RCSL No: 30-M328

Edition: 1982.12.10

Author: Jens K. Andreasen

Title:

CPU721, Technical Manual

RCSL No 42-I 2164

A/S REGNECENTRALEN
af 1979

Keywords:

CPU721, Technical Manual

(238 printed pages)

Abstract:

This paper contains all drawings and functional description to the Central Processor Unit, CPU721.

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL No 42-I 2164

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

TABLE OF CONTENTS (continued) PAGE

1. SHORT DESCRIPTION 1.

The RC3803A CPU is a 16 bit general purpose microprogrammed minicomputer. Basic instruction execution time is less than one microsecond. The CPU provides indexed, base page, relative and multi-level indirect addressing modes and can directly address 32K words of memory. Sixteen levels of interrupt are provided to allow extensive peripherals management.

A block diagram of the CPU is shown in fig. 1. The CPU performs all data handling, logic and arithmetic functions.

The instruction register is 16 bits in length and is loaded from memory at the beginning of each instruction with the instruction to be executed.

The microprogrammed control logic consists of the control PROM's and the address generation circuits. The microprogrammed control logic controls the sequence of execution of each instruction, including fetch of the next instruction. Each instruction is executed by a sequence of microprogrammed steps. Each step or microprogram state is defined by the contents of the PROM output register, and exists for exactly one clock period (150 ns). In each microprogram state the control PROM's set up conditions for the ALU, registers and branch PROM's.

The sequence of the microprogram states is determined by the next address control. The next address controls decode the contents of the instruction register and generate a series of PROM addresses necessary to execute the instruction in question.

Other indicators, such as carry-out and zero detect, are also monitored by the next address control to modify the address sequence as required for proper instruction execution.

The ALU, register control and status logic, picoinstruction control PROM's and sixteen registers are provided by four AM 2901 four bit microcontroller LSI integrated circuits.

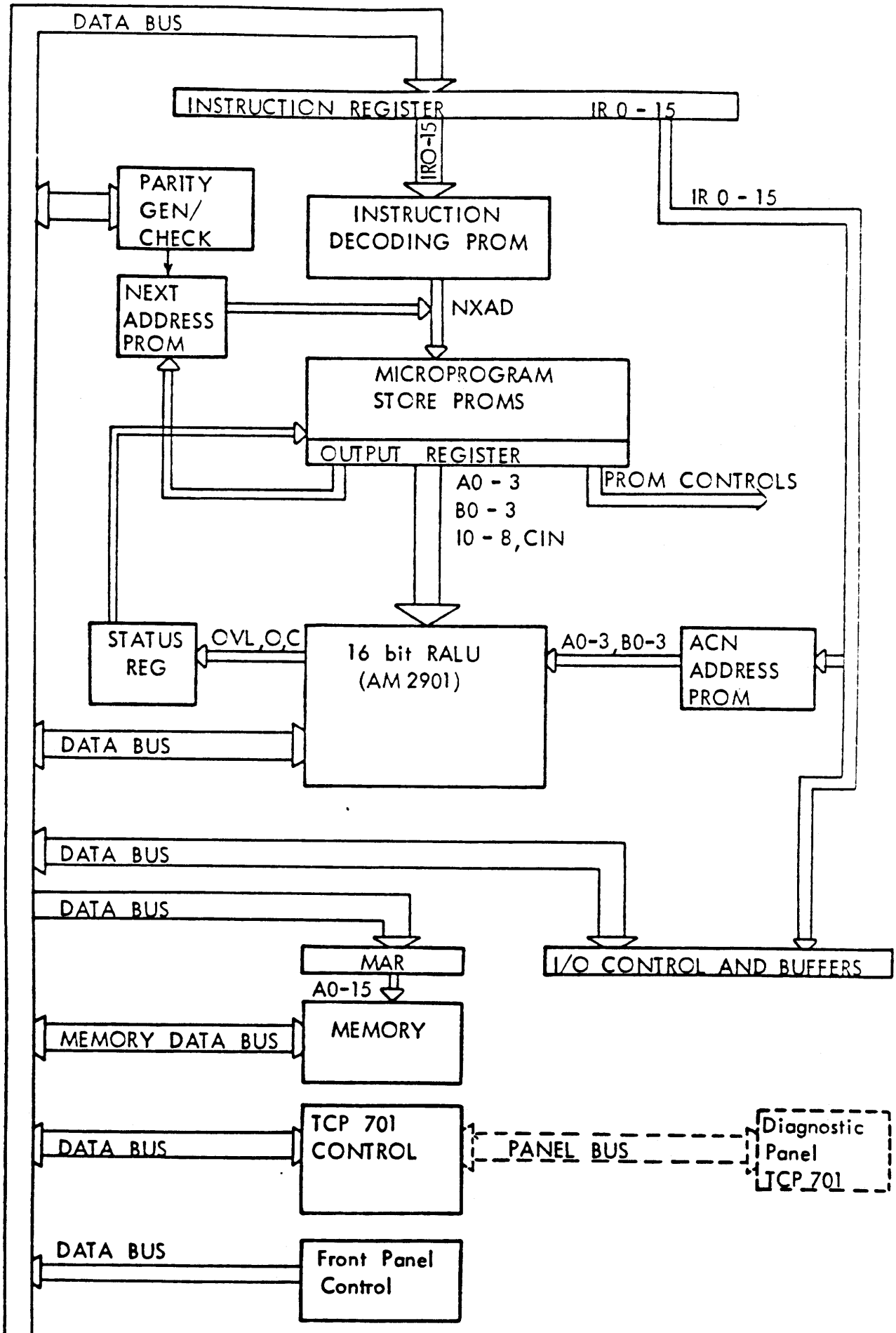


Figure 1: Block Diagram.

Associated with the AM 2901's are the displacement address generator and carry/shift controls. The displacement address generator converts the 8 bit displacement contained in memory reference instructions into a signed 16 bit number by extending the sign bit of the displacement. This is required for proper memory address generation in memory reference instructions. The carry and shift controls provide the unique decoding, carry storage and skip decode.

The I/O controls consist of the six bit device address buffer register and the instruction register decode PROM's.

2. DATA PATHS 2.

The block diagram, fig. 1, shows the CPU721 data flow. As can be seen from this illustration, the CPU is organized around a CPJ Data Bus, -,BUS 0-15.

The contents of this Data Bus can be loaded into Instruction Register, RALU, Memory Address Register, Data input to Memory, I/O Data Bus, Data Register to TCP701.

The Data Bus can be supplied with information from RALU, Data from Memory, I/O Data Bus, Data Bus from TCP701, the contents of the Autoload address Counter and Autoload Program Memory.

3. REGISTER ALU UNIT 3.

Architechture.

A detailed block diagram of the bipolar microprogrammable microprocessor structure is shown in fig. 2. The circuit is a four-bit slice cascadable to any number of bits. Therefore, all data paths within the circuit are four bits wide. The two key elements in the fig. 2 block diagram are the 16-word by 4-bit 2-port RAM and the high-speed ALU.

Data in any of the 16 words of the Random Access Memory (RAM) can be read from the A-port of the RAM as controlled by the 4-bit A address field input. Likewise, data in any of the 16 words of the RAM as defined by the B address field input can be simultaneously read from the B-port of the RAM. The same code can be applied to the A select field and B select field in which case the identical file data will appear at both the RAM A-port and B-port outputs simultaneously.

When enabled by the RAM write enable (RAM EN), new data is always written into the file (word) defined by the B address field of the RAM. The RAM data input field is driven by a 3-input multiplexer. This configuration is used to shift the ALU output data (F) if desired. This three-input multiplexer scheme allows the data to be shifted up (right) one bit position, shifted down (left) one bit position or not shifted in either direction.

The RAM A-port data outputs and RAM B-port data outputs drive separate 4-bit latches. These latches hold the RAM data while the clock input is LOW. This eliminates any possible race conditions that could occur while new data is being written into the RAM.

The high-speed Arithmetic Logic Unit (ALU) can perform three binary arithmetic and five logic operations on the two 4-bit input words R and S.

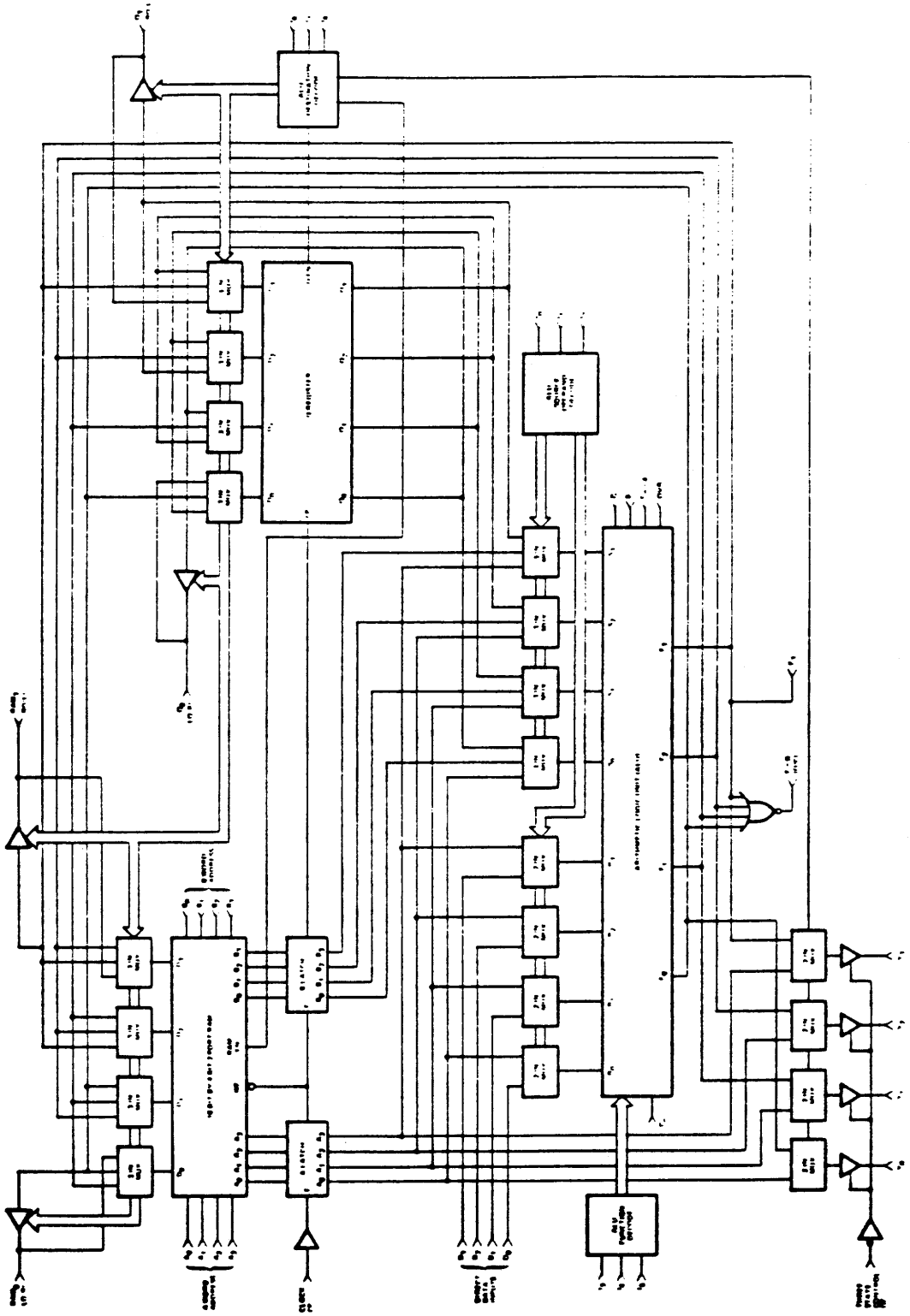


Figure 2: Detailed Am2901 Microprocessor Block Diagram.

Mnemonic	MICRO CODE I_2	I_1	I_0	Octal Code	ALU SOURCE OPERANDS R	S
AQ	L	L	L	0	A	Q
AB	L	L	H	1	A	B
ZQ	L	H	L	2	0	Q
ZB	L	H	H	3	0	B
ZA	H	L	L	4	0	A
DA	H	L	H	5	D	A
DQ	H	H	L	6	D	Q
DZ	H	H	H	7	D	0

Figure 3: ALU Source Operand Control.

Mnemonic	MICRO CODE I_5	I_4	I_3	Octal Code	ALU Function	SYMBOL
ADD	L	L	L	0	R Plus S	R + S
SUBR	L	L	H	1	S Minus R	S - R
SUBS	L	H	L	2	R Minus S	R - S
OR	L	H	H	3	R OR S	R ∨ S
AND	H	L	L	4	R AND S	R ∧ S
NOTRS	H	L	H	5	$\overline{R}$ AND S	$\overline{R}$ ∧ S
EXOR	H	H	L	6	R EX-OR S	R ⊻ S
EXNOR	H	H	H	7	R EX-NOR S	$\overline{R \veebar S}$

Figure 4: ALU Function Control.

Mnemonic	MICRO CODE I_8	I_7	I_6	Octal Code	RAM FUNCTION Shift	Load	Q-REG FUNCTION Shift	Load	Y OUTPUT	RAM SHIFTER RAM_0	RAM_3	Q SHIFTER Q_0	Q_3
QREG	L	L	L	0	X	NONE	NONE	F → Q	F	X	X	X	X
NOP	L	L	H	1	X	NONE	X	NONE	F	X	X	X	X
RAMA	L	H	L	2	NONE	F → B	X	NONE	A	X	X	X	X
RAMF	L	H	H	3	NONE	F → B	X	NONE	F	X	X	X	X
RAMQD	H	L	L	4	DOWN	F/2 → B	DOWN	Q/2 → Q	F	F_0	IN_3	Q_0	IN_3
RAMD	H	L	H	5	DOWN	F/2 → B	X	NONE	F	F_0	IN_3	Q_0	X
RAMQU	H	H	L	6	UP	2F → B	UP	2Q → Q	F	IN_0	F_3	IN_0	Q_3
RAMU	H	H	H	7	UP	2F → B	X	NONE	F	IN_0	F_3	X	Q_3

Figure 5: ALU Destination Control.

OCTAL I_5 4 3	I_{210} OCTAL / ALU Function / ALU Source	0 A, Q	1 A, B	2 0, Q	3 0, B	4 0, A	5 D, A	6 D, Q	7 D, 0
0	C_n = L R Plus S	A+Q	A+B	Q	B	A	D+A	D+Q	D
	C_n = H	A+Q+1	A+B+1	Q+1	B+1	A+1	D+A+1	D+Q+1	D+1
1	C_n = L S Minus R	Q−A−1	B−A−1	Q−1	B−1	A−1	A−D−1	Q−D−1	−D−1
	C_n = H	Q−A	B−A	Q	B	A	A−D	Q−D	−D
2	C_n = L R Minus S	A−Q−1	A−B−1	−Q−1	−B−1	−A−1	D−A−1	D−Q−1	D−1
	C_n = H	A−Q	A−B	−Q	−B	−A	D−A	D−Q	D
3	R OR S	A∨Q	A∨B	Q	B	A	D∨A	D∨Q	D
4	R AND S	A∧Q	A∧B	0	0	0	D∧A	D∧Q	0
5	$\overline{R}$ AND S	$\overline{A}$∧Q	$\overline{A}$∧B	Q	B	A	$\overline{D}$∧A	$\overline{D}$∧Q	0
6	R EX-OR S	A⊻Q	A⊻B	Q	B	A	D⊻A	D⊻Q	D
7	R EX-NOR S	$\overline{A \veebar Q}$	$\overline{A \veebar B}$	$\overline{Q}$	$\overline{B}$	$\overline{A}$	$\overline{D \veebar A}$	$\overline{D \veebar Q}$	$\overline{D}$

Figure 6: Source Operand and ALU Function Matrix.

The R input field is driven from a 2-input multiplexer, while the S input is driven from a 3-input multiplexer. Both multiplexers also have an inhibit capability; that is, no data is passed. This is equivalent to a "zero" source operand.

Referring to fig. 2, the ALU R-input multiplexer has the RAM A-port and the direct data inputs (d) connected as inputs. Likewise, the ALU S-input multiplexer has the RAM A-port, the RAM B-port and the Q register connected as inputs.

This multiplexer scheme gives the capability of selecting various pairs of the A, B, D, Q and "0" inputs as source operands to the ALU. These five inputs, when taken two at a time, result in ten possible combinations of source operand pairs. These combinations include AB, AD, AQ, A0, BD, BQ, B0, DQ, D0 and Q0. It is apparent that AD, AQ and A0 are somewhat redundant with BD, BQ and B0 in that if the A address and B address are the same, the identical function results. Thus, there are only seven completely non-redundant source operand pairs for the ALU. The Am2901 microprocessor implements eight of these pairs. The microinstruction inputs used to select the ALU source operands are the I0, I1 and I2 inputs. The definition of I0, I1 and I2 for the eight source operand combinations are shown in fig. 3. Also shown is the octal code for each selection.

The two source operands not fully described as yet are the D input and Q input. The D input is the four-bit wide direct data field input. This port is used to insert all data into the working registers inside the device. Likewise, this input can be used in the ALU to modify any of the internal data files. The Q register is a separate 4-bit file intended primarily for multiplication and division routines but it can also be used as an accumulator or holding register for some applications.

The ALU itself is a high-speed arithmetic/logic operator capable of performing three binary arithmetic and five logic functions. The I3, I4 and I5 microinstruction inputs are used to select the ALU function. The definition of these inputs is shown in fig. 4. The octal code is also shown for reference. The normal technique

for cascading the ALU of several devices is in a look-ahead carry mode. Carry generate, G, and carry propagate, P, are outputs of the device for use with a carry-look-ahead-generator such as the Am2902 ('182). A carry-out Cn + 4, is also generated and is available as an output for use as the carry flag in a status register. Both carry-in (Cn) and carry-out (Cn + 4) are active HIGH.

The ALU has three other status-oriented outputs. These are F3, F = 0, and overflow (OVR). The F3 output is the most significant (sign) bit of the ALU and can be used to determine positive or negative results without enabling the three-state data outputs. F3 is non-inverted with respect to the sign bit output Y3. The F = 0 output is used for zero detect. It is an open-collector output and can be wire OR'ed between microprocessor slices. F = 0 is HIGH when all F outputs are LOW. The overflow output (OVR) is used to flag arithmetic operations that exceed the available two's complement number range. The overflow output (OVR) is HIGH when overflow exists. That is, when Cn + 3 and Cn + 4 are not the same polarity.

The ALU data output is routed to several destinations. It can be a data output of the device and it can also be stored in the RAM or the Q register. Eight possible combinations of ALU destination functions are available as defined by the I6, I7 and I8 microinstruction inputs. These combinations are shown in fig. 5.

The four-bit data output field (Y) features three-state outputs and can be directly bus organized. An output control (OE) is used to enable the three-state outputs. When OE is HIGH, the Y outputs are in the high-impedance state.

A two-input multiplexer is also used at the data output such that either the A-port of the RAM or the ALU outputs (F) are selected at the device Y outputs. This selection is controlled by the I6, I7 and I8 microinstruction inputs. Refer to fig. 5 for the selected output for each microinstruction code combination.

As was discussed previously, the RAM inputs are driven from a three-input multiplexer. This allows the ALU outputs to be entered non-shifted, shifted up one position or shifted down one position. The shifter has two ports; one is labelled RAM0-LO/RI and the other is labelled RAM3-RO/LI. Both of these ports consist of a buffer-driver with a three-state output and an input to the multiplexer. Thus, in the shift up mode, the RO buffer is enabled and the RI multiplexer input is enabled. Likewise, in the shift down mode, the LO buffer and LI input are enabled. In the no-shift mode, both the LO and RO buffers are in the high-impedance state and the multiplexer inputs are not selected. This shifter is controlled from the I6, I7 and I8 microinstruction inputs as defined in fig. 5.

Similarly, the Q register is driven from a 3-input multiplexer. In the no-shift mode, the multiplexer enters the ALU data into the Q register. In either the shift-up or shift-down mode, the multiplexer selects the Q register data appropriately shifted up or down. The Q shifter also has two ports; one is labelled Q0-LO/RI and the other is Q3-RO/LI. The operation of these two ports is similar to the RAM shifter and is also controlled from I6, I7 and I8 as shown in fig. 5.

The clock input to the Am2901 controls the RAM, the Q register and the A and B data latches. When enabled, data is clocked into the Q register on the LOW-to-HIGH transition of the clock. When the clock input is HIGH, the A and B latches are open and will pass whatever data is present at the RAM outputs. When the clock input is LOW, the latches are closed and will retain the last entered. If the RAM-EN is enabled, new data will be written into the RAM file (word) defined by the B address field when the clock input is LOW.

4. MICROPROGRAM FLOW DESCRIPTION 4.

For details about the microprogram flow, refer to flow diagram FC1-21.

4.1 Power-Up 4.1

Refer to FC1.

When the machine is first powered up, the address of the first microinstruction is RAR 210. Zero is loaded into the Memory Address Register (MAR) and program execution is started by fetching the first instruction from memory address 0.

4.2 Instruction Fetch 4.2

Refer to FC1.

Instruction execution begins with the Fetch cycle (RAR 1). The instruction is fetched from memory and loaded into the instruction register. After the instruction has been fetched, a multiway branch is made based on decode of the instruction. Bit 0-7 of the instruction is used as an address to a table given directly the branch address (refer to subsection 6.2.1).

4.3 Instruction Execution 4.3

The last microstep in each instruction sequence increments the Program Counter and simultaneously loads the memory address register with the incremented contents of the program counter in preparation for fetch of the next instruction. This occurs for all instructions except those which load the program counter with a new value (jump).

After the program counter has been incremented, a multi-way branch is performed and is controlled by the presence of SKIP conditions, DMA requests, INTerrupt requests and the state of the RUN flip-flop.

Highest priority is given to the SKIP condition, since the SKIP is part of the prior instruction. DMA requests receive second priority. DMA requests are serviced only when the CPU is in RUN mode.

After SKIP and DMA requests have been serviced, and a Front Panel request is present, a branch is made to the Front Panel service portion of the microprogram. In like manner, an interrupt request causes control to be passed to the interrupt service section of the microcode, after execution of which the same branch decision is repeated on further interrupt or DMA activity. If there is no further interrupt or DMA activity, control returns to the instruction fetch block.

Interrupt requests are not serviced when the CPU is in STOP mode.

If Break mode is selected (BREAK switch on the Diagnostic Panel), the program counter is incremented in the last step in each instruction, but the multi-way branch mentioned above is not performed. A direct jump to the BREAK mode part of the microcode is performed. After performing the BREAK sequence (FC10) the multi-way branch mentioned above is performed if no Break Condition exists.

4.3.1 Memory Reference Instruction 4.3.1

Refer to FC3.

In all memory reference instructions step one brings the instruction word from memory and simultaneously loads it into the instruction register and the Q register of the AM2901.

A displacement address generator is used to convert the 8-bit displacement contained in the memory reference instruction into a signed 16-bit number. This is done by extending the sign bit of the displacement. The sign of the 8-bit displacement (a 0 in bit 8 is a plus and a 1 is a minus) is extended into bits 7 through 0 to make a signed 8 bit number into a proper signed 16 bit number.

Instruction execution time for an LDA/STA is four microcycles consisting of Op Fetch (RAR 1); Q + PC (RAR 7), AC2 and AC3 (RAR 10-11) ---> MAR; ACN ---> MEM (RAR 33) or MEM ---> ACN (RAR 32); and PC + 1 ---> PC, MAR (RAR 217).

For direct or indirect memory addressing the address calculation routine has four entry points corresponding to base page, PC relative, AC2 relative and AC3 relative.

These four routines generate an effective address by using the contents of Q as is or by adding Q to PC or AC2 or AC3.

The resulting address is loaded into MAR.

If additional calculation such as further indirect or auto increment or decrement are required in order to establish the desired address, these steps would be included as additional operations in the indirect branch. The indirect address is read from memory into Q in step 16; then a multiway branch is performed depending upon auto increment or decrement or multi indirect chain is required in the address calculation.

There are two loops in the microcode for indirect address calculation, one for only 1 level indirect address chain (or the last step in a multi chain) and one for the second indirect address in a chain (BIT 0 = 1 in the word read out from memory in step RAR 16). Both loops test for auto increment or decrement.

For increment or decrement memory and skip on condition (ISZ, DSZ), the addressed memory location is incremented or decremented and stored in Q (RAR 35, 36). Q is stored back into the addressed memory location (RAR 37); Q is tested for "0", and PC is incremented 2 steps (RAR 131, 130) if Q is "0".

4.3.2 Jump Instruction

4.3.2

Refer to FC2 and FC4.

If the instruction to be performed is a jump direct, such as a JMP, JMP PC, JMP AC2 or JMP AC3, control passes immediately to one block which executes the instruction.

In the case of the JMP, the contents of Q are loaded into the program counter and simultaneously loaded into the memory address register in preparation for fetch of the next instruction (RAR 2). This is a jump directly to one of the 256 locations in the base page. If the JMP PC, JMP AC2 or JMP AC3 instruction is executed, the contents of Q are added to the contents of the program counter (RAR 3), accumulator AC2 (RAR 4) or accumulator AC3 (RAR 5) respectively, and loaded into the program counter and the memory address register.

After the program counter and memory address register have been loaded, multiway branches are performed under the control of the interrupt request, DMA request or the HALT request. If one of these three conditions prevail, the microprogram will branch to one of the three blocks of servicing microcode associated with these functions. Otherwise, the microprogram will continue to instrution fetch.

For jump to subroutine (JSR) the prior value in PC incremented one (pointing to the instruction after JSR) is stored in AC3 and the new address is loaded into PC and MAR. The program counter is "clipped" to 15 bits (RAR 27-30). In other words, the high order bit of the program counter is set to 0.

4.3.3 Arithmetic Instruction

4.3.3

Refer to FC5.

Arithmetic instructions set up the ALU to execute the arithmetic instruction in one step if possible.

Arithmetic instructions which do not shift can be executed in 2 steps; one step to set up the ALU function (RAR 42-51) and one dummy step to test the result for 0 (RAR 56).

Arithmetic instructions which shift, then test the result for 0, are executed in 4 steps; one step does the arithmetic combination, one step to shift Q (RAR 62-63 if LOAD, RAR 64-65 if NO LOAD), one step loads Q into the destination accumulator if LOAD is specified, or loads Q into Q if NO LOAD is specified, and one dummy step to test the result for 0.

Arithmetic instructions requiring byte swap use 7 microsteps; one step does the arithmetic combination, 5 step to shift Q left 8 places (this is done by using AC15 temporarily, performing the function AC15 + AC15 and shifting the result before storing back into AC15, giving a total shift of 2 steps left), and one dummy step to test the result for 0.

4.3.4 I/O Instructions

Refer to FC6.

In the first microstep of the I/O instruction the I/O device address is gated from the last six bits of the instruction register. These bits represent the device code -,DS 0 to -,DS 5. This gating is performed through the whole I/O instruction executing.

An I/O decode branch is made which indicate whether a normal I/O instruction or an I/O instruction to device 77, device 1 or device 2 is to be executed. Device code 77 is used to perform special functions as INTEN, INTDS, READS, MSKO, INTA, IORST and HALT. Device code 1 is used for the memory extension instruction DICP ac, 1 and the instructions LDB and STB. Device code 2 is used for extended 3803 instruction set.

-,GENIODT is used to generate DATO A-C, DATI A-C pulses and -,GENIOP is used to generate STRT, CLR and IOPLS signals.

The I/O decode branch selects one of six blocks of microcode for normal I/O instruction execution. These four blocks correspond with the I/O functions of No I/O, Data In, Data Out, I/O Skip,

device 1 and device 2. At the end of any of the I/O execute states, the normal multi-way branch based on Skip, Diagnostic Panel, DMA or interrupt is made in exit to the instruction field.

4.4 Interrupt 4.4

Refer to FC7, FC3.

The interrupt service microcode is shown on Flow Diagram FC7. If any interrupt request is honored at the microcode branch point prior to fetching the next instruction, this routine is entered.

The first microstep (RAR 107) loads the memory address register with zeroes, and the interrupt enable flip-flop is cleared to lock out any additional interrupt. In the next microstep (RAR 110), the contents of the program counter is stored in memory location 0. In the microstep (RAR 111), the memory address register is forced to point upon memory location 1.

The next microstep (RAR 16) transfers the contents of memory to Q, in microstep RAR 22 the contents of Q is loaded into the memory address register, and the last microstep (RAR 31) transfers the contents of Q into the program counter completing the interrupt sequence. At this point, the routine exits to the normal op-fetch entry. The user's interrupt program picks up at this point to implement the specific interrupt routine required by the user.

The mask out instruction is used to control the state of the interrupt Disable flag in the device. When a device completes a given operation, it automatically sets its Done flag. If the Interrupt Disable flag is clear, the device will request another interrupt. If the Interrupt Disable flag is set, the device cannot set another interrupt. The mask out instruction causes the Interrupt Disable flags to be set based on the contents of the accumulator specified in the instruction. A zero mask clears all Interrupt Disable flags.

The priority of the device operating in the interrupt mode is based on the physical location of the device controller on the I/O BUS. When the Internal Acknowledge instruction is issued, the device code for the device which is physically closest to the CPU and whose Done flag is set and whose Interrupt Disable flag is clear is loaded into the specified accumulator and the device is serviced. Device priority can also be controlled by use of the SKIP instruction.

4.5 DMA 4.5

Refer to FC7 and FC8.

Four DMA modes are available (DMA IN, DMA OUT, DMA INCREMENT and DMA ADD TO MEMORY).

DMA entry occurs if a DMA request (-,DCHR) is honored at the normal microprogram multi-way branch point before the next instruction fetch. During the first part of the DMA microcode (FC7) Data Channel Acknowledge (DCHA) is sent out on the I/O BUS to get a memory address from the device with highest priority among the devices requesting data channel service. Then a multi-way branch is performed to the function determined by -,DCHM0 and -,DCHM1 from the current device. All data channel requests are honored according to the relative position of the requesting device on the I/O BUS. That device requesting data channel service which is physically closest on the bus is serviced first, then the next closest device, and so on, until all requests have been honored. The synchronization of new requests occurs concurrently with the honoring of other requests, so if a device continually requests the data channel, that device can prevent all devices further out on the bus gaining access to the channel.

4.6 Diagnostic Panel 4.6

Refer to FC9.

Special microcodes provide for interaction with the Diagnostic Panel (TCP 701). The diagnostic panel routines are entered when the RUN flip-flop is off.

The diagnostic panel routines provide for Examine, Examine Next, Deposit, Deposit next, autoload and Examine, Deposit any of the 16 registers and START.

4.7 BREAK 4.7

Refer to FC10.

When the BREAK Switch on the Diagnostic Panel is enabled, all normal instructions return are changed to jump directly to the BREAK routine. There are 4 BREAK routines, one for SKIP instructions, one for arithmetic instructions, one for I/O SKIP and one for all other instructions.

During the BREAK routine the register AC5 is compared with the program counter, and if equal the next instruction to be executed is fetched from AC6 and not from memory, else the normal multiway branch based on Skip, Diagnostic Panel requests, DMA or interrupt is made in exit to the instruction field.

5. MICROPROGRAM FORMAT DESCRIPTION 5.

A microprogram word is 86 bits.

On fig. 7 is shown a layout of the microprogram word where the signals are listed in logical fields. These fields are described in detail in the following sections.

5.1 Next Address Field 5.1

This field is used to specify a Next Address if the microprogram should go to this address unconditionally. The Next Address Field consists of 9 bits, resulting in 512 direct jump possibilities. NXAD 0-7 is coded binary to give the next microaddress, NXAD 7 is the least significant bit. NXADP selects one of the two 256 word pages (BANK 0 and BANK 1). The lower page (NXADP=0) is called BANK 0 and the upper page (NXADP=1) is called BANK 1.

5.2 Next Address Select Field 5.2

This field is used to determine whether the next address calculation is conditional, unconditional or jump from fetch cycle to the start address of the sequence to execute the current fetched instruction.

This field consists of 8 bits, NXAD SEL A-C (NeXt ADdress SELect A-C), SEL 1ST OP (SELect 1ST OPerand), SEL 2ND OP (SELect 2ND OPerand) and MPLSB SEL 0-2 (MicroProgram LSB SELect 0-2). The logical function of the field which consists of NXAD SEL A-C and SEL 1ST OP is shown in fig. 7. SEL 2ND OP is used to separate next address map in BANK 0 and BANK 1.

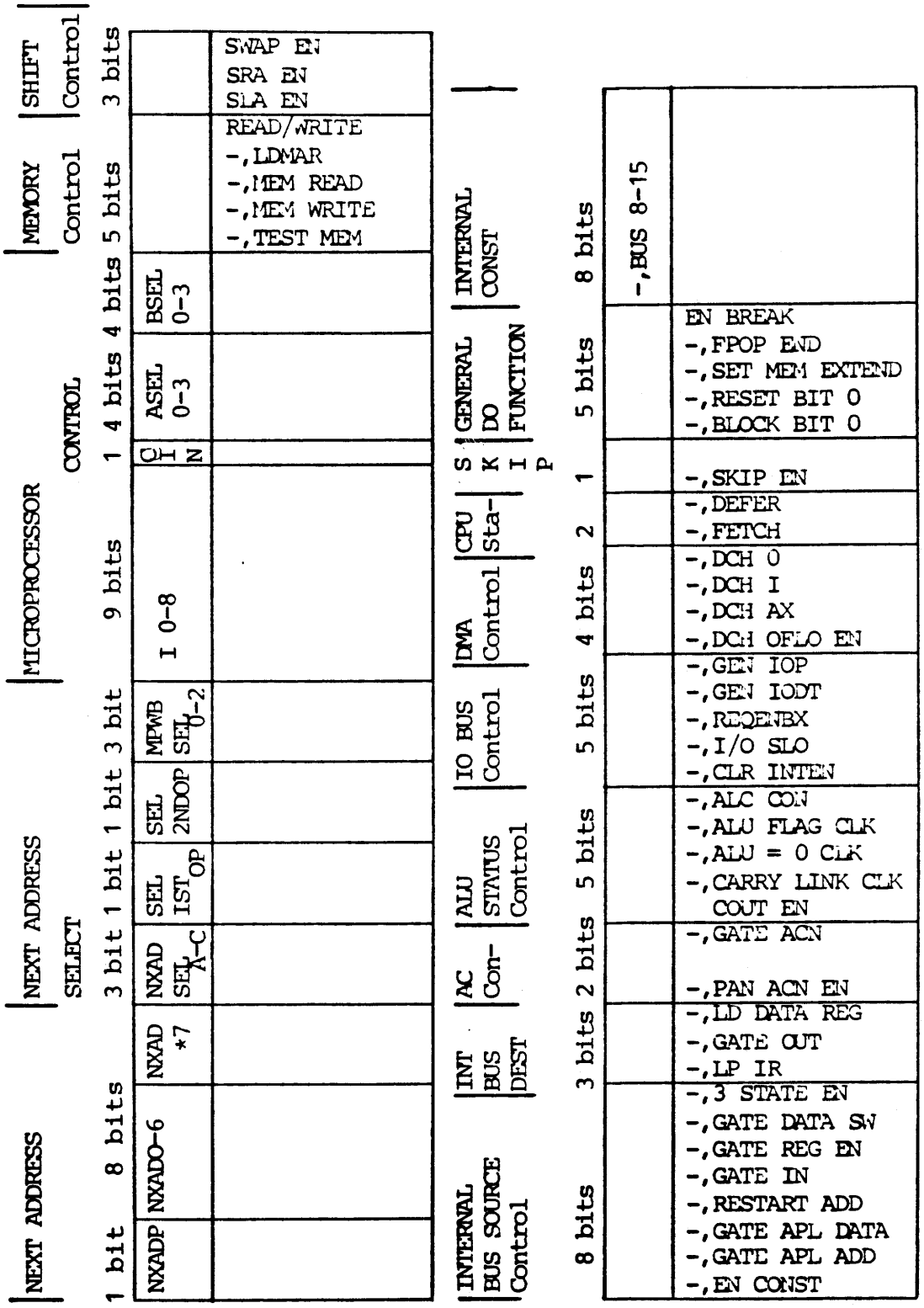


Figure 7: Microinstruction Word Format.

NXAD SEL 1ST OP	C	B	A	FUNCTION
0	0	0	0	SELECT CONDITION GROUP 0
0	0	0	1	- - - 1
0	0	1	0	- - - 2
0	0	1	1	- - - 3
0	1	0	0	- - - 4
0	1	0	1	- - - 5
0	1	1	0	- - - 6
0	1	1	1	- - - 7
1	0	0	0	NOT USED
1	0	0	1	NOR USED
1	0	1	0	NOT USED
1	0	1	1	SELECT FETCH JUMP
1	1	0	0	Unconditional JUMP
1	1	0	1	NOT USED
1	1	1	0	Unconditional JUMP
1	1	1	1	NOT USED

If unconditional jump is selected, the Next Address Field (section 5.1) is used to specify the Next Address.

If conditional jump is selected, NXAD C-A together with the output from the condition selector is used as an 8 bits address to a Conditional Next Address table (subsection 6.2.1). If fetch jump is selected, the value on the memory bus (BUS 0-7) is used as an 8 bits address to a Start Address table.

MPLSB SEL 0-2 are used to select one of eight signals as source for the LSB of the next microprogram address (NXAD 7) (section 6.3).

5.3 Microprocessor Control Field

5.3

This field contains the necessary control lines to the microprocessor AM 2901.

This field is logically divided into three subfields:

I 0-8, CIN ; microinstruction field
A 0-3 ; Read Only address input to the register stack in AM 2901
B 0-3 ; Read-Modify-Write address input to the register stack.

5.3.1 Microinstruction Field 5.3.1

There are eight source operand pairs available to the ALU as selected by I 0-2.

The ALU can perform eight functions; five logic and three arithmetic. The I 3-5 instruction inputs control this function selection. The carry input, CIN, also affects the ALU results when in the arithmetic mode. The CIN input has no effect in the logic mode. I 6-8 is used to determine whether data from the ALU, shifted or unshifted, is to be deposited in the Q-register or in the register stack. For further details about the microinstruction control lines to AM 2901, refer to chapter 3.

5.3.2 Register Stack Address Inputs 5.3.2

The Register Stack consists of 16 words of Random Access Memory.

Data in any of the 16 registers can be read from the A-port of the Stack by the 4-bit A address field input to AM 2901. Likewise, data in any of the 16 registers as defined by the B address field input can be simultaneously read from the B-port of the Stack.

New data is always written into the register defined by the B address field. For further detail about AM 2901, refer to chapter 3.

The layout of the register Stack is shown in the following table:

A(B) SEL 0	A(B) SEL 1	A(B) SEL 2	A(B) SEL 3	REGISTER MNEMONIC
0	0	0	0	AC0
0	0	0	1	AC1
0	0	1	0	AC2
0	0	1	1	AC3
0	1	0	0	PC (program Counter)
0	1	0	1	BREAK ADDRESS
0	1	1	0	BREAK INSTRUCTION
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	SPARE REGISTERS
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	USED TEMPORARILY DURING REGISTER SHIFT

5.4 Memory Control 5.4

This field contains the necessary control lines to control memory.

-,LD MAR (-,LoaD Memory Address Register) loads the contents on the internal data bus into the memory address register.

-,MEM READ (-,MEMory READ) starts the memory in a Read cycle from the address specified by the contents of the memory address register.

-,MEM WRITE (-,MEMory WRITE) starts the memory with a write cycle of the data on the internal bus into the memory address specified by the contents of the memory address register.

-,TEST MEM is used by the microprogram to test the state of the Memory Busy Status.

READ/WRITE generates the START signal to the memory in a read or write cycle.

5.5 Shift Control 5.5

This three bits field is used to select the new inputs to the Q-register or the Register Stack during shift operations.

SLAEN (Shift Left ENable) connects the SHIFT GENERATOR (chapter 7) to the LSB end of the Q-register (Q15 SLI/SRO).

SRA EN (Shift R ENable) connects the SHIFT GENERATOR (chapter 7) to the MSB end of the Q-register (Q0 SRI/SLO).

SWAP EN (SWAP ENable) connects the MSB output from the Register Stack Shifter (R0 SRI/SLO) to the LSB input to the Register Stack Shifter (R15 SLI/SRO).

5.6 Internal Bus Source Control 5.6

This field contains control lines to determine the source for the internal data bus.

-,GATE APL ADD (-,GATE Automatic Program Load ADDress) gates the contents of the Automatic Program Load Address Counter out on the data bus.

-,GATE APL DATA (-,GATE Automatic Program Load DATA) gates the contents in the addressed location of the autoload PROM's out on the data bus.

-,RESTART ADD (-,RESTART ADDress) gates the contents of the Data Switches (TCP 701) out on the data bus and sets the CPU in RUN state. If the Diagnostic Panel is not connected, a "0" is gated out on the data bus.

-,GATE IN (-,GATE INput) gates the contents on the I/O Data Bus out on the internal data bus.

-,GATE REG EN (-,GATE REGister ENable) gates the contents of the Register Select Switches (AC SEL 0-3) out on the internal data bus bit 1-4. This function is only used if the Diagnostic Panel TCP 701 is enabled.

-,GATE DATA SW (-,GATE DATA Switches) gates the contents of the Data switches on the Diagnostic Panel or the Data switches on the Front Panel of CPU 708 out on the data bus depending on the state of the ENABLE TCP switch.

-,3 STATE EN (-,3 STATE ENable) gates the data from AM 2901 out on the internal data bus.

-,EN CONST (ENable CONSTant) gates a constant from the microprogram PROM out on the internal bus.

5.7 Internal Bus Destination Control 5.7

This field contains control lines to determine the destination for the contents on the internal data bus.

-,LD IR (-,LoaD Instruction Register) loads the contents on the data bus into the instruction register.

-,GATE OUT (-,GATE OUTput) gates the contents on the internal data bus out on the I/O Data BUS.

-,LD DATA REG (-,Load DATA REGister) loads the contents on the internal data bus into Data Register. The contents of the Data Register is shown on the Diagnostic Panel, if connected.

5.8 Accumulator Control 5.8

This field contains control Lines to determine whether the Re-

gister Stack Address Inputs (section 5.3.2) is delivered directly in the microprogram word or decoded from the instruction register when one or two accumulator instruction is executed.

-,GATE ACN (-,GATE ACcumulator eNable) disables ASEL 0-3 and BSEL 0-3 from the microprogram and enables decoding of these lines from the instruction register bit 1-4.

-,PAN ACN EN (-,PANel ACcumulator ENable) enables decoding of ASEL 0-3 and BSEL 0-3 from the state of ACSEL switches on the Diagnostic Panel.

5.9 ALU Status Control

5.9

This field contains control lines to control updating of the carry register and ALU = 0 status register with the status from the ALU after executing an arithmetic or logic instruction.

COUT EN (Carry OUTput ENable) enables the carry output from the AM 2901 array. -,CARRY LINK CLK (-,CARRY LINK CLocK) loads the new calculated carry from the CARRY GENERATOR (chapter 7) into the Carry Register.

-,ALU = 0 CLK (-,ALU = 0 CLocK) loads the state of the zero decoder output from AM 2901 into the ALU = 0 Register.

-,ALU FLAG CLK (-,ALU FLAG CLocK) is used to latch the state of BUS 0 and BUS 15 of the internal data bus, the enabled carry output from the AM 2901 array and the overflow (OVR) status from AM 2901 array after an arithmetic-logic microfunction. The latched BUS 0, BUS 15 and carry are used by the CARRY/SHIFTER GENerator (chapter 7).

-,ALC CON (-,ALu Condition CONtrol) enables a decoder on instruction register bit 8-15 to determine if the result from the arithmetic-logic operation should be loaded into the Q-register or in the destination accumulator specified in the two accumulator instruction formats.

5.10 I/O Bus Control 5.10

This field contains the necessary control Lines to generate the control signals on the programmed I/O BUS (chapter 11).

-,CLR INTEN (-,CLeaR INTerrupt ENable) clears Interrupt On to prevent the processor from responding to interrupt requests.

-,I/O SLO (-,Input/Output device SeLect Out) places instruction register bit 10-15 on the Device Selection lines (-,DS 0 - -,DS 5)).

-,RQENB X (-,ReQusst ENaBle X) allows all devices on the I/O BUS to request program interrupts or data channel access. -,GEN IODT (-,GENerate Input/Output Data pulse) is used to generate data strobe pulses on the I/O BUS (i.e., DATO A-B, DATI A-B, INTA, IORST and MSKO) depending on the contents of the instruction register bit 5-7 and bit 10-15.

-,GEN IOP (-,GENerate Input/Output Pulses) is used to generate I/O BUS control signals as STRT, CLR, IOPL and to set/reset Interrupt On flag depending on instruction register bit 8-9 and bit 10-15.

5.11 DMA Control 5.11

By the control lines in this field the Data Channel is controlled.

-,DCH OFLO EN (-,Data CHannel OverFLOw ENable) is used to strobe the OVerFlow status out on the Data Channel control line OVFLO. OVFLO is generated by the processor during a data channel cycle that increments memory or adds to memory.

-,DCHA X (-,Data CHannel Acknowledge) is generated by the processor at the beginning of a data channel cycle. If a device receives DCHA while it is also receiving DCHP and its DCH REQ flip-flop is set, it places the memory address to be used for data channel access on the I/O Data lines 0-15.

-,DCHI (-,Data CHannel Input) is generated by the processor for data channel input to place the data register of the device selected by DCHA on the data lines.

-,DCHO (-,Data CHannel Output) is generated by the processor for data channel output after the word from memory or the arithmetic result has been placed on the data lines to load the contents of the lines into the data register of the devices selected by DCHA.

5.12 CPU Status 5.12

These two status lines are used to drive indicators on the front panel of CPU708.

-,FETCH indicates that the CPU is reading an instruction from memory.

-,DEFER indicates that the next microcycle will be used to follow an indirection chain.

5.13 General DO Functions 5.13

This field contains 5 control lines, which cannot be placed in the other standard fields.

-,BLOCK BIT0 is used to "clip" bit 0 when transferring PC to memory address register.

-,RESET BIT0 is used to place logical zero on the MSB shift input (R0 SRI/SLO) of the register stack shifter. This function is used in the jump subroutine instruction to mask out bit 0 in the PC before saving PC.

-,SET MEM EXTEND is used to set the flip-flop enabling bus 0 as input to the address register.

-,FPOP END (-,Front Panel OPeration END) indicates the end of a front panel operation loop in the microprogram.

EN BREAK (ENable BREAK) used during the break loop to lock the break switch, so operation of this switch is not registrated by the CPU before the break loop is terminated.

5.14 SKIP Control 5.14

This field contains only one control line, -,SKIP EN (-,SKIP ENable), used to enable the SKIP condition GENerator (chapter 8).

5.15 Internal Constant 5.15

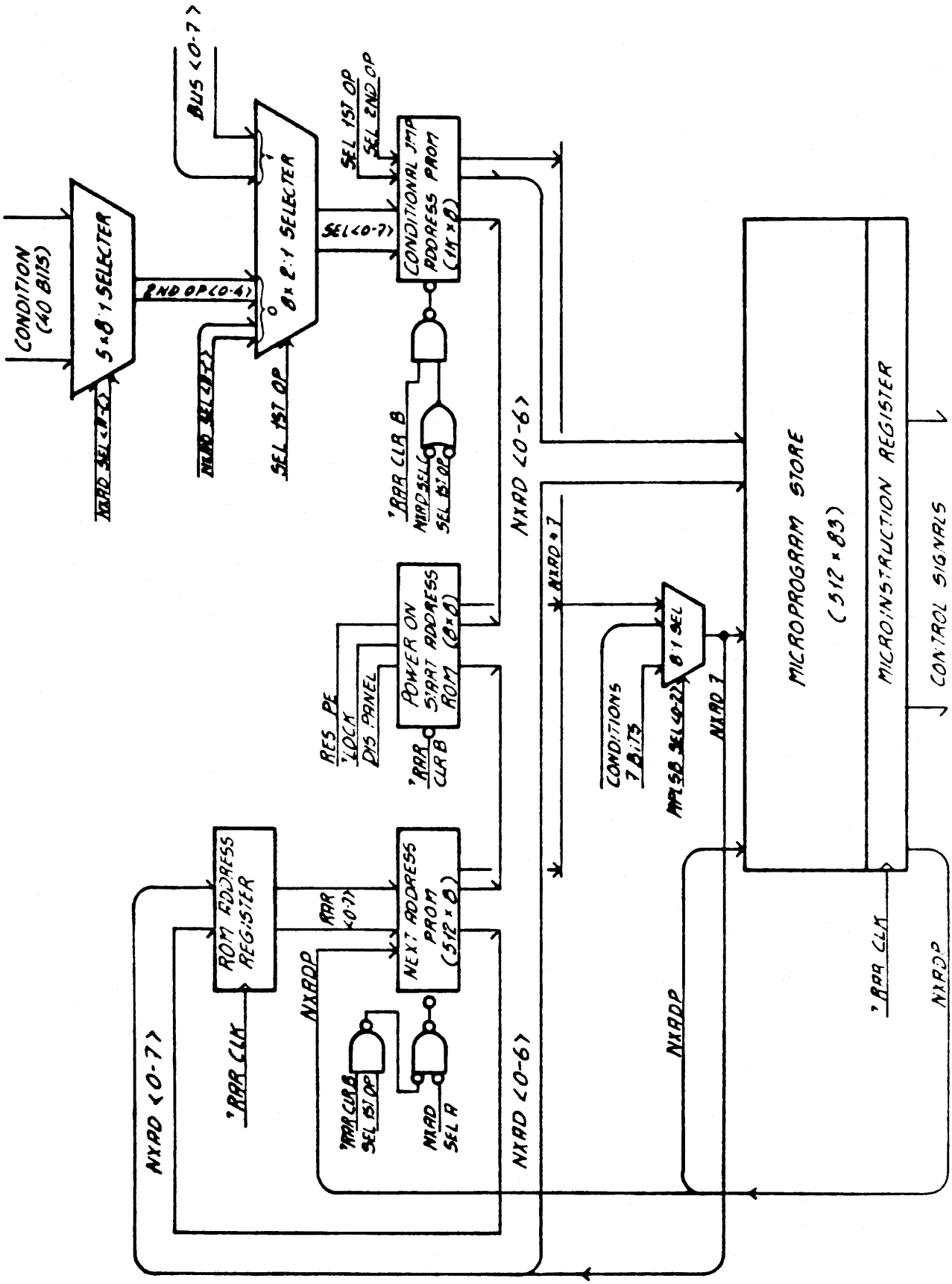
This field provides the CPU721 with an 3 bit constant. The constant is gated out on the internal data bus -,DBUS8-15 (see 5.6).

6. NEXT MICROADDRESS CALCULATION 6.

The sequence of the microprogram states is determined by the next address controls. The next address controls decode the contents of the instruction register and generate a series of ROM addresses necessary to execute the instruction in question.

Other indicators as carry-out and zero detect are also monitored by the next address control to modify the address sequence as required for proper instruction execution.

For details about the next microaddress calculation, refer to block diagram, fig. 8.



CPU 721

Figure 8: MICRO SEQUENSER BLOCK DIAGRAM FIG 8.0

6.1 Condition Selector 6.1

This circuit consists of 5 multiplexers (8 to 1). The Condition Group to be selected is determined by the next address select field (NXAD SEL A-C) contained in each microword. In fig. 9 is shown the relation between NXAD SEL A-C and the Condition Group to be selected.

SEL	NXAD				
1ST OP	C	B	A	CONDITION GROUP	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	Conditional
0	1	0	0	4	JUMP
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0		
1	0	0	1		NOT USED
1	0	1	0		
1	0	1	1	FETCH	
1	1	0	0	Unconditional jump	
1	1	0	1	NOT USED	
1	1	1	0	Unconditional jump	
1	1	1	1	NOT USED	

Figure 9: Condition Group select.

In the following sections each Condition Group is described.

In fig. 20 is shown a table containing a list of signals for each condition group.

NEXT ADDR MAP FOR CONDITION GROUP 0, BANK 0. ADDR 0(Oct)8-37(Oct) IN ROB 382-383

BASE ADDR	IR5	IR6	IR7	IR8	IR9	NEXT ADDR (OCTAL)	FUNCTION
00000	0	0	0	0	0	34	
	0	0	0	0	1	34	
	0	0	0	1	0	34	
	0	0	0	1	1	34	
	0	0	1	0	0	34	
	0	0	1	0	1	34	
	0	0	1	1	0	34	
	0	0	1	1	1	34	
	0	1	0	0	0	34	
	0	1	0	0	1	34	
	0	1	0	1	0	34	
	0	1	0	1	1	34	
	0	1	1	0	0	34	
	0	1	1	0	1	34	
	0	1	1	1	0	34	
	0	1	1	1	1	34	
	1	0	0	0	0	34	
	1	0	0	0	1	34	
	1	0	0	1	0	34	
	1	0	0	1	1	34	
	1	0	1	0	0	34	
	1	0	1	0	1	34	
	1	0	1	1	0	375	LDB
	1	0	1	1	1	212	MEM EXTEND
	1	1	0	0	0	34	<SPARE>
	1	1	0	0	1	34	
	1	1	0	1	0	376	STB
	1	1	0	1	1	34	
	1	1	1	0	0	34	<SPARE>
	1	1	1	0	1	34	
	1	1	1	1	0	34	
	1	1	1	1	1	34	

Figure 10: Next ADDR MAP for Condition Group 0, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 0, BANK 1. ADDR 1000(Oct)-1037(Oct) IN ROB 382-383

BASE ADDR	IR5	IR6	IR7	IR8	IR9	NEXT ADDR (OCTAL)	FUNCTION
10000	0	0	0	0	0	3	
	0	0	0	0	1	3	<SPARE>
	0	0	0	1	0	3	
	0	0	0	1	1	3	
	0	0	1	0	0	1	IDFY
	0	0	1	0	1	3	
	0	0	1	1	0	3	
	0	0	1	1	1	3	
	0	1	0	0	0	3	
	0	1	0	0	1	3	
	0	1	0	1	0	3	
	0	1	0	1	1	3	<SPARE>
	0	1	1	0	0	3	
	0	1	1	0	1	3	
	0	1	1	1	0	3	
	0	1	1	1	1	3	
	1	0	0	0	0	3	
	1	0	0	0	1	3	
	1	0	0	1	0	3	
	1	0	0	1	1	3	
	1	0	1	0	0	100	BMOVE
	1	0	1	0	1	4	WMOVE
	1	0	1	1	0	120	SCHEL
	1	0	1	1	1	155	SFREE
	1	1	0	0	0	273	LINK
	1	1	0	0	1	304	REMEL
	1	1	0	1	0	320	PLINK
	1	1	0	1	1	171	FETCH(Musil)
	1	1	1	0	0	240	IKADD
	1	1	1	0	1	214	TKVAL
	1	1	1	1	0	61	COMP
	1	1	1	1	1	3	<SPARE>

Figure 11: Next ADDR MAP for Condition Group 0, Bank 1.

NEXT ADDR MAP FOR CONDITION GROUP 1, BANK 0. ADDR 40(Oct)-77(Oct) IN ROB 382-383

BASE ADDR	-,DEV 01/ xx	-,DEV 77/ xx	IR5	IR6	IR7	NEXT ADDR (OCTAL)	FUNCTION
00001	0	0	0	0	0	256	
	0	0	0	0	1	256	
	0	0	0	1	0	256	
	0	0	0	1	1	256	DEV 2
	0	0	1	0	0	256	INSTRUCTIONS
	0	0	1	0	1	256	(RC3803)
	0	0	1	1	0	256	
	0	0	1	1	1	256	
	0	1	0	0	0	211	
	0	1	0	0	1	211	
	0	1	0	1	0	211	
	0	1	0	1	1	211	DEV 1
	0	1	1	0	0	211	INSTRUCTIONS
	0	1	1	0	1	211	(DICP ac. 1,
	0	1	1	1	0	211	LDB and STB)
	0	1	1	1	1	132	SKIP
	1	0	0	0	0	73	NIO CPU
	1	0	0	0	1	71	DIA CPU
	1	0	0	1	0	72	DOA CPU
	1	0	0	1	1	71	DIB CPU
	1	0	1	0	0	72	DOB CPU
	1	0	1	0	1	73	DIC CPU
	1	0	1	1	0	73	DOC CPU
	1	0	1	1	1	132	SKIP
	1	1	0	0	0	73	NIO
	1	1	0	0	1	71	DIA
	1	1	0	1	0	72	DOA
	1	1	0	1	1	71	DIB
	1	1	1	0	0	72	DOB
	1	1	1	0	1	71	DIC
	1	1	1	1	0	72	DOC
	1	1	1	1	1	132	SKIP

Figure 12: Next ADDR MAP for Condition Group 1, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 2, BANK 0. ADDR 100(Oct)-137(Oct) IN ROB 382-383

BASE ADDR	APL B	-,FPOP BUSY B	-,FPOP 2B	-,FPOP IB	-,FPOP 0B	NEXT ADDR (OCTAL)	FUNCTION
00010	0	0	0	0	0	126	NOP
	0	0	0	0	1	123	DEP NEXT
	0	0	0	1	0	124	DEP MEM
	0	0	0	1	1	120	EXAM NEXT
	0	0	1	0	0	121	EXAM MEM
	0	0	1	0	1	117	DEP REG
	0	0	1	1	0	116	EXAM REG
	0	0	1	1	1	207	START
	0	1	0	0	0	126	
	0	1	0	0	1	126	
	0	1	0	1	0	126	
	0	1	0	1	1	126	NOP
	0	1	1	0	0	126	
	0	1	1	0	1	126	
	0	1	1	1	0	126	
	0	1	1	1	1	126	
	1	0	0	0	0	257	
	1	0	0	0	1	257	
	1	0	0	1	0	257	
	1	0	0	1	1	257	
	1	0	1	0	0	257	
	1	0	1	0	1	257	
	1	0	1	1	0	257	
	1	0	1	1	1	257	AUTO
	1	1	0	0	0	257	LOAD
	1	1	0	0	1	257	
	1	1	0	1	0	257	
	1	1	0	1	1	257	
	1	1	1	0	0	257	
	1	1	1	0	1	257	
	1	1	1	1	0	257	
	1	1	1	1	1	257	

Figure 13: Next ADDR MAP for Condition Group 2, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 2, BANK 1. ADDR 1100(Oct)-1137(Oct) IN ROB 382-383

BASE ADDR	APL B	-,FPOP BUSY B	-,FPOP 2B	-,FPOP IB	-,FPOP 0B	NEXT ADDR (OCTAL)	FUNCTION
00010	0	0	0	0	0	3	
	0	0	0	0	1	3	
	0	0	0	1	0	3	
	0	0	0	1	1	3	
	0	0	1	0	0	3	
	0	0	1	0	1	3	
	0	0	1	1	0	3	
	0	0	1	1	1	3	
	0	1	0	0	0	3	
	0	1	0	0	1	3	
	0	1	0	1	0	3	
	0	1	0	1	1	3	
	0	1	1	0	0	3	
	0	1	1	0	1	3	No Conditional
	0	1	1	1	0	3	Jump
	0	1	1	1	1	3	
	1	0	0	0	0	3	
	1	0	0	0	1	3	
	1	0	0	1	0	3	
	1	0	0	1	1	3	
	1	0	1	0	0	3	
	1	0	1	0	1	3	
	1	0	1	1	0	3	
	1	0	1	1	1	3	
	1	1	0	0	0	3	
	1	1	0	0	1	3	
	1	1	0	1	0	3	
	1	1	0	1	1	3	
	1	1	1	0	0	3	
	1	1	1	0	1	3	
	1	1	1	1	0	3	
	1	1	1	1	1	3	

Figure 14: Next ADDR MAP for Condition Group 2, Bank 1.

NEXT ADDR MAP FOR CONDITION GROUP 3, BANK 0. ADDR 140(Oct)-177(Oct) IN ROB 382-383

BASE ADDR	EN BREAK	-,SKIP	BREAK COND	-,DCHM IB	-,DCHM OB	NEXT ADDR (OCTAL)	FUNCTION
00011	0	0	0	0	0	103	ADDR TO MEM
	0	0	0	0	1	102	INCR MEM
	0	0	0	1	0	100	DMA IN
	0	0	0	1	1	101	DMA OUT
	0	0	1	0	0	103	ADD TO MEM
	0	0	1	0	1	102	INCR MEM
	0	0	1	1	0	100	DMA IN
	0	0	1	1	1	101	DMA OUT
	0	1	0	0	0	103	ADD TO MEM
	0	1	0	0	1	102	INCR MEM
	0	1	0	1	0	100	DMA IN
	0	1	0	1	1	101	DMA OUT
	0	1	1	0	0	103	ADD TO MEM
	0	1	1	0	1	102	INCR MEM
	0	1	1	1	0	100	DMA IN
	0	1	1	1	1	101	DMA OUT
	1	0	0	0	0	140	SKIP RETURN
	1	0	0	0	1	140	
	1	0	0	1	0	140	
	1	0	0	1	1	140	
	1	0	1	0	0	137	BREAK
	1	0	1	0	1	137	
	1	0	1	1	0	137	
	1	0	1	1	1	137	
	1	1	0	0	0	141	NO SKIP RETURN
	1	1	0	0	1	141	
	1	1	0	1	0	141	
	1	1	0	1	1	141	
	1	1	1	0	0	137	BREAK
	1	1	1	0	1	137	
	1	1	1	1	0	137	
	1	1	1	1	1	137	

Figure 15: Next ADDR MAP for Condition Group 3, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 4, BANK 0. ADDR 200(Oct)-237(Oct) IN ROB 382-383

BASE ADDR	RUN B	-,DCHR SYNC	INTEN DLY'D	-,INTR SYNC	SKIP B	NEXT ADDR (OCTAL)	FUNCTION
00100	0	0	0	0	0	130	SKIP
	0	0	0	0	1	127	FRONT PANEL
	0	0	0	1	0	130	SKIP
	0	0	0	1	1	127	FRONT PANEL
	0	0	1	0	0	130	SKIP
	0	0	1	0	1	127	FRONT PANEL
	0	0	1	1	0	130	SKIP
	0	0	1	1	1	127	FRONT PANEL
	0	1	0	0	0	130	SKIP
	0	1	0	0	1	127	FRONT PANEL
	0	1	0	1	0	130	SKIP
	0	1	0	1	1	127	FRONT PANEL
	0	1	1	0	0	130	SKIP
	0	1	1	0	1	127	FRONT PANEL
	0	1	1	1	0	130	SKIP
	0	1	1	1	1	127	FRONT PANEL
	1	0	0	0	0	130	SKIP
	1	0	0	0	1	76	DMA
	1	0	0	1	0	130	SKIP
	1	0	0	1	1	76	DMA
	1	0	1	0	0	130	SKIP
	1	0	1	0	1	76	DMA
	1	0	1	1	0	130	SKIP
	1	0	1	1	1	76	DMA
	1	1	0	0	0	130	SKIP
	1	1	0	0	1	1	FETCH
	1	1	0	1	0	130	SKIP
	1	1	0	1	1	1	FETCH
	1	1	1	0	0	130	SKIP
	1	1	1	0	1	107	INTERRUPT
	1	1	1	1	0	130	SKIP
	1	1	1	1	1	1	FETCH

Figure 16: Next ADDR MAP for Condition Group 4, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 5, BANK 0. ADDR 240(Oct)-277(Oct) IN ROB 382-383

BASE ADDR	SELECT RUN B	-,DC RSTB -,DCHR SYNC	BUS INTEN DLY'D	-,AI ADR -,INTR SYNC	AI DCR SKIP B	NEXT ADDR (OCTAL)	FUNCTION
00101	0	0	0	0	0	127	
	0	0	0	0	1	127	
	0	0	0	1	0	127	
	0	0	0	1	1	127	RESET
	0	0	1	0	0	127	
	0	0	1	0	1	127	
	0	0	1	1	0	127	
	0	0	1	1	1	127	
	0	1	0	0	0	20	AUTO INC, 1 level
	0	1	0	0	1	17	AUTO DECR, 1 level a
	0	1	0	1	0	22	a, 1 level
	0	1	0	1	1	22	a, 1 level
	0	1	1	0	0	24	AUTO INC, 2ND level a
	0	1	1	0	1	23	AUTO DEC, 2ND level a
	0	1	1	1	0	26	SECOND a
	0	1	1	1	1	26	SECOND a
	1	0	0	0	0	127	
	1	0	0	0	1	127	
	1	0	0	1	0	127	
	1	0	0	1	1	127	RESET
	1	0	1	0	0	127	
	1	0	1	0	1	127	
	1	0	1	1	0	127	
	1	0	1	1	1	127	
	1	1	0	0	0	20	AUTO INC, 1 level a
	1	1	0	0	1	17	AUTO DEC, 1 level a
	1	1	0	1	0	22	a, 1 level
	1	1	0	1	1	22	a, 1 level
	1	1	1	0	0	20	AUTO INC, 1 level a
	1	1	1	0	1	17	AUTO DEC, 1 level a
	1	1	1	1	0	22	a, 1 level
	1	1	1	1	1	22	a, 1 level

Figure 17: Next ADDR MAP for Condition Group 5, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 6, BANK 0. ADDR 300(Oct)-337(Oct) IN ROB 382-383

BASE ADDR	IR8	IR9	IR12	IR13	IR14	NEXT ADDR (OCTAL)	FUNCTION
00110	0	0	0	0	0	56	SKP
	0	0	0	0	1	56	SZC,SNC
	0	0	0	1	0	56	SZR,SNR
	0	0	0	1	1	56	SEZ,SBN
	0	0	1	0	0	56	#SKP
	0	0	1	0	1	56	#SZC,SNC
	0	0	1	1	0	56	#SZR,SNR
	0	0	1	1	1	56	#SEZ,SBN
	0	1	0	0	0	62	L SKP
	0	1	0	0	1	62	L SZC,SNC
	0	1	0	1	0	62	L SZR,SNR
	0	1	0	1	1	62	L SEZ,SBN
	0	1	1	0	0	56	L#SKP
	0	1	1	0	1	64	L#SZC,SNC
	0	1	1	1	0	64	L#SZR,SNR
	0	1	1	1	1	64	L#SEZ,SBN
	1	0	0	0	0	63	R SKP
	1	0	0	0	1	63	R SZC,SNC
	1	0	0	1	0	63	R SZR,SNR
	1	0	0	1	1	63	R SEZ,SBN
	1	0	1	0	0	56	R#SKP
	1	0	1	0	1	65	R#SZC,SNC
	1	0	1	1	0	65	R#SZR,SNR
	1	0	1	1	1	65	R#SEZ,SBN
	1	1	0	0	0	52	S SKP
	1	1	0	0	1	52	S SZC,SNC
	1	1	0	1	0	52	S SZR,SNR
	1	1	0	1	1	52	S SEZ,SBN
	1	1	1	0	0	56	S#SKP
	1	1	1	0	1	56	S#SZC,SNC
	1	1	1	1	0	56	S#SZR,SNR
	1	1	1	1	1	56	S#SEZ,SBN

Figure 18: Next ADDR MAP for Condition Group 6, Bank 0.

NEXT ADDR MAP FOR CONDITION GROUP 7, BANK 0. ADDR 340(Oct)-377(Oct) IN ROB 382-383

BASE ADDR	SELECT	IR1	IR2	IR3	IR4	NEXT ADDR (OCTAL)	FUNCTION
00111	0	0	0	0	0	31	JUMP a
	0	0	0	0	1	27	JSR
	0	0	0	1	0	35	ISZ
	0	0	0	1	1	36	DSZ
	0	0	1	0	0	32	LDA0
	0	0	1	0	1	32	LDA1
	0	0	1	1	0	32	LDA2
	0	0	1	1	1	32	LDA3
	0	1	0	0	0	33	STA0
	0	1	0	0	1	33	STA1
	0	1	0	1	0	33	STA2
	0	1	0	1	1	33	STA3
	0	1	1	0	0	0	NOT USED
	0	1	1	0	1	0	
	0	1	1	1	0	0	
	0	1	1	1	1	0	
	1	0	0	0	0	31	JUMP a
	1	0	0	0	1	27	JSR
	1	0	0	1	0	35	ISZ
	1	0	0	1	1	36	DSZ
	1	0	1	0	0	32	LDA0
	1	0	1	0	1	32	LDA1
	1	0	1	1	0	32	LDA2
	1	0	1	1	1	32	LDA3
	1	1	0	0	0	33	STA0
	1	1	0	0	1	33	STA1
	1	1	0	1	0	33	STA2
	1	1	0	1	1	33	STA3
	1	1	1	0	0	0	NOT USED
	1	1	1	0	1	0	
	1	1	1	1	0	0	
	1	1	1	1	1	0	

Figure 19: Next ADDR MAP for Condition Group 7, Bank 0.

6.1.1 Condition Group 0, Bank 0 6.1.1

The Condition Group 0, Bank 0 consists of the Instruction Register bit 9-5 (IR 9-5).

This Condition Group is used during execution of I/O instructions to device code 01, i.e., the Memory Extension Instruction DICP ac, 1 and the instructions LDB and STB (FC6).

6.1.2 Condition Group 0, Bank 1 6.1.2

The Condition Group 0, Bank 1 consists of the Instruction Register bit 9-5 (IR 9-5).

This Condition Group is used during execution of I/O instructions to device code 2, i.e. the instructions from the Extended instruction set (FC11, FC14-21).

6.1.3 Condition Group 1, Bank 0 6.1.3

The Condition Group 1 consists of the Instruction Register bit 5-7 (IR 5-7) and the signals -,DEV 77/xx and -,DEV 01/xx.

-,DEV 77/xx is true when the Instruction Register bit 10-15 contain all "1"'s.

-,DEV 01/xx is true when the Instruction Register bit 10-15 contain device code 1.

This Condition Group is used during execution of programmed I/O instructions to determine whether normal I/O instructions or special instructions to device 77 or device 1 or device 2 are to be executed.

-,DEV 77/xx and -,DEV 01/xx are both true when the Instruction Register bit 10-15 contains device code 2. In that case an instruction from the Extended Instruction set is executed.

6.1.4 Condition Group 1, Bank 1

6.1.4

Not used.

6.1.5 Condition Group 2, Bank 0

6.1.5

The Condition Group 2 consists of -,FPOP 0-2B, -,FPOP BUSY B and APL B.

This Condition Group is used during Diagnostic Panel (FC9) to determine which Diagnostic Panel function is to be executed. The following table shows the relation between the signals -,FPOP 0-2B and the function to be executed.

-,FPOP B			Diagnostic Panel
0	1	2	Function
0	0	0	NOT USED
0	0	1	EXAM MEMORY
0	1	0	DEP MEMORY
0	1	1	EXAM REG
1	0	0	DEP NEXT
1	0	1	DEP REG
1	1	0	EXAM NEXT
1	1	1	START

The Diagnostic Panel operations are only executed if -,FPOP BUSY B = 0;

APL B = 1 overrides all other Diagnostic Panel operations, and starts an automatic program load routine.

6.1.6 Conditional Group 2, Bank 1

6.1.6

Unconditional JUMP.

6.1.7 Conditional Group 3, Bank 0 6.1.7

The Condition Group 3 consists of the signals -,DCHM 0-1B, BREAK COND, -,SKIP and EN BREAK.

This Condition Group is used during Data Channel routine and during the BREAK routine. The signal EN BREAK is used to distinguish the Data Channel routine from the BREAK routine. When EN BREAK = 0 the Data Channel routine is executed. -,DCM 0-1B determines whether data channel operation is to be executed.

-,DCHM

1B	0B	DMA FUNCTION
0	0	ADD TO MEM
0	1	INCR MEM
1	0	DMA IN
1	1	DMA OUT

When EN BREAK = 1, the BREAK routine is executed, BREAK COND indicates whether or not AC5 is equal to the Program Counter (AC4).

6.1.8 Condition 3, Bank 1 6.1.8

Not used.

6.1.9 Condition Group 4, Bank 0 6.1.9

This Condition Group consists of the signals -,SKIP, -,INTR SYNC, INTEN DLY'D, -,DCHR SYNC and RUN.

This Condition Group is used at the end of each instruction to give a multi-way branch controlled by the presence of SKIP conditions, DMA and/or Interrupt requests, and the state of the RUN flip-flop. Highest priority is given to SKIP since SKIP is part of the prior instruction.

-,INTR SYNC and -,DCHR SYNC contain Interrupt request and Data Channel request synchronized to the microprogram.

INTEN DLY'D represent the Interrupt ON flip-flop, set by the instruction INTEN, and cleared by the instruction INTDS and in the Interrupt microcode when an interrupt request is honored.

6.1.10 Condition Group 4, Bank 1 6.1.10

Not used.

6.1.11 Condition Group 5, Bank 0 6.1.11

The Condition Group 5 consists of the signals AIDCR, -,AIADR, BUS 0, -,DCRST, SELECT. This Condition Group is used during an indirect address calculation to determine whether the calculated "address" contains another indirect level or not (BUS 0) and whether the calculated address points upon an autonome location (a location to be decremented or incremented when this location is addressed indirectly).

AIDCR indicates the state of BUS 12, i.e. memory location 30-37 (Oct) are addressed.

-,AIADR indicates that the internal bus (-,BUS 0-15) contains an address in the range 20-27 (Oct) or 30-37 (Oct).

SELECT indicates whether the CPU721 is supplied with 64K bytes memory or with 128K bytes memory.

6.1.12 Condition Group 5, Bank 1 6.1.12

Not used.

6.1.13 Condition Group 6, Bank 0 6.1.13

The Condition Group 6 consists of the Instruction Register bit 8-9 and bit 12-14 (IR 8-9 and IR 12-14).

This Condition Group is used during arithmetic and logical instructions to determine whether the result should be shifted or not (IR 8-9) and to specify the test of the result (IR 12-14).

IR8	IR9	SHIFT FUNCTION
0	0	NO SHIFT
0	1	SHIFT LEFT
1	0	SHIFT RIGHT
1	1	BYTE SWAP

IR13	IR14	SKIP FUNCTION
0	0	SKIP/NO SKIP
0	1	SKIN ON CARRY
1	0	SKIP ON RESULT
1	1	SKIP ON BOTH CARRY AND/OR RESULT

IR12	LOAD FUNCTION
0	LOAD
1	NO LOAD

6.1.14 Condition Group 6, Bank 1 6.1.14

Not used.

6.1.15 Condition Group 7, Bank 0 6.1.15

The Condition Group 7 consists of the Instruction Register bit 1-4 (IR 1-4) and SELECT.

This Condition Group is used during execution of memory reference instructions after the address calculation to determine the instruction to be executed.

IR1	IR2	IR3	IR4	INSTRUCTION MNEMONIC
0	0	0	0	JMP
0	0	0	1	JSR
0	0	1	0	ISZ
0	0	1	1	DSZ
0	1	0	0	LDA
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	STA
1	0	0	1	
1	0	1	0	
1	0	1	1	

6.1.16 Condition Group 7, Bank 1

6.1.16

Not used.

			2 ND OP 0	2 ND OP 1	2 ND OP 2	2 ND OP 3	2 ND OP 4
0	0	0	IR 9	IR 8	IR 7	IR 6	IR 5
0	0	1	IR 7	IR 6	IR 5	-,DEV 77/xx	-,DEV 01/xx
0	1	0	-,FPOP0 B	-,FPOP 1 B	-,FPOP2B	-,FPOP BUSY B	APL B
0	1	1	-,DCHM 0 B	-,DCHM 1 B	BREAK COND	-,SKIP B	EN BREAK
1	0	0	-,SKIP B	-,INTR SYNC	INTEN DLY'D	-,DCHR SYNC	RUN B
1	0	1	-,AIDCR	-,AI ADR	BUS 0	-,DC RST	SELECT
1	1	0	IR 14	IR 13	IR 12	IR 9	IR 8
1	1	1	IR 4	IR 3	IR 2	IR 1	SELECT

Figure 20: Condition Selector.

6.2 Next Address Maps

The NEXT ADDRESS MAPS consist of three PROM memory:

1) 1K x 8 words consisting next address during fetch and conditional jump.

2) 512 x 8 words consisting next address during unconditional jump.

3) 32 x 8 words consisting next address during power restart.

6.2.1 Conditional Jump Map

The capacity of this PROM memory is 1K x 8 words. This memory is used as a look up table for the next microaddress during fetching an instruction and during conditional jump.

The address input (SEL 0-7), SEL 1ST OP) to this memory is calculated as shown in fig. 21.

SEL 1ST OP	SEL 0	1	2	3	4	5	6	7	FUNCTION
0	NXAD SEL C	NXAD SEL B	NXAD SEL A	2ND OP 4	2ND OP 3	2ND OP 2	2ND OP 1	2ND OP 0	Conditional jump
1	BUS 0	BUS 1	BUS 2	BUS 3	BUS 4	BUS 5	BUS 6	BUS 7	FETCH

Figure 21: Conditional jump and fetch address input.

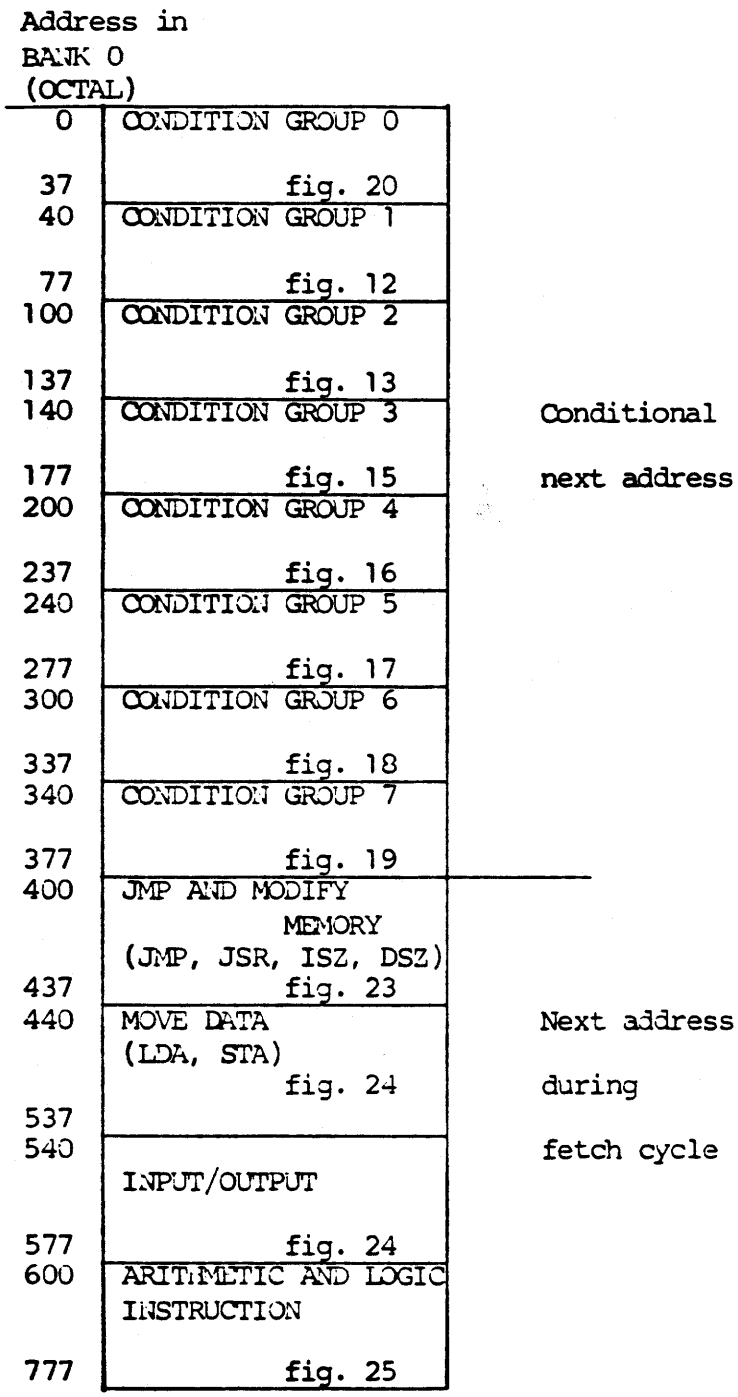


Figure 22: Next Address MAP Layout in BANK 0.

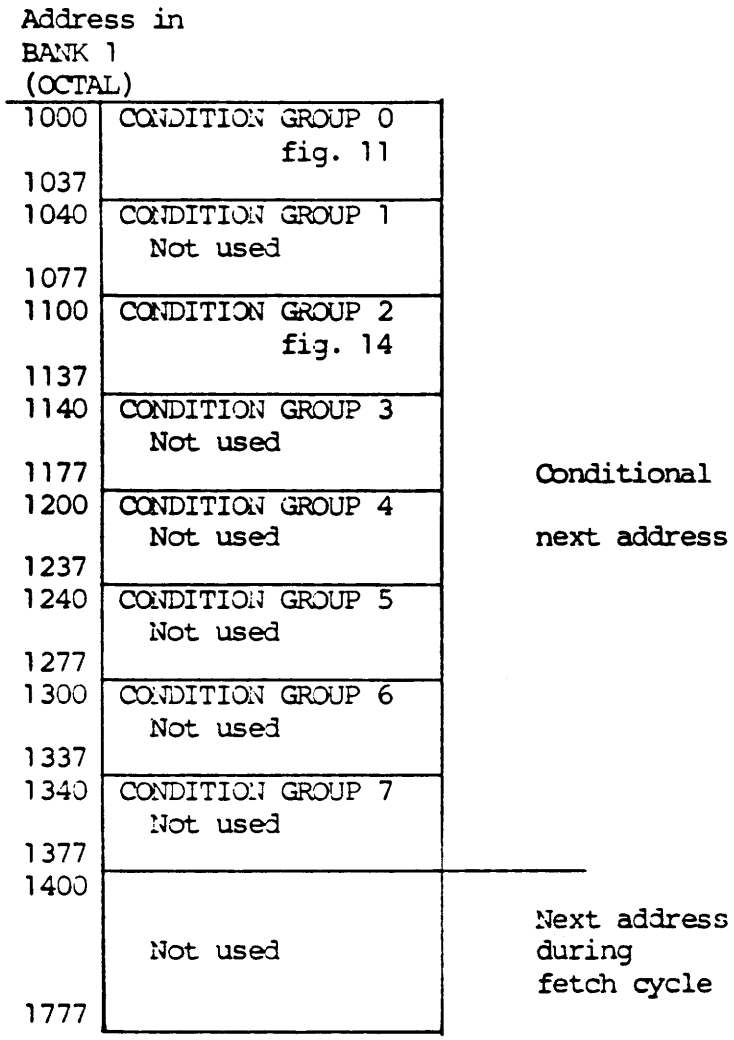


Figure 22: Next Address Map Layout in BANK 1.

NEXT ADDR MAP DURING FETCH OF AN
JMP AND MODIFY MEMORY INSTRUCTION
ADDR 400 - 437 IN ROB 382-383

BASE				FUN C		1	INDEX		NEXT	INSTR	ADDR
ADDR	BUS0	BUS1	BUS2	BUS3	BUS4	BUS5	BUS6	BUS7	ADDR	MNEMONI	CALC
1	0	0	0	0	0	0	0	0	2		Direct
1	0	0	0	0	0	0	0	1	3		Relative
1	0	0	0	0	0	0	1	0	4		Index (AC2)
1	0	0	0	0	0	0	1	1	5	JMP	Index (AC3)
1	0	0	0	0	0	1	0	0	12		Indirect
1	0	0	0	0	0	1	0	1	13		Indirect, Rel
1	0	0	0	0	0	1	1	0	14		Indirect, Index
1	0	0	0	0	0	1	1	1	15		Indirect, Index
1	0	0	0	0	1	0	0	0	6		Direct
1	0	0	0	0	1	0	0	1	7		Relative
1	0	0	0	0	1	0	1	0	10		Index (AC2)
1	0	0	0	0	1	0	1	1	11	JSR	Index (AC3)
1	0	0	0	0	1	1	0	0	12	JSR	Indirect
1	0	0	0	0	1	1	0	1	13		Indirect, Rel
1	0	0	0	0	1	1	1	0	14		Indirect Index
1	0	0	0	0	1	1	1	1	15		Indirect, Index
1	0	0	0	1	0	0	0	0	6		Direct
1	0	0	0	1	0	0	0	1	7		Relative
1	0	0	0	1	0	0	1	0	10		Index (AC2)
1	0	0	0	1	0	0	1	1	11	ISZ	Index (AC2)
1	0	0	0	1	0	1	0	0	12		Indirect
1	0	0	0	1	0	1	0	1	13		Indirect, Rel
1	0	0	0	1	0	1	1	0	14		Indirect, Index
1	0	0	0	1	0	1	1	1	15		Indirect, Index
1	0	0	0	1	1	0	0	0	6		Direct
1	0	0	0	1	1	0	0	1	7		Relative
1	0	0	0	1	1	0	1	0	10		Index (AC2)
1	0	0	0	1	1	0	1	1	11	DSZ	Index (AC3)
1	0	0	0	1	1	1	0	0	12		Indirect
1	0	0	0	1	1	1	0	1	13		Indirect, Rel
1	0	0	0	1	1	1	1	0	14		Indirect, Index
1	0	0	0	1	1	1	1	1	15		Indirect, Index

Figure 23: Start Address Jump and Modify Memory Instruction.

NEXT ADDR MAP DURING FETCH OF AN
MOVE DATA INSTRUCTION
ADDR 440 - 537 IN ROB 382-383

BASE		FUN C		AC		1	INDEX		NEXT	INSTR	ADDR
ADDR	BUS0	BUS1	BUS2	BUS3	BUS4	BUS5	BUS6	BUS7	ADDR	MNEMONI	CALC
1	0	0	1	x	x	0	0	0	6		Direct
1	0	0	1	x	x	0	0	1	7		Relative
1	0	0	1	x	x	0	1	0	10		Index (AC2)
1	0	0	1	x	x	0	1	1	11	LDA	Index (AC3)
1	0	0	1	x	x	1	0	0	12		Indirect
1	0	0	1	x	x	1	0	1	13		Indirect, Rel
1	0	0	1	x	x	1	1	0	14		Indirect, Index
1	0	0	1	x	x	1	1	1	15		Indirect, Index
1	0	1	0	x	x	0	0	0	6		Direct
1	0	1	0	x	x	0	0	1	7		Relative
1	0	1	0	x	x	0	1	0	10		Index (AC2)
1	0	1	0	x	x	0	1	1	11	STA	Index (AC3)
1	0	1	0	x	x	1	0	0	12		Indirect
1	0	1	0	x	x	1	0	1	13		Indirect, Rel
1	0	1	0	x	x	1	1	0	14		Indirect, Ind
1	0	1	0	x	x	1	1	1	15		Indirect, Inde

xx: 00 (AC0), 01 (AC1), 10 (AC2), 11 (AC3).

NEXT ADDR MAP DURING FETCH OF AN
INPUT/OUTPUT INSTRUCTION
ADDR 540 - 577 IN ROB 382-383

BASE				AC		XFER			NEXT	INSTR
ADDR	BUS0	BUS1	BUS2	BUS3	BUS4	BUS5	BUS6	BUS7	ADDR	MNEMONI
1	0	1	1	x	x	0	0	0	70	NIO
1	0	1	1	x	x	0	0	1	70	DIA
1	0	1	1	x	x	0	1	0	70	DOA
1	0	1	1	x	x	0	1	1	70	DIB
1	0	1	1	x	x	1	0	0	70	DOB
1	0	1	1	x	x	1	0	1	70	DIC
1	0	1	1	x	x	1	1	0	70	DOC
1	0	1	1	x	x	1	1	1	70	SKP

xx: 00 (AC0), 01 (AC1), 10 (AC1), 10 (AC2), 11 (AC3)

Figure 24: Start Address Move Data Instruction INPUT/OUTPUT Instruction.

NEXT ADD MAP DURING FETCH OF AN
ARITHMETIC AND LOGIC INSTRUCTION
ADDR 600 - 777 IN ROB 382-383

BASE ADDR	BUS0	SOURCE AC BUS1	SOURCE AC BUS2	DEST AC BUS3	DEST AC BUS4	FUNCTION BUS5	FUNCTION BUS6	FUNCTION BUS7	NEXT ADDR	INSTR MNEMONI
1	1	x	x	y	y	0	0	0	42	COM
1	1	x	x	y	y	0	0	1	43	NEG
1	1	x	x	y	y	0	1	0	44	MOV
1	1	x	x	y	y	0	1	1	45	INC
1	1	x	x	y	y	1	0	0	46	ADC
1	1	x	x	y	y	1	0	1	47	SUB
1	1	x	x	y	y	1	1	0	50	ADD
1	1	x	x	y	y	1	1	1	51	AND

xx: 00 (AC0), 01 (AC1), 10 (AC2), 11 (AC3)
yy: 00 (AC0), 01 (AC1), 10 (AC2), 11 (AC3)

Figure 25: Start Address Arithmetic and Logic Instruction.

NEXT ADD MAP DURING POWER UP
ADDR 0 - 3 IN PROM 650

BASE ADDR	-, LOCK	DIS PANEL	NEXT ADDR (OCTAL)	FUNCTION
0 0 0	0	0	210	JMP 0
	0	1	210	JMP 0
	1	0	210	JMP 0
	1	1	210	JMP 0
1	0	0	245	Restart in the
1	0	1	245	address selected
1	1	0	245	by the data-
1	1	1	245	switches

DIS PANEL = 1 if the ENABLE TCP switch is in the upper state.

-,LOCK = = 1 All times the Diagnostic Panel (TCP 701) is not supplied with a POWER/LOCK switch, else this signal follows the state of this switch.

RESTART is executed if the operator pushes AUTO with the Reset party Error switch in the Reset state. The program starts with a JMP to the address determined by Dataswitches 0, 10-15 on the front panel of the CPU.

If SW0 = 0 JMP to the address (SW 10-15).
If SW0 = 1 JMP @ address (SW 10-15)

Figure 26: Power up address map.

6.2.2 Unconditional Jump 6.2.2

Refer to section 5.1.

6.2.3 Power Restart Add 6.2.3

When the machine is powered up, this map contains the address of the first microinstruction to be executed. Refer to fig. 26.

6.3 Extended Conditional Jump 6.3

The MPLSB SEL 0-2 selects one of eight signals as the LSB (NXAD7) of the next microaddress (see fig. 27).

MPLSB SEL 0	1	2	NXAD7
0	0	0	NXADx7 (unconditional jump)
0	0	1	-,INTR SYNC
0	1	0	-,DCHR SYNC
0	1	1	ALU = 0
1	0	0	-,DEV 2
1	0	1	ALU COUT
1	1	0	ALU LOW
1	1	1	F3

Figure 27: Selection of least significant microaddress bit.

The eight signals are NXAD7, -,INTR SYNC, -,DCHR SYNC, ALU = 0, -,DEV 2, ALU COUT, ALU LOW and F3.

-,INTR SYNC and -,DCHR SYNC are I/O BUS interrupt- and DMA-request signals. ALU = 0 is the output from the status register (see chapter 7).

-,DEV 2 = 0 if the fetched instruction is an instruction from the extended instruction set of the RC3803 (BMOVE, WMOVE, SCHEL, SFREE, LINK REMEL, PLINK, FETCH, TKADD, TKVAL, COMP).

ALU COUT is the carry output from the Am2901.

ALU LOW = 1 if the data output from an ALU operation is zero.

7. CARRY/SHIFTER CIRCUIT 7.

On the block diagram, fig. 28 is shown the CARRY/SHIFTER CIRCUIT. Refer to logic diagrams CPU 016 and CPU 018.

7.1 Carry Logic 7.1

There is a block of logic which functions to set up the base value of the carry for an arithmetic operation, and to receive and store the carry results from that operation. The base value of the carry is set up without disturbing the previous carry, so that the previous carry value can be saved if an arithmetic instruction specifies "NO LOAD".

For details about arithmetic and logical instructions, refer to FC5. In the first microstep (RAR 42-51) the ALU function specified in the instruction is executed. The carry out from the ALU, BIT 0 and BIT 15 of the result is saved in a status register. The outputs from this register (COUT BFR, B0 BFR, B15 BFR) together with Instruction Register bit 8-12 (IR 8-12) and the output of the carry register (-,CARRY REG) are used as address inputs to a 512 x 4 PROM (ROB 380); this PROM gives the new carry input to the carry register, shift inputs to the Q shifter and carry information to the SKIP circuit (chapter 8).

The contents of this PROM (ROB 380) are shown on figs. 29-35.

In the last step of an arithmetic-logical instruction the value on -,NEW LINK is loaded into the carry register.

If "LOAD" is specified in the instruction, -,NEW LINK and -,CARRY GEN are both equal to the new carry calculated during instruction execution.

If "NO LOAD" is specified, -,NEW LINK is equal to the previous carry (-,CARRY REG), but -,CARRY GEN is the new calculated carry used to determine skip, if SKIP on Carry is specified.

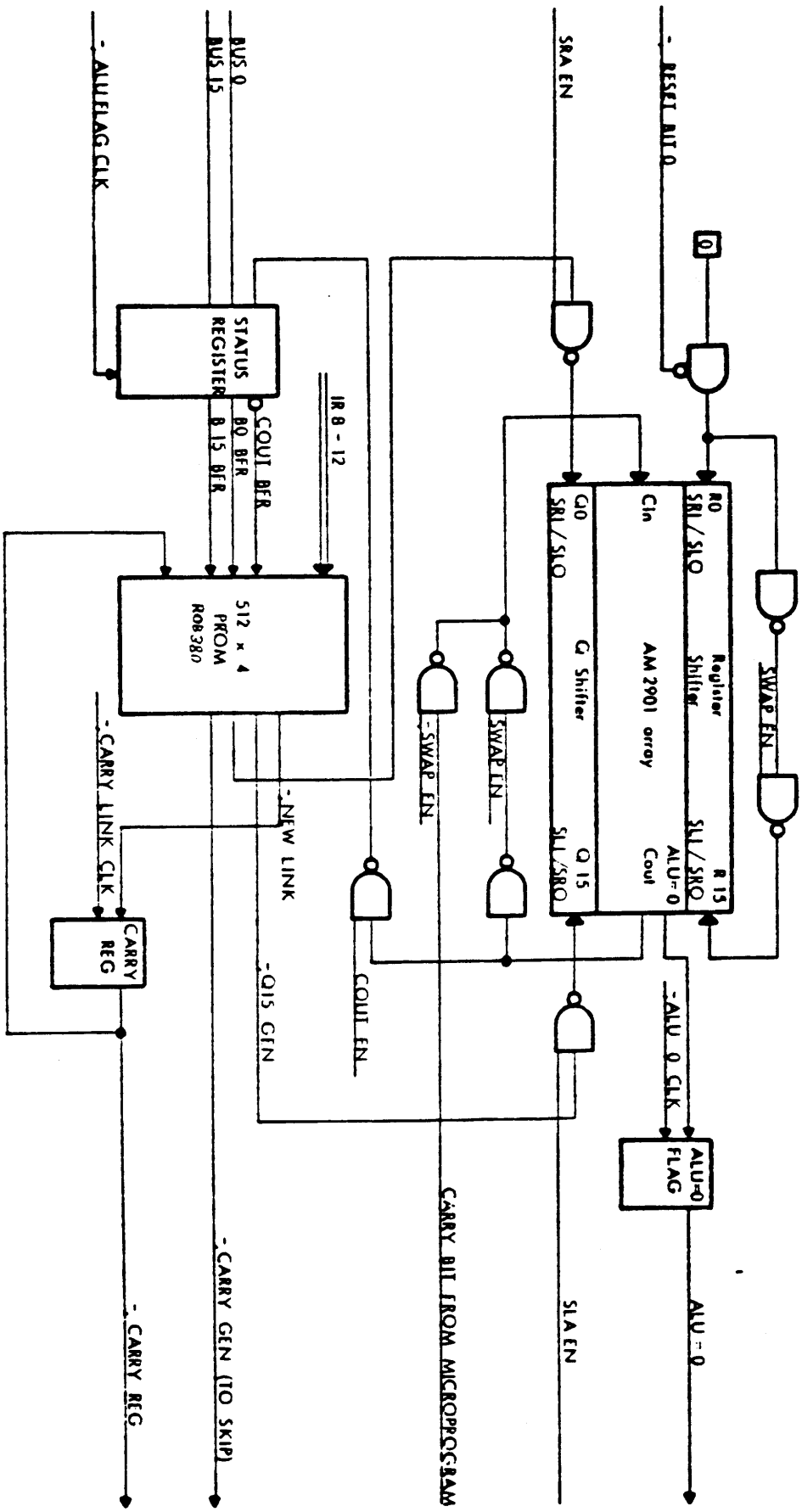


Figure 28: CARRY/SHIFTER Circuit.

NC LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-CARRY REG	OUTPUT FUNCTION				COMMENTS
									-,CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
IR12	IR8	IR9	IR10	IR11									
0	1	0	0	0	0	0	0	0	1	0	1	1	CURRENT
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	0	0	1	0	
			0	0	0	0	1	1	0	1	1	0	
			0	0	0	1	0	0	1	0	1	1	-,CARRY GEN =
			0	0	0	1	0	1	1	1	1	1	-,B15 BFR
			0	0	0	1	1	0	0	0	1	0	
			0	0	0	1	1	1	0	1	1	0	-,Q0 GEN =
			0	0	1	0	0	0	1	1	1	1	
			0	0	1	0	0	1	1	0	1	1	COUT + (-,CARRY REG)
			0	0	1	0	1	0	0	1	1	0	
			0	0	1	0	1	1	0	0	1	0	
			0	0	1	1	0	0	1	1	1	1	
			0	0	1	1	0	1	1	0	1	1	
			0	0	1	1	1	0	0	1	1	0	
			0	0	1	1	1	1	0	0	1	0	
			0	1	0	0	0	0	1	1	1	1	Z
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	0	1	1	0	
			0	1	0	0	1	1	0	1	1	0	-,CARRY GEN =
			0	1	0	1	0	0	1	1	1	1	-,B15 BFR
			0	1	0	1	0	1	1	1	1	1	
			0	1	0	1	1	0	0	1	1	0	
			0	1	0	1	1	1	0	1	1	0	-,Q0 GEN =
			0	1	1	0	0	0	1	0	1	1	
			0	1	1	0	0	1	1	0	1	1	COUT + 1
			0	1	1	0	1	0	0	0	1	0	
			0	1	1	0	1	1	0	0	1	0	
			0	1	1	1	0	0	1	0	1	1	
			0	1	1	1	0	1	1	0	1	1	
			0	1	1	1	1	0	0	0	1	0	
			0	1	1	1	1	1	0	0	1	0	
			1	0	0	0	0	0	1	0	1	1	O
			1	0	0	0	0	1	1	0	1	1	
			1	0	0	0	1	0	0	0	1	0	
			1	0	0	0	1	1	0	0	1	0	-,CARRY GEN =
			1	0	0	1	0	0	1	0	1	1	-,B15 BFR
			1	0	0	1	0	1	1	0	1	1	
			1	0	0	1	1	0	0	0	1	0	
			1	0	0	1	1	1	0	0	1	0	
			1	0	1	0	0	0	1	1	1	1	
			1	0	1	0	0	1	1	1	1	1	-,Q0 GEN =
			1	0	1	0	1	0	0	1	1	0	
			1	0	1	0	1	1	0	1	1	0	COUT + 0
			1	0	1	1	0	0	1	1	1	1	
			1	0	1	1	0	1	1	1	1	1	
			1	0	1	1	1	0	0	1	1	0	
			1	0	1	1	1	1	0	1	1	0	
			1	1	0	0	0	0	1	1	1	1	C
			1	1	0	0	0	1	1	0	1	1	
			1	1	0	0	1	0	0	1	1	0	
			1	1	0	0	1	1	0	0	1	0	
			1	1	0	1	0	0	1	1	1	1	-,CARRY GEN =
			1	1	0	1	0	1	1	0	1	1	-,B0 BFR
			1	1	0	1	1	0	0	1	1	0	
			1	1	0	1	1	1	0	0	1	0	
			1	1	1	0	0	0	1	0	1	1	
			1	1	1	0	0	1	1	1	1	1	
			1	1	1	0	1	0	0	0	1	0	-,Q0 GEN =
			1	1	1	0	1	1	0	0	1	0	
			1	1	1	1	0	0	1	0	1	1	COUT +
			1	1	1	1	0	1	1	1	1	1	
			1	1	1	1	1	0	0	0	1	0	; (-,CARRY REG)
			1	1	1	1	1	1	0	1	1	0	

Figure 29: CARRY and SHIFT PROM Contents LOAD, NO SHIFT.

NO LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-,CARRY REG	OUTPUT FUNCTION				COMMENTS
IR12	IR8	IR9	IR10	IR11					-,CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
0	0	0	0	0	0	0	0	0	0	1	1	0	
			0	0	0	0	0	1	1	1	1	1	Base value of Carry = Current State
			0	0	0	0	1	0	0	1	1	0	
			0	0	0	0	1	1	1	1	1	1	
			0	0	0	1	0	0	0	1	1	0	
			0	0	0	1	0	1	1	1	1	1	-,CARRY GEN =
			0	0	0	1	1	0	0	1	1	0	COUT BFR +
			0	0	0	1	1	1	1	1	1	1	-,CARRY REG
			0	0	1	0	0	0	1	1	1	1	
			0	0	1	0	0	1	0	1	1	0	
			0	0	1	0	1	0	1	1	1	1	
			0	0	1	0	1	1	0	1	1	0	
			0	0	1	1	0	0	1	1	1	1	
			0	0	1	1	0	1	0	1	1	0	
			0	0	1	1	1	0	1	1	1	1	
			0	0	1	1	1	1	0	1	1	0	
			0	1	0	0	0	0	1	1	1	1	
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	1	1	1	1	Z
			0	1	0	0	1	1	1	1	1	1	
			0	1	0	1	0	0	1	1	1	1	-,CARRY GEN =
			0	1	0	1	0	1	1	1	1	1	
			0	1	0	1	1	0	1	1	1	1	COUT BFR +
			0	1	0	1	1	1	1	1	1	1	
			0	1	1	0	0	0	0	1	1	0	
			0	1	1	0	0	1	0	1	1	0	
			0	1	1	0	1	0	0	1	1	0	
			0	1	1	0	1	1	0	1	1	0	
			0	1	1	1	0	0	0	1	1	0	
			0	1	1	1	0	1	0	1	1	0	
			0	1	1	1	1	0	0	1	1	0	
			0	1	1	1	1	1	0	1	1	0	
			1	0	0	0	0	0	0	1	1	0	
			1	0	0	0	0	1	0	1	1	0	
			1	0	0	0	1	0	0	1	1	0	
			1	0	0	0	1	1	0	1	1	0	
			1	0	0	1	0	0	0	1	1	0	
			1	0	0	1	0	1	0	1	1	0	
			1	0	0	1	1	0	0	1	1	0	
			1	0	0	1	1	1	0	1	1	0	
			1	0	1	0	0	0	1	1	1	1	
			1	0	1	0	0	1	1	1	1	1	
			1	0	1	0	1	0	1	1	1	1	O
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	1	1	1	1	-,CARRY GEN =
			1	0	1	1	0	1	1	1	1	1	
			1	0	1	1	1	0	1	1	1	1	COUT BFR + O
			1	0	1	1	1	1	1	1	1	1	
			1	1	0	0	0	0	1	1	1	1	
			1	1	0	0	0	1	0	1	1	0	
			1	1	0	0	1	0	1	1	1	1	
			1	1	0	0	1	1	0	1	1	0	C
			1	1	0	1	0	0	1	1	1	1	
			1	1	0	1	0	1	0	1	1	0	
			1	1	0	1	1	0	1	1	1	1	-,CARRY GEN =
			1	1	0	1	1	1	0	1	1	0	
			1	1	1	0	0	0	0	1	1	0	COUT BFR +
			1	1	1	0	0	1	0	1	1	0	
			1	1	1	0	1	0	1	1	1	1	-,(-,CARRY REG)
			1	1	1	0	1	1	0	1	1	0	
			1	1	1	1	0	0	0	1	1	0	
			1	1	1	1	0	1	1	1	1	1	
			1	1	1	1	1	0	0	1	1	0	
			1	1	1	1	1	1	1	1	1	1	

Figure 30: CARRY and SHIFT PROM Contents LOAD, SHIFT LEFT.

NO LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-,CARRY REG	OUTPUT FUNCTION				COMMENTS
									-,CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
	IR8	IR9	IR10	IR11									
	0	1	0	0	0	0	0	0	1	1	0	1	CURRENT
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	1	1	0	1	
			0	0	0	0	1	1	1	1	1	1	-,CARRY GEN =
			0	0	0	1	0	0	0	1	0	0	-,B0 BFR
			0	0	0	1	0	1	0	1	1	0	
			0	0	0	1	1	0	0	1	0	0	
			0	0	0	1	1	1	0	1	1	0	
			0	0	1	0	0	0	1	1	1	1	-,Q15 GEN =
			0	0	1	0	0	1	1	1	0	1	
			0	0	1	0	1	0	1	1	1	1	COUT + (-,CARRY REG)
			0	0	1	0	1	1	1	1	0	1	
			0	0	1	1	0	0	0	1	1	0	
			0	0	1	1	0	1	0	1	0	0	
			0	0	1	1	1	0	0	1	1	0	
			0	0	1	1	1	1	0	1	0	0	
			0	1	0	0	0	0	1	1	1	1	Z
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	1	1	1	1	
			0	1	0	0	1	1	1	1	1	1	-,CARRY GEN =
			0	1	0	1	0	0	0	1	1	0	-,B0 BFR
			0	1	0	1	0	1	0	1	1	0	
			0	1	0	1	1	0	0	1	1	0	
			0	1	0	1	1	1	0	1	1	0	-,Q15 GEN =
			0	1	1	0	0	0	1	1	0	1	
			0	1	1	0	0	1	1	1	0	1	COUT + 1
			0	1	1	0	1	0	1	1	0	1	
			0	1	1	0	1	1	1	1	0	1	
			0	1	1	1	0	0	0	1	0	0	
			0	1	1	1	0	1	0	1	0	0	
			0	1	1	1	1	0	0	1	0	0	
			0	1	1	1	1	1	0	1	0	0	
			1	0	0	0	0	0	1	1	1	1	O
			1	0	0	0	0	1	1	1	0	1	
			1	0	0	0	1	0	1	1	0	1	
			1	0	0	0	1	1	1	1	0	1	-,CARRY GEN =
			1	0	0	1	0	0	0	1	0	0	-,B0 BFR
			1	0	0	1	0	1	0	1	0	0	
			1	0	0	1	1	0	0	1	0	0	
			1	0	0	1	1	1	0	1	0	0	
			1	0	1	0	0	0	1	1	1	1	-,Q15 GEN =
			1	0	1	0	0	1	1	1	1	1	
			1	0	1	0	1	0	1	1	1	1	COUT + 0
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	0	1	1	0	
			1	0	1	1	0	1	0	1	1	0	
			1	0	1	1	1	0	0	1	1	0	
			1	0	1	1	1	1	0	1	1	0	
			1	1	0	0	0	0	1	1	1	1	C
			1	1	0	0	0	1	1	1	0	1	
			1	1	0	0	1	0	1	1	1	1	
			1	1	0	0	1	1	1	1	0	1	-,CARRY GEN =
			1	1	0	1	0	0	0	1	1	0	-,B0 BFR
			1	1	0	1	0	1	0	1	0	0	
			1	1	0	1	1	0	0	1	1	0	
			1	1	0	1	1	1	0	1	0	0	
			1	1	1	0	0	0	1	1	0	1	
			1	1	1	0	0	1	1	1	1	1	-,Q15 GEN =
			1	1	1	0	1	0	1	1	0	1	
			1	1	1	0	1	1	1	1	1	1	COUT +
			1	1	1	1	0	0	0	1	0	0	
			1	1	1	1	0	1	0	1	1	0	-,(-,CARRY REG)
			1	1	1	1	1	0	0	1	0	0	
			1	1	1	1	1	1	0	1	1	0	

Figure 31: CARRY and SHIFT PROM Contents LOAD, SHIFT RIGHT.

NO LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-,CARRY REG	OUTPUT FUNCTION				COMMENTS
									-CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
IR12	IR8	IR9	IR10	IR11									
0	1	1	0	0	0	0		0	0	1	1	0	Current
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	0	1	1	0	-,CARRY GEN =
			0	0	0	0	1	1	1	1	1	1	
			0	0	0	1	0	0	0	1	1	0	COUT + (-,CARRY REG)
			0	0	0	1	0	1	1	1	1	1	
			0	0	0	1	1	0	0	1	1	0	
			0	0	0	1	1	1	1	1	1	1	
			0	0	1	0	0	0	0	1	1	0	
			0	0	1	0	0	1	0	1	1	0	
			0	0	1	0	1	0	1	1	1	1	
			0	0	1	0	1	1	0	1	1	0	
			0	0	1	1	0	0	1	1	1	1	
			0	0	1	1	0	1	0	1	1	0	
			0	0	1	1	1	0	1	1	1	1	
			0	0	1	1	1	1	0	1	1	0	
			0	1	0	0	0	0	1	1	1	1	Z
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	1	1	1	1	
			0	1	0	0	1	1	1	1	1	1	
			0	1	0	1	0	0	1	1	1	1	-,CARRY GEN =
			0	1	0	1	0	1	1	1	1	1	
			0	1	0	1	1	0	1	1	1	1	COUT + 1
			0	1	0	1	1	1	1	1	1	1	
			0	1	1	0	0	0	0	1	1	0	
			0	1	1	0	0	1	0	1	1	0	
			0	1	1	0	1	0	0	1	1	0	
			0	1	1	0	1	1	0	1	1	0	
			0	1	1	1	0	0	0	1	1	0	
			0	1	1	1	0	1	0	1	1	0	
			0	1	1	1	1	0	0	1	1	0	
			0	1	1	1	1	1	0	1	1	0	
			1	0	0	0	0	0	0	1	1	0	O
			1	0	0	0	0	1	0	1	1	0	
			1	0	0	0	1	0	0	1	1	0	
			1	0	0	0	1	1	0	1	1	0	-,CARRY GEN =
			1	0	0	1	0	0	0	1	1	0	
			1	0	0	1	0	1	0	1	1	0	COUT + 0
			1	0	0	1	1	0	0	1	1	0	
			1	0	0	1	1	1	0	1	1	0	
			1	0	1	0	0	0	1	1	1	1	
			1	0	1	0	0	1	1	1	1	1	
			1	0	1	0	1	0	1	1	1	1	
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	1	1	1	1	
			1	0	1	1	0	1	1	1	1	1	
			1	0	1	1	1	0	1	1	1	1	
			1	0	1	1	1	1	1	1	1	1	
			1	1	0	0	0	0	1	1	1	1	C
			1	1	0	0	0	1	0	1	1	0	
			1	1	0	0	1	0	1	1	1	1	
			1	1	0	0	1	1	0	1	1	0	-,CARRY GEN =
			1	1	0	1	0	0	1	1	1	1	
			1	1	0	1	0	1	0	1	1	0	COUT +
			1	1	0	1	1	0	1	1	1	1	-,(-,CARRY REG)
			1	1	0	1	1	1	0	1	1	0	
			1	1	1	0	0	0	0	1	1	0	
			1	1	1	0	0	1	0	1	1	1	
			1	1	1	0	1	0	0	1	1	0	
			1	1	1	0	1	1	0	1	1	1	
			1	1	1	1	0	0	0	1	1	0	
			1	1	1	1	0	1	1	1	1	1	
			1	1	1	1	1	0	0	1	1	0	
			1	1	1	1	1	1	1	1	1	1	

Figure 32: CARRY and SHIFT PROM Contents LOAD, SWAP.

NO LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-CARRY REG	OUTPUT FUNCTION				COMMENTS
IR 12	IR8	IR9	IR10	IR11					-,CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
0	0	0	0	0	0	0	0	0	0	1	1	0	Current
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	0	1	1	0	
			0	0	0	0	1	1	1	1	1	1	-,CARRY GEN =
			0	0	0	1	0	0	0	1	1	0	COUT + CARRY REG
			0	0	0	1	0	1	1	1	1	1	
			0	0	0	1	1	0	0	1	1	0	
			0	0	0	1	1	1	1	1	1	1	
			0	0	1	0	0	0	1	1	1	0	-,NEW LINK =
			0	0	1	0	0	1	0	1	1	1	-,CARRY REG
			0	0	1	0	1	0	1	1	1	0	
			0	0	1	0	1	1	0	1	1	1	
			0	0	1	1	0	0	1	1	1	0	
			0	0	1	1	0	1	0	1	1	1	
			0	0	1	1	1	0	1	1	1	0	
			0	0	1	1	1	1	0	1	1	1	
			0	1	0	0	0	0	1	1	1	0	
			0	1	0	0	0	1	1	1	1	1	Z
			0	1	0	0	1	0	1	1	1	0	
			0	1	0	0	1	1	1	1	1	1	
			0	1	0	1	0	0	1	1	1	0	-,CARRY GEN =
			0	1	0	1	0	1	1	1	1	1	COUT + 1
			0	1	0	1	1	0	1	1	1	0	
			0	1	0	1	1	1	1	1	1	1	
			0	1	1	0	0	0	0	1	1	0	
			0	1	1	0	0	1	0	1	1	1	
			0	1	1	0	1	0	0	1	1	0	
			0	1	1	0	1	1	0	1	1	1	
			0	1	1	1	0	0	0	1	1	0	
			0	1	1	1	0	1	0	1	1	1	
			0	1	1	1	1	0	0	1	1	0	
			0	1	1	1	1	1	0	1	1	1	
			1	0	0	0	0	0	1	1	1	0	
			1	0	0	0	0	1	1	1	1	1	C
			1	0	0	0	1	0	1	1	1	0	
			1	0	0	0	1	1	1	1	1	1	-,CARRY GEN =
			1	0	0	1	0	0	1	1	1	0	COUT + 0
			1	0	0	1	0	1	1	1	1	1	
			1	0	0	1	1	0	1	1	1	0	
			1	0	0	1	1	1	1	1	1	1	
			1	0	1	0	0	0	1	1	1	0	
			1	0	1	0	0	1	1	1	1	1	
			1	0	1	0	1	0	1	1	1	0	
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	1	1	1	0	
			1	0	1	1	0	1	1	1	1	1	
			1	0	1	1	1	0	1	1	1	0	
			1	0	1	1	1	1	1	1	1	1	
			1	1	0	0	0	0	1	1	1	0	
			1	1	0	0	0	1	0	1	1	1	C
			1	1	0	0	1	0	1	1	1	0	
			1	1	0	0	1	1	0	1	1	1	
			1	1	0	1	0	0	1	1	1	0	
			1	1	0	1	0	1	0	1	1	1	
			1	1	0	1	1	0	1	1	1	0	
			1	1	0	1	1	1	0	1	1	1	-,CARRY GEN =
			1	1	1	0	0	0	0	1	1	0	COUT BFR +
			1	1	1	0	0	1	1	1	1	1	-,(-,CARRY REG)
			1	1	1	0	1	0	0	1	1	0	
			1	1	1	0	1	1	1	1	1	1	
			1	1	1	1	0	0	0	1	1	0	
			1	1	1	1	0	1	1	1	1	1	
			1	1	1	1	1	0	0	1	1	0	
			1	1	1	1	1	1	1	1	1	1	

Figure 33: CARRY and SHIFT PROM Contents NO LOAD, NO SHIFT.

NO LOAD	SHIFT		CARRY		COUT BFR	B0 BFR	B15 BFR	-,CARRY REG	OUTPUT FUNCTION				COMMENTS
									-,CARRY GEN	-,QC GEN	-,Q15 GEN	-,NEW LINK	
IR12	IR8	IR9	IR10	IR11									
1	0	1	0	0	0	0	0	0	1	1	0	0	CURRENT
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	1	1	0	0	
			0	0	0	0	1	1	1	1	1	1	
			0	0	0	1	0	0	0	1	0	0	
			0	0	0	1	0	1	0	1	1	1	-,CARRY GEN =
			0	0	0	1	1	0	0	1	0	0	-,B 0 BFR
			0	0	0	1	1	1	0	1	1	1	
			0	0	1	0	0	0	1	1	1	0	
			0	0	1	0	0	1	1	1	0	1	-,Q15 GEN =
			0	0	1	0	1	0	1	1	1	0	COUT +-,CARRY REG
			0	0	1	0	1	1	1	1	0	1	
			0	0	1	1	0	0	0	1	1	0	
			0	0	1	1	0	1	0	1	0	1	
			0	0	1	1	1	0	0	1	1	0	
			0	0	1	1	1	1	0	1	0	1	
			0	1	0	0	0	0	1	1	1	0	Z
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	1	1	1	0	
			0	1	0	0	1	1	1	1	1	1	
			0	1	0	1	0	0	0	1	1	0	-,CARRY GEN =
			0	1	0	1	0	1	0	1	1	1	-,B0 BFR
			0	1	0	1	1	0	0	1	1	0	
			0	1	0	1	1	1	0	1	1	1	
			0	1	1	0	0	0	1	1	0	0	-,Q15 GEN =
			0	1	1	0	0	1	1	1	0	1	
			0	1	1	0	1	0	1	1	0	0	COUT + 1
			0	1	1	0	1	1	1	1	0	1	
			0	1	1	1	0	0	0	1	0	0	
			0	1	1	1	0	1	0	1	0	1	
			0	1	1	1	1	0	0	1	0	0	
			0	1	1	1	1	1	0	1	0	1	
			1	0	0	0	0	0	1	1	0	0	O
			1	0	0	0	0	1	1	1	0	1	
			1	0	0	0	1	0	1	1	0	0	
			1	0	0	0	1	1	1	1	0	1	
			1	0	0	1	0	0	0	1	0	0	-,CARRY GEN =
			1	0	0	1	0	1	0	1	0	1	-,B0 BFR
			1	0	0	1	1	0	0	1	0	0	
			1	0	0	1	1	1	0	1	0	1	
			1	0	1	0	0	0	1	1	1	0	-,Q15 GEN =
			1	0	1	0	0	1	1	1	1	1	COUT + 0
			1	0	1	0	1	0	1	1	1	0	
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	0	1	1	0	
			1	0	1	1	0	1	0	1	1	1	
			1	0	1	1	1	0	0	1	1	0	
			1	0	1	1	1	1	0	1	1	1	
			1	1	0	0	0	0	1	1	1	0	C
			1	1	0	0	0	1	1	1	0	1	
			1	1	0	0	1	0	1	1	1	0	
			1	1	0	0	1	1	1	1	0	1	
			1	1	0	1	0	0	0	1	1	0	
			1	1	0	1	0	1	0	1	0	1	-,CARRY GEN =
			1	1	0	1	1	0	0	1	1	0	-,B0 BFR
			1	1	0	1	1	1	0	1	0	1	
			1	1	1	0	0	0	1	1	0	0	
			1	1	1	0	0	1	1	1	1	1	
			1	1	1	0	1	0	1	1	0	0	-,Q15 GEN =
			1	1	1	0	1	1	1	1	1	1	COUT +
			1	1	1	1	0	0	0	1	0	0	-,(-,CARRY REG)
			1	1	1	1	0	1	0	1	1	1	
			1	1	1	1	1	0	0	1	0	0	
			1	1	1	1	1	1	0	1	1	1	

Figure 34: CARRY and SHIFT PROM Contents NO LOAD, SHIFT LEFT.

NO LOAD	SHIFT		CARRY		COUT BFR	B 0 BFR	B 15 BFR	-CARRY REG	OUTPUT FUNCTION				COMMENTS
									-CARRY GEN	-,Q0 GEN	-,Q15 GEN	-,NEW LINK	
IR12	IR8	IR9	IR10	IR11									
1	1	1	0	0	0	0	0	0	0	1	1	0	CURRENT
			0	0	0	0	0	1	1	1	1	1	
			0	0	0	0	1	0	0	1	1	0	-,CARRY GEN =
			0	0	0	0	1	1	1	1	1	1	COUT + (-,CARRY REG)
			0	0	0	1	0	0	0	1	1	0	
			0	0	0	1	0	1	1	1	1	1	
			0	0	0	1	1	0	0	1	1	0	
			0	0	0	1	1	1	1	1	1	1	
			0	0	1	0	0	0	0	1	1	0	
			0	0	1	0	0	1	0	1	1	0	
			0	0	1	0	1	0	1	1	1	0	
			0	0	1	0	1	1	0	1	1	1	
			0	0	1	1	0	0	1	1	1	0	
			0	0	1	1	0	1	0	1	1	1	
			0	0	1	1	1	0	1	1	1	0	
			0	0	1	1	1	1	0	1	1	1	
			0	1	0	0	0	0	1	1	1	0	Z
			0	1	0	0	0	1	1	1	1	1	
			0	1	0	0	1	0	1	1	1	0	
			0	1	0	0	1	1	1	1	1	1	-,CARRY GEN =
			0	1	0	1	0	0	1	1	1	0	COUT + 1
			0	1	0	1	0	1	1	1	1	1	
			0	1	0	1	1	0	1	1	1	0	
			0	1	0	1	1	1	1	1	1	1	
			0	1	1	0	0	0	0	1	1	0	
			0	1	1	0	0	1	0	1	1	1	
			0	1	1	0	1	0	0	1	1	0	
			0	1	1	0	1	1	0	1	1	1	
			0	1	1	1	0	0	0	1	1	0	
			0	1	1	1	0	1	0	1	1	1	
			0	1	1	1	1	0	0	1	1	0	
			0	1	1	1	1	1	0	1	1	1	
			1	0	0	0	0	0	0	1	1	0	C
			1	0	0	0	0	1	0	1	1	1	-,CARRY GEN =
			1	0	0	0	1	0	0	1	1	0	COUT + 0
			1	0	0	0	1	1	0	1	1	1	
			1	0	0	1	0	0	0	1	1	0	
			1	0	0	1	0	1	0	1	1	1	
			1	0	0	1	1	0	0	1	1	0	
			1	0	0	1	1	1	0	1	1	1	
			1	0	1	0	0	0	1	1	1	0	
			1	0	1	0	0	1	1	1	1	1	
			1	0	1	0	1	0	1	1	1	0	
			1	0	1	0	1	1	1	1	1	1	
			1	0	1	1	0	0	1	1	1	0	
			1	0	1	1	0	1	1	1	1	1	
			1	0	1	1	1	0	1	1	1	0	
			1	0	1	1	1	1	1	1	1	1	
			1	1	0	0	0	0	1	1	1	0	C
			1	1	0	0	0	1	0	1	1	1	
			1	1	0	0	1	0	1	1	1	0	
			1	1	0	0	1	1	0	1	1	1	
			1	1	0	1	0	0	1	1	1	0	-,CARRY GEN =
			1	1	0	1	0	1	0	1	1	1	COUT +
			1	1	0	1	1	0	1	1	1	0	-,(-,CARRY REG)
			1	1	0	1	1	1	0	1	1	1	
			1	1	1	0	0	0	0	1	1	0	
			1	1	1	0	0	1	1	1	1	1	
			1	1	1	0	1	0	0	1	1	0	
			1	1	1	0	1	1	1	1	1	1	
			1	1	1	1	0	0	0	1	1	0	
			1	1	1	1	0	1	1	1	1	1	
			1	1	1	1	1	0	0	1	1	0	
			1	1	1	1	1	1	1	1	1	1	

Figure 35: CARRY and SHIFT PROM Contents NO LOAD, SWAP.

7.2 Shifter Logic

The SHIFTER consists of a block of logic gicing inputs to the Register and Q shifter in the AM 2901 array (refer to chapter 3).

7.2.1 Q-Shifter

The inputs to the Q-shifter, Q0 SRI/SLO and Q15 SLI/SRO are delivered from the PROM (ROB 380) mentioned in section 7.1. -,Q0 GEN is gated out on Q0 SRI/SLO by the microprogram do-function SRAEN (Shift Right ENable).

-,Q15 GEN is gated out on Q15 SLI/SRO by the microprogram do-function SLAEN (Shift Left ENable).

The Q-shifter is used during executing an arithmetic-logical instruction to perform the shift operation (one place Left or Right) specified in the instruction field.

For details about the value of the outputs from ROB 380 (-,Q0 GEN and -,Q15 GEN) refer to figs. 29-35.

7.2.2 Register Shifter

The Register Shifter is used to perform the SWAP operation perhaps specified in an arithmetic-logical instruction and to "CLIP" PC when executing jump to subroutine (JSR) and before entering the BREAK routine.

During SWAP operation R0 SRI/SLO is looped back to R15 SLI/SRO, and COUT back to CIN. This is necessary because the SWAP routine contains 4 steps, each performs a 2 steps left shift. This is done by using AC15 temporarily, performing the function AC15 + AC15 and shifting the result before storing back into AC15.

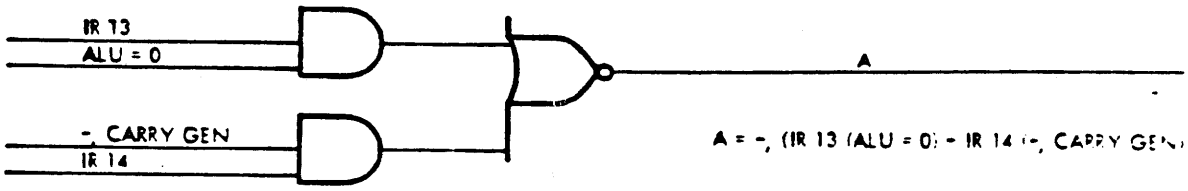
The "clip" operation of PC is done in two steps. One step to shift PC left and then shift PC right with zero into R0 SRI/SLO, enabled by the microprogram do-function -,RESET BIT 0.

8. SKIP CONDITION DETECTOR 8.

Refer to Logic Diagrams CPU 018 and CPU 033.

The Skip Detector is used to detect SKIP condition when executing arithmetic-logical instructions, ISZ-DSZ instructions, and I/O SKIP instructions.

Refer to figs. 36-38 for details about the SKIP detector.



ARITHMETIC & LOGICAL SKIP TABLE				A
IR 13	IR 14	IR 15	SKIP FUNC	
0	0	0	DON'T SKIP	1
0	0	1	SKIP	1
0	1	0	SKIP IF C = 0	CARRY GEN
0	1	1	SKIP IF C = 1	CARRY GEN
1	0	0	SKIP IF Z = 0	-, ALU = 0
1	0	1	SKIP IF Z = 1	-, ALU = 0
1	1	0	SK. IF C=0 or Z=0	-, (ALU = 0 + -, CARRY GEN)
1	1	1	SK. IF C=1 and Z=1	-, (ALU = 0 + -, CARRY GEN)

(-, (ALU = 0)) x CARRY GEN

I/O SKIP TABLE			- I/O SKP
IR 8	IR 9	SKIP FUNC	
0	0	SKIP IF BUSY ≠ 0	-, SEL B
0	1	SKIP IF BUSY = 0	-, SEL B
1	0	SKIP IF DONE ≠ 0	-, SEL D
1	1	SKIP IF DONE = 0	-, SEL D

-, I/O SKP = IR 8 x (-, SEL D) + (-, IR 8) x (-, SEL B)

Figure 36: Arithmetic and Logical and I/O Skip Tables.

ISZ, DSZ, I/C, ARITH & LOGIC				SKIP INPUT				
IR 0	IR 1	IR 4	IR 15	ALU=0	I/C SKP	A	-, SKIP	COMMENTS
0	0	0	0	0	0	0	1	
0	0	0	0	0	0	1	1	-, SKIP = (-, ALU = 0)
0	0	0	0	0	1	0	1	
0	0	0	0	0	1	1	1	
0	0	0	0	1	0	0	0	
0	0	0	0	1	0	1	0	ISZ, DSZ
0	0	0	0	1	1	0	0	
0	0	0	0	1	1	1	0	INCREMENT (DECREMENT) AND
0	0	0	1	0	0	0	1	SKIP IF ZERO
0	0	0	1	0	0	1	1	
0	0	0	1	0	1	0	1	
0	0	0	1	0	1	1	1	
0	0	0	1	1	0	0	0	
0	0	0	1	1	0	1	0	
0	0	0	1	1	1	0	0	
0	0	0	1	1	1	1	0	
0	0	1	0	0	0	0	1	
0	0	1	0	0	0	1	1	
0	0	1	0	0	1	0	1	
0	0	1	0	0	1	1	1	
0	0	1	0	1	0	0	0	
0	0	1	0	1	0	1	0	
0	0	1	0	1	1	0	0	
0	0	1	0	1	1	1	0	
0	0	1	1	0	0	0	1	
0	0	1	1	0	0	1	1	
0	0	1	1	0	1	0	1	
0	0	1	1	0	1	1	1	
0	0	1	1	1	0	0	0	
0	0	1	1	1	0	1	0	
0	0	1	1	1	1	0	0	
0	0	1	1	1	1	1	0	
0	1	0	0	0	0	0	1	
0	1	0	0	0	0	1	1	
0	1	0	0	0	1	0	0	SKP BN, DN
0	1	0	0	0	1	1	0	
0	1	0	0	1	0	0	1	
0	1	0	0	1	0	1	1	
0	1	0	0	1	1	0	0	
0	1	0	0	1	1	1	0	
0	1	0	1	0	0	0	1	
0	1	0	1	0	0	1	1	I/O SKIP IF DONE # 0
0	1	0	1	0	1	0	0	I/O SKIP IF BUSY # 0
0	1	0	1	0	1	1	0	
0	1	0	1	1	0	0	1	
0	1	0	1	1	0	1	1	
0	1	0	1	1	1	0	0	
0	1	0	1	1	1	1	0	
0	1	1	0	0	0	0	0	
0	1	1	0	0	0	1	0	
0	1	1	0	0	1	0	1	SKP BZ, DZ
0	1	1	0	0	1	1	1	
0	1	1	0	1	0	0	0	
0	1	1	0	1	0	1	0	
0	1	1	0	1	1	0	1	I/O SKIP IF DONE = 0
0	1	1	0	1	1	1	1	I/O SKIP IF BUSY = 0
0	1	1	1	0	0	0	0	
0	1	1	1	0	0	1	0	
0	1	1	1	0	1	0	1	
0	1	1	1	0	1	1	1	
0	1	1	1	1	0	0	0	
0	1	1	1	1	0	1	0	
0	1	1	1	1	1	0	1	
0	1	1	1	1	1	1	1	

760814 KNEH 761028 IGH 761102LLM

Figure 37: Skip PROM Contents, Part I of II.

ISZ, DSZ, I/C, ARITH & LOGIC				SKIP INPUT				
IR 0	IR 1	IR 9	IR 15	ALU=0	I/C SKP	A	-, SKIP	COMMENTS
1	0	0	0	0	0	0	0	COM, NEG, MOV, INC, ADC, SUB, ADD, AND -, SKIP = (IR 15) ⋅ (-, A) + (-, IR 15) ⋅ A IR 15 = 0 SKIP IF ZERO IR 15 = 1 SKIP IF NONZERO
1	0	0	0	0	0	1	1	
1	0	0	0	0	1	0	0	
1	0	0	0	0	1	1	1	
1	0	0	0	1	0	0	0	
1	0	0	0	1	0	1	1	
1	0	0	0	1	1	0	0	
1	0	0	0	1	1	1	1	
1	0	0	1	0	0	0	1	
1	0	0	1	0	0	1	0	
1	0	0	1	0	1	0	1	
1	0	0	1	0	1	1	0	
1	0	0	1	1	0	0	1	
1	0	0	1	1	0	1	0	
1	0	0	1	1	1	0	1	
1	0	0	1	1	1	1	0	
1	0	1	0	0	0	0	0	
1	0	1	0	0	0	1	1	
1	0	1	0	0	1	0	0	
1	0	1	0	0	1	1	1	
1	0	1	0	1	0	0	0	
1	0	1	0	1	0	1	1	
1	0	1	0	1	1	0	0	
1	0	1	0	1	1	1	1	
1	0	1	1	0	0	0	1	
1	0	1	1	0	0	1	0	
1	0	1	1	0	1	0	1	
1	0	1	1	0	1	1	0	
1	0	1	1	1	0	0	1	
1	0	1	1	1	0	1	0	
1	0	1	1	1	1	0	1	
1	0	1	1	1	1	1	0	
1	0	0	0	0	0	0	0	
1	0	0	0	0	0	1	1	
1	0	0	0	0	1	0	0	
1	0	0	0	0	1	1	1	
1	0	0	0	1	0	0	0	
1	0	0	0	1	0	1	1	
1	0	0	0	1	1	0	0	
1	0	0	0	1	1	1	1	
1	0	0	1	0	0	0	1	
1	0	0	1	0	0	1	0	
1	0	0	1	0	1	0	1	
1	0	0	1	0	1	1	0	
1	0	0	1	1	0	0	1	
1	0	0	1	1	0	1	0	
1	0	0	1	1	1	0	1	
1	0	0	1	1	1	1	0	
1	0	1	0	0	0	0	0	
1	0	1	0	0	0	1	1	
1	0	1	0	0	1	0	0	
1	0	1	0	0	1	1	1	
1	0	1	0	1	0	0	0	
1	0	1	0	1	0	1	1	
1	0	1	0	1	1	0	0	
1	0	1	0	1	1	1	1	
1	0	1	1	0	0	0	1	
1	0	1	1	0	0	1	0	
1	0	1	1	0	1	0	1	
1	0	1	1	0	1	1	0	
1	0	1	1	1	0	0	1	
1	0	1	1	1	0	1	0	
1	0	1	1	1	1	0	1	
1	0	1	1	1	1	1	0	
							-, SKP EN = 1	

Figure 38: Skip PROM Contents, Part II of II.

9. MEMORY INTERFACE 9.

For details refer to Logic Diagrams CPU 006, 019 and 020 and Timing Diagram TC1.

The memory interface contains Memory Address Register (MAR), Parity Generator and Checker and Control Logic.

9.1 Memory Address Register 9.1

The Memory Address Register is a 16 bits register loaded from the internal bus (BUS 0-15) by the microprogram do-function -,LD MAR. The most significant address bit, A15, is normal logic "0", but in Memory Extension mode this address bit is loaded from BUS 0.

All addresses are present min. 50 ns. prior to MEM START.

9.2 Parity Generator and Checker 9.2

During Memory Write Cycles 2 parity bits, one per byte (ODD parity), are generated and written into memory together with the 16 data bits.

During Memory Read Cycles 16 data bits + 2 parity bits are read out from memory. The 2 parity bits are checked, and if a parity is detected an error flip-flop is set to indicate the error. There is one error flip-flop per parity bit.

The outputs from the error flip-flops are used to drive indicators on the front panel of the CPU and to stop the clock generator if the "STOP on Error" switch on the front panel is in the STOP state. The error flip-flops are reset by the RESET switch on the front panel or by power reset.

9.3 Control Logic

A memory cycle is started by MEM START generated by READ/WRITE. -,MEMORY BUSY is sent from the memory as response to MEM START to indicate when the current cycle has been completed and the next cycle can be initiated.

During read cycles the clock generator is stopped from MEM START and until data is available from memory indicated by DATA AVAIL.

During write cycles the clock generator is stopped from MEM START to the memory is ready again (-,MEMORY BUSY = 1). Before the microprogram starts a memory cycle the state of -,MEMORY BUSY is tested by the microprogram do-function -,TEST MEM.

Between memory cycles, the memory system is prevented from operating by the low state of the signal DATA RETAIN.

9.4 Memory Data Bus

The internal Bus (-,BUS 0-15) is connected directly to the memory. The signal READ/RESTORE controls the direction of the data flow between CPU and memory; when this signal is in high level, a Read cycle is in progress.

10. TCP 701 INTERFACE 10.

This interface contains all necessary logic circuits required to interconnect the CPU and the Diagnostic Panel TCP 701.

All these logic circuits are further described in the following sections.

10.1 Panel Bus 10.1

Refer to Logic Diagrams CPU 034 and CPU 036.

The Panel BUS, -,PAN BUS 0-15, is a 16 bits data bus used to transfer data between CPU and Diagnostic Panel.

If -,RESTART ADD = 0 (START), -,READS = 0 (during execution of the READS instruction) or -,GATE DATA SW = 0 (during the Panel functions R dep, Dep and Ex) the signal -,CON DATA causes the Diagnostic Panel to place the contents of the Data Switches on the Panel Bus. In all other situations (-,CON DATA = 1) the contents of the Data Bus latch are transferred from CPU to Diagnostic Panel to be shown on the Data Lights.

10.2 Function Switch Decoder 10.2

Refer to Logic Diagrams CPU 035 and 037 and Timing Diagrams TC6, TC7 and TC8.

From the Diagnostic Panel is received a 4 bits value (-,CON 0-2, -,CPU MODE) to indicate which switch the operator has pushed. The STROBE signal indicates when -,CON 0-2, -,CPU MODE are valid.

On fig. 39 is shown the relation between -,CON 0-2, -,CPU MODE and the Function Switches. -,CPU MODE = 0 indicates a switch function which requests microprogram service (Exam MEM, Exam Next, Dep, Dep Next, Exam Reg, Dep Reg and START). All other switch functions (except AUTO) will not request microprogram access.

	-,CON			-,CPU		-,FPOP		
FUNCTION	0	1	2	MODE	X	0	1	2
RESET	1	1	1	1	1			
STOP	0	1	1	1	1			
NOT USED	1	0	1	1	1			
CONT	0	0	1	1	1			
INSTR. STEP	1	1	0	1	1			
MICRO STEP	0	1	0	1	1			
NOT USED	1	0	0	1	1			
PROG LOAD	0	0	0	1	1			
START	1	1	1	0	1	1	1	1
EXAMINE REGISTER	0	1	1	0	1	0	1	1
DEPOSIT REGISTER	1	0	1	0	1	1	0	1
EXAM MEMORY	0	0	1	0	1	0	0	1
EXAM NEXT	1	1	0	0	1	1	1	0
DEPOSIT	0	1	0	0	1	0	1	0
DEPOSIT MEMORY NEXT	1	0	0	0	1	1	0	0
NOT USED	0	0	0	0	1	0	0	0

Figure 39: Diagnostic Panel Functions.

DIAGNOSTIC PANEL FUNCTION DECODER
PROM 616

ADDR INPUTS					OUTPUT SIGNALS								FUNCTION
-, FPCP 0	-, FPOP 1	-, FPOP 2	-, CPU MODE	x	-SET DC RST	-SET CLR RAR	-, PL TCP	-CPU FUNC	-, INSTR STEP	-, MI CRO STEP	-SET RUN	-CLR RUN	
0	0	0	0	0	0	1	1	1	1	1	1	0	
0	0	0	0	1	0	1	1	1	1	1	1	0	
0	0	0	1	0	1	1	1	1	1	1	1	1	
0	0	0	1	1	1	1	0	1	1	1	1	1	AUTO LOAD
0	0	1	0	0	1	1	1	1	1	1	1	1	
0	0	1	0	1	1	1	1	0	1	1	1	0	EXAM MEM
0	0	1	1	0	1	1	1	1	1	1	1	1	
0	0	1	1	1	1	1	1	1	1	1	0	1	CONTINUE
0	1	0	0	0	1	1	1	1	1	1	1	1	
0	1	0	0	1	1	1	1	0	1	1	1	0	DEP MEM
0	1	0	1	0	1	1	1	1	1	1	1	1	
0	1	0	1	1	1	1	1	1	1	0	1	1	MICRO STEP
0	1	1	0	0	1	1	1	1	1	1	1	1	
0	1	1	0	1	1	1	1	0	1	1	1	0	EXAM REG
0	1	1	1	0	1	1	1	1	1	1	1	1	
0	1	1	1	1	1	1	1	1	1	1	1	0	STOP
1	0	0	0	0	1	1	1	1	1	1	1	1	
1	0	0	0	1	1	1	1	0	1	1	1	0	DEP NEXT
1	0	0	1	0	1	1	1	1	1	1	1	1	
1	0	0	1	1	1	1	1	1	1	1	1	1	
1	0	1	0	0	1	1	1	1	1	1	1	1	
1	0	1	0	1	1	1	1	0	1	1	1	0	DEP REG
1	0	1	1	0	1	1	1	1	1	1	1	1	
1	0	1	1	1	1	1	1	1	1	1	1	1	
1	1	0	0	0	1	1	1	1	1	1	1	1	
1	1	0	0	1	1	1	1	0	1	1	1	0	EXAM NEXT
1	1	0	1	0	1	1	1	1	1	1	1	1	
1	1	0	1	1	1	1	1	1	0	1	1	0	INSTR STEP
1	1	1	0	0	1	1	1	1	1	1	1	1	
1	1	1	0	1	1	1	1	0	1	1	1	1	START
1	1	1	1	0	1	1	1	1	1	1	1	1	
1	1	1	1	1	0	1	1	1	1	1	1	0	RESET

x : Logic Diagram CPU 035, IC 23 - 10.

Figure 40: Diagnostic panel function decoder PROM.

-,CON 0-2 buffered (-,FPOP 0-2) is used by the microprogram as conditions, and is also decoded in ROM 616 (refer to fig. 40) to control the RUN flip-flop, AUTO flip-flop, MICRO STEP Logic and INSTR. STEP Logic.

The ACcumulator Select switches (ACSEL 0-3) are gated out on -,BUS 1-4 during the front panel microprogram routine and loaded into the instruction register bit 1-4, IR 1-4.

10.3 Run Flip-Flop 10.3

Refer to Logic Diagram CPU 037.

The state of this flip-flop forces the microprogram to go on with program execution (RUN = 1) or enter the front panel service routine (RUN = 0). The RUN flip-flop is set by -,RESTART ADDRESS = 0 (START function), -,GATE APL ADD = 0 (Autoload) and -,SET RUN = 0 (from the Diagnostic Panel decoder, when pushing CONTINUE). The RUN flip-flop is reset by -,HALT = 0 (executing the HALT instruction) and -,CLR RUN = 0 (from the Diagnostic Panel, when pushing RESET or STOP).

10.4 Micro Addr. Drivers 10.4

Refer to Logic Diagram CPU 038.

These drivers transfer the microprogram address to the Diagnostic Panel, -,PAN RAR 7-0. -,PAN RAR 7 is the least significant bit of the address.

10.5 Break Function 10.5

Refer to Logic Diagrams CPU 011 and 041.

The state of the BREAK Switch synchronized with the microprogram, -,BREAK, is used to access the lower part of the ROB379.

When the microprogram is in the break routine, indicated by the signal EN BREAK = 1, the state of the Break Switch is "frozen" to the set state.

11. I/O BUS INTERFACE 11.

For details refer to Logic Diagrams CPU 029, 030, 031, 032 and 033, and Timing Diagrams TC2 and TC3.

This part of the CPU takes care of the communication between CPU and peripherals. This communication embodies a single 48-line bus connecting the control processor to all interfaces. Data is transferred on the bus along 16 parallel, bidirectional, data lines. Control signals are carried along dedicated, unidirectional, lines. In addition to specifying a unique function, each control signal generated by the CPU provides all timing necessary to perform that function. Data transfers are synchronous; no hand-shaking occurs between the interface and the CPU.

The data channel and program interrupt facility each use their own single request and priority lines. The two request lines are run in parallel to all interfaces, so that an interface requiring either data channel or program interrupt service needs only ground the appropriate line and wait for the CPU to respond. The serial priority lines are independent and are chained from interface to interface, so that priority for service is granted to the interface closest to the CPU on the chain.

11.1 I/O Data Bus Drivers and Receivers 11.1

Refer to Logic Diagrams CPU 029 and 030.

All data and addresses for both channel and programmed I/O are transferred between CPU and interfaces attached to the I/O bus via the 16 bidirectional lines, -,DATA 0-15. The interrupt disable mask and interrupt acknowledge information are also carried on these lines.

The bus drivers (75453) gate the information on the internal data bus, -,BUS 0-15, out on the I/O Data Bus, -,DATA 0-15, controlled by the microprogram do-function, -,GATE OUT.

The bus receivers (74S241) transfer the information on the I/O Data Bus, -,DATA 0-15, to the internal Data Bus, -,BUS 0-15, controlled by the microprogram do-function -,GATE IN and the instruction decoded -,READS.

11.2 Programmed Control Circuit

Refer to Logic Diagrams CPU 031 and 032.

The control signals are decoded from IR 5-15 (Instruction Register bit 5-15) and enabled by two microprogram do-functions, -,GENIODT and -,GENIOP. -.GENIODT is used to generate data strobe pulses on the I/O Bus (i.e., DATO A-B, DATI A-B, INTA, IORST and MSKO). -,GENIOP is used to generate control signals as STRT, CLR, IOPL and to set/reset Interrupt On flag.

Refer to the decoder truthtables for IC 122, IC 132 and IC 123. These truthtables are shown on figs. 41-43.

-,DEV 77/xx	IR5	IR6	IR7	Control Signal
1	0	0	0	
1	0	0	1	-,DTI A
1	0	1	0	
1	0	1	1	-,DTI B
1	1	0	0	
1	1	0	1	-,DTI C
1	1	1	0	
1	1	1	1	
0	0	0	0	
0	0	0	1	-,READS
0	0	1	0	
0	0	1	1	-,INTA
0	1	0	0	
0	1	0	1	-,IORST
0	1	1	0	
0	1	1	1	

Figure 41: Truthtable for Decoder IC 143 (-,GENIODT = 0).

-,DEV 77/xx	IR5	IR6	IR7	Control Signal
1	0	0	0	
1	0	0	1	
1	0	1	0	-,DTO A
1	0	1	1	
1	1	0	0	-,DTO B
1	1	0	1	
1	1	1	0	-,DTO C
1	1	1	1	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	-,MSKO x
0	1	0	1	
0	1	1	0	-,HALT
0	1	1	1	

Figure 42: Truthtable for Decoder IC 133 (-,GENIODT = 0).

-,DEV

77/xx	IR8	IR9	Control Signal
1	0	0	
1	0	1	-,STRT
1	1	0	-,CLR
1	1	1	-,IOP
0	0	0	
0	0	1	-,INTEN
0	1	0	-,INTDS
0	1	1	

Figure 43: Truthtable for Decoder IC 141 (-,GENIODT = 0).

All the control signals from the decoders are further amplified before they are sent out on the I/O Bus. All these control signals are further described below (the signal names used are those present on the I/O BUS).

DATO A-C Asserted by the CPU during the execution of a DO A-C instruction, after the CPU has placed the contents of the internal Data Bus, -,BUS 0-15, on -,DATA 0-15. Should cause the interface selected by -,DS 0-5 to load its A-C output buffer with the data on -,DATA 0-15.

-,DATI A-C Asserted by the CPU during the execution of a DI A-C instruction. Should cause the interface selected by -,DS 0-5 to place the contents of its A-C input buffer on -,DATA 0-15.

STRT Asserted by the CPU during the execution of any I/O instruction (except an I/O SKIP) in which bit 8-9 = 01. Not asserted during DI A-C instructions until after the data transfer has occurred. Usually used to initiate peripheral operation by setting the Busy flag to 1 and the Done flag to 0.

CLR Asserted by the CPU during the execution of any I/O instruction (except an I/O SKIP) in which bit 8-9 = 10. Not asserted during DI A-C and DO A-C instructions until after the data transfer has occurred. Used to terminate peripheral operation by setting the Busy and Done flags to 0.

IOPLS Asserted by the CPU during the execution of any I/O instruction (except an I/O SKIP) in which bit 8-9 = 11. Not asserted in DI A-C and DO A-C instructions until after the data transfer has occurred. Usually used to initiate special peripheral operations.

-,MSKO Asserted by the CPU during the execution of the MSKO instruction, after the contents of the designated accumulator have been placed on -,DATA 0-15. Used to load the contents of -,DATA 0-15 into the interrupt disable flip-flops of all interfaces using the interrupt system.

INTA Asserted by the CPU during the execution of the INTA instruction. If an interface receives INTA while it is also receiving -,INTP IN asserted and while it is requesting interrupt service, it should place its device code on -,DATA 10-15.

IORST Asserted by the CPU during the IORST instruction or when the Diagnostic Panel RESET switch is activated. IORST is also issued prior to CPU operation at power turn-on and when power is removed.

-,READS This is an internal CPU signal not sent out on the I/O Bus. -,READS strobes the contents of the Data switches on the Diagnostic Panel or the Front Panel of the CPU out on the internal Data Bus.

-,HALT Internal CPU signal. Asserted by the CPU during the HALT instruction; used to reset the RUN flip-flop.

-,INTEN Internal CPU signal. Asserted by the CPU during the execution of an NIOS, CPU instruction, used to set the Interrupt On flip-flop.

-,INTDS Internal CPU signal. Asserted by the CPU during the execution of an NIOC, CPU instruction, used to reset the Interrupt On flip-flop.

-,RQENB Asserted by the CPU to synchronize program interrupt and data channel requests from all interfaces. This signal is generated by the microprogram in the fetch cycle and during DMA cycles.

-,INTR Asserted by an interface to request program interrupt service. This signal is synchonized to the microprogram to cause the microprogram to enter the interrupt service routine if INTEN DLY'D = 1. INTEN DLY'D is the Interrupt On flip-flop delayed one instruction time.

-,SEL B Asserted by the interface selected by -,DS 0-5 if its Busy flag is set to one.

-,SEL D Asserted by the interface selected by -,DS 0-5 if its Done flag is set to one.

-,DS 0-5 These lines carry the low-order six bits of the instruction (IR 10-15) currently being executed. The microprogram generates the signal -,I/O SLO to strobe IR 10-15 out on -,DS 0-5 during I/O instruction execution. Only the interface whose device code corresponds to that carried on these lines should respond to control signals generated on the I/O Bus.

11.3 Data Channel Control Circuit 11.3

Refer to Logic Diagrams CPU 012, CPU 032 and CPU 033 and Timing Diagram TC2.

In the following description all the Data Channel I/O Bus signals are further described.

-,DCHR Asserted by a device when it requires data service. This signal is synchronized to the microprogram to condition the microprogram to enter the Data Channel service routine when the current instruction execution is terminated.

-,DCHA Asserted by the CPU at the beginning of each data channel cycle. Should cause the interface that is receiving an asserted -,DCHP IN signal and whose DCH REQ flip-flop is set, to set its DCH SEL flip-flop and place the memory address to be used for this transfer on the data lines and the mode on the data channel mode lines.

-,DCHM 0-1 Asserted by the interface whose DCH SEL flip-flop is set to inform the CPU of the type of data channel cycle to be performed, as follows:

-,DCHM 0	-,DCHM 1	FUNCTION
0	0	Output
0	1	Increment Memory
1	0	Input
1	1	Add. to Memory

-,DCH 1 Asserted by the CPU for data channel input. Should cause the interface whose DCH SEL flip-flop is set to place the contents of its input register on the data lines -,DATA 0-15.

-,DCH 0 Asserted by the CPU for data channel output, after the data word has been placed on the data lines -,DATA 0-15. Should cause the priority-selected interface to load the data from the data lines -,DATA 0-15.

OVFLO Asserted by the CPU during a data channel cycle that increments or adds to memory when the result exceeds $2^{16} - 1$.

12. INSTRUCTION REGISTER DECODER 12.

This circuit contains two decoders, further described in the following sections.

12.1 Device Address Decoder 12.1

Refer to Logic Diagram CPU 033 and 015.

The outputs from this decoder are -,DEV 77/xx (true when IR 10-15 contain device addr. 77), -,DEV 01/xx (true when IR 10-15 contain device addr. 01), -,DEV 2 (true when IR 10-15 contain addr. 02) and ARITH LD Q.

ARITH LD Q is used during execution of an arithmetic-logic instruction to determine whether the result of the ALU operation should be loaded into Q or into the Destination accumulator.

If IR 8 = IR 9 = 0, ARITH LD Q follows the NO LOAD bit of the instruction, IR 12.

If IR (8,9) # (0,0) (shift or swap), ARITH LD Q is always logic 1.

12.2 Accumulator Address Decoder 12.2

Refer to Logic Diagram CPU 013.

Normally, the accumulator addresses, ASEL 0-3 and BSEL 0-3, are supplied by the microprogram; but in some microaddresses ASEL 0-3 and B SEL 0-3 are supplied by a decoding of instruction register IR 1-4.

This decoder is used during one or two accumulator instructions (-,PAN ACN EN = 1) and in the front panel service routine (-,PAN ACN EN = 0).

The truthtable for this decoder is shown in fig. 44.

-,PANEL ACN EN	Source IR 1	2	Dest IR 3	4	B SEL 3	2	1	0	NC	A SEL 3	2	1	0	NC
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	0	0	0	0	0
1	0	0	1	0	0	0	1	0	2	0	0	0	0	0
1	0	0	1	1	0	0	1	1	3	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	0	1	0	0	0	1	1	0	0	0	1	1
1	0	1	1	0	0	0	1	0	2	0	0	0	1	1
1	0	1	1	1	0	0	1	1	3	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	0	2
1	1	0	0	1	0	0	0	1	1	0	0	1	0	2
1	1	0	1	0	0	0	1	0	2	0	0	1	0	2
1	1	0	1	1	0	0	1	1	3	0	0	1	0	2
1	1	1	0	0	0	0	0	0	0	0	0	1	1	3
1	1	1	0	1	0	0	0	1	1	0	0	1	1	3
1	1	1	1	0	0	0	1	0	2	0	0	1	1	3
1	1	1	1	1	0	0	1	1	3	0	0	1	1	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	0	0	1	0	2	0	0	0	0	0
0	0	0	1	1	0	0	1	1	3	0	0	0	0	0
0	0	1	0	0	0	1	0	0	4	0	0	0	0	0
0	0	1	0	1	0	1	0	1	5	0	0	0	0	0
0	0	1	1	0	0	1	1	0	6	0	0	0	0	0
0	0	1	1	1	0	1	1	1	7	0	0	0	0	0
0	1	0	0	0	1	0	0	0	8	0	0	0	0	0
0	1	0	0	1	1	0	0	1	9	0	0	0	0 / 1	0
0	1	0	1	0	1	0	1	0	10	0	0	0 / 1	0	0
0	1	0	1	1	1	0	1	1	11	0	0	0 / 1	0 / 1	0
0	1	1	0	0	1	1	0	0	12	0	0	0	0	0
0	1	1	0	1	1	1	0	1	13	0	0	0	0	0
0	1	1	1	0	1	1	1	0	14	0	0	0	0	0
0	1	1	1	1	1	1	1	1	15	0	0	0	0	0

-, PANEL ACN EN = 1 : REG FILE ADDR DURING ARITH & LOGIC INSTR.

-, PANEL ACN EN = 0 : REG FILE ADDR DURING FRONT PANEL OPERATION

A addr : Only Read
B addr : Read / Write

Figure 44: Register File Address ROM 763.

13. CLOCK GENERATOR 13.

Refer to Logic Diagram CPU 021 and Timing Diagram TC4.

The master clock is a 20 MHz crystal oscillator.

The output from the crystal oscillator is divided by x 2 to generate a 10 MHz clock signal. The 20 MHz clock signal is also divided by a 4 bits counter to generate the microprogram step clock.

This counter will normally count 0 - 1 - 2 and then be reset to zero again, except in conditionally steps, where the signal SEL 1ST OP disables the reset to the counter. For this reason a conditional step has a counter sequence as follows: 0 - 1 - 2 - 3.

The length of an unconditional step is 150 ns.

The length of a conditional step is 200 ns.

When the microprogram waits upon memory to be ready, the counter IC 73 is held in parallel load mode until the signal -,STOP CLOCK switches to logic 1, then the counter starts in state 0.

14. POWER MONITOR 14.

For details refer to Logic Diagram CPU 021 and Timing Diagram TC5.

In the event of a power failure it is important that the CPU be halted before reliable power is lost. For this reason two signals from the power supply are monitored by the CPU. These two signals are -,PINT and POK. -,PINT gives an early warning of a power failure. POK is cleared when any of the DC voltages fall below their nominal thresholds.

In case of power down, -,PINT gives an early warning. -,PINT interrupts the CPU and sets the PWR FAIL flag. From this point there is approximately 1 to 2 ms. before MEM OK disappears and prevents the memory system from operating.

When power is restored, a 50 ms. delay is initiated after MEM OK and -,PWR FAIL have both gone high. The restart function is performed at the end of this delay, causing a jump direct to location 0.

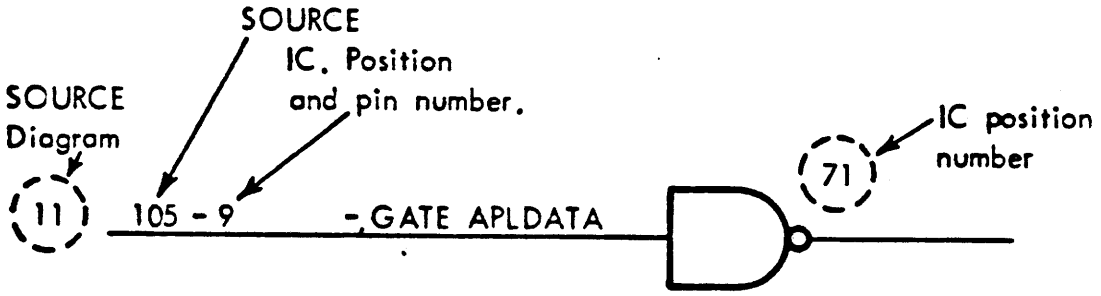
15. LOGIC DIAGRAM DESCRIPTION 15.

15.1 Signal Names 15.1

In CPU721 a signal can be either asserted when low or when high, depending on how it is defined. To distinguish between the two types of signals a naming convention has been adapted which defines the relationship between the logical and electrical levels of a signal. If the signal name includes a "-," before the signal name, as -,BUS 8, then that signal is asserted when it is at a low electrical level; conversely, a signal without "-,", BUS 8, is asserted when high.

15.2 Logic Diagram References 15.2

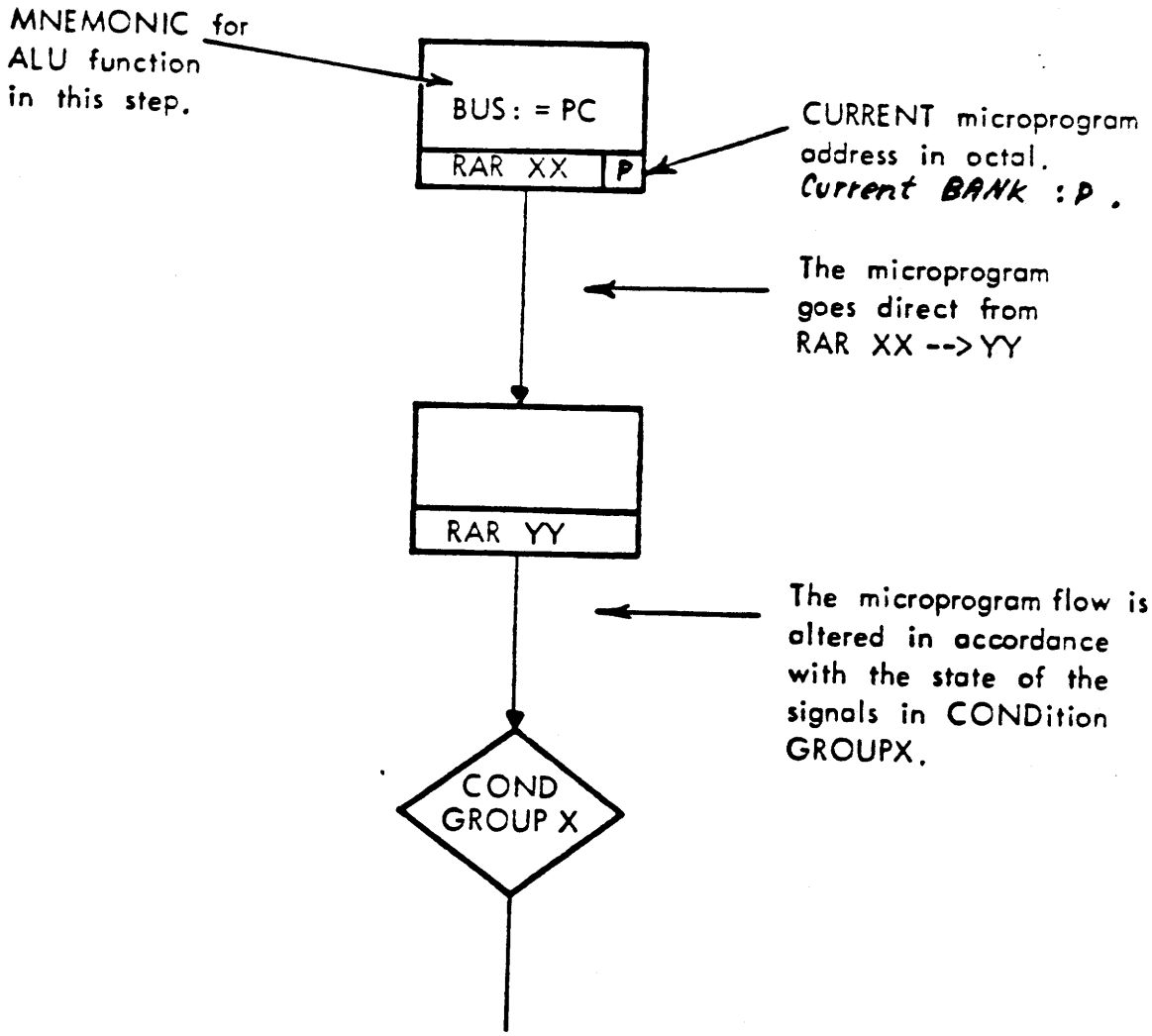
Refer to the figure below.



This means that the signal "-,GATE APL DATA" is generated on diagram No 11 by a logic circuit placed in position 105. The signal is generated by IC 105 pin number 9.

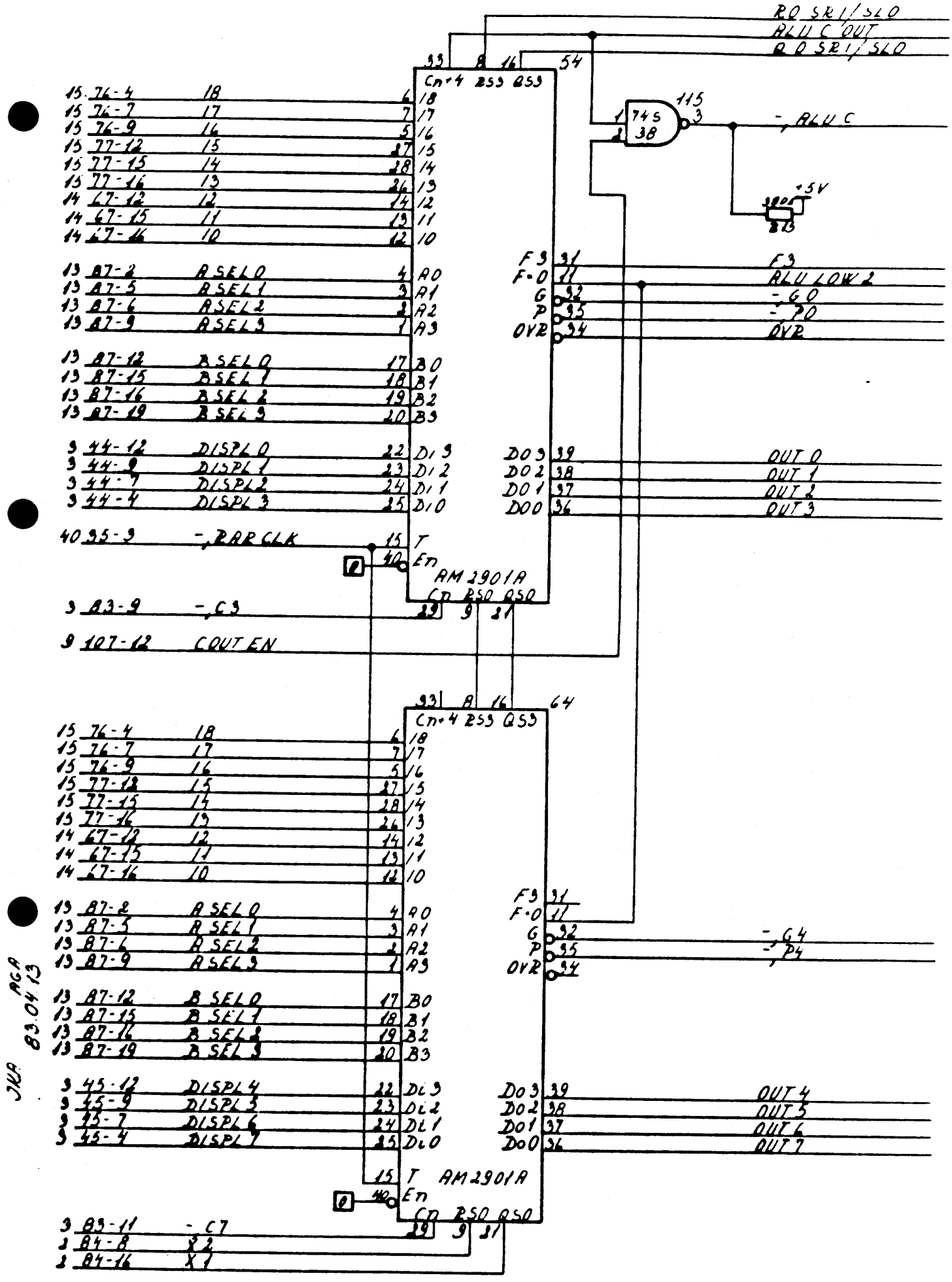
16. FLOW DIAGRAM DESCRIPTION 16.

To understand the microprogram flow diagrams, refer to the figure below.



17. LOGIC DIAGRAMS 17.

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
ALU COUT		p. 10 p. 42	ALU Carry OUTput The carry-out of the AM 2901A array.
⁻; ALU COUT		p. 16	⁻; ALU Carry OUTput The carry-out of the AM 2901A array gated by COUT EN.
ALU LOW 2		p. 26	ALU LOW output 2 This is an open collector output which goes high if the data on the ALU outputs bit 0-7 are all low.
⁻; G0, ⁻; G4		p. 3	⁻; carry Generate outputs The carry generate outputs of the AM 2901A are used for carry-lookahead.
OUT 0-7		p. 5	OUTput bit 0-7 Output data contain either the outputs of the ALU or the data on the A-port of the register stack (16-8).
OVR		p. 16	OVeRflow Indicates that the result of an arithmetic two's complement operation has overflowed into the sign-bit.
⁻; P0, ⁻; P4		p. 3	⁻; carry Propagate outputs Used for carry lookahead.
Qo SRI/SLO		p. 18	Qo Shift Right Input/Shift Left Output. Q-register shift.
Ro SRI/SLO		p. 18	Ro Shift Right Input/Shift Left Output. Register stack shift.
F3		p. 42	The most significant ALU output bit (sign).



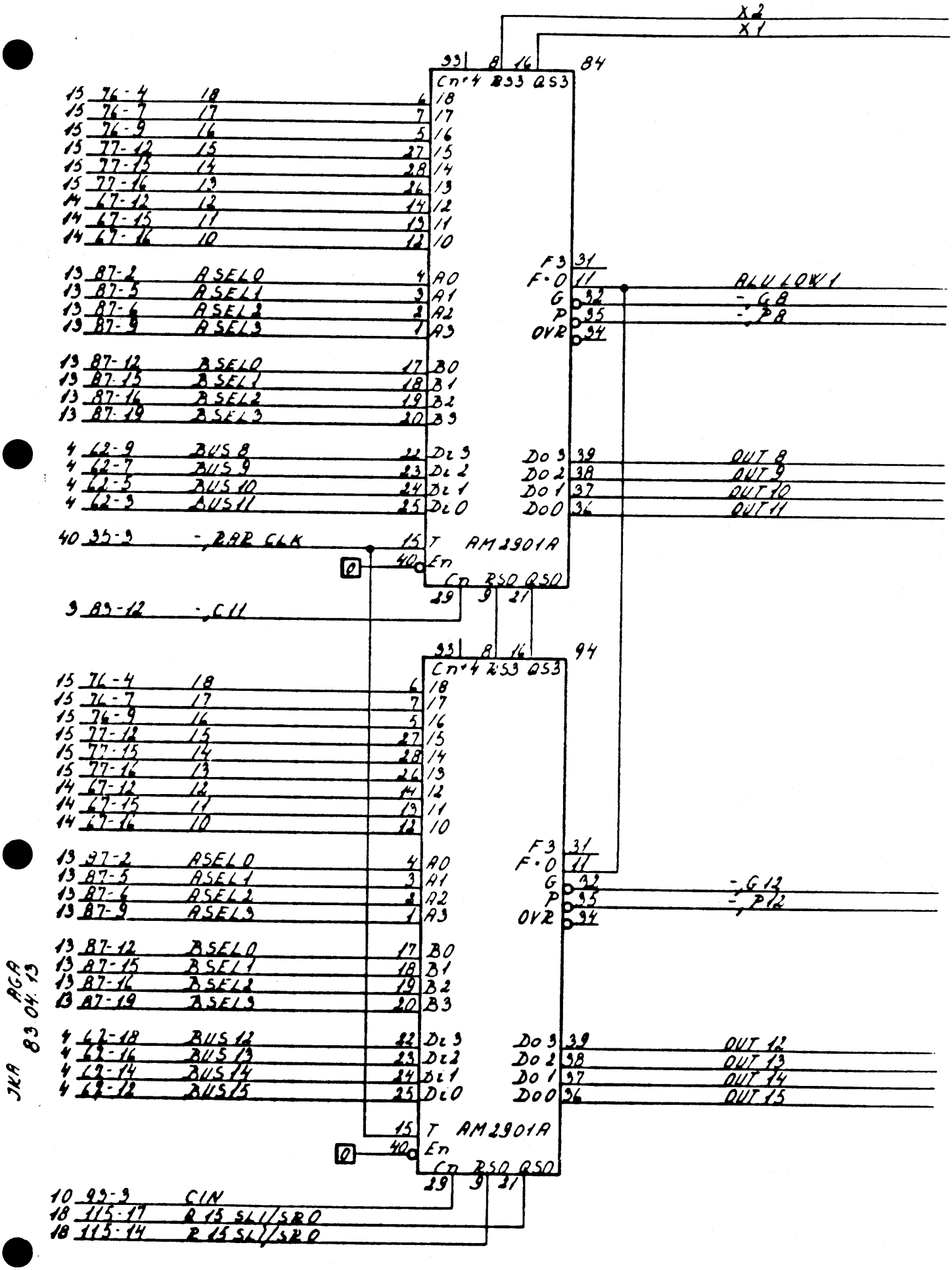
CPU 721

R 13574

MICROPROCESSOR SLICES BIT 0-3 & 4-7
LOGIC Diagram

1

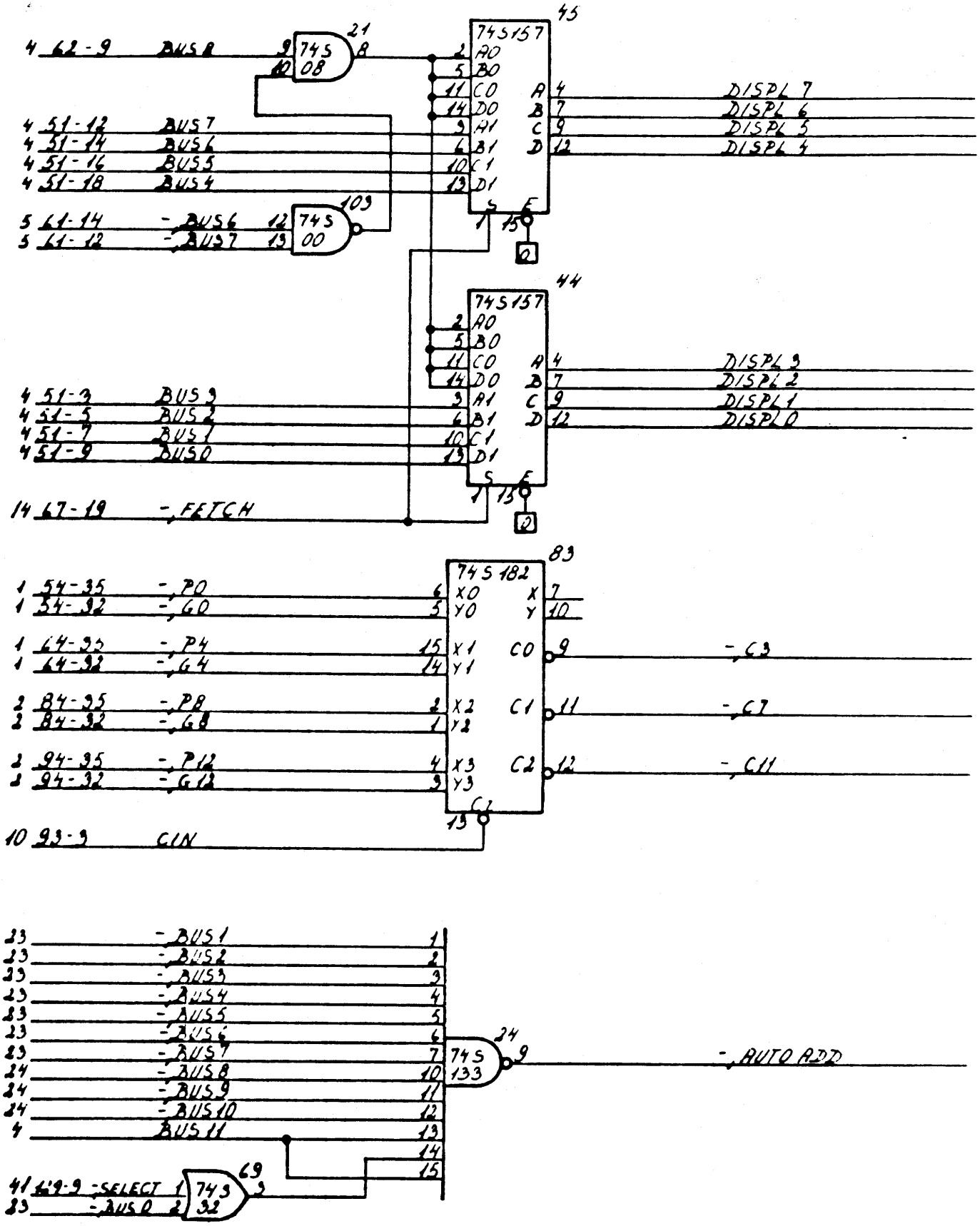
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
ALU LOW 1		p. 26	ALU LOW output 1 This is an open collector output which goes high if the data on the ALU outputs, bit 8-15 are all low.
¯; G8, ¯; G12		p. 3	¯; carry Generate outputs Used for carry lookahead.
OUT 8-15		p. 5	OUTput bit 8-15 Output data contain either the outputs of the ALU or the data on the A-port of the register stack (16-8).
¯; P8, ¯; P12		p. 3	¯; carry Propagate outputs Used for carry lookahead.
x1		p. 1	Q_7 Shift Right Input/Shift Left output.
x2		p. 1	R_7 Shift Right Input/Shift Left output.



CPU 721 - MICROPROCESSOR SLICES BIT 8-11 & 12-15
LOGIC Diagram

R 13575

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-; AUTO ADD		p. 11	AUTO increment/decrement ADDress.
-; C3		p. 1	-; Carry 3 Output from the carry look ahead generator.
-; C7		p. 1	-; Carry 7 Output from the carry look ahead generator.
-; C11		p. 2	-; Carry 11 Output from the carry look ahead generator.
DISPL 0-7		p. 1	DISPLacement bit 0-7 This data line contains either BUS 0-7 or the sign of the displacement in the instruction set extended through bit 0-7.

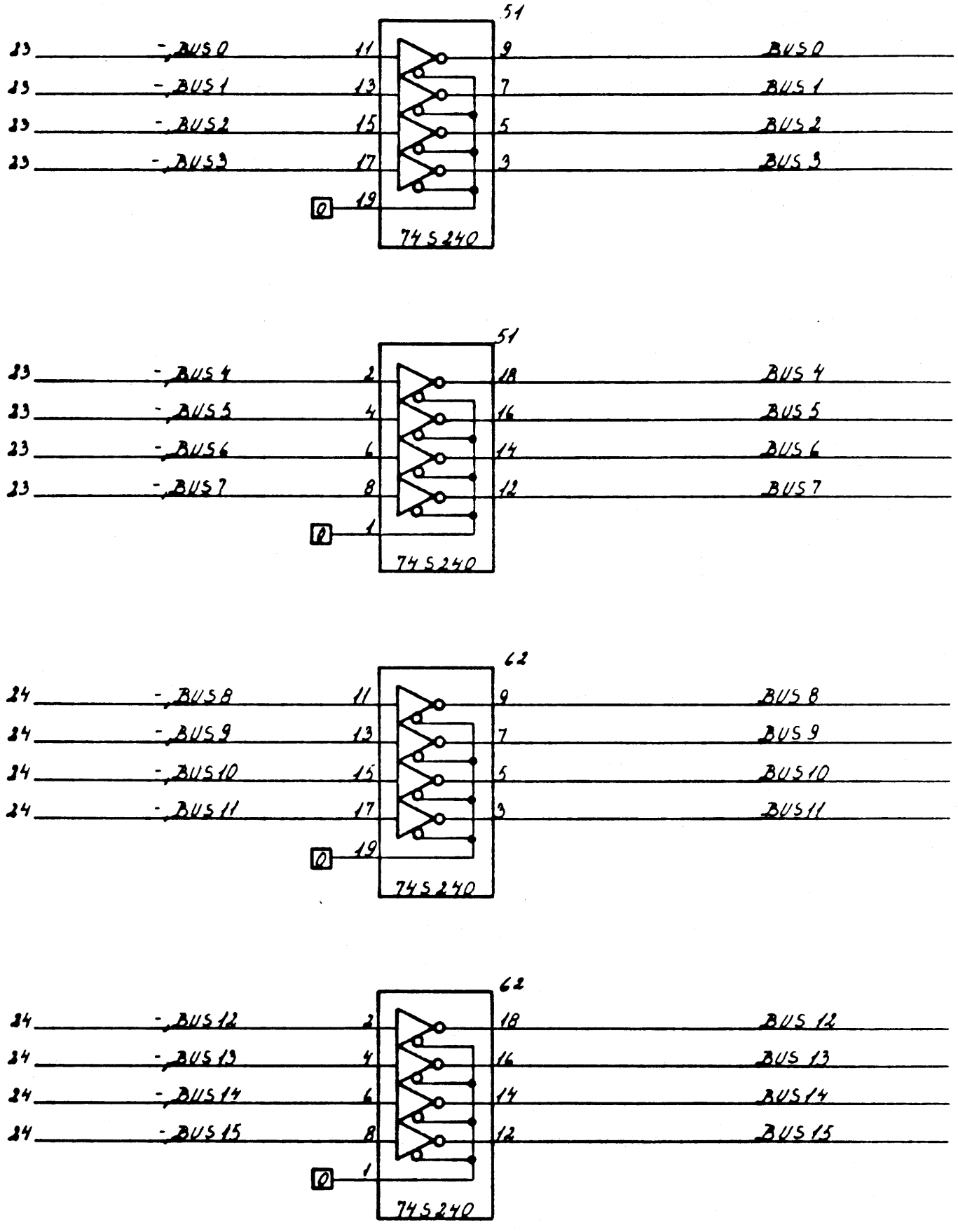


CPU 721

AUTO INCR/DECR DECODER, CARRY LOOK AHEAD
SIGN EXTEND CIRCUIT
LOGIC DIAGRAM

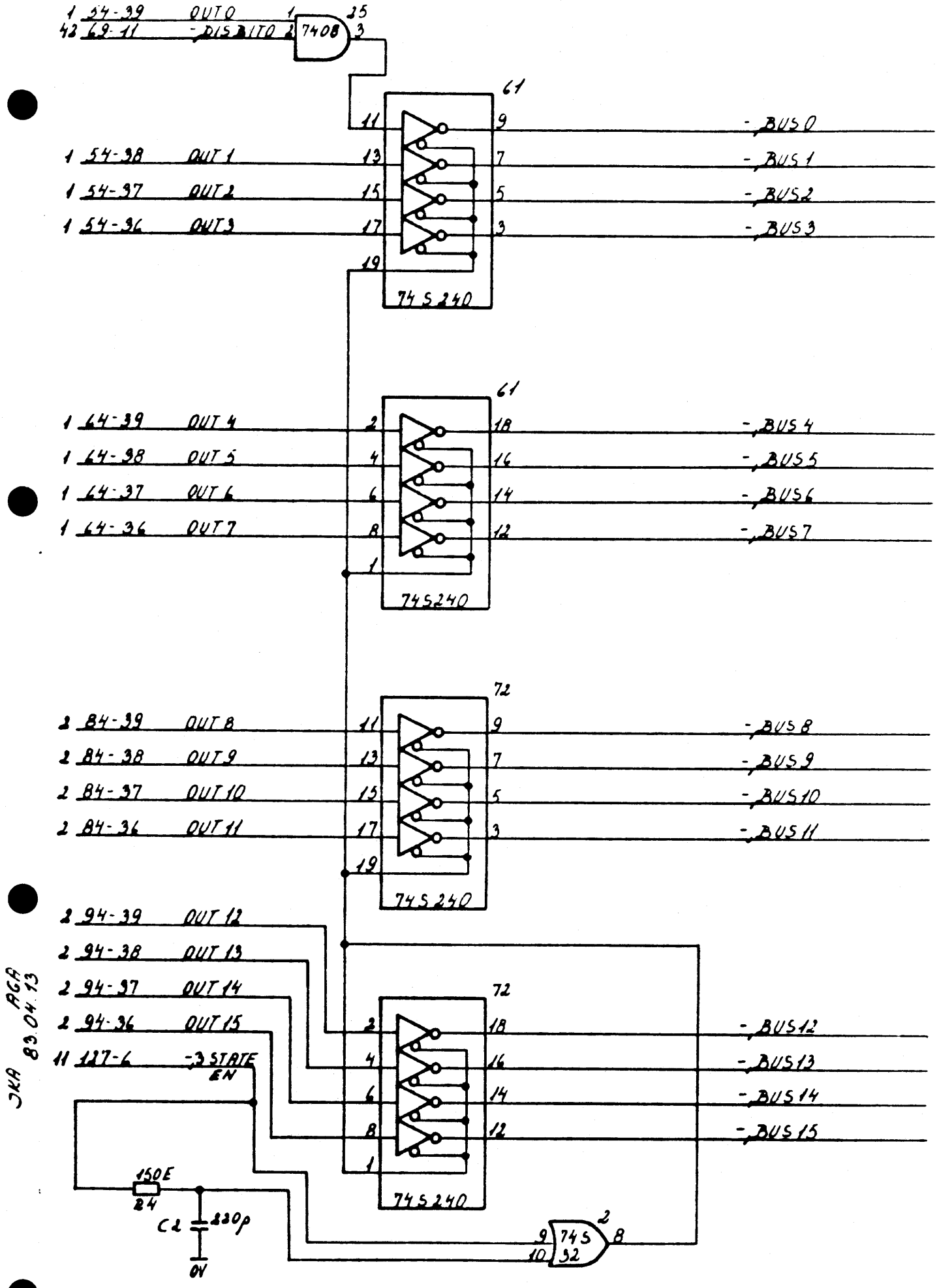
2 13576

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
BUS 0		p. 3 p. 6 p. 8 p. 16 p. 17 p. 41	Internal BUS bit 0.
BUS 1-7		p. 3 p. 6 p. 8	Internal BUS bit 1-7.
BUS 8, 11		p. 2 p. 3 p. 6	Internal BUS bit 8, 11.
BUS 9-10, 13-14		p. 2 p. 6	Internal BUS bit 9-10, 13-14.
BUS 15		p. 2 p. 6 p. 16	Internal BUS bit 15.
BUS 12		p. 2 p. 6 p. 11	Internal BUS bit 12



JKA RGA 83 04 13

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻; BUS 0-7		p. 23	⁻; internal BUS bit 0-7.
⁻; BUS 8-15		p. 24	⁻; internal BUS bit 8-15.



JKA AGA 83.04.13

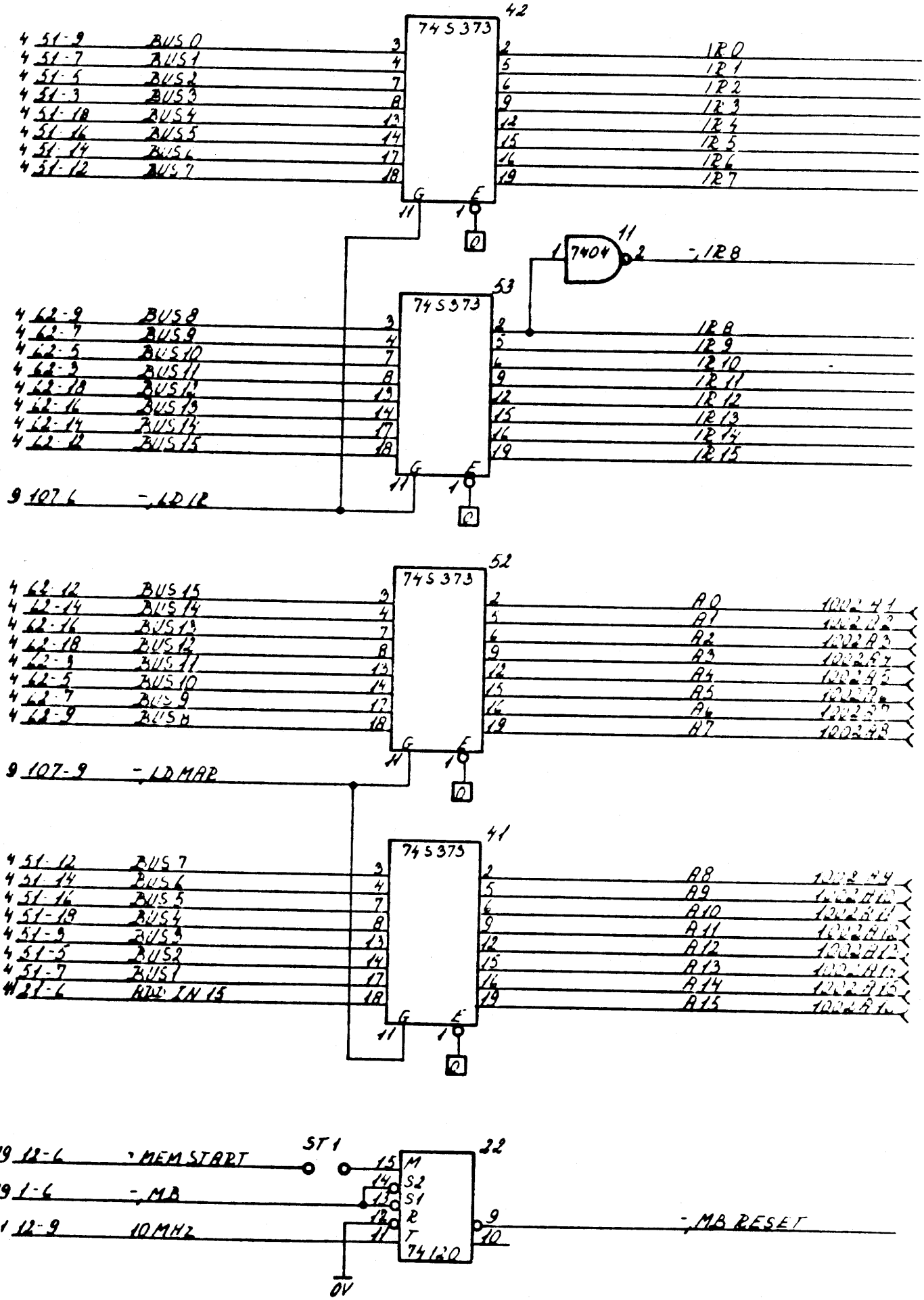
CPU 721

R 13578

INVERTER FOR MICROPROCESSOR OUTPUT
LOGIC DIAGRAM

5

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
A0-A15		1002	memory Address 0-15. A0 is the least significant bit of the memory address. A15 is normal zero except when memory extension is selected.
IR 15		p. 18 p. 33 p. 22	Instruction Register bit 15.
IR 14-13		p. 17 p. 18 p. 33 p. 22	Instruction Register bit 14-13.
IR 12		p. 16 p. 17 p. 33 p. 22	Instruction Register bit 12.
IR 11-10		p. 16 p. 33 p. 22	Instruction Register bit 11-10.
IR 9		p. 16 p. 17 p. 18 p. 31 p. 33	Instruction Register bit 9.
IR 8		p. 16 p. 17 p. 31 p. 33	Instruction Register bit 8.
IR 7-5		p. 17 p. 31	Instruction Register bit 7-5.
IR 4-2		p. 17 p. 13	Instruction Register bit 4-2.
IR 1		p. 18 p. 17 p. 13	Instruction Register bit 1
IR 0		p. 18	Instruction Register bit 0.
-, IR 8		p. 33	-, Instruction Register bit 8.
-, MB RESET		p. 19	-, Memory Busy RESET. Reset pulse to the Memory Start flip-flop, generated on the Leading edge of Memory Busy.



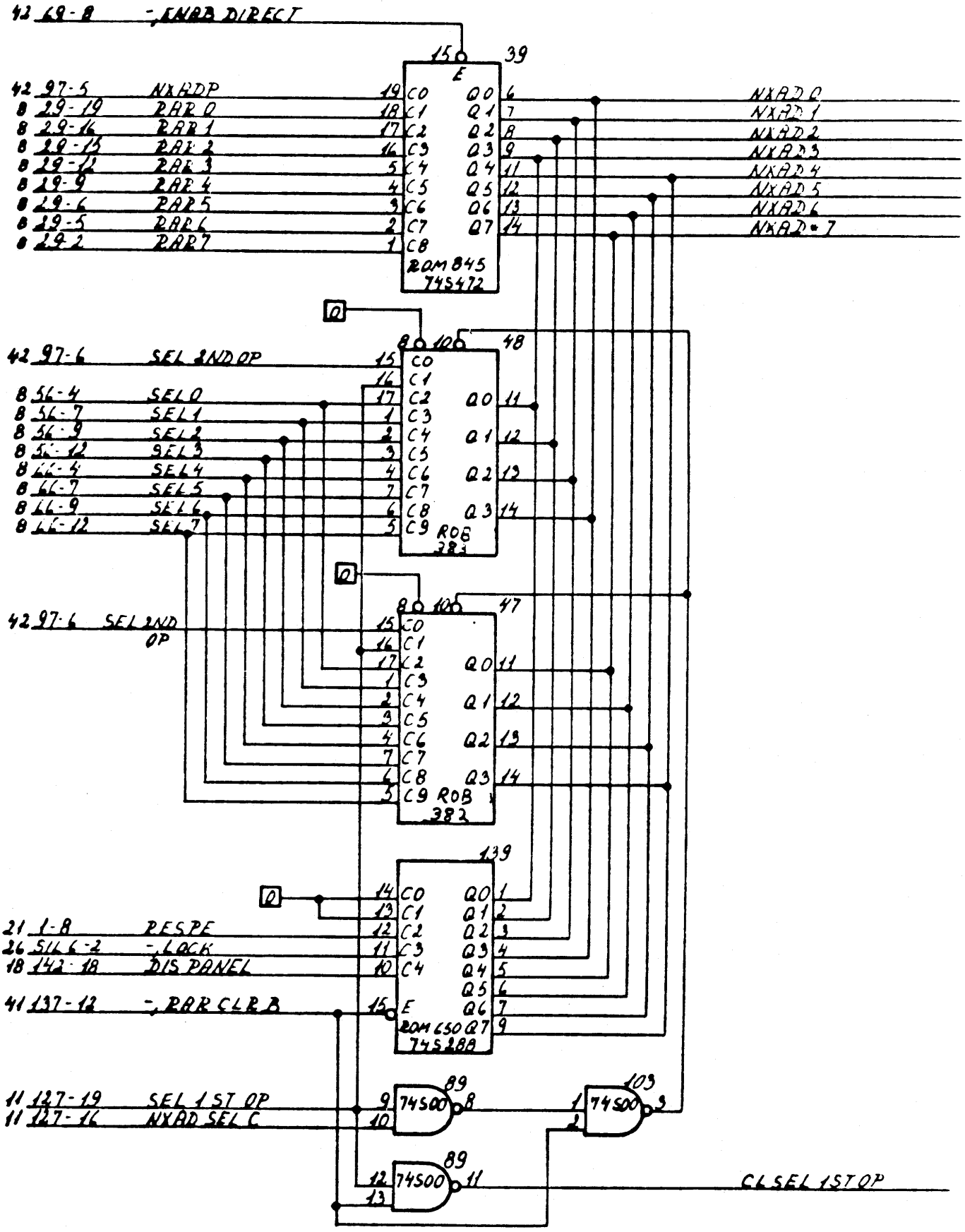
CPU 721

R 19579

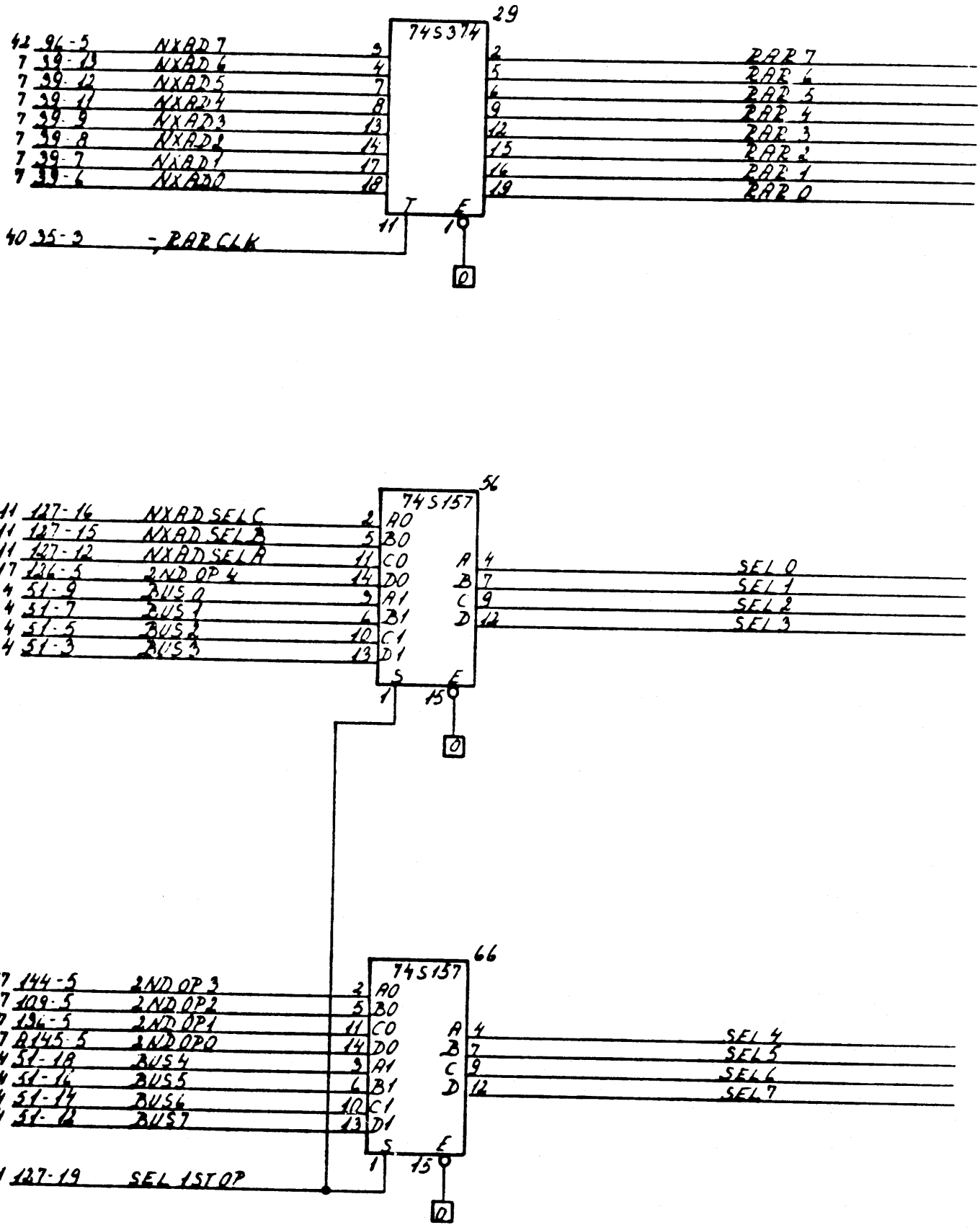
INSTRUCTION REGISTER-MEMORY ADDRESS REGISTER LOGIC DIAGRAM

6

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
NXAD 0-6		p. 8 p. 9 p. 10 p. 11 p. 12 p. 13 p. 14 p. 15 p. 16 p. 41 p. 42	NeXt micro ADdress bit 0-7. NXAD 7 is the least significant bit of the address. During fetch or conditional jump the next address is read out from ROM Unconditional jump address is read out from ROM Start address after power up is read out from ROM
NXAD * 7		p. 42	
CL SEL 1ST OP		p. 42	CLocked SEL 1ST OP


42 69-8 -,ENAB DIRECT
15
E
39
42 97-5 NXADP
8 29-19 PAR 0
8 29-16 PAR 1
8 29-15 PAR 2
8 29-12 PAR 3
8 29-9 PAR 4
8 29-6 PAR 5
8 29-5 PAR 6
8 29-2 PAR 7
C0 C1 C2 C3 C4 C5 C6 C7 C8
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
ROM 845
74S472
NXAD 0
NXAD 1
NXAD 2
NXAD 3
NXAD 4
NXAD 5
NXAD 6
NXAD - 7
Q
8
10
48
42 97-6 SEL 2ND OP
8 56-4 SEL 0
8 56-7 SEL 1
8 56-9 SEL 2
8 56-12 SEL 3
8 66-4 SEL 4
8 66-7 SEL 5
8 66-9 SEL 6
8 66-12 SEL 7
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9
Q0 Q1 Q2 Q3
ROB
383
47
ROB
382
139
21 1-8 RESPE
26 56-2 -LOCK
18 142-18 DIS PANEL
41 137-12 -,RAR CLR B
C0 C1 C2 C3 C4 E
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
ROM C50
74S288
11 127-19 SEL 1ST OP
11 127-16 NXAD SEL C
89
74S00
109
CL SEL 1ST OP
JKA 89.04.13 AG.A

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
RAR 0-7		p. 7 p. 38	Rom Address Register bit 0-7.
SEL 0-7		p. 7	SELect bit 0-7 These signals contain an eight bits address to the NEXT ADDRESS table. During the Fetch cycle (SEL 1ST OP = 1) these signals simply are BUS 0-7. During conditional jump (SEL 1ST OP = 0) these signals are the outputs from the condition selector extended to an eight bits address by help of NXAD SEL C-A (condition selection).

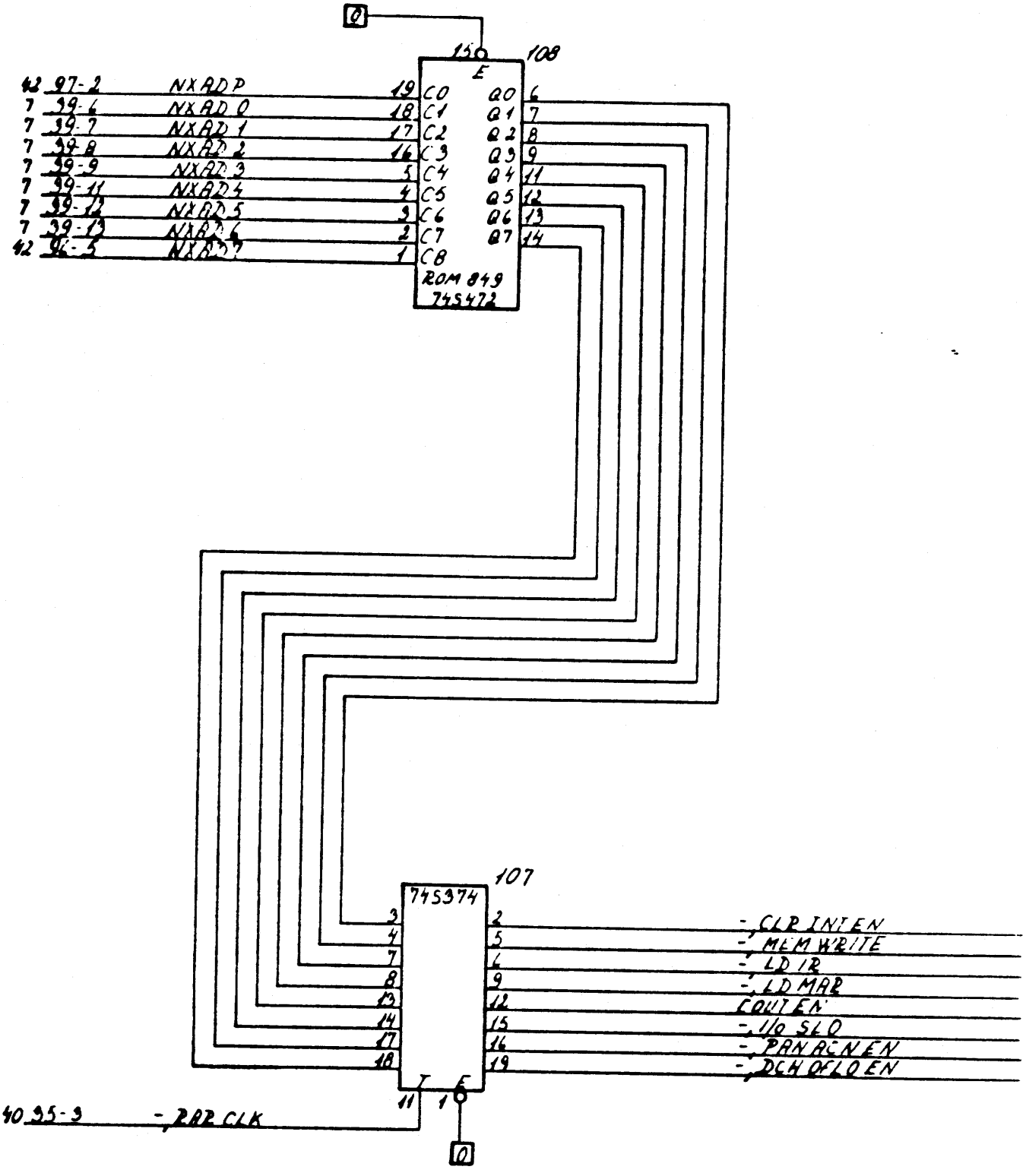


CPU 721

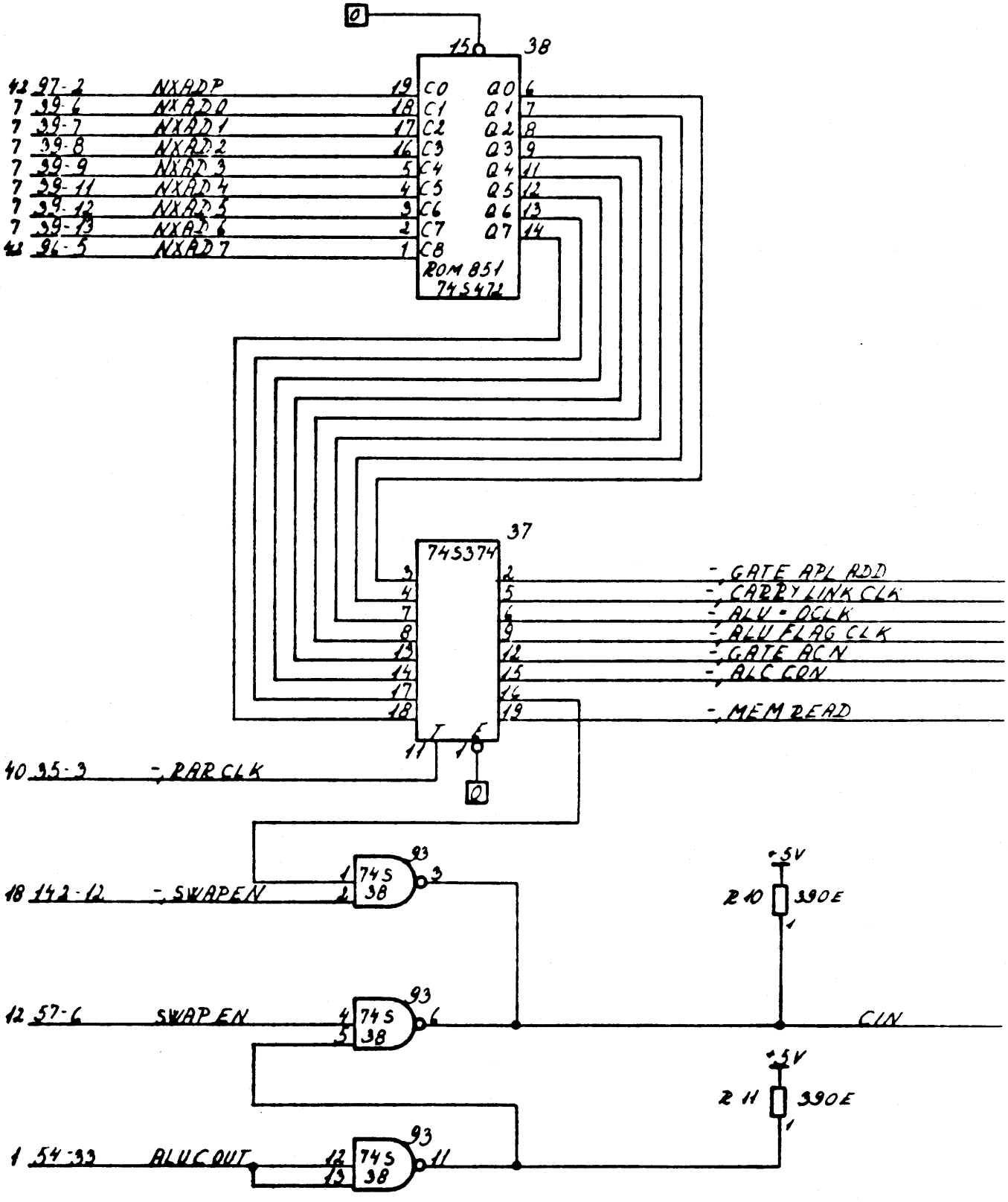
R 13581

MICRO ADDRESS REGISTER
OP SELECT CIRCUIT
LOGIC DIAGRAM

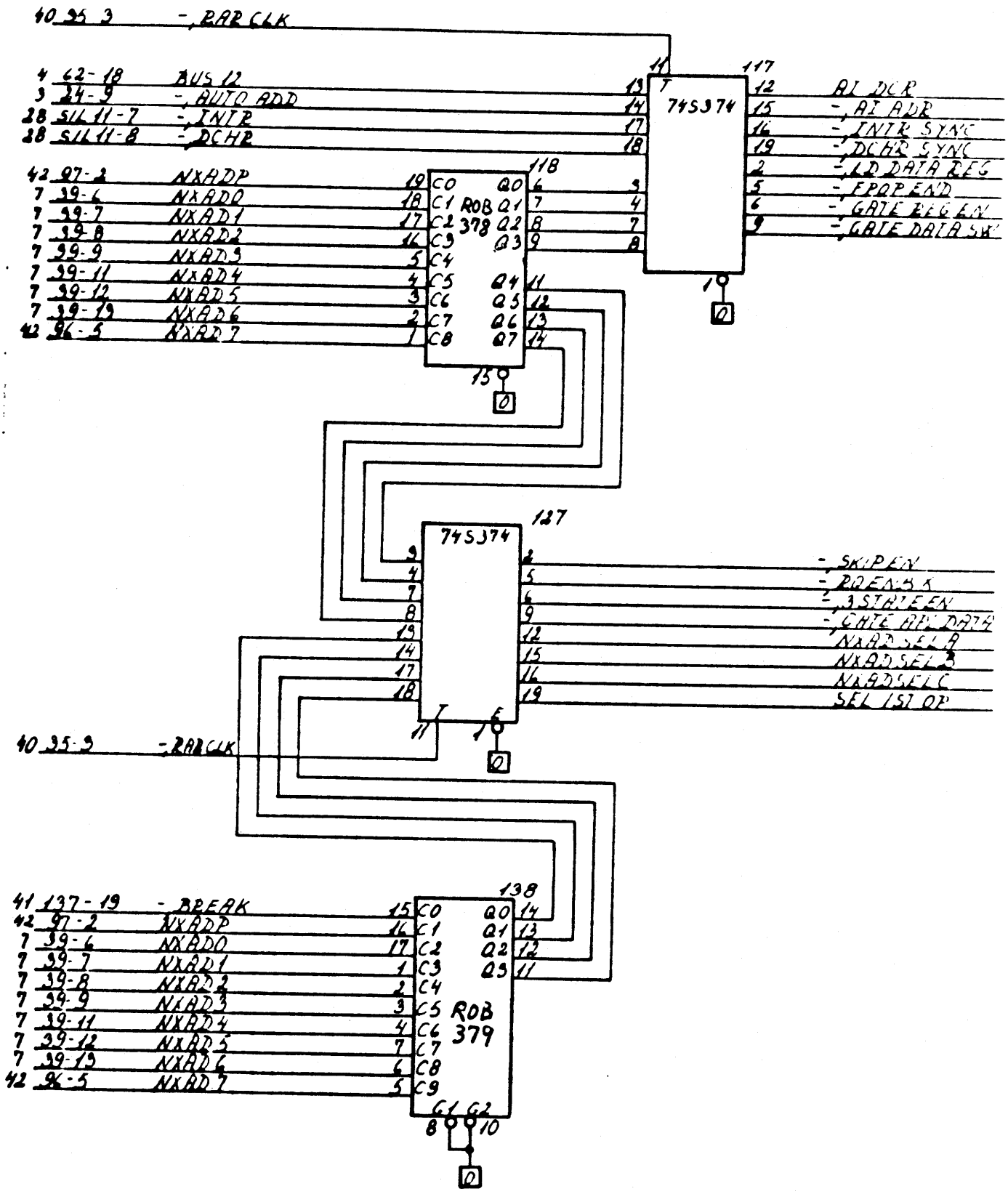
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~; CLR INT EN		p. 31	~; CLeaR INTerrupt ENable Clear Interrupt ON to prevent the processor from responding to further interrupt requests.
COUT EN		p. 1	Carry OUT ENable Used to enable carry out from the AM 2901A array.
~; DCH OFLO EN		p. 33	~; Data CHannel OverFLOw ENable Used to strobe the Overflow status out on the Data Channel control line OVFLO.
~; I/O SLO		p. 33	~; Input/Output device SeLect Out Places instruction register bit 10-15 on the Device Selection lines (DS 0-5).
~; LD IR		p. 6	~; LoaD Instruction Register Loads the contents on the data bus into the instruction register.
~; LD MAR		p. 6 p. 21	~; LoaD Memory Address Register Loads the contents on the data bus into the memory address register.
~; MEM WRITE		p. 19 p. 20	~; MEMory WRITE Starts the memory up with a write cycle.
~; PAN ACN EN		p. 13	~; PANel ACcumulator ENable Enables decoding of ASEL 0-3 and BSEL 0-3 from the state of the AC Sel switches on the Diagnostic Panel TCP 701.


Q
E
108
ROM 849
74S472
NXADP
NXAD0
NXAD1
NXAD2
NXAD3
NXAD4
NXAD5
NXAD6
NXAD7
CO
C1
C2
C3
C4
C5
C6
C7
C8
QO
Q1
Q2
Q3
Q4
Q5
Q6
Q7
107
74S374
-CLR INTEN
-MEMWRITE
-LD IR
-LD MAR
EOUTEN
-I/O SLO
-PANACNEN
-DCHOFLOEN
-RAR CLK
JLA
AGH
83 04 13

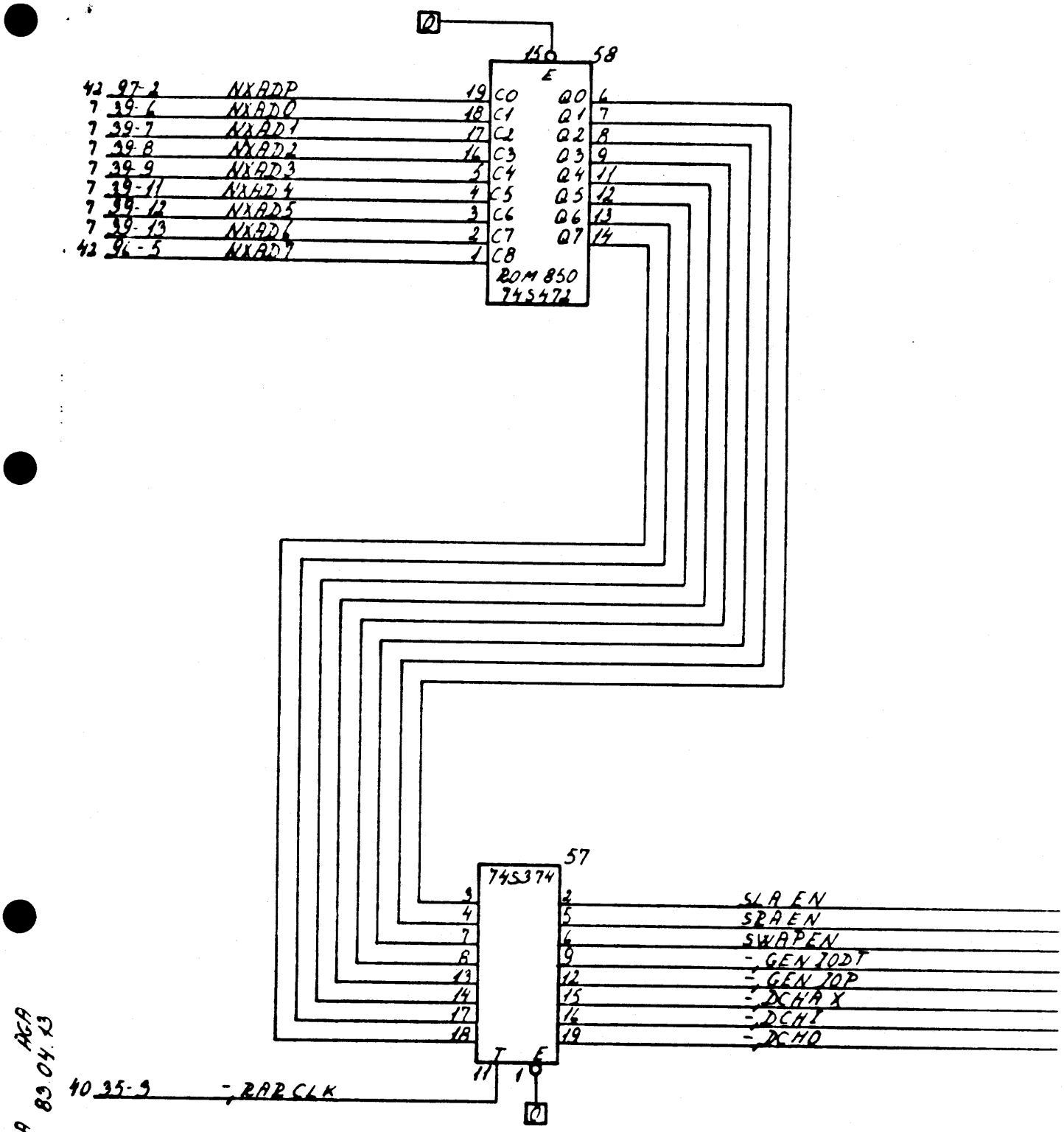
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~ ALC CON		p. 15	~ ALu Condition CONtrol Enables a decoder on IR 8-15 to determine if the result from the arithmetic operation should be loaded into Q or the specified distination accumulator.
~ ALU = 0 CLK		p. 18	~ ALU = 0 CLocK Loads the state of the zero decoder output from AM 2901A into the ALU = 0 status register.
~ ALU FLAG CLK		p. 16	~ ALU FLAG CLocK Updates the ALU status register.
~ CARRY LINK CLK		p. 18	~ CARRY LINK CLocK Loads the new carry from the Carry Generator into the Carry Register.
CIN		p. 2 p. 3	Carry INput Carry Input to the AM 2901 array.
~ GATE ACN		p. 13	~ GATE ACcumulator eNable Enables decoding of ASEL 0-3 and BSEL 0-3 from IR 1-4.
~ GATE APL ADD		p. 22 p. 37	~ GATE Automatic Program Load ADDress Gates the contents of the APL address counter out on the data bus.
~ MEM READ		p. 19 p. 20	~ MEMory READ Starts the memory up in a READ cycle.


38
NXADP
NXAD0
NXAD1
NXAD2
NXAD3
NXAD4
NXAD5
NXAD6
NXAD7
ROM 851
74S472
37
74S374
-GATE APL ADD
-CARRY LINK CLK
-ALU-OCLK
-ALU FLAG CLK
-GATE ACN
-ALC CON
-MEM READ
-RAR CLK
74S 38
-SWAPEN
SWAPEN
ALUCOUT
+5V
R 10 390E
R 11 390E
CIN

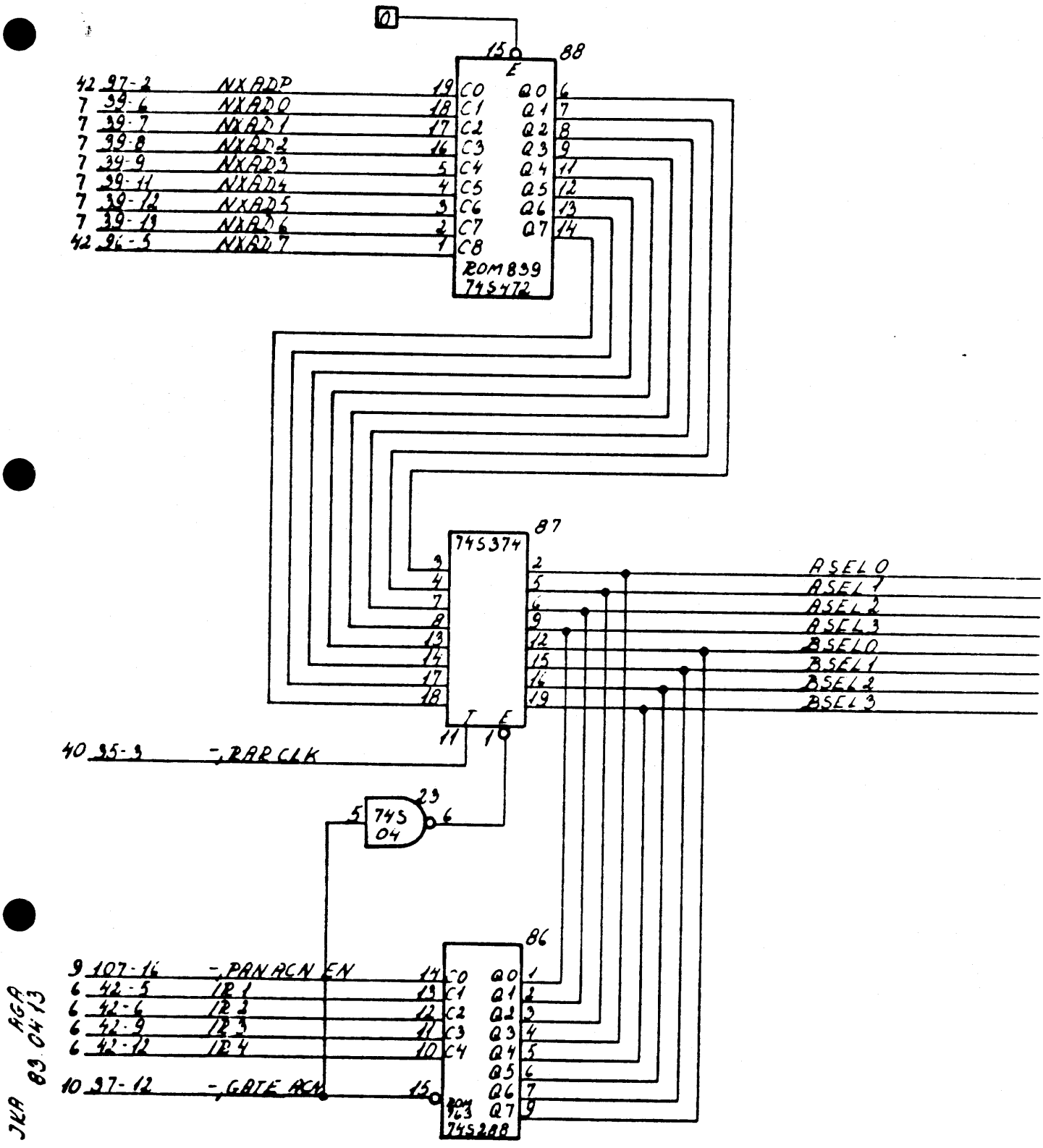
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻; GATE APL DATA		p. 18 p. 22	⁻; GATE Automatic Program Load DATA The contents of the addressed location of the autoload PROM is gated out on the data bus.
NXAD SEL A		p. 8 p. 17 p. 42	NeXt ADdress SEL A-C Determines whether the next address calculation is conditional,
NXAD SEL B		p. 8 p. 17	unconditional or jumps from fetch cycle to the start-address of the microsequence to execute the
NXAD SEL C		p. 7 p. 8 p. 17	current instruction.
⁻; RQEN B x		p. 32	⁻; ReQuest ENaBle x Allows all devices on the I/O BUS to request program interrupts or data channel requests.
SEL 1ST OP		p. 7 p. 8 p. 21	SELect 1ST OPerand MSB address input to the next address table.
⁻; SKIP EN		p. 18	⁻; SKIP ENable Enables SKIP condition GENerator.
⁻; 3 STATE EN		p. 5	⁻; 3 STATE ENable Gates data from AM 2901 out on the internal data bus.
-, AIADR		p. 17	-, Automatic Indexing ADdRess
AIDCR		p. 17	Automatic Indexing DeCRement
-, DCHR SYNC		p. 17 p. 42	-, Data CHannel Requests SYNChronized
-,FPOP END		p. 37	-, Front Panel OPeration END
-, GATE DATA SW		p. 34	-, GATE DATA SWitches
-, GATE REG EN		p. 35	-, GATE REGister ENable
-, INTR SYNC		p. 17 p. 42	-, INTerrupt Requests SYNChronized
-, LD DATA REG		p. 36	-, LoaD DATA REGister


40 35 3 - RAR CLK
4 62-18 BUS 12
3 24-9 - AUTO ADD
28 SIL 11-7 - INTR
28 SIL 11-8 - DCHR
42 97-2 NXADP
7 39-6 NXAD0
7 39-7 NXAD1
7 39-8 NXAD2
7 39-9 NXAD3
7 39-11 NXAD4
7 39-12 NXAD5
7 39-13 NXAD6
42 96-5 NXAD7
118
ROB 378
117
74S374
AI DCR
- AI ADR
- INTR SYNC
- DCHR SYNC
- LD DATA REG
- EPOP END
- GATE REG EN
- GATE DATA SW
127
74S374
- SKIP EN
- RDEN MX
- 3 STATE EN
- GATE RR DATA
NXAD SEL A
NXAD SEL B
NXAD SEL C
SEL IST OP
40 35-9 - RAR CLK
138
ROB 379
41 137-19 - BREAK
42 97-2 NXADP
7 39-6 NXAD0
7 39-7 NXAD1
7 39-8 NXAD2
7 39-9 NXAD3
7 39-11 NXAD4
7 39-12 NXAD5
7 39-13 NXAD6
42 96-5 NXAD7
JKA AGA 83 04 13

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-; DCHA x		p. 32	-; Data CHannel Acknowledge x Strobes the memory address from current device out on the I/O BUS.
-; DCH I		p. 32	-; Data CHannel Input.
-; DCH O		p. 32	-; Data CHannel Output.
-; GEN IODT		p. 31	-; GENerate Input Output DaTa pulse Used to generate the I/O BUS signals DATO A-B, DATI A-B, INTA, MSKO and IORST.
-; GEN IOP		p. 31	-; GENerate Input Output Pulse Used to generate STRT, CLR and IOPL.
SLA EN		p. 18	Shift Left Accumulator ENable Connets SHIFT GENERATOR to Q_{15} SLI/SRO.
SRA EN		p. 18	Shift Right Accumulator ENable Connects SHIFT GENERATOR to Qo SRI/SLO.
SWAP EN		p. 10 p. 18	SWAP ENable Connects Ro SRI/SLO to R_{15} SLI/SRO, and ALU COUT to CIN.

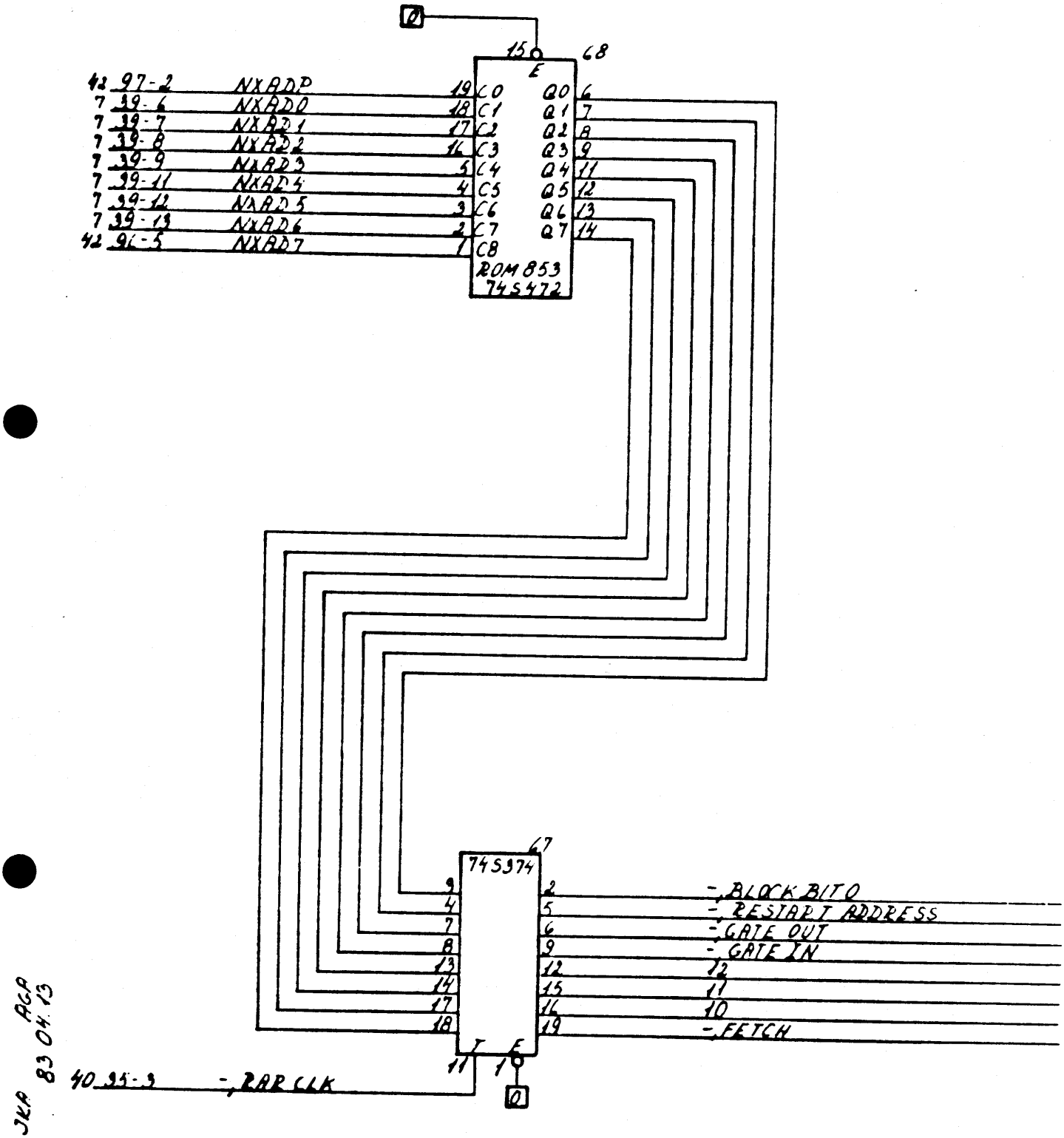

58
E
42 97-2 NXADP
7 39-6 NXAD0
7 39-7 NXAD1
7 39-8 NXAD2
7 39-9 NXAD3
7 39-11 NXAD4
7 39-12 NXAD5
7 39-13 NXAD6
42 96-5 NXAD7
19 C0
18 C1
17 C2
16 C3
5 C4
4 C5
3 C6
2 C7
1 C8
Q0 6
Q1 7
Q2 8
Q3 9
Q4 11
Q5 12
Q6 13
Q7 14
ROM 850
74S472
57
74S374
9
4
7
8
13
14
17
18
2
5
4
9
12
15
16
19
SLAEN
SRAEN
SWAPEN
-GEN IODT
-GEN IOP
-DCHAX
-DCHI
-DCHO
11
1
40 35-9 -RAR CLK
JKA 83 04 23 AGA

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
A SEL 0-3		p. 1 p. 2	A register SELect 0-3 Read only register address input to AM 2901 array.
B SEL 0-3		p. 1 p. 2	B register SELect 0-3 read-write register address input to the AM 2901 array.

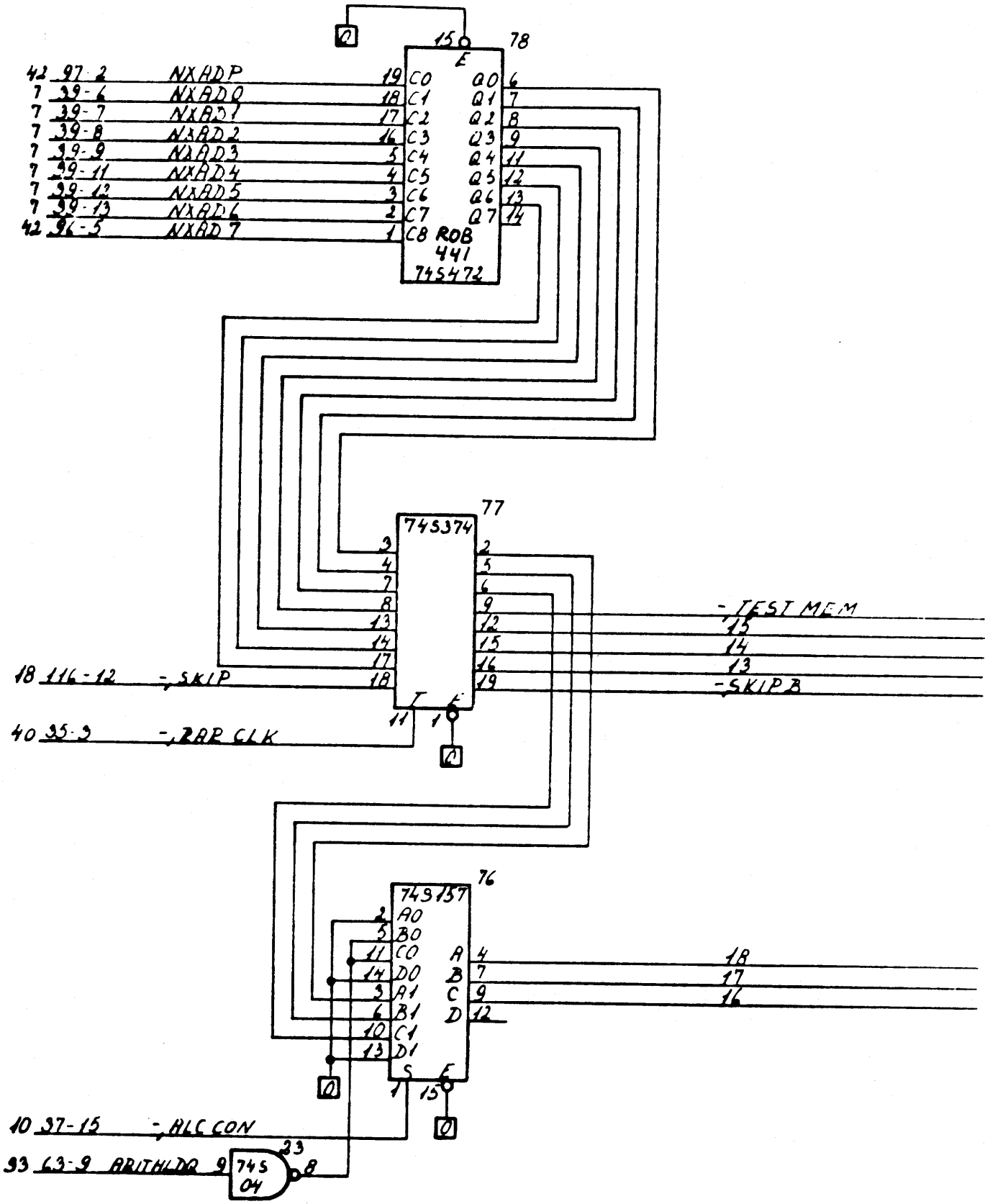


JKA 83.04.13 AGA

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-, BLOCK BIT 0		p. 42	-,BLOCK BIT 0 Forces a logical zero on bit 0 from the AM2901 array.
-, FETCH		p. 3 p. 37 p. 40 p. 31	-, FETCH Indicates that the CPU is reading an instruction from memory.
-, GATE IN		p. 30	-, GATE INput Gates the contents on the I/O Data BUS out on the internal bus.
-, GATE OUT		p. 40	-, GATE OUTput Gates the contents on the internal bus out on the I/O Data BUS.
I 0-2		p. 1 p. 2	micro Instruction bit 0-2 ALU source operand controls.
-, RESTART ADDRESS		p. 34 p. 37	-, RESTART ADDRESS Gates the contents of the Data Switches (TCP 701) out on the internal data bus and sets the CPU into the RUN state.


Ø
15
E
68
42 97-2 NXADP
7 39-6 NXAD0
7 39-7 NXAD1
7 39-8 NXAD2
7 39-9 NXAD3
7 39-11 NXAD4
7 39-12 NXAD5
7 39-13 NXAD6
42 96-5 NXAD7
19 C0
18 C1
17 C2
16 C3
5 C4
4 C5
3 C6
2 C7
1 C8
ROM 853
74S472
Q0 6
Q1 7
Q2 8
Q3 9
Q4 11
Q5 12
Q6 13
Q7 14
67
74S374
9
4
7
8
13
14
17
18
2
5
6
9
12
15
16
19
-BLOCK BIT 0
-RESTART ADDRESS
-GATE OUT
-GATE IN
12
11
10
-FETCH
T
E
11
1
Ø
40 95-3 -RAR CLK
JKA AGA 83 04 13

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
I 3-4		p. 1 p. 2	micro Instruction bit 3-4 ALU function control.
I 6-8		p. 1 p. 2	micro Instruction bit 6-8 ALU result destination control.
; SKIP B		p. 17	; SKIP Buffer Output from the SKIP condition Generator synchronized with the microprogram.
; TEST MEM		p. 19	; TEST MEMory Used to test the state of the Memory Busy status.



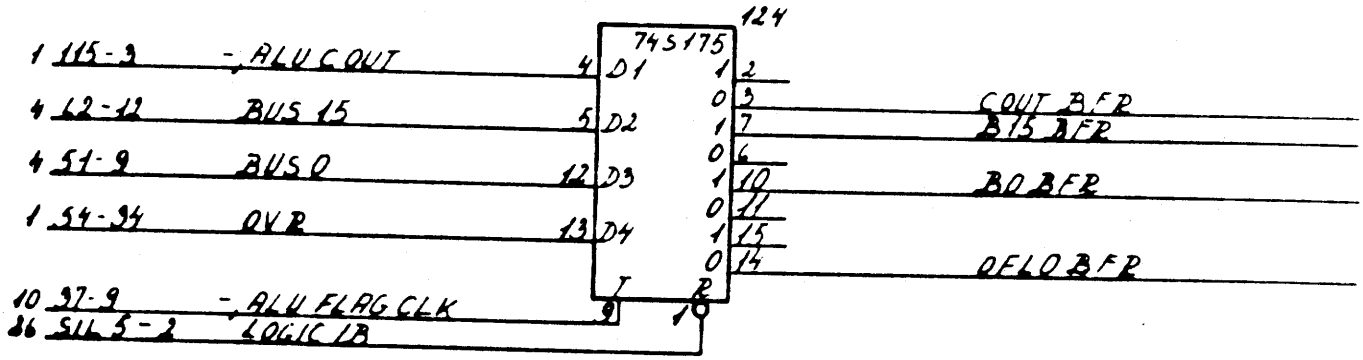
JKA 83 04 13 AGA

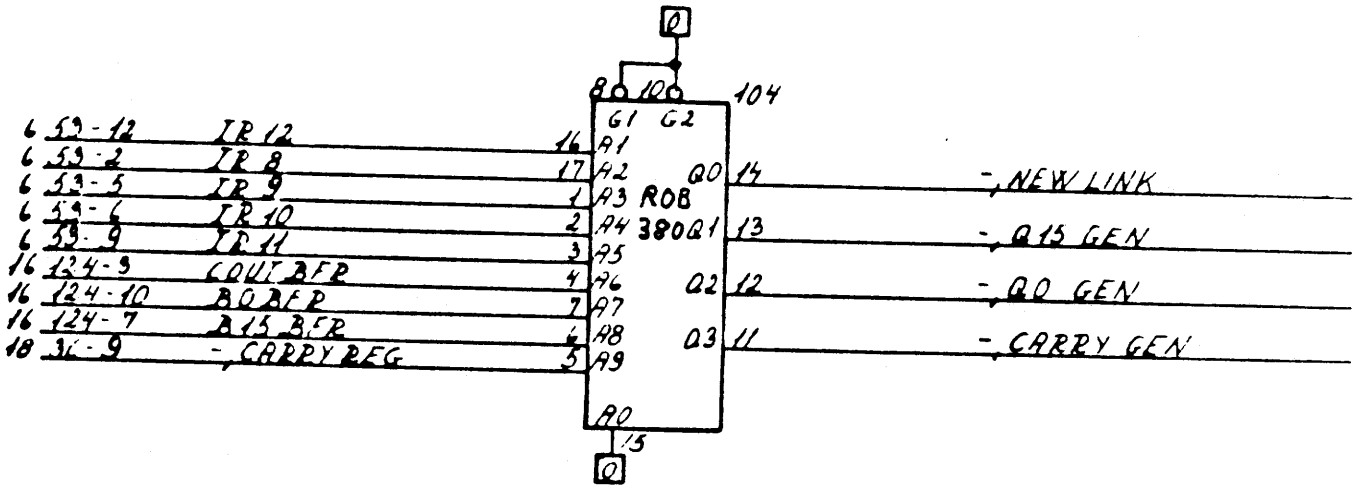
CPU 721

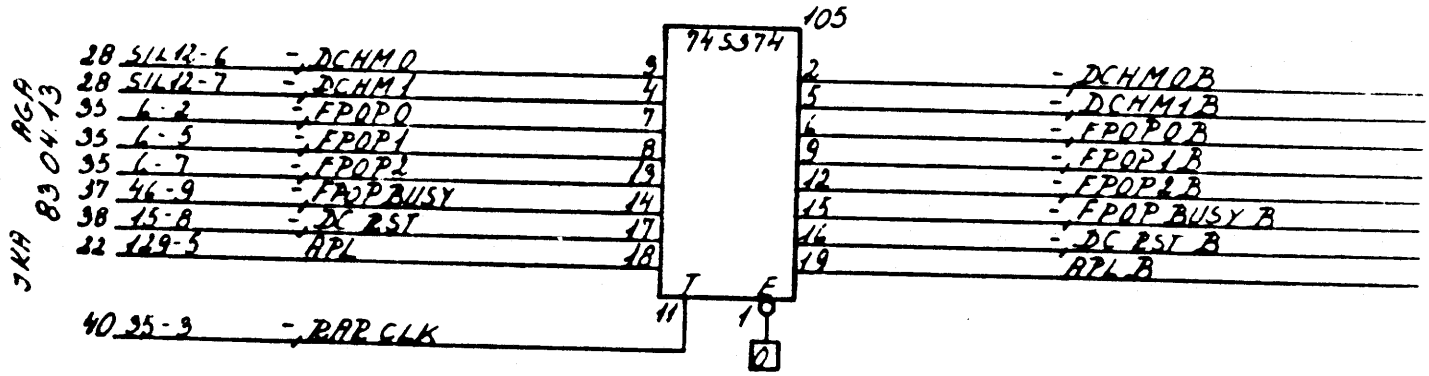
R 13588

MICROPROGRAM STORE
Logic Diagram

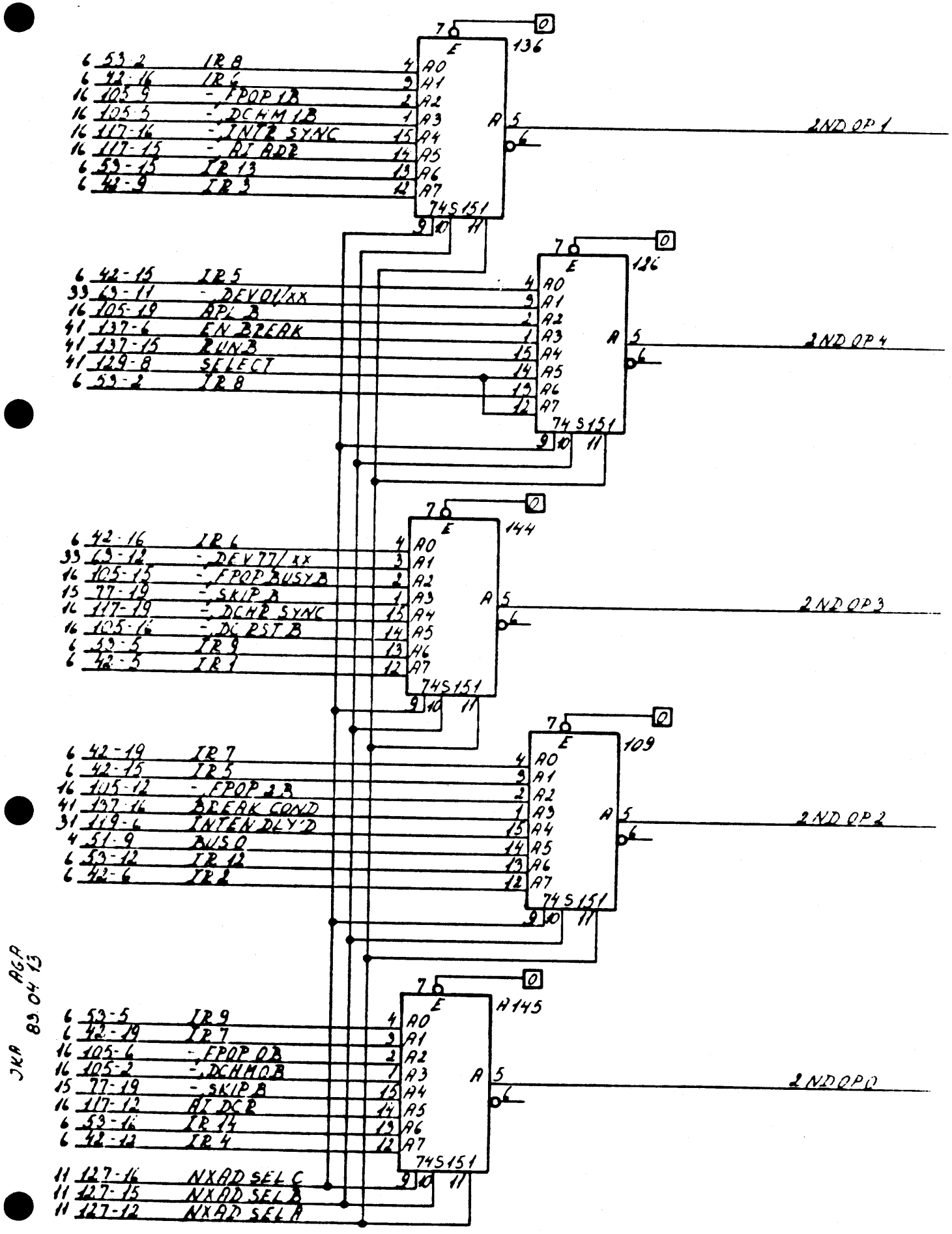
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
APL B		p. 17	Automatic Program Load Buffered
B0 BFR		p. 16 p. 18	Bus bit 0 BuFfeRed
B15 BFR		p. 16	Bus bit 15 BuFfeRed
COUT BFR		p. 16	Carry OUTput BuFfeRed
⁻; CARRY GEN		p. 18	⁻; CARRY GENerator
⁻; DCHM 0-1 B		p. 17	⁻; Data CHannel Mode bit 0-1 Buffered
⁻; DC RST B		p. 17	⁻; DC ReSeT Buffered
⁻; FPOP 0-2 B		p. 17	⁻; Front Panel OPeration bit 0-2 Buffered
⁻; FPOP BUSY B		p. 17	⁻; Front Panel OPeration BUSY Buffered
⁻; NEW LINK		p. 18	⁻; NEW LINK
OFLO BFR		p. 33	OverFLOw status BuFfeRed
⁻; Qo GEN		p. 18	⁻; Qo shift GENerator
⁻; Q_{15} GEN		p. 18	⁻; Q_{15} shift GENerator


124
74S175
1 115-3 -ALU COUT
4 62-12 BUS 15
4 51-9 BUS 0
1 54-34 OVR
10 37-9 -ALU FLAG CLK
86 SIL 5-2 LOGIC 1B
COUT BFR
B15 BFR
B0 BFR
OFLO BFR

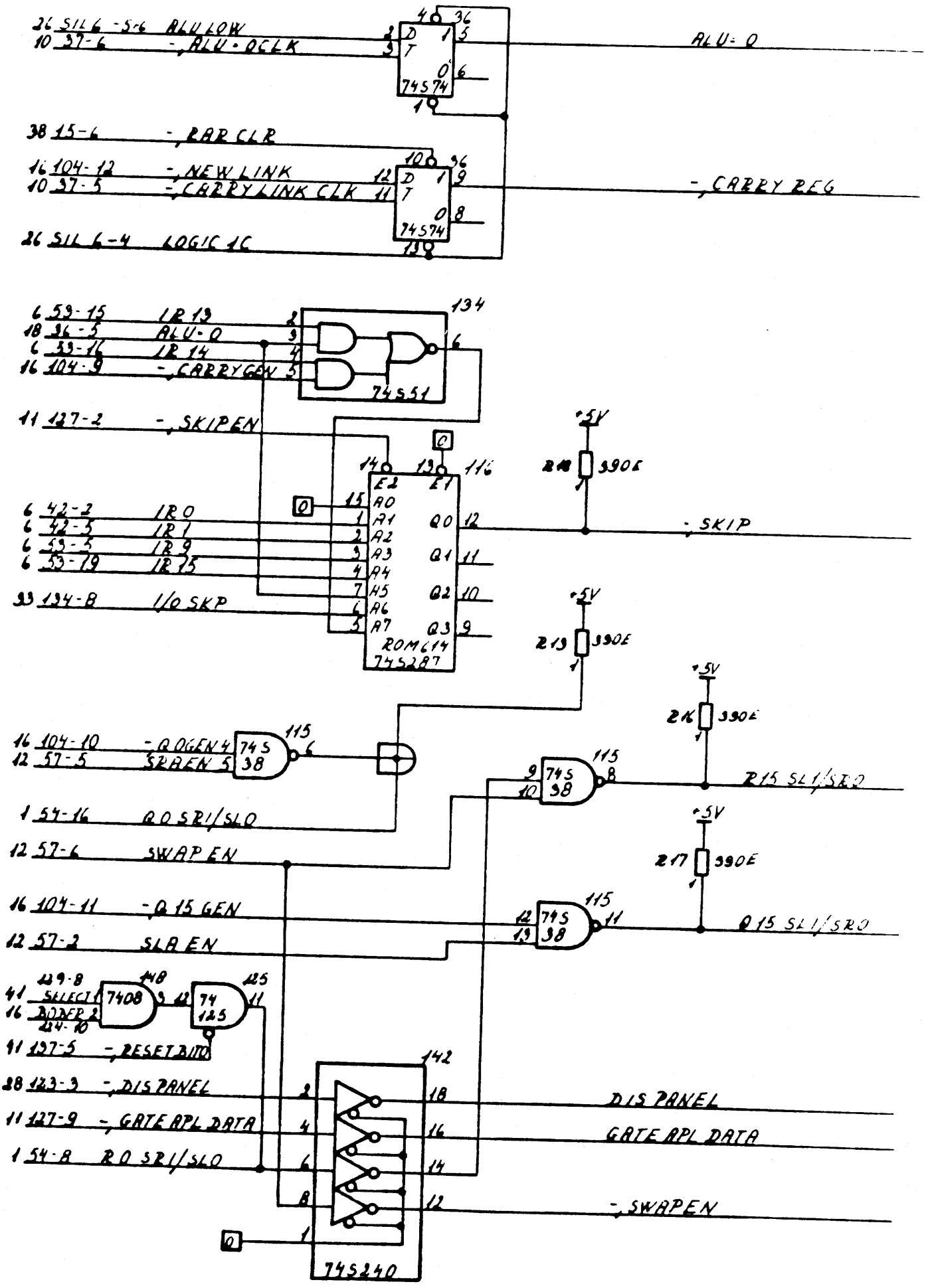

104
ROB
380Q1
6 59-12 IR 12
6 59-2 IR 8
6 59-5 IR 9
6 59-6 IR 10
6 59-9 IR 11
16 124-9 COUT BFR
16 124-10 B0 BFR
16 124-7 B15 BFR
18 3C-9 -CARRY REG
-NEW LINK
-Q15 GEN
-Q0 GEN
-CARRY GEN


105
74S374
28 SIL 12-6 -DCHM0
28 SIL 12-7 -DCHM1
35 6-2 -FPOP0
35 6-5 -FPOP1
35 6-7 -FPOP2
37 46-9 -FPOP BUSY
38 15-A -DC RST
22 129-5 APL
40 25-9 -RAR CLK
-DCHM0B
-DCHM1B
-FPOP0B
-FPOP1B
-FPOP2B
-FPOP BUSY B
-DC RST B
APL B

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
2ND OP 0-4		p. 8	2ND OPerand bit 0-4 Outputs from the condition selector; used as the five least significant address inputs to the conditional next address table.


136
6 53-2 IR 8
6 42-16 IR 6
16 105-9 - FPOP 1B
16 105-5 - DCHM 1B
16 117-16 - INTR SYNC
16 117-15 - AI ADR
6 53-13 IR 13
6 42-9 IR 3
74S151
2ND OP 1
126
6 42-15 IR 5
33 43-11 - DEVOUTXX
16 105-19 APL B
41 137-6 EN BREAK
41 137-15 RUN B
41 129-8 SELECT
6 53-2 IR 8
2ND OP 4
144
6 42-16 IR 6
33 43-12 - DEV77/XX
16 105-13 - FPOP BUSY B
15 77-19 - SKIP B
16 117-19 - DCHR SYNC
16 105-16 - DC RST B
6 53-5 IR 9
6 42-3 IR 1
2ND OP 3
109
6 42-19 IR 7
6 42-15 IR 5
16 105-12 - FPOP 2 B
41 137-16 BREAK COND
31 119-6 INTEN DLY'D
4 51-9 BUS 0
6 53-12 IR 12
6 42-6 IR 2
2ND OP 2
H145
6 53-5 IR 9
6 42-19 IR 7
16 105-6 - FPOP 0B
16 105-2 - DCHM 0B
15 77-19 - SKIP B
16 117-12 AI DCR
6 53-16 IR 14
6 42-13 IR 4
2NDOPD
11 127-16 NXAD SEL C
11 127-15 NXAD SEL B
11 127-12 NXAD SEL A
JKA 83 04 13 AGP

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
ALU = 0		p. 18 p. 42	ALU = 0 Zero status from the ALU.
~ CARRY REG		p. 16 p. 40	~ CARRY REGister The Carry Out status register.
DIS PANEL		p. 34 p. 35 p. 38 p. 7	DISable PANEL Indicates the state of the ENABLE TCP switch.
Q_{15} SLI/SRO		p. 2	Q15 Shift Left Input/Shift Right Output Input to Q-shifter.
R_{15} SLI/SRO		p. 2	R15 Shift Left Input/Shift Right Output Input to Register-shifter.
~ SKIP		p. 15	~ SKIP Output from the SKIP condition decoder.
~ SWAP EN		p. 10	~ SWAP ENable Connects Ro SRI/SLO to R_{15} SLI/SRO, and ALU COUT to CIN.
GATE APL DATA		p. 22	GATE Automatic Program Load DATA

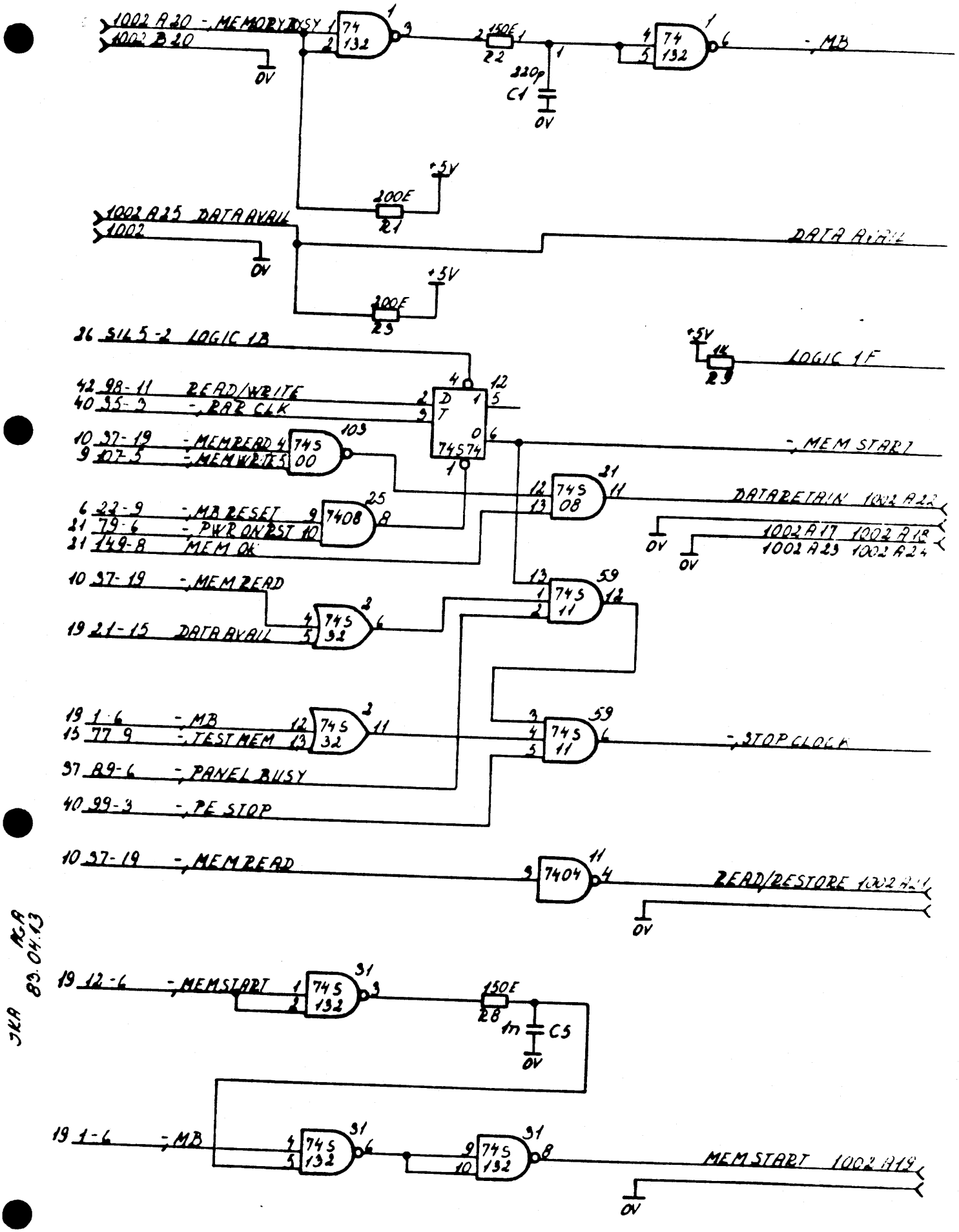


CPU 721

R 13591

ALU·0 F/F and CARRY LINK F/F
SKIP DECODER
SHIFT LEFT/RIGHT SERIAL INPUT
LOGIC DIAGRAM

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
DATA AVAIL		p. 19	DATA AVAILable Indicates that data is ready from memory during a read cycle.
DATA RETAIN		1002	DATA RETAIN Used to disable memory start when not in use.
LOGIC 1E		p. 22 p. 31 p. 37	LOGIC 1E
LOGIC 1F		p. 37 p. 40	LOGIC 1F
⁻; MB		p. 19 p. 6	⁻; Memory Busy Indicates that memory is not ready to accept a new memory start.
MEM START		1002	MEMory START Starts the memory with a Read or Write cycle.
READ/RESTORE		1002	READ/RESTORE Indicates to memory that the requested cycle should be a Read.
⁻; STOP CLOCK		p. 21	⁻; STOP CLOCK Used to stop the microprogram if: Parity Error stop occurs, Memory not ready, or panel busy.



JKA KGA 83.04.13

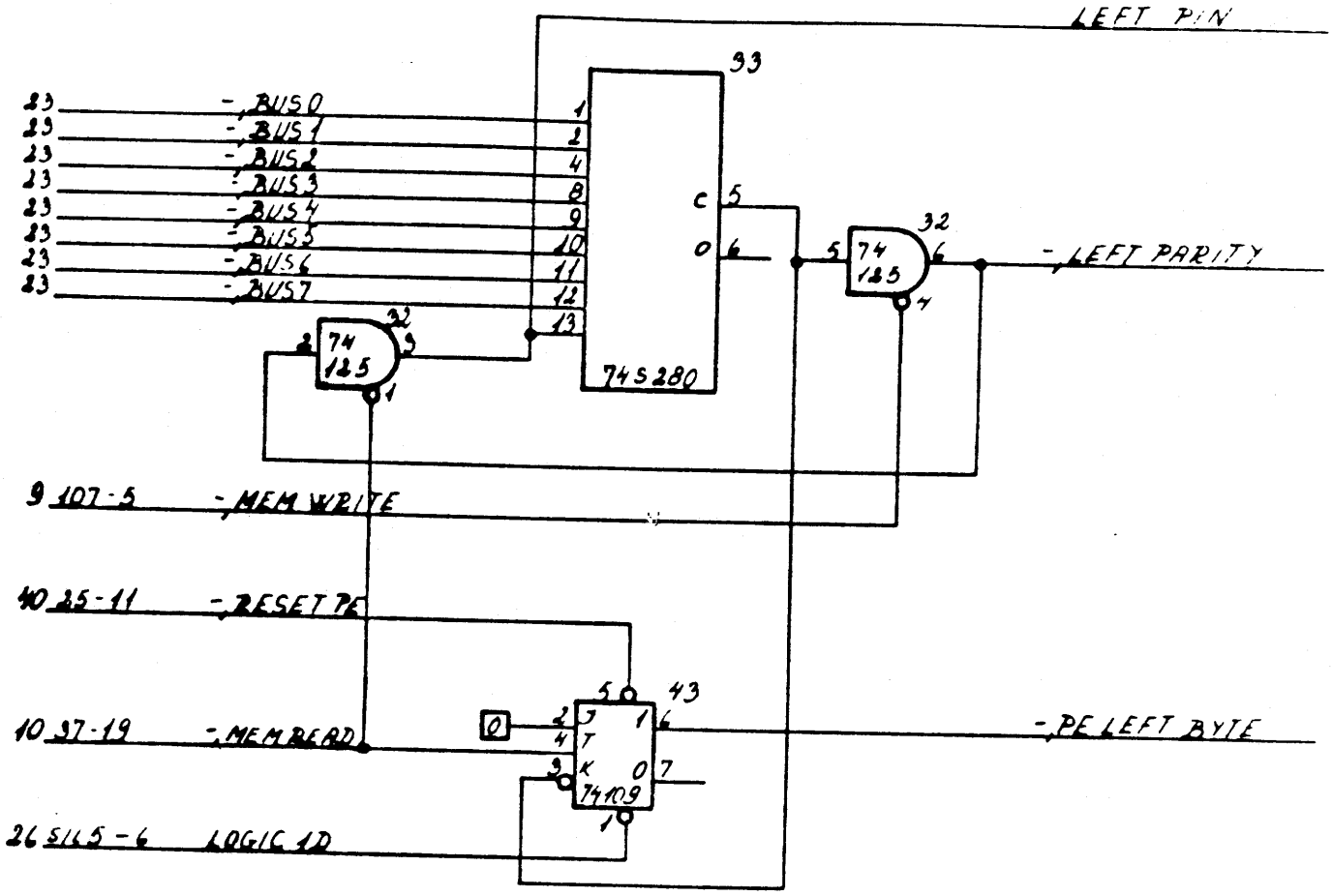
CPU 721

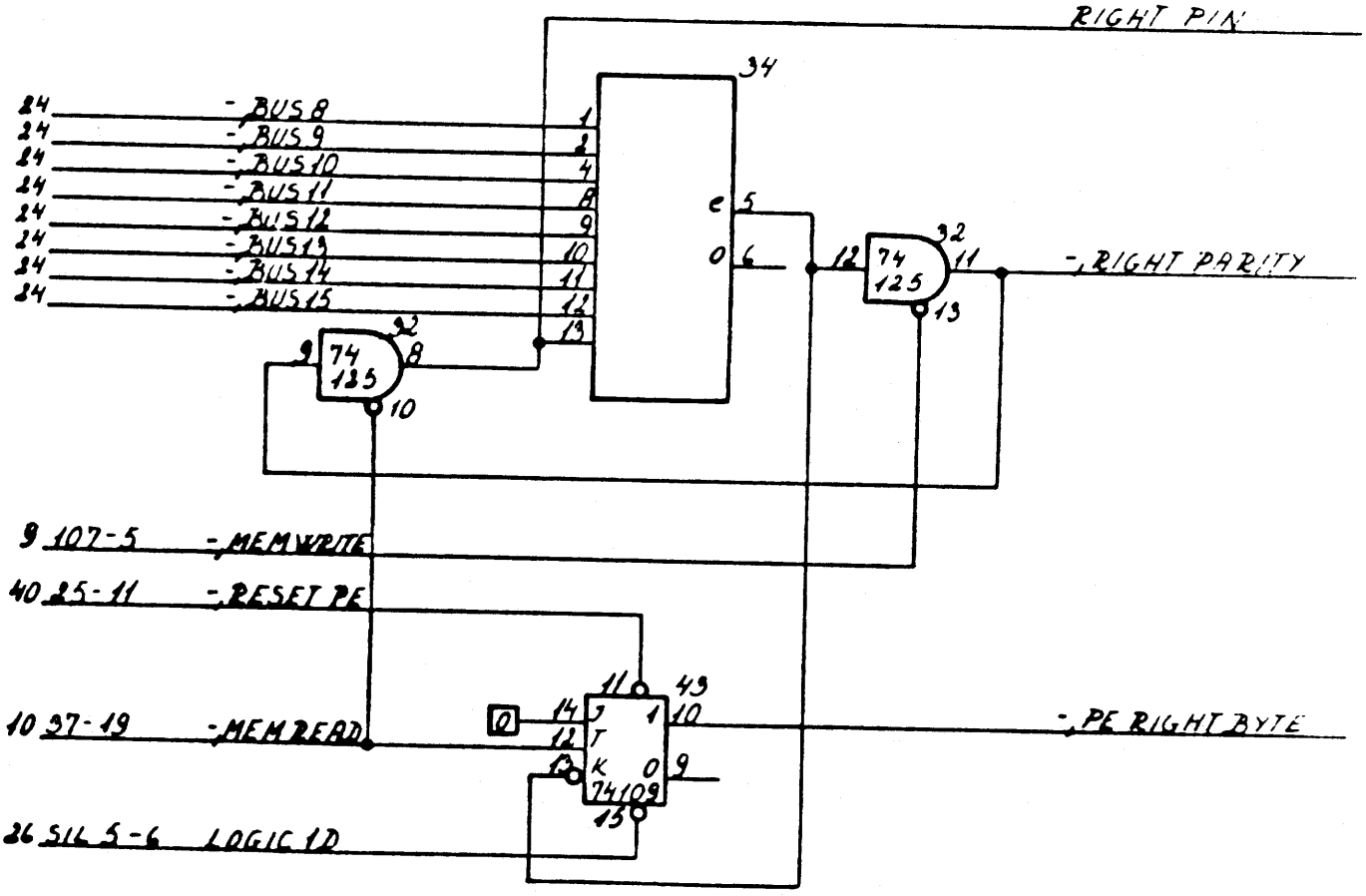
R19592

MEMORY INTERFACE
CONTROL LOGIC
LOGIC DIAGRAM

19

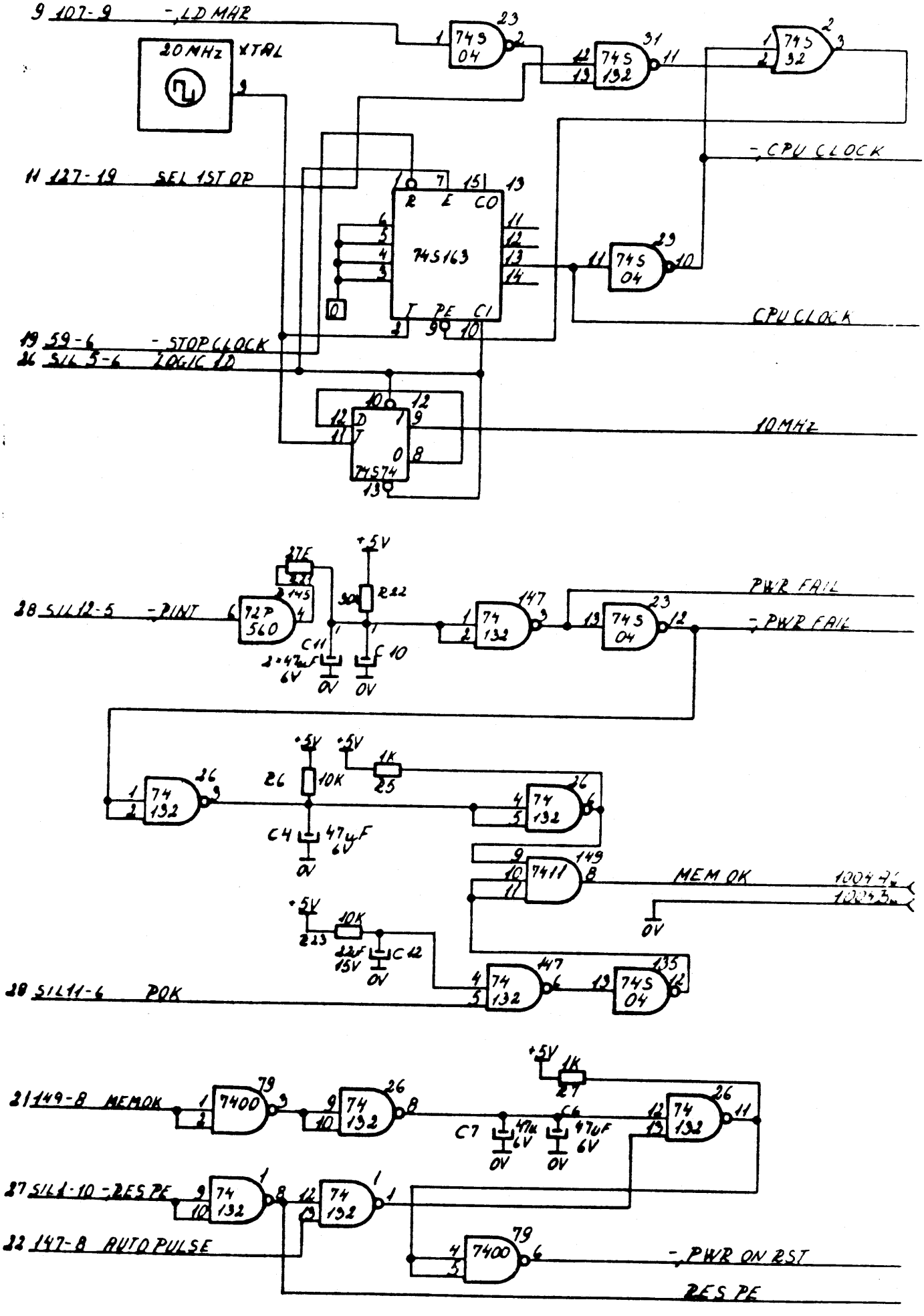
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~; LEFT PARITY		p. 26	~; LEFT PARITY bit Left parity bus to or from memory.
LEFT PIN		p. 26	LEFT Parity INput Received left parity bit during a Read cycle.
~; PE LEFT BYTE		p. 40	~; Parity Error LEFT BYTE LEFT BYTE Parity Error Buffer.
~; PE RIGHT BYTE		p. 40	~; Parity ERROR RIGHT BYTE RIGHT BYTE Parity Error Buffer.
~; RIGHT PARITY		p. 26	~; RIGHT PARITY bit Right parity bus to or from memory.
RIGHT PIN		p. 26	RIGHT Parity INput Received right parity bit during a Read cycle.





JWH AGH 89.04.13

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
CPU CLOCK		p. 37	CPU CLOCK (frequency 6.6 MHz).
-; CPU CLOCK		p. 40	-; CPU CLOCK. Same as above.
LOGIC 1G		p. 33 p. 38	LOGIC 1 G.
MEM OK		p. 19 1004	MEMory power OK.
-; PWR FAIL		p. 33	-; PoWeR FAILure This signal will be the first to go low in the event of a power failure, approximately 1 to 2 msec. before MEM OK.
PWR FAIL		p. 33	
-; PWR ON RST		p. 19 p. 22 p. 35 p. 37 p. 38 p. 40	-; PoWeR ON ReSeT Used to reset the logic circuits after power is turned on.
10 MHZ		p. 35 p. 38 p. 6	10 MHZ clock.
RES PE		p.7	RESet Parity Error. Generates an restart address to the microprogram when the AUTO switch is activated while the Reset Parity Error switch is standing in the reset state.



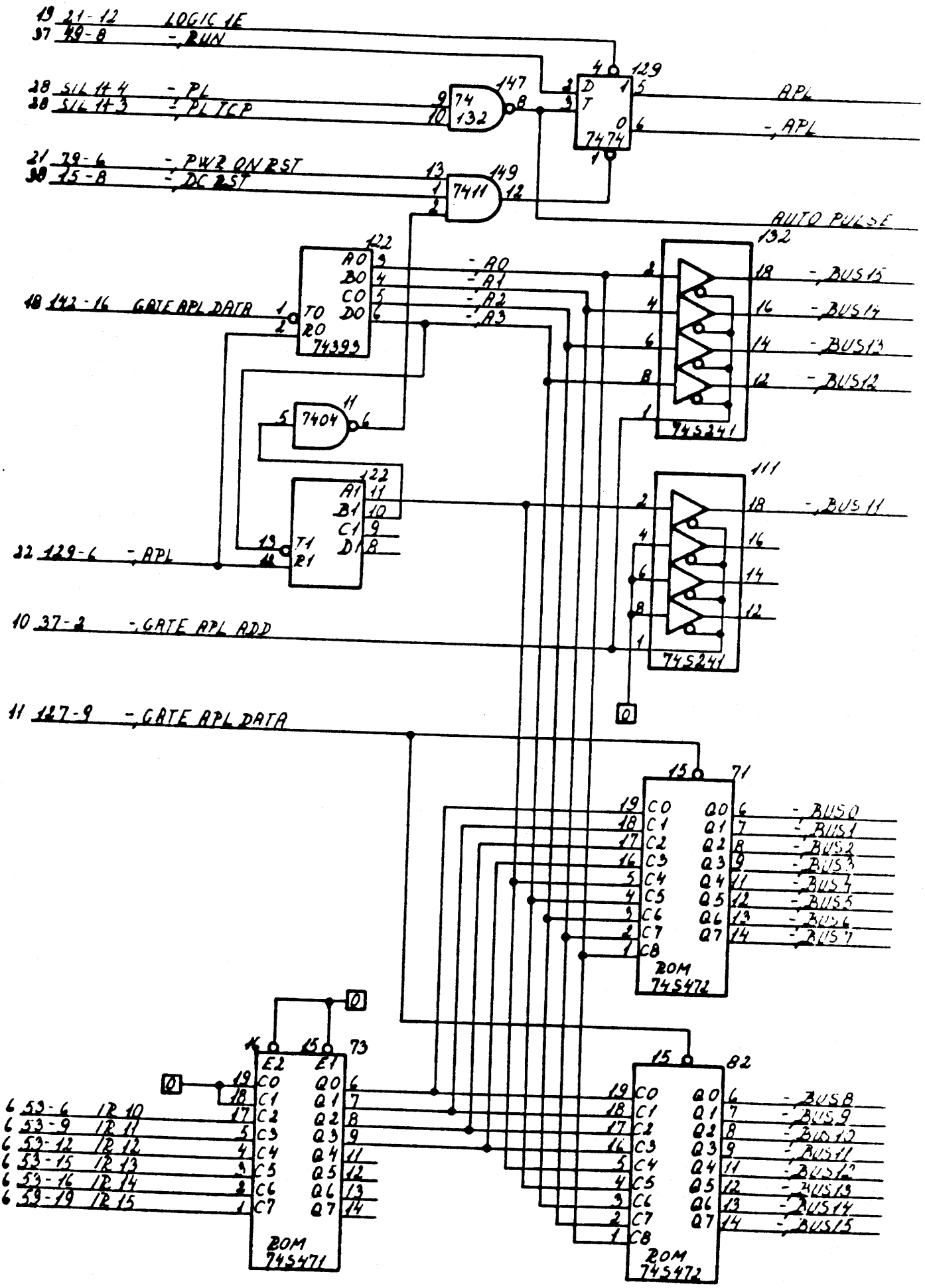
JWA AGA 83.04.13

CPU 721

2 13594

CLOCK GENERATOR & POWER ON RESET
LOGIC DIAGRAM

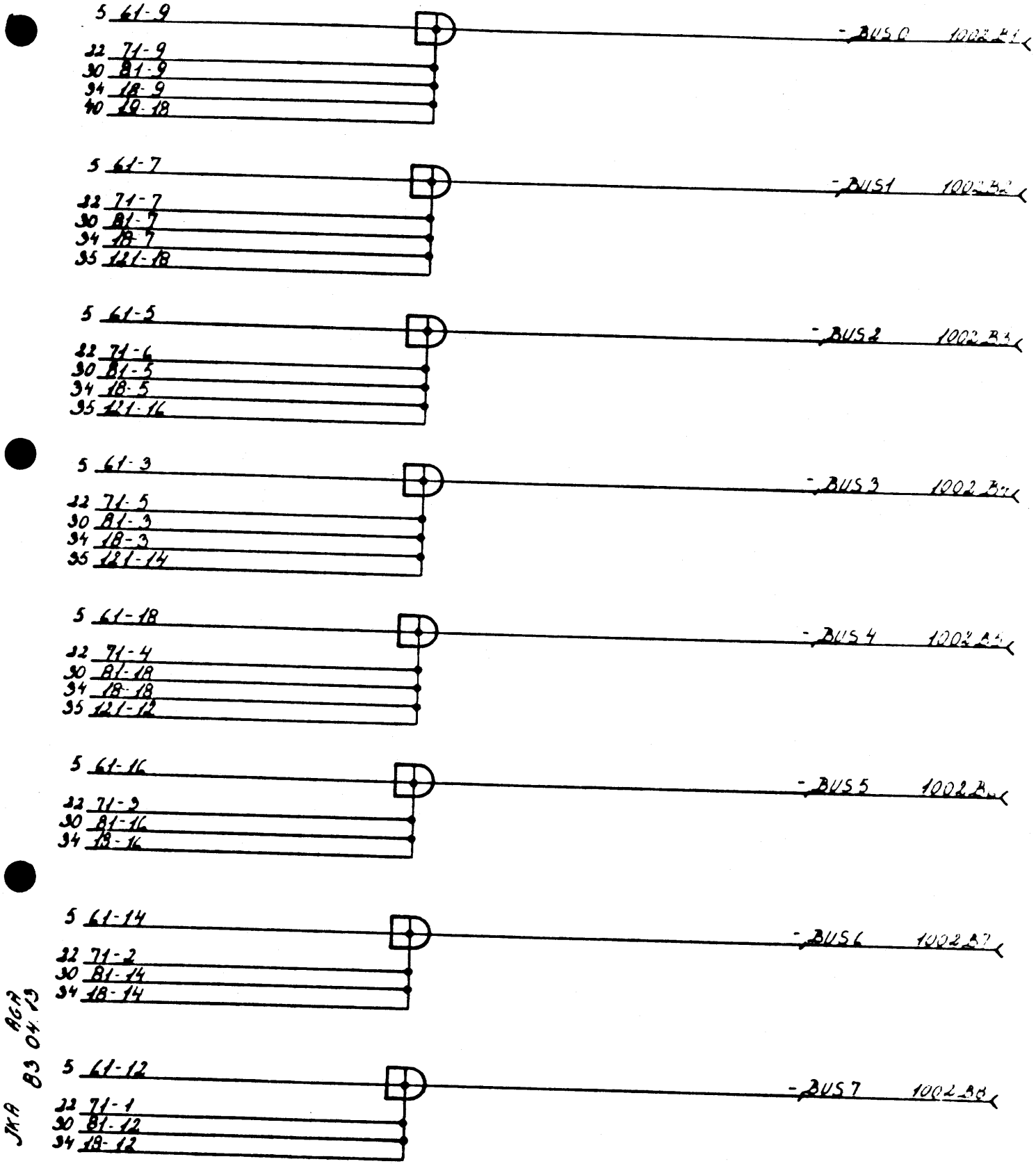
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
APL		p. 16	Automatic Program Load This signal is true the period taken to move the contents of the Auto-load Program proms into main memory location 0-37.
¬ BUS 8-15		p. 24	¬ internal BUS bit 8-15 On this logic Diagram is shown two sources on the internal data bus: 1. The autoload address counter. 2. The output from the Autoload Program memory (Right Byte).
¬ BUS 0-7		p. 23	¬ internal BUS bit 0-7 Output from the Autoload Program memory (Left Byte).
AUTO PULSE		p. 21	AUTO PULSE



CPU 721

AUTOMATICAL PROGRAM LOAD

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻, BUS 0-7		p. 3 p. 4 p. 20 p. 25 p. 30 p. 36 1002	⁻, internal BUS bit 0-7 The destination for the internal data bus is the ⁻, (BUS 1-10) = 0 Decoder, BUS inverter, parity checker/generator, I/O Data Bus drivers, and the Diagnostic Panel data bus.

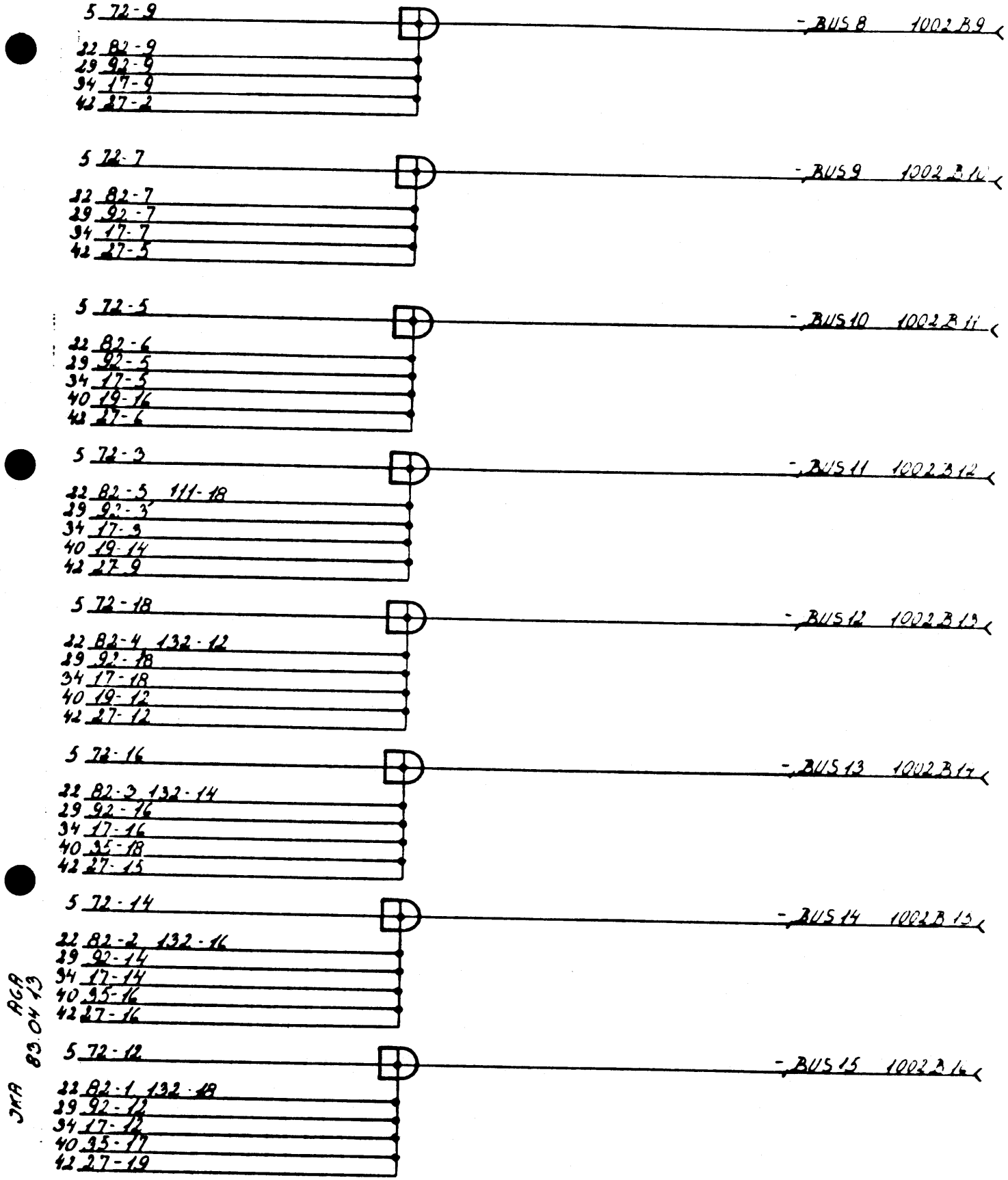


AGA
B3 04 13
JKA

CPU 721
R 13596

BUS WIRE OR BIT 0-7
LOGIC DIAGRAM

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻, BUS 8-10		p. 3 p. 4 p. 20 p. 25 p. 29 p. 36 1002	⁻, Internal BUS bit 8-15 The destination for the internal data bus is the ⁻, (BUS 1-10) = 0 Decoder, BUS inverter, parity checker/generator, I/O Data Bus drivers and the Diagnostic Panel data bus.
⁻, BUS 11-15		p. 4 p. 20 p. 25 p. 29 p. 36 1002	



AGA 83.04.13

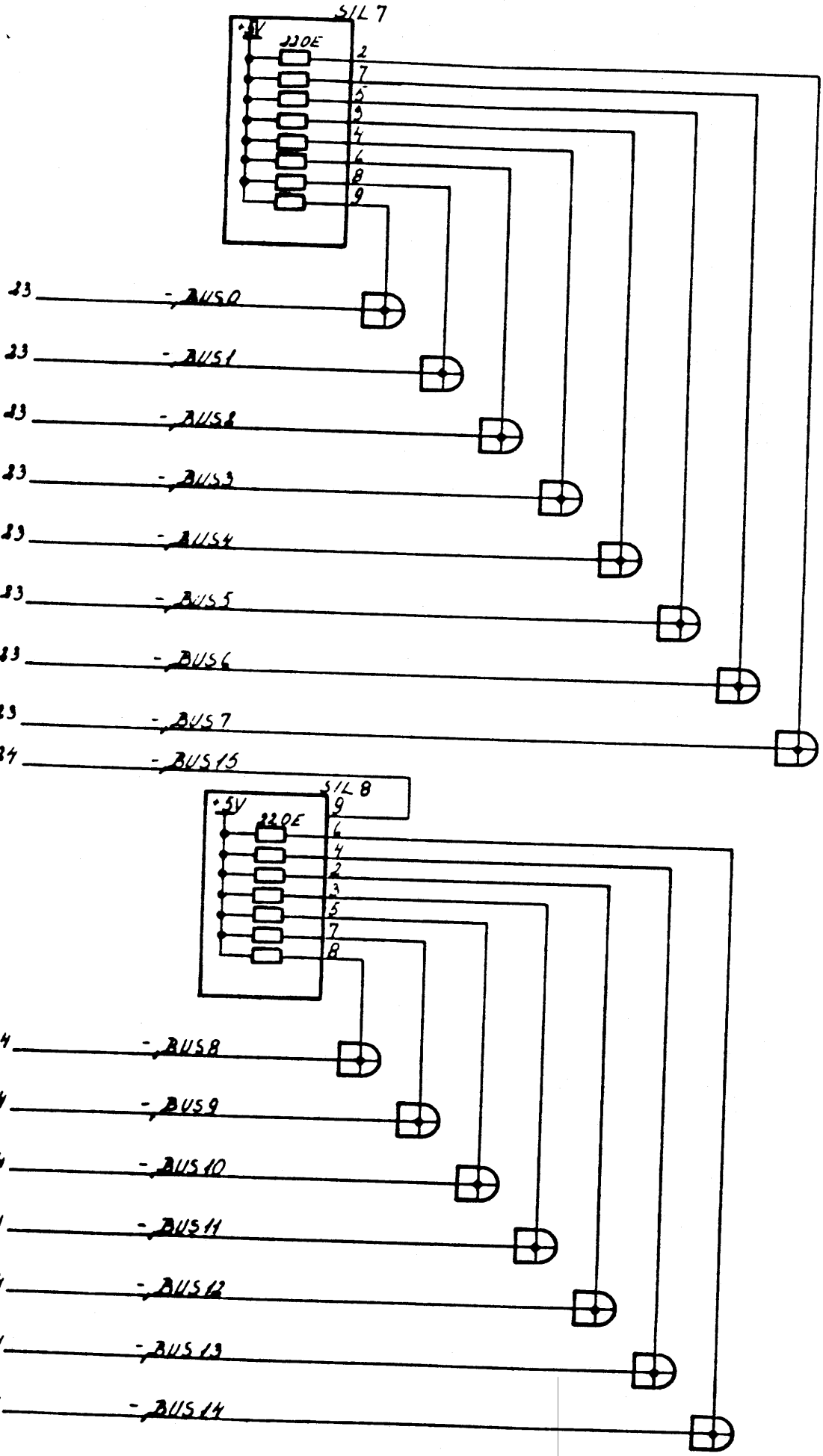
JKA

CPU 721

R 13597.

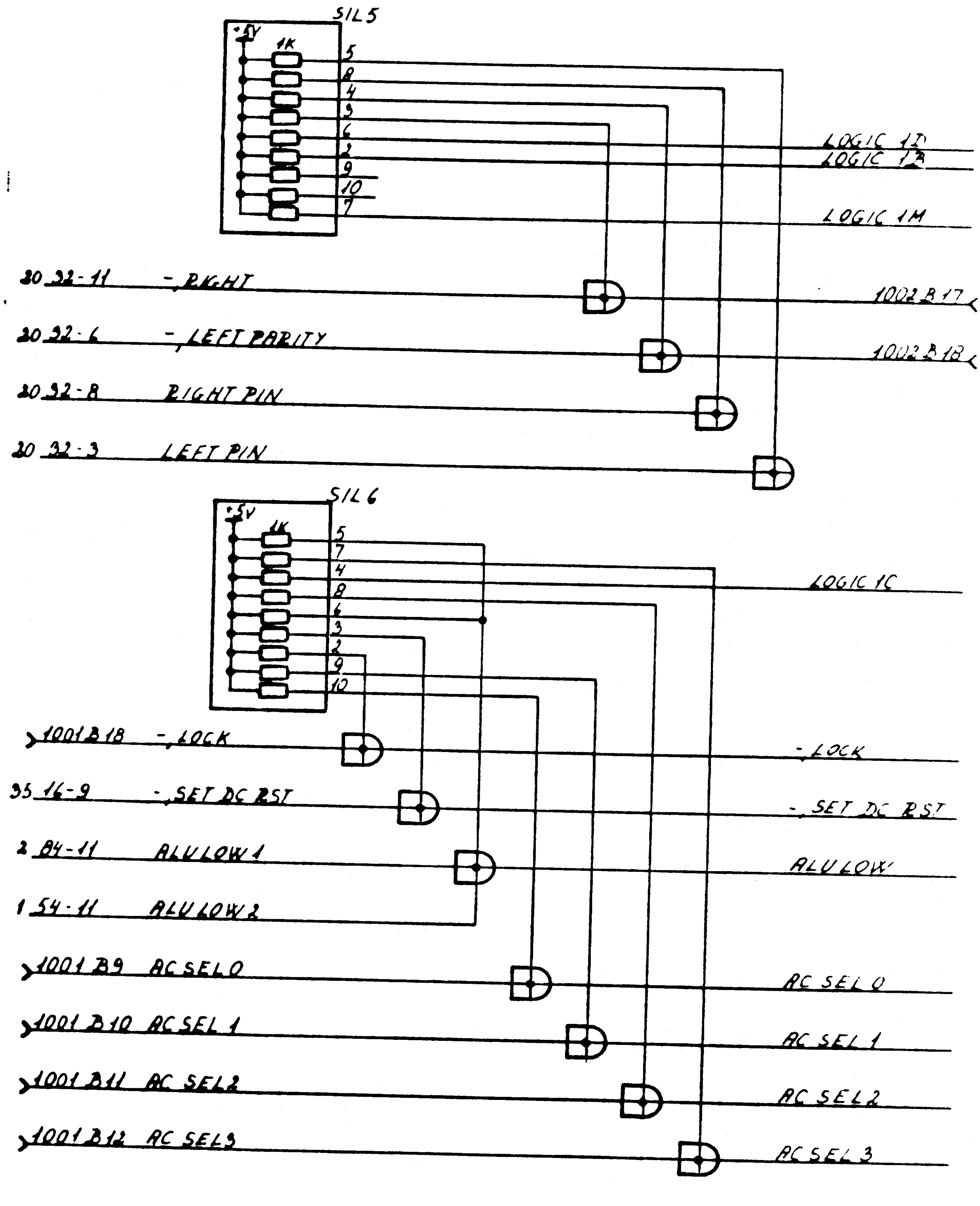
BUS WIRE OR BIT 8-15
LOGIC DIAGRAM

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION

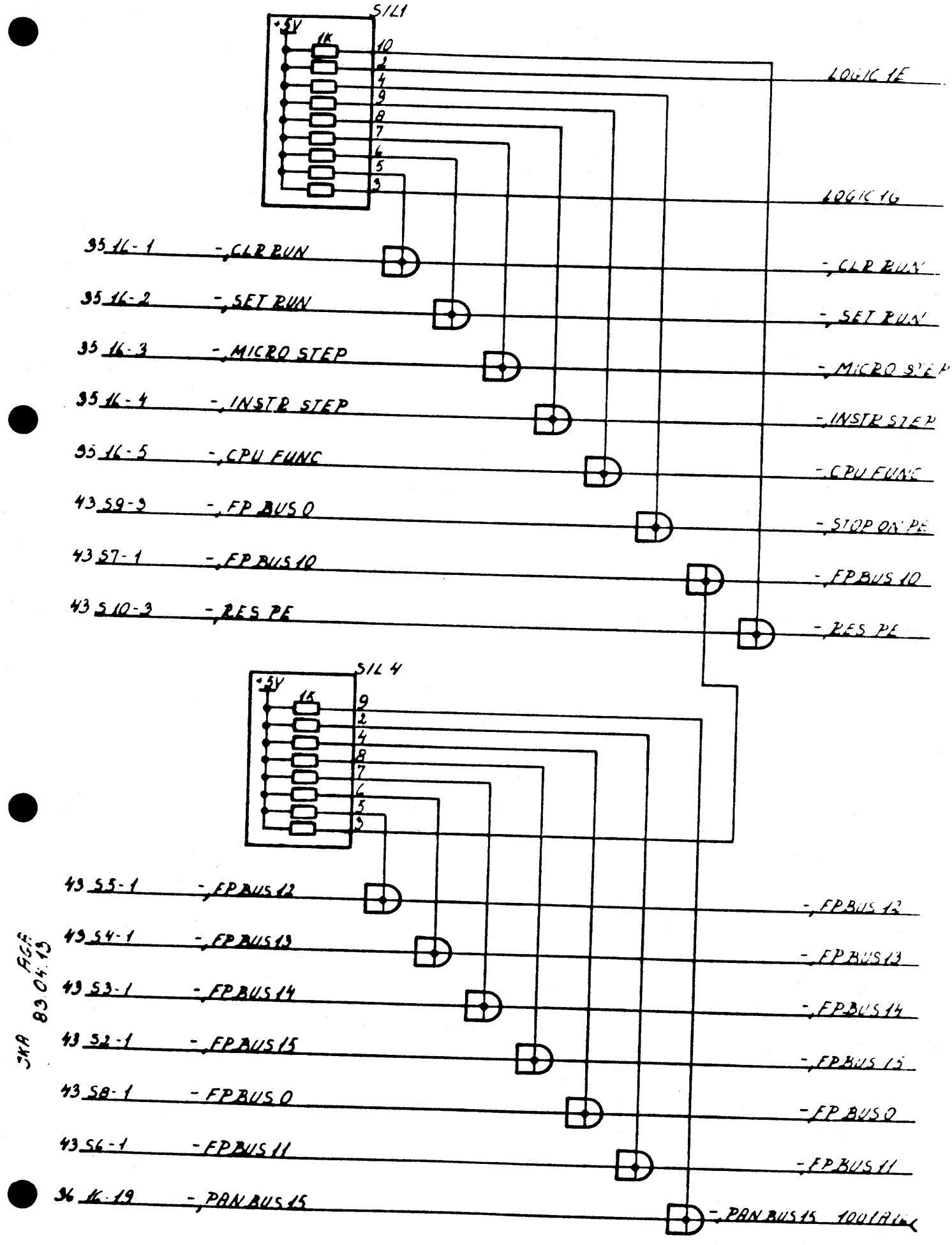


AGA 83.04.13 JMH

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
AC SEL 0-3		p. 35 1001	ACcumulator SELect bit 0-3 Register stack address from the Diagnostic Panel TCP 701.
ALU LOW		p. 18 p. 41 p. 42	ALU LOW Indicates that the output from the ALU contains all zeroes.
⁻; LEFT PARITY		1002	⁻; LEFT PARITY bit Left parity bus to or from memory.
⁻; LOCK		p. 7	⁻; LOCK The state of the POWER ON/LOCK switch on the Diagnostic Panel. If this switch is not mounted, the ⁻; LOCK signal is always logic high.
LOGIC 1 B		p. 16 p. 19 p. 31 p. 32	LOGIC 1 B.
LOGIC 1 C		p. 18	LOGIC 1 C.
LOGIC 1 D		p. 20 p. 21	LOGIC 1 D.
⁻; RIGHT PARITY		1002	⁻; RIGHT PARITY bit Right parity to or from memory.
⁻; SET DC RST		p. 38	⁻; SET DC ReSeT Used to generate IORST.
⁻,PAN BUS 15		p. 27	

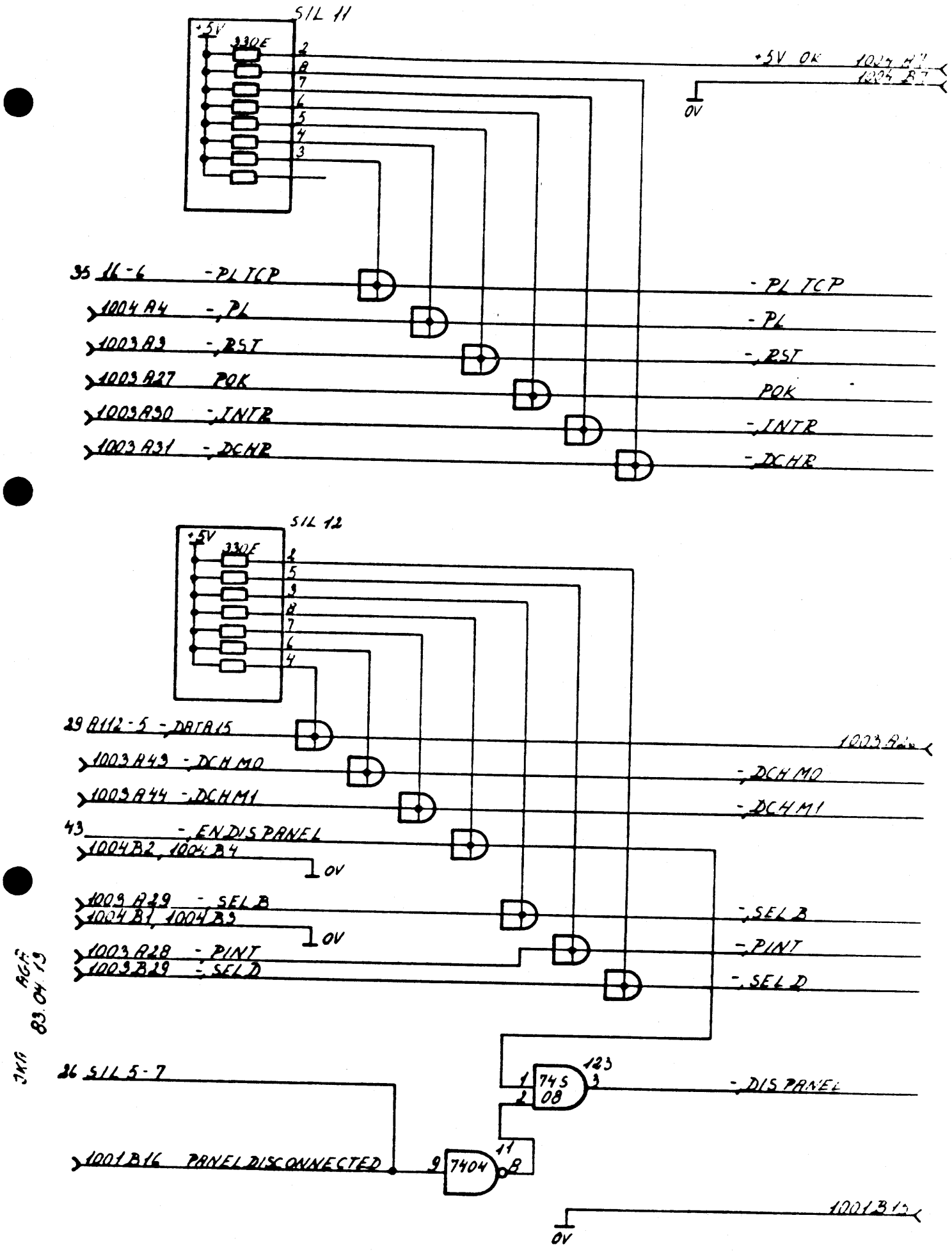

SIL5
+5V
1K
5
8
4
3
6
2
9
10
7
LOGIC 1D
LOGIC 1B
LOGIC 1M
20 32-11 -RIGHT
1002 B17
20 32-6 -LEFT PARITY
1002 B18
20 32-8 RIGHT PIN
20 32-3 LEFT PIN
SIL6
+5V
1K
5
7
4
8
6
3
2
9
10
LOGIC 1C
1001 B18 -LOCK
-LOCK
35 16-9 -SET DC RST
-SET DC RST
2 B4-11 ALU LOW 1
ALU LOW
1 54-11 ALU LOW 2
1001 B9 AC SEL 0
AC SEL 0
1001 B10 AC SEL 1
AC SEL 1
1001 B11 AC SEL 2
AC SEL 2
1001 B12 AC SEL 3
AC SEL 3
JYF 83 04 13

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
; CLR RUN		p. 37	; CLeaR RUN Used to reset the RUN flip flop.
; CPU FUNC		p. 37	; CPU FUNCtion Indicates that the started Diagnostic Panel operation requires microprogram execution.
; FP BUS 0, 10-15		p. 40 p. 39	; Front Panel BUS bit 0, 10-15 Autoload information used when Diagnostic Panel is not connected or not enabled.
; INSTR STEP		p. 37	; INSTRuction STEP.
; MICRO STEP		p. 37	; MICROprogram STEP.
; PAN BUS 15		1001	; PANel data BUS 15.
; RES PE		p. 40 p. 39 p. 21	; RESet Parity Error Used to clear the Parity Error indicator register.
; SET RUN		p. 37	; SET RUN Sets the RUN flip flop.
; STOP ON PE		p. 40 p. 39	; STOP ON Parity Error Switches input from the CPU 708 Front Panel, determining whether the CPU should both indicate parity error and stop the microprogram execution, or the CPU should only indicate parity errors.



JKA 83 04 15 FGR

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~ DATA 15		1003	~ I/O DATA bus bit 15.
~ DCHM 0-1		p. 16	~ Data CHannel Mode bit 0-1.
~ DCHR		p. 11	~ Data CHannel Request.
~ DIS PANEL		p. 18 p. 34 p. 35 p. 37 p. 40 p. 41	~ DISable diagnostic PANEL.
~ INTR		p. 11 p. 33	~ INTerrupt Request.
~ PINT		p. 21	~ Power INTerrupt.
~ PL		p. 22	~ set Program Load.
~ PL TCP		p. 22	~ set Program Load from TCP.
POK		p. 21	Power OK.
~ RST		p. 35	~ ReSeT External signal to simulate the RESET button on the Diagnostic Panel.
~ SEL D		p. 33 p. 41	~ SELect Done The state of the DONE flip flop in the addressed I/O Device.
~ SEL B		p. 33	~ SELect Busy The state of the BUSY flip flop in the addressed I/O Device.

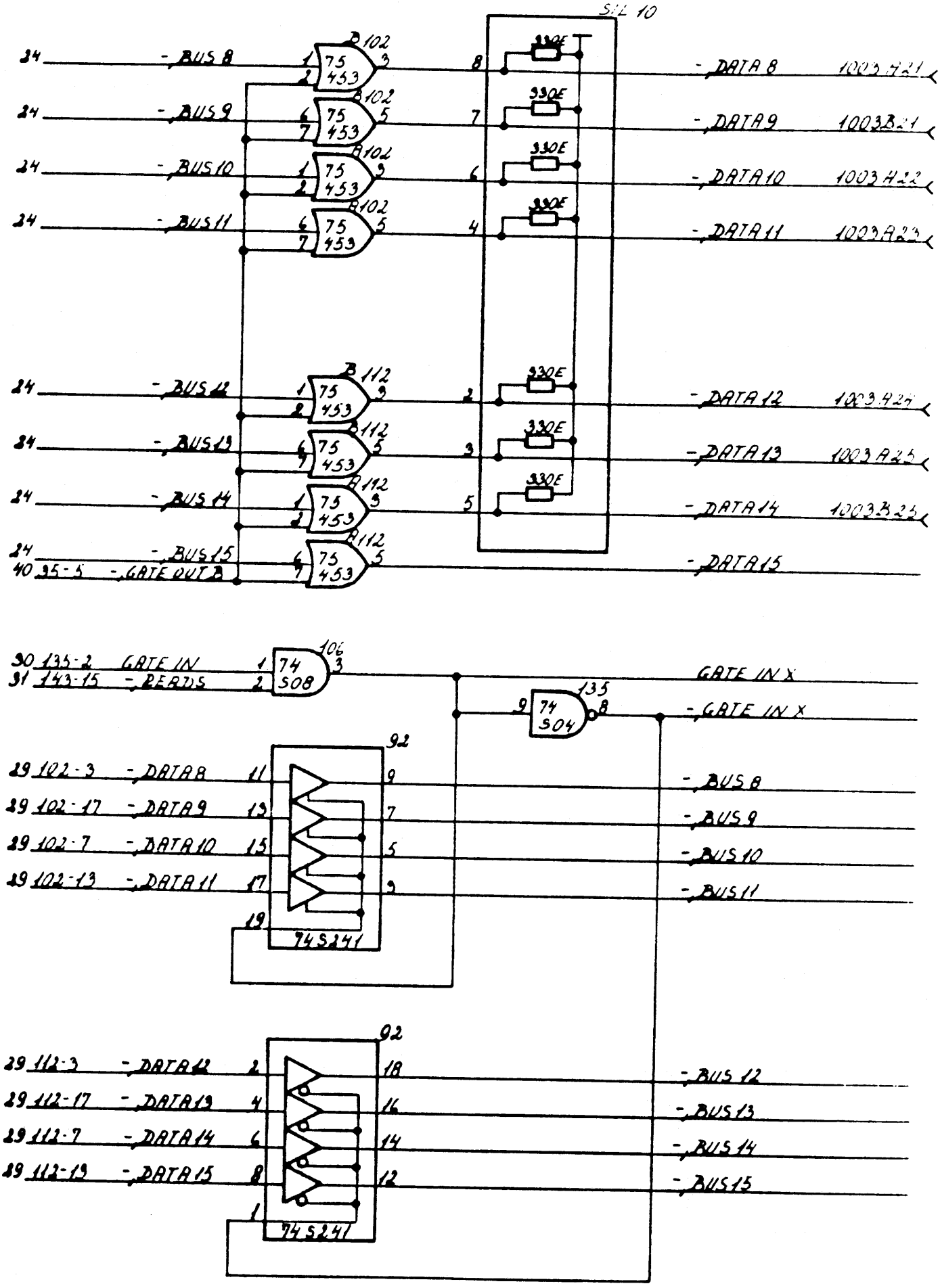


AGF 83.04.13

CPU 721
R 13601

330E PULL UP CIRCUITS
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-; BUS 8-15		p. 24	-; internal BUS bit 8-15 The shown source on the internal BUS is the I/O Data Bus receivers.
-; DATA 8-14		1003	-; I/O DATA BUS bit 8-14.
-; DATA 15		p. 28	-; I/O DATA BUS bit 15.
GATE INX		p. 30	GATE INput Gates the contents on the I/O Data Bus out on the internal bus.
-; GATE INX		p. 30	



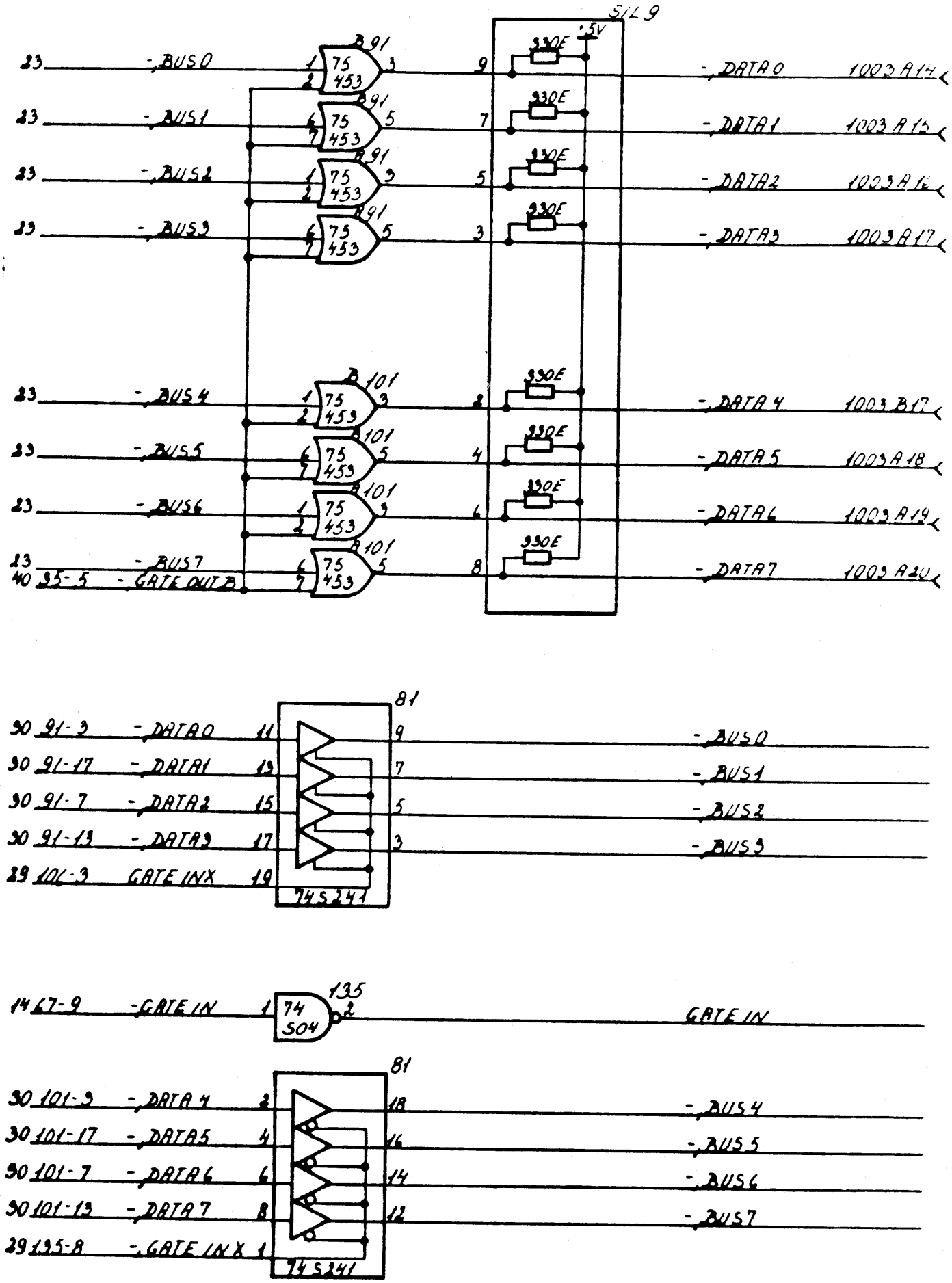
JKH AGA 83 04 13

CPU 721

1 2 13602

DATA BUS DRIVERS AND RECEIVERS
BIT 8-15
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~; BUS 0-7		p. 23	~; internal BUS bit 0-7 The shown source on the internal BUS is the I/O Data Bus Receivers.
~; DATA 0-7		1003	~; I/O DATA bus bit 0-7.
GATE IN		p. 29	GATE INput Gates the contents on the I/O Data Bus out on the internal bus.

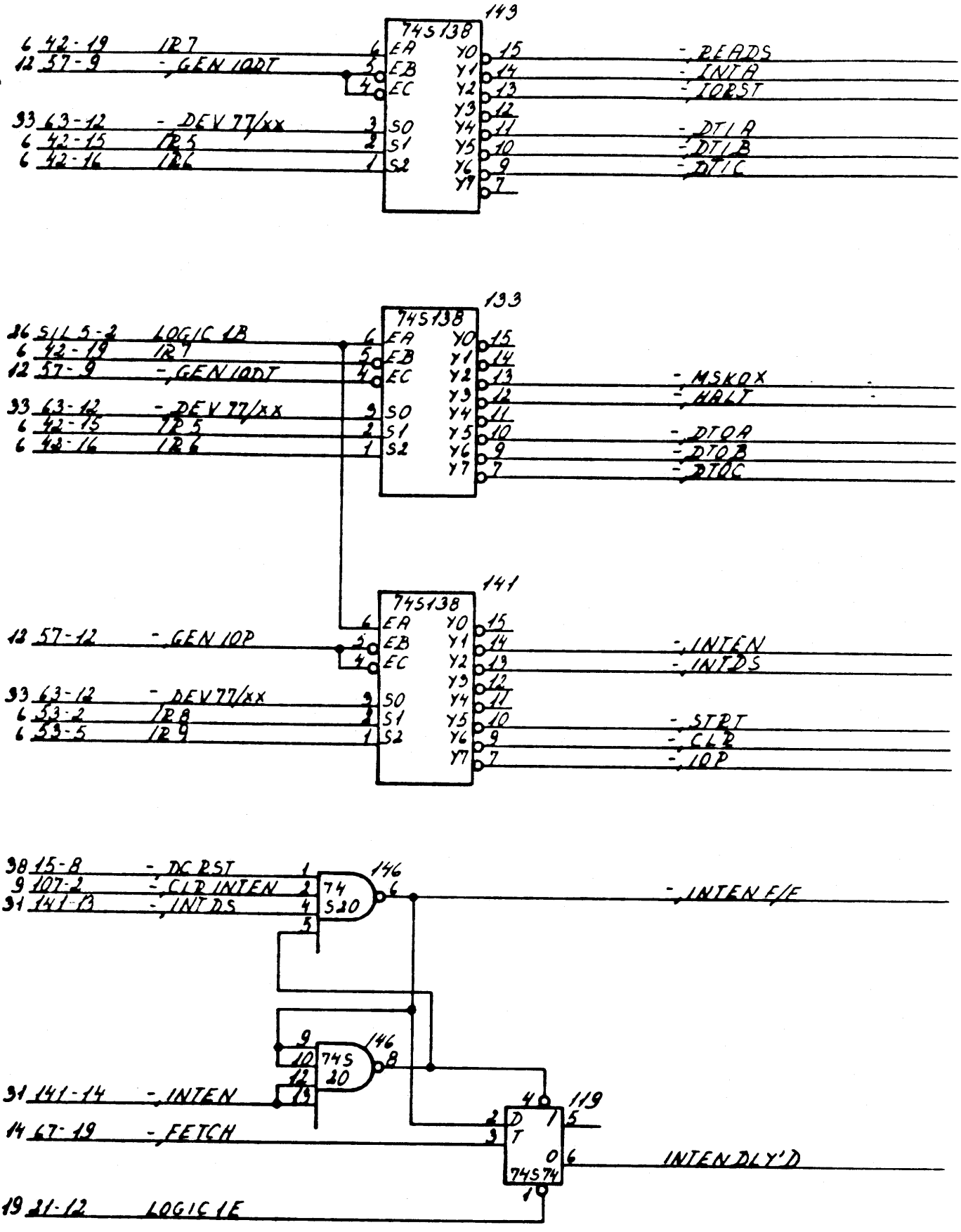


JKA 83 04 AGA 13

CPU 721
R13603

DATA BUS DRIVERS AND RECEIVERS
BIT 0-7
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~, CLR		p. 32	~, CLeaR I/O control signal.
~, DTI A-C		p. 32	~, DaTa Input A-C I/O data input strobes.
~, DTO A-C		p. 32	~, DaTa Output A-C I/O Data output strobes.
~, HALT		p. 37	~, HALT Resets the RUN flip flop.
~, INTA		p. 32	~, INTerrupt Acknowledge Device address strobe.
INTEN DLY'D		p. 17	INTerrupt ENable DeLaY'D.
~, INTEN F/F		p. 40 p. 33	~, INTerrupt ENable Flip/Flop.
~, INTEN		p. 31	~, INTerrupt ENable Sets the INT. ENable flip flop.
~, INTDS		p. 31	~, INTerrupt DiSable Resets the INT. ENable flip flop.
~, IOP		p. 32	~, Input/Output Pulse I/O Control Signal.
~, IORST		p. 32	~, Input/Output ReSeT I/O control signal.
~, MSKO x		p. 32	~, MaSK Output x Strobes priority mask out on the I/O Data BUS.
~, READ S		p. 34 p. 29	~, READ Switches BUS:= Data Switches.
~, STRT		p. 32	~, STaRT I/O control signal.



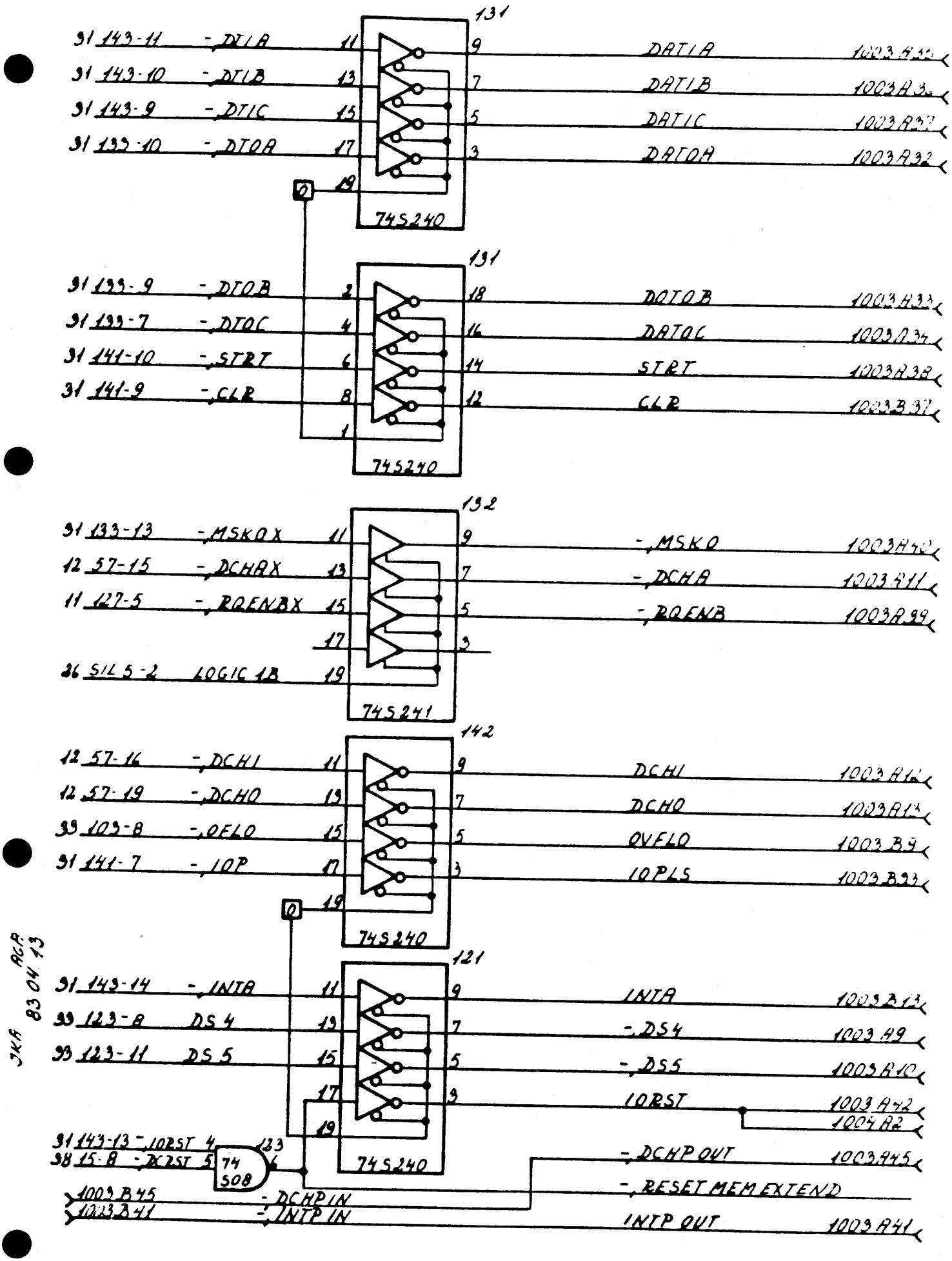
JWA REP 83 04 13

CPU 721
1R13604

BUS CONTROL SIGNAL DECODERS
INTERRUPT ENABLE F/F
Logic Diagram

31

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
CLR		1003	CLeaR.
DATI A-C		1003	DATa Input A-C.
DATAO A-C		1003	DATa Output A-C.
-; DCHA		1003	-; Data CHannel Acknowledge.
DCHI		1003	Data CHannel Input.
DCHO		1003	Data CHannel Output.
-; DCHP OUT		1003	-; Data CHannel Priority OUT.
-; DS 4-5		1003	-; Device Select bit 4-5.
INTA		1003	INTerrupt Acknowledge.
-; INTP OUT		1003	-; INTerrupt Priority OUT.
IOPLS		1003	Input/Output PuLSe.
IORST		1003	Input/Output ReSeT.
-; MSKO		1003	-; MaSK Output.
OVFLO		1003	OVerFLOw.
-; RESET MEM EXTEND		p. 41	-; RESET MEMory EXTEND flip flop.
-; RQENB		1003	-; ReQuest ENaBle.
STRT		1003	STaRT.



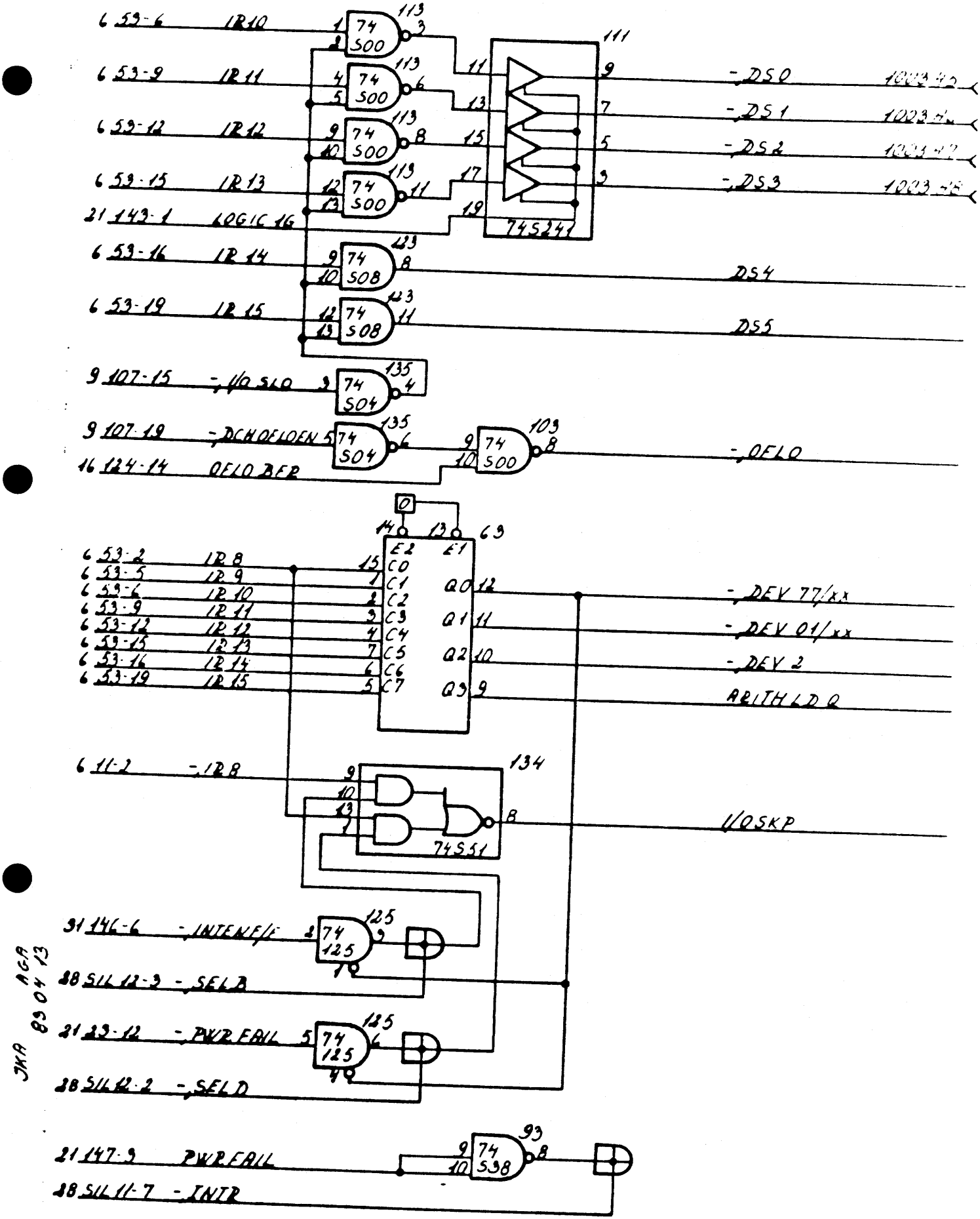
CPU 721

R 13605

BUS CONTROL SIGNAL DRIVERS
Logic Diagram

32

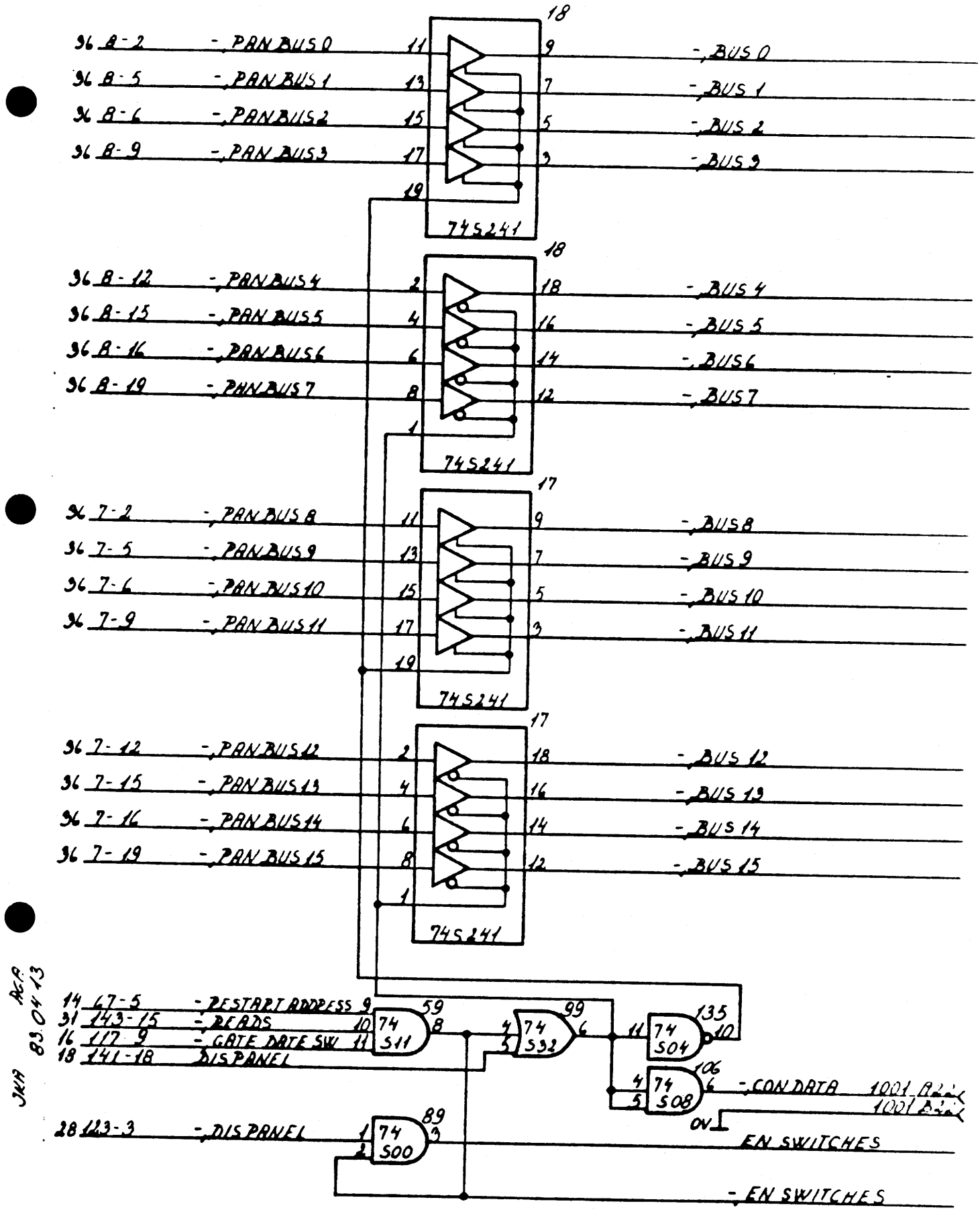
SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
ARITH LD Q		p. 15	ARITHmetic LoaD Q Determines whether the ALU result is loaded into Q or the Destination Accumulator.
-, DEV 01/xx		p. 17 p. 41	-, DEVice 01/xx.
-, DEV 77/xx		p. 17 p. 31 p. 33	-, DEVice 77/xx.
-, DEV 2		p. 42	-, DEVice 2.
DS 5-4		p. 32	Device Select 5-4, see below.
-, DS 3-0		1003	-, Device Select bit 3-0 Device address out on the I/O Bus.
INTPO		p. 32	INTerrupt Priority Out.
I/O SKP		p. 18	Input/Output SKiP condition Tests the state of DONE or BUSY depending upon IR 8.
-, OFLO		p. 32	-, OverFLOw Overflow status to the Data Channel.



CPU 721
R 13606

BUS ADDRESS LINE DRIVERS
POWER FAIL INTERRUPT CIRCUIT
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻; BUS 0-7		p. 23	⁻; internal BUS bit 0-7 The shown source on the Bus is the Diagnostic Panel Bus Receivers.
⁻; BUS 8-15		p. 24	⁻; internal BUS bit 8-15 See above.
⁻; CON DATA		1001	⁻; CONtrol DATA Used to disable the data switches in the Diagnostic Panel from the ⁻; PAN BUS.
EN SWITCHES		p. 36	ENable SWITCHES Disables the Data Register from the ⁻; PAN BUS, during read switches.
⁻; EN SWITCHES		p. 40	⁻; ENable SWITCHES Same as above.

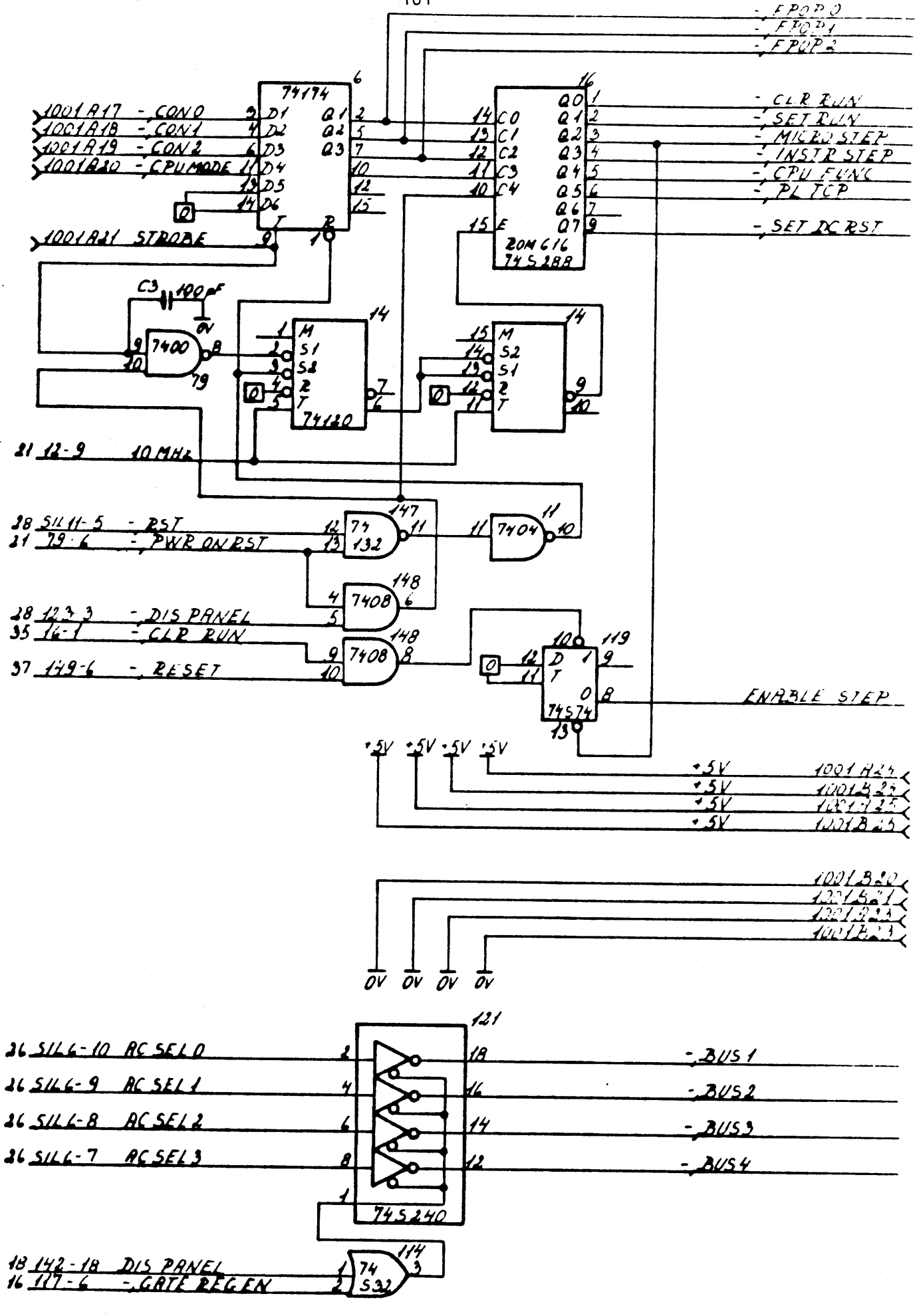


CPU 721
R13607

SERVICE PANEL DATA BUS RECEIVERS
Logic Diagram

94

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~ BUS 1-4		p. 23	~ internal BUS bit 1-4 This source on the internal bus is the AC SEL switches on the Diagnostic Panel.
~ CLR RUN		p. 27 p. 37	~ CLeaR RUN.
~ CPU FUNC		p. 27 p. 37	~ CPU FUNCtion Diagnostic Panel operation which requires microprogram execution.
ENABLE STEP		p. 37	ENABLE STEP.
~ FPOP 0-2		p. 16	~ Front Panel OPeration bit 0-2 The value on these lines determines the operation requested from the Diagnostic Panel.
~ INSTR STEP		p. 27 p. 37	~ INSTRuction STEP.
~ MICRO STEP		p. 27 p. 37	~ MICROinstruction STEP.
~ PL TCP		p. 28	~ set Program Load from TCP.
~ SET DC RST		p. 26 p. 38	~ SET DC ReSeT Used to generate IORST.
~ SET RUN		p. 27 p. 37	~ SET RUN Sets the RUN flip flop.

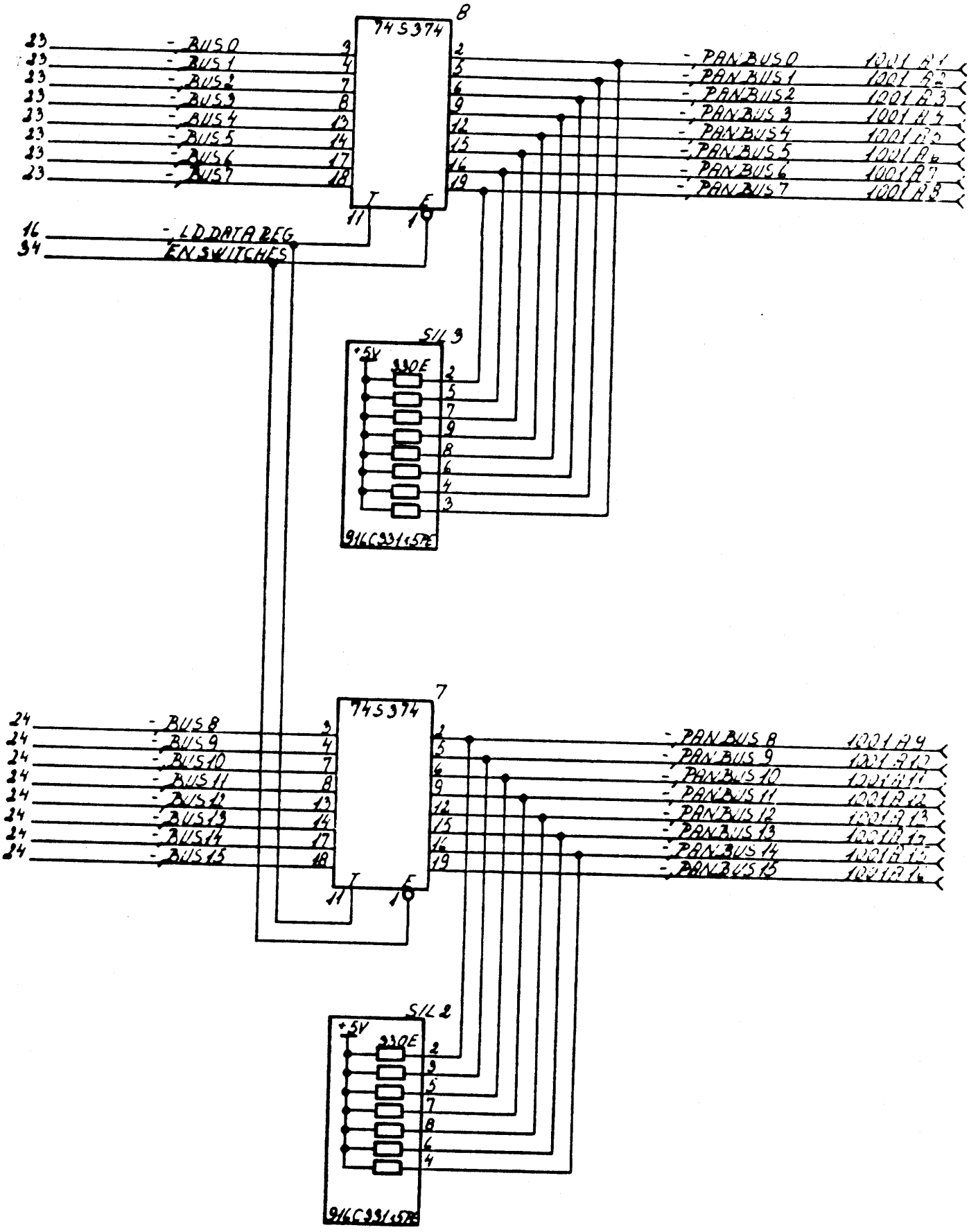


CPU 781

.R 13608

FRONT PANEL CONTROL DECODER
ACCUMULATOR SELECT RECEIVERS & GATES
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-, PAN BUS 0-14		1001 p. 34	-, PANel BUS bit 0-14 Bidirectional data bus between the CPU and Diagnostic Panel. Transfers the information on the internal Data Bus out on the indicators in the Diagnostic Panel, TCP 701. Transfers the information on Diagnostic Panel data switches to the internal CPU Data Bus.
-, PAN BUS 15		p. 27 p. 34	-, PANel BUS bit 15 Same as above.



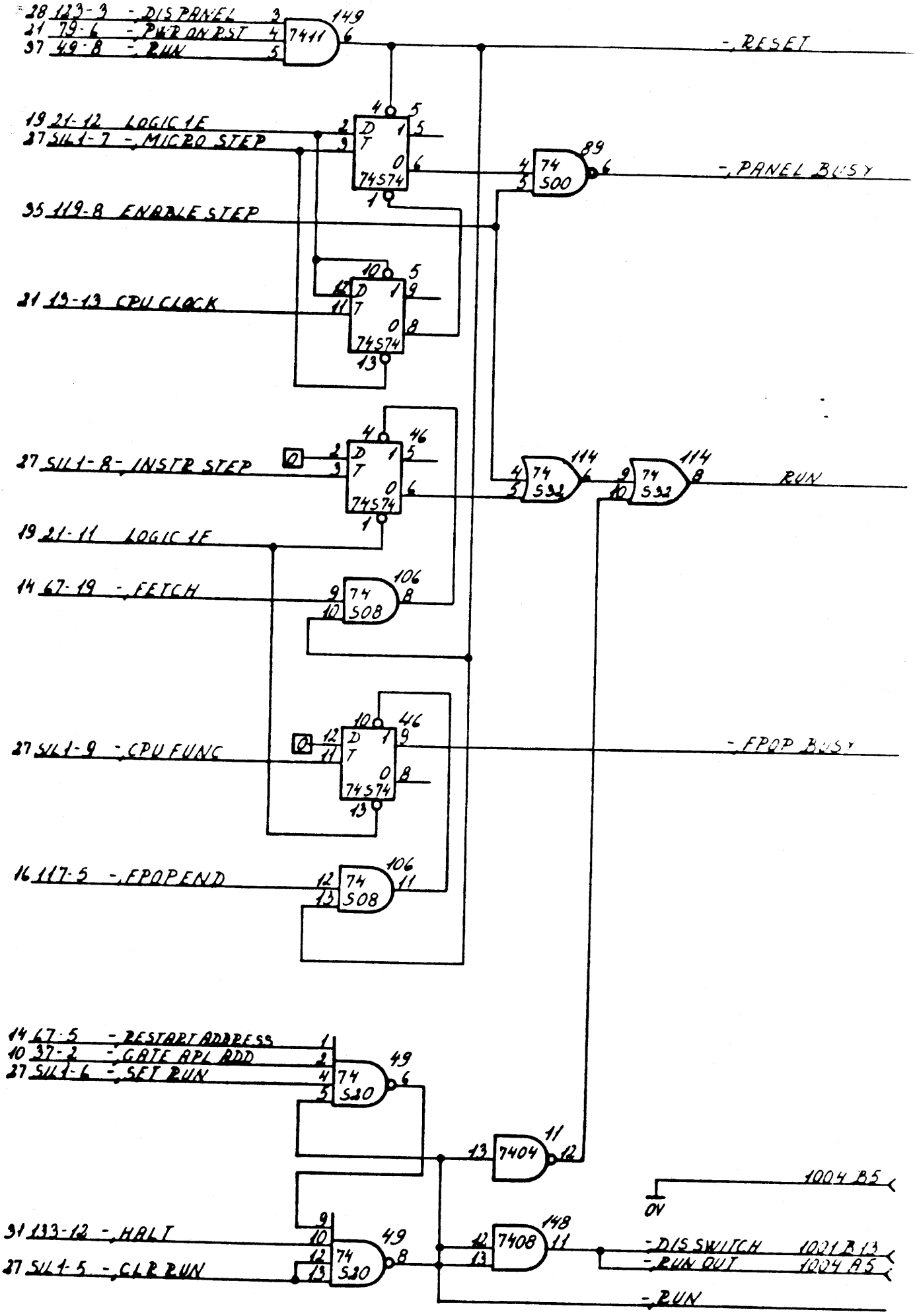
JKH FGA 83 04 19

CPU 721

R12609

DATA BUS LATCH & DRIVERS TO SERVICE PANEL
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
⁻; DIS SWITCH		1001	⁻; DISable SWITCHes Used to disable the function switches on the Diagnostic Panel (not RESET, STOP) when the CPU is in the RUN state.
⁻; FPOP BUSY		p. 16	⁻; Front Panel OPeration BUSY Indicates that the Diagnostic Panel has requested an operation Reset by the microprogram when the current operation is terminated.
⁻; PANEL BUSY		p. 19	⁻; PANEL BUSY Indicates that an INSTR STEP or MICRO STEP is in progress.
⁻; RESET		p. 35	⁻; RESET
RUN		p. 41	RUN The RUN state of the CPU.
⁻ RUN OUT		1004	RUN OUTput.
⁻; RUN		p. 22	⁻; RUN.



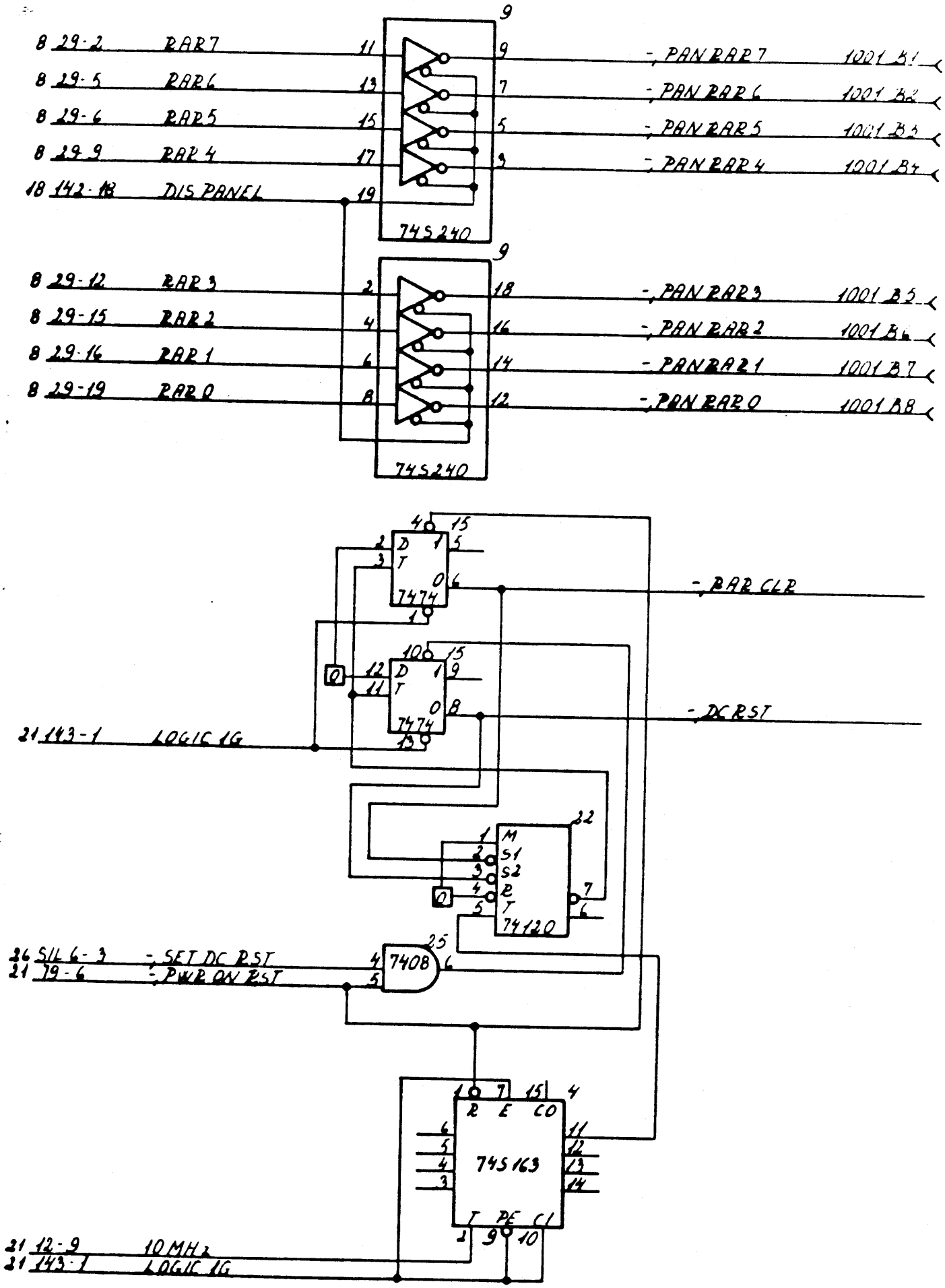
JHA AGA 83 04 13

CPU 721

pr 19 6 10

MICROSTEP - INSTRUCTIONSTEP - CPU PANEL FUNCTION
RUN/STOP F/F
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~, DC RST		p. 16 p. 22 p. 32 p. 31	~, DC ReSeT Clears the CPU and generates IORST.
~, PAN RAR 0-7		1001	~, PANel Rom Address Register bit 0-7 Output to drive indicators on the Diagnostic Panel.
~, RAR CLR		p. 41 p. 18	~, Rom Address Register CLeaR Generated after power on reset to determine the start address of the microprogram.

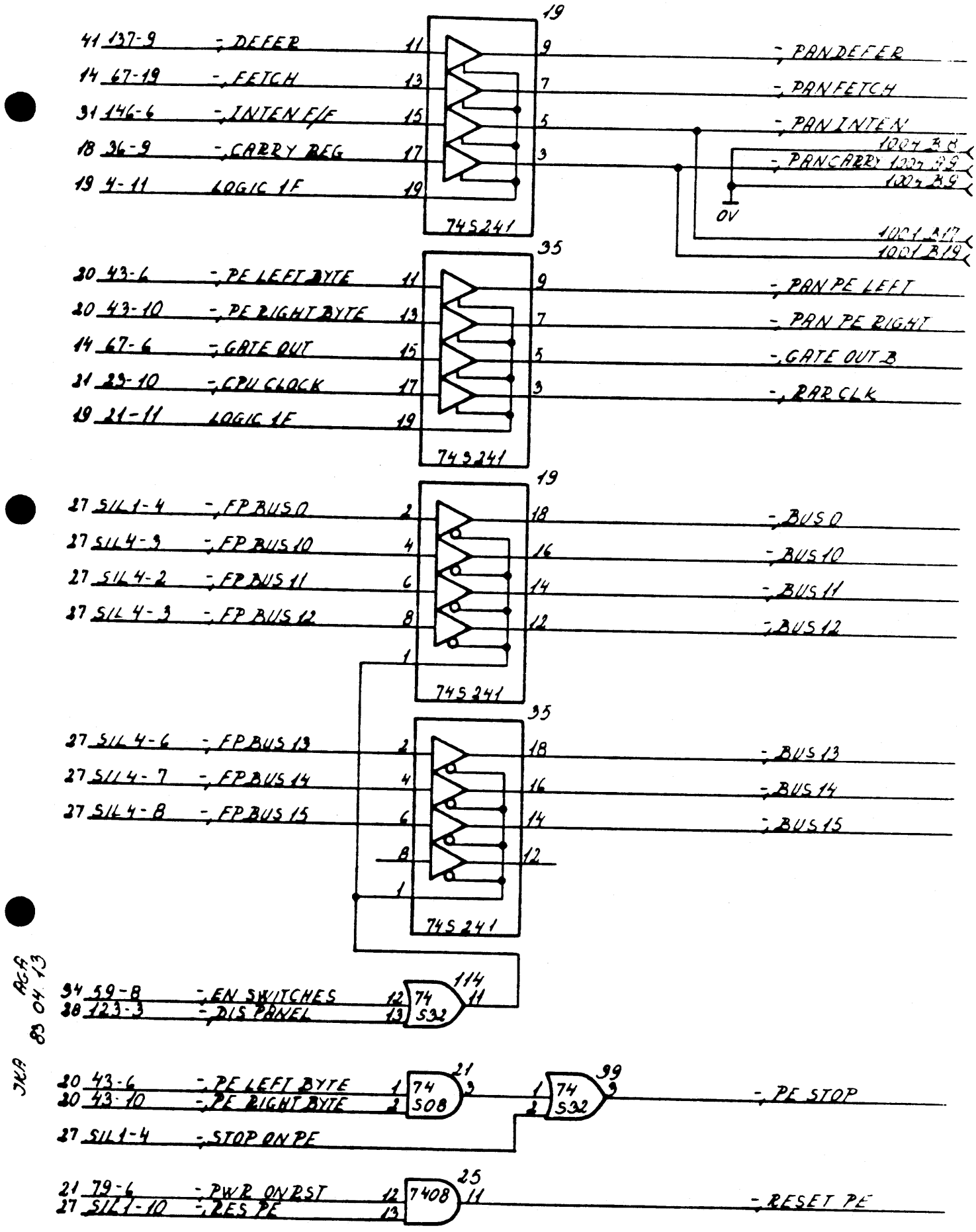


JKA 03.04.13 AGP

CPU 721
B 13611

MICROADDRESS DRIVERS
POWER ON RESTART CIRCUIT
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
~; BUS 0		p. 23	~; internal BUS 0
~; BUS 10-15		p. 24	~; internal BUS 10-15 The data switches on the Front Panel gated out on the internal bus.
~; GATE OUT B		p. 29 p. 30	~; GATE OUTput Buffer Gates the contents on the internal bus out on the I/O Data Bus.
~; PAN CARRY		p. 39 1004 1001	~; PANel CARRY indicator.
~; PAN DEFER		p. 39	~; PANel DEFER indicator.
~; PAN FETCH		p. 39	~; PANel FETCH indicator.
~; PAN INTEN		p. 39 1001	~; PANel INTerrupt ENable indicator.
~; PAN PE LEFT		p. 39	~; PANel Parity Error LEFT byte indicator.
~; PAN PE RIGHT		p. 39	~; PANel Parity Error RIGHT byte indicator.
~; PE STOP		p. 19	~; Parity Error STOP.
~; RAR CLK		p. 1 p. 2 p. 8 p. 9 p. 10 p. 11 p. 12 p. 13 p. 14 p. 15 p. 16 p. 41 p. 42	~; Rom Address Register CLocK.
~; RESET PE		p. 20	~; RESET Parity Error indicator register.

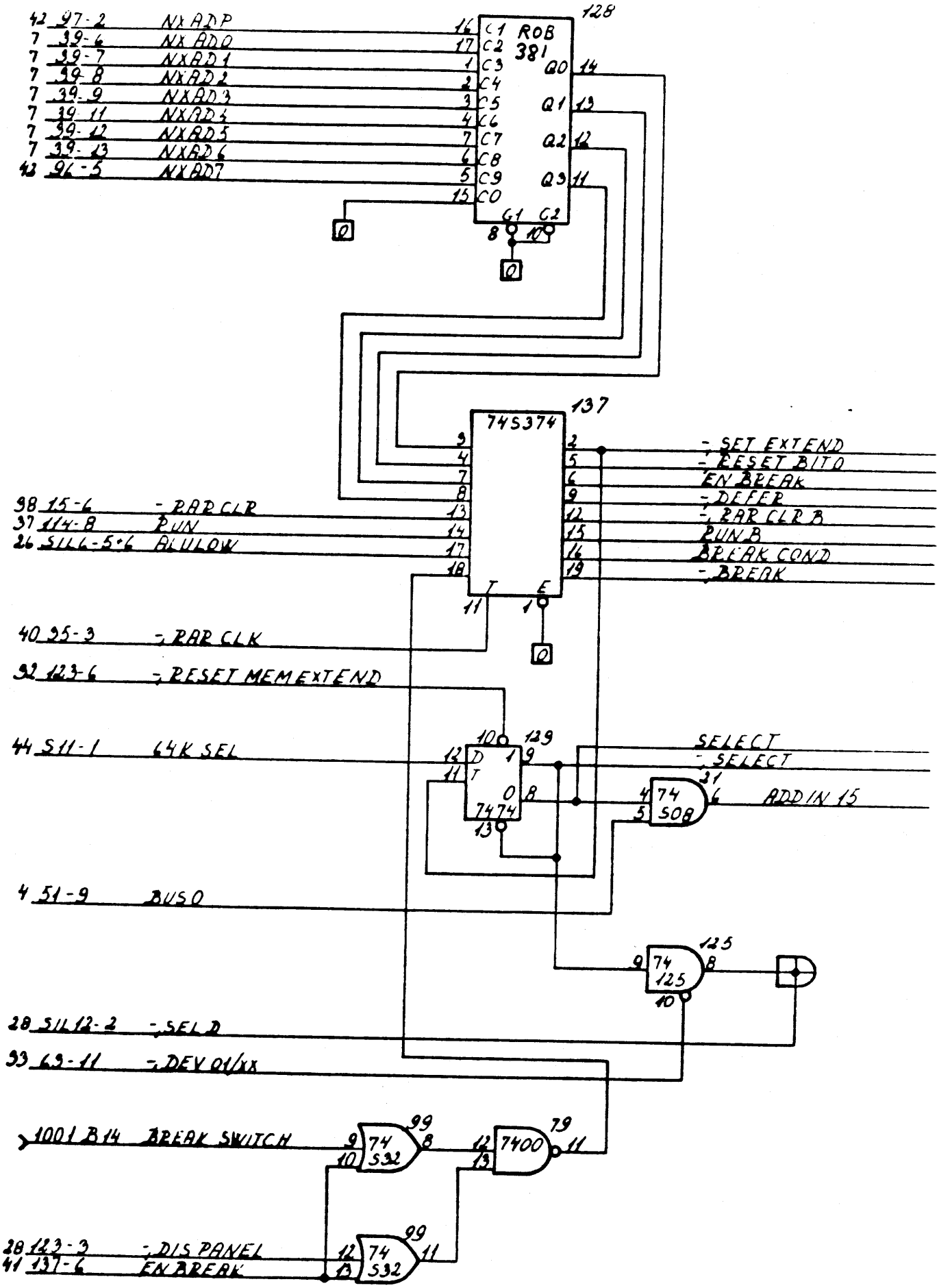


CPU721

R12612

CPU FRONT PANEL DRIVERS AND RECEIVERS
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
ADD IN 15		p. 6	ADDress INput bit 15 Logic zero if memory extension is not selected, else Bus 0.
BREAK COND		p. 17	BREAK CONDition Indicates that PC is equal to AC5.
-; BREAK		p. 11	-; BREAK switch.
-; DEFER		p. 40	-; DEFER Indicates that the next microcycle will be used to follow an indirection chain.
EN BREAK		p. 17	ENable BREAK Indicates that the microprogram is in the BREAK loop.
-; RAR CLR B		p. 7	-; Rom Address Register CLeaR Buffer Generated after power on reset to determine the start address of the microprogram.
-; RESET BIT 0		p. 18	-; RESET BIT 0 Places a zero as input to the MSB end of the register shifter (Qo SRI/SLO).
RUN B		p. 17	RUN Buffer Indicates that the CPU is in the RUN state.
-, SET MEM EXTEND		p. 41	
SELECT		p. 17 p. 18 p. 42	Memory extension SELECTed.
-; SELECT		p. 3	



CPU 721

MICROPROGRAM STORE
Logic Diagram

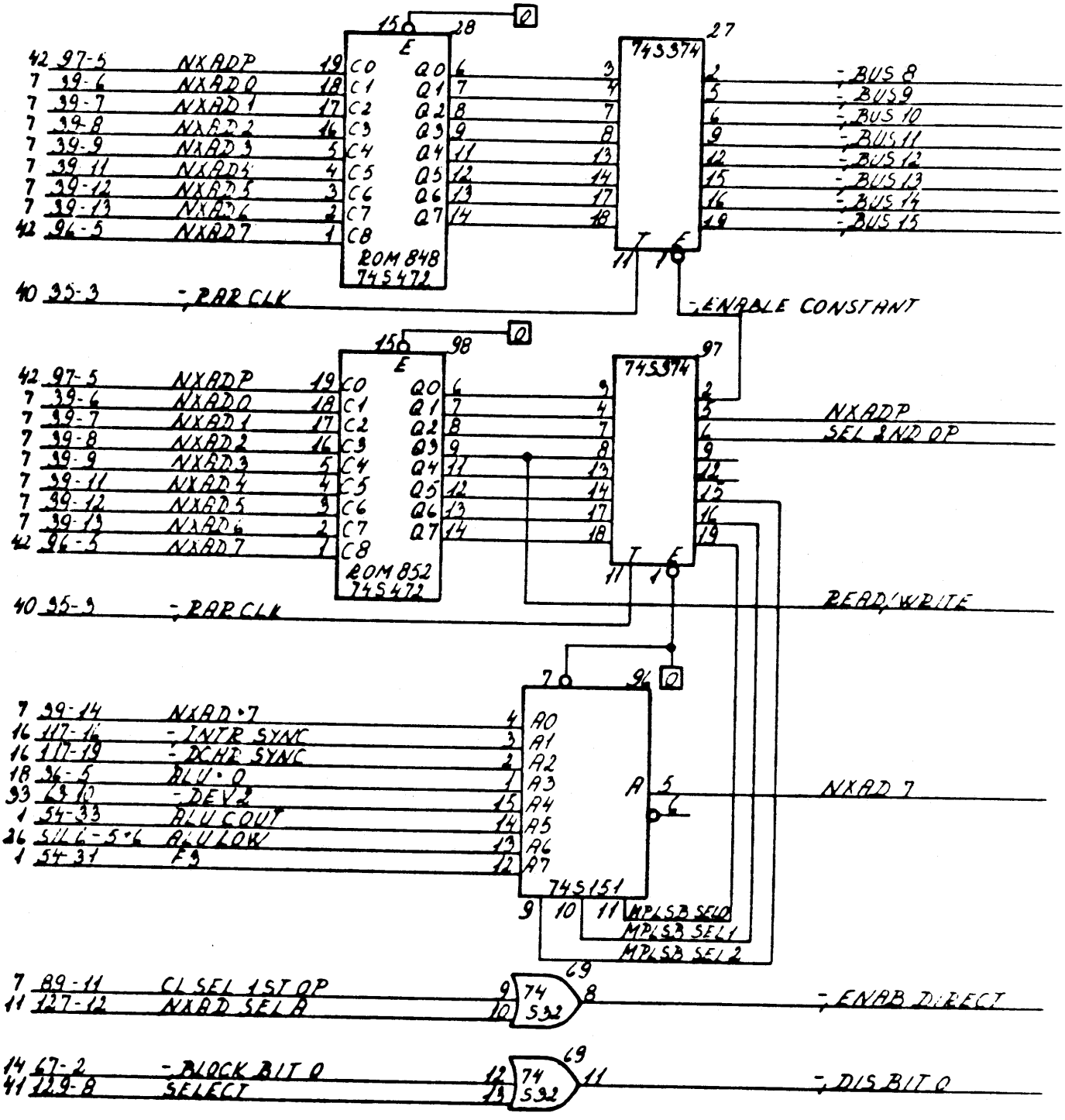
SIGNAL	DESTINATION	DESCRIPTION
-, BUS 8-15	p. 24	Internal BUS 8-15
-, DIS BIT 0	p. 5	DISable BIT 0 Forces a logical zero on bit 0 from the AM 2901A array in 64K mode.
-, ENAB DIRECT	p. 7	ENABle next microaddress DIRECT
NX AD 7	p. 8 p. 9 p. 10 p. 11 p. 12 p. 13 p. 14 p. 15 p. 16 p. 41 p. 42	NeXt micro ADdress bit 7 The least significant bit in the address.
NX ADP	p. 9 p. 10 p. 11 p. 12 p. 13 p. 14 p. 15 p. 16 p. 41 p. 42	NeXt ADdress Page Used to separate BANK 0 and BANK 1 in the micro program.
SEL 2ND OP	p. 7	SELect 2ND OPerand Used to separate NEXT address map in BANK 0 and BANK 1.
READ/WRITE	p. 19	READ or WRITE cycle in memory.

Designed by

Drawn by

Dwg. Source Check

Unit		
CPU721	MICROPROGRAM STORE EXTENSION	CPU 042
Dwg. No. R21386	Signal List	of 42



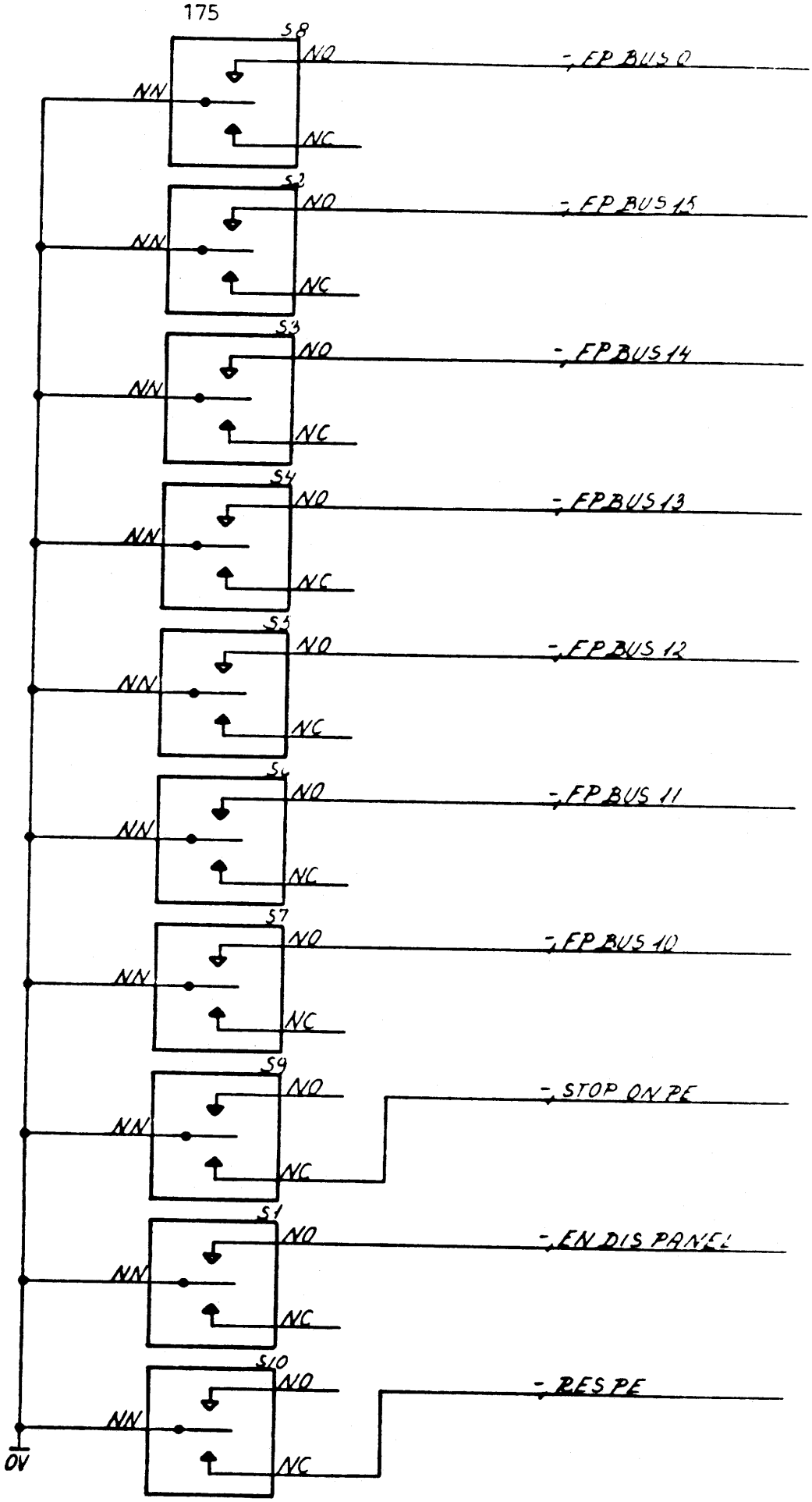
AGA 83.04.13

JKA

CPU 721
R 136 14

MICROPROGRAM STORE EXTENSION
Logic Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
-,FP BUS 0,10-15		p. 27	-, Front Panel Bus 0 Autoload information used when Diagnostic Panel is not enabled.
-,STOP ON PE		p. 27	-, STOP ON Parity Error Switches input from CPU721 Front-Panel, determining whether the CPU should both indicate parity error and stop the microprogram execution, or the CPU should only indicate parity errors.
-,EN DIS PANEL		p. 28	-,ENable DISable PANEL Enables or disables diagnostic panel.
-,RES PE		p. 27	-,RESet Parity Error Used to clear the Parity Error indicator register.



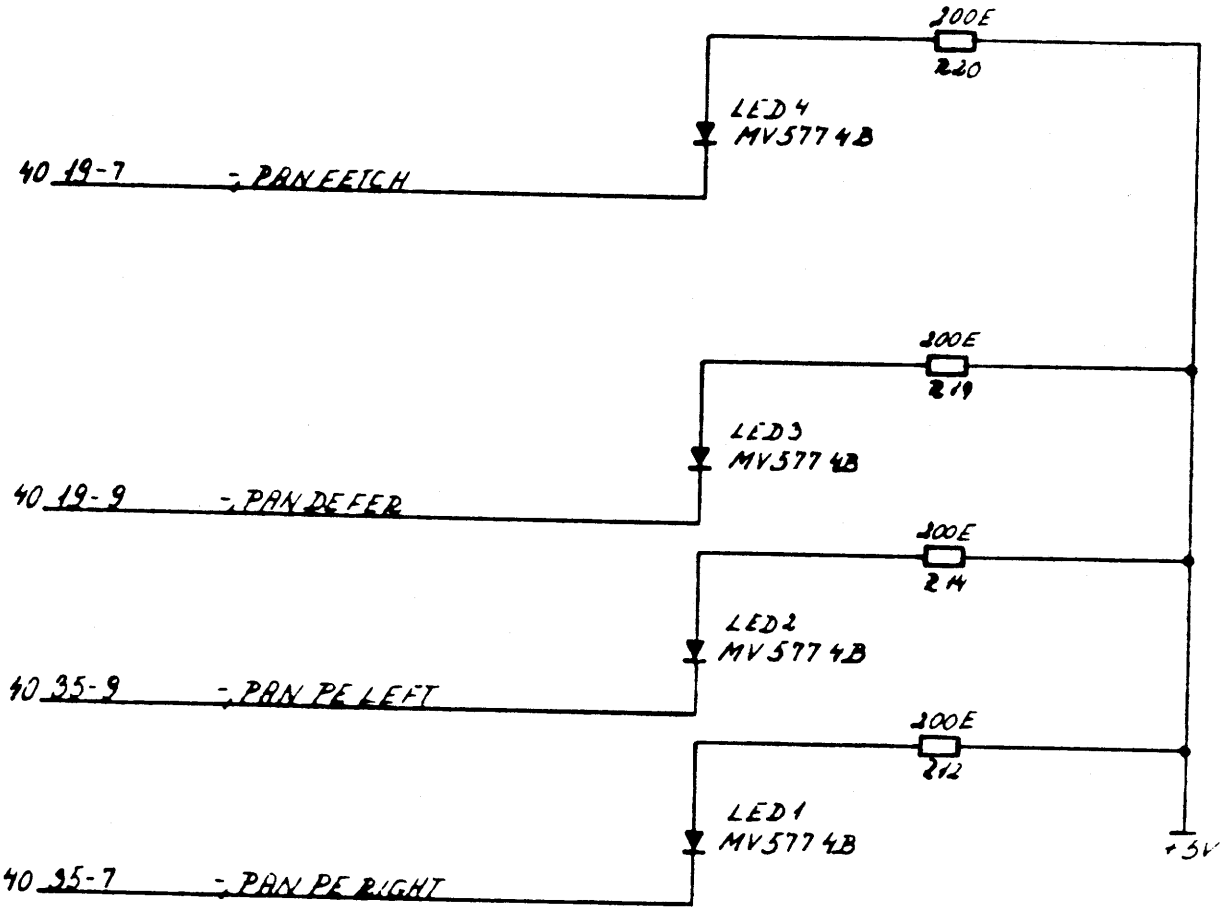
JKP AGD 83.04.13

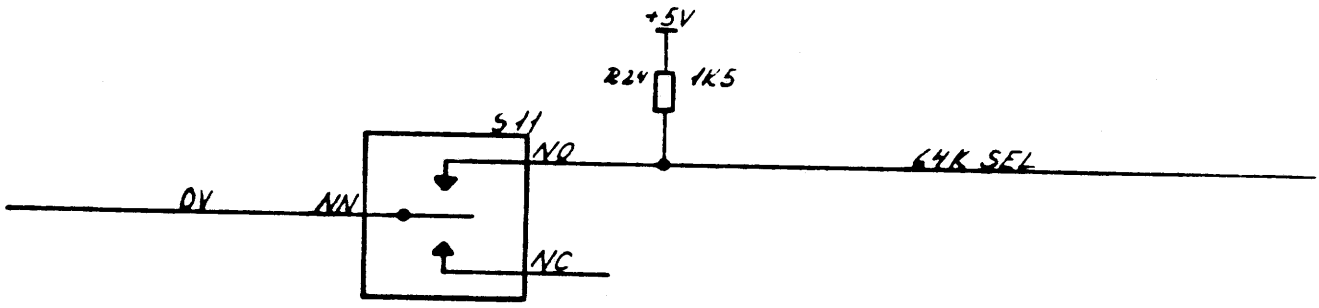
CPU 721

2 13615

FRONT PANEL SWITCHES
Circuit Diagram

SIGNAL	ORIGIN	DESTINATION	DESCRIPTION
64 K SEL		p. 41	Memory extension selected.



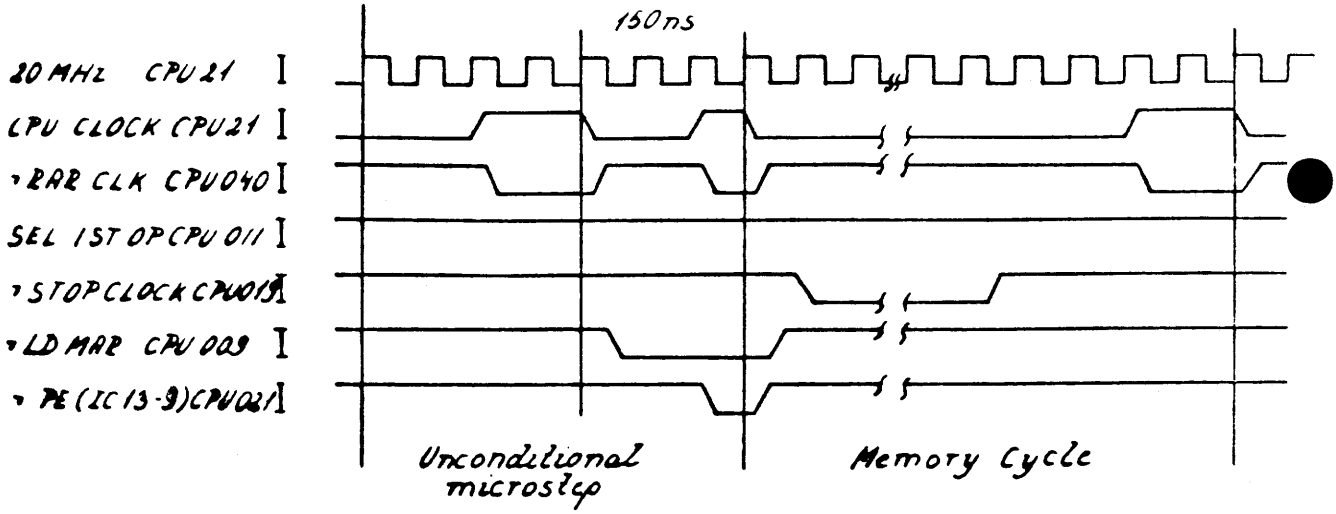


JKH AGH 89 04 11

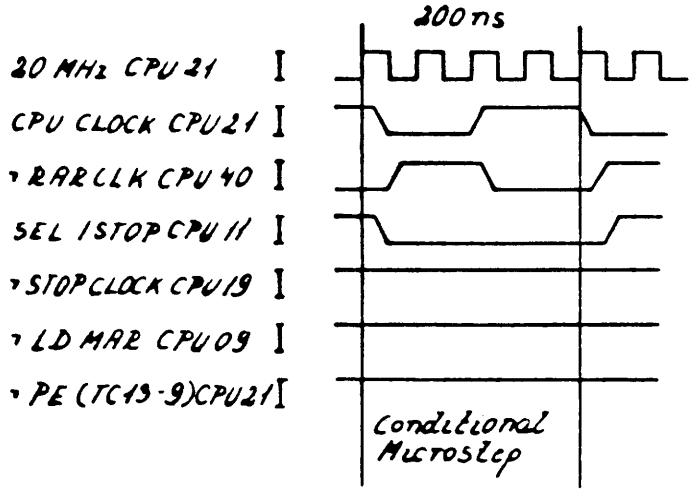
CPU 721
R 13616

FRONT PANEL INDICATORS
Circuit Diagram

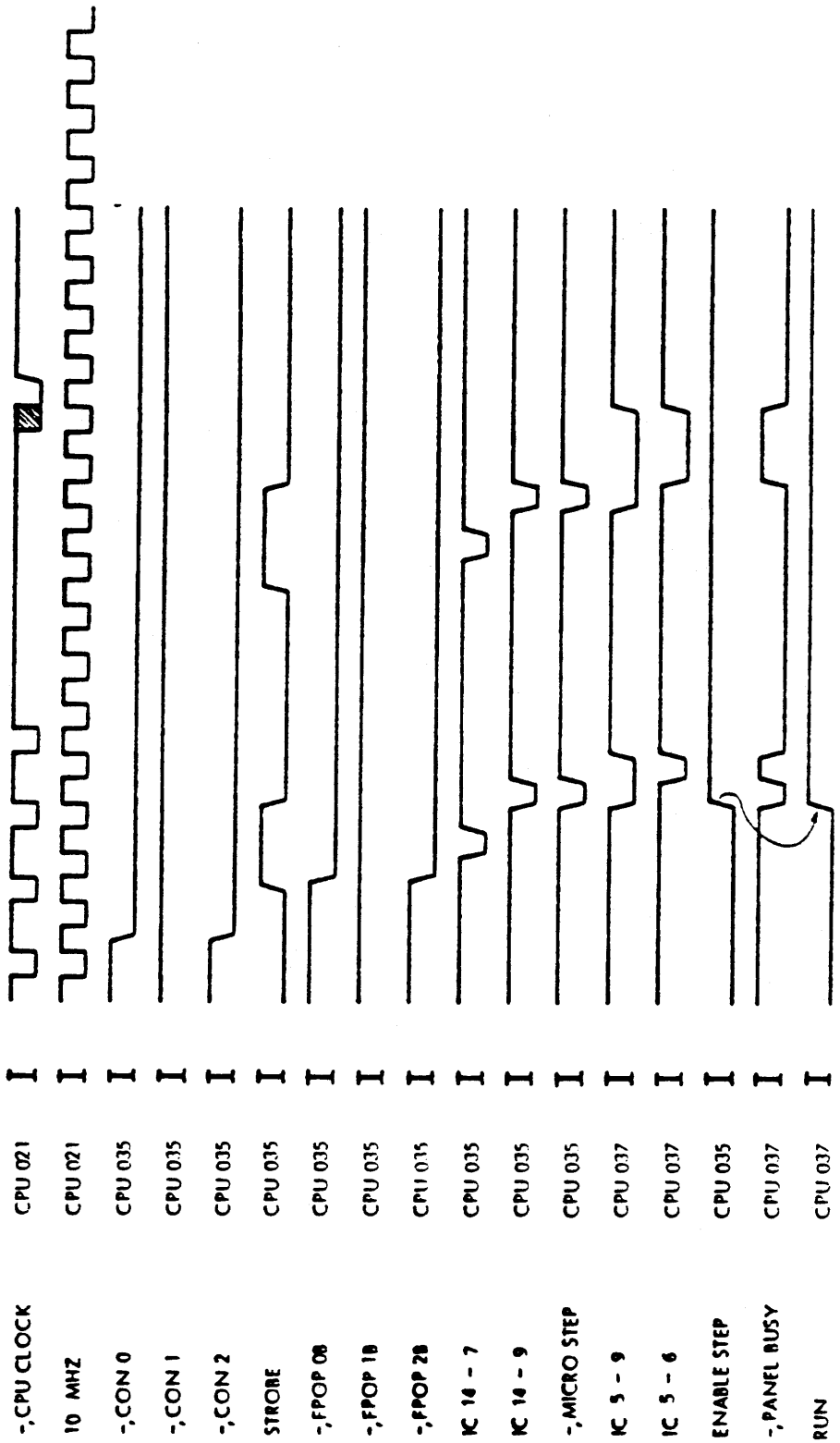
44



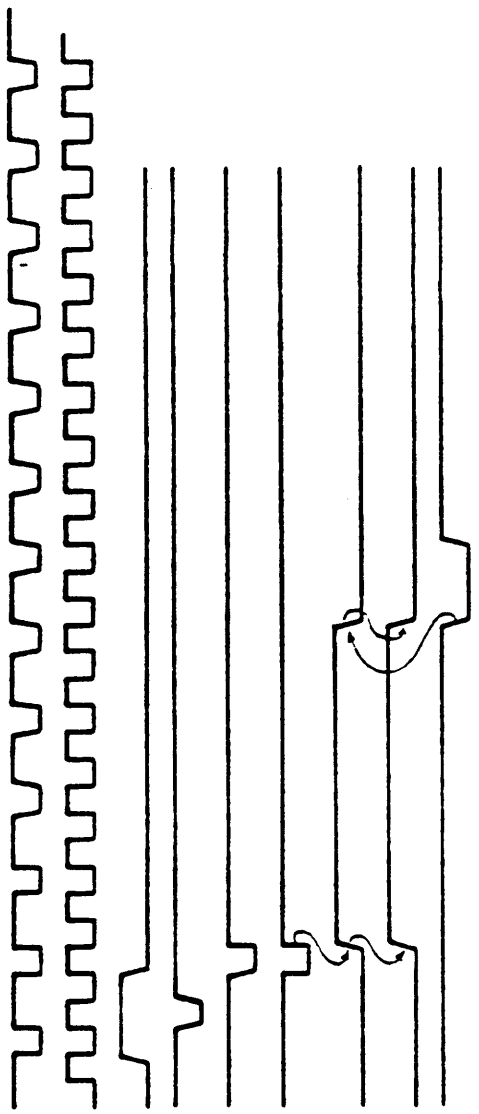
Clock timing



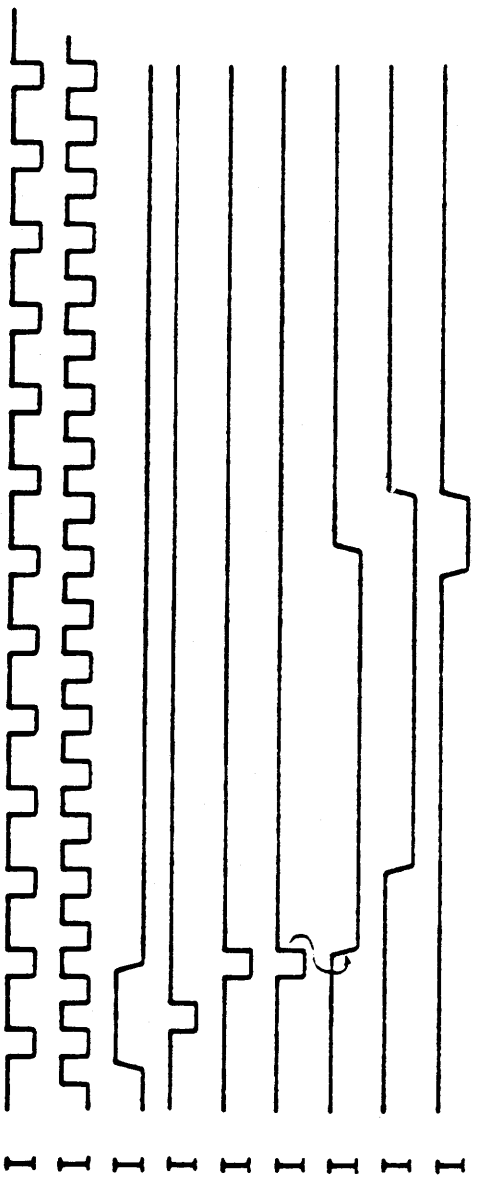
Clock timing


-,CPU CLOCK
10 MHZ
-,CON 0
-,CON 1
-,CON 2
STROBE
-,FPOP 08
-,FPOP 18
-,FPOP 28
IC 14 - 7
IC 14 - 9
-,MICRO STEP
IC 5 - 9
IC 5 - 6
ENABLE STEP
-,PANEL BUSY
RUN
CPU 021
CPU 021
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 035
CPU 037
CPU 037
CPU 035
CPU 037
CPU 037

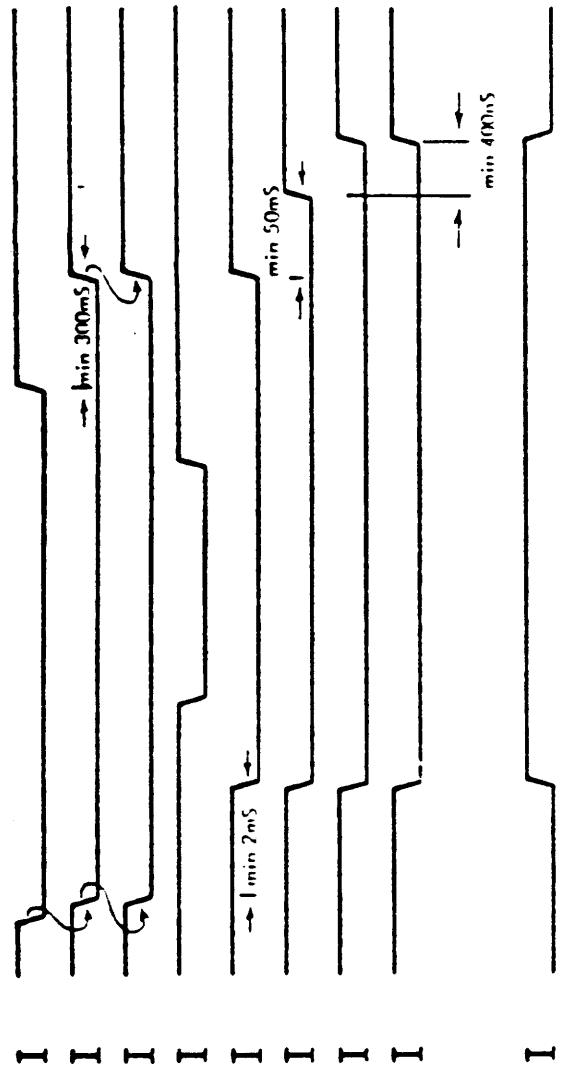
MICRO - STEP



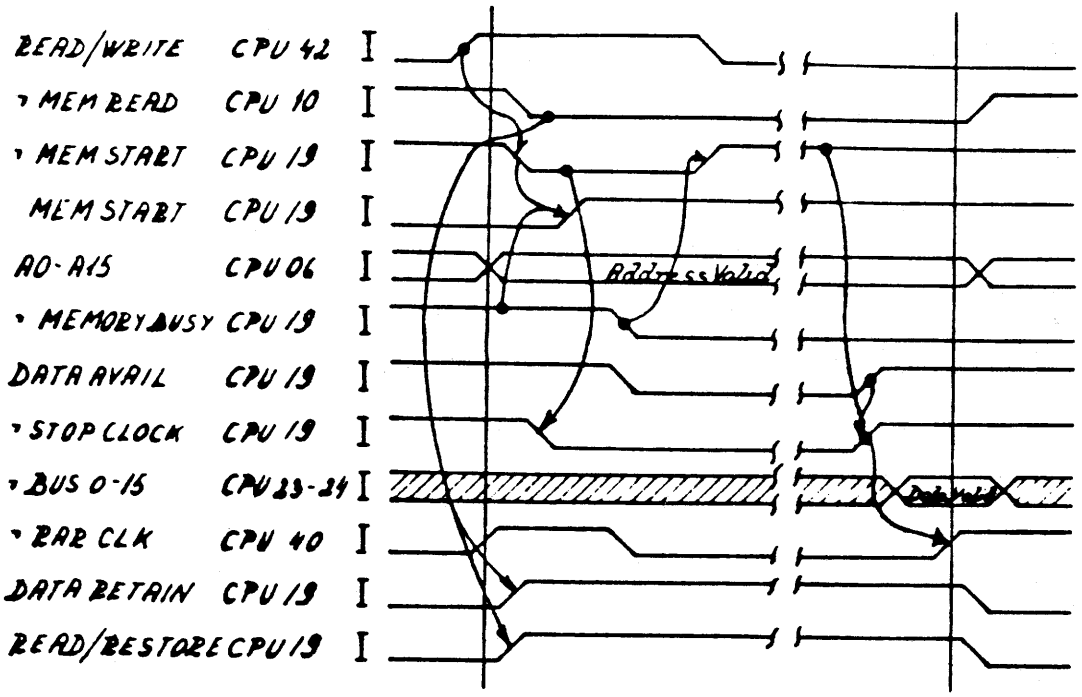
INSTRUCTION STEP



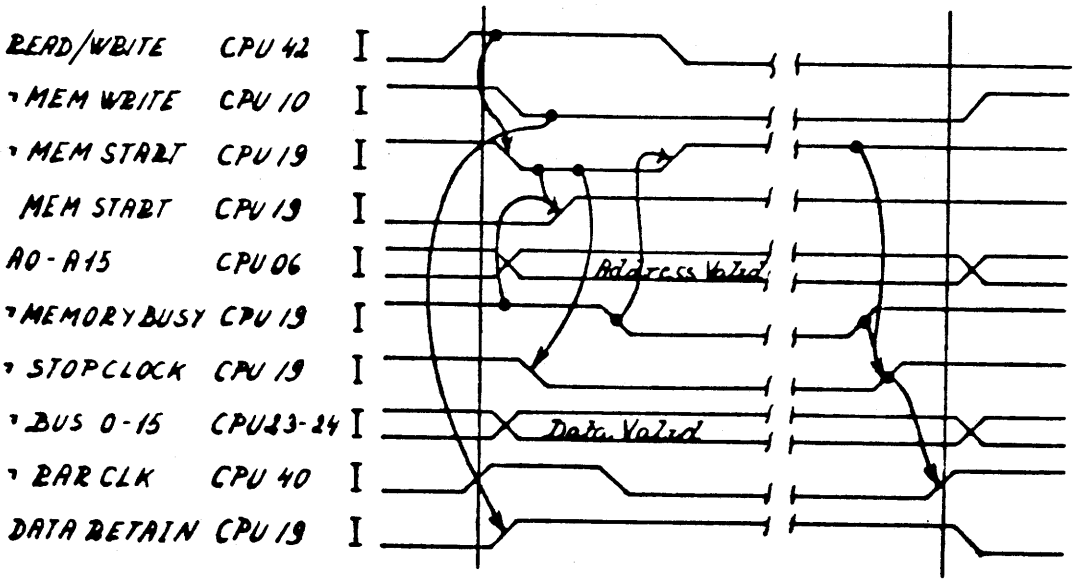
Ex-Ex next - Dep - Dep next - Rex - Rdep
start Diagnostic Panel Function



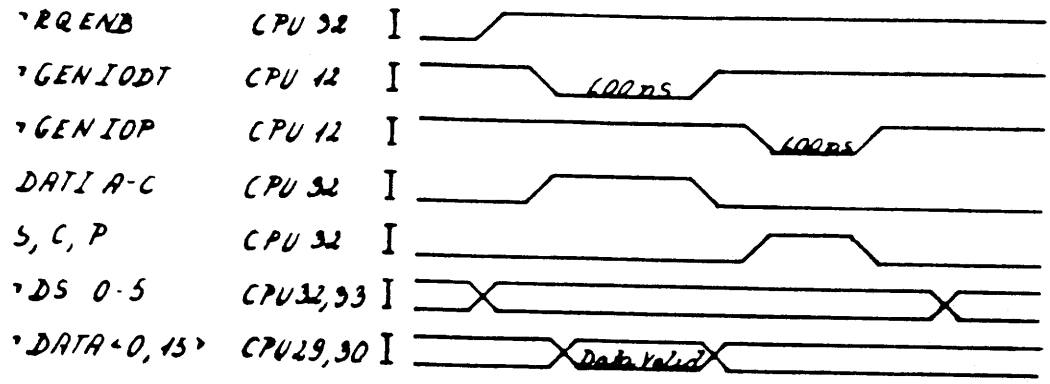
POWER FAIL AND RESTART



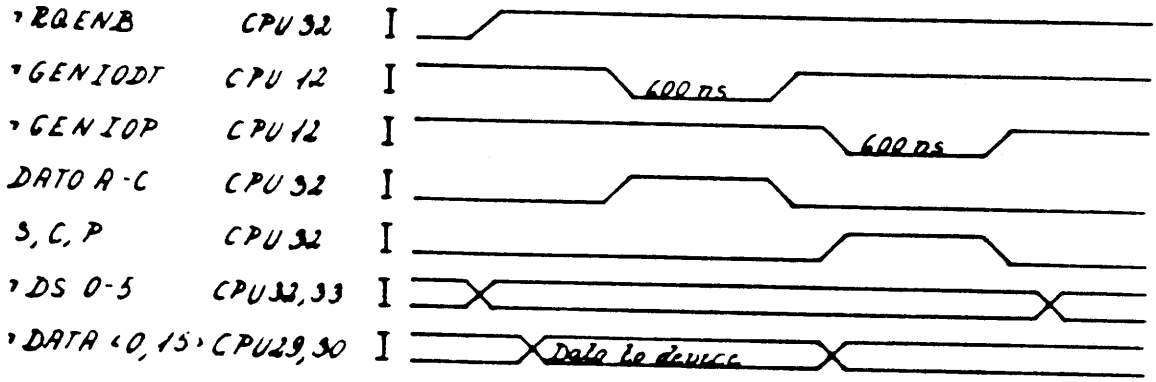
Memory read Cycle



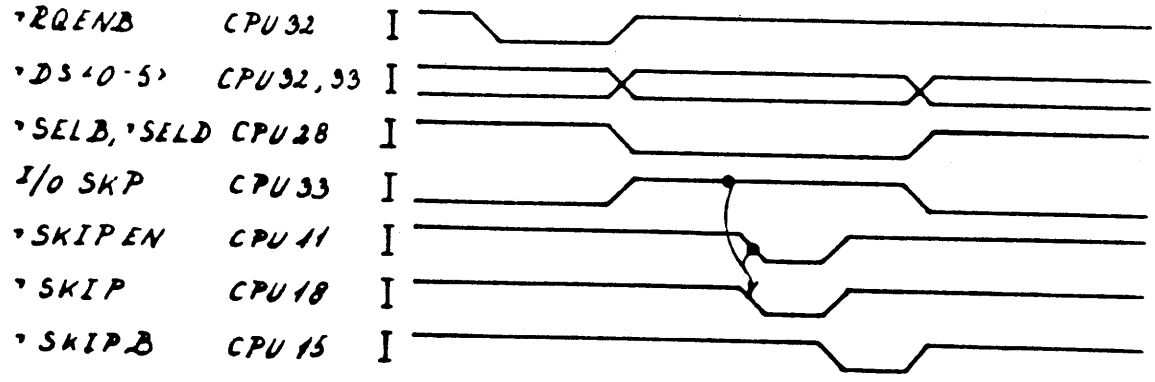
Memory Write Cycle



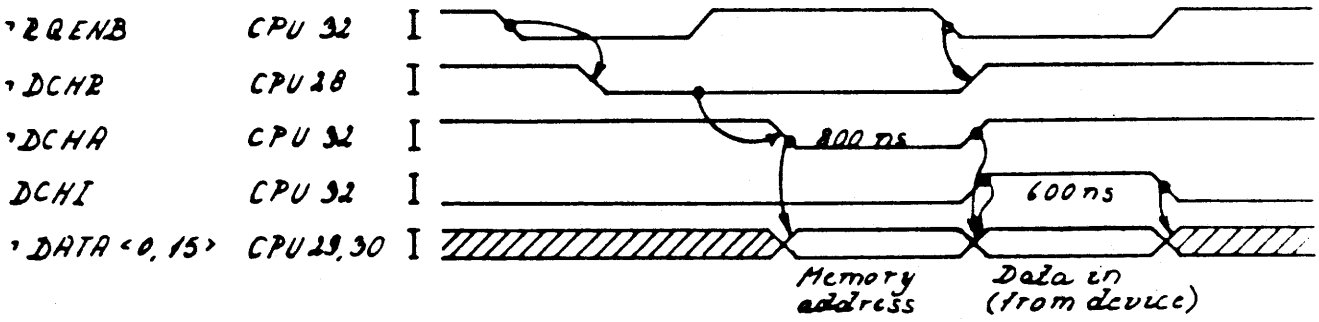
Programmed I/o input



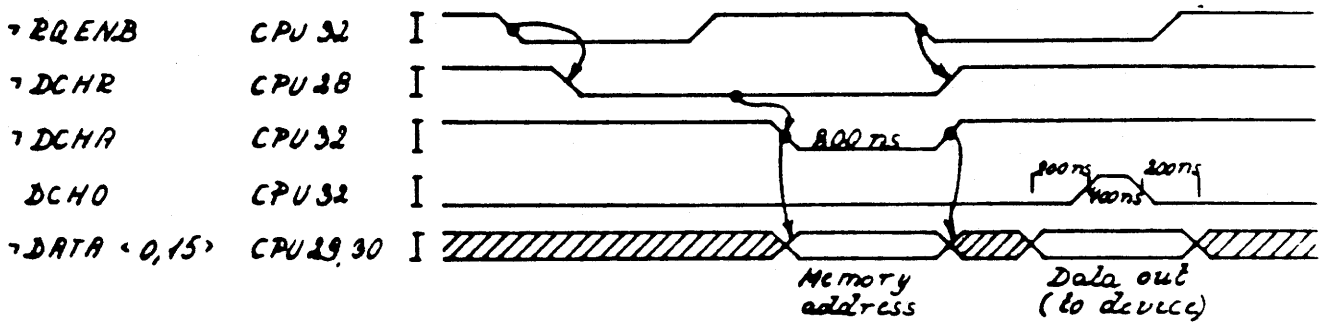
Programmed I/o output



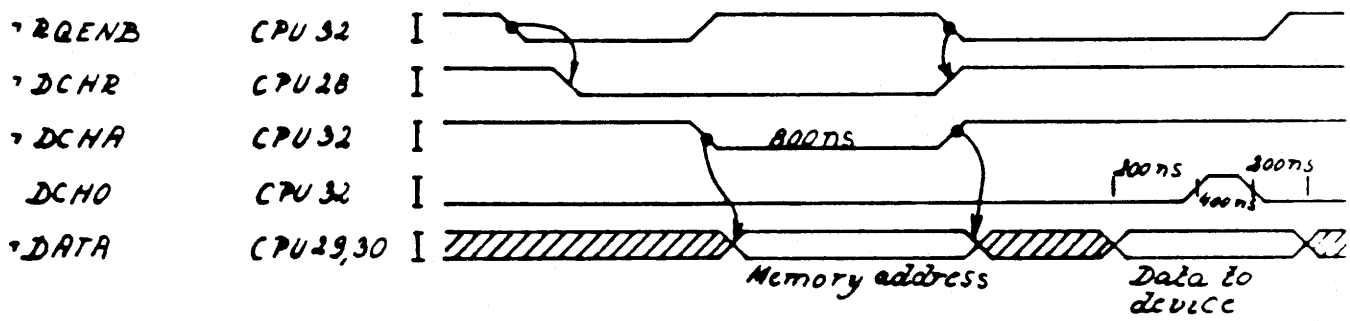
Programmed I/o skip



Data channel input



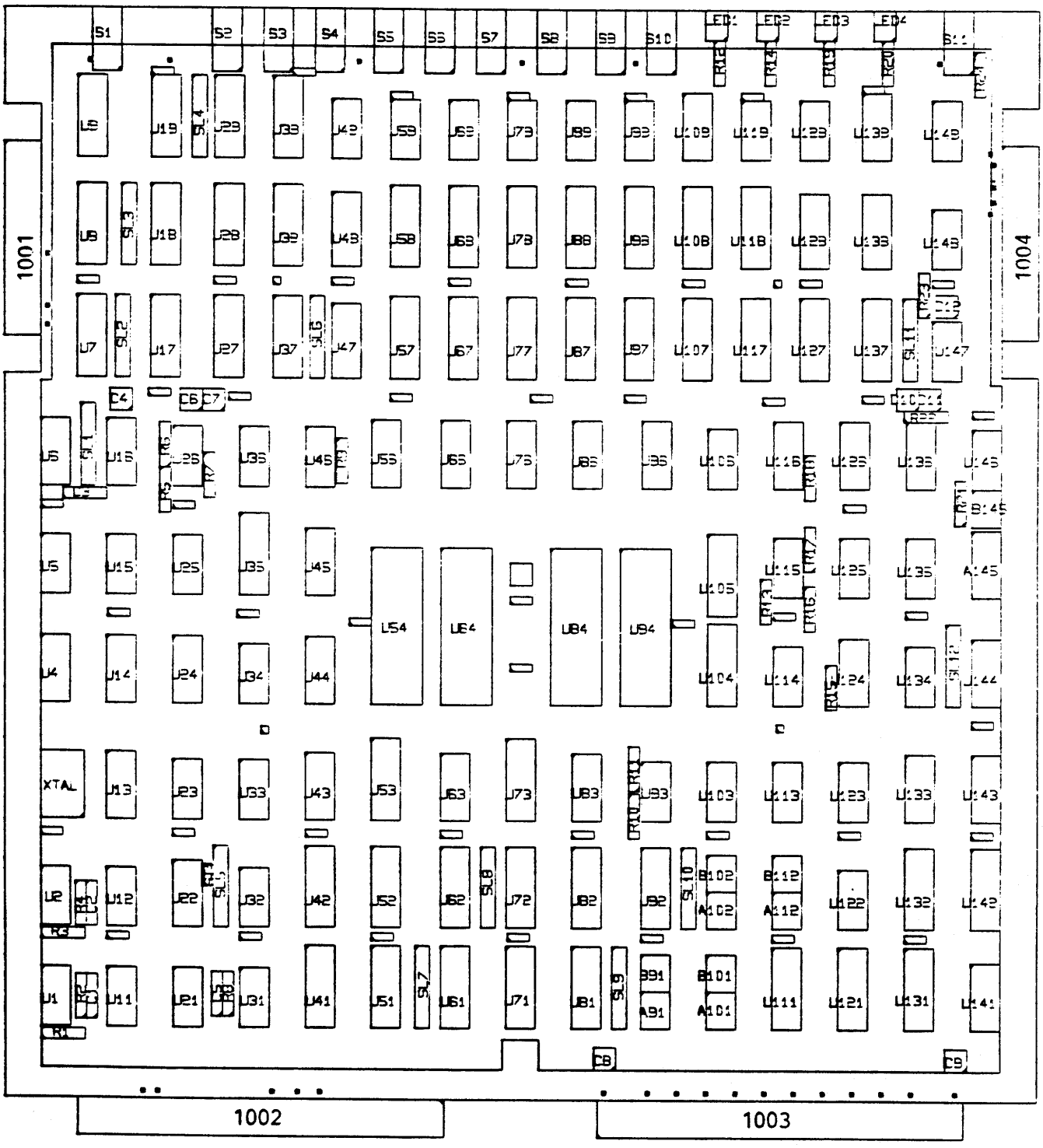
Data channel output



Data Channel increment.

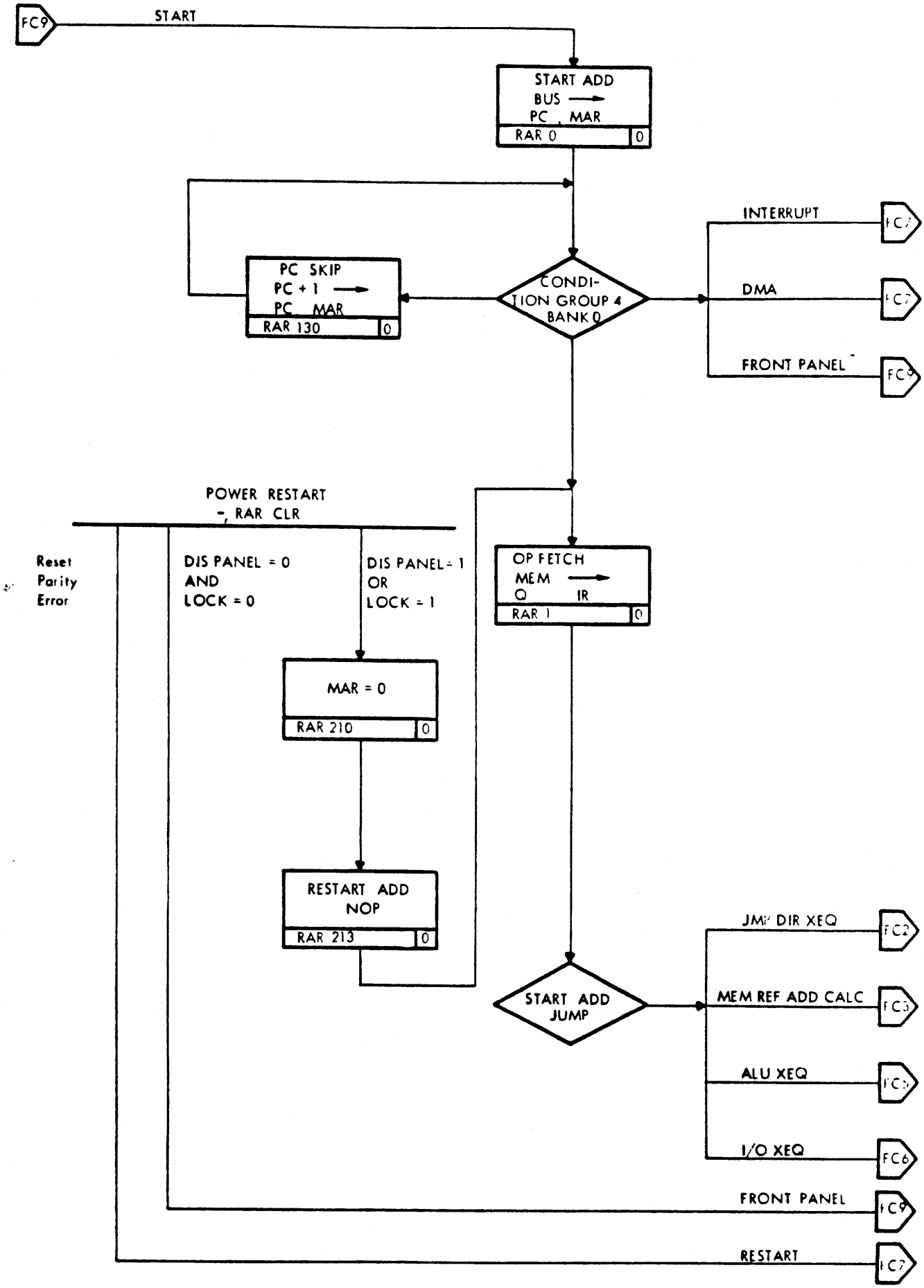
19. Assembly Drawing

19.



CPU721 Assembly drawing.

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DC - FUNCTIONS
0	CONDITION GROUP 4 BANK 0	; RESTART ADD ; LD DATA REG ; LD MAR ; CLR INTEN ; TEST MEM
1	JUMP TO START ADD	; LD IR ; MEM READ ; LD DATA REG ; RQENBX ; FETCH
130	CONDITION GROUP 4 BANK 0	; BLOCK BIT 0 ; 3 STATE EN ; LD MAR ; CIN
210	DIRECT	; LD MAR ; CLR INTEN ; 3 STATE EN ; TEST MEM
213	DIRECT	; RESTART ADD ; TEST MEM



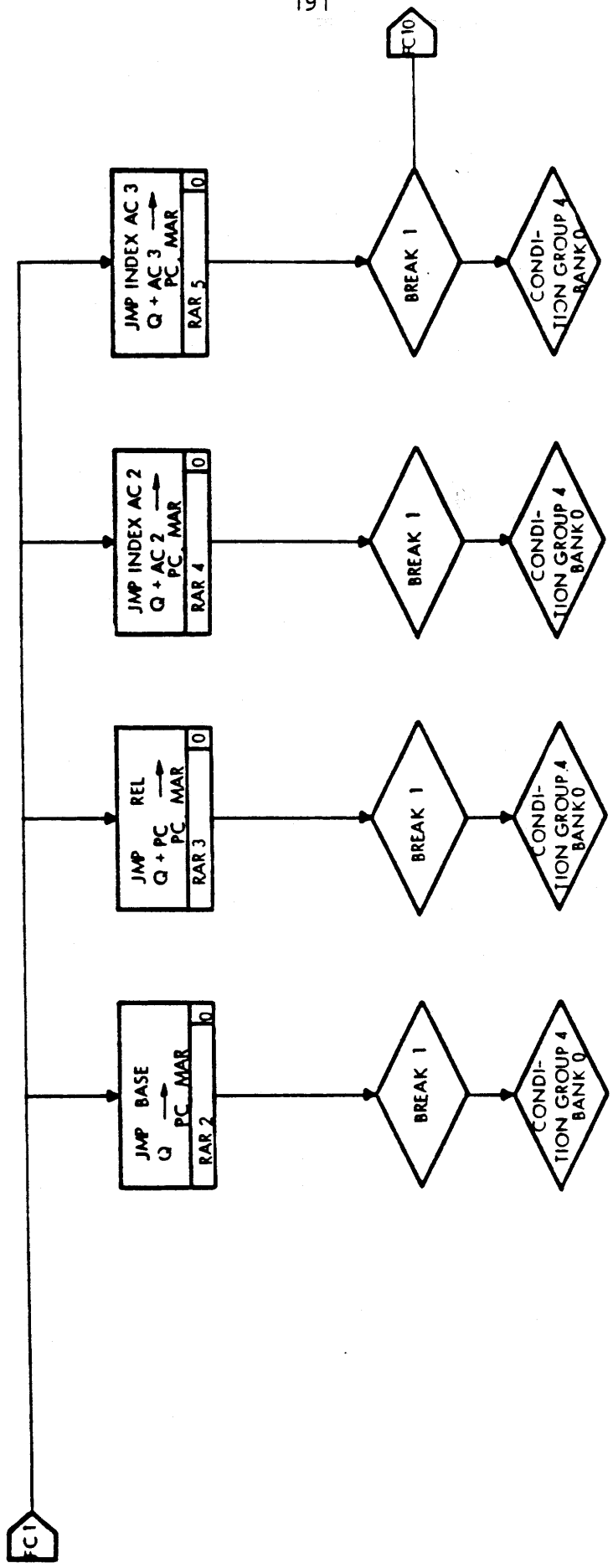
Unit CPU721

Dwg. No R13523

MICROPROGRAM FLOWCHART
OP FETCH, RESTART, SKIP
Flow Diagram

FC 1

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
2	CONDITION GROUP 4 BANK 0	; 3 STATE EN ; LD MAR ; BLOCK BIT 0
3	CONDITION GROUP 4 BANK 0	; 3 STATE EN ; LD MAR ; BLOCK BIT 0
4	CONDITION GROUP 4 BANK 0	; 3 STATE EN ; LD MAR ; BLOCK BIT 0
5	CONDITION GROUP 4 BANK 0	; 3 STATE EN ; LD MAR ; BLOCK BIT 0



Unit
CPU 721

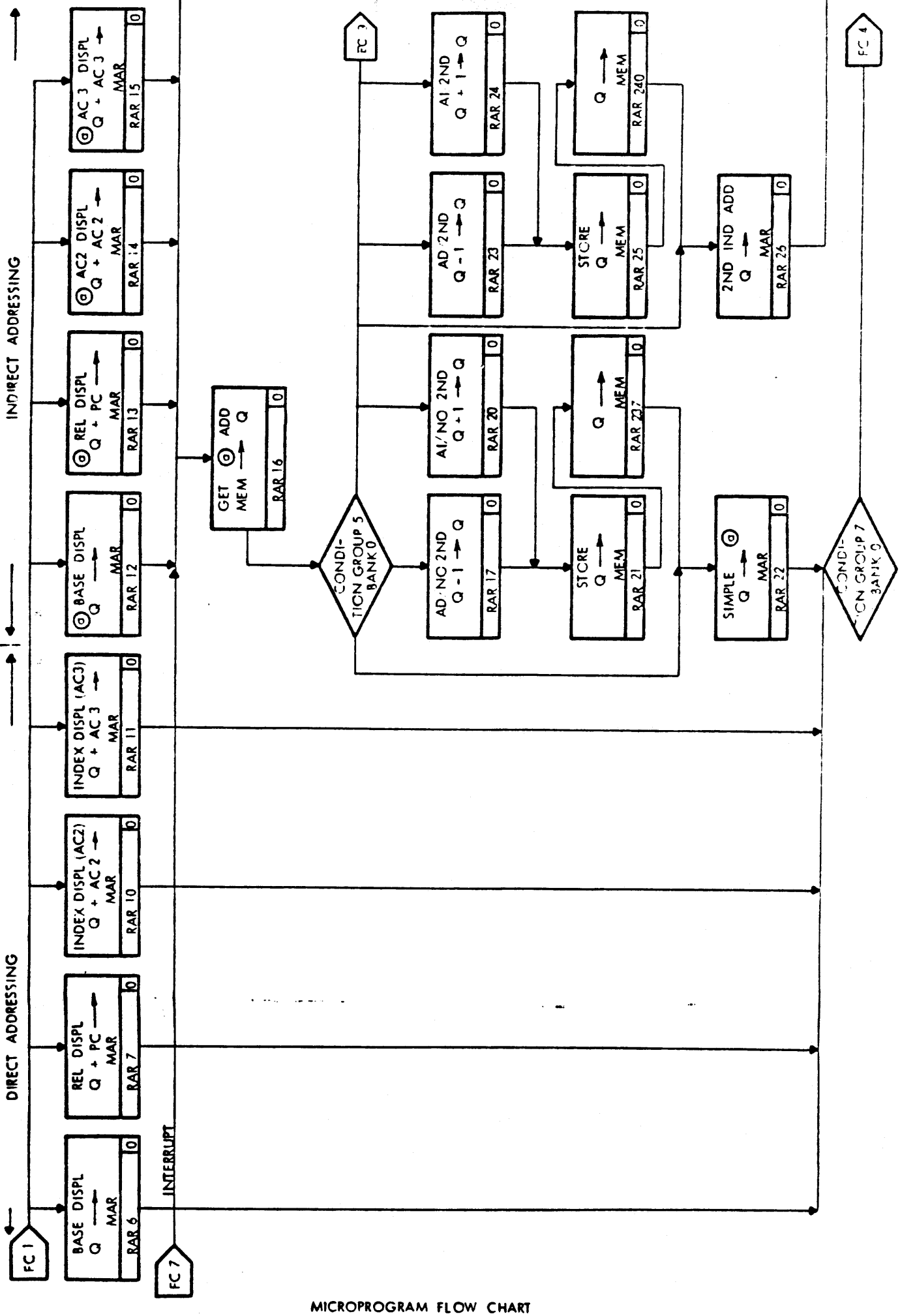
Dwg. No.
R13524

MICROPROGRAM FLOW CHART
JMP DIRECT EXECUTE

Flow Diagram

FC 2

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
6,7	CONDITION GROUP 7 BANK 0	-, TEST MEM -, LD MAR -, BLOCK BIT 0
10,11	CONDITION GROUP 7 BANK 0	-, TEST MEM -, LD MAR
12,13,14,15	DIRECT	-, LD MAR -, 3 STATE EN -, DEFER -, BLOCK BIT 0
16	CONDITION GROUP 5 BANK 0	-, MEM READ -, LD DATA REG
17,23	DIRECT	-, LD DATA REG -, 3 STATE EN
20,24	DIRECT	-, LD DATA REG -, CIN -, 3 STATE EN
21,25, 237, 240	DIRECT	-, MEM WRITE -, TEST MEM -, LD DATA REG
22	CONDITION GROUP 7 BANK 0	-, LD MAR -, 3 STATE EN -, BLOCK BIT 0
26	DIRECT	-, 3 STATE EN -, LD MAR -, DEFER -, BLOCK BIT 0



MICROPROGRAM FLOW CHART
MEMORY REFERENCE ADDRESS CALCULATION

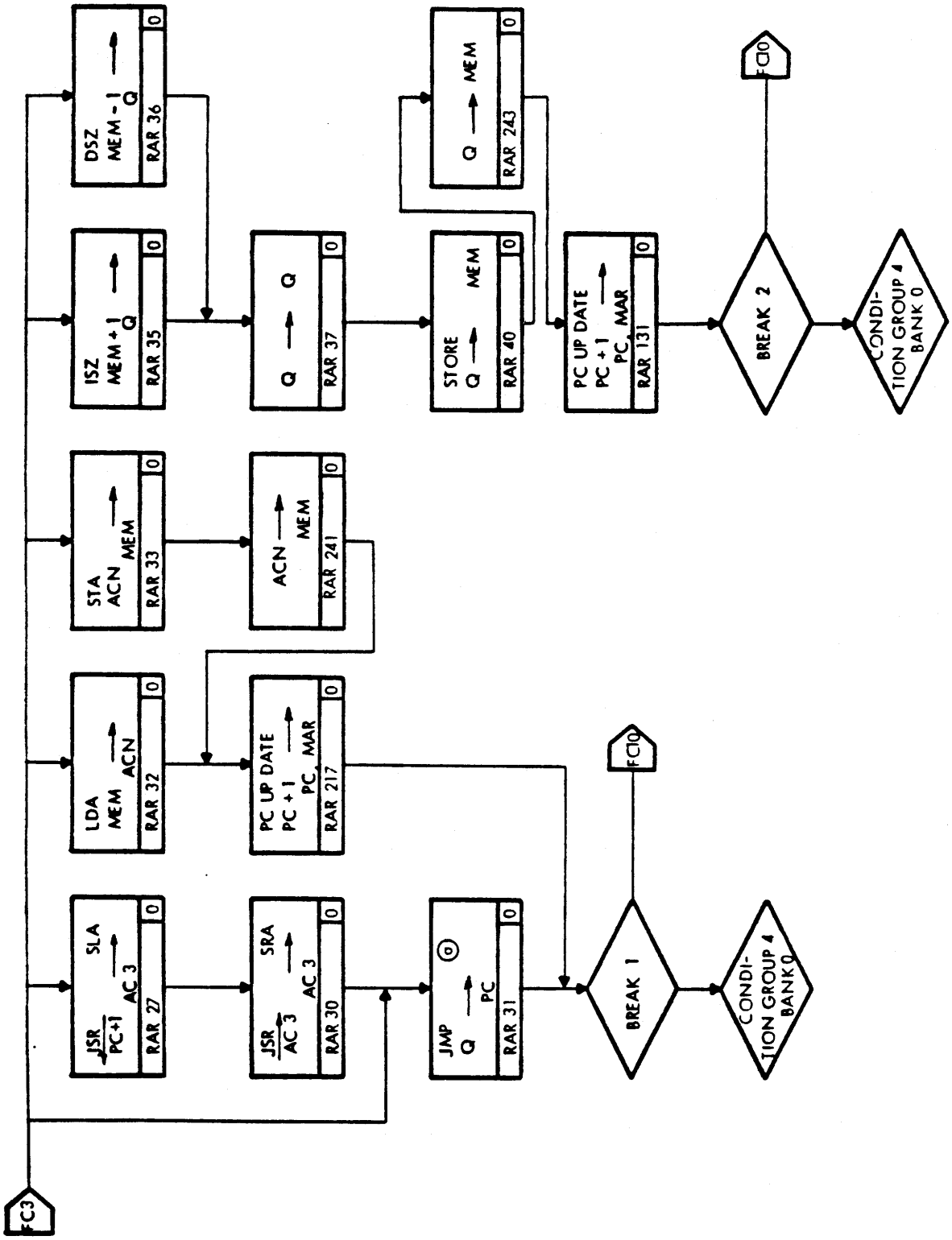
FC 3

CPU 721

R13525

Flow Diagram

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DC - FUNCTIONS
27	DIRECT	; LD DATA REG ; CIN ; 3 STATE EN ; ALU FLAG CLK
30	DIRECT	; 3 STATE EN ; LD DATA REG ; RESET BIT 0
31	CONDITION GROUP 4 BANK 0	; LD MAR ; 3 STATE EN ; BLOCK BIT 0
32	DIRECT	; MEM READ ; GATE ACN ; LD DATA REG
33, 241	DIRECT	; LD DATA REG ; MEM WRITE ; GATE ACN ; 3 STATE EN
35	DIRECT	; MEM READ ; CIN ; ALU= 0 CLK ; LD DATA REG
36	DIRECT	; LD DATA REG ; MEM READ ; ALU= 0 CLK
37	DIRECT	; ALU= 0 CLK ; TEST MEM ; LD DATA REG
40, 243	DIRECT	; LD DATA REG ; MEM WRITE ; 3 STATE EN ; SKIP EN
131	CONDITION GROUP 4 BANK 0	; 3 STATE EN ; LD MAR ; CIN ; BLOCK BIT 0
217	CONDITION GROUP 4 BANK 0	; LD DATA REG ; LD MAR ; CIN ; 3 STATE EN ; BLOCK BIT 0



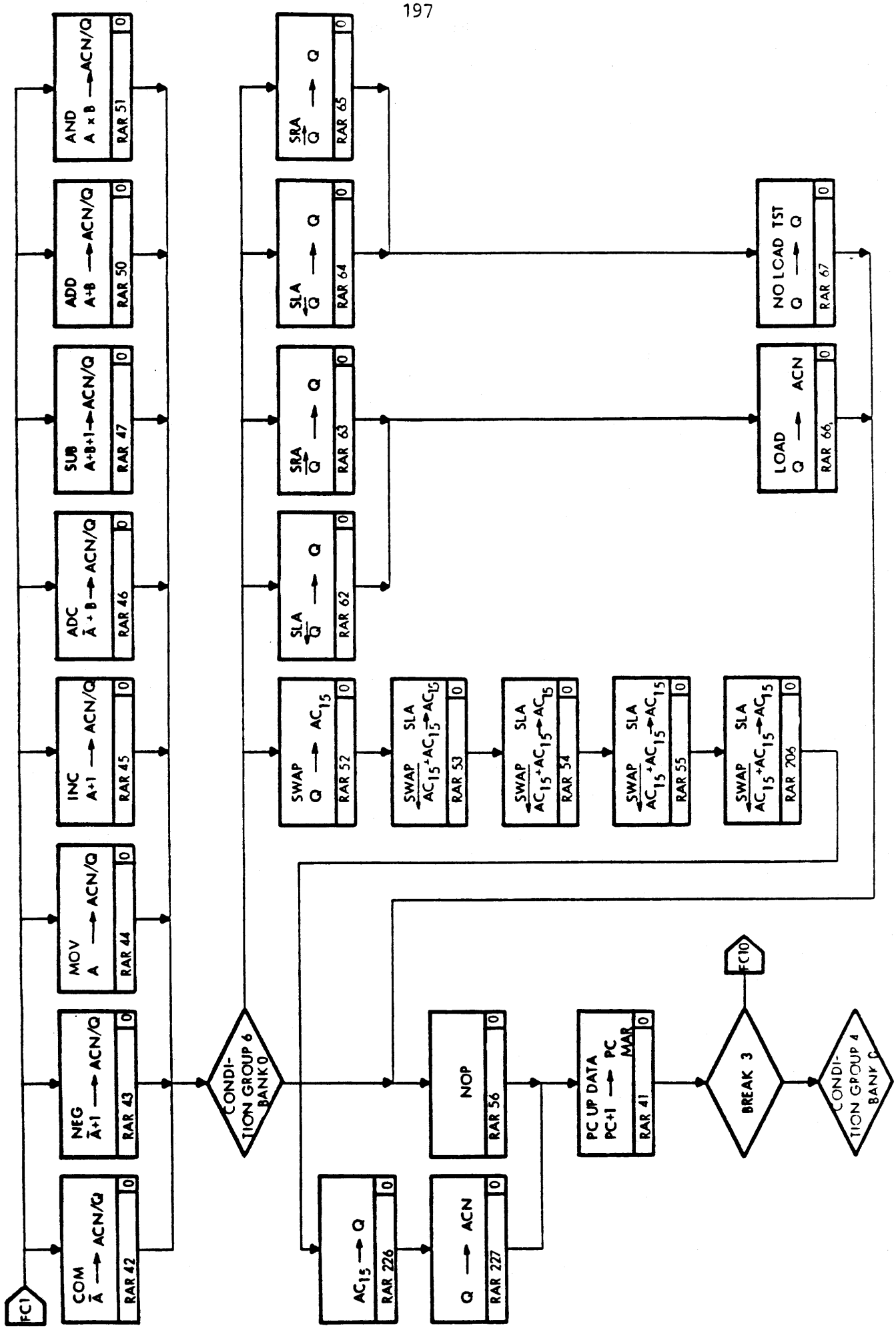
Unit CPU 721
Dwg. No. R13526

MICROPROGRAM FLOW CHART
MEMORY REFERENCE EXECUTE

Flow Diagram

FC 4

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
42 51	CONDITION GROUP 6 BANK 0	-, 3 STATE EN -, ALU = 0 CLK -, ALU FLAG CLK -, ALCCON -, GATE ACN -, LD DATA REG
43,45,47	CONDITION GROUP 6 BANK 0	-, 3 STATE EN -, ALU = 0 CLK -, ALU FLAG CLK -, ALCCON -, GATE ACN COUT EN -, CIN -, LD DATA REG
44,46,50	CONDITION GROUP 6 BANK 0	-, 3 STATE EN -, ALU = 0 CLK -, ALU FLAG CLK -, ALC CON -, GATE ACN COUT EN -, LD DATA REG
52	DIRECT	-, 3 STATE EN -, LD DATA REG
53,54,55,206	DIRECT	SWAP EN -, 3 STATE EN -, LD DATA REG
56	DIRECT	-, 3 STATE EN -, SKIP EN
62,64	DIRECT	COUT EN -, 3 STATE EN SLA EN -, LD DATA REG
63,65	DIRECT	-, LD DATA REG SRA EN -, 3 STATE EN COUT EN
66	DIRECT	-, LD DATA REG -, GATE ACN -, ALU = 0 CLK -, 3 STATE EN
67	DIRECT	-, LD DATA REG -, 3 STATE EN -, ALU = 0 CLK
226	DIRECT	COUT EN
227	DIRECT	-, GATE ACN COUT EN -, SKIP EN
41	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR -, CIN -, CARRY LINK CLK -, BLOCK BIT 0

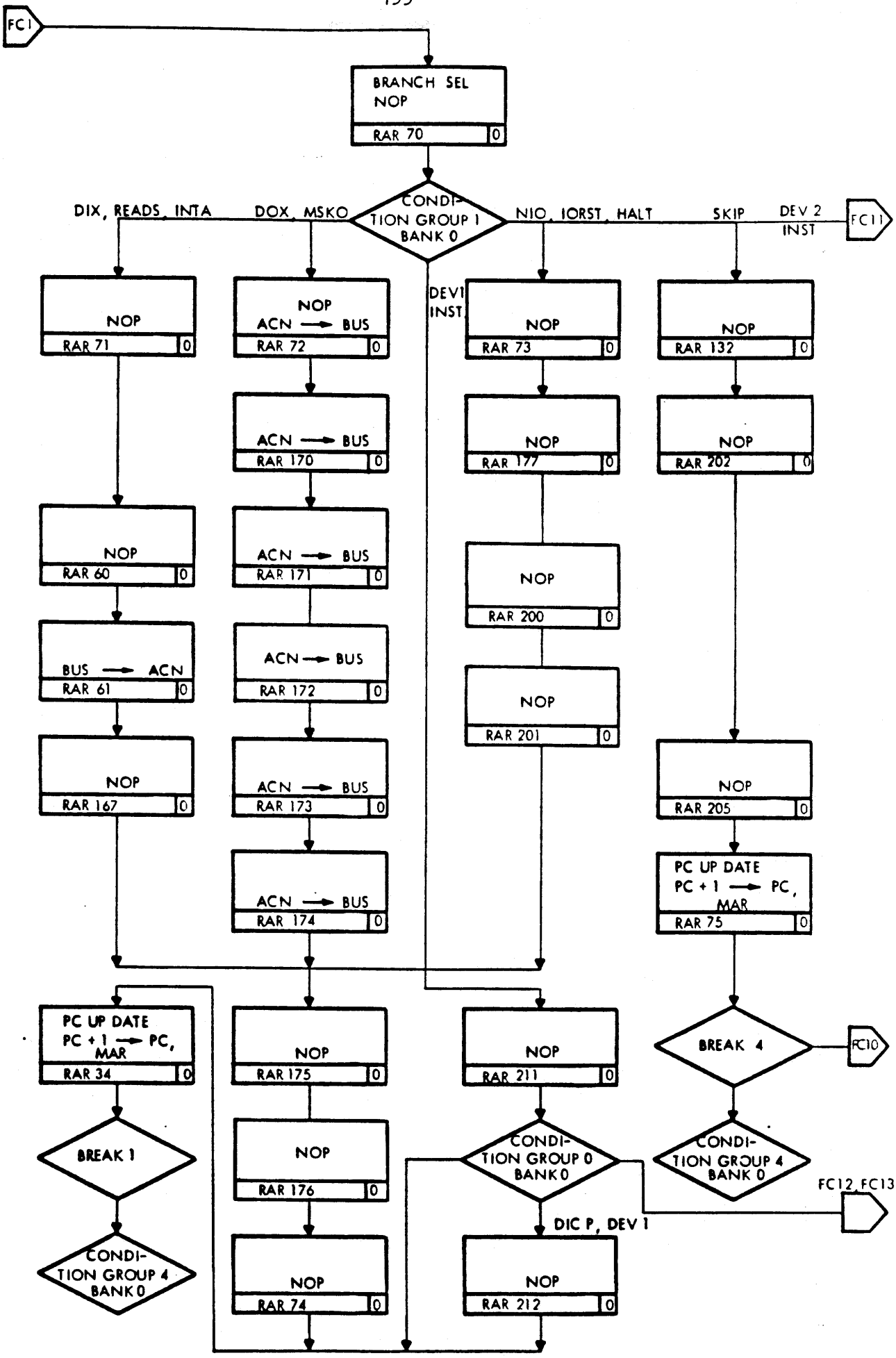


Unit CPU721
Dwg. No. R13527

MICROPROGRAM FLOW CHART
ARITHMETIC AND LOGICAL INSTRUCTIONS
Flow Diagram

FC 5

FOM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
34	CONDITION GROUP 4 BANK 0	-, I/O SLO -, LD MAR -, CIN -, 3 STATE EN - BLOCK BIT 0
60,71	DIRECT	-, LD DATA REG -, GEN IODT -, GATE IN -, I/O SLO
61	DIRECT	-, I/O SLO -, GATE ACN -, GEN IODT -, GATE IN -, LD DATA REG
70	CONDITION GROUP 1 BANK 0	-, I/O SLO
72	DIRECT	-, LD DATA REG -, 3 STATE EN -, I/O SLO -, GATE ACN
73	DIRECT	-, GEN IODT -, I/O SLO
74, ~~175~~, 176	DIRECT	-, I/O SLO -, GEN IOP
75	CONDITION GROUP 4 BANK 0	-, TEST MEM -, 3 STATE EN -, CIN -, LD MAR -, I/O SLO - BLOCK BIT 0
132, 167, 202	DIRECT	-, I/O SLO
170, 174	DIRECT	-, 3 STATE EN -, GATE OUT -, I/O SLO -, GATE ACN
171, 172, 173	DIRECT	-, 3 STATE EN -, GEN IODT -, GATE OUT -, I/O SLO -, GATE ACN
177, 200, 201	DIRECT	-, GEN IODT -, I/O SLO
205	DIRECT	-, SKIP EN -, I/O SLO
211	CONDITION GROUP 0 BANK 0	-, I/O SLO
212	DIRECT	-, I/O SLO -, SET EXTEND
175	DIRECT	-, I/O SLO -, GEN IOP -, TEST MEM

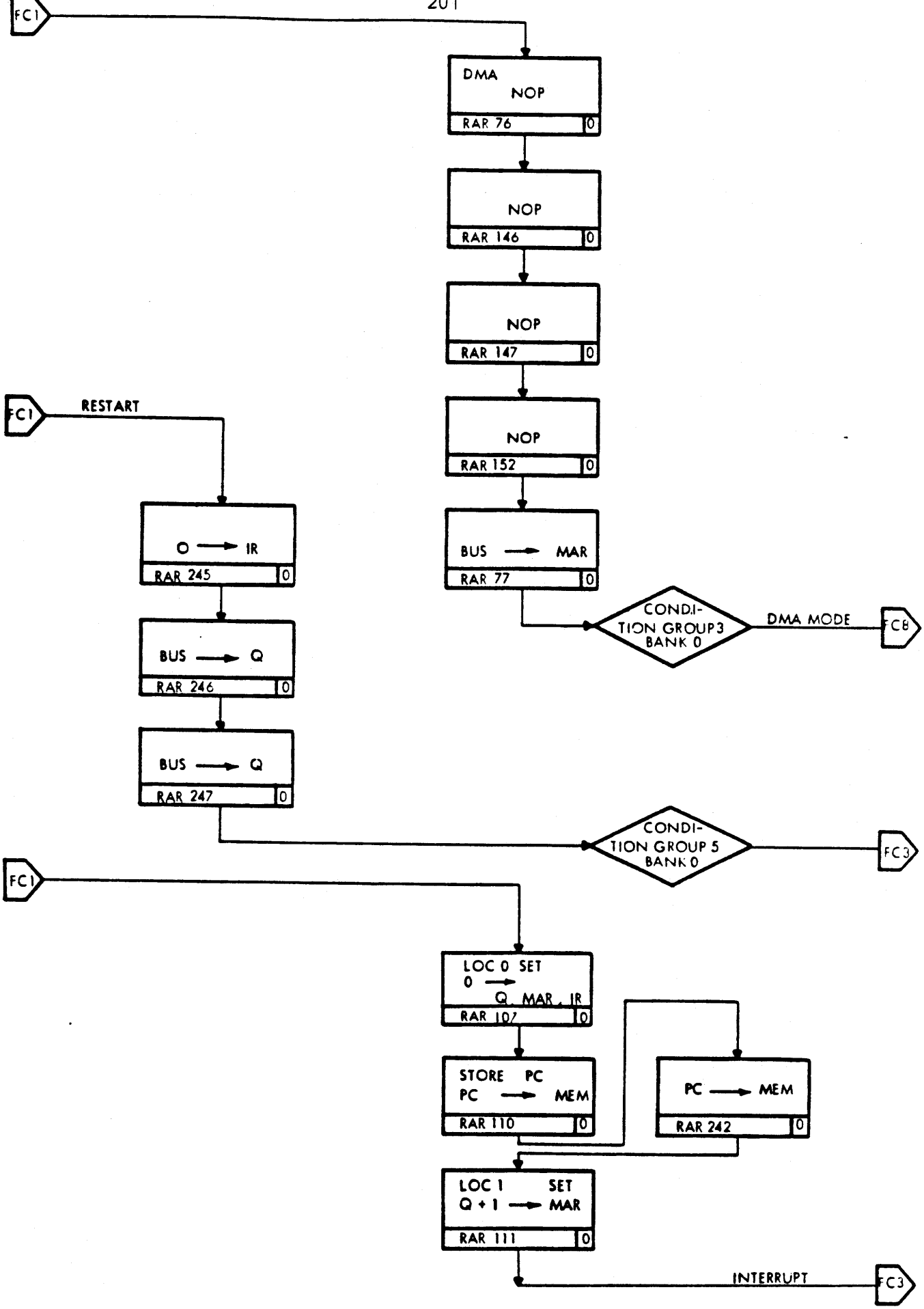


Unit
CPU721
Dwg. No.
R13528

MICROPROGRAM FLOW CHART
INPUT / OUTPUT INSTRUCTION EXECUTE
INTERRUPT AND SKIP INSTRUCTION EXECUTE
Flow Diagram

FC 6

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
76	DIRECT	-, DCHAX
77	CONDITION GROUP 3 BANK 0	-; LD MAR -; DCHAX -; GATE IN
107	DIRECT	-; LD MAR -; LD IR -; CLR INTEN
110, 242	DIRECT	-; LD DATA REG -; MEM WRITE -; CLR INTEN -, 3 STATE EN - BLOCK BIT 0
111	DIRECT	-; 3 STATE EN -; LD MAR -; CIN -; CLR INTEN
146, 147, 152	DIRECT	-; DCHAX
245	DIRECT	-; LDIR
246	DIRECT	-; RESTART ADD
247	CONDITION GROUP 5 BANK 0	-; RESTART ADD



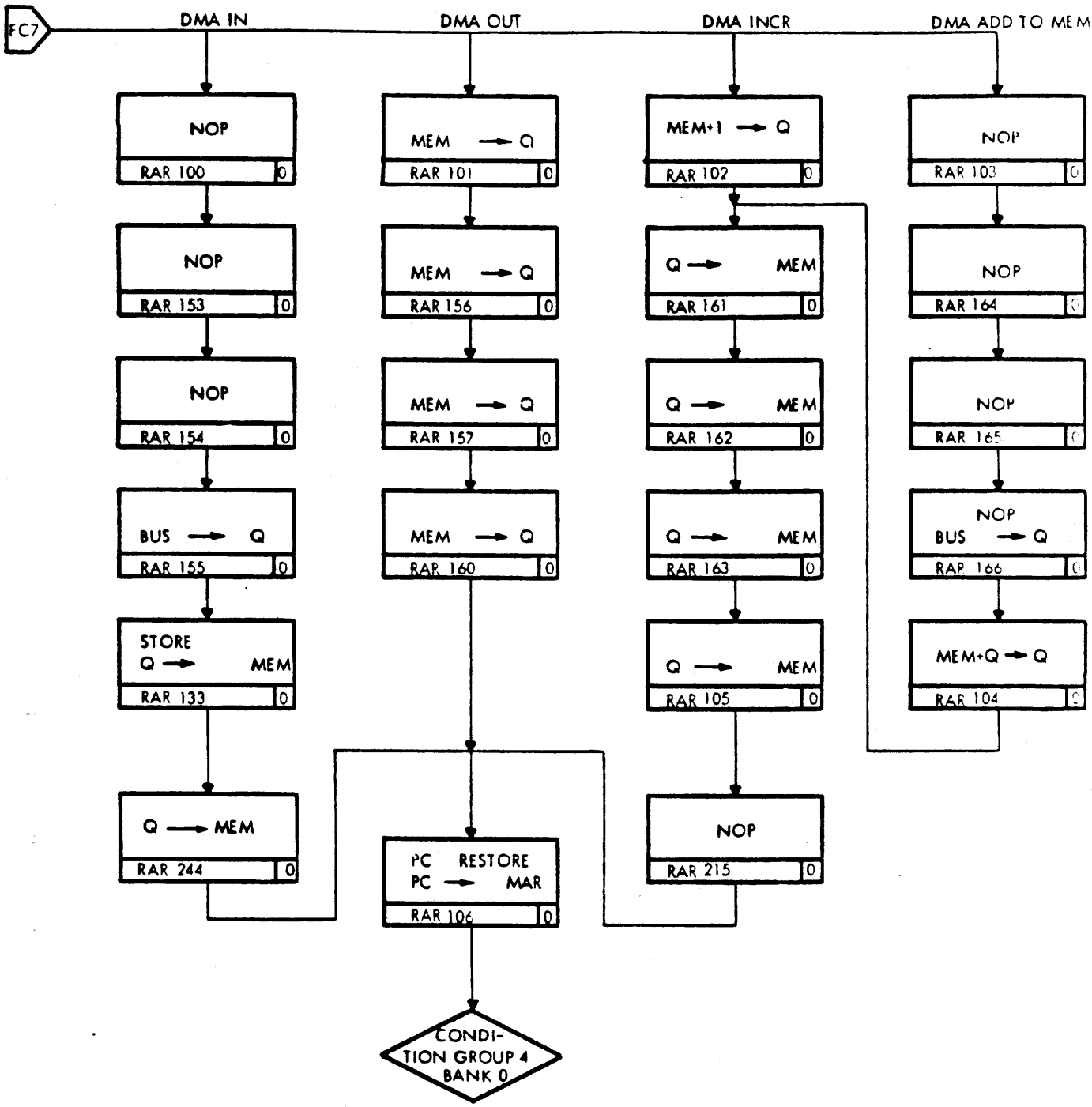
Unit
CPU 721

Dwg. No.
R13529

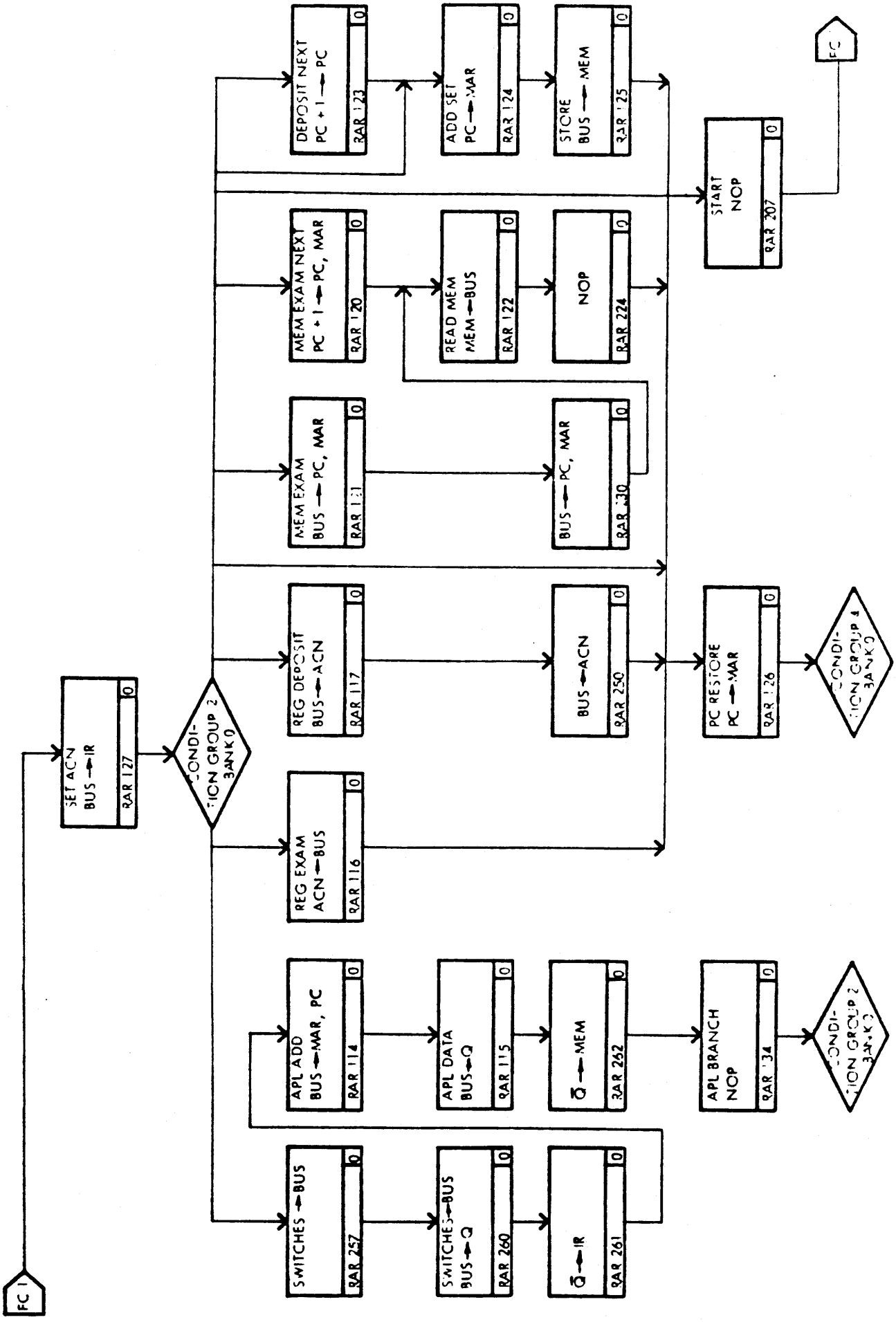
MICROPROGRAM FLOW DIAGRAM
INTERRUPT EXECUTE
DMA ADDRESS INPUT
Flow Diagram

FC 7

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
100, 103	DIRECT	-, LD DATA REG -, REQENBX -, DCHI
101	DIRECT	-, LD DATA REG -, RQENBX -, GATE OUT -, MEM READ
102	DIRECT	-, TEST MEM -, LD DATA REG -, RQENBX -, MEM READ -, CIN -, ALU FLAG CLK
104	DIRECT	-, TEST MEM -, LD DATA REG -, RQENBX -, MEM READ -, ALU FLAG CLK
105	DIRECT	-, TEST MEM -, LD DATA REG -, 3 STATE EN -, GATE OUT -, MEM READ -, DCH OFLO EN
106	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR -, BLOCK BIT 0
153, 154 164, 165	DIRECT	-, DCHI -, RQENBX
133, 244	DIRECT	-, LD DATA REG -, 3 STATE EN -, MEM WRITE
155	DIRECT	-, RQENBX -, DCHI -, GATE IN
156, 157	DIRECT	-, RQENBX -, DCHO -, GATE OUT -, MEM READ
160	DIRECT	-, RQENBX -, GATE OUT -, MEM READ
161	DIRECT	-, 3 STATE EN -, RQENBX -, GATE OUT -, MEM WRITE
162	DIRECT	-, RQENBX -, 3 STATE EN -, DCHO -, GATE OUT -, MEM WRITE
163	DIRECT	-, DCHO -, GATE OUT -, MEM WRITE -, DCH OFLO EN
166	DIRECT	-, TEST MEM -, RQENBX -, DCHI -, GATE IN
215	DIRECT	-, DCH OFLO EN


FC7
DMA IN
DMA OUT
DMA INCR
DMA ADD TO MEM
NOP
RAR 100 0
MEM → Q
RAR 101 0
MEM+1 → Q
RAR 102 0
NOP
RAR 103 0
NOP
RAR 153 0
MEM → Q
RAR 156 0
Q → MEM
RAR 161 0
NOP
RAR 164 0
NOP
RAR 154 0
MEM → Q
RAR 157 0
Q → MEM
RAR 162 0
NOP
RAR 165 0
BUS → Q
RAR 155 0
MEM → Q
RAR 160 0
Q → MEM
RAR 163 0
NOP
BUS → Q
RAR 166 0
STORE
Q → MEM
RAR 133 0
Q → MEM
RAR 105 0
MEM+Q → Q
RAR 104 0
Q → MEM
RAR 244 0
PC RESTORE
PC → MAR
RAR 106 0
NOP
RAR 215 0
CONDI-
TION GROUP 4
BANK 0

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
114	DIRECT	-, TEST MEM -, LD MAR -, GATE APL ADD
115	DIRECT	-, GATE APL DATA -, LD DATA REG
116	DIRECT	-, FPOP END -, LD DATA REG -, PAN ACN -, GATE ACN -, 3 STATE EN
117, 250	DIRECT	-, PAN ACN EN -, GATE ACN -, GATE DATA SW -, FPOP END -, LD DATA REG
120	DIRECT	-, TEST MEM -, LD MAR -, CIN -, 3 STATE EN
121, 230	DIRECT	-, LD MAR -, TEST MEM -, GATE DATA SW
122	DIRECT	-, FPOP END -, LD DATA REG -, TEST MEM -, MEM READ
123	DIRECT	-, 3 STATE EN -, CIN -, LD DATA REG
124	DIRECT	-, TEST MEM -, LD MAR -, 3 STATE EN -, -BLOCK BIT 0
125	DIRECT	-, MEM WRITE -, TEST MEM -, GATE DATA SW -, FPOP EN -, LD DATA REG
126	CONDITION GROUP 4 BANK 0	-, TEST MEM -, LD MAR -, 3 STATE EN
127	CONDITION GROUP 2 BANK 0	-, LD IR -, GATE REG EN
134	CONDITION GROUP 2 BANK 0	-, CIN
207	DIRECT	-, RESTART ADD
224	DIRECT	
257, 260	DIRECT	-, GATE DATA SW
261	DIRECT	-, LD IR -, LD DATA REG
262	DIRECT	-, TEST MEM -, MEM WRITE -, LD DATA REG

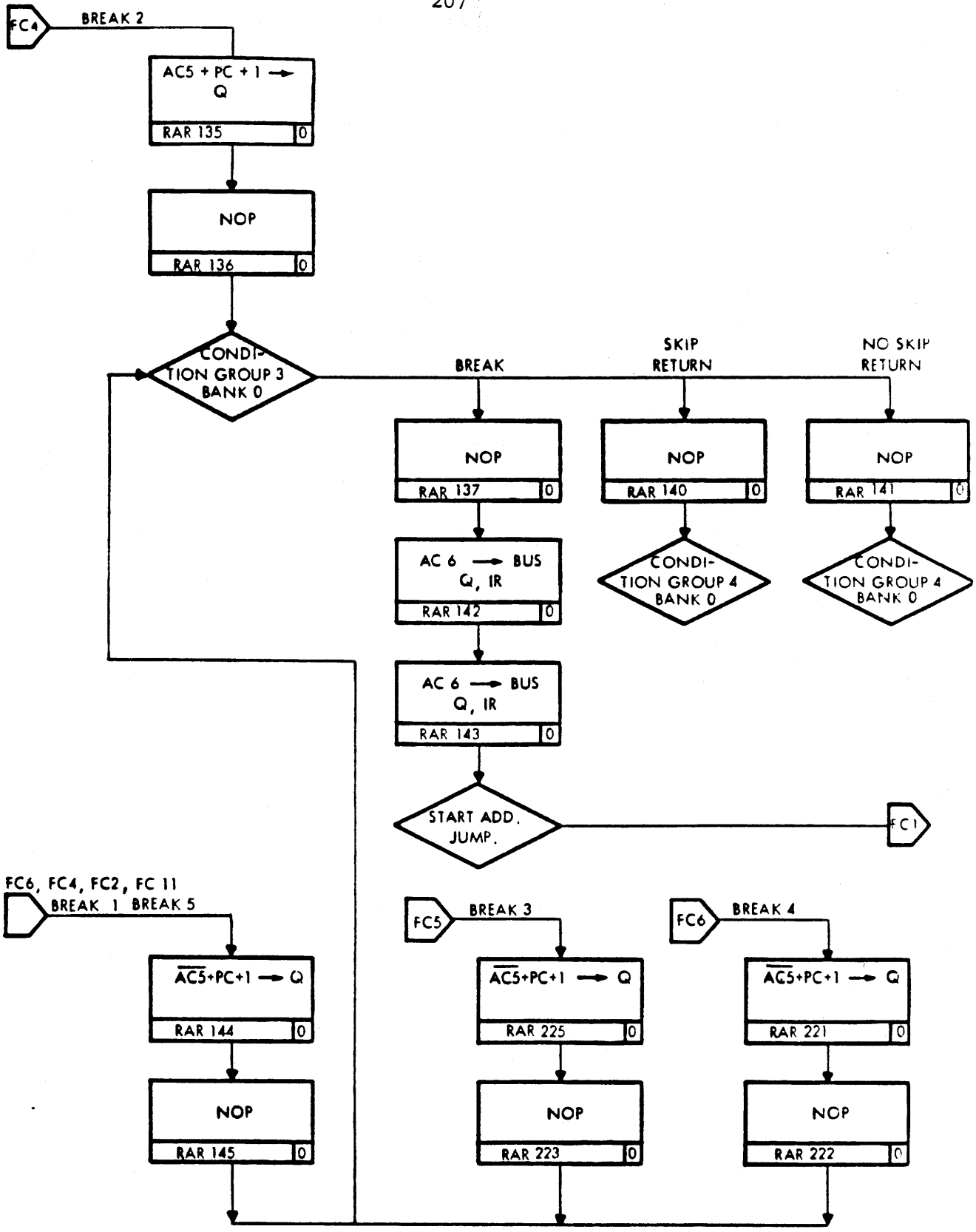


CPU721
R13531

MICROPROGRAM FLOW CHART
FRONT PANEL OPERATIONS
Flow Diagram

FC 9

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
135	DIRECT	EN BREAK -, 3 STATE EN -, SKIP EN -, CIN
136	CONDITION GROUP 3 BANK 0	-, SKIP EN EN BREAK
137	DIRECT	-, 3 STATE EN -, CIN
140	CONDITION GROUP 4 BANK 0	
141	CONDITION GROUP 4 BANK 0	
142	DIRECT	-, 3 STATE EN -, FETCH -, LD IR -, LD DATA REG
143	JUMP TO START ADD	-, 3 STATE EN -, FETCH -, LD IR -, LD DATA REG
144	DIRECT	-, 3 STATE EN -, CIN EN BREAK
145	DIRECT	EN BREAK
221	DIRECT	EN BREAK -, 3 STATE EN -, SKIP EN -, I/O SLO -, CIN
222	CONDITION GROUP 3 BANK 0	-, SKIP EN -, I/O SLO EN BREAK
223	CONDITION GROUP 3 BANK 0	EN BREAK -, SKIP EN -, CARRY LINK CLK
225	DIRECT	-, 3 STATE EN -, SKIP EN -, CIN -, CARRY LINK CLK



Unit
CPU 721
Dwg. No.
R13532

MICROPROGRAM FLOWCHART
BREAK TEST EXECUTE
Flow Diagram

FC 10

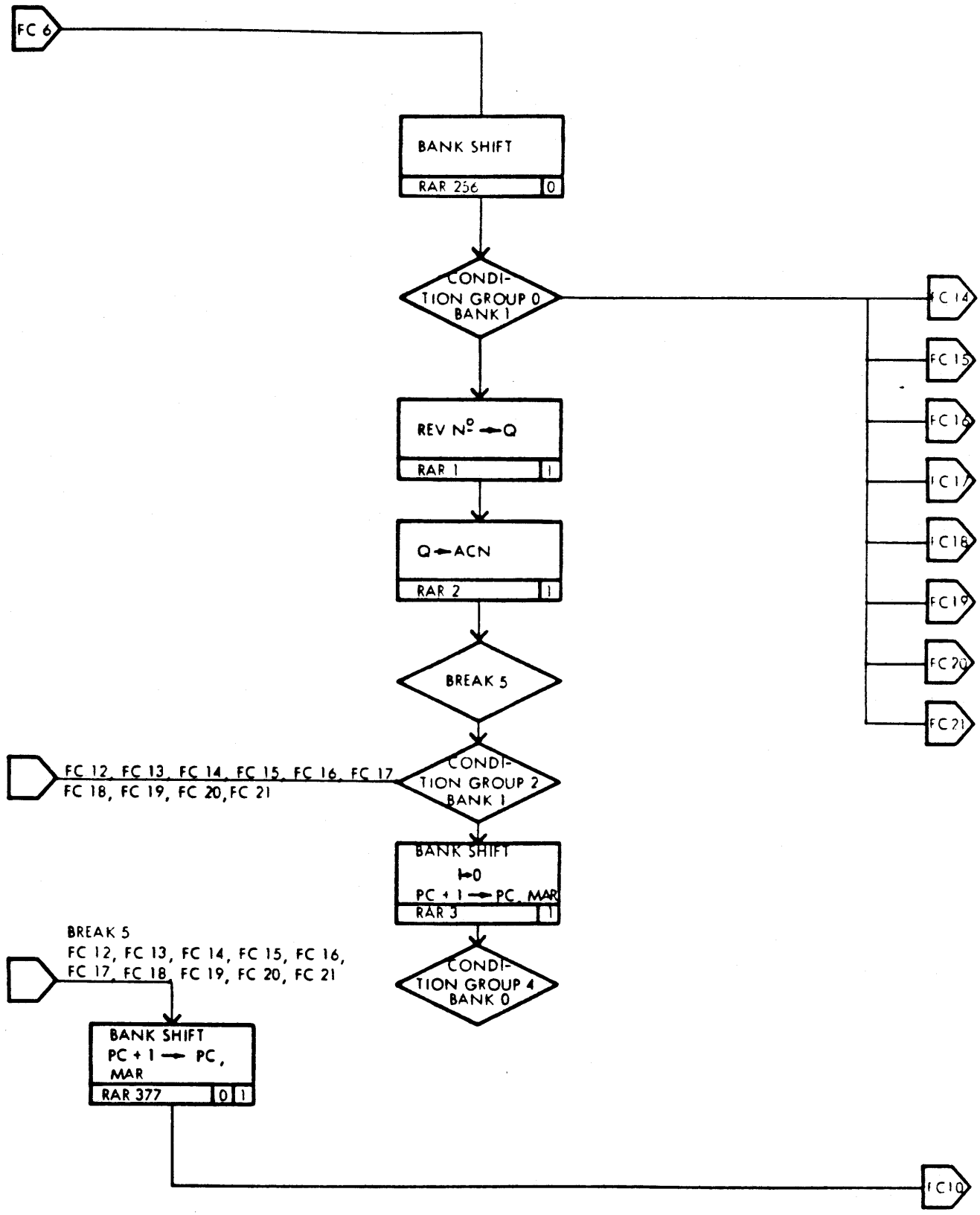
ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
1	DIRECT	-, LD DATA REG -, ENABLE CONSTANT
2	CONDITION GROUP 2 BANK 1	-, LD DATA REG -, 3 STATE EN -, ALC CON
3	CONDITION GROUP 4 BANK 0	-, LD DATA REG -, 3 STATE EN -, LD MAR -, C IN
256	CONDITION GROUP 0 BANK 1	
377	DIRECT	-, 3 STATE EN -, LD MAR -, C IN

A/S REGNE TRALEN

Designed by

Drawn by

Dwg. Office Check


FC 6
BANK SHIFT
RAR 256 0
CONDI-
TION GROUP 0
BANK 1
FC 14
FC 15
FC 16
FC 17
FC 18
FC 19
FC 20
FC 21
REV N° → Q
RAR 1 1
Q → ACN
RAR 2 1
BREAK 5
FC 12, FC 13, FC 14, FC 15, FC 16, FC 17
FC 18, FC 19, FC 20, FC 21
CONDI-
TION GROUP 2
BANK 1
BANK SHIFT
1→0
PC + 1 → PC, MAR
RAR 3 1
BREAK 5
FC 12, FC 13, FC 14, FC 15, FC 16,
FC 17, FC 18, FC 19, FC 20, FC 21
CONDI-
TION GROUP 4
BANK 0
BANK SHIFT
PC + 1 → PC,
MAR
RAR 377 0 1
FC 10

ROM ADDRESS (OCTAL) 210	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
42	DIRECT	-, RESET BIT 0
43, 46, 47	DIRECT	-, ENABLE CONSTANT
44	DIRECT	-, LD DATA REG -, 3 STATE EN -, LD MAR
45	DIRECT	-, RQEN BX -, MEM READ
50	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, 3 STATE EN -, LD MAR
51	DIRECT	-, LD DATA REG -, 3 STATE EN
52, 53, 54	DIRECT	-, LD DATA REG -, 3 STATE EN SWAP EN
55	DIRECT	-, LD DATA REG -, 3 STATE EN SWAP EN COND SEL 0
56, 57	CONDITION GROUP 6, 2 BANK 1	
60	DIRECT	-, LD DATA REG -, 3 STATE EN COND SEL 0
375	DIRECT	-, 3 STATE EN -, ALU FLAG CLK

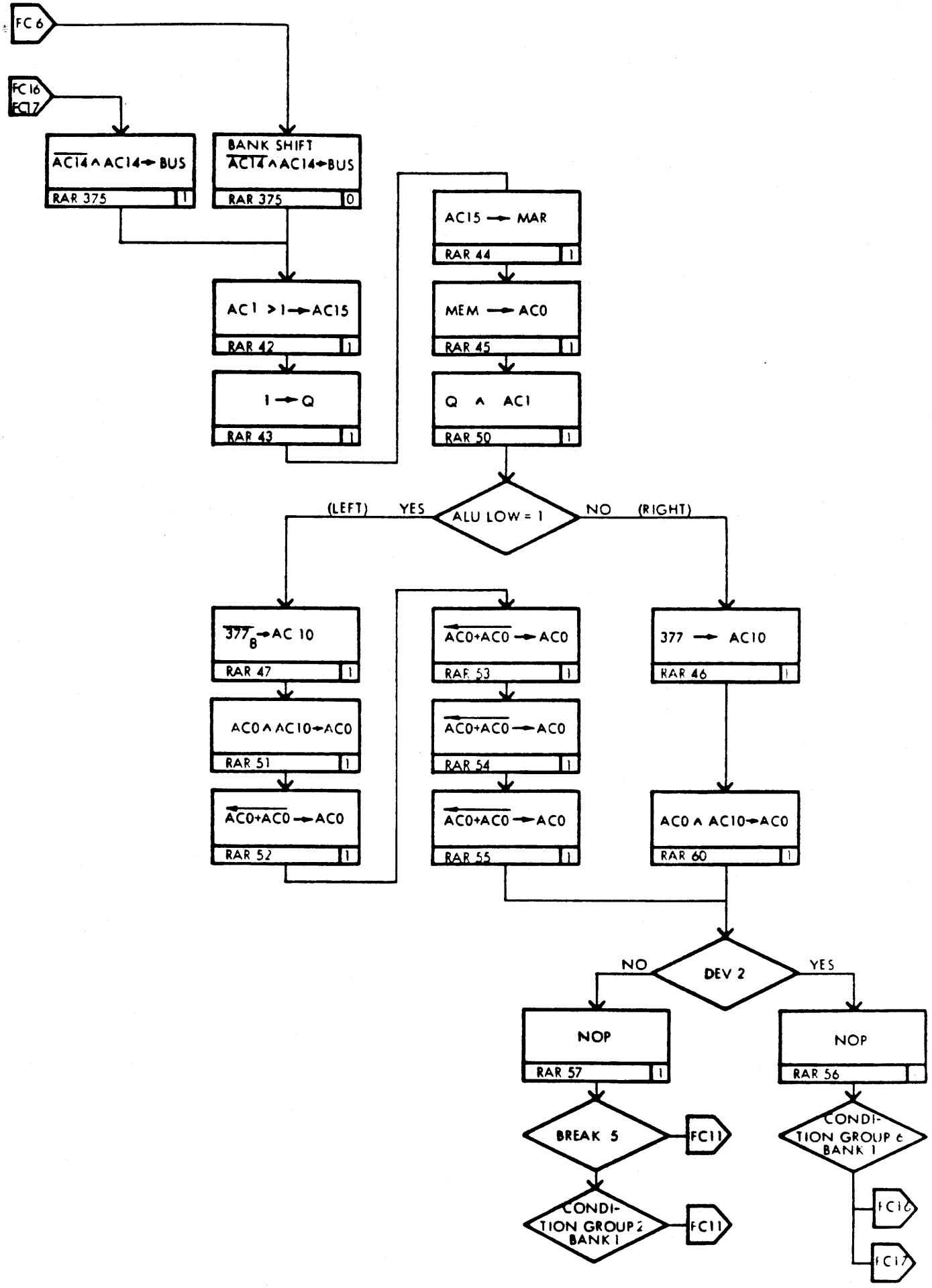
A/S REGNECENTRALEN

Designed by

Drawn by

Dwg. Office Check

Unit CPU721	LOAD BYTE	FC 12
Dwg. No. R21396	Signal List	12 of 21


FC 6
FC 16
FC 17
$\overline{AC14} \wedge AC14 \rightarrow BUS$
RAR 375 1
BANK SHIFT
$\overline{AC14} \wedge AC14 \rightarrow BUS$
RAR 375 0
AC1 > 1 → AC15
RAR 42 1
1 → Q
RAR 43 1
AC15 → MAR
RAR 44 1
MEM → AC0
RAR 45 1
Q ∧ AC1
RAR 50 1
(LEFT) YES ALU LOW = 1 NO (RIGHT)
$377_8 \rightarrow$ AC 10
RAR 47 1
AC0 ∧ AC10 → AC0
RAR 51 1
$\overleftarrow{AC0+AC0} \rightarrow$ AC0
RAR 52 1
$\overleftarrow{AC0+AC0} \rightarrow$ AC0
RAR 53 1
$\overleftarrow{AC0+AC0} \rightarrow$ AC0
RAR 54 1
$\overleftarrow{AC0+AC0} \rightarrow$ AC0
RAR 55 1
377 → AC10
RAR 46 1
AC0 ∧ AC10 → AC0
RAR 60 1
NO DEV 2 YES
NOP
RAR 57 1
NOP
RAR 56
BREAK 5
FC11
CONDI-TION GROUP 2 BANK 1
FC11
CONDI-TION GROUP 6 BANK 1
FC16
FC17

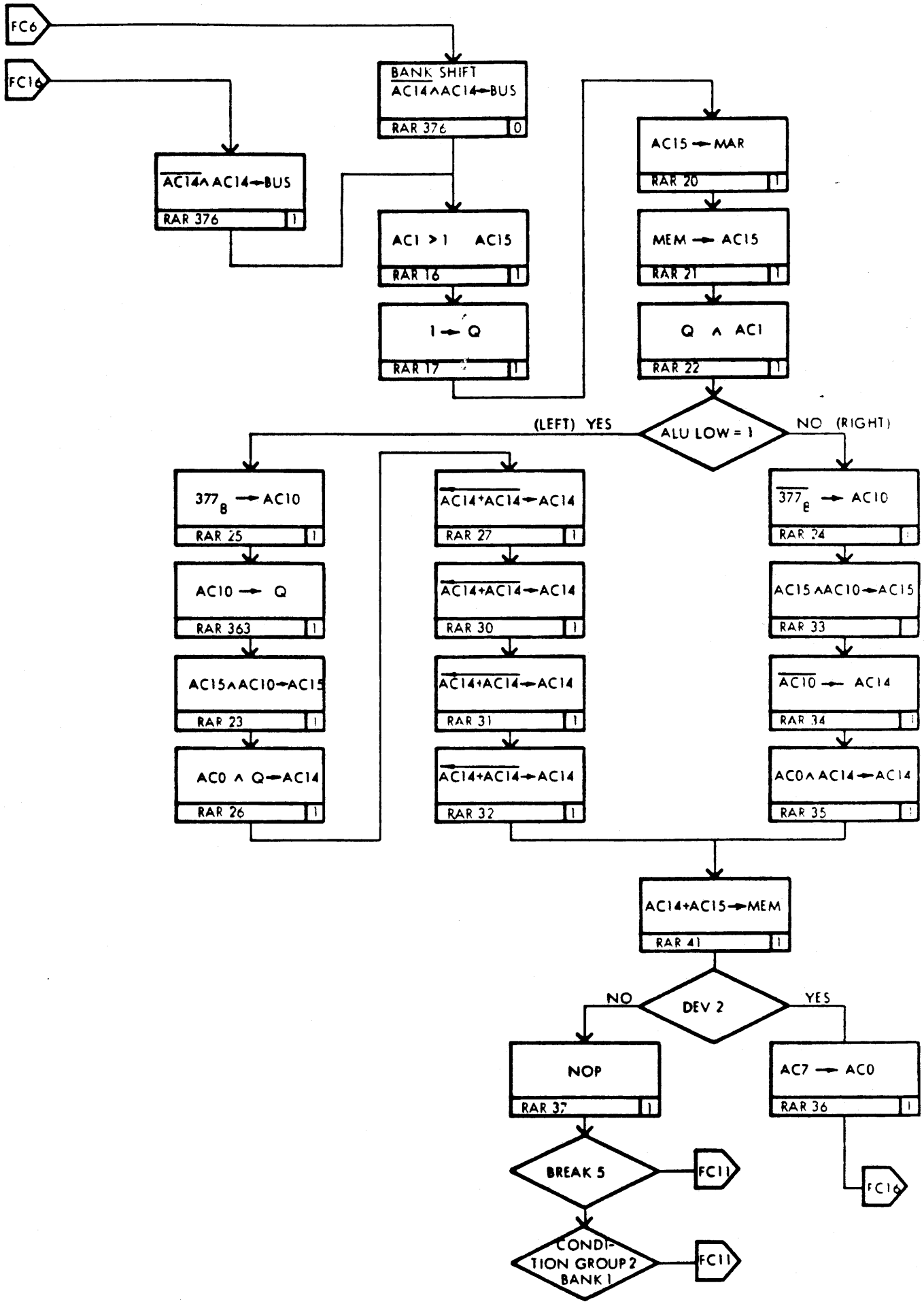
ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
16	DIRECT	-, RESET BIT 0
17, 24, 25	DIRECT	-, ENABLE CONSTANT
20	DIRECT	-, LD DATA REG -, 3 STATE EN -, LD MAR
21	DIRECT	-, LD DATA REG -, MEM READ
22	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, 3 STATE EN
23	DIRECT	-, LD DATA REG
26	DIRECT	-, LD DATA REG -, 3 STATE EN
27, 30, 31, 32	DIRECT	-, LD DATA REG -, 3 STATE EN SWAP EN
33, 34, 35, 363	DIRECT	
36	DIRECT	COND SEL 1
37	CONDITION GROUP 2 BANK 1	
41	DIRECT	-, LD DATA REG -, 3 STATE EN -, RQEN BX -, MEM WRITE COND SEL 0 -, TEST MEM
376	DIRECT	-, 3 STATE EN -, ALU FLAG CLK

A/S REGNECENTRALEN

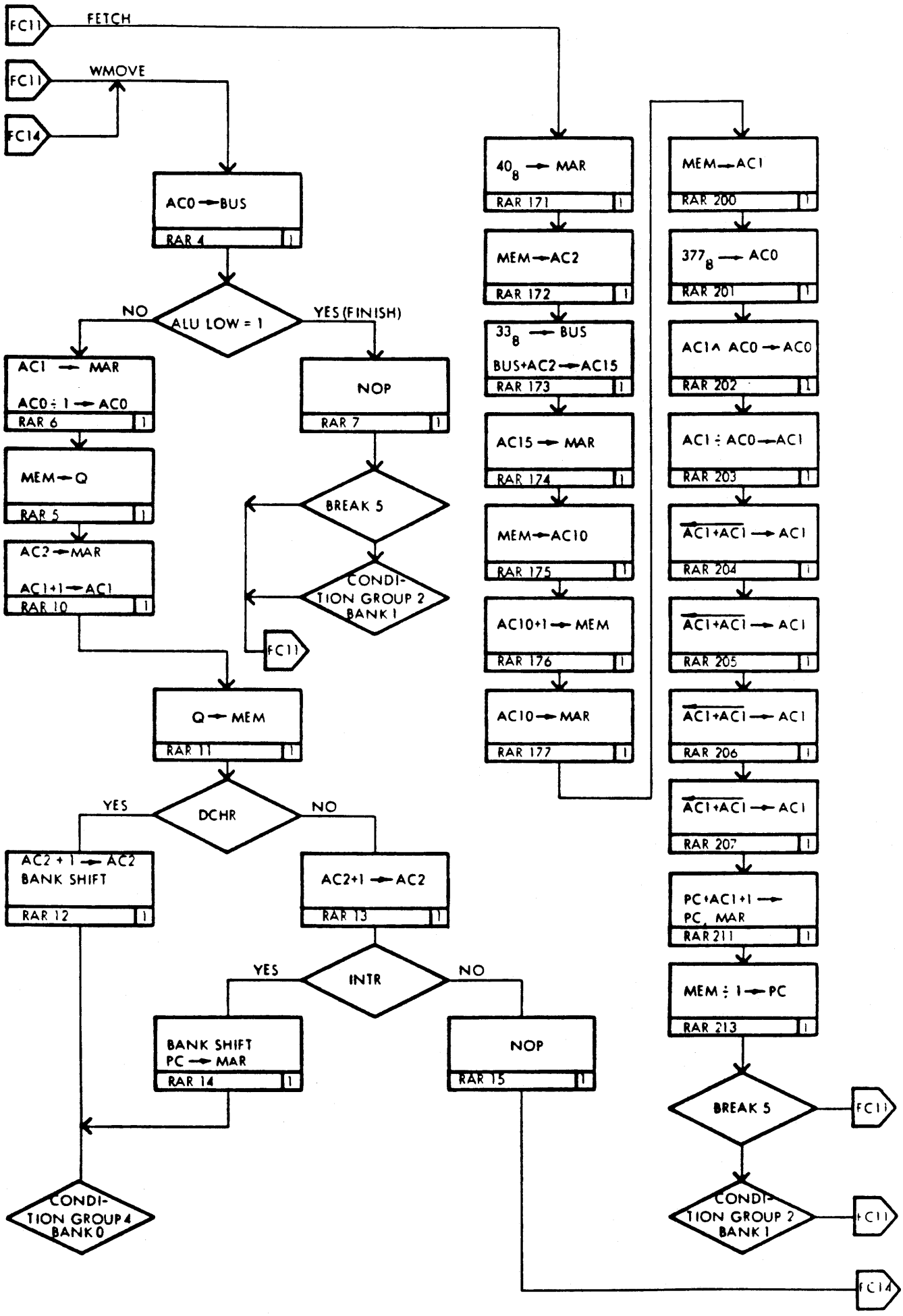
Designed by

Drawn by

Dwg. Office Check


FC6
FC16
BANK SHIFT
AC14∧AC14←BUS
RAR 376 0
AC14∧AC14←BUS
RAR 376 1
AC1 > 1 AC15
RAR 16 1
1 ← Q
RAR 17 1
AC15 ← MAR
RAR 20 1
MEM ← AC15
RAR 21 1
Q ∧ AC1
RAR 22 1
ALU LOW = 1
(LEFT) YES
NO (RIGHT)
377B ← AC10
RAR 25 1
AC10 ← Q
RAR 363 1
AC15∧AC10←AC15
RAR 23 1
AC0 ∧ Q←AC14
RAR 26 1
AC14+AC14←AC14
RAR 27 1
AC14+AC14←AC14
RAR 30 1
AC14+AC14←AC14
RAR 31 1
AC14+AC14←AC14
RAR 32 1
377B ← AC10
RAR 24 1
AC15∧AC10←AC15
RAR 33
AC10 ← AC14
RAR 34 1
AC0∧AC14←AC14
RAR 35 1
AC14+AC15←MEM
RAR 41 1
NO
DEV 2
YES
NOP
RAR 37 1
AC7 ← AC0
RAR 36 1
BREAK 5
FC11
CONDI-
TION GROUP 2
BANK 1
FC11
FC16

ROM ADDRESS (OCTAL) 214	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
4	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, 3 STATE EN -, LD MAR
5	DIRECT	-, MEM READ
6	DIRECT	-, LD DATA REG -, 3 STATE EN -, LD MAR
7	CONDITION GROUP 2 BANK 1	
10	DIRECT	-, LD DATA REG -, 3 STATE EN -, RQEN BX -, LD MAR -, C IN
11	DIRECT	-, LD DATA REG COND SEL 1 -, 3 STATE EN -, MEM WRITE -, TEST MEM
12	CONDITION GROUP 4 BANK 0	-, C IN
13	DIRECT	COND SEL 2 -, C IN
14	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR
15, 202	DIRECT	
171	DIRECT	-, LD MAR -, ENABLE CONSTANT
172, 175, 200	DIRECT	-, LD DATA REG -, MEM READ -, TEST MEM
173, 201	DIRECT	-, ENABLE CONSTANT
174, 177	DIRECT	-, 3 STATE EN -, LD MAR
176	DIRECT	-, 3 STATE EN -, MEM WRITE -, TEST MEM -, C IN
203	DIRECT	-, C IN
204, 205, 206, 207	DIRECT	SWAP EN
211	DIRECT	-, 3 STATE EN -, RQEN BX -, LD MAR -, C IN
213	CONDITION GROUP 2 BANK 1	-, LD DATA REG -, MEM READ


FC11
FETCH
FC11
WMOVE
FC14
ACO → BUS
RAR 4
NO
ALU LOW = 1
YES (FINISH)
AC1 → MAR
ACO ÷ 1 → ACO
RAR 6
NOP
RAR 7
MEM → Q
RAR 5
BREAK 5
AC2 → MAR
AC1+1 → AC1
RAR 10
CONDI-
TION GROUP 2
BANK 1
FC11
Q → MEM
RAR 11
YES
DCHR
NO
AC2 + 1 → AC2
BANK SHIFT
RAR 12
AC2+1 → AC2
RAR 13
YES
INTR
NO
BANK SHIFT
PC → MAR
RAR 14
NOP
RAR 15
CONDI-
TION GROUP 4
BANK 0
40₈ → MAR
RAR 171
MEM → AC2
RAR 172
33₈ → BUS
BUS+AC2 → AC15
RAR 173
AC15 → MAR
RAR 174
MEM → AC10
RAR 175
AC10+1 → MEM
RAR 176
AC10 → MAR
RAR 177
MEM → AC1
RAR 200
377₈ → ACO
RAR 201
AC1 ∧ ACO → ACO
RAR 202
AC1 ÷ ACO → AC1
RAR 203
AC1+AC1 → AC1
RAR 204
AC1+AC1 → AC1
RAR 205
AC1+AC1 → AC1
RAR 206
AC1+AC1 → AC1
RAR 207
PC+AC1+1 →
PC, MAR
RAR 211
MEM ÷ 1 → PC
RAR 213
BREAK 5
FC11
CONDI-
TION GROUP 2
BANK 1
FC11
FC14

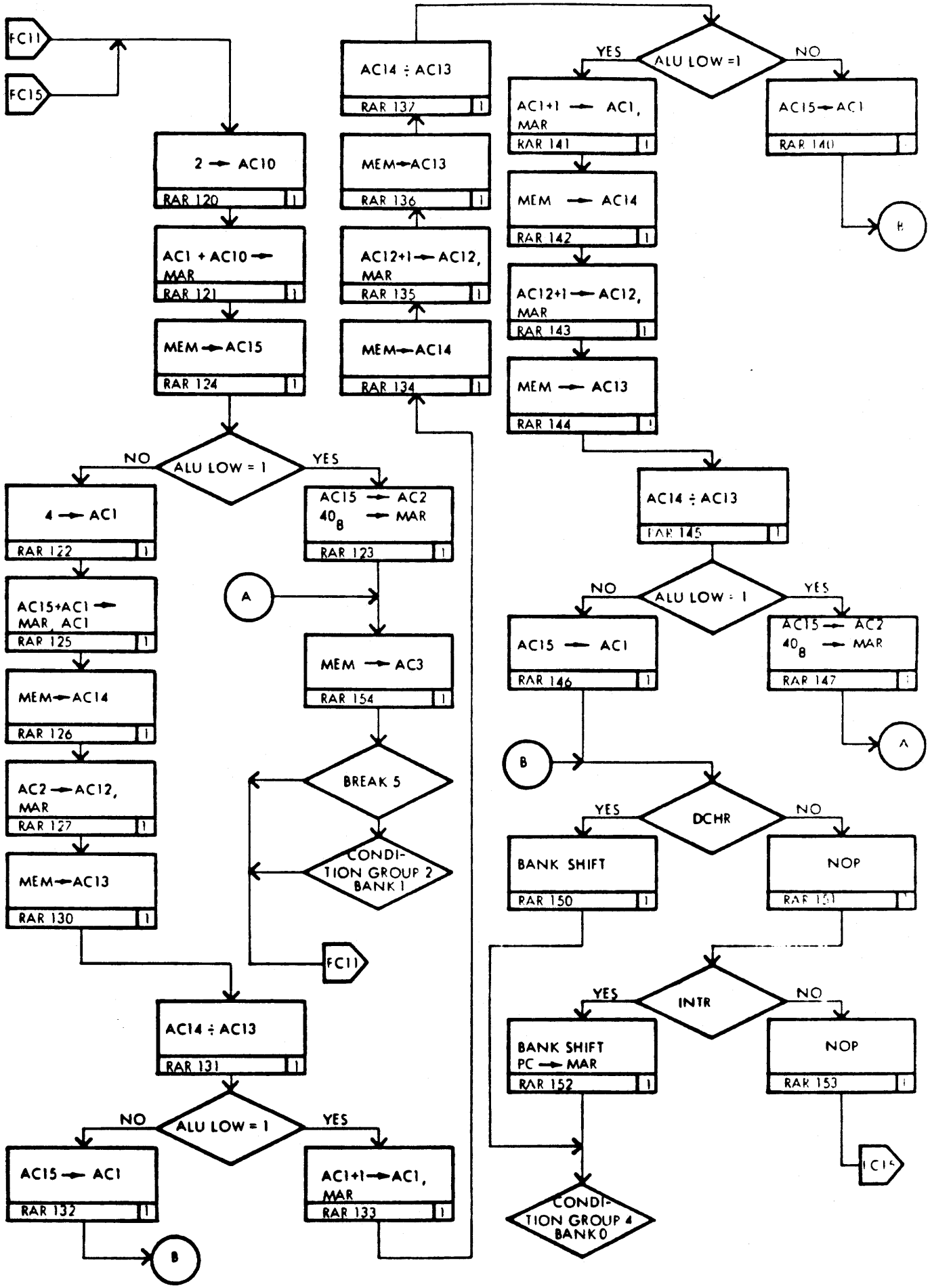
ROM ADDRESS (OCTAL) 216	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
120	DIRECT	-, ENABLE CONSTANT
121	DIRECT	-, LD DATA REG -, 3 STATE EN -, LD MAR
123, 147	DIRECT	-, RQEN BX -, LD MAR -, ENABLE CONSTANT
124	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, MEM READ
125, 127	DIRECT	-, 3 STATE EN -, LD MAR
126, 134, 142	DIRECT	-, LD DATA REG -, MEM READ
130, 136, 144	DIRECT	-, LD DATA REG -, RQEN BX -, MEM READ
131, 137, 145	DIRECT	-, 3 STATE EN COND SEL 0 COND SEL 1
132, 140, 146	DIRECT	-, 3 STATE EN COND SEL 1
133, 135, 141, 143	DIRECT	-, 3 STATE EN -, LD MAR -, C IN
150	CONDITION GROUP 4 BANK 0	
151	DIRECT	COND SEL 2
152	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR
153	DIRECT	
154	CONDITION GROUP 2 BANK 1	-, LD DATA REG -, MEM READ
122	DIRECT	-, ENABLE CONSTANT -, RQEN BX

A/S REGNEC ENTRALEN

Designed by

Drawn by

Dwg. Office Check


FC11
FC15
2 → AC10
RAR 120
AC1 + AC10 → MAR
RAR 121
MEM → AC15
RAR 124
NO
ALU LOW = 1
YES
4 → AC1
RAR 122
AC15+AC1 → MAR, AC1
RAR 125
MEM → AC14
RAR 126
AC2 → AC12, MAR
RAR 127
MEM → AC13
RAR 130
AC15 → AC2
40B → MAR
RAR 123
A
MEM → AC3
RAR 154
BREAK 5
CONDI-TION GROUP 2 BANK 1
FC11
AC14 ÷ AC13
RAR 131
NO
ALU LOW = 1
YES
AC15 → AC1
RAR 132
B
AC1+1 → AC1, MAR
RAR 133
AC14 ÷ AC13
RAR 137
MEM → AC13
RAR 136
AC12+1 → AC12, MAR
RAR 135
MEM → AC14
RAR 134
YES
ALU LOW = 1
NO
AC1+1 → AC1, MAR
RAR 141
AC15 → AC1
RAR 140
B
MEM → AC14
RAR 142
AC12+1 → AC12, MAR
RAR 143
MEM → AC13
RAR 144
AC14 ÷ AC13
RAR 145
NO
ALU LOW = 1
YES
AC15 → AC1
RAR 146
AC15 → AC2
40B → MAR
RAR 147
A
B
YES
DCHR
NO
BANK SHIFT
RAR 150
NOP
RAR 151
YES
INTR
NO
BANK SHIFT
PC → MAR
RAR 152
NOP
RAR 153
FC15
CONDI-TION GROUP 4 BANK 0

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
100	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, 3 STATE EN -, RQEN BX -, LD MAR -, ALU = 0 CLK
101, 117	DIRECT	-, LD DATA REG -, 3 STATE EN -, C IN
102, 111, 115, 345	DIRECT	
103	CONDITION GROUP 2 BANK 1	
104, 110	DIRECT	-, LD DATA REG -, 3 STATE EN
112	CONDITION GROUP 4 BANK 0	
113	DIRECT	COND SEL 2
114	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR
116	DIRECT	-, 3 STATE EN -, C IN -, ALU = 0 CLK COND SEL 1
212	DIRECT	COND SEL 0 COND SEL 1 -, LD MAR
344	DIRECT	-, C IN
346	CONDITION GROUP 4 BANK 0	
107	DIRECT	-, RQEN BX

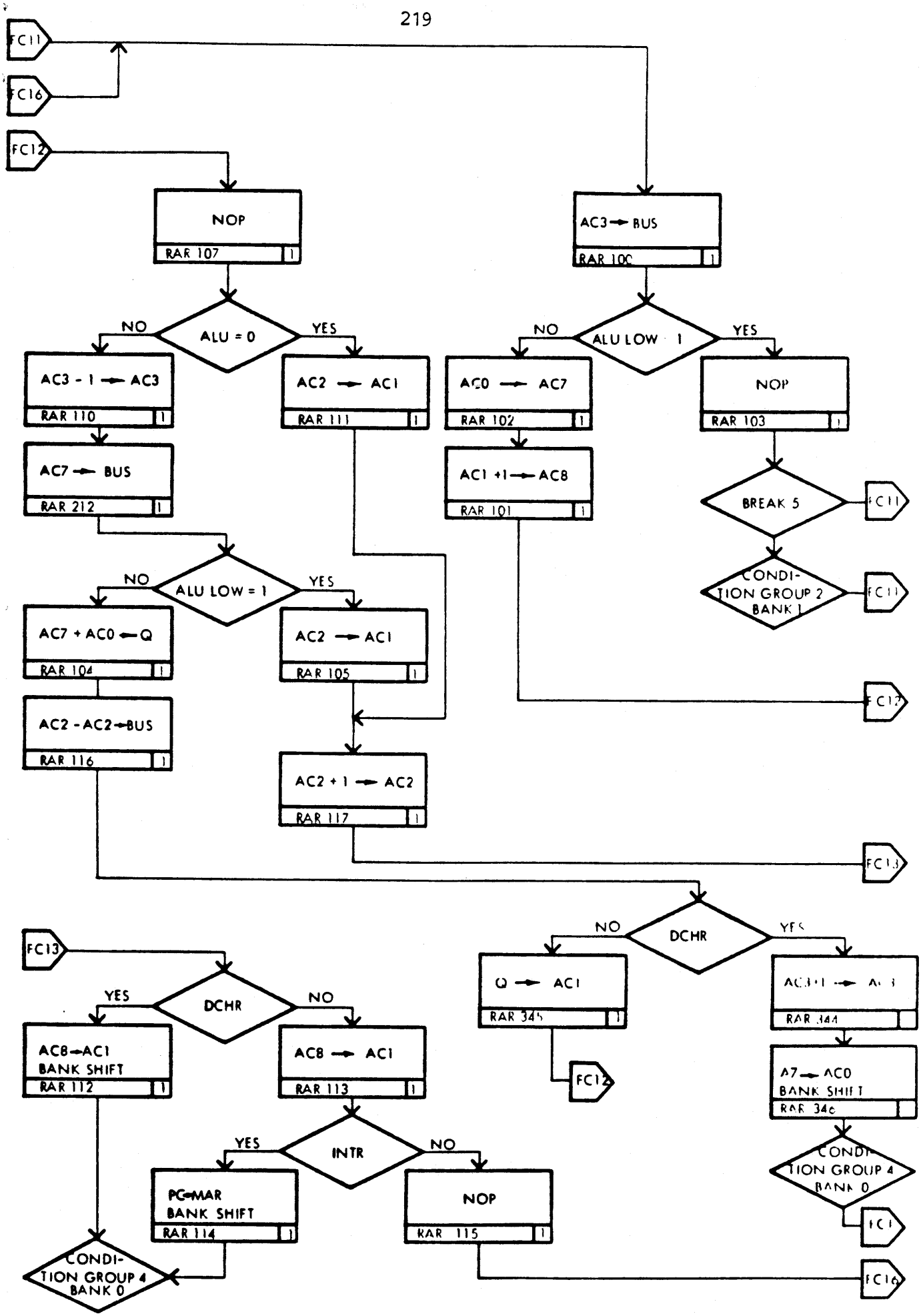
A/S REGNECENTRALEN

Designed by

Drawn by

Dwg. Office Check

Unit CPU721	BYTE MOVE	FC 16
Dwg. No. R21392	Signal List	16 of 21


FC11
FC16
FC12
NOP
RAR 107
AC3 → BUS
RAR 100
NO
ALU = 0
YES
NO
ALU LOW 1
YES
AC3 - 1 → AC3
RAR 110
AC2 → AC1
RAR 111
AC0 → AC7
RAR 102
NOP
RAR 103
AC7 → BUS
RAR 212
AC1 +1 → AC8
RAR 101
BREAK 5
FC11
CONDI-
TION GROUP 2
BANK 1
FC11
NO
ALU LOW = 1
YES
AC7 + AC0 → Q
RAR 104
AC2 → AC1
RAR 105
AC2 - AC2 → BUS
RAR 116
AC2 + 1 → AC2
RAR 117
FC12
FC13
NO
DCHR
YES
FC13
YES
DCHR
NO
Q → AC1
RAR 345
FC12
AC8 → AC1
BANK SHIFT
RAR 112
AC8 → AC1
RAR 113
A7 → AC0
BANK SHIFT
RAR 346
YES
INTR
NO
PC → MAR
BANK SHIFT
RAR 114
NOP
RAR 115
CONDI-
TION GROUP 4
BANK 0
FC1
CONDI-
TION GROUP 4
BANK 0
FC16

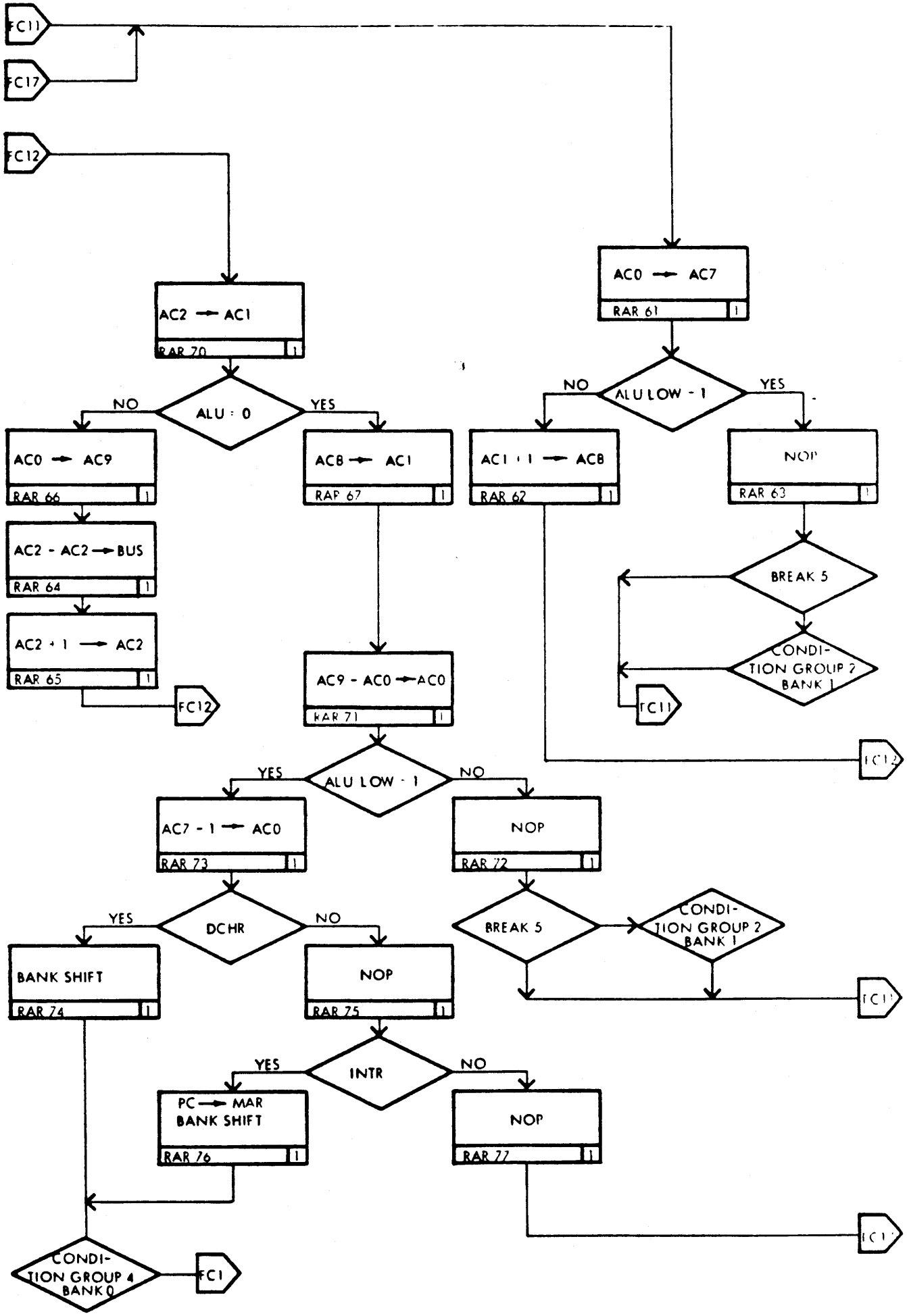
ROM ADDRESS (OCTAL) 220	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
61	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, 3 STATE EN -, RQEN BX -, LD MAR -, ALU = 0 CLK
62	DIRECT	-, C IN
63	CONDITION GROUP 2 BANK 1	
64	DIRECT	-, 3 STATE EN -, C IN -, ALU = 0 CLK
65	DIRECT	-, LD DATA REG -, 3 STATE EN -, C IN
66, 67, 77	DIRECT	
70	DIRECT	COND SEL 1 COND SEL 2 -, LD MAR -, RQEN BX
71	DIRECT	COND SEL 0 COND SEL 1 -, LD MAR -, C IN
72	CONDITION GROUP 2 BANK 1	-, LD DATA REG -, 3 STATE EN
73	DIRECT	COND SEL 1
74	CONDITION GROUP 4 BANK 0	
75	DIRECT	COND SEL 2
76	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR

A/S REGNECENTRALEN

Designed by

Drawn by

Dwg. Office Check



CPU 721
R13539

MICROPROGRAM FLOW CHART
BYTE COMPARE (COMP)
Flow Diagram

FC 17

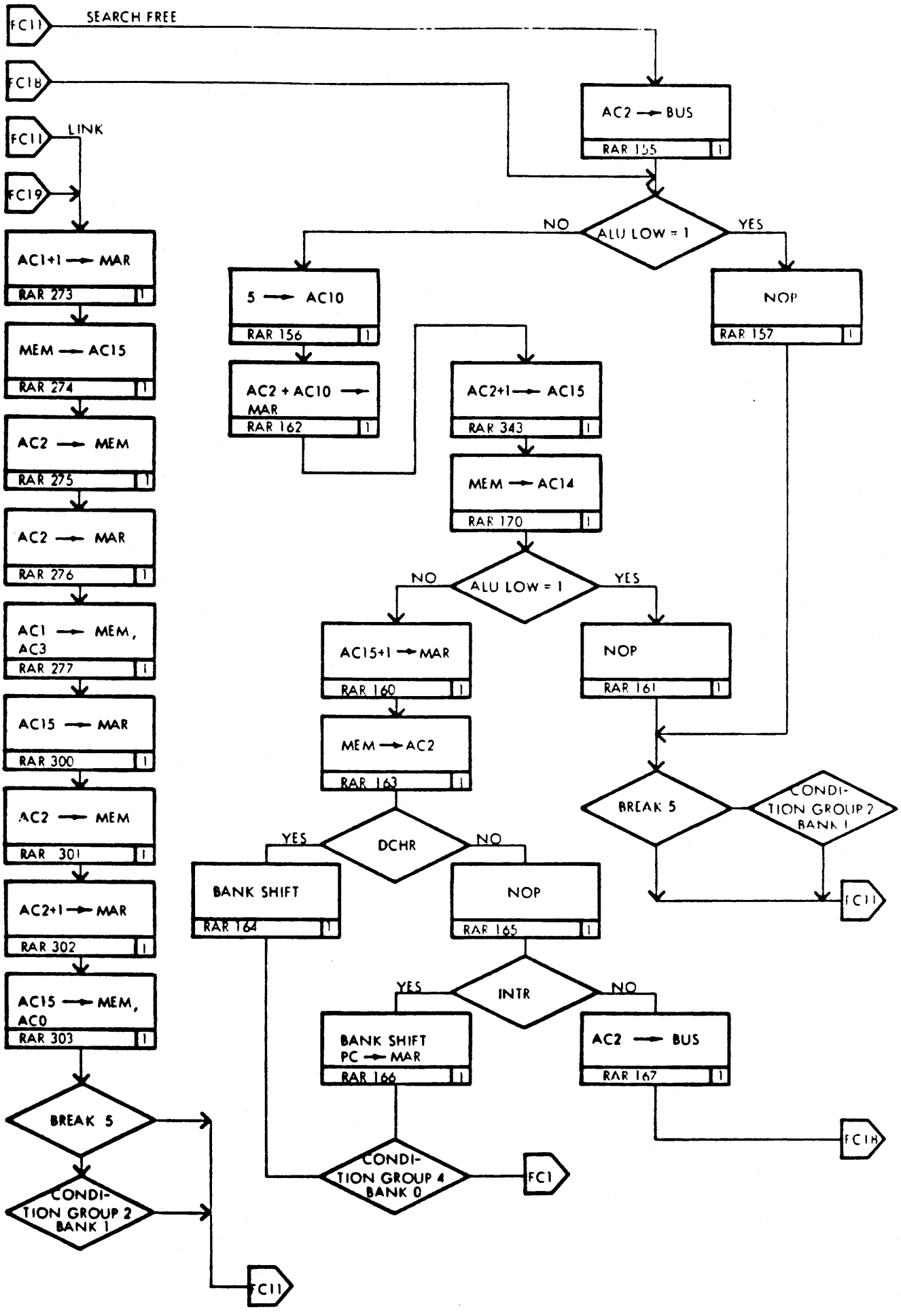
ROM ADDRESS (OCTAL) 222	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
155, 167	DIRECT	-, COND SEL 0 -, COND SEL 1 -, 3 STATE EN -, LD MAR
156	DIRECT	-, ENABLE CONSTANT
157	CONDITION GROUP 2 BANK 1	
160, 302	DIRECT	-, 3 STATE EN -, LD MAR -, C IN
161, 343	DIRECT	
162, 273	DIRECT	-, 3 STATE EN -, LD MAR
163	DIRECT	-, LD DATA REG COND SEL 1
164	CONDITION GROUP 4 BANK 0	
165	DIRECT	COND SEL 2
166	CONDITION GROUP 4 BANK 0	-, 3 STATE EN -, LD MAR
170	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1 -, RQEN BX -, MEM READ
274	DIRECT	-, LD DATA REG -, MEM READ -, TEST MEM
275, 301	DIRECT	-, 3 STATE EN -, MEM WRITE -, TEST MEM
276, 300	DIRECT	-, 3 STATE EN -, LD MAR
277	DIRECT	-, 3 STATE EN -, MEM WRITE -, TEST MEM -, RQEN BX
303	CONDITION GROUP 2 BANK 1	-, 3 STATE EN -, MEM WRITE -, TEST MEM

A/S REGNECENTRALEN

Designed by

Drawn by

Dwg Office Check



CPU721
R13540

MICROPROGRAM FLOW CHART
MUSIL LINK (LINK), SEARCH FREE
Flow Diagram.

FC 1H

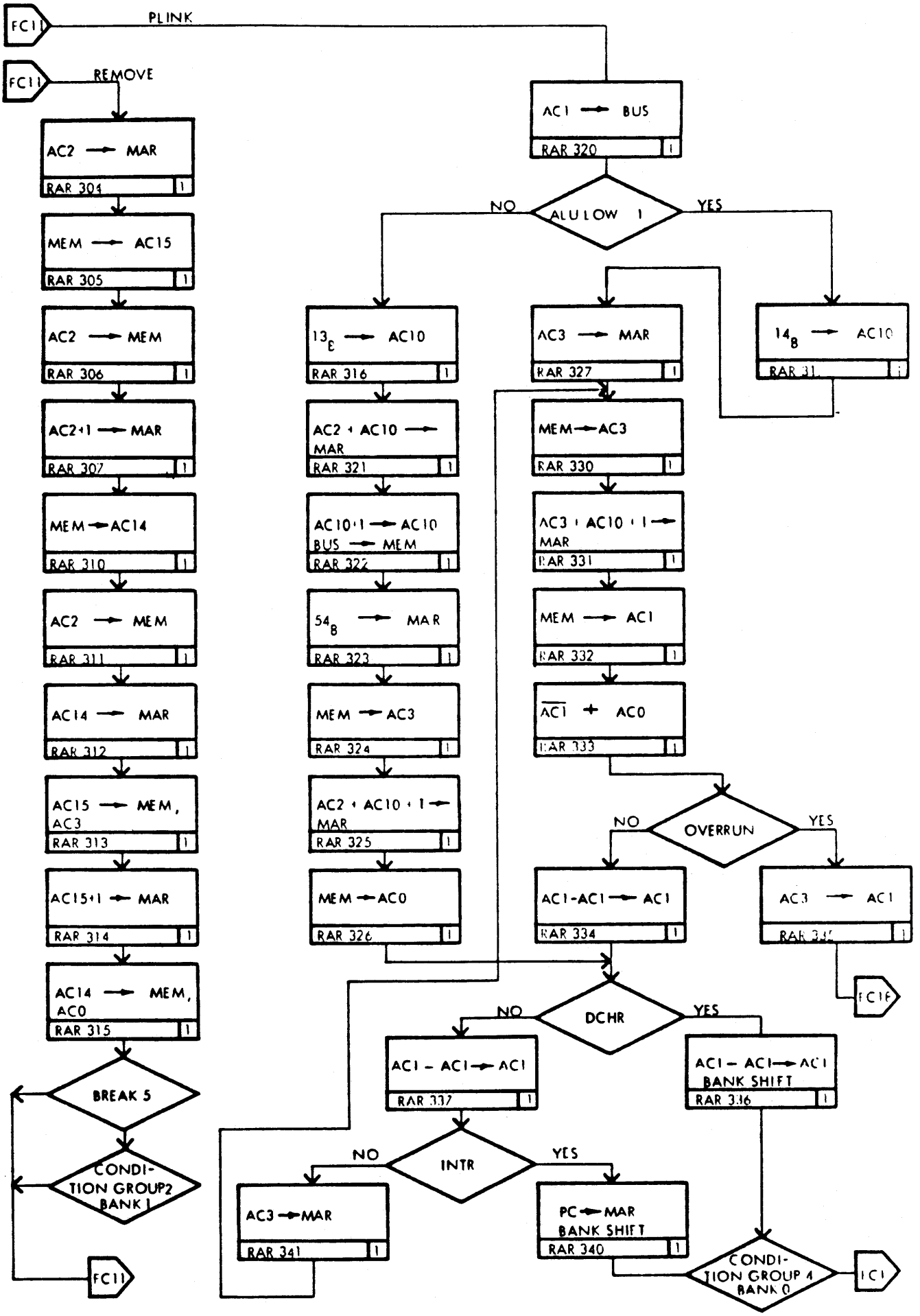
ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
304, 312, 321, 327, 341	DIRECT	-; 3 STATE EN -; LD MAR
305, 310	DIRECT	-, LD DATA REG -, MEM READ -, TEST MEM
324, 330		-, LD DATA REG -, MEM READ
306, 313	DIRECT	-, LD DATA REG -, 3 STATE EN -, MEM WRITE -, TEST MEM
307, 325, 331, 314	DIRECT	-; 3 STATE EN -; LD MAR -; C IN
311	DIRECT	-; LD DATA REG -; 3 STATE EN -; RQEN BX -; MEM WRITE -; TEST MEM
315	CONDITION GROUP 2 BANK 1	-; LD DATA REG -; 3 STATE EN -; MEM WRITE -; TEST MEM
316, 317	DIRECT	-; ENABLE CONSTANT
320	DIRECT	COND SEL 0 COND SEL 1 -; LD MAR
322	DIRECT	-; MEM WRITE -; TEST MEM -; C IN
323	DIRECT	-; RQEN BX -; LD MAR -; ENABLL CONSTANT
332	DIRECT	-; LD DATA REG -; RQEN BX -; MEM READ
333	DIRECT	-; LD DATA REG COND SEL 0 COND SEL 1 -; 3 STATE EN -; LD MAR
334	DIRECT	-; C IN COND SEL 1
326	DIRECT	-; LD DATA REG -; MEM READ COND SEL 1
335	DIRECT	
336	CONDITION GROUP 4 BANK 0	-; C IN
337	DIRECT	COND SEL 2 -; C IN
340	CONDITION GROUP 4 BANK 0	-; 3 STATE EN -; LD MAR

AS REQUESTED

Designed by

Drawn by

Dwg Office Check



CPU 721
R13541

MICROPROGRAM FLOW CHART
MUSIL REMOVE (REMOVE) MUSIL PRIORITY LINK (PLINK)
Flow Diagram

FC 19

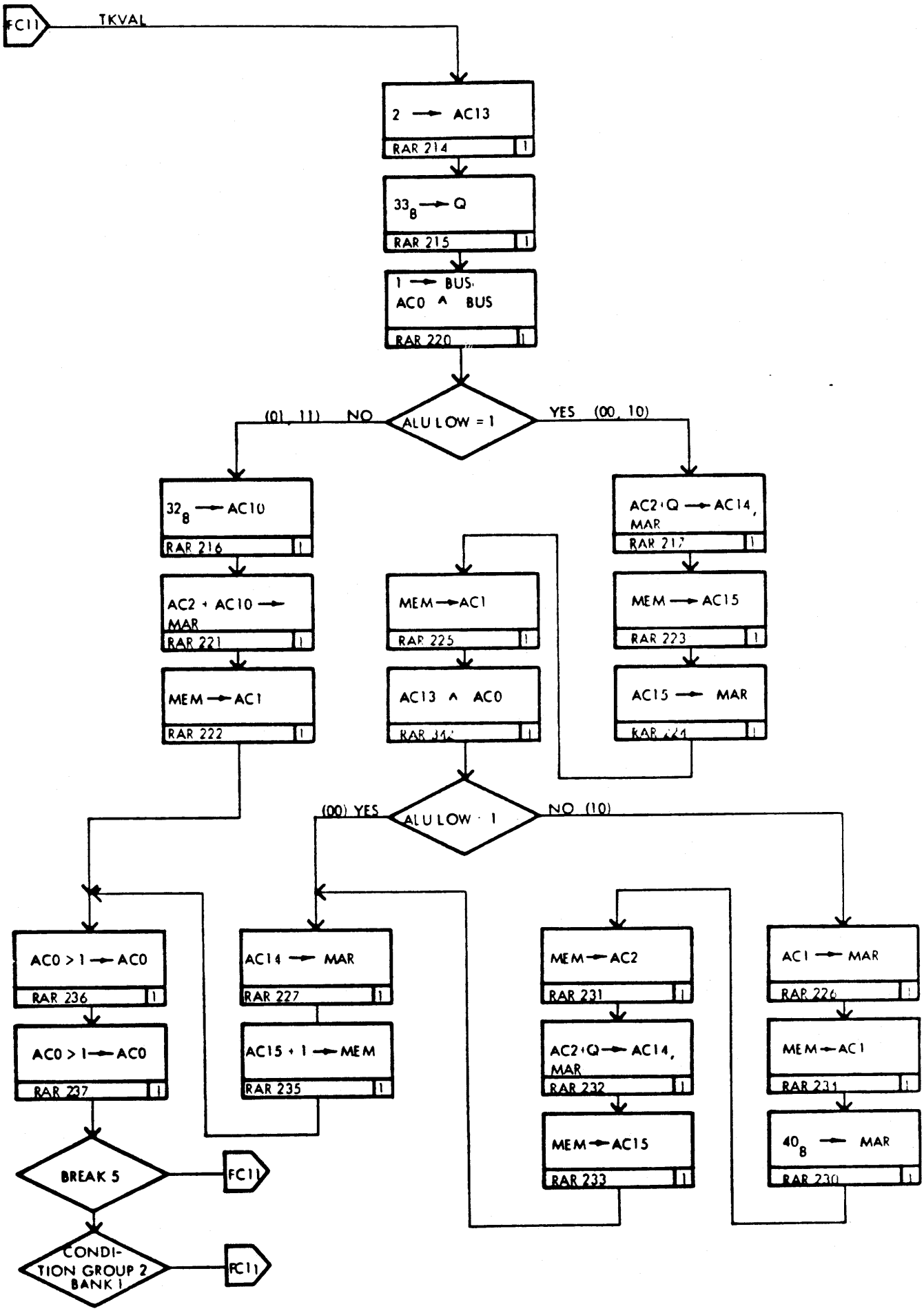
ROM ADDRESS (OCTAL) 226	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
214, 215, 216	DIRECT	-, ENABLE CONSTANT
217, 224, 226, 232	DIRECT	-, 3 STATE EN -, LD MAR
220	DIRECT	COND SEL 0 -, LD MAR -, ENABLE CONSTANT
221, 227	DIRECT	-, 3 STATE EN -, RQEN BX -, LD MAR
222, 223, 225, 231, 233, 234	DIRECT	-, LD DATA REG -, MEM READ
230	DIRECT	-, LD MAR -, EN CONSTANT
235	DIRECT	-, 3 STATE EN -, MEM WRITE -, TEST MEM -, C IN
236	DIRECT	-, RESET BIT 0
237	CONDITION GROUP 2 BANK 1	-, RESET BIT 0
342	DIRECT	-, LD DATA REG COND SEL 0 COND SEL 1

A/S REGNECENTRALEN

Designed by

Drawn by

Dwg. Office Check



CPU 721
R13542

MICROPROGRAM FLOW CHART
TAKE VALUE (TKVAL)
Flow Diagram

FC20

ROM ADDRESS (OCTAL)	ADDRESS CALCULATION TYPE	DO - FUNCTIONS
240, 252, 263, 266	DIRECT	-, ENABLE CONSTANT
241, 264, 267	DIRECT	-, 3 STATE EN -, LD MAR
242	DIRECT	-, MEM READ -, TEST MEM
243	DIRECT	-, LD DATA REG -, 3 STATE EN -, MEM WRITE -, TEST MEM -, C IN
244	DIRECT	-, LD DATA REG -, 3 STATE EN -, RQEN BX -, LD MAR
246, 265, 272	DIRECT	-, LD DATA REG -, MEM READ
247	DIRECT	COND SEL 0 COND SEL 1 -, LD MAR -, ENABLE CONSTANT
250, 251	DIRECT	-, RESET BIT 0
253	CONDITION GROUP 2 BANK 1	-, RESET BIT 0
254	DIRECT	COND SEL 0 COND SEL 1 -, LD MAR
255	DIRECT	-, C IN
256, 257, 260, 261	DIRECT	SWAP EN
262	DIRECT	
270	DIRECT	-, LD DATA REG -, TEST MEM -, MEM READ
271	DIRECT	-, RQEN BX -, LD MAR -, ENABLE CONSTANT

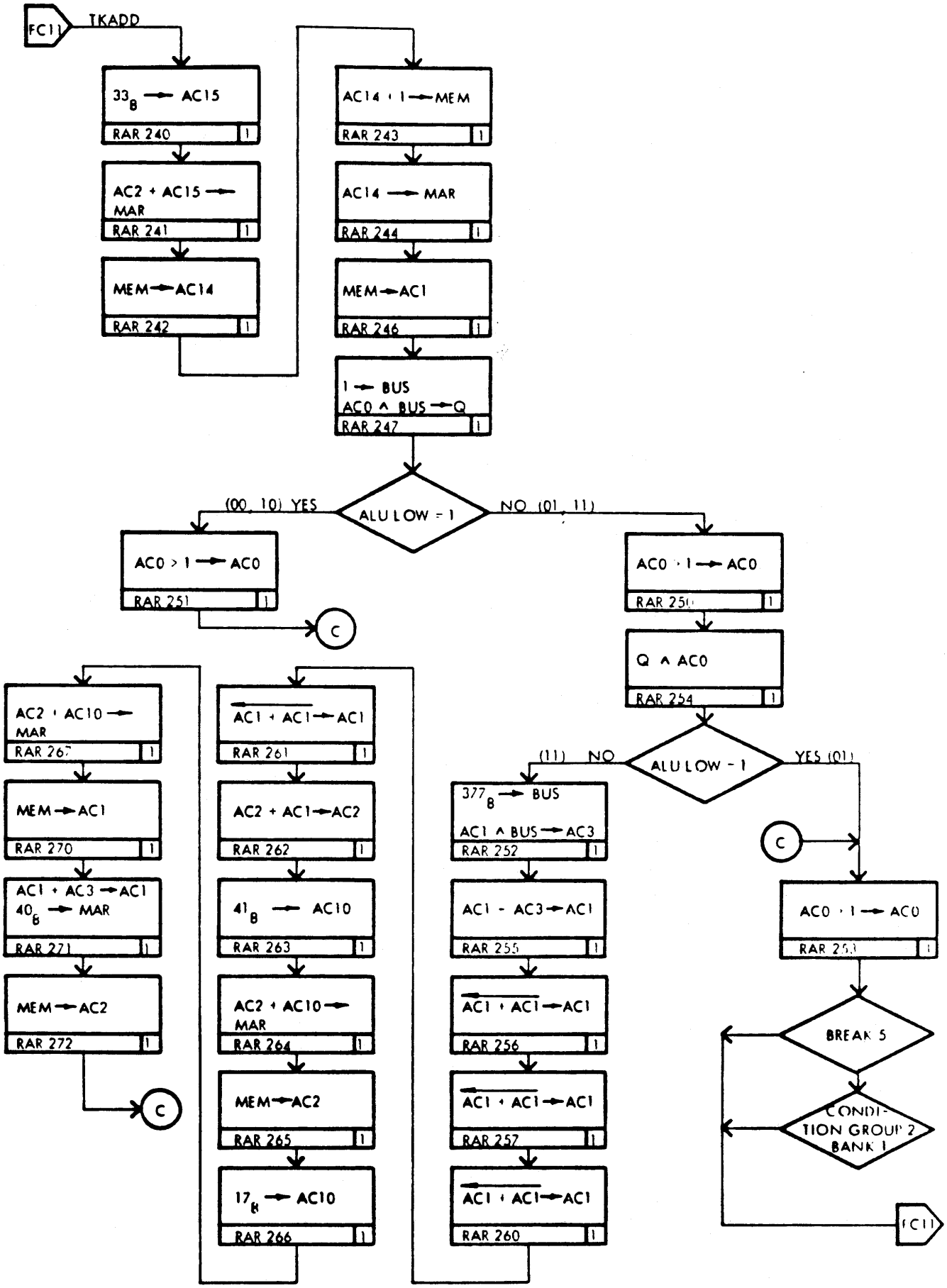
A/S REGNECENTRALEN

Designed by

Drawn by

Dwg. Office Check

Unit CPU721	TAKE ADDRESS	FC 21
Dwg. No. R21399	Signal List	21 of 21


FC11
TKADD
33B → AC15
RAR 240
AC2 + AC15 → MAR
RAR 241
MEM → AC14
RAR 242
AC14 + 1 → MEM
RAR 243
AC14 → MAR
RAR 244
MEM → AC1
RAR 246
1 → BUS
AC0 ∧ BUS → Q
RAR 247
ALU LOW = 1
(00, 10) YES
NO (01, 11)
AC0 + 1 → AC0
RAR 251
C
AC0 + 1 → AC0
RAR 250
Q ∧ AC0
RAR 254
ALU LOW = 1
(11) NO
YES (01)
377B → BUS
AC1 ∧ BUS → AC3
RAR 252
AC1 - AC3 → AC1
RAR 255
AC1 + AC1 → AC1
RAR 256
AC1 + AC1 → AC1
RAR 257
AC1 + AC1 → AC1
RAR 260
C
AC0 + 1 → AC0
RAR 253
BREAK 5
CONDITION GROUP 2 BANK 1
FC11
AC2 + AC10 → MAR
RAR 267
MEM → AC1
RAR 270
AC1 + AC3 → AC1
40B → MAR
RAR 271
MEM → AC2
RAR 272
C
AC1 + AC1 → AC1
RAR 261
AC2 + AC1 → AC2
RAR 262
41B → AC10
RAR 263
AC2 + AC10 → MAR
RAR 264
MEM → AC2
RAR 265
17B → AC10
RAR 266

RETURN LETTER

Title: CPU721, Technical Manual RCSL No.: 30-M328

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

4241288

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

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

Affix
postage
here

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