RCSL No: 44-RT1803

Edition: August 1977

Author: Mogens Verner Pedersen

Title:

TECHNICAL MANUAL
AMX702

RCSL No 42-I 2164

A/S REGNECENTRALEN
af 1979

Keywords:

RC3600, Asynchronous Telemultiplexor.

Abstract:

This paper contains the drawings and the description of the AMX702 Asynchronous Telemultiplexor.

(72 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL No 42-I 2164

FOREWORD

The AMX 702 is a reviced edition of AMX 701. The hardware is the same, but all the IC's have been replaced and renumbered, so the diagrams for AMX 701 or AMX 702 cannot be used for the other controller. The two editions are completely plug compatible and seen from a software point of view they are completely identically.

CONTENTS Page

DESCRIPTION

The Asynchronous Multiplexer AMX 702 is a controller to the RC 3600 computer, which enables the computer to communicate with up to 8 asynchronous terminals.

Normally the multiplexer is connected to the terminals via the junction panel VJP 703.

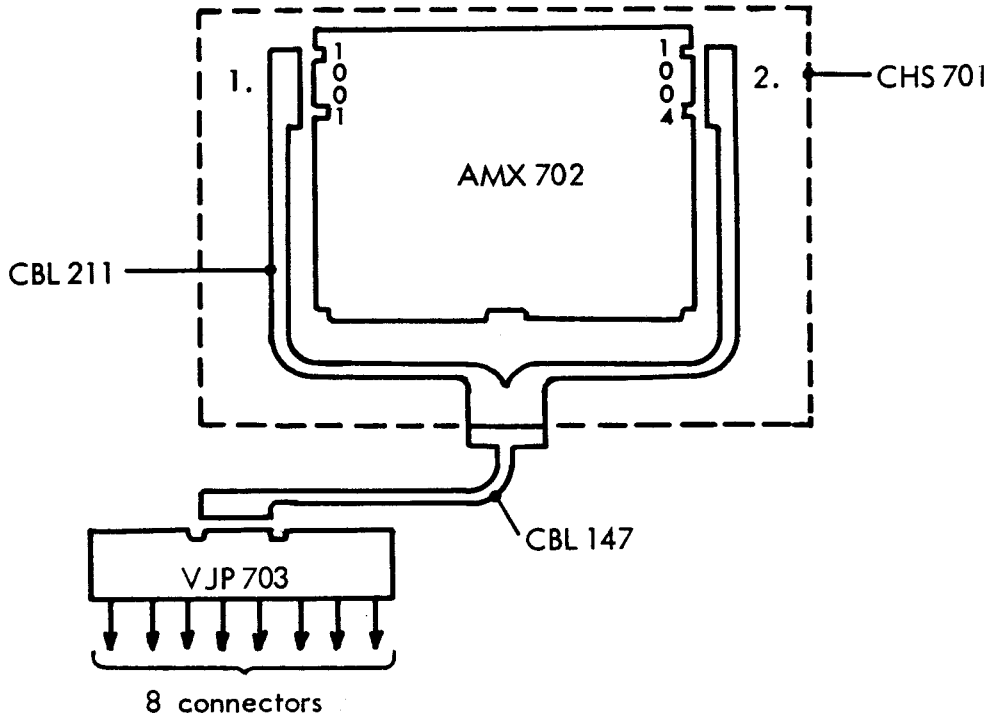


Fig. 1

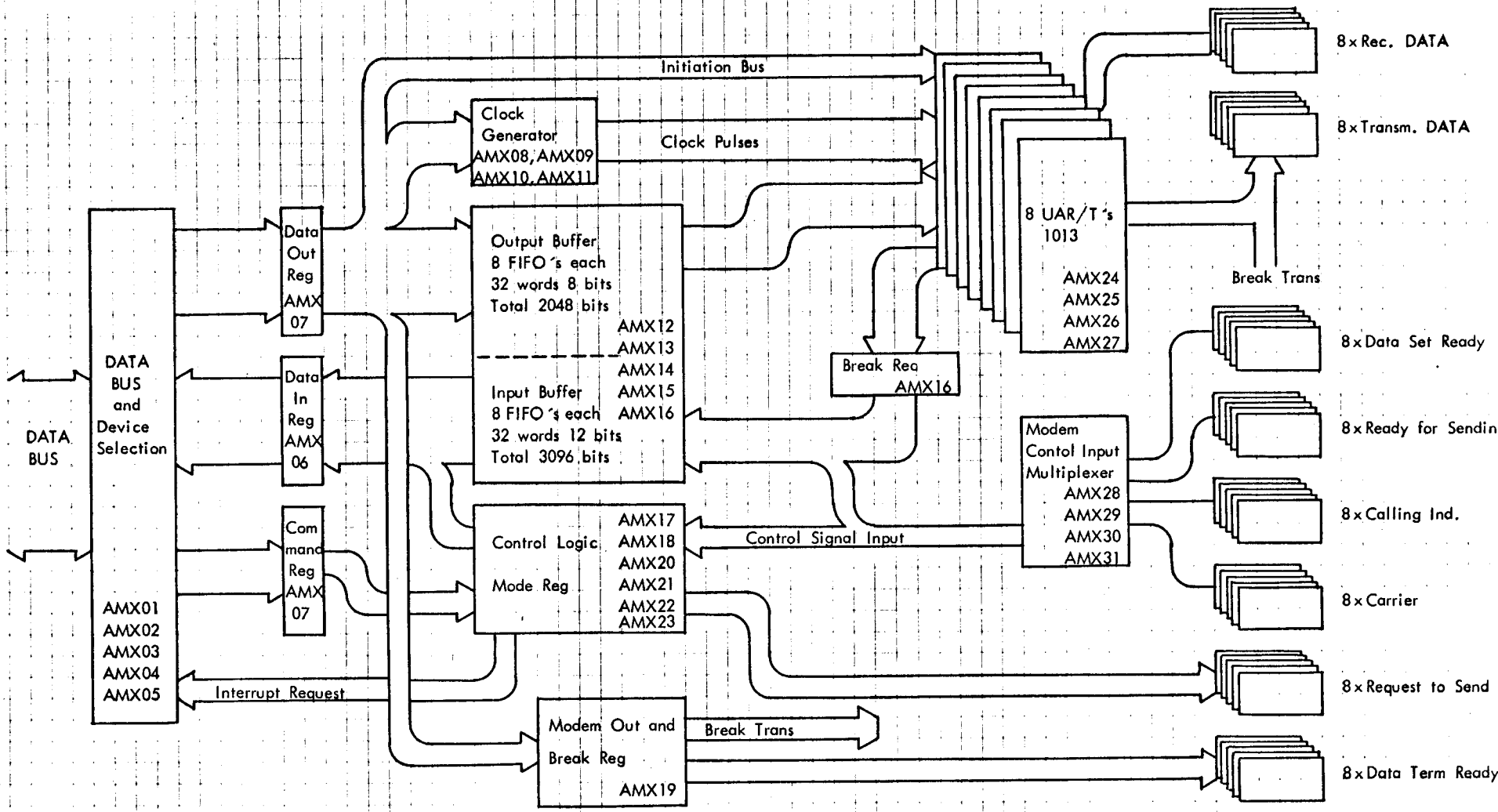
DATA PATHS in AMX 702

Figure 2 shows the Data Paths in AMX 702, and is also a block diagram for the unit. In each block is written the page number showing where in the logic diagrams, each block is drawn. The function of the controller is described in the Programmers Reference Manual and its contents is supposed to be wellkonwn.

The description follows the diagrams for the Data Paths in AMX 702.

A/S REGNECENTRALEN | Designed by | Drawn by | Dwg. Office Check | Design Check | Replaces Dwg. No. | due to ECN | Replaced by Dwg. No

A 11201



DATA PATH IN AMX 701/702

Figure 2

DATA BUS Connection and Device Selection

The bus connection is made in the normal way.

When a Data Out command with correct device number arrives, then the command is stored in Command Register, and the data is stored in the Data Out Register. The signal DATA OUT command is sent to the Control Logic to say that an event requesting action from Control Logic has occurred in the I/O Bus.

DATA OUT Register

In the DATA OUT Register the following information is stored:

1. After a DOA instruction the command is stored in bit (12:15), and the Channel No. in bit (5:7).
2. After a DOB instruction the character is to be transferred to the Output Buffer stored in bit (8:15) and the Channel No. in bit (5:7).
3. After a DOC instruction the initialization conditions are stored in bit (10:15).

DATA IN Register

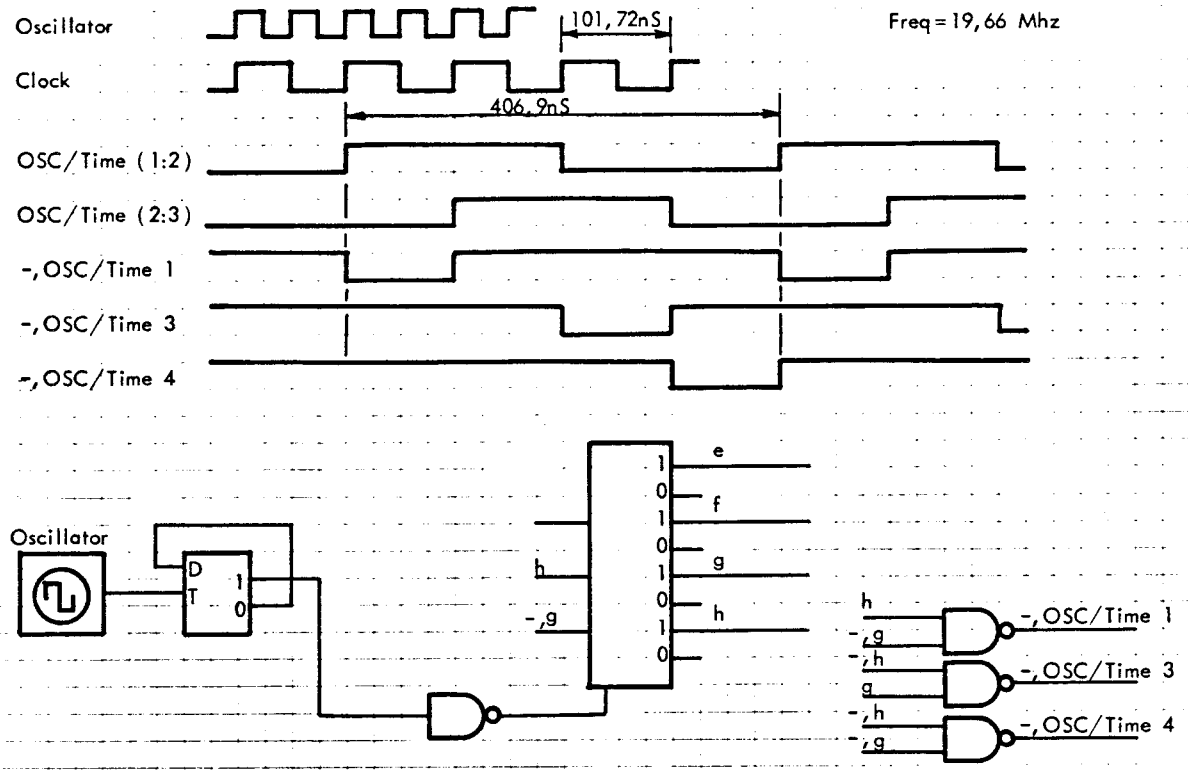
Input characters, their status bits and status information concerning the controller itself are stored in the DATA IN Register.

