
RCSL No: 30-M282

Edition: June 1981

Author: Aage Jørgensen

Title:

CPU820

Technical Manual

Keywords:

CPU820, RC8000-50, RC8000-55

Abstract:

This manual contains technical information including functional description and logic diagrams for the CPU820 processing unit.

(244 printed pages)

Copyright © 1981, A/S Regnecentralen af 1979
RC Computer A/S

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

TABLE OF CONTENTS	PAGE
1. INTRODUCTION	1
2. FUNCTIONAL DESCRIPTION	2
2.1 CPU821 Data Paths	2
2.1.1 General Registers	2
2.1.2 Q-Register	2
2.1.3 Arithmetic Logic Unit	4
2.1.4 Scratchpad	4
2.1.5 Constant Memory	4
2.1.6 Immediate Operand	4
2.1.7 Loop Counter	5
2.1.8 Half Word Manipulator	5
2.1.9 Exception Control	5
2.1.10 Instruction Counter	6
2.1.11 Displacement Register	6
2.1.12 Relative Address Register	6
2.1.13 Switch Register	6
2.1.14 CPU Status Register	7
2.1.15 Control Output Register	7
2.1.16 Logical Address Status Register	8
2.1.17 CPU Bus	9
2.2 CPU822 Data Paths	9
2.2.1 Bus and Cache Interface Unit	9
2.2.1.1 Logical Address Register	9
2.2.1.2 Unit Function Register	9
2.2.1.3 Base Register	12
2.2.1.4 Address Limit Registers	12
2.2.1.5 Prefetch Instruction Counter	13
2.2.1.6 System Bus Address Register	13
2.2.1.7 Data Out Register	13
2.2.1.8 Data In Register	13
2.2.1.9 Prefetch Instruction Register	14
2.2.1.10 I/O Status Register	14
2.2.1.11 CAM Control Register	16
2.2.2 Interrupt Control Unit	17
2.2.2.1 Interrupt Limit Registers	17
2.2.2.2 Interrupt Level Register	17
2.2.2.3 Interrupt Register	18

TABLE OF CONTENTS (continued)	PAGE
2.2.3 Real Time Clock	18
2.2.3.1 Real Time Clock Register	18
2.2.4 Technician Console Interface	19
2.2.4.1 Technician Console Status Register	19
2.2.4.2 Technician Console Data In Register	20
2.2.4.3 Technician Console Data Out Register	20
2.3 Microprogram Control Unit	20
2.3.1 Microinstruction Address Register	21
2.3.2 Subroutine Return Stack	21
2.3.3 Microprogram Jump Address	23
2.3.4 Microprogram Interrupt Address	23
2.3.5 Address Calculation Decoding Map	23
2.3.6 Instruction Execution Decoding Map	24
2.4 Microinstructions	24
2.4.1 P Field = MIR(0)	25
2.4.2 NEXT Field = MIR(1:4)	25
2.4.3 CL Field = MIR(5,6)	27
2.4.4 F Field = MIR(7:9)	28
2.4.5 ALU-D Field = MIR(10,11)	28
2.4.6 ALU-OP Field = MIR(12:14)	28
2.4.7 ALU-FUNC Field = MIR(15:17)	29
2.4.8 ALU-C Field = MIR(18,19)	29
2.4.9 A Field = MIR(20:24)	29
2.4.10 B Field = MIR(25:29)	30
2.4.11 UNIT FUNC/TEST Field = MIR(30:25)	30
2.4.12 INT/EXT REG DEST Field = MIR(42:47)	31
2.4.13 T Field = MIR(30)	32
2.4.14 COND SELECT Field = MIR(31:35)	33
2.4.15 Format 0 : Read/Load Scratchpad	34
2.4.16 Format 1 : Read/Load Register	34
2.4.17 Format 2 : Read Constant	35
2.4.18 Format 3 : Read External Register	36
2.4.19 Format 4 : Half-Word Manipulator	37
2.4.20 Format 5 : Load Immediate	38
2.4.21 Format 6 : Conditional Jump	38
2.4.22 Format 7 : Shift, Multiply and Divide	39

TABLE OF CONTENTS (continued)	PAGE
3. BLOCK DIAGRAMS	43
4. LOGIC DIAGRAMS	51
4.1 CPU821 Logic Diagrams	51
4.2 CPU822 Logic Diagrams	127
5. TIMING DIAGRAMS	199
5.1 CPU821 Timing Diagrams	199
5.2 CPU822 Timing Diagrams	207
6. PROM's	214
6.1 Microprogram PROM Position List	214
6.2 ROA307 Listing	215
6.3 ROA308, ROA309 and ROA310 Listing	216
6.4 ROA311 Listing	217
6.5 ROA312 Listing	218
6.6 ROA313 Listing	220
7. PLUG LISTS AND CABLES	222
7.1 CPU821 Plug Lists	222
7.2 CPU822 Plug Lists	226
7.3 Internal Cables, CBL826 and CBL904	231
8. ASSEMBLY DRAWINGS	233

1. INTRODUCTION

1.

CPU820, which is the basic processing unit for RC8000 Model 50 and Model 55, consists of two modules (printed circuit boards), CPU821 and CPU822. The performance of the CPU820 may be increased by extending it with a cache memory module, CAM801/803, and/or a floating point unit, FPU801, as shown on fig. 1.

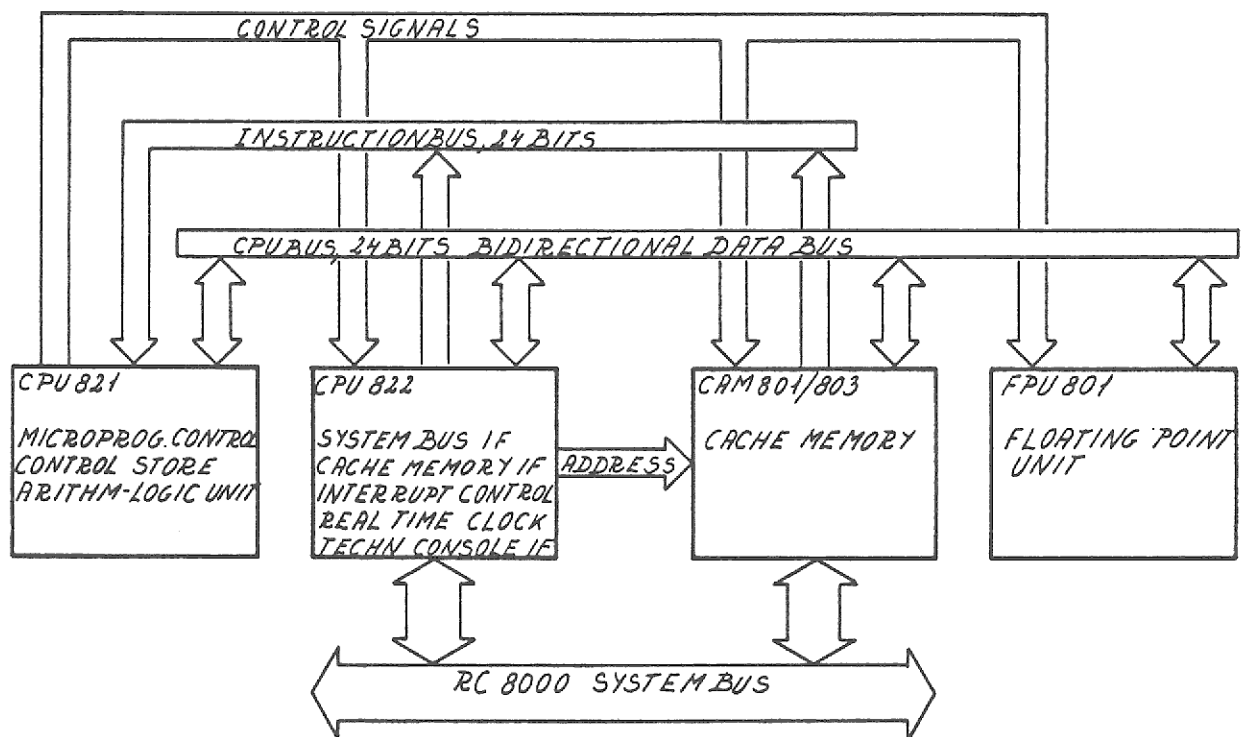


Figure 1: CPU820 extended with cache memory and floating point unit.

The processing unit modules are interconnected by means of a PCB backplane bus consisting of a 24-bit bidirectional data bus (CPUBUS), a 24-bit instruction bus and a number of control signals. Communication between the processing unit, the main memory and the device controllers takes place via the RC8000 System Bus.

This manual describes the CPU820 modules, CPU821 and CPU822. The CAM801/803 and FPU801 modules are described in separate manuals.

2. FUNCTIONAL DESCRIPTION

2.

2.1 CPU821 Data Paths

2.1

The principal data paths in the CPU821 are 24-bit wide and are shown on the block diagram on fig. 2. An array of 6 IDM2901A-1 4-bit slice processing elements constitutes the kernel of the CPU821 data path structure. It contains the General Registers (accumulators), the Q-register, and an arithmetic logic unit. The 2901 array receives data from external registers and operators via the SBUS (Source Bus), which is a tri-state bus. Data to external registers and operators are transferred via the DBUS (Destination Bus).

The following subsections give a short description of the CPU821 registers and operators.

2.1.1 General Registers

2.1.1

The 16 General Registers are located in a dual-port RAM in the 2901. The contents of 2 General Registers may simultaneously be accessed via the A-port, GRA, and the B-port, GRB. GRA and GRB may be used as inputs to the ALU and GRA may be transferred directly to the DBUS. The General Register may be loaded with the output from the ALU. A shift network at the input to the General Registers may pass or shift the ALU output 1 bit position left or right before it is loaded.

2.1.2 Q-Register

2.1.2

The Q-Register which is located in the 2901 may be used as an accumulator and as an extension to the General Registers in shift operations for shifting 48-bit operands. In the latter case Q holds the least significant 24-bit of the 48-bit operand. It is only possible to shift Q in conjunction with a General Register.

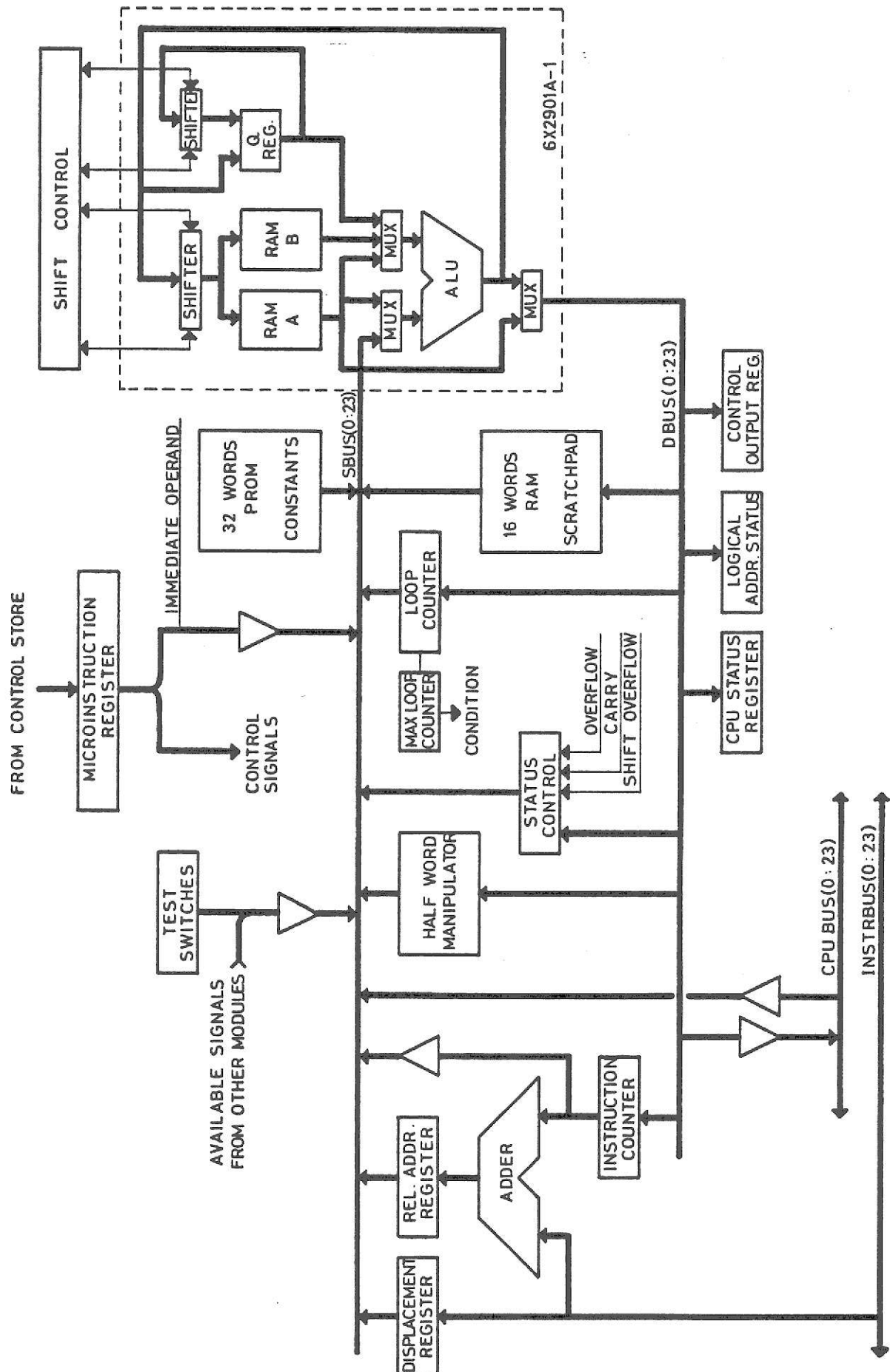


Figure 2: CPU821 data paths.

2.1.3 Arithmetic Logic Unit

2.1.3

The Arithmetic Logic Unit, ALU, inside the 2901 can perform 3 arithmetic and 5 logic operations on 2 operands. The General Registers, the Q-Register, the SBUS and ZERO can be selected as operands for the ALU in 8 different combinations. The ALU output, F, can be transferred to the General Registers, the Q-Register and the DBUS.

2.1.4 Scratchpad

2.1.4

The Scratchpad (SCRATCHP) is a register file with 16 24-bit words. It can be used as both source and destination but not within the same microcycle. When it is used as source the complement of the data loaded into the addressed location is transferred to the SBUS.

2.1.5 Constant Memory

2.1.5

The Constant Memory (CONSTANT) is a read only memory with 32 24-bit words, which is used to hold frequently used constants. It can only be used as a source.

2.1.6 Immediate Operand

2.1.6

Bits (20:43) of the Microinstruction Register (MIR) may be transferred to the SBUS and used as a 24-bit immediate operand in the current microinstruction.

$$SBUS(0:23) = MIR(20:43)$$

2.1.7 Loop Counter

2.1.7

The Loop Counter is a 24-bit counter addressable as both source and destination register. The primary function of LC is to count the number of times a microprogram loop is executed. Execution of a microinstruction where the NEXT-field specifies a LOOP RETURN will increment LC by 1. The most significant bit of LC LC(0), is a test condition.

An 8-bit counter, LCMAX(0:7), is associated LC. This counter may be used to control that a loop is executed maximum 49 times. LCMAX cannot be addressed from the microprogram. It is loaded with -48 (dec.) when LC is loaded and incremented by 1 simultaneously with LC. The testcondition MAXLOOP indicates that $LC \geq 0$ or $LCMAX \geq 0$.

MAXLOOP = LC(0)! -,LCMAX(0)

2.1.8 Half Word Manipulator

2.1.8

The Half Word Manipulator (HWM) is a combinatorial network which performs operations on data from the DBUS and transfers the result to the SBUS. The HWM performs such operations as 12-bit sign extension, half word swapping, and half word masking. One of the microinstruction formats is assigned to control the operation of the HWM.

2.1.9 Exception Control

2.1.9

Exception Control (EXC) is a combinatorial network which is used to insert various status bits into bit positions 22 and 23 of a 24-bit word. EXC receives input from the DBUS and transfers the modified word to the SBUS. EXC is addressable as 4 source registers. Each of the addresses defines an operation.

2.1.10 Instruction Counter

2.1.10

The Instruction Counter (IC) is a 24-bit counter/register addressable as both source and destination. IC(0) and IC(23) is always 0. Execution of a microinstruction where the NEXT field specifies EXECUTE, MIR(1:4) = 1111, will increment IC by 2.

2.1.11 Displacement Register

2.1.11

The Displacement Register (DISP) is a 24-bit source register. DISP contains the displacement of the RC8000 instruction which is being executed. Execution of a microinstruction where the NEXT field specifies NEXT INSTR transfers the instruction displacement from the Instruction Bus, INSTRBUS, to DISP.

$DISP(0:23) := 12 \text{ ext } INSTRBUS(12) \text{ con } INSTRBUS(12:23)$

2.1.12 Relative Address Register

2.1.12

The Relative Address Register (RELADDR) is a 24-bit source register containing $DISP+IC$. Execution of a microinstruction with the NEXT field = NEXT INSTR will load RELADDR.

$RELADDR := 12 \text{ ext } INSTRBUS(12) \text{ con } INSTRBUS(12:23) + IC.$

2.1.13 Switch Register

2.1.13

The Switch Register (SWITCH) is an 8-bit pseudo register which contains the state of the MODE switches, the TEST switch and the availability signals from other modules in the CPU chassis. The bits are assigned in the following way.

SWITCH(16) = POWER RESTART MODE
 SWITCH(17) = POWER-UP TEST MODE
 SWITCH(18) = CONTINUOUS TEST MODE
 SWITCH(19) = CONSOLE TC MODE
 SWITCH(20) = TEST ON
 SWITCH(21) = CPU822 NOT AVAILABLE
 SWITCH(22) = CAM801 NOT AVAILABLE
 SWITCH(23) = FPU801 NOT AVAILABLE

SWITCH is only addressable as a source register, and the contents are transferred to the SBUS as defined below.

SBUS(0:15) is undefined
 SBUS(16:23) = SWITCH > 16:23)

2.1.14 CPU Status Register

2.1.14

The CPU Status Register (CPUST) is a 6-bit destination register. CPUST is used to hold a copy of bits from the CPU Status Word which are used in the instruction decoding and for control purposes. CPUST is loaded from the DBUS as defined below.

MON MODE: = DBUS(0)
 ESC MODE: = DBUS(1)
 AFTER AM: = DBUS(2)
 AFTER ESC: = DBUS(3)
 INT MASK: = DBUS(4)
 DISABLE: = DBUS(20)

2.1.15 Control Output Register

2.1.15

The Control Output Register (CONTROLOUT) is an 8-bit destination register which is used for different control purposes. The register is loaded with DBUS(16:23). The contents of the register are used in the following way:

BIT	CONTROL FUNCTION
16	CPUSYSRST. Controls the SYSTEM RESET signal on the System Bus.
17	RUN. Controls the RUN lamp on the Operators Control Panel (OCP).
18	AUTOLOADING. Controls the AUTOLOAD lamp on the OCP.
19	-, SINGLEINSTR. Used to control single instruction execution. The signal generates an interrupt (CPUINTR) when it is 0.
20	-, RSTREMAUTO. A 0 resets the Remote Autoload interrupt flip-flop.
21	-, CPULAMP. Controls the lamp, CPU, on the CPU821 front panel.
22	-, MEMLAMP. Controls the lamp, MEMORY, on the CPU821 front panel.
23	-, CACHELAMP. Controls the lamp, CACHE, on the CPU821 front panel.

2.1.16 Logical Address Status Register

2.1.16

The Logical Address Status Register (LASTAT) is a 5-bit destination register, which is used to store status information concerning the logical address. LASTAT is automatically updated when the Logical Address Register (LOGADDR) on the CPU822 is loaded. In addition it has its own destination address. The following status bits are stored in the register:

```

-,WADDR: = if F(0:20) = 0 then 0 else 1.
          F is the result from the 2901 ALU.
EA(0):   = DBUS(0)
EA(21):  = DBUS(21)
EA(22):  = DBUS(22)
ODD:     = DBUS(23)

```

2.1.17 CPU Bus

2.1.17

The CPU Bus (CPUBUS) is a 24-bit bidirectional backplane bus which is used to transfer data between modules (CPU, Cache Memory, Floating Point Unit) installed in the CPU chassis. Data may be transferred from the CPUBUS to the SBUS and from the DBUS to the CPUBUS. Within the same microinstruction it is only possible to transfer data in one direction on the CPUBUS. The data flow on the CPUBUS is controlled from the CPU821, which transfers source and destination addresses to other modules. Source and destination addresses are generated by microinstruction fields.

2.2 CPU822 Data Paths

2.2

The CPU822 consists of 4 functional units which operate asynchronously compared to the microinstruction execution. The units are: the Bus and Cache Interface Unit, the Interrupt Control Unit, the Real Time Clock and the Technicians Console Interface. Fig. 3 shows the CPU822 data paths.

2.2.1 Bus and Cache Interface Unit

2.2.1

The Bus and Cache Interface Unit (BCI) perform address limit check, convert logical addresses to physical addresses, perform memory and I/O read operations and prefetch instructions. If the Cache Memory (CAM) is installed in the system the BCI will automatically direct memory read operations and instruction prefetches to the CAM. The BCI registers and the operation of the BCI are described in the following subsections.

2.2.1.1 Logical Address Register

2.2.1.1

The Logical Address Register (LOGADDR) is a 24-bit destination register which is used to hold the logical address during memory and I/O accesses.

2.2.1.2 Unit Function Register

2.2.1.2

The unit Function Register (UFR) is a 5-bit register, which is loaded with the contents of the Unit Function field of the micro-instruction. UFR is loaded simultaneously with the Logical Address Register.

UFR(1:5): = MIR(31:35)

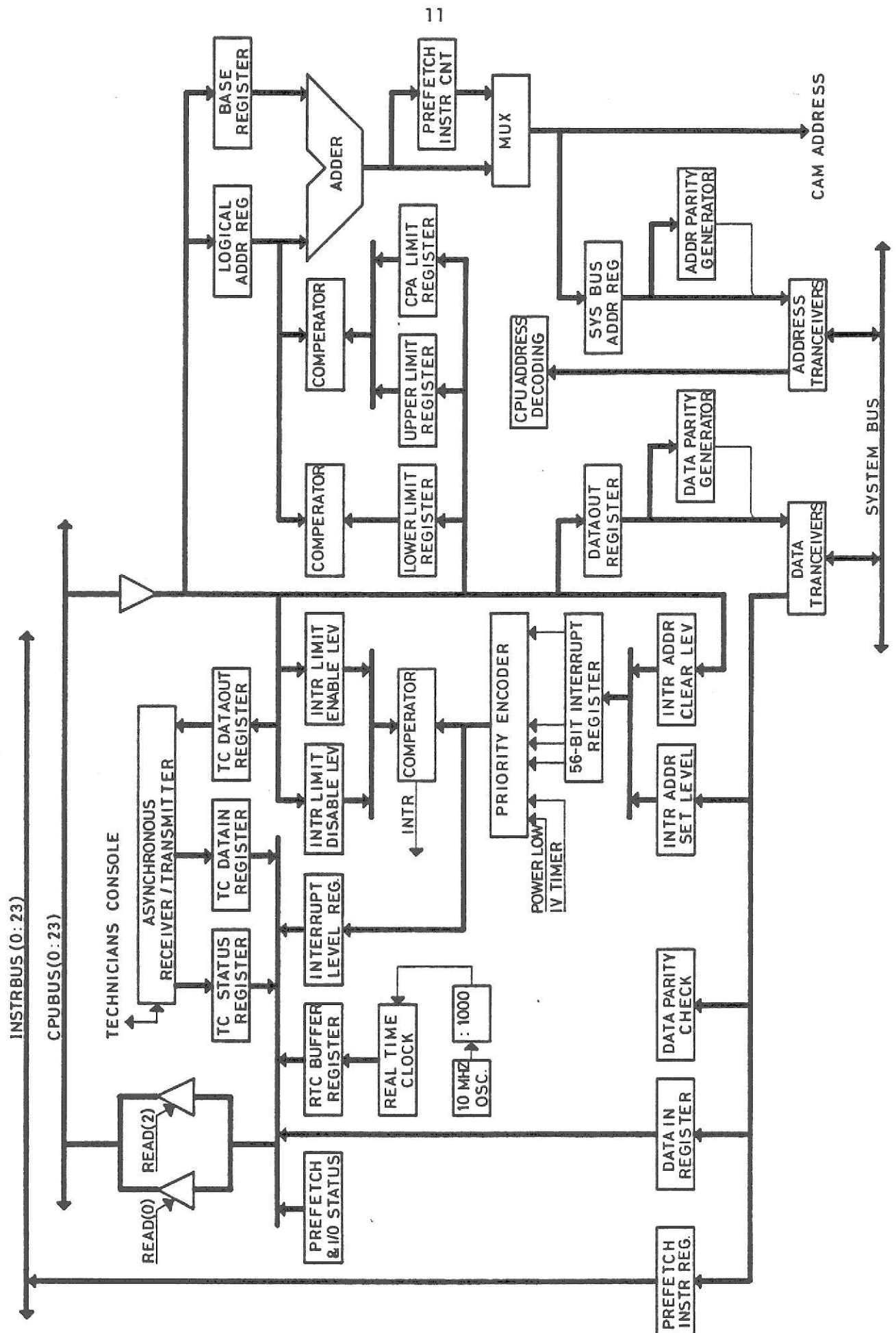


Figure 3: CPU822 data paths.

The contents of UFR together with special decoded control signals from the CPU821 control the operation of the BCI. The contents of UFR are used in the following way.

BIT	CONTROL FUNCTION
1	START. Starts a memory or I/O operation.
2	CONDITIONAL START. A memory operation is started if the current RC8000 instruction is a memory reference instruction.
3	JUMP. The word fetched from the addressed location is loaded into the Prefetch Instruction Register (PREFIR), and the Prefetch Instruction Counter (PREFIC) is loaded with the physical address.
4	WRITE. The contents of the Data Out Register (DATAOUT) are stored in the addressed location.
5	PHYSICAL ADDRESS. The logical address is used directly as a physical without limit check and base address relocation.

2.2.1.3 Base Register

2.2.1.3

The Base Register (BASE) is a 24-bit destination register. The contents of BASE may be added to LOGADDR to form the physical address.

2.2.1.4 Address Limit Registers

2.2.1.4

Access to main memory is controlled by 3 limit registers, the Lower Limit Register (LLIM), the Upper Limit Register (ULIM) and the CPA Limit Register (CPA). All three registers are 24-bit destination registers. The LLIM and ULIM registers define the logical address limits of the memory are, where both read and write access is permitted. The CPA register defines the upper limit (physical address) of the non relocatable memory area, where only read access is permitted.

2.2.1.5 Prefetch Instruction Counter

2.2.1.5

The Prefetch Instruction Counter (PREFIC) is a 22-bit counter, which is used as address source for instruction prefetch. Execution of a microinstruction, where the NEXT field specifies NEXT INSTRUCTION, will increment PREFIC by 2 and normally start an instruction fetch (prefetch may be inhibited). The PREFIC is not directly accessible from the microprogram. It may be loaded with the Physical Address (PHYSADDR) by specifying a JUMP in the UNIT FUNCTION field.

PREFIC(1:22): = PHYSADDR(1:22)

2.2.1.6 System Bus Address Register

2.2.1.6

The System Bus Address Register (SBADDR) is a 23-bit register, used to hold the System Bus address during memory and I/O accesses via the System Bus. SBADDR is not accessible from the microprogram.

2.2.1.7 Data Out Register

2.2.1.7

The Data Out Register (DATAOUT) is a 24-bit destination register which is used to hold data to be transferred from the CPU to the addressed destination. Once an output operation has been started the contents of the register must not be altered before termination of the operation.

2.2.1.8 Data In Register

2.2.1.8

The Data In Register (DATAIN) is a 24-bit source register used as buffer register for data received via the System Bus in input operations initiated by the CPU. The contents of the register are undefined from the start of an input operation until termination.

If termination is caused by a NACK or a TIME OUT, DATAIN will be loaded with the current data on the System Bus. DATAIN responds to two source addresses, one of which always selects the CPU822 DATAIN register. The other address is common to the address of a corresponding 'DATAIN' register on the cache memory module. When the cache is installed the DATAIN register on this module will automatically respond to the common address, otherwise the CPU822 DATAIN responds to the common address.

Execution of a microinstruction, which addresses DATAIN, will be delayed until termination of the memory or I/O operation, if $CL \geq 1$. CL is the Cycle Length field of the microinstruction.

2.2.1.9 Prefetch Instruction Register

2.2.1.9

The Prefetch Instruction Register (PREFIR) is a 24-bit register used as buffer register for the prefetched instruction. The contents of PREFIR are transferred to the CPU821 via the Instruction Bus (INSTRBUS). If the cache memory module is installed, a corresponding instruction register on this module, is automatically selected as source for the INSTRBUS. PREFIR is not addressable from the microprogram.

2.2.1.10 I/O Status Register

2.2.1.10

The I/O Status Register (IOSTAT) is a 16-bit source register containing status information about instruction prefetch and memory and I/O accesses. The contents of IOSTAT are transferred to the CPUBUS as described below.

CPUBUS(0:3), CPUBUS(16:19) are undefined

CPUBUS(4:15): = IOSTAT(4:15)

CPUBUS(20:23): = IOSTAT(20:23)

IOSTAT responds to two source addresses one of which unconditionally selects IOSTAT. The other address is common to IOSTAT and a corresponding status register on the cache memory module, CAMSTAT. If the cache is installed the common address selects CAMSTAT, otherwise it selects IOSTAT. Note that only bits (21:23) of IOSTAT and CAMSTAT have the same meaning. Execution of a microinstruction, which addresses IOSTAT (or CAMSTAT) will be delayed until termination of the current memory or I/O operation if $CL \geq 1$. CL is the Cycle Length field of the microinstruction. The contents of IOSTAT are described below.

BIT	DESCRIPTION
4:9	Shows the address selected by the CPU device address selection switches. IOSTAT(4:8) = ADDRESS(16:20) IOSTAT(9) = ADDRESS PARITY
10	LOGADDR < ULIM or LOGADDR < CPA. Output from the comparator which compares LOGADDR and ULIM/CPA. Undefined during memory and I/O operations.
11	LOGADDR < LLIM. Output from the comparator which compares LOGADDR and LLIM. Undefined during memory and I/O operations.
12:15	PREFETCH STATUS. Valid after termination of an instruction fetch. IOSTAT(12) = 0 IOSTAT(13) = PARITY ERROR IOSTAT(14) = TIME OUT IOSTAT(15) = NACK
20:23	I/O STATUS. Valid after termination of a memory or I/O operation. IOSTAT(20) = LIMIT VIOLATION IOSTAT(21) = PARITY ERROR IOSTAT(22) = TIME OUT IOSTAT(23) = NACK

2.2.1.11 CAM Control Register

2.2.1.11

The CAM Control Register (CAMCNTR) is a 1-bit destination register, which generates the control signal, BYPASSCAM.
BYPASSCAM: = CPUBUS(23)

When BYPASSCAM = 1 the system will act as if the cache memory is not installed.

2.2.2 Interrupt Control Unit

2.2.2

The Interrupt Control Unit (ICU) receives interrupt requests via the System Bus and sets the corresponding bit in the Interrupt Register (IR). A priority encoder generates an Interrupt Level (ILEV) corresponding to the bit in IR with the highest priority. Lowest bit numbers have highest priority. The contents of ILEV are compared with the contents of one of the two Interrupt Limit registers (INTRLIM). If $ILEV \leq INTRLIM$ the ICU interrupts the CPU.

From the microprogram is it possible to load the Interrupt Limit Registers, read the Interrupt Level Register and clear bits in IR.

2.2.2.1 Interrupt Limit Registers

2.2.2.1

The Interrupt Disable Limit Register (IDLIM) and the Interrupt Enable Limit Register (IELIM) are two 8-bit registers addressable as one 16-bit destination register from the microprogram.

$IDLIM(0:7) := CPUBUS(4:11)$

$IELIM(0:7) := CPUBUS(16:23)$

The DISABLE bit in the CPU Status Register selects IDLIM or IELIM as the current interrupt limit (ILIM).

if $DISABLE = 1$ then $ILIM = IDLIM$

else $ILIM = IELIM$

2.2.2.2 Interrupt Level Register

2.2.2.2

The Interrupt Level Register (INTRLEV) is an 8-bit register addressable as both source and destination register. When INTRLEV is addressed as source the contents are transferred to the CPUBUS.

$CPUBUS(0:15)$ is undefined

$CPUBUS(16:23) = INTRLEV(16:23)$

Addressing INTRLEV as destination will cause it to be loaded with ILEV. Because this operation must be synchronized with the internal ICU clock, it is necessary to insert a delay between the microinstruction which loads INTRLEV and the microinstruction which reads the contents of INTRLEV. This delay must be ≥ 700 ns.

2.2.2.3 Interrupt Register

2.2.2.3

The Interrupt Register (IR) is a 58-bit register in which interrupt requests are stored.

- IR(6): POWER LOW interrupt
- IR(7): INTERVAL TIMER interrupt
- IR(8:63): DEVICE CONTROLLER interrupts.

Level 0 to 5 are not available as hardware interrupts. IR is only addressable as a destination register. Execution of a microinstruction with IR as destination will reset the interrupt request bit selected by CPUBUS(18:23). The delay from the execution of a microinstruction resetting an IR bit to the CPU interrupt is affected is max:

900 ns + execution time for one microinstr.

Microinstructions used to reset IR bits must be separated by at least 700 ns.

2.2.3 Real Time Clock

2.2.3

The Real Time Clock consists of a 16-bit counter which is incremented by 1 every 0.1 ms. and counts modulo 65536. The value of the counter may be read under microprogram control. In addition the counter is used to generate an interrupt every 25.6 ms. (Interval Timer Interrupt).

2.2.3.1 Real Time Clock Register

2.2.3.1

The Real Time Clock Register (RTC) is a 16-bit register addressable as both source and destination register. The current value

of the RTC counter is transferred to RTC, when it is addressed as destination (LOAD command). The contents of RCT are valid max. 200 ns. after the LOAD command. The contents of RTC are transferred to the CPUBUS when addressed as source.

CPUBUS(0:23) = 8 ext 0 con RTC(8:23)

2.2.4 Technician Console Interface

2.2.4

The Technician Console (TC) is a V.24 compatible terminal connected to the CPU through an asynchronous serial I/O port controlled by an UART (Universal Asynchronous Receiver/Transmitter). Three registers are used for communication between the UART and the microprogram.

2.2.4.1 Technician Console Status Register

2.2.4.1

The Technician Console Status Register (TCSTAT) is a 3-bit source register used to control communication with TC. The contents of TCSTAT are transferred to CPUBUS when addressed as source.

CPUBUS(0:20) is undefined

CPUBUS(21:23) = TCSTAT(21:23)

TCSTAT has the following status bits:

<u>BIT</u>	<u>DESCRIPTION</u>
21	-, DATASET RDY. A 0 indicates that TC is connected and is ready to use.
22	DATA AVAIL. A 1 indicates that a character has been received by the UART and is available in the TCDAIN register. Addressing TCDAIN as destination resets DATA AVAIL.
23	TCOUT FULL. A 0 indicates that the UART transmitter is ready to accept a character. TCOUT FULL changes to 1 when TCDAOUT is loaded.

The state of TCSTAT(22) and TCSTAT(23) is delayed one microcycle relative to microinstructions controlling their state.

2.2.4.2 Technician Console Data In Register

2.2.4.2

The Technician Console Data In Register (TCDAIN) is an 8-bit source register containing characters received by the UART. The contents of TCDAIN are transferred to CPUBUS when addressed as source.

CPUBUS(0:15) is undefined

CPUBUS(16:23) = TCDAIN(16:23)

2.2.4.3 Technician Console Data Out Register

2.2.4.3

The Technician Console Data Out Register (TCDAOUT) is an 8-bit destination register. Characters loaded into TCDAOUT are outputted to the Technician Console.

TCDAOUT(0:7): = CPUBUS(16:23)

2.3 Microprogram Control Unit

2.3

The microprogram control unit, which is located on the CPUS21 printed circuit board, is built around the 2911A microprogram sequencer as shown on fig. 4.

The Control Store (CS) is addressed via a 12-bit tri-state bus, the Control Store Address Bus. The contents of the addressed CS location are loaded into the Microinstruction Register (MIR), which holds the microinstruction during its execution. The next microinstruction to be executed is fetched from CS during the execution of the current microinstruction in order to minimize microinstruction cycle time. The Control Store Address (CSADDR) may be selected from a number of sources as described in the following subsections.

2.3.1 Microinstruction Address Register

2.3.1

The Microinstruction Address Register (MAR) is part of the 2911A sequencer and is used for sequential microprogram addressing. In each microcycle MAR is loaded with CSADDR+1.

2.3.2 Subroutine Return Stack

2.3.2

The Subroutine Return Stack (STACK) is a 4 word x 12 bit register file operating as a push-pop stack, i.e. last in/first out (LIFO) structure. The STACK is used for saving of subroutine return addresses and for microprogram loop control. Associated with the STACK is a stack pointer (SP), pointing at the word on the top of the STACK.

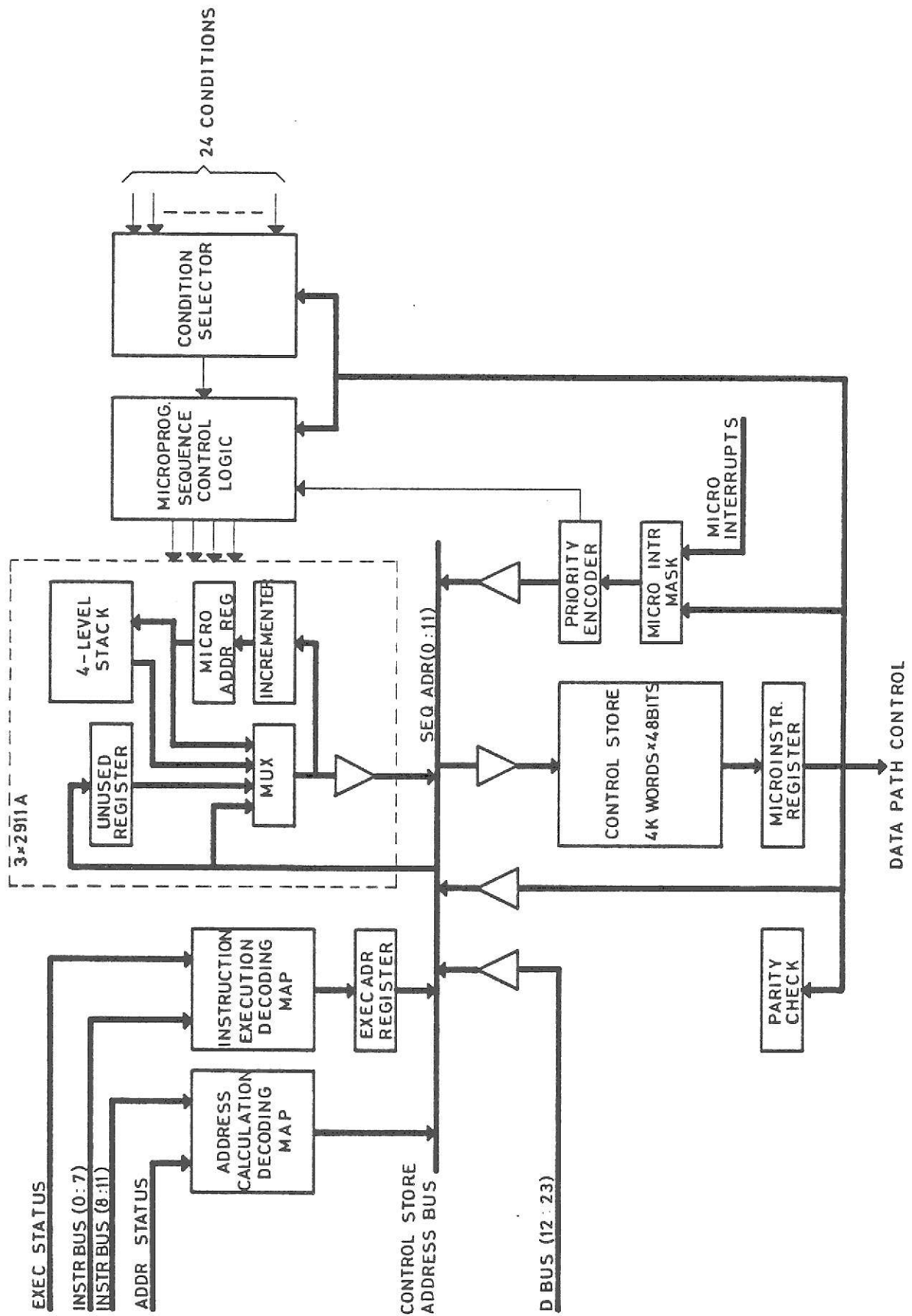


Figure 4: Microprogram control unit.

2.3.3 Microprogram Jump Address

2.3.3

The CSADDR for microprogram jumps may be selected from two sources: the Microinstruction Register, MIR(36:47) and the Destination Bus, DBUS(12:23). The last possibility makes it possible to use the General Registers and the Q-register as address sources. This may e.g. be useful for table look-up.

2.3.4 Microprogram Interrupt Address

2.3.4

In case of microprogram interrupt the CSADDR is generated by an 8-input priority encoder. Depending on the priority of the interrupt condition, the Interrupt Address (TEST) will address one of the first eight CS locations.

2.3.5 Address Calculation Decoding Map

2.3.5

RC8000 instructions are executed in two steps. The first step is calculation of the effective address (EA). Microprogram start addresses for the EA calculation routines are generated by the Address Calculation Decoding Map (ADDRMAP), which is a 512 words x 12 bits PROM. The entry address for ADDRMAP is generated as described below.

<u>ENTRY ADDR BIT</u>	<u>SOURCE</u>
0	PREFETCH ERROR
1	0, not used
2	ESCAPE MODE
3	AFTER ESCAPE
4	AFTER AM
5	INSTRBUS(8)
6	(9)
7	(10)
8	(11)

2.3.6 Instruction Execution Decoding Map

2.3.6

Second step in the execution of a RC8000 instruction is execution of the instruction itself. Microprogram start addresses for the instruction execution routines are generated by the Instruction Execution Decoding Map (EXECMAP), which is a 1 K words x 12 bit PROM. The entry address for EXECMAP is generated as described below.

<u>ENTRY ADDR BIT</u>	<u>SOURCE</u>
0	ESCAPE MODE
1	FPU AVAILABLE
2	INSTRBUS(6)
3	(7)
4	INSTRBUS(0)
5	(1)
6	(2)
7	(3)
8	(4)
9	(5)

The microprogram address generated by EXECMAP is saved in a register, EXECADDR, because execution of the current instruction is overlapped with the decoding of the next instruction. EXECADDR is used as CSADDR source.

2.4 Microinstructions

2.4

The CPU820 microinstruction repertoire comprises 8 different formats as shown in fig. 5. The microinstructions are 48 bits wide and consist of a number of fields. Microinstruction fields common to two or more formats are described in the following subsections. Fields referring to a single format are described in connection with that format.

2.4.1 P Field = MIR(0)

2.4.1

The P field is the parity bit for the microinstruction. Odd parity is used. In case of parity error the processor stops immediately and the indicator 'CS PARITY ERROR' will be lit. MIR contains the faulty microinstruction, which is not executed. CSADDR depends on the NEXT field of the faulty microinstruction. In order to proceed after control store parity error it is necessary to turn power off and then on again.

2.4.2 NEXT Field = MIR(1:4)

2.4.2

The NEXT field defines CSADDR for the next microinstruction to be executed. The following abbreviations are used to describe the function of the NEXT field.

CSADDR:	control store address.
MAR:	microinstruction address register.
STACK:	subroutine return stack.
SP:	stack pointer for STACK.
ADDRMAP:	address calculation decoding map.
EXECMAP:	instruction execution decoding map.
EXECADDR:	buffer register for EXECMAP.
TESTCOND:	condition selected by 'COND SEL' field.
PREFEN:	control signal generated by instruction decoding logic.
PREFIR:	prefetch instruction register.
PREFIC:	prefetch instruction counter.
DISP:	displacement register.
RELADDR:	relative address register.
MEM(A):	contents of main memory location A.

FORMAT 0: Read/Load Scratchpad

STATE OF TEXAS																						
0 1 4 5 6 7 9 10				19 20				24 25				29 30				35 36				41 42		47
P	NEXT		CL	F		ALU				A		B		UNIT FUNC/ TEST		SCRATCHP. ADDR.		INT/EXT REG DEST				
			0,0,0	D	OP	FUNC	C															

FORMAT 1: Read/Load Register

0 1		4 5 6 7		9 10		19 20				24 25		29 30		35 36		41 42		47	
P	NEXT	CL	F 0, 0, 1		ALU				A		B		UNIT FUNC/ TEST		INT REG SOURCE		INT/EXT REG DEST		
					D	OP	FUNC	C											

FORMAT 2: Read Constant

0 1		4 5		6 7		9 10		19 20		24 25		29 30		35 36		41 42		47	
P	NEXT	CL	F 0,1,0	ALU				A	B	UNIT FUNC/ TEST	CONSTANT ADDR	INT/EXT REG DEST							
				D	OP	FUNC	C												

FORMAT 3: Read External Register

0	1	4	5	6	7	9	10	19	20	24	25	29	30	35	36	41	42	47
P NEXT		CL	F	ALU				A		B		UNIT FUNC/ TEST		EXT REG SOURCE		INT/EXT REG DEST		
			0, 1, 1	D	OP	FUNC	C											

FORMAT 4: Half-Word Manipulator

0	1	4	5	6	7	9	10	19	20	24	25	29	30	35	36	41	42	47
P	NEXT	CL	F	ALU				A		B		NOT USED		MANIPULATOR FUNCTION		NOT USED		
			1 0 0	D	OP	FUNC	C											

FORMAT 5: Load Immediate

0 1		4 5 6 7		9 10		19 20												43 44		47	
P	NEXT	CL	F	ALU				24-BIT IMMEDIATE OPERAND										NOT USED			
			1, 0, 1	D	OP	FUNC	C														

FORMAT 6: Conditional Jump

0	1	4	5	6	7	9	10	19	20	24	25	27	28	29	30	31	35	36	47
P	NEXT	CL	F	ALU				A	NOT USED	S	X	T	COND SELECT		JUMP ADDRESS				
			1,1,0	D	OP	FUNC	C												

FORMAT 7: Shift, Multiply, Divide

0 1		4 5 6 7		9 10		19 20		24 25		29 30 31		35 36 37 38		40 41 42 43 44		47	
P	NEXT	CL	F 1 1 1	ALU				A	B	T	COND SELECT	N S	SI	M,D	TST	NOT USED	
				D	OP	FUNC	C										

Figure 5: Microinstruction formats.

<u>NEXT</u>	<u>FUNCTION</u>
0000	POP and CONTINUE. CSADDR: = MAR; MAR: = MAR+1; SP: = SP-1
0001	PUSH and CONTINUE (loop set-up). CSADDR: = MAR; SP: = SP+1; STACK(SP): = MAR, MAR: = MAR+1
0011	CONTINUE. CSADDR: = MAR; MAR: = MAR+1;
1000	SUBROUTINE RETURN. CSADDR: = STACK(SP); MAR: = STACK(SP)+1; SP: = SP-1;
1011	CONDITIONAL LOOP RETURN. SELCOND = 0: CSADDR: = STACK(SP); MAR: = STACK(SP)+1; SELCOND = 1: CSADDR: = MAR; SP: = SP-1; MAR: = MAR+1;
1110	NEXT INSTRUCTION. CSADDR: = ADDRMAP; MAR: = ADDRMAP+1; EXECADDR: = EXECMAP; DISP: = 12 ext PREFIR(12) con PREFIR(12:23); RELADDR: = 0 con IC(1:22) con 0 + 12 ext PREFIR(12) con PREFIR(12:23); PREFIC: = PREFIC + 2; if PREFEN = 1 then PREFIR: = MEM(PREFIC);
1111	EXECUTE. CSADDR: = EXECADDR; MAR: = EXECADDR+1; IC: = IC+2;

2.4.3 CL Field = MIR(5,6)

2.4.3

CL controls the microinstruction cycle length.

<u>CL</u>	<u>CYCLE LENGTH</u>
00	150 ns.
01	200 ns.
10	250 ns.
11	300 ns.

Most microinstructions may be executed in 150 ns. Certain combinations of format, ALU function, source and destination addresses require a longer cycle length.

2.4.4 F Field = MIR(7:9)

2.4.4

The F field defines the microinstruction format and thereby the usage of bit(20:47) of the microinstruction.

2.4.5 ALU-D Field = MIR(10,11)

2.4.5

The ALU-D field selects destination for the ALU output (F).

<u>ALU-D</u>	<u>FUNCTION</u>
00	Q: = F; DBUS: = F;
01	DBUS: = F;
10	GRB: = F; DBUS: = GRA;
11	GRB: = F; DBUS: = F;

2.4.6 ALU-OP Field = MIR(12:14)

2.4.6

The ALU-OP field selects the two operands, R and S, for the ALU.

<u>ALU-OP</u>	<u>OPERAND R</u>	<u>Operand S</u>
000	GRA	Q
001	GRA	GRB
010	ZERO	Q
011	ZERO	GRB
100	ZERO	GRA
101	SBUS	GRA
110	SBUS	Q
111	SBUS	ZERO

2.4.7 ALU-FUNC Field = MIR(15:17)

2.4.7

The ALU can perform three arithmetic and five logic functions. The ALU-FUNC field selects one of these functions. Cin is the carry input to the least significant position of the ALU controlled by the ALU-C field.

<u>ALU-FUNC</u>	<u>ALU FUNCTION</u>
000	R+S+Cin
001	-R+S-1+Cin
010	R-S-1+Cin
011	R S
100	R&S
101	-, R&S
110	R exor S
111	-, (R exor S)

2.4.8 ALU-C Field = MIR(18,19)

2.4.8

The ALU-C field controls the carry input, Cin, to the ALU.

<u>ALU-C</u>	<u>Cin</u>
00	0
01	1
10	CARRY
11	ADDCOND

2.4.9 A Field = MIR(20:24)

2.4.9

The A field addresses the A port of the general registers, GRA. MIR(20) is used to select direct or indirect addressing of GRA.

<u>MIR(20)</u>	<u>MIR(23,24)</u>	<u>GRA ADDRESS</u>
0	00	MIR(21,22) con WFIELD(0,1)
0	01	MIR(21,22) con WPRE(0,1)
0	10	MIR(21,22) con XFIELD(0,1)
0	11	MIR(21,22) con EA(21,22)
1	-	MIR(21:24)

2.4.10 B Field = MIR(25:29)

2.4.10

The B field addresses the B port of the general registers, GRB. The B field is also used as destination address for the general registers. MIR(25) is used to select direct or indirect addressing of GRB.

<u>MIR(25)</u>	<u>MIR(28,29)</u>	<u>GRB ADDRESS</u>
0	00	MIR(26,27) con WFIELD(0,1)
0	01	MIR(26,27) con WPRE(0,1)
0	10	MIR(26,27) con XFIELD(0,1)
0	11	MIR(26,27) con EA(21,22)
1	-	MIR(26:29)

2.4.11 UNIT FUNC/TEST Field = MIR(30:35)

2.4.11

This field has two functions controlled by MIR(30).

