

AD-A140 299

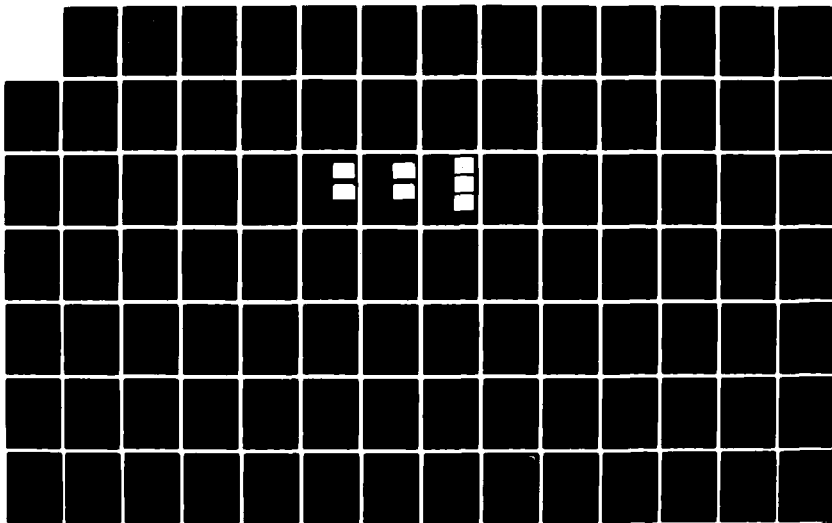
ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE
OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83
DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

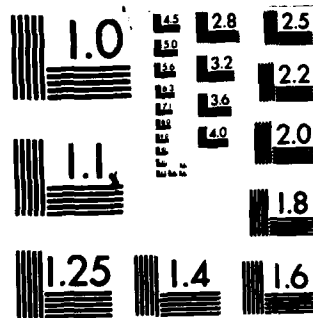
1/4

UNCLASSIFIED

F/G 9/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 140299

DTIC FILE COPY

L.E. Lesster
D.B. Zeitlin
W.B. Hall

**David W. Taylor Ship Research and Development Center
Annapolis Laboratory
Annapolis, Maryland 21402**

WESTINGHOUSE ELECTRIC CORPORATION
Oceanic Division
P.O. Box 1488
Annapolis, Maryland 21404

APR 20 1984

A

84 04 19 083

Table of Contents

This volume is divided into three separate reports which document the three stages of development.

Volume Summary

A. Phase I Report

Flux Synthesis and PWM Synthesis Techniques Theory and Evaluation

B. Phase IIA Report

Development of PWM and Flux Feedback Control Microprocessor Circuits and Software

C. Phase IIB Report

Development of Three Phase Power Bridge and Evaluation of Motor Controller

Volume Summary

The three reports assembled in this volume represent work performed periodically over a span of 5 years, as part of a step-by-step development program.

The purpose of ^{this} the development program was to investigate a promising alternative technique for control of a squirrel cage induction motor for subsea propulsion or hydraulic power applications. The technique uses microprocessor based generation of the pulse width modulation waveforms, which in turn permits use of a true integral volt-second pulse width control for the generation of low harmonic content sine waves from a 3 phase ³ 'Graetz' transistor power bridge.

In conjunction with the PWM Technique the novel use of a microprocessor based calculation of a signal to represent air gap flux in the motor was attempted. Externally this involves only sampling output currents and voltages to the motor and does not involve intrusion into the motor to place sense windings or Hall effect devices.

The purpose of this latter investigation was to generate a signal proportional to air gap flux which could be used in feedback control of the PWM microprocessor to vary the output pulsewidths of the bridge. This in turn could be used to maintain the air gap flux in the motor constant despite varying load.

During development it became apparent that keeping the air gap flux level constant did not represent an efficient control mechanism for the motor. This technique evolved from the normal use of induction motors on fixed frequency where better performance could be obtained on heavy load if the supply voltage could be boosted to maintain air gap flux despite increased voltage drop in the stator resistance and leakage inductance.

In reality an unloaded or lightly loaded induction motor can be run at very much lower voltages than 'nominal' for any given frequency and the purpose of feedback should be to increase the voltage approximately with load to maintain just sufficient flux to prevent under-excited operation with consequent high slip frequencies.

The use of the air gap flux synthesis technique appeared to work in principle, but difficulties were encountered in using it at low frequencies. Calculation of the feedback signal was very sensitive to the phase relationship of the motor current and voltage, and distortion in the feedback pre-sampling filters which is evident in the low frequency oscilloscope pictures (Phase II, Task B Report) is of serious consequence here. Further, the air gap flux signal as derived via a reactive power calculation is double valued and for use as a regulation signal, care must be taken to ensure that the load remains always on the over-excited side or always on the under-excited side of the optimum excitation level for any particular load and frequency. While this can be assumed statically, in a dynamic loading situation, unless the feedback loop is fast compared with load change this cannot be assured.

Session For
DTIS GRA:
DTIC TAB
UN
J
A-1

File on file



For purposes of checking out other key points of the motor controller development independently of the flux feedback signal, a slight modification of the feedback signal was made. This permitted sensing and feedback of the motor load current and provided satisfactory demonstration and of overload protection by ramping down in frequency. During this mode of operation the output voltage was controlled to have essentially constant volts/Hz except for the drop due to stator resistive and reactive components. Satisfactory operation in this mode indicates that regulation at some constant air gap flux would be possible although not necessarily desirable from an optimum control point of view.

In summary, the following was achieved:

1. Full microprocessor based, sine wave, 3 phase, variable frequency, variable amplitude output waveform generation with volt-second control of the pulsewidths based on $\int v dt$ calculations. This was performed using an 8 bit M6800 processor in conjunction with pulse timer modules in an interrupt driven configuration.
2. Proof of principle of air gap flux synthesis technique and demonstration of overload ramp down dynamic load shedding.
3. Construction and test of a 7.5 kw 3 phase transistor bridge driving a 32 Hz induction motor.
4. Test of the complete motor controller on a dynamometer, and test of all principles involved.

These tests showed a sound hardware design which provided high quality sine wave currents to the motor over extended periods of time at all power levels. These tests are showed that with some extra investigation in the feedback area, an extremely viable, high efficiency motor controller for deep ocean application has been developed.

ADVANCED MOTOR CONTROLLER

PHASE I REPORT

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
SUMMARY & RECOMMENDATIONS	1
DISCUSSION	3
Flux Synthesis Control Loop (Derivation)	3
PWM Synthesis	6
1. Background	6
2. Hardware	6
3. Software	8
4. Harmonic Analysis	19
System Operation	30
APPENDICES	
A. Derivation of Flux Synthesis and Flux Loop Program	A 1
B. Basic Aspects of Modulation Strategy	B 1
C. Harmonic Analysis Program Output	C 1
D. Average Values of PWM Steps - Computer Output	D 1

MICROPROCESSOR - BASED THREE PHASE MOTOR CONTROLLER

Introduction

The objective of this study was to define an approach to digital synthesis of waveforms for a low harmonic content sine wave output PWM 3 ϕ variable frequency transistorized induction motor drive inverter, to analyze an air gap flux synthesis system using microprocessor supervisory control and to provide block diagram information and block response characteristics for computer modeling of the motor drive system.

Summary & Recommendation

Initially it was thought that an M6800 microprocessor would be too slow for PWM generation for this application. However, careful analysis of the capabilities of the processor as calculator of numbers rather than a generator of timing waveforms led to consideration of its use to calculate pulsewidths and timing intervals, numbers which can be used to control a single chip 3-channel timer. This component turns the numbers into serial outputs as a function of time - the pulse width modulation waveforms - which are amplified by the slave power switches of a 3 ϕ power bridge. Amplitude scaling of the output as a function of demanded motor voltage, controlled via the air gap flux feedback loop, is effected by digital multiplication using a high speed hardware multiplier.

The air gap flux synthesis and feedback control loop relies on sensing the voltages and currents to the motor and simulating the magnetizing component of the reactive power drawn by the motor. Since the output waveforms from the bridge to the motor are not smooth sinusoides, the sensed inputs from these must be filtered by equal filters to maintain phase relationship between current and voltage. Once filtered, analog/digital conversion turns these inputs into a form suitable for a microprocessor calculation routine which modifies the voltage output of the PWM drive in response to motor load requirements.

The PWM subsystem block was designed in software and hardware using an AMI 6800 microprocessor development system for the software development and an AMI prototyping general purpose microprocessor board for hardware. Pictures of waveforms are provided.

Imperfections in the triangle crossing modulation approach, which is an easy-to-implement technique in analog circuitry, were observed. The microprocessor approach to PWM permits use of an optimum integral technique which noticeably reduces harmonics.

The work performed in this study has covered the basic requirements for microprocessor control of a sine wave output motor drive system and in addition has developed a novel approach to PWM synthesis. It is recommended that further work in this area be addressed to tying the PWM control and air gap flux synthesis loops together in both software and hardware and reduction to practice of a full inverter/motor system.

Discussion

Flux Synthesis Control Loop

In attempting to control the air gap flux in an induction motor it is necessary to maintain a given magnetizing current in the magnetizing inductance component of the primary (stator) winding of the motor. In the equivalent circuit of the induction motor (Figure 1) it is the current I_m which is to be kept constant in the presence of varying load currents and hence varying voltage drops across R_1 , L_1 and changes with temperature of winding resistances.

The input power to the motor can be considered to have two components. The true power $VI \cos \phi$ and the reactive power $VI \sin \phi$ or, if $V = V_1 \sin \omega t$ & $I = I_1 \sin(\omega t - \phi)$ the power components are $1/2 VI \cos \phi$ and $1/2 VI \sin \phi$. The true power input includes power lost in winding resistances and iron loss as well as in equivalent load resistor. The reactive power component, however, by definition is independent of all the resistance components and represents the product of the terminal voltage and the quadrature currents flowing in the leakage inductance L_1 , the magnetizing inductance L_m and the secondary (rotor) inductance L_2 .

These inductances are fixed once the motor is made and do not change with load, etc. It remains then to find the 'voltage drop' in L_1 due to I_m and I_2 so that the 'voltage' applied to L_m and hence the magnetizing current I_m can be found.

The flux control loop block diagram is shown in Figure 2, and the full derivation as well as the microprocessor assembly language program for flux loop control is given in Appendix A. During the development of the block diagram, numerical constants were selected on the basis of measurements made on the motor chosen for testing. Closed loop operation with that motor has not been performed and it is anticipated that some adjustments to the constants will be required for proper operation in that mode.

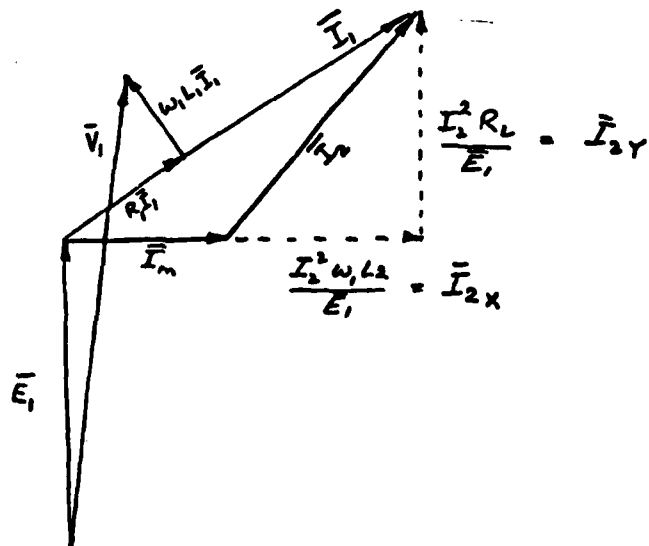
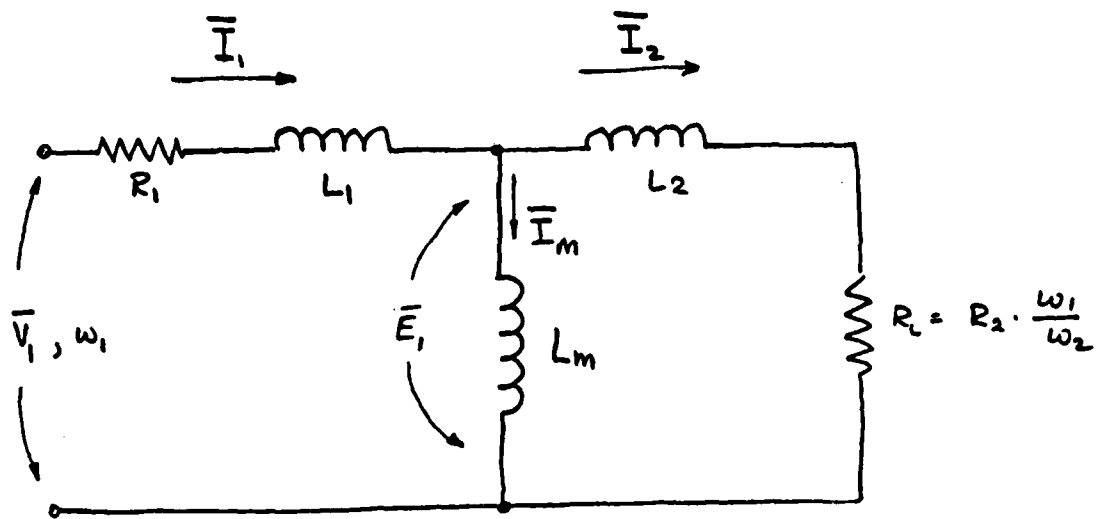


FIG 1 - Equivalent Circuit of Induction Motor

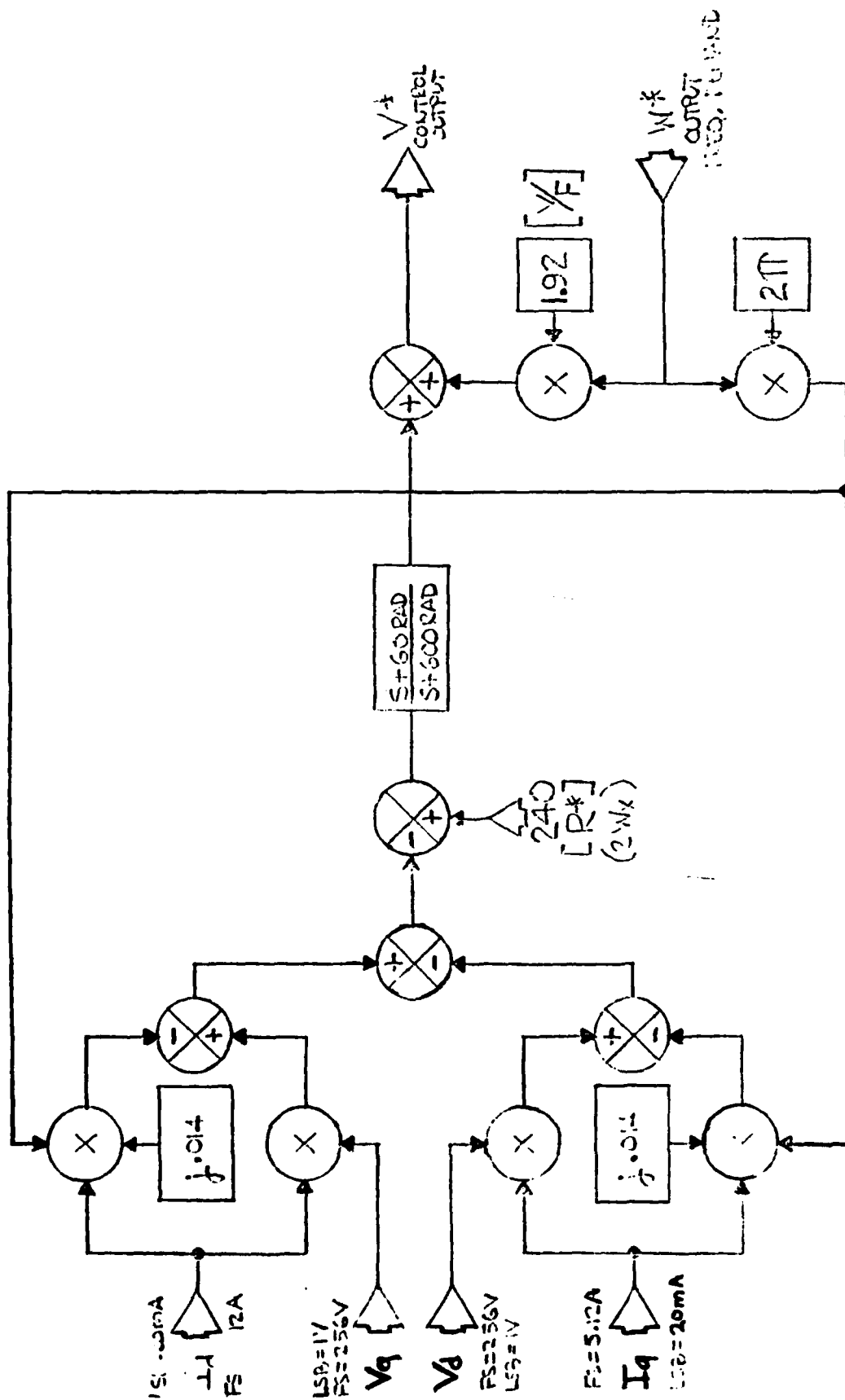


FIGURE 2
μP CONTROLLED AC MOTOR DRIVE
FLUX LOOP CALCULATION BLOCK DIAGRAM

Pulse Width Modulation Synthesis

Background

Initially it was thought that the M6800 would be too slow to generate the pulse width modulated wave forms. However, a closer look at the problem, as outlined in Appendix B, indicated that by using table look up and a hardware multiplier it should be possible to implement algorithms which generate time related bytes from which the serial PWM signals for the three phases can be generated. Such a system was devised and is described below.

Hardware

Figure 3 shows a functional block diagram of the hardware used to create the PWM waveforms. The central processing unit is an AMI 6800 with a 1 MHz clock, 8K bytes of EPROM for program and timing sequence data storage, 1K bytes of RAM for scratchpad storage, a Pulse Timer Module containing 3 independent timers with output signals, an 8x8 bit multiplier to speed up calculations and a keyboard interface used to command different voltages and frequencies.

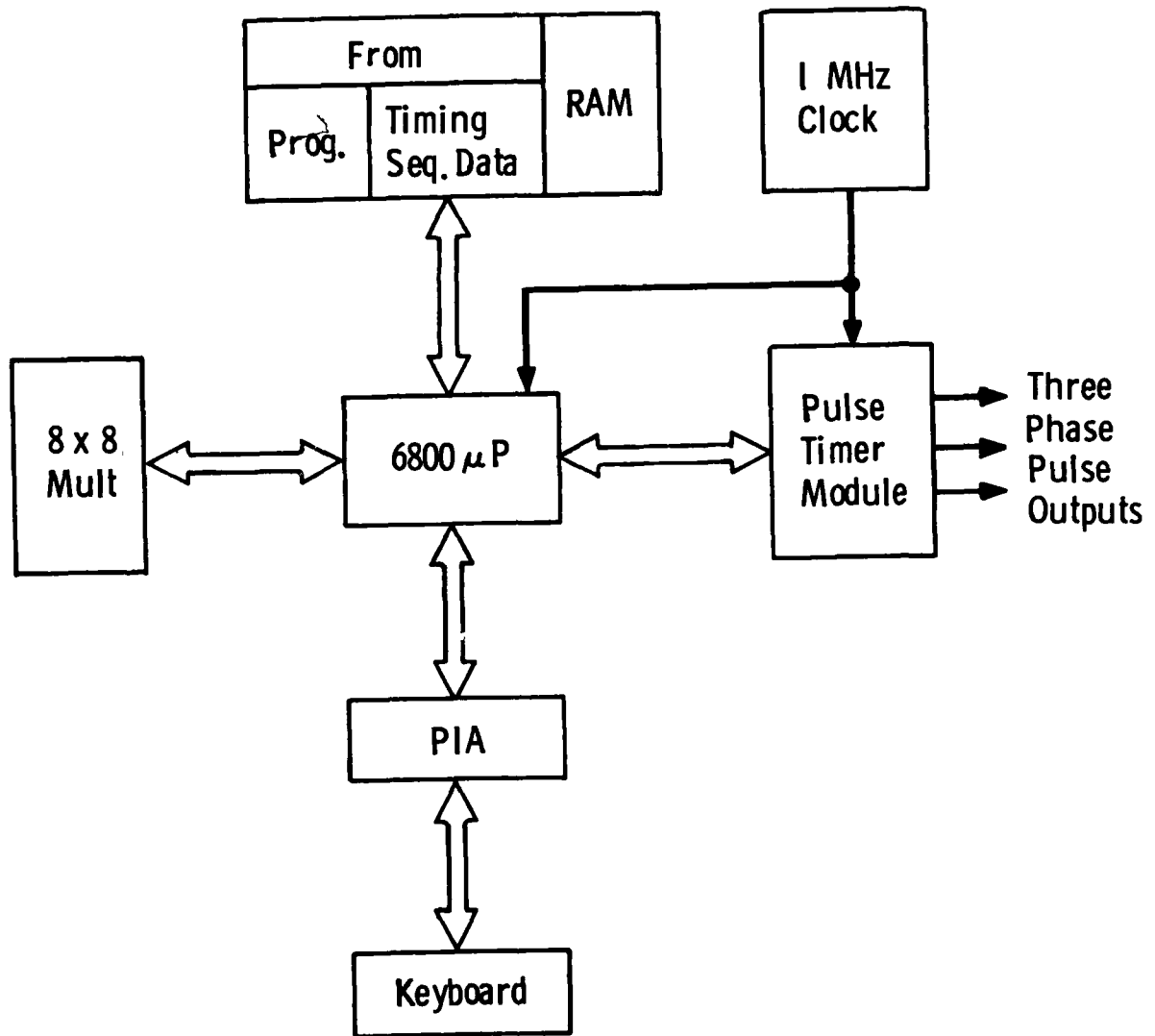


Figure 3 HARDWARE

Software

A frequency table is stored in PROM for each frequency that is to be output. It contains all the information required to generate the PWM waveforms for a number of voltage steps. Stored in the Frequency Table are:

1. The number of pulse periods in a 60 degree segment of the output sine wave (n).
2. The number of timer clocks in a pulse period (NCLKS).
3. A scale factor table containing the scaling data for a number of different voltage steps.
4. A baseline timing sequence for a given voltage containing, for each pulse period, the width of the dominant pulse (t_d), the width of the first complementary pulse (t_{c1}) and the centerpoint of the dominant pulse (CP).

These parameters are sufficient to generate timing sequences for a large number of voltage steps for a given frequency while at the same time keeping the memory requirements low.

Synthesizing the PWM waveforms for a given frequency and voltage requires generating the proper clock counts to the three timers in the Pulse Timer Module. (The timer outputs change state each time the counter reaches zero.) In order to do this, some temporary values must be defined. See Figure 4 for timing relationships.

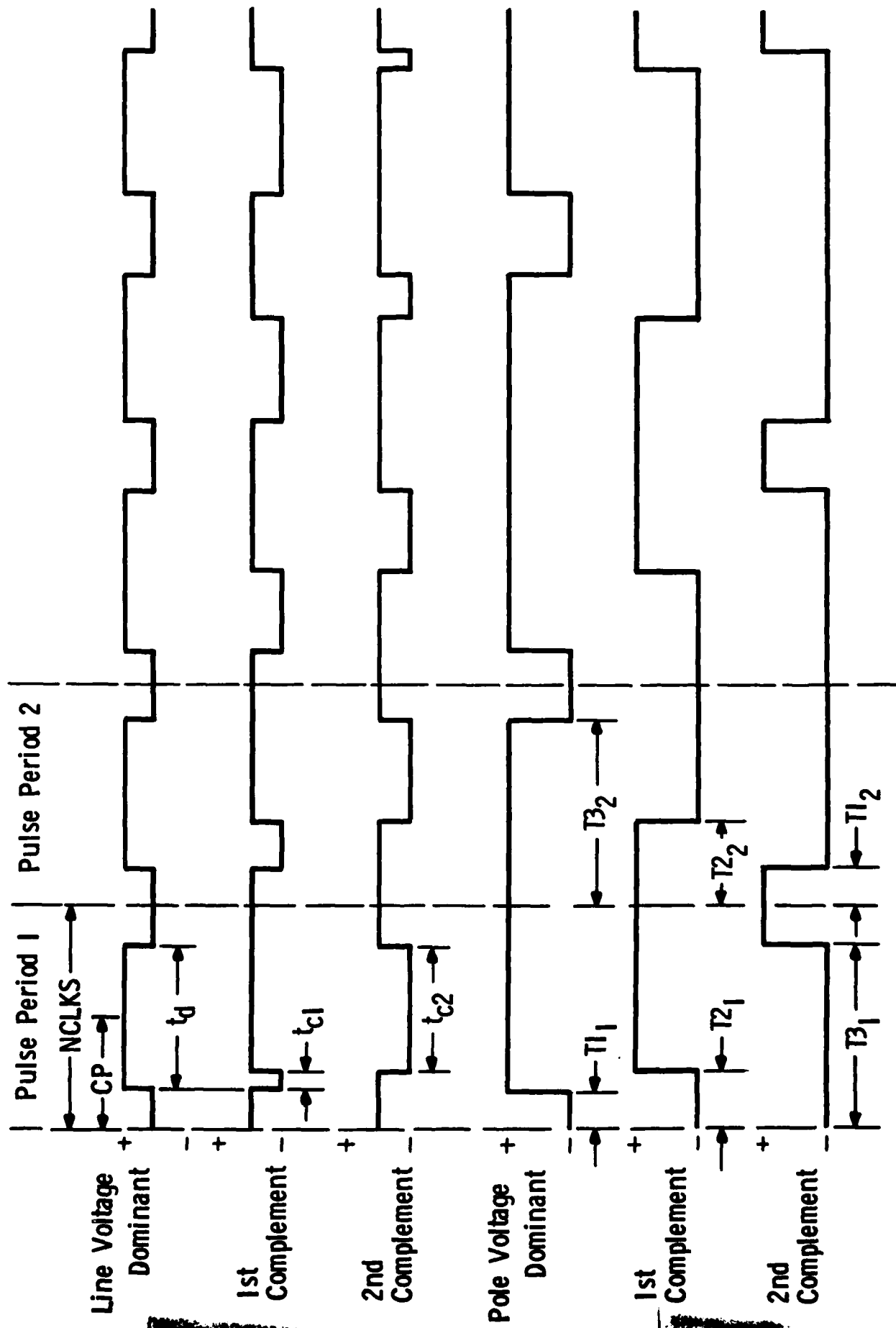


Figure 4 CONTROLLER TIMING DIAGRAM

T1 = time from start of a pulse period to the beginning of the dominant pulse.

T2 = time from start of a pulse period to the beginning of the 1st complementary pulse.

T3 = time from start of a pulse period to the beginning of the 2nd complementary pulse.

It can be seen that these times correspond to switching times where:

T2 = switching time of the 1st complementary pole voltage

T1 and T3 = switching time of the dominant or 2nd complementary pole depending on the pulse period.

The pulse lengths will vary proportionally with the amplitude of the output sine wave during the pulse interval. Therefore, the stored pulse widths in the baseline sequence must be scaled to the proper voltage. The scale factors are stored in the scale factor table contained in the frequency table and are computed such that:

$$SF = \frac{V \text{ desired}}{V \text{ baseline}}$$

The scaled pulsewidths now become:

$$t'_d = t_d * SF$$

$$t'_{c1} = t_{c1} * SF$$

where t_d and t_{c1} are the pulse widths stored in the baseline timing sequence for this pulse interval.

It can be seen that:

$$T1 = CP - t_d/2$$

$$T2 = T1 + t_c$$

$$T3 = T1 + t_d$$

In order to resolve the phase polarities and keep the optimal switching sequence, for the i^{th} pulse period,

$$TC1_i = T2$$

and, if i is odd

$$TD_i = T1_i$$

$$TC2_i = T3_i$$

else, if i is even

$$TD_i = T3_i$$

$$TC2_i = T1_i$$

where TD_i , $TC1_i$ and $TC2_i$ are the pole switching times of the dominant, 1st complementary and 2nd complementary phases, respectively.

Now, since a single timer controls each pole, the time difference between firings on each pole must be computed and saved in memory for the pulse timer module controller.

$$TD_i = TD_i - TD_{i-1} + NCLKS$$

$$TC1_i = TC1_i - TC1_{i-1} + NCLKS$$

$$TC2_i = TC2_i - TC2_{i-1} + NCLKS$$

These counts are computed for each pulse interval and saved in three memory buffers, one for each timer. Also saved is the time from the last switching of each phase to the end of the pulse interval ($NCLKS-TD_n$, $NCLKS-TC1_n$, $NCLKS-TC2_n$). This is done to keep the phase relationships constant by forcing each phase to execute an identical number of clocks in a 60° segment. Figure 5 shows the contents of the computed timing buffers.

With the timing counts now computed and stored in memory it becomes a matter of getting the proper count to the proper timer to preserve the phase relationships between the line voltages. Each of the three timers controls a single phase of the three phase signal.

When one of the timers counts down to zero it sends an interrupt to the processor. At that time the processor outputs the next count from the timing buffer to the timer latches, checks to see if this is the end of a 60° segment and sets up for the next interrupt. This procedure is followed by each of the three timers until the end of the 60° segment.

At the boundary between segments, the processor must resolve phases and insure that the timer clock counts for each of the three timers over the segment are identical. It does this by adding the remaining time in the last pulse interval (e.g. $NCLKS-TD_n$) to the first switching time of the proper phase (e.g. $TC2_1$) of the new timing buffer (see Figure 3). Also at this time, the phase identifier for the timer is updated to reflect its new status.

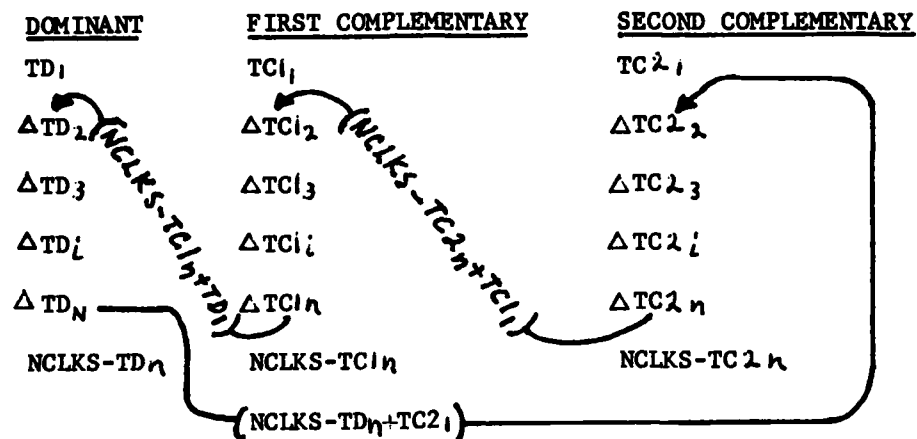


FIGURE 5. TIMING BUFFER CONTENTS AND SIGNAL FLOW FOR EACH OF THE THREE TIMERS

The boundary between segments is the only place that a new timing sequence can be started. This maintains the integrity of the system as well as insures the proper phasing relationships for all signals. Thus, if the same frequency were maintained over a 180° interval 3^*NCLKS counts would be output to each timer.

Two separate and independent routines control the synthesis of the PWM waveforms. The Computer Timing Sequence (CTS) routine accepts as inputs the voltage and frequency commands, computes the counts required for the timer controller and saves the counts in a timing buffer. The Pulse Timer Module Interrupt Service Routine (PTMIS) responds to interrupts from the timer, extracts the next count computed by CTS routine from the timing buffer and outputs it to the timer. Communication between the two routines requires two timing buffers. The CTS routine fills one buffer with the new timing sequence while the Pulse Timer Module Interrupt routine gets its counts from the other buffer. At the 60° segment boundary the timer controller routine switches to the newly completed buffer and flags the CTS routine to begin a new calculation. Figures 6 and 7 contain flow charts of the two routines.

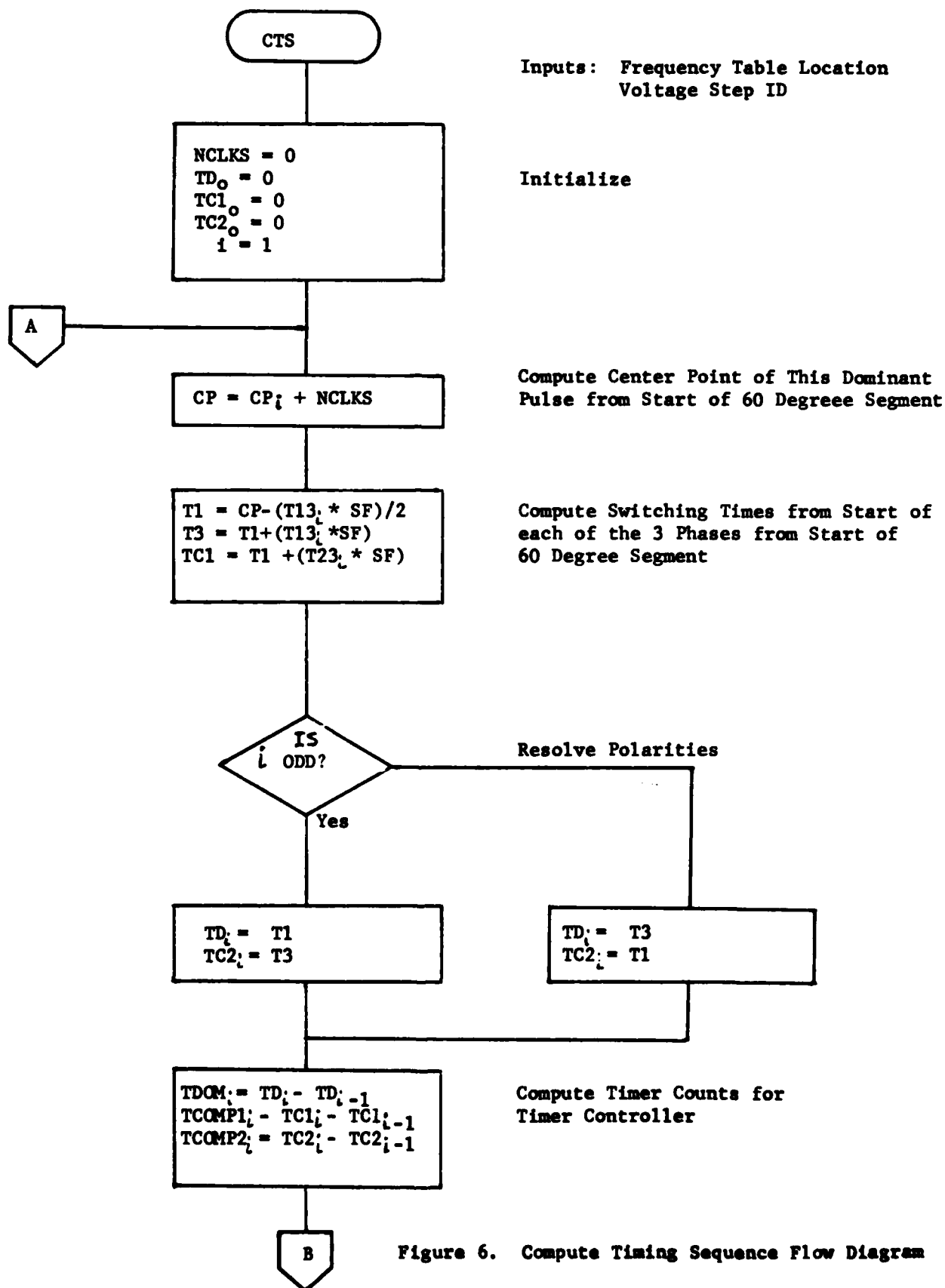


Figure 6. Compute Timing Sequence Flow Diagram

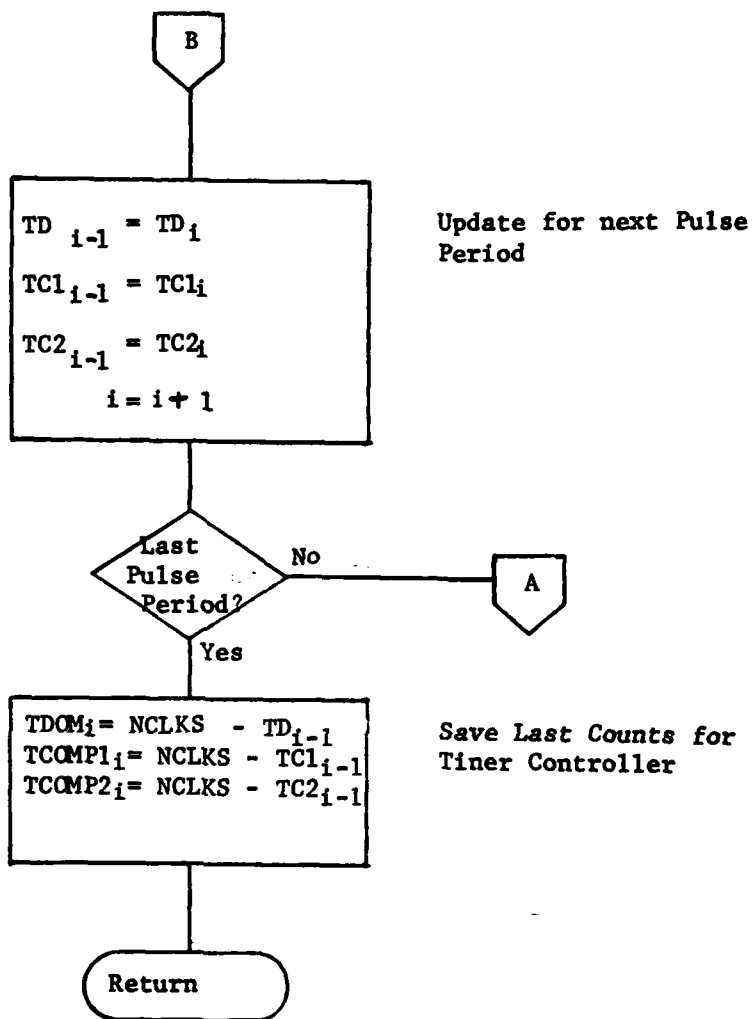


Figure 6. (cont.)

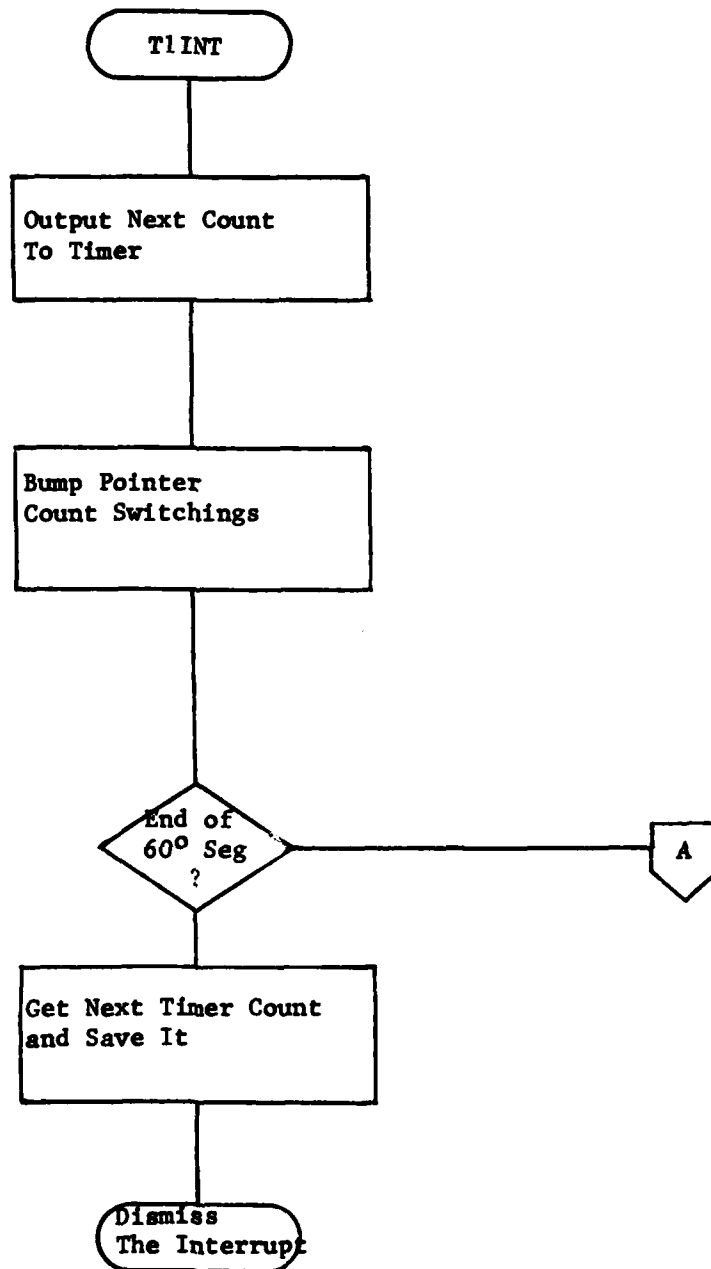


Figure 7. Time #1 Interrupt Service Routine Flow Diagram

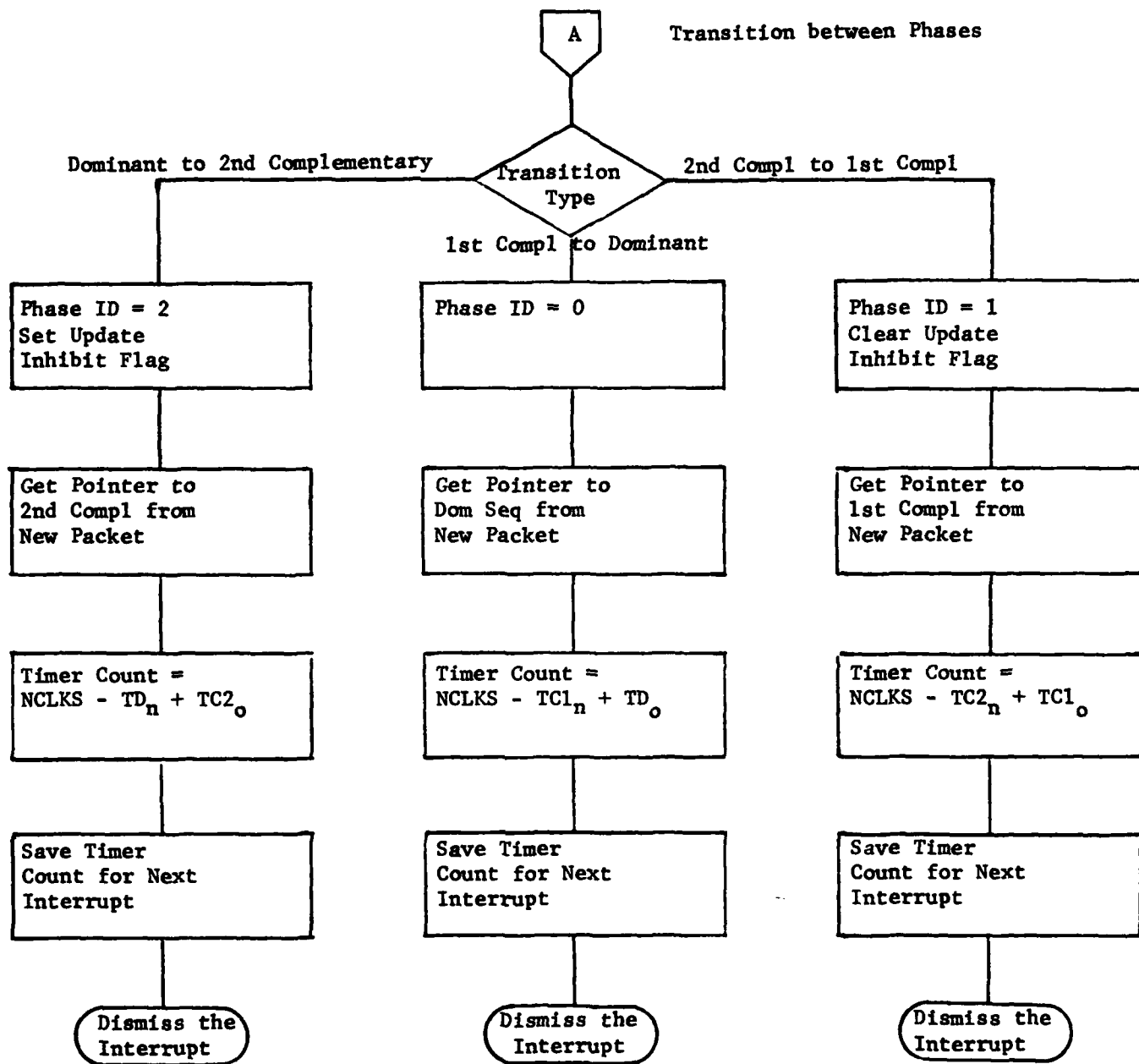


Figure 7. (cont.)

Harmonic Analysis

An harmonic analysis of the PWM waveform was performed in order to arrive at an optimum pulse train within the constraints of the modulation scheme. Harmonic content was traded off against the number of PWM steps and the position of the power-on portion within each step.

Computer programs were created to aid in the analysis and tradeoff study. These programs were written in BASIC and were run on both a Hewlett Packard 9830 and on CDC time sharing service.

The set of pole switching times used in the first portion of the project were derived from the traditional PWM scheme of carrier frequency triangle wave coincidence with the (sine wave) modulation waveform. This scheme has some weaknesses, which are shown below.

Let $f(x)$ be the modulating waveform, and let the triangle wave have a frequency higher than $f(x)$. Then Figure 8 represents one carrier cycle,

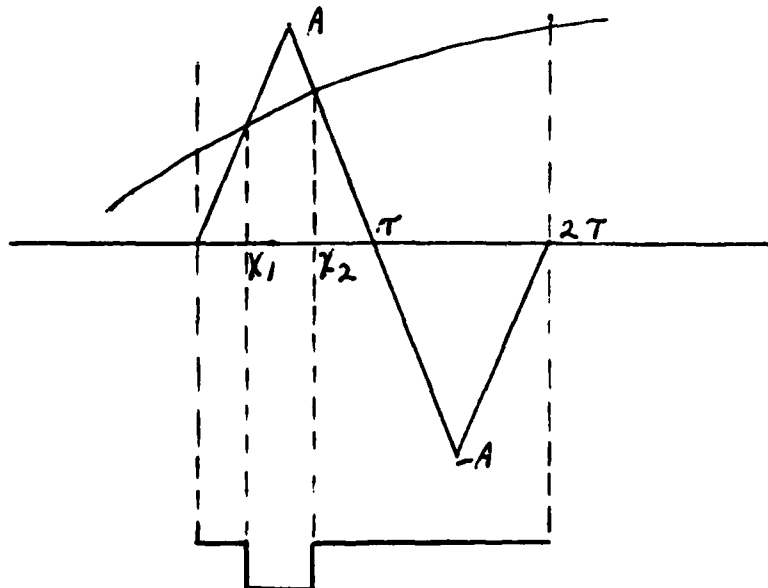


FIG. 8

$$[1] \quad f(x_1) = 2A \left(\frac{x_1}{T} \right)$$

$$[2] \quad f(x_2) = 2A \left(1 - \frac{x_2}{T} \right)$$

$$\text{OR } x_1 = f(x_1) \left(\frac{T}{2A} \right)$$

$$x_2 = \left(\frac{T}{2A} \right) [2A - f(x_2)]$$

$$[3] \quad \text{DUTY CYCLE AVERAGE, } D = \frac{T - (x_2 - x_1)}{2T}$$

$$\text{OR } D = \frac{f(x_2) + f(x_1)}{4A}$$

which is the average value of $f(x)$ from x_1 to x_2 using a trapezoidal approximation, but taken over the entire period of $2T$. That is, it is the average value of a straight line between $f(x_1)$ and $f(x_2)$, taken as if $f(x_1)$ and $f(x_2)$ were at the segment ends. Because this method extends the slope of its straight line approximation over the entire integration interval, but bases it only on the value of $f(x)$ at x_1 and x_2 , it yields values higher than the actual average for curves with negative second derivatives and lower than average for curves with positive second derivatives.

For carrier frequencies much greater than the modulating frequency this PWM scheme is generally acceptable due to its ease of implementation in an analog system. In a digital system, however, such as the one under development here, freedom to choose switching times without impacting the complexity of the modulation scheme allows more accurate output waveform synthesis.

Since the motor acts as an integrating filter, it is desirable to generate the PWM such that the duty cycle of each step is proportional to the integral of the output sine wave over the step period. A computer program was used to generate these integral values for all the possible numbers of steps to be considered. The results of the program, which computes

$$-G(X) = \frac{\cos(X_1) - \cos(X_2)}{(X_1 - X_2)}, \quad \text{appears in Appendix D.}$$

The harmonic content of the output wave is of concern in two areas. First, the odd harmonics must be sufficiently suppressed for efficient forward operation of the motor. Second, the output must be easily conditioned to provide clean waveforms for flux calculations.

A computer program was written to generate the desired PWM waveform for any combination of the carrier and output frequencies of interest, and then perform a harmonic analysis on the PWM. The on portion within each step may be chosen as beginning at the start of the step or as being centered within the step. Program output includes harmonic amplitudes relative to the fundamental. For the runs shown, levels below one percent were suppressed from the printout for clarity, although any printing threshold may be chosen when using this program. The program and its output appears in Appendix C.

Plotting and analyzing this output yielded a set of PWM steps per output cycle for each discrete output frequency from seven to sixty hertz. Initially all possibilities of steps per output cycle were permitted in order to learn what optimum combinations existed. Then, due to the mechanization of the modulation scheme in the microprocessor, only odd multiples of three steps per output cycle

were allowed in the analysis. The elimination of some of the even numbers of steps caused large jumps in carrier frequency at the higher output rates. This effect was eased somewhat by shifting all the carriers up to an average of about 700Hz from the previous 540 Hz. See Figures 9 and 10.

One goal in the manipulation of the harmonic data was to choose a combination which keeps unwanted frequencies in the flux loop input down at least 30 dB with only simple filtering. With the step/output combinations finally chosen, this goal was met with a 70 Hz pole low pass filter. The design and response of the test filter appear in Figures 11 and 12.

As a first test of the harmonic analysis and modulation scheme, one phase of the processor output was interfaced with a breadboard filter and a 540 Hz carrier with 36 Hz modulation was transmitted through it. The resultant sine wave observed on an oscilloscope looked good and was delayed about 4 ms as expected. Amplitude control with the processor was exercised satisfactorily. The 540 Hz carrier was used since values for it had been previously stored in the processor memory. Photographs of the oscilloscope observations appear in Figure 13. Carrier frequencies under 500 Hz produced unacceptably distorted output waveforms. Inspection of the waveform with only 9 steps per cycle chosen by the triangular-intercept method showed distortion due to the errors of that scheme.

Later, the one-third horsepower three phase motor was operated by processor generated signals through a motor driver and the phase A current monitored. Figure 14 shows photographs of the sensed current, both filtered for processing and unfiltered. These waveforms confirm the choice of PWM parameters.

EUGENE DIETZEN CO.
MADE IN U. S. A.

NO. 1211 "LITZGE" GRAPH PAPER
SEMI-LOGARITHMIC
2 CYCLES X 10 DIVISIONS PER INCH

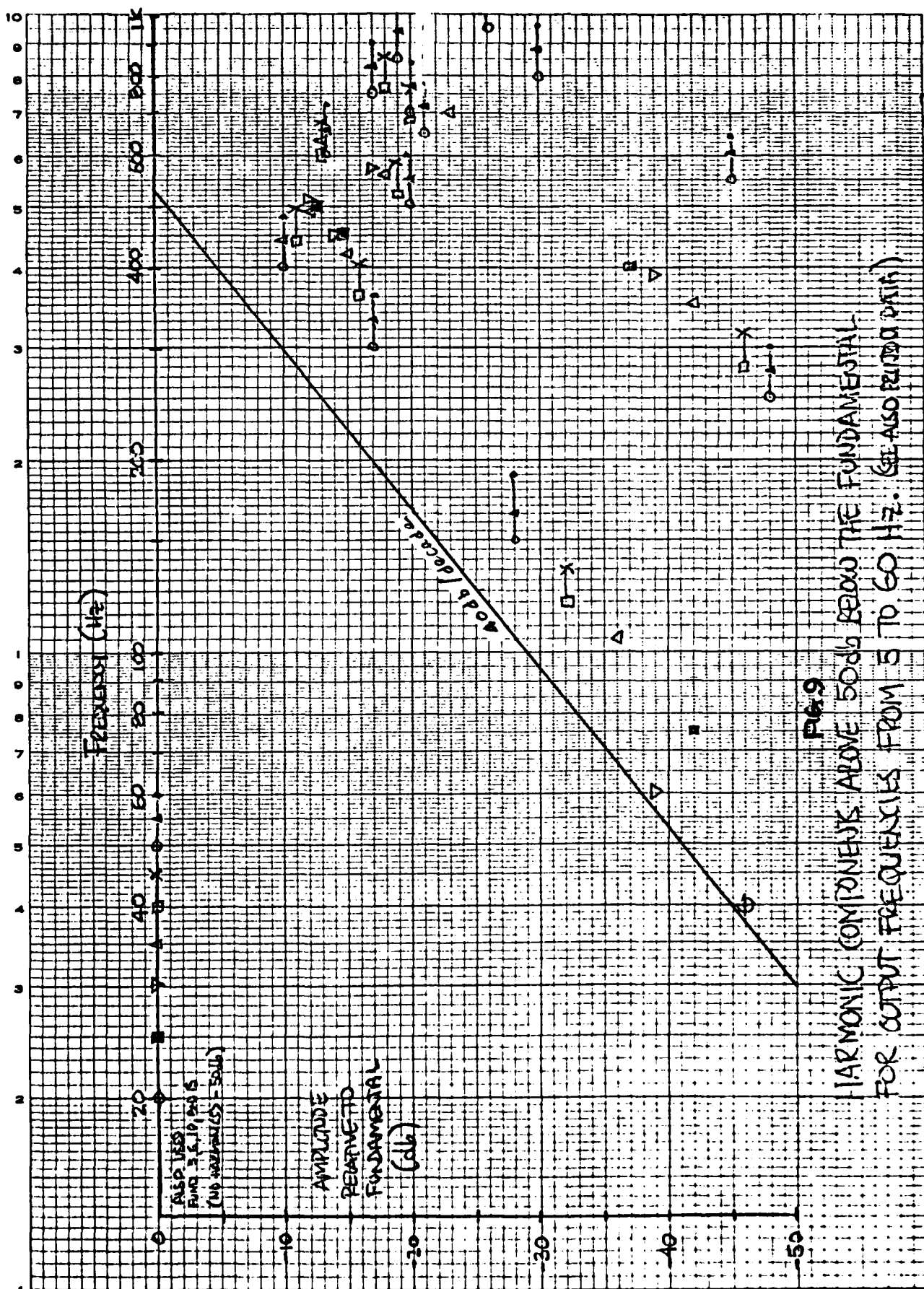


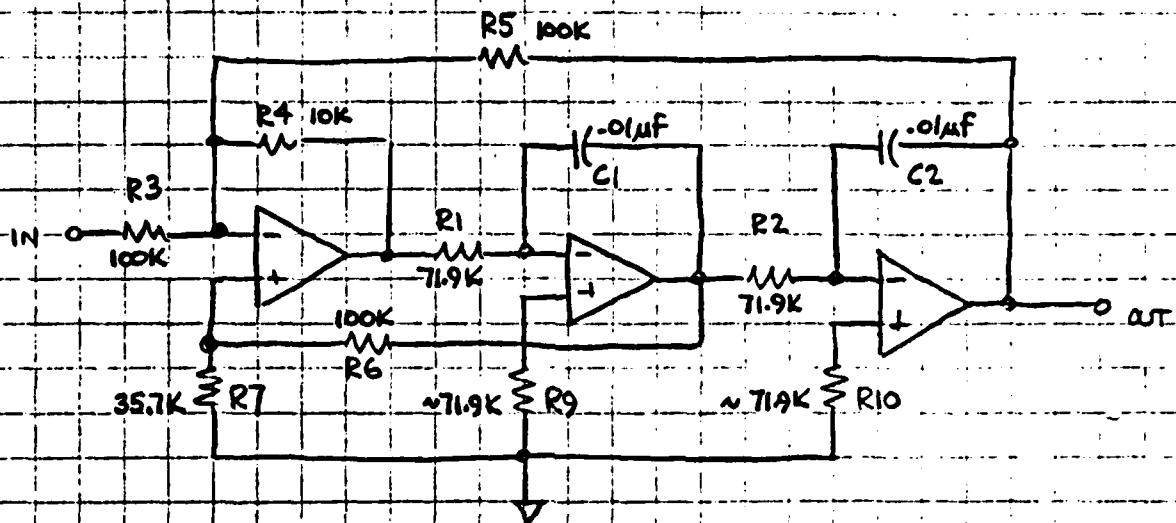
FIG. 9
HARMONIC COMPONENTS ABOVE 500 Hz ABOVE THE FUNDAMENTAL
FOR OUTPUT FREQUENCIES FROM 5 TO 60 Hz. (SEE ALSO FIG. 8)

DBZ
1-12-1978



FIG. 11 2 POLE LPF, CRITICALLY DAMPED, $f_c = 70\text{Hz}$ DESIGN

"UNIVERSAL-SHTE VARIABLE" CONFIGURATION, INVERTING LPF MODE.



① CHOOSE:

$$\begin{aligned} R5/R4 &= 10 \\ R4 &= 10\text{K} \\ R5 &= 100\text{K} \\ R3 &= 100\text{K} \\ R6 &= 100\text{K} \end{aligned}$$

②

$$R1 = R2 = \left(\frac{1}{\omega_0} \right) \sqrt{\frac{R4}{R5 C1 C2}}$$

CHOOSE THEN

$$C1 = C2 = 0.01\mu\text{F}$$

$$R1 = R2 = \left(\frac{1}{2\pi \times 70} \right) \sqrt{\frac{1}{(0.01 \times 10^{-6})^2}} = 71.9\text{K} = R1 = R2$$

③

$$R7 = \frac{100\text{K}\Omega}{3.8Q-1} \quad (\text{FOR UNITY GAIN})$$

FOR CRITICAL DAMPING $Q=1$

SO

$$R7 = \frac{100\text{K}}{2.8} = 35.7\text{K} = R7$$

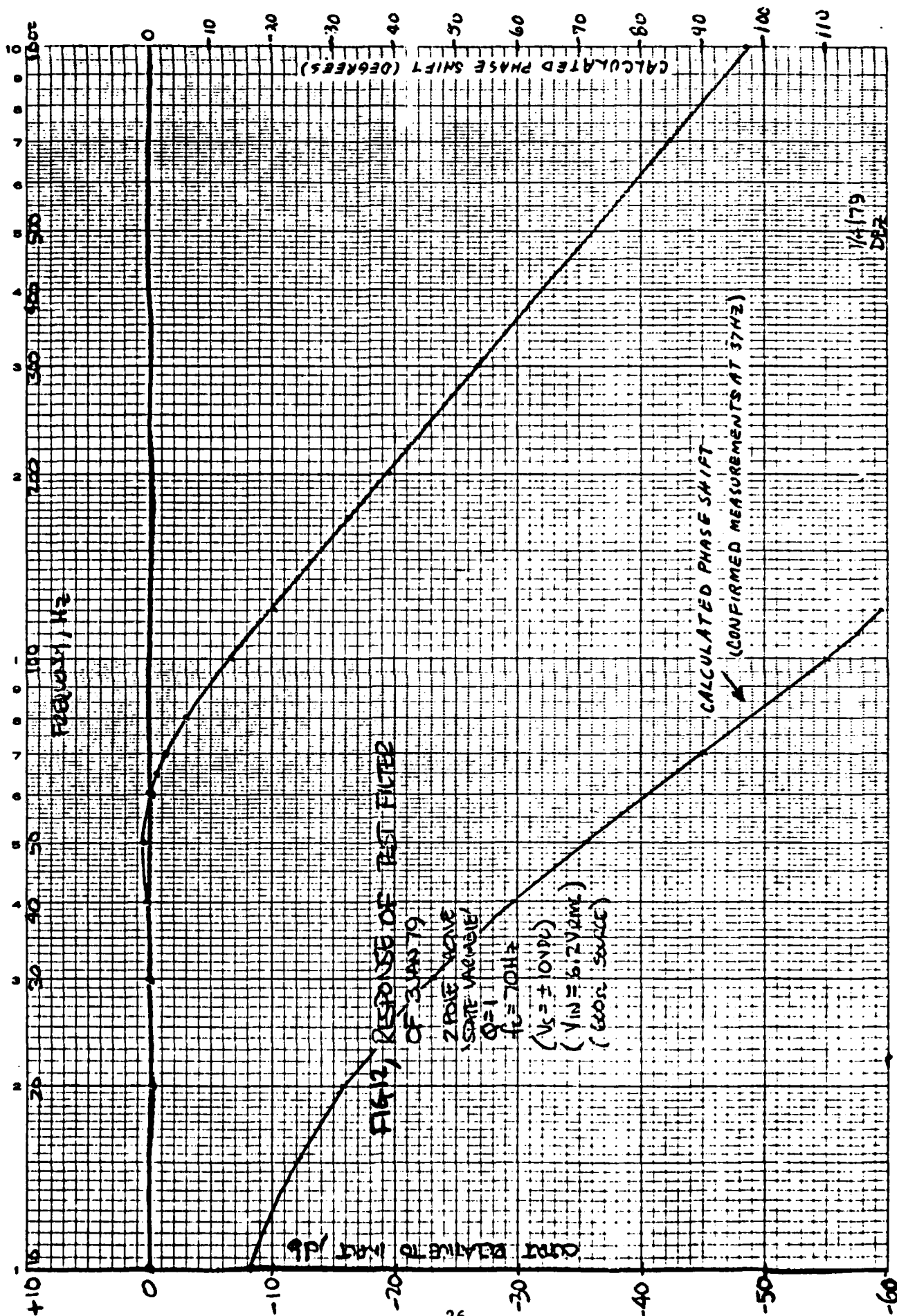
④

$$R_{10} = R9 = R1 = 71.9\text{K}, \text{ USE } 68\text{K}$$

DEZ 1/2/79

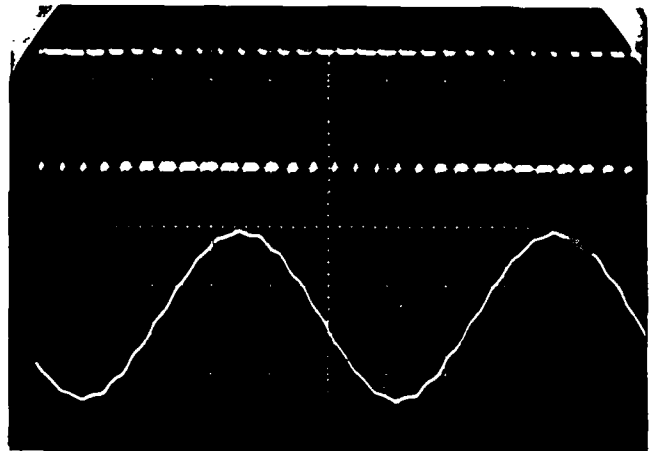
NO. 341-L210 DIETZEN GRAPH PAPER
SEMI-LOGARITHMIC
2 CYCLES X 10 DIVISIONS PER INCH

EUGENE DIETZEN CO.
MADE IN U. S. A.



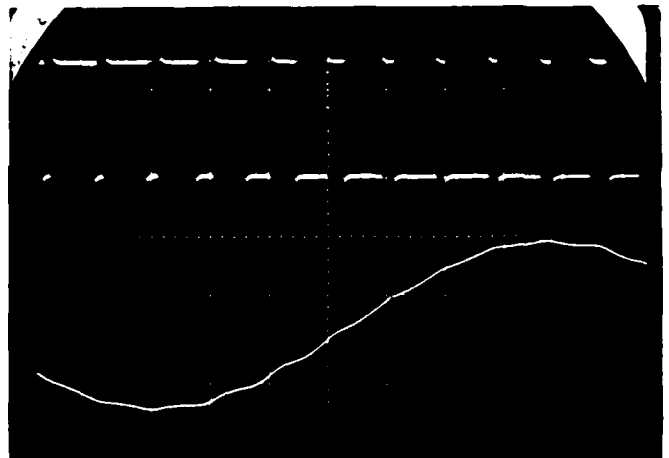
1/4/79
DBZ

- PWM Waveform, 2V/CM →
- 36 Hz Sine Wave Modulation
- 540 Hz Carrier Horiz.
- Scale: 5 ms/CM



- Filtered Output, 1V/CM →
- 2 Pole 70 Hz Filter

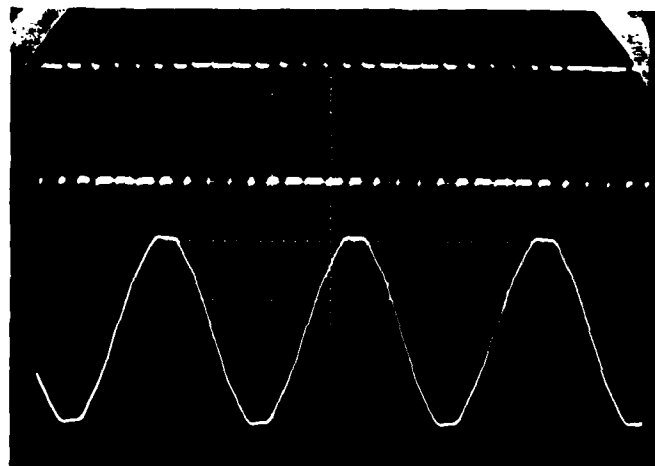
- PWM Waveform, 2V/CM →
- 36 Hz Sinewave Modulation
- 540 Hz Carrier Horiz.
- Scale: 2 ms/CM



- Filtered Output, 1V/CM →
- 2 Pole 70 Hz Filter

Figure 13 PWM WAVEFORMS

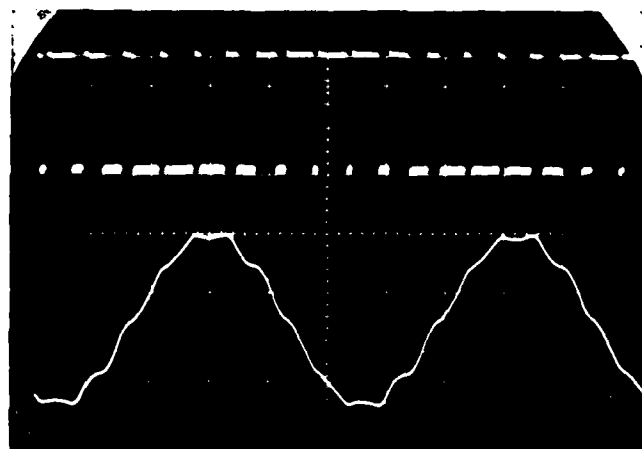
- PWM Waveform, 2V/CM →
- 60 Hz Sine Modulation
- 540 Hz Carrier Horiz.
- Scale: 5 ms/CM



- Filtered Output, 1V/CM →
- 2 Pole 70 Hz Filter

(Flat-Topping due to Incorrect Modulation Pulse Widths)

- PWM Waveform, 2V/CM →
- 37 Hz Sine Wave Modulation
- 333 Hz Carrier Horiz.
- Scale: 5 ms/CM

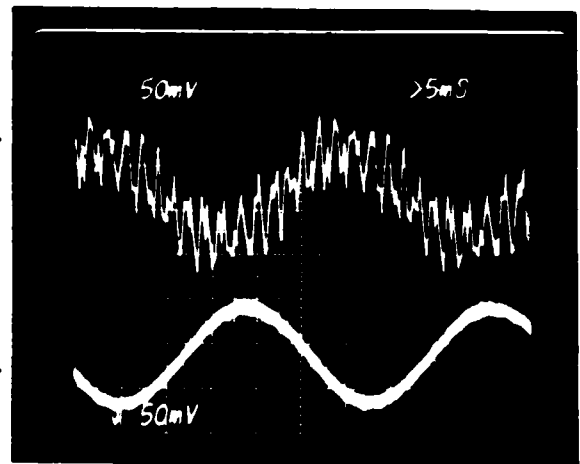


- Filtered Output, 1V/CM →
- 2 Pole 70 Hz Filter

Inadequate number of PWM pulses per modulation cycle (low carrier frequency) causes rough output waveform.

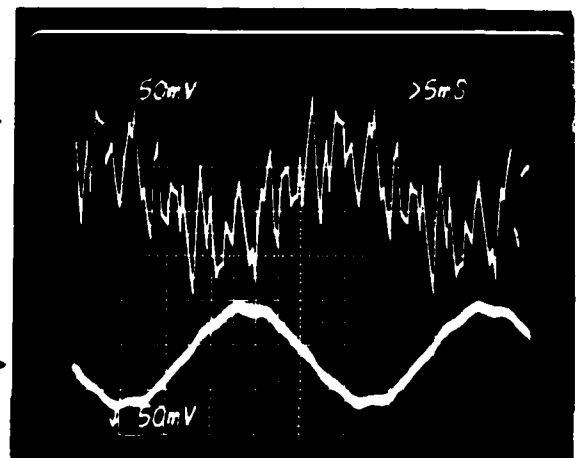
Figure 13 PWM WAVEFORMS
(Continued)

- ΦA Motor Current, 0.5A/CM $\Rightarrow$
36 Hz Sine Wave Modulation
540 Hz Carrier Horiz.
Scale: 5 ms/CM



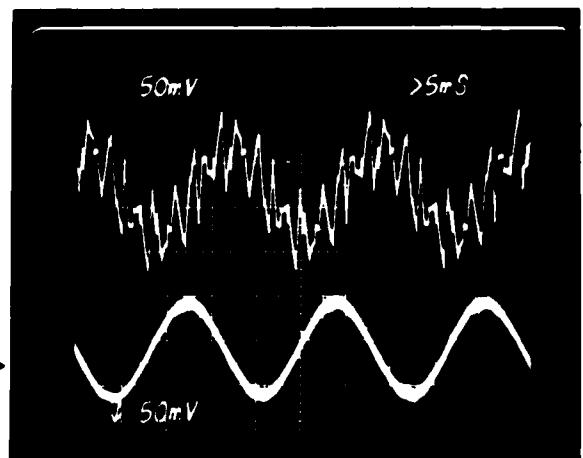
- Filtered Current Sense, 0.5A/CM $\Rightarrow$
2 Pole 70 Hz Filter

- ΦA Motor Current, 0.5A/CM $\Rightarrow$
37 Hz Sine Wave Modulation
333 Hz Carrier Horiz.
Scale: 5 ms/CM



- Filtered Current Sense, 0.5A/CM $\Rightarrow$
2 Pole 70 Hz Filter

- ΦA Motor Current, 0.5A/CM $\Rightarrow$
60 Hz Sine Wave Modulation
540 Hz Carrier Horiz.
Scale: 5 ms/CM



- Filtered Current Sense, 0.5A/CM $\Rightarrow$

Figure 14 MOTOR WAVEFORMS

SYSTEM OPERATION

Figure 15 is a block diagram of the overall system. In the complete system, sensor conditioners operate on the current transformer and voltage probe inputs to provide analog voltages proportional to each of the three phase currents and voltages. I_B and I_C are added differentially and scaled, as are V_A and V_B . The four functions existing after the scaling and addition processes are then filtered. Each filter is identical so as to impose the same delay in each signal path. Thus, the phase relation among the d-q variables resulting at the output of the filters is preserved. Each variable is converted to a digital signal with an 8 BIT A/D converter and is transmitted to the microprocessor through a PIA.

The microprocessor and a motor driver were interfaced and successfully operated a three phase one-third horsepower motor over three pre-programmed speeds and many amplitudes.

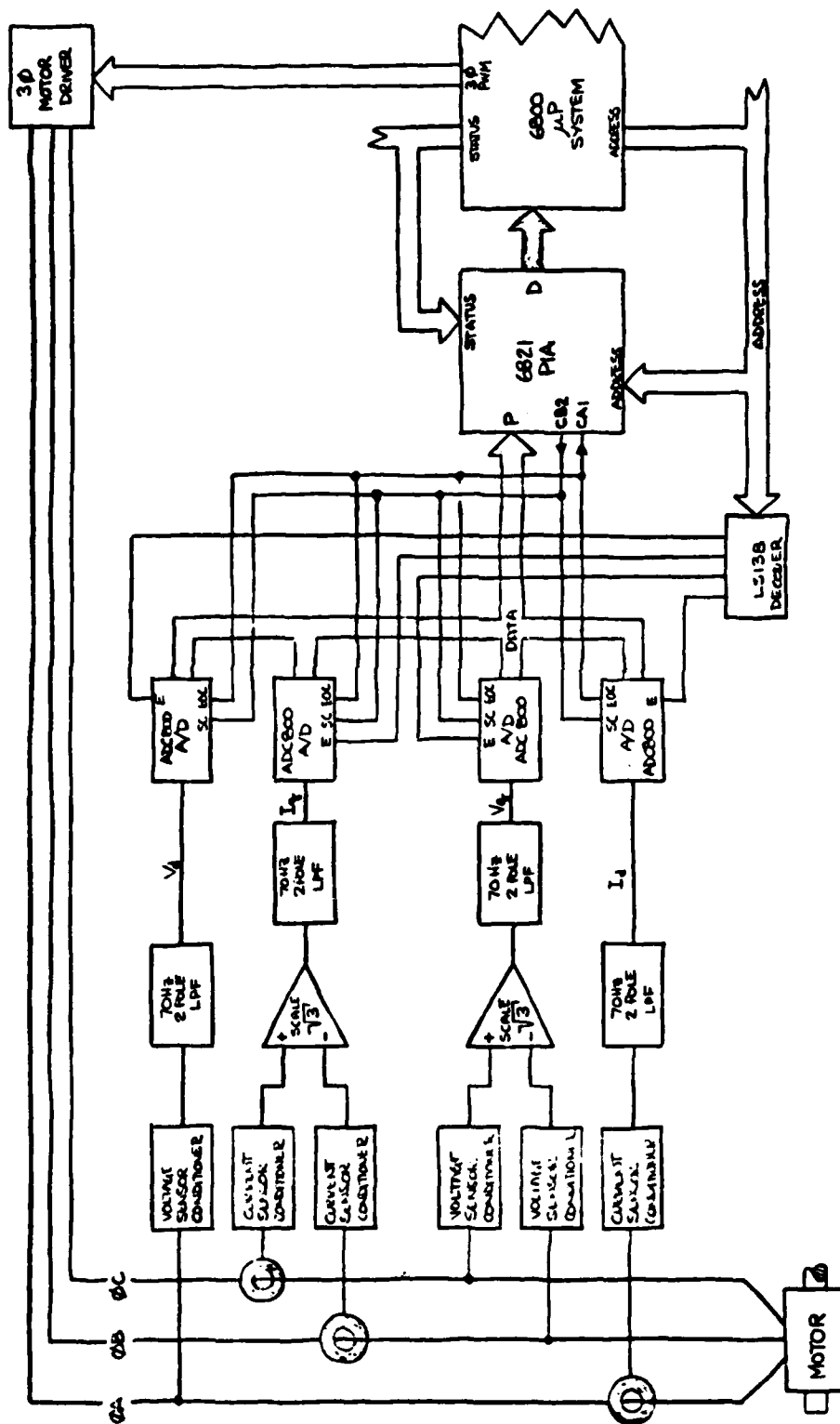


FIGURE 15
 µP CONTROLLED AC MOTOR DRIVE
 OVERALL BLOCK DIAGRAM

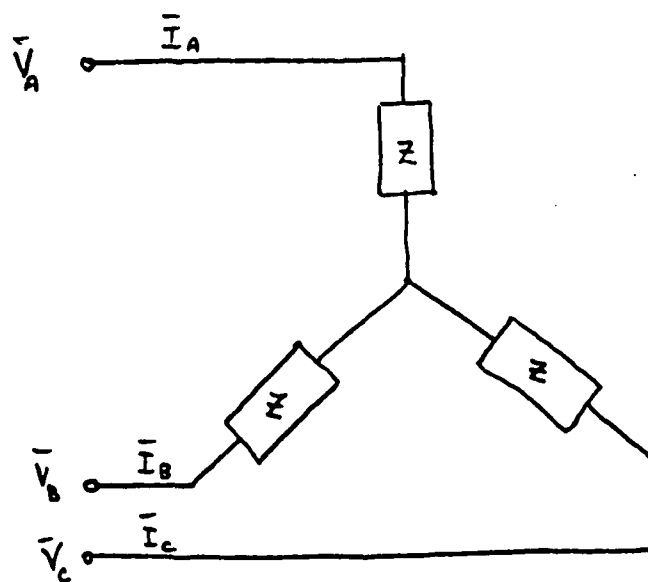
Appendix A

DERIVATION OF AIR GAP FLUX SYNTHESIS SIGNAL

Let applied voltage $\bar{V} = V \sin \omega t$ and then consequent current $\bar{I} = I \sin(\omega t - \phi)$ and the reactive power in one phase

$$W_x = \frac{1}{2} V I \sin \phi$$

For a 3-phase system



$$\bar{V}_A = V \sin \omega t$$

$$\bar{V}_B = V \sin(\omega t - 2\pi/3)$$

$$\bar{V}_C = V \sin(\omega t + 2\pi/3)$$

$$\bar{I}_A = I \sin(\omega t - \phi)$$

$$\bar{I}_B = I \sin(\omega t - 2\pi/3 - \phi)$$

$$\bar{I}_C = I \sin(\omega t + 2\pi/3 - \phi)$$

$$Z = \frac{V}{I} (\cos \phi + j \sin \phi)$$

FIG. 1a

The three phase system can be expressed in terms of d - q (direct and quadrature) variables.

Three phase vector diagram:

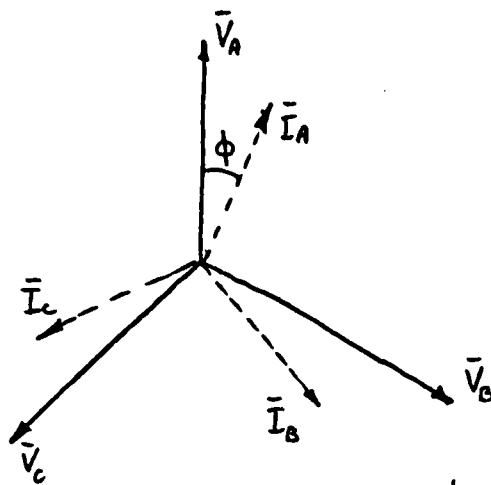
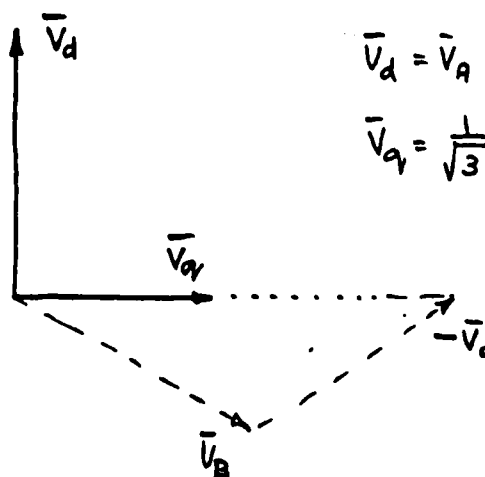


FIG. 1b

d - q equivalent vector diagram:

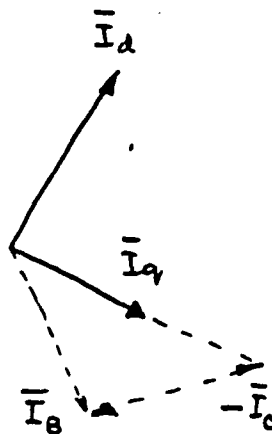


$$\bar{V}_d = \bar{V}_A = V \sin \omega t$$

$$\bar{V}_q = \frac{1}{\sqrt{3}} (\bar{V}_B - \bar{V}_C) = -V \cos \omega t$$

FIG. 1c

FIG. 1d



$$\bar{I}_d = \bar{I}_a = I \sin(\omega t - \phi)$$

$$\bar{I}_q = \frac{1}{\sqrt{3}} (\bar{I}_a - \bar{I}_c) = -I \cos(\omega t - \phi)$$

Products can be formed for $\bar{V}_d \bar{I}_q$ and $\bar{V}_q \bar{I}_d$ as follows:

$$\begin{aligned} \bar{V}_d \bar{I}_q &= -VI \sin \omega t \cdot \cos(\omega t - \phi) \\ &= -\frac{VI}{2} \sin \phi - \frac{VI}{2} \sin(2\omega t - \phi) \end{aligned}$$

$$\begin{aligned} \bar{V}_q \bar{I}_d &= -VI \cos \omega t \cdot \sin(\omega t - \phi) \\ &= \frac{VI}{2} \sin \phi - \frac{VI}{2} \sin(2\omega t - \phi) \end{aligned}$$

Each is composed of a DC term and an AC term at twice the frequency of the applied voltage. Taking the difference of the two terms doubles the DC component and cancels the AC components

$$\bar{V}_q \bar{I}_d - \bar{V}_d \bar{I}_q = VI \sin \phi$$

which can be seen to be twice the reactive power, W_x , in one phase. Thus, the first objective of the flux synthesis technique is to generate the d-q variables from the three phase currents and voltages. This is shown in the block diagram as follows:

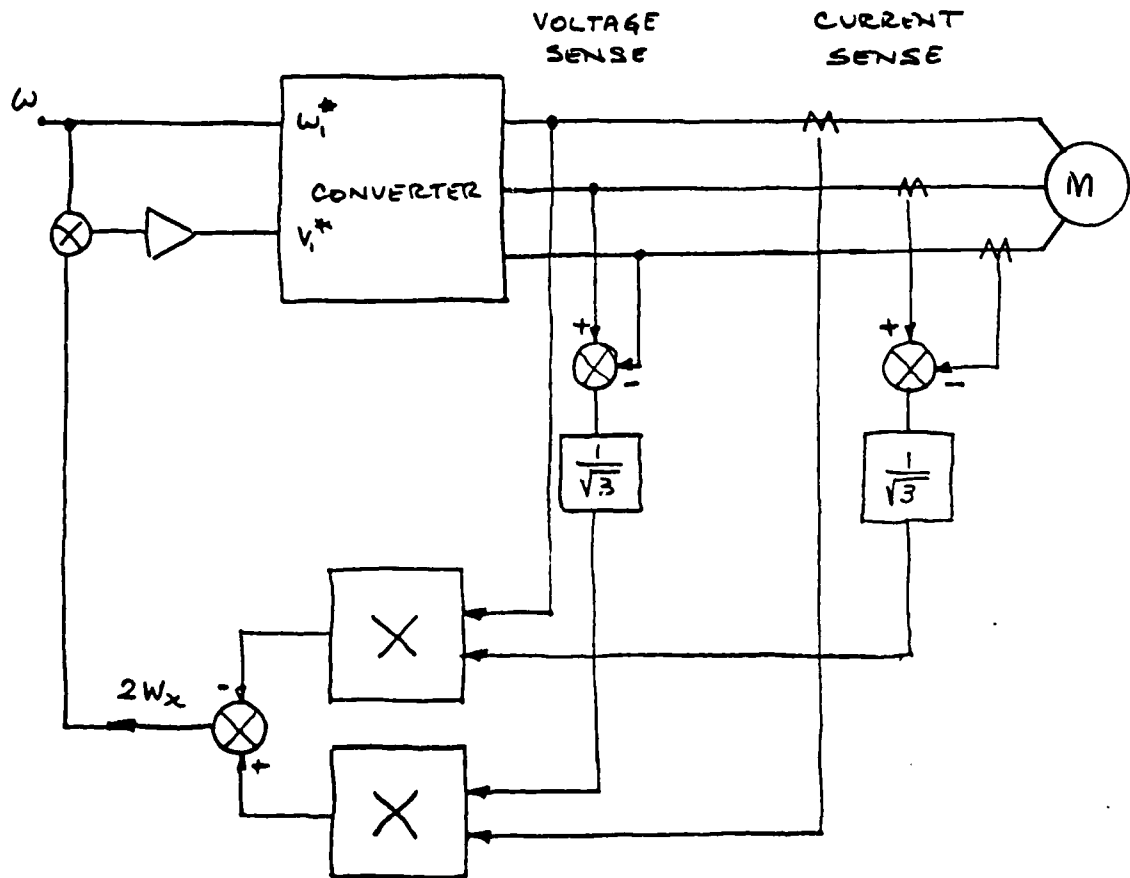


FIG. 2a

Thus it is possible to generate a signal proportional to the reactive power, W_x , by measuring only the voltages and currents at the motor terminals.

Since the real objective is to maintain the component of W_x due to L_m flowing in L_m , W_x must be broken down into its component terms.

Referring to the equivalent circuit (Figure 2b)

$$W_x = \frac{1}{2} \bar{I}_1^2 \omega L_1 + \frac{1}{2} \bar{I}_m^2 \omega L_m + \frac{1}{2} \bar{I}_2^2 \omega L_2$$

The first term is a function of the measured current, a fixed frequency and fixed inductance and is known. The second term is the desired component of W_x and the third term is a term that is undefined unless $\bar{I}_2$ can be accounted for.

Referring to the vector diagram of Figure 2c

$$\begin{aligned} \bar{I}_1^2 &= (I_m + I_{2x})^2 + I_{2y}^2 \\ &= I_m^2 + 2 I_m I_{2x} + I_2^2 \end{aligned}$$

but

$$\bar{I}_m = \frac{E_1}{\omega, L_m}$$

$$\bar{I}_{2x} = \bar{I}_2^2 \frac{\omega, L_2}{E_1}$$

$$\bar{I}_1^2 = I_m^2 + 2\bar{I}_2^2 \frac{L_2}{L_m} + \bar{I}_2^2$$

$$\bar{I}_1^2 - I_m^2 = \frac{L_m + 2L_2}{L_m} \cdot \bar{I}_2^2$$

which allows substitution for $\bar{I}_2^2$ in terms of $\bar{I}_1^2 - I_m^2$.

$$W_x = \frac{1}{2} \omega, L_1 \bar{I}_1^2 + \frac{1}{2} \bar{I}_m^2 \omega, L_m + \frac{1}{2} \frac{L_m}{L_m + 2L_2} \omega, L_2 (\bar{I}_1^2 - I_m^2)$$

$$2W_x = \omega, L_1 \bar{I}_1^2 + \omega, L_m \frac{L_m + L_2}{L_m + 2L_2} \cdot I_m^2 + \omega, \frac{L_m L_2}{L_m + 2L_2} I_1^2$$

$$W_x = \frac{\omega_1}{2(L_m + 2L_2)} \left[(L_m^2 + L_m L_2) I_m^2 + (L_m L_2 + L_m L_1 + 2L_1 L_2) I_1^2 \right]$$

i.e., W_x is a function of the derived I_m and known terms L_m , L_2 , L_1 and W_1 .
Thus, by operating on W_x appropriately I_m and hence the air gap flux can be determined.

Rearranging the previous equations we have:

$$\begin{aligned} \frac{L_m (L_m + L_2)}{2 (L_m + 2L_2)} \omega_1 I_m^2 &= W_x - \frac{1}{2} \omega_1 I_1^2 \left(L_1 + \frac{L_m L_2}{L_m + L_2} \right) \\ &= W_x - \frac{1}{2} I_1^2 \cdot \omega_1 \left(L_1 + \frac{L_m L_2}{L_m + L_2} \right) \end{aligned}$$

i.e.

$$K, \omega, I_m^2 = W_x - \frac{1}{2} \bar{I}_1^2 \cdot \omega, L'$$

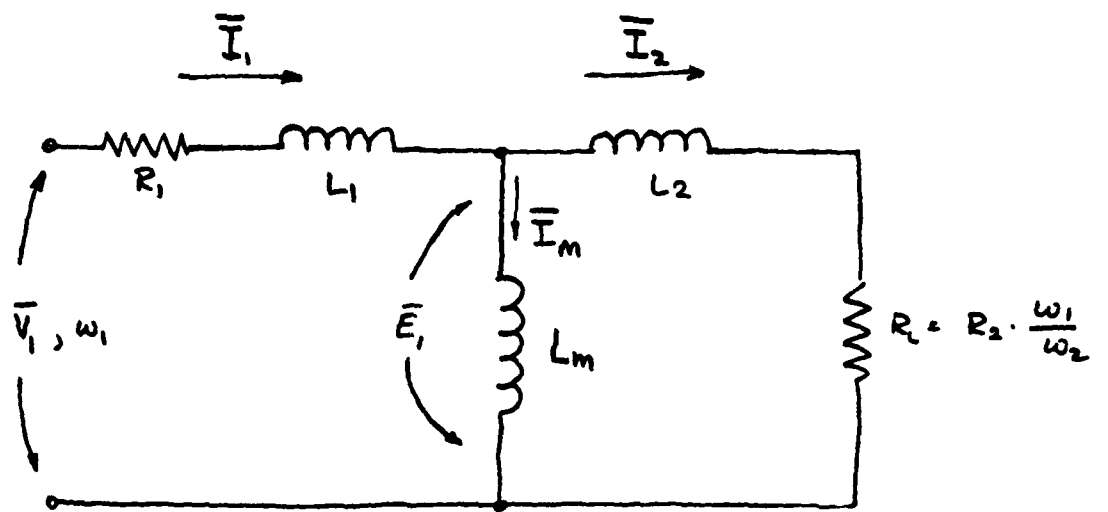


FIG. 2 b

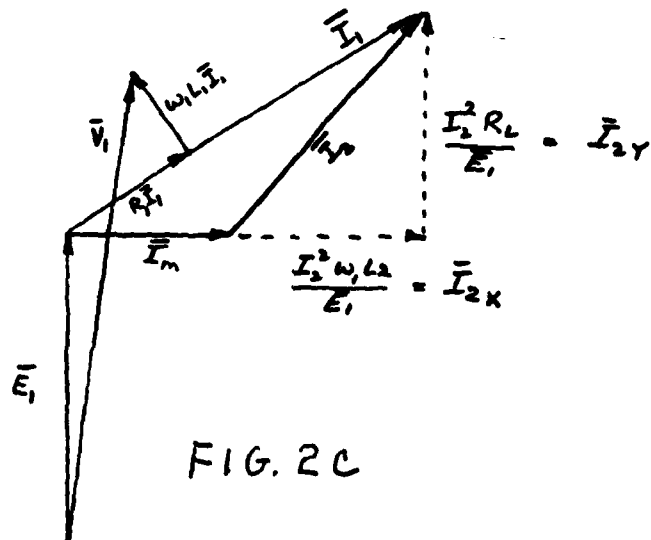
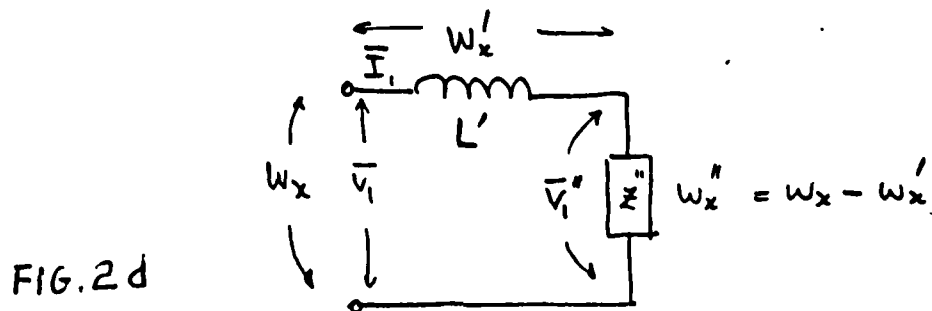


FIG. 2 c

from which a revised block diagram can be drawn to allow for the required reduction in W_x by a term proportional to the reactive power that would be developed in an inductor $L_1 + L_m L_2 / (L_m + L_2)$ if the stator current were to flow through it. To compute $W_x - \frac{1}{2} I_1^2 \omega L'$ we can reduce the voltage components of W_x by an amount equivalent to the voltage drop across L' due to $\bar{I}_1$ flowing. A simplified equivalent circuit is as follows:



The feedback function is to keep $\bar{I}_m$ constant for any value set for W . It does this by modifying the convertor output voltage which in turn causes a change in $\bar{I}_m$. Since this is a closed loop feedback involving a load with transfer characteristic $\frac{1}{s + T_i}$ where T_i reflects the $\frac{L}{R}$ time constant of the induction motor the feedback function will need compensation of the form $\frac{s + T_i}{s}$ i.e. an integrating term to bring down the overall gain as frequency increases and a lead term to compensate for the lag introduced by the inductive time constant of the load. This steers the phase angle away from -180 until the gain has become less than unity.

Figure 3 shows a block diagram of a flux regulator loop following the model of Figure 2 i.e. without allowing for the drop in L_1 . The scaled inputs V_q, I_d and V_d, I_q are multiplied together and then subtracted to produce $\bar{V} \bar{I} \sin \phi$ (W_x) which is the control signal CTR.

This is kept within bounds by a clamping circuit and is compared with a reference signal. The consequent error signal, again held within limits is fed to a loop which provides the $\frac{s+2}{s}$ function. The output is again held within limits and is used to modify the voltage control level V^* which would normally be kept constant for any given input frequency W^* . A microprocessor program for this loop has been assembled and run and is attached.

To account for the voltage drop in L_1 and use the model of Figure 4, a term $(j\omega \frac{k_v}{k_1} L')$ must be subtracted from each voltage input. This adds an extra subroutine in the program to multiply $k_1 \bar{I}_q$ and $k_1 \bar{I}_d$ by the factor L' and ω_1 and subtract them from V_d and V_q .

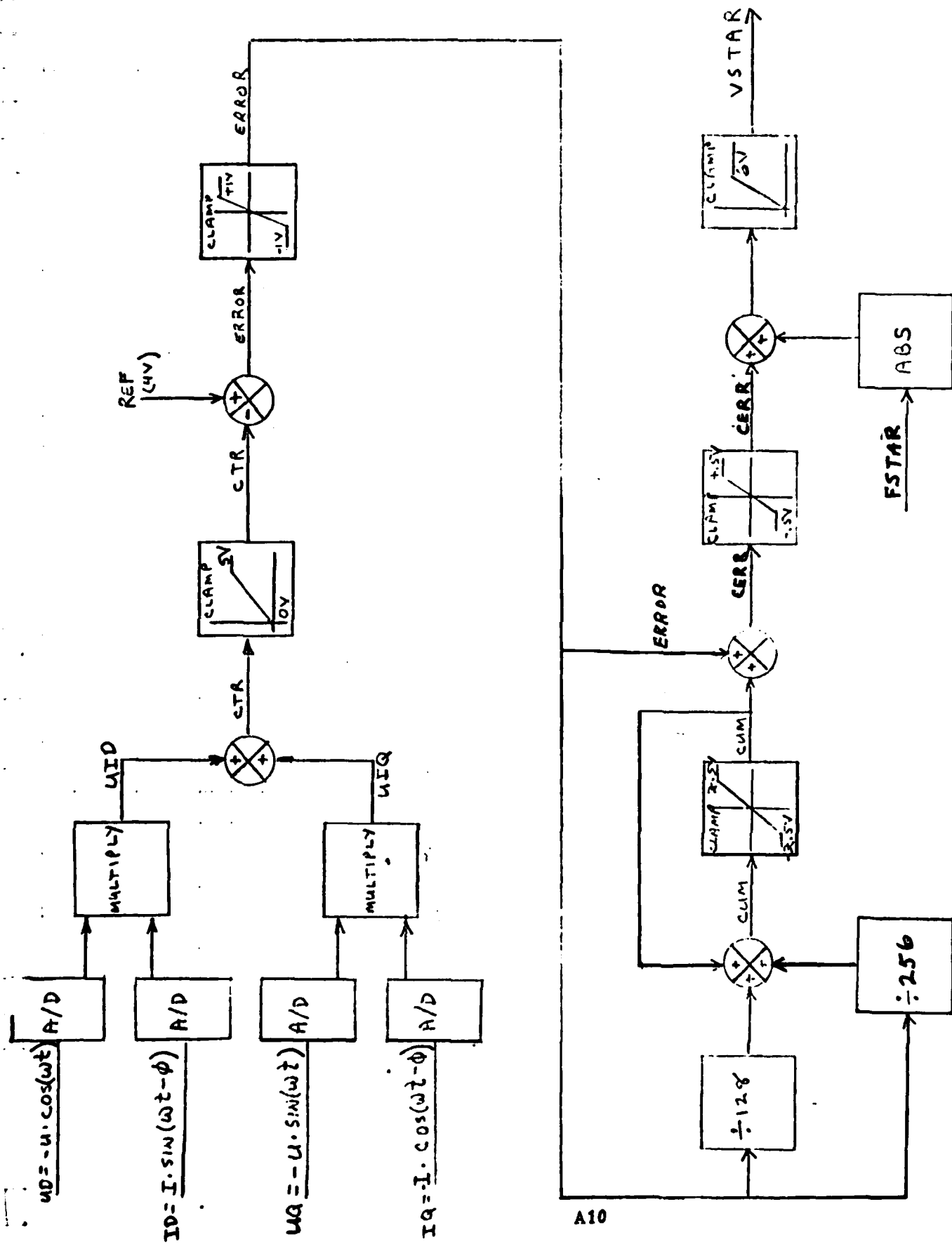


FIG. 3 BLOCK DIAGRAM-FLUX REGULATOR LOOP

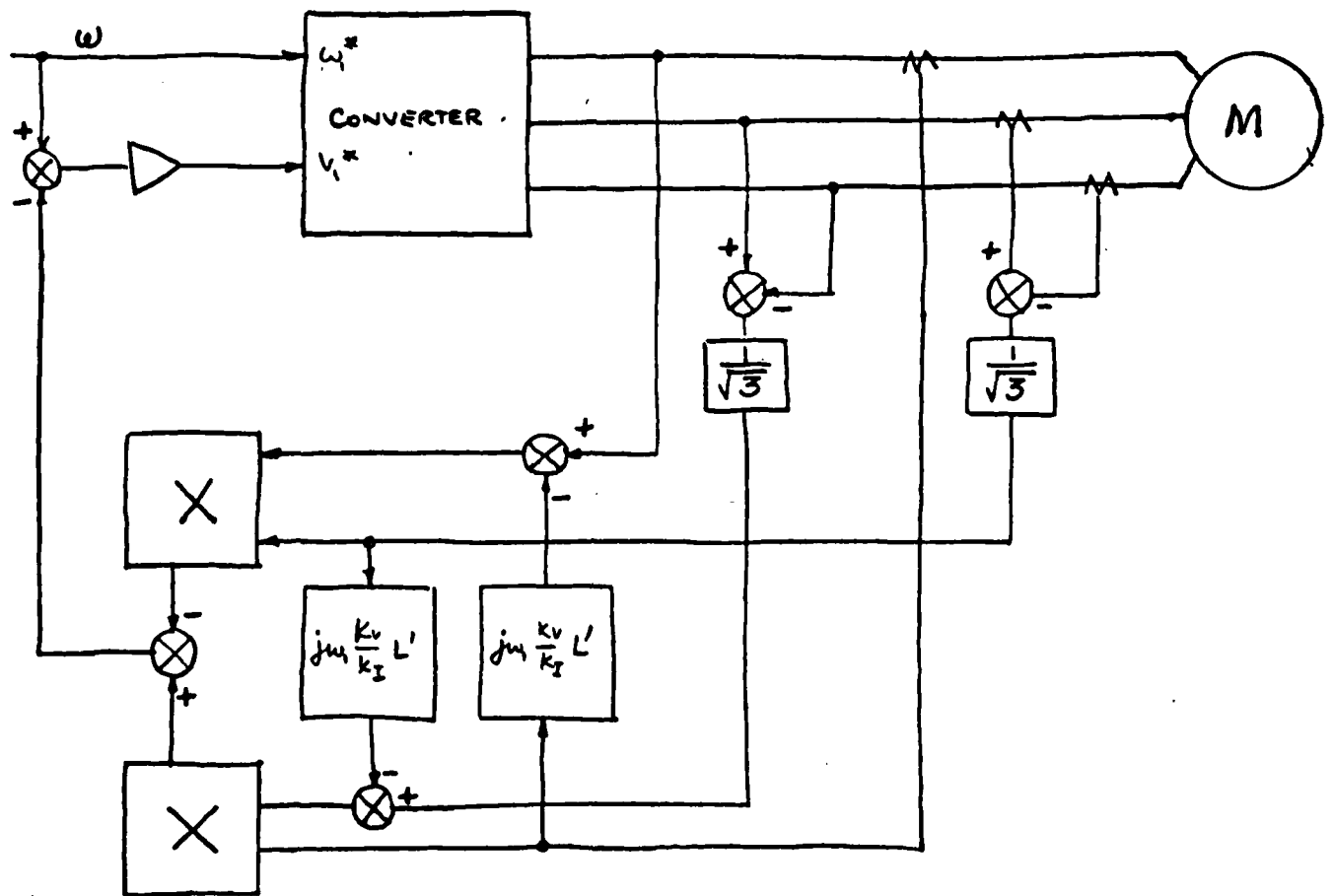


FIG. 4

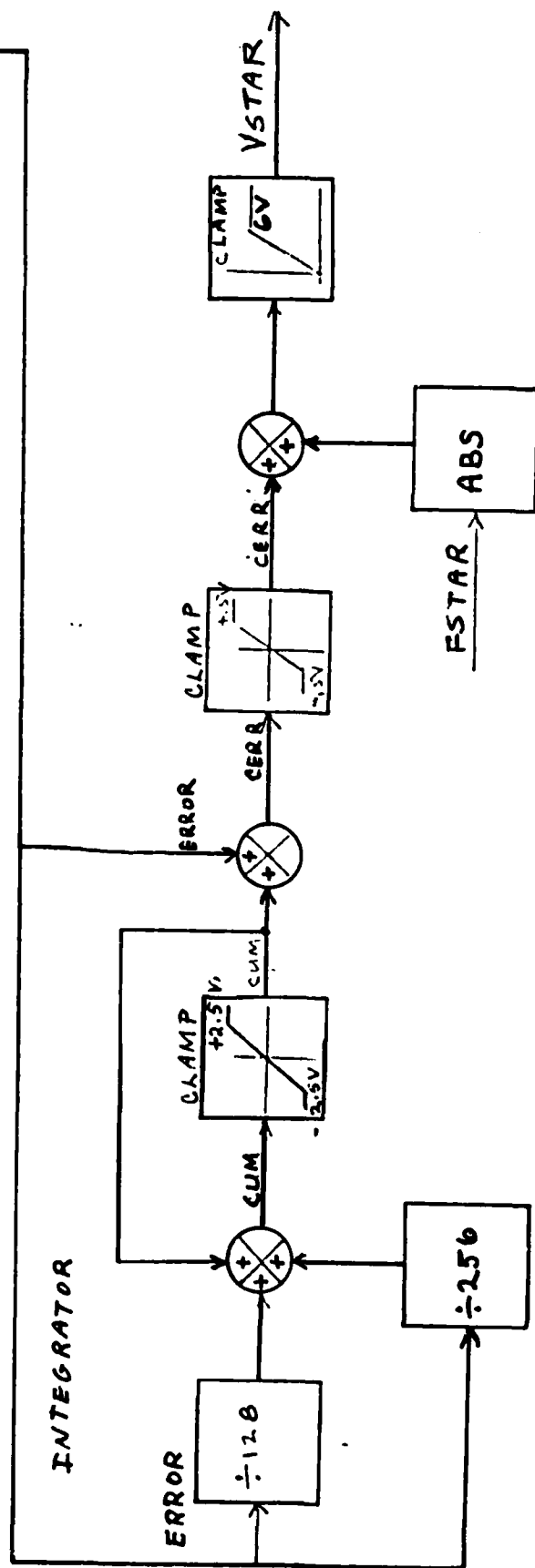
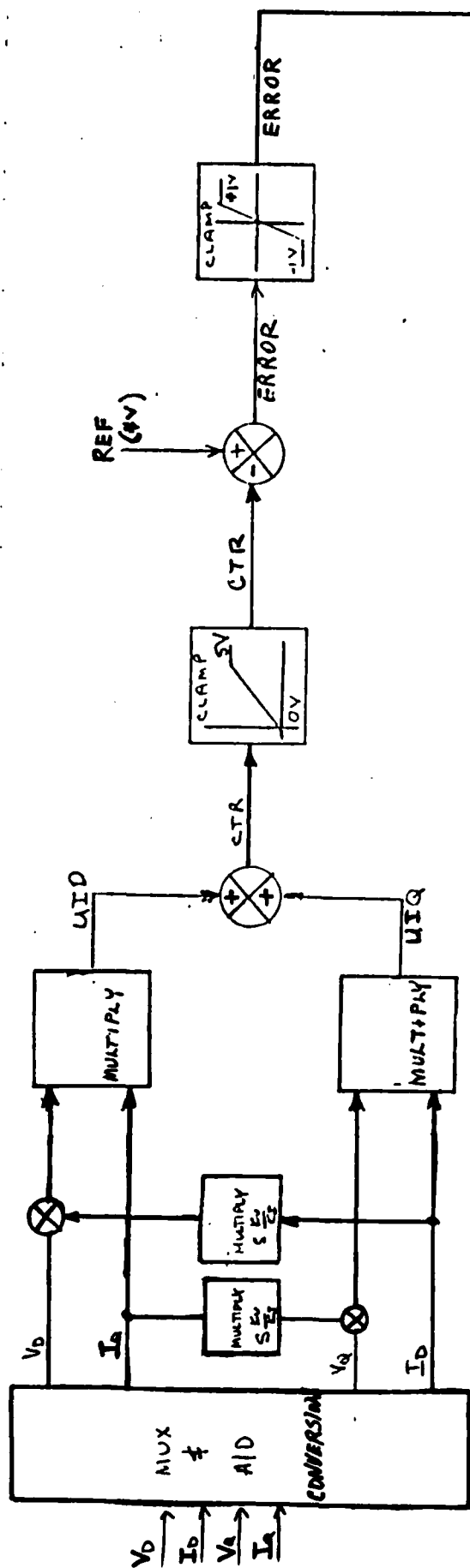


FIG.5 BLOCK DIAGRAM- FLUX REGULATOR LOOP

Implementing the Integrator

The accumulated error signal (CUM) is defined as:

$$CUM = \int_0^t 2 \times ERROR \, dt$$

This can be computed as:

$$CUM = CUM + 2 \times ERROR \times \Delta T$$

Assuming a flux loop cycle time of .005 seconds the factor $2 \times \Delta T = .01$.