CLOCK Generator

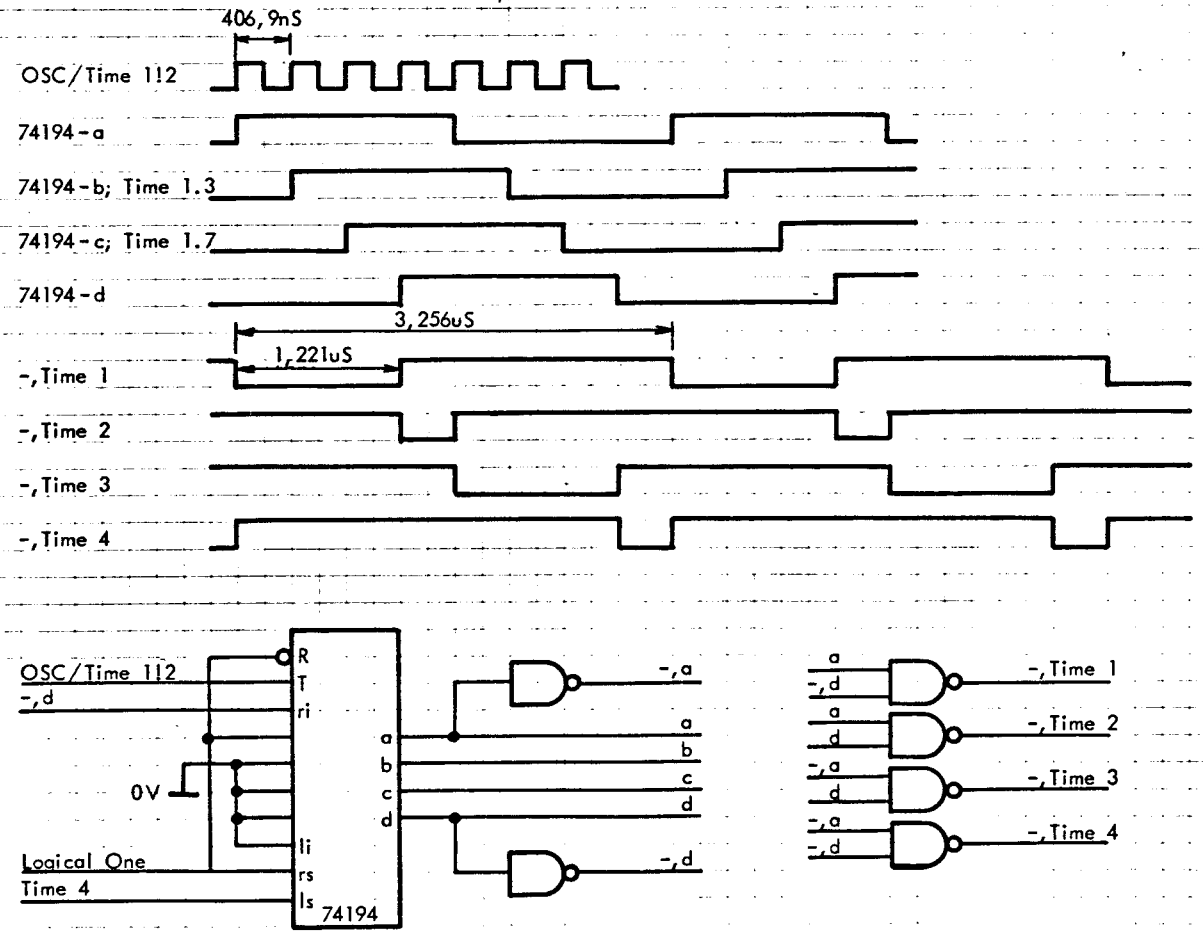
The CLOCK Generator generates clock pulses for:

1. Oscillator Clock System
2. Logic Clock System
3. Clock pulses to the UAR/T's.

TIMING for OSCILLATOR CLOCK SYSTEM



TIMING for LOGIC CLOCK SYSTEM



AMX 701
AMX 702

TIMING DIAGRAM FOR MASTER CLOCK GENERATOR
LOGIC DIAGRAM AMX 08

Figure 3

R 11282

Clock Generator, continued:

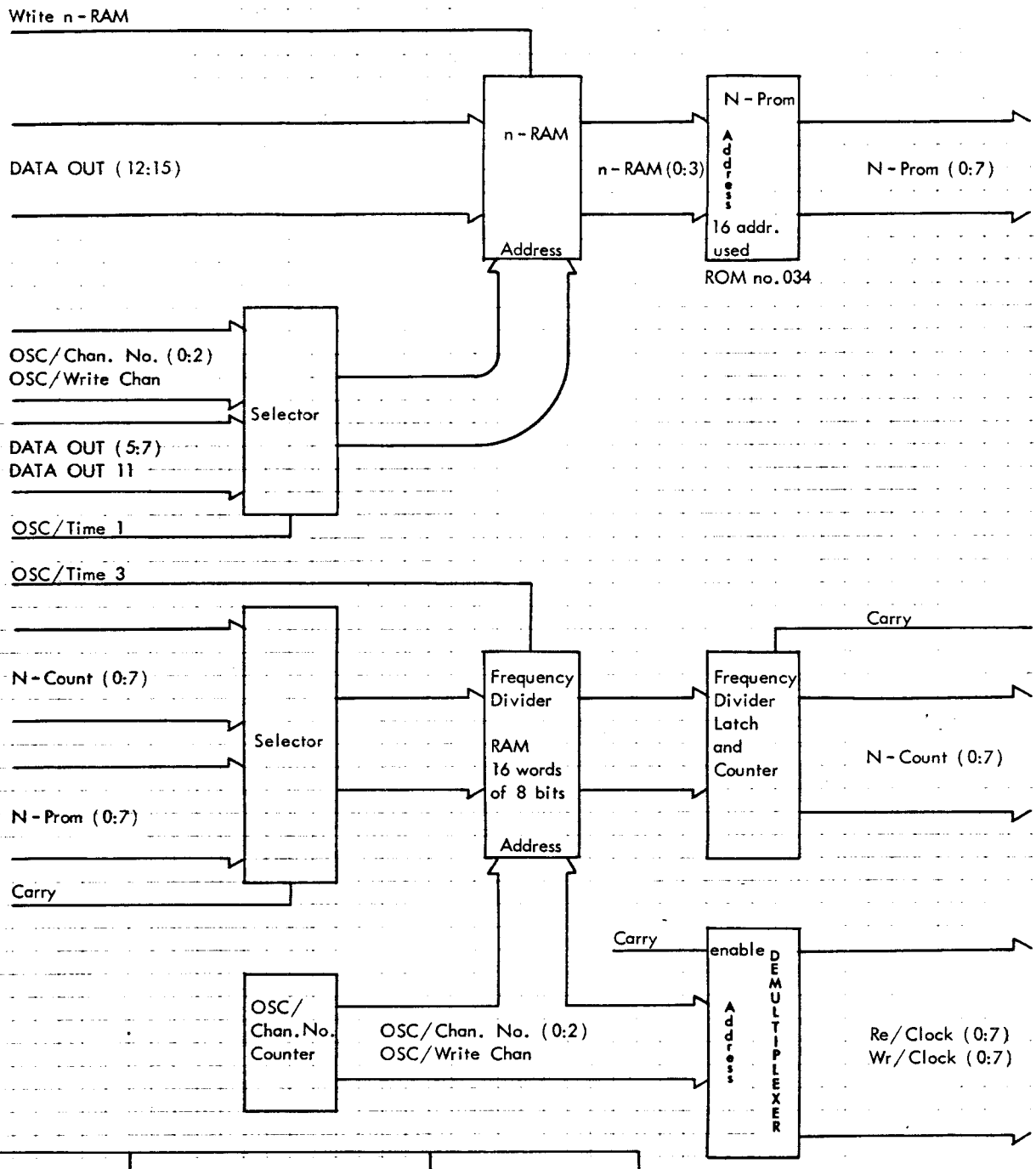
The two first systems are described in the timing diagram on figure 3 and the logic diagram AMX 08.

A block diagram of the clock pulses to the UAR/T's is provided in Figure 4. The UAR/T is a Universal Asynchronous Receiver/Transmitter and is a Large Scale Integrated Circuit. The UAR/T needs clock pulses for both the receiver and the transmitter parts. The clock pulses to the UAR/T must have a frequency 16 times the required bit rate. In AMX 702 the bit rate is controlled by the program, which may use different bit rates for receiver and for transmitter.

The OSC/Chan. No. Counter is a free running counter and its output is used as address for the different RAM's and demultiplexers in the circuit. The circuit is running in four steps and in these four steps the circuits take care of the clock for the receiver or the transmitter part of one channel.

The functions of the circuit in each step are:

1. The OSC/Chan. No. Counter changes and addresses an 8 bit number in the Frequency Divider RAM.
2. The number stored in the RAM is loaded into the Frequency Divider Latch and Counter.
3. The number in the Frequency Divider Latch and Counter is increased by one, and the Carry is one if the new number is 240, else it is zero.
4. If Carry is zero the new number is written back into the same position in the Frequency Divider RAM. If Carry is one the clock is set to one in the right flip-flop in the demultiplexer, and a number is written into the Frequency Divider RAM from N-Prom. The output from the N-Prom is a coding of the contents of the n-RAM.



Time (shifting to)	Action	Action if DOC instruction
1	Change Chan. No.	
2	Load Latch	Write n - RAM
3	Count in Latch	
4	Write back in RAM Set Clock = 1 if Carry	

AMX 701/
AMX 702

BLOCK DIAGRAM FOR GENERATION OF CLOCK
TO UAR/T
Logic Diagram AMX 09, AMX 10, AMX 11

Figure 4

A 11283

CLOCK Generator, continued:

If the addressed n-RAM contains the number 3, it means the channel has to operate at 1200 bps (see section 3.2 in the Reference Manual). The number 3 in the n-RAM is converted to the number 232 by the N-Prom, and after a Carry the number written into the Frequency Divider, RAM is 232 in this case. Next time the channel is addressed by the OSC/Chan. No. Counter, the number 232 is loaded into the Latch, and increased by one, and written back in the RAM again. This is repeated until the number in the Latch is 240, which gives a Carry, sets the clock flip-flop to one again and loads the RAM back to 232. In this way the clock flip-flop is set to one at a rate 8 times lower than the maximum bit rate 240-232=8. The maximum bit rate is 9600 bps, so in this case the bit rate will be 1200 bps.

