The Construction and Use of a Microprocessor

Robert Glaser

Abstract

The following is a report on a project worked on while employed at the Johns Hopkins University under Dr. C. R. Westgate during the summer of 1974. The equipment discussed was designed and built during this period.

Recently the use of microprocessors as integral parts of equipment has become more and more frequent due to technological development. To learn more about these computers on a chip a small microprocessor was constructed. The Intel MCS-4 microprocessor was chosen for this project. Programmable read only memory was used for the processor's program memory and a programmer was built to permit the programming of these chips.

A paper tape reader was constructed and used with the microprocessor system. Several programs were written for the processing system which allowed the three units to be used together.

The Construction and Use of a Microprocessor

Robert Glaser

I. Introduction

Digital computer systems have applications which include both calculation and control. For control purposes, the minicomputer offers many advantages. Minicomputer systems are more flexible, can be easily personalized for a particular customer's requirements, and can be more easily changed or updated than fixed logic design systems. The programming of a minicomputer is a much easier and more straightforward procedure than designing a controller with random logic. However, for many purposes, a control system must be extremely complex in order to warrant the implementation of a conventional minicomputer, simply because of the cost involved. With the advent of LSI technology it has become possible to fabricate an entire central processor unit on a single chip. Utilizing this advance, it is possible to group several integrated circuits together to form a microprocessor. These microprocessors are also suitable for calculation as well as control if the speed requirements are not too demanding. For this reason microprocessors are also called microcomputers. Very often a microprocessor would be sufficient to perform the control process. In such cases it would be economical to use a microprocessor where it would not be to use a minicomputer. As a first step toward assembling a microprocessor facility to investigate the problems and advantages peculiar to microprocessors, it was desired to build a microprocessor.

In this project an Intel MCS-4 microprocessor, a four-bit processor, was constructed along with a programmer for its program memory. It was programmed to perform a task for demonstration purposes. Following is a description of the construction and utilization of this system.

Section II deals with the operation of the MCS-4 microcomputer, with the manner in which reprogrammable memory is used in the system, and with the input/output.

Section III describes the fabrication of the processor. Construction details and additional circuitry are included.

Operation of the ROM programmer and its assembly are covered in section IV. Actual use of the system as a programming system is discussed in section V. The program which allows the processor to control the ROM programmer and a paper tape reader is described. With this system the processor can be programmed from paper tape prepared on a BASIC system.

Section VI is a summary of the project. A high speed paper tape reader that can be used with the MCS-4 system is discussed in Appendix A. Appendix B contains the BASIC program used in the programming system, and Appendix C contains the assembly language program used in that system. Appendix D contains a program used with the processing system to generate Morse code from ASCII encoded paper tapes.

II. Description of MCS-4 Operation

The basic MCS-4 system consists of three chips: the central processor unit (CPU) 4004, random access memory (RAM) 4002, and read only memory (ROM) 4001. The CPU has an instruction set consisting of 45 arithmatic and logical instructions. Internal to the 4004 are sixteen four-bit registers. The organization of the MCS-4 is such that it has separate program memory and read/write memory. The program memory is ROM and cannot be modified by the processor. For temporary data storage the RAM is used. The 4001 ROM is a 256 x 8 memory that must be purchased with a particular program stored in it. The 4002 RAM is a 320 bit memory organized into four registers of twenty four-bit characters each. Additionally, both the ROM and the RAM are used for input/output operations. The RAM has a four-bit output port per chip. The 4001 ROM can be ordered with a total of four input/output bits. A minimum system configuration consists of one 4004 and one 4001. A maximum system may have up to sixteen 4001s and sixteen 4002s. Figure 1 shows the system inter-connection.

The system just described is well suited for a dedicated task where the processor will be running a single program. For flexibility a programmable program memory is needed. The Intel 1702A chip is a programmable, erasable 256 x 8 read only memory. This programmable read only memory (PROM) is erased by exposure to ultraviolet light and must be programmed with a specialized ROM programmer. With this type of memory, different programs can be tried and debugged. However, the 4001 contains decoding not present in the 1702A, so that it is necessary to provide extra logic when using the PROM in the MCS-4 system. Intel manufactures two chips which do all of this: the 4008 and the 4009. The 4008 has as input the four-bit data bus of the 4004, and it receives three four-bit bytes of address information. Eight bits go to the address inputs of the PROMS; the other four bits are decoded to provide a maximum of sixteen chip select lines. The eight data bits containing program information from the 1702A are fed into the 4009 which multiplexes these into two four-bit bytes into the data bus of the processor. In this manner the combination of the 4008, 1702A, and the 4009 emulate the operation of the 4001 ROM. The 4008 also provides output information that indicates whether an operation is an input, output, or program read operation. This information is decoded externally to the above chips and is used to enable different input/output latches. This emulates the input/output ports of the 4001 but gives greater flexibility because the number of ports is independent of the number of PROMs. Figure 2 shows this system interconnection.

III. Assembly

The processor constructed consists of a 4004, four 4002s, four 1702As, a 4008, a 4009, four four-bit input ports, four four-bit output ports, plus the four four-bit output ports of the RAMs. Twenty one TTL chips were employed as support electronics. The processor circuit is shown in Fig. 3. The processor needs a 750 kHz two-phase clock. This was constructed from a square wave oscillator that triggers two monostable multivibrators. It thus provides the ability to vary the duty cycle of each phase of the clock independently. The processor also requires a reset pulse generator. This is necessary to reset all the internal registers in the processor and to begin program execution at program memory location 0. The reset generator is an RS flip-flop enabled by a pushbutton which triggers a one shot multivibrator driving associated transistor stages to provide the proper voltage levels. The 4004 has an input pin labeled Test which is the closest approximation the processor has to an interrupt. In the jump conditional (JCN) instruction a bit may be set which instructs the program counter to jump if there is a signal at the Test input. A pulse generator was used to drive the Test input because it was thought that this would be the most convenient manner in which to use the signal. However, after some use, it was found to be more convenient to have an RS flip-flop drive the Test input; an external signal could set the flip-flop, and when the processor acknowledged the request it could reset the flip-flop. The clock, reset, and Test generators are shown schematically in Fig. 4.

The MCS-4 requires power supply voltages of +5 and -9 volts to be compatible with TTL. The regulated power supply is shown in Fig. 5.

The power supply was built on vectorboard and mounted below the chassis. The processor proper was constructed on a printed circuit board and mounted above the chassis. Sixteen pin DIP sockets were used for all input/output connections. These were found to be space saving and easy to use. The printed circuit board is 9 x 9 inches.

The PC board stock material was 1/16 inch, G10 glass epoxy, double-sided two ounce copper clad board. A layout was made double sized on clear acetate film using opaque tape for traces, prepared integrated circuit pads, and circular pads for the other components. Separate sheets of acetate were used for the top and bottom sides of the board, taking care that all pads were coincident. The layouts were then photographed with Kodalith Ortho film using diffuse backlighting. This gave a 4 x 5 inch negative of each layout. A contact print was then made giving a positive of each, followed by an enlargement giving an exact size negative of the layouts. Care was taken to keep all processing identical for the two layouts to insure that both layouts would match. The PC board was then prepared by vigorously cleaning with steel wool. Three registration holes were drilled around the perimeter. The board was then dipped into KPR photoresist in the dark and baked at about 175° F for ten minutes. This dipping and baking process was repeated. The two negatives were then taped to the board, and lined up with the predrilled registration holes, all in semi-darkness. Exposure of each side was then made with a 375 watt photo flood lamp at a distance of approximately 17 inches for two minutes. The board then was agitated in KPR developer for several minutes, and rinsed in hot water, followed by another bake. Etching was done with a ferric chloride solution until both sides were complete. Drilling all holes on a drill press and cleaning again with steel wool completed the board.

IV. ROM Programmer

In order to use the microprocessor it is necessary to be able to program the read only memories to be plugged into the processor board. The 1702As are relatively simple to employ in the read mode, but require complex timing voltages to be programmed. The circuitry in Intel's MP7-03 PROM Programmer was copied. Figure 6 shows this schematic. The programmer was constructed on a 9 x 9 inch printed circuit board, and was mounted above the chassis. The power supply (Figure 7) was built on vector-board and mounted below the chassis with the exception of the transformers and power transistors. Unlike the processor, a front panel was made out of a 7 x 19 inch rack panel. Eight TTL compatible LEDs were placed on the panel to allow reading the memory contents. Sixteen toggle switches were used, half for data input and half for address selection, plus two power switches and four control switches. The ROM programmer is a self complete unit, and of course does not have to be used solely with the MCS-4. All data, address, and program leads were led to a connector on the rear of the chassis to allow control of the programmer by other devices. It was originally intended to program a PDP-11/20 to perform this task, but it was found that the MCS-4 could easily do this task.

The programmer can selectively put ones in the desired locations, but cannot make a one a zero. Erasure is accomplished with an ultraviolet source. It was found that the 1702A could be erased and reprogrammed several times, but that the more times it was erased, the longer time it subsequently required for erasure. It is expected that the useful time of the ROM is about six to twelve erasures.

V. Programming System

The complete programming system consists of the MCS-4 microprocessor, ROM programmer, paper tape reader, ultraviolet source, and use of a BASIC system with a suitable program. Henry Berkley wrote the BASIC program which produces a binary paper tape. While running the program, a beginning address location is entered, and all MCS-4 machine code is entered with ones and zeroes. When all data are entered, a command is given and a listing is produced followed by the binary tape. The first character on the tape is always a rubout (all ones). There is no MCS-4 command corresponding to this character and this indicates the start of programming information to the microprocessor. Similarly, at the end of the program material another rubout is added to indicate the end of the tape.

The ROM is erased using the ultraviolet source and is then inserted into the ROM programmer. Connections are made which run from all address and program terminals of the programmer to output ports of the microprocessor. Eight parallel output bits from the paper tape reader go to input ports of the microprocessor and to the data inputs of the programmer. An output bit from the micro-processor goes to the tape increment input of the reader. An MCS-4 program (already prepared, see Appendix C), PROG.RAM is put into the microprocessor. All address and data toggle switches of the programmer are placed into the center position (unconnected) and power to all the units is then applied. Depressing the reset button on the processor starts PROG.RAM, and the tape is read. The proper address is given the programmer by the processor, and the processor gives the program command at the proper instant. The high voltage on the programmer should not be turned on until the system is running and reading blank leader tape to prevent false programming. In this manner the ROM is programmed.

This system is immensely better than programming ROMs by hand on the programmer. Apart from the ease and time savings, it makes no mistakes versus programming by hand, where it is easy to make a mistake. It takes only a single one where there should be a zero to ruin a programming attempt. With this system it is still difficult to assemble the proper machine code on paper. Berkley has recently written a cross assembler for the MCS-4 in BASIC, and this should make programming the MCS-4 a simple and painless task.

VI. Summary

The microprocessor system constructed is quite versatile and reliable. As an example, before the MCS-4 was built it was planned to write a demonstration program for a PDP-11/20 to send perfect Morse code. This was done with the microprocessor instead of with the minicomputer. Oscilloscope display programs formerly done with the PDP are now planned for control by the MCS-4. This demonstrates that a microprocessor can often free minicomputers from tasks which do not demand the power and speed of conventional minicomputers. Control of laboratory experiments by the microprocessor is planned. The printed circuit board techniques used give a durable final product and facilitates duplication. The ROM programmer will also be used for a second generation microprocessor soon to be constructed. Reliability of the system has been good, the only failure of the processor was damage of one of the TTL output gates. These are quickly and easily replaced.

Appendix A -- Construction of a Fast Paper Tape Reader

As a separate project, a reader unit was put together for use with a PDP-11/20. However, the reader was found to be of great use with the MCS-4 system.

The reader itself is a Tally R2050 photoelectric tape reader. It requires +5 VDC at 2.3 amps and +48 VDC at 2.2 amps, and has 8 parallel TTL compatible output bits as well as an advance pulse input. The completed unit also was required to buffer the inputs and outputs of the PDP-11 to prevent burnout of any more of the general purpose interface components in the computer. The output of the reader is fed into the lower eight bits of the sixteen bit input word of the PDP, so these inputs are switched from the reader to a general purpose input port. The schematic of the reader unit is shown in Fig. 8.

The electronics were built into a minibox, and both the reader and the electronics package were attached to a 19 inch rack panel. The front panel contains connectors for general purpose input and output, while the rear of the minibox has connectors which interface with the PDP-11. For use with the microprocessor the input and output of the MCS-4 are connected at the rear interface. The reader can be used at speeds of up to 250 characters per second. This reader reduces the reading of the PDP-11 assembler software from thirteen minutes to one and one-half minutes. This great reduction permits more flexible use of the PDP-11 in the undergraduate minicomputer laboratory.

Appendix B -- BASIC Program

```
10 ! BINARY TAPE PROGRAM          H.BERKLEY          08-OCT-74
20 PRINT "BINARY  V-HB"
30 INPUT "FILE";F$
40 INPUT "NEW OR OLD";Y$
50 IF LEFT(Y$,1%)<>"N" THEN OPEN F$ FOR INPUT AS FILE 5%
       ELSE OPEN F$ FOR OUTPUT AS FILE 5%: MAT B%=ZER: B%(0%)=0%
60 DIM #5%,B%(255%)
70 INPUT "STARTING LOC (HEX)";N7$
75 H9$="0123456789ABCDEF"
77 N%=FNH%(N7$)
80 PRINT FNH$(N%);" ";
85 IF B%(N%) THEN PRINT FNB$(B%(N%));
90 INPUT B$
95 IF LEFT(B$,1%)="/" THEN 220
100 B$=LEFT(B$,8%)
110 IF B$="" THEN 200
120 C%=0%
130 IF LEFT(B$,1%)="1" THEN C%=C%+1%
140 B$=RIGHT(B$,2%)
150 IF NOT B$="" THEN C%=C%*2%: GOTO 130
160 B%(N%)=C%
200 N%=N%+1%: IF N%<>256% THEN 80
220 INPUT "FINISHED";Y$
230 IF LEFT(Y$,1%)="N" THEN GOTO 70
231 CLOSE 1%: OPEN F$ AS FILE 1%: GOSUB 2000: V$=SYS(CHR$(0%))
235 INPUT "PUNCH TO (HEX)";K7$: E%=FNH%(K7$)
240 INPUT "PUNCH DEVICE";D$
250 IF D$="" THEN D$="KB:"
270 INPUT "READY PUNCH AND HIT <CR>";Y$
275 OPEN D$ AS FILE 1%,MODE 1%
280 PRINT #1%:PRINT #1%,CHR$(0%); FOR I%=1% TO 100%
290 PRINT #1%,CHR$(255%);
295 PRINT #1%,CHR$(B%(I%)); FOR I%=0% TO E%
300 PRINT #1%,CHR$(255%);
305 PRINT #1%,CHR$(0%); FOR I%=1% TO 100%
310 CLOSE I% FOR I%=1% TO 12%
320 GOTO 30
1000 DEF FNH%(H8$) ! CHANGE HEX TO A NUMBER
1010 H8$=RIGHT("00"+H8$,LEN(H8$)+1%)
1020 FNH%=INSTR(1,H9$,LEFT(H8$,1%))*16%+INSTR(1%,H9$,RIGHT(H8$,2%))-17%
1030 FNEND
1200 DEF FNH$(Y8%) ! RETURNS A 2 DIGIT HEX STRING FOR NUMBER
1210 FNH$=MID(H9$,Y8%/16%+1%,1%)+MID(H9$,Y8%-(Y8%/16%)*16%+1%,1%)
1220 FNEND
2000 U$="### \        \"
2010 PRINT F$,TIME$(0%),DATE$(0%)
2020 PRINT FNH$(N%)+"  "+FNB$(B%(N%)) FOR N%=0% TO 255%
2030 RETURN
3000 DEF FNB$(L%)
3020 MAT D%=ZER
3030 FOR LI%=1% TO 8%
3040 D%(LI%)=1% IF (256%/2%↑LI%) AND L%
3050 NEXT LI%
3060 K$=""
3070 FOR K%=1% TO 8%
3080 IF D%(K%)=1% THEN K$=K$+"1" ELSE K$=K$+"0"
3090 NEXT K%
3100 FNB$=K$: FNEND
9999 END
```

Appendix C -- Program for Programming System

The MCS-4 program PROG.RAM is run by the processor to operate the programming system described in section V. Figure 9 shows the flow chart. When started, the processor reads the tape and checks to see if the character is a rubout. If it is not, the processor checks to see if there has been a rubout read previously, and if not, the tape is moved. In this manner the tape is moved until the first rubout is read, and then stores information that the rubout has been read. The tape is incremented and the processor sends a program pulse to the programmer (the output ports have been previously initialized to zero). The output ports are incremented and the program loops back. Until the second rubout is received, the programming continues. When the last rubout is read, the processor goes into an endless loop which continuously increments the address lines. (This can be used to check ROMs to determine whether they have been properly erased. By looking at the data lights on the programmer, one can see whether any bits remain set.)

PROG.RAM

To ROM Programmer: program pulse - RAM output port 0, bit 0
address lines - ROM output ports 0, 1

To reader: tape increment - RAM output port 1, bit 0
data lines - ROM input ports 0, 1

```
         Mnemonic              Address (hexadecimal)

              FIM, 0              12
              0, 0                13
              FIM, 1              14
              1, 0                15
              FIM, 2              16
              4, 0                17
              LDM, 1              18
              SRC, 0              19
              WMP                 1A
              SRC, 2              1B
              WMP                 1C
              CLB                 1D
              SRC, 0              1E
              WRR                 1F
              SRC, 1              20
              WRR                 21

RUBTEST:      SRC, 0              24
              RDR                 25
              CMA                 26
              JCN, 12             27
              NORUB               28
              SRC, 1              29
              RDR                 2A
              CMA                 2B
              JCN, 12             2C
              NORUB               2D

RUBOUT:       LD, 1               30
              JCN, 4              31
              NOSET2              32
              NOP                 33
              JUN, 0              34
              LOOP                35

NORUB:        LD, 1               38
              JCN, 12             39
              PROG                3A

NOSET1:       JMS, 0              3B
              TAPE INC            3C
              JUN, 0              3D
              RUBTEST             3E

NOSET2:       INC, 1              41
              JUN, 0              42
              NOSET1              43
```

```
PROG:          CLB                  46
               SRC, 0               47
               WMP                  48
               JMS, 0               49
               DELAY                4A
               LDM, 1               4B
               WMP                  4C
               JMS, 0               4D
               DELAY                4E

INCADD:        ISZ, 11              51
               W                    52
               INC, 10              53

W:             LD, 11               54
               WRR                  55
               LD, 10               56
               SRC, 1               57
               WRR                  58
               JUN, 0               59
               NOSET1               5A

TAPE INC:      CLB                  5D
               SRC, 2               5E
               WMP                  5F

M:             ISZ, 13              60
               M                    61
               IAC                  62
               WMP                  63

N:             ISZ, 12              66
               N                    67
               ISZ, 15              68
               N                    69
               ISZ, 14              6A
               N                    6B
               BBL, 0               6C

DELAY:         LDM, 13              6F
               XCH, 6               70

D:             ISZ, 9               71
               D                    72
               ISZ, 8               73
               D                    74
               ISZ, 7               75
               D                    76
               ISZ, 6               77
               D                    78
               BBL, 0               79

LOOP:          SRC, 0               7C
               ISZ, 11              7D
               A                    7E
               INC, 10              7F
```

```
A:              LD, 11                        80
                WRR                           81
                LD, 10                        82
                SRC, 1                        83
                WRR                           84

                LDM, 14                       87
                XCH, 7                        88

B:              ISZ, 9                        89
                B                             8A
                ISZ, 8                        8B
                B                             8C
                ISZ, 7                        8D
                B                             8E
                JUN, 0                        8F
                LOOP                          90
```

Appendix D -- Programming the MCS-4 to Generate Morse Code

A program, MORSEC.ODE, was written to allow the MCS-4 to send International Morse. The program nearly fills one PROM and permits the processor to control a paper tape reader and a code tone oscillator. The text to be transmitted is placed in ASCII on standard paper tape from a teletype. The tape is then fed to the reader and the processor decodes the information from ASCII into the proper one bit sequence necessary for code. A four-bit input port controls the sending speed -- with MORSEC.ODE useful speeds of 5, 7 1/2, 10, 13, 17, 22, 32, and 64 words per minute are possible. With minor modifications eight bits of input could be used giving more of a continuous speed control.

Figure 10 shows the flow chart for MORSEC.ODE. When reset the program begins at location 0, and the proper initializations are made. The input port is then read, and the speed is determined. The reader reads the tape and loops until a valid character is read. When this happens the CPU fetches indirect information stored at the memory address corresponding to the ASCII byte. The information fetched is constructed in such a way that the first three bits tell how many dits and dahs that particular character is, minus one. Lengths of five, six, and seven are reserved for special characters of length greater than five. The next five bits of information give the sequence of dits and dahs, zero corresponding to a dit, and one corresponding to a dah. When the CPU receives the coded information, it checks to see if the character read is one of the three special characters, period, comma, and space and deals with these individually. If a special character is not encountered, the processor goes into a loop, with the lower five bits of information put into the accumulator with carry. After sending the dit or dah, the accumulator is rotated left, and generates the next dit or dah. After completing the character, the processor waits the right amount of time, and loops back to read the next character. Subroutines are used to increment the tape, time, send dits and dahs, and send the special characters.

The system is easy to use; the keyed oscillator (Fig. 11) and the reader are connected to the processor, the tape is loaded, and the reset button is hit. During sending, the speed can be changed by placing a different input at the input port and resetting the processor. A more elaborate system would allow the processor to read input from a keyboard at a varying rate and send code appropriately, but this would require the processor to be interfaced with a teletype or some other keyboard.

MORSEC.ODE

Interfaces:

```
     ROM input ports -  0  high order bits from reader
                        1  low order bits from reader
                        2  speed set inputs:  input   speed

                                                3       5 wpm
                                                7       7 1/2
                                                9      10
                                               11      13
                                               12      17
                                               13      22
                                               14      32
                                               15      64

     RAM output ports  -  0  bit  0  oscillator
                          1  bit  0  reader increment
```

```
          Mnemonic               Address (hexadecimal)

           FIM, 1                  6
           0,0                     7
           FIM, 2                  8
           1,0                     9
           FIM, 3                  A
           2,0                     B
           FIM, 4                  C
           4,0                     D
           SRC, 3                  E
           RDR                     F
           XCH, 3                  10
           SRC, 4                  11
           LDM, 1                  12
           WMP                     13

READ:      SRC, 1                  16
           RDR                     17
           XCH,0                   18
           LD, 0                   19
           JCN, 4                  1A
           INC                     1B
           IAC                     1C
           JCN, 4                  1D
           INC                     1E
           SRC, 2                  1F
           RDR                     20
           XCH, 1                  21

DECODE :   FIN, 5                  24
           LD, 10                  25
           RAR                     26
           XCH, 10                 27
           TCC                     28
           XCH, 9                  29
           LDM, 9                  2A
           XCH, 7                  2B
           LD, 10                  2C
           ADD, 7                  2D
```

```
            JCN, 2                        2E
            PERIOD                        2F
            IAC                           30
            JCN, 2                        31
            COMMA                         32
            IAC                           33
            JCN, 2                        34
            SPACE                         35
            LD, 10                        36
            IAC                           37
            XCH, 10                       38

NEXT:       LD, 9                         3A
            CLC                           3B
            JCN, 4                        3C
            S                             3D
            STC                           3E

S:          LD, 11                        3F
            JCN, 2                        40
            DASH                          41

DOT:        RAL                           44
            XCH, 11                       45
            TCC                           46
            XCH, 9                        47
            JMS, O                        48
            DIT                           49
            JUN, O                        4A
            DEC                           4B

DASH:       RAL                           4C
            XCH, 11                       4D
            TCC                           4E
            XCH, 9                        4F
            JMS, O                        50
            DAH                           51

DEC:        XCH, 10                       52
            DAC                           53
            XCH, 10                       54
            LD, 10                        55
            JCN, 12                       56
            NEXT                          57
            JUN, O                        58
            WAIT                          59

INC:        SRC, 4                        5A
            CLB                           5B
            WMP                           5C
            FIM, 6                        5D
            O, O                          5E

B:          ISZ, 13                       5F
            B                             60
            IAC                           61
            WMP                           62
```

```
C:          ISZ, 13                 63
            C                       64
            ISZ, 12                 65
            C                       66
            JUN, O                  67
            READ                    68

DIT:        SRC, 1                  6A
            LDM, 1                  6B
            WMP                     6C
            JMS, O                  6D
            TIME                    6E
            CLB                     6F
            WMP                     70
            JMS, O                  71
            TIME                    72
            BBL, O                  73

DAH:        SRC, 1                  76
            LDM, 1                  77
            WMP                     78
            JMS, O                  79
            TIME                    7A
            JMS, O                  7B
            TIME                    7C
            JMS, O                  7D
            TIME                    7E
            CLB                     7F
            WMP                     80
            JMS, O                  81
            TIME                    82
            BBL, O                  83

TIME:       LD, 3                   84
            XCH, 13                 85

E:          FIM, 7                  86
            14, O                   87

D:          ISZ, 15                 88
            D                       89
            ISZ, 14                 8A
            D                       8B
            ISZ, 12                 8C
            E                       8D
            ISZ, 13                 8E
            E                       8F
            BBL, O                  90

PERIOD:     JMS, O                  91
            DIT                     92
            JMS, O                  93
            DAH                     94
            JMS, O                  95
            DIT                     96
            JMS, O                  97
            DAH                              98
            JMS, O                           99
            DIT                              9A
            JMS, O                           9B
            DAH                              9C
            JUN, O                           9D
            WAIT                             9E

COMMA:      JMS, O                           EO
            DAH                              E1
            JMS, O                           E2
            DAH                              E3
            JMS, O                           E4
            DIT                              E5
            JMS, O                           E6
            DIT                              E7
            JMS, O                           E8
            DAH                              E9
            JMS, O                           EA
            DAH                              EB
            JUN, O                           EC
            WAIT                             ED

SPACE:      JMS, O                           FO
            TIME                             F1
            JMS, O                           F2
            TIME                             F3
            JMS, O                           F4
            TIME                             F5
            JMS, O                           F6
            TIME                             F7
            JMS, O                           F8
            TIME                             F9
            JMS, O                           FA
            TIME                             FB
            JUN, O                           FC
            WAIT                             FD

'SPACE':                                     AO
',':                                         AC
'-':                                         AD
'.':                                         AE
'/':                                         AF
O:                                           BO
1:                                           B1
2:                                           B2
3:                                           B3
4:                                           B4
5:                                           B5
6:                                           B6
7:                                           B7
8:                                           B8
9:                                           B9
A:                                           C1
B:                                           C2
C:                                           C3
D:                                           C4
```

```
E:                                          C5
F:                                          C6
G:                                          C7
H:                                          C8
I:                                          C9
J:                                          CA
K:                                          CB
L:                                          CC
M:                                          CD
N:                                          CE
O:                                          CF
P:                                          D0
Q:                                          D1
R:                                          D2
S:                                          D3
T:                                          D4
U:                                          D5
V:                                          D6
W:                                          D7
X:                                          D8
Y:                                          D9
Z:                                          DA

WAIT:        JMS, O                         A1
             TIME                           A2
             JMS, O                         A3
             TIME                           A4
             JMS, O                         A5
             TIME                           A6
             JUN, O                         A7
             INC                            A8
```

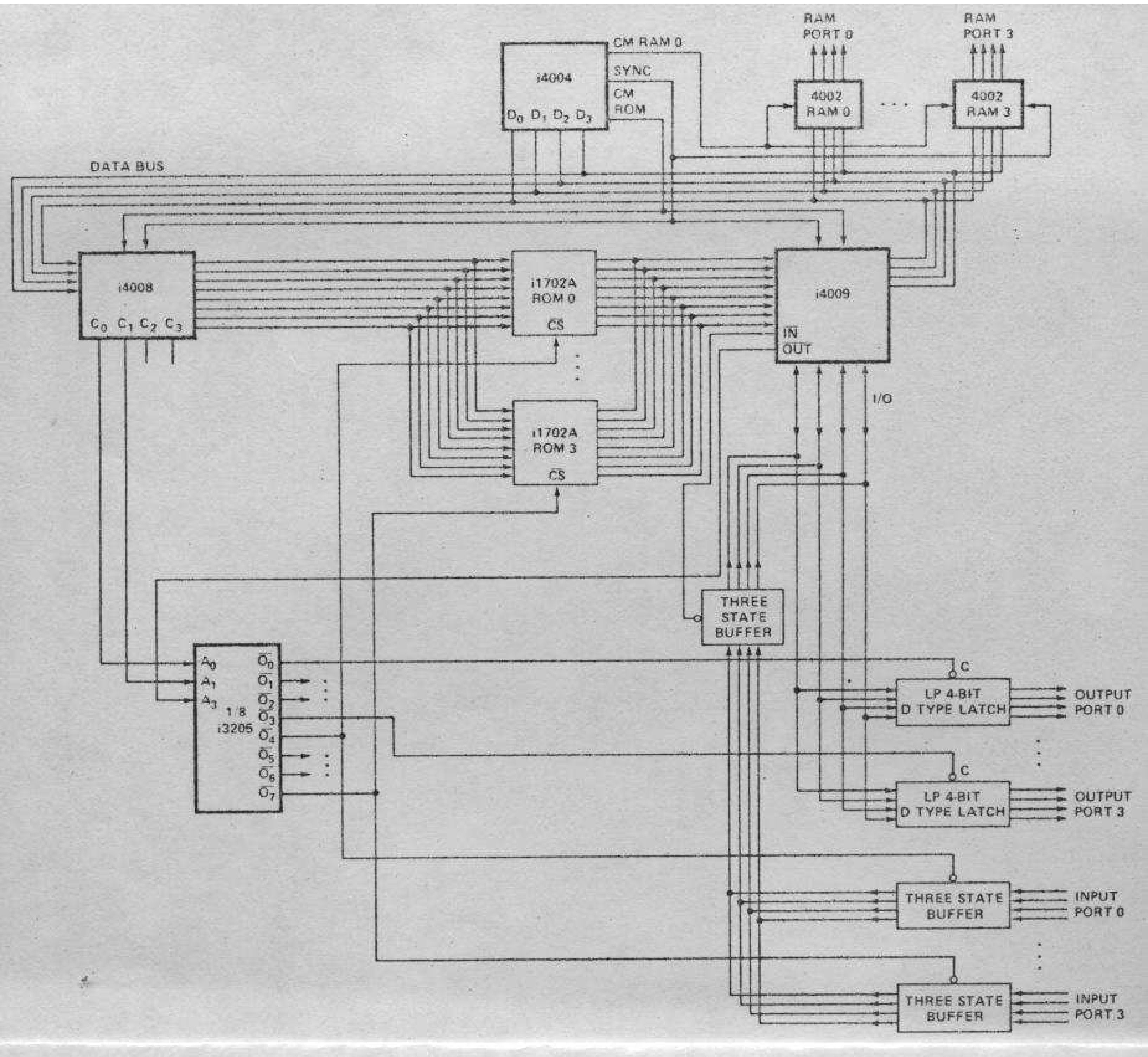


Fig. 2 - USE OF 4008, 4009, AND 1702A

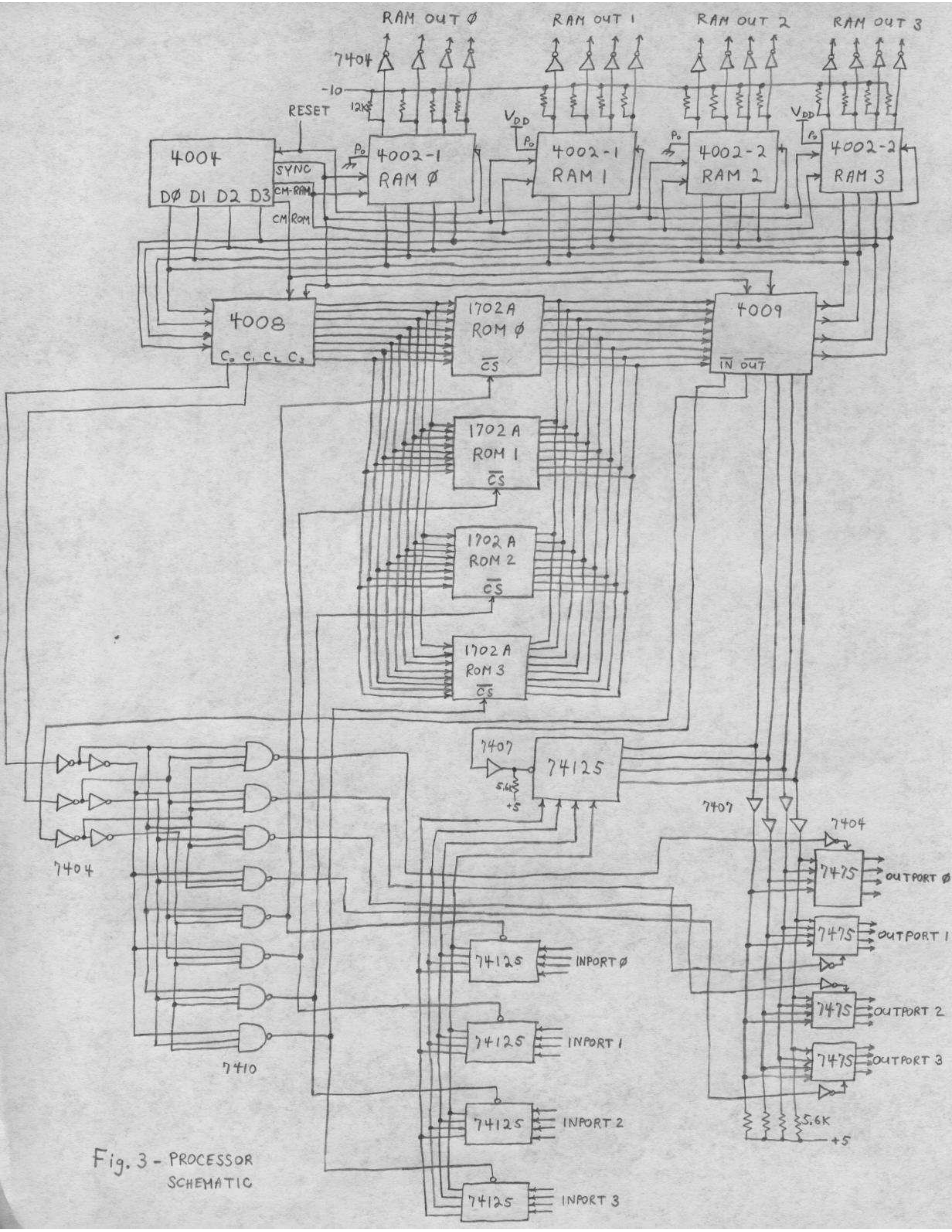
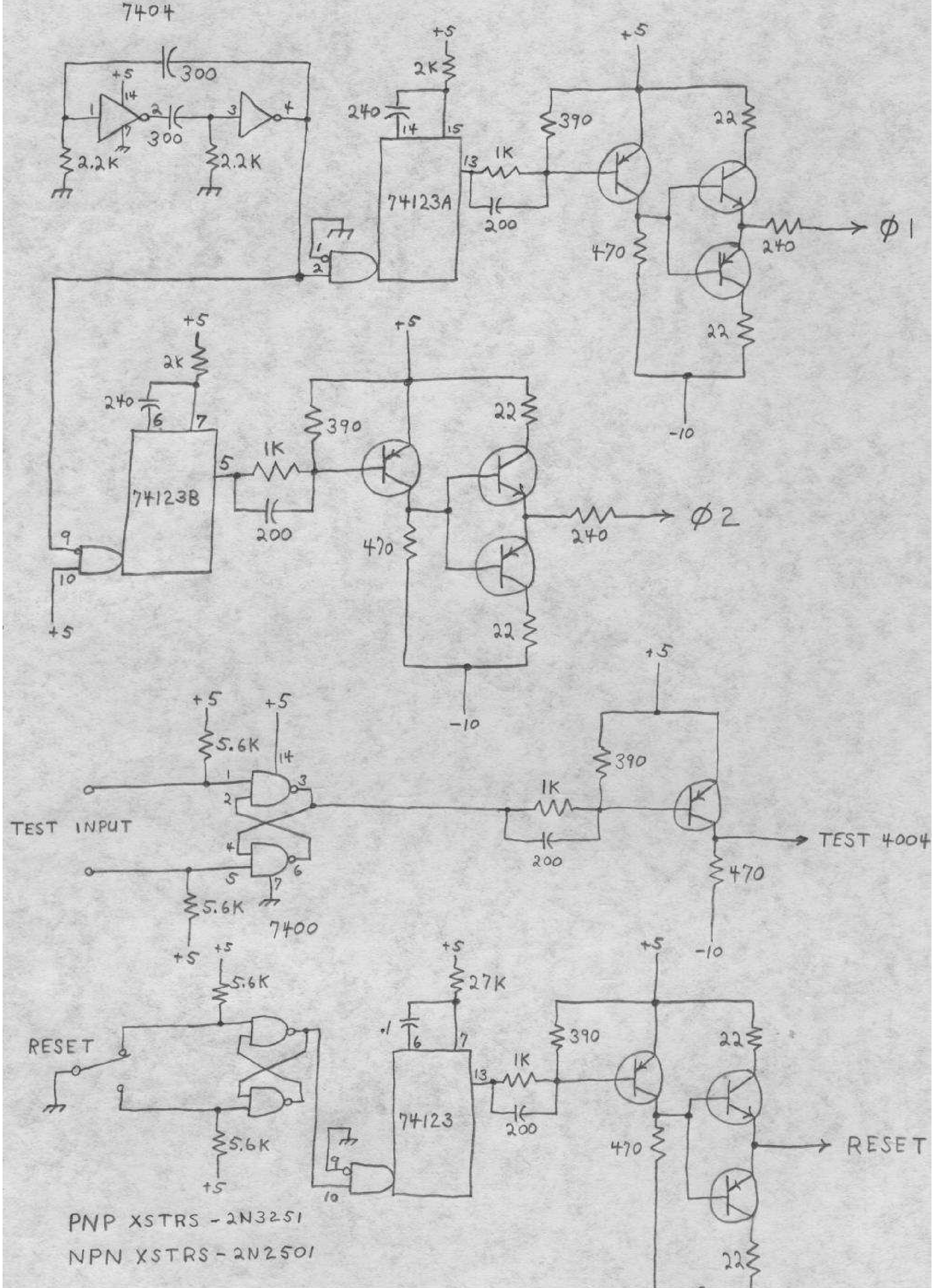


Fig. 3 - PROCESSOR SCHEMATIC



PNP XSTRS - 2N3251
NPN XSTRS - 2N2501

Fig. 4 - CLOCK, TEST, AND RESET GENERATORS

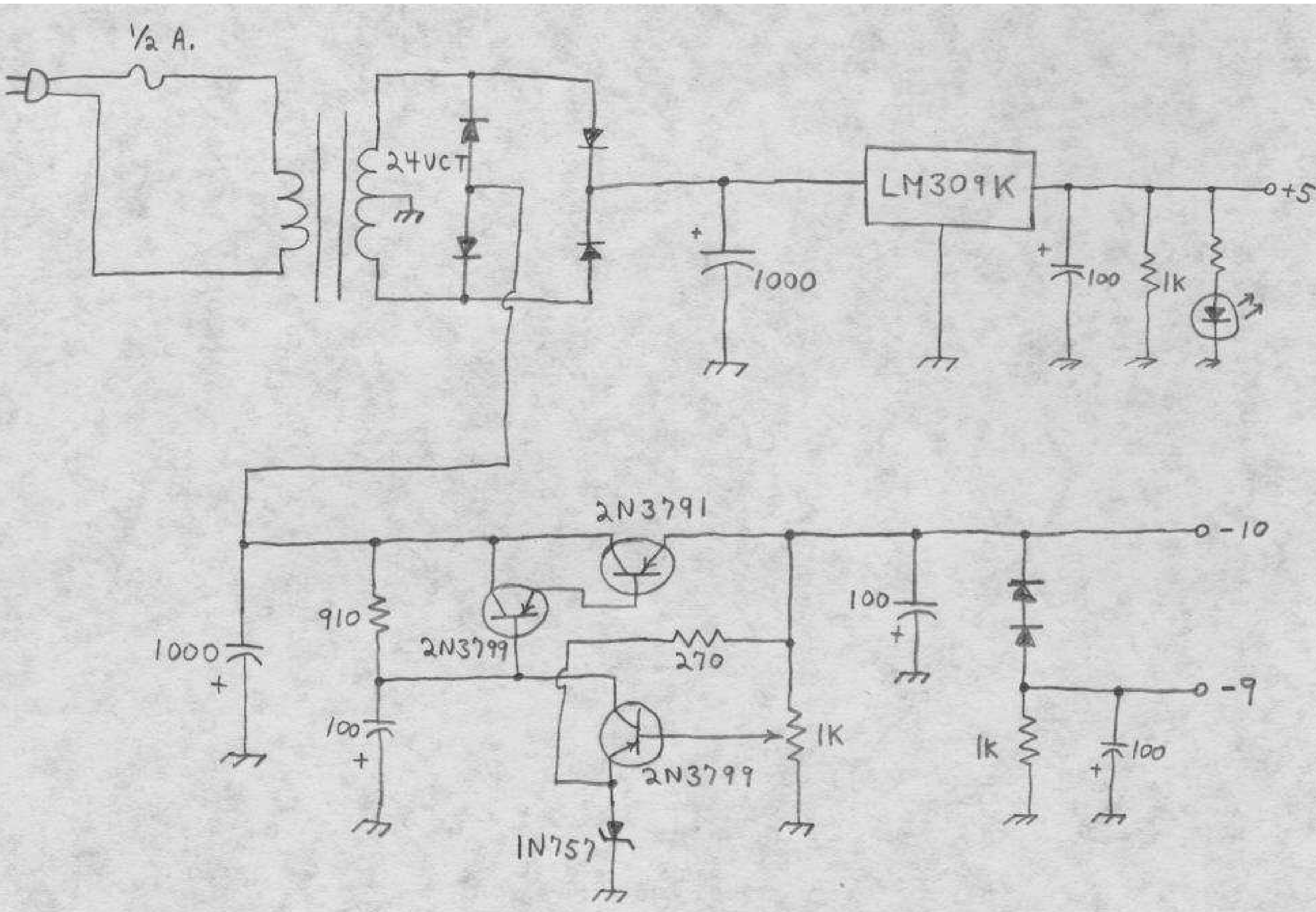


Fig. 5 - PROCESSOR POWER SUPPLY

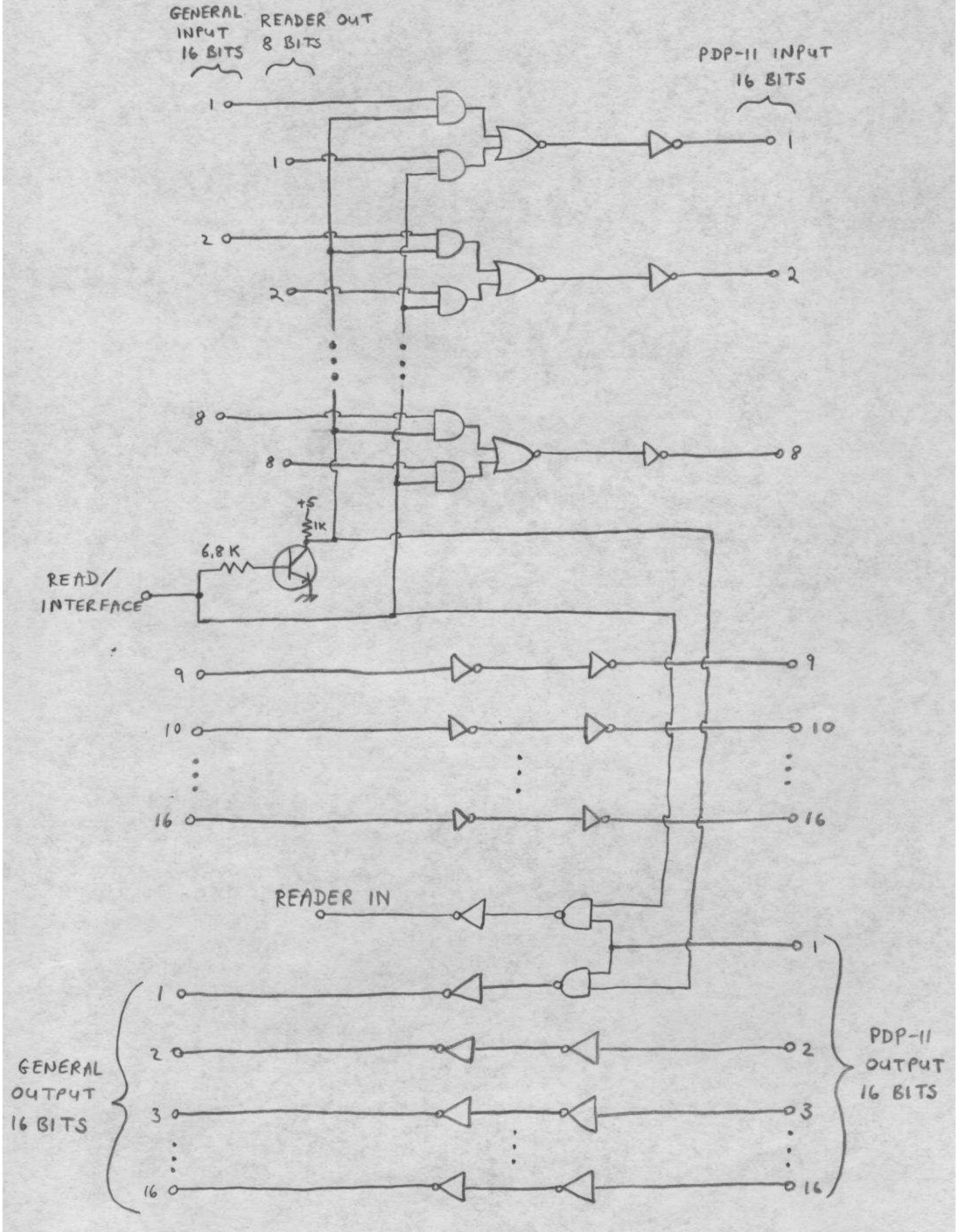


Fig. 8A - LOGIC FOR READER ASSEMBLY

7451 - 4
7404 - 9
7400 - 1

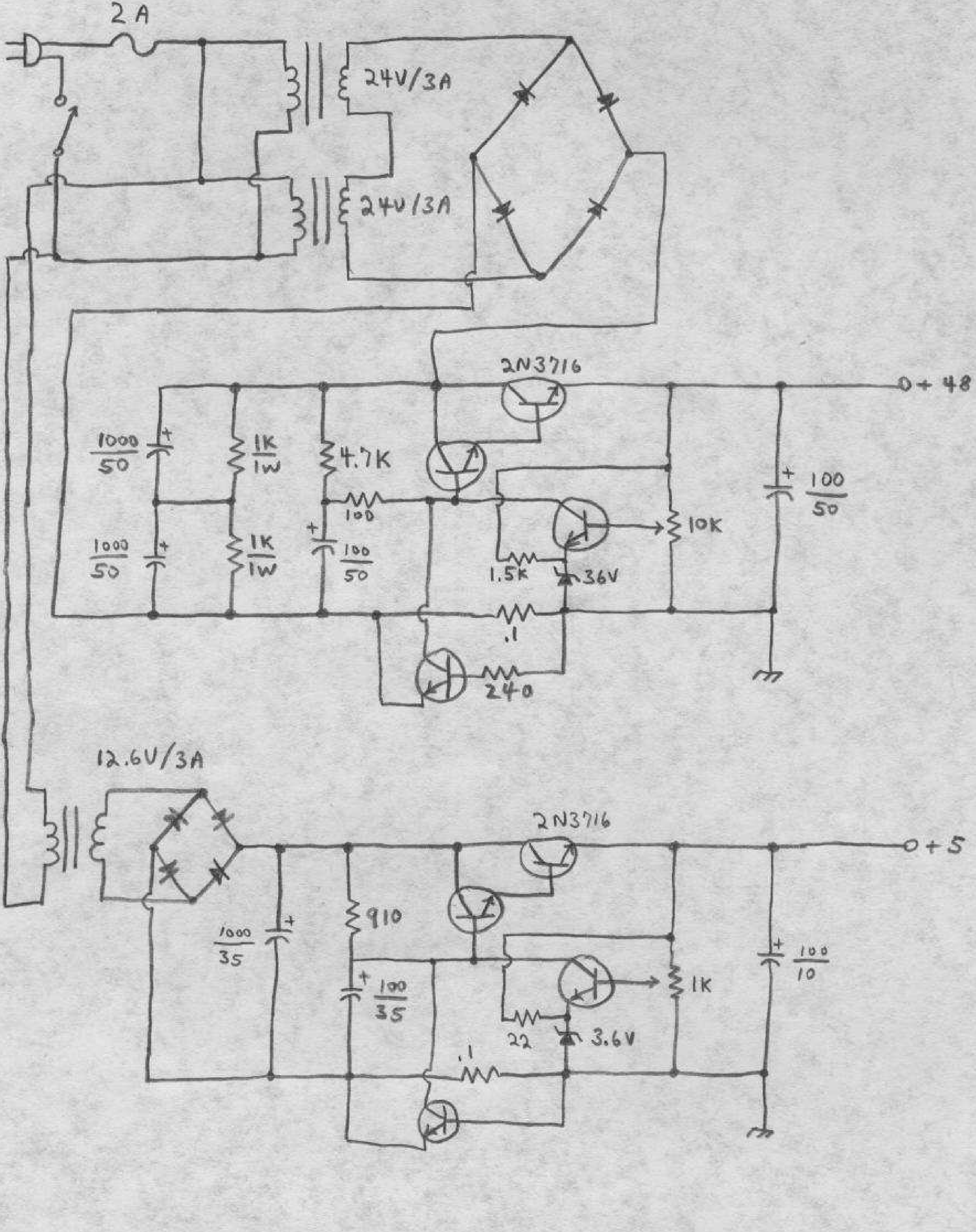


Fig. 8B - READER POWER SUPPLY

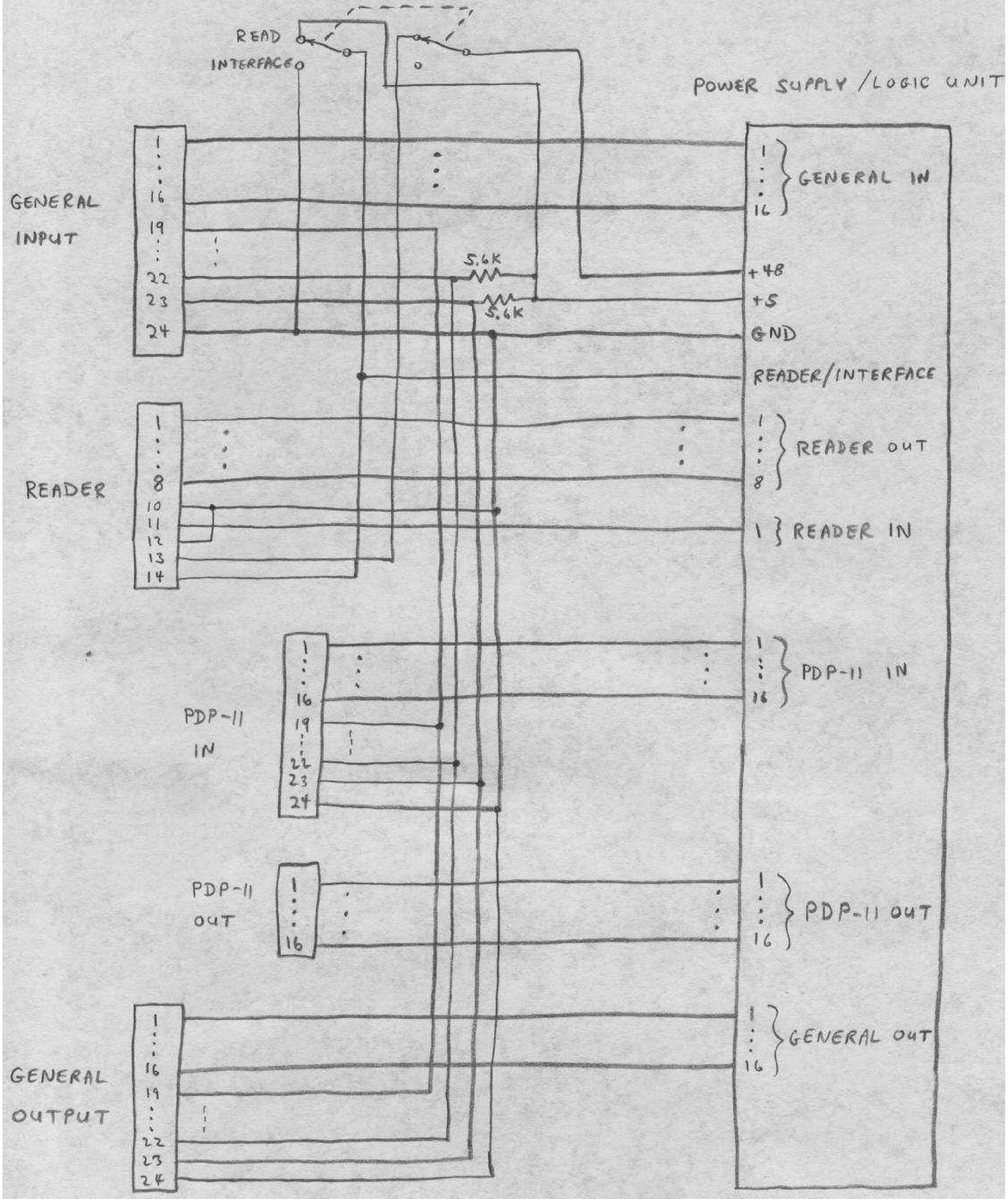


Fig. 8C - WIRING INTERCONNECTION FOR READER ASSEMBLY

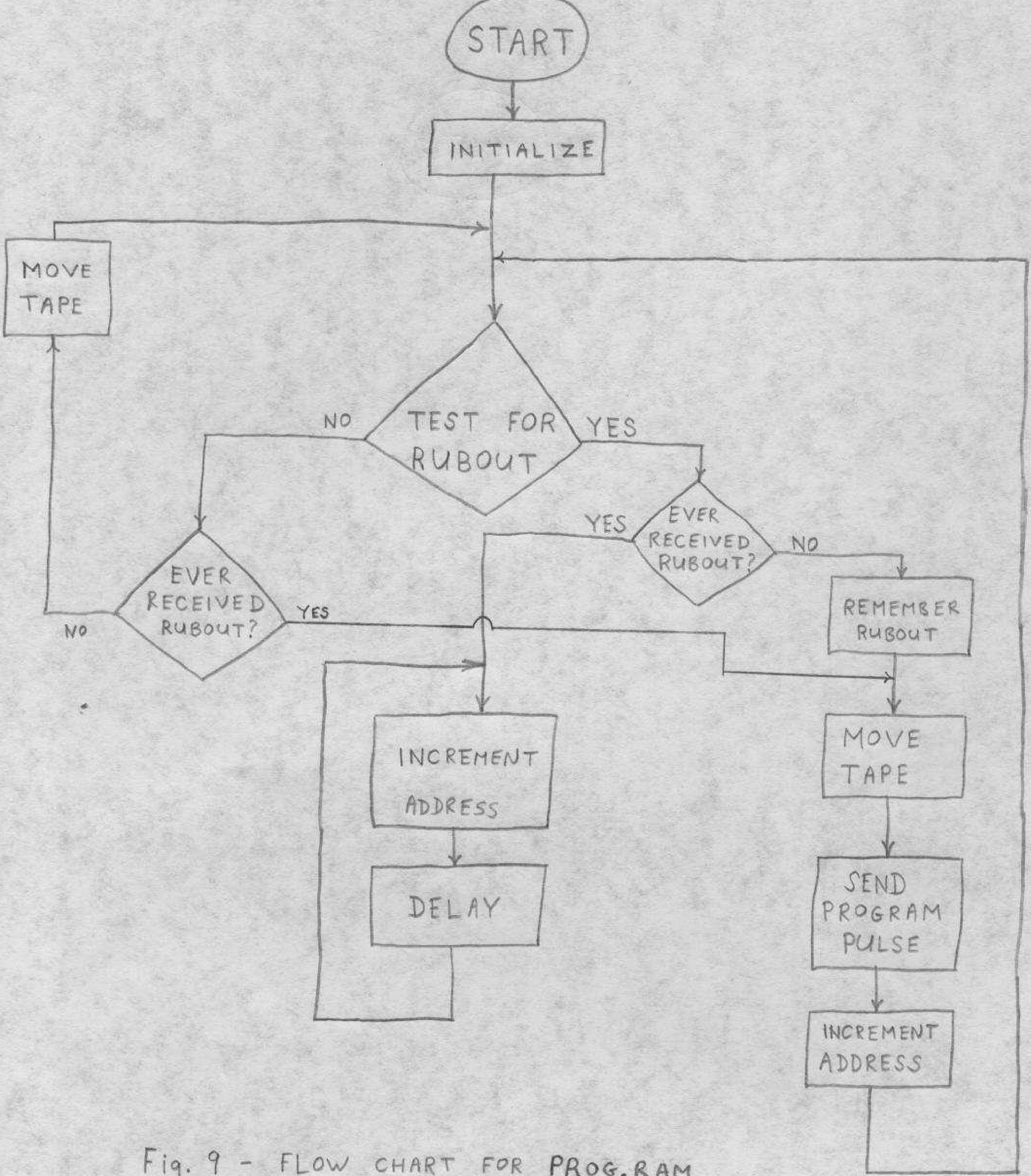


Fig. 9 - FLOW CHART FOR PROG.RAM

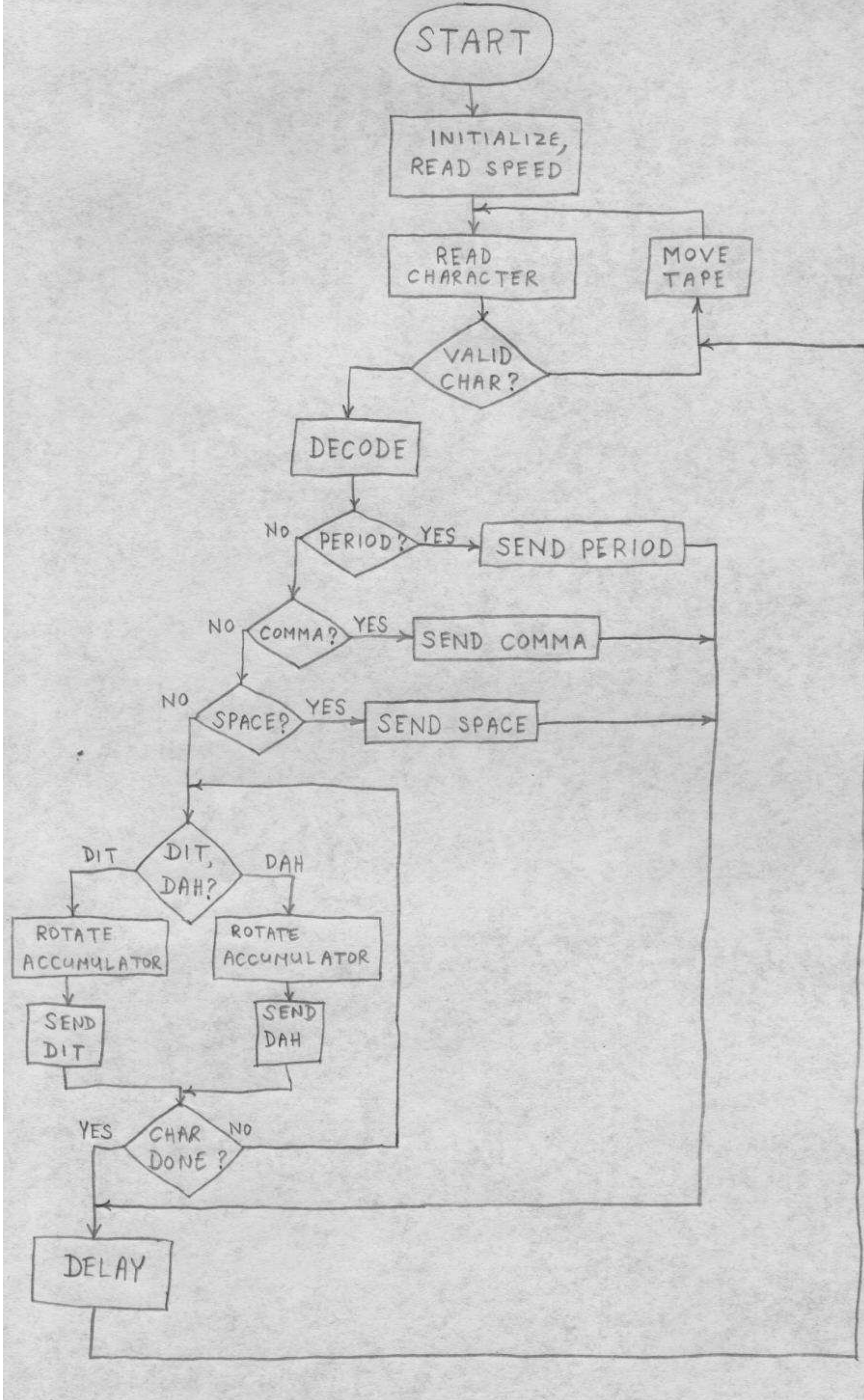


Fig. 10 – FLOW CHART FOR MORSEC.ODE

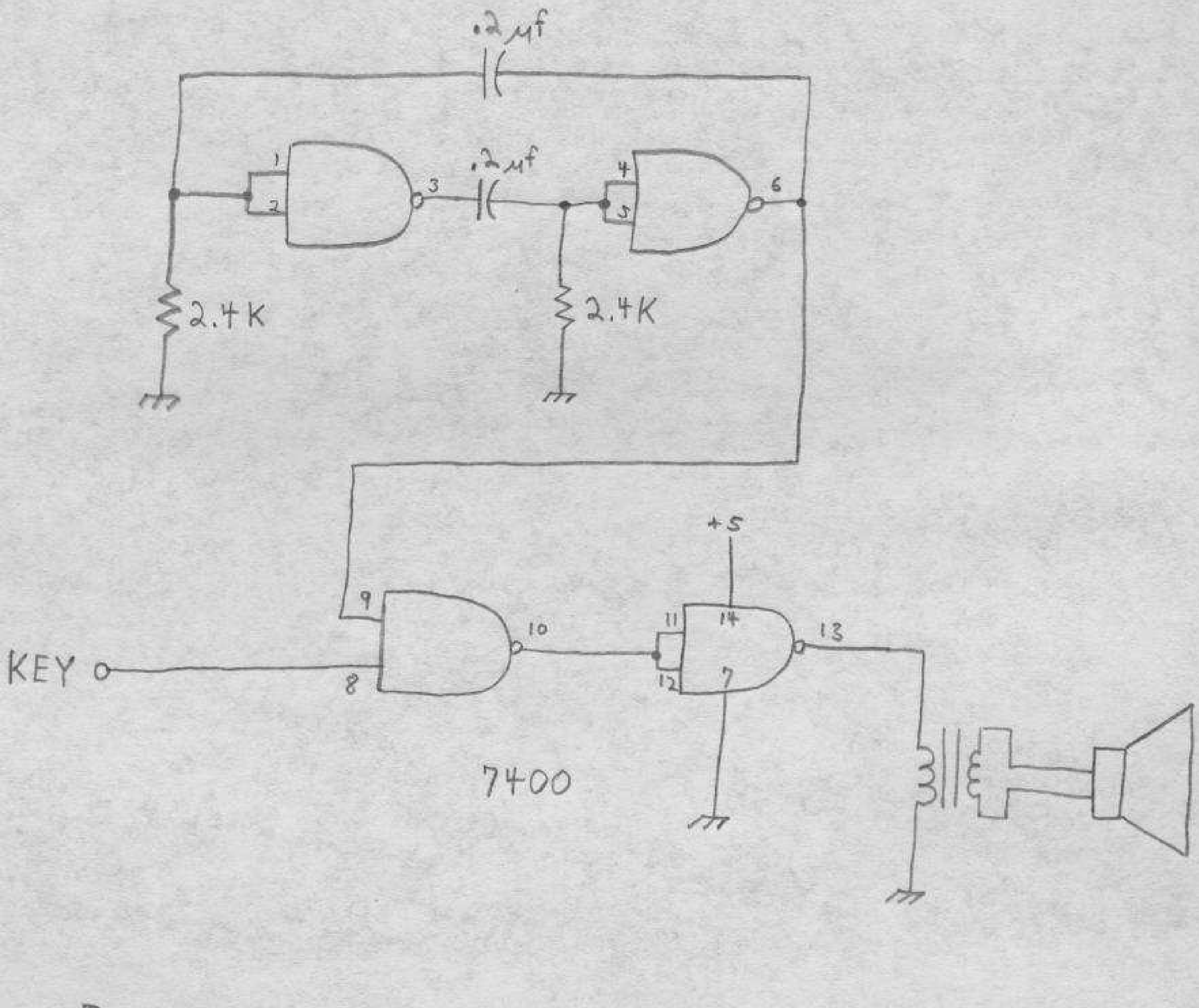


Fig. 11 - CODE OSCILLATOR