MIR(30) = 0: MIR(31:35) may be used as a UNIT FUNCTION to control units connected to the CPBUS. The interpretation of the field depends on the unit concerned.

UNIT FUNCTION(0:5) = 0 con MIR(31:35).

MIR(30) = 1: MIR(31:35) is used as mask bits for microprogram interrupt conditions. MASK BIT = 1 enables the interrupt condition. When one or more of the enabled conditions are 1, one of the first eight CS locations, will be selected as the next CS address independent of the NEXT field. The address is generated by a priority encoder and the condition corresponding to CS location 0 has the highest priority.

<u>CS LOCATION</u>	<u>INTR. CONDITION</u>	<u>MASK BIT</u>
0	POWER UP	UNMASKABLE
1	CPU INTR	MIR(31)
2	LIMIT VIOLATION	MIR(35)
3	I/O ERROR	MIR(32)
4	OVFL & INT. MASK	MIR(33)
5	SHIFT OVFL & INT MASK	MIR(34)
6	EXT INTR	MIR(31)
7	UNUSED	

CPU INTR indicates interrupt from one or more of the following sources, which may be tested individually by conditional jumps.

SINGLE INSTRUCTION INTR.

TC INPUT

AUX. INTR. (unused)

CAM ERROR

OCF AUTOLOAD

REMOTE AUTOLOAD

EXT INTR indicates interrupt from a device connected to the SYSTEM BUS. EXT INTR is generated by the Interrupt Control Unit on the CPU822.

2.4.12 INT/EXT REG DEST Field = MIR(42:47)

2.4.12

This field is used to address destination registers located on the CPU821 and on modules connected to the CPUBUS.

INT/EXT

<u>REG DEST</u>	<u>CPU821 DESTINATION REGISTERS</u>
00	No load.
01	Logical Address Status (LASTAT).
02	Scratchpad (SCRATCHP). Format(0) only.
03	Instruction Counter (IC).
04	Loop Counter (LC).
05	CPU Status (CPUST).
06	Control Output (CONTROLOUT).
07	Not used.

INT/EXT

REG DEST CPU822 DESTINATION REGISTERS

10	Not used.
11	Logical Address (LOGADDR) & LASTAT.
12	Data Out (DATAOUT).
13	Base (BASE).
14	Lower Limit (LLIM).
15	Upper Limit (ULIM).
16	CPA Limit (CPA).
17	Real Time Clock (RTC), load command.
20	Clear Intr. Reg. (IR) bit.
21	Interrupt Limit (IDLIM & IELIM).
22	Interrupt Level (INTRLEV), load command.
23	Techn. Console Data Out (TCDAOUT).
24	CAM Bypass Control.
25	Clear TC interrupt & TCSTAT(22).

CAMB01/803 DESTINATION REGISTERS

40	CAM Test Data
41	CAM Control

FPUB01 DESTINATION REGISTERS

70	FPU FRACTION(0:23)
71	FPU FRACTION(24:35) con EXP(0:11)

2.4.13 T Field = MIR(30)

2.4.13

The T field is used to specify whether the true or the complemented value of the condition selected by the COND SELECT field is assigned SELCOND.

T=0: SELCOND = complemented value of condition.

T=1: SELCOND = true value of condition.

2.4.14 COND SELECT Field = MIR(31:35)

2.4.14

The COND SELECT field selects 1 of the 24 conditions listed below.

COND

<u>SELECT</u>	<u>SELECTED CONDITION</u>	
00	ZERO	
01	NEG:	DBUS(0)
02	NZERO:	F $\diamond$ 0
03	OVFL:	arithm. overflow
04	CARRY:	carry from ALU bit(0)
05	SIGN:	F(0) exor OVR 1)
06	NORMO:	DBUS(0) $\diamond$ DBUS(1)
07	NORM1:	DBUS(1) $\diamond$ DBUS(2)
10	DIVOVFL:	SHLINK $\diamond$ CARRY 2)
11	LC(0)	
12	MAXLOOP:	-,LC(0) !-,LCMAX(0)
13	-,WADDR	
14	EA(0)	
15	Not used	
16	MONMODE	
17	ESCMODE	
20	CPUINTR	
21	EXTINTR:	device interrupt
22	-,TC	INPUT READY
23	-,AUXINTR:	unused
24	-,CAMERROR	
25	-,OCP	AUTOLOAD
26	-,REMOTE	AUTOLOAD
27	POWER	LOW

Notes:

1) OVR is the overflow signal from the 2901.

2) SHLINK is the bit shifted out by a shift microinstruction.

The conditions: NEG, NZERO, OVFL, CARRY, SIGN, SHLINK, NORMO, NORM1 and DIVOVFL are updated by all microinstruction formats with the exception of format 6, conditional jump. These conditions are delayed one microcycle compared to the microinstructions which generate them.

2.4.15 Format 0 : Read/Load Scratchpad

2.4.15

Format 0 operates on the Scratchpad (SCRATCHP). The SCRATCHP ADDR field = MIR(38:41) addresses one of the 16 locations. The complemented contents of the addressed location are transferred to the SBUS.

SBUS = $\neg$ SCRATCHP(SCRATCHP ADDR)

The addressed SCRATCHP location is loaded with the contents of DBUS when the INT/EXT REG DEST field, MIR(42:47), specifies SCRATCHP as destination.

SCRATCHP(SCRATCHP ADDR): = DBUS

The SCRATCHP cannot be used as both source and destination within the same microcycle.

2.4.16 Format 1 : Read/Load Register

2.4.16

Source registers located on the CPU821 (internal source registers) may be accessed by means of format 1 microinstructions. The contents of the register addressed by the INT REG source field, MIR(36:41), are transferred to the SBUS.

INT REG

<u>SOURCE</u>	<u>SOURCE REGISTER/FUNCTION</u>
00	Displacement (DISP)
10	Instruction Counter (IC)
20	DISP + IC
30	Loop Counter (LC)

40	DBUS(0:21) con OVFL con CARRY
41	DBUS(0:21) con SHIFTOVFL con 0
42	DBUS(0:21) con 0 con 0
43	DBUS(0:21) con 0 con 0
50	Switch Register (SWITCH)

2.4.17 Format 2 : Read Constant

2.4.17

Constants in the Constant Memory (CONST) are addressed by the CONSTANT ADDR field, MIR(36:41). The addressed constant is transferred to the SBUS.

SBUS = CONST(CONSTANT ADDR)

CONSTANT	
<u>ADDR</u>	<u>CONSTANT, octal</u>
00	0000 7000
01	0000 7760
02	0000 0046
03	0000 7777
04	0000 0027
05	0000 0026
06	0000 0025
07	0000 0017
10	0000 0016
11	0000 0014
12	0000 0013
13	0000 0012
14	0000 0003
15	0000 0004
16	0000 0020
17	0000 0040

CONSTANT

<u>ADDR</u>	<u>CONSTANT, octal</u>
20	0000 0100
21	0000 0200
22	0000 1000
23	0000 4000
24	0020 0000
25	0040 0000
26	0100 0000
27	0200 0000
30	0000 0006
31	0400 0000
32	1000 0000
33	2000 0000
34	0000 0007
35	4000 0000
36	0000 0010
37	0000 0001

2.4.18 Format 3 : Read External Register

2.4.18

When this format is used the SBUS receives data from the CPUBUS. The format may therefore be used to access source registers connected to the CPUBUS. The register address is specified by the EXT REG SOURCE field, MIR(36:41).

EXT REG

<u>SOURCE</u>	<u>CPU822 SOURCE REGISTERS</u>
00	Data In (DATAIN)
01	I/O Status (IOSTAT)
02	Real Time Clock (RTC)
03	Interrupt Level (INTRLEV)
04	Techn. Console Data In (TCDAIN)
05	Techn. Console Status (TCSTAT)

EXT REG

SOURCE CAMB01/803 - CPU822 SOURCE REGISTERS

40 Data In (DATAIN) 1)
 41 I/O Status (IOSTAT) 1)

FPU801 SOURCE REGISTERS

70 FRACTION(0:23)
 71 FRACTION(24:35) con EXP(0:11)
 72 EXCEPTION(22,23)

- 1) The DATAIN and IOSTAT registers on the CAM respond automatically to the common addresses when the CAM is installed, otherwise the registers on the CPU822 respond to the addresses.

2.4.19 Format 4 : Half-Word Manipulator

2.4.19

When this format is used the SBUS receives data from the Half-Word Manipulator, HWM. The operation of the HWM is controlled by the MANIPULATOR FUNCTION field, MIR(36:41), and the ODD bit in the IASTAT register.

MANIP.

<u>FUNCTION</u>	<u>ODD</u>	<u>SBUS(0:11)</u>	<u>SBUS(12:23)</u>
00	0	12 ext 0	DBUS(0:11)
00	1	12 ext 0	DBUS(12:23)
01	0	12 ext DBUS(0)	DBUS(0:11)
01	1	12 ext DBUS(12)	DBUS(12:23)
02	0	DBUS(12:23)	12 ext 0
02	1	12 ext 0	DBUS(12:23)
03	0	12 ext 0	12 ext 1
03	1	12 ext 1	12 ext 0
04	x	12 ext DBUS(12)	DBUS(12:23)
05	x	DBUS(12:23)	DBUS(0:11)
06	x	12 ext 0	DBUS(0:11)
07	x	DBUS(12:23)	12 ext 0

When the HWM is used, the ALU-D field must be controlled in the following way.

ALU-D = 10: DBUS: = GRA; GRB: = F;

2.4.20 Format 5 : Load Immediate

2.4.20

The 24-bit IMMEDIATE OPERAND field, MIR(20:43), is transferred to the SBUS. The Q-register is the only register, which can be used in format 5 microinstructions.

2.4.21 Format 6 : Conditional Jump

2.4.21

This microinstruction format executes conditional jumps and conditional subroutine calls. The S field, MIR(28), is used to specify subroutine call. The JUMP ADDRESS field, MIR(36:47), or DBUS(12:23) may be selected as address source. The X field, MIR(29), selects address source. The condition (SELCOND) specified by the T and the COND SELECT fields determines the address of the next microinstruction to be executed.

SELCOND = 0: NEXT field selects CSADDR.

SELCOND = 1: CSADDR is controlled by the S and X fields as described below.

<u>S,X</u>	<u>FUNCTION</u>
00	JUMP CSADDR: = JUMP ADDR; MAR: = JUMP ADDR+1;
01	INDEXED JUMP. CSADDR: = DBUS(12:23); MAR: = DBUS(12:23)+1;
10	CALL. CSADDR: = JUMP ADDR; SP: = SP+1; STACK(SP): = MAR; MAR: = JUMP ADDR + 1
11	INDEXED CALL. CSADDR: = DBUS(12:23); SP: = SP+1; STACK(SP): = MAR; MAR: = DBUS(12:23)+1;

A format 6 microinstruction with NEXT = 1110 (NEXT INSTRUCTION) will start an instruction prefetch regardless of the current value of PREFEN (PREFETCH ENABLE). This feature may be used for the skip case of an RC8000 skip instruction.

2.4.22 Format 7 : Shift, Multiply and Divide

2.4.22

This format is especially intended for support of shift, multiply and divide instructions.

The NS field, MIR(36), is used in connection with the ALU-D field to control shift operations as specified below.

<u>NS</u>	<u>ALU-D</u>	<u>FUNCTION</u>
0	00	GRB con Q: = SHIN con F con Q(0:22);
0	01	GRB: = SHIN con F(0:22);
0	10	GRB con Q: = F(1:23) con Q con SHIN;
0	11	GRB: = F(1:23) con SHIN;
1	00	Q: = F; DBUS: = F;
1	01	DBUS: = F;
1	10	GRB: = F; DBUS: = GRA;
1	11	GRB: = F; DBUS: = F;

The SI field, MIR(37,38), controls the input (SHIN) to the vacated position in shifts.

<u>SI</u>	<u>SHIN, right shift</u>	<u>SHIN, left shift</u>
00	ZERO	ZERO
01	SHLINK	SHLINK
10	F(0)	ADDCOND
11	SIGNEXT	NOT USED

SIGNEXT = sign extension.

The M field, MIR(40), in connection with the ALU-OP field and the condition, ADDCOND, are especially intended to be used in multiplication routines.

<u>M</u>	<u>ALU-OP</u>	<u>ADDCOND</u>	<u>OPERAND R</u>	<u>OPERAND S</u>
0	000	X	GRA	Q
0	001	X	GRA	GRB
0	010	X	ZERO	Q
0	011	X	ZERO	GRB
0	100	X	ZERO	GRA
0	101	X	SBUS	GRA
0	110	X	SBUS	Q
0	111	X	SBUS	ZERO
1	0X0	0	GRA	Q
1	0X0	1	ZERO	Q
1	0X1	0	GRA	GRB
1	0X1	1	ZERO	GRB
1	1X0	0	ZERO	GRA
1	1X0	1	SBUS	Q
1	1X1	0	SBUS	GRA
1	1X1	1	SBUS	ZERO

The D field, MIR(41), in connection with the ALU-FUNC field and ADDCOND are especially intended for use in division routines.

<u>D</u>	<u>ALU-FUNC</u>	<u>ADDCOND</u>	<u>ALU FUNCTION</u>
0	000	X	R+S+Cin
0	001	X	-R+S-1+Cin
0	010	X	R-S-1+Cin
0	011	X	R!S
0	100	X	R&S
0	101	X	-, R&S
0	110	X	R exor S
0	111	X	-, (R exor S)
1	00X	0	R+S+Cin
1	00X	1	-R+S-1+Cin
1	01X	0	R-S-1+Cin
1	01X	1	R!S
1	10X	0	R&S
1	10X	1	-, R&S
1	11X	0	R exor S
1	11X	1	-, (R exor S)

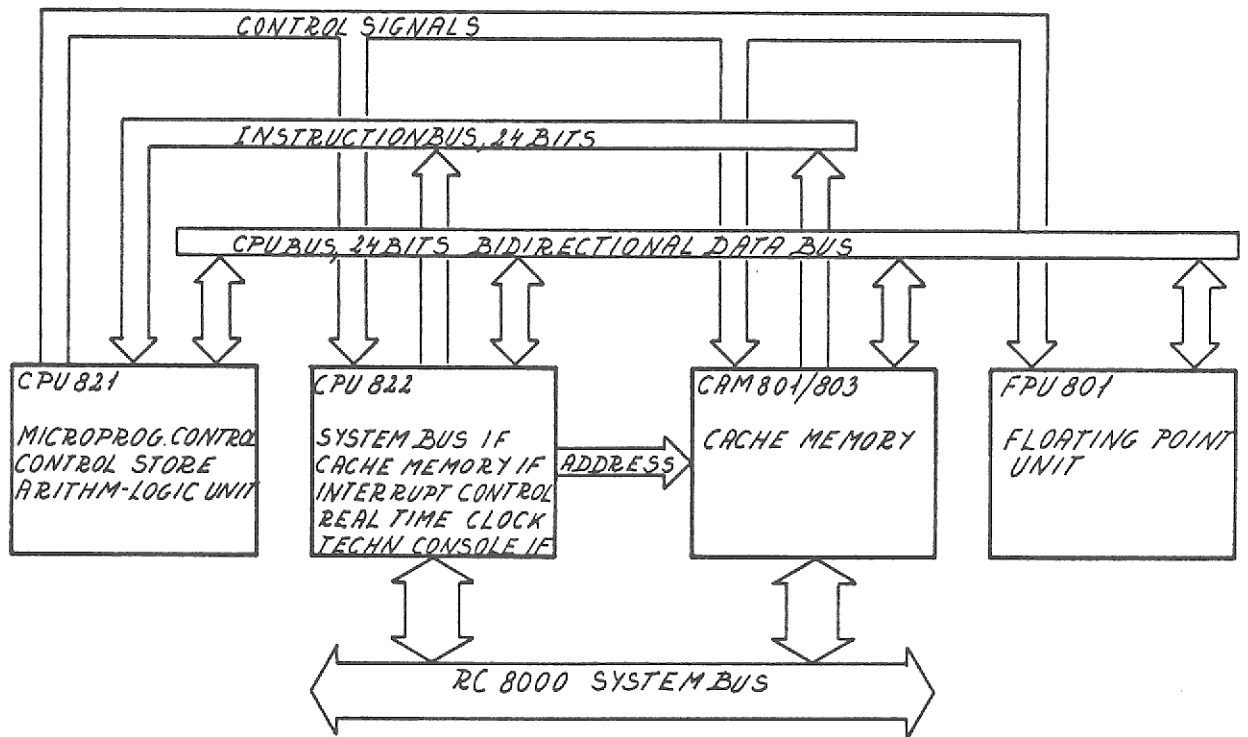
The TST field, MIR(42,43), controls the conditions, ADDCOND and DIVCOND. These conditions are only affected by format 7 microinstructions.

<u>TST</u>	<u>FUNCTION</u>
00	DIVSIGN: = F(0)
01	ADDCOND: = F(0) exor -,DIVSIGN
10	ADDCOND: = -,Q(23)
11	CONDITIONS UNCHANGED

The shift overflow condition, SHIFTOVFL, can only be set by execution of a format 7 microinstruction.

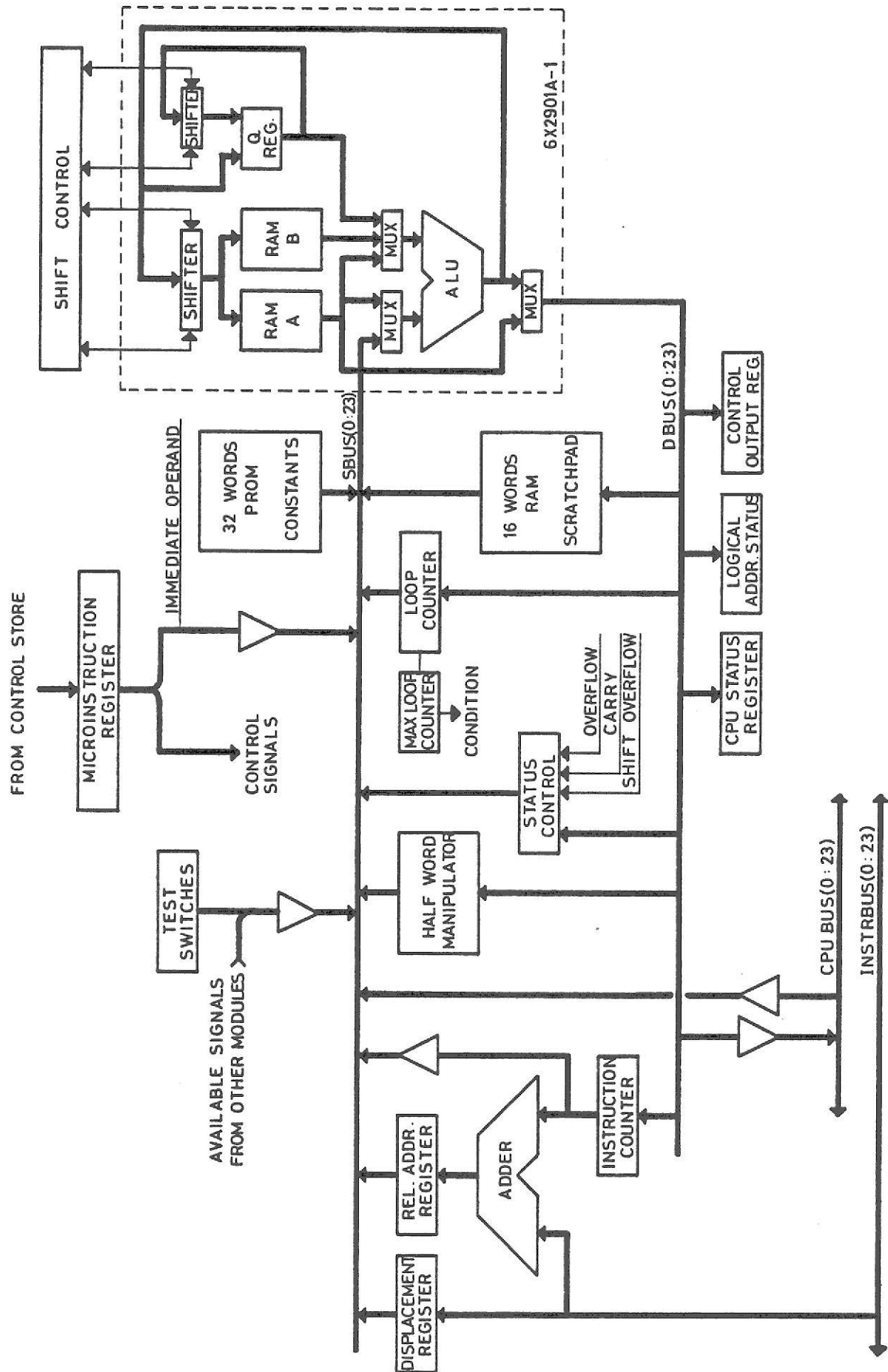
Set SHIFTOVFL = FORMAT 7 & (DBUS(0) exor DBUS(1))

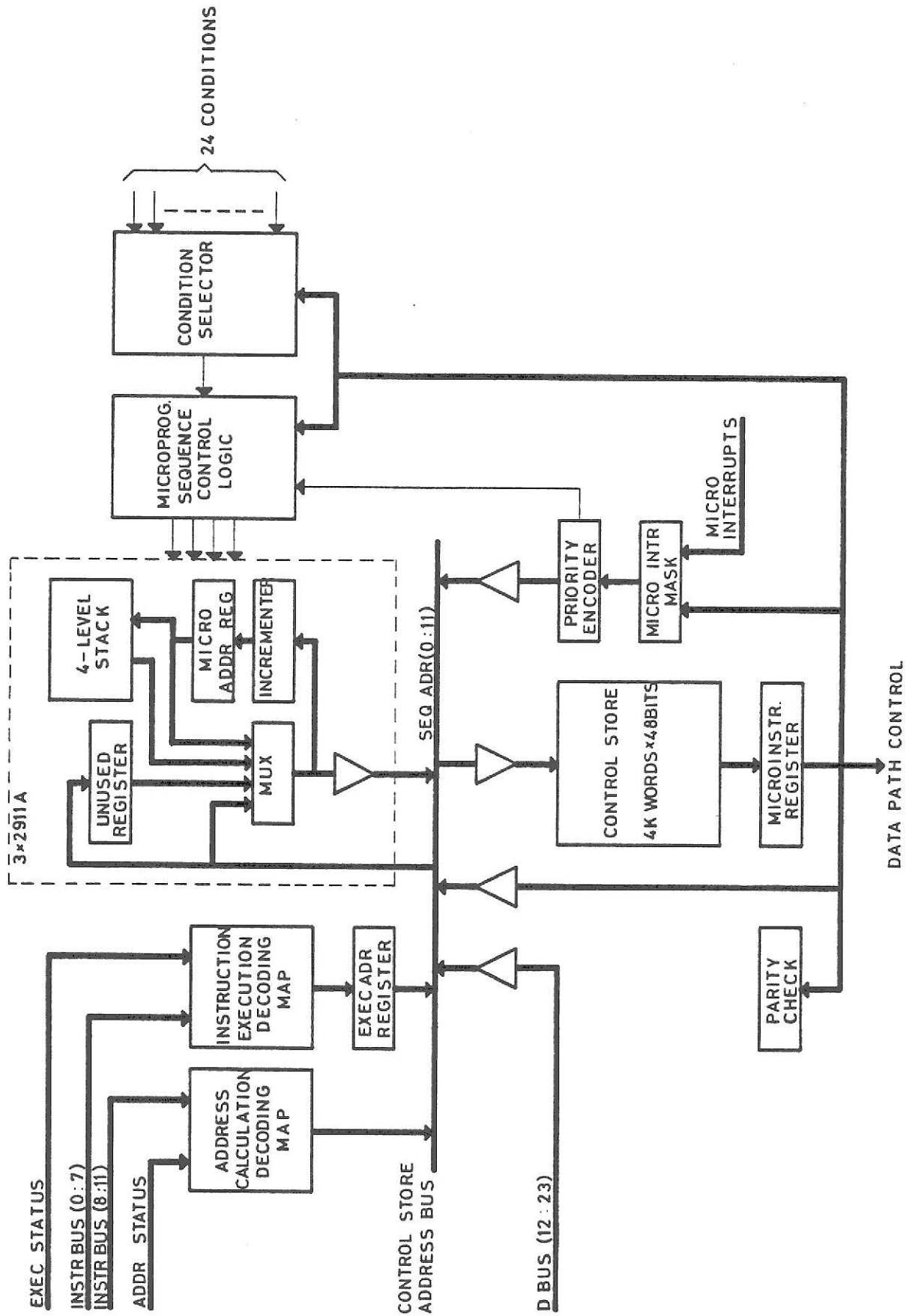
Once SHIFTOVFL has been set it remains set until execution of a format 0,1,2,3,4 or 5 microinstruction, which clears it.

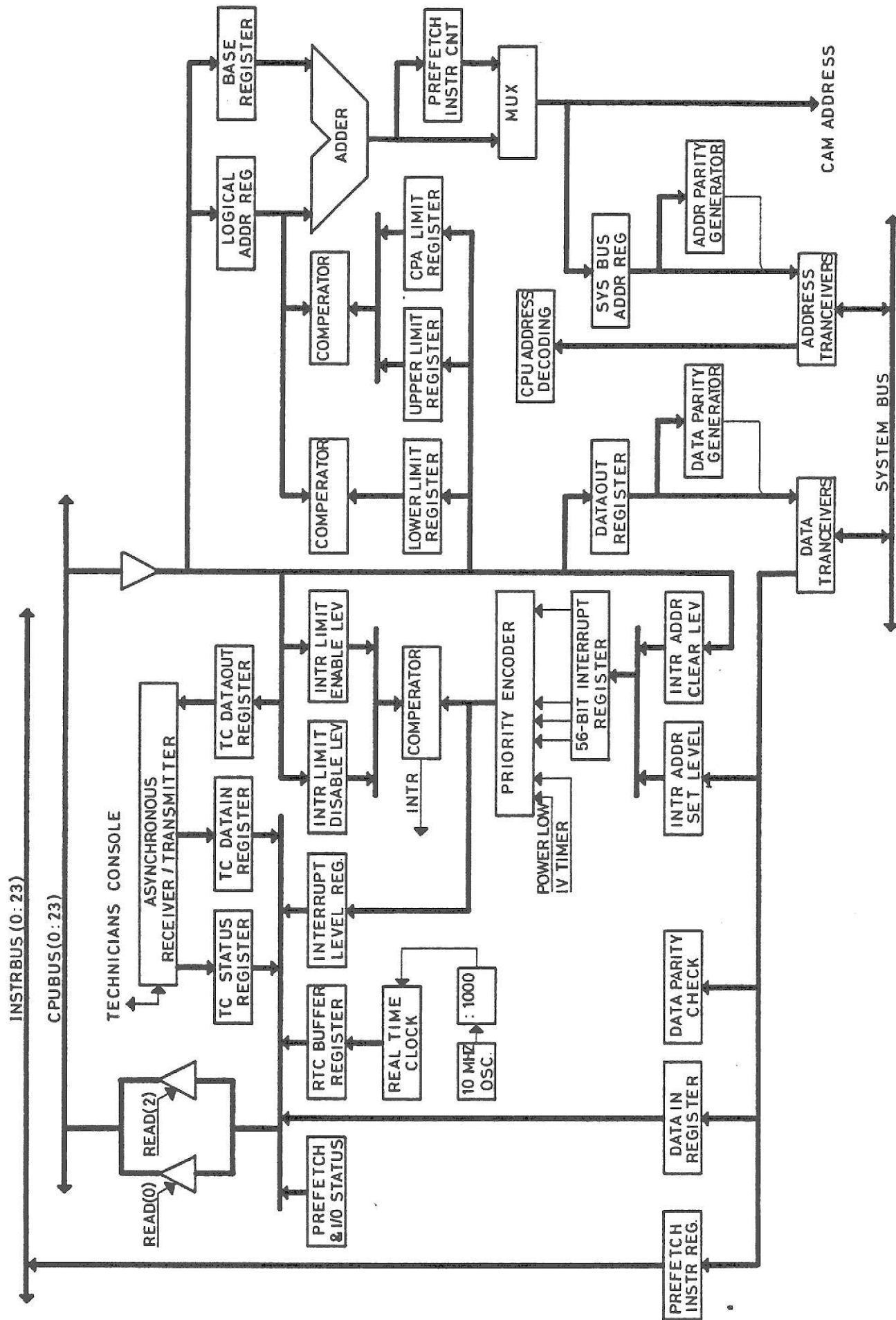


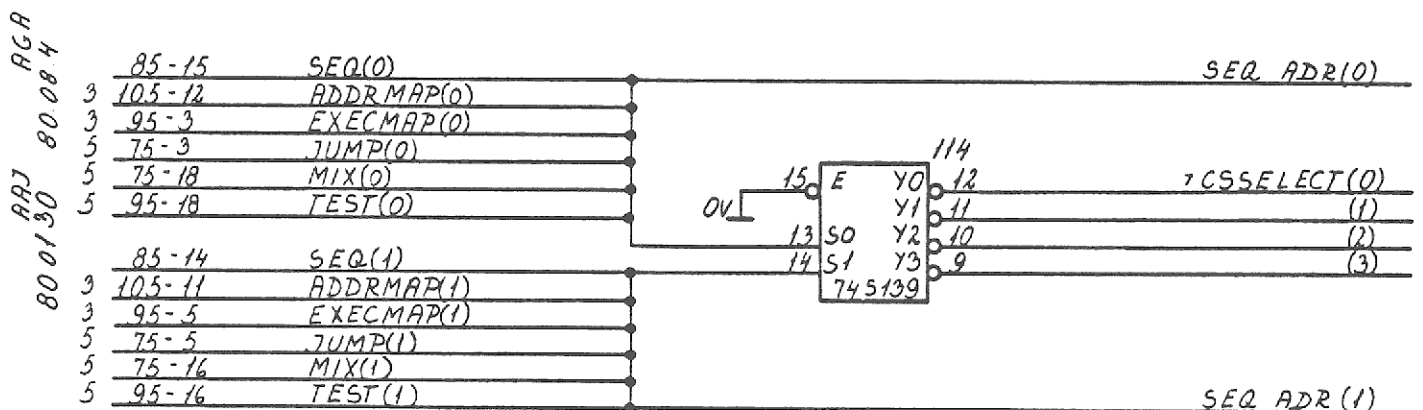
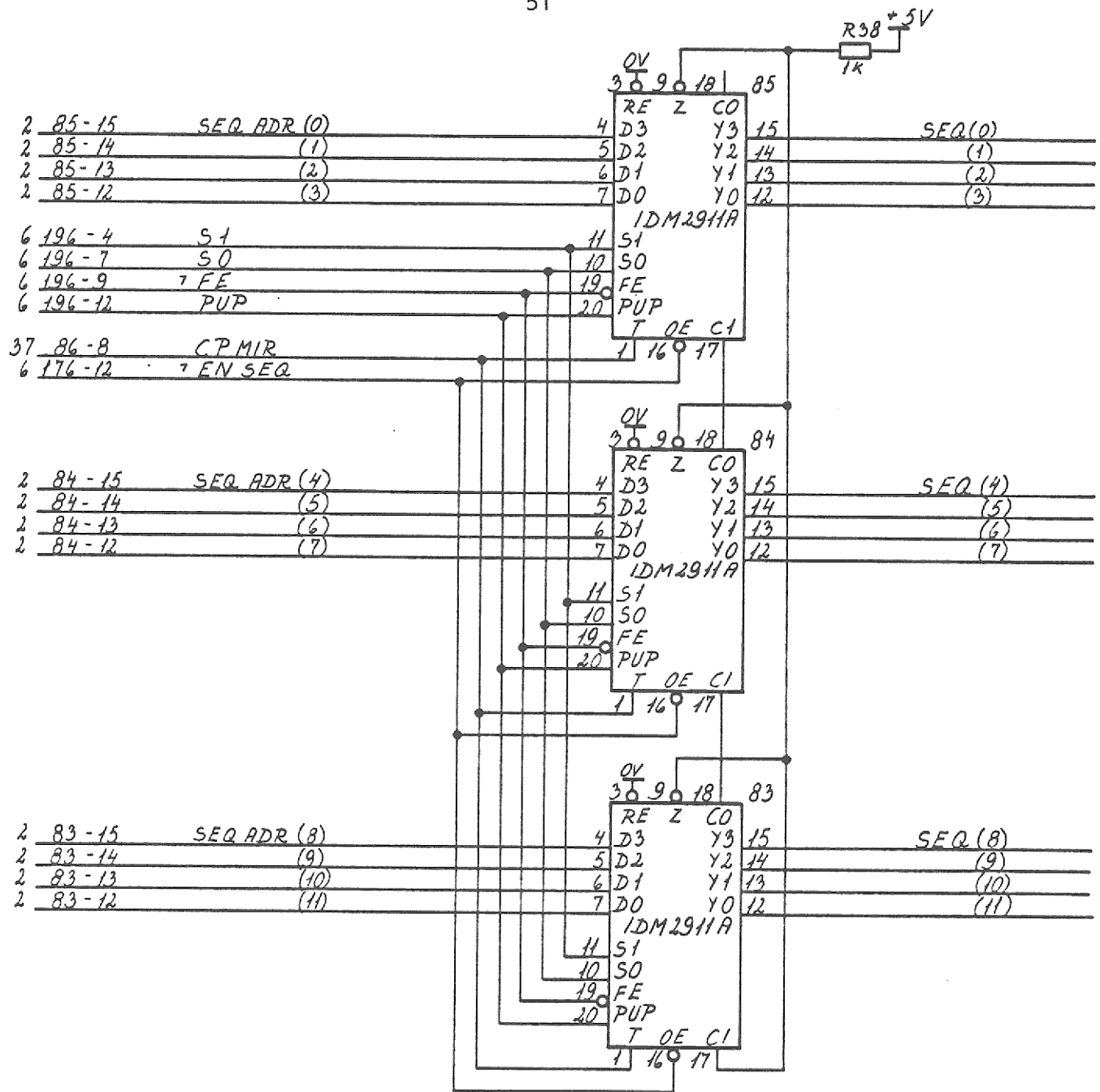
8106-11
AAJ
8106-15
AGA

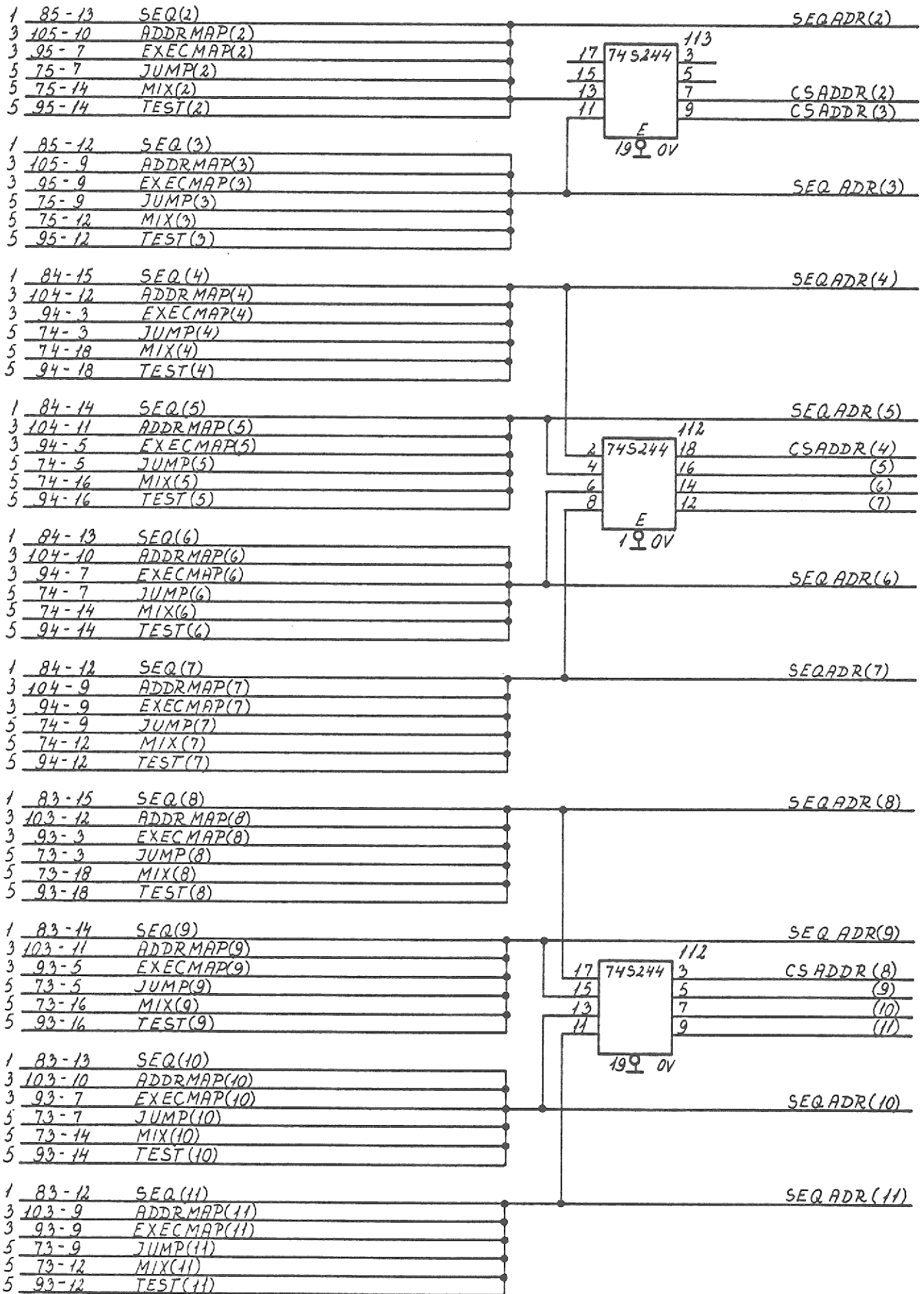
CPU 820 CPU 820 extended with CACHE MEMORY and FLOATING POINT UNIT

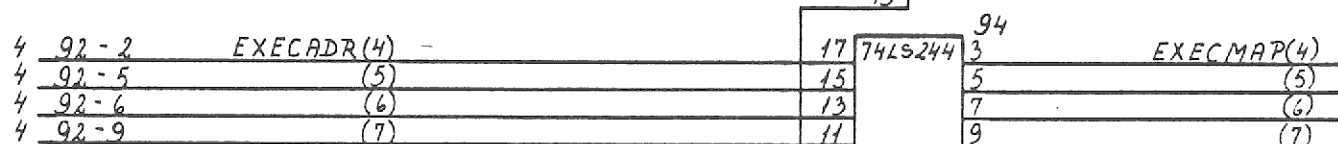
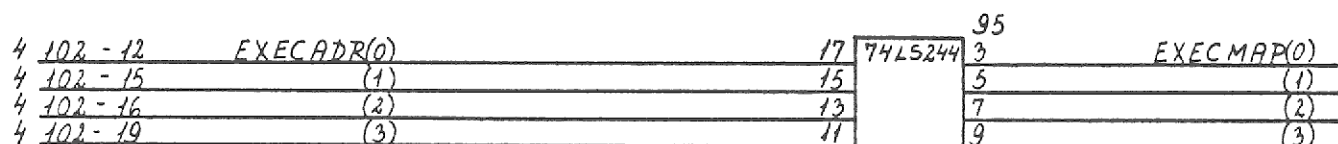
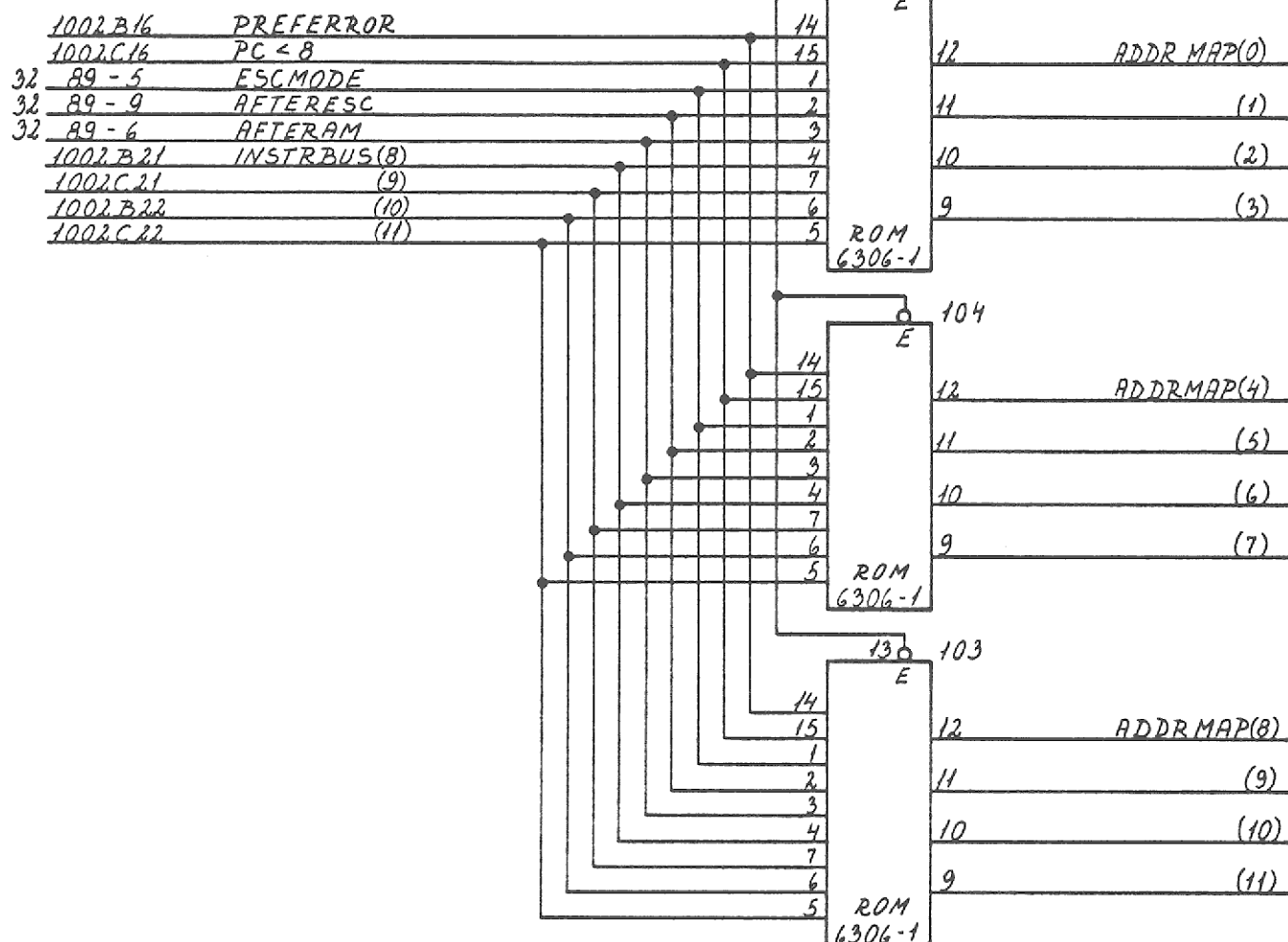




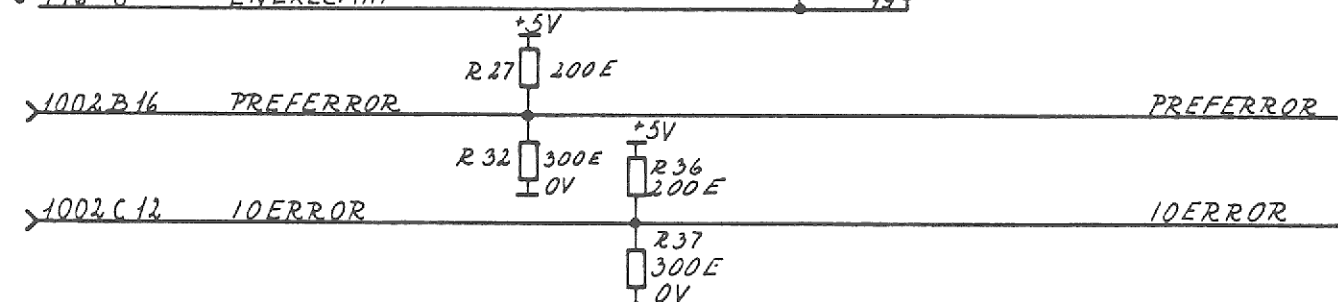


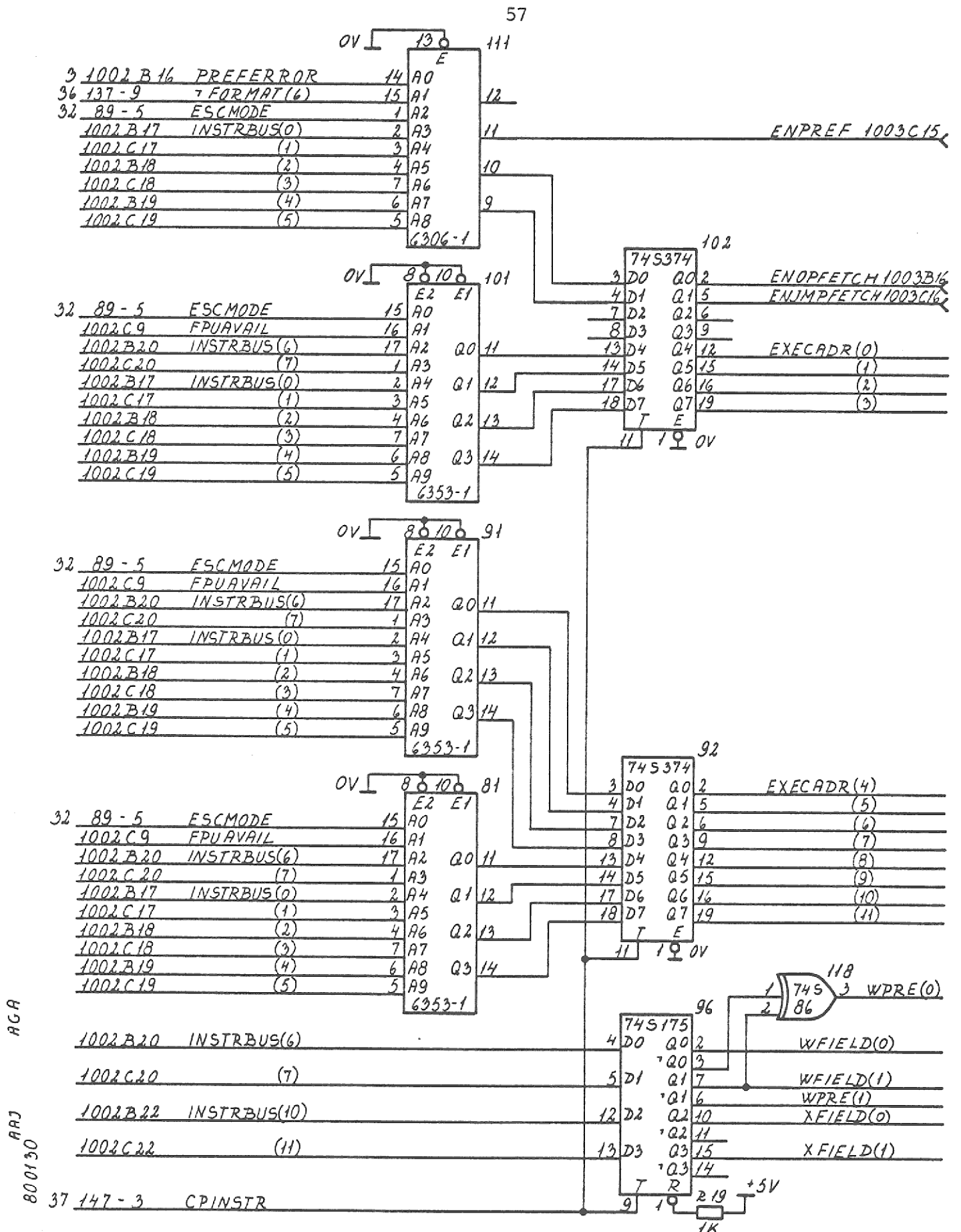


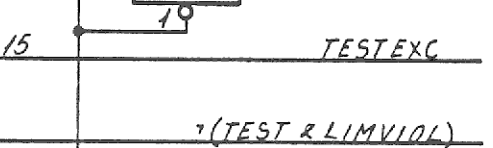
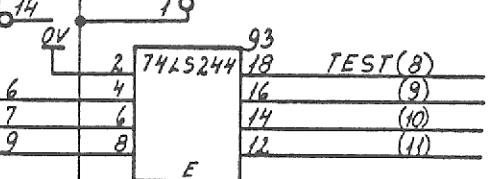
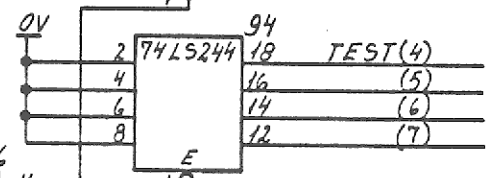




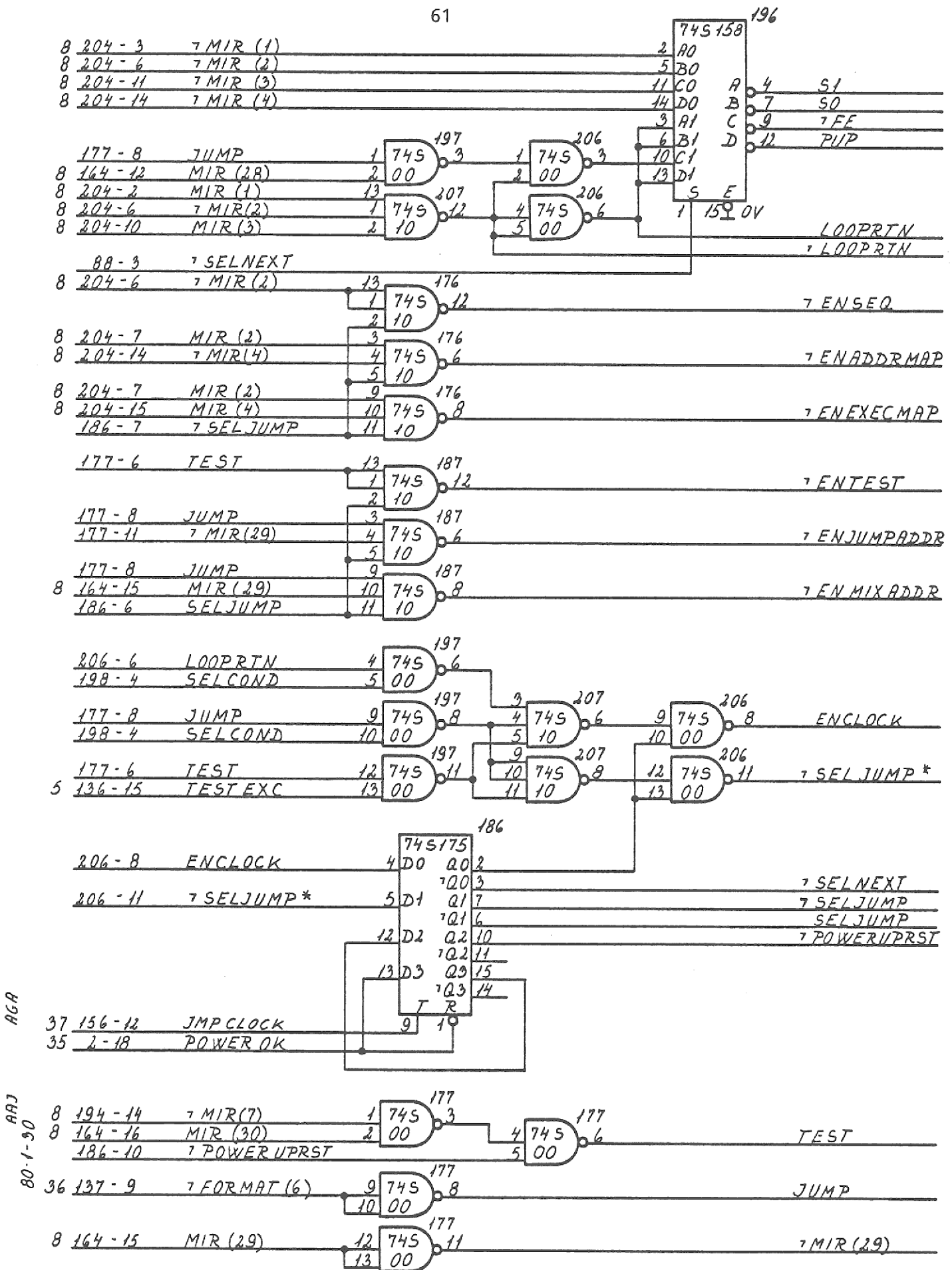
6 176 - 8 7 ENEXECMAP





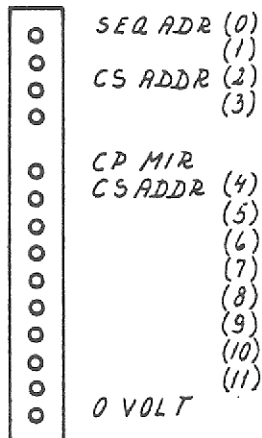


800130
AAJ



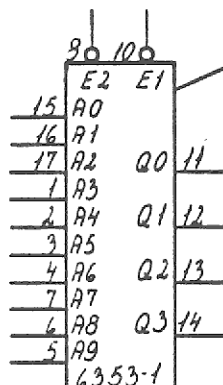
1 114-12 7 CS SELECT (0)
 1 114-11 (1)
 1 114-10 (2)
 1 114-9 (3)

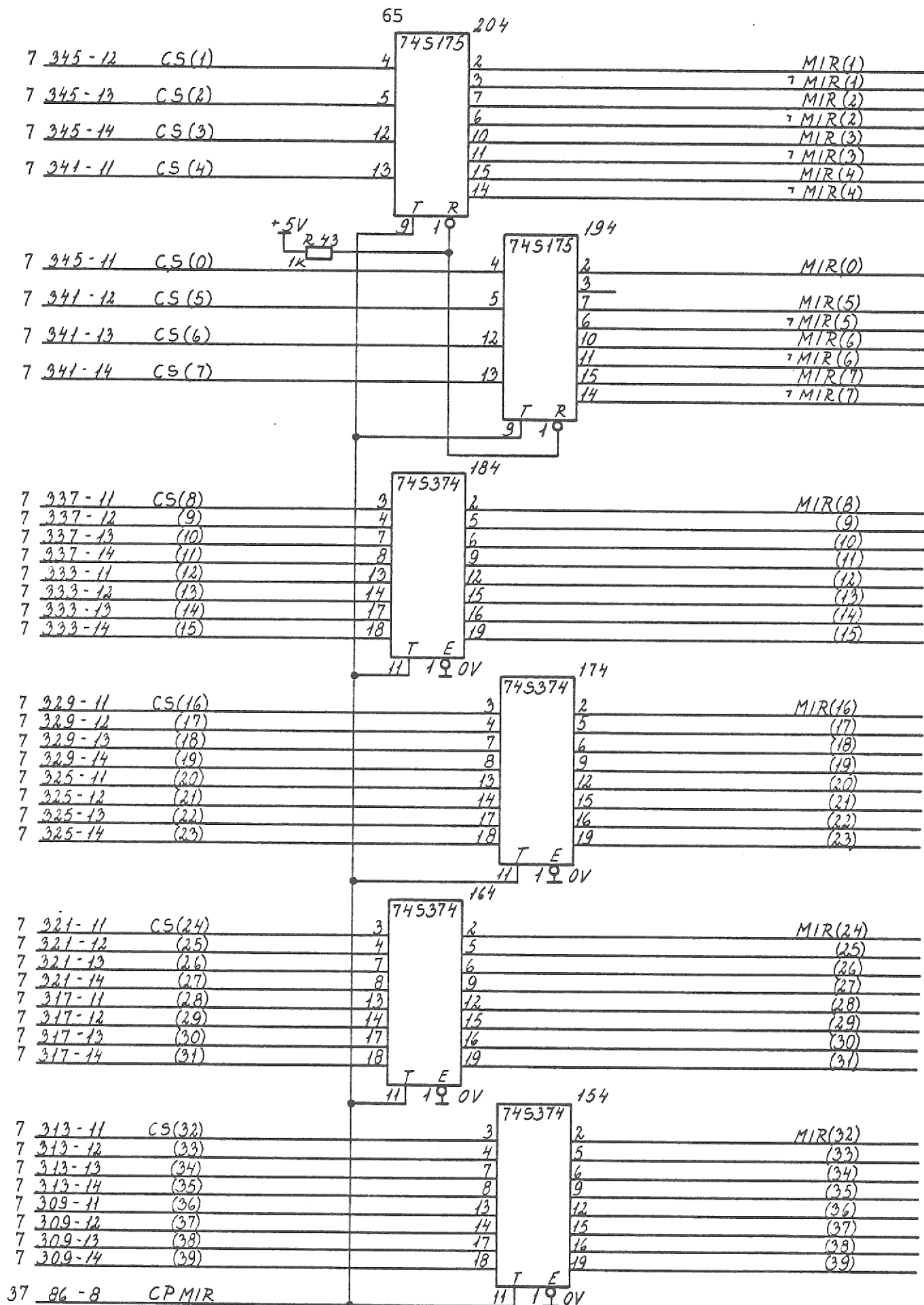
Test Points for
Control Store Address



2 113-7 CS ADDR (2) 15
 2 113-9 (3) 16
 2 112-18 (4) 17
 2 112-16 (5) 1
 2 112-14 (6) 2
 2 112-12 (7) 3
 2 112-3 (8) 4
 2 112-5 (9) 7
 2 112-7 (10) 6
 2 112-9 (11) 5

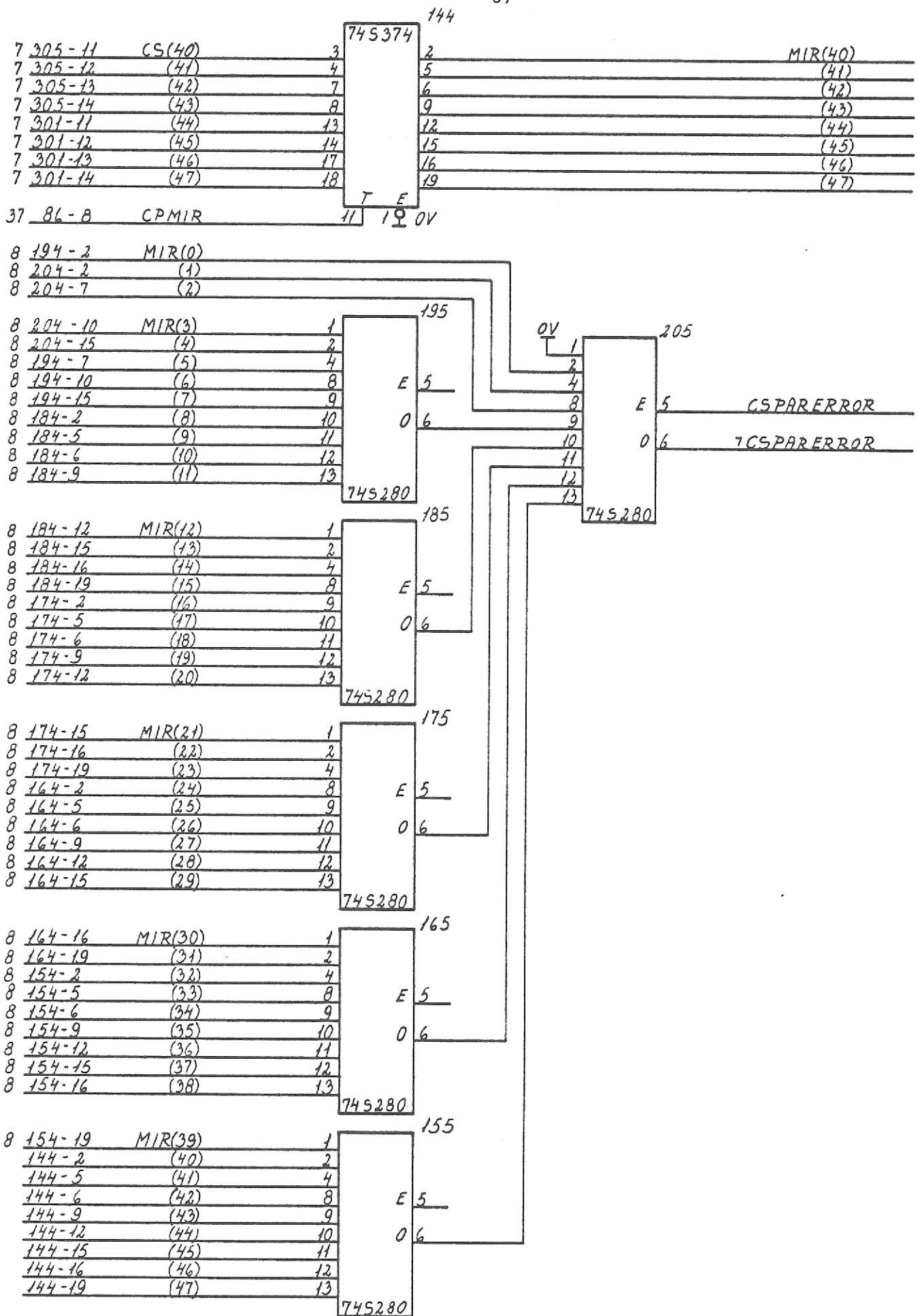
0V	0V	0V	0V	11	
8 10	8 10	8 10	8 10	12	CS (0)
348	347	346	345	13	(1)
				14	(2)
					(3)
344	343	342	341	11	CS (4)
				12	(5)
				13	(6)
				14	(7)
340	339	338	337	11	CS (8)
				12	(9)
				13	(10)
				14	(11)
336	335	334	333	11	CS (12)
				12	(13)
				13	(14)
				14	(15)
332	331	330	329	11	CS (16)
				12	(17)
				13	(18)
				14	(19)
328	327	326	325	11	CS (20)
				12	(21)
				13	(22)
				14	(23)
324	323	322	321	11	CS (24)
				12	(25)
				13	(26)
				14	(27)
320	319	318	317	11	CS (28)
				12	(29)
				13	(30)
				14	(31)
316	315	314	313	11	CS (32)
				12	(33)
				13	(34)
				14	(35)
312	311	310	309	11	CS (36)
				12	(37)
				13	(38)
				14	(39)
308	307	306	305	11	CS (40)
				12	(41)
				13	(42)
				14	(43)
304	303	302	301	11	CS (44)
				12	(45)
				13	(46)
				14	(47)





CPU 821
R 12929

MICROINSTRUCTION REGISTER



30 120-9 SHIFT 1/0 Q(0)
30 128-7 SHIFT 1/0 R(0)

			33	8	16	80
			CN*4 RS3 QS3			
29	150-1	18	6	18		
8	184-6	MIR(10)	7	17		
8	184-9	MIR(11)	5	16		
8	184-19	MIR(15)	27	15		
8	174-2	MIR(16)	28	14		
29	138-9	13	26	13		
8	184-12	MIR(12)	14	12		
29	138-7	11	13	11		
8	184-16	MIR(14)	12	10		
8	174-15	MIR(21)	4	A0		
8	174-16	(22)	3	A1		
31	149-6	AADDR(2)	2	A2		
31	156-6	(3)	1	A3		
8	164-6	MIR(26)	17	B0		
8	164-9	(27)	18	B1		
31	159-6	BADDR(2)	19	B2		
31	168-6	(3)	20	B3		
13	99-5	SBUS(0)	22	D3		
13	99-7	(1)	23	D2		
13	99-9	(2)	24	D1		
13	99-11	(3)	25	D0		
37	86-10	CPALU	15	T		
			IDM2901A-1			
			40	OE		
29	90-11	CARRY(4)	29	9	21	

F3 31 F(0)
F-0 11 F(0:3) = 0
G 32 G(0)
P 35 P(0)
OVR 34 OVR

Y3 39 DBUS(0)
Y2 38 (1)
Y1 37 (2)
Y0 36 (3)

			33	8	16	70
			CN*4 RS3 QS3			
29	150-1	18	6	18		
8	184-6	MIR(10)	7	17		
8	184-9	MIR(11)	5	16		
8	184-19	MIR(15)	27	15		
8	174-2	MIR(16)	28	14		
29	138-9	13	26	13		
8	184-12	MIR(12)	14	12		
29	138-7	11	13	11		
8	184-16	MIR(14)	12	10		
8	174-15	MIR(21)	4	A0		
8	174-16	(22)	3	A1		
31	149-6	AADDR(2)	2	A2		
31	158-6	(3)	1	A3		
8	164-6	MIR(26)	17	B0		
8	164-9	(27)	18	B1		
31	159-6	BADDR(2)	19	B2		
31	168-6	(3)	20	B3		
13	69-5	SBUS(4)	22	D3		
13	69-7	(5)	23	D2		
14	69-9	(6)	24	D1		
14	69-11	(7)	25	D0		
37	86-10	CPALU	15	T		
			IDM2901A-1			
			40	OE		
29	90-12	CARRY(8)	29	9	21	
11	103-8	SHIFT 1/0 R(8)				
11	60-16	SHIFT 1/0 Q(8)				

F3 31 F(4:7) = 0
F-0 11 F(4:7) = 0
G 32 G(1)
P 35 P(1)
OVR 34

Y3 39 DBUS(4)
Y2 38 (5)
Y1 37 (6)
Y0 36 (7)

SHIFT 1/0 R(8)
 SHIFT 1/0 Q(8)

			33	8	16	60
			CN*4	RS3	QS3	
29	150-1	18	6	18		
8	184-6	MIR(10)	7	17		
8	184-9	MIR(11)	5	16		
8	184-19	MIR(15)	27	15		
8	174-2	MIR(16)	28	14		
29	138-9	13	26	13		
8	184-12	MIR(12)	14	12		
29	138-7	11	13	11		
8	184-16	MIR(14)	12	10		
8	174-15	MIR(21)	4	A0		
8	174-16	(22)	3	A1		
31	149-6	AADDR(2)	2	A2		
31	158-6	(3)	1	A3		
8	164-6	MIR(26)	17	B0		
8	164-9	(27)	18	B1		
31	159-6	BADDR(2)	19	B2		
31	168-6	(3)	20	B3		
14	59-5	SBUS(8)	22	D3		
14	59-7	(9)	23	D2		
14	59-9	(10)	24	D1		
14	59-11	(11)	25	D0		
37	86-10	CPALU	15	T		
			IDM2901A-1			
			OE			
29	97-9	CARRY	29	9	21	

F3 31
 F=0 11
 G 32
 P 35
 OVR 34

F(8:11) = 0
 G(2)
 P(2)

Y3 39
 Y2 38
 Y1 37
 Y0 36

DBUS(8)
 (9)
 (10)
 (11)

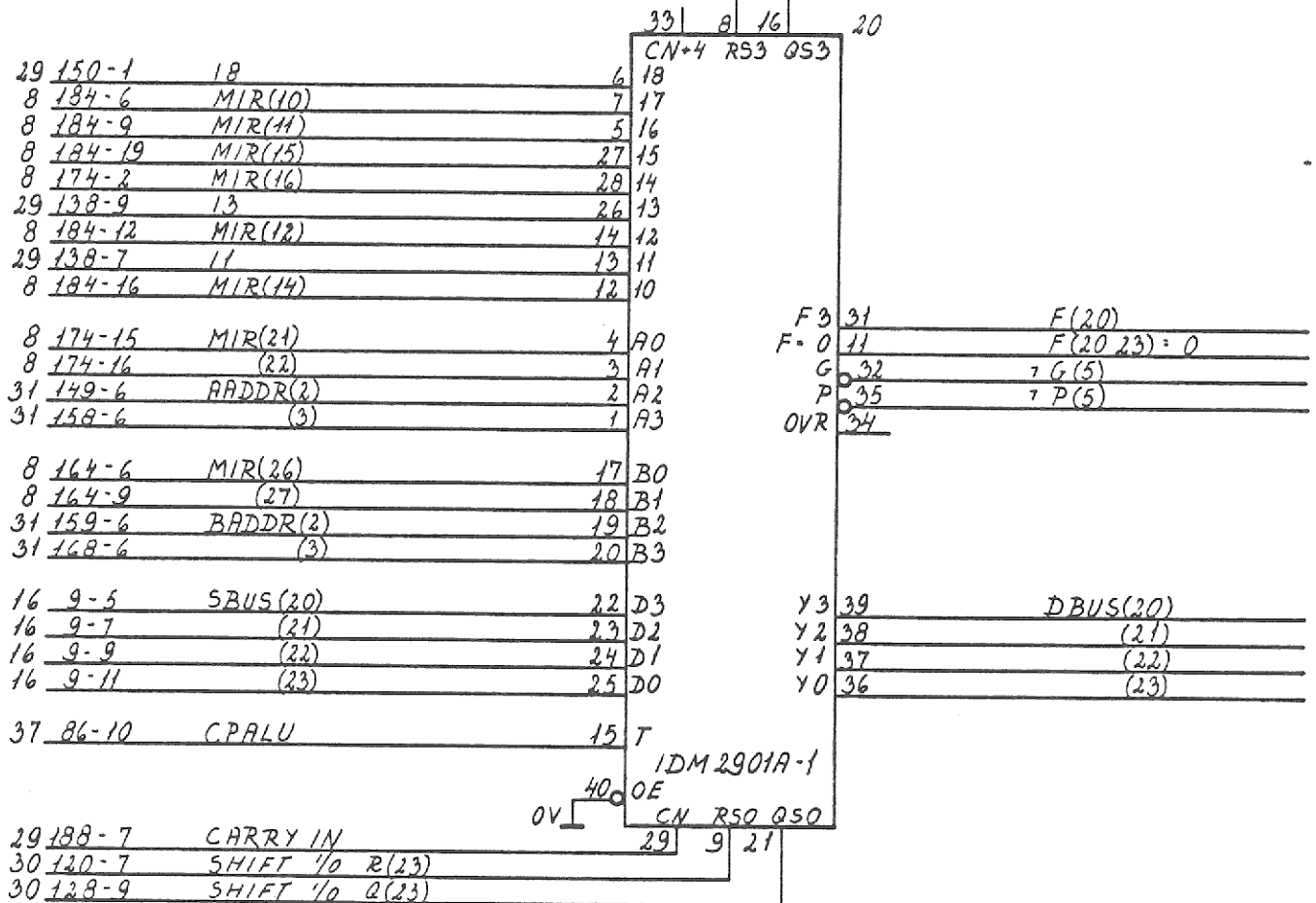
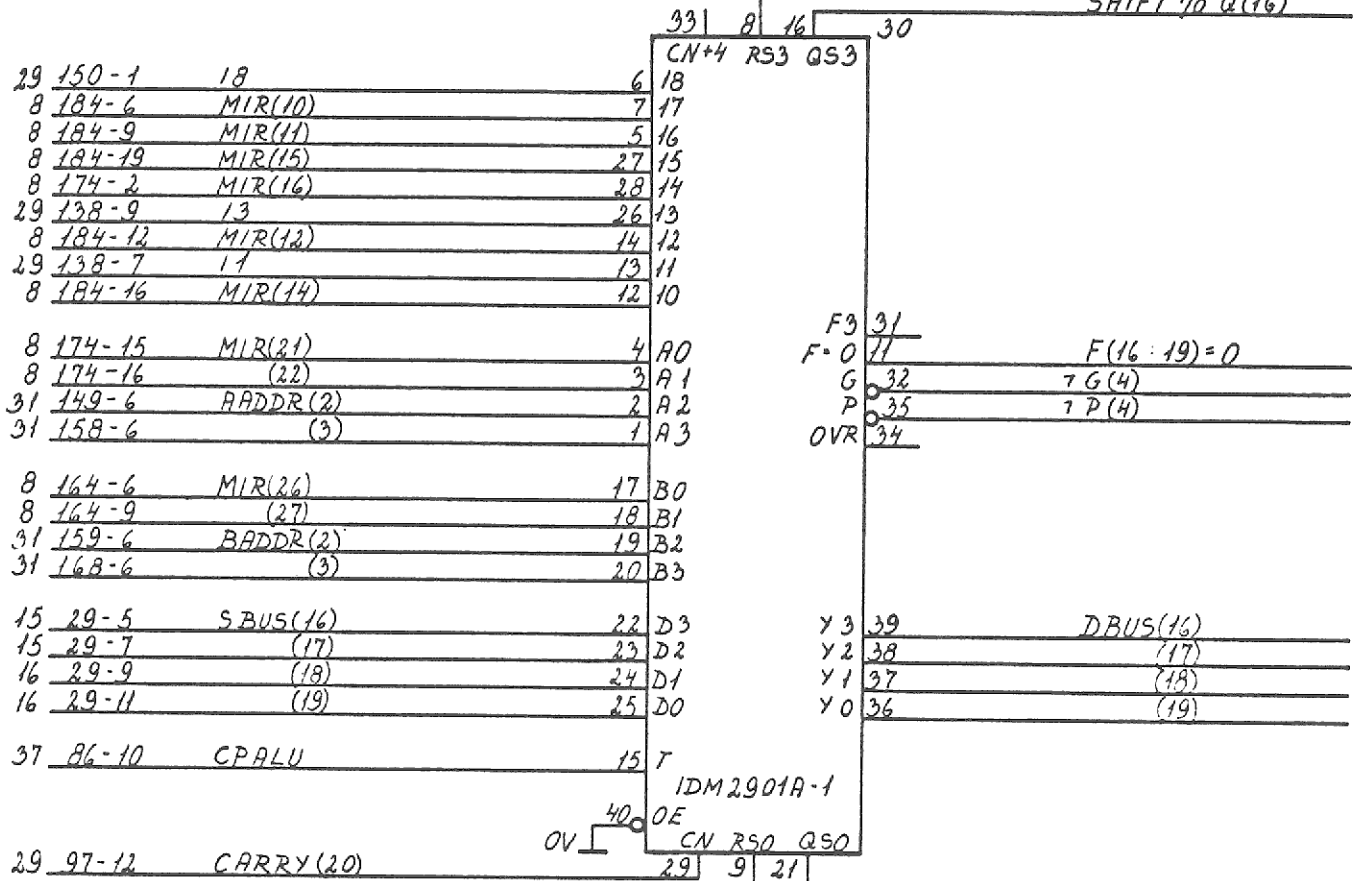
			33	8	16	40
			CN*4	RS3	QS3	
29	150-1	18	6	18		
8	184-6	MIR(10)	7	17		
8	184-9	MIR(11)	5	16		
8	184-19	MIR(15)	27	15		
8	174-2	MIR(16)	28	14		
29	138-9	13	26	13		
8	184-12	MIR(12)	14	12		
29	138-7	11	13	11		
8	184-16	MIR(14)	12	10		
8	174-15	MIR(21)	4	A0		
8	174-16	(22)	3	A1		
31	149-6	AADDR(2)	2	A2		
31	158-6	(3)	1	A3		
8	164-6	MIR(26)	17	B0		
8	164-9	(27)	18	B1		
31	159-6	BADDR(2)	19	B2		
31	168-6	(3)	20	B3		
15	39-5	SBUS(12)	22	D3		
15	39-7	(13)	23	D2		
15	39-9	(14)	24	D1		
15	39-11	(15)	25	D0		
37	86-10	CPALU	15	T		
			IDM2901A-1			
			OE			
29	97-11	CARRY(16)	29	9	21	
12	30-8	SHIFT 1/0 R(16)				
12	30-16	SHIFT 1/0 Q(16)				

F3 31
 F=0 11
 G 32
 P 35
 OVR 34

F(12:15) = 0
 G(3)
 P(3)

Y3 39
 Y2 38
 Y1 37
 Y0 36

DBUS(12)
 (13)
 (14)
 (15)

SHIFT 1/0 R(16)
 SHIFT 1/0 Q(16)


75

17	99-5	SCRATCHP(0)		SBUS(0)
18	78-1	CONST(0)		
28	71-3	CBUSIN(0)		
19	88-11	HWORD(0)		
21	72-2	DISP(0)		
22	44-18	ICS(0)		
23	43-2	RELADDR(0)		
25	47-18	LCS(0)		
27	79-18	EX(0)		
18	62-18	IMOP(0)		

17	99-7	SCRATCHP(1)		SBUS(1)
18	78-2	CONST(1)		
28	71-5	CBUSIN(1)		
19	88-12	HWORD(1)		
21	72-5	DISP(1)		
22	44-16	ICS(1)		
23	43-5	RELADDR(1)		
25	47-16	LCS(1)		
27	79-16	EX(1)		
18	62-16	IMOP(1)		

17	99-9	SCRATCHP(2)		SBUS(2)
18	78-3	CONST(2)		
28	71-7	CBUSIN(2)		
19	88-13	HWORD(2)		
21	72-6	DISP(2)		
22	44-14	ICS(2)		
23	43-6	RELADDR(2)		
25	47-14	LCS(2)		
27	79-14	EX(2)		
18	62-14	IMOP(2)		

17	99-11	SCRATCHP(3)		SBUS(3)
18	78-4	CONST(3)		
28	71-9	CBUSIN(3)		
19	88-14	HWORD(3)		
21	72-9	DISP(3)		
22	44-12	ICS(3)		
23	43-9	RELADDR(3)		
25	47-12	LCS(3)		
27	79-12	EX(3)		
18	62-12	IMOP(3)		

AGA

17	69-5	SCRATCHP(4)		SBUS(4)
18	78-5	CONST(4)		
28	61-3	CBUSIN(4)		
19	68-11	HWORD(4)		
21	72-12	DISP(4)		
22	44-3	ICS(4)		
23	43-12	RELADDR(4)		
25	47-3	LCS(4)		
27	79-3	EX(4)		
18	62-3	IMOP(4)		

RAJ
800130

17	69-7	SCRATCHP(5)		SBUS(5)
18	78-6	CONST(5)		
28	61-5	CBUSIN(5)		
19	68-12	HWORD(5)		
21	72-15	DISP(5)		
22	44-5	ICS(5)		
23	43-15	RELADDR(5)		
25	47-5	LCS(5)		
27	79-5	EX(5)		
18	62-5	IMOP(5)		

77

17	69-9	SCRATCHP(6)	
18	78-7	CONST(6)	SBUS(6)
28	61-7	CBUSIN(6)	
19	68-13	HWORD(6)	
21	72-16	DISP(6)	
22	44-7	ICS(6)	
23	43-16	RELADDR(6)	
25	47-7	LCS(6)	
27	79-7	EX(6)	
18	62-7	IMOP(6)	

17	69-11	SCRATCHP(7)	SBUS(7)
18	78-9	CONST(7)	
28	61-9	CBUSIN(7)	
19	68-14	HWORD(7)	
21	72-19	DISP(7)	
22	44-9	ICS(7)	
23	43-19	RELADDR(7)	
25	47-9	LCS(7)	
27	79-9	EX(7)	
18	62-9	IMOP(7)	

17	59-5	SCRATCHP(8)	SBUS(8)
18	48-1	CONST(8)	
28	51-3	CBUSIN(8)	
19	58-11	HWORD(8)	
21	52-2	DISP(8)	
22	24-18	ICS(8)	
23	23-2	RELADDR(8)	
25	27-18	LCS(8)	
27	49-18	EX(8)	
18	42-18	IMOP(8)	

17	59-7	SCRATCHP(9)	SBUS(9)
18	48-2	CONST(9)	
28	51-5	CBUSIN(9)	
19	58-12	HWORD(9)	
21	52-5	DISP(9)	
22	24-16	ICS(9)	
23	23-5	RELADDR(9)	
25	27-16	LCS(9)	
27	49-16	EX(9)	
18	42-16	IMOP(9)	

17	59-9	SCRATCHP(10)	SBUS(10)
18	48-3	CONST(10)	
28	51-7	CBUSIN(10)	
19	58-13	HWORD(10)	
21	52-6	DISP(10)	
22	24-14	ICS(10)	
23	23-6	RELADDR(10)	
25	27-14	LCS(10)	
27	49-14	EX(10)	
18	42-14	IMOP(10)	

17	59-11	SCRATCHP(11)	SBUS(11)
18	48-4	CONST(11)	
28	51-9	CBUSIN(11)	
19	58-14	HWORD(11)	
21	52-9	DISP(11)	
22	24-12	ICS(11)	
23	23-9	RELADDR(11)	
25	27-12	LCS(11)	
27	49-12	EX(11)	
18	42-12	IMOP(11)	

AGA

ARJ
800130

17	39-5	SCRATCHP(12)	79		
18	48-5	CONST(12)			SBUS(12)
28	41-3	CBUSIN(12)			
20	38-11	HWORD(12)			
21	52-12	DISP(12)			
22	24-3	ICS(12)			
23	23-12	RELADDR(12)			
25	27-3	LCS(12)			
27	49-3	EX(12)			
18	42-3	IMOP(12)			

17	39-7	SCRATCHP(13)			SBUS(13)
18	48-6	CONST(13)			
28	41-5	CBUSIN(13)			
20	38-12	HWORD(13)			
21	52-15	DISP(13)			
22	24-5	ICS(13)			
23	23-15	RELADDR(13)			
25	27-5	LCS(13)			
27	49-5	EX(13)			
18	42-5	IMOP(13)			

17	39-9	SCRATCHP(14)			SBUS(14)
18	48-7	CONST(14)			
28	41-7	CBUSIN(14)			
20	38-13	HWORD(14)			
21	52-16	DISP(14)			
22	24-7	ICS(14)			
23	23-16	RELADDR(14)			
25	27-7	LCS(14)			
27	49-7	EX(14)			
18	42-7	IMOP(14)			

17	39-11	SCRATCHP(15)			SBUS(15)
18	48-9	CONST(15)			
28	41-9	CBUSIN(15)			
20	38-14	HWORD(15)			
21	52-19	DISP(15)			
22	24-9	ICS(15)			
23	23-19	RELADDR(15)			
25	27-9	LCS(15)			
27	49-9	EX(15)			
18	42-9	IMOP(15)			

17	29-5	SCRATCHP(16)			SBUS(16)
18	18-1	CONST(16)			
28	31-3	CBUSIN(16)			
20	28-11	HWORD(16)			
21	32-2	DISP(16)			
22	4-18	ICS(16)			
24	3-2	RELADDR(16)			
25	7-18	LCS(16)			
27	19-18	EX(16)			
18	22-18	IMOP(16)			
34	67-18	SWITCH(16)			

17	29-7	SCRATCHP(17)			SBUS(17)
18	18-2	CONST(17)			
28	31-5	CBUSIN(17)			
20	28-12	HWORD(17)			
21	32-5	DISP(17)			
22	4-16	ICS(17)			
24	3-5	RELADDR(17)			
25	7-16	LCS(17)			
27	19-16	EX(17)			
18	22-16	IMOP(17)			
34	67-16	SWITCH(17)			

AGA

RAJ
80 0130

17 29-9 SCRATCHP(18)
 18 18-3 CONST(18)
 28 31-7 CBUSIN(18)
 20 28-13 HWORD(18)
 21 32-6 DISP(18)
 22 4-14 ICS(18)
 24 3-6 RELADDR(18)
 25 7-14 LCS(18)
 27 19-14 EX(18)
 18 22-14 IMOP(18)
 34 67-14 SWITCH(18)

SBUS(18)

17 29-11 SCRATCHP(19)
 18 18-4 CONST(19)
 28 31-9 CBUSIN(19)
 20 28-14 HWORD(19)
 21 32-9 DISP(19)
 22 4-12 ICS(19)
 24 3-9 RELADDR(19)
 25 7-12 LCS(19)
 27 19-12 EX(19)
 18 22-12 IMOP(19)
 34 67-12 SWITCH(19)

SBUS(19)

17 9-5 SCRATCHP(20)
 18 18-5 CONST(20)
 28 21-3 CBUSIN(20)
 20 8-11 HWORD(20)
 21 32-12 DISP(20)
 22 4-3 ICS(20)
 24 3-12 RELADDR(20)
 25 7-3 LCS(20)
 27 19-3 EX(20)
 18 22-3 IMOP(20)
 34 67-3 SWITCH(20)

SBUS(20)

17 9-7 SCRATCHP(21)
 18 18-6 CONST(21)
 28 21-5 CBUSIN(21)
 20 8-12 HWORD(21)
 21 32-15 DISP(21)
 22 4-5 ICS(21)
 24 3-15 RELADDR(21)
 25 7-5 LCS(21)
 27 19-5 EX(21)
 18 22-5 IMOP(21)
 34 67-5 SWITCH(21)

SBUS(21)

17 9-9 SCRATCHP(22)
 18 18-7 CONST(22)
 28 21-7 CBUSIN(22)
 20 8-13 HWORD(22)
 21 32-16 DISP(22)
 22 4-7 ICS(22)
 24 3-16 RELADDR(22)
 25 7-7 LCS(22)
 27 19-7 EX(22)
 18 22-7 IMOP(22)
 34 67-7 SWITCH(22)

SBUS(22)

17 9-11 SCRATCHP(23)
 18 18-9 CONST(23)
 28 21-9 CBUSIN(23)
 20 8-14 HWORD(23)
 21 32-19 DISP(23)
 22 4-9 ICS(23)
 24 3-9 RELADDR(23)
 25 7-9 LCS(23)
 27 19-9 EX(23)
 18 22-9 IMOP(23)
 34 67-9 SWITCH(23)

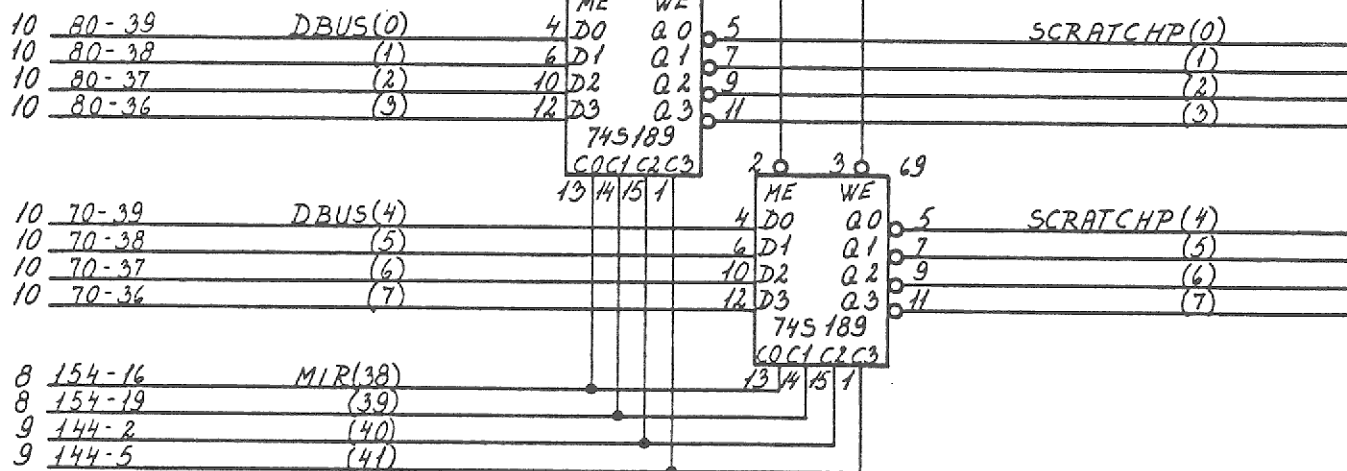
SBUS(23)

RGA

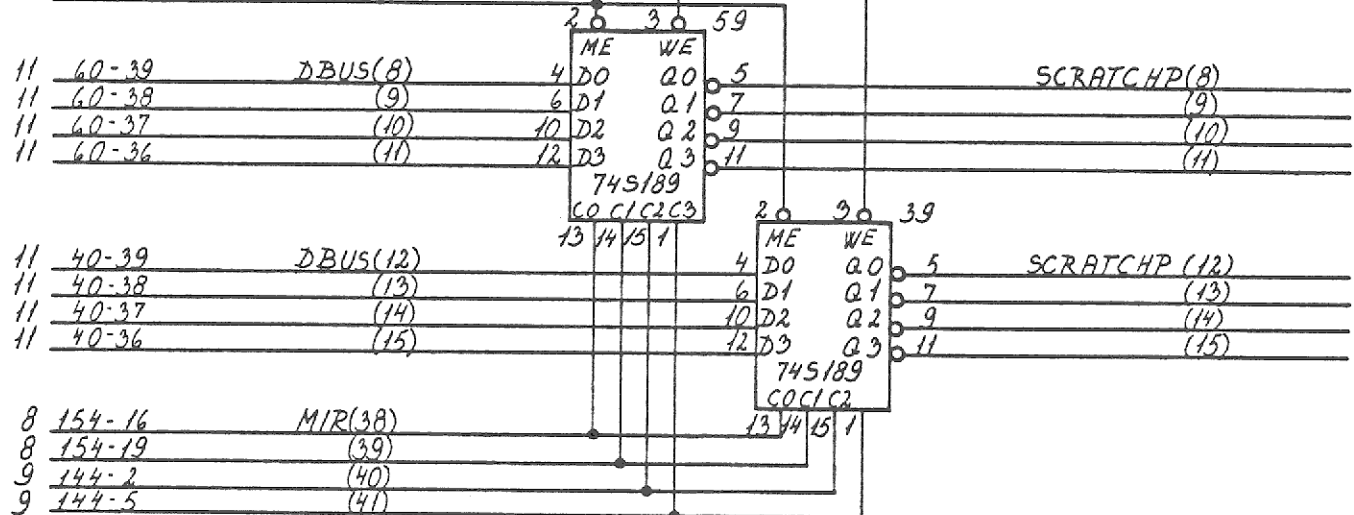
ARJ
800130

37 147-8 CP SCRATCHP
36 137-15 7 EN SCRATCHP

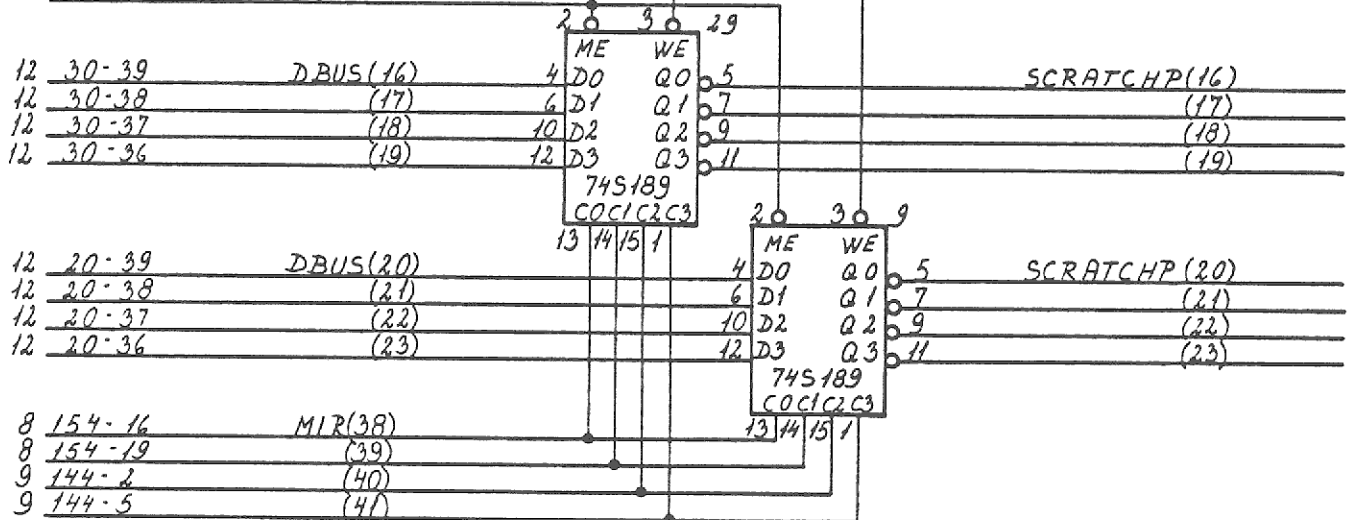
83



37 147-8 CP SCRATCHP
36 137-15 7 EN SCRATCHP



37 147-8 CP SCRATCHP
36 137-15 7 EN SCRATCHP



AGA

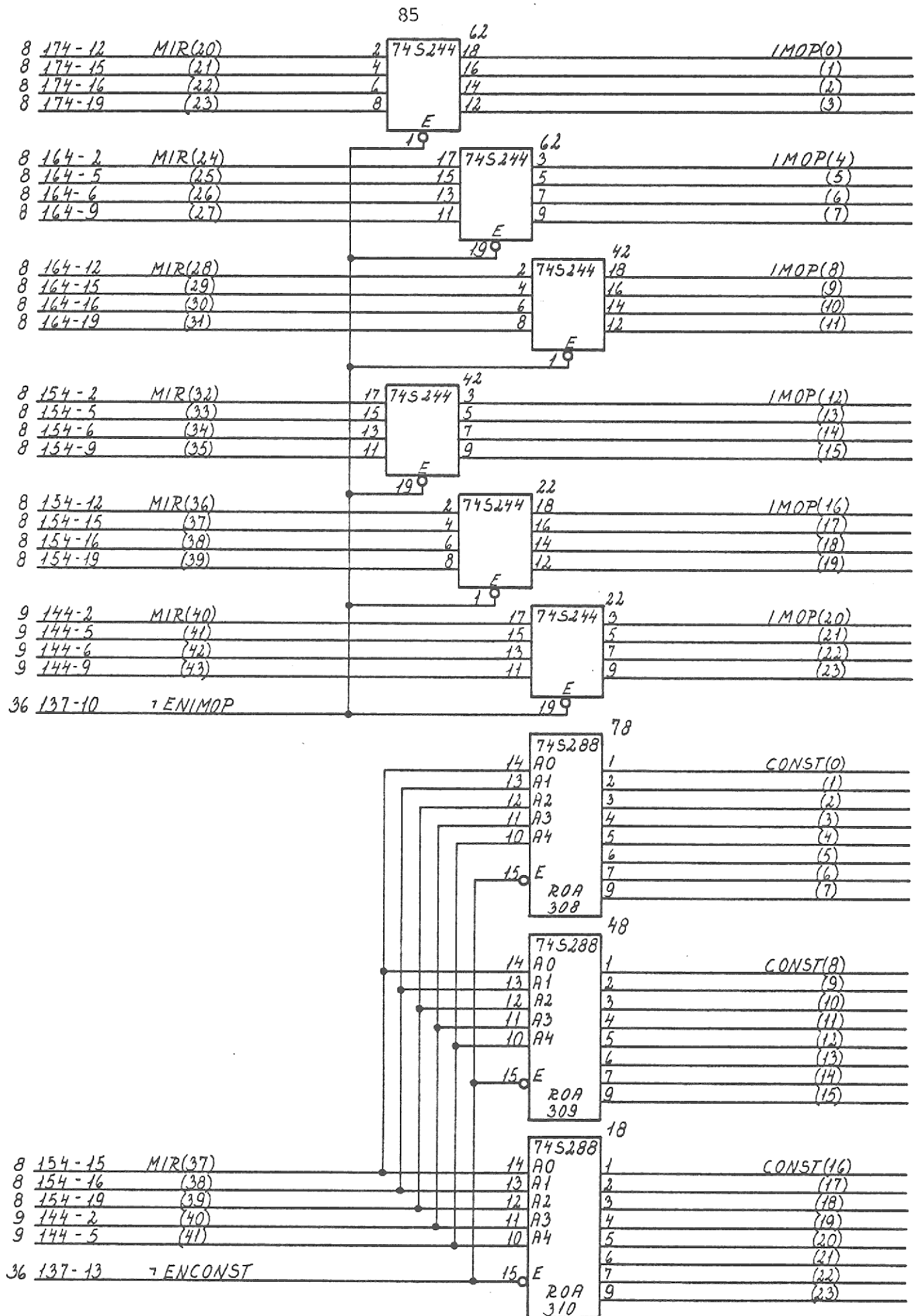
RAJ
80 0130

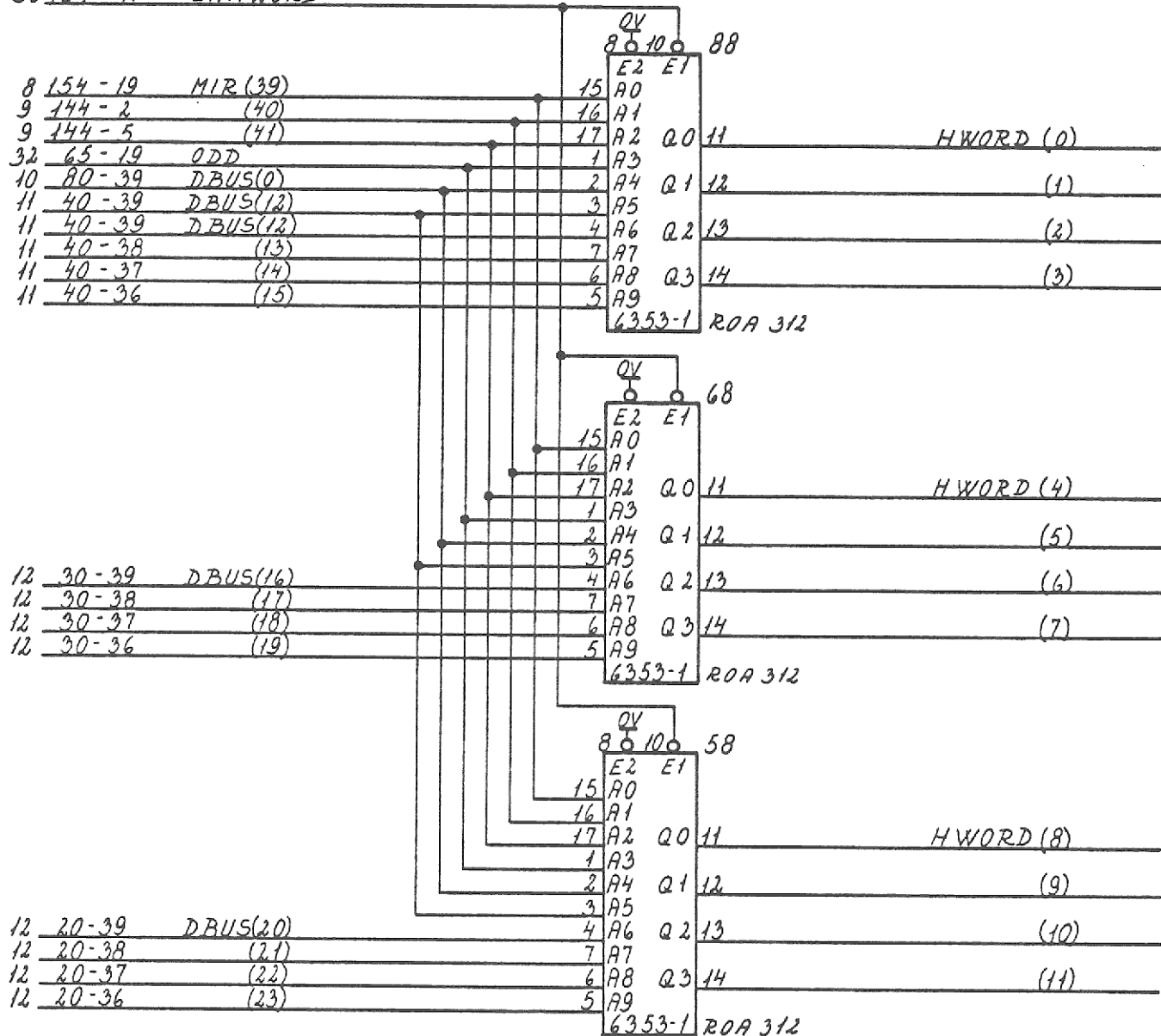
CPU 821

SCRATCHPAD MEMORY, 16 WORDS x 24 BITS

R 12938

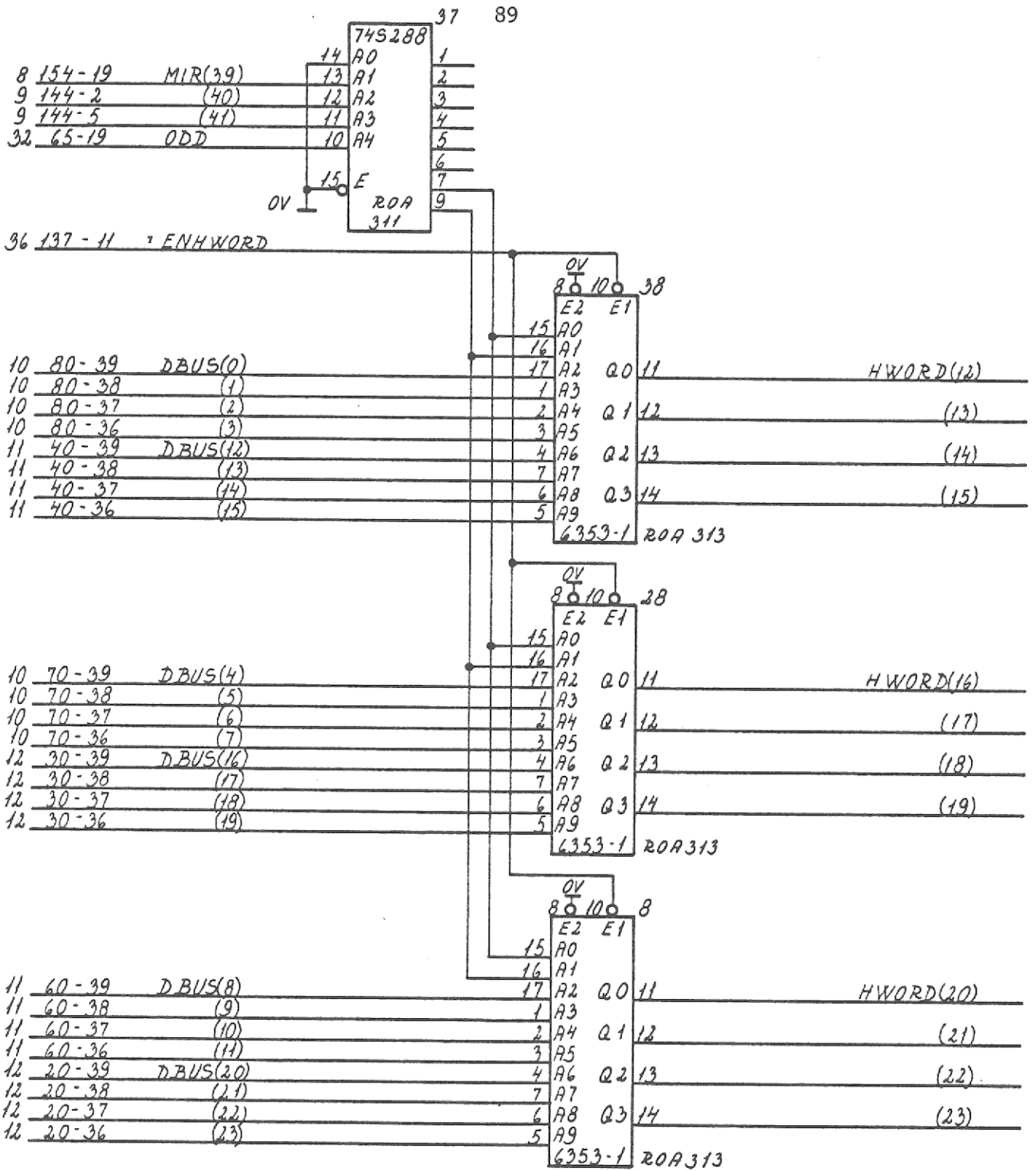
17





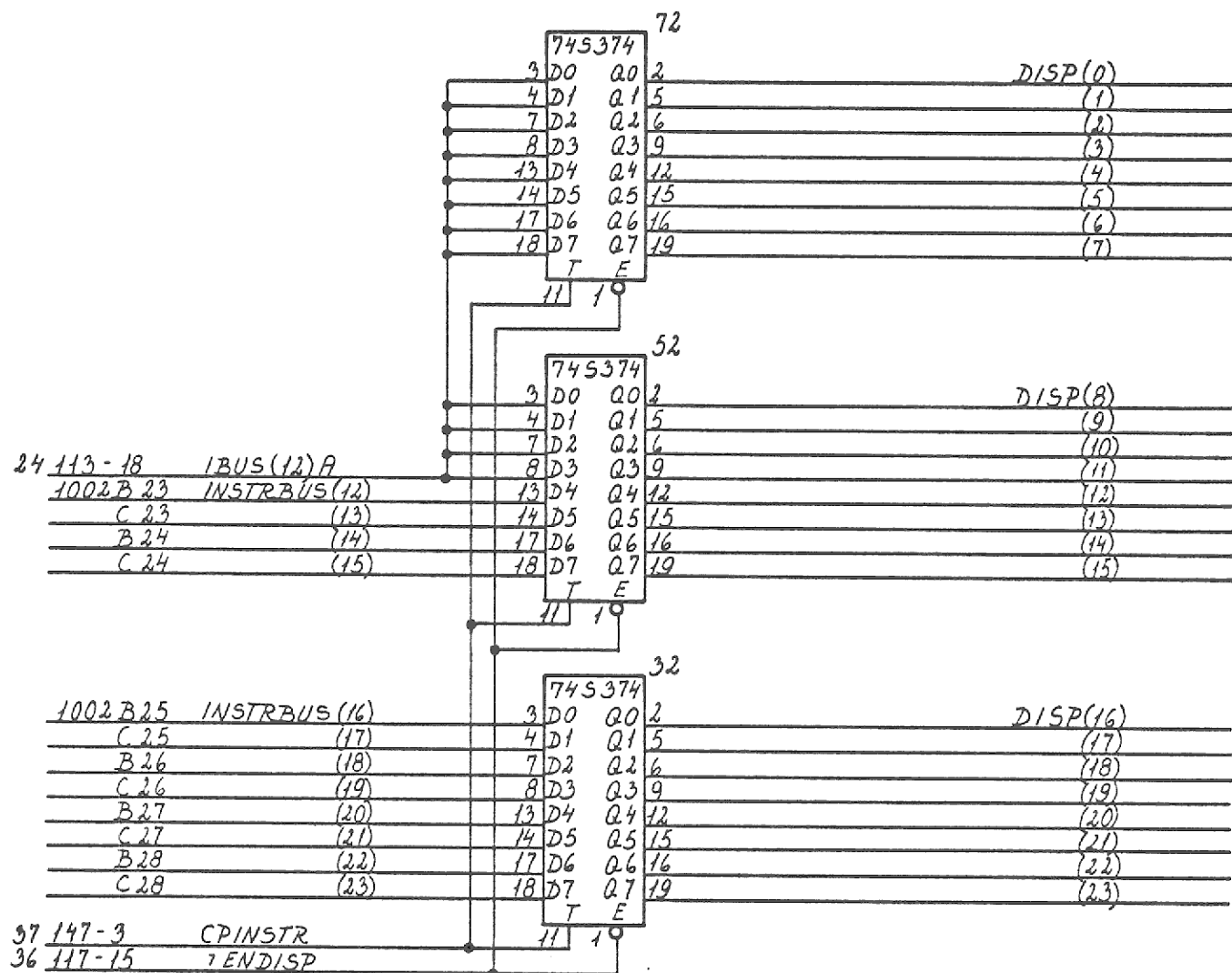
AGA

ARJ
80 0130



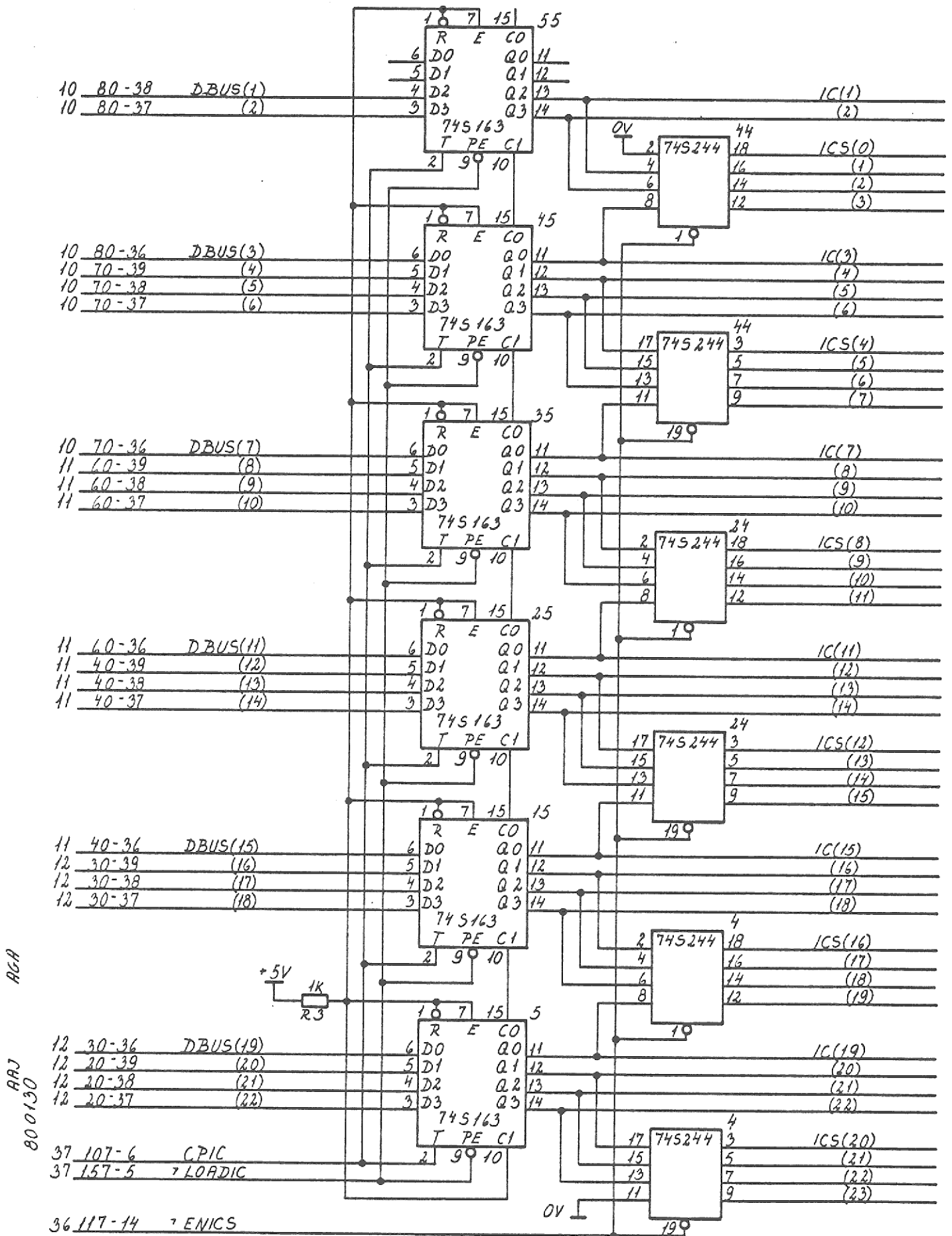
AGA

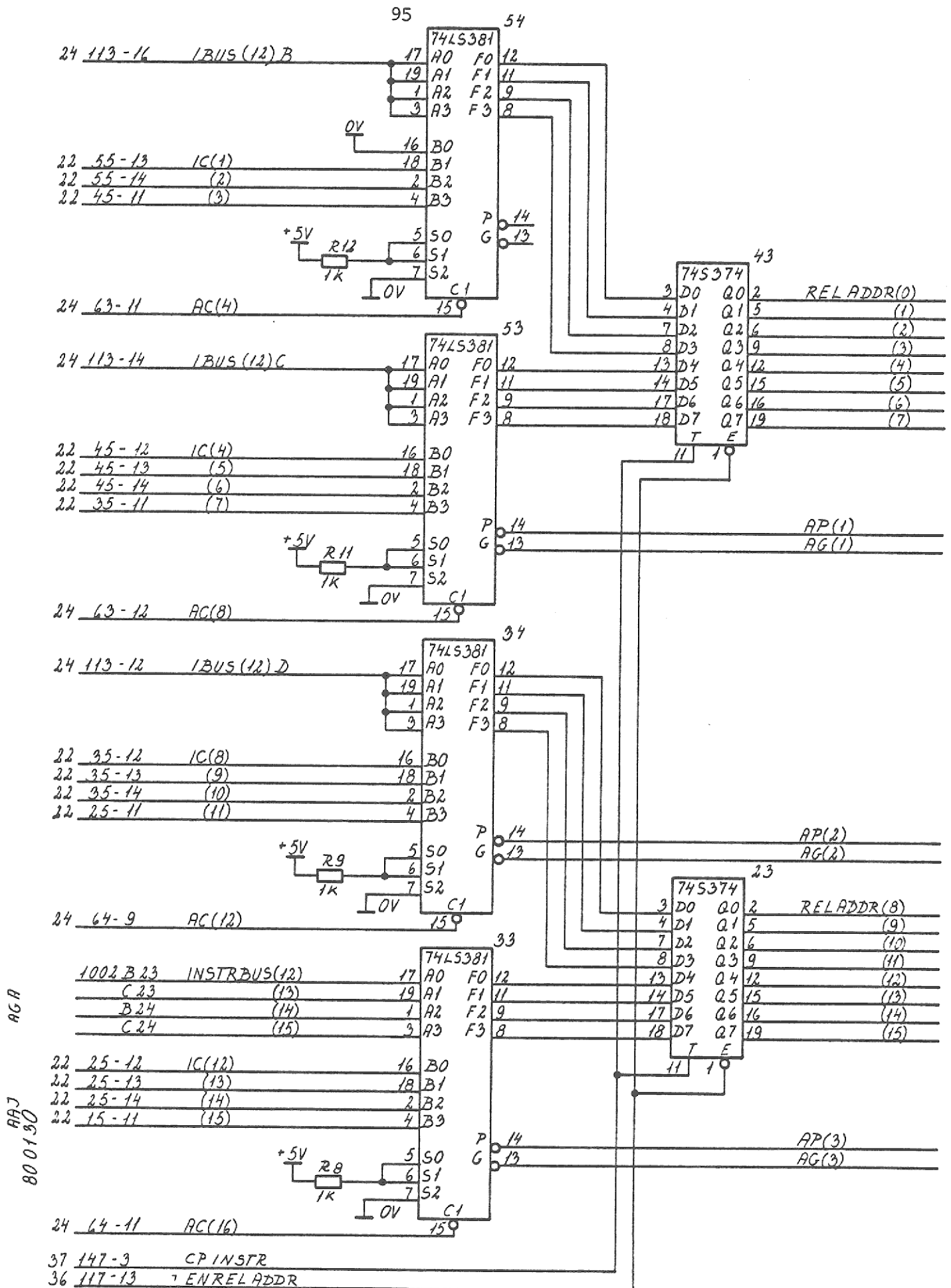
ARJ
80 0130

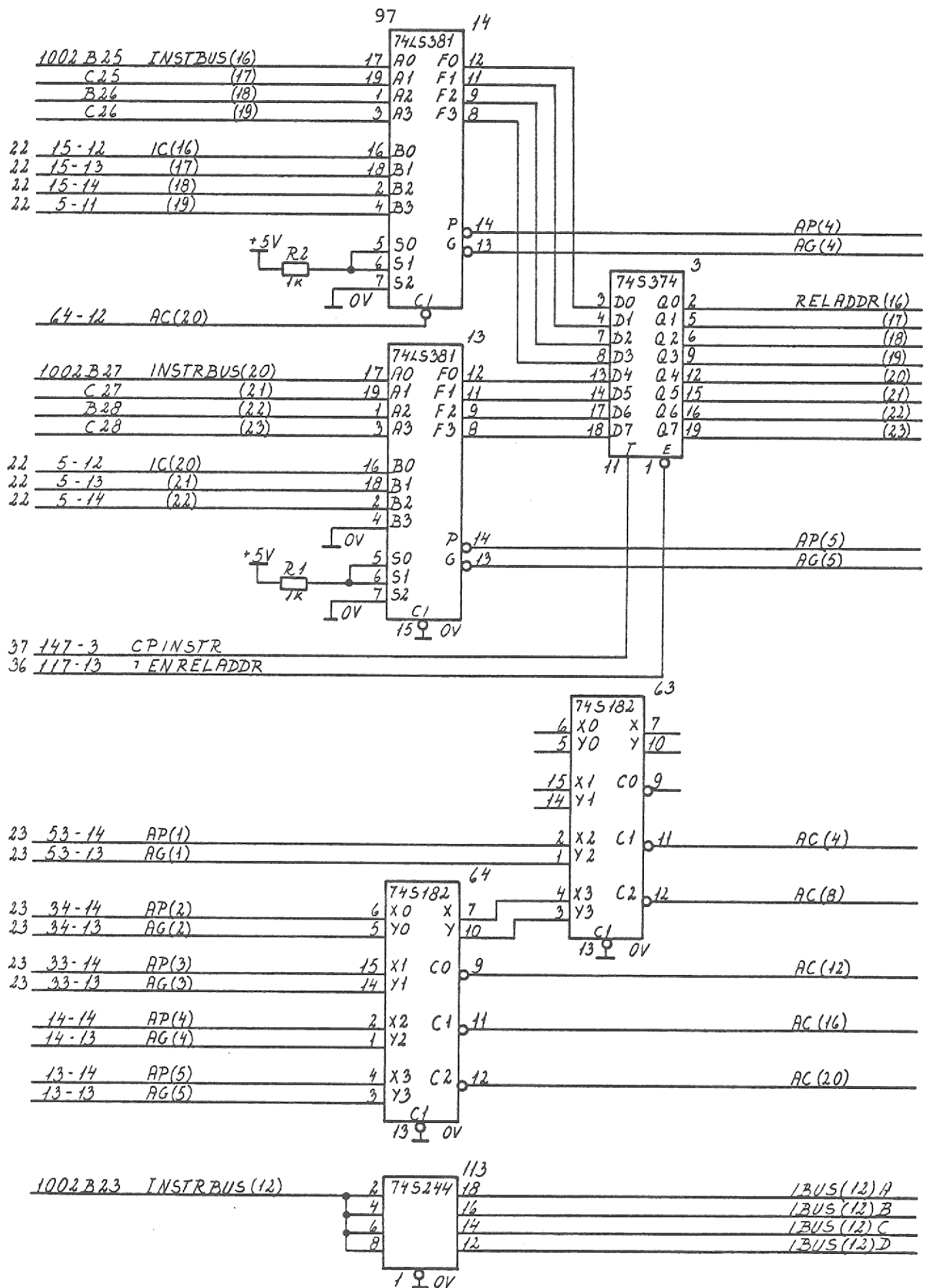


AGA

ARJ
80 0130

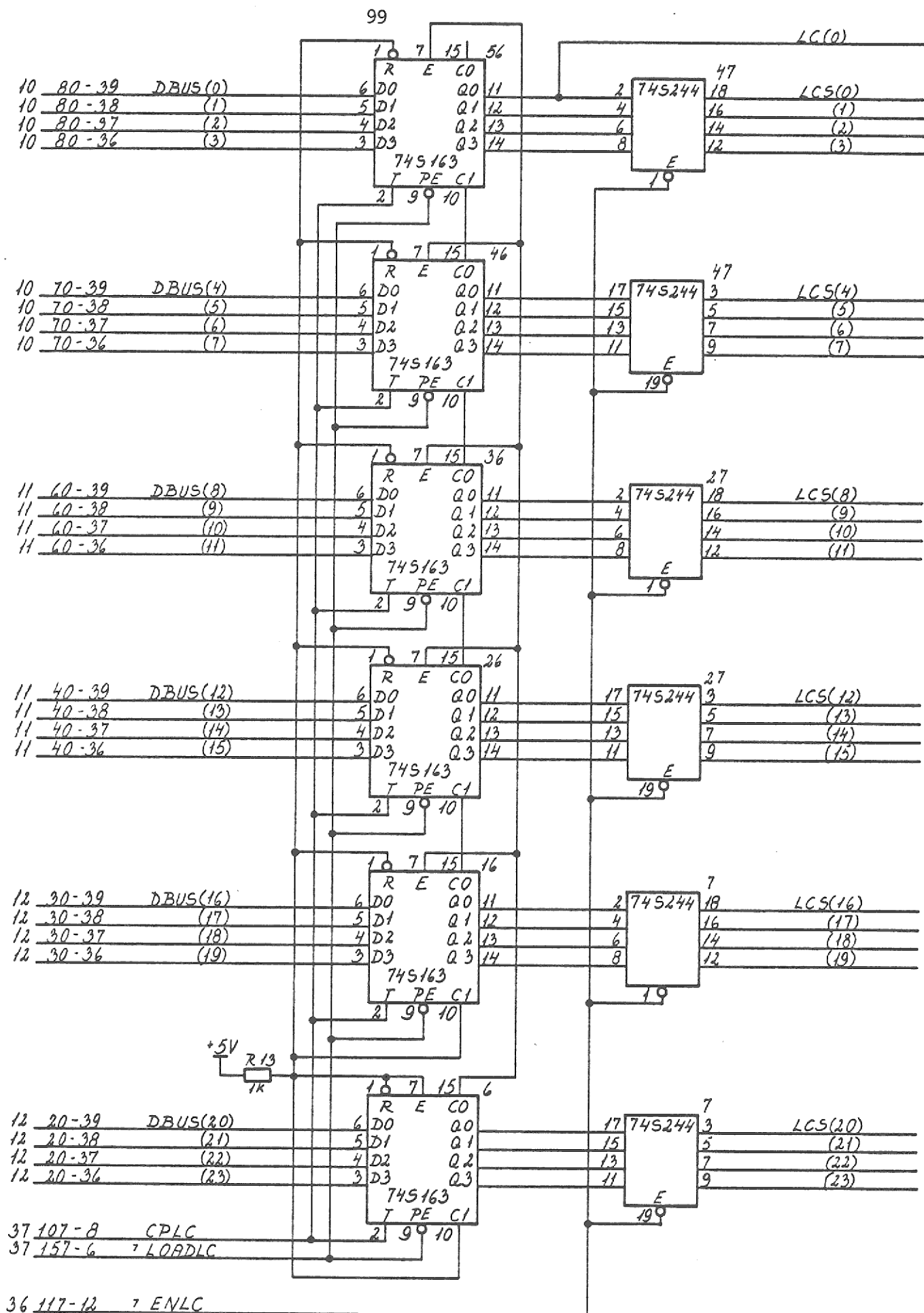






AGA

RAJ
800130



CPU821

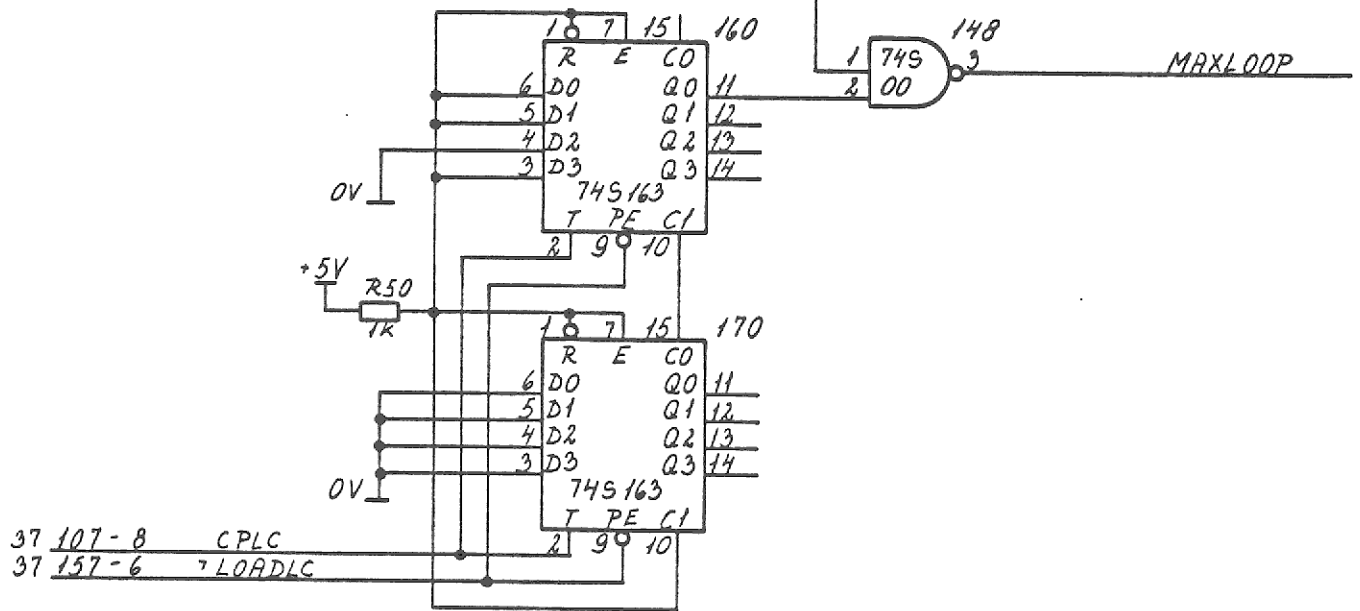
LOOP COUNTER

R 12946

25

25 56-11 LC(0)

101



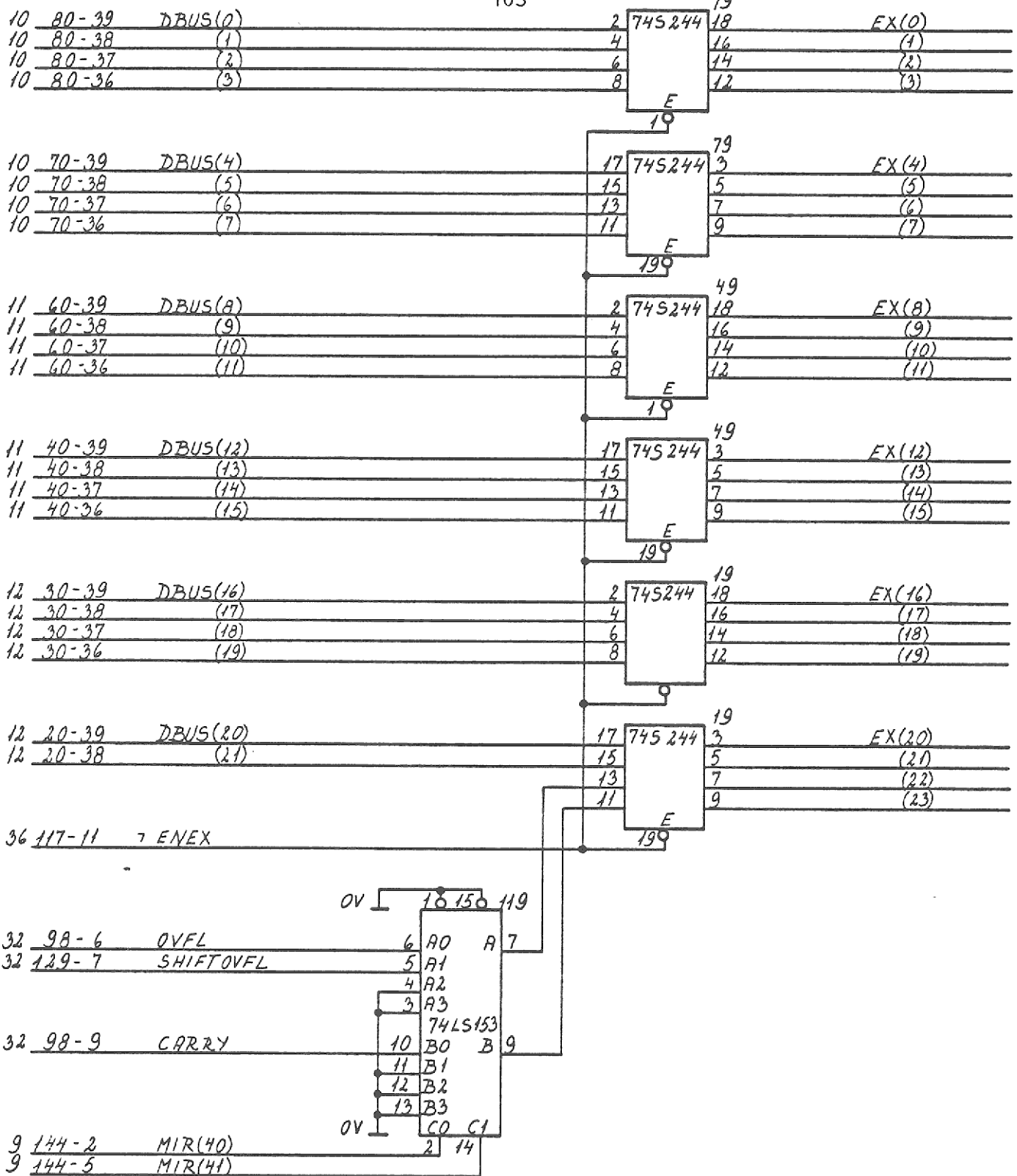
37 107-8 CPLC
37 157-6 LOADLC

AGA

AAJ
800130

CPU 821
R 12947

MAXIMUM LOOP COUNT CONTROL



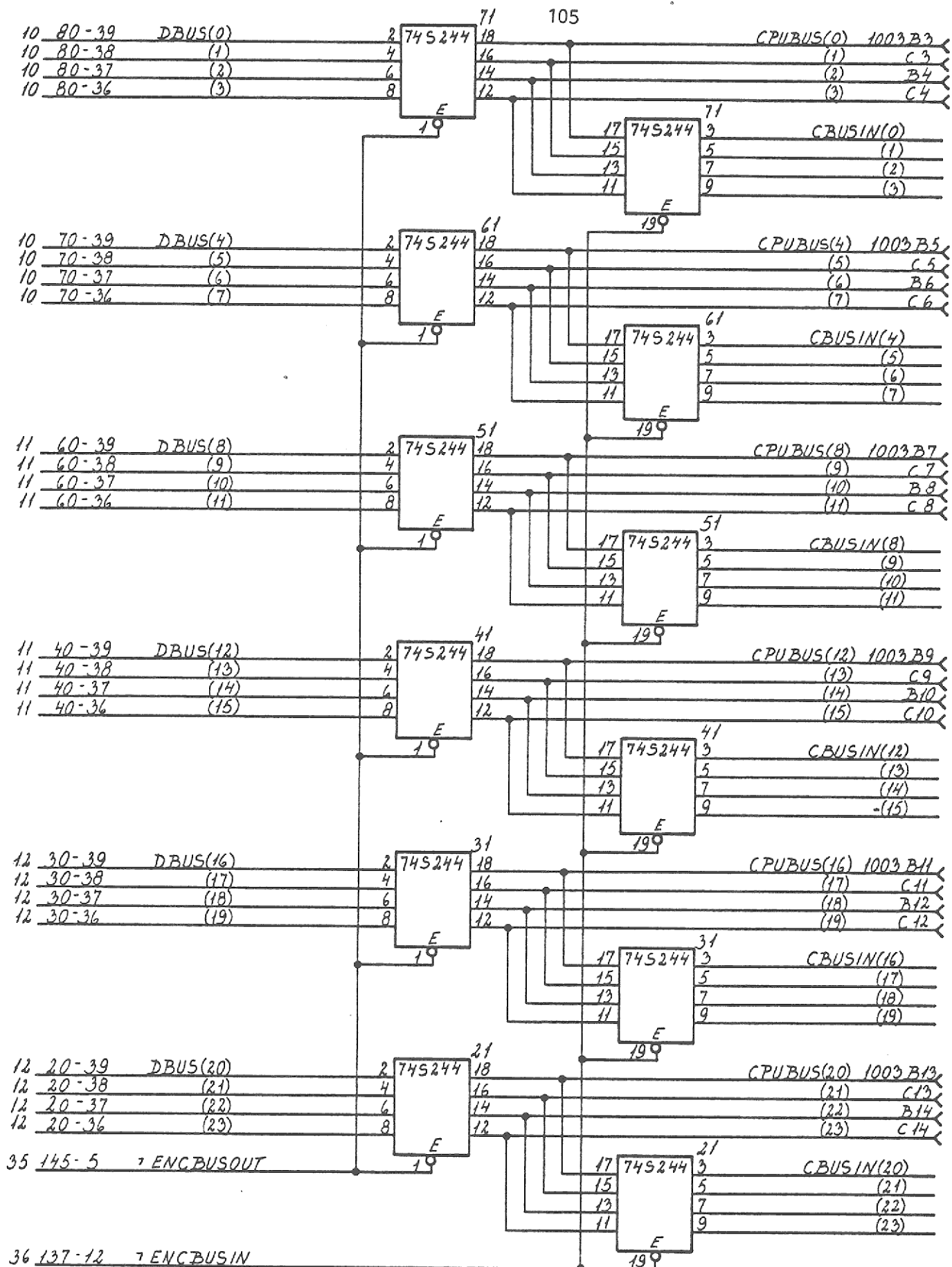
AGA

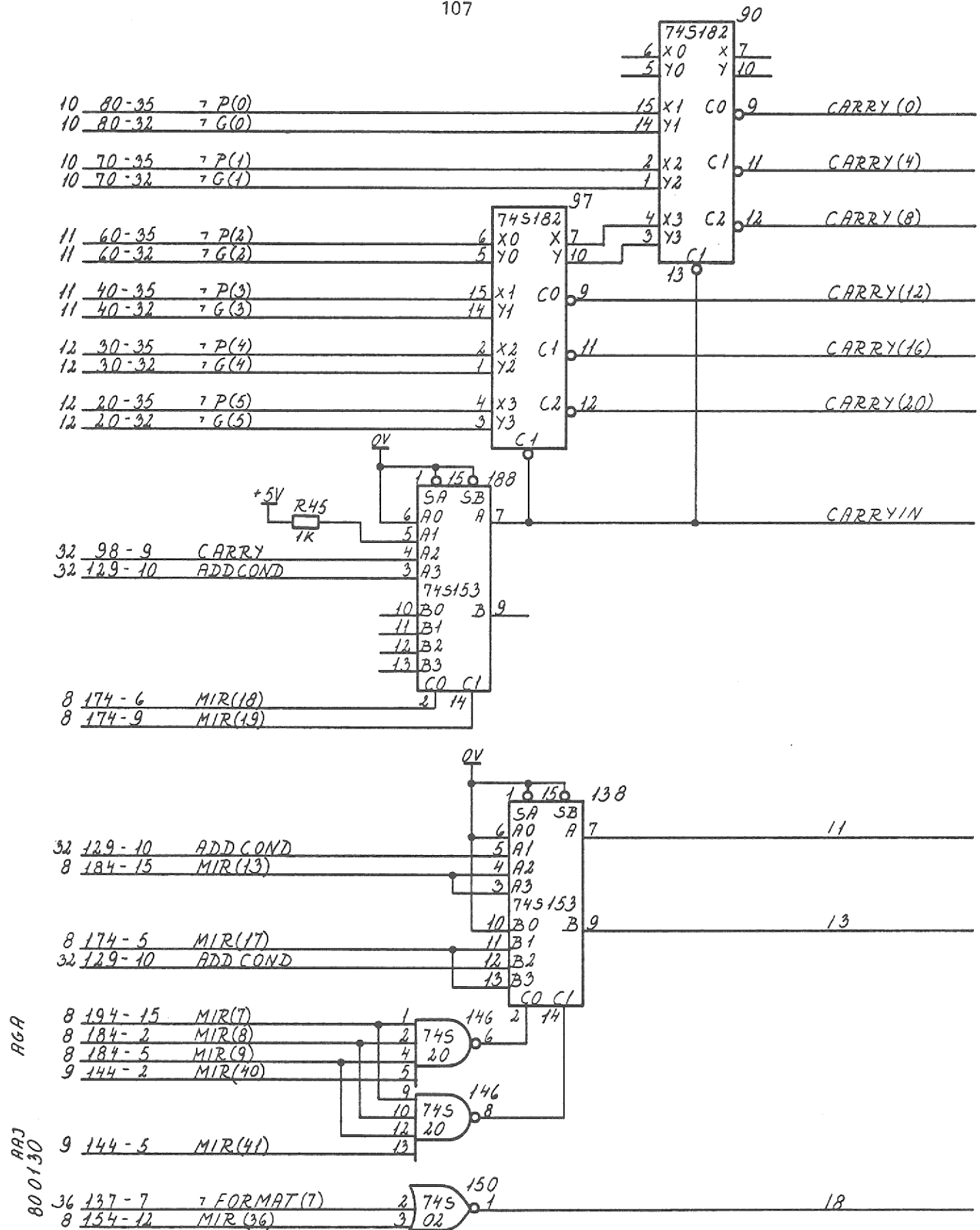
AAJ
800130

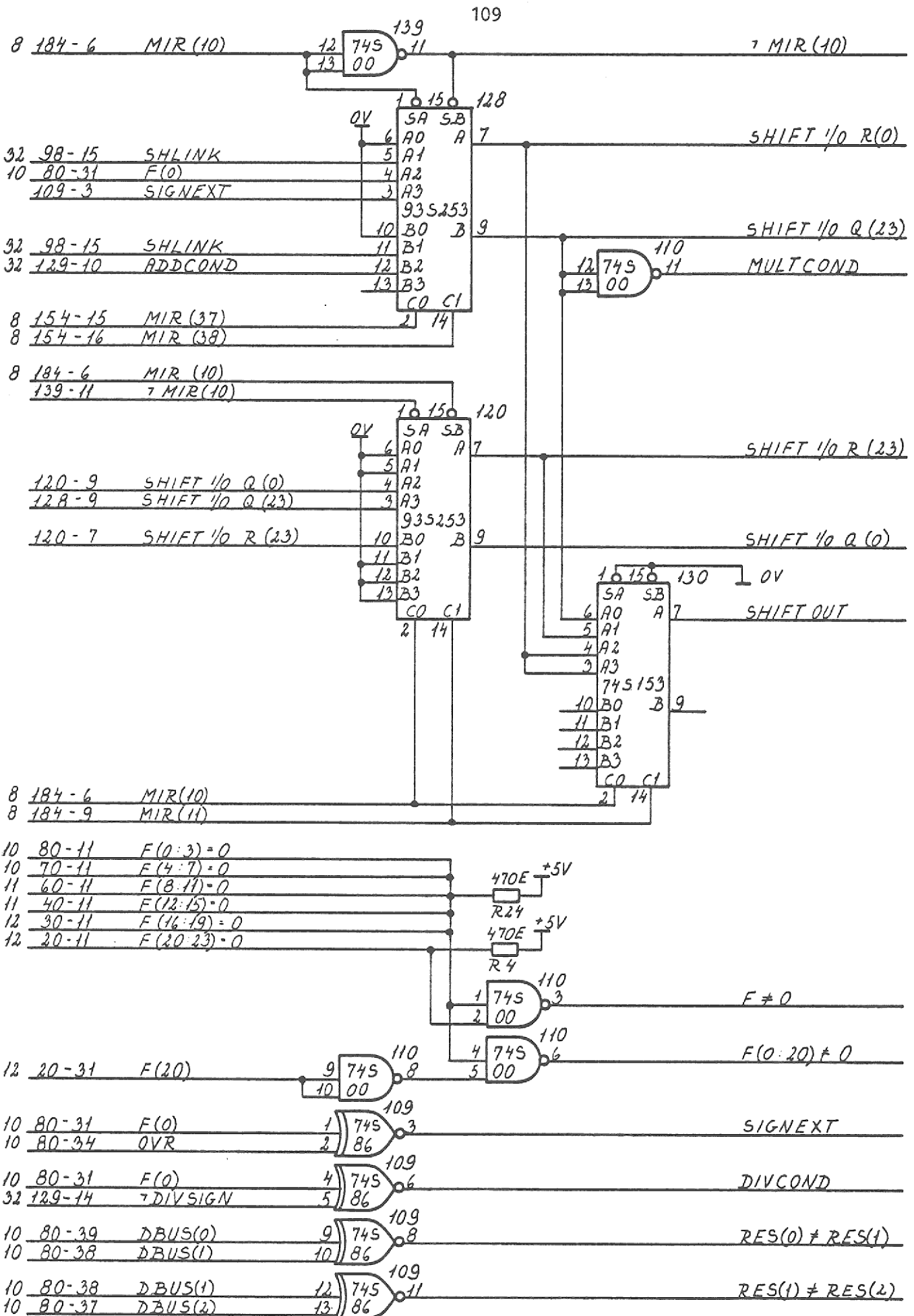
AG-A

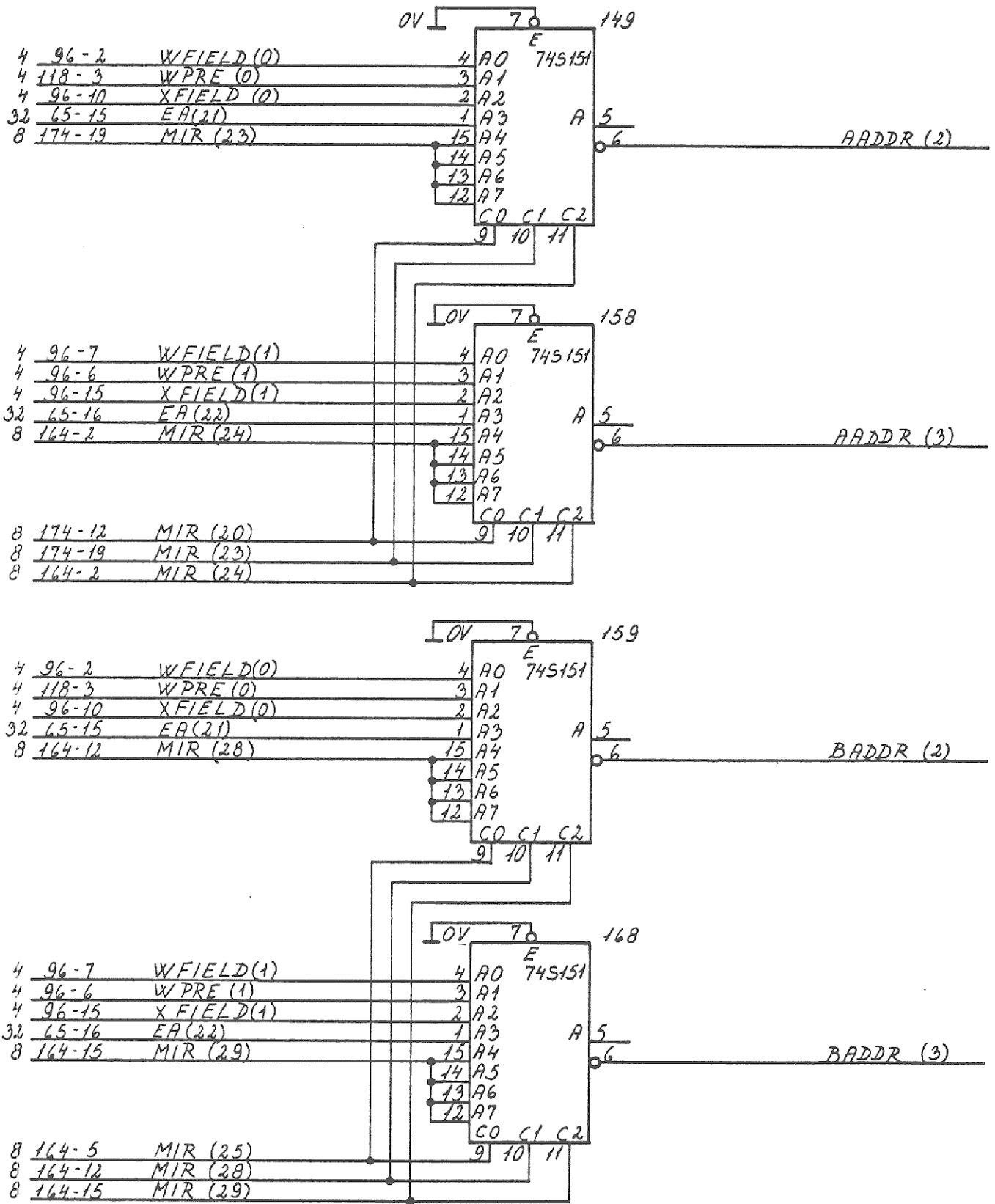
RAJ

800130





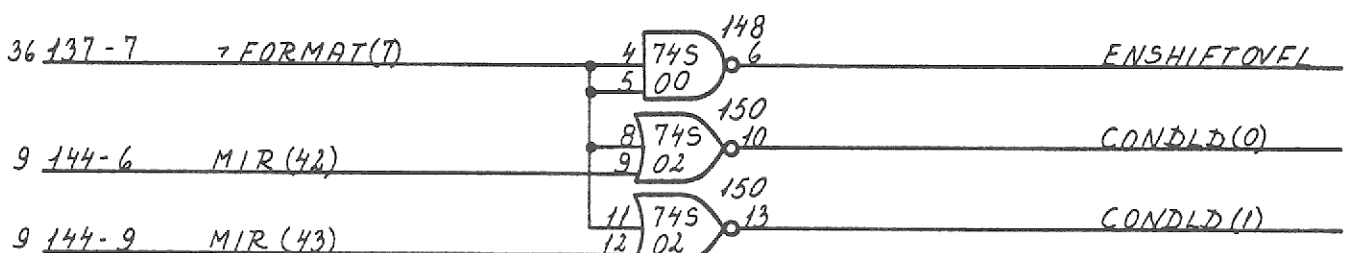
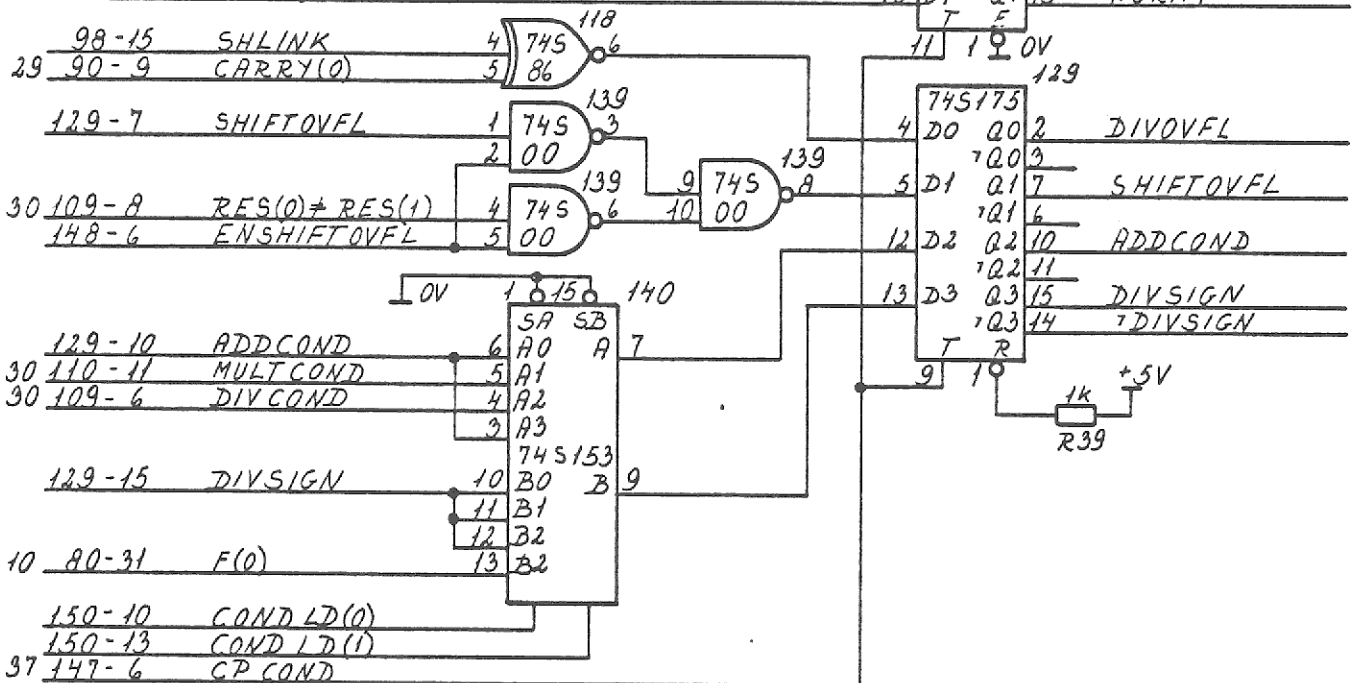
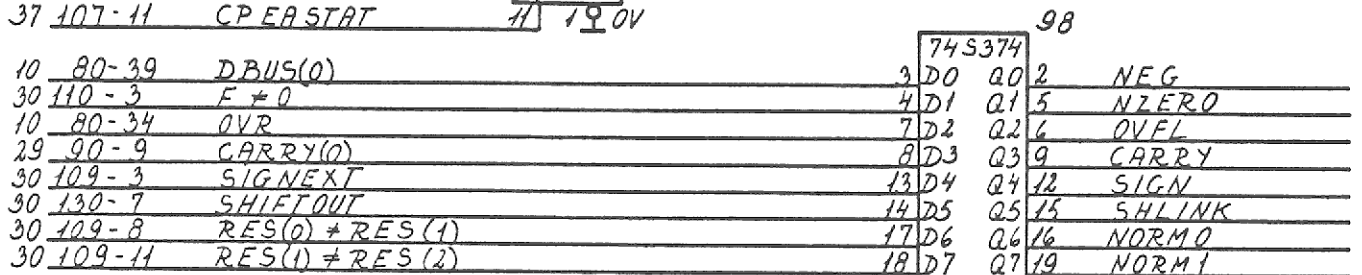
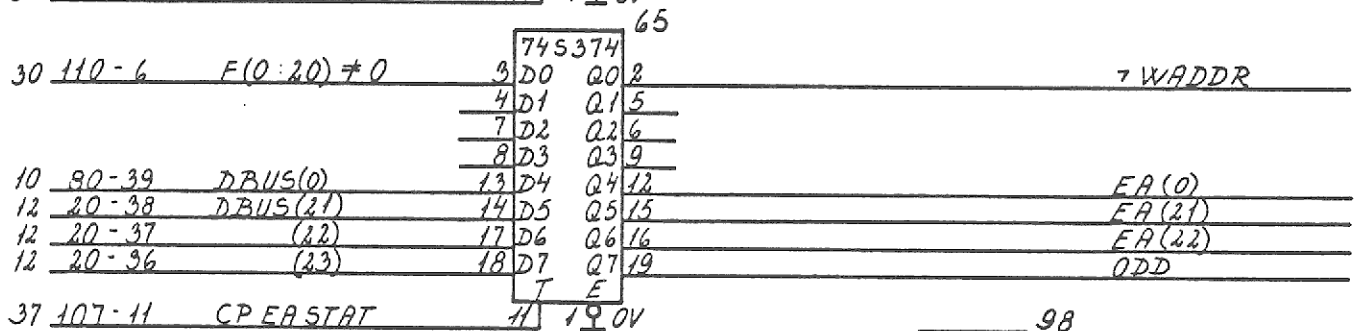
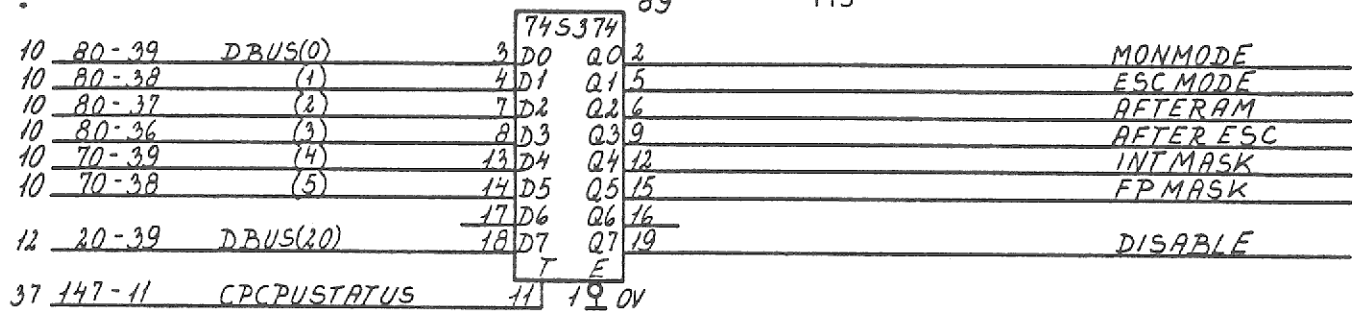


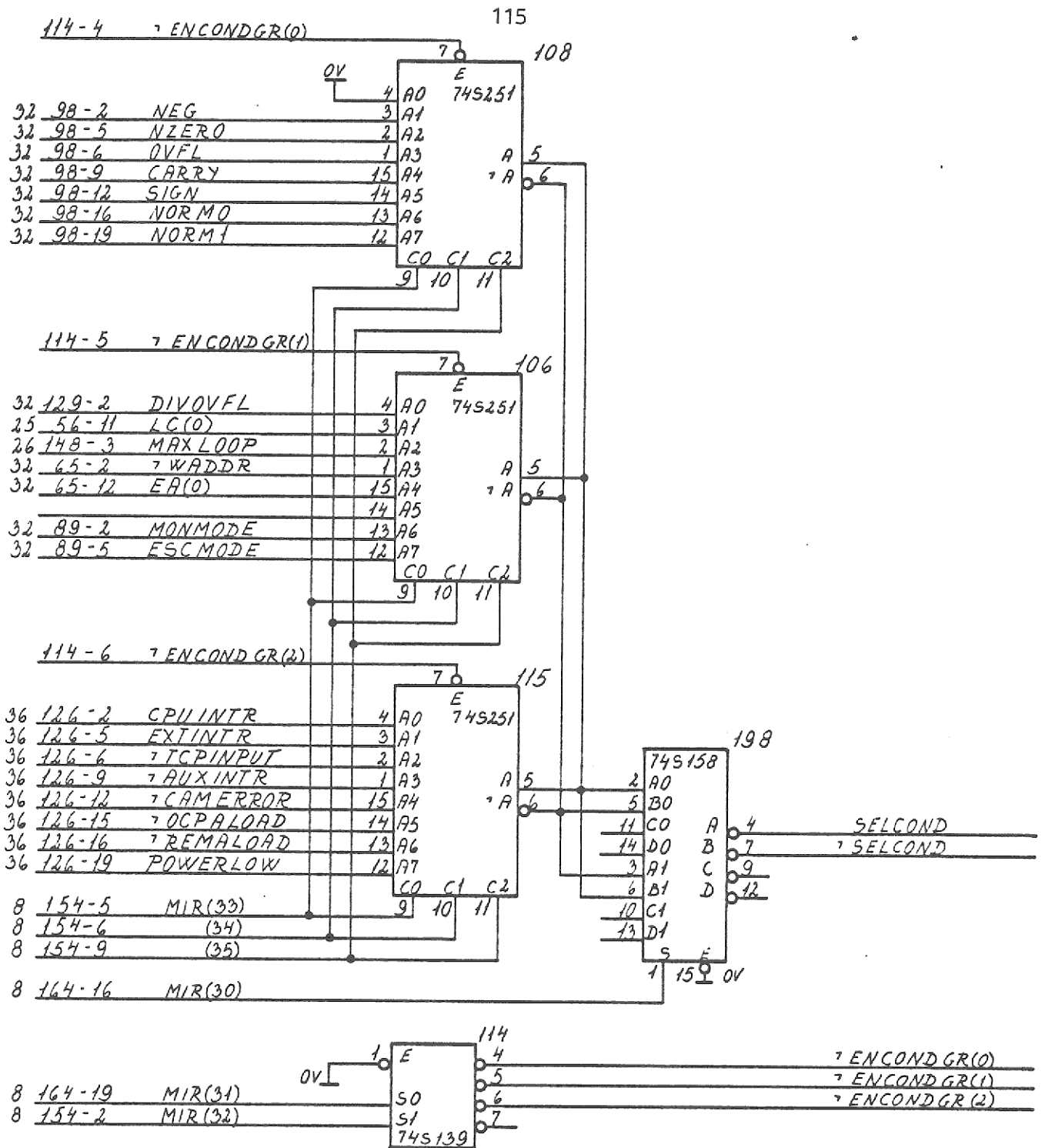


AGA

AAJ

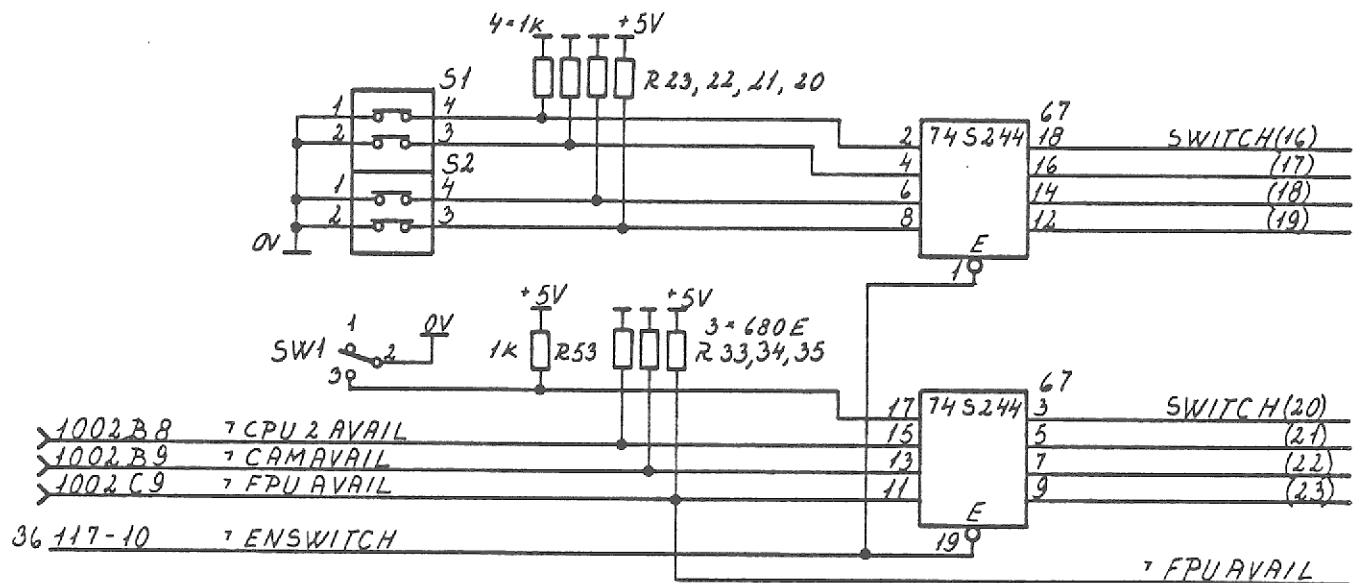
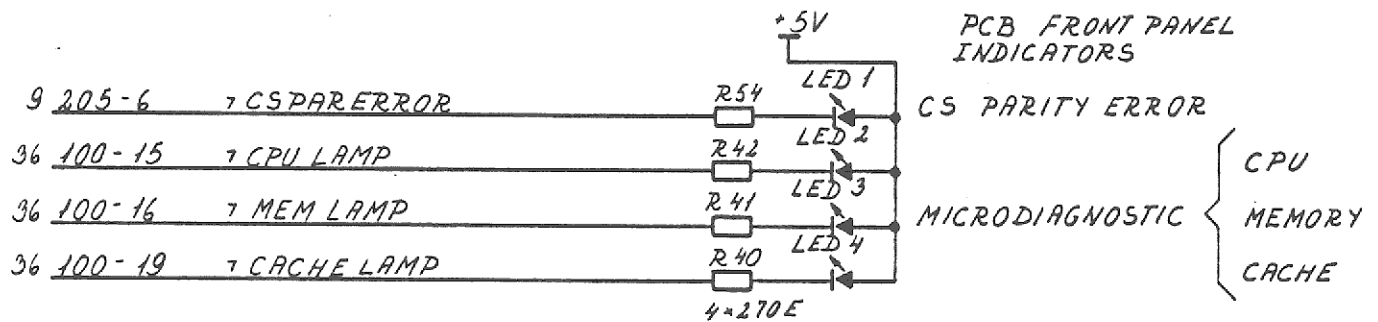
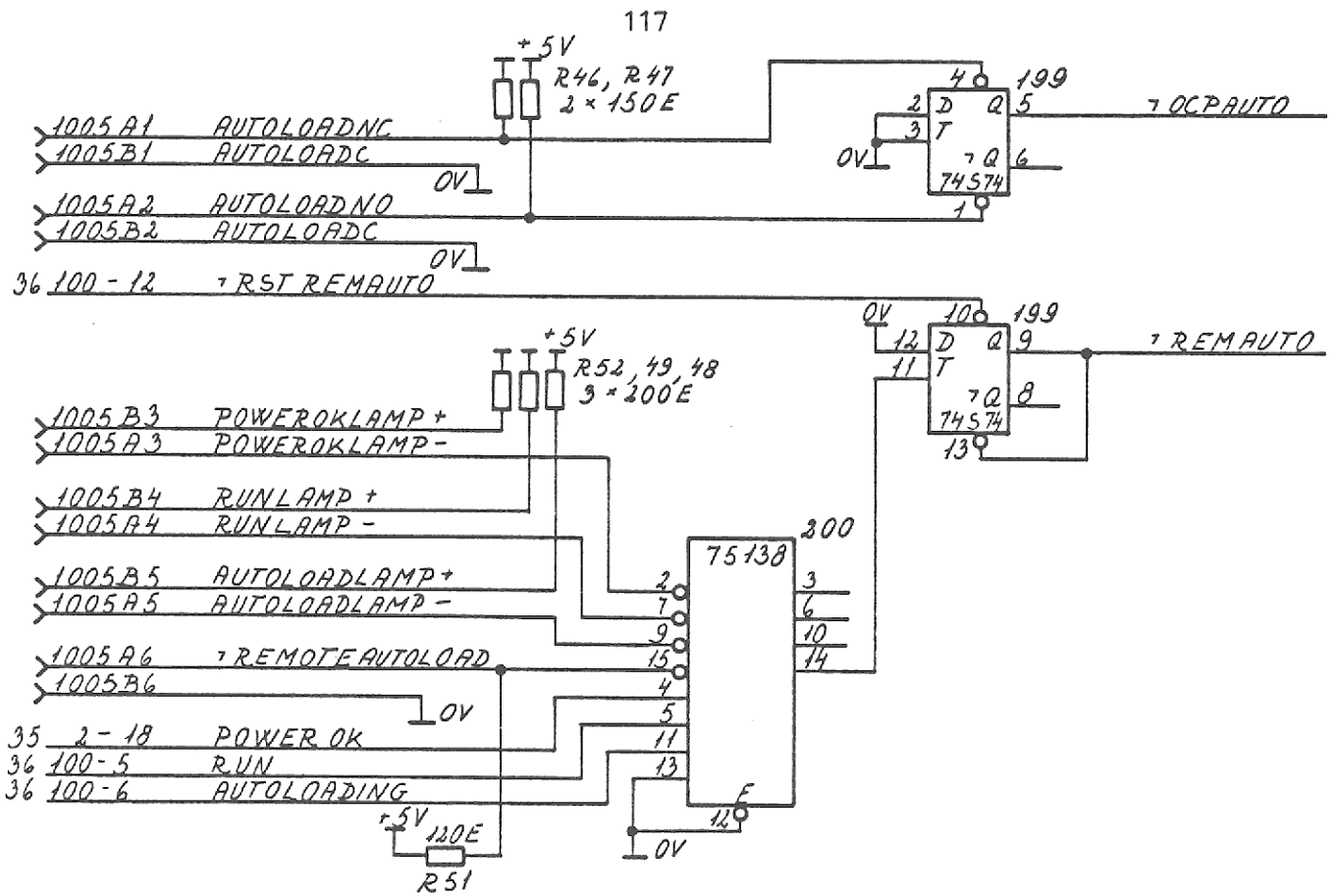
800130



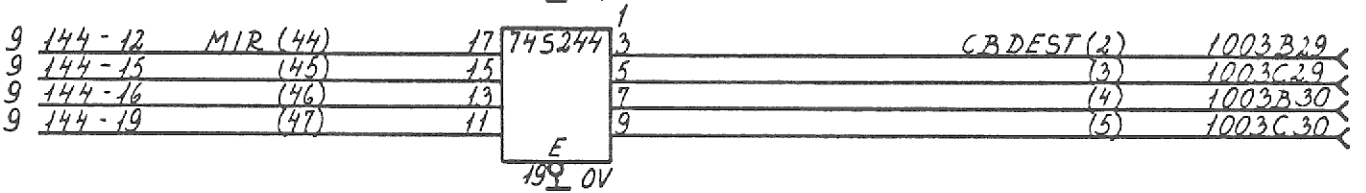
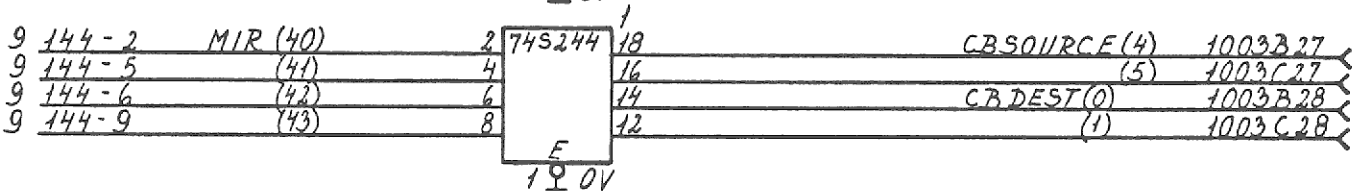
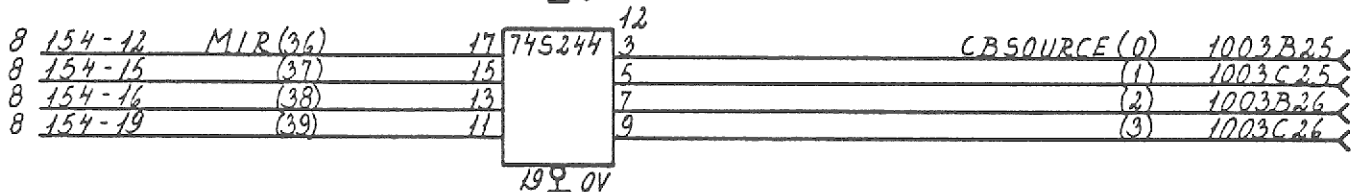
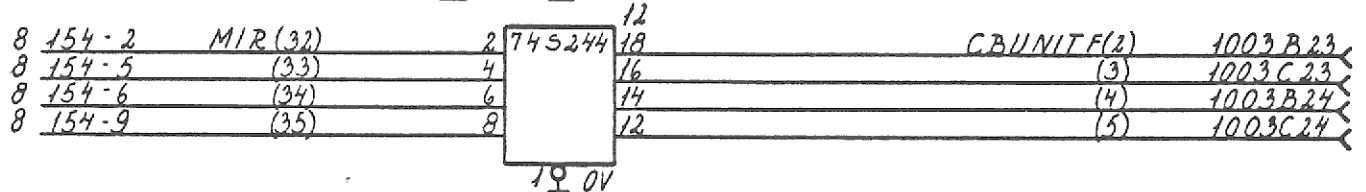
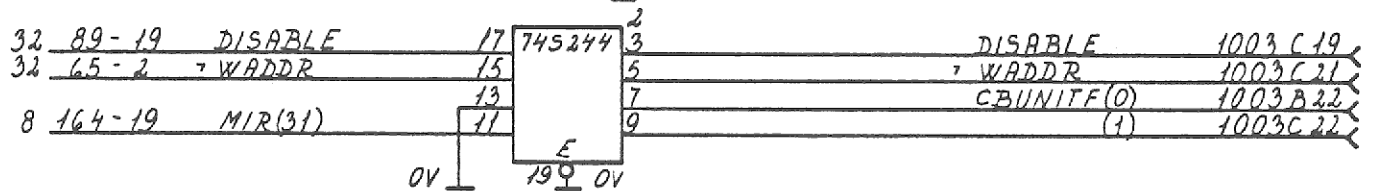
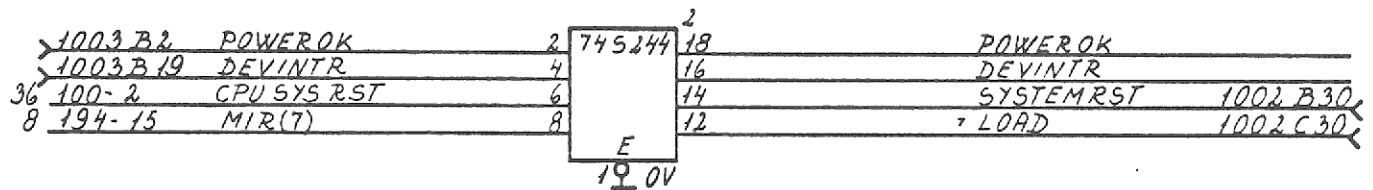
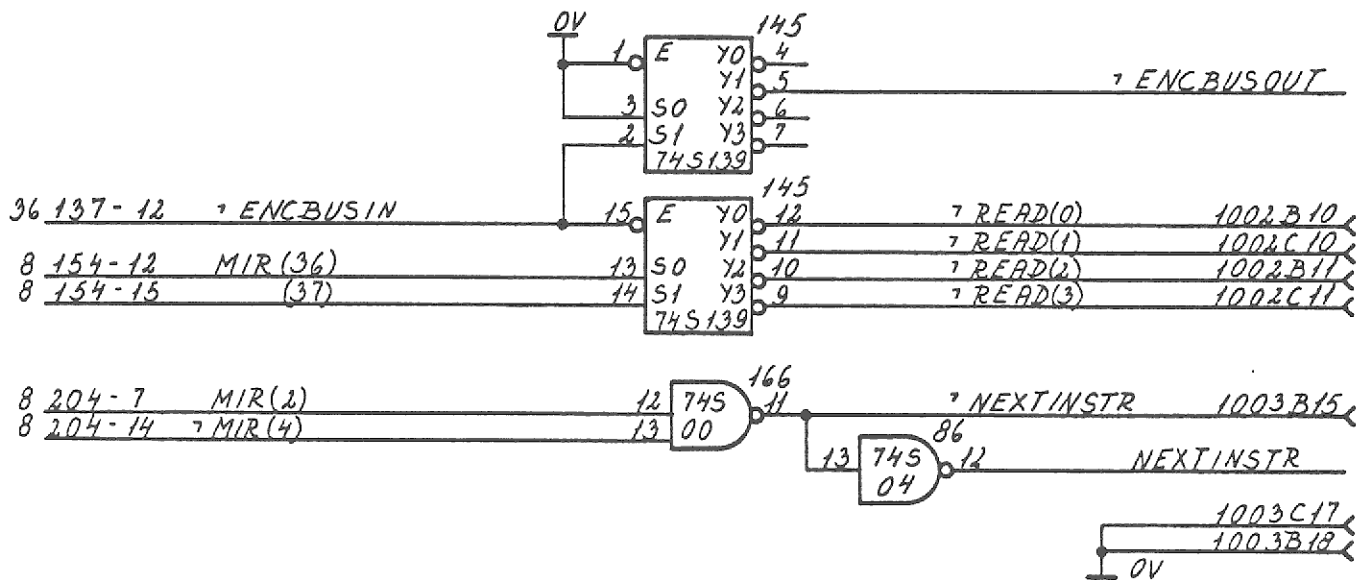


AGA

AAJ
800130



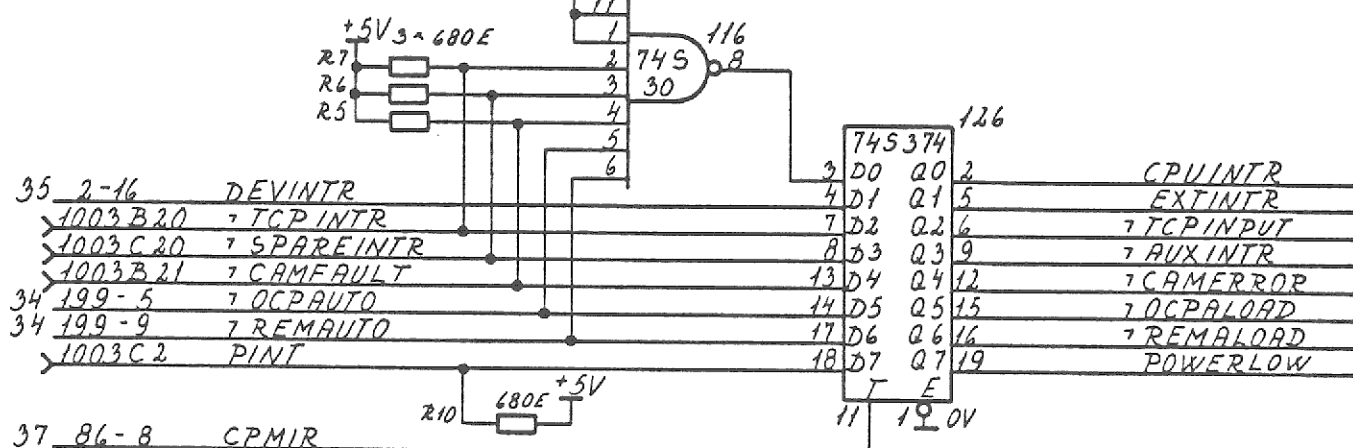
1001B2 → SELIN(0)
 1001B3 → SEL OUT(0)
 1001B4 → SELIN(1)
 1001B5 → SEL OUT(1)



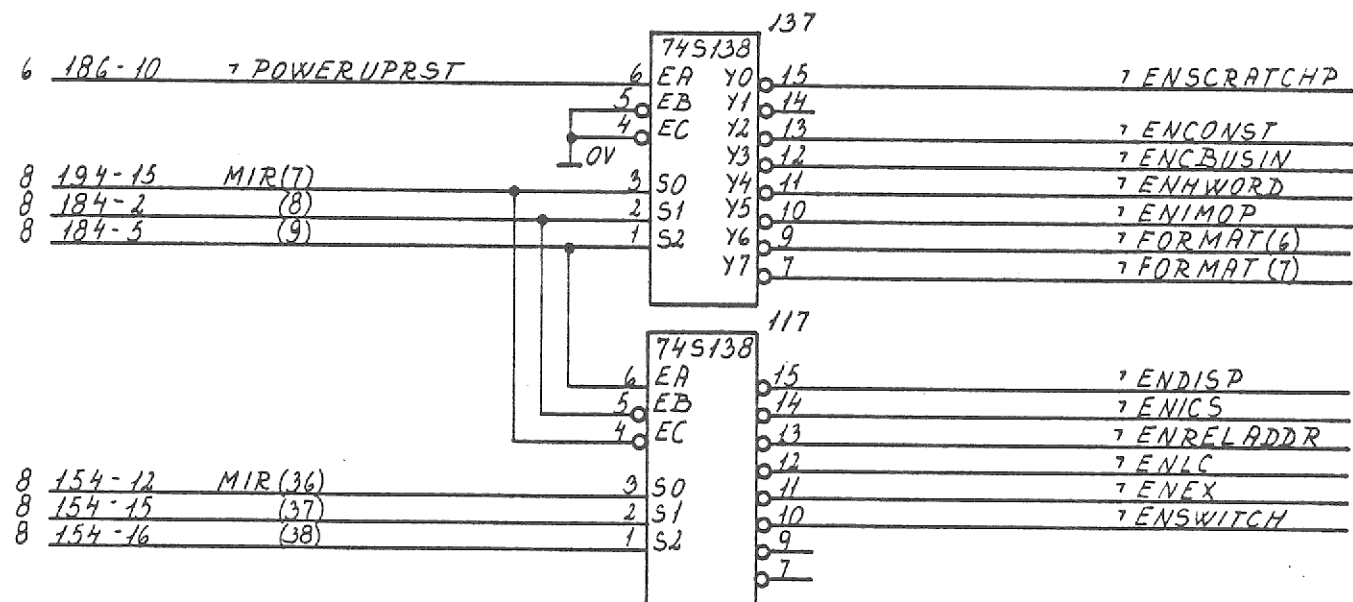
12	30-39	DBUS(16)	3	D0	Q0	2	CPU SYS RST
12	30-38	(17)	4	D1	Q1	5	RUN
12	30-37	(18)	7	D2	Q2	6	AUTOLOADING
12	30-36	(19)	8	D3	Q3	9	SINGLE INSTR
12	20-39	(20)	13	D4	Q4	12	RST REM AUTO
12	20-38	(21)	14	D5	Q5	15	CPU LAMP
12	20-37	(22)	17	D6	Q6	16	MEM LAMP
12	20-36	(23)	18	D7	Q7	19	CACHE LAMP

37 107-3 CPGNTRLOUT 11 10 OV

100-9 SINGLE INSTR



37 86-8 CPMIR



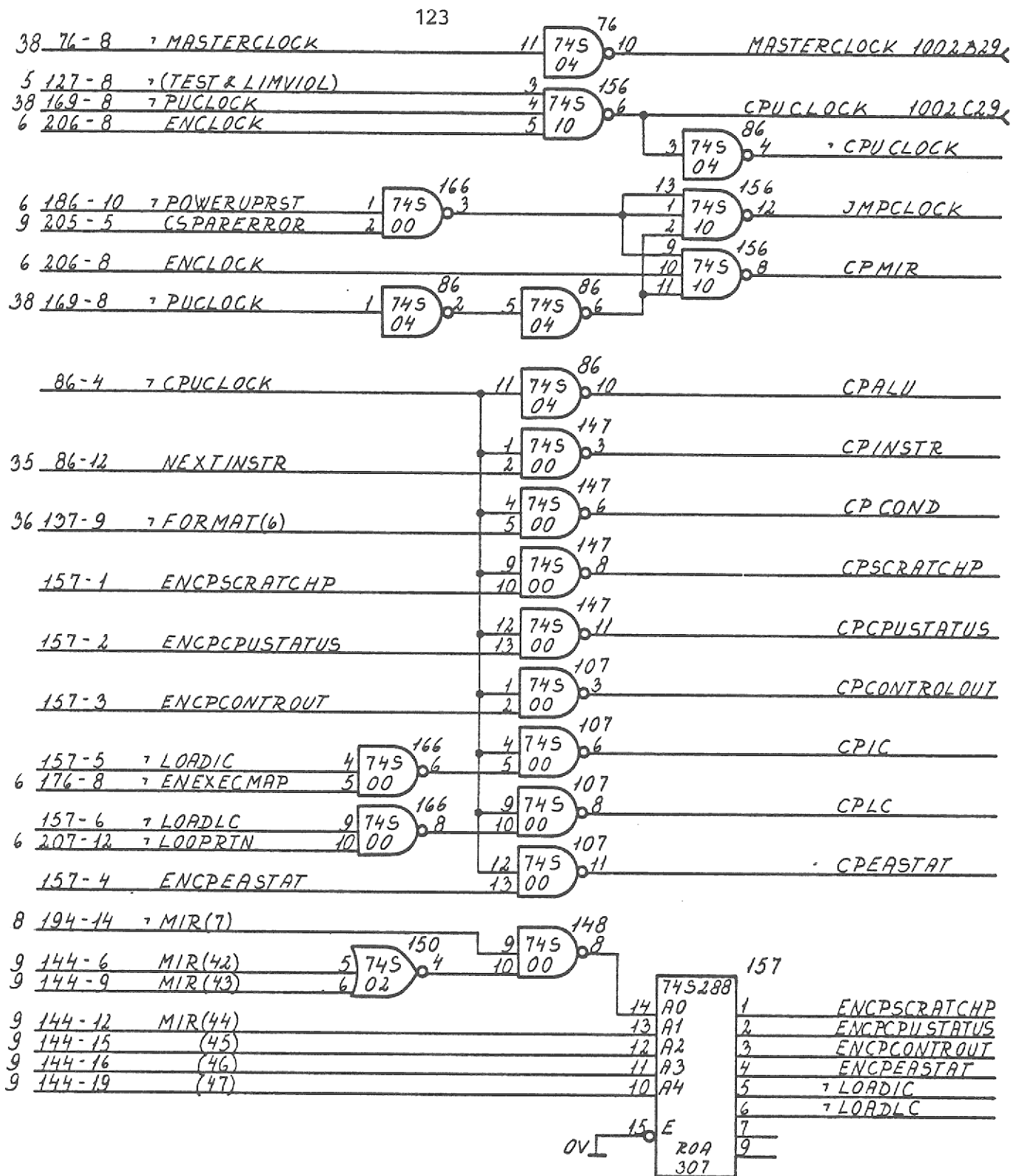
CPU 821

R 12957

CONTROL OUTPUT REGISTER, INTERRUPT STATUS REGISTER 36
& SBUS SOURCE ADDRESS DECODING

AGA

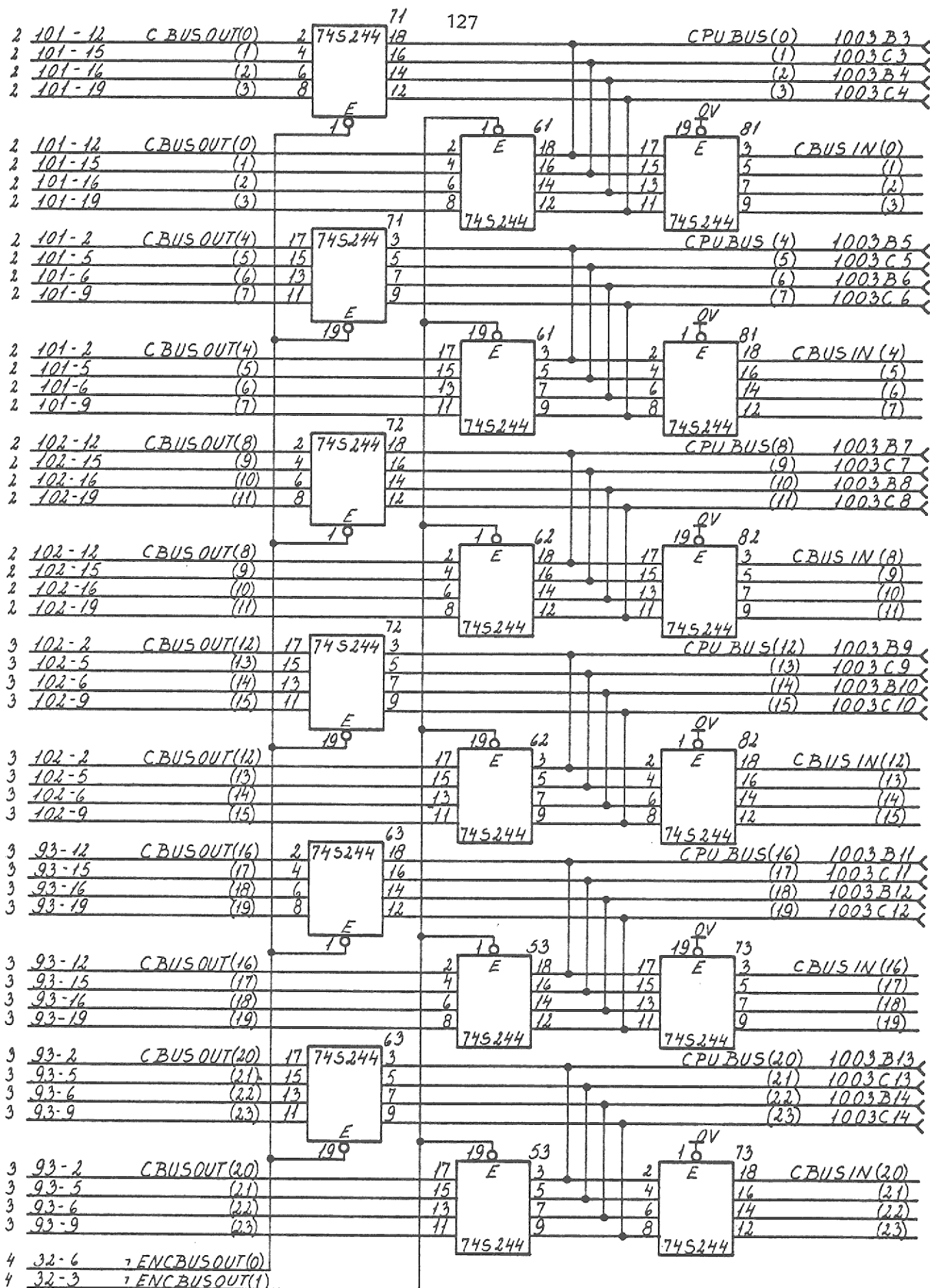
AAJ
800130



AGA

ARJ
800130

800318
RAJ
800901
ACR



17 101-12 DATA IN (0) 129 CBUS OUT (0)
25 51-2 RTC (0)

17 101-15 DATA IN (1) CBUS OUT (1)
25 51-5 RTC (1)

17 101-16 DATA IN (2) CBUS OUT (2)
25 51-6 RTC (2)

17 101-19 DATA IN (3) CBUS OUT (3)
25 51-9 RTC (3)

17 101-2 DATA IN (4) CBUS OUT (4)
25 51-12 RTC (4)
19 91-18 IO STAT (4)

17 101-5 DATA IN (5) CBUS OUT (5)
25 51-15 RTC (5)
19 91-16 IO STAT (5)

17 101-6 DATA IN (6) CBUS OUT (6)
25 51-16 RTC (6)
19 91-14 IO STAT (6)

17 101-9 DATA IN (7) CBUS OUT (7)
25 51-19 RTC (7)
19 91-12 IO STAT (7)

17 102-12 DATA IN (8) CBUS OUT (8)
25 52-2 RTC (8)
19 91-3 IO STAT (8)

17 102-15 DATA IN (9) CBUS OUT (9)
25 52-5 RTC (9)
19 91-5 IO STAT (9)

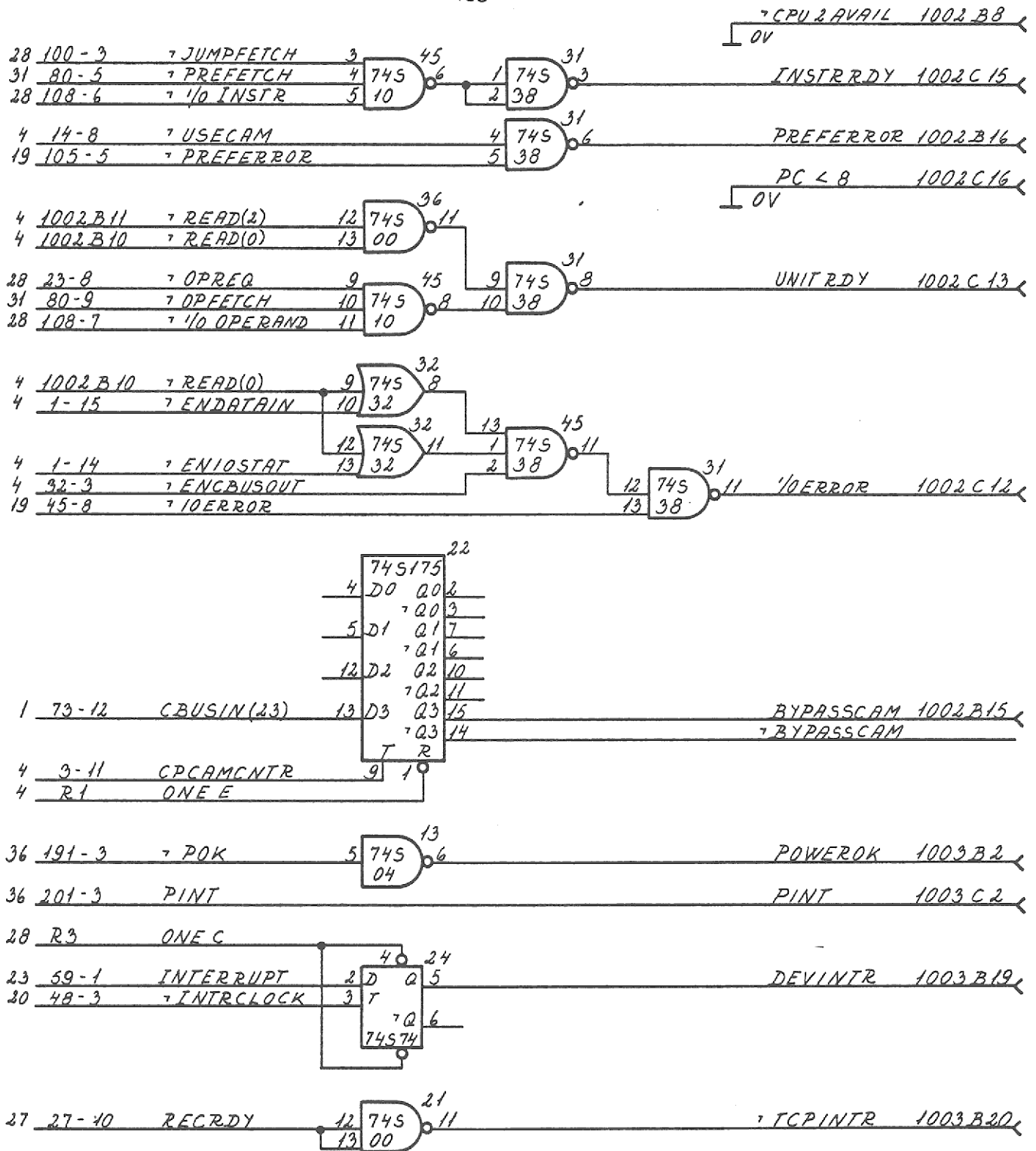
17 102-16 DATA IN (10) CBUS OUT (10)
25 52-6 RTC (10)
19 91-7 IO STAT (10)

17 102-19 DATA IN (11) CBUS OUT (11)
25 52-9 RTC (11)
19 91-9 IO STAT (11)

RAJ
8003 18
ACA
800901

17	102-2	DATA IN(12)	131	CBUS OUT(12)
25	52-12	RTC(12)		
19	92-18	I O STAT(12)		
17	102-5	DATA IN(13)		CBUS OUT(13)
25	52-15	RTC(13)		
19	92-16	I O STAT(13)		
17	102-6	DATA IN(14)		CBUS OUT(14)
25	52-16	RTC(14)		
19	92-14	I O STAT(14)		
17	102-9	DATA IN(15)		CBUS OUT(15)
25	52-19	RTC(15)		
19	92-12	I O STAT(15)		
17	93-12	DATA IN(16)		CBUS OUT(16)
25	43-2	RTC(16)		
23	83-2	INTR LEV(16)		
26	28-18	TC DAIN(16)		
26	38-2	TC STAT(16)		
17	93-15	DATA IN(17)		CBUS OUT(17)
25	43-5	RTC(17)		
23	83-5	INTR LEV(17)		
26	28-16	TC DAIN(17)		
26	38-5	TC STAT(17)		
17	93-16	DATA IN(18)		CBUS OUT(18)
25	43-6	RTC(18)		
23	83-6	INTR LEV(18)		
26	28-14	TC DAIN(18)		
26	38-6	TC STAT(18)		
17	93-19	DATA IN(19)		CBUS OUT(19)
25	43-9	RTC(19)		
23	83-9	INTR LEV(19)		
26	28-12	TC DAIN(19)		
26	38-9	TC STAT(19)		
17	93-2	DATA IN(20)		CBUS OUT(20)
25	43-12	RTC(20)		
19	92-3	I O STAT(20)		
23	83-12	INTR LEV(20)		
26	28-3	TC DAIN(20)		
26	38-12	TC STAT(20)		
17	93-5	DATA IN(21)		CBUS OUT(21)
25	43-15	RTC(21)		
19	92-5	I O STAT(21)		
23	83-15	INTR LEV(21)		
26	28-5	TC DAIN(21)		
26	38-15	TC STAT(21)		
17	93-6	DATA IN(22)		CBUS OUT(22)
25	43-16	RTC(22)		
19	92-7	I O STAT(22)		
23	83-16	INTR LEV(22)		
26	28-7	TC DAIN(22)		
26	38-16	TC STAT(22)		
17	93-9	DATA IN(23)		CBUS OUT(23)
25	43-19	RTC(23)		
19	92-9	I O STAT(23)		
23	83-19	INTR LEV(23)		
26	28-9	TC DAIN(23)		
26	38-19	TC STAT(23)		

ACA
800901
RAJ
800318



AGA
RAJ
800318 800901

1	81-3	CBUS IN(0)	3	D0	Q0	2	155	
1	81-5	(1)	4	D1	Q1	5	137	LOG ADDR(0)
1	81-7	(2)	7	D2	Q2	6		(1)
1	81-9	(3)	8	D3	Q3	9		(2)
1	81-18	(4)	13	D4	Q4	12		(3)
1	81-16	(5)	14	D5	Q5	15		(4)
1	81-14	(6)	17	D6	Q6	16		(5)
1	81-12	(7)	18	D7	Q7	19		(6)
				T	E			(7)

1	82-3	CBUS IN(8)	3	D0	Q0	2	157	
1	82-5	(9)	4	D1	Q1	5		LOG ADDR(8)
1	82-7	(10)	7	D2	Q2	6		(9)
1	82-9	(11)	8	D3	Q3	9		(10)
1	82-18	(12)	13	D4	Q4	12		(11)
1	82-16	(13)	14	D5	Q5	15		(12)
1	82-14	(14)	17	D6	Q6	16		(13)
1	82-12	(15)	18	D7	Q7	19		(14)
				T	E			(15)

1	73-3	CBUS IN(16)	3	D0	Q0	2	159	
1	73-5	(17)	4	D1	Q1	5		LOG ADDR(16)
1	73-7	(18)	7	D2	Q2	6		(17)
1	73-9	(19)	8	D3	Q3	9		(18)
1	73-18	(20)	13	D4	Q4	12		(19)
1	73-16	(21)	14	D5	Q5	15		(20)
1	73-14	(22)	17	D6	Q6	16		(21)
1	73-12	(23)	18	D7	Q7	19		(22)
				T	E			(23)

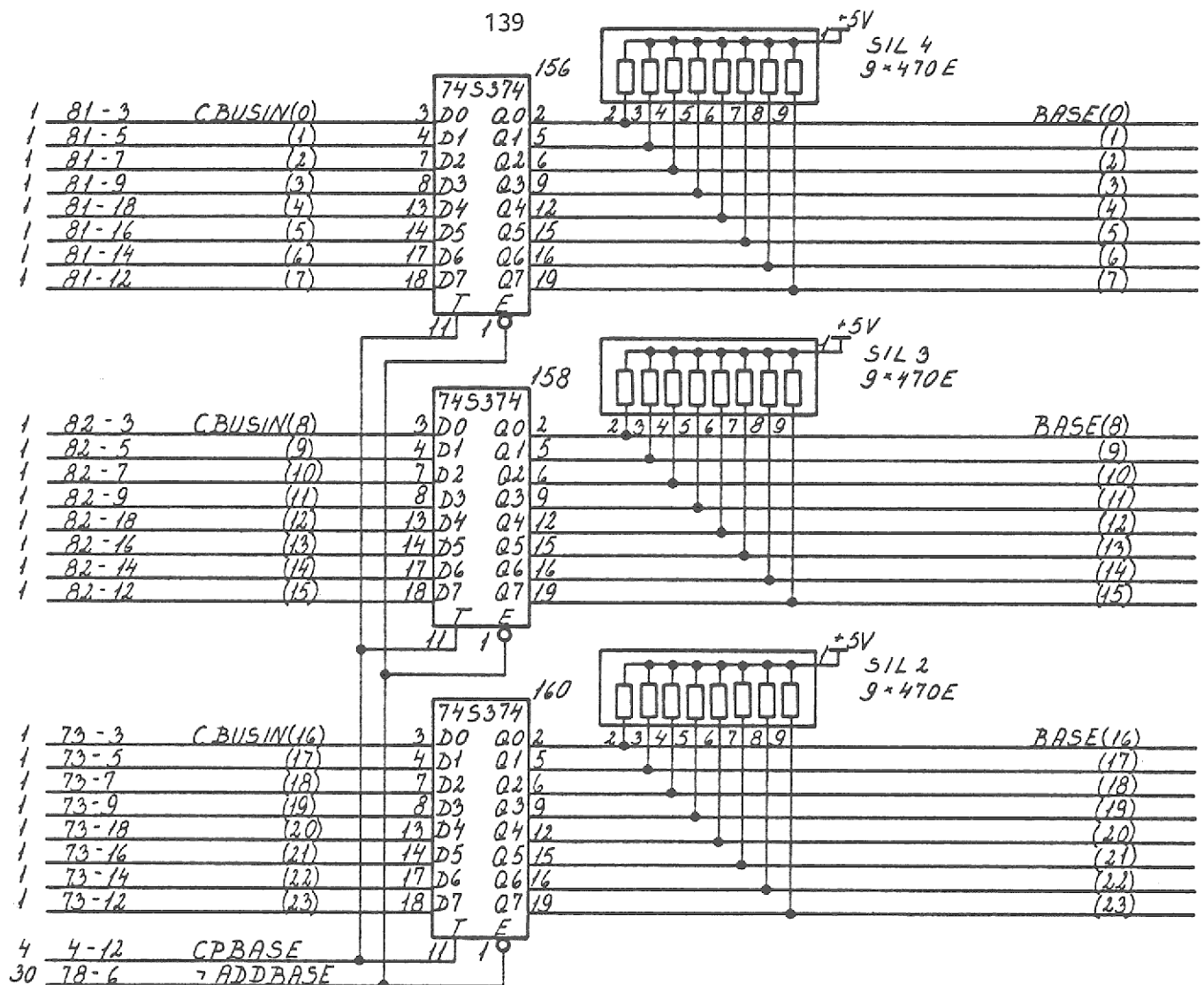
4	4-14	CP LOG ADDR	11	T	E		145	
1	81-3	CBUS IN(0)	3	D0	Q0	2		LLIM(0)
1	81-5	(1)	4	D1	Q1	5		(1)
1	81-7	(2)	7	D2	Q2	6		(2)
1	81-9	(3)	8	D3	Q3	9		(3)
1	81-18	(4)	13	D4	Q4	12		(4)
1	81-16	(5)	14	D5	Q5	15		(5)
1	81-14	(6)	17	D6	Q6	16		(6)
1	81-12	(7)	18	D7	Q7	19		(7)
				T	E			

1	82-3	CBUS IN(8)	3	D0	Q0	2	147	
1	82-5	(9)	4	D1	Q1	5		LLIM(8)
1	82-7	(10)	7	D2	Q2	6		(9)
1	82-9	(11)	8	D3	Q3	9		(10)
1	82-18	(12)	13	D4	Q4	12		(11)
1	82-16	(13)	14	D5	Q5	15		(12)
1	82-14	(14)	17	D6	Q6	16		(13)
1	82-12	(15)	18	D7	Q7	19		(14)
				T	E			(15)

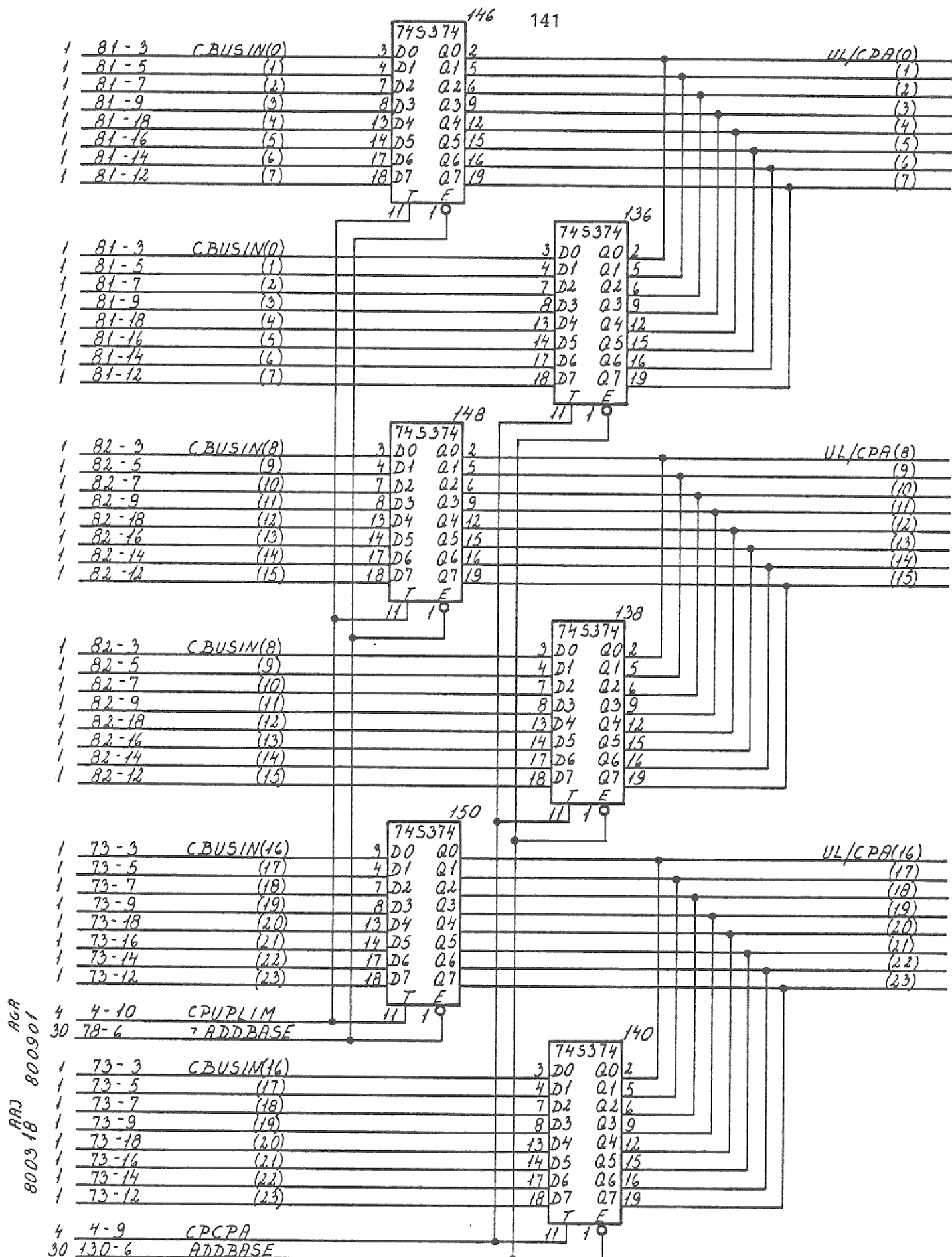
1	73-3	CBUS IN(16)	3	D0	Q0	2	149	
1	73-5	(17)	4	D1	Q1	5		LLIM(16)
1	73-7	(18)	7	D2	Q2	6		(17)
1	73-9	(19)	8	D3	Q3	9		(18)
1	73-18	(20)	13	D4	Q4	12		(19)
1	73-16	(21)	14	D5	Q5	15		(20)
1	73-14	(22)	17	D6	Q6	16		(21)
1	73-12	(23)	18	D7	Q7	19		(22)
				T	E			(23)

4	4-11	CP LLIM	11	T	E		0V	
---	------	---------	----	---	---	--	----	--

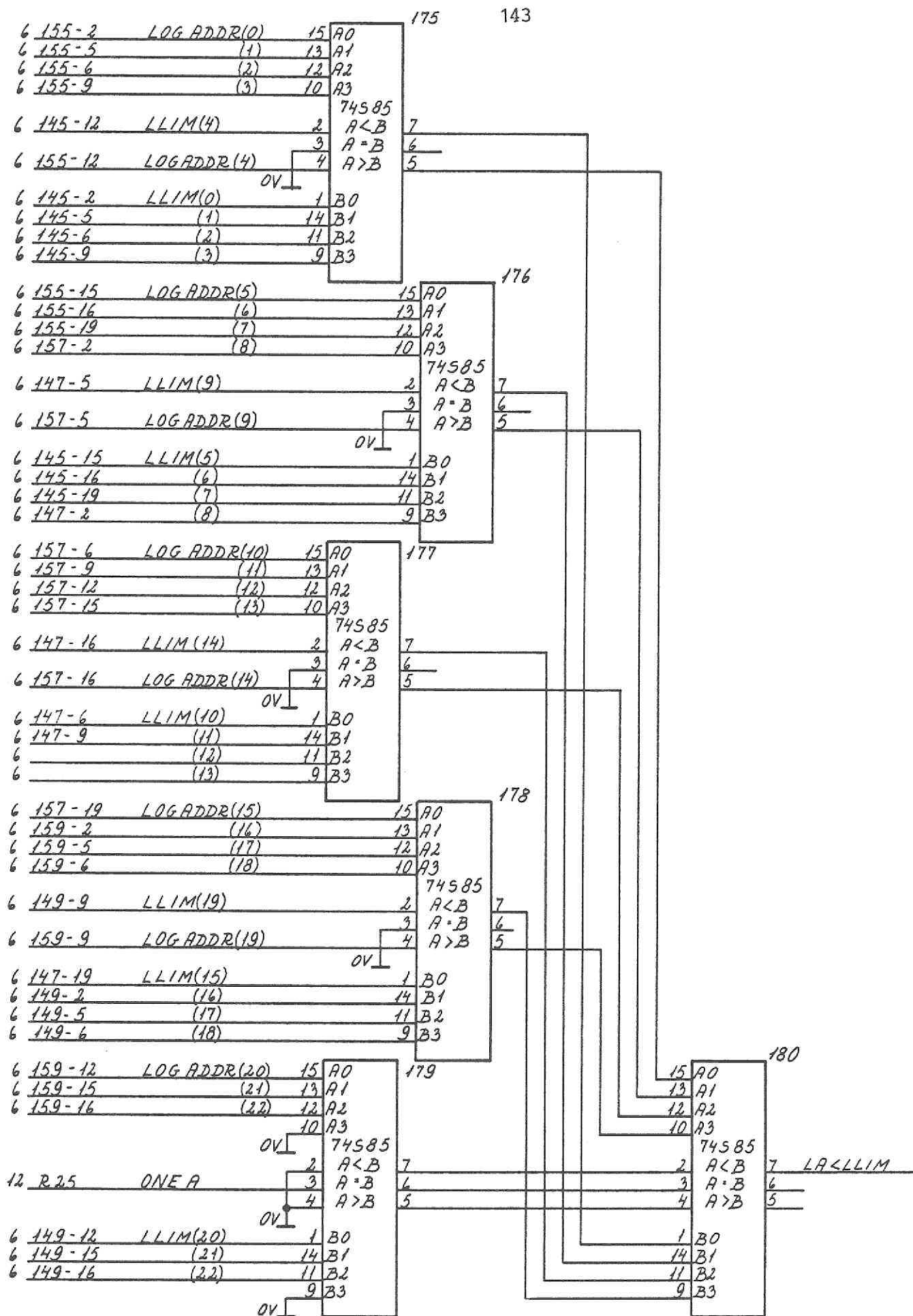
AGA
800318
800901



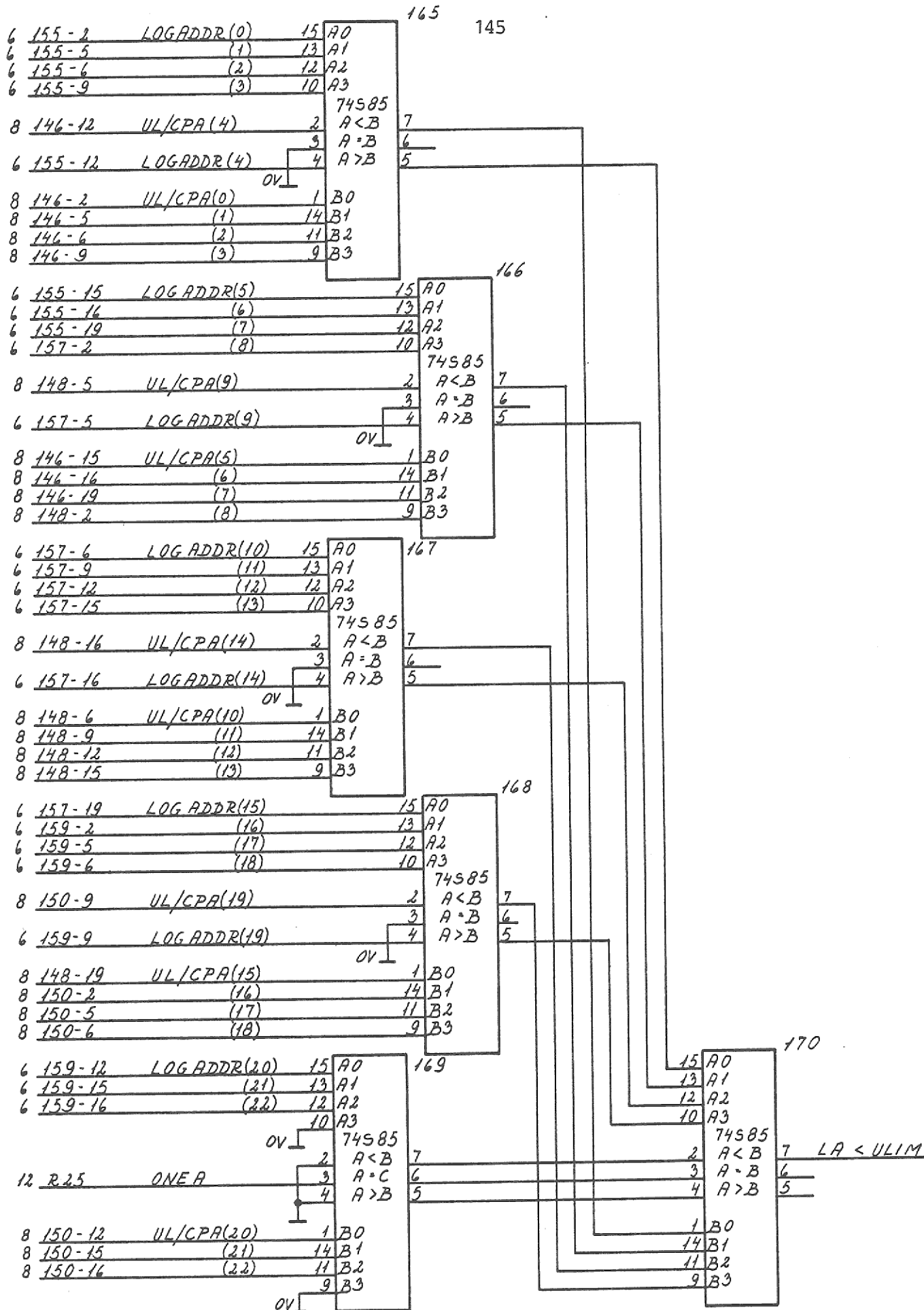
800318
RRJ
ACA
800901

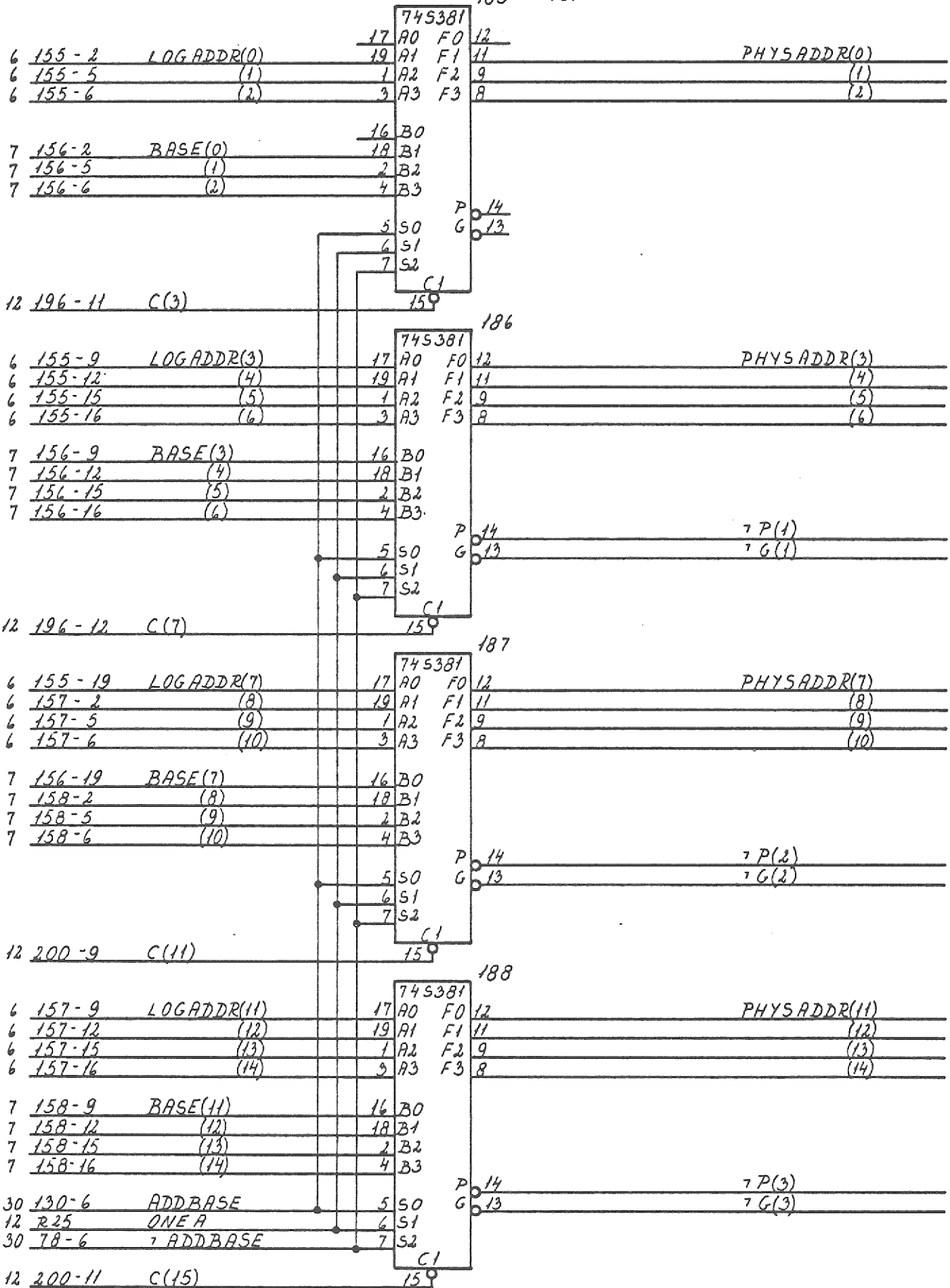


AGA
800318
RAJ
800901

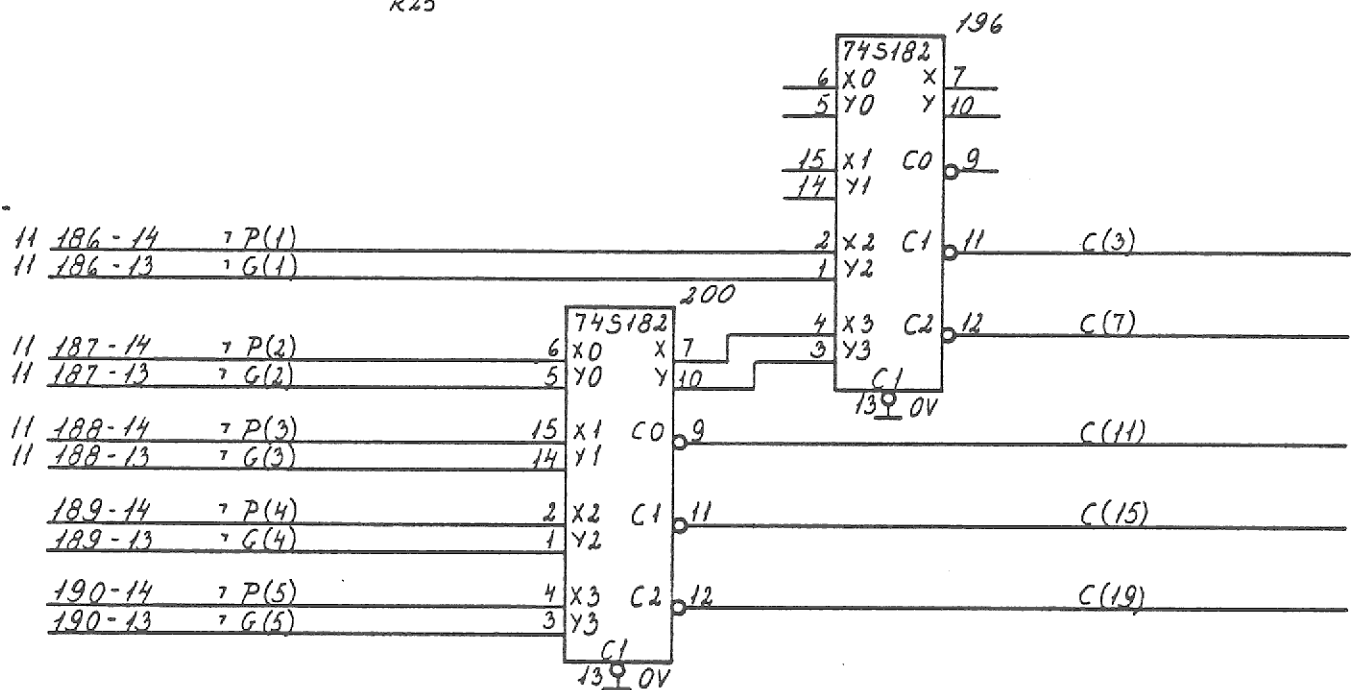
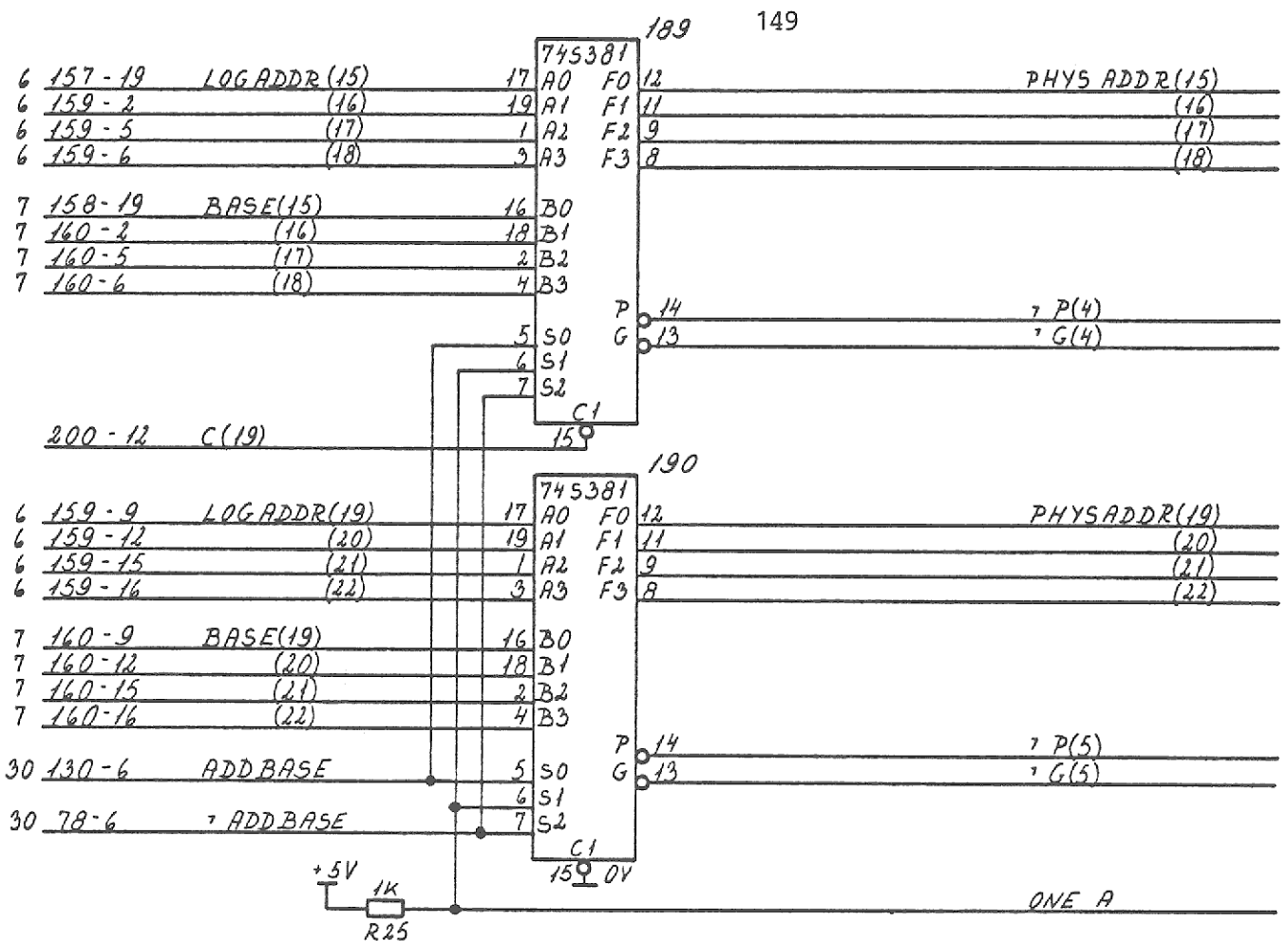


AGA
RAJ 800901
800518



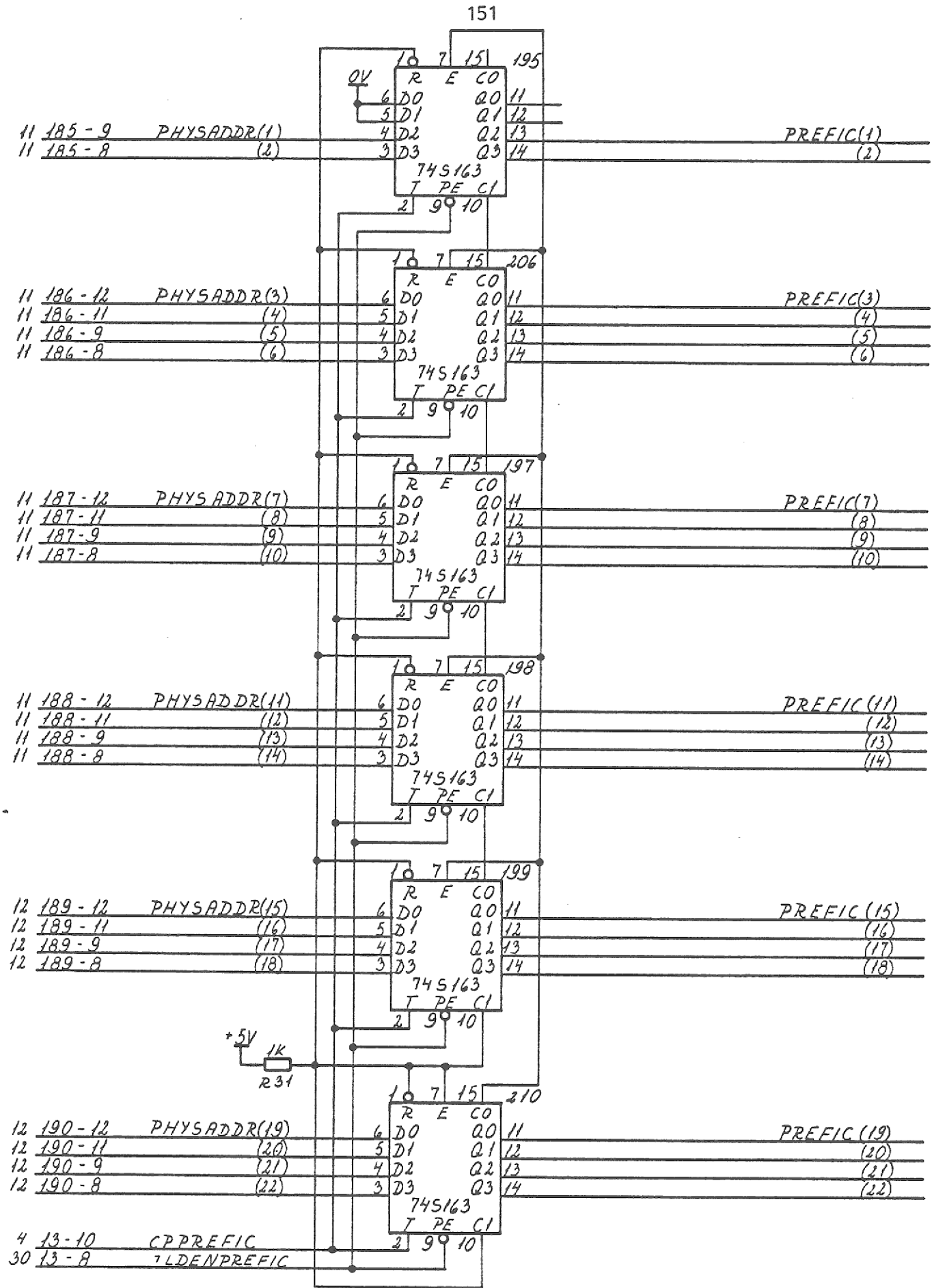


ARJ
800318
800901
AGA



AAJ AGA
800318 800901

ARJ
800318
RCA
800901



14	217-7	10 ADDR(4)	3	D0	Q0	2	SB ADDR(4)
14	217-9	(5)	4	D1	Q1	5	(5)
14	217-12	(6)	7	D2	Q2	6	(6)
14	207-4	(7)	8	D3	Q3	9	(7)
14	216-7	10 ADDR(0)	13	D4	Q4	12	SB ADDR(0)
14	216-9	(1)	14	D5	Q5	15	(1)
14	216-12	(2)	17	D6	Q6	16	(2)
14	217-4	(3)	18	D7	Q7	19	(3)

G E

11 1 0V

14	208-7	10 ADDR(12)	3	D0	Q0	2	SB ADDR(12)
14	208-9	(13)	4	D1	Q1	5	(13)
14	208-12	(14)	7	D2	Q2	6	(14)
14	209-4	(15)	8	D3	Q3	9	(15)
14	207-7	10 ADDR(8)	13	D4	Q4	12	SB ADDR(8)
14	207-9	(9)	14	D5	Q5	15	(9)
14	207-12	(10)	17	D6	Q6	16	(10)
14	208-4	(11)	18	D7	Q7	19	(11)

G E

11 1 0V

14	220-7	10 ADDR(20)	3	D0	Q0	2	SB ADDR(20)
14	220-9	(21)	4	D1	Q1	5	(21)
14	220-12	(22)	7	D2	Q2	6	(22)
			8	D3	Q3	9	
14	209-7	10 ADDR(16)	13	D4	Q4	12	SB ADDR(16)
14	209-9	(17)	14	D5	Q5	15	(17)
14	209-12	(18)	17	D6	Q6	16	(18)
14	220-4	(19)	18	D7	Q7	19	(19)

G E

11 1 0V

184-2	SB ADDR(4)	1	183
184-5	(5)	2	
184-6	(6)	4	
		8	
184-9	(7)	9	E 5
184-12	(10)	10	0 6
184-15	(11)	11	
184-16	(12)	12	
184-19	(13)	13	

0V 74S280

174-2	SB ADDR(12)	1	173
174-5	(13)	2	
174-6	(14)	4	
		8	
174-9	SB ADDR(15)	9	E 5
174-12	(8)	10	0 6
174-15	(9)	11	
174-16	(10)	12	
174-19	(11)	13	

0V 74S280

164-2	SB ADDR(20)	1	163
164-5	(21)	2	
164-6	(22)	4	
		8	
164-9	SB ADDR(15)	9	E 5
164-12	(16)	10	0 6
164-15	(17)	11	
164-16	(18)	12	
164-19	(19)	13	

74S280

IO ADDR PAR

AAJ 800318 AGA 800901

1	81-18	CBUSIN(4)	3	D0	Q0	2	DATAOUT(4)
1	81-16	(5)	4	D1	Q1	5	(5)
1	81-14	(6)	7	D2	Q2	6	(6)
1	81-12	(7)	8	D3	Q3	9	(7)
1	81-3	CBUSIN(0)	13	D4	Q4	12	DATAOUT(0)
1	81-5	(1)	14	D5	Q5	15	(1)
1	81-7	(2)	17	D6	Q6	16	(2)
1	81-9	(3)	18	D7	Q7	19	(3)

T E

11 1 0V

1	82-18	CBUSIN(12)	3	D0	Q0	2	DATAOUT(12)
1	82-16	(13)	4	D1	Q1	5	(13)
1	82-14	(14)	7	D2	Q2	6	(14)
1	82-12	(15)	8	D3	Q3	9	(15)
1	82-3	CBUSIN(8)	13	D4	Q4	12	DATAOUT(8)
1	82-5	(9)	14	D5	Q5	15	(9)
1	82-7	(10)	17	D6	Q6	16	(10)
1	82-9	(11)	18	D7	Q7	19	(11)

T E

11 1 0V

1	73-18	CBUSIN(20)	3	D0	Q0	2	DATAOUT(20)
1	73-16	(21)	4	D1	Q1	5	(21)
1	73-14	(22)	7	D2	Q2	6	(22)
1	73-12	(23)	8	D3	Q3	9	(23)
1	73-3	CBUSIN(16)	13	D4	Q4	12	DATAOUT(16)
1	73-5	(17)	14	D5	Q5	15	(17)
1	73-7	(18)	17	D6	Q6	16	(18)
1	73-9	(19)	18	D7	Q7	19	(19)

T E

11 1 0V

154-2	DATAOUT(4)	1	153
154-5	(5)	2	
154-6	(6)	4	
154-9	(7)	8	E 5
154-12	DATAOUT(0)	9	
154-15	(1)	10	0 6
154-16	(2)	11	
154-19	(3)	12	

13 0V 74S280

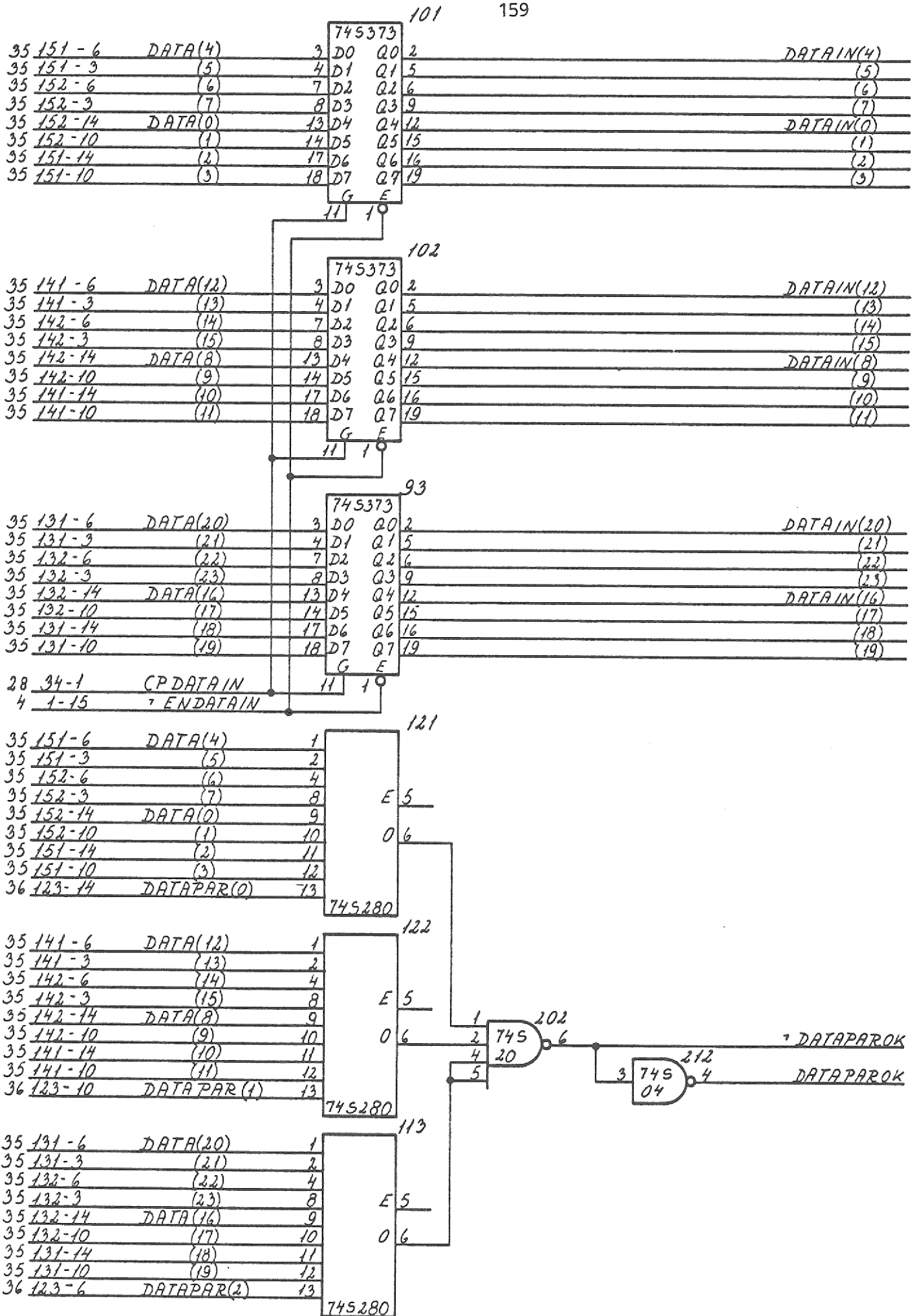
144-2	DATAOUT(12)	1	143
144-5	(13)	2	
144-6	(14)	4	
144-9	(15)	8	E 5
144-12	DATAOUT(8)	9	
144-15	(9)	10	0 6
144-16	(10)	11	
144-19	(11)	12	

13 0V 74S280

134-2	DATAOUT(20)	1	133
134-5	(21)	2	
134-6	(22)	4	
134-9	(23)	8	E 5
134-12	DATAOUT(16)	9	
134-15	(17)	10	0 6
134-16	(18)	11	
134-19	(19)	12	

13 0V 74S280

RAJ 800318 AGA 800901



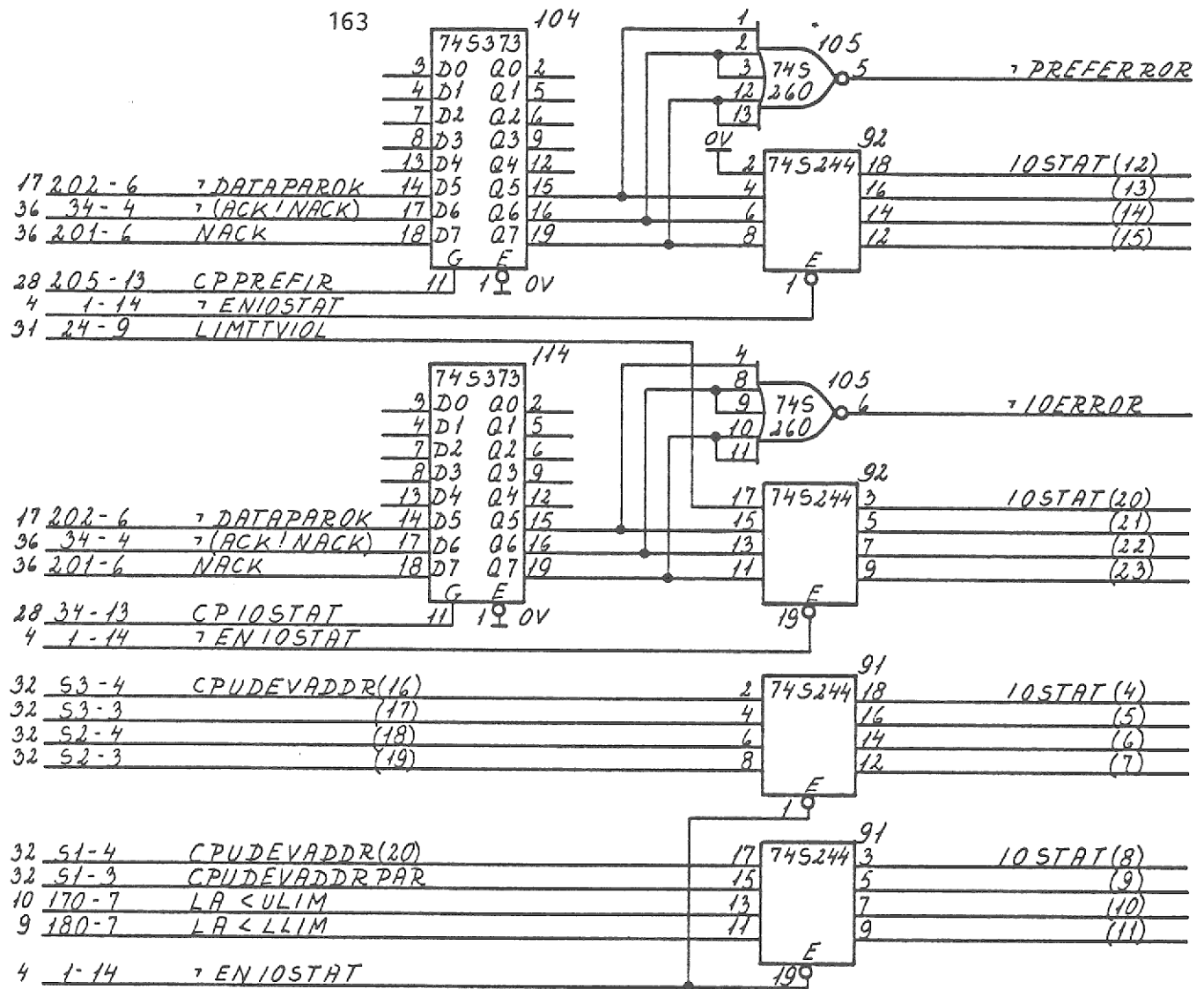
35	152-14	DATA(0)	3	D0	Q0	2	INSTRBUS(0)	1002 B 17
35	152-10	(1)	4	D1	Q1	5	(1)	1002 C 17
35	151-14	(2)	7	D2	Q2	6	(2)	1002 B 18
35	151-10	(3)	8	D3	Q3	9	(3)	1002 C 18
35	151-6	(4)	13	D4	Q4	12	(4)	1002 B 19
35	151-3	(5)	14	D5	Q5	15	(5)	1002 C 19
35	152-6	(6)	17	D6	Q6	16	(6)	1002 B 20
35	152-3	(7)	18	D7	Q7	19	(7)	1002 C 20
			G		E			

35	142-14	DATA(8)	3	D0	Q0	2	INSTRBUS(8)	1002 B 21
35	142-10	(9)	4	D1	Q1	5	(9)	1002 C 21
35	141-14	(10)	7	D2	Q2	6	(10)	1002 B 22
35	141-10	(11)	8	D3	Q3	9	(11)	1002 C 22
35	141-6	(12)	13	D4	Q4	12	(12)	1002 B 23
35	141-3	(13)	14	D5	Q5	15	(13)	1002 C 23
35	142-6	(14)	17	D6	Q6	16	(14)	1002 B 24
35	142-3	(15)	18	D7	Q7	19	(15)	1002 C 24
			G		E			

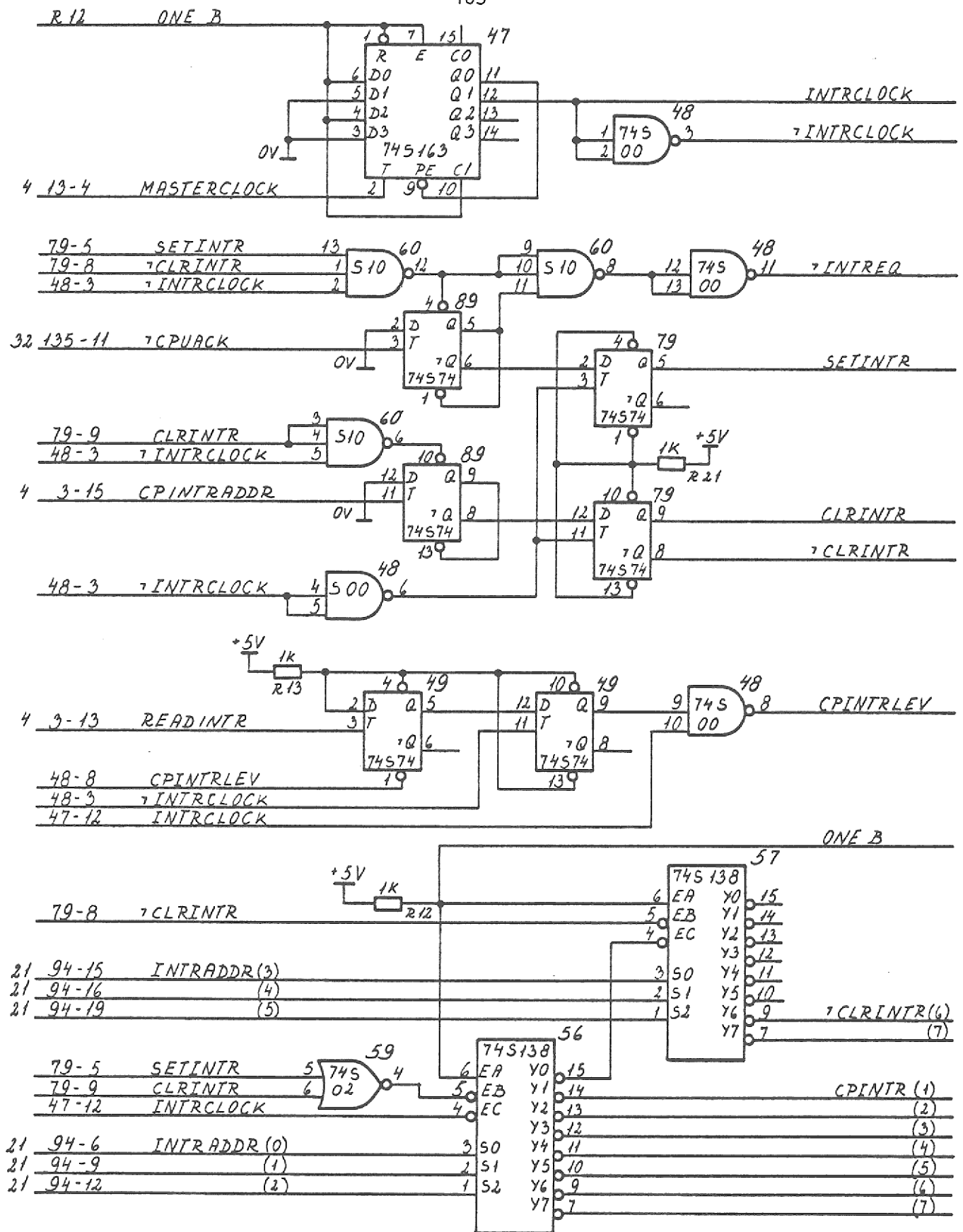
35	132-14	DATA(16)	3	D0	Q0	2	INSTRBUS(16)	1002 B 25
35	132-10	(17)	4	D1	Q1	5	(17)	1002 C 25
35	131-14	(18)	7	D2	Q2	6	(18)	1002 B 26
35	131-10	(19)	8	D3	Q3	9	(19)	1002 C 26
35	131-6	(20)	13	D4	Q4	12	(20)	1002 B 27
35	131-3	(21)	14	D5	Q5	15	(21)	1002 C 27
35	132-6	(22)	17	D6	Q6	16	(22)	1002 B 28
35	132-3	(23)	18	D7	Q7	19	(23)	1002 C 28
			G		E			

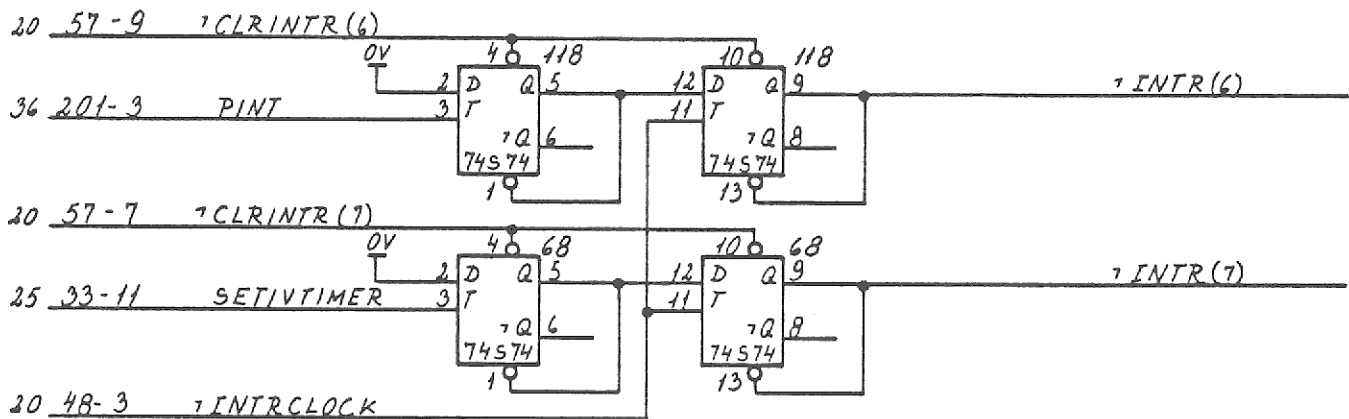
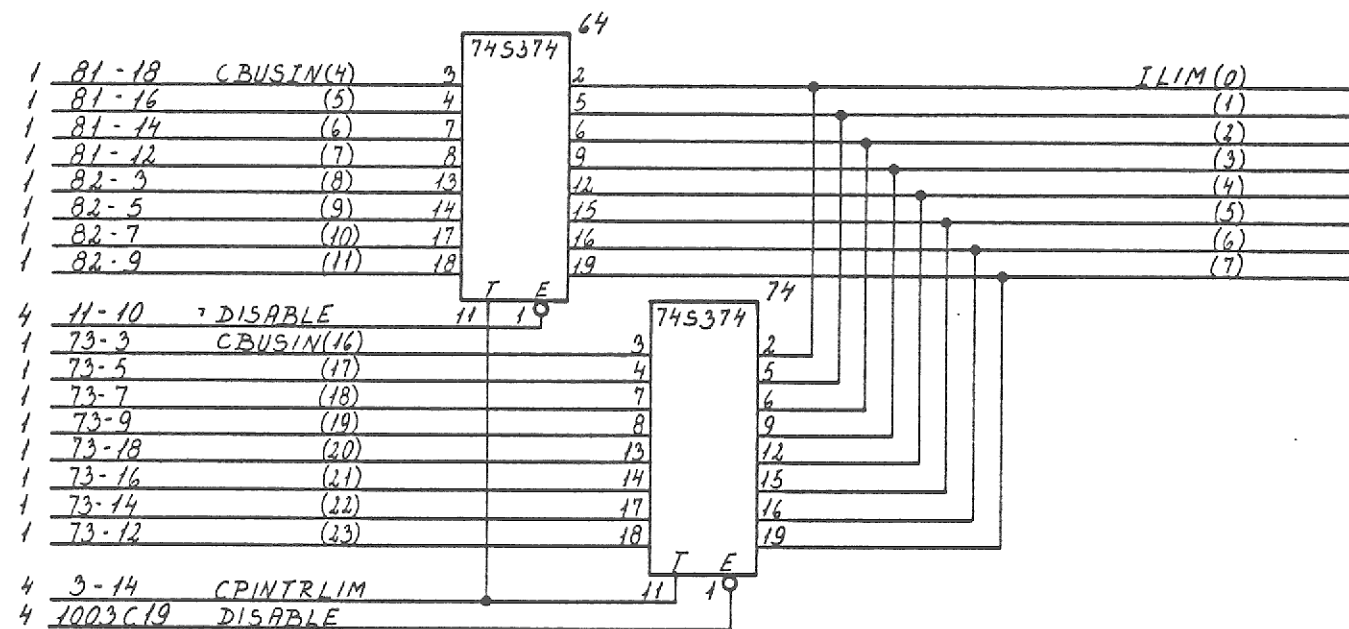
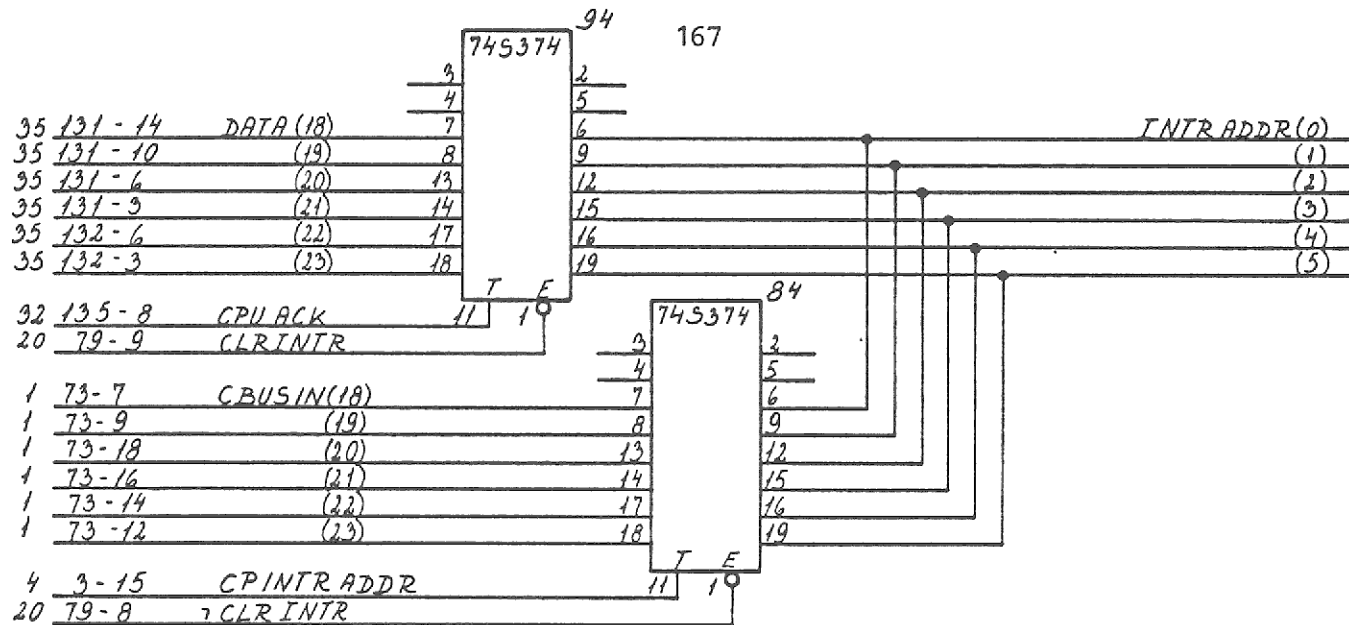
28 205-13 CPPREFIR
4 14-11 USECAM

800518
ARJ
800901
AGA

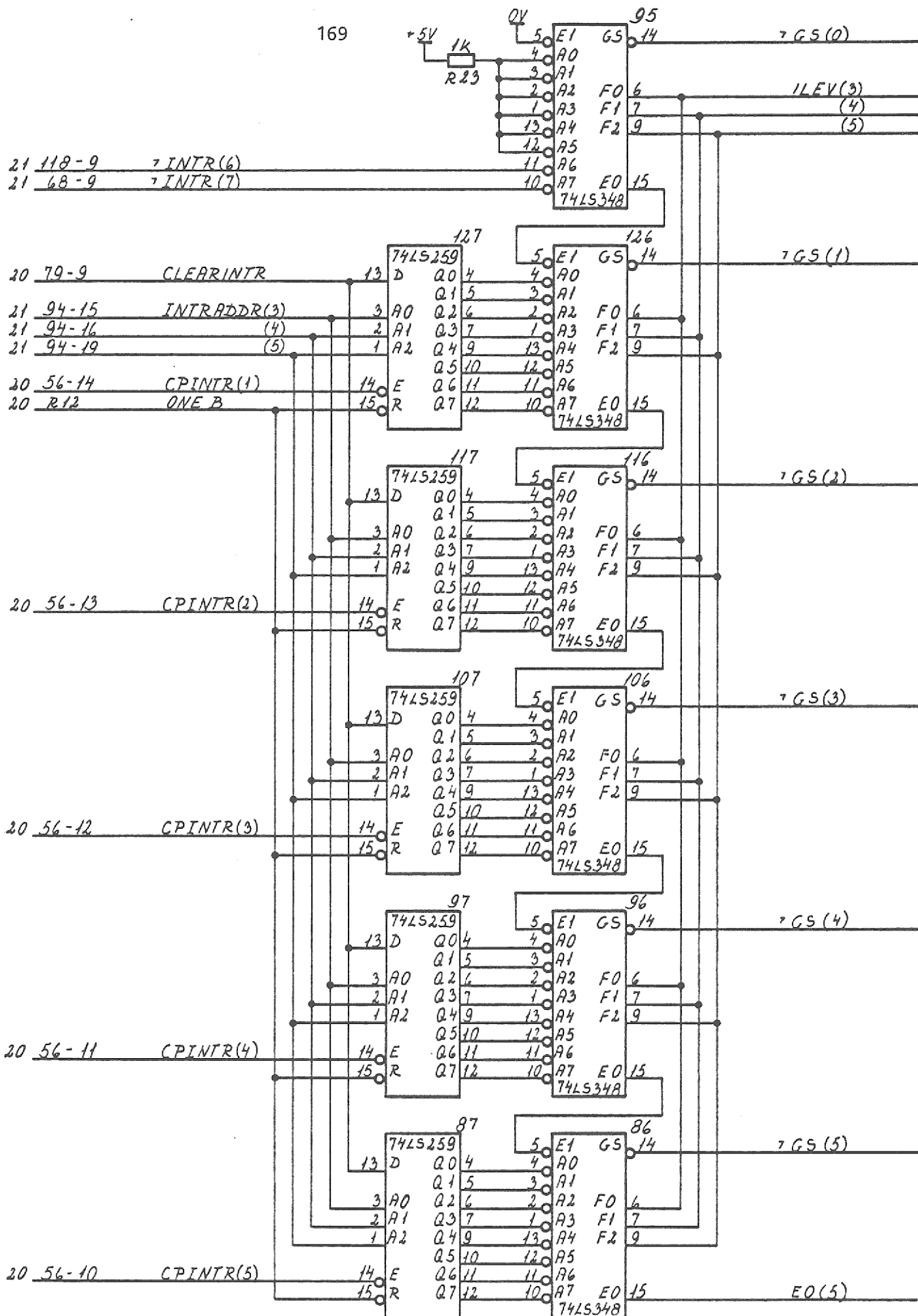


ARJ AGA
800318 800901

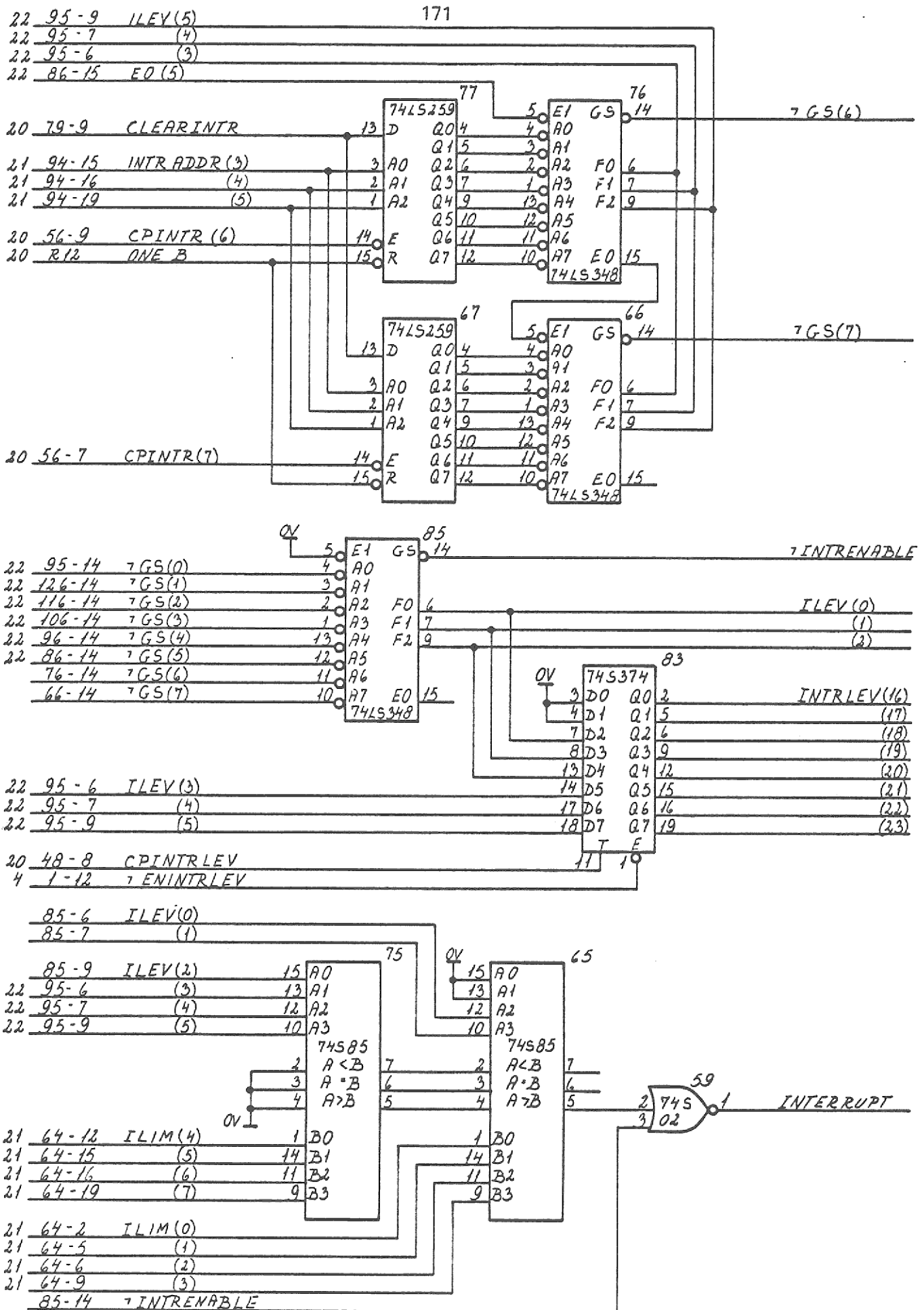


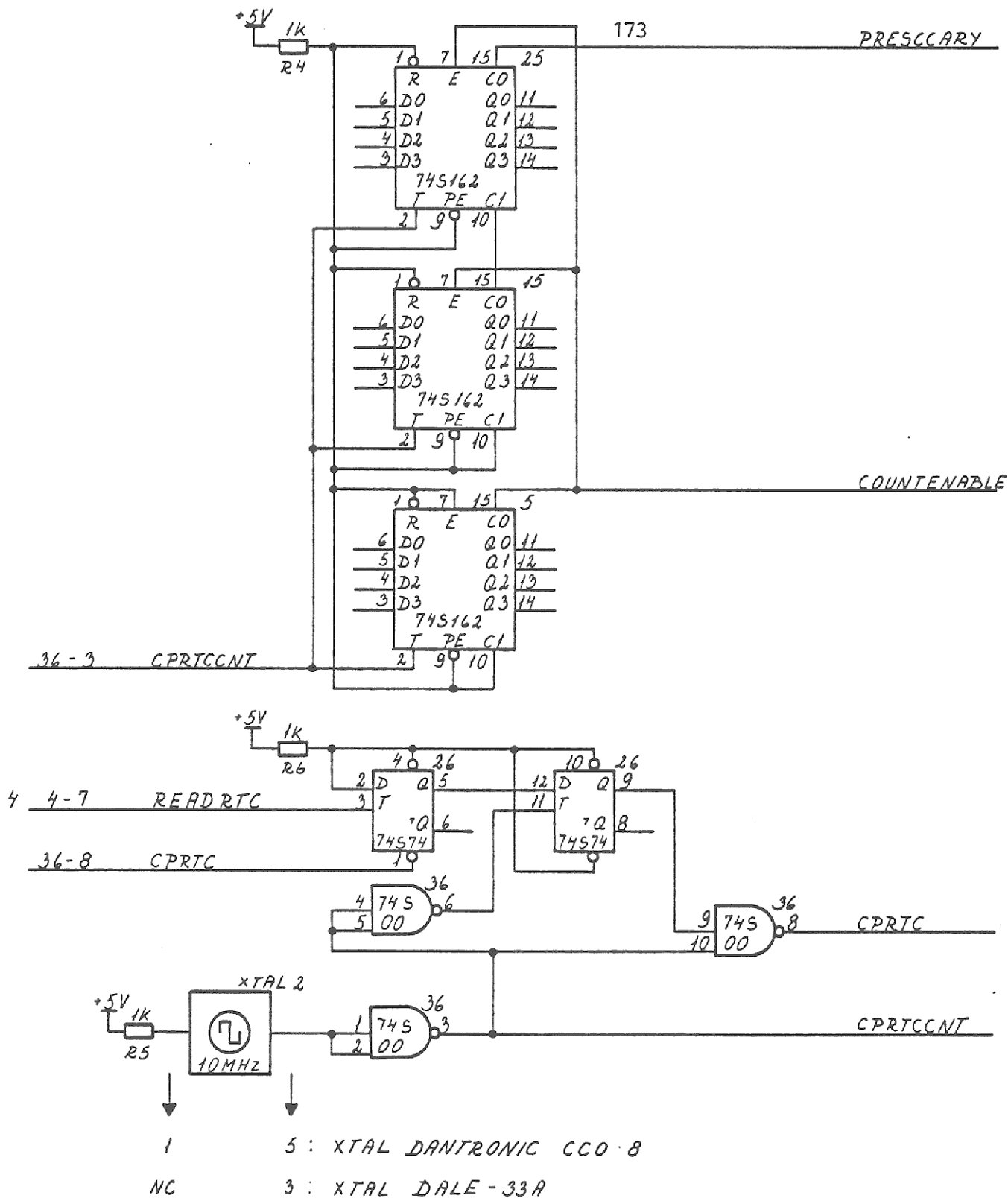


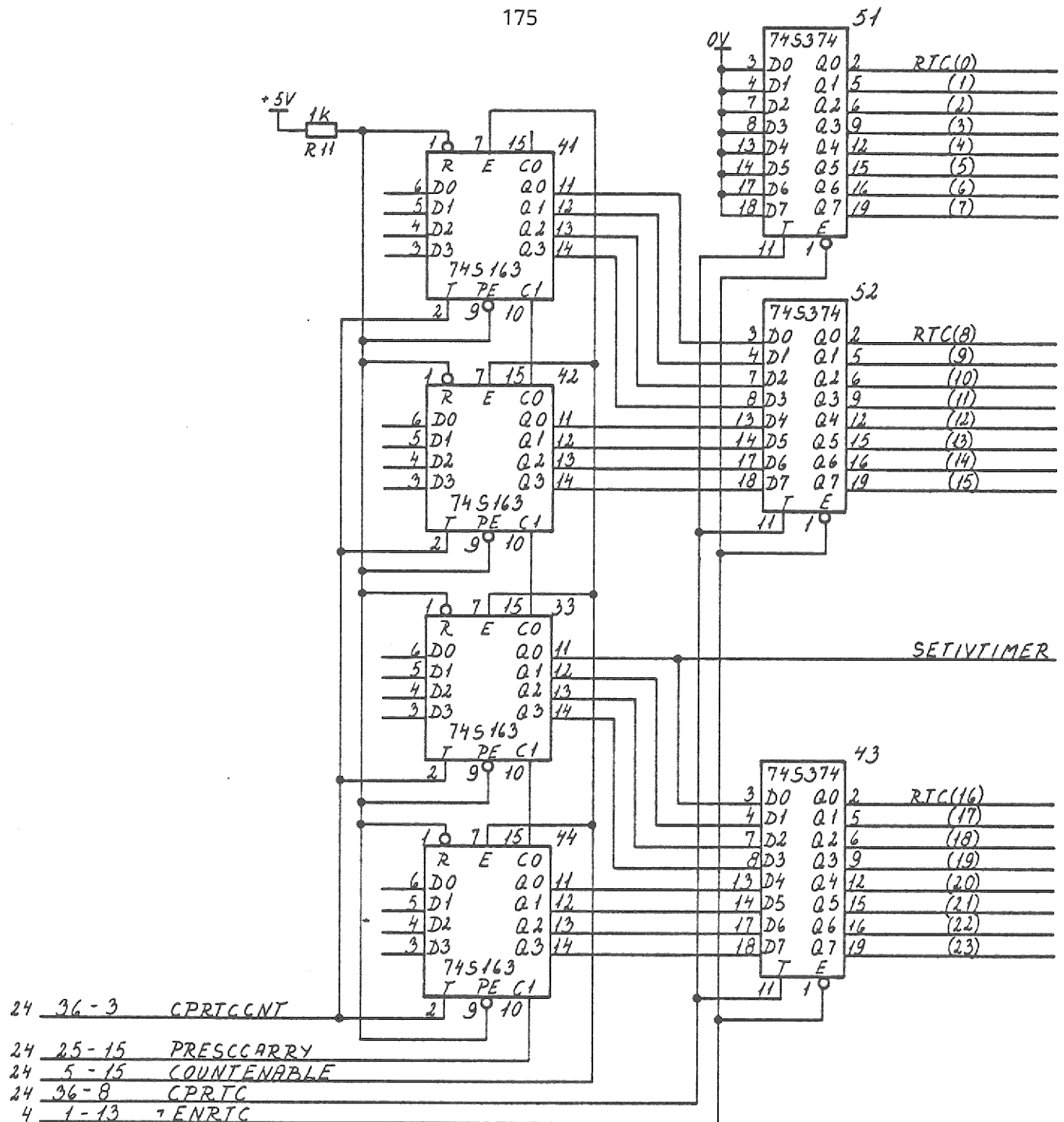
AGA
800901
RAJ
800318



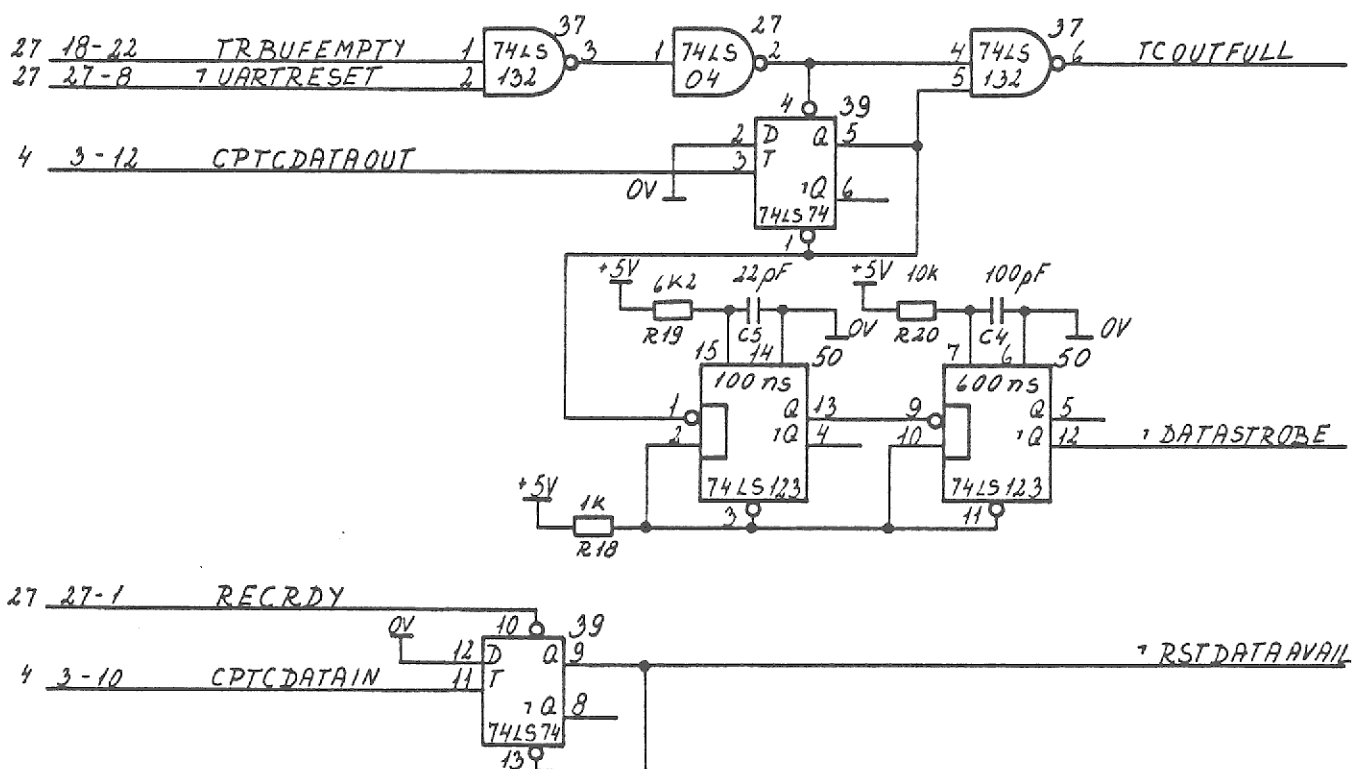
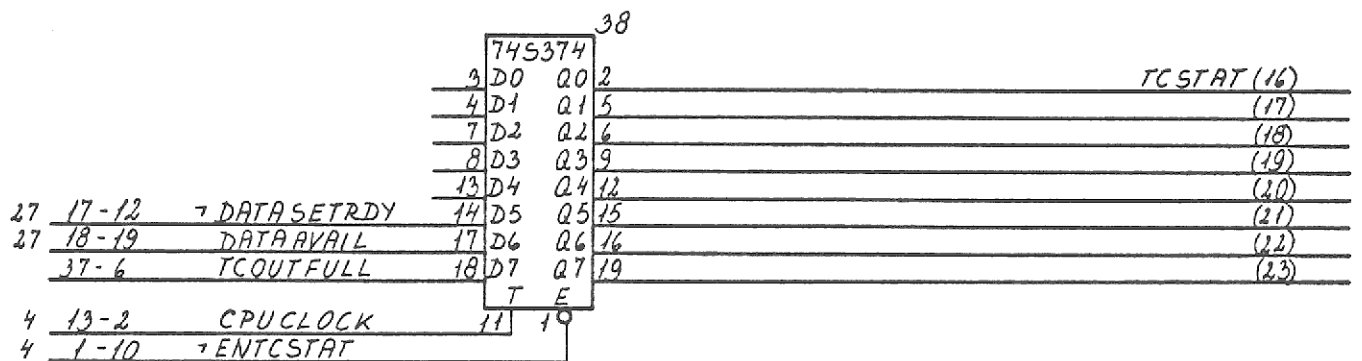
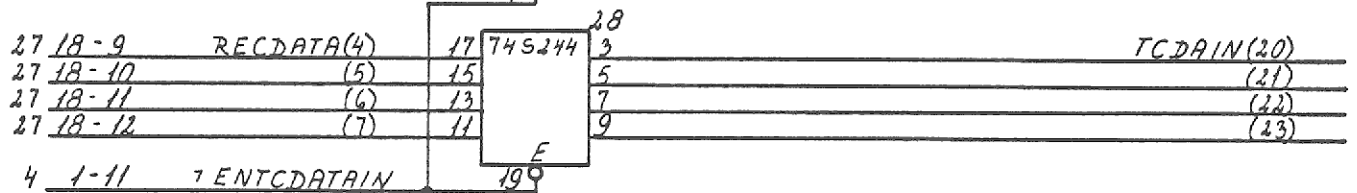
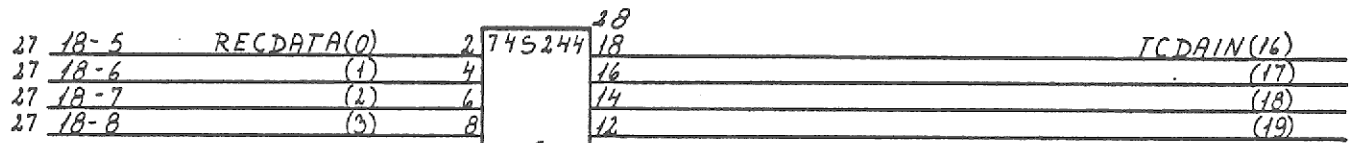
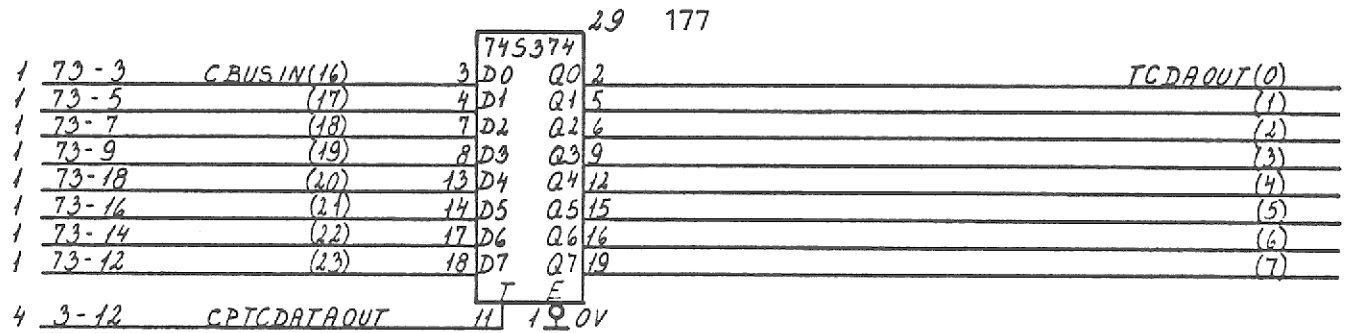
AGA
800901
ARJ
800318

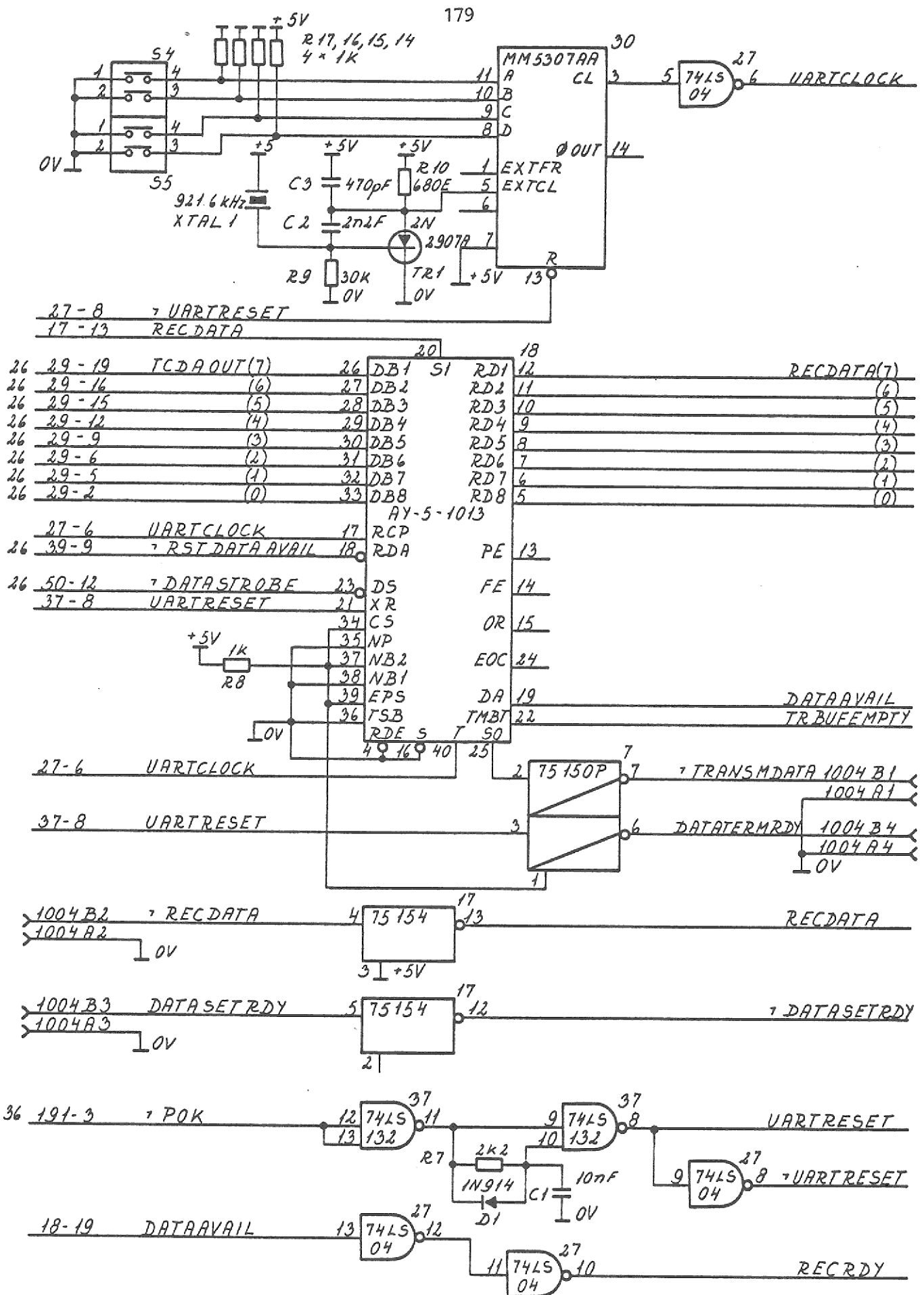


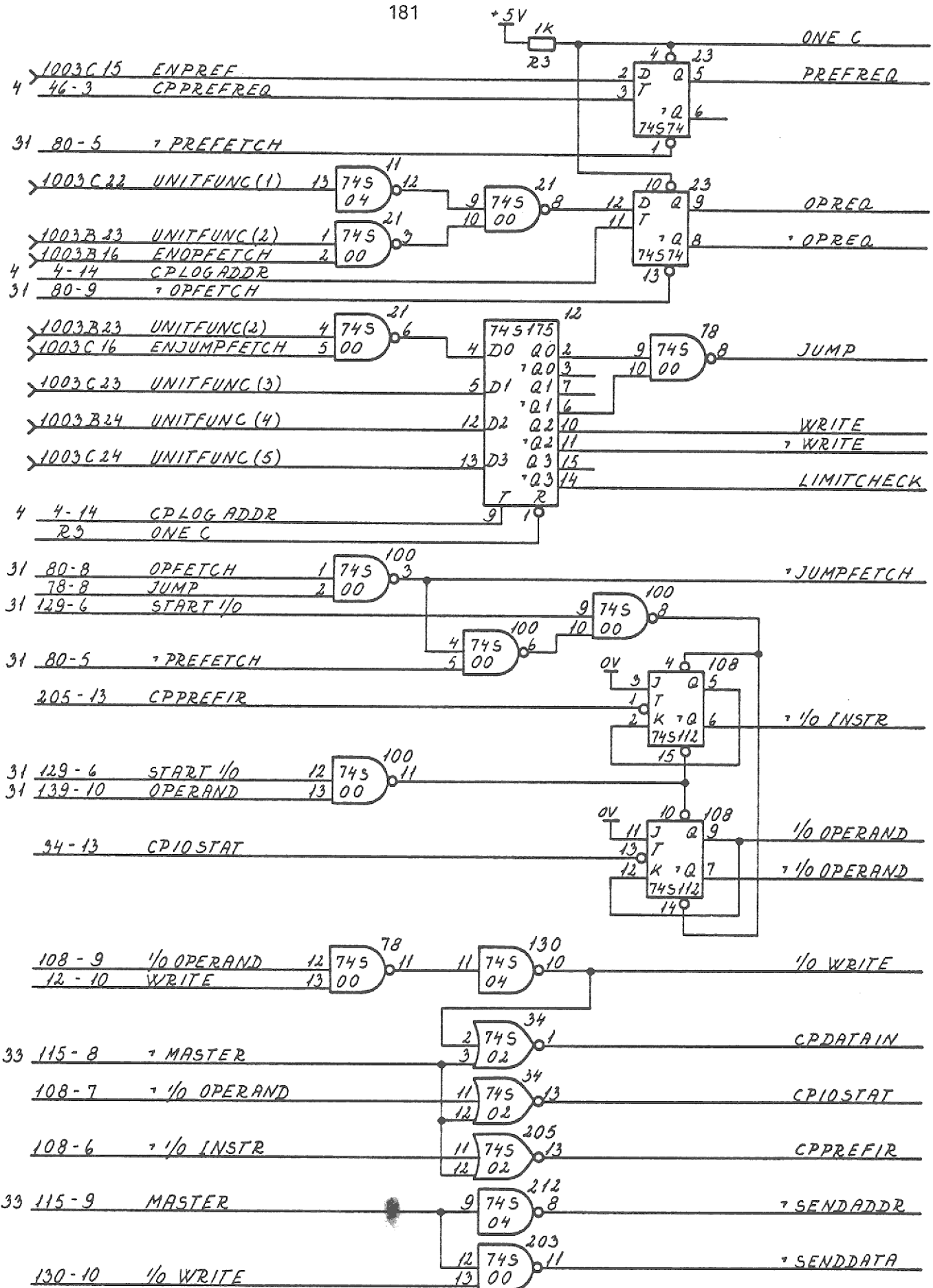


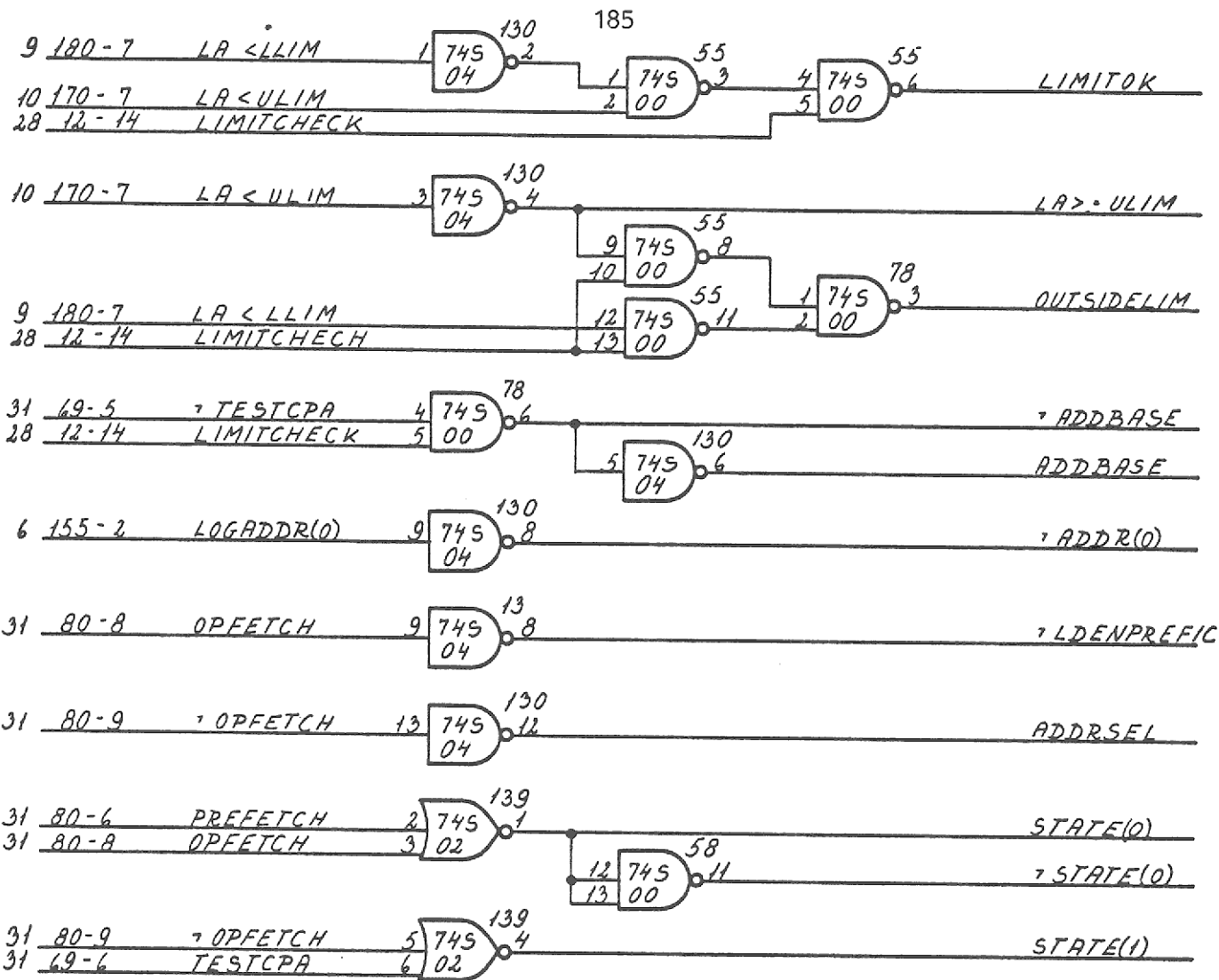


800318
RAJ
800901
AGA

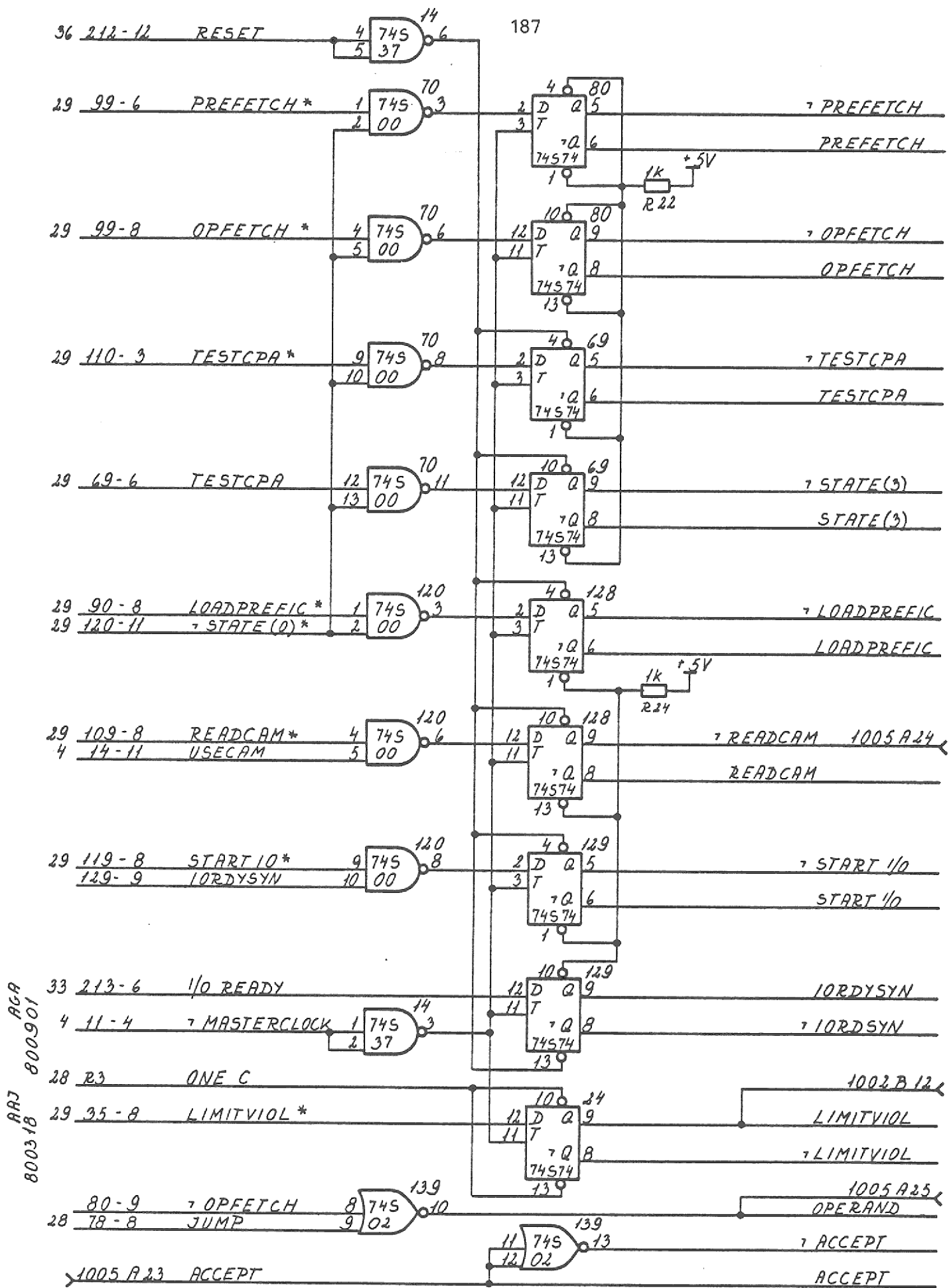




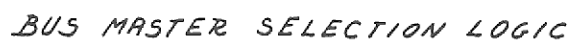


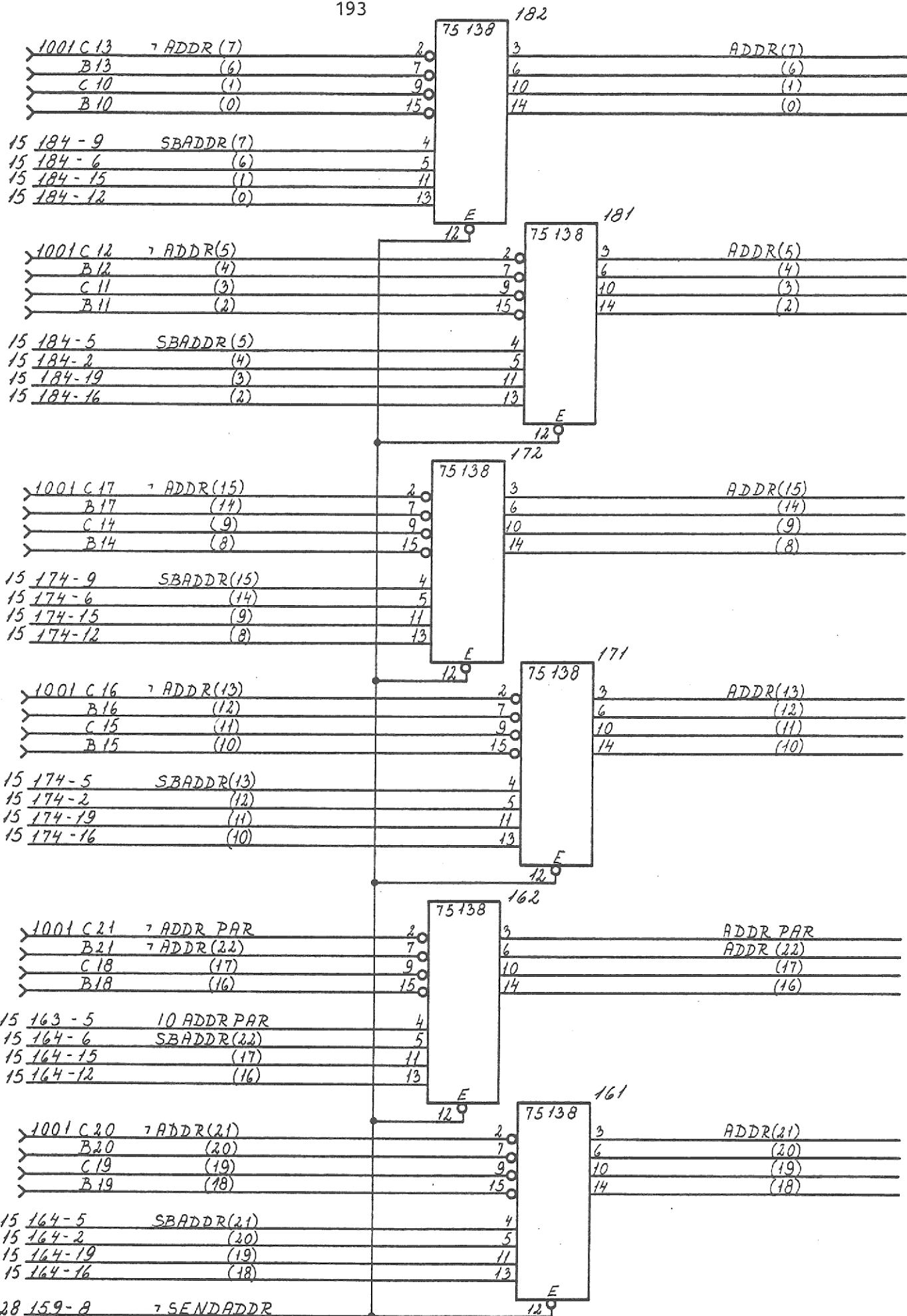


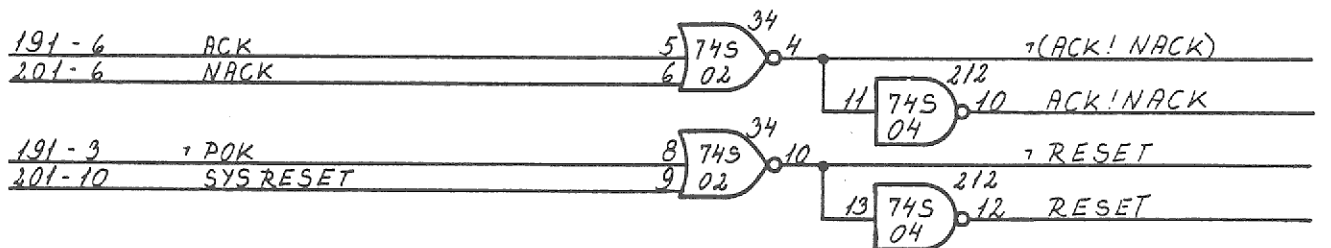
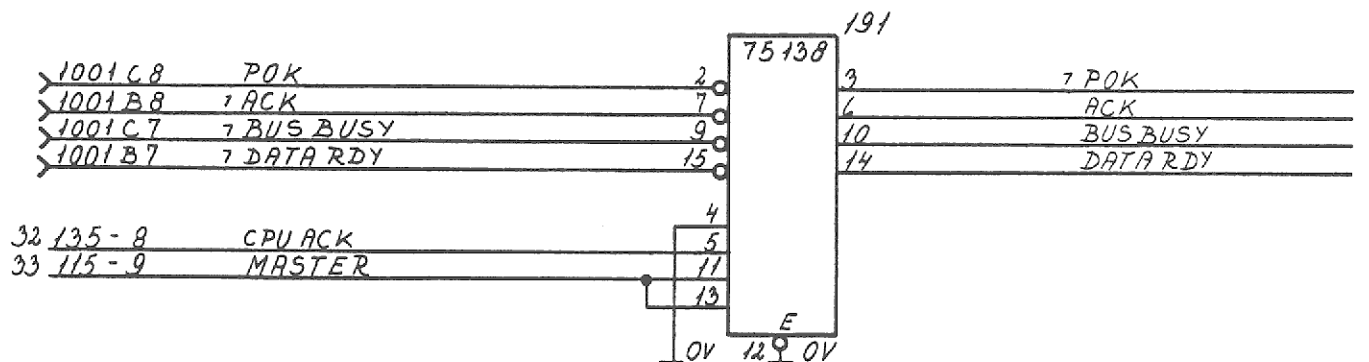
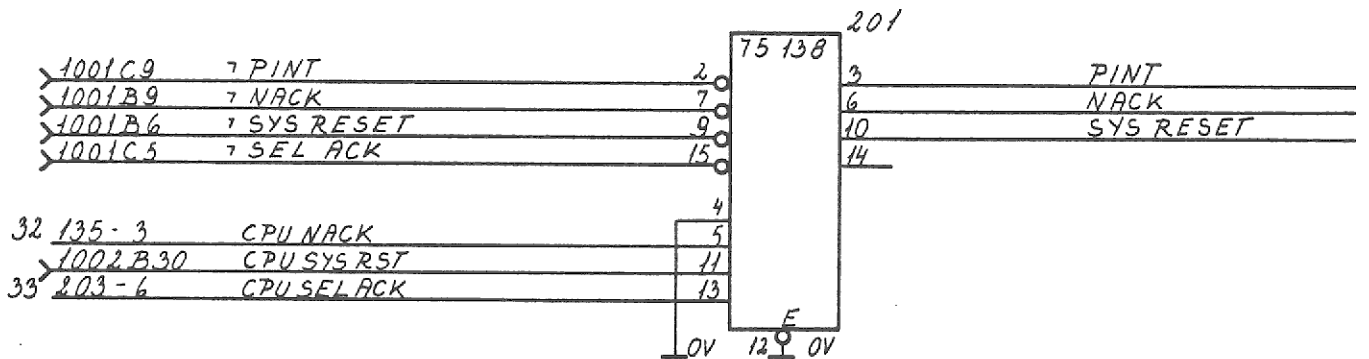
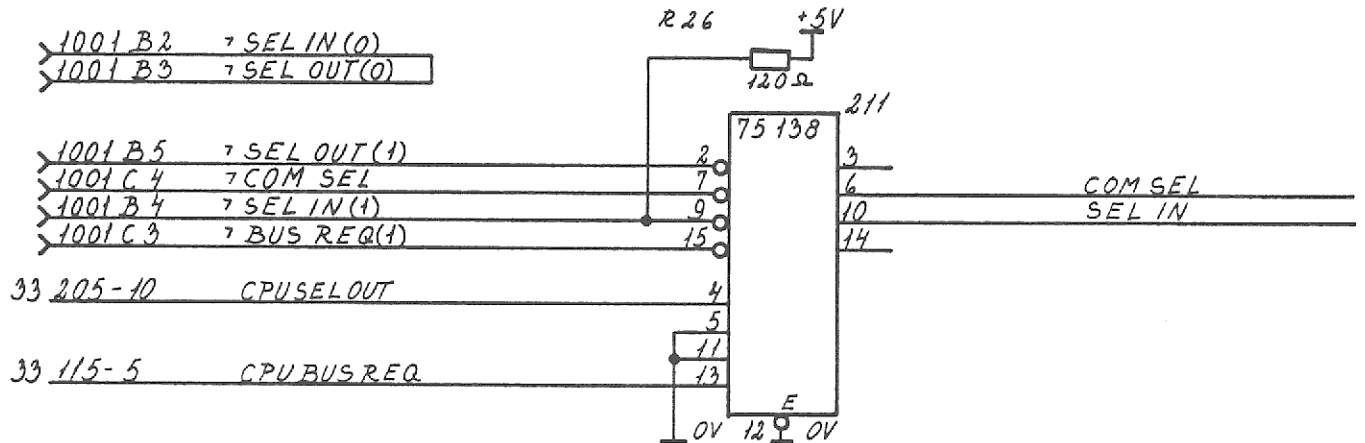
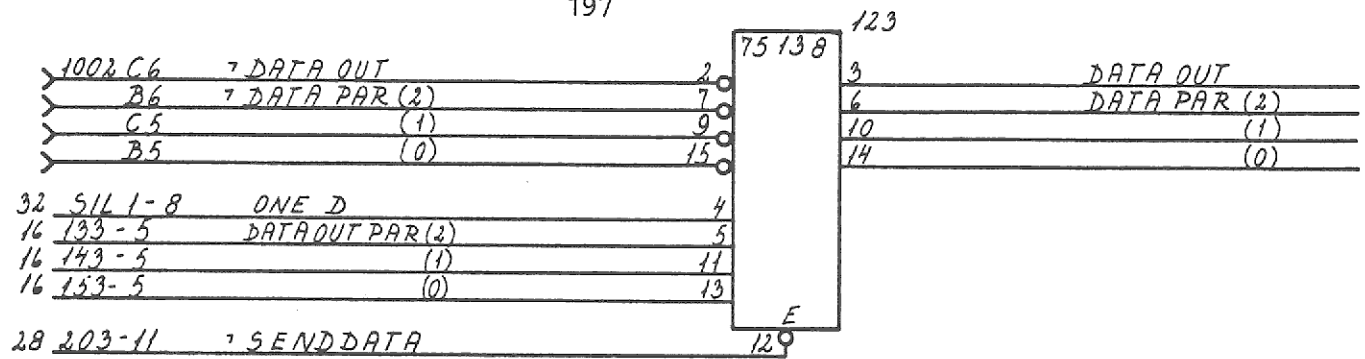
ARJ
800318
AGA
800901

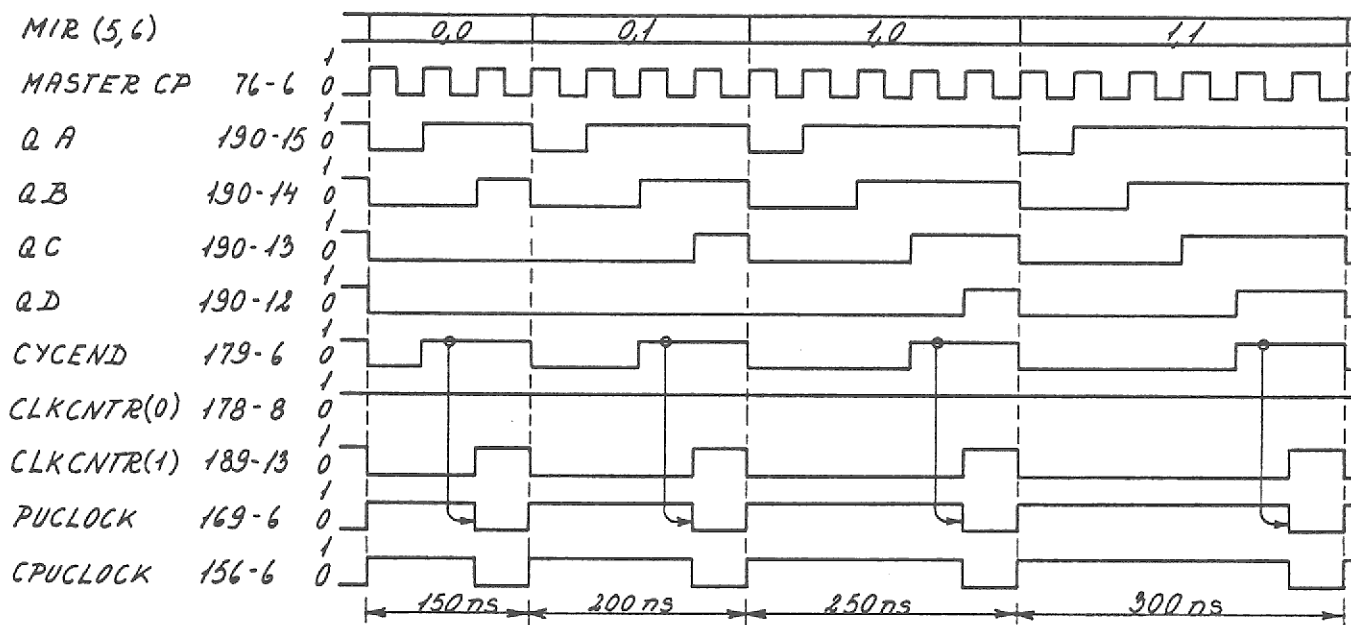


CPU 822
R 13047



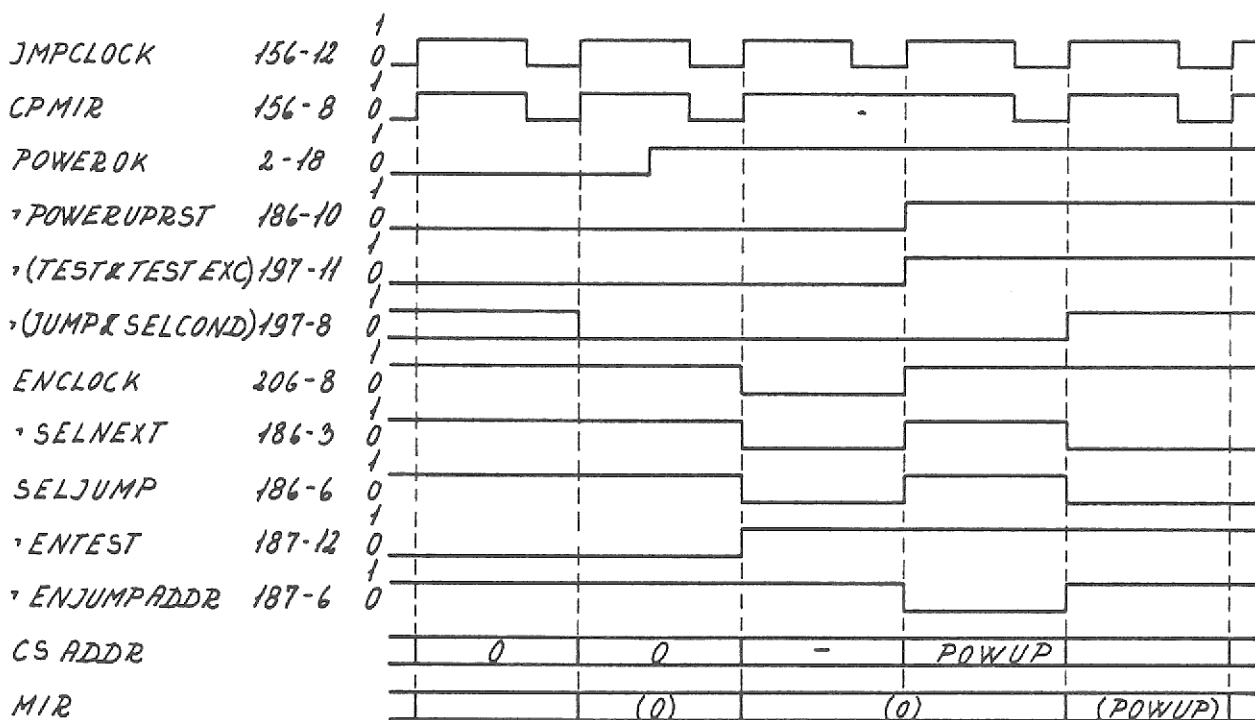




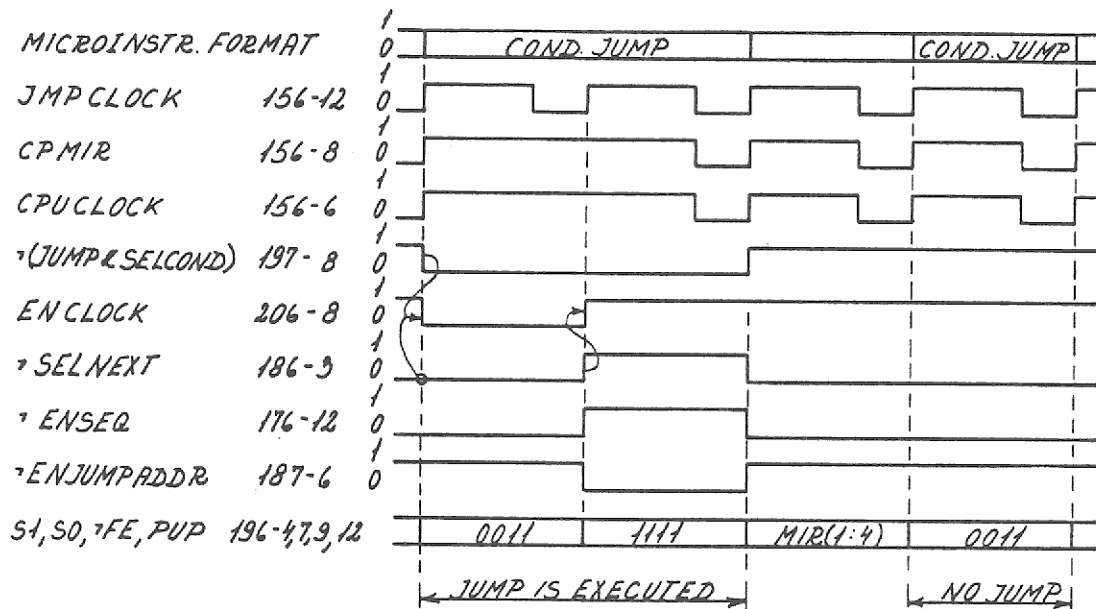


CLKCNTNTR(0,1)	FUNCTION
00	HOLD
01	NOT USED
10	SHIFT RIGHT
11	LOAD

POWER UP

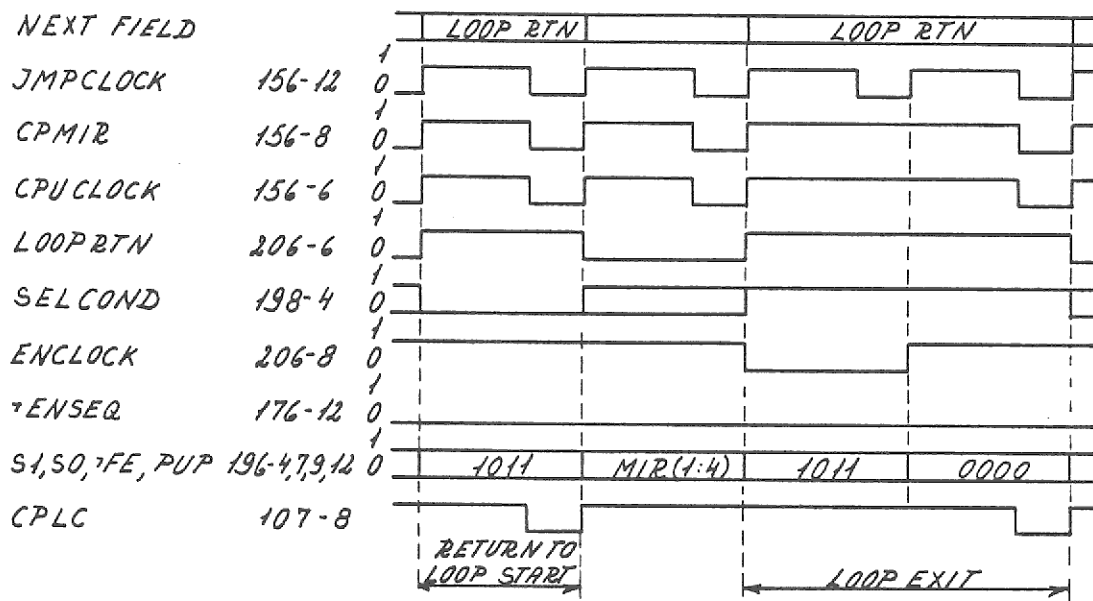


Control store location 0
contains a jump to POWUP.

CONDITIONAL JUMP

When a jump is executed

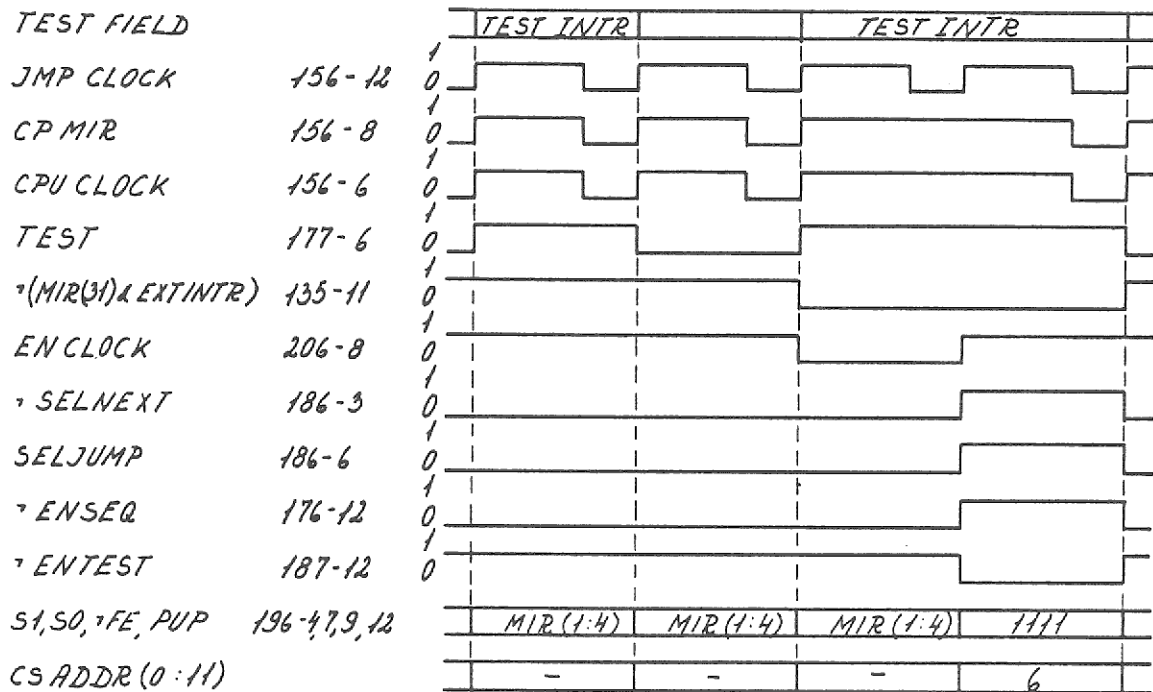
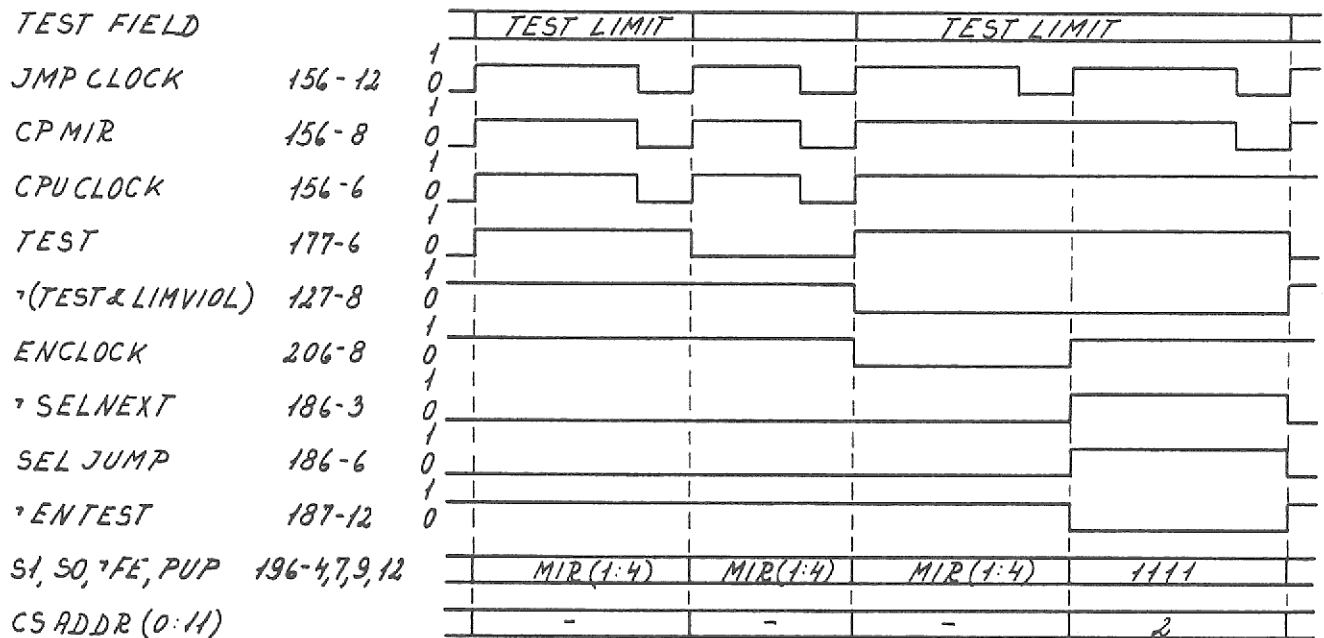
the microinstruction time is doubled.

LOOP RETURN

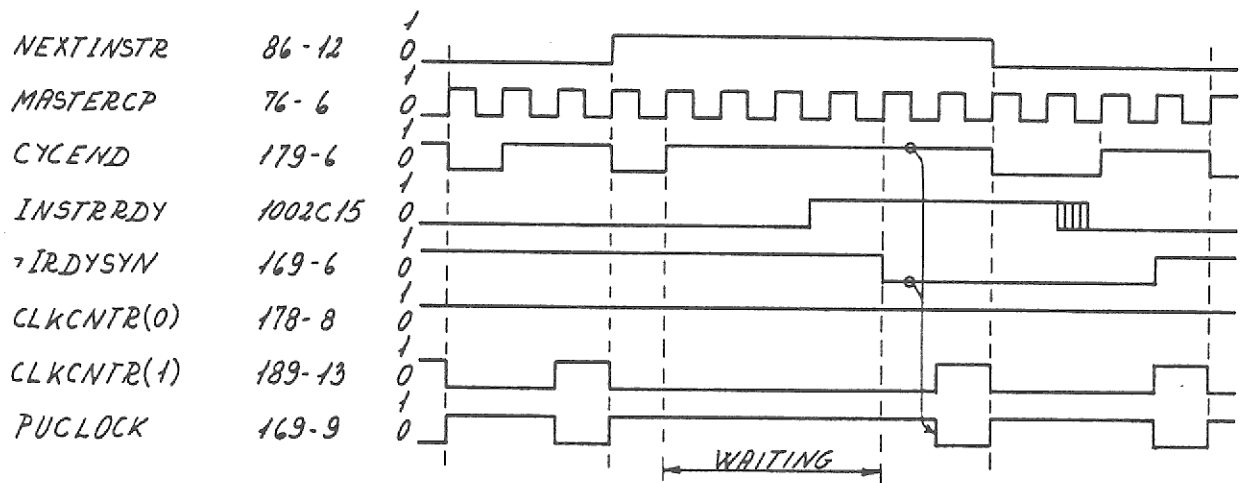
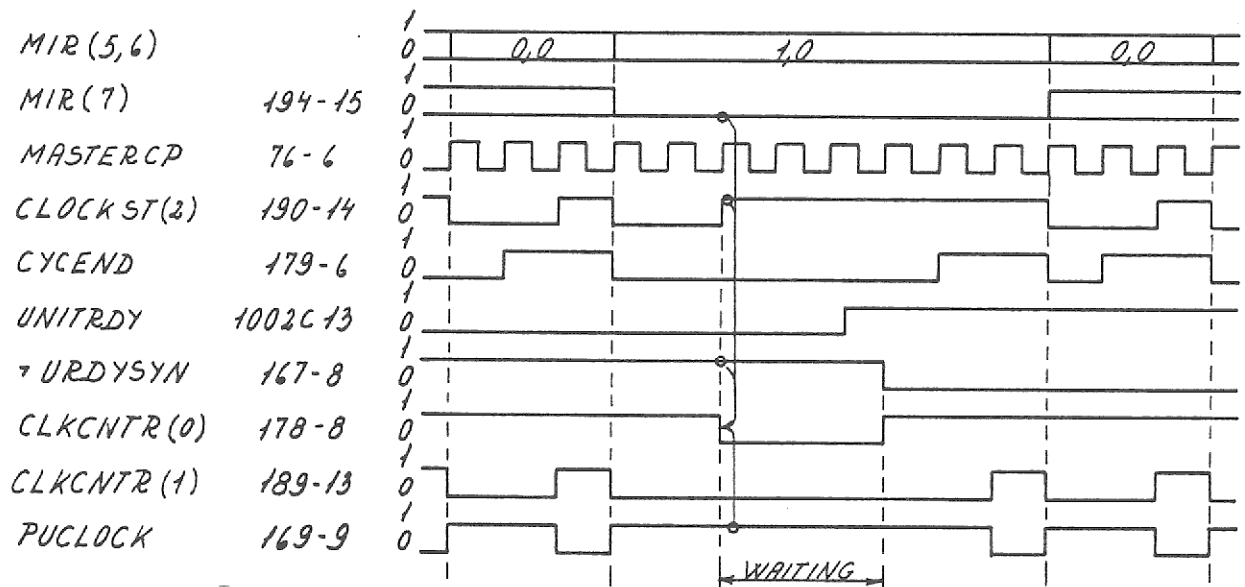
The microinstruction time

is doubled at loop exit.

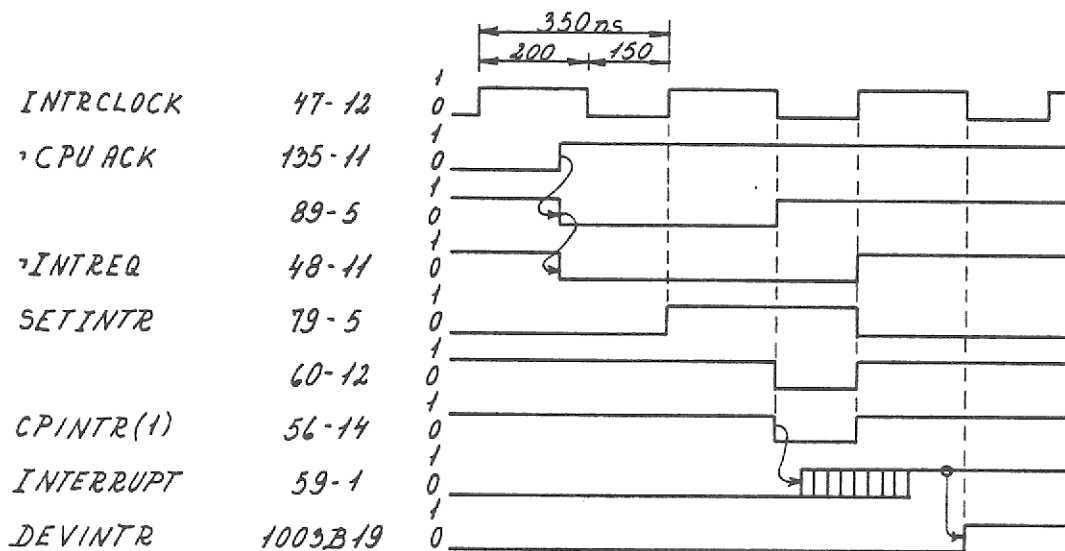
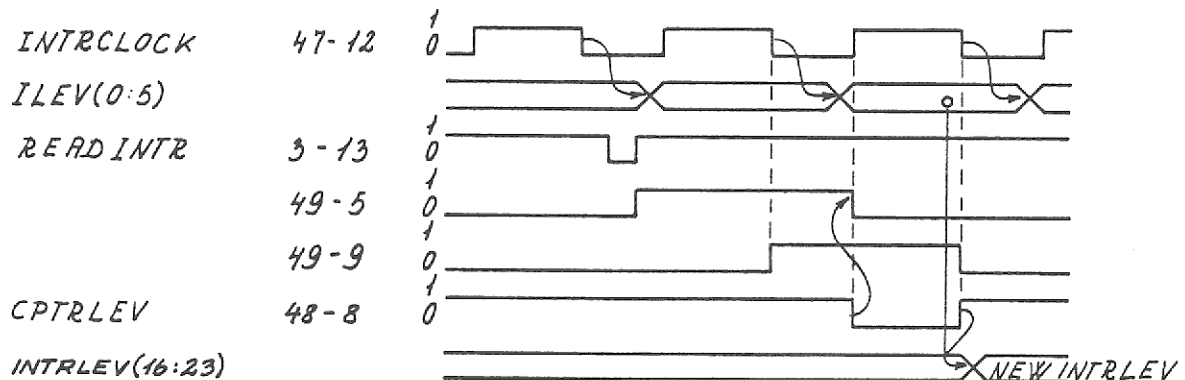
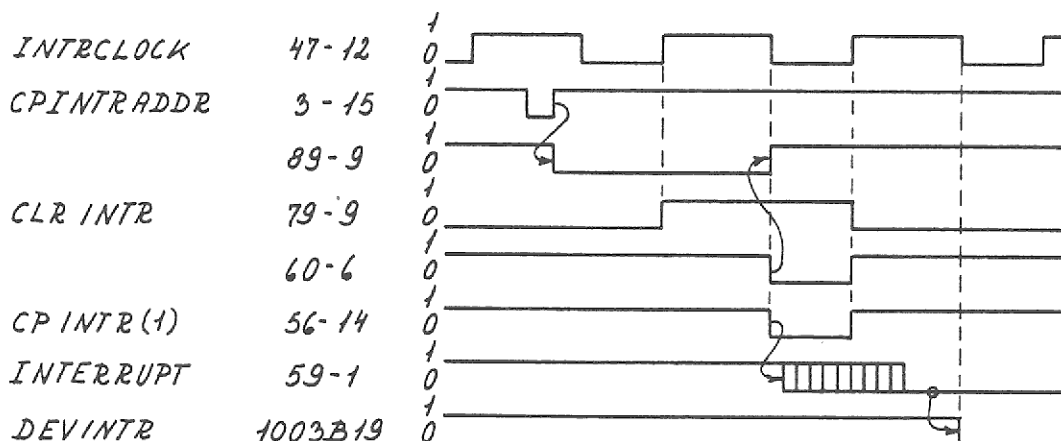
800130 J
RAJ
ACA
810701

TEST EXTERNAL INTERRUPTTEST LIMIT VIOLATION

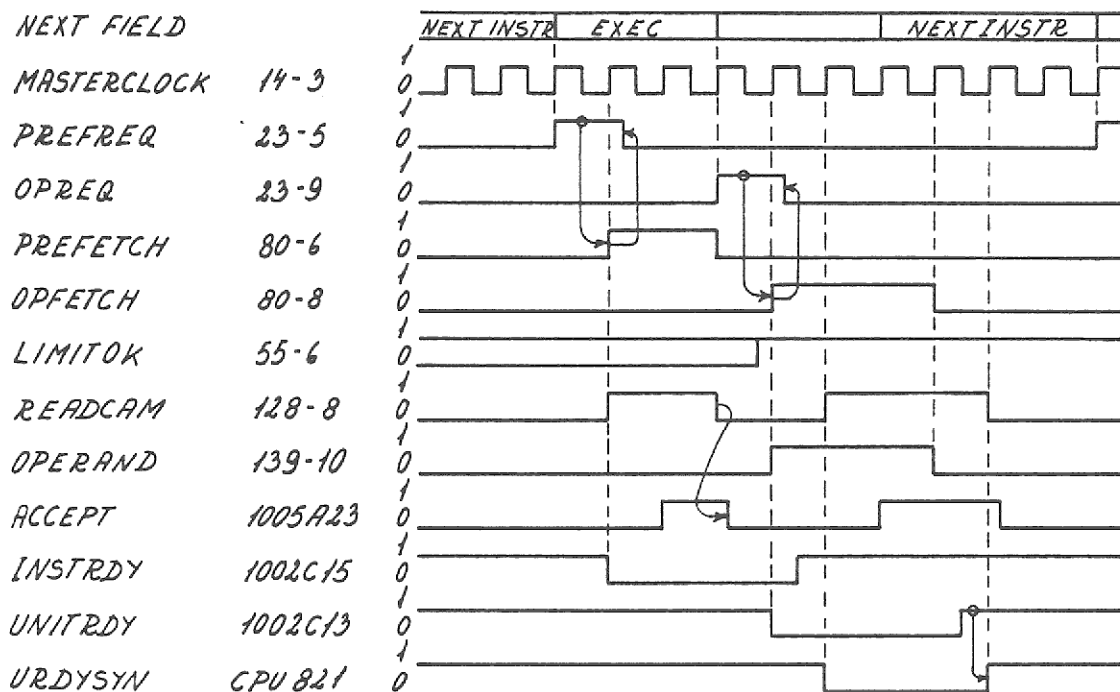
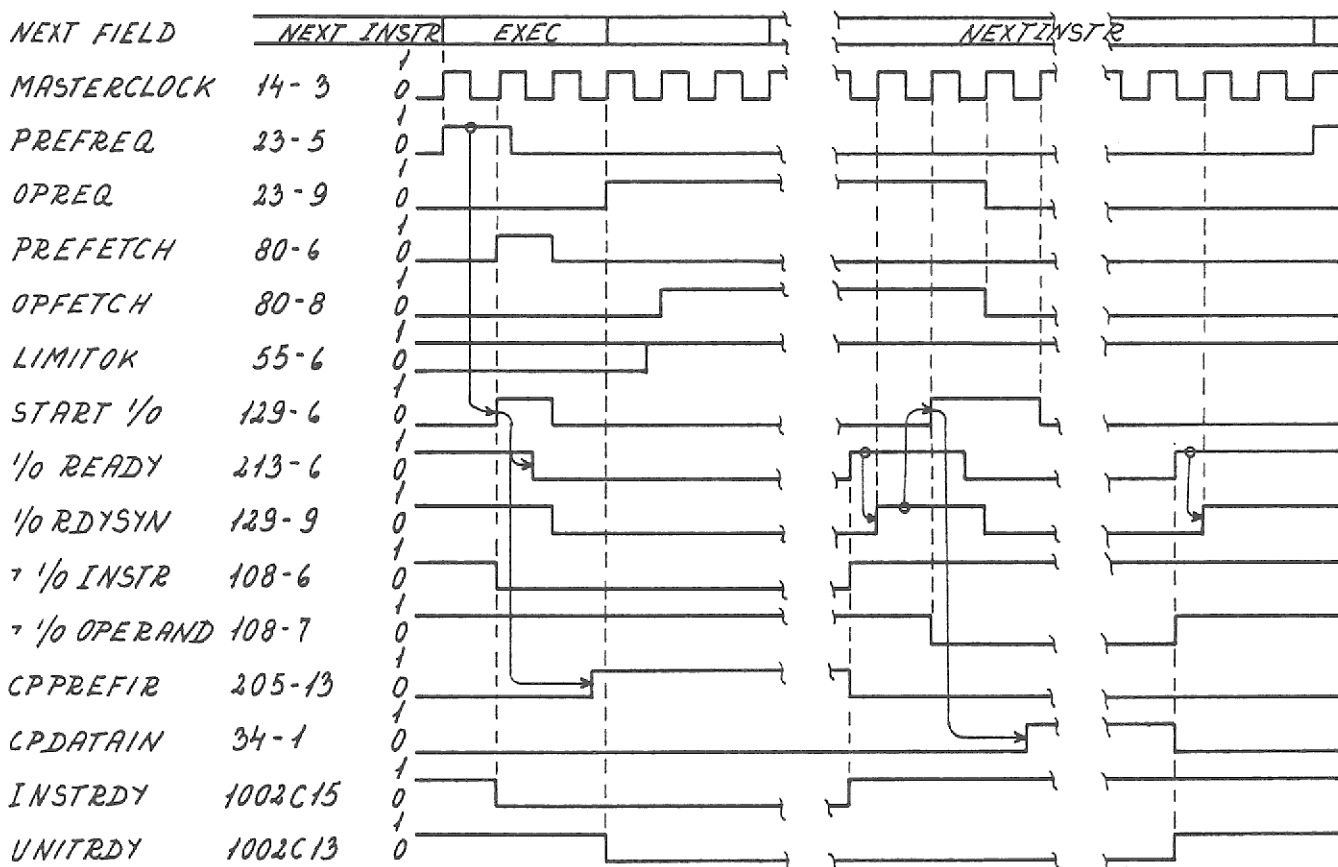
In case of trap on limit violation
CPU CLOCK is disabled.

INSTRUCTION FETCH SYNCHRONIZATIONUNIT SYNCHRONIZATION

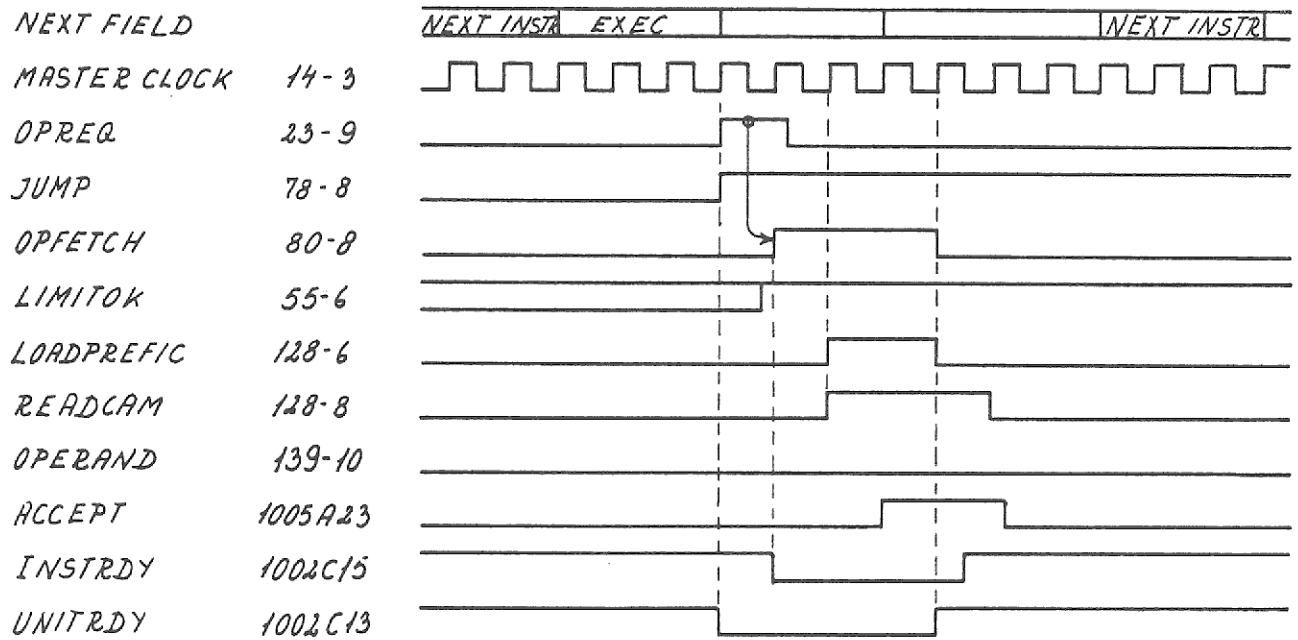
800130
AAJ
810702
RGA

SET INTERRUPT LEVEL (external interrupt request)READ INTERRUPT LEVELCLEAR INTERRUPT LEVEL

800130 AGA
RAJ 810703

REGISTER LOAD (with CAM, address within limits).REGISTER LOAD (without CAM, address within limits).

800130
B10703
AGA

JUMP LINK (with CAM, address within limits).

800130
MSJ
810703
AGA

MICROPROGRAM PROM POSITION LIST

<u>PROM NO</u>	<u>POSITION</u>	<u>TYPE</u>
ROA 330	345	6353-1
331	341	-
332	337	-
333	333	-
334	329	-
335	325	-
336	321	-
337	317	-
338	313	-
339	309	-
340	305	-
341	301	-
ROA 342	346	6353-1
343	342	-
344	338	-
345	334	-
346	330	-
347	326	-
348	322	-
349	318	-
350	314	-
351	310	-
352	306	-
353	302	-
ROA 354	347	6353-1
355	343	-
356	339	-
357	335	-
358	331	-
359	327	-
360	323	-
361	319	-
362	315	-
363	311	-
364	307	-
365	303	-
ROA 366	105	6306-1
367	104	-
368	103	-
ROA 369	101	6353-1
370	91	-
371	81	-
ROA 372	111	6306-1

ROA 307, DESTINATION ADDR. DECODING PROM

<u>OCTAL ADDR</u>	<u>01</u>	<u>02</u>	<u>03</u>	<u>04</u>	<u>05</u>	<u>06</u>	<u>07</u>	<u>08</u>
0	0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	0	0
2	1	0	0	0	1	1	0	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0	0
5	0	1	0	0	1	1	0	0
6	0	0	1	0	1	1	0	0
7	0	0	0	0	1	1	0	0
10	0	0	0	0	1	1	0	0
11	0	0	0	1	1	1	0	0
12	0	0	0	0	1	1	0	0
13	0	0	0	0	1	1	0	0
14	0	0	0	0	1	1	0	0
15	0	0	0	0	1	1	0	0
16	0	0	0	0	1	1	0	0
17	0	0	0	0	1	1	0	0
20	0	0	0	0	1	1	0	0
21	0	0	0	0	1	1	0	0
22	0	0	0	0	1	1	0	0
23	0	0	0	0	1	1	0	0
24	0	0	0	0	1	1	0	0
25	0	0	0	0	1	1	0	0
26	0	0	0	0	1	1	0	0
27	0	0	0	0	1	1	0	0
30	0	0	0	0	1	1	0	0
31	0	0	0	0	1	1	0	0
32	0	0	0	0	1	1	0	0
33	0	0	0	0	1	1	0	0
34	0	0	0	0	1	1	0	0
35	0	0	0	0	1	1	0	0
36	0	0	0	0	1	1	0	0
37	0	0	0	0	1	1	0	0

01 = ENCPSCRATCHP
 02 = ENCPCPUSTATUS
 03 = ENCPCONTROUT
 04 = ENCPEASTAT
 05 = -,LOADIC
 06 = -,LOADLC
 07 = UNUSED
 08 = UNUSED

ROA 308, ROA 309 and ROA 310 CONSTANT PROMs

OCTAL ADDRESS	ROA 308 CONTENTS 01-08	ROA 309 CONTENTS 01-08	ROA 310 CONTENTS 01-08
0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 0	0 0 0 0 0 0 0 0
1	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1	1 1 1 1 0 0 0 0
2	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 1 0 0 1 1 0
3	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1
4	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 0 1 1 1
5	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 0 1 1 0
6	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 0 1 0 1
7	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1
10	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 0
11	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 0 0
12	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 1
13	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0
14	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1
15	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0
16	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0
17	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
20	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0
21	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
22	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0
23	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0
24	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
25	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
26	0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
27	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
30	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 0
31	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
32	0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
33	0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
34	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 1
35	1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
36	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0
37	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1

ROA 311, HALF-WORD MANIPULATOR CONTROL PROM

<u>OCTAL ADDR</u>	<u>OCTAL CONTENTS</u>
-------------------	-----------------------

0	000
1	001
2	000
3	001
4	002
5	001
6	003
7	002

10	001
11	001
12	000
13	000
14	000
15	000
16	002
17	002

20	000
21	000
22	000
23	000
24	000
25	000
26	000
27	000

30	000
31	000
32	000
33	000
34	000
35	000
36	000
37	000

ROA 312, HALF-WORD MANIPULATOR PROM, BITS(0:11)A0 = 0 , ADDRESSES 0-777

A1:A5	A6:A9																
	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
11	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
12	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
13	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
15	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
16	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
20	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
21	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
22	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
23	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
24	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
25	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
26	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
27	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
31	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
33	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
34	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
35	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
36	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
37	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	

Addresses: octal

Contents: octal

ROA 312, HALF-WORD MANIPULATOR PROM, BITS(0:11)A0 = 1 , ADDRESSES 1000-1777

A1:A5	A6:A9																
	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
1	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
3	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
5	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
10	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
11	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
12	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
13	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
14	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
15	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
16	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
17	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
21	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
22	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
23	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
24	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
25	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
26	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
27	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
30	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
31	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
32	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
33	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
34	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
35	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
36	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
37	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	

Addresses: octal

Contents: octal

ROA 313, HALF-WORD MANIPULATOR PROM, BITS(12:23)

A0 = 0 , ADDRESSES 0-777

A1:A5	A6:A9																
	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
1	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	
2	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02	
3	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	
4	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	04	
5	05	05	05	05	05	05	05	05	05	05	05	05	05	05	05	05	
6	06	06	06	06	06	06	06	06	06	06	06	06	06	06	06	06	
7	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	
14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
20	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
21	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
22	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
23	00	01	02	03	04	05	06	07	10	11	12	12	13	15	16	17	
24	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
25	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
26	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
27	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
30	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
31	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
32	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
33	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
34	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
35	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
36	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	
37	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	

Addresses: octal

Contents: octal

ROA 313, HALF-WORD MANIPULATOR PROM, BITS(12:23)A0 = 1 , ADDRESSES 1000-1777

<u>A1:A5</u>	<u>A6:A9</u>																
	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
11	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
15	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
16	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
17	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
20	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
21	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
22	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
23	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
24	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
25	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
26	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
27	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
30	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
31	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
32	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
33	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
34	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
35	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
36	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	
37	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	

Addresses: octal

Contents: octal

CPU 821 CONNECTOR 1001

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLT		-, SELIN(0)		
3	-	1001B2	-, SELOUT(0)		
4	-		-, SELIN(1)		
5	-	1001B4	-, SELOUT(1)		
6	-				
7	-				
8	-				
9	-				
10	-				
11	-				
12	-				
13	-				
14	-				
15	-				
16	-				
17	-				
18	-				
19	-				
20	-				
21	-				
22	-				
23	-				
24	-				
25	-				
26	-				
27	-				
28	-				
29	-				
30	-				
31	-				
32	+5 VOLT		+5 VOLTS		+5 VOLTS

CPU 821 CONNECTOR 1002

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLT				
3	-				
4	-				
5	-				
6	-				
7	-				
8	-		-, CPU2AVAIL		
9	-		-, CAMAVAIL		-, FPUAVAIL
10	-	145-12	-, READ(0)	145-11	-, READ(1)
11	-	145-10	-, READ(2)	145-9	-, READ(3)
12	-		LIMVIOL		I/O ERROR
13	-				UNITRDY
14	-				
15	-				INSTRRDY
16	-		PREFERROR		PC<8
17	-		INSTRBUS(0)		INSTRBUS(1)
18	-		(2)		(3)
19	-		(4)		(5)
20	-		(6)		(7)
21	-		(8)		(9)
22	-		(10)		(11)
23	-		(12)		(13)
24	-		(14)		(15)
25	-		(16)		(17)
26	-		(18)		(19)
27	-		(20)		(21)
28	-		(22)		(23)
29	-	76-10	MASTERLOCK	156-6	CPULOCK
30	-	2-14	SYSTEMRST	2-12	-, LOAD
31					
32	+5 VOLT		+5 VOLTS		+5 VOLTS

CPU 821 CONNECTOR 1003

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLT		POWEROK		PINT
3	-	71-18	CPUBUS(0)	71-16	CPUBUS(1)
4	-	71-14	(2)	71-12	(3)
5	-	61-18	(4)	61-16	(5)
6	-	61-14	(6)	61-12	(7)
7	-	51-18	(8)	51-16	(9)
8	-	51-14	(10)	51-12	(11)
9	-	41-18	(12)	41-16	(13)
10	-	41-14	(14)	41-12	(15)
11	-	31-18	(16)	31-16	(17)
12	-	31-14	(18)	31-12	(19)
13	-	21-18	(20)	21-16	(21)
14	-	21-14	(22)	21-12	(23)
15	-	166-11	-, NEXT INSTR	111-11	ENPREF
16	-	102-2	ENOPFETCH	102-5	ENJMPFETCH
17	-			0 VOLT	
18	-	0 VOLT			
19	-		DEVINTR	2-3	DISABLE
20	-		-, TCPINTR		-, SPAREINTR
21	-		-, CAMFAULT	2-5	-, WADDR
22	-	2-7	CBUNITF(0)	2-9	CBUNITF(1)
23	-	12-18	(2)	12-16	(3)
24	-	12-14	(4)	12-12	(5)
25	-	12-3	CBSOURCE(0)	12-5	CBSOURCE(1)
26	-	12-7	(2)	12-9	(3)
27	-	1-18	(4)	1-16	(5)
28	-	1-14	CBDEST(0)	1-12	CBDEST(1)
29	-	1-3	(2)	1-5	(3)
30	-	1-7	(4)	1-9	(5)
31					
32	+5 VOLT		+5 VOLTS		+5 VOLTS

CPU 821 CONNECTOR 1005

<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>	<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>
A1		AUTOLOAD NC	B1	0 VOLT	AUTOLOAD C
A2		AUTOLOAD NO	B2	0 VOLT	AUTOLOAD C
A3	200-2	POWEROKLAMP-	B3	R52	POWEROKLAMP+
A4	200-7	RUNLAMP-	B4	R49	RUNLAMP+
A5	200-9	AUTOLOADLAMP-	B5	R48	AUTOLOADLAMP+
A6		-, REMOTE AUTOLOAD	B6		0 VOLT
A7		UNUSED	B7		UNUSED
A8		-	B8		-
A9		-	B9		-
A10		-	B10		-
A11		-	B11		-
A12		-	B12		-
A13		-	B13		-
A14		-	B14		-
A15		-	B15		-
A16		-	B16		-
A17		-	B17		-
A18		-	B18		-
A19		-	B19		-
A20		-	B20		-
A21		-	B21		-
A22		-	B22		-
A23		-	B23		-
A24		-	B24		-
A25		-	B25		-

CPU 822 CONNECTOR 1001

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLT		-, SELIN(0)		-, BUSREQ(0)
3	-	1001B2	-, SELOUT(0)	211-15	-, BUSREQ(1)
4	-		-, SELIN(1)		-, COMSEL
5	-	211-2	-, SELOUT(1)	201-15	-, SELACK
6	-	201-9	-, SYSRESET		POB
7	-	191-15	-, DATARDY	191-9	-, BUSBUSY
8	-	191-7	-, ACK		POK
9	-	201-7	-, NACK		-, PINT
10	-	182-15	-, ADDR(0)	182-9	-, ADDR(1)
11	-	181-15	(2)	181-9	(3)
12	-	181-7	(4)	181-2	(5)
13	-	182-7	(6)	182-2	(7)
14	-	172-15	(8)	172-9	(9)
15	-	171-15	(10)	171-9	(11)
16	-	171-7	(12)	171-2	(13)
17	-	172-7	(14)	172-2	(15)
18	-	162-15	(16)	162-9	(17)
19	-	161-15	(18)	161-9	(19)
20	-	161-7	(20)	161-2	(21)
21	-	162-7	(22)	162-2	-, ADDRPAR
22	-	152-15	-, DATA(0)	152-9	-, DATA(1)
23	-	151-15	(2)	151-9	(3)
24	-	151-7	(4)	151-2	(5)
25	-	152-7	(6)	152-2	(7)
26	-	142-15	(8)	142-9	(9)
27	-	141-15	(10)	141-9	(11)
28	-	141-7	(12)	141-2	(13)
29	-	142-7	(14)	142-2	(15)
30	-	132-15	(16)	132-9	(17)
31	+12 VOLTS		+12 VOLTS		+12 VOLTS
32	+5 VOLTS		+5 VOLTS		+5 VOLTS

CPU 822 CONNECTOR 1002

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLT	131-15	-, DATA(18)	131-9	-, DATA(19)
3	-	131-7	(20)	131-2	(21)
4	-	132-7	(22)	132-2	(23)
5	-	123-15	-, DATAPAR(0)	123-9	-, DATAPAR(1)
6	-	123-7	(2)	123-2	-, DATAOUT
7	-				
8	-	0 VOLT	-, CPU2AVAIL		
9	-		-, CAMAVAIL		
10	-		-, READ(0)		-, READ(1)
11	-		-, READ(2)		-, READ(3)
12	-	24-9	LIMITVIOL	31-11	I/O ERROR
13	-			31-8	UNITRDY
14	-				
15		22-15	BYPASSCAM	31-3	INSTRRDY
16	-	31-6	PREFERROR	0 VOLT	PC<8
17	-	111-2	INSTRBUS(0)	111-5	INSTRBUS(1)
18	-	111-6	(2)	111-9	(3)
19	-	111-12	(4)	111-15	(5)
20	-	111-16	(6)	111-19	(7)
21	-	112-2	(8)	112-5	(9)
22	-	112-6	(10)	112-9	(11)
23	-	112-12	(12)	112-15	(13)
24	-	112-16	(14)	112-19	(15)
25	-	103-2	(16)	103-5	(17)
26	-	103-6	(18)	103-9	(19)
27	-	103-12	(20)	103-15	(21)
28	-	103-16	(22)	103-19	(23)
29	-		MASTERLOCK		CPUCLOCK
30	-				-, LOAD
31	-12 VOLTS		-12 VOLTS		-12 VOLTS
32	+5 VOLTS		+5 VOLTS		+5 VOLTS

CPU 822 CONNECTOR 1003

PIN	A ROW	B ROW		C ROW	
	SIGNAL	GEN	SIGNAL	GEN	SIGNAL
1	+5 VOLTS		+5 VOLTS		+5 VOLTS
2	0 VOLTS	13-6	POWEROK	201-3	PINT
3	-	71-18	CPUBUS(0)	71-16	CPUBUS(1)
4	-	71-14	(2)	71-12	(3)
5	-	71-3	(4)	71-5	(5)
6	-	71-7	(6)	71-9	(7)
7	-	72-18	(8)	72-16	(9)
8	-	72-14	(10)	72-12	(11)
9	-	72-3	(12)	72-5	(13)
10	-	72-7	(14)	72-9	(15)
11	-	63-18	(16)	63-16	(17)
12	-	63-14	(18)	63-12	(19)
13	-	63-3	(20)	63-5	(21)
14	-	63-7	(22)	63-9	(23)
15	-		-,NEXTINSTR		ENPREF
16	-		ENOPFETCH		ENJUMPFETCH
17	-				
18	-				
19	-	24-5	DEVINTR		DISABLE
20	-	21-11	-,TCPINTR		
21	-				-,WADDR
22	-		UNITFUNC(0)		UNITFUNC(1)
23	-		(2)		(3)
24	-		(4)		(5)
25	-		CBSOURCE(0)		CBSOURCE(1)
26	-		(2)		(3)
27	-		(4)		(5)
28	-		CBDEST(0)		CBDEST(1)
29	-		(2)		(3)
30	-		(4)		(5)
31					
32	+5 VOLTS	-	+5 VOLTS		+5 VOLTS

CPU 822 CONNECTOR 1004

<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>	<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>
A1		0 VOLT	B1	7-7	-, TRANSMDATA
A2		0 VOLT	B2		-, RECDATA
A3		0 VOLT	B3		DATASETDRY
A4		0 VOLT	B4	7-6	DATATERMRDY
A5		UNUSED	B5		UNUSED
A6		-	B6		-
A7		-	B7		-
A8		-	B8		-
A9		-	B9		-
A10		-	B10		-
A11		-	B11		-
A12		-	B12		-
A13		-	B13		-
A14		-	B14		-
A15		-	B15		-
A16		-	B16		-
A17		-	B17		-
A18		-	B18		-
A19		-	B19		-
A20		-	B20		-
A21		-	B21		-
A22		-	B22		-
A23		-	B23		-
A24		-	B24		-
A25		-	B25		-

CPU 822 CONNECTOR 1005

<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>	<u>PIN</u>	<u>GEN</u>	<u>SIGNAL</u>
A1	220-12	CAMADDR(22)	B1		0 VOLT
A2	220-9	(21)	B2		-
A3	220-7	(20)	B3		-
A4	220-4	(19)	B4		-
A5	209-12	(18)	B5		-
A6	209-9	(17)	B6		-
A7	209-7	(16)	B7		-
A8	209-4	(15)	B8		-
A9	208-12	(14)	B9		-
A10	208-9	(13)	B10		-
A11	208-7	(12)	B11		-
A12	208-4	(11)	B12		-
A13	207-12	(10)	B13		-
A14	207-9	(9)	B14		-
A15	207-7	(8)	B15		-
A16	207-4	(7)	B16		-
A17	217-12	(6)	B17		-
A18	217-9	(5)	B18		-
A19	217-7	(4)	B19		-
A20	217-4	(3)	B20		-
A21	216-12	(2)	B21		-
A22	216-9	(1)	B22		-
A23		ACCEPT	B23		-
A24	128-9	-, READCAM	B24		-
A25	139-10	OPERAND	B25		-

CBL 826 OCP and REMOTE AUTOLOAD CABLE

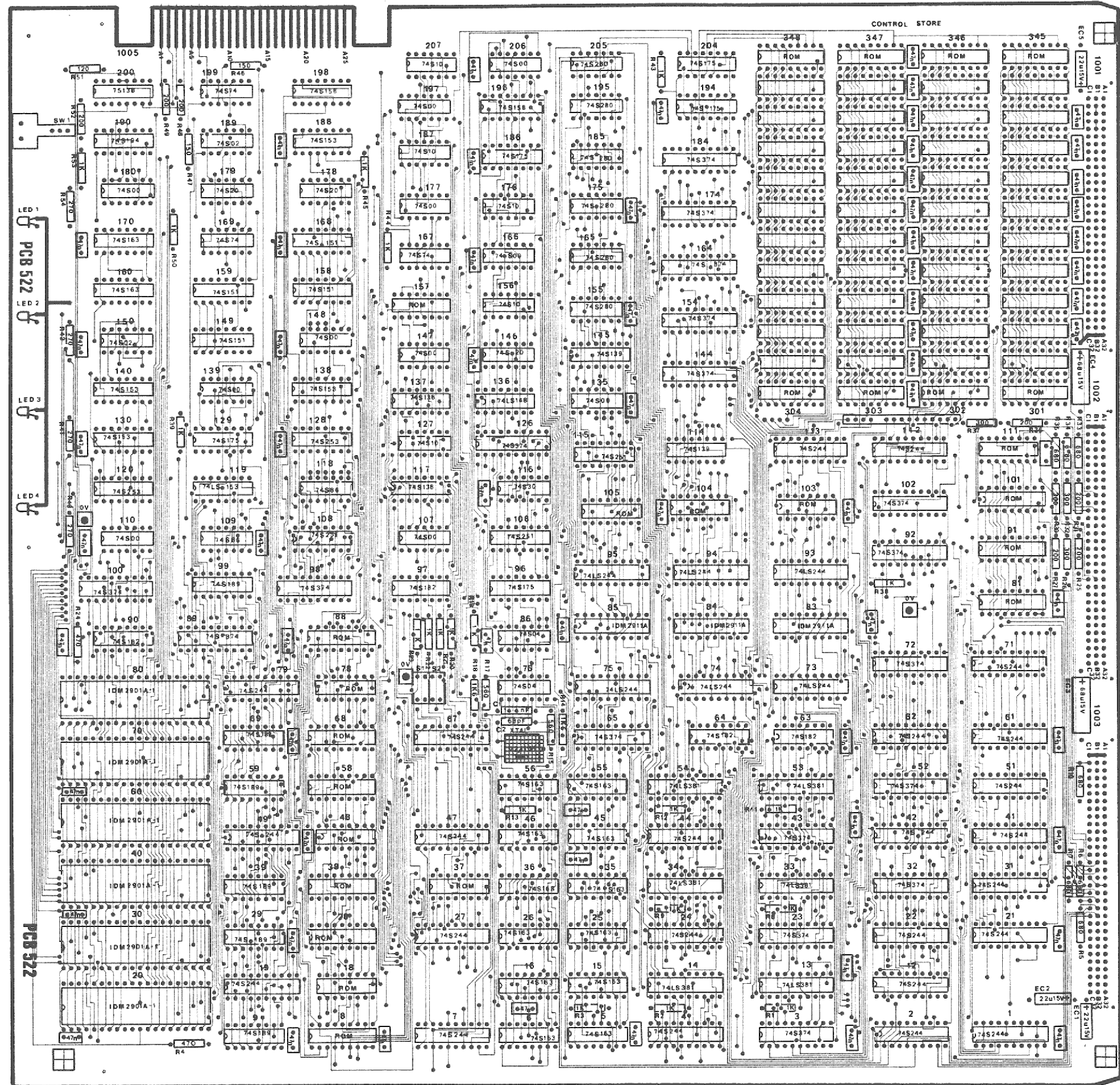
<u>J1</u>		<u>J1005</u>	<u>SIGNAL</u>
A2	—○—	A1	AUTOLOAD NC
A3	—○—	B1	AUTOLOAD C
A4	—○—	A2	AUTOLOAD NO
A5	—○—	B2	AUTOLOAD C
A6	—○—	A3	POWEROKLAMP-
A7	—○—	B3	POWEROKLAMP+
B7	—○—	A4	RUNLAMP-
B6	—○—	B4	RUNLAMP+
B5	—○—	A5	AUTOLOADLAMP-
B3	—○—	B5	AUTOLOADLAMP+
<u>J2</u>			
A2	—○—	A6	-, REMOTE AUTOLOAD
A3	—○—	B6	0 VOLT

CBL 904 TECHNICIAN CONSOLE CABLE

<u>J1004</u>		<u>J1</u>	<u>J2</u>	<u>SIGNAL</u>
B1	—○—	2	6	-, TRANSM DATA
A1	—○—			
B2	—○—	3	3	-, RECDATA
A2	—○—			
B3	—○—	6	5	DATASET RDY
A3	—○—			
B4	—○—	20	1	DATATERM RDY
A4	—○—	7	9	0 VOLT

Pos 77 (S1 - S2).
Operating mode selection

Pos 66 (XTAL)
Frequency: 20.0000 MHz



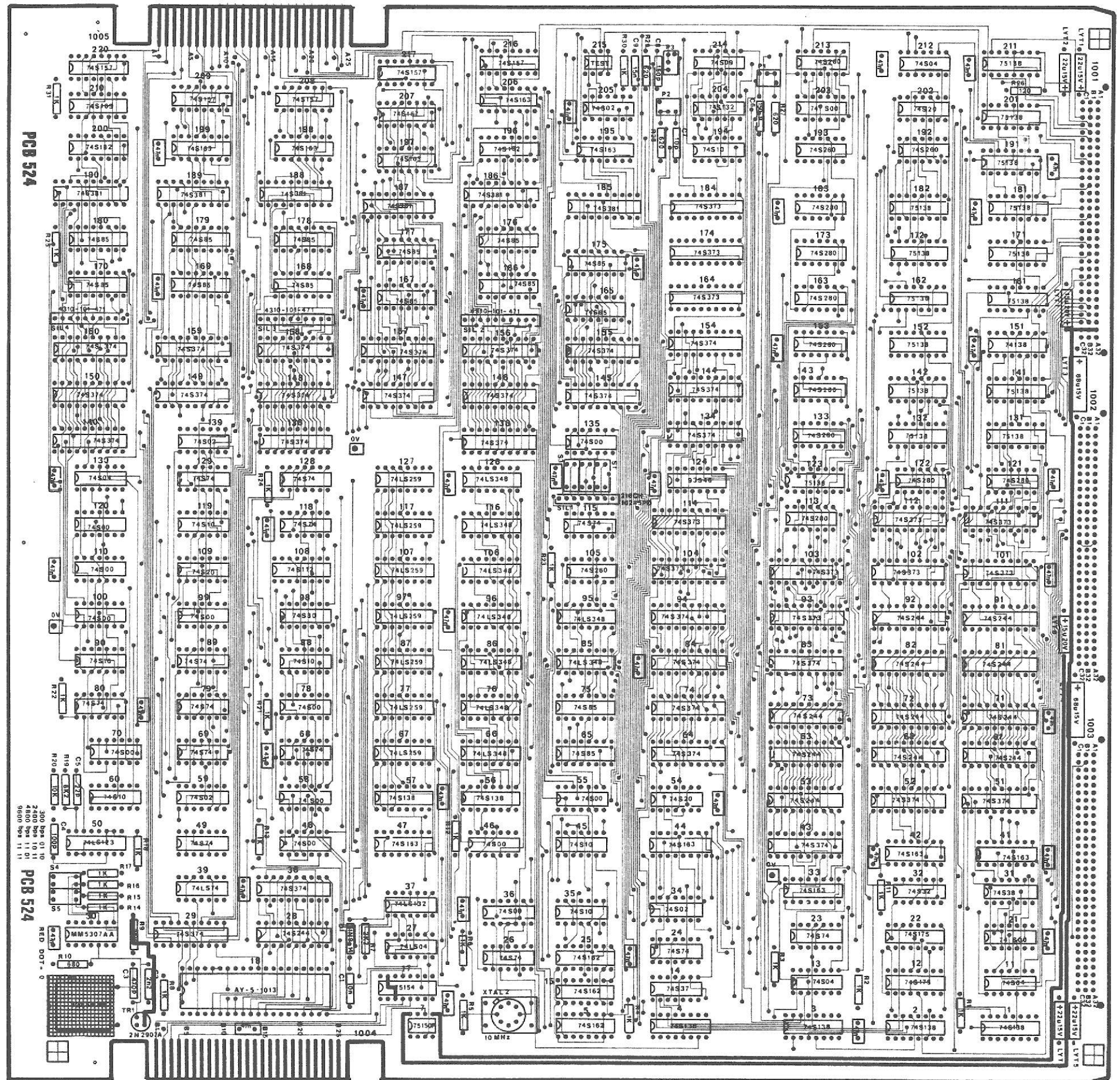
Pos 125 (S1 - S3)
CPU device address.

Pos 40 (S4 - S5)
Transmission speed for
technician console.

P1: DATARDY deskew delay.

P2: SELIN deskew delay.

P3: ACK deskew delay.



RETURN LETTER

Title: CPU820, Technical Manual

RCSL No.: 30-M282

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

..... Fold here

..... Do not tear - Fold here and staple

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark