RC900

DCB451 Satellite Adapter

Hardware reference manual

PN: 99111115

PN: 99200389

RC Computer

Keywords:
DCB451, MF341, Satellite Adapter, RC940, RC900

Abstract:
This paper contains a hardware description of the DCB451 Satellite Adapter board. The DCB451 (MF341) is used as a satellite terminal when plugged into the base of an RC940 monitor and connected with an RC930 keyboard. The board also has a serial asynchronous controller for communicating with the RC950 system unit.

Date:
january 1988

Author:
Jens Peter Jakobsen

TABLE OF CONTENTS **PAGE**

TABLE OF CONTENTS (continued) **PAGE**

1. GENERAL INFORMATION

1.1 Introduction

The DCB451 is a 80186 based CRT display terminal controller that plugs into the base of an RC940 color monitor or a special version of the RC45 paperwhite monitor, RC45/900.

The board provides:

- 80186 CPU
- 256 K bytes System + Display RAM
- 64 K bytes Program ROM
- Keyboard interface (RC930)
- Serial interface
- Parallel printer interface
- Auto identification of monitor type

This manual describes the controller in three major sections.

- Short description and Specifications.
- Programming information.
- Hardware description with circuit- and timing diagrams.

1.2 Short Description

The DCB451 is based on a 80186 CPU which besides its normal CPU tasks handles the central part of the CRT-controller function. The main RAM of 256K bytes is used both as system RAM and as display buffer.

The remaining part of the CRT-controller consists of a number of discrete registers/latches, shift register, counter, multiplexer, two 16*6 bit Palette RAM's and a 8K byte static RAM character generator.

A dual UART, 2681 is used for serial interface to the keyboard (RC930) and to the Asynchronous V24 connection.

A parallel printer interface compatible to the RC750 interface (and the AT-interface at plug level) is also provided.

Two clock oscillators are provided on the controller board so that character rates for two different monitors can be supported. The clock oscillator is selected by a monitor type identification pin in the monitor connector.

A 48 pin EURO connector is provided for connecting a Debug Board, DBU451 to the controller.

1.3 Specifications

1.3.1 Performance Specifications

80186 clock	: 5.239 MHz (RC940 monitor) or 6.667 MHz (RC45/900 monitor) 7.65 MHz maximum
Data Bus Size	: 16 bits
EPROM	: 2-27256 (64 Kbytes) or 2-27512 (128 Kbytes)
RAM	: 256 Kbytes
Character Generator	: 8 Kbytes allowing 2 character sets of 256 characters
Palette	: Seperate foreground and background palette RAM's 16*6 bits
Character Cell Width	: 8 or 9 dots.
Character Cell Height	: 1 to 16 scan lines

1.3.2 Environmental Specifications

Operating Temperature : 10°C to 40°C
Relative Humidity : 20% to 85% (non condensing)
Max Wet-bulb temperature: 30°C
Altitude : 0 to 2000 m.

1.3.3 Physical Specifications

Width : 283 mm
Length : 317 mm
Height : 30 mm
Weight : 650 g

1.3.4 Power Specifications

Power Dissipation: max 11,5W
+5V Supply : +5 ±5%, max 2A
+12V Supply : +12V ±10%, max 65mA (120mA with keyboard)
-12V Supply : -12V ± 10% max 60mA

2. PROGRAMMING INFORMATION

2.1 80186 CPU

2.1.1 Interrupts

The 80186 interrupt controller is connected to the following external interrupt sources:

80186 INPUT	SOURCE
INT0	COMINT, SCN2681
INT1	KBINT , 2681 Keyboard Rx Rady
INT2	DBINT , Transmit interrupt from Debug board.
INT3	PRNINT, Parallel printer interrupt
NMI	RXINT , Receive interrupt from Debug board.

2.1.2 DMA Channels

The two integrated DMA channels are used as follows:

80186 INPUT	SOURCE
DRQ0	CRT ADAPTER, Horizontal retrace from Timer0 output.
DRQ1	Pin C6 in Debug connector, Unused.

For further information on how DMA channel0 is used see section 2.2.

2.1.3 Timers

timer0 and 1 are used by the CRT adapter for Horizontal and Vertical timing respectively. Programming information for these two timers is found in section 2.2.

Timer2 is for software use.

2.1.4 Memory Chip Selects

The six programmable Memory Chip Selects are used as follows:

Signal	Use
LCS :	Chip Select for the 256 Kbyte RAM.
MCS0:	Not used
MCS1:	Not used
MCS2:	Not used
MCS3:	Chip Select for the Program ROM on the Debug board.
UCS :	On board program ROM, normally 64 Kbytes, can be extended to 128 Kbytes.

2.1.5 DYNAMIC RAM Refresh

The 256 Kbyte dynamic RAM needs 256 refresh cycles each 4 ms.

The RAM control logic will automatically perform refresh cycles as long as the RAM is not accessed by the CPU.

One refresh cycle takes 4 character times i.e. 256 refresh cycles takes 1024 character times (2048 CPU clock cyles).

This means that the CPU should leave the RAM unaccessed (execute code from program ROM) for the following times each 4 mS:

CPU CLOCK FREQUENCY	TIME FOR 256 REFRESH CYCLES
5.239 MHz	0.391mS (RC940)
6.667 MHz	0.307mS (RC45/900)

2.1.6 Peripheral Chip Selects

The seven peripheral chip select lines are used as follows:

Chip Select Line	I/O Address Range (HEX)	
PCS0	0-07F	Not used
PCS1	80-0FF	Debug board
PCS2	100-17F	2681 Duart, COM + Keyboard
PCS3	180-1FF	CRT-mode register (write) mode + monitor type (read)
PCS4	200-27F	CRT lineaddress register
PCS5	280-2FF	Printer Data port (R/W)
PCS6	300-37F	Printer Status (R), control (W)

MX,EX Programming: MS = 0, EX = 1 (Peripherals mapped into I/O space and 7PCS lines available)

READY, WAIT STATE Programming:
R2-R0 bits in the PACS register should be programmed for 1 wait state inserted when PCS0-3 are asserted (R_2, R_1, R_0 = 0,0,1)

R2-R0 bits in the MPCS register should be programmed for 0 wait state inserted when PCS4-6 are asserted, external RDY also used. (R_2, R_1, R_0 = 0,0,0)

2.2 CRT Adapter

2.2.1 General Description

The attached monitor may be either monocrome (paper white) or color (RC940). Text display is supported for both types. To determine which kind is actually attached, the program may read bit 4 in the mode register (one = monocrome). The character clock frequency equal to half the CPU clock frequency is automatically adapted to the attached monitor type. This results in a slightly increased CPU performance in the monocrome case.

The CRT adapter supports text with characters 8 or 9 dots wide and from 1 to 16 dots high.

To conform with the monitor specs (line frequency), 80 characters plus framing must be displayed per row.

The CRT system fetches the text to be displayed, the character fonts and the palette contents in the memory, according to a scanline descriptor list in the memory.

The Sync pulse timing is controlled by the two timers in the CPU: Timer 0 drives the horizontal sync pulses and triggers DMA0 with the purpose of fetching a word in the line control table. Timer 1 drives the vertical sync pulses.

Through this description, the term scanline refers to the time for one entire horizontal scan period.

During every scanline a lineblock is transferred to the CRT adapter from the dual port main memory. The location and use of the lineblock is specified for each scanline in a list of scanline descriptors from which a word is fetched (by DMA0) in the beginning of the scanline.

A lineblock always starts at a 256 byte boundary and has a length in words equal to the number of characterclocks in a scanline minus the horizontal synctime.

The lineblock length is for

monocrome: 98 words

polycrome: 86 words

The scanlinedescriptor list may reside anywhere in the memory inclusive the ROM area and has a length in words equal to the number of scanlines in a frame inclusive vertical blanking.

DMA0 controls the access to the list, and should be reinitialized at the beginning of each frame.

The words in the scanline descriptor list have the following format:

bits 15,14 scanline function (LF)
    LF = 0 blank and load characterset 1
    LF = 1 blank and load characterset 0
    LF = 2 blank and load palette
    LF = 3 display text

bits 13 to 4 lineblock address (LBA)
    The start address of the line block equals 256*LBA.

bits 3 to 0 dotrownumber (DRN)
    The dotrow to be addressed in the character generator.

2.2.2 Text

The text is stored in lineblocks called textlines. Each textline is repeatedly accessed for each scanline showing the same character row only with different dot rows addressed in the charactergenerator.

This creates an easy way to display textlines with variable highth or to soft scroll.

The textline contains characters occupying one word each.

The character has the following format:
bits 7 to 0 value
    This is used together with the actual dotrow and optionally bit 11 as entry into the charactergenerator.

bits 11 to 8 foreground
    Used to select one of the 16 colours loaded into the foreground palette.

bit 11 alternative charset.
    If this option is selected (see mode register), bit 11 is used to select characterset. (It still selects foreground palette entry).

bits 15 to 12 Background
    Used to select one of the 16 colours loaded into the backgroundpalette.

The textline must contain characters for horizontal blanking (selecting black), and for coloured border:

Monocrome (RC45/900):
    character 0 to 9 : blank
    character 10 to 13 : left border
    character 14 to 93 : displayed text
    character 94 to 97 : right border
    character 98 to 99 : blank

Color (RC940):
    character 0 to 5 : blank
    character 6 to 89 : displayed text

2.2.3 Charactergenerator

The charactergenerator is a RAM with room for two charactersets each with 256 fonts (dot patterns). Each font describe the dotpattern of one character 8 (9 see below) dots wide and 16 dots high.

The value of a single dot selects the palette to be displayed in a pixel picture element). A one selects foreground and zero select background.

The fonts are transferred to the charactergenerator during scanlines with the linefunction field of the scanline descriptor equal to 0 or 1.

The lineblock contains the pattern of the dot row specified in DRN for a number of fonts in the characterset specified in LF. All words in the lineblock are transferred as dotpatterns.

The format for a dotpattern word is:

bits 7 to 0 charactervalue
bits 15 to 8 dot pattern
dot 1 = bit 15
dot 2 = bit 14
etc
dot 8 = bit 8
dot 9 is displayed either as zero or equal to dot 8. The last case requires that the feature is enabled and the char. value is between 192 and 223.

2.2.4 Palette

The palette is two 16 word by 6 bit RAM's one controlling the foreground colour and the other the background.

During display foreground is selected whenever the dot value coming out of the videoshiftregister is one.

Which foreground and background palette entry is to be used in the display of a certain character is specified in the character word.

The six output bits control the three colour beams in the polycrome CRT.

R and R* control the Red,
G and G* control the Green and
B and B* control the Blue

Each colour beam is controllable in four levels e.g. the red beam has the following apparent intensities:

```
R*,R = 1,1 : off
R*,R = 0,1 : 1/3
R*,R = 1,0 : 2/3
R*,R = 0,0 : full intensity
```

Especially the following palette outputs give the following colours on both monitortypes:

```
R*,G*,B*,R,G,B = 111111 : black
R*,G*,B*,R,G,B = 000111 : dark grey
R*,G*,B*,R,G,B = 111000 : white
R*,G*,B*,R,G,B = 000000 : intensified
```

The palettes are loaded by means of a lineblock with words of the following format:

bits 15 to 12 background
    adresses the background palette

bits 11 to 8 foreground
    addresses the foreground palette

bit 7 = 1: load background palette
bit 7 = 0: don't load background palette
bit 6 = 1: load foreground palette
bit 6 = 0: don't load foreground palette
bit 5 = R*
bit 4 = G*
bit 3 = B*
bit 2 = R
bit 1 = G
bit 0 = B

Note all words in the lineblock are used.

2.2.5 Timers

The integrated timers Timer 0 and Timer 1 in the 80186 CPU are used to generate syncpulses and to trigger the DMA accesses. Both timers should be programmed to alternate between maxcount A and B in a continous way.

The syncpulse is active when maxcount A is used.

Timer 0 generates horizontal sync and activates DMA 0 when the syncpulse goes active. Timer 0 should be programmed to internal clocking without prescaler. When the maxcount registers and the control register for timer 1 has been loaded, the control register for timer 0 (addr. 56H) should be programmed to C003H.

Timer 1 generates vertical sync. The input to timer 1 is connected to the output of timer 0. Timer 1 should be programmed for external clocking. The control register for timer 1 (addr. 5EH) should be programmed to E007H if interrupt is wanted otherwise to C007H. If interrupt is enabled, an interrupt is generated both on the front- and trailing edge of the vertical syncpulse.

The max count registers should be programmed in dependence on the attached monitortype:

Monochrome:

Timer	Max count	address	value (decimal)
0	A	52H	6
0	B	54H	49
1	A	5AH	7
1	B	5CH	408

Polycrome (350 line mode):

Timer	Max count	address	value (decimal)
0	A	52H	5
0	B	54H	43
1	A	5AH	13
1	B	5CH	356

DMA 0 is triggered by the front edge of the horizontal syncpulse and is supposed to fetch a word in the scanline descriptor list and write it into I0 address 200H. DMA 0 should be programmed to have highest priority.

DMA0 should be reinitialised at the beginning of each frame. This may be done using the timer 1 interrupt and the WAIT-instruction.

In this configuration the WAIT-instruction always stops the programexecution until the beginning of the next scanline, giving the software maximum time for manipulation of DMA0.

To compensate for possible disable time, the algorithm should be:

1: wait; (until the beginning of next scanline).
2: read timer 1 count register; (addr. 58H)
3: add to address of first descriptor;
4: write into DMA0 address register; (addr. C0H)

The second scanline during vertical sync is the scanline controlled by the first word in the scanline descriptor list.

2.2.6 Cursor, Blink and underline

The attributes Blink and underline are supported in software.

The cursor is simulated in software.

2.2.7 Mode register

On IO address 180H is located the mode set register which, with the exception of Bit 4, is both write- and read-able.

Bit 0 Dot 9 en.
When this bit is set and bit 1 is cleared and the displayed character value is in the range C0H to DFH, the ninth dot is displayed equal to the eighth. Otherwise if displayed it is zero.

Bit 1 640
If this bit is clear, 9 dots will be displayed per character.
If it is set 8 dots will be displayed per character.
Some time is required for the phaselocked dot oscillator to readjust after alteration of this bit before accesses to main memory are safe. Due to this it is necessary to execute code in ROM for app. 100 microseconds after alteration of bit 1.

Bit 2 H.Pol.
If this bit is set, the horizontal syncpulses are active low.

Bit 3 V.Pol.
If this bit is set, the vertical sync pulses are active low.

Encoding of Bit 2 and Bit 3, Sync. Polarity:

Monitor/mode	H.POL	V.POL	BIT2	BIT3
RC940/350 lines	POS	NEG	0	1
RC940/400 lines	POS	POS	0	0
RC940/480 lines	NEG	POS/NEG	1	X
RC45/900	POS	POS	0	0

Bit 4 Read: Polycrome
If this bit is a zero, a monocrome monitor is attached to the DCB.

Write: Reverse Contrast Control
This bit must be zero normally, if set then the function of the contrast +/- keys on the keyboard is reversed.

Bit 5 Alt chset en.
If this bit is set, bit 11 in a character selects the alternative character set.

Bit 6 Chset sel
If this bit is zero, characterset 0 is selected as the primary and characterset 1 the alternative.
If the bit is one, characterset 1 is selected as the primary and characterset 0 the alternative.

Bit 7 Hide en. (Hide enable)
If this bit is one, the characters having bit 15 = 1 are hidden (displayed as background only).

After reset all the bits 0 to 7 except bit 4 will be zero.

Bits 8 to 15 are not supported i.e. they give undefined results on read.

2.3 Serial Interface

The serial interface to keyboard and V24 line is provided by a dual full-duplex asynchronous receiver/transmitter (DUART) SCN2681. This controller also contains a 7 bit multifunction input port (IP0-IP6), an 8 bit multifunction output port (OP0-OP7) and a programmable counter timer.

The DUART contains 16 I/O registers at the following addresses:

2681 REGISTER ADDRESSING

ADDRESS	READ	WRITE
100	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
102	Status Register A (SRA)	Clock Select Reg. A (CSRA)
104	*Reserved*	Command Register A (CRA)
106	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
108	Input Port change Reg. (IPCR)	Aux. Control Register (ACR)
10A	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
10C	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
10E	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
110	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
112	Status Register B (SRB)	Clock Select Reg. B (CSRB)
114	*Reserved*	Command Register B (CRB)
116	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
118	*Reserved*	*Reserved*
11A	Input Port	Output Port Conf. Reg. (OPCR)
11C	Start Counter Command	Set Output Port Bits Command
11E	Stop Counter Command	Reset Output Port Bits Command

For further information on the 2681 DUART see ref. 2.

