**RCSL No:** 30-M282

**Edition:** June 1981

**Author:** Aage Jørgensen

**Title:**

# CPU820
# Technical Manual

RCSL 42-11592

A/S REGNECENTRALEN
af 1979

**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)



Printed by A/S Regnecentralen af 1979, Copenhagen



RCSL 42-i1592

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

TABLE OF CONTENTS (continued) PAGE

## 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.

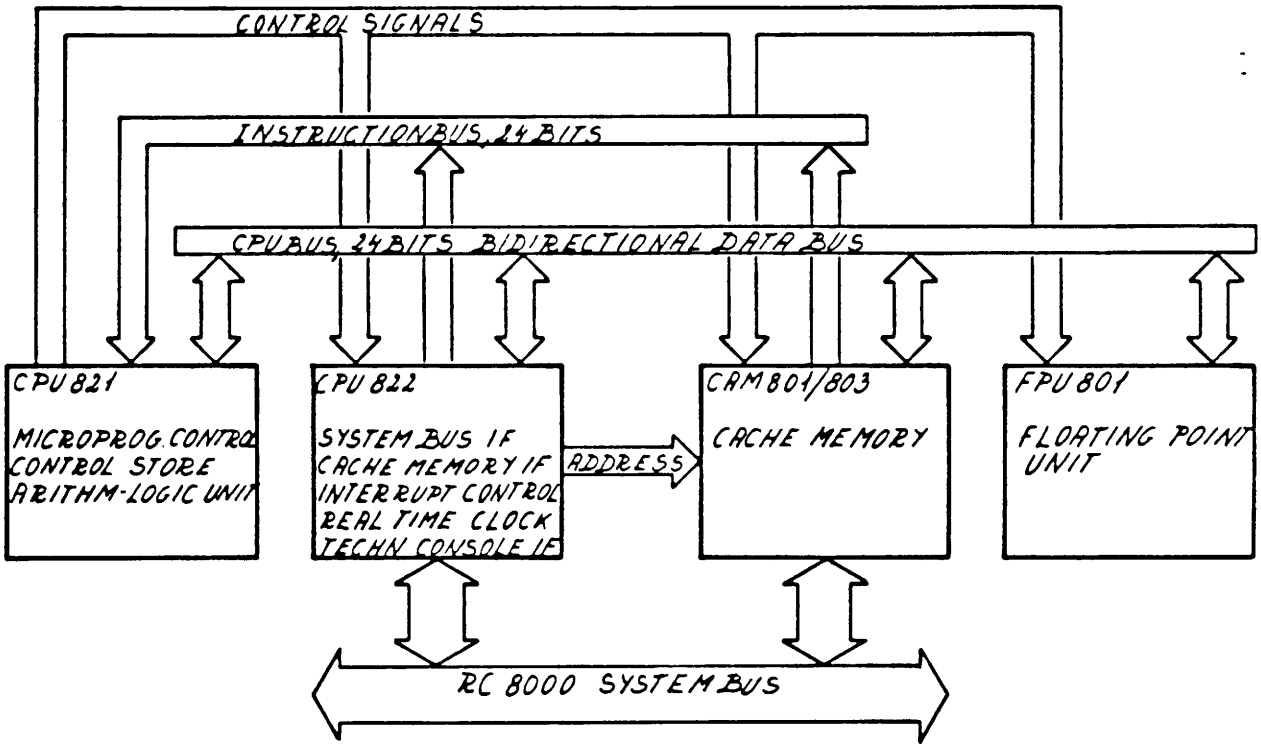



Figure 1: CPU820 extended with cache memory and floating point unit.

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

## 2.1 CPU821 Data Paths

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

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

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.

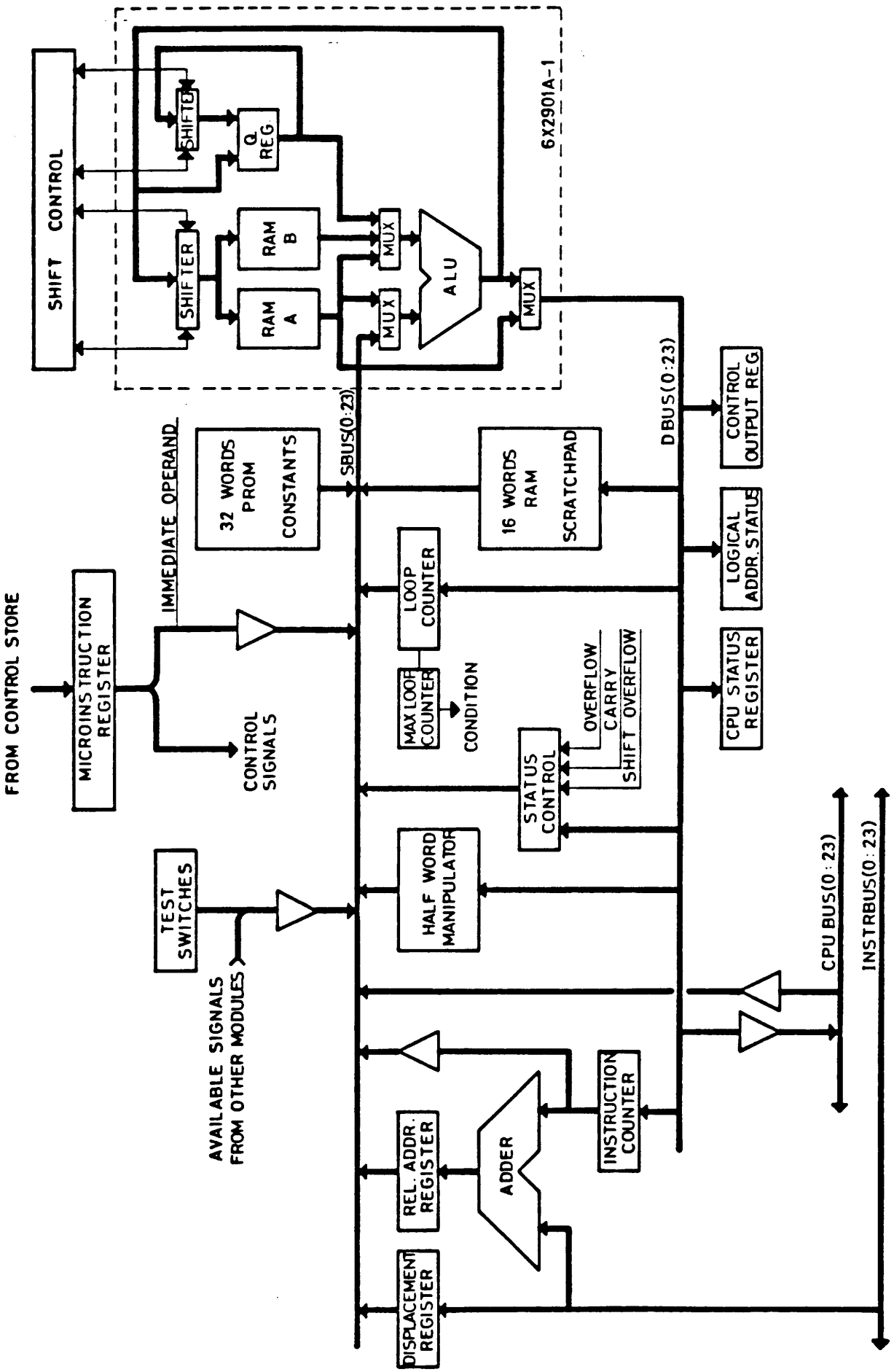



Figure 2: CPU821 data paths.

### 2.1.3 Arithemetic Logic Unit



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



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



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



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 adddressed 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 >= 0 or LCMAX >= 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

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

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 ext INSTRBUS(12) con INSTRBUS(12:23)

### 2.1.12 Relative Address Register

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 ext INSTRBUS(12) con INSTRBUS(12:23) + IC.

### 2.1.13 Switch Register

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) = CONTINOUS 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



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)

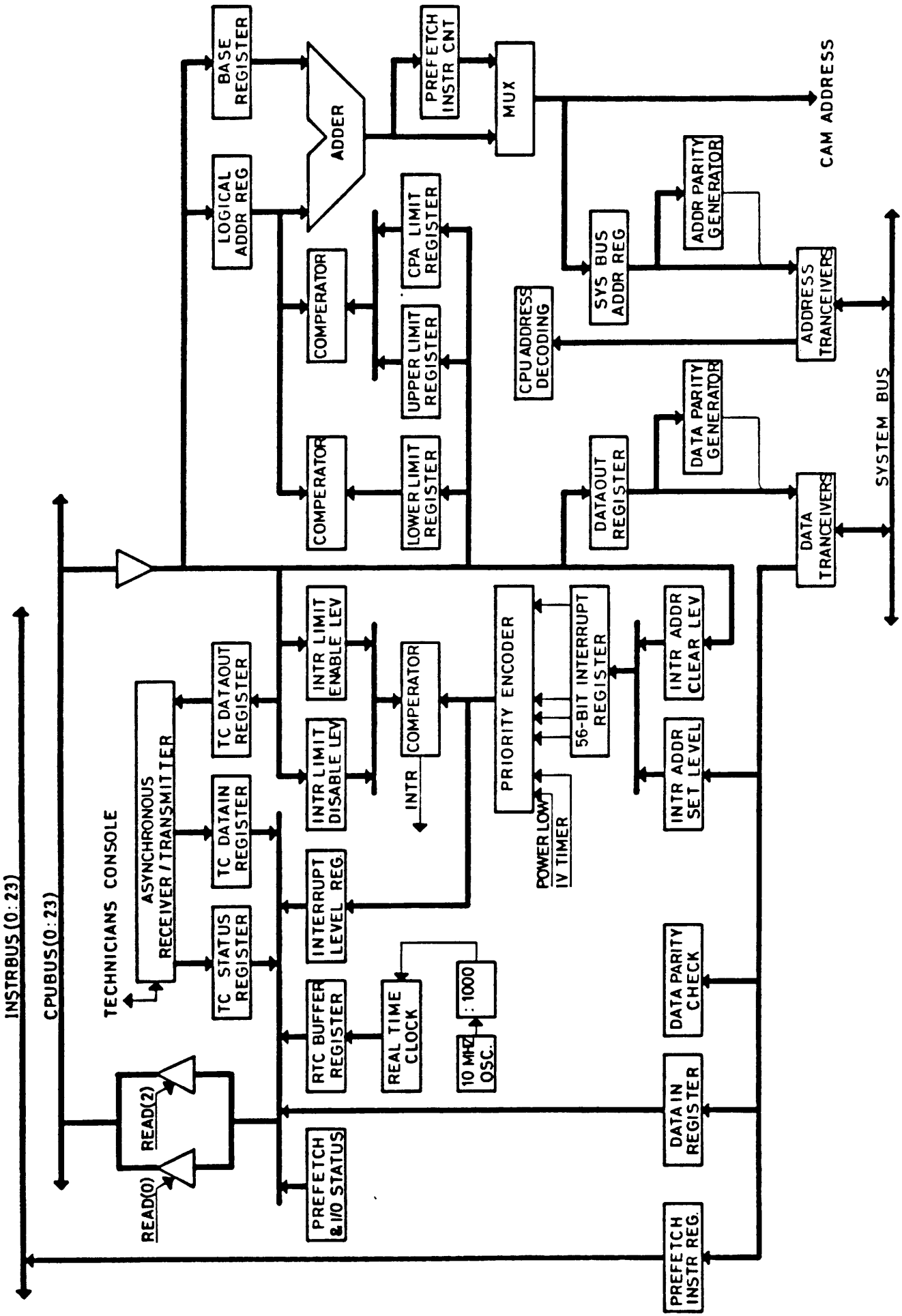



Figure 3: CPU822 data paths.

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

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

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 >= 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 >= 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.<br>IOSTAT(4:8) = ADDRESS(16:20)<br>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 | LOGAADR < 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.<br>IOSTAT(12) = 0<br>IOSTAT(13) = PARITY ERROR<br>IOSTAT(14) = TIME OUT<br>IOSTAT(15) = NACK |
| 20:23 | I/O STATUS. Valid after termination of a memory or I/O operation.<br>IOSTAT(20) = LIMIT VIOLATION<br>IOSTAT(21) = PARITY ERROR<br>IOSTAT(22) = TIME OUT<br>IOSTAT(23) = NACK |

### 2.2.1.11 CAM Control Register



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 <= 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:

| BIT | DESCRIPTION |
|---|---|
| 21 | -,DATASETRDY. A 0 indicates that TC is connected and is ready to use. |
| 22 | DATAAVAIL. A 1 indicates that a character has been received by the UART and is available in the TCDAIN register. Addressing TCDAIN as destination resets DATAAVAIL. |
| 23 | TCOUTFULL. A 0 indicates that the UART transmitter is ready to accept a character. TCOUTFULL 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 Technicican Console Data In Register 2.2.4.2

The Technicican 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 CPU821 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.

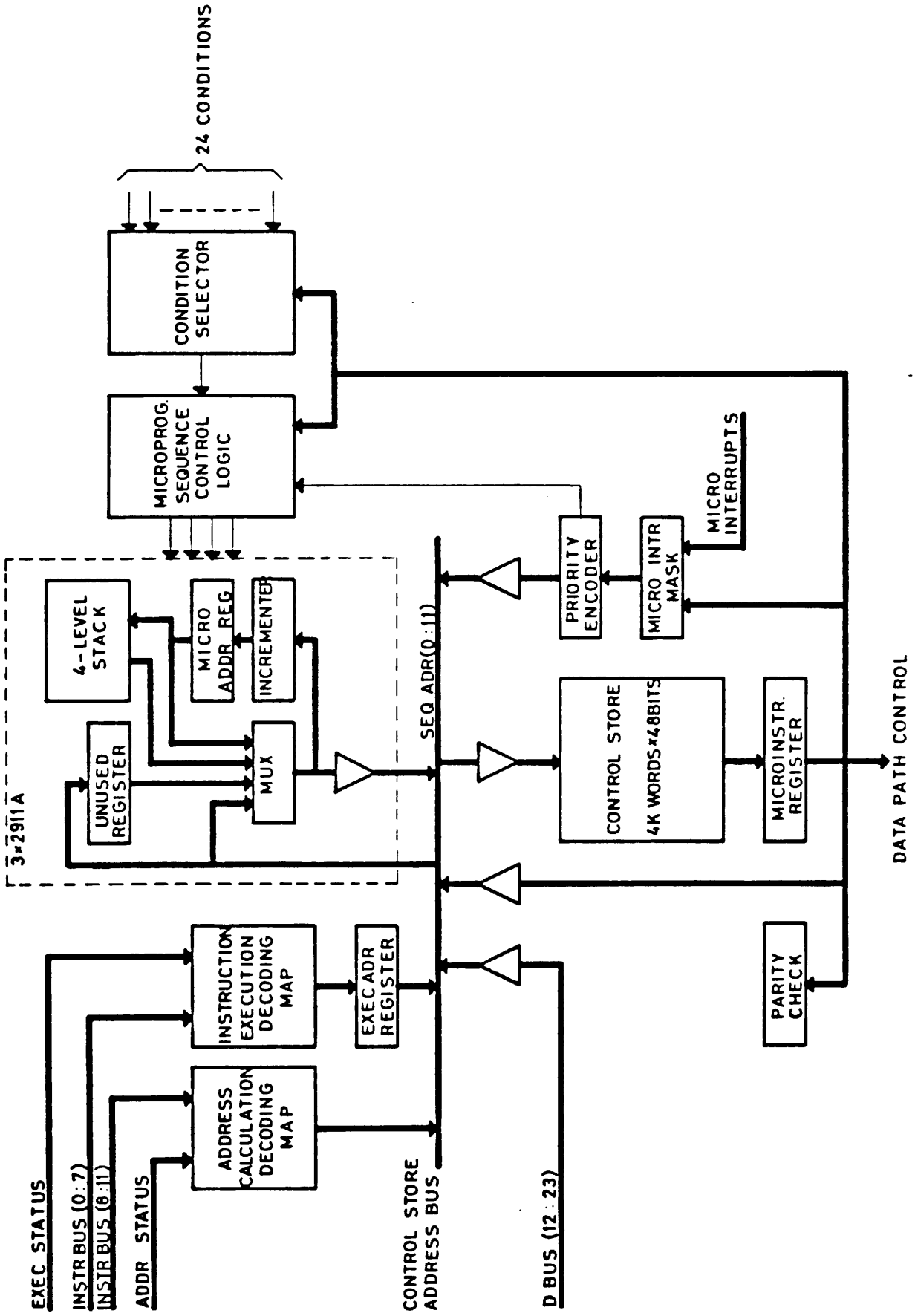



Figure 4: Microprogram control unit.

### 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.

| ENTRY ADDR BIT | SOURCE |
|---|---|
| 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



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.

| ENTRY ADDR BIT | SOURCE |
|---|---|
| 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



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 sub-sections. Fields referring to a single format are described in connection with that format.

### 2.4.1 P Field = MIR(0) 2.4.1

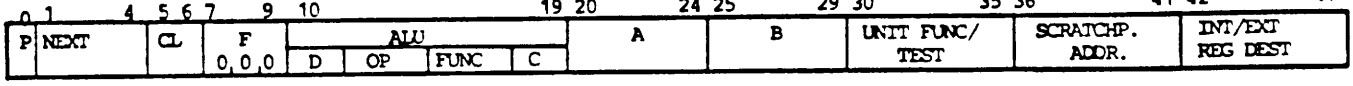
The P field is the parity bit for the microinstruction. Odd parity is used. In case of parity error the processor stops immediately and the indicator 'CS PARITY ERROR' will be lit. MIR contains the faulty microinstruction, which is not executed. CSADDR depends on the NEXT field of the faulty microinstruction. 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

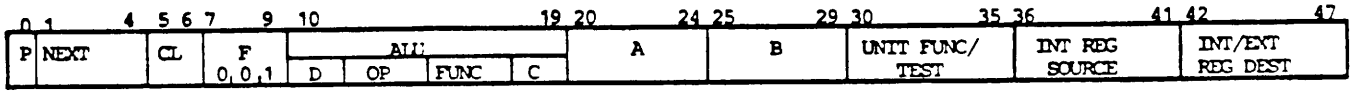
The NEXT field defines CSADDR for the next microinstruction to be executed. The following abbreviations are used to describe the function of the NEXT field.

| | |
|---|---|
| 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. |

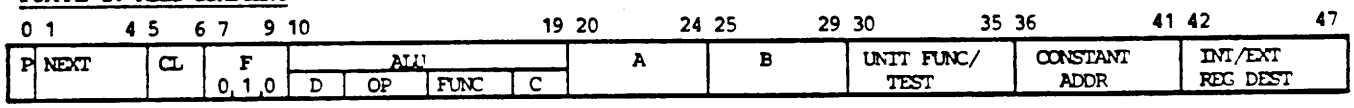
FORMAT 0: Read/Load Scratchpad



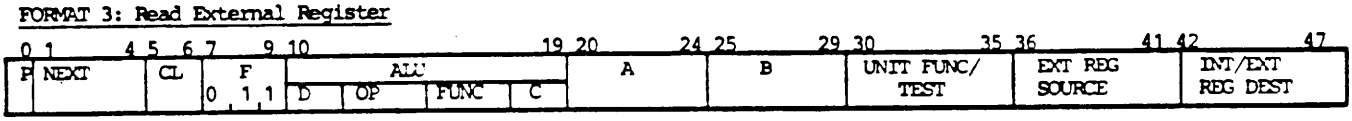

FORMAT 1: Read/Load Register



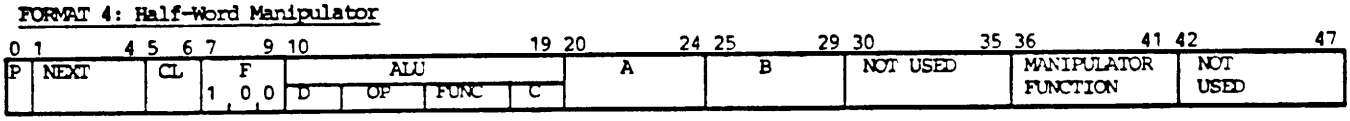

FORMAT 2: Read Constant



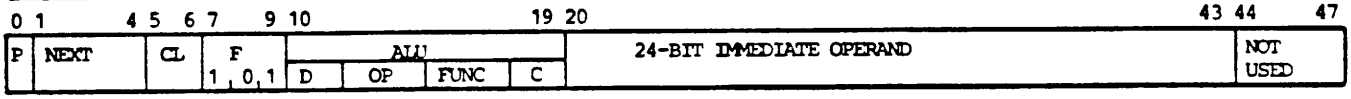

FORMAT 3: Read External Register



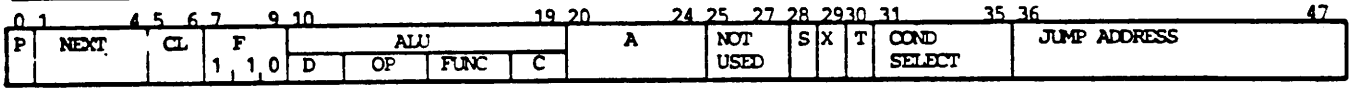

FORMAT 4: Half-Word Manipulator




FORMAT 5: Load Immediate




FORMAT 6: Conditional Jump




FORMAT 7: Shift, Multiply, Divide

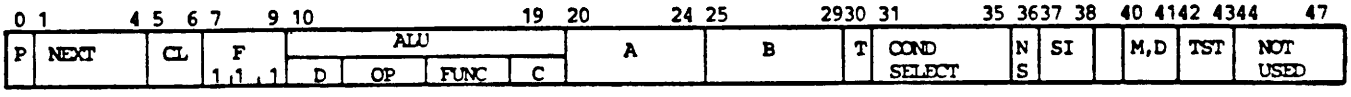



Figure 5: Microinstruction formats.

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

## 2.4.3 CL Field = MIR(5,6) 2.4.3

CL controls the microinstruction cycle length.

| CL | CYCLE LENGTH |
|---|---|
| 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).

| ALU-D | FUNCTION |
|---|---|
| 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.

| ALU-OP | OPERAND R | Operand S |
|---|---|---|
| 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.

| ALU-FUNC | ALU FUNCTION |
|---|---|
| 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.

| ALU-C | Cin |
|---|---|
| 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.

| MIR(20) | MIR(23,24) | GRA ADDRESS |
|---|---|---|
| 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.

| MIR(25) | MIR(28,29) | GRB ADDRESS |
|---|---|---|
| 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 independant of the NEXT field. The address is generated by a priority encoder and the condition corresponding to CS location 0 has the highest priority.

| CS LOCATION | INTR. CONDITION | MASK BIT |
|---|---|---|
| 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
OCP 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 REG DEST | CPU821 DESTINATION REGISTERS |
|---|---|
| 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). |
| | |
| | CAM801/803 DESTINATION REGISTERS |
| 40 | CAM Test Data |
| 41 | CAM Control |
| | |
| | FPU801 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 SELECT | SELECTED CONDITION | | |
|---|---|---|---|
| 00 | ZERO | | |
| 01 | NEG: | DBUS(0) | |
| 02 | NZERO: | F <> 0 | |
| 03 | OVFL: | arithm. overflow | |
| 04 | CARRY: | carry from ALU bit(0) | |
| 05 | SIGN: | F(0) exor OVR | 1) |
| 06 | NORM0: | DBUS(0) <> DBUS(1) | |
| 07 | NORM1: | DBUS(1) <> DBUS(2) | |
| 10 | DIVOVFL: | SHLINK <> 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 shiftet out by a shift microinstruction.

The conditions: NEG, NZERO, OVFL, CARRY, SIGN, SHLINK, NORM0, 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 = -,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

| SOURCE | SOURCE REGISTER/FUNCTION |
|---|---|
| 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 ADDR | CONSTANT, octal |
|---|---|
| 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 ADDR | CONSTANT, octal |
|---|---|
| 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 SOURCE | CPU822 SOURCE REGISTERS |
|---|---|
| 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 | CAM801/803 - 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 LASTAT register.

MANIP.

| FUNCTION | ODD | SBUS(0:11) | SBUS(12:23) |
|---|---|---|---|
| 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) | DUBS(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.

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

| NS | ALU-D | FUNCTION |
|---|---|---|
| 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.

| SI | SHIN, right shift | SHIN, left shift |
|---|---|---|
| 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.

| M | ALU-OP | ADDCOND | OPERAND R | OPERAND S |
|---|---|---|---|---|
| 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.

| D | ALU-FUNC | ADDCOND | ALU FUNCTION |
|---|---|---|---|
| 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) |

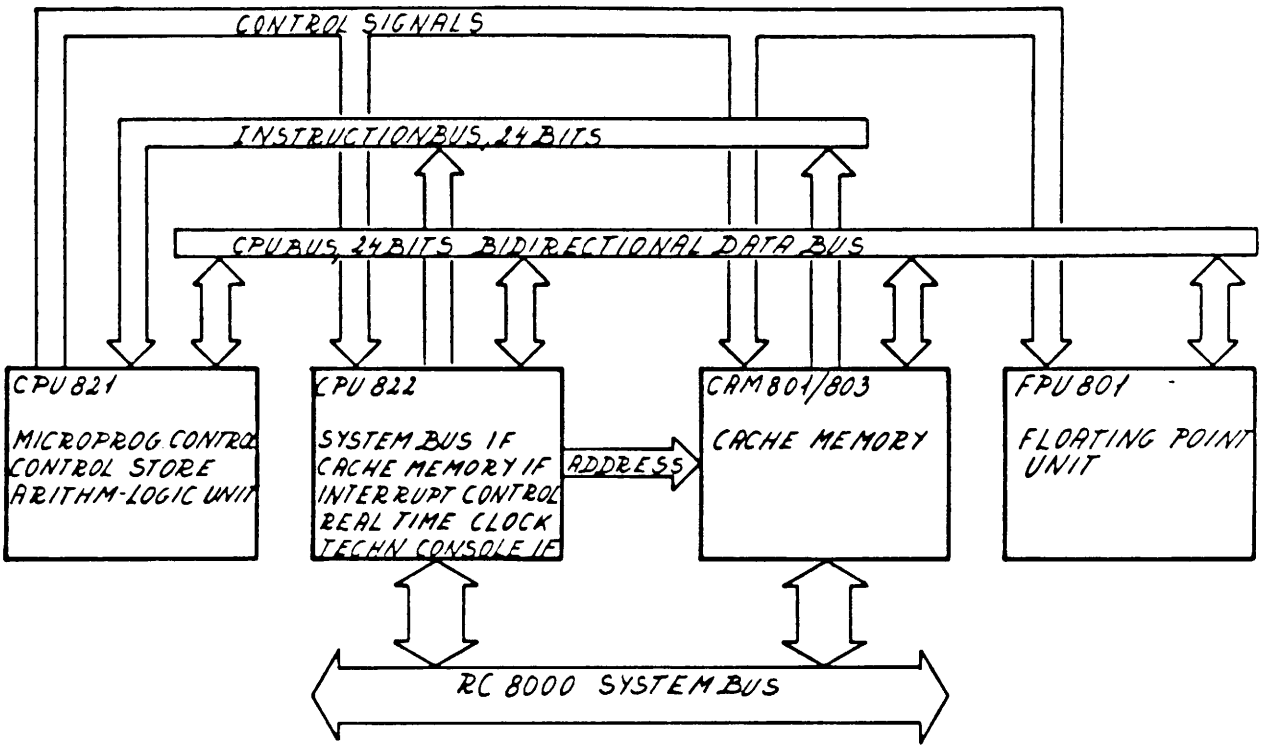
The TST field, MIR(42,43), controls the conditions, ADDCOND and DIVCOND. These conditions are only affected by format 7 microinstructions.

| TST | FUNCTION |
|---|---|
| 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.

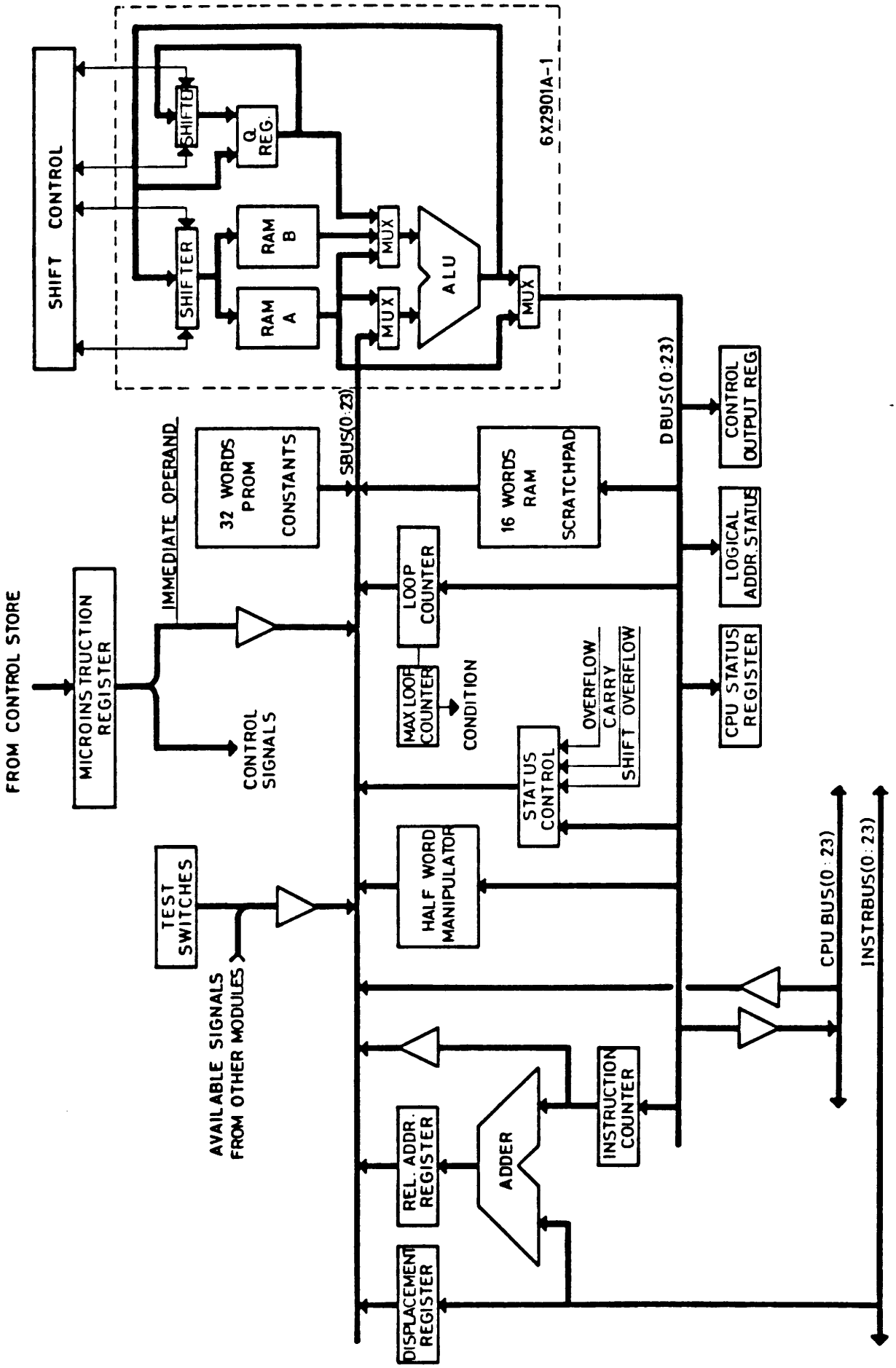
Set SHIFTOVFL = FORMAT 7 & (DBUS(0) exor DBUS(1))