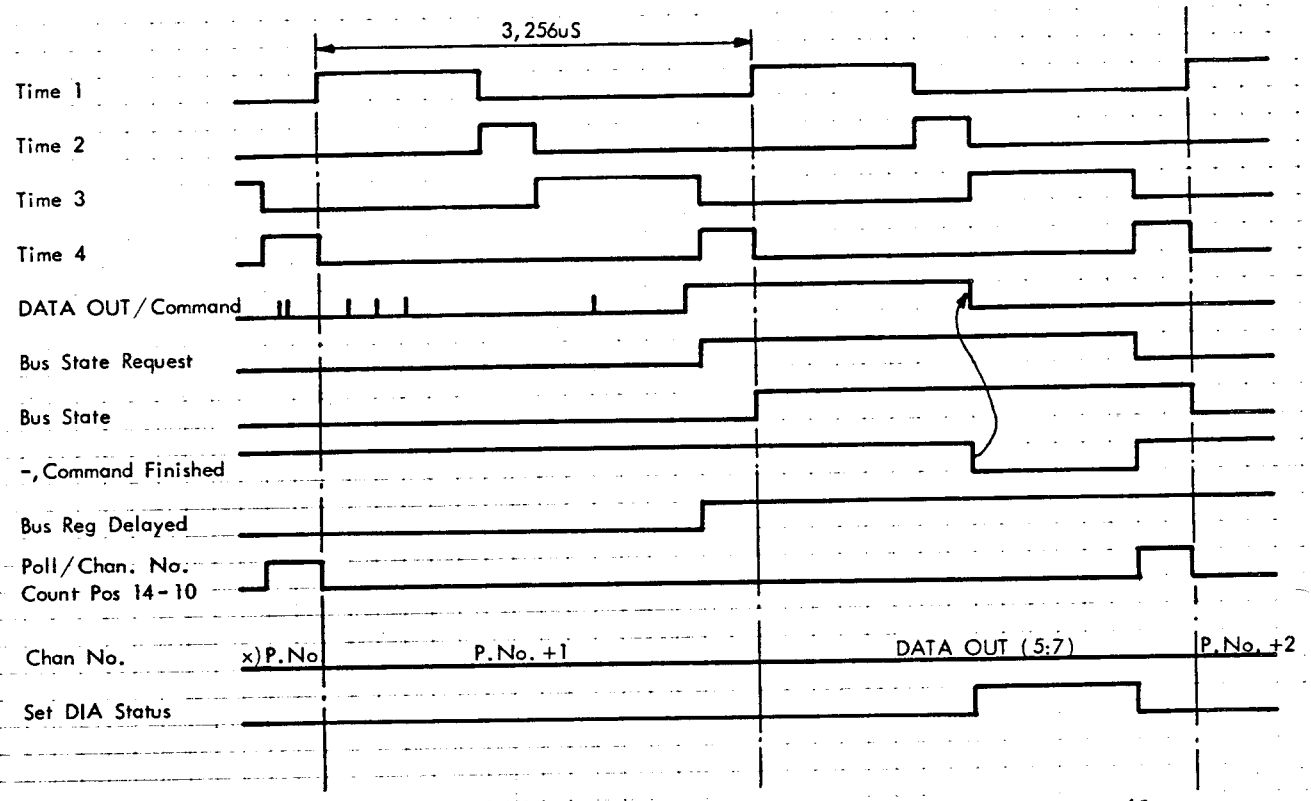
The contents of the n-RAM is changed by a DOC command with DATA OUT 10=1. Figure 5 shows the timing for the DOC command. The new contents of the n-RAM are written into the RAM at OSC/Time 1 and has no unwanted influence on the operation of the Frequency Divider.

OUTPUT BUFFER and INPUT BUFFER

The AMX 702 contains 8 channels and each channel contains an Output Buffer and an Input Buffer. Each Output Buffer has 32 words of 8 bits and each Input Buffer has 32 words of 12 bits. The buffers are "first in first out" buffers (FIFO) and the block diagram on figure 6 shows how these 16 FIFO's are made out of 5 RAMs, each containing 256 words of 4 bits.

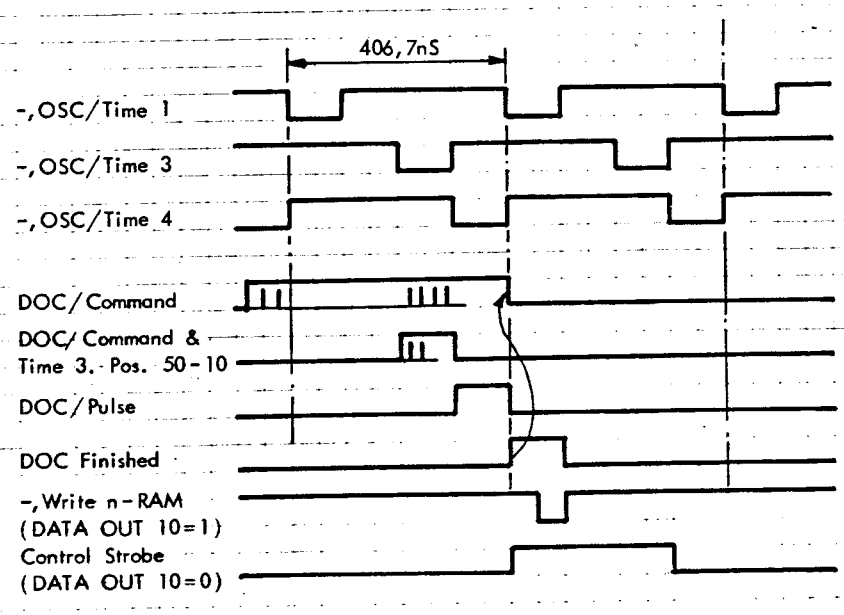
The buffer is addressed by the Write Pointer (used when writing into a buffer) or the Read Pointer (used when reading from a buffer) and by the Chan. No. (0:2). The Chan. No. is used as the three most significant bits in the addressing.

CLOCK for DOA or DOB COMMAND



x) P.No. is the number, which is stored in the Poll Channel Number Counter pos. 63.

CLOCK for DOC COMMAND

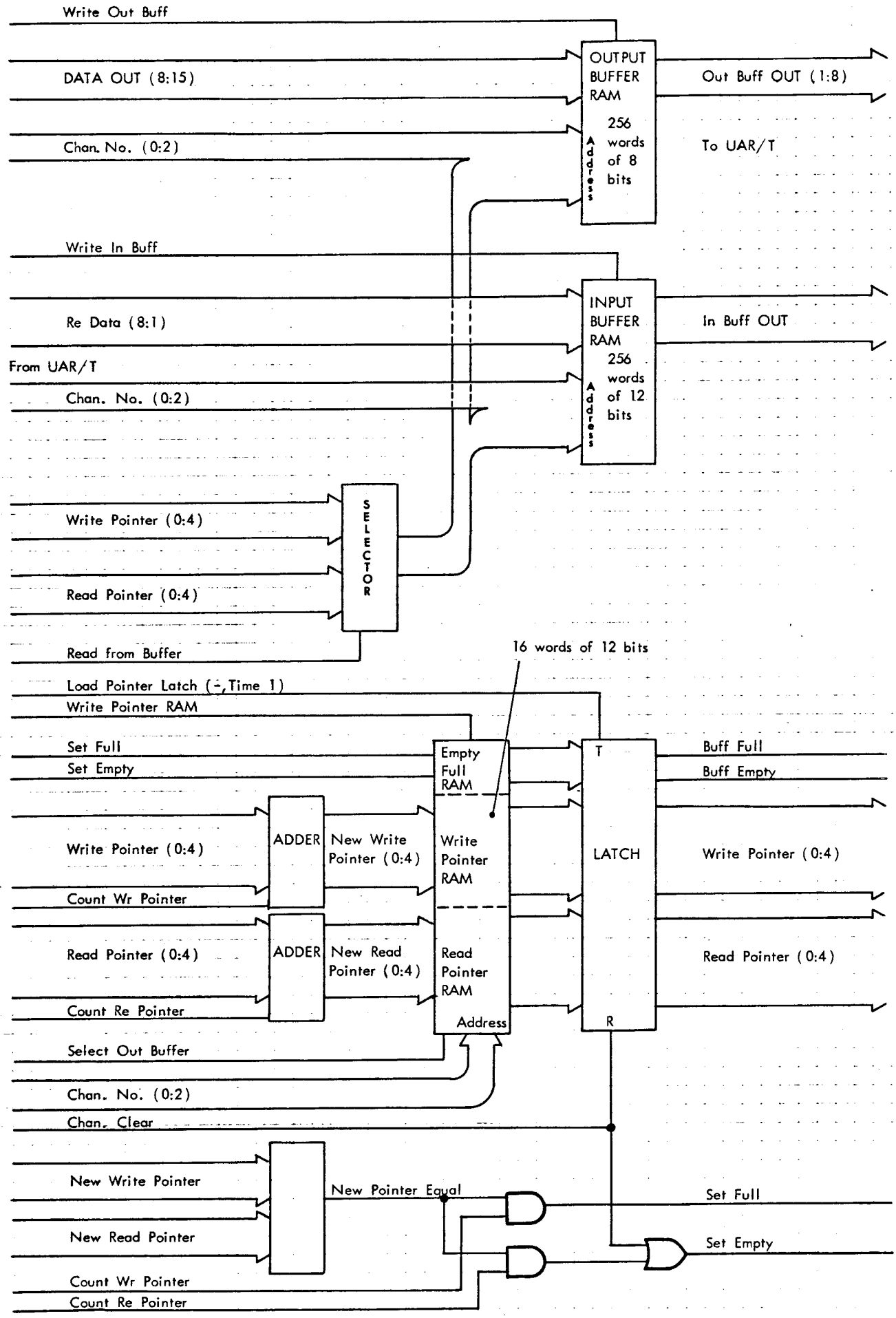


AMX 702

R 12114

TIMING DIAGRAM FOR DOA, DOB AND DOC COMMAND
Logic Diagram AMX 08 and AMX 17

Figure 5



AMX 701
AMX 702

BLOCK DIAGRAM FOR POINTERS AND BUFFERS
Logic Diagram AMX 12, AMX 13, AMX 14 and AMX 15

Figure 6

R 11285

OUTPUT BUFFER and INPUT BUFFER, continued:

When the Control Logic want to read a character from the Input Buffer it is done in the following way:

1. The correct Write Pointer, Read Pointer and Empty Full status is addressed, and loaded into the Latch.
2. The Input Buffer is now addressed and if Buff Empty = 0, then a strobe is sent to the Data In Reg loading the contents of the addressed cell in the Input Buffer into the Data In Reg.
3. If Buff Empty = 0 the Control Logic sets Count Re Pointer = 1 and the new Read Pointer is increased by one. Now the new Read Pointer and the new Write Pointer are compared and if they are equal the set Empty signal goes to one.
4. The new Read Pointer and the new Write Pointer are written into the Pointers RAM again and in case of Buff Empty = 0, the Read Pointer is increased by one. If the new Pointer is equal after a Read, this means that the buffer goes empty and the status is stored in the Empty Full RAM, too. (If they go equal during a write this means that the buffer goes full and this is stored, too).