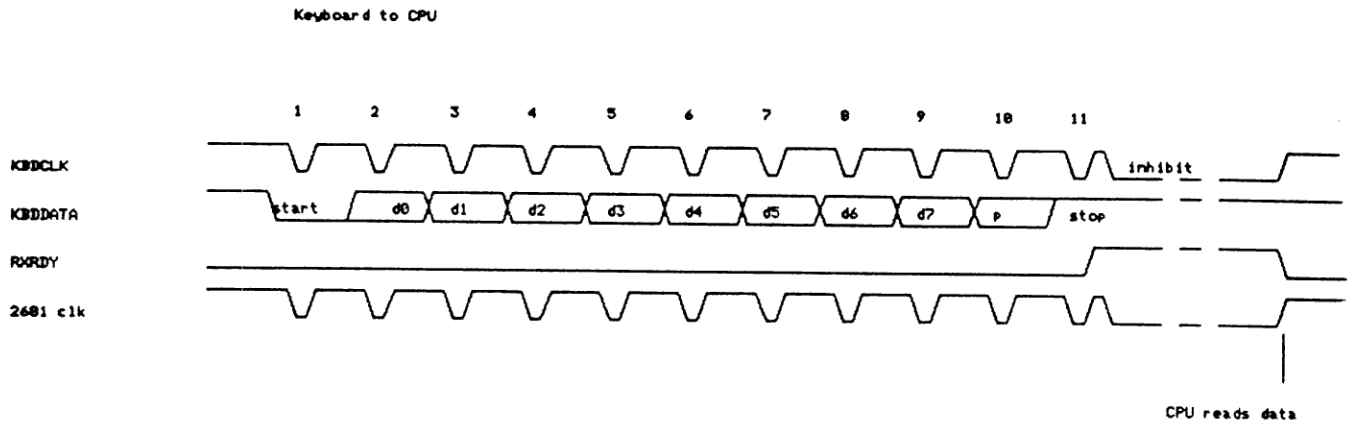
2.3.1 Keyboard Interface

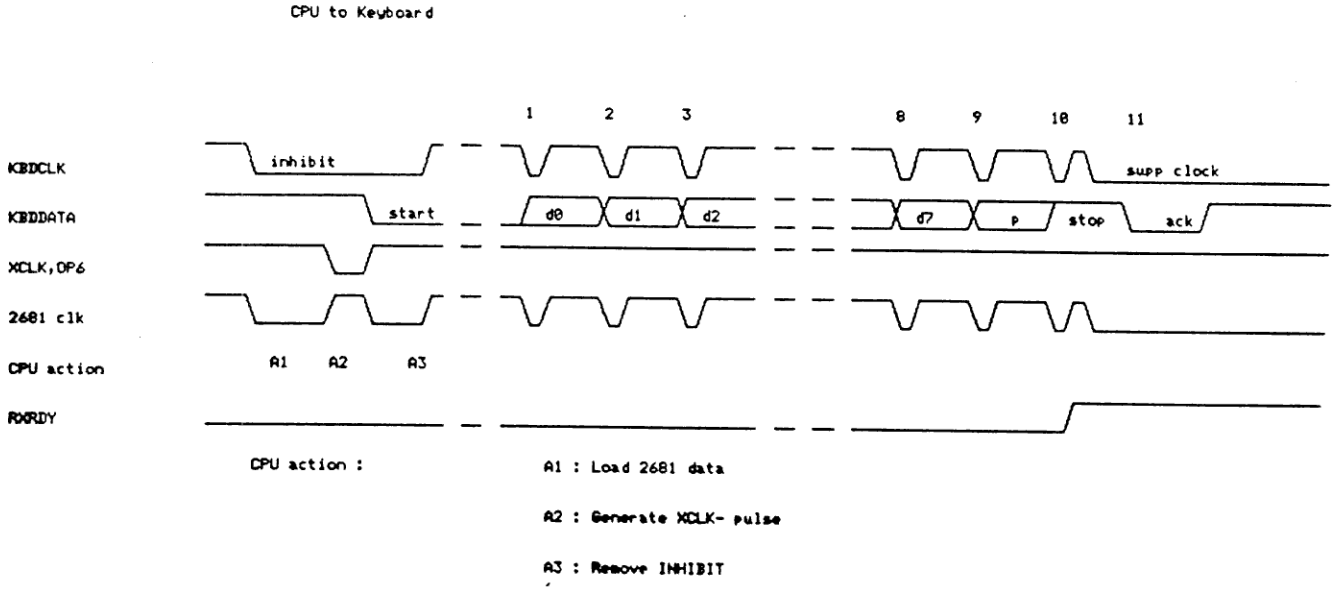
The 2681 B-channel is used for handling the serial interface to a RC930 keyboard.

The Receiver and Transmitter are both clocked externally by the keyboard clock line (KBDCLK), Channel B clock Select Register (CSRB) should be programmed to all ones.

The Receiver and Transmitter should be initialized for 1 startbit, 8 databit, odd parity and one stopbitt. OPCR bit 5 must be 1 (OP5 = RXRDY).

The bidirectional data transfer on the two keyboard lines KBDCLK and KBDDATA is handled as shown in the following timing diagrams.





2.3.1.1 Keyboard to CPU

To receive data from the keyboard the CPU only has to monitor the RXRDY flag in the 2681 either by polling or interrupt.

RXRDY indicates that the keyboard has clocked a data byte into the 2681 receiver.

The KBDCLK line will automatically be held low as long as RXRDY is active thereby inhibiting futher transmissions from the keyboard until the CPU has read the data register in the 2681 receiver.

2.3.1.2 CPU to Keyboard

Transmission from CPU to keyboard requires the data flow direction to be reversed. This is signalled to the keyboard by the CPU holding both the KBDCLK and KBDDATA lines low.

Before doing so it should be verified that the keyboard is not in the middle of clocking a data byte to the receiver. This is ensured by first checking that no KBDCLK's has been received, IP(4) (RXACTIVE INPUT) should read 0 and that a full data byte has not been received, RXRDY should also be 0.

Next the KBDCLK line should be forced low (keyboard INHIBIT) by resetting bit 4 in 2681-OPR. Ensure that the 2681 Receiver data register is empty and that the receiver is ready to readback what is transmitted.

Generate "XCLK", an extra clock pulse only seen by the 2681, by setting bit6 in OPR and resetting it again. This clocks the start bit out of the transmitter thereby forcing the KBDATA line low.

Wait 60 mS and then remove INHIBIT by setting bit 4 in OPR.

Now the keyboard is free to clock data out of the 2681 transmitter. When the stop bit has been clocked out the 2681 receiver will set RXRDY and INHIBIT KBDCLK thereby suppressing the "ACK-clock" from the keyboard. INHIBIT should be maintained until the ACK signal has been removed by the keyboard to avoid this being seen as a start bit by the receiver.

2.3.2 V24 Interface

The 2681 A-channel is used for a V24 Asynchronous interface using a 9 pin connector.

Baud rates can be selected internally between 75 and 19200 baud.

The receiver and Transmitter is connected to the 9 pin connector through line-receiver and driver. In addition there is four modem signal inputs and 2 modem signal outputs connected to IP0-IP3 and OP0, OP2 respectively, see section 2.3.4.

2.3.3 Sound

A small Piezoelectric sound element is connected to the Counter/timer output (OP3).

A suitable counter source is the crystal clock (3.6864 MHz) divided by 16 (ACR-bits 6 to 4 = 0,1,1).

The sound is turned on and off by OPCR bits 3 and 2. Bit 3 must always be 0, then bit 2 ="1" turns on, and bit 2 = "0" turns off the sound.

Two volume levels of the sound can be selected by OPR bit 7, a "0" selects HIGH and a "1" selects LOW volume.

2.3.4 IP/OP Registers

The 7 input lines IP0-IP7 can be read at the input port at I/O address 11A hex.

Bit	Description
0	-CTS input (Pin 8)
1	-CI input (Pin 9)
2	-DSR input (Pin 6)
3	-CD input (Pin 1)
4	Keyboard RX ACTIVE. Low as long as -RXRDYB (OP5) is active low (as long as KBDCLK is held low by the 2681). Otherwise RXACTIVE goes high on the first KBDCLK high to low transition.
5 and 6	readback of KBDCLK line (A "1" is read when KBDCLK is high or XCLK (OP6) is low)
7	Always "1".

The 8 output lines OP0-OP7 are configured by writing to the OPCR (output port conf. register) at I/O address 11A hex as follows:

Bit	Value	Description
1,0	0,0	OP2 = -OPR(2)
3,2	0,0	OP3 = -OPR(3), Sound OFF
	0,1	OP3 = C/T output, Sound ON
4	0	OP4 = -OPR(4), -INHIBIT
5	1	OP5 = -RXRDY
6	0	OP6 = -OPR(6), -XCLK
7	0	OP7 = -OPR(7), -VOLUME

The output port register OPR is used as follows:

Bit	Description
0	RTS, V24 signal
1	0, Not used
2	DTR, V24 signal
3	0, Not used
4	-INHIBIT, a 0 forces the KBDCLK line low.
5	0, Not used
6	-XCLK, generates local RX/TX clock pulses.
7	-VOLUME, 0 = High, 1 = Low volume.

2.4 Parallel Interface

An overview of the electrical signals used in the interface is shown below.

Pin number	Name
1	-STROBE
2-9	8 data bit
10	-ACK
11	BUSY
12	PE
13	SLCT
14	-UTOFDXT
15	-ERROR
16	-INIT
17	-SLCTIN

The interface consists of 4 registers.

- Data output register, directly controlling the data pins if enabled.

- Data input port, reflecting the state of the data pins at the time of reading.

- Control output register, directly controlling four control output pins and enabling of data output register and interrupt.

- Status read port, reflecting the state of the 8 control/status pins at the time of reading.

The registers have the following layouts:

Data output register (OUT 280H):

Bit	Connector Pin no.	Description
0	2	If the output register is enabled
1	3	(i.e. control register, bit 4 = 0),
2	4	then a bit in the register directly
3	5	controls the corresponding connector
4	6	pin as follows:
5	7	
6	8	
7	9	

Bit state	TTL output
0	LOW
1	HIGH

Data input port (IN 280H):

Bit	Connector Pin no.	Description
0	2	Read back of data output register,
1	3	or if this is disabled the state of
2	4	the connector pins.
3	5	
4	6	
5	7	
6	8	
7	9	

Pin TTL level	Bit state read
LOW	0
HIGH	1

Control register (OUT 300H):

Bit 0-3 of this register are connected through open collector inverts to corresponding connector pins (all four having pull up resistors to +5V).

Bit	Signal (Pin no.)	Description
0	-STROBE (1)	See above
1	-AUTOFDXT (14)	See above
2	-INIT (16)	See above
3	-SLCTIN (17)	See above
4	OUT DISABLE	Output register disable if 0: enables output register line drivers if 1: three-states the output register and allows pins 2-9 to be used for inputs.
5		NOT USED
6		NOT USED
7	INT DISABLE	Interrupt disable if 0: enables interrupts when BUSY input pin (11) is LOW. if 1: disables interrupts.

Status input port (IN 300H):

Each bit in this port represents the inverse state of a pin in the connector. The 5 LSB are inputs only while the 3 MSB inputs the state of 3 of the open collector outputs.

Bit	Connector Pin no.	Description
0	11	NOT BUSY, 0 when input signal BUSY is high.
1	10	ACK, 0 when input signal is high.
2	15	ERROR, 0 when input signal is high.
3	12	NOT PE, 0 when input signal is high.
4	13	NOT SLCT, 0 when input signal is high.
5	1	STROBE, 0 when I/O signal is high.
6	16	INIT, 0 when I/O signal is high.
7	17	SLCTIN, 0 when I/O signal is high.

3. TECHNICAL DESCRIPTION

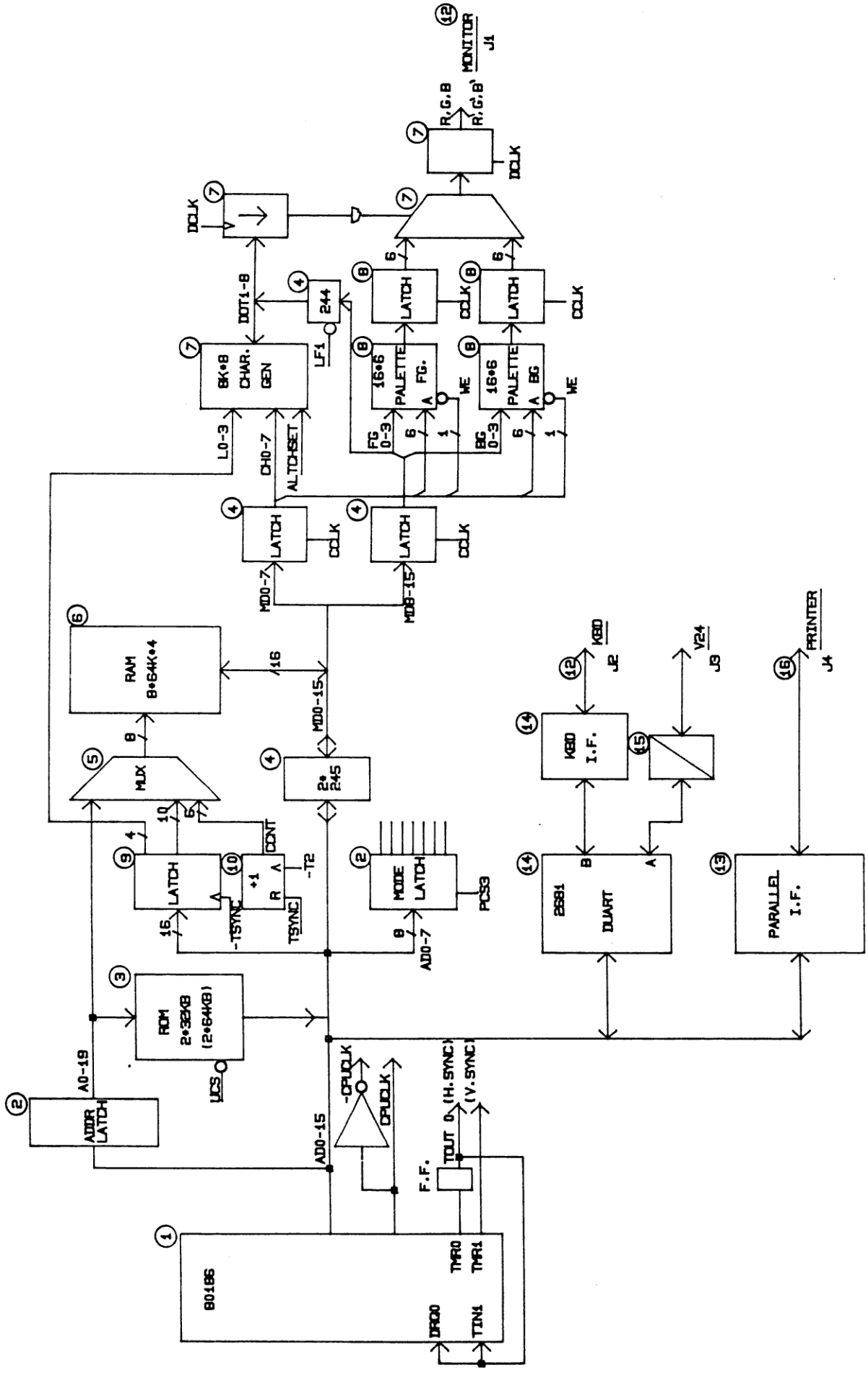
3.1 Introduction

This chapter contains a technical description of the DCB451 hardware.

The following block diagram shows the data paths and some of the most important signal names. Numbers in circles refer to a diagram page number in section 3.3. where the detailed circuit diagram can be found.

The following technical description refer to these diagrams page numbers.

3.1.1 Block Diagram



3.2 Short Technical Description

Diagram page 1 shows the 80186 CPU.
Timer 0 output is resynchronized to -T3 at half the character clock rate.

Diagram page 2 shows the address latches and the CRT mode register. All outputs of this register except bit 4, REVCONT (Reverse contrast function) can be read back via the buffer at the bottom of the page. Bit 4 reads the monitor identification signal -MONOCROM.

Diagram page 3 shows the two program ROM's which are placed at the top of the CPU address range.

A buffer, U48 enables readback of the instantantaneous data on the DOT1-8 bus.

Diagram page 4 shows the memory data bus buffers which allow the CPU to access the ROM's and I/O devices without waiting for a CPU access cycle in the CRT/CPU memory access sequence.

The two 8 bit registers receive a 16 bit character + attribute from the RAM at each character clock, CCLK.

LF1 = 0 enables pixel data to be loaded into the character generator.

