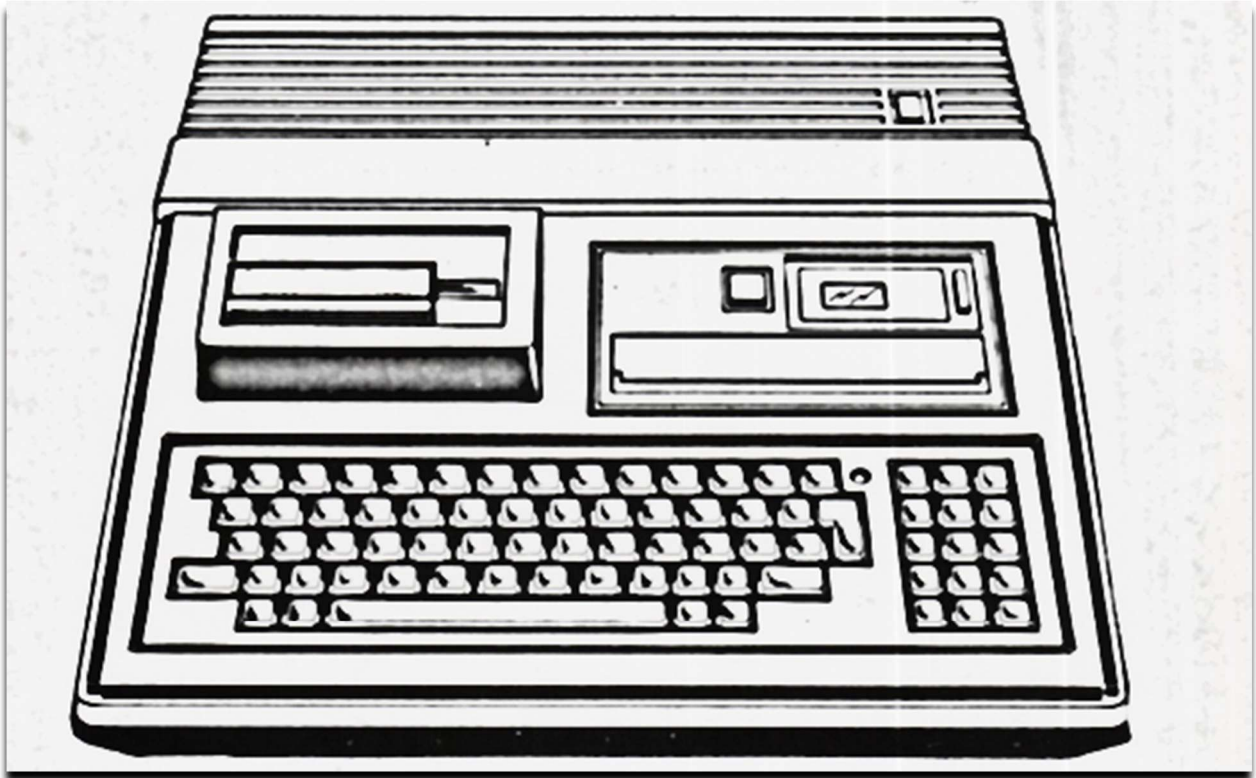


Service note
PHILIPS NEDERLAND EINDHOVEN
by TECHNICAL SERVICE

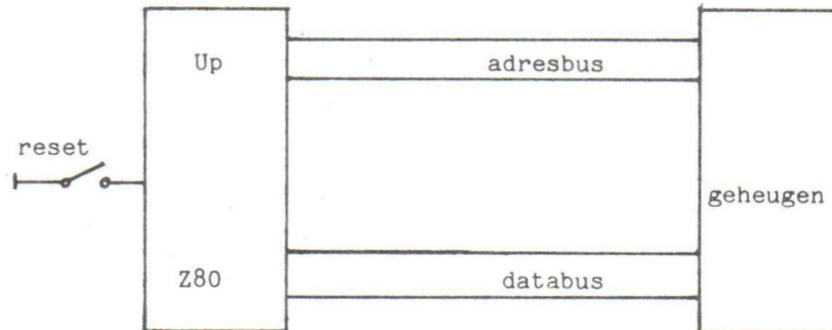
The Philips P2000T home computer



HARDWARE ASPECTS AND REPAIR POSSIBILITIES

The heart of a microcomputer is formed by a so-called microprocessor; in the P2000 T, this is a Z80. This processor has an 8-bit data bus and a 16-bit address bus.

The general operation of the processor can be explained as follows:



If the processor is reset, the address bus will contain the address 0 (zero). The memory receives this via the address bus and "responds" by putting the information on the data bus, which is located in memory location 0.

The processor reads this information from the data bus and, depending on the content of the information, performs a certain action. (The information is an instruction for the processor).

Once the processor has completed the action, it increments the address counter by one. The address bus now addresses location 1 of the memory, and the processor reads the information from address 1, executes it, increments the address counter again, and so on.

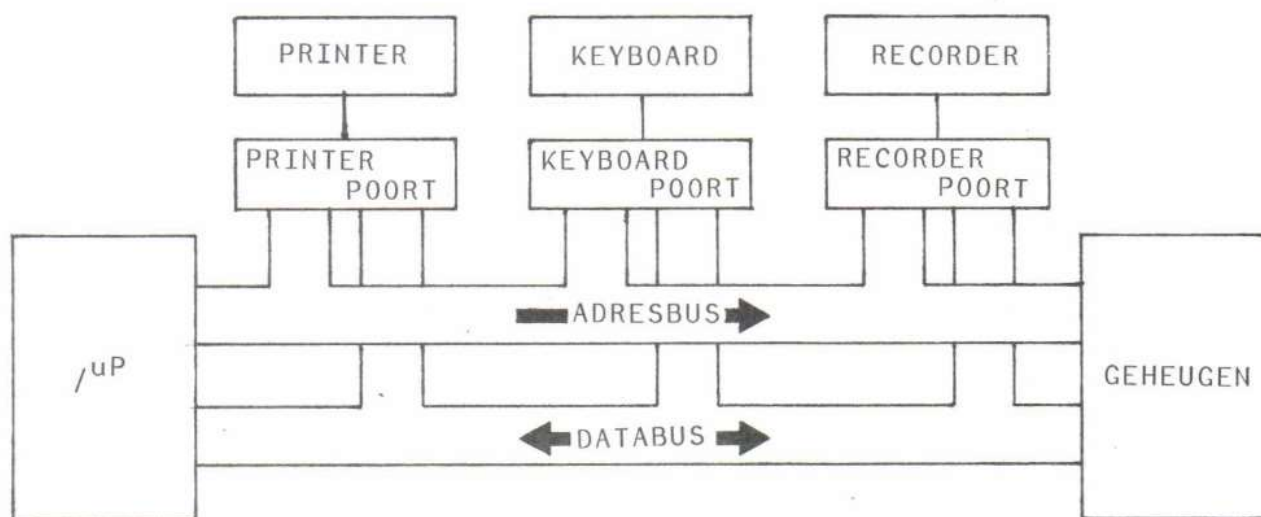
If this cycle is not interrupted, all 65536 locations of the memory will be accessed in this manner. The speed at which this occurs depends on the clock frequency of the processor (2.5 MHz in the P2000 T) and the complexity of the retrieved instruction. (The processor struggles more with certain instructions than with others.)

The format of the instructions, such as those concerning the data bus, is an 8-bit configuration of ones and zeros. This is commonly denoted in hexadecimal, and an 8-bit word of this kind is called a byte. The memory is therefore composed of locations that can precisely hold one such byte.

Which action the processor needs to perform after reading a specific instruction is determined by the structure of the processor itself. Each type of processor has its own set of instructions, known as the instruction set. The memory always contains a series of instructions, a program, written in these processor-specific commands, the machine language.

Considering figure 1 again, it represents a circuit that will undoubtedly work, but the "outside world" cannot see it. We want to have the ability to make something visible on a screen or modify the memory using a keyboard.

To achieve this, we expand the diagram with so-called input and output ports.



The operation of such a port is essentially similar to a memory location (memory-mapped I/O). The processor places an address on the bus, and the interface "recognizes" it as its own and processes the information on the data bus.

Note that, for example, with the recorder port, both information can be written to the recorder and read from it (bidirectional).

If the memory now contains the correct program (the software), a complete computer is described in the above diagram.

However, if we want to make a change in the software, we will have to modify certain 280 instructions at specific locations in the memory. This requires in-depth knowledge of both the processor and the computer system.

To solve this problem, higher-level programming languages and translators are used. The principle is simple: You enter a command in "plain language," for example, through the keyboard, and a translator (interpreter) translates this command into instructions that the processor can understand and stores them in memory. Such a translator is often simply part of the memory and is controlled by the processor.

The memory consists partly of pre-installed software by the manufacturer (e.g., control of various ports) and partly of free space that the user can utilize.

The fixed part is in (E)PROM, and the free space is RAM. The interpreter is part of the fixed memory but will store the translated commands in RAM#.

In this way, it is achieved that one can write programs without knowledge of the machine language of the processor.

* The translated commands are not immediately stored as instructions for the 280 in the general memory, but in an interpreter code, which, after executing the program, result in the execution of subprograms within the interpreter program itself.

When using a translation program that operates based on a so-called compiler, the entered commands are directly stored as instructions for the microprocessor in the memory.

Let's now consider how everything is implemented in the P2000 T. In this regard, we need to keep the following in mind: The processor is capable of addressing 2¹⁶ memory locations, often referred to as 64K (1K = 1024 locations). As described above, certain parts of it are reserved for control functions and interpreter. This leads to the following memory layout:

Address :

0x0000 - 0x0FFF	(4K)	- Monitor program, fixed control routines
0x1000 - 0x4FFF	(16K)	- Interpreter
0x5000 - 0x5FFF	(4K)	- Video RAM
0x6000 - 0x9FFF	(16K)	- User RAM standard
0xA000 - 0xDFFF	(16K)	- User RAM optional
0xE000 - 0xFFFF	(8K)	- Not used in P2000 T

Furthermore, the following input/output ports are utilized:

Port 00 - 09 - Keyboard

10 - 1F - Output to recorder and printer

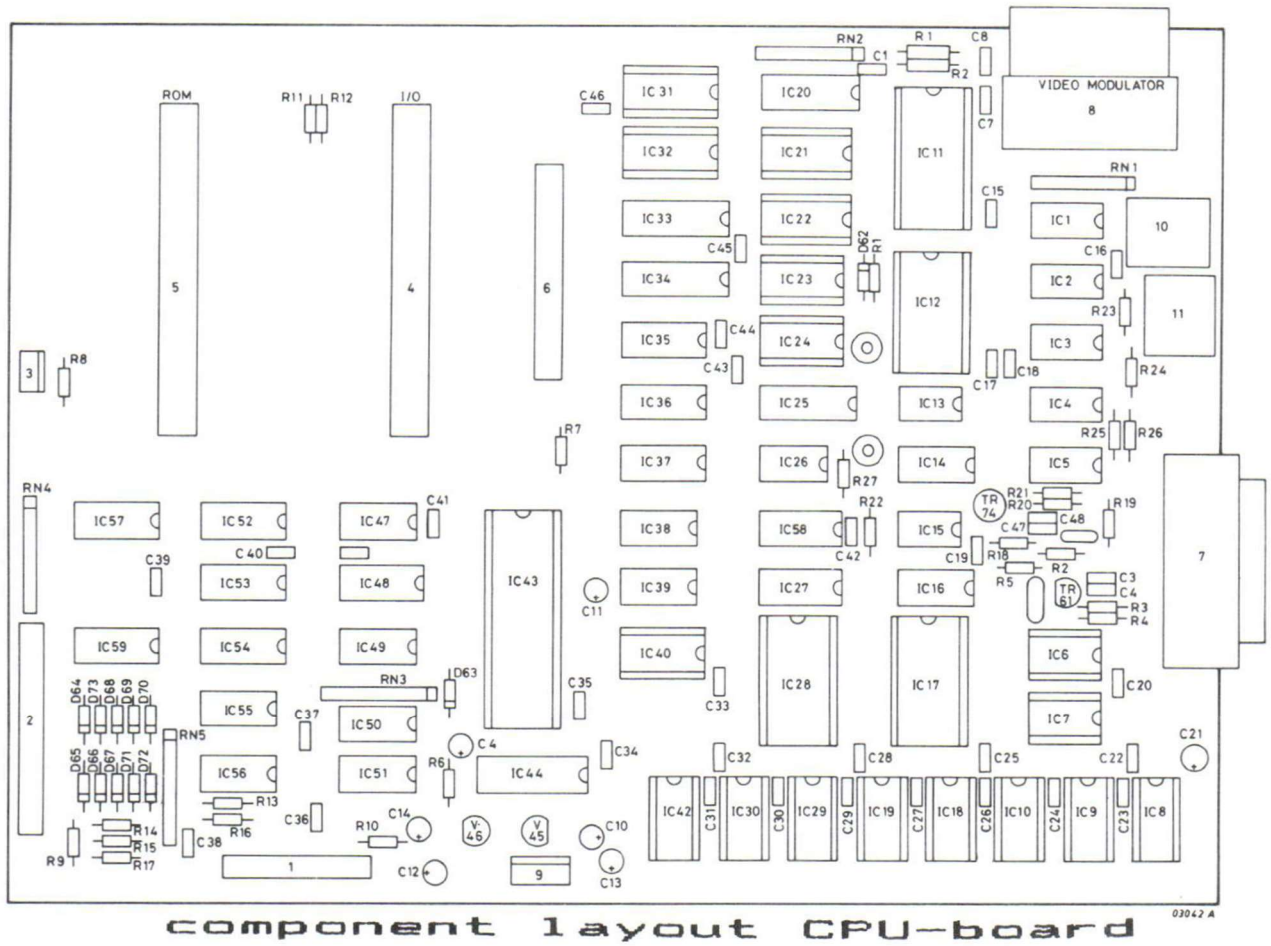
20 - 2F - Input from recorder and printer

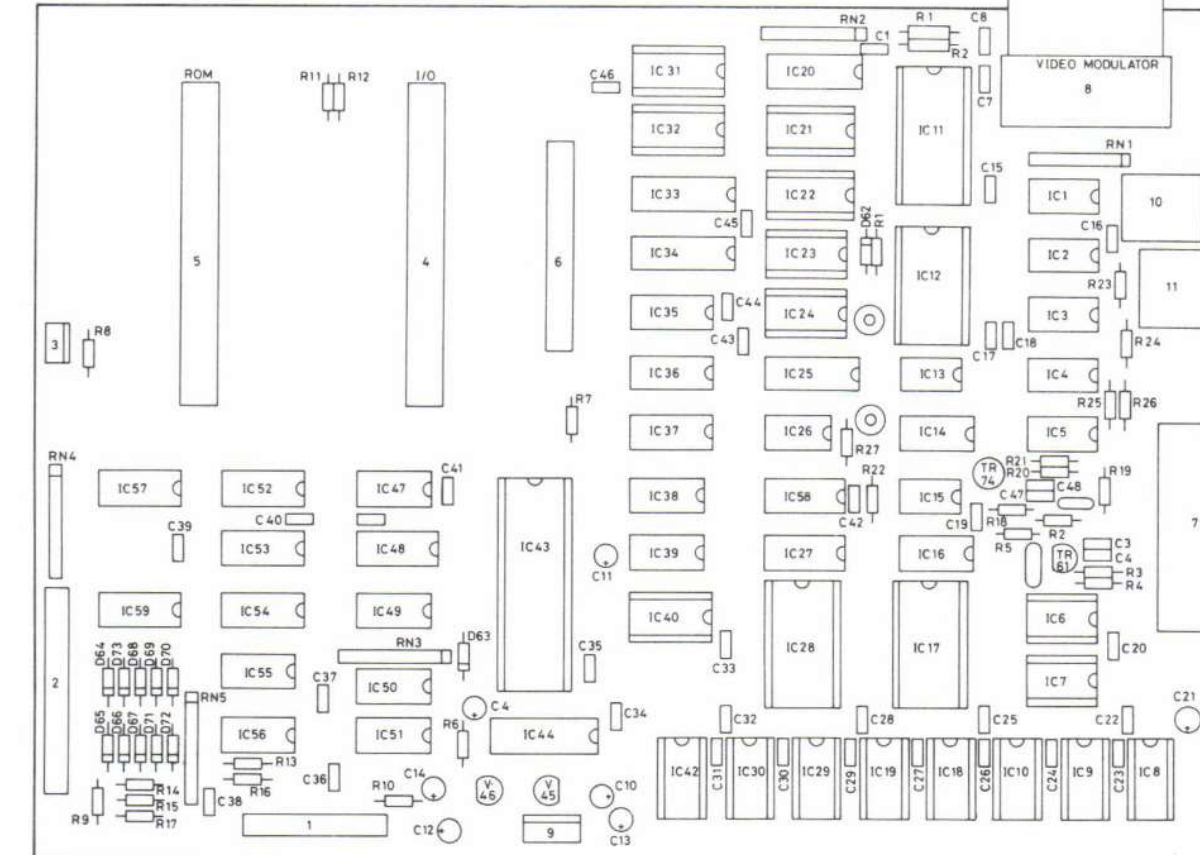
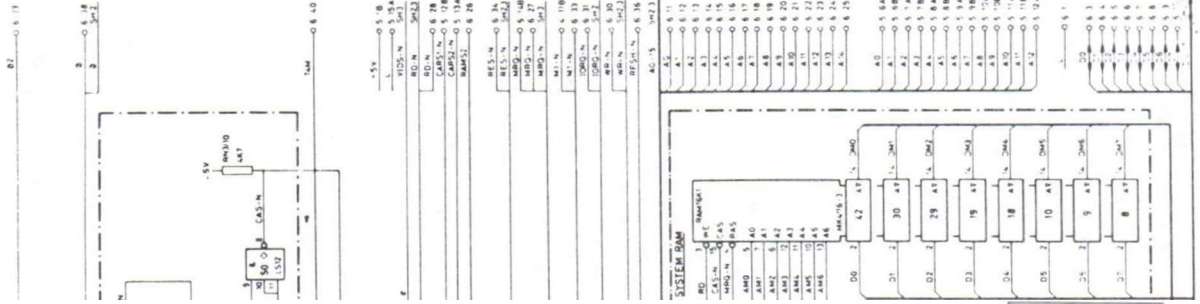
30 - 3F - Scroll register video

50 - 5F - Beeper

With this knowledge and the use of some commands in the programming language BASIC, it is possible to test certain parts of the computer.

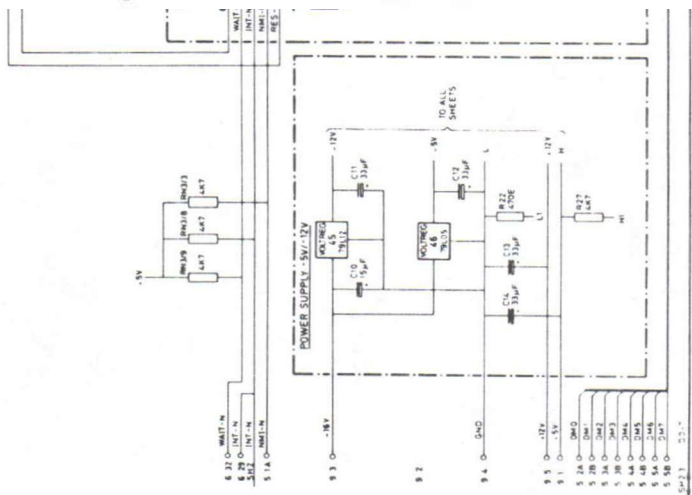
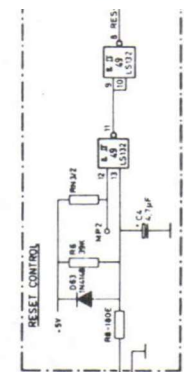
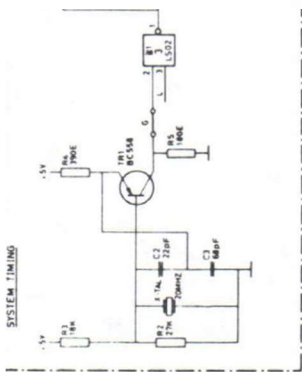
Using OUT instructions, one can write information to a port, and using INP, a port can be read. The commands POKE and PEEK provide the ability to write to or read from a specific location in memory.

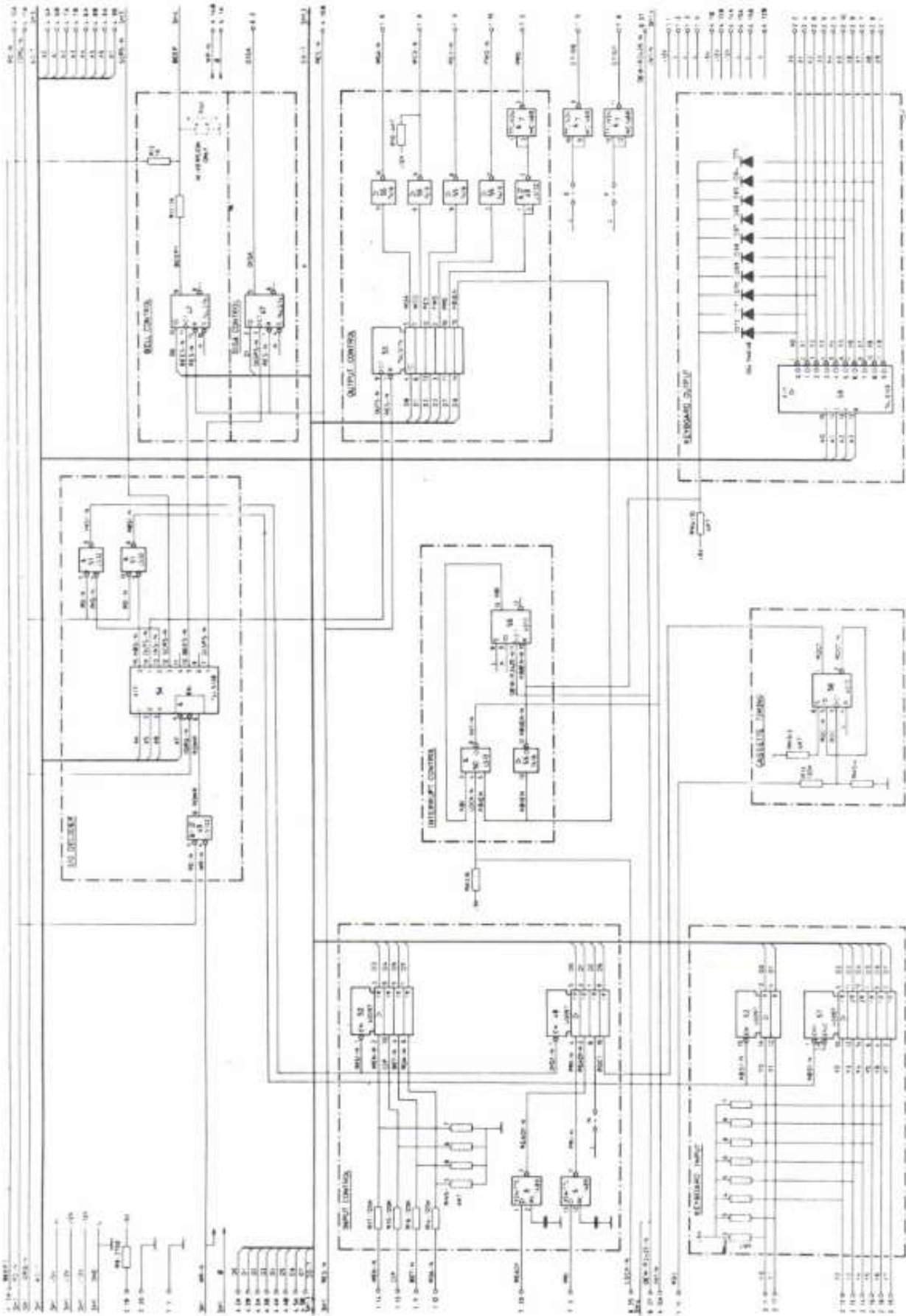




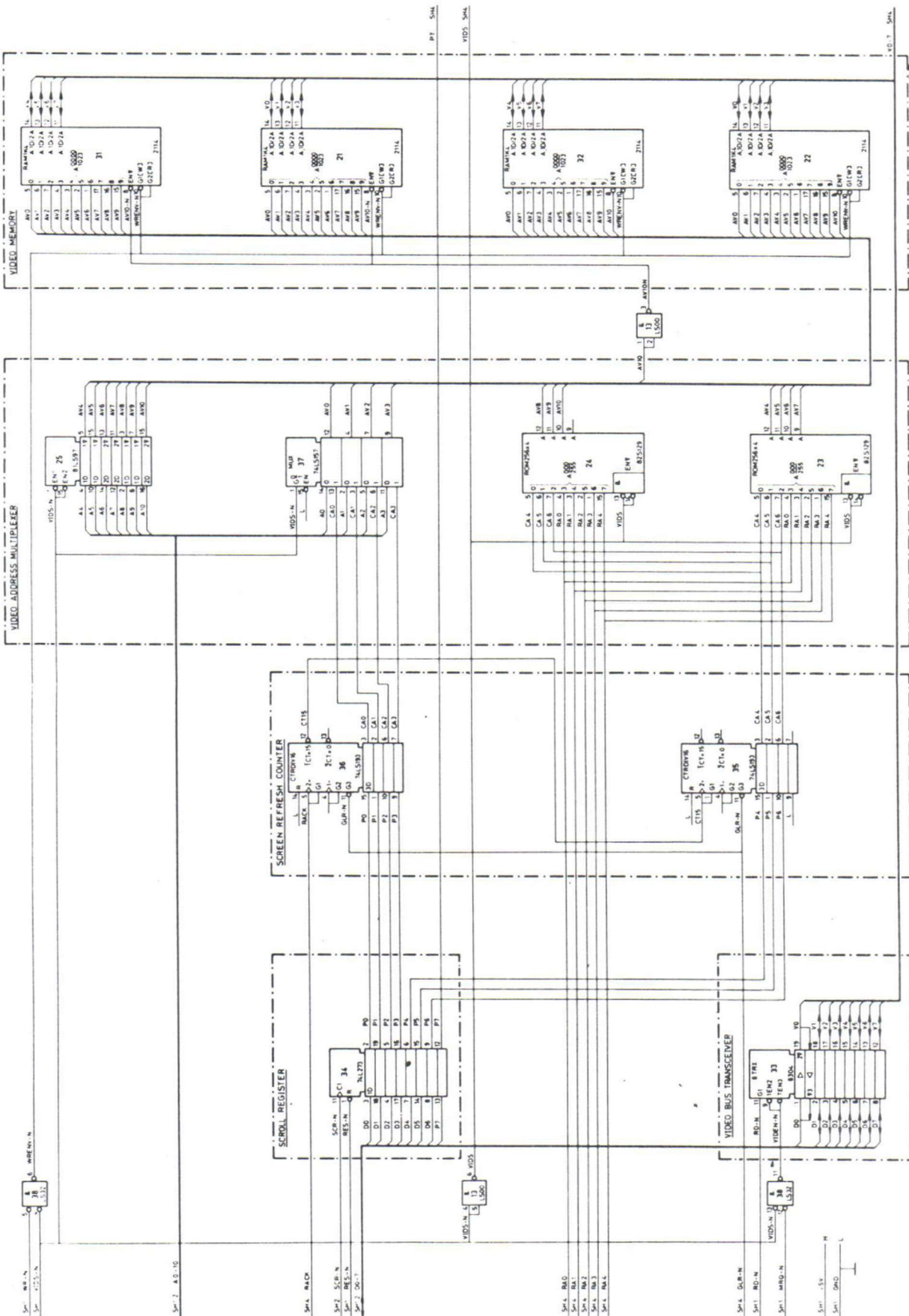
component layout CPU-board

02042 A

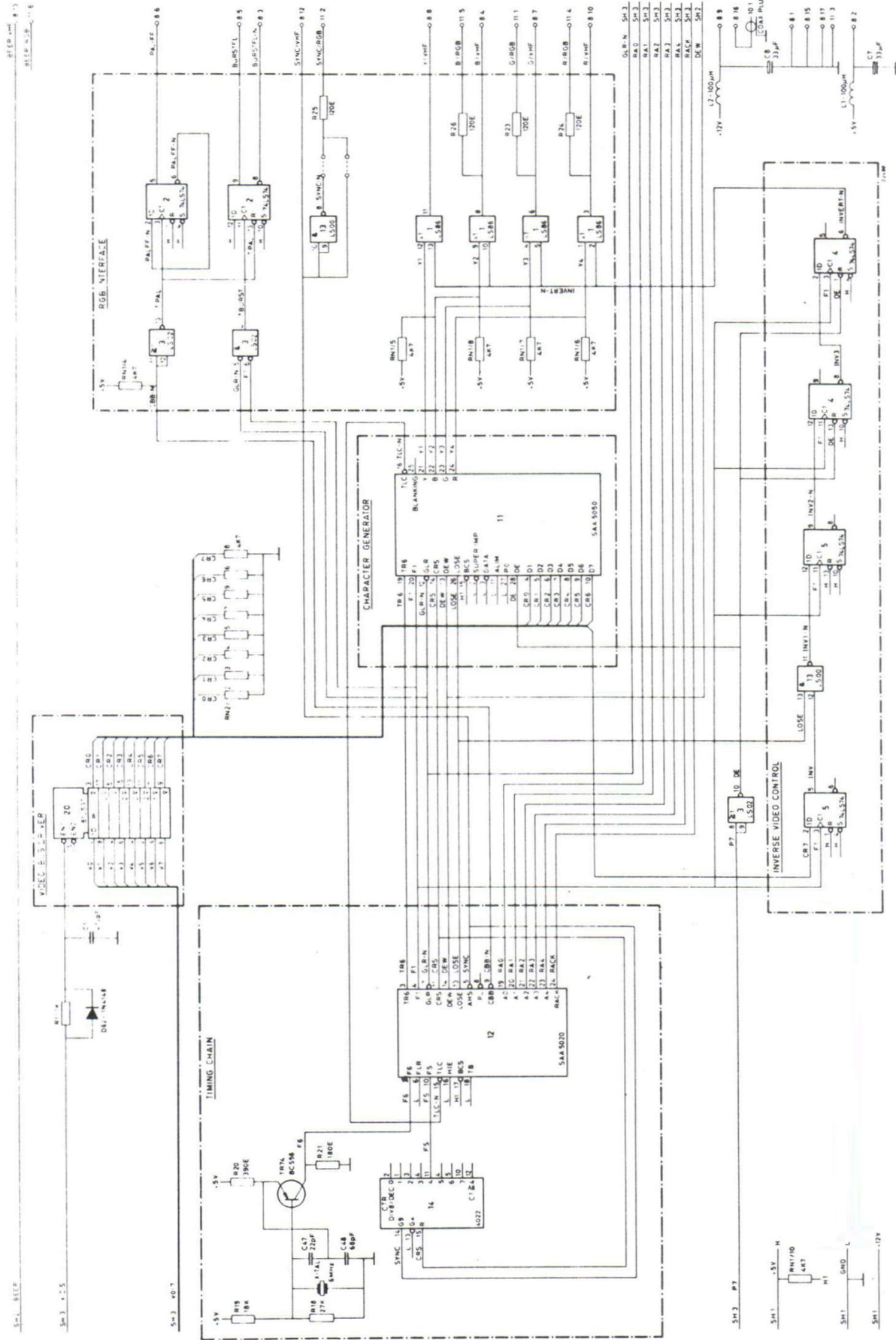




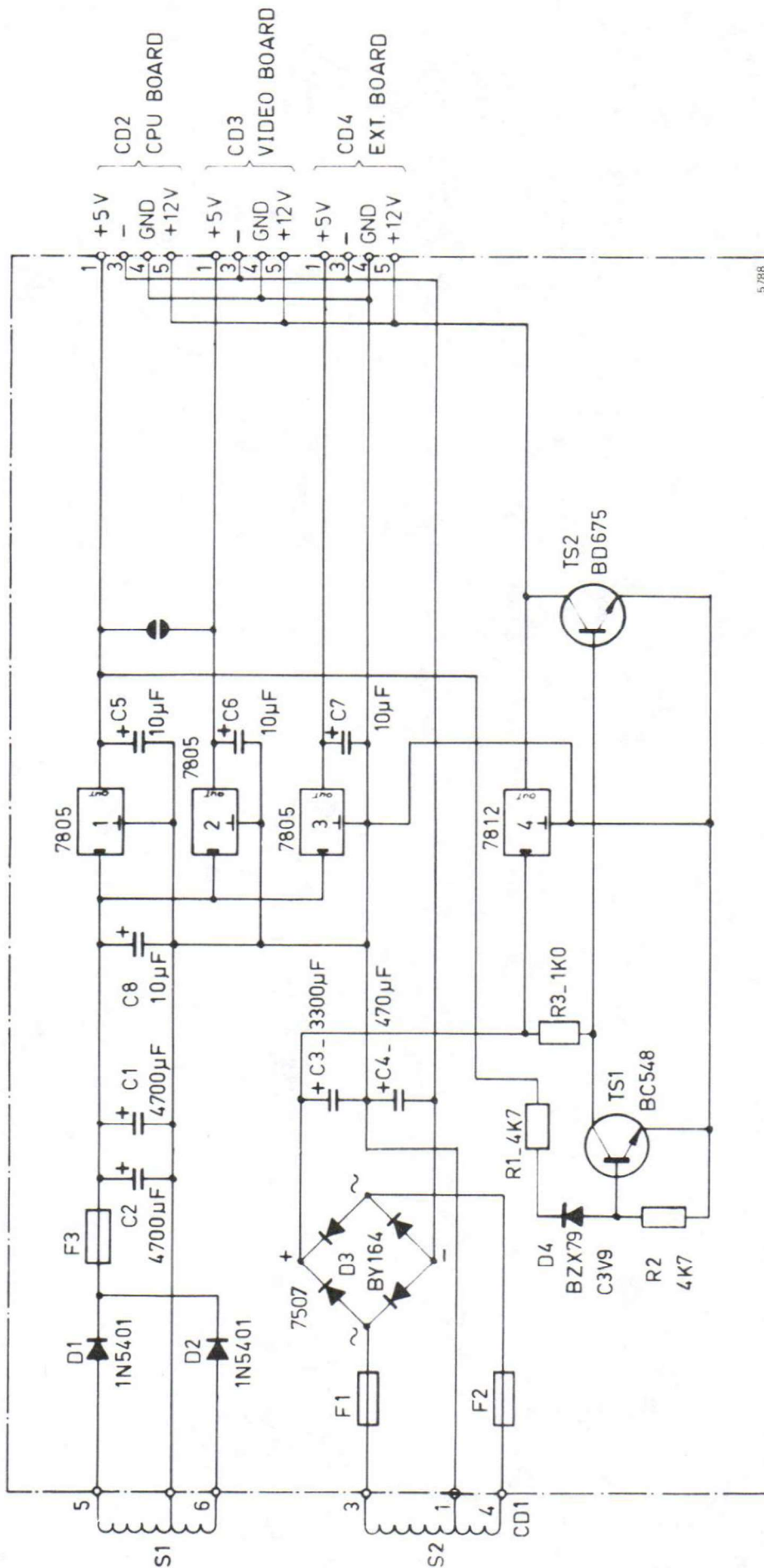
CPU-board deel 2

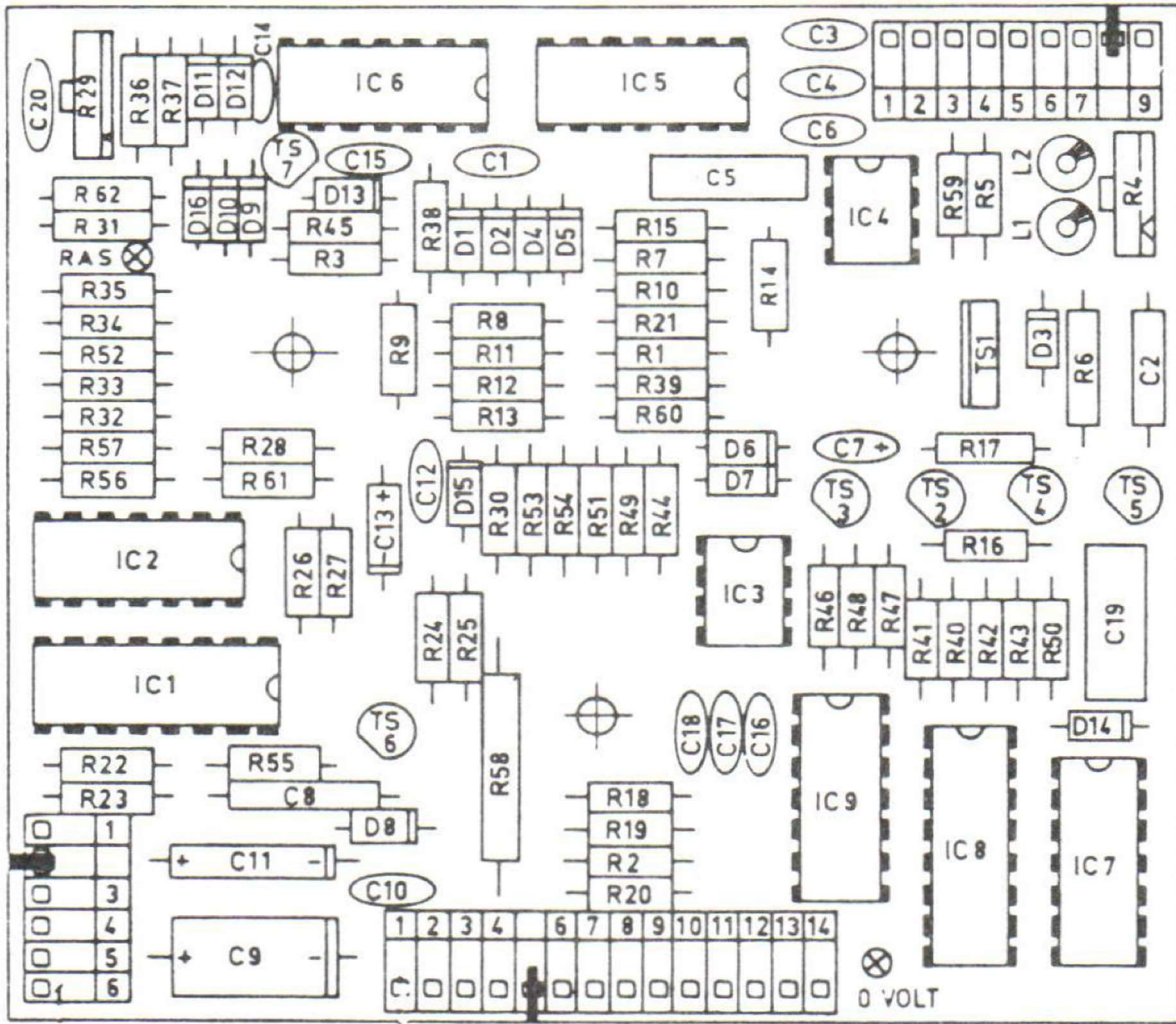


CPU-board deel 3



CPU-board deel 4





component layout
cassette recorder

BITS				Col		Row										
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀									
				0	1	2	2a	3	3a	4	5	6	6a	7	7a	
0	0	0	0	0	NUL*	DLE*			0		Ⓜ	P	-		p	
0	0	0	1	1	Alpha ⁿ Red	Graphics Red			1		A	Q	a		q	
0	0	1	0	2	Alpha ⁿ Green	Graphics Green	"		2		B	R	b		r	
0	0	1	1	3	Alpha ⁿ Yellow	Graphics Yellow	£		3		C	S	c		s	
0	1	0	0	4	Alpha ⁿ Blue	Graphics Blue	\$		4		D	T	d		t	
0	1	0	1	5	Alpha ⁿ Magenta	Graphics Magenta	%		5		E	U	e		u	
0	1	1	0	6	Alpha ⁿ Cyan	Graphics Cyan	&		6		F	V	f		v	
0	1	1	1	7	Alpha ⁿ White	Graphics White	'		7		G	W	g		w	
1	0	0	0	8	Flash	Conceal Display	(8		H	X	h		x	
1	0	0	1	9	Steady	Contiguous Graphics)		9		I	Y	i		y	
1	0	1	0	10	End Box	Separated Graphics	*		:		J	Z	j		z	
1	0	1	1	11	Start Box	ESC	+		;		K	←	k		¼	
1	1	0	0	12	Normal Height	Black Background	,		<		L	½	l			
1	1	0	1	13	Double Height	New Background	-		=		M	→	m		¾	
1	1	1	0	14	SO	Hold Graphics	.		>		N	↑	n		-	
1	1	1	1	15	SI	Release Graphics	/		?		O	#	o		■	

Control characters shown in columns 0 and 1 are normally displayed as spaces

* These control characters are reserved for compatibility with other data codes.

** These control characters are presumed before each row begins.

Codes may be referred to by their column and row eg. 2/5 refers to %

□ Character rectangle
Black represents display colour.
White represents background.

Storingzoeken P2000 T

We can divide the P2000 T into several segments and test them using special software.

This software is included in a test module, which can be inserted into the first slot of the P2000.

The following tests are available:

1. RAM.

The RAM (user) memory of the P2000 is built around the memory chip 1116. This chip has 16,384 locations of 1 bit each. To process 8 bits, 8 chips are used in parallel.

Addressing the chip occurs in 2 steps (multiplexed), controlled by Row (RAS) and Column (CAS) signals. The test fills all locations using a specific formula and then reads back the information. If errors occur, they will be displayed on the screen.

2. V24.

On the back of the P2000 T, there is a 25-pin connector with the following connections:

2 - serial data in

3 - serial data out

5,6 - +10V

7 - ground (gnd)

20 - input data terminal ready (DTR)

The connector is used for connecting a modem and printer, and the handling of signals is controlled by the software through input/output ports 10 and 20. This test can be used to verify the proper functioning of the inputs and outputs.

3. Keyboard.

During normal use of the P2000 T, a keyboard scan is performed every 20 ms via a software subroutine initiated by the video pulse DEW. By controlling ports 00 through 09, it checks if a key is pressed and, if so, which key.

The test requires all keys to be pressed one by one and displays the result on the screen.

4. Video.

The video component is an independent module of the P2000 T. On one hand, the processor writes information intended for the screen into the video RAM, and on the other hand, the video component reads from the RAM and displays the information on the screen. The video component operates at its own clock frequency and is built around the ICs SAA 5020 and SAA 5050, which are also used in television receivers with teletext.

The test displays all used symbols and colors on the screen, with a width of 80 characters.

5. Tape drive.

The MDCR (Mini Digital Cassette Recorder) is a unit on which the information from the P2000 T is digitally stored (frequency modulation). The processor also generates the signals for motor control and transmits them through a port.

During the test, all functions of the recorder are tested.