VISUAL PERCEPTION AND ANALYSIS OF MUSICAL SCORES

by

Jeffrey Lynn Entwisle

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January, 1973

Signature of Author.
Department of Humanities, January 24,1973

Certified by. / / Thesis Supervisor

Accepted by....
Chairman, Departmental Committee on Theses

VISUAL PERCEPTION AND ANALYSIS OF MUSICAL SCORES

by

Jeffrey Lynn Entwisle

Sumitted to the Department of Humanities on January 24, 1973 in partial fulfillment of the requirements for the Degree of Bachelor of Science.

ABSTRACT

This thesis documents research on the recognition of notes by computer and the devices and programs necessary to enable a mini-computer to see and analyze standard musical texts. This system requires in part the use of a television camera and a home brew interface device to transfer a representation of the camera image to the computer and eventually to a large disk storage medium. A large portion of the thesis is devoted to the developement of programs that format the camera image to conform to a particular set of hardware and software configurations. The rest of the thesis describes methods that were effective in determining features of the musical score seen by the computer (such as pitch and time) that would be essential in any musical analysis. In addition, methods for the extraction of additional features about the musical score are presented.

Thesis Supervisors: Nicholas Negroponte-John Buttrick
Titles: Associate Professor of Architecture
Professor of Humanities

ACKNOWLEDGMENTS

I wish to give special thanks to Professor Nicholas Negroponte for his assistance and guidance in the course of my research and for his patience in the preparation of this thesis.

I am indebted to Andy Lippman for his insight into the problem and to my co-workers in the Architecture Machine Group for their undying counsel and encouragement.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

I. INTRODUCTION

This thesis documents the results of research carried out by the author at the Architecture Machine Group, Massachusetts Institue of Technology. This work was sponsored mainly by Professor Negroponte, and in part by Proffessor John Buttrick and was carried out from June, 1972 to the present.

The idea for my thesis gre out of my involvement with the Undergraduate Research Program at M.I.T., particularly at the Architecture Machine. Attention there focused on experimental systems that could eventually lead to the construction of a Responsive Environment, that is, an architectural man machine system capable of reacting "intelligently" to subtle changes in the movement and intent of its inhabitants. Interesting results, I concluded, could be realized if the results of these experiments were coupled with the fairly well defined reaction expected of a competent, shall we say pianist, or other virtuoso, and required to react responsively to a composer by synthesising his musical intent, precisely according to time honored standards, and subtly as well, through the musical score environment.

Design and implementation of a television interface for an Interdata mini-computer served a two fold purpose. As an extention to a Responsive Environment, the television provided yet another feature about the behavior of the inhabitants thereby making the environment more responsive. Secondly, it provided a universal graphics terminal that closely models the human eye and makes possible research into the kinds of processes used by man in perceiving architectural designs. A small extention to this system, now with an "eye", could be imagined that would be smart enough to be able to read and play a sheet of music.

CHAPTER II BACKGROUND

2.I Overall Machine Structure

The Architecture Machine is a multi-processor environment that allows the user to edit, assemble, and compile FORTRANor ASSEMBLY language programs. Peripherals such as printer, reader, punch, etc., are shared among the processors with each switchable under manual or program control to any processor.

A fixed head disk is used to contain the MAGIC operating system and user related programs. Space is allocated on the disk for large graphics data bases and programs available to easily aquire sections of time dependent data at a time independent rate. These programs convert input from various terminals (Sylvania Tablet, Magnetic Tape, television camera) to x-y addressable disk grids in addition to routines to access the grids by transfer to and from FORTRAN arrays.

2.2 SHARED BUS

The shared bus is an electronic switching system designed to allow four Interdata processors to share a single set of peripheral devices. The idea for the bus grew out of two observations. One, that most of the processors only needed some of the peripherals some of the time, and secondly, that back-up peripherals needed immediate switchability to enable the Architecture Machine to function efficiently.

The first set, consisting of devices like the line printer, tape-drives, paper tape reader and punch, are primarily used in burst mode, that is, a single user will use these devices continuously for short periods of time. A single user will list all of his files at once, or punch a backup when he leaves. The second set consists of Sylvania Tablets, the ARDS', and several home-brew real time peripherals including a television camera, and pressure sensitive pen. These devices are primarily used continuously by any single user and must be available for dedication to any processor. Some devices, like the disk, require both kinds of switchability.

In configuration, the shared bus consists of three cages; two of which contain the peripherals controllers themselves, and the third the switching electronics to attach these peripherals to any processor. Above these cages is an indicator/control panel, to control all devices separately, and to display the current status of each device.

In operation, the twenty seven lines of the multi-plexor bus of each of the Interdatas, is buffered and sent into a switch card, one for each peripheral controller. This card

contains all of the gating electronics required to attach that controller to any processor. When a switch gates a controller to a specific processor, that peripheral is locked onto the bus of that processor, and appears unavailable to other processors. Therefore, the ahred bus is invisible to the processor. When a data transfer is attempted, either it will pass directly to the device required, or the processor will sense that the device is not available and will attempt the transfer at a later time.

2.3 MAGIC

MAGIC is an interactive system that provides the user with editing, assembling, and compiling programs that can be operated on files or directories containg source programs or object texts. Files are stored in disjoint collections called directories. The data base is organized into directories and each directory contains an arbitrary number of files.

This system is meant to allow several different users to store files on the disk without regard to naming conflicts. By only having a subset of directories available at a given time, a disk being used by two persons could have a separate directory for each (e.g., directories 'TOM' and 'HARRY'). Thus when TOM uses the MAGIC system he might have only one directory active at a time. In this case, a file reference is resolved by searching all the currently open directories. Open directories are ordered, that is, if two open directories each have the file 'A.FORTRAN' then the file which belongs to the directory closer to the start of the active directory chain will be found (somewhat of a simplification).

MAGIC allows the user to form his own repertoire of commands out of his applications programs. Programs may be invoked by MAGIC commands rather than at the program level. In addition, specific input-output configurations can be assigned to allow the user to access any of a number of shared peripherals.

3.1 SENARIO

Picture, if you will, a student of music theory entering the machine room and sitting down at a console. Behind him lurks an automated television camera. To his left a Sylvania Tablet and pressure pen. In front, an ARDS with a keyboard below. The student types commands and begins to draw on the sylvania tablet. On the screen appears his picture and seems to follow his movement. When he is done, another typed command is issued and minutes later his synthesised score is heard from within the bowels of the Architecture Machine. Not satisfies with his own composition, the student places a Bach score borrowed from the library in view of the television camera. More commands are issued and again, minutes later, the synthesised score is heard. In addition, the notes are redrawn nicely on the ARDS screen and sections may be picked to be played over.

Such are the possibilities. In reality, this system has only developed to a stage of limited score complexity recognition and does not play the score. It does, however, normalize and identify notes.

3.2 VIDICON INTERFACE

The task of designing and building a device to make a visual image available to a mini-computer processor in digital form, proved to be a complex engineering problem with a state-of-the-art solution. Numerous attempts had been instigated with little success. Before remarking upon the results of my investigation and solution to the problem, I would like to present some alternative systems for the digitization ofimages from "visual" devices.

3.3.1 Vidicon Vs. Vidisecter

Two type of visible light scanning devices or optical scanning devices (O.S.D.) are most readily available today.

The first O.S.D. is the Vidisecter. It employs a random access photomultiplier or light sensitive unit with appropriate lenses and electronics, to interrogate any point in its field of vision for the level of light present. The interface or device that converts and transfers data to the processor, in this case, is fairly uncomplicated. As long as the image is stationary, the processor may take any amount of time to completely analyze the image as any point is immediately accessible from the Vidisecter. No data structure need be provided on-line to preserve previous data transfers. The Vidisecter is completely time independent thereby freeing the processor from complex data transfers. However, the unit has several limitations that hinder its wide acceptance. First of all, it is very insensitive to light. Bright sunlight is the bare minimum level required to produce an adequate image. Even though it is sensitive to a range of

light magnitudes, the difference between the lightest and darkest image is small compared to the overall magnitude of light required to produce any image, therby resulting in poor contrast. Secondly, it is a good deal more expensive than the vidicon (90,000 from Texas Instruments).

The second O.S.D. is the Vidicon. It is perhaps the most readily available of the O.S.D's due to its wide use in the broadcast industry. It is noted for it's sensitivity to light and remarkable resolution. It provides a wide range of grey tones (similar to human vision range) and some units are sensitive to color as well. It employs a sequential scanning system that produces a standardized composite video signal. This signal may be easily monitored by a television set or transfered over long distances by a single cable or by radio emmision. Most people who have used Vidicons (i.e Stanford and Edinborough) have attempted to model an interface after a Vidisecter interface, that is, make a Vidicon look like a Vidisecter. A great deal of difficulty was encountered by these people because of the complex timing and critical conversion circuits needed to make the Vidicon seem randomly accesible. Somewhat like shooting at a moving target with a target so fast one can't see it.

The use of the Vidicon and a different interface was determined to be the most appropriate solution for several reasons. By making the Vidicon line by line accesible instead of point by point accesible, it would be possible to greatly reduce the complexity of the interface without sacrificing resolution of the image. This system, however, would require

data storage mediums within the interface to save video data for later and slower transferal to the processor. In addition, large, on line, storage devices would be required to facilitate random access at a time independent rate, since the Vidicon would not be directly point by point accessible. The interface storage medium would have to be fast and easily interfaced to standard ttl logic. A read/write time of less than 40 nsec would be required to save video lines 256 points wide. The analog to digital conversion circuits must likewise be fast to keep up with the rapid variations in light level available from the Vidicon (on the order of 10 MHZ.). The recent introduction of fast integrated circuits to perform functions of analog to digital convertion(signetics NE-529) and bulk storage (Fairchile #93410) greatly alleviated these problems and helped reduce overall expense and complexity. A large on line storage medium was also available (disk) making random access of the Vidicon image quite feasible. In fact, disk GRID programs had already been written to access stored disk data in this manner.

An important consideration in the choice of O.S.D's was the Vidicon's wide grey scale.Salient features of the image could be detected through tonal analysis. Shadows and subtle intent is conveyed through tone and is important to a human observer. This makes grey tones important in a Responsive Environment.

MOST IMPORTANTLY, the Vidicon could be used under normal lighting conditions, thus making the camera unknown to the

inhabitants of the system. The addition of color would extend this feature even further.

3.2.2 CONVENTIONS

Lines in the interface are labeled and then suffixed by a number indicating their active state. Suffix "0" refers to a voltage of between 0.0 and 0.9 volts. Suffix "1" refers to a voltage of between 0.9 and 5.0 volts. Gates are numbered with their integrated circuit location and signals emanating from the gate are labeled with the i.c. pin number. Signals that go to places not on the page are labeled with 1) page number where the signal can be found, 2) the location on that page (e.g., 6b2 indicated page 6 location b2).

The Interdata is an eight bit, variable word length processor and bit representaion is in standard HEX notation. The left most bit of the left most byte is the most significant. Signals to and from the Interdata are labeled with a number of the form xxx-z where z is "1" or "0".

3.2.3 Interdata I/O

The Interdata Model 5 computer employs two eight bit data lines and various control and command lines to facilitate transfer of information bewteen the processor and its up to 256 peripherals. Eight data available lines (DAL) provide information from the processor to the interface and eight data request lines (DRL) provide information to the processor from the interface. Five control lines (SRL,CMDL, DAL,DRL,ADRS) direct the function of the DAL'S and the DRL'S.

3.2.4 I/O Programming

In general, there are three steps necessary to control the peripherals under processor command. The first step employs output commands to communicate states used by the interface. These commands may direct the specific operation of the interface. Status commands are used whenever the exact state of hte peripheral is in question. Finally, read and write data instructions are issued that transfer "live" data to and from the peripheral. Read and write data instructions typically transfer data a byte at a time. An extended I/O feature provides for read and write block instructions that automatically simulate multiple read and write data instructions, plus status checks.

3.2.5 Capabilities

The interface is capable of converting any horizontal line of video data into 256 three bit "points" of binary data and storing this line in an internal 256 by 3 isopolar memory. Seven levels of grey are thus produced for the entire Vidicon scan. It is possible to convert, sequentially, lines that are from one to eight lines apart, depending on the speed of the processor and the sycronization with associated peripherals. Conversion may be halted at the end of any line and restarted from the end of that line. Vertical retrace time as well as conversion of the last line and certain data errors may be sensed.

3.2.6 Limitations

The video data is converted and stored as fast as the circuitry will run. No sample and hold circuits are provided. Therefore, the binary data may be off as much as one level. Poor adjustment of the interface or camera may result in malfunction of the Interface as well as poor level separation. Only one interlace of the Vidicon scan is used in conversion. Half the full resolution in the y direction is thus not realized.

3.2.7 Operation

Typically, input from the Vidicon is accomplished in three phases; setup, initialization, and data transfer.

3.2.7.1 Setup

This phase presently requires the use of an oscilloscope and adjustments of at least two potentiometers. Furture developement will include automatic adjustment but a manual adjustment procedure is presented here as a guide to immediate users. The Vidicon camera must be used with a monitor. The subject drawing is placed before the camera lens and the lighting and contrast of the Vidicon adjusted to produce a sharp, clear, contrasty image on the monitor. The edges of the viewing feild is to be avoided as these areas are ignored by conversion routines. The oscilloscope should be triggered off the line and the output of the second vieo amplifier monitored(TP-1) . VR1 is then adjusted untill the pedastal is cutoff (at about 1 volt). This will result in pure video signal for the a/d converters. The sync separator should now be monitored (TTP-3) and VR3 adjusted untill a "0" level is seen about every thirtieth of a second and of approxiately 10 micro second duration. This down time corresponds to the vertical retrace time and is critical for corect conversion of video data.

3.2.7.2 Initialization

This phase is accomplished in software through the use of available output commands. Typically, System Clear is issued twice. Commands that direct the function of the

interface are now executed. It is necessary to establish the line increment (output command SET LINE INCREMENT) and the command to halt upon data error (DATA ERROR STOP) at this time. System Clear is then reissued and data transfer begun immediately.

3.2.7.3 Input

The interface is now prepared to track horizontal lines and convert at the appropriate line. There are two commands that will start the interface tracking electronics. Output commands START begins conversion at line one and continues conversion at the last line converted plus the line increment. This operation will stop only if output command HALT is issued or DATA ERROR STOP has been issued previously and a data error has been detected. Reissuance of the START commands will begin at the end of the last line converted, i.e., the next line that was available becomes the first line available after a HALT sequence.

The 256 points of three bit video data are read by 128 consecutive read data's (or the appropriate read block instruction). The data is packed two points per byte, right justified on 4 bit boundaries. Reading may only begin(BUSY status drops) after the next vertical retrace time and at the correct line location. Conversion of the last line is detected by EOM status (x'02'). If a read block instruction is employed, only EOM is a necessary indicator. Sense status operations are automatically provided by this instruction.

If input of a single line is desired, output command SINGLE LINE CONVERT will perform a START and halt at the end of the converted line automatically. In any case, START or SINGLE LINE CONVERT must be issued immediately before any read instructions or errors may result.

3.2.7.4 COMMANDS

Issuing one of the following output commands causes the interface to perform the specified function.

x'80' START

Resumes tracking horizontal lines from the line that was last converted or initiates tracking from line one if SYSTEM CLEAR has been issued immediately beforehand.

x'40' HALT

STops tracking electronics at the end of the present line.

x'20' SYSTEM CLEAR

Resets tracking electronics and initiates all functions to the top (first) line

x'10 SINGLE LINE CONVERT

Like START but performs a HALT after conversion of the next available line.

x'02' DATA ERROR STOP

Will enable EXAMINE status and perform a HALT if any of the following conditions hold true;

1) Input of digital data to the processor is not completed by the time the next line is ready for conversion.

2) Conversion of a line is occuring and the processor attempts to read the interface buffer.

x'01' SET LINE INCREMENT

Indicates that the next write data instruction contains the increment between lines that are to be converted. The increment may be greater than zero but less than x'f'.

3.2.7.5 STATUS

The status indications from the Vidicon are described below. Each indicator is gated onto the appropriate data line during the time of a sense status intruction.

x'80' VERTICAL RETRACE

DRL 0 active during vertical retrace time.

x'20' DATA ERROR

DRL 2 active if a data error was detected since the last sense status instruction.

x'04' EXAMINE

DRL 5 active is data error stop has been issued and a data error has been detected since the last sense status intruction.

x'02' EOM

DRL 6 active when the bottom horizontal line is encountered.

3.3 HUNCH ENVIRONMENT

HUNCH is a set of programs which enables the user to communicate with the computer in a natural way through the medium of sketching. The sketcher draws on a Sylvania Tablet with a pen in an unrestricted manner, while the computer attempts to comprhend the user's intent by analysing the data, dynamics, and sequence of the sketch. There have been other computer systems developed which purport to allow a person to sketch using a computer, but all require that the user learn some specialized skill, language, or style. HUNCH tries to keep the communication as natural as possible, on the part of the user, relying on additional information unavailable to a person looking at the sketch post facto to simplify the problems of understanding intent.

The programs are invoked through commands to the operating system. At this time, the commands must be explicitly requested by the user from the user's console, but modifications to the compiler on the system will shortly enable them to be invoked implicitly under program control. Initialization of the HUNCH system is accomplished by the MAGIC command HUNCH16 (or HUNCH17). This program loads a set of constants into core which established the environment for subsequent input data and programs. These constans determine such items as where input data are to be deposited, the initial limits of the GRID map (used by all GRID programs) and the initial configuration of input and structure buffers. There are numerous commands available to the user to draw and anal-

yze sketched data. The program HUNCH16 (or HUNCH17) is the only HUNCH routine required in the transfer of video data to the disk.

3.4 GRID Package

The GRID package provides the FORTRAN user with a set of x-y addressable grid (up to 1024 by 1024) which are used mainly to store sketches and more generally, data from a number of graphics terminals including a 200 point per second Sylvania Tablet-graphics pen for drawing made by hand, and a television camera for input of predrawn sketches. This type of storage medium allows a sketch to become completely independent of time and provides a computer sratch-pad for analysis of comples architectural configurations.

Each grid is stored as a bit map on a fixed head disk storage device, with each 1024 point line represented by a 128 byte record. A grid can consist of between 1 and 1024 such lines and any number of separate grids may be used limited only by the size of the disk. When a 1024 by 1024 grid is used to represent a sketch drawn with the Sylvania Tablet (4096 by 4096), a bit is set on the grid if a line from the Tablet passed through a square 4 by 4 tablet coordinates in size. When used with the television camera, each line of video data is stored one after another until the entire sketch is scanned.

The data on the grid is accessed by means of an assembly language program. A "window" of arbitrary width and heigth can be transferred to a FORTRAN user's array and likewise, data may be transferred from a FORTRAN array to the grid. A scale may be specified in addition to the size and

location of the window. Imagine, if you will, a slide projector with various and moveable size screens and a large number of lenses with different focal lengths. The size and position of the slide screen in many respects is similar to specifying the size and location of the window on the grid. Changing the lens of the projector has much the same effect as changing the "scale" of the window. The scale may be so varied, that one one element of the array represents any number of bits set within the window (from 1 to 1024) and may, indeed, represent the entire grid.

It is possible, therfore, to perform many operations on the grid and easily aquire large data bases from which conclusions about the form and structure of the data may be drawn regardless of the real time nature of the image. This is particularly handy in the analysis of "visual" images as the video data rate is magnitudes faster than the maximal data transfer rate over the I/O bus channel. Data that would prove particularly difficult to analyze in real time now becomes readily accessible to the FORTRAN user.

3.5 CAMERA

3.5.1 Overall Structure

It was my desire to format the visual data and store it on disk in a manner that would be compatible with other input devices. Thus, existing programs could be used to manipulate the data. Initially, I concluded, a stand-alone viewing program would be intructive in determining the manner of data formation best suited to producing the sharpest and most noise free image on disk. This was accomplished by inputting a line at a time fmm the Vidicon and redrawing the line on the ARDS between video points that conformed to a particular set of restrictions. (see description of CAMERA for these restrictions). This bit representaion was searched for contiguous patterns and these patterns plotted in the form of lines and spaces on the ARDS screen. (In this type of display unit, the image reamains after it is plotted for an indefinate period of time).

Then disk routines were employed upon the bit representation to save the formatted "visual" image for later retreival.

3.5.2 Disk Routines

Depending on the routine used, data may be placed in two locations on the GRID, the center and lower left hand corner. Starting from the top and working down, each of the 256 lines of video data are formatted and then written to the disk.

3.5.3 CAMERA Program

Purpose

This program is used mainly to display video data on the ARDS as quickly as possible. Tape routines are included to save formatted data representation on magnetic tape.

General

CAMERA will operate in any MAGIC environment. It accepts commands via logical unit 5 (lu5) and ouputs messages via lu 5. All other I/O is performed with internal subroutines.

Command Level

Command level is indicated by the message READY. Input to the command interpreter is allowed only after the message V: is printed. Illegal commands are ignored and a ? printed.

Execution Level

Commands are executed according to type (see below). Depressing break on lu5 causes termination of execution level and return to the command interpreter.

Commands

EOF, SKIP, and REW are like standard MAGIC tape commands and are executed immediately.

INPUT, and OUT commands are used to specify the particular input and output devices required.

DRAW, THRESH, and MASK commands invoke data conversion

routines on raw video data after a GO command is encountered.

QUIT returns control to MAGIC

GO causes execution of data rouitnes

CLEAR restarts the CAMERA program.

Data Routines

Three bit per point video is converted into a bit per point representation as follows.

DRAW WHITE

Bit is set only if the magnitude of the corr-esponding video point((video)) is greater than the magnitude of THRESH ((THRESH)).

DRAW BLACK

Bit is set only if (video) is less than (THRESH)

DRAW LEVEL

Bit is set only if (video) equals (THRESH)

MASK xxxx

Bit is set only if (video) is not equal to (xxxx). Any combination of video levels may be masked.

COMMAND	ARG1	ARG2
INPUT	TAPE	16
	----	17
	CAMERA	--
OUT	TAPE	16
	----	17
EOF	16	--
	17	--
SKIP	16	--
	17	--
REWIND	16	--
	17	--
MASK	(0-7)	--
THRESH		(0-7)
DRAW	BLACK	--
	WHITE	--
	LEVEL	--
GO	----	--
CLEAR	----	--
QUIT	----	--

TABLE 3.5.1

CAMERA Commands

3.5.4 EYE

Purpose

MAGIC level command to transfer entire camera image from Vidicon to disk Grid. Places 256 by 256 array in lower left corner of Grid.

Useage

M:EYE lev B,W,T

Causes a bit to be turned on at the appropriate position on the disk Grid under the following conditions:

B,W,T equals the character 'B' and (video)* is less than (lev).

B,W,T equals c'W' and (video) is greater than (lev).

B,W,T equals c'T' and (video) equals (lev)

B,W,T are pneumonics that indicate the type of grey tone is to be saved, i.e., black, white, or threshold. "lev" sets a center point above which things are white and below which things are black.

Programs

HUNCH16 (or HUNCH 17) . LOADMOD and EYE.LOADMOD are the only programs required.

Execution

The following sequence should be followed;

```
M:HUNCH16 (HUNCH17)
M:EYE x x (Defaults to 4 B)
```

*(x) means 'the magnitude of x'

3.5.5 EYE2

PURPOSE

MAGIC level command to transfer entire camera image from Vidicon to disk Grid. Places 256 by 256 array in center of Grid.

USEAGE

Exactly like EYE.

3.6 MUSIC Structure

The overall procedure necessary to scan and convert a predrawn music score to a set of features that describe the structure of the score is accomplished in two steps. First, the score is placed before the camera lens and adjusted for best center placement on the monitor. The appropriate adjustments are made to the televison camera and interface. Then the EYE routines are used to transfer the score image to the disk. The MUSIC program is then invoked at the MAGIC level and output observed on the ARDS screen through a DISPLAY-GRID command. See Figure 3.6.1.

The depth of study necessary to fully understand this problem wouls surely fill a report much larger than mine. I desired only to construct the minimal structure necessary to convert the disk video data representation of a score into a series of parameters about the score that could be used in a ploy phonic synthesis of the notes. Such parameters as pitch and time would be essential in such an analysis. In the initial experiments, however, I chose to look only for pitch and only for whole notes. I took this course to reduce the size and complexity of the MUSIC Program and to make possible at least an unfinished solution to my original intent. Nonetheless, additional parameters finding routines could easily be added by extending the type of analysis routines used to identify the pitch of a note. The essence of the process is retained in the pitch finding routines, i.e., the process a human

observer might use in describing a musical text. Indeed, complete synthesis of many voiced orchestral text could be envisioned.

3.6.1 Assumptions

Briefly, I presumed the following conditions would be followed in the initial MUSIC experiment.

1) Only whole notes are allowed
2) Only one line of text is allowed
3) Staff lines must be as horizontal as possible.
4) No other musical construct is allowed except for notes.

3.6.2 MUSIC Program

This main FORTRAN Program performs several functions. First of all,it initializes and sets up the disk in accordance with GRID regulations. Subroutine STAFF is called to return the absolute x and y coordinates of the beginning and end of the staff lines (see figure 3.6.2). These coordinates are used in MUSIC to redraw the staff lines on the ARDS (see figure 3.6.3). NOTES is invoked to determine to determine the location of notes on and between staff lines. An entire horizontal scan occurs before moving up to the next half-staff interval (see figure 3.6.4). These absolute x and y coordinates are used to draw circles about the point the note was found an overlaying as much as possible the original note (see figures 3.6.5 and 3.6.6). A letter signifying the pitch of the note (G, A, or D for example) is drawn inside the circle. After the end of the last or top staff line is

reached, an exit to MAGIC is invoked.

3.6.3 STAFF Program

This routine extracts a slice of data that is five bits wide and as long as the maximun height of the score (in GRID coordinates and not to exceed 256). A linear search is performed on this array untill five distinct clusters of bits are found both at the beginning and end of the disk score. The xand y coordinates in addition to the width of the beginning and end of each line is returned to the main MUSIC program.

3.6.4 NOTES Program

This routine starts at a specified x and y location on the grid and performs linear density measurements in the x direction (see figure 3.6.4). A square window with a width equal to the distance between the nearest staff lines is taken and the number of bits set or turned on in that square is used to determine the prescence or abscence of a note at that particular spot. If the density does not exceed that expected of a staff line passing through the square, then the square is moved over one grid x value and another density measurment taken. If the density exceeds this minimun value but does not equal or exceed an average density expected of a whole note in that spot, the spot is ignored and the window again moved over one x value. When the average density is exceeded, a maximal density routine is used to determine more or less, the center position of the note. At the place maximal density is encountered, the x and y coordinates of the center of the window are returned in the array

NLOCX. In addition, the number of notes found along the initial x direction is returned.

CHAPTER IV EXTENTIONS

There are many additional features that could be added, both to the Vidicon interface and Camera disk routines to increase the speed, accuracy, and reliability of data transferal. Outlines below are some changes that would significantly achieve these goals;

1) automatic setup electronics to adjust the video interface for maximal contrast regardless of the camera settings
2) extension of the picture data base to 512 by 512
3) Multi-pass transfer of video data, that is converting many lines of data per video scan instead of just one.
4) Additional interface storage to increase information transferal rate.
5) Video Sample and Hold circuitry to increase resolution
6) Extension to eight levels of grey.
7) Use of additional grids to save all levels of grey.
8) Syncronization of the camera to the disk.
9) Programs to determine horizontal and vertical lines (data reduction).
10) Smearing routines to fill in places most likely missed by the interface.
11) Electro-mechanical systems to properly align the camera and monitor for best picture.
12) Video D/A electronics to vies converted data on-line.

The music program could easily be extended to resolve features of a score such as the following;

1) note meter finding.
2) extension to multi-staff scores
3) note playing interface with polyphonic output.
4) Tape storage of input scores.

CHAPTER V RESULTS and CONCLUSIONS

In examining discrete levels from the CAMERA program, I observed that at level zero a ring was formed on the ARDS screen roughly eliptical in nature and bearing no resemblance to the original drawing, regardless of the drawing, and therefore could be ignored by any other data routines. At level one I observed more of this ring in additional to features of the original drawing. This ring apparently is produced at the outer edges of the Vidicon scan and may be considered as noise only. Initially the step I took to reduce noise of this kind was to essentially "frame" the camera image , ignoring data falling outside the frame. Other noise reduction routines such as ANDing certain levels together were not attempted in favor of the more reliable "frame". Optimal results were obtained by ignoring the top and bottom quarter of the viewing field as well as the left and right eighth.

In viewing black on white pictures (such as that found in newsprint) most useful information contained in levels below four while white on black pictures produced the most useful information at levels above four. By "most useful" I mean the most recognizable reproduction on the ARDS.

These few routines resulted in images deemed clean enough to be used by GRID programs and extentions to it were belayed in favor of completion of MUSIC routines.

As far as note recognition goes,I feel a great deal more research could be done in trying to simulate a human virtuoso. In so doing I beleive one could learn a great deal

about the process of "teaching" people how to play musical instruments.

REFERENCES

1. Negroponte,Nicholas, personal communication

2. Lippman, Andy, personal communication

3. Herot, Christopher, personal communication.

4. Taggart, James, personal communication

5. Fortran IV Reference Manual, Publication Number B29-220, Interdata, Inc., 1970.

ILLUSTRATIONS

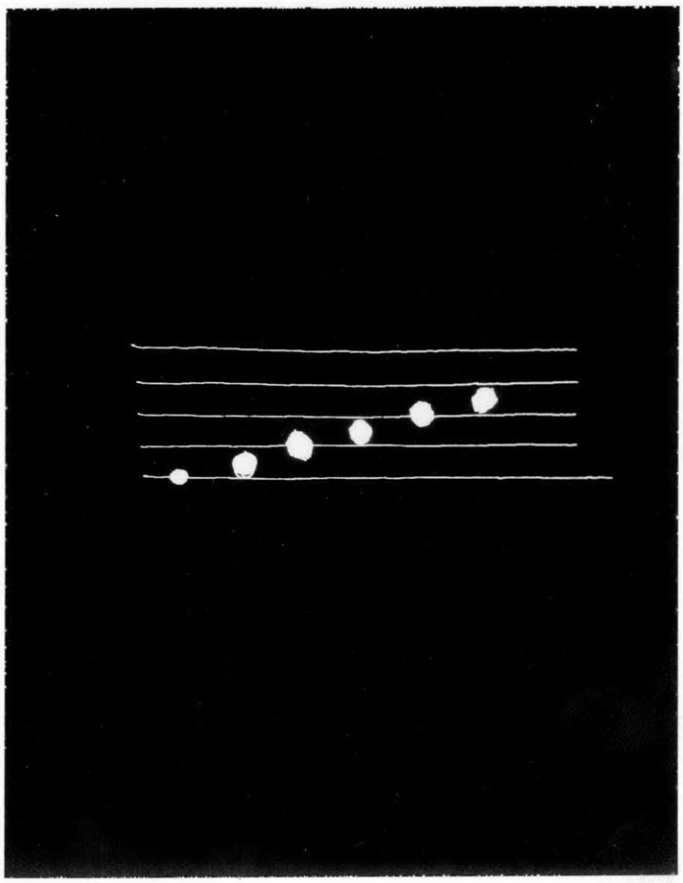


FIGURE 3.6.1
Original Drawing.

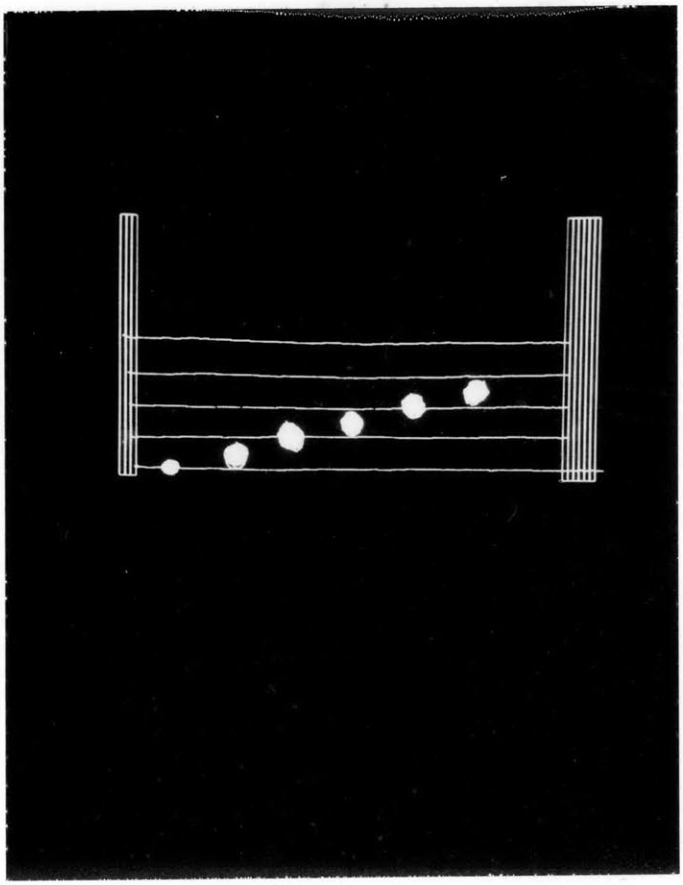


FIGURE 3.6.2
Looking for Staff lines.

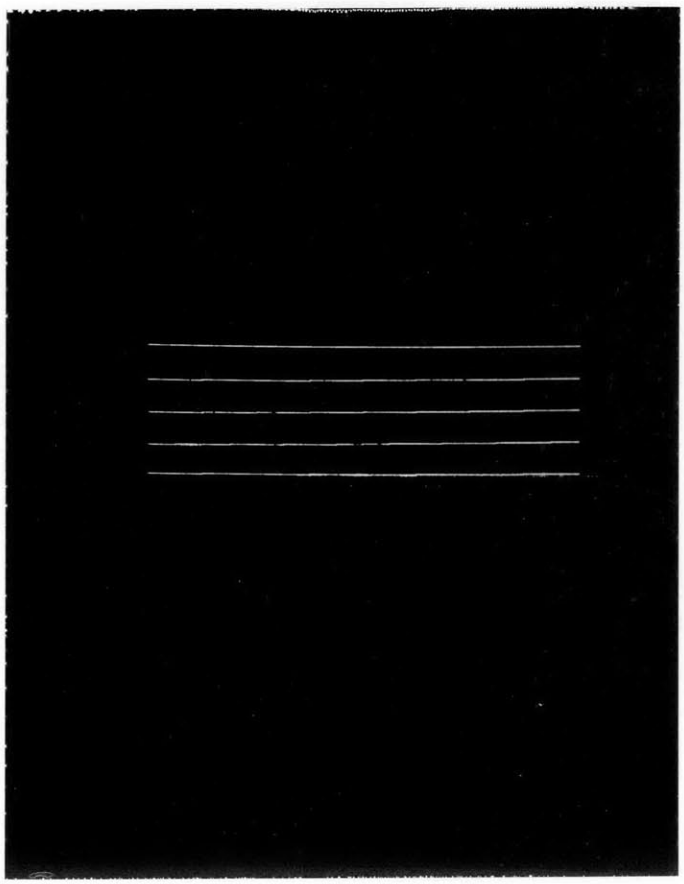


FIGURE 3.6.3
Staff lines found and redrawn.

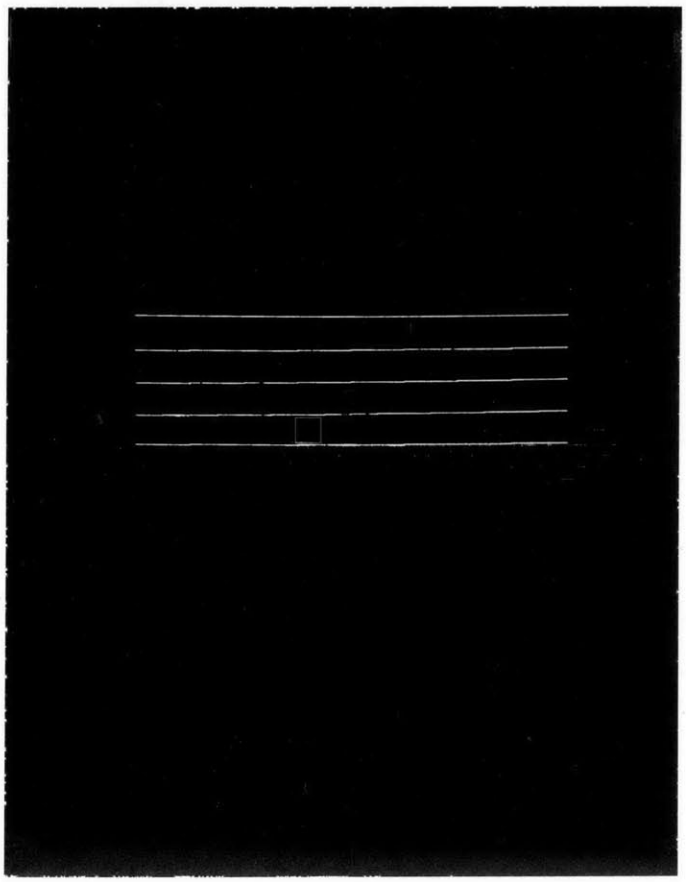


FIGURE 3.6.4
Density measurements initiated

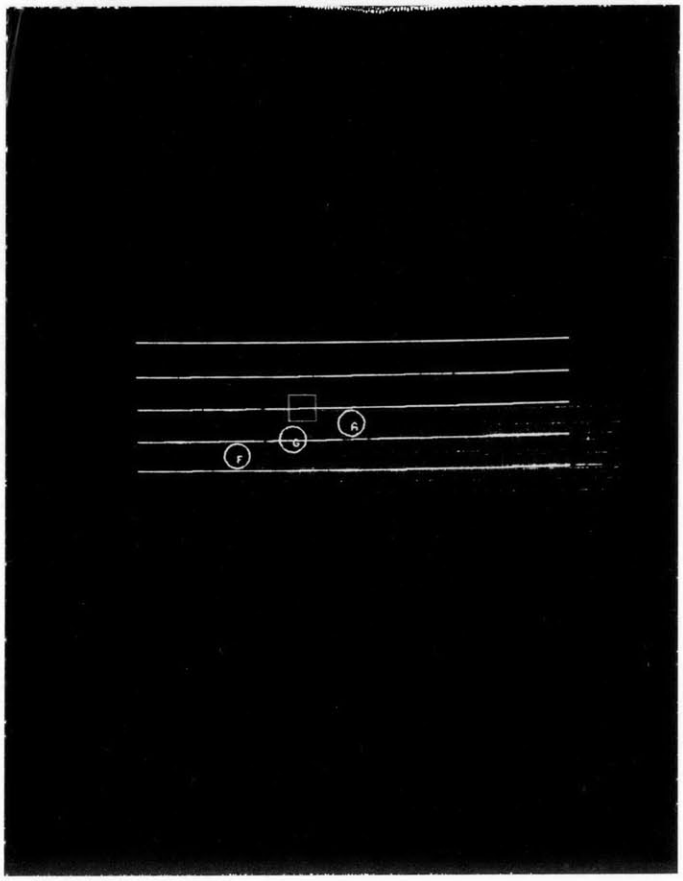


FIGURE 3.6.5
Notes found and labeled

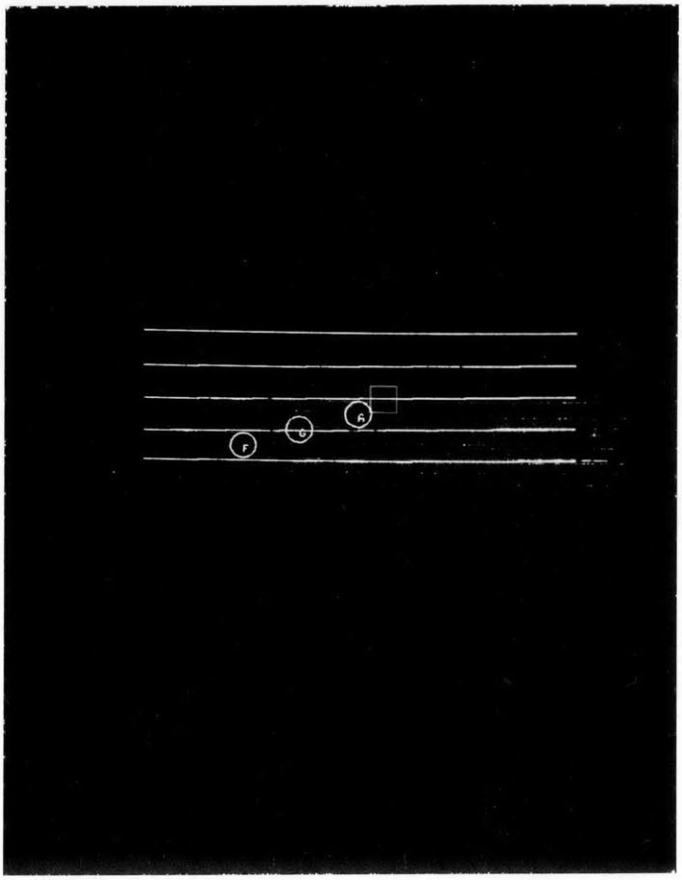


FIGURE 3.6.6
Search continues.

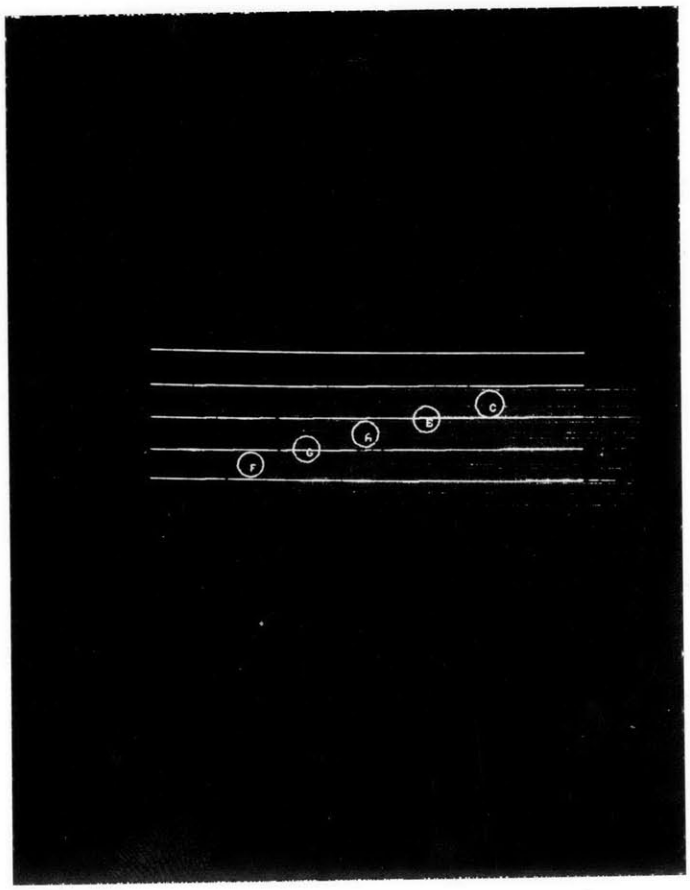


FIGURE 3.6.7
Final score analysis

APPENDIX A.

CAMERA Program Listings

```
DELETE EYE.LOADMOD
OSLOAD
OUT VIDEO-DISK
BI 2A62
LO CONSTANTS
LI EYE
LI GETLIN
LI DISK-MASK
LI INIGRD
LI CAMINZ
XOUT
EN
BIAS 2A62
LOAD VIDEO-DISK
CL 2B00
CREATEUSER EYE
DELETE VIDEO-DISK.TEXT

**********************************************************************

DELETE EYE2.LOADMOD
OSLOAD
OUT VIDEO-DISK
BI 2A62
LO CONSTANTS
LI EYE
LI GETLIN
LI DISK-MASK
LI INIGRD
LI CAMINZ
XOUT
EN
BIAS 2A62
LOAD VIDEO-DISK
CL 2B00
CREATEUSER EYE2
DELETE VIDEO-DISK.TEXT

**********************************************************************

DELETE DISPLAY-VIDEO.LOADMOD
OSLOAD
OUT CAMERA
BI 4000
LO VIDEO
LI CINDX
LI ALPHA
LI FIND
LI MTPAK
```

:

```
LI CAMINZ
LI SHOW
LI CAMPAK
XOUT
EN
BIAS 4000
LOAD CAMERA
CREATEUSER DISPLAY-VIDEO
DELETE CAMERA.TEXT


**********************************************************
```

```
00B8R                          END
ABUFF    0038R
BUFEND   00B7R
BUFFER   0034R
NXTLN    0020R
START    0000R
0000R C810     START   LHI    1,X'CF'
       00CF
0004R C820             LHI    2,C'20'
       3230
0008R C830             LHI    3,X'10'
       0010
000CR C840             LHI    4,1
       0001
0010R C850             LHI    5,0
       0000
0014R 9E12             OCR    1,2
0016R 9E12             OCR    1,2
0018R 9E14             OCR    1,4
001AR 9A14             WDR    1,4
001CR 9E12             OCR    1,2
001ER 9E12             OCR    1,2
0020R 9E13     NXTLN   OCR    1,3
0022R D710             RB     1,BUFFER
       0034R
0026R 0A54             AHR    5,4
0028R 9A45             WDR    4,5
002AR 9D17             SSR    1,7
002CR 4320             BFC    2,NXTLN
       0020R
0030R 4300             B      START
       0000R
0034R 0038R    BUFFER  DC     ABUFF
0036R 00B7R            DC     BUFEND
0038R          ABUFF   DS     128
00B7R          BUFEND  EQU    *-1
00B8R                  END
ABUFF    0038R
BUFEND   00B7R
BUFFER   0034R
NXTLN    0020R
START    0000R
```

```
  00D2R                    END
  ARDS     000C
  BLOWS    00B2R
* DOT      005CR
  EXIT     0072R
  FIRST    000B
  GS       007AR
  OUTREG   0024R
  POINT    0080R
  SAVER    000B
* SETPT    0000R
  SETUP    0096R
  SIGN     00CAR
  SRBUFF   0082R
  STAT     007ER
  VAL1     000D
  VAL2     000E
  VALOUT   000CR
* VECTOR   004CR
  WRITE    0078R
  WRTMOD   003AR
  XVECT    007CR
  0000R                    ENTRY SETPT,VECTOR,DOT
  *
  *REG 13 CONTAINS X VALUE IN ARDS COORDINATES
  *REG 14 CONTAINS Y VALUE (ARDS COOORDINATES)
  *
  * REGISTER ASSIGNMENTS
  000B            FIRST    EQU   11
  000C            ARDS     EQU   12
  000B            SAVER    EQU   11
  000D            VAL1     EQU   13
  000E            VAL2     EQU   14
  0000R D0B0      SETPT    STM   FIRST,SRBUFF
        0082R
  0004R 41F0               BAL   15,WRTMOD
        003AR
  0008R DAC0               WD    ARDS,GS
        007AR
  000CR 08BE      VALOUT   LHR   SAVER,VAL2
  000ER 41F0               BAL   15,SETUP
        0096R
  0012R 41F0               BAL   15,OUTREG
        0024R
  0016R 08DB               LHR   VAL1,SAVER
  0018R 41F0               BAL   15,SETUP
        0096R
  001CR 41F0               BAL   15,OUTREG
        0024R
  0020R 4300               B     EXIT
        0072R
  0024R DDC0      OUTREG   SS    ARDS,STAT
        007ER
  0028R 4290               BTC   9,*-4
```

```
      0024R
002CR 9ACD              WDR    ARDS,VAL1
002ER DDC0              SS     ARDS,STAT
      007ER
0032R 4290              BTC    9,*-4
      002ER
0036R 9ACE              WDR    ARDS,VAL2
0038R 030F              BR     15
003AR C8C0     WRTMOD   LHI    ARDS,X'A'
      000A
003ER DEC0              OC     ARDS,WRITE
      0078R
0042R DDC0              SS     ARDS,STAT
      007ER
0046R 4290              BTC    9,*-4
      0042R
004AR 030F              BR     15
*
004CR D0B0     VECTOR   STM    FIRST,SRBUFF
      0082R
0050R 41F0              BAL    15,WRTMOD
      003AR
0054R DAC0              WD     ARDS,XVECT
      007CR
0058R 4300              B      VALOUT
      000CR
*
005CR D0B0     DOT      STM    FIRST,SRBUFF
      0082R
0060R 41F0              BAL    15,WRTMOD
      003AR
*GO TO REG MODE
0064R 07DD              XHR    VAL1,VAL1
0066R D3E0              LB     VAL2,POINT
      0080R
006AR 41F0              BAL    15,OUTREG
      0024R
006ER 4300              B      EXIT
      0072R
*
0072R D1B0     EXIT     LM     FIRST,SRBUFF
      0082R
0076R 030F              BR     15
* CONSTANTS ETC...
0078R A8A8     WRITE    DC     X'A8A8'
007AR 1D1D     GS       DC     X'1D1D'
007CR 1E1E     XVECT    DC     X'1E1E'
007ER 0000     STAT     DC     *-*
*PERIOD
0080R 2E2E     POINT    DC     X'2E2E'
0082R          SRBUFF   DS     64-FIRST-FIRST-FIRST-FIRST
*
0096R 0BEE     SETUP    SHR    VAL2,VAL2
0098R 40E0              STH    VAL2,SIGN
```

```
        00CAR
009CR 08ED              LHR     VAL2,VAL1
009ER 4310              BNM     BLOWS
        00B2R
00A2R CBD0              SHI     VAL1,1
        0001
00A6R C7D0              XHI     VAL1,X'FFFF'
        FFFF
00AAR C8E0              LHI     VAL2,1
        0001
00AER 40E0              STH     VAL2,SIGN
        00CAR
00B2R 08ED      BLOWS   LHR     VAL2,VAL1
00B4R CCE0              SRHL    VAL2,5
        0005
00B8R C4E0              NHI     VAL2,X'1F'
        001F
00BCR C6E0              OHI     VAL2,X'40'
        0040
00C0R C4D0              NHI     VAL1,X'1F'
        001F
00C4R CDD0              SLHL    VAL1,1
        0001
00C8R C6D0              OHI     VAL1,0
        0000
00CAR           SIGN    EQU     *-2
00CCR C6D0              OHI     VAL1,X'40'
        0040
00D0R 030F              BR      15
00D2R                   END
  ARDS      000C
  BLOWS     00B2R
* DOT       005CR
  EXIT      0072R
  FIRST     000B
  GS        007AR
  OUTREG    0024R
  POINT     0080R
  SAVER     000B
* SETPT     0000R
  SETUP     0096R
  SIGN      00CAR
  SRBUFF    0082R
  STAT      007ER
  VAL1      000D
  VAL2      000E
  VALOUT    000CR
* VECTOR    004CR
  WRITE     0078R
  WRTMOD    003AR
  XVECT     007CR
```

```
    02C2R                                   END
  ADDR     0008
  BCOND    00EAR
  BEND     0211R
  BUFFER   018ER
* CAMINZ   0000R
  CBUFF    0009
  CEND     000B
  CLEAR    0096R
  DATA     0068R
  EOM      0146R
  EX       0130R
  EXIT     012AR
  FIRST    0006
  FS       028CR
  GET      00DAR
  INC      000A
  INIZ     0048R
  LAST     000E
  LEVEL    0040R
  LNSKIP   0092R
* MASK     0240R
* MASKZ    0216R
  MKEEP    0264R
  MLEV     028ER
  MOVE     0118R
  MSKEND   0299R
  MSKLUP   00F2R
  MSTOR    0280R
  NEXT     010AR
  ONE      009AR
  ONELN    0214R
  OUT      0134R
  PAD      000E
  RADR     007AR
  RBUFF    029AR
  READ     007CR
  SETINC   0098R
  SHIFNM   00E0R
  SHIFT    000C
  SMASK    0290R
  START    0192R
  STATS    0212R
  THRESH   0094R
  VADR     0132R
  VAL      000D
* VIDCON   009CR
  WHITE    0034R
  WRK1     0007
  WRK2     0006
  ZBUFF    0184R
  ZDONE    017CR
  ZERO     0150R
  ZIP      0168R
```

```
ZSTOR   016AR
*      CALLING SEQUENCE AS FOLLOWS:
*           CALL      CAMINZ(THRESHOLD,PRNTCODE)
*                      INITIALIZES THE CAMERA
*                      THRESHOLD AND PRNTCODE USED IN VIDCON
*
*           X = VIDCON(BUFF)
*                      TAKES 256*3 BIT VIDEO LINE AND CONVERTS TO
*                      32 BYTE (256 BIT)  BIT PER POINT ARRAY
*                      3 BIT VIDEO POINT IS COMPARED TO THRESHOLD
*                      AND A '1' BIT RETURNED IF ANY OF THE FOLL-
*                      OWING CONDITIONS HOLD TRUE
*                           1) WBL<0  AND VIDEO>THRESHOLD
*                           2) WBL=0  AND VIDEO<THRESHOLD
*                           3) WBL=0  AND VIDEO=THRESHOLD
*                      EACH CALL TO VIDCON INCREMENTS THE CAMERA TO
*                      THE NEXT LINE OF VIDEO DATA
*                      WHEN THE LAST LINE ISENCOUNTERED X=/0
*
*           CALL MASK(LEVEL)
*                EACH CALL TO MASK ADDS LEVEL TO A TABLE OF VIDEO
*                LEVELS THAT ARE TO BE IGNORED BY VIDCON.
*                     0<=LEVEL<=7
*
*           CALL MASKZ
*                      INITIALIZES MASK TABLLE
*
*
*      REGISTER ASSIGNMENTS
*
000E            PAD      EQU   14
000E            LAST     EQU   14
000D            VAL      EQU   13
000C            SHIFT    EQU   12
000B            CEND     EQU   11
000A            INC      EQU   10
0009            CBUFF    EQU   9
0008            ADDR     EQU   8
0007            WRK1     EQU   7
0006            WRK2     EQU   6
0006            FIRST    EQU   6
*
0000R                    ENTRY CAMINZ,VIDCON,MASK,MASKZ
*SAVE REGISTERS
0000R D0E0      CAMINZ   STM   LAST,RBUFF
      029AR
*RETURN INDEX FROM FORTRAN
0004R 48EF               LH    PAD,0(15)
      0000
0008R 40E0               STH   PAD,RADR
      007AR
*THRESHOLD
000CR 48EF               LH    PAD,2(15)
      0002
```

```
0010R 48EE              LH      PAD,0(PAD)
      0000
0014R D2E0              STB     PAD,THRESH
      0094R
*CONDITION CODE
0018R 48EF              LH      PAD,4(15)
      0004
001CR 48EE              LH      PAD,0(PAD)
      0000
0020R 4210              BM      WHITE
      0034R
0024R 4220              BP      LEVEL
      0040R
0028R C8E0              LHI     PAD,X'4380'
      4380
002CR 40E0              STH     PAD,BCOND
      00EAR
0030R 4300              B       INIZ
      0048R
0034R C8E0      WHITE   LHI     PAD,X'4280'
      4280
0038R 40E0              STH     PAD,BCOND
      00EAR
003CR 4300              B       INIZ
      0048R
0040R C8E0      LEVEL   LHI     PAD,X'4230'
      4230
0044R 40E0              STH     PAD,BCOND
      00EAR
*       START CAMERA OPERATIONS
0048R C8E0      INIZ    LHI     PAD,X'CF'
      00CF
*SYSTEM CLEAR
004CR DEE0              OC      PAD,CLEAR
      0096R
*JUST TO MAKE SURE
0050R DEE0              OC      PAD,CLEAR
      0096R
*SET INCREMENT
0054R DEE0              OC      PAD,SETINC
      0098R
*OF ONE
0058R DEE0              OC      PAD,ONE
      009AR
*CLEAR AGAIN
005CR DEE0              OC      PAD,CLEAR
      0096R
0060R DEE0              OC      PAD,CLEAR
      0096R
0064R C870              LHI     WRK1,30
      001E
*SKIP SOME LINES
0068R 41F0      DATA    BAL     15,READ
      007CR
```

```
006CR CB70              SHI    WRK1,1
      0001
0070R 4230              BNZ    DATA
      0068R
*     ALL DONE
0074R D1E0              LM     LAST,RBUFF
      029AR
0078R 430F              B      0(15)
      0000
007AR           RADR    EQU    *-2
* READD ROUTINR FOR CAMERA
007CR C8E0      READ    LHI    PAD,X'CF'
      00CF
0080R DEE0              OC     PAD,ONELN
      0214R
0084R D7E0              RB     PAD,BUFFER
      018ER
0088R DDE0              SS     PAD,STATS
      0212R
*LAST LINE
008CR 4220              BTC    2,EOM
      0146R
0090R 030F              BR     15
0092R 0000      LNSKIP  DC     *-*
0094R 0000      THRESH  DC     *-*
0096R 2020      CLEAR   DC     X'2020'
0098R 0101      SETINC  DC     X'0101'
009AR 0101      ONE     DC     X'0101'
*
009CR D060      VIDCON  STM    FIRST,RBUFF
      029AR
*RETURN INDEX
00A0R 48EF              LH     PAD,0(15)
      0000
00A4R 40E0              STH    PAD,VADR
      0132R
*CALLING BUFFER ADDRESS
00A8R 488F              LH     ADDR,2(15)
      0002
00ACR C8C0              LHI    SHIFT,4
      0004
*SET UP FOR BYTE HANDLIN
00B0R 40C0              STH    SHIFT,SHIFNM
      00E0R
00B4R 0BDD              SHR    VAL,VAL
*     GET BUFFER OF DATA
*GET DATA
00B6R 41F0              BAL    15,READ
      007CR
00BAR 41F0              BAL    15,ZERO
      0150R
00BER 0192R             DC     START
00C0R 0010              DC     16
00C2R 41F0              BAL    15,ZERO
```

```
      0150R
00C6R 0211R            DC     BEND
*IGNRE FIRST AND LAST 32 POINTS
00C8R FFF0             DC     -16
00CAR C890             LHI    CBUFF,START
      0192R
00CER C8A0             LHI    INC,1
      0001
00D2R C8B0             LHI    CEND,BEND
      0211R
00D6R C860             LHI    WRK2,15
      000F
*GET BYTE OF DATA
00DAR D3E9     GET     LB     PAD,0(CBUFF)
      0000
*THE FOUR GETS CHANGED
00DER CCE0             SRHL   PAD,4
      0004
00E0R          SHIFNM  EQU    *-2
*CUT OUT UPPER 4 BITS
00E2R C4E0             NHI    PAD,X'F'
      000F
00E6R D4E0             CLB    PAD,THRESH
      0094R
*BRANCH CONDITION HERE
00EAR 4200     BCOND   NOP    NEXT
      010AR
*CHECK THE MASK
00EER C870             LHI    WRK1,SMASK
      0290R
00F2R D4E7     MSKLUP  CLB    PAD,0(WRK1)
      0000
00F6R 4330             BE     NEXT
      010AR
00FAR CA70             AHI    WRK1,1
      0001
00FER C570             CLHI   WRK1,MSKEND
      0299R
0102R 4320             BNP    MSKLUP
      00F2R
*O.K. SET BIT
0106R C6D0             OHI    VAL,1
      0001
010AR 0866     NEXT    LHR    WRK2,WRK2
010CR 4330             BZ     OUT
      0134R
0110R CB60             SHI    WRK2,1
      0001
0114R CDD0             SLHL   VAL,1
      0001
0118R 48E0     MOVE    LH     PAD,SHIFNM
      00E0R
011CR 07EC             XHR    PAD,SHIFT
*CHANGE AROUND SHIFT NUMBER
```

```
011ER 40E0              STH    PAD,SHIFNM
      00E0R
*EXCEPT EVERY OTHER TIME
0122R 4330              BZ     GET
      00DAR
*ADD TO CBUFF ADDRESS
0126R C190              BXLE   CBUFF,GET
      00DAR
*NORMAL END
012AR D160     EXIT     LM     FIRST,RBUFF
      029AR
012ER 0BEE              SHR    14,14
*RETURN
0130R 430F     EX       B      0(15)
      0000
0132R          VADR     EQU    *-2
*STORE RESULT IN OUTPUT BUFFER
0134R 40D8     OUT      STH    VAL,0(ADDR)
      0000
0138R 0BDD              SHR    VAL,VAL
013AR C860              LHI    WRK2,15
      000F
013ER CA80              AHI    ADDR,2
      0002
*CFH ON HALLOWEEN
0142R 4300              B      MOVE
      0118R
*LAST LINE RETURN
0146R D160     EOM      LM     FIRST,RBUFF
      029AR
*IF 14=/0 ITS LAST LINE
014AR 08EA              LHR    14,INC
014CR 4300              B      EX
      0130R
0150R D0B0     ZERO     STM    11,ZBUFF
      0184R
*BUFFER ADDRESS
0154R 48EF              LH     14,0(15)
      0000
0158R C8B0              LHI    11,2
      0002
*NUMBER OF WORDS TO ZERO
015CR 48DF              LH     13,2(15)
      0002
0160R 4220              BP     ZIP
      0168R
0164R C8B0              LHI    11,-2
      FFFE
0168R 07CC     ZIP      XHR    12,12
016AR 40CE     ZSTOR    STH    12,0(14)
      0000
016ER 08DD              LHR    13,13
0170R 4330              BZ     ZDONE
      017CR
```

```
0174R 0AEB            AHR    14,11
0176R 0BDB            SHR    13,11
0178R 4300            B      ZSTOR
      016AR
017CR D1B0    ZDONE   LM     11,ZBUFF
      0184R
0180R 430F            B      4(15)
      0004
0184R         ZBUFF   DS     10
*BUFFER FOR READ BLOCK
018ER 0192R   BUFFER  DC     START
0190R 0211R           DC     BEND
0192R         START   DS     128
0211R         BEND    EQU    *-1
0212R         STATS   DS     2
0214R 1010    ONELN   DC     X'1010'
*
0216R D060    MASKZ   STM    FIRST,RBUFF
      029AR
*RETURN INDEX
021AR 48EF            LH     PAD,0(15)
      0000
021ER 40E0            STH    PAD,VADR
      0132R
0222R 0BEE            SHR    PAD,PAD
0224R D2E0            STB    PAD,SMASK
      0290R
0228R 48E0            LH     PAD,FS
      028CR
022CR D2E0            STB    PAD,SMASK+1
      0291R
0230R 40E0            STH    PAD,SMASK+2
      0292R
0234R 40E0            STH    PAD,SMASK+4
      0294R
0238R 40E0            STH    PAD,SMASK+6
      0296R
023CR 4300            B      EXIT
      012AR
*
0240R D060    MASK    STM    FIRST,RBUFF
      029AR
0244R 48EF            LH     PAD,0(15)
      0000
0248R 40E0            STH    PAD,VADR
      0132R
024CR 48EF            LH     PAD,2(15)
      0002
0250R 48EE            LH     PAD,0(PAD)
      0000
0254R D2E0            STB    PAD,MLEV
      028ER
0258R C890            LHI    CBUFF,SMASK
      0290R
```

```
025CR C8A0              LHI     INC,1
      0001
0260R C8B0              LHI     CEND,MSKEND
      0299R
0264R D3E9      MKEEP   LB      PAD,0(CBUFF)
      0000
0268R D4E0              CLB     PAD,MLEV
      028ER
026CR 4330              BE      EXIT
      012AR
0270R D4E0              CLB     PAD,FS
      028CR
0274R 4330              BE      MSTOR
      0280R
0278R C190              BXLE    CBUFF,MKEEP
      0264R
027CR 4300              B       EXIT
      012AR
0280R 48E0      MSTOR   LH      PAD,MLEV
      028ER
0284R D2E9              STB     PAD,0(CBUFF)
      0000
0288R 4300              B       EXIT
      012AR
028CR FFFF      FS      DC      X'FFFF'
028ER 0000      MLEV    DC      *-*
0290R 00FF      SMASK   DC      X'00FF'
0292R FFFF              DC      X'FFFF'
0294R FFFF              DC      X'FFFF'
0296R FFFF              DC      X'FFFF'
0298R FFFF              DC      X'FFFF'
0299R           MSKEND  EQU     *-1
029AR           RBUFF   DS      64-FIRST-FIRST-FIRST-FIRST
02C2R                   END
ADDR     0008
BCOND    00EAR
BEND     0211R
BUFFER   018ER
* CAMINZ  0000R
CBUFF    0009
CEND     000B
CLEAR    0096R
DATA     0068R
EOM      0146R
EX       0130R
EXIT     012AR
FIRST    0006
FS       028CR
GET      00DAR
INC      000A
INIZ     0048R
LAST     000E
LEVEL    0040R
LNSKIP   0092R
```

```
* MASK     0240R
* MASKZ    0216R
  MKEEP    0264R
  MLEV     028ER
  MOVE     0118R
  MSKEND   0299R
  MSKLUP   00F2R
  MSTOR    0280R
  NEXT     010AR
  ONE      009AR
  ONELN    0214R
  OUT      0134R
  PAD      000E
  RADR     007AR
  RBUFF    029AR
  READ     007CR
  SETINC   0098R
  SHIFNM   00E0R
  SHIFT    000C
  SMASK    0290R
  START    0192R
  STATS    0212R
  THRESH   0094R
  VADR     0132R
  VAL      000D
* VIDCON   009CR
  WHITE    0034R
  WRK1     0007
  WRK2     0006
  ZBUFF    0184R
  ZDONE    017CR
  ZERO     0150R
  ZIP      0168R
  ZSTOR    016AR
```

```
  032CR                    END
  BADMES  0048R
  BLACK   006AR
* BUFDMP  00DCR
  BUFF    00C6R
  BWT     030AR
* CAMINZ  00A6R
* DISKZ   00A2R
  DOOIT   0078R
* EYE     0000R
  FIN     00D6R
  FINIS   00DER
  GCB     00E8R
* GETLIN  00BAR
* HIGHX   008AR
* HIGHY   008ER
* INIGRD  0092R
  LINE    0000
  LOOP    00B2R
* LOWX    007ER
* LOWY    0082R
  R10     000A
  R11     000B
  R12     000C
  R13     000D
  R14     000E
  R15     000F
  SAVE    030CR
  SIZE    0306R
  STORE   0074R
  TAPE    00E6R
  THR     0070R
  THRESH  0308R
* VIDCON  00C2R
  WHITE   0062R
  *     USED TO PUT CAMERA IMAGE IN LOWER
  *     LEFT HAND CORNER OF GRID
  *
  0000R                    ENTRY EYE
  0000R                    EXTRN BUFDMP
  0000R                    EXTRN CAMINZ,VIDCON
  0000R                    EXTRN INIGRD
  0000R                    EXTRN DISKZ,GETLIN
  0000R                    EXTRN LOWX,LOWY,HIGHX,HIGHY
  *
  0000            LINE     EQU   0
  000A            R10      EQU   10
  000B            R11      EQU   11
  000C            R12      EQU   12
  000D            R13      EQU   13
  000E            R14      EQU   14
  000F            R15      EQU   15
  *
  0000R D000      EYE      STM   0,SAVE
```

```
      030CR
*ADDRESS OF COMMAND ARGUMENT (THRESHOLD)
0004R 48A1              LH      R10,4(1)
      0004
*DEFAULT IF NO ARG
0008R 4330              BZ      DOOIT
      0078R
000CR 4AA1              AH      R10,6(1)
      0006
*THRESHOLD
0010R D3AA              LB      R10,-1(R10)
      FFFF
*CONVERT TO HEX
0014R C4A0              NHI     R10,X'000F'
      000F
0018R 40A0              STH     R10,THRESH
      0308R
*ADDRESS OF W,B,T
001CR 48A1              LH      R10,8(1)
      0008
0020R 4330              BZ      DOOIT
      0078R
*
0024R D3AA              LB      R10,0(R10)
      0000
*'W'=WHITE PICTURE
0028R C5A0              CLHI    R10,X'0057'
      0057
002CR 4330              BE      WHITE
      0062R
*'B'=BLACK PICTURE
0030R C5A0              CLHI    R10,X'0042'
      0042
0034R 4330              BE      BLACK
      006AR
*'T'=THRESHOLD PICTURE
0038R C5A0              CLHI    R10,X'0054'
      0054
003CR 4330              BE      THR
      0070R
*
*MUST BE A BAD ARGUMENT
0040R E120              SVC     2,BADMES
      0048R
0044R 4300              B       FINIS
      00DER
*
0048R 0007    BADMES    DC      7,22
      0016
*INVALID ARGUMENT '
004CR 4559              DC      C'EYE:
      453A
      2049
      4E56
```

```
       414C
       4944
       2041
       5247
       554D
       454E
       5420
*
0062R C8A0     WHITE   LHI    R10,-1
       FFFF
0066R 4300             B      STORE
       0074R
006AR 07AA     BLACK   XHR    R10,R10
006CR 4300             B      STORE
       0074R
0070R C8A0     THR     LHI    R10,1
       0001
0074R 40A0     STORE   STH    R10,BWT
       030AR
*
0078R C8A0     DOOIT   LHI    R10,-2048
       F800
007CR 40A0             STH    R10,LOWX
       0000F
0080R 40A0             STH    R10,LOWY
       0000F
0084R C8A0             LHI    R10,-1016
       FC08
0088R 40A0             STH    R10,HIGHX
       0000F
008CR 40A0             STH    R10,HIGHY
       0000F
*
0090R 41F0             BAL    R15,INIGRD
       0000F
0094R 0008             DC     8
0096R 00E6R            DC     A(TAPE)
0098R 00E8R            DC     A(GCB)
009AR 0306R            DC     A(SIZE)
*
009CR C8F0             LHI    R15,GCB
       00E8R
*ZERO THE DISK
00A0R 41E0             BAL    R14,DISKZ
       0000F
*
00A4R 41F0             BAL    R15,CAMINZ
       0000F
00A8R 0006             DC     6
00AAR 0308R            DC     A(THRESH)
00ACR 030AR            DC     A(BWT)
*
00AER C8A0             LHI    R10,256
       0100
```

```
00B2R C8F0     LOOP     LHI    R15,GCB
       00E8R
00B6R 080A              LHR    LINE,R10
00B8R 41E0              BAL    R14,GETLIN
       0000F
00BCR 4000              STH    LINE,BUFF
       00C6R
00C0R 41F0              BAL    R15,VIDCON
       0000F
00C4R 0004              DC     4
00C6R          BUFF     DS     2
*BNZ FINIS   CHECK J.E.RETURN CODE
00C8R 08EE              LHR    R14,R14
00CAR 4230              BNZ    FIN
       00D6R
00CER CBA0              SHI    R10,1
       0001
00D2R 4300              B      LOOP
       00B2R
*
*
00D6R C8F0     FIN      LHI    R15,GCB
       00E8R
00DAR 4190              BAL    9,BUFDMP
       0000F
00DER D100     FINIS    LM     0,SAVE
       030CR
00E2R 07FF              XHR    R15,R15
00E4R 030E              BR     R14
*
*
00E6R          TAPE     DS     2
00E8R          GCB      DS     542
0306R 0004     SIZE     DC     4
0308R 0004     THRESH   DC     4
030AR 0000     BWT      DC     0
*
030CR          SAVE     DS     32
032CR                   END
BADMES   0048R
BLACK    006AR
* BUFDMP  00DCR
BUFF     00C6R
BWT      030AR
* CAMINZ  00A6R
* DISKZ   00A2R
DOOIT    0078R
* EYE     0000R
FIN      00D6R
FINIS    00DER
GCB      00E8R
* GETLIN  00BAR
* HIGHX   008AR
* HIGHY   008ER
```

```
* INIGRD   0092R
  LINE     0000
  LOOP     00B2R
* LOWX     007ER
* LOWY     0082R
  R10      000A
  R11      000B
  R12      000C
  R13      000D
  R14      000E
  R15      000F
  SAVE     030CR
  SIZE     0306R
  STORE    0074R
  TAPE     00E6R
  THR      0070R
  THRESH   0308R
* VIDCON   00C2R
  WHITE    0062R
```

```
  0072R                    END
  ADONE    004AR
* ALPHA    0000R
  CR       0066R
  ERETT    005AR
  MORE     000CR
  RRRR     006AR
  SPAC2    0062R
  T3       0064R
  TEN      0068R
  ZRO      000AR
  *          CALLING SEQUENCE
  *               BAL     15,ALPHA
  *               DC      A(ARG)      ADDRESS OF ARG
  *               DC      A(NUM)      ADDRESS OF ARG
  *          RETURNS TO (15) + 8 IF ERROR FOUND
  0000R                    ENTRY ALPHA
  *SAVE REGISTERS
  0000R D0C0    ALPHA      STM    12,RRRR
        006AR
  *ARG
  0004R 48EF               LH     14,0(15)
        0000
  *NUMBER REGISTER
  0008R C8C0               LHI    12,0
        0000
  *THE NUMBER ZERO
  000AR          ZRO       EQU    *-2
  *GET DATA
  000CR D3DE     MORE      LB     13,0(14)
        0000
  *EXIT ON SPACE
  0010R D4D0               CLB    13,SPAC2
        0062R
  0014R 4330               BE     ADONE
        004AR
  *EXIT ON C
  0018R D4D0               CLB    13,CR
        0066R
  001CR 4330               BE     ADONE
        004AR
  *SHIFTT OUT REAL NUMBER
  0020R CCD0               SRHL   13,4
        0004
  *MAKE SURE ITS ALPHANUMERIC
  0024R D4D0               CLB    13,T3
        0064R
  *ERROR RETURN
  0028R 4230               BNE    ERETT
        005AR
  *GET SET UP FOR MULTIPLY
  002CR 08DC               LHR    13,12
  *ZERO OUT MSB
  002ER 48C0               LH     12,ZRO
```

```
      000AR
*MULTIPLY BY TEN
0032R 4CC0              MH      12,TEN
      0068R
*ANSWER IN 13
0036R 08CD              LHR     12,13
*GEET REAL NUMBER
0038R D3DE              LB      13,0(14)
      0000
*AND OUT TOP 4 BITS
003CR C4D0              NHI     13,X'0F'
      000F
*ADD NEW NUMBER
0040R 0ACD              AHR     12,13
*GO GET MORE NUMBER
0042R CAE0              AHI     14,1
      0001
0046R 4300              B       MORE
      000CR
*GET NUMBER ADDRESS
004AR 48DF    ADONE     LH      13,2(15)
      0002
*RETURN NUMBER
004ER 40CD              STH     12,0(13)
      0000
0052R D1C0              LM      12,RRRR
      006AR
*GOOD RETURN
0056R 430F              B       8(15)
      0008
005AR D1C0    ERETT     LM      12,RRRR
      006AR
*ERROR RETURN
005ER 430F              B       4(15)
      0004
*SPACE
0062R 2000    SPAC2     DC      X'2000'
0064R 0300    T3        DC      X'0300'
*CR
0066R 0D00    CR        DC      X'0D00'
0068R 000A    TEN       DC      10
006AR         RRRR      DS      8
0072R                   END
ADONE   004AR
* ALPHA  0000R
CR      0066R
ERETT   005AR
MORE    000CR
RRRR    006AR
SPAC2   0062R
T3      0064R
TEN     0068R
ZR0     000AR
```

```
  0330R                    END
  BADMES  0048R
  BLACK   006AR
* BUFDMP  00E0R
  BUFF    00CAR
  BWT     030ER
* CAMINZ  00A6R
* DISKZ   00A2R
  DOOIT   0078R
* EYE     0000R
  FIN     00DAR
  FINIS   00E2R
  GCB     00ECR
* GETLIN  00BAR
* HIGHX   008AR
* HIGHY   008ER
* INIGRD  0092R
  LINE    0000
  LOOP    00B2R
* LOWX    007ER
* LOWY    0082R
  R10     000A
  R11     000B
  R12     000C
  R13     000D
  R14     000E
  R15     000F
  SAVE    0310R
  SIZE    030AR
  STORE   0074R
  TAPE    00EAR
  THR     0070R
  THRESH  030CR
* VIDCON  00C6R
  WHITE   0062R
  *
  *     USED TO PUT CAMERA IMAGE IN CENTER OF GRID
  0000R                    ENTRY EYE
  0000R                    EXTRN BUFDMP
  0000R                    EXTRN CAMINZ,VIDCON
  0000R                    EXTRN INIGRD
  0000R                    EXTRN DISKZ,GETLIN
  0000R                    EXTRN LOWX,LOWY,HIGHX,HIGHY
  *
  0000            LINE     EQU   0
  000A            R10      EQU   10
  000B            R11      EQU   11
  000C            R12      EQU   12
  000D            R13      EQU   13
  000E            R14      EQU   14
  000F            R15      EQU   15
  *
  0000R D000      EYE      STM   0,SAVE
        0310R
```

```
*ADDRESS OF COMMAND ARGUMENT (THRESHOLD)
0004R 48A1              LH      R10,4(1)
      0004
*DEFAULT IF NO ARG
0008R 4330              BZ      DOOIT
      0078R
000CR 4AA1              AH      R10,6(1)
      0006
*THRESHOLD
0010R D3AA              LB      R10,-1(R10)
      FFFF
*CONVERT TO HEX
0014R C4A0              NHI     R10,X'000F'
      000F
0018R 40A0              STH     R10,THRESH
      030CR
*ADDRESS OF W,B,T
001CR 48A1              LH      R10,8(1)
      0008
0020R 4330              BZ      DOOIT
      0078R
*
0024R D3AA              LB      R10,0(R10)
      0000
*'W'=WHITE PICTURE
0028R C5A0              CLHI    R10,X'0057'
      0057
002CR 4330              BE      WHITE
      0062R
*'B'=BLACK PICTURE
0030R C5A0              CLHI    R10,X'0042'
      0042
0034R 4330              BE      BLACK
      006AR
*'T'=THRESHOLD PICTURE
0038R C5A0              CLHI    R10,X'0054'
      0054
003CR 4330              BE      THR
      0070R
*
*MUST BE A BAD ARGUMENT
0040R E120              SVC     2,BADMES
      0048R
0044R 4300              B       FINIS
      00E2R
*
0048R 0007    BADMES    DC      7,22
      0016
*INVALID ARGUMENT '
004CR 4559              DC      C'EYE:
      453A
      2049
      4E56
      414C
```

```
       4944
       2041
       5247
       554D
       454E
       5420
*
0062R C8A0     WHITE   LHI    R10,-1
       FFFF
0066R 4300             B      STORE
       0074R
006AR 07AA     BLACK   XHR    R10,R10
006CR 4300             B      STORE
       0074R
0070R C8A0     THR     LHI    R10,1
       0001
0074R 40A0     STORE   STH    R10,BWT
       030ER
*
0078R C8A0     DOOIT   LHI    R10,-518
       FDFA
007CR 40A0             STH    R10,LOWX
       0000F
0080R 40A0             STH    R10,LOWY
       0000F
0084R C8A0             LHI    R10,518
       0206
0088R 40A0             STH    R10,HIGHX
       0000F
008CR 40A0             STH    R10,HIGHY
       0000F
*
0090R 41F0             BAL    R15,INIGRD
       0000F
0094R 0008             DC     8
0096R 00EAR            DC     A(TAPE)
0098R 00ECR            DC     A(GCB)
009AR 030AR            DC     A(SIZE)
*
009CR C8F0             LHI    R15,GCB
       00ECR
*ZERO THE DISK
00A0R 41E0             BAL    R14,DISKZ
       0000F
*
00A4R 41F0             BAL    R15,CAMINZ
       0000F
00A8R 0006             DC     6
00AAR 030CR            DC     A(THRESH)
00ACR 030ER            DC     A(BWT)
*
*START AT LINE 640
00AER C8A0             LHI    R10,X'280'
       0280
```

```
00B2R C8F0     LOOP     LHI    R15,GCB
      00ECR
00B6R 080A              LHR    LINE,R10
00B8R 41E0              BAL    R14,GETLIN
      0000F
*TELL JEFF TO START AT MIDDLE OF SCREEN
00BCR CA00              AHI    LINE,48
      0030
00C0R 4000              STH    LINE,BUFF
      00CAR
00C4R 41F0              BAL    R15,VIDC0N
      0000F
00C8R 0004              DC     4
00CAR          BUFF     DS     2
*BNZ FINIS  CHECK J.E.RETURN CODE
00CCR 08EE              LHR    R14,R14
00CER 4230              BNZ    FIN
      00DAR
00D2R CBA0              SHI    R10,1
      0001
00D6R 4300              B      LOOP
      00B2R
*
*
00DAR C8F0     FIN      LHI    R15,GCB
      00ECR
00DER 4190              BAL    9,BUFDMP
      0000F
00E2R D100     FINIS    LM     0,SAVE
      0310R
00E6R 07FF              XHR    R15,R15
00E8R 030E              BR     R14
*
*
00EAR          TAPE     DS     2
00ECR          GCB      DS     542
030AR 0004     SIZE     DC     4
030CR 0004     THRESH   DC     4
030ER 0000     BWT      DC     0
*
0310R          SAVE     DS     32
0330R                   END
BADMES  0048R
BLACK   006AR
* BUFDMP  00E0R
BUFF    00CAR
BWT     030ER
* CAMINZ  00A6R
* DISKZ   00A2R
DOOIT   0078R
* EYE     0000R
FIN     00DAR
FINIS   00E2R
GCB     00ECR
```

```
* GETLIN  00BAR
* HIGHX   008AR
* HIGHY   008ER
* INIGRD  0092R
  LINE    0000
  LOOP    00B2R
* LOWX    007ER
* LOWY    0082R
  R10     000A
  R11     000B
  R12     000C
  R13     000D
  R14     000E
  R15     000F
  SAVE    0310R
  SIZE    030AR
  STORE   0074R
  TAPE    00EAR
  THR     0070R
  THRESH  030CR
* VIDCON  00C6R
  WHITE   0062R
```

```
  0172R                    END
  BBUFF    0130R
  BITS     000E
  BITTST   00EAR
  BITVAL   0136R
  BUFADR   0052R
* DLT      00C0R
  DRAW     00ACR
  DRDIT    00BER
  EVEN     000C
  EX       0044R
  DXIT     0138R
  EXIT1    0104R
  EXIT2    0108R
  FIRST    0006
  FLAG     0008
  GT       009CR
  INC      000C
  LOCK     00A0R
  MASK     000D
  MORE     0082R
  NEW      0114R
  NFOUND   0124R
  ODD      000D
  ONE      0009
  PAD      0007
  RBUFF    014AR
  RETADR   0112R
  SET      0090R
* SETPT    009AR
  SETX     0148R
* SHOW     0054R
  START    0078R
  SWORD    00E2R
  TADR     004AR
* TELL     0000R
* VECTOR   00B8R
  VWORD    00CER
  WRK1     0006
  XADR     000A
  XEND     000B
  XINC     004CR
  XPOS     000D
  XSET     00C2R
  YINC     004ER
  YVAL     0050R
  **
  * CALLING SEQUENCE AS FOLLOWS:
  *
  *      CALL TELL(XBUFF,XDEM,YDEM)
  *           PARAMETERS ARE USED IN SHOW:
  *                XBUFF-LOCATION OF PLOTTABLE BUFFER
  *                XDEM-SIZE OF BUFFER IN BITS
  *                YDEM NUMBER OF LINES TO BE DRAWN
```

```
*
*      CALL SHOW
*           TEH ARRAY 'BUFF' IS SCANNED FOR
*           CONTINGUOUS BIT PATTERNS.
*           THE ARRAY IS PLOTTED SUCH THAT IF ALL BITS
*           WERE ON ('1') A LINE WOULD OCCUR ON THE ARDS
*           OF MAXIMUN LENGHT (APPREOXIATLY 8.5 INCHES)
*           AND IN THE HORIZONTAL PLANE.EACH BIT LOCATION
*           HAS A CORESSPONDING LOCATION ON THE ARDS.
*           IF ONLY ONE BIT OCCURS, A PERIOD IS DRAWN
*           AT THE CORESSPONDING LOCATION ON THE ARDS.
*           THE INITIAL CALL TO TELL, POSITIONS THE
*           BEAM TO THE UPPER LEFT HAND CORNER OF THE
*           SCREEN.SUBSEQUENT CALL TO SHOW,CAUSES
*           THE 'Y' LOCATION OF THE BEAM TO BE LOW-
*           ERED ABOUT (1024/YDEM)*.007 INCHES.
*           SHOW TRIES TO PAINT A 6.5 BY 8.5
*           IMAGE AS QUICKLY AS POSSIBLE.
*
*
*      REGISTER ASSIGNMENTS
0007            PAD     EQU   7
0008            FLAG    EQU   8
0009            ONE     EQU   9
000A            XADR    EQU   10
000B            XEND    EQU   11
000C            INC     EQU   12
000C            EVEN    EQU   12
000D            ODD     EQU   13
000D            XPOS    EQU   13
0006            WRK1    EQU   6
0006            FIRST   EQU   6
*
*               START TELL
*
0000R                   EXTRN SETPT,VECTOR,DOT
0000R                   ENTRY SHOW,TELL
0000R D060      TELL    STM   FIRST,RBUFF
      014AR
*RETURN ADDRESS
0004R 487F              LH    PAD,0(15)
      0000
0008R 4070              STH   PAD,TADR
      004AR
000CR 487F              LH    PAD,2(15)
      0002
*SAVE BUFFER ADDRESS
0010R 4070              STH   PAD,BUFADR
      0052R
*GET XDEM
0014R 487F              LH    PAD,4(15)
      0004
0018R 4877              LH    PAD,0(PAD)
      0000
```

```
001CR 0BCC              SHR     EVEN,EVEN
*SET UP TO FIGURE INC
001ER C8D0              LHI     ODD,1024
      0400
0022R 0DC7              DHR     EVEN,PAD
0024R 40D0              STH     ODD,XINC
      004CR
0028R 487F              LH      PAD,6(15)
      0006
*FIGURE YINC
002CR 4877              LH      PAD,0(PAD)
      0000
0030R 0BCC              SHR     EVEN,EVEN
0032R C8D0              LHI     ODD,1024
      0400
0036R 0DC7              DHR     EVEN,PAD
0038R 40D0              STH     ODD,YINC
      004ER
*STARTING Y VALUE
003CR C870              LHI     PAD,700
      02BC
0040R 4070              STH     PAD,YVAL
      0050R
0044R D160      EX      LM      FIRST,RBUFF
      014AR
0048R 430F              B       0(15)
      0000
004AR           TADR    EQU     *-2
004CR 0000      XINC    DC      *-*
004ER 0000      YINC    DC      *-*
0050R 0000      YVAL    DC      *-*
0052R 0000      BUFADR  DC      *-*
*
*                  START SHOW
*
0054R D060      SHOW    STM     FIRST,RBUFF
      014AR
0058R 487F              LH      PAD,0(15)
      0000
005CR 4070              STH     PAD,TADR
      004AR
*STARTING XVAL
0060R C8D0              LHI     XPOS,-512
      FE00
0064R 48C0              LH      INC,XINC
      004CR
*1023
0068R C8B0              LHI     XEND,511
      01FF
006CR 48A0              LH      XADR,BUFADR
      0052R
0070R C870              LHI     PAD,X'8000'
      8000
0074R 4070              STH     PAD,BITVAL
```

```
0078R 48EA     START   LH      14,0(XADR)
      0000
007CR 09DB             CHR     XPOS,XEND
007ER 4220             BP      EXIT
      0138R
0082R 41F0     MORE    BAL     15,BITTST
      00EAR
0086R 00E2R            DC      SWORD
0088R 0090R            DC      SET
008AR 0ADC             AHR     XPOS,INC
008CR 4300             B       MORE
      0082R
0090R 48E0     SET     LH      14,YVAL
      0050R
0094R 40D0             STH     XPOS,SETX
      0148R
0098R 41F0             BAL     15,SETPT
      0000F
009CR 48EA     GO      LH      14,0(XADR)
      0000
00A0R 0ADC     LOOK    AHR     XPOS,INC
00A2R 41F0             BAL     15,BITTST
      00EAR
00A6R 00CER            DC      VWORD
00A8R 00A0R            DC      LOOK
00AAR 0BDC             SHR     XPOS,INC
00ACR 4BD0     DRAW    SH      XPOS,SETX
      0148R
00B0R 4330             BZ      DRDOT
      00BER
00B4R 07EE             XHR     14,14
00B6R 41F0             BAL     15,VECTOR
      0000F
00BAR 4300             B       XSET
      00C2R
00BER 41F0     DRDOT   BAL     15,DOT
      0000F
00C2R 4AD0     XSET    AH      XPOS,SETX
      0148R
00C6R 0ADC             AHR     XPOS,INC
00C8R 0ADC             AHR     XPOS,INC
00CAR 4300             B       START
      0078R
00CER CAA0     VWORD   AHI     XADR,2
      0002
00D2R 09DB             CHR     XPOS,XEND
00D4R 4220             BP      DRAW
      00ACR
00D8R 4330             BE      DRAW
      00ACR
00DCR 0BDC             SHR     XPOS,INC
00DER 4300             B       GO
      009CR
00E2R CAA0     SWORD   AHI     XADR,2
```

```
      0002
00E6R 4300              B       START
      0078R
*     BITTST HERE
*
*
*PAGE   4


*     REGISTER ASSIGNMENT
000D            MASK    EQU     13
000E            BITS    EQU     14
*
00EAR D0D0      BITTST  STM     MASK,BBUFF
      0130R
00EER 48D0              LH      MASK,BITVAL
      0136R
00F2R 4330              BZ      NEW
      0114R
00F6R 04ED              NHR     BITS,MASK
00F8R 4330              BZ      NFOUND
      0124R
00FCR 48EF              LH      BITS,2(15)
      0002
0100R 40E0              STH     BITS,RETADR
      0112R
0104R CCD0      EXIT1   SRHL    MASK,1
      0001
0108R 40D0      EXIT2   STH     MASK,BITVAL
      0136R
010CR D1D0              LM      MASK,BBUFF
      0130R
0110R 4300              B       *-*
      0000
0112R           RETADR  EQU     *-2
0114R 48EF      NEW     LH      BITS,0(15)
      0000
0118R 40E0              STH     BITS,RETADR
      0112R
011CR C8D0              LHI     MASK,X'8000'
      8000
0120R 4300              B       EXIT2
      0108R
0124R CAF0      NFOUND  AHI     15,4
      0004
0128R 40F0              STH     15,RETADR
      0112R
012CR 4300              B       EXIT1
      0104R
0130R           BBUFF   DS      6
0136R 8000      BITVAL  DC      X'8000'
*UPDATE NEW LINE LOCATION
0138R 4870      EXIT    LH      PAD,YVAL
      0050R
013CR 4B70              SH      PAD,YINC
```

```
         004ER
  0140R  4070                STH     PAD,YVAL
         0050R
  0144R  4300                B       EX
         0044R
  0148R  0000      SETX      DC      *-*
  014AR            RBUFF     DS      64-FIRST-FIRST-FIRST-FIRST
  0172R                      END
  BBUFF    0130R
  BITS     000E
  BITTST   00EAR
  BITVAL   0136R
  BUFADR   0052R
* DOT      00C0R
  DRAW     00ACR
  DRDOT    00BER
  EVEN     000C
  EX       0044R
  EXIT     0138R
  EXIT1    0104R
  EXIT2    0108R
  FIRST    0006
  FLAG     0008
  GO       009CR
  INC      000C
  LOOK     00A0R
  MASK     000D
  MORE     0082R
  NEW      0114R
  NFOUND   0124R
  ODD      000D
  ONE      0009
  PAD      0007
  RBUFF    014AR
  RETADR   0112R
  SET      0090R
* SETPT    009AR
  SETX     0148R
* SHOW     0054R
  START    0078R
  SWORD    00E2R
  TADR     004AR
* TELL     0000R
* VECTOR   00B8R
  VWORD    00CER
  WRK1     0006
  XADR     000A
  XEND     000B
  XINC     004CR
  XPOS     000D
  XSET     00C2R
  YINC     004ER
  YVAL     0050R
```

```
  006ER                    END
* CINDX    0000R
  D2       006CR
  IDONE    004ER
  IERR     0046R
  IUPDAT   0012R
  02       000E
  RESTOR   003AR
  RRBF     0060R
  T1       0010R
  T2       001ER
  *      CALLING SEQUENCE AS FOLLOWS
  *           BAL CINDX,15
  *           DC   A(ARG)
  *           DC   A(TABLE)
  *           DC   A(INDEX)
  *
  *      ARG IS TERMINATED BY X'0D' - ONE BYTE
  *      TABLE TERMINATED BY CR X'0D'
  *      RETURNS TO (15) + 6 IF ARG IS NOT FOUND
  *           RETURNS TO (15) + 10 IF ARG IS FOUND
  0000R                    ENTRY CINDX
  0000R D0A0     CINDX     STM    10,RRBF
        0060R
  0004R 48DF               LH     13,0(15)
        0000
  0008R 48CF               LH     12,2(15)
        0002
  000CR C8E0               LHI    14,1
        0001
  0010R 08BC     T1        LHR    11,12
  *CHECK FOR END OF TABLE X'0D'
  0012R D3AC     IUPDAT    LB     10,0(12)
        0000
  0016R D4A0               CLB    10,D2
        006CR
  *YEP END OF BUFFER-ERROR RETURN
  001AR 4330               BE     IERR
        0046R
  *GET BYTE OF ARGUMENT
  001ER D3AD     T2        LB     10,0(13)
        0000
  *LOOK FOR CR
  0022R D4A0               CLB    10,D2
        006CR
  *DONE - RETYRN
  0026R 4330               BE     IDONE
        004ER
  *DO SOME REAL CHECKING
  002AR D4AC               CLB    10,0(12)
        0000
  *NOT YET-BACK UP A BIT
  002ER 4230               BNE    RESTOR
        003AR
  *SO FAR SO GOOD
```

```
0032R 0ADE              AHR     13,02
0034R 0ACE              AHR     12,02
*KEEP LOOKING
0036R 4300              B       IUPDAT
      0012R
*RESTORE TABLE POINTER
003AR 08CB    RESTOR    LHR     12,11
*KEEP TRACK OF INDEX
003CR 0ACE              AHR     12,02
*GET FIRST BYTE OF DATA AGAIN
003ER 48DF              LH      13,0(15)
      0000
*TRY AGAIN
0042R 4300              B       T1
      0010R
*RESTORE REGISTERS
0046R D1A0    IERR      LM      10,RRBF
      0060R
*ERROR RETUURN
004AR 430F              B       6(15)
      0006
*INC TO NEXT CHARACTER
004ER 0ABE    IDONE     AHR     11,02
*GET ADDRESS OF INDEX
0050R 48FF              LH      15,4(15)
      0004
*RETURN INDEX
0054R 40BF              STH     11,0(15)
      0000
0058R D1A0              LM      10,RRBF
      0060R
005CR 430F              B       10(15)
      000A
*REGISTER BUFFER
0060R           RRBF    DS      12
000E            02      EQU     14
006CR 0D00      D2      DC      X'0D00'
006ER                   END
* CINDX   0000R
  D2       006CR
  IDONE    004ER
  IERR     0046R
  IUPDAT   0012R
  02       000E
  RESTOR   003AR
  RRBF     0060R
  T1       0010R
  T2       001ER
 EOJ
```

M:

```
  008AR                    END
  CK2      001CR
  D        0088R
* FIND     0000R
  FNDD     005AR
  LKING    0010R
  NFNDD    0076R
  01       000C
  RBUFF    007ER
  RRDR     0074R
  SKIPP    0034R
  SPAC     0086R
  *          CALLING SEQUENCE AS FOLLOWS:
  *               BAL     FIND,15
  *               DC      A(ARG)
  *               DC      A(TABLE)
  *          ARG CONTAINS VARIABLE BYTE ARRGUMENT THAT IS TO BE
  *          FOUND AMONG THE COMMANDS IN TABLE. TABLE CONTAINS
  *          VARIABLE LENGTH CHARACTERS DELINEATED BY SPACES
  *          AND ALIGNED ON EVEN VOUNDARIES
  *          THE NEXT SEQUENTIAL HALFWORD CONTAINS THE ADDRESS
  *          OF THE LOCATION THAT 'FIND' SHOULD RETURN TO
  *          IF IT FINDS THE ARG IN THE TABLE
  *          END OF TABLE SIGNIFIES BY X'0D'
  0000R                    ENTRY FIND
  *SAVE REGISTERS
  0000R D0C0     FIND      STM   12,RBUFF
        007ER
  *ARGUMENT
  0004R 48EF               LH    14,0(15)
        0000
  *TABLE
  0008R 48DF               LH    13,2(15)
        0002
  *CONSTANT
  000CR C8C0               LHI   12,1
        0001
  000C           01        EQU   12
  *DONE YET-LOOK FOR ZERO
  0010R D3FD     LKING     LB    15,0(13)
        0000
  0014R D4F0               CLB   15,D
        0088R
  *GONE THROUGH TABLE-RETURN
  0018R 4330               BE    NFNDD
        0076R
  *SPACE IN TABLE?
  001CR D4F0     CK2       CLB   15,SPAC
        0086R
  *YES-FOUND IT
  0020R 4330               BE    FNDD
        005AR
  *COMPARE TO REAL ARGUMENT
  0024R D4FE               CLB   15,0(14)
```

```
      0000
*NO GOOD YET
0028R 4230              BNE    SKIPP
      0034R
*SO FAR SO GOOD
002CR 0AEC              AHR    14,01
002ER 0ADC              AHR    13,01
*KEEP CHECKING
0030R 4300              B      LKING
      0010R
*SKIP TP NEXT COMMAND
0034R 0ADC     SKIPP    AHR    13,01
0036R D3FD              LB     15,0(13)
      0000
*KEEPP GOING
003AR D4F0              CLB    15,SPAC
      0086R
*GET TO NEXT COMMAND
003ER 4230              BNE    SKIPP
      0034R
*ALIGN TO EVEN BOUNDARY
0042R CCD0              SRHL   13,1
      0001
0046R CDD0              SLHL   13,1
      0001
*GO TO NEXT COMMAND
004AR CAD0              AHI    13,4
      0004
004ER 48E0              LH     14,RBUFF+6
      0084R
*RESTORE ARG ADDRESS
0052R 48EE              LH     14,0(14)
      0000
0056R 4300              B      LKING
      0010R
*ALIGN TO EVEN BOUNDARY
005AR CCD0     FNDD     SRHL   13,1
      0001
005ER CDD0              SLHL   13,1
      0001
*GET RETURN ADDRESS
0062R CAD0              AHI    13,2
      0002
0066R 48FD              LH     15,0(13)
      0000
*LOCATION IN JUMP TABLE
006AR 40F0              STH    15,RRDR
      0074R
006ER D1C0              LM     12,RBUFF
      007ER
*RETURN THROUGH TABLE
0072R 4300              B      *-*
      0000
0074R          RRDR     EQU    *-2
```

```
      *NOT FOUND
      0076R D1C0     NFNDD    LM     12,RBUFF
            007ER
      007AR 430F               B      4(15)
            0004
      *REGISTER BUFFER
      007ER          RBUFF    DS     8
      0086R 2000     SPAC     DC     X'2000'
      0088R 0D00     D        DC     X'0D00'
      008AR                   END
      CK2       001CR
      D         0088R
    * FIND      0000R
      FNDD      005AR
      LKING     0010R
      NFNDD     0076R
      01        000C
      RBUFF     007ER
      RRDR      0074R
      SKIPP     0034R
      SPAC      0086R
     EOJ
M:
```

```
  01A2R                    END
  ADDNUM   0134R
* ADTAPE   0118R
* BAKSPC   00F6R
  BKSP     019FR
  BST      000D
  ENDPLY   00B8R
* FILEMK   00CAR
  FINIS    001AR
  FLMRK    019ER
  FP       005AR
* INITAP   0000R
* NOROOM   00A2R
* NUMBER   019AR
  R10      000A
  R11      000B
  R12      000C
  R13      000D
  R14      000E
  R15      000F
  READ     01A1R
  REWIND   019CR
  RTN1     000F
  SAVE     017AR
  SAVER    0160R
  SEEK     011ER
* SKIP     010ER
  STATUS   000B
  SVST     000A
  TAPE     000A
  TAPERR   0176R
* TDUMP    0034R
* TREAD    0072R
  TRY      0082R
* WAIT     0144R
  WFNM     0154R
  WRITE    01A0R
  WRK1     000C
  *REGISTER ASSIGNMENTS
  000A             TAPE     EQU    10
  000B             STATUS   EQU    11
  000C             WRK1     EQU    12
  000D             BST      EQU    13
  000F             RTN1     EQU    15
  *FIRST REGISTER T BE SAVED
  000A             SVST     EQU    10
  000A             R10      EQU    10
  000B             R11      EQU    11
  000C             R12      EQU    12
  000D             R13      EQU    13
  000E             R14      EQU    14
  000F             R15      EQU    15
  *
  0000R                     ENTRY INITAP,FILEMK,SKIP,TDUMP,TREAD
```

```
0000R                          ENTRY NUMBER,WAIT
0000R                          ENTRY BAKSPC
0000R                          EXTRN NOROOM,ADTAPE
*REWINDS THE TAPE
0000R 40F0     INITAP  STH     R15,SAVE+10
      0184R
0004R 41F0             BAL     R15,SAVER
      0160R
0008R 48A0             LH      TAPE,ADTAPE
      0000F
000CR 41F0             BAL     RTN1,WFNM
      0154R
0010R DEA0             OC      TAPE,REWIND
      019CR
0014R 0BCC             SHR     WRK1,WRK1
0016R 40C0             STH     WRK1,NUMBER
      019AR
001AR 48A0     FINIS   LH      R10,SAVE
      017AR
001ER 48B0             LH      R11,SAVE+2
      017CR
0022R 48C0             LH      R12,SAVE+4
      017ER
0026R 48D0             LH      R13,SAVE+6
      0180R
002AR 48E0             LH      R14,SAVE+8
      0182R
002ER 48F0             LH      R15,SAVE+10
      0184R
0032R 030F             BR      RTN1
*
*
*
0034R 40F0     TDUMP   STH     R15,SAVE+10
      0184R
0038R 41F0             BAL     R15,SAVER
      0160R
003CR 48A0             LH      TAPE,ADTAPE
      000AR
0040R 41F0             BAL     RTN1,WFNM
      0154R
0044R 9DAB             SSR     TAPE,STATUS
0046R C4B0             NHI     STATUS,X'0020'
      0020
*IF EOT WRITE A FILE MARK
004AR 4330             BZ      FP
      005AR
004ER DEA0             OC      TAPE,FLMRK
      019ER
0052R 41F0             BAL     RTN1,WFNM
      0154R
0056R DEA0             OC      TAPE,BKSP
      019FR
005AR 41F0     FP      BAL     RTN1,WFNM
```

```
      0154R
005ER DEA0              OC      TAPE,WRITE
      01A0R
0062R 96AD              WBR     TAPE,BST
0064R 9DAB              SSR     TAPE,STATUS
*IF EOT
0066R C4B0              NHI     STATUS,X'0020'
      0020
006AR 4230              BNZ     NOROOM
      0000F
006ER 4300              B       FINIS
      001AR
*
*
*
*READ A BLOCK
0072R 40F0      TREAD   STH     R15,SAVE+10
      0184R
0076R 41F0              BAL     R15,SAVER
      0160R
007AR C8C0              LHI     WRK1,10
      000A
007ER 48A0              LH      TAPE,ADTAPE
      003ER
0082R 41F0      TRY     BAL     RTN1,WFNM
      0154R
0086R DEA0              OC      TAPE,READ
      01A1R
008AR 97AD              RBR     TAPE,BST
*IF EX IS OFF
008CR 4340              BFC     4,FINIS
      001AR
0090R 9DAB              SSR     TAPE,STATUS
0092R 08FB              LHR     RTN1,STATUS
*IFF EOF
0094R C4B0              NHI     STATUS,X'40'
      0040
0098R 4230              BNZ     ENDPLY
      00B8R
009CR C4F0              NHI     RTN1,X'0020'
      0020
00A0R 4230              BNZ     NOROOM
      006CR
00A4R CAC0              AHI     WRK1,-1
      FFFF
00A8R 4330              BZ      TAPERR
      0176R
00ACR 41F0              BAL     RTN1,WFNM
      0154R
*TRY AGAIN
00B0R DEA0              OC      TAPE,BKSP
      019FR
00B4R 4300              B       TRY
      0082R
```

```
*
00B8R 41F0     ENDPLY  BAL    RTN1,WFNM
      0154R
*BACKSPACE OVER FILE
00BCR DEA0             OC     TAPE,BKSP
      019FR
*MARK SO NEXT WRITE WILL CONTINUE FILE
00C0R D1A0             LM     SVST,SAVE
      017AR
00C4R 48FF             LH     RTN1,2(RTN1)
      0002
00C8R 030F             BR     RTN1
*
*
00CAR 40F0     FILEMK  STH    R15,SAVE+10
      0184R
00CER 41F0             BAL    R15,SAVER
      0160R
00D2R 48A0             LH     TAPE,ADTAPE
      0080R
00D6R 41F0             BAL    RTN1,WFNM
      0154R
00DAR DEA0             OC     TAPE,FLMRK
      019ER
00DER 48C0             LH     WRK1,NUMBER
      019AR
00E2R CAC0             AHI    WRK1,1
      0001
00E6R 40C0             STH    WRK1,NUMBER
      019AR
00EAR 41F0             BAL    RTN1,WFNM
      0154R
*BACKSPACE OVER FILE MARK JUST WRITTEN
00EER DEA0             OC     TAPE,BKSP
      019FR
00F2R 4300             B      FINIS
      001AR
*
*
00F6R 40F0     BAKSPC  STH    R15,SAVE+10
      0184R
00FAR 41F0             BAL    R15,SAVER
      0160R
00FER 48A0             LH     TAPE,ADTAPE
      00D4R
0102R 41F0             BAL    RTN1,WFNM
      0154R
0106R DEA0             OC     TAPE,BKSP
      019FR
010AR 4300             B      FINIS
      001AR
*
*
010ER 40F0     SKIP    STH    R15,SAVE+10
```

```
        0184R
0112R 41F0              BAL     R15,SAVER
        0160R
0116R 48A0              LH      TAPE,ADTAPE
        0100R
011AR 41F0              BAL     RTN1,WFNM
        0154R
011ER DEA0      SEEK    OC      TAPE,READ
        01A1R
0122R 41F0              BAL     RTN1,WFNM
        0154R
0126R 9DAB              SSR     TAPE,STATUS
0128R C4B0              NHI     STATUS,X'40'
        0040
012CR 4230              BTC     3,ADDNUM
        0134R
0130R 4300              B       SEEK
        011ER
0134R 48C0      ADDNUM  LH      WRK1,NUMBER
        019AR
0138R CAC0              AHI     WRK1,1
        0001
013CR 40C0              STH     WRK1,NUMBER
        019AR
0140R 4300              B       FINIS
        001AR
*
*
*
0144R 40F0      WAIT    STH     R15,SAVE+10
        0184R
0148R 41F0              BAL     R15,SAVER
        0160R
014CR 41F0              BAL     RTN1,WFNM
        0154R
0150R 4300              B       FINIS
        001AR
*
*
0154R 9DAB      WFNM    SSR     TAPE,STATUS
0156R C4B0              NHI     STATUS,X'10'
        0010
015AR 4330              BFC     3,WFNM
        0154R
015ER 030F              BR      RTN1
*
*
0160R 40A0      SAVER   STH     R10,SAVE
        017AR
0164R 40B0              STH     R11,SAVE+2
        017CR
0168R 40C0              STH     R12,SAVE+4
        017ER
016CR 40D0              STH     R13,SAVE+6
```

```
          0180R
  0170R 40E0                STH     R14,SAVE+8
          0182R
  0174R 030F                BR      15
  *
  0176R E130      TAPERR    SVC     3,0
          0000
  *
  *
  017AR           SAVE      DS      32
  019AR 0000      NUMBER    DC      0
  019CR B8B8      REWIND    DC      X'B8B8'
  019ER B291      FLMRK     DC      X'B291'
  019FR           BKSP      EQU     *-1
  01A0R A2A1      WRITE     DC      X'A2A1'
  01A1R           READ      EQU     *-1
  01A2R                     END
  ADDNUM   0134R
* ADTAPE   0118R
* BAKSPC   00F6R
  BKSP     019FR
  BST      000D
  ENDPLY   00B8R
* FILEMK   00CAR
  FINIS    001AR
  FLMRK    019ER
  FP       005AR
* INITAP   0000R
* NOROOM   00A2R
* NUMBER   019AR
  R10      000A
  R11      000B
  R12      000C
  R13      000D
  R14      000E
  R15      000F
  READ     01A1R
  REWIND   019CR
  RTN1     000F
  SAVE     017AR
  SAVER    0160R
  SEEK     011ER
* SKIP     010ER
  STATUS   000B
  SVST     000A
  TAPE     000A
  TAPERR   0176R
* TDUMP    0034R
* TREAD    0072R
  TRY      0082R
* WAIT     0144R
  WFNM     0154R
  WRITE    01A0R
  WRK1     000C
```

```
  038AR                    END
  AD1       024ER
  AD2       01FER
* ADTAPE    0262R
  AFLAG     025CR
* ALPHA     0208R
  AMSG      02E7R
  ARAS      02FBR
  ARGT      0346R
  ATPP      02A1R
  BLACK     0054R
  CADR      0036R
  CAMBF     026AR
  CAMEND    0289R
* CAMINZ    015ER
  CBEND     02BDR
  CBUFF     02AAR
* CINDX     023CR
  CLEAR     012ER
  CONSOL    02A2R
  DEVICE    0368R
  DLEV      005ER
  DRAW      0042R
  EMSG      02E0R
  ENUM      020CR
  EOFMSG    02D8R
  EQEST     02C9R
  EQUALS    01E8R
  EQUAT     0096R
  ERAS      02FAR
  ERASE     02F2R
  ERDY      02D7R
  ERET      02F1R
  ESTATS    02F4R
  ETP       0296R
* FILEMK    00B2R
* FIND      024AR
  GOOD      0032R
  IDX       020AR
* INITAP    00E8R
  LEVEL     0090R
  LOC       024CR
  LTAB      0376R
  MADX      0144R
* MASK      014ER
  NEXT      0222R
  NEXT1     0216R
  NEXT2     022AR
  NMASK     028CR
* NOROOM    00E2R
  NTCR      0240R
  OFLAG     025ER
  ONE       0002
  OTAB      035AR
```

```
  OTAPE   0108R
  OUT     0102R
  QEST    02C6R
  QUEST   02BER
  QUIT    012AR
  RCONSL  001ER
  RDEVC   0258R
  RDY     02D2R
  READY   02CAR
  RESET   0010R
  RET     02F0R
  RETDR   0256R
  RETRN   02E8R
  REWND   009CR
  RFLAG   0260R
  RSTAT   02A4R
  SEVTN   00D4R
* SHOW    01E0R
  SIXTN   00C6R
* SKIP    00FCR
  SPACE   0266R
  STATUS  02EAR
  TABLE   02FCR
* TDUMP   01D4R
* TELL    0174R
  TEOFT   00F2R
  THRESH  0264R
  TPEEND  028ER
* TREAD   01A2R
  TSCREN  0166R
  VARDS   0120R
  VCONT   00FAR
  VEOF    00AAR
  VGCAM   01ACR
  VGO     0154R
  VGO1    017ER
* VIDCON  01AER
  VINP    0068R
  VMASK   0132R
  VOARDS  01D6R
  VOUTB   01BAR
  VRCAM   0086R
  VRTAB   037CR
  VRTAP   006ER
  VSKIP   00B8R
  VVV     0254R
  WBL     0268R
  WDEVC   025AR
  WHITE   0048R
  WQUEST  003AR
  XDEM    028AR
  0000R                 EXTRN FIND,CINDX,ALPHA
  0000R                 EXTRN MASK
  0000R                 EXTRN SHOW,TELL,CAMINZ,VIDCON
```

```
0000R                    EXTRN INITAP,FILEMK,SKIP
0000R                    EXTRN TDUMP,TREAD
0000R                    ENTRY NOROOM,ADTAPE
*GET ADDRESS FOR 'V' :
0000R E190               SVC   9,1
      0001
0004R D3E0               LB    14,VVV
      0254R
0008R D2EF               STB   14,0(15)
      0000
000CR C820               LHI   2,1
      0001
0002          ONE        EQU   2
0010R 0B11    RESET      SHR   1,1
*DEFAULT TO NO ARDS OUTPUT
0012R 4010               STH   1,AFLAG
      025CR
*DEFAULT TO CAMERA INPUT
0016R 4010               STH   1,RFLAG
      0260R
*DEFAULT TO NO TAPE OUTPUT
001AR 4010               STH   1,OFLAG
      025ER
*READY MESSAGE
001ER E110    RCONSL     SVC   1,RETRN
      02E8R
0022R 4810               LH    1,STATUS
      02EAR
0026R 4230               BNZ   RCONSL
      001ER
002AR E110               SVC   1,READY
      02CAR
*GET A LINE
002ER E110               SVC   1,CONSOL
      02A2R
*GO INTERPRET COMMAND
0032R 41F0    GOOD       BAL   15,FIND
      0000F
0036R 02AAR   CADR       DC    CBUFF
0038R 02FCR              DC    TABLE
*NOT FOUND
003AR E110    WQUEST     SVC   1,QUEST
      02BER
*GO TRY AGAIN
003ER 4300               B     RCONSL
      001ER
*
*     DRAW COMMANDS HERE
*
*GET ARGUMENT FOR DRAW
0042R 41F0    DRAW       BAL   15,NEXT
      0222R
0046R 0346R              DC    ARGT
0048R C810    WHITE      LHI   1,-1
```

```
      FFFF
*CODE FOR PRINT WHITE
004CR 4010              STH    1,WBL
      0268R
*DONE-GET ANOTHER LINE
0050R 4300              B      RCONSL
      001ER
0054R 0B11     BLACK    SHR    1,1
0056R 4010              STH    1,WBL
      0268R
*GET ANOTHER LINE
005AR 4300              B      RCONSL
      001ER
005ER 0812     DLEV     LHR    1,2
0060R 4010              STH    1,WBL
      0268R
0064R 4300              B      RCONSL
      001ER
*
*     READ COMMANDS
*
*GET ARG FOR INPUT COMMAND
0068R 41F0     VINP     BAL    15,NEXT
      0222R
006CR 037CR             DC     VRTAB
*FLAG FOR READING TAPE
006ER 0812     VRTAP    LHR    1,ONE
0070R 4010              STH    1,RFLAG
      0260R
*GET TAPE DEVIDE NUMBER
0074R 41F0              BAL    15,NEXT1
      0216R
0078R 0368R             DC     DEVICE
007AR 48E0              LH     14,ADTAPE
      0262R
007ER 40E0              STH    14,RDEVC
      0258R
0082R 4300              B      RCONSL
      001ER
*FLAG FOR INPUTTING CAMERA
0086R 0B11     VRCAM    SHR    1,1
0088R 4010              STH    1,RFLAG
      0260R
008CR 4300              B      RCONSL
      001ER
*
*     HERE TO SET LEVEL
*
*GO GET ARGUMENT FOR LEVEL
0090R 41F0     LEVEL    BAL    15,NEXT
      0222R
0094R 0376R             DC     LTAB
*GET NUMBER FOR LEVEL
0096R 41F0     EQUAT    BAL    15,EQUALS
```

```
      01E8R
009AR 0264R            DC     THRESH
*
*     TAPE COMMAND
*
*GET DEVICE NUMBER
009CR 41F0    REWND    BAL    15,NEXT
      0222R
00A0R 0368R            DC     DEVICE
00A2R 41F0             BAL    15,INITAP
      0000F
*REWIND AND RETURN TO CONSOL
00A6R 4300             B      RCONSL
      001ER
*GET DEVICE NUMBER
00AAR 41F0    VEOF     BAL    15,NEXT
      0222R
00AER 0368R            DC     DEVICE
00B0R 41F0             BAL    15,FILEMK
      0000F
*WRITE EOF AND RETURN
00B4R 4300             B      RCONSL
      001ER
*GET DEVICE NUMBER
00B8R 41F0    VSKIP    BAL    15,NEXT
      0222R
00BCR 0368R            DC     DEVICE
00BER 41F0             BAL    15,SKIP
      0000F
*SKIP TP EOF AND RETURN TO CONSOL
00C2R 4300             B      RCONSL
      001ER
*COME HERE FOR DEVICE 16
00C6R C810    SIXTN    LHI    1,X'16'
      0016
*SAVE FOR MTPAK
00CAR 4010             STH    1,ADTAPE
      0262R
00CER 48F0             LH     15,RETDR
      0256R
*RETURN TO CAALLER
00D2R 030F             BR     15
*COME HERE FOR DEVICE 17
00D4R C810    SEVTN    LHI    1,X'17'
      0017
00D8R 4010             STH    1,ADTAPE
      0262R
00DCR 48F0             LH     15,RETDR
      0256R
*RETURN TO CALLER
00E0R 030F             BR     15
*
*     COME HERE WHEN EOT ENCOUNTERED
*
```

```
00E2R E110     NOROOM  SVC    1,RETRN
      02E8R
*REWIND TAPE
00E6R 41F0             BAL    15,INITAP
      00A4R
*PRINT ERROR MESSAGE
00EAR E110             SVC    1,TPEEND
      028ER
*REST IMPORTANT FLAGS
00EER 4300             B      RESET
      0010R
*
*     COME HERE WHEN EOF ENCOUNTERED
*
00F2R E110     TEOFT   SVC    1,EOFMSG
      02D8R
00F6R 4300             B      RCONSL
      001ER
*
*     COME HERE TO CONTINUE OPERATIONS
*
*GO TO OTHER SIDE OF EOF
00FAR 41F0     VCONT   BAL    15,SKIP
      00C0R
*CONTINUE
00FER 4300             B      VGO1
      017ER
*
*     OUTPUT COMMANDS HERE
*
0102R 41F0     OUT     BAL    15,NEXT
      0222R
*GET ARG  FOR OUT COMMAND
0106R 035AR            DC     OTAB
*OUTPUT TAPE FLAG
0108R 0812     OTAPE   LHR    1,ONE
010AR 4010             STH    1,OFLAG
      025ER
*GET TAPE DEVICE
010ER 41F0             BAL    15,NEXT1
      0216R
0112R 0368R            DC     DEVICE
0114R 48E0             LH     14,ADTAPE
      0262R
*STORE WRITE TAPE DEVICE
0118R 40E0             STH    14,WDEVC
      025AR
011CR 4300             B      RCONSL
      001ER
*1 TO WRITE ARDS
0120R 0812     VARDS   LHR    1,ONE
0122R 4010             STH    1,AFLAG
      025CR
0126R 4300             B      RCONSL
```

```
      001ER
*
*     COME HERE TO EXIT TO MAGIC
*
*EXIT TO MAGIC
012AR E130    QUIT    SVC   3,0
      0000
*
*     COME HERE TO CLEAR COMMAND LIST
*
012ER 4300    CLEAR   B     RESET
      0010R
*
*     COME HERE TO IGNORE SOME LEVELS
*
*FIND ARG
0132R 41F0    VMASK   BAL   15,CINDX
      0000F
0136R 0266R           DC    SPACE
0138R 02AAR           DC    CBUFF
013AR 0144R           DC    MADX
*ERROR RETURN
013CR 4300            B     WQUEST
      003AR
*CONVERT TO HEX
0140R 41F0            BAL   15,ALPHA
      0000F
0144R 0000    MADX    DC    *-*
0146R 028CR           DC    NMASK
*ERROR RETURN
0148R 4300            B     WQUEST
      003AR
*GO TO MASK ROUTINE
014CR 41F0            BAL   15,MASK
      0000F
0150R 0004            DC    4
0152R 028CR           DC    NMASK
*
* COME HERE TO EXECUTE
*
0154R 48E0    VGO     LH    14,RFLAG
      0260R
0158R 4230            BNZ   TSCREN
      0166R
*INITIALIZE CAMERA
015CR 41F0            BAL   15,CAMINZ
      0000F
0160R 0006            DC    6
0162R 0264R           DC    THRESH
0164R 0268R           DC    WBL
0166R 48E0    TSCREN  LH    14,AFLAG
      025CR
016AR 4330            BZ    VGO1
      017ER
```

```
*ERASE THE SCREEN
016ER E110              SVC    1,ERASE
      02F2R
*SET UP FOR DRAW
0172R 41F0              BAL    15,TELL
      0000F
0176R 0008              DC     8
0178R 026AR             DC     CAMBF
017AR 028AR             DC     XDEM
017CR 028AR             DC     XDEM
017ER C8E0     VG01     LHI    14,X'A'
      000A
0182R 9DEF              SSR    14,15
0184R 4240              BTC    4,RCONSL
      001ER
0188R 48E0              LH     14,RFLAG
      0260R
*ITS NOT THE TAPE
018CR 4330              BZ     VGCAM
      01ACR
*GET DEVICE NUMBER TO MTPAK
0190R 48E0              LH     14,RDEVC
      0258R
0194R 40E0              STH    14,ADTAPE
      0262R
0198R C8D0              LHI    13,CAMBF
      026AR
019CR C8E0              LHI    14,CAMEND
      0289R
*GO READ A LINE FROM TAPE
01A0R 41F0              BAL    15,TREAD
      0000F
*GO TO EOF MESSAGE
01A4R 4200              NOP    TEOFT
      00F2R
*GO OUTPUT THE LINE
01A8R 4300              B      VOUTB
      01BAR
01ACR 41F0     VGCAM    BAL    15,VIDCON
      0000F
01B0R 0004              DC     4
01B2R 026AR             DC     CAMBF
01B4R 08EE              LHR    14,14
01B6R 4230              BNZ    RCONSL
      001ER
*DETERMINE OUTPUTTER
01BAR 48E0     VOUTB    LH     14,OFLAG
      025ER
*WELL ITS NOT THE TAPE DRIVE
01BER 4330              BZ     VOARDS
      01D6R
*DEVICE ADDRESS TO MTPAK
01C2R 48E0              LH     14,WDEVC
      025AR
```

```
01C6R 40E0              STH    14,ADTAPE
      0262R
01CAR C8D0              LHI    13,CAMBF
      026AR
01CER C8E0              LHI    14,CAMEND
      0289R
*DUMP TO TAPE
01D2R 41F0              BAL    15,TDUMP
      0000F
*WRITE TO ARDS ?
01D6R 48E0     VOARDS   LH     14,AFLAG
      025CR
*NO
01DAR 4330              BZ     VG01
      017ER
01DER 41F0              BAL    15,SHOW
      0000F
01E2R 0002              DC     2
01E4R 4300              B      VG01
      017ER
*
*
*GET ADDRESS OF NUMBER
01E8R 48FF     EQUALS   LH     15,0(15)
      0000
*ADDRESS
01ECR 40F0              STH    15,ENUM
      020CR
*GET LOCATION OF LAST COMMAND
01F0R 48F0              LH     15,LOC
      024CR
01F4R 40F0              STH    15,AD2
      01FER
01F8R 41F0              BAL    15,CINDX
      0134R
01FCR 0266R             DC     SPACE
01FER 0000     AD2      DC     *-*
0200R 020AR             DC     IDX
0202R 4300              B      WQUEST
      003AR
0206R 41F0              BAL    15,ALPHA
      0142R
020AR 0000     IDX      DC     *-*
*ADDRESS TO STORE RESULT
020CR 0000     ENUM     DC     *-*
*NO GOOD
020ER 4300              B      WQUEST
      003AR
*O.K. GET NEXT COMMAND
0212R 4300              B      RCONSL
      001ER
0216R 48E0     NEXT1    LH     14,LOC
      024CR
*GET PROPER LOCATION IN BUFFER
```

```
021AR 40E0              STH    14,NTCR
      0240R
*DONT RESET START OF BUFFER
021ER 4300              B      NEXT2
      022AR
0222R 48E0     NEXT     LH     14,CADR
      0036R
0226R 40E0              STH    14,NTCR
      0240R
022AR 48EF     NEXT2    LH     14,0(15)
      0000
022ER 40E0              STH    14,AD1
      024ER
*INCREMENT 15 TO RETURN ADDRESS
0232R CAF0              AHI    15,2
      0002
0236R 40F0              STH    15,RETDR
      0256R
*LOOK FOR SPACE
023AR 41F0              BAL    15,CINDX
      01FAR
023ER 0266R             DC     SPACE
*STARTING LOCATION IN CBUFF
0240R 0000     NTCR     DC     *-*
0242R 024CR             DC     LOC
*ERROR RETURN
0244R 4300              B      WQUEST
      003AR
*NOW GET ARGUMENT
0248R 41F0              BAL    15,FIND
      0034R
*ARGUMENT ADDRESS
024CR 0000     LOC      DC     *-*
*TABLE ADDRESS
024ER 0000     AD1      DC     *-*
*NOT FOUND
0250R 4300              B      WQUEST
      003AR
0254R 5620     VVV      DC     C'V'
*USED IN NEXT SUBROUTINE
0256R 0000     RETDR    DC     *-*
*INPUT TAPE DEVICE ADDRESS
0258R 0000     RDEVC    DC     *-*
*OUTPUT TAPE DEVICE ADDRESS
025AR 0000     WDEVC    DC     *-*
*ARDS OUTPUT FLAG
025CR 0000     AFLAG    DC     *-*
*OUTPUT TAPE FLAG
025ER 0000     OFLAG    DC     *-*
*READ TAPE OR CAMERA
0260R 0000     RFLAG    DC     *-*
*TAPE DEFAULT DEVICE ADDRESS
0262R 0016     ADTAPE   DC     X'16'
0264R 0000     THRESH   DC     *-*
```

```
0266R 200D     SPACE   DC     X'200D'
*FLAG FOR WHITE OR BLACK PRINT
0268R 0000     WBL     DC     *-*
026AR          CAMBF   DS     32
0289R          CAMEND  EQU    *-1
028AR 0100     XDEM    DC     256
028CR 0000     NMASK   DC     *-*
028ER 2005     TPEEND  DC     X'2005'
0290R 0000             DC     *-*
0292R 0296R            DC     ETP
0294R 02A1R            DC     ATPP
*OF TAPE'
0296R 454E     ETP     DC     C'END
      4420
      4F46
      2054
      4150
      4520
02A1R          ATPP    EQU    *-1
*READ UNIT 5
02A2R 4805     CONSOL  DC     X'4805'
02A4R 0000     RSTAT   DC     *-*
02A6R 02AAR            DC     CBUFF
02A8R 02BDR            DC     CBEND
02AAR          CBUFF   DS     20
02BDR          CBEND   EQU    *-1
*WRITE QUESTION MARK
02BER 2005     QUEST   DC     X'2005'
02C0R 0000             DC     *-*
02C2R 02C6R            DC     QEST
02C4R 02C9R            DC     EQEST
*? '
02C6R 2020     QEST    DC     C'
      3F20
02C9R          EQEST   EQU    *-1
*WRITE READY MESSAGE
02CAR 2005     READY   DC     X'2005'
02CCR 0000             DC     *-*
02CER 02D2R            DC     RDY
02D0R 02D7R            DC     ERDY
*READY'
02D2R 2052     RDY     DC     C'
      4541
      4459
02D7R          ERDY    EQU    *-1
*WRITE EOF MESSAGE
02D8R 2005     EOFMSG  DC     X'2005'
02DAR 0000             DC     *-*
02DCR 02E0R            DC     EMSG
02DER 02E7R            DC     AMSG
02E0R 2A2A     EMSG    DC     C'**EOF**'
      454F
      462A
      2A20
```

```
    02E7R           AMSG    EQU   *-1
02E8R 2005      RETRN   DC      X'2005'
02EAR 0000      STATUS  DC      *-*
02ECR 02F0R             DC      RET
02EER 02F1R             DC      ERET
02F0R 0D0A      RET     DC      X'0D0A'
02F1R           ERET    EQU     *-1
02F2R 2005      ERASE   DC      X'2005'
02F4R 0000      ESTATS  DC      *-*
02F6R 02FAR             DC      ERAS
02F8R 02FBR             DC      ARAS
*LINE FEED AND ERASE
02FAR 0A0C      ERAS    DC      X'0A0C'
02FBR           ARAS    EQU     *-1
*'
02FCR 4452      TABLE   DC      C'DR
      2020
0300R 0042R             DC      DRAW
0302R 5448              DC      C'THR'
      5220
0306R 0090R             DC      LEVEL
0308R 5245              DC      C'REW'
      5720
030CR 009CR             DC      REWND
*'
030ER 474F              DC      C'GO
      2020
0312R 0154R             DC      VGO
0314R 454F              DC      C'EOF'
      4620
0318R 00AAR             DC      VEOF
031AR 534B              DC      C'SKI'
      4920
031ER 00B8R             DC      VSKIP
0320R 4F55              DC      C'OUT'
      5420
0324R 0102R             DC      OUT
0326R 494E              DC      C'INP'
      5020
032AR 0068R             DC      VINP
032CR 5155              DC      C'QUI'
      4920
0330R 012AR             DC      QUIT
*'
0332R 434F              DC      C'CO
      2020
0336R 00FAR             DC      VCONT
*'
0338R 434C              DC      C'CL
      2020
033CR 012ER             DC      CLEAR
033ER 4D41              DC      C'MAS'
      5320
0342R 0132R             DC      VMASK
```

```
0344R 0D0D              DC      X'0D0D'
*'
0346R 5748      ARGT    DC      C'WH
      2020
034AR 0048R             DC      WHITE
*'
034CR 424C              DC      C'BL
      2020
0350R 0054R             DC      BLACK
*'
0352R 4C45              DC      C'LE
      2020
0356R 005ER             DC      DLEV
0358R 0D0D              DC      X'0D0D'
*'
035AR 5441      OTAB    DC      C'TA
      2020
035ER 0108R             DC      OTAPE
0360R 4152              DC      C'ARD'
      4420
0364R 0120R             DC      VARDS
0366R 0D0D              DC      X'0D0D'
*'
0368R 3136      DEVICE  DC      C'16
      2020
036CR 00C6R             DC      SIXTN
*'
036ER 3137              DC      C'17
      2020
0372R 00D4R             DC      SEVTN
0374R 0D0D              DC      X'0D0D'
0376R 3D20      LTAB    DC      C'='
0378R 0096R             DC      EQUAT
037AR 0D0D              DC      X'0D0D'
*'          COMMANDS FOR READ
037CR 5441     VRTAB    DC      C'TA
      2020
0380R 006ER             DC      VRTAP
*'
0382R 4341              DC      C'CA
      2020
0386R 0086R             DC      VRCAM
0388R 0D0D              DC      X'0D0D'
038AR                   END
AD1       024ER
AD2       01FER
* ADTAPE  0262R
AFLAG     025CR
* ALPHA   0208R
AMSG      02E7R
ARAS      02FBR
ARGT      0346R
ATPP      02A1R
BLACK     0054R
```

```
  CADR     0036R
  CAMBF    026AR
  CAMEND   0289R
* CAMINZ   015ER
  CBEND    02BDR
  CBUFF    02AAR
* CINDX    023CR
  CLEAR    012ER
  CONSOL   02A2R
  DEVICE   0368R
  DLEV     005ER
  DRAW     0042R
  EMSG     02E0R
  ENUM     020CR
  EOFMSG   02D8R
  EQEST    02C9R
  EQUALS   01E8R
  EQUAT    0096R
  ERAS     02FAR
  ERASE    02F2R
  ERDY     02D7R
  ERET     02F1R
  ESTATS   02F4R
  ETP      0296R
* FILEMK   00B2R
* FIND     024AR
  GOOD     0032R
  IDX      020AR
* INITAP   00E8R
  LEVEL    0090R
  LOC      024CR
  LTAB     0376R
  MADX     0144R
* MASK     014ER
  NEXT     0222R
  NEXT1    0216R
  NEXT2    022AR
  NMASK    028CR
* NOROOM   00E2R
  NTCR     0240R
  OFLAG    025ER
  ONE      0002
  OTAB     035AR
  OTAPE    0108R
  OUT      0102R
  QEST     02C6R
  QUEST    02BER
  QUIT     012AR
  RCONSL   001ER
  RDEVC    0258R
  RDY      02D2R
  READY    02CAR
  RESET    0010R
  RET      02F0R
```

```
  RETDR    0256R
  RETRN    02E8R
  REWND    009CR
  RFLAG    0260R
  RSTAT    02A4R
  SEVTN    00D4R
* SHOW     01E0R
  SIXTN    00C6R
* SKIP     00FCR
  SPACE    0266R
  STATUS   02EAR
  TABLE    02FCR
* TDUMP    01D4R
* TELL     0174R
  TEOFT    00F2R
  THRESH   0264R
  TPEEND   028ER
* TREAD    01A2R
  TSCREN   0166R
  VARDS    0120R
  VCONT    00FAR
  VEOF     00AAR
  VGCAM    01ACR
  VGO      0154R
  VGO1     017ER
* VIDCON   01AER
  VINP     0068R
  VMASK    0132R
  VOARDS   01D6R
  VOUTB    01BAR
  VRCAM    0086R
  VRTAB    037CR
  VRTAP    006ER
  VSKIP    00B8R
  VVV      0254R
  WBL      0268R
  WDEVC    025AR
  WHITE    0048R
  WQUEST   003AR
  XDEM     028AR
```

APPENDIX B.

MUSIC Program Listings

```
OSLOAD
OUT MUSIC.PROGRAM
BC 0A0A
BIAS 2A62
LO CONSTANTS
LI IHENTRY
LI MUSIC
LI NOTES
LI STAFF
LI INIGRD
LI TPOINT
LI WNWONE
LI GETLIN
LI DISK-MASK
LI IOA
LI WSQR
LI GZAP
LI HNDLER
LI SREAD
LI WINDOW
LI CIRCLE
ED RTL
LI IHEND
MA 3
XOUT
EN
DELETE FIND-NOTES.LOADMOD
BIAS 2A62
LOAD MUSIC.PROGRAM
CL 2B00
CREATEUSER FIND-NOTES
```

****

```
      INTEGER*2 NLOCX(10),LINE(11,6),INOTE(11)
      INTEGER*2 TAPE,GCB(79),SIZE
      INTEGER*2 IX,IY,K,J,JPOS,IPOS
      INTEGER*2 SLICE(5,256)
      INTEGER*2 X1,X2,Y1,Y2
      INTEGER*2 RIGHT,TOP,DCODE,BOTT,LEFT
      INTEGER*2 IERR,RADIUS
      COMMON BOTT,LEFT,RIGHT,TOP,DCODE,SLICE
      DATA SIZE/1/
      DATA INOTE(1),INOTE(2),INOTE(3)/1HD,1HE,1HF/
      DATA INOTE(4),INOTE(5),INOTE(6)/1HG,1HA,1HB/
      DATA INOTE(7),INOTE(8),INOTE(9)/1HC,1HD,1HE/
      DATA INOTE(10),INOTE(11)/1HF,1HG/
C     SET UP DISK
      DCODE=1
      CALL INIGRD(TAPE,GCB,SIZE)
      CALL GREAD(GCB)
      BOTT=GCB(4)
      LEFT=GCB(6)
      TOP=GCB(5)
      RIGHT=GCB(7)
C     GET STAFF LOCATIONS
      CALL STAFF(LINE,IERR)
      DO 699 IJ=2,10,2
      X1=(LINE(IJ,1)*4)-2048
      Y1=(LINE(IJ,2)*4)-2048
      X2=(LINE(IJ,4)*4)-2048
      Y2=(LINE(IJ,5)*4)-2048
      CALL TPPLIN(X1,Y1,X2,Y2)
699   CONTINUE
      IF(IERR) 100,120,100
100   CALL IOA(15HERROR IN STAFF↑  )
      GO TO 999
C     DO IT FOR EACH LNE
120   DO 200 I=1,11
      K=I+1
      J=I-1
      IF(I.EQ.1) GO TO 710
      IF(I.EQ.11) GO TO 720
      IY=MINO(LINE(K,2),LINE(K,5))
      IX=MAXO(LINE(J,2),LINE(J,5))
      IX=IX+MAXO(LINE(J,3),LINE(J,6))
      GO TO 750
710   IX=MAXO(LINE(I,2),LINE(I,5))
      IY=MINO(LINE(K,2),LINE(K,5))
      IX=(2*IX)-IY
      GO TO 750
720   IX=MAXO(LINE(J,3),LINE(J,6))
      IX=IX+MAXO(LINE(J,2),LINE(J,5))
      IY=MINO(LINE(I,2),LINE(I,5))
      IY=(2*IY)-IX
750   CALL NOTES(NLOCX,NNOTE,LINE(1,1),IX,IY)
195   IF(NNOTE) 160,200,170
```

```
160     CALL IOA(15HERROR IN NOTES↑   )
        GO TO 999
170     DO 400 J=1,NNOTE
C       NOW DRAW THE NOTE VALUE
        JPOS=(((IX+IY)/2)*4)-2048
        IPOS=(NLOCX(J)*4)-2048
        RADIUS=2*IABS(IX-IY)
        CALL CIRCLE(RADIUS,IPOS,JPOS)
        CALL TPOINT(IPOS,JPOS)
        CALL IOA(3H↑A↑,INOTE(I))
400     CONTINUE
200     CONTINUE
999     STOP
        END
```

****

```
      SUBROUTINE NOTES(NLOCX,NNOTE,X,Y1,Y2)
      INTEGER*2 NLOCX(10),SLICE(5,256)
      INTEGER*2 YMIN,YMAX,Y1,Y2
      INTEGER*2 XDIM,BOTT,LEFT,RIGHT,TOP
      INTEGER*2 DCODE,NNOTE,IY,II
      INTEGER*2 SIZE,BITS,DENSTY,AVDEN
      INTEGER*2 IJUMP,X,MIN,XEND
      INTEGER*2 D1
      REAL MINCNT
      COMMON BOTT,LEFT,RIGHT,TOP,DCODE,SLICE
      XDIM=RIGHT-LEFT
      PERCNT=0.60
      MINCNT=0.15
      CALL WTHRU(1)
      IJUMP=1
      NNOTE=0
      YMAX=MAX0(Y1,Y2)
      YMIN=MIN0(Y1,Y2)
      SIZE=YMAX-YMIN
      BITS=SIZE*SIZE
      WRK1=PERCNT*FLOAT(BITS)
      AVDEN=IFIX(WRK1)
      WRK1=MINCNT*FLOAT(BITS)
      MIN=IFIX(WRK1)
      XEND=LEFT+XDIM
      DO 100 I=X,XEND
      IX=I
      CALL WIN(DENSTY,SIZE,IX,YMIN,II,DCODE,1,1)
      IF(DENSTY.GT.MIN) GO TO 400
      IJUMP=1
      GO TO 100
400   IF(DENSTY.LT.AVDEN) GO TO 150
      GO TO (200,600),IJUMP
200   NNOTE=NNOTE+1
700   D1=DENSTY
      IJUMP=2
      NLOCX(NNOTE)=IX+(SIZE/2)
      GO TO 100
600   IF(D1.LT.DENSTY) GO TO 700
      GO TO 100
150   IJUMP=1
100   CONTINUE
999   CALL WTHRU(0)
      RETURN
      END
```

****

```
      SUBROUTINE STAFF(LINE,IERR)
      INTEGER*2 LINE(11,6),SLICE(5,256)
      INTEGER*2 XBIAS,YDIM,XDIM
      INTEGER*2 BOTT,LEFT,RIGHT,TOP
      INTEGER*2 DCODE,IPT,IERR,YSIZE
      INTEGER*2 IX,IY,I,J,N,NJ,L
      COMMON BOTT,LEFT,RIGHT,TOP,DCODE,SLICE
      DATA XDIM/5/
      DATA YSIZE/256/
      CALL WTHRU(1)
      IERR=0
      YDIM=TOP-BOTT
      IF(YDIM.LE.YSIZE) GO TO 139
      YDIM=YSIZE
139   DO 666 L=1,4,3
      DO 120 XBIAS=1,YDIM,XDIM
      IPT=0
      IX=XBIAS+LEFT
      IF(L .EQ. 1) GO TO 700
      IX=RIGHT-XBIAS-XDIM+1
700   CONTINUE
      CALL WIN(SLICE,1,IX,BOTT,III,DCODE,XDIM,YDIM)
      DO 299 NJ=1,XDIM
      IPT=0
      N=NJ
      IF(L .EQ. 1) GO TO 759
      N=XDIM-NJ+1
759   CONTINUE
      DO 100 I=1,YDIM
      IF(SLICE(N,I) .EQ. 0) GO TO 100
      IPT=IPT+2
      LINE(IPT,L)=IX+N
      LINE(IPT,L+1)=I+BOTT
      DO 150 J=I,YDIM
      IF(SLICE(N,J) .EQ. 0) GO TO 170
150   CONTINUE
      IERR=1
      GO TO 999
170   LINE(IPT,L+2)=J-I
      IF(IPT.EQ.10) GO TO 666
      I=J
100   CONTINUE
299   CONTINUE
120   CONTINUE
666   CONTINUE
C------INTERPOLATE BETWEEN LINES
      DO 500 I=3,9,2
      IJ=I+1
      IN=I-1
      LINE(I,1)=LINE(IN,1)
      LINE(I,4)=LINE(IN,4)
      LINE(I,3)=1
      LINE(I,6)=1
```

```
        LINE(I,2)=(LINE(IJ,2)+LINE(IN,2)+LINE(IN,3))
        LINE(I,2)=LINE(I,2)/2
        LINE(I,5)=(LINE(IJ,5)+LINE(IN,5)+LINE(IN,6))
        LINE(I,5)=LINE(I,5)/2
500     CONTINUE
C------NOW GET TOP AND BOTTOM SPACE
        LINE(1,1)=LINE(2,1)
        LINE(1,2)=(2*(LINE(2,2)+LINE(2,3)))
        LINE(1,2)=LINE(1,2)-LINE(3,2)
        LINE(1,3)=1
        LINE(1,4)=LINE(2,4)
        LINE(1,5)=(2*(LINE(2,5)+LINE(2,6)))
        LINE(1,5)=LINE(1,5)-LINE(3,5)
        LINE(1,6)=1
C------GET TOP LINE
        LINE(11,1)=LINE(10,1)
        LINE(11,2)=(2*LINE(10,2))-LINE(9,2)
        LINE(11,2)=LINE(11,2)+LINE(9,3)
        LINE(11,3)=1
        LINE(11,4)=LINE(10,4)
        LINE(11,5)=(2*LINE(10,5))-LINE(9,5)
        LINE(11,5)=LINE(11,5)+LINE(9,6)
        LINE(11,6)=1
999     CALL WTHRU(0)
        RETURN
        END
```

****

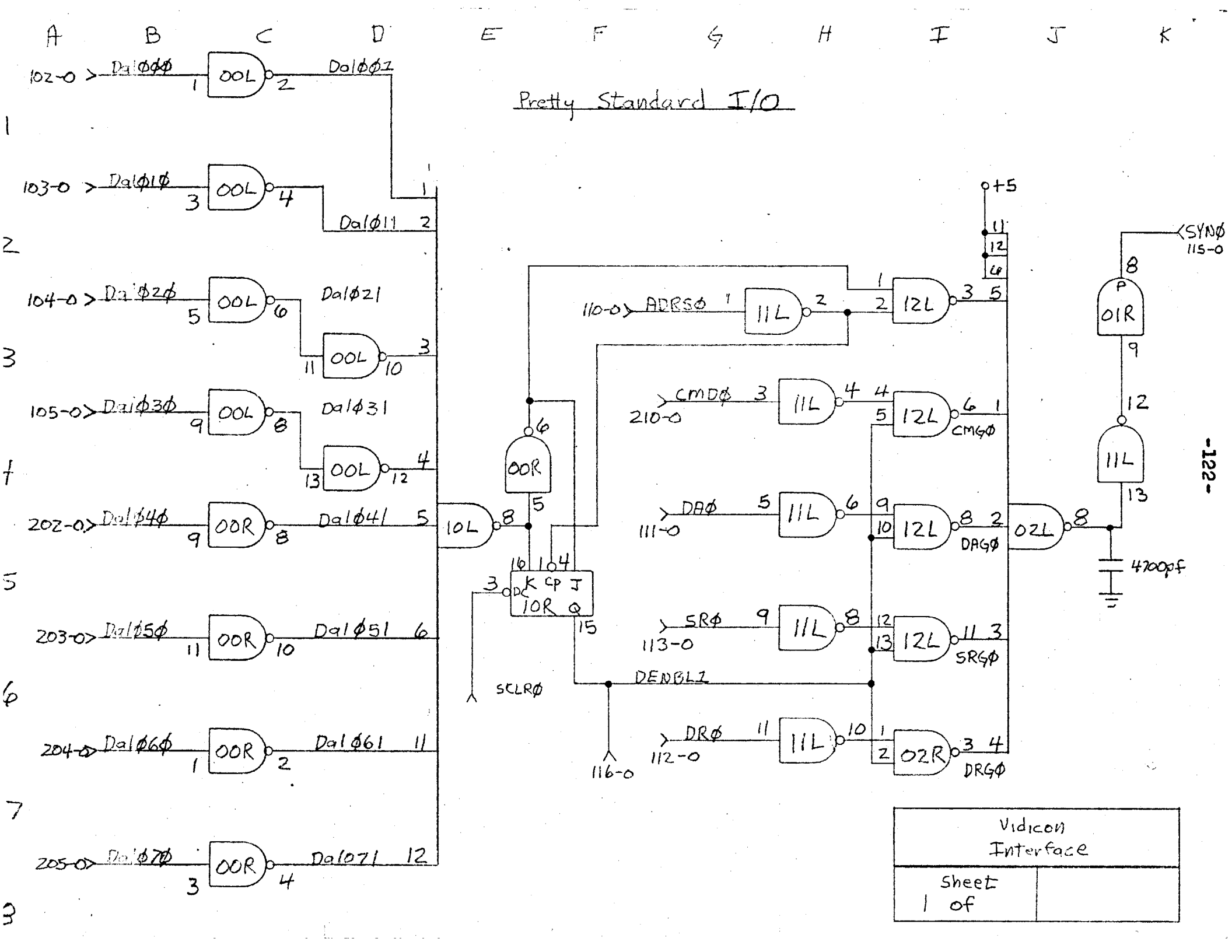
```
      INTEGER*2 NLOCX(10),LINE(11,6),INOTE(11)
      INTEGER*2 TAPE,GCB(79),SIZE
      INTEGER*2 IX,IY,K,J,JPOS,IPOS
      INTEGER*2 SLICE(5,256)
      INTEGER*2 X1,X2,Y1,Y2
      INTEGER*2 RIGHT,TOP,DCODE,BOTT,LEFT
      INTEGER*2 IERR,RADIUS
      COMMON BOTT,LEFT,RIGHT,TOP,DCODE,SLICE
      DATA SIZE/1/
      DATA INOTE(1),INOTE(2),INOTE(3)/1HD,1HE,1HF/
      DATA INOTE(4),INOTE(5),INOTE(6)/1HG,1HA,1HB/
      DATA INOTE(7),INOTE(8),INOTE(9)/1HC,1HD,1HE/
      DATA INOTE(10),INOTE(11)/1HF,1HG/
C     SET UP DISK
      DCODE=1
      CALL INIGRD(TAPE,GCB,SIZE)
      CALL GREAD(GCB)
      BOTT=GCB(4)
      LEFT=GCB(6)
      TOP=GCB(5)
      RIGHT=GCB(7)
C     GET STAFF LOCATIONS
      CALL STAFF(LINE,IERR)
      DO 699 IJ=2,10,2
      X1=(LINE(IJ,1)*4)-2048
      Y1=(LINE(IJ,2)*4)-2048
      X2=(LINE(IJ,4)*4)-2048
      Y2=(LINE(IJ,5)*4)-2048
      CALL TPPLIN(X1,Y1,X2,Y2)
699   CONTINUE
      IF(IERR) 100,120,100
100   CALL IOA(15HERROR IN STAFF!  )
      GO TO 999
C     DO IT FOR EACH LNE
120   DO 200 I=1,11
      K=I+1
      J=I-1
      IF(I.EQ.1) GO TO 710
      IF(I.EQ.11) GO TO 720
      IY=MINO(LINE(K,2),LINE(K,5))
      IX=MAXO(LINE(J,2),LINE(J,5))
      IX=IX+MAXO(LINE(J,3),LINE(J,6))
      GO TO 750
710   IX=MAXO(LINE(I,2),LINE(I,5))
      IY=MINO(LINE(K,2),LINE(K,5))
      IX=(2*IX)-IY
      GO TO 750
720   IX=MAXO(LINE(J,3),LINE(J,6))
      IX=IX+MAXO(LINE(J,2),LINE(J,5))
      IY=MINO(LINE(I,2),LINE(I,5))
      IY=(2*IY)-IX
750   CALL NOTES(NLOCX,NNOTE,LINE(1,1),IX,IY)
195   IF(NNOTE) 160,200,170
```

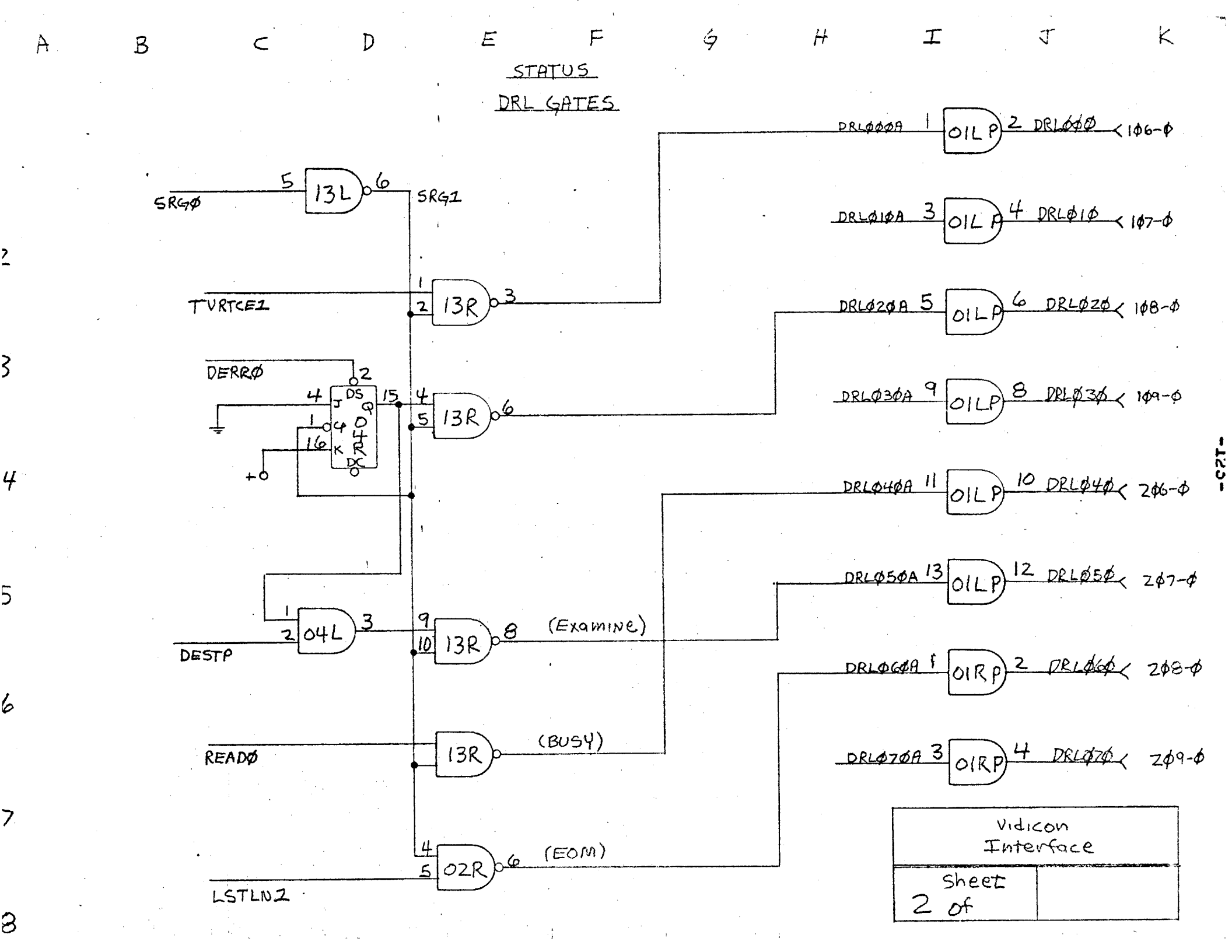
```
160     CALL IOA(15HERROR IN NOTES↑   )
        GO TO 999
170     DO 400 J=1,NNOTE
C       NOW DRAW THE NOTE VALUE
        JPOS=(((IX+IY)/2)*4)-2048
        IPOS=(NLOCX(J)*4)-2048
        RADIUS=IABS(IX-IY)/2
        CALL CIRCLE(RADIUS,IPOS,JPOS)
        CALL TPOINT(IPOS,JPOS)
        CALL IOA(3H↑A↑,INOTE(I))
400     CONTINUE
200     CONTINUE
999     STOP
        END
```

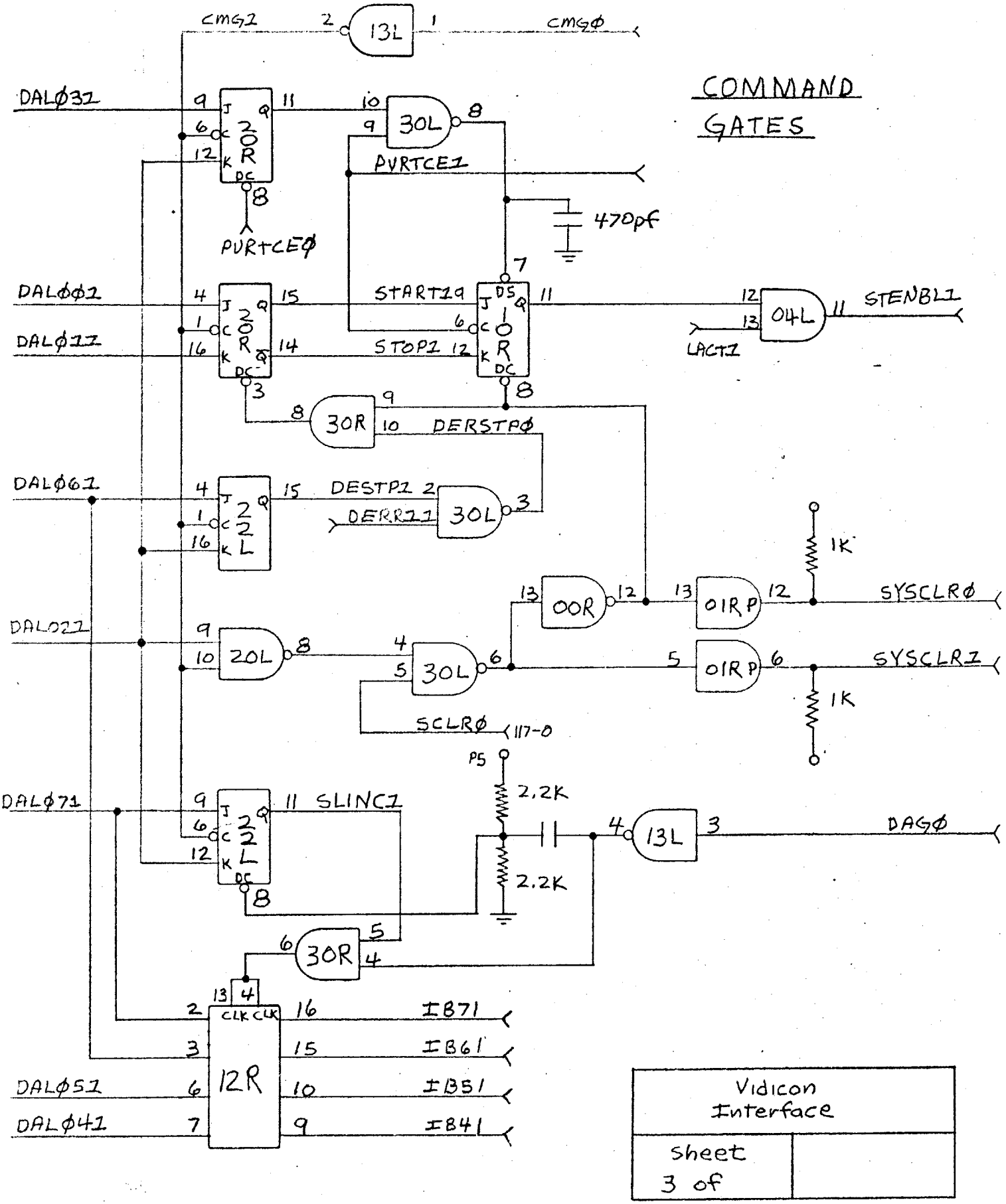
****

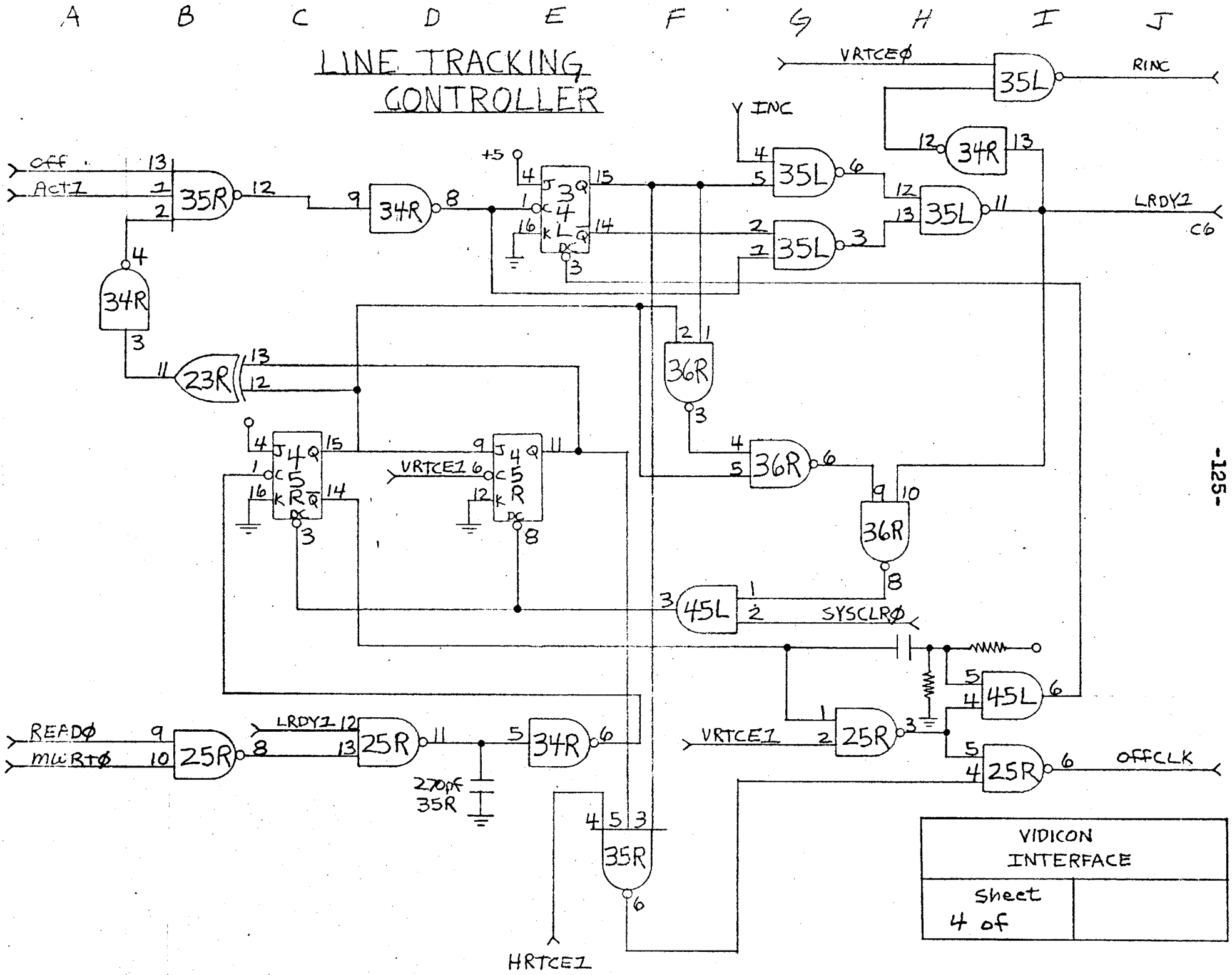
APPENDIX C.

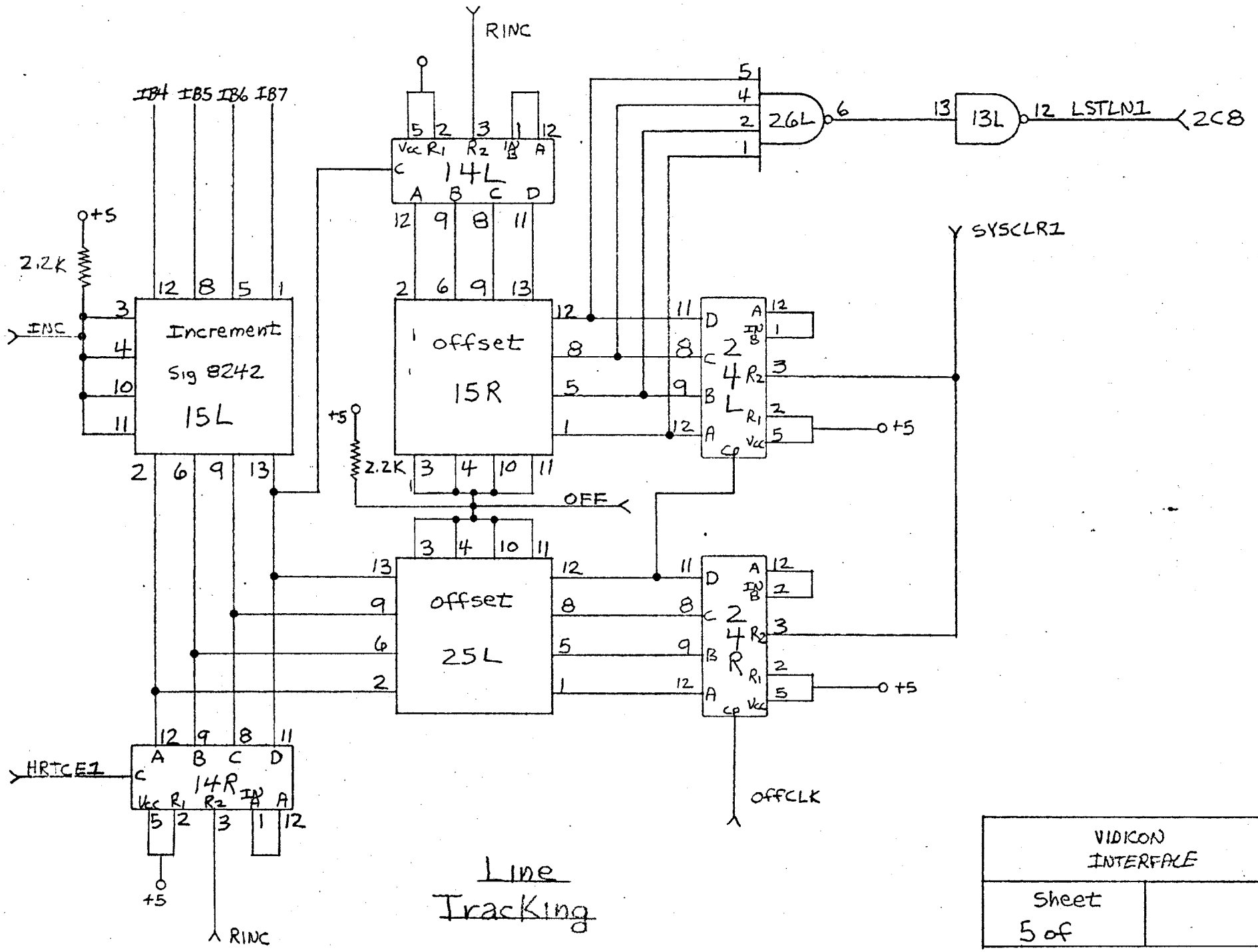
Vidicon Interface Drawings


Pretty Standard I/O
102-0
Dal ØØØ
00L
Dal ØØ1
103-0
Dal Ø1Ø
00L
Dal Ø11
104-0
Dal Ø2Ø
00L
Dal Ø21
00L
105-0
Dal Ø3Ø
00L
Dal Ø31
00L
202-0
Dal Ø4Ø
00R
Dal Ø41
203-0
Dal Ø5Ø
00R
Dal Ø51
204-0
Dal Ø6Ø
00R
Dal Ø61
205-0
Dal Ø7Ø
00R
Dal 071
10L
00R
10R
SCLRØ
DENBL2
116-0
110-0
ADRSØ
11L
12L
210-0
CMDØ
CMGØ
111-0
DAØ
DAGØ
113-0
SRØ
SRGØ
112-0
DRØ
02R
DRGØ
02L
+5
01R
SYNØ
115-0
11L
4700pf
-122-
Vidicon Interface
Sheet 1 of

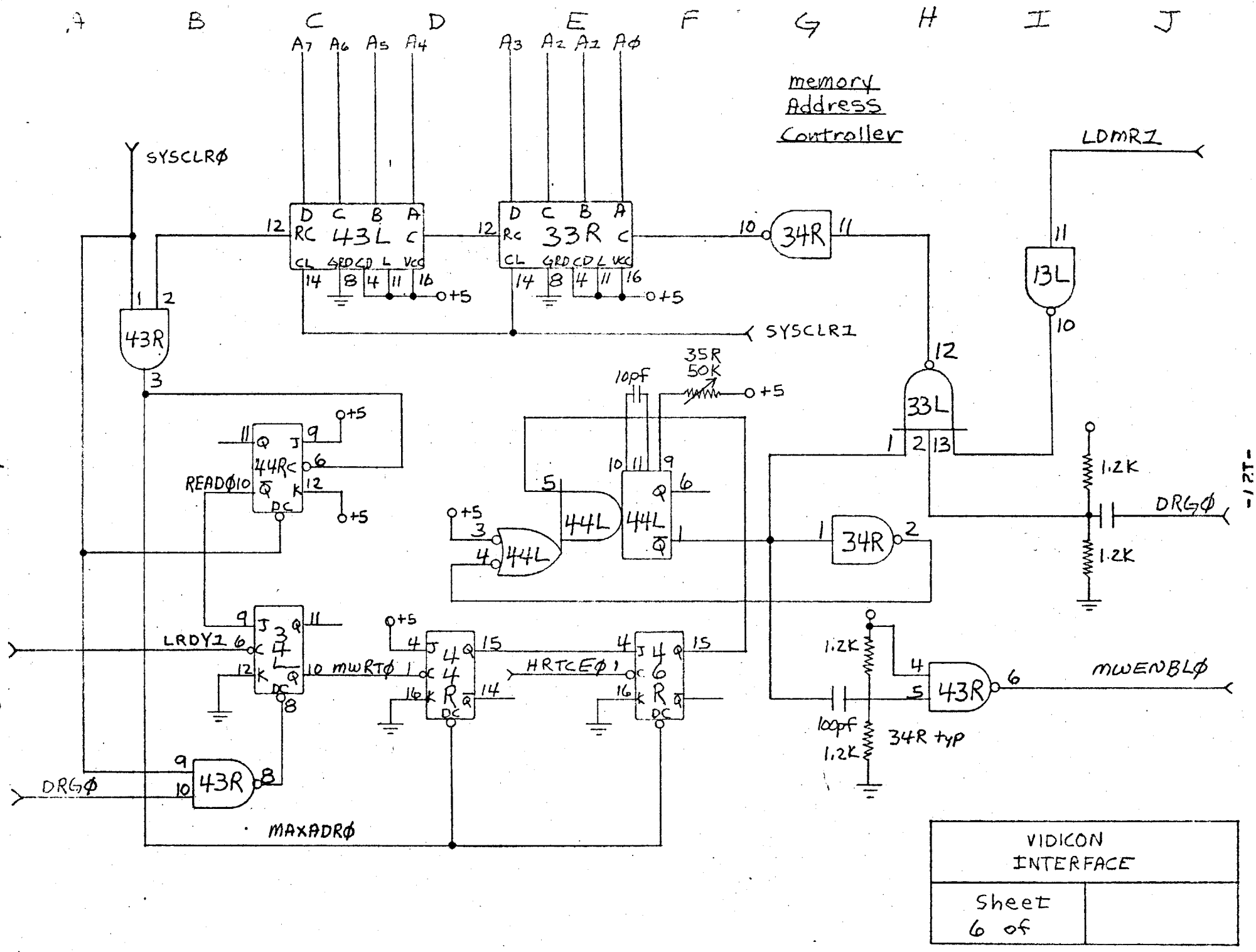

STATUS
DRL GATES
SRGØ
13L
SRG1
TVRTCE1
13R
DERRØ
DS
J
Q
K
DC
04L
DESTP
(Examine)
READØ
(BUSY)
02R
(EOM)
LSTLN2
DRLØØØA
01LP
DRLØØØ
1Ø6-Ø
DRLØ1ØA
DRLØ1Ø
1Ø7-Ø
DRLØ2ØA
DRLØ2Ø
1Ø8-Ø
DRLØ3ØA
DRLØ3Ø
1Ø9-Ø
DRLØ4ØA
DRLØ4Ø
2Ø6-Ø
DRLØ5ØA
DRLØ5Ø
2Ø7-Ø
DRLØ6ØA
01RP
DRLØ6Ø
2Ø8-Ø
DRLØ7ØA
DRLØ7Ø
2Ø9-Ø
Vidicon Interface
Sheet 2 of

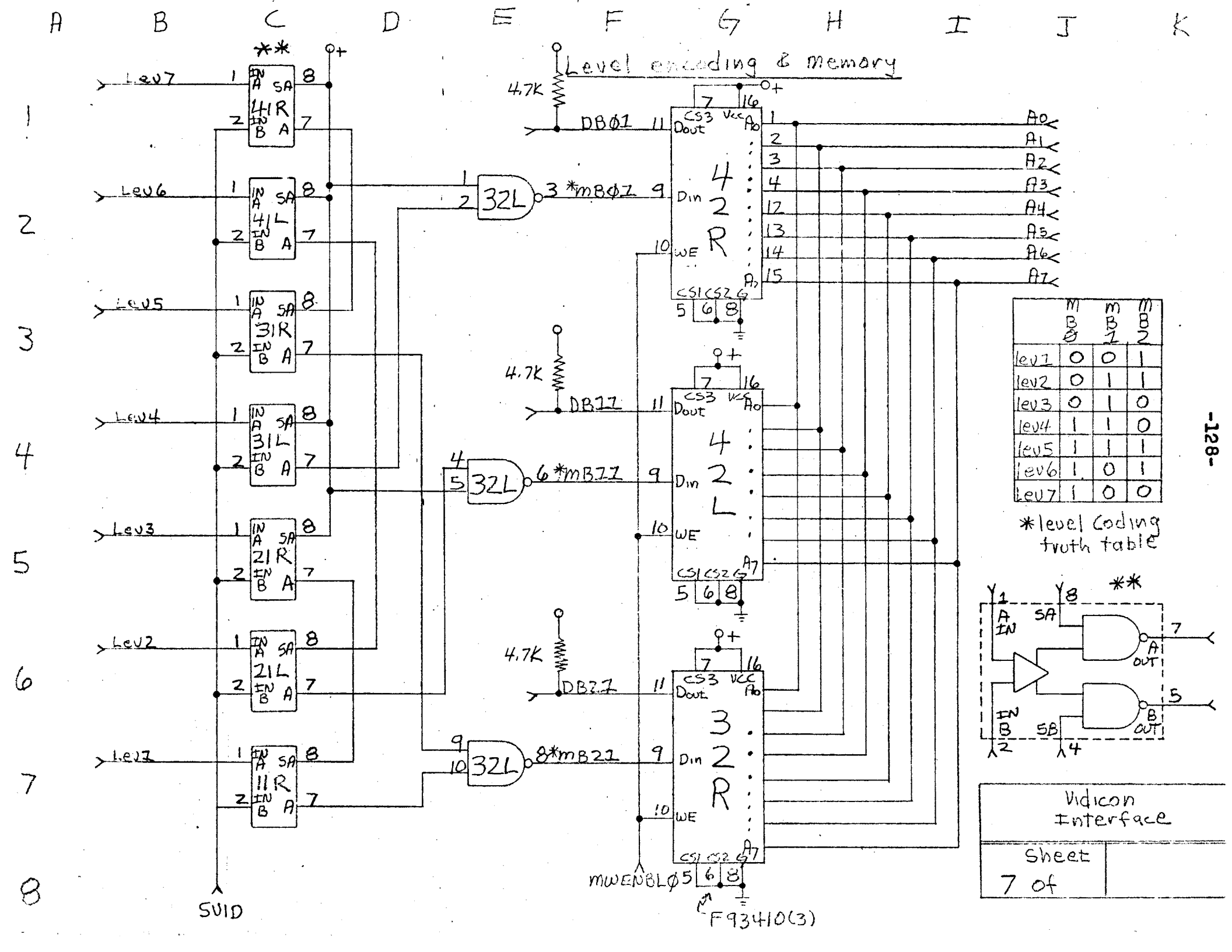

COMMAND GATES
CMG2
CMGØ
13L
DALØ32
30L
PVRTCE1
PVRTCEØ
470pf
DALØØ2
DALØ11
START19
STOP1
04L
STENBL1
LACT1
30R
DERSTPØ
DALØ61
DESTP1
DERR11
22L
1K
00R
01RP
SYSCLRØ
SYSCLR1
DAL022
20L
SCLRØ
117-0
P5
2.2K
DALØ71
SLINC1
DAGØ
DALØ51
DALØ41
12R
IB71
IB61
IB51
IB41
Vidicon Interface
sheet 3 of


LINE TRACKING CONTROLLER
VRTCEØ
35L
RINC
Y INC
34R
Off
Act1
35R
34R
+5
LRDY1
C6
35L
35L
34R
23R
36R
36R
36R
VRTCE1
45L
SYSCLRØ
45L
READØ
MWRTØ
25R
LRDY1
25R
34R
VRTCE1
25R
25R
OFFCLK
220pf
35R
35R
HRTCE1
VIDICON INTERFACE
Sheet 4 of
-125-



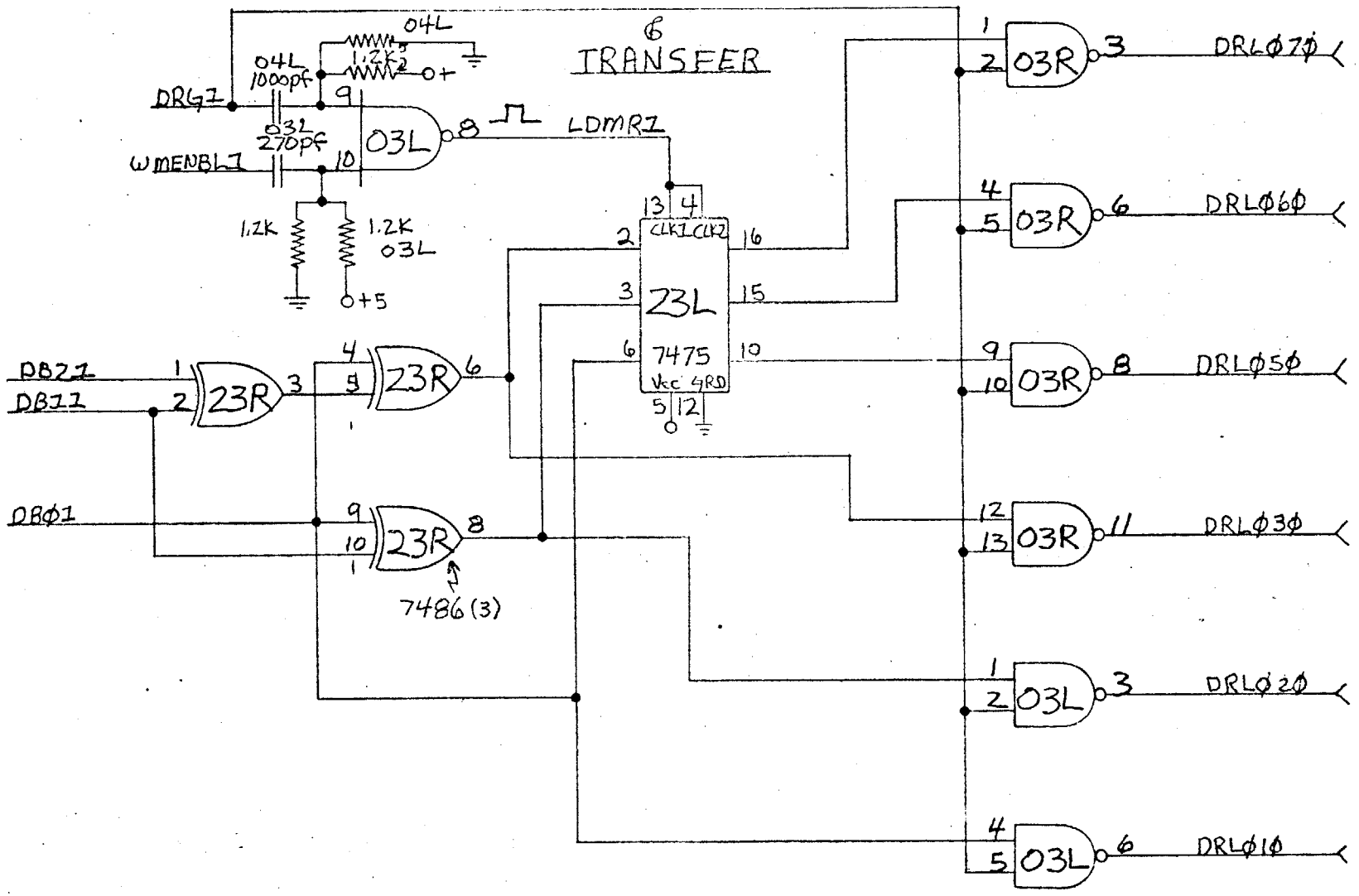
VIDICON INTERFACE	
Sheet 5 of	


memory
Address
Controller
SYSCLRØ
A7 A6 A5 A4
A3 A2 A1 AØ
43L
33R
34R
43R
13L
33L
44R
44L
35R
50K
10pf
+5
SYSCLR1
LDMR1
READØ10
LRDY1
mWRTØ
HRTCEØ1
DRGØ
MAXADRØ
mWENBLØ
1.2K
100pf
34R typ
VIDICON
INTERFACE
Sheet
6 of
-12T-



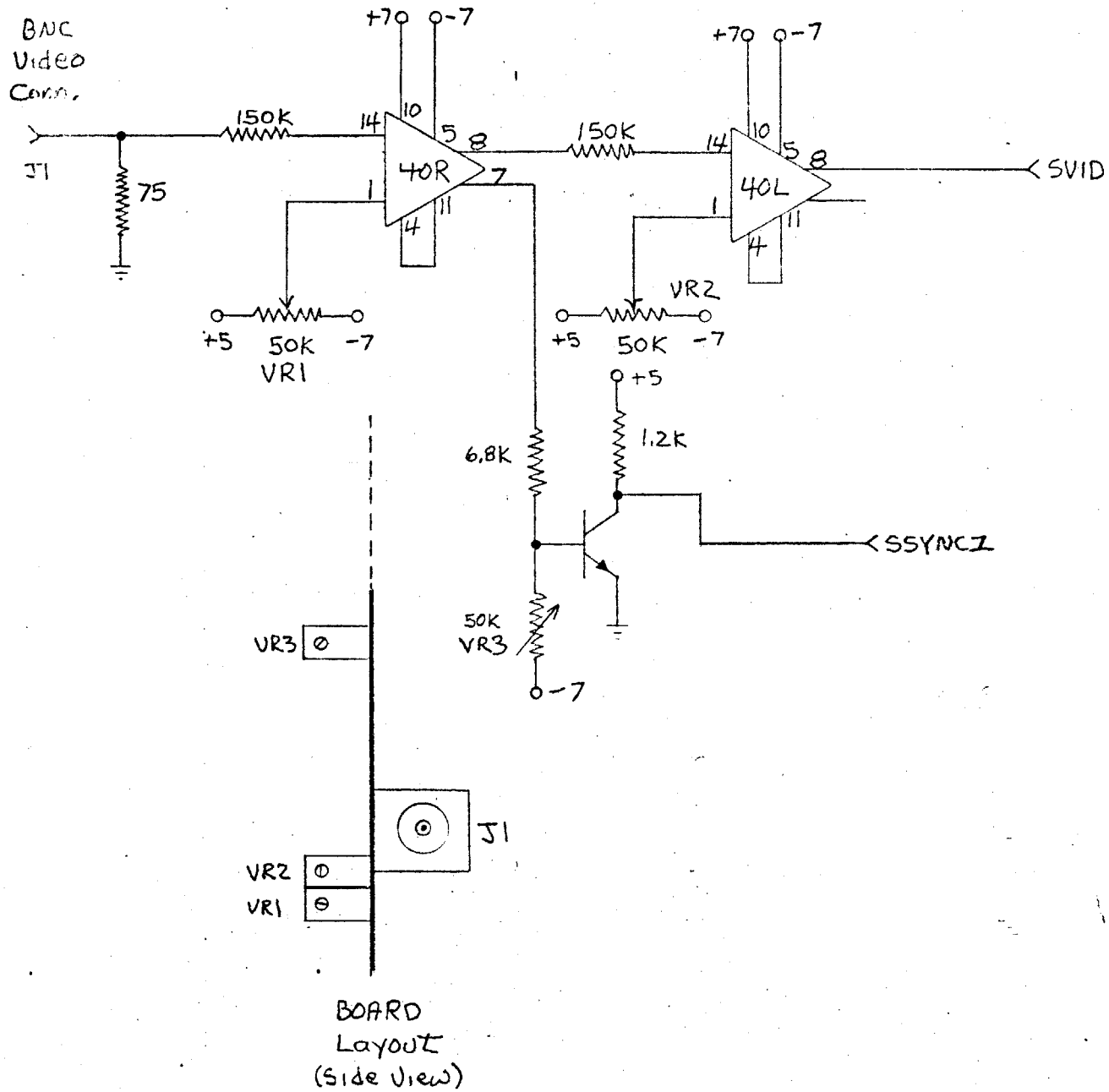
	MB0	MB1	MB2
lev1	0	0	1
lev2	0	1	1
lev3	0	1	0
lev4	1	1	0
lev5	1	1	1
lev6	1	0	1
Lev7	1	0	0

MEMORY OUTPUT DECODING



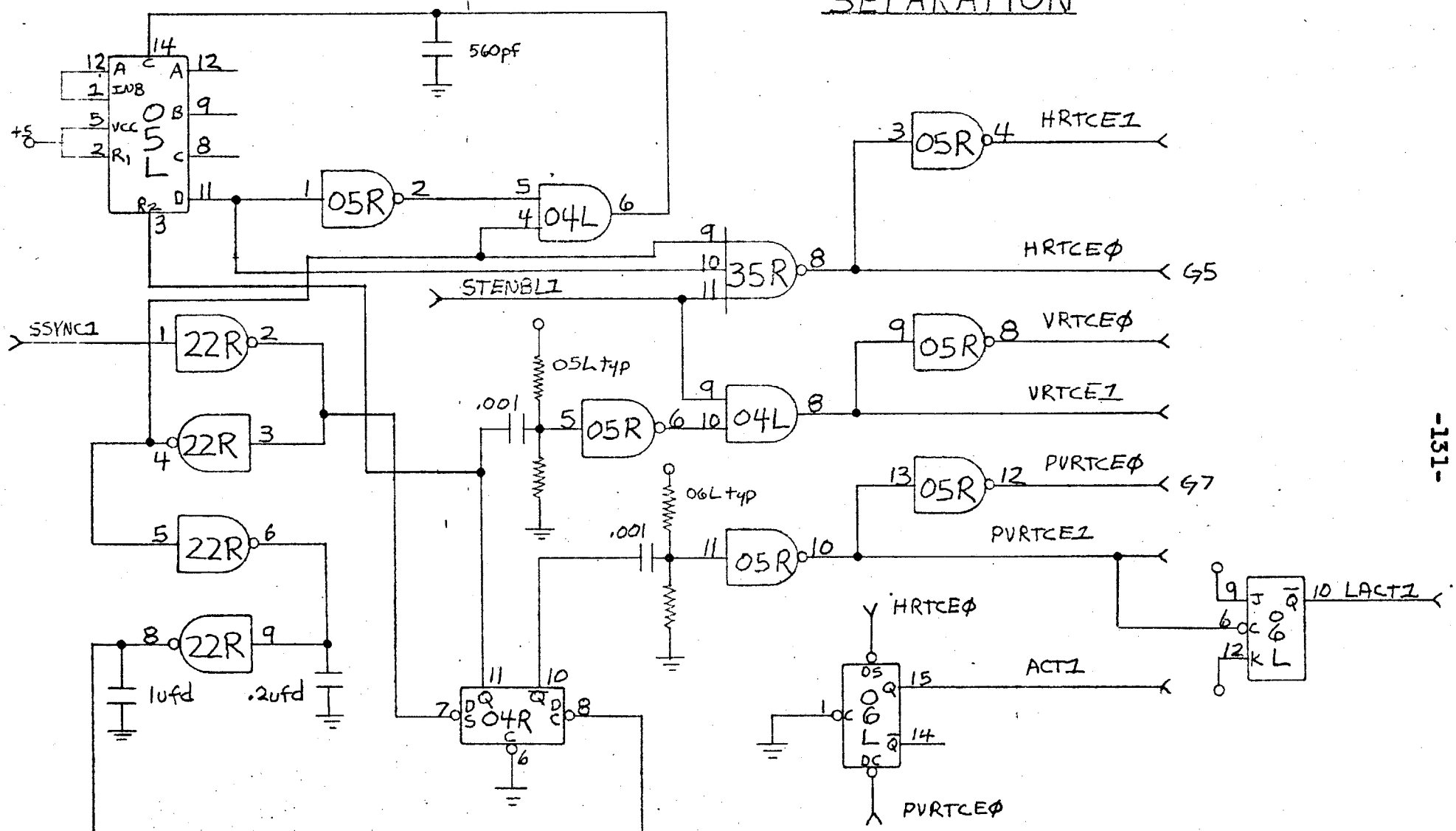
Vidicon Interface	
Sheet 8 of	

VIDEO AMP'S



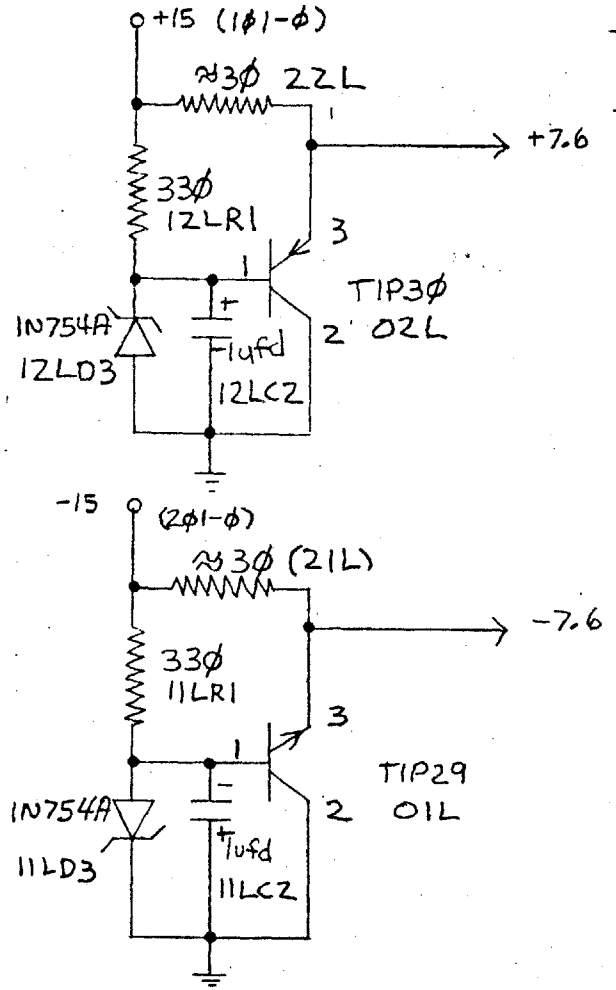
Vidicon Interface	
Sheet 9 of	

SYNC SEPARATION



VIDICON INTERFACE	
Sheet 10 of	

Power Supply



Vidicon Interface	
Sheet of	

APPENDIX B.

Thesis Writing Programs

```
0302R                   END
CLINE   0274R
CLST    027CR
COM     0288R
COMMNT  0094R
CRFIND  00B6R
EOFMES  0134R
ERRMES  017ER
FF      0194R
FORMFD  018CR
GARB    019ER
IC      004F
INBUF   01C0R
LF      01A8R
LINLIM  02DER
LISTER  0000R
LOOP1   004ER
LOOP2   0072R
LOOP4   0100R
MAXCHR  02DAR
MC      003D
NCHIN   02DCR
NCLINE  0210R
NCLST   0218R
NEWLIN  00C2R
NONCOM  0224R
OKAY1   00E4R
OULIN1  01BAR
PAUSE   02D8R
PLINE   01AAR
READ    01B2R
READER  0028R
REOM    00E6R
RERROR  0172R
RESTRT  0010R
ROKAY   004CR
RSTAT   01B4R
RUNAV   014AR
SAVER1  0300R
SAVREG  02E0R
SKIP    01A0R
UNVMES  015ER
WAIT    0196R
WSTAT   0198R
*
* LISTER 8 JAN 1973
* LISTS FORTRAN SOURCE PROGRAMS INPUT TO L.U.1
* ON L.U. 3 IN ACCORDANCE WITH THESIS FORMAT.
*
003D            MC      EQU   61
004F            IC      EQU   79
*
*SAVE ALL
0000R D000      LISTER  STM   0,SAVREG
```

```
      02E0R
0004R C810              LHI   1,1000
      03E8
0008R 41D0              BAL   13,NEWLIN
      00C2R
000CR 4010              STH   1,SAVER1
      0300R
0010R 4810    RESTRT    LH    1,SAVER1
      0300R
0014R C820              LHI   2,1
      0001
0018R 4830              LH    3,LINLIM
      02DER
001CR 0862              LHR   6,2
001ER 4870              LH    7,MAXCHR
      02DAR
0022R 0896              LHR   9,6
0024R 48A0              LH    10,NCHIN
      02DCR
*
*READ A LINE
0028R E110    READER    SVC   1,READ
      01B2R
*GET STATUS
002CR 4840              LH    4,RSTAT
      01B4R
0030R 4330              BZ    ROKAY
      004CR
*MOVE STATUS BITS
0034R CC40              SRHL  4,8
      0008
*EOM STATUS
0038R C540              CLHI  4,X'0088'
      0088
003CR 4330              BE    REOM
      00E6R
*AUNAVILABLE STATUS
0040R C540              CLHI  4,X'00A0'
      00A0
0044R 4330              BE    RUNAV
      014AR
*OTHER ERRORS
0048R 4300              B     RERROR
      0172R
*
004CR 0755    ROKAY     XHR   5,5
*GET INPUT CHAR.
004ER D345    LOOP1     LB    4,INBUF(5)
      01C0R
*IS CR?
0052R C540              CLHI  4,X'000D'
      000D
0056R 4330              BZ    CRFIND
      00B6R
```

```
    005AR C150              BXLE  5,LOOP1
      004ER
* NO CR FOUND, TRUNCATE LINE
005ER 0885              LHR   8,5
0060R CB80              SHI   8,1
      0001
0064R 07BB              XHR   11,11
*GET FIRST INPUT CHAR.
0066R D340              LB    4,INBUF
      01C0R
*IS A C (COMMENT LINE)?
006AR C540              CLHI  4,X'0043'
      0043
*YES.
006ER 4330              BE    COMMNT
      0094R
*
* NON-COMMENT LINE, MOVE REMAINING CHARS.
* ONTO NEXT LINE WITH CC1-6='     1'
*
*GET INPUT CHAR.
0072R D348    LOOP2     LB    4,INBUF(8)
      01C0R
*STOREIN OUTPUT BUFFER
0076R D24B              STB   4,NONCOM(11)
      0224R
007AR 0AB9              AHR   11,9
007CR C180              BXLE  8,LOOP2
      0072R
0080R 41D0              BAL   13,NEWLIN
      00C2R
*PRINT FIRST LINE
0084R E110              SVC   1,PLINE
      01AAR
0088R 41D0              BAL   13,NEWLIN
      00C2R
*PRINT CONTINUATION LINE
008CR E110              SVC   1,NCLINE
      0210R
*GO BACK FOR NEXT INPUT LINE
0090R 4300              B     READER
      0028R
*
* COMMENT LINE, MOVE REMAINING CHARS.
* ONTO NEXT LINE WITH CC1-6'C-----'.
*
0094R D348    COMMNT    LB    4,INBUF(8)
      01C0R
0098R D24B              STB   4,COM(11)
      0288R
009CR 0AB9              AHR   11,9
009ER C180              BXLE  8,COMMNT
      0094R
00A2R 41D0              BAL   13,NEWLIN
```

```
      00C2R
*PRINT INPUT LINE
00A6R E110              SVC   1,PLINE
      01AAR
00AAR 41D0              BAL   13,NEWLIN
      00C2R
*PRINT CONTINUATION LINE
00AER E110              SVC   1,CLINE
      0274R
*GO BACK FOR NEXT INPUT
00B2R 4300              B     READER
      0028R
*
* NICE SHORT LINE, PRINT STRAIGHT
* NICE SHORT LINE, PRINT STRAIGHT
*PAGE   3


*
00B6R 41D0    CRFIND    BAL   13,NEWLIN
      00C2R
*PRINT IT
00BAR E110              SVC   1,PLINE
      01AAR
*NEXT LINPUT
00BER 4300              B     READER
      0028R
*
* NEXT ROUTINE CHECKS IF ENOUGH ROOM IS
* LEFT ON THE PAGE FOR THE NEXT LINE, ELSE,
* IT SKIPS TO A NEW PAGE
*
*INCREMENT LINE COUNT
00C2R C110    NEWLIN    BXLE  1,OKAY1
      00E4R
00C6R E110              SVC   1,SKIP
      01A0R
00CAR E110              SVC   1,SKIP
      01A0R
00CER E110              SVC   1,SKIP
      01A0R
*GO TO A NEW PAGE
00D2R E110              SVC   1,SKIP
      01A0R
*SKIP A LINE...
00D6R E110              SVC   1,SKIP
      01A0R
*...OR TWO
00DAR E110              SVC   1,SKIP
      01A0R
*WAIT FOR CHARACTER
00DER E110              SVC   1,WAIT
      0196R
*CLEAR LINE COUNTER
00E2R 0711              XHR   1,1
```

```
*RETURN
00E4R 030D     OKAY1    BR     13
*
* END OF FILE ON INPUT
*
00E6R 41D0     REOM     BAL    13,NEWLIN
      00C2R
00EAR E110              SVC    1,SKIP
      01A0R
00EER 41D0              BAL    13,NEWLIN
      00C2R
00F2R E110              SVC    1,SKIP
      01A0R
00F6R 41D0              BAL    13,NEWLIN
      00C2R
00FAR 0788              XHR    8,8
00FCR C840              LHI    4,C'**'
      2A2A
*STORE A *
0100R D248     LOOP4    STB    4,INBUF(8)
      01C0R
0104R C180              BXLE   8,LOOP4
      0100R
*PRINT A LINE OF ASTERISKS
0108R E110              SVC    1,PLINE
      01AAR
010CR 41D0              BAL    13,NEWLIN
      00C2R
0110R E110              SVC    1,SKIP
      01A0R
0114R 41D0              BAL    13,NEWLIN
      00C2R
0118R E110              SVC    1,SKIP
      01A0R
*INFORM USER
011CR E120              SVC    2,EOFMES
      0134R
*RESTORE MOST REGS.
0120R D120              LM     2,SAVREG+4
      02E4R
*SAVE LINE COUNTER
0124R 4010              STH    1,SAVER1
      0300R
*RESTORE ALL REGS.
0128R D100              LM     0,SAVREG
      02E0R
*PAUSE
012CR E120              SVC    2,PAUSE
      02D8R
*GO TO RESTART POINT
0130R 4300              B      RESTRT
      0010R
*
*SVC 2 CODE
```

```
0134R 0007     EOFMES   DC     7
*LENGTH
0136R 0012              DC     18
*COMPLETE '
0138R 4C49              DC     C'LISTING
      5354
      494E
      4720
      434F
      4D50
      4C45
      5445
      2020
*
* L.U. 1 UNAVAILABLE
*
014AR E120     RUNAV    SVC    2,UNVMES
      015ER
*SAVE LINE COUNTER
014ER 4010              STH    1,SAVER1
      0300R
*RESTORE ALL REGISTERS
0152R D100              LM     0,SAVREG
      02E0R
0156R E120              SVC    2,PAUSE
      02D8R
015AR 4300              B      RESTRT
      0010R
*
015ER 0007     UNVMES   DC     7
0160R 0010              DC     16
*UNAVAILABLE'
0162R 4C55              DC     C'LU1
      3120
      554E
      4156
      4149
      4C41
      424C
      4520
*
* OTHER ERROR
*
0172R E120     RERROR   SVC    2,ERRMES
      017ER
0176R D100              LM     0,SAVREG
      02E0R
*STOP
017AR E130              SVC    3,0
      0000
*
017ER 0007     ERRMES   DC     7
0180R 000A              DC     10
*ERROR '
```

```
0182R 4C55              DC      C'LU1
      3120
      4552
      524F
      5220
*
*SVC OUTPUT CODE
018CR 2803    FORMFD    DC      X'2803'
018ER 0000              DC      0
0190R 0194R             DC      FF
0192R 0194R             DC      FF
0194R 0C0C    FF        DC      X'0C0C'
*
*
*STATUS CHECK
0196R 4803    WAIT      DC      X'4803'
0198R 0000    WSTAT     DC      *-*
019AR 019ER             DC      GARB
019CR 019FR             DC      GARB+1
019ER 0000    GARB      DC      *-*
*
01A0R 2803    SKIP      DC      X'2803'
01A2R 0000              DC      0
01A4R 01A8R             DC      LF
01A6R 01A9R             DC      LF+1
01A8R 200D    LF        DC      X'200D'
*
01AAR 2803    PLINE     DC      X'2803'
01ACR 0000              DC      0
*BUFFER START ADDR.
01AER 01BAR             DC      OULIN1
*BUFFER END ADDR.
01B0R 01FCR             DC      INBUF+MC-1
*
*ASCII INPUT SVC CODE
01B2R 4801    READ      DC      X'4801'
*READ STATUS
01B4R 0000    RSTAT     DC      0
01B6R 01C0R             DC      INBUF
01B8R 020FR             DC      INBUF+IC
*
*'
01BAR 2020    OULIN1    DC      C'
      2020
      2020
*INPUT LINE BUFFER
01C0R         INBUF     DS      IC
*
0210R 2803    NCLINE    DC      X'2803'
0212R 0000              DC      0
0214R 0218R             DC      NCLST
0216R 0260R             DC      NONCOM+MC-1
*
*'
```

```
0218R 2020     NCLST   DC     C'
      2020
      2020
*1'
021ER 2020             DC     C'
      2020
      2031
*SPACE FOR NON-COMMENT LINE
0224R          NONCOM  DS     80
*
0274R 2803     CLINE   DC     X'2803'
0276R 0000             DC     0
0278R 027CR            DC     CLST
027AR 02C4R            DC     COM+MC-1
*
*'
027CR 2020     CLST    DC     C'
      2020
      2020
0282R 432D             DC     C'C-----'
      2D2D
      2D2D
*SPACE FOR COMMENT LINE
0288R          COM     DS     80
*
*SVC PAUSE CODE
02D8R 0001     PAUSE   DC     1
*
* SYSTEM CONSTATNS
*
*MAX. CHARS./LINE
02DAR 003D     MAXCHR  DC     MC
*LENGTH OF INPUT LINE
02DCR 004F     NCHIN   DC     IC
*LINES/PAGE
02DER 0034     LINLIM  DC     52
*REGISTER SAVE AREA
02E0R          SAVREG  DS     32
*EXTRA SPACE FOR R1
0300R          SAVER1  DS     2
*
0302R                  END
CLINE    0274R
CLST     027CR
COM      0288R
COMMNT   0094R
CRFIND   00B6R
EOFMES   0134R
ERRMES   017ER
FF       0194R
FORMFD   018CR
GARB     019ER
IC       004F
INBUF    01C0R
```

```
LF        01A8R
LINLIM    02DER
LISTER    0000R
LOOP1     004ER
LOOP2     0072R
LOOP4     0100R
MAXCHR    02DAR
MC        003D
NCHIN     02DCR
NCLINE    0210R
NCLST     0218R
NEWLIN    00C2R
NONCOM    0224R
OKAY1     00E4R
OULIN1    01BAR
PAUSE     02D8R
PLINE     01AAR
READ      01B2R
READER    0028R
REOM      00E6R
RERROR    0172R
RESTRT    0010R
ROKAY     004CR
RSTAT     01B4R
RUNAV     014AR
SAVER1    0300R
SAVREG    02E0R
SKIP      01A0R
UNVMES    015ER
WAIT      0196R
WSTAT     0198R
EOJ
M:
```

```
02F8R                   END
BUFF    0220R
CADR    00D4R
CKCOM   009CR
COMCK   006CR
COMENT  0090R
COUT    00BCR
CR      028ER
CRSKP   0038R
EBUFF   0283R
EXIT    0214R
FAVER   020ER
FAVER1  0210R
FLAG    0284R
FLAG2   0286R
FLOOP   01EAR
FORM    0170R
FORMFD  028AR
GOGO    01B4R
LF      028CR
LOOP    012ER
NEW     0014R
NEXTLN  00F0R
OUTBF   0294R
PAVER   0168R
PAVER1  016AR
PAVER2  016CR
PAVER3  016ER
PRBUF   00FCR
PRET    0154R
PRETTY  0150R
PRINT   00E8R
PRNTR   011AR
RBUFF   0218R
RSTAT   021AR
SAVER   020CR
SKIP    0064R
SKP     0180R
SPACE   0292R
STAR    0290R
STATS   0288R
TABLN   01CAR
TAVER   0212R
TOF     000CR
UPCOM   0084R
WAIT    01A4R
WAIT1   01AER
* EL SUPER WHOOPIE ASSEMBLY LISTER FOLLOWS
*
*INPUTS ASSEMBLY LISTING FROM LU 1
*OUTPUTS TO DEVICE NUMBER AT ORG+2
*
*IFF ALL DATA SWTS ARE UP
*PROGRAM WAITS AFTER END OF PAGE
```

```
*UNTILL 'BREAK' IS DEPRESSED AND
*RELEASED
*OTHERWISE IT CONTINUES PRINTING
*WITH PROPER FORMAT FOR CONTINUOS
*PAPER
*TO ALIGN 8.5 BY 11.0 PAPER IN TTY
* USE FOLLOWING SEQUENCE
*   1)ALIGN TOP OF PAPER WITH TOP
*     OF RIBBON GUIDE
*   2)MANUALLY LINE FEED 4 TIMES
*   3)DEPRESS BREAK
*
*
*BREAK CAUSES RETURN TO M.A.G.I.C.
*
0000R C810              LHI    1,X'02'
      0002
*MAX NUM LINES
0004R C8C0              LHI    12,53
      0035
0008R C8B0              LHI    11,1
      0001
000CR C8A0     TOF      LHI    10,0
      0000
*FORM AND TABULATE
0010R 41F0              BAL    15,FORM
      0170R
*READ TAPE UNIT
0014R E110     NEW      SVC    1,RBUFF
      0218R
0018R 48E0              LH     14,RSTAT
      021AR
001CR 4230              BNZ    EXIT
      0214R
*SEE IFF COMMENT LINE
0020R D3E0              LB     14,BUFF+16
      0230R
0024R D4E0              CLB    14,STAR
      0290R
0028R 4330              BE     COMENT
      0090R
002CR C880              LHI    8,OUTBF
      0294R
*CHEXK FOR CR IN FIRST 32 CHARACTERS
0030R C8D0              LHI    13,32
      0020
0034R C8E0              LHI    14,BUFF
      0220R
0038R D39E     CRSKP    LB     9,0(14)
      0000
003CR D490              CLB    9,FORMFD
      028AR
0040R 4330              BE     SKIP
      0064R
```

```
*SAVE FOR REVERSAL LATER
0044R D298              STB    9,0(8)
      0000
0048R D490              CLB    9,CR
      028ER
*REGULAR LINE-GO PRINT
004CR 4330              BE     PRINT
      00E8R
0050R CAE0              AHI    14,1
      0001
0054R CA80              AHI    8,1
      0001
0058R CBD0              SHI    13,1
      0001
*MAYBEE A COMMENT-GO SEE
005CR 4330              BZ     COMCK
      006CR
0060R 4300              B      CRSKP
      0038R
0064R E110     SKIP     SVC    1,RBUFF
      0218R
0068R 4300              B      NEW
      0014R
006CR D3DE     COMCK    LB     13,0(14)
      0000
0070R D4D0              CLB    13,SPACE
      0292R
*CHECK FOR COMMENT
0074R 4330              BE     CKCOM
      009CR
0078R D2D8              STB    13,0(8)
      0000
007CR D4D0              CLB    13,CR
      028ER
0080R 4330              BE     PRINT
      00E8R
*NEXT CHARACTER
0084R CAE0     UPCOM    AHI    14,1
      0001
0088R CA80              AHI    8,1
      0001
008CR 4300              B      COMCK
      006CR
0090R 41F0     COMENT   BAL    15,PRBUF
      00FCR
0094R 022ER             DC     BUFF+14
0096R FFFF              DC     -1
*PRINT LINE STRAIGHT
0098R 4300              B      NEXTLN
      00F0R
009CR CAE0     CKCOM    AHI    14,1
      0001
*LOOK FOR SOMETHING OTHER THAN SPACE
00A0R D3DE              LB     13,0(14)
```

```
      0000
00A4R D4D0              CLB    13,SPACE
      0292R
00A8R 4330              BE     CKCOM
      009CR
00ACR D390              LB     9,CR
      028ER
00B0R D298              STB    9,0(8)
      0000
*SEE IF CR OR LF
00B4R D4D0              CLB    13,CR
      028ER
00B8R 4330              BE     PRINT
      00E8R
00BCR 40E0     COUT     STH    14,CADR
      00D4R
00C0R 41F0              BAL    15,PRBUF
      00FCR
00C4R 0292R             DC     SPACE
00C6R 0002              DC     2
00C8R 41F0              BAL    15,PRBUF
      00FCR
00CCR 0290R             DC     STAR
00CER 0001              DC     1
00D0R 41F0              BAL    15,PRBUF
      00FCR
00D4R 0000     CADR     DC     *-*
00D6R FFFF              DC     -1
*GO PRINT THE REAL LINE
00D8R 41F0              BAL    15,TABLN
      01CAR
00DCR C1A0              BXLE   10,PRINT
      00E8R
00E0R C8A0              LHI    10,0
      0000
00E4R 41F0              BAL    15,FORM
      0170R
00E8R 41F0     PRINT    BAL    15,PRBUF
      00FCR
00ECR 0294R             DC     OUTBF
00EER FFFF              DC     -1
00F0R 41F0     NEXTLN   BAL    15,TABLN
      01CAR
00F4R C1A0              BXLE   10,NEW
      0014R
00F8R 4300              B      TOF
      000CR
*  PRINTER ROUTINE
00FCR 4090     PRBUF    STH    9,PAVER
      0168R
0100R 40D0              STH    13,PAVER1
      016AR
0104R 40E0              STH    14,PAVER2
      016CR
```

```
0108R 40F0              STH    15,PAVER3
      016ER
010CR 07EE              XHR    14,14
010ER 40E0              STH    14,FLAG2
      0286R
0112R 48EF              LH     14,0(15)
      0000
0116R 48DF              LH     13,2(15)
      0002
011AR 4890     PRNTR    LH     9,FLAG
      0284R
011ER 4230              BNZ    LOOP
      012ER
*GET DATA
0122R D39E              LB     9,0(14)
      0000
*FLAG CR AND LF
0126R D490              CLB    9,CR
      028ER
012AR 4330              BE     PRETTY
      0150R
012ER DD10     LOOP     SS     1,STATS
      0288R
0132R 4240              BTC    4,EXIT
      0214R
0136R 4280              BTC    8,LOOP
      012ER
013AR D39E              LB     9,0(14)
      0000
013ER 9A19              WDR    1,9
0140R CAE0              AHI    14,1
      0001
*CHECK FOR LAST CHARACTER
0144R CBD0              SHI    13,1
      0001
0148R 4330              BZ     PRET
      0154R
014CR 4300              B      PRNTR
      011AR
0150R 40E0     PRETTY   STH    14,FLAG2
      0286R
0154R 4890     PRET     LH     9,PAVER
      0168R
0158R 48D0              LH     13,PAVER1
      016AR
015CR 48E0              LH     14,PAVER2
      016CR
0160R 48F0              LH     15,PAVER3
      016ER
0164R 430F              B      4(15)
      0004
0168R 0000     PAVER    DC     *-*
016AR 0000     PAVER1   DC     *-*
016CR 0000     PAVER2   DC     *-*
```

```
016ER 0000     PAVER3  DC     *-*
0170R 40F0     FORM    STH    15,FAVER1
      0210R
0174R 40F0             STH    15,FLAG
      0284R
0178R 40E0             STH    14,SAVER
      020CR
*LF TEN SPACES
017CR C8E0             LHI    14,5
      0005
0180R 41F0     SKP     BAL    15,PRBUF
      00FCR
0184R 028CR            DC     LF
0186R 0002             DC     2
0188R CBE0             SHI    14,1
      0001
018CR 4230             BNZ    SKP
      0180R
0190R 41F0             BAL    15,PRBUF
      00FCR
0194R 028CR            DC     LF
0196R 0001             DC     1
*FRONT PANEL
0198R C8F0             LHI    15,1
      0001
019CR 9BFE             RDR    15,14
019ER 08EE             LHR    14,14
01A0R 4230             BNZ    GOGO
      01B4R
*WAIT FOR BREAK
01A4R 9D1E     WAIT    SSR    1,14
*FOR FOR BREAK TO BE RELEASED
01A6R 4240             BTC    4,WAIT1
      01AER
01AAR 4300             B      WAIT
      01A4R
01AER 9D1E     WAIT1   SSR    1,14
01B0R 4240             BTC    4,WAIT1
      01AER
01B4R 41F0     GOGO    BAL    15,TABLN
      01CAR
01B8R 07FF             XHR    15,15
01BAR 40F0             STH    15,FLAG
      0284R
01BER 48F0             LH     15,FAVER1
      0210R
01C2R 48E0             LH     14,SAVER
      020CR
01C6R 430F             B      0(15)
      0000
* ROUTINE TO CR LF AND TAB 20
01CAR 40F0     TABLN   STH    15,TAVER
      0212R
01CER 40F0             STH    15,FLAG
```

```
      0284R
01D2R 41F0            BAL   15,PRBUF
      00FCR
01D6R 028ER           DC    CR
01D8R 0001            DC    1
01DAR 41F0            BAL   15,PRBUF
      00FCR
01DER 028CR           DC    LF
01E0R 0001            DC    1
01E2R 40E0            STH   14,FAVER
      020ER
*4 SPACES
01E6R C8E0            LHI   14,2
      0002
01EAR 41F0    FLOOP   BAL   15,PRBUF
      00FCR
01EER 0292R           DC    SPACE
01F0R 0002            DC    2
01F2R CBE0            SHI   14,1
      0001
01F6R 4230            BNZ   FLOOP
      01EAR
01FAR 07FF            XHR   15,15
01FCR 40F0            STH   15,FLAG
      0284R
0200R 48E0            LH    14,FAVER
      020ER
0204R 48F0            LH    15,TAVER
      0212R
0208R 430F            B     0(15)
      0000
020CR 0000    SAVER   DC    *-*
020ER 0000    FAVER   DC    *-*
0210R 0000    FAVER1  DC    *-*
0212R 0000    TAVER   DC    *-*
*
* EXIT BACK TO MAGIC
0214R E130    EXIT    SVC   3,0
      0000
*
*
0218R 4801    RBUFF   DC    X'4801'
021AR 0000    RSTAT   DC    *-*
021CR 0220R           DC    BUFF
021ER 0283R           DC    EBUFF
0220R         BUFF    DS    100
0283R         EBUFF   EQU   *-1
0284R 0000    FLAG    DC    *-*
0286R 0000    FLAG2   DC    0
0288R 0000    STATS   DC    *-*
028AR 0C0C    FORMFD  DC    X'0C0C'
028CR 0A0A    LF      DC    X'0A0A'
028ER 0D0D    CR      DC    X'0D0D'
0290R 2A2A    STAR    DC    X'2A2A'
```

```
0292R 2020     SPACE    DC     X'2020'
0294R          OUTBF    DS     100
02F8R                   END
BUFF     0220R
CADR     00D4R
CKCOM    009CR
COMCK    006CR
COMENT   0090R
COUT     00BCR
CR       028ER
CRSKP    0038R
EBUFF    0283R
EXIT     0214R
FAVER    020ER
FAVER1   0210R
FLAG     0284R
FLAG2    0286R
FLOOP    01EAR
FORM     0170R
FORMFD   028AR
GOGO     01B4R
LF       028CR
LOOP     012ER
NEW      0014R
NEXTLN   00F0R
OUTBF    0294R
PAVER    0168R
PAVER1   016AR
PAVER2   016CR
PAVER3   016ER
PRBUF    00FCR
PRET     0154R
PRETTY   0150R
PRINT    00E8R
PRNTR    011AR
RBUFF    0218R
RSTAT    021AR
SAVER    020CR
SKIP     0064R
SKP      0180R
SPACE    0292R
STAR     0290R
STATS    0288R
TABLN    01CAR
TAVER    0212R
TOF      000CR
UPCOM    0084R
WAIT     01A4R
WAIT1    01AER
```