Writing from a buffer is almost the same. What is different is that the Write Pointer is increased instead of the Read Pointer, and that a strobe is sent to the UARIT instead of the Data In Reg.

Clearing a buffer is done by means of the Control Logic by setting all bits in Read and in Write Pointer to logical one and by setting the Empty bit to one.

UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER

The Universal Asynchronous Receiver/Transmitter (UAR/T) in the AMX702 is an LSI device constructed on a single monolithic chip utilizing MTNS P-channel enhancement mode transistors. The controller uses 8 UAR/T's, one for each channel. Each UAR/T contains a Receiver part, a Transmitter part and Control part common for both the Receiver part and the Transmitter part.

1. Receiver part.

   When a character is received it is checked for correct parity and for correct stopbit, and the character is stored in the Re Data Reg in the UAR/T. The signal Re Data Available goes to one, telling the Control Logic that a character and its status bits are ready. The output from the UAR/T is tristate logic and the Control Logic is polling the Re Data Registers in the UAR/T's by setting the signal Enable Re Data to one, which transfers the data from the Re Data Reg to the output of the UAR/T. When the Control Logic has transferred a character and its status bits from Re Data Reg to the Input Buffer, it sets the signal Reset Data Available to one, which resets the signal Re Data Available.

2. Transmitter part.

   The Control Logic transfers data from the Output Buffer to the UAR/T by setting the signal Write Strobe to one, which loads the character into Data Holding Register in the UAR/T. The UAR/T starts to transmit the character with a start bit, 5 to 8 data bits, and possibly one parity bit (odd or even), and with one or two stop bits. When the Data Holding Register can accept a new character the signal Trans Buffer Empty goes to one.

3. Control part.

   The Control part of the UAR/T takes care of the following parameter for the Receiving and Transmitting:

   1. Char. length (5,6,7 or 8 bits).
   2. Odd/even parity or no parity.
   3. one or two stop bits (only for transmitting).

   Which parameter a UAR/T has to use is loaded in the Control Bit Holding Register in the UAR/T by a DOC command. The DOC command sets the signal Control Strobe to one , loading the DATA OUT (11:15) into the Control Bit Holding Register as shown on figure 5.

MODEM CONTROL INPUT MULTIPLEXER

The Modem Control Input signal is multiplexed, so that the input from the correct channel is avaiable for the Control Logic.

MODEM OUT and BREAK REGISTER

These two registers are constructed on two addressable Latches. The IORST sets all signals to logical zero.

CONTROL LOGIC

The Control Logic controlling the 8 UAR/T's, the Output Buffer, the Input Buffer, the Data Out Register and the Data In Register. Figure 5 shows a timing diagram for the Control Logic. The unit has two states namely the Bus State and the Poll State.

In Bus State the logic takes care of DOA commands, DOB commands and IORST, and the Chan. No. is DATA OUT (5:7) for the DOA and the DOB command.

In the Poll State the Control Logic senses the Modem Control Input signals, it transfers data from Output Buffer to the UAR/T or it transfers received data from UAR/T to the Input Buffer. The Chan. No. is now the output from the Poll/Chan. No. counter, like it is under a IORST, although an IORST is executed in Bus State.

The timing diagram shows how the signal DATA OUT/command sets the Bus State Request flip-flop, which stops the Poll/Chan. No. counter, and sets the unit in Bus State. In Bus State the signal Command Finished clears the DATA OUT/command signal and ends the Bus State.

CONTROL LOGIC, continued:

The IORST sets the Bus State and the AMX/Reset Reg, and sets the Poll/Chan. No. counter to zero. Poll/Chan. No. is used as Chan. No. during AMX/Reset and Command Finished is not set to one until Poll/Chan. No. 8 is one. Poll/Chan. No. 8 is one when the counter has reached to 8 and All Channel is then cleared by AMX/Reset.

The microprogram run by the Control Logic is explained in details in the 8 pages for Bus State for AMX 702 and in the 3 pages for Poll State for AMX 702. The microprogram used by the Logic in Bus State is determined by the arrived command, and for the Poll State it is determined by priority logic in the controller.

The priority is:

1. Poll/Modem Status
2. Poll/Re Data
3. Poll/Trans Data

INTERRUPT REQUEST

The controller has no BUSY flag and no DONE flag, but it can send an Interrupt Request. It has a flip-flop called Dummy DONE, which operates like a normal DONE flag but it cannot be sensed by the program. Dummy DONE is set by the Control Logic when input arrives to the Input Buffer and when an Output Buffer goes empty. Dummy DONE is only set to one if the controller has sent the command Enable Interrupt and setting Dummy DONE clears the Enable Interrupt flip-flop. If Dummy DONE is one and if Interrupt Disable is zero, the controller sends an Interrupt Request. Interrupt Disable is controlled by Mask bit 2.

BUS STATE for AMX 701/AMX 702

Time (shifting to)	AMX / Reset (IORST)	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch x) Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg	Set Select Out Buffer = 0 Set Count Re Pointer = 0 Set Count Wr Pointer = 0 Set Read from Buffer = 0 Set Chan. Clear = 0
4	Set Receive-mode = 0 Set Transmit-mode = 0	

x) Set DATA TERM READY OFF

Set Trans Data to logic one (Stop Break)

Microprogram for AMX 701/AMX 702

AMX / Reset

BUS STATE for AMX 701/AMX 702

Time (shifting to)	DOA/Command Receive, Stop Receive Transmit, Stop Transmit	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg	Set Count Re Pointer = 0 Set Count Wr Pointer = 0 Set Read from Buffer = 0 If Transmit = 1 Then Select Out Buffer = 1 If Receive ! Transmit = 1 And DATA OUT 11 = 1 Then Chan. Clear = 1
4	If Chan. Clear = 1 Then Write Pointer RAM x)	

x) If Receive = 1 Then Set Receive-mode = 1
If Stop Receive = 1 Then Set Receive-mode = 0
If Transmit = 1 Then Set Transmit-mode = 1
If Stop Transmit = 1 Then Set Transmit-mode = 0

Microprogram for AMX 701/702

DOA/Command

BUS STATE for AMX 701/AMX 702

Time (shifting to)	DOA/Command Select In Buffer	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg If Buff Empty = 1 Then Set Empty Status	Set Select Out Buffer = 0 Set Count Re Pointer = 1 Set Count Wr Pointer = 0 Set Read from Buffer = 1 Set Chan. Clear = 0
4	If Buff Empty = 0 Then Write Pointer RAM	

Microprogram for AMX 701/AMX 702

DOA/Command

BUS STATE for AMX 701/AMX 702

Time (shifting to)	DOA/Command Select Modem Status	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg Setting: DIA DATA 0=Calling Ind Reg DIA DATA 1=Carrier Reg DIA DATA 2=Data Set Ready Reg	Set Select Out Buffer = 0 Set Count Re Pointer = 0 Set Count Wr Pointer = 0 Set Read from Buffer = 0 Set Chan. Clear = 0
4		

BUS STATE for AMX 701/AMX 702

<table>
<tr><td>Time
(shifting to)</td><td colspan="2">DOA/ Command
Select In Buff Status
Select Out Buff Status</td></tr>
<tr><td>1.3</td><td colspan="2">Set Receive-mode and Transmit-mode Register
Set Modem signal Register</td></tr>
<tr><td>1.7</td><td colspan="2"></td></tr>
<tr><td>2</td><td colspan="2">Load Pointer Latch
Load Buff Full Reg
Load Buff Empty Reg</td></tr>
<tr><td>3</td><td>Load DIA Status Reg
Setting:
DIA DATA 0 = Not Buff Full
DIA DATA 1 = Buff Empty
DIA DATA 2 = Logical zero</td><td rowspan="2">Set Count Re Pointer = 0
Set Count Wr Pointer = 0
Set Read from Buffer = 0
Set Chan. Clear = 0
If Select Out Buff Status = 1
Then Select Out Buffer = 1</td></tr>
<tr><td>4</td><td></td></tr>
</table>

Microprogram for AMX 701/AMX 702

DOA/ Command

BUS STATE for AMX 701 /AMX 702

Time (shifting to)	DOA/Command Set Data Term. Ready, Clear Data Term. Ready, Start Break, Stop Break	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch x) Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg	Set Select Out Buffer = 0 Set Count Re Pointer = 0 Set Count Wr Pointer = 0 Set Read from Buffer = 0 Set Chan. Clear = 0
4		

x) If Set Data Term. Ready = 1 Then Set DATA TERM READY ON
If ClearData Term. Ready = 1 Then Set DATA TERM READY OFF
If Start Break = 1 Then Set Trans Data to logic zero
If Stop Break = 1 Then Set Trans Data to logic one

Microprogram for AMX 701/AMX 702

DOA/Command

BUS STATE for AMX 701 /AMX 702

Time (shifting to)	DOA/ Command Clear One Character	
1.3	Set Receive - mode and Transmit - mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg Setting: DIA DATA 0 = Not Set Empty DIA DATA 1 = 0 DIA DATA 2 = 0	Set Select Out Buffer = 1 Set Count Re Pointer = 1 Set Count Wr Pointer = 0 Set Read from Buffer = 0 Set Chan. Clear = 0
4	If Buff Empty = 0 Then Write Pointers RAM	

Microprogram for AMX 701/AMX 702

DOA/ Command

BUS STATE for AMX 701 /AMX 702

Time (shifting to)	DOB/Command	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7		
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg	Set Select Out Buffer = 1 Set Count Re Pointer = 0 Set Count Wr Pointer = 1 Set Read from Buffer = 0 Set Chan. Clear = 0
4	If Buff Full=0 Then Write Pointer RAM and Write Out Buffer	

Microprogram for AMX 701 /AMX 702

DOB/Command

POLL STATE for AMX 701/AMX 702

Time (shifting to)	Poll/Modem Status	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7	Set Poll Priority	
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg Write In Buff x) If Buff Full=0 Then Write Pointer RAM Set Dummy Done	Set Select Out Buffer = 0 Set Count Re Pointer = 0 Set Count Wr Pointer = 1 Set Read from Buffer = 0 Set Chan. Clear = 0
4		

x) If Buff Full = 0 Then Input to In Buff from Modem Reg

If Buff Full = 1 Then Input to In Buff is full Status

Microprogram for AMX 701 /AMX 702

Poll/Modem Status

BUS STATE for AMX 701 /AMX 702

Time (shifting to)	Poll/Re Data	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7	Set Poll Priority	
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg Write In Buff x) If In Buff Full = 0 Then Write Pointer RAM Reset Data Available Set Dummy Done	Set Select Out Buffer = 0 Set Count Re Pointer = 0 Set Count Wr Pointer = 1 Set Read from Buffer = 0 Set Chan. Clear = 0
4		