Diagram page 5 shows the dynamic RAM address multiplexer. T1 selects between ROW and Column address, T2 selects between CRT or CPU address, see table on the diagram page.

Diagram page 6 shows the dynamic RAM which is divided in two banks, A and B. Each bank has its own set of control signals. (For example WEA and WEB).

Diagram page 7 shows the 8 Kbyte RAM character generator and the video shift register. The two data selectors 74AS157 selects between foreground and background color at dot rate. The selector inputs may change at character boundarys only.

The 6 bit color information is resynchronized by an octal D-register, to minimize skew between the viedo signals.

Diagram page 8 shows the seperate foreground and background palette RAM's and the pipeline registers.

HSYNC is delayed 2 character clocks with respect to TSYNC. From these signals a horizontal sync pulse for the monitor is generated. This pulse is two character clocks shorter than the internal TSYNC.

Diagram page 9 shows the Line Start Address registers that are loaded by help of the 80186 DMA channel 0 at the start of each scanline.

LSA0-9 address the Dynamic RAM together with the Character counter outputs CCNT 0-6 (diagram page 10).

LF0 and LF1 determine the scanline function:

LF1	LF0	Function
0	0	Blank display and load characterset 1
0	1	Blank display and load characterset 0
1	0	Blank display and load Palette RAM(s)
1	1	Display text.

ALTCHSET selects which characterset is used for display or which characterset is to be loaded.

DISP(lay) enables the video output (inverse of blank). Display is disabled during horizontal sync (TSYNCD) and when loading charactergenerator or palette RAM's.

WFG, WBG are write foreground and write background palette write enable pulses respectively.

DOT 9 is the rightmost dot in a charactercell when displaying 9 dots/char. It will be always 0 (background) when DOT9EN is 0. When DOT9EN is 1 then DOT9 will be equal to DOT8 in some of the characters in the charactergenerator (char. value 192-223) for all other characters DOT9 will be 0.

WCHARSET is the writepulses for the character generator.

ARDY is Asynchronous ready to the 80186. When ARDY and SRDY are both low the 80186 inserts wait-states until one of these signals goes high. CPU Accesses to RAM lowers -LCS and thereby ARDY is held low until the memory scquencer (PAL's on page 10) pulses SRDY high.

An IOWRITE (by DMA0) to the LSA register (PCS4) during TSYNC will also lower ARDY and extend the I/O write until the end of the TSYNC pulse at which time data will be latched in the LSA register.

HSYNC OUT and VSYNC OUT are the synchronization signals for the monitor.

Diagram page 10 shows the character clock timing counter dividing the dotclock DCLK by 8 when the signal 640DOTS is high, otherwise it divides by 9, thereby selecting the character cell width. LDVSR, T2, T1, 50 Counter state outputs are used by the memory timing PAL's, see PAL-descriptions in sect. 3.4 and timing diagrams in sect. 3.5.

CCNT 0-6 are the character counter outputs, counting at character rate, starting from zero at the beginning of each scanline.

Diagram page 11 shows the CPU clock X-tal oscillator and the phase/frequency comparator keeping the character clock positive going edge coincident (within a few nS) with each second positive going edge of -CPUCLK.

The CPU clock is generated by one of two selectable X-tal oscillators. The monitor identification signal MONOCROM determines which oscillator is operating and connected to the clock amplifier and buffer converting the ECL signal to TTL level.

The last (third) ECL line receiver of the 10116 is used as a Voltage Controlled Oscillator generating the Dotclock DCLK.

The frequency of the Dotclock will be either 8/4 or 9/4 of the CPU X-tal oscillator, depending on how many dots/char is selected in the mode register.

Diagram page 12 shows the 32 pin DIN Monitor connector and the associated Video signal and Power EMI filters.

Also shown is the keyboard connector and Contrast/Brightness signal buffers. Contrast and Brightness signals are both 200 Hz signals generating analog control voltages to the monitor proportional to their duty cycles.

The keyboard power is derived from the +12V supply by a small 9V regulator serving as a current limiter in case the keyboard connector should be shorted.

Diagram page 13 shows the parallel printer interface.

Diagram page 14 shows the 2681 DUART and the keyboard interface.

A quad comparator LM339 is used as drivers and receivers on the KBDCLK and KBDDATA lines, the comparator used for receiving KBDCLK forms a schmitt trigger. One half of the 74HCT74 is used as a clock signal inverter, the other half indicates when clock pulses have occurred and that the keyboard therefore may be sending data to the DUART.

Diagram page 15 shows the V24 line receivers and transmitters and their protection filters.

Diagram page 16 shows the Piezoelectric sound transducer and its driver.

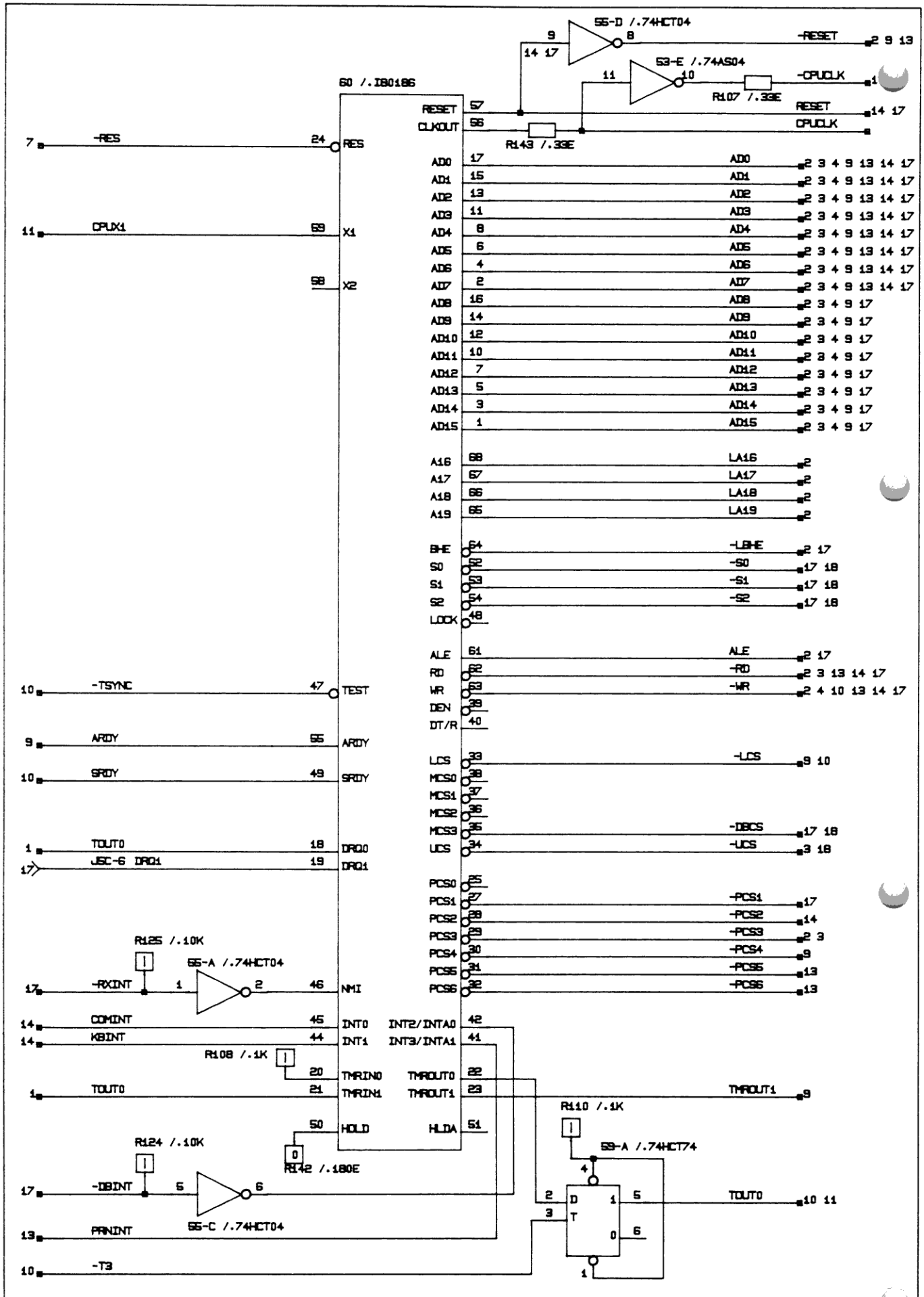
VOL=1 gives a 10VPP drive voltage and VOL=0 gives 5Vpp drive.

The system reset signal -RES is generated on power up and may also be manually generated from the Debug Board.

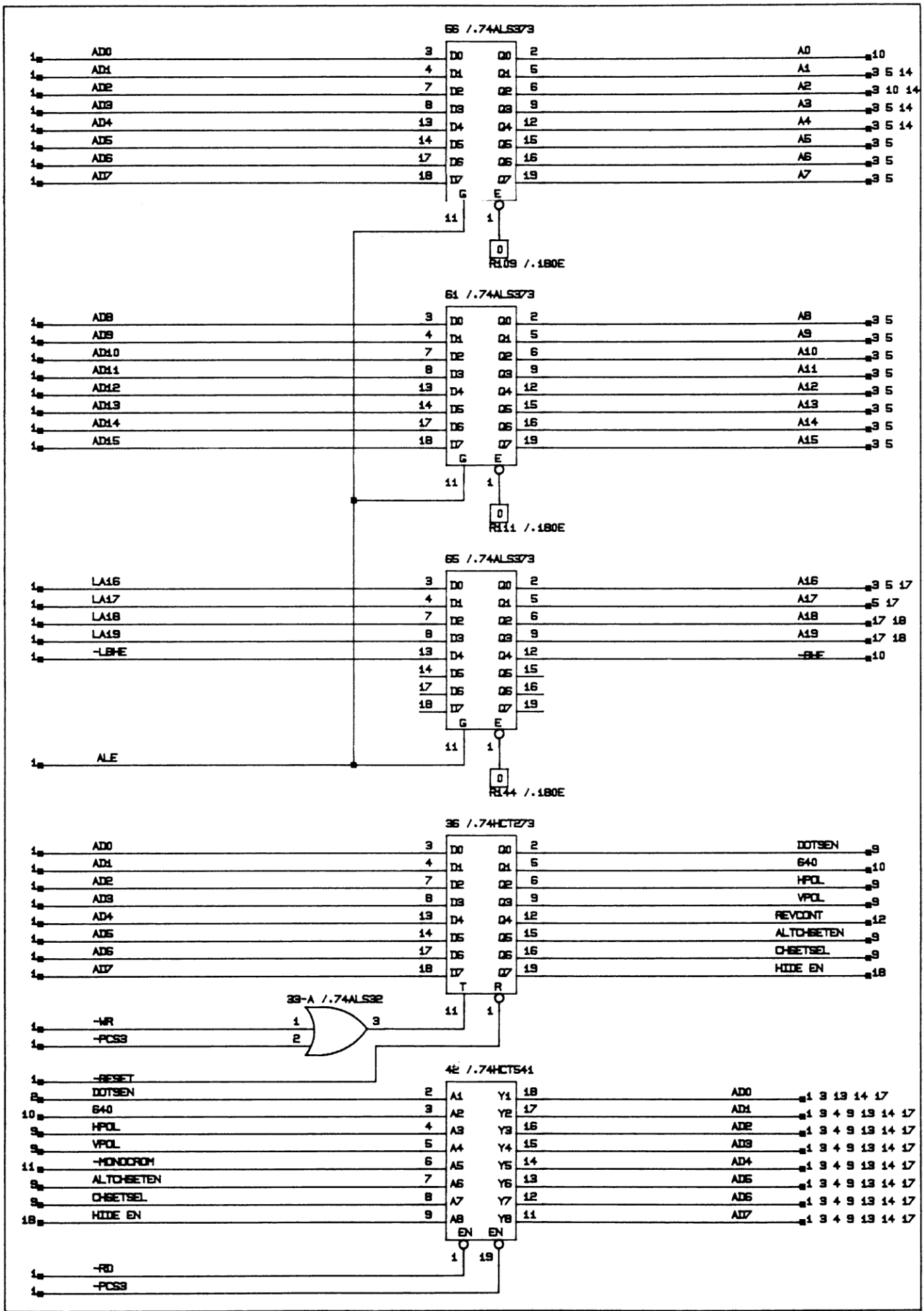
Also shown is the pinout of the parallel printer connector.

Diagram page 17 shows the Debug Board connector pinout.

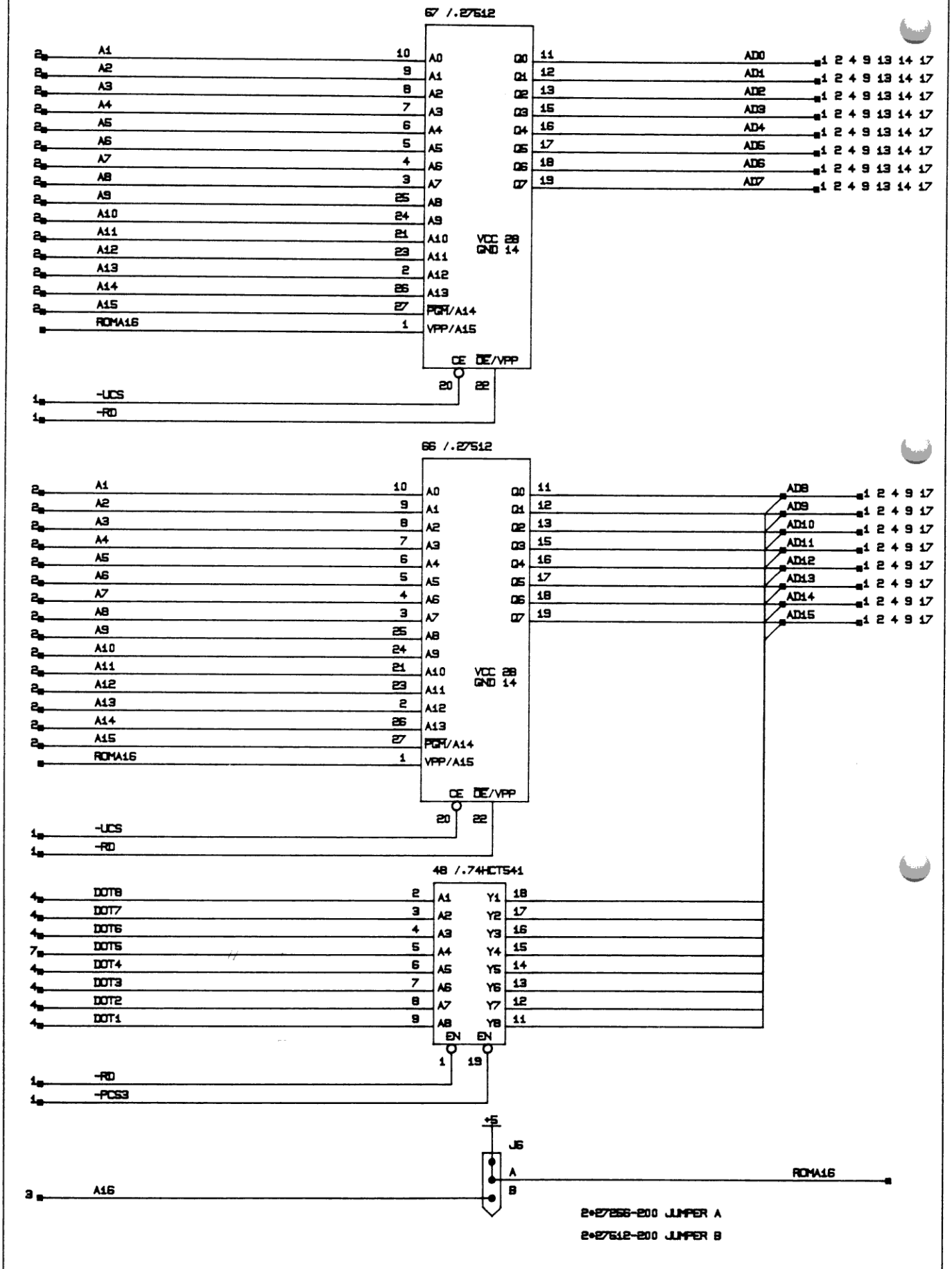
3.3 Logic Diagrams



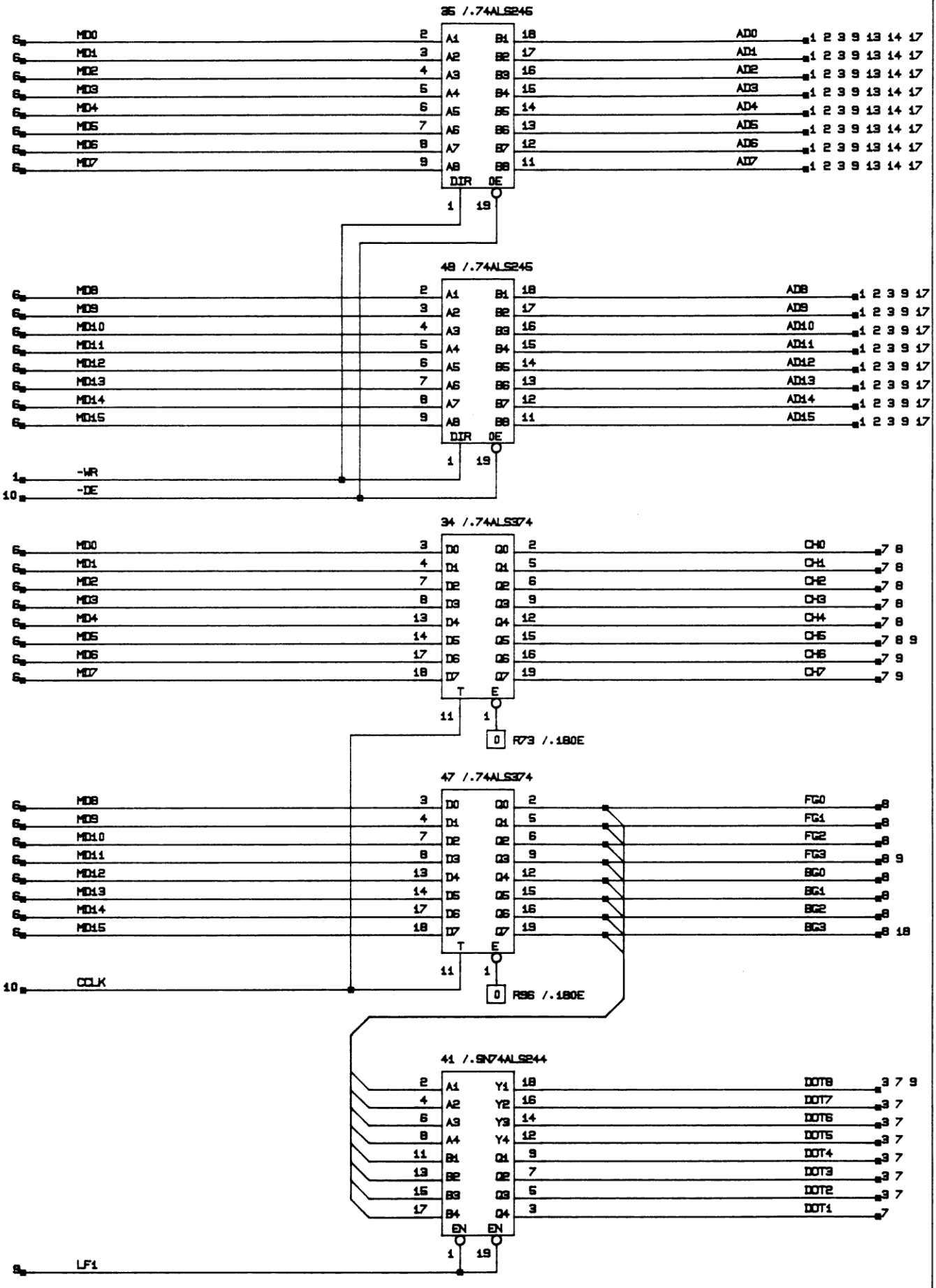
UNIT	DESIGN		
DCB451	PKA	CPU	1
DWG. NO.	DRAWN LIJ 880217		



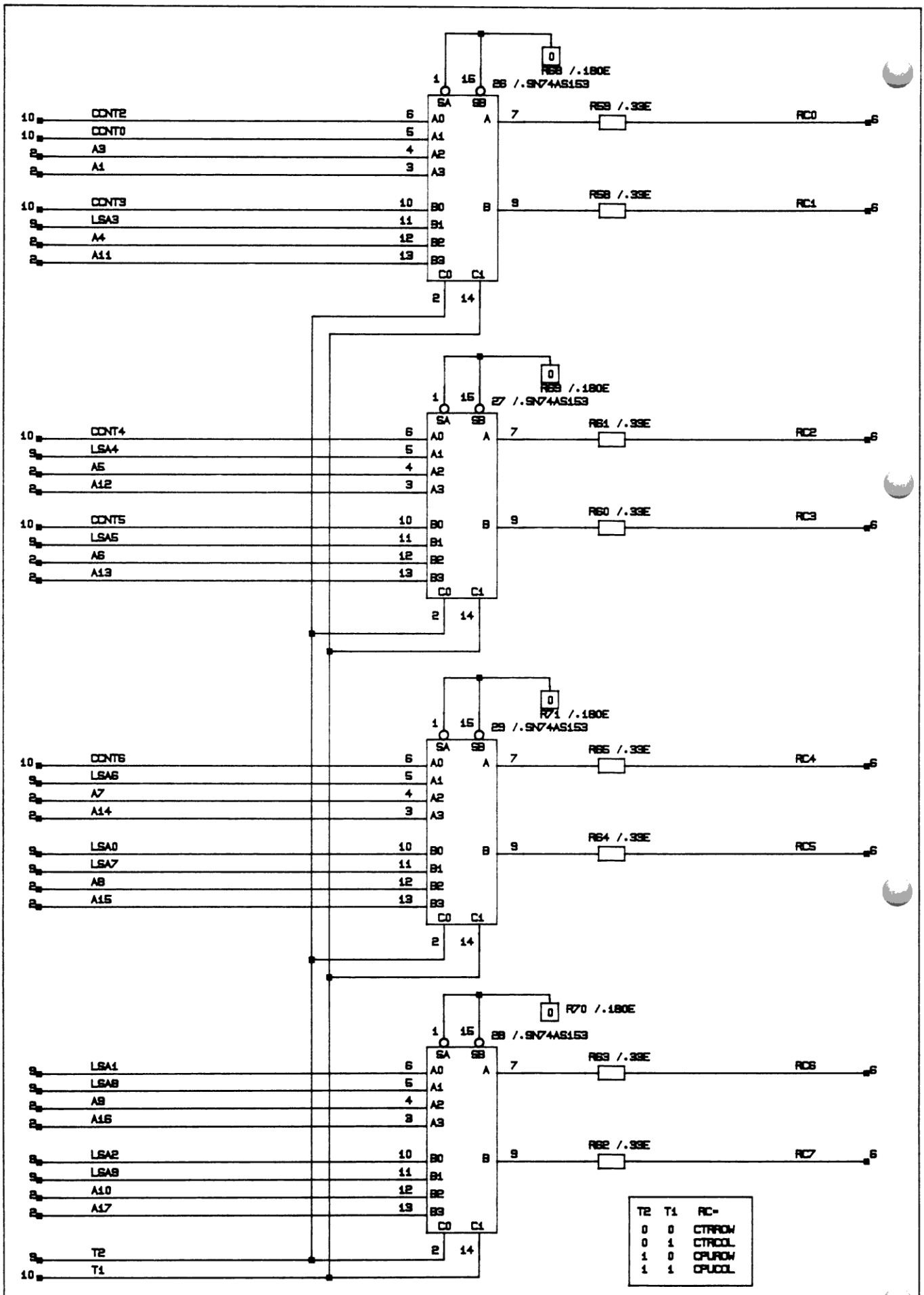
UNIT	DESIGN		
DCB451	PKA	ADDRESS LATCHES, MODE REGISTER	2
DWG. NO.	DRAWN LIJ 880212		



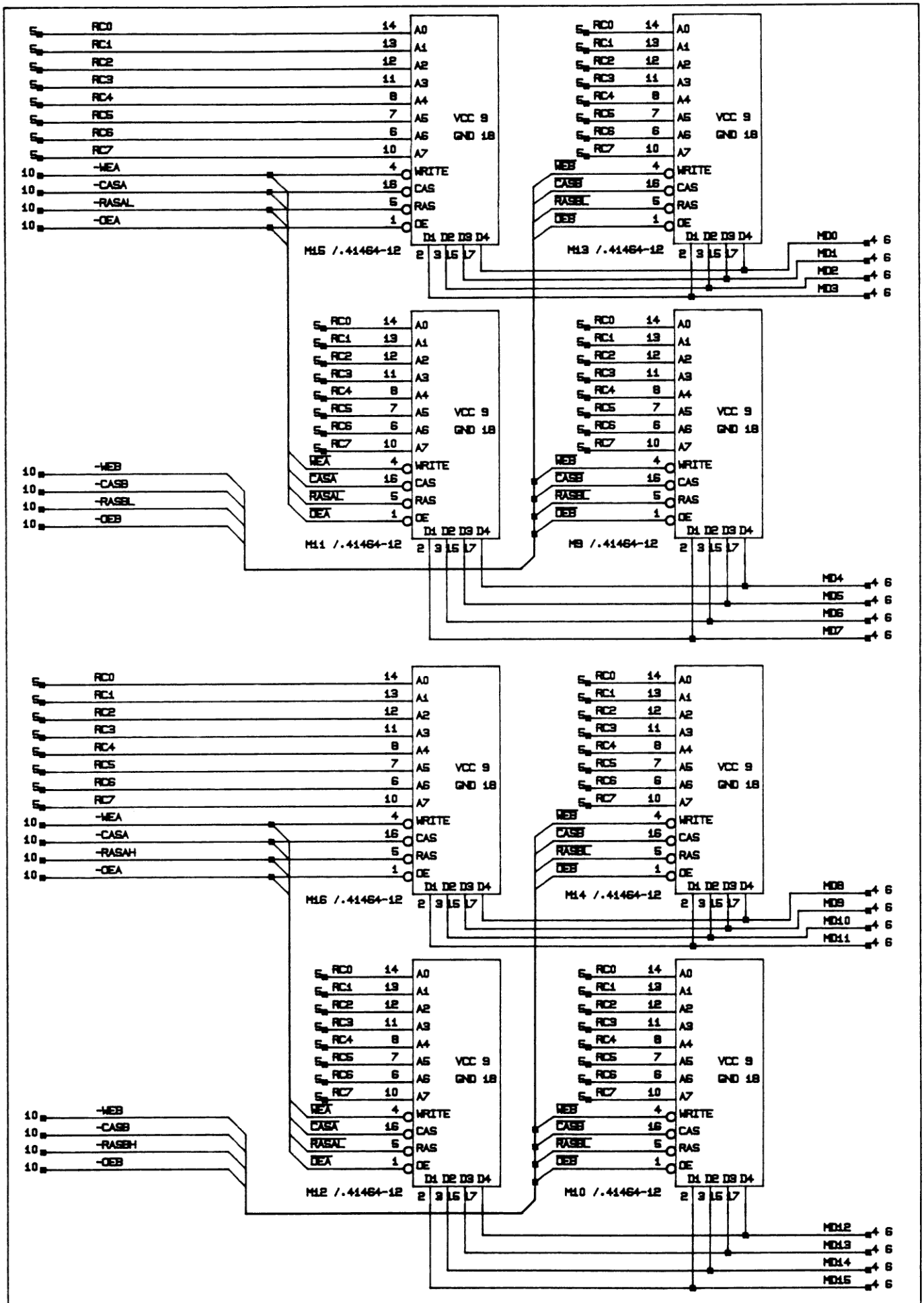
UNIT	DESIGN		
DCB451	PKA	ROM	3
DWG. NO.	DRAWN LIJ 880217		



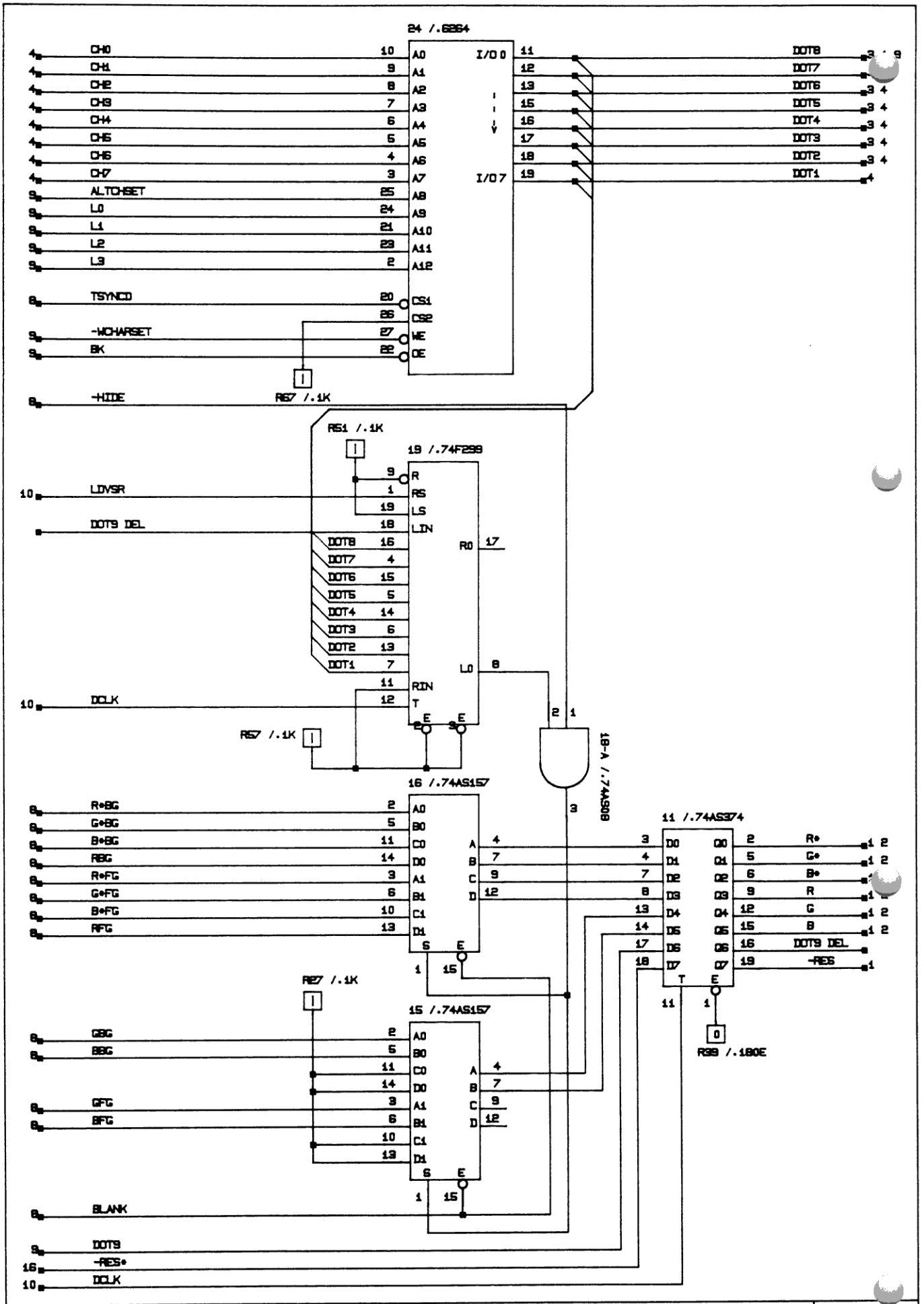
UNIT	DESIGN		
DCB451	PKA	MEMORY BUS TRANSFER, CHARACTER PIPELINE	4
DWG. NO.	DRAWN LIJ 880217		



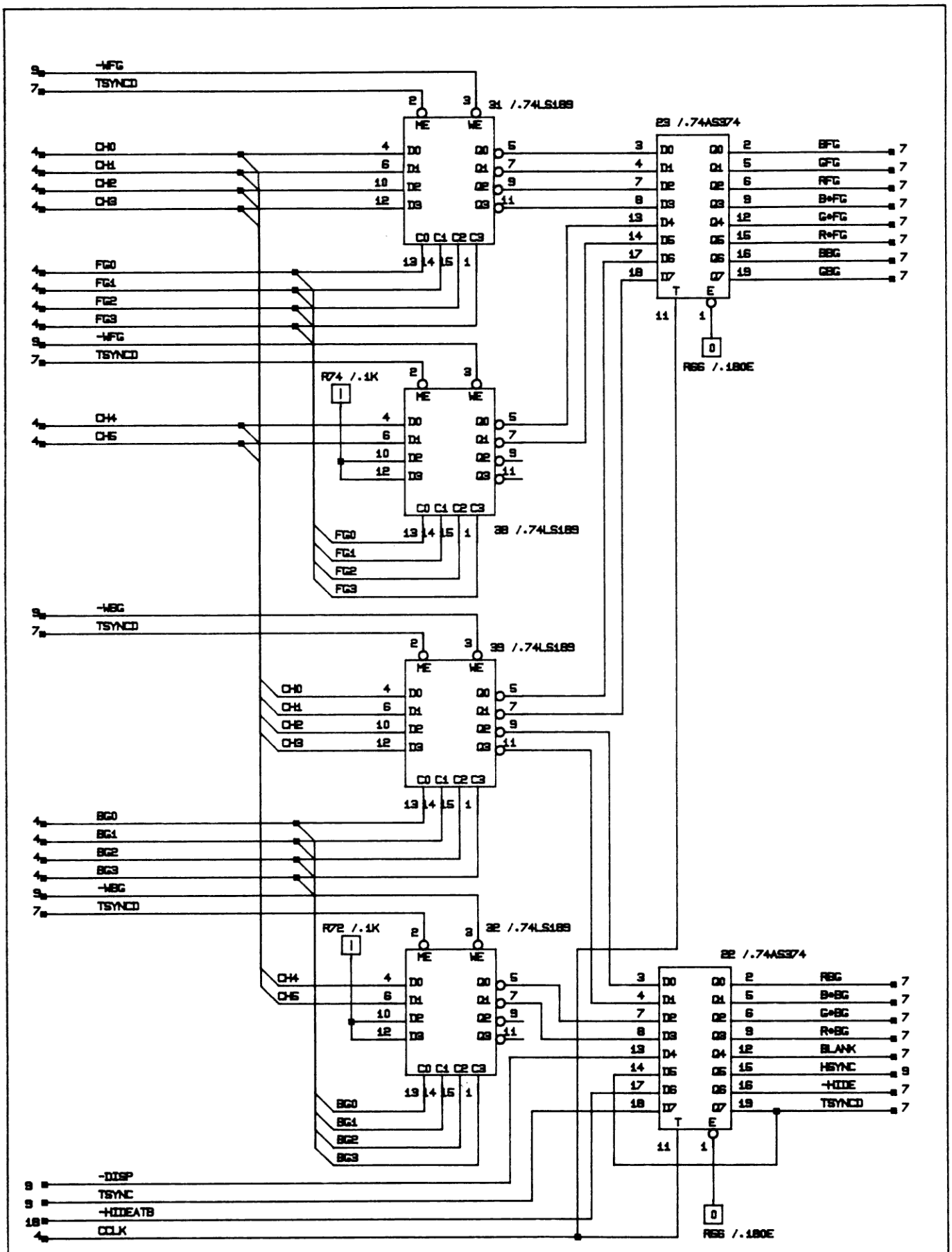
UNIT	DESIGN		
DCB451	PKA	ADDRESS MUX	5
DWG. NO.	DRAWN LTJ 880212		



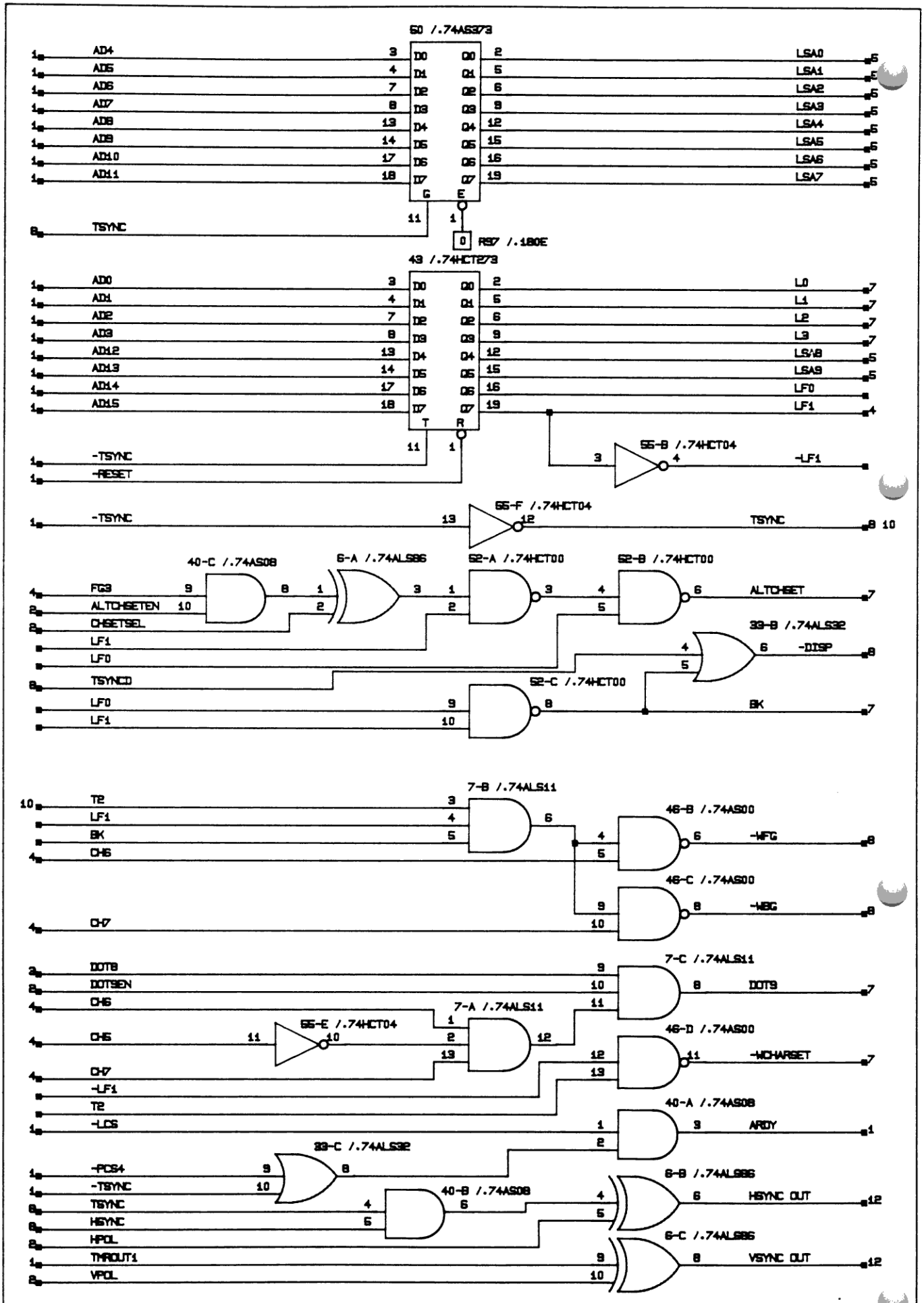
UNIT	DESIGN		
DCB451	PKA	256K MEMORY	6
DWG. NO.	DRAWN LIJ 880215		



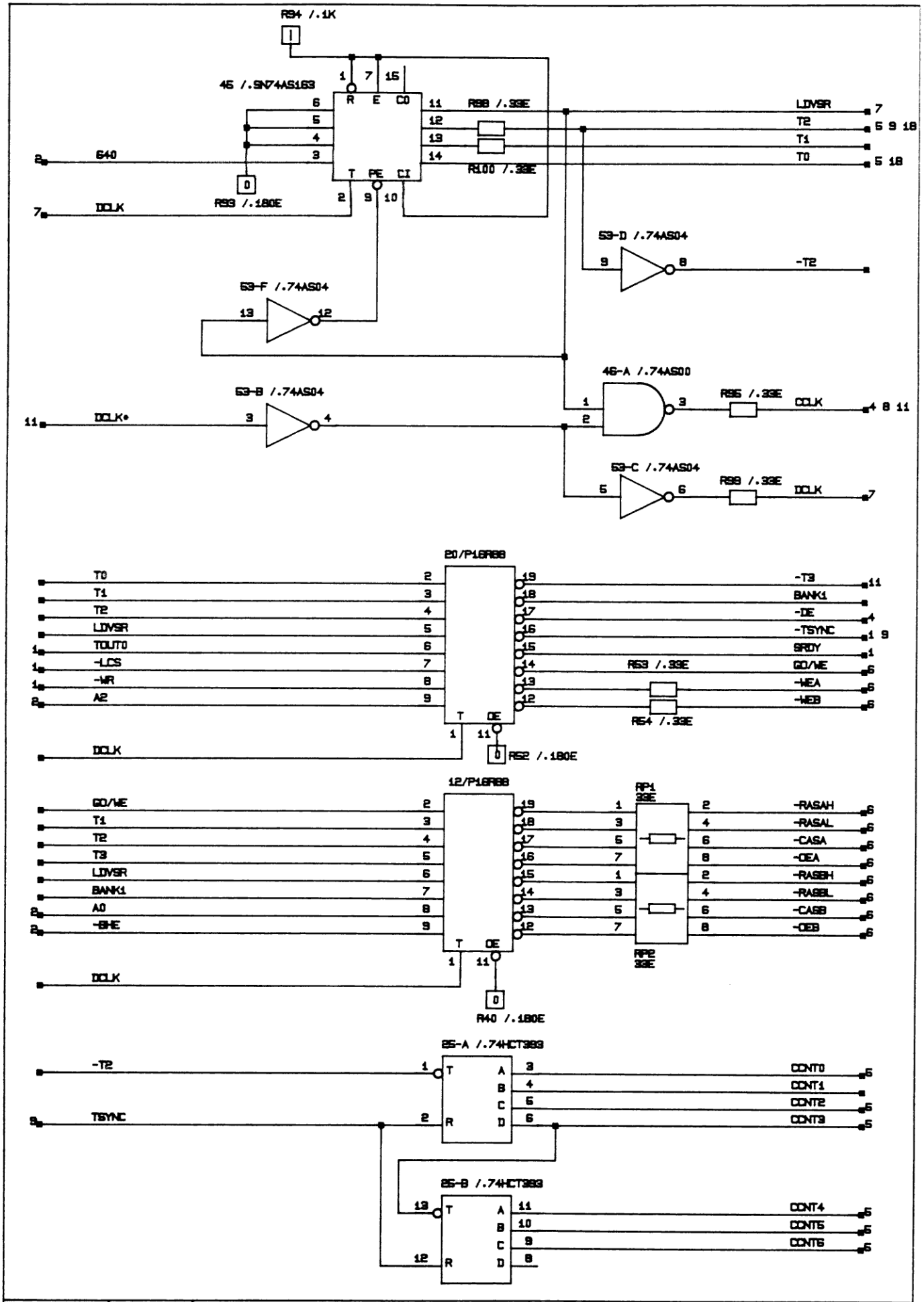
UNIT DCB451	DESIGN PKA	CHARACTER GENERATOR AND VIDEO SHIFT REGISTER	7
DWG. NO.	DRAWN LIJ 880217		



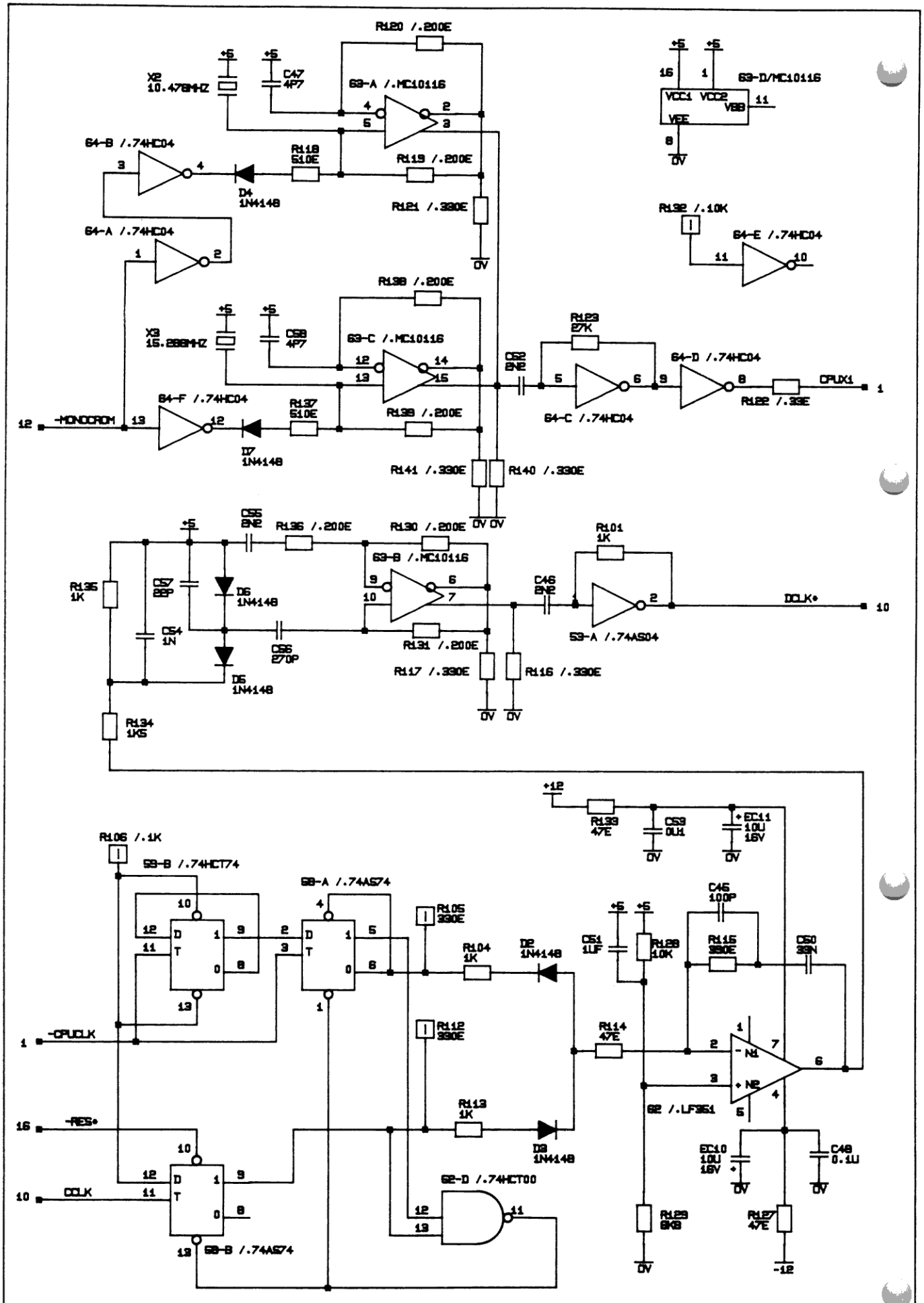
UNIT	DESIGN		
DCB451	PKA	PALLETTE	8
DWG. NO.	DRAWN LTJ 880215		



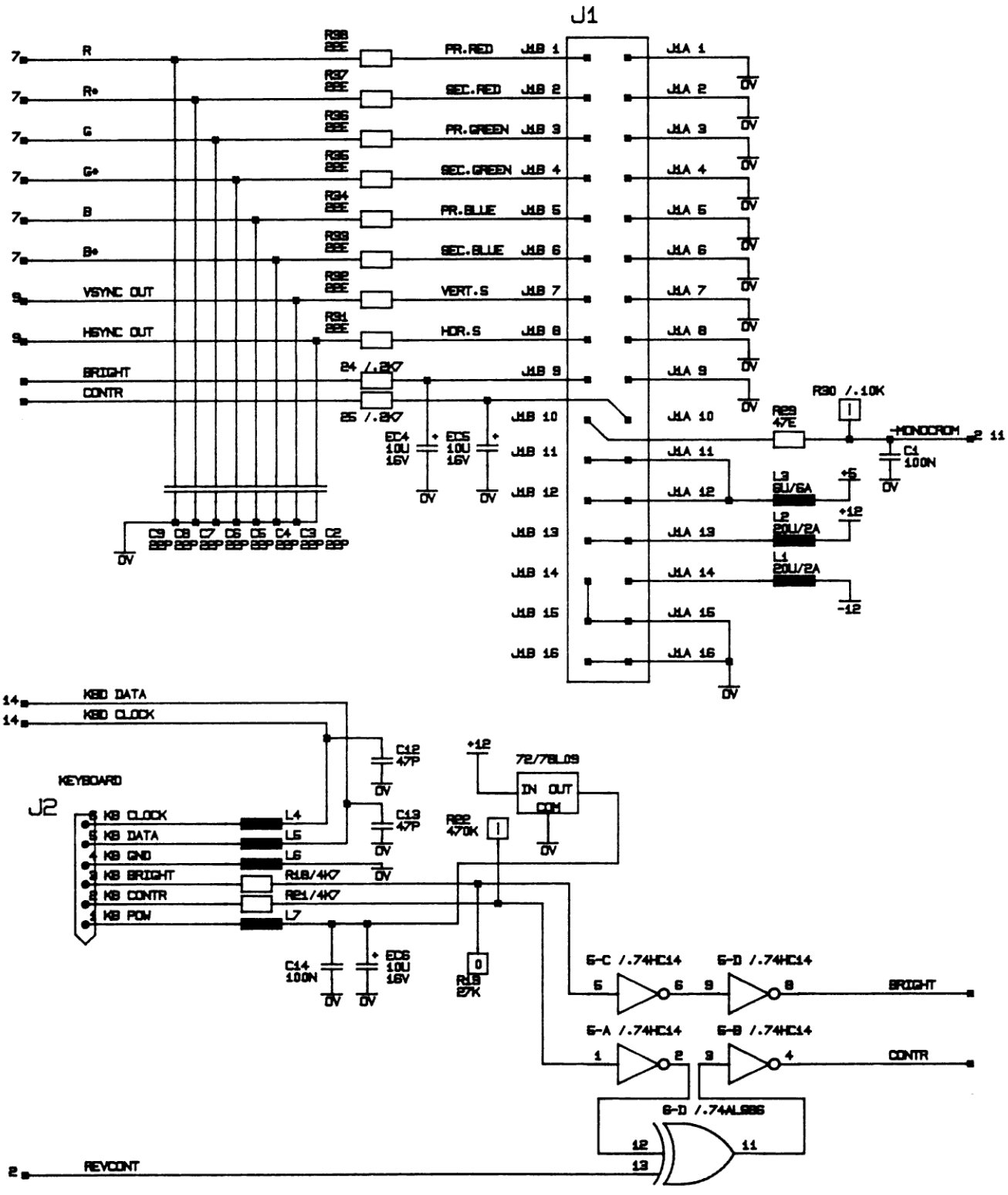
UNIT DCB451	DESIGN PKA	LINE CONTROL REGISTER, ATTRIBUT LOGIC AND WAITSTATE LOGIC	9
DWG. NO.	DRAWN LIJ 880217		



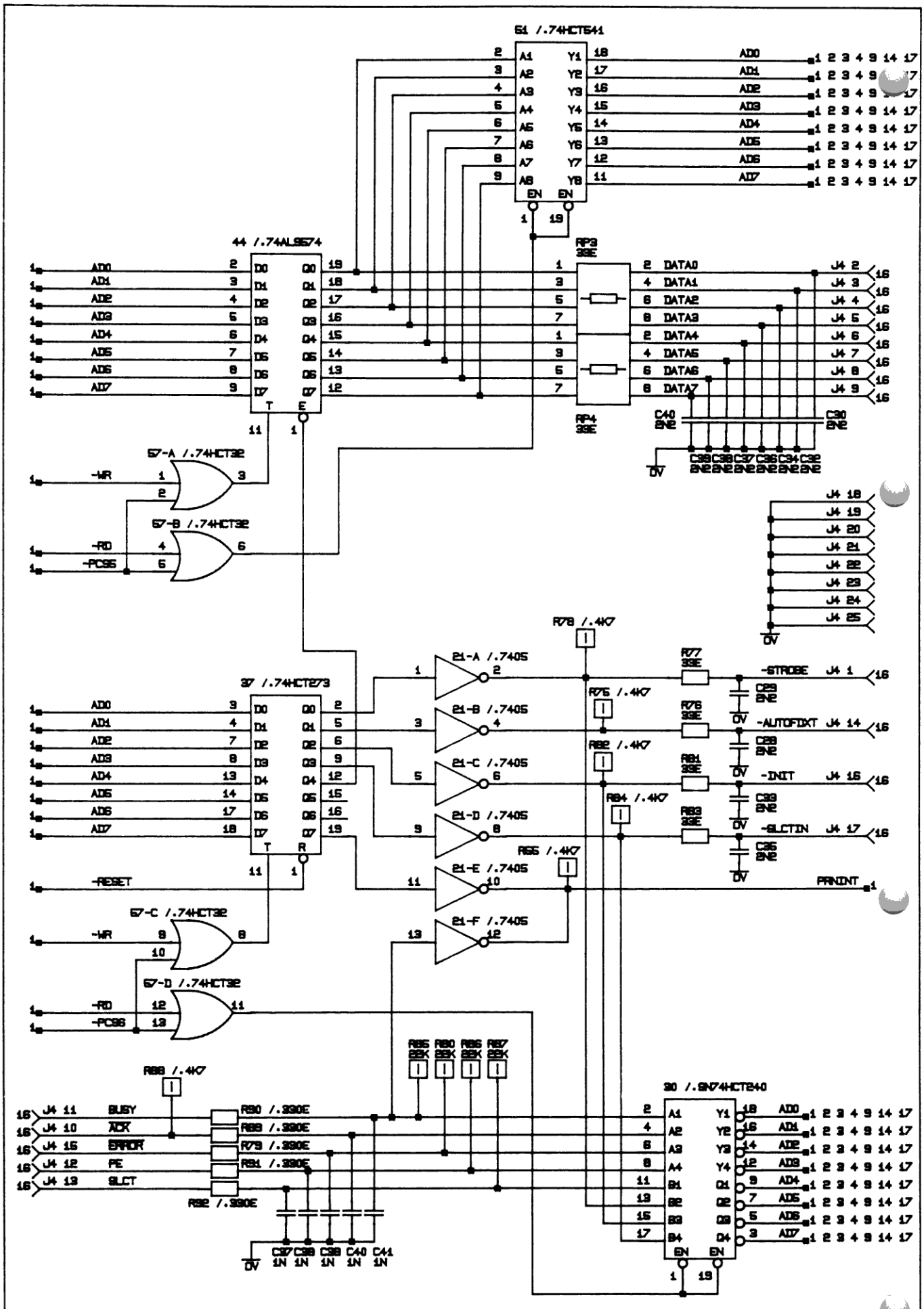
UNIT	DESIGN		
DCB451	PKA	TIMING AND CONTROL	10
DWG. NO.	DRAWN LIJ 880217		



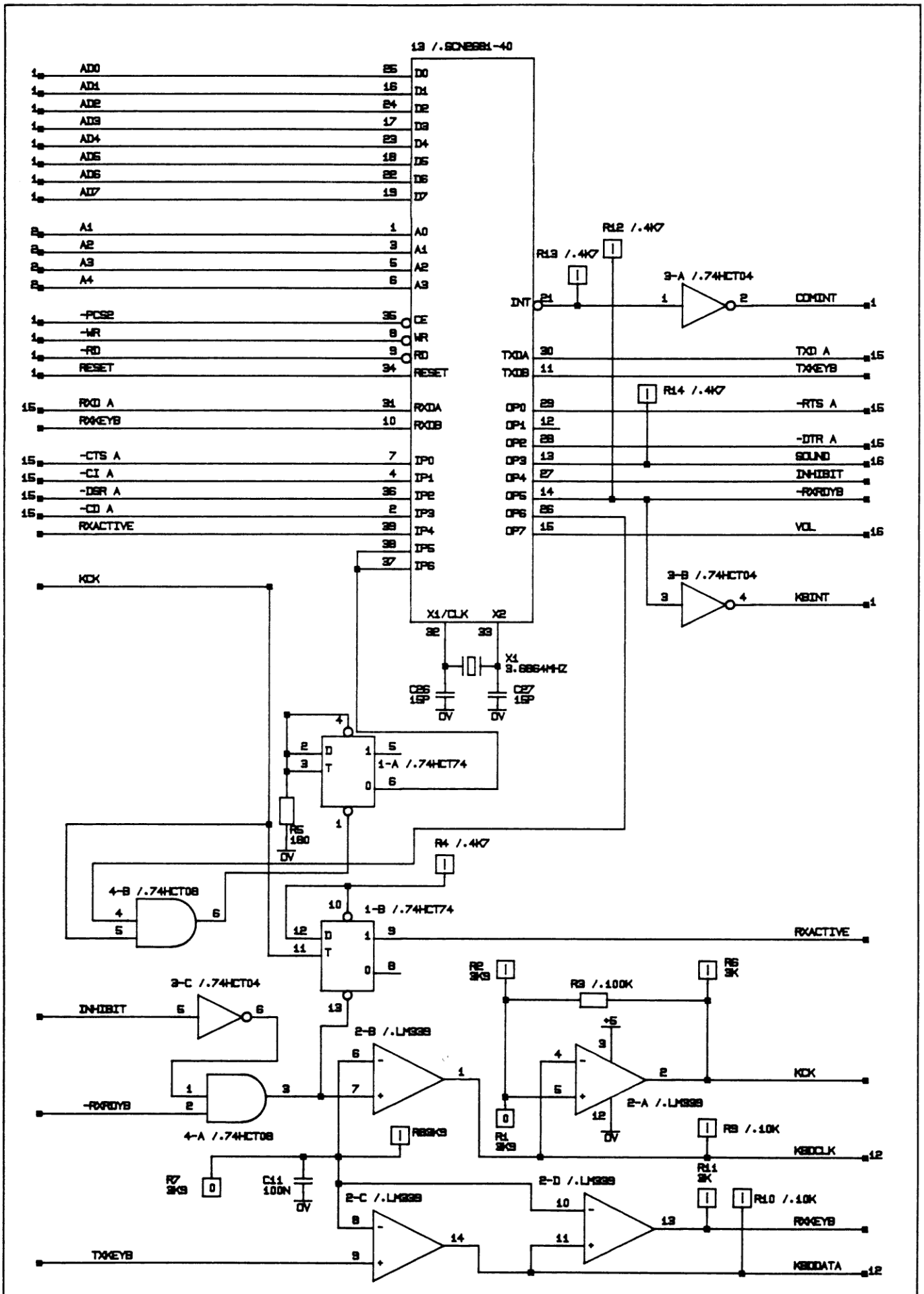
UNIT DCB451	DESIGN JPJ	X-TAL OSCILLATORS, VCO, PHASE COMPARATOR AND LOOP AMPLIFIER	11
DWG. NO.	DRAWN LTJ 880217		



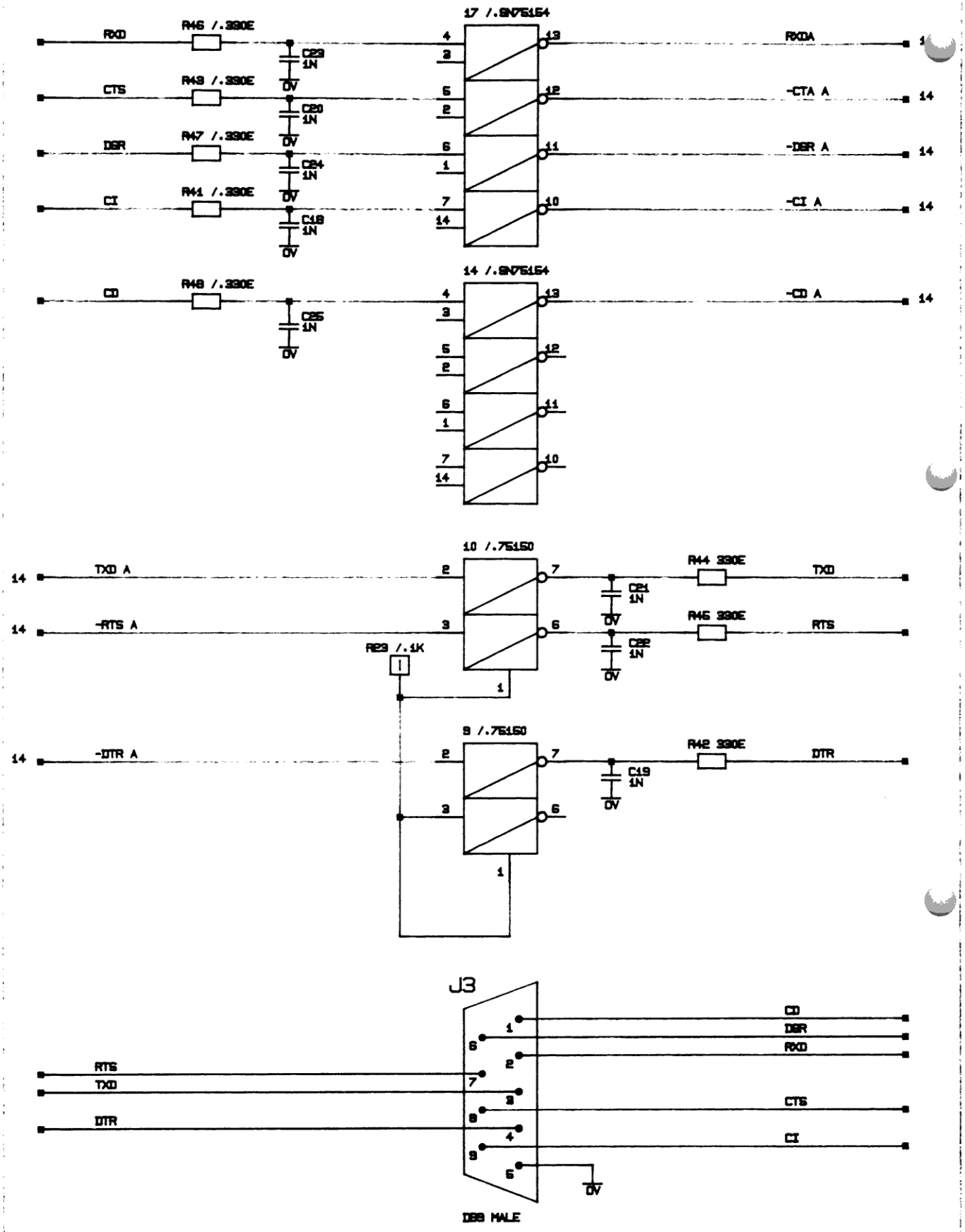
UNIT	DESIGN		
CPU451	JPJ	MONITOR AND KEYBOARD CONNECTORS	12
DWG. NO.	DRAWN LIJ 880215		



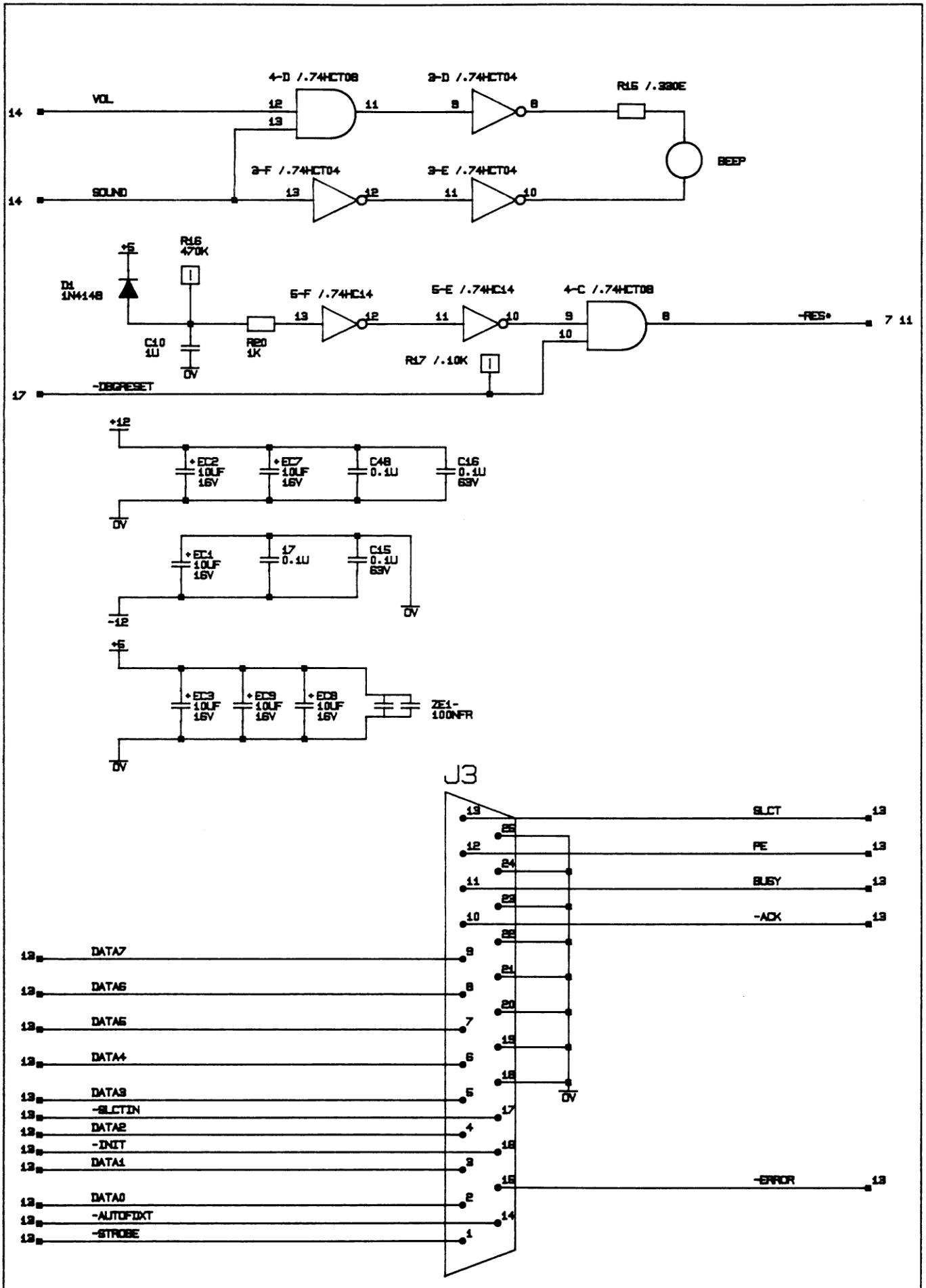
UNIT	DESIGN		
DCB451	JPJ	PARALLEL PRINTER INTERFACE	13
DWG. NO.	DRAWN LIJ 880217		



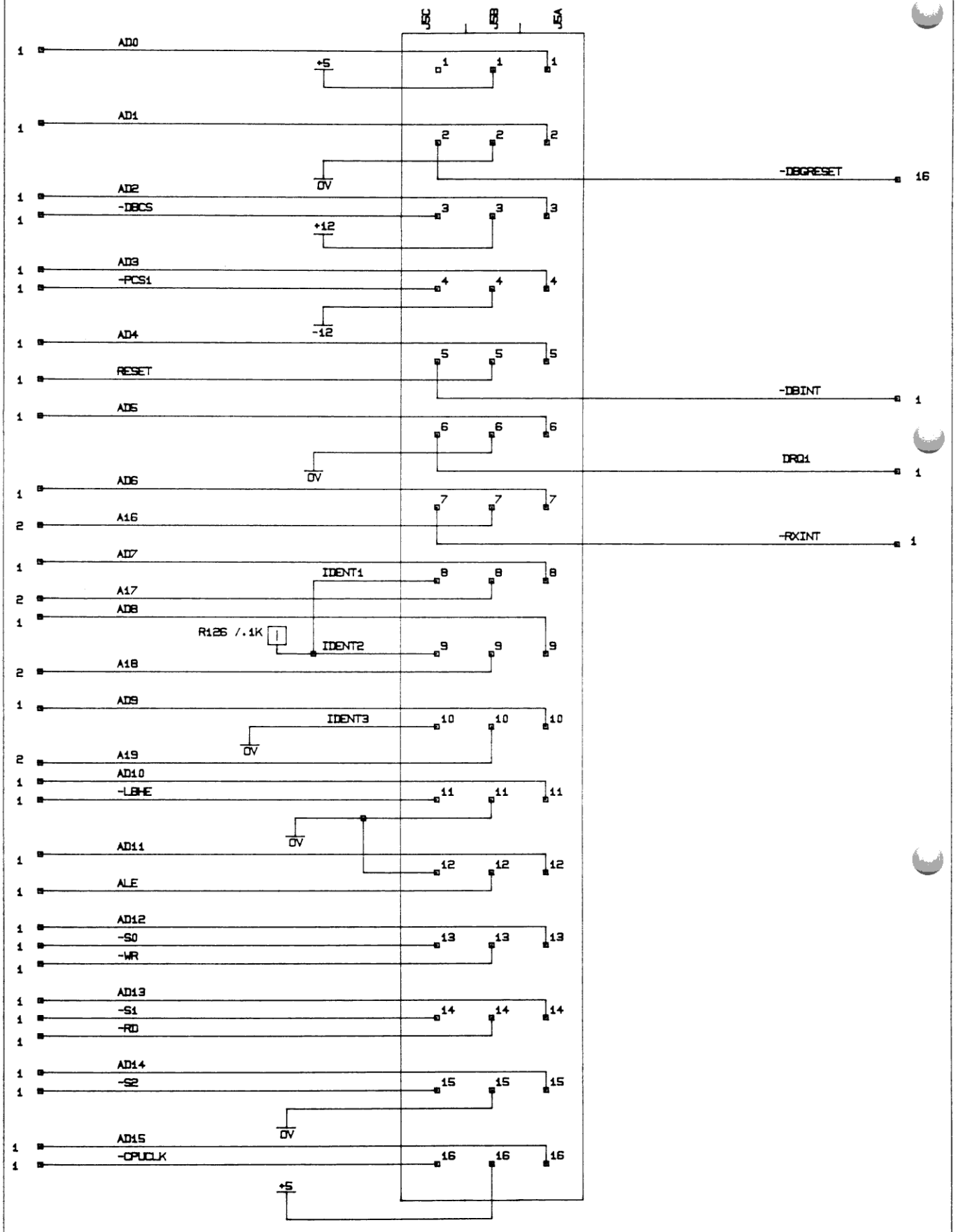
UNIT	DESIGN	DUART LINE AND KEYBOARD INTERFACE	14
DCB451	JPJ		
DWG. NO.	DRAWN LIJ 880217		



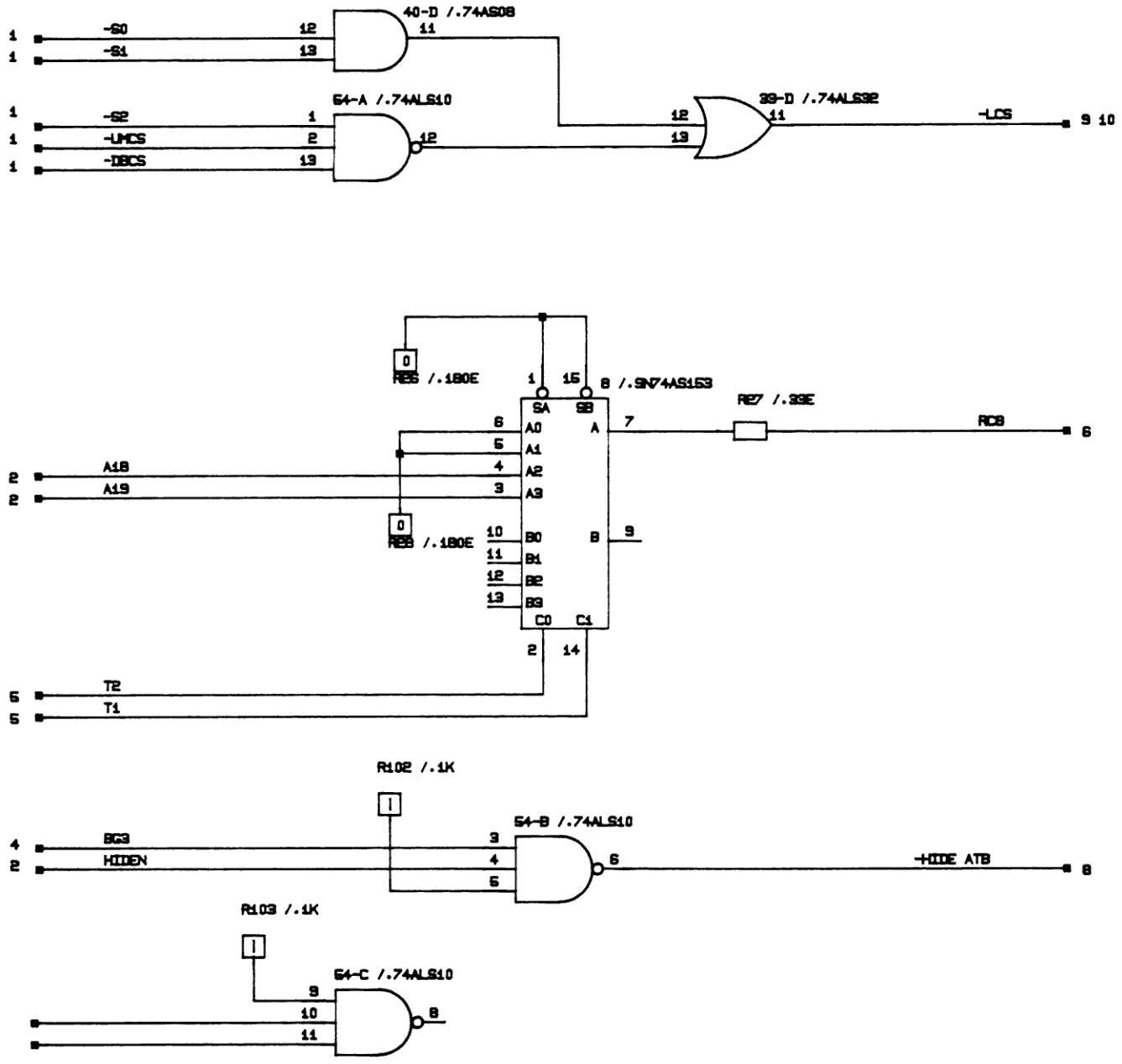
UNIT DCB451	DESIGN JPJ	LINE RECEIVERS AND TRANSMITTERS	15
DWG. NO.	DRAWN LTJ 880215		



UNIT DCB451	DESIGN JPJ	SOUND OUTPUT, POWER RESET, DECOUPLING. PARALLEL PRINTER CONNECTOR	16
DWG. NO.	DRAWN LTJ 880217		



UNIT	DESIGN		
DCB451	JPJ	DEBUGGER CONNECTOR	17
DWG. NO.	DRAWN LIJ 880215		



UNIT DCB451	DESIGN PKA	RANDOM LOGIC	18A
DWG. NO.	DRAWN LIJ 880217		

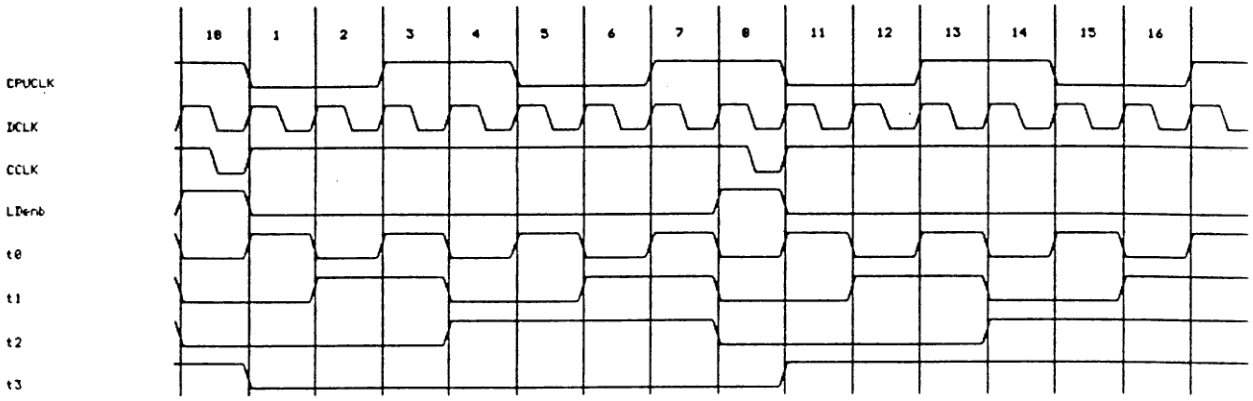
3.4 PAL-Descriptions

```
MODULE DCBPAL1
TITLE  'DCB451 pal 1 memory bus controller
                        Author:   Peter Koch Andersson
                        Edition:  12/10-1987
                        Revision: 2';

PAT134  DEVICE   'P16R8';
      t0,t1,t2  PIN 2,3,4;  " I Timestate bits 0:2.
      !t3       PIN 19;     " O Timestate bit (MSB).
      ldvsr     PIN 5;      " I LoaD VideoShift Register.
      tm        PIN 6;      " I TiMer0 from 80186
                            "    (horizontal retrace).
      !tms      PIN 16;     " O Timer0 syncronized to
                            "    ldvsr for second char.
      !lcs      PIN 7;      " I Lower mem.Chip Select.
      !wr,a2    PIN 8,9;    " I WRite, Address bit 2.
      !bank1    PIN 18;     " O Memorybank selector indicates
                            "    which bank is used by the CRT.
      !de       PIN 17;     " O Data output enable
      !go       PIN 14;     " O Different def's at diff. times:
                            "    t=0-2: proceed with CRT-access
                            "    t=4-6: proceed with CPU-access
                            "    t=8-10:write operation to mem.
                            "    t=12-14:RAS hold for CRT-access.
      srdy      PIN 15;     " O SRDY to 80186.
      !wea,!web PIN 13,12;  " O Writeenable to memorybanks.

      H,L,X,C,Z =1,0,.x.,.c.,.z.;

      input     = Æ a2,wr,lcs,tm,ldvsr,t2,t1,t0 Å;
      output    = Æ t3,bank1,de,tms,srdy,go,wea,web Å;
      t         = Æ t3,t2,t1,t0 Å;

EQUATIONS

        t3      := t3 & !ldvsr
                 § !t3 &  ldvsr;
        t3      := ldvsr & t2;      " --Test reset
        bank1   :=  tms & (t==3) & !a2
                 § !bank1 & (t==-1) & !tms & !tm
                 §  bank1 & (t!=-1);
        tms     := tm  & ldvsr & t3
                 § tms & (!ldvsr § !t3);
        go      := (t==-1) & !tm
                 § (t== 3) & lcs & !tms & a2 & !bank1
                 § (t== 3) & lcs & !tms & !a2 & bank1
                 § (t== 3) & lcs &  tms
                 § (t==11) & !tms
                 § (go & !t1) § (srdy & wr & ldvsr);
        !srdy   := !srdy & (t!=5)
                 § !srdy & !go
                 § t3;
        de      := srdy & t3
                 § de & (t!=13) & t3;
        wea     := go & !bank1 & (t==10)
                 § wea & ((t==11)§(t==12));
        web     := go &  bank1 & (t==10)
                 § web & ((t==11)§(t==12));

END DCBPAL1
```

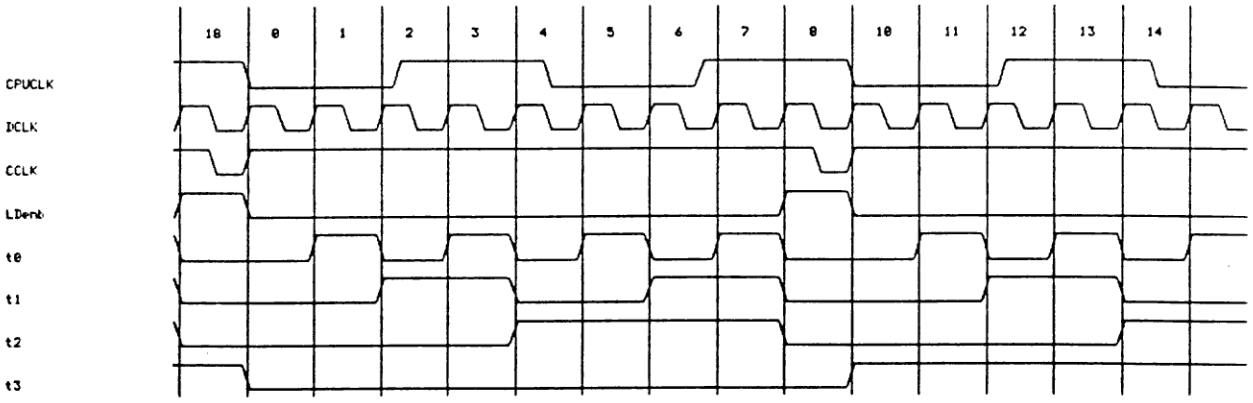
```
MODULE DCBPAL2"                                  page 1


TITLE  'DCB451 pal 2 DRAM controller
                        Author:   Peter Koch Andersson
                        Edition:  13/10-1987
                        Revision: 2 ';


PAT135  DEVICE   'P16R8';


    t1,t2,!t3      PIN 3,4,5;   " I Timestate bits 1:3.
    ldvsr          PIN 6;       " I Load VideoShift Register.
    !go            PIN 2;       " I Different def's at diff. times:
                                "    t=0-2: proceed with CRT-access
                                "    t=4-6: proceed with CPU-access
                                "    t=8-10:write operation to mem.
                                "    t=12-14:RAS hold for CRT-access
    a0,!bhe        PIN 8,9;     " I Address bit 0 ,byte-high-enb.
    !bank1         PIN 7;       " I Indicates which bank is used
                                "    by the CRT.
    !rasah,!rasal PIN 19,18;    " O RAS to bank a,high-,low-byte
    !rasbh,!rasbl PIN 15,14;    " O RAS to bank b,high-,low-byte
    !casa,!casb    PIN 17,13;   " O CAS to bank a,b
    !oea,!oeb      PIN 16,12;   " O Output Enable to bank a,b


        H,L,X,C,Z                  =1,0,.x.,.c.,.z.;
        input = Æa0,bhe,bank1,ldvsr,t3,t2,t1,goÅ;
        output= Ærasah,rasal,casa,oea,rasbh,rasbl,casb,oebÅ;
```

```
EQUATIONS

        rasah   := ldvsr & t3 & go & bank1       " CRT access
                 § t2 & !t3 & go & !bank1 & bhe " CPU access
                 § !t3 & ldvsr & !bank1         " Refresh
                 § rasah & !t3                  "  hold
                 § rasah & !t2                  "
                 § rasah & !t1 & bank1 & go;    "

        rasa1   := ldvsr & t3 & go & bank1       " CRT access
                 § t2 & !t3 & go & !bank1 & !a0 " CPU access
                 § !t3 & ldvsr & !bank1         " Refresh
                 § rasa1 & !t3                  "  hold
                 § rasa1 & !t2                  "
                 § rasa1 & !t1 & bank1 & go;    "

        casa    :=  bank1 & rasah & t1          " CRT access
                 §  bank1 & rasa1 & t1          "
                 §  bank1 & casa & t2           "
                 § !bank1 & t1 & t2 & !t3       " CPU access
                 § !bank1 & casa & !t2 & !oea   "  or refresh
                 § !bank1 & casa & rasah        "
                 § !bank1 & casa & rasa1;       "

        oea     := !bank1 & !go & !t1 & !t2 & t3        "
                 § !bank1 & oea & rasah         "
                 § !bank1 & oea & rasa1         "
                 §  bank1 & t2 & t1 ;           "

        rasbh   := ldvsr & t3 & go & !bank1      " CRT access
                 § t2 & !t3 & go & bank1 & bhe  " CPU access
                 § !t3 & ldvsr & bank1          " Refresh
                 § rasbh & !t3                  "  hold
                 § rasbh & !t2                  "
                 § rasbh & !t1 & !bank1 & go;   "

        rasb1   := ldvsr & t3 & go & !bank1      " CRT access
                 § t2 & !t3 & go & bank1 & !a0  " CPU access
                 § !t3 & ldvsr & bank1          " Refresh
                 § rasb1 & !t3                  "  hold
                 § rasb1 & !t2                  "
                 § rasb1 & !t1 & !bank1 & go;   "

        casb    := !bank1 & rasbh & t1          " CRT access
                 § !bank1 & rasb1 & t1          "
                 § !bank1 & casb & t2           "
                 § bank1 & t1 & t2 & !t3        " CPU access
                 § bank1 & casb & !t2 & !oeb    "  or refresh
                 § bank1 & casb & rasbh         "
                 § bank1 & casb & rasb1;        "

        oeb     := bank1 & !go & !t1 & !t2 & t3 "
                 § bank1 & oeb & rasbh          "
                 § bank1 & oeb & rasb1          "
                 § !bank1 & t2 & t1;            "

END DCBPAL2
```

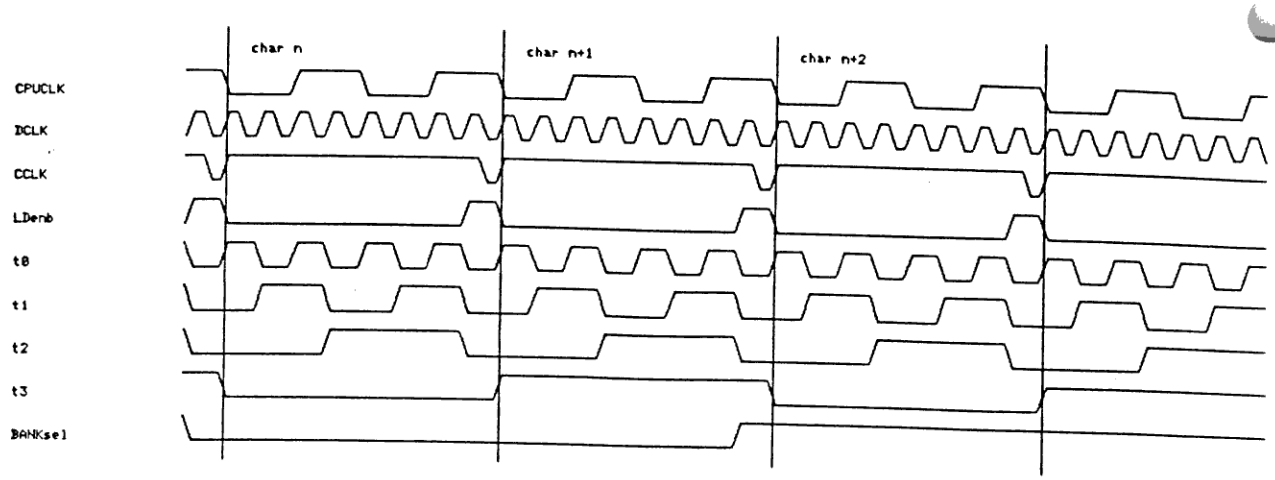
3.5 Timing Diagrams



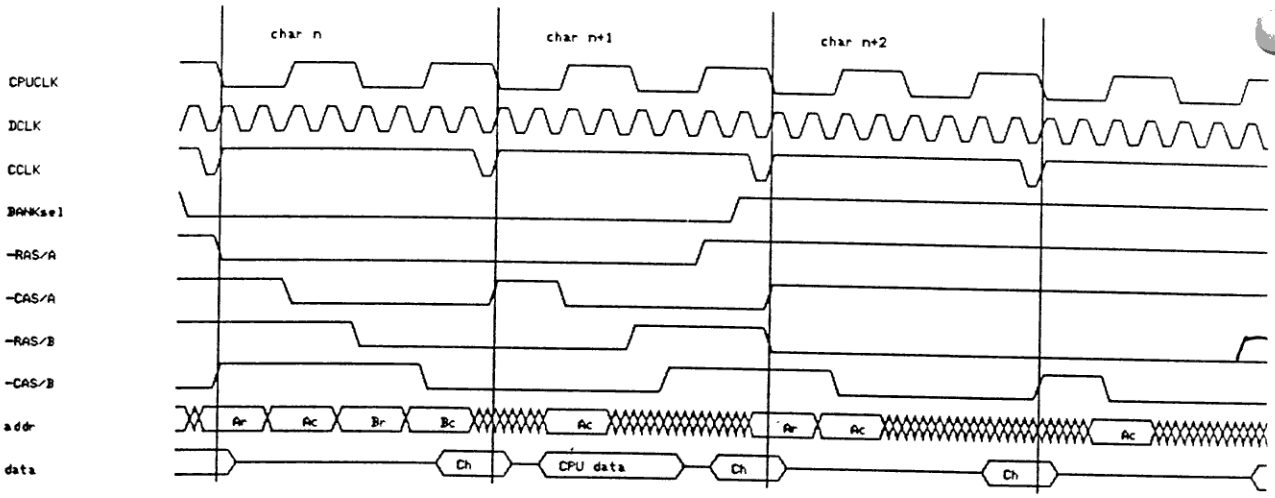
8 dots / character



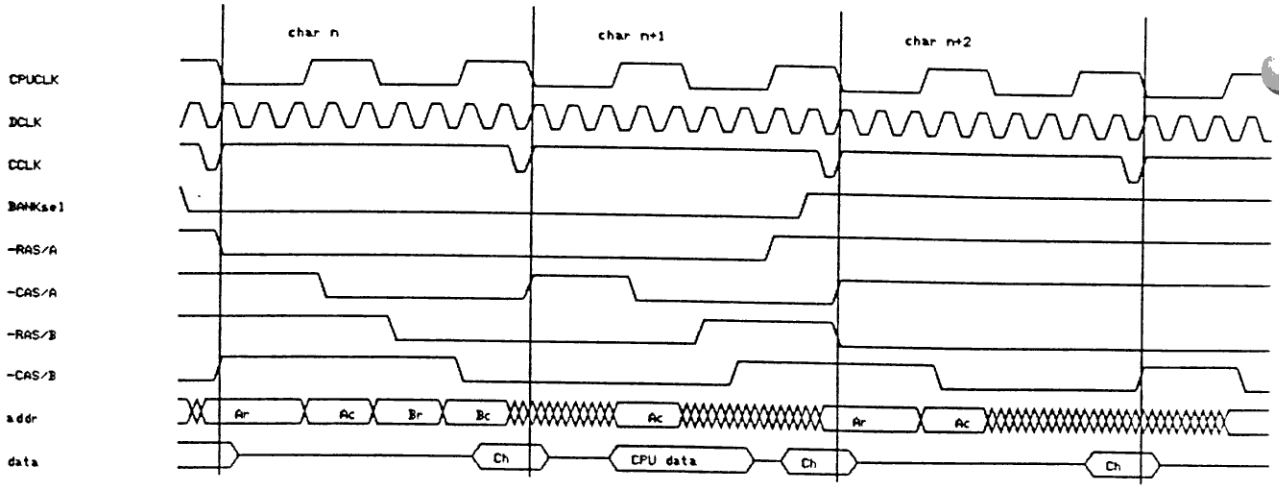
9 dots / character



Memory timing



8 dots / character

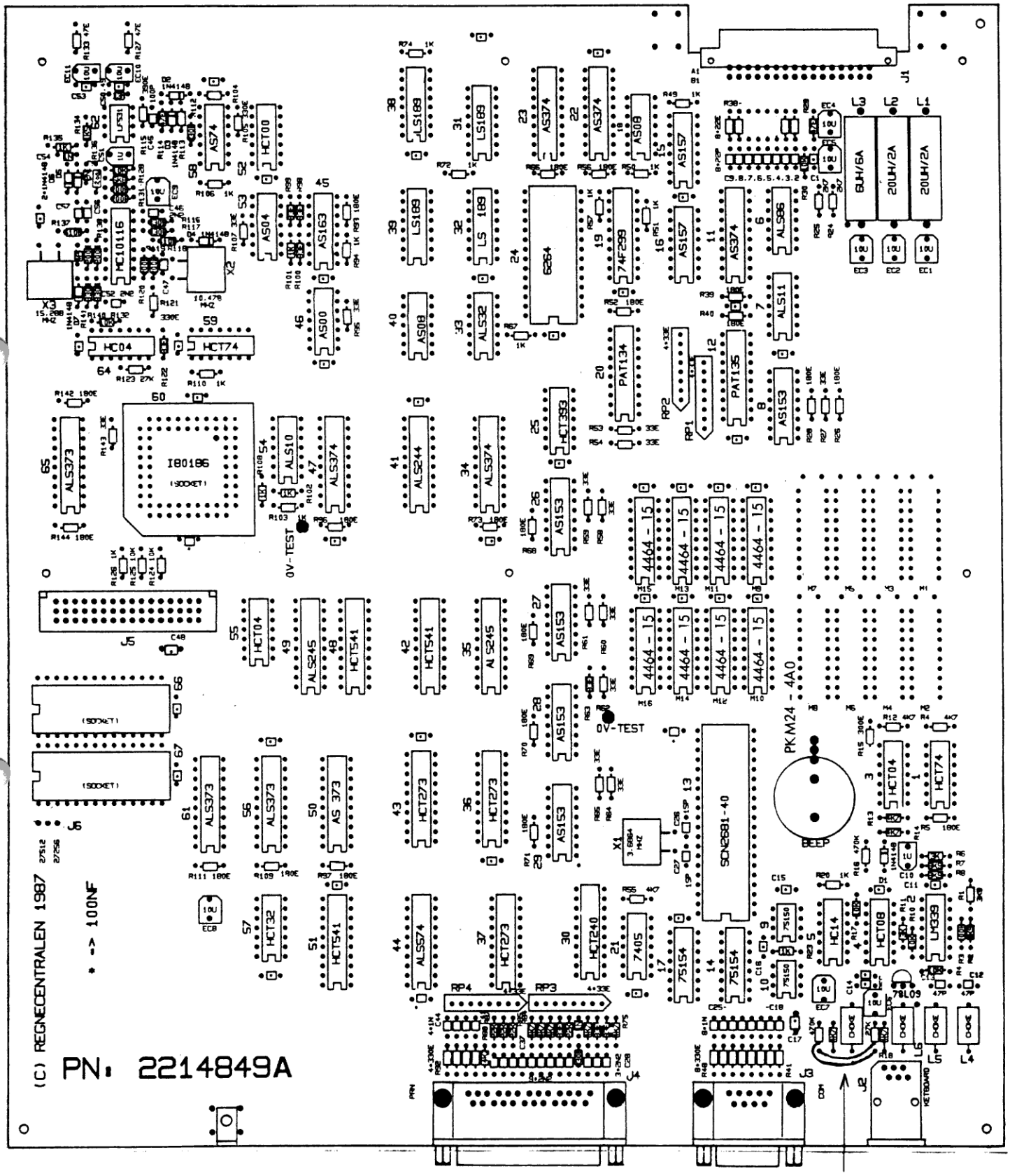


9 dots / character

Ar = CRT row address, Ac = CRT colouмn address

Br = CPU row address, Bc = CPU colouмn address

3.6 Assembly Drawing



3.7 Connectors

3.7.1 J1 Monitor Connector

Pin No	Signal Name
1 a	Signal return (RTN)
1 b	Primary RED
2 a	RTN
2 b	Secondary RED
3 a	RTN
3 b	Primary GREEN
4 a	RTN
4 b	Secondary GREEN
5 a	RTN
5 b	Primary BLUE
6 a	RTN
6 b	Secondary BLUE
7 a	RTN
7 b	Vertical sync.
8 a	RTN
8 b	Horizontal sync.
9 a	RTN
9 b	BRIGHTNESS
10 a	CONTRAST
10 b	-MONOCROM
11 a	+5V
11 b	+5V
12 a	+5V
12 b	+5V
13 a	+12V
13 b	+12V
14 a	-12V
14 b	0V
15 a	0V
15 b	0V
16 a	0V
16 b	0V

N.C. not connected.

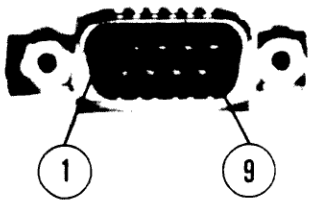
3.7.2 J2 Keyboard Connector

Pin 1 is at top to the left when the connector is seen from the rear.

Pin	Signal name
1	KB POW (+9V)
2	KB CONTR
3	KB BRIGHT
4	KB GND (0V)
5	KB DATA
6	KB CLOCK

3.7.3 J3 V24 9 pin connector

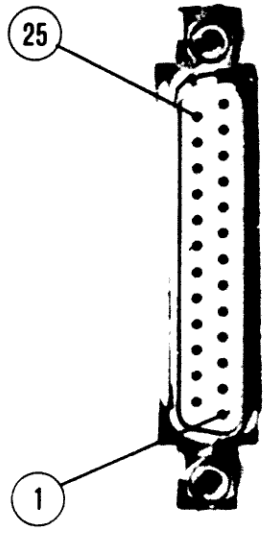
The V.24 connector seen from the rear:



Pin	Signal name
1	CD (Carrier detect)
2	RXD (Receive Data)
3	TXD (Transmit Data)
4	DTR (Data Terminal Ready)
5	Signal Ground
6	DSR (Data Set Ready)
7	RTS (Request To Send)
8	CTS (Clear To Send)
9	CI (Calling Indicator)

3.7.4 J4 Parallel Printer Connector

Pin	Name	I/O
1	-Strobe	O
2	Data bit 0	(I)/O
3	Data bit 1	(I)/O
4	Data bit 2	(I)/O
5	Data bit 3	(I)/O
6	Data bit 4	(I)/O
7	Data bit 5	(I)/O
8	Data bit 6	(I)/O
9	Data bit 7	(I)/O
10	-Ack	I
11	Busy	I
12	PE	I
13	Slct	I
14	-Auto feed xt	O
15	-Error	I
16	-Init	O
17	-Slct in	O
18..25	Ground	



3.7.5 J5 Debug Board Connector

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
A1	AD0	B1	+5V	C1	N.C.
A2	AD1	B2	0V	C2	-DEBUG RESET
A3	AD2	B3	+12V	C3	-DBCS
A4	AD3	B4	-12V	C4	-PCS1
A5	AD4	B5	CPURESET	C5	-DBINT
A6	AD5	B6	0V	C6	DRQ1
A7	AD6	B7	A16	C7	-RXINT
A8	AD7	B8	A17	C8	IDENT1 (1)
A9	AD8	B9	A18	C9	IDENT2 (1)
A10	AD9	B10	A19	C10	IDENT3 (0)
A11	AD10	B11	0V	C11	-BHEN
A12	AD11	B12	ALE	C12	0V
A13	AD12	B13	-WR	C13	-S0
A14	AD13	B14	-RD	C14	-S1
A15	AD14	B15	0V	C15	-S2
A16	AD15	B16	+5V	C16	-CPUCLK

4. REFERENCES

(1) Embedded Controller Handbook, (80186),
Intel, 1987.

(2) SCN2681 Dual Asynchronous Receiver/Transmitter (DUART)
Product Specification
Philips, 1985.