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.

ETHERNET

An Ethernet is the principal means of communications between an Alto and the outside world. The object was to design a communication system which could grow smoothly to accommodate several buildings full of personal computers and the facilities needed for their support. The Ethernet is a broadcast. multi-drop. packet-switching. bit-serial. digital communications network: it connects up to 256 nodes. separated by as much as 1 kilometer. with a 2.94 megbits/sec. channel. Control of the Ethernet is distributed among the communication computers to eliminate the reliability problems of an active central controller. to avoid a bottleneck in a system. rich in parallelism. and to reduce the fixed costs which make small systems uneconomical.

The Ethernet is intended to be an efficient. low-level packet transport mechanism which gives its best efforts to delivering packets. but it is not error free. Even when transmitted without source-detected interference. a packet may not reach its destination without error: thus. packets are delivered only with high probability. Stations requiring a residual error rate lower than that provided by this bare packet transport mechanism must follow mutually agreed upon packet protocols.

Alto Ethernets come in three pieces: the transceiver. the interface. and the microcode. The transceiver is a small device which taps into the passing Ether. inserting and extracting bits under the control of the interface while disturbing the Ether as little as possible. The same device is used to connect all types of Ethernet interfaces to the Ether. so the transceiver design is not specific to the Alto. and will not be described here.

When a program wishes to send a packet, it must first turn off the receiver if it is on. If the receiver is actively copying a packet into memory. the transmitter should wait for the receiver to finish (a maximum of about 1.5 msec. assuming 250-300 word packets). The program can tell whether the receiver is actively transferring or idle by zeroing the first word of the input buffer before starting the receiver. When the program wants to start the transmitter. it checks the first word of the input buffer: if it is still zero. input has not yet begun and the interface may be reset and the transmitter started with a high probability of not missing an incoming packet. There is still a small window between testing the word and starting the transmitter when a packet can arrive and be missed. but paragraph two warned that the Ethernet is not error free anyway. so missing a few more packets should be harmless.

The first word of all Ethernet packets must contain the address to which the packet is destined in the left byte. and the address of the sender (or 'source') in the right byte. Reveivers examine at least the destination byte. and in some cases (not in Altos) the source byte to determine whether to copy the message into memory as it passes by. Address zero has special meaning to the Ethernet. Packets with destination zero are broadcast packets. and all receivers will receive them. If a program wishes to receive all packets on the Ether regardless of address (useful for debugging and diagnostic programs). it should use zero. A host which does this is said to be promiscuous. Address 377 (octal) is reserved for Ethernet booting. Address 376 (octal) is reserved as the destination for diagnostic messages.

ETHERNET HARDWARE

The Ethernet hardware consists of a FIFO buffer, an output shift register and phase encoder, a clock recovery circuit, an input shift register, a CRC register, and one microcode task. Packets on the Ether are phase encoded and transmitter synchronous: it is the responsibility of the receiver to decide where a packet begins (and thus establish the phase of the data clock), separate the clock from the data, and deserialize the incoming bit stream. The purpose of the write register is to synchronize data transfers between the input shift register whose clock is derived from the incoming data, and the FIFO which is synchronous to the processor system clock. The large FIFO is necessary because the Ethernet task has relatively low priority, and the worst case latency from request to task wakeup is on the order of 20 microseconds. The phase encoder uses the system clock (one Ethernet bit time is two clock periods).

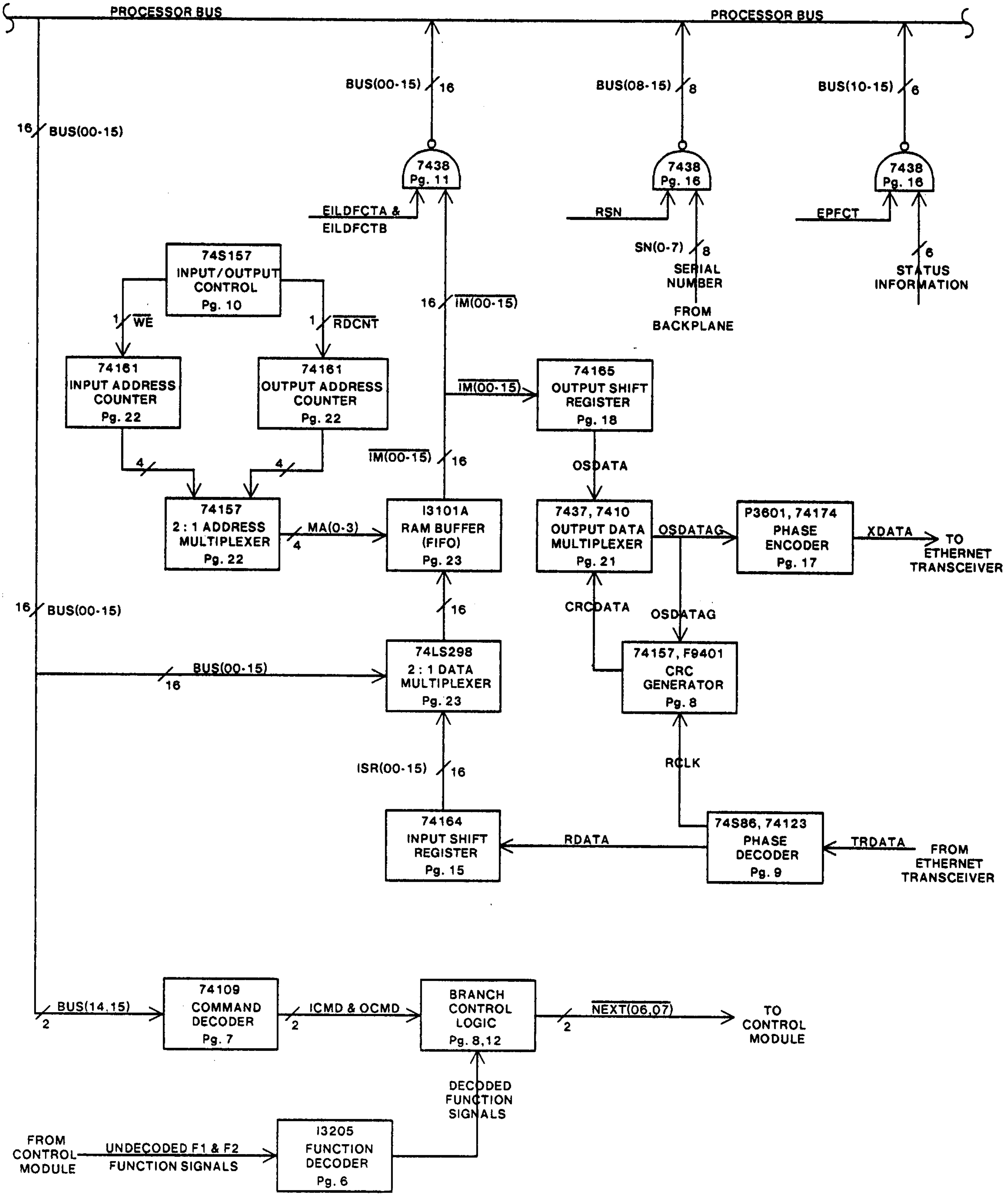
Included in the clock recovery section is a one-shot which is retriggered by each level transition of a passing packet. This detects the envelope of a packet and is called its 'carrier'. Ethernet phase encoders mark the beginning of a packet by prefixing a single 1 bit, called the sync bit, to the front of all transmissions. The leading edge of the sync bit of a packet will trigger the carrier one-shot of a listening receiver and establish the receiver clock phase. The sync bit is clocked into the input shift register and recirculated every 16 bit times thereafter to mark the presence of a complete word in the register. If carrier drops without the sync bit at the end of the register, the transmission was incomplete, and is flagged in the hardware status bits. When the shift register if full, the word is transferred to the write register where it sits until the FIFO control has synchronized its presence and there is room to accept it. If the shift register fills up again before the word has been transferred from the write register to the FIFO, data has been lost and the input data late flip flop is set.

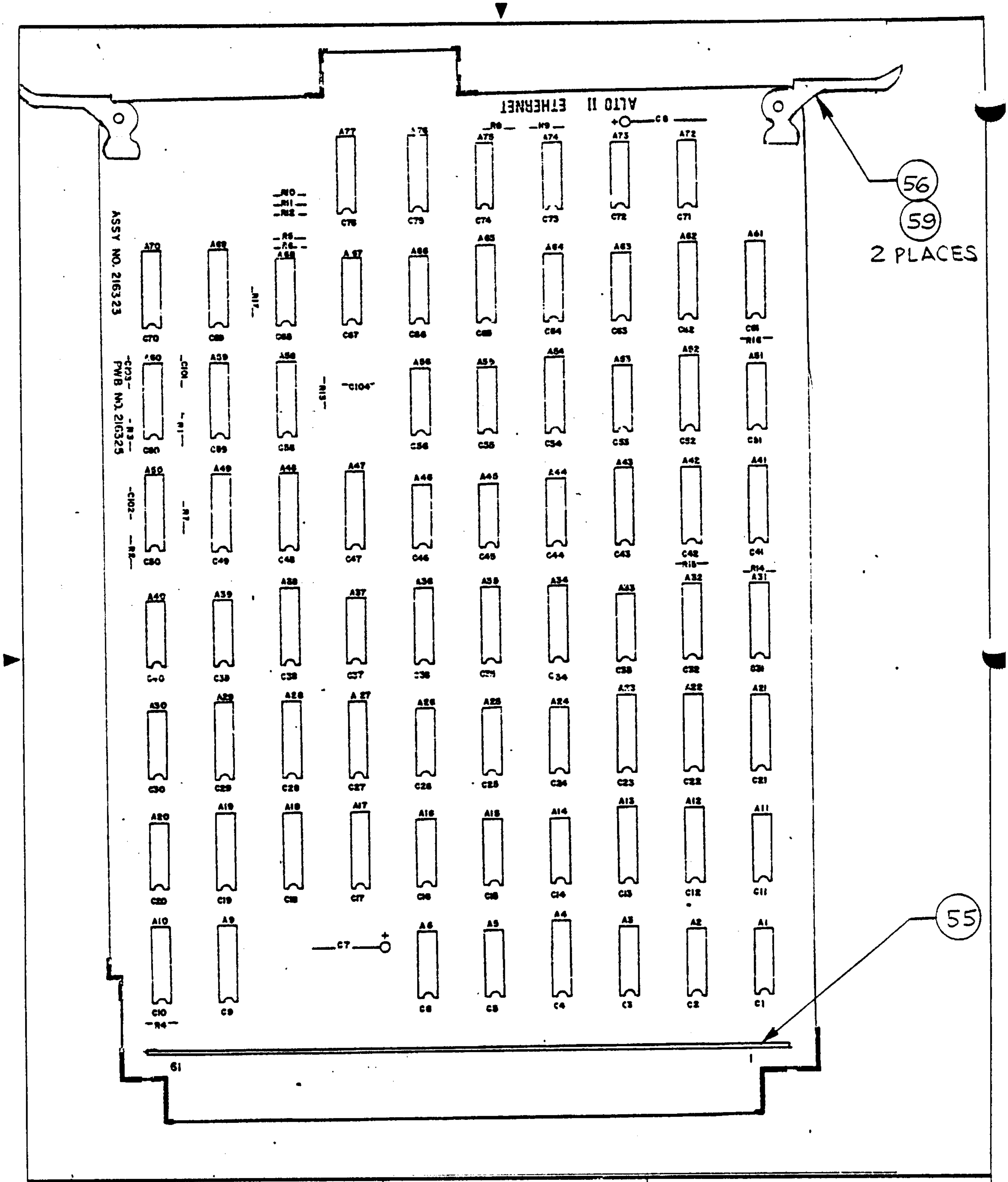
Ethernet transmitters accumulate a 16 bit cyclic redundancy checksum on the data as it is serialized, and append it to an outgoing packet after the last data word. As a receiver deserializes an incoming packet it recomputes the checksum over the data plus the appended CRC word. If the resulting receiver checksum is non-zero, the received packet is assumed to be in error; and the condition is flagged in the hardware status byte.

The phase encoder is started when the microcode has decremented the countdown to zero, there is no carrier present, and either the FIFO is full, or if the message is less than 16 words long, all of it has been transferred to the FIFO. The phase encoder will not start up while there is carrier present. This means that collisions can only happen because of delay in sensing carrier between widely spaced transmitters. Collisions are detected at the transceiver by comparing the data the interface is supplying to the data being received off the Ether. If the two are not identical, a signal is returned to the interface which sets the collision flip flop causing a wakeup request to the microcode which resets the interface.

The interface and the transceiver are connected together by three twisted pairs for signals plus supply voltages and ground supplied from the interface. The signals are (1) transmitted data to the transceiver, (2) received data from the transceiver, and (3) the collision signal from the transceiver indicating interference.

ETHERNET MODULE




ALTO II ETHERNET
ASSY NO. 216323
PWB NO. 216325
A77 A76 A75 A74 A73 A72
R8 R9 C8
R10 R11 R12 R5 R6
C76 C75 C74 C73 C72 C71
A70 A69 A68 A67 A66 A65 A64 A63 A62 A61
R17
C70 C69 C68 C67 C66 C65 C64 C63 C62 C61
R16
A60 A59 A58 A56 A55 A54 A53 A52 A51
C103 C101 R13 C104 R3 R1
C60 C59 C58 C56 C55 C54 C53 C52 C51
A50 A49 A48 A47 A46 A45 A44 A43 A42 A41
C102 R7 R2
C50 C49 C48 C47 C46 C45 C44 C43 C42 C41
R15 R14
A40 A39 A38 A37 A36 A35 A34 A33 A32 A31
C40 C39 C38 C37 C36 C35 C34 C33 C32 C31
A30 A29 A28 A27 A26 A25 A24 A23 A22 A21
C30 C29 C28 C27 C26 C25 C24 C23 C22 C21
A20 A19 A18 A17 A16 A15 A14 A13 A12 A11
C20 C19 C18 C17 C16 C15 C14 C13 C12 C11
A10 A9 C7 + A6 A5 A4 A3 A2 A1
C10 C9 C6 C5 C4 C3 C2 C1
R4
61
1
56
59
2 PLACES
55

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.

Title

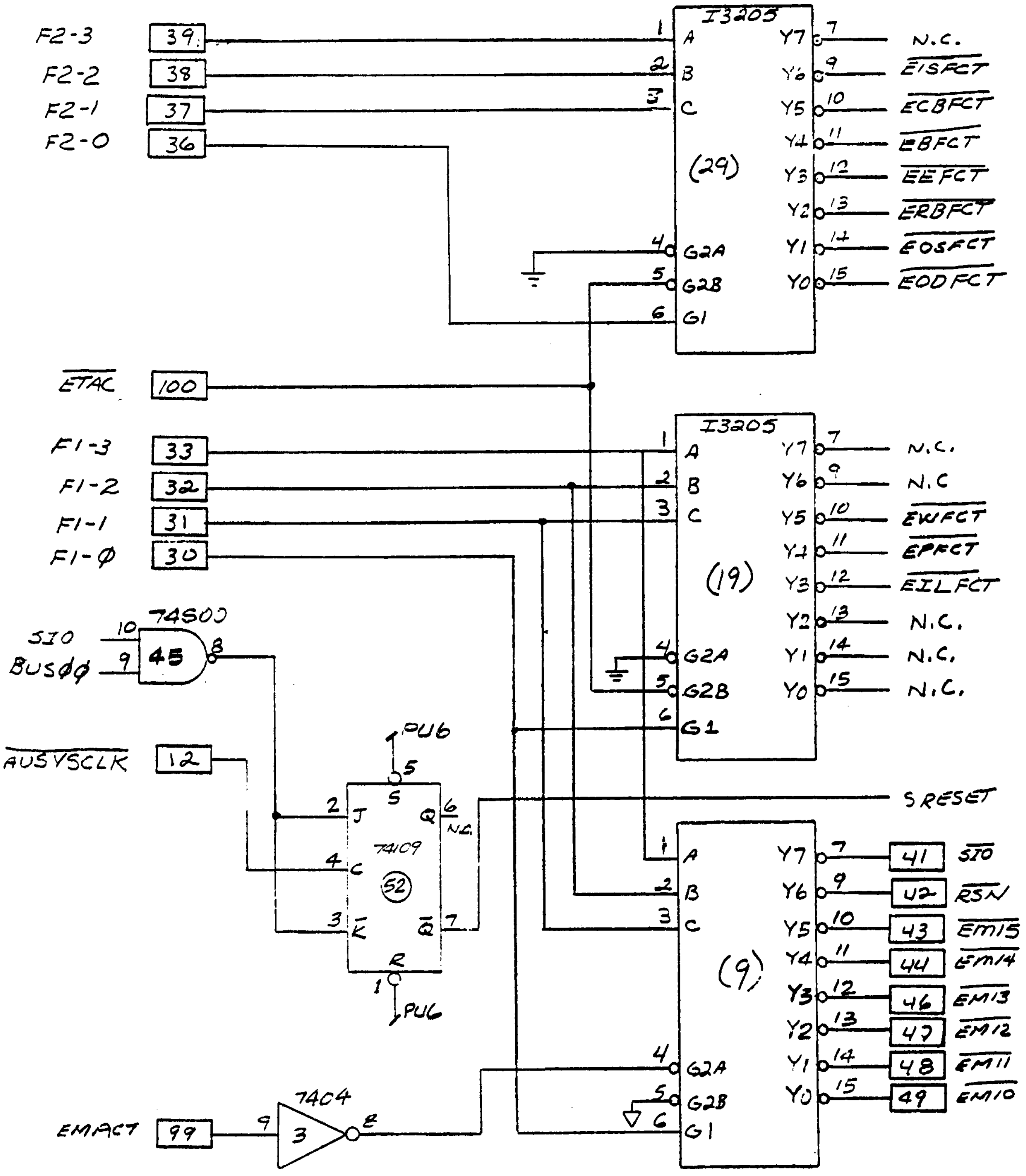
ASSEMBLY, PRINTED WIRING- ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323	G
Sheet 3 Of	

1586(3/73)



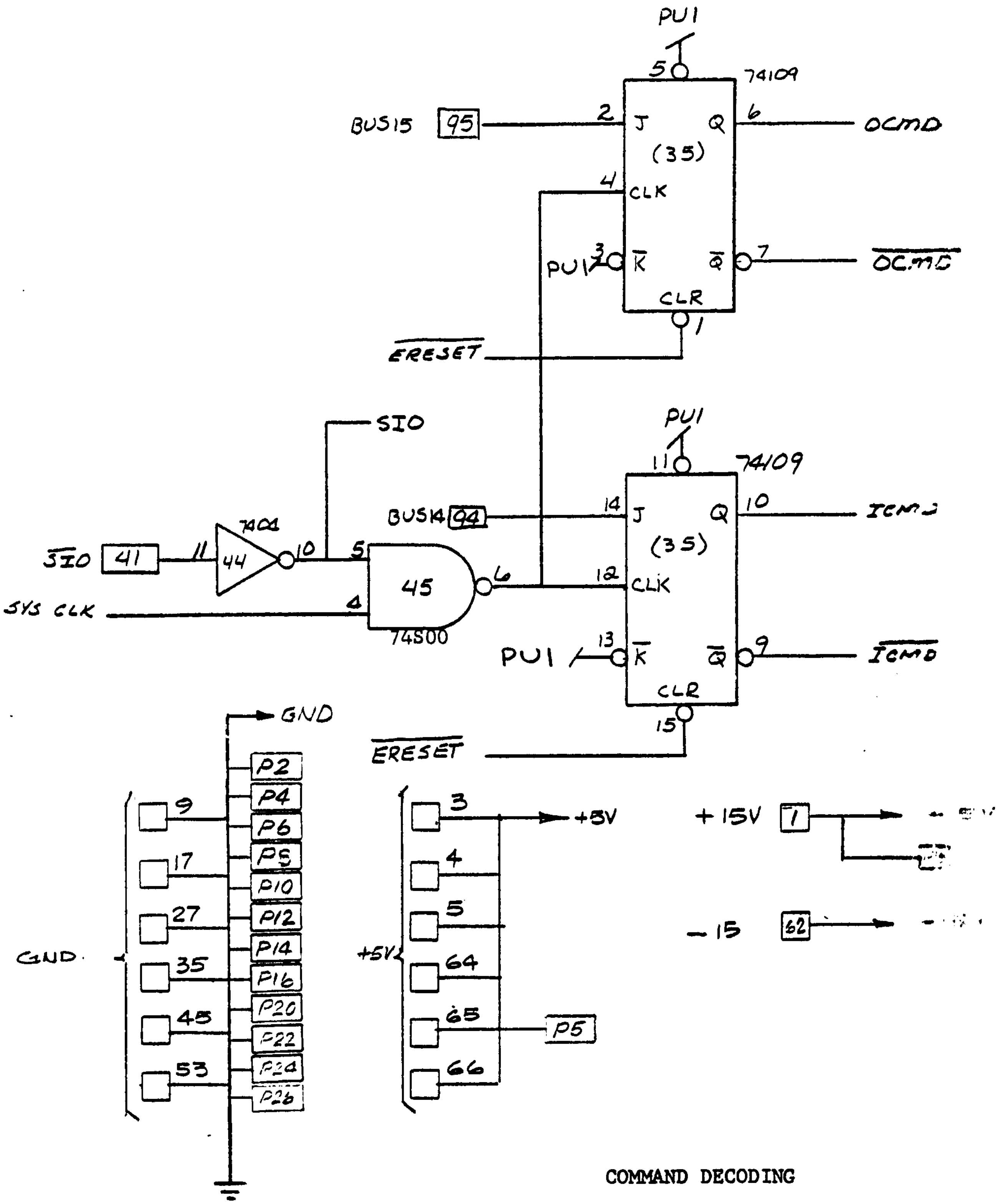
COMMAND DECODING

Title ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323	G
Sheet 6 Of	



COMMAND DECODING

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.

Title ALTO II

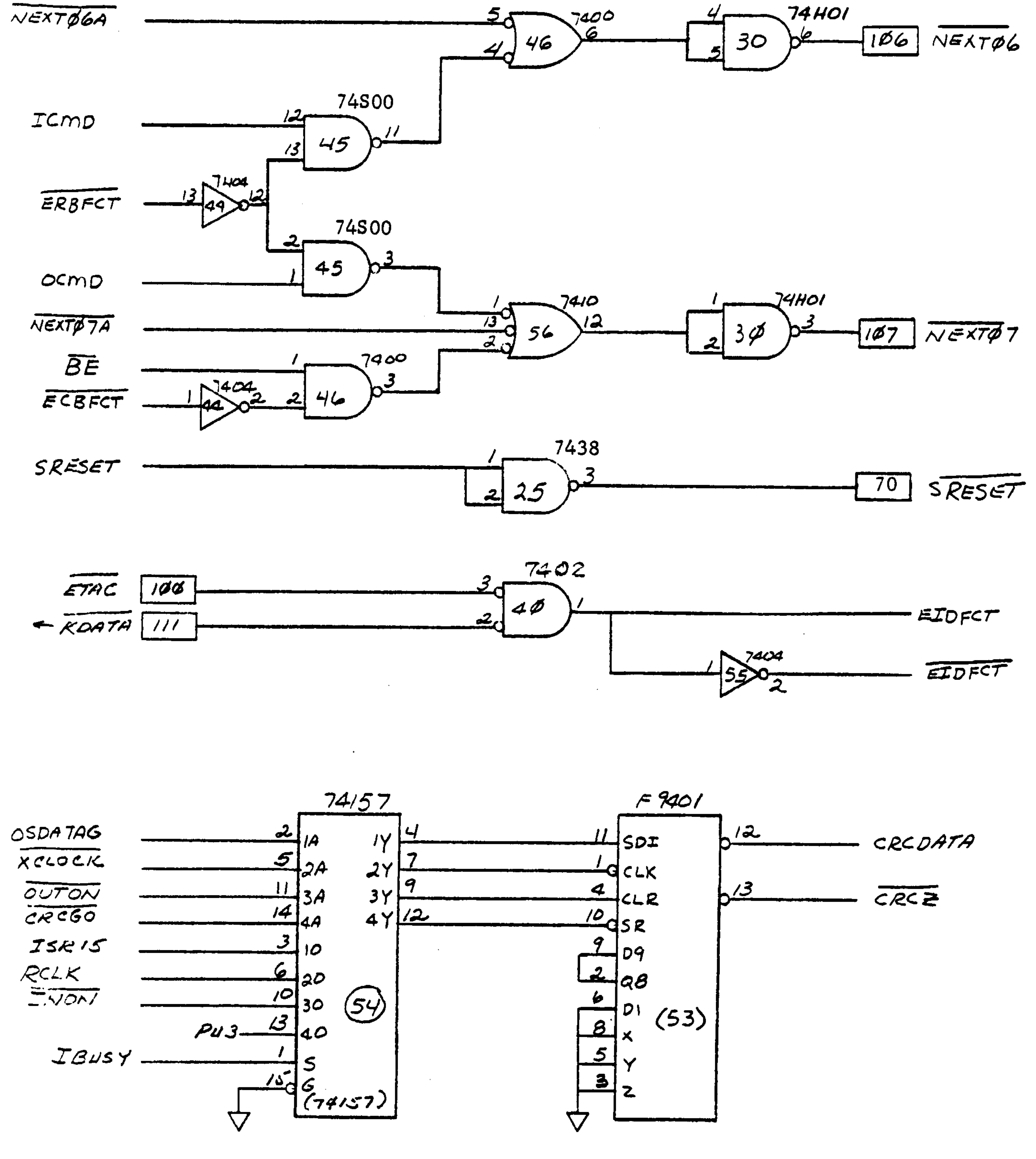
ASSEMBLY,PRINTED WIRING

ETHERNET

Xerox Corporation
El Segundo, California

216323 G

Sheet 7 Of



BRANCH LOGIC

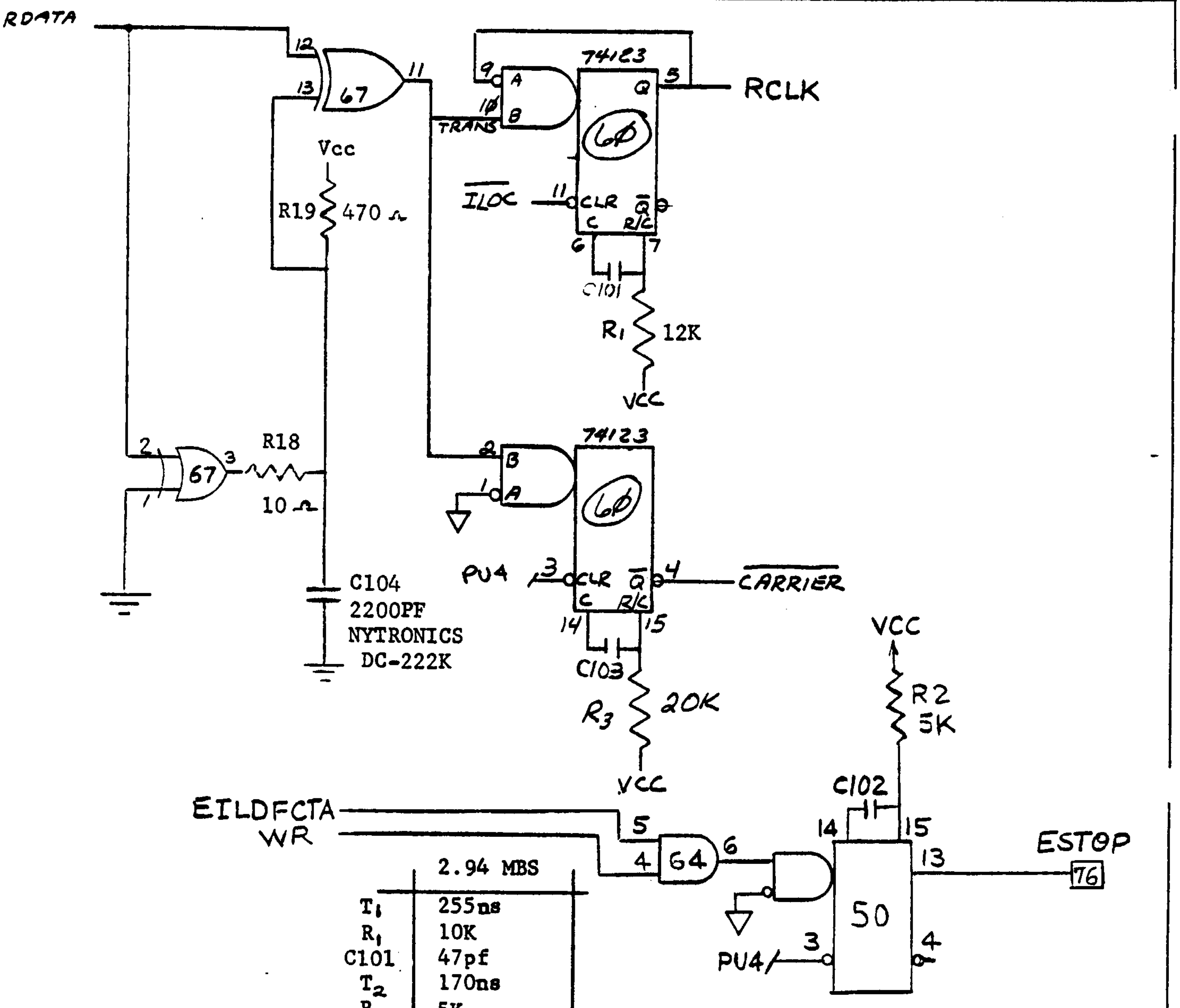
Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323 | G

Sheet 8 Of



	2.94 MBS
T_1	255ns
R_1	10K
C101	47pf
T_2	170ns
R_2	5K
C102	68pf
T_3	425ns
R_3	20K
C103	39pf
C104	2200Pf
R18	10 Ω
R19	470 Ω

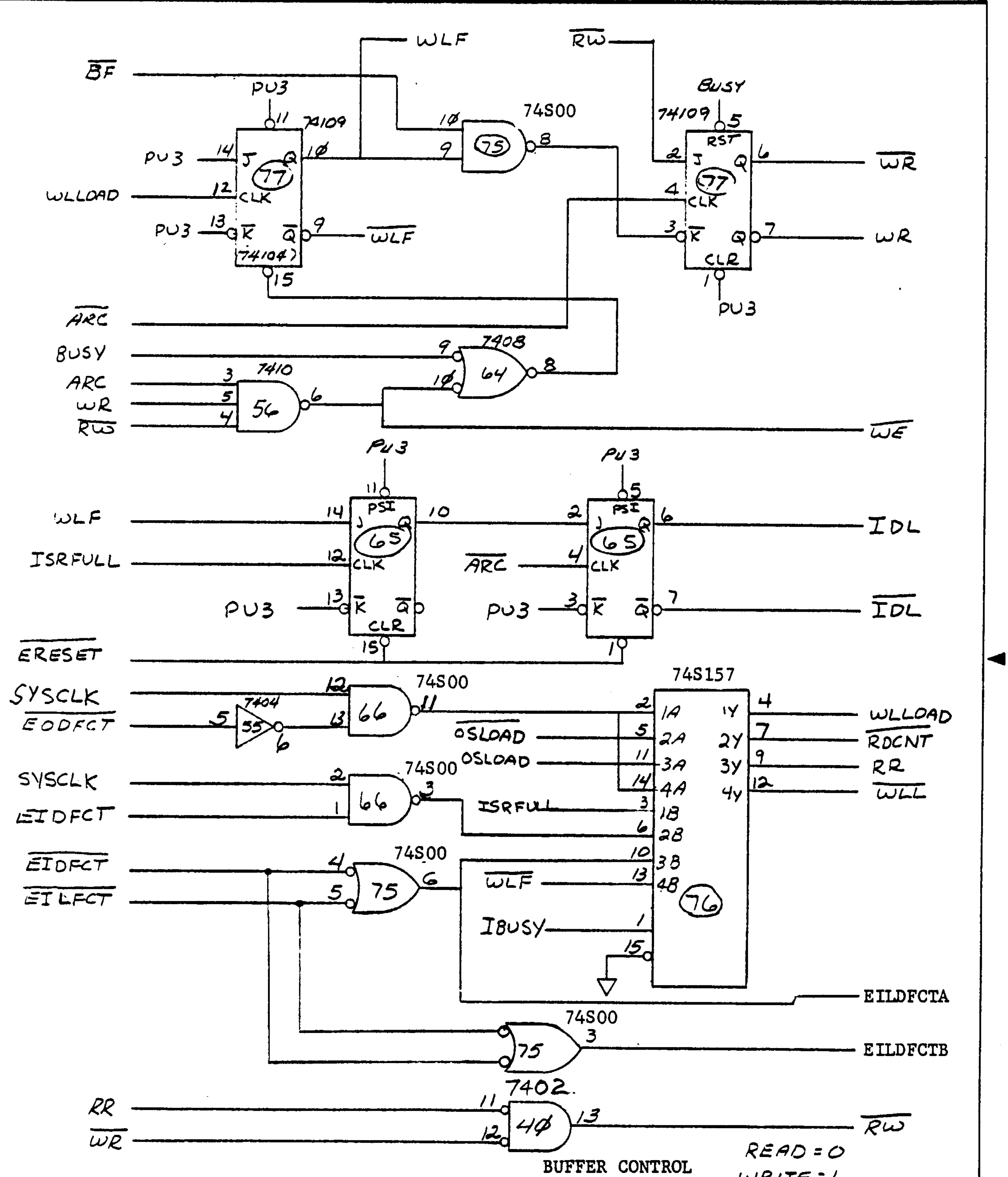
Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323 G

Sheet 9 Of



Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

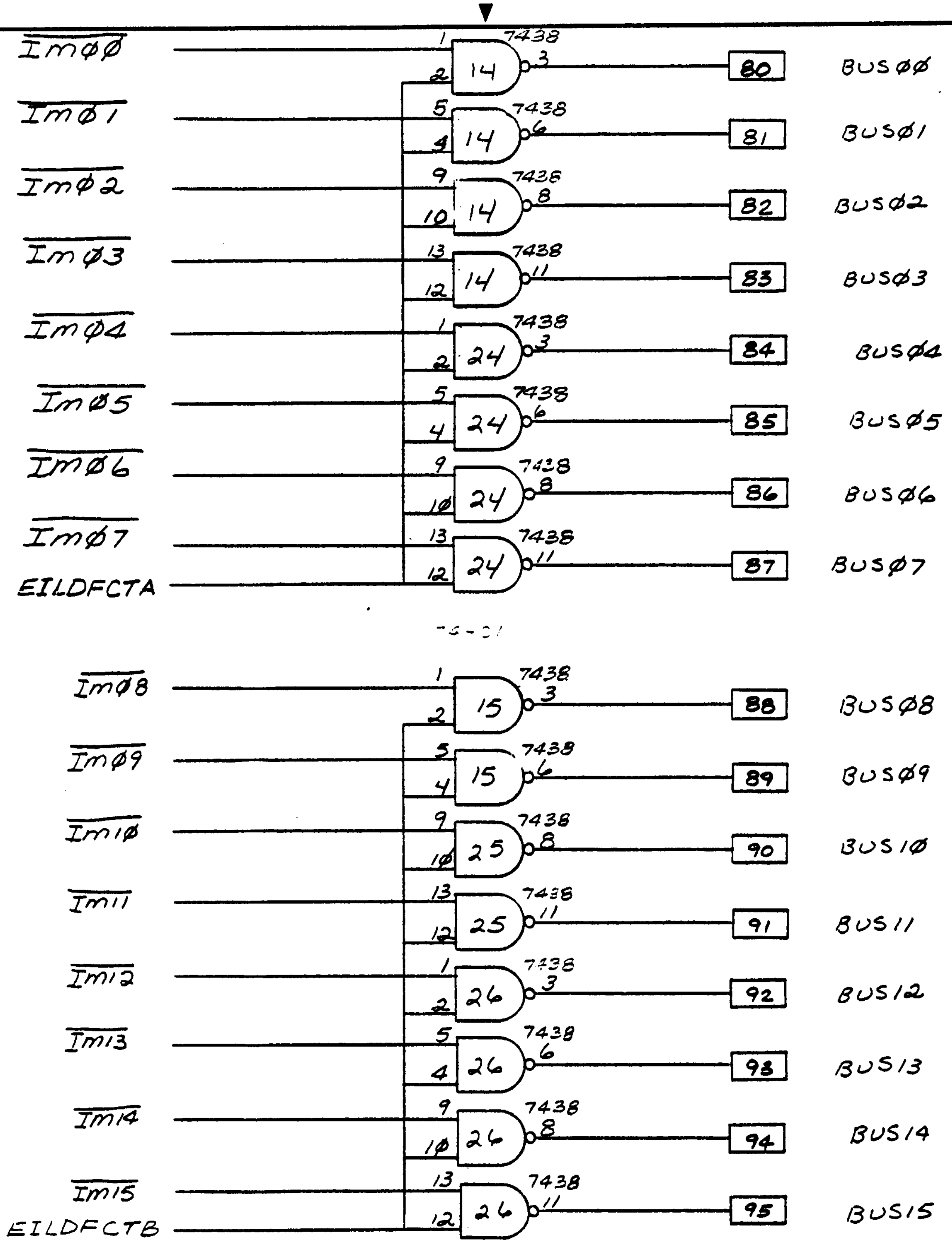
Xerox Corporation
El Segundo, California

XEROX

21632?	G

Sheet 10 OF

1586(3/73)



PROCESSOR BUS DRIVERS

Title ALTO II

ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

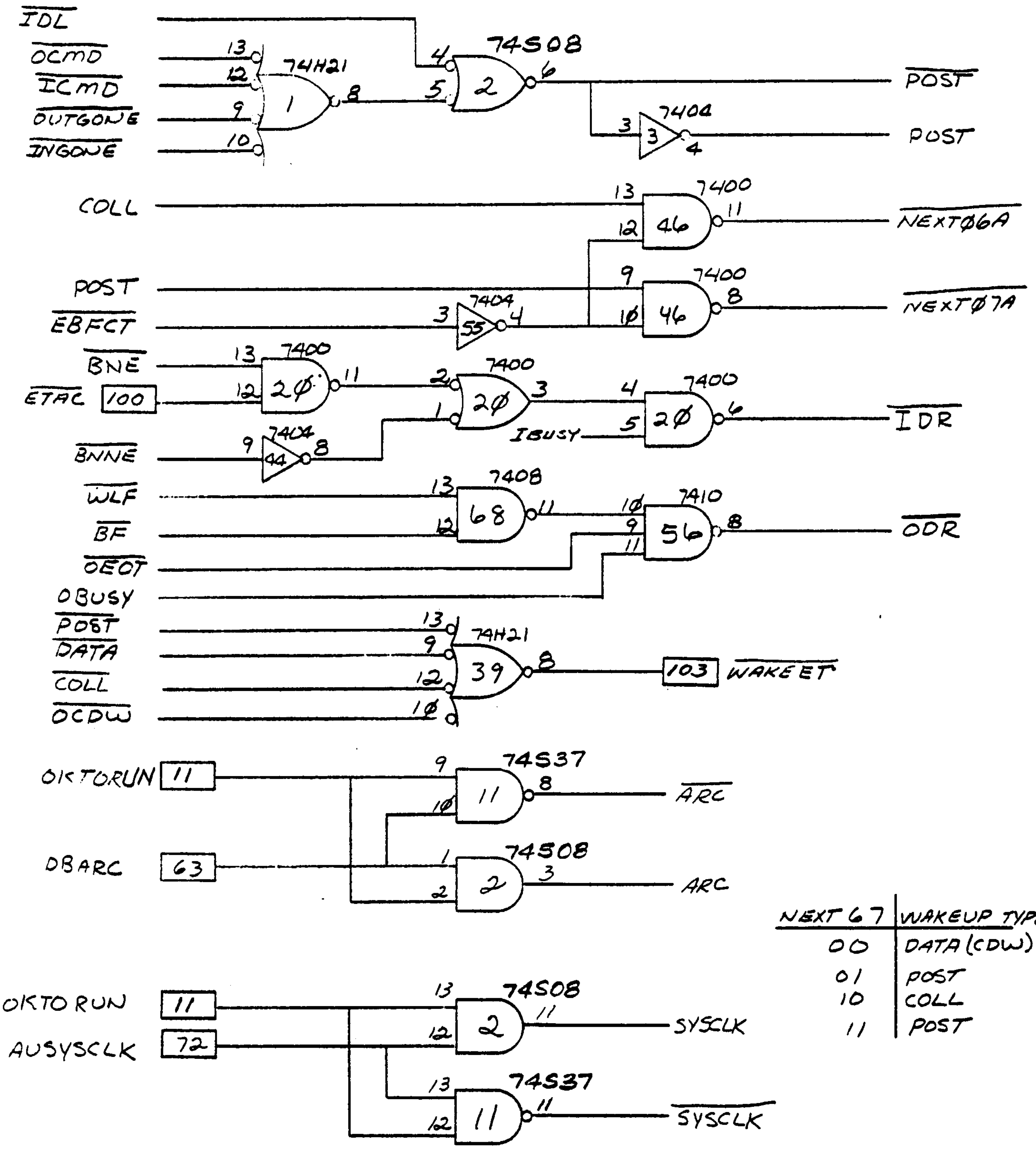
XEROX

216323

G

Sheet 11 Of

1586(3/73)



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

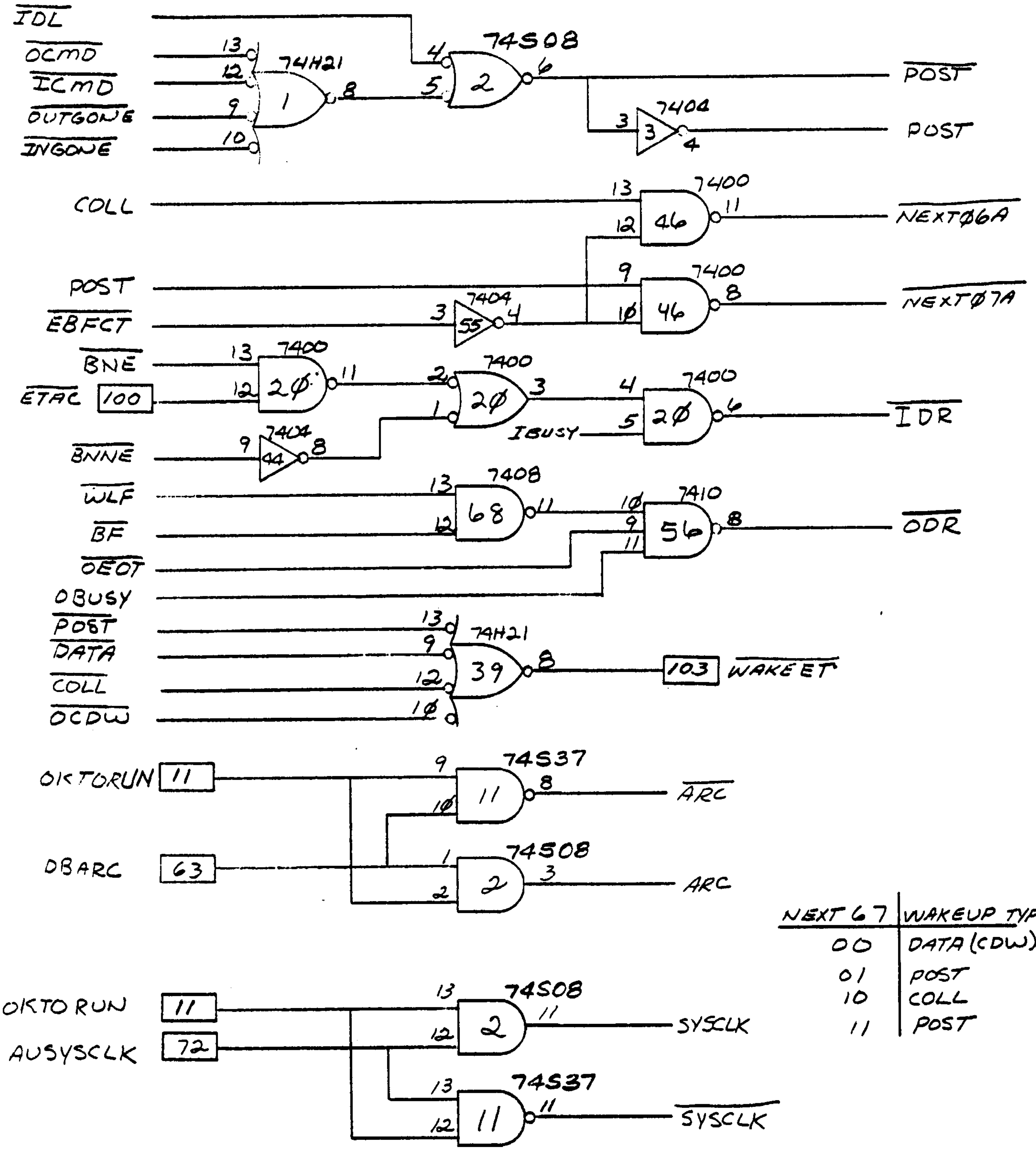
ALTO INTERFACE

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323	G
Sheet 12 of	



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

ALTO INTERFACE

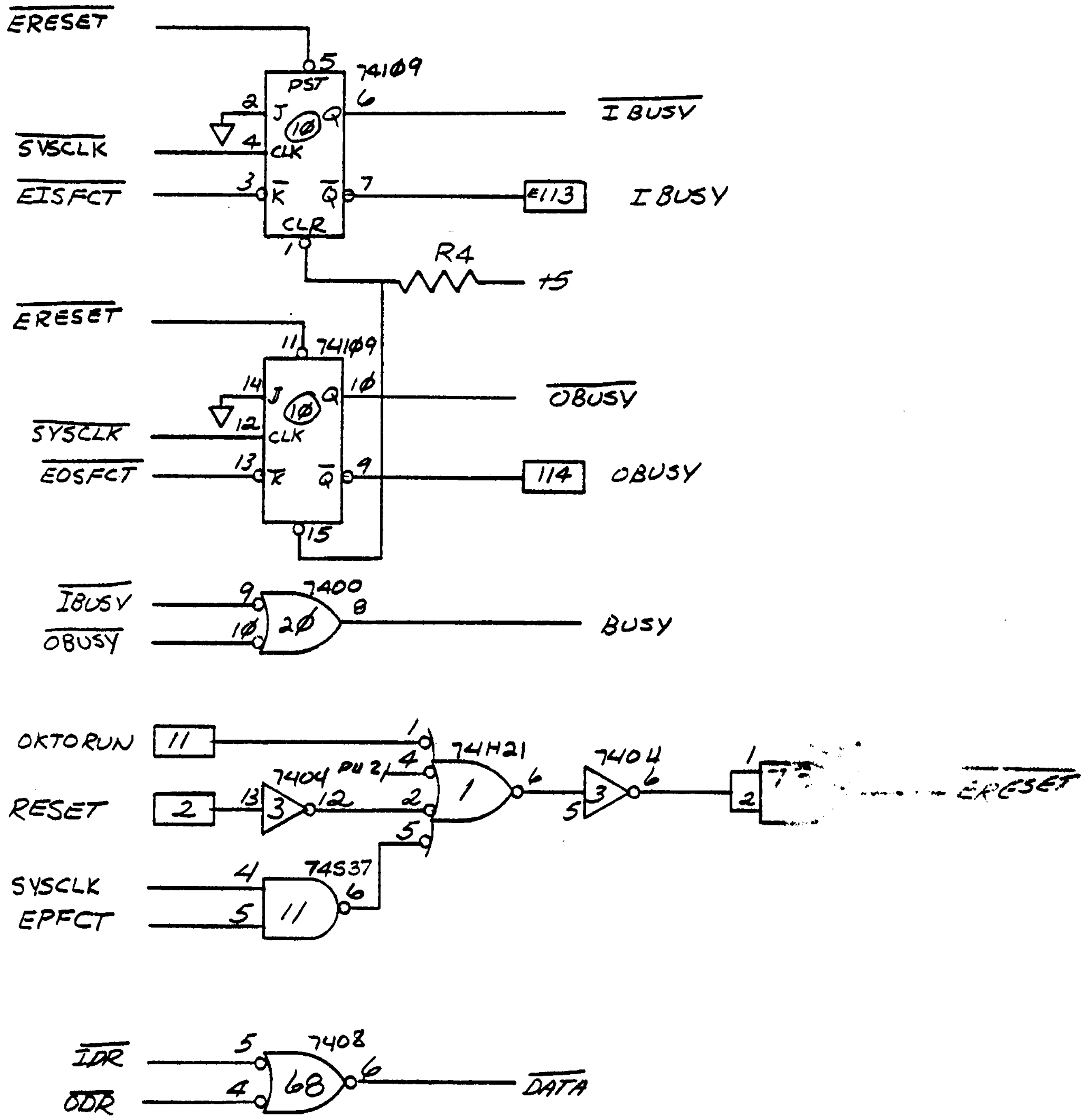
Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323 G

Sheet 12 of



ALTO INTERFACE

Title ALTO II

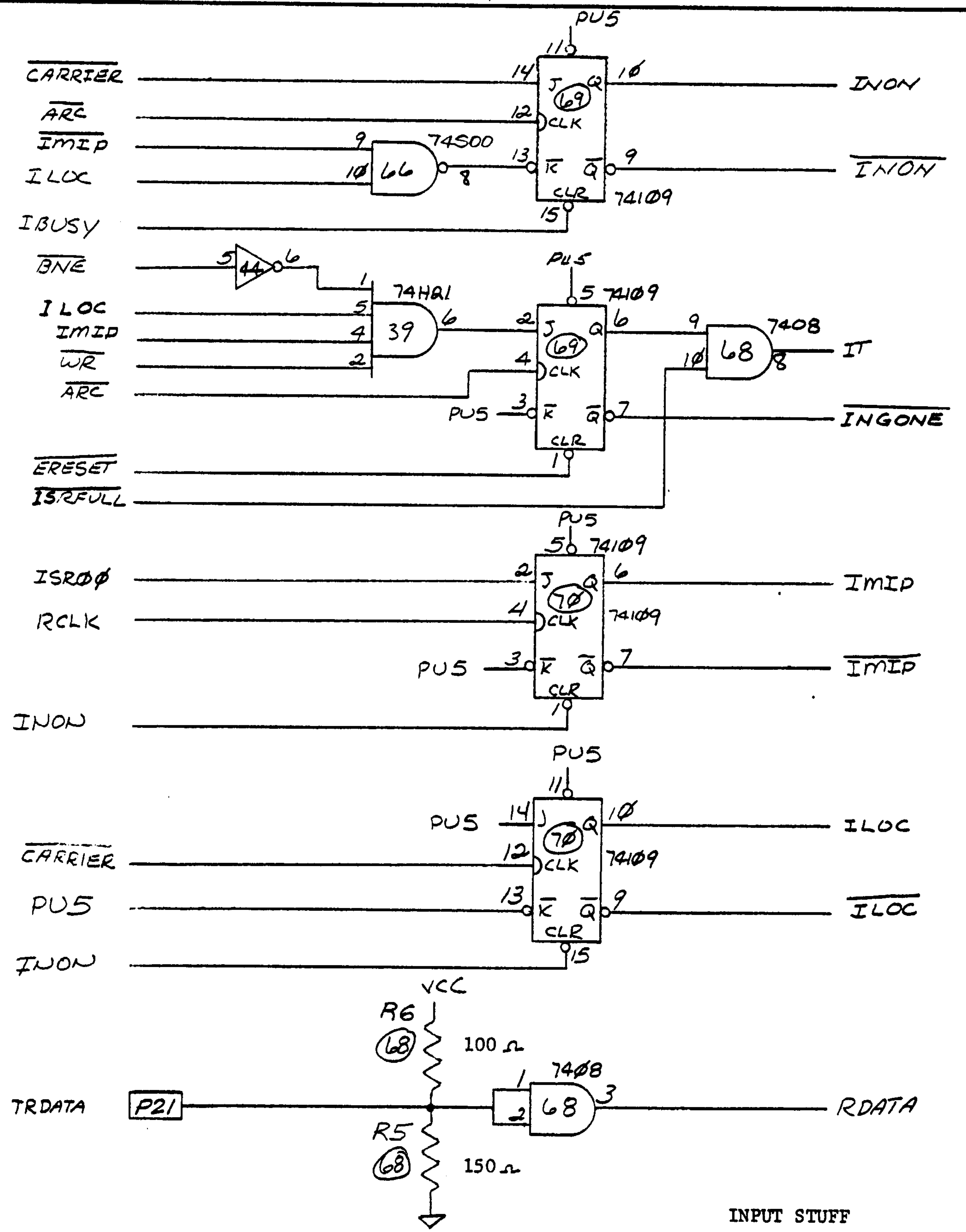
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox [illegible]
El Seg[illegible]

XEROX

3

Sheet



INPUT STUFF

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.

Title

ALTO II

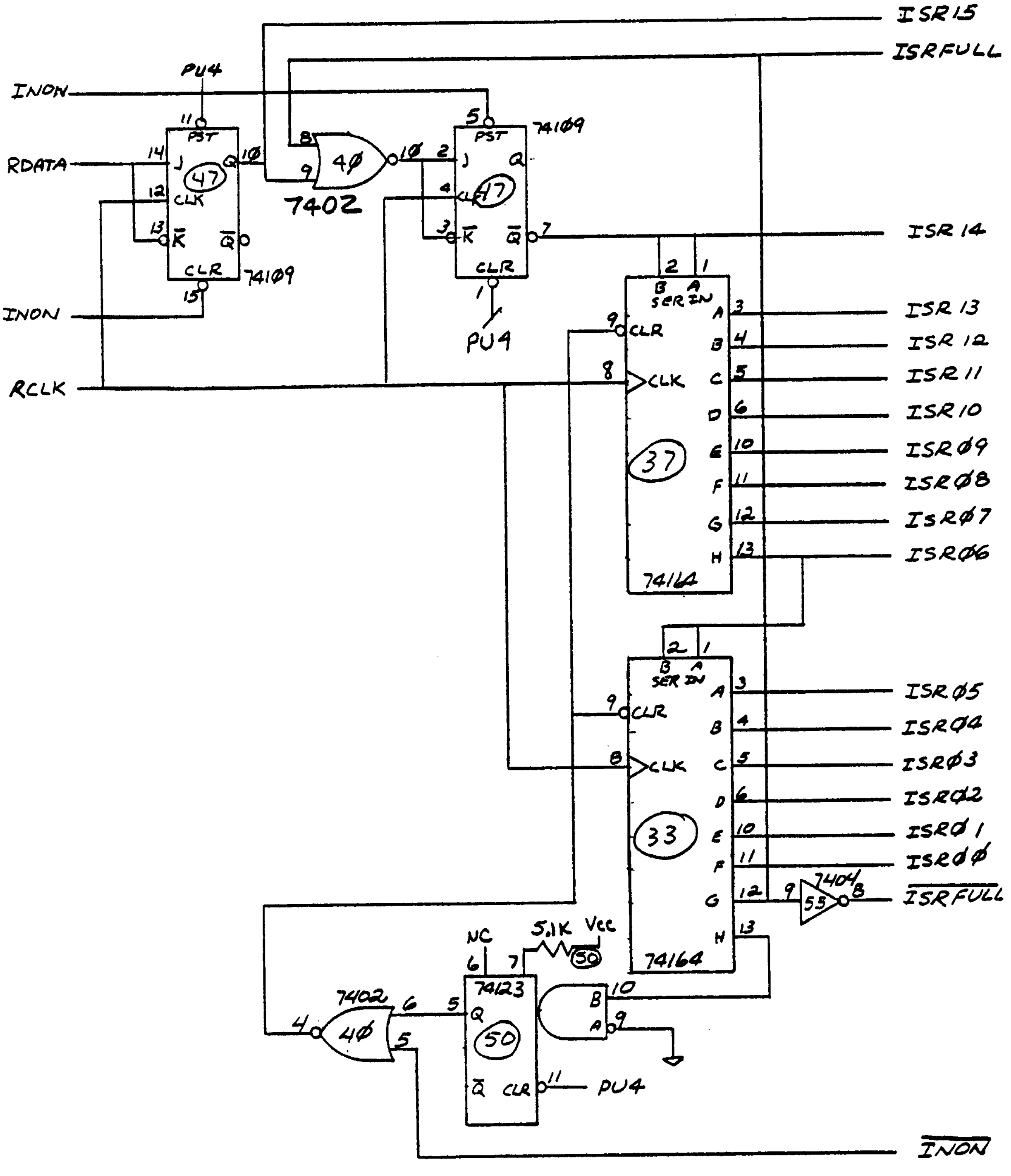
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323 G

Sheet 14 of



INPUT SHIFT REGISTER

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

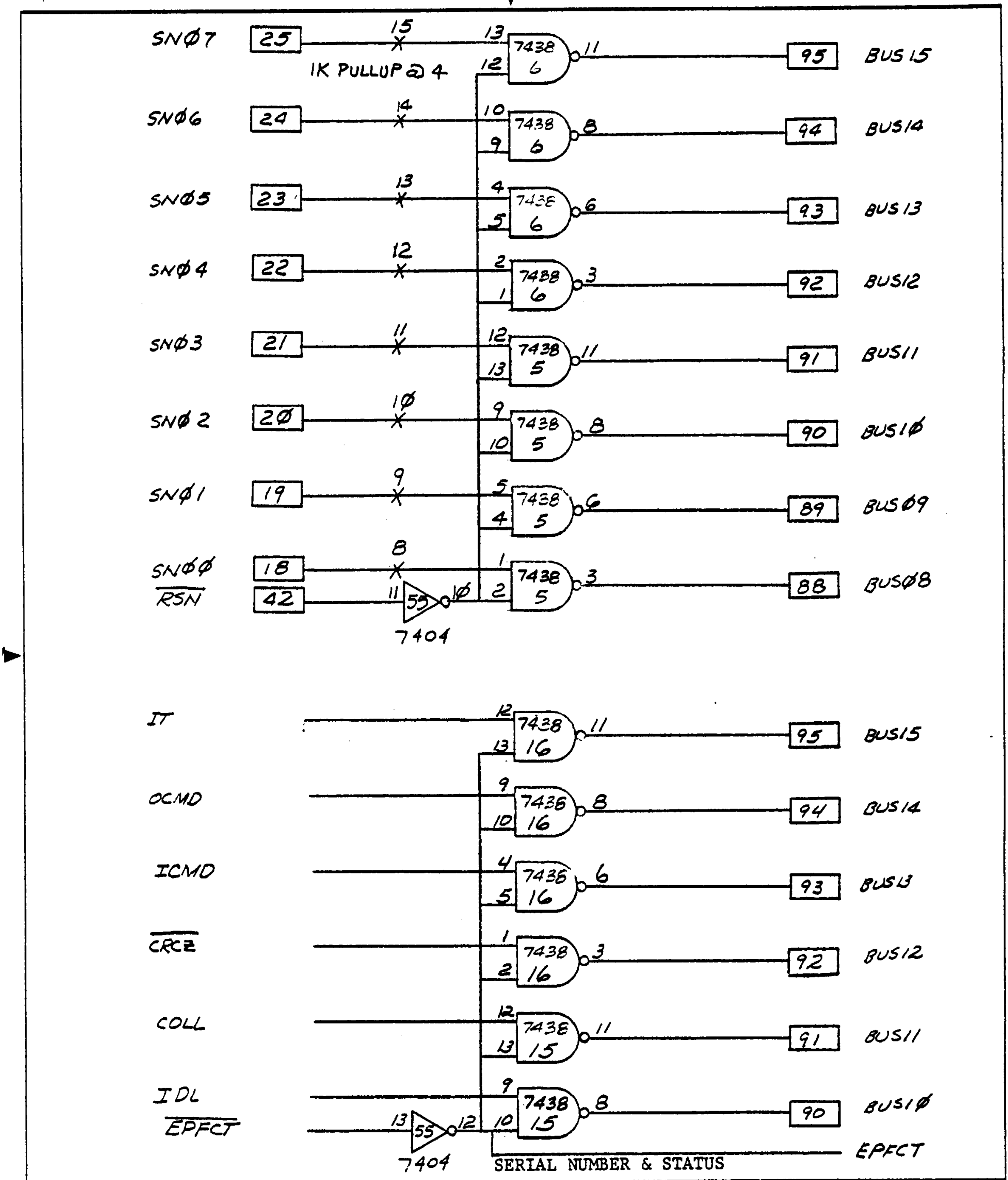
Xerox Corporation
El Segundo, California

XEROX

216323

G

Sheet 15 Of



SERIAL NUMBER & STATUS

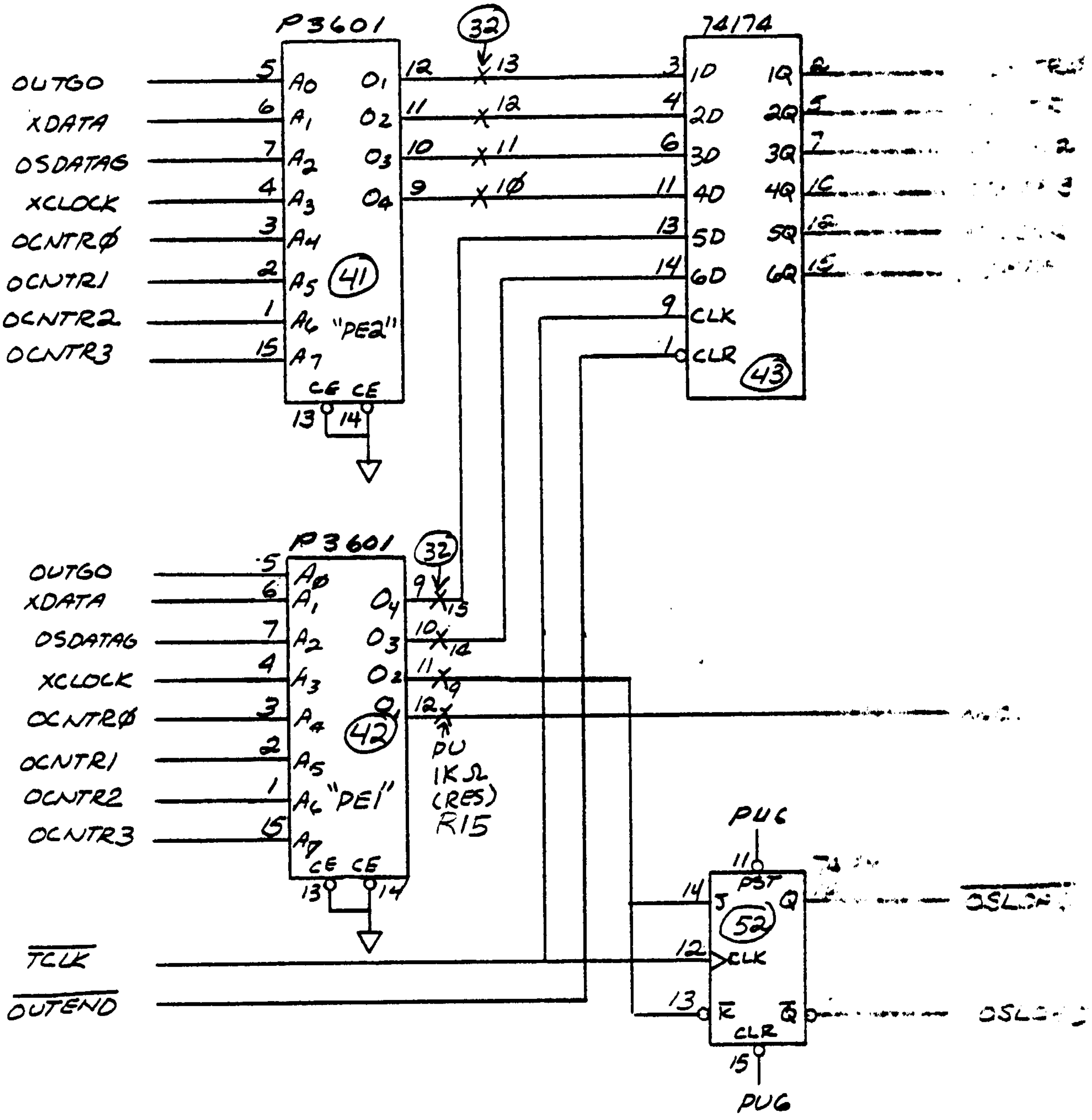
Title
ALTO II
ASSEMBLY,PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California
XEROX

216323 G

Sheet 16 Of

1586(3/73)



OUTPUT S.R. & P[illegible]

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.

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

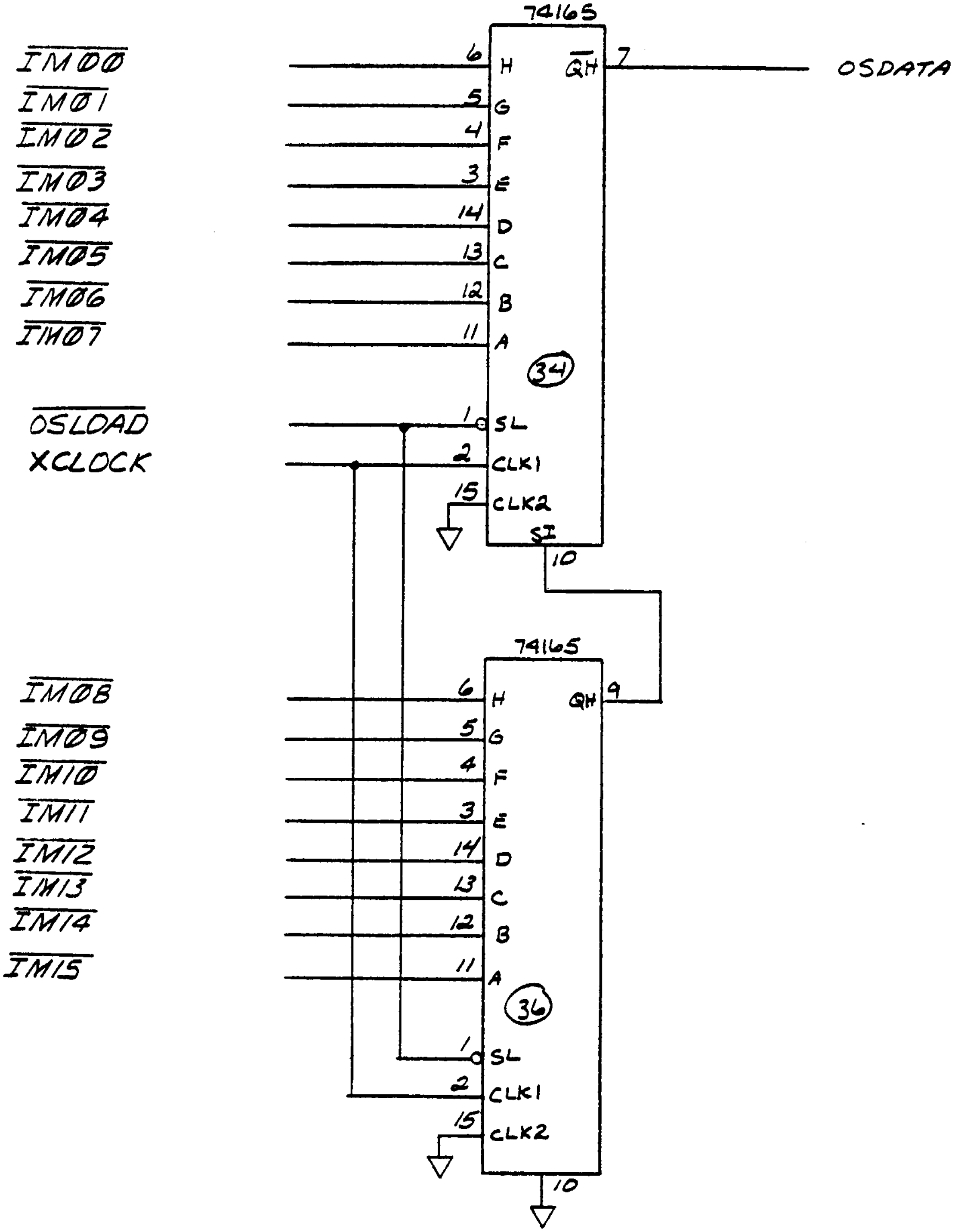
Xerox C[illegible]
El Segu[illegible]

XEROX

G

Sheet

1586(3/73)



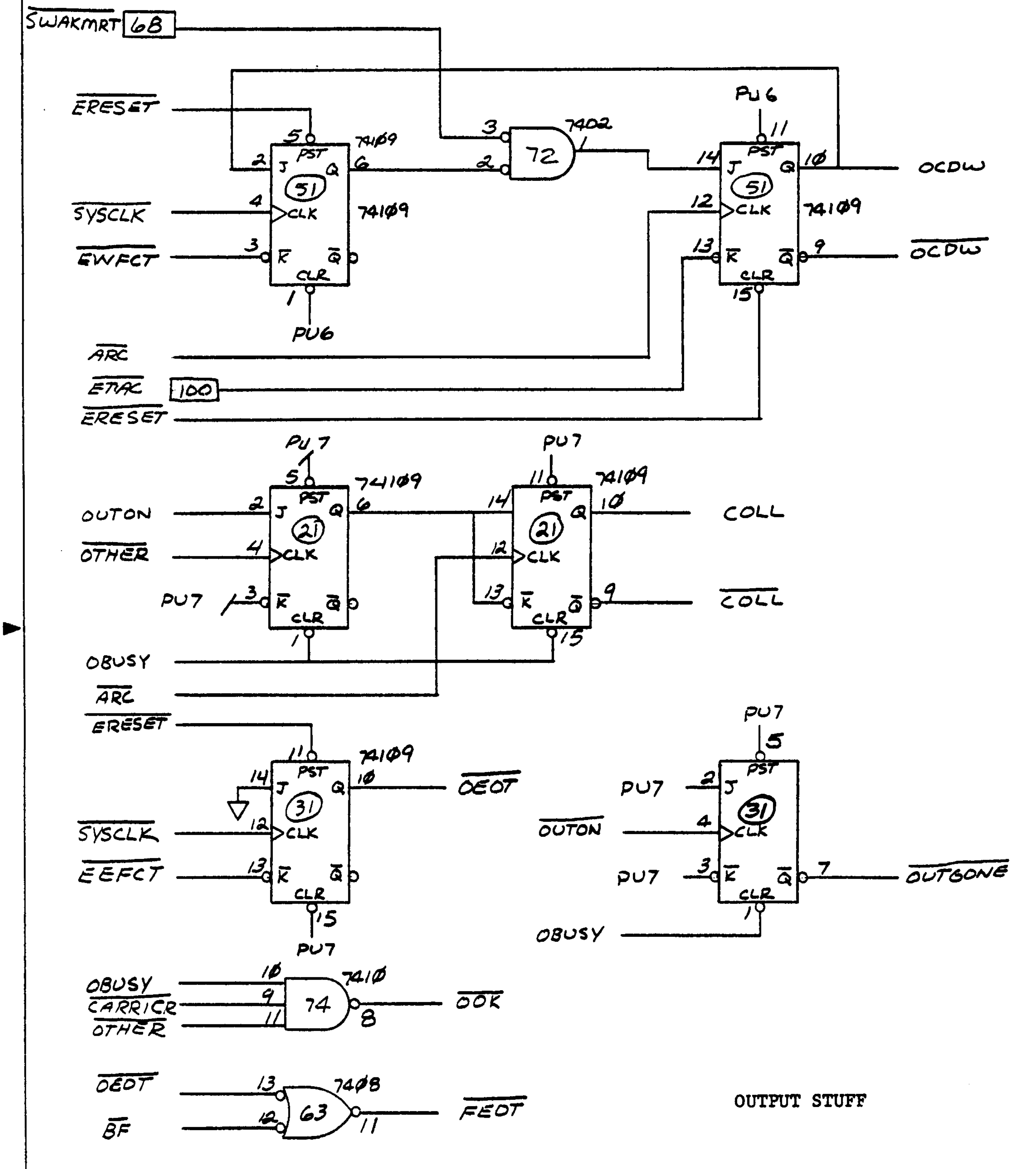
OUTPUT S.R. & PHASE ENCODER

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California
XEROX

216323 G

Sheet 18



OUTPUT STUFF

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.

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

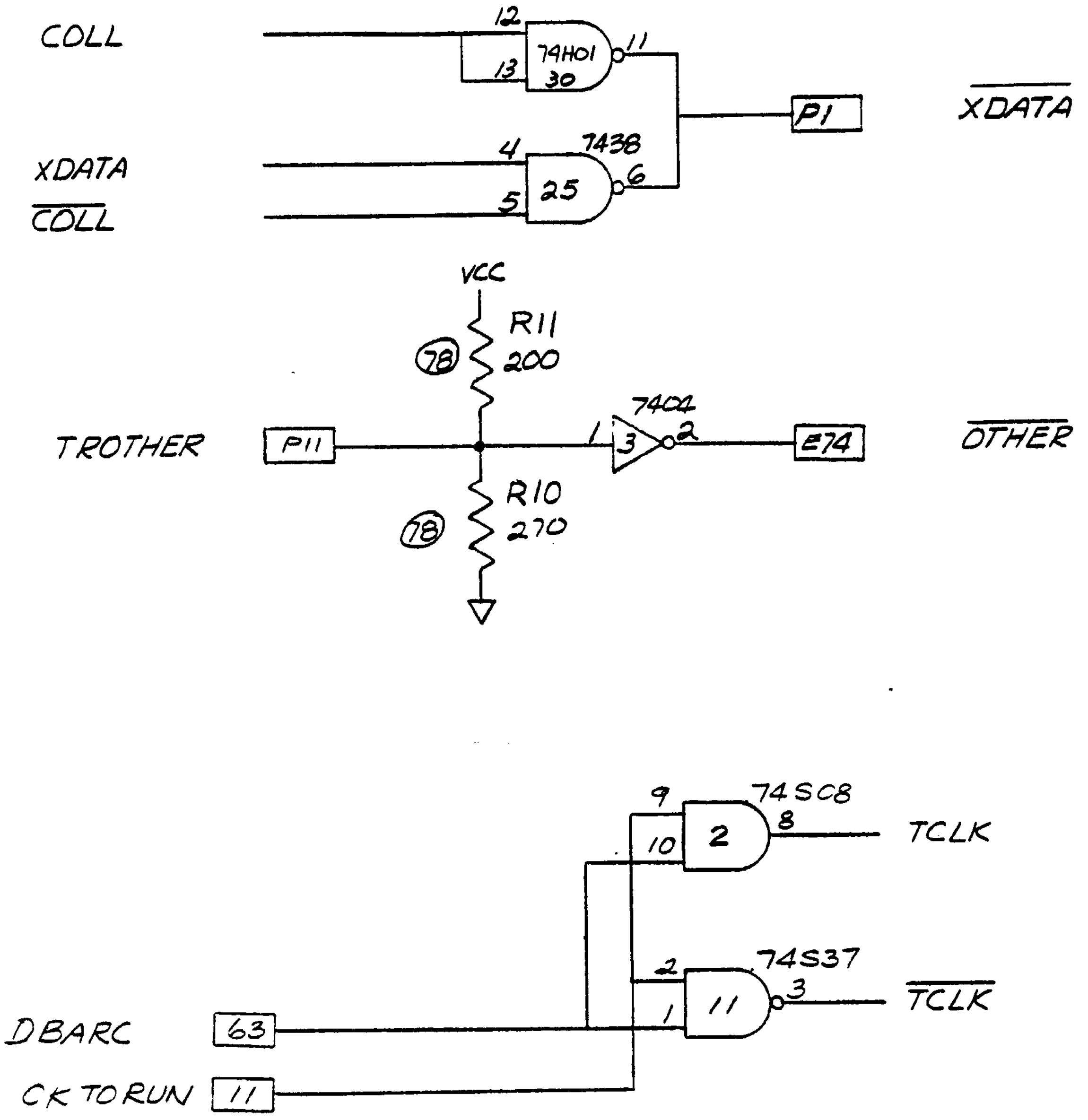
Xerox Corporation
El Segundo, California

XEROX

216323 G

Sheet 19 Of

1586(3/73)



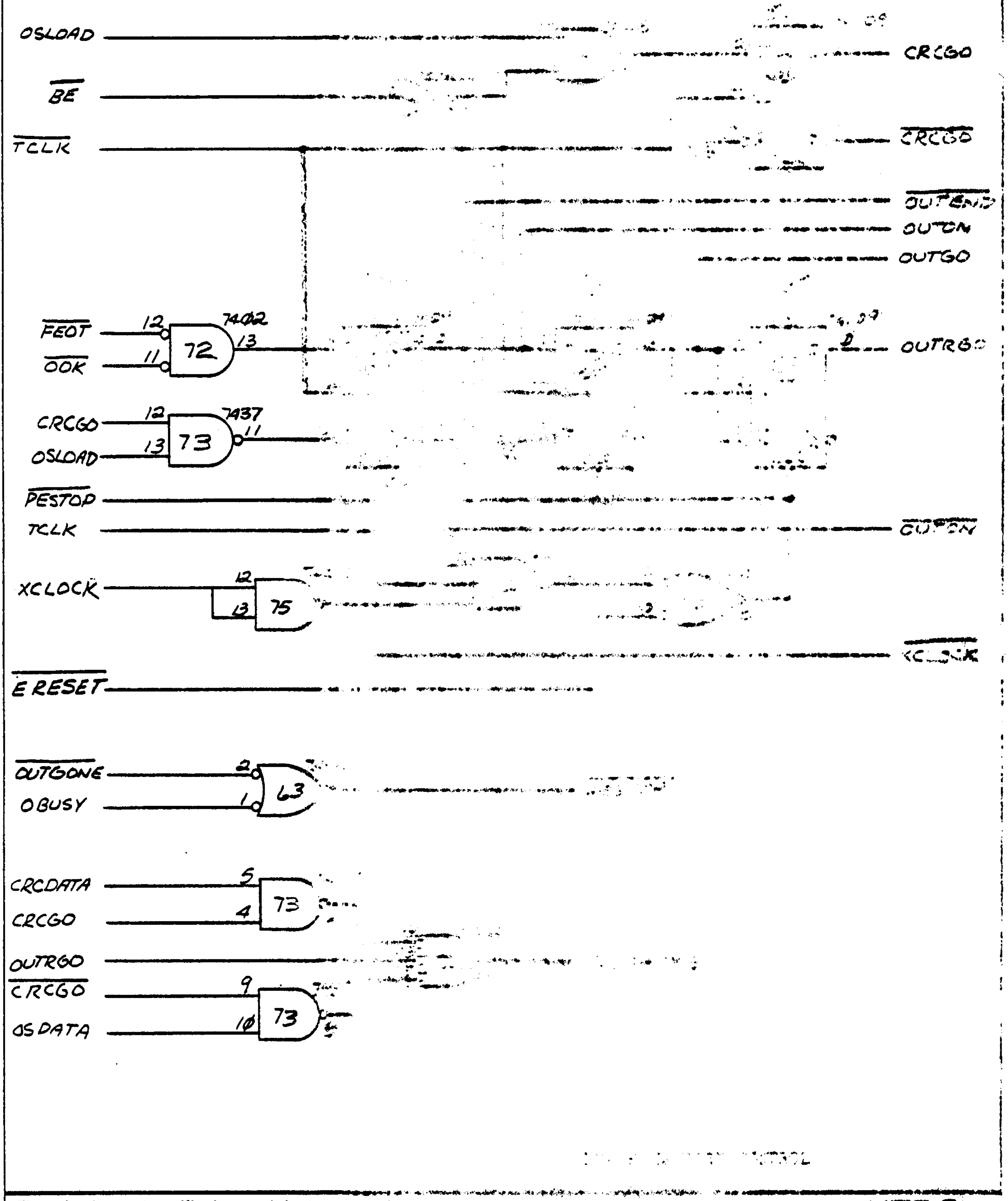
OUTPUT STUFF

Title
ALTO II
ASSEMBLY,PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

216323	G

Sheet 20 of

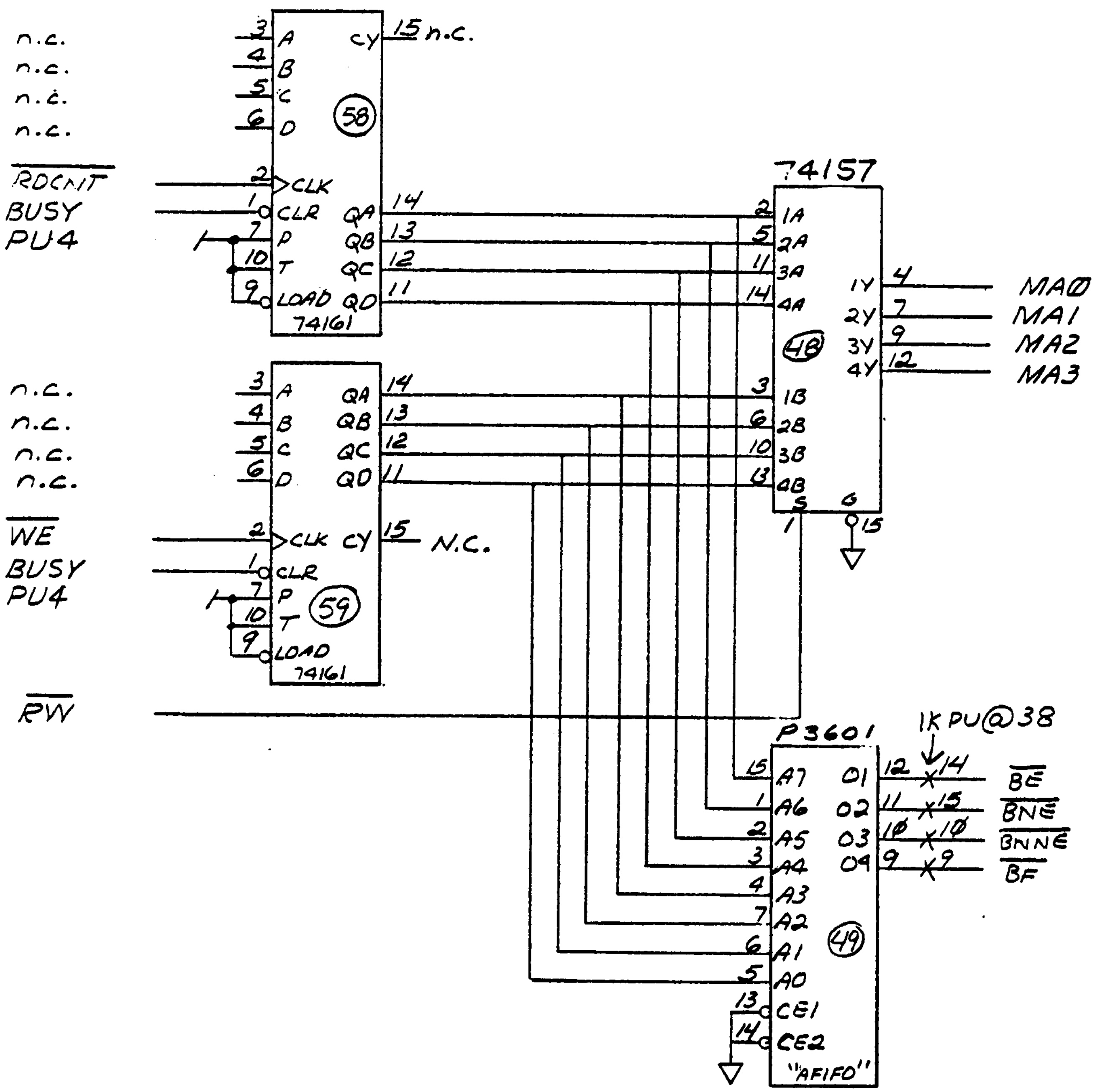


CONTROL

XEROX

6

1586(3/73)



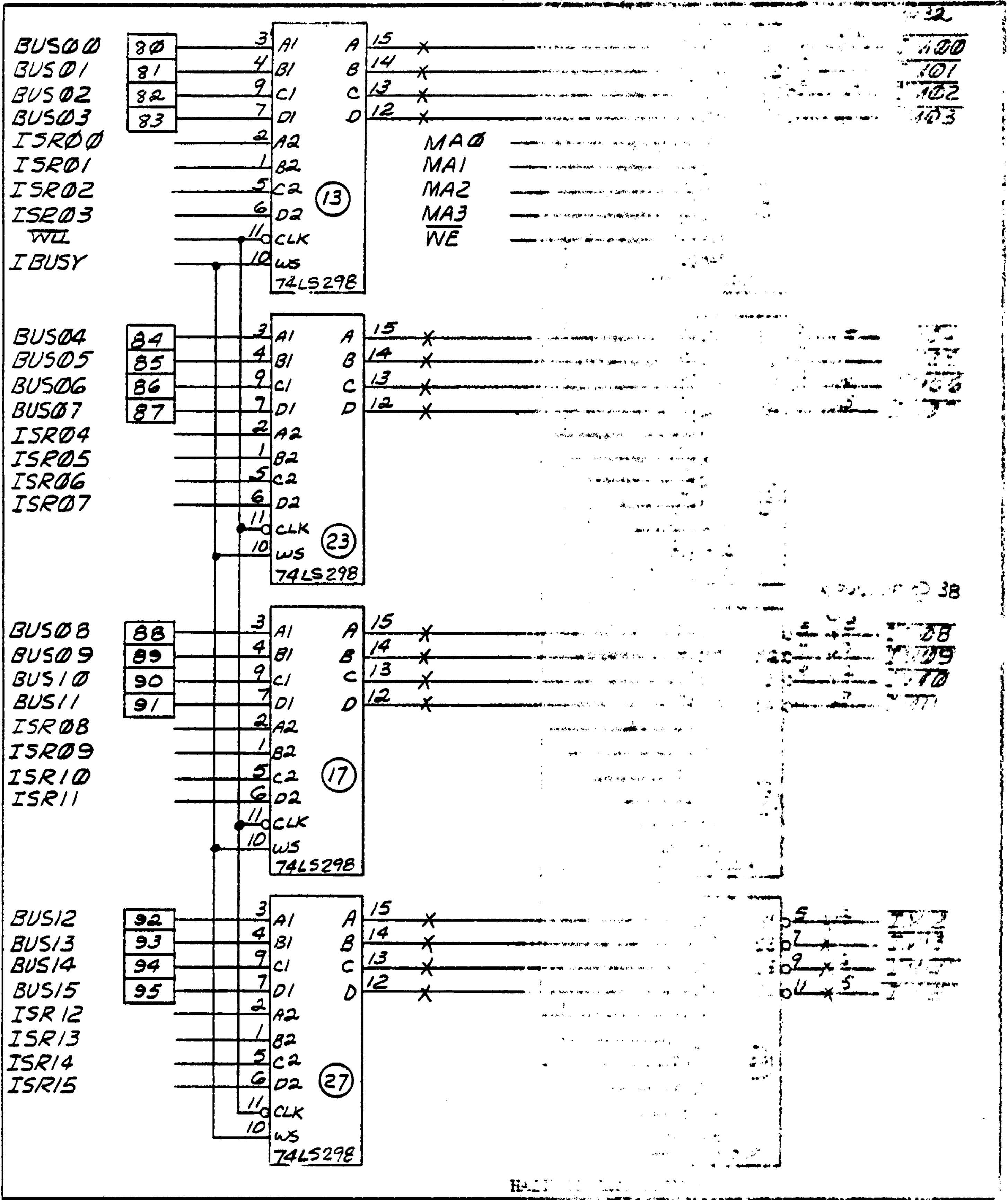
HALF DUPLEX BUFFER

Title
ALTO II
ASSEMBLY, PRINTED WIRING
ETHERNET

Xerox Corporation
El Segundo, California

XEROX

216323	G
Sheet 22 of	



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.

1586(3/73)

Title

ALTO II

ASSEMBLY, PRIN

ETHERNE

XEROX

216323 G

23 of

Wire No.	Term	From	To	Term	Wire Type	Notes	Signal	Chg. Let.
1		J1 - 13	J2 - 1		3			
2		25	2					
3		12	3					
4		24	4					
5		11	5					
6		23	6					
7		10	7					
8		22	8					
9		9	9					
10		21	10					
11		8	11					
12		20	12					
13		7	13					
14		-	14			8		
15		19	15					
16		6	16					
17		18	17					
18		5	18					
19		17	19					
20		4	20					
21		16	21					
22		3	22					
23		15	23					
24		2	24					
25		14	25					
26		J1 - 1	J2 - 26		3			

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

Title

ALTO II

ASSEMBLY, CABLE -

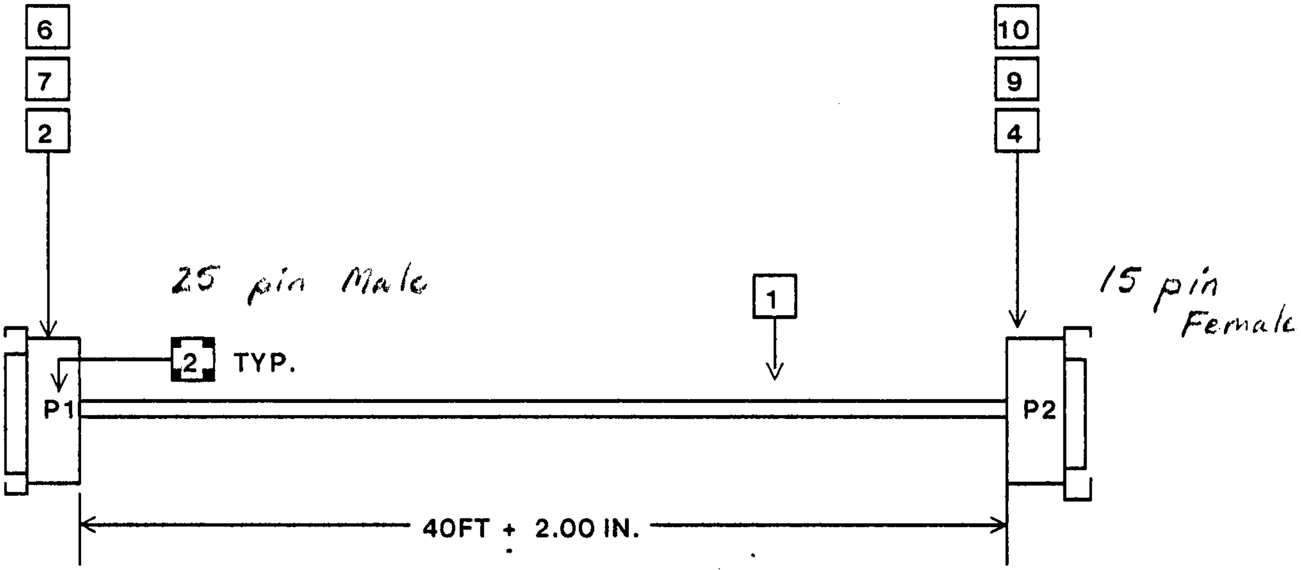
ETHERNET (INTERNAL)

Xerox Corporation
El Segundo, California

XEROX

216413 | C

Sheet 5 Of 5



PROPRIETARY NOTE ON SHEET 1 APPLIES TO ALL SHEETS

FILE: ETHEREXCABLE.SIL-3

UNLESS OTHERWISE SPECIFIED: DIMENSIONS IN: REFER TO FOR DWG INTERPRETATION

THIRD ANGLE PROJECTION

□ MILLIMETERS

□ INCHES

MAXIMUM SURFACE TEXTURE

TITLE ALTO II ASSEMBLY, CABLE-ETHERNET (EXT)	DWG. SIZE A4	DWG. NO. 216411 SHEET 3 OF 5	SHEET REV E

WIRE NO.	TERM	FROM	TO	TERM	WIRE TYPE	NOTES	SIGNAL
1		P1-12	P2-9		1	BROWN TWISTED PAIR	SPARE
2		P1-24	P2-1		1	BLACK	SPARE GND
3		P1-13	P2-3		1	GREEN	XDATA'
4		P1-25	P2-4		1	BLACK	GND
5		P1-16	P2-5		1	WHITE	TRDATA
6		P1-3	P2-6		1	BLACK	TRDATA GND
7	1	P1-8	P2-7		1	BLUE	TROTHER
8		P1-20	P2-15		1	BLACK	TROTHER GND
9		P1-11	P2-10		1	RED	+5V
10		P1-23	P2-11		1	BLACK	+5V GND
11		P1-19	P2-13		1	YELLOW	+15V
12		P1-6	P2-14		1	BLACK TWISTED PAIR	+15V GND

PROPRIETARY NOTE ON SHEET 1 APPLIES TO ALL SHEETS

FILE: ETHEREXCABLE.SIL-5

TITLE ALTO II ASSEMBLY,CABLE-ETHERNET (EXTERNAL)	DWG. SIZE A4	DWG. NO. 216411 SHEET 5 OF 5	SHEET REV. E

Inter-Office Memorandum

To Ron Cude

Date 25 February 75

From David Boggs

Location Palo Alto

Subject Debugging Alto Ethernets

Organization Parc

XEROX

This procedure uses EDP, the Ethernet debugging Program which is documented in a companion memo by Bob Metcalfe. This test procedure does not need the debugger interface. I have debugged over 50 interfaces this way and never touched the microcode.

References to signals in the interface will follow this pattern: SIGNALNAME, <chip>-<pin>, [schematic page number].

The Ethernet interface has two features not strictly connected with being an Ethernet. They are the serial number drivers and the program controlled boot feature.

The serial number feature decodes an EMULATOR SPECIFIC function (F1-16 - Read Serial Number, RSN), and enables 8 bus drivers which drive BUS08-15. The other input to each of these drivers come out to the backpanel, and are the machine serial number you wire on before shipping an Alto. The Ethernet uses the number returned by this function as its address, and these wires MUST be installed before the interface can be checked out. When EDP cranks up, it will read the serial number and tell you what it is in the top area of the display. If it is 377, something is not working or you haven't wired it.

The boot feature decodes EMULATOR SPECIFIC fuction (F1-17, SIO) and ANDs this with BUS00. If the result is true, a 74H01 is enabled which pulls down on SRESET', booting the machine just as if you had mashed the boot button. EDP will test this feature with the Boot command. If it is successful, you have to start EDP again. Ramload, which you will no doubt be hearing about soon uses the boot feature too.

To debug an Alto Ethernet interface, you need:

A known to be working Ethernet Interface
Two working transceivers connected by some Ether
Two working Altos

Start up EDP on the Alto with the working interface. Call this 'Alto A'. Extend the suspect board from the other Alto (presumably this is in a test stand where you can get at the board easily), connect up the Ethernet cables and power up the machine. call this 'Alto B'. Problems can be broken down into three major classes of increasing difficulty: Won't boot, bad transmitter, and bad receiver.

The general order of testing is:

Does the machine boot?
Does EDP give a reasonable serial number?
Does the EDP RESET TEST work?
Does the EDP OUTPUT TEST work?

Does the Collision stuff work?
Does the EDP INPUT TEST work?
Does the EDP ECHO TEST work?

Once the machine boots and gives a reasonable serial number, try the ECHO TEST. If it works, you are in luck, the interface works. Check the collision stuff and ship it.

WON'T BOOT

This is fairly simple. The only signals going between the Ethernet board and the rest of the Alto are:

The processor bus
The function busses
Wake task 7 (WAKEET' - E103)
Task 7 active (ETAC - E100)
SRESET'
RESET, OKTORUN, Clocks
NEXT 06 & 07.

The signals driven from the board are: the processor and NEXT bus, WAKEET' and SRESET'. If SRESET' is low (25-3 [3]), the machine is continuously booting; fix it. Step down the bus. All of the wires should be wiggling. If they are not, remove the chips on the Ethernet board, one at a time, corresponding to that bit:

BUS00 - 03:	13,14,45
BUS05 - 07:	23,24
BUS08 - 11:	17,15,25,5
BUS12 - 15:	16,6,26,27,35.

If it is a 74H01, either its output is shorted, or it is enabled when it shouldnt be. If it is a 74298, the chip is bad. Assured that it is not the bus being clobbered, look at WAKEET' 39-8 [2]. If it is low, one of the inputs to the WAKEET' gate is low. None of them should be. Booting should generate ERESET' 11-3, [2], which should clear every flip flop that matters. Next look at INPUTS to the NEXT bus drivers (30-4 and 30-1 [3]). They should both be low, otherwise the NEXT bus is being polluted. This is likely if WAKEET' is low, so make sure the interface is not spuriouly requesting wakeups before believing something is wrong in the NEXT bus area. If wakeups are happening, NEXT bus activity in the device is to be expected.

If you have removed all the gates on the processor bus, the next bus and SRESET', and it STILL wont boot, somebody is clobbering a clock, reset, OKTORUN, or the fuction buses.

Now run the EDP RESET test (on Alto B). This test wakes up the microcode which reads the hardware status, resets the hardware, and goes back to sleep. The status EDP is looking for is 2771. While this test is running, OCMD 35-6 [1] and ICMD 35-10 [1] should be wiggling, IBUSY 10-6 [2] and OBUSY 10-10 [2] should be low, IT 16-12 [4], CRCZ' 16-1 [4], COLL 15-12 [4], and IDL 15-9 [4] should be low. The EPFCT' 19-12 [1] and SIO 9-15 [1] decoder outputs should be wiggling.

BAD TRANSMITTER

Start the EDP INPUT test on Alto A. Start the EDP OUTPUT test on Alto B with destination Alto A. Does Alto B say everything is ok? Ifso, the transmitter appears to work from the originator's viewpoint (the transmitter status is ok but the data could be screwed up). Does Alto A say everything is ok? Ifso, the transmitter probably works. If Alto B immediatly slips off the deep end when this test is started, the buffer is probably screwed up in such a way as to cause continuous wakeup requests, which locks out the emulator. Swap the AFIFO rom. Check counters 58 and 59 [9] to make sure they look like they are counting binarily. If you tell EDP to turn off the display while poking around in the buffer area, the microcode responds much quicker and the jitter on many of the signals is dramatically reduced.

If the Alto is alive but the test is failing, look at OCMD 35-6 [1]. It should wiggle. So should OBUSY 10-10 [2]. If these are ok, look at 71-13 [11]. This must go high for the transmitter to start up. If not, FEOT' 63-11 [13] or OOK' 73-8 [13] is broken. All the inputs to OOK' should be high. One of the inputs to FEOT' should be low (in this case OEOT'). On the following three cycles after 71-13 goes high, 62-10 [11], then 62-6 [11], then 61-10 [11] should go high. If 61-10 goes high, the transmitter has started. OSLOAD 73-13 [11] should pulse every 5.44 microseconds, and XCLOCK 75-11 [11] should be a square wave with a period of 340 ns. Serial data should start coming out of 34-7 [12]. From there it goes into 72-10 and out of 73-6 [11] and into the phase encoder ROMs. These ROMs also produce XCLOCK and OSLOAD, so swap them out if these signals are broken. 41-9,10,11&12 [12] are a 4 bit counter, with 41-9 the LSB. These lines should count to 15 and then an OSLOAD should happen. Make sure TCLK' 63-6 [13] is ticking! It should have a period of 170 ns. If not check the jumper from 52-5 [13] to ground. It should be there or else the transmitter will run at half speed (a feature we have never used). Look at 43-15 [12]: this should have phase encoded data. When an OSLOAD (Output Shift register LOAD) happens and the Buffer is empty, (BE' is low), the message is over, except for shifting out the check character which has been accumulating in the CRC chip, 53 [3]. 61-6 [11] will set, OSDATAG, 73-6 [11] will switch from OSDATA, 72-10 [11], to CRCDATA,72-4 [11], the 16 bit CRC will shift out, and then 72-11 [11] will go low, and everything will shut down.

Assuming that Alto B is satisfied that the transmitter works, look at the input test running on Alto A. If it is not getting messages at all (no activity on the screen), set its serial number to zero which will make it accept any message. You do this by saying "WO". When you say "W" it will say something like: "I am serial Number:" and you say "0" (thats a zero, folks). then hit <CR>. Now restart the INPUT test. Getting anything? Ifso, print the packet (say "P"). If the packet is so long it scroll off the top of the screen, (it shouldn't, but then the interface is broken), you can stop the printout by hitting the space bar. LOOK at it. does the "IN" column on Alto B match the "OUT" column on Alto A when you print the packet there? If not, there is a problem in the data path. Figure out which bits are bad and start swapping 3101s, 74298s and 74165s. REMEMBER: The 'print packet' command in EDP prints BYTES. EVEN numbered bytes are bits 0 to 7 of a 16 bit word. If Alto A complains of bad CRC, replace chip 53. These chips fail often.

TESTING THE COLLISION STUFF

While running the EDP OUTPUT test, unplug the coax cable. EDP should show lots of "C"s. Whenever the transmitter starts up, COLL should set, and the microcode should wakeup and reset the interface. The cloud of stuff producing OCDW 51-10 [13] is a circuit which wakes up the output microcode every memory refresh time. When a collision happens, the microcode will use this to count memory refresh intervals until the next attempt to start the transmitter.

TESTING THE RECEIVER

Start the INPUT test on Alto B. Start the OUTPUT test on Alto A with destination Alto B. Does Alto B say everything is ok? Ifso, the receiver probably works and you can skip over this to the ECHO test. If the INPUT test on Alto B doesnt get anything, make him accept any message (type "WO<cr>"). Got anything? Ifso, there is a bad bit in the data path somewhere which screwed up the destination address so that Alto B would not accept it since it wasnt addressed to him. Its probably not the 74298s or 3101s since they are used for transmitting too. Its either the input shift register or the Bus drivers. Compare the packet being transmitted from Alto A (type "P" for 'Print packet') (the "out" column) with that being received by Alto B (again 'Print packet', but look at the "in" column). If the entire message is crap, there is (hopefully) a bad status indication. IT probably means the carrier 1-shot time constant is too short. Jumper another capacitor across C3. If that fixes it, replace the capacitor. CRC errors MIGHT be the CRC chip, but since this chip is used for transmitting too, this is unlikely if the transmitter works ok. On input, the CRC chip has its own private clock called RCLKP 50-13 [5]. This should be a square wave with period 340 ns. If it is not close to a square wave, diddle with C2 and R2 until it is.

We have now exhausted all of the easy problems. Start up the OUTPUT test on A, and the INPUT test on B. Look at 68-1 [7]. Thats phase encoded data coming into the interface. Now look at 68-3 [7]. It should look much better, but roughly the same. Is it also at 67-12, 57-13&14, 47-13&14 [5]? For EACH transition (low-to-high or high-to-low) there should be a 20 ns or so pulse at 67-11 [5]. That's called TRANS. Set the scope for a horizontal rate of 5 us/div. and hook channel 1 to RDATA 67-12 [5], for instance. Get the sync right so that the packet stays steady. It should just fit in the screen (total packet length of about 45 usec). Now with channel 2, look at 67-11 TRANS [5]. TRANS ok? Now look at 60-5 RCLK [5]. You should see a periodic signal approximately 255 ns up and 85 ns down. Now look at 50-13 RCLKP [5]. You should see a square wave: 170 ns up and 170 ns down. Now look at 60-13 CARRIER [5]. It should go high at the first transition of RDATA, and stay high until about 400 ns after RDATA goes low for the last time.

Look at 69-10 (INON) [7]. It should be high when the packet starts and go low when CARRIER (60-13) drops. If INON drops prematurely, make the receiver accept any message (type "WO<cr>", and then restart the INPUT test). Now INON should stay up for the entire message. If it doesn't, the ILOC 70-10 [7] flip flop is bad, since you have already assured yourself that once CARRIER comes on, it stays on for the duration of the packet. When the first 17 bits of a packet have arrived in the ISR, IMIP 70-6 [7] should go high, or 340 ns before the first ISRFULL pulse. If all these look reasonable, lets move on to the input shift register.

Look at 33-12 ISRFULL [6] (still with channel 2; leave channel 1 alone). You should see a pulse of duration 340 ns every 5.44 usec. When the packet is over, it should stay high for a longer time. This longer time is dependent on how soon the microcode gets around to clearing the interface at the end of a packet, so the trailing edge will jitter (it will be much shorter if the display if off). The trailing edge of all of these ISRFULL pulses should cause the 1-shot to fire (50-5) [6] clearing the shift registers. Each time an ISRFULL happens, the clock inputs to the buffer register should be pulsed (delayed perhaps by up to 170 ns), and the contents of the ISR should drop into the 74298s. After the packet has ended (IMIP and ILOC true) and after the microcode has read the last interesting word (BNE' 44-5 [7]) 69-6 INGONE [7] should set. As mentioned earlier, ISRFULL should still be high, so 68-8 IT [7] should be low.

The receiver now works. (Heh heh). The final test is:

ECHO TEST

This test shoots packets full of random numbers back and forth between the two interfaces, with the originator of the packet making sure that what comes back is what he sent out. The first 4 bytes will be different, so he ignores differences in them. On Alto A, say "ES". It is now an 'echo server'. Whenever it hears a packet whose second word contains 700, it will send the packet back to whomever it came from having changed the 700 into a 701. On Alto B, say "EU<Alto A><cr>Y". This Alto is now an 'echo user'. <Alto A> is the serial number of the 'echo server'. The 'echo user' makes up random length packets full of random numbers and sends them to the 'echo server', who sends them back to the 'user', who makes sure they came back undamaged. If you reply "N" instead of "Y" to 'Random packets?', it sends the same packet all the time so that you can look at it with a scope.

If you run this test, and it gives no errors, the Ethernet interface works. You can quote me.

Inter-Office Memorandum

To	Ethernet Debuggers	Date	January 12, 1976
From	David Boggs	Location	Coyote Hill
Subject	Dummy Transceivers	Organization	Parc

XEROX

I have designed a debugging aid for the Ethernet which I call a Dummy Transceiver. It is a small PC board (1.5" x 3.5") with a DA-15 connector at either end, one IC, 4 LEDs, some resistors and some test points. Figure 1 is the schematic. Here are some of its uses. Can you think of others?

1) Plug one end into the Ethernet connector on the rear apron of an Alto. Connect a standard Ethernet interface cable between the other connector of the Dummy Transceiver and the rear apron connector of another Alto (or Nova). You now have a private, 2-node, *Transceiverless Ethernet.* This allows debugging defective boards on a private network which cannot clobber the main net. It also removes the added variable of transceivers, simplifying debugging.

The Dummy Transceiver is the n=2 case of a completely connected Ethernet. By adding more inputs to the gates, more nodes could be accommodated. The limiting factor is complexity and, of course, reliability.

If two nodes are all there will ever be on some Ethernet, using a Dummy Transceiver to connect them is more economical. If the machines are separated by more than the length of an interface cable, the Dummy Transceiver can be used to concatenate two cables. If this isn't enough, then transcivers and coax should be used since the interface cable line drivers and receivers are not designed to drive longer pieces of cable.

2) Plugging a Dummy Transceiver into the rear apron connector of an Alto allows the transmitter to operate. A common failure mode of Ethernet interfaces is transmitter timeouts: for some reason, the transmitter control logic refuses to send a packet. If the output test fails with an interface cable and transceiver attached, but works with a Dummy Transceiver, the problem is in the cable or transceiver. If the output test works with the Dummy Transceiver plugged into the other end of the cable (at the transceiver connector), then the transceiver is the cause of the problem.

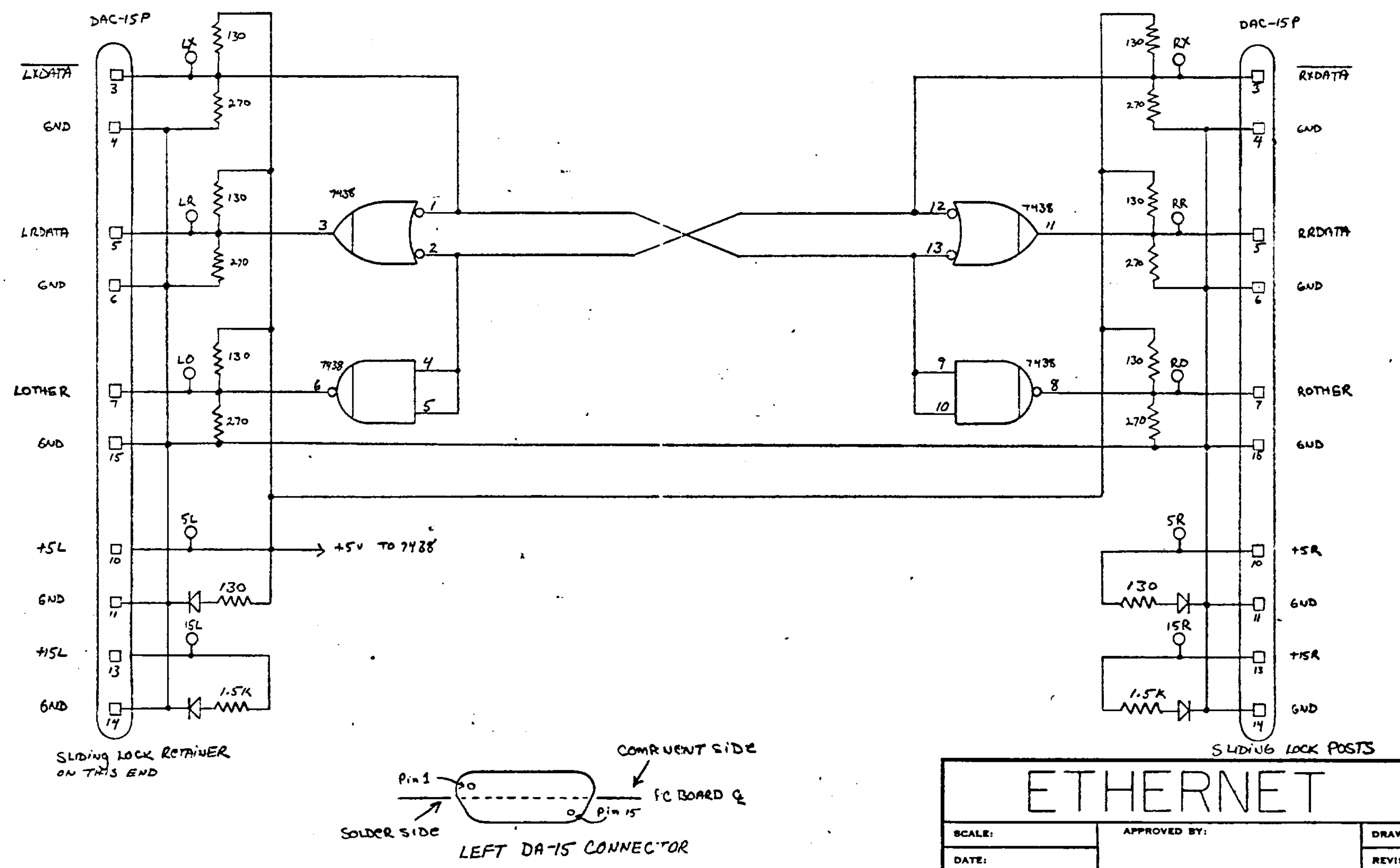
3) If the Ethernet interface is full duplex (Novas are, Altos aren't), then the whole interface can be tested since the Dummy Transceiver loops output back into input on both connectors.

4) The LEDs next to each connector light up if +5 and +15 are present at the connector.

5) Next to each connector is a row of test points for looking at the signals on the interface cable.

OPERATIONAL NOTE: The end of the Dummy Transceiver with the slide lock supplys power to the IC and all pullup resistors. This end must be plugged into a computer for the Dummy Transceiver to work.

PC layout and manufacturing by Tat Lam.



ETHERNET

SCALE:

DATE:

APPROVED BY:

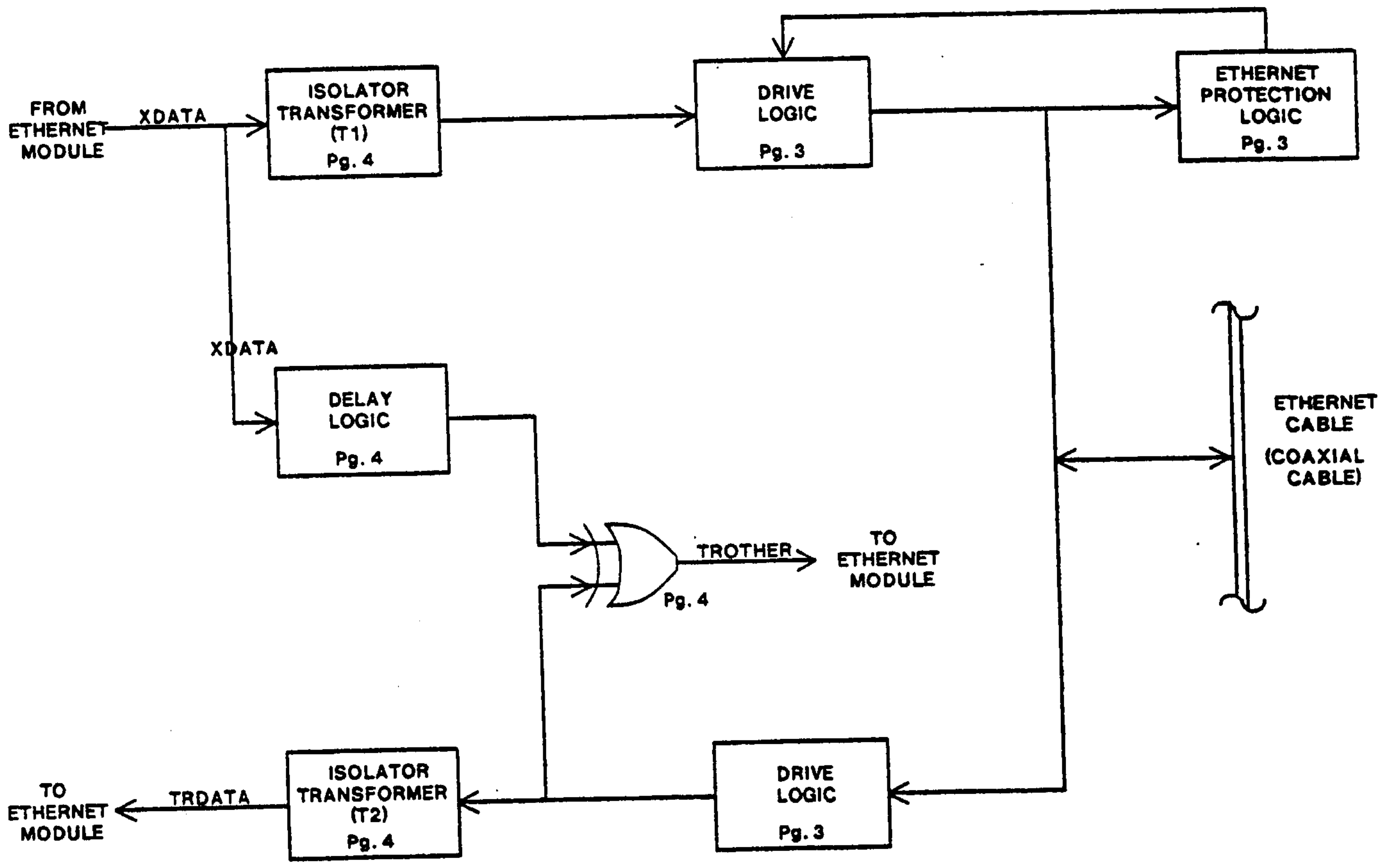
DRAWN BY db

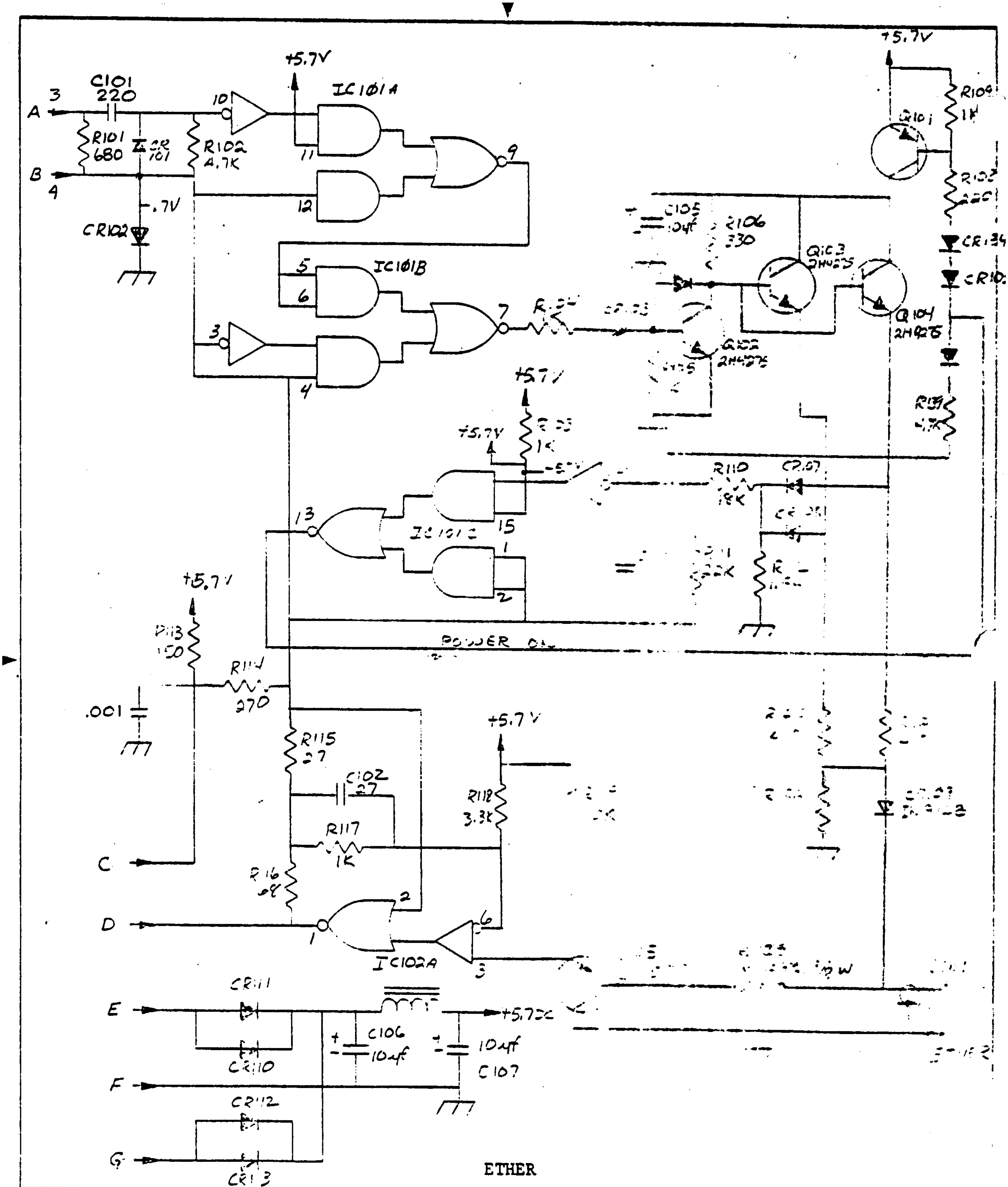
REVISED

DUMMY TRANSCEIVER

DRAWING NUMBER

ETHERNET TRANSCEIVER





ETHER

Title ALTO II

SPEC, PROCUREMENT-

ETHERNET TRANSCEIVER

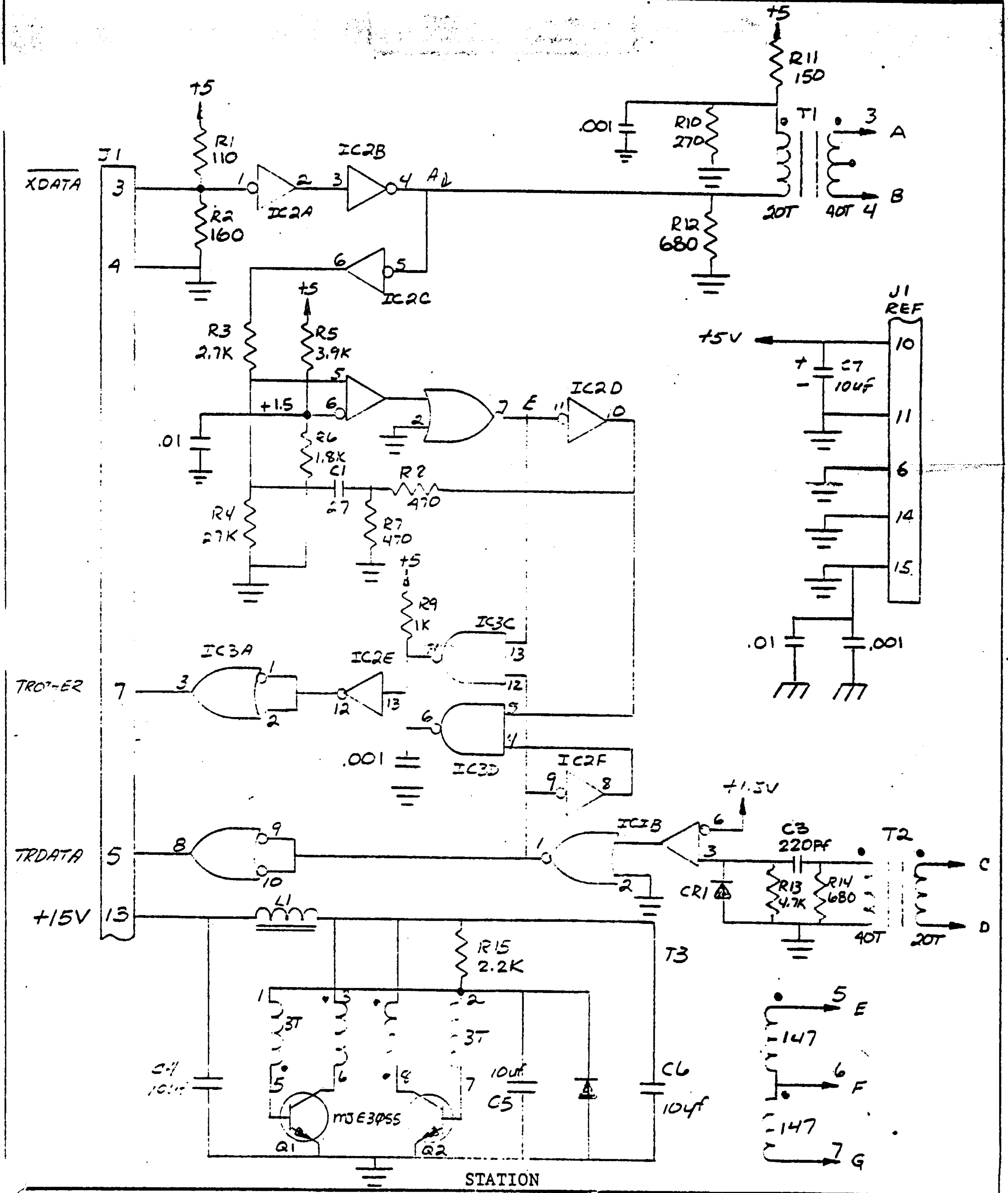
Sequence Columns

XEROX

209926

D

Sheet 2



STATION

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.

Title ALTO II

SPEC, PRO CUREMENT

ETHERNET TRANSCEIVER

Xerox Corporation
El Segundo, California

XEROX

209926	D

Sheet 4 Of 4