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

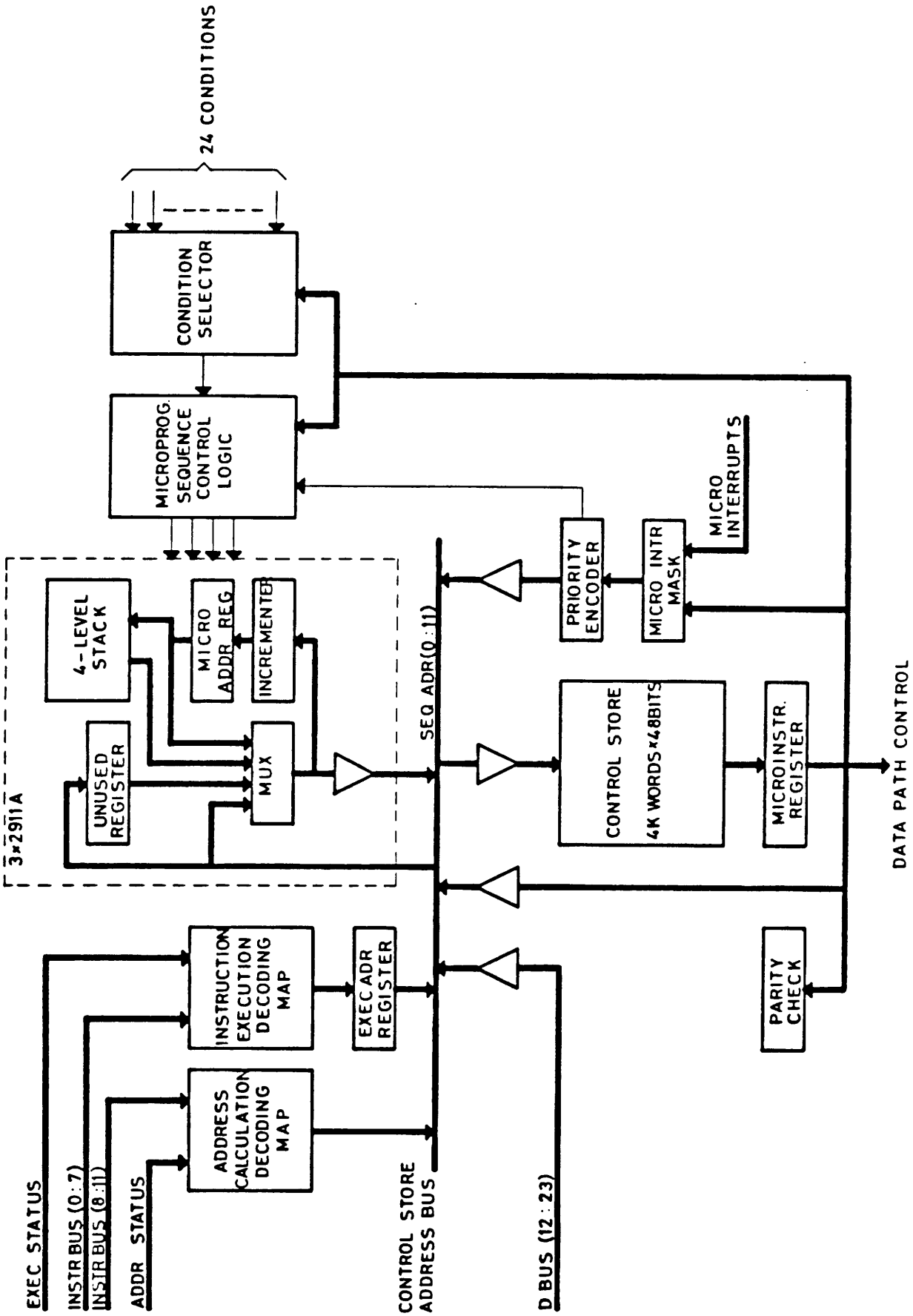



8106 11 AHJ 8106 15 AGA

CPU820 CPU 820 extended with CACHE MEMORY and FLOATING POINT UNIT



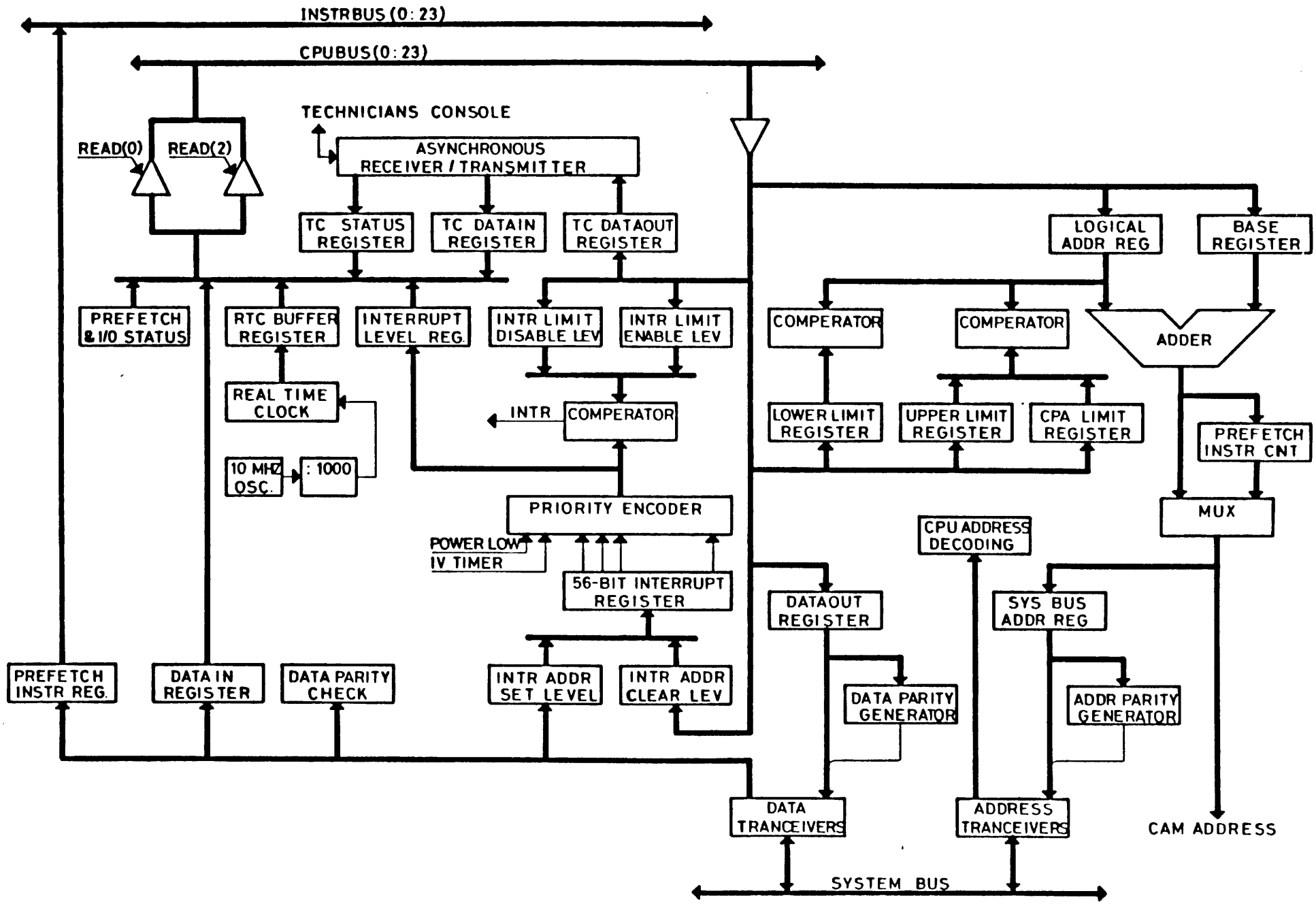

CPU 821 DATA PATHS

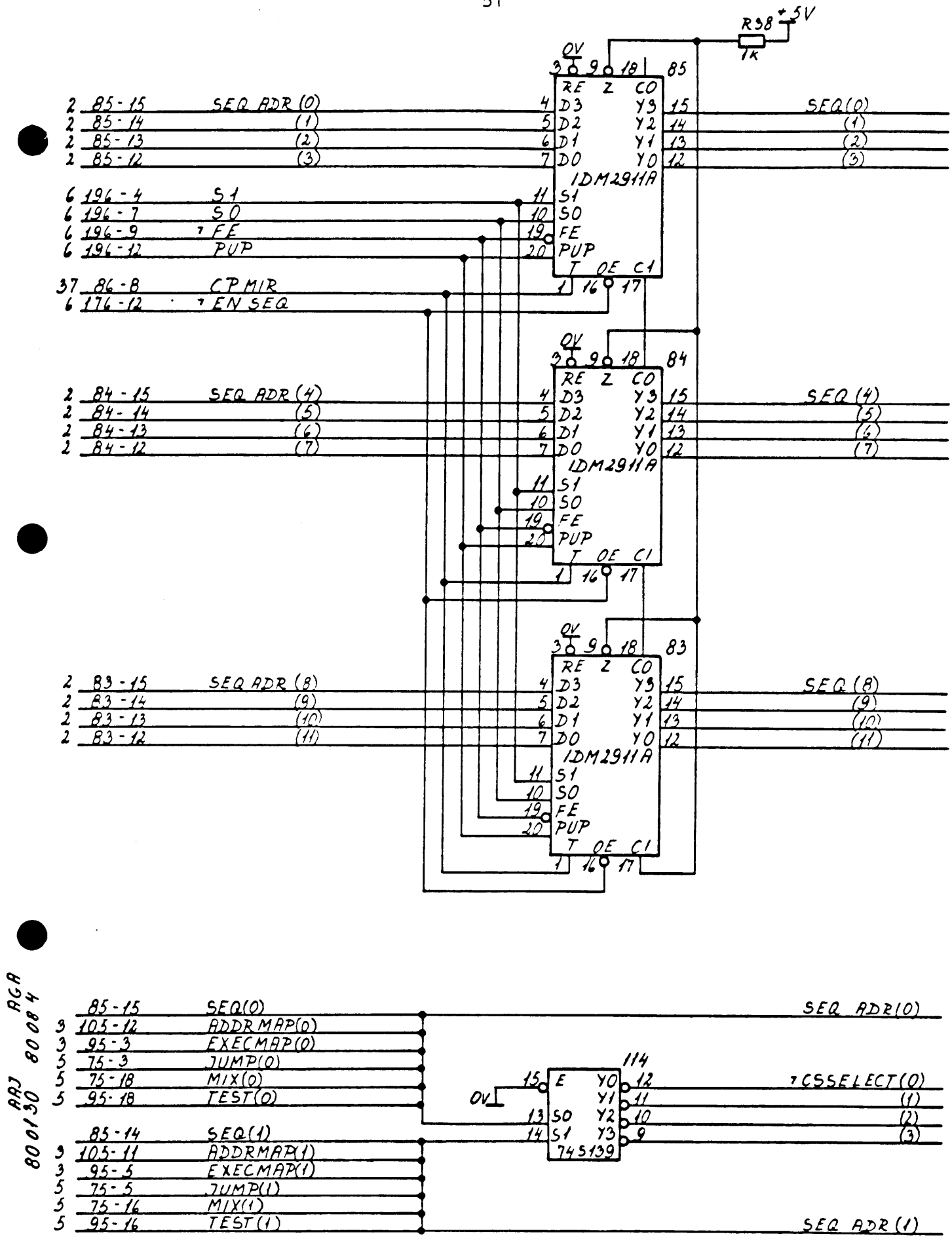





CONTROL STORE &
CONTROL STORE ADDRESS GENERATION

# CPU 822 DATA PATHS







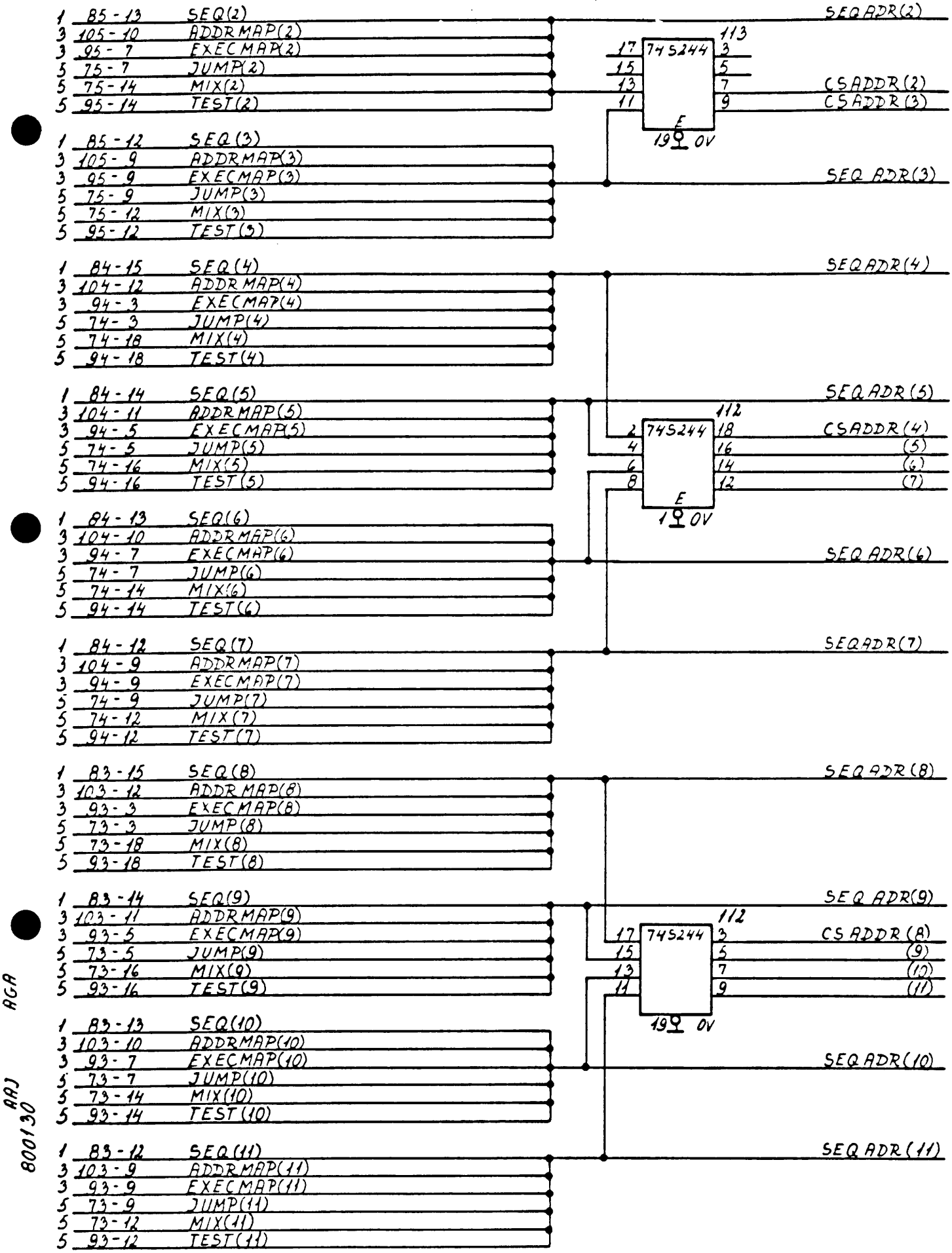


CPU 821

R 12922

CONTROL STORE ADDRESS GENERATION
MICROPROGRAM SEQUENCER

1




AGA

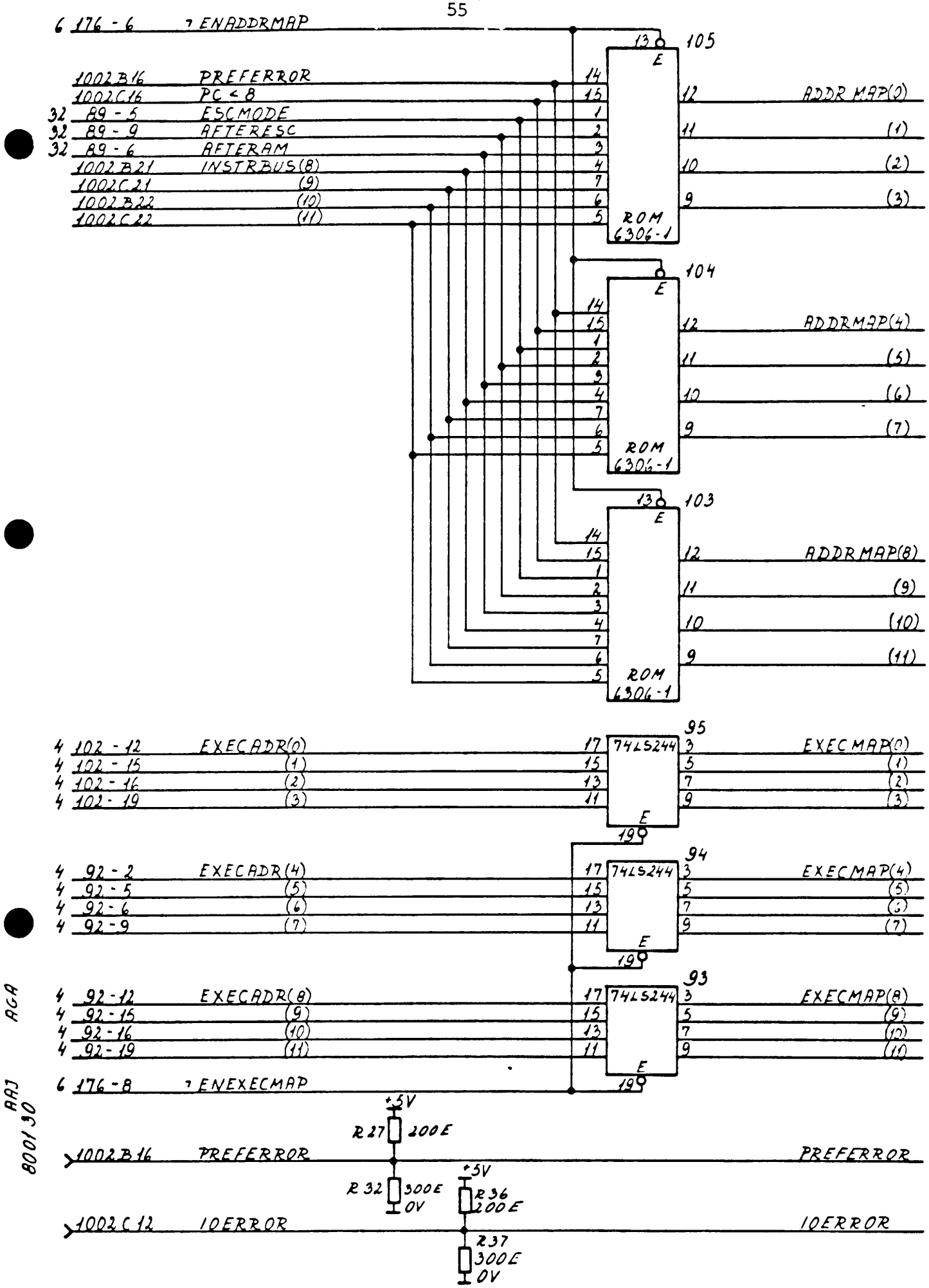
AAJ 800130

CPU 821

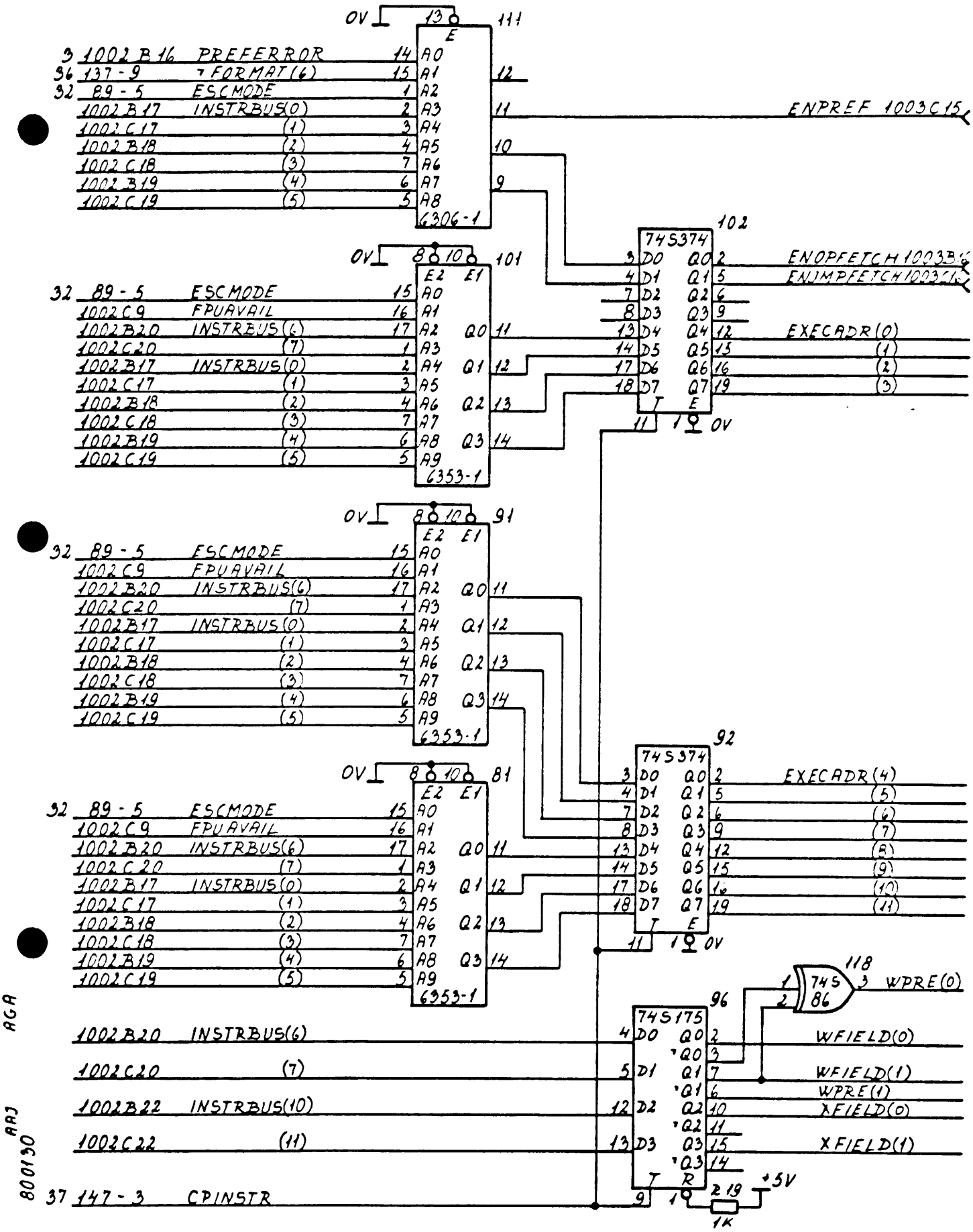
CONTROL STORE ADDRESS

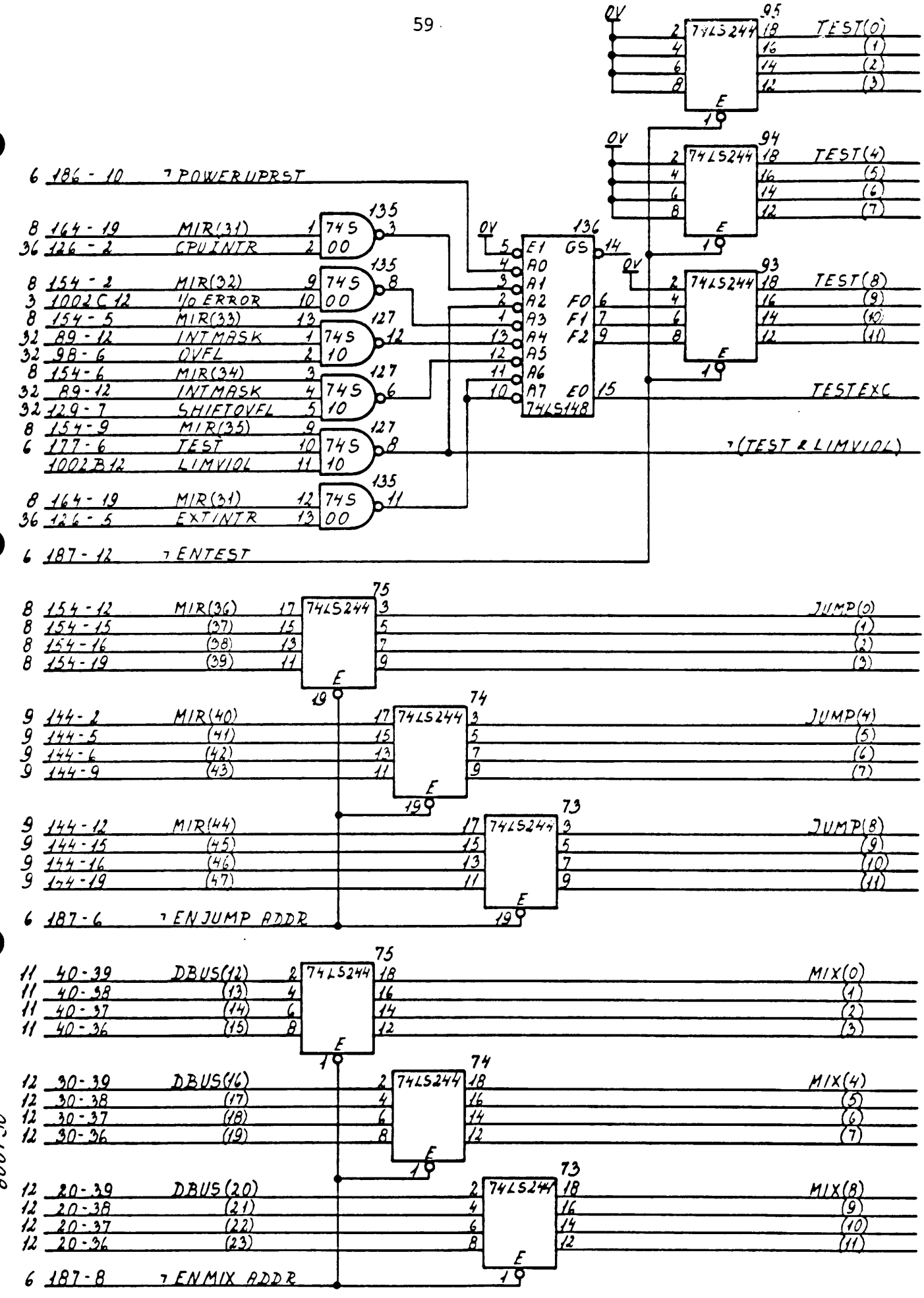
2

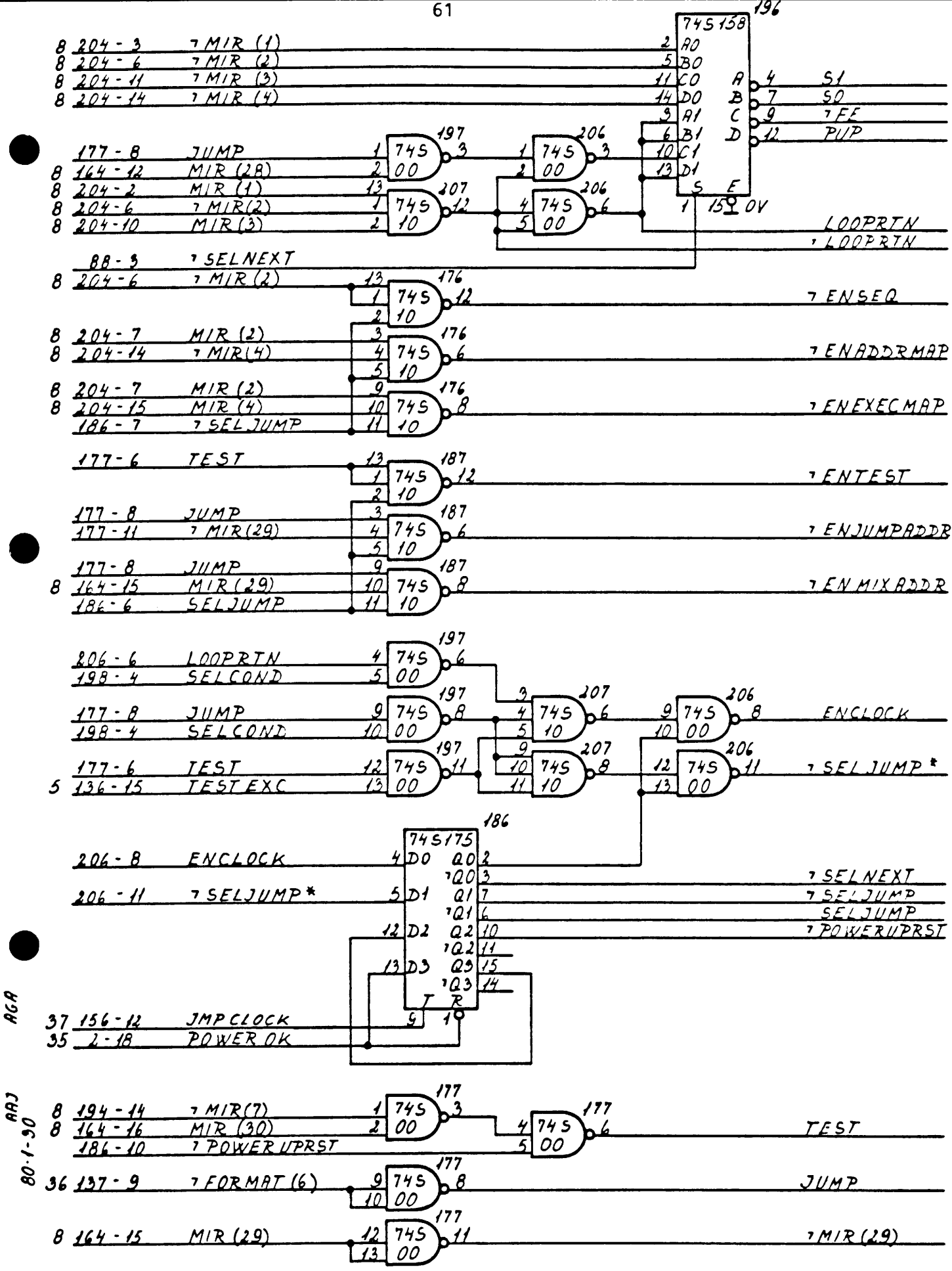
R 12923




AGA
AAJ
800130


111
6306-1
PREFERROR
FORMAT(6)
ESCMODE
INSTRBUS(0)
ENPREF 1003C15
101
6353-1
FPUAVAIL
INSTRBUS(6)
INSTRBUS(10)
CPINSTR
102
74S374
ENOPFETCH 1003B
ENIMPFETCH 1003C
EXECADR(0)
91
92
EXECADR(4)
81
118
74S86
WPRE(0)
96
74S175
WFIELD(0)
WFIELD(1)
WPRE(1)
XFIELD(0)
XFIELD(1)
+5V
1K
AGA
AAJ
800130


6 186-10 ¬POWERUPRST
8 164-19 MIR(31)
36 126-2 CPUINTR
8 154-2 MIR(32)
3 1002C12 I/O ERROR
8 154-5 MIR(33)
32 89-12 INTMASK
32 98-6 OVFL
8 154-6 MIR(34)
32 89-12 INTMASK
32 129-7 SHIFTOVFL
8 154-9 MIR(35)
6 177-6 TEST
1002B12 LIMVIOL
8 164-19 MIR(31)
36 126-5 EXTINTR
6 187-12 ¬ENTEST
74LS00
74LS148
74LS244
TEST(0)
TEST(4)
TEST(8)
TESTEXC
¬(TEST & LIMVIOL)
8 154-12 MIR(36)
8 154-15 (37)
8 154-16 (38)
8 154-19 (39)
9 144-2 MIR(40)
9 144-5 (41)
9 144-6 (42)
9 144-9 (43)
9 144-12 MIR(44)
9 144-15 (45)
9 144-16 (46)
9 144-19 (47)
6 187-6 ¬ENJUMP ADDR
JUMP(0)
JUMP(4)
JUMP(8)
11 40-39 DBUS(12)
11 40-38 (13)
11 40-37 (14)
11 40-36 (15)
12 30-39 DBUS(16)
12 30-38 (17)
12 30-37 (18)
12 30-36 (19)
12 20-39 DBUS(20)
12 20-38 (21)
12 20-37 (22)
12 20-36 (23)
6 187-8 ¬ENMIX ADDR
MIX(0)
MIX(4)
MIX(8)
AGA
ADJ 80013.0

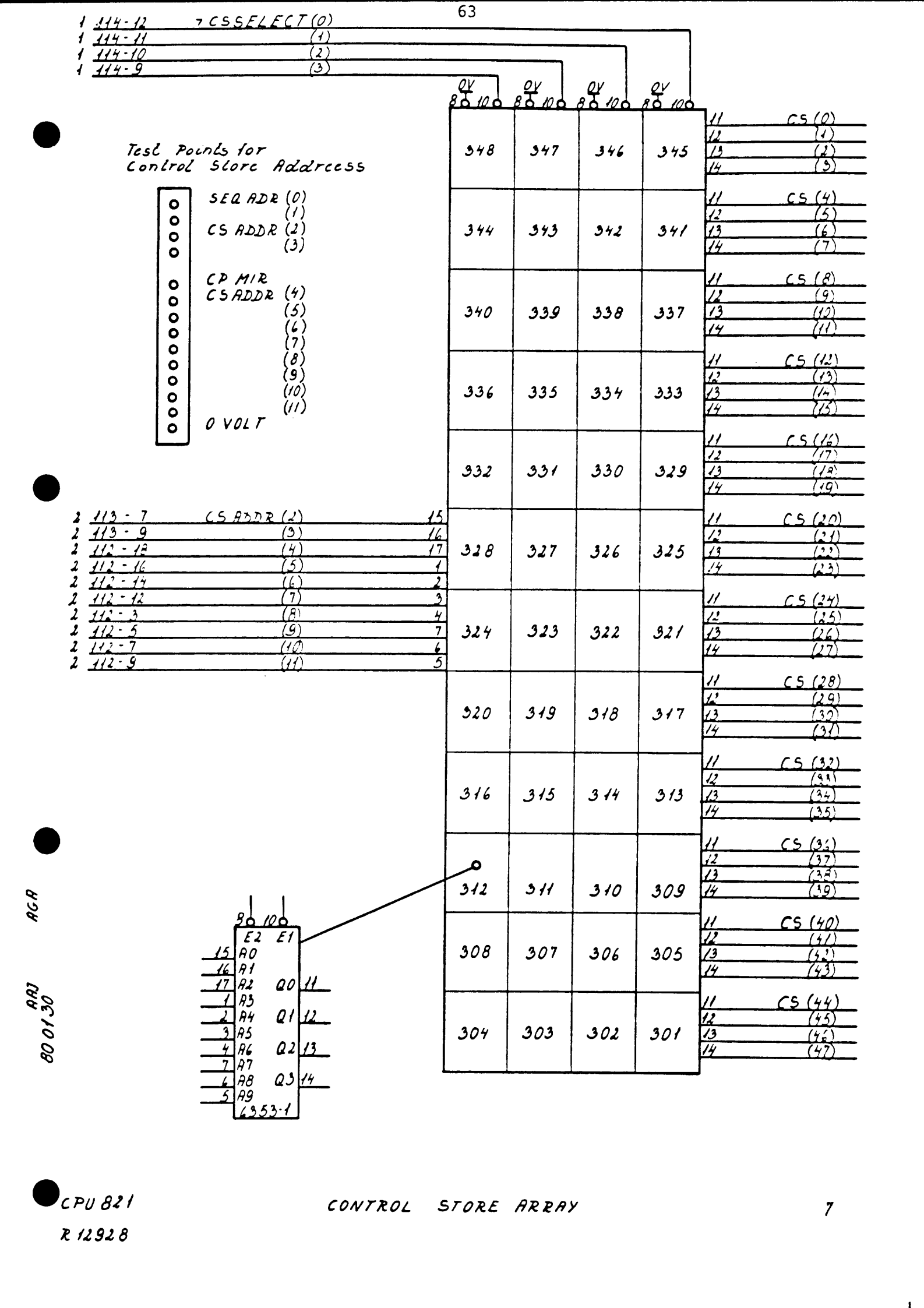



CPU 821

CONTROL STORE ADDRESS SELECTION

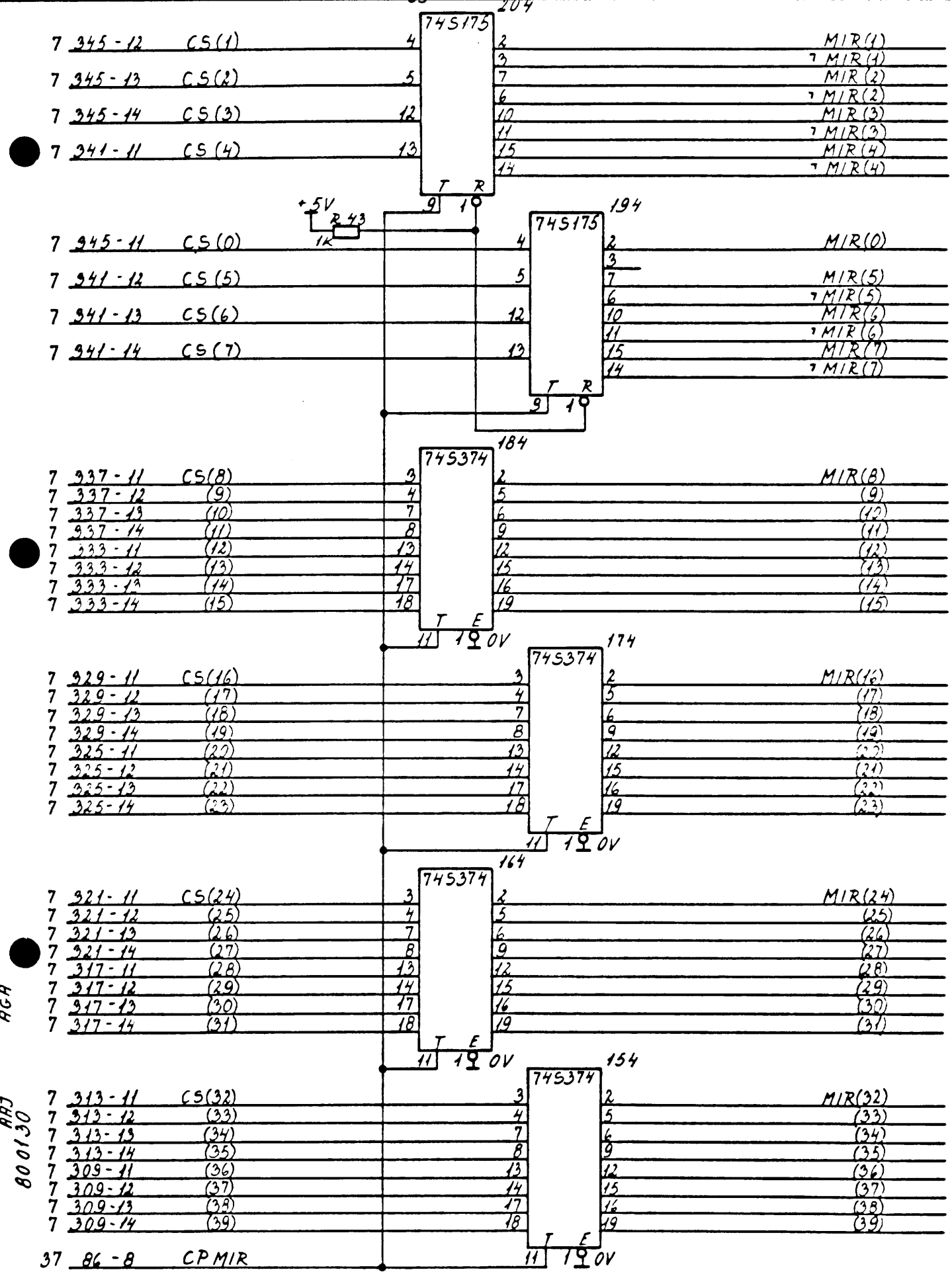
6

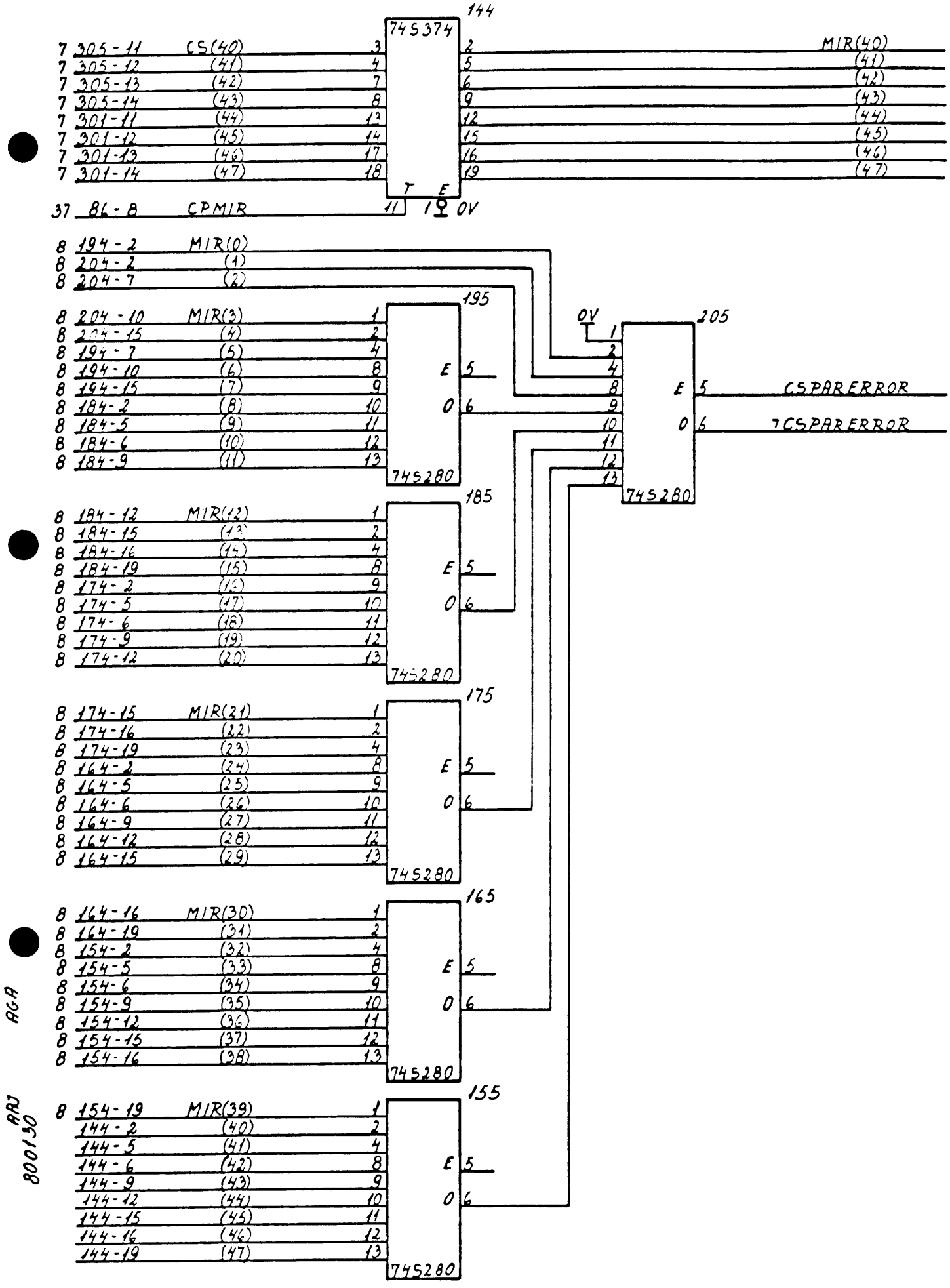
R 12927

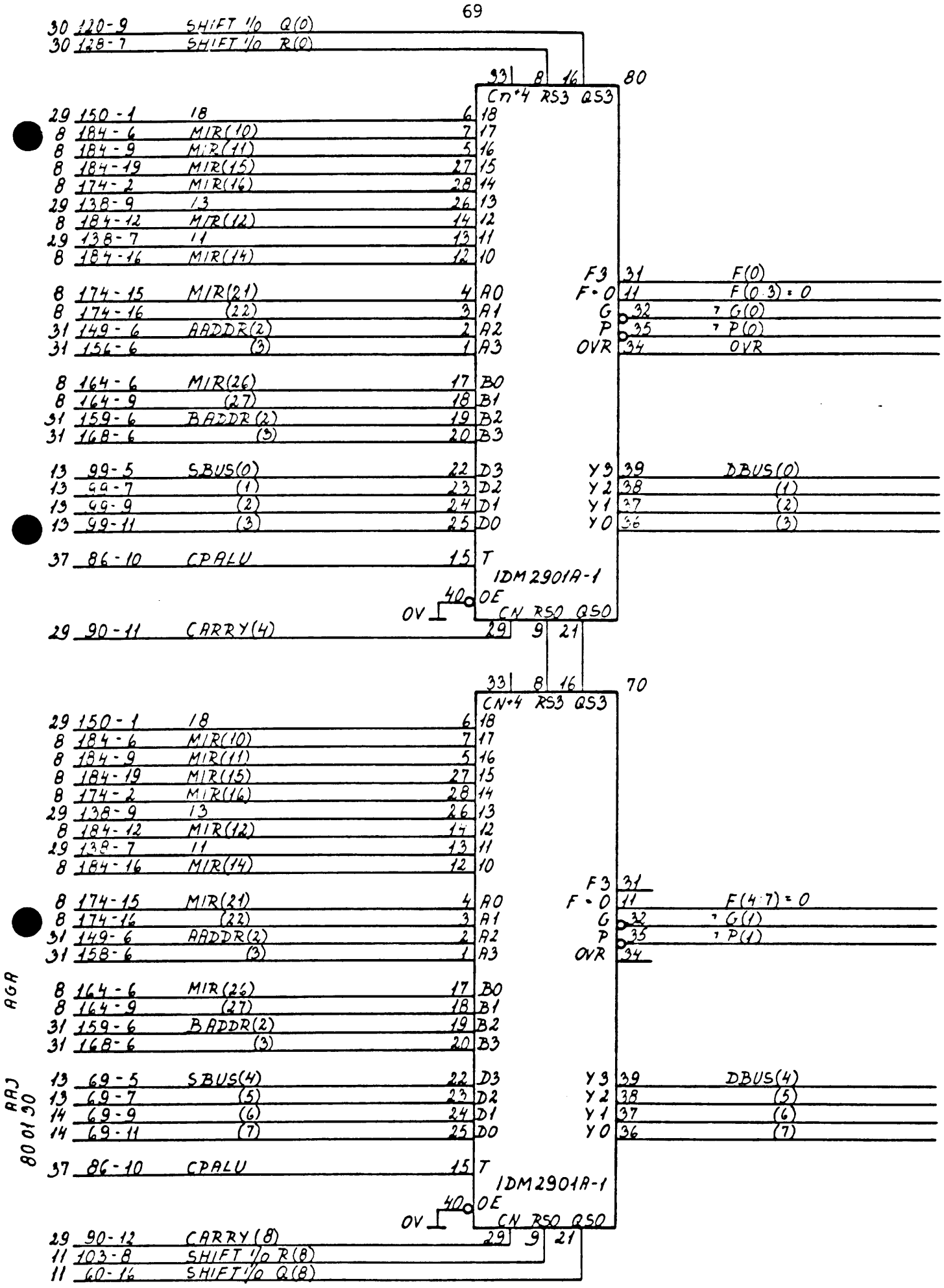


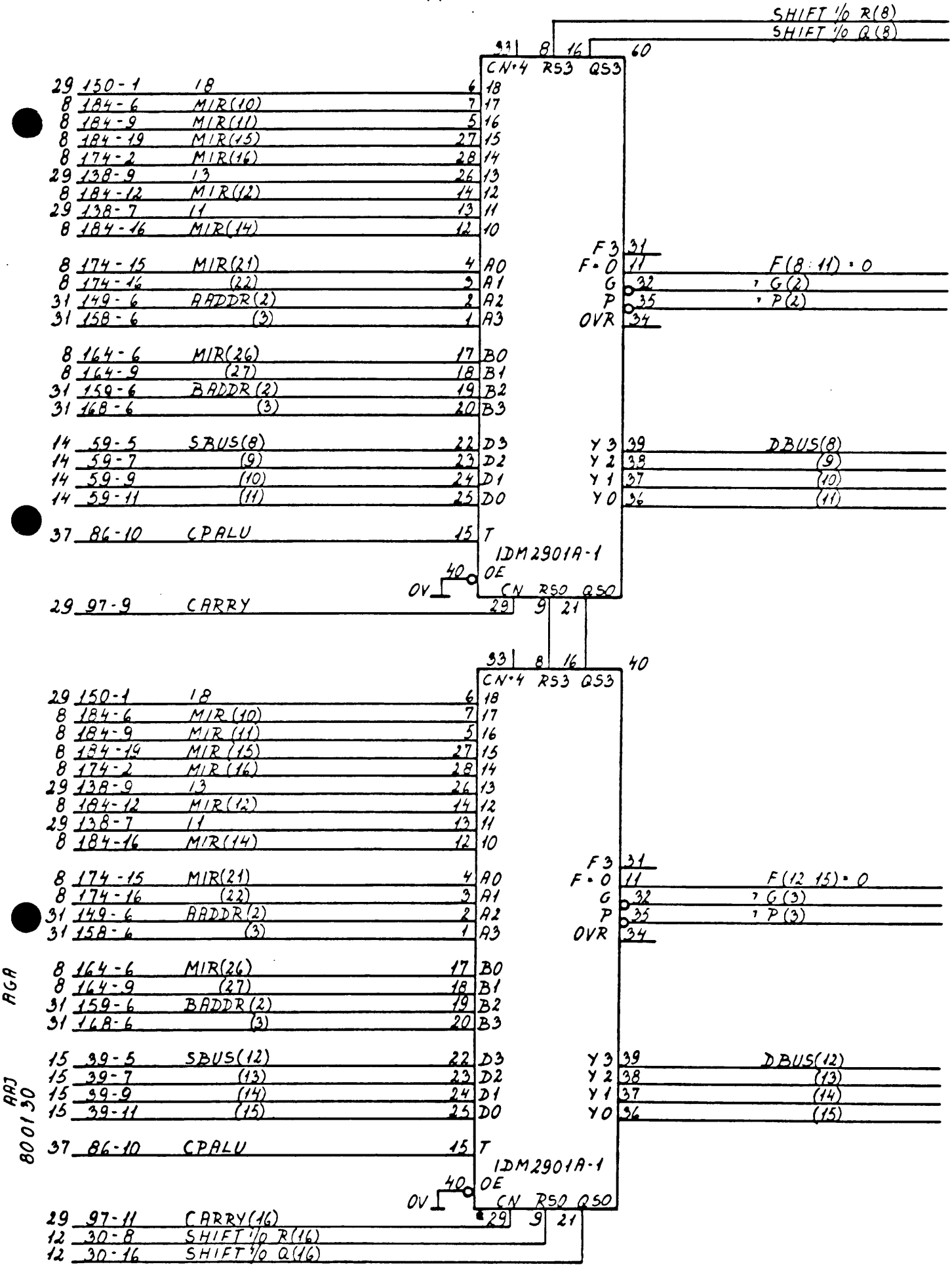

AGA

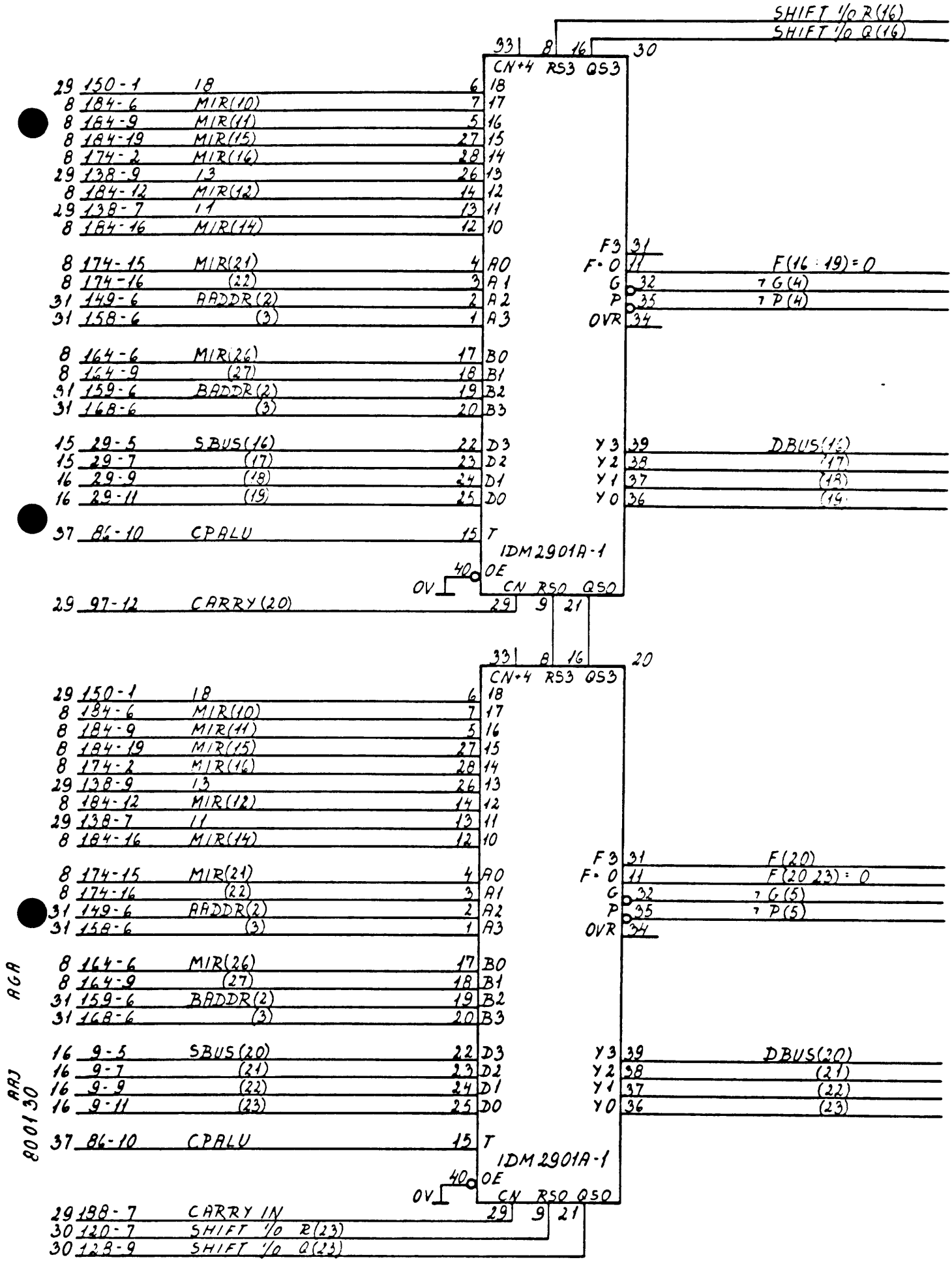
AAJ 80130

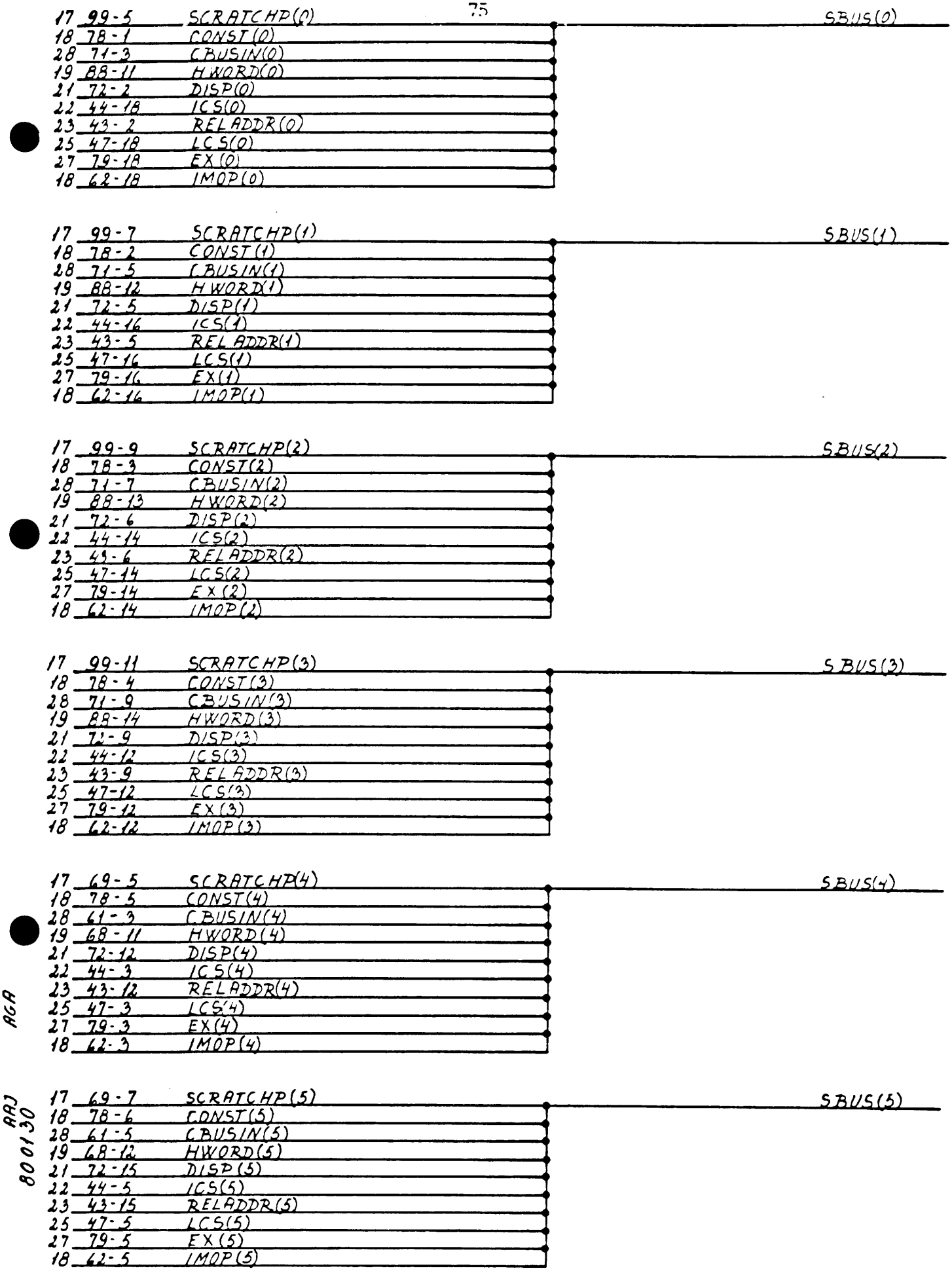

204
745175
7 345-12 CS(1)
7 345-13 CS(2)
7 345-14 CS(3)
7 341-11 CS(4)
MIR(1)
MIR(2)
MIR(3)
MIR(4)
+5V
R43
1K
194
745175
7 945-11 CS(0)
7 941-12 CS(5)
7 941-13 CS(6)
7 941-14 CS(7)
MIR(0)
MIR(5)
MIR(6)
MIR(7)
184
745374
7 937-11 CS(8)
7 337-12 (9)
7 337-13 (10)
7 937-14 (11)
7 333-11 (12)
7 333-12 (13)
7 333-13 (14)
7 333-14 (15)
MIR(8)
0V
174
745374
7 929-11 CS(16)
7 329-12 (17)
7 329-13 (18)
7 329-14 (19)
7 325-11 (20)
7 325-12 (21)
7 325-13 (22)
7 325-14 (23)
MIR(16)
0V
164
745374
7 321-11 CS(24)
7 321-12 (25)
7 321-13 (26)
7 321-14 (27)
7 317-11 (28)
7 317-12 (29)
7 317-13 (30)
7 317-14 (31)
MIR(24)
0V
154
745374
7 313-11 CS(32)
7 313-12 (33)
7 313-13 (34)
7 313-14 (35)
7 309-11 (36)
7 309-12 (37)
7 309-13 (38)
7 309-14 (39)
MIR(32)
0V
37 86-8 CPMIR
AGA
ARJ 80 01 30

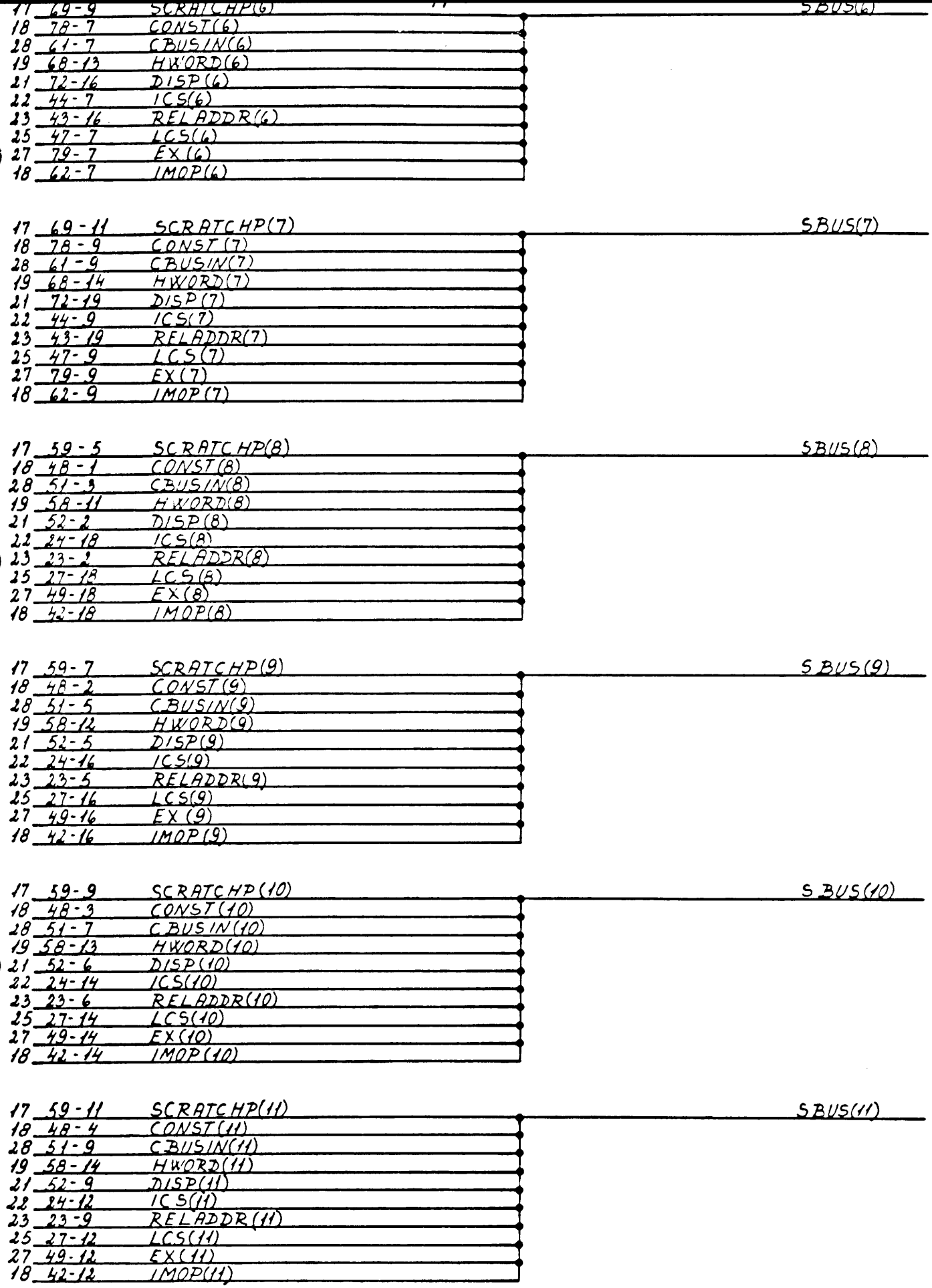

144
74S374
7 305-11 CS(40) 3 2 MIR(40)
7 305-12 (41) 4 5 (41)
7 305-13 (42) 7 6 (42)
7 305-14 (43) 8 9 (43)
7 301-11 (44) 13 12 (44)
7 301-12 (45) 14 15 (45)
7 301-13 (46) 17 16 (46)
7 301-14 (47) 18 19 (47)
T E
37 86-8 CPMIR 11 1 0V
8 194-2 MIR(0)
8 204-2 (1)
8 204-7 (2)
195
8 204-10 MIR(3) 1
8 204-15 (4) 2
8 194-7 (5) 4
8 194-10 (6) 8
8 194-15 (7) 9
8 184-2 (8) 10
8 184-5 (9) 11
8 184-6 (10) 12
8 184-9 (11) 13
E 5
O 6
74S280
0V
205
1
2
4
8
9
10
11
12
13
E 5 CSPARERROR
O 6 ¬CSPARERROR
74S280
185
8 184-12 MIR(12) 1
8 184-15 (13) 2
8 184-16 (14) 4
8 184-19 (15) 8
8 174-2 (16) 9
8 174-5 (17) 10
8 174-6 (18) 11
8 174-9 (19) 12
8 174-12 (20) 13
E 5
O 6
74S280
175
8 174-15 MIR(21) 1
8 174-16 (22) 2
8 174-19 (23) 4
8 164-2 (24) 8
8 164-5 (25) 9
8 164-6 (26) 10
8 164-9 (27) 11
8 164-12 (28) 12
8 164-15 (29) 13
E 5
O 6
74S280
165
8 164-16 MIR(30) 1
8 164-19 (31) 2
8 154-2 (32) 4
8 154-5 (33) 8
8 154-6 (34) 9
8 154-9 (35) 10
8 154-12 (36) 11
8 154-15 (37) 12
8 154-16 (38) 13
E 5
O 6
74S280
155
8 154-19 MIR(39) 1
144-2 (40) 2
144-5 (41) 4
144-6 (42) 8
144-9 (43) 9
144-12 (44) 10
144-15 (45) 11
144-16 (46) 12
144-19 (47) 13
E 5
O 6
74S280
AGA
ARJ
800130


30 120-9 SHIFT I/O Q(0)
30 128-7 SHIFT I/O 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
F3 31 F(0)
F=0 11 F(0:3)=0
G 32 ▽G(0)
P 35 ▽P(0)
OVR 34 OVR
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
Y3 39 DBUS(0)
Y2 38 (1)
Y1 37 (2)
Y0 36 (3)
37 86-10 CPALU 15 T
IDM2901A-1
OV 40 OE
CN RS0 QS0
29 90-11 CARRY(4) 29 9 21
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
F3 31
F=0 11 F(4:7)=0
G 32 ▽G(1)
P 35 ▽P(1)
OVR 34
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
Y3 39 DBUS(4)
Y2 38 (5)
Y1 37 (6)
Y0 36 (7)
37 86-10 CPALU 15 T
IDM2901A-1
OV 40 OE
CN RS0 QS0
29 90-12 CARRY(8) 29 9 21
11 103-8 SHIFT I/O R(8)
11 60-16 SHIFT I/O Q(8)
AGA
RAJ
80 01 30


SHIFT I/O R(8)
SHIFT I/O Q(8)
CN+4 RS3 QS3
IDM2901A-1
CN RS0 QS0
OE
OV
CPALU
CARRY
MIR(10)
MIR(11)
MIR(15)
MIR(16)
MIR(12)
MIR(14)
MIR(21)
MIR(26)
AADDR(2)
BADDR(2)
SBUS(8)
SBUS(12)
DBUS(8)
DBUS(12)
F(8:11)=0
F(12:15)=0
CARRY(16)
SHIFT I/O R(16)
SHIFT I/O Q(16)
AGA
AAJ
8001 30


SHIFT I/O R(16)
SHIFT I/O Q(16)
CN+4 RS3 QS3
IDM2901A-1
OE
CN RSO QSO
CPALU
CARRY(20)
SBUS(16)
MIR(10)
MIR(11)
MIR(15)
MIR(16)
MIR(12)
MIR(14)
MIR(21)
MIR(26)
BADDR(2)
DBUS(16)
F(16:19)=0
F(20)
F(20:23)=0
SBUS(20)
DBUS(20)
CARRY IN
SHIFT I/O R(23)
SHIFT I/O Q(23)


17 99-5 SCRATCHP(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)
SBUS(0)
17 99-7 SCRATCHP(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)
SBUS(1)
17 99-9 SCRATCHP(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)
SBUS(2)
17 99-11 SCRATCHP(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)
SBUS(3)
17 69-5 SCRATCHP(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)
SBUS(4)
AGA
17 69-7 SCRATCHP(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)
SBUS(5)
AAJ
80130


17 69-9 SCRATCHP(6)
18 78-7 CONST(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)
SBUS(6)
17 69-11 SCRATCHP(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)
SBUS(7)
17 59-5 SCRATCHP(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)
SBUS(8)
17 59-7 SCRATCHP(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)
SBUS(9)
17 59-9 SCRATCHP(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)
SBUS(10)
17 59-11 SCRATCHP(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)
SBUS(11)
AGA
ARJ
80 01 30

| | | | |
|---|---|---|---|
| 17 | 39 - 5 | SCRATCHP(12) | SBUS(12) |
| 18 | 48 - 5 | CONST(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) | |

HGA

| | | | |
|---|---|---|---|
| 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) | |

AAJ 800130

| | | | |
|---|---|---|---|
| 17 | 29-9 | SCRATCHP(18) | SBUS(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) | |

| | | | |
|---|---|---|---|
| 17 | 29-11 | SCRATCHP(19) | SBUS(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) | |

| | | | |
|---|---|---|---|
| 17 | 9-5 | SCRATCHP(20) | SBUS(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 | 19-9 | EX(20) | |
| 18 | 22-3 | IMOP(20) | |
| 34 | 67-3 | SWITCH(20) | |

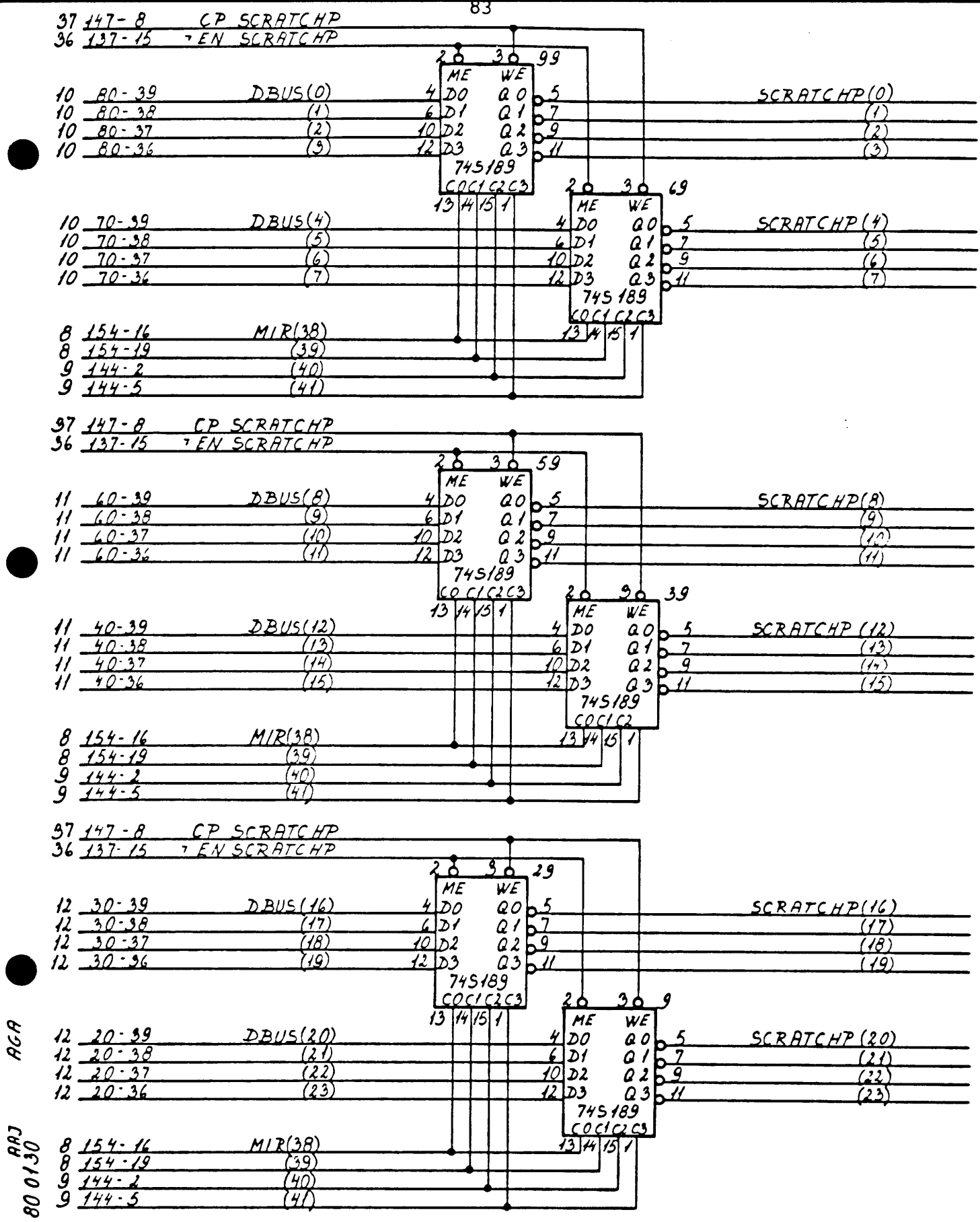
| | | | |
|---|---|---|---|
| 17 | 9-7 | SCRATCHP(21) | SBUS(21) |
| 18 | 18-6 | CONST(21) | |
| 28 | 21-5 | CBUSIN(21) | |
| 20 | 8-12 | HWORD(21) | |
| 21 | 32-15 | DISP(21) | |
| 22 | 4-5 | ICS(21) | |
| 24 | 3-15 | RELADDR(21) | |
| 25 | 7-5 | LCS(21) | |
| 27 | 19-5 | EX(21) | |
| 18 | 22-5 | IMOP(21)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

| | | | |
|---|---|---|---|
| 17 | 9-9 | SCRATCHP(22) | SBUS(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) | |

AGA

| | | | |
|---|---|---|---|
| 17 | 9-11 | SCRATCHP(23) | SBUS(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) | |

ARJ
80/130




AGA

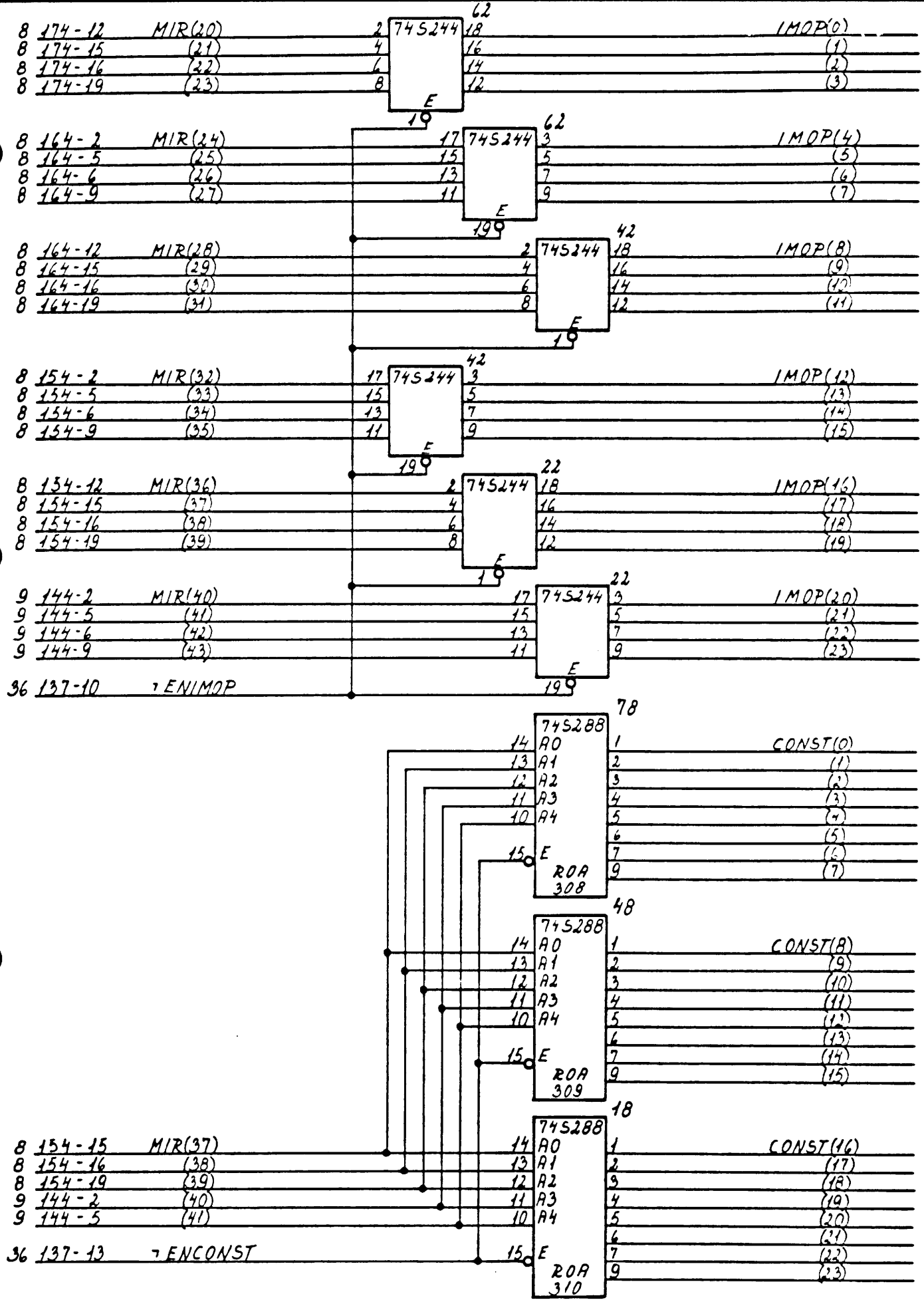
ANJ 80 0130

CPU 821

R 12938

SCRATCHPAD MEMORY, 16 WORDS x 24 BITS

17




AGA

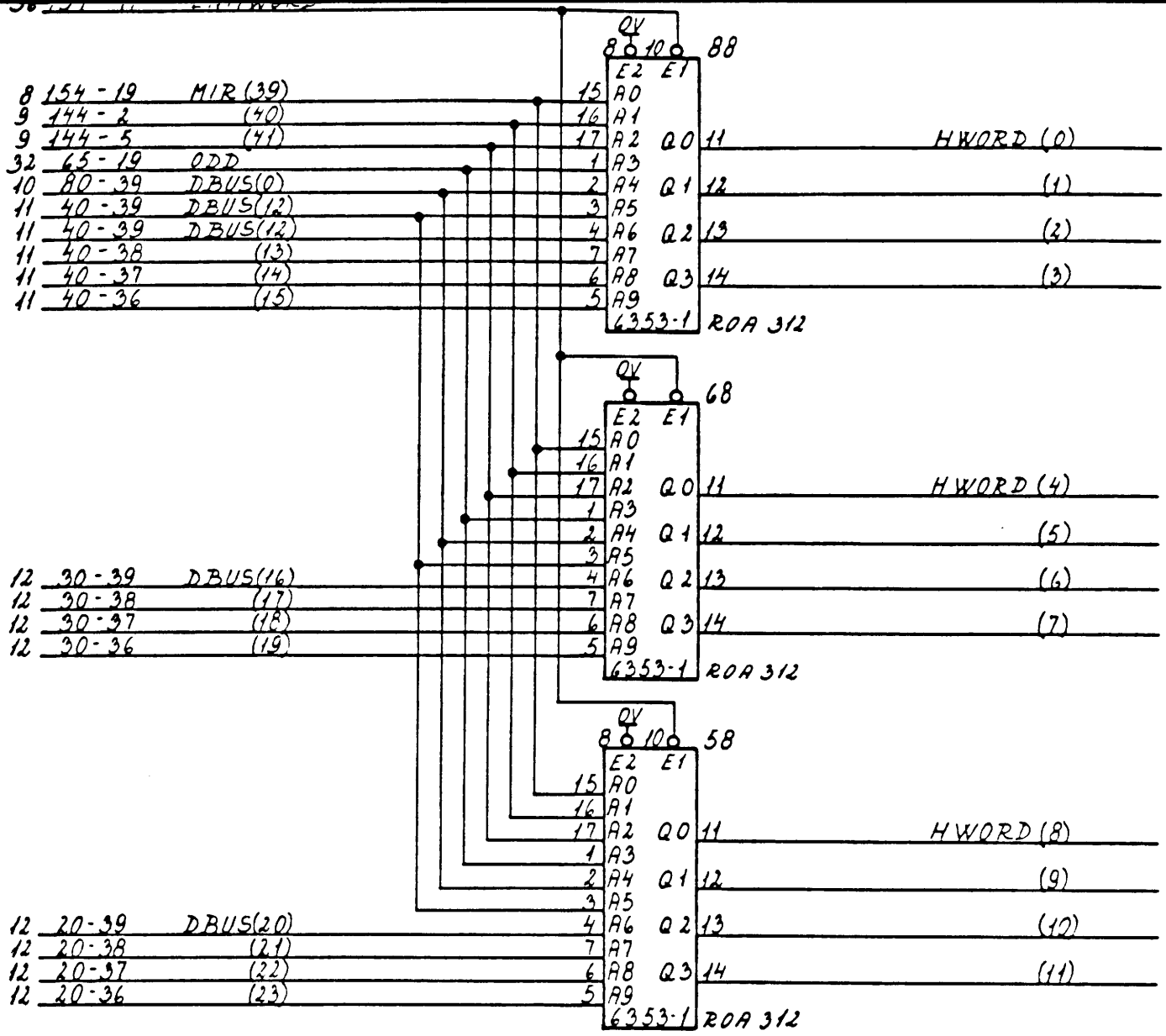
AAJ 80 01 30

CPU 821

IMMEDIATE OPERAND DRIVERS AND CONSTANT STORE

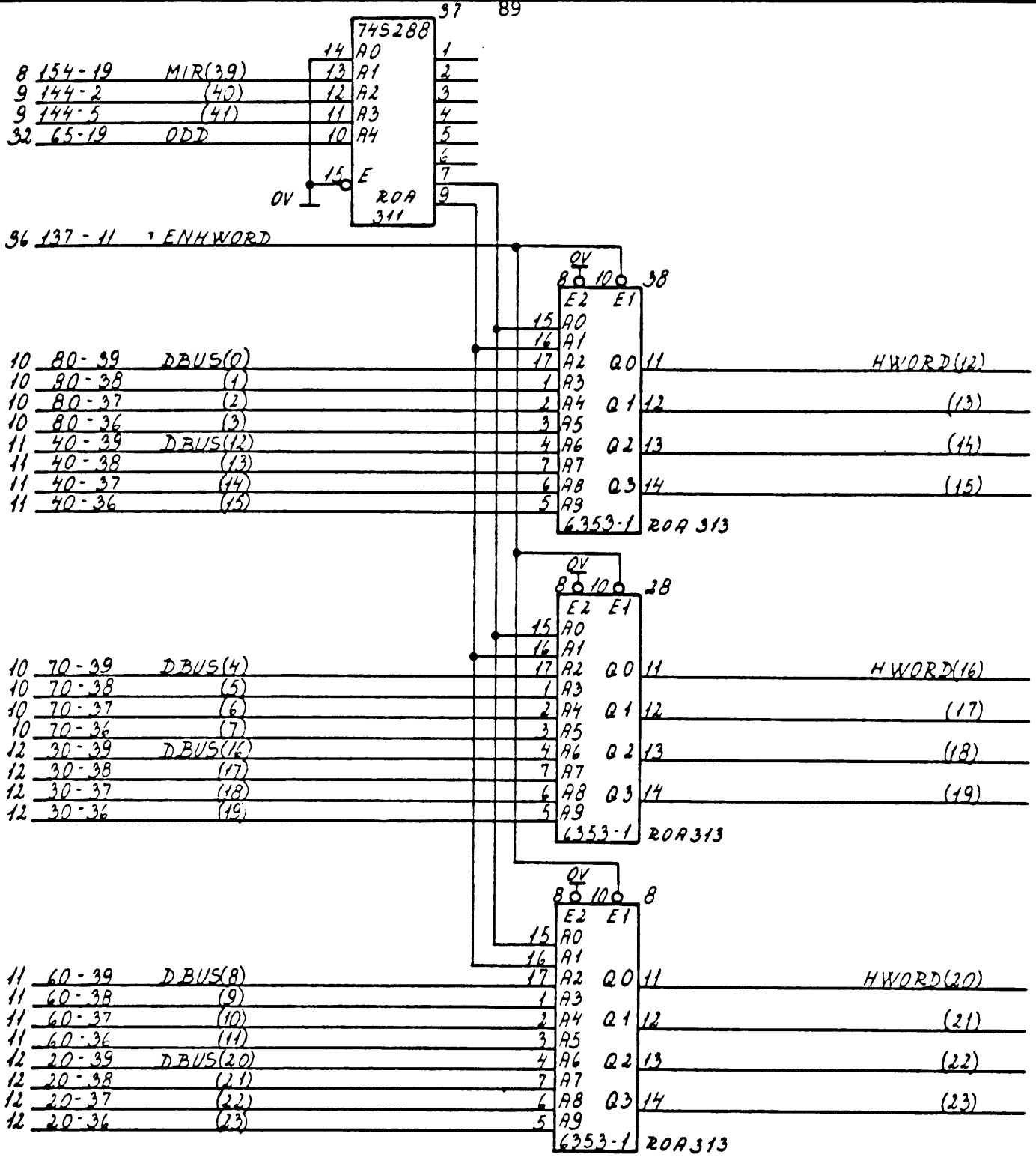
18

2 12 939




NGA

80 01 30 ARJ



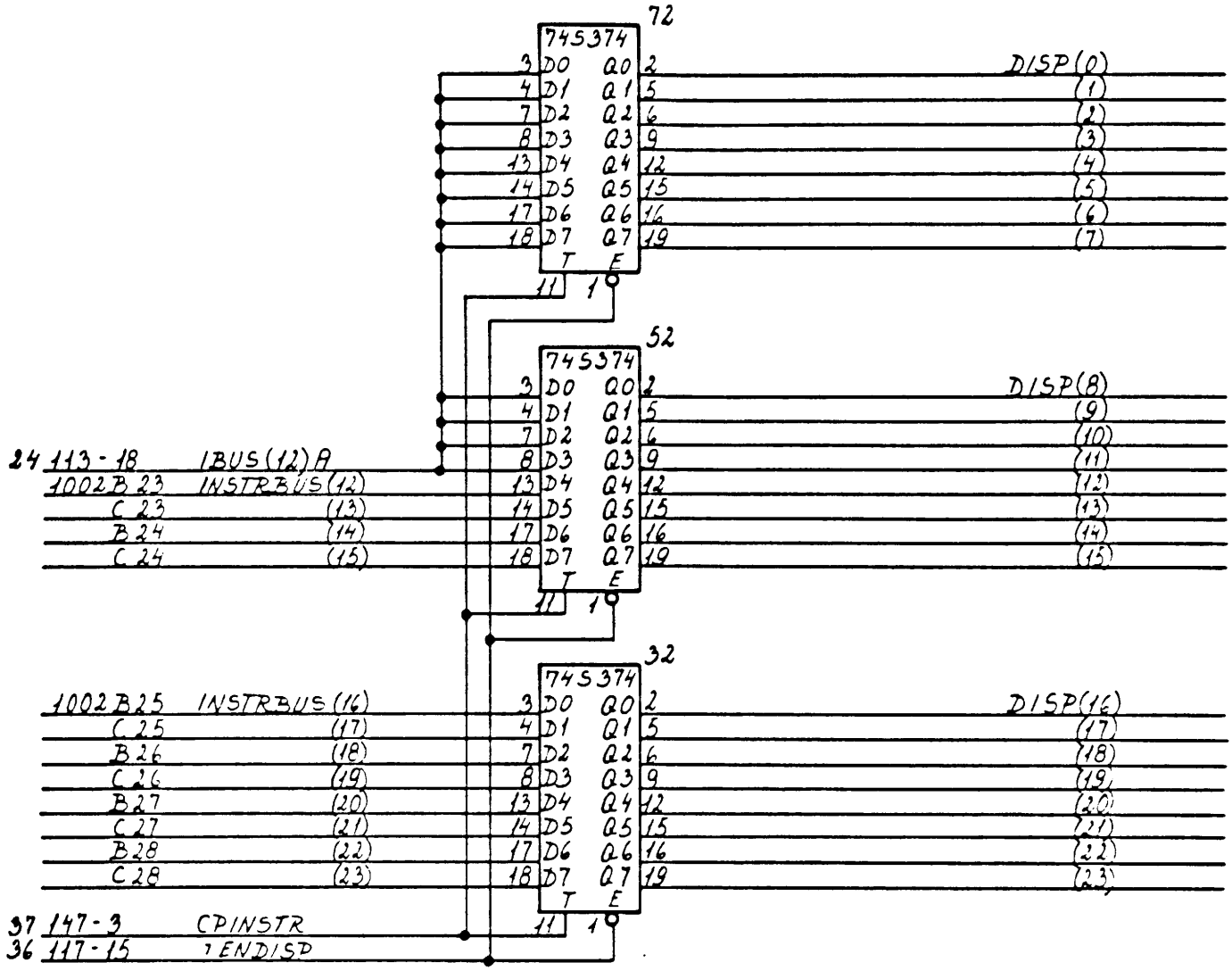

AGA

AAJ
80 01 30

CPU 821

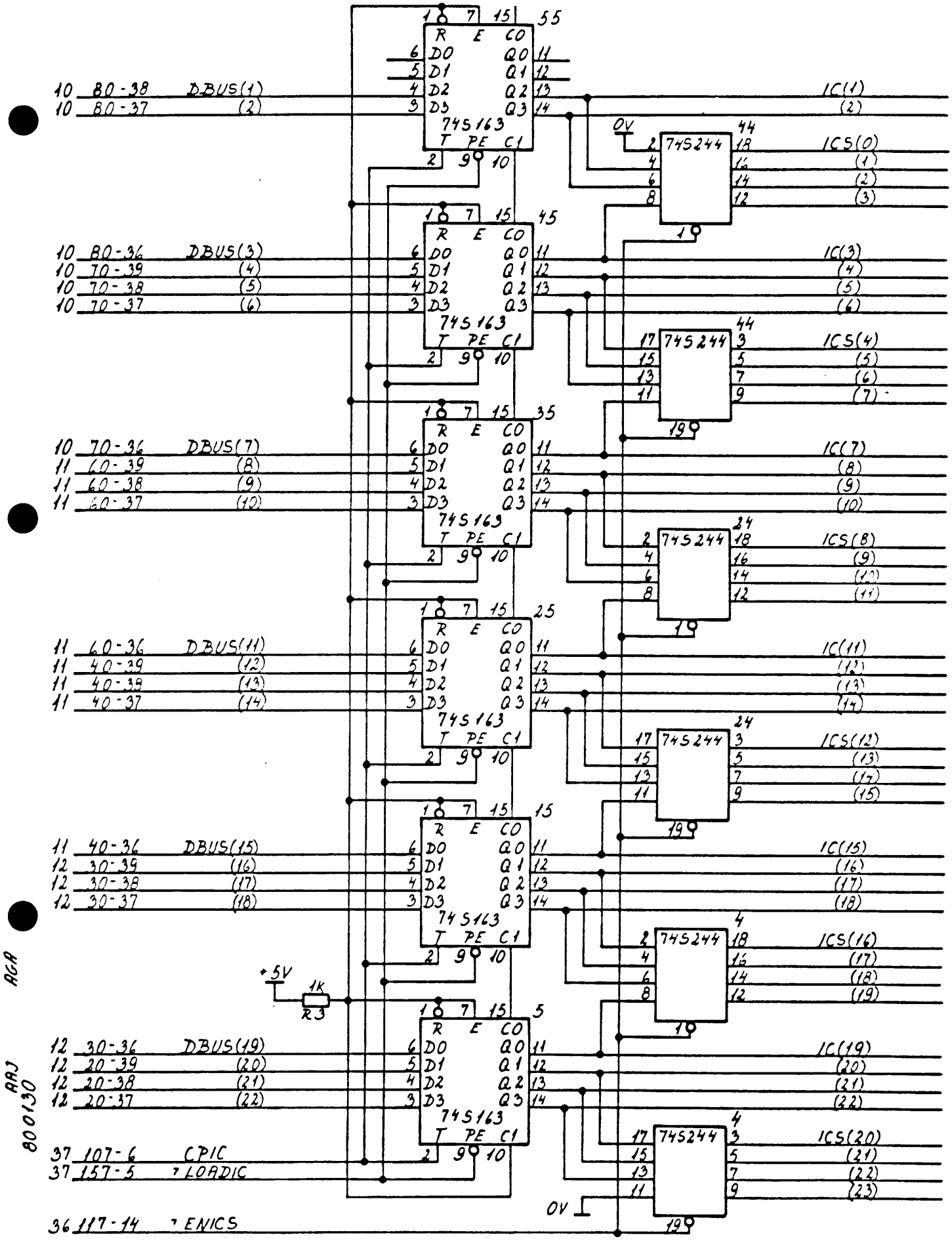
R 12941

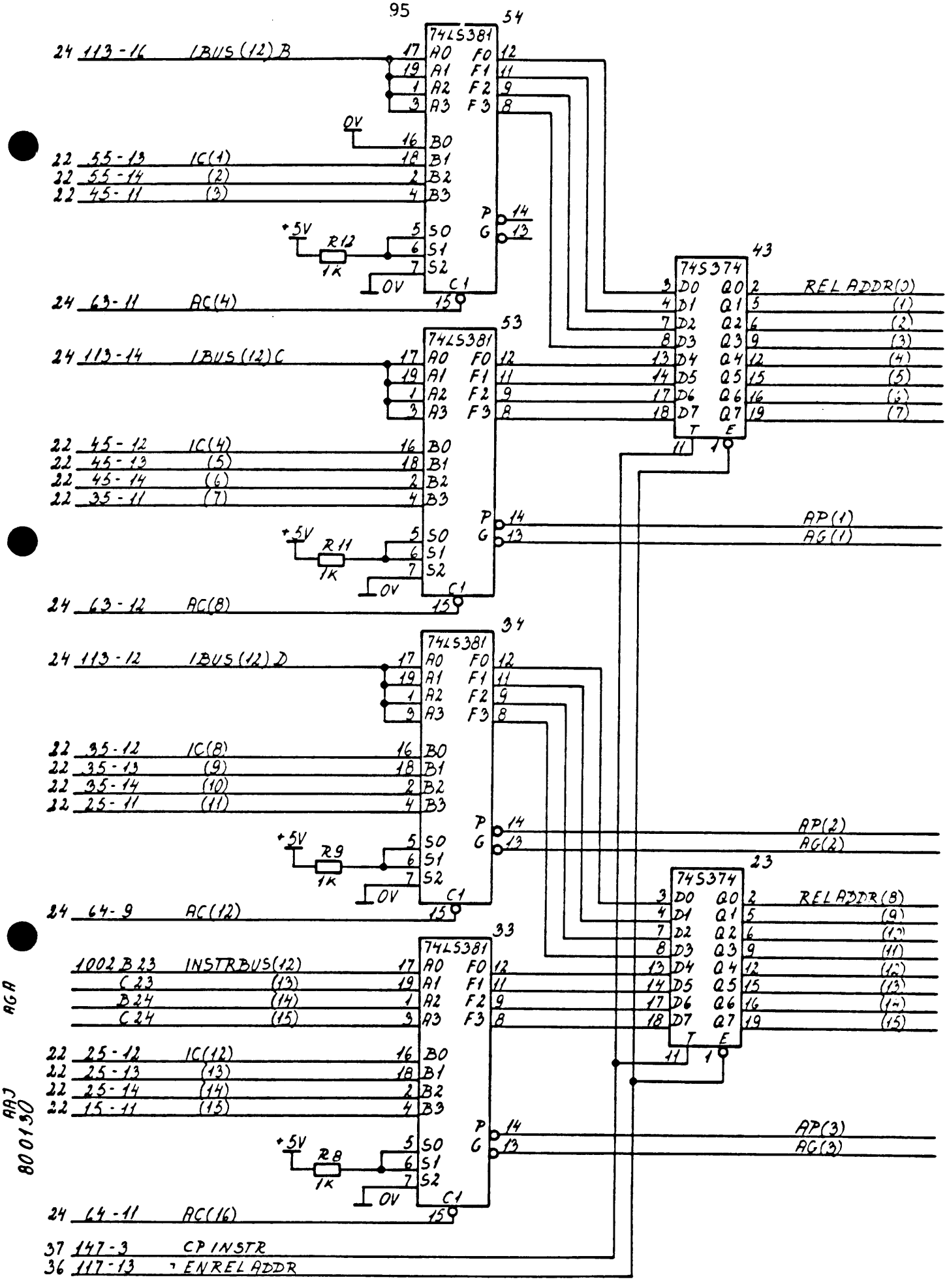
HALF-WORD MANIPULATOR, BITS (12 23)

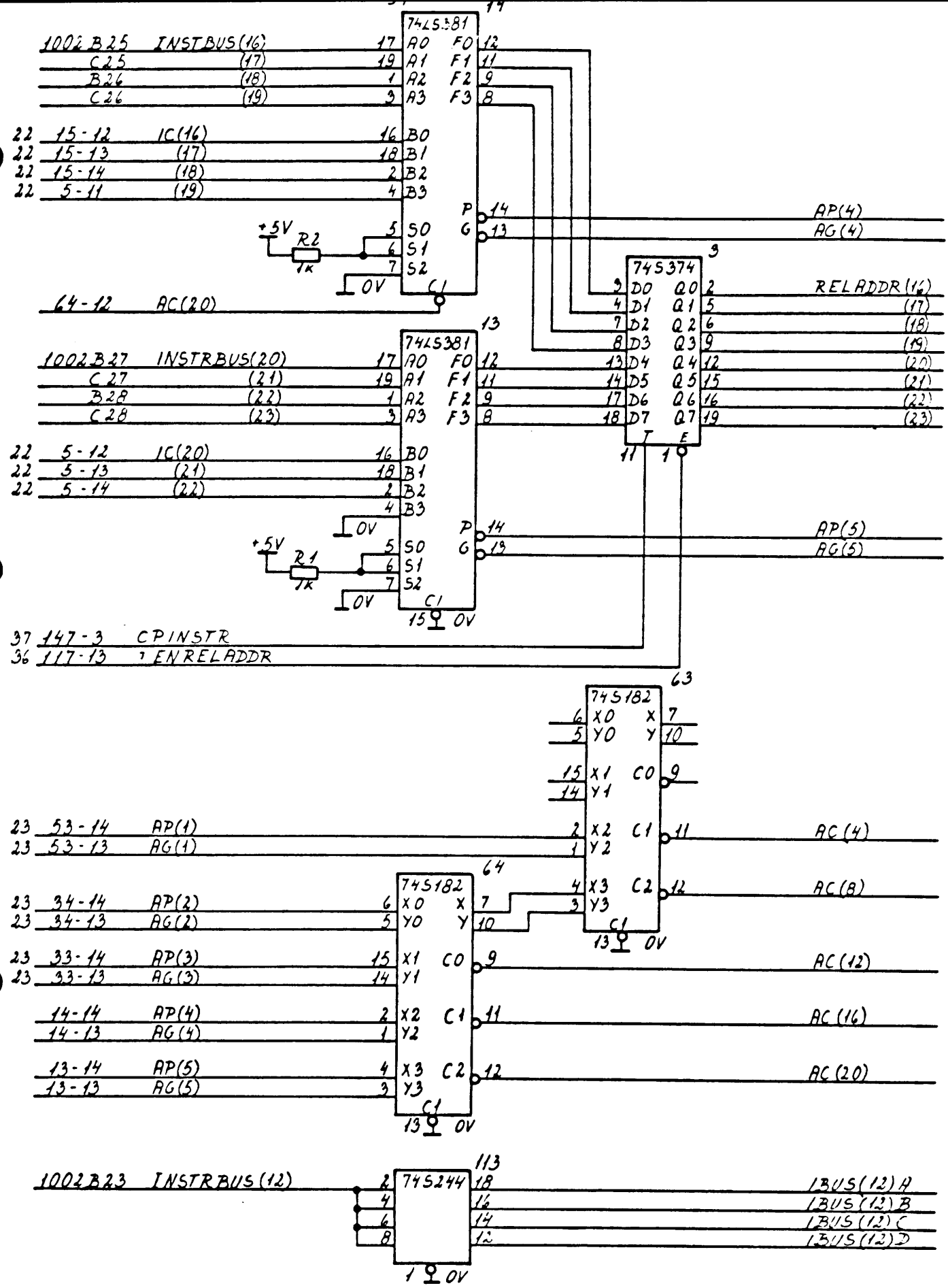


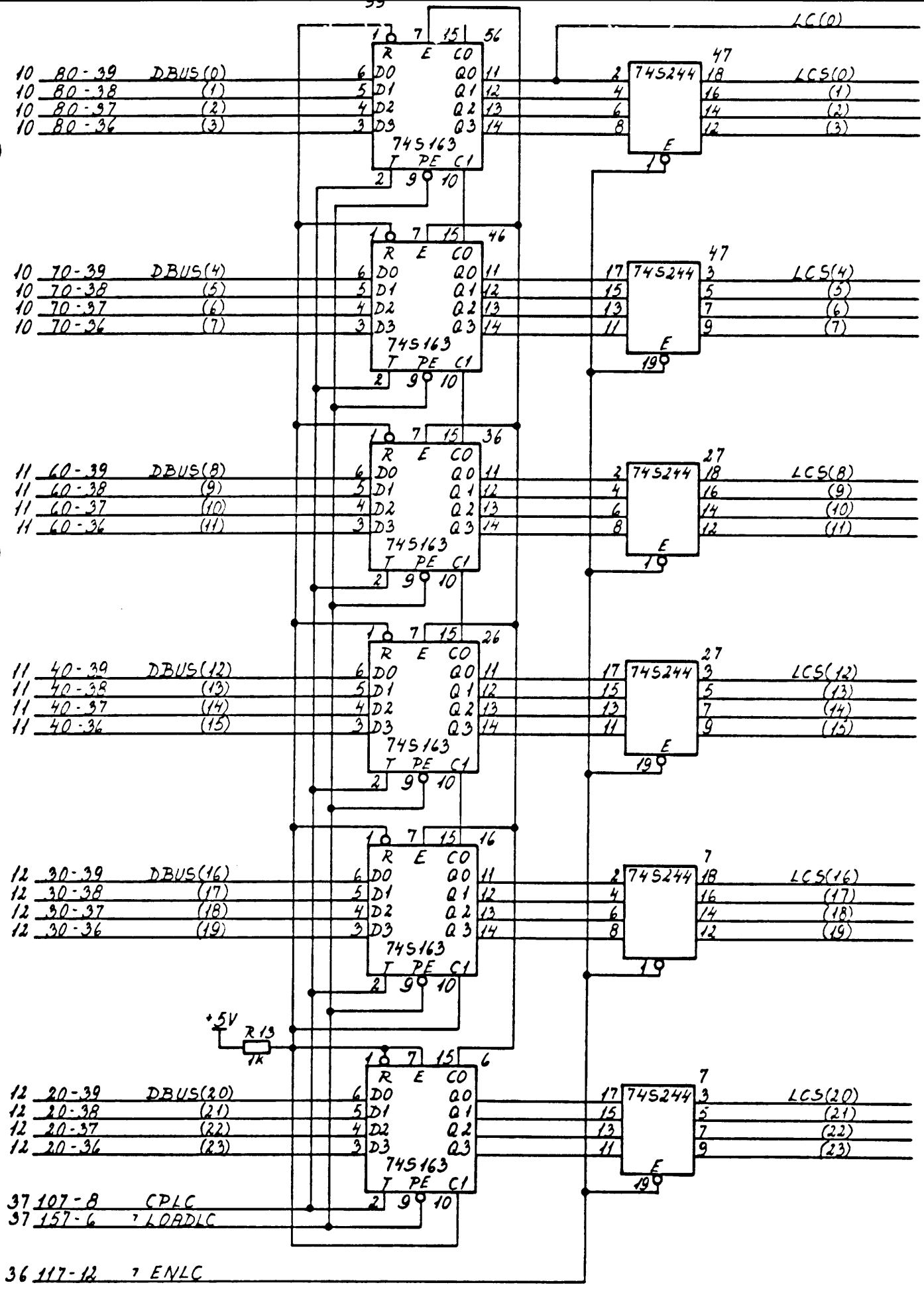

AGA

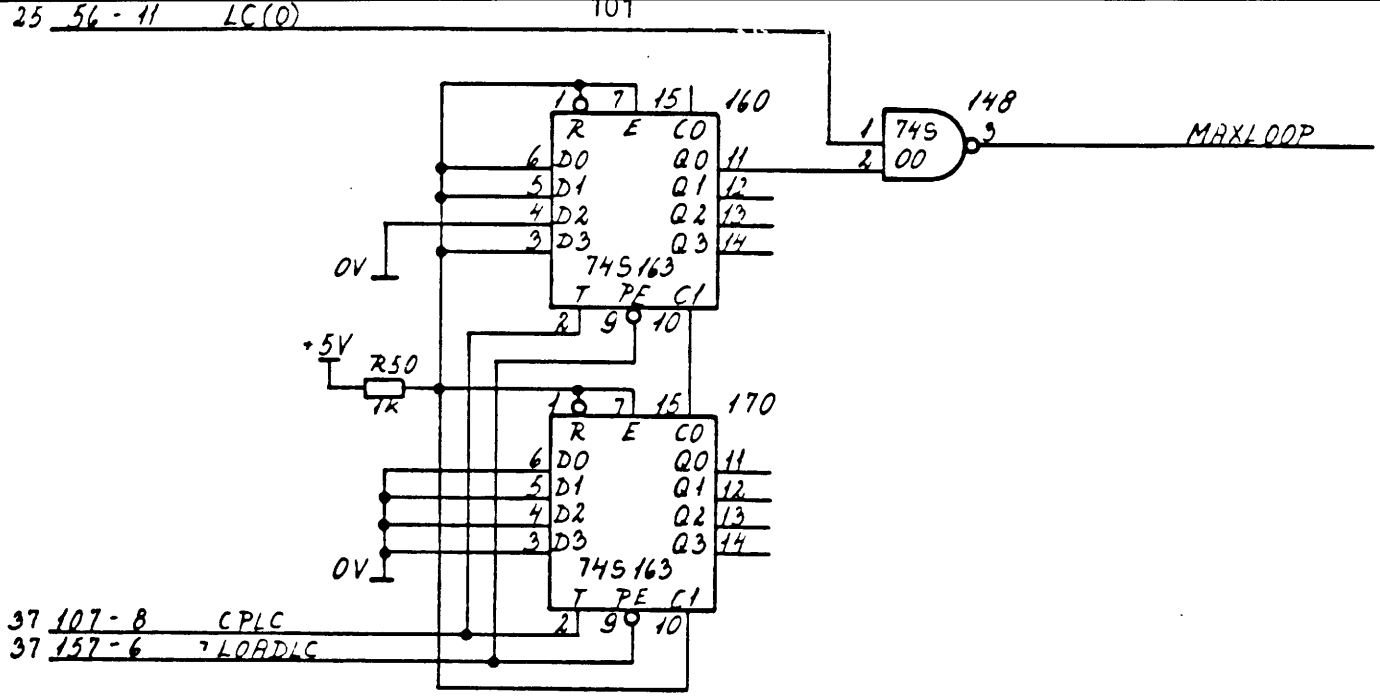
ARJ
80 01 30


55
45
35
25
15
5
74S163
74S244
44
24
4
10 80-38 DBUS(1)
10 80-37 (2)
10 80-36 DBUS(3)
10 70-39 (4)
10 70-38 (5)
10 70-37 (6)
10 70-36 DBUS(7)
11 60-39 (8)
11 60-38 (9)
11 60-37 (10)
11 60-36 DBUS(11)
11 40-39 (12)
11 40-38 (13)
11 40-37 (14)
11 40-36 DBUS(15)
12 30-39 (16)
12 30-38 (17)
12 30-37 (18)
12 30-36 DBUS(19)
12 20-39 (20)
12 20-38 (21)
12 20-37 (22)
37 107-6 CPIC
37 157-5 ↑LOADIC
36 117-14 ↑ENICS
IC(1)
IC(3)
IC(7)
IC(11)
IC(15)
IC(19)
ICS(0)
ICS(4)
ICS(8)
ICS(12)
ICS(16)
ICS(20)
0V
+5V
1K
R3
AGA
AAJ
80130


54
74LS381
24 113-16 IBUS(12)B
A0 A1 A2 A3
F0 F1 F2 F3
OV
B0 B1 B2 B3
22 55-13 IC(1)
22 55-14 (2)
22 45-11 (3)
P G
+5V R12 1K
S0 S1 S2
C1
24 63-11 AC(4)
53
74LS381
24 113-14 IBUS(12)C
22 45-12 IC(4)
22 45-13 (5)
22 45-14 (6)
22 35-11 (7)
R11
24 63-12 AC(8)
34
74LS381
24 113-12 IBUS(12)D
22 35-12 IC(8)
22 35-13 (9)
22 35-14 (10)
22 25-11 (11)
R9
24 64-9 AC(12)
33
74LS381
1002B23 INSTRBUS(12)
C23 (13)
B24 (14)
C24 (15)
22 25-12 IC(12)
22 25-13 (13)
22 25-14 (14)
22 15-11 (15)
R8
24 64-11 AC(16)
43
74S374
D0 D1 D2 D3 D4 D5 D6 D7
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
T E
REL ADDR(0)
(1)
(2)
(3)
(4)
(5)
(6)
(7)
AP(1)
AG(1)
23
74S374
REL ADDR(8)
(9)
(11)
(13)
(15)
AP(2)
AG(2)
AP(3)
AG(3)
37 147-3 CP INSTR
36 117-13 ENREL ADDR
AG A
AAJ 800130

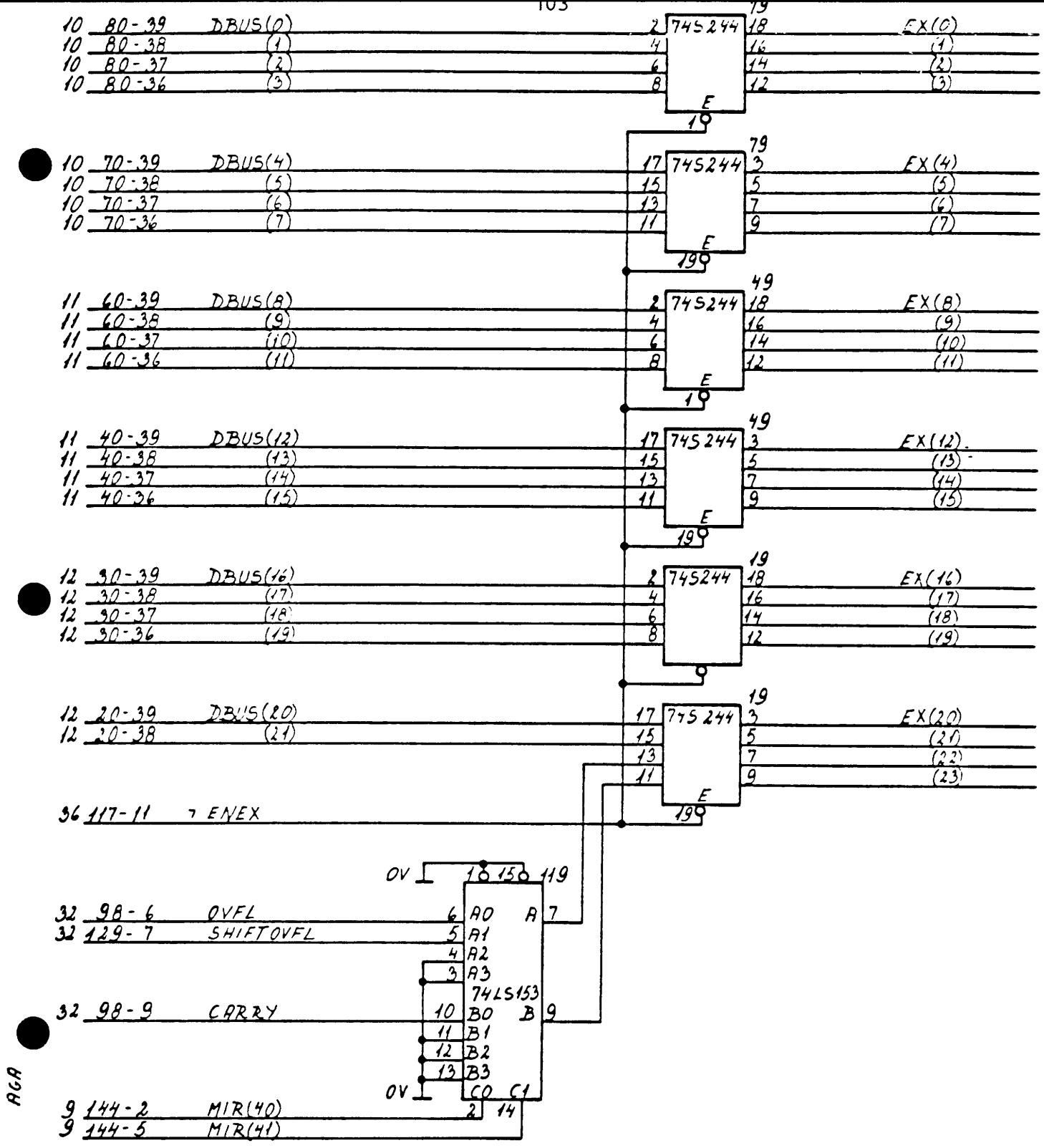

74LS381
74LS374
74LS182
74LS244
INSTBUS(16)
INSTRBUS(20)
IC(16)
IC(20)
AC(20)
CPINSTR
ENRELADDR
AP(4)
AG(4)
AP(5)
AG(5)
RELADDR(16)
AP(1)
AG(1)
AP(2)
AG(2)
AP(3)
AG(3)
AC(4)
AC(8)
AC(12)
AC(16)
AC(20)
INSTRBUS(12)
IBUS(12)A
IBUS(12)B
IBUS(12)C
IBUS(12)D
+5V
R1
R2
1K
OV


LC(0)
10 80-39 DBUS(0)
10 80-38 (1)
10 80-37 (2)
10 80-36 (3)
74S163
74S244
LCS(0)
LCS(1)
LCS(2)
LCS(3)
10 70-39 DBUS(4)
10 70-38 (5)
10 70-37 (6)
10 70-36 (7)
LCS(4)
LCS(5)
LCS(6)
LCS(7)
11 60-39 DBUS(8)
11 60-38 (9)
11 60-37 (10)
11 60-36 (11)
LCS(8)
LCS(9)
LCS(10)
LCS(11)
11 40-39 DBUS(12)
11 40-38 (13)
11 40-37 (14)
11 40-36 (15)
LCS(12)
LCS(13)
LCS(14)
LCS(15)
12 30-39 DBUS(16)
12 30-38 (17)
12 30-37 (18)
12 30-36 (19)
LCS(16)
LCS(17)
LCS(18)
LCS(19)
+5V
R13
1K
12 20-39 DBUS(20)
12 20-38 (21)
12 20-37 (22)
12 20-36 (23)
LCS(20)
LCS(21)
LCS(22)
LCS(23)
37 107-8 CPLC
37 157-6 ↑LOADLC
36 117-12 ↑ENLC
AGA
AAJ
8001130



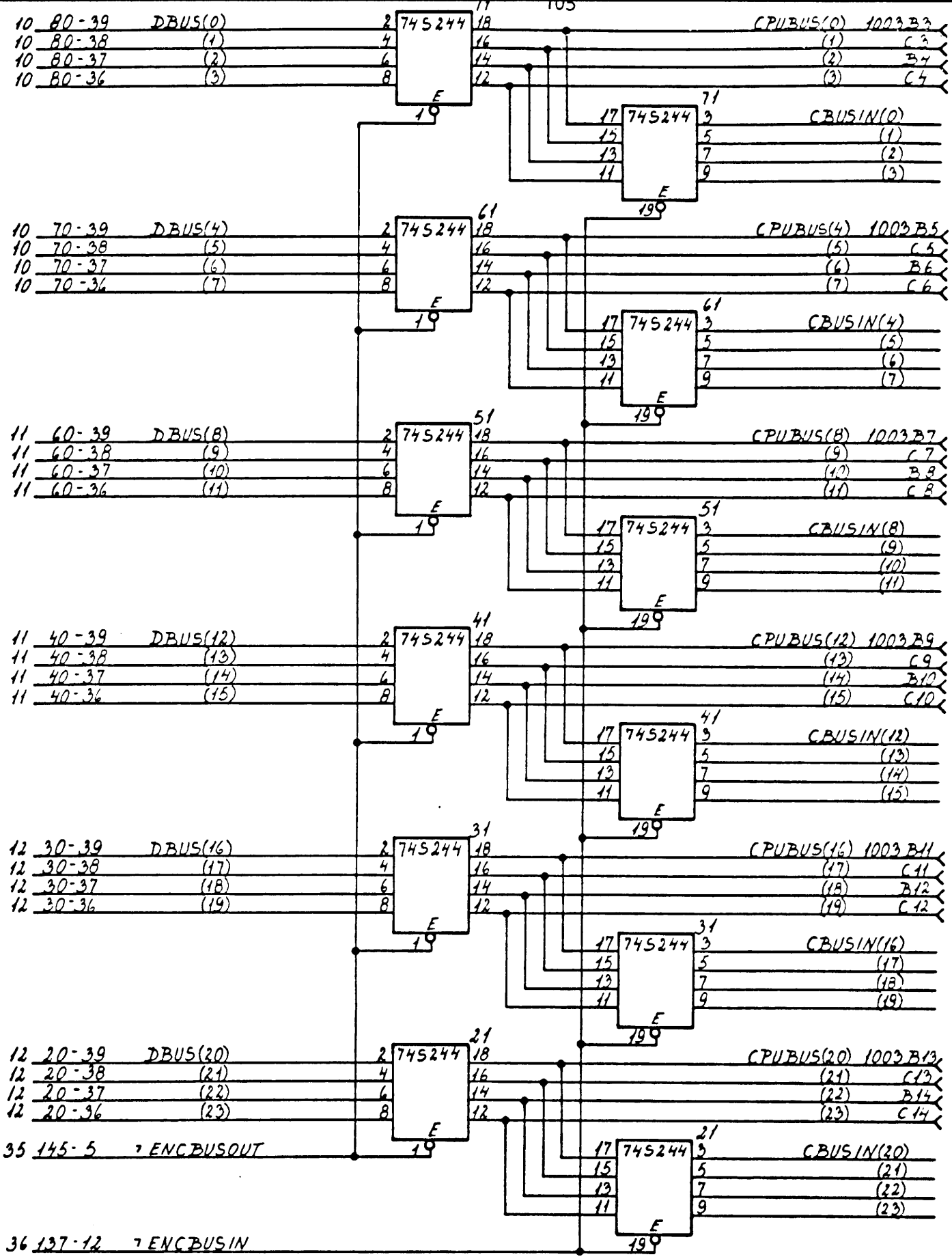

AGA

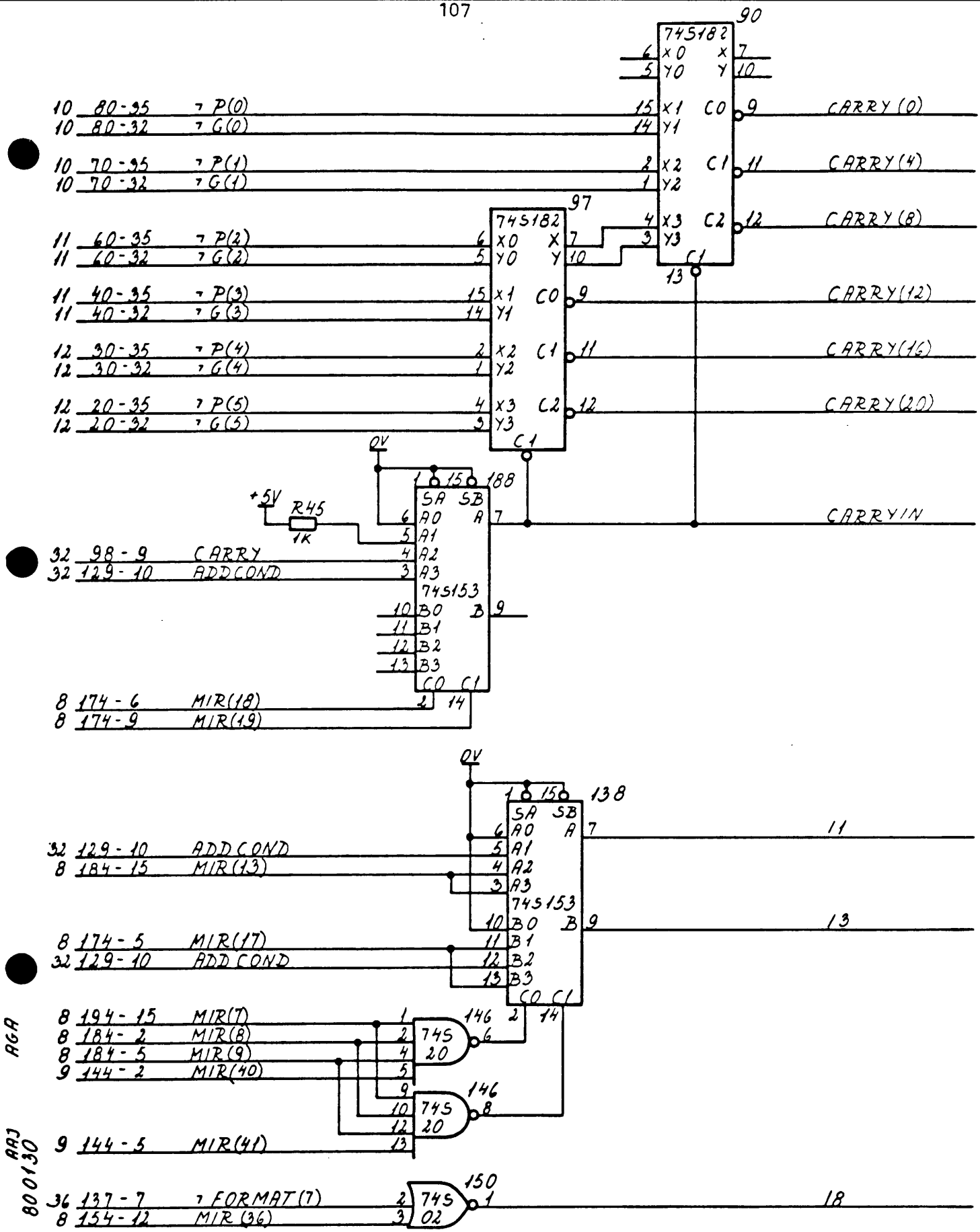
RAJ
800130

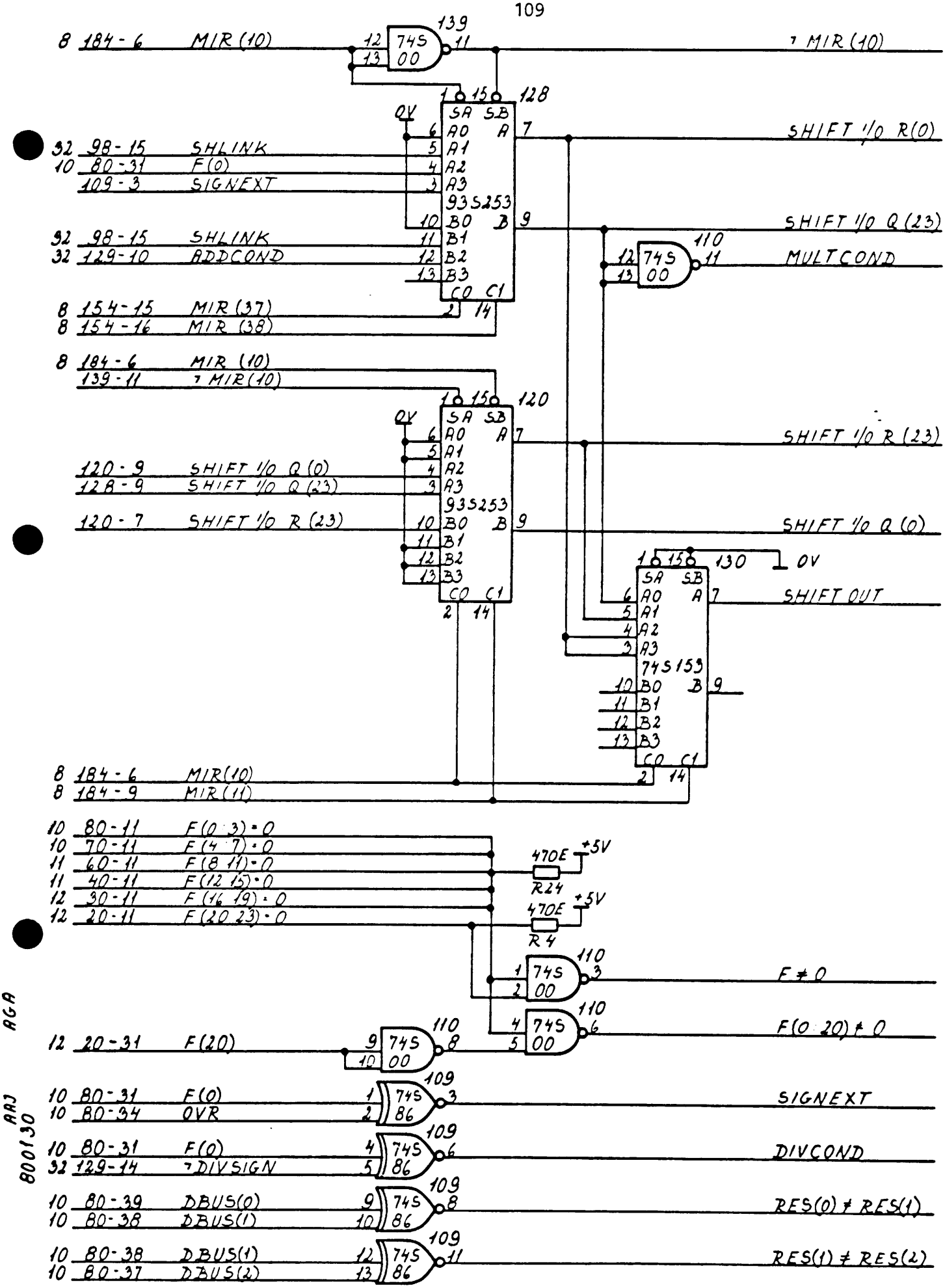



AGA

ARJ
800130

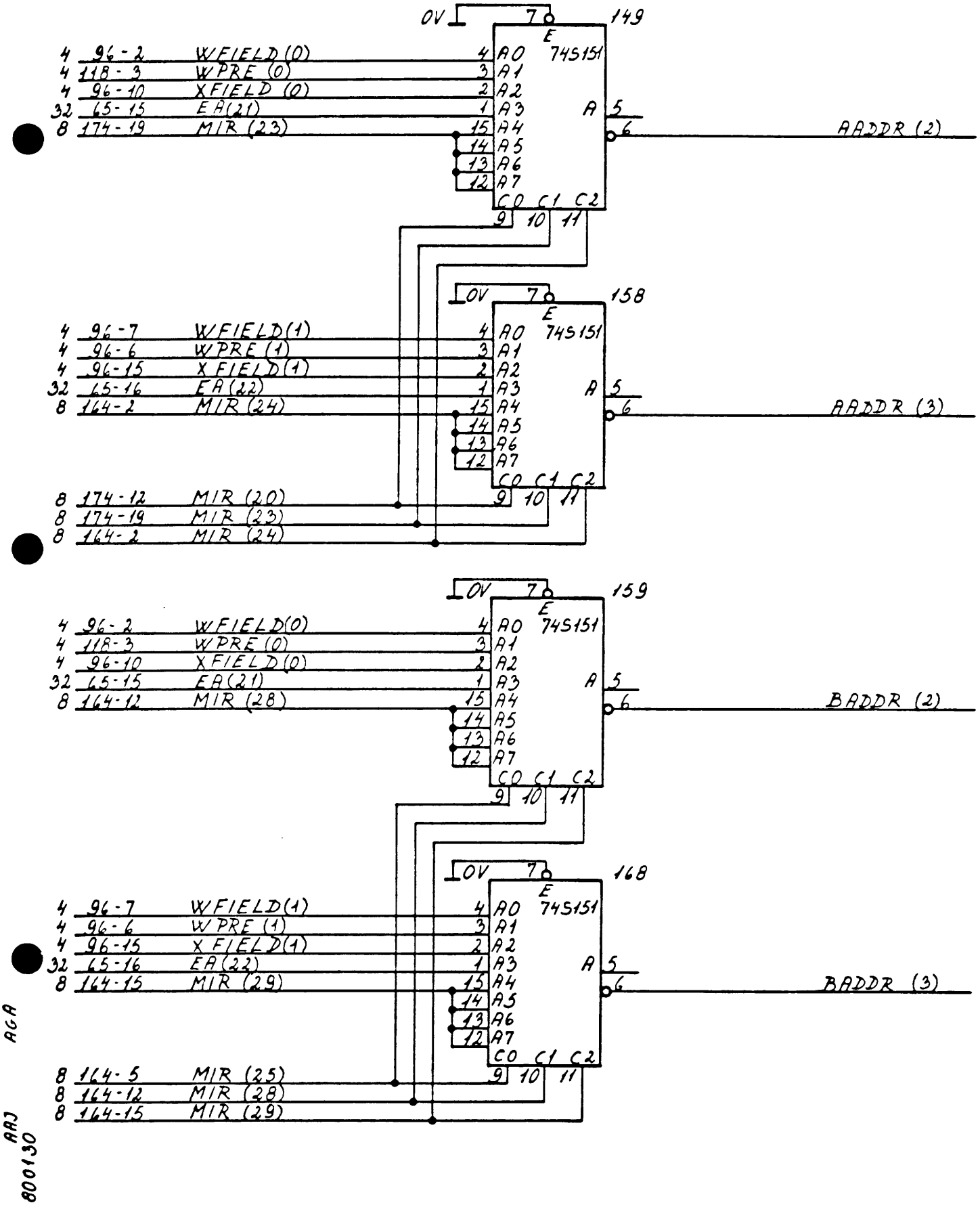

10 80-39 DBUS(0)
10 80-38 (1)
10 80-37 (2)
10 80-36 (3)
10 70-39 DBUS(4)
10 70-38 (5)
10 70-37 (6)
10 70-36 (7)
11 60-39 DBUS(8)
11 60-38 (9)
11 60-37 (10)
11 60-36 (11)
11 40-39 DBUS(12)
11 40-38 (13)
11 40-37 (14)
11 40-36 (15)
12 30-39 DBUS(16)
12 30-38 (17)
12 30-37 (18)
12 30-36 (19)
12 20-39 DBUS(20)
12 20-38 (21)
12 20-37 (22)
12 20-36 (23)
35 145-5 ENCBUSOUT
36 137-12 ENCBUSIN
74S244
CPUBUS(0) 1003B3
CPUBUS(4) 1003B5
CPUBUS(8) 1003B7
CPUBUS(12) 1003B9
CPUBUS(16) 1003B11
CPUBUS(20) 1003B13
CBUSIN(0)
CBUSIN(4)
CBUSIN(8)
CBUSIN(12)
CBUSIN(16)
CBUSIN(20)

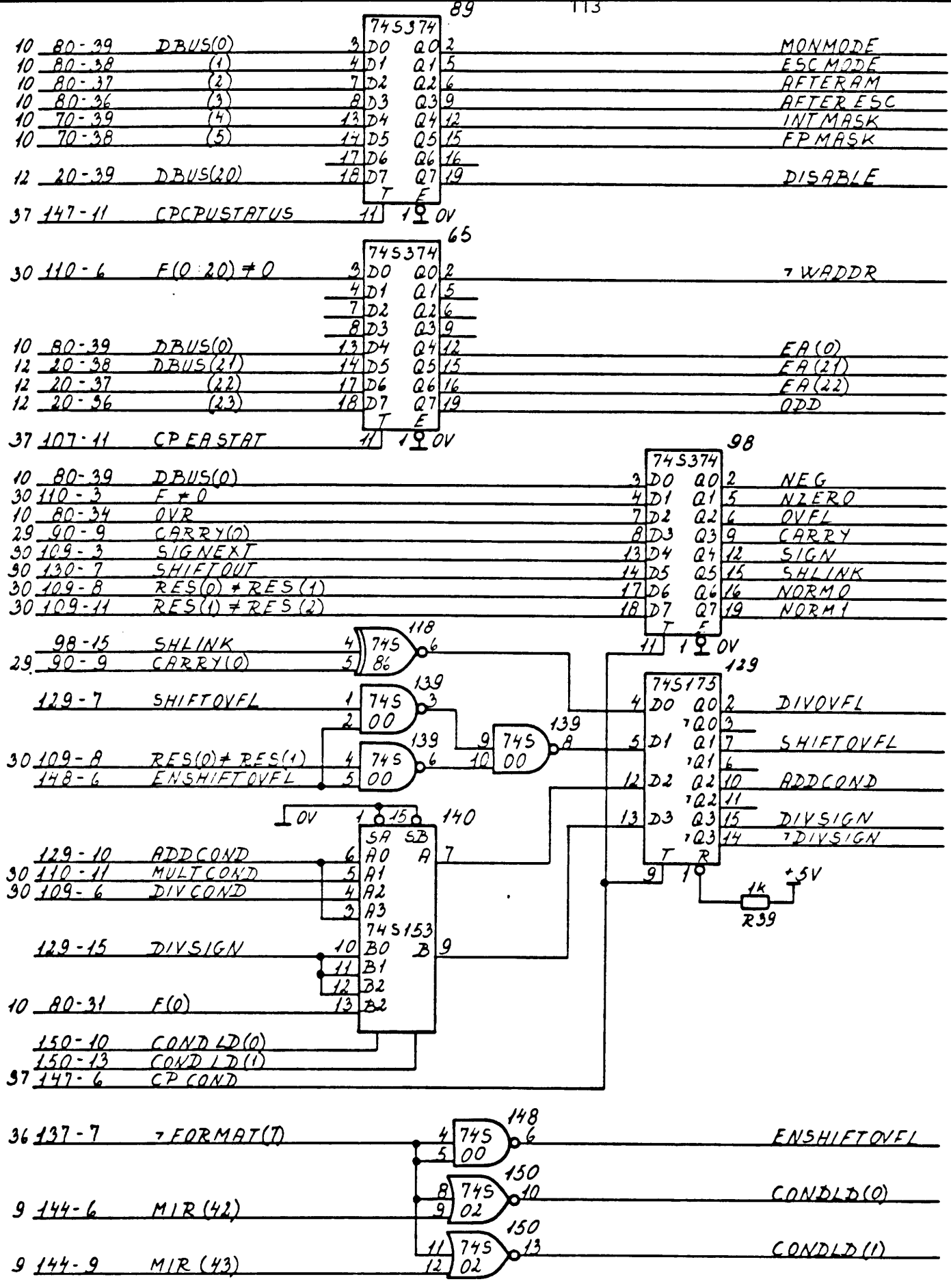

90
74S182
10 80-35 ⊐ P(0)
10 80-32 ⊐ G(0)
10 70-35 ⊐ P(1)
10 70-32 ⊐ G(1)
CARRY(0)
CARRY(4)
CARRY(8)
97
11 60-35 ⊐ P(2)
11 60-32 ⊐ G(2)
11 40-35 ⊐ P(3)
11 40-32 ⊐ G(3)
12 30-35 ⊐ P(4)
12 30-32 ⊐ G(4)
12 20-35 ⊐ P(5)
12 20-32 ⊐ G(5)
CARRY(12)
CARRY(16)
CARRY(20)
+5V
R45
1K
188
CARRY IN
32 98-9 CARRY
32 129-10 ADDCOND
74S153
8 174-6 MIR(18)
8 174-9 MIR(19)
138
32 129-10 ADDCOND
8 184-15 MIR(13)
8 174-5 MIR(17)
32 129-10 ADD COND
8 194-15 MIR(7)
8 184-2 MIR(8)
8 184-5 MIR(9)
9 144-2 MIR(40)
146
74S20
9 144-5 MIR(41)
AAJ 80 0130
36 137-7 ⊐ FORMAT(7)
8 154-12 MIR(36)
150
74S02
AGA

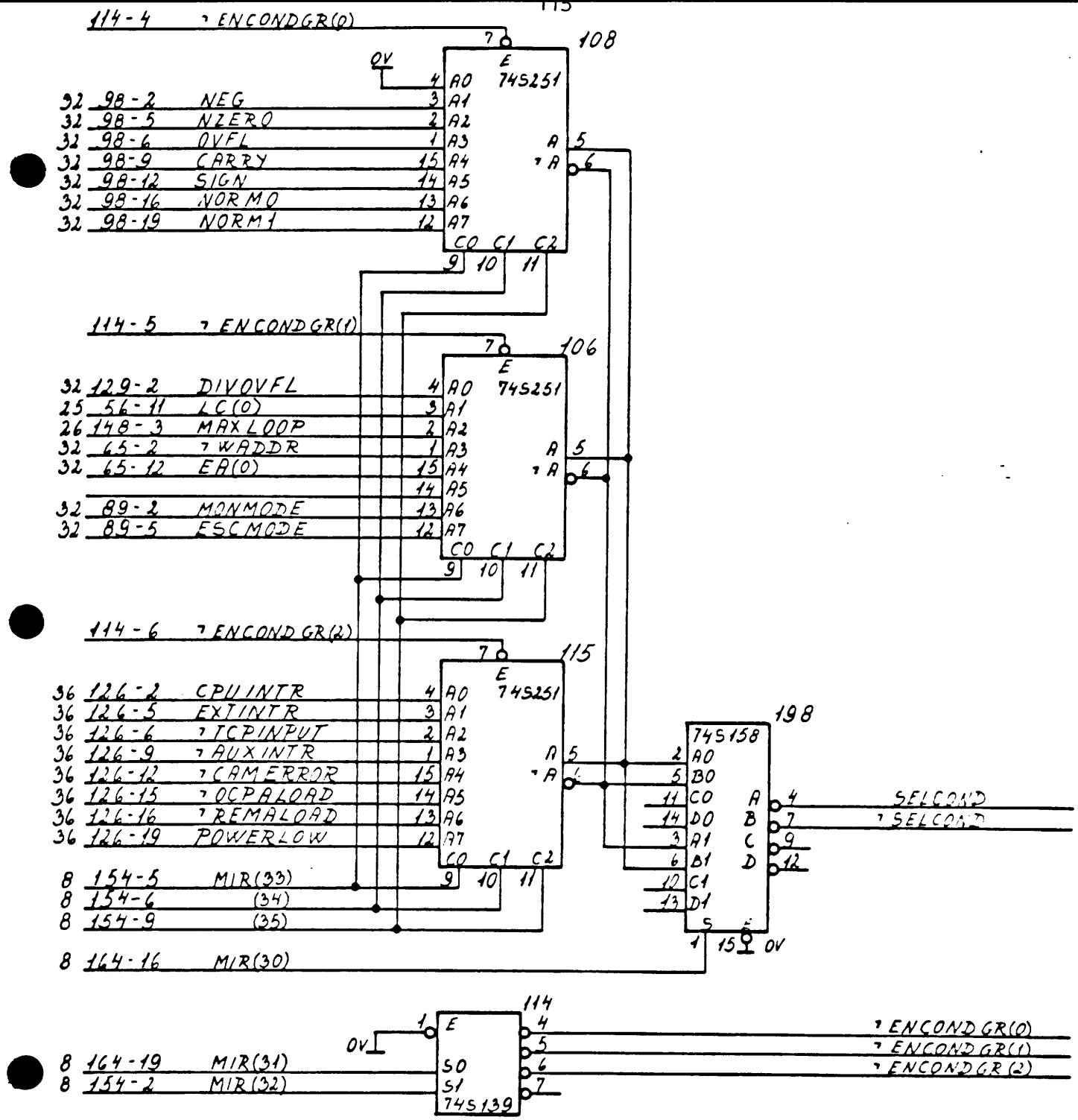



AGA

AAJ 800130

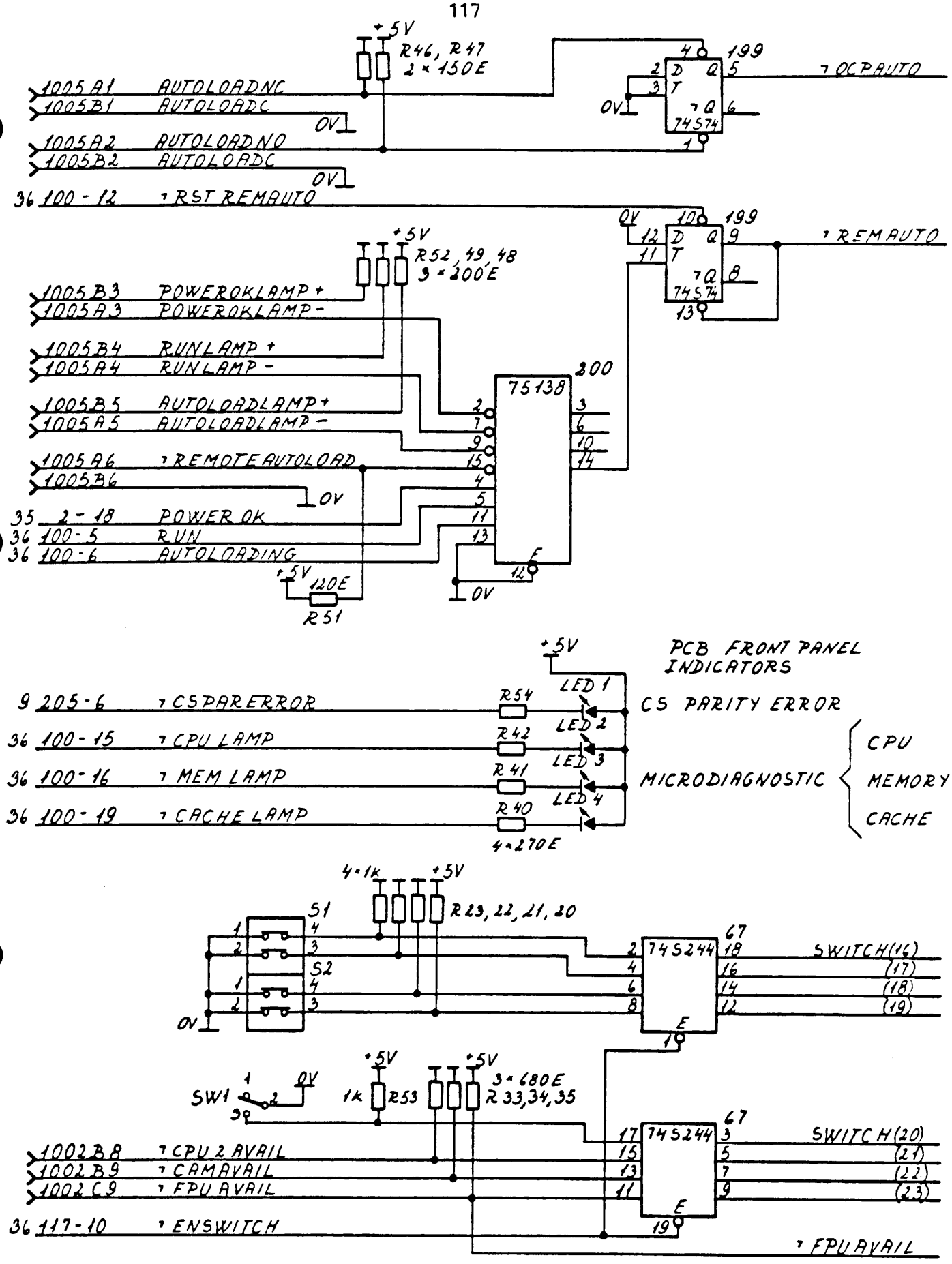

149
158
159
168
74S151
4 96-2 WFIELD(0)
4 118-3 WPRE (0)
4 96-10 XFIELD (0)
32 65-15 EA(21)
8 174-19 MIR (23)
AADDR (2)
4 96-7 WFIELD(1)
4 96-6 WPRE (1)
4 96-15 XFIELD(1)
32 65-16 EA(22)
8 164-2 MIR (24)
AADDR (3)
8 174-12 MIR (20)
8 174-19 MIR (23)
8 164-2 MIR (24)
4 96-2 WFIELD(0)
4 118-3 WPRE (0)
4 96-10 XFIELD (0)
32 65-15 EA(21)
8 164-12 MIR (28)
BADDR (2)
4 96-7 WFIELD(1)
4 96-6 WPRE (1)
4 96-15 XFIELD(1)
32 65-16 EA(22)
8 164-15 MIR (29)
BADDR (3)
8 164-5 MIR (25)
8 164-12 MIR (28)
8 164-15 MIR (29)
AGA
ARJ
800130

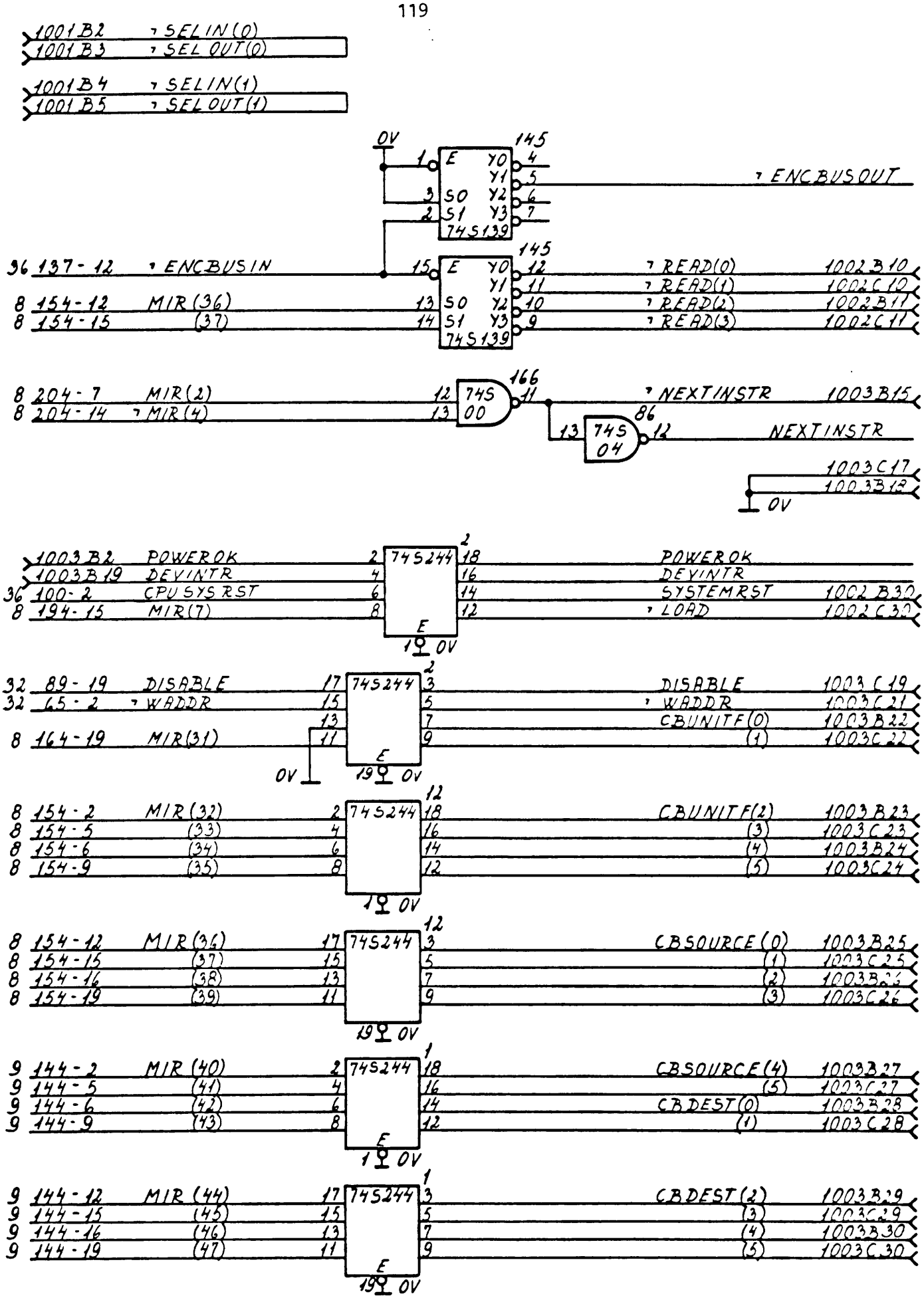

89
74S374
10 80-39 DBUS(0)
10 80-38 (1)
10 80-37 (2)
10 80-36 (3)
10 70-39 (4)
10 70-38 (5)
12 20-39 DBUS(20)
37 147-11 CPCPUSTATUS
MONMODE
ESCMODE
AFTERAM
AFTERESC
INTMASK
FPMASK
DISABLE
65
30 110-6 F(0:20)≠0
10 80-39 DBUS(0)
12 20-38 DBUS(21)
12 20-37 (22)
12 20-36 (23)
37 107-11 CPEASTAT
→WADDR
EA(0)
EA(21)
EA(22)
ODD
98
10 80-39 DBUS(0)
30 110-3 F≠0
10 80-34 OVR
29 90-9 CARRY(0)
30 109-3 SIGNEXT
30 130-7 SHIFTOUT
30 109-8 RES(0)≠RES(1)
30 109-11 RES(1)≠RES(2)
NEG
NZERO
OVFL
CARRY
SIGN
SHLINK
NORM0
NORM1
118
74S86
98-15 SHLINK
29 90-9 CARRY(0)
139
74S00
129-7 SHIFTOVFL
30 109-8 RES(0)≠RES(1)
148-6 ENSHIFTOVFL
129
74S175
DIVOVFL
SHIFTOVFL
ADDCOND
DIVSIGN
→DIVSIGN
+5V
1K
R39
140
74S153
129-10 ADDCOND
30 110-11 MULTCOND
30 109-6 DIVCOND
129-15 DIVSIGN
10 80-31 F(0)
150-10 CONDLD(0)
150-13 CONDLD(1)
37 147-6 CP COND
AGA
36 137-7 →FORMAT(7)
9 144-6 MIR(42)
9 144-9 MIR(43)
148
150
150
ENSHIFTOVFL
CONDLD(0)
CONDLD(1)
AAJ
800130



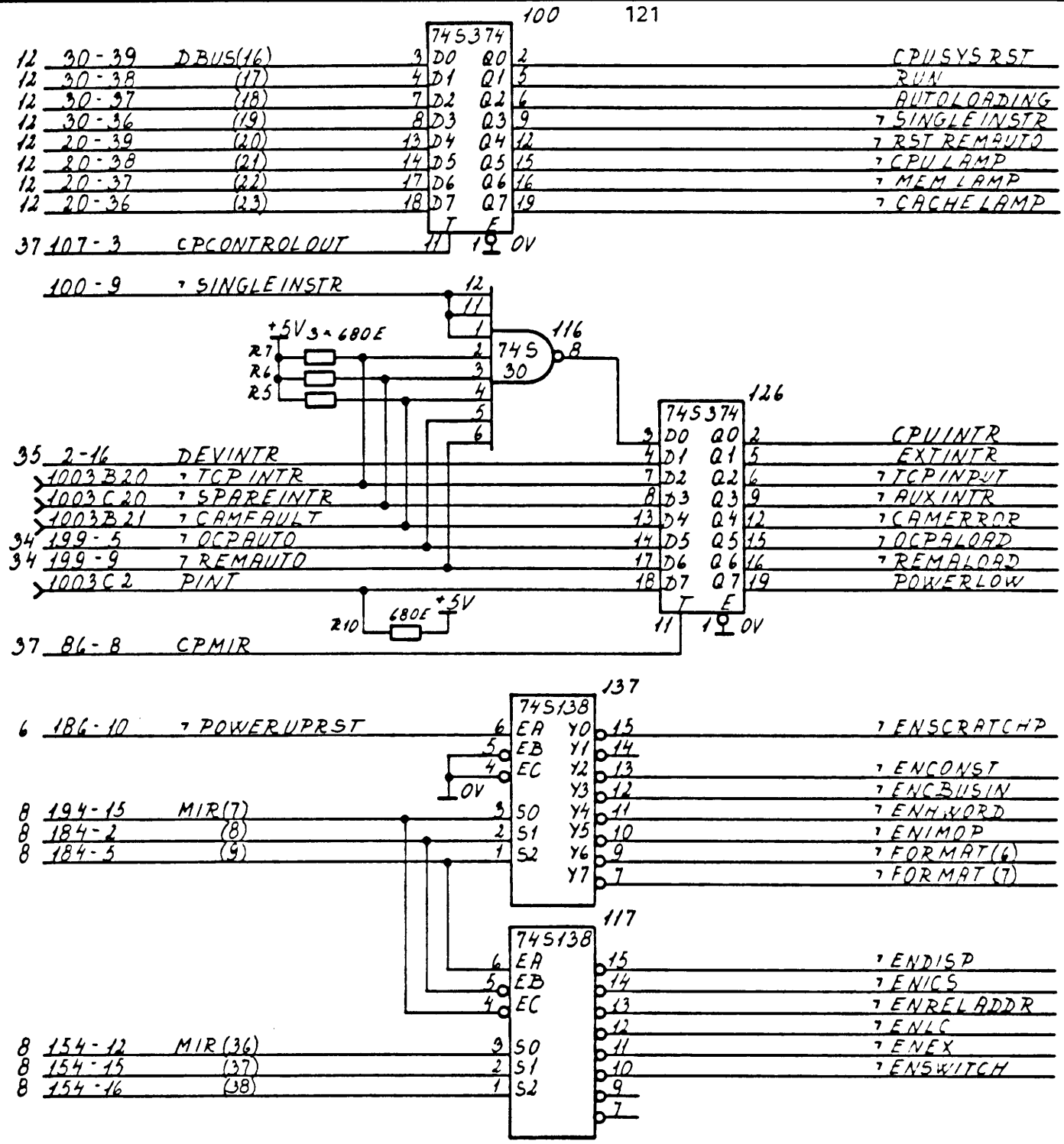

AGA

AAJ 800130


+5V
R46, R47
2 × 150E
1005 A1 AUTOLOADNC
1005 B1 AUTOLOADC
OV
1005 A2 AUTOLOADNO
1005 B2 AUTOLOADC
OV
36 100 - 12 ¬RST REMAUTO
199
74S74
¬QCPAUTO
OV
¬REMAUTO
+5V
R52, 49, 48
3 × 200E
1005 B3 POWEROKLAMP +
1005 A3 POWEROKLAMP -
1005 B4 RUNLAMP +
1005 A4 RUNLAMP -
1005 B5 AUTOLOADLAMP +
1005 A5 AUTOLOADLAMP -
1005 A6 ¬REMOTE AUTOLOAD
1005 B6
OV
35 2 - 18 POWER OK
36 100 - 5 RUN
36 100 - 6 AUTOLOADING
75138
300
+5V
120E
R51
OV
+5V
PCB FRONT PANEL
INDICATORS
9 205 - 6 ¬CSPARERROR
R54
LED 1
CS PARITY ERROR
36 100 - 15 ¬CPU LAMP
R42
LED 2
CPU
36 100 - 16 ¬MEM LAMP
R41
LED 3
MICRODIAGNOSTIC
MEMORY
36 100 - 19 ¬CACHE LAMP
R40
LED 4
CACHE
4 × 270E
4 × 1k
+5V
R23, 22, 21, 20
S1
S2
OV
74S244
67
SWITCH(16)
(17)
(18)
(19)
SW1
1k
R53
+5V
+5V
3 × 680E
R33, 34, 35
SWITCH(20)
(21)
(22)
(23)
1002 B8 ¬CPU 2 AVAIL
1002 B9 ¬CAMAVAIL
1002 C9 ¬FPU AVAIL
36 117 - 10 ¬ENSWITCH
¬FPU AVAIL
AGA
AAJ
800130




AGA
AAJ
800130



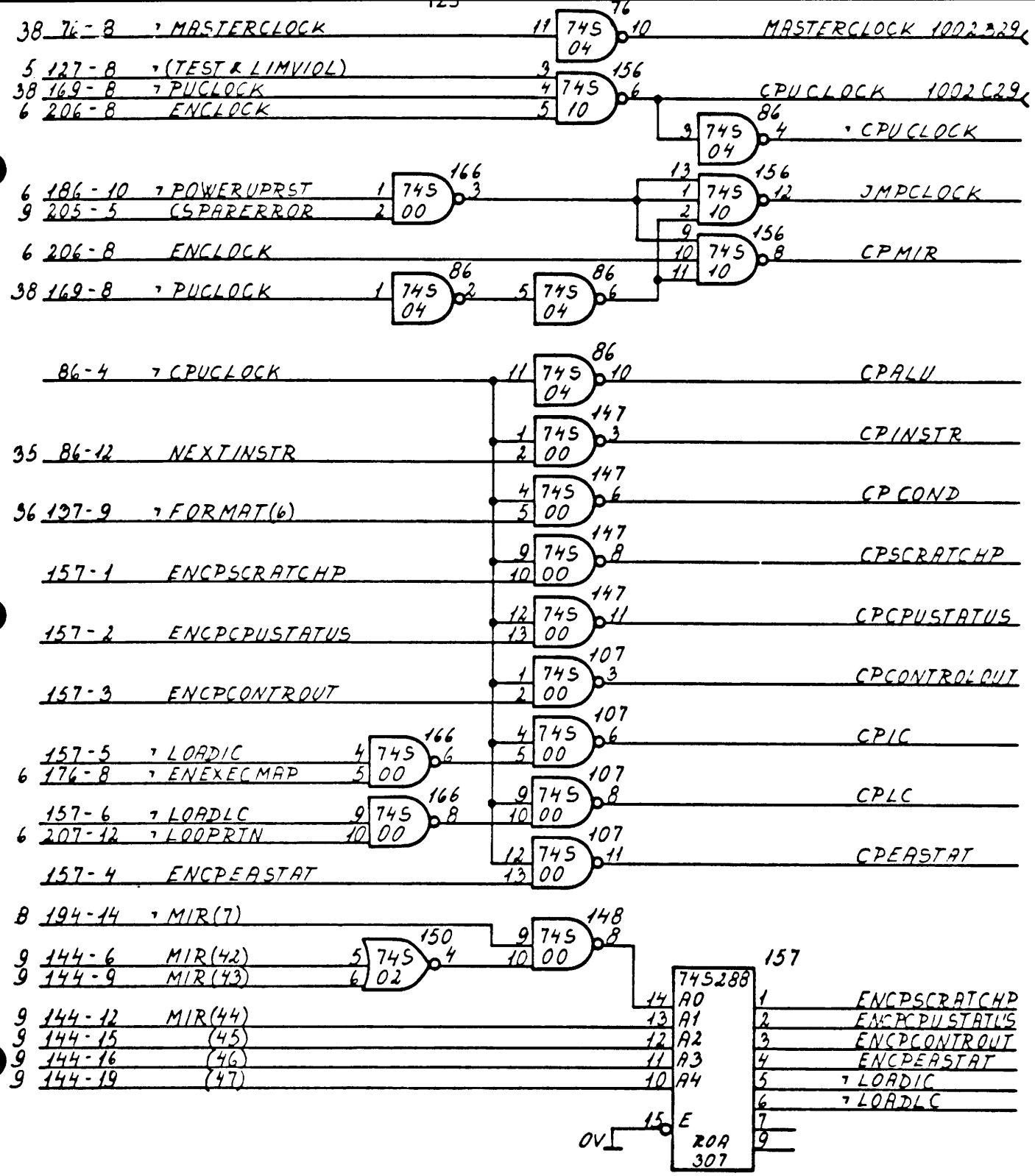

AGA

AAJ
800130

CPU 821

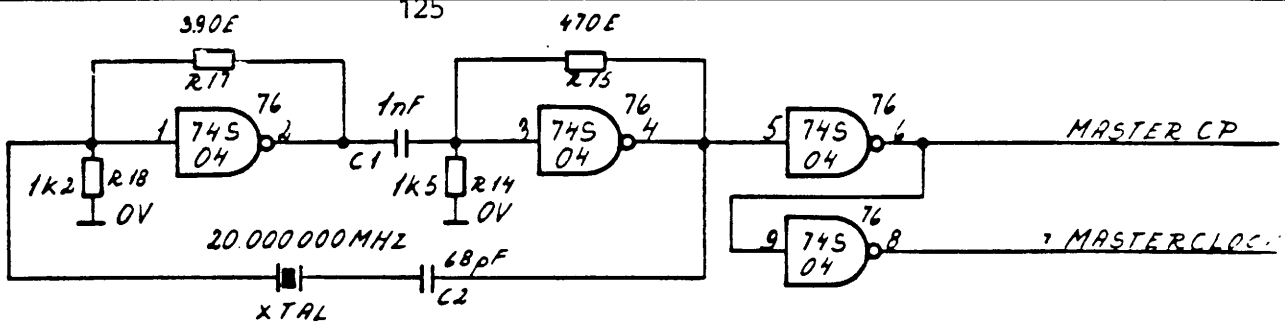
R 12957

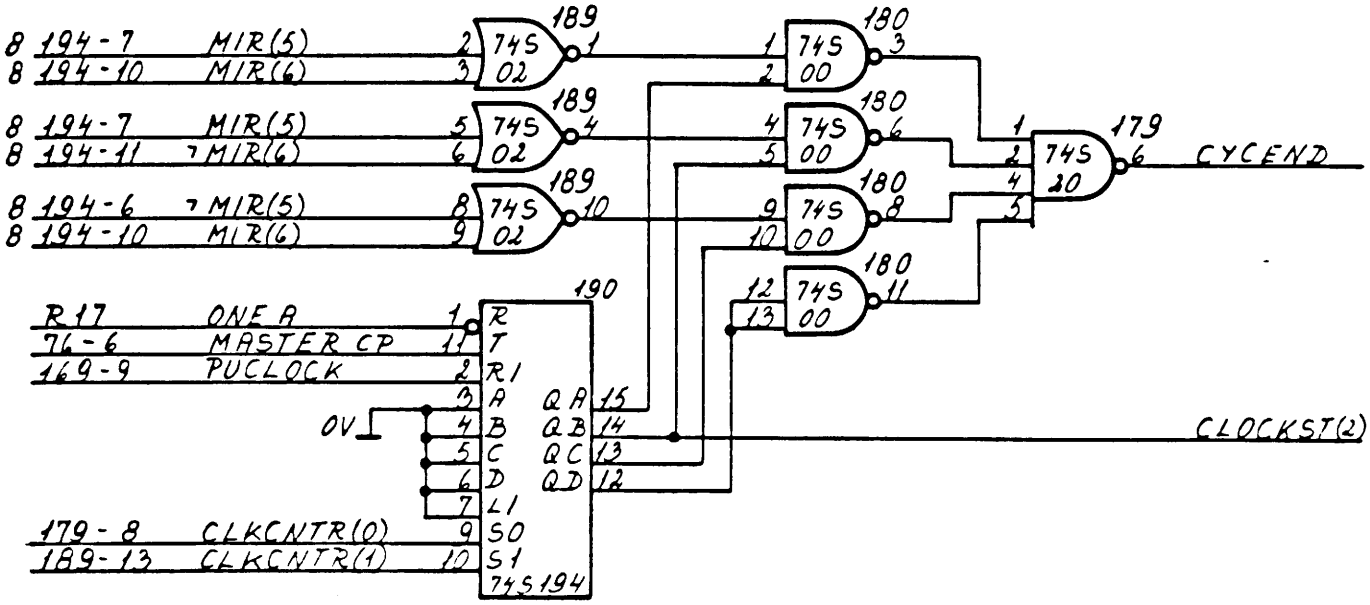
CONTROL OUTPUT REGISTER INTERRUPT STATUS REGISTER & SBUS SOURCE ADDRESS DECODING

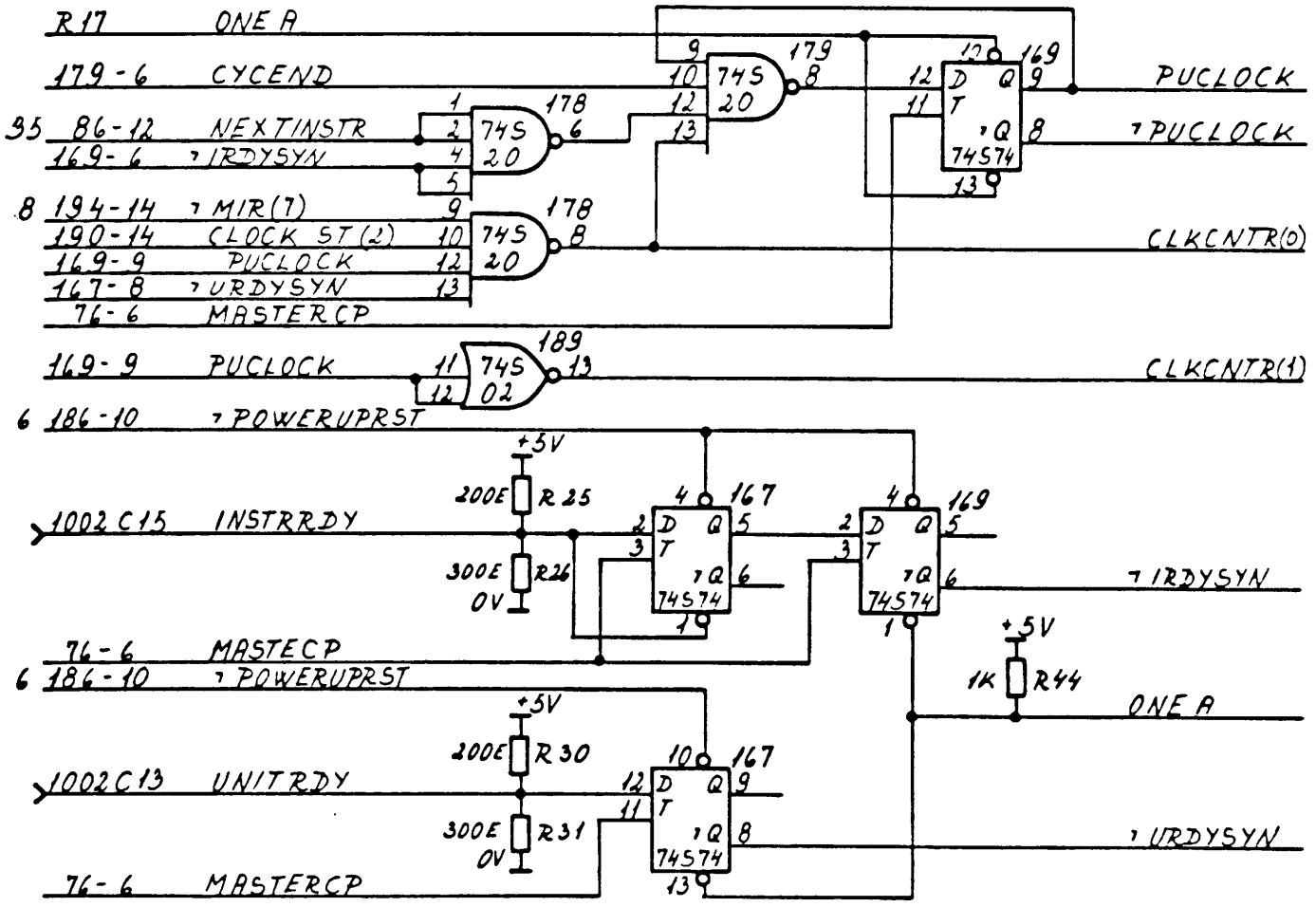



RGA

800130 ARJ



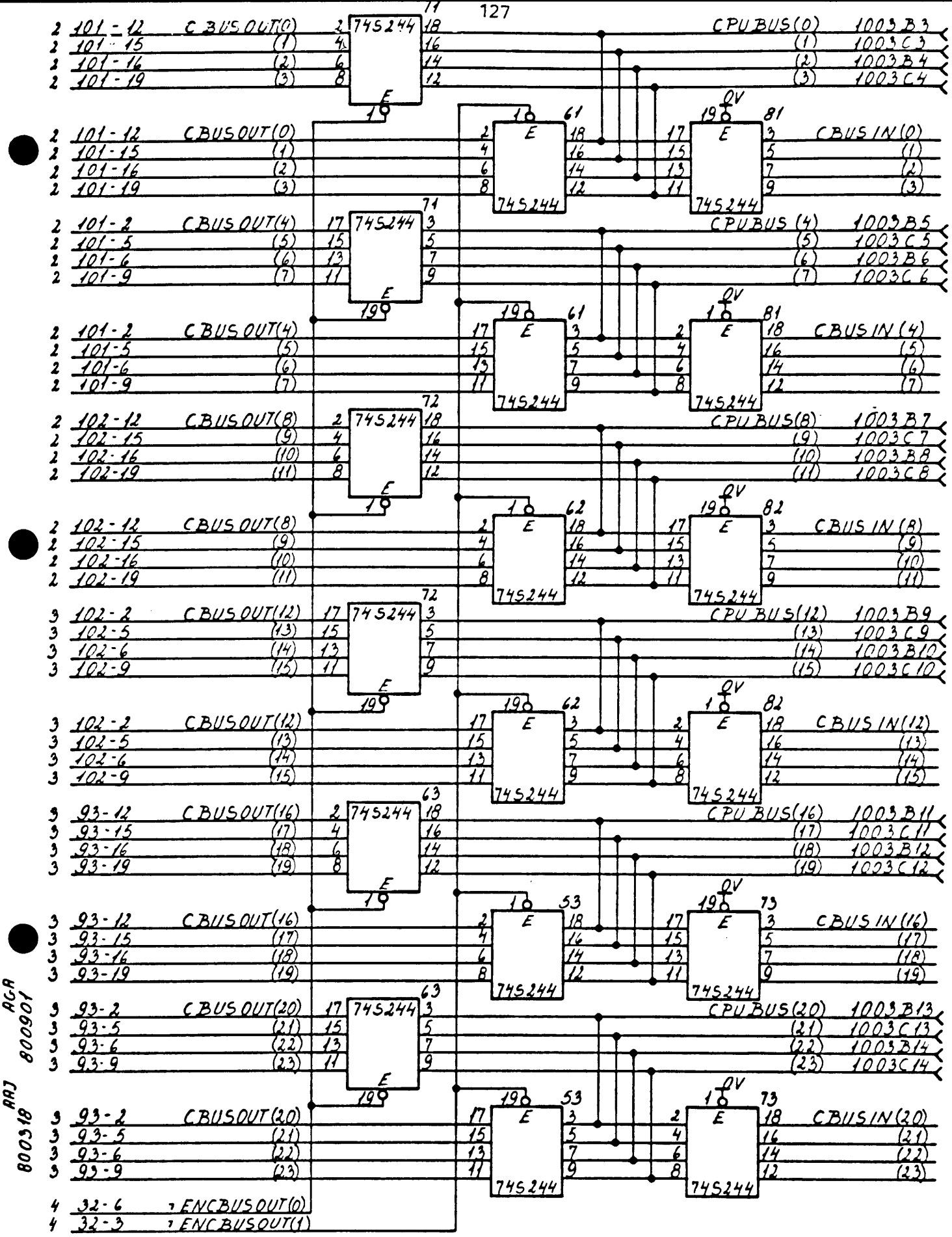





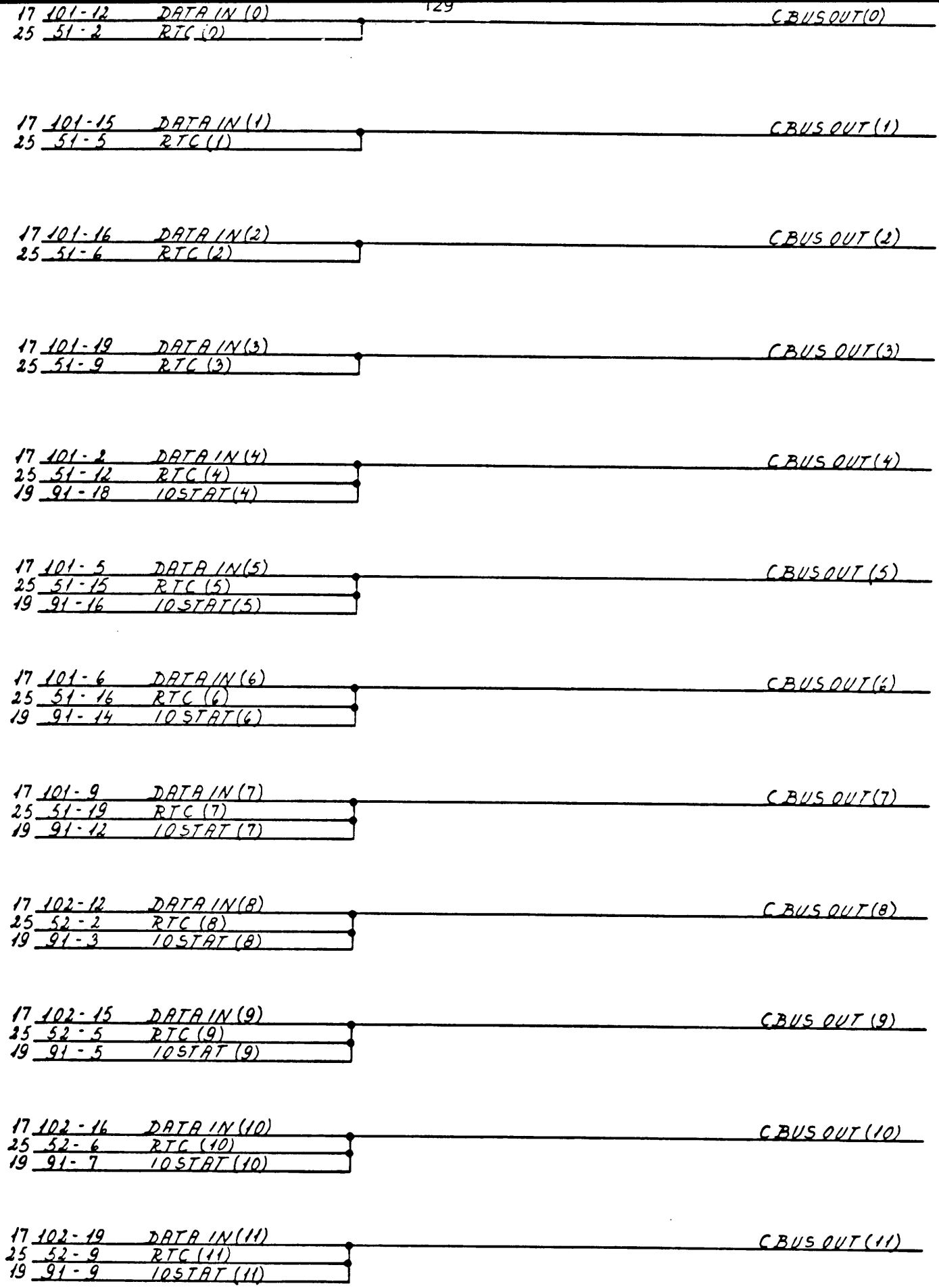


AGA
AAJ 80 01 30

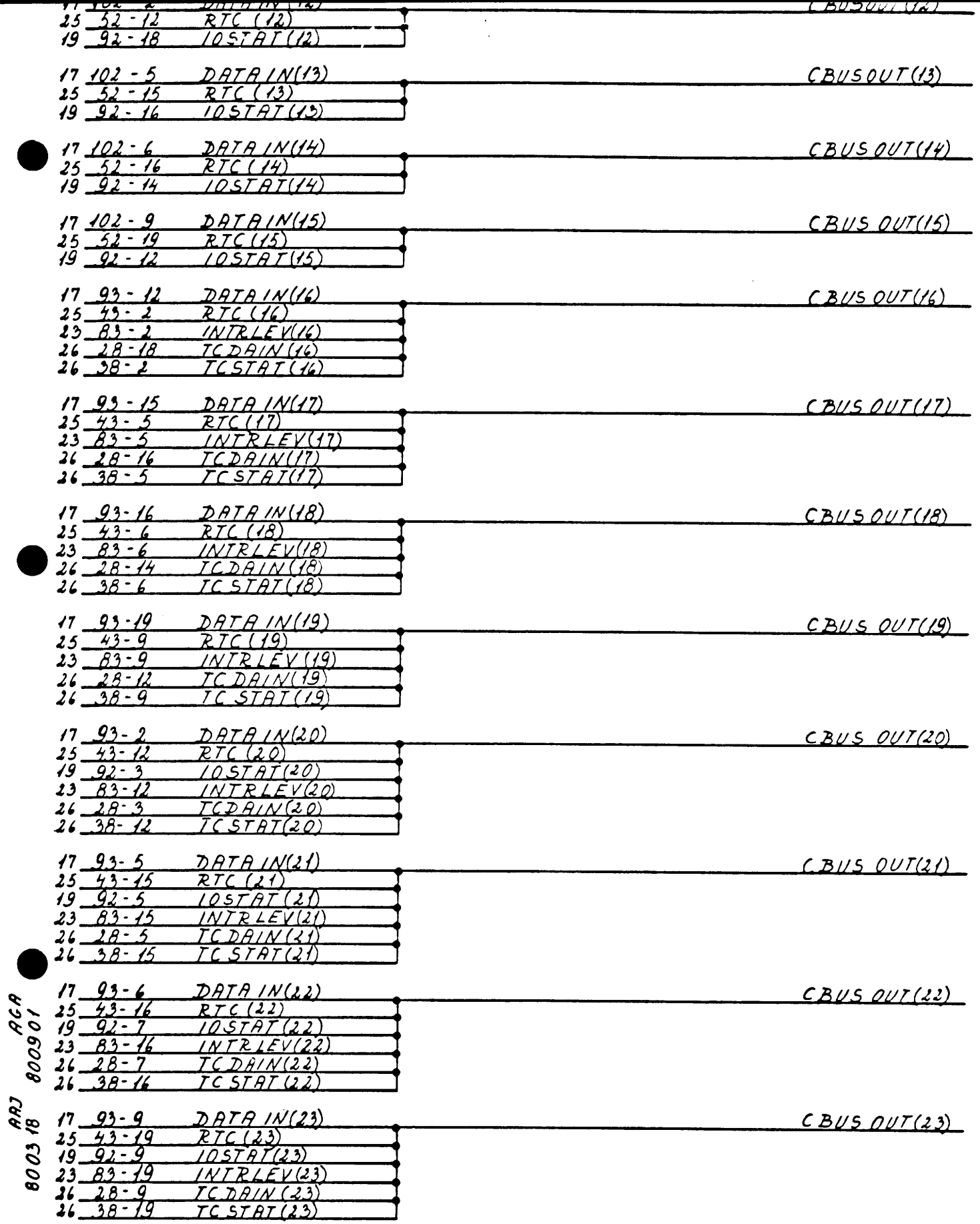



CPU 822 CPUBUS DRIVERS & RECEIVERS 1

R 13015

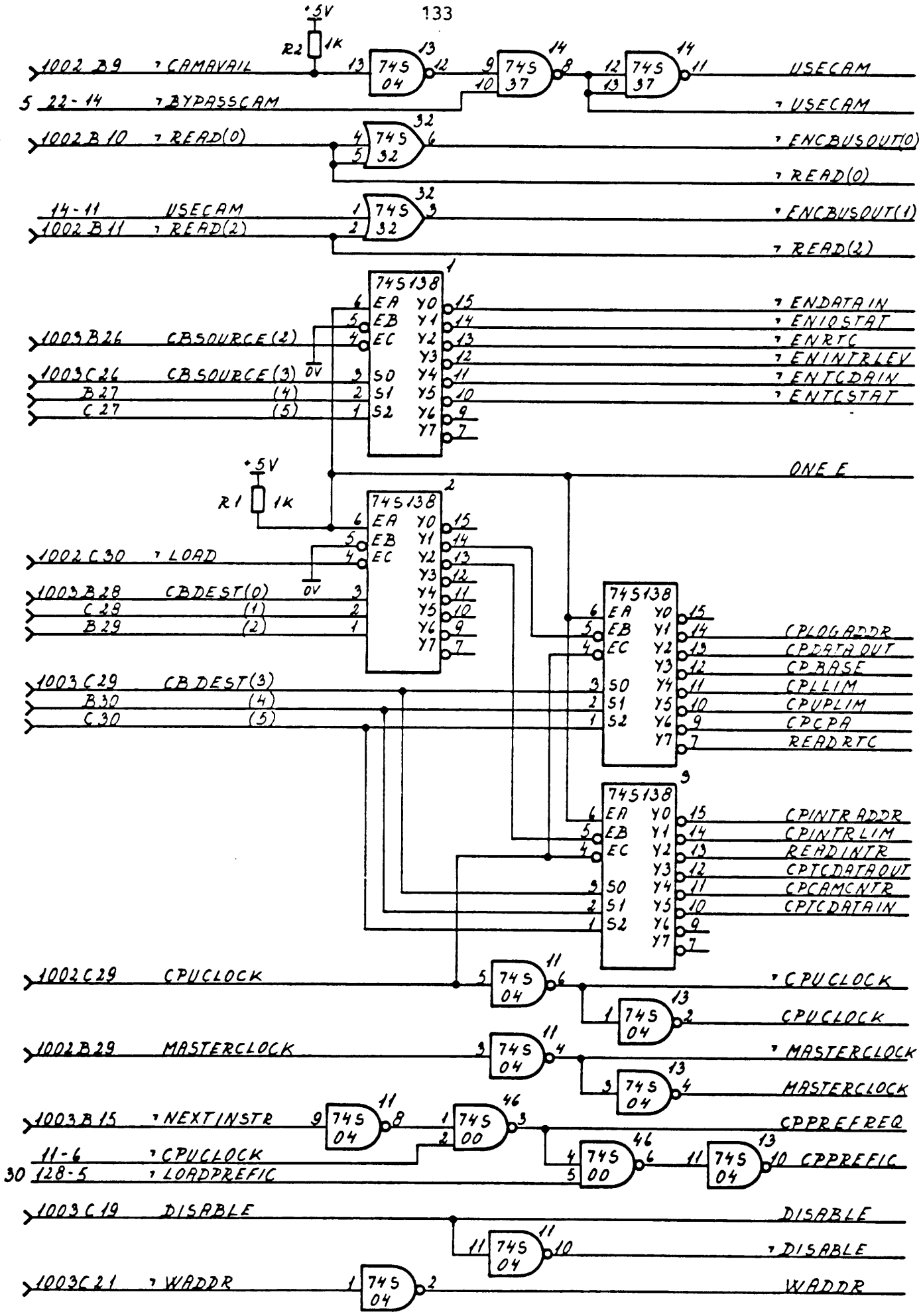



RCA 800901
ARJ 8003 18

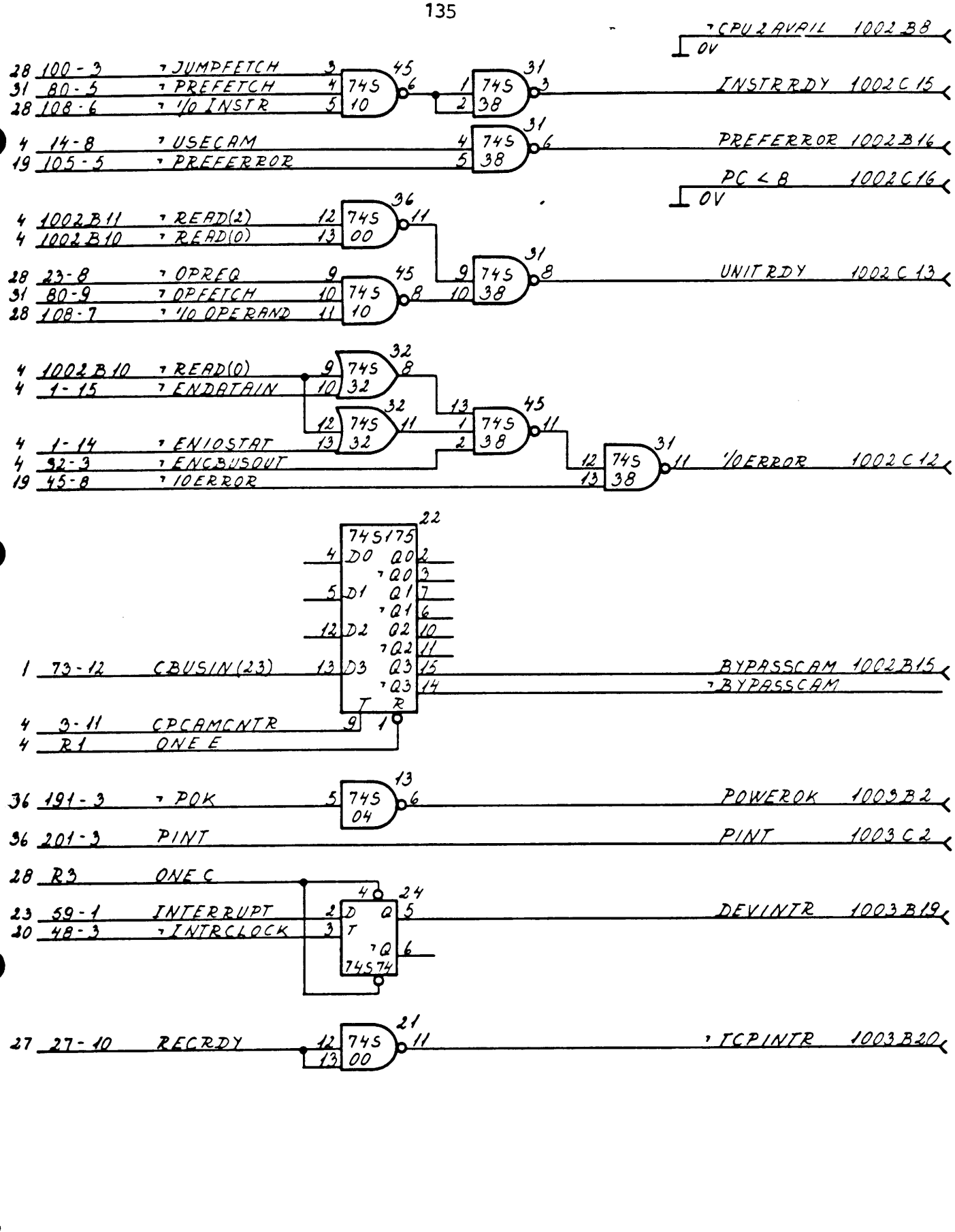



ACA 800901

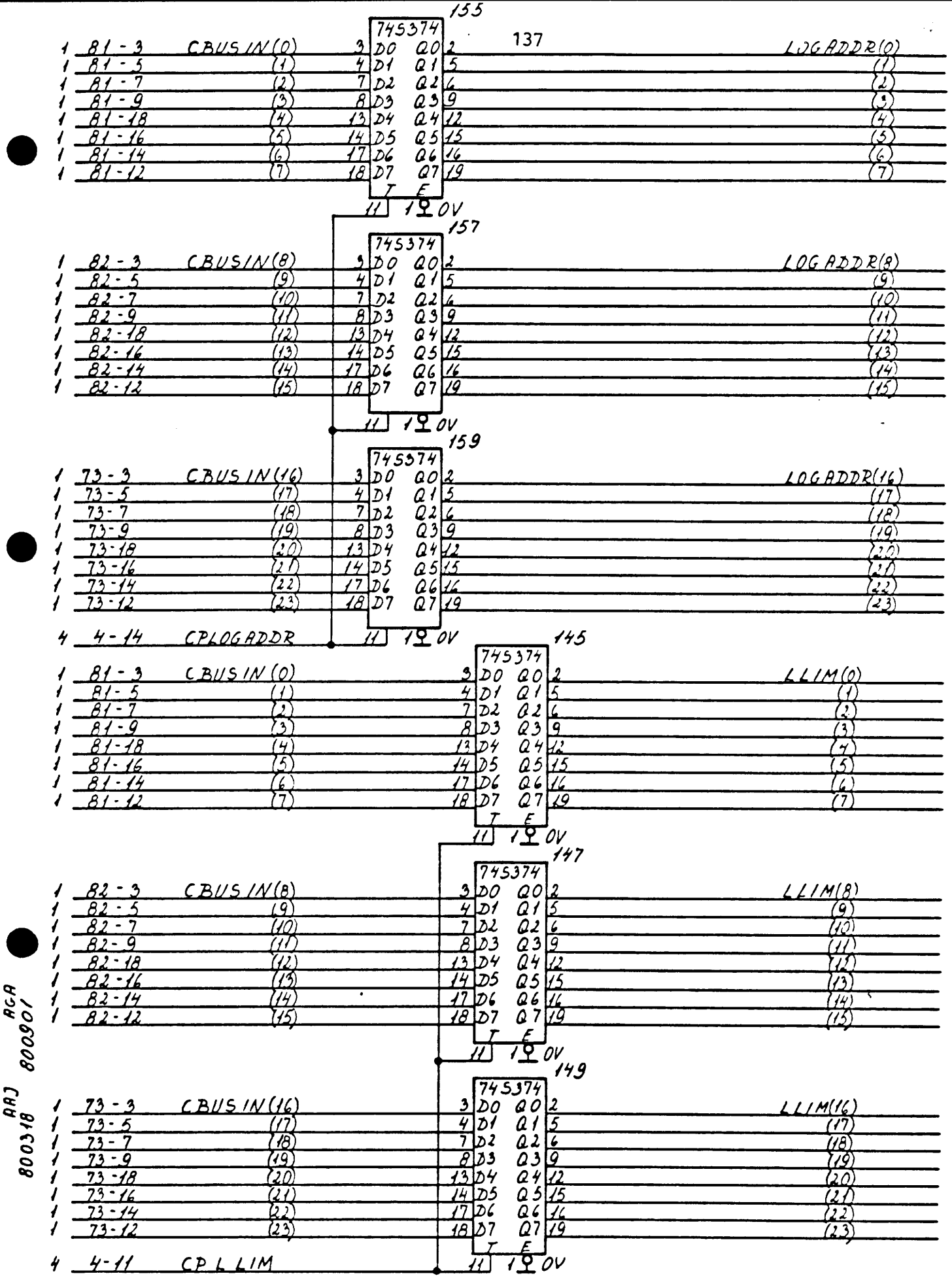
ADJ 800318

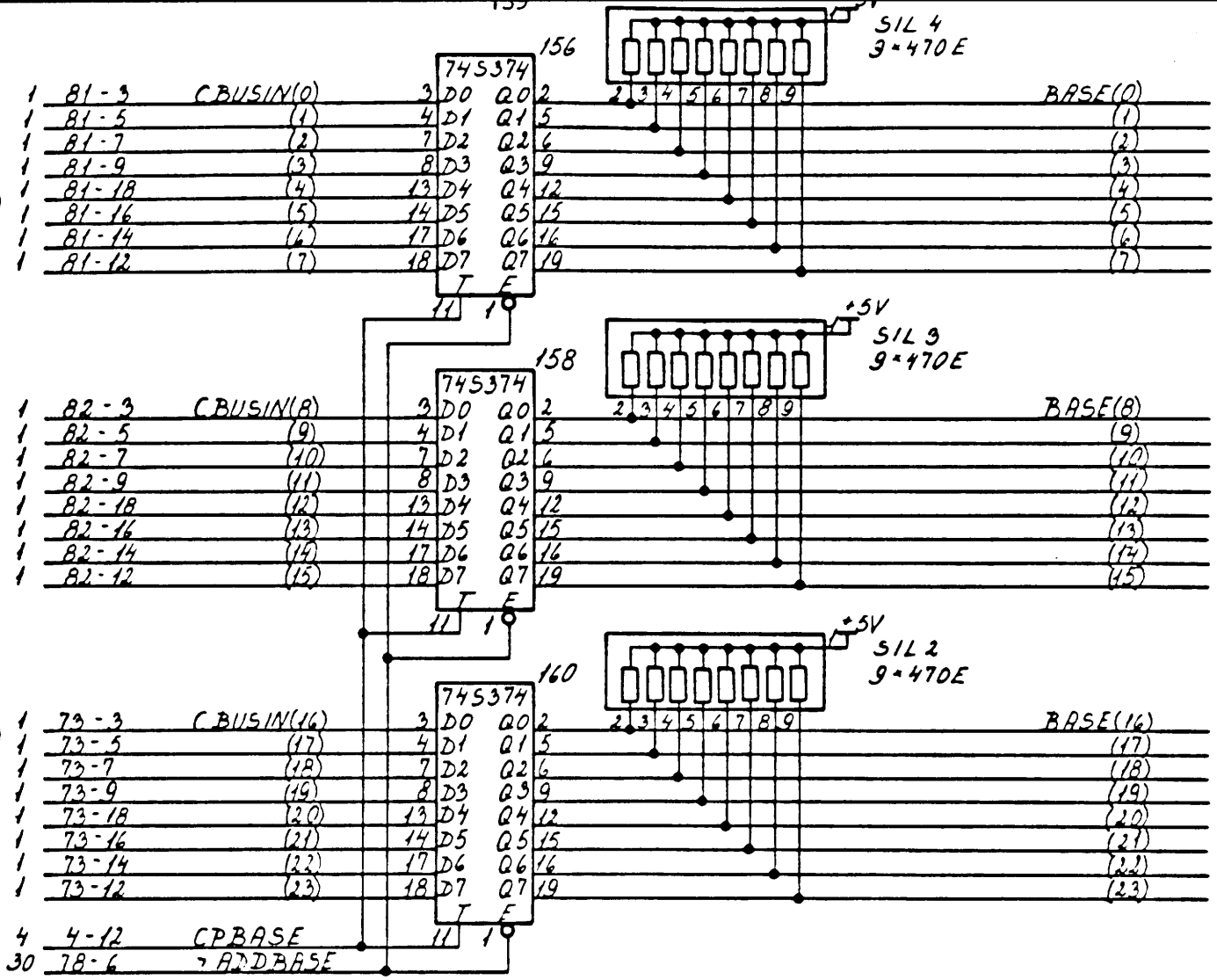



AGA 800901
ARJ 800318

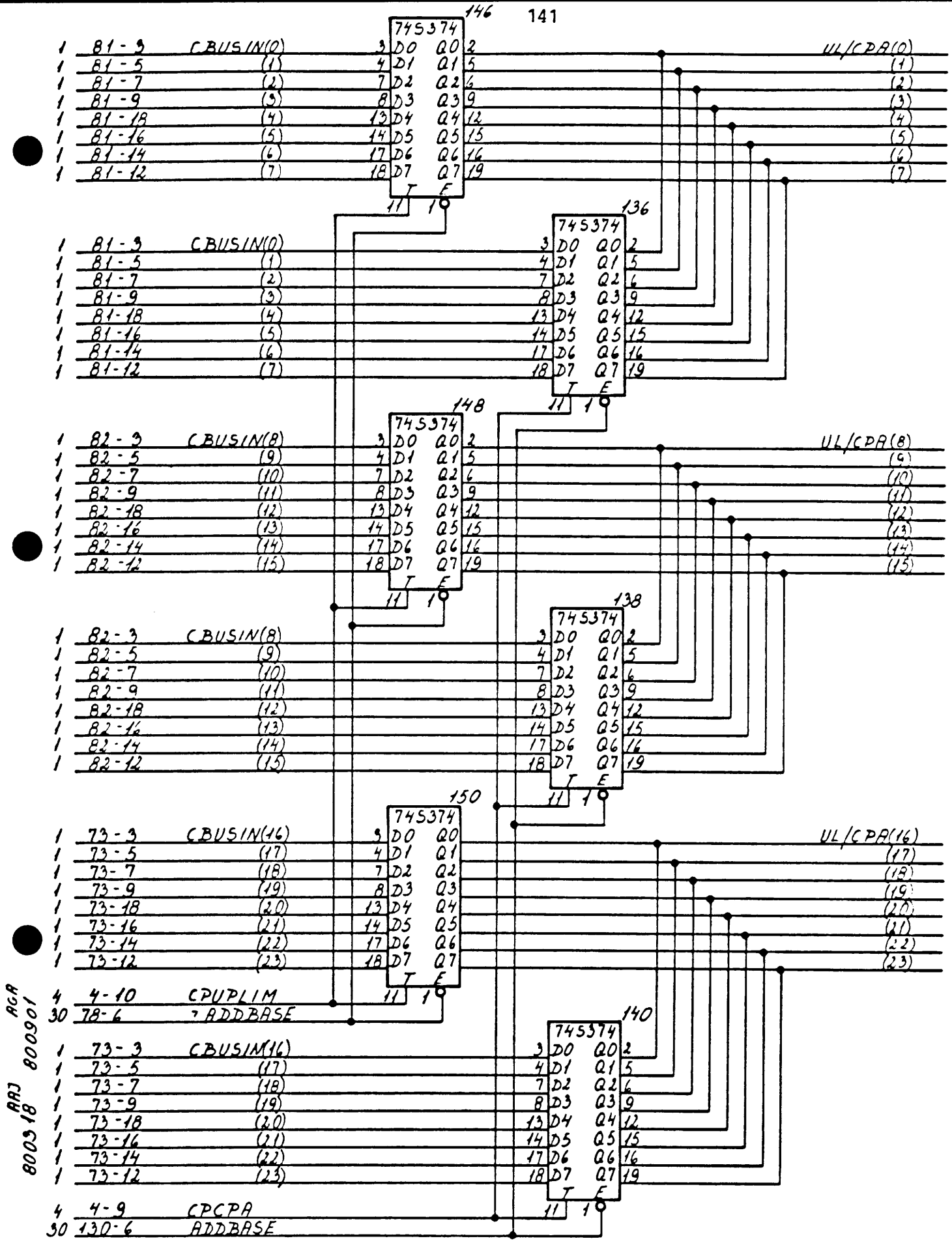


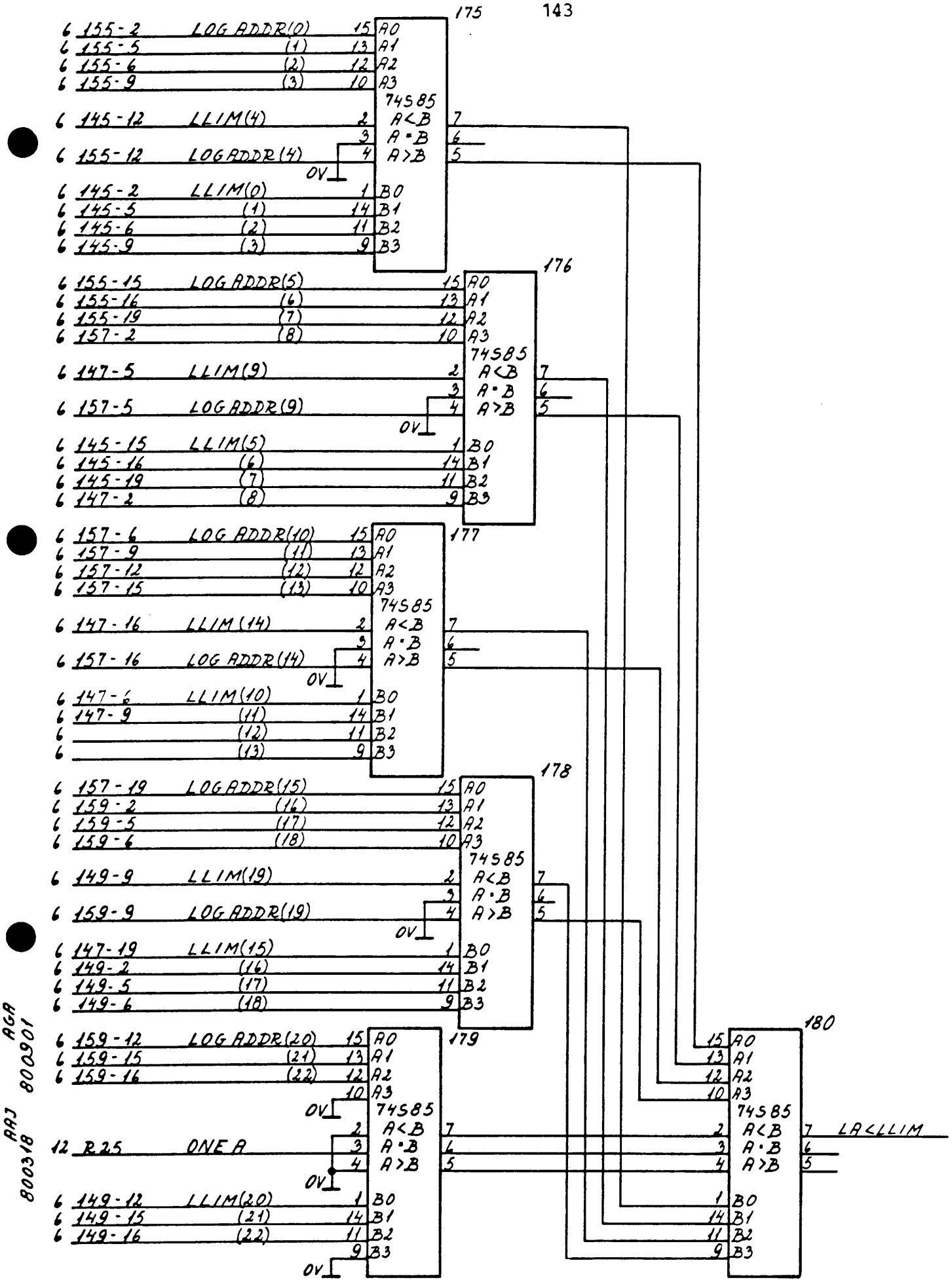

AGA 800901
AAJ 800318

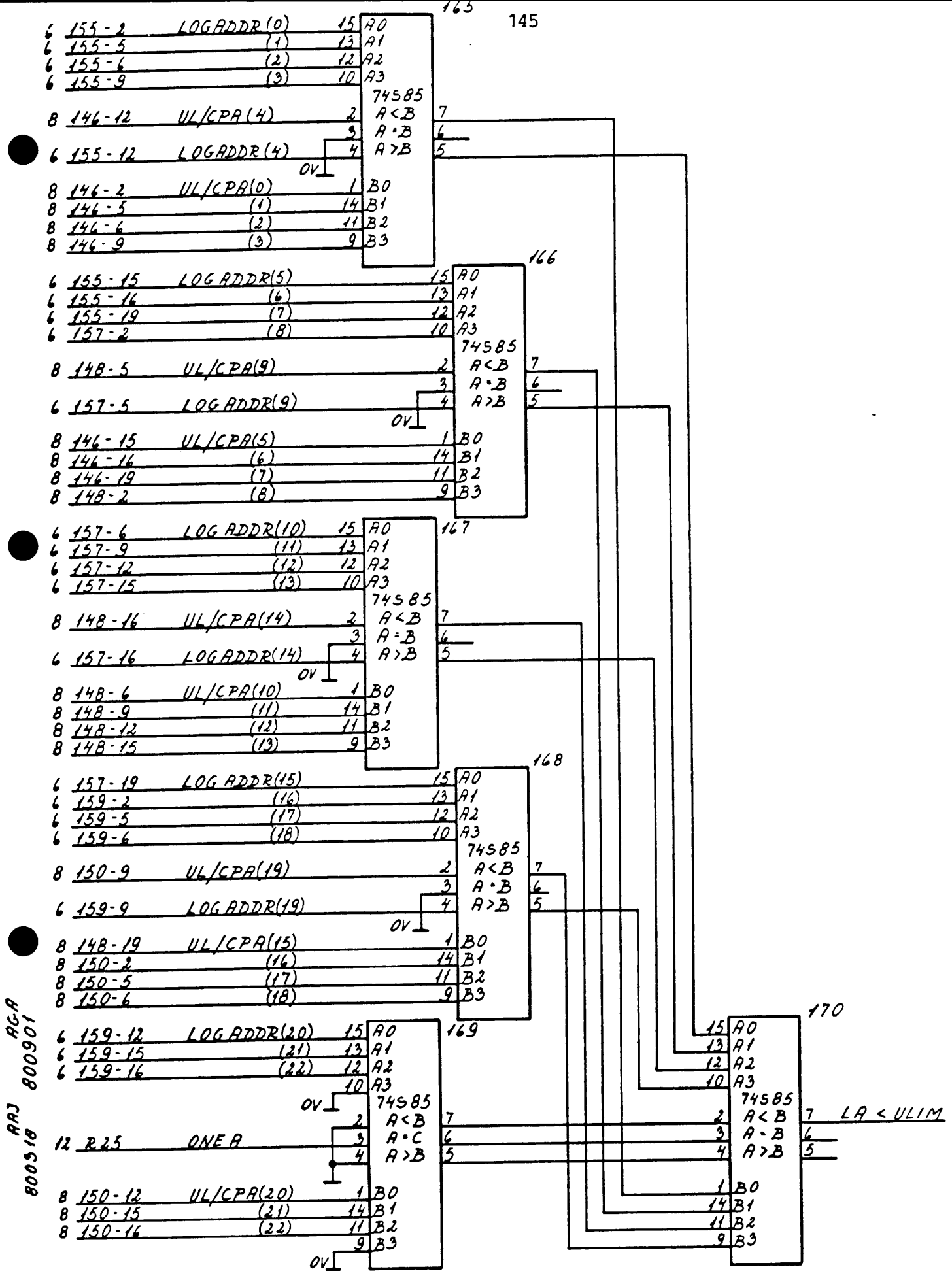

155
745374
137
1 81-3 CBUSIN(0) 3 D0 Q0 2 LOGADDR(0)
1 81-5 (1) 4 D1 Q1 5 (1)
1 81-7 (2) 7 D2 Q2 6 (2)
1 81-9 (3) 8 D3 Q3 9 (3)
1 81-18 (4) 13 D4 Q4 12 (4)
1 81-16 (5) 14 D5 Q5 15 (5)
1 81-14 (6) 17 D6 Q6 16 (6)
1 81-12 (7) 18 D7 Q7 19 (7)
T E
11 10V
157
745374
1 82-3 CBUSIN(8) 3 D0 Q0 2 LOGADDR(8)
1 82-5 (9) 4 D1 Q1 5 (9)
1 82-7 (10) 7 D2 Q2 6 (10)
1 82-9 (11) 8 D3 Q3 9 (11)
1 82-18 (12) 13 D4 Q4 12 (12)
1 82-16 (13) 14 D5 Q5 15 (13)
1 82-14 (14) 17 D6 Q6 16 (14)
1 82-12 (15) 18 D7 Q7 19 (15)
11 10V
159
745374
1 73-3 CBUSIN(16) 3 D0 Q0 2 LOGADDR(16)
1 73-5 (17) 4 D1 Q1 5 (17)
1 73-7 (18) 7 D2 Q2 6 (18)
1 73-9 (19) 8 D3 Q3 9 (19)
1 73-18 (20) 13 D4 Q4 12 (20)
1 73-16 (21) 14 D5 Q5 15 (21)
1 73-14 (22) 17 D6 Q6 16 (22)
1 73-12 (23) 18 D7 Q7 19 (23)
4 4-14 CPLOGADDR 11 10V
145
745374
1 81-3 CBUSIN(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
11 10V
147
745374
1 82-3 CBUSIN(8) 3 D0 Q0 2 LLIM(8)
1 82-5 (9) 4 D1 Q1 5 (9)
1 82-7 (10) 7 D2 Q2 6 (10)
1 82-9 (11) 8 D3 Q3 9 (11)
1 82-18 (12) 13 D4 Q4 12 (12)
1 82-16 (13) 14 D5 Q5 15 (13)
1 82-14 (14) 17 D6 Q6 16 (14)
1 82-12 (15) 18 D7 Q7 19 (15)
T E
11 10V
149
745374
1 73-3 CBUSIN(16) 3 D0 Q0 2 LLIM(16)
1 73-5 (17) 4 D1 Q1 5 (17)
1 73-7 (18) 7 D2 Q2 6 (18)
1 73-9 (19) 8 D3 Q3 9 (19)
1 73-18 (20) 13 D4 Q4 12 (20)
1 73-16 (21) 14 D5 Q5 15 (21)
1 73-14 (22) 17 D6 Q6 16 (22)
1 73-12 (23) 18 D7 Q7 19 (23)
T E
4 4-11 CP L LIM 11 10V
AGA 800901
AAJ 800318




AGA 800901
ARJ 800318


146
745374
D0 Q0
D1 Q1
D2 Q2
D3 Q3
D4 Q4
D5 Q5
D6 Q6
D7 Q7
T E
1 B1-3 CBUSIN(0) 3
1 B1-5 (1) 4
1 B1-7 (2) 7
1 B1-9 (3) 8
1 B1-18 (4) 13
1 B1-16 (5) 14
1 B1-14 (6) 17
1 B1-12 (7) 18
11 1
UL/CPA(0) 2
(1) 5
(2) 6
(3) 9
(4) 12
(5) 15
(6) 16
(7) 19
136
1 B1-3 CBUSIN(0) 3
1 B1-5 (1) 4
1 B1-7 (2) 7
1 B1-9 (3) 8
1 B1-18 (4) 13
1 B1-16 (5) 14
1 B1-14 (6) 17
1 B1-12 (7) 18
11 1
148
1 B2-3 CBUSIN(8) 3
1 B2-5 (9) 4
1 B2-7 (10) 7
1 B2-9 (11) 8
1 B2-18 (12) 13
1 B2-16 (13) 14
1 B2-14 (14) 17
1 B2-12 (15) 18
11 1
UL/CPA(8) 2
(9) 5
(10) 6
(11) 9
(12) 12
(13) 15
(14) 16
(15) 19
138
1 B2-3 CBUSIN(8) 3
1 B2-5 (9) 4
1 B2-7 (10) 7
1 B2-9 (11) 8
1 B2-18 (12) 13
1 B2-16 (13) 14
1 B2-14 (14) 17
1 B2-12 (15) 18
11 1
150
1 73-3 CBUSIN(16) 3
1 73-5 (17) 4
1 73-7 (18) 7
1 73-9 (19) 8
1 73-18 (20) 13
1 73-16 (21) 14
1 73-14 (22) 17
1 73-12 (23) 18
11 1
UL/CPA(16) 2
(17) 5
(18) 6
(19) 9
(20) 12
(21) 15
(22) 16
(23) 19
4 4-10 CPUPLIM
30 78-6 ADDBASE
AGA 800901
AAJ 800318
140
1 73-3 CBUSIN(16) 3
1 73-5 (17) 4
1 73-7 (18) 7
1 73-9 (19) 8
1 73-18 (20) 13
1 73-16 (21) 14
1 73-14 (22) 17
1 73-12 (23) 18
11 1
4 4-9 CPCPA
30 130-6 ADDBASE

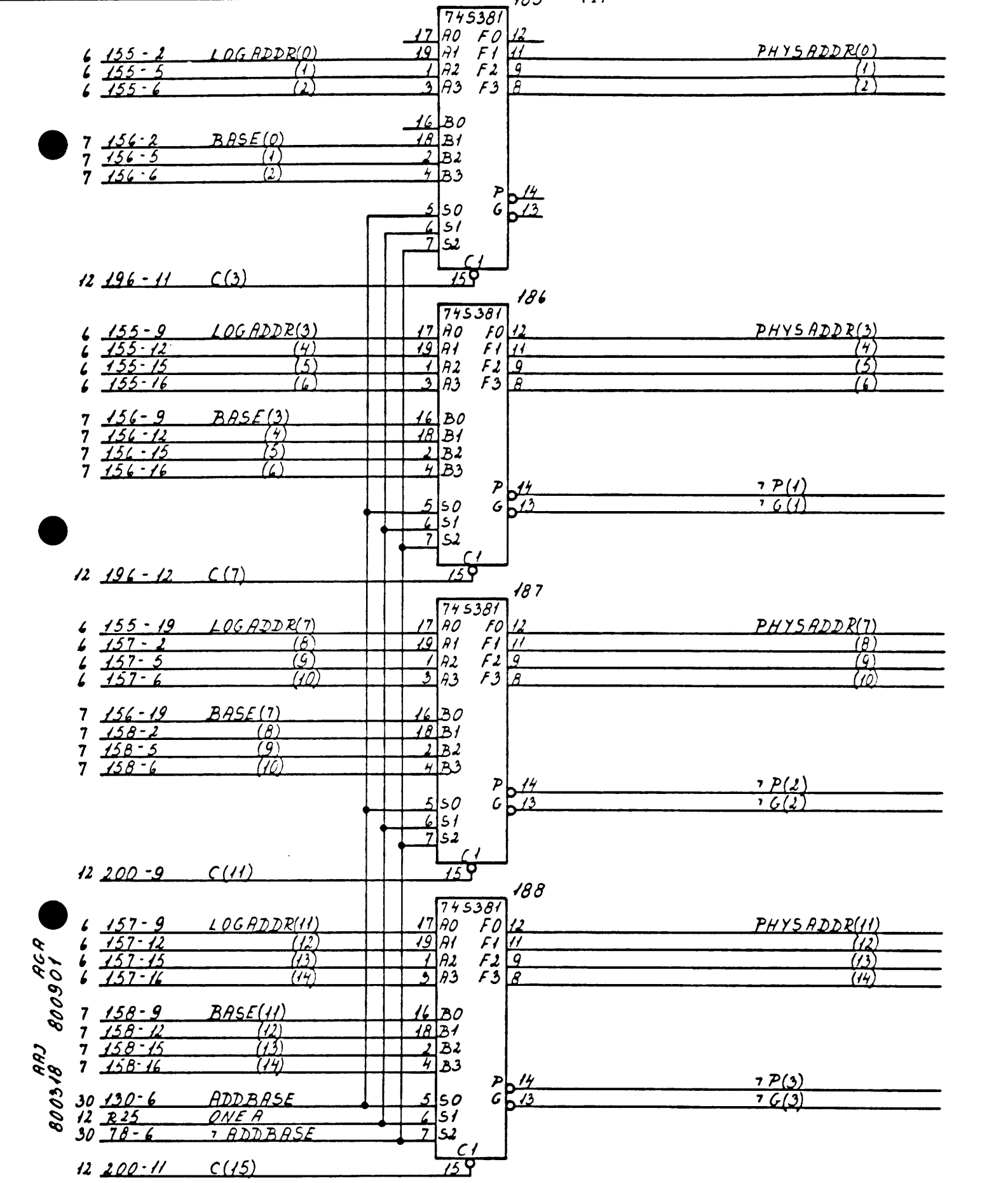

175
155-2 LOG ADDR(0) 15 A0
155-5 (1) 13 A1
155-6 (2) 12 A2
155-9 (3) 10 A3
74S85
145-12 LLIM(4) 2 A<B 7
3 A=B 6
155-12 LOG ADDR(4) 4 A>B 5
0V
145-2 LLIM(0) 1 B0
145-5 (1) 14 B1
145-6 (2) 11 B2
145-9 (3) 9 B3
176
155-15 LOG ADDR(5) 15 A0
155-16 (6) 13 A1
155-19 (7) 12 A2
157-2 (8) 10 A3
74S85
147-5 LLIM(9) 2 A<B 7
3 A=B 6
157-5 LOG ADDR(9) 4 A>B 5
0V
145-15 LLIM(5) 1 B0
145-16 (6) 14 B1
145-19 (7) 11 B2
147-2 (8) 9 B3
177
157-6 LOG ADDR(10) 15 A0
157-9 (11) 13 A1
157-12 (12) 12 A2
157-15 (13) 10 A3
74S85
147-16 LLIM(14) 2 A<B 7
3 A=B 6
157-16 LOG ADDR(14) 4 A>B 5
0V
147-6 LLIM(10) 1 B0
147-9 (11) 14 B1
(12) 11 B2
(13) 9 B3
178
157-19 LOG ADDR(15) 15 A0
159-2 (16) 13 A1
159-5 (17) 12 A2
159-6 (18) 10 A3
74S85
149-9 LLIM(19) 2 A<B 7
3 A=B 6
159-9 LOG ADDR(19) 4 A>B 5
0V
147-19 LLIM(15) 1 B0
149-2 (16) 14 B1
149-5 (17) 11 B2
149-6 (18) 9 B3
179
159-12 LOG ADDR(20) 15 A0
159-15 (21) 13 A1
159-16 (22) 12 A2
10 A3
0V
74S85
2 A<B 7
12 R25 ONE A 3 A=B 6
4 A>B 5
0V
149-12 LLIM(20) 1 B0
149-15 (21) 14 B1
149-16 (22) 11 B2
9 B3
0V
180
15 A0
13 A1
12 A2
10 A3
74S85
2 A<B 7 LA<LLIM
3 A=B 6
4 A>B 5
1 B0
14 B1
11 B2
9 B3
AGA 800901
AAJ 800318

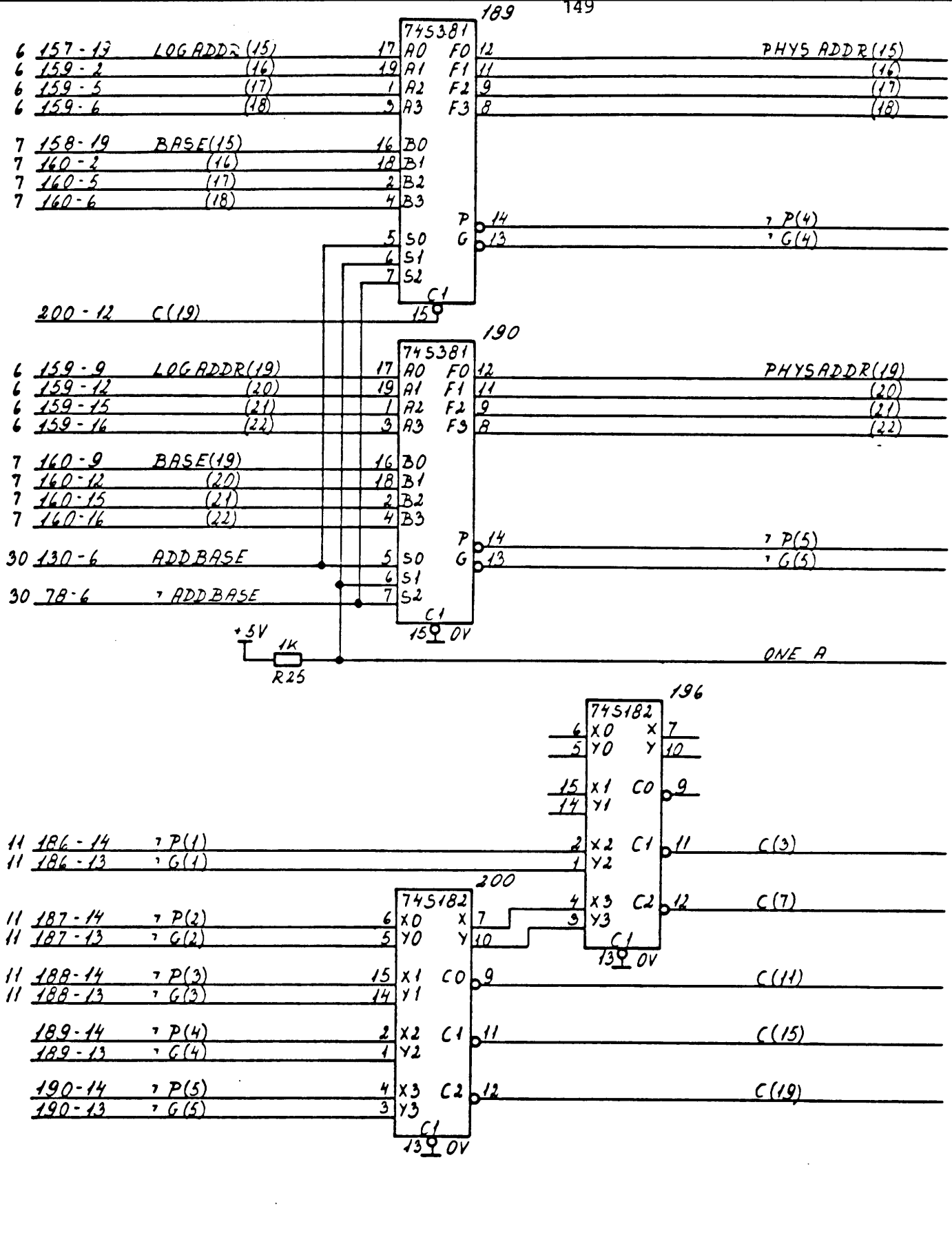



CPU 822

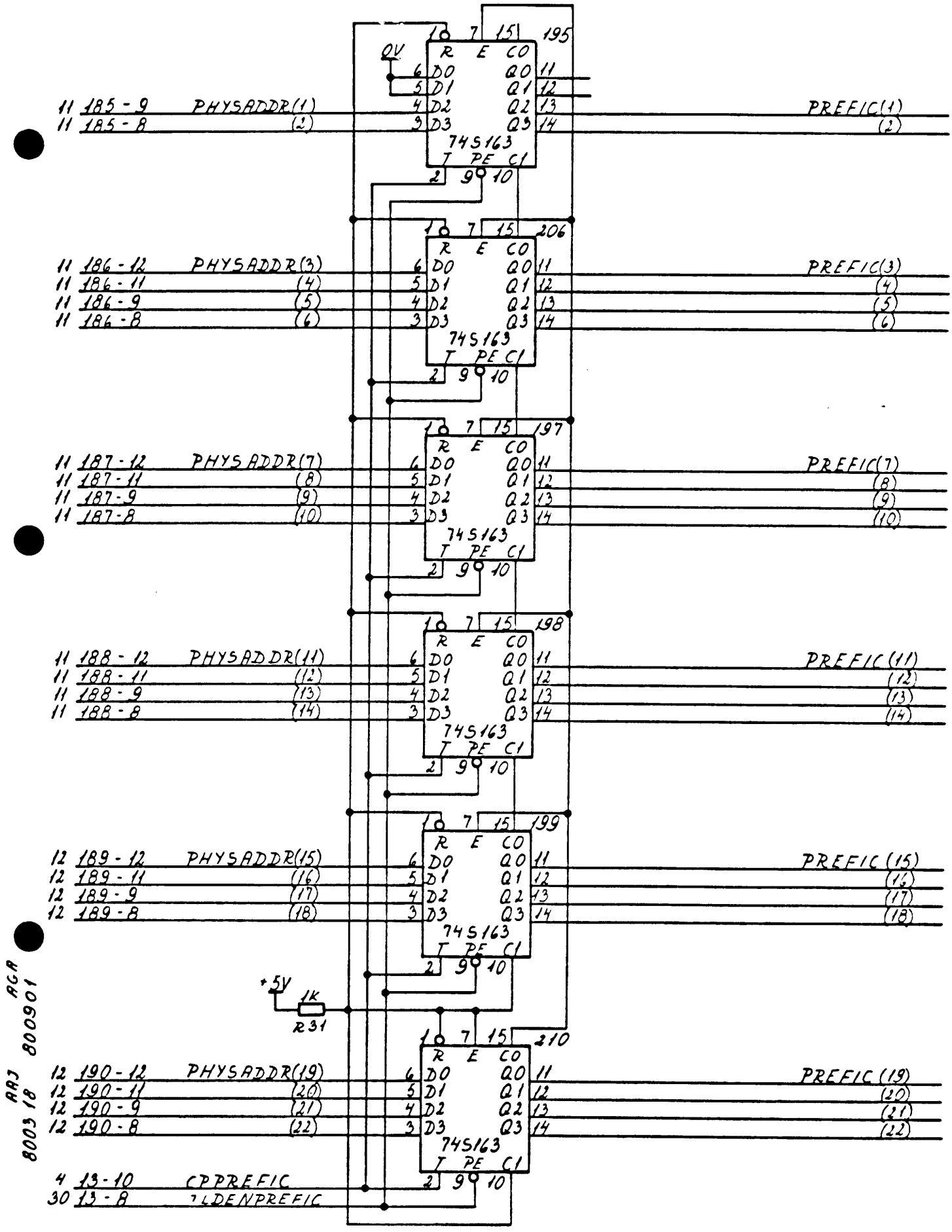
R 13024

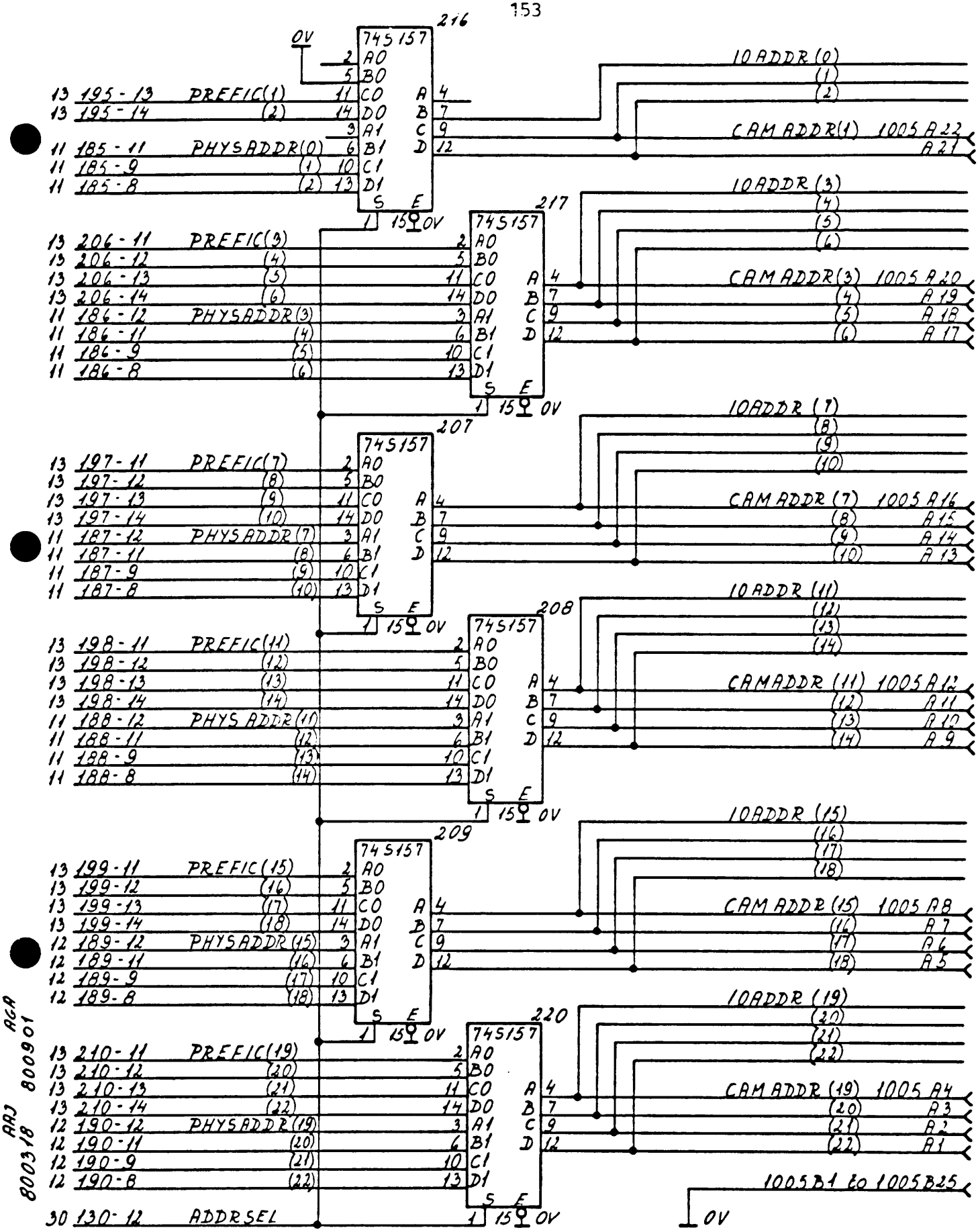
UPPER LIMIT & CPA LIMIT COMPERATOR

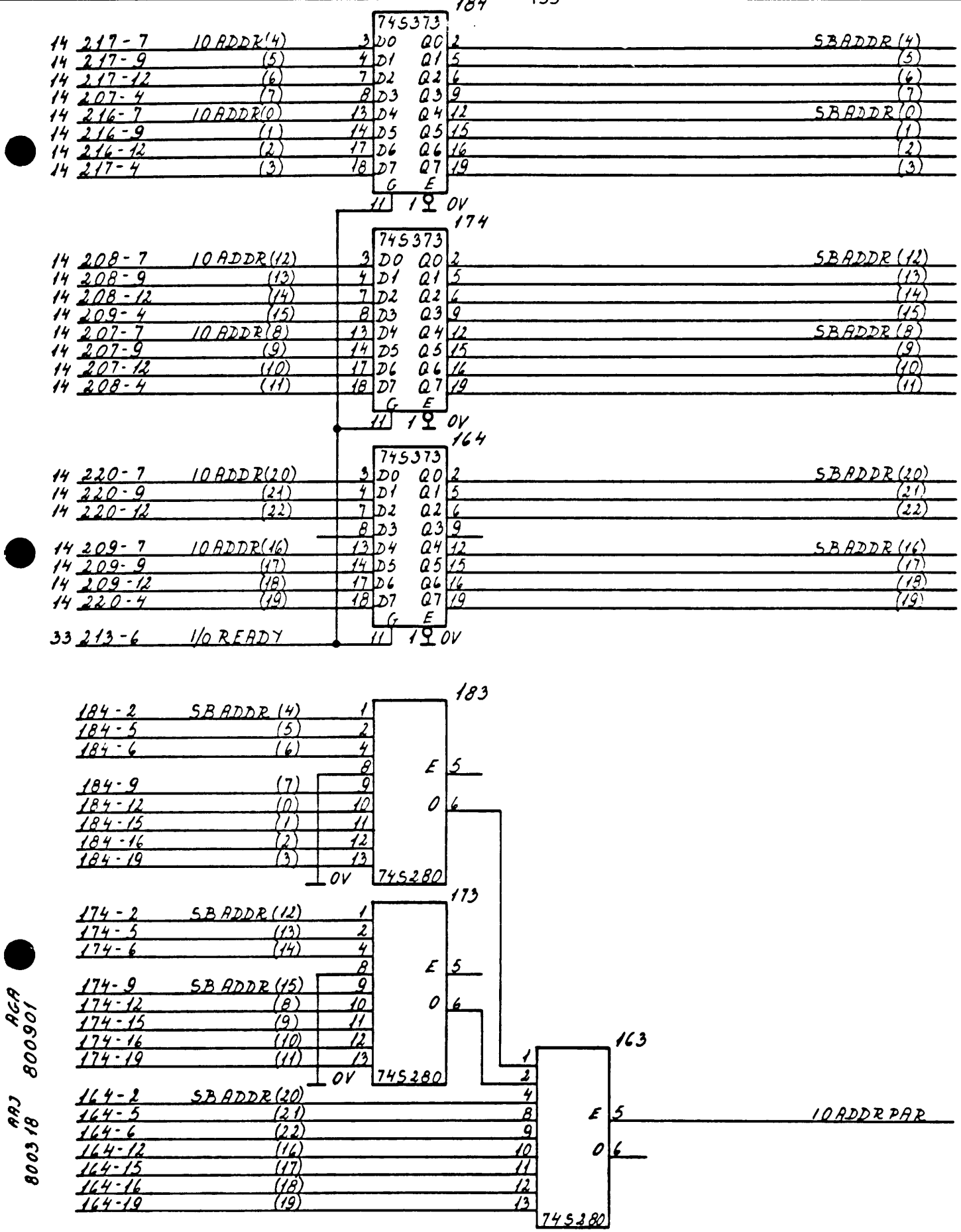

185
74S381
A0 A1 A2 A3
F0 F1 F2 F3
B0 B1 B2 B3
S0 S1 S2
C1
P G
6 155-2 LOGADDR(0)
6 155-5 (1)
6 155-6 (2)
7 156-2 BASE(0)
7 156-5 (1)
7 156-6 (2)
12 196-11 C(3)
PHYSADDR(0)
(1)
(2)
186
6 155-9 LOGADDR(3)
6 155-12 (4)
6 155-15 (5)
6 155-16 (6)
7 156-9 BASE(3)
7 156-12 (4)
7 156-15 (5)
7 156-16 (6)
12 196-12 C(7)
PHYSADDR(3)
(4)
(5)
(6)
7 P(1)
7 G(1)
187
6 155-19 LOGADDR(7)
6 157-2 (8)
6 157-5 (9)
6 157-6 (10)
7 156-19 BASE(7)
7 158-2 (8)
7 158-5 (9)
7 158-6 (10)
12 200-9 C(11)
PHYSADDR(7)
(8)
(9)
(10)
7 P(2)
7 G(2)
188
6 157-9 LOGADDR(11)
6 157-12 (12)
6 157-15 (13)
6 157-16 (14)
7 158-9 BASE(11)
7 158-12 (12)
7 158-15 (13)
7 158-16 (14)
30 130-6 ADDBASE
12 R25 ONE A
30 78-6 ↑ ADDBASE
12 200-11 C(15)
PHYSADDR(11)
(12)
(13)
(14)
7 P(3)
7 G(3)
AGA 800901
ARJ 800318

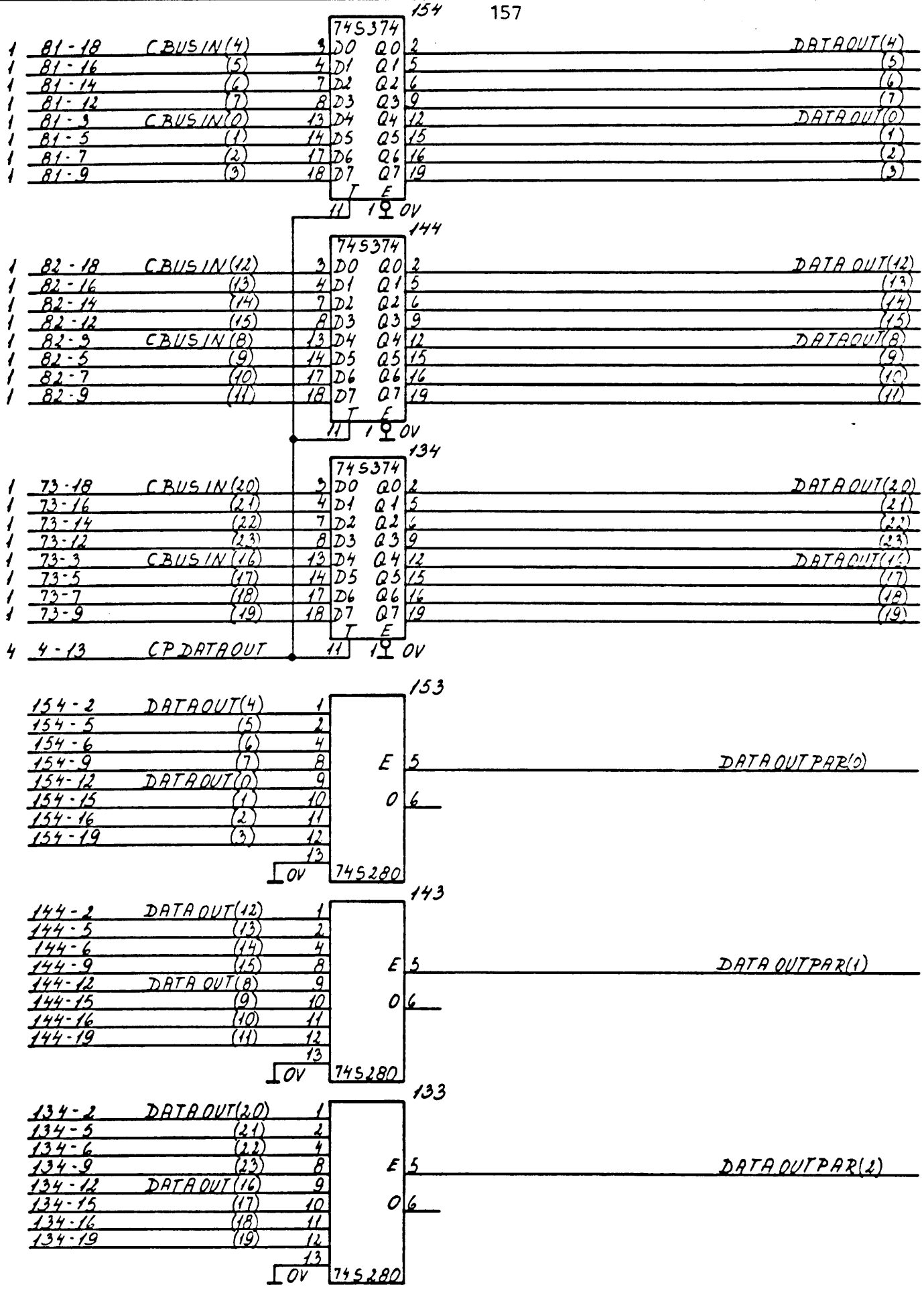



RGA 800901
AAJ 800318

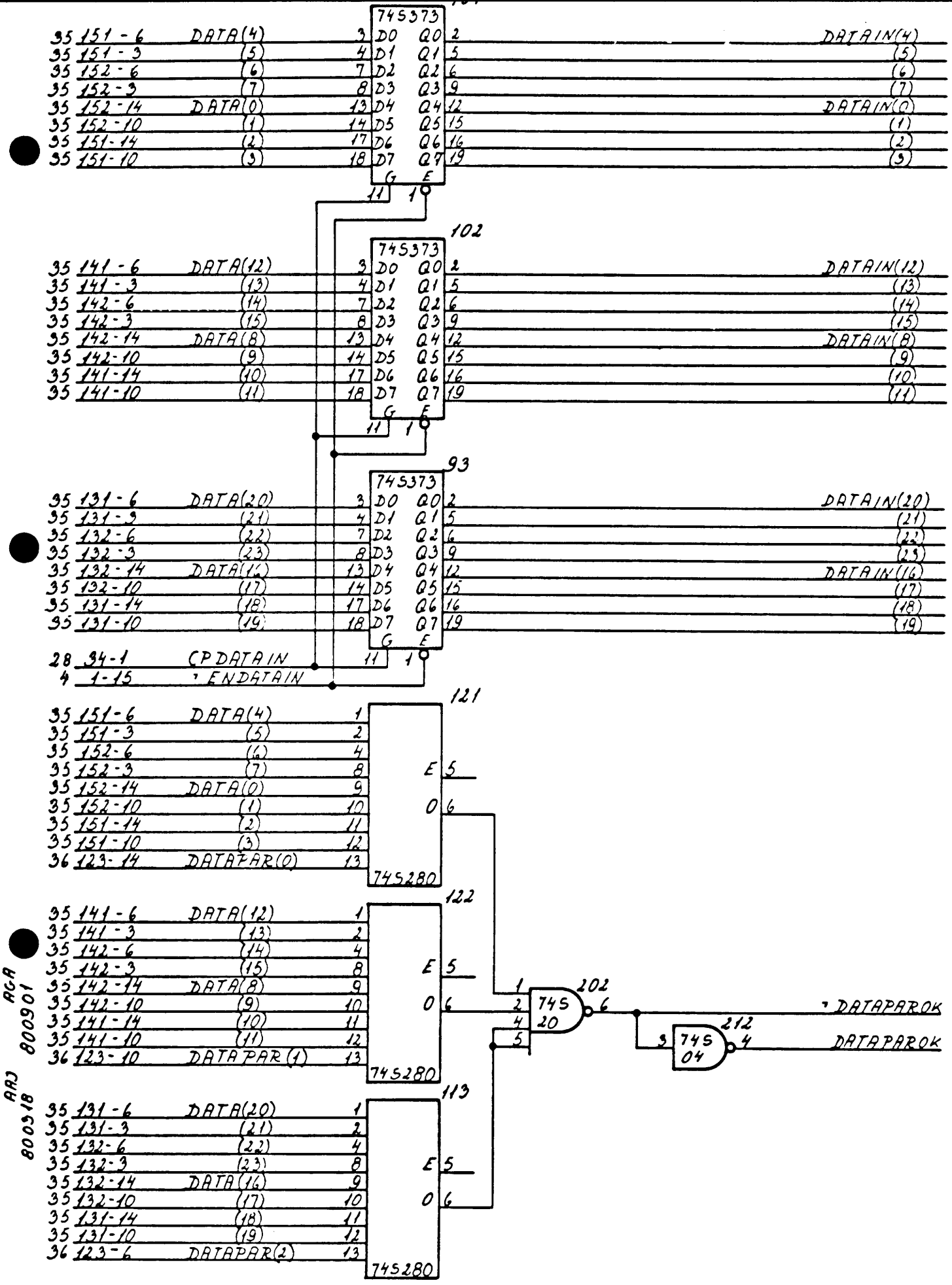

195
206
197
198
199
210
74S163
R E CO
T PE CI
DO D1 D2 D3
Q0 Q1 Q2 Q3
QV
11 185 - 9 PHYSADDR(1)
11 185 - 8 (2)
PREFIC(1)
(2)
11 186 - 12 PHYSADDR(3)
11 186 - 11 (4)
11 186 - 9 (5)
11 186 - 8 (6)
PREFIC(3)
(4)
(5)
(6)
11 187 - 12 PHYSADDR(7)
11 187 - 11 (8)
11 187 - 9 (9)
11 187 - 8 (10)
PREFIC(7)
(8)
(9)
(10)
11 188 - 12 PHYSADDR(11)
11 188 - 11 (12)
11 188 - 9 (13)
11 188 - 8 (14)
PREFIC(11)
(12)
(13)
(14)
12 189 - 12 PHYSADDR(15)
12 189 - 11 (16)
12 189 - 9 (17)
12 189 - 8 (18)
PREFIC(15)
(16)
(17)
(18)
+5V
1K
R31
12 190 - 12 PHYSADDR(19)
12 190 - 11 (20)
12 190 - 9 (21)
12 190 - 8 (22)
PREFIC(19)
(20)
(21)
(22)
4 13 - 10 CPPREFIC
30 13 - 8 LDENPREFIC
AGA 800901
AAJ 800318

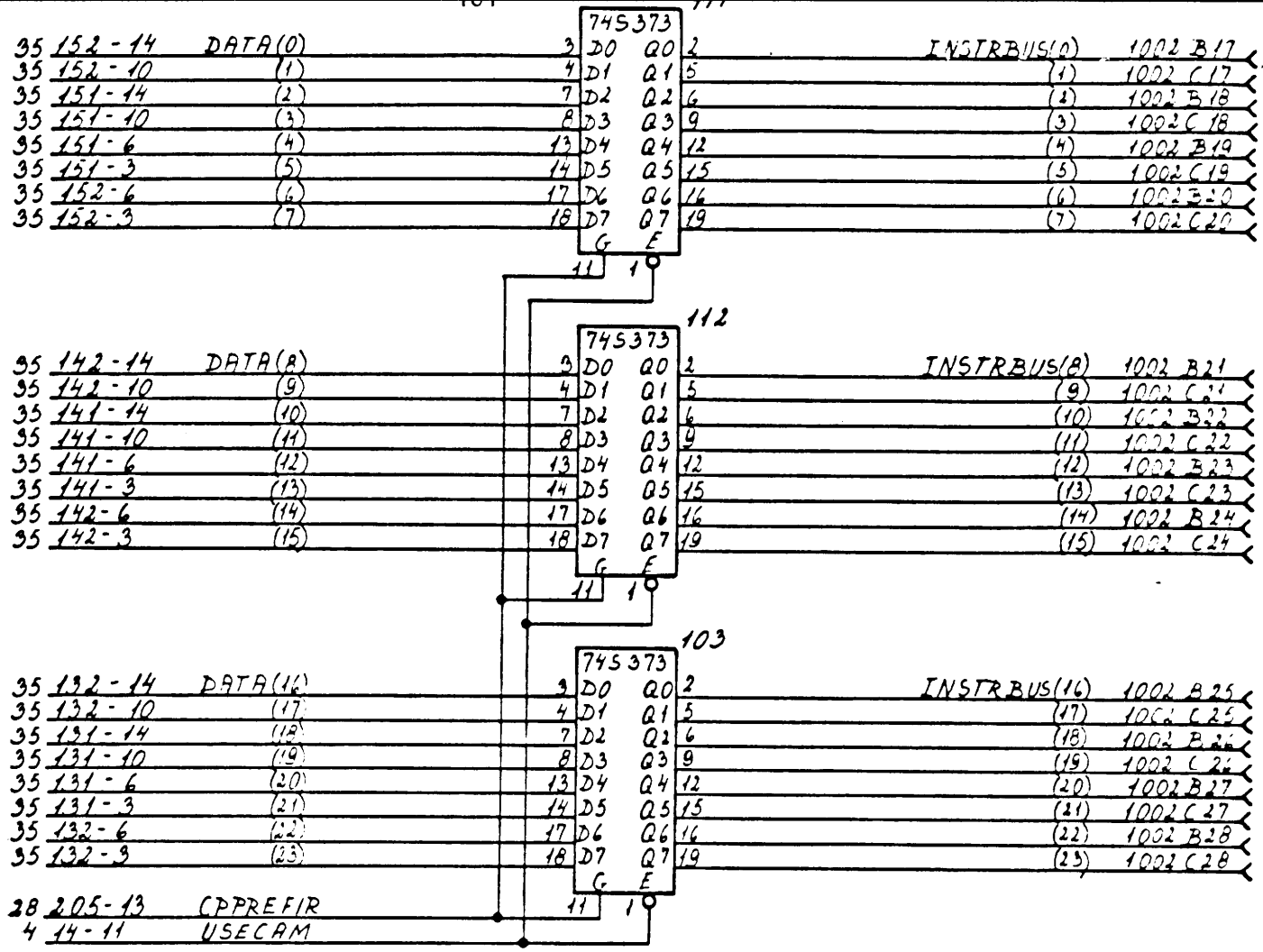

216
74S157
0V
2 A0
5 B0
11 C0
14 D0
3 A1
6 B1
10 C1
13 D1
S E
1 15 0V
A 4
B 7
C 9
D 12
13 195-13 PREFIC(1)
13 195-14 (2)
11 185-11 PHYSADDR(0)
11 185-9 (1)
11 185-8 (2)
10ADDR(0)
(1)
(2)
CAMADDR(1) 1005A22
A21
217
13 206-11 PREFIC(3)
13 206-12 (4)
13 206-13 (5)
13 206-14 (6)
11 186-12 PHYSADDR(3)
11 186-11 (4)
11 186-9 (5)
11 186-8 (6)
10ADDR(3)
(4)
(5)
(6)
CAMADDR(3) 1005A20
(4) A19
(5) A18
(6) A17
207
13 197-11 PREFIC(7)
13 197-12 (8)
13 197-13 (9)
13 197-14 (10)
11 187-12 PHYSADDR(7)
11 187-11 (8)
11 187-9 (9)
11 187-8 (10)
10ADDR(7)
(8)
(9)
(10)
CAMADDR(7) 1005A16
(8) A15
(9) A14
(10) A13
208
13 198-11 PREFIC(11)
13 198-12 (12)
13 198-13 (13)
13 198-14 (14)
11 188-12 PHYSADDR(11)
11 188-11 (12)
11 188-9 (13)
11 188-8 (14)
10ADDR(11)
(12)
(13)
(14)
CAMADDR(11) 1005A12
(12) A11
(13) A10
(14) A9
209
13 199-11 PREFIC(15)
13 199-12 (16)
13 199-13 (17)
13 199-14 (18)
12 189-12 PHYSADDR(15)
12 189-11 (16)
12 189-9 (17)
12 189-8 (18)
10ADDR(15)
(16)
(17)
(18)
CAMADDR(15) 1005A8
(16) A7
(17) A6
(18) A5
220
13 210-11 PREFIC(19)
13 210-12 (20)
13 210-13 (21)
13 210-14 (22)
12 190-12 PHYSADDR(19)
12 190-11 (20)
12 190-9 (21)
12 190-8 (22)
10ADDR(19)
(20)
(21)
(22)
CAMADDR(19) 1005A4
(20) A3
(21) A2
(22) A1
1005B1 to 1005B25
0V
30 130-12 ADDRSEL
AGA 800901
AAJ 800318


184
74S373
14 217-7 IOADDR(4) 3 D0 Q0 2 SBADDR(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 IOADDR(0) 13 D4 Q4 12 SBADDR(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
174
74S373
14 208-7 IOADDR(12) 3 D0 Q0 2 SBADDR(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 IOADDR(8) 13 D4 Q4 12 SBADDR(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
164
74S373
14 220-7 IOADDR(20) 3 D0 Q0 2 SBADDR(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 IOADDR(16) 13 D4 Q4 12 SBADDR(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
33 213-6 I/O READY 11 1 0V
183
184-2 SBADDR(4) 1
184-5 (5) 2
184-6 (6) 4
8 E 5
184-9 (7) 9
184-12 (0) 10 0 6
184-15 (1) 11
184-16 (2) 12
184-19 (3) 13
0V 74S280
173
174-2 SBADDR(12) 1
174-5 (13) 2
174-6 (14) 4
8 E 5
174-9 SBADDR(15) 9
174-12 (8) 10 0 6
174-15 (9) 11
174-16 (10) 12
174-19 (11) 13
0V 74S280
163
1
2
164-2 SBADDR(20) 4
164-5 (21) 8 E 5 IOADDRPAR
164-6 (22) 9
164-12 (16) 10 0 6
164-15 (17) 11
164-16 (18) 12
164-19 (19) 13
74S280
AGA 800901
AAJ 800318



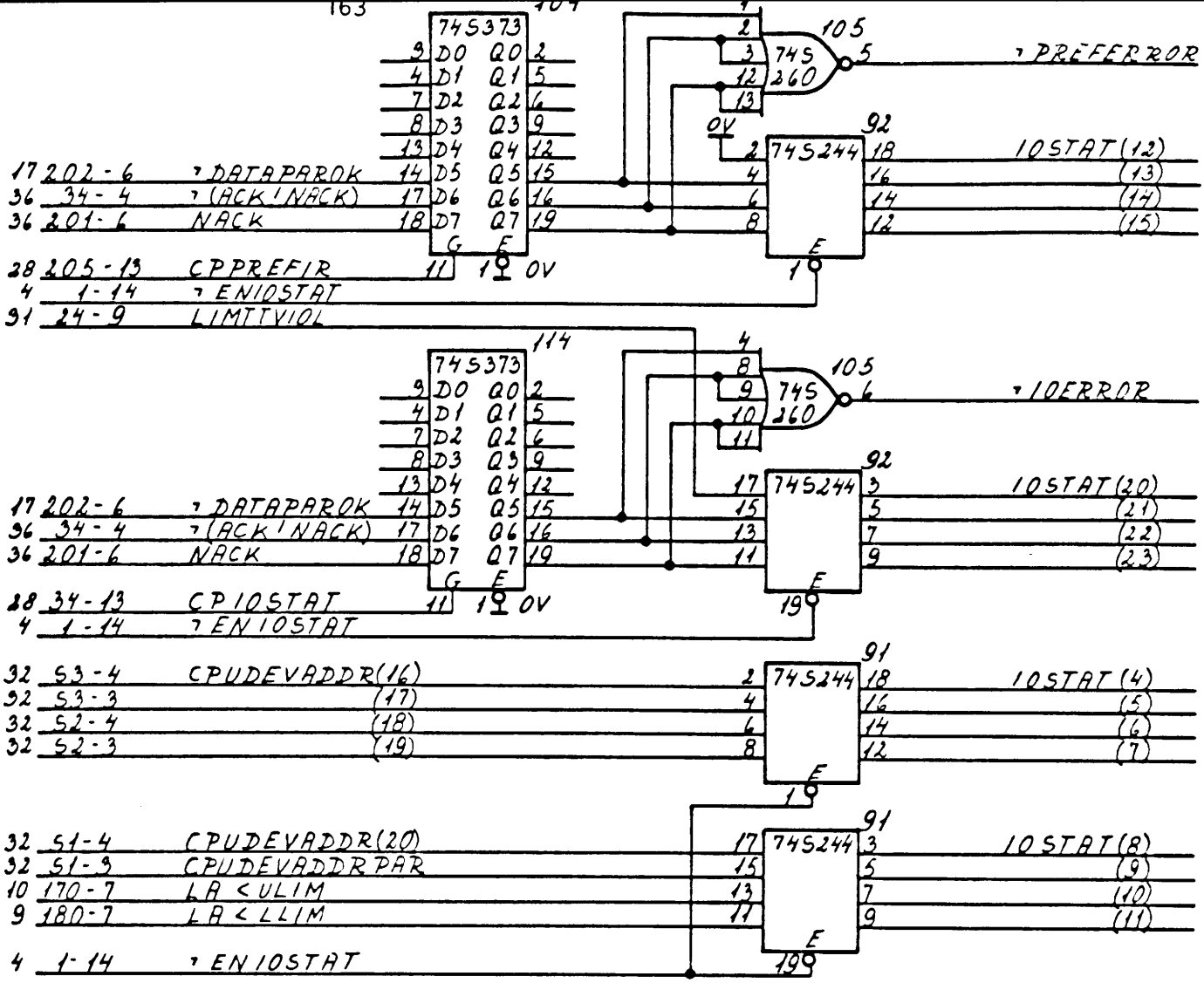

AGA 800901
AAJ 800318


101
74S373
35 151-6 DATA(4) 3 D0 Q0 2 DATAIN(4)
35 151-3 (5) 4 D1 Q1 5 (5)
35 152-6 (6) 7 D2 Q2 6 (6)
35 152-3 (7) 8 D3 Q3 9 (7)
35 152-14 DATA(0) 13 D4 Q4 12 DATAIN(0)
35 152-10 (1) 14 D5 Q5 15 (1)
35 151-14 (2) 17 D6 Q6 16 (2)
35 151-10 (3) 18 D7 Q7 19 (3)
G E
11 1
102
74S373
35 141-6 DATA(12) 3 D0 Q0 2 DATAIN(12)
35 141-3 (13) 4 D1 Q1 5 (13)
35 142-6 (14) 7 D2 Q2 6 (14)
35 142-3 (15) 8 D3 Q3 9 (15)
35 142-14 DATA(8) 13 D4 Q4 12 DATAIN(8)
35 142-10 (9) 14 D5 Q5 15 (9)
35 141-14 (10) 17 D6 Q6 16 (10)
35 141-10 (11) 18 D7 Q7 19 (11)
G E
11 1
93
74S373
35 131-6 DATA(20) 3 D0 Q0 2 DATAIN(20)
35 131-9 (21) 4 D1 Q1 5 (21)
35 132-6 (22) 7 D2 Q2 6 (22)
35 132-3 (23) 8 D3 Q3 9 (23)
35 132-14 DATA(16) 13 D4 Q4 12 DATAIN(16)
35 132-10 (17) 14 D5 Q5 15 (17)
35 131-14 (18) 17 D6 Q6 16 (18)
35 131-10 (19) 18 D7 Q7 19 (19)
G E
11 1
28 34-1 CPDATAIN
4 1-15 *ENDATAIN
121
35 151-6 DATA(4) 1
35 151-3 (5) 2
35 152-6 (6) 4
35 152-3 (7) 8 E 5
35 152-14 DATA(0) 9
35 152-10 (1) 10 O 6
35 151-14 (2) 11
35 151-10 (3) 12
36 123-14 DATAPAR(0) 13
74S280
122
35 141-6 DATA(12) 1
35 141-3 (13) 2
35 142-6 (14) 4
35 142-3 (15) 8 E 5
35 142-14 DATA(8) 9
35 142-10 (9) 10 O 6
35 141-14 (10) 11
35 141-10 (11) 12
36 123-10 DATAPAR(1) 13
74S280
AGA 800901
113
35 131-6 DATA(20) 1
35 131-3 (21) 2
35 132-6 (22) 4
35 132-3 (23) 8 E 5
35 132-14 DATA(16) 9
35 132-10 (17) 10 O 6
35 131-14 (18) 11
35 131-10 (19) 12
36 123-6 DATAPAR(2) 13
74S280
AAJ 800318
202
1
2 74S 6
4 20
5
212
3 74S 4
04
*DATAPAROK
DATAPAROK

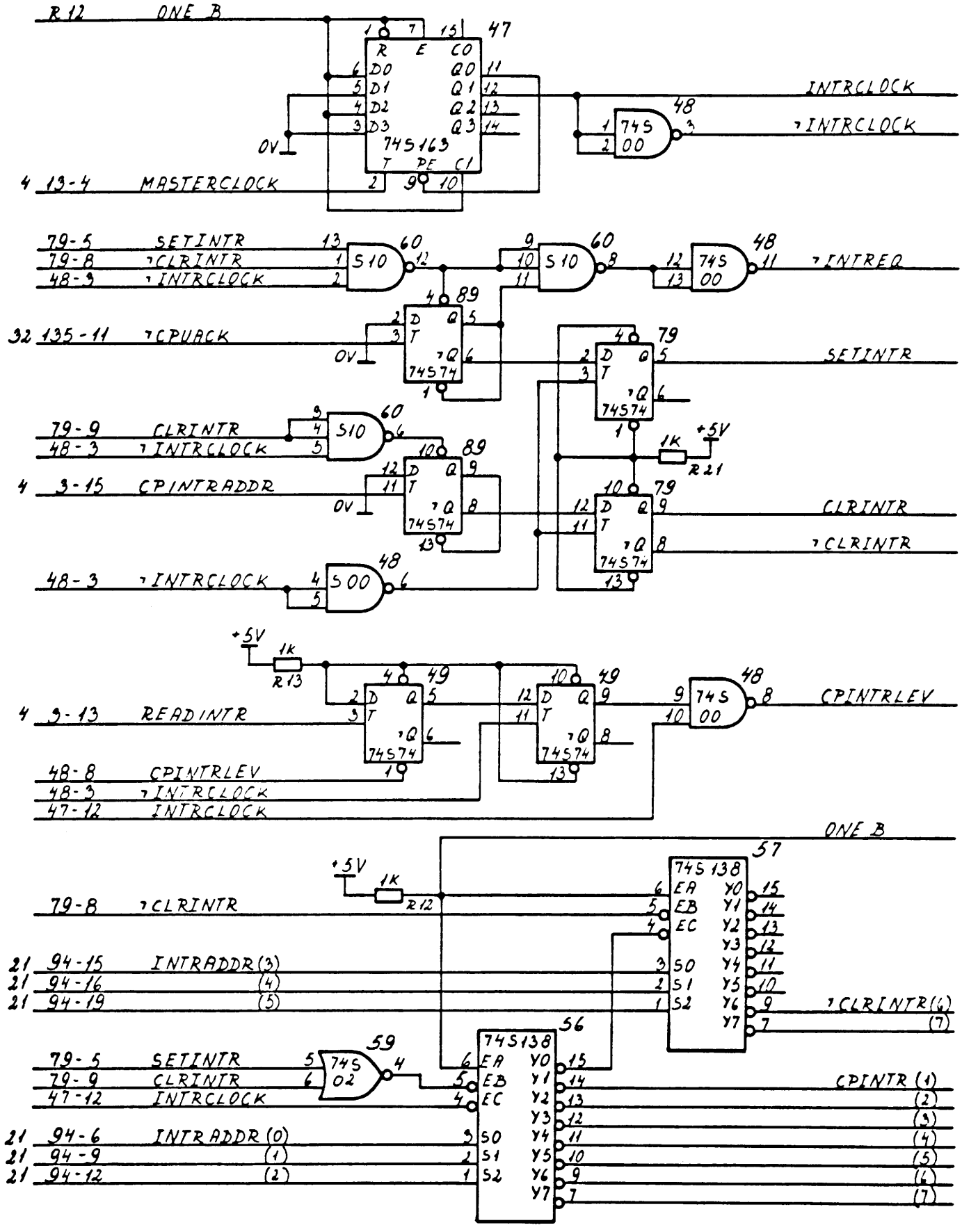



AGN 800901

ADJ 800318

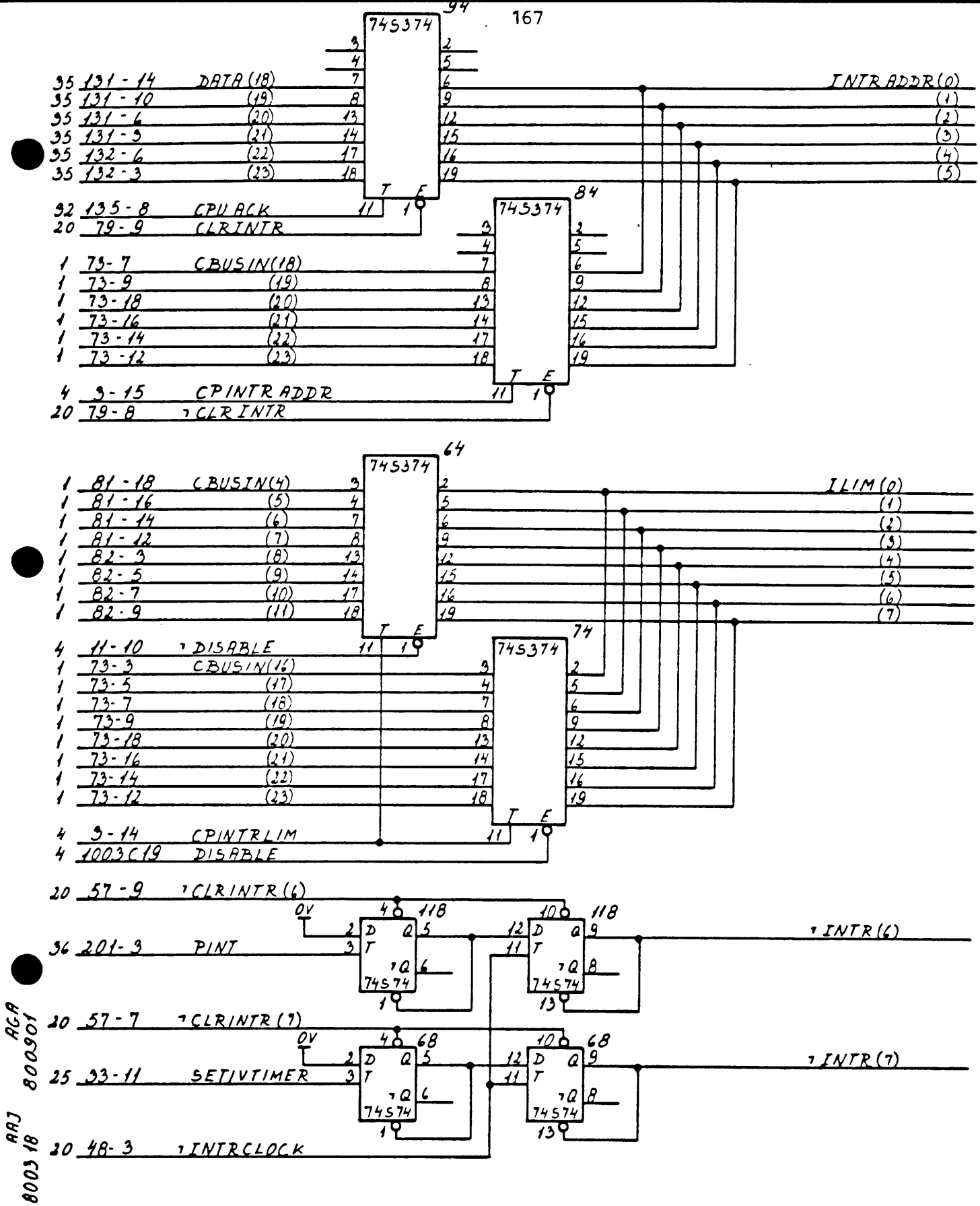


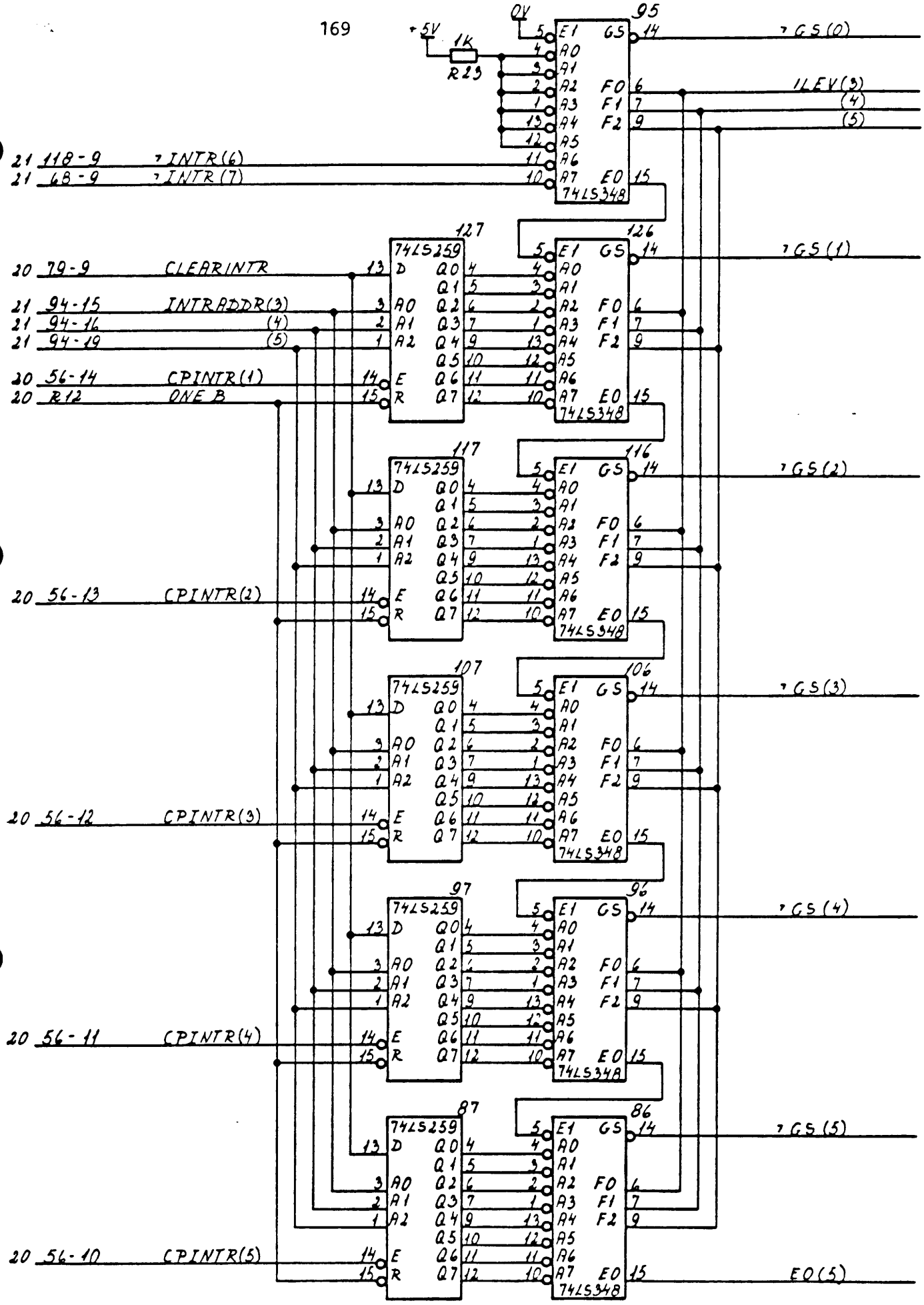

AGA 800901
AAJ 800318



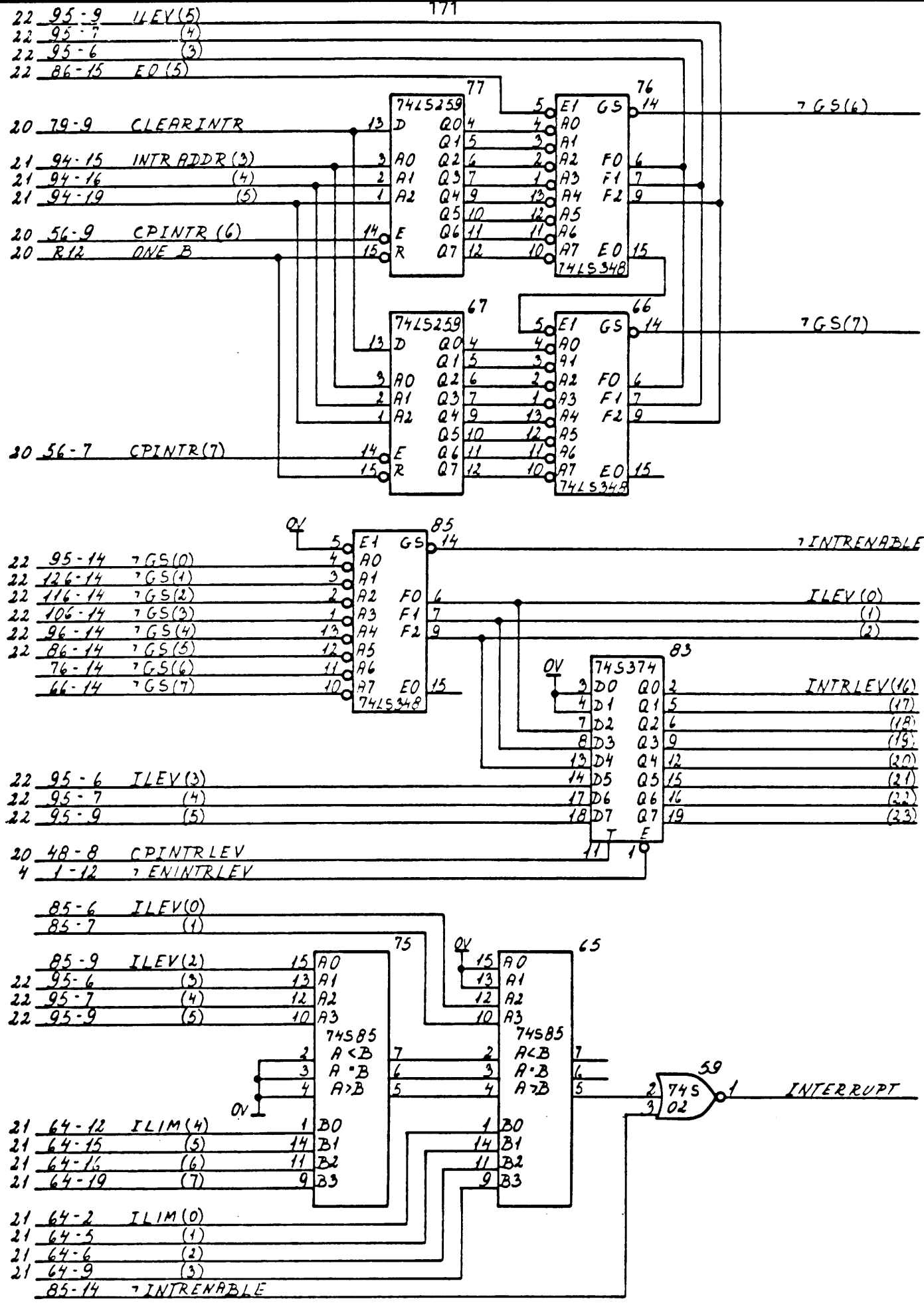

RGA 800901

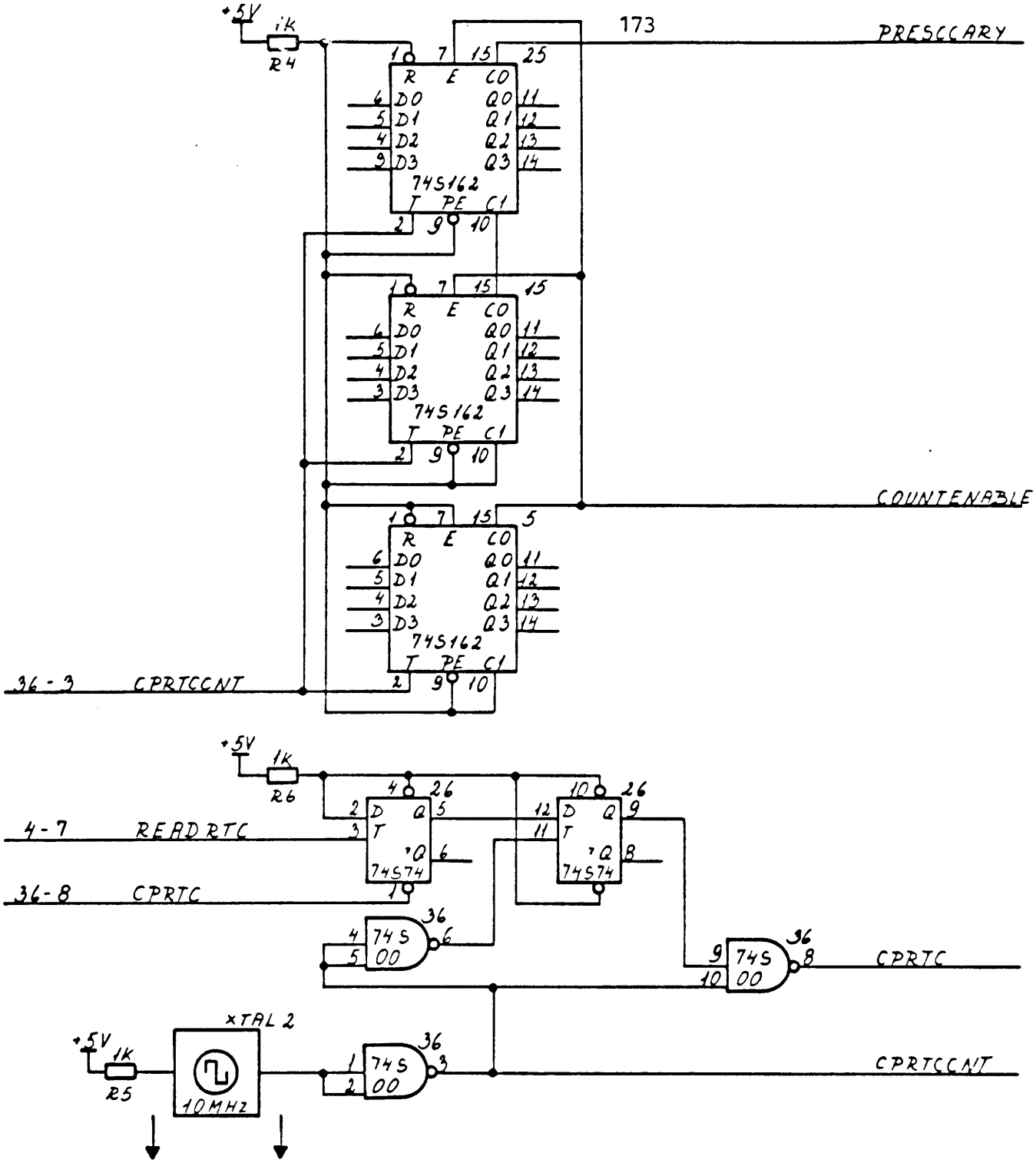
RAJ 800318


94
74S374
35 131-14 DATA(18)
35 131-10 (19)
35 131-6 (20)
35 131-3 (21)
35 132-6 (22)
35 132-3 (23)
INTRADDR(0)
(1)
(2)
(3)
(4)
(5)
92 135-8 CPU ACK
20 79-9 CLRINTR
84
1 73-7 CBUSIN(18)
1 73-9 (19)
1 73-18 (20)
1 73-16 (21)
1 73-14 (22)
1 73-12 (23)
4 9-15 CPINTR ADDR
20 79-8 ↑CLR INTR
64
1 81-18 CBUSIN(4)
1 81-16 (5)
1 81-14 (6)
1 81-12 (7)
1 82-3 (8)
1 82-5 (9)
1 82-7 (10)
1 82-9 (11)
ILIM(0)
(1)
(2)
(3)
(4)
(5)
(6)
(7)
4 11-10 ↑DISABLE
74
1 73-3 CBUSIN(16)
1 73-5 (17)
1 73-7 (18)
1 73-9 (19)
1 73-18 (20)
1 73-16 (21)
1 73-14 (22)
1 73-12 (23)
4 9-14 CPINTRLIM
4 1003C19 DISABLE
20 57-9 ↑CLRINTR(6)
OV
118
118
36 201-3 PINT
↑INTR(6)
74S74
20 57-7 ↑CLRINTR(7)
68
68
25 33-11 SETIVTIMER
↑INTR(7)
20 48-3 ↑INTRCLOCK
AGA 800901
AAJ 800318




AGA 800901
ARJ 800318

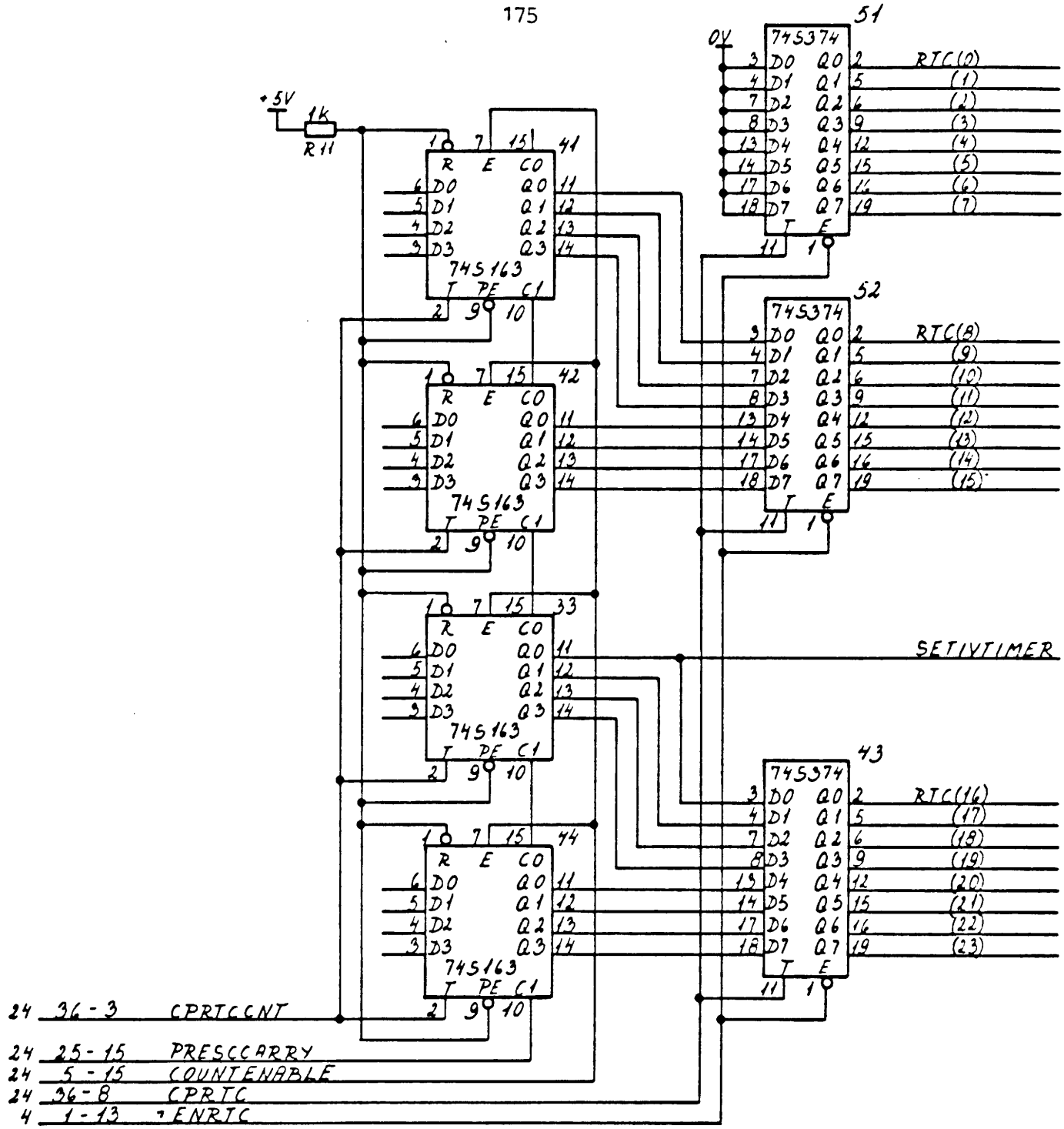

22 95-9 ILEV(5)
22 95-7 (4)
22 95-6 (3)
22 86-15 EO(5)
77
74LS259
76
74LS348
20 79-9 CLEARINTR
21 94-15 INTR ADDR(3)
21 94-16 (4)
21 94-19 (5)
20 56-9 CPINTR(6)
20 R12 ONE B
7 GS(6)
67
66
7 GS(7)
20 56-7 CPINTR(7)
QV
85
22 95-14 7 GS(0)
22 126-14 7 GS(1)
22 116-14 7 GS(2)
22 106-14 7 GS(3)
22 96-14 7 GS(4)
22 86-14 7 GS(5)
76-14 7 GS(6)
66-14 7 GS(7)
7 INTRENABLE
ILEV(0)
(1)
(2)
83
74S374
INTRLEV(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
22 95-6 ILEV(3)
22 95-7 (4)
22 95-9 (5)
20 48-8 CPINTRLEV
4 1-12 7 ENINTRLEV
85-6 ILEV(0)
85-7 (1)
75
65
85-9 ILEV(2)
22 95-6 (3)
22 95-7 (4)
22 95-9 (5)
74S85
59
74S 02
INTERRUPT
21 64-12 ILIM(4)
21 64-15 (5)
21 64-16 (6)
21 64-19 (7)
21 64-2 ILIM(0)
21 64-5 (1)
21 64-6 (2)
21 64-9 (3)
85-14 7 INTRENABLE
AGA 800901
AAJ 800318



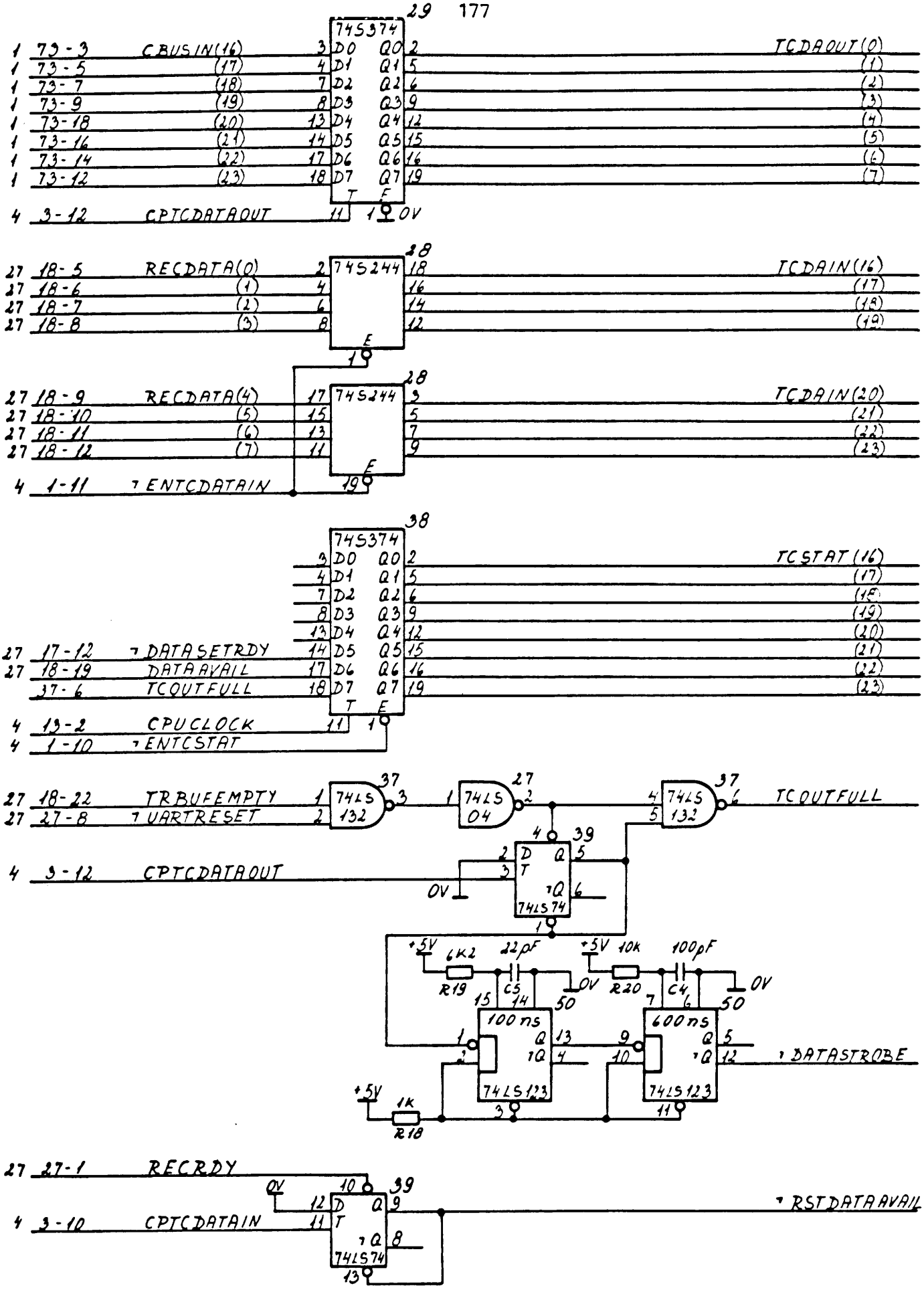

1 5 : XTAL DANTRONIC CCO·8

NC 3 : XTAL DALE-33A

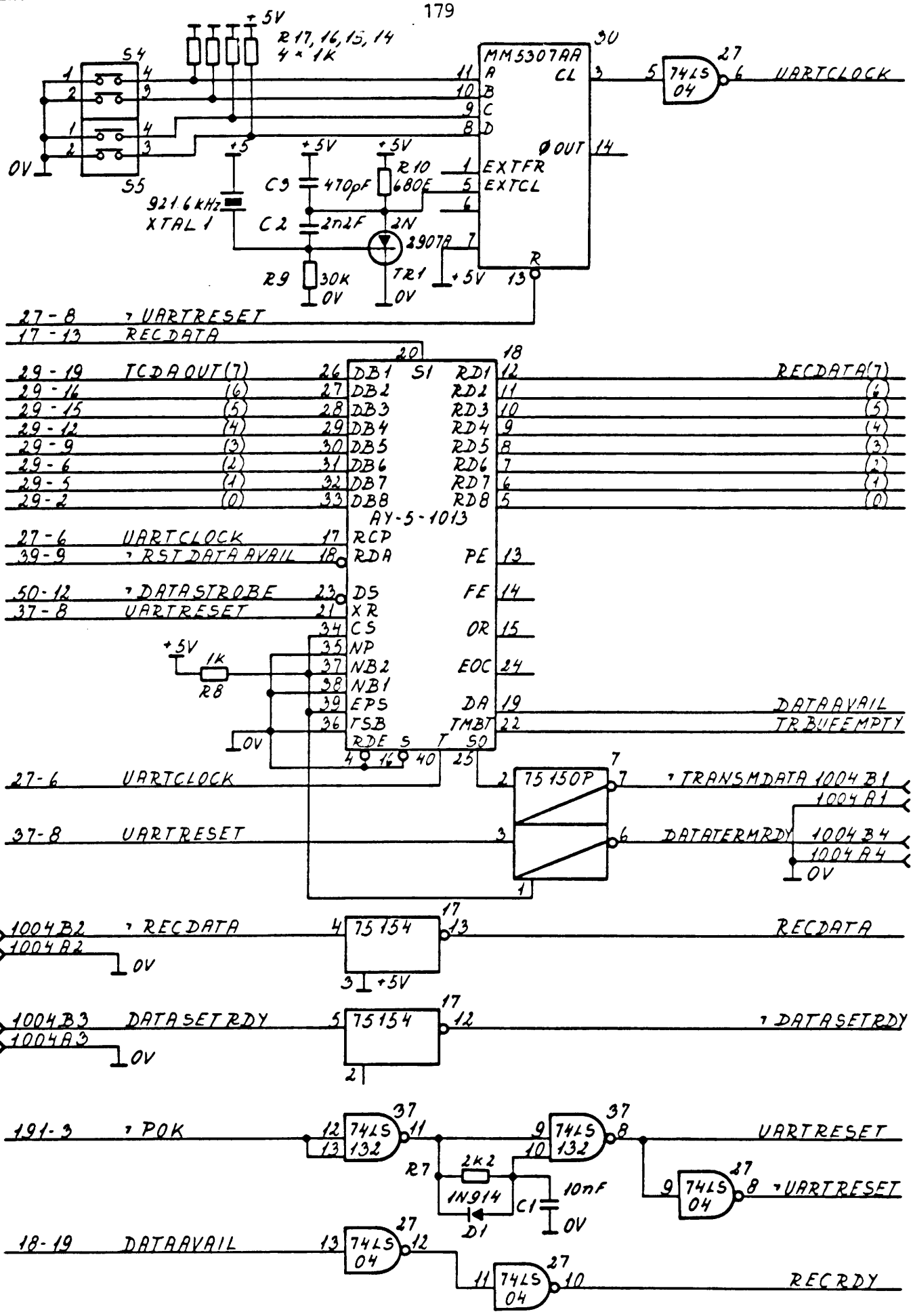
AGA 800901
ARJ 800318



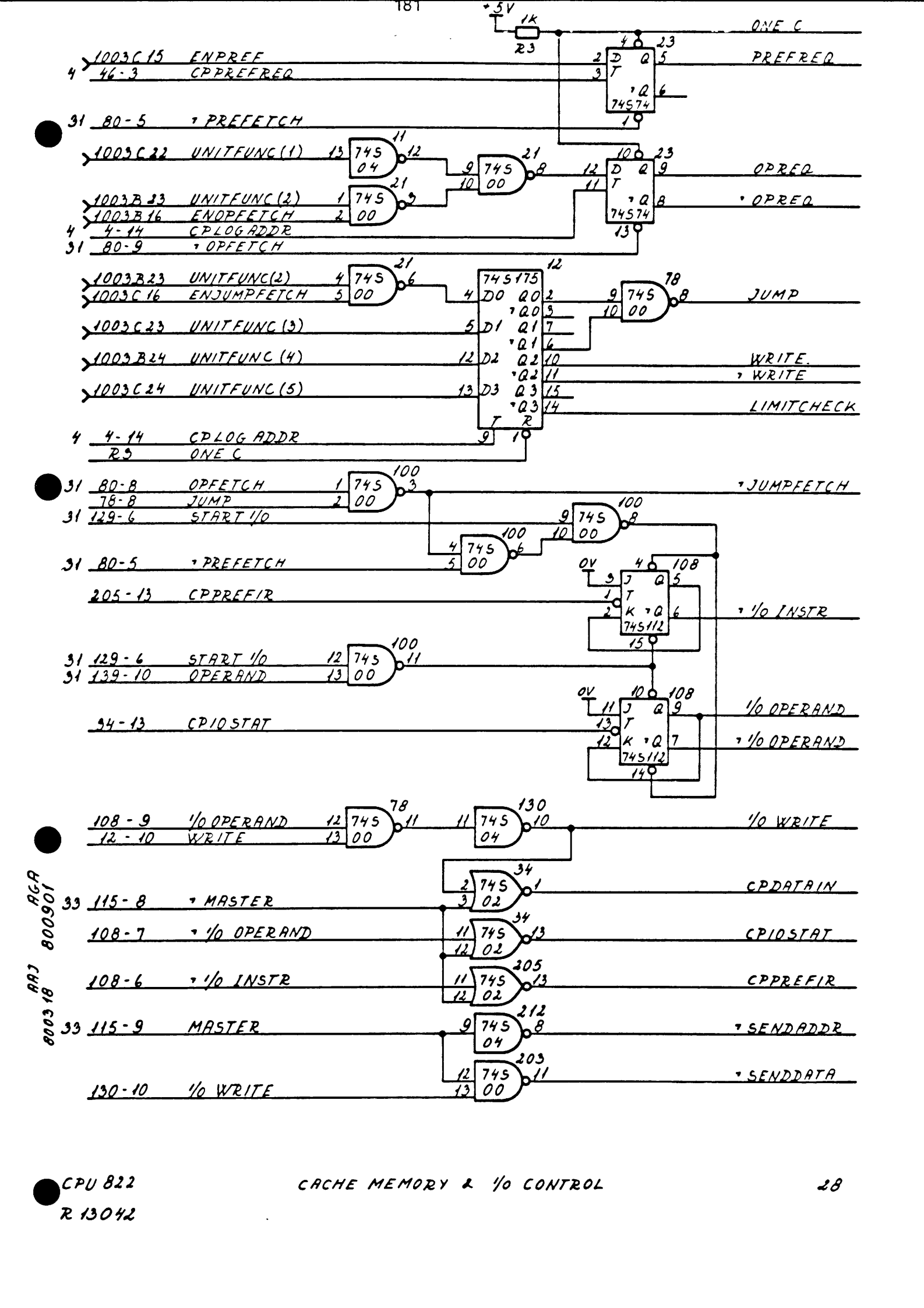

AGA 800901
AAJ 800318

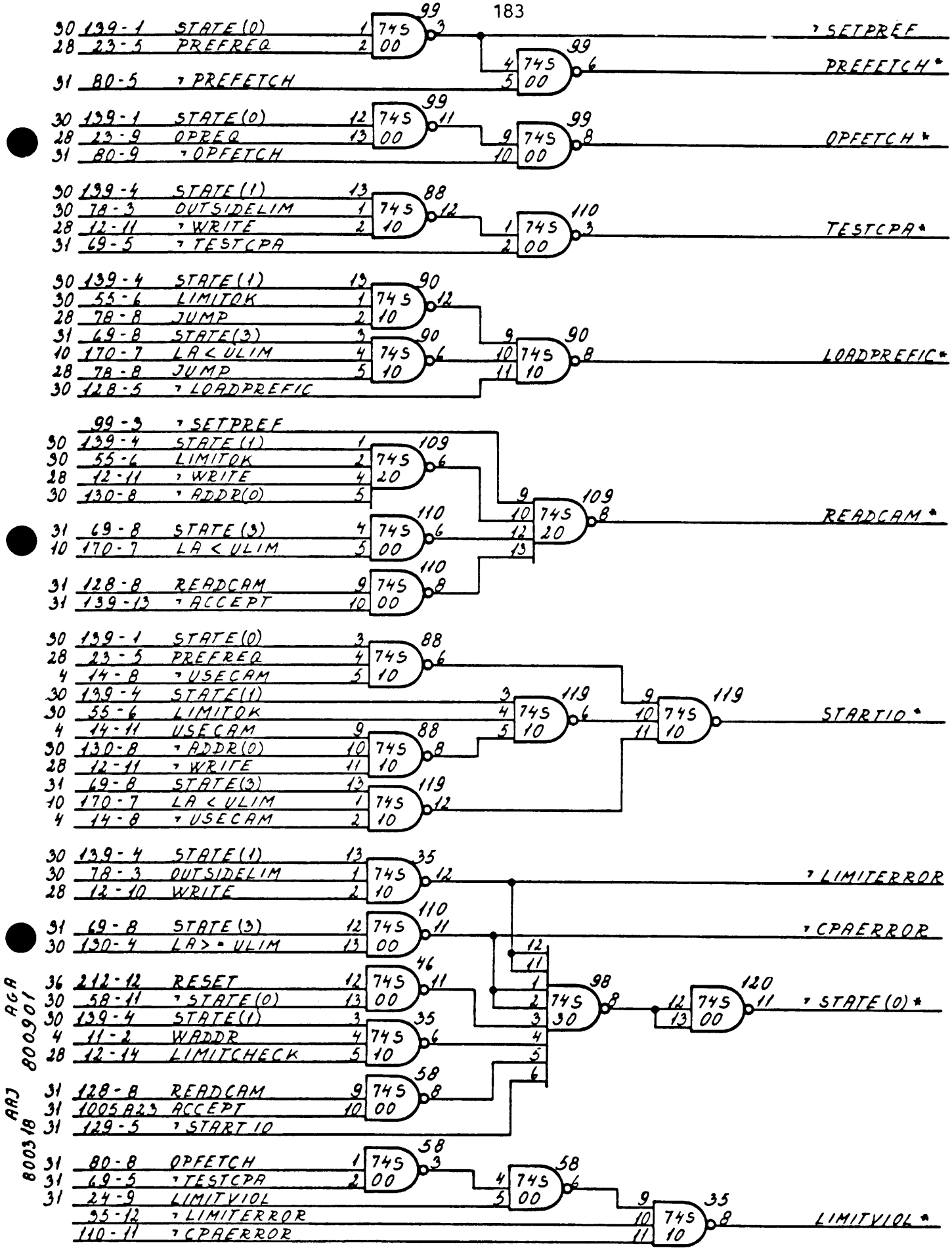


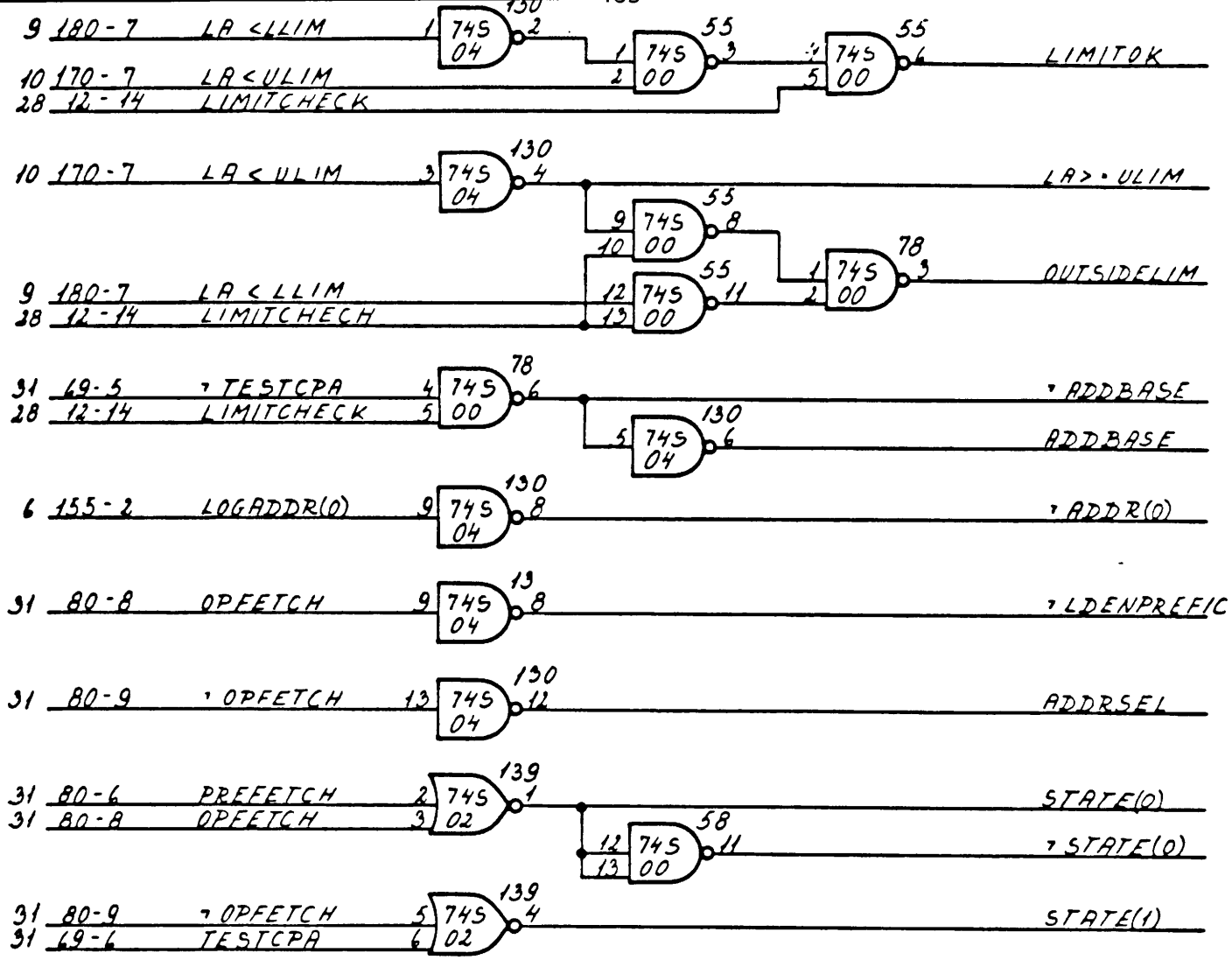

AAJ 800901 AGA 800901




PCA 800901
AAJ 800318

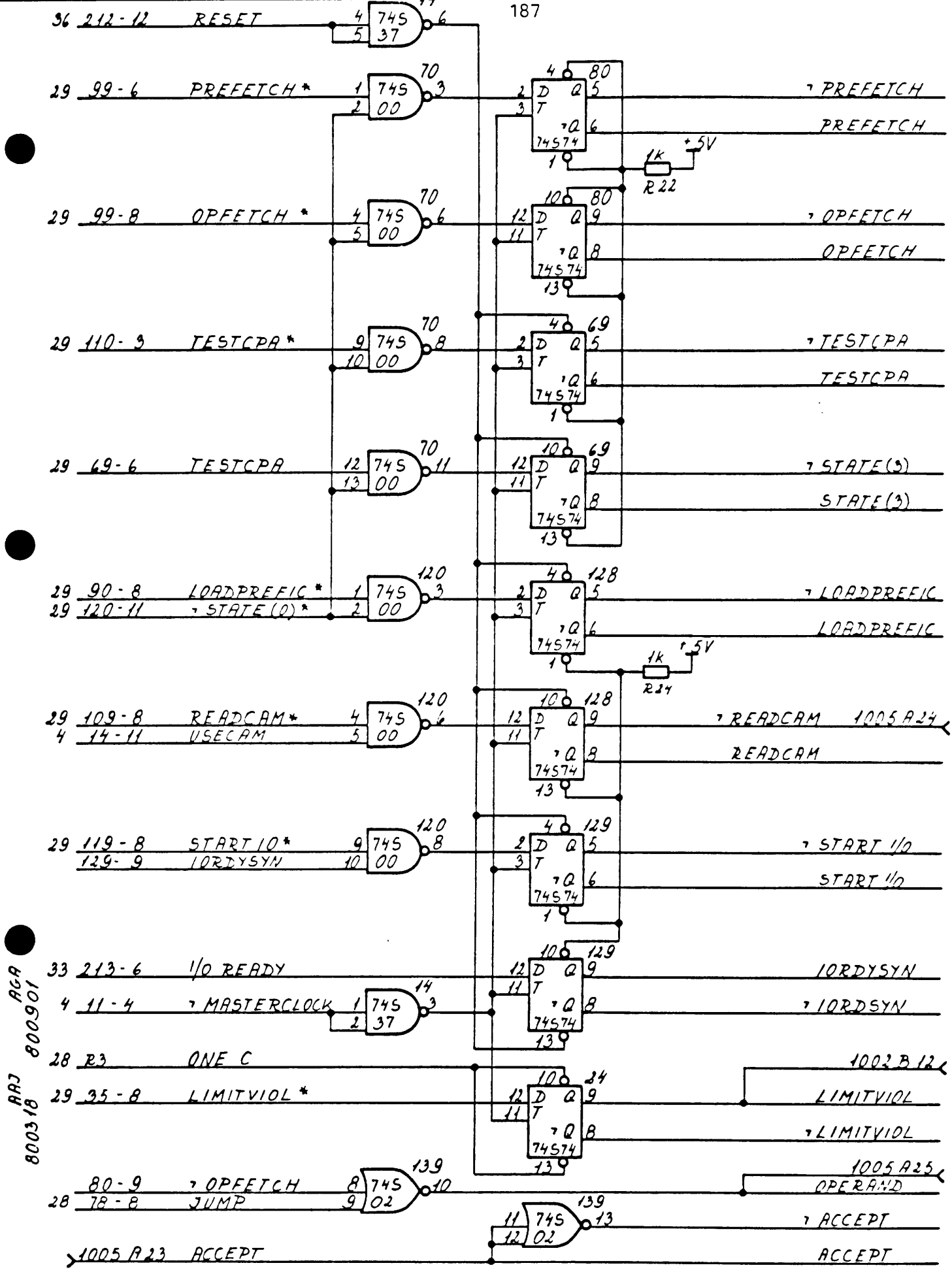

+5V
1K
R3
ONE C
1003C15 ENPREF
4 46-3 CPPREFREQ
PREFREQ
74S74
31 80-5 *PREFETCH
1003C22 UNITFUNC(1)
1003B23 UNITFUNC(2)
1003B16 ENOPFETCH
4 4-14 CPLOGADDR
31 80-9 *OPFETCH
74S04
74S00
OPREQ
*OPREQ
1003B23 UNITFUNC(2)
1003C16 ENJUMPFETCH
1003C23 UNITFUNC(3)
1003B24 UNITFUNC(4)
1003C24 UNITFUNC(5)
74S175
JUMP
WRITE
*WRITE
LIMITCHECK
4 4-14 CPLOG ADDR
R9 ONE C
31 80-8 OPFETCH
78-8 JUMP
*JUMPFETCH
31 129-6 START I/O
31 80-5 *PREFETCH
205-13 CPPREFIR
74S112
*I/O INSTR
31 129-6 START I/O
31 139-10 OPERAND
94-13 CPIOSTAT
I/O OPERAND
*I/O OPERAND
108-9 I/O OPERAND
12-10 WRITE
I/O WRITE
AGA 800901
33 115-8 *MASTER
CPDATAIN
108-7 *I/O OPERAND
CPIOSTAT
AAJ 800318
108-6 *I/O INSTR
CPPREFIR
33 115-9 MASTER
*SENDADDR
130-10 I/O WRITE
*SENDDATA

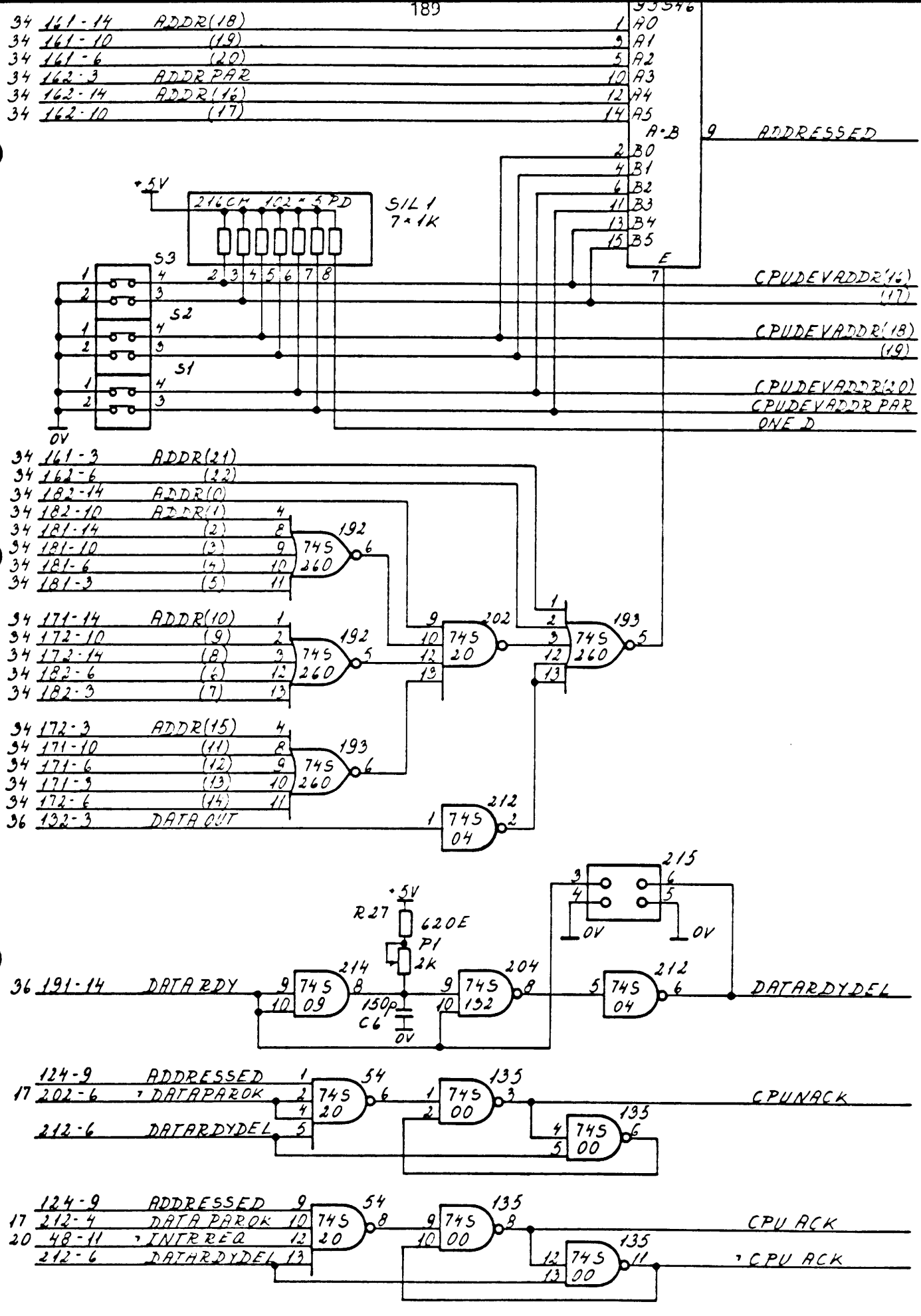

30 139-1 STATE(0)
28 23-5 PREFREQ
99 74S 00
~SETPREF
31 80-5 ~PREFETCH
PREFETCH*
30 139-1 STATE(0)
28 23-9 OPREQ
31 80-9 ~OPFETCH
OPFETCH*
30 139-4 STATE(1)
30 78-3 OUTSIDELIM
28 12-11 ~WRITE
31 69-5 ~TESTCPA
88 74S 10
110 74S 00
TESTCPA*
30 139-4 STATE(1)
30 55-6 LIMITOK
28 78-8 JUMP
31 69-8 STATE(3)
10 170-7 LA < ULIM
28 78-8 JUMP
30 128-5 ~LOADPREFIC
90 74S 10
LOADPREFIC*
99-3 ~SETPREF
30 139-4 STATE(1)
30 55-6 LIMITOK
28 12-11 ~WRITE
30 130-8 ~ADDR(0)
109 74S 20
31 69-8 STATE(3)
10 170-7 LA < ULIM
110 74S 00
31 128-8 READCAM
31 139-13 ~ACCEPT
READCAM*
30 139-1 STATE(0)
28 23-5 PREFREQ
4 14-8 ~USECAM
88 74S 10
30 139-4 STATE(1)
30 55-6 LIMITOK
4 14-11 USECAM
30 130-8 ~ADDR(0)
28 12-11 ~WRITE
31 69-8 STATE(3)
10 170-7 LA < ULIM
4 14-8 ~USECAM
119 74S 10
START10*
30 139-4 STATE(1)
28 12-10 WRITE
35 74S 10
~LIMITERROR
31 69-8 STATE(3)
30 130-4 LA > = ULIM
110 74S 00
~CPAERROR
AGA 800901
36 212-12 RESET
30 58-11 ~STATE(0)
30 139-4 STATE(1)
4 11-2 WADDR
28 12-14 LIMITCHECK
46 74S 00
35 74S 10
98 74S 30
120 74S 00
~STATE(0)*
AAJ 800318
31 128-8 READCAM
31 1005A23 ACCEPT
31 129-5 ~START10
58 74S 00
31 80-8 OPFETCH
31 69-5 ~TESTCPA
31 24-9 LIMITVIOL
95-12 ~LIMITERROR
110-11 ~CPAERROR
35 74S 10
LIMITVIOL*



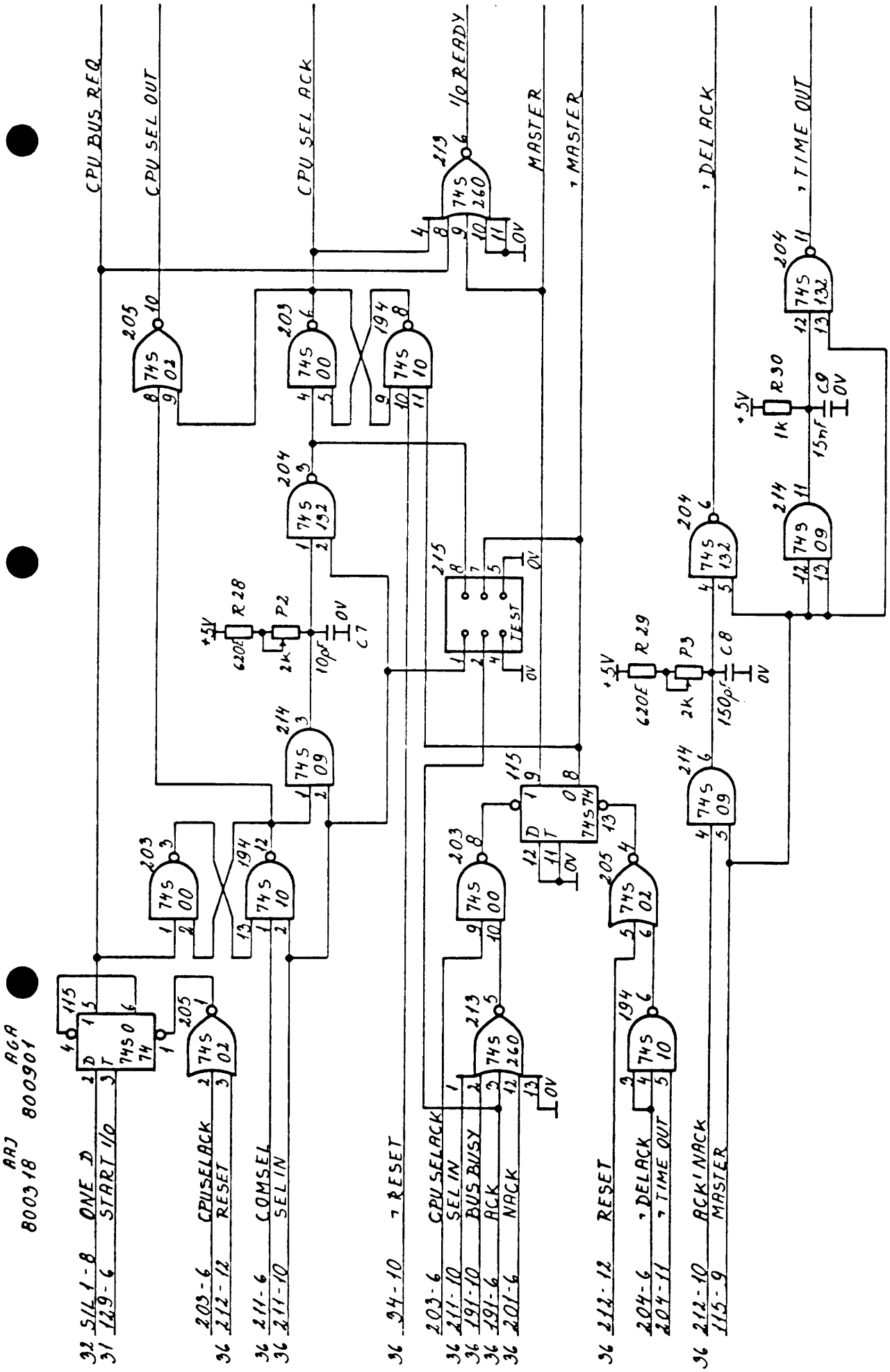

AGA 800901

AAJ 800318


36 212-12 RESET
29 99-6 PREFETCH*
29 99-8 OPFETCH*
29 110-9 TESTCPA*
29 69-6 TESTCPA
29 90-8 LOADPREFIC*
29 120-11 ~STATE(0)*
29 109-8 READCAM*
4 14-11 USECAM
29 119-8 START IO*
129-9 IORDYSYN
33 213-6 I/O READY
4 11-4 ~MASTERCLOCK
28 R3 ONE C
29 35-8 LIMITVIOL*
80-9 ~OPFETCH
28 78-8 JUMP
1005 A23 ACCEPT
74S 37
74S 00
74S 02
74S74
+5V
1K
R22
R24
~PREFETCH
PREFETCH
~OPFETCH
OPFETCH
~TESTCPA
TESTCPA
~STATE(3)
STATE(3)
~LOADPREFIC
LOADPREFIC
~READCAM
READCAM
1005 A24
~START I/O
START I/O
IORDYSYN
~IORDSYN
1002 B12
LIMITVIOL
~LIMITVIOL
1005 A25
OPERAND
~ACCEPT
ACCEPT
AGA 800901
AAJ 800318



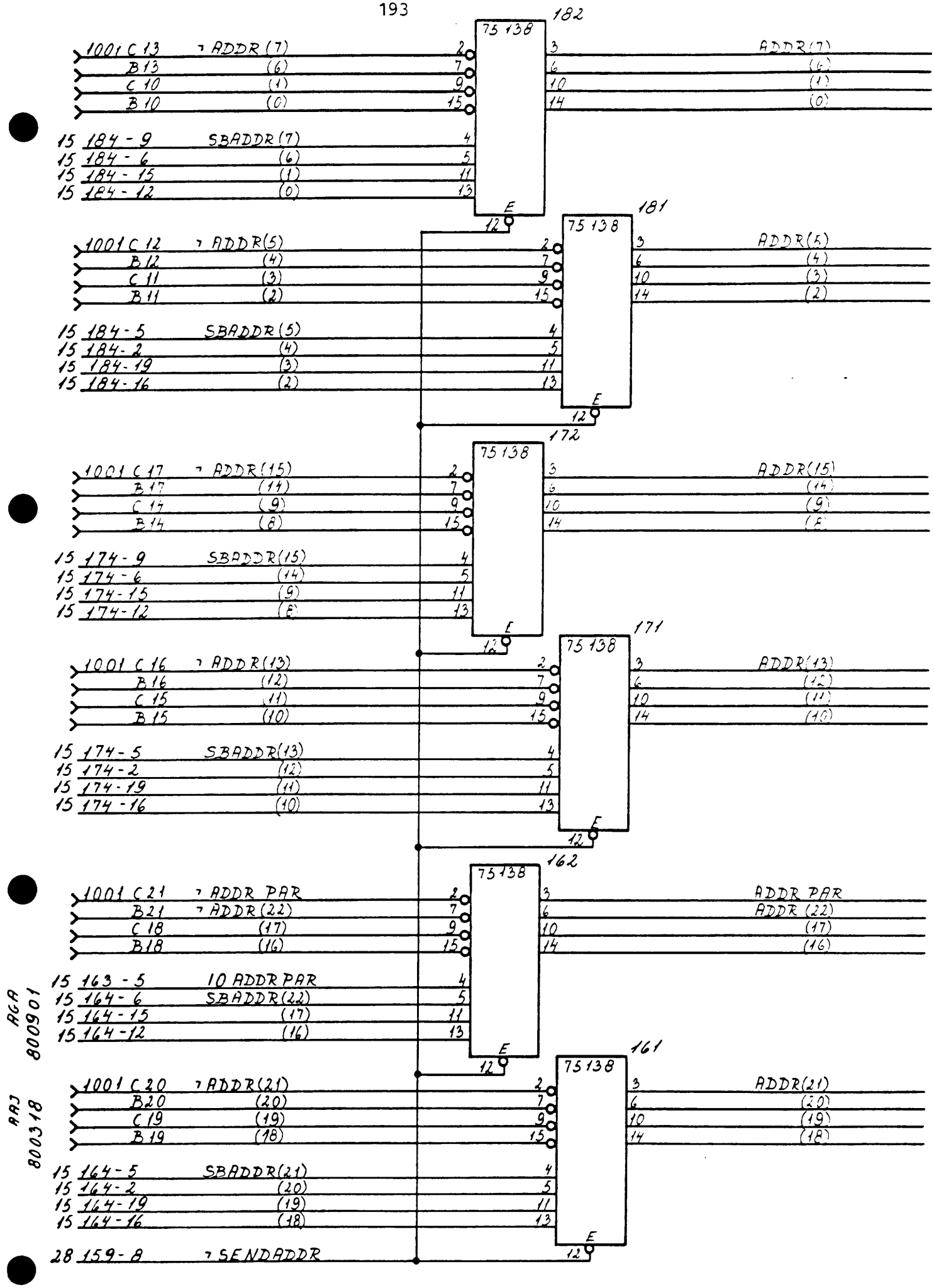

AGA 800901
AAJ 800318

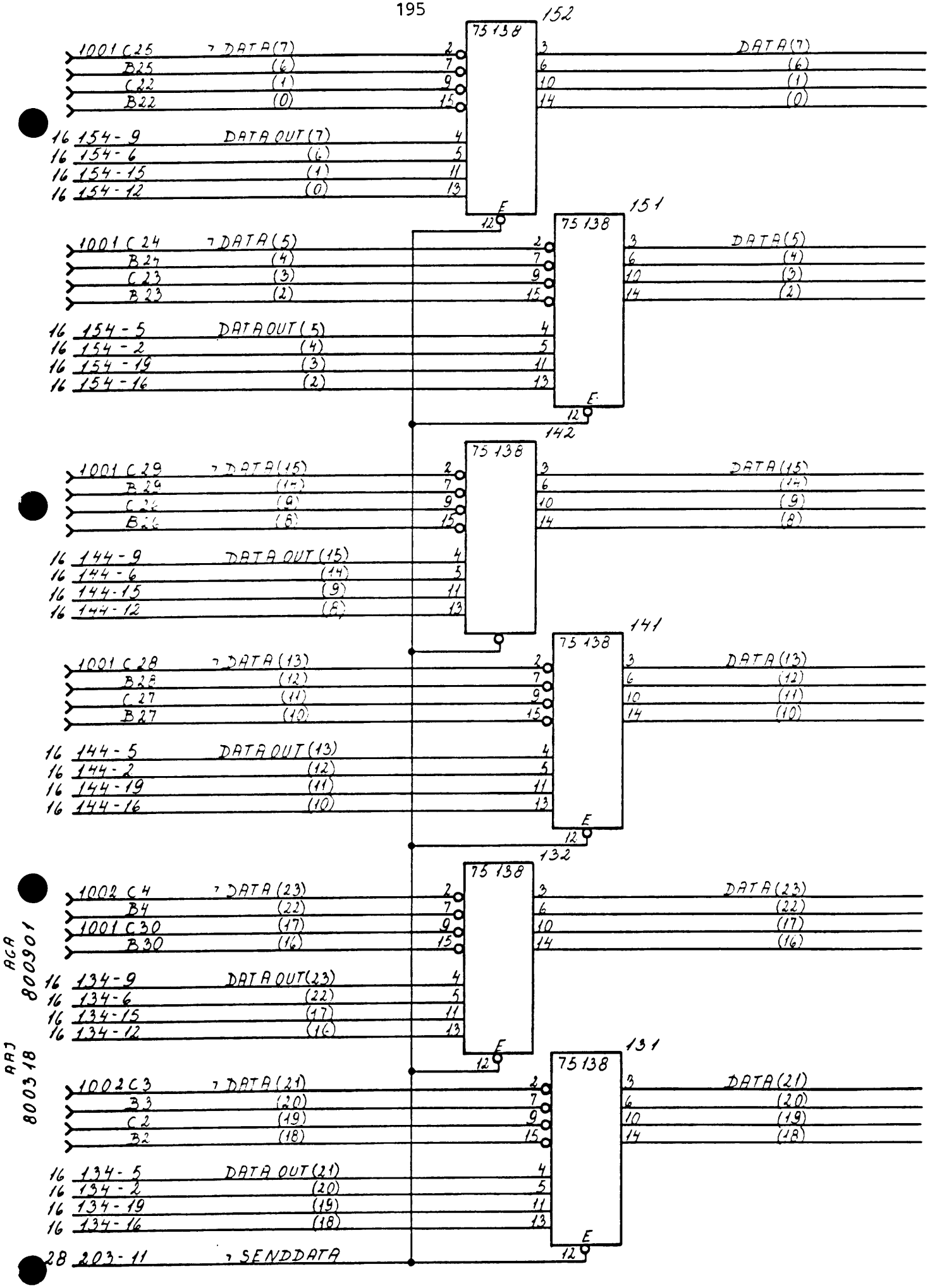


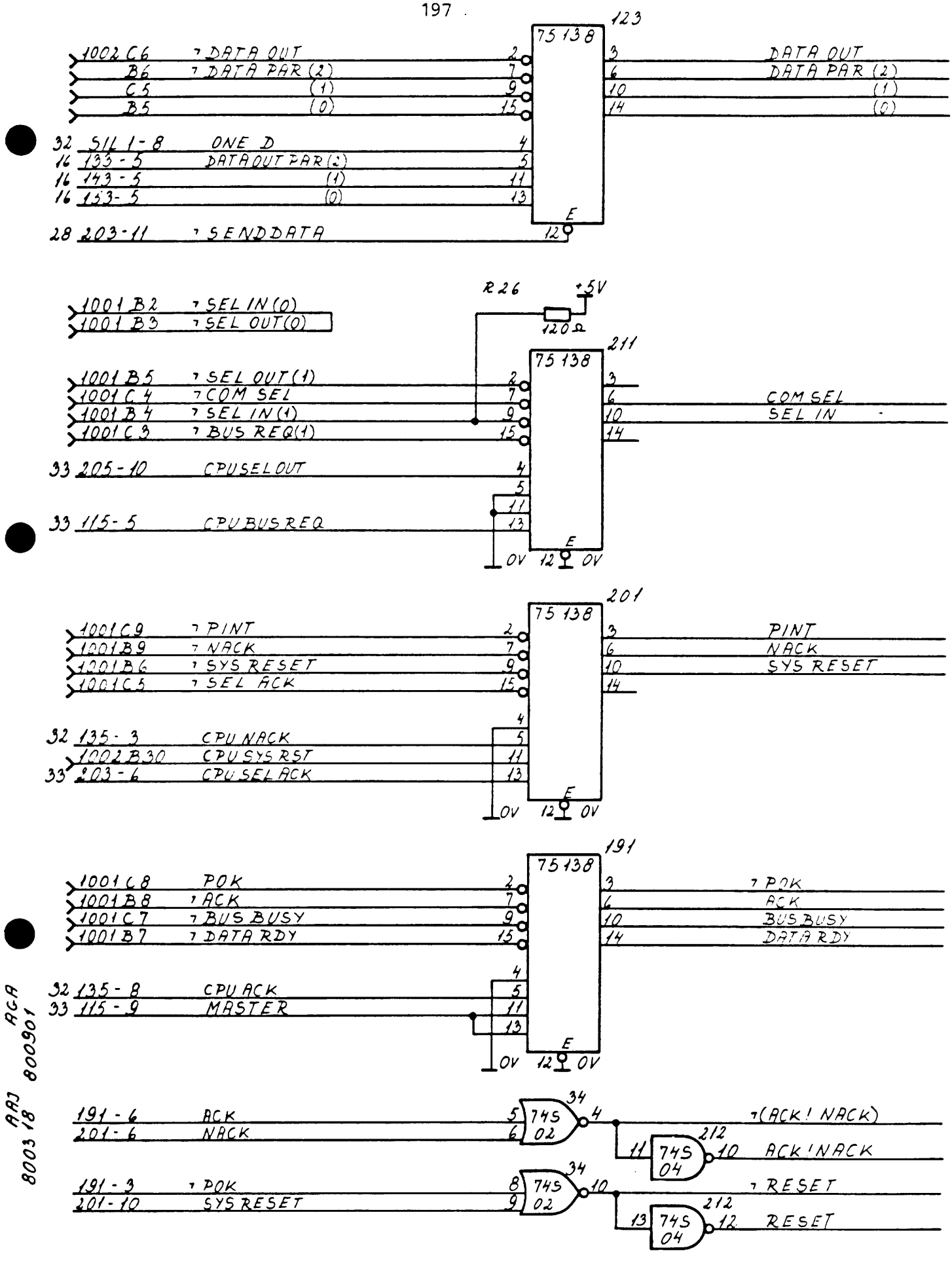

ARJ 800318 RGA 800901

CPU822
R13047

BUS MASTER SELECTION LOGIC


182
75 138
1001 C 13 ADDR (7)
B 13 (6)
C 10 (1)
B 10 (0)
15 184 - 9 SBADDR (7)
15 184 - 6 (6)
15 184 - 15 (1)
15 184 - 12 (0)
ADDR (7)
(6)
(1)
(0)
E
12
181
75 138
1001 C 12 ADDR (5)
B 12 (4)
C 11 (3)
B 11 (2)
15 184 - 5 SBADDR (5)
15 184 - 2 (4)
15 184 - 19 (3)
15 184 - 16 (2)
ADDR (5)
(4)
(3)
(2)
E
12
172
75 138
1001 C 17 ADDR (15)
B 17 (14)
C 17 (9)
B 14 (8)
15 174 - 9 SBADDR (15)
15 174 - 6 (14)
15 174 - 15 (9)
15 174 - 12 (8)
ADDR (15)
(14)
(9)
(8)
E
12
171
75 138
1001 C 16 ADDR (13)
B 16 (12)
C 15 (11)
B 15 (10)
15 174 - 5 SBADDR (13)
15 174 - 2 (12)
15 174 - 19 (11)
15 174 - 16 (10)
ADDR (13)
(12)
(11)
(10)
E
12
162
75 138
1001 C 21 ADDR PAR
B 21 ADDR (22)
C 18 (17)
B 18 (16)
15 163 - 5 10 ADDR PAR
15 164 - 6 SBADDR (22)
15 164 - 15 (17)
15 164 - 12 (16)
ADDR PAR
ADDR (22)
(17)
(16)
E
12
161
75 138
1001 C 20 ADDR (21)
B 20 (20)
C 19 (19)
B 19 (18)
15 164 - 5 SBADDR (21)
15 164 - 2 (20)
15 164 - 19 (19)
15 164 - 16 (18)
ADDR (21)
(20)
(19)
(18)
E
12
28 159 - 8 SENDADDR
RGA 800901
RAJ 800318

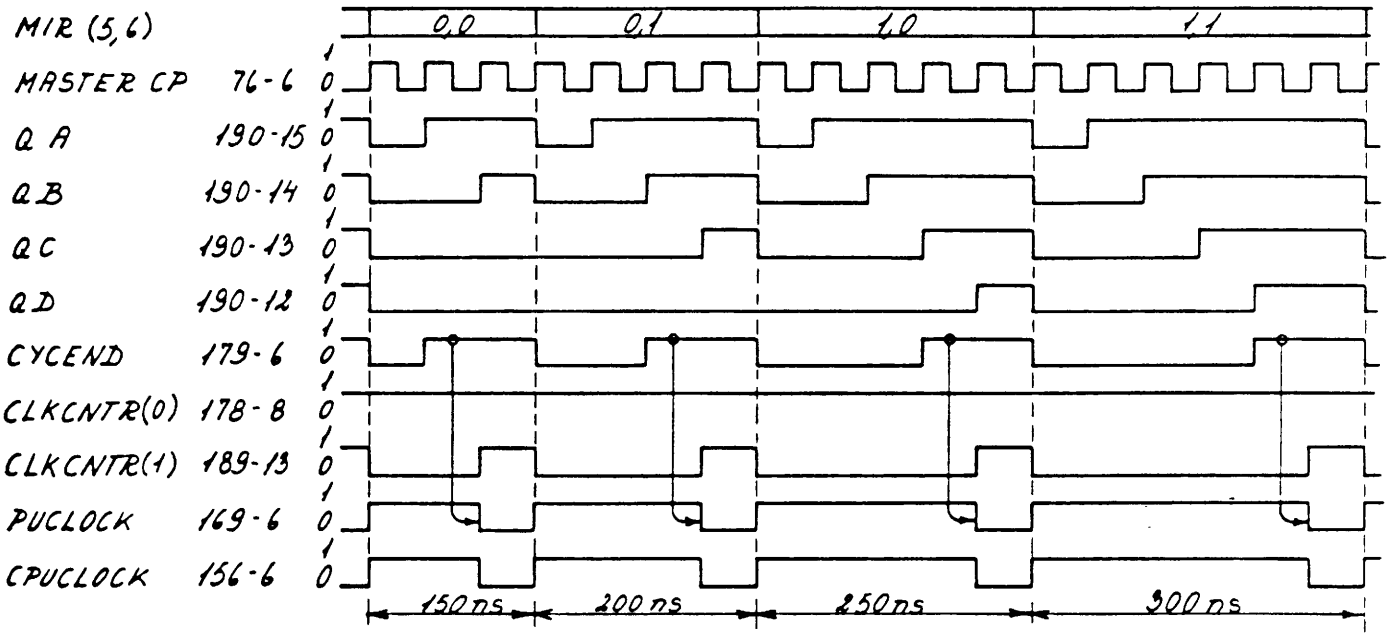

152
75138
1001 C25 DATA(7)
B25 (6)
C22 (1)
B22 (0)
16 154-9 DATA OUT(7)
16 154-6 (6)
16 154-15 (1)
16 154-12 (0)
DATA(7)
(6)
(1)
(0)
151
1001 C24 DATA(5)
B24 (4)
C23 (3)
B23 (2)
16 154-5 DATA OUT(5)
16 154-2 (4)
16 154-19 (3)
16 154-16 (2)
DATA(5)
(4)
(3)
(2)
142
1001 C29 DATA(15)
B29 (14)
C26 (9)
B26 (8)
16 144-9 DATA OUT(15)
16 144-6 (14)
16 144-15 (9)
16 144-12 (8)
DATA(15)
(14)
(9)
(8)
141
1001 C28 DATA(13)
B28 (12)
C27 (11)
B27 (10)
16 144-5 DATA OUT(13)
16 144-2 (12)
16 144-19 (11)
16 144-16 (10)
DATA(13)
(12)
(11)
(10)
132
1002 C4 DATA(23)
B4 (22)
1001 C30 (17)
B30 (16)
16 134-9 DATA OUT(23)
16 134-6 (22)
16 134-15 (17)
16 134-12 (16)
DATA(23)
(22)
(17)
(16)
131
1002 C3 DATA(21)
B3 (20)
C2 (19)
B2 (18)
16 134-5 DATA OUT(21)
16 134-2 (20)
16 134-19 (19)
16 134-16 (18)
DATA(21)
(20)
(19)
(18)
28 203-11 SENDDATA
AGA 800901
AAJ 800318




CPU822 TRANCEIVERS FOR SYSTEMBUS CONTROL SIGNALS 96

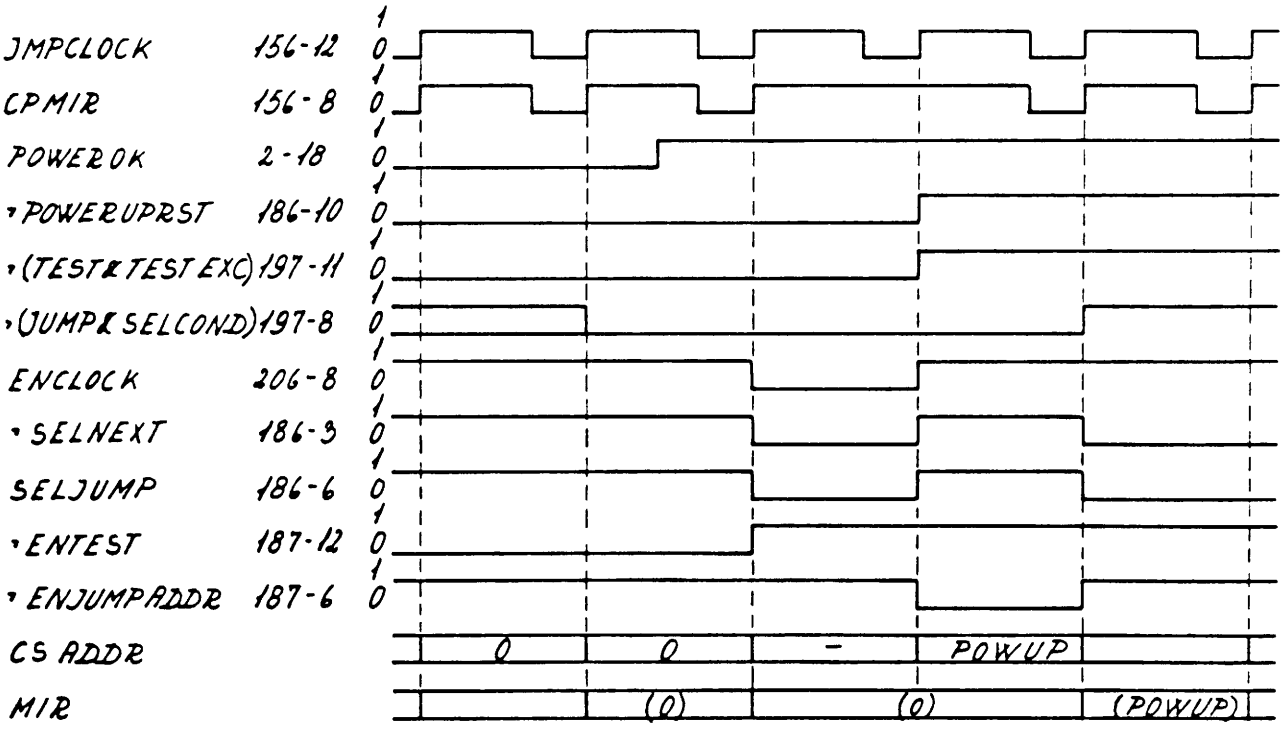
R 13050

## VARIABLE MICROCYCLE




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

## POWER UP



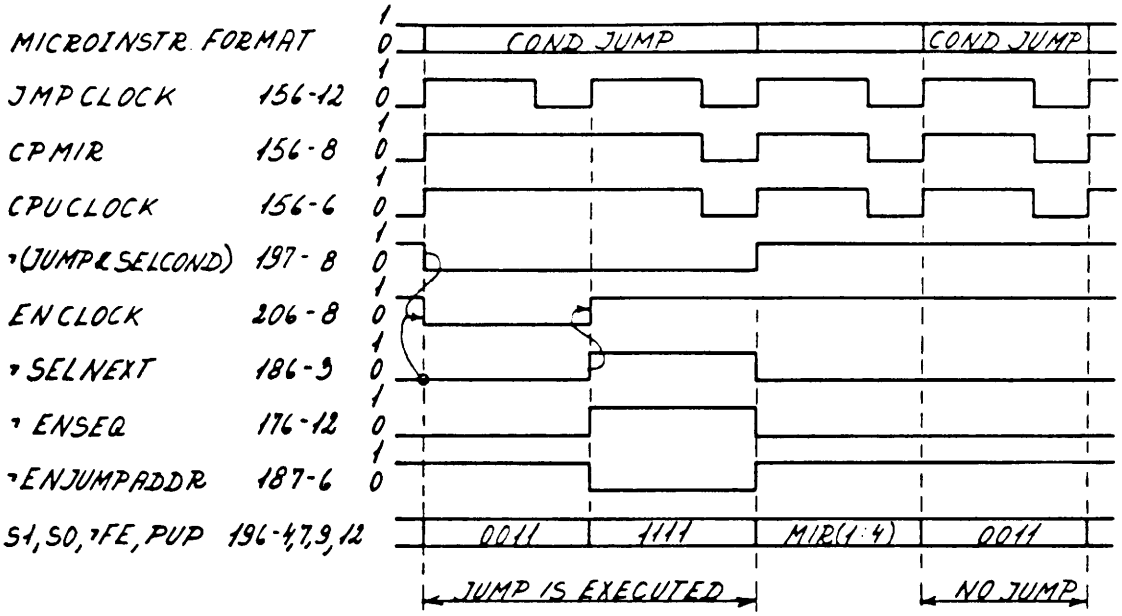

Control store location 0
contains a jump to POWUP.

AAJ 800130 AGA 810630



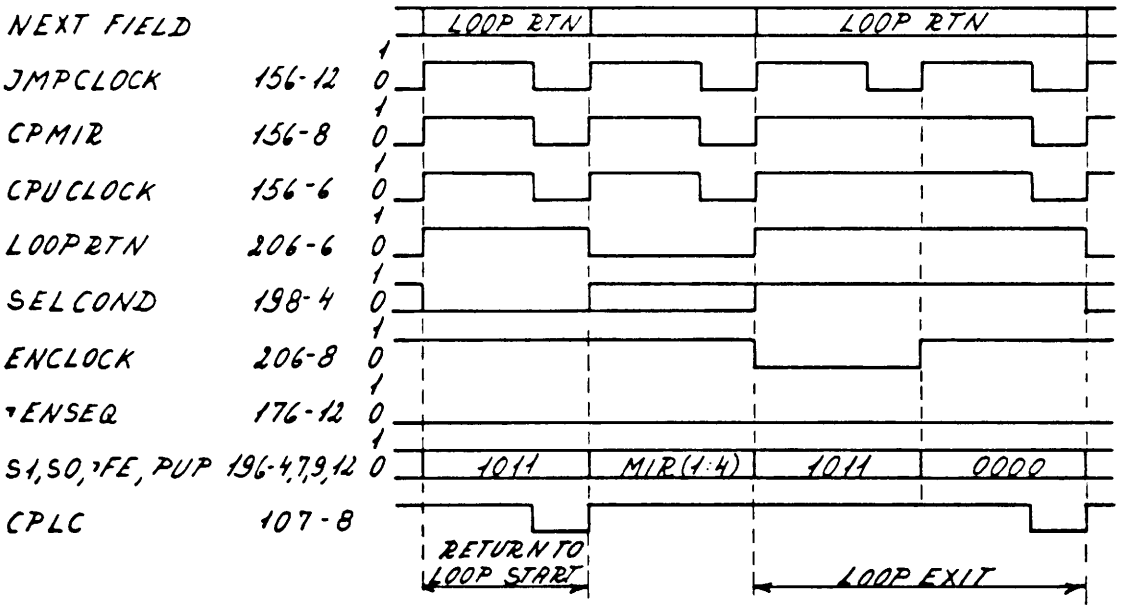
TIMING DIAGRAMS FOR
VARIABLE MICROCYCLE & POWER UP

## CONDITIONAL JUMP




When a jump is executed
the microinstruction time is doubled.

## LOOP RETURN



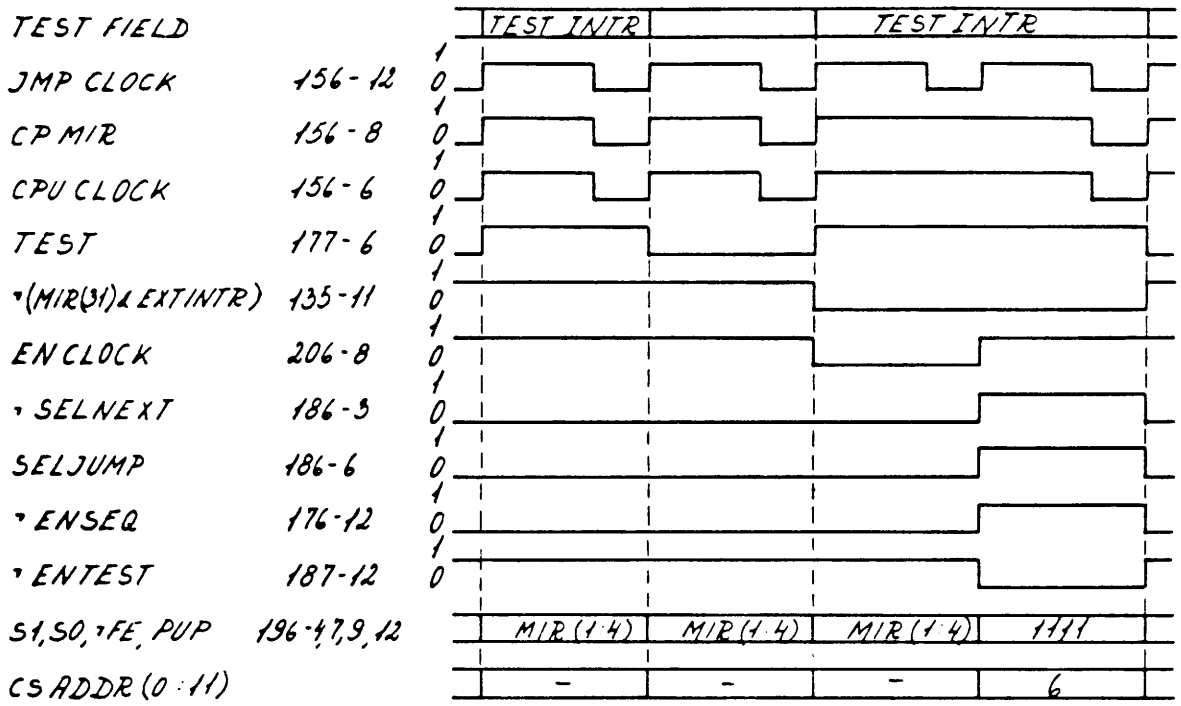

The microinstruction time
is doubled at loop exit.



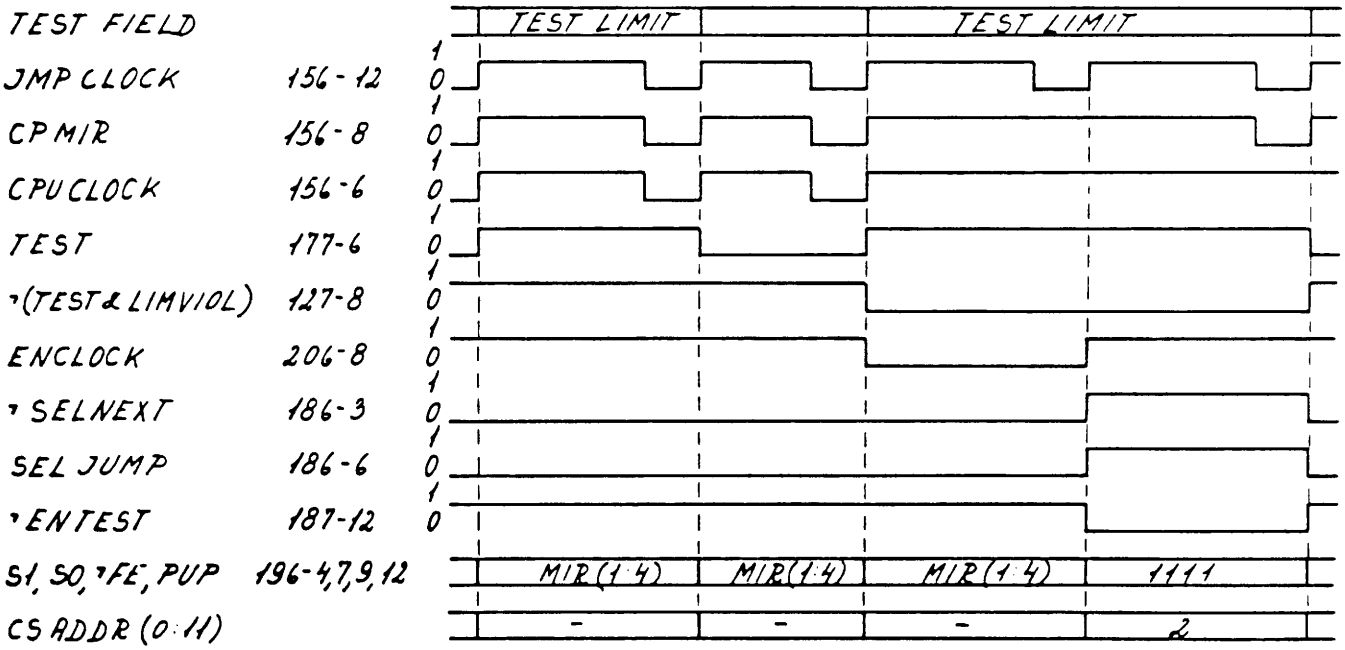


TIMING DIAGRAMS FOR
CONDITIONAL JUMP & CONDITIONAL LOOP RETURN

## TEST EXTERNAL INTERRUPT




## TEST LIMIT VIOLATION



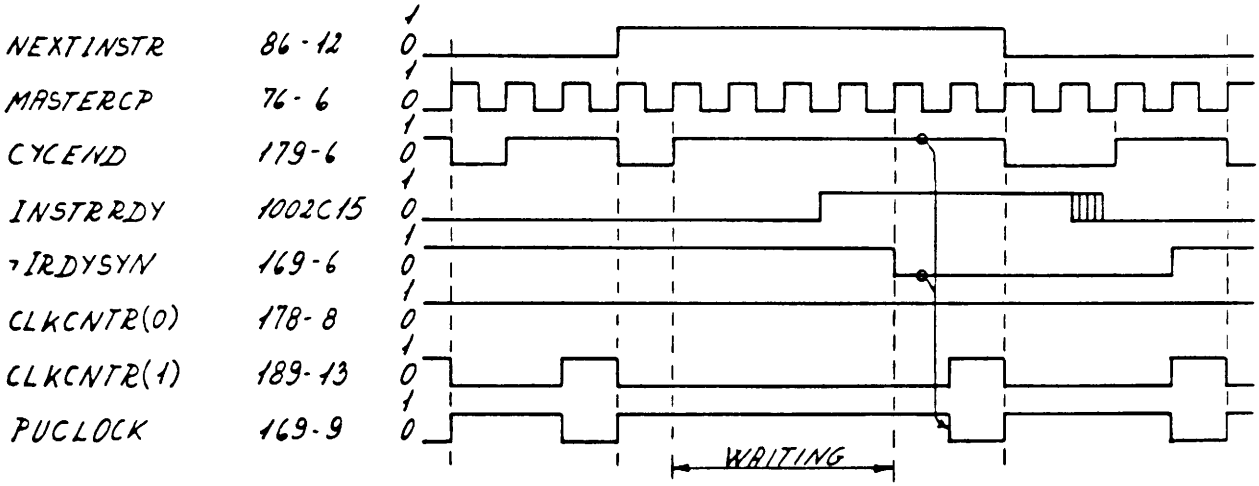

In case of trap on limit violation
CPU CLOCK is disabled.

800130 AHJ 810701 AGA

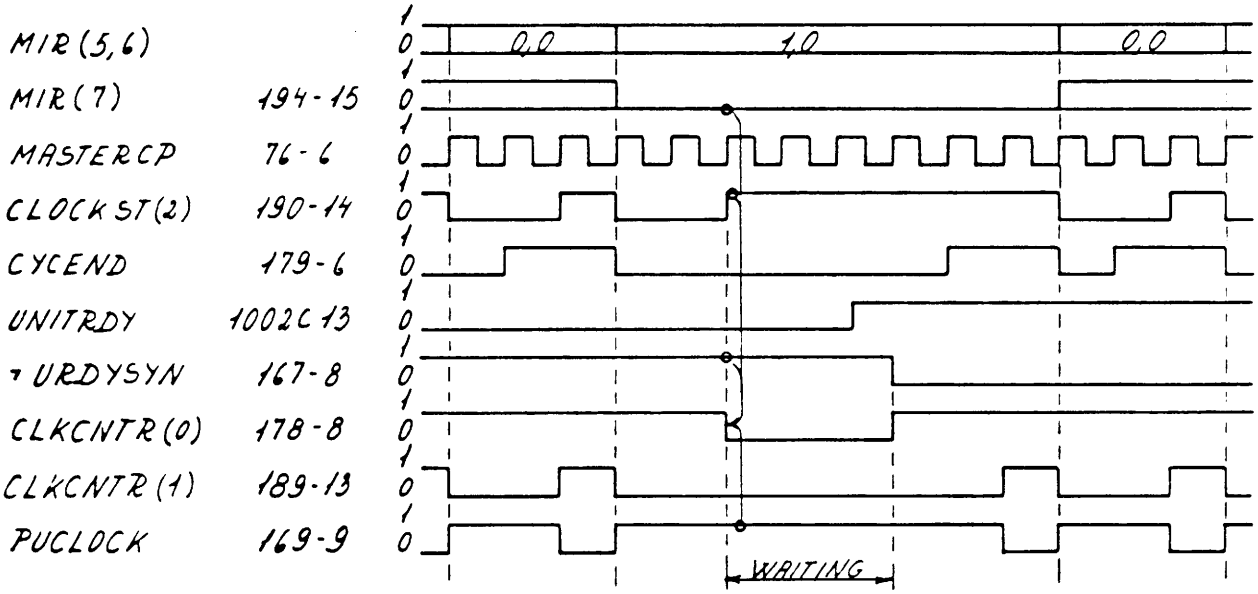


TIMING DIAGRAMS FOR
TEST EXT. INTERRUPT & TEST LIMIT VIOLATION

## INSTRUCTION FETCH SYNCHRONIZATION




## UNIT SYNCHRONIZATION

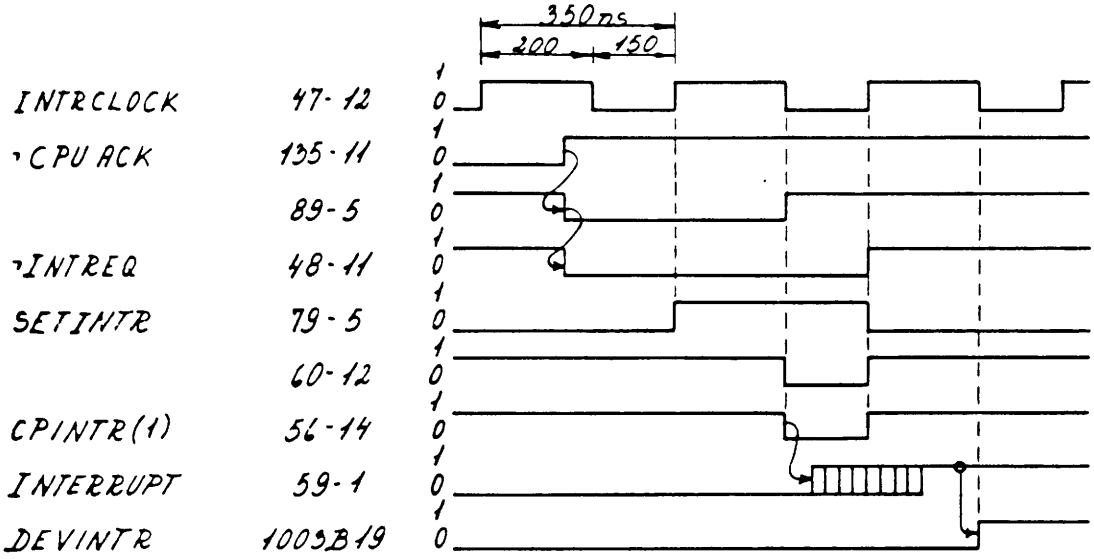



AGA 810702
AAJ 800130

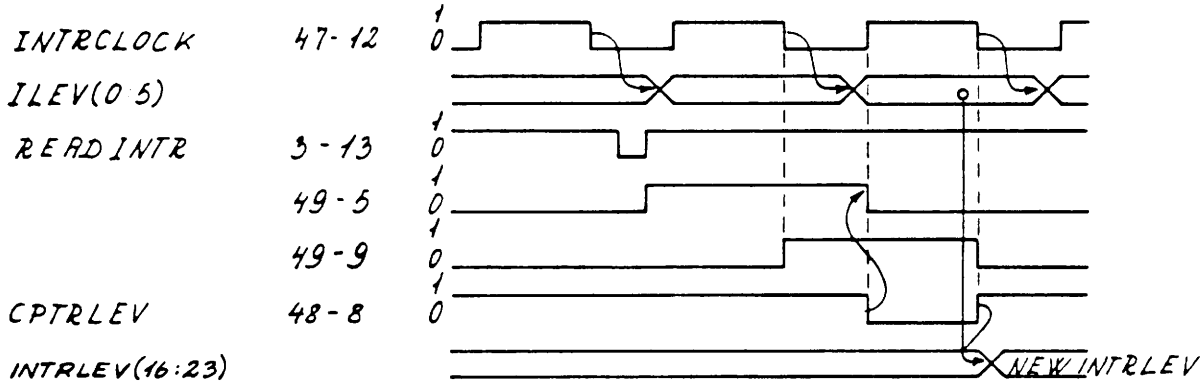
CPU821

TIMING DIAGRAMS FOR
INSTRUCTION FETCH SYNCHRONIZATION &
UNIT SYNCHRONIZATION

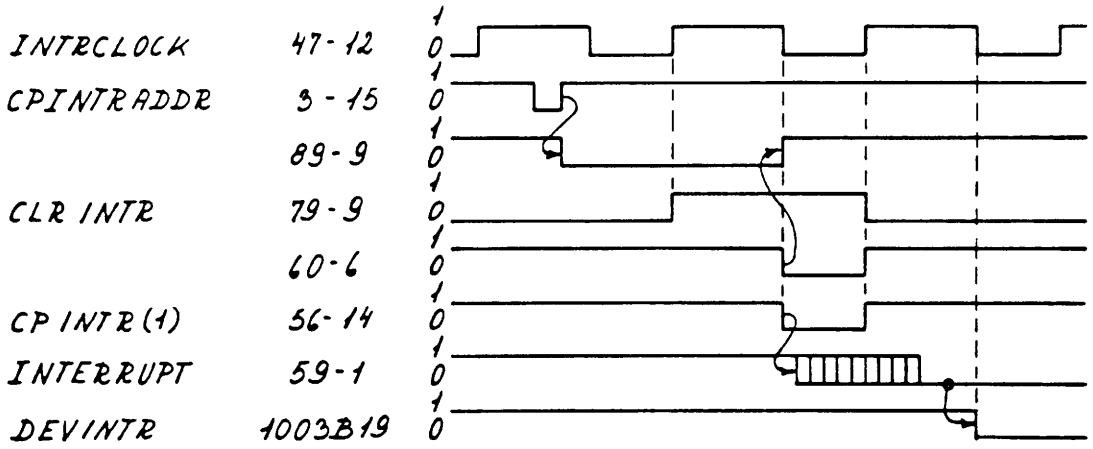
## SET INTERRUPT LEVEL (external interrupt request)




## READ INTERRUPT LEVEL




## CLEAR INTERRUPT LEVEL

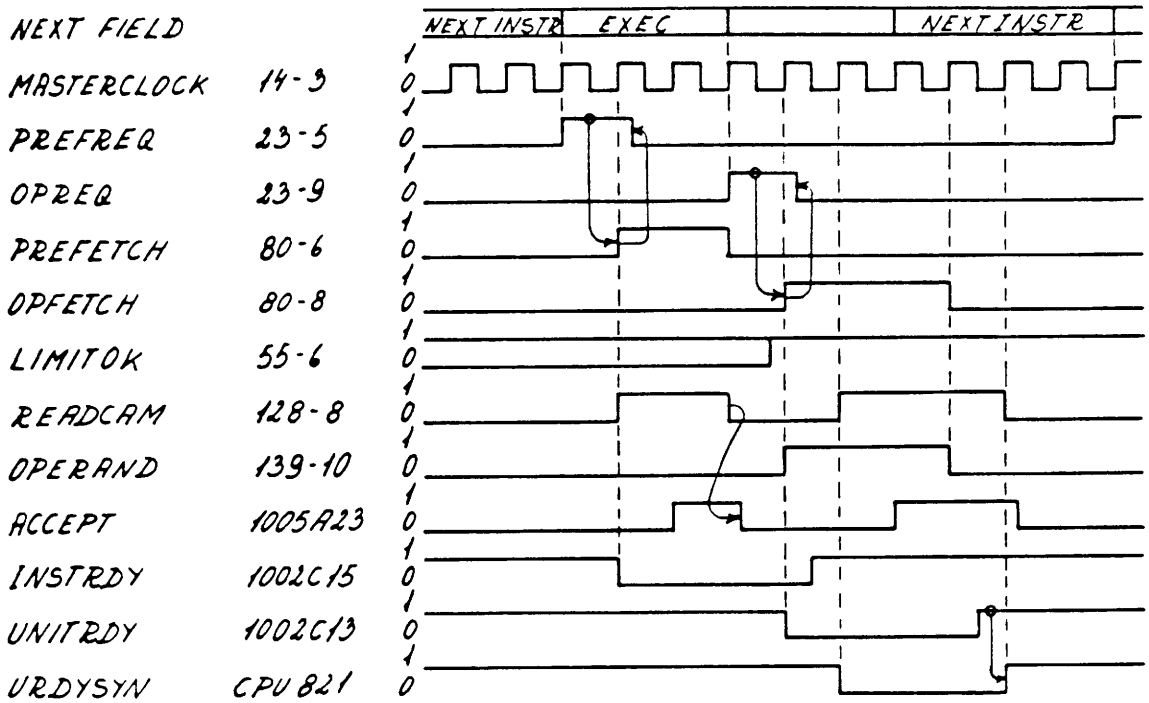



AGA 810703
AAJ 800130

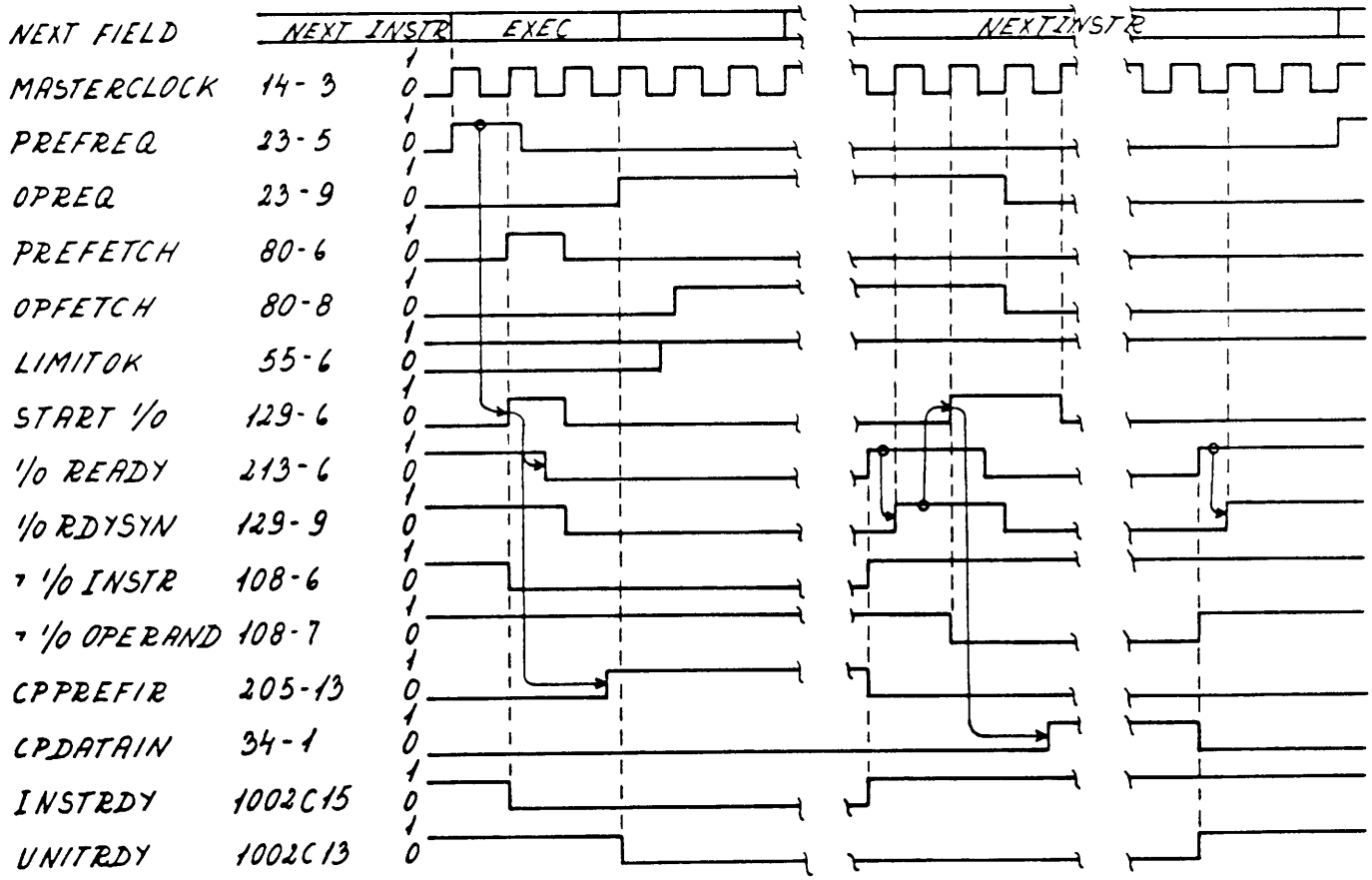
CPU 822

TIMING DIAGRAMS FOR
SET, READ and CLEAR INTERRUPT LEVEL

REGISTER LOAD (with CAM, address within limits)




REGISTER LOAD (without CAM, address within limits)

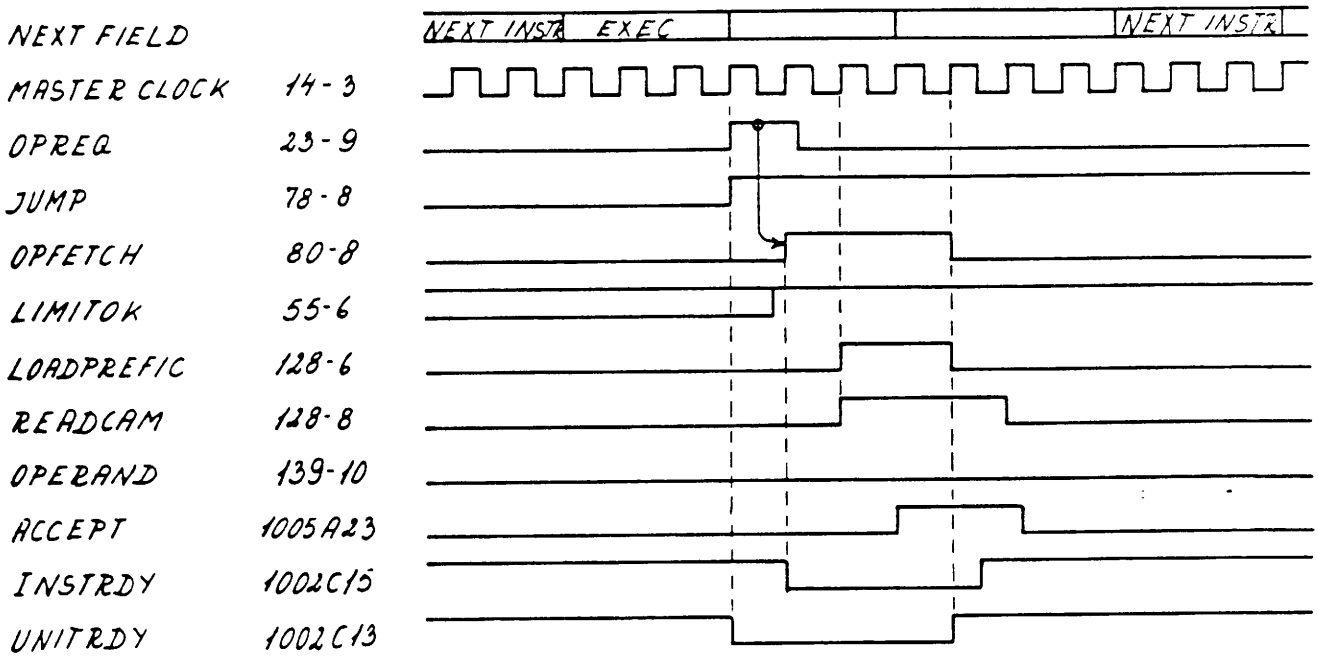



AGA 810703
AAJ 800130



TIMING DIAGRAMS FOR
INSTRUCTION & OPERAND FETCHES

JUMP LINK (with CAM, address within limits)








TIMING DIAGRAM FOR
JUMP LINK INSTRUCTION

MICROPROGRAM PROM POSITION LIST

| PROM NO | POSITION | TYPE |
|---|---|---|
| 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

| OCTAL ADDR | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
|---|---|---|---|---|---|---|---|---|
| 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 1 0 | 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

| OCTAL ADDR | OCTAL CONTENTS |
|---|---|
| 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

| 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 | 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 | -,NEXTINSTR | 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

| PIN | GEN | SIGNAL | PIN | GEN | SIGNAL |
|---|---|---|---|---|---|
| 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 | | -,REMOTEAUTOLOAD | 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

| PIN | GEN | SIGNAL | PIN | GEN | SIGNAL |
|---|---|---|---|---|---|
| A1 | | 0 VOLT | B1 | 7-7 | -,TRANSMDATA |
| A2 | | 0 VOLT | B2 | | -,RECDATA |
| A3 | | 0 VOLT | B3 | | DATASETRDY |
| 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

| PIN | GEN | SIGNAL | PIN | GEN | SIGNAL |
|---|---|---|---|---|---|
| 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

| J1 | | J1005 | SIGNAL |
|---|---|---|---|
| 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 | AUTLOADLAMP+ |

J2

| | | | |
|---|---|---|---|
| A2 | | A6 | -,REMOTEAULOAD |
| A3 | | B6 | 0 VOLT |

## CBL 904 TECHNICIAN CONSOLE CABLE

| J1004 | | J1 | | J2 | SIGNAL |
|---|---|---|---|---|---|
| B1 | | 2 | —— | 6 | -,TRANSMDATA |
| A1 | | | | | |
| B2 | | 3 | | 3 | -,RECDATA |
| A2 | | | | | |
| B3 | | 6 | —— | 5 | DATASETRDY |
| A3 | | | | | |
| B4 | | 20 | —— | 1 | DATATERMRDY |
| A4 | | 7 | —— | 9 | 0 VOLT |

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

421 1288

..... Fold here .....

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

Affix
postage
here

REGNECENTRALEN
af 1979
Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark