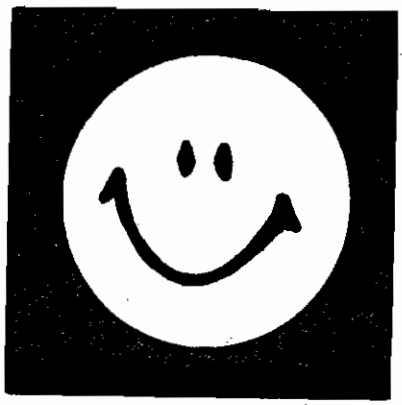
FRED

A

FLEXIBLE RECREATIONAL & EDUCATIONAL DEVICE

BY

JOSEPH A. WEISBECKER



SYSTEM MANUAL

JULY 1972

Herb Johnson Dec 2016

A few years ago, a paper copy was scanned, of a FRED manual of July 1972 from among the documents found as part of Joseph Weisbecker's papers in the Sarnoff Collection. I processed that scan and produced a "clean" PDF file. A copy was provided to the Collection.

In May 2014 I obtained documents from Anthony "Toni" Robbi, a colleage of Joseph Wesibecker at RCA. Those documents included selected pages of the FRED manual of July 1972 (as identified by title page) and later dates. Some of Robbi's documents include pages not found in the Sarnoff Collection's Weisbecker copy. Some were better or worse copies of pages.

I've merged Robbi's pages into an updated version of the 1972 FRED manual. Robbi's FRED documents were used in legal proceedings, so they happen to have a number stamp sequentially per page, from "36289" and up. This allows one to identify those pages, in this PDF document.

The Weisbecker document as previously described, goes from the title page, to a series of handdrawn schematics, to text pages starting with page number 28. Robbi's pages start with the same title page. Then an acknowledgement page, an index, then a text page "Section I" followed by page numbers 2 through 27, which are not in Weisbecker's copy. The images in Robbi's copy which appear to be available in Weisbecker's copy, are generally are not superior in quality.

It is plausible to assume, that Robbi's copy was of pages borrowed (but not returned to) the copy from Weisbecker's papers. Again: Robbi's pages have a stamped sequence number from 36389 or higher, which identifies them.

The scope of the combined document is as follows. Almost every document has Weisbecker's name or initials. There's about 150 physical pages.

Physical pages 1 to about 100, seem to be "the FRED manual" of July 1972. There's about 40 numbered text pages within the document, which describe the FRED computer. There's many hand-drawn schematics, illustrations. There's many pages of hand-coded listings on a standardized coding form. These pages, seem to comprise the "FRED manual" as suggested by the table of contents page. That page indicates the "sample programs" "subroutines" are in the last section of the manual.

At about physical page 100, are a series of documents under RCA letterhead, addressed to "Microprocessor Group" and dated Aug 1972 and later. These appear to be updates, design and coding changes, for the FRED computers produced. They are arranged in the order

scanned when found. The first is a note from Dec 1972, about a change in memory chips.
The second and remainder, are titled "FRED NOTE #" and are numbered from 1 to 16 in order.
#1 is dated Aug 1972; #16 is dated July 1973. FRED Note #1, is a 1972 description of a two-chip implementation of the "processor" and the "external device interface", four chips for memory or other I/O, and discussion of "1976 target costs for a final product". The first LSI product was a 2-chip solution before consolidations into a single LSI 1802 processor.

Herb Johnson

---- ACKNOWLEDGEMENT ----

The experimental FRED prototype described here has been made possible through the efforts of the following:

A. D. Robbi

F. M. Russo

A. R. Marcantonio

R. O. Winder

---- FRED ----

SECTION I - SYSTEM DESCRIPTION

A. GENERAL

FRED is an experimental model of a new computer system for use in schools and homes. This type of computer should cost no more than a Hi-Fi system or set of encyclopedia. Low cost will result from large scale integration (LSI) of electronic circuits combined with clever system design.

In 1965 the circuits used in FRED would have cost $10,000. The current experimental version of FRED contains about $400 worth of standard TTL and MOS circuits. By 1976 the circuit cost should be less than $80 (assuming custom LSI). To this, of course, must be added nominal packaging costs.

Even with inexpensive electronics, input/output devices could push costs above acceptable levels. A simple teletype machine costs almost twice as much as the target selling price for FRED.

FRED input/output costs are reduced to an absolute minimum by the use of any ordinary, unmodified TV set and audio cassette player. These will already be owned by many FRED users.

The TV set display provides economical output capability in a form suitable for a wide range of game and educational uses. The audio cassette recorder provides a convenient means for loading programs, and can be used for computer controlled audio presentations as well (including step by step operating instructions).

In addition to the above, FRED incorporates a simple gravity fed card reader permitting user responses and/or parameter entry. This device has no moving parts and provides a much more flexible input mode than a simple equivalent cost keyboard.

Despite its simplicity and low cost FRED still maintains all the characteristics of a stored program machine. It can be programmed for any number of different functions as simply as changing cassettes. A large FRED customer base would also comprise a significant continuing market for support kits, worksheets, booklets, cassette programs, and optional attachments to extend the usefulness of the minimum system.

B. SYSTEM BLOCK DIAGRAM

The block diagram of the experimental FRED prototype is shown in Figure 1. Unlike a final product this system utilizes standard TTL and MOS circuit packages. (Refer to manufacturers' literature for circuit details). A production version of FRED would utilize custom LSI circuits and would not include a debug panel or hex keyboard. These latter components have been included in the experimental version to facilitate program preparation and debugging. A product version of FRED would use only prepared programs (in a minimum system configuration).

A number of other devices and FRED enhancements are possible. These will not be described here. Separate reports will be issued to describe such alternatives as they are developed.

Logic and memory circuits for the experimental FRED prototype are contained on seven printed circuit cards (M1, P1, P2, P3, E1, E2, E3) as shown in Figure 2.

The memory (M1) utilizes 40 chips, 32 of which are 256 bit static MOS RAM chips. By 1976 an equivalent 1,024 byte memory should be available in the form of 2 - 4096 bit packages at a cost close to 0.2¢/bit.

The computer logic is contained on two printed circuit cards (P1, P2) and utilizes about 90 standard 7400 series TTL packages currently costing approximately $90. Slight design modifications would permit the computer logic to be fabricated on one 40 pin custom LSI chip. Chip complexity would be on the order of 1000 gates combined with a 256 bit RAM.

P3 contains clocking and switch interface circuits which could be reduced considerably for a final product design.

E1, E2, and E3 average 26 or so TTL packages each plus several phase locked loop chips. They represent the interface circuits required for the TV display, cassette start/stop or read/write, and the card reader. Final product design would not require tape write capability with further interface cost reduction achieved via custom MSI.

C. COMPUTER

The computer is byte oriented. Data and instructions are stored in memory as 8 bit bytes. Up to 64K memory bytes can be addressed although only 1,024 memory bytes are provided in the minimum system. Each instruction requires two machine cycles for execution. A machine cycle is about 2.5μs in duration resulting in an instruction execution rate of 200K ops/sec. A direct memory access channel permits independent, asynchronous input/output at rates up to 100,000 bytes/sec. External program interrupt is also provided.

Figure 3 shows the data flow paths within the computer. (Detailed logic will be found in Section IV.) Register naming, and memory addressing conventions are described below.

RØ, R1 represent an addressable array of sixteen 16-bit R registers. RØ is the least significant R register byte and R1 is the most significant R register byte. X, P, and N are three 4-bit registers. The contents of X/P/N specify one of the 16 R registers. R(N) is used here to denote the R register specified by the 4 bits contained in the N register. RØ(N) specifies the low order 8 bits (byte) of the R register selected by N. R1(N) specifies the high order byte of an R register. The contents of a specified R register (2 bytes) can be transferred to A and C registers. The 16 bits in A are used to address M. The 16 bits in C can be incremented or decremented by "1" and written back into R.

M(R(N)) refers to a one byte memory location addressed by the contents of R(N). This indirect addressing system is basic to the simplicity and generality of the processor.

D is an 8 bit data register which has one bit right shift built in. F is an 8 bit logic network for performing binary add, subtract, logical and or, and exclusive or on two 8-bit operands. One of the operands is the bus byte and the other is the contents of D.

I is a 4 bit (digit) instruction register. 4 bit operation codes are placed in this register and decoded to control instruction execution.

Bytes can be read onto the bus from any of the registers, memory, or external interfaces. A bus byte can, in turn, be transferred to a register, memory, or external interface.

Most of the R registers are available for use as data registers, address registers, or program counters. However, if applications requiring external interrupts or cycle stealing are anticipated, R(Ø), R(1), and R(2) should be reserved for these functions. Detailed operation of the computer is best described in terms of its instruction set. This description will be found in section II.

D. TV INTERFACE

FRED is connected to the antenna terminals of any standard TV set. Any contiguous 128 byte memory area can be displayed as an array of 1024 dots on the TV screen. A "1" bit is displayed as a white dot. A "Ø" bit leaves its display position on the screen black.

The 128 memory bytes (1024 bits) can be displayed as a 32x32 dot array (Figure 4) or as a 16x32 dot array (Figure 5). Memory byte ØØ+base address always occupies the upper left hand corner of the TV screen. Bits within a byte are displayed as shown.

The instructions provided for activating the TV display and selecting display formats are described in Section II. The detailed logic of the TV set interface are provided in Section IV.

After the display has been selected and activated by a program, the TV interface circuits cause the following actions.

Every 17ms a program interrupt occurs. This corresponds to the TV vertical sync pulse and indicates the start of a new TV frame (approximately 60 times/sec.). The system programmer must provide an interrupt routine (see sample "DISPLAY" program in Section V) which resets R(Ø) to the start of the desired 128 byte memory display area. About 4ms after each program interrupt the TV interface circuits will request display bytes from memory as sequentially addressed by R(Ø). These bytes will be supplied by the direct memory channel (via cycle stealing) completely independent of normal program execution which is resumed following program interrupt.

In the 32x32 display mode M(R(Ø)) bytes will be supplied to the TV set with the approximate timing shown in Figure 6A. The approximate timing for the 16x64 display mode is shown in Figure 6B.

Note that when the display is activated only 128x60 machine cycles are required for display refresh each second. This means that display refresh degrades the normal computer instruction rate by well under 5%.

E. CARD READER

FRED utilizes a 3"x5" punched card as input. Figure 7 illustrates this card. Each card edge (X and Y) can have up to eight bytes punched. Each byte is punched as two hex digits. Each hex digit has a parity bit (P) on the card. Odd parity insures that each hex digit will have at least one hole punched. Edge X of the card shown would be read into memory as the following sequence of bytes: $1-23-45-67-89-AB-CD-EF.

The reader contains six light sources and photo-diodes (1, 2, 4, 8, P) to sense the card holes. Photodiode "C" indicates that a card is being dropped past the read head. Top and bottom card notches prevent misreading the top/bottom edge as a digit.

Cards are manually dropped into a slot and read photoelectrically while falling. The cards fall into a hopper and maintain their order when removed. This type of reader provides an extremely inexpensive means for program and parameter loading. It is particularly attractive for loading short programs. Plastic cards could be provided for use by younger children.

Card reader interface circuits collect hex digit pairs into bytes which are then stored in memory under program control. Interface circuits also check parity of each card digit as read. Parity bits are removed before each hex digit pair enters memory.

F. CASSETTE PLAYER

Almost any inexpensive audio cassette player can be attached to FRED via jacks A and B. Jack A should be plugged into the "remote start-stop" jack of the player. Jack B should be plugged into the "remote speaker output" jack of the player. These connections permit the recorder to be controlled by program.

A headphone jack (F) has also been provided. This permits use of FRED in environments where audio material on tapes would distract a non-user. Headphones used should have their own volume controls. Jacks A and B are located on the back panel. Jack "F" is located on the front panel.

In general the cassette player is first "REWOUND" then placed in the "PLAY" state with its volume control set at midrange. From this point on the program can start and stop the player. A computer speaker can also be switched on and off under program control. This permits running the tape past certain sections with no sound.

A 4.2kc tone recorded on tape causes the tape to stop automatically at the end of the tone. This feature permits the tape to be divided into "FRAMES" which can be sensed by program.

Another mode of tape operation permits data/or program bytes to be read from tape and stored sequentially in memory. Figure 8 illustrates the format in which bytes are recorded on tape.

Recording is bit serial. Each byte on tape is represented by 10 bits. A start bit (S) is always "1". This is followed by the 8 byte bits (Ø-7). A parity bit (P) makes the 10 bits even. Parity and start bits are used by the interface circuits and dropped prior to storing each byte in memory. Each byte from tape is stored sequentially in memory via the direct memory channel (cycle stealing).

Each bit on tape is represented by a 5ms tone burst. A 5.2kc tone represents "0" and a 6.2kc tone represents "1". Tones are separated by 5ms gaps. This recording system results in bits which are self clocking. Bit packing density is on the order of 50 bits/inch. Data is loaded in memory at a 10 byte/second rate. The low packing density and self clocking features permit reliable operation with standard audio cassette tapes and players.

G. OPTIONAL SYSTEM COMPONENTS

These components have been included in the experimental FRED prototype for program preparation and debugging. They would be omitted from a final minimum product.

Cassette interface circuits include those required for recording memory bytes on tape. Procedures for recording bytes will be described in Section IV in the "WRITE" program.

Control switches will be discussed in Section III under operating procedures.

A hex switch box has been provided. This set of 16 switches connects to FRED via the card reader interface circuits and functions as a direct substitute for the card reader. Any byte can be entered by depressing two hex switches. The most significant hex digit of a byte must be entered first. Program control of the hex switches is identical to that for the card reader.

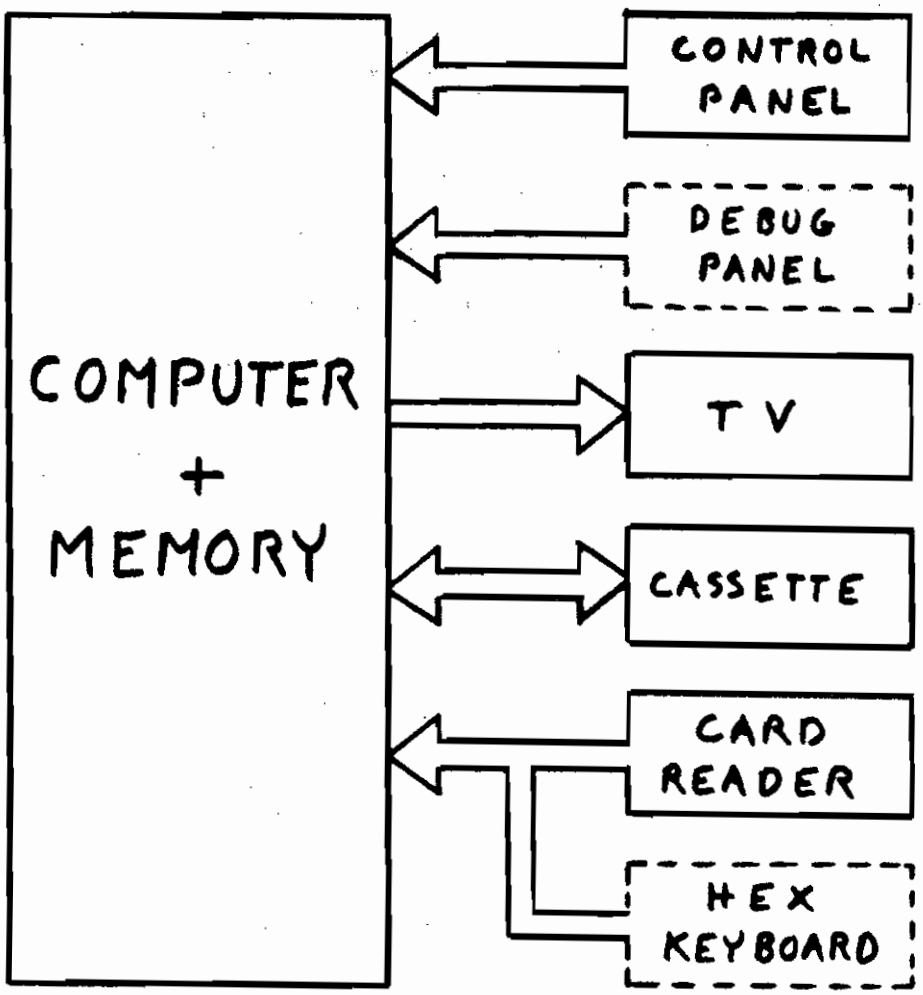
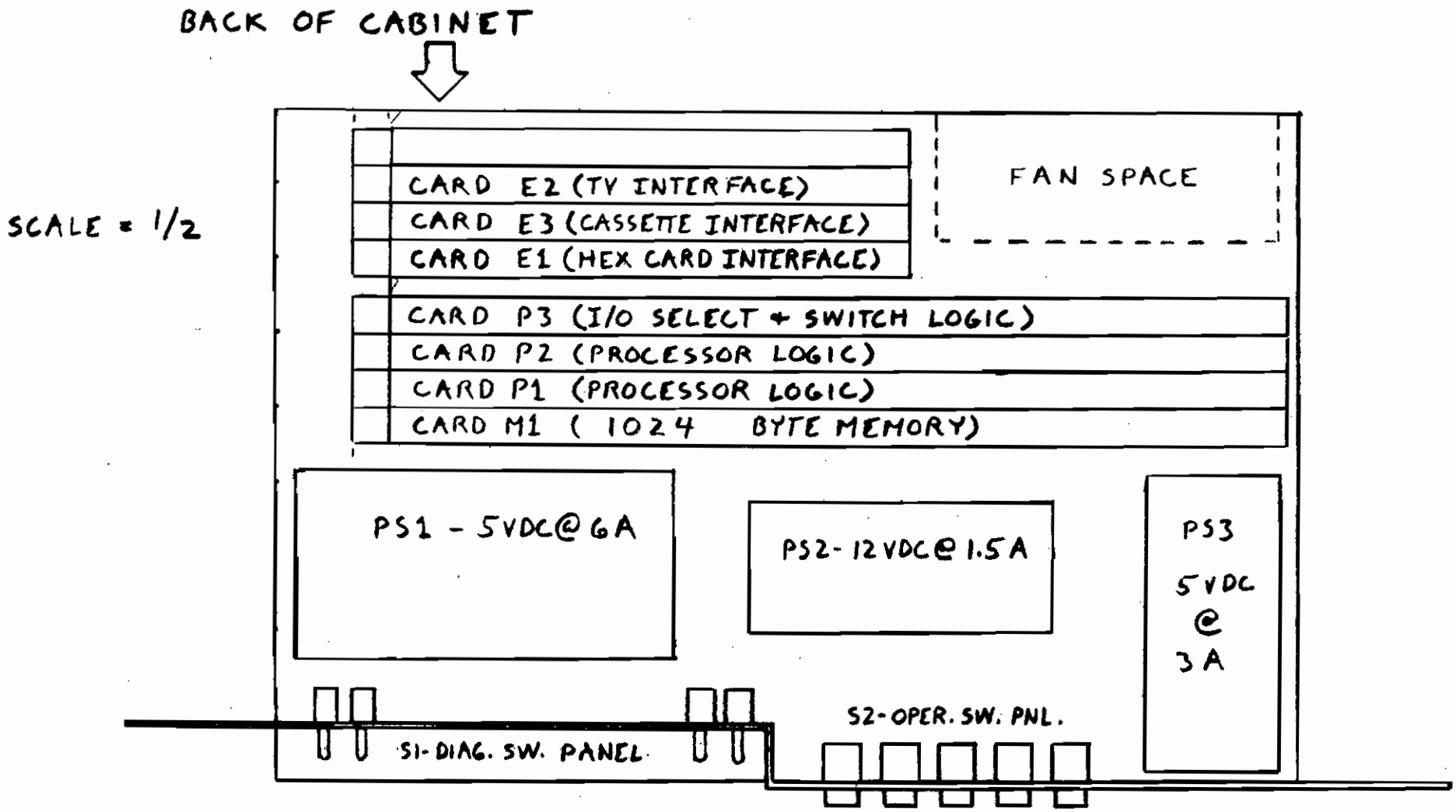


FIGURE 1

SYSTEM BLOCK DIAGRAM

JAW



TOP VIEW OF MAJOR COMPONENT LAYOUT

NOTE: DIL PACKAGES ARE MOUNTED ON FRONT SIDE OF PLUG-IN CARDS
TOP PIN (1) = +5
BOTTOM PINS (B5-B6) = GND

FIGURE 2

JAW 3-1-72

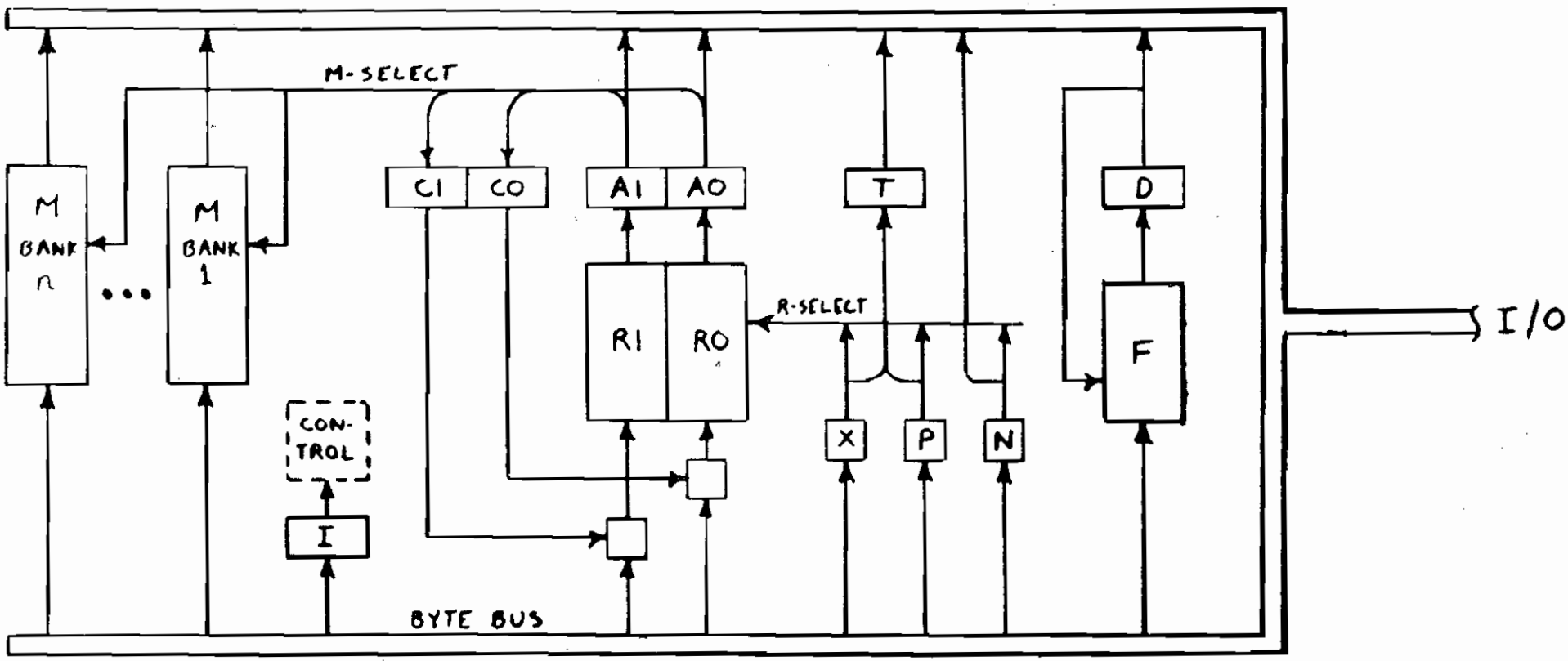
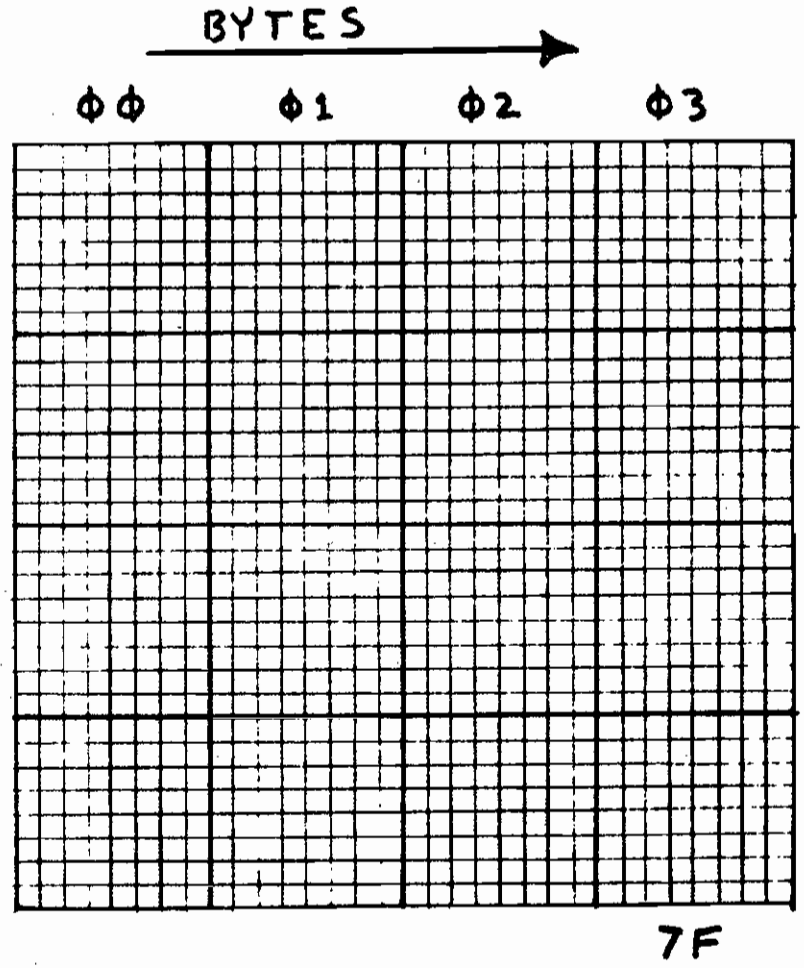


FIGURE 3
COMPUTER BLOCK DIAGRAM

JAW



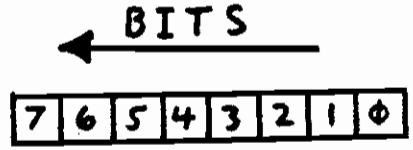


FIGURE 4

32 x 32
TV DISPLAY

PMR

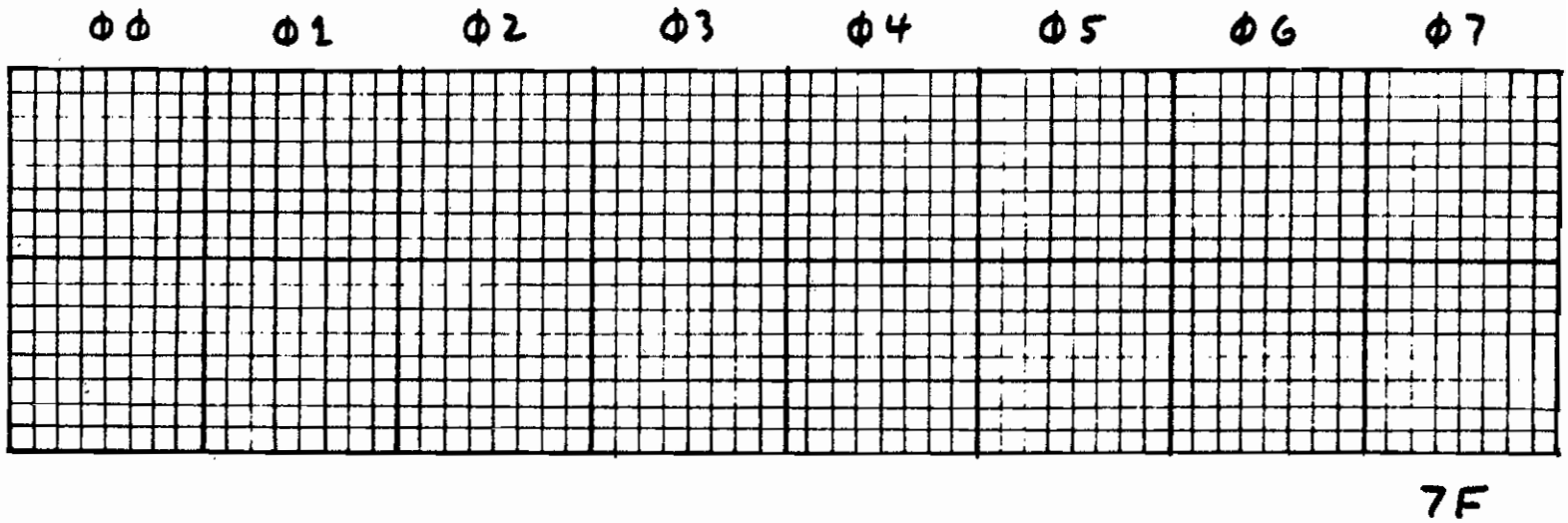


FIGURE 5

16 × 64 TV DISPLAY

PMR

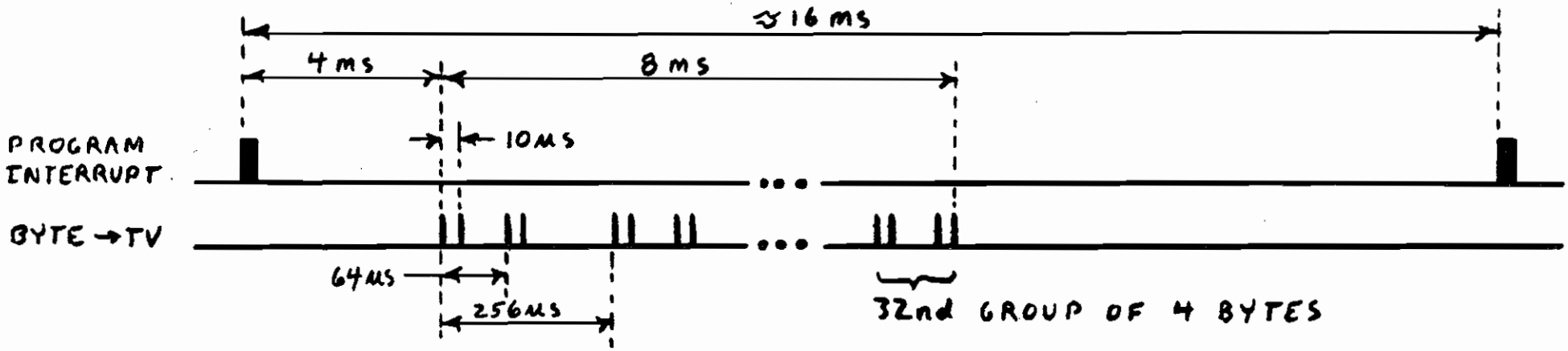


FIG 6A - 32 × 32 TV TIMING

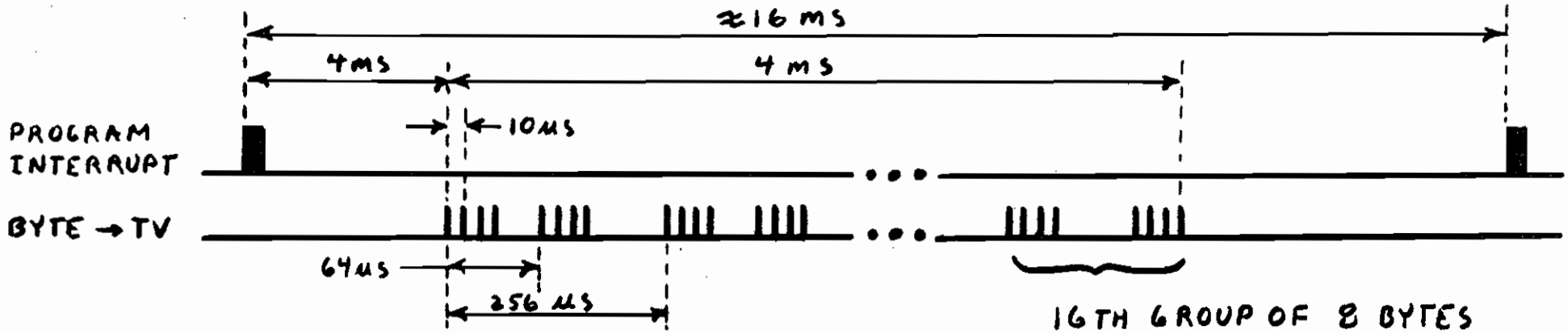


FIG. 6B - 16 × 64 TV TIMING

JAW

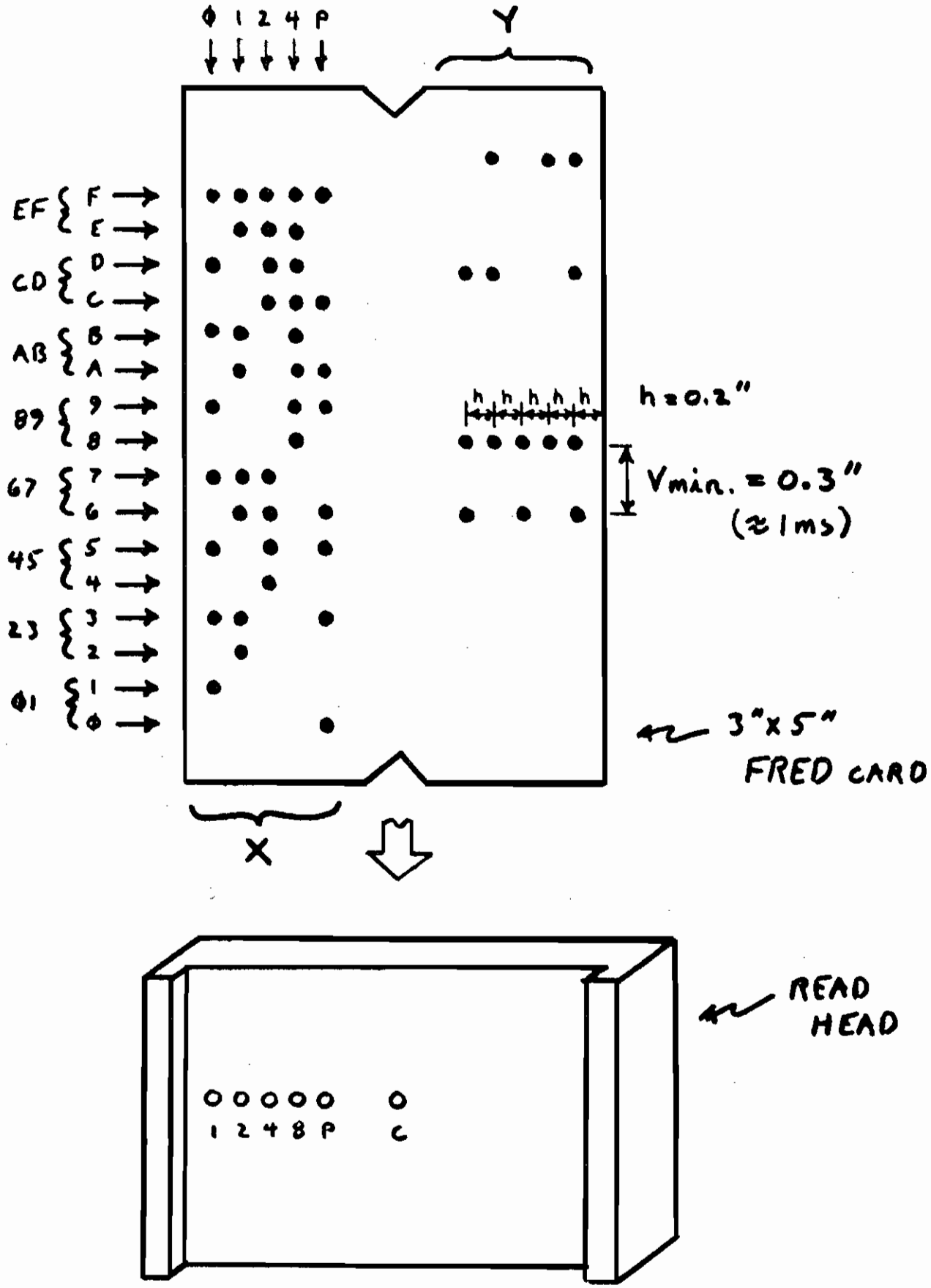


FIG. 7 - CARD FORMAT

JAW

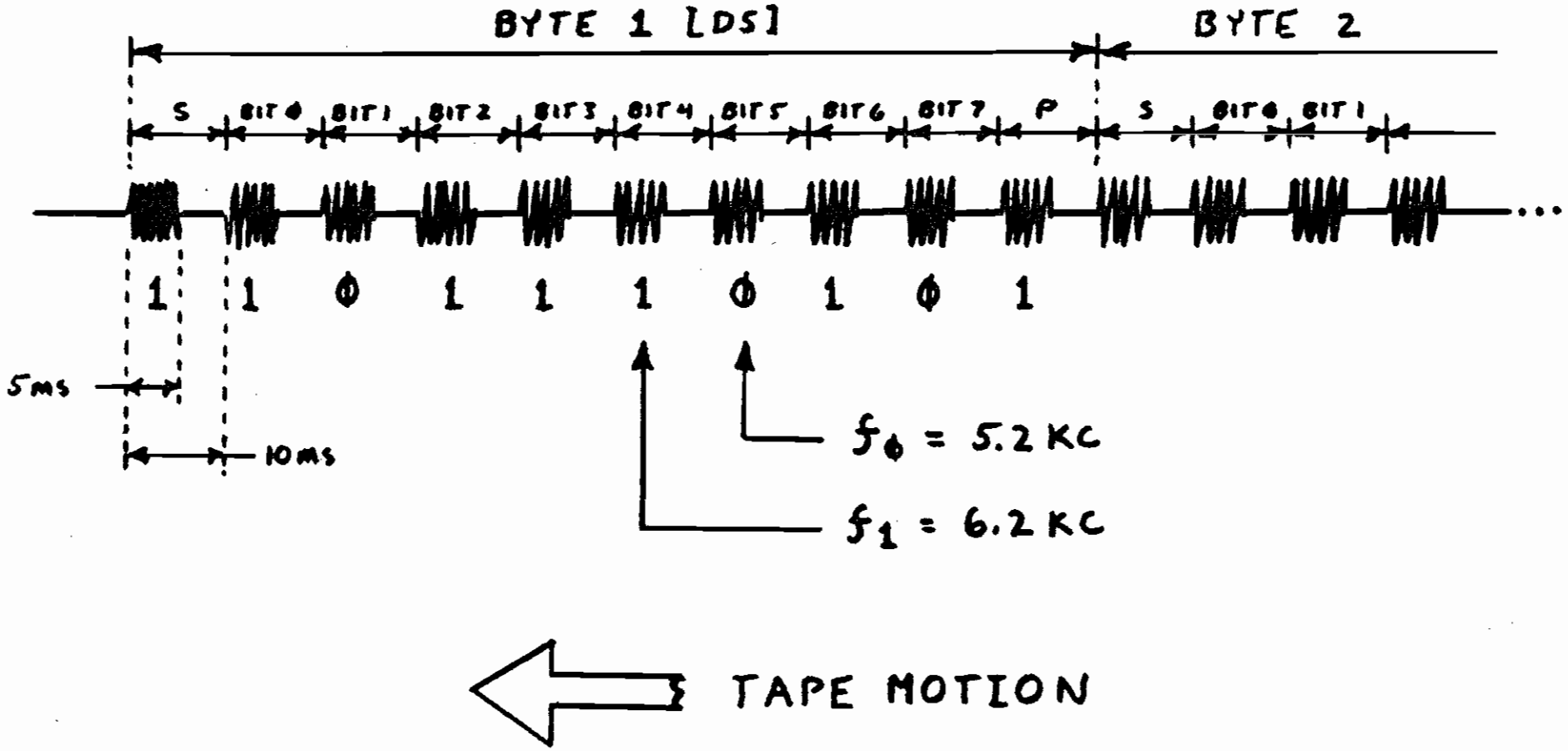


FIGURE 8 - TAPE DATA FORMAT

JAW

SECTION II - ORDER CODE

A. PROCESSING INSTRUCTIONS

The operation of the computer is best described in terms of its instruction set which is shown in the micro-instruction summary (figure 9).

A one byte instruction format is used. Two machine cycles are required per instruction. The first machine cycle causes an 8-bit instruction to be fetched from M and placed in the I and N registers [M(R(P))→I,N]. This is accomplished by gating the contents of P to select R. R(P) is then gated to A and C. While waiting for the M(R(P)) byte access time, C is incremented by 1 and replaces the original contents of R(P). The most significant digit (4 bits) of M(R(P)) is gated to I via the bus. The LSD of M(R(P)) is gated to N. At the end of the instruction fetch machine cycle, I and N contain the 8 bit instruction originally addressed by the program counter [R(P)] and the program counter has been incremented by 1 so that it now points to the next instruction byte in sequence.

At this point it should be noted that any of the 16R registers could be used as the program counter. Multiple program counters are facilitated.

The next machine cycle always executes the instruction contained in I and N. The execution of the data handling and branch instructions are described below (I2 denotes that the digit in I has the value of 2):

I1 - R(N)+1
The 16 bits in the R register specified by the current digit in N is incremented.

I2 - R(N)-1
The 16 bits of R(N) are decremented by 1.

I4 - M(R(N))→D,R(N)+1
The M byte addressed by R(N) is read from M and placed in D. R(N) is incremented by 1.

I5 - D→M(R(N))
The byte in D is written to the M byte location addressed by R(N).

I8 - R0(N)→D
The least significant byte of R(N) is placed in D.

I9 - R1(N)→D
The most significant byte of R(N) is placed in D.

IA - D→R0(N)
The byte in D replaces the least significant byte of R(N).

IB - D→R1(N)
The byte in D replaces the most significant byte of R(N).

IC - D0→R00(N)
The least significant 4 bits (digit) in D replaces the least significant digit of R(N).

ID - N→P
The 4 bit digit in N is placed in P. This effectively changes the current program counter and constitutes a branch.

IE - N→X
The 4 bit digit in N is placed in X.

IF - Perform function specified by digit in N:
N0 - M(R(X))→D
N1 - M(R(X)) "OR" D→D
N2 - M(R(X)) "AND" D→D
N3 - M(R(X)) "EXCL.OR" D→D
N4 - (M(R(X)) + D→D [BIN.ADD,FINAL CARRY→DF]
N5 - M(R(X)) - D→D [BIN.SUBT.,FINAL CARRY→DF]
N6 - SHIFT D RIGHT 1 BIT [LSB→DF]

Note that a flag bit (DF) is provided. This flag can be tested by the following branch instruction.

I3 - Conditional branch
N specifies the condition to be tested.
N0 - unconditional branch
N1 - byte in D not all zeros
N2 - byte in D all zeros
N3 - D flag (DF) equals 1
N4 - external flag 1 set
N5 - external flag 2 set
N6 - external flag 3 set
N7 - external flag 4 set

The last four tests will be discussed with I/O device instructions. If the condition specified by N exists the M byte following the I3 instruction is read from M and replaces the least significant byte of R(P). This permits direct branching within a 256 byte mini-page. If the specified test condition is not present the M byte following I3 is skipped and the next instruction in sequence will be fetched.

The "DECUCE" program described in Section V illustrates the use of many of the above instructions.

B. CARD READER INSTRUCTIONS

In general input/output devices are activated in a two step process. First the specific device is "selected" by a "61" instruction with the M(R(X)) byte specifying a device number. Next the selected device is "activated" by a "62" instruction with M(R(X)) specifying the desired mode of operation.

The card reader is "selected" by executing a 61 instruction with M(R(X)) = 01. Once selected, the card reader (device #01) remains selected until another 61 instruction is executed or the computer is manually reset.

Execution of a "62" instruction while the card reader is "selected" activates the card reader interface circuits. Two modes of operation (program/direct) are possible.

A "62" instruction with M(R(X)) = 01 "activates" the card reader in the "program" mode. The card reader will remain "activated" in this mode even when no longer selected.

While active, the program mode card logic causes external flag 1 (EF1) to be set after each pair of hex digits is read from a card, indicating that a byte is ready to be stored in memory. A "34" instruction should be

used to test this condition periodically in the program. When the flag is set, a "68" instruction will cause the input byte to be stored at M(R(X)). R(X) is unmodified. Note that the card reader must be selected prior to testing its flag or attempting to store an input byte. When operating in this mode, flag sensing and byte storage should be programmed to occur at a faster rate than anticipated byte entry. The byte storage instruction (68) resets EF1.

The card reader (after selection) can optionally be placed in the "direct" mode by executing a 62 instruction with M(R(X))=Ø2. In this mode of operation each pair of card digits will be automatically stored at M(R(Ø)). R(Ø) will be automatically incremented by 1 following storage of each input byte. Input byte storage will occur even if the card reader is no longer selected. R(Ø) should initially contain the memory address of the first byte to be stored.

When the hex switch panel is used the card reader should be disconnected.

C. TV DISPLAY INSTRUCTIONS

The TV display is always operated in the "direct" mode (cycle stealing) and also utilizes the computer program interrupt facility.

To activate the display, the following program steps are required.

First the display must be selected by a "61" instruction with M(R(X))=Ø2 (R(X) is incremented by 1). Next the display is activated by a "62" instruction with M(R(X))=Ø1/Ø2 (R(X)+1 is performed). If M(R(X))=Ø1 a 32x32 bit display results. If M(R(X))=Ø2 the display format will be 16x64 bits.

As soon as the display is activated, the display interface circuits cause memory bytes to be read from memory as needed. All bits of each

byte are displayed as shown in Figures 4/5. The direct memory access circuits are used by the TV interface. Each byte is read automatically from M(R(Ø)) and R(Ø) incremented by one. The display steals 128x60 or 7680 machine cycles per second to maintain a 1024 bit refresh rate of 60 refresh cycles/sec.

The 128 byte memory area to be displayed must be defined by R(Ø). The TV circuits cause a program interrupt 60 times per second (after displaying byte number 128). This program interrupt always causes an automatic transfer of control to the instruction addressed by R(1) (P is automatically set to 1). An interrupt routine addressed by R(1) should be provided which initializes R(Ø) to the address of the first byte of a 128 byte memory display area.

Two additional actions occur when an interrupt occurs. The contents of X and P are placed in T, and X is set to 2. Instruction I7 is provided to facilitate returning to normal processing following interrupt. I7 with N=8 causes T to be stored at M(R(X)). This instruction permits the interrupted values of X&P to be saved. I7 with N=Ø causes M(R(X)) to be placed in X&P effecting a return after interrupt. The "7Ø" instruction also causes R(X)+1 and an interrupt mask (IM) to be reset. IM is always set by an interrupt. IM inhibits interrupts and must be reset by a dummy "7Ø" instruction when the display is initially activated.

Examples of interrupt programming provided in Section V will clarify the above. Note again that once activated the display will continue to function as above even when no longer selected.

D. CASSETTE CONTROL INSTRUCTIONS

Four type "63" instructions are provided for tape control. These instructions control tape motion and sound directly. No "select" instruction

is required prior to their execution. They are completely independent from the cassette read/write instructions to be described subsequently.

The following assumes that the cassette player has been placed in its "play" state.

Execution of a "63" instruction with M(R(X))=01 will set a "tape run" switch. It will also reset a "speaker on" switch. Tape will run but no sound will be heard.

Execution of a "63" instruction with M(R(X))=03 will set the "tape run" switch and also set the "speaker on" switch. Tape will run with sound on.

In either of the above cases the "tape run" switch will be automatically reset at the end of any 4.2kc stop tone encountered on tape (whether heard or not).

The "tape run" switch is equivalent to EF2. If EF2=1 the "tape run" switch is set. If EF2=0 "tape run" is reset. A "35" instruction can be used to test whether or not the tape is running.

A "63" instruction with M(R(X))=00 will reset the "tape run" switch and the "speaker on" switch.

A "63" instruction with M(R(X))=02 will reset the "tape run" switch and set the "speaker on" switch.

These four instructions provide flexible cassette control. The "CLUE" program in Section V illustrates the use of these instructions. Note that pressing the control panel "RESET" switch always sets both the "tape run" and "speaker on" switches. "RESET" must be pressed to permit manual control of the external cassette player.

E. CASSETTE READ/WRITE INSTRUCTIONS

Cassette data can be read in two modes (program or direct). The cassette player is first "selected" by a "61" instruction with M(R(X))=03.

Subsequent execution of a "62" instruction with M(R(X))=Ø4 "activates" the cassette player in the "program" mode. Activation also sets the "tape run" and "speaker on" switches. (The tape run switch can be subsequently reset by either a stop tone on tape or execution of an appropriate "63" instruction.)

In the "program" mode bits read from tape are accumulated into 8 bit bytes. When a complete byte is ready to be stored in memory, EF1 is set. The program must test EF1 and store input bytes via a "68" instruction as required. Operation is identical to that for the card reader "program" mode.

"Selecting" the cassette and placing it in the "direct" mode sets the "tape run" switch and causes tape bytes to be stored in memory via cycle stealing. This mode of operation is identical to that described for the card reader. (The tape run switch can be subsequantly reset by either a stop tone on the tape or execution of an appropriate "63" instruction.) Once "selected" the cassette player is activated in the "direct" mode by execution of a "62" instruction with M(R(X))=(Ø8).

Both of the above modes of operation assume that the cassette player be manually preset to its "play" state.

A third mode of tape recorder operation is permitted in the experimental FRED prototype. This "write" mode is activated by executing a "62" instruction with M(R(X))=4Ø. In this mode sequential memory bytes are recorded on an audio tape in the form shown in Figure 8. Bytes are addressed by R(Ø) and transmitted to the tape interface circuits automatically via the computer cycle stealing (direct) mode. A procedure for recording data on tape will be described with the "write" program in Section V.

F. MISCELLANEOUS AND SIMULTANEOUS I/O

An idle instruction is provided which permits display of a memory byte in the 8 bus bit lights on the control panel. Execution of an "ØN" instruction will cause suspension of subsequent instruction execution until a program interrupt or direct memory channel access occurs. It should be noted that if the card reader, TV, or cassette has been activated in the "direct" mode the instruction following an idle will be executed after the next direct I/O memory cycle or program interrupt. During suspension of instruction execution M(R(N)) will appear in the 8 bit lights on the control panel.

A "62" instruction with M(R(X))=ØØ will turn-off (or deactivate) any "selected" I/O device. Since only one device at a time should be operating in the direct mode this instruction permits turning off one direct device prior to activating another.

FRED permits a variety of simultaneously running I/O device combinations. One of the most useful combinations comprises TV display + card input + tape voice. This could be achieved as follows:

1. Select and activate TV (direct mode)
2. Select and activate card reader (program mode)
3. Set "tape run" switch (reset by stop tone on tape).

In this manner the cassette palyer could be used to ask a student a series of questions separated by stop tones. While the tape is running the card reader can be monitored by program for a proper response. Simultaneously the TV display can be indicating the correctness of student responses to the questions.

An error flag (EF4) has been provided which can be tested by a "JF" instruction. EF4 is set by a card reader/cassette read parity error. It is reset by turning off (deactivating) the error device.

The above describes all minimum FRED instructions. A variety of new devices and instructions could be added to this basic system. Discussion of these possible enhancements is not within the scope of this report.

SECTION III - OPERATING PROCEDURES

A. NORMAL OPERATION

These "normal" operating procedures utilize the control panel shown in Figure 1Ø. This would comprise the only operating panel available on a final product. The switches and lights provided are described below:

BIT LIGHTS - 8 bit lights (Ø-7) monitor the 8 bit computer data bus.

STOP LIGHT - Indicates that the basic computer clock is stopped.

READY LIGHT - Indicates that the computer is in an idle state.

ERROR LIGHT - Comes on with a cassette/card parity error. Generally signifies that program must be restarted and/or reloaded.

OFF/ON SWITCH- Turns power on/off.

RESET SWITCH - This momentary contact switch should be depressed to reset FRED. It turns off the clock, deactivates all I/O devices, sets register P=Ø, sets R(Ø)=ØØØØ, and puts the computer in the idle state. After reset, stop and ready lights will both be on. Error should be off.

READ SWITCH - Turning this switch on starts the clock and activates the cassette read direct mode.

d. Enter program cards in proper sequence via card reader. Program will be automatically loaded in memory starting at M(0000).

e. READ & CARD switches off

f. Press RESET

g. Press RUN. Program execution will begin at M(0001).

If during step "d" the error light comes on, repeat steps b, c, d. Note that after each card the bit lights will show the last card byte entered in memory.

3. PROGRAM RESTART

a. RESET

b. RUN

4. MEMORY READ

In certain applications a user is permitted to enter his own programs via cards he has punched himself. A sequence of memory bytes can be examined as follows:

a. RESET

b. Press RUN. Each subsequent depression of run will display the next memory byte in sequence via the bit lights starting at M(0000).

B. DEBUG PANEL

1. GENERAL

The debug panel provided on the experimental FRED prototype is shown in Figure 11. It facilitates prototype program debugging and development of new I/O attachments. Several of the more generally useful operations possible will be described. The detailed logic in Section IV should be

examined to determine feasibility of any desired manual operation not discussed here.

The following lists all debug switch functions provided:

0-7 SWITCHES - Set corresponding computer bus bit to "1" when up.

R SELECT - Select R(Ø-F)

RØ SWITCH - Select RØ

R1 SWITCH - Select R1

WR SWITCH - Write bus byte to selected R byte

WM SWITCH - Write bus byte to M location specified by selected R.

WP SWITCH - Write bus byte to P.

WIN SWITCH - Write bus byte to I&N

A1 SWITCH - Display R1 of selected register in bit lights on control panel

AØ SWITCH - Display RØ of selected register in bit lights.

M SWITCH - Display M byte addressed by selected register in bit lights.

D SWITCH - Display D register in bit lights

T SWITCH - Display T register in bit lights.

N SWITCH - Display N register in bit lights

PX SWITCH - Transfer P,X registers to T for subsequent display.

CM SWITCH - Used to clear memory

RPT SWITCH - Causes same machine cycle to be repeated

SP SWITCH - Stops clock. Left on it permits one step operation

ST SWITCH - Starts clock.

MANUAL SWITCH- Must be on to enable majority of debug panel functions. Must always be off during normal operation.

MASP SWITCH - When on computer is stopped if M location specified by ØØ-Ø1-1Ø switches is addressed.

2. SETTING REGISTERS

Any R register can be set to any value as follows:

a. RESET, MANUAL on

b. SELECT R via R select

c. Select R byte via RØ/R1

d. Set Ø-7 switches to desired bit pattern

e. Press WR to set bit pattern into selected register byte.

This procedure can be used to preset R(Ø) to any starting address prior to normal memory load/read operations.

3. SETTING MEMORY BYTE

Any memory byte can be set to any value as follows:

a. RESET, MANUAL on

b. Set R(Ø) to address of memory byte

c. Set Ø-7 switch to bit pattern

d. Press WM to store bit pattern in M.

INSTRUCTION SUMMARY

0	N	IDLE, M(R(N)) → LIGHTS
1	N	R(N)+1
2	N	R(N)-1
4	N	M(R(N)) → D,R(N)+1
5	N	D → M(R(N))
8	N	R0(N) → D
9	N	R1(N) → D
A	N	D → R0(N)
B	N	D → R1(N)
C	N	D0 → R00(N)
7	8	T → M(R(X))

D	N	N → P
E	N	N → X
F	0	M(R(X)) → D
F	1	M(R(X))/D → D
F	2	M(R(X))&D → D
F	3	M(R(X))+D → D
F	4	M(R(X)) PLUS D → D , FC → DF
F	5	M(R(X)) MINUS D → D, FC → DF
F	6	SHIFT D, 1 B R → DF
6	8	INPUT BYTE → M(R(X))
7	0	M(R(X)) → XP,R(X)+1 RESET IM

3	0	Y	Y → R0(P) UNCONDITIONAL BRANCH
3	1	Y	Y → R0(P) IF D≠0
3	2	Y	Y → R0(P) IF D=0
3	3	Y	Y → R0(P) IF DF=1
3	4	Y	Y → R0(P) IF EF1=1 (INPUT BYTE READY)
3	5	Y	Y → R0(P) IF EF2=1 (TAPE ON)
3	6	Y	Y → R0(P) IF EF3=1
3	7	Y	Y → R0(P) IF EF4=1 (ERROR)

6	2	& M(R(X))=	0 0	TURN SELECTED I/O OFF, R(X)+1

6	1	& M(R(X))=	0 1	SELECT INPUT (CARD/SWITCH), R(X)+1
6	2	& M(R(X))=	0 1	SET SELECT INPUT TO PROGRAM MODE,R(X)+1
6	2	& M(R(X))=	0 2	SET SELECT INPUT TO DIRECT MODE ,R(X)+1

6	1	& M(R(X))=	0 3	SELECT TAPE I/O, R(X)+1
6	2	& M(R(X))=	0 4	SET TAPE IN TO PROGRAM MODE,R(X)+1
6	2	& M(R(X))=	0 8	SET TAPE IN TO DIRECT MODE ,R(X)+1
6	2	& M(R(X))=	4 0	SET TAPE OUT TO WRITE MODE ,R(X)+1

6	3	& M(R(X))=	0 0	TAPE OFF & SPEAKER OFF,R(X)+1
6	3	& M(R(X))=	0 1	TAPE ON & SPEAKER OFF,R(X)+1
6	3	& M(R(X))=	0 2	TAPE OFF & SPEAKER ON ,R(X)+1
6	3	& M(R(X))=	0 3	TAPE ON & SPEAKER ON ,R(X)+1

6	1	& M(R(X))=	0 2	SELECT TV OUT ,R(X)+1
6	2	& M(R(X))=	0 1	SET TV TO 32x32 MODE,R(X)+1
6	2	& M(R(X))=	0 2	SET TV TO 16x64 MODE,R(X)+1

RESET SWITCH: 0000 → R(0),IDLE,ALL I/O OFF,0 → P,CLOCK OFF

RUN: BEGINS EXECUTION WITH INSTRUCTION @ M(R(1))

FIGURE-9

7-15-72

JAW

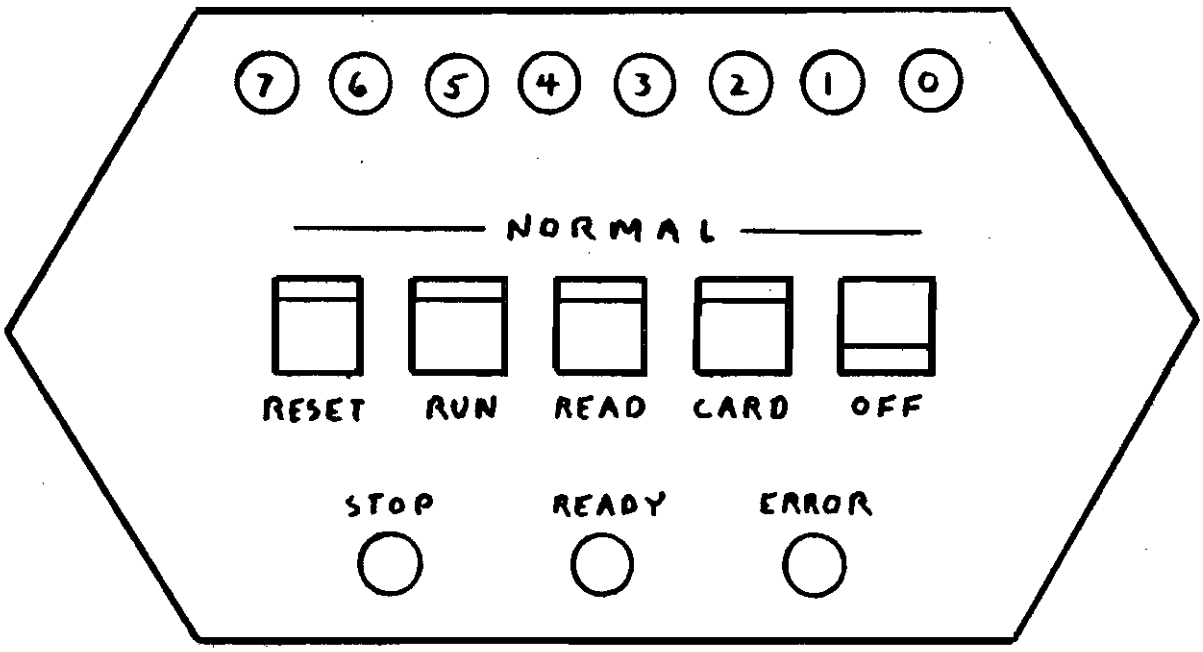


FIGURE 10

CONTROL PANEL

(S2)

JAW

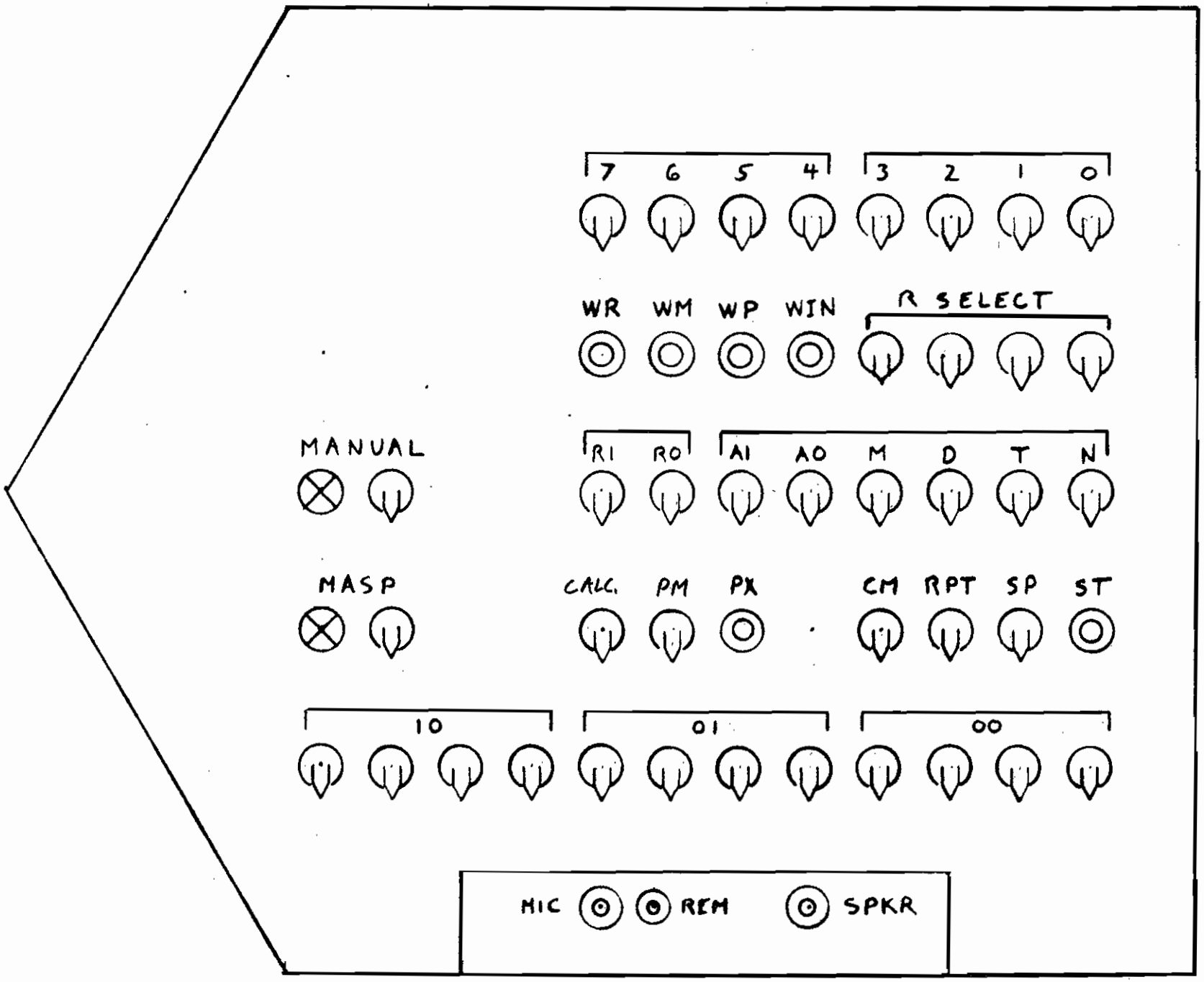


FIGURE 11- DEBUG PANEL (S1) JAW 3-1-72

	M1	P1	P2	P3	E1	E2	E3
7400 - 4x2 NAND	2	13	3	6	2	5	5
7401 - 4x2 NAND (o.c.)	2		14	1	3	1	4
7402 - 4x2 NOR			5		2		1
7404 - HEX INVERT		6	2	6	1	1	
7405 - HEX INVERT (o.c.)				1			
7408 - 4x2 AND		1	2			1	
7410 - 3x3 NAND		4	2	1	4		1
7420 - 2x4 NAND		2	1		1	1	1
7427 - 3x3 NOR		1					
7430 - 1x8 NAND		1					
7474 - 2 D FF		1	1	1	2	4	5
7475 - 4 LATCH		1	7				
7485 - 4 BIT COMPARE	3						
7486 - 4 x EXCL. OR							
7489 - 4 x 16 RAM			4			2	
7493 - 4 BIT CTR.		1				4	
7495 - 4 BIT L/R SHIFT			2		1		
7496 - 5 BIT REG.					1		2
74121 - ONE SHOT				1	3	5	2
74151 - 8 BIT SELECT		1				1	
74154 - 4 → 16 DECODE		1					
74155 - 3 → 8 DECODE	1	1	1	1	1		
74164 - 8 BIT SHIFT REG.							1
74175 - QUAD D FF		2	2	1			
74180 - PARITY CHECK					1		2
74181 - 4 BIT ALU			2				
74192 - DEC. CTR.							2
74193 - 4 BIT +/- CTR			4				
1101/2501 - 256 BIT RAM	32						
567 - TONE DETECTOR					3	1	
	40	36	52	19	25	26	26

→ 224

TABLE I

JAW

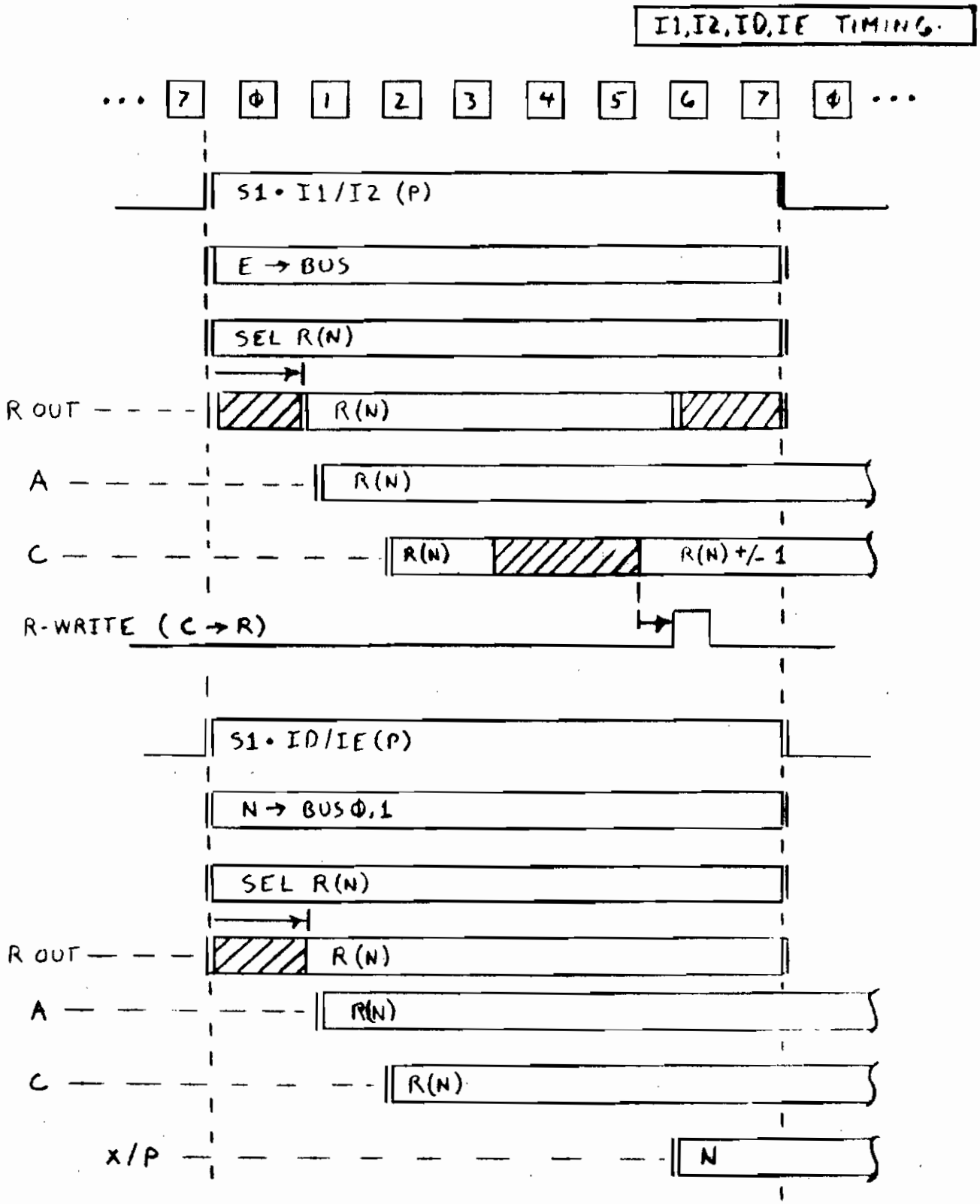


FIGURE 12

JAW

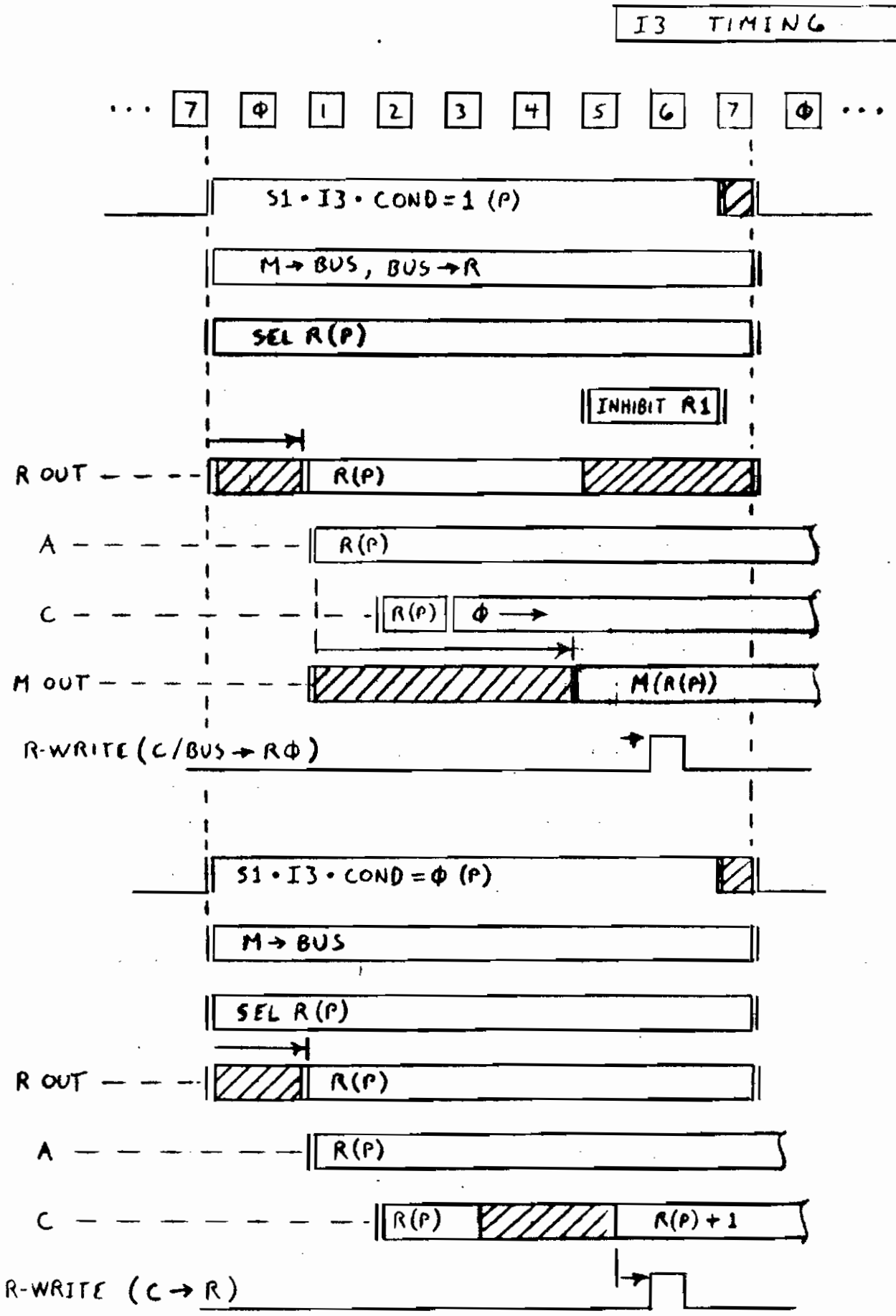


FIGURE 13

TAW

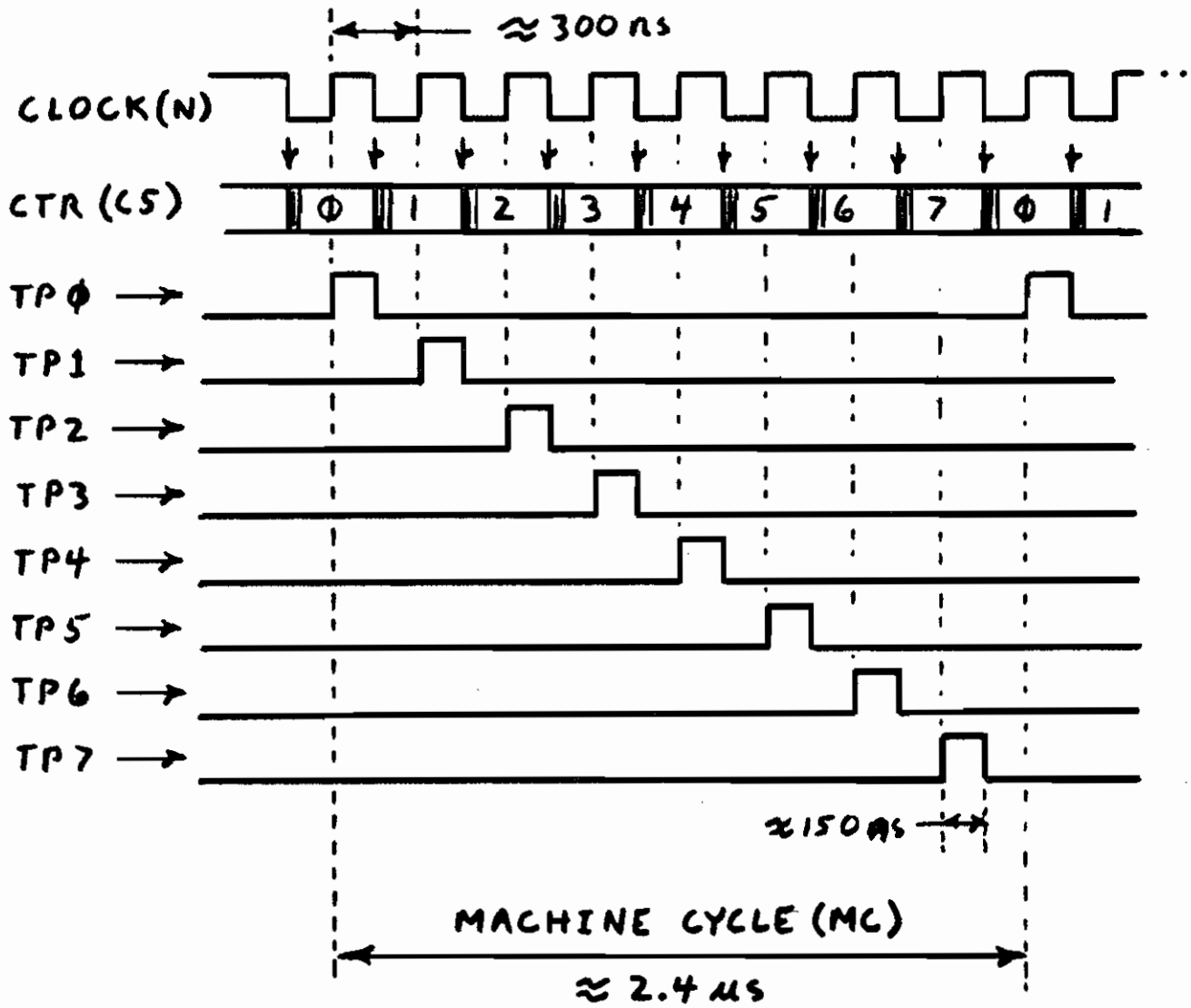


FIGURE 14

BASIC COMPUTER TIMING

JAW

SECTION IV - DETAILED LOGIC

Logic-1 through logic-25 shows the detailed logic design of FRED. Table I shows the number of chips used in the design. Three power supplies are provided and all circuits were mounted on cards as shown in Figure 2. Figures 12 and 13 provide sample timing for selected logic.

In general, the logic circuits on one page are all contained on the same printed circuit card. For example logic-10 circuits will all be found on the "P1" plug-in card.

Logic symbols contain both a chip location on the printed circuit card (A3, B5, etc.) and a chip type (7400, 74121, etc.) refer to manufacturers literature for details of each chip type. Multiple wire buses are indicated by slash marks.

Logic 1 through 4 illustrates the control panel and debug panel wiring.

Logic-7 shows the memory. It comprises 32-256 bit fully decoded, static RAM chips. Outputs, inputs, and addresses, are connected directly to form a 1024 byte memory bank. The least significant 8 bits of an address select a byte location within a chip. The next two bits are decoded to select one of 4 groups of 8 chips.

Operation of the memory merely comprises applying address levels. After the access time, the addressed byte appears as DC levels on the memory output lines. Read is non destructive.

Writing is performed by applying address and write data followed by a write pulse (MWR).

Logic-5 shows the logic associated with various control switches used for starting, stopping, and resetting the computer. The master clock oscillator is also shown in logic-5.

Logic-6 shows circuits for isolating external I/O buses from the computer data bus. A channel select register (CSR38Ø) and external "N" digit decoder (C4) are also shown. These circuits are required for I/O device selection by the "61" instruction.

Logic-8 through 14 shows the main computer logic contained on P1&P2 cards.

Logic-8 shows the time pulse generator which controls timing of all operations. 8 time pulses are provided as shown by the timing diagram of Figure 14. Eight 150 time pulses separated by 150 yield a nominal machine cycle of 2.4μs. In order to maintain the required TV display refresh cycle the time pulse oscillator should not be adjusted to cause machine cycles to exceed 3μs.

A machine cycle comprises 8 time pulses. There are four different types of machine cycles. These four types are defined by the two bit register "S" shown in logic-11 (C4). The possible states of this register activate 4 lines SØ, S1, S2, and S3. These levels define one of four types of machine cycles at any given time.

In normal operation only SØ and S1 types of machine cycles occur. SØ defines an instruction fetch machine cycle. SØ is combined with time pulses to cause a new instruction to be fetched from memory and placed in the I and N registers. S1 follows SØ and activates the instruction decoder (logic-11). One of 16 instruction lines (IØ-IF) is activated depending on the instruction code in I. This line is combined with the time pulses to cause the specified instruction function to be executed.

Normally the sequence SØ-S1-SØ-S1... is repeated causing alternating instruction fetch and execute cycles. Circuits shown in logic-10 and 11 permit this SØ-S1 sequence to be modified. External in/out request lines

are provided for direct memory access by an external device. Activation of either line causes an S2 machine cycle to be inserted as follows. SØ-S1-SØ-S1-S2-SØ...

For an input request, S2 causes R(Ø) to address the memory and stores an input byte at the addressed location. R(Ø) is incremented by 1 so that a sequence of input bytes will be stored in sequential memory locations. For an output request, S2 fetches the byte at M(R(Ø)) and places it on the output bus. R(Ø) is again incremented by 1.

A program interrupt line is also provided (logic-10). Activating this line causes an S3 cycle to occur after the next S1 cycle. S3 causes X and P to be placed in T. P is then set to 1 and X to 2. Resumption of the normal SØ-S1 sequence will then result in execution of an interrupt routine specified by R(1).

In general, the logic circuits contained on "P1" provide the basic timing, instruction decoding, and instruction execution control circuits (logic-8 to 11). "P2" contains the registers and arithmetic/logic circuits as shown in logic 12, 13, and 14.

Logic-15 and 16 shows the card interface circuits contained on E1. Two FF's (B1) are set to specify "direct/program" card input mode. A hex card digit (any hole punched) detected by the card reader photodiodes activate gates A4 and B4 (logic-16). The 5-bit hex digit is set into a 5 bit register (HR @ B3). Parity is checked at this point. A second hex digit moves the first digit (4 bits only) down to register "B2". The second digit then enters "B3" and is parity checked. A counter (C2) keeps track of which digit has just been read. Reading the second digit causes activation of either the "EF1" or "IN request" bus lines to notify the computer that an input card byte is ready.

The card reader photodiode circuits are shown in logic-24. These plug in to the interface circuits via jack J/K.

The optional hex switch panel (logic-25) can also be plugged into J/K. Local hex switch logic simulates a card digit for each switch depression.

Card "E1" also contains the 3 tone detectors/generators required for cassette read/write operations (logic-17).

Logic 18 shows the cassette control circuits and connecting jack wiring. The two FF controlled relays control the cassette motor and a local computer speaker.

Tape read circuits are shown in logic-19. The "Ø" and "1" phase locked loop tone detector outputs (logic-17) are converted to a parallel 8-bit byte via a shift register (BØ in logic-19). The "EF1" or "IN request" bus lines are activated to notify the computer that a tape byte is ready for transfer to memory. Parity checking circuits for tape data are also provided.

Logic-20 shows the tape write circuits. A 100 cycle/sec oscillator (B5) provides the recording bit rate timing. The "Ø" and "1" phase locked loop tone detectors also provide "Ø" and "1" tone oscillators for bit recording. The "OUT request" bus line is activated by the write logic for each memory byte required. The memory byte is placed in shift register "A2,A3" (logic-20) and has start and parity bits added. The resulting 1Ø bits are then recorded as the appropriate serial sequence of tones.

Logic-21, 22, 23 show the interface circuits required for the TV display. A horizontal sync oscillator running at 60x256 cycles per second is provided. (Logic-21, E1). This is divided by 256 by the "H" counter to

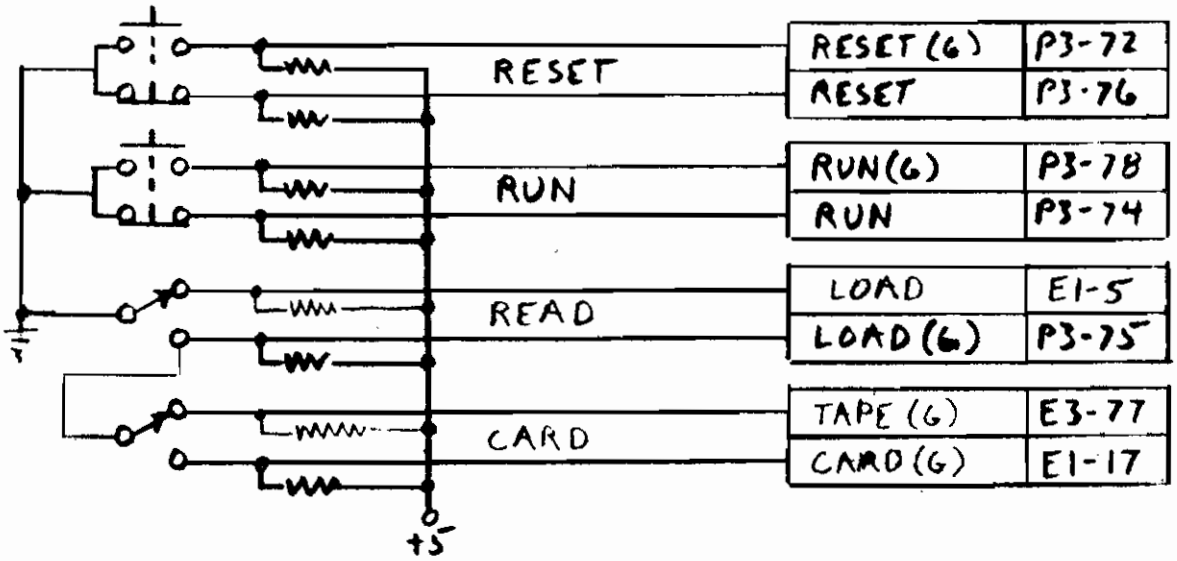
generate vertical sync pulses at a rate of 60/second, the "H" counter outputs are used to indicate the current display line.

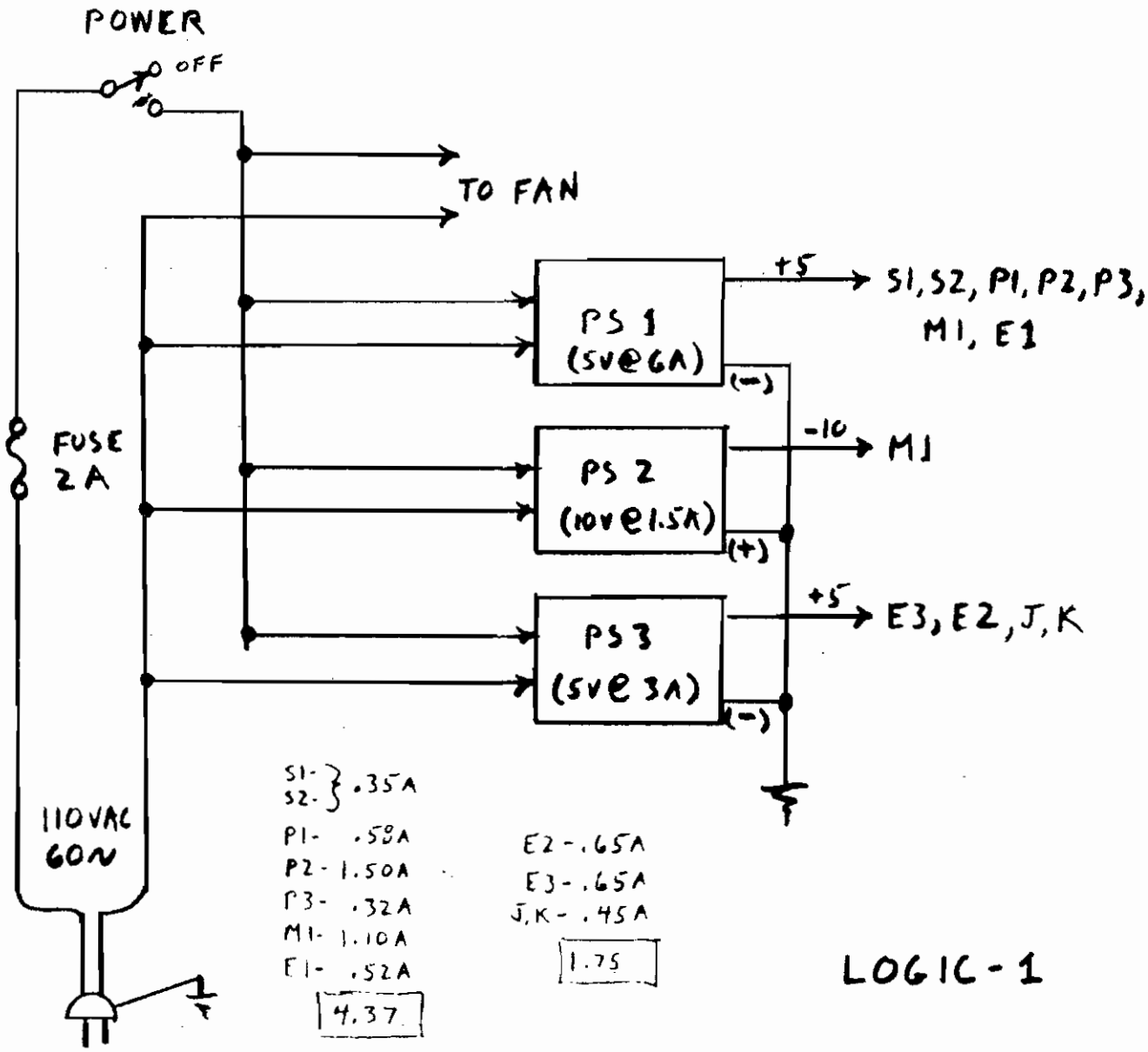
Each bit to be displayed uses four horizontal line times. Each dot (or bit) is two lines high and is separated from the next dot by two line times. This technique permits better screen utilization for the low resolution FRED display of 1024 dots or bits.

Certain horizontal synch pulses start a spot oscillator (E4, D4 in logic-21). 32/64 spot pulses are generated depending on the preset TV display mode. The "S" counter output are used to control memory byte fetches at appropriate times.

Memory bytes are placed in a buffer (AØ, A1 in logic-22). "S" counter outputs, buffer bytes, and the spot pulses are combined to form a beam modulating "video" signal. This signal is combined with horizontal and vertical sync pulses. The combined signal then modulates the output of an RF generator to form the signal applied to the antenna terminals of any TV set (logic-23).

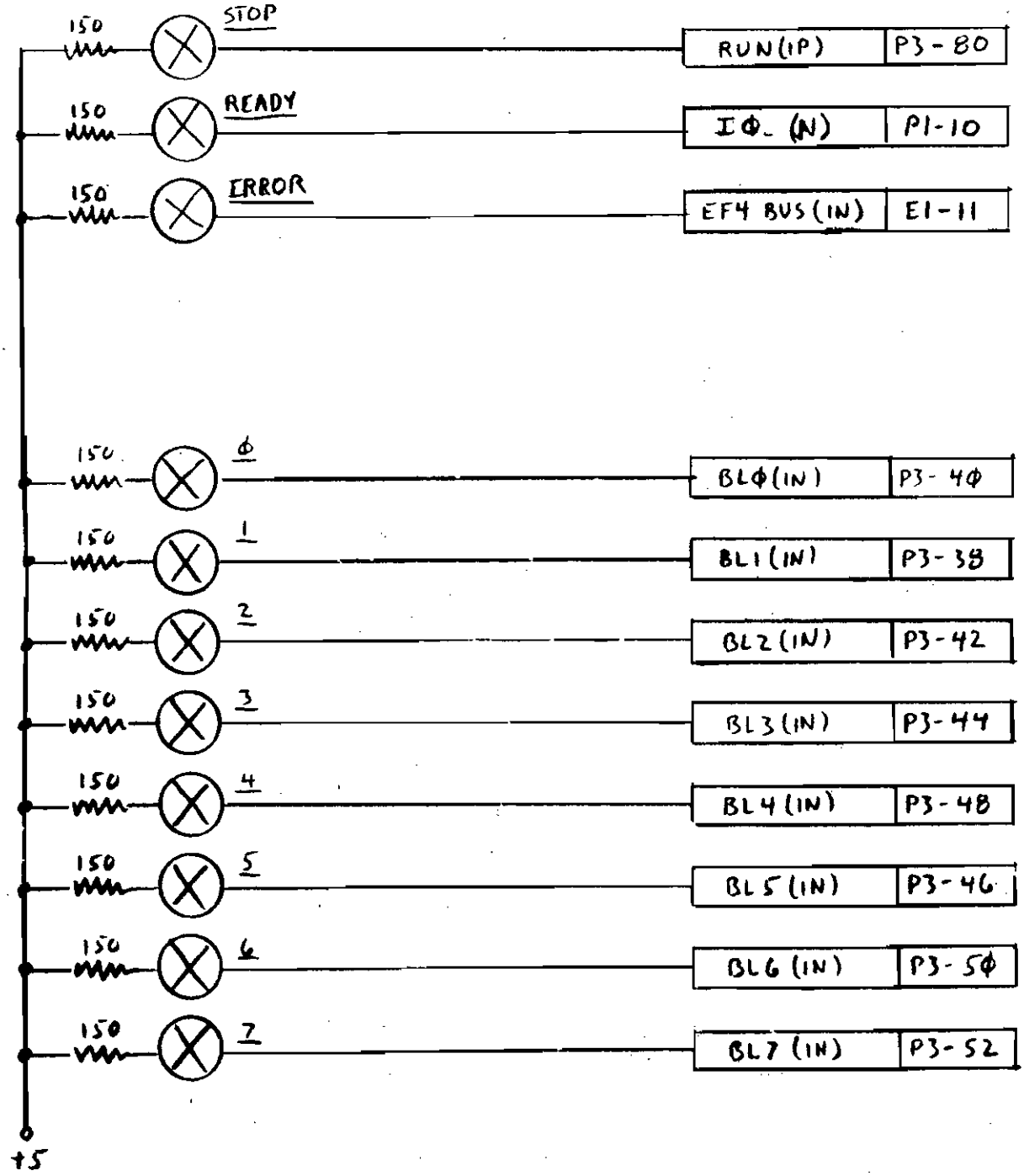
CONTROL PANEL & POWER





LOGIC - 1

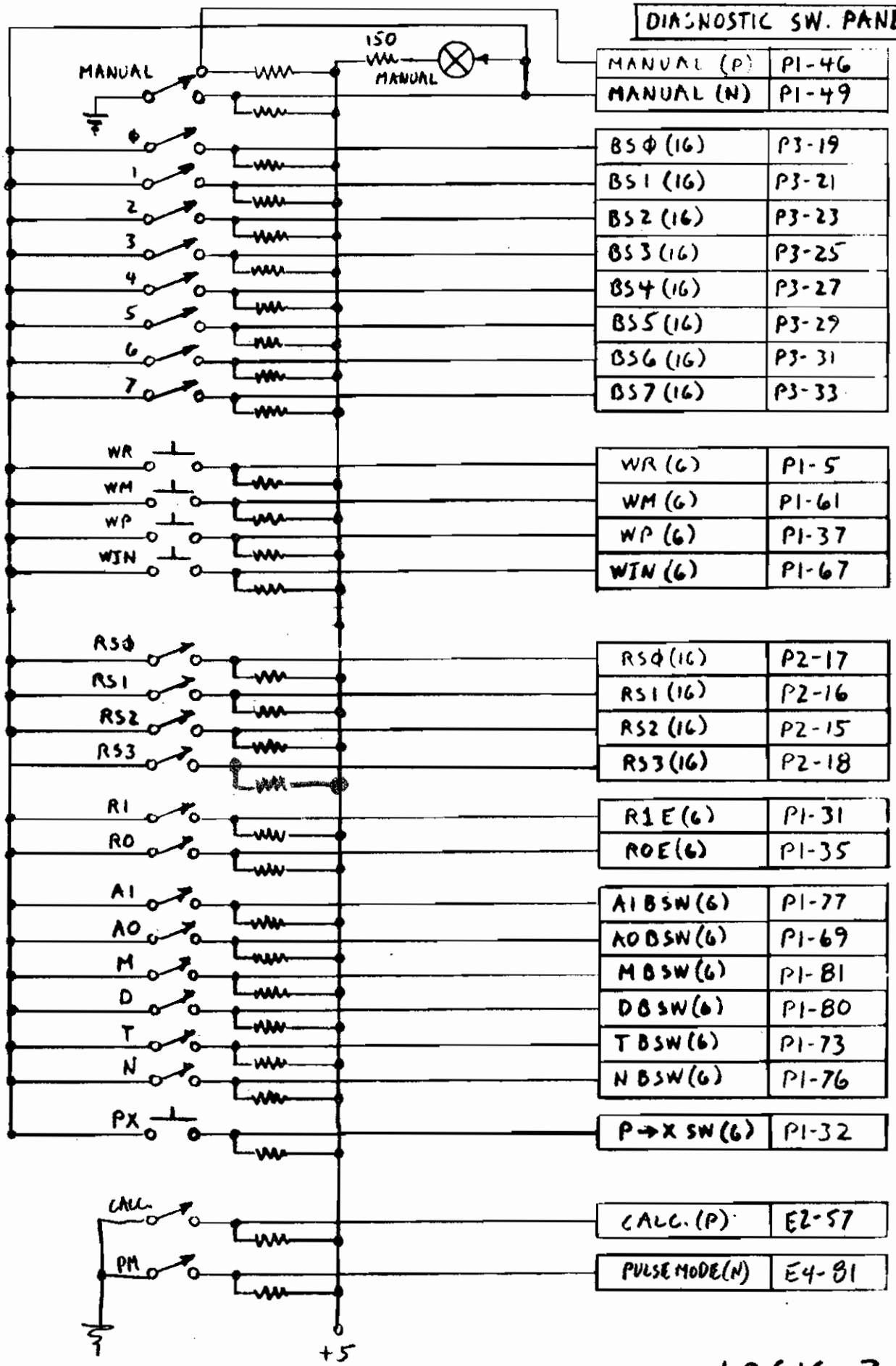
JAW 5-18-72



LIGHTS = 3V @ 15MA

LOGIC-2

DIAGNOSTIC SW. PANEL S1-A



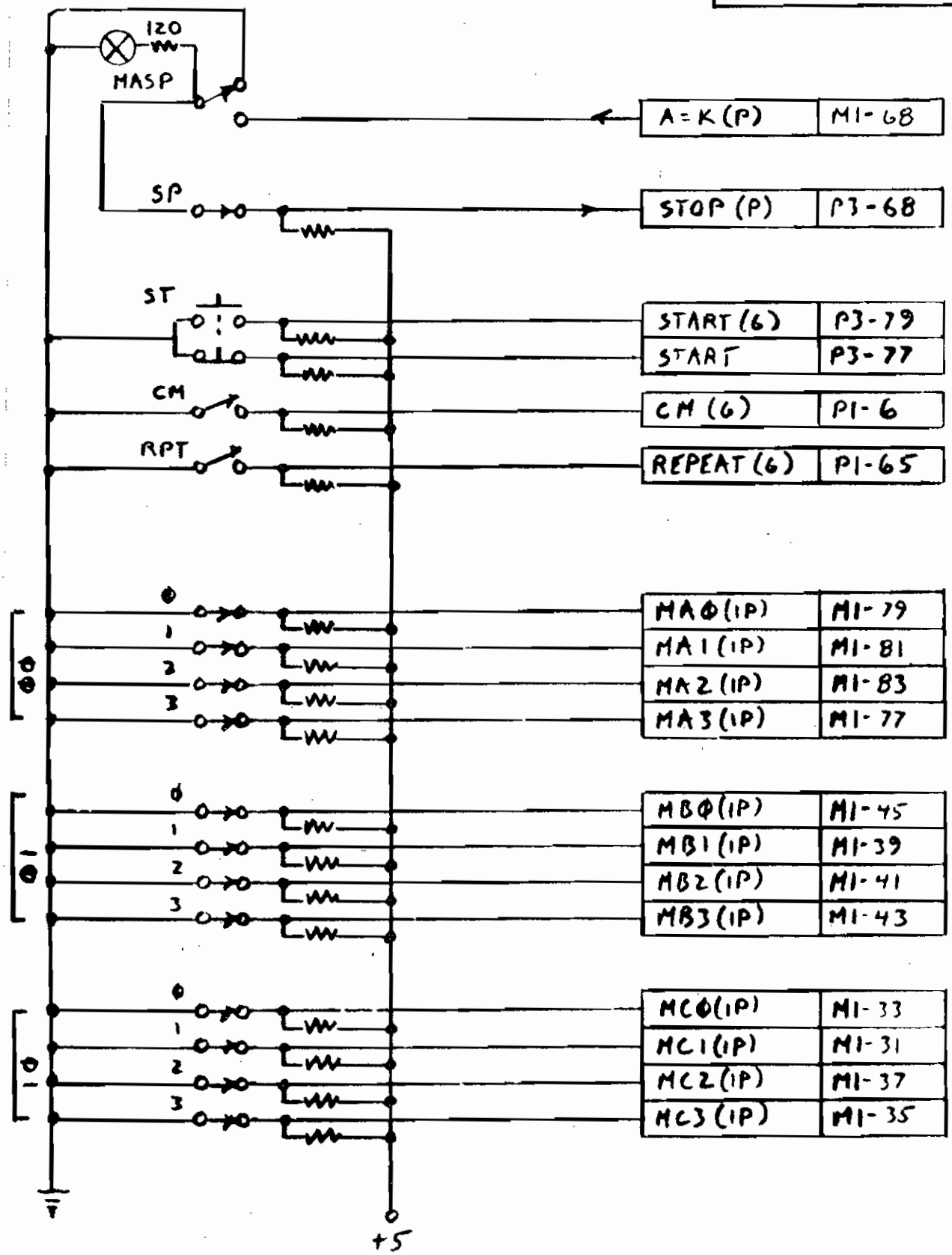
MANUAL (P)	P1-46
MANUAL (N)	P1-49
BSØ (16)	P3-19
BS1 (16)	P3-21
BS2 (16)	P3-23
BS3 (16)	P3-25
BS4 (16)	P3-27
BS5 (16)	P3-29
BS6 (16)	P3-31
BS7 (16)	P3-33
WR (6)	P1-5
WM (6)	P1-61
WP (6)	P1-37
WIN (6)	P1-67
RSØ (16)	P2-17
RS1 (16)	P2-16
RS2 (16)	P2-15
RS3 (16)	P2-18
RIE (6)	P1-31
ROE (6)	P1-35
AIBSW (6)	P1-77
AOBSW (6)	P1-69
MBSW (6)	P1-81
DBSW (6)	P1-80
TBSW (6)	P1-73
NBSW (6)	P1-76
P→X SW (6)	P1-32
CALC. (P)	E2-57
PULSE MODE (N)	E4-81

ALL R=1K UNLESS OTHERWISE SPECIFIED

LOGIC-3

JAW 3-1-72

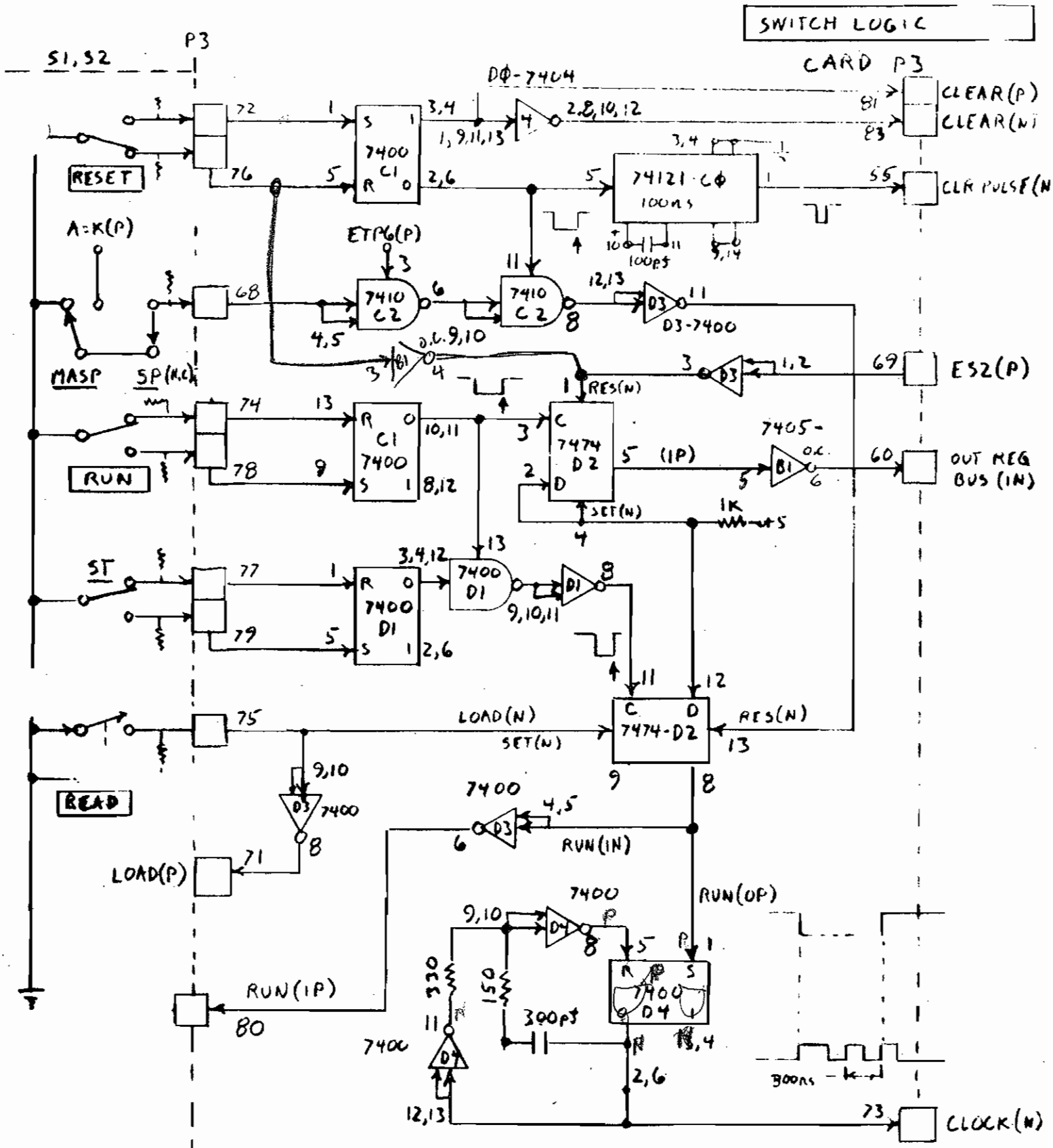
DIAGNOSTIC SW PANEL SI-8.



ALL R=1K
UNLESS OTHERWISE
SPECIFIED

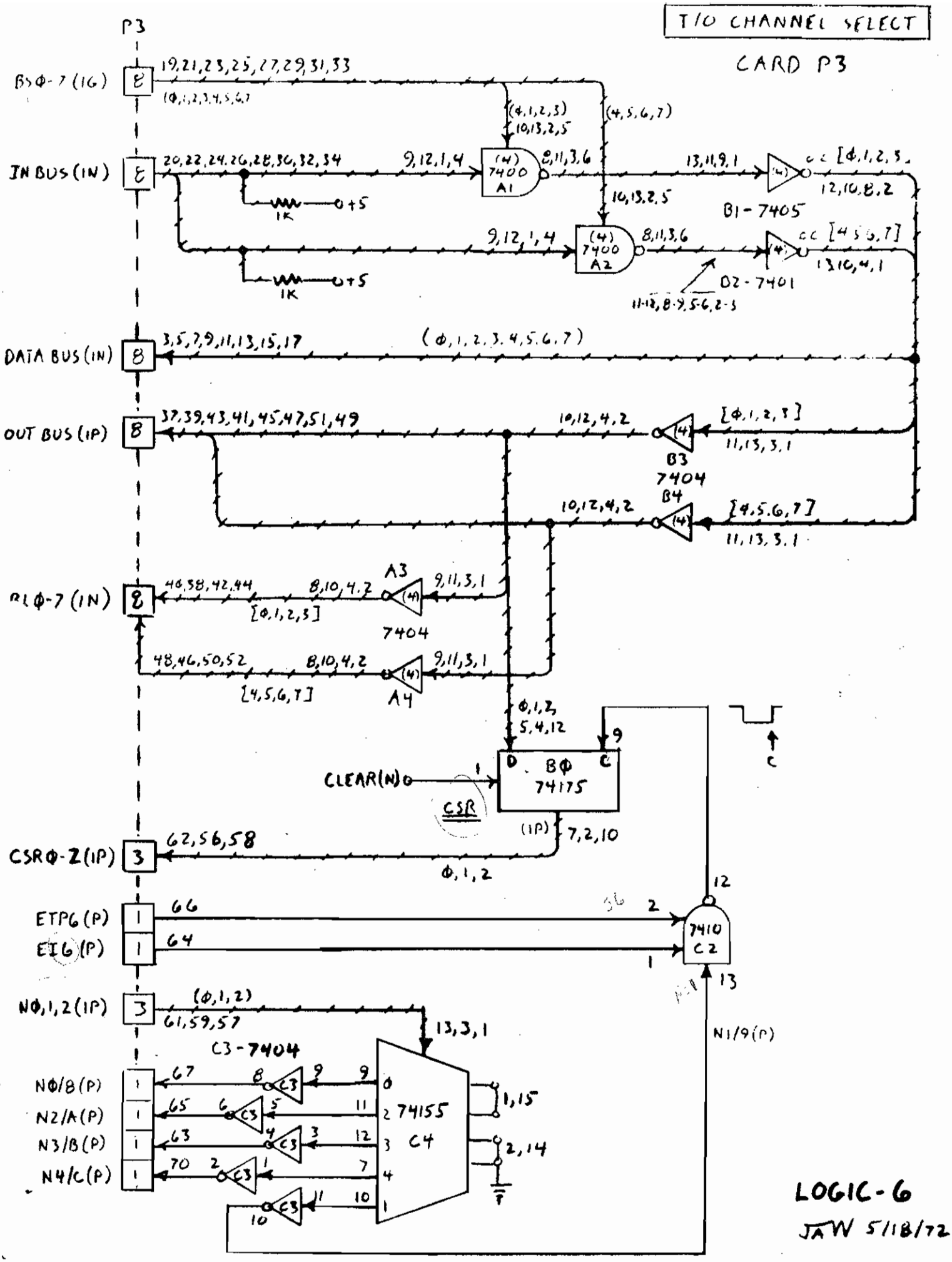
LOGIC-4

JAW 5-18-72.



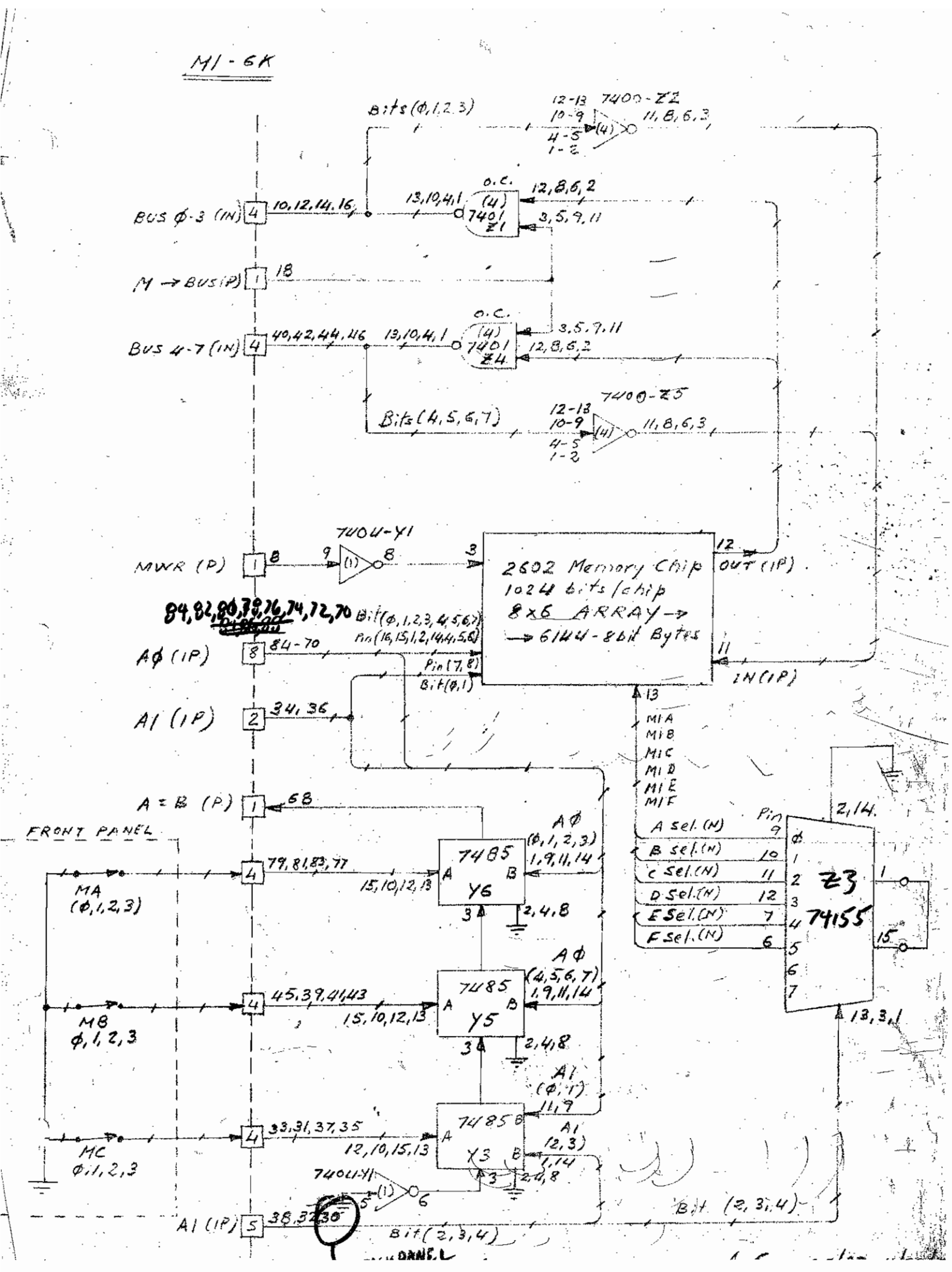
LOGIC-5

JAW 1/20/72

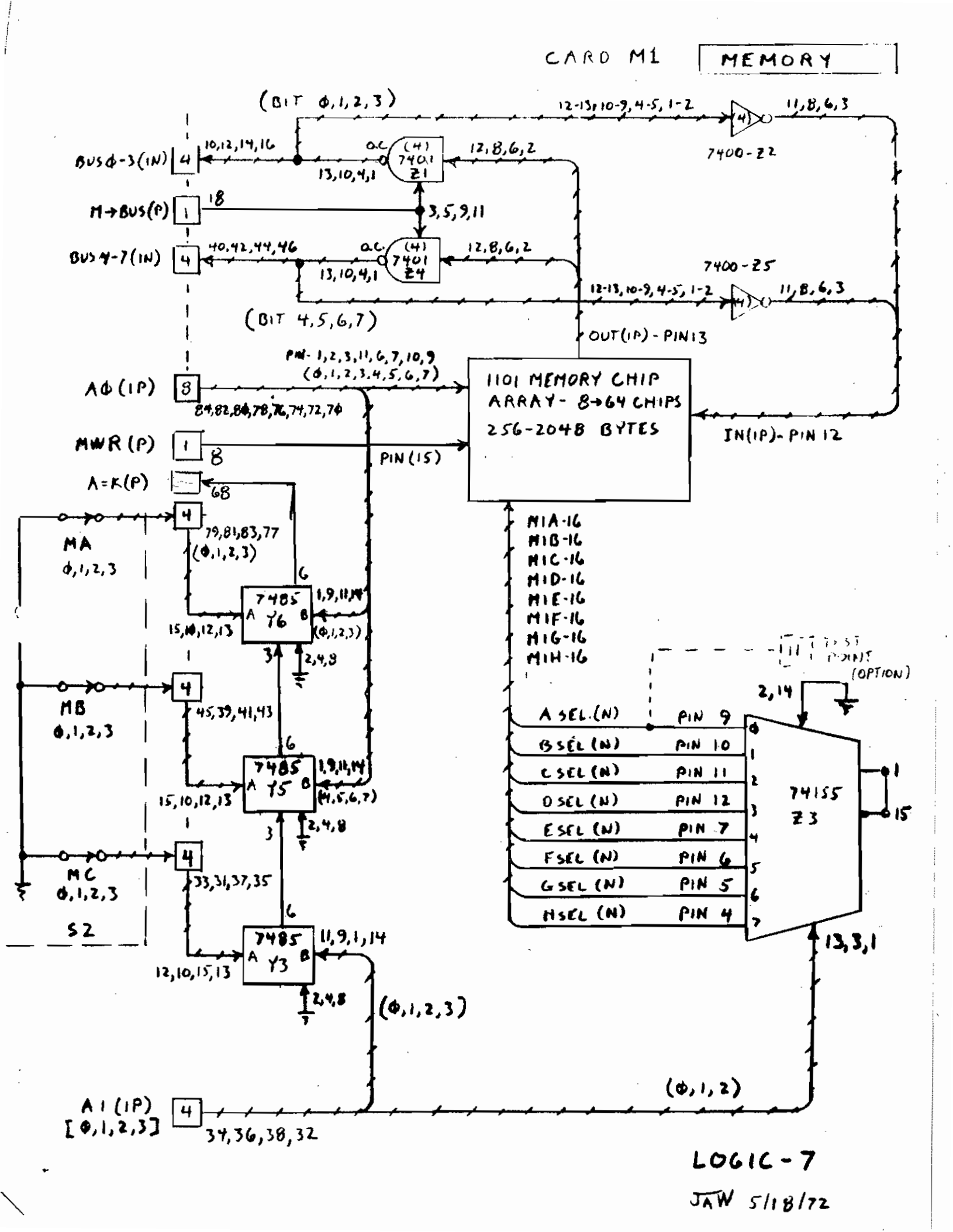


LOGIC-6

JAW 5/18/72

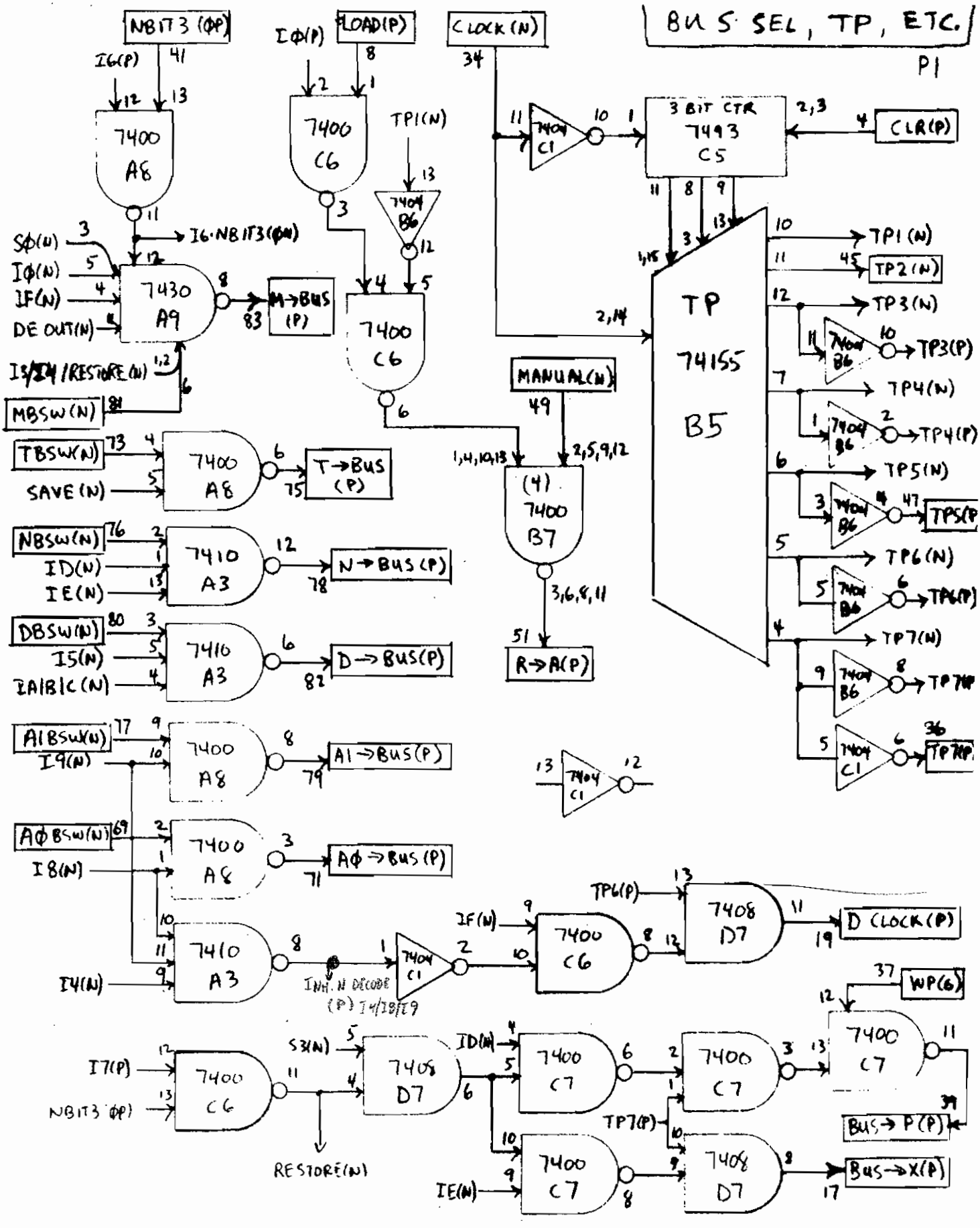

M1 - 6K
Bits (0,1,2,3)
12-13
10-9
4-5
1-2
7400-Z2
11,8,6,3
BUS 0-3 (IN)
10,12,14,16
13,10,4,1
o.c.
(4)
7401
Z1
12,8,6,2
3,5,9,11
M → BUS (IP)
18
BUS 4-7 (IN)
40,42,44,46
Z4
3,5,7,11
Bits (4,5,6,7)
7400-Z3
MWR (P)
7404-Y1
2602 Memory Chip
1024 bits/chip
8x6 ARRAY →
→ 6144 - 8bit Bytes
OUT (IP)
IN (IP)
Bit (0,1,2,3,4,5,6,7)
Pin (16,15,1,2,14,4,5,6)
Pin (7,8)
Bit (0,1)
A0 (IP)
84-70
A1 (IP)
34,36
A = B (P)
68
MIA
MIB
MIC
MID
MIE
MIF
A sel. (N)
B sel. (N)
C sel. (N)
D Sel. (N)
E Sel. (N)
F sel. (N)
Pin
Z3
74155
2,14
13,3,1
FRONT PANEL
MA
(0,1,2,3)
MB
0,1,2,3
MC
0,1,2,3
79,81,83,77
45,39,41,43
33,31,37,35
7485
Y6
Y5
Y3
A0
(0,1,2,3)
1,9,11,14
(4,5,6,7)
A1
(0,1)
11,9
(2,3)
1,14
15,10,12,13
12,10,15,13
2,4,8
7404-Y1
38,32,30
Bit (2,3,4)

CARD M1 | MEMORY



LOGIC-7

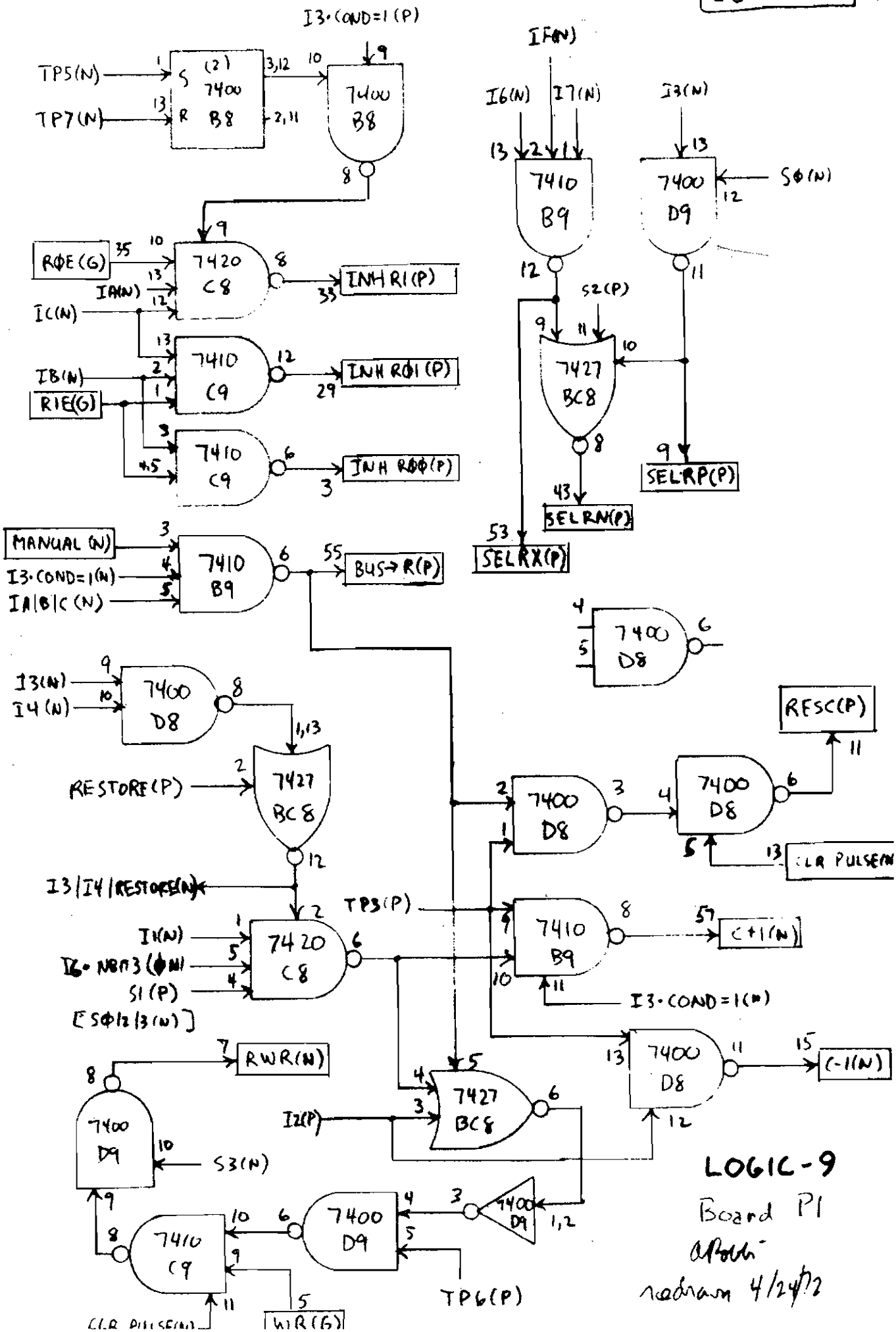
JAW 5/18/72



Board P1

LOGIC-8

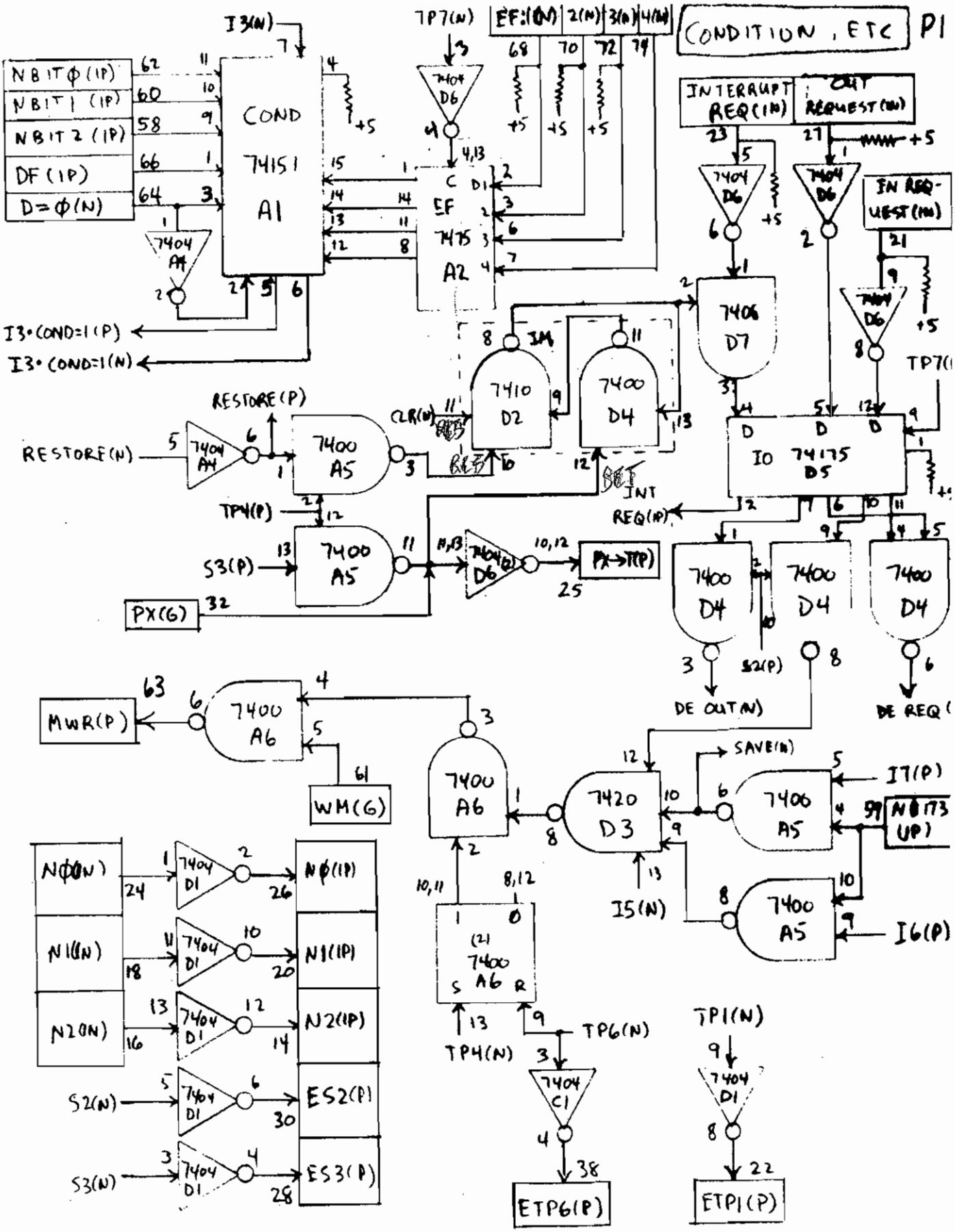
A Robb redrawn 4/19/72



LOGIC-9

Board P1

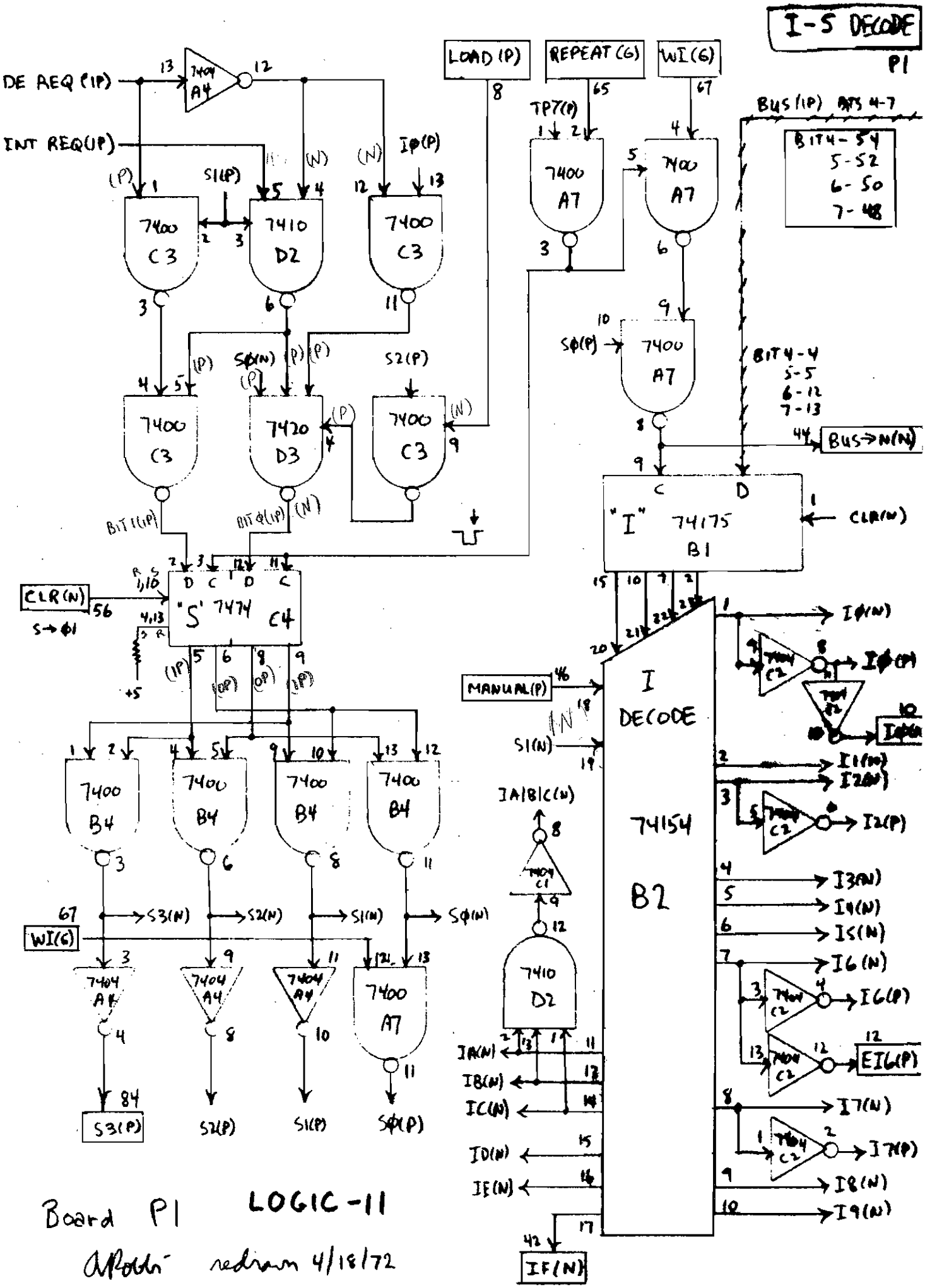
redraw 4/24/72

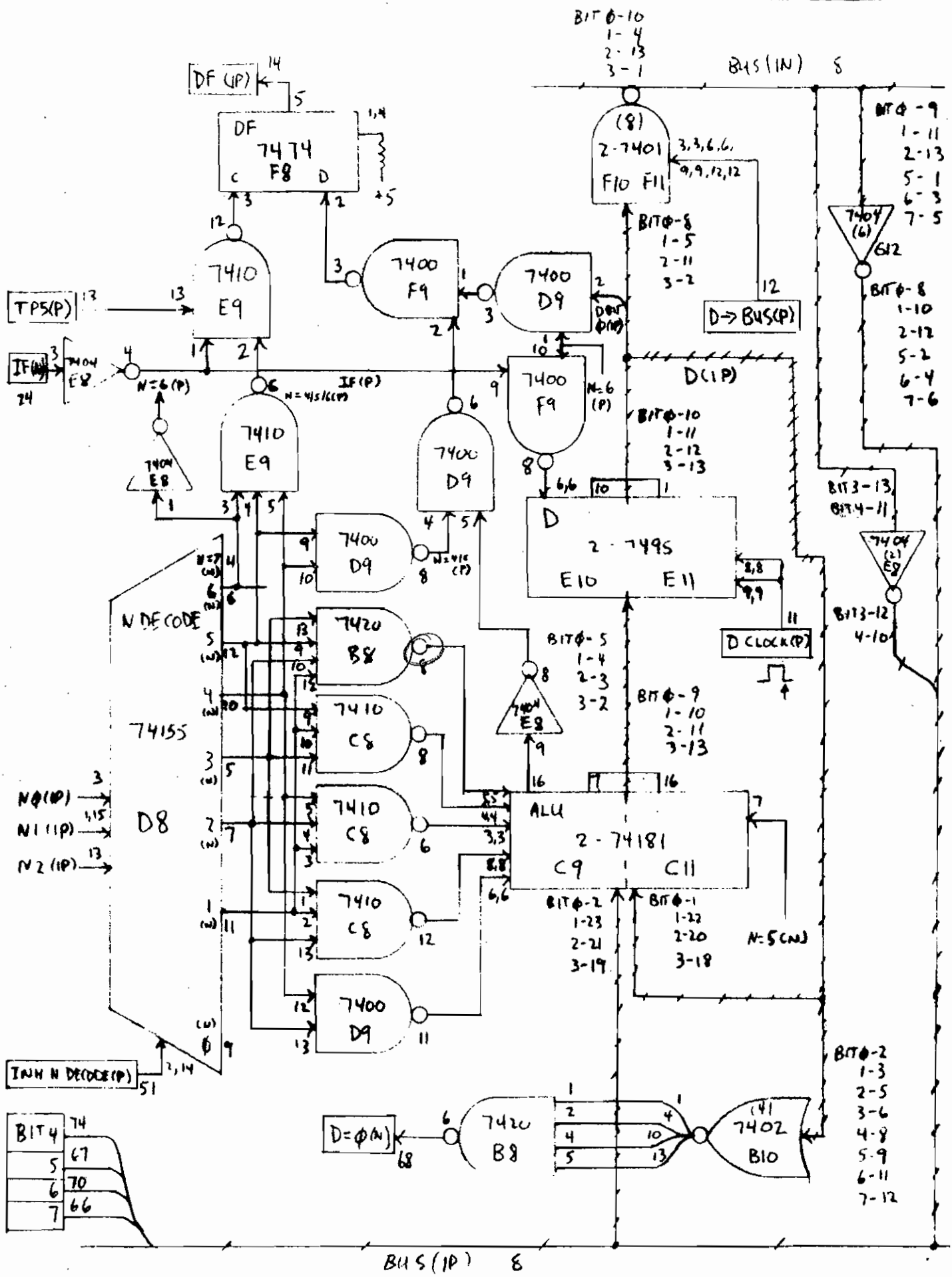


Board P1

LOGIC-10

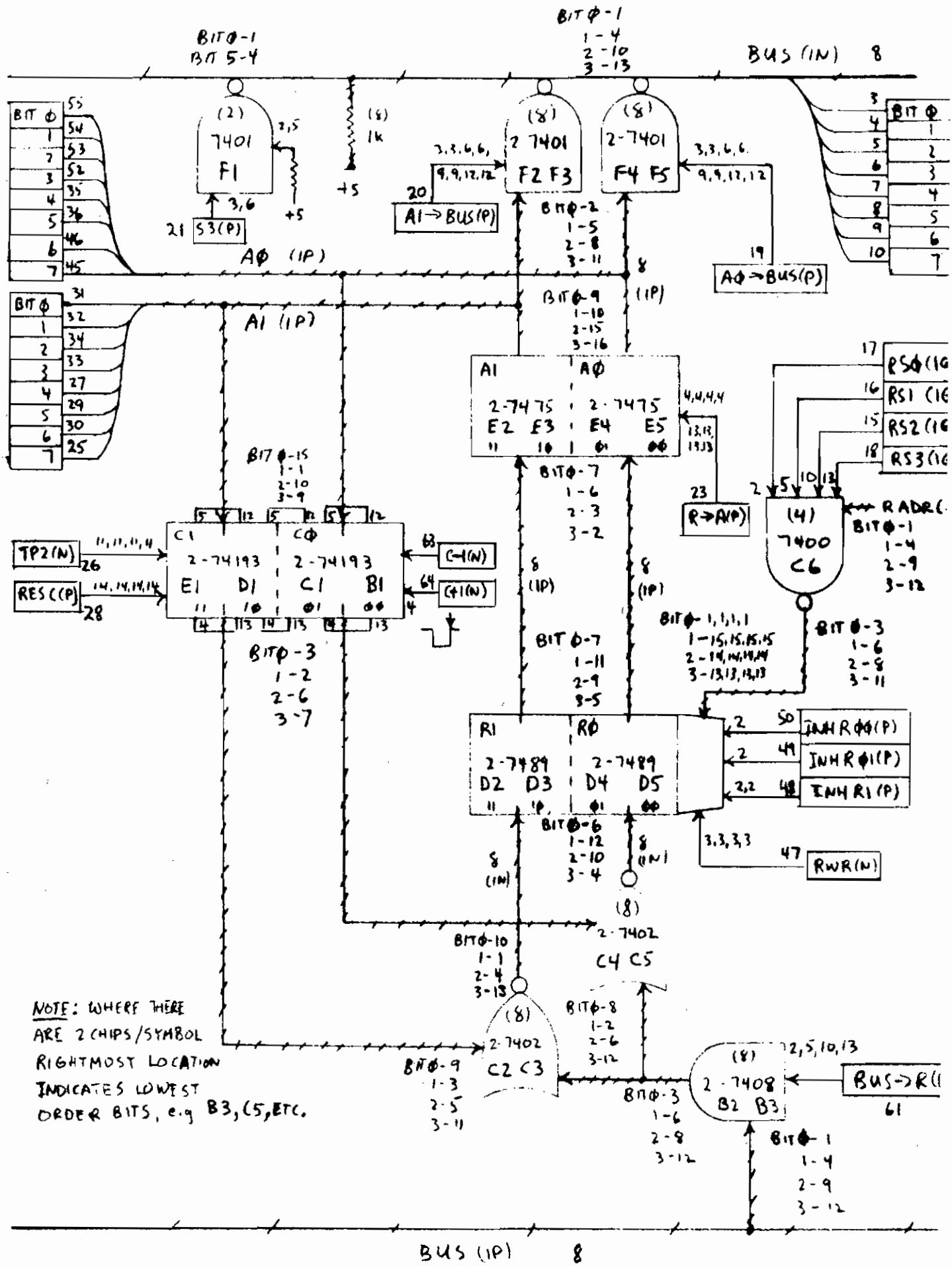
a Bochi rodrom 4/20/72


I-5 DECODE
P1
DE REQ (IP)
INT REQ(IP)
LOAD (P)
REPEAT (G)
WI(G)
TP7(P)
BUS (IP) BITS 4-7
BIT 4 - 5 4
5 - 5 2
6 - 5 0
7 - 4 8
7400 C3
7410 D2
7400 A7
7420 D3
BIT 4 - 4
5 - 5
6 - 12
7 - 13
BUS⇒N(N)
"I" 74175 B1
CLR(N)
CLR(N)
S→ΦI
"S" 7474 C4
+5
7400 B4
MANUAL(P)
S1(N)
IA|B|C(N)
I DECODE
74154
B2
I0(N)
I1(N)
I2(N)
I2(P)
I3(N)
I4(N)
I5(N)
I6(N)
I6(P)
EI6(P)
I7(N)
I7(P)
I8(N)
I9(N)
IA(N)
IB(N)
IC(N)
ID(N)
IE(N)
IF(N)
WI(G)
S3(N)
S2(N)
S1(N)
SΦ(N)
S3(P)
S2(P)
S1(P)
SΦ(P)
Board P1
LOGIC-11
redrawn 4/18/72



LOGIC-12

redrawn 4/21/72 ARobbi

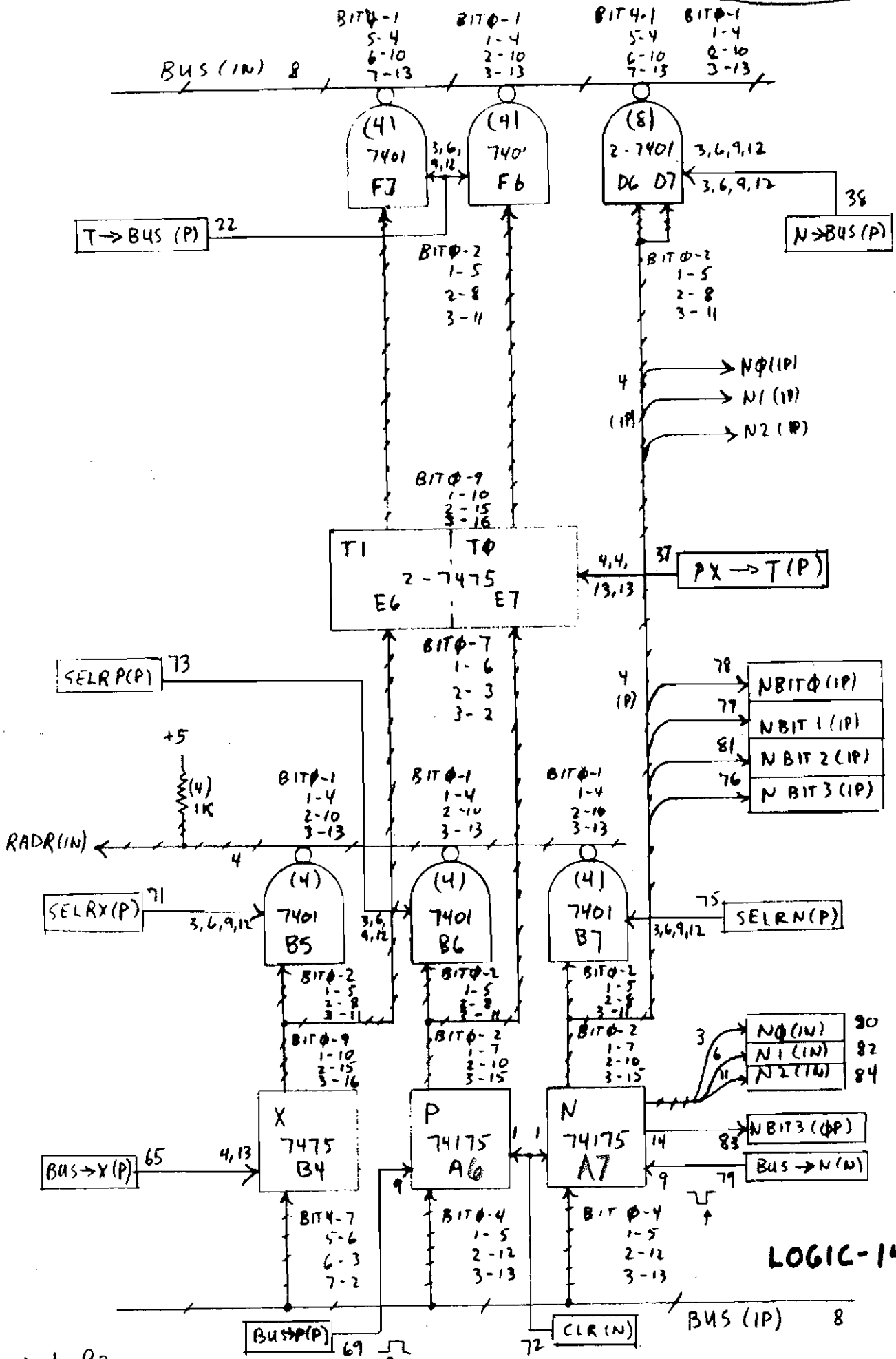


NOTE: WHERE THERE ARE 2 CHIPS/SYMBOL RIGHTMOST LOCATION INDICATES LOWEST ORDER BITS, e.g B3, C5, ETC.

P2 Board **LOGIC-13**

A Rann - nodraw 4/20/72

P-N-X-1 / P2



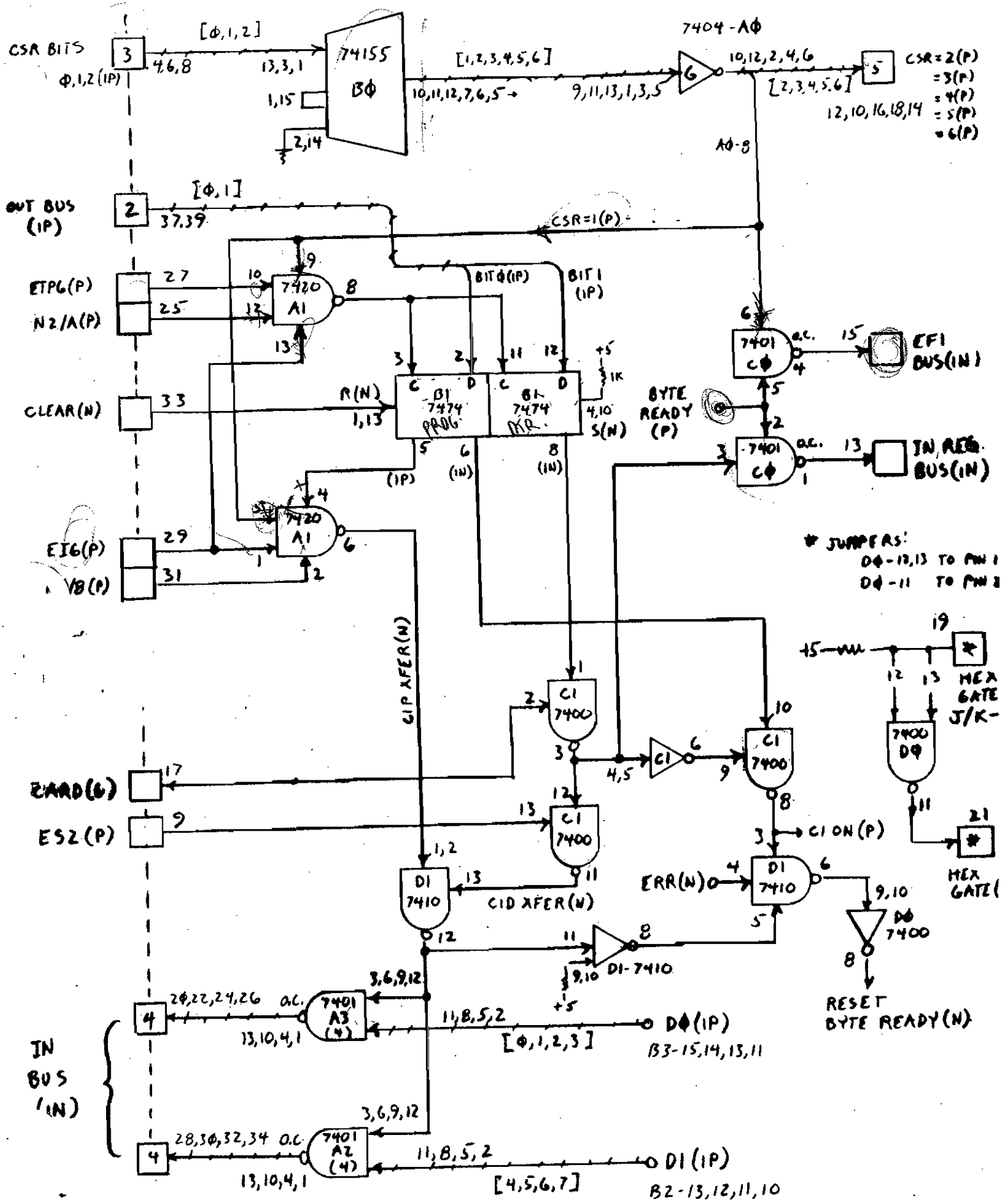
LOGIC-14

Board P2

redraw a Rom 4/24/72

CARD E1

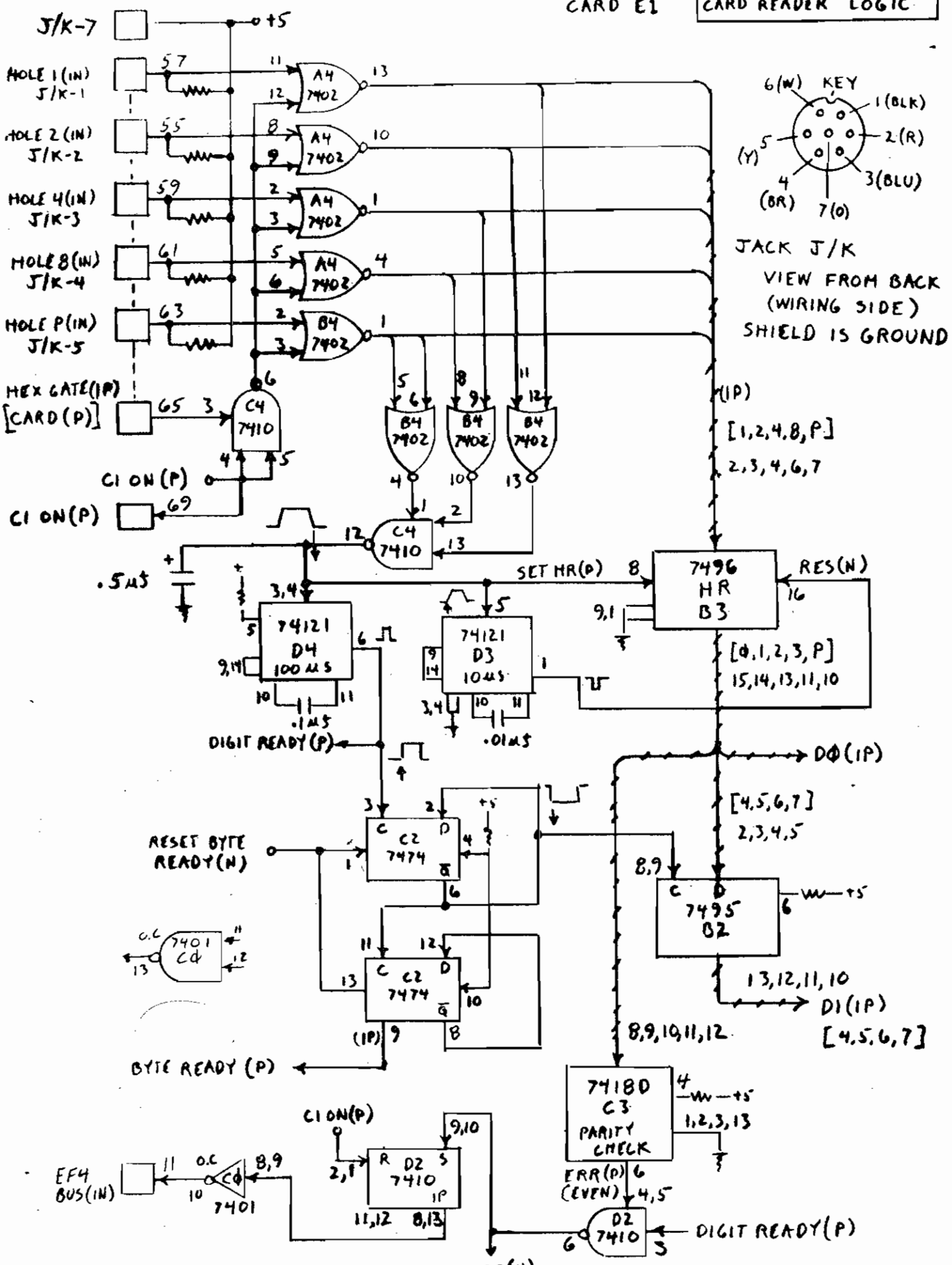
CARD READER LOGIC



JAW 5-18-72

CARD E1

CARD READER LOGIC

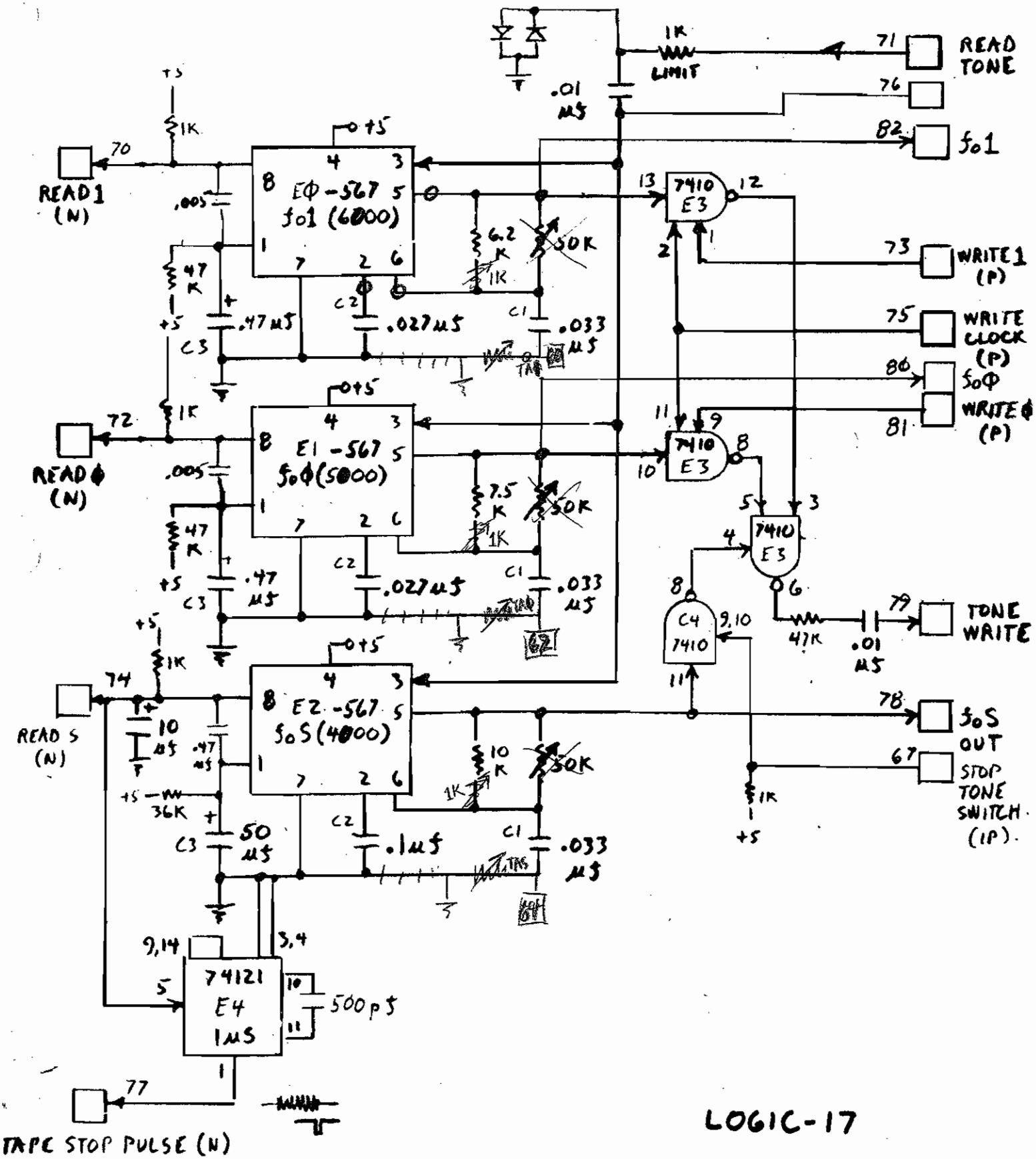


LOGIC-16

JAN 5-18-72

CARD E1

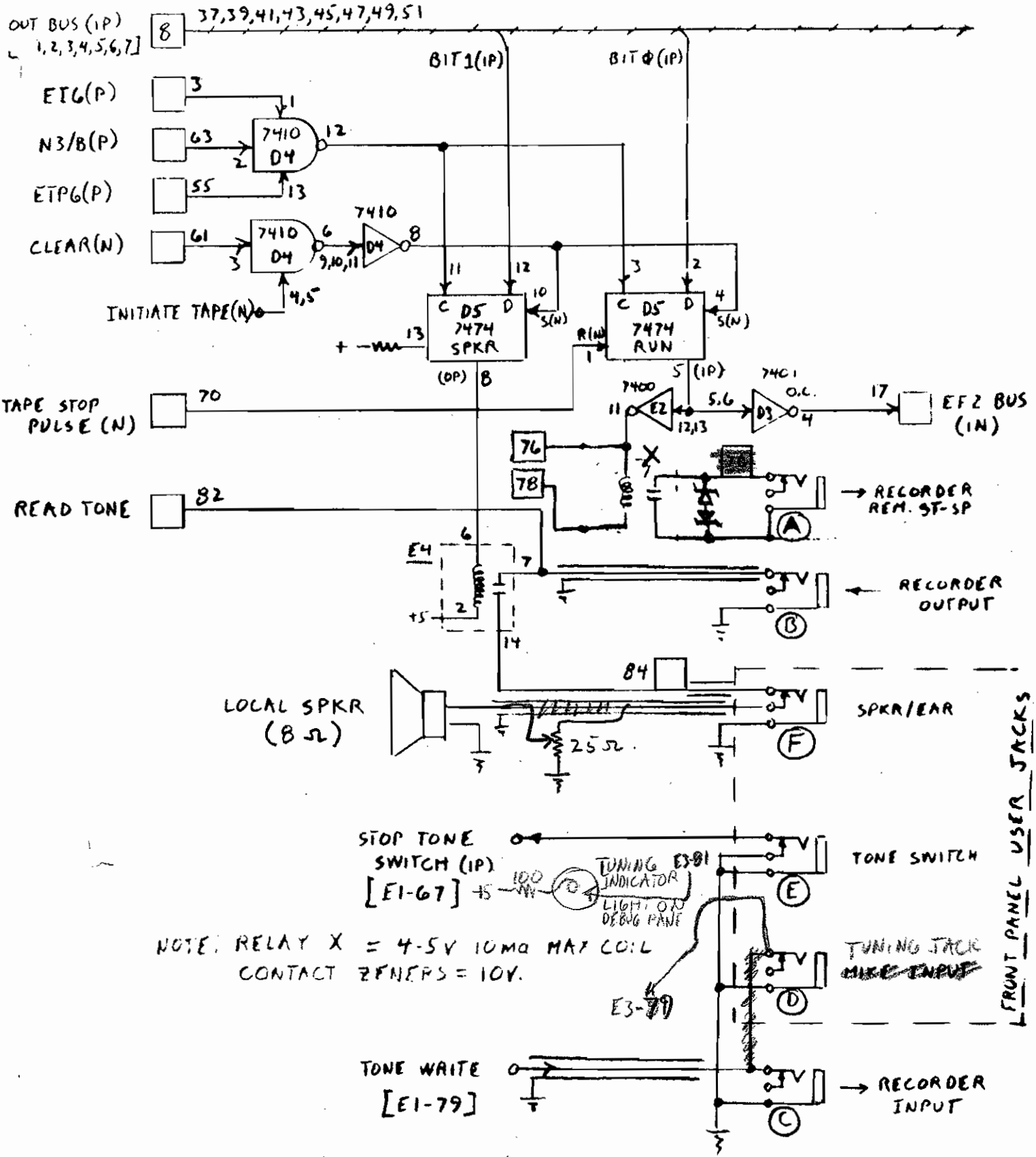
TAPE TONE CKTS



LOGIC-17

JAW 5-18-72

CARD E3 | TAPE CONTROL LOGIC



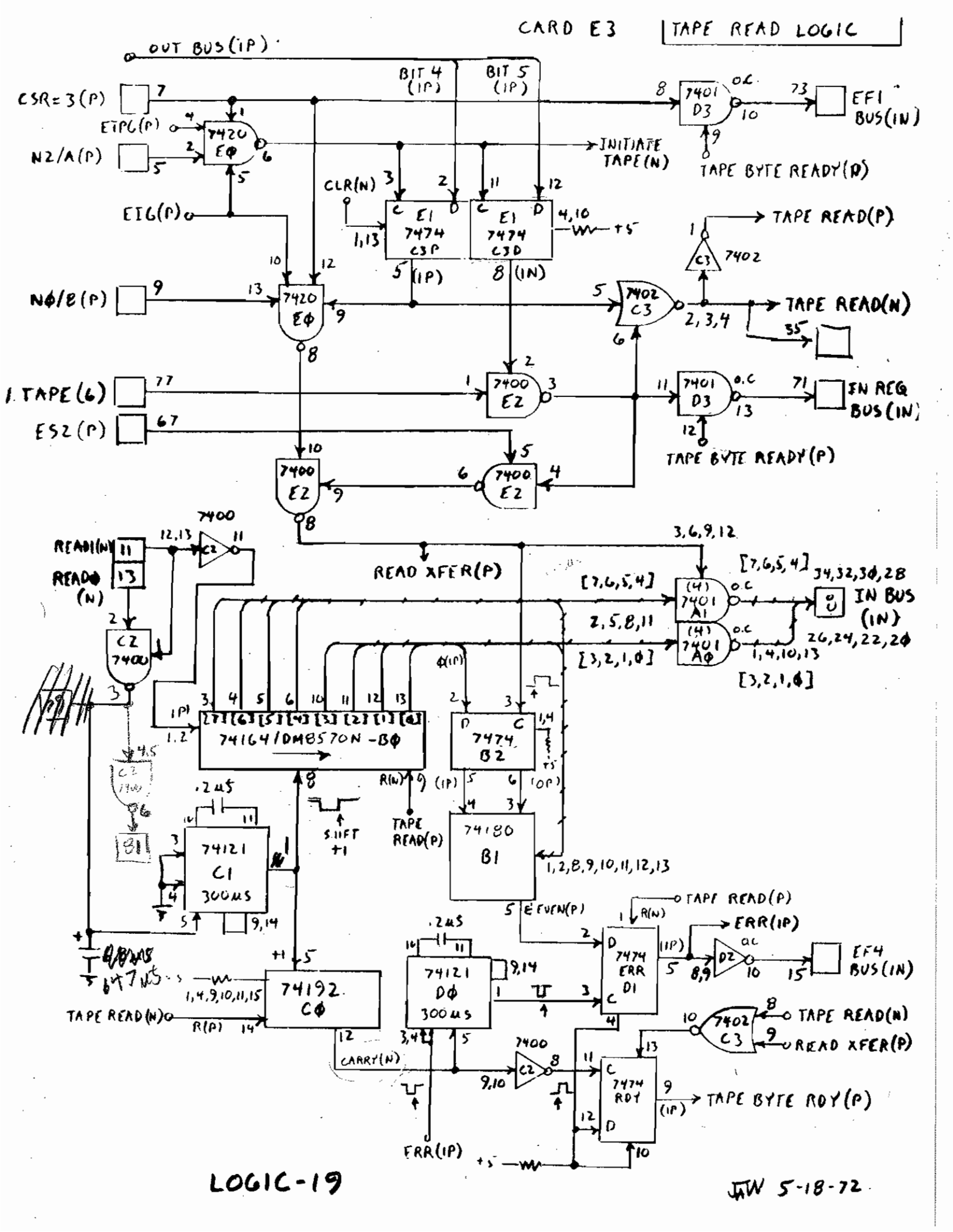
NOTE: RELAY X = 4-5V 10ma MAX COIL
CONTACT ZENERS = 10V.

*NOTE: ISOLATE JACK "A" FROM CHASSIS

LOGIC-18

JAW 5-18-72

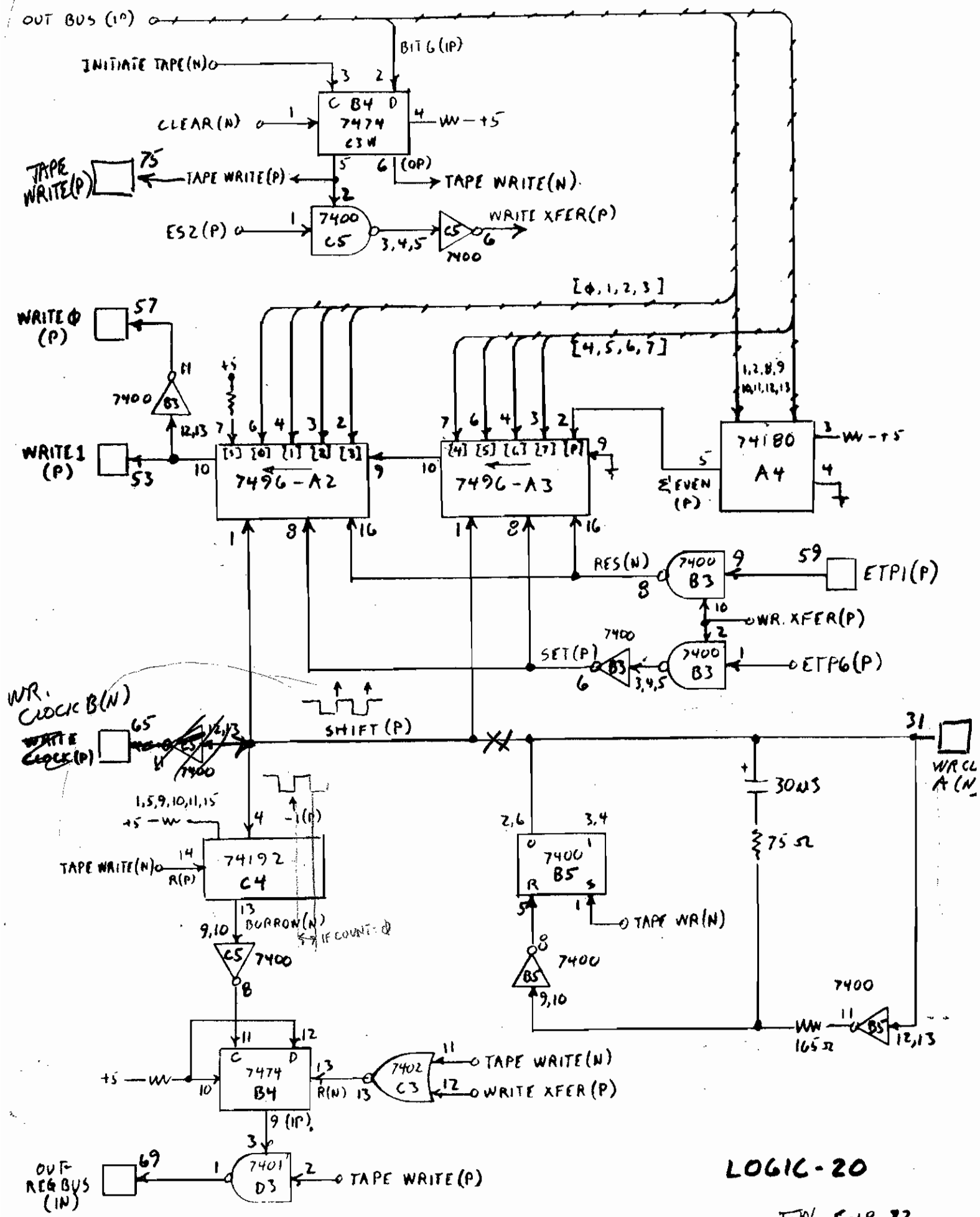
CARD E3 | TAPE READ LOGIC



LOGIC-19

JTW 5-18-72

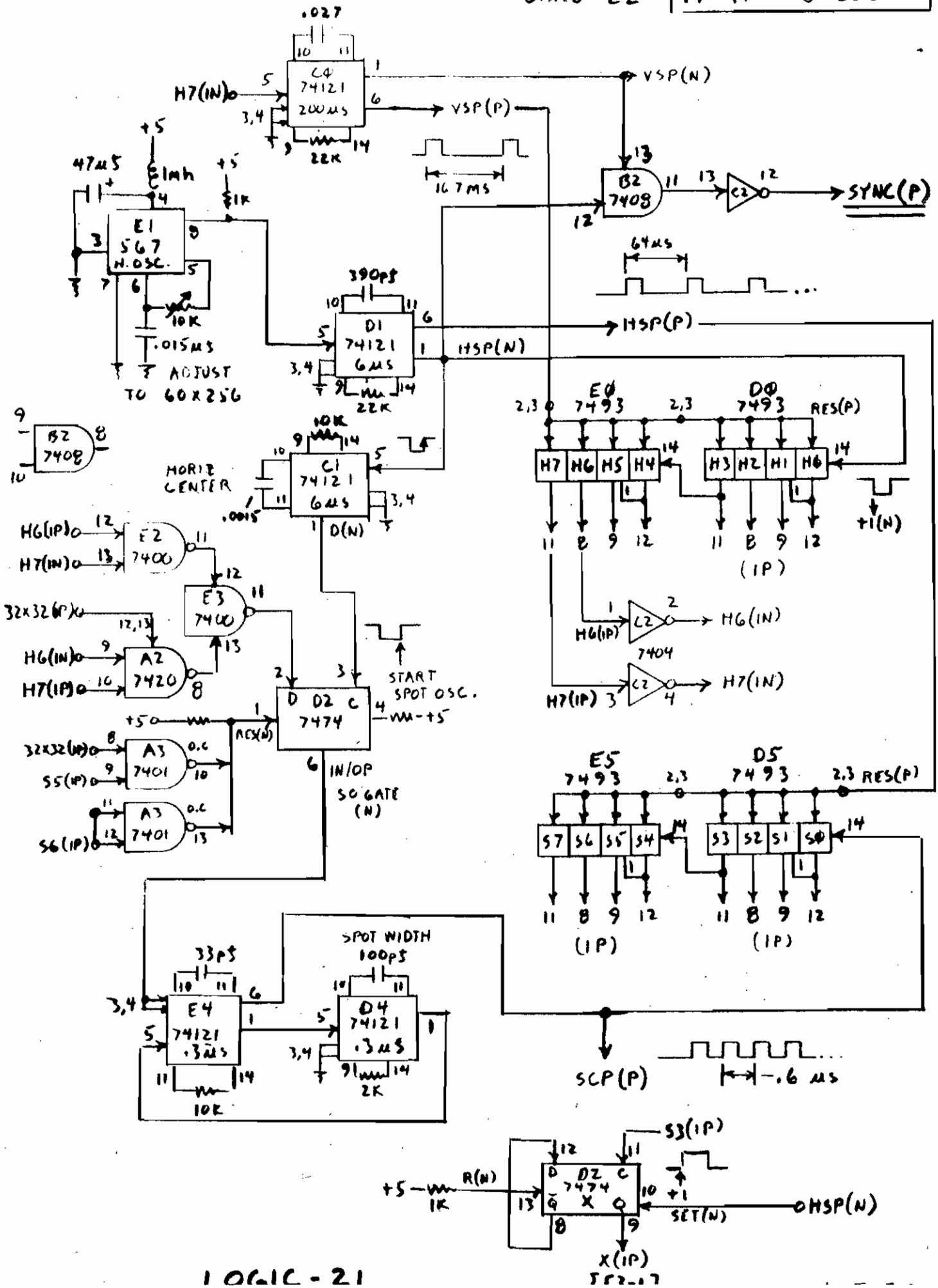
CARD E3 | TAPE WRITE LOGIC



LOGIC - 20

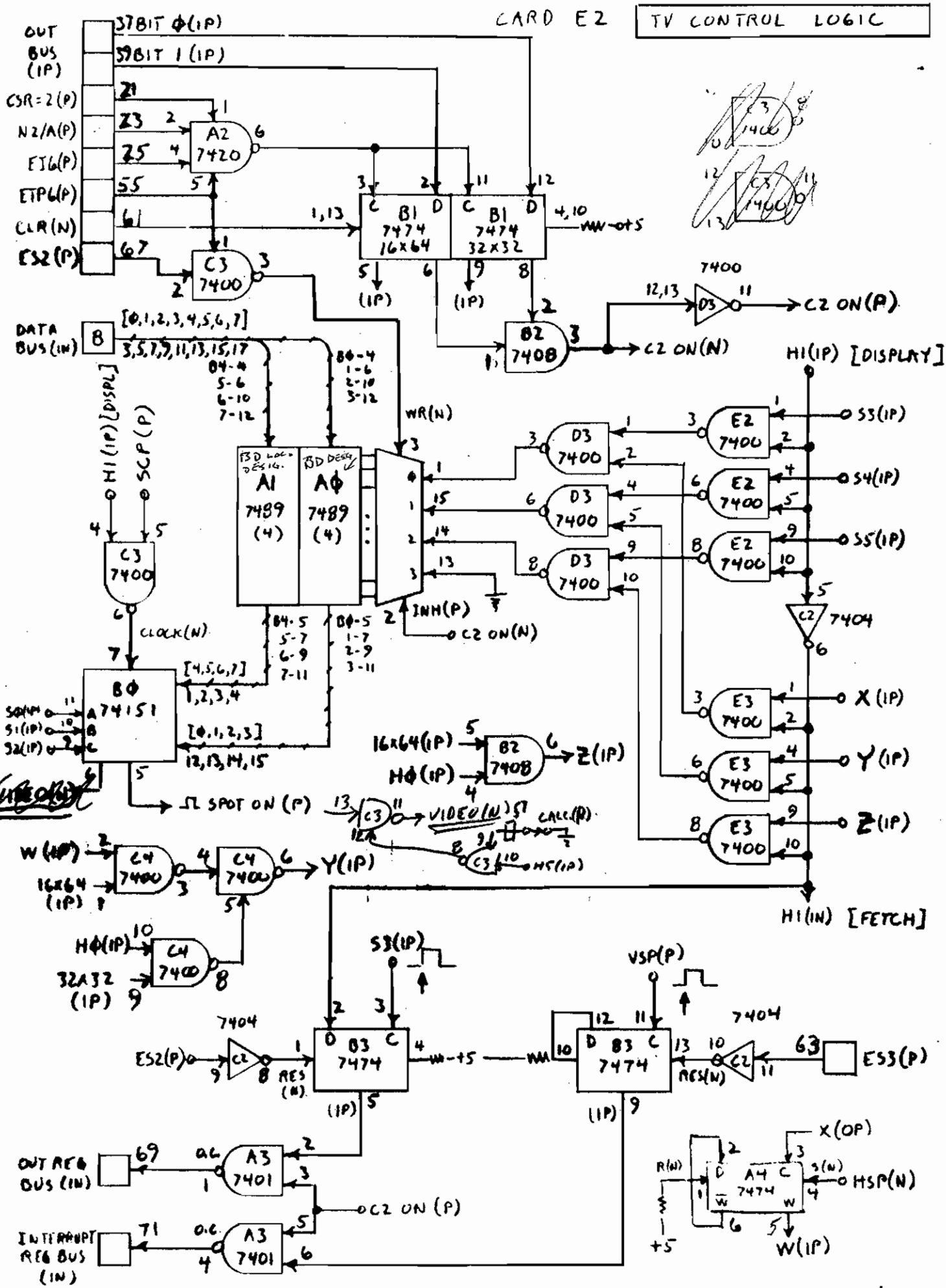
JAW 5-18-72

CARD E2 | TV TIMING LOGIC



CARD E2

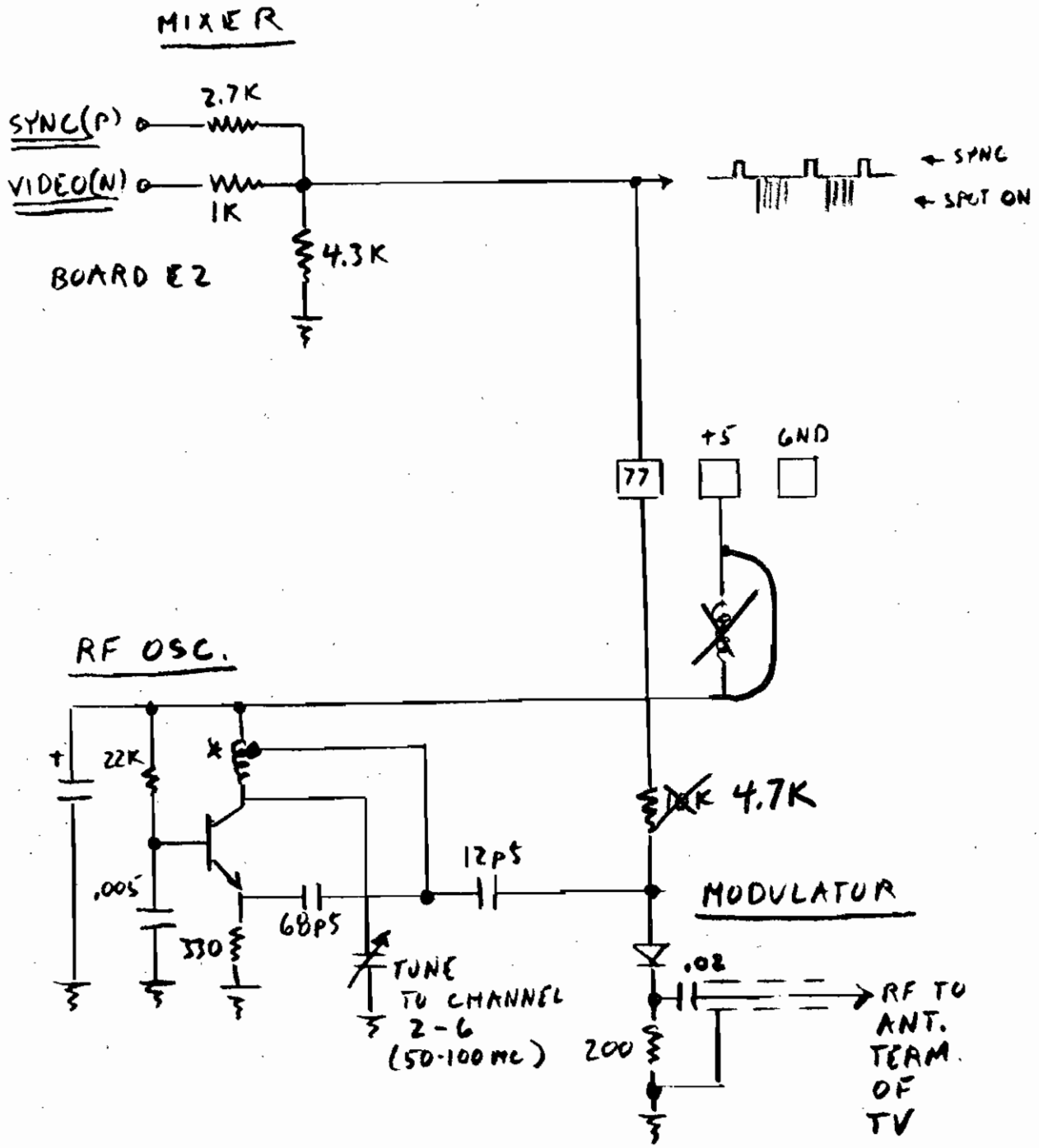
TV CONTROL LOGIC



LOGIC-22

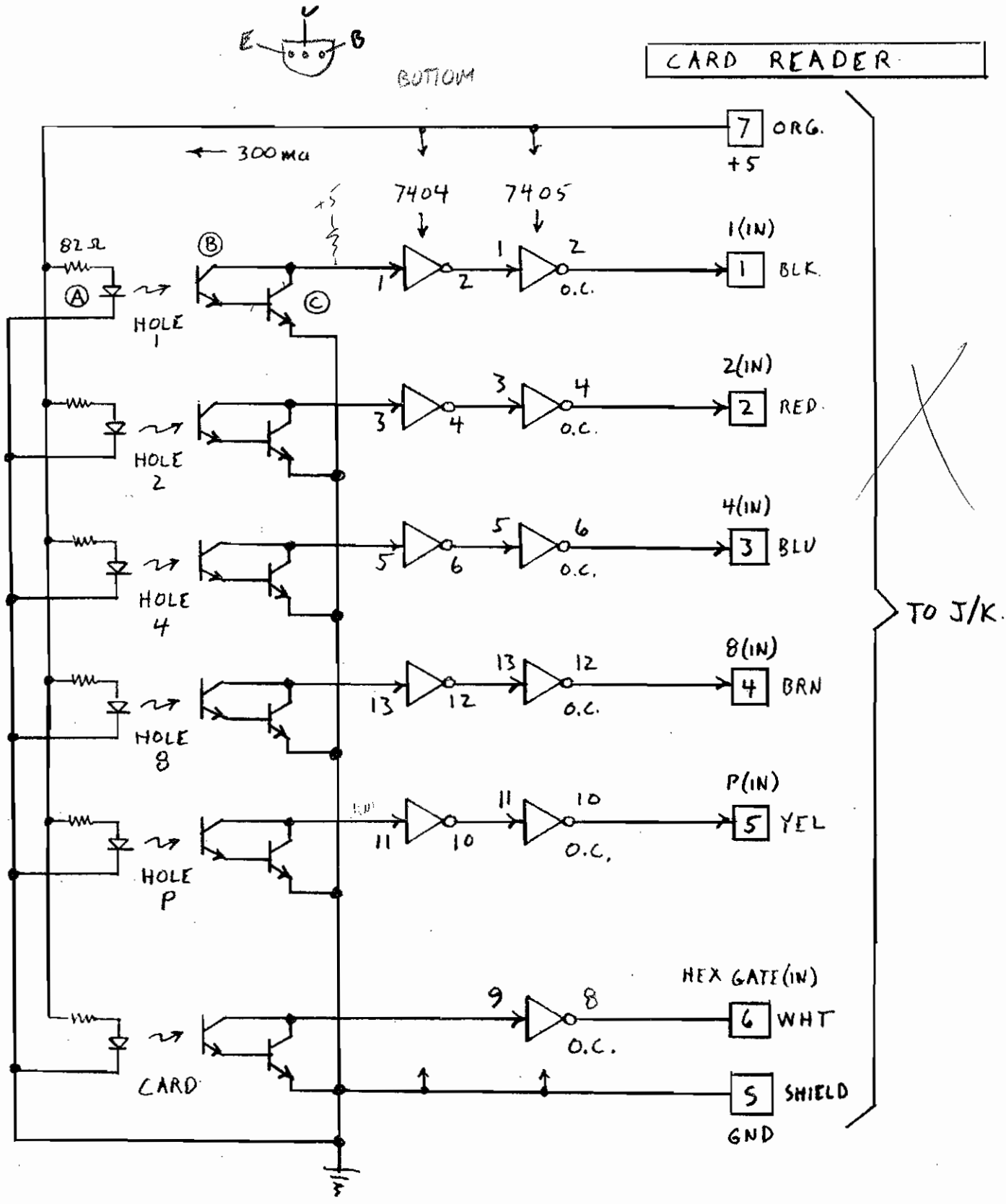
JAW 5/1.8/72

TV RF CKTS



★ 4.5 TURNS # 20 WIRE
1/4" DIA. 3/8" LONG
TAPE 1 TURN FROM + END

JAW 5-18-72.



(A) = MOTOROLA MLED 50

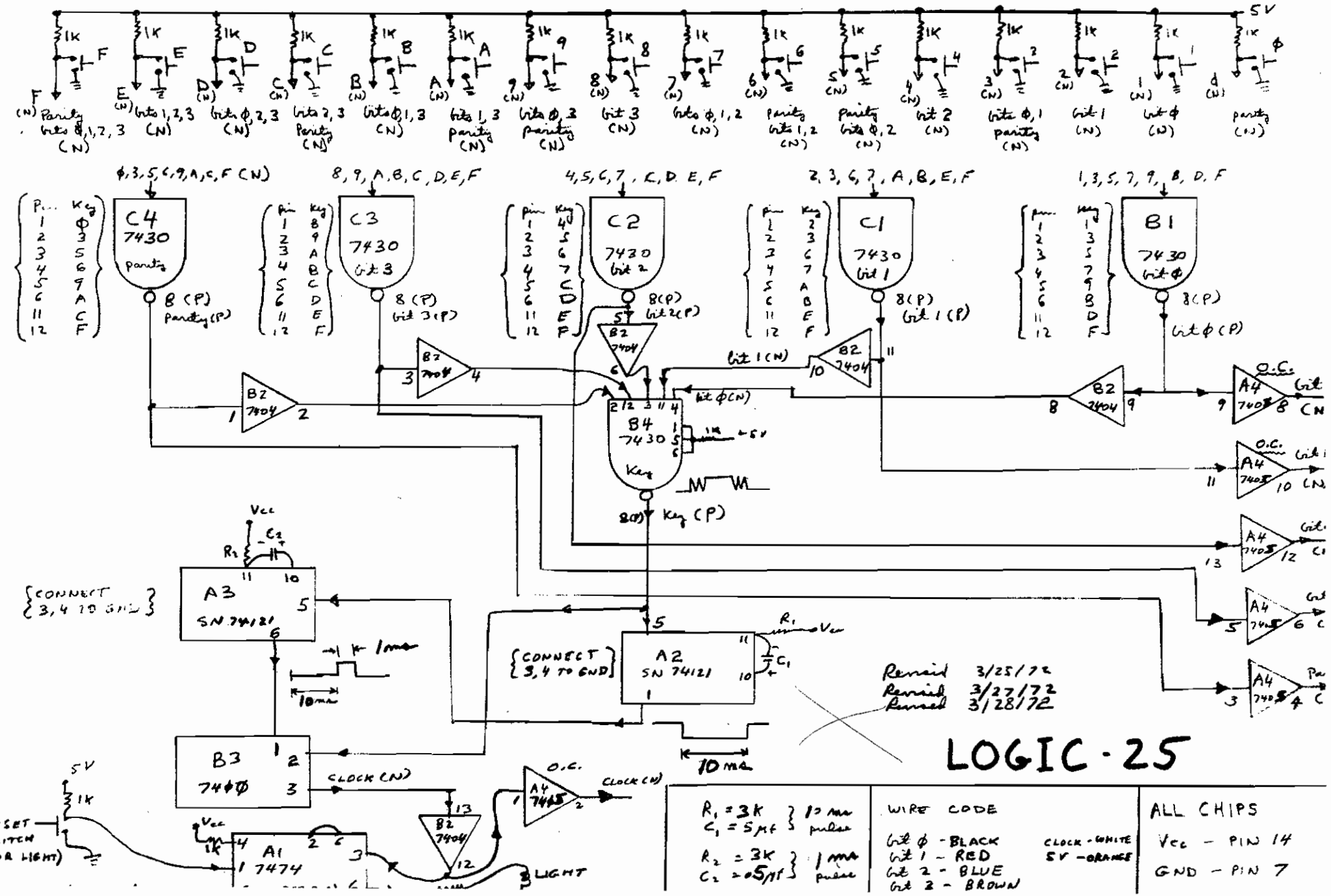
(B) = MOTOROLA MRD 450

(C) = 2N3391

LOGIC-24

JAW 4/1/72

HEX KEYBOARD MICROPROCESSOR



Revised 3/25/72
Revised 3/27/72
Revised 3/28/72

LOGIC-25

$R_1 = 3K$ } 10 ma pulse
$C_1 = 5\mu f$ }

$R_2 = 3K$ } 1 ma pulse
$C_2 = .05\mu f$ }

WIRE CODE

bit 0 - BLACK
bit 1 - RED
bit 2 - BLUE
bit 3 - BROWN
CLOCK - WHITE
5V - ORANGE

ALL CHIPS
Vcc - PIN 14
GND - PIN 7

SECTION V - SAMPLE PROGRAMS

The examples presented in this section were selected to clarity the operation of the system. Examples of input-output modes and interrupt processing are provided. All examples are in machine language. The coding form shown in Figure 15 was designed for use with FRED. Each sheet provides space for 48 bytes or instructions.

The "S" column can be used to name specified memory locations (bus-routine entry points, table base addresses, etc.). The "P" column facilitates keeping track of the register being used as the P counter for each section of code. The "X" column can be used to indicate which register has been assigned to X. The "M" column should contain consecutive memory addresses. The (M) column represents the contents of each memory location. This column represents the sequence of bytes to be placed in memory. The last two columns facilitate self documentation of programs.

A. DEDUCE (PROCESSING INSTRUCTIONS)

This program is illustrated by flowchart A and the following two coding sheets. "DEDUCE" illustrates the use of the card reader and utilizes the 8-bit lights on the control panel for output.

The program should be contained on FRED cards or a cassette (followed by a "stop" tone). In addition to the program, a player card is required. This card is shown on the DEDUCE card sheet.

The player first loads the program starting at M(ØØØØ) as described in Section III. He then chooses one of the 8 lights (Ø-7). Pressing RESET, then RUN, initiates the program. First, the light pattern "1111ØØØØ" is displayed. The player responds by entering "ØØ" with the "NO" card if his chosen light is off and "11" (or "YES") if it's on. The program displays two more patterns of lights, waiting for a player card response following each pattern. After the last (third) player response the program deduces the chosen light, turns it on, and stops. A Reset-run permits the game to be repeated.

The program requires only 7Ø bytes. This is relatively good memory utilization since only 18 average (4 byte) 360 type machine instructions could be provided in the same memory space.

The program flow as shown is straightforward. The section of code M1 partially initializes registers. Since R(Ø) will be subsequently required as a card input byte storage pointer the program counter is changed to R(5) leaving R(Ø) free. Initialization is resumed at M2. The card reader is selected and set to the direct mode. As discussed previously, the direct mode causes an automatic memory store cycle (at M(R(Ø))) to occur for each byte entered (2 hex digits).

The idle instruction (IØ) at M3 halts the program until an input byte storage cycle occurs. Following the stolen input byte storage cycle, the instruction following idle is fetched and executed. Since R(Ø) is automatically incremented during each input byte storage cycle it must be decremented to point to the byte just entered. An idle instruction always causes M(R(N)) to be placed on the byte bus. The idle instruction here, causes M(R(1)) to be placed on the bus. The 8 bus lights therefore display

the contents of M(R(1)) for the duration of the idle instruction. In this manner the 3 bit patterns at MA are displayed sequentially as required.

The storage of an input byte causes program resumption (after idle) and the newly entered byte at IA is tested. From the responses to the 3 light patterns the number of the chosen light is readily determined. This is converted to a single light pattern via table MT. The use of an IC type of instruction to perform this table look up is illustrated in program segment M7.

It should be again noted that while programs are normally loaded starting at M(ØØØØ), normal initiation (RESET-RUN) always skips M(ØØØØ). Program execution therefore begins at M(ØØØ1) leaving M(ØØØØ) free to be used in any manner desired.

B. DISPLAY (INTERRUPT AND CYCLE STEALING)

The next sample program (SB-display) illustrates the use of the display and program interrupt. The program flowchart is shown with detailed coding on the following two sheets. It requires 2Ø8 bytes of memory, 128 of which are required as display refresh storage. The program itself requires only 8Ø bytes.

128 bytes (1024 bits) of memory are displayed in a 32x32 bit format on the TV. M(Ø1ØØ-Ø17F) is used as the area of memory displayed.

The program is loaded starting at M(ØØØØ). It is initiated at M(ØØØ1) by reset and run switches. The display will be blank at this point. Any pattern of dots desired can now be displayed. First two hex digits are entered specifying a displayed byte address. A second byte is then entered representing the actual 8 bit pattern to be displayed at the previously entered address. This second set of 8 bits will immediately appear on the

display screen. Repeating the above permits construction of any dot pattern on the display screen.

The typical display page following the display flowchart illustrates how the word "JOE" could be displayed as an array of dots. The sequence of bytes (in hex notation) could be punched in a set of FRED cards. When this sequence of bytes is entered the "display" program would yield the TV screen pattern of dots shown.

Code M1, M2 initializes registers and sets a 1 bit counter (K) to Ø. K specifies whether an input byte specifies a display address or an 8-bit display pattern. Display refresh requires R(Ø) for cycle stealing and R(1) and R(2) for program interrupt. M1 code therefore changes the program counter from R(Ø) to R(3) and resumes initialization at M2.

The C1 loop initially clears the memory area to be displayed. M3 indicates the use the I6 type of instruction to activate the display interface. From this point on, the display interface will automatically steal memory cycles to retrieve memory bytes for display refresh. R(Ø) is used for this purpose. TV generated program interrupts are also required to reset R(Ø) to the beginning of the memory display area after 128 bytes have been refreshed. This program interrupt will be ignored until the interrupt mask (IM) is reset. IM is always set when the machine is cleared. Initial IM reset is performed by a dummy "7Ø" instruction in M3. Note that P and X are set by this instruction so that careful use of this instruction is required.

After activation of the TV display, the card reader is selected. Note that, once initiated, TV display operation is independent of external device selection. The card reader is now set to the program mode and EF1 is monitored

in loop M4 to determine when an input byte is available to be stored in memory. The stored input byte is then used to modify R(A) or displayed byte M(R(A)) depending on the value of K. Input monitoring is then resumed until a new input byte is entered.

Since the TV display causes program interrupts an interrupt routine is provided (I1-I2-I3). This routine is automatically entered when an interrupt occurs. P is set to 1 and X to 2 by the interrupt. Instruction "78" in I1 stores X and P values of the interrupted program which are contained in T. Instruction "7Ø" in I3 restores the original values of X and P thereby effecting return to the interrupted routine. This instruction also resets the interrupt mask which is set each time an interrupt occurs.

A side benefit resulting from the simple machine structure is the ability to store its state in two bytes (D, X, and P).

C. CLUE (CASSETTE CONTROL)

CLUE illustrates the use of the cassette player audio control instructions. Eight cards with pictures of objects on them are provided as shown in the following table. The "byte" column indicates the two hex digit byte punched in each card. This byte will be read by the card reader whenever a card is entered by the FRED user.

CARD	PICTURE	BYTE
Ø	TOP	ØØ
1	LAMP	Ø1
2	CAR	Ø2
3	JAR	Ø3
4	EGG	Ø4
5	RING	Ø5
6	APPLE	Ø6
7	BOAT	Ø7

FRED will randomly select one of the 8 pictured objects. The user can then enter groups of four object cards and FRED will "tell" him whether or not the chosen object was among the four entered. After 3 such clues the user must guess FRED's chosen object. As can be seen this program is actually an introduction to the concept of intersecting subsets for young children.

In addition to the above cards a magnetic tape cassette is provided. It is divided into a number of "FRAMES" separated by stop tones. This tape has the following format. Note that " " indicates voice on tape.

FRAME 1:

Deduce program recorded as series of 0 and 1 tones followed by:

"Turn off the read switch."

"Wait until after the tone, then press reset."

STOP TONE 1

FRAME 2:

"Press the run switch"

STOP TONE 2

FRAME 3:

"FRED will now choose one of the eight objects shown on the cards. Can you guess which one FRED chose? You can have three clues. For your first clue drop any four cards into the card slot"

STOP TONE 3

FRAME 4:

"No, FRED's object is not any of those four".

STOP TONE 4

FRAME 5:

"Yes, the chosen object is one of those four".

STOP TONE 5

FRAME 6:

"Now, put in another group of four cards for your second clue."

STOP TONE 6

FRAME 7:

"No, the object isn't one of those".

STOP TONE 7

FRAME 8:

"Yes, the object is one of those."

STOP TONE 8

FRAME 9:

"This will be your last clue. Put in one more group of four cards".

STOP TONE 9

FRAME 10:

"No, it wasn't there".

STOP TONE 10

FRAME 11:

"Yes, it was there."

STOP TONE 11

FRAME 12:

"Now, drop in the card you think FRED chose"

STOP TONE 12

FRAME 13:

"Wrong, FRED wins"

STOP TONE 13

FRAME 14:

"Congratulations! You guessed right".

STOP TONE 14 (END OF TAPE)

FRED is first turned on and RESET. The above cassette is placed in the recorder and rewound. The cassette recorder is then placed in PLAY mode. FRED is RESET and the READ switch turned on. The program will be loaded in memory starting at M(ØØØØ). Stop tone 1 will automatically stop the tape at the beginning of FRAME 2.

Note that the user was given voice instructions to turn off the READ switch and to RESET FRED in FRAME 1. RESET will automatically start the cassette recorder and the user will hear the voice instructions to "press the RUN switch" provided in FRAME 2. At this point program execution will begin with the instruction at M(ØØØ1). STOP TONE 2 will automatically stop the tape prior to FRAME 3.

Flowchart C illustrates the subsequent operation of the CLUE program. The 3 pages following flowchart C show the detailed coding.

Since the RUN switch may be pressed prior to STOP TONE 2, M1 prevents further program execution until after STOP TONE 2 causes the tape to stop just prior to FRAME 3. At this time, program execution will be resumed at M2. The tape is restarted with the speaker on so that the user will hear FRAME 3 played. While FRAME 3 is being played "M3-M4-MP" generates a random number (N) between Ø-7. The first card entered freezes N. The "M18-M19" loop receives the first group of 4 cards requested in FRAME 3.

After STOP TONE 3 the "M5-M11" sequence plays or skips FRAMES 4 and 5 depending on whether a card equal to N was entered in the first group. "M14" then plays FRAME 6, requesting the next group of four cards.

After the third group of cards has been entered, "M15" adjusts the program parameters "C" and "K" so that after one more card (the final guess) the answer in FRAME 13 or 14 is played. The idle instruction at "M21" ends the program. Resetting and rewinding the tape permits playing again.

D. WRITE (TAPE WRITE PROCEDURE)

The following outlines one procedure for preparing FRED cassettes. For illustrative purposes, all steps in preparing the "CLUE" tape previously discussed will be described.

First, a program for writing an area of memory to tape must be available. A suitable tape write program is shown in flowchart D with detailed coding on the following page. This program occupies the first 48 bytes of memory.

While FRED tapes can be played on most inexpensive cassette players, they should only be prepared on high quality recorders. Best results have been achieved using cassettes loaded with chromium-dioxide tapes. The high frequency boost and low drop-out of these tapes tends to compensate for otherwise poor performance of low quality players.

The following sample procedure for preparation of the "CLUE" tape requires that the hi-level (AUX/LINE) input of a high quality audio tape recorder be connected to jack "C" of FRED. Jacks A, B, and C are at the back of the computer cabinet. Jacks A and B are not used for recording and should not be connected.

Under no circumstances should a tape recorder with "automatic level control" recording circuits be used. This type of recorder will invariably produce unreadable tapes.

A "normally closed" switch should be plugged into jack E on the FRED front panel. This can be the "remote" start-stop switch contained on most cassette recorder microphones. This switch, when momentarily opened, will record a stop tone on the tape.

With the above connections made, the tape recorder should be positioned to the beginning of tape and placed in the RECORD mode with PAUSE on. (A recorder PAUSE mode is assumed here).

Assuming that the TAPE WRITE program has been previously punched on FRED cards, it is now loaded in memory. To load the WRITE program:

1. press RESET
2. READ on
3. CARD on
4. Enter WRITE program cards in proper sequence via card reader [M(ØØØØ) to M(ØØ2D)]
5. Enter 2-byte "END" address via parameter card (or deactivate reader and enter "END" address via optional hex switch panel).

The most significant byte of the 2-byte "END" address should be entered first followed by the least significant byte. The "END" address will be automatically stored in M(ØØ2E)-M(ØØ2F).

The "END" address is calculated by:

EA=N+2

where:

EA=2-byte END address (hex)

N =The number of bytes in the program/data to be recorded on tape.

In this example the "CLUE" program will be recorded on tape. The "CLUE" program has ØØ67 bytes (hex). The required "END" address will therefore be ØØ67+ØØ3Ø+ØØØ2 or ØØ99.

Following the "END" address the "CLUE" program must be loaded in memory. This would normally be done via the optional hex keyboard. Entering the "CLUE" program at this point will load it into M(ØØ3Ø) through M(ØØ97).

CARD and READ switches are now turned off and the RESET switch pressed. Pressing RUN will now cause recording to begin. Thie RESET-RUN sequence may be repeated as often as desired. Since the recorder is in "PAUSE" the

tape is not moving and the program can be "RUN" to set the proper record levels at the recorder.

After setting the proper levels and the "WRITE" program has terminated, start the tape in the record mode. After the required "leader" delay press RESET then RUN to initiate actual recording. Stop the tape as soon as the WRITE program terminates. Connect a microphone to the recorder "mike" input, start the tape, and record the voice portion of Frame 1. Momentarily open the switch connected to FRED jack E to record STOP TONE 1. Continue to record voice FRAMES 2-14 with STOP TONES 2-14.

The master CLUE tape is now complete. As many cassettes as desired can now be made from this master tape.

The previous procedure could, or course, be modified. It has been found convenient to use a two channel reel to reel recorder for making master tapes. Any high quality audio units are applicable.

E. SUBROUTINES

Because of the limited instruction set, extensive use of subroutines is anticipated. This is not illustrated in the above examples. The machine structure facilitates branch and link functions. One method is described below. For example, the following register conventions might be established:

R(3) - Main program counter (ØØAØ)

R(4) - Call routine pointer (Ø1Ø1)

R(5) - Subroutine program counter (Ø2XX)

If ( ) represents initial R(3), R(4), and R(5) contents with R(3) pointing to a subroutine call instruction, then the following illustrates one possible call and return sequence:

Main Program (P3)

M Location	Contents	Instruction	Comments
ØØAØ	D4	4→P	Jump to call routine
ØØA1	2Ø	---	Subroutine identifier
ØØA2	P1	---	Parameter 1
⋮	⋮	⋮	

Call Routine (P4)

M Location	Contents	Instruction	Comments
Ø1ØØ	D3	3 → P	Return to main program
Ø1Ø1	43	M(R(3))→D, R(3)+1	Put subroutine
Ø1Ø2	A5	D→RØ(5)	Identifier in R(5)
Ø1Ø3	D5	5→P	Jump to subroutine
Ø1Ø4	3Ø	ØØ→RØ(P)	Go to Ø1ØØ
Ø1Ø5	ØØ	---	---

Subroutine (P5)

M Location	Contents	Instruction	Comments
Ø22Ø	43	M(R(3))→D, R(3)+1	Get P1 from M (ØØA2)
⋮	⋮	⋮	⋮
Ø232	D4	4→P	Return to M (Ø1Ø4)

Using the above system requires only two program bytes to call a subroutine. These two bytes can then be followed by as many one byte subroutine parameters as required. It should be noted that the use of one byte instructions permits a calling routine of only six bytes. Other subroutine calling techniques could, of course, be used.

The above system also solves the problem of branching between 256 byte mini-pages. One subroutine can be a "go to" subroutine. A "go to" function would then be specified as follows:

DN, GT, XX, YY.

This four byte sequence would enter a call routine specified by R(N) which in turn selects the "go to" subroutine specified by "GT". The "go to" subroutine would place "XX" and "YY" into the original program counter and return to this "go to" address (via the call routine as above). In this manner a branch to any memory location would only require four bytes.

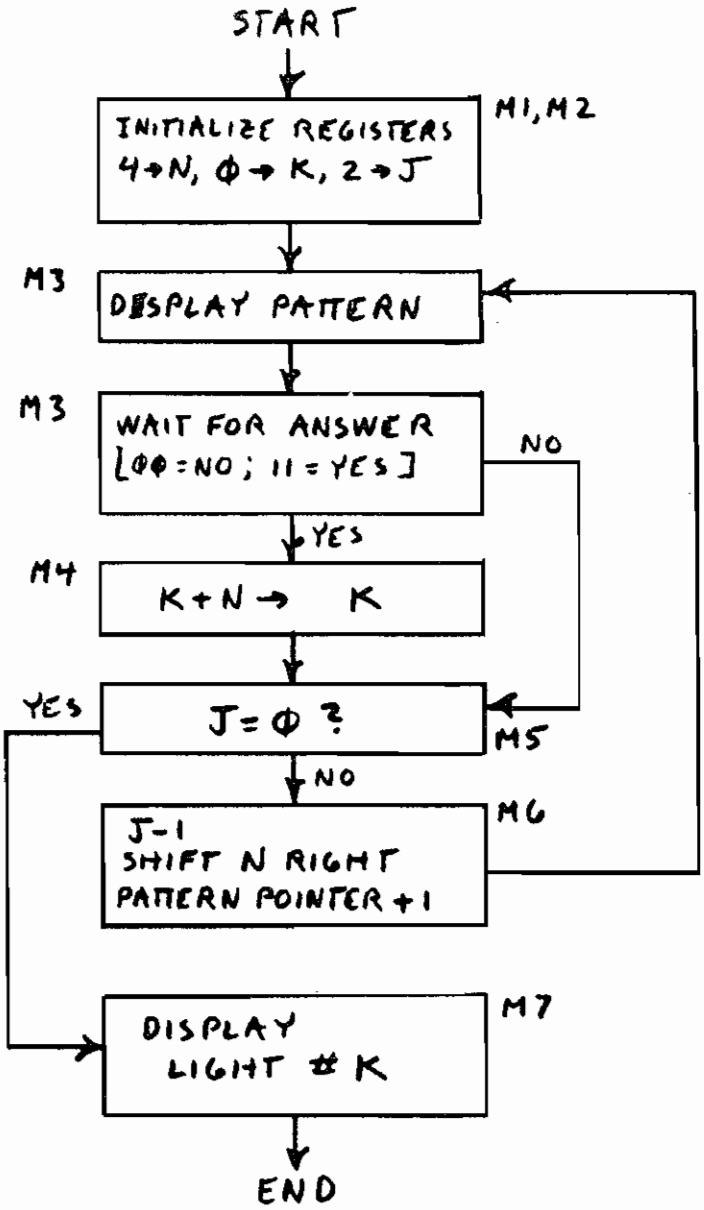
It is advisable to set up sub-routine calling conventions whenever the program exceeds 256 bytes.

FIG. 15 - CODING FORM

NO. PROGRAM DATE PAGE

	S	P	X	M	(m)	INSTRUCTION	comments
card							
card							
card							
card							

SA- SAMPLE PROGRAM A - DEDUCE



R(Φ) = INPUT ANSWER BYTE ADDRESS
R(1) = PATTERN POINTER
M(R(2)) = K
RΦ(3) = J
RΦ(4) = N
R(5) = PROGRAM COUNTER

FLOWCHART A

JAW 12/7/71

NO. 5A PROGRAM DEDUCE DATE 12/7/71 PAGE 12

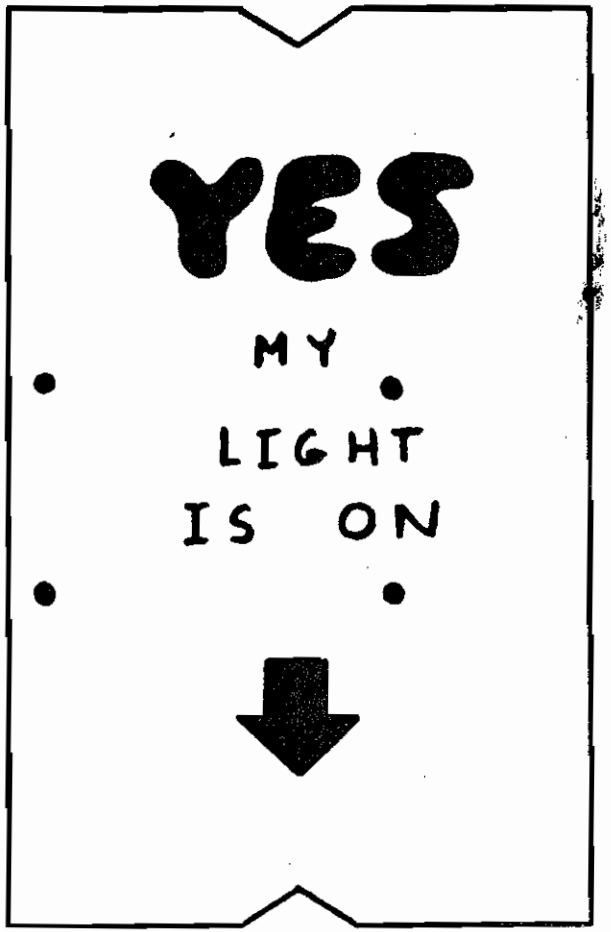
	S	P	X	M	(m)	INSTRUCTION	comments
card-1	IA	—	—	ΦΦΦΦ	ΦΦ	NOT EXECUTED	INPUT ANSWER BYTE STORAGE
	M1	Φ	—	ΦΦΦ1	90	R1(Φ)→D	"ΦΦ"→D
		Φ	—	2	B1	D→R1(1)	
		Φ	—	3	B2	D→R1(2)	
		Φ	—	4	B5	D→R1(5)	
		Φ	—	5	4Φ	M(R(Φ))→D, R(Φ)+1	
		Φ	—	6	Φ2	"Φ2"	
		Φ	—	7	A3	D→RΦ(3)	2→J
		Φ	—	8	4Φ	M(R(Φ))→D, R(Φ)+1	
		Φ	—	9	18	"18"	
		Φ	—	A	A5	D→RΦ(5)	M2→R(5)
		Φ	—	B	D5	5→P	GO TO M2
card-2		—	—	ΦΦΦC	ΦΦ	—	K
	MA	—	—	ΦΦΦD	F0	"1111 ΦΦΦΦ"	PATTERN TABLE
		—	—	E	CC	"11ΦΦ11ΦΦ"	
		—	—	F	AA	"1Φ1Φ1Φ1Φ"	
	MT	—	—	ΦΦ1Φ	Φ1		BINARY TO DECIMAL CONVERSION TABLE
		—	—	1	Φ2		
		—	—	2	Φ4		
		—	—	3	Φ8		
		—	—	4	10		
		—	—	5	20		
		—	—	6	40		
		—	—	7	80		
card-3	M2	5	—	ΦΦ1E	45	M(R(5))→D, R(5)+1	
		5	—	9	Φ4	"Φ4"	
		5	—	A	A4	D→RΦ(4)	4→N
		5	—	B	45	M(R(5))→D, R(5)+1	
		5	—	C	ΦD	"ΦD"	
		5	—	D	A1	D→RΦ(1)	MA→PATTERN POINTER
		5	—	E	95	R1(5)→D	
		5	—	F	AΦ	D→RΦ(Φ)	IA→R(Φ)
		5	—	ΦΦ2Φ	45	M(R(5))→D, R(5)+1	
		5	—	1	ΦC	"ΦC"	
		5	—	2	A2	D→RΦ(2)	K ADDRESS→R(2)
		5	—	3	E5	5→X	
card-4		5	5	4	G1	SELECT CARD RDR.	
		5	5	5	Φ1	"Φ1"	
		5	5	6	62	SET DIRECT MODE	
		5	5	7	Φ2	"Φ2"	
	M3	5	—	ΦΦ2E	Φ1	IDLE	M(R(1))→LIGHTS
		5	—	9	2Φ	R(Φ)-1	RESET R(Φ) TO IA AFTER INPUT BYTE
		5	—	A	4Φ	M(R(Φ))→D, R(Φ)+1	
		5	—	B	2Φ	R(Φ)-1	
		5	—	C	32	D=Φ?	
		5	—	D	32	32 YES →	CHOSEN LIGHT OFF; SKIP TO M5
	M4	5	—	ΦΦ2E	E2	2→X	
		5	2	F	84	RΦ(4)→D	N→D

JaW

NO. 5A PROGRAM DEDUCE DATE 12/7/71 PAGE 6/2

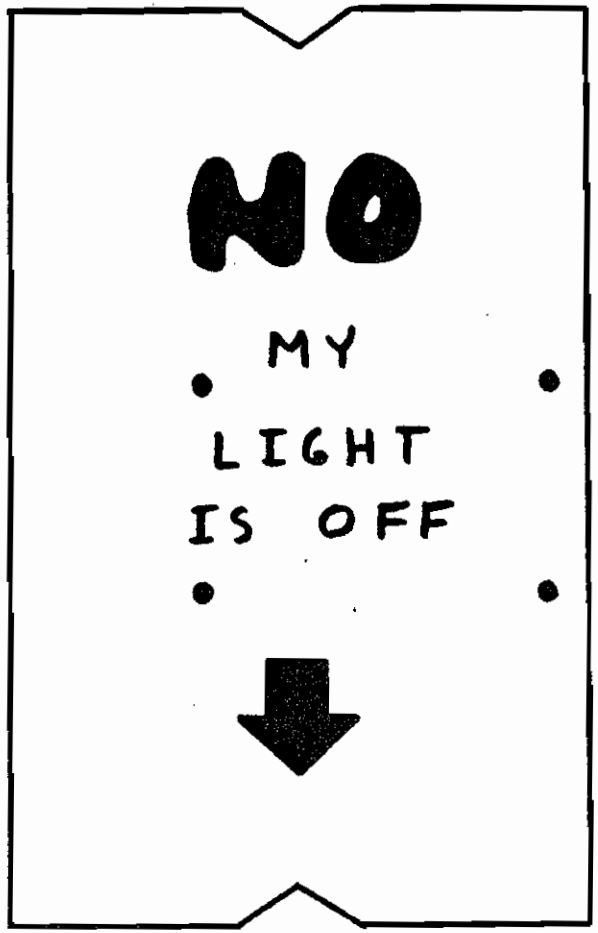
	S	P	X	M	(m)	INSTRUCTION	comments
card-5		5	2	0030	F4	M(R(x)) + D → D	K+N
		5	2	1	52	D → M(R(x))	K+N → K
	M5	5	–	0032	83	RQ(3) → D	
		5	–	3	32	D = 0 ?	
		5	–	4	3C	"3C" YES → J=0 → SKIP TO M7	
	M6	5	–	0035	23	R(3) - 1	J - 1
		5	–	6	11	R(1) + 1	PATTERN POINTER + 1.
		5	–	7	84	RQ(4) → D	
		5	–	8	F6	SHIFT RIGHT	
		5	–	9	A4	D → RQ(4).	SHIFT N'
		5	–	A	30	BRANCH	
		5	–	B	28	"28"	RETURN TO M3
card-6	M7	5	–	003C	11	R(1) + 1	M7 → R1
		5	–	D	42	M(R(2)) → D, R(2) + 1	
		5	–	E	C1	DQ → RQQ(1)	K → R1
		5	–	F	22	R(2) - 1	
		5	–	0040	95	R1(5) → D	
		5	–	1	52	D → M(R(2))	RESET K TO 00
		5	–	2	E5	5 → X	
		5	5	3	62	TURN SWITCHES OFF	
		5	5	4	00	0 0	
	END	5	5	0045	01	IDLE	CHOSEN LIGHT ON
card							
card							
card							

FRONT



READ (11)

BACK

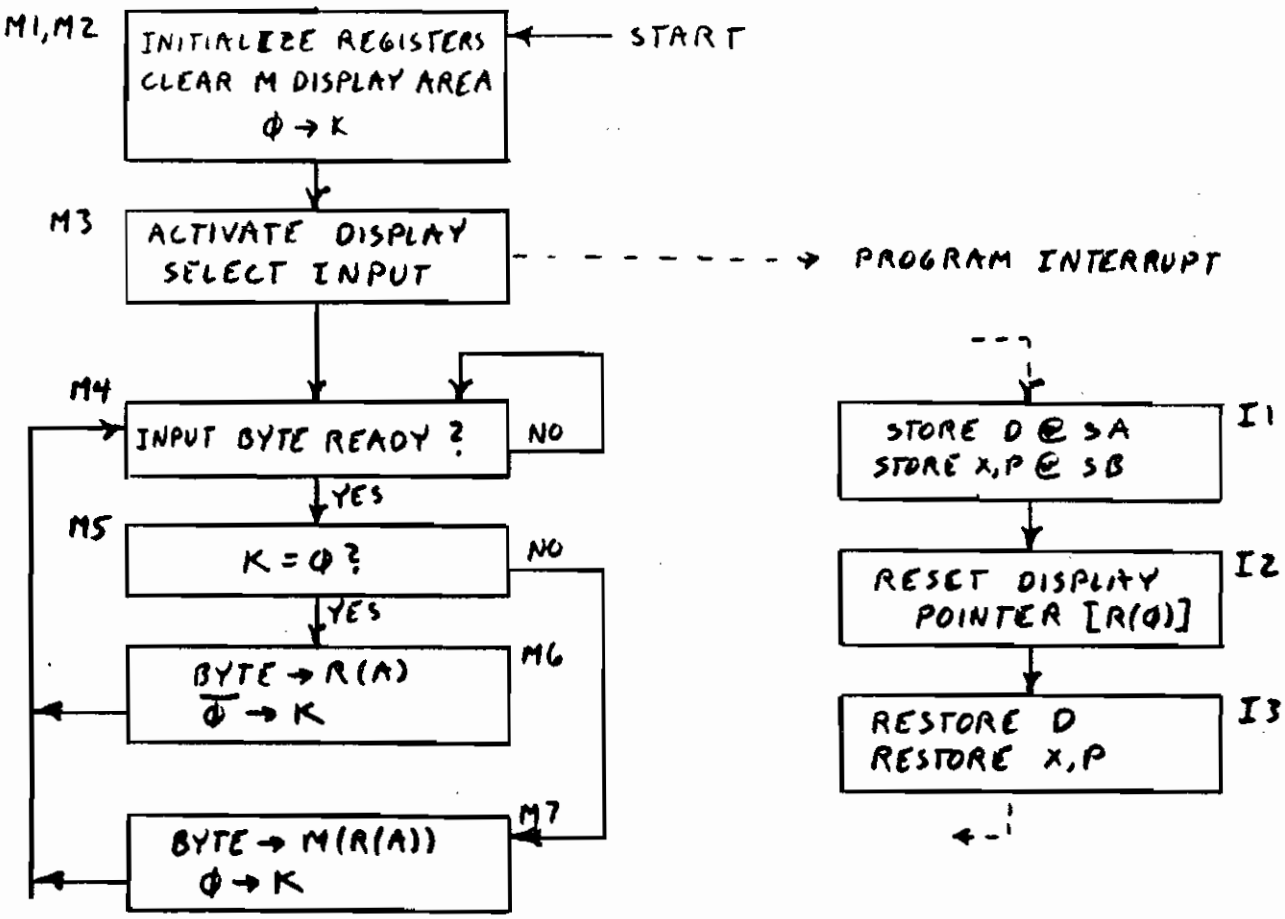


READ (00)

DEDUCE CARD

JAW

SB - SAMPLE PROGRAM B - DISPLAY



M(Ø1ØØ - Ø17F) = M DISPLAY AREA (128 BYTES).
R(Ø) = DISPLAY POINTER
R(1) = INTERRUPT PROGRAM COUNTER
R(2) = INTERRUPT STORAGE POINTER
R(3) = MAIN PROGRAM COUNTER
M(R(4)) = INPUT BYTE STORAGE (TEMPORARY)
RØ(5) = K
R(A) = DISPLAY MODIFICATION POINTER

FLOWCHART B

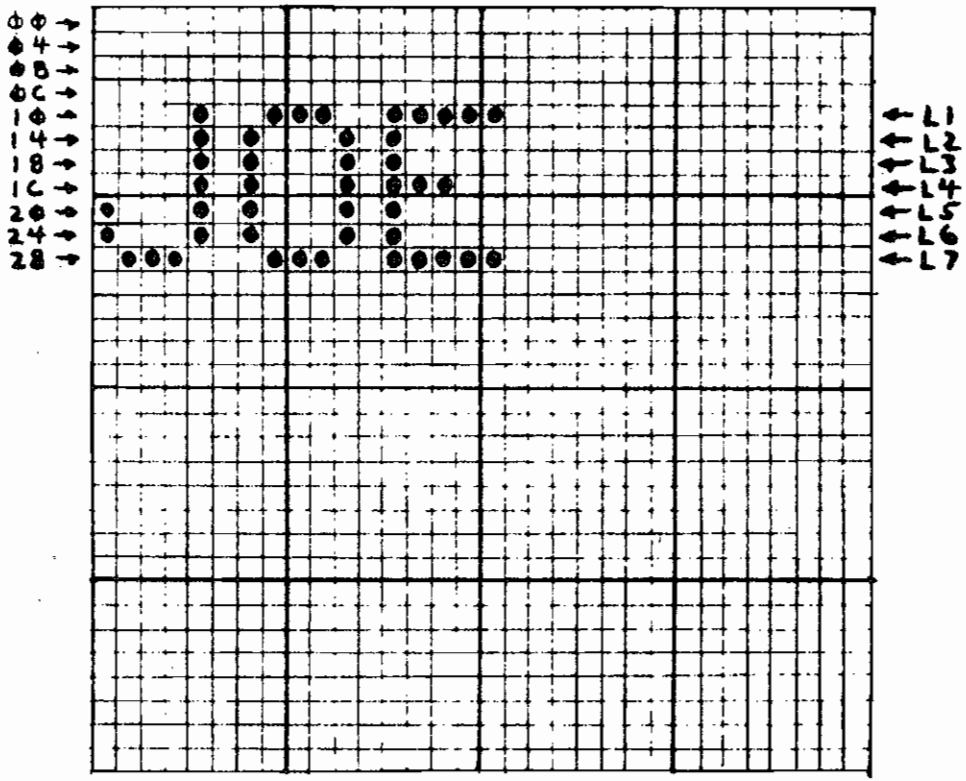
JAW 12/7/71

	S	P	X	M	(m)	INSTRUCTION	comments
card-1	BS	–	–	ΦΦΦΦ	ΦΦ	——	INPUT BYTE STORAGE
	M1	Φ	–	ΦΦΦ1	9Φ	R1(Φ) → D	
		Φ	–	2	B1	D → R1(1)	
		Φ	–	3	B2	D → R1(2)	
		Φ	–	4	B3	D → R1(3)	
		Φ	–	5	B4	D → R1(4)	
		Φ	–	6	A4	D → RΦ(4)	BS → R(4)
		Φ	–	7	A5	D → RΦ(5)	Φ → K
		Φ	–	8	4Φ	M(R(Φ)) → D, R(Φ)+1	
		Φ	–	9	1C	"M2"	
		Φ	–	A	A3	D → RΦ(3)	M2 → R(3)
		Φ	–	B	D3	3 → P	GO TO M2
card-2	5B	–	–	ΦΦΦC	ΦΦ	——	X,P SAVE LOCATION
	5A	–	–	ΦΦΦD	ΦΦ	——	D SAVE LOCATION
	I3	1	2	ΦΦΦE	FΦ	M(R(X)) → D	5A → D
		1	2	F	22	R(2) – 1	
		1	2	ΦΦ1Φ	7Φ	M(R(X)) → XP, R(X)+1	5B → X,P (RETURN), RESET IM
	I1	1	2	ΦΦ11	52	D → M(R(2))	SAVE D @ 5A
		1	2	2	22	R(2) – 1	
		1	2	3	78	T → M(R(X))	SAVE X,P @ 5B
	I2	1	2	ΦΦ14	91	R1(1) → D	
		1	2	5	AΦ	D → RΦ(Φ)	
		1	2	6	41	M(R(1)) → D, R(1)+1	
		1	2	7	Φ1	"Φ1"	
card-3		1	2	8	BΦ	D → R1(Φ)	Φ1ΦΦ → R(Φ)
		1	2	9	12	R(2)+1	
		1	2	A	3Φ	BRANCH	
		1	2	B	ΦE	"I3"	GO TO I3
	M2	3	–	ΦΦ1C	43	M(R(3)) → D, R(3)+1	
		3	–	D	11	"I1"	
		3	–	E	A1	D → RΦ(1)	I1 → R(1)
		3	–	F	43	M(R(3)) → D, R(3)+1	
		3	–	ΦΦ2Φ	ΦD	"5A"	
		3	–	1	A2	D → RΦ(2)	5A → R(2)
		3	–	2	43	M(R(3)) → D, R(3)+1	
		3	–	3	Φ1	"Φ1"	
card-4		3	–	4	BA	D → R1(A)	
		3	–	5	43	M(R(3)) → D, R(3)+1	
		3	–	6	7F	"7F"	
		3	–	7	AA	D → RΦ(A)	Φ17F → R(A)
	C1	3	–	ΦΦ28	93	R1(3) → D	
		3	–	9	5A	D → M(R(A))	ΦΦ → M(R(A))
		3	–	A	8A	RΦ(A) → D	
		3	–	B	32	D = Φ?	
		3	–	C	3Φ	"M3" YES →	SKIP TO M3 (END CLEAR)
		3	–	D	2A	R(A) – 1	
		3	–	E	3Φ	BRANCH	
		3	–	F	28	"C1" →	REPEAT FROM C1

NO. 5B PROGRAM DISPLAY DATE 12/7/71 PAGE 2/2

	S	P	X	M	(m)	INSTRUCTION	comments
card-5	M3	3	-	ØØ3Ø	E3	3 → X	
		3	3	1	61	SELECT DISPLAY	
		3	3	2	Ø2	"Ø2"	
		3	3	3	62	SET DISPLAY = 32 x 32	
		3	3	4	Ø1	"Ø1"	
		3	3	5	7Ø	3 → P, 3 → X, R(X)+1	RESET IM
		3	3	6	33	"33"	
		3	3	7	61	SELECT CARD RDR.	
		3	3	8	Ø1	"Ø1"	
		3	3	9	62	SET IN = PROGRAM MODE	
		3	3	A	Ø1	"Ø1"	
	M4	3	-	ØØ3B	34	EF1 = 1 ?	TEST FOR INPUT BYTE
card-6		3	-	C	3F	"M5"	→ BYTE READY → GO TO M5
		3	-	D	3Ø	BRANCH	
		3	-	E	3B	"M4"	→ REPEAT FROM M4
	M5	3	-	ØØ3F	E4	4 → X	
		3	4	ØØ4Ø	6B	BYTE → M(R(X))	INPUT BYTE → 5B
		3	4	1	B5	RØ(5) → D	
		3	4	2	32	D = Ø ?	
		3	4	3	4A	"M6" YES	→ K = Ø - GO TO M6
	M7	3	4	ØØ44	FØ	M(R(X)) → D	
		3	4	5	5A	D → M(R(X))	BYTE → M(R(X))
		3	4	6	93	R1(3) → D	
		3	4	7	A5	D → RØ(5)	Ø → K
card-7		3	4	8	3Ø	BRANCH	
		3	4	9	3B	"M4"	RETURN TO M4
	M6	3	4	ØØ4A	FØ	M(R(X)) → D	
		3	4	B	AA	D → RØ(A)	
		3	4	C	B3	RØ(3) → D	
		3	4	D	A5	D → RØ(5)	$\overline{Ø}$ → K
		3	4	E	3Ø	BRANCH	
		3	4	F	3B	"M4"	RETURN TO M4
card							

JAW



BYTE-ID

0	0	1	0	1	1	1	0
7	6	5	4	3	2	1	0

2 E

*

TYPICAL DISPLAY

L1 L2 L3 L4 L5

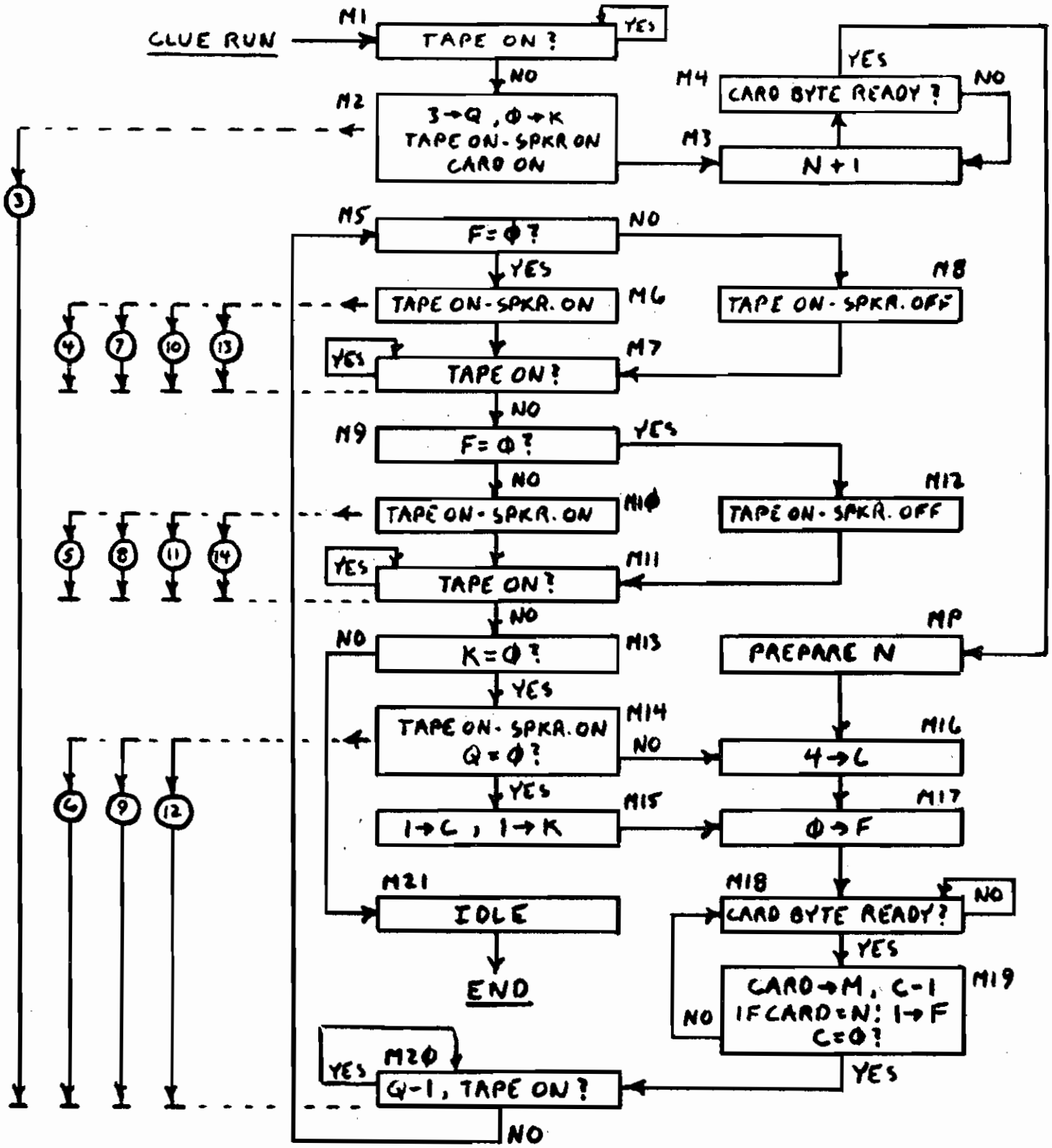
10-09-11-CF-12-80-14-0A-15-28-18-0A-19-28-1C-0A-1D-2E-20-8A-21-28

*

24-8A-25-28-28-71-29-CF-2A-80

L6 L7

JAW



R(Ø) - PROGRAM COUNTER
RØ(3) - QUESTION COUNTER (Q)
RØØ(5) - CARD NUMBER (N)
RØ(6) - ANSWER FLAG (F)
[Ø = NO, 1 = YES]

RØ(7) - K
R(A) - CARD ADDRESS
RØ(4) - CARD COUNT (C)

FLOWCHART C

JAW

NO. SC PROGRAM CLUE DATE 7/15/72 PAGE 1/3

	S	P	X	M	(m)	INSTRUCTION	comments
card	CARD	—	—	ΦΦΦΦ	ΦΦ	——	CARD BYTE INPUT STORAGE
	M1	Φ	—	Φ1	35	TAPE ON?	←
		Φ	—	2	Φ1	"M1" YES	→
	M2	Φ	—	Φ3	9Φ	R1(Φ) → D	
		Φ	—	4	AA	D → RΦ(A)	
		Φ	—	5	BA	D → R1(A)	ΦΦΦΦ → R(A)
		Φ	—	6	A7	D → RΦ(7)	Φ → K
		Φ	—	7	4Φ	M(R(Φ)) → D, R(Φ)+1	
		Φ	—	8	Φ3	"Φ3"	
		Φ	—	9	A3	D → RΦ(3)	3 → Q
		Φ	—	A	EΦ	Φ → X	
		Φ	Φ	B	63	TAPE ON - SPKR. ON	
card		Φ	Φ	C	Φ3	"Φ3"	
		Φ	Φ	D	61	SELECT CARD	
		Φ	Φ	E	Φ1	"Φ1"	
		Φ	Φ	F	62	SET PROG. MODE	
		Φ	Φ	ΦΦ1Φ	Φ1	"Φ1"	
	M3	Φ	—	11	15	R(5)+1	N+1
	M4	Φ	—	12	34	EF1=1?	
		Φ	—	3	5A	"MP" YES →	→ GO TO MP
		Φ	—	4	3Φ	REPEAT FROM	
		Φ	—	5	11	"M3" →	→ M3
	M5	Φ	—	16	B6	RΦ(6) → D	
		Φ	—	7	31	D≠Φ?	F≠Φ?
card		Φ	—	8	2Φ	"ME" YES →	→ F≠Φ → GO TO MB
	M6	Φ	—	19	EΦ	Φ → X	
		Φ	Φ	A	63	TAPE ON - SPKR. ON	
		Φ	Φ	B	Φ3	"Φ3"	
	M7	Φ	Φ	1C	35	TAPE ON?	←
		Φ	Φ	D	1C	"M7" YES	→
		Φ	Φ	E	3Φ	GO TO	
		Φ	Φ	F	25	"M9" →	→ GO TO M9
card	M8	Φ	—	ΦΦ2Φ	EΦ	Φ → X	
		Φ	Φ	1	63	TAPE ON - SPKR OFF	
		Φ	Φ	2	Φ1	"Φ1"	
		Φ	Φ	3	3Φ	GO TO	
		Φ	Φ	4	1C	"M7" →	→ GO TO M7
card	M9	Φ	Φ	25	B6	RΦ(6) → D	
		Φ	Φ	6	32	D=Φ?	F=Φ?
		Φ	Φ	7	2E	"M12" YES →	→ F=Φ → GO TO M12
	M1Φ	Φ	Φ	28	63	TAPE ON - SPKR. ON	
		Φ	Φ	9	Φ3	"Φ3"	
	M11	Φ	Φ	2A	35	TAPE ON?	←
		Φ	Φ	B	2A	"M11" YES	→
		Φ	Φ	C	3Φ	GO TO	
		Φ	Φ	D	32	"M13" →	→ GO TO M13
	M12	Φ	Φ	2E	63	TAPE ON - SPKR OFF	
		Φ	Φ	F	Φ1	"Φ1"	

JAW

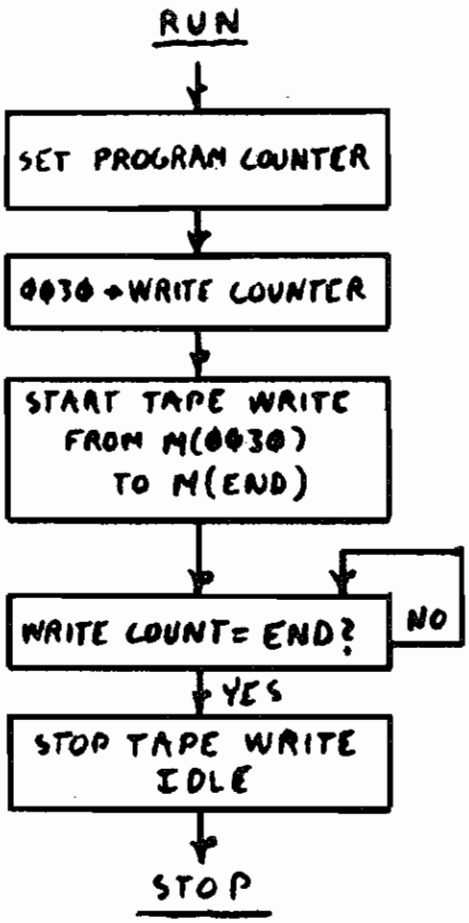
	S	P	X	M	(m)	INSTRUCTION	comments
card		3	3	ØØ3Ø	3Ø	GO TO	
		Ø	Ø	1	2A	"M11" →	→ GO TO M11
	M13	Ø	Ø	32	87	RØ(7) → D	
		Ø	Ø	3	31	D ≠ Ø?	K ≠ Ø?
		Ø	Ø	4	63	"M21" YES →	→ K ≠ Ø → GO TO M21
	M14	Ø	Ø	35	63	TAPE ON - SPKR ON	
		Ø	Ø	6	Ø3	"Ø3"	
		Ø	Ø	7	83	RØ(3) → D	
		Ø	Ø	8	31	D ≠ Ø?	Q ≠ Ø?
		Ø	Ø	9	4Ø	"M16" YES →	→ Q ≠ Ø → GO TO M16
	M15	Ø	Ø	3A	4Ø	M(R(Ø)) → D, R(Ø)+1	
		Ø	Ø	B	Ø1	"Ø1"	
card		Ø	Ø	C	A4	D → RØ(4)	1 → C
		Ø	Ø	D	A7	D → RØ(7)	1 → K
		Ø	Ø	E	3Ø	GO TO	
		Ø	Ø	F	43	"M17" →	→ GO TO M17
	M16	Ø	—	ØØ4Ø	4Ø	M(R(Ø)) → D, R(Ø)+1	
		Ø	—	1	Ø4	"Ø4"	
		Ø	—	2	A4	D → RØ(4)	4 → C
	M17	Ø	—	43	9Ø	R1(Ø) → D	
		Ø	—	4	A6	D → RØ(6)	Ø → F
	M18	Ø	—	45	EA	A → X	
		Ø	A	6	34	EF1 = 1?	
		Ø	A	7	4A	"M19" YES →	→ GO TO M19
card		Ø	A	8	3Ø	REPEAT	
		Ø	A	9	45	"M18" →	→ GO TO M18
	M19	Ø	A	4A	68	CARD → M(R(X))	
		Ø	A	B	24	R(4) - 1	C - 1
		Ø	A	C	85	RØ(5) → D	
		Ø	A	D	F3	M(R(X)) ⊕ D → D	
		Ø	A	E	31	D ≠ Ø?	
		Ø	A	F	53	"53" YES →	→ CARD ≠ N
card		Ø	A	ØØ5Ø	4Ø	M(R(Ø)) → D, R(Ø)+1	
		Ø	A	1	Ø1	"Ø1"	
		Ø	A	2	A6	D → RØ(6)	1 → F
		Ø	A	53	84	RØ(4) → D	
		Ø	A	4	31	D ≠ Ø?	
		Ø	A	5	45	"M18" YES →	→ C ≠ Ø → GO TO M18
	M2Ø	Ø	A	56	35	TAPE ON?	
		Ø	A	7	56	"M2Ø" YES →	
		Ø	A	8	3Ø	GO TO	
		Ø	A	9	65	"2ØA" →	→ GO TO M2ØA
	MP	Ø	—	5A	9Ø	R1(Ø) → D	
		Ø	—	B	AF	D → RØ(F)	
		Ø	—	C	85	RØ(5) → D	
		Ø	—	D	CF	DØ → RØØ(F)	
		Ø	—	E	8F	RØ(F) → D	
		Ø	—	F	F6	SHIFT D RIGHT	

JAW

NO. SC PROGRAM CLUE DATE 7/15/72 PAGE 3/3

	S	P	X	M	(m)	INSTRUCTION	comments
card		Ø	—	ØØ6Ø	A5	D → RØ(5)	
		Ø	—	1	3Ø	GO TO	
		Ø	—	2	4Ø	"M16"	→ GO TO M16
	M21	Ø	—	63	ØØ	IDLE	
				4	FF	——	
	M20A			65	23	R(3) - 1	Q - 1
				66	3Ø	GO TO	
				67	16	"M5"	→ GO TO M5
card							
card							
card							
card							

JAW



REGISTERS

R(0) - WRITE COUNTER
R(3) - PROGRAM COUNTER
R(4) - END POINTER

CONNECTIONS

FRED JACK C
TO RECORDER LINE/AUX.
RECORD INPUT.

TAPE WRITE PROCEDURE:

1. LOAD WRITE PROGRAM [STARTING @ M(0060)]
2. ENTER 2 BYTE END ADDRESS
3. LOAD DATA/PROGRAM TO BE RECORDED ON TAPE
   [WILL START AT M(0030)]
4. RESET
5. POSITION TAPE AND SET TO RECORD [TAPE WILL RUN]
6. PRESS RUN : M(0030) TO M(END-2) WILL
   BE RECORDED ON TAPE.

FLOWCHART D

JAW

NO. SD PROGRAM TAPE WRITE DATE 2/7/72 PAGE 1/1

	S	P	X	M	(m)	INSTRUCTION	comments
card		—	—	ΦΦΦΦ	ΦΦ	——	NOT USED
	M1	Φ	—	1	90	R1(Φ) → D	
		Φ	—	2	B3	D → R1(3)	
		Φ	—	3	B4	D → R1(4)	
		Φ	—	4	4Φ	M(R(Φ)) → D, R(Φ)+1	
		Φ	—	5	ΦB	"ΦB"	
		Φ	—	6	A3	D → RΦ(3)	ΦΦ "M2" → PROG. COUNTER
		Φ	—	7	D3	3 → P	→ GO TO M2
	M2	3	—	ΦB	43	M(R(3)) → D, R(3)+1	
		3	—	9	(3Φ)	(23) "3Φ"	
		3	—	A	AΦ	D → RΦ(Φ)	ΦΦ3Φ → WRITE COUNTER
		3	—	B	43	M(R(3)) → D, R(3)+1	
card		3	—	C	(2E)	(21) "EA1"	
		3	—	D	A4	D → RΦ(4)	"EA1" → END POINTER
	M3	3	—	ΦE	E3	3 → X	
		3	3	F	61	SELECT TAPE	
		3	3	ΦΦ1Φ	Φ3	"Φ3"	
		3	3	1	62	START WRITE	
		3	3	2	4Φ	"4Φ"	START TAPE WRITE
		3	3	3	E4	4 → X	
	M4	3	4	14	90	R1(Φ) → D	
		3	4	5	F3	M(R(X)) ⊕ D → D	
		3	4	6	31	D ≠ Φ ?	
		3	4	7	14	"M4" YES →	WRITE COUNTER ≠ END → M4
card		3	4	8	14	R(4)+1	
		3	4	9	80	RΦ(Φ) → D	
		3	4	A	F3	M(R(X)) ⊕ D → D	
		3	4	B	32	D = Φ ?	
		3	4	C	2Φ	YES →	WRITE COUNTER = END → M5
		3	4	D	24	R(4)-1	
		3	4	E	3Φ	REPEAT	
		3	4	F	14	"M4" →	GO TO M4
card	M5	3	4	ΦΦ2Φ	(E3)	3 → X	ΦΦ
		3	3	21	(62)	STOP WRITE	} END. ADDR EA1 03
		3	3	22	(ΦΦ)	"ΦΦ"	} EA2 FF
		3	3	3	63	STOP TAPE	← 1ST PROG. BYTE TO BE WRITTEN
card		3	3	4	2Φ	"2Φ"	
		3	3	5	2Φ	R(Φ)-1	
		3	3	6	ΦΦ	IDLE	
		—	—	7	ΦΦ	——	} "FILL" BYTES
		—	—	8	ΦΦ	——	
		—	—	9	ΦΦ	——	
		—	—	A	ΦΦ	——	
		—	—	B	ΦΦ	——	
		—	—	C	ΦΦ	——	
		—	—	D	ΦΦ	——	
	EA1	—	—	ΦΦ2E	—		} END WRITE ADDRESS
	EA2	—	—	ΦΦ2F	—		

JAW

Internal Correspondence

RCA

To Microprocessor Group Location Date December 27, 1972

From A. Gonzalez Location E-201C Telephone 3231

Subject New Microprocessor Memory Board (6K bytes)

The original FRED Microprocessor memory board was implemented using the 1101 Memory Chip (256x1), providing 1024-8 bit bytes of storage.

The modified version shown on Figure 1 uses the SIGNETICS 2602B Memory Chip (1024x1). The board has a capacity 6144-8 bit bytes, although the present FRED Microprocessor wiring allows for only 4K bytes to be addressed.

Both the 1101 and 2602 chips are fully decoded static MOS RAMs.

Angel Gonzalez

Angel Gonzalez
/ck

Distribution:
FRED Distribution
C. Haney, WPB

MASP NOT IN
ADD FOLLOWING WIRE
FOR FRED
P2-27 TO M1-30

M1 - 6K

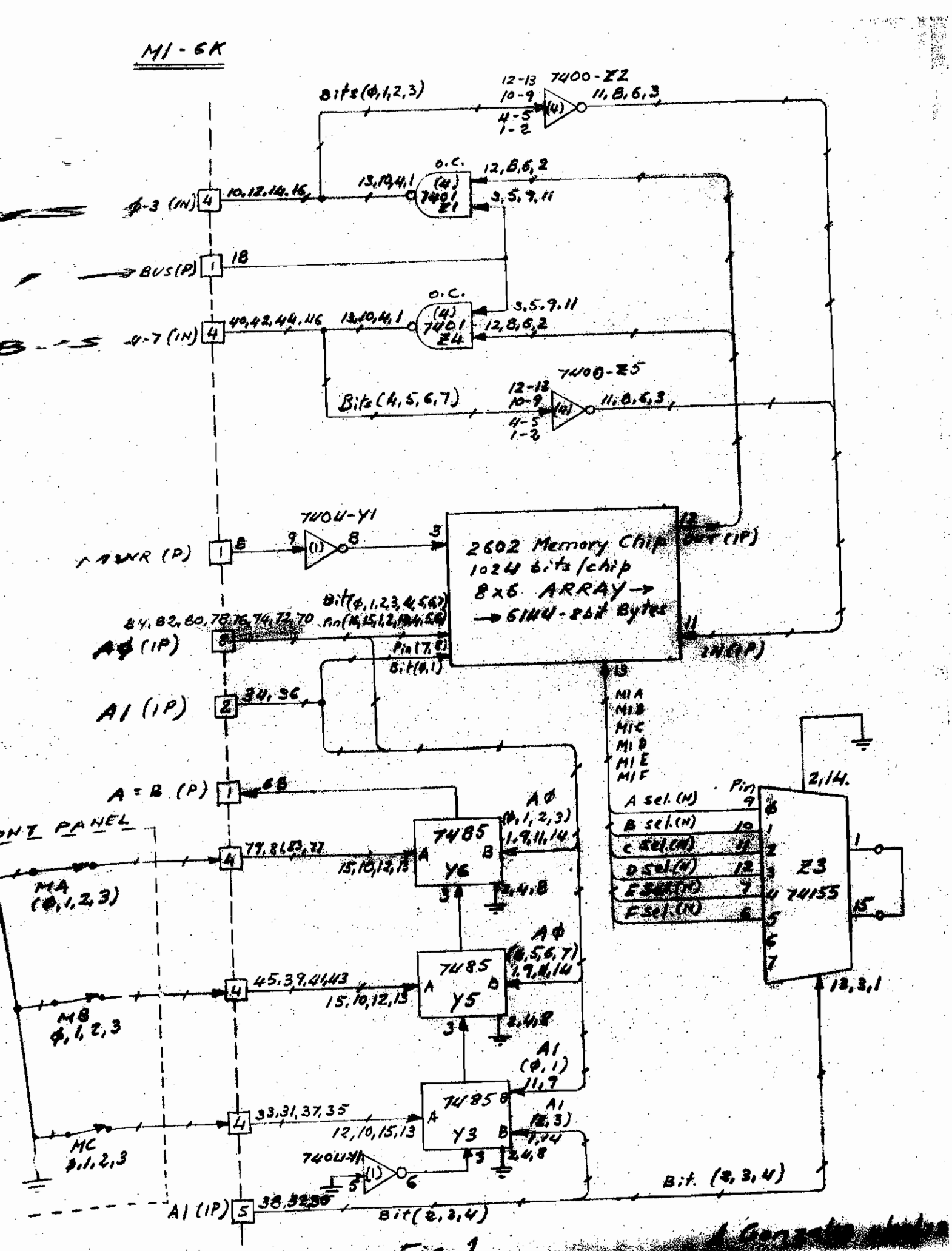

Bits(0,1,2,3)
12-13
10-9
4-5
1-2
7400-Z2
11,8,6,3
0.C.
12,8,6,2
13,10,4,1
7401 Z1
3,5,9,11
10,12,14,16
BUS(P)
18
7401 Z4
40,42,44,46
Bits(4,5,6,7)
7400-Z5
7404-Y1
2602 Memory Chip
1024 bits/chip
8x6 ARRAY →
→ 6144-8bit Bytes
Bit(0,1,2,3,4,5,6,7)
84,82,80,78,76,74,72,70
Pin(17,8)
Bit(0,1)
34,36
MIA
MIB
MIC
MID
MIE
MIF
A sel.(N)
B sel.(N)
C sel.(N)
D sel.(N)
E sel.(N)
F sel.(N)
Pin
2,14.
Z3
74155
13,3,1
A=B (P)
66
7485 Y6
7485 Y5
7485 Y3
77,81,83,87
45,39,41,43
33,31,37,35
15,10,12,13
MA (0,1,2,3)
MB 0,1,2,3
MC 0,1,2,3
7404-Y4
38,32,36
Bit(2,3,4)
Bit. (2,3,4)

Fig. 1

M1 - 6K

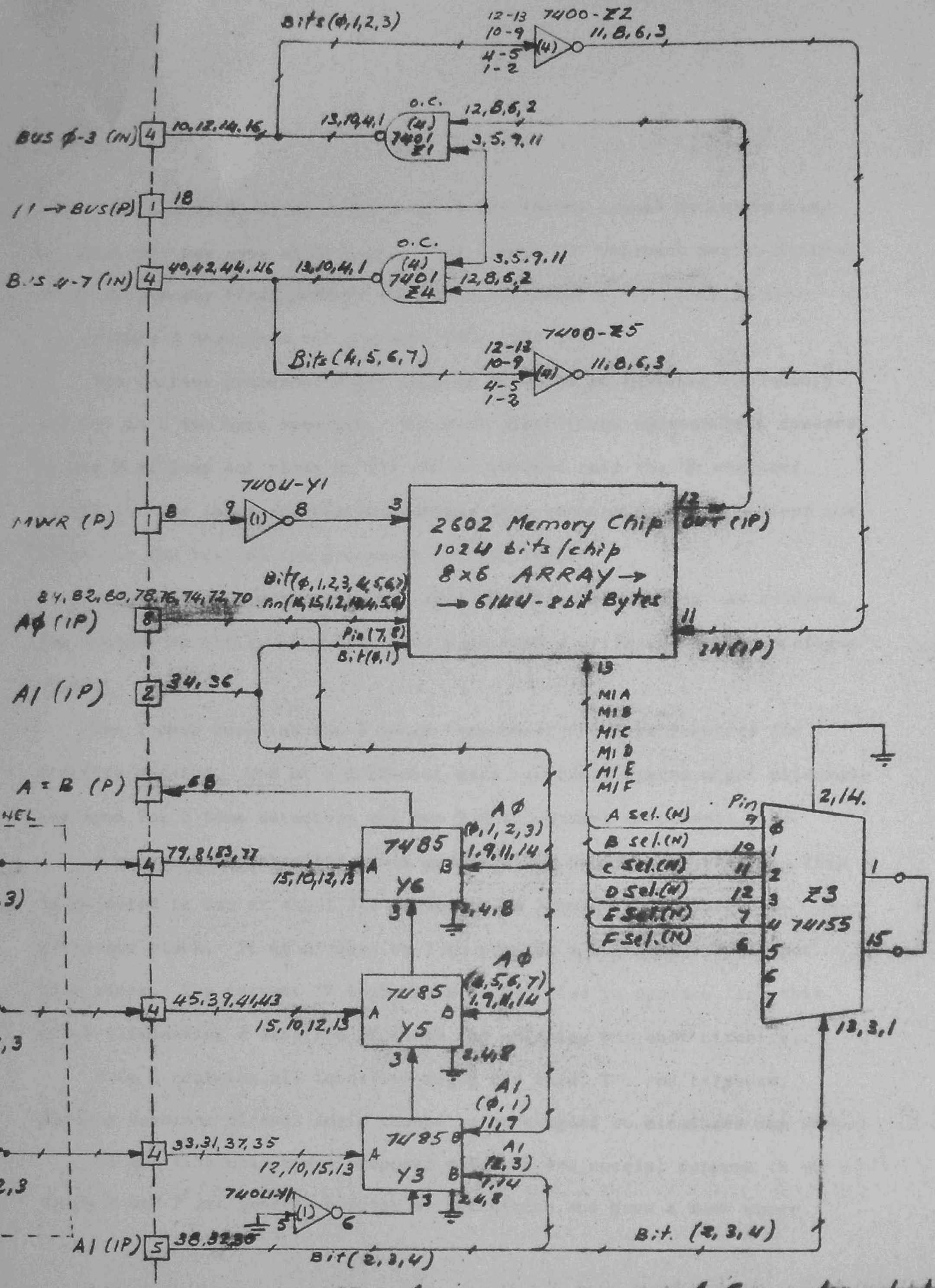


Fig. 1

A. Gonzalez sketch

Internal Correspondence

RCA

To Distribution Location Date August 14, 1972

From J. A. Weisbecker Location E-201A Telephone 3325

Subject Six Chip Design Approach

FRED NOTE #1

The current FRED model is an experimental version meant only for application development and testing. It is not a physical prototype of a final product design. A FRED design suitable for a final product is outlined here together with target costs.

Figure 1 illustrates the manner in which FRED could be fabricated using only 6 LSI chips (5 types).

Figure 2 lists what are felt to be reasonable 1976 cost targets for a final product. These are, of course, very sensitive to volume. While no attempt to project volume is made here, the following factors should be considered:

A. Magnavox is projecting 100,000 units this year for its new $100 electronic TV game called odyssey in the consumer market.

B. 10% penetration of schools would result in sales of 20,000 units @2/school.

C. It has been projected that 35% of all high schools will have computers for educational purposes by 1975.

FRED might also be packaged as an integral part of a standard TV set. This would permit consumer advertising, distribution, and servicing

costs to be shared, eliminating much of the trauma caused by introducing a completely new type of product. This "smart TV" approach merits further study before any final product decisions are made.

Figure 3 describes the proposed FRED chip set.

The current processor logic must be modified to transmit the memory address as a two byte sequence. The most significant address byte appears on the M address out lines at TP1 and is clocked into the MA register (X chip). The least significant address byte remains on the M address out lines for the rest of the processor machine cycle.

A 1024x4 bit memory chip with self contained bus gating was assumed. The design can easily be modified to cope with a different M chip configuration.

The T chip contains the 3 audio tone burst decoders required for cassette reading. Use of a different data recording system might eliminate the need for 2 tone detectors and the R/C adjustment components.

A master R-C controlled clock generator is provided on chip X. This is adjusted to run at about 3.3 megahertz to provide a 300 ns/cycle processor clock. It is divided by 2 to provide a 600 ns/cycle TV spot time clock. The current TV logic must be modified to operate from this clock eliminating a separate TV clock and existing one shot circuits.

Chip E contains all interface logic for card, TV, and cassette. Current versions of this logic should be redesigned to eliminate one shots.

Of the five chip types proposed only two are special purpose (X and E). Chips M and P are general purpose by definition and have a much wider

August 14, 1972
J. A. Weisbecker
FRED NOTE #1

market than just the FRED system. Chip T can be designed as a general purpose triple phase locked loop device. Chip X is unique to FRED but is only a 16 pin MSI unit requiring minimum development effort. Chip E is the only unique LSI chip required.

J. A. Weisbecker

/ck

Distribution:

R. O. Winder
N. L. Gordon
P. M. Russo
A. D. Robbi
A. R. Marcantonio
B. J. Call

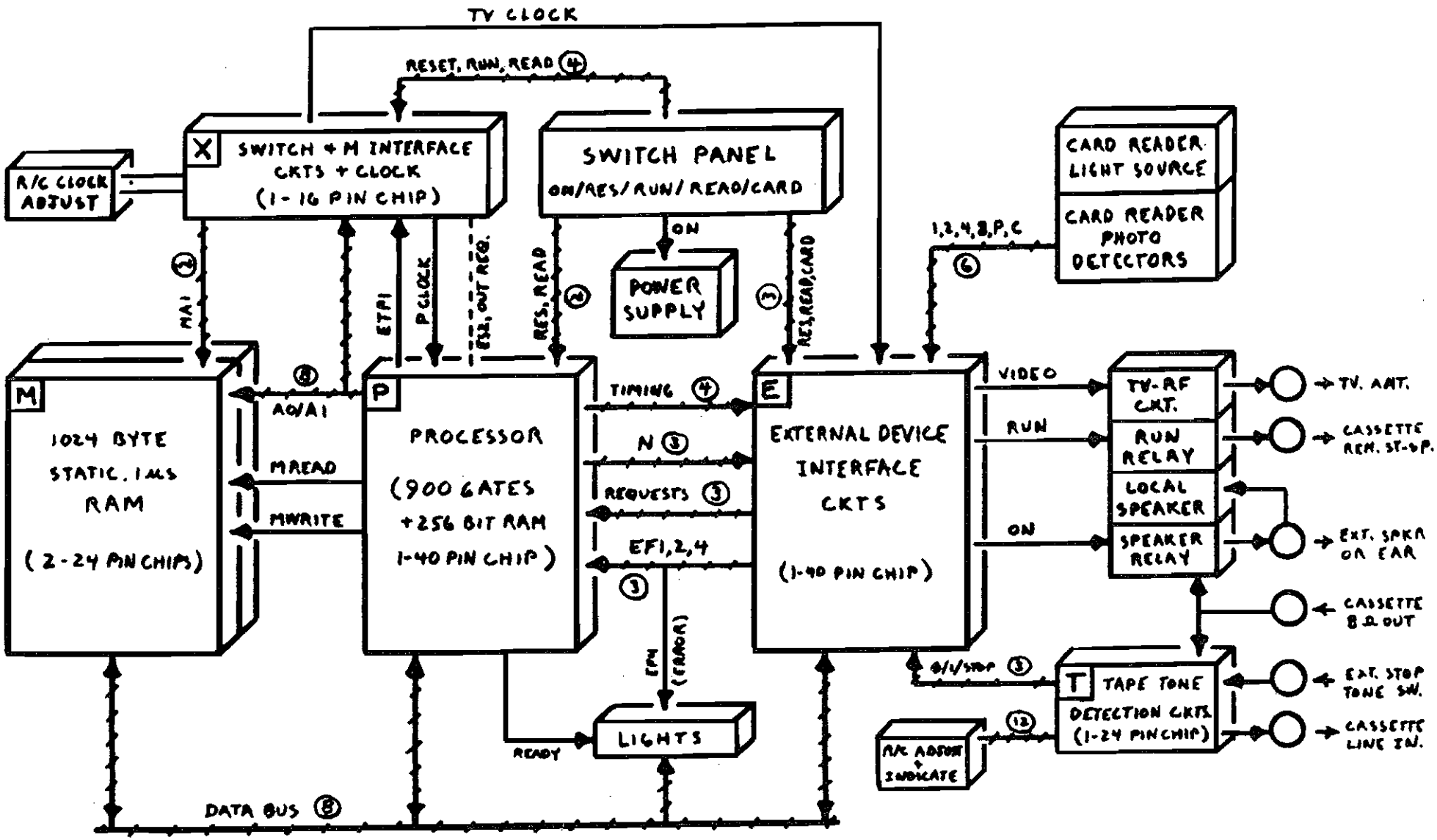


FIGURE 1 - 6 CHIP "FRED" DESIGN

JAW 8-11-72

CHIP COSTS	MIN	MAX
M x2	6.0	14.0
X x1	2.0	3.0
P x1	5.0	12.0
E x1	5.0	15.0
T x1	2.0	3.0
TOT. CHIP	20.0	47.0
MISC. PARTS		
CABINET (PLASTIC)	1.3	3.0
5 SWITCHES	1.0	2.5
1 SPEAKER	.4	1.0
11 LIGHTS	1.1	2.8
6 PHOTODETECTORS	1.2	3.0
8 CAPACITORS	.8	1.6
1 POT.	.8	1.0
4 TRIMMERS	.8	1.6
CARD LIGHT SOURCE	.9	2.0
6 JACKS	.9	1.5
POWER SUPPLY	3.0	6.0
P.C. CARD	.8	2.0
RF CIRCUIT	1.0	2.0
2 RELAYS	1.0	2.0
TOT. MISC.	15.0	32.0
TOTAL PARTS	35.0	78.0
ASSEMBLY & TEST	15.0	30.0
TOTAL UNIT COST	50.0	108.0

FIGURE 2- TARGET "FRED" COSTS

JAW 8-11-72

P- 40 PIN PROCESSOR CHIP:

1- 2	POWER	
3-10	M ADRESS OUT (A0/A1)	M INTERFACE
11-13	ETP1,MREAD,MWRITE	M INTERFACE
14-21	8 BIT DATA BUS (IN/OUT)	M INTERFACE
22-25	I/O FLAG BUS (EF1,2,3,4)	I/O INTERFACE
26-28	N BITS 0,1,2 OUT	I/O INTERFACE
29-31	IN/OUT/INTERRUPT REQUEST BUS	I/O INTERFACE
32-35	I/O TIMING (ETP6,EI6,ES2,ES3)	I/O INTERFACE
36-40	RESET,PCLOCK,READY,READ,SPARE	I/O INTERFACE

M- 24 PIN STATIC 4096 BIT RAM:

1- 2	POWER
3-12	10 BIT M ADDRESS (1/1024-4 BIT WORDS)
13-16	4 BIT DATA BUS (IN/OUT)
17-19	MREAD,MWRITE,CHIP ENABLE

X- 16 PIN SWITCH & M INTERFACE + CLOCK:

1- 2	POWER
3- 4	MA1 REGISTER OUT (BIT 0,1)
5- 6	ETP1,ES2 INPUTS
7-10	RUN0/1,READ,RESET SWITCH INPUTS
11-13	PCLOCK,TVCLOCK,OUT REQUEST (OUTPUTS)
14-16	EXT. R/C CLOCK ADJUST,SPARE

E- 40 PIN EXTERNAL DEVICE (I/O) INTERFACE:

1- 2	POWER
3- 8	1,2,4,8,P,C PHOTOCELL INPUTS
9-13	CASSETTE RUN,SPKR ON,0/1/STOP TONE INPUTS
14-21	DATA BUS (IN/OUT)
22-25	I/O FLAGS (EF1,2,3,4)
26-28	N BITS 0,1,2 IN
29-31	IN,OUT,INTERRUPT REQUESTS
32-35	TIMING (ETP6,EI6,ES2,ES3)
36-40	RESET,READ,CARD,VIDEO OUT,TVCLOCK IN

T- 24 PIN TAPE TONE DETECTOR:

1- 2	POWER
3- 5	INPUT,EXT. SWITCH,RECORD OUTPUT
6- 9	0,1,STOP TONE DETECTOR OUTPUTS
10-24	EXTERNAL R/C DETECTOR ADJUST,SPARES

FIGURE 3-"FRED" CHIP SET

JAW 8-11-72

Internal Correspondence

RCA

To	Distribution	Location		Date	August 14, 1972
From	J. A. Weisbecker	Location	E-201A	Telephone	3325

Subject Logic Tracing Aids

FRED NOTE #2

A detailed set of FRED system logic diagrams are provided in the FRED manual. This note provides two aids for logic tracing. These will be useful for debugging, maintenance, or design changes.

The FRED system diagram is keyed to individual logic diagrams (L13, L6, etc.) and plug-in cards (P3, E1, etc.). This is most useful for debugging or maintenance.

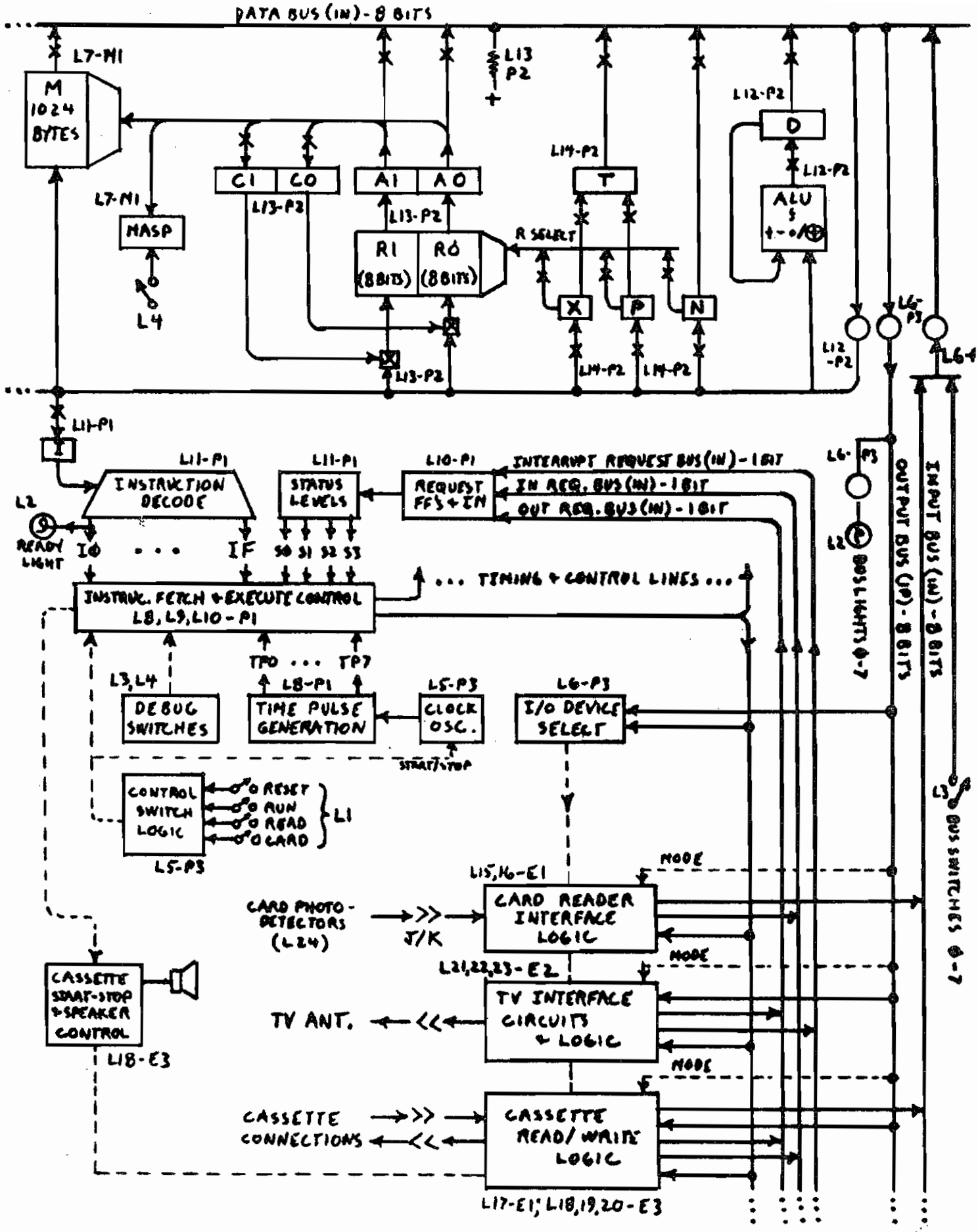
The second aid is a table listing control pulses and levels generated for each unique machine cycle. This is invaluable for understanding the control logic (mainly card P1) and will be useful in any subsequent redesign.

J. A. Weisbecker

/ck

Distribution:

R. O. Winder
N. L. Gordon
P. M. Russo
A. D. Robbi
A. R. Marcantonio
B. J. Call



FRED - SYSTEM DIAGRAM

NOTES: A→C @ TP2, * = DON'T CARE CASE
* UNLESS EXT. REQ BY TP7 OCCURS
** UNLESS READ SWITCH IS ON

		S	→BUS TP0-7	→R 0-7	R SEL. 0-7	R INH. 0-7	R INH. 5-6	R→A TP1	C TP3	MWR TP4&5	→R (RWR) TP6	F 0-7	D TP6	D 0-7	MSC	NEXT T#7	
N	IDLE	1	M	C *	R(N)	—	—	R(N)**	—	—	—	*	—	—		S1 *	STAYS IN S1 UNTIL I/O REQ.
N	R(N)+1	1	—*	C	R(N)	—	—	R(N)	+1	—	C	*	—	—		S0 *	
N	R(N)-1	1	—*	C	R(N)	—	—	R(N)	-1	—	C	*	—	—		S0 *	
N	COND=0: R(P)+1	1	M*	C	R(P)	—	—	R(P)	+1	—	C	*	—	—		S0 *	
	COND=1: M(R(P))→RØ(P)	1	M	C/BUS	R(P)	—	R1	R(P)	RES	—	BUS	*	—	—		S0 *	BRANCH WITHIN 256 BYTE PAGE
N	M(R(N))→D, R(N)+1	1	M	C	R(N)	—	—	R(N)	+1	—	C	BUS	F→D	INH N		S0 *	
N	D→M(R(N))	1	D	C *	R(N)	—	—	R(N)	—	→M(A)	—	*	—	—		S0 *	
N	N=0KKK: M(R(X)), KKK→OUT; R(X)+1	1	M	C	R(X)	—	—	R(X)	+1	—	C	*	—	—		S0 *	DATA/CONTROL BYTE OUT
	N=1KKK: IN→M(R(X)), KKK→OUT	1	—	C *	R(X)	—	—	R(X)	—	→M(A)	—	*	—	—		S0 *	DATA/STATUS BYTE IN
N	N=0KKK: M(R(X))→PX, R(X)+1, RESET IM	1	M	C	R(X)	—	—	R(X)	+1	—	C	*	—	—	BUS→PX(7) RES. IM(4)	S0 *	RESTORE AFTER INTERRUPT
	N=1KKK: T→M(R(X))	1	T	C *	R(X)	—	—	R(X)	—	→M(A)	—	*	—	—		S0 *	INTERRUPT SAVE
N	RØ(N)→D	1	AØ	C *	R(N)	—	—	R(N)	—	—	—	BUS	F→D	INH N		S0 *	
N	R1(N)→D	1	A1	C *	R(N)	—	—	R(N)	—	—	—	BUS	F→D	INH N		S0 *	
N	D→RØ(N)	1	D	C/BUS	R(N)	R1	—	RØ(N)*	RES	—	BUS	*	—	—		S0 *	
N	D→R1(N)	1	D	C/BUS	R(N)	RØ	—	R1(N)*	RES	—	BUS	*	—	—		S0 *	
N	DØ→RØØ(N)	1	D	C/BUS	R(N)	R1,RØ1	—	RØØ(N)*	RES	—	BUS	*	—	—		S0 *	
N	N→P	1	N	C *	R(N)*	—	—	R(N)*	—	—	—	*	—	—	BUSØ→P (TP7)	S0 *	
N	N→X	1	N	C *	R(N)*	—	—	R(N)*	—	—	—	*	—	—	BUS1→X (TP7)	S0 *	
N	N=0: M(R(X))→D	1	M	C *	R(X)	—	—	R(X)	—	—	—	BUS	F→D	—		S0 *	
	N=1: M(R(X))/D→D (OR)	1	M	C *	R(X)	—	—	R(X)	—	—	—	/	F→D	—		S0 *	
	N=2: M(R(X))•D→D (AND)	1	M	C *	R(X)	—	—	R(X)	—	—	—	•	F→D	—		S0 *	
	N=3: M(R(X))⊕D→D (EXCL. OR)	1	M	C *	R(X)	—	—	R(X)	—	—	—	⊕	F→D	—		S0 *	
	N=4: M(R(X))+D→D (ADD)	1	M	C *	R(X)	—	—	R(X)	—	—	—	(+)	F→D	—	CB→DF (TP4)	S0 *	FINAL CARRY→DF
	N=5: M(R(X))-D→D (SUBTRACT)	1	M	C *	R(X)	—	—	R(X)	—	—	—	(—)	D→D	—	CB→DF (TP4)	S0 *	FINAL CARRY→DF
	N=6: SHIFT RIGHT D→D (SHIFT)	1	M*	C *	R(X)*	—	—	R(X)*	—	—	—	BUS*	D→D	SHIFT	DBIT0→DF (TP4)	S0 *	D BIT Ø→DF
INSTR. FETCH	M(R(P))→I-N, R(P)+1	Ø	M	C	R(P)	—	—	R(P)	+1	—	C	*	—	—		S1	DECODE I, SELECT S1
DMA REQUEST	BUS→M(R(Ø)), R(Ø)+1	2	—	C	R(Ø)	—	—	R(Ø)	+1	→M(A)	C	*	—	—		SØ	DIRECT INPUT XFER
DMA REQUEST	M(R(Ø))→BUS, R(Ø)+1	2	M	C	R(Ø)	—	—	R(Ø)	+1	—	C	*	—	—		SØ	DIRECT OUTPUT XFER
INTERRUPT REQ.	PX→T, 1→P, 2→X, SET IM	3	"21"	C *	R(N)*	—	—	R(N)*	+1*	—	—	*	—	—	AX→T (N) BUS→PX(7)	SØ	SET IM @ TP4

Internal Correspondence RCA

To	Distribution	Location		Date	August 14, 1972
From	J. A. Weisbecker	Location	E-201A	Telephone	3325
Subject	Use of The Type 62 Instruction				

FRED NOTE #3

The FRED manual states that a 62 instruction with M(R(X))=ØØ will turn a selected I/O device off. This is misleading when the cassette (tape) has been selected. In this case the tape read/write operation will be terminated but the "tape run" and "speaker on" switches will be set by the 62 instruction.

J. A. Weisbecker

/ck

Distribution:

R. O. Winder
N. L. Gordon
P. M. Russo
A. D. Robbi
A. R. Marcantonio
B. J. Call

Internal Correspondence

RCA

To Distribution Location Date August 17, 1972

From J. A. Weisbecker Location E-201A Telephone 3325

Subject Cassette "Tuning" Modification

FRED NOTE #4

This note describes a FRED design modification which permits "tuning" FRED to compensate for differences in speeds between various cassette recorders.

An adjustment control and tuning light are provided on the debug panel (located above the manual switch and light shown on Figure 11 in the FRED manual). The adjustment control permits a simultaneous +/-10% variation in the three tone detector center frequencies (EØ, E1, E2 in logic 17).

This control should normally be left at its center position (50). For certain tapes and/or cassette recorders it may be necessary to reposition this control toward 0/100 to compensate for wider than normal variations in cassette speeds. This "tuning" process is facilitated by the tuning light. The adjustment control is set to produce maximum light brightness (minimum flicker) while reading data. If desired an external meter may be attached via jack "D" and the adjustment control set for maximum meter deflection. The meter should be set to read the average value of the light wave form. This average value will decrease as the difference between tape Ø/1 frequencies and tone detector center frequencies increase.

When recording data/stop tones the adjustment control should always be set to its center position (50 on the dial).

J. A. Weisbecker

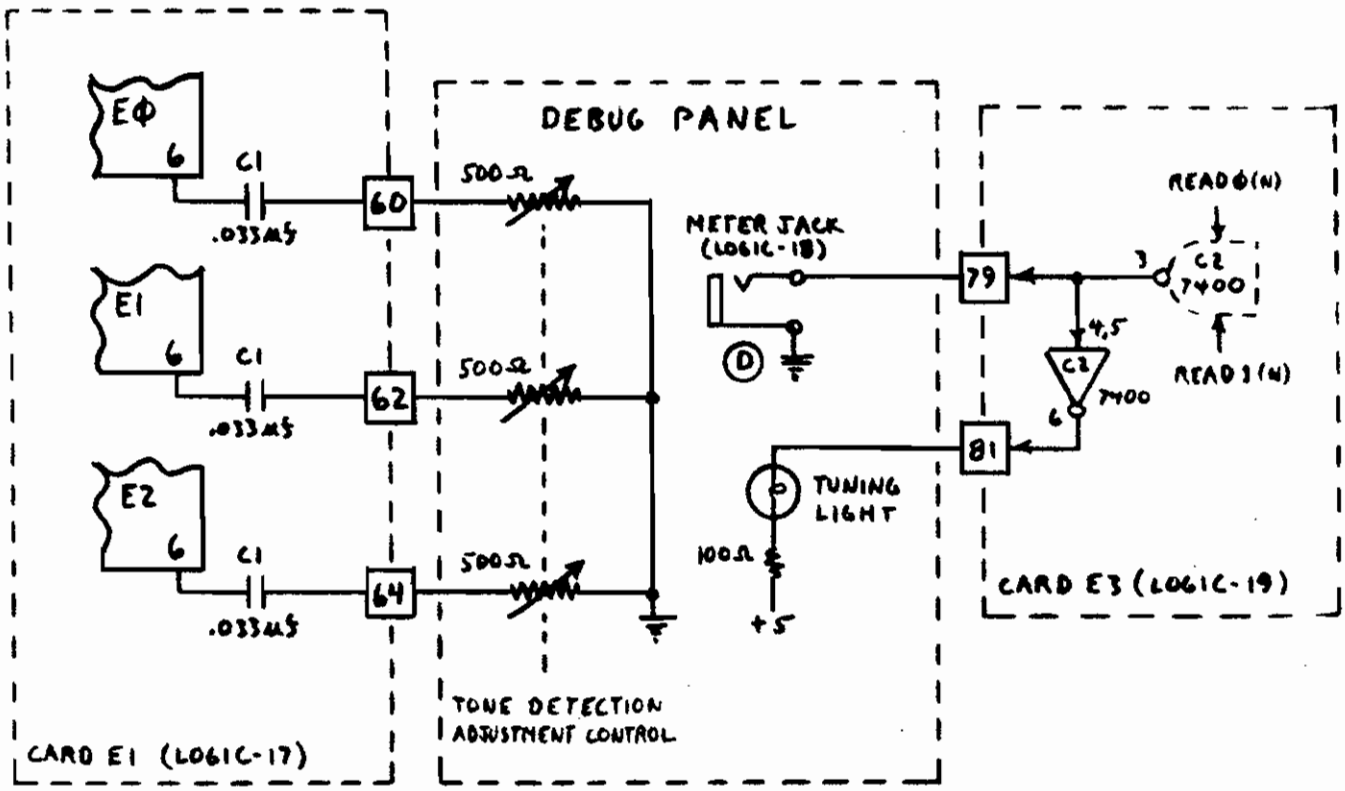


FIGURE 1- CKT. MODIFICATIONS

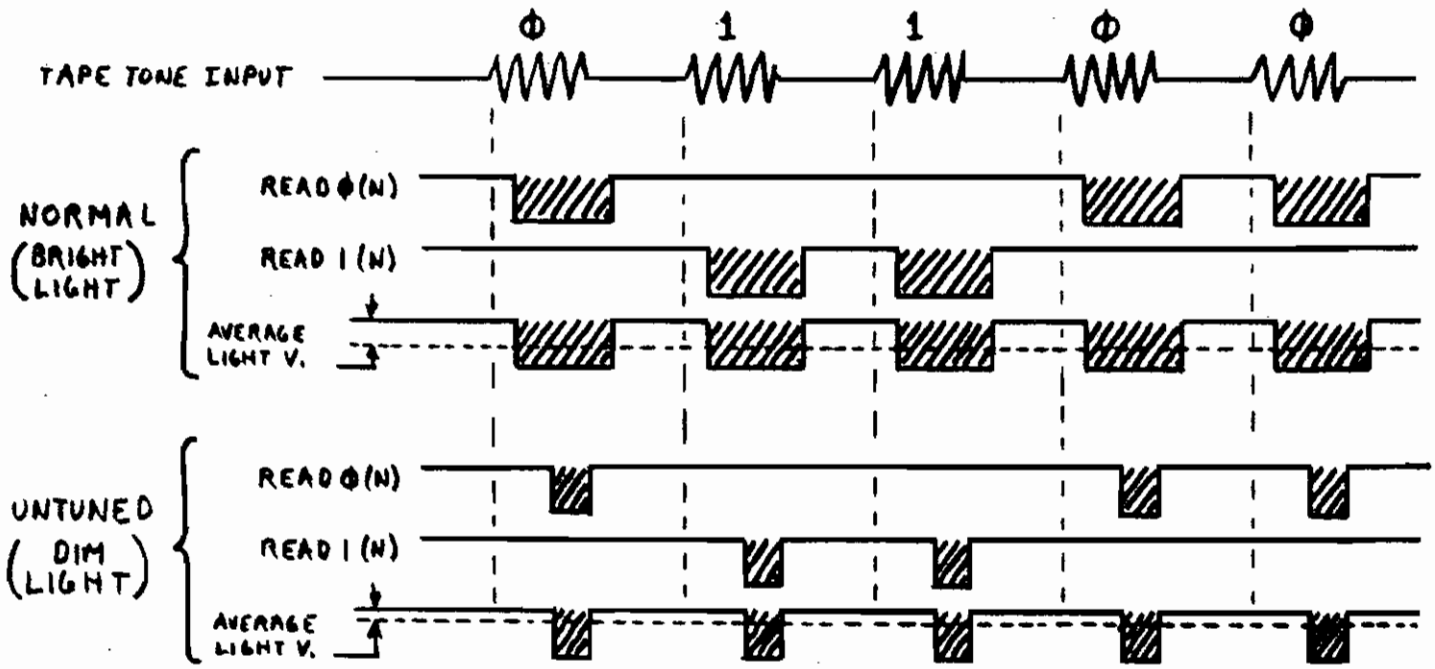


FIGURE 2-TUNING WAVEFORMS

JAW 8-16-72

Internal Correspondence

RCA

To	Distribution	Location		Date	August 23, 1972
From	J. A. Weisbecker	Location	E-201A	Telephone	3325
Subject	Cassette "Read" Instruction Change				

FRED NOTE #5

The cassette "Read" instruction described in the FRED System Manual (July 1972) is incorrect. In Figure 9 the two "Tape IN" instructions should be:

62 & M(R(X)) = 10 for program read mode.

62 & M(R(X)) = 20 for direct read mode.

This change should also be made on page 15 of the manual .

J. A. Weisbecker

JAW:ln

Internal Correspondence

RCA

To	Distribution	Location		Date	August 30, 1972
From	J. A. Weisbecker	Location	E-201A	Telephone	3325
Subject	FRED Programs				

FRED Note #6

An initial list of 18 available FRED programs is provided here. This list will be updated as required. Individual authors should be contacted for detailed information. Programming aids (assembler, simulator, etc.) that do not run on FRED are not included. All of the following run on the basic 1024 byte memory system.

1. TIC-TAC-TOE (J. A. Weisbecker 11/22/71)

   Play TIC-TAC-TOE with the computer.

2. DISPLAY-SB (J. A. Weisbecker 12/7/71)

   Any pattern of spots can be displayed on the TV screen.

3. DEDUCE-SA (J. A. Weisbecker 12/7/71)

   The computer deduces which of 8 lights a player chooses by asking 3 questions.

4. TAPE WRITE-SD (J. A. Weisbecker 2/7/72)

   Permits data in memory to be recorded on a cassette.

5. WRITE TEST TAPE (J. A. Weisbecker 3/25/72)

   Records a test tape with alternating "Ø" and "1" blocks.

6. BINIM (J. A. Weisbecker 4/8/72)

   Plays a simplified version of NIM in binary notation.

7. PROGRAMMING APTITUDE TEST (J. A. Weisbecker 4/16/72)

   "asks" and scores eight trick questions.

8. R TEST (A. D. Robbi 5/10/72)

   Detects register faults.

9. ERASE (J. A. Weisbecker 5/19/72)

   Unique 2 player game which is a dynamic NIM variant.

10. DEMONSTRATION (J. A. Weisbecker 5/24/72)

    A 3 minute general description of FRED by FRED.

11. MM CHECKER (P. M. Russo 6/7/72)

    Tests memory with a variety of bit patterns.

12. SLIDE (J. A. Weisbecker 6/30/72)

    Sliding block puzzles on TV.

13. FLIP (J. A. Weisbecker 7/6/72)

    Novel state switching network game and puzzle.

14. CLUE-SC (J. A. Weisbecker 7/15/72)

    Introduction to intersecting subsets for young children in game form.

15. SPACE WAR (C. T. Wu 7/22/72)

    Duel with the computer controlled space ship on TV.

16. M TEST (J. A. Weisbecker 7/26/72)

    Tests memory for addressing faults.

17. "21" (A. D. Robbi 8/1/72)

    Simulates the card game of "21".

18. BOWLING (J. A. Weisbecker 8/16/72)

    Simplified TV bowling game for two players.

Joe Weisbecker

J. A. Weisbecker

/ck

Distribution:

R. O. Winder
N. L. Gordon
P. M. Russo
A. D. Robbi
A. R. Marcantonio
B. J. Call
C. T. Wu

19 - ARC DRILL (J.A. WEISBECKER 9/11/78)
20 - CALCULATOR (R.C. WINDER)
21 - MINUET (P. RUSSO)
22 - LUCKY PATH (P. RUSSO)
23 - PATTERNS (A. ROBBI)
24 - CARDTRAN (A. MARCANTONIO)
25 - MATCH (A. MARCANTONIO)
26 - BREAKPATH (P.M. RUSSO)
27 - DIRECT (")
28 - PATTERN NUMBERS (")
29 - DRAW (")
30 - ANIMATE (")
31 - 2K CALC. (R.C. WINDER)
32 - NEW 31 (A. ROBBI)
33 - NEW MATRIX (DRAW)
34 - 301-A-DOT (A.D. ROBBI)
35 - SPEEDWAY (A.D. ROBBI)
36 - MERRY XMAS/HAPPY NEW YEAR (JOHN/A.D. ROBBI)
37 - DIGIT DISPLAY (A. MARCANTONIO)
38 - BOARD TUTOR (A.D. ROBBI)

Internal Correspondence

RCA

JAW

To Distribution Location Date October 4, 1972

From J. A. Weisbecker Location E-201A Telephone 3325

Subject Volume and Display Modifications

FRED NOTE #7

The modifications described here have been added to all FRED systems (1-4).

Figure 1 illustrates the incorporation of an 8x64 display option switch on the debug panel. This was incorporated primarily for the current calculator program and will not be provided in a final product. When the switch is activated only the top 8 lines of the 16x64 display will appear on TV.

Figure 2 illustrates a speaker volume control located to the right of the power "on" switch. Speaker listening volume should be adjusted with this control rather than the cassette volume control. The cassette volume control should be set for optimum program loading and not changed.

J. A. Weisbecker

/jr

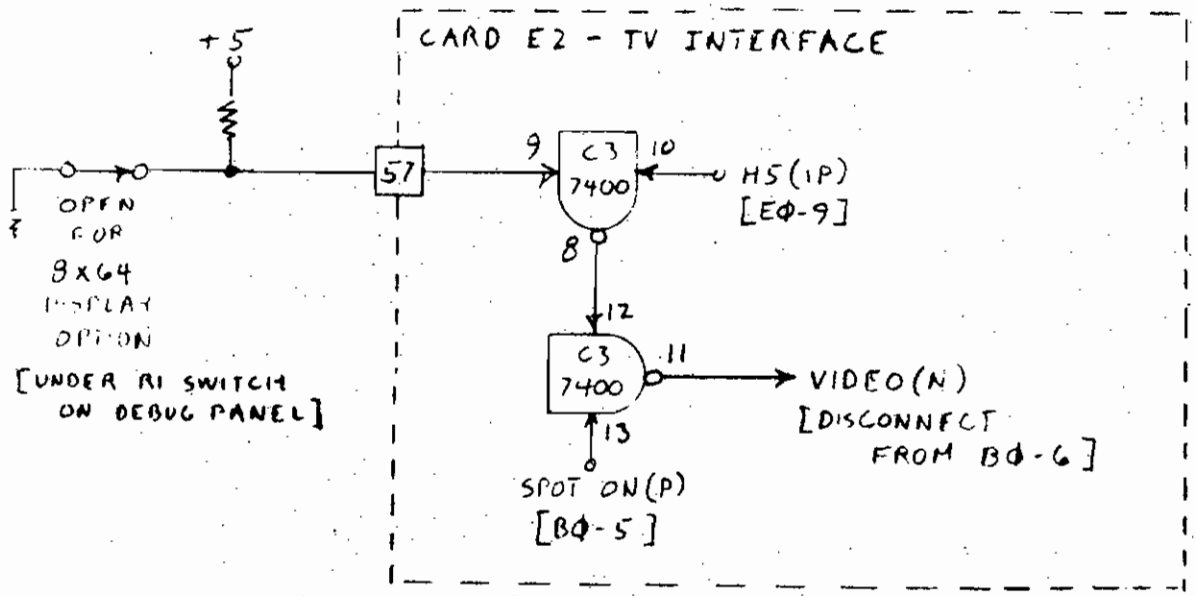


FIG. 1 - 8×64 DISPLAY OPTION

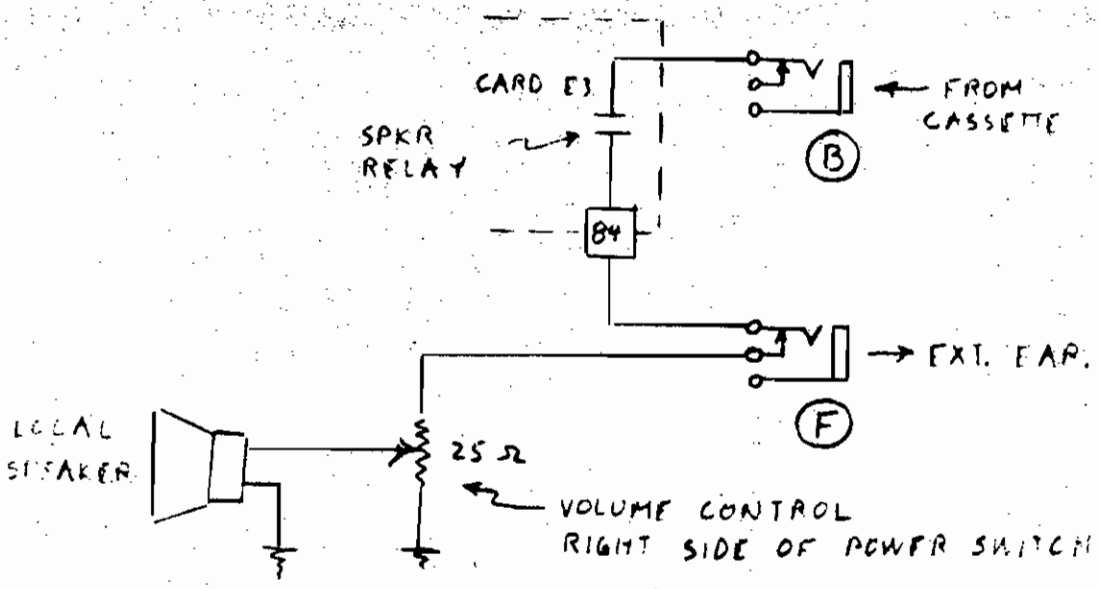


FIG. 2 - VOLUME CONTROL

JAW 10-2-72

Internal Correspondence RCA

To	Distribution	Location		Date	October 4, 1972
From	J. A. Weisbecker	Location	E-201A	Telephone	3325
Subject	Keyboard Attachment				

FRED NOTE #8

This note describes a keyboard interface which permits use of an inexpensive (bouncy) keyboard or touchpad device with FRED. The keyboard plugs into the existing card reader interface plug. (The card reader should be disabled while using the keyboard).

The keyboard layout provides for 16 input switches. Pressing a switch enters one hex digit as shown in Table I, Column A. Placing the "HEX-BYTE" mode switch in the "BYTE" position causes each key depression to provide a byte (2 sequential hex digit) output as shown in Column B. Holding "SHIFT" down during key depression produces the byte codes in Column C.

With appropriate overlays this keyboard can be used as a card reader substitute with many programs. It provides an inexpensive calculator keyboard. In the "HEX" mode it permits program and parameter loading.

The logic design incorporates sequential key scanning so that only the first key pressed is sampled. Bounce is effectively suppressed (both make and break) so any cheap switch can be used. The current logic requires only 14 TTL packages. A final design could be readily incorporated on a single 40 pin (or less) MSI chip.

October 4, 1972
J. A. Weisbecker
FRED NOTE #8

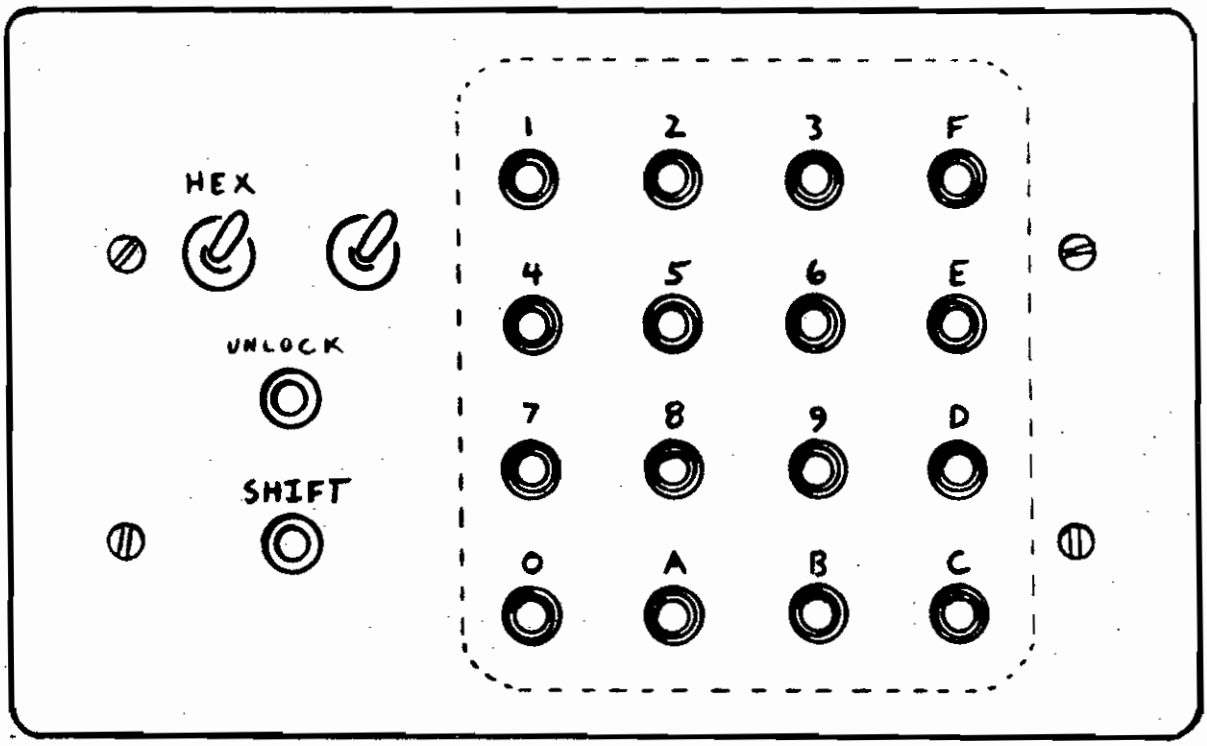
Any subsequent programs which require a keyboard should be written to utilize the codes provided by this design.

J. A. Weisbecker

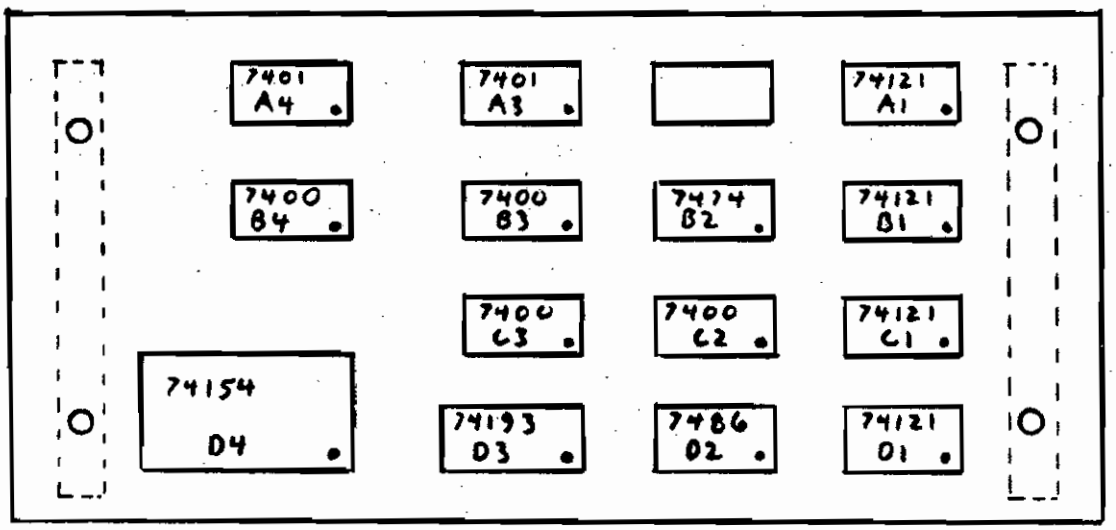
/jr

SWITCH	A	B	C
0	Ø	ØØ	1Ø
1	1	Ø1	11
2	2	Ø2	12
3	3	Ø3	13
4	4	Ø4	14
5	5	Ø5	15
6	6	Ø6	16
7	7	Ø7	17
8	8	Ø8	18
9	9	Ø9	19
A	A	ØA	1A
B	B	ØB	1B
C	C	ØC	1C
D	D	ØD	1D
E	E	ØE	1E
F	F	ØF	1F

TABLE I



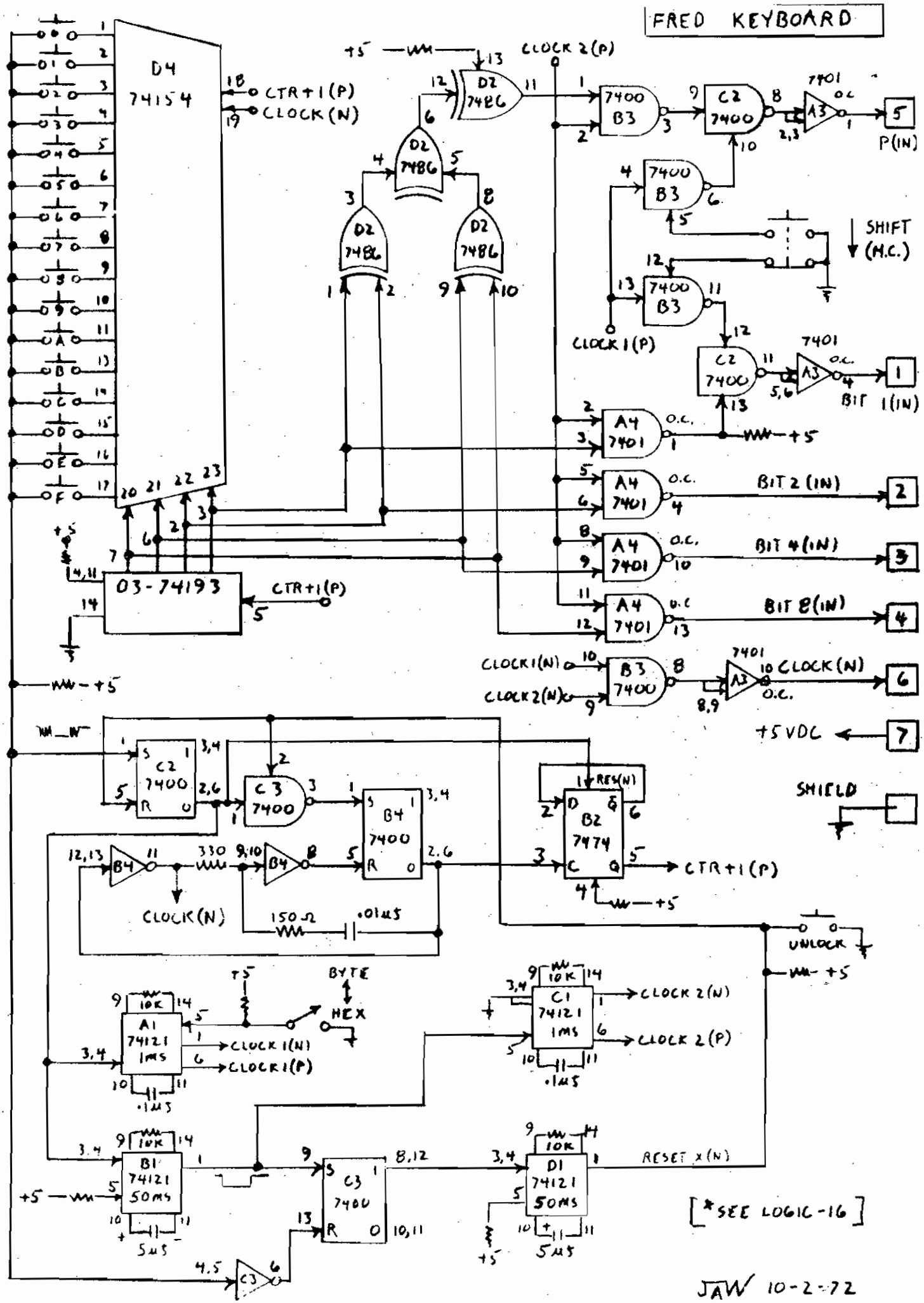
FRED KEYBOARD LAYOUT



KEYBOARD LOGIC CARD

JAW 10-2-72

FRED KEYBOARD



Internal Correspondence

RCA

To	FRED Distribution	Location		Date	February 7, 1973
From	J. A. Weisbecker	Location	Princeton, E-201A	Telephone	
Subject	FRED Note #9				

A new audio cassette tape format for the basic FRED system is described here. A subsequent FRED Note will describe the detailed logic modifications required. The current FRED tape instructions are used for the new tape format. A panel switch permits old/new format tapes to be played or recorded.

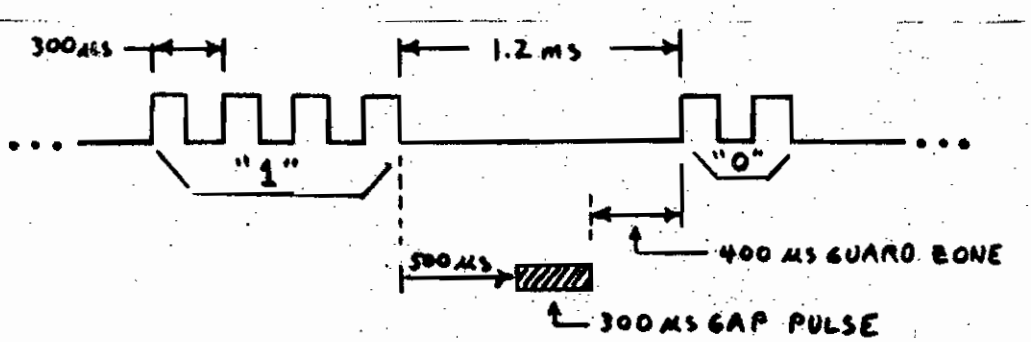
The new tape format uses a pulse counting approach to improve performance and reliability while still utilizing unmodified, inexpensive audio cassette recorders. This new format will permit program and data entry at 50 bytes/sec. A 600% improvement over the current tone detection system described in the FRED manual results.

In addition to higher transfer rates, tape speed variations of + 30% to - 50% can be tolerated. Tuning adjustments are eliminated, and head alignment is less critical.

Several different schemes might be considered when attempting to use standard audio cassette recorders for digital data. Any approach must include self clocking characteristics to permit use with mono recorders. Insensitivity to speed variations is a must. Cassette binding, variable drive speed, and out of round capstans are normal with inexpensive recorders.

The original FRED tone detection approach was found to be limited in maximum performance, sensitive to normally encountered speed and head alignment variations, and generally unsatisfactory. An FSK approach could dramatically improve performance but would still be speed sensitive so that reliability might be questionable. Certain other known digital recording techniques appear impractical with the AC coupled record-playback circuits used in audio recorders.

The proposed system eliminates the above problems by using a cycle counting system as shown below.



A binary "1" is recorded as 4 cycles of a 3.3 KC frequency followed by a 1.2 MS gap. A binary "0" is recorded as 2 cycles followed by a 1.2 MS gap, a rate of 500 bits/sec. results if "0"s and "1"s are mixed equally. Since tape packing density is 250 bits/inch, pretested tape may be required to eliminate drop-out problems.

In many audio recorders distortion and transient response make recovery of the first cycle of the 2 and 4 cycle groups marginal. For this reason during playback either 1/2 cycles is treated as a binary "0", and 3/4 cycles as a binary "1". This technique has provided reliable operation with a variety of recorders. The use of 3.3 KC is well within the frequency response of inexpensive recorders and permits reasonable margins in head alignment.

With the timing shown above, the tape could speed up by 30% or slow down by 50% without introducing error. To achieve this margin a playback gap detector is set to detect a gap of 500 μs. (The first cycle of the next bit should not occur for at least another 300 μs for error free operation).

With the new tape format, program response time to the byte ready flag must be within 200 μsec. (program mode). The increased data transfer rate should be taken into account when writing new programs. None of the existing standard FRED programs should be affected. The 4 KC stop tone currently used will not be changed. Data/program blocks should be preceeded by stop tones to eliminate noise problems in spaces between blocks or at the beginning of tape. The new program load procedure is:

1. "Reset"
2. Rewind and play cassette
3. Wait until after stop tone preceeding program
4. "Reset" immediately followed by "Read on" to load program.

Any of the existing tape writing programs can be used to record tapes. Some experimentation will be required to establish optimum recording levels, etc. for specific recorders. I have had minimum problems making good tapes with three different recorders. Copying tapes also seems to present no major problems. Several reasonably good tapes have even been made on recorders with automatic record level circuits in them. Satisfactory results have been achieved using good quality iron oxide audio tape. Chromium dioxide tape again appears to provide extra margin and should probably be used for best results.

Joe Weisbecker

J. A. Weisbecker

JAW:ln

Internal Correspondence

RCA

To	FRED Distribution	Location		Date	March 7, 1973
From	J. A. Weisbecker	Location	E-201A	Telephone	3325

Subject: Improved Cassette System - Detailed Logic

FRED Note #10

FRED Note #9 describes an improved cassette record/playback system which permits 50 byte/sec. transfer rates and improved reliability. This note describes the detailed logic, FRED modifications, and use of this new cassette system. It will be incorporated in all FRED systems.

Pages 1 and 2 show the added logic via new card E4. Page 3 describes the required modifications for cards E1, E3, and back panel. The layout of the new E4 card is shown on page 4.

Page 4 also shows the format and timing of the new recording system. This has been modified from that described in FRED note #9. 5 cycles are now recorded for binary "1". On playback, four or more cycles are interpreted as "1" while 1, 2, or 3 cycles represent "0". This permits dropping or picking up an extra cycle without error. Amplifier overshoot in recorders was found to add cycles, particularly in the case of saturation recording/playback. Saturation recording and playback, however, is desirable to reduce dropouts. The 2/5 cycle system was therefore adopted.

Page 5 illustrates typical recording and playback signals. Note that a positive clipping level is used on playback. For best results, inversion of the output signal from that shown in "C" should be corrected with the

transformer system shown at the bottom of the page. This can be built into the cable connecting the play output to the FRED tape input. It should be used with the RCA recorder and any others that invert the "C" signal from that shown. The other recorders listed on Page 5 can be directly connected to FRED for playback.

The FRED record output is shown at "A". It should be differentiated and reduced in amplitude with the RC network shown. This circuit can be built into the "record" cable and used with the recorder mike input. TDK-SD tape has been used for recording. It has high output, low drop out, and is readily available. Other tapes could be used but the TDK-SD seems optimum for iron oxide types. Computer tapes could also be used. Lower output tapes could require a modified clipping amplifier (page 2). Chromium dioxide tapes can also be used but require modified record bias for optimum results.

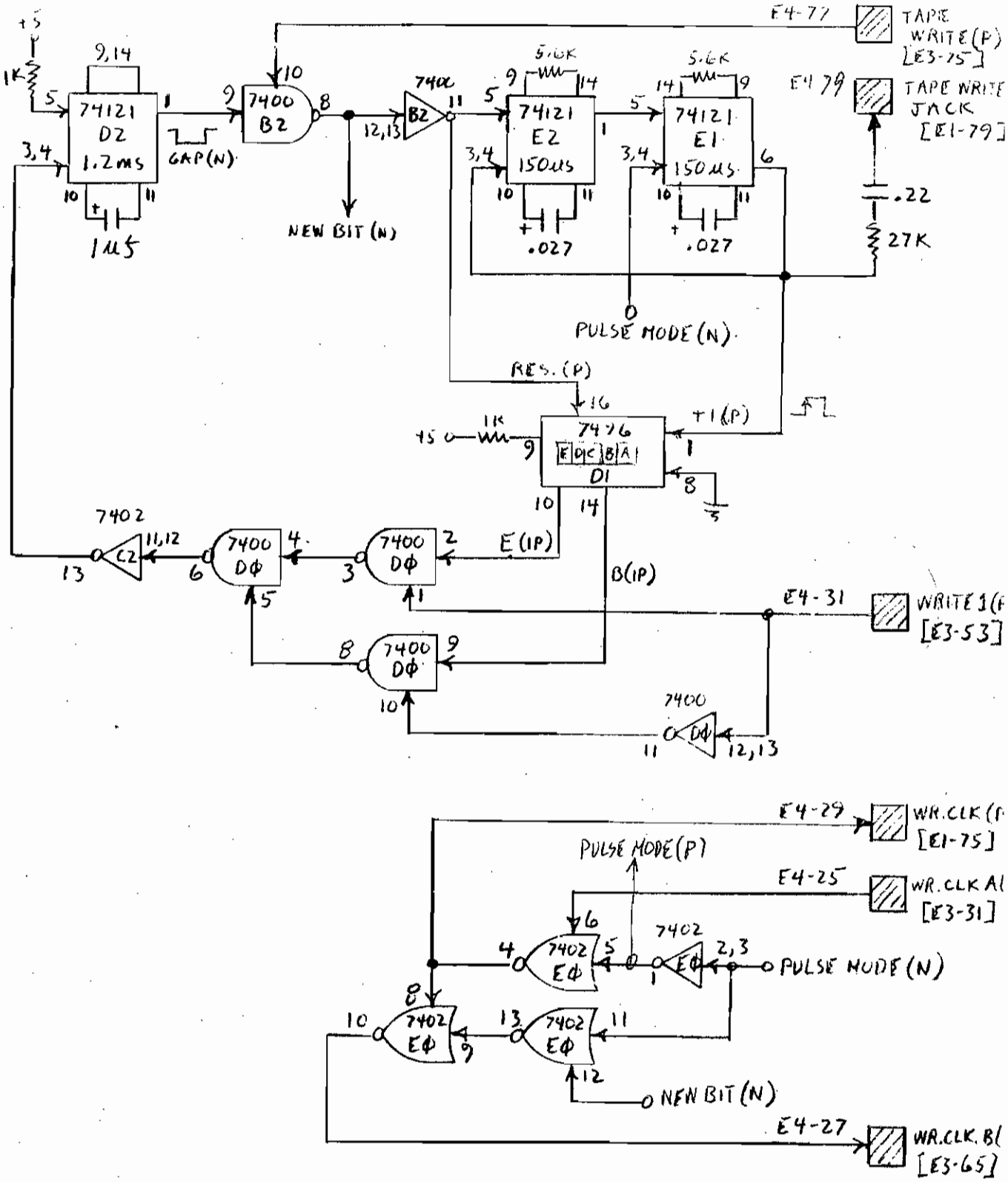
All demonstration programs have been converted to the new pulse mode (PM) system. Some FRED systems will maintain the ability to read/write both tone and pulse tapes via a debug panel switch. Pulse mode tapes have a stop tone immediately preceeding program/data blocks to eliminate spurious noise problems. After this stop tone a "reset-read" sequence is required when manually loading programs from tape.

J. A. Weisbecker

/ck

CARD E4

PULSE MODE WRITE LOGIC



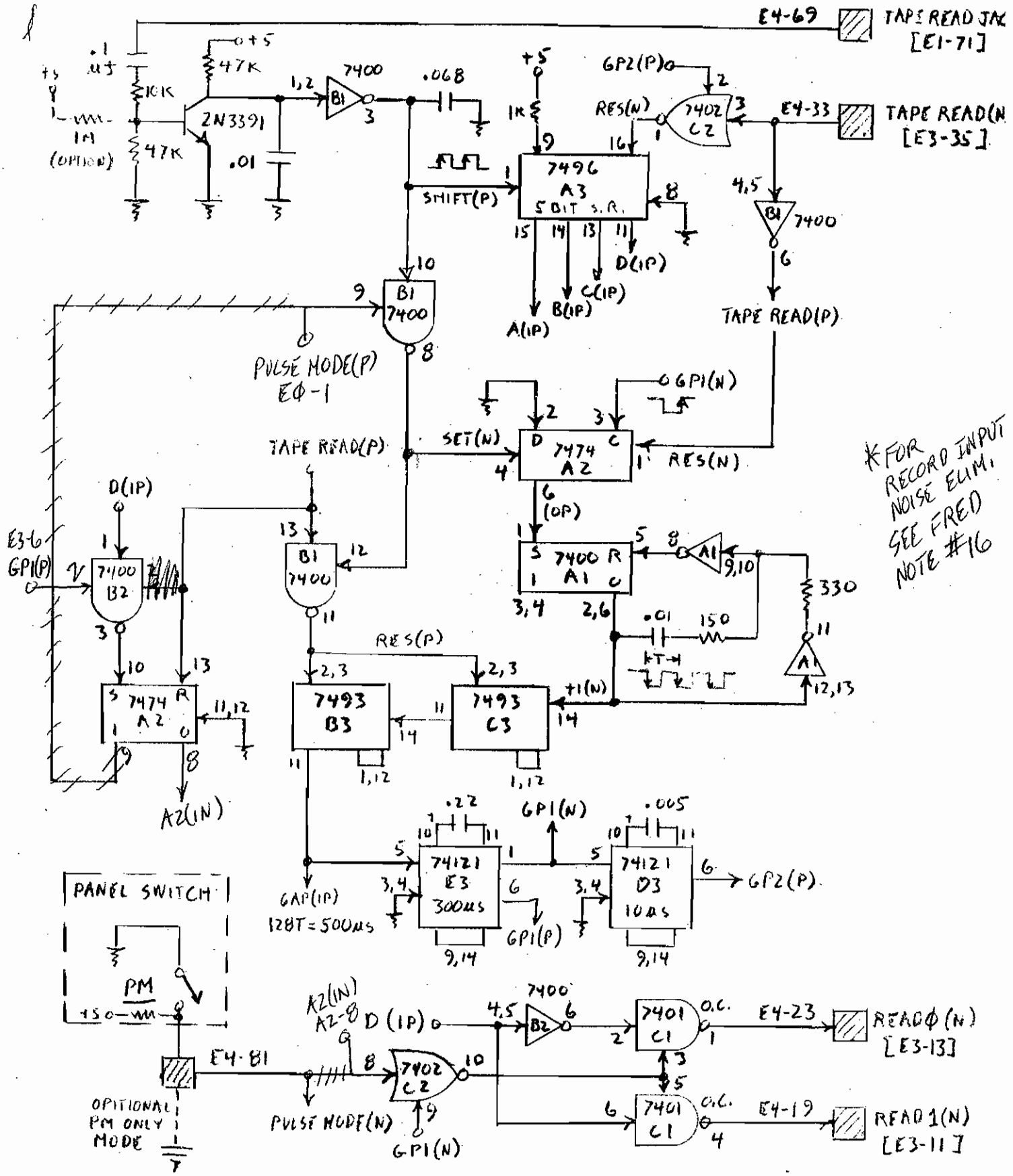
LOGIC 18-A

PAGE 1

JAW 3-3-73

CARD E4

PULSE MODE READ LOGIC



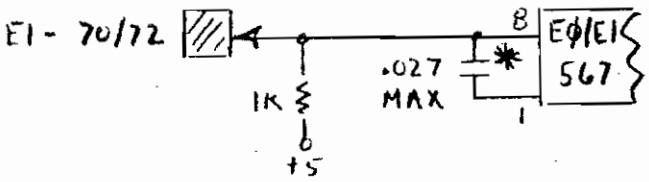
* FOR RECORD INPUT NOISE ELIM. SEE FRED NOTE #16

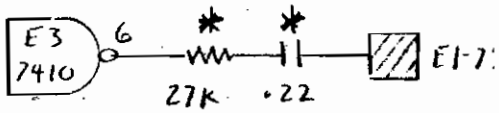
LOGIC 18-B

PAGE 2

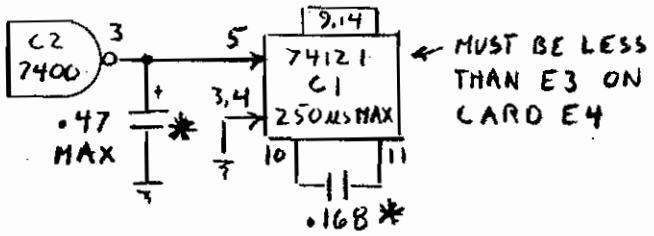
JAW 3-3-73

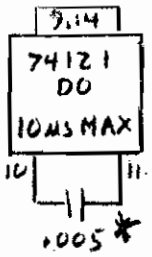
LOGIC-17 [CARD E1]



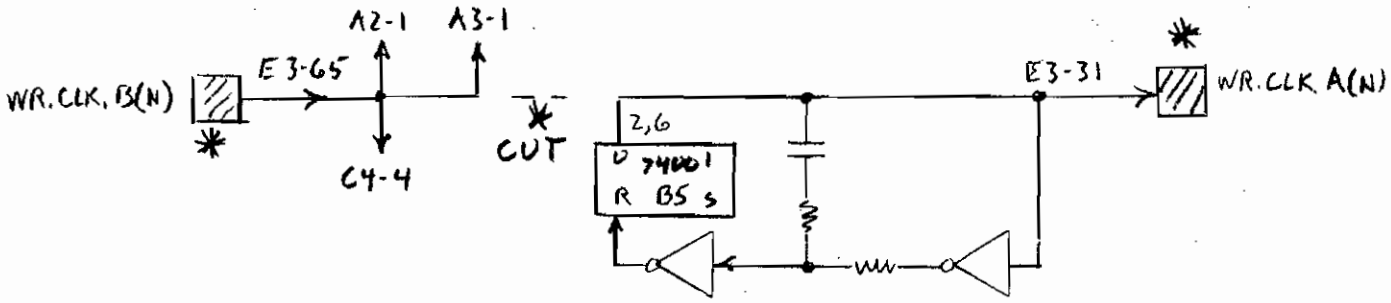


LOGIC-19 [CARD E3]





LOGIC-20 [CARD E3]



WIRING CHANGES:

BACK PANEL:

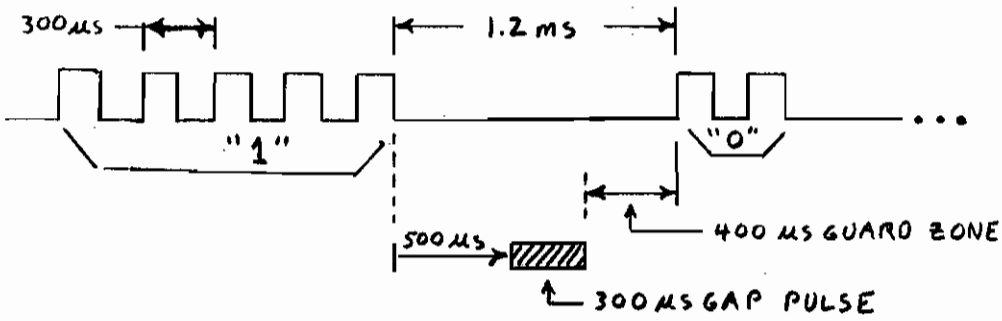
DELETE:	E1-75 to E3-65	WRITE CLOCK(P)
ADD:	E4-29 to E1-75	WRITE CLOCK(P)
	E4-69 to E1-71	TAPE OUT
	E4-33 to E3-35	TAPE READ(N)
	E4-81 to PM SWITCH	
	E4-19 to E3-11	READ 1 (N)
	E4-23 to E3-13	READ 0 (N)
	E4-77 to E3-75	TAPE WRITE(P)
	E4-79 to E1-79	WRITE OUT
	E4-31 to E3-53	WRITE 1 (P)
	E4-27 to E3-65	CLOCK B(N)
	E4-25 to E3-31	CLOCK A(N)

CARD E3 CHANGES:

DELETE:	A3-1 to B5-13 & B5-2,6 to C5-12,13
SHORT:	C5-11,12,13
ADD:	C3-2,3,4 to PIN 35
	C5-2 to PIN 75
	A3-1 to C5-11,12,13
	B5-13 to PIN 31

E1, E3 CARD MODIFICATIONS

JAW 3-3-73



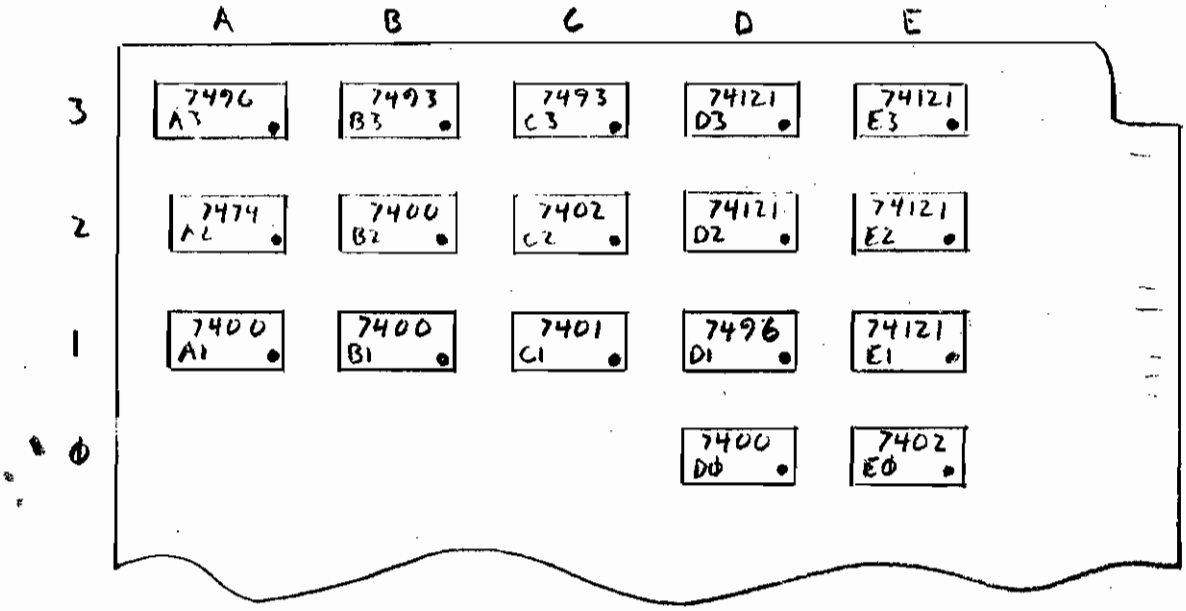
"1" = 2.55 ms ≈ 390 BITS/SEC

"Φ" = 1.65 ms ≈ 605 BITS/SEC

AVG. ≈ 480 BITS/SEC = 48 BYTES/SEC.

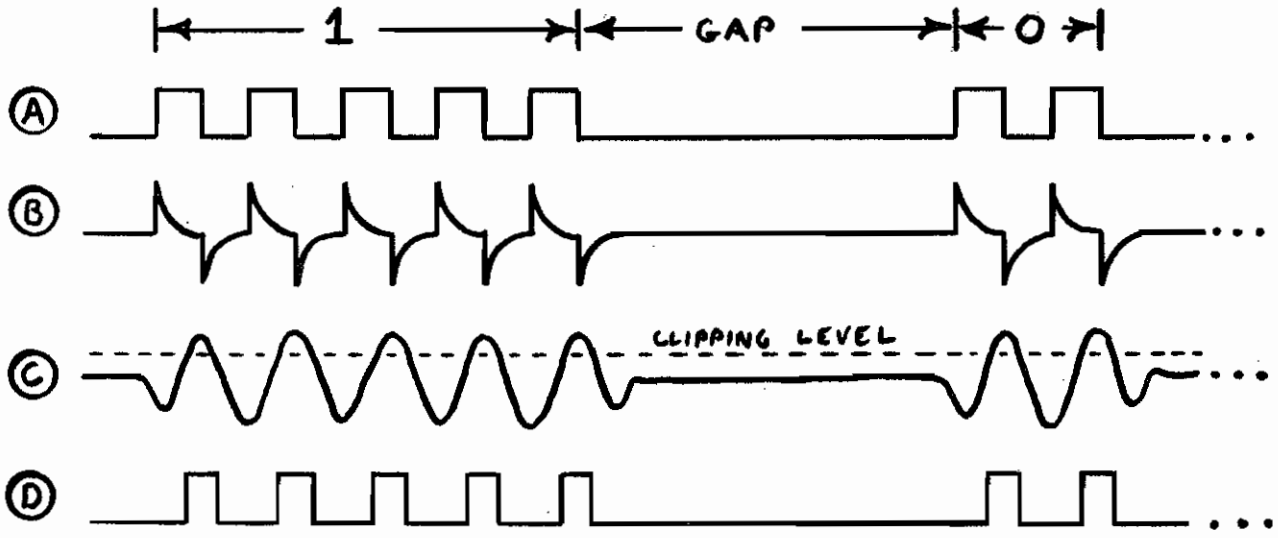
SEE FRED MANUAL FOR 10 BIT CODE USED.

TIMING

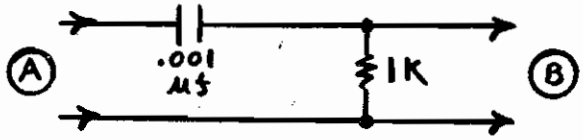


CARD E4 - PIN/WIRING SIDE

JAW 2-7-73



Ⓐ COMPUTER OUTPUT Ⓑ RECORDER MIKE INPUT

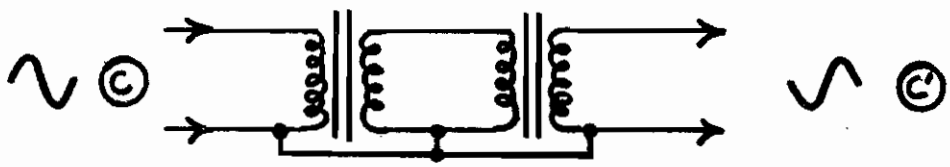


DIFFERENTIATING CKT. FOR RECORDING

Ⓒ RECORDER PLAYBACK OUTPUT FOR:

REALISTIC	CTR-9
WOLLENSAK	4050
SHARP	RD-428U
PANASONIC	RQ-209AS
DECCA	DTP-192
CRAIG	2609

STANDARD TAPES SHOULD BE RECORDED TO YIELD Ⓒ ON PLAYBACK. RECORDERS SUCH AS RCA YZB524S YIELD INVERTED OUTPUT WHEN USING STANDARD TAPES. AN 8Ω TO 8Ω INVERTER IS REQUIRED FOR USE WITH THESE RECORDERS. THE INVERTER IS ALSO REQUIRED FOR CERTAIN RECORD/COPY SETUPS.



JAW 3-3-73

Internal Correspondence

RCA

To	FRED Distribution	Location		Date	3/13/73
From	J. A. Weisbecker	Location	E-201A	Telephone	3325
Subject	Improved Card Reader Circuits				

FRED NOTE #11

Two changes have been made in the card reader circuits shown in logic -24 of the FRED system manual. A new logic-24 is attached.

First, the photodection ckt. has been modified as shown for more reliable hole sensing. The new ckt. was designed and tested by P. M. Russo.

Second, all output signals have been gated with the "card in" detection ckt. This eliminates previously encountered interaction between the card reader and keyboard.

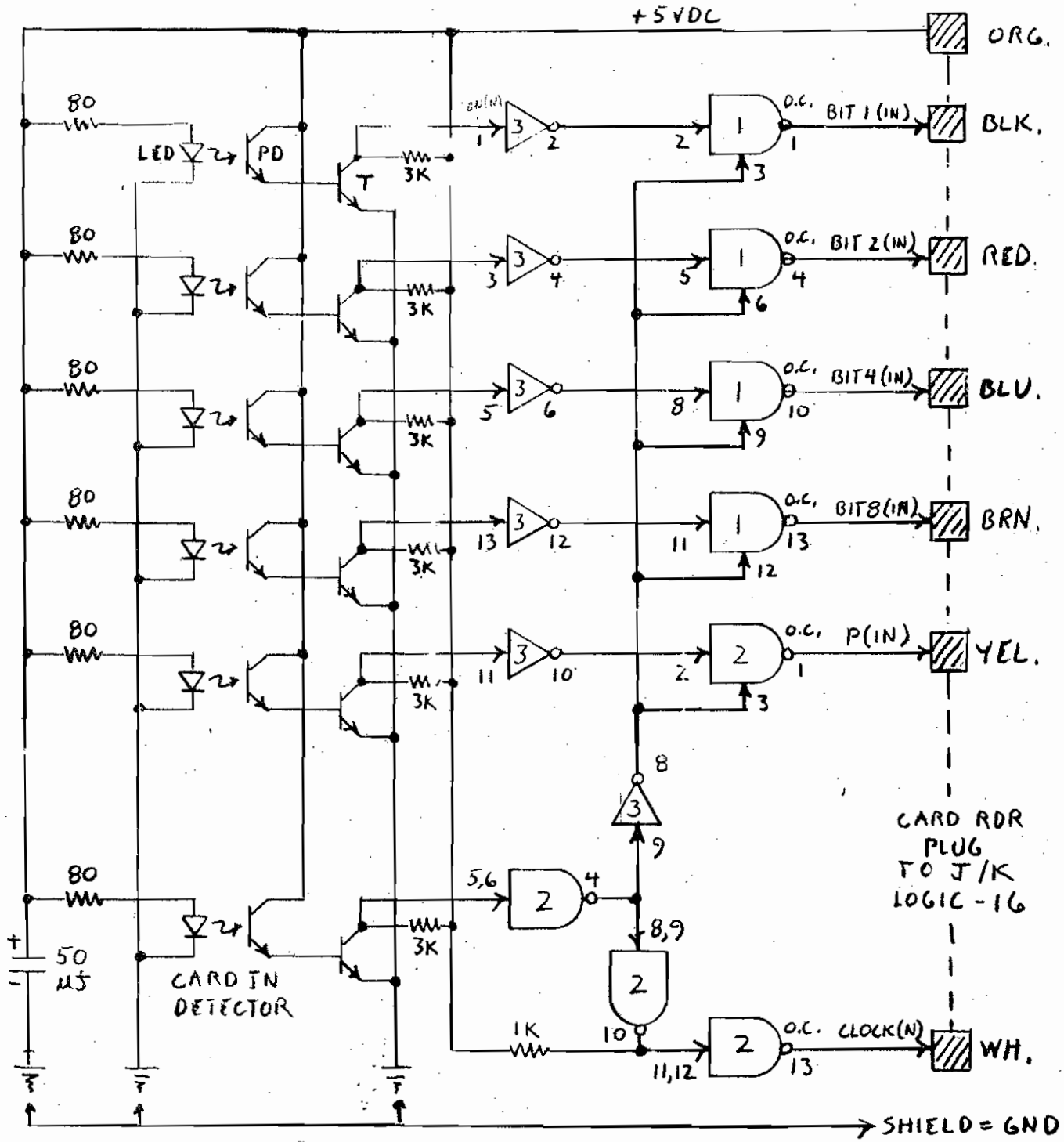
Attachment (FRED note #8). Both devices may now be connected at the same time with no interaction.

Joe Weisbecker

J. A. Weisbecker

/ck

CARD READER



LED = MOT. MLED 50
PD = MOT. MRD 450
T = 2N3391

IC-1 = 7401
IC-2 = 7401
IC-3 = 7404

LOGIC - 24

JAW 2-7-73.

Internal Correspondence

RCA

To FRED Distribution Location Date April 16, 1973

From J. A. Weisbecker Location E-201A Telephone 3325

Subject New FRED Programming Conventions

FRED NOTE #12

Next year the FRED system will be packaged using LSI chips. The debug panel will be omitted in this next design as well as in a final product design. It will however, still be possible to write and debug programs using only the new FRED systems. To achieve this, special utility routines are needed and certain conventions must be observed when writing programs.

From now on it is important to start programs at M(0088). The first three memory bytes will be "XX3088". M(0008) to M(0087) should be used as a 128 byte TV display area. Programs should not depend on the initial contents of this area. This is the only programming restriction required by the new program preparation system.

Two new utility routines are available initially. Utility #1 permits loading or displaying any memory locations on TV. The hex keyboard is used to specify memory locations to be loaded/displayed. This "Mload/display" routine is first loaded normally at M(0000) to M(0087). After "reset-run", 00/01/02/03 (hex) is entered to specify one of the following modes:

00 - Sequential byte load

First enter the M address (XXXX) at which you wish to start entering a byte sequence, then enter the desired sequence of bytes. Each byte entered and its address will be displayed (in binary) on the bottom line of the TV display. The byte is always displayed as the rightmost 8 bits and the address as the leftmost 16 bits.

01 - Single Byte load

Enter the M address (XXXX) at which you wish to write a byte, then enter the byte (YY). Repeat this (XXXXYY) for each byte you want to enter. Each byte with its address is also displayed on TV.

02 - Sequential Byte display

Enter the M address (XXXX) at which you want to start. Each time you enter any byte, the next M byte in sequence will be displayed. Bytes you enter are ignored.

03 - Single byte display

Enter an address (XXXX) followed by any byte (YY) to display the byte at that address. Entered byte (YY) is ignored.

Utility routine #1 can be used to initially load programs in memory or it can be loaded to modify/display portions of a resident program. Loading "003088" followed by "reset-run" will initiate a resident program following the use of utility routine #1.

Utility routine #2 permits writing memory to tape. It is normally loaded at M(0000) to M(0087). After "reset-run", enter an M address (XXXX) several bytes past the last byte you want dumped to tape. Entering

any byte will now cause M(0000) to M(XXXX) to be written to tape. This is repeated if another byte is entered after the tape write is concluded. The first 3 bytes written to tape are always "XX3088".

This MDUMP routine can be used to write a resident program to tape after it has been loaded/modified. Both utility routines depend on the programming conventions previously described. Other utility routines will be developed to permit examination of R registers following program execution and permit other debugging functions. With a complete set of utility routines the hardwired debug panel (and its cost) is readily eliminated from future LSI FRED systems. It is advantageous to begin bypassing the existing debug panel now in preparation for subsequent hardware designs. It is not a good idea to perpetuate the current system of imbedding a tape write routine in programs as the hardware debug panel is required for its use.

J. A. Weisbecker

/cmk

Internal Correspondence

RCA

To FRED Distribution Location Date May 1, 1973

From J. A. Weisbecker Location E-201A Telephone 3325

Subject Cassette Errors - Causes and Cures

FRED NOTE # 13

FRED users will find that the following problems account for the majority of cassette errors. Many of these problems are readily corrected.

Problem #1 - Improper Output Phase

Certain recorders provide the wrong phase output for FRED data tapes. This situation is remedied with an inverting transformer or reversing the head connections. All tapes read marginally, if at all, when this condition exists.

Problem #2 - Improper cassette seating

Always make sure that cassette is properly seated.

Problem #3 - Dirty heads

Heads should be cleaned periodically. Signal output will improve if head was dirty. Capstan and pinch roller cleaning is also recommended.

Problem #4 - Improper volume/tone Control Adjustment

Tone control (if provided) should always be set to "high". Volume control set too low will enhance dropouts. Setting volume too high on some recorders can yield excessive sensitivity to tape noise.

Problem #5 - Misaligned Head

Head alignment can be checked by scope or ear. Adjust for maximum signal output and/or peak high frequency response.

Problem #6 - Bad Tape

Stop tape at error and visually inspect for dirt, damage (crease/crinkle), etc. Electrical damage can occur if tape is subjected to magnetic field or partial erasure in recorder.

Most of the above problems can be corrected by FRED users. Damaged tapes must be replaced, of course. Careful cassette handling and proper head/roller cleaning will minimize the possibility of such damage.

J. A. Weisbecker

/cmk

Internal Correspondence

RCA

To	FRED Distribution	Location		Date	May 1, 1973
From	J. A. Weisbecker	Location	E-201A	Telephone	3325

Subject Reset Logic Change

FRED NOTE # 14

An oversight in the FRED reset switch logic should be corrected with the following change in all systems. RE: FRED system manual: logic-5.

On card P3 add C1-5 to B1-3 and B1-4 to D2-1. This will permit the reset switch to reset FF D2. Otherwise the possibility of the first load operation, after power on, being improper exists. This is a minor nuisance type problem and is readily corrected as above.

J. A. Weisbecker

/cmk

Internal Correspondence

RCA

To FRED Distribution Location Date May 24, 1973

From J. A. Weisbecker Location E-201A Telephone 3325

Subject FRED Light Gun Attachment -- FRED NOTE #15

This note describes an extremely simple and inexpensive light gun attachment for the FRED system. The gun detects light from the TV screen and simulates closure of a hex switch. It will readily detect an 8x8 dot square from a distance of up to 2 feet. In conjunction with appropriate programs a number of target shooting games are possible. The gun may also be used as a limited function light pen for parameter input. In this latter application, screen areas are alternately activated and those "hit" are defined by the time at which gun output occurs. Gun output is detected by an EF1 test instruction after card input has been selected and activated. The gun could also be connected to an individual card bit input permitting several guns to be used simultaneously.

In volume production this gun would have a manufacturing cost well under $10. It could sell for the same or less than the current "Odyssey" target shooting gun. It would provide much greater value, however, since scoring, target presentation, etc. are computer controlled.

The construction of the FRED light gun is shown in Figure 1. It's circuit is shown in Figure 2. Unlike the "Odyssey" gun, the FRED gun may be used in normally lighted rooms. Ambient room light will not cause

false outputs. This is achieved via a phase locked loop (567) adjusted to respond only to a 60 cycle light source. This is the frame refresh rate of the TV display. 120 cycle incandescent or flourescent light sources are ignored. The circuit shown generates a single pulse via an RC circuit for each trigger switch depression. A FRED input pulse is only generated when the gun is aimed at a bright portion of the TV display in coincidence with the trigger pulse. The one shot (74121) is readily eliminated in a final design, reducing cost further. An inexpensive photodiode (D) and transistor (T) yield satisfactory operation.

A prototype light gun has been constructed for use with system #1. Our target shooting program is available for demonstration. A variety of other programs using this light gun are possible and will be developed for demonstration in the future. The simple light gun attachment adds a new dimension to the FRED system and is well worth the added cost. It should be offered as an option to the basic system.

Joe Weisbecker

Joseph A. Weisbecker

/cmk

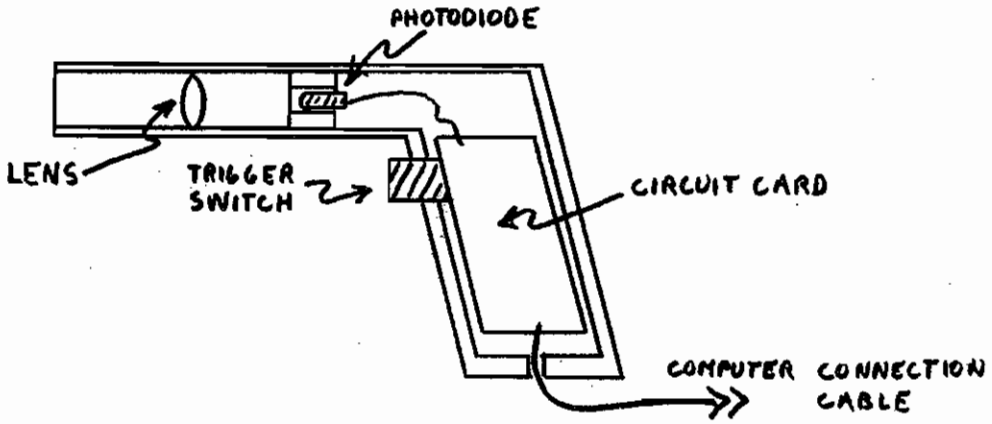


FIGURE 1 - GUN CONSTRUCTION

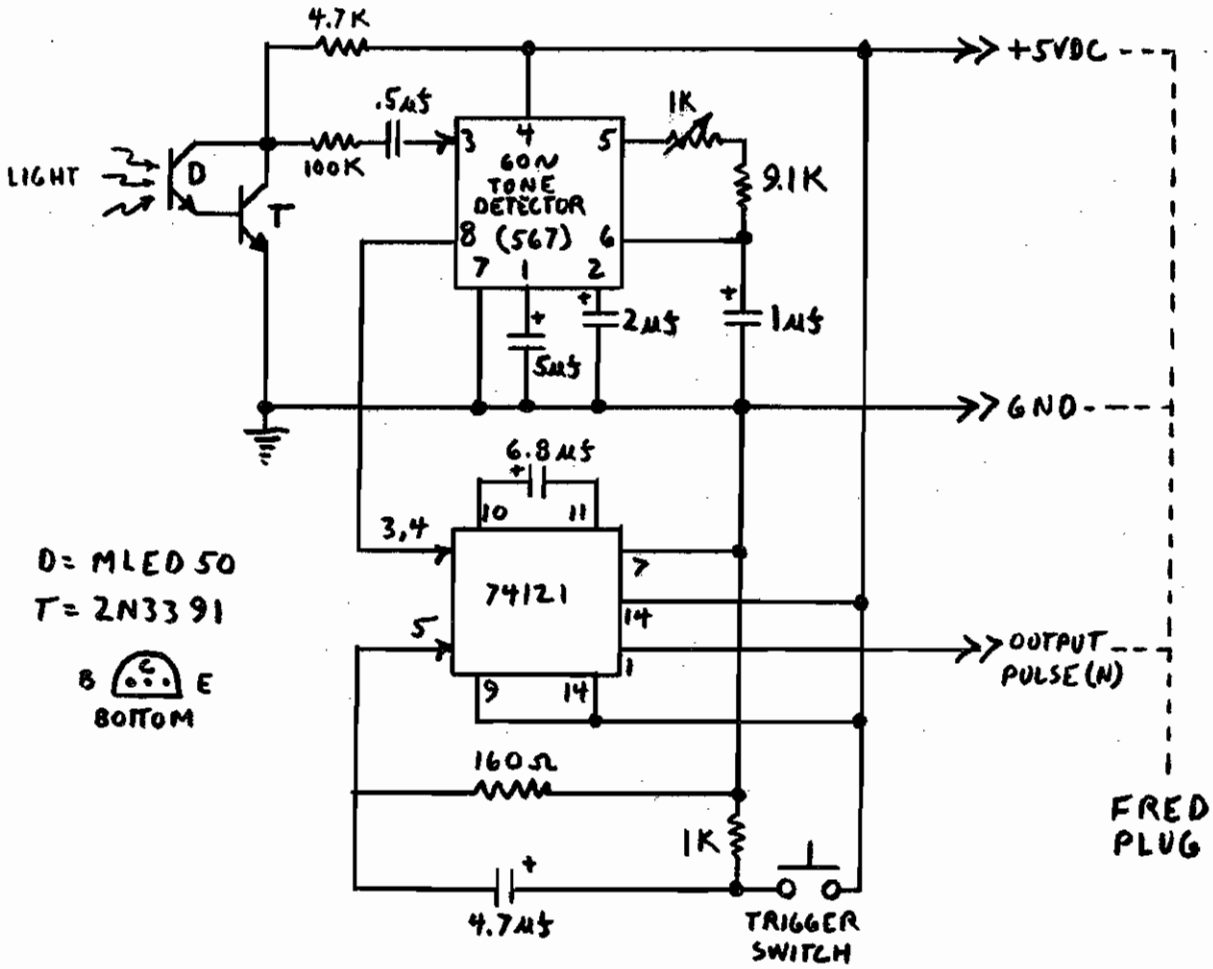


FIGURE 2 - FRED LIGHT GUN CKT.

JAW 5-26-73

Internal Correspondence

RCA

To FRED Distribution Location Date July 2, 1973

From J. A. Weisbecker Location Telephone 3325

Subject Improved Ckt. for Record Player Input

FRED NOTE #16

Bill Beyers is having 45 RPM records made for trial use as FRED program input. Programs are recorded using the same pulse code format as that used for FRED cassettes. This system was described in FRED note #10. These records can be loaded using inexpensive, unmodified, audio record players.

Unlike tapes, records seem to be prone to noise in the form of clicks preceeding the data. These are mistaken for data pulses by the current tape input circuits. Decreasing record player volume reduces this noise problem but makes the data marginal. A simple change in the tape read logic shown in FRED note #10 greatly improves its use with records. The following modifications to board E4 will incorporate this change in existing FRED systems.

At C2-8 replace "pulse mode (N)" signal with "A2(IN)" from A2-8.

At B2-2 replace "tape read (P)" signal with "GP(iP)" from E3-6.

Remove wire from A2-9 to B1-9.

Add signal "pulse mode (P)" from EØ-1 to B1-9.

These wiring changes will not affect the ability to read cassette data. They cause all input pulses to be ignored until a "1" bit code is recognized. A "1" bit requires a group of at least 4 pulses followed by

a gap. Record noise preceeding data generally takes the form of single pulses and is therefore ignored. The first bit of each byte is always "1" and will turn on the read logic when detected.

Joe Weisbecker

JAW/cmk