REPORT NO. UMTA-WA-06-0009-79-1

VEHICLE DATA ACQUISITION SYSTEM

OCTOBER 1979

DEPARTMENT OF TRANSPORTATION ★ UNITED STATES OF AMERICA ★

AUTOMATED GUIDEWAY TRANSIT TECHNOLOGY PROGRAM

U.S. DEPARTMENT OF TRANSPORTATION
Urban Mass Transportation Administration
Office of Technology Development and Deployment
Washington, D.C. 20590

This document is available to the U.S. public through
the National Technical Information Service,
Springfield, Virginia 22161

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
WA-06-0009-79-1		

4. Title and Subtitle Vehicle Data Acquisition System	5. Report Date October 1979
	6. Performing Organization Code 272
7. Author(s) Andrew Haug	8. Performing Organization Report No.
9. Performing Organization Name and Address Port of Seattle P. O. Box 1209 Seattle, Washington 98111	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. WA-06-0009
12. Sponsoring Agency Name and Address U. S. Department of Transportation Urban Mass Transportation Administration Office of Technology Development and Deployment Washington, D.C. 20590	13. Type of Report and Period Covered Final Report
	14. Sponsoring Agency Code UTD-42

15. Supplementary Notes

16. Abstract

A Vehicle Data Acquisition System (VDAS) has been developed by the Port of Seattle which collects and continuously records 20 minutes of data from 34 test points in a specially instrumented Sea-Tac Satellite Transit System vehicle. The recorded data is useful to maintenance personnel to rapidly diagnose intermittent and total vehicle failures, thus reducing vehicle downtime and maintenance costs. This system uses an onboard microprocessor system based on the Intel 8080 and a semiconductor memory to record data which can then be dumped to a microprocessor terminal in the maintenance area for analysis. A six-month in-service test on one vehicle demonstrated that the VDAS can speed up corrective maintenance as well as identify marginal conditions and incipient failures.

17. Key Words	18. Distribution Statement
Automated Guideway Transit Data Acquisition System Vehicle Maintenance	This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

☆ U. S. GOVERNMENT PRINTING OFFICE : 1973 725-504/328

I. BACKGROUND

A. General

In October, 1976, the Port of Seattle requested Department of Transportation (DOT) funding for a Vehicle Data Acquisition System (VDAS) which was to be used on the Satellite Transit System (STS) at Sea-Tac International Airport. That request was granted in July, 1977 and this document constitutes the final report for the project.

The motivating factor behind the VDAS concerned the amount of time and energy involved in maintaining the operational readiness of the STS vehicles (12 cars of the SLT class). At the time the grant request was made, 75% of maintenance labor was being expended on troubleshooting and only 25% on actual repair. The maintenance group realized that as the passenger volume increased at Sea-Tac, the need for more vehicles would make the system more complex and the system integrity would begin to disintegrate. Normally, during the off peak periods of the year, half the twelve car fleet is required to provide service. Thus, if a car had a known intermittent problem, the luxury exists of keeping that vehicle out of service until the problem is finally isolated. Due to the aforementioned anticipated increase in passenger volume, the Port has already purchased twelve new vehicles for future delivery and is presently running at full capacity. Obviously, the proposed data acquisition system has come at a most convenient time.

B. History

Since 1973, the Sea-Tac International Airport has been operating a subway system providing transportation between the main terminal and the two satellite terminals through two independent underground loops (see Figure 1). A separate shuttle section interconnects the two loops in the main airport terminal.

During the six years of service the maintenance on the transit vehicles has been continually updated and improved. Most mechanical equipment failures could be found and corrected through periodic maintenance procedures. Electronic control malfunctions on the vehicles have always been hard to find due to their dynamic nature. The vehicles are unmanned in normal service and it would be a tedious task to manually monitor a vehicle while it is in that mode. The idea was born to develop a system that monitors the vehicle control logic in the same manner in which a flight recorder assembles data on commercial airliners. The ability to look at the sequence of events occurring prior to the perceived fault would allow technicians to spend more time improving the transit system rather than tracking down intermittent repetitious fault conditions.

C. Project Theory

We began the project with two distinct goals in mind:

1. To significantly reduce the maintenance costs of the STS through decreasing the number of failures and maintaining a high level of availability.

2. To determine if sufficient long-term maintenance cost improvement could be realized by increasing the number of VDAS units in the vehicle fleet.

The VDAS would primarily be used in identifying elusive, intermittent problems by "capturing" critical performance parameters during the most recent twenty minutes of operation. This data would be analyzed, and a picture of the operating system before, during and after failure would be developed.

II. DESIGN CONSIDERATIONS

The STS vehicles are automatically controlled by hardwired wayside controls. A computer system supervises the train operation in the loop, but it does not have to be running for loop control.

The vehicles receive speed and stop commands from the wayside through antennas underneath the vehicle, stopping in alignment with the station door and all interlocking switching for multiple car trains. All of the ATO equipment is located in the vehicles' control consoles at the front and rear (see Figures 2 and 3).

The VDAS ties into these different control circuits at strategic monitor points. These points were carefully selected such that examination of their status at any given time should yield accurate system information directly related to sub-system function, thereby isolating problems to specific sub-systems. The maintenance crew keeps two spare electronic sub-system units on hand at all times. Thus, when the VDAS indicates a faulty unit, it can quickly be replaced resulting in significant reduction in downtime.

Based upon past failure history the Maintenance personnel selected 34 test points for the VDAS (See Appendix A). They consist of four major types of signals:

1. Static relay voltages. All relays feed off the +36 VDC vehicle battery.

2. Static signals between control units. These are digital in nature, alternating between 0VDC and either +6VDC or +15 VDC.

3. Analog signals within a feedback loop. Threshold levels are indicative of different modes of operation.

4. Pulse integrations. Peak detectors and control circuitry for interrupt enabling.

A breakdown of the number of test points located within specific subsystems may be found in Table 1.

Early in the system development, we decided to use as many off-the-shelf products as possible. The Port is not set up for developmental work and, in addition, Port labor costs are too high for that type of activity.

We chose a microprocessor for the VDAS controller and initially wanted to use a cassette tape for transferring the accumulated data. The cassette idea was later scrapped in favor of a microdiskette system for its better reliability and ease of usage. The diskette is not a dedicated unit. When a failure occurs, the maintenance personnel connect the diskette drive to the vehicle VDAS and dump the contents of memory onto the diskette for data reduction and analysis.

A Prolog microprocessor system was chosen primarily for its convenient 19 in. rack mounting which is the vehicle format. (See Figure 2.) Height limitations within the vehicle eliminate most of the exisiting single board computers. It might be emphasized that any of the commercially available microprocessing systems are capable of performing the tasks necessary were it not for the aforementioned size limitation. If this had been known in advance, the vehicles could probably have been designed to accommodate other systems.

For program development and data formatting, a SOL computer was purchased along with a North Star Disk and accompanying Disk Operating System (DOS) (see Figure 4). 8080 assembly coding and Basic program generation capabilities are part of the SOL software configuration. The 8080 assembly code is used for the actual data acquisition on the vehicle and a Basic language program is resident in the SOL computer for data reduction and formatting.

We chose to consider the acquired data as a single contiguous file which would simplify our data handling procedures. The system was designed such that acquired data would be stored in consecutive memory locations until the top of RAM was reached (see Figure 5 for memory mapping). At that point, the address counter would "roll-over," allowing the newest data to be stored at location 2000, writing over previously stored information. To keep track of the proper sequencing in the data stored, the relative time of occurrence accompanies the data bytes (see Figure 6 for data storage formatting).

We did consider using an Intel development system which would have reduced the software development time. The advantage with development systems is their ability to function in-circuit as the software is being written. For our purposes, this advantage did not outweigh the considerably higher cost.

To acquire the data from the vehicle itself, we designed signal-conditioning circuit cards which were compatible with the Prolog mainframe. This resulted in a reliable customized interface suitable for accurate reading of the parameters we were after. For systems requiring a greater number of monitoring points, this may become an unwieldy solution, but for the number of points we were monitoring, it was not an insurmountable task.

One of the problems we ran into interfacing with the STS vehicle concerned ensuring that the VDAS would not interfere with the normal operation of the ATO controls. This was solved through the vehicle contractor, Westinghouse Corp., and resulted in a termination board (see Figure 7) installed between the test points and the interface electronics. With the termination board installed, any type of VDAS circuit violation such as shorted output, or voltage feedback, would not affect the vehicle circuitry.

Another design problem occurred when we decided to use the diskette drive. Having already purchased the Prolog computer, we were faced with the task of interfacing the North Star Disk drive to it. The Prolog System had an inverted data bus and used differing timing signals for its strobing, for example. These problems were solved through the addition of another interface card (Figures 8) which conditioned these signals for acceptance by the disk controller.

Because the test vehicle operates off a non-standard voltage level (36 VDC), we had some difficulty locating a suitable power supply for the VDAS. We needed +5, +15, and -15 VDC. We eventually found a supply with sufficient reliability to satisfy our requirements. Previous supplies had a tendency to burn out during revenue operation, causing considerable delay in the VDAS evaluation.

We wanted to be able to interrupt the VDAS software at any time (i.e., in the event of a system failure) to allow us to dump the memory contents on disk. Also, we knew that a timing relationship was required for the data to have any meaning. To achieve these goals, we developed a hardware priority interrupt system using a Prolog priority interrupt card. In this way, a real-time clock would generate an interrupt, the interrupt card would determine its priority and branch to the appropriate service routine while holding the other interrupts in a high impedance state until the task was finished. The dump procedure is initiated by depressing a button on the VDAS front panel (see Figure 2).

One consideration that has a direct bearing on the efficacy of the system is the amount of memory required. We needed to store about 20 minutes' worth of vehicle data to allow the Maintenance personnel sufficient time to get to the test vehicle following a failure and initiate the dump procedure. We started with the assumption of 6 data words, which yields one bit for each of 34 monitored conditions plus eight bits for resolving the vehicle speed to 0.1 mph, leaving six expansion bits which are unused at this time. In addition, two words relating to the running real time are required. Referring to Figure 9, this would require 96,000 words of core memory.

This is an obvious problem as most microprocessors have the capability of addressing only 65K of memory, the 8080 included. So we decided to employ the maximum 65K less whatever the controlling software required, but only store the data whenever a change occurred in any given bit of the 6 data words. This has been a workable solution yielding 56K of usable core memory for data storage (main program and related utilities constitute 9K of total memory). This allows about 12 minutes of data storage which has proven sufficient for our needs.

III. EQUIPMENT

A. Hardware

1. SOL Terminal Computer--See Figure 10.
   This equipment was purchased for software development and data analysis. It operates as a stand alone computer under control of a program contained in a personality module which allows the assembled output of developed programs to be compatible with the VDAS CPU, in this case the Intel 8080. The mainframe is designed around the S-100 bus, an industry standard geared for the hobby market. In essence, SOL combines a Central Processing Unit (CPU) with several S-100 peripheral modules-memory, keyboard interface, video interface, audio cassette interface, plus parallel/serial I/O interfaces. The system also includes a five slot backplane board for adding other I/O or memory modules that are S-100 compatible.

   The SOL uses an 8080 as its CPU for control of system functions. The internal structure of the CPU employs 3 separate buses:

   i. 16 bit address bus.
   ii. 8 bit bi-directional data bus.
   iii. 28 bit Control bus.

   The S-100 bus is a composite of these buses and other control signals arranged in a standardized format such that I/O modules can be developed independent of the host system yet with the certainty that they will be operational when installed in the S-100 backplane.

   In addition to the S-100 bus structure, the SOL uses an eight bit keyboard input port, an eight bit parallel input port, an eight bit parallel output port and an eight bit sense switch logic input port.

   A Universal Asynchronous Receiver/Transmitter (UART) is installed for processing serial data from the audio cassette I/O circuits and the serial input line. This is advantageous since the UART would otherwise have to be incoporated as a software module.

All CPU data and address lines are tri-stated to allow multiple device connection. The SOL also has video generation capability in ROM which allows a dot matrix format for video output.

As previously mentioned, we decided to use a diskette for mass storage rather than the audio cassette. We chose a North Star Micro-Disk System for its availability and S-100 bus compatibility. To allow the disk to "talk" to the SOL, we purchased a software package which contained all the I/O routines required to interface the North Star software with the SOL hardware. This package fit entirely within the 256 byte block North Star has allocated for its I/O area thus taking nothing away from the user's workspace, and without requiring any hardware interfacing.

Aside from the disk unit, one other peripheral was installed in the SOL system, a Practical Automation Printer.

This unit gives us hardcopy of our system software and the data from the VDAS. It has selectable baud rates (up to 1200 bps in serial mode) and uses a 20mA current loop or 8 bit parallel bus for data input.

2. Prolog (See Figure 11)

The basic Prolog System consists of eight printed circuit boards:

i. Priority Interrupt (#8118-1)
ii. I/O (#8113-1)
iii. 32 bit input (#8114)
iv. CPU (#8821)
v. RAM (#8119 - 4 each)

These boards are manufactured by Prolog to fit into a 19 inch wide card rack with the S-100 bus backplane wiring. Their usage is fairly straightforward.

To provide the necessary signal conditioning to the test points prior to inputting them to the Prolog System, the Port built four prototype custom PCB's:

i. 9001 - 36 vdc input card inputs relay logic to system via opto-isolators.
ii. 9002 - CMOS input card. Provides high input impedance and also contains hardware tachometer scaling for determining vehicle speed.
iii. 9003 - OP AMP Analog input card.
iv. 9004 - Control Card.

These cards were wire-wrapped on a standard Prolog utility dip card (#P561) having a 56 pin card edge connector consistent with their off-the-shelf system boards. Details of these custom PCB's are shown in Appendix B.

To allow transferral of acquired data, we purchased a controller from North Star and adapted it for use with the Prolog System (see Figure 8). The controller was configured such that when the vehicle experienced a failure, a technician would connect the disk unit to the Prolog System and press the "DUMP" button (see Figure 2) generating a hardware interrupt, halting the CPU and branching to the memory dump routine. We ran into some initial problems in locating the software properly because the I/O routine required certain locations for its use and we had previously used those same locations. This is definitely something which should be researched before purchasing a similar system.

B. Software

There are two major programs developed for the system:

1. VDAS Data Handler

2. Data Analysis Routine

The SOL computing system, in conjunction with the North Star DOS, was used for program development.

This section will address just the routines developed for the VDAS itself rather than delve into all the necessary bookkeeping procedures peculiar to our particular system. The rationale behind this is that the important information is the manner in which the data was gathered, and how that data was assembled into a form usable by our maintenance group. This information would be valuable to future users, whereas the manner of implementation would not necessarily be of value to a user with a different system, which would probably be the case.

1. VDAS Data Handler

This is an assembly language program written for the Intel 8080 microprocessor. Data from the Westinghouse vehicle is input to the processor every 100 msec. If any data has changed from the last sample, all data is stored in random access memory along with a time reference. If no data has altered, no storage will take place.

A memory wrap-around is provided when the top of memory (FFFF hex - see Figure 5) is reached. The program revitalizes the memory pointer to the beginning of storage (2000 hex) such that "new" data will write over "old" data. In this manner it is possible to get at least twelve (12) minutes worth of data at any given time.

The program is interrupt driven such that when not executing one of the interrupt routines, the CPU is in a halt state with interrupts enabled (the processor automatically disables interrupts upon initiation of an interrupt subroutine). When an interrupt is received a "RST" (restart) instruction vectors the software to the start address of the routine which handles the particular interrupt. The only exception is the "Power-On" Routine which takes advantage of the fact that the processor automatically vectors to address zero when power comes on. Thus, the hardware interrupt procedure provided by the Prolog System is not needed and address zero contains logic appropriate for handling the power-on situation.

The remaining logic is arranged as a continuous assembly of the following interrupt modules:

i. Monitor
ii. Disable
iii. Memory Dump
iv. Reinitialize

The monitor routine (see Appendix C for flow diagrams) stores data from the vehicle into memory. It contains logic to determine whether data has been altered since the last time data was stored. If that is true, the data is stored. A memory limit check is also provided incorporating the memory wraparound feature when memory overflow occurs. It is driven with a clock which provides an interrupt every 100 msec.

The Disable routine captures one last frame of data including the time reference--and then disables the real time clock interrupt to "freeze the memory." It also readies the system for dumping the contents of memory by initializing the appropriate handshaking flags. The time reference is not correlated to Greenwich time. It has meaning only within the VDAS frame of reference.

The memory dump routine outputs the contents of memory from 2000-FFFF hex onto a microdiskette with the oldest data placed at the beginning of the diskette. If the overflow flag has been set, then the current memory pointer points to the oldest data. If overflow has not occurred, then the dump takes place from 2000 hex to the current value of the memory pointer.

The reinitialize routine clears flags, turns off indicators and clears memory of old test data. The real-time clock is re-enabled and the system is once again ready for a new set of data.

There are separate subroutines called by the interrupt routines which accomplish straightforward tasks such as memory limit checking, writing to disk and inputing data from the test vehicle.

All of the interrupt routines are designed to be completed within the interrupt disable timeout which is automatically generated as part of the hardware interrupting procedure. Following completion of their tasks, the routines return to the idle mode with all interrupts fully enabled.

2. Data Analysis

The Data Analysis program is a Basic language program which is used for gathering maintenance information. It is run only on the SOL computer and uses the data dumped out from the VDAS as input. The principle features of this program are:

i. Test point data from the Westinghouse vehicle is displayed in binary format on a CRT or a line printer depending on user selection. The data is identified with appropriate heading and a legend is included at the end of each display for further explanation (see Figure 12 for a representative example).

ii. Three levels of resolution are provided to enable focusing on an area of interest within a considerable amount of data. The lowest level of resolution encompasses the entire data with a scan of 48 frames, which is approximately 1 out of every 150 data points or frames. The medium level of resolution involves a scan between any two points within the low resolution scan. There are 10 frames which is about 1 out of 15 data points. The high resolution scan encompasses every frame within a specified range.

The program is structured such that the operator may go from one level of resolution to another with relative ease. The scan at each resolution may be truncated to add even further flexibility in usage. Also, it is possible to effect a pause in the program by keyboard entry. This enables the user to examine data on the CRT more easily.

iii. The user may choose between a total scan or time-specified scan at the lowest resolution upon entry to the program. The time-specified scan can be used effectively if the user knows approximately the time between when a fault occurred and the "stop monitor" button is depressed. Entering this time will allow the user to truncate the display to the area of interest more quickly.

IV. TEST PERIOD RESULTS

The VDAS data test points were chosen by the maintenance group based upon their previous experience in diagnosing vehicle failures. For instance, there is a point which determines direction (forward or reverse), this point, in conjunction with another point which determines speed (brake, acceleration, maintain speed) can indicate the dynamic performance of the vehicle at any time. If the data indicated that the control logic was in an accelerating reverse mode when it should have been in forward braking, you can immediately narrow your choices to the appropriate areas.

Figure 12 shows a sample Data Analysis Output. The time is referenced to the vehicle data itself, having no relationship to the actual time of occurrence. The time reference does, however, indicate a timing relationship between the actual test points themselves. It is very important to plot the dynamic performance of each test point versus the time of occurrence. It must be remembered that data is stored only when one of the data points alters state. With a sampling rate of 100 ms there often are periods of time upwards of one to three seconds where data is not stored. By plotting events according to their true time of occurrence, one may develop an accurate assessment of how the control logic was behaving during the time a failure occurred.

Appendix A contains the vehicle control schematics showing where each of the test points are located. As can be seen, they tend to follow a certain line of logic from inception to completion. In this manner, the technicians can isolate failures to a given sub-system. If the data shows that a start command was received (test point 3) but that the motors weren't energized (test point 4), then a fault is indicated in the motor control logic.

Appendix D contains data analyses for three separate failures recorded by the VDAS.

The first failure occurred during an "add train" operation where the test vehicle was being placed in revenue operation. The data shows a loss of "code load" 1.8 seconds after the wayside transmitters came on, and only .3 second after the wayside transmitters started sending the D.O.C. word.

The second failure occurred at the main terminal in the South loop. The test vehicle was operating in a two-car configuration and the vehicles were given a command to reverse their direction following a scheduled stop at the main terminal.

In both cases, it appears that the interference from the wayside Info transmission prevented speed commands from being decoded. This is evidenced by the intermittent nature of code loaded after the A relay was reset. This is due to jamming of the speed code signal by the wayside information signal.

The third failure occurred during normal operation as the vehicle was attempting to accelerate from 15 to 20 mph. The lack of an overspeed failure (test points 31 and 32) and presence of a balance failure (test point 33) and an A-relay dropout 1 second later suggests a ramp failure or tach failure. Specifically, "+ ramp" is suspect because of a speed decrease (test point 20) and is one of the two servo inputs, the other being "Tach 1." The 20 mph crystal oscillator which generates the "+ ramp" signal probably didn't start oscillating. The stability of code loaded (test point 34) from the speed decoder does not suggest a failure in the decode circuitry.

In addition to these examples, we found many other faults through a combination of VDAS and troubleshooting techniques. For example, the vehicle stopped on a reset from the wayside controller. Upon examination of the VDAS data we discovered that Tach 2 (test point 21) had dropped out for no apparent reason. (Tach 1 & 2 are identical circuits backing each other up. We determine speed from Tach 1 and simply detect the absence or presence of Tach 2.) The technicians immediately examined the Tach 2 signal and found that the waveform was rounded showing 5MHZ noise on the edges. The Tach 2 circuitry was replaced and the car re-entered revenue operation after being out of service a mere three minutes.

Another advantage we have found in using the VDAS concerns the isolation of recurring trouble spots. Quite often, a failure is caused by noise on the vehicle send and receive circuitry. VDAS will pinpoint this problem and document it in such a fashion as to convince management that it is worthwhile to re-design the antennas or associated circuitry causing the problem. In this manner, the VDAS is used as a tool to reduce the amount of time spent on troubleshooting and increase the man-hours expended on improving system efficiency.

V. CONCLUSION AND RECOMMENDATIONS

The maintenance group is thoroughly convinced of the efficacy of the VDAS. They only wish they had more test units available. This would be the prime recommendation arising from the project. We configured one test vehicle to determine if the system had value and wanted to ascertain whether installation of the VDAS on all vehicles would be cost effective. Both goals have been met. The VDAS pinpoints vehicle problems, and is the only way we know of which will provide maintenance with dynamic system information. We have seen problems on the test vehicle reflected in other vehicles, but have no way of knowing if the other vehicle is exhibiting identical symptoms. Yet, even from information gleaned from the single test vehicle, we can narrow the guesses to a manageable few in the instances where similar problems appear in other vehicles.

The VDAS should be viewed as a tool, not a panacea to all system problems. While it is possible to monitor a great many testpoints through expansion, it is felt that this approach would only serve to confuse the issue which is to significantly reduce vehicle downtime. This end result is better served by carefully selecting testpoints according to their own inherent value within the system. This, in conjunction with highly trained personnel, can reduce that downtime through failure analyses conducted with a systematic approach provided by using the VDAS as a diagnostic tool. The results from the data analysis must be viewed from the systems viewpoint.

It is important that personnel performing the analysis know exactly what should be happening at any given time. The VDAS can indicate something is awry, but it takes the skill of a trained technician to interpret that clue properly.

We are presently forging ahead with the task of incorporating VDAS into all of our vehicles. It is not necessary to have a computer in each vehicle. We feel that by configuring each vehicle such that a computing unit can easily be installed and removed as an individual vehicle exhibits an intermittent problem we can reduce the total system cost while maintaining a high level of system integrity. Our plans call for three or four computing units for a total fleet of 24 vehicles.

It is also important to keep accurate records of system problems such that an empirical database may be constructed. This is especially true in the case of those problems which are intermittent. They may occur only once every week or less, but with proper record keeping, enough information will be gathered such that the solution will eventually come to light. These records can also be very helpful in training new personnel and keeping present personnel up-to-date on the vehicle electronics.

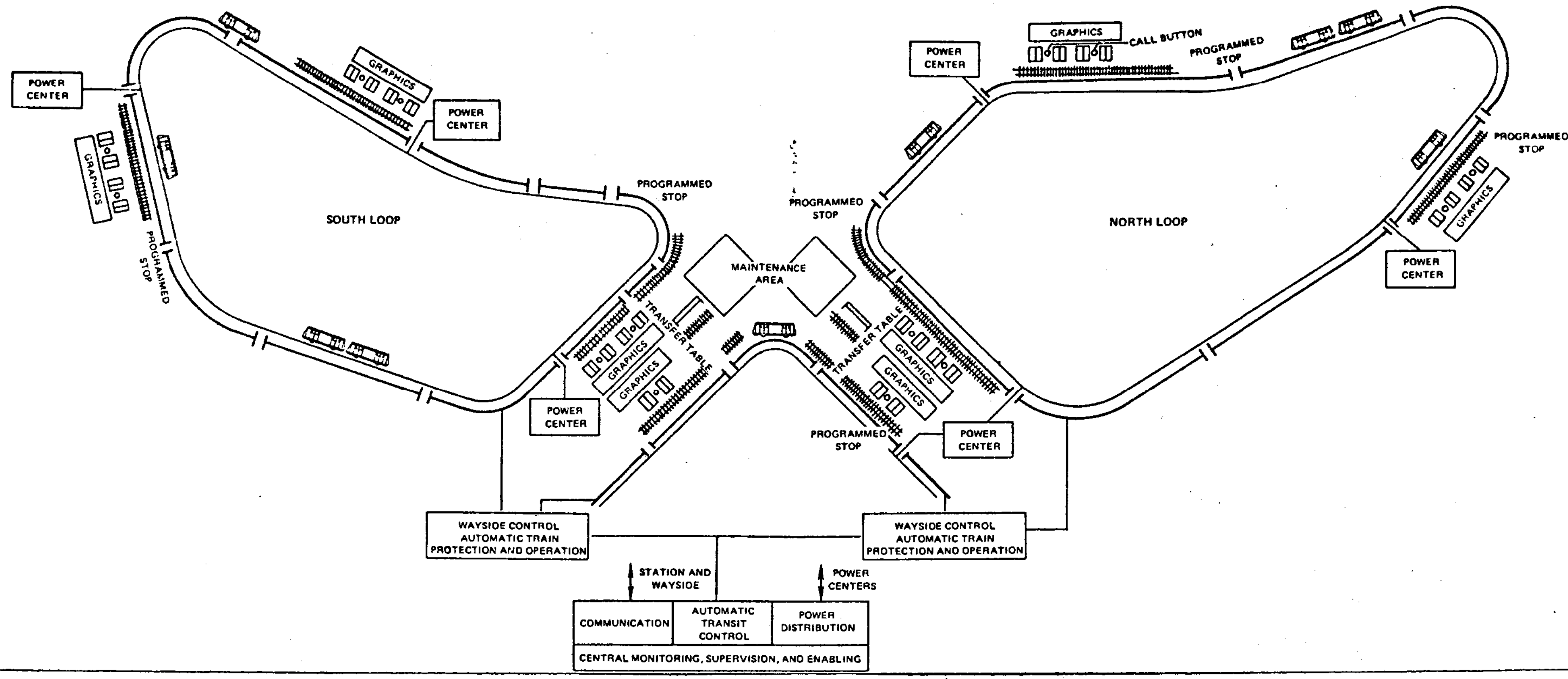


FIGURE 1

SEA-TAC INTERNATIONAL AIRPORT SATELLITE TRANSIT SYSTEM

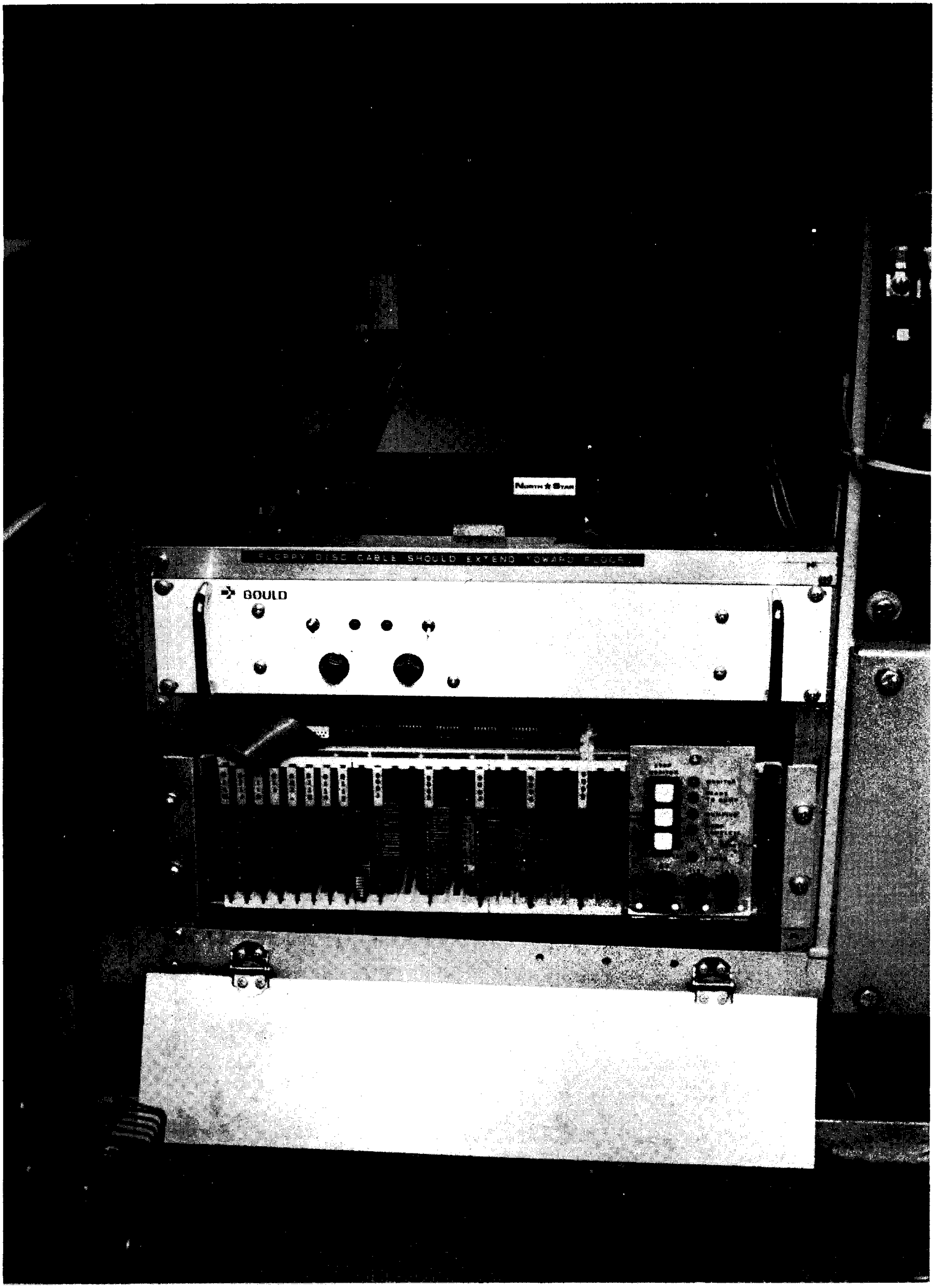


FIGURE 2

Picture of VDAS Control Circuitry

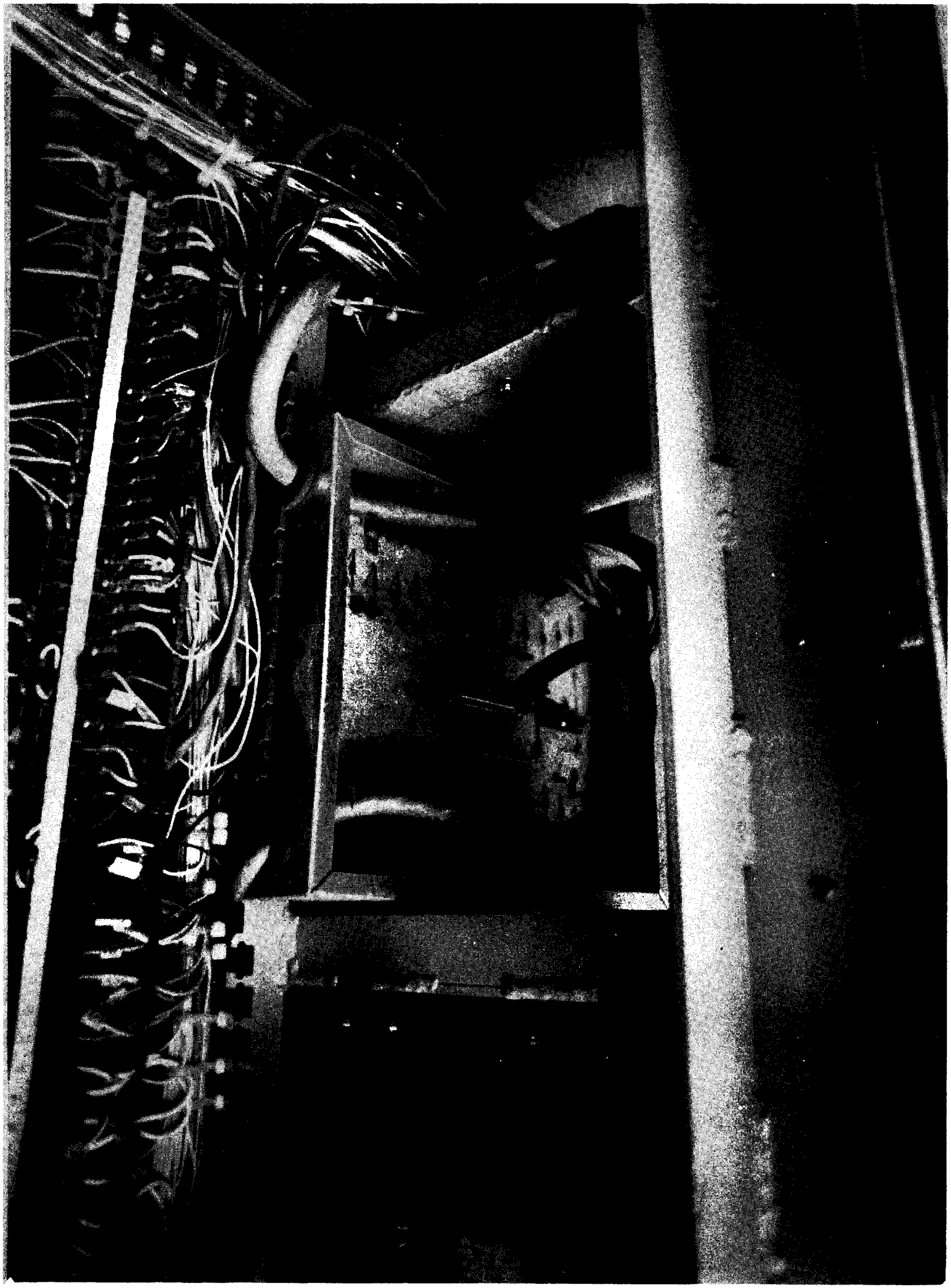


FIGURE 3

Picture of VDAS Termination Board

FIGURE 4

Picture of SOL Computer with North Star Disk

PROLOG

0000	003F	Interrupt service initialize routine
0040	00FF	Power on initialize
0100	017F	Power on routine
0180	0277	Input routine
0280	02FF	Re-initialize routine
0300	037F	Dump routine
0380	03FF	Disable routine
0400	0BFF	Unused
0C00	0DFF	Subroutines: STWD1, LDWD1, TRNIN, LIMIT
0E00	0FFF	Disk write routine
1000	17FF	RAM
1800	1FFF	Disk boot
2000	FFFF	User memory (RAM)

SOL

0000	1FFF	AL-8 portion of ALS-8
2000	29FF	DOS
2A00	5FFF	North Star Basic
6000	BFFF	User memory (RAM)
C000	C7FF	SOLOS ROM
C800	CBFF	SOLOS RAM
CC00	CFFF	Video display module (RAM)
D000	E7FF	Unused
E800	EBFF	North Star ROM
EC00	FFFF	Unused

(partitioning not to scale)

PORT OF SEATTLE COMMISSION

PROJECT STS-VDAS

TITLE MEMORY MAP: PROLOG & SOL SYSTEMS

DRAWN BY A. HAUG DATE ____ APPROVED ____

DESIGN NO.

DRAWING NO.

POS 120-9

FIGURE 5

30 TESTPOINTS

DEVICE ADDRESS

BIT ASSIGNMENT #

DEVICE ADDRESS	(MSB) 8	7	6	5	4	3	2	(LSB) 1
IN '00	TP-7H BR —	TP-7L DOC	TP-6 DTX	TP-5 MDL	TP4 MC2	TP-3 FP4	TP2 PR	TP 1 LS
IN '01	TP-17H — BV	TP-17L —	TP-16H — CR	TP-16L —	TP-11 REL	TP-10 ZS	TP-9 LBC	TP-8 FLIN
IN '02	TP-24 B/P	TP-23 UPD	TP-22 A REL	TP-21 TACH 2	TP-19H — AP	TP-19L — X0 ≙ B X1 ≙ P	TP-18H — XP 01 ≙ P	TP-18L — 10 ≙ B
IN '03	TP-32 OS II	TP-31 OS I	TP-30 P SIG	TP-29 DCMD	TP-28 DSEN	TP-27 CB	TP-26 CA	TP-25 PCS
IN '04							TP-34 COLD	TP-33 BAL
IN '05	TP-20 BUSY	TP-20 BIT 8 (128)	TP-20 BIT 7 (64) TACH #1	TP20- BIT 6 (32) (NX .1) MPH	TP-20 BIT 5 (16)	TP-20 BIT 4 (8)	TP-20 BIT 3 (4)	TP-20 BIT 2 (2)
OUT '06	NOT USED	DISABLE TIME INTERRUPT	(#4) DUMP COMPLETE INDICATOR	(#5) ERROR INDICATOR	NOT USED	(#3) DUMPING INDICATOR	(#2) READY TO DUMP	(#1) MONITOR INDICATOR

PORT OF SEATTLE COMMISSION

DESIGN NO.

PROJECT STS - VDAS

TITLE BIT ASSIGNMENT IN PROLOG SYSTEM

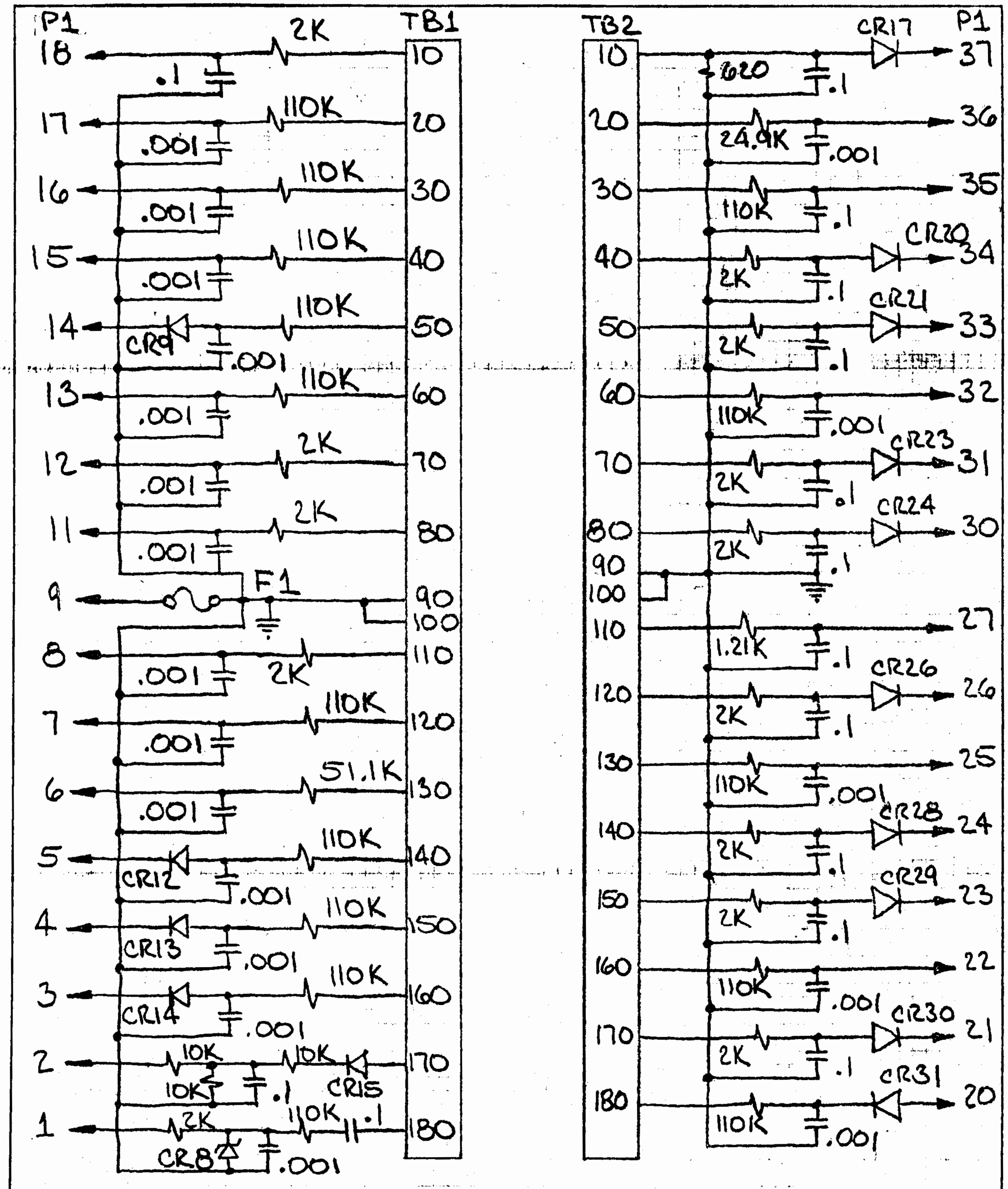
DRAWING NO.

DRAWN BY P. STUTZ DATE 2/24/78 APPROVED

POS 120-9

REVISED 3/14/78

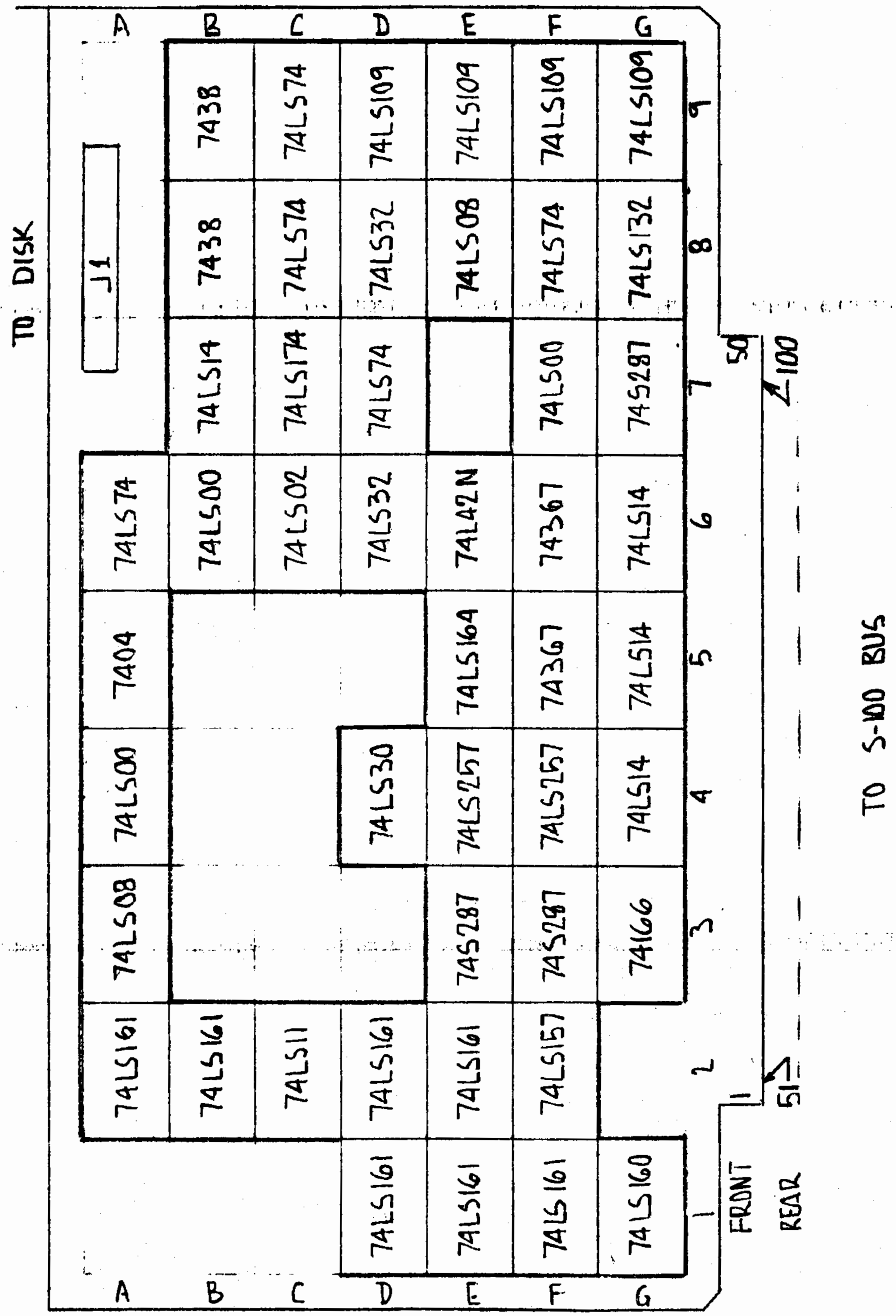
FIGURE 6



PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS- VDAS	
TITLE VDAS Termination Board	DRAWING NO.
DRAWN BY A. HAUG DATE 9/7/79 APPROVED	

POS 120-9

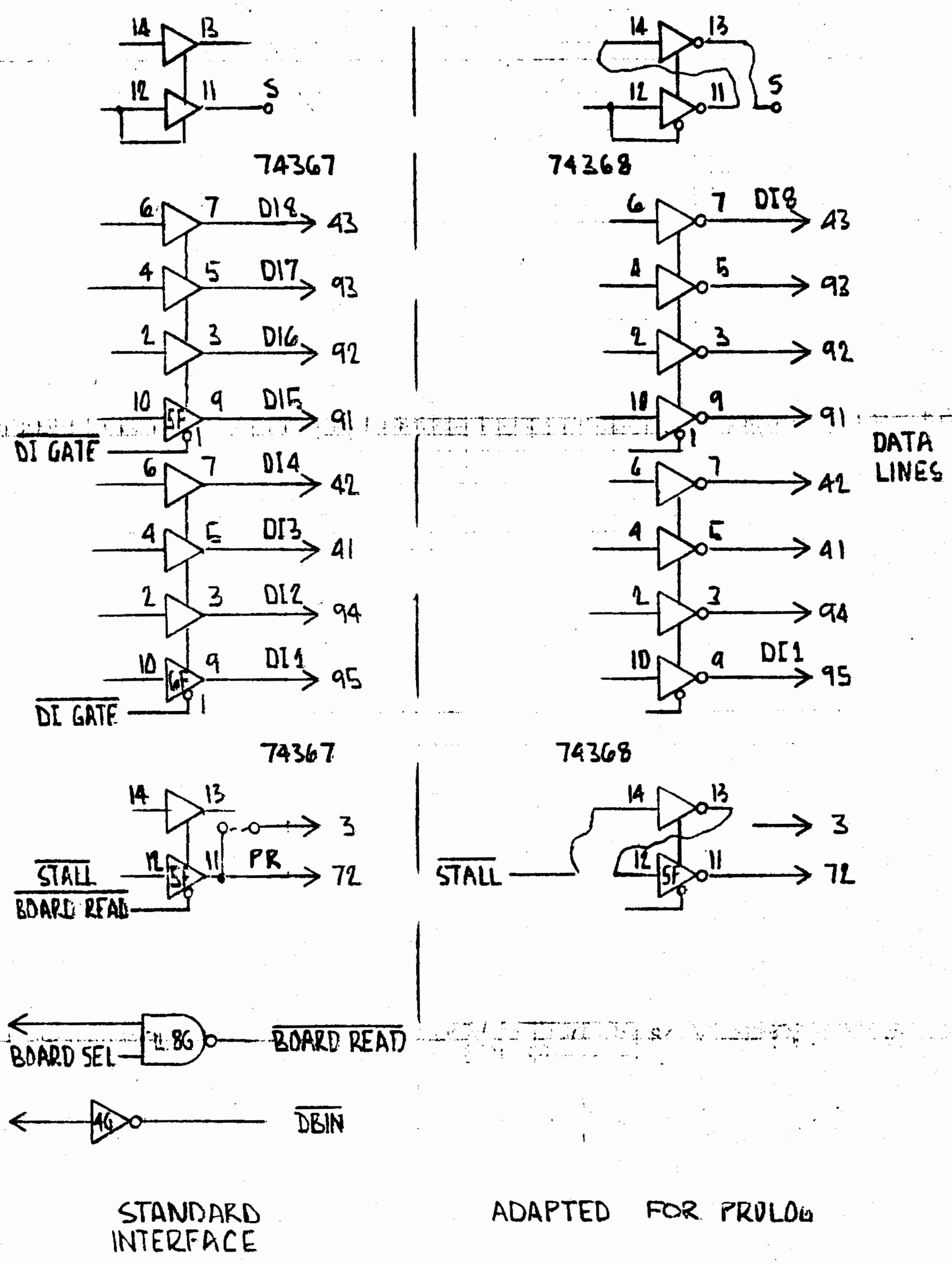
FIGURE 7



PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS- VDAS	
TITLE NORTHSTAR DISK I/O-CARD ASSEMBLY	DRAWING NO.
DRAWN BY P. STUTZ DATE 3/6/78 APPROVED	

POS 120-9

FIGURE 8 a



PORT OF SEATTLE COMMISSION

DESIGN NO.

DRAWING NO.

PROJECT STS - VDAS

TITLE ADAPT NORTHSTAR-DISK-I/O TO PROLOG

DRAWN BY PETER STUTZ DATE 3/6/78 APPROVED

POS 120-9

FIGURE 8b

M: Memory Amount (# of 8 bit words)

S: Sample rate(sec)

T: Length of time which may be stored using M & S above (min)

B: Word length (bits)

The following equation may be used to determine core memory size:

$$M = \frac{BT}{S} \times \frac{60\ \text{sec}}{1\ \text{min}} = \frac{60BT}{S}$$

In our system, the following are given:

$M = 56K = 57{,}344$ (1K = 1024)

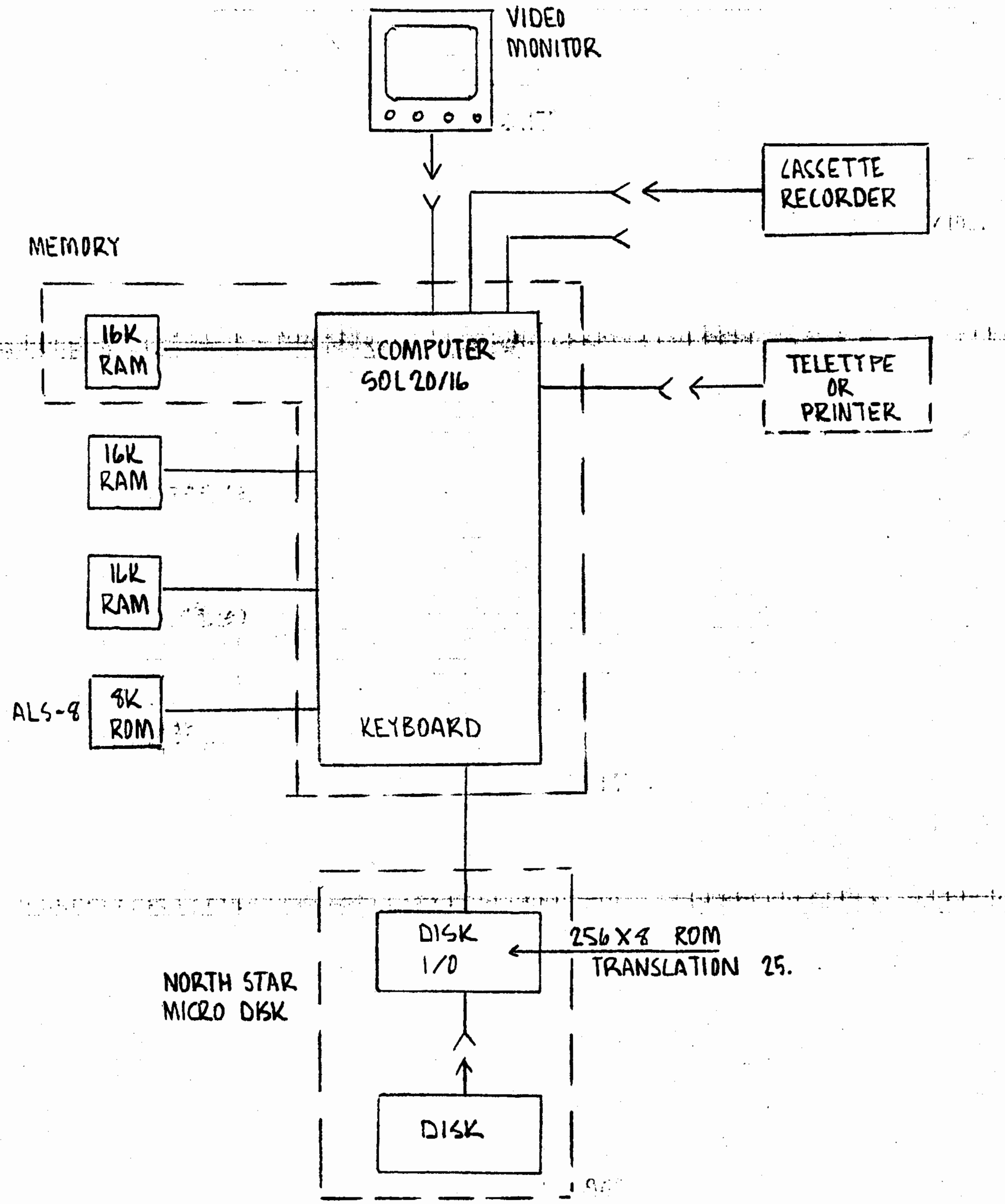
$S = .1\,\text{sec}$

$B = 8$ bits

$$T = \frac{MS}{60B} = \frac{(57344)(.1)}{60(8)} \qquad 11.95\ \text{min}$$

FIGURE 9

Core Memory Size Calculation



REVISED 11/29/77

PORT OF SEATTLE COMMISSION

PROJECT: STS - VDAG

TITLE: IN HOUSE DEVELOPMENT & PLOTTING SYSTEM

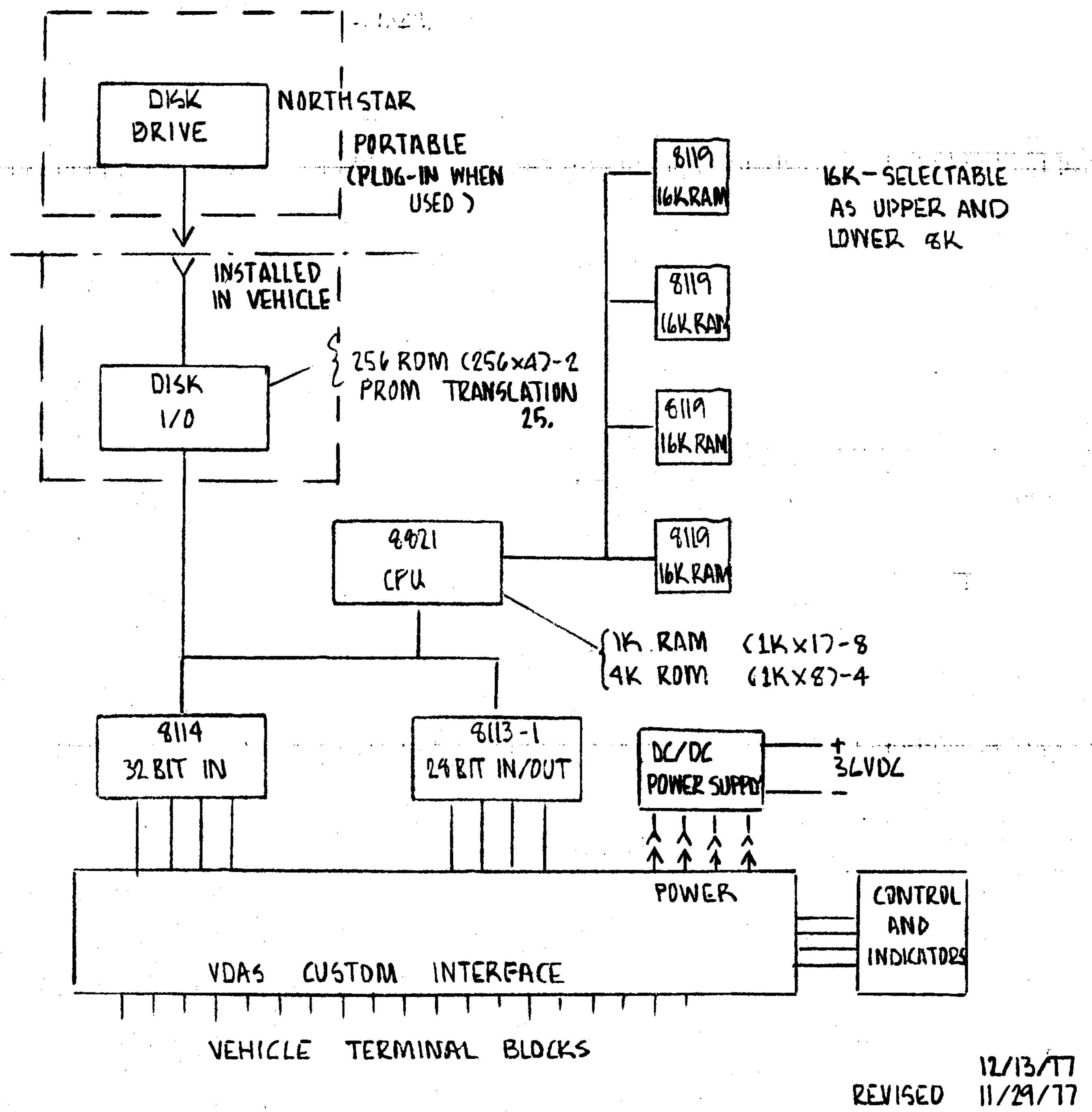
DRAWN BY: P. STUTZ DATE: 11-22-77 APPROVED:

DESIGN NO.

DRAWING NO.

POS 120-9

FIGURE 10



PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS- VDAS	
TITLE VEHICLE COMPUTER HARDWARE (PROLOG)	DRAWING NO.
DRAWN BY P. STUTZ DATE 11-22-77 APPROVED	

POS 120-9

FIGURE 11

DATA TIME	
1	LINE SW CLOSED
2	PR RELAY CLOSED
3	FP RELAY CLOSED
4	MAIN CONTACTOR CLOSED
5	MOTOR OVERLOAD
6	DOC TX TO WAYSIDE
7L	ANY DOC - VEH. OR INFO. RX
7H	BIT RATE + $\overline{\text{DOC}}$
8	FLIN + REL
9	LOW BRAKE CLAMP
10	ZERO SPEED
11	REL FROM INFO RX
16L	PROP
16H	HARD PROP
17L	PROGRAM STOP BRAKING
17H	MAX. CONTROL BRAKING
18L	PROP. (XP SIG.)
18H	BRAKING (XP SIG.)
19L	PROP. (AP SIG.)
19H	"
20	TACH 1 SPEED
21	TACH 2 > 2.5 VOLTS
22	A RELAY PULLED IN
23	UPDATE RECEIVED
24	PROP.
25	PCS TX ON
26	COUP. #1 END
27	COUP. #2 END
28	FWD SENSE FROM REVERSER
29	FWD + $\overline{\text{CAR AHEAD}}$
30	P SIG. INTERLOCK
31	TACH 1 FAULT
32	TACH 2 FAULT
33	BALANCE FAULT
34	CODE LOAD

FIGURE 12a

Data Analysis Overlay Showing Bit Placement

TEST DATA

(TP 1 THRU 34)

DATA PNT	TIME (SEC)	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
2570	5125.5	1	0	0	1	0	0	11	1	1	1	0	11	00	01	01	5.5	0	0	1	1	1	1	1	0	1	1	0	0	1	1
2571	5227.9	1	0	1	1	0	0	11	1	1	1	0	11	00	01	11	5.5	0	0	1	1	1	1	1	0	1	1	0	0	1	1
2572	5330.3	1	0	0	0	1	0	11	1	1	1	0	11	00	01	00	5.5	1	0	1	1	1	1	1	0	1	1	0	0	1	1
2573	5432.7	1	0	1	0	1	0	11	1	1	1	0	11	00	01	10	5.5	1	0	1	1	1	1	1	0	1	1	0	0	1	1
2574	5535.1	1	0	0	1	1	0	11	1	1	1	0	11	00	01	01	5.5	1	0	1	1	1	1	1	0	1	1	0	0	1	1
2575	5637.5	1	0	1	1	1	0	11	1	1	1	0	11	00	01	11	5.5	1	0	1	1	1	1	1	0	1	1	0	0	1	1
2576	5739.9	1	0	0	0	0	1	11	1	1	1	0	11	00	01	00	5.5	0	1	1	1	1	1	1	0	1	1	0	0	1	1
2577	5842.3	1	0	1	0	0	1	11	1	1	1	0	11	00	01	10	5.5	0	1	1	1	1	1	1	0	1	1	0	0	1	1
2578	5944.7	1	0	0	1	0	1	11	1	1	1	0	11	00	01	01	5.5	0	1	1	1	1	1	1	0	1	1	0	0	1	1
2579	6047.1	1	0	1	1	0	1	11	1	1	1	0	11	00	01	11	5.5	0	1	1	1	1	1	1	0	1	1	0	0	1	1
2580	6149.5	1	0	0	0	1	1	11	1	1	1	0	11	00	01	00	5.5	1	1	1	1	1	1	1	0	1	1	0	0	1	1
2581	6251.9	1	0	1	0	1	1	11	1	1	1	0	11	00	01	10	5.5	1	1	1	1	1	1	1	0	1	1	0	0	1	1
2582	6354.3	1	0	0	1	1	1	11	1	1	1	0	11	00	01	01	5.5	1	1	1	1	1	1	1	0	1	1	0	0	1	1
2583	6456.7	1	0	1	1	1	1	11	1	1	1	0	11	00	01	11	5.5	1	1	1	1	1	1	1	0	1	1	0	0	1	1

STOP IN LINE 9135LEGEND

1-FAULT

0-OK

B-BRAKE

P-PROPULSION

U(OR X)-UNDEFINED

TEST POINT 20 GIVEN IN MPH

FIGURE 12b

Sample Test Data Analysis Output

Sub-system	# TP's
Car Logic	1
Traction Controller	3
Brake/Propulsion	4
Speed Maintaining	7
Speed Decoder	1
Information Receiver	3
Vehicle Transmitter	1
Train Line Wire	5
AC Power Controls	5
	30

Four test points not shown deal with the vehicle doors. These points were later determined to be so obvious as to not require system monitoring (ie. doors are either open or closed).

TABLE 1

APPENDIX A

Test Point Drawings

VDAS

BASIC GUIDELINES FOR INTERCONNECTION

- ALL 36VDC SIGNAL THRU OPTICAL ISOLATORS

- ~~NO PHYSICAL CONNECTION BETWEEN BATTERY GROUND AND SYSTEM GROUND~~

EXCLUDE
12/13/77 P. STUTZ

- ALL INTERCONNECTIONS ON TERMINAL BLOCKS, EITHER ALREADY EXISTING, SPARE TERMINALS OR TB-3 ~~(TOTALLY UNUSED SO FAR).~~ TB3-55/56/57/58 USED FOR RECENT MODIFICATION

- USE SPARE CIRCUITRY ON BOARDS WHERE AVAILABLE (UNUSED BUFFERS ETC.)

- NO CHANGE TO EXISTING CIRCUIT BOARDS UNLESS ABSOLUTELY NO OTHER CHOICE.

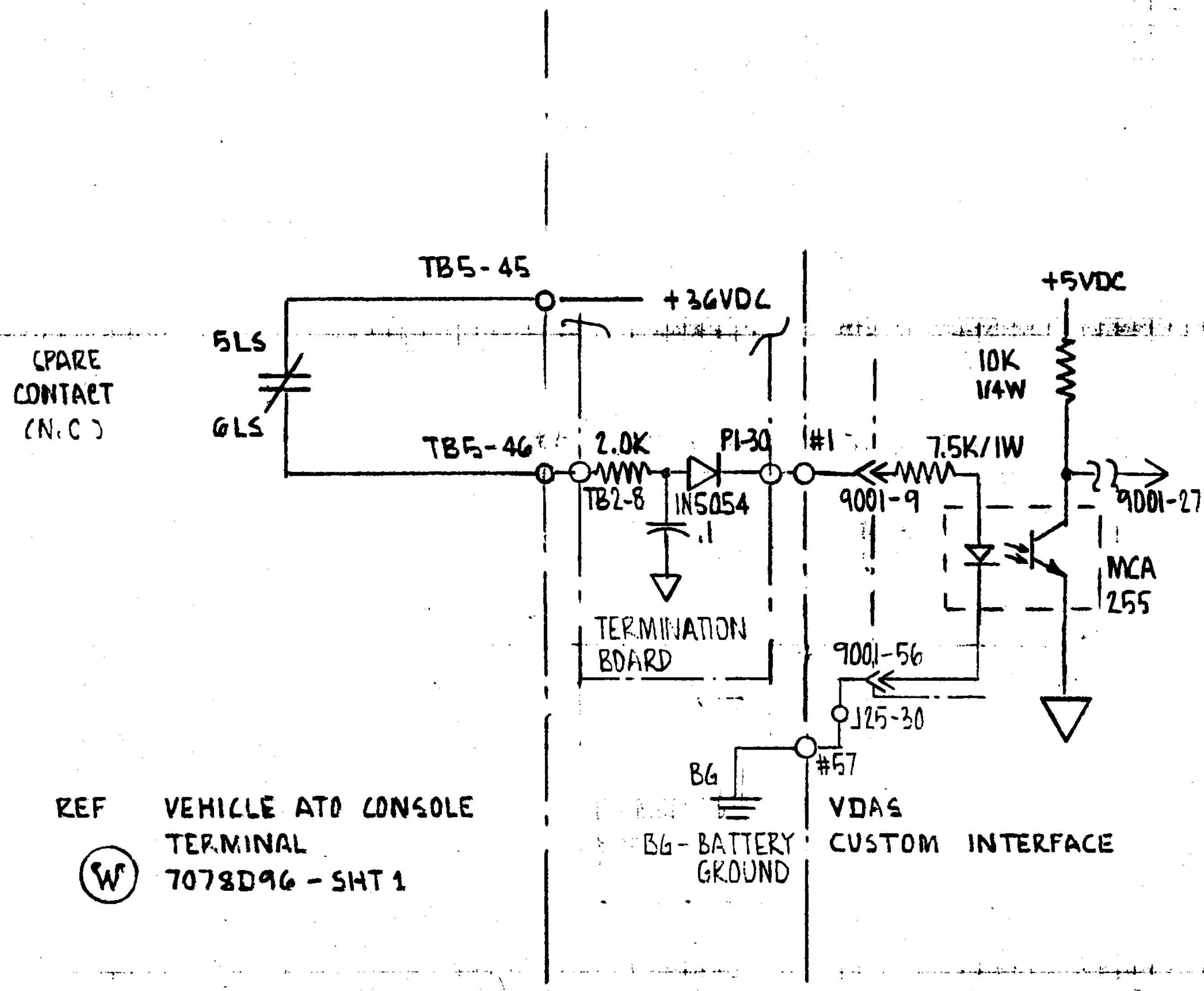
- VDAS CRADLE, 104-PIN CONNECTOR NUMBERS ARE DESIGNATED WITH "#" AND PIN NUMBER. B

REV C	7/20/78
REV B	4/3/78
REV A	12/13/77

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE GUIDELINES FOR INTERCONNECT	DRAWING NO.
DRAWN BY P. STUTZ DATE 11-4-77 APPROVED	LCD P-330

POS 120-9

SHT 1 (32)

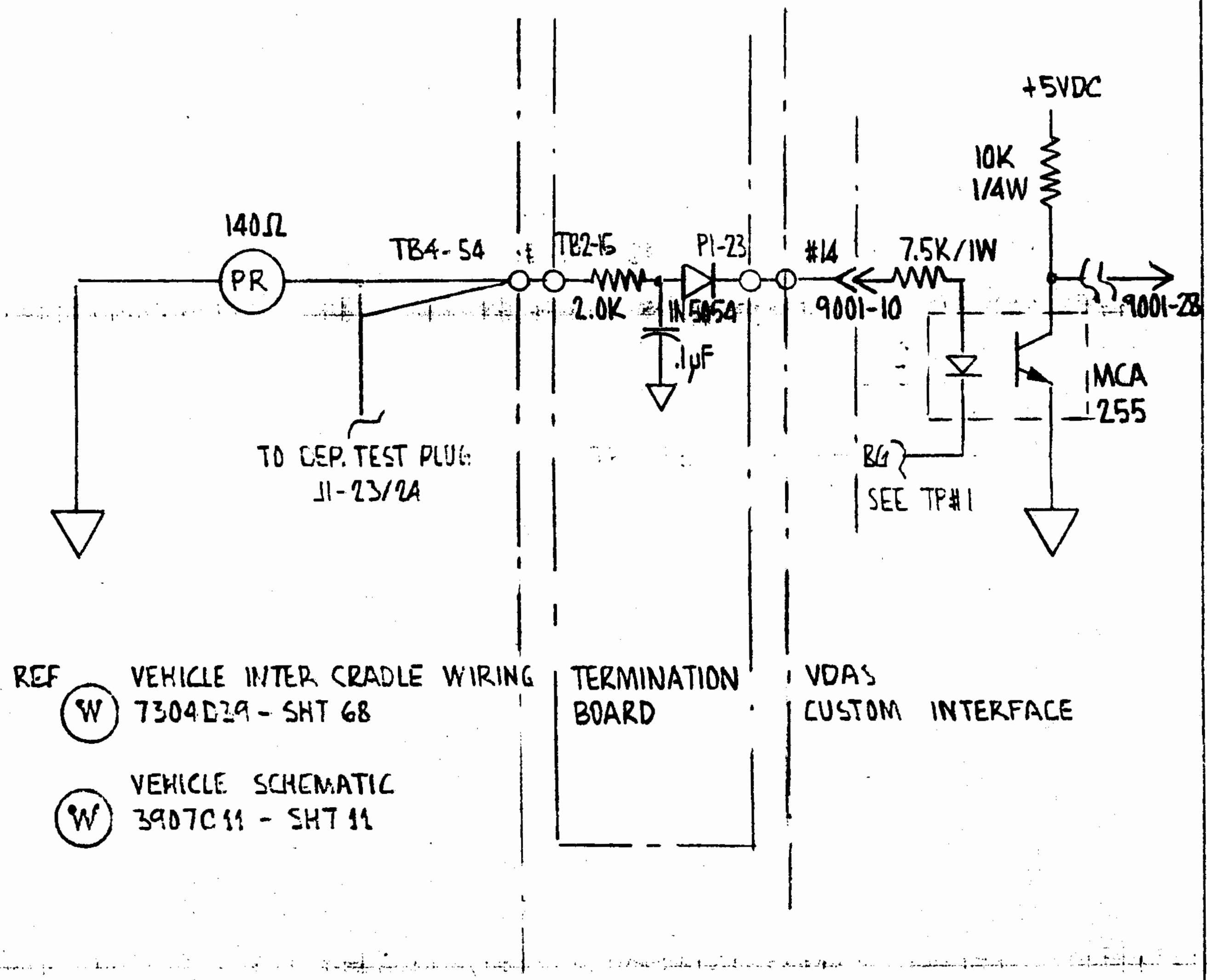


REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #1 - LINE SWITCH	DRAWING NO.
DRAWN BY P. STUTZ DATE 10-31-77 APPROVED	LCO P-330

POS 120-9

SHT 2 (32)

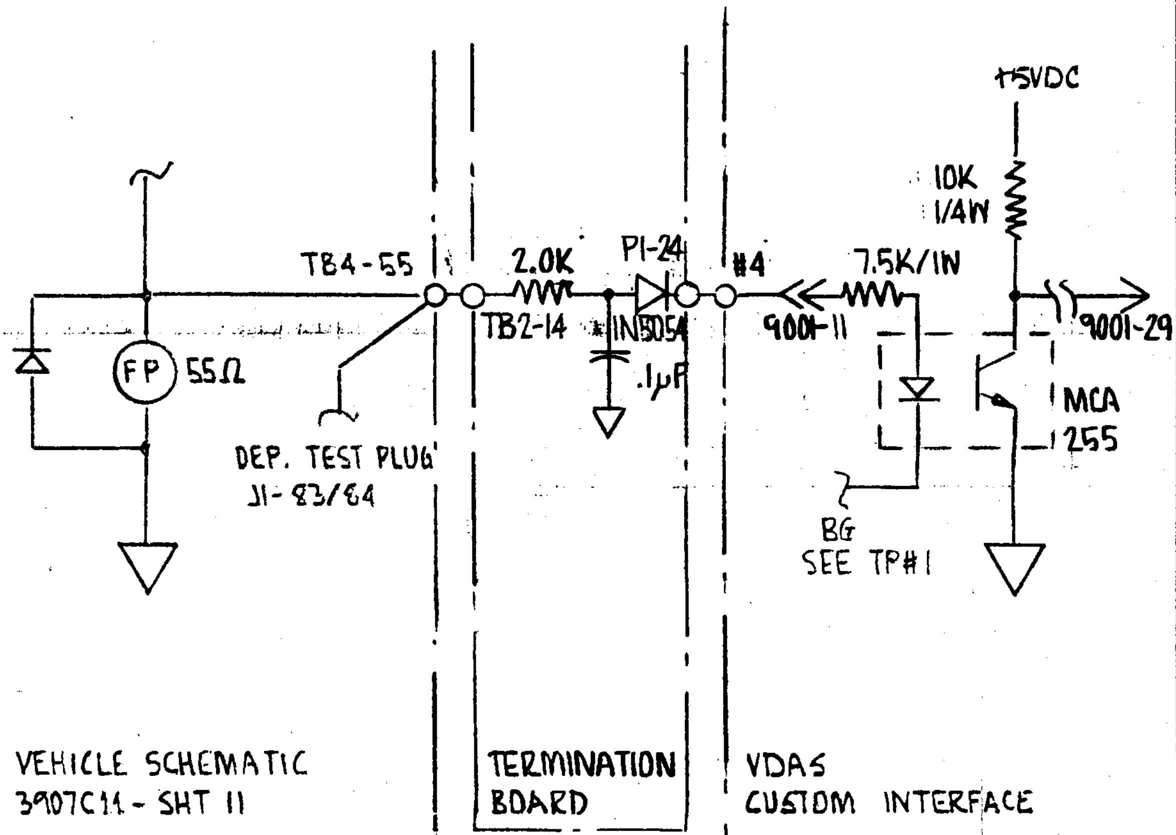


REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #2 - POWER RELAY	DRAWING NO.
DRAWN BY P. STUTZ DATE 10-31-77 APPROVED	LCO P-330

POS 120-9

SHT 3 (32)



REF.

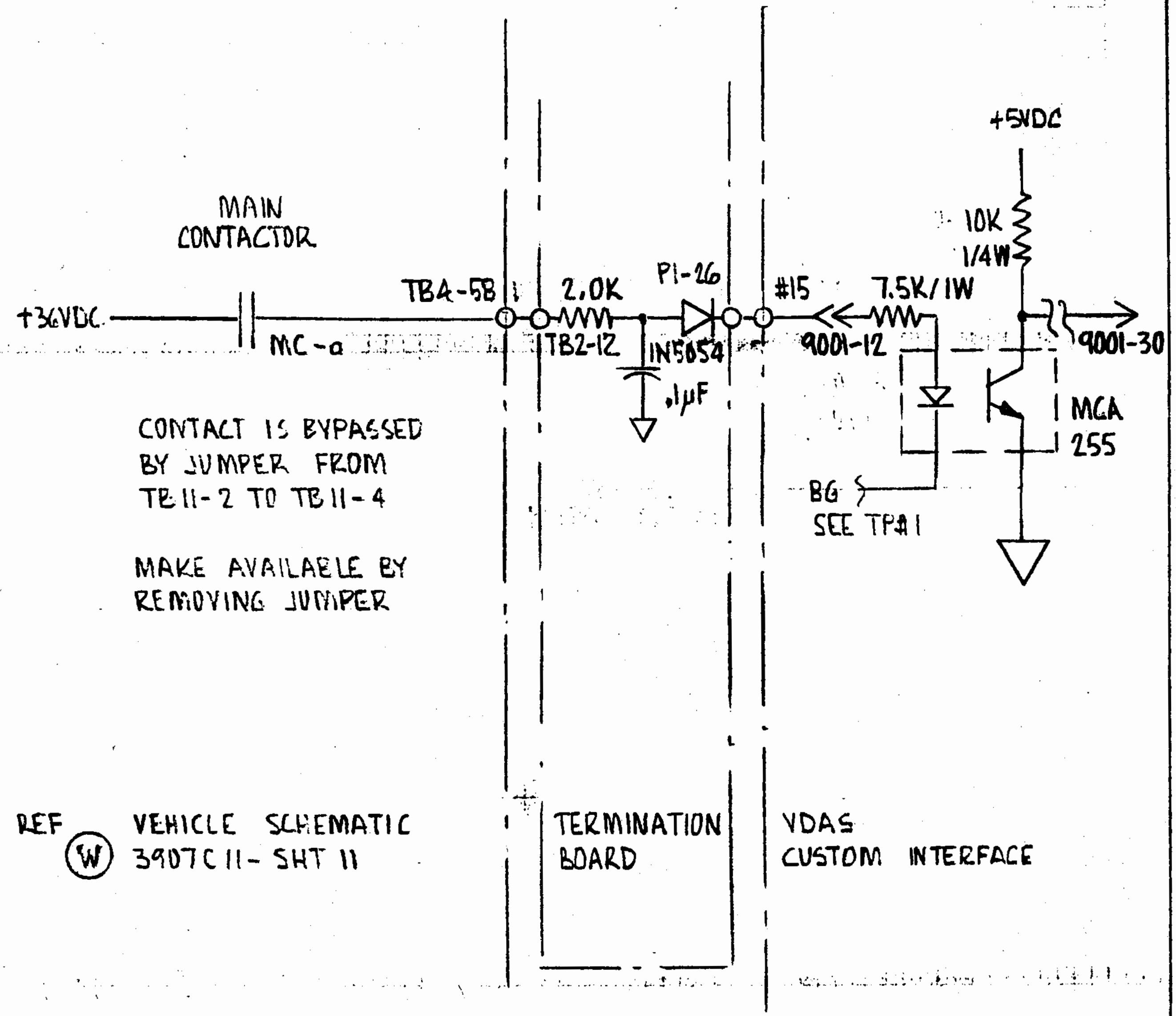
(W) VEHICLE SCHEMATIC 3907C11 - SHT 11

(W) VEH. ATD CONSOLE 7078D96 - SHT 1

(W) INTER-CRADLE WIRING 7304D39 - SHT 69

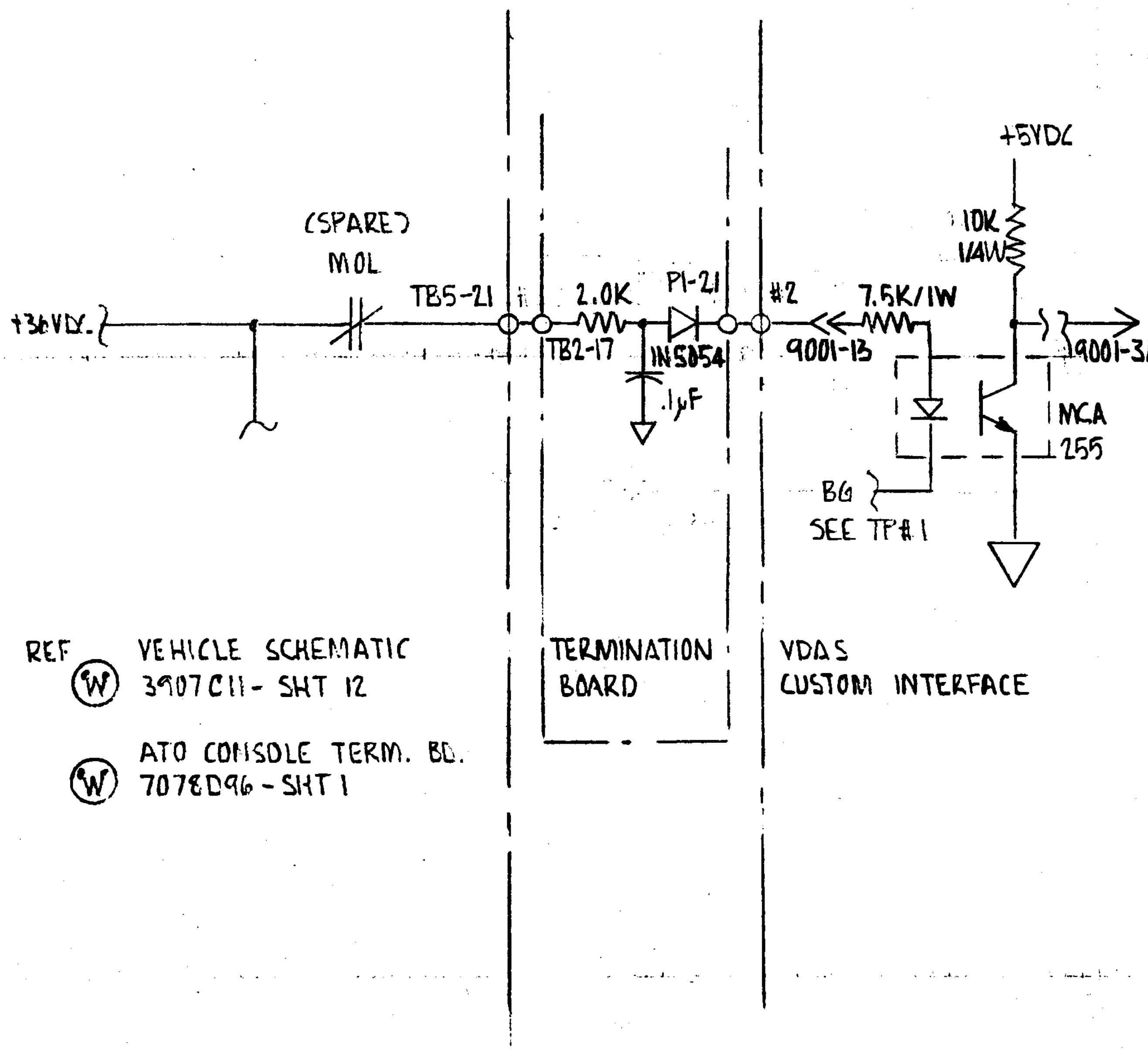
REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #3 - GATING RELAY (FP 4)	DRAWING NO. LCO P-330
DRAWN BY P. STUTZ DATE 10-31-77 APPROVED	



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS-VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT # 4 – MAIN CONTACTOR	DRAWING NO.
DRAWN BY P. STUTZ DATE 11-31-77 APPROVED	LCD P-330



REF VEHICLE SCHEMATIC
W 3907C11 - SHT 12

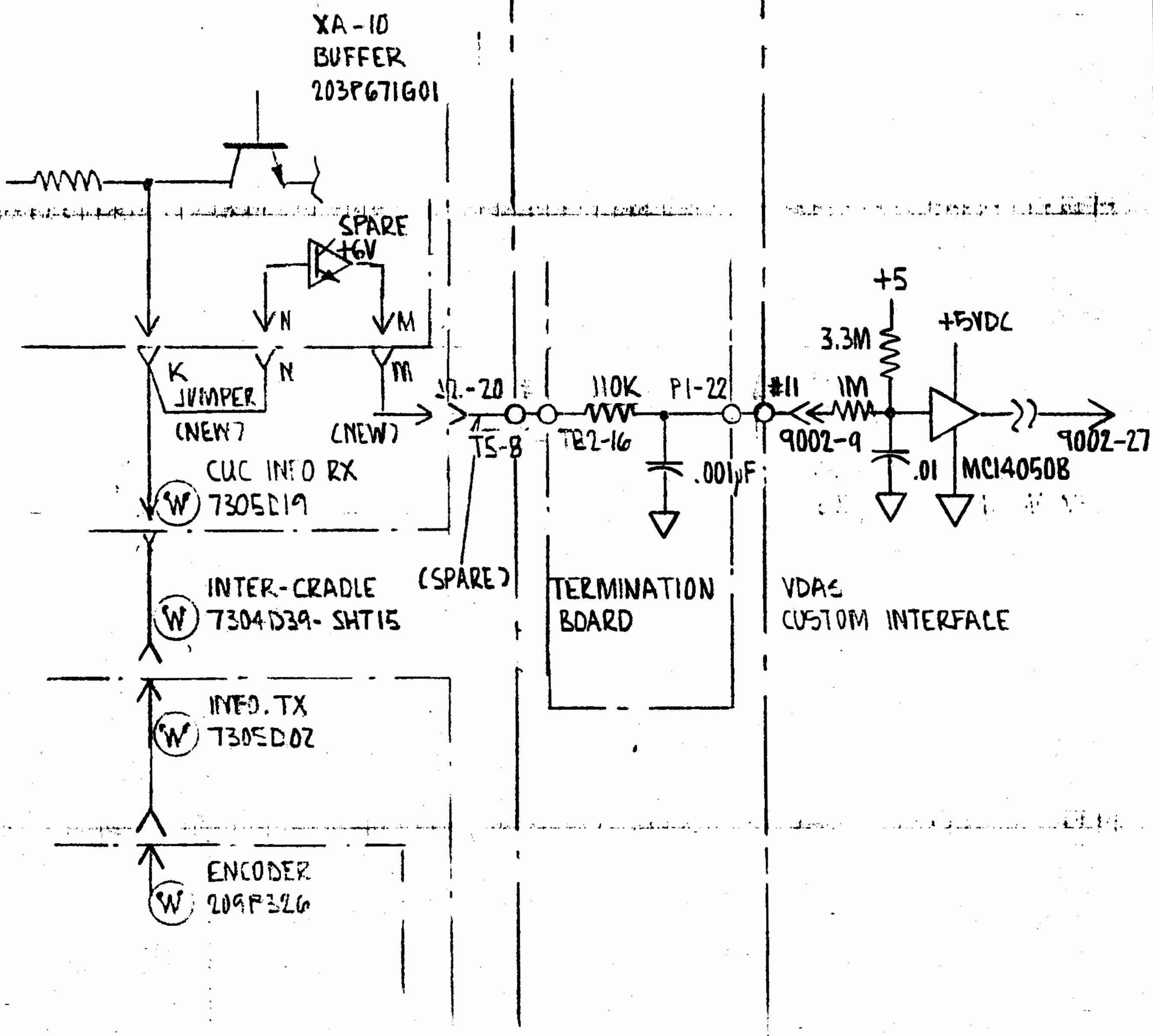
ATO CONSOLE TERM. BD.
W 7078D96 - SHT 1

REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #5 - MOTOR OVERLOAD	DRAWING NO.
DRAWN BY P. STUTZ DATE 10-31-77 APPROVED	LCO P-330

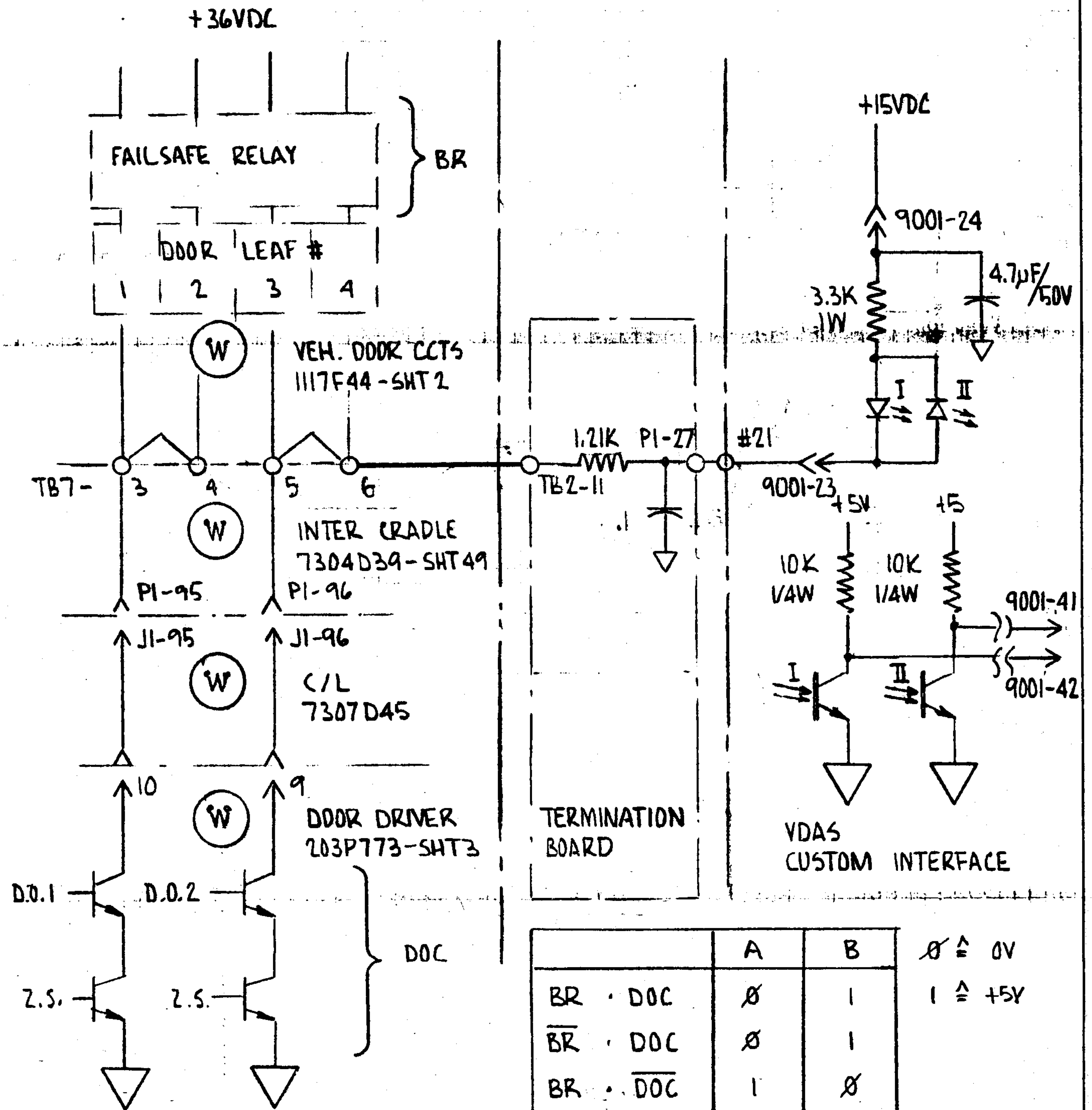
POS 120-9

SHT 6 (32)



REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #6 - DOOR OPEN COMMAND TX (FROM VEHICLE)	DRAWING NO.
DRAWN BY P.STUTZ DATE 10-31-77 APPROVED	LCO P-330



	A	B
BR · DOC	Ø	1
$\overline{BR}$ · DOC	Ø	1
BR · $\overline{DOC}$	1	Ø
$\overline{BR}$ · $\overline{DOC}$	1	1

Ø ≙ 0V
1 ≙ +5V

REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

PROJECT STS- VDAS TESTPOINT DEFINITION

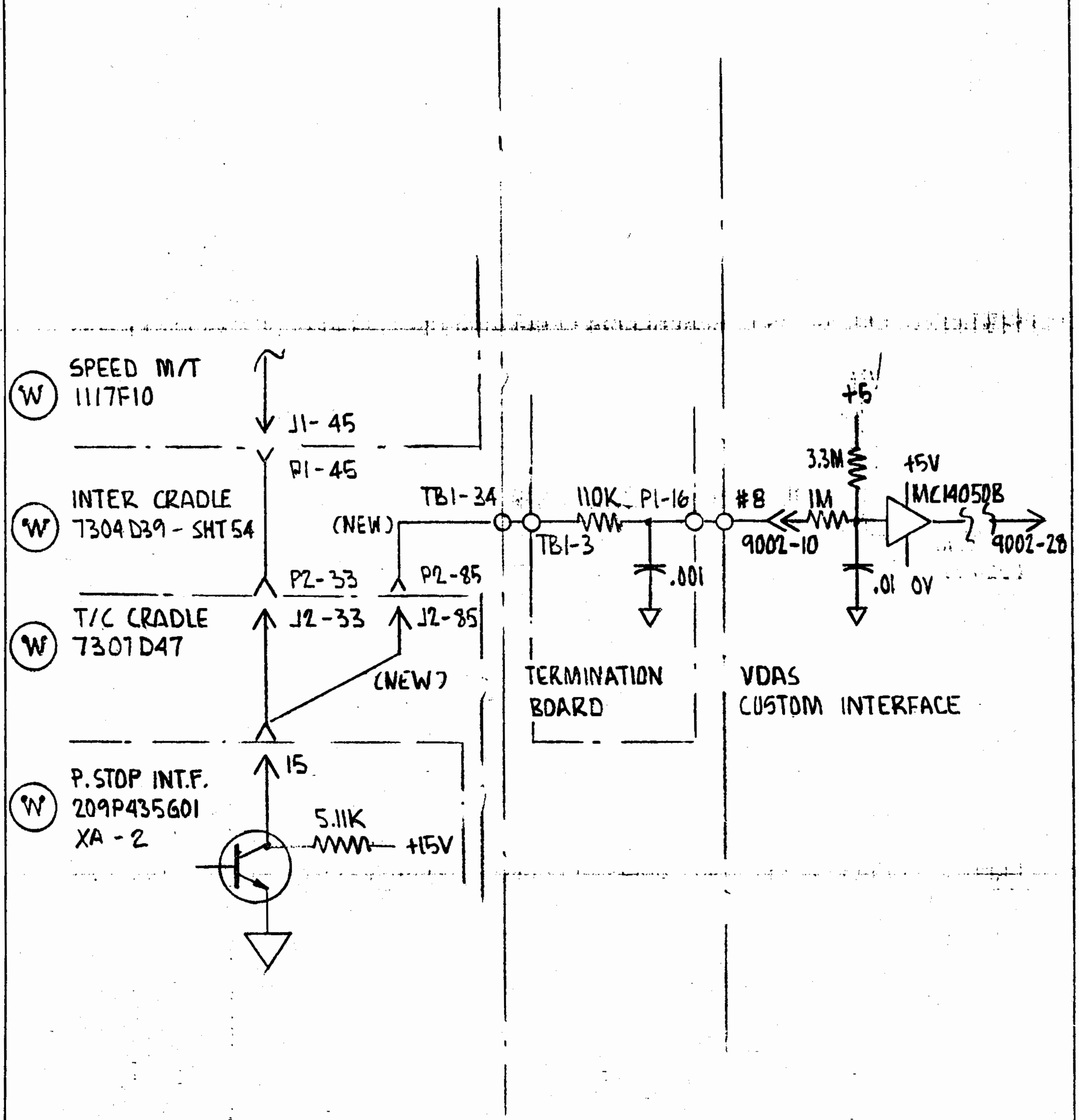
TITLE TESTPOINT # 7 – DOOR OPEN COMMAND (RECEIVE)

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9



REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

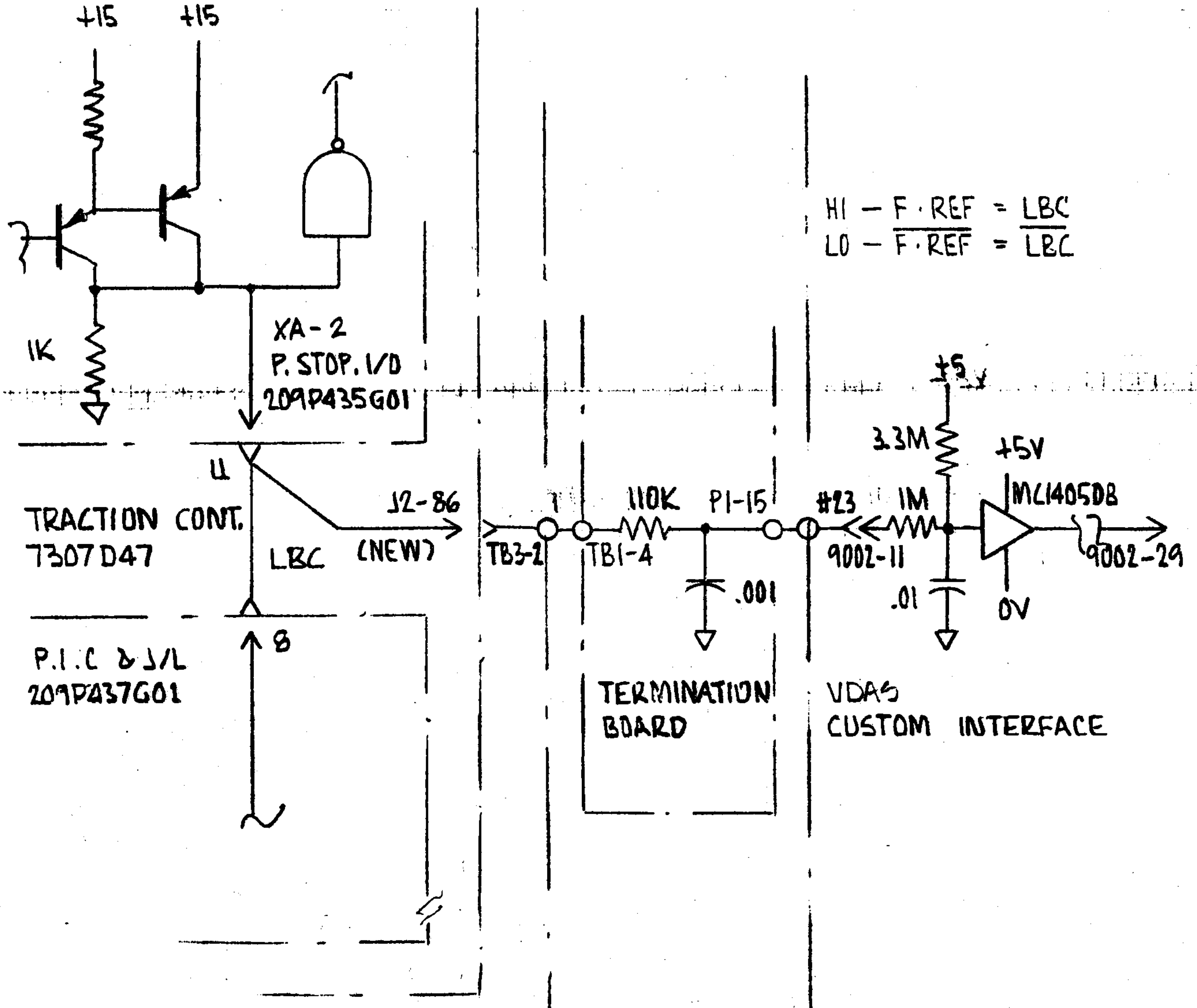
TITLE TESTPOINT #4 - FLIN • $\overline{\text{REL}}$

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9



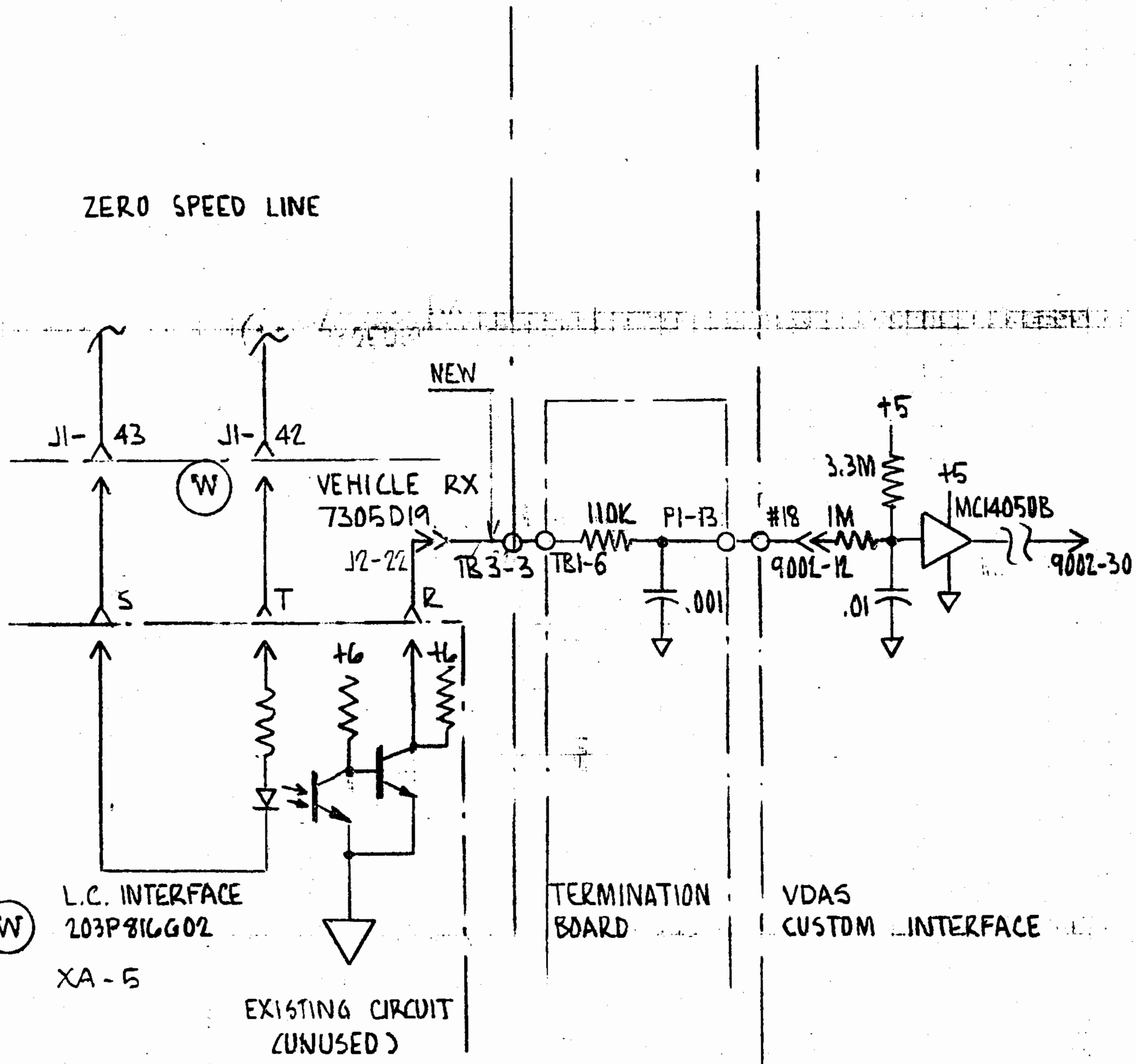
REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT # 9 - LBC	DRAWING NO.
DRAWN BY P. STUTZ DATE 11-1-77 APPROVED	LCO P-330

POS 120-9

SHT 10 (32)

ZERO SPEED LINE

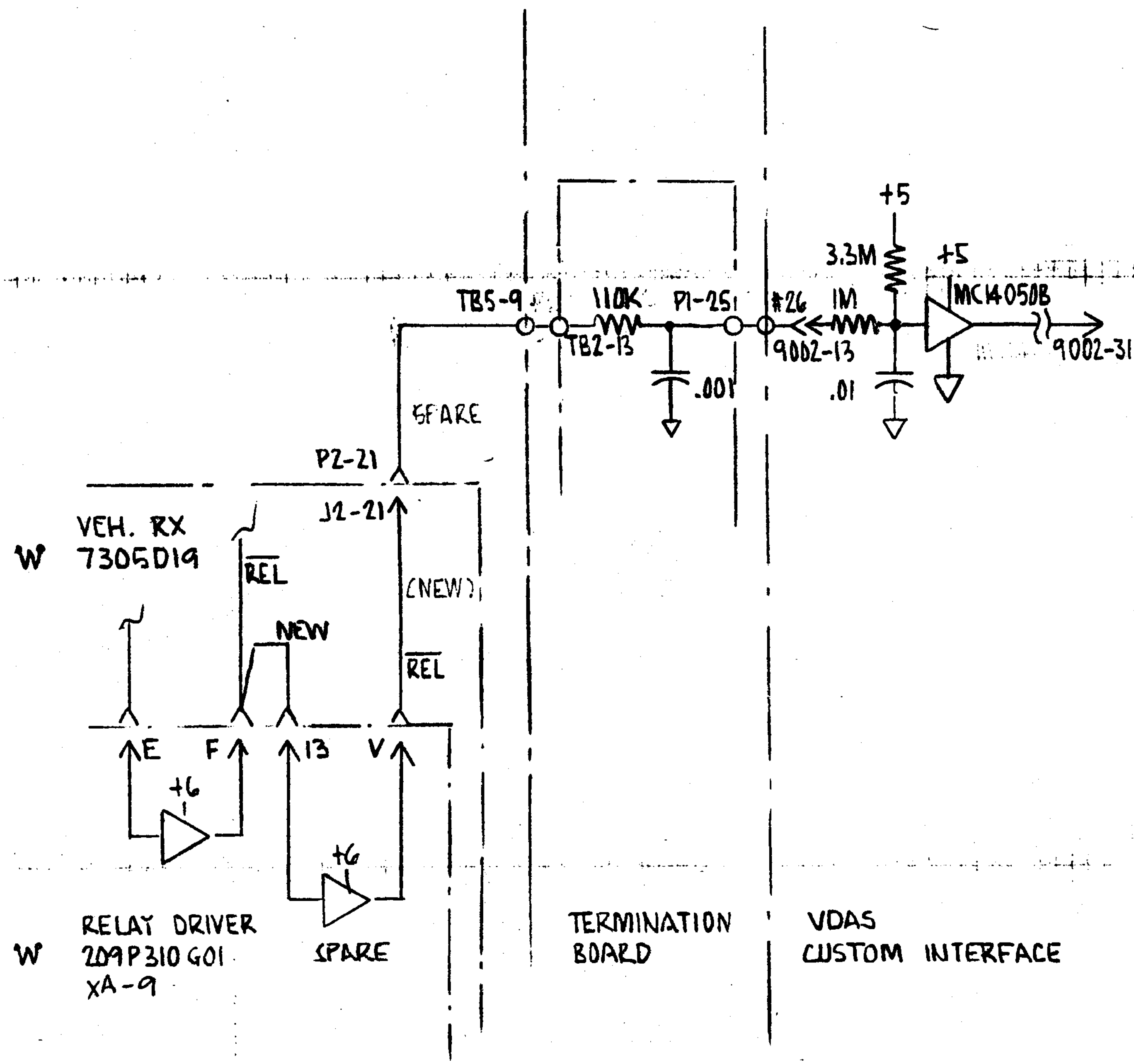


REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS VDAS TESTPOINT DEFINITION	
TITLE TESTPOINT #10 - ZERO SPEED	DRAWING NO.
DRAWN BY P. STUTZ DATE 11-1-77 APPROVED	LCO P-330

POS 120-9

SHT 11 (32)



REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

TITLE TESTPOINT # 11 - REL

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

DESIGN NO.

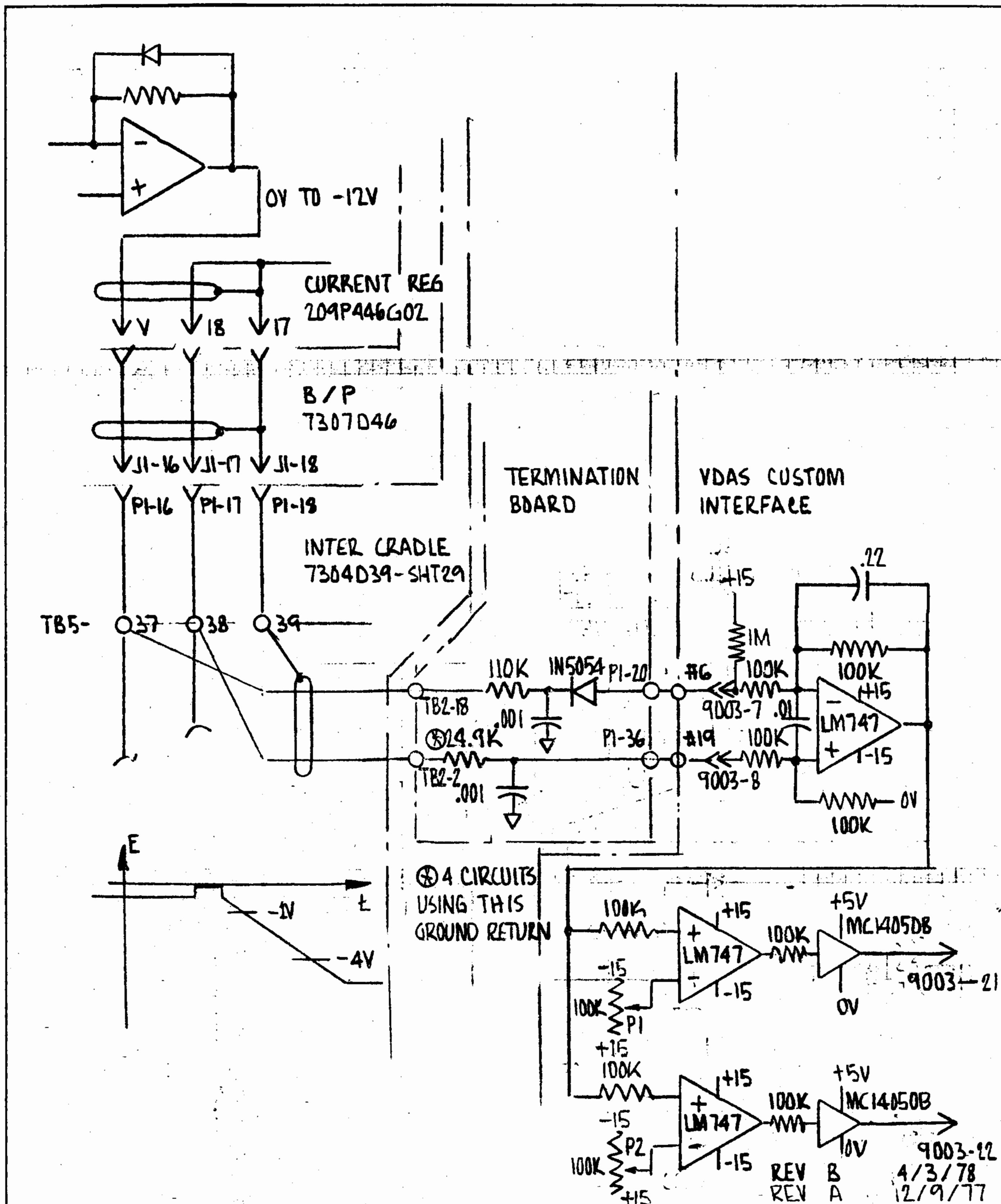
DRAWING NO. LCO P-330

USE TB3-3 DOOR LEAF #1
TB3-4 DOOR LEAF #2
TB3-5 DOOR LEAF #3
TB3-6 DOOR LEAF #4

REV C 7/20/78
REV B 4/3/78
WILL NOT BE USED
IN VDAS PER
MIKE KISMAN
REV A 12/9/77

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - VDAS TESTPOINT DEFINITION	
TITLE TESTPOINTS 12/13/14/15 - DOOR LEAF OPEN LIMIT SWITCH	DRAWING NO.
DRAWN BY P. STUTZ DATE 11/8/77 APPROVED	LCO P-330

POS 120-9



PORT OF SEATTLE COMMISSION REV C 7/20/78

DESIGN NO.

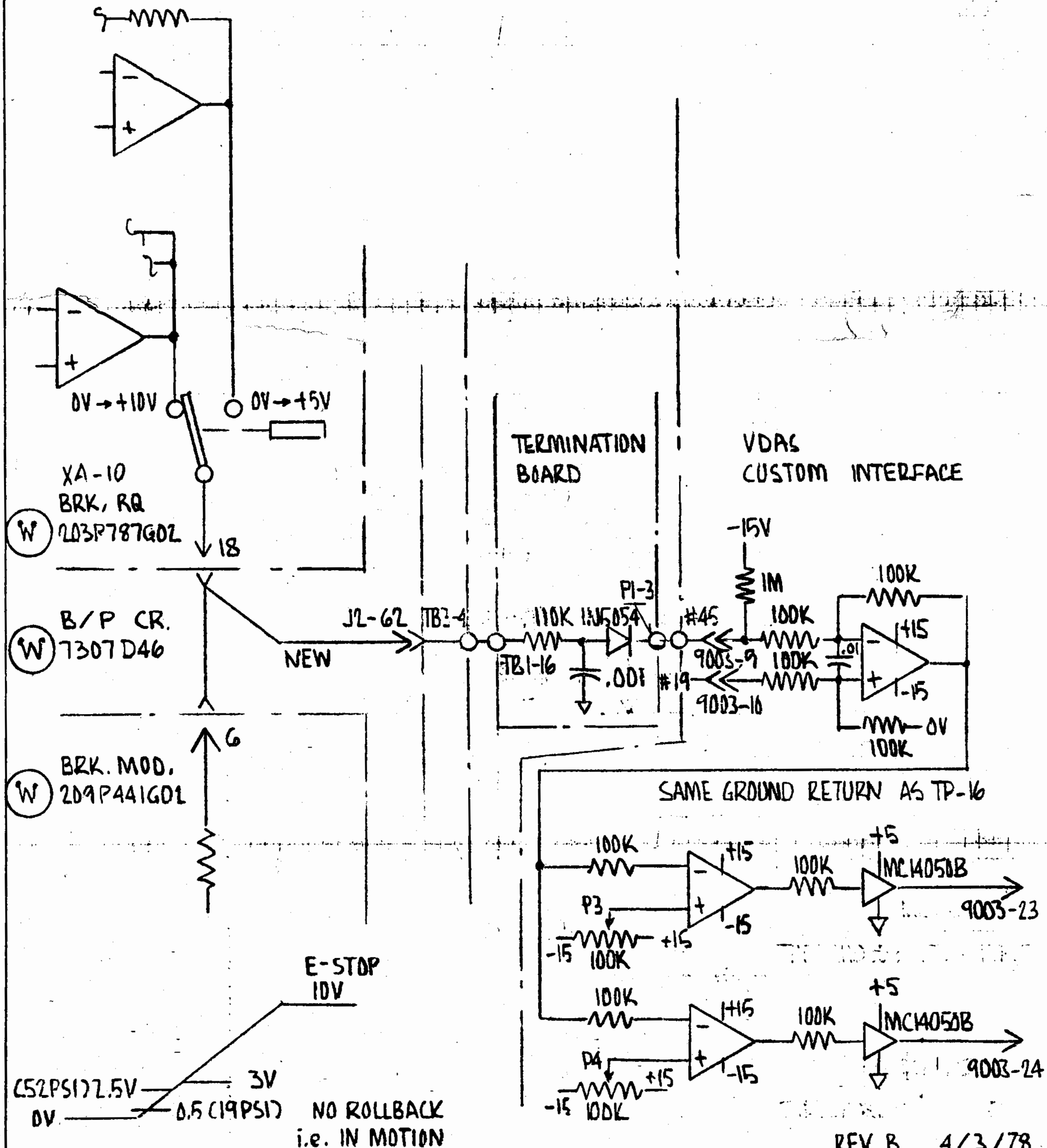
PROJECT STS - VDAS TESTPOINT DEFINITION

TITLE TESTPOINT # 16 - CR (FIRING PULSE)

DRAWING NO. LCO P-330

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

POS 120-9



REV B 4/3/78

REV C 7/20/78 REV A 12/9/77

PORT OF SEATTLE COMMISSION

DESIGN NO.

PROJECT STS - VDAS TESTPOINT DEFINITION

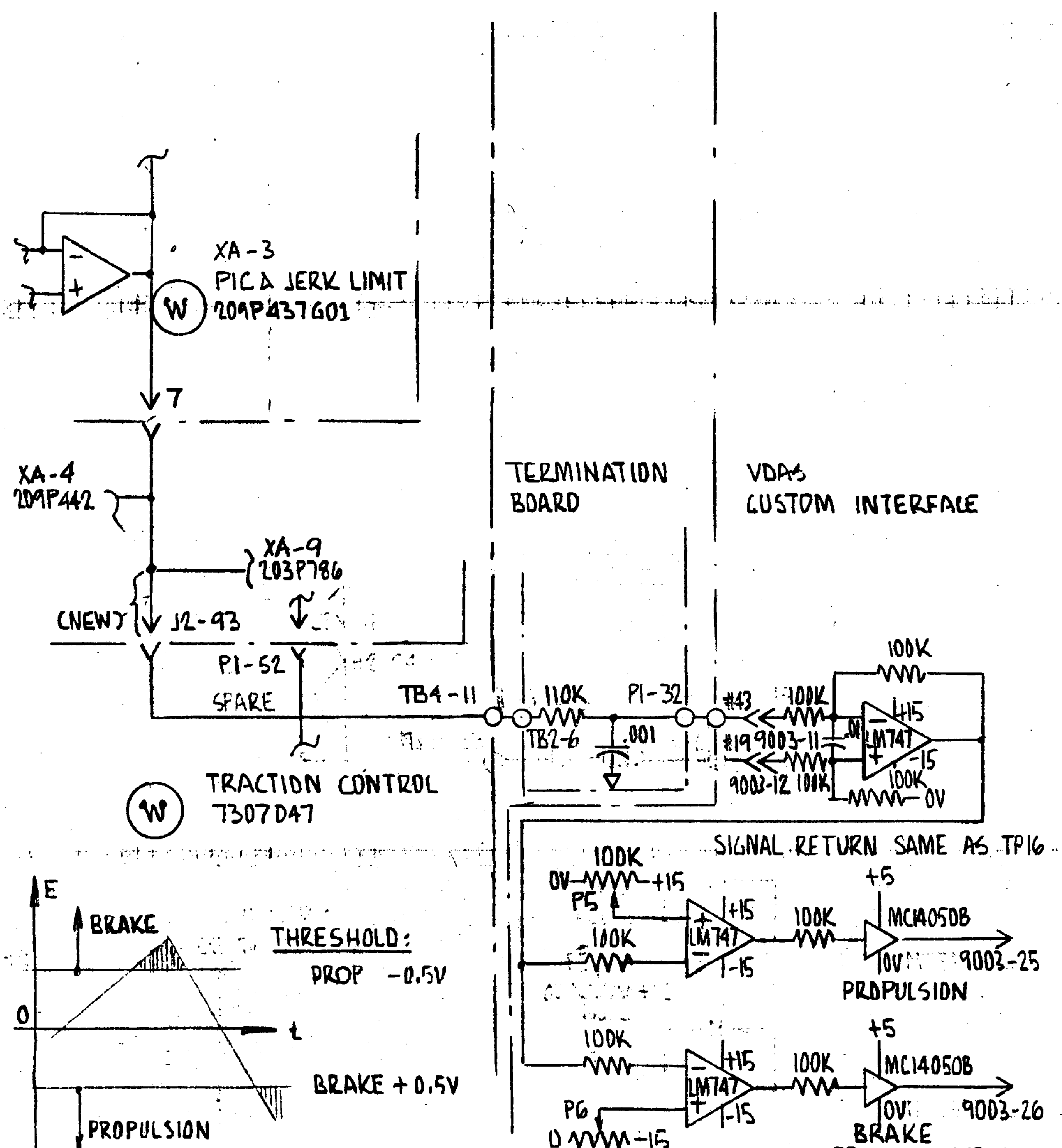
TITLE TESTPOINT #17 - BV

DRAWING NO. LCO P-330

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

POS 120-9

SHT 15 (32)



REV B 4/3/78
REV C 7/20/78 REV A 12/19/77

PORT OF SEATTLE COMMISSION

PROJECT STS-VDAS TESTPOINT DEFINITION

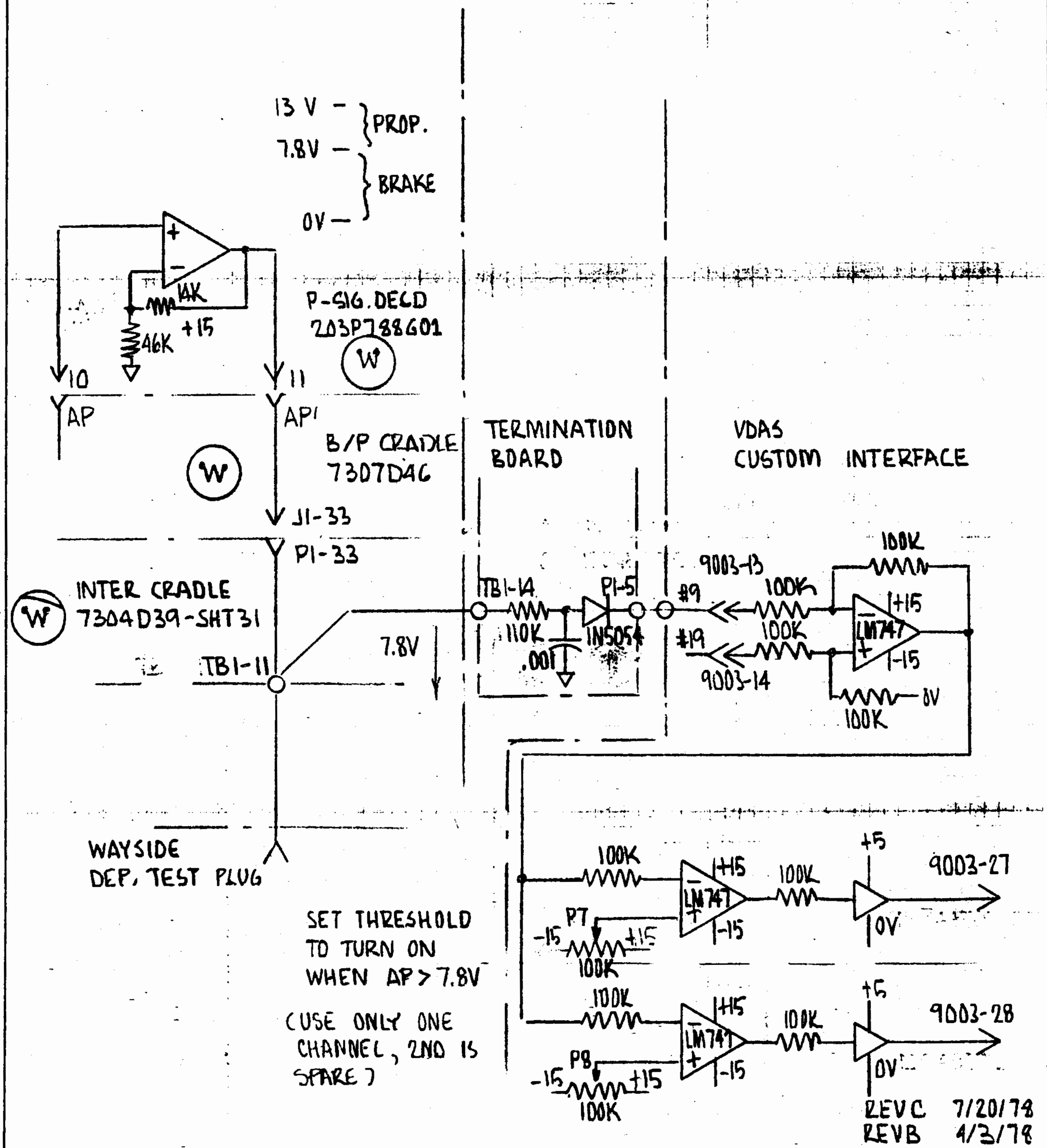
TITLE TESTPOINT #18 - X-P SIGNAL

DRAWN BY P. STUTZ DATE 11-1-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9



PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #19 - AP SIGNAL

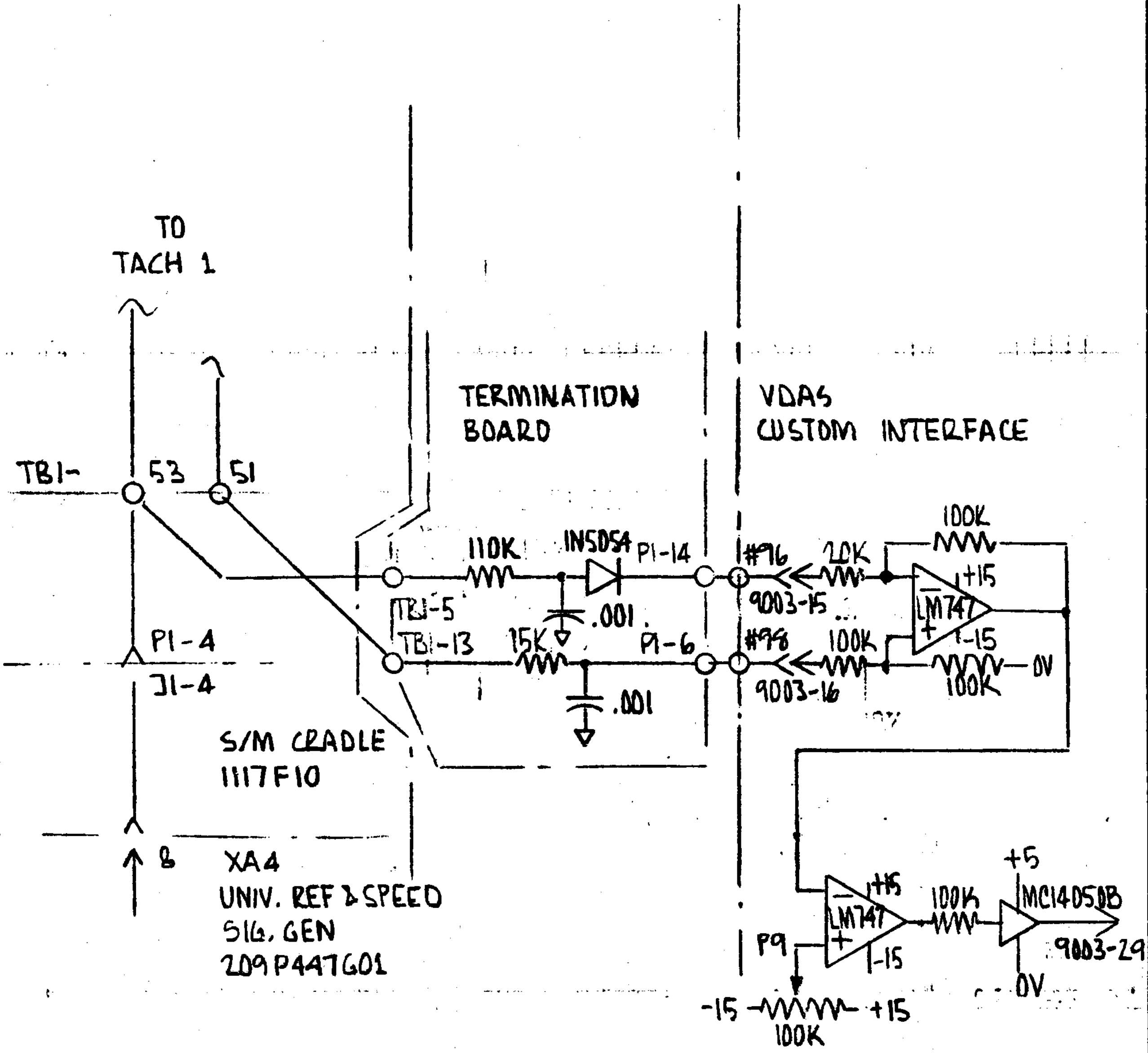
DRAWN BY P. STUTZ DATE 11-2-77 APPROVED

DESIGN NO.

DRAWING NO. LC0 - P330

POS 120-9

SHT 17 (32)



INFO ONLY

ATO TACH DETECTOR
THRESHOLD 2.1V

102 HZ / MPH

THRESHOLD SET
AT 2.5V PEAK
AT TB1-53

GATE COUNTER AND
ENTER COUNT AS
8-BIT NUMBER INTO
COMPUTER
ONE-SHOT CIRCUIT FOR
SIGNAL CONDITIONING

REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #20 - TACH #1

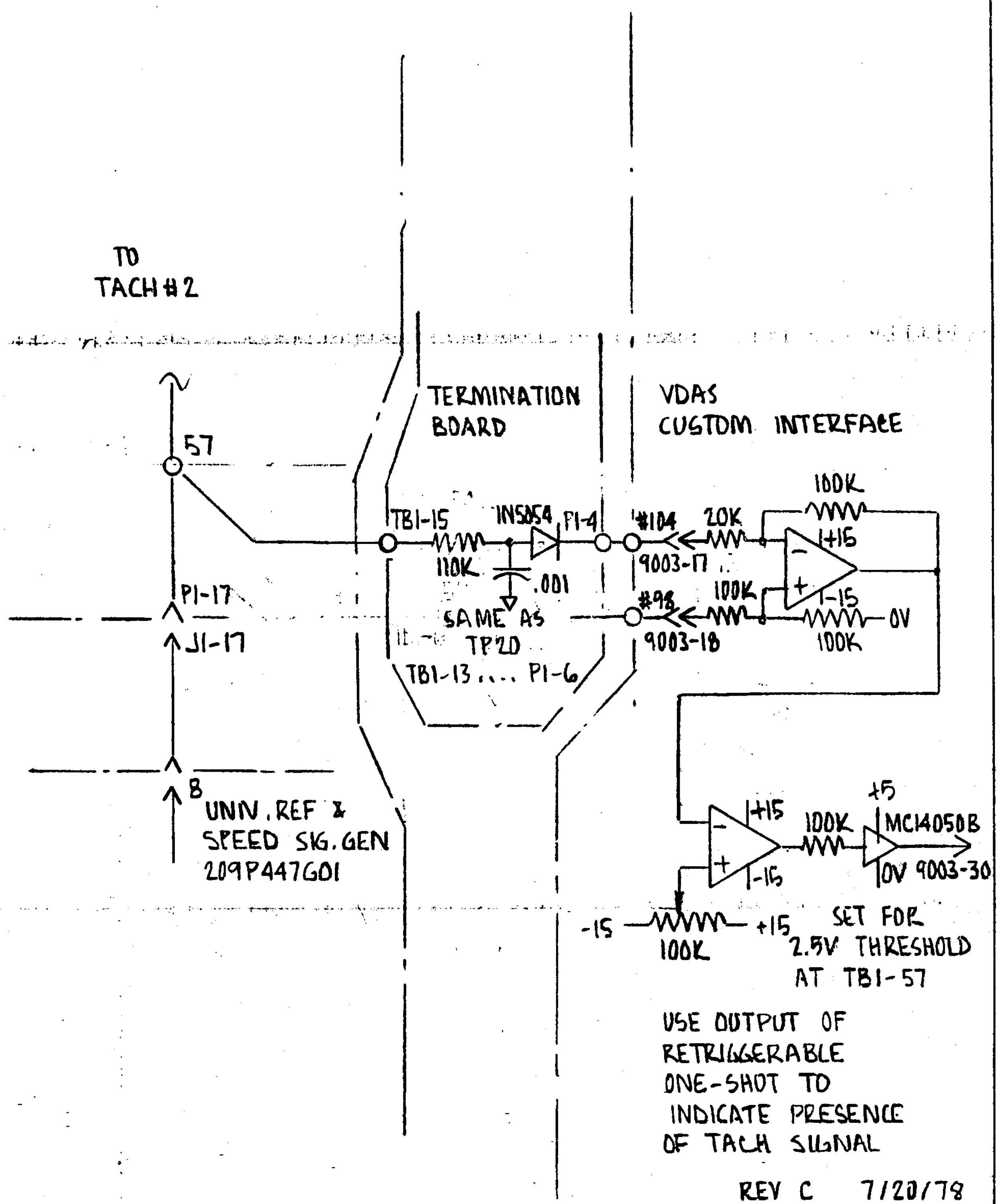
DRAWN BY P.STUTZ DATE 11-2-73 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9

SHT 18 (32)



USE OUTPUT OF
RETRIGGERABLE
ONE-SHOT TO
INDICATE PRESENCE
OF TACH SIGNAL

REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

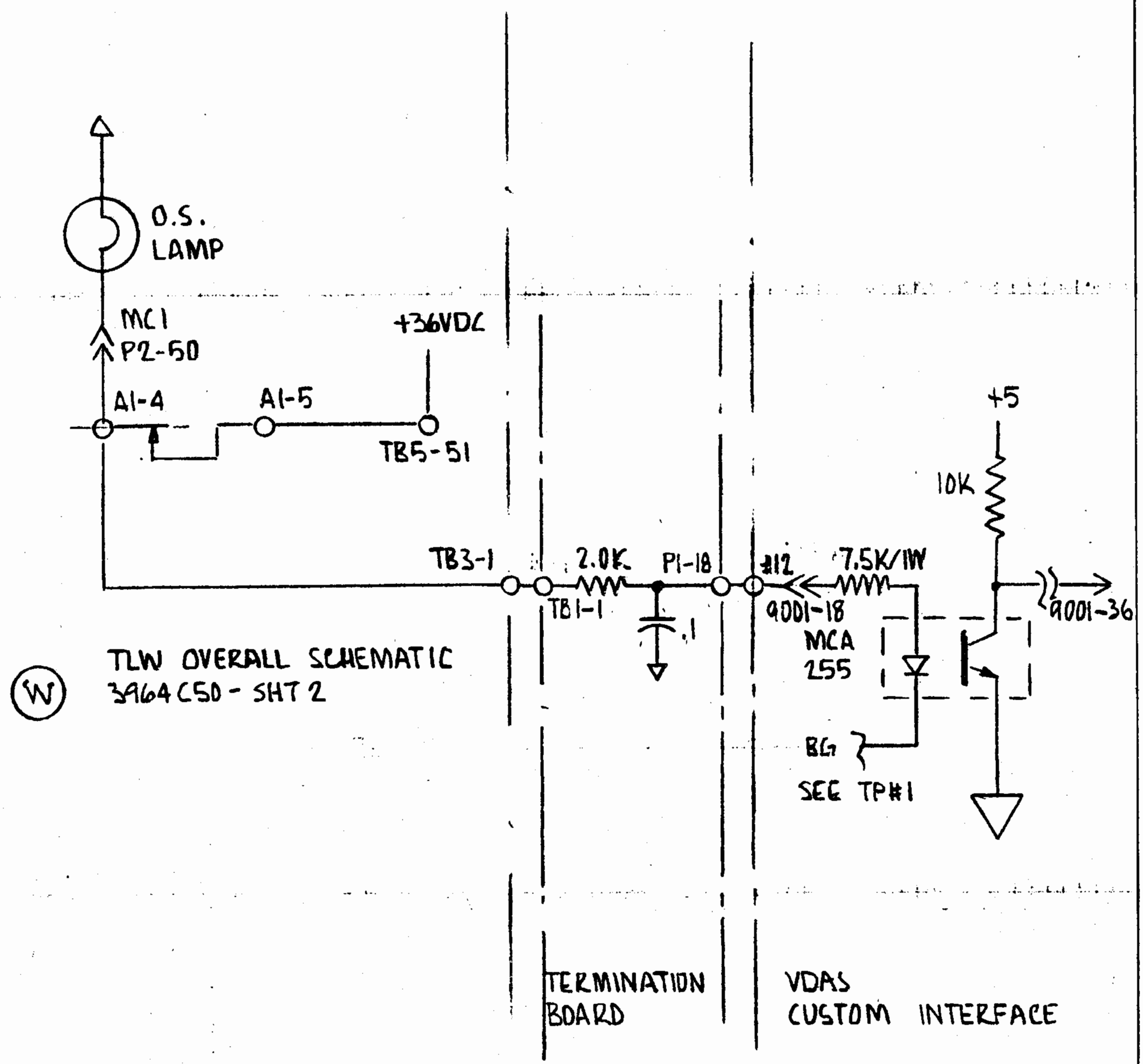
PROJECT STS - VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #21 - TACH #2

DRAWN BY P. STUTZ DATE 11-2-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

DESIGN NO.

PROJECT STS - VDAS TESTPOINT DEFINITION

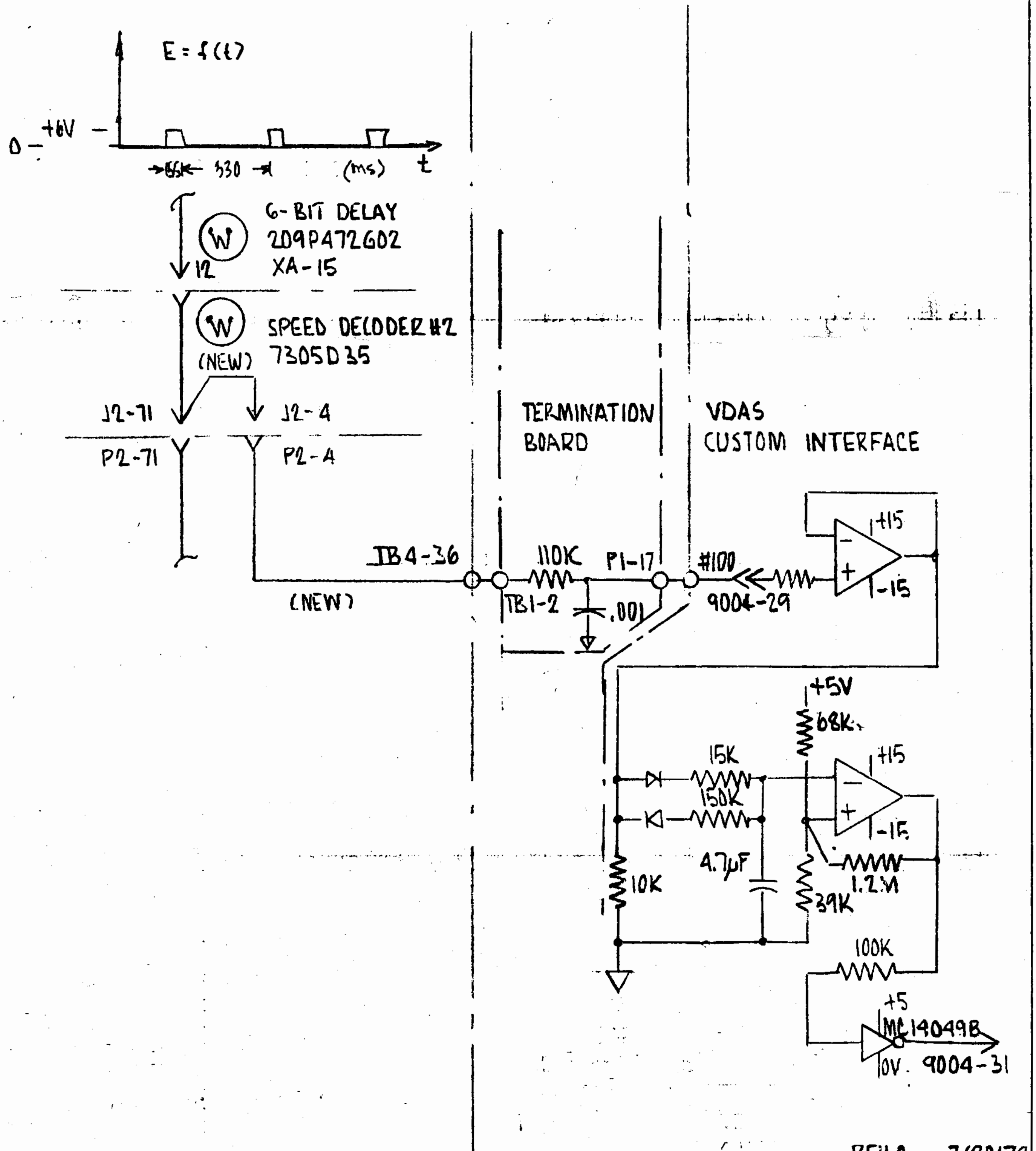
TITLE TESTPOINT #22 - "A" RELAY

DRAWN BY P. STUTZ DATE 11-2-77 APPROVED

DRAWING NO. LCO P-330

POS 120-9

SHT 20 (32)



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

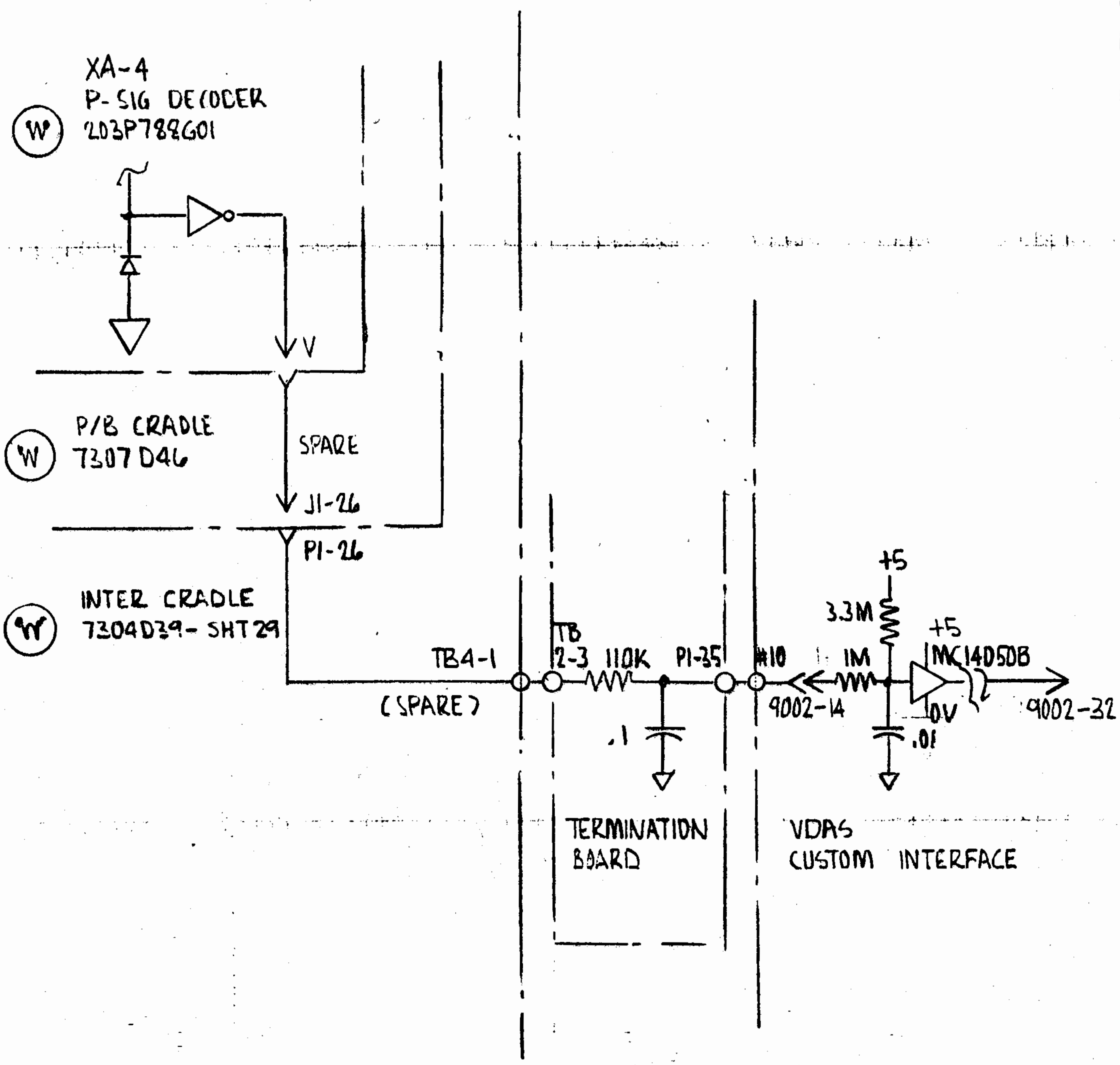
TITLE TESTPOINT #23 - UPDATE

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCO-P-330

POS 120-9



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS-VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #24 - B/P (BRAKE/PROPULSION)

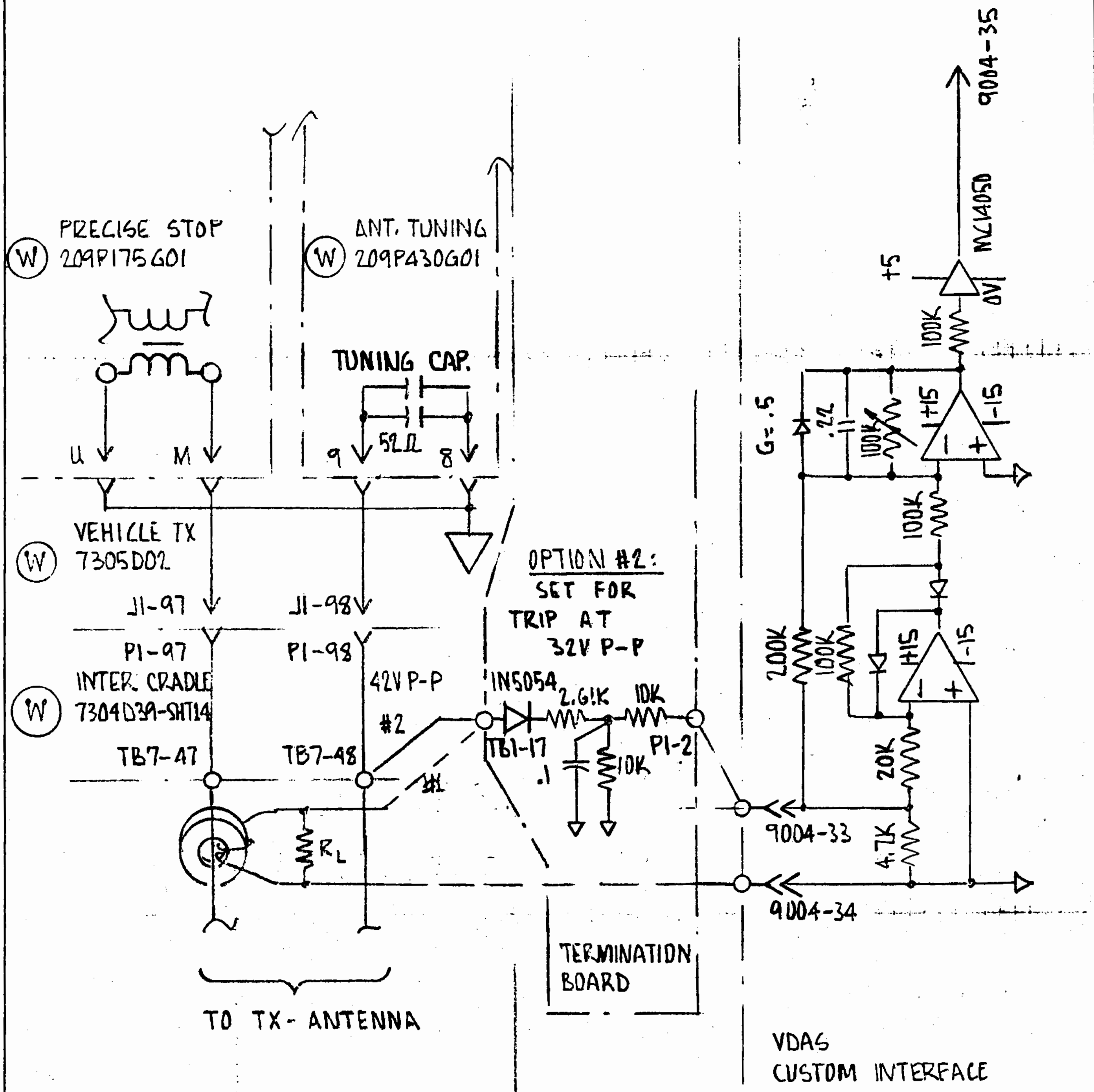
DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCD P-330

POS 120-9

SHT 22 (32)



OPTION :
1. CURRENT TRANSFORMER
2. TUNING CAPACITOR — IN USE

REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION

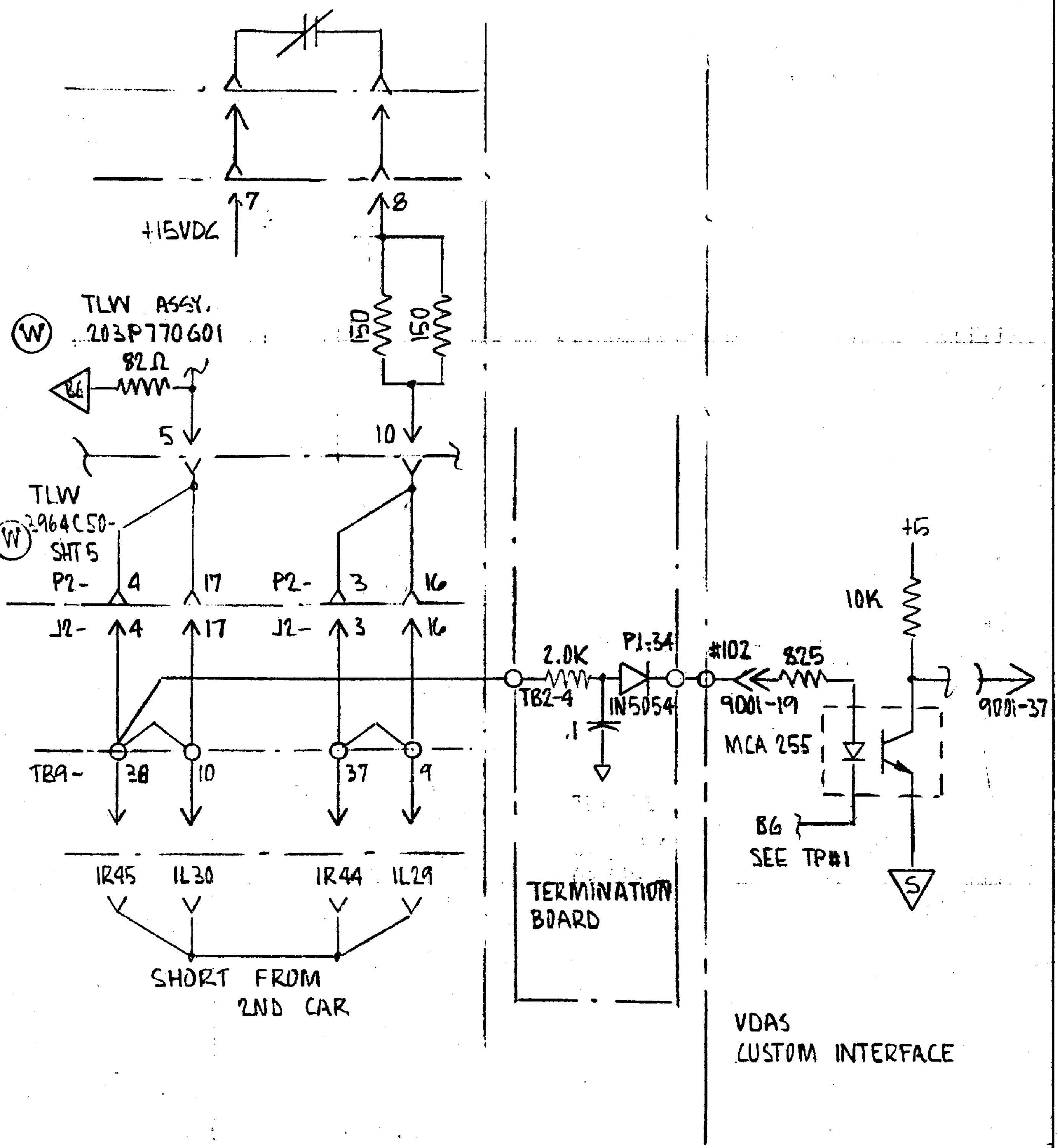
TITLE TESTPOINT #25 - VEHICLE TX, TX CURRENT

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS-VDAS TESTPOINT DEFINITION

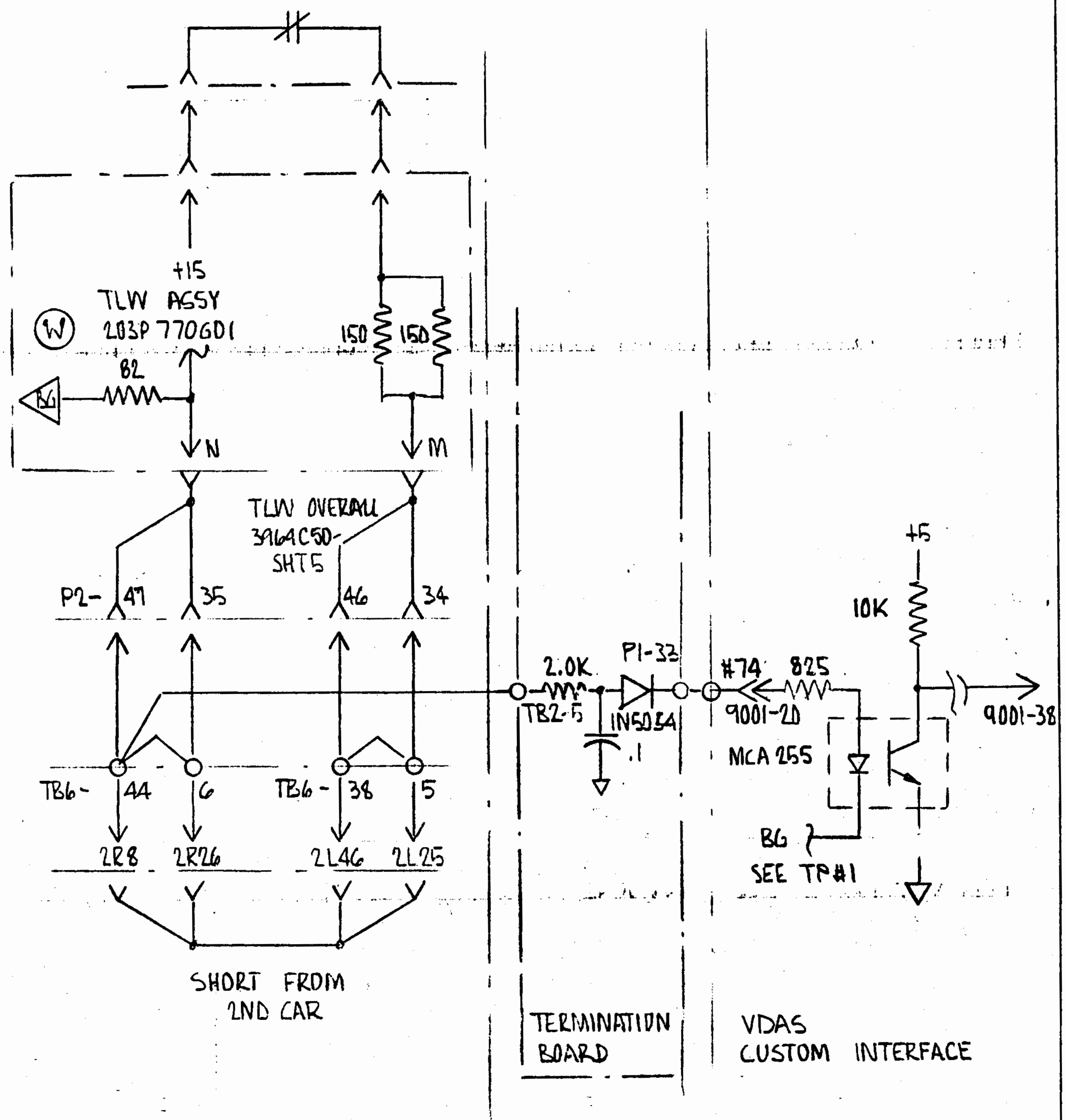
TITLE TESTPOINT # 26 - CAR AHEAD

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330

POS 120-9



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS-VDAS TESTPOINT DEFINITION

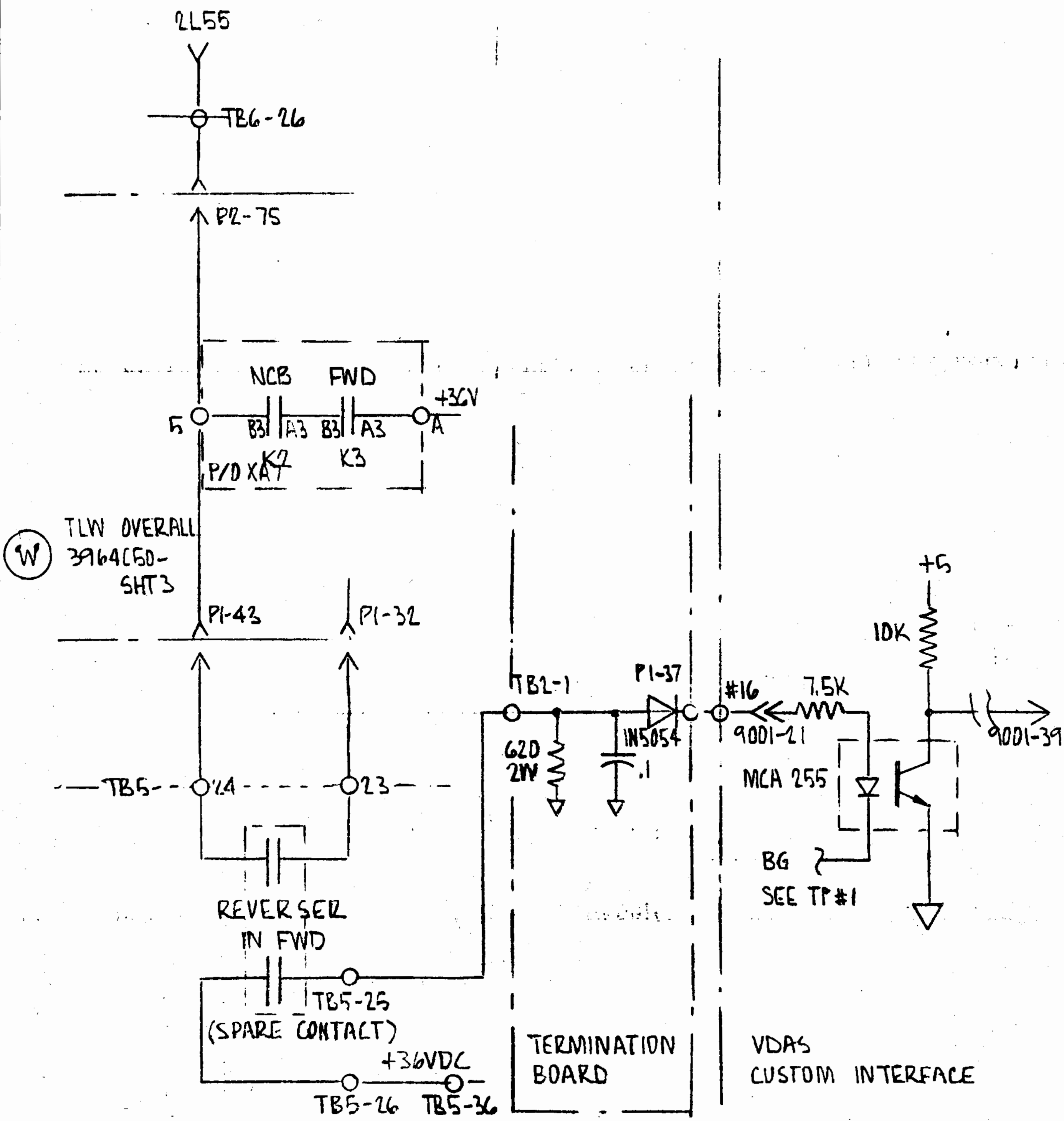
TITLE TESTPOINT #27 - CAR BEHIND

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCD P-330

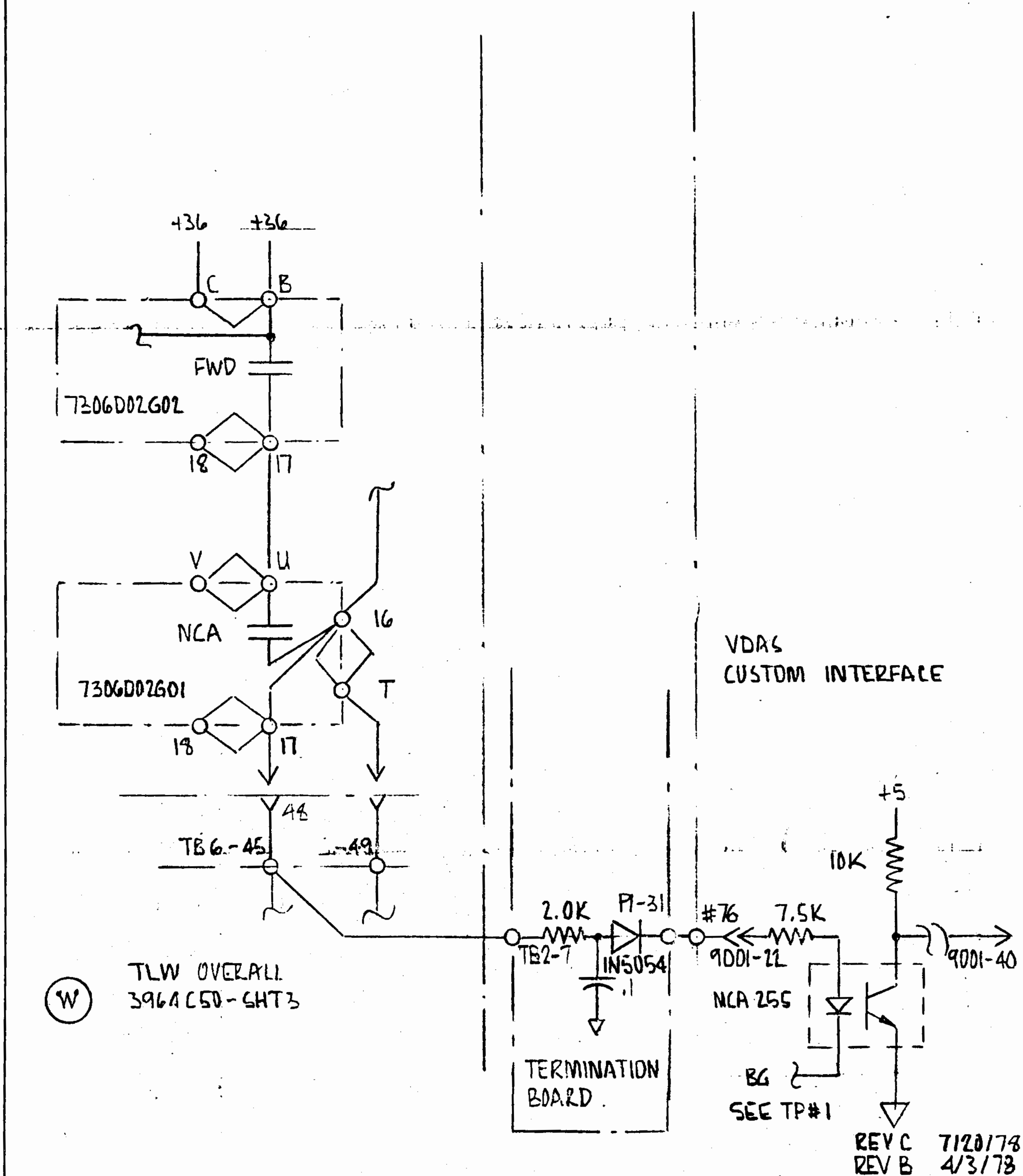
POS 120-9



REV B 4/3/78

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS - TESTPOINT DEFINITION	
TITLE TESTPOINT #24 - DIRECTION SENSE (FWD)	DRAWING NO. LCD P-330
DRAWN BY P. STUTZ DATE 11-7-77 APPROVED	

POS 120-9



PORT OF SEATTLE COMMISSION

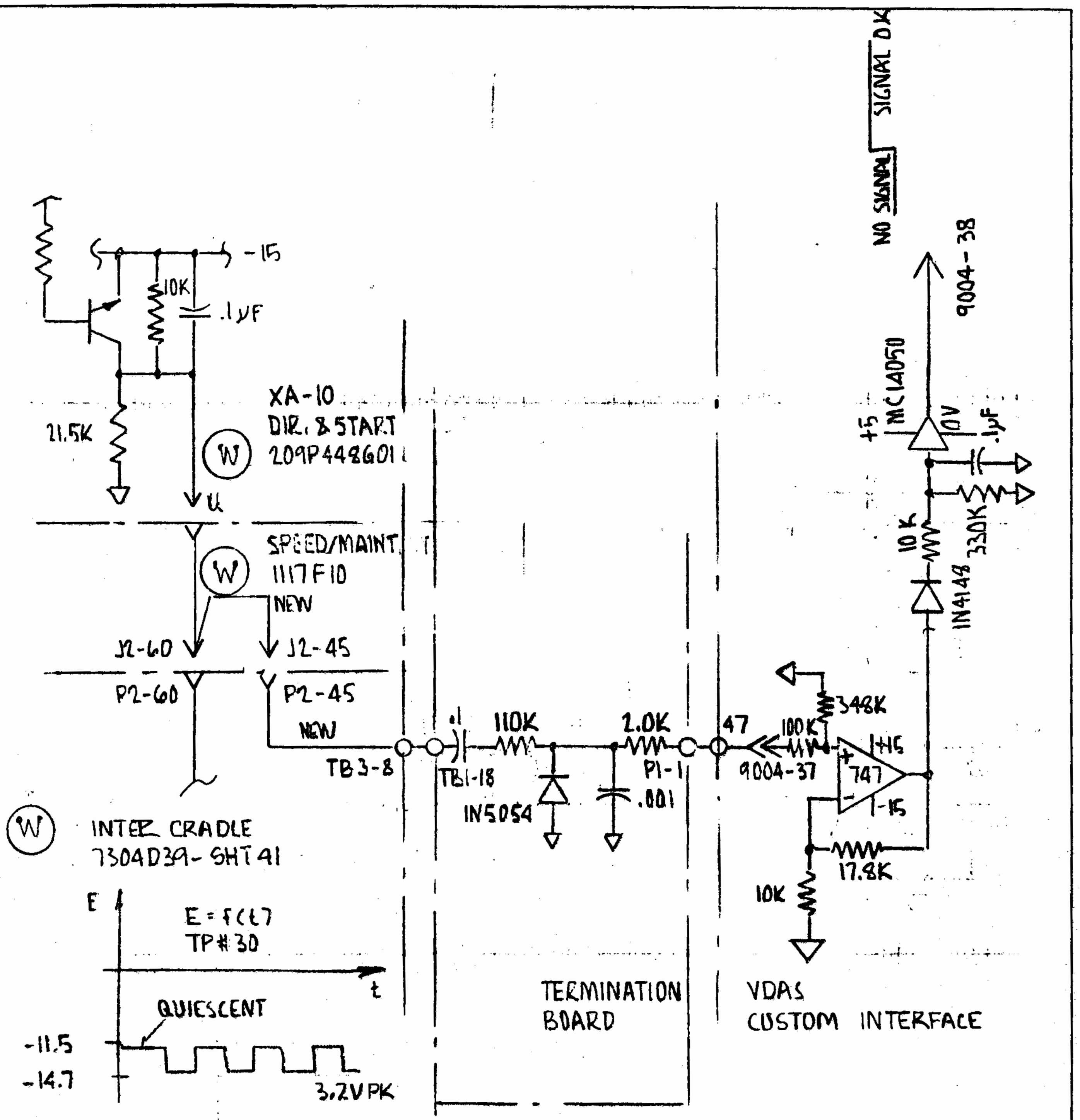
PROJECT STS- VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #29 - DIRECTION COMMAND (FWD)

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-33D



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

DESIGN NO.

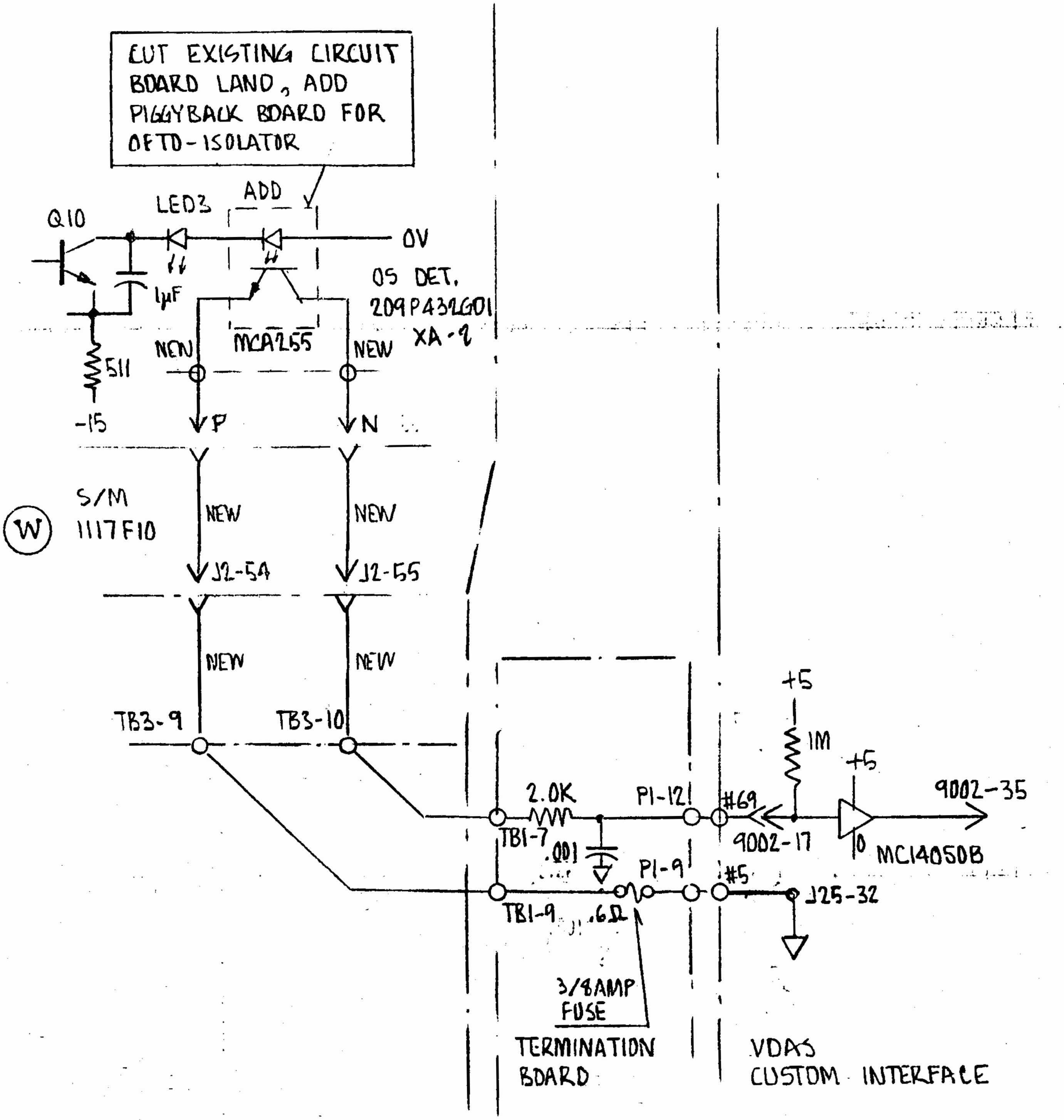
PROJECT STS – VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #30 – P SIGNAL INTERLOCK

DRAWING NO. LCO P-330

DRAWN BY P. STUTZ DATE 11-7-77 APPROVED

POS 120-9



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

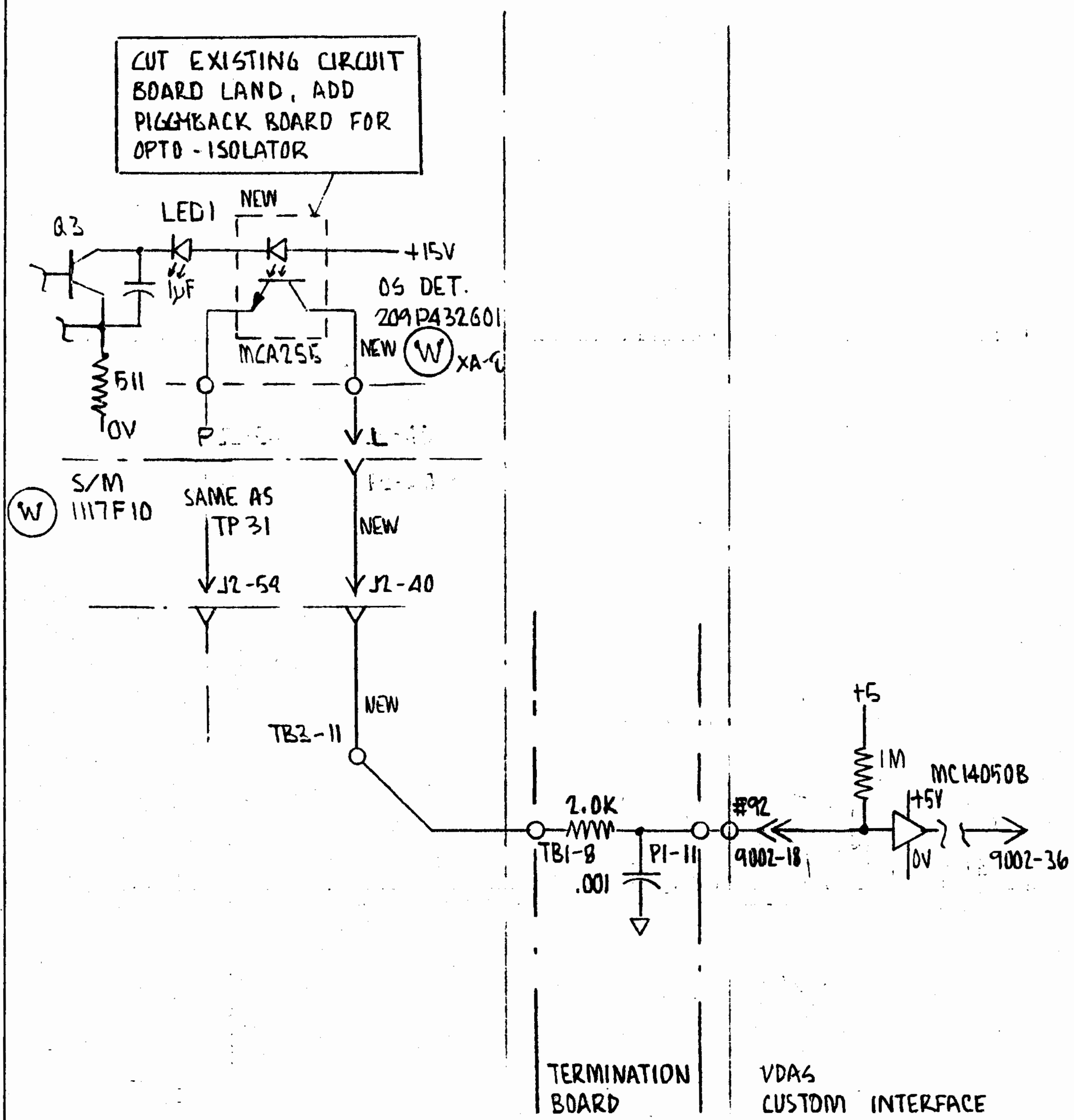
PROJECT STS- VDAS TESTPOINT DEFINITION

TITLE TESTPOINT #31 – OS II / OVERSPEED TACH 1

DRAWN BY P. STUTZ DATE 11-8-77 APPROVED

DESIGN NO.

DRAWING NO. LCO P-330



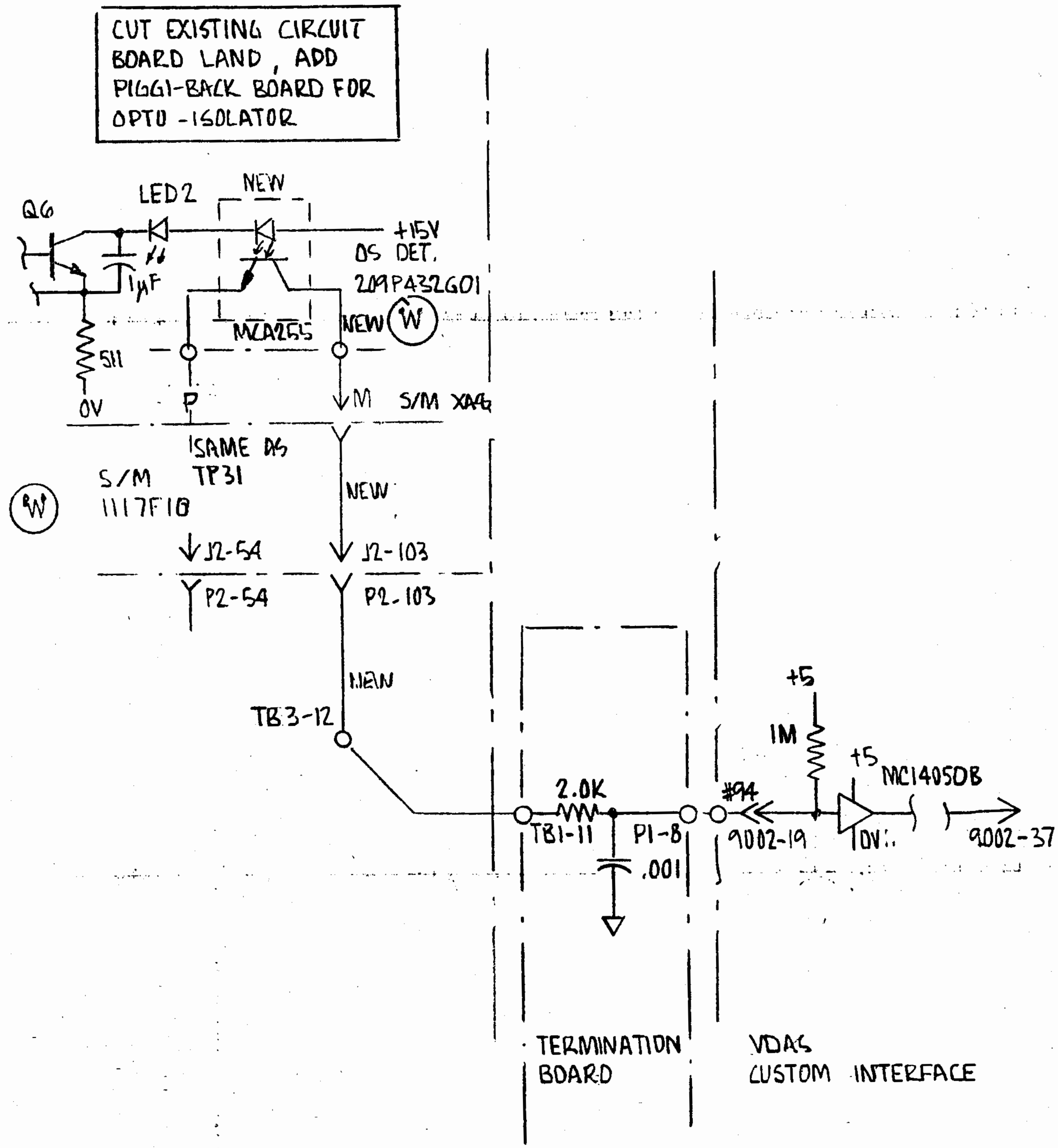
REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS TESTPOINT DEFINITION	DESIGN NO.
TITLE TESTPOINT #32 OS I OVERSPEED TACH 2	DRAWING NO. LCO P-330
DRAWN BY P. STUTZ DATE 11-8 77 APPROVED	

POS 120-9

SHT 30 (32)



REV C 7/20/78
REV B 4/3/78

PORT OF SEATTLE COMMISSION

PROJECT: STS - VDAS TESTPOINT DEFINITION

TITLE: TESTPOINT #33 - BAL / BALANCE TACH1 VS. TACH2

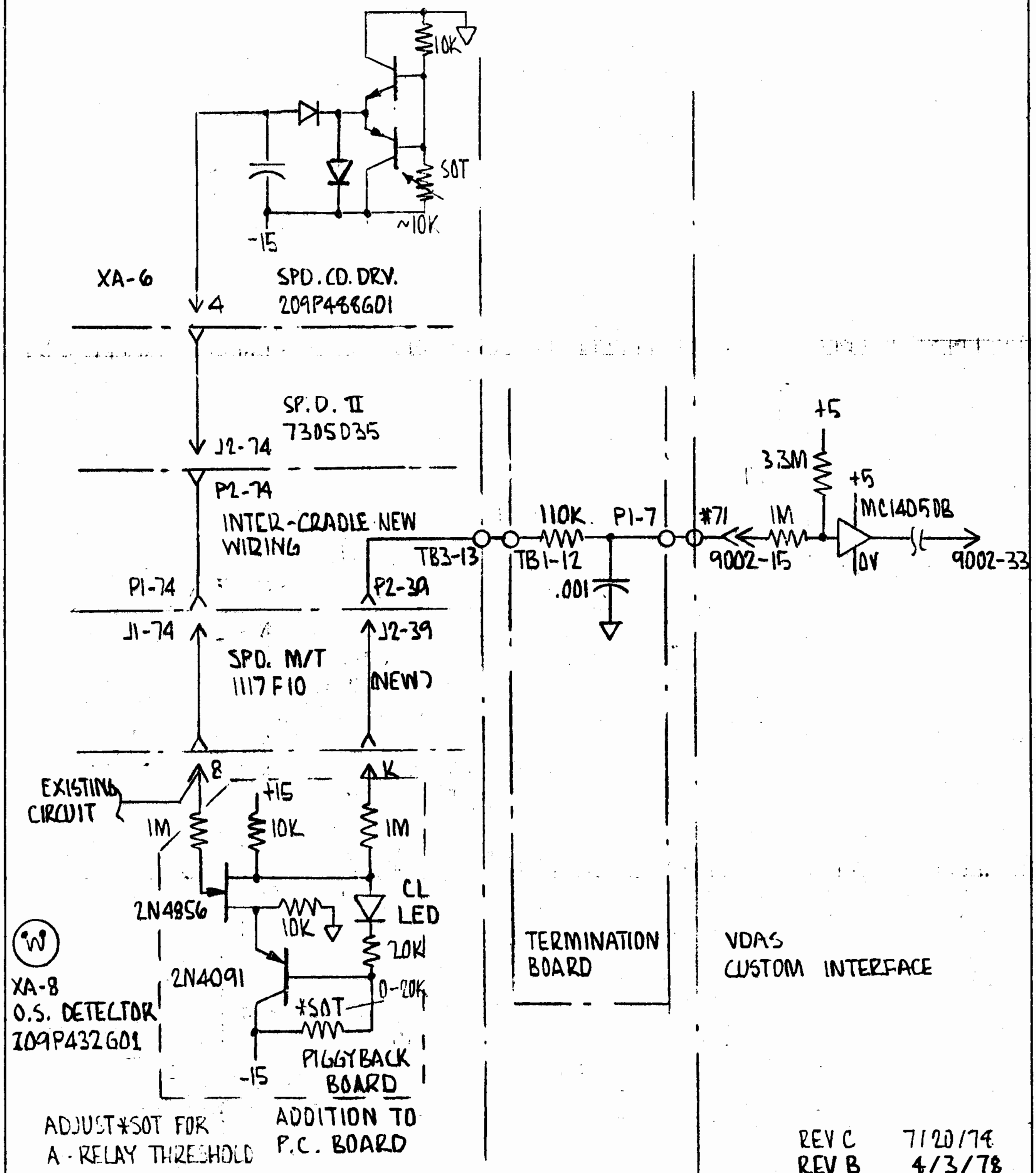
DRAWN BY: P. STUTZ DATE: 11-8-77 APPROVED:

DESIGN NO.:

DRAWING NO.: LCO P-330

POS 120-9

SHT 31 (32)



REV C 7/20/78
REV B 4/3/78
REV A 12/9/77

PORT OF SEATTLE COMMISSION

PROJECT: STS - VDAS TESTPOINT DEFINITION

TITLE: TESTPOINT # 34 - CODE LOAD

DRAWN BY: P. STUTZ DATE: 11-4-77 APPROVED:

DESIGN NO.:

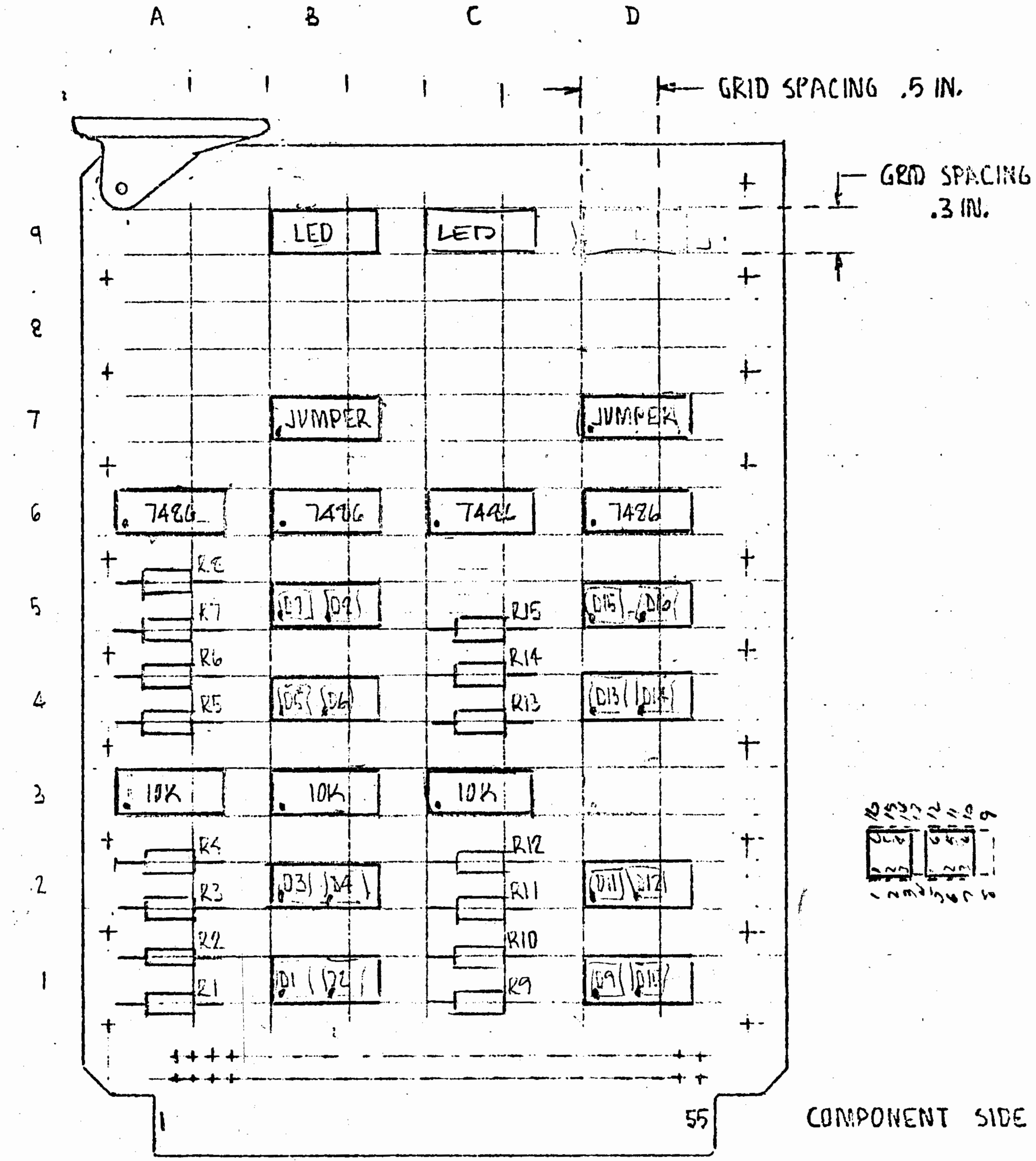
DRAWING NO.: LCD P-33D

POS 120-9

SHT 32 (32)

APPENDIX B

Prolog System Information

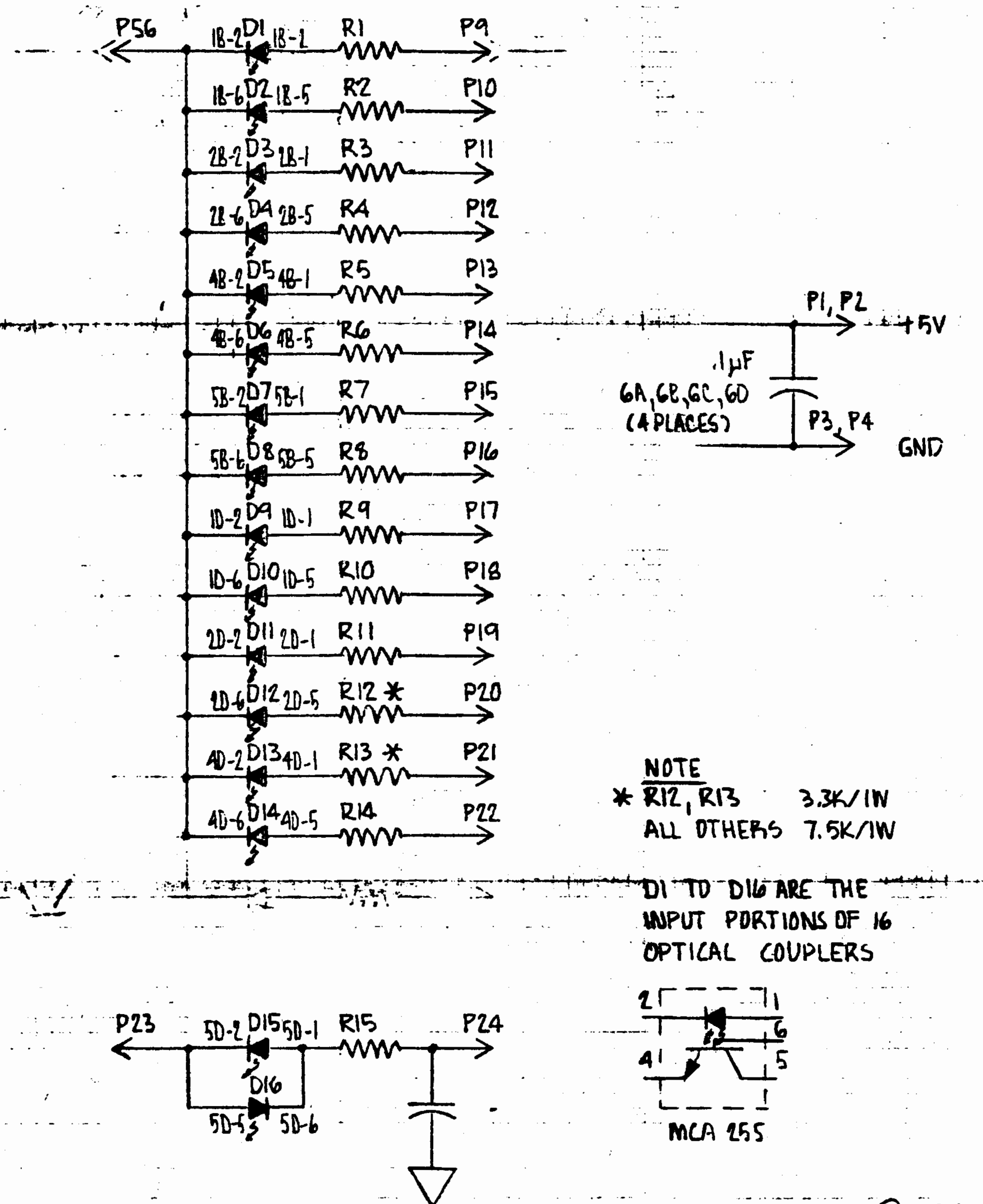


PROLOG PART NUMBER #P561

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT PROLOG UTILITY DIP CARD	
TITLE	DRAWING NO.
DRAWN BY P. STUTZ DATE APPROVED	

POS 120-9

9221

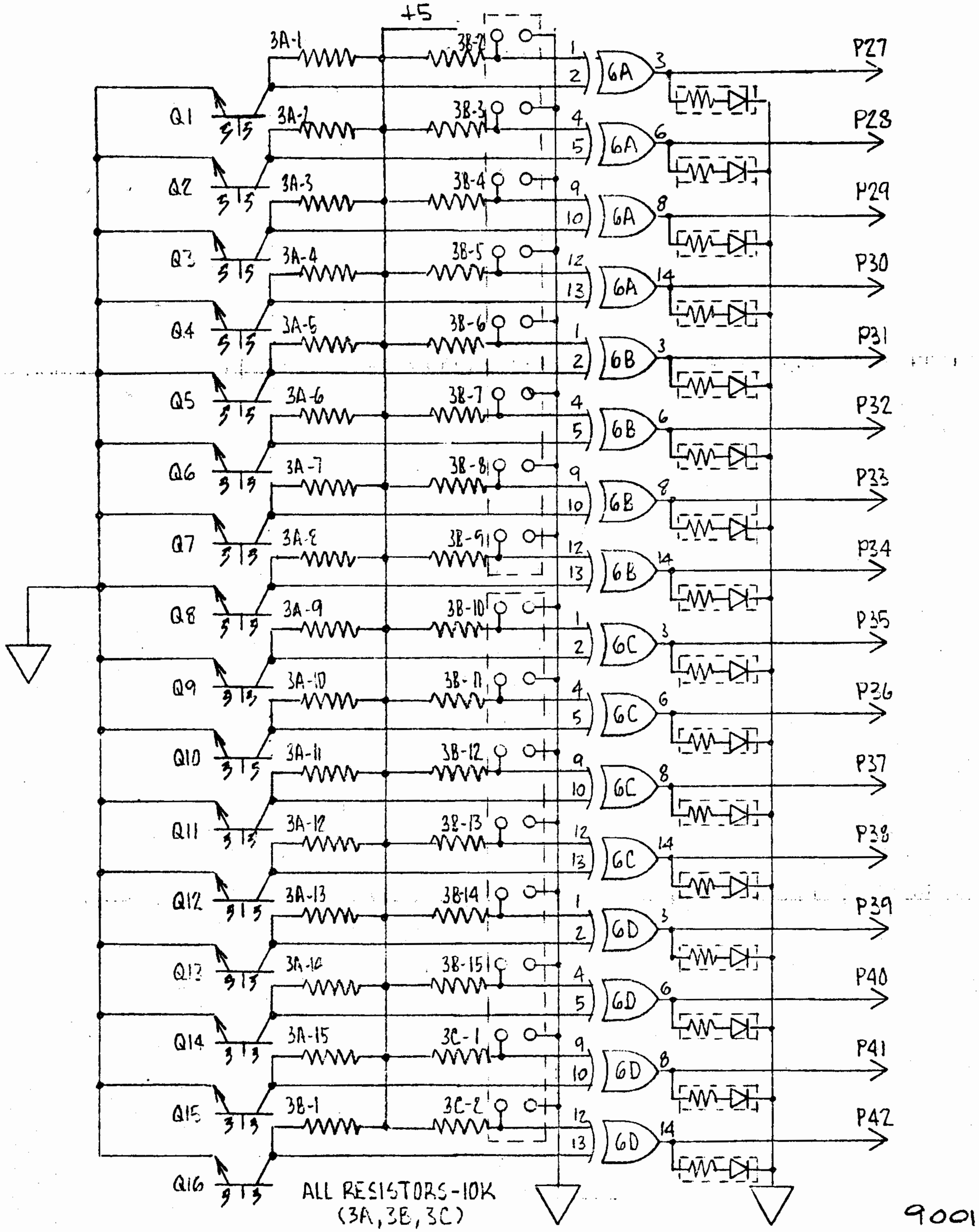


9001

PORT OF SEATTLE COMMISSION

PROJECT STS- VDAC	DESIGN NO.
TITLE ISOLATED INPUT CARD (36VDC - TTL LEVEL)	DRAWING NO.
DRAWN BY P. STUTZ DATE 12/19/77 APPROVED	

POS 128-9



9001

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAC

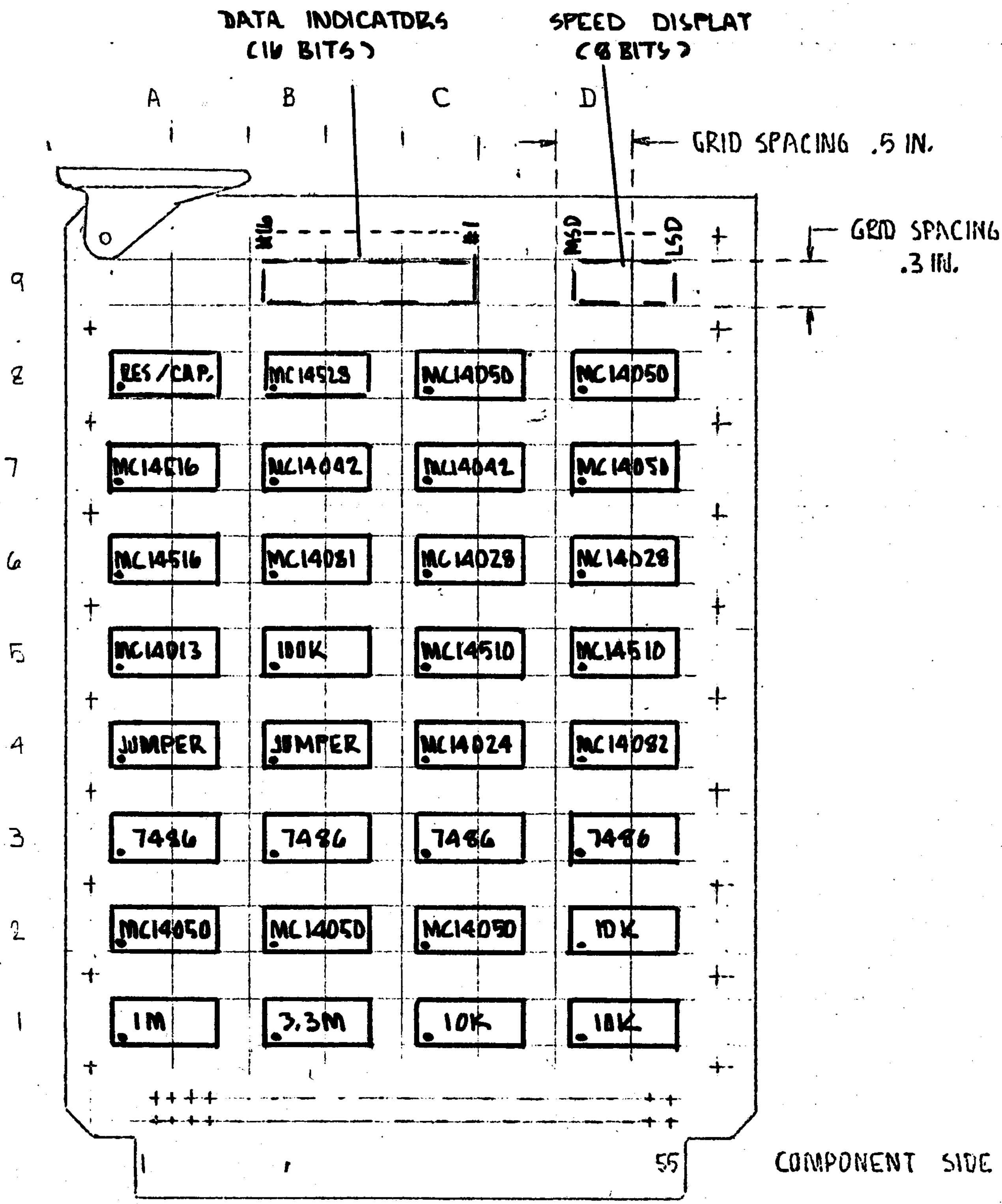
TITLE ISOLATED INPUT CARD (36VDC - TTL LEVEL)

DRAWN BY P. STUTZ DATE 12/19/77 APPROVED

DESIGN NO.

DRAWING NO.

POS 120-9



PROLOG PART NUMBER #P561

PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT PROLOG UTILITY DIP CARD	
TITLE HI-Z INPUT, SPEED INDICATOR	DRAWING NO.
DRAWN BY P. STUTZ DATE ______ APPROVED ______	

P09 120-9

0002

PORT OF SEATTLE COMMISSION

PROJECT STS - VCAS

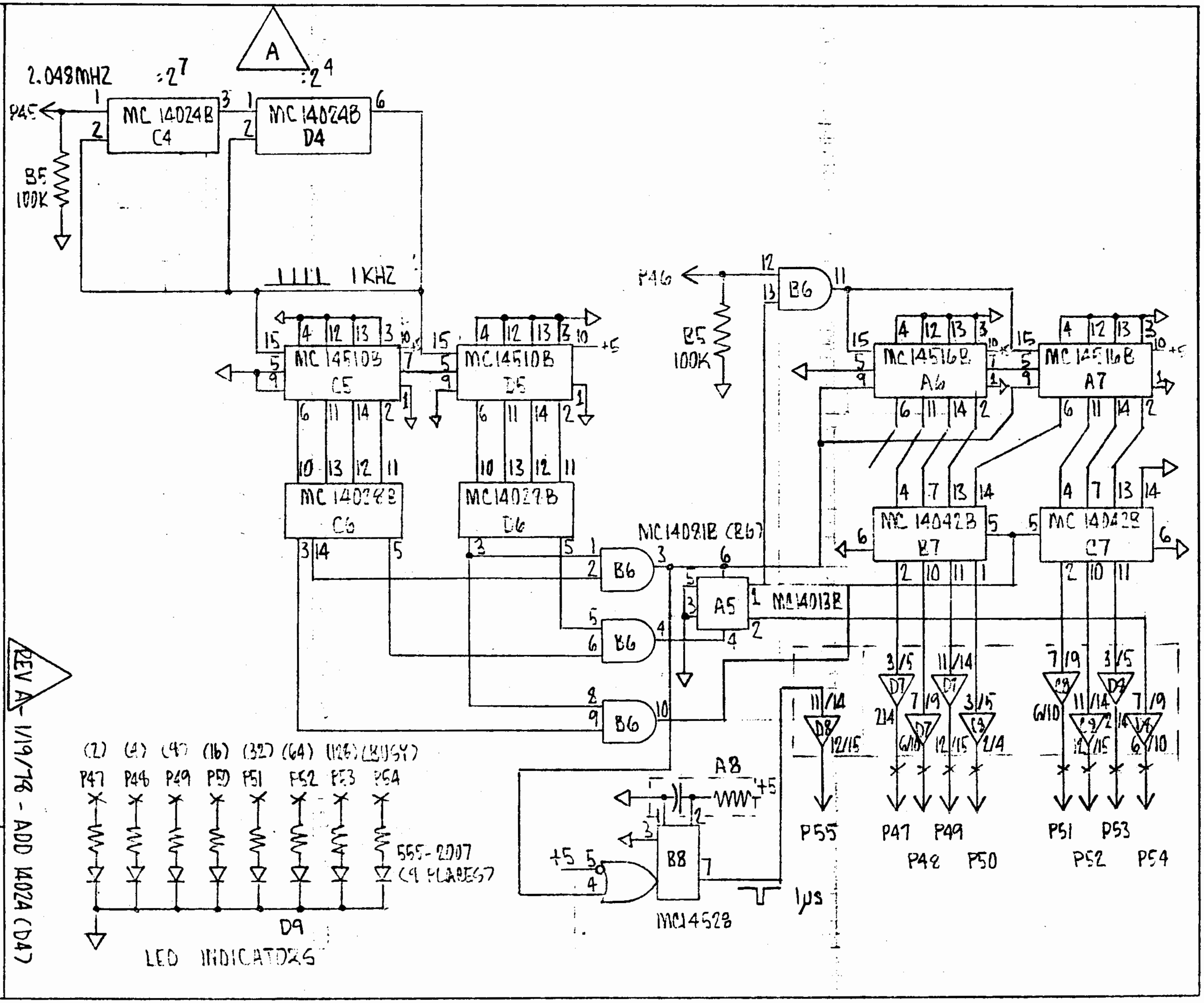
TITLE TACH SIGNAL SCALER

DRAWN BY P. STUTZ DATE 12-27-77 APPROVED

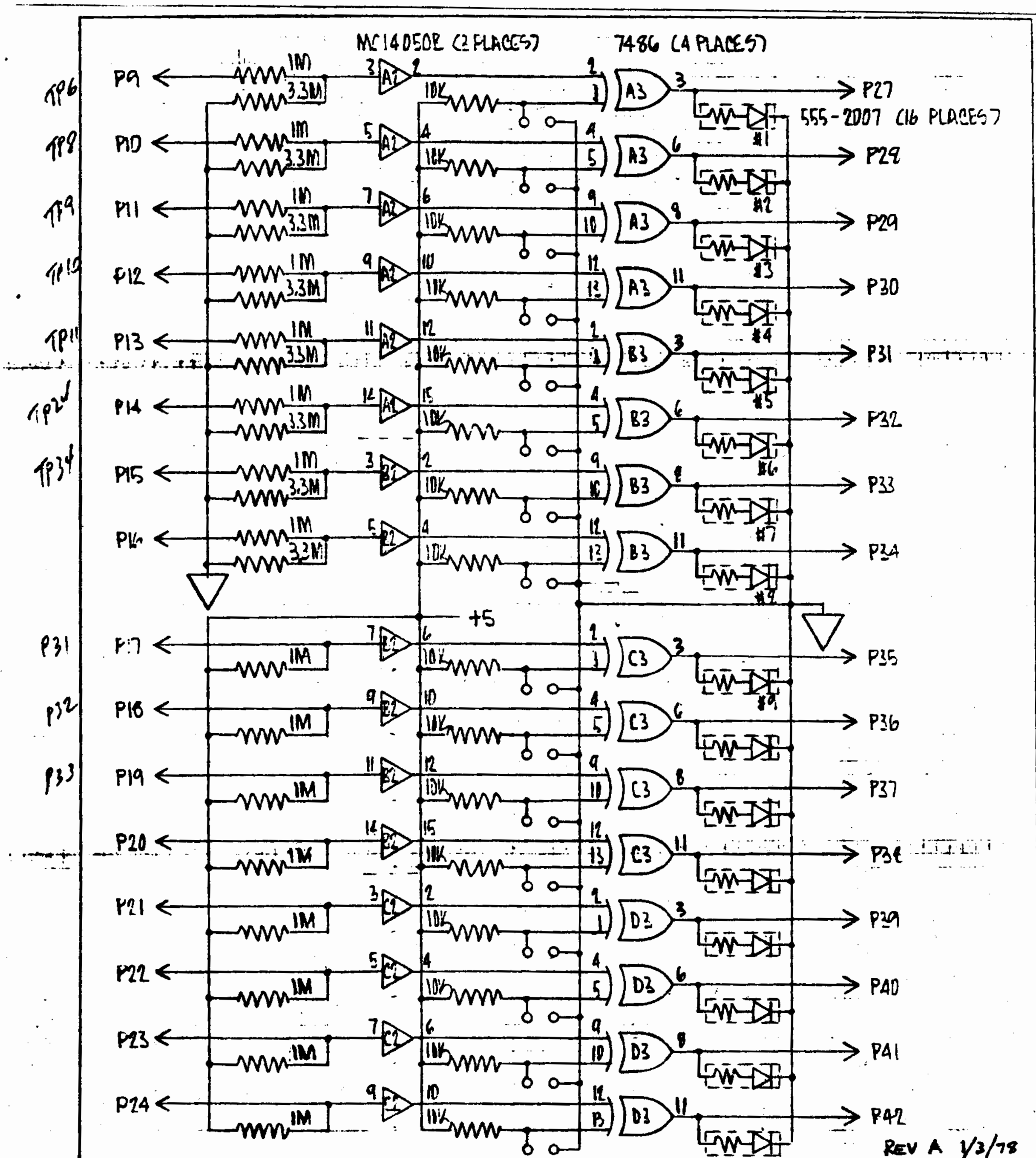
DESIGN NO.

DRAWING NO.

POS 120-9



9002



PORT OF SEATTLE COMMISSION

PROJECT STS- VDAS

TITLE CMOS INPUT CARD (+6V, +15V LEVEL)

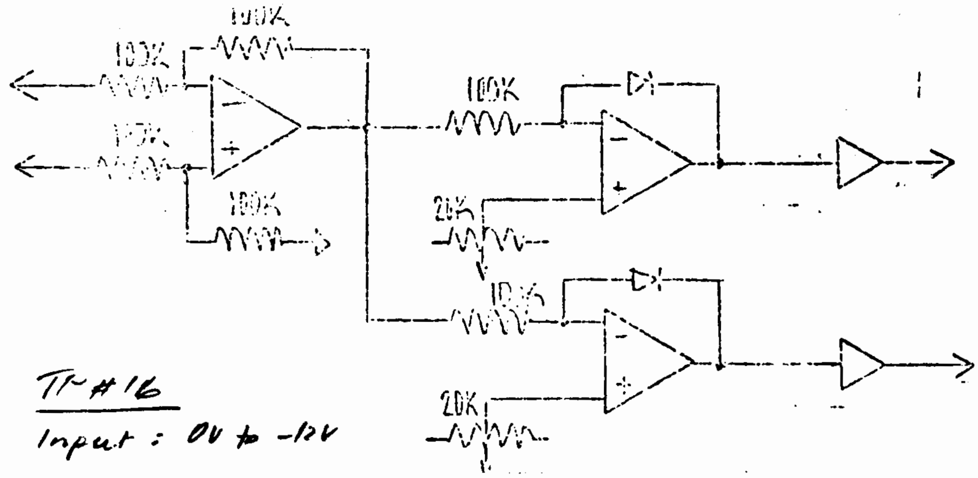
DRAWN BY P. STUTZ DATE 12/23/77 APPROVED

DESIGN NO.

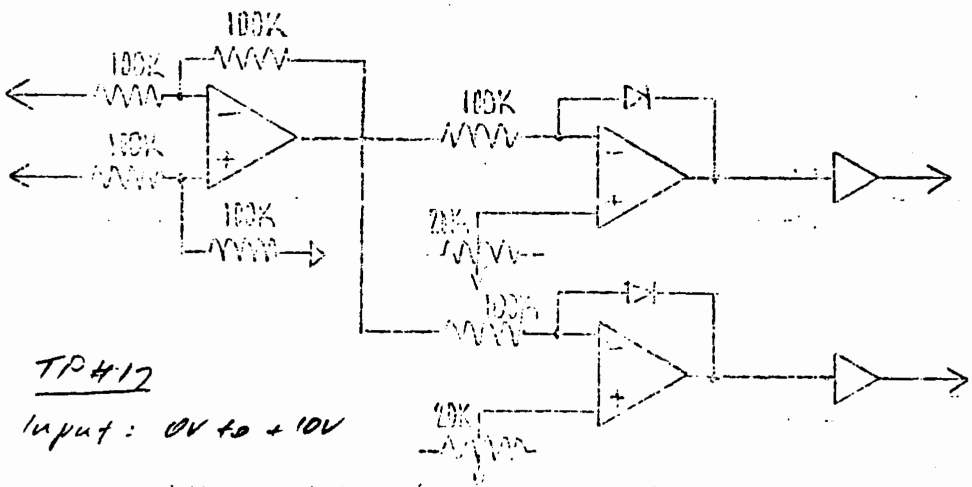
DRAWING NO.

POS 120-9

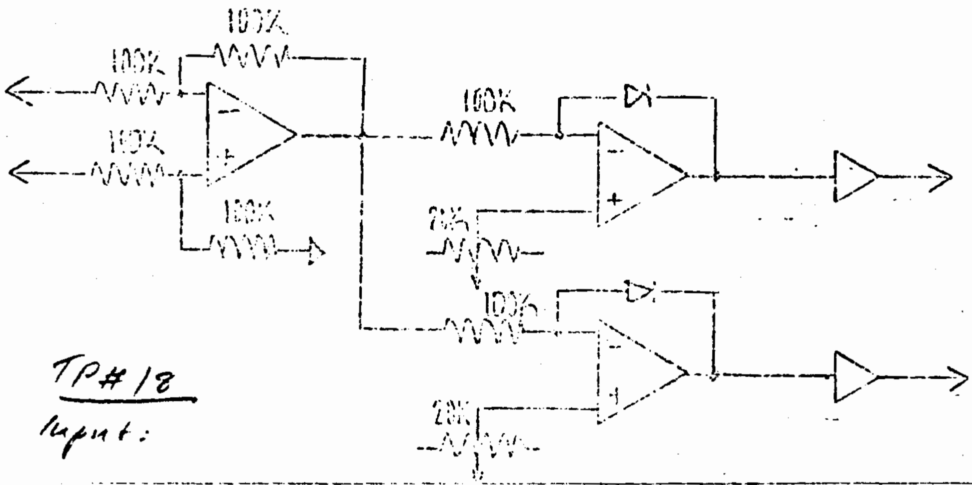
9002



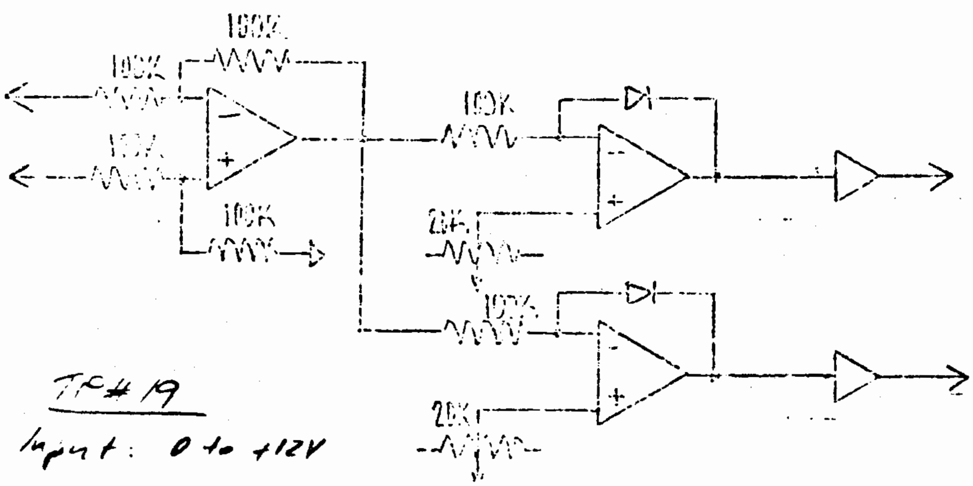
TP#16
Input: 0V to -12V



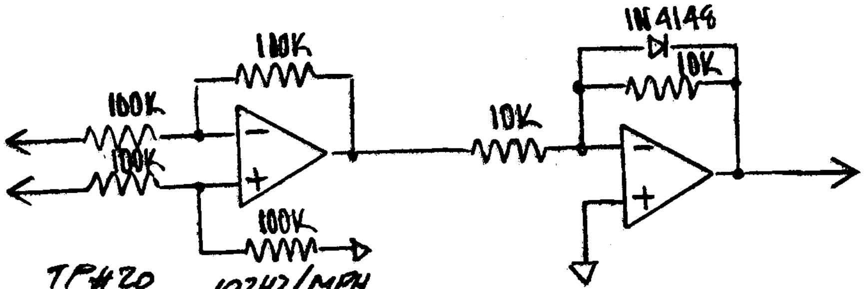
TP#17
Input: 0V to +10V



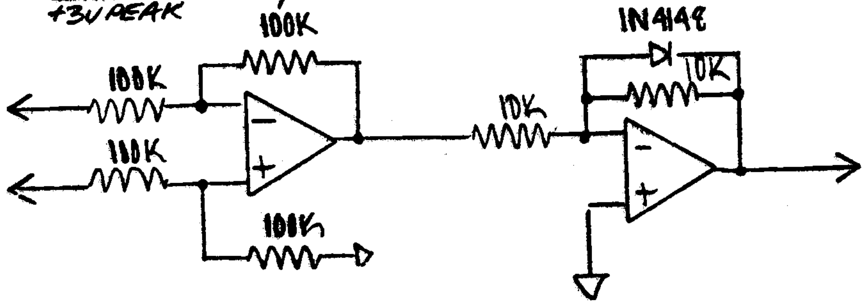
TP#18
Input:



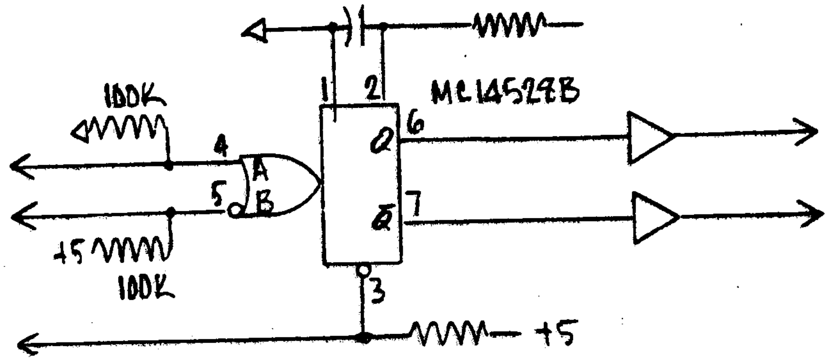
TP#19
Input: 0 to +12V



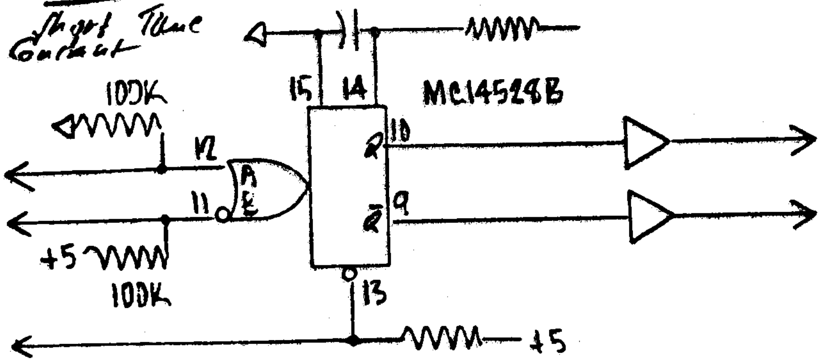
TP#20
+3V PEAK
10HZ/MPH



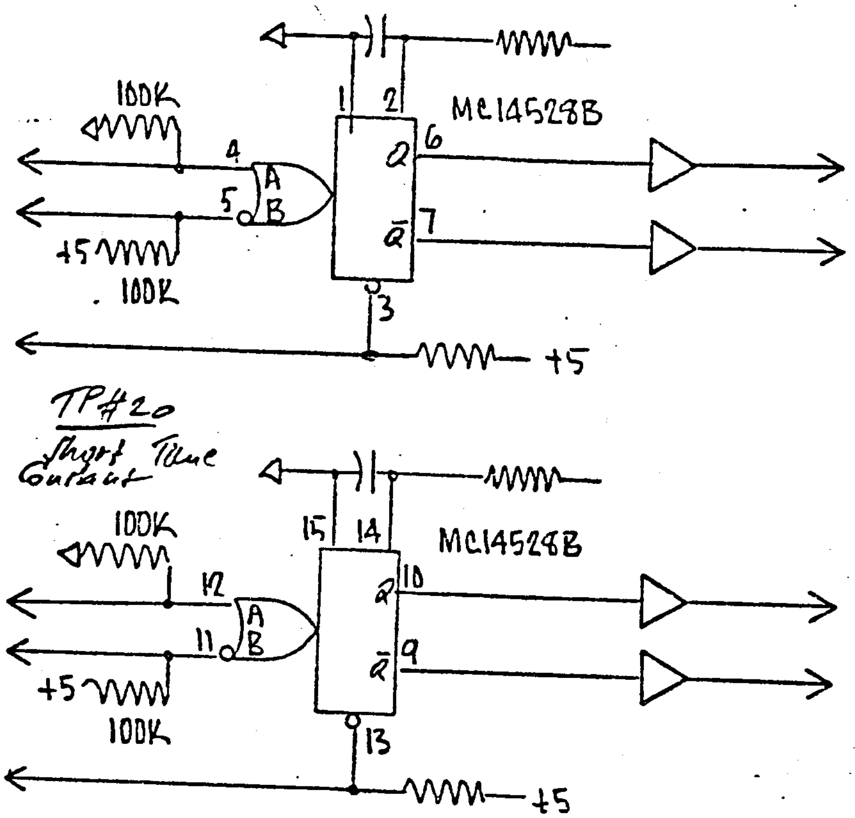
TP#21
+3V PEAK
10ZHZ/MPH



TP#20
Short Time Constant

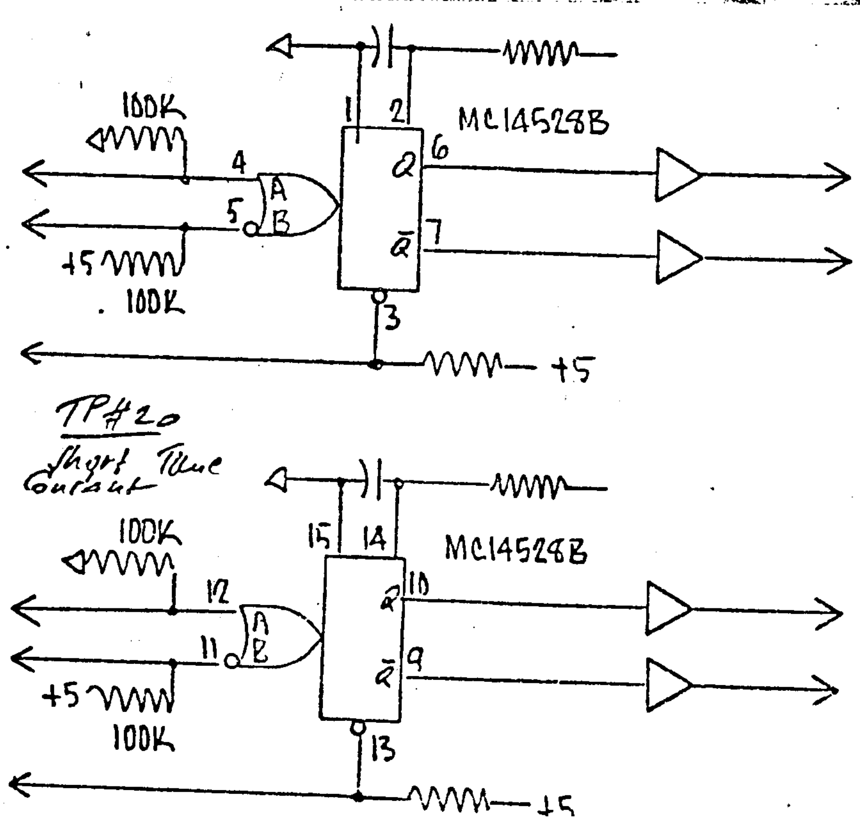


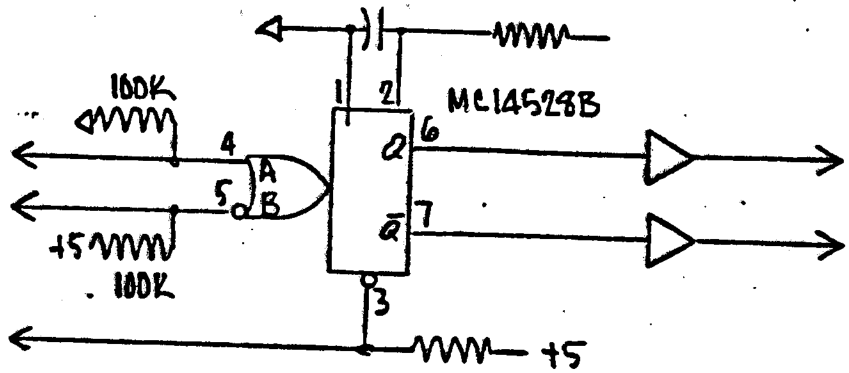
TP#21
Time Constant long enough to maintain Q



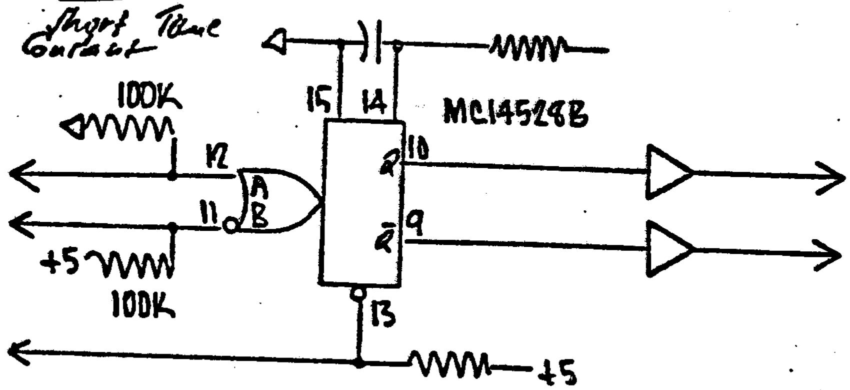
TP#21

Time Constant long enough to maintain Q

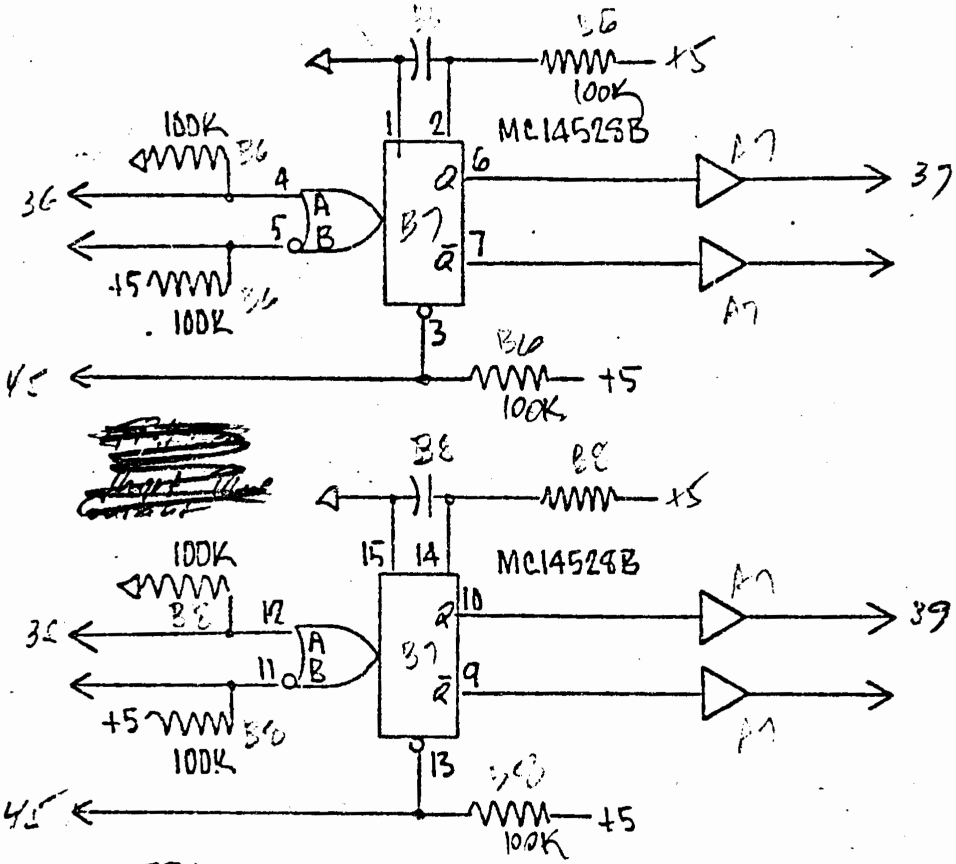




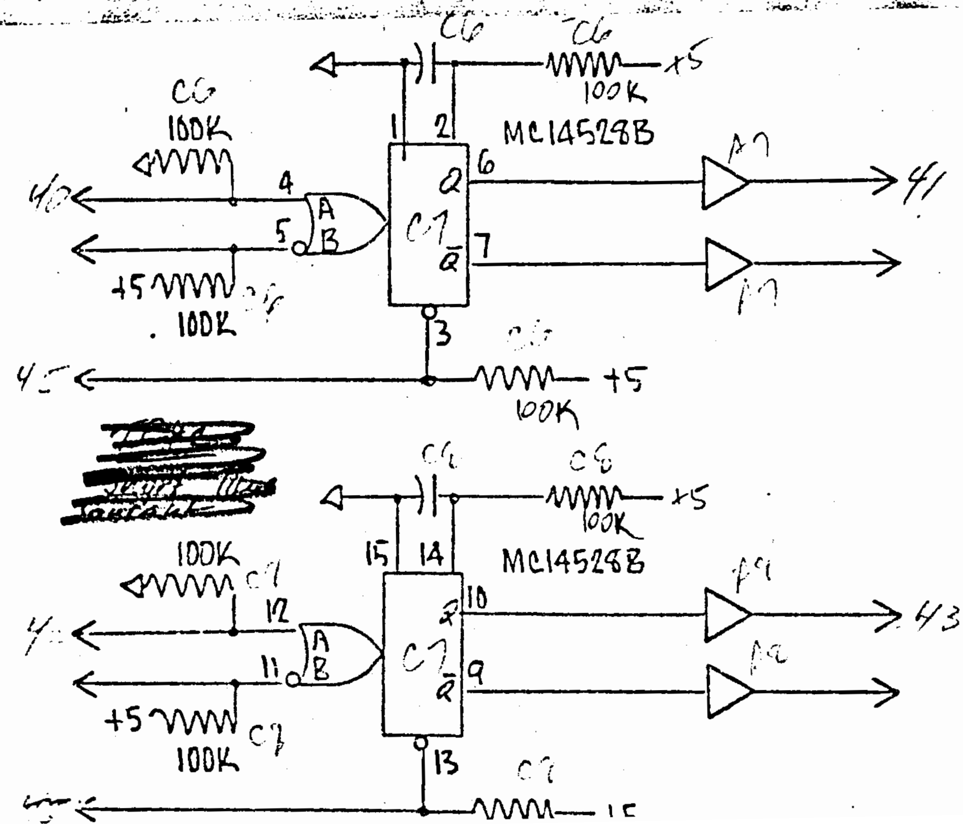
TP#20
Short Time Constant



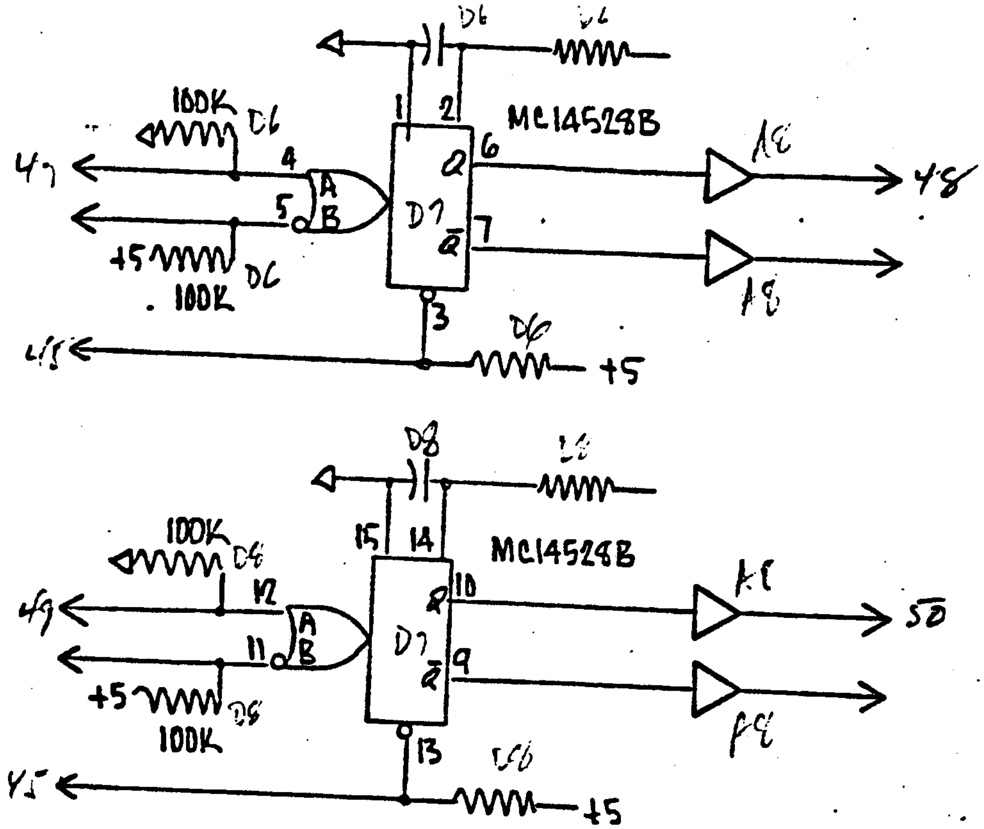
TP#21
Time Constant long enough to maintain Q



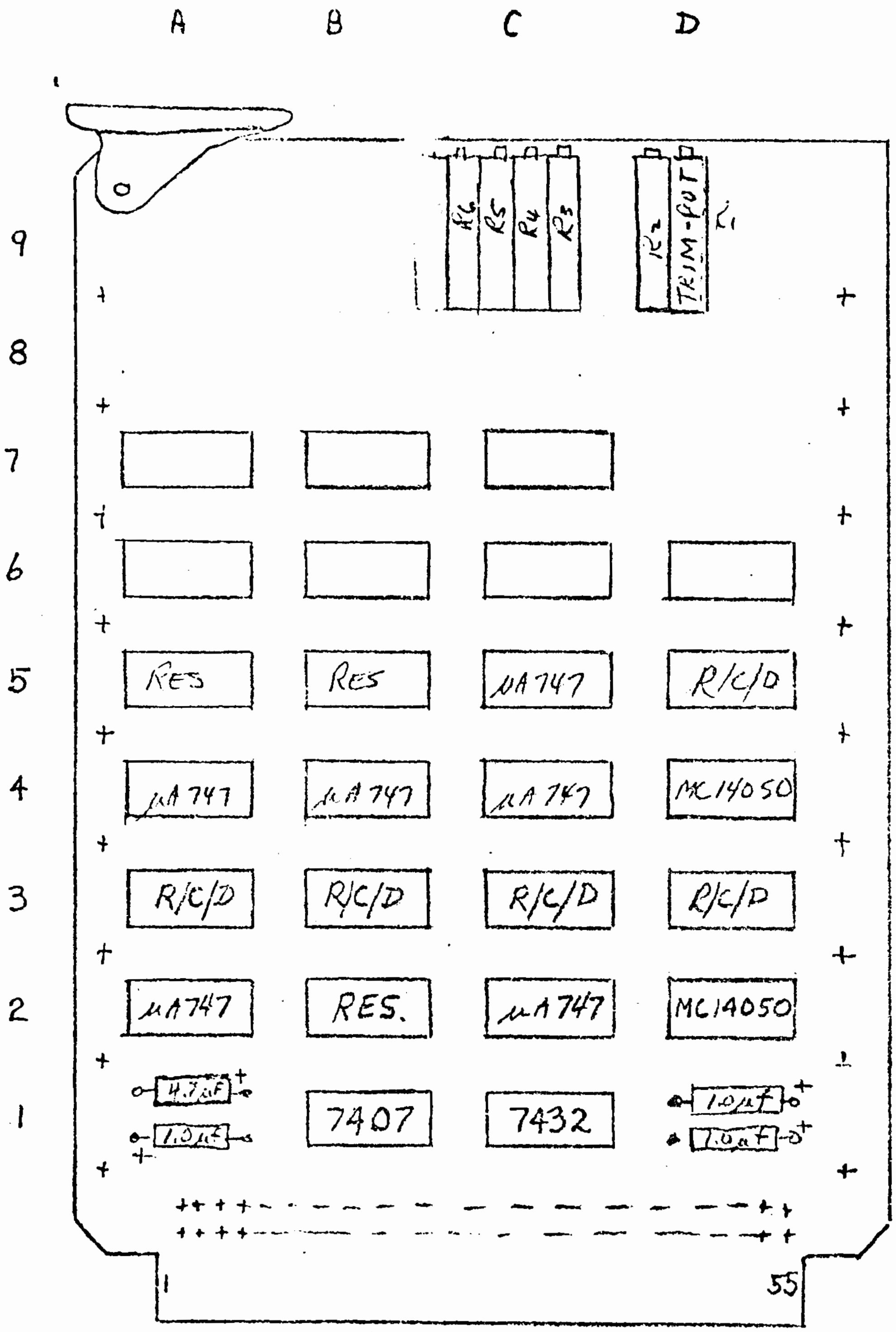
100K
MC14528B
+5
100K
B7
A
B
Q
37
39
100K
MC14528B
+5
100K
45



C6
100K
MC14528B
+5
100K
C7
A
B
Q
41
43
100K
MC14528B
+5
100K
45

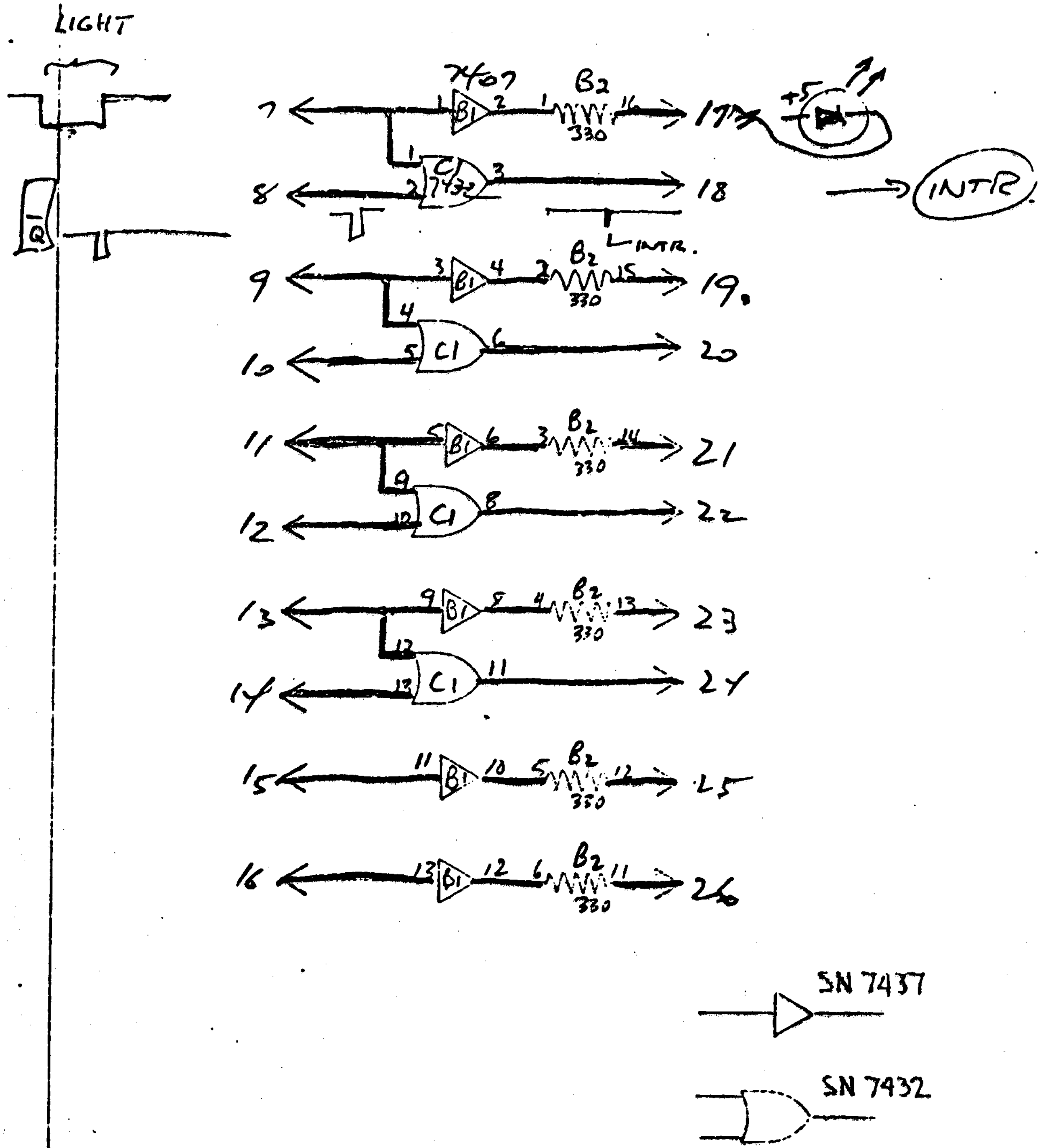


1,2 +5VDC
3,4 GND
5,6 -5VDC (NOT USED)
53,54 -12VDC
55,56 +12VDC

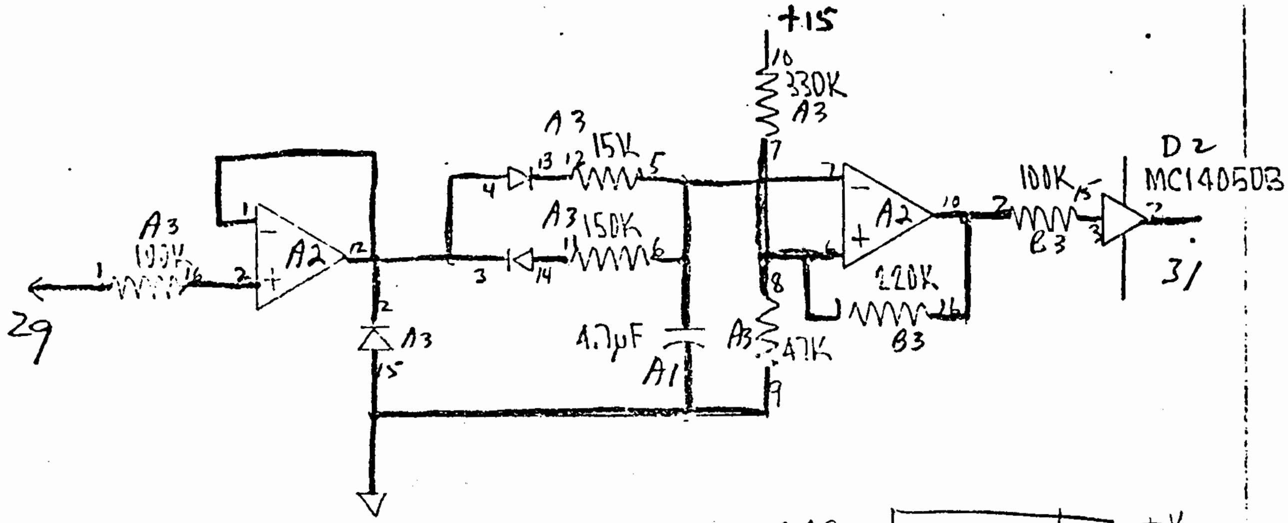


Control Card

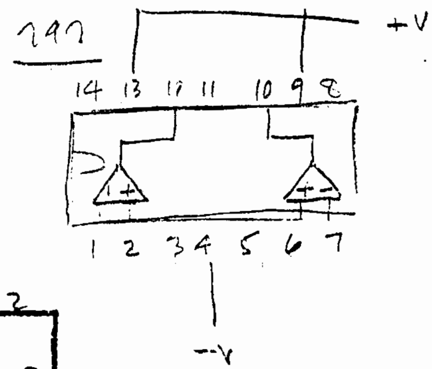
TITLE MISC

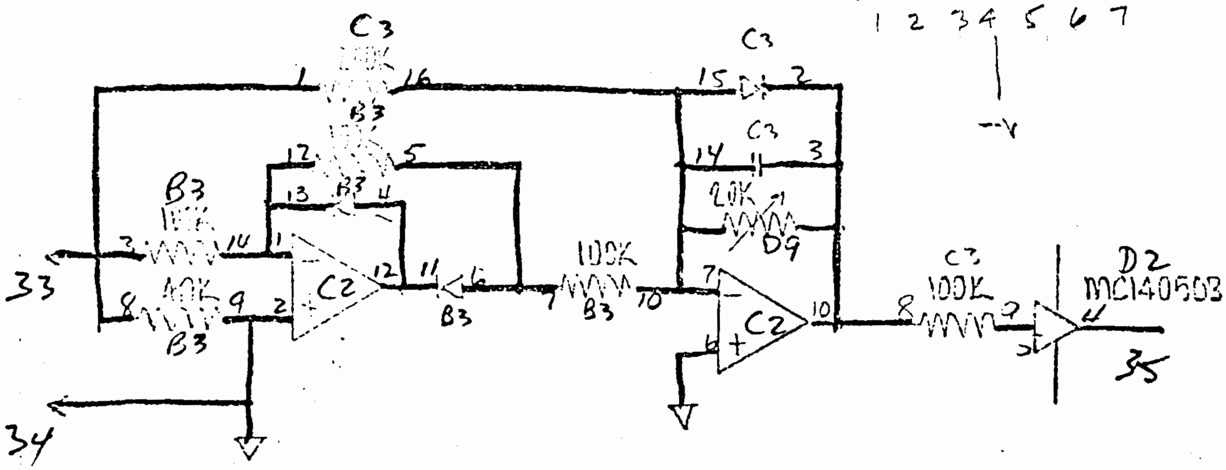


INTERRUPT GATING

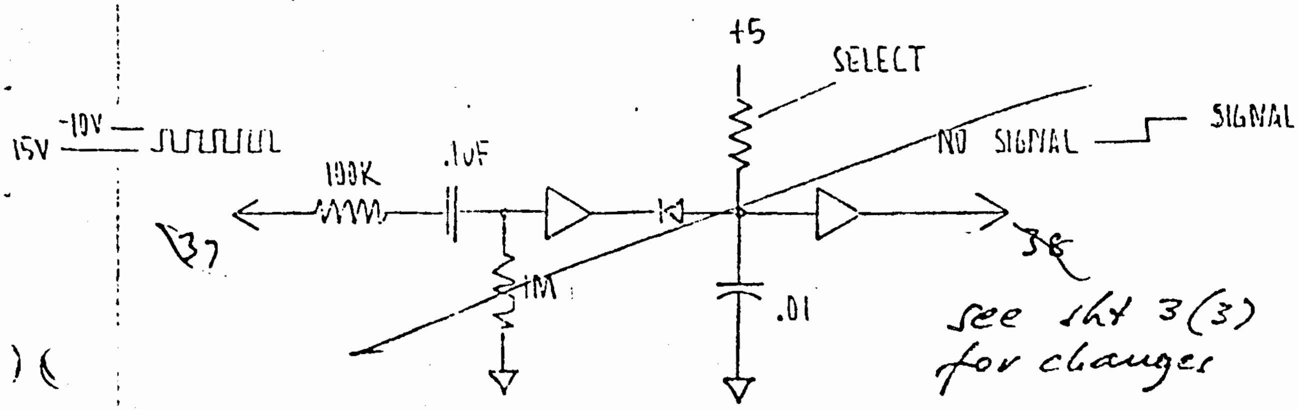


TP# 22





TP# 25



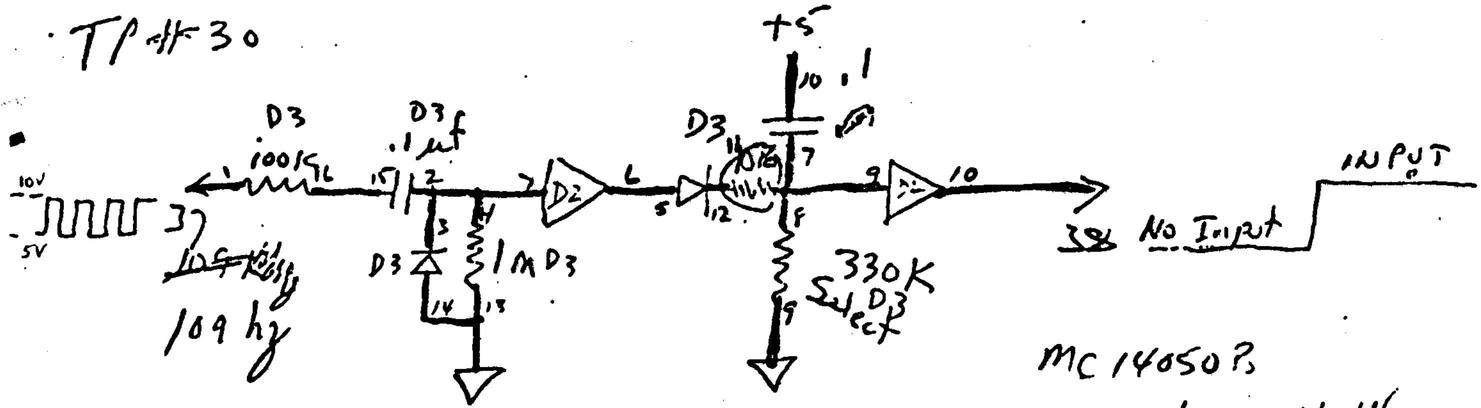
see sht 3(3) for changes

TP#30

Sht 2(3)

MC 14050B

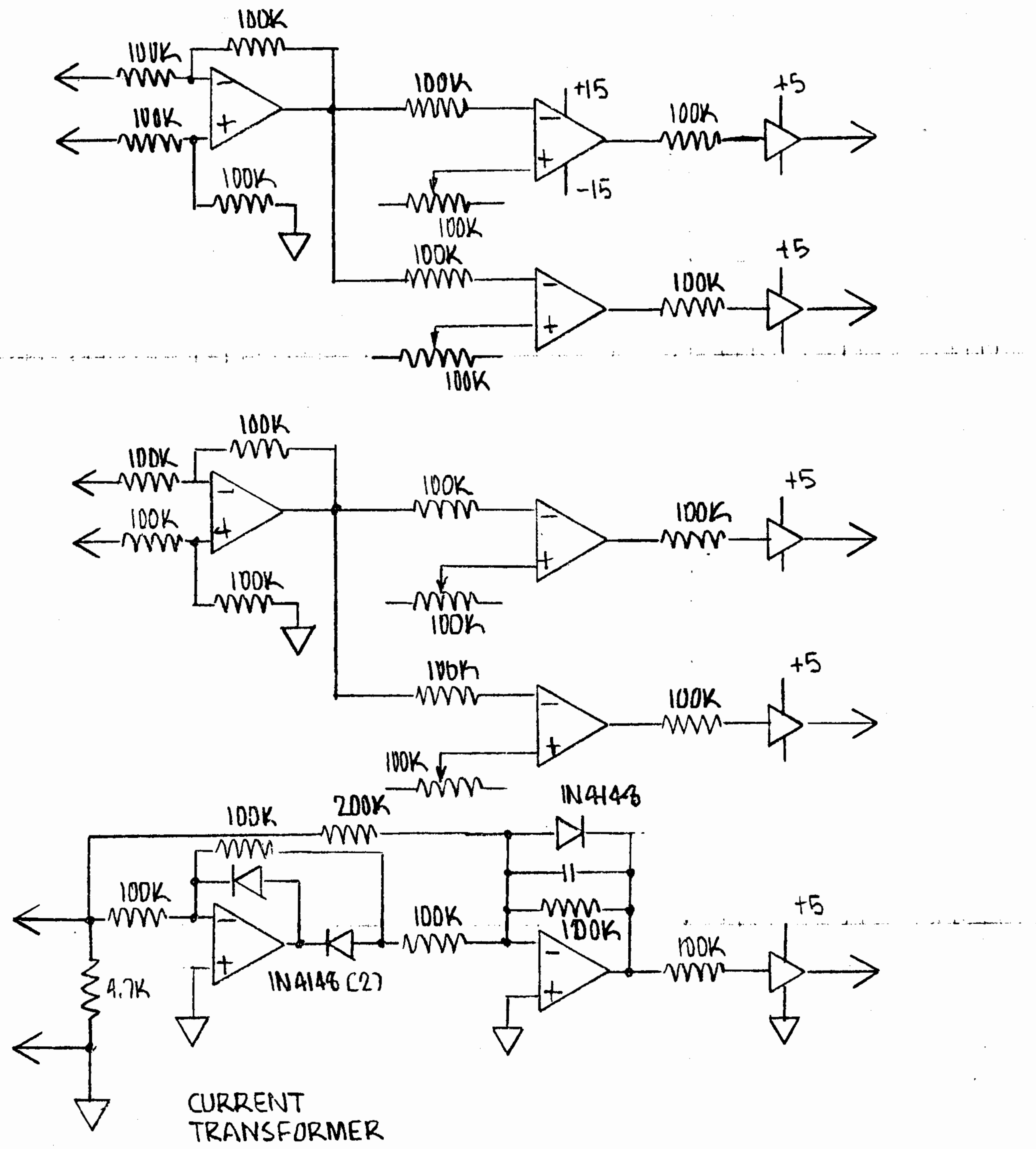
TP #30



MC 14050B
ground pin 11, 14

+5

+15
-15
COM.



NOTE:
ALL RESISTORS - 1/4W
OP. AMPS 747
BUFFERS MC 14050B

PORT OF SEATTLE COMMISSION

DESIGN NO.

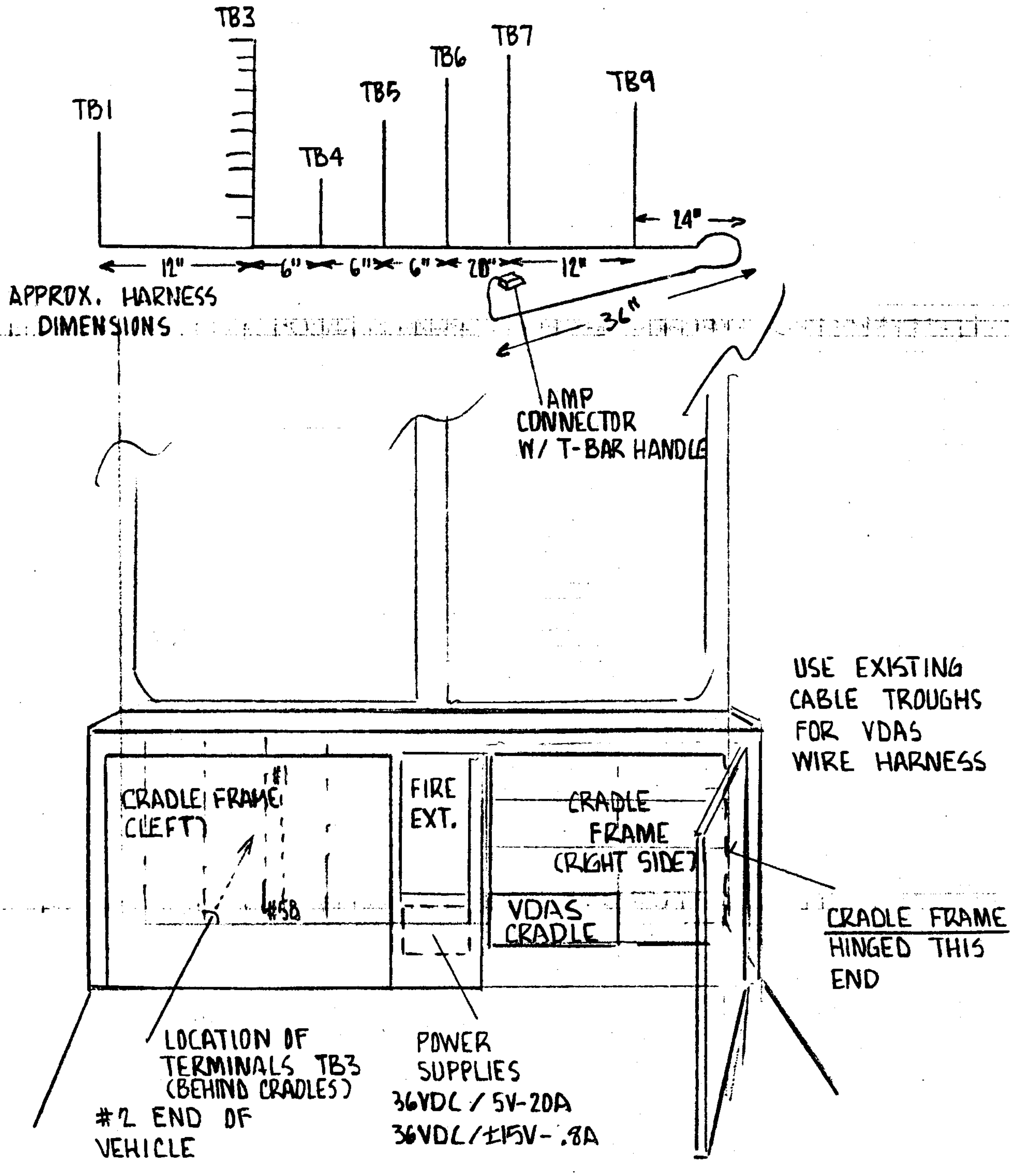
PROJECT STS - VDAS

TITLE SPARE CHANNELS (CONTROL CARD)

DRAWING NO.

DRAWN BY P. STUTZ DATE 2/16/78 APPROVED

POS 120-9



PORT OF SEATTLE COMMISSION

DESIGN NO.

PROJECT STS - VDAS

TITLE LOCATION OF VDAS IN VEHICLE (#2 END)

DRAWN BY P. STUTZ DATE 11-29-77 APPROVED

DRAWING NO. SK T711-1 (6)

POS 120-9

TESTPOINT #		TERMINAL #		CRADLE CONN.	
1		TB5-46		J1-1	
2		TB4-54		J1-2	
3		TB4-55		J1-3	
4		TB3-1		J1-4	
5		TB5-21		J1-5	
6		TB5-8		J1-34	
7		TB7-3		J1-6	
8		TB1-34		J1-30	
9		TB3-12		J1-31	
10		TB3-14		J1-35	
11		TB5-19		J1-36	
12		TB3-3		J1-7	
13		TB3-4		J1-8	
14		TB3-5		J1-9	
15		TB3-6		J1-10	
16		TB5-37		J1-68	*
	(B/P COM)	TB5-38		J1-69	
17		TB3-15		J1-70	*
	(B/P COM)	TB5-38		J1-71	
18		TB4-11		J1-73	*
	(T/C COM)	TB4-12		J1-74	
19		TB1-11		J1-75	*
		TB5-38		J1-76	
20		TB1-53		J1-42	*
		TB1-51		J1-43	
21		TB1-55		J1-44	*
		TB1-57		J1-45	
22		TB3-8		J1-11	
23		TB3-16		J1-57	
24		TB4-7		J1-32	
25		TB3-18		J1-47	*
[illegible]		TB3-19		J1-48	
26		TB9-39		J1-12	
27		TB6-44		J1-13	
28		TB5-24		J1-14	
29		TB6-45		J1-15	
30		TB3-21		J1-58	
31		TB3-24	COM TB3-23	J1-61	COM J1-64
32		TB3-25		J1-62	
33		TB3-26		J1-63	
34		TB3-28		J1-60	

* SHIELDED PAIR
SHIELD ONLY
TERMINATED
AT TB3

PORT OF SEATTLE COMMISSION

PROJECT: STS - VDAS - CAR WIRING

TITLE: TERMINAL TO VDAS WIRING - TESTPOINTS

DRAWN BY: P. STUTZ DATE: 11-28-77 APPROVED:

DESIGN NO.

DRAWING NO. SK 7711-26

POS 120-9

DESIGNATION	TERMINAL	CRADLE CONN.
GND	TB3-54	J1-39
	-55	J1-40
	-56	J1-52
	-57	J1-54
	-58	J1-55
+5VDC	-46	J1-50
	-47	J1-51
	-48	J1-53
	-49	J1-66
	-50	J1-67
-15VDC	-45	J1-95
	-44	J1-96
	-43	J1-97
COM	-42	J1-98
	-41	J1-99
+15VDC	-40	J1-100
	-39	J1-101
	TB3-38	J1-102

PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS - CAR WIRING

TITLE TERMINAL TO VDAS WIRING - POWER

DRAWN BY P. STUTZ DATE 11-28-77 APPROVED

DESIGN NO.

DRAWING NO. SK 7711-3(6)

POS 120-9

TERMINAL TB3-	CRADLE CONNECTOR J1-	
2	19	
7	20	
9	21	
10	22	
11	23	
13	25	
15	26	
17	27	
20	80	
22	81	
27	82	
28	83	
29	84	
30	85	
31	86	
32	87	
33	88	
34	89	
35	90	
36	91	
37	92	
38	93	
39	94	
40	95	
56	41	GND
57	28	GND
58	29	GND
46	37	+5VDC
47	38	+5VDC
48	49	+5VDC

PORT OF SEATTLE COMMISSION

DESIGN NO.

PROJECT STS - VDAS - CAR WIRING

TITLE TERMINAL TO VDAS WIRING - SPARES, ETC.

DRAWN BY P. STUTZ DATE 11-24-77 APPROVED

DRAWING NO. SK 7711-4 (6

POS 120-9

VEHICLE TERMINAL BLOCK ASSIGNMENT

– #1 END –

TB 3

TESTPOINT DESIGNATION

NOTE: "S" – SPARE

		TB 3	Testpoint	Testpoint	
J1-4	J-19	1	4	"S"	SPARE
J1-7	J1-8		12	13	
J1-9	J1-10		14	15	
J1-20	J1-11		"S"	22	SPARE
J1-21	J1-22	10	"S"	"S"	SPARE
J1-23	J1-31		"S"	9	"
J1-25	J1-35		"S"	10	
J1-26	J1-57		"S"	23	
J1-27	J1-47		"S"	25	
J1-48	J1-80	20	25	"S"	(COM) SPARE
J1-58	J1-81		30	"S"	SPARE
J1-64	J1-61		31/32/33	31	(COM)
J1-62	J1-63		32	33	
J1-82	J1-60		"S"	34	SPARE
J1-83	J1-84	30			
J1-85	J1-86				
J1-87	J1-88				SPARES
J1-89	J1-90				
J1-91	J1-102				
J1-101	J1-100	40			+15VDC
J1-99	J1-98				COM
J1-97	J1-96				-15VDC
J1-95	J1-50				
J1-51	J1-53				+5VDC
J1-66	J1-67	50			
N.C.	N.C.				NOT CONNECTED
N.C.	J1-39				
J1-40	J1-52				GND
J1-54	J1-55	58			

PORT OF SEATTLE COMMISSION

PROJECT STS – VDAS –

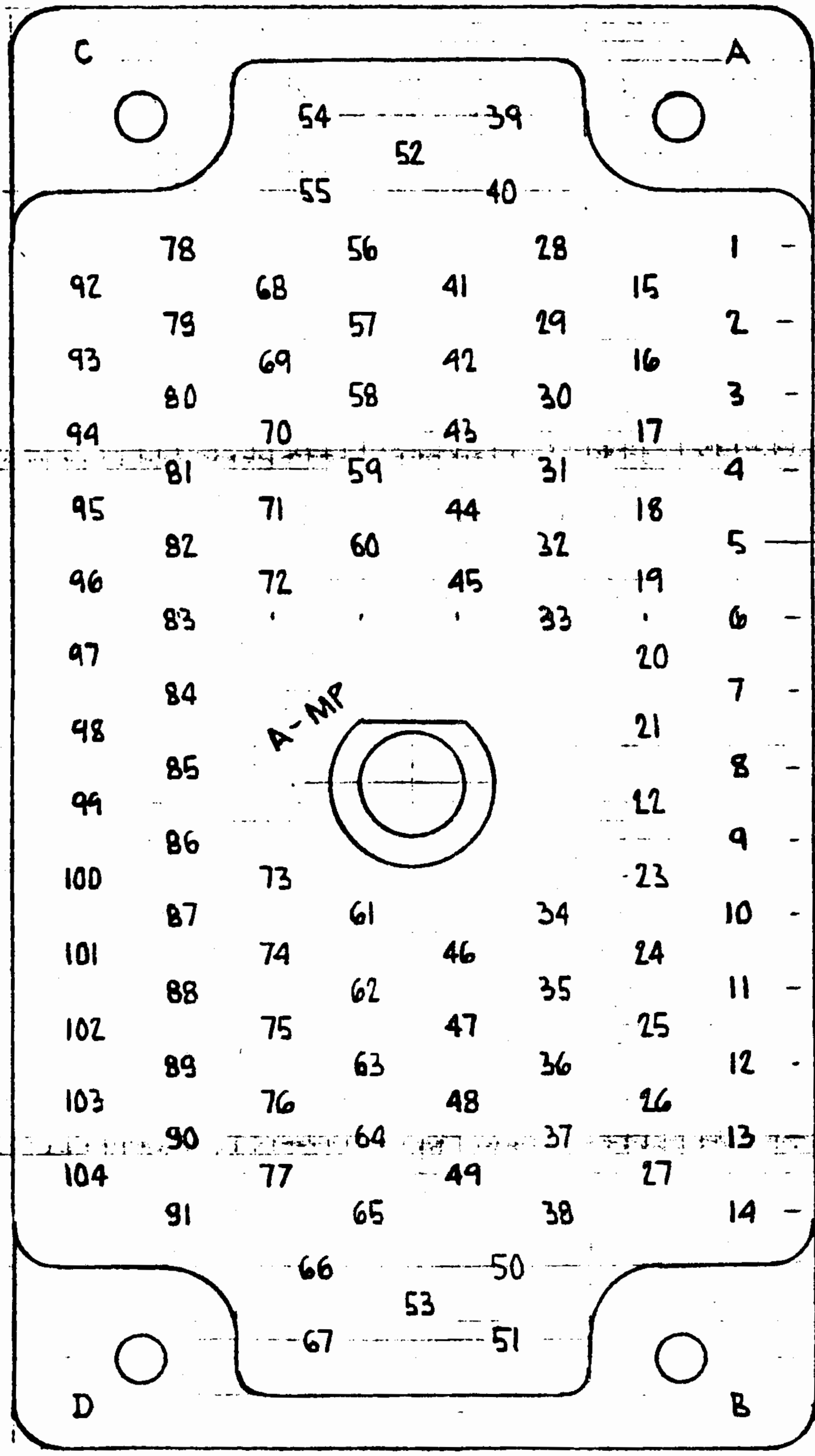
TITLE VEHICLE TERMINAL BLOCK ASSIGNMENT

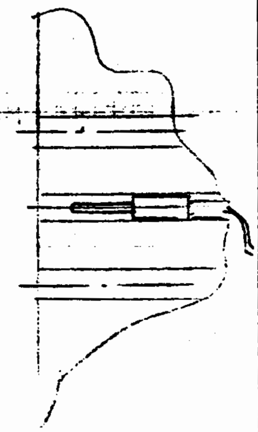
DRAWN BY P. STUTZ DATE 11-28-77 APPROVED

DESIGN NO.

DRAWING NO. SK 7711-5 (6)

POS 120-9



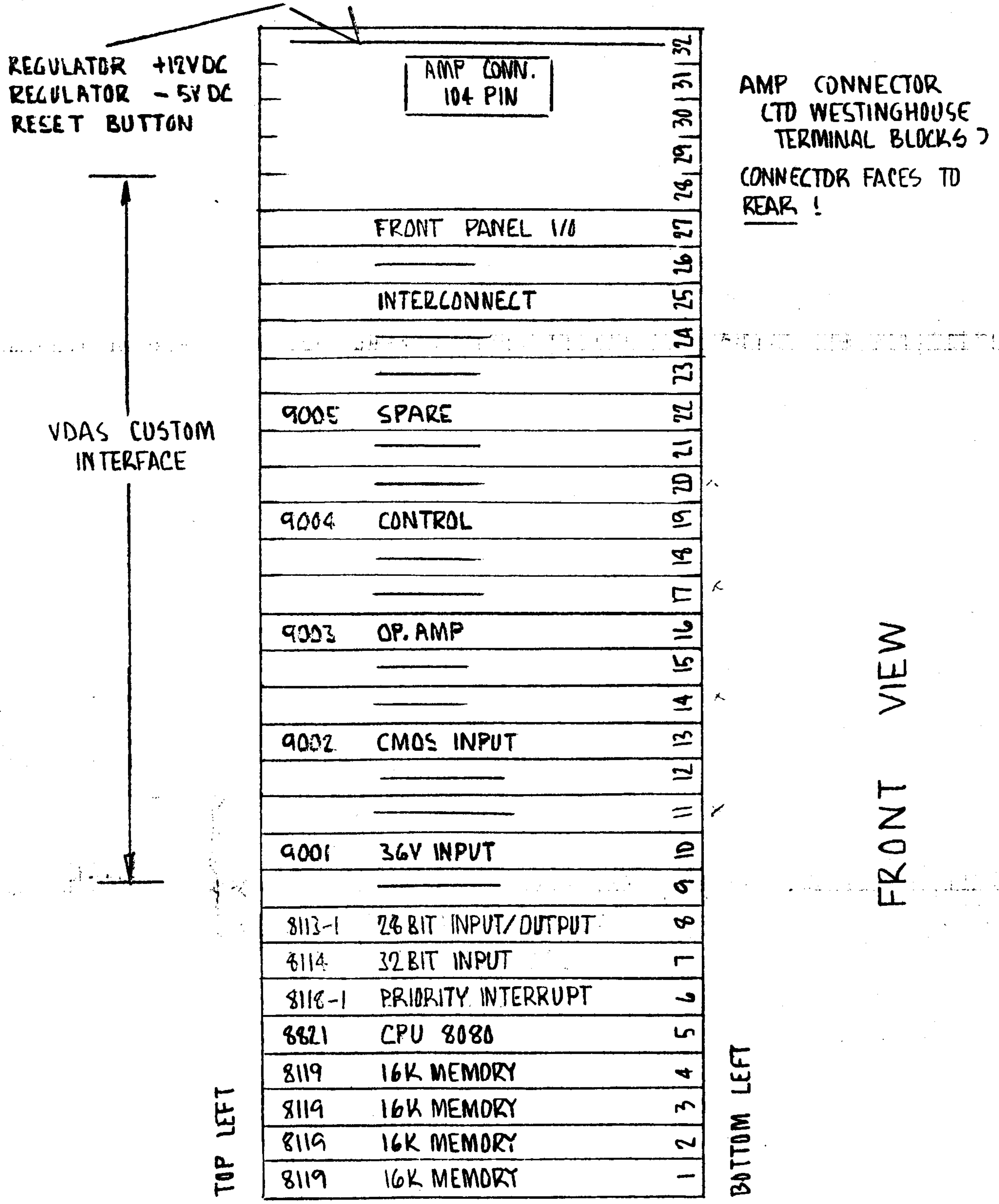


VIEW TOWARDS CRADLE CONNECTOR PINS.

CRADLE WIRING TO BE AT REAR, WIRE WRAP

PORT OF SEATTLE COMMISSION		DESIGN NO.
PROJECT	STS - VDAS - CRADLE CHASSIS CONNECTOR	
TITLE	PIN NUMBERS	DRAWING NO.
DRAWN BY	P. STUTZ DATE 11-22-77 APPROVED	

POS 120-9



PORT OF SEATTLE COMMISSION

PROJECT STS - VDAS - VEHICLE CARD CAGE

TITLE ASSEMBLY DETAILS FRONT VIEW

DRAWN BY P. STUTZ DATE 11-16-77 APPROVED

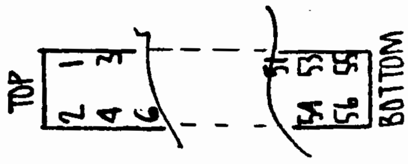
DESIGN NO.

DRAWING NO.

POS 120-9

1	8119	16K MEMORY
2	8119	16K MEMORY
3	8119	16K MEMORY
4	8119	16K MEMORY
5	8821	8080 CPU
6	8118	PRIORITY INTERRUPT
7	8114	32 BIT INPUT
8	8113-1	24 BIT INPUT/OUTPUT
9		—
10	9001	36V INPUT
11		—
12		—
13	9002	CMOS INPUT
14		—
15		—
16	9003	OP. AMP
17		—
18		—
19	9004	CONTROL
20		—
21		—
22	9005	SPARE
23		—
24		—
25		INTERCONNECT
26		—
27		FRONT PANEL I/O
28–32		FRONT PANEL FUSES INDICATOR LIGHTS PUSH BUTTONS

AMP CONN.



SOCKET PIN NUMBER ASSIGNMENT

REAR VIEW (TYPICAL)

REAR VIEW

WIRE WRAP PLANE

PORT OF SEATTLE COMMISSION

PROJECT: STS - VDAS - VEHICLE CARD CAGE

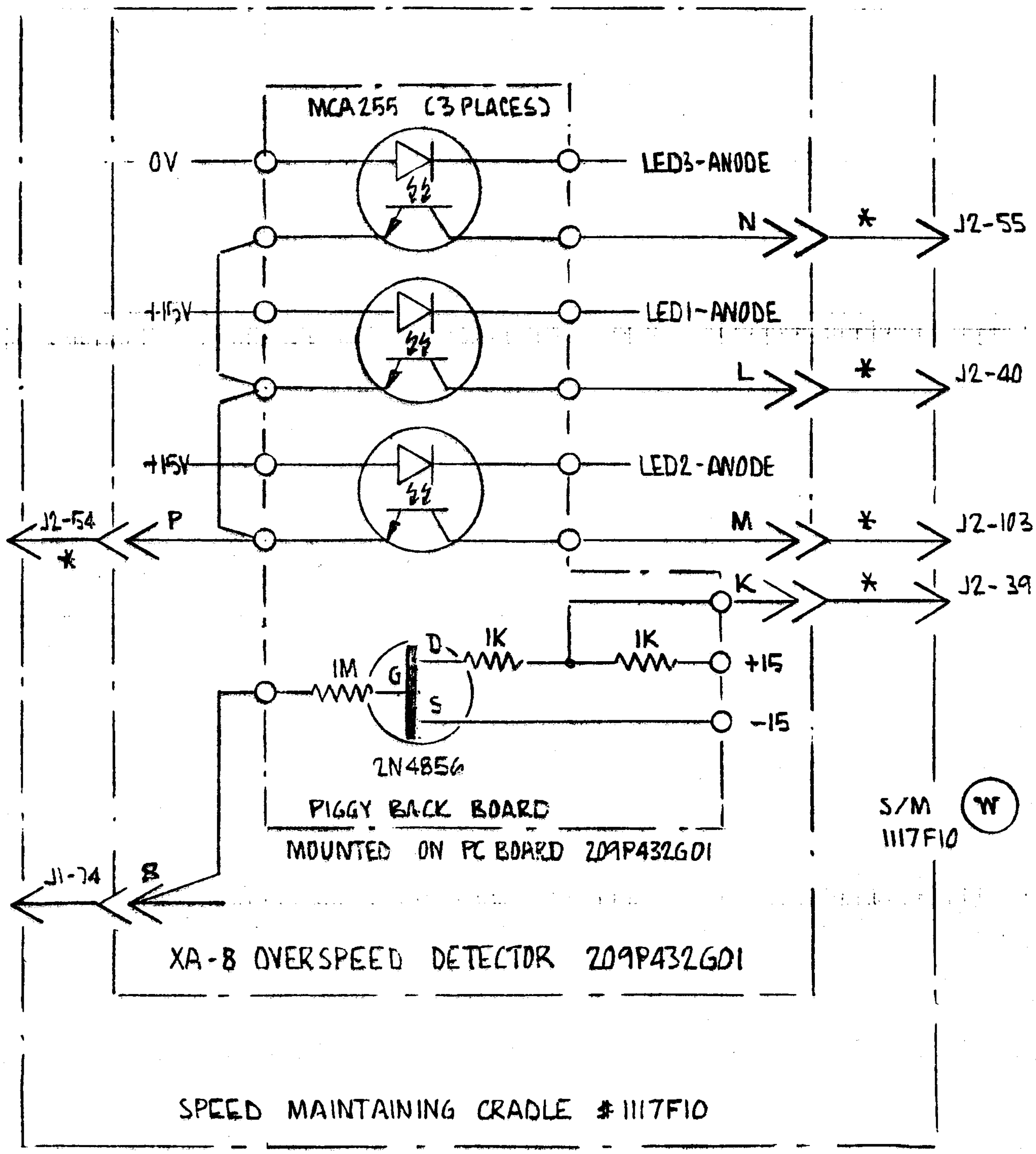
TITLE: ASSEMBLY DETAILS - REAR VIEW

DRAWN BY: P. STUTZ DATE: 11-16-77 APPROVED:

DESIGN NO.

DRAWING NO.

POS 120-9



NOTE:
* - ADD TO CRADLE WIRING

REF: TESTPOINT DEFINITIONS-REV A -12/9/77

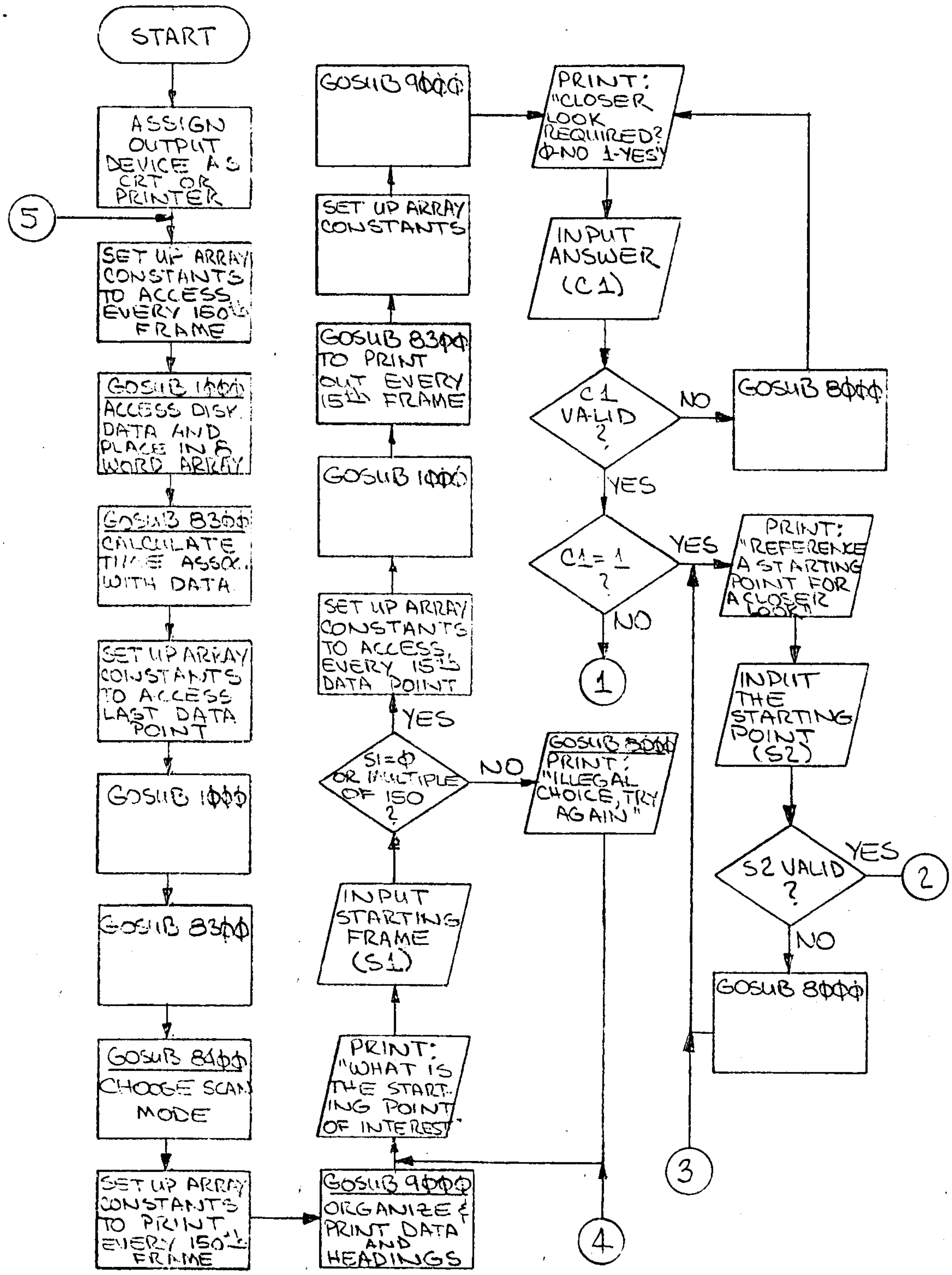
PORT OF SEATTLE COMMISSION	DESIGN NO.
PROJECT STS-VDAS	
TITLE PIGGY BACK BOARD TO OVERSPD. DET #209P432G01	DRAWING NO.
DRAWN BY P. STUTZ DATE 2/7/78 APPROVED	

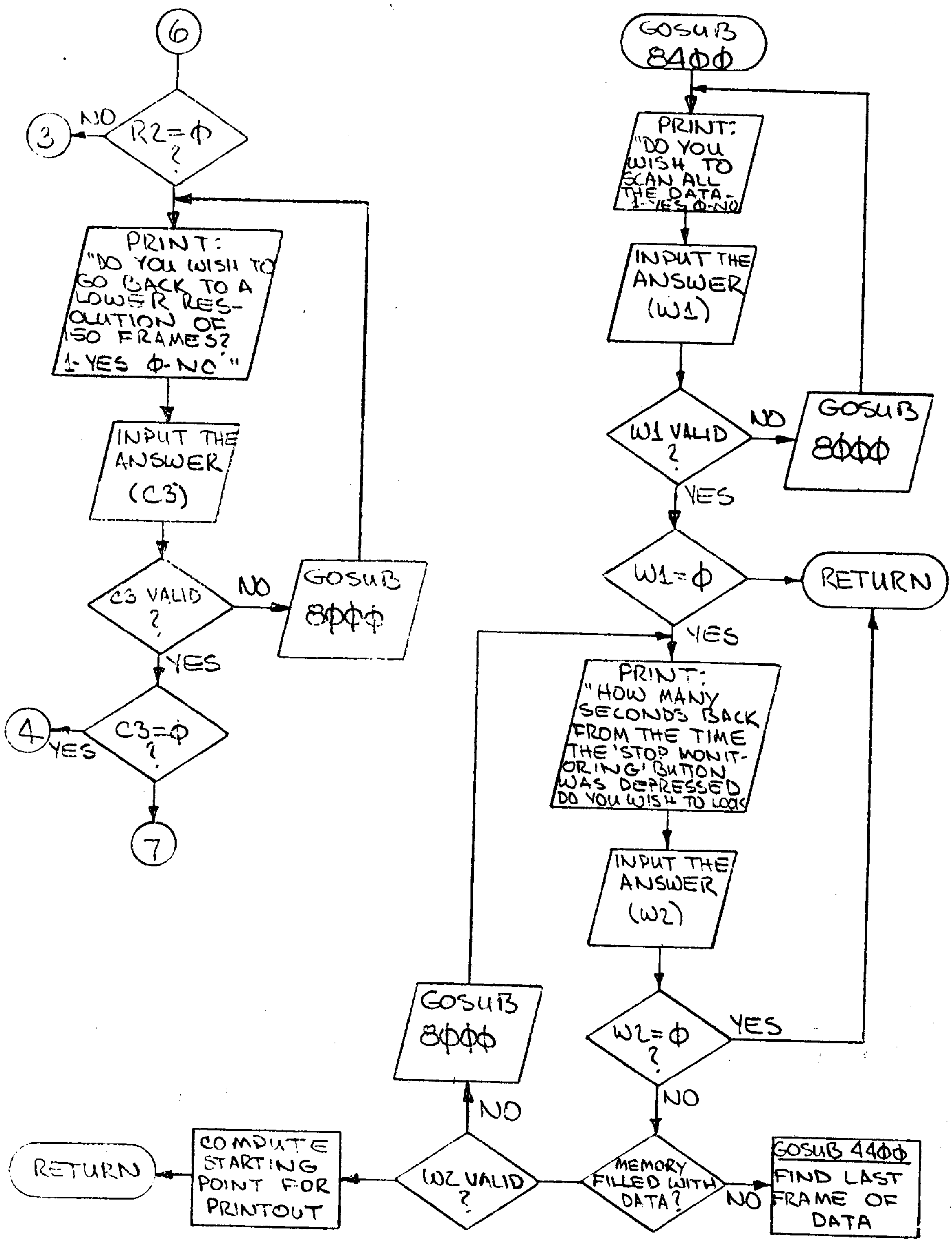
POS 120-9

APPENDIX C

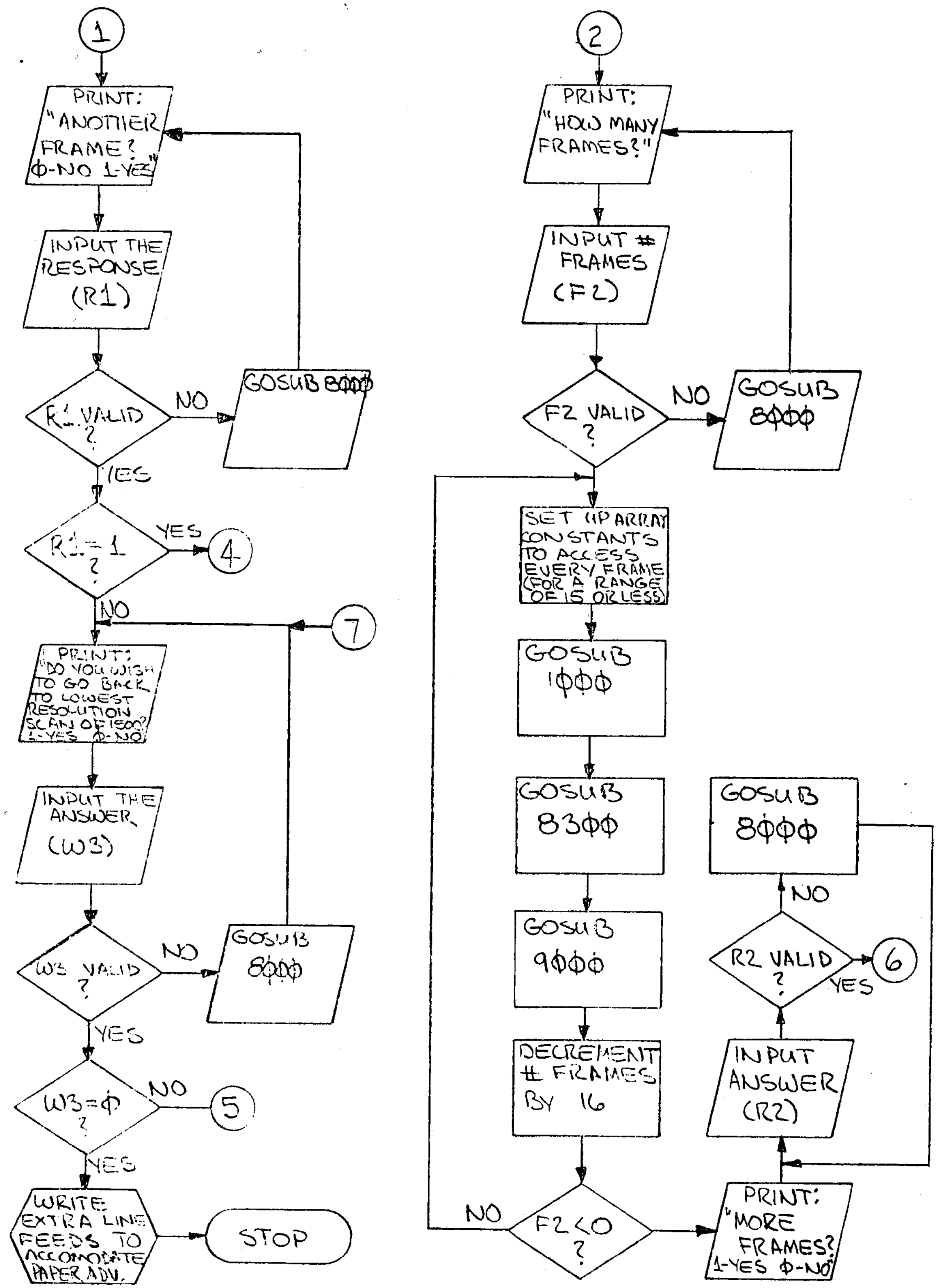
VDAS Data Handler & Data Analysis Software Listings

DATA ANALYSIS PROGRAM

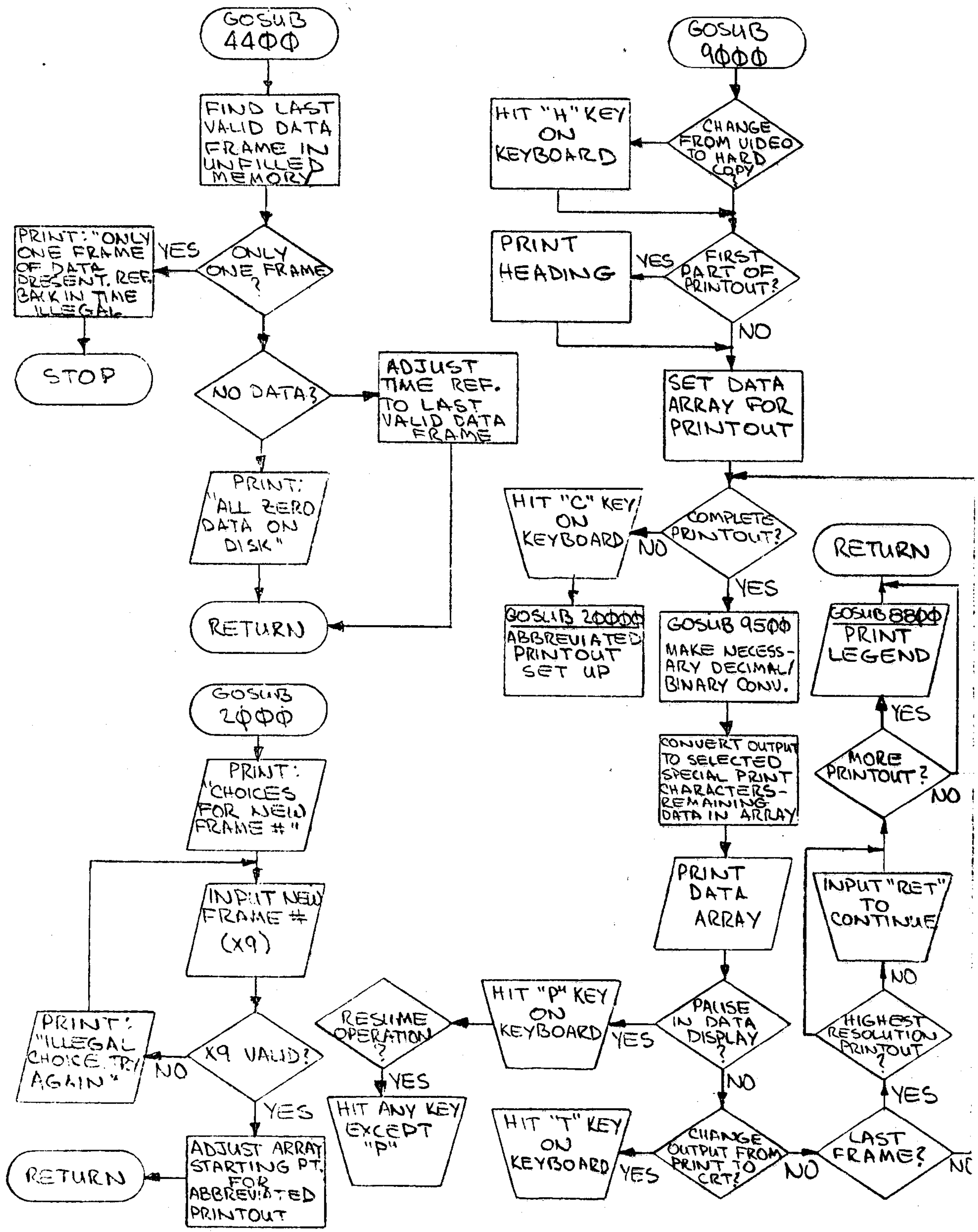




6
R2=Φ ?
NO
3
PRINT: "DO YOU WISH TO GO BACK TO A LOWER RES-OLUTION OF 150 FRAMES? 1-YES Φ-NO"
INPUT THE ANSWER (C3)
C3 VALID ?
NO
GOSUB 8ΦΦΦ
YES
C3=Φ ?
YES
4
7
GOSUB 84ΦΦ
PRINT: "DO YOU WISH TO SCAN ALL THE DATA- 1-YES Φ-NO"
INPUT THE ANSWER (W1)
W1 VALID ?
NO
GOSUB 8ΦΦΦ
YES
W1=Φ
RETURN
YES
PRINT: "HOW MANY SECONDS BACK FROM THE TIME THE 'STOP MONIT-ORING' BUTTON WAS DEPRESSED DO YOU WISH TO LOOK
INPUT THE ANSWER (W2)
W2=Φ ?
YES
NO
GOSUB 8ΦΦΦ
NO
MEMORY FILLED WITH DATA?
NO
GOSUB 44ΦΦ FIND LAST FRAME OF DATA
W2 VALID ?
COMPUTE STARTING POINT FOR PRINTOUT
RETURN



1
PRINT: "ANOTHER FRAME? Φ-NO 1-YES"
INPUT THE RESPONSE (R1)
R1 VALID ?
NO
GOSUB 8ΦΦΦ
YES
R1 = 1 ?
YES
4
NO
7
PRINT: "DO YOU WISH TO GO BACK TO LOWEST RESOLUTION SCAN OF 1500? 1-YES Φ-NO"
INPUT THE ANSWER (W3)
W3 VALID ?
NO
GOSUB 8ΦΦΦ
YES
W3=Φ ?
NO
5
YES
WRITE EXTRA LINE FEEDS TO ACCOMODATE PAPER ADV.
STOP
2
PRINT: "HOW MANY FRAMES?"
INPUT # FRAMES (F2)
F2 VALID ?
NO
GOSUB 8ΦΦΦ
SET UP ARRAY CONSTANTS TO ACCESS EVERY FRAME (FOR A RANGE OF 15 OR LESS)
GOSUB 1ΦΦΦ
GOSUB 83ΦΦ
GOSUB 9ΦΦΦ
DECREMENT # FRAMES BY 16
F2<0 ?
NO
PRINT: "MORE FRAMES? 1-YES Φ-NO"
INPUT ANSWER (R2)
R2 VALID ?
YES
6
NO
GOSUB 8ΦΦΦ


GOSUB 44ΦΦ
FIND LAST VALID DATA FRAME IN UNFILLED MEMORY
ONLY ONE FRAME ?
YES
PRINT: "ONLY ONE FRAME OF DATA PRESENT. REF. BACK IN TIME ILLEGAL"
STOP
NO DATA?
ADJUST TIME REF. TO LAST VALID DATA FRAME
PRINT: "ALL ZERO DATA ON DISK"
RETURN
GOSUB 2ΦΦΦ
PRINT: "CHOICES FOR NEW FRAME #"
INPUT NEW FRAME # (X9)
X9 VALID?
NO
PRINT: "ILLEGAL CHOICE. TRY AGAIN"
YES
ADJUST ARRAY STARTING PT. FOR ABBREVIATED PRINTOUT
RETURN
GOSUB 9ΦΦΦ
CHANGE FROM VIDEO TO HARD COPY
HIT "H" KEY ON KEYBOARD
FIRST PART OF PRINTOUT?
YES
PRINT HEADING
NO
SET DATA ARRAY FOR PRINTOUT
COMPLETE PRINTOUT?
NO
HIT "C" KEY ON KEYBOARD
GOSUB 2ΦΦΦΦ
ABBREVIATED PRINTOUT SET UP
YES
GOSUB 95ΦΦ
MAKE NECESSARY DECIMAL/BINARY CONV.
CONVERT OUTPUT TO SELECTED SPECIAL PRINT CHARACTERS-REMAINING DATA IN ARRAY
PRINT DATA ARRAY
PAUSE IN DATA DISPLAY ?
YES
HIT "P" KEY ON KEYBOARD
RESUME OPERATION ?
YES
HIT ANY KEY EXCEPT "P"
NO
CHANGE OUTPUT FROM PRINT TO CRT?
YES
HIT "T" KEY ON KEYBOARD
NO
LAST FRAME?
NO
YES
HIGHEST RESOLUTION PRINTOUT ?
NO
INPUT "RET" TO CONTINUE
MORE PRINTOUT ?
YES
NO
GOSUB 88ΦΦ
PRINT LEGEND
RETURN

```
1 REM... THIS PROGRAM PUTS TRAIN DATA INTO
2 REM... A MEANINGFUL FORMAT FOR ANALYSIS
3 REM...1 JUNE 78
5 REM...D1 IS THE DATA DISPLAY VARIABLE
6 INPUT"TYPE 1 FOR HARD COPY 0 FOR CRT? ",D1
7    REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
8 IF D1<0 OR D1>1 THEN GOSUB 8000\IF D1<0 OR D1>1 THEN 6
9 LINE #0,80 \ LINE #1,80
10 REM...INITIALIZE ARRAY STARTING POINTER
11 K1=0
15 DIM D(50,8),O(50),P$(20),P(40),N(50),T(50)
18 REM...D2 IS THE PROGRAMMING DECISION DISPLAY VARIABLE
19 D2=0
20 REM...ASSIGN DISPLAY VARIABLE VALUES AS FOLLOWS ,
21 REM...0-CRT,1-SERIAL OUTPUT,2-PARALLEL OUTPUT,
22 REM...3-CUSTOM OUTPUT
23 REM******INPUT EVERY 150TH FRAME OF DATA FROM DISK******
24 OPEN #0,"TDATA"
25 X1=1200
27 REM...SET ABBREVIATED DISPLAY VARIABLE FOR LOWEST RESOLUTION
28 S7=1
30 FOR I1=0 TO 56400 STEP X1
40 J=I1/X1
45 N(J)=I1/8
46 GOSUB 1000
48 REM...CALCULATE TIMES
49 GOSUB 8300
50 NEXT I1
53 I1=57336
55 J=48
56 N(J)=I1/8
62 GOSUB 1000
63 REM...CALCULATE TIMES
65 CLOSE #0
67 GOSUB 8300
68 REM REM...CHOOSE SCAN MODE
69 GOSUB 8400
70 L1=48
72 L2=1
73 REM....IF FULL SCAN IS DESIRED,SET ARRAY ACCORDINGLY
74 IF K1=0 THEN B1=0
75 REM***PRINT OUT TEST POINT DATA AT EVERY 150TH FRAME***
80 GOSUB 9000
90 REM... NOW,TAKE A CLOSER LOOK AT THE DATA
100 REM...FIRST,PICK A STARTING POINT
110 ! #D2,"WHAT IS THE STARTING POINT OF INTEREST?"
120 ! #D2,"...PICK A DATA POINT NUMBER OFF PRINTOUT"
121 ! #D2,"VALID CHOICES ARE:0,150,300,450,600,750,900,1050,"
122 ! #D2,"1200,1350,1500,1650,1800,1950,2100,2250,2400,2550,"
123 ! #D2,"2700,2850,3000,3150,3300,3450,3600,3750,3900,4050,"
124 ! #D2,"4200,4350,4500,4650,4800,4950,5100,5250,5400,5550,"
125 ! #D2,"5700,5850,6000,6150,6300,6450,6600,6750,6900,&7050"
126 ! #D2,
130 INPUT S1
135 U1=S1
136 REM...CHECK FRAME NUMBER FOR VALIDITY
138 IF U1=0 THEN 160
139 REM...IF TOO LARGE,SAY SO,GO BACK & GET A NEW NUMBER
140 IF U1>7050 THEN GOSUB 8000
141 IF U1>7050 THEN 100
142 U1=U1-150
144 IF U1=0 THEN 160
145 REM..IF NO MULTIPLE OF 150,SAY SO,GO BACK& GET ONE THAT IS
146 IF U1<0 THEN GOSUB 8000
148 IF U1<0 THEN 100
```

```
149 GOTO 142
150 GOSUB 8000
155 GOTO 100
160 REM***** NOW,INPUT EVERY 15TH FRAME OF DATA FROM DISK****
170 OPEN #0,"TDATA"
175 S0=S1*8
177 REM...SET ABBREVIATED PRINT VARIABLE FOR MEDIUM RESOLUTION
178 S7=2
180 IF S0>=56400 THEN X4=935
190 IF S0<56400 THEN X4=1200
195 X2=120
200 FOR I1=S0 TO S0+X4 STEP X2
205 J=(I1-S0)/X2
210 N(J)=I1/8
220 GOSUB 1000
221 REM...CALCULATE TIMES
222 GOSUB 8300
223 NEXT I1
226 CLOSE #0
227 IF X4>1100 THEN X5=10
230 IF X4<1100 THEN X5=7
335 L1=X5
338 L2=1
340 B1=0
355 REM***PRINT OUT TEST POINT DATA FOR EVERY 15TH DATA POINT***
360 GOSUB 9000
370 ! #D2,
400 ! #D2,"CLOSER LOOK REQUIRED?"
410 ! #D2,"0-NO,1-YES"
420 INPUT C1
425 IF C1=0 OR C1=1 THEN 440
428  REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
430 GOSUB 8000
435 GOTO 400
438 REM...IF CLOSER LOOK NOT WANTED,CONSIDER MORE PRINTOUT...
439 REM......AT THIS RESOLUTION.
440 IF C1=0 THEN PRINT #D2,"ANOTHER FRAME?"
450 IF C1=0 THEN PRINT #D2,"0-NO,1-YES"
452 REM...IF CLOSER LOOK WANTED,CONTINUE ON.....
455 IF C1=1 THEN 500
465 INPUT R1
470 IF R1=1 OR R1=0 THEN 480
472  REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
474 GOSUB 8000
477 GOTO 440
478 REM...IF MORE PRINTOUT NOT WANTED AT THIS RESOLUTION...
479 REM......CONSIDER A LOWER RESOLUTION PRINTOUT
480 IF R1=0 THEN 730
482 REM...IF MORE PRINTOUT WANTED AT THIS RESOLUTION....
483 REM.........GO BACK AND GET IT.
490 IF R1=1 THEN 100
500 IF X4<1100 THEN X8=(X4+1)/8
502 IF X4>1100 THEN X8=X4/8
504 REM...INITIALIZE HEADING & LEGEND PRINT VARIABLE
505 K4=1
509 ! #D2,"REFERENCE A STARTING POINT FOR A CLOSER LOOK"
512 INPUT S2
514 IF S2<7153 THEN L3=15
515 IF S2>=7153 THEN L3=7167-S2
523 S4=S2*8
525 IF S2<0 OR S2>7167 THEN 540
528 IF K4>1 THEN 552
530 ! #D2,"HOW MANY FRAMES?"
531 ! #D2,"(IF STARTING FRAME>=7152,NUMBER OF FRAMES"
532 ! #D2,"WILL BE PREDETERMINED AND RUN THRU LAST FRAME."
```

```
533 IF S2>=7152 THEN 552
534 INPUT F2
535 IF S2+F2>7167 THEN ! #D2,"TOO MANY FRAMES FOR SELECTED"
536 IF S2+F2>7167 THEN ! #D2,"STARTING POINT."
537 IF S2+F2<=7167 THEN 552
538 REM...IF TOO MANY FRAMES OR ILLEGAL STARTING POINT SELECTED.
539 REM.....SAY SO,GO BACK AND GET VALID INFORMATION
540 GOSUB 8000
550 GOTO 500
551 REM...SET ARRAY LENGTH AND SAMPLE RATE
552 IF S4<57224 THEN X7=120
553 IF S4>=57224 THEN X7=57336-S4
554 IF S4=57216 THEN F2=15
556 IF S4>=57224 THEN F2=(57336-S4)/8
557 K4=K4+1
558 X3=8
559 B1=0
560 REM...IF LESS THAN 15 FRAMES REMAIN,SET ARRAY ACCORDINGLY.
561 L1=L3\IF F2<15 THEN L1=F2
562 L2=1
563 REM*******INPUT EACH FRAME OF DATA FROM DISK*******
564 OPEN #0,"TDATA"
565 FOR I1=S4 TO S4+X7 STEP X3
567 J=(I1-S4)/X3
569 N(J)=I1/8
572 GOSUB 1000
575 REM...CALCULATE TIMES
576 GOSUB 8300
580 NEXT I1
590 CLOSE #0
595 REM...SET ABBREVIATED PRINT DISPLAY FOR HIGHEST RESOLUTION
598 S7=3
599 REM******PRINT OUT TEST POINT DATA FOR EACH FRAME*******
600 GOSUB 9000
605 REM...ADJUST NUMBER OF FRAMES AND STARTING POINT.....
606 REM......TO PRINT OUT MORE DATA
610 F2=F2-16
611 S2=S2+16
612 REM...IF FRAMES ALL PRINTED,CONTINUE ON.........
614 IF F2=0 THEN 620
615 REM... IF NOT,GO BACK AND DO MORE
618 IF F2>0 THEN 514
620 ! #D2,"MORE FRAMES?"
630 ! #D2,"0-NO,1-YES"
640 INPUT R2
645 IF R2=0 OR R2=1 THEN GOTO 680
648  REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
650 GOSUB 8000
655 GOTO 620
670 REM...IF MORE FRAMES ARE TO BE PRINTED,GO BACK AND DO IT
680 IF R2=1 THEN 500
685 REM...IF NOT,CONSIDER LOWER RESOLUTION SCAN
690 ! #D2,"DO YOU WISH TO GO BACK TO A LOWER RESOLUTION SCAN"
695 ! #D2,"OF 15 FRAMES?"
700 ! #D2,"1-YES,0-NO"
705 INPUT C3
707  REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
710 IF C3<0 OR C3>1 THEN GOSUB 8000
712 IF C3<0 OR C3>1 THEN 690
715 REM...IF LOWER RESOLUTION SCAN WANTED,GO BACK......
720 IF C3=1 THEN 100
725 REM...IF NOT,CONSIDER LOWEST RESOLUTION SCAN
730 IF C3=0 THEN ! #D2,"DO YOU WISH TO GO BACK TO LOWEST"
735 IF C3=0 THEN ! #D2,"RESOLUTION SCAN OF 150 FRAMES."
740 IF C3=0 THEN ! #D2,"1-YES,0-NO"
```

```
750 IF CS=0 THEN INPUT W3
752  REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
755 IF W3<0 OR W3>1 THEN GOSUB 8000
757 IF W3<0 OR W3>1 THEN 730
760 IF W3=1 THEN GOTO 24
995 REM...DO LINE FEEDS BEFORE EXITING PROGRAM
999 GOTO 30000
1000 REM...READ & DISPLAY INCOMING DATA
1010 READ #0 %I1,&D(J,1),&D(J,2),&D(J,3),&D(J,4)
1020 READ #0 %I1+4,&D(J,5),&D(J,6),&D(J,7),&D(J,8)
1030 ! #D2,
1040 ! #D2,D(J,1),D(J,2),D(J,3),D(J,4),D(J,5),D(J,6),D(J,7),D(J,8)
1050 RETURN
4400 REM******NON-WRAPAROUND ROUTINE******
4401 REM...SET NON-WRAPAROUND VARIABLE
4402 K6=1
4405 FOR J=48 TO 0 STEP -1
4410 IF D(J,1)+ D(J,2)+D(J,3)+ D(J,4)<>0 THEN 4416
4415 IF D(J,5)+ D(J,6)+ D(J,7)+ D(J,8)=0 THEN 4425
4416 IF J=0 THEN ! " CONTINUE PROGRAM BY TYPING 'CONT'"
4417 IF J=0 THEN ! "ONLY ONE FRAME OF DATA PRESENT !"
4418 IF J=0 THEN ! "REFERENCE BACK IN TIME ILLEGAL !"
4419 IF J=0 THEN K5=1
4420 IF J=0 THEN STOP
4421 IF J=0 THEN RETURN
4422 IF J=0 THEN RETURN
4423 T(48)=T(J)
4424 RETURN
4425 NEXT J
4430 ! #D2,"ALL ZERO DATA ON DISK"
4435 RETURN
8000 REM******INVALID ENTRY ROUTINE******
8005 ! #D2,
8010 ! #D2,"ILLEGAL CHOICE,TRY AGAIN"
8015 ! #D2,
8020 RETURN
8300 REM******TIME CALCULATION******
8310 T1=256*D(J,1)
8320 T(J)=(T1+D(J,2))/10
8330 RETURN
8400 REM******SCAN MODE DECISION ROUTINE******
8410 ! #D2,"DO YOU WISH TO SCAN ALL THE DATA?"
8420 ! #D2,"1-YES,0-NO"
8430 INPUT W1
8435 REM..IF ANSWER ILLEGAL,SAY SO,GO BACK & GET VALID ANSWER
8440 IF W1<0 OR W1>1 THEN GOSUB 8000
8445 IF W1<0 OR W1>1 THEN 8400
8450 IF W1=0 THEN 8465
8455 REM...IF TOTAL SCAN REQUESTED,GO NO FURTHER HERE....
8460 RETURN
8462 REM......OTHERWISE,GET TIME REFERENCE.
8465 ! #D2,"HOW MANY SECONDS BACK FROM THE TIME"
8466 ! #D2,"THE 'STOP MONITOR' BUTTON WAS DEPRESSED"
8467 ! #D2,"DO YOU WISH TO LOOK?"
8468 ! #D2,
8469 K1=1\REM...SET PARTIAL SCAN VARIABLE
8470 INPUT W2
8471 IF W2=0 AND S2=2 THEN LET B1=15
8472 IF W2=0 AND S7=1THEN LET B1=48
8473 IF W2=0 THEN 8525
8474 REM..IF NO TIME ENTRY FOUND,ENTER NON-WRAPAROUND ROUTINE
8475 IF T(48)+T(47)=0 THEN GOSUB 4400
8477 REM...IF ONLY ONE FRAME OF DATA FOUND,GO NO FURTHER.....
8478 IF K5=1 THEN RETURN
8479 REM..IF OPERATOR ATTEMPTS TO GO BACK TOO FAR(IN TIME)....
```

```
8480 REM...SAY SO,GO BACK & GET VALID TIME
8483 IF T(48)-W2<T(0) THEN GOSUB 8000
8485 IF T(48)-W2<T(0) THEN 8465
8487 REM...COMPUTE TIME ADJUSTMENT
8490 T3=T(48)-W2
8492 REM.....TO FIND STARTING POINT FOR PRINTOUT.
8495 FOR J=0 TO 48
8500 IF T3<T(J) THEN 8510
8505 GOTO 8520
8508 REM...SET STARTING POINT
8510 B1=J-1
8511 REM..RESET LAST TIME ENTRY FOR NON-WRAPAROUND PRINTOUT
8512 IF K6=1 THEN T(48)=T(47)
8515 RETURN
8520 NEXT J
8525 RETURN
8800 REM...PRINT LEGEND
8810 ! #D1,TAB(10),"LEGEND"
8815 ! #D1,TAB(10),"------"
8820 ! #D1,TAB(10),"1-TRUE "
8830 ! #D1,TAB(10),"0-FALSE"
8840 ! #D1,TAB(10),"B-BRAKE"
8850 ! #D1,TAB(10),"P-PROPULSION"
8860 ! #D1,TAB(10),"U(OR X)-UNDEFINED"
8870 ! #D1,TAB(10),"TEST POINT 20 GIVEN IN MPH"
8880 RETURN
9000 REM**PRINT HEADING(HARDCOPY AVAILABLE BY HITTING 'H' KEY)**
9001 IF INP(252)=72 THEN D1=1
9002 IF S7=3 AND K4>2 THEN 9020
9003 ! #D1,TAB(35),"TEST DATA"
9004 ! #D1,TAB(33),"(TP 1 THRU 34)"
9005 ! #D1,
9006 ! #D1,"DATA TIME",TAB(31),1,1,1,TAB(38),1,TAB(41),1,TAB(44),1,
9008 ! #D1,TAB(48),2,TAB(52),2,2,2,2,2,2,2,2,2,
9010 ! #D1,TAB(70),3,3,3,3,3
9012 ! #D1," PNT (SEC)",TAB(12),1,2,3,4,5,6,7,
9013 ! #D1,TAB(27),8,9,
9014 ! #D1,TAB(31),0,1,6,TAB(38),7,TAB(41),8,TAB(44),9,
9016 ! #D1,TAB(48),0,TAB(52),1,2,3,4,5,6,7,8,9,
9018 ! #D1,TAB(70),0,1,2,3,4
9019 ! #D1,"---- -----"
9020 FOR J=B1 TO L1 STEP L2
9021 REM...IF ABBREVIATED PRINTOUT WANTED,HIT 'C' KEY
9022 IF INP(252)=67 THEN GOSUB 20000
9030 FOR K=1 TO 8
9035 REM...FIRST,MAKE NECESSARY DECIMAL-TO-BINARY CONVERSIONS
9040 IF K>2 AND K<8 THEN GOSUB 9500
9045 REM....THEN CONVERT SELECTED BINARY DATA TO LETTERS
9050 IF K=3 THEN GOSUB 9600
9060 IF K=4 THEN GOSUB 9650
9070 IF K=5 THEN GOSUB 9700
9080 IF K=6 THEN GOSUB 9750
9090 IF K=7 THEN GOSUB 9800
9100 IF K=8 THEN GOSUB 9850
9102 REM..GO TO NEXT WORD
9105 NEXT K
9110 REM******PRINT DATA & HEADINGS*******
9120 GOSUB 9900
9124 REM...IF PAUSE IS WANTED DURING PRINTOUT,HIT'P' KEY
9126 IF INP(252)=80 THEN 9126
9127 REM...IF SWITCH FROM PRINTER TO CRT IS NEEDED,HIT 'T' KEY
9128 IF INP(252)=84 THEN D1=0
9129 REM******THEN,GO TO NEXT FRAME ******
9130 NEXT J
9131 REM...SKIP PROGAMMED PAUSE IF HIGHEST RESOLUTION PRINTOUT
```

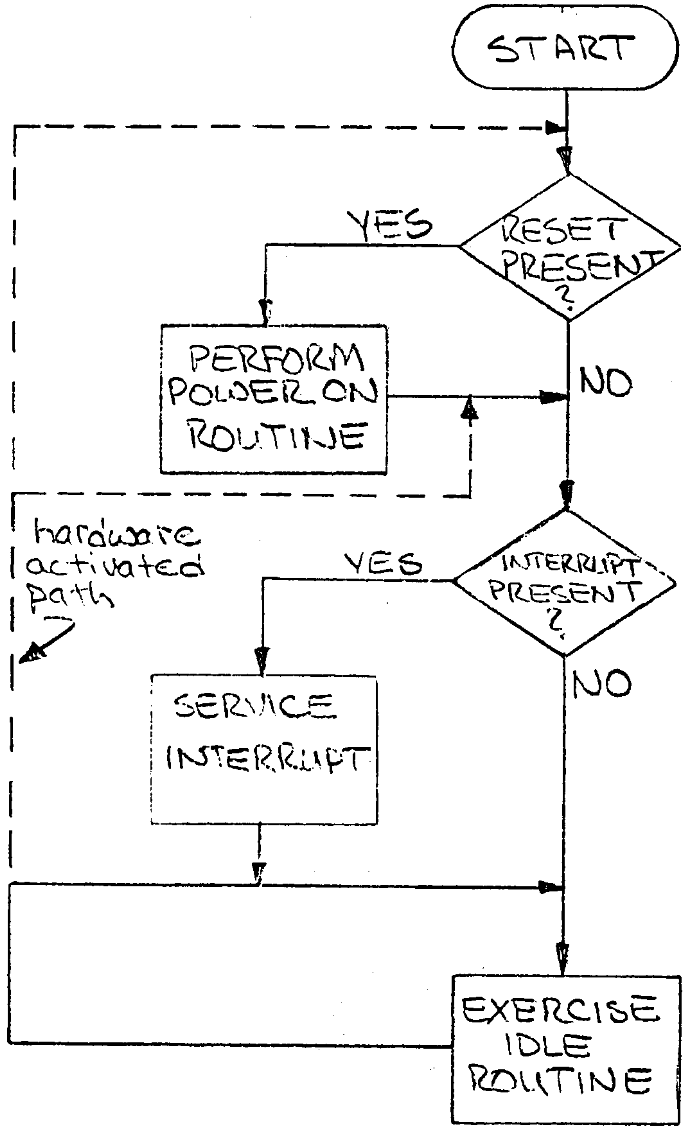
```
9132 IF S7=3 THEN 9138
9133 INPUT"HIT RETURN TO CONTINUE......",Q$
9134 IF S7<>3 THEN 9140
9135 RESTORE
9136 REM...IF PRINTOUT NOT COMPLETE,SKIP PRINTING LEGEND
9138 IF F2>15 THEN RETURN
9140 REM****** PRINT LEGEND******
9150 GOSUB 8800
9160 RETURN
9500 REM******DECIMAL TO BINARY CONVERSION******
9503 D8=128
9508 D9=D(J,K)
9515 FOR B=1 TO 8
9518 Q=((K-3)*8)+(B+2)
9524 IF D9<D8 THEN 9536
9527 O(Q)=1
9530 D9=D9-D8
9533 GOTO 9539
9536 O(Q)=0
9539 D8=D8/2
9545 NEXT B
9551 RETURN
9600 REM******CREATE PRINT ARRAY FOR DATA WORD#0******
9605 REM...CONVERT SELECTED BINARY DATA TO LETTERS
9608 IF O(4)*O(3)=1 THEN LET P$(1,2)="11"
9611 IF O(4)+O(3)=0 THEN LET P$(1,2)="00"
9614 IF O(4)=0 AND O(3)=1 THEN LET P$(1,2)="01"
9617 IF O(4)=1 AND O(3)=0 THEN LET P$(1,2)="10"
9620 REM...PUT REMAINING OUTPUT ARRAY VALUES IN PRINT ARRAY
9623 P(6)=O(5)
9626 P(5)=O(6)
9629 P(4)=O(7)
9632 P(3)=O(8)
9635 P(2)=O(9)
9638 P(1)=O(10)
9640 RETURN
9650 REM******CREATE PRINT ARRAY FOR DATA WORD#01******
9653 REM...FIRST,CONVERT SELECTED BINARY DATA TO LETTERS
9655 IF O(12)*O(11)=1 THEN LET P$(3,4)="11"
9658 IF O(12)+O(11)=0 THEN LET P$(3,4)="00"
9661 IF O(12)=0 AND O(11)=1 THEN LET P$(3,4)="01"
9664 IF O(12)=1 AND O(11)=0 THEN LET P$(3,4)="10"
9667 IF O(14)*O(13)=1 THEN LET P$(5,6)="11"
9670 IF O(14)+O(13)=0 THEN LET P$(5,6)="00"
9673 IF O(14)=0 AND O(13)=1 THEN LET P$(5,6)="01"
9676 IF O(14)=1 AND O(13)=0 THEN LET P$(5,6)="10"
9678 REM...THEN,PUT REMAINING OUTPUT IN PRINT ARRAY
9680 P(11)=O(15)
9682 P(10)=O(16)
9684 P(9)=O(17)
9686 P(8)=O(18)
9695 RETURN
9700 REM******CREATE PRINT ARRAY FOR DATA WORD #2******
9703 REM...FIRST,CONVERT SELECTED BINARY DATA TO LETTERS
9705 IF O(23)*O(24)=1 THEN LET P$(7,8)="11"
9708 IF O(23)+O(24)=0 THEN LET P$(7,8)="00"
9711 IF O(23)=1 AND O(24)=0 THEN LET P$(7,8)="01"
9714 IF O(23)=0 AND O(24)=1 THEN LET P$(7,8)="10"
9717 IF O(25)*O(26)=1 THEN LET P$(9,10)="11"
9720 IF O(25)+O(26)=0 THEN LET P$(9,10)="00"
9723 IF O(25)=1 AND O(26)=0 THEN LET P$(9,10)="01"
9726 IF O(25)=0 AND O(26)=1 THEN LET P$(9,10)="10"
9729 REM...THEN,PUT REMAINING OUTPUT IN PRINT ARRAY
9730 P(24)=O(19)
9733 P(23)=O(20)
```

```
9736 P(22)=O(21)
9738 P(21)=O(22)
9740 RETURN
9750 REM****** CREATE PRINT ARRAY FOR DATA WORD#03******
9755 REM...SO,PUT ALL BINARY OUTPUT INPRINT ARRAY
9758 P(32)=O(27)
9760 P(31)=O(28)
9762 P(30)=O(29)
9764 P(29)=O(30)
9767 P(28)=O(31)
9770 P(27)=O(32)
9773 P(26)=O(33)
9776 P(25)=O(34)
9780 RETURN
9800 REM****** CREATE PRINT ARRAY FOR DATA WORD#04******
9805 REM...ONLY TWO BINARY VALUES INVOLVED
9810 REM...PUT THEM IN PRINT ARRAY
9815 P(34)=O(41)
9820 P(33)=O(42)
9825 RETURN
9850 REM******CREATE PRINT ARRAY FOR DATA WORD#05******
9855 P(20)=D(J,8)/5
9860 RETURN
9900 REM******PRINT TEST POINT DATA******
9935 ! #D1,N(J),TAB(4),T(J),
9941 ! #D1,TAB(12),P(1),P(2),P(3),P(4),P(5),P(6),TAB(25),P$(1,2),
9942 ! #D1,TAB(27),P(8),P(9),
9943 ! #D1,TAB(31),P(10),P(11),TAB(36),P$(5,6),TAB(39),P$(3,4),
9944 ! #D1,TAB(42),P$(9,10),TAB(45),P$(7,8),
9945 ! #D1,         P(20),TAB(52),P(21),P(22),P(23),P(24),
9946 ! #D1,P(25),P(26),P(27),P(28),P(29),
9947 ! #D1,TAB(70),P(30),P(31),P(32),P(33),P(34)
9975 RETURN
20000 REM******ABBREVIATED PRINTOUT ROUTINE*******
20005 ! #D2,"CHOICES FOR NEW FRAME NUMBER ARE:"
20010 IF S7=1 THEN 20100
20020 IF S7=2 THEN 20200
20030 IF S7=3 THEN 20300
20100 REM...FOR LOW RESOLUTION SCAN
20105 ! #D2,
20110 FOR J=0 TO 10
20115 ! #D2,N(J),
20120 NEXT J
20125 ! #D2,
20130 FOR J=11 TO 20
20135 ! #D2,N(J),
20140 NEXT J
20145 ! #D2,
20150 FOR J=21 TO 30
20155 ! #D2,N(J),
20160 NEXT J
20165 ! #D2,
20170 FOR J=31 TO 40
20175 ! #D2,N(J),
20180 NEXT J
20185 ! #D2,
20188 FOR J=40 TO 47
20190 ! #D2,N(J),
20192 NEXT J
20194 ! #D2,
20195 INPUT"NEW FRAME NUMBER? ",X9
20196 J=X9/150
20197 IF X9>7050 THEN ! #D2,"ILLEGAL CHOICE, TRY AGAIN !"
20198 IF X9>7050 THEN 20000
20199 GOTO 20500
```

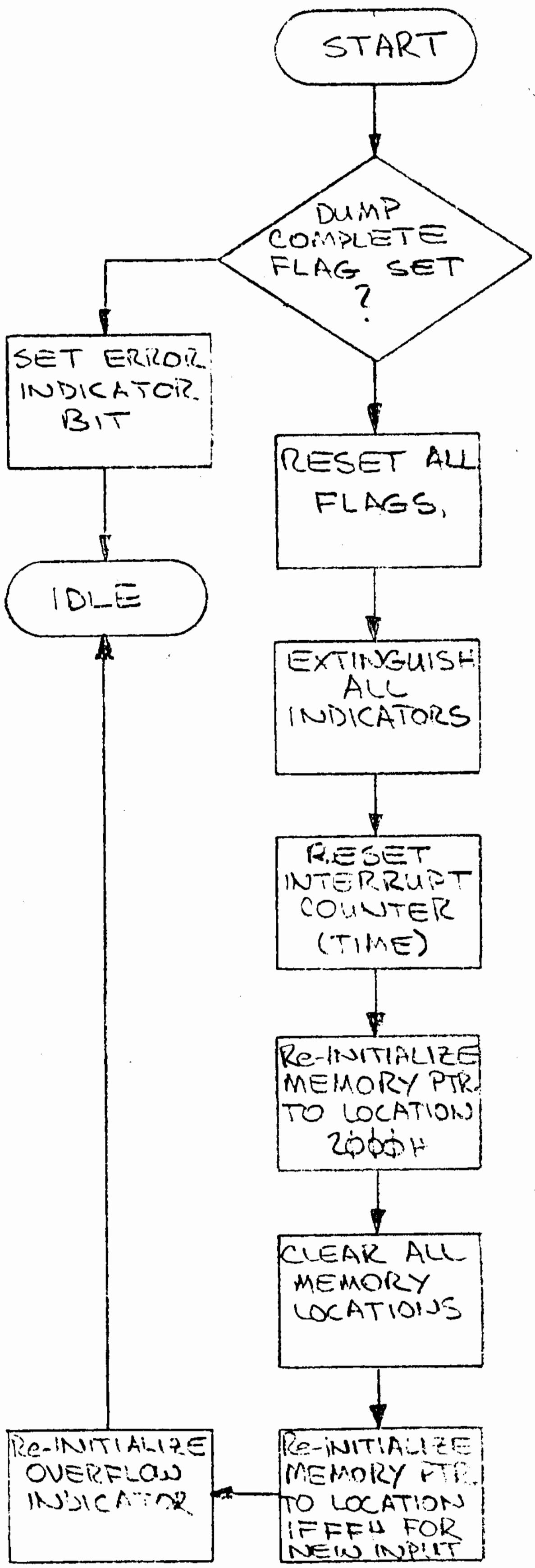
```
20200 REM...FOR MEDIUM RESOLUTION SCAN
20210 FOR J=0 TO 7
20215 ! #D2,N(J),
20220 NEXT J
20225 ! #D2,
20230 IF X5=7 THEN 20280
20235 FOR J=8 TO 10
20240 ! #D2,N(J),
20245 NEXT J
20250 ! #D2,
20280 INPUT"NEW FRAME NUMBER? ",X9
20285 IF X9>(S0+X4)/8 THEN ! #D2,"ILLEGAL CHOICE,TRY AGAIN!"
20288 IF X9>(S0+X4)/8 THEN 20000
20290 J=(X9-S1)/15
20295 GOTO 20500
20300 REM...FOR HIGH RESOLUTION SCAN
20305 IF S2>=7155 THEN 20350
20310 FOR J=0 TO 10
20315 ! #D2,N(J),
20320 NEXT J
20325 ! #D2,
20330 FOR J=11 TO X7/8
20335 ! #D2,N(J),
20340 NEXT J
20342 ! #D2,
20344 GOTO 20375
20350 FOR J=0 TO X7/8
20355 ! #D2,N(J),
20360 NEXT J
20365 ! #D2,
20375 INPUT"NEW FRAME NUMBER? ",X9
20380 IF X9>(S4+X7)/8 THEN ! #D2,"ILLEGAL CHOICE,TRY AGAIN!"
20385 IF X9>(S4+X7)/8 THEN 20000
20390 J=X9-S2
20500 RETURN
30000 REM******PAPER ADVANCE******
30010 FOR I=1 TO 19
30020 ! #1,
30030 NEXT
30040 STOP
```

VDAS DATA ACQUISITION PROGRAM

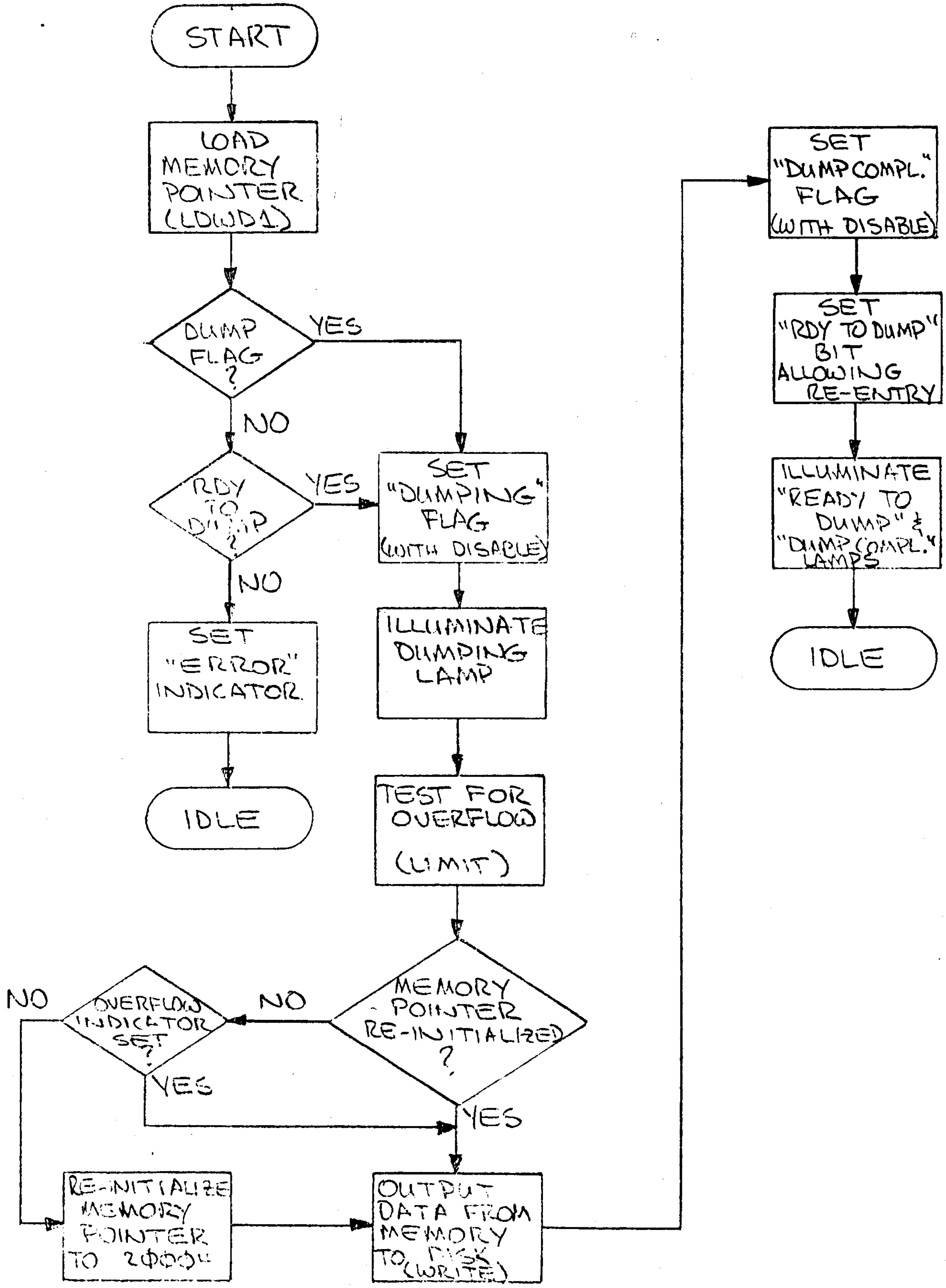
-EXEC-



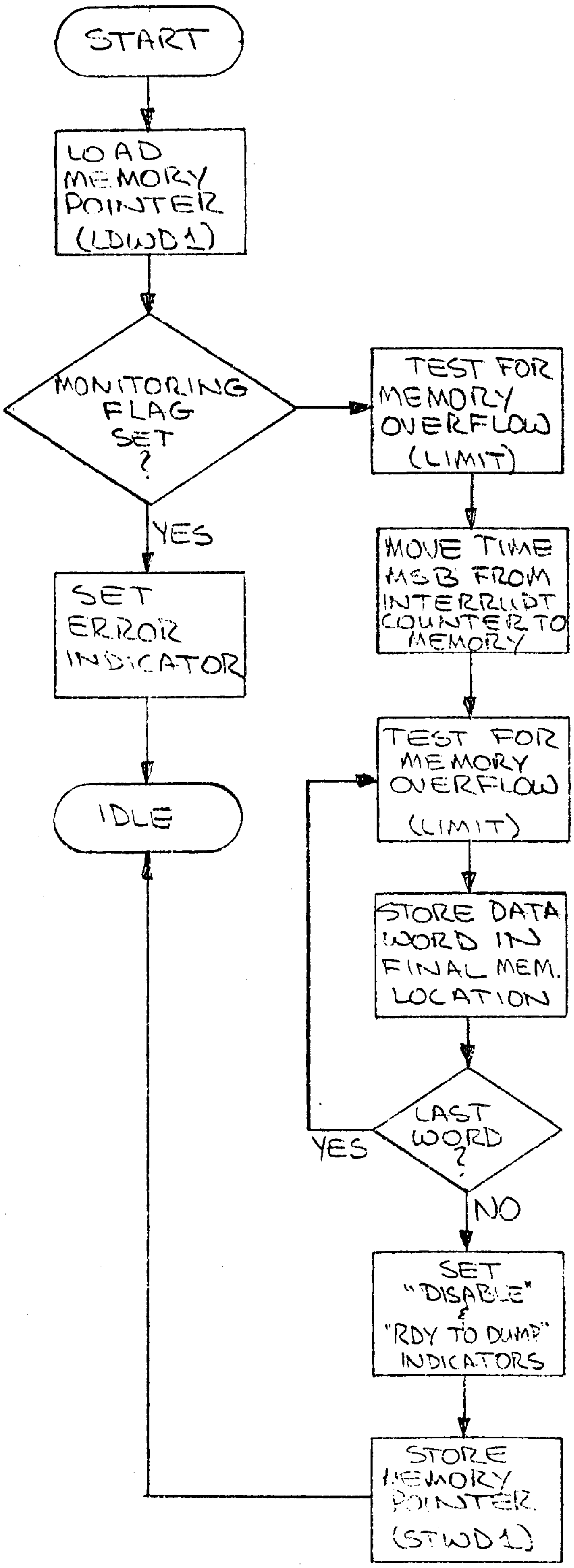
RE-INITIALIZE ROUTINE



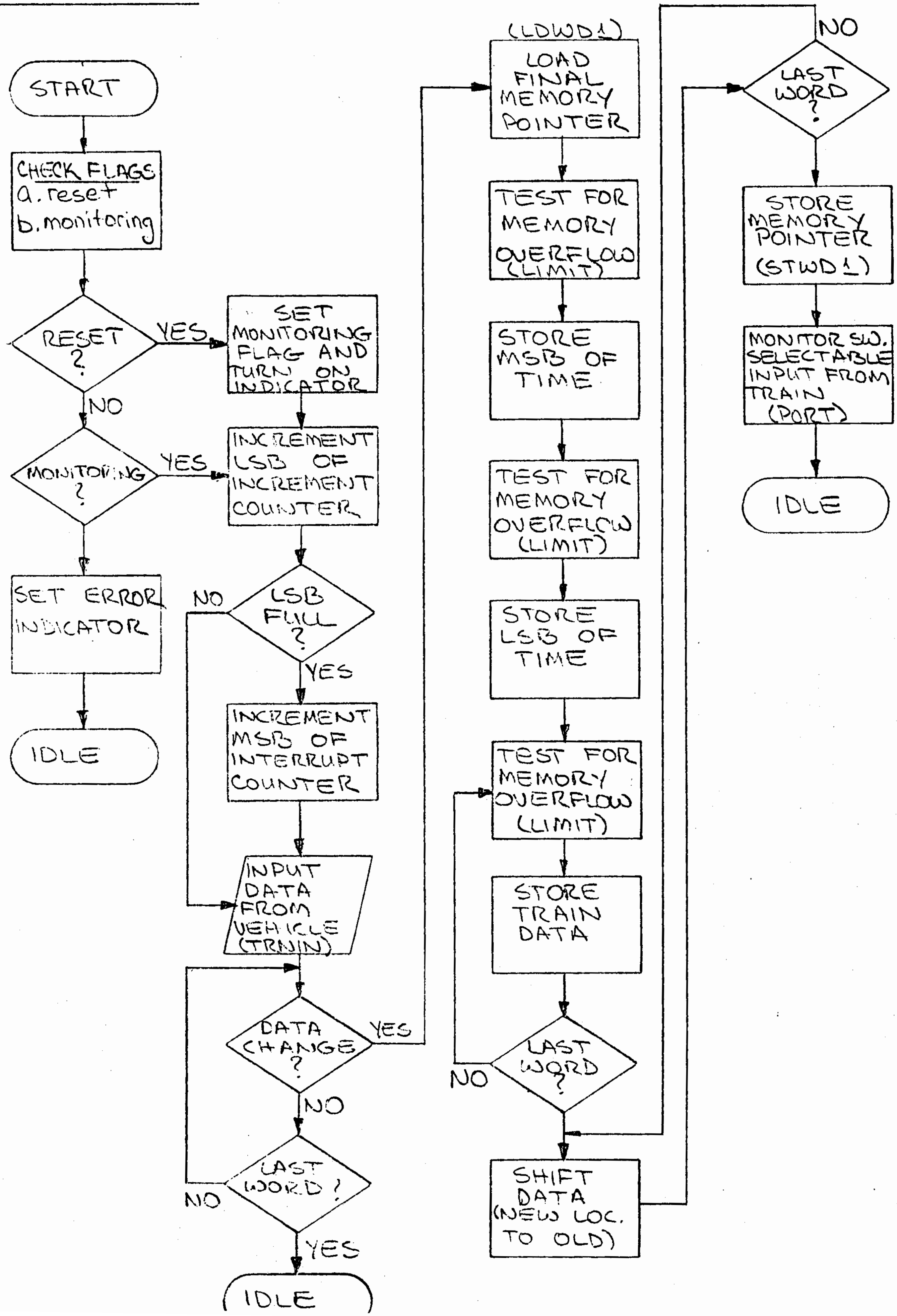
OUTPUT ROUTINE



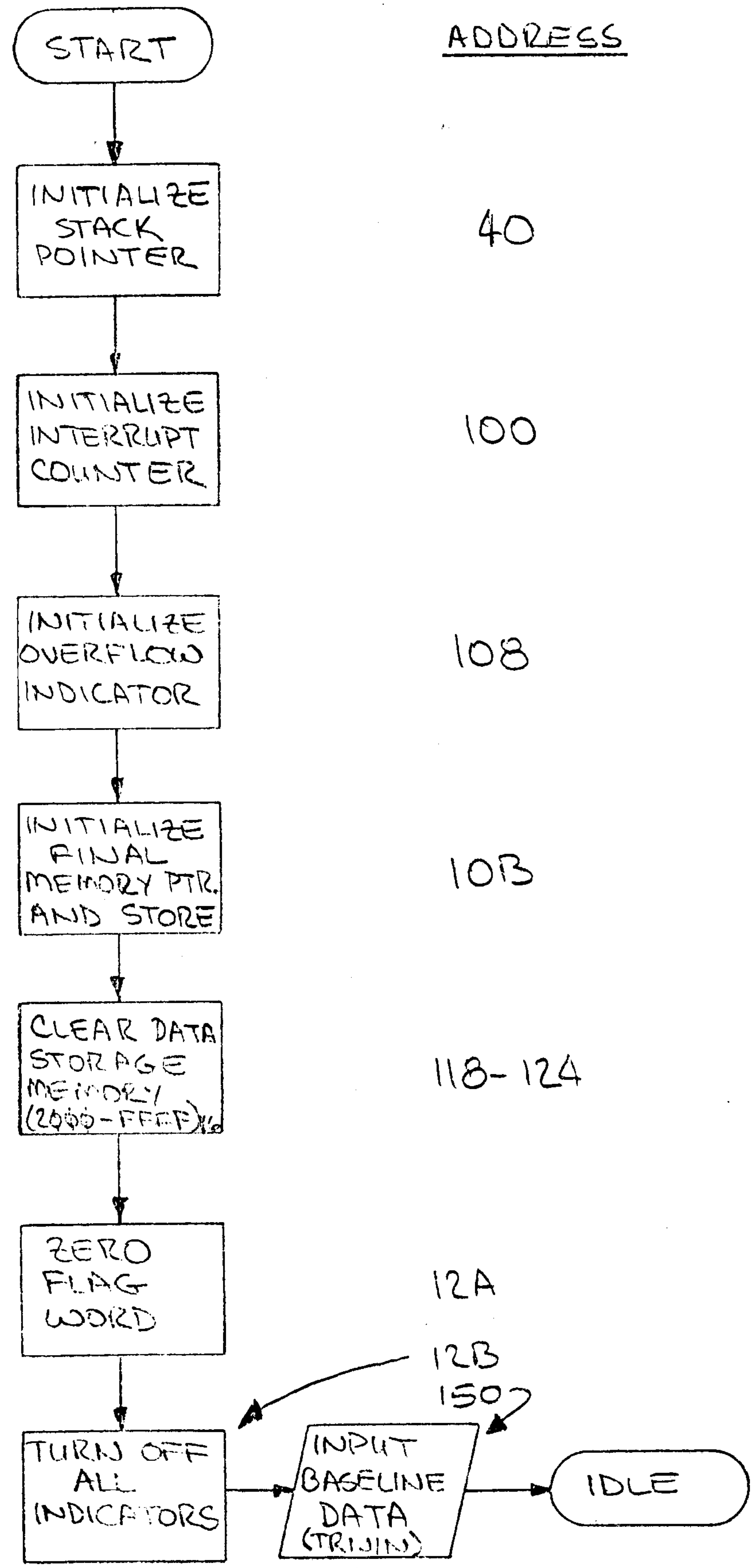
DISABLE ROUTINE



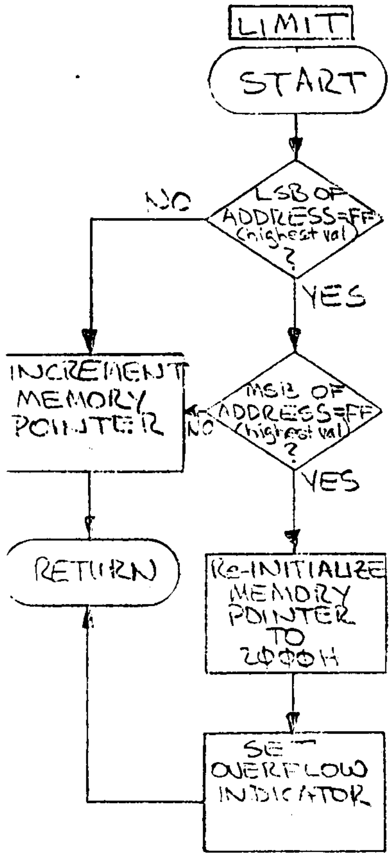
INPUT ROUTINE

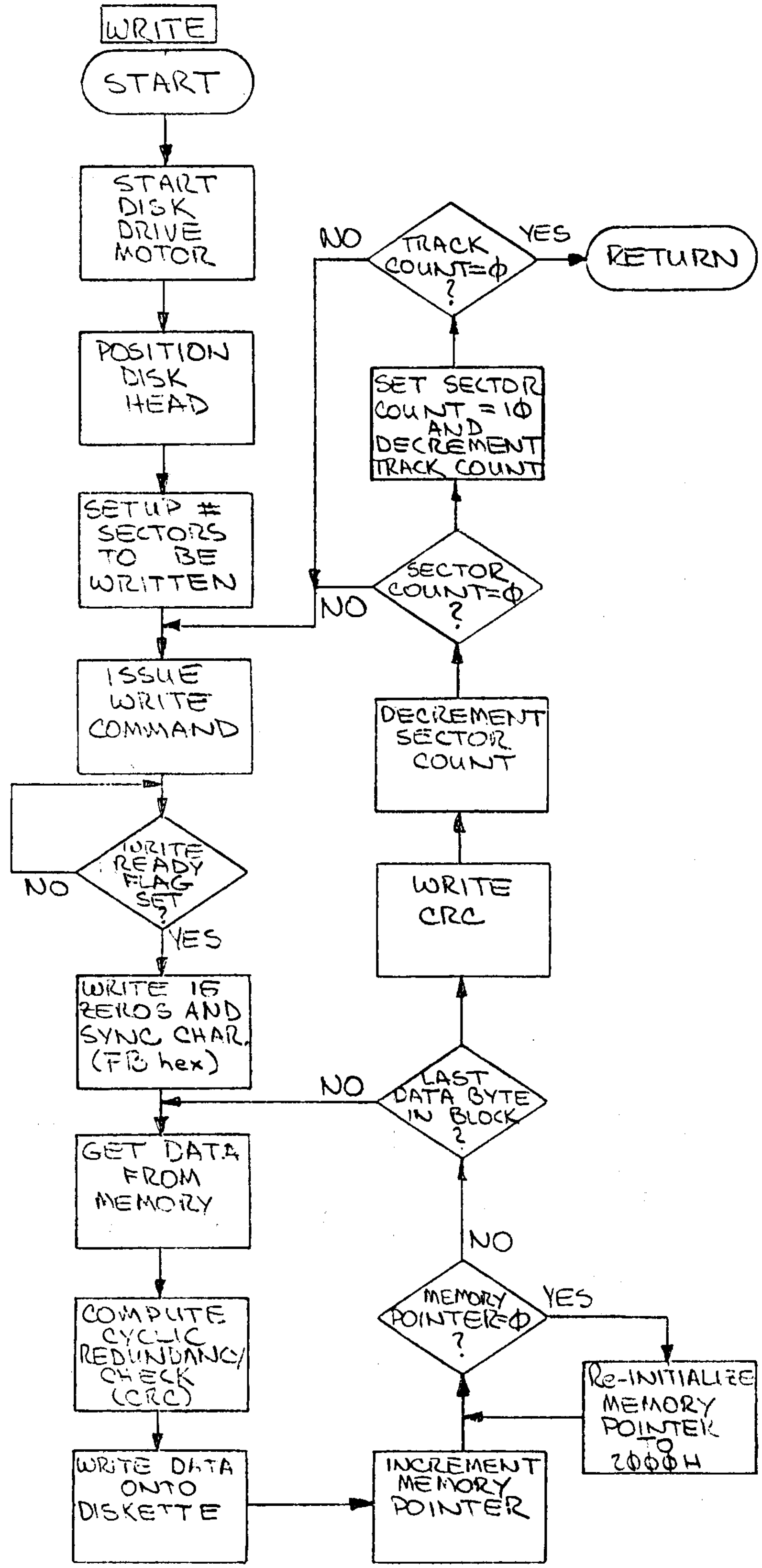


Power On Routine



SUBROUTINES





VDAS II - 6/28/78

PROM LABELED ASSM2 (000-3FF)

```
READY
ASSMX 0 8000

0000 00          0020 START NOP
0001 C3 40 00    0030  JMP PINIT
0004             0040  DS START+18H-$
0018 C3 80 01    0050 INT4 JMP VDASI ;SERVICE INPUT ROUTINE
001B             0060  DS START+20H-$
0020 C3 80 02    0070 INT3 JMP STRTO ;SERVICE REINITIALIZE ROUTINE
0023             0080  DS START+28H-$
0028 C3 00 03    0090 INT2 JMP DUMP ;SERVICE OUTPUT ROUTINE
002B             0100  DS START+30H-$
0030 C3 80 03    0110 INT1 JMP DSABL ;SERVICE DISABLE ROUTINE
0033             0120  DS START+40H-$
0040 31 FF       0130 PINIT DW 0FF31H ;INITIALIZE STACK POINTER
0042 13          0140  DB 13H
0043 C3 00 01    0150 INT0 JMP PWRON ;SERVICE POWER ON ROUTINE
0046 FB          0160 IDLE EI ;ENABLE INTERRUPTS
0047 76          0170  HLT ;WAIT
0048 C3 46 00    0180  JMP IDLE ;SPIN HERE UNTIL ANOTHER INTERRUPT OCCURS
004B             0190  DS START+100H-$
0100 3E 00       0200 PWRON MVI A,00H ;INITIALIZE INTERRUPT COUNTER
0102 32 00 10    0210  STA 1000H
0105 32 01 10    0220  STA 1001H
0108 32 05 10    0225  STA 1005H ;INITIALIZE OVERFLOW INDICATOR
010B 21 FF 1F    0230 MMPNT LXI H,1FFFH ;INITIALIZE FINAL MEMORY POINTER
010E 7C          0240  MOV A,H ;AND STORE AWAY
010F 32 03 10    0250  STA 1003H
0112 7D          0260  MOV A,L
0113 32 04 10    0270  STA 1004H
0116 3E 00       0275  MVI A,00H ;ZERO ACCUMULATOR
0118 77          0280 ZMEM MOV M,A ;CLR FINAL MEMORY LOCATION
0119 23          0290  INX H
011A 7C          0300  MOV A,H
011B EE 00       0310  XRI 00H ;TEST FOR LIMIT ADDRESS
011D CA 25 01    0320  JZ FLGWD
0120 3E 00       0330  MVI A,00H ;IF NOT LIMITING,CONTINUE CLEARING
0122 C3 18 01    0340  JMP ZMEM.
0125 11 02 10    0350 FLGWD LXI D,1002H ;SET FLAG POINTER
0128 3E 00       0360 FLG1 MVI A,00H ;ZERO ACCUMULATOR
012A 12          0370  STAX D
012B D3 06       0380  OUT 06H ;AND TURN OFF ALL DISPLAYS
012D 11 10 10    0390 BSLIN LXI D,1010H ;SET UP BASELINE STORAGE POINTER
0130 CD 50 01    0395  CALL TRNIN ;BRING DATA FROM TRAIN
0133 C3 46 00    0396  JMP IDLE ;RETURN TO EXEC
0136             0398  DS START+150H-$
0150 DB 00       0400 TRNIN IN 00H ;INPUT TRAIN DATA
0152 12          0410  STAX D ;AND STORE AWAY
0153 13          0420  INX D ;AND MOVE TO NEXT BASELINE DATA LOCATION
0154 13          0430  INX D
0155 DB 01       0440  IN 01H ;ETCETERA
0157 12          0450  STAX D
0158 13          0460  INX D
0159 13          0470  INX D
015A DB 02       0480  IN 02H
015C 12          0490  STAX D
```

```
015D 13          0500  INX D
015E 13          0510  INX D
015F DB 03       0520  IN 03H
0161 12          0530  STAX D
0162 13          0540  INX D
0163 13          0550  INX D
0164 DB 04       0560  IN 04H
0166 12          0570  STAX D
0167 13          0580  INX D
0168 13          0590  INX D
0169 DB 05       0600  IN 05H
016B 12          0610  STAX D
016C C9          0620  RET ;RETURN TO MAIN PROGRAM
016D             0630  DS START+180H-$
0180 3A 02 10    0650 VDASI LDA 1002H ;LOAD FLAGS
0183 EE 00       0660  XRI 00H ;RESET?
0185 CA 90 01    0670  JZ FLG2 ;IF SO CONTINUE
0188 EE 01       0672  XRI 01H ;ALREADY MONITORING?
018A CA 97 01    0674  JZ INCTM ;IF SO,INCREMENT INTR.CNTR.
018D C3 E0 03    0690  JMP ERRIN ;IF NOT SET,OPERATION INVALID
0190 3E 01       0700 FLG2 MVI A,01H ;SET "MONITOR FLAG"(& REST ERROR IND.)
0192 32 02 10    0710  STA 1002H
0195 D3 06       0720  OUT 06H ;AND TURN ON INDICATOR
0197 11 01 10    0730 INCTM LXI D,1001H ;GO TO LSB OF INTR. CNTR.
019A 1A          0740  LDAX D ;LOAD CONTENTS INTO ACCUMULATOR
019B 3C          0750  INR A ;AND INCREMENT
019C 12          0760  STAX D ;AND STORE AWAY
019D EE 00       0770  XRI 00H ;LSB FULL?
019F C2 A6 01    0780  JNZ DATN1 ;IF NOT,INPUT TRAIN DATA
01A2 1B          0790  DCX D ;OTHERWISE,GO TO MSB OF INTERRUPT CNTR.
01A3 1A          0800  LDAX D ;BRING CONTENTS INTO ACCUMULATOR
01A4 3C          0810  INR A ;AND INCREMENT
01A5 12          0820  STAX D ;AND STORE AWAY
01A6 11 11 10    0830 DATN1 LXI D,1011H ;SETUP BUFFER STORAGE
01A9 CD 50 01    0840  CALL TRNIN ;AND BRING IN NEW TRAIN DATA
01AC 11 10 10    0850 INCMP LXI D,1010H ;SET UP MEM. PNTR TO EXAMINE OLD DATA
01AF 0E 06       0860  MVI C,06H ;SET WORD SIZE COUNTER
01B1 1A          0870 WDCMP LDAX D ;MOVE OLD DATA TO ACCUMULATOR
01B2 13          0880  INX D ;BUMP POINTER TO NEW DATA
01B3 EB          0885  XCHG ;SET UP FOR COMPARISON
01B4 AE          0890  XRA M ;COMPARE NEW DATA WITH OLD
01B5 EB          0895  XCHG ;RESTORE REGISTERS
01B6 C2 C1 01    0900  JNZ STRWD ;IF DIFFERENT,GOTO DATA STORAGE ROUTINE
01B9 0D          0910  DCR C ;DECREMENT WORD SIZE COUNTER
01BA CA F5 01    0920  JZ EXIT2 ;IF DONE PREPARE TO EXIT
01BD 13          0930  INX D ;IF NOT, MOVE TO NEXT OLD DATA
01BE C3 B1 01    0940  JMP WDCMP ;AND MAKE MORE COMPARISONS
01C1 CD 60 02    0950 STRWD CALL LDWD1 ;LOAD FINAL MEMORY POINTER
01C4 CD 40 02    0960  CALL LIMIT ;TEST FOR MEMORY OVERFLOW
01C7 3A 00 10    0970  LDA 1000H ;STORE MSB OF TIME
01CA 77          0980  MOV M,A
01CB CD 40 02    0990  CALL LIMIT
01CE 3A 01 10    1000  LDA 1001H ;STORE LSB OF TIME
01D1 77          1010  MOV M,A
01D2 11 11 10    1020  LXI D,1011H ;GO TO FIRST DATA WORD
01D5 0E 06       1030  MVI C,06H ;AND SET WORD SIZE COUNTER
01D7 CD 40 02    1040 LIM3 CALL LIMIT ;TEST FOR MEMORY OVERFLOW
01DA 1A          1050  LDAX D ;MOVE WORD INTO ACCUMULATOR
01DB 77          1060  MOV M,A ;AND THEN STORE IN FINAL MEMORY
01DC 13          1070  INX D ;GO TO NEXT BUFFER WORD
01DD 13          1080  INX D
01DE 0D          1090  DCR C ;DECREMENT WORD SIZE COUNTER
01DF C2 D7 01    1100  JNZ LIM3 ;IF NOT DONE,GO BACK AND STORE AGAIN
01E2 11 1B 10    1200 ODTNW LXI D,101BH ;GO TO END OF INPUT BUFFER
01E5 0E 06       1205  MVI C,06H ;SET WORD SIZE COUNTER
01E7 1A          1210 XFER LDAX D ;LOAD WORD INTO ACCUMULATOR
```

```
01E8 1B           DCX D ;GO TO OLD DATA LOCATION
01E9 12           STAX D ;STORE NEW DATA INTO OLD DATA LOCATION
01EA 0D           DCR C ;DECREMENT WORD SIZE COUNTER
01EB CA F2 01     JZ RSTR1 ;DONE? IF SO,PREPARE TO EXIT
01EE 1B           DCX D ;IF NOT, GO TO NEXT NEW DATA LOCATION
01EF C3 E7 01     JMP XFER ;AND GO BACK AND PRANSFER MORE DATA
01F2 CD 70 02    RSTR1 CALL STWD1 ;STORE FINAL MEMORY POINTER
01F5             PORT EQU 1080H ;DEFINE INPUT INSTRUCTION AREA
01F5 21 80 10    EXIT2 LXI H,1080H ;SET MEMORY POINTER TO RAM
01F8 36 DB        MVI M,0DBH ;PUT INPUT INSTRUCTION IN RAM
01FA 23           INX H ;INCREMENT MEMORY POINTER
01FB DB 07        IN 07H ;INPUT SWITCH SETTINGS
01FD E6 07        ANI 07H ;MASK OUT UNUSED SWITCHES
01FF FE 07        CPI 07H ;SWITCH SETTING SEVEN?
0201 CA 0E 02     JZ TIME ;IF SO,MONITOR CLOCK
0204 77           MOV M,A ;IF NOT,STORE SWITCH SETTINGS
0205 23           INX H ;INCREMENT MEMORY POINTER
0206 36 C9        MVI M,0C9H ;LOAD MEMORY WITH 'RETURN' INSTRUCTION
0208 CD 80 10     CALL PORT ;EXECUTE INSTRUCTIONS IN RAM
020B D3 05        OUT 05H ;DISPLAY SWITCH SETTINGS ON LEDS
020D C9           RET ;AND GO BACK TO IDLE
020E 3A 01 10    TIME LDA 1001H ;LOAD LSB OF TIME
0211 D3 05        OUT 05H ;AND OUTPUT TO LEDS
0213 C9           RET ;AND GO BACK TO IDLE
0214              DS START+240H-$
0240 7D          LIMIT MOV A,L ;CHECK FOR MEMORY OVERFLOW
0241 EE FF        XRI 0FFH
0243 CA 48 02     JZ TSMSB
0246 23          BUMP INX H
0247 C9           RET  ;GO BACK TO MAIN PROGRAM
0248 7C          TSMSB MOV A,H
0249 EE FF        XRI 0FFH ;LIMIT REACHED?
024B CA 51 02     JZ REINT
024E C3 46 02     JMP BUMP
0251 21 00 20    REINT LXI H,2000H ;IF SO,REINITIALIZE MEMORY POINTER
0254 3E 01       SETOV MVI A,01H ;AND SET OVERFLOW INDICATOR
0256 32 05 10     STA 1005H
0259 C9          EXIT1 RET ;AND GO BACK TO MAIN PROGRAM
025A              DS START+260H-$
0260 3A 03 10    LDWD1 LDA 1003H ;LOAD FINAL MEMORY INTO H&L
0263 67           MOV H,A
0264 3A 04 10     LDA 1004H
0267 6F           MOV L,A
0268 C9           RET ;GO BACK TO MAIN PROGRAM
0269              DS START+270H-$
0270 7C          STWD1 MOV A,H ;MOVE FINAL MEMORY POINTER TO BUFFER
0271 32 03 10     STA 1003H
0274 7D           MOV A,L
0275 32 04 10     STA 1004H
0278 C9           RET ;GO BACK TO MAIN PROGRAM
0279              DS START+280H-$
0280 3A 02 10    STRTO LDA 1002H ;LOAD FLAGS
0283 EE 60        XRI 60H ;"DUMP COMPLETE SET(ALONG WITH DISABLE)?
0285 C2 E0 03     JNZ ERRIN ;IF NOT,ERROR CONDITION PRESENT
0288 3E 00        MVI A,00H ;IF SO,RESET FLAGS
028A 32 02 10     STA 1002H ;AND STORE AWAY
028D D3 06        OUT 06H ;TURN OFF INDICATORS
028F 32 00 10     STA 1000H ;RESET MSB OF TIME
0292 32 01 10     STA 1001H ;RESET LSB OF TIME
0295 21 00 20    SETUP LXI H,2000H ;INITIALIZE MEMORY POINTER
0298 AF          BGCL2 XRA A ;ZERO ACCUMULATOR
0299 77          CLMM3 MOV M,A ;AND XFER TO MEMORY
029A 7D           MOV A,L ;TEST LSB OF ADDRESS
029B EE FF        XRI 0FFH ;AGAINST MAX. ADDRESS
029D CA A4 02     JZ LMTST ;IF MAX.,TEST MSB
02A0 23           INX H ;IF NOT,GO TO NEXT ADDRESS
```

```
02A1 C3 98 02      1476  JMP BGCL2 ;AND CONTINUE TO CLEAR MEMORY
02A4 7C            1478 LMTST MOV A,H ;TEST MSB OF ADDRESS
02A5 EE FF         1479  XRI OFFH ;AGAINST MAX.ADDRESS
02A7 CA AE 02      1480  JZ STUP2 ;IF MAX.,STOP CLEARING
02AA 23            1482  INX H ;IF NOT,GO TO NEXT ADDRESS
02AB C3 98 02      1484  JMP BGCL2 ;AND CONTINUE TO CLEAR MEMORY
02AE 21 FF 1F      1486 STUP2 LXI H,1FFFH ;REINITIALIZE MEM. PNTR. FOR NEW INPUT
02B1 CD 70 02      1488  CALL STWD1 ;AND STORE AWAY
02B4 3E 00         1490  MVI A,00H ;INITIALIZE OVERFLOW INDICATOR
02B6 32 05 10      1492  STA 1005H ;AND STORE AWAY
02B9 C9            1494 EXIT3 RET ;GO BACK TO IDLE
02BA               1500  DS START+300H-$
0300 CD 60 02      1510 DUMP CALL LDWD1
0303 3A 02 10      1520  LDA 1002H ;LOAD FLAGS
0306 EE 60         1530  XRI 60H ;TEST REENTRY
0308 CA 1B 03      1540  JZ DBIT ;IF SO,SET "DUMPING" FLAG BIT(WITH DISABLE)
030B 3A 02 10      1550 ENTR1 LDA 1002H ;IF NOT,LOAD FLAGS AGAIN
030E EE 42         1560  XRI 42H ;AND CHECK "READY TO DUMP" FLAG
0310 C2 E0 03      1570  JNZ ERRIN ;IF NOT SET,INDICATE ERROR
0313 3E 44         1580  MVI A,44H ;IF SO,SET "DUMPING"FLAG
0315 32 02 10      1590  STA 1002H ;AND STORE AWAY
0318 C3 1D 03      1600  JMP DLED ;PROCEED TO TURNING ON INDICATOR
031B 3E 44         1610 DBIT MVI A,44H ;IF SO,SET "DUMPING" FLAG
031D D3 06         1620 DLED OUT 06H ;ILLUMINATE LED
031F CD 40 02      1630 TEST  CALL LIMIT ;TEST FOR OVERFLOW
0322 7C            1640  MOV A,H ;CHECK FOR REINITIALIZATION
0323 EE 20         1650  XRI 20H
0325 C2 31 03      1655  JNZ OVFLW
0328 7D            1660  MOV A,L
0329 EE 00         1665  XRI 00H
032B C2 31 03      1670  JNZ OVFLW
032E C3 42 03      1675  JMP DSKO ;IF SO,GO AND OUTPUT TO DISK
0331 3A 05 10      1680 OVFLW LDA 1005H ;IF NOT,TEST OVERFLOW INDICATOR
0334 EE 00         1685  XRI 00H ;SET?
0336 CA 3C 03      1690  JZ RESET ;IF NOT,REINITIALIZE FIRST
0339 C3 3F 03      1695  JMP WEXIT ;IF SO,GO DIRECTLY TO DISK OUT
033C 21 00 20      1700 RESET LXI H,2000H
033F C3 42 03      1705 WEXIT JMP DSKO
0342 CD 00 0E      1710 DSKO CALL WRITE
0345 3E 60         1780 QTCLR MVI A,60H ;SET "DUMP COMPLETE"FLAG(WITH DISABLE SET)
0347 32 02 10      1785  STA 1002H ;AND STORE AWAY
034A F6 02         1790  ORI 02H ;SET "RDY TO DUMP"BIT TO ALLOW REENTRY
034C D3 06         1795  OUT 06H ;TURN ON "RDY TO DUMP" & "DUMP COMPLETE" LEDS
034E C9            1798 EXIT4 RET ;AND GO BACK TO IDLE
034F               1800  DS START+380H-$
0380 CD 60 02      1810 DSABL CALL LDWD1 ;LOAD FINAL MEMORY POINTER
0383 3A 02 10      1820  LDA 1002H ;CHECK MONITORING FLAG
0386 EE 01         1830  XRI 01H ;SET?
0388 C2 E0 03      1840  JNZ ERRIN ;IF NOT,GIVE ERROR INDICATION
038B CD 40 02      1850 TIMN2 CALL LIMIT ;TEST FOR MEMORY OVERFLOW
038E 3A 00 10      1860  LDA 1000H ;LOAD MSB OF TIME
0391 77            1870  MOV M,A ;AND PUT INTO STORAGE
0392 CD 40 02      1880  CALL LIMIT ;TEST AGAIN
0395 3A 01 10      1890  LDA 1001H ;LOAD LSB OF TIME
0398 77            1900  MOV M,A ;AND PUT INTO STORAGE
0399 11 11 10      1910  LXI D,1011H ;GO TO NEW DATA
039C 0E 06         1920  MVI C,06H ;SET WORD SIZE COUNTER
039E CD 40 02      1930 DATN2 CALL LIMIT ;TEST
03A1 1A            1940  LDAX D ;BRING NEW DATA IN
03A2 77            1950  MOV M,A ;AND STORE AWAY IN FINAL STORAGE
03A3 13            1990 NEWRD INX D ;MOVE TO NEXT NEW WORD
03A4 13            2000  INX D
03A5 0D            2010  DCR C ;DECREMENT WORD SIZE COUNTER
03A6 C2 9E 03      2020  JNZ DATN2 ;IF NOT DONE,GET MORE DATA
03A9 3E 42         2030 FLAG2 MVI A,42H ;SET"RDY TO DUMP" AND DISABL CLK" FLGS
03AB 32 02 10      2040  STA 1002H ;STORE AWAY
```

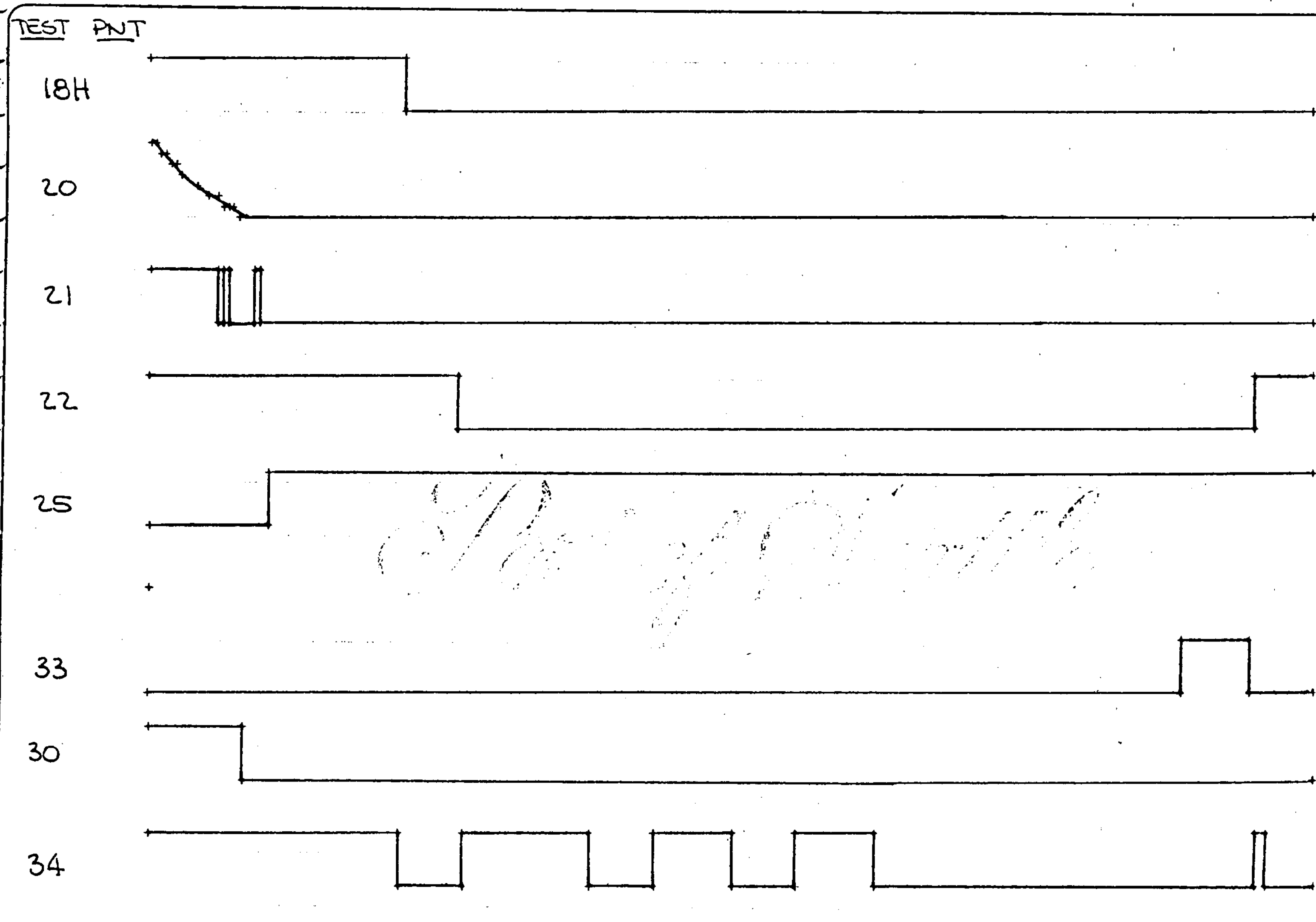
```
03AE D3 06          2060 OUT 06H ;TURN ON "RDY TO DUMP" INDICATOR
03B0 CD 70 02       2070 RSTR2 CALL STWD1 ;STORE FINAL MEMORY POINTER
03B3 C9             2080  RET ;GO BACK TO IDLE
03B4                2081  DS START+3E0H-$
03E0 3A 02 10       2082 ERRIN LDA 1002H ;LOAD FLAG WORD
03E3 F6 10          2084  ORI 10H ;COMBINE WITH "ERROR" BIT
03E5 D3 06          2086  OUT 06H ;TURN ON ERROR INDICATOR
03E7 C9             2088  RET ;AND GO BACK TO IDLE
03E8                2105  DS START+0E00H-$
0E00                2110 BASE EQU 1800H
0E00                2115 *******THE FOLLOWING ROUTINE PLACES 56K OF TEST DATA
0E00                2116 **************ON DISK FOR LATER ANALYSIS************
0E00 3A 90 1B       2120 WRITE   LDA BASE+390H ;START DRIVE MOTOR
0E03 16 32          2130         MVI D,50 ;WAIT 50 SECTOR TIMES
0E05 CD D0 19       2140         CALL BASE+1D0H
0E08 3A 01 1B       2150         LDA BASE+301H ;LOAD HEAD
0E0B 16 0D          2160         MVI D,13 ;WAIT 13 SECTOR TIMES
0E0D CD D0 19       2170         CALL BASE+1D0H
0E10 3A 1C 1B       2180         LDA BASE+31CH ;SET UP HEAD TO MOVE OUT
0E13 3A 10 1B       2190 TRACK0  LDA BASE+310H ;GET STATUS
0E16 E6 01          2200         ANI 1 ;MASK FOR TRACK ZERO FLAG
0E18 C2 2B 0E       2210         JNZ FIRST ;EXIT IF TRACK ZERO FOUND
0E1B 3A 09 1B       2220         LDA BASE+309H ;SET STEP FLIP FLOP
0E1E E3             2230         XTHL ;PAUSE
0E1F E3             2240         XTHL
0E20 3A 08 1B       2250         LDA BASE+308H ;RESET STEP FLIP FLOP
0E23 16 02          2260         MVI D,2 ;WAIT 2 SECTOR TIMES
0E25 CD D0 19       2270         CALL BASE+1D0H
0E28 C3 13 0E       2280         JMP TRACK0 ;LOOP UNTIL TRACK ZERO FOUND
0E2B 3A 1D 1B       2290 FIRST   LDA BASE+31DH ;SET UP HEAD TO MOVE IN
0E2E 01 17 04       2300         LXI B,400H+23 ;B=4=SECTORS,C=23=TRACKS
0E31 CD CE 19       2310 FIND    CALL BASE+1CEH ;WAIT FOR NEXT SECTOR
0E34 3A 30 1B       2320         LDA BASE+330H ;GET "B" STATUS
0E37 E6 0F          2330         ANI 0FH ;MASK FOR SECTOR COUNT
0E39 FE 06          2340         CPI 6 ;LOOK FOR SECTOR 6
0E3B C2 31 0E       2350         JNZ FIND
0E3E 3A 04 1B       2360 WRIT    LDA BASE+304H ;WRITE COMMAND
0E41 3A 10 1B       2370 WRSTAT  LDA BASE+310H ;GET STATUS
0E44 E6 08          2380         ANI 8 ;MASK FOR "WRITE RDY"FLAG
0E46 CA 41 0E       2390         JZ WRSTAT ;WAIT FOR WRITE READY
0E49 C5             2400         PUSH B ;SAVE COUNTERS
0E4A 11 00 1A       2410         LXI D,BASE+200H ;SET UP TO WRITE ZEROES
0E4D 01 0F 00       2420         LXI B,15 ;ZEROES=15
0E50 1A             2430 ZERLP   LDAX D ;WRITE A ZERO
0E51 0D             2440         DCR C ;COUNT
0E52 C2 50 0E       2450         JNZ ZERLP ;IF NOT DONE,WRITE MORE ZEROES
0E55 1E FB          2460         MVI E,0FBH ;IF DONE,SET UP "SYNC"
0E57 1A             2470         LDAX D ;WRITE IT
0E58 7E             2480 WRLP    MOV A,M ;GET DATA FROM VDAS MEMORY
0E59 5F             2490         MOV E,A ;SET UP DATA FOR WRITING
0E5A A8             2500         XRA B ;COMPUTE CRC
0E5B 07             2510         RLC
0E5C 47             2520         MOV B,A
0E5D 1A             2530         LDAX D ;WRITE DATA
0E5E 23             2540         INX H ;BUMP MEMORY POINTER
0E5F 7C             2550         MOV A,H ;CHECK FOR MEMORY OVERFLOW
0E60 B5             2560         ORA L
0E61 C2 66 0E       2570         JNZ CONT ;IF NOT REACHED,DON'T WRAPAROUND
0E64 26 20          2580         MVI H,20H ;IF SO,REINITIALIZE TO 2000H
0E66 0D             2590 CONT    DCR C ;COUNT BYTES
0E67 C2 58 0E       2600         JNZ WRLP ;WRITE 256 TIMES
0E6A 58             2610         MOV E,B ;WRITE CRC
0E6B 1A             2620         LDAX D
0E6C C1             2630         POP B ;GET TRACK & SECTOR COUNTERS
0E6D CD CE 19       2640         CALL BASE+1CEH ;WAIT FOR NEXT SECTOR
0E70 05             2650         DCR B ;COUNT SECTORS
```

```
0E71 C2 3E 0E        2660        JNZ WRIT ;LOOP 10 SECTORS PER TRACK
0E74 3A 09 1B        2670        LDA BASE+309H ;SET HEAD FLIP FLOP
0E77 E3              2680        XTHL ;PAUSE
0E78 E3              2690        XTHL
0E79 3A 08 1B        2700        LDA BASE+308H ;RESET HEAD FLIP FLOP
0E7C 16 0A           2710        MVI D,10 ;WAIT 10 SECTORS
0E7E CD D0 19        2720        CALL BASE+1D0H
0E81 06 0A           2730        MVI B,10 ;RESET SECTOR COUNT TO 10
0E83 0D              2740        DCR C ;COUNT TRACK
0E84 C2 3E 0E        2750        JNZ WRIT ;LOOP 23 TRACKS
0E87 C9              2760        RET ;GO BACK TO "DUMP"ROUTINE

BASE    1800    2120 2140 2150 2170 2180 2190 2220 2250 2270 2290 2310
                2320 2360 2370 2410 2640 2670 2700 2720
BGCL2   0298    1476 1484
BSLIN   012D
BUMP    0246    1324
CLMM3   0299
CONT    0E66    2570
DATN1   01A6    0780
DATN2   039E    2020
DBIT    031B    1540
DLED    031D    1600
DSABL   0380    0110
DSKO    0342    1675 1705
DUMP    0300    0090
ENTR1   030B
ERRIN   03E0    0690 1430 1570 1840
EXIT1   0259
EXIT2   01F5    0920
EXIT3   02B9
EXIT4   034E
FIND    0E31    2350
FIRST   0E2B    2210
FLAG2   03A9
FLG1    0128
FLG2    0190    0670
FLGWD   0125    0320
IDLE    0046    0180 0396
INCMP   01AC
INCTM   0197    0674
INT0    0043
INT1    0030
INT2    0028
INT3    0020
INT4    0018
LDWD1   0260    0950 1510 1810
LIM3    01D7    1100
LIMIT   0240    0960 0990 1040 1630 1850 1880 1930
LMTST   02A4    1472
MMPNT   010B
NEWRD   03A3
ODTNW   01E2
OVFLW   0331    1655 1670
PINIT   0040    0030
PORT    1080    1283
PWRON   0100    0150
QTCLR   0345
REINT   0251    1321
RESET   033C    1690
RSTR1   01F2    1230
RSTR2   03B0
SETOV   0254
SETUP   0295
START   0000    0040 0060 0080 0100 0120 0190 0398 0630 1299 1339 1357
                1400 1500 1800 2081 2105
```

```
STRTO   0280     0070
STRWD   01C1     0900
STUP2   02AE     1480
STWD1   0270     1245 1488 2070
TEST    031F
TIME    020E     1271
TIMN2   038B
TRACK   0E13     2280
TRNIN   0150     0395 0840
TSMSB   0248     1306
VDASI   0180     0050
WDCMP   01B1     0940
WEXIT   033F     1695
WRIT    0E3E     2660 2750
WRITE   0E00     1710
WRLP    0E58     2600
WRSTA   0E41     2390
XFER    01E7     1240
ZERLP   0E50     2450
ZMEM    0118     0340

FILE /PCOMP/

PCOMP 9000  92C6
LIST
0010 * THIS PROGRAM IS USED FOR REPROGRAMMING
0020 * 1K PROMS AFTER REASSEMBLY
0030 * ALL UNCHANGED LOCATIONS ARE SET TO DDH
0040 * ALL CHANGED LOCATIONS ARE SET TO NEW VALUE
0050 * LOAD OLD OBJECT CODE AT 8000H
0060 * LOAD NEW OBJECT CODE AT A000H
0070 * EDITED CODE IS LOADED AT B000H
0100  LXI H,8000H
0105  LXI D,0A000H
0110 CHECK MOV A,M
0114  DW 0FF31H
0117  DB 55H
0130  XCHG
0133  SHLD 5500H
0140  MOV B,M
0150  XRA B
0160  JZ CLEAR
0162 DELTA PUSH D
0165  LXI D,1000H
0170  DAD D
0180  MOV M,B
0190  LHLD 5500H
0195  POP D
0200  XCHG
0210  INX D
0220  INX H
0222  MOV A,H
0225  XRI 90H
0228  JZ QUIT
0230  JMP CHECK
0240 CLEAR MVI B,0DDH
0250  JMP DELTA
0260 QUIT HLT

EXEC 2028
```

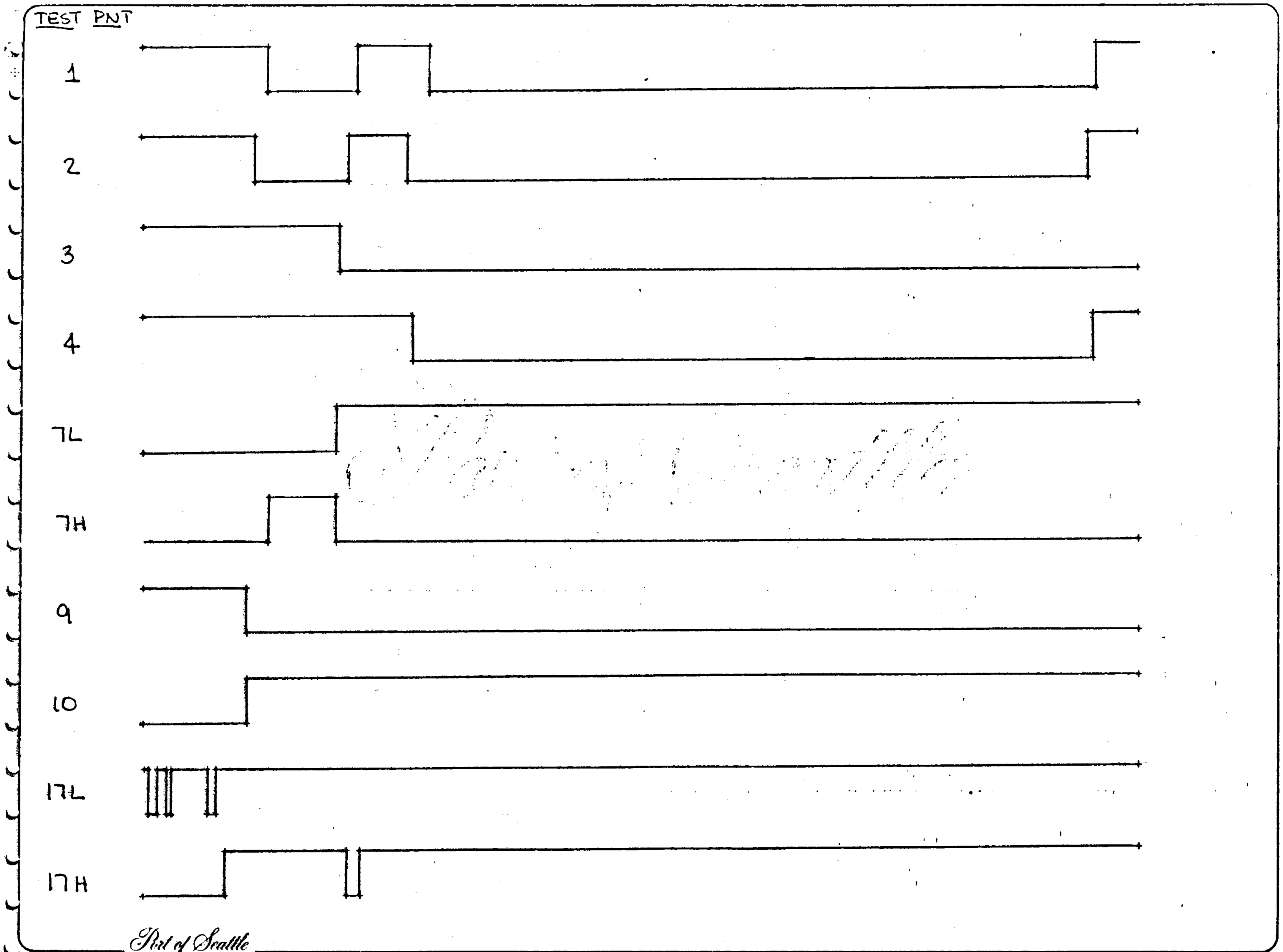
APPENDIX D

Data Analysis



Data Analysis : Vehicle failure #1
(based on data outputed 10/13/78)

Port of Seattle



TEST PNT
1
2
3
4
7L
7H
9
10
17L
17H
Port of Seattle

10/13/78

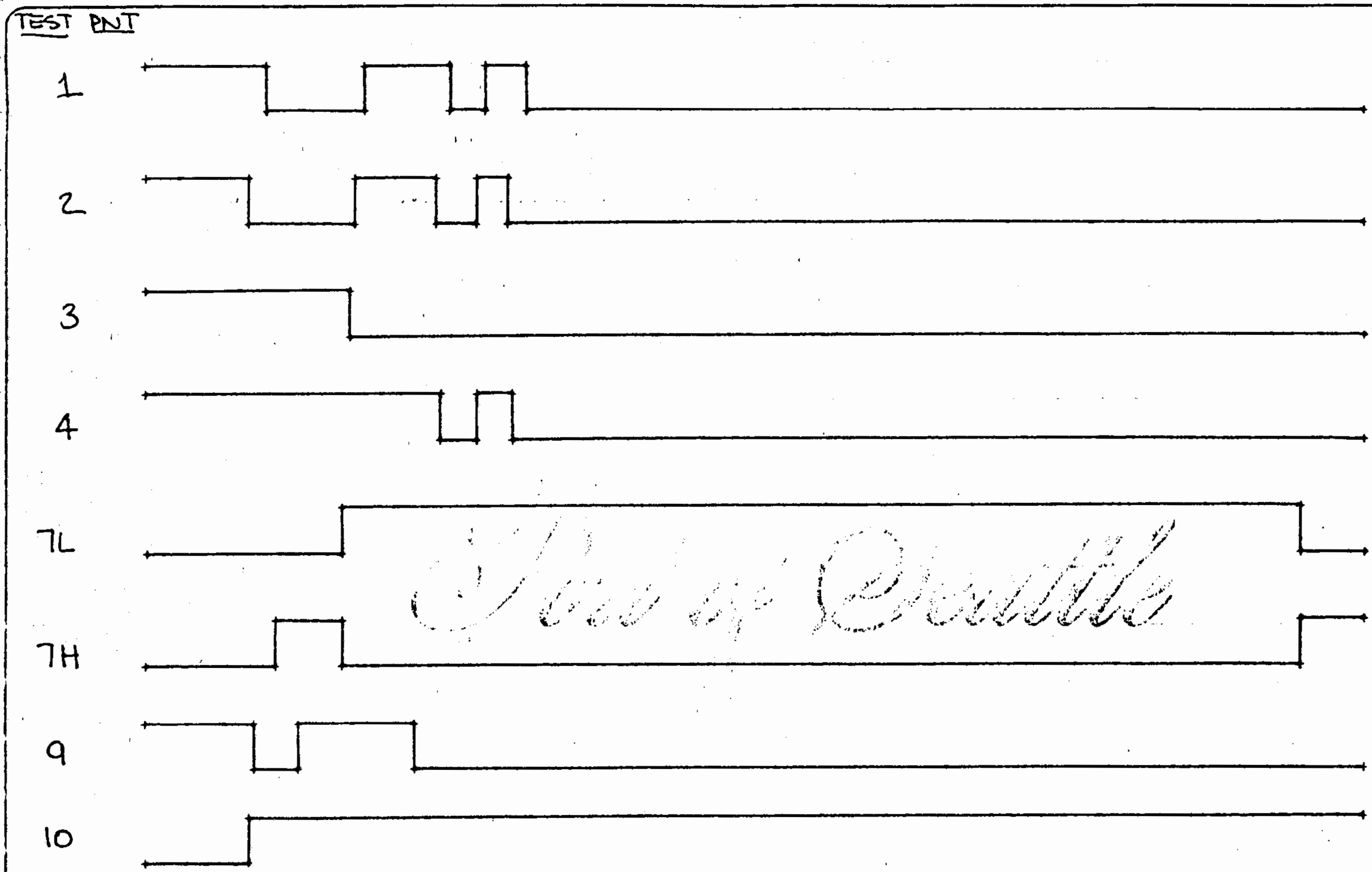
STOP IN LINE 31000

TEST DATA

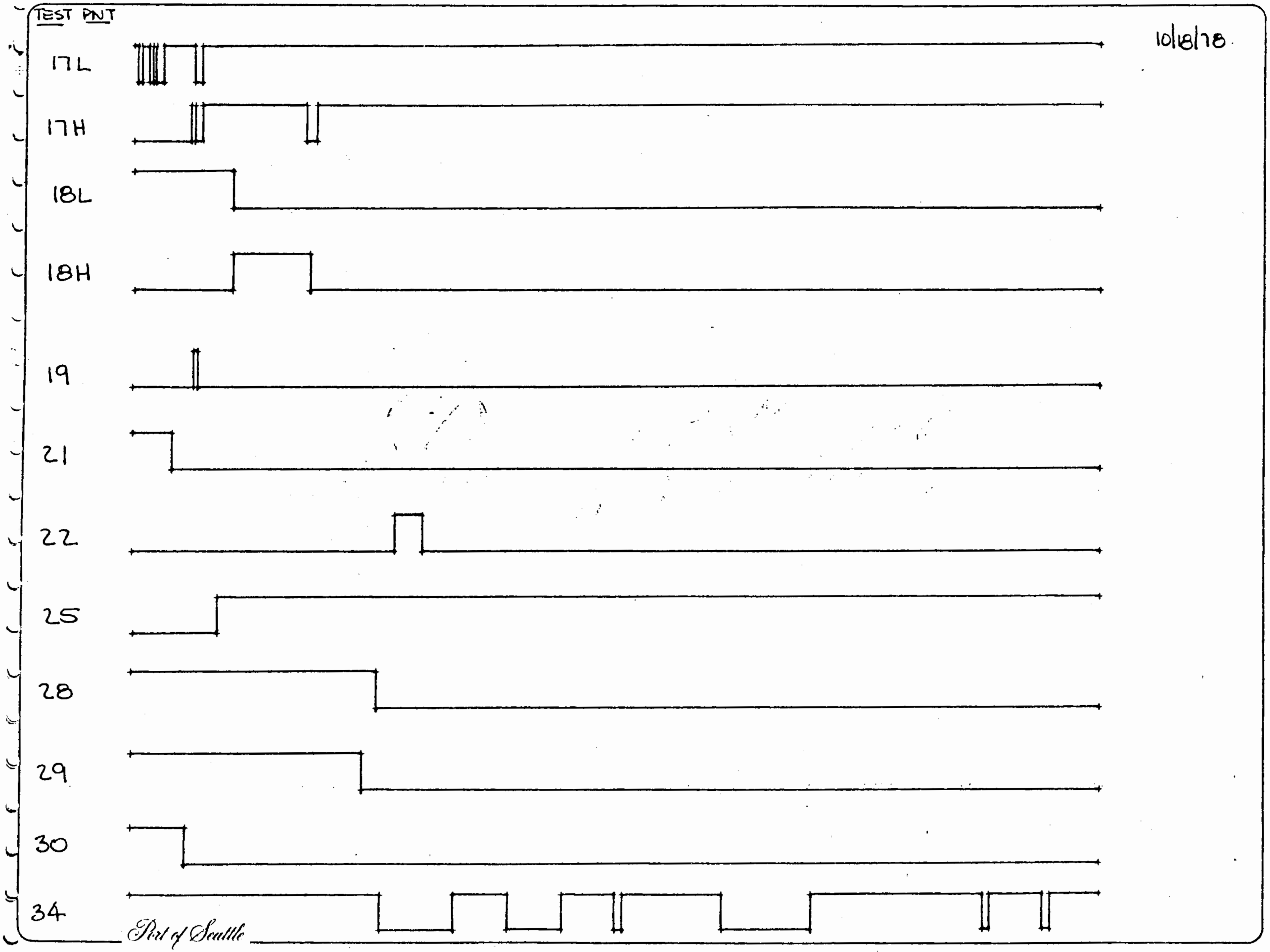
(TP 1 THRU 34)

DATA PNT	TIME (SEC)	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
----	-----																														
940	1554.9	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	1.4	1	1	0	0	0	0	0	0	0	1	0	0	0	1
941	1555	1	1	1	1	0	0	00	1	1	0	0	00	00	01	00	1.4	1	1	0	0	0	0	0	0	0	1	0	0	0	1
942	1555.1	1	1	1	1	0	0	00	1	1	0	0	00	00	01	00	1.2	1	1	0	0	0	0	0	0	0	1	0	0	0	1
943	1555.2	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	1.2	1	1	0	0	0	0	0	0	0	1	0	0	0	1
944	1555.3	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1
945	1555.4	1	1	1	1	0	0	00	1	1	0	0	00	00	01	00	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1
946	1555.5	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	.8	1	1	0	0	0	0	0	0	0	1	0	0	0	1
947	1555.8	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	.6	1	1	0	0	0	0	0	0	0	1	0	0	0	1
948	1556	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	.4	1	1	0	0	0	0	0	0	0	1	0	0	0	1
949	1556.2	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	.4	0	1	0	0	0	0	0	0	0	1	0	0	0	1
950	1556.3	1	1	1	1	0	0	00	1	1	0	0	00	00	01	00	.2	1	1	0	0	0	0	0	0	0	1	0	0	0	1
951	1556.4	1	1	1	1	0	0	00	1	1	0	0	00	00	01	00	.2	0	1	0	0	0	0	0	0	0	1	0	0	0	1
952	1556.5	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	.2	0	1	0	0	0	0	0	0	0	1	0	0	0	1
953	1556.6	1	1	1	1	0	0	00	1	1	0	0	00	10	01	00	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1
954	1556.7	1	1	1	1	0	0	00	1	1	0	0	00	11	01	00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
955	1556.9	1	1	1	1	0	0	00	1	1	0	0	00	11	01	00	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
956	1557	1	1	1	1	0	0	00	1	1	0	0	00	11	01	00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
957	1557.2	1	1	1	1	0	0	00	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
958	1557.4	1	0	1	1	0	0	00	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
959	1557.7	1	0	1	1	0	0	01	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
960	1557.8	0	0	1	1	0	0	01	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
961	1559.2	0	0	1	1	0	0	10	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
962	1559.3	0	0	0	1	0	0	10	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
963	1559.4	0	0	0	1	0	0	10	1	0	1	0	00	10	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
964	1559.5	0	1	0	1	0	0	10	1	0	1	0	00	10	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
965	1559.7	1	1	0	1	0	0	10	1	0	1	0	00	11	01	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
966	1559.8	1	1	0	1	0	0	10	1	0	1	0	00	11	00	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
967	1560.8	1	0	0	1	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
968	1560.9	1	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
969	1561.3	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
970	1563.3	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
971	1564.5	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
972	1566	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
973	1567.2	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
974	1568.7	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
975	1574.5	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
976	1575.8	0	0	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
977	1575.9	0	1	0	0	0	0	10	1	0	1	0	00	11	00	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
978	1576	0	1	0	1	0	0	10	1	0	1	0	00	11	00	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
979	1576.1	1	1	0	1	0	0	10	1	0	1	0	00	11	00	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
980	1577	1	1	0	1	0	0	10	1	0	1	0	00	11	00	00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0

LEGEND



Data Analysis : Vehicle failure #2
(based on data outputed 10/18/78)


TEST PNT
10/8/78
17L
17H
18L
18H
19
21
22
25
28
29
30
34
Port of Seattle

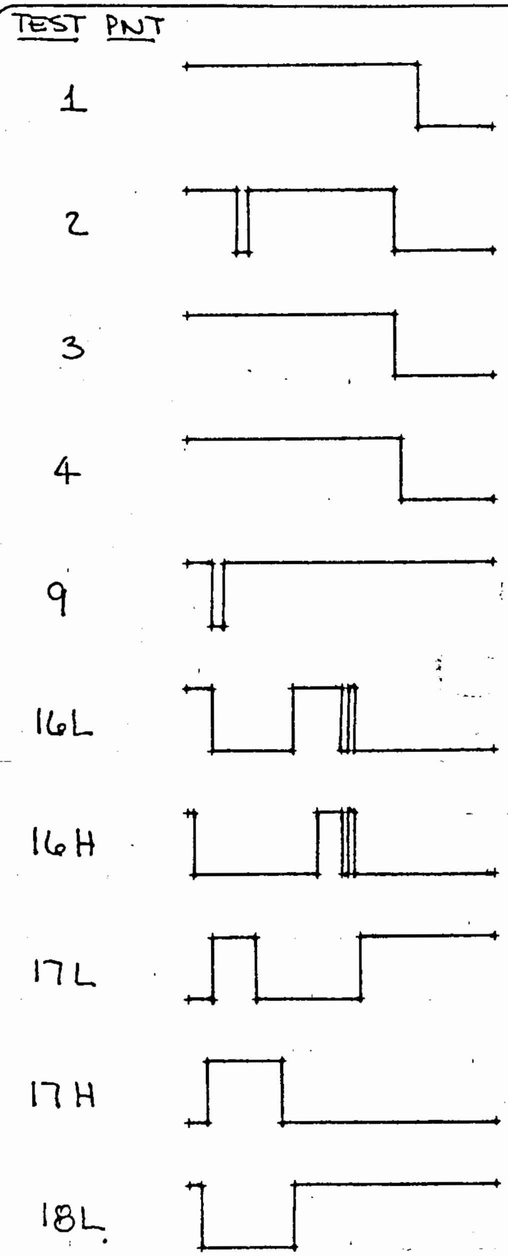
10/18/78

STOP IN LINE 31000

TEST DATA
(TP 1 THRU 34)

DATA PNT	TIME (SEC)	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
6680	3880.2	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	1.2	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6681	3880.3	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	1.2	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6682	3880.4	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	1.2	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6683	3880.5	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	1	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6684	3880.6	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	1	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6685	3880.7	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	1	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6686	3880.8	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	.8	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6687	3881	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	.6	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6688	3881.2	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	.4	1	0	0	0	0	1	0	1	1	1	0	0	0	1
6689	3881.3	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	.4	0	0	0	0	0	1	0	1	1	1	0	0	0	1
6690	3881.5	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	.2	0	0	0	0	0	1	0	1	1	1	0	0	0	1
6691	3881.7	1	1	1	1	0	0	00	0	1	0	0	00	10	10	00	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1
6692	3881.8	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
6693	3881.9	1	1	1	1	0	0	00	0	1	0	0	00	00	10	11	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
6694	3882	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
6695	3882.1	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
6696	3882.5	1	0	1	1	0	0	00	0	1	1	0	00	11	10	00	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
6697	3882.6	1	0	1	1	0	0	00	0	0	1	0	00	11	10	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6698	3882.9	0	0	1	1	0	0	00	0	0	1	0	00	11	10	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6699	3883	0	0	1	1	0	0	00	0	0	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6700	3883.1	0	0	1	1	0	0	01	0	0	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6701	3883.6	0	0	1	1	0	0	01	0	1	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6702	3884.6	0	0	1	1	0	0	10	0	1	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6703	3884.8	0	0	0	1	0	0	10	0	1	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6704	3884.9	0	1	0	1	0	0	10	0	1	1	0	00	11	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6705	3885.1	1	1	0	1	0	0	10	0	1	1	0	00	10	01	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6706	3885.2	1	1	0	1	0	0	10	0	1	1	0	00	10	00	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6707	3885.4	1	1	0	1	0	0	10	0	1	1	0	00	11	00	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6708	3886.2	1	1	0	1	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
6709	3886.7	1	0	0	1	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
6710	3886.8	1	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
6711	3887	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
6712	3887.1	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6713	3887.2	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6714	3887.6	0	1	0	1	0	0	10	0	0	1	0	00	11	00	00	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
6715	3887.8	1	1	0	1	0	0	10	0	0	1	0	00	11	00	00	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
6716	3888.3	1	0	0	1	0	0	10	0	0	1	0	00	11	00	00	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
6717	3888.4	1	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6718	3888.7	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6719	3889.3	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6720	3890.8	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6721	3892.3	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6722	3893.8	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6723	3894	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6724	3896.8	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6725	3899.3	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6726	3904.1	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6727	3904.3	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6728	3905.8	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6729	3906	0	0	0	0	0	0	10	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
6730	3907.4	0	0	0	0	0	0	01	0	0	1	0	00	11	00	00	0	0	0	0	0	1	1	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1

Data Analysis : Vehicle failure # 3
(based on data outputed 10/30/18)



TEST PNT

18 H

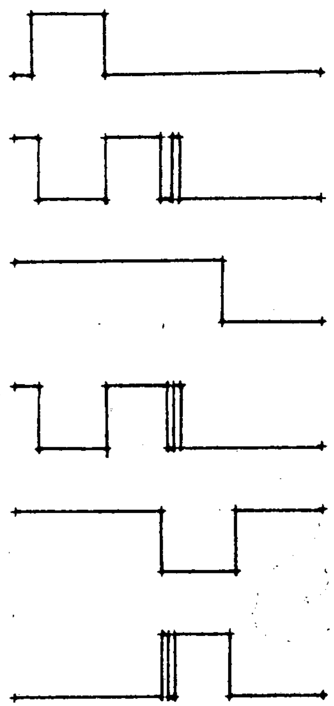
19 L&H

22

24

30

33



10/30/78

Port of Seattle

10/30/78

STOP IN LINE 31000

TEST DATA
(TP 1 THRU 34)

DATA PNT	TIME (SEC)	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
6135	6449.4	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6136	6449.5	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6137	6449.9	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6138	6450	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6139	6450.8	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6140	6450.9	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6141	6451.7	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6142	6451.9	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6143	6453.3	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6144	6453.7	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6145	6453.9	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6146	6454.6	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6147	6454.7	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6148	6455.8	1	1	1	1	0	0	00	0	1	0	0	10	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6149	6455.9	1	1	1	1	0	0	00	0	1	0	0	10	00	10	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6150	6456	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6151	6456.1	1	1	1	1	0	0	00	0	1	0	0	10	00	10	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6152	6456.2	1	1	1	1	0	0	00	0	1	0	0	10	00	00	11	14.8	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6153	6456.3	1	1	1	1	0	0	00	0	1	0	0	10	00	01	11	14.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6154	6456.4	1	1	1	1	0	0	00	0	0	0	0	00	10	01	00	14.6	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6155	6456.6	1	1	1	1	0	0	00	0	1	0	0	00	10	01	00	14.6	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6156	6456.7	1	1	1	1	0	0	00	0	1	0	0	00	10	01	00	14.8	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6157	6456.8	1	0	1	1	0	0	00	0	1	0	0	00	11	01	00	14.6	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6158	6456.9	1	0	1	1	0	0	00	0	1	0	0	00	10	01	00	14.4	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6159	6457	1	1	1	1	0	0	00	0	1	0	0	00	10	01	00	14.2	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6160	6457.1	1	1	1	1	0	0	00	0	1	0	0	00	00	01	00	14	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6161	6457.2	1	1	1	1	0	0	00	0	1	0	0	00	00	01	00	13.6	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6162	6457.4	1	1	1	1	0	0	00	0	1	0	0	00	00	01	00	13.4	1	1	0	0	0	0	0	0	0	1	0	0	0	1
6163	6457.5	1	1	1	1	0	0	00	0	1	0	0	00	00	00	11	13.4	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6164	6457.7	1	1	1	1	0	0	00	0	1	0	0	10	00	10	11	13.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6165	6457.8	1	1	1	1	0	0	00	0	1	0	0	10	00	10	11	13.4	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6166	6458.1	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	13.4	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6167	6458.3	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	13.6	1	1	0	1	0	0	0	0	0	1	0	0	0	1
6168	6458.4	1	1	1	1	0	0	00	0	1	0	0	11	00	10	00	13.6	1	1	0	1	0	0	0	0	0	0	0	0	1	1
6169	6458.5	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	13.6	1	1	0	0	0	0	0	0	0	0	0	0	0	1
6170	6458.6	1	1	1	1	0	0	00	0	1	0	0	11	00	10	11	13.6	1	1	0	1	0	0	0	0	0	0	0	0	1	1
6171	6458.7	1	1	1	1	0	0	00	0	1	0	0	00	00	10	00	13.8	1	1	0	0	0	0	0	0	0	0	0	0	1	1
6172	6458.8	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	13.4	1	1	0	0	0	0	0	0	0	0	0	0	1	1
6173	6458.9	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	13.6	1	1	0	0	0	0	0	0	0	0	0	0	1	1
6174	6459.2	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	12.8	1	1	0	0	0	0	0	0	0	0	0	0	1	1
6175	6459.3	1	1	1	1	0	0	00	0	1	0	0	00	11	10	00	12.6	1	1	0	0	0	0	0	0	0	0	0	0	1	1
6176	6459.4	1	0	0	1	0	0	00	0	1	0	0	00	11	10	00	12.4	1	0	0	0	0	0	0	0	0	0	0	0	1	1
6177	6459.5	1	0	0	0	0	0	00	0	1	0	0	00	11	10	00	11.8	1	0	0	0	0	0	0	0	0	0	0	0	0	1
6178	6459.6	1	0	0	0	0	0	00	0	1	0	0	00	11	10	00	11.6	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6179	6459.7	1	0	0	0	0	0	00	0	1	0	0	00	11	10	00	11.2	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6180	6459.8	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	10.8	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6181	6459.9	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	10.2	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6182	6460	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	9.8	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6183	6460.1	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	9.4	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6184	6460.2	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	8.8	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6185	6460.3	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	8.4	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6186	6460.4	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	7.8	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6187	6460.5	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	7.4	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6188	6460.6	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	7	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6189	6460.7	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	6.6	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6190	6460.8	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	6	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6191	6460.9	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	5.4	1	0	0	0	0	0	0	0	0	1	0	0	0	1
6192	6461	0	0	0	0	0	0	00	0	1	0	0	00	11	10	00	5	1	0	0	0	0	0	0	0	0	1	0	0	0	1