFT REGISTER		
		DRAWING NUMBER 11 of 32



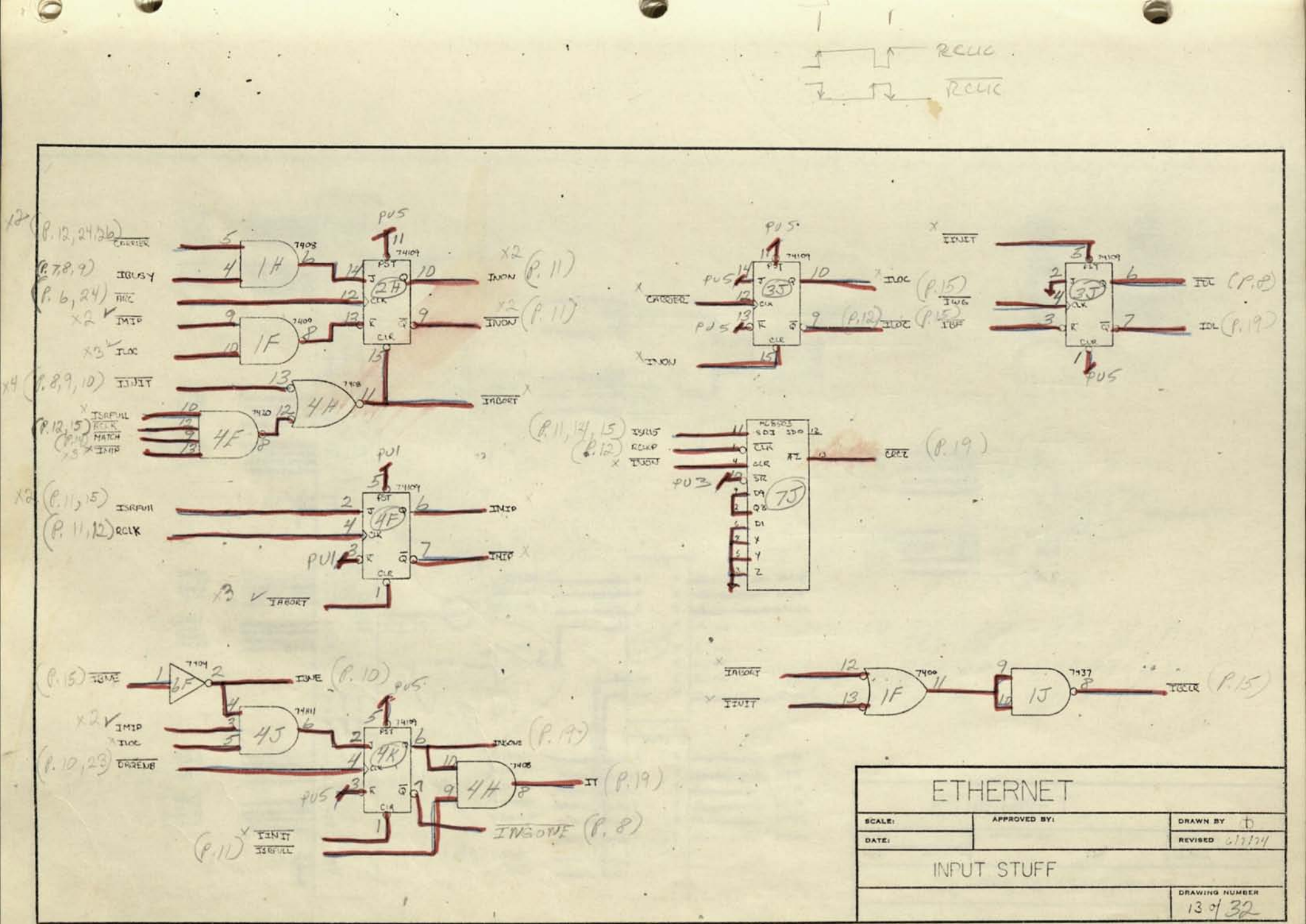
	2.94 MBS	1.47 MBS
T_1	255 ns.	510 ns.
R_1	10 K	20 K
C_1	47 pf.	47 pf.
T_2	170 ns.	340 ns.
R_2	5K	20K
C_2	68 pf.	33 pf.
T_3	425 ns.	850 ns.
R_3	20K	30K
C_3	39 pf.	62 pf.

ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE: 1/2/74		REVISED

PHASE DECODER

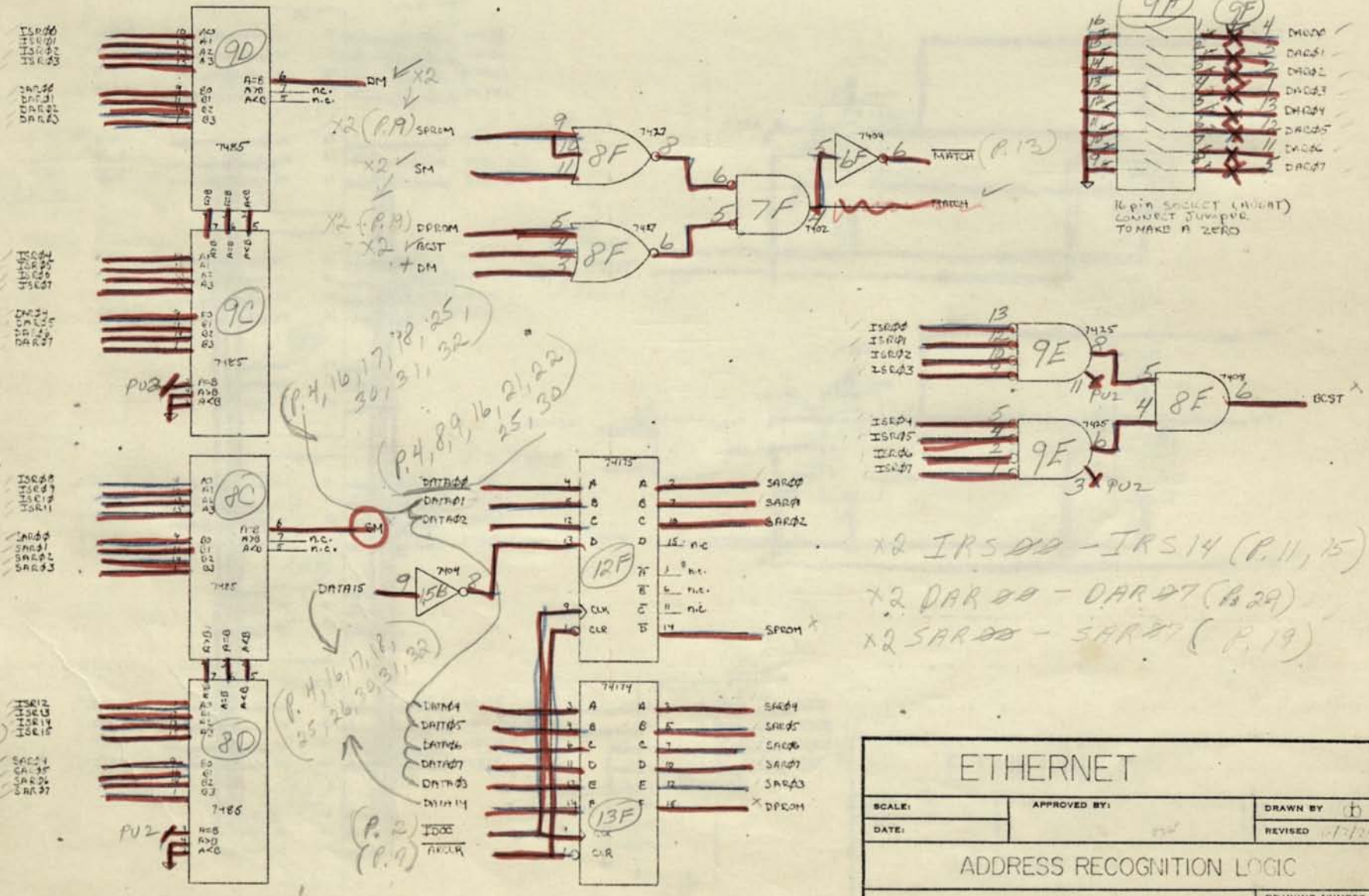
DRAWING NUMBER 12 01 32



ETHERNET	
SCALE:	APPROVED BY:
DATE:	DRAWN BY: D
	REVISED: 6/17/74
INPUT STUFF	
	DRAWING NUMBER: 13 of 32

11X17 PRINTED ON NO. 1000H CLEARPRINT

ISR = 10001001
DAR = 00001001



(P. 4, 16, 17, 28, 25, 30, 31, 32)

(P. 4, 8, 9, 16, 21, 22, 25, 30)

(P. 4, 16, 17, 28, 25, 26, 30, 31, 32)

(P. 11, 13, 15)

(P. 2) (P. 7)

×2 IRS08 – IRS14 (P. 11, 15)

×2 DAR08 – DAR07 (P. 29)

×2 SAR08 – SAR07 (P. 19)

ETHERNET

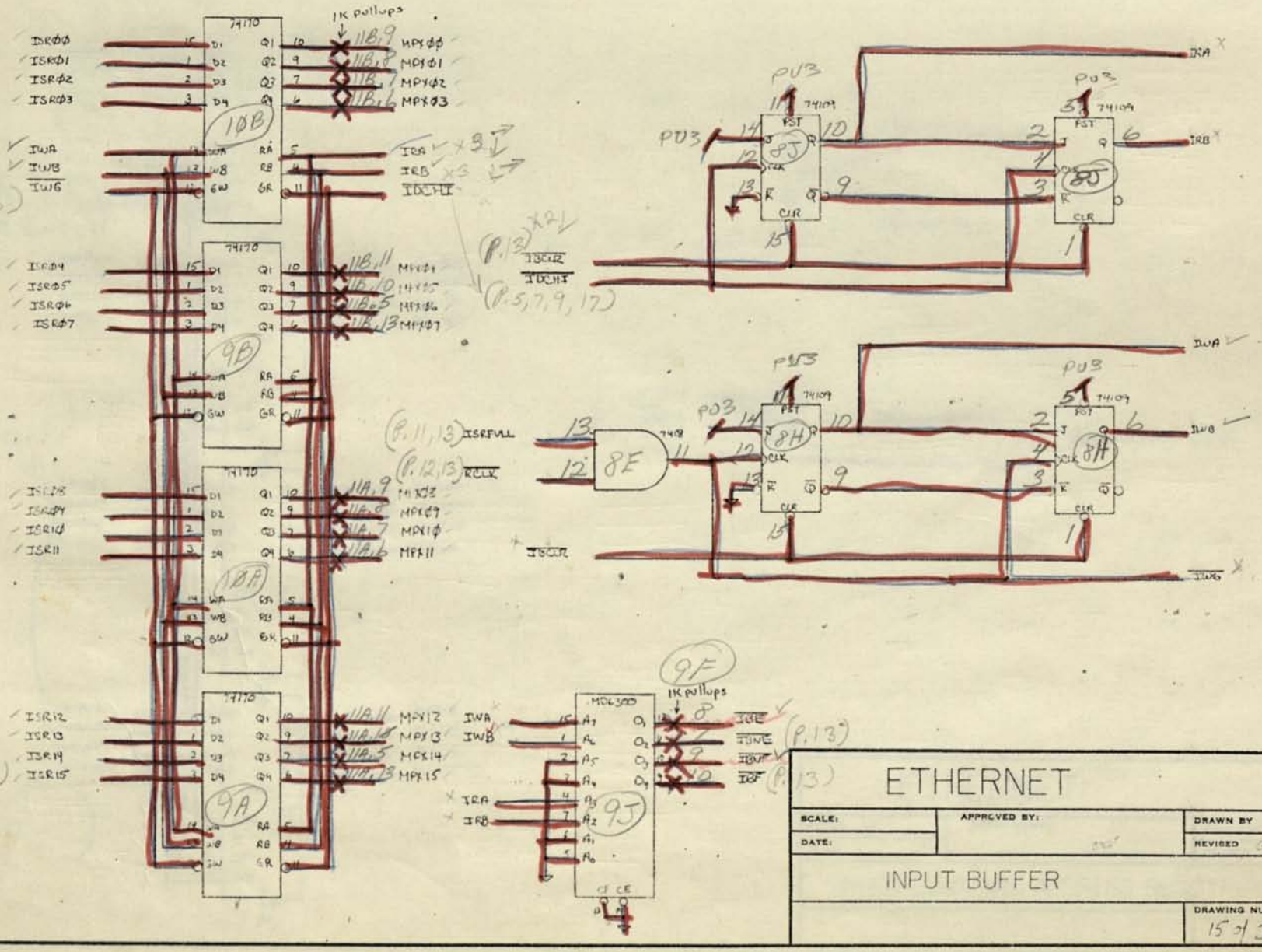
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 1/7/24

ADDRESS RECOGNITION LOGIC

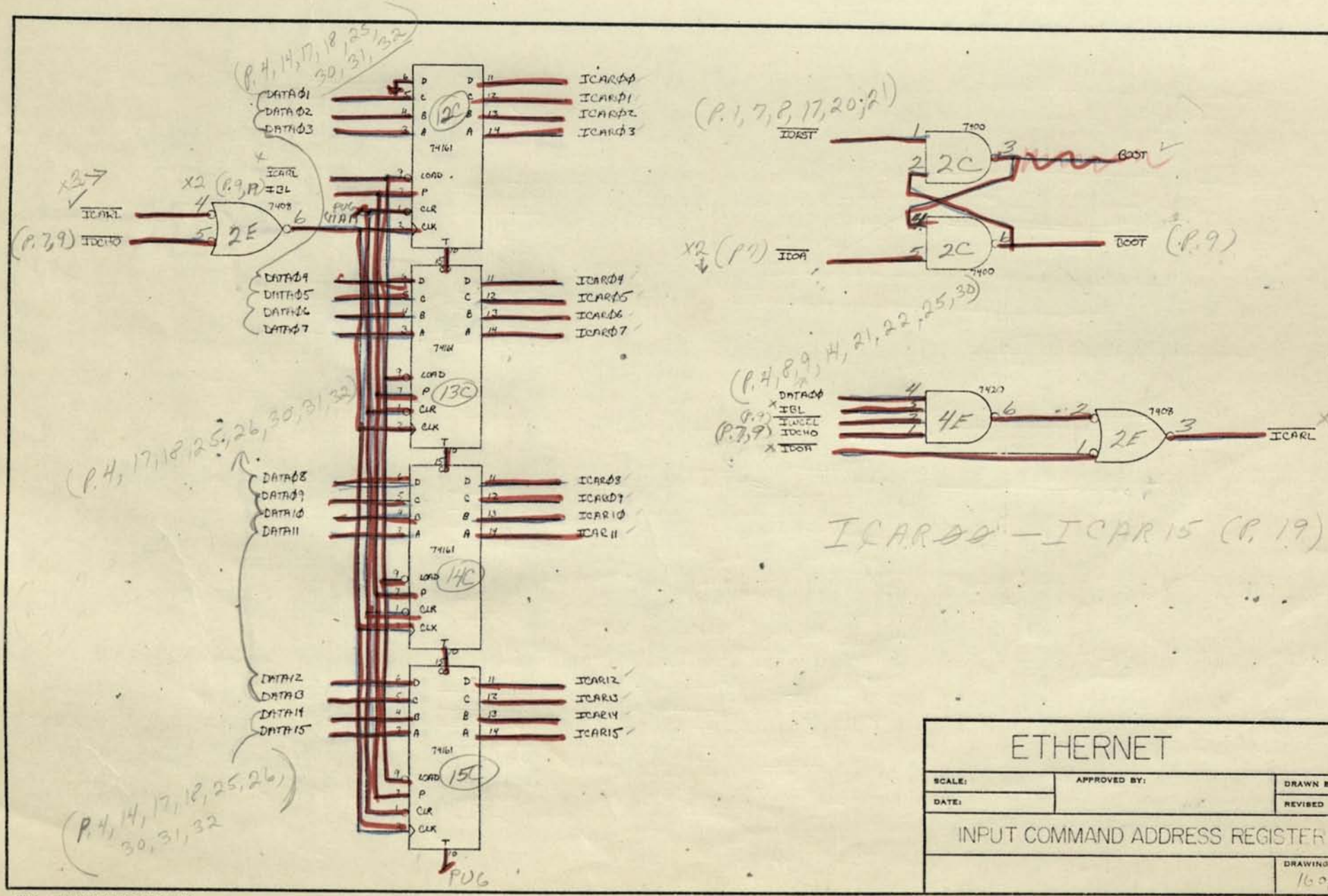
DRAWING NUMBER 14 of 32

IRS00 — IRS15 (P. 11, 13, 14)

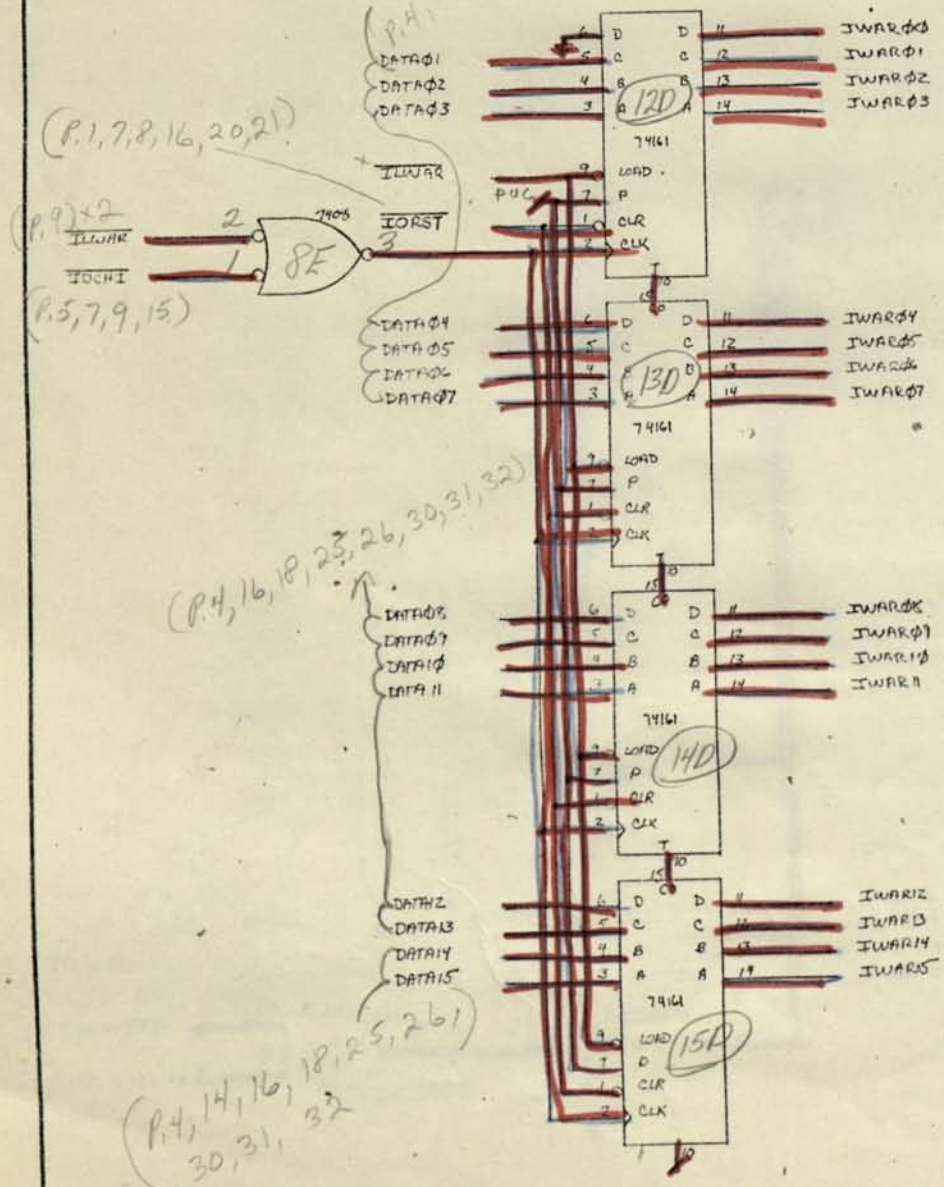
MPX00 — MPX15 (P. 5, 19, 29)



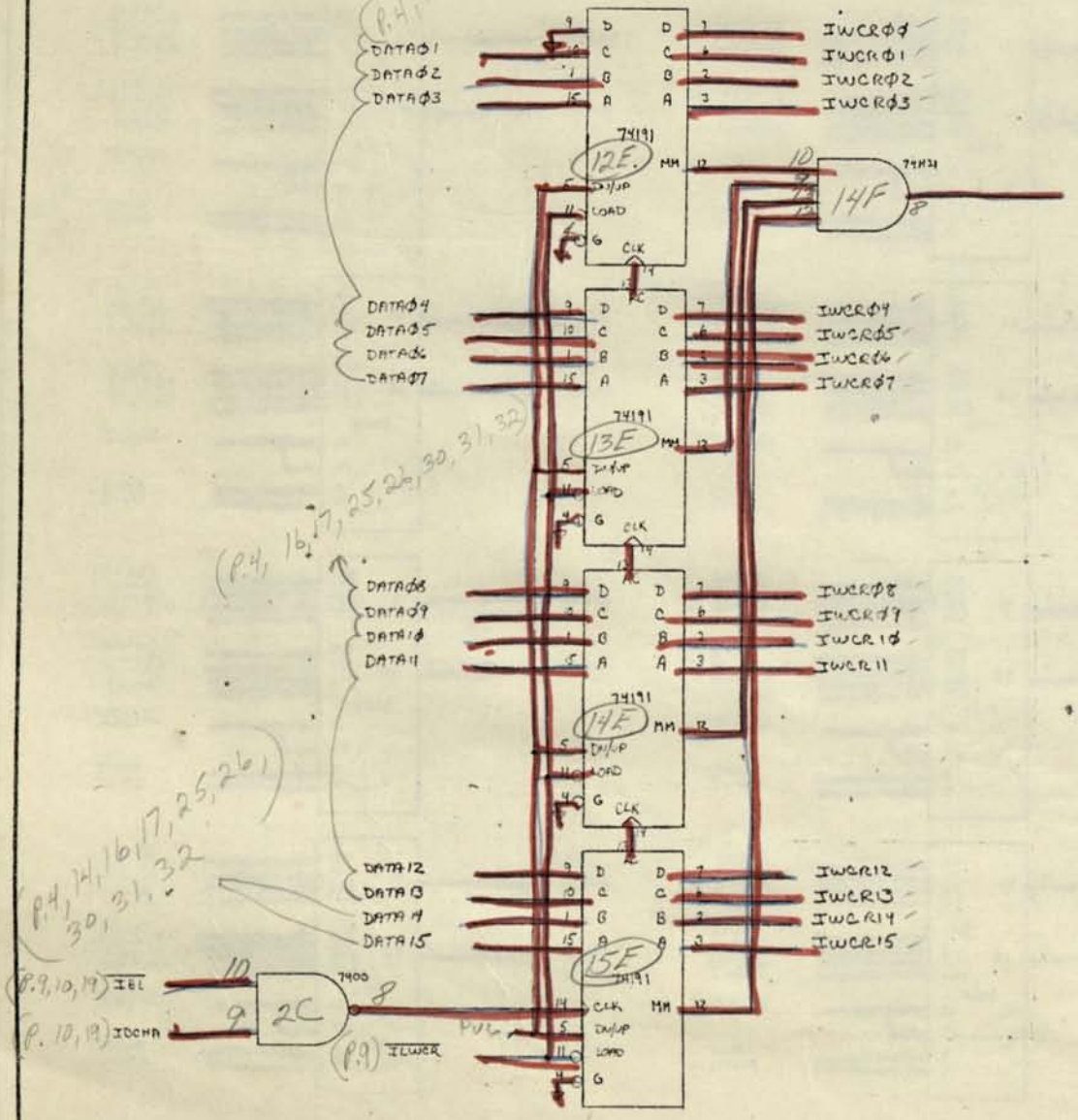
IWAR	0	000	000	000	001	111		
ICAR	0	00	1000	1001	1010			



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED
INPUT COMMAND ADDRESS REGISTER		
		DRAWING NUMBER 16 of 32



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 9/7/14
INPUT WORD ADDRESS REGISTER		
		DRAWING NUMBER 17 of 32

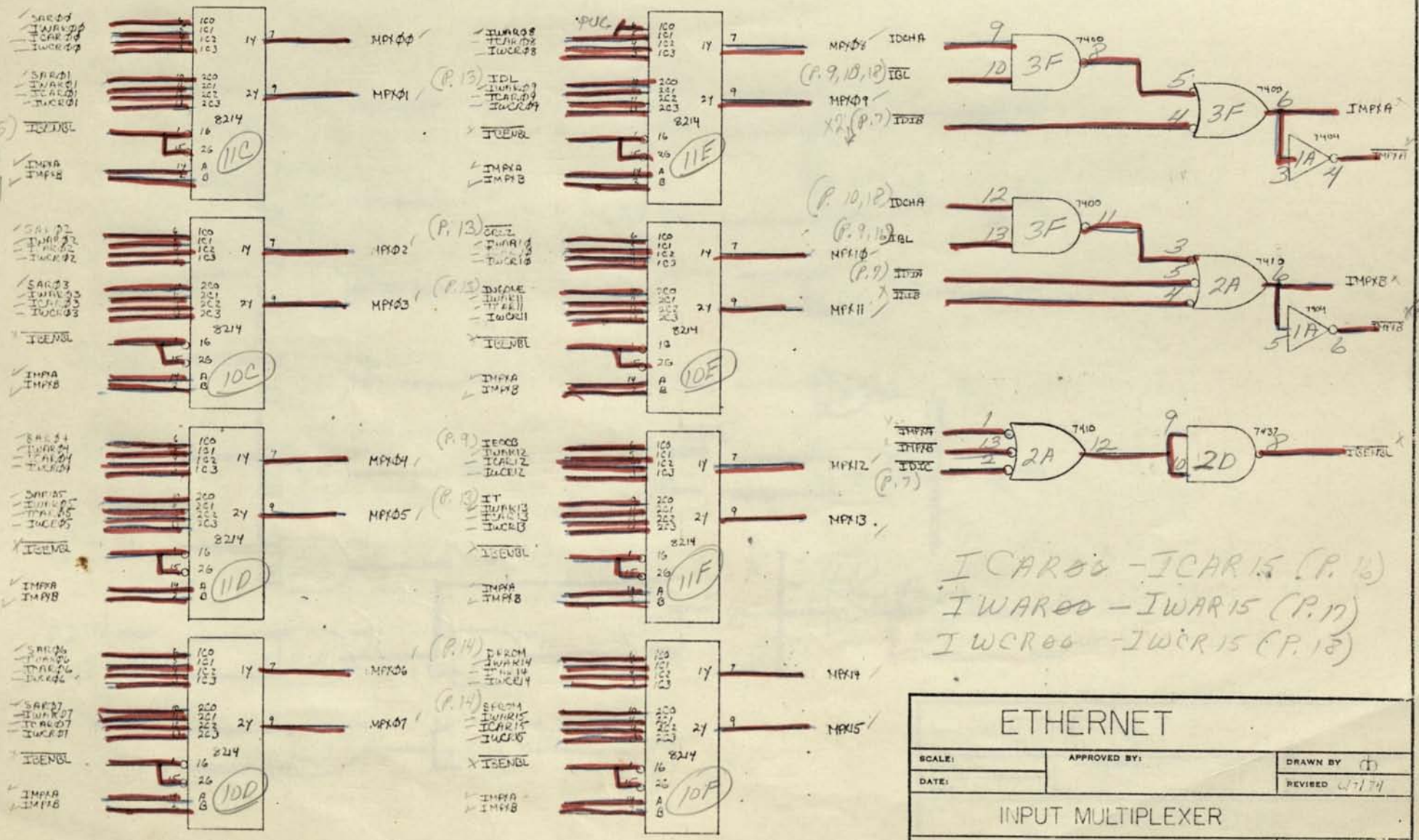


IWCR00 - IWCR15 (P.19)

ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 6/11/74
INPUT WORD COUNT REGISTER		
		DRAWING NUMBER 184 32

MPX00 – MPX15 (P. 5, 15, 29)
SAR00 – SAR07 (P. 14)

A	B	
0	0	STATUS
0	1	CAR
1	0	WAR
1	1	WCR



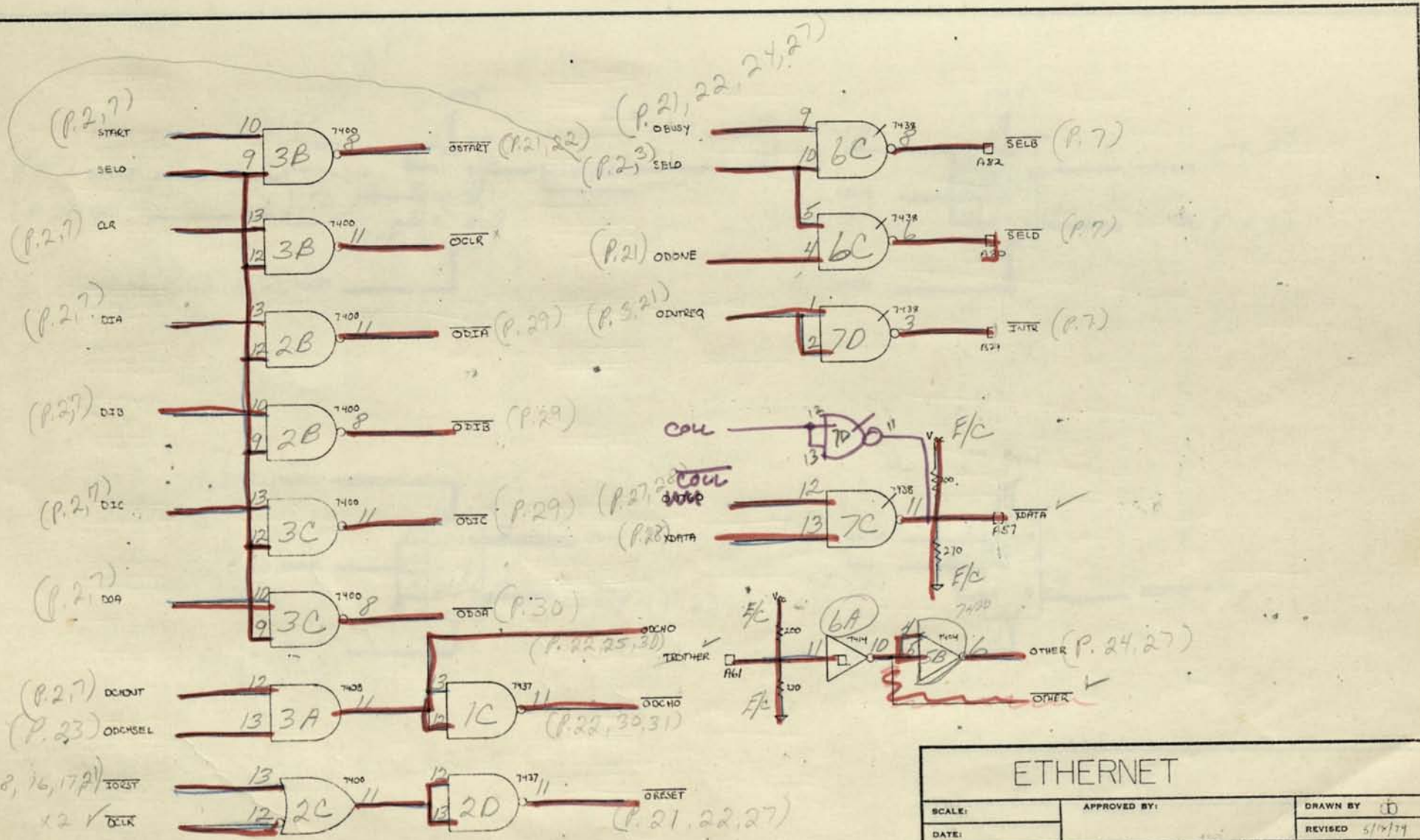
ICAR00 – ICAR15 (P. 16)
IWAR00 – IWAR15 (P. 17)
IWCR00 – IWCR15 (P. 18)

ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 4/1/74

INPUT MULTIPLEXER

DRAWING NUMBER 19 of 32

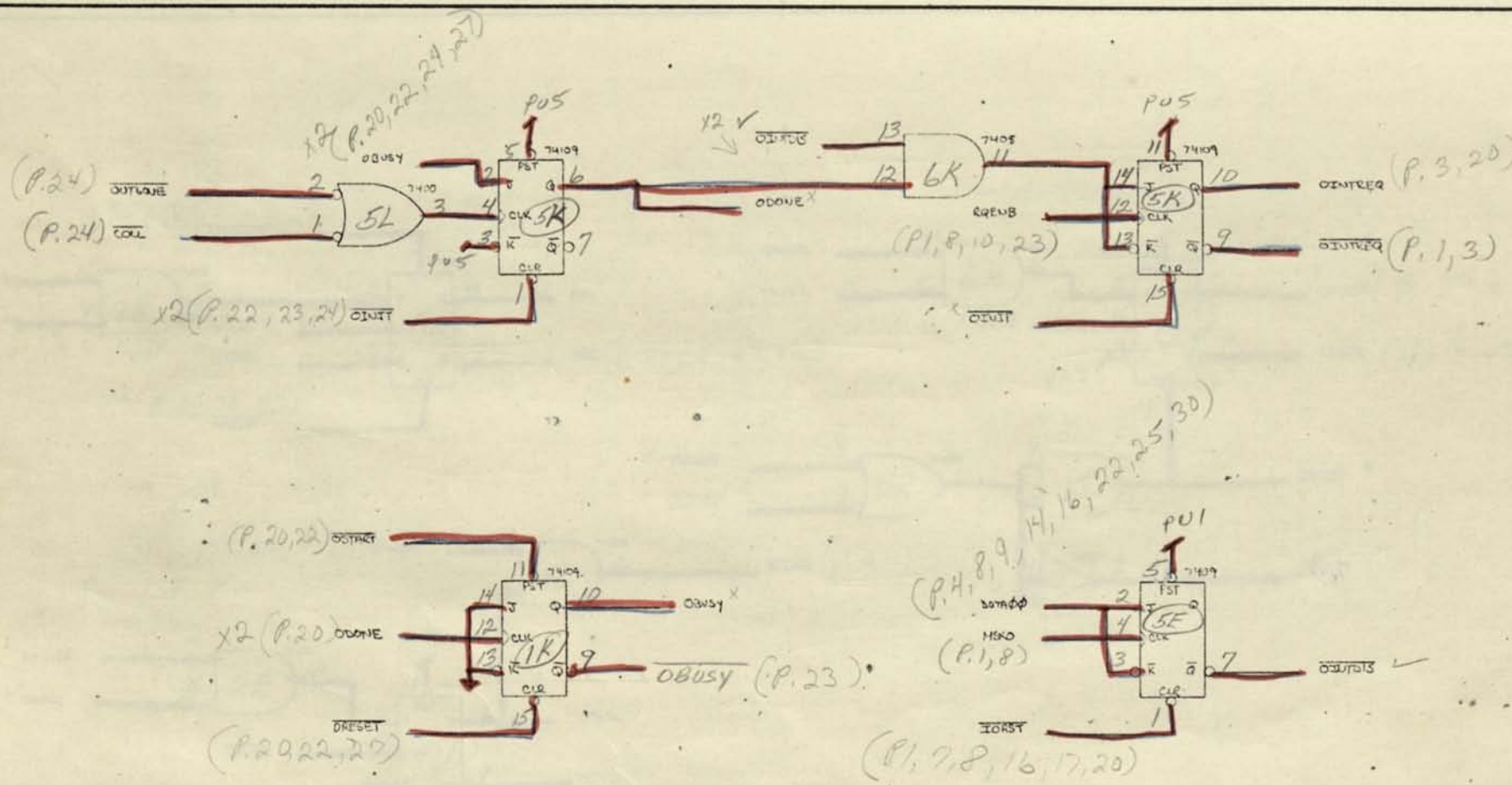


ETHERNET

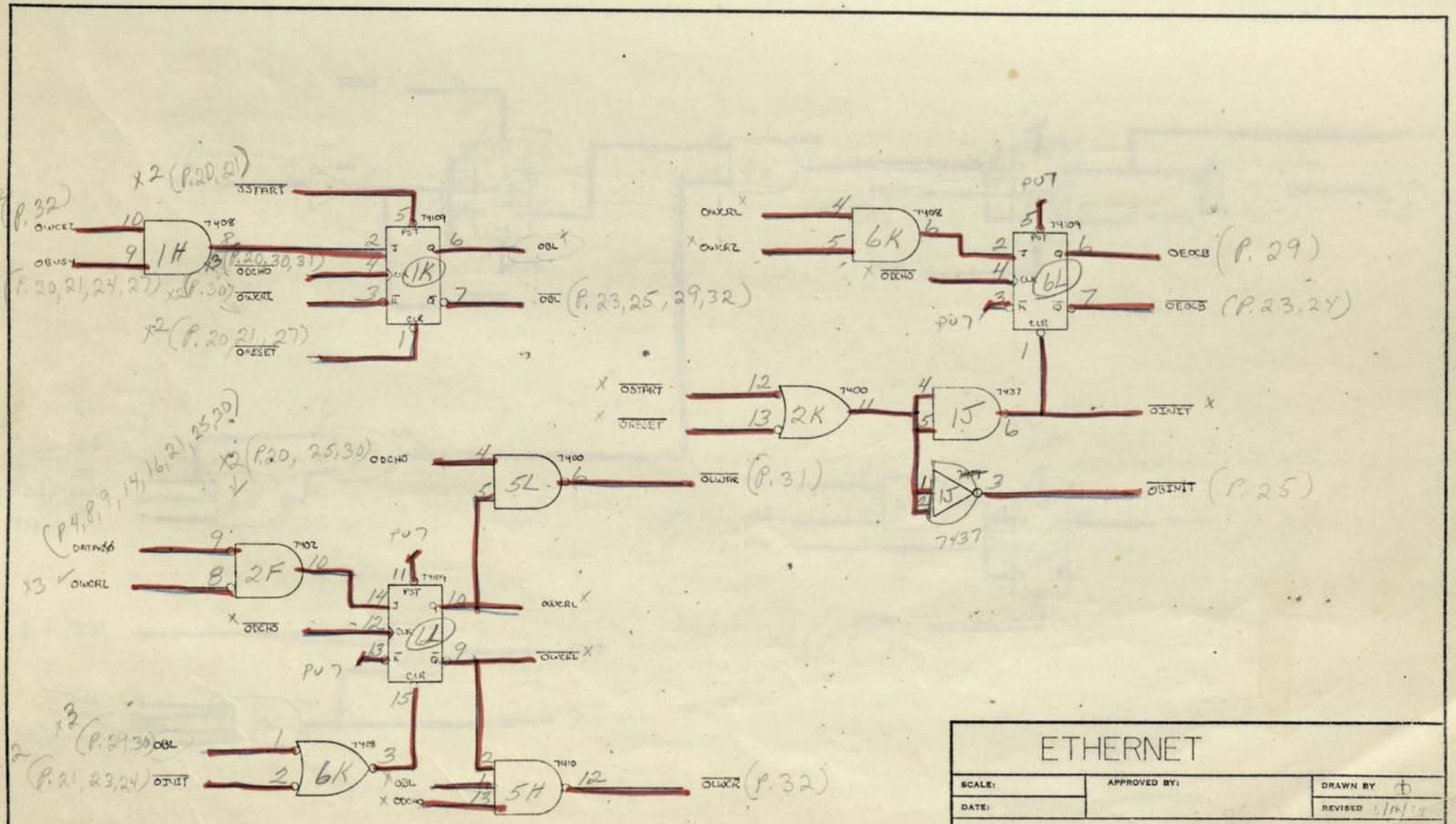
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/15/79

OUTPUT CONTROL STUFF

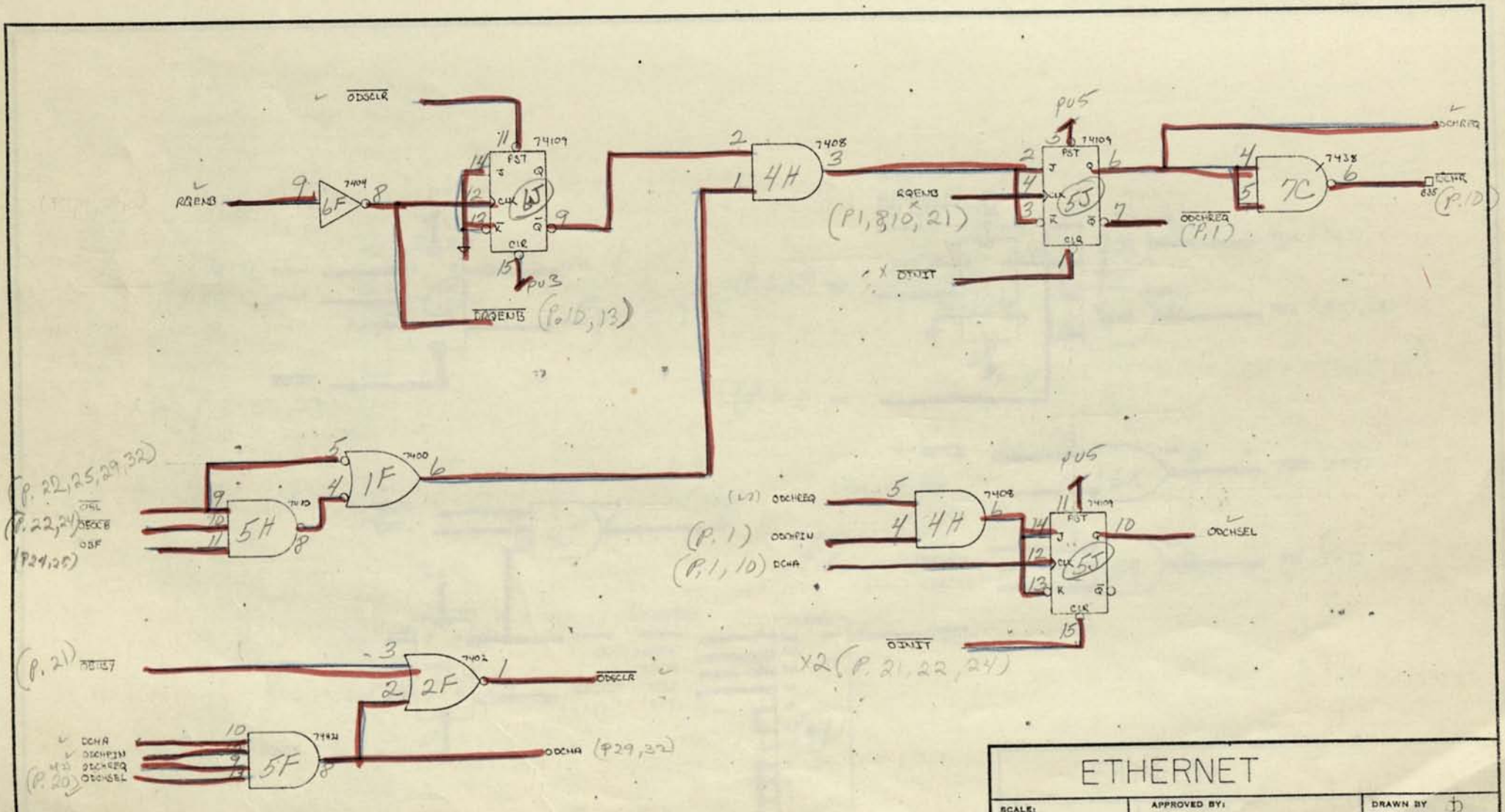
DRAWING NUMBER 20 of 32



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/1/19
OUTPUT INTERRUPT LOGIC		
		DRAWING NUMBER 21 of 32



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 1/14/74
OUTPUT BLOCK LINK LOGIC		
		DRAWING NUMBER 22 of 32



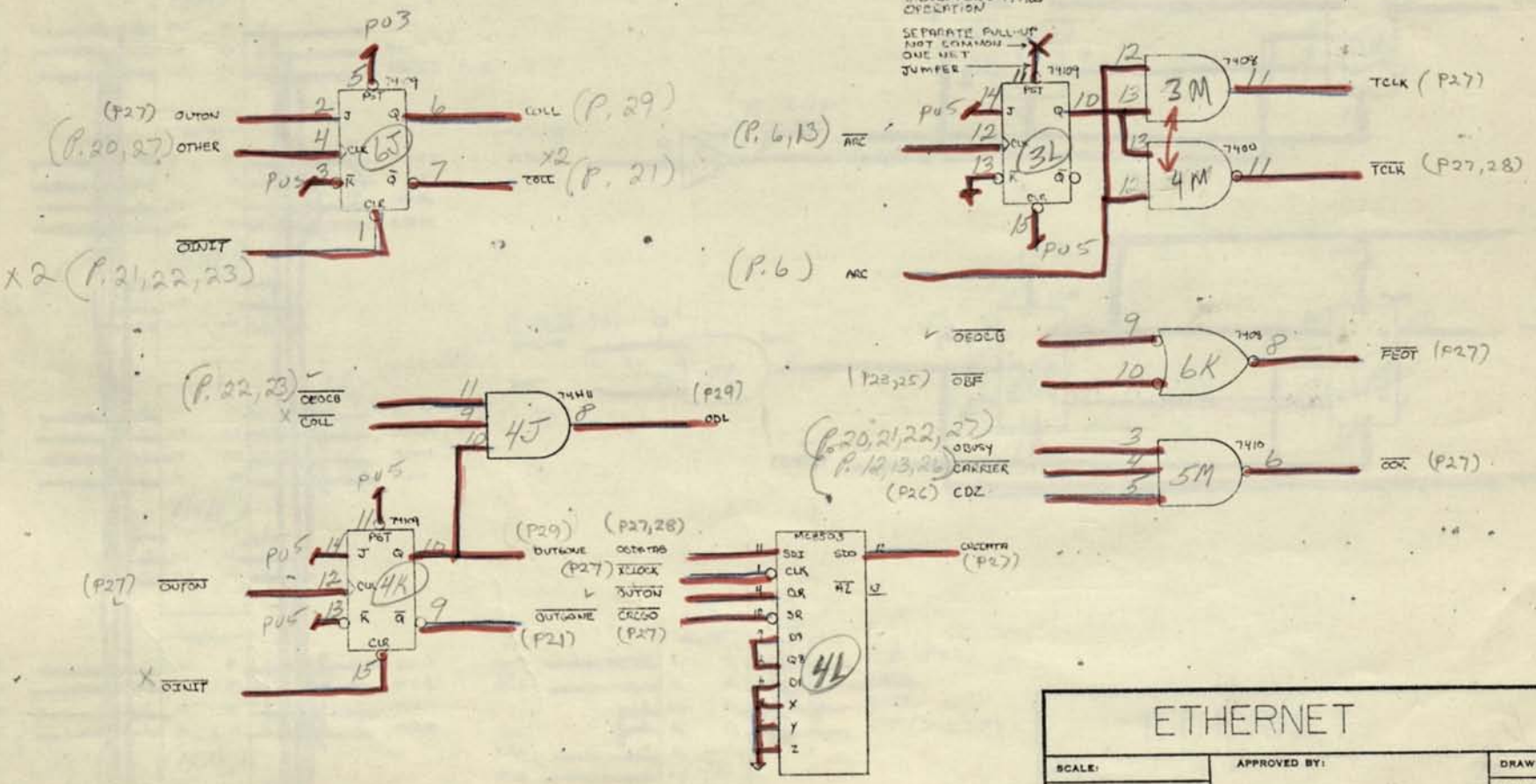
ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 6/7/84

OUTPUT DATA CHANNEL LOGIC

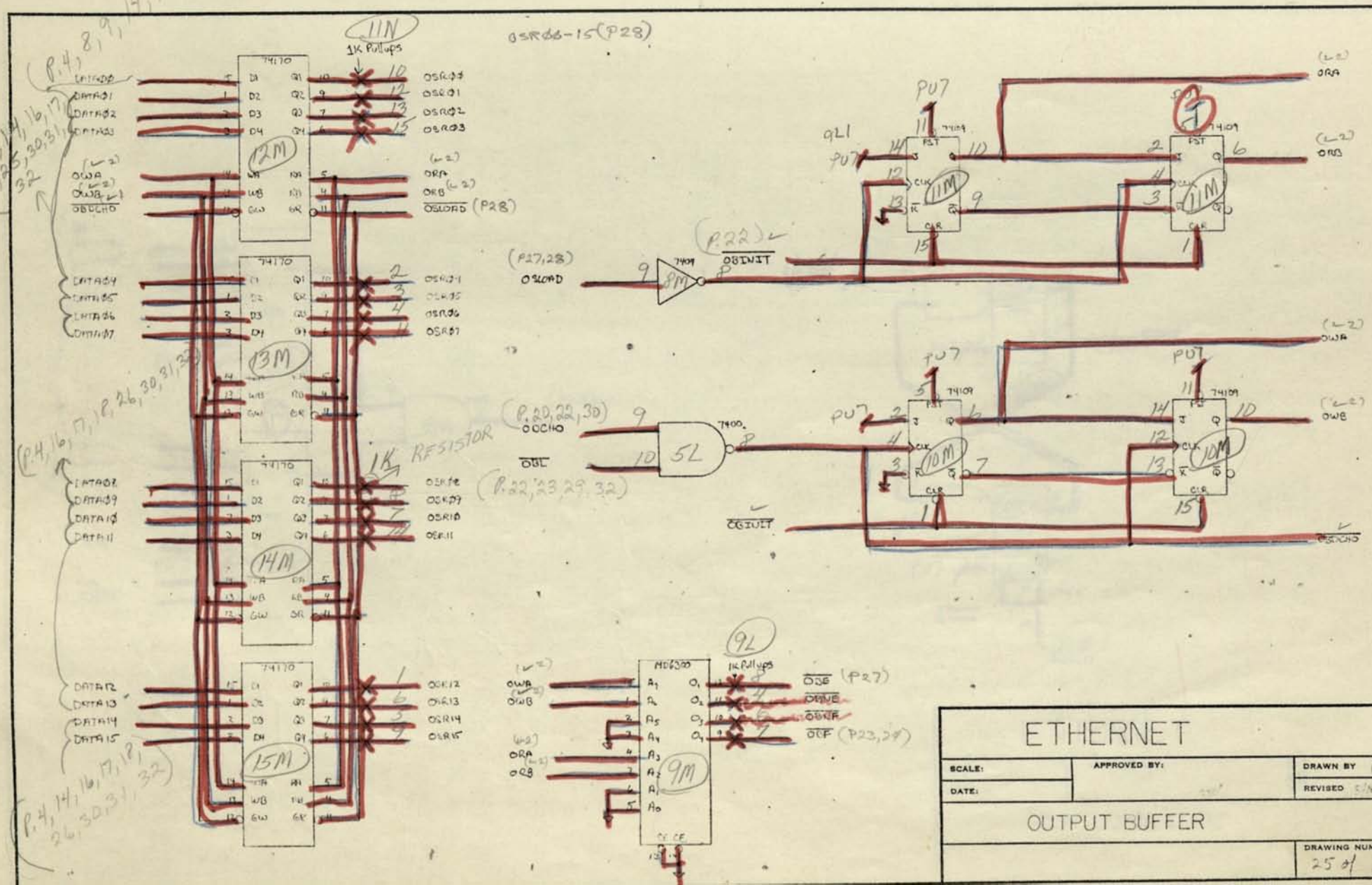
DRAWING NUMBER 23-1 32

4M=NAND
3M=AND



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/14/74
OUTPUT STUFF		
		DRAWING NUMBER 24 of 32

WRITE = 00
Read = 01

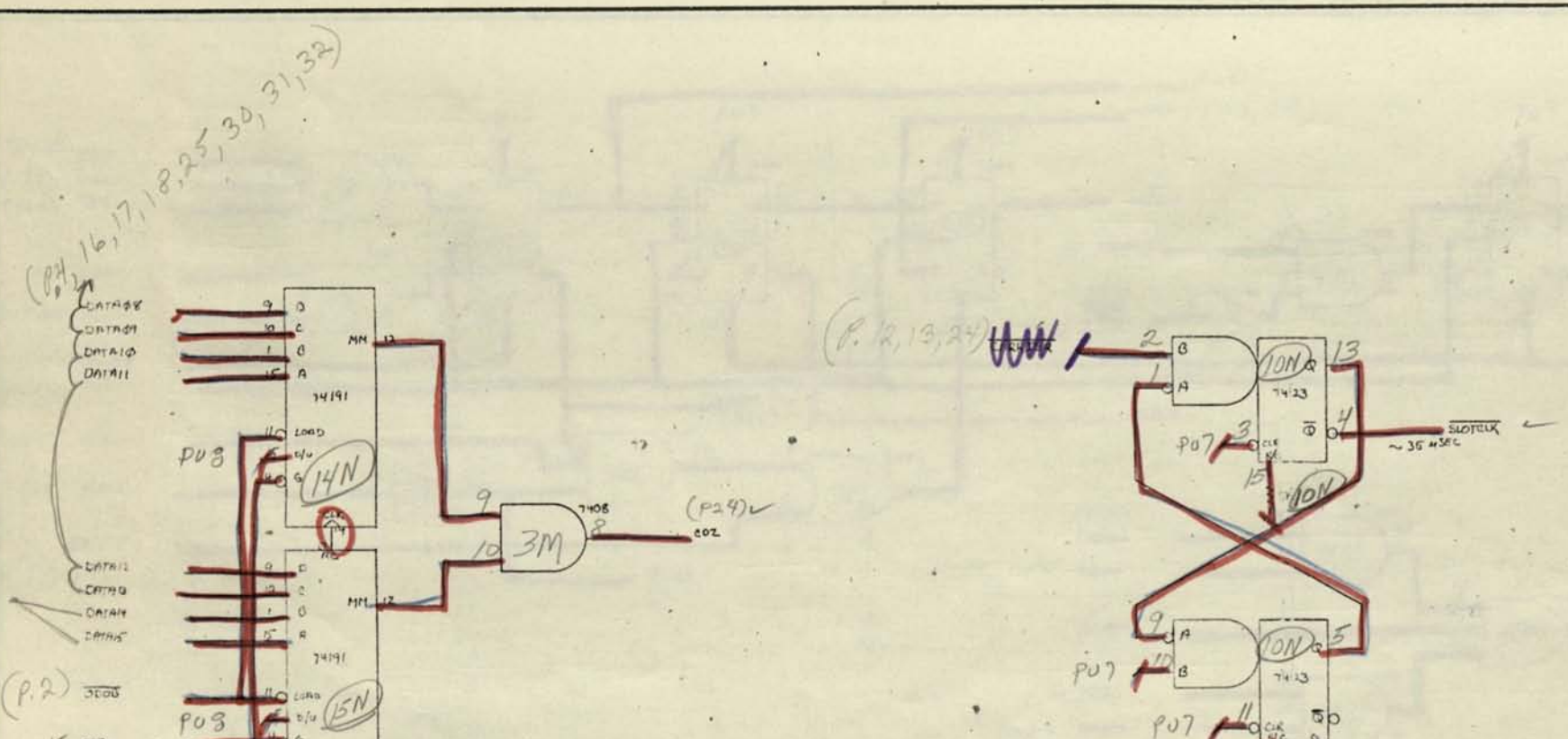


ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/8/84

OUTPUT BUFFER

DRAWING NUMBER 25 of 32

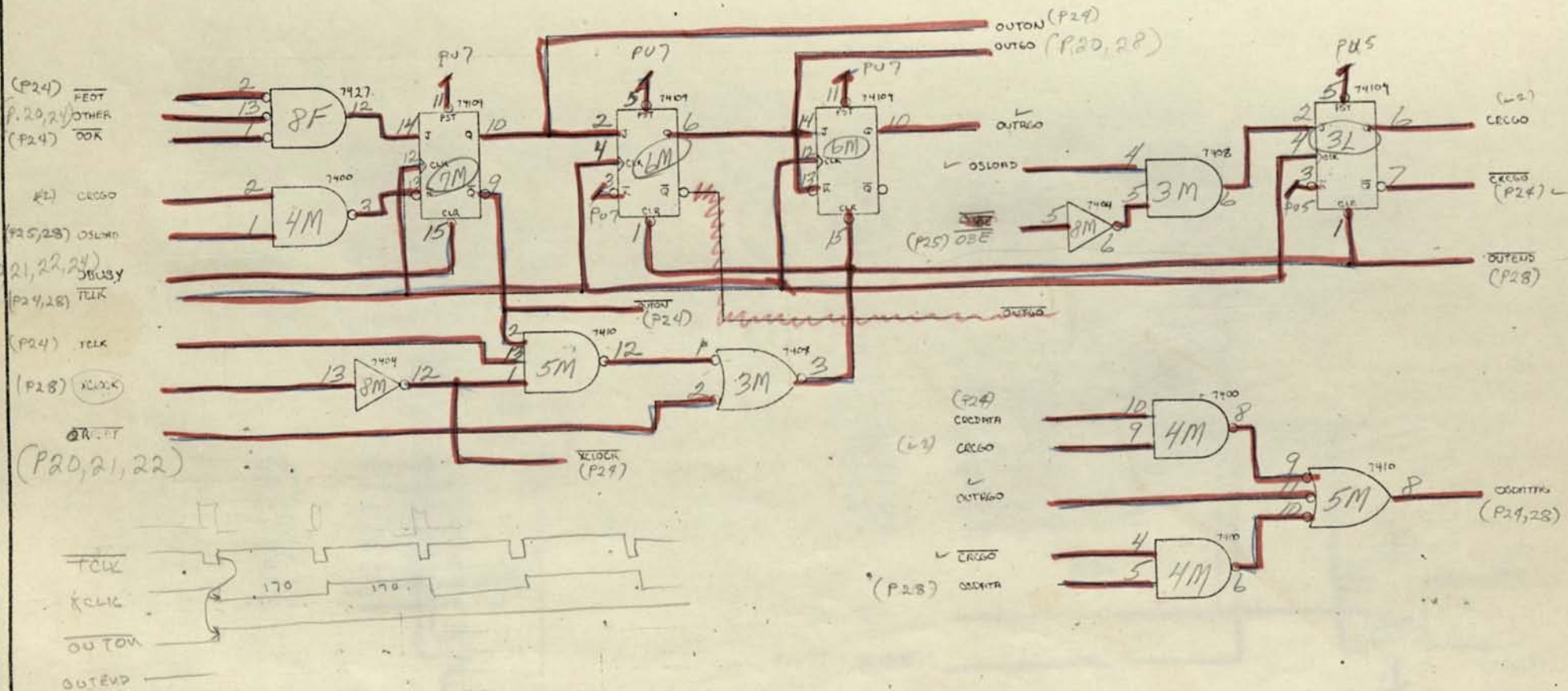


ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/14/74

COUNTDOWN LOGIC

DRAWING NUMBER 26 of 32

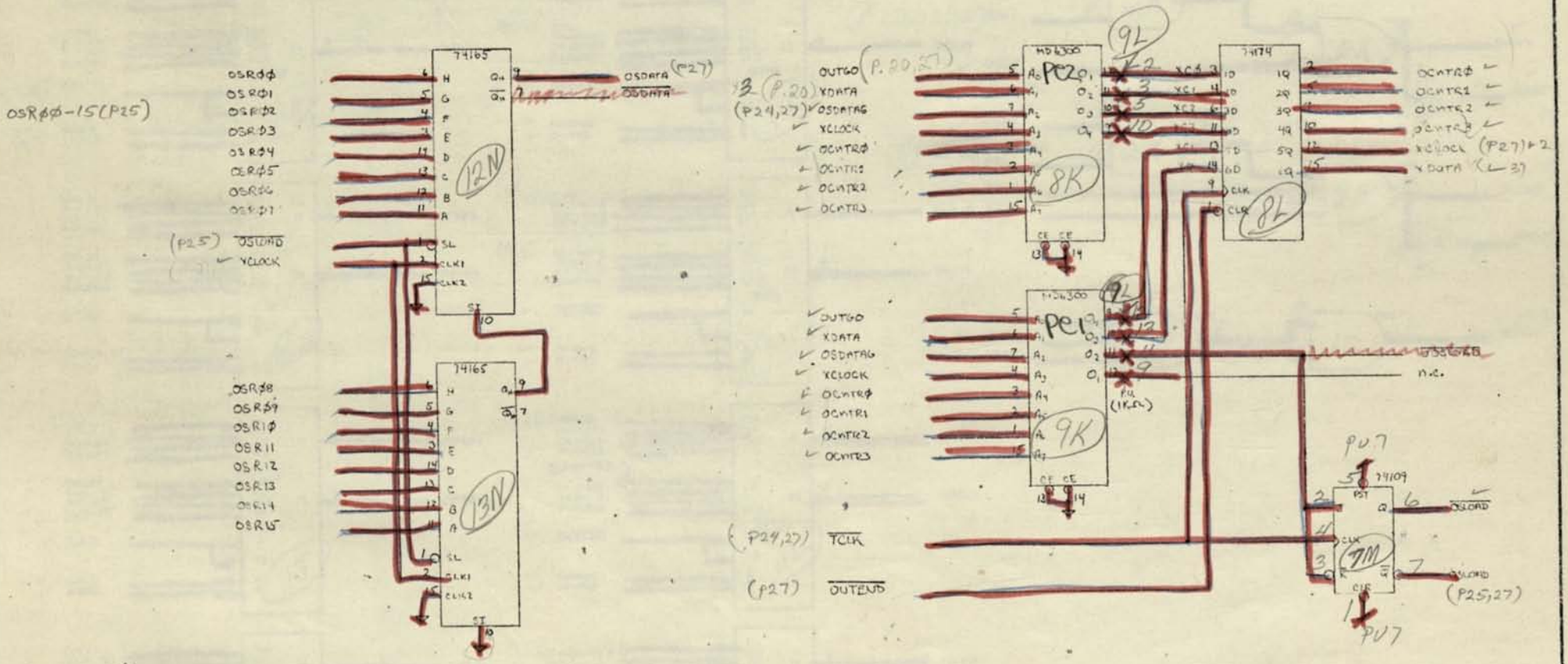


ETHERNET

SCALE:	APPROVED BY:	DRAWN BY cb
DATE: 1/3/74		REVISED 7/14/74

PHASE ENCODER CONTROL

DRAWING NUMBER 27 of 32



ETHERNET

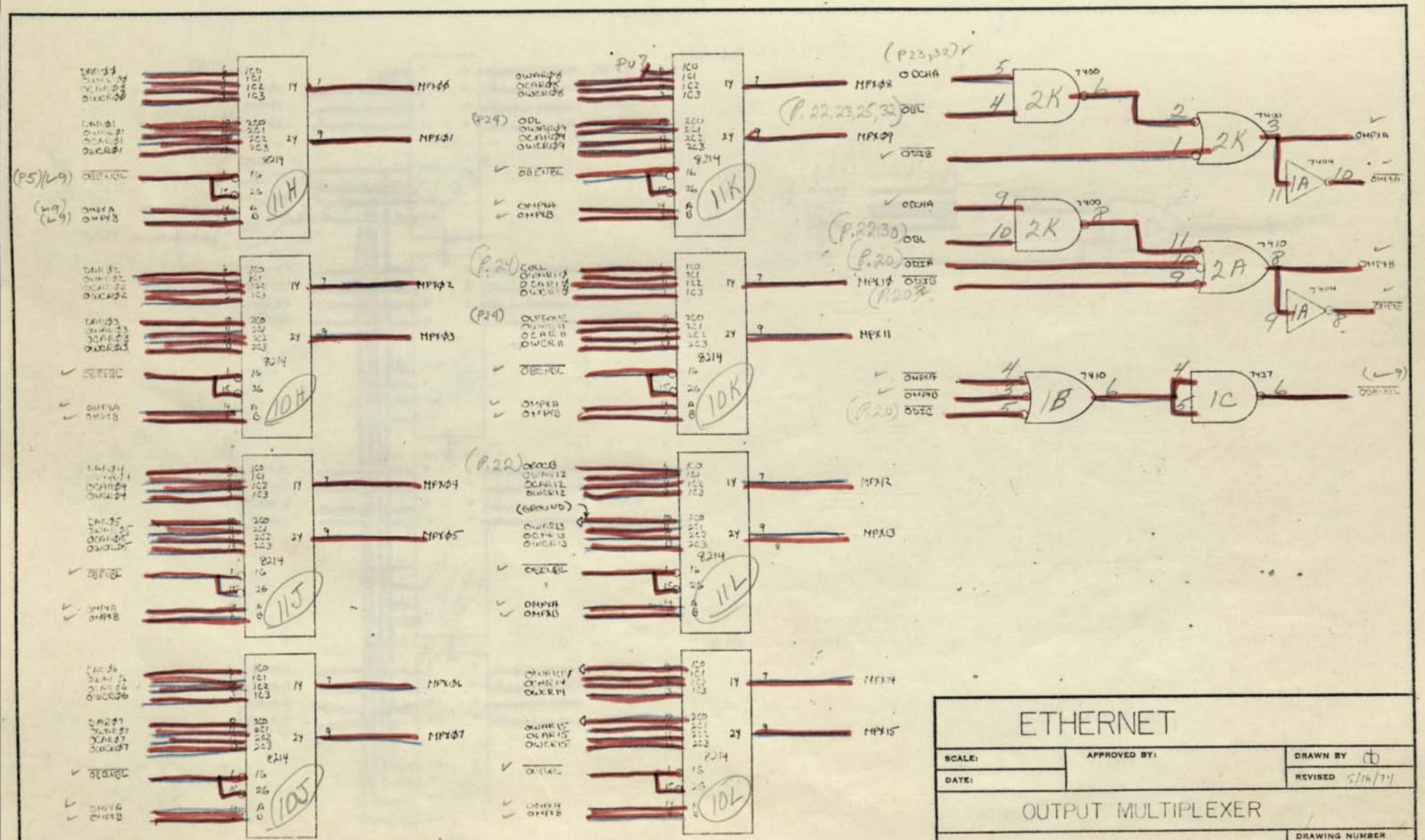
SCALE:	APPROVED BY:	DRAWN BY
DATE: 1/2/74		REVISED 5/18/71

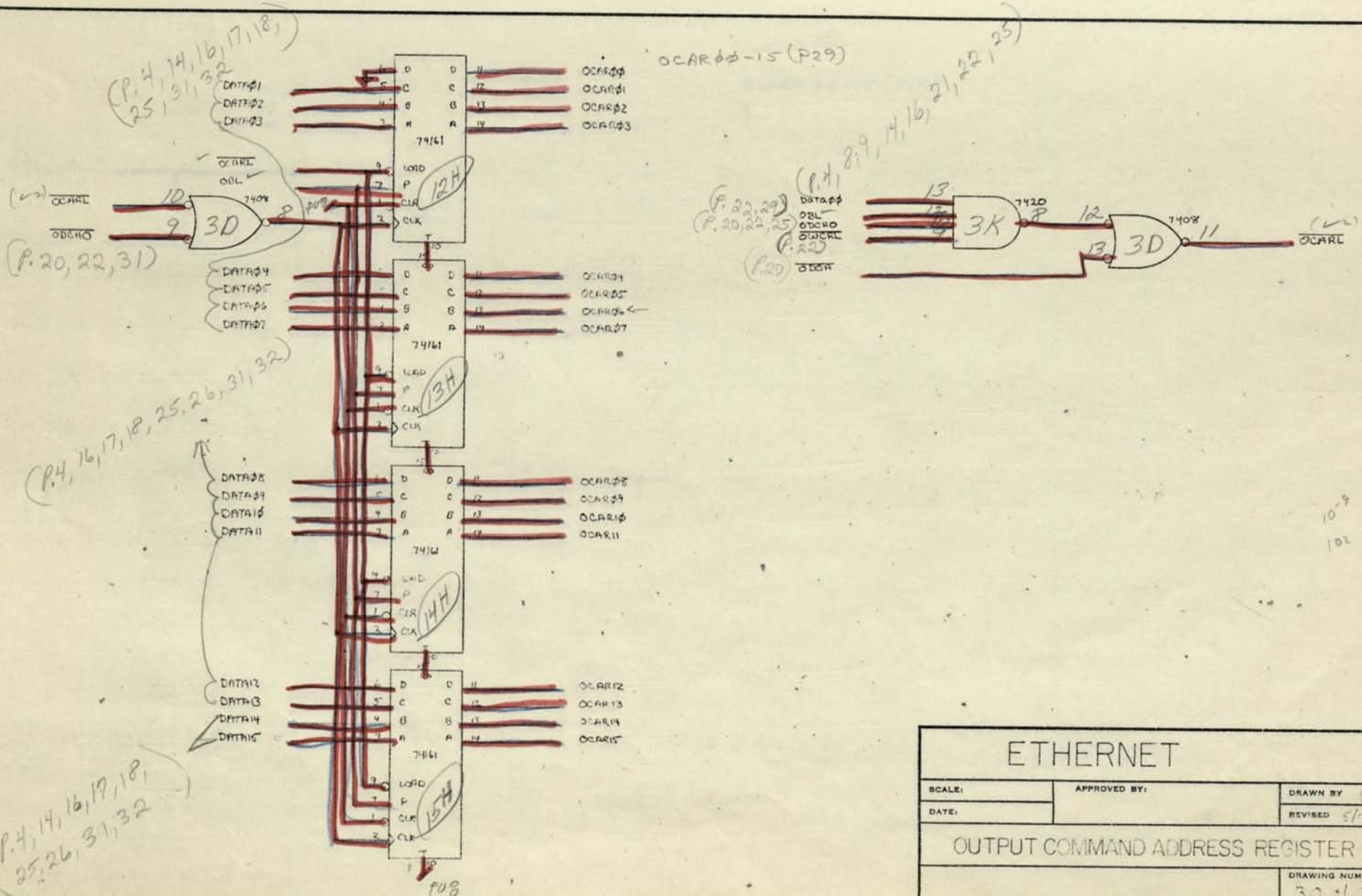
OUTPUT S.R. & PHASE ENCODER

DRAWING NUMBER 24 of 32

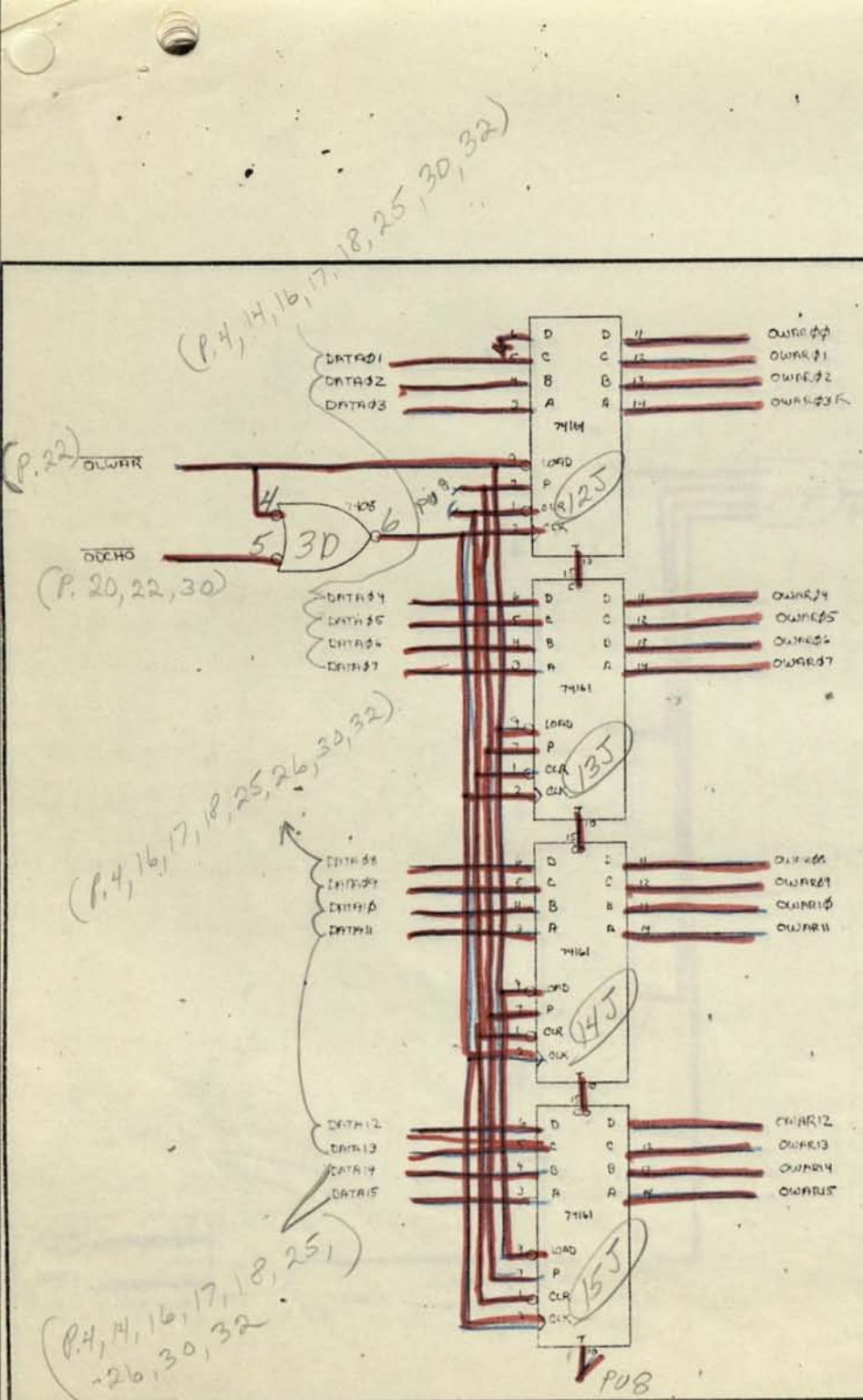
MPX00 – MPX15 (P.5, 15, 19)
DAR00 – DAR07 (P. 14)
OWAR00-15 (P31)
OCAR00-15 (P30)
OWCR00-15 (P32)

A	B	
0	0	STATUS
0	1	CAR
1	0	WAR
1	1	WCR





ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/14/14
OUTPUT COMMAND ADDRESS REGISTER		
		DRAWING NUMBER 32 V 73



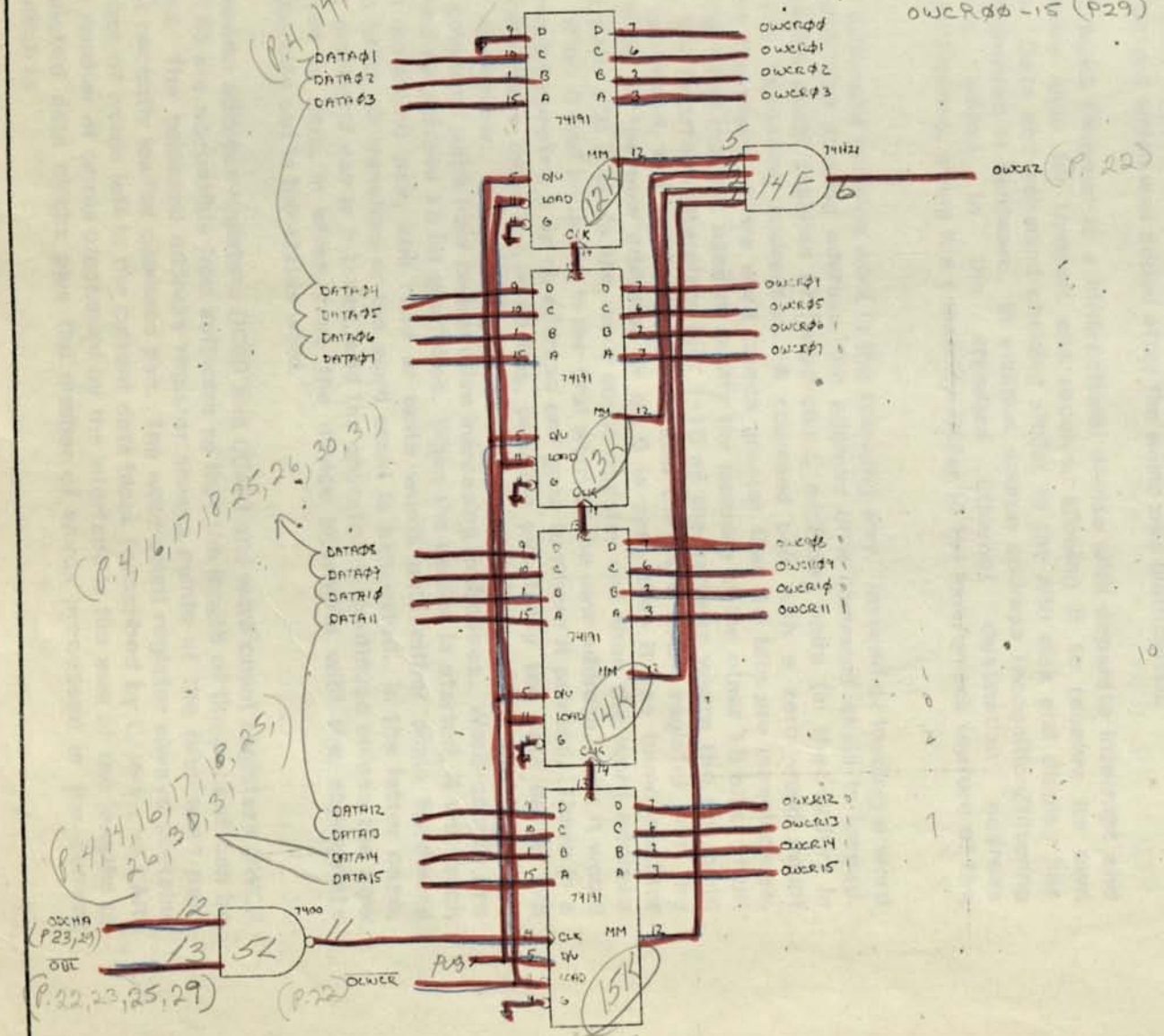
OWAR00-15 (P29)

ETHERNET

SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED 5/14/14

OUTPUT WORD ADDRESS REGISTER

DRAWING NUMBER 31 of 32



ETHERNET		
SCALE:	APPROVED BY:	DRAWN BY
DATE:		REVISED
OUTPUT WORD COUNT REGISTER		
		DRAWING NUMBER 32 of 32

Inter-Office Memorandum

To: Ethernet Watchers

Date: a.d. VI Id. Iun., a.u.c. MMDIIXXVII

From: David Boggs and Bob Metcalfe

Location: Coyote Hill

Subject: Aether-retendum ab NOVIS II

Organization: Parc

XEROX

This is a major revision of the memo by the same name dated April 15 1974. It corrects many typos, omissions, and obscurities, and describes the command chaining feature which was added after the memo was distrubuted.

The MK II NOVA Ethernet is a bi-directional device with separate interrupt and data channel logic for transmit and receive, allowing it to receive its own messages. Data and command chaining similar to the Alto disk and display has been implemented in hardware. In addition, source address recognition/filtering has been added to the standard Ethernet destination address recognition/filtering, giving it a capability similar to the lock/unlock feature of the MCA.

Data and command chaining work in the following way. Instead of loading a word count and buffer starting address, the address of a *command* chain is loaded. The command chain consists of [word-count, address] pairs (in that order) in consecutive increasing addresses. A command pair with a zero word count terminates the chain. Since word counts greater than 15 bits are meaningless, bit 0 of the word count is used to modify the meaning of the other 15 bits. If bit 0 is one, the interface interprets bits 1-15 as the address where the command chain is continued, and again attemps to load the word count register with the word it finds at the new address. If bit 0 is zero, then it has found another command pair, and it loads the word count register and then the word address register. If bit 0 had been one in the first word at the new address, then it would have again interpreted it as an address and gone to where it pointed. There is a limit on how many times it can do this, because eventually the data buffer will over- or underflow. In the absense of the command chain bit, the interface fetches command pairs from consecutive increasing addresses. Word-counts are interpreted as *positive* 15 bit quantities. When the device is started, it will fetch the first command pair, and read or write words until either some hardware condition halts all transfers or the word count is exhausted. In the latter case, the next command pair is fetched, and the operation is continued unless the new word count is zero, in which case the device interrupts with the appropriate completion bits set in the status word.

The command address registers (ICAR and OCAR) and word count registers (IWCR and OWCR) are accessible from software so that the length of the packet can be computed. The command address register always points at the first word past the most recently loaded command pair. The word count register always contains the number of words left in the current data block (described by CAR-1 and CAR-2). The number of words processed by the interface is the sum of the lengths of all completed data blocks plus the number of words precessed in the current block, which is

The number thus calculated can be very misleading if the interface is still running, or halted due to an abnormal condition. Prophesy and entrail reading are not recommended.

Address recognition and message filtering is accomplished in the receiver by specifying an 8 bit address or the condition *promiscuity* for each of the two address fields of the first word of an Ethernet packet. Specifying promiscuity for either of the fields makes that half of the address a "wild card" - anything in that field of an incoming packet matches. Packets with a zero destination byte are called *broadcast* packets, and always match whatever destination conditions were specified by the receiver. The destination address is fixed by an 8 bit switch register in hardware; the source address is an 8 bit register loadable from software. If the receiver is listening, a message will be accepted if the following conditions are satisfied:

(brdcst % destprom % destmatch) & (srcprom % srcmatch)

Any packet heard by the receiver which does not meet these conditions will be ignored. *START* forces the receiver to be destination specific and source promiscuous. *IOPLS* starts the receiver with the most recent address specifications (ala MCA).

To make it possible to boot over the Ethernet, if a *START* is issued **after** an *IORST* and **before** a *DOA* to the receiver, scatter-write is disabled in the receiver, and instead it will accept an infinite (65 K word) message and store it starting at location 0. This is the sequence of events that would be performed by putting the receiver device address in the switch register and mashing the Nova's *program load* button. It is hoped that this combination is sufficiently different from normal use that it will be hard to accidentally do this. The transmitter does not have this capability (or latent trap as the case may be).

TRANSMITTER - Device Code 74, Mask bit 0

START Count down to zero and attempt to transmit
CLEAR Resets device
IOPLS unused
IORST Resets device

DIA Read *O*utput *C*ommand *A*ddress *R*egister (*OCAR*)
DIB Read *O*utput *W*ord *C*ount *R*egister (*OWCR*)
DIC Read Transmitter Status

structure **TRANSMITTER STATUS**:

```
[
DAR byte      //Destination Address Register
GD bit 1      //Gravity Detect. 1 if you have an Ethernet
ODL bit 1     //Output Data Late
COLL bit 1    //COLLision - transmit attempt failed
OG bit 1      //OutGone - transmitter stopped
EOCB bit 1    //End Of Command Block
unused bit 3  //unused
]
```

DOA Load *OCAR*
DOB Load Count Down Register (bits 0-7 ignored)
DOC unused

Gravity Detect will be one as long as 1) you posess an Ethernet interface with power applied and 2) the Law of gravity holds. Absense of Gravity Detect is cause for grave concern. ODL means the output buffer went dry before the end of the command chain was reached because the device could not get enough data channel cycles. This is the symtom that will occur if command chains get too baroque. The COLLision bit indicates that a collision occurred on the last transmit attempt. A new random countdown must be loaded, OCAR must be reinitialized, and the transmitter restarted. See the authors for details on random countdown generation. The algorithm is expected to evolve with time. OG and EOCB will both be one upon successful completion of a transmission. If the tranmsission was aborted, EOCB may be zero. If OG is ever 0 at the end of a successful or aborted transmission, call an Ethernet repairman immediately.

RECEIVER - Device Code 73, Mask bit 0

START Clear *DPROM*, set *SPROM*, *START* receiver
CLEAR Reset device. do not modify state of address recognition logic
IOPLS *START* receiver. do not modify state of address recognition logic
IORST Arm the *BOOT* mechanism, reset device

DIA Read *I*nput *C*ommand *A*ddress *R*egister (*ICAR*)
DIB Read *I*nput *W*ord *C*ount *R*egister (*IWCR*)
DIC Read Receiver Status

structure **RECEIVER STATUS:**

```
[
SAR byte        //Source Address Register
GD bit 1        //Gravity Detect
IDL bit 1       ///Input Data Late
CRCZ' bit 1      //1 means checksum error
IG bit 1        ///nGone - packet ended.
EOCB bit 1       //End Of Command Block
IT bit 1        ///Illegal Termination
DPROM bit 1      //Destination PROMiscuous
SPROM bit 1      //Source PROMiscuous
]
```

DOA LOAD *ICAR*
DOB unused
DOC Set Address Recognition/Filtering Logic

structure **ADDRESS LOGIC:**

```
[
SAR byte        //Source Address Register
unused bit 6    //unused
DPROM bit 1      //make Destination PROMiscuous
SPROM bit 1      //make Source PROMiscuous
]
```

Input Data Late indicates inability to get enough data-channel cycles. The receiver buffer overflowed, so the packet should be ignored. An occasional data late is not cause for concern on a machine with many data-channel devices. If CRCZ' is one, a data checksum error has occurred, and the packet should be ignored. This checksum is computed by the transmitter during the serializing operation and appended to the end of the transmission. The receiver recomputes it during the deserializing process, and checks for equality. Since this only assures correctness through the ether and the serial parts of the interface, users are advised to include a software checksum on all packets they value.

The checksum generated by the hardware is entirely invisible at the software level except for the CRCZ' bit in the receiver status word. IG indicates completion of a packet. If EOCB is one, the end of the command chain was encountered, and IG will generally be zero indicating that there was still data coming in. A message exactly as long as the sum of the lengths of all the data blocks in a command chain will cause both of these bits to be set, but in most cases EOCB is an indication of buffer overflow. Since packets on the Ethernet are composed of 16 bit words, the hardware checks all packets to make sure they end on word boundries. If the IT bit is set, the packet did not end on a word boundry and should be ignored.

ECCE !

Inter-Office Memorandum

To Nova Ethernet Debuggers Date 27 April 75

From David Boggs Location Palo Alto

Subject NEDP Program Organization Parc

XEROX

NEDP, the Nova Ethernet Debugging Program does just what its name says: helps debug Nova Ethernets. It is written mostly in BCPL with a smathering of assembly language. The user interface is patterned after EDP, the Alto Ethernet Debugging program.

This program was written assuming a fast output device; some of the more involved commands produce long-winded replys. You will be sorry if all you have is a teletype. For the sake of brevity, familiarity with the operation of the Nova Ethernet Interface is assumed.

The top level prompt is ">". Commands are triggered by the first letter of the command name. Keys that do not correspond to legal commands are ignored. Typing <space>, <esc>, or <del> aborts the command and exits to the top level of the command parser. Typing "?" at any point will cause a list of the legal commands for that context to be printed. Some tests have no logical end - they continue running until a key is struck, at which point the test is stopped and the command parser is invoked.

Several entities repeatedly referenced in the following command description need definition. The program maintains queues of 'buffers'. A buffer consists of two parts: a header containing control information, and a body into which incoming data is written or from which outgoing data is read. Commands are provided for setting and reading all of the useful items in the header. The 'last input buffer' (LIB) and 'last output buffer' (LOB) are the buffers most recently processed by a test. 'Templates' define the contents of parts of a buffer. There are currently two types of template: Command chain templates and packet templates. A command chain template is a 'relocatable' command chain - addresses are expressed as small offsets. A packet template specifies the body of an output packet. When the program is initialized, the templates are set to default values which experience has shown to be most generally usefull. 'Switches' are boolean statics controlling aspects of the global behavior of the program such as suppressing error messages so that test loops run fast enough to sync a scope on.

The following paragraphs describe the operation of the commands in detail. Following that is a skeleton list of the command structure.

R - STATIC REGISTER TESTS. The static register tests load random numbers into the program accessable registers, read them back and check for equality. This will find bad bits in the registers, bus multiplexors and bus drivers. "A" or "F" tests the Address Filtering registers - the Source Address Register (SAR), the Source PROMiscuous bit (SPROM), and the Destination PROMiscuous bit (DPROM). "I" tests the Input Command Address Register (ICAR). "O" tests the Output Command Address Register (OCAR). "C" tests all of the registers. If an error is detected and the SCOPE switch

is false, the value written into the register, the value read back, and their XOR are printed. At this point you can loop reading and writing the failing value, continue the test with more random numbers, or stop and return to the command parser.

B - Find the BUFFERS. This is useful only for debugging NEDP. It searches the queues checking for buffers that may have gotten lost.

D - DEBUGGER. This too is mostly useful for debugging NEDP. It causes the program to enter DDT. First a check is made to see if a breakpoint is set, and if so it executes it. If no breakpoint is set, a 'GOTO' must be executed to enter the debugger and it is not possible to resume execution, but only to restart the program. In this case it informs you where to set a breakpoint so that future calls on the debugger are coroutine calls. In addition it tells you where the program starts so that, the breakpoint set, the program can be restarted without returning to the DOS exec. If DDT was not loaded, it tells you and punts, returning to the command parser. Normally the breakpoint is set and a 'break file' is made, so that the process of setting the debugger entry breakpoint and restarting the program is only done once before the program is released for general use.

T - TOGGLE a switch. This flips one of the global switches. If the switch was true, it becomes false and vice-versa. Currently there are three switches: Scope, Verbose, and Random. If the Scope switch is true, all printout during execution of a test is suppressed and errors which would normally stop the test do not. This is useful once you know an error is occuring and you want to look at it with a scope. If the Verbose switch is true, a number of parameters which are normally defaulted at the beginning of the Input, Output and Loopback tests are explicitly prompted for. Verbose = false is equivalent to responding with <cr> to all of the questions asked if Verbose = true. If Random is true, packets of random length full of random numbers are used in the Output, Loopback and Echo User tests. The packets are filled with the packet template, and then if there is room, padded with random numbers.

S - SET part of the buffer header. Four things can be set: Templates, Partner, Initload, and Filtering specs. If a template is to be set, you must specify whether it is for command chains or output packets. A command chain is part of the header of a buffer. There is space for up to ten 'blocks'. A block is either a 'link block' or a 'data block'. A link block contains the number of the block where the command chain is continued (must be less than 10). No check is made on whether the target block is already in use. A data block contains an offset into the body of the buffer and the number of words to read or write starting at that offset. A data block with a zero word count terminates the chain. Note that is is possible to set up command chain loops and other ridiculous situations which the hardware will happily execute, hanging the data channel and killing the machine. Before using this command, print the template for command chains to get an idea of what one looks like. The output template is a list of words which is used to fill output packets. It is possible to set the length of the template, and change an arbitrary number of words beginning at an arbitrary offset in the template. The 'Partner' is the address to which output packets will be directed during the Output and Echo User tests. A partner of 0 (broadcast) is not allowed. 'Initload' sets the initial value of the load - the mask used to generate random retransmission intervals. (16-Initload) is the number of consecutive collisions before a load overflow happens. The 'Filtering specs' are the contents of SAR, SPROM and DPROM.

Q - QUIT. Cleans up, puts DOS back together and exits to the DOS exec. <ctlA> will exit in a clean way too.

! - INITIALIZE. Cleans up the queues and initializes all user setable values to the defaults in force when the program was started.

N - Reads the serial NUMBER of the interface (DAR) and prints it out.

P - PRINT. Prints many things: Packets, Initload, Filtering specs, Command Chains, Statistics, and Templates. Print Packets prints LIB and LOB and their XOR. If the last test to run was Output, then there will not be a last input buffer, so no input packet is printed. Printout can be stopped by hitting the space bar at any time. You must 'lead' it a little though, since the output is buffered, and the buffer must drain before printout ceases. If you miss you can ask it to print it again. Print packets will try its best to get something printed, even if the command chains are defective. Print Initload and Print Filtering specs do the obvious things. Print Command chains asks you to specify the input or output chain, and then prints the 'real' chain for LIB or LOB. The 'real' chain has bound addresses as opposed to a template which has relocatable addresses. Print Statistics prints a summary of all the status words returned during the previous test and should be the first thing looked at when a test fails. Print Template asks you to specify Command chain or packet template and then prints the requested one.

The following tests are the real 'core' of NEDP. They all keep a running count of the number of times an attempt to perform the test was made, the number of times the attempt succeeded, and the number of times it failed. For each 64 successes, a "!" is printed. For each failure a cryptic abbreviation is printed, and for any failure following a success, a "~" character is printed. A list of the cryptic abbreviations and what they mean follows the command list. All of these tests run until a key is hit or some unrecoverable disaster occurs. When one of the tests is stopped, the number of attempts, successes, and failures is printed.

I - INPUT test. Starts up the receiver with the current filtering specs and sucks in packets.

O - OUTPUT test. Starts sending packets to the partner. If no partner has been specified, it prompts for one. If the random switch is true, the packets are of random length and filled out with random numbers past the end of the template.

L - LOOPBACK test. Starts up the transmitter sending to the receiver. The receiver is made source specific so that it will not be bothered by other machines sending packets to it. The incoming packets are checked for match against the outgoing packets. If the comparison fails and Scope is false, the test is stopped so that the packets can be printed and the failing bits located.

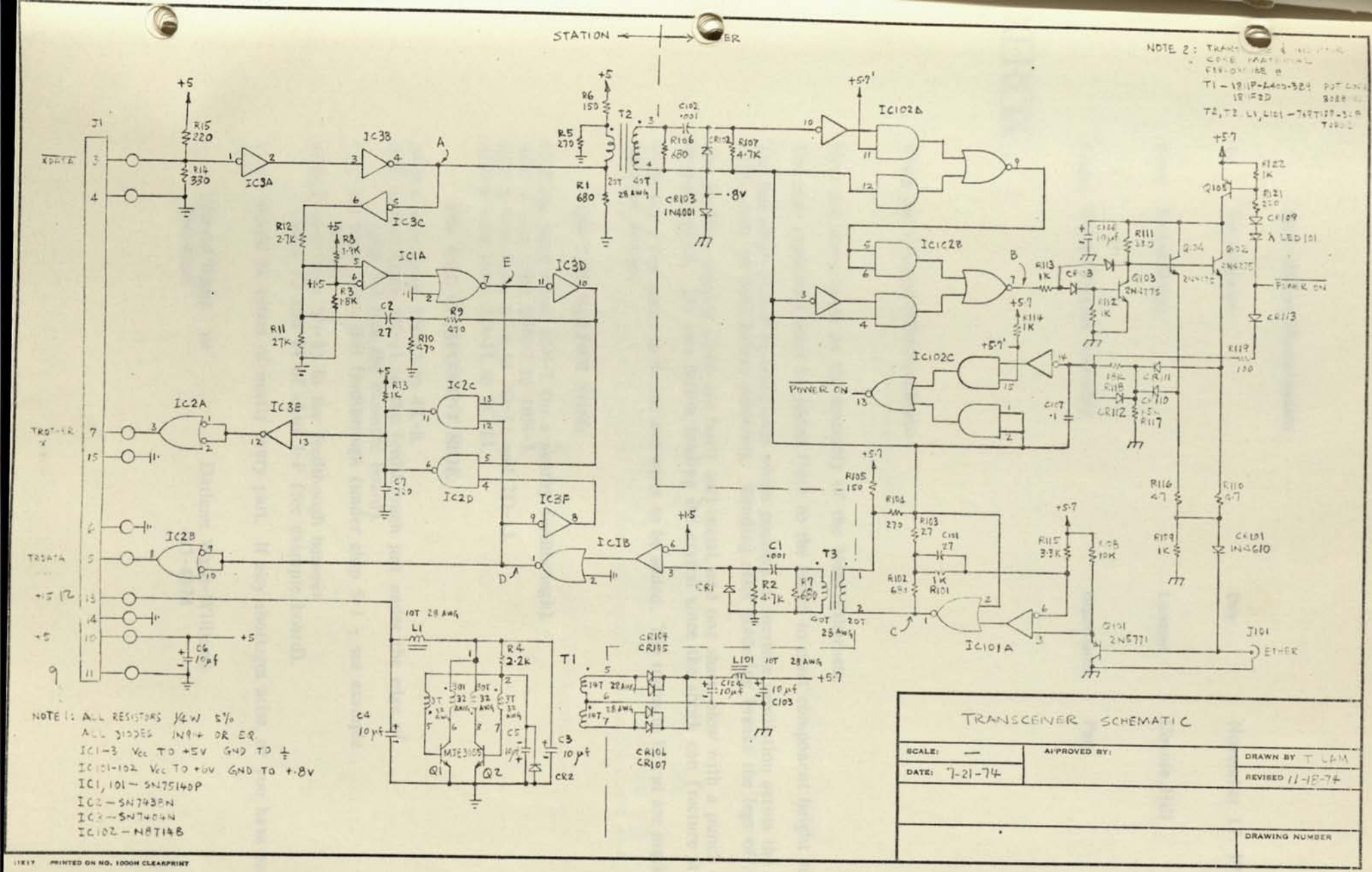
E - ECHO tests. There are two echo tests: Echo User and Echo Server. Echo User is the same routine as the loopback test except that a partner other than 'self' is used, and the check on input timeouts is relaxed. If no partner has been set, the program prompts you for one. Echo Server is the other end: it turns on the receiver and sends any packet of type ECHOME back to its source having changed the type to IMANECHO. If the packet was received damaged, it sends back a two word packet of type IMABADECHO.

? - Prints the legal commands in the context in which the ? was typed.

```
R - Static Register Tests
    A|F - SAR,SPROM,DPROM
    I - ICAR
    O - OCAR
    C - all of the above Cyclicly
B - Find the Buffers
D - Debugger
T - Toggle the
    S - Scope switch
    V - Verbose switch
    R - Random switch
S - Set
    T - Template for
        C - Command Chains
        P - output Packets
    P - Partner
    I - Initload
    F - Filtering specs
Q - Quit
! - Initialize NEDP
N - serial Number is
P - Print
    P - Packets
    I - Initload
    F - Filering specs
    C - Command Chains
        I - Input
        O - Output
    S - Statistics
    T - Template for
        C - Command Chains
        P - Packets
I - Input Test
O - Output Test
L - Loopback Test
E - Echo Test
    U - user
    S - Server
? - Command list


-- Cryptic abbreviations

Ib - Input bad; bad receiver status
Ob - Output bad; bad transmitter status
B - Broadcast message was received
C - Packet experienced one or more collisions going out
Ot - Output timeout; Transmitter should have sent packet long ago
It - Input timeout; Receiver should have received a packet long ago
S - Source bad; got a packet from some one I didnt expect
T - Type bad; got a message of wrong type - such as IMABADECHO
L - Length bad; message is not length I expected it to be
D - Data compare error; would stop except scope is true
```



Inter-Office Memorandum

To Versatronics

From David Boggs

Subject Nova Ethernet Assembly

Date November 17, 1976

Location Coyote Hill

Organization Parc

XEROX

Filed on: <Boggs>NEthAsm.bravo

Here are some notes on the assembly of the Nova Ethernets.

The grey capacitors must be seated flush to the board to meet component height requirements.

The bus strips should be layed over where possible to increase ventilation across the board. The strips must be bent before soldering. Bending after soldering breaks the legs off.

The Southco injector rivets are fairly soft metal, and one sharp blow with a punch on an anvil should suffice. Do this before loading the crystal since the shock can fracture it.

There are a few traces to be cut and wires to be added. The traces to be cut are marked with red adhesive arrows.

ON THE SOLDER SIDE:

CUT the trace from 10N-2 (to a nearby feedthrough).
ADD a wire from 10N-2 to 10N-3.
ADD a wire from 6J-6 to 7D-12 and 7D-13.
ADD a wire from 7D-11 to 7C-11.

ON THE COMPONENT SIDE:

ADD a wire from 5N-13 to 4K-6.
ADD a wire from 5N-11 to the feedthrough just under the edge of chip 5-J (see the example board).
CUT the trace from this feedthrough (under chip 5-J - see example board)
ADD a wire from 5N-12 to the feedthough between pins 13 and 14 of chip 3-F (see example board).

There should be extras of nearly every part. If any shortages arise, or you have any questions, call:

David Boggs or
494-4365

Darlene Ward-Williams
493-4374

Inter-Office Memorandum

To M & M Designs

Date 14 February 1975

From David Boggs

Location Coyote Hill

Subject Nova Ethernet Change

Organization Parc

XEROX

Filed on: <Boggs>2-14-75NEthMods.bravo

There are a few changes to the Nova Ethernet as well. Some of them correspond to the changes you made to the Alto Ethernet.

1) Remove the signal Carrier' from the B input of 1-shot 10-N (pin 2). Tie the input to a pull up net (such as PU7).

2) Add another 7438 with its output tied to XDATA' (in parallel with 7C-11). Connect the inputs together and to COLL. There are two free 7438s at position 7D.

3) Replace the piece of tape from 3D-14 to the feed-through located about 1/2 inch above it. It must have peeled off while the film was being made. It is on the check print, but not on the film.

The pages containing the affected signals are attached. The signals and changes are shown in red.

Inter-Office Memorandum

To File

Date May 28, 1977

From David Boggs

Location Coyote Hill

Subject Nova Ethernet Device Codes

Organization Parc

XEROX

Filed on: <Boggs>NEthDevCodes.bravo

To use more than one Nova Ethernet interface in the same machine, the device codes of all but one of the boards must be changed. This memo describes the installation of switches to accomplish this.

Parts required

2 AMP hexidecimal rotary switches, part number 53137-1.
#30 insulated wire
waxed cable lacing string

Installation

Install the switches in board positions 8N and 9N, with the screw slots facing the board stiffener. Cut the traces between pin 7 and pin 8 (ground) on both board positions. Drill two holes in the stiffener so that a small screw driver can be inserted to turn the switches. Tie the wires together with lacing string so that they lie flat on the board among the components.

Wiring

Note that the switches are installed upside down in the board. All numbers will refer to pin positions on the board, not the switches.

Install the following jumpers:

9N-12	to	9N-9	|	
9N-11	to	9N-10	|	use inside row of holes
9N-3	to	9N-8	|	
9N-3	to	9N-11	|	
9N-4	to	9N-12	|	use outside row of holes

Cut the following traces:

trace leaving 7A-5 before the feed through (solder side)
trace entering 7A-12 (component side)

trace from 7B-1 to 6D-6 between the feed throughs (component side)

Install the following wires:

7A-5	to	9N-14		
7A-12	to	9N-6		
7B-1	to	9N-7	|	use the feed through holes
6D-6	to	9N-9	|	mentioned above

This completes the modification of the logic which responds to INTA instructions.

Add add the following wires:

4D-3	to	8N-14	(msb center pole)
5C-1	to	8N-3	
5C-2	to	8N-4	
4D-11	to	8N-6	
5A-9	to	8N-11	
5A-8	to	8N-12	
4D-4	to	8N-7	(lsb center pole)
5A-5	to	8N-10	
5A-6	to	8N-9	

Cut the following traces:

trace leaving 5A-6
trace leaving 5A-8
trace leaving 5C-2

This completes the modification of the device select logic

ETHERNET TRANSCEIVER ELECTRICAL CHARACTERISTICS

1.0 Station

1.1 Input:- $\overline{\text{XDATA}}$
- 1.11 Resistance: 132 ohms nominal
- 1.12 Logic: Negative ("0" volt) true
- 1.13 Interface: 5 volt TTL compatible

1.2 Outputs:- TROTHER & TRDATA
- 1.21 Type of drive: Open collector
- 1.22 Low state current sinking: 48 ma max.
- 1.23 Logic: Positive ($\geq$ 2 volts) true
- 1.24 Interface: 5 volt TTL compatible

2.0 Ether

2.1 Input:- ETHER
- 2.11 Resistance: $\geq$ 100k ohms
- 2.12 Capacitance: $\leq$ 10pf
- 2.13 Bias current: $\leq$ -15ua
- 2.14 Max. input voltage without damage: +10 volts, -.4 volts

2.2 Output:- ETHER
- 2.21 Drive voltage: +3 volts (Nominal) into 37.5 ohms
- 2.22 Rise & Fall times: 12ns typical
- 2.23 Waveform: Pulse

2.3 Power shut down at output incase of catastrophic failure at output stage.

3.0 Transceiver

- 3.31 Frequency range: 1.3 MHz to 3.3 MHz
- 3.32 Duty cycle range: 30% to 70% or 140 ns whichever is closer to 50%
- 3.33 Waveform: Pulse
- 3.34 Ether ground potential noise immunity: 50 volts DC reducing to 15v peak to peak at 100k Hz
- 3.35 Power requirement: +15 volts $\pm$ 2% @ 80 ma (typical)
  +5 volts $\pm$ 5% @ 110 ma (typical)
- 3.36 Operating temperature: 10°C to 40°C
- 3.37 Storage temperature: 0°C to 50°C
- 3.38 Size: 2¼" H x 4" W x 2¼" D

ETHERNET STATION PARTS LIST

Ckt No.	Mfr*	Part No.	Description

Resistors are fixed, composition/film, ±5%, 1/4 watt unless otherwise indicated.

R1			680 ohm
R2			4.7k
R3			1.8k
R4			2.2k
R5			270
R6			150
R7			680
R8			3.9k
R9			470
R10			470
R11			27k
R12			2.7k
R13			1k
R14			330
R15			220

Capacitors

C1	El	DM-19-102J	Mica 1000pf 5% 500v
C2	El	DM-15-270J	Mica 27pf 5% 500v
C3	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C4	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C5	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C6	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C7	El	DM-15-221J	Mica 220pf 5% 500v

Inductor

L1			150uH

Transformers

T1			Power transformer
T2			Coupling transformer
T3			Coupling transformer

Semiconductor devices

D1	TI	1N914	Diode
D2	TI	1N914	Diode
Q1	TI	TIP3055	NPN power transistor
Q2	TI	TIP3055	NPN power transistor
IC1	TI	SN75140P	Dual line receivers
IC2	TI	SN7438N	Quad 2-Input + NAND Buffers
IC3	TI	SN7404N	Hex Inverters

ETHERNET ETHER PARTS LIST

Ckt No.	Mfr	Part No.	Description
Resistors are fixed, composition/film, ±5%, 1/4 watt unless otherwise indicated.			
R101			1k ohm
R102			680
R103			27
R104			270
R105			150
R106			680
R107			4.7k
R108			10k
R109			1k
R110			4.7
R111			330
R112			1k
R113			1k
R114			1k
R115			3.3k
R116			4.7
R117			1.5k
R118			18k
R119			100
R120			220
R121			1k
Capacitors			
C101	El	DM-15-270J	Mica 27pf 5% 500v
C102	El	DM-19-102J	Mica 1000pf 5% 500v
C103	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C104	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C105	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C106	Sp	196D106X9020JA1	Tantalum 10uf 10% 20v
C107	Sp	HY-360	Ceramic 0.1uf 12v
Semiconductor Devices			
CR101	F	1N4610	Diode
CR102	TI	1N914	Diode
CR103	TI	1N4001	Diode
CR104	TI	1N914	Diode
CR105	TI	1N914	Diode

ETHERNET ETHER PARTS LIST

Ckt No.	Mfr*	Part No.	Description
CR106	TI	1N914	Diode
CR107	TI	1N914	Diode
CR108	TI	1N914	Diode
CR109	TI	1N914	Diode
CR110	TI	1N914	Diode
CR111	TI	1N914	Diode
CR112	TI	1N914	Diode
CR113	TI	1N914	Diode
Q101	F	2N5771	PNP transistor
Q102	F	2N4275	NPN transistor
Q103	F	2N4275	NPN transistor
Q104	F	2N4275	NPN transistor
Q105	GE	D43C2	PNP transistor
IC101	TI	SN75140P	Dual line receivers
IC102	Sig	N8T14B	Triple line receivers
LED101	Mon	MV5024	Red light emitting diode

*Note:

El Elmenco
Sp Sprague Electric Co.
TI Texas Instruments, Inc. Semiconductor div.
F Fairchild Camera & Instrument Corp. Semiconductor div.
GE General Electric Co. Semiconductor div.
Sig Signetics Corp.
Mon Monsanto Co. LED div.

To	M & M Designs	Date	March 27, 1977
From	David Boggs	Location	Coyote Hill
Subject	Nova Ethernet Changes	Organization	Parc

XEROX

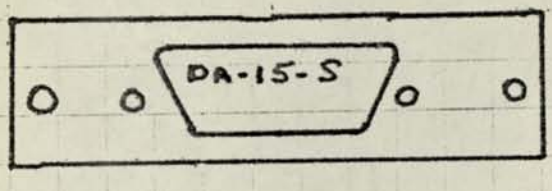
Filed on: <Boggs>3-27-77NEthMods.bravo

Please make the following changes to the Nova Ethernet artwork, make a new set of filmwork, and then send us everything. I am sending my truth-copy of the logic drawings to aid making the changes.

On page 8, Input Interrupt Logic, delete -InGone from 5H-4 and add the gate shown. I used one of the spare positions, but if you can find an unused 7400 nearby, that would be better. Don't kill yourself trying, though.

On page 12, Phase Decoder, change the pin spacing of C1, C2, and C3 to accomodate Nytronics deci-caps. Also add the three 10 pf caps at the R/C junction points on the 1-shots. These should also be spaced for deci-caps.

NOVA ETHERNET



PIN		NOVA PIN	
1		—	
2		—	
3	XDATA	10 A 57	TWISTED PAIR
4	GND	10 A 33	
5	RDATA	10 A 63	TWISTED PAIR
6	GND	10 A 34	
7	OTHER	10 A 61	TWISTED PAIR
8		—	
9		—	
10	+5V	10 A 3	
11	GND	10 A 1	
12		—	
13	+15V	10 A 10	
14	GND	10 A 99	
15	GND	10 A 34	

Specifications

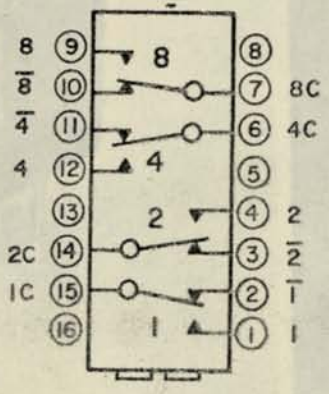
Electrical

Rated Load (@ 28 VDC, Resistive, Switching)	100 MA
Contact Resistance	50 Milliohms (Max.)
Dielectric Strength	300 VAC 60 Hz
Dc Insulation Resistance	1,000 Megohms

Mechanical

Contact Material	Phosphor Bronze, Gold-over-Nickel Plated
Detent Release Torque	1 Oz.-In. (Min.)
Life Expectancy	2,000 Revolutions

Pin Configuration



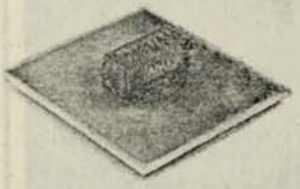
Truth Table (BCD and BCD Complement)

Pole Pos.	8	4	2	1	$\bar{8}$	$\bar{4}$	$\bar{2}$	$\bar{1}$
0					•	•	•	•
1				•	•	•	•	
2			•		•	•		•
3			•	•	•	•		
4		•			•		•	•
5		•		•	•		•	
6		•	•		•			•
7		•	•	•	•			
8	•					•	•	•
9	•			•		•	•	
A	•		•			•		•
B	•		•	•		•		
C	•	•					•	•
D	•	•		•			•	
E	•	•	•					•
F	•	•	•	•				

•=CONNECTION

Mounting

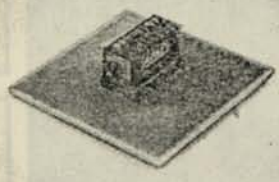
The hexadecimal rotary switch is specifically designed to be mounted into a pc board using the same packaging methods as for 16-lead IC's. For board mounting, AMP has available a wide selection of packaging devices, as shown below. For your convenience, we have also included the number of the catalog containing detailed specifications of each particular product. These catalogs are available, write: AMP Incorporated, Harrisburg, Pennsylvania.



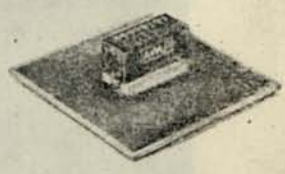
Miniature Spring Sockets (AMP Catalog No. 414-5)

Miniature Spring Socket Receptacles (AMP Catalog No. 414-5)

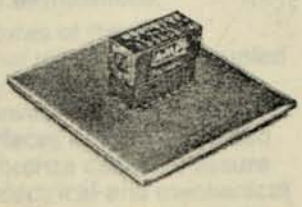
One- and Two-Piece IC Packaging Receptacles (AMP Catalog No. 425-2)



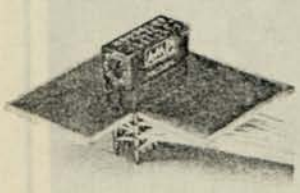
Low Profile DIP Headers w/Solder Posts (AMP Catalog No. 491-9)



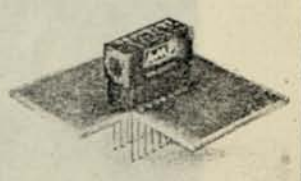
Low Profile DIP Strips w/Solder Posts (AMP Catalog No. 491-9)



Standard Profile DIP Headers with Solder Posts (AMP Catalog No. 491-9)

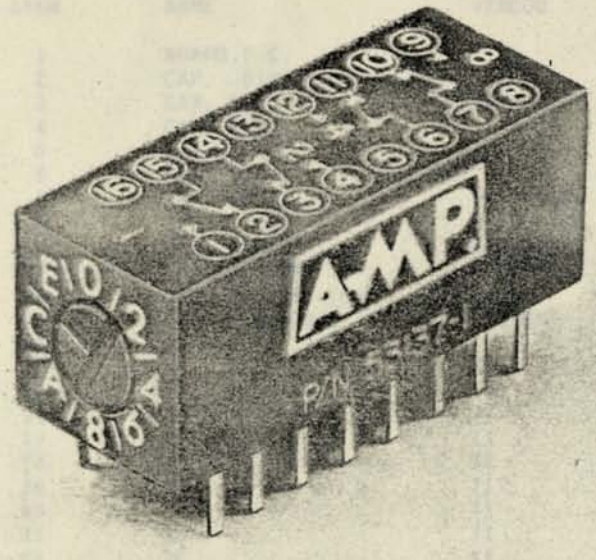


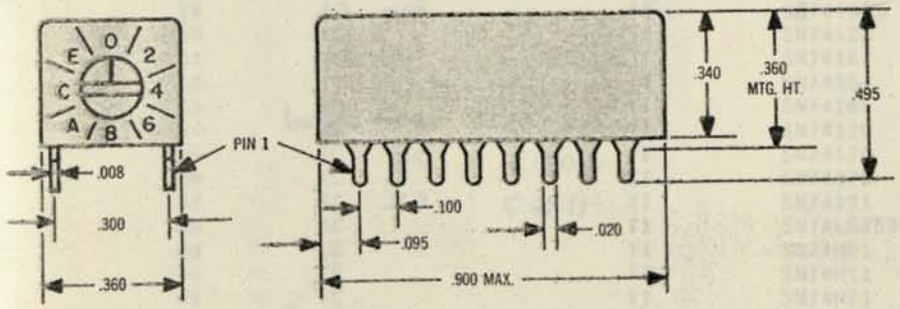
Standard Profile DIP Headers with TERMI-POINT Clip Posts (AMP Catalog No. 491-9)



Standard Profile DIP Headers with Wrap-Type Posts (AMP Catalog No. 491-9)

Hexadecimal Rotary Switch (16-Position, 4PDT)





Features

- 16 Pin DIP
- 16 Step Rotary
- 8-4-2-1 Encoding with Complement
- 4 Pole Double Throw
- Positive Detent
- Screw Slot Actuation
- PC Edge Mount

The AMP Hexadecimal Rotary Switch is a 16 pin DIP package with 4 Form "C" switches which are operated by an 8-4-2-1 binary encoded cam. A screwdriver slot in the end turns the cam in either direction to the desired "positive detent" setting. The setting is indicated by a pointer next to the slot, with associated legend per Truth Table.

The hexadecimal switch is especially designed for printed circuit board mounting and is completely compatible with the latest packaging techniques. Input/output pins are spaced on .100" x .300" centers, allowing it to be mounted in a board by the normal methods used for 16-lead IC's. For board mounting, AMP has available a variety of packaging devices—receptacles, sockets, DIP headers, etc.—which also offer a choice of interconnections using soldered, wrap-type or TERMI-POINT clip terminations.

These features of the AMP hexadecimal rotary switch, coupled with a fully enclosed design that provides environmental protection for the surfaces of the gold plated phosphor bronze contacts, assure excellent electrical and mechanical performance as well as one of the most economical means of manually programming various types of electrical/electronic equipment.

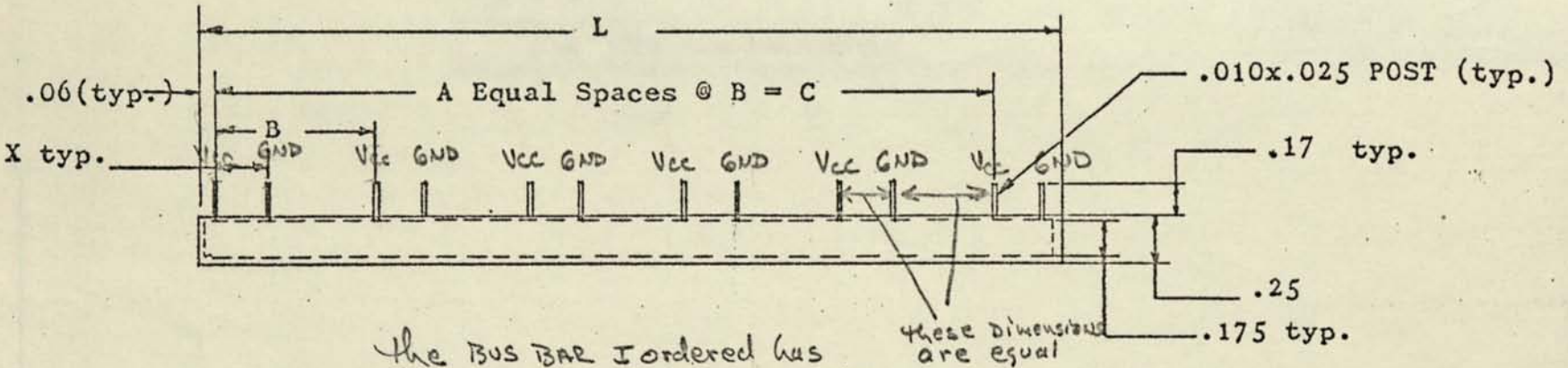
****
PARTS LIST
****

XEROX Palo Alto Research Center

NAME: NOVA ETHERNET CONTROL CARD
FILE: NETH.ASM

ITEM	NAME	VENDOR	PART NUMBER	QUANTITY	COMMENTS
1	BOARD,P.C.			1	
2	CAP, .01MF		CK06BX-103K	59	
3	CAP, 10PF	NYTRONICS	DC-100K	3	DECI-CAP
4	CAP, 47PF		CK05BX-470K	2	
5	CAP, 68PF		CK05BX-680K	1	
6	CAP, 100PF		CK05BX-101K	13	
7	CAP, 150PF	NYTRONICS	DC-151K	1	DECI-CAP
8	CAP, 270PF	NYTRONICS	DC-271K	1	DECI-CAP
9	CAP, 470PF	NYTRONICS	DC-471K	1	DECI-CAP
10	CAP, 1000PF	NYTRONICS	DC-102K	1	DECI-CAP
11	CAP, 6800PF		CK05BX-682K	1	
12	CAP, 6.8UF,25V	SPRAGUE		4	ORANGE DROP
13	IC		I3601/MD6300	4	NFIFO,PE1&2
14	IC	FAIRCHILD	F9401	2	
15	IC	TI	SN7400	10	
16	IC	TI	SN7402	2	
17	IC	TI	SN7404	10	
18	IC	TI	SN7408	12	
19	IC	TI	SN7410	5	
20	IC	TI	SN7414	1	
21	IC	TI	SN7420	2	
22	IC	TI	SN7425	1	
23	IC	TI	SN7427	1	
24	IC	TI	SN7430	2	
25	IC	TI	SN7437	3	
26	IC	TI	SN7438	8	
27	IC	TI	SN7485	4	
28	IC	TI	SN7486	1	
29	IC	TI	SN74109	22	
30	IC	TI	SN74123	3	
31	IC	TI	SN74161	16	
32	IC	TI	SN74164	2	
33	IC	TI	SN74165	2	
34	IC	TI	SN74170	8	
35	IC	TI	SN74174	2	
36	IC	TI	SN74175	1	
37	IC	TI	SN74191	10	
38	IC	TI	SN74LS253	16	
39	IC	TI	SN74H01	1	
40	IC	TI	SN74H11	2	
41	IC	TI	SN74H21	2	
42	RESISTOR		100-OHM,1/4W,5%	1	
43	RESISTOR		120-OHM,1/4W,5%	3	
44	RESISTOR		200-OHM,1/4W,5%	2	
45	RESISTOR		330-OHM,1/4W,5%	5	
46	RESISTOR		390-OHM,1/4W,5%	3	
47	RESISTOR		1K,1/4W,5%	21	
48	RESISTOR		5.1K,1/4W,5%	3	
49	RESISTOR		10K,1/4W,5%	1	
50	RESISTOR		20K,1/4W,5%	2	
51	RESISTOR-NET	BECKMAN	898-1-R1.0K	1	
52	RESISTOR-NET	BECKMAN	899-1-R1.0K	4	
53	DIODE		SILICON.	2	
54	CRYSTAL	SENTRY	SCM-16, 5.88MHZ	1	32 PF SHUNT LOAD
55	16 PIN SOCKET	AUGAT	516-AG10D	1	
56	BUS	BUSSCO	B10200-1000-6-.500	12	
57	BUS	BUSSCO	B10200-1000-14-.500	2	
58	INJECTOR	SOUTHCO	90-0-6503-11	2	
59	RIVET	SOUTHCO	90-0-5858-24	2	
60	STIFFENER			1	
61	SCREW		4-40x1/4	4	

REVISIONS			
SYM	DESCRIPTION	DATE	APPROVED
A	.17 Was .200	1/72	



The BUS BAR I ordered has the X Dimension equal To 1/2B

PART NO	B
B-10200-1	.900
B-10200-2	1.000
B-10200-3	1.100
B-10200-4	1.200

TYPICAL CALL-OUT

B - 10200 -1 -5 -.300

- X Dimension
- A NO. Eq. Spaces
- Part No.

B-10200-1-14-.500 IS what I ordered

NOTES:

1. Conductors- .010 Copper Alloy 110, Half-Hard
2. Insulation- .004 Thk. P.V.F.
3. Finish- Solder Plate, .0002 min.
4. Terminal Nos. - Ref. Only
5. Hypot- 500 VDC

EXCEPT AS NOTED DIM. ARE IN INCHES AND PER MIL-STD-8
.XXX .XX ANGLES
±.010 ±.03 ±0°30'

MATERIAL See Notes

DR N. Lobo 8/17/71

CHK J. Little

APPD R. Perry

LAMINATED VERTICAL CIRCUIT BOARD BUS

ELECTRONIC ENGINEERING INC.

EL SEGUNDO, CALIF.

SIZE	CODE IDENT NO.	NUMBER
A	29593	B-10200

SCALE None | REV - A | SHEET 1-1

LAL	Revisions				209590	A
	Rev.	Description	Chk.	Date	Approved	
	A	ENGRG RELEASE	AC	4-17 75		

ENGINEERING RELEASE
7FC86

6-30-75

Dist. Code 299

These drawings and specifications, and the data contained therein, are the exclusive property of Xerox Corporation and/or Rank Xerox, Ltd. issued in strict confidence and shall not, without the prior written permission of Xerox Corporation or Rank Xerox, Ltd., be reproduced, copied or used for any purpose whatsoever, except the manufacture of articles for Xerox Corporation or Rank Xerox, Ltd.

Notes Unless Specified

1. Tolerances
   .XX ±.030
   .XXX ±.010
   Angular ±1°
2. Break All Sharp Edges .010 Approx.
3. Mach. Surfaces √
4. All Dim. in Inches

Drawn	H. SMITH	3-17-75
Check	[illegible]	4-24-75
Appr.	Silverstone	4-18-75

Material

Xerox Corporation
El Segundo, California

XEROX

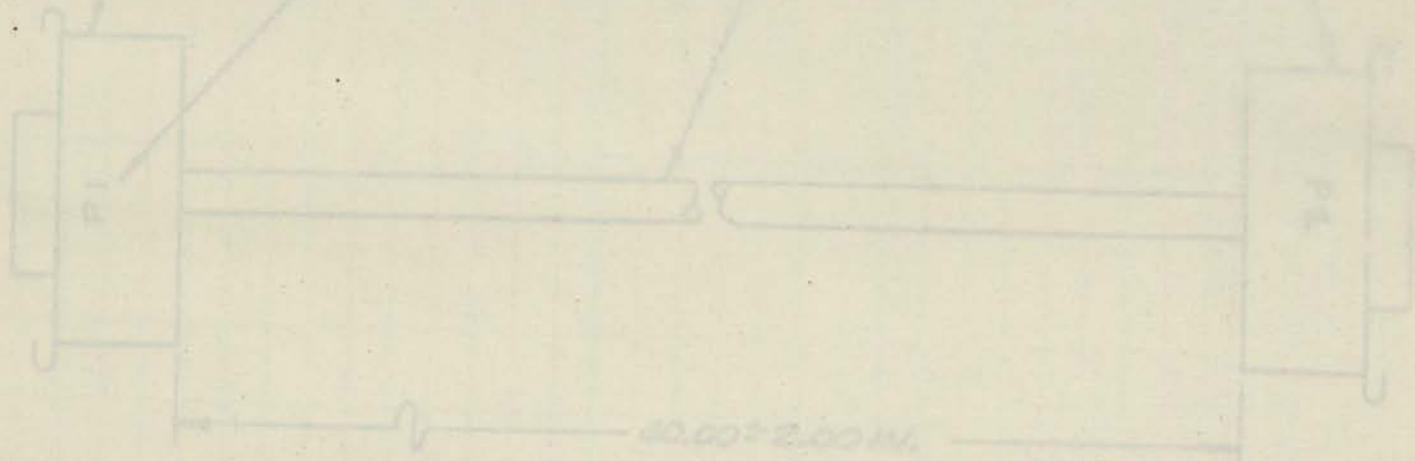
ASSEMBLY, CABLE - ETHERNET (EXTERNAL)

Model No. First Use ALTO	Finish	Code Ident. 18338	Size A	Dwg. No. 209590	Change Letter A
Next Assy. First Use 209584		Scale NONE	Do Not Scale Drawing	Sheet 1 OF 5	

564(3-73)

NOTES: UNLESS OTHERWISE SPECIFIED:

1. FABRICATE PER XEROX SPEC 209454.
2. MARK CHARACTERS .12 HIGH "NEWS GOTHIC".



Title

ASSEMBLY, CABLE - ETHERNET (EXTERNAL)

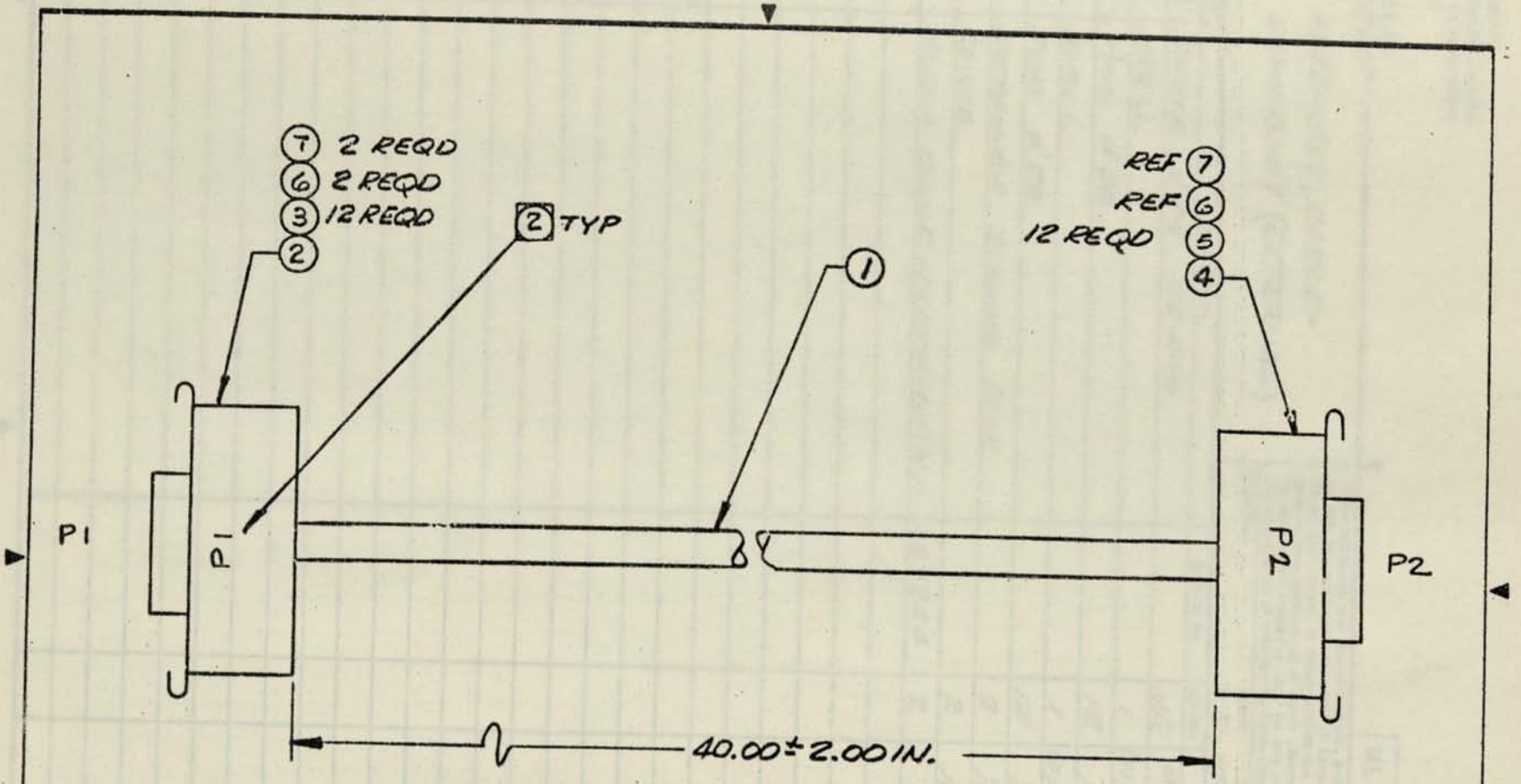
Xerox Corporation
El Segundo, California

XEROX

209590 A

Sheet 2 of 5

112(3/73)



Title

ASSEMBLY, CABLE - ETHERNET (EXTERNAL)

Xerox Corporation
El Segundo, California

XEROX

209590 A

Sheet 3 of 5

112(3/73)

Xerox Corporation
701 South Aviation Boulevard
El Segundo, California 90245
213 679-4511

XEROX

Material List

ML	Drawing No.	Rev.
	209590	A

Drawing No. ML 209590 Rev. A

Drawing Title: ASSEMBLY, CABLE-ETHERNET (EXTERNAL)

Model No.

Date: 3-17-75

Sheet: 4 of 5

Item No.	Drawing Title	Drawing No.	No. Req.	Remarks
1	CABLE, 6 PR, 22 AWG		A/R	BELDON P/N 8747
2	SHELL		1	(P1) CANNON P/N DAC-15P
3	PINS #20		12	CANNON P/N 030-1952-000
4	SHELL		1	(P2) CANNON P/N DAC-15S
5	PINS #20		12	CANNON P/N 030-1953-000
6	RETAINER, SLIDING LOCK		2	CANNON P/N DA-51220-1
7	COVER		2	CANNON P/N DA-51210-1
8	STRAP, CABLE IDENTIFICATION	167244	2	

Wire No.	Term	From	To	Term	Wire Type	Notes	Signal	Chg. Let.
1	3	P1-9	P2-9	5	1	BRN } TWISTED PAIR	SPARE	
2		1	1			BLK }	SPARE	
3		3	3			GRN	XDATA	
4		4	4			BLK	XDATA	
5		5	5			WHT	TRDATA	
6		6	6			BLK	TRDATA	
7		7	7			BLU	TROTHER	
8		15	15			BLK	TROTHER	
9		10	10			RED	+5V	
10		11	11			BLK	+5V	
11		13	13			YEL } TWISTED PAIR	+15V	
12	3	P1-14	P2-14	5	1	BLK }	+15V	

1. Ref Item No's in Applicable Material List.
2. Ref Designations Are Abbreviated. Prefix Each Designation With:

Title: ASSEMBLY, CABLE - ETHERNET (EXTERNAL)

Xerox Corporation
El Segundo, California

XEROX

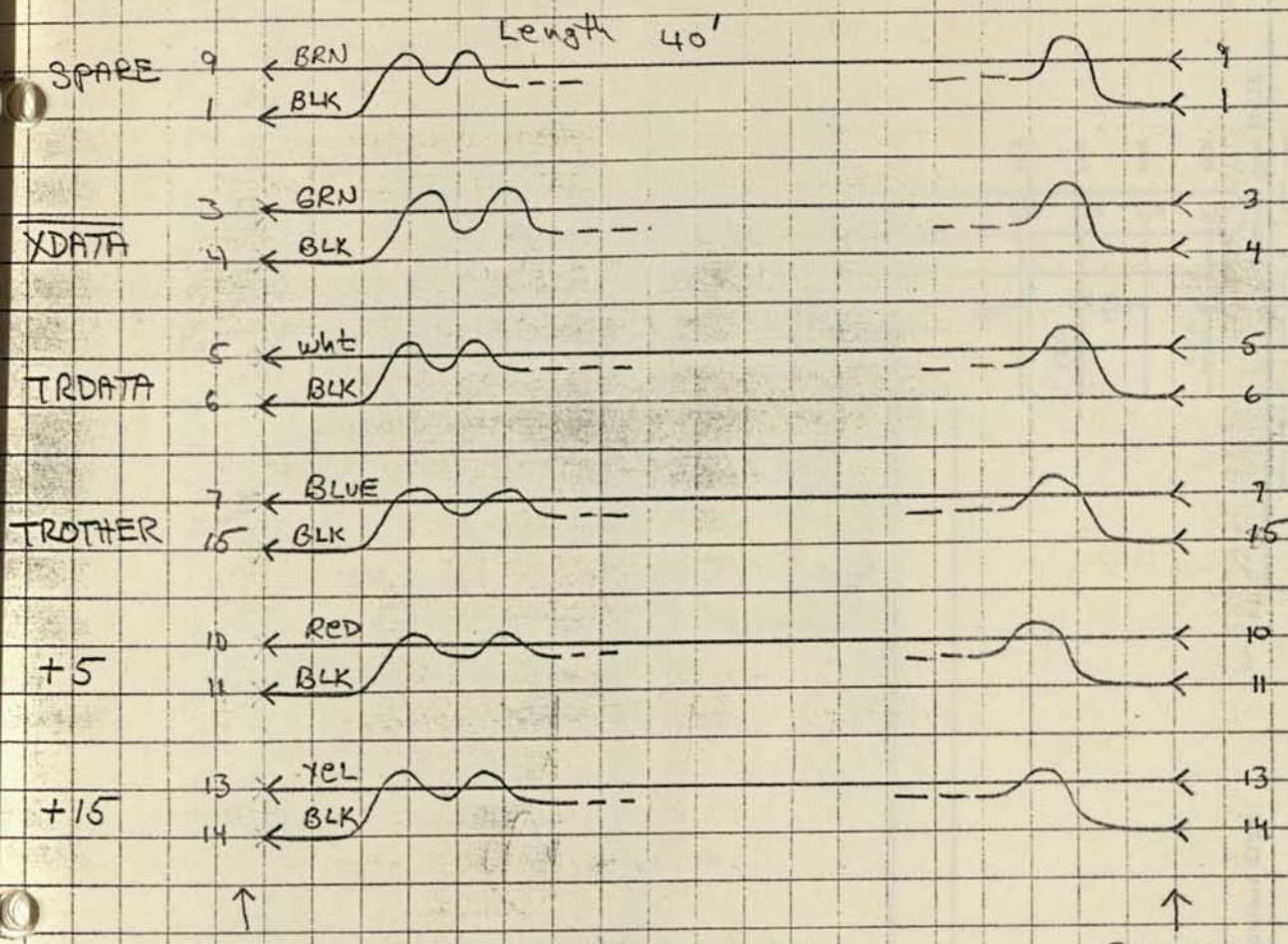
209590 A

Sheet 5 of 5

565(3/73)

ALTO

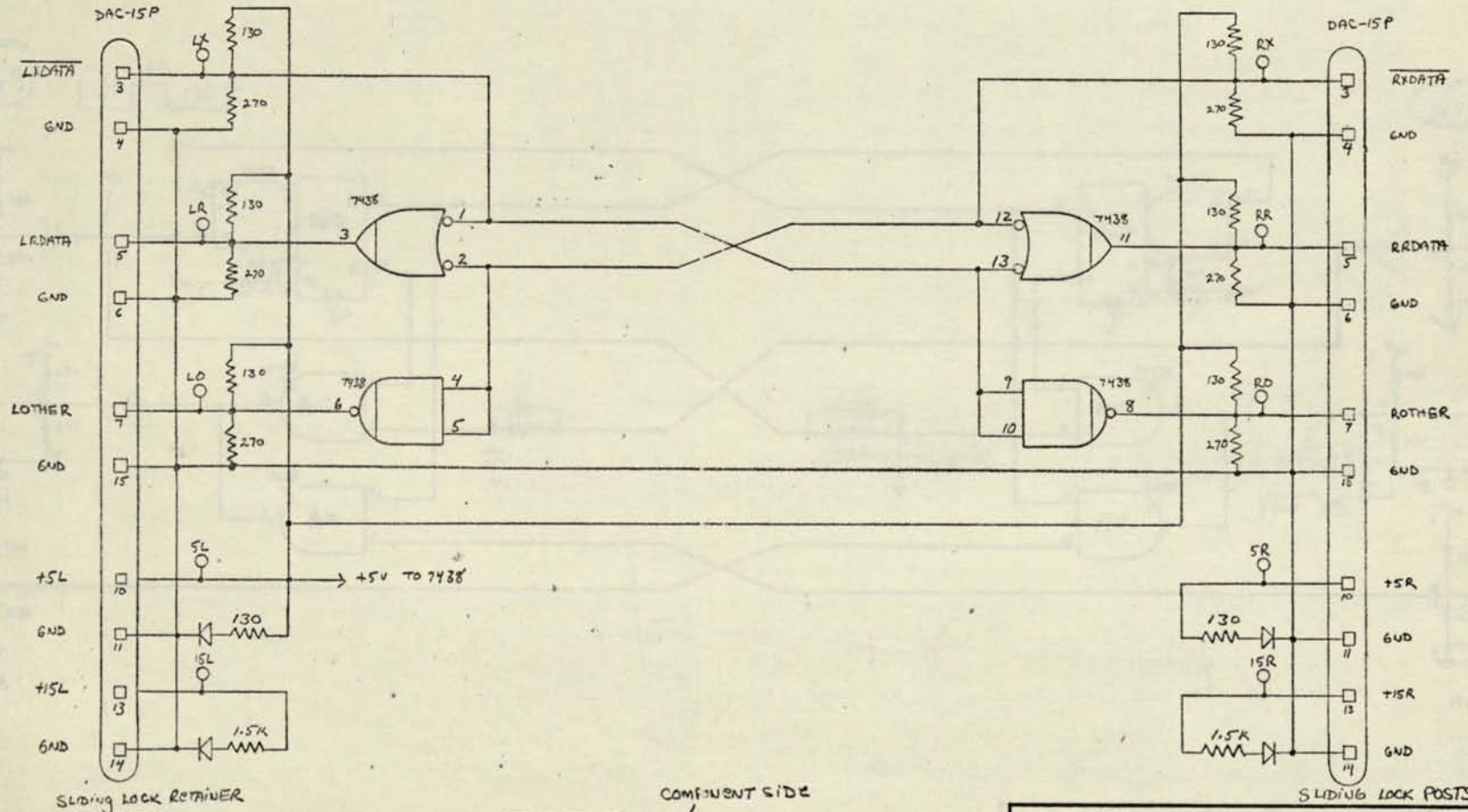
ETHERNET EXTERNAL CABLE 1 of 1 6/26/74 DB

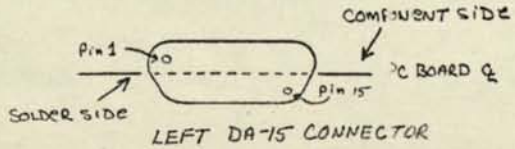


CANNON CABLE: BELDEN #747 CANNON

QTY	PART
(1)	DAC-15P SHELL
(1)	DA-51210-1 COVER
(12)	030-1952-000 PINS
(1)	DA-51220-1 SLIDING LOCK RETAINER

QTY	PART
(1)	DAC-15S SHEL
(1)	DA-51210-1 COVER
(12)	030-1953-000 PINS
(1)	DA-51220-1 SLIDING LOCK RETAINER





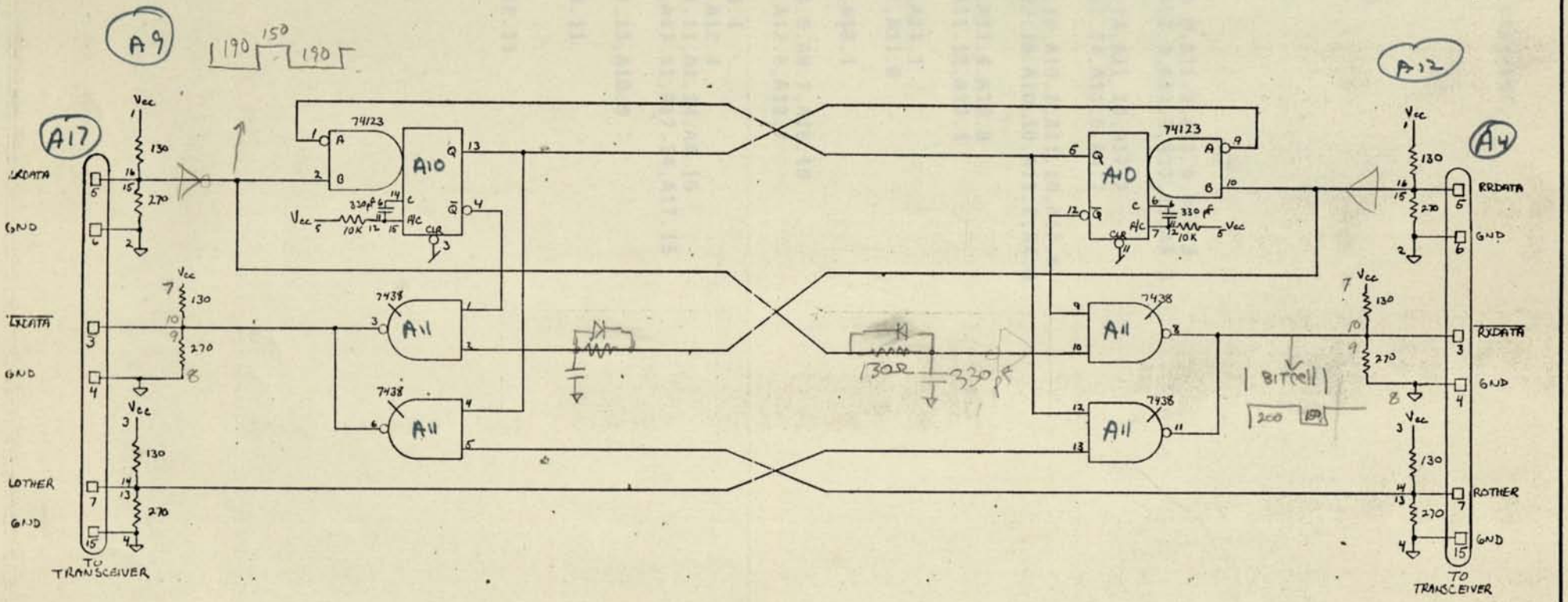
LEFT DA-15 CONNECTOR

ETHERNET

SCALE:	APPROVED BY:	DRAWN BY cb
DATE:		REVISED

DUMMY TRANSCEIVER

DRAWING NUMBER



ETHERNET

SCALE:	APPROVED BY:	DRAWN BY db
DATE:		REVISED

PACKET REPEATER

DRAWING NUMBER

```
@NOVA

**** an interim repeater


A4=A4: PLT

A9=A9: PLT
A10=A10: SN74123
A11=A11: SN7438
A12=A12: PLT

A17=A17: PLT
A19=A19: PLT
@

LXDATA':A9.10,A9.9,A11.3,A11.6,A17.3
RXDATA':A12.10,A12.9,A11.8,A11.11,A4.3

LOTHER:A9.13,A9.14,A11.13,A17.7
ROTHER:A12.13,A12.14,A11.5,A4.7

LRDATA:A9.15,A9.16,A10.2,A11.10,A17.5
RRDATA:A12.15,A12.16,A10.10,A11.2,A4.5

LCARRIER:A10.13,A11.4,A10.9
RCARRIER:A10.5,A11.12,A10.1

LCARRIER':A10.4,A11.1
RCARRIER':A10.12,A11.9

PU:A10.3,A10.11,A19.1

VCC:A9.1,A9.3,A9.5,A9.7,A19.16
VCC:A12.1,A12.3,A12.5,A12.7

GND:A9.2,A9.8,A9.4
GND:A12.2,A12.8,A12.4
GND:A4.4,A4.6,A4.11,A4.14,A4.15
GND:A17.4,A17.6,A17.11,A17.14,A17.15

PLUS15:A4.13,A17.13,A19.9

:A10.15,A9.12,A9.11
:A10.14,A9.6

:A10.7,A12.12,A12.11
:A10.6,A12.6
@
```

```
* FILE CREATED  6-OCT-75 21:44:50
* REPET          TOP WRAPS


GND001          A9*2      to    A1*8
GND002          A17*15    to    A9*8
GND003          A12*8     to    A20*7
GND005          A4*11     to    A4*6
GND005          A4*4      to    A4*14
GND007          A17*6     to    ~~C1*7~~ A17*8
LCARRIER        A10*13    to    A10*9
LOTHER          A9*14     to    A9*13
LRDATA          A10*2     to    A9*16
LRDATA          A9*15     to    A17*5
-LXDATA         A11*6     to    A9*9
-LXDATA         A9*10     to    A17*3
PLUS15          A19*9     to    A17*13
PU              A10*11    to    A19*1
RCARRIER        A10*5     to    A10*1
ROTHER          A12*13    to    A12*14
RRDATA          A11*2     to    A12*15
RRDATA          A12*16    to    A4*5
-RXDATA         A11*13    to    A11*10
-RXDATA         A12*10    to    A12*9
~~VCC002          A9*7      to    IL*16~~   A9*7 TO A9*1
~~VCC003          A9*1      to    IL*16~~
VCC004          A12*7     to    A20*16
VCC005          A12*1     to    A11*16
XXX001          A9*12     to    A9*11
XXX003          A12*11    to    A12*12
```

```
* FILE CREATED  6-OCT-75 21:44:53
* REPET          BOTTOM WRAPS


GND001        A9*4      to   A9*2
GND002        A17*14    to   A17*15
GND002        A9*8      to   A10*8
GND003        A12*4     to   A12*8
GND004        A5*8      to   A12*2
GND005        A5*7      to   A4*11
GND005        A4*6      to   A4*4
GND005        A4*14     to   A4*15
GND006        A18*7     to   A17*11
GND007        A17*4     to   A17*6
LCARRIER      A11*4     to   A10*13
-LCARRIER     A11*1     to   A10*4
LOTHER        A11*15    to   A9*14
LOTHER        A9*13     to   A17*7
LRDATA        A11*12    to   A10*2
LRDATA        A9*16     to   A9*15
-LXDATA       A11*3     to   A11*6
-LXDATA       A9*9      to   A9*10
PLUS15        A4*13     to   A19*9
PU            A10*3     to   A10*11
RCARRIER      A11*14    to   A10*5
-RCARRIER     A11*11    to   A10*12
ROTHER        A11*5     to   A12*13
ROTHER        A12*14    to   A4*7
RRDATA        A10*10    to   A11*2
RRDATA        A12*15    to   A12*16
-RXDATA       A4*3      to   A11*13
-RXDATA       A11*10    to   A12*10
VCC001        C3*16     to   A19*16
VCC002        A9*5      to   A9*7
VCC003        A9*3      to   A9*1
VCC004        A12*5     to   A12*7
VCC005        A12*3     to   A12*1
XXX001        A10*15    to   A9*12
XXX002        A9*6      to   A10*14
XXX003        A10*7     to   A12*11
XXX004        A12*6     to   A10*6

Traces to be cut

A4*7    A9*7    A10*7   A4*8    A4*18   A9*8    A9*16   A12*7   A12*8   A12*16
A17*7   A17*8   A17*16  A19*7   A19*8   A19*16
```

* FILE CREATED 6-OCT-75 21:44:54
* REPET MAIN WIRELIST

The chips run parallel to their names

SUBCARDS A C
B D

	0	17	34	51	68	85
17	-	-	-	-	-	-
16	-	-	-	-	-	-
15	-	-	-	-	-	-
14	-	-	-	-	-	-
13	PLT	PLT	-	-	-	-
12	-	7438	PLT	-	-	-
11	-	74123	-	-	-	-
10	-	PLT	PLT	-	-	-
9	-	-	-	-	-	-
8	-	-	-	-	-	-
7	-	-	-	-	-	-
6	-	-	-	-	-	-
5	-	-	-	-	-	-
4	-	-	-	-	-	-
3	-	-	-	-	-	-
2	-	-	-	-	-	-
1	-	-	-	-	-	-

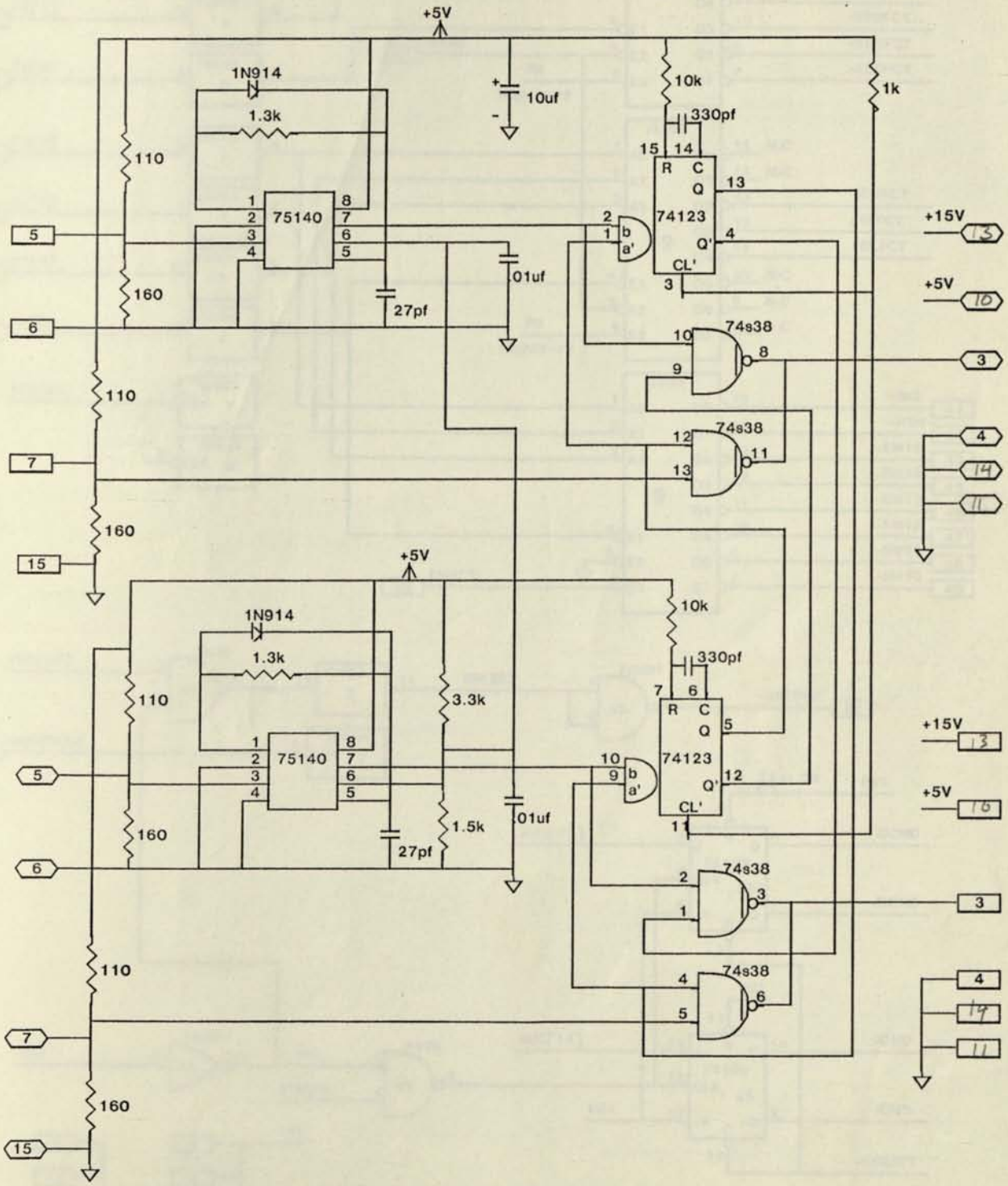
$$ +:							0	$
GND001:	A9*4	A9*2	A1*8				0	1
GND002:	A17*14	A17*15	A9*8	A10*8			0	2
GND003:	A12*4	A12*8	A20*7				0	3
GND004:	A5*8	A12*2					0	4
GND005:	A5*7	A4*11	A4*6	A4*4	A4*14	A4*15	0	5
GND006:	A18*7	A17*11					0	6
GND007:	A17*4	A17*6	C1*7				0	7
LCARRIER:	A11*4	A10*13	A10*9				-128	8
-LCARRIER:	A11*1	A10*4					-144	9
LOTHER:	A11*15	A9*14	A9*13	A17*7			16	10
LRDATA:	A11*12	A10*2	A9*16	A9*15	A17*5		32	11
-LXDATA:	A11*3	A11*6	A9*9	A9*10	A17*3		-960	12
PLUS15:	A4*13	A19*9	A17*13				0	13
PU:	A10*3	A10*11	A19*1				32	14
RCARRIER:	A11*14	A10*5	A10*1				-128	15
-RCARRIER:	A11*11	A10*12					-144	16
ROTHER:	A11*5	A12*13	A12*14	A4*7			16	17
RRDATA:	A10*10	A11*2	A12*15	A12*16	A4*5		32	18
-RXDATA:	A4*3	A11*13	A11*10	A12*10	A12*9		-960	19
VCC001:	C3*16	A19*16					0	20
VCC002:	A9*5	A9*7	IL*16				0	21
VCC003:	A9*3	A9*1	IL*16				0	22
VCC004:	A12*5	A12*7	A20*16				0	23
VCC005:	A12*3	A12*1	A11*16				0	24
XXX001:	A10*15	A9*12	A9*11				0	25
XXX002:	A9*6	A10*14					0	26
XXX003:	A10*7	A12*11	A12*12				0	27
XXX004:	A12*6	A10*6					0	28

* This card uses 63 wires

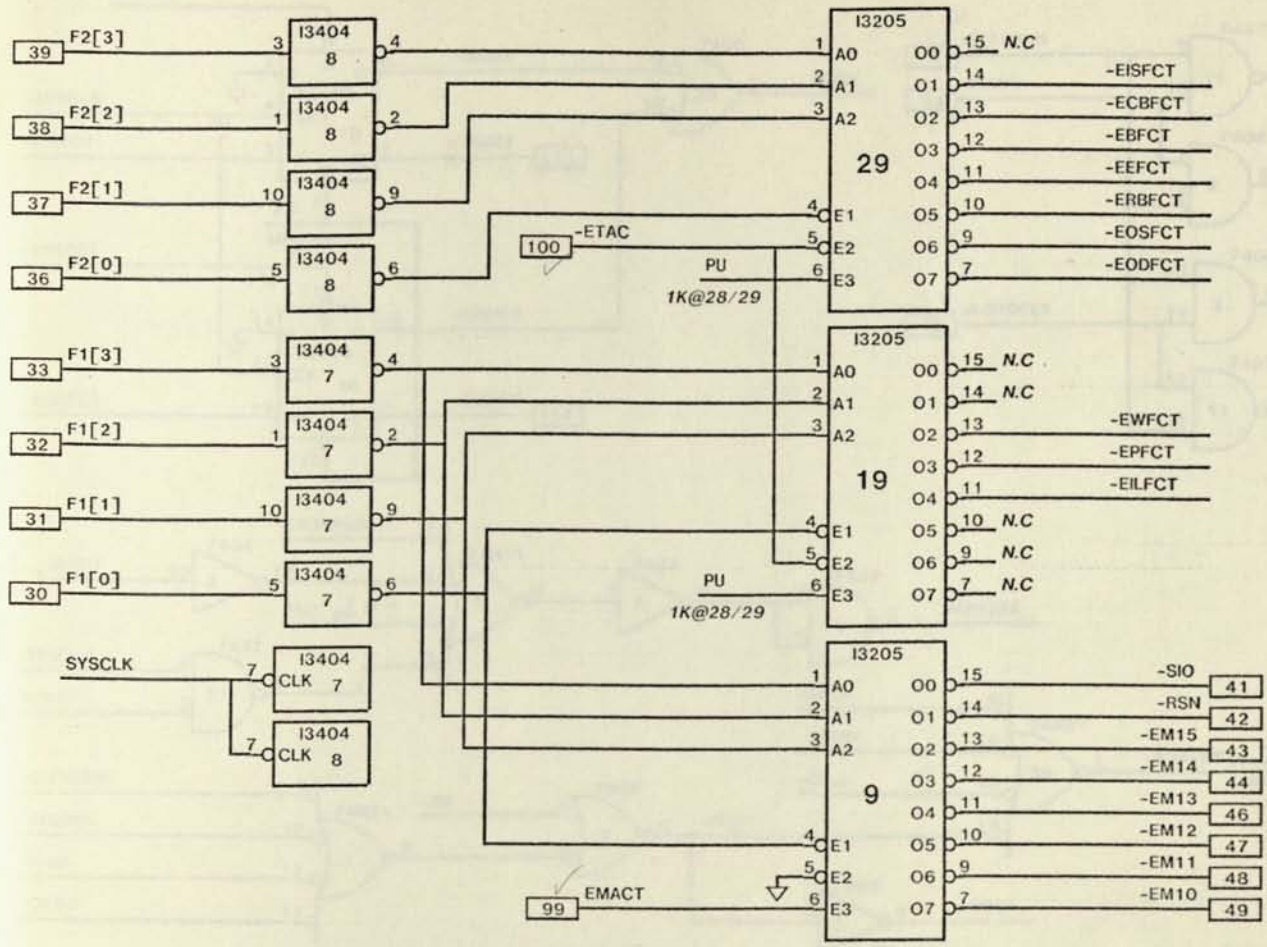
Traces to be cut

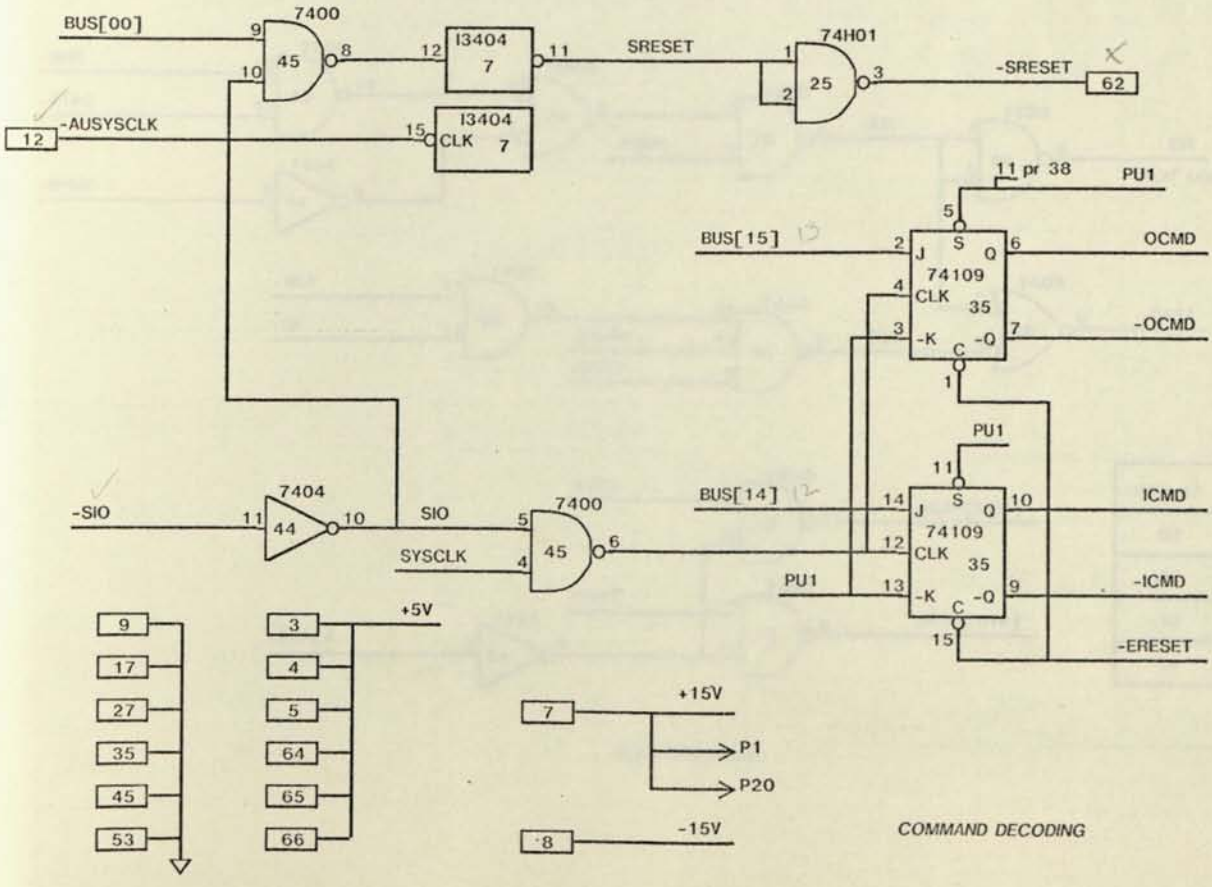
A4*7	A9*7	A10*7	A4*8	A4*16	A9*8	A9*16	A12*7	A12*8	A12*16
A17*7	A17*8	A17*16	A19*7	A19*8	A19*16				

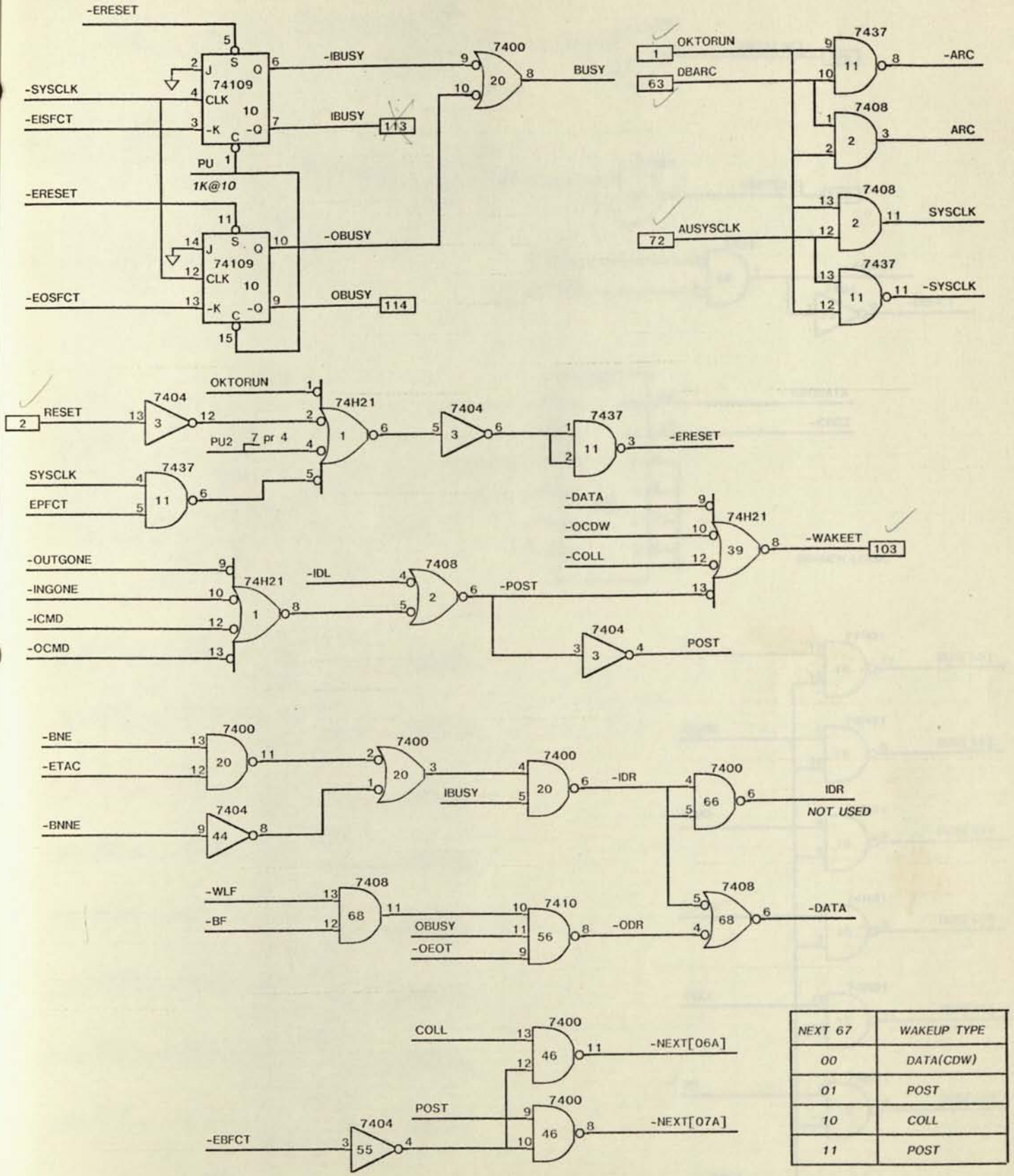
A1*8	GND001	A4*3	-RXDATA	A4*4	GND005	A4*5	RRDATA
A4*6	GND005	A4*7	ROTHER	A4*11	GND005	A4*13	PLUS15
A4*14	GND005	A4*15	GND005	A5*7	GND005	A5*8	GND004
IL*16	VCC003	A9*1	VCC003	A9*2	GND001	A9*3	VCC003
A9*4	GND001	A9*5	VCC002	A9*6	XXX002	A9*7	VCC002
A9*8	GND002	A9*9	-LXDATA	A9*10	-LXDATA	A9*11	XXX001
A9*12	XXX001	A9*13	LOTHER	A9*14	LOTHER	A9*15	LRDATA
A9*16	LRDATA	A10*1	RCARRIER	A10*2	LRDATA	A10*3	PU
A10*4	-LCARRIER	A10*5	RCARRIER	A10*6	XXX004	A10*7	XXX003
A10*8	GND002	A10*9	LCARRIER	A10*10	RRDATA	A10*11	PU
A10*12	-RCARRIER	A10*13	LCARRIER	A10*14	XXX002	A10*15	XXX001
A11*1	-LCARRIER	A11*2	RRDATA	A11*3	-LXDATA	A11*4	LCARRIER
A11*5	ROTHER	A11*6	-LXDATA	A11*10	-RXDATA	A11*11	-RCARRIER
A11*12	LRDATA	A11*13	-RXDATA	A11*14	RCARRIER	A11*15	LOTHER
A11*16	VCC005	A12*1	VCC005	A12*2	GND004	A12*3	VCC005
A12*4	GND003	A12*5	VCC004	A12*6	XXX004	A12*7	VCC004
A12*8	GND003	A12*9	-RXDATA	A12*10	-RXDATA	A12*11	XXX003
A12*12	XXX003	A12*13	ROTHER	A12*14	ROTHER	A12*15	RRDATA
A12*16	RRDATA	IL*16	VCC002	A17*3	-LXDATA	A17*4	GND007
A17*5	LRDATA	A17*6	GND007	A17*7	LOTHER	A17*11	GND006
A17*13	PLUS15	A17*14	GND002	A17*15	GND002	A18*7	GND006
A19*1	PU	A19*9	PLUS15	A19*16	VCC001	A20*7	GND003
A20*16	VCC004	C1*7	GND007	C3*16	VCC001		



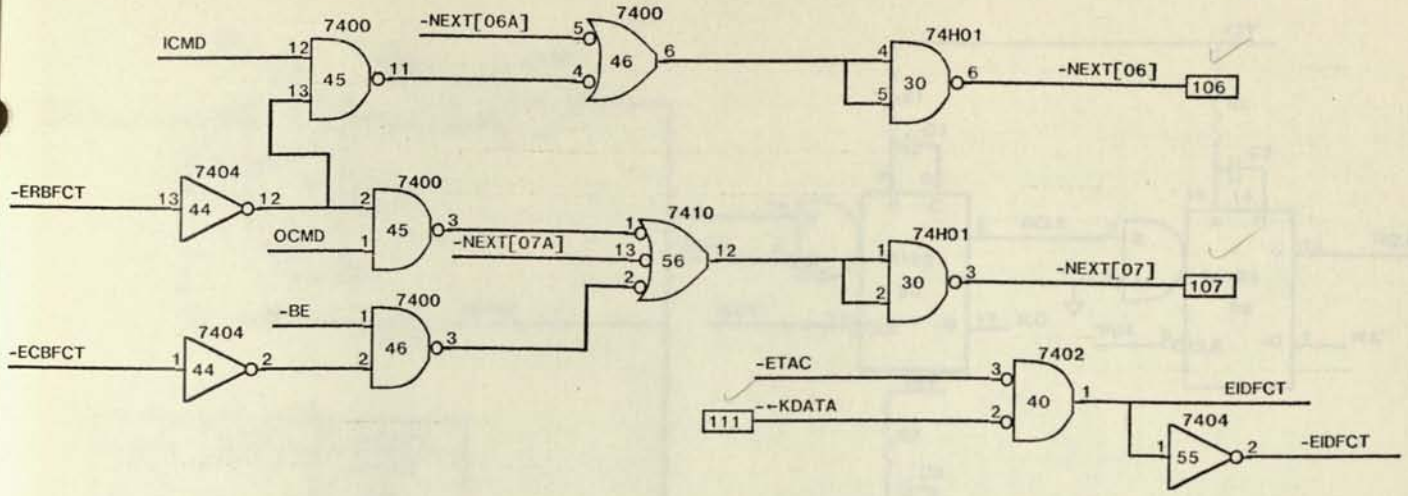
XEROX PARC	Project	ETHERNET REPEATER	File REPEATER	Designer D BOGGS	Rev B	Date 4-8-78	Page 1

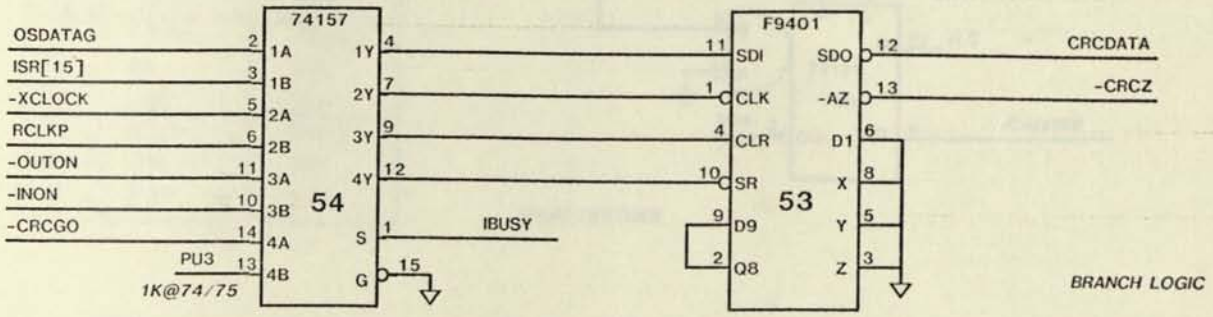

F2[3]
F2[2]
F2[1]
F2[0]
I3404 8
I3205
29
A0
A1
A2
E1
E2
E3
O0
O1
O2
O3
O4
O5
O6
O7
N.C
-EISFCT
-ECBFCT
-EBFCT
-EEFCT
-ERBFCT
-EOSFCT
-EODFCT
-ETAC
PU
1K@28/29
F1[3]
F1[2]
F1[1]
F1[0]
I3404 7
19
-EWFCT
-EPFCT
-EILFCT
SYSCLK
I3404 CLK 7
I3404 CLK 8
9
-SIO
-RSN
-EM15
-EM14
-EM13
-EM12
-EM11
-EM10
EMACT

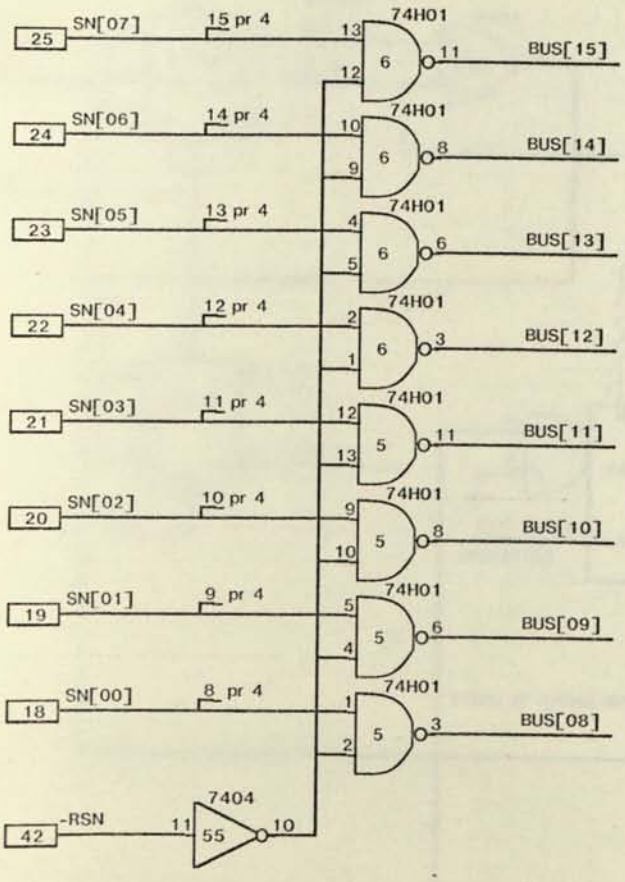

BUS[00]
7400
45
I3404 7
SRESET
74H01
25
-SRESET
-AUSYSCLK
I3404 CLK 7
11 pr 38
PU1
BUS[15]
74109
CLK
35
-K
-Q
J
S
Q
C
OCMD
-OCMD
BUS[14]
ICMD
-ICMD
-ERESET
-SIO
7404
44
SIO
SYSCLK
+5V
+15V
P1
P20
-15V
COMMAND DECODING

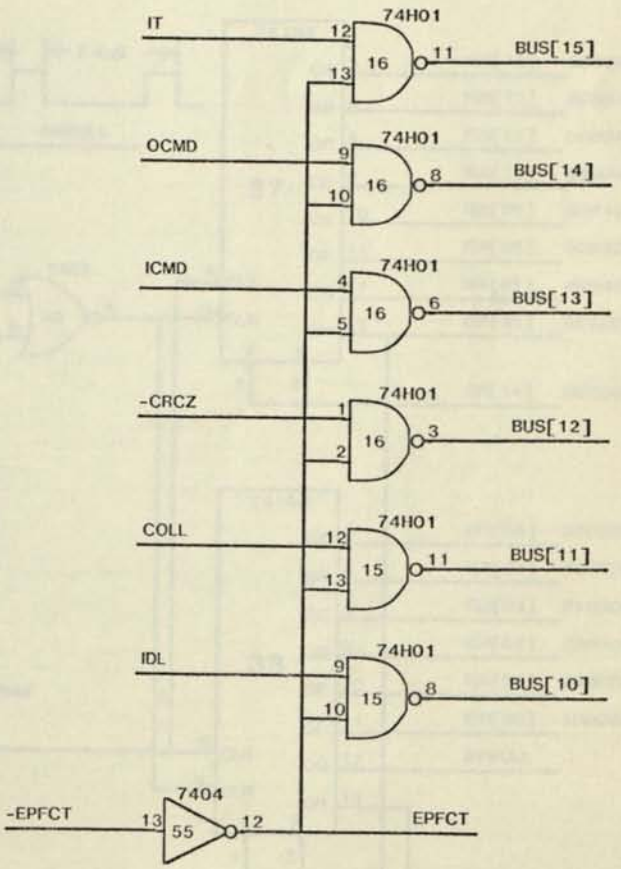


NEXT 67	WAKEUP TYPE
00	DATA(CDW)
01	POST
10	COLL
11	POST

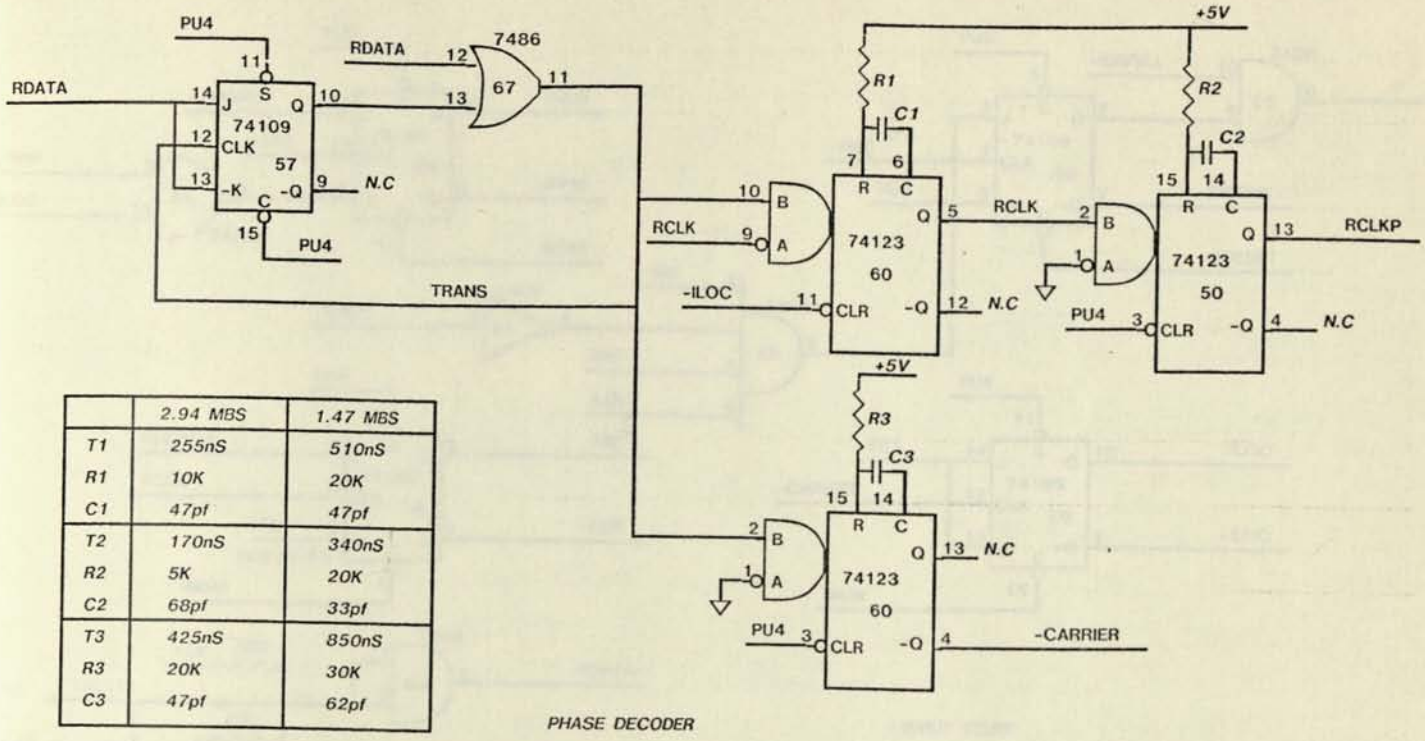






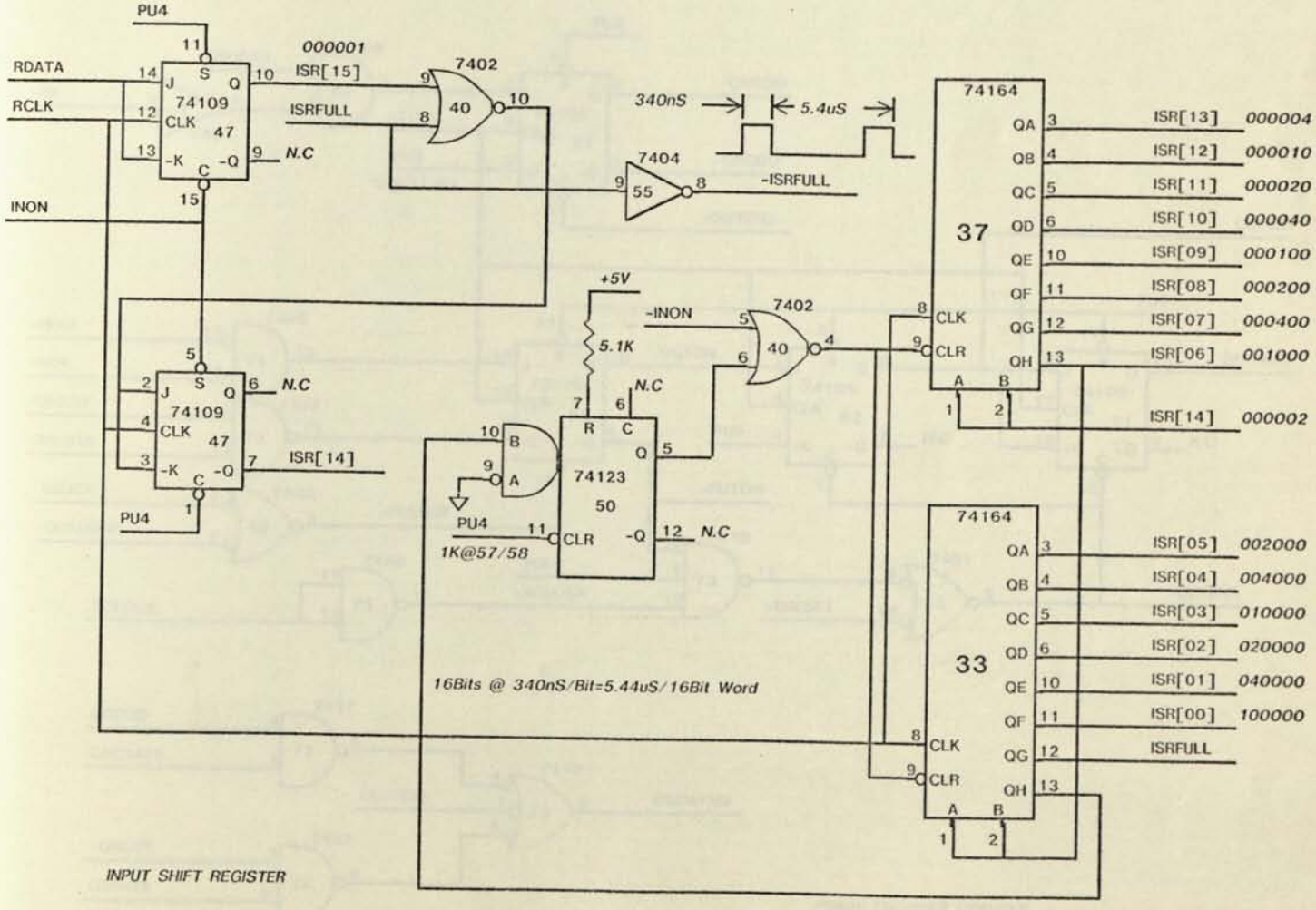


SERIAL NUMBER & STATUS

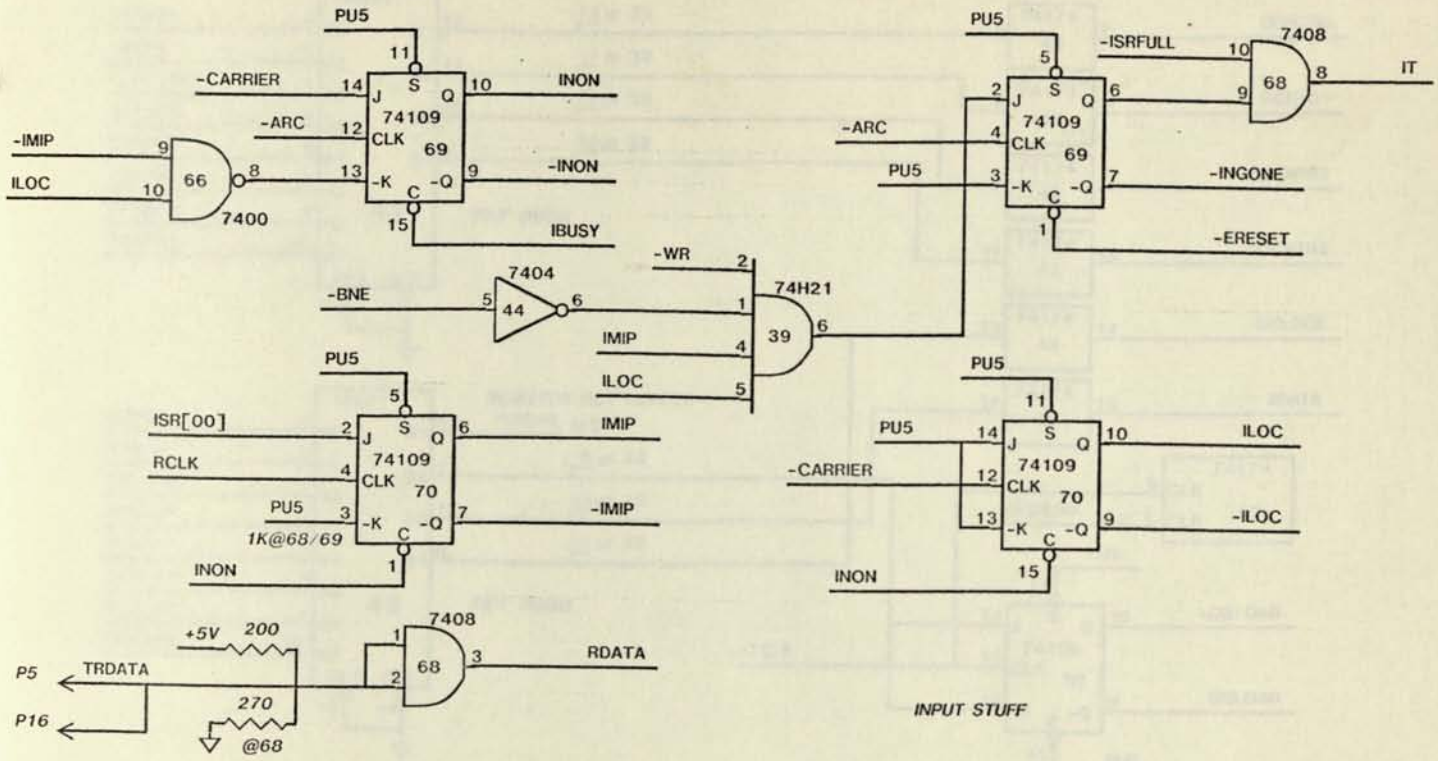


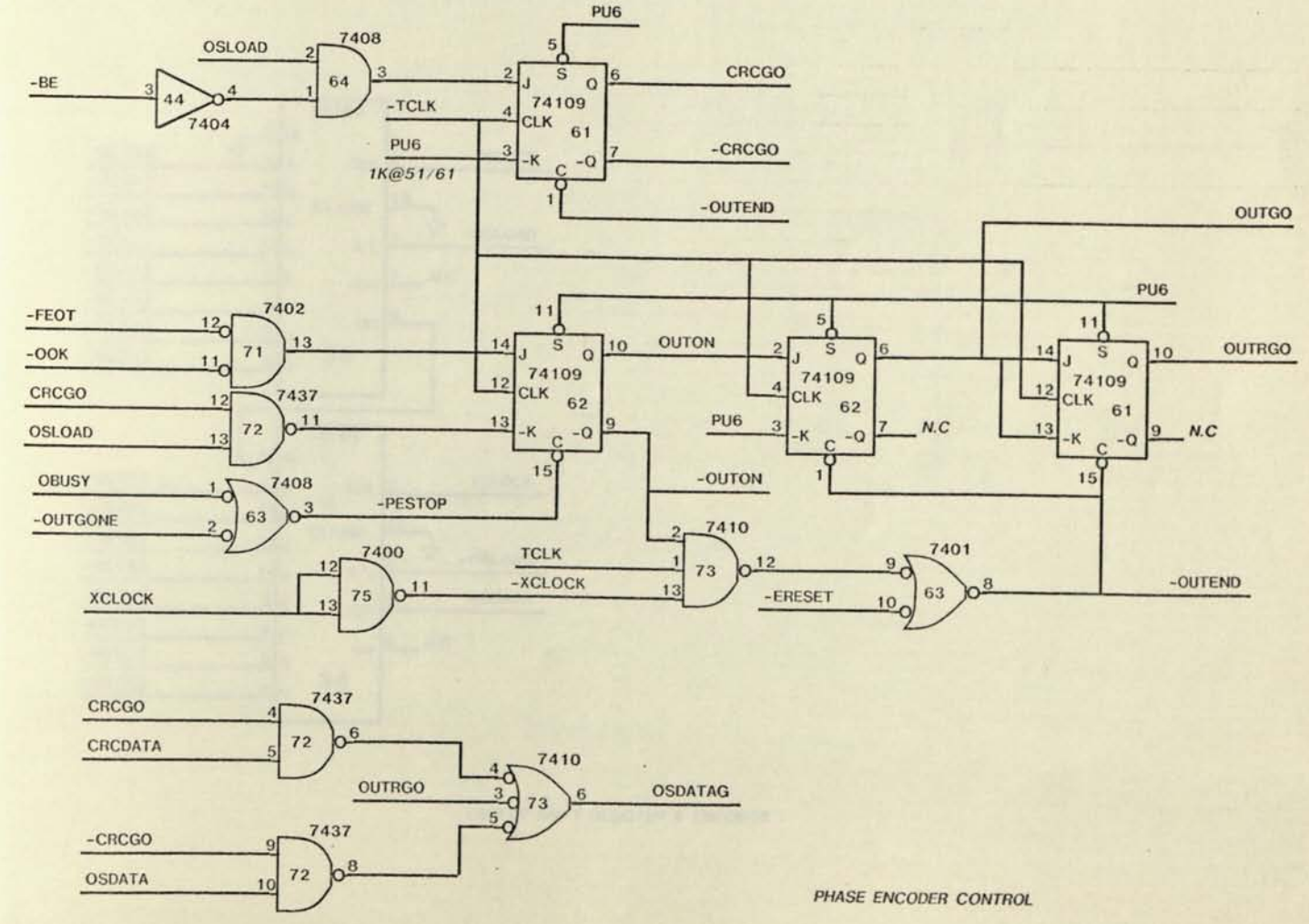
	2.94 MBS	1.47 MBS
T1	255nS	510nS
R1	10K	20K
C1	47pf	47pf
T2	170nS	340nS
R2	5K	20K
C2	68pf	33pf
T3	425nS	850nS
R3	20K	30K
C3	47pf	62pf

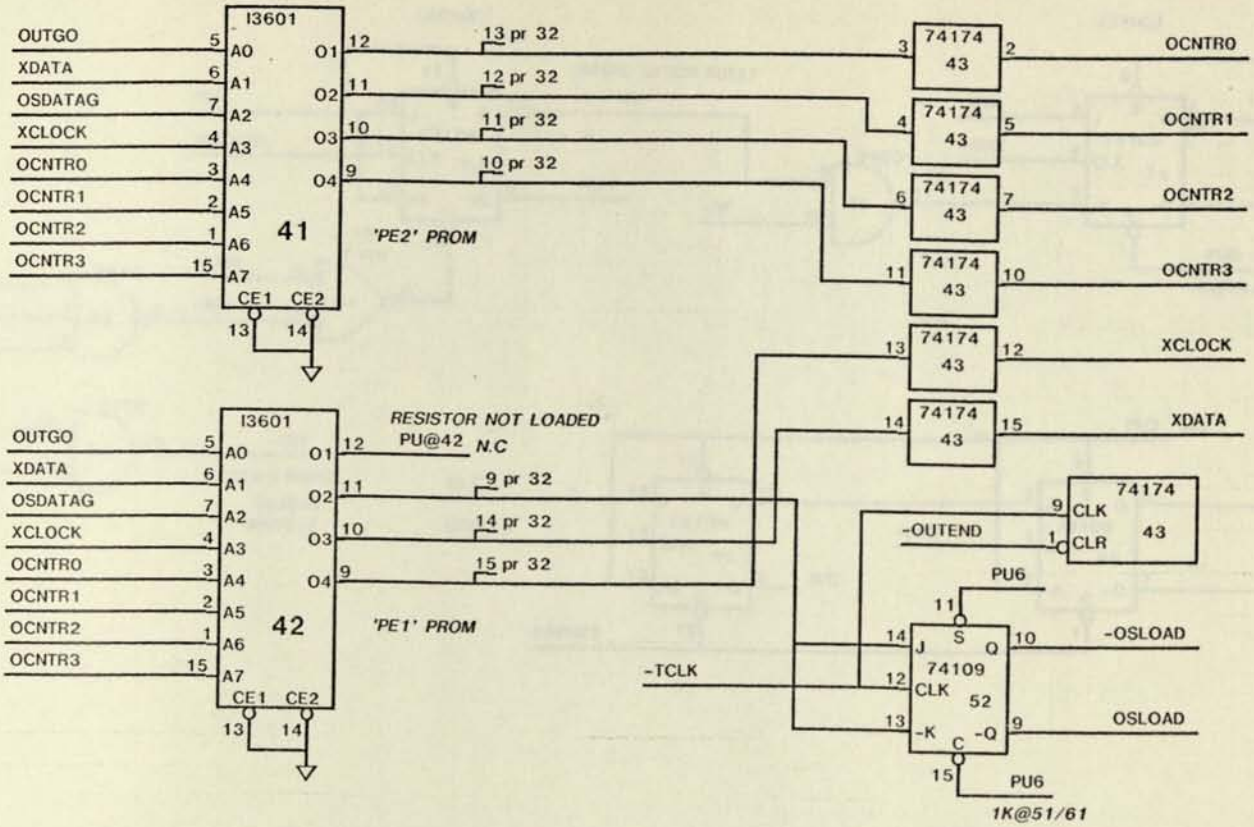
PHASE DECODER

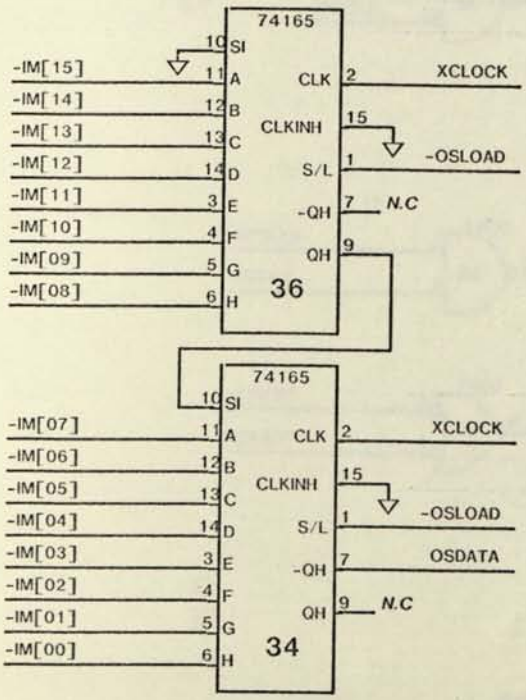


INPUT SHIFT REGISTER

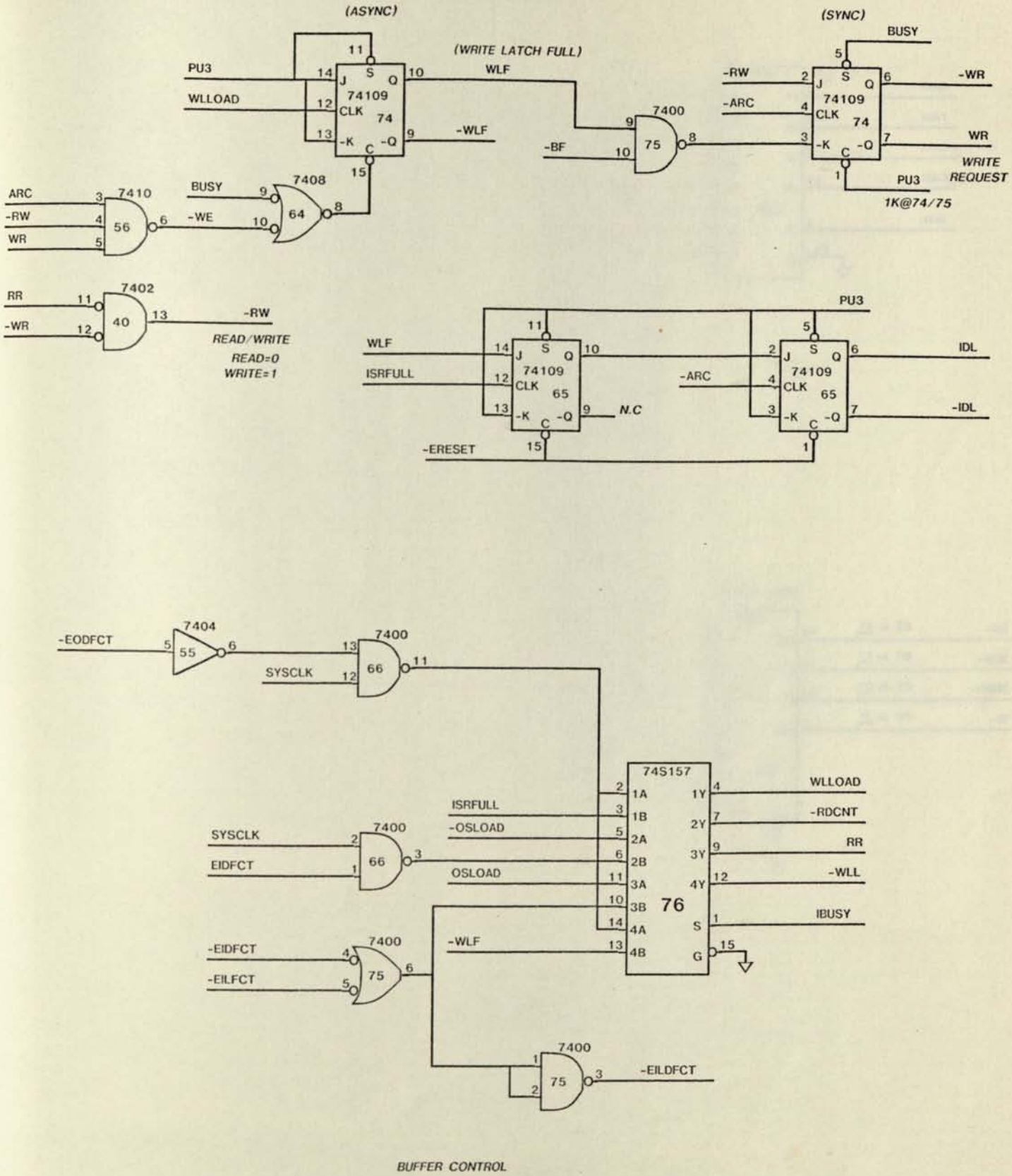

PU5
-CARRIER
-ARC
-IMIP
ILOC
66
7400
74109
CLK
69
INON
-INON
IBUSY
-BNE
7404
44
-WR
74H21
39
IMIP
ILOC
ISR[00]
RCLK
PU5
1K@68/69
70
-IMIP
INON
+5V
200
270
@68
P5
TRDATA
P16
7408
68
RDATA
-ISRFULL
IT
-INGONE
-ERESET
INPUT STUFF

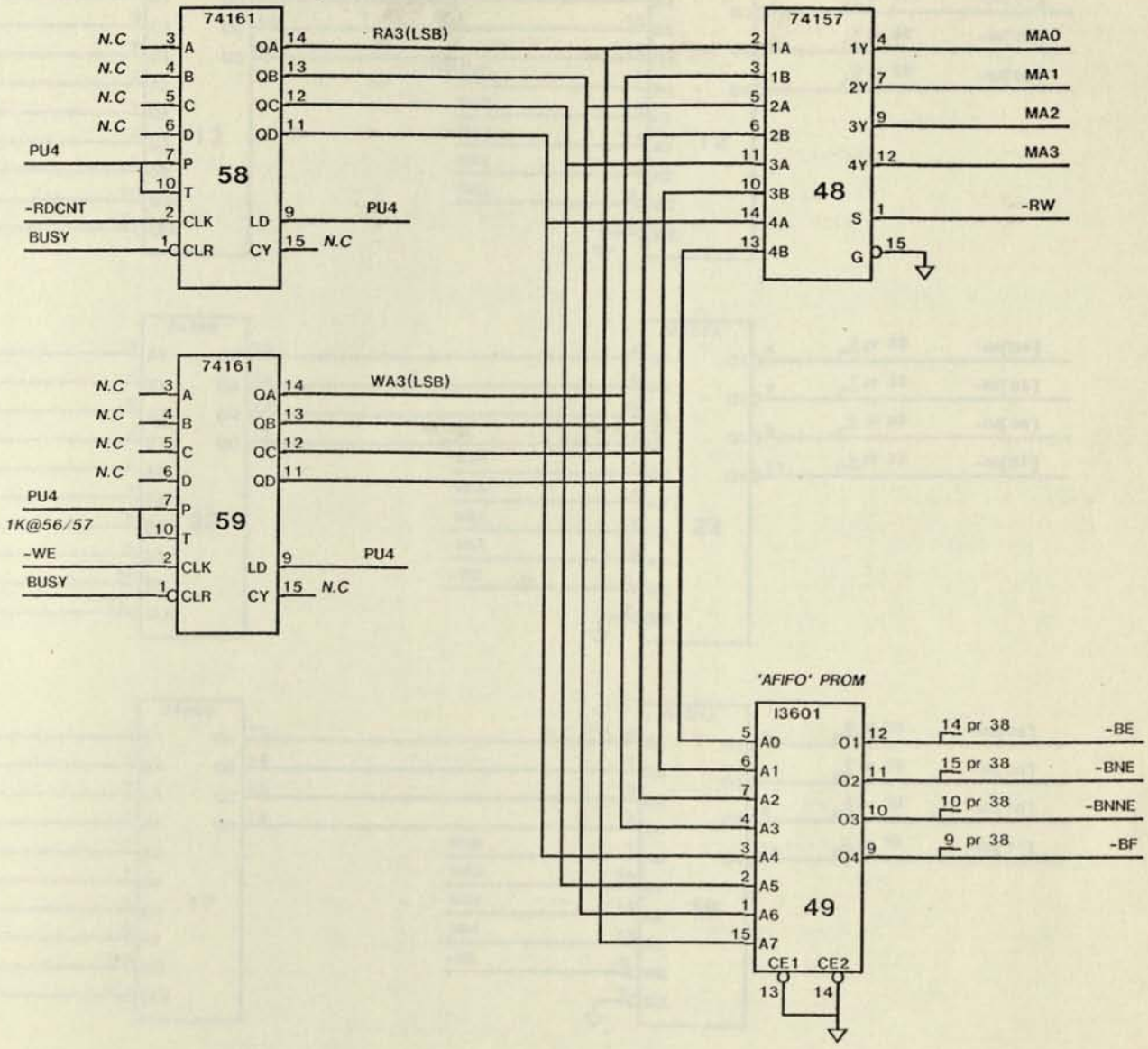

PU6
OSLOAD
-BE
7404
44
7408
64
-TCLK
1K@51/61
74109
CLK
61
CRCGO
-CRCGO
-OUTEND
OUTGO
-FEOT
-OOK
7402
71
7437
72
OSLOAD
OUTON
62
PU6
N.C
OUTRGO
61
OBUSY
-OUTGONE
7408
63
-PESTOP
-OUTON
7410
73
TCLK
-XCLOCK
XCLOCK
7400
75
-ERESET
7401
63
-OUTEND
CRCDATA
OSDATAG
-CRCGO
OSDATA
PHASE ENCODER CONTROL



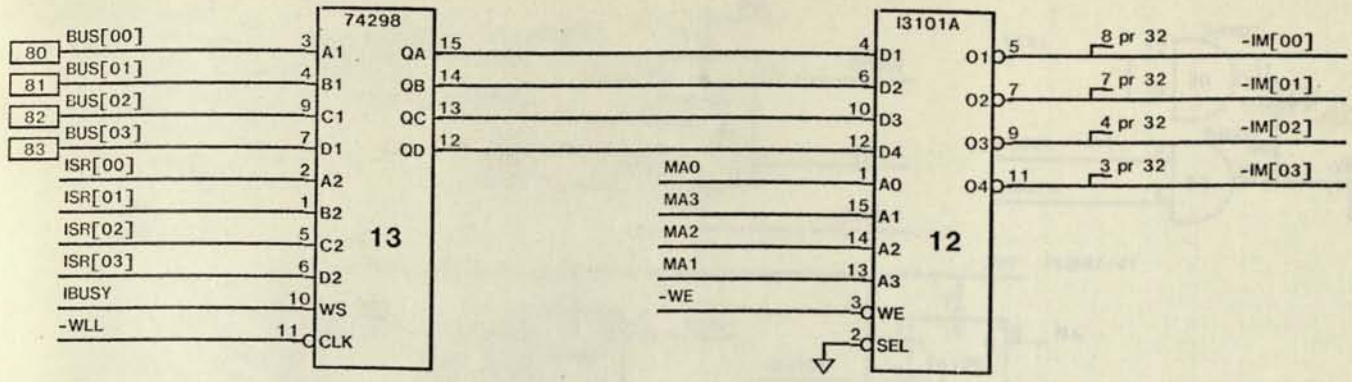


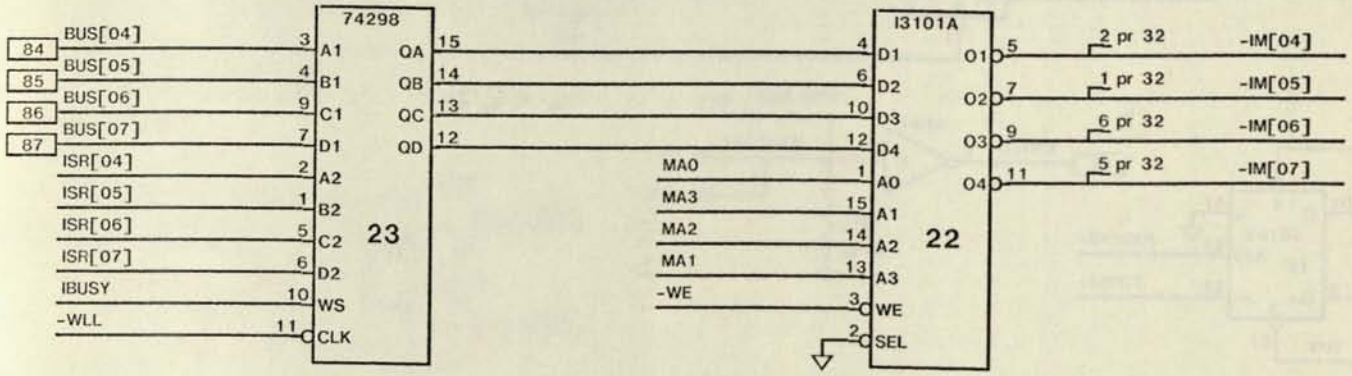
OUTPUT SHIFT REGISTER & ENCODER

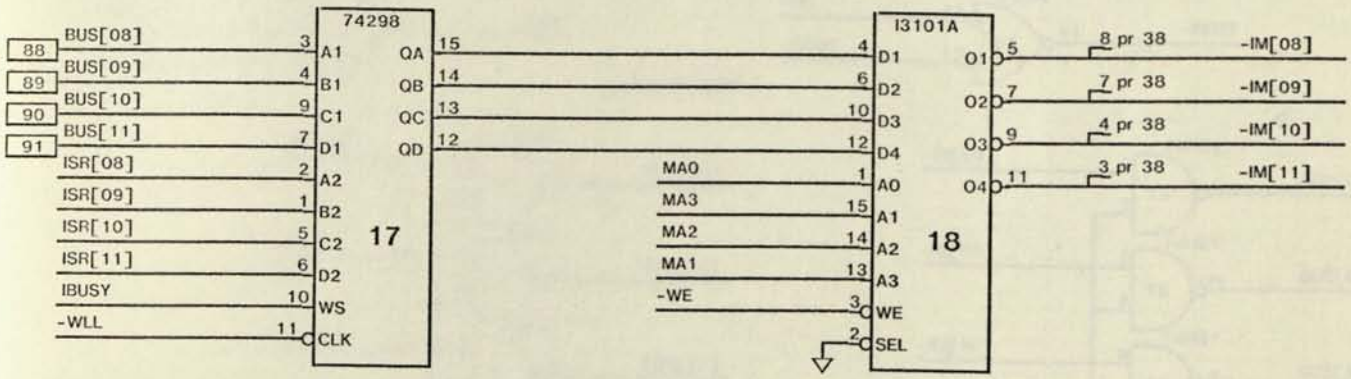

(ASYNC)
(WRITE LATCH FULL)
(SYNC)
PU3
WLLOAD
WLF
-WLF
74109
74
CLK
-K
-Q
-BF
7400
75
BUSY
-RW
-ARC
-WR
WR
WRITE REQUEST
PU3
1K@74/75
ARC
-RW
WR
7410
56
BUSY
-WE
7408
64
RR
-WR
7402
40
-RW
READ/WRITE
READ=0
WRITE=1
WLF
ISRFULL
-ARC
-ERESET
N.C
IDL
-IDL
65
-EODFCT
7404
55
SYSCLK
66
EIDFCT
OSLOAD
-OSLOAD
ISRFULL
-EIDFCT
-EILFCT
-WLF
74S157
76
1A
1B
2A
2B
3A
3B
4A
4B
1Y
2Y
3Y
4Y
S
G
WLLOAD
-RDCNT
RR
-WLL
IBUSY
-EILDFCT
BUFFER CONTROL

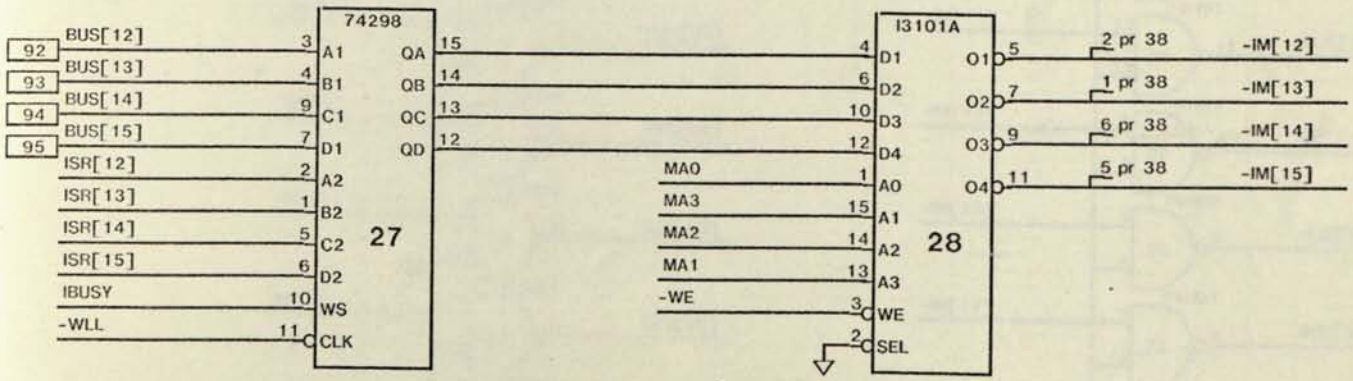


HALF DUPLEX BUFFER

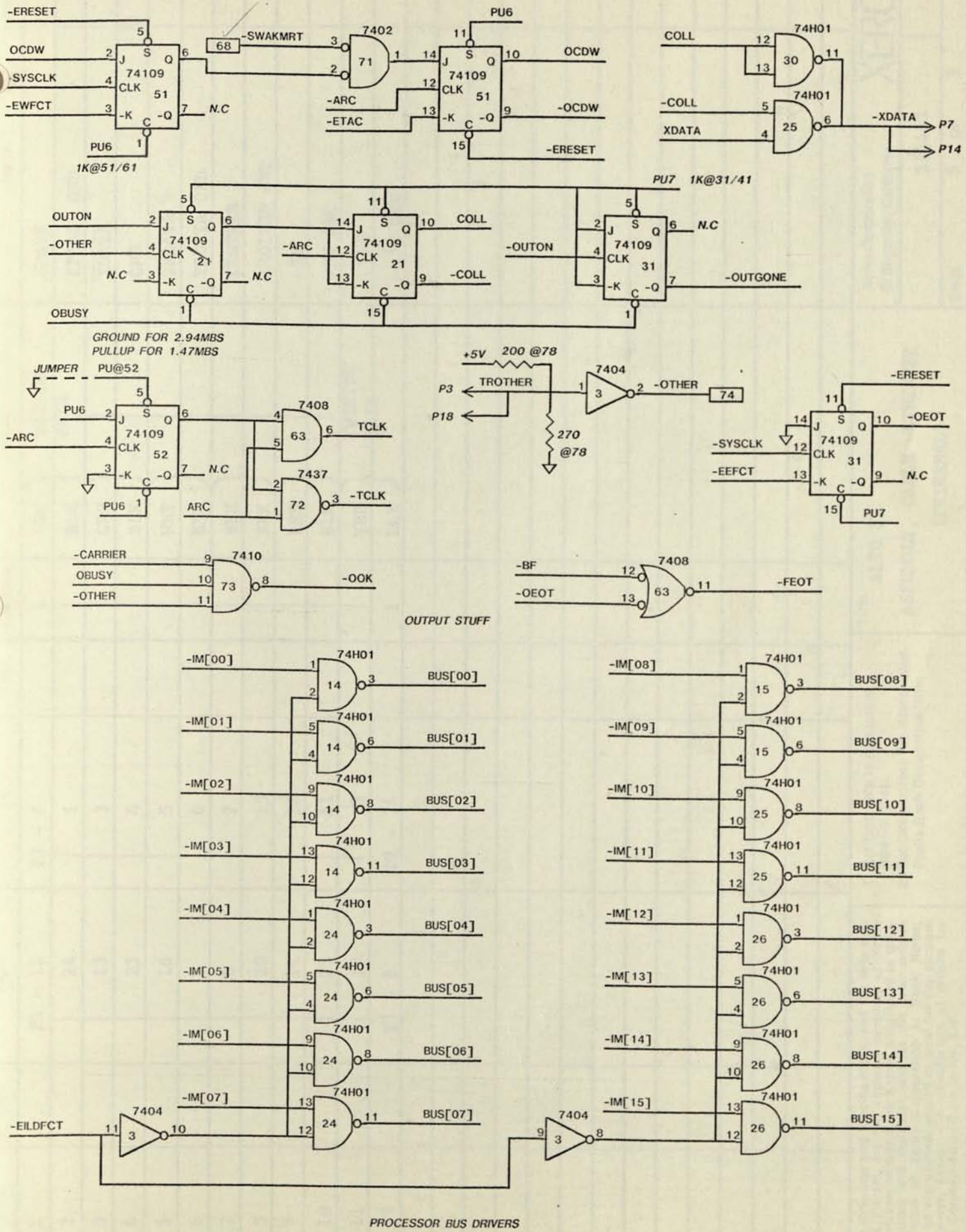








HALF DUPLEX BUFFER


-ERESET
OCDW
-SYSCLK
-EWFCT
74109
CLK
51
PU6
1K@51/61
N.C
68
-SWAKMRT
7402
71
PU6
-ARC
-ETAC
OCDW
-OCDW
-ERESET
COLL
-COLL
XDATA
74H01
30
25
-XDATA
P7
P14
PU7
1K@31/41
OUTON
-OTHER
OBUSY
21
-OUTON
31
-OUTGONE
GROUND FOR 2.94MBS
PULLUP FOR 1.47MBS
JUMPER
PU@52
52
TCLK
-TCLK
7408
7437
63
72
+5V
200 @78
P3
P18
270
@78
TROTHER
7404
-OTHER
74
-EEFCT
-OEOT
-CARRIER
7410
73
-OOK
-BF
-FEOT
OUTPUT STUFF
-IM[00]
-IM[01]
-IM[02]
-IM[03]
-IM[04]
-IM[05]
-IM[06]
-IM[07]
BUS[00]
BUS[01]
BUS[02]
BUS[03]
BUS[04]
BUS[05]
BUS[06]
BUS[07]
-EILDFCT
-IM[08]
-IM[09]
-IM[10]
-IM[11]
-IM[12]
-IM[13]
-IM[14]
-IM[15]
BUS[08]
BUS[09]
BUS[10]
BUS[11]
BUS[12]
BUS[13]
BUS[14]
BUS[15]
PROCESSOR BUS DRIVERS

Wire No.	Term	From	To	Term	Wire Type	Notes	Signal	Chg. Let.
1		P1 - 12	P2 - 9		1	BRN } TWISTED PAIR	SPARE	
2		24	1			BLK }	SPARE GND	
3		13	3			GRN }	$\overline{\text{XDATA}}$	
4		25	4			BLK }	GND	
5		16	5			WHT }	TRDATA	
6		3	6			BLK }	TRDATA GND	
7		8	7			BLU }	TROTHER	
8		20	15			BLK }	TROTHER GND	
9		11	10			RED }	+5V	
10		23	11			BLK }	+5V GND	
11		19	13			YEL } TWISTED PAIR	+15V	
12		P1 - 6	P2 - 14		1	BLK }	+15V GND	

1. Ref Item No's in Applicable Material List.
2. Ref Designations Are Abbreviated. Prefix Each Designation With:

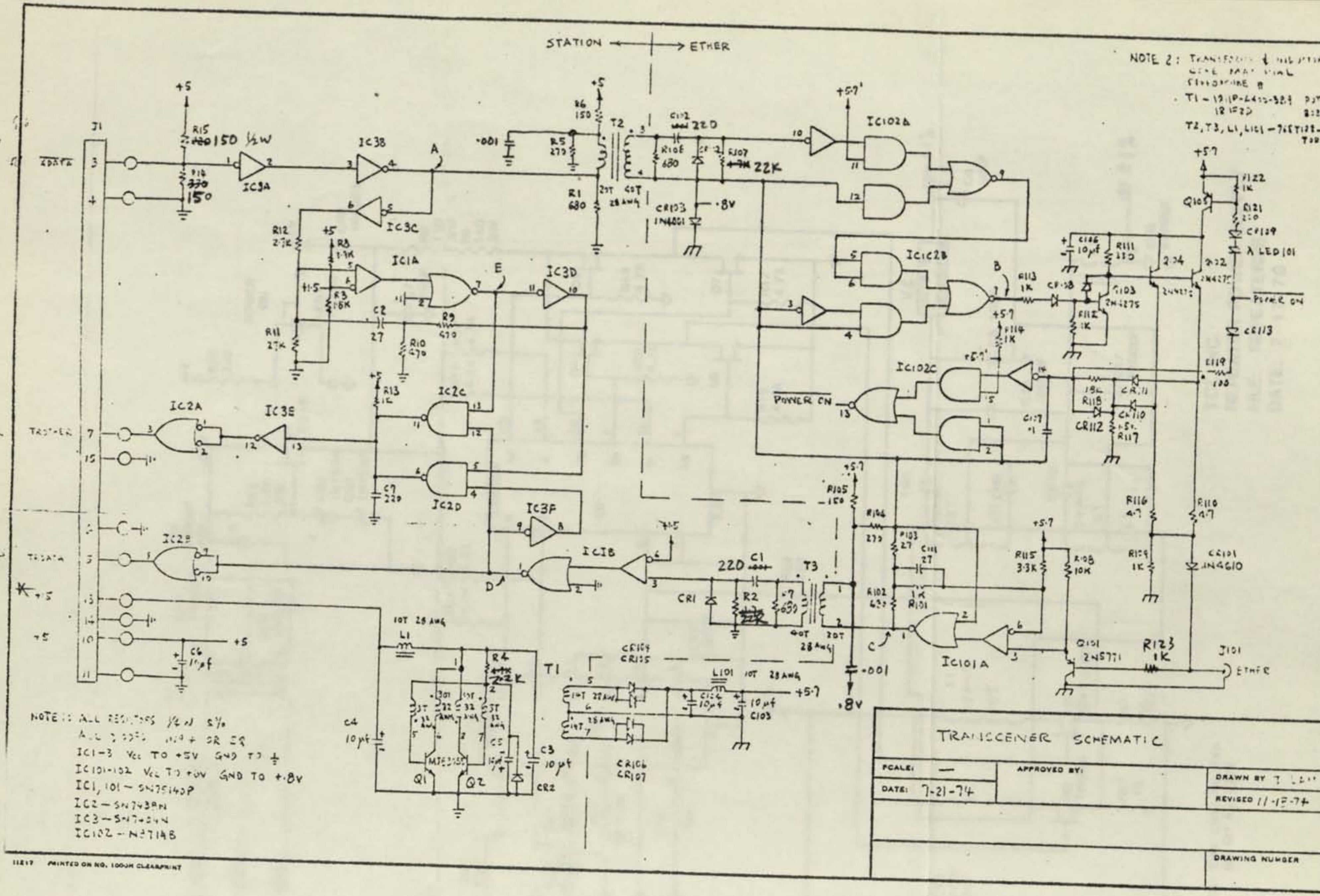
Title
ALTO II
ASSEMBLY, CABLE -ETHERNET
(EXTERNAL)

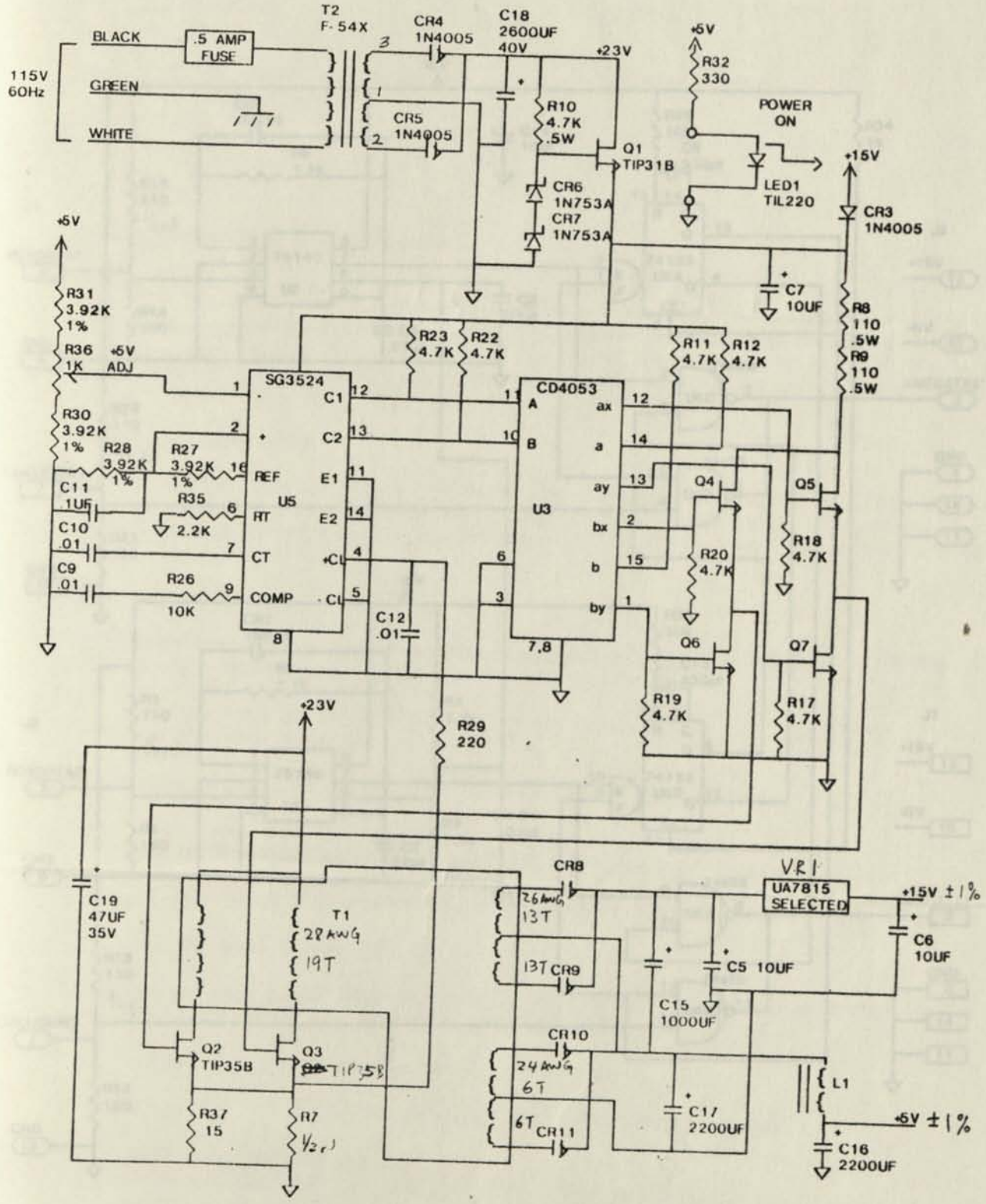
Xerox Corporation
El Segundo, California

XEROX

216411 D

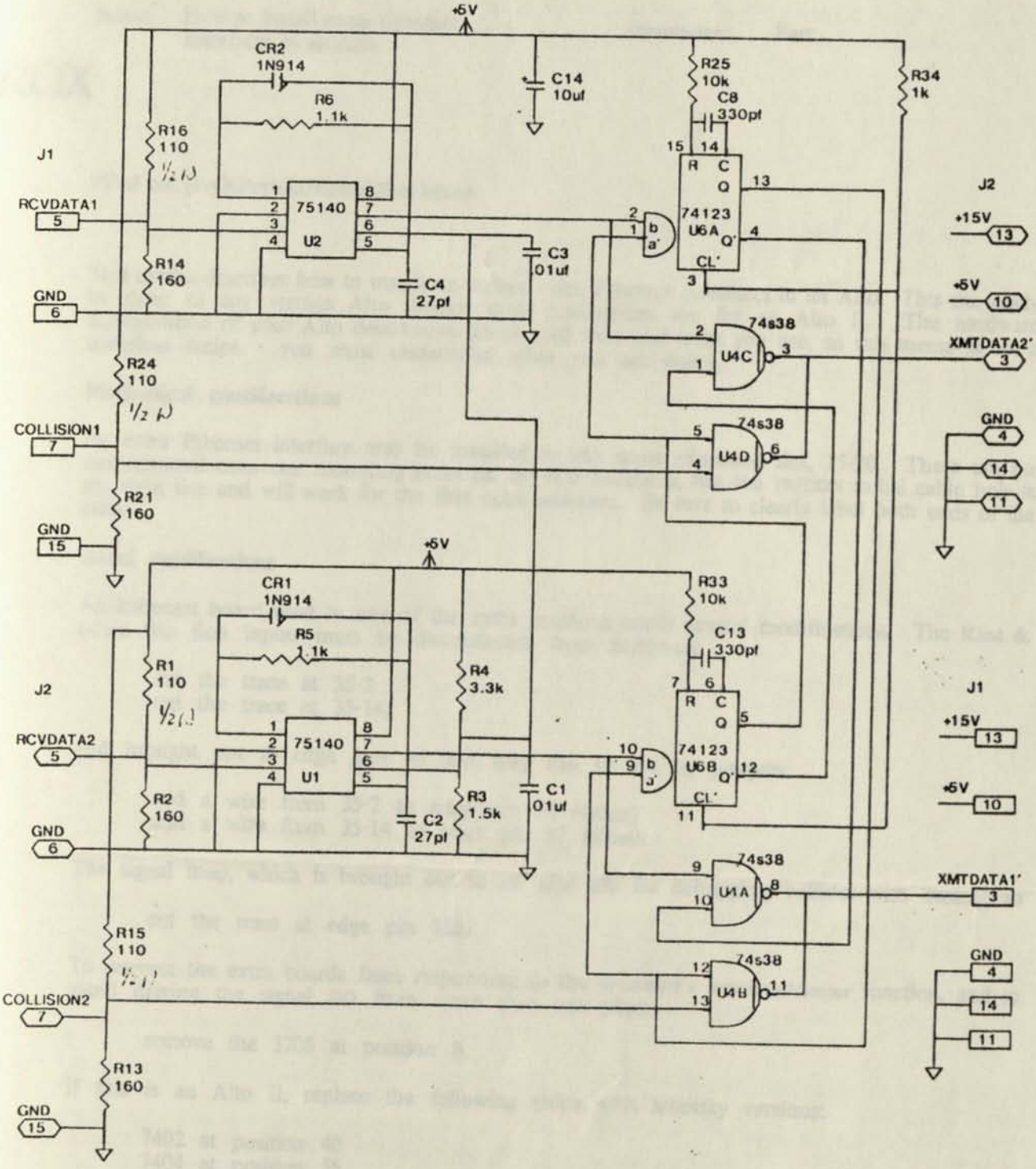
Sheet 5 of 5


STATION ← → ETHER
NOTE 2: TRANSFORMERS & INDUCTOR
T1 - 13IIP-6435-3B4
T2, T3, L1, L101
J1
J101
ETHER
IC3A
IC3B
IC3C
IC3D
IC3E
IC3F
IC1A
IC1B
IC2A
IC2B
IC2C
IC2D
IC101A
IC102A
IC102B
IC102C
POWER ON
LED 101
MJE3055
Q1
Q2
Q101
2N5771
Q103
2N4275
R123
1K
+5
+5.7
-.8V
NOTE 1: ALL RESISTORS 1/4 W 5%
IC1-3 Vcc TO +5V GND TO ⏚
IC101-102 Vcc TO +5V GND TO -.8V
IC1, 101 - SN75140P
IC2 - SN7438N
IC3 - SN7404N
IC102 - N8T14B
TRANSCEIVER SCHEMATIC
SCALE: —
DATE: 7-21-74
APPROVED BY:
DRAWN BY
REVISED 11-13-74
DRAWING NUMBER
PRINTED ON NO. 1000H CLEARPRINT



NOTE: CR8-CR11 GE114A
Q4-Q7 A5T2222

TCL INC.
REPEATER POWER SUPPLY
FILE: REPEATERPS
DATE: 7-17-78



ROX	Project		File	Designer	Rev	Date	Page

Inter-Office Memorandum

To	Alto Gateway Project	Date	July 31, 1978
From	David Boggs	Location	Palo Alto
Subject	How to install extra Ethernet Interfaces in an Alto	Organization	Parc

XEROX

Filed on: [Ivy]<Portola>ExtraEther.bravo

This memo describes how to install up to two extra Ethernet interfaces in an Alto. This can easily be done to any vintage Alto though these instructions are for an Alto II. The hardware configuration of your Alto determines which card slots and tasks you use, so this memo is not a complete recipe - you must understand what you are doing.

Mechanical considerations

An extra Ethernet interface may be installed in any spare processor slot, 15-20. There are no uncommitted connector mounting holes on the rear bulkhead, but the TRICON radial cable hole is the right size and will work for the first extra interface. Be sure to clearly label both ends of the cable.

Board modifications

An Ethernet board used in one of the extra positions needs several modifications. The ICmd & OCmd flip flop inputs must be disconnected from BUS[14-15]:

cut the trace at 35-2
cut the trace at 35-14,

and brought out to edge pins so that they can be set by jumpers:

add a wire from 35-2 to edge pin 98 (OCmd)
add a wire from 35-14 to edge pin 97 (ICmd).

The signal IBusy, which is brought out to an edge pin for debugging, collides with TaskA', so

cut the trace at edge pin 113.

To prevent the extra boards from responding to the emulator's ReadSerialNumber function, and to avoid driving the signal SIO from more than one place,

remove the 3205 at position 9.

If this is an Alto II, replace the following chips with schottky versions:

7402 at position 40
7404 at position 55
74157 at position 48
7438s at positions 14, 15, 24, 25, 26.

A modified board will work in a normal Ethernet slot if you replace the 3205 at position 9 and jumper pin 14-98 to 14-95 and pin 14-97 to 14-94 on the backplane.

Backplane modifications

An Ethernet board needs some signals which are not present on the standard processor bus slots. These are available on the corresponding pins of slot 14, the standard Ethernet.

SysClk'	12
AuSysClk	72
←KData'	111
EmAct'	99
SWakMRT'	68

In addition, two BUS bits must be connected to the Cmd flip flops, and a task must be assigned by connecting the board's Active and Wakeup signals. These are discussed below.

Note that SReset' and EStop are not wired on extra interfaces. SReset is the signal which boots the machine, and it is sufficient for the standard Ethernet to yank on it; besides, SIO decoding on the extra boards is disabled. EStop is the signal which stops the clocks for one cycle to fix a long path in the interface. Installing Schottky chips in the path makes it unnecessary to do this.

Host addresses

The host address logic in an extra Ethernet interface is disabled by removing the 3205, so the SIO instruction returns the address set by the jumpers on the standard Ethernet interface in slot 14. Host jumpers on the extra slots are not required.

Tasks, SIO bits, and Page 1 locations

The choice of task for an extra Ethernet interface is invisible to the emulator level program. An active interface consumes about 15% of an Alto, which is low enough that any of the four uncommitted tasks available on the backplane will work. Pick one of them and wire its wakeup and active pins on the control board (slot 11) to EtherWakeup' (pin 103) and EtherActive' (pin 100) on the extra Ethernet board. The table below gives the pin numbers on the control board for the uncommitted tasks.

Task	Wakeup'	Active'
1	113	119
2	58	52
5	60	102
6	104	101

The microcode for the extra interfaces use page 1 locations 630-640B and 642-652B in the same way that the standard Ethernet uses 600-610B. The extra interfaces may be assigned different host addresses than the standard one by putting different numbers in 640B and 652B, but as mentioned above, SIO returns the address set on the backplane of the standard interface so you must invent a new way to get the additional addresses. Unless there is a compelling reason, I recommend that additional interfaces use the same host address as the standard one.

The emulator task signals an Ether task by placing a value on BUS and executing the SIO emulator function. Each Ethernet interface checks two BUS bits during an SIO and wakes up its task if either bit is one. The task then performs some action which ends up modifying its page 1 locations. Thus the software must know the correspondence between SIO bits and page 1 locations. I recommend the following correspondence:

SIO bits	page 1	
14 & 15	600-610B	(standard Ethernet interface)
12 & 13	630-640B	
10 & 11	642-651B	

where the MSB of the pair sets the ICmd FF by being wired to pin 97, and the LSB sets the OCmd FF by being wired to pin 98. The MESA and BCPL PUP packages assume this; if you do it differently you forfeit compatibility. BUS[0-15] are on pins 80-95.

Microcode

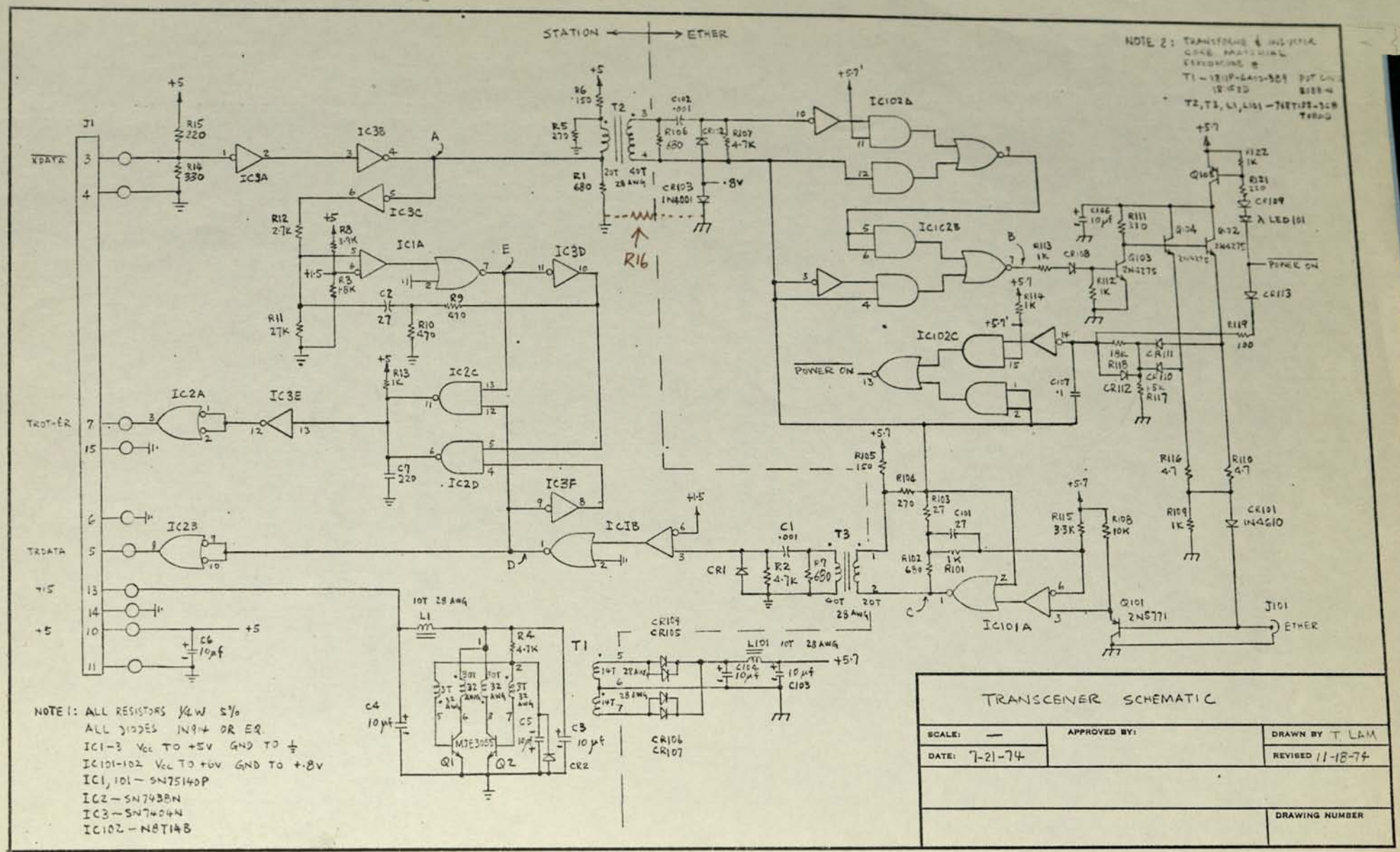
Files ExtraEther1.mu and ExtraEther2.mu contain copies of the Ether microcode which use page 1 locations 630-640B and 642-652B respectively. These files do not define task numbers or R-registers to be used. File ExtraEther.mu is an example of how to do this, assigning tasks 2 and 3, and registers 14-17B, and adding enough other definitions to make a stand-alone ram image for two extra Ethernets. These files are stored in [Ivy]<Portola>GatewayMc.dm

Note that the registers must be in the first group of 32 since the Ether hardware can't be ram-related (function and bus sources collide with the ram). This is a problem for MESA, since only one R-register is available. Another one can be freed by rewriting the memory refresh task to eliminate its use of ClockTemp. To get the next two, the cursor must be sacrificed. This involves deleting all references to CurX and CurData from the MRT, Cursor, and DVT tasks.

Revision History

July 20, 1978

First release.



NOTE 1: ALL RESISTORS ¼W 5%
ALL DIODES IN914 OR EQ.
IC1-3 V_{cc} TO +5V GND TO ⏚
IC101-102 V_{cc} TO +6V GND TO +.8V
IC1, 101 — SN75140P
IC2 — SN7438N
IC3 — SN7404N
IC102 — N8T14B

TRANSCEIVER SCHEMATIC		
SCALE: —	APPROVED BY:	DRAWN BY T. LAM
DATE: 7-21-74		REVISED 11-18-74
		DRAWING NUMBER