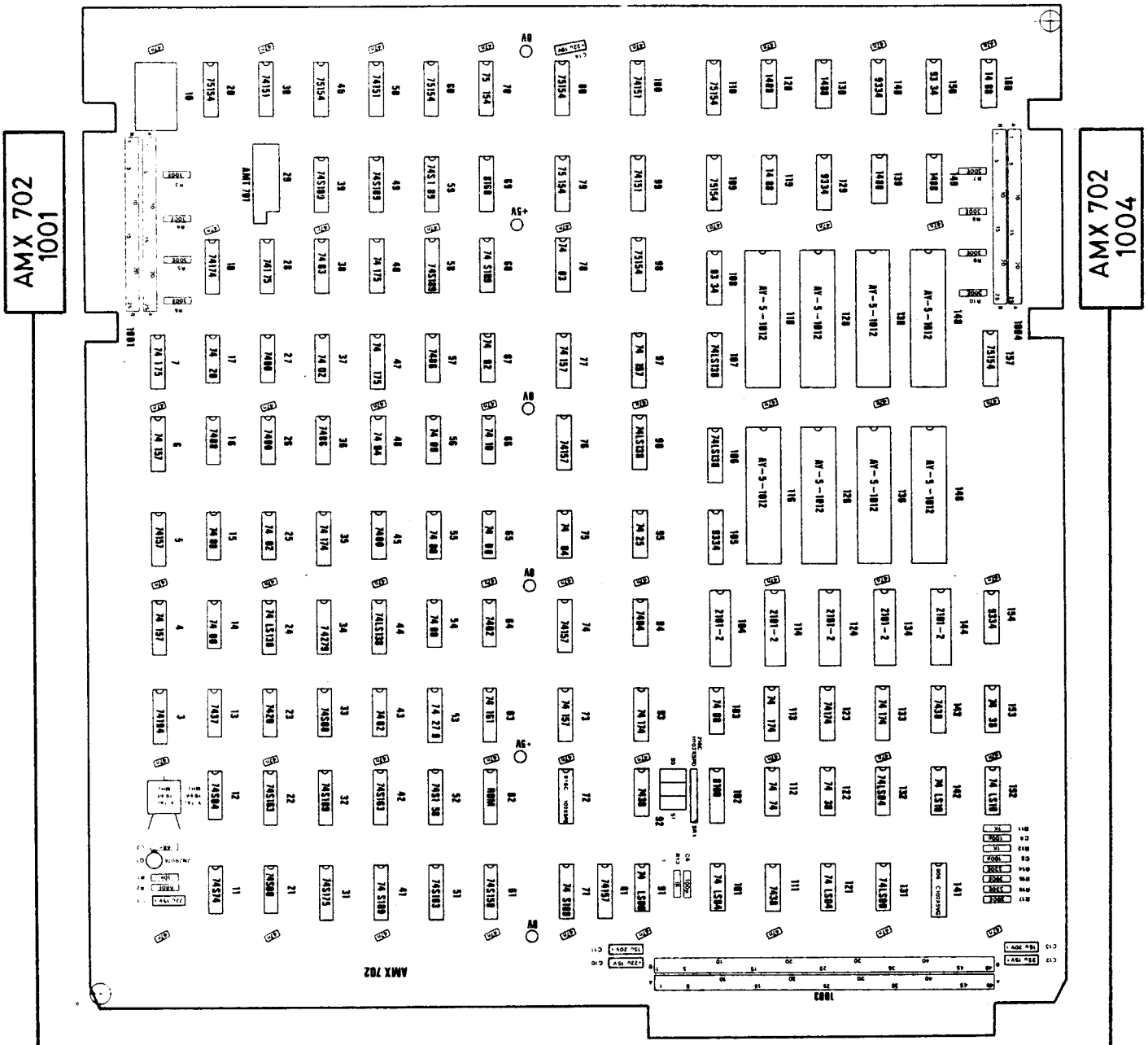
x) If Buff Full = 0 Then Input to In Buffer from UAR/T
If Buff Full = 1 Then Input to In Buffer is full Status

Microprogram for AMX 701/AMX 702

Poll/Re Data

POLL STATE for AMX 701/AMX 702

Time (shifting to)	Poll/Transmitter	
1.3	Set Receive-mode and Transmit-mode Register Set Modem signal Register	
1.7	Set Poll Priority	
2	Load Pointer Latch Load Buff Full Reg Load Buff Empty Reg	
3	Load DIA Status Reg If Buff Empty = 0 Then Write Strobe = 1 and Write Pointer RAM If Pointers Equal = 1 Then Dummy Done = 1	Set Select Out Buffer = 1 Set Count Re Pointer = 1 Set Count Wr Pointer = 0 Set Read from Buffer = 1 Set Chan. Clear = 0
4		


AMX 702
1001
AMX 702
1004
CBL 211
AMX 702-J1

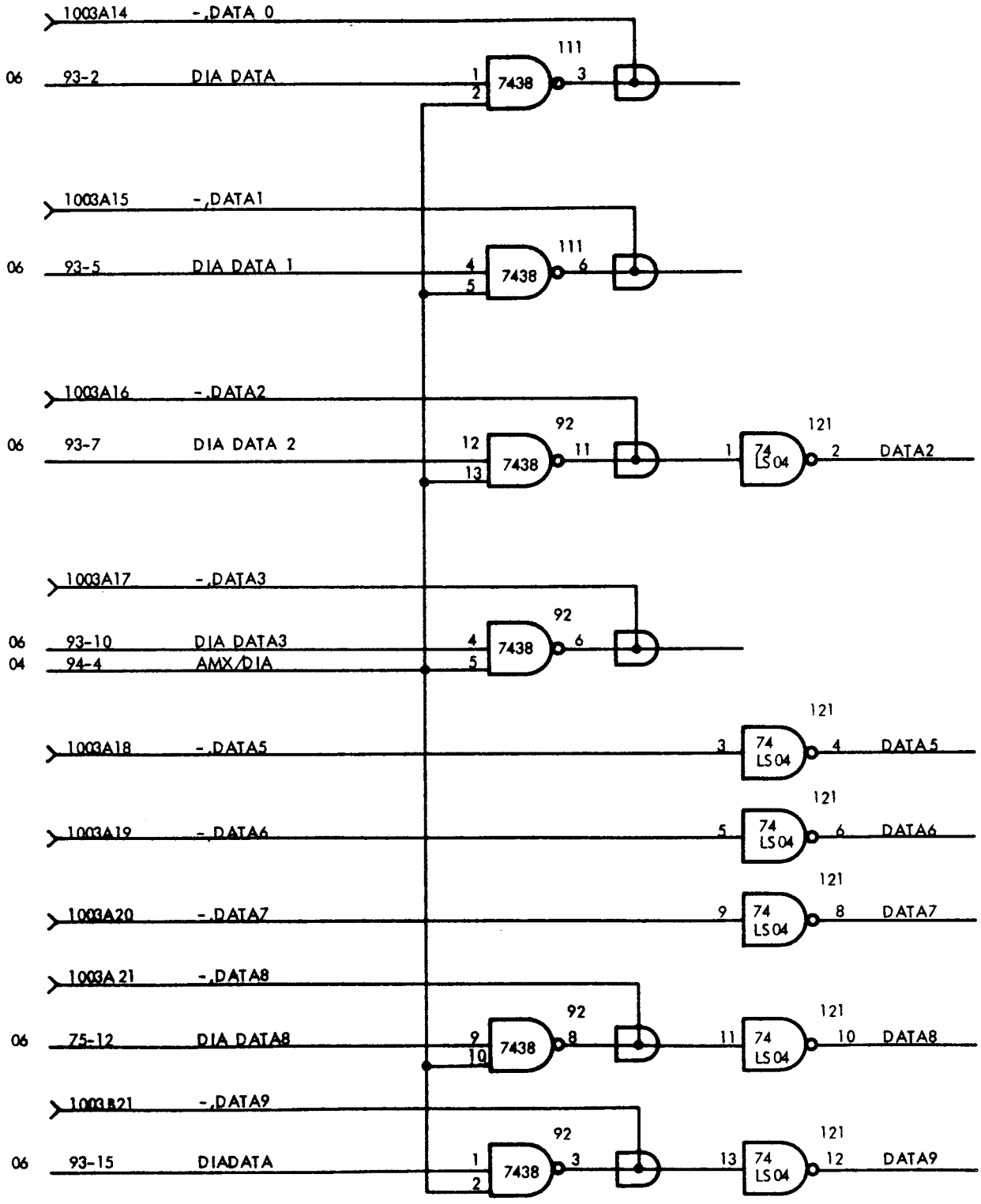
Connector AMX 702 1001	Gen. Addr.	Signal Name	Connector AMX 702 J1
A 1		Trans Data 0	1
B 1		OV	12
A 2		Received Data 0	3
B 2		OV	6
A 3		Req to Send 0	2
B 3		Ready for Sending 0	4
A 4		Data Set Ready 0	9
B 4		Carrier 0	11
A 5		Data Term. Ready 0	5
B 5		Calling Ind. 0	7
A 6		Power On 0	10
B 6		OV	8
A 7		Trans Data 1	13
B 7		OV	24
A 8		Received Data 1	15
B 8		OV	18
A 9		Req to Send 1	14
B 9		Ready for Sending 1	16
A10		Data Set Ready 1	21
B10		Carrier 1	23
A11		Data Term. Ready 1	17
B11		Calling Ind. 1	19
A12		Power On 1	22
B12		OV	20
A13		Trans Data 2	25
B13		OV	42

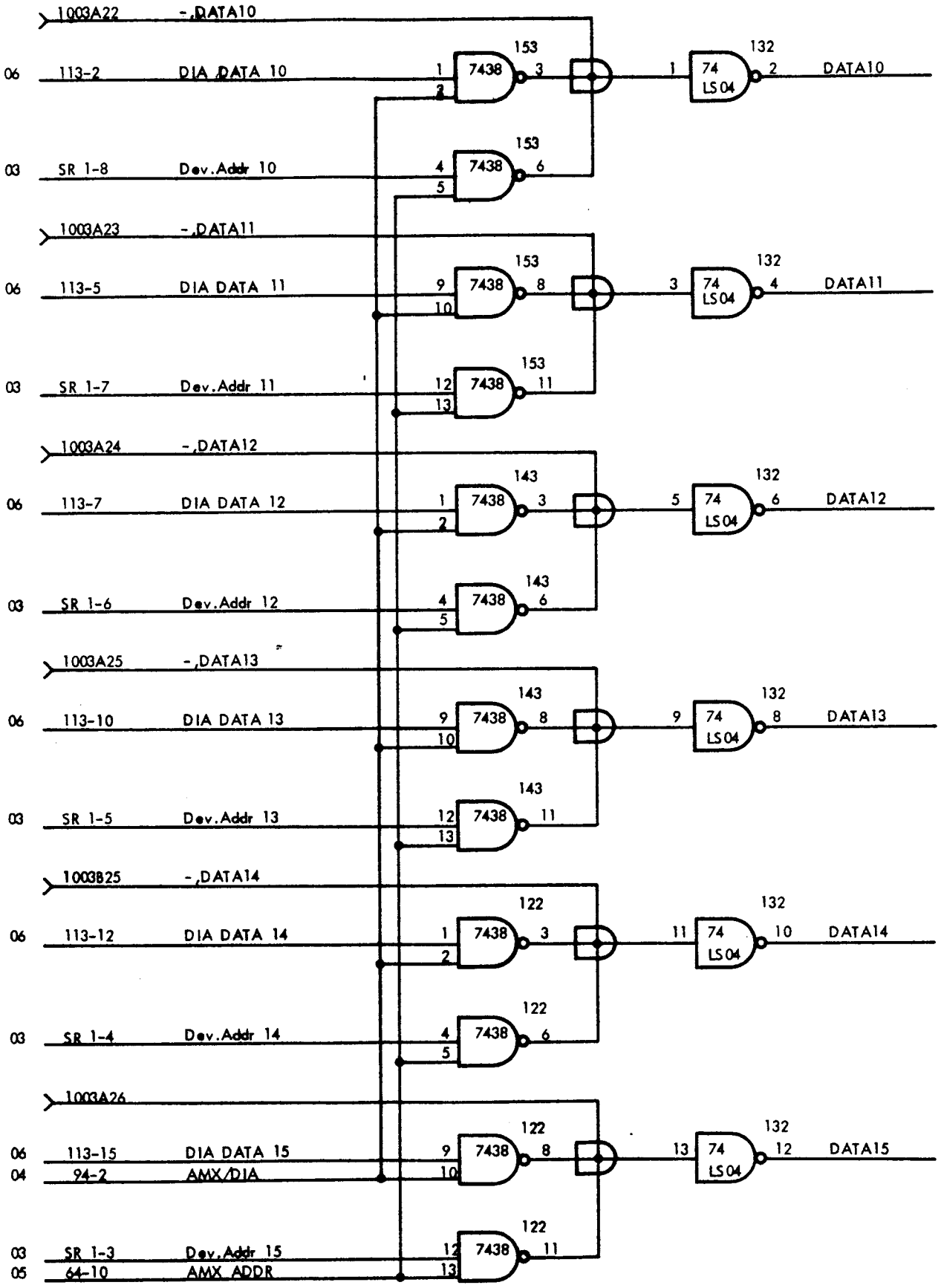
Connector AMX 702 1001	Gen. Addr.	Signal Name	Connector AMX 702 J1
A14		Received Data 2	51
B14		OV	48
A15		Req to Send 2	26
B15		Ready for Sending 2	50
A16		Data Set Ready 2	45
B16		Carrier 2	43
A17		Data Term. Ready 2	49
B17		Calling Ind. 2	47
A18		Power On 2	44
B18		OV	46
A19		Trans Data 3	41
B19		OV	30
A20		Received Data 3	39
B20		OV	36
A21		Req to Send 3	40
B21		Ready for Sending 3	38
A22		Data Set Ready 3	33
B22		Carrier 3	31
A23		Data Term. Ready 3	37
B23		Calling Ind. 3	35
A24		Power On 3	32
B24		OV	34
A25			
B25			

CBL 211

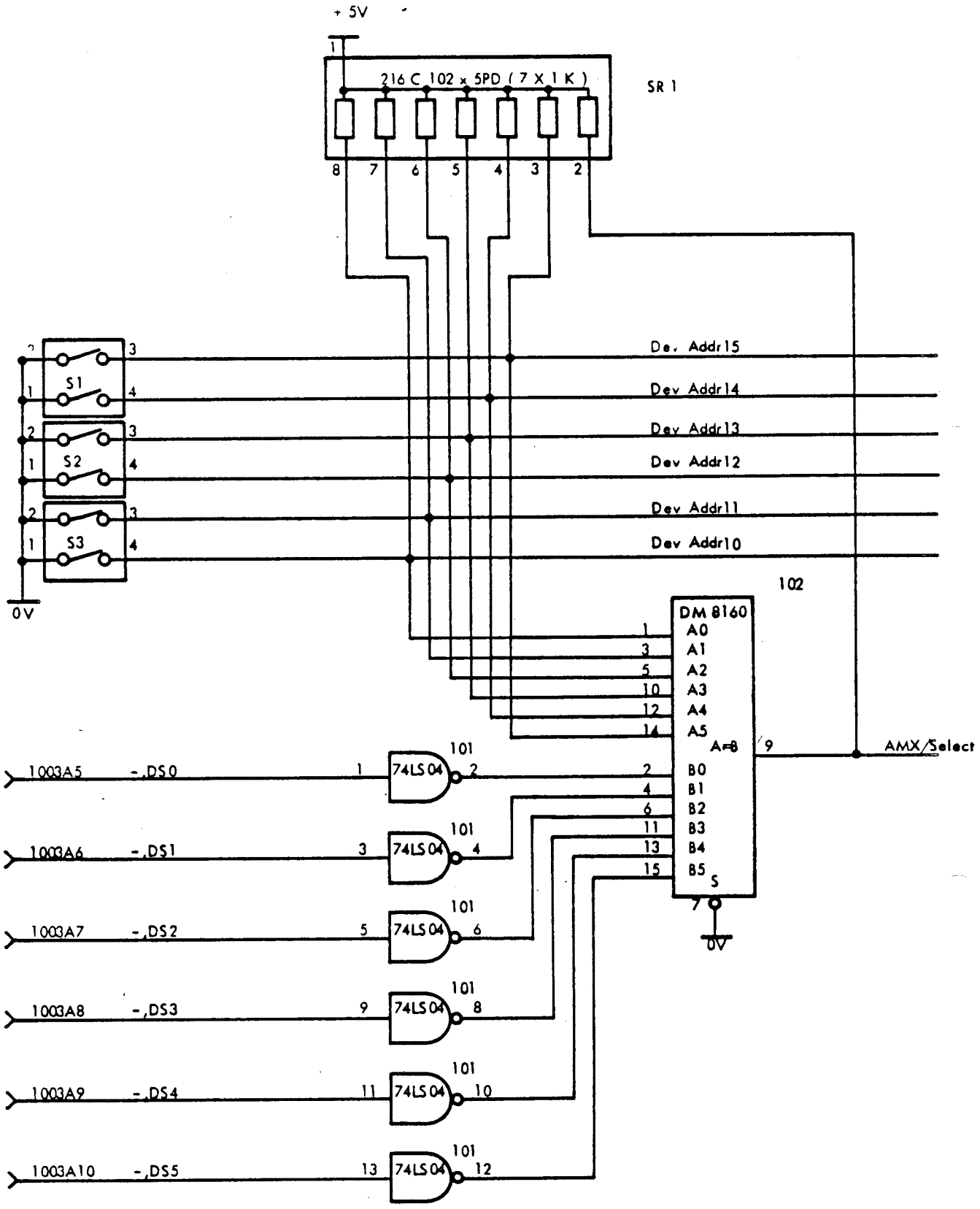
Connector AMX 702 1004	Gen. Addr.	Signal Name	Connector AMX 702 J1	Connector AMX 702 1004	Gen. Addr.	Signal Name	Connector AMX 702 J1
A 1		Trans Data 4	29	A14		Received Data 6	75
B 1		OV	60	B14		OV	98
A 2		Received Data 4	27	A15		Req to Send 6	74
B 2		OV	54	B15		Ready for Sending 6	100
A 3		Req to Send 4	28	A16		Data Set Ready 6	95
B 3		Ready for Sending 4	52	B16		Carrier 6	93
A 4		Data Set Ready 4	57	A17		Data Term. Ready 6	99
B 4		Carrier 4	59	B17		Calling Ind. 6	97
A 5		Data Term. Ready 4	53	A18		Power On 6	94
B 5		Calling Ind. 4	55	B18		OV	96
A 6		Power On 4	58	A19		Trans Data 7	91
B 6		OV	56	B19		OV	80
A 7		Trans Data 5	61	A20		Received Data 7	89
B 7		OV	72	B20		OV	86
A 8		Received Data 5	63	A21		Req to Send 7	90
B 8		OV	66	B21		Ready for Sending 7	88
A 9		Req to Send 5	62	A22		Data Set Ready 7	83
B 9		Ready for Sending 5	64	B22		Carrier 7	81
A10		Data Set Ready 5	69	A23		Data Term. Ready 7	87
B10		Carrier 5	71	B23		Calling Ind. 7	85
A11		Data Term. Ready 5	65	A24		Power On 7	82
B11		Calling Ind. 5	67	B24		OV	84
A12		Power On 5	70	A25			
B12		OV	68	B25			
A13		Trans Data 6	73				
B13		OV	92				

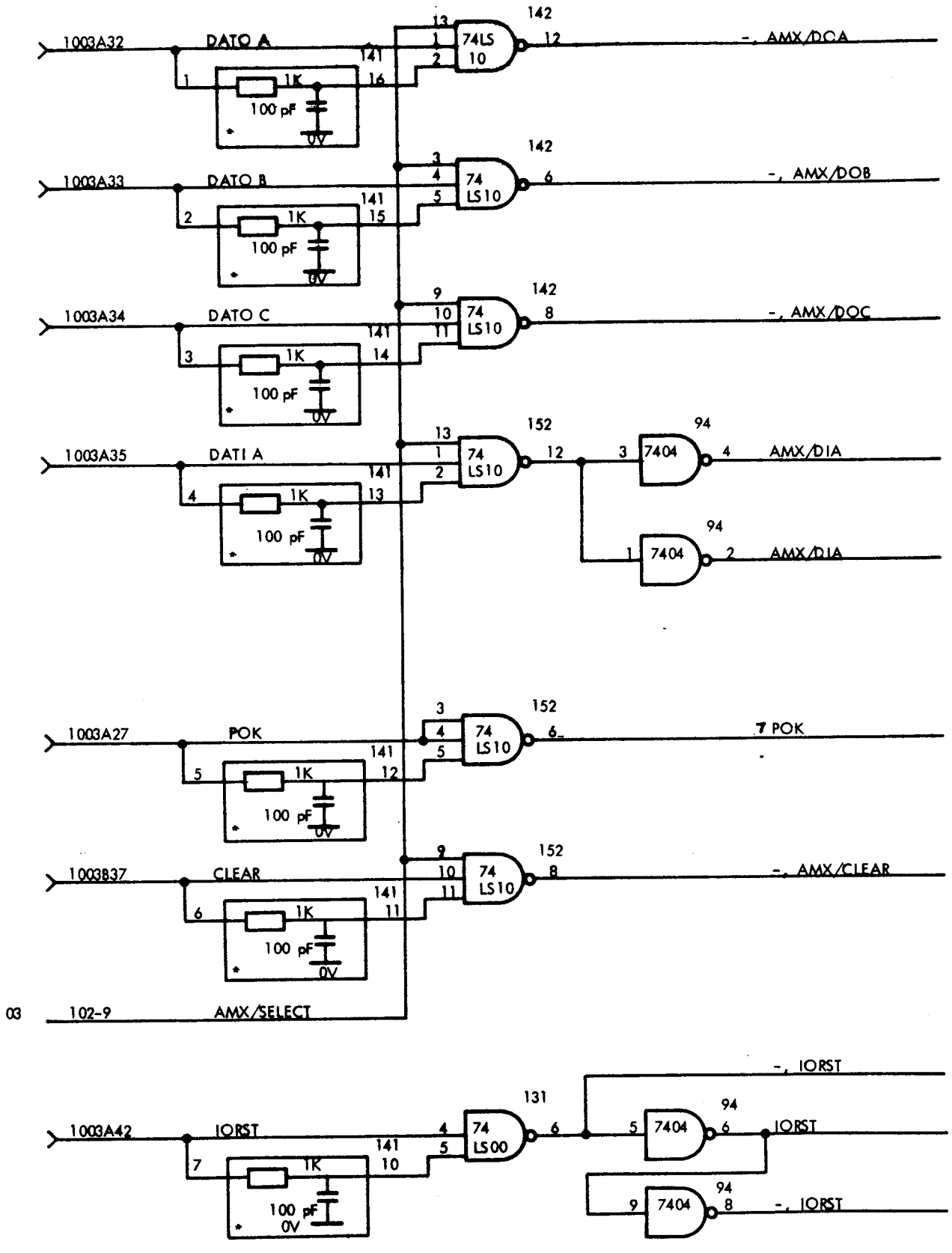
CBL 211

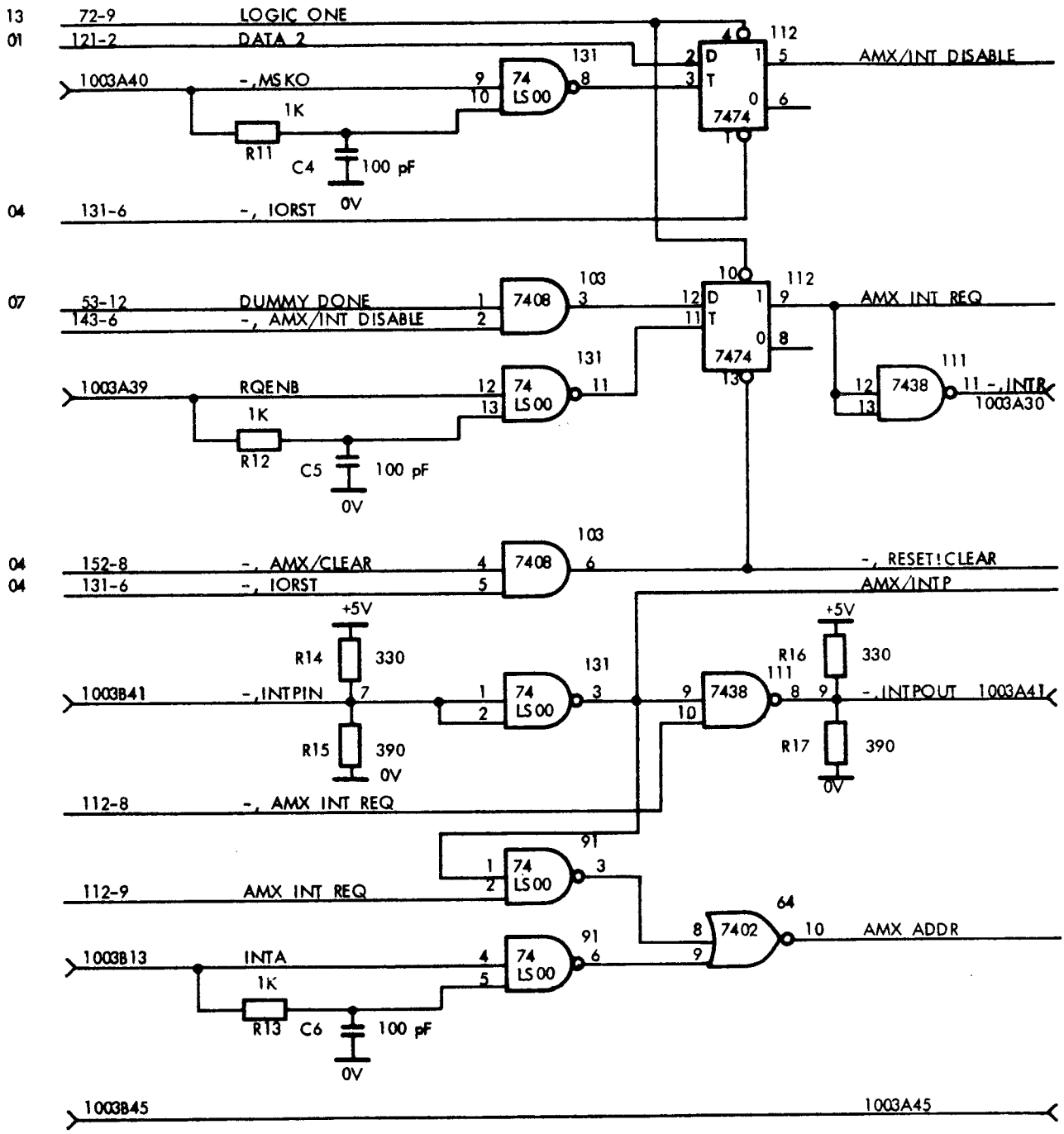

1003A14 -,DATA 0
06 93-2 DIA DATA
111
7438
1003A15 -,DATA1
06 93-5 DIA DATA 1
111
7438
1003A16 -,DATA2
06 93-7 DIA DATA 2
92
7438
121
74 LS04
DATA2
1003A17 -,DATA3
06 93-10 DIA DATA3
04 94-4 AMX/DIA
92
7438
1003A18 -,DATA5
121
74 LS04
DATA5
1003A19 -,DATA6
121
74 LS04
DATA6
1003A20 -,DATA7
121
74 LS04
DATA7
1003A21 -,DATA8
06 Z5-12 DIA DATA8
92
7438
121
74 LS04
DATA8
1003B21 -,DATA9
06 93-15 DIADATA
92
7438
121
74 LS04
DATA9

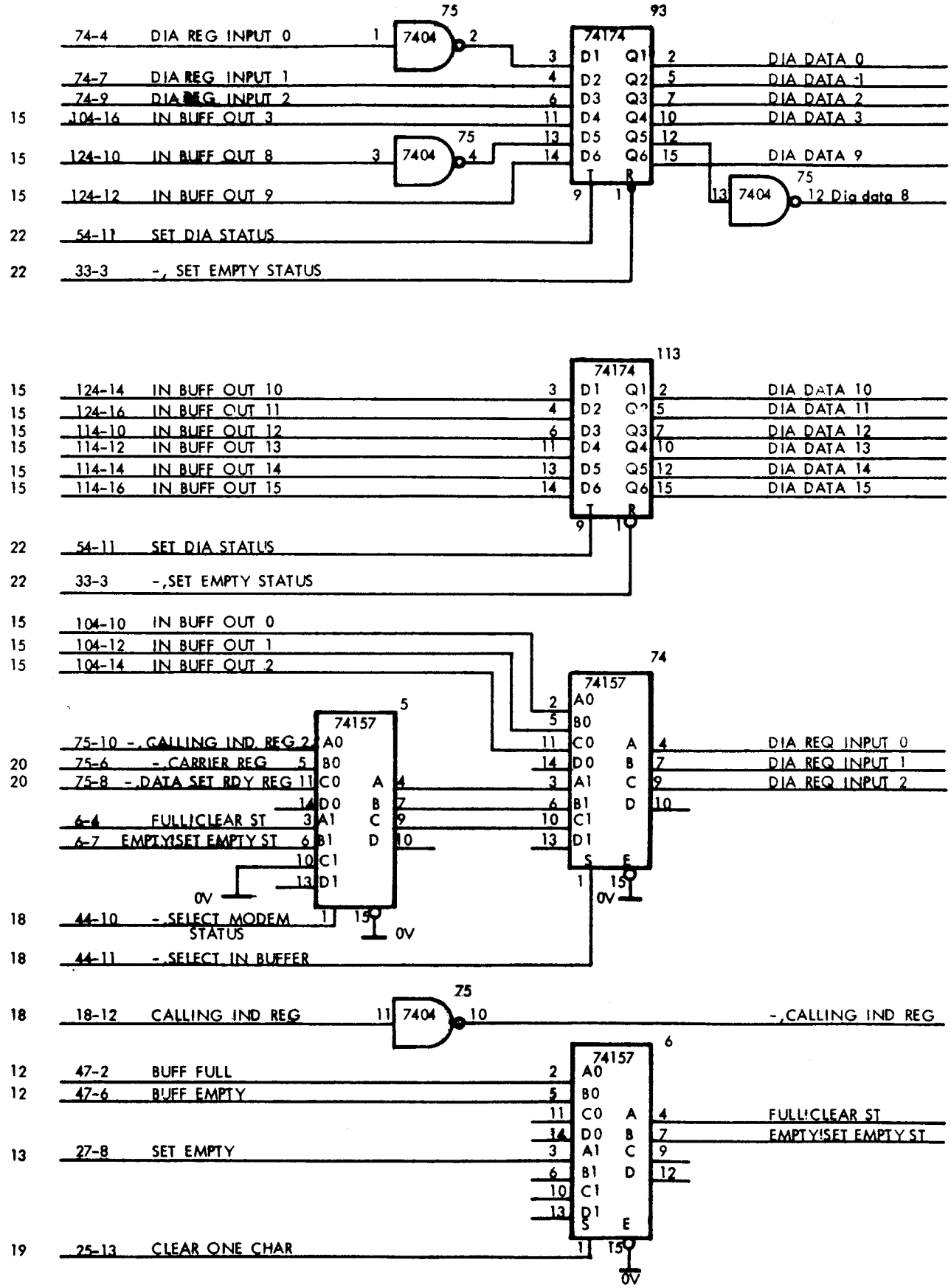


IN-OUT BUS DATA (10 : 15)
Logic diagram


+ 5V
216 C 102 x 5PD (7 X 1 K)
SR 1
8 7 6 5 4 3 2
S1
S2
S3
0V
Dev Addr15
Dev Addr14
Dev Addr13
Dev Addr12
Dev Addr11
Dev Addr10
102
DM 8160
A0
A1
A2
A3
A4
A5
A=B
B0
B1
B2
B3
B4
B5
S
AMX/Select
1003A5 -,DS0
1003A6 -,DS1
1003A7 -,DS2
1003A8 -,DS3
1003A9 -,DS4
1003A10 -,DS5
74LS04
101


1003A32
DATO A
1K
100 pF
0V
74LS 10
142
-, AMX/DOA
1003A33
DATO B
74 LS10
-, AMX/DOB
1003A34
DATO C
-, AMX/DOC
1003A35
DATI A
152
7404
94
AMX/DIA
1003A27
POK
7 POK
1003B37
CLEAR
-, AMX/CLEAR
03
102-9
AMX/SELECT
1003A42
IORST
74 LS00
131
-, IORST
IORST
* IC 906 C 102 X 5VG


13 72-9 LOGIC ONE
01 121-2 DATA 2
1003A40 -,MSKO
1K
R11
C4 100 pF
0V
04 131-6 -, IORST
74 LS00
7474
AMX/INT DISABLE
07 53-12 DUMMY DONE
143-6 -, AMX/INT DISABLE
7408
AMX INT REQ
1003A39 RQENB
R12 C5 100 pF
7438
-,INTR 1003A30
04 152-8 -, AMX/CLEAR
04 131-6 -, IORST
-, RESET!CLEAR
AMX/INTP
+5V
R14 330
R15 390
1003B41 -,INTPIN
R16 330
R17 390
-,INTPOUT 1003A41
112-8 -, AMX INT REQ
112-9 AMX INT REQ
7402
AMX ADDR
1003B13 INTA
R13 C6 100 pF
1003B45
1003A45

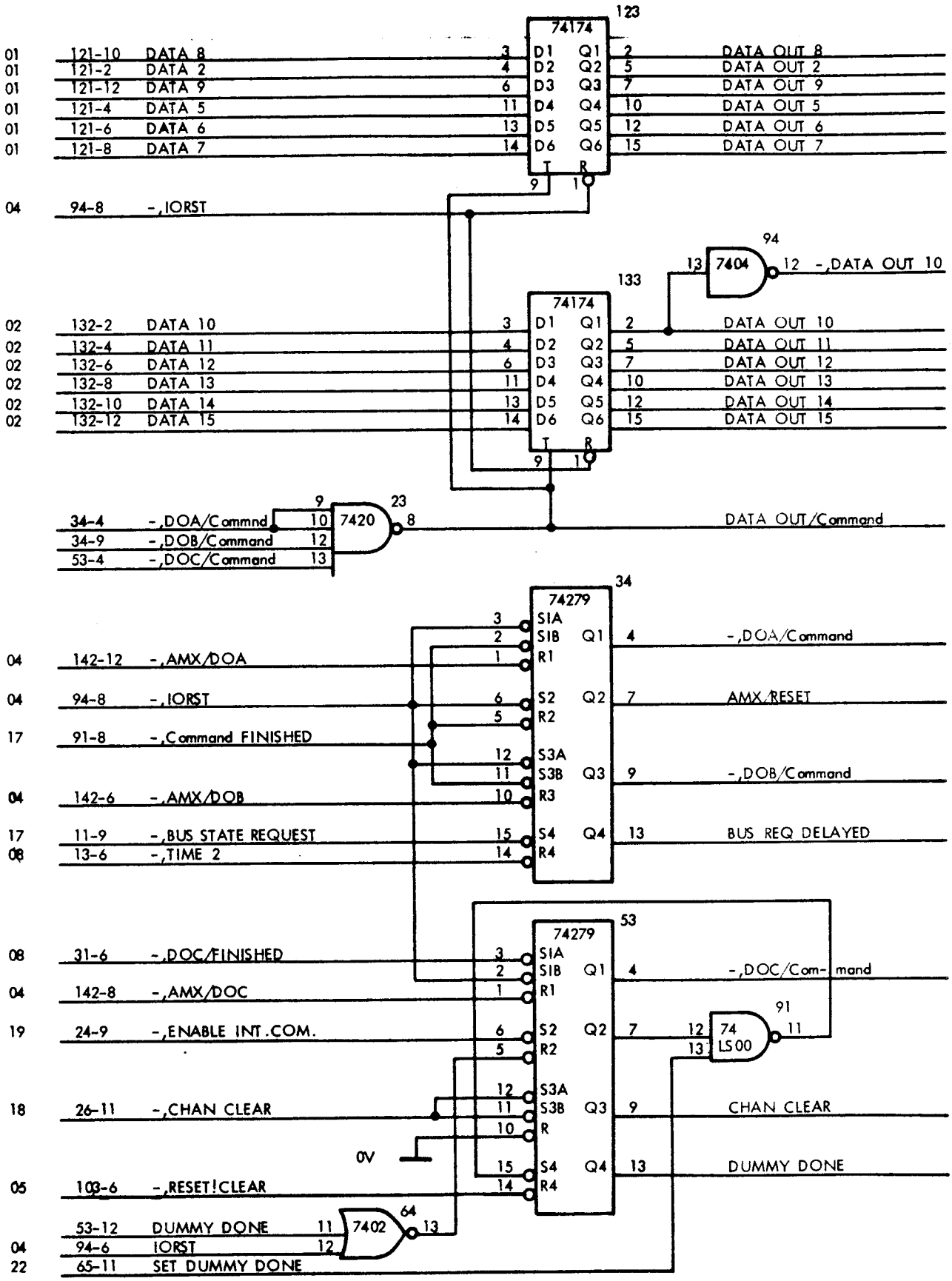


AMX 702 DATA IN REGISTER and SELECTOR

Logic diagram

AMX 06

R 12086

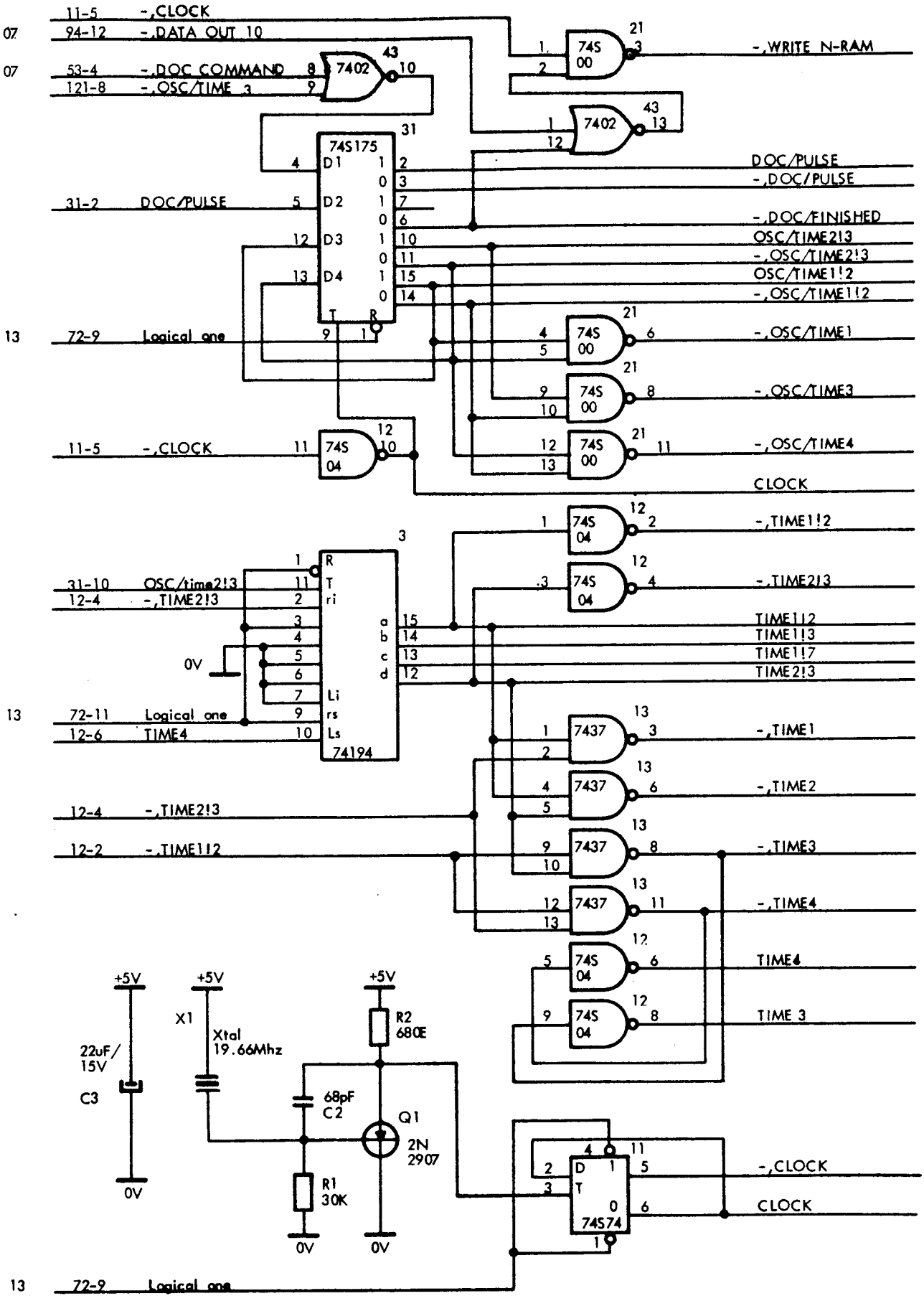


AMX 702

DATA OUT REGISTER and COMMAND REGISTER
Logic diagram

AMX 07

R 12087


11-5 -,CLOCK
07 94-12 -,DATA OUT 10
07 53-4 -,DOC COMMAND
121-8 -,OSC/TIME 3
7402
74S175
31-2 DOC/PULSE
13 72-9 Logical one
74S 00
74S 04
-,WRITE N-RAM
DOC/PULSE
-,DOC/PULSE
-,DOC/FINISHED
OSC/TIME2!3
-,OSC/TIME2!3
OSC/TIME1!2
-,OSC/TIME1!2
-,OSC/TIME1
-,OSC/TIME3
-,OSC/TIME4
CLOCK
-,TIME1!2
-,TIME2!3
31-10 OSC/time2!3
12-4 -,TIME2!3
0V
13 72-11 Logical one
12-6 TIME4
74194
TIME1!2
TIME1!3
TIME1!7
TIME2!3
7437
-,TIME1
-,TIME2
-,TIME3
-,TIME4
12-4 -,TIME2!3
12-2 -,TIME1!2
TIME4
TIME 3
+5V
22uF/ 15V
C3
X1
Xtal 19.66Mhz
R2 680E
68pF C2
Q1 2N 2907
R1 30K
74S74
-,CLOCK
CLOCK
13 72-9 Logical one