In order to take advantage of the binary nature of a computer, where multiplication and division by powers of 2 are merely shift operations, it is convenient to estimate: $2 \times \Delta T = .01 \approx \left(\frac{1}{128} + \frac{1}{256} \right)$

The integration equations then become:

$$CUM = CUM + \frac{ERROR}{128} + \frac{ERROR}{256}, \text{ and the ERROR}$$

signal is shifted and added as required to compute the error increment for the time interval.

```

00001                                NAM      FLUX
00002                                *
00003                                * FLUX LOOP PROGRAM
00004                                *
00005                                OPT      0
00006 0100                        ORG      $100
00007                                *
00008                                *
00009                                * BLOCK 2. MULTIPLICATION
00010                                *      UD*ID: UD*ID
00011                                *
00012 0100 B6 0400 START LDA A UD
00013 0103 F6 0401      LDA B ID
00014 0106 B0 0172      JSR SMPY      UID TO A,B
00015 0109 B7 0404      STA A UID      TEMP SAVE UID
00016 010C F7 0405      STA B UID+1
00017                                *
00018 010F B6 0402      LDA A UD
00019 0112 F6 0403      LDA B ID
00020 0115 B0 0172      JSR SMPY      UID TO A,B
00021                                *
00022                                * BLOCK 3. CTR=UID+UID
00023                                *
00024 0118 FB 0405      ADD B UID+1      ADD LSB'S
00025 011B B9 0404      ADC A UID      ADD MSB'S + CARRY
00026                                *
00027                                *
00028                                * BLOCK 4. ERROR=REF-CTR
00029                                *      ERROR=CLAMP(ERROR)
00030                                * (CTR IS IN ACC A)
00031                                *
00032 011E 40      NEG A -CTR
00033 011F 8B 66      ADD A #166      REF-CTR (REF=66HEX=4VOLTS)
00034 0121 06 1A      LDA B #1A      CLAMP FOR ERROR (=1 VOLT)
00035 0123 B0 018F      JSR CLAMP      CLAMP ERROR
00036 0126 B7 0406      STA A ERROR      SAVE CLAMPED ERROR SIGNAL
00037                                *
00038                                *
00039                                * BLOCK 5. CUM=CUM+ERROR/128+ERROR/256
00040                                *      CUM=CLAMP(CUM) (CLAMP=2.5 V)
00041                                * (ERROR IS IN ACC A)
00042                                *
00043 0129 16      TAB      ERROR/256
00044 012A 47      ASR A EXTEND      SIGN IN ACC A
00045 012B 47      ASR A
00046 012C 47      ASR A
00047 012D 47      ASR A
00048 012E 47      ASR A
00049 012F 47      ASR A
00050 0130 47      ASR A
00051 0131 B7 0407      STA A ER256      TEMP SAVE ERROR/256 (16 BIT
00052 0134 F7 0408      STA B ER256+1
00053                                *
00054 0137 58      ASL B ERROR/128
00055 0138 49      ROL A
00056                                *
00057 0139 FB 0408      ADD B ER256+1      ERROR/128+ERROR/256
00058 013C B9 0407      ADC A ER256      (16 BIT ADD)

```



```

00059 013F F8 040A      ADD B   CUM+1      + CUM
00060 0142 B9 0409      ADD A   CUM
00061                      *
00062 0145 F7 040A      STA B   CUM+1      SAVE CUM (LSB'S)
00063 0148 C6 40        LDA B   #F40      CLAMP FOR CUM (=2.5 V)
00064 014A BD 018F      JSR      CLAMP      CLAMP CUM
00065 014D B7 0409      STA A   CUM      SAVE CUM (16 BITS)
00066                      *
00067                      *
00068                      * BLOCK 6.  CERR=ERROR+CUM
00069                      *          CERR=CERR/2  (RESCALE CERR)
00070                      * (MOST SIGNIFICANT 8 BITS OF CUM IN ACC A)
00071                      *
00072 0150 BB 0406      ADD A   ERROR      ERROR+CUM=CERR
00073 0153 47          ASR A   CERR/2
00074                      *
00075                      *
00076                      * BLOCK 7.  CERR=CLAMP(CERR)
00077                      *
00078 0154 C6 07        LDA B   #7          CLAMP FOR CERR (= 5 VOLTS)
00079 0156 BD 018F      JSR      CLAMP
00080 0159 B7 0406      STA A   CERR      SAVE CERR (8 BITS)
00081                      *
00082                      *
00083                      * BLOCK 8.  READ FSTAR  FSTAR=ABS(FSTAR)
00084                      *
00085 015C F6 040C      LDA B   FSTAR      FREQ REF
00086 015F 2A 01        BPL      ++3
00087 0161 50          NEG B   ABS          FSTAR
00088                      *
00089                      *
00090                      * BLOCK 9.  VSTAR=CERR+FSTAR
00091                      *          VSTAR=CLAMP(VSTAR)
00092                      * (CERR IS IN ACC A, FSTAR IS IN ACC B)
00093                      *
00094 0162 1B          ABA          CERR+FSTAR
00095 0163 2A 01        BPL      ++3
00096 0165 4F          CLR A   IF          VSTAR NEG. VSTAR=0
00097 0166 C6 40        LDA B   #F40      MAX VSTAR (=6V)
00098 0168 11          CBA          CLAMP VSTAR TO +6V
00099 0169 2B 01        BMI      ++3
00100 016B 17          TBA
00101 016C B7 040D      STA A   VSTAR      SAVE VSTAR
00102 016F 7E 0100      JMP      START     REPEAT PROGRAM

```

```

00104      *
00105      * SMPY--SIGNED MULTIPLY ROUTINE
00106      *
00107      *
00108      * CALLING SEQUENCE: JSR     SMPY
00109      * INPUTS:  ACCA=MULTIPPLICAND
00110      *          ACCB=MULTIPLIER
00111      * OUTPUTS: 16 BIT PRODUCT IN ACCA,ACCB
00112      *
00113 0172 B7 040E SMPY  STA A  MCAND  SAVE MULTIPPLICAND
00114 0175 4F      CLR A  INIT    RESULT
00115 0176 CE 0008      LD%    #8    8 BITS
00116      *
00117 0179 C5 01  SMPY1 BIT B  #1    TEST LSB OF MULTIPLIER
00118 017B 27 05      BEQ    SMPY2  IF ZERO, NO ADD
00119 017D BB 040E      ADD A  MCAND  ADD MULTIPPLICAND TO PARTIAL
00120 0180 29 02      BVS    SMPY3  TEST FOR ARITH OVERFLOW
00121 0182 47      SMPY2 ASR A  EXTEND SIGN
00122 0183 49      ROL A
00123 0184 46      SMPY3 ROR A  POSITION PARTIAL RESULT
00124 0185 56      ROR B
00125 0186 09      DEX      COUNT BITS
00126 0187 26 F0      BNE    SMPY1  GO BACK IF MORE BITS
00127      *
00128 0189 24 03      BCC    SMPY4  TEST SIGN OF MULTIPLIER
00129 018B B0 040E      SUB A  MCAND  IF NEG. ADJUST RESULT
00130 018E 39      SMPY4 RTS      RETURN

```

```

00132 *
00133 * CLAMP--SUBROUTINE TO CLAMP ACC A TO +/- ACC B
00134 *
00135 * INPUTS:  ACCA=VALUE TO BE CLAMPED
00136 *          ACCB=CLAMP VALUE (MUST BE POSITIVE)
00137 *
00138 * OUTPUTS: IF (ACCA > ACCB) THEN ACCA=ACCB
00139 *          IF (ACCA < -ACCB) THEN ACCA=-ACCB
00140 *          ELSE ACCA IS UNCHANGED
00141 *
00142 018F 8A 00 CLAMP  ORA A  #0      TEST SIGN OF A
00143 0191 2B 05 BMI     CLMP1   BRANCH IF NEG
00144 *
00145 0193 11      CBA          COMPARE ACCUMULATORS
00146 0194 2B 01 BMI     ++3     IF MINUS A IS OK
00147 0196 17      TBA          ELSE SET ACCA=ACCB
00148 0197 39      RTS          RETURN
00149 *
00150 0198 50 CLMP1  NEG B  NEGATE  CLAMP
00151 0199 11      CBA          COMPARE ACCUMULATORS
00152 019A 2A 01 BPL     ++3     IF PLUS A IS OK
00153 019C 17      TBA          ELSE SET ACCA=-ACCB
00154 019D 50      NEG B  RESTORE ACCB
00155 019E 39      RTS          RETURN

```

```

00157      *
00158      * DEFINE DATA LOCATIONS
00159      *
00160 0400      ORG      $100
00161 0400 00      UD      FCB      0      INPUTS (8 BIT VARIABLES)
00162 0401 00      ID      FCB      0
00163 0402 00      UQ      FCB      0
00164 0403 00      IQ      FCB      0
00165 0404 0000    UID      FDB      0      UD*ID (16 BITS)
00166 0406 00      ERROR    FCB      0      ERROR SIGNAL
00167 0407 0000    ER256    FDB      0      ERROR/256 (16 BITS)
00168 0409 0000    CUM      FDB      0      CUMULATIVE ERROR (16 BITS)
00169 0408 00      CERP      FCB      0      COMPENSATED ERROR
00170 040C 00      FSTAR    FCB      0      FREQ REF
00171 040D 00      VSTAR    FCB      0      VOLTAGE REQUEST
00172 040E 00      MCAND    FCB      0      MULTIPLICAND STORAGE
00173      *
00174      END

```

TOTAL ERRORS 00000

APPENDIX B

BASIC ASPECTS OF THE MODULATION MECHANISM

The type of pulse width modulation studied applies to inverter systems made up of three thyristor poles as shown on Figure 1, forming a "3-phase bridge". The poles are connected across a dc link E_d , consisting of a dc source of fixed magnitude. It is practical to define the pole output voltages V_{AO} , V_{BO} and V_{CO} , relative to the mid-point voltage 0 of the dc source, here termed dc neutral. Therefore, the dc source is represented by two batteries of $E_d/2$ volts each, although in an actual system the dc neutral may not be physically accessible. Each transistor has a free-wheeling diode in antiparallel connection and the firing sequence is such that as one transistor in a given pole is turned on, the reciprocal one in the same pole goes off. Therefore, the transistor system of Figure 1 can be represented by a bi-directional, double throw switch system. The switches are periodically and independently activated from one position to the other, connecting alternately the positive and negative terminal of the dc source to the load buses. The object of the modulation strategy is to perform the switching with such timing and sequence that at the terminals of the load a 3-phase ac voltage appears whose fundamental component has the desired amplitude and frequency, and whose other components have a minimum of unwanted side effects. In addition to the pole-to-neutral instantaneous voltage V_{AO} , V_{BO} and V_{CO} ("pole voltages"), the instantaneous pole-to-pole voltages V_{AB} , V_{BC} and V_{CA} ("line voltages") can also be considered.

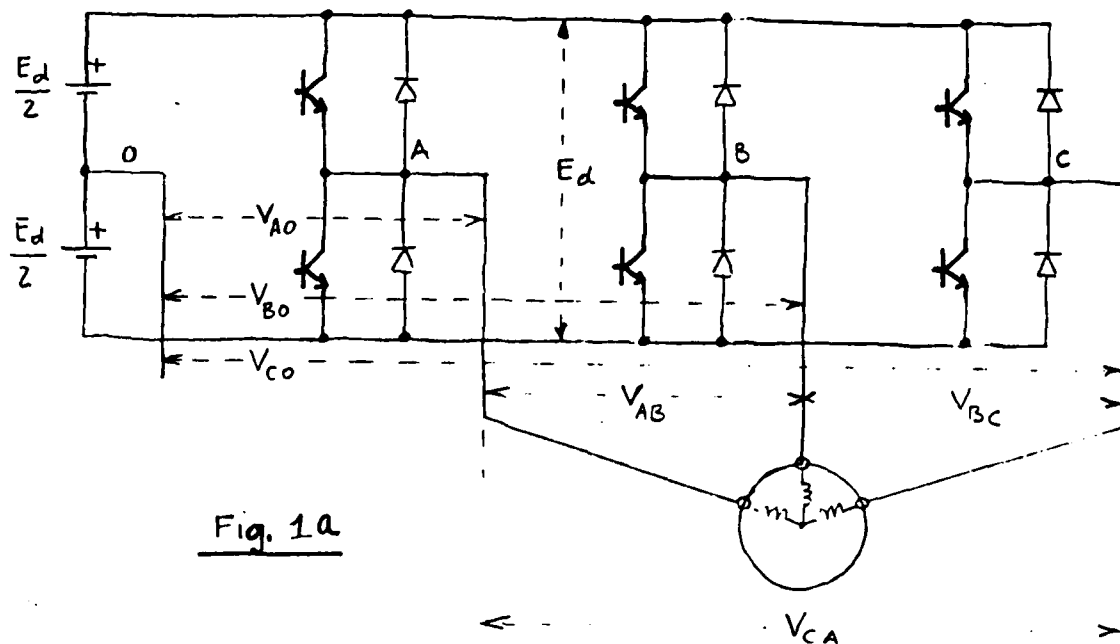


Fig. 1a

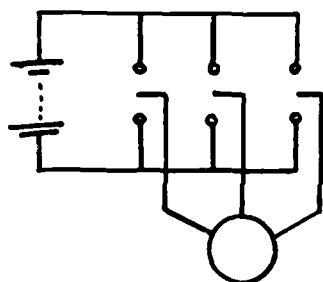


Fig. 1b

$$|E_{AB}| = |E_{BC}| + |E_{CA}| \text{ at any instant}$$

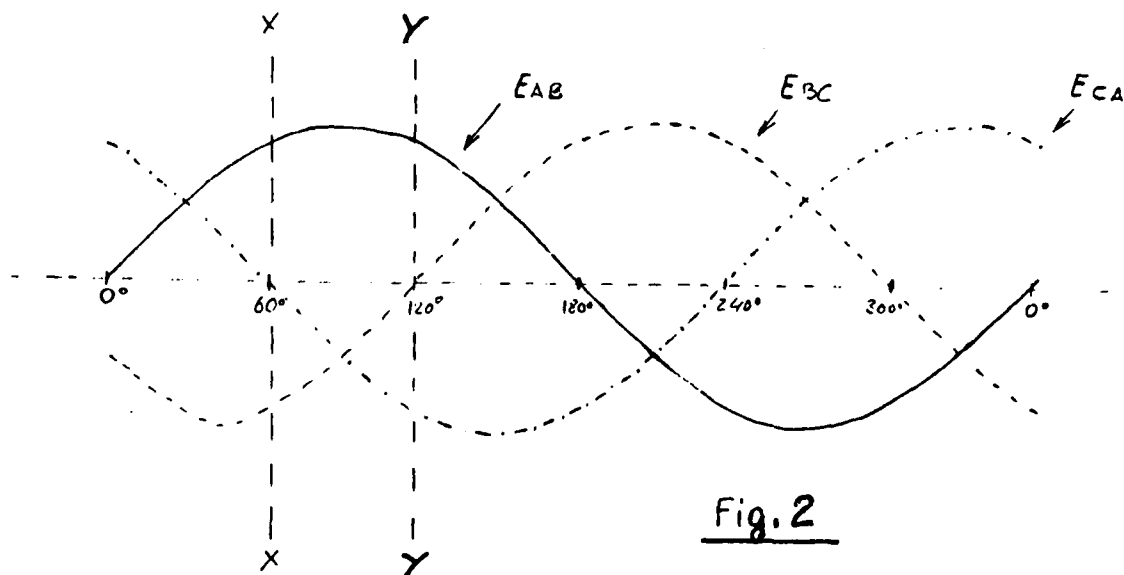


Fig. 2

Permissible Levels of the Line Voltages

It can be seen that a given pole voltage can assume only two discrete levels, $+E_d/2$ or $-E_d/2$, or for brevity, + and -. On the other hand, a line voltage can assume three discrete levels, a + level corresponding to $+E_d$ volts, a - level corresponding to $-E_d$ volts and a 0 level, obtained when the two corresponding poles are in the same state.

While there is no limitation in our freedom to impose to each one of the three pole voltages either of the two possible levels we desire the situation is different as far as the line voltages are concerned. In this case, imposing a given level to one of the line voltages interacts with the freedom of the others to assume any level. This is the interphase compatibility constraint, better illustrated by considering the following table, listing on the left side all the possible combinations of levels that one can confer to the pole voltages (8 in all, one per row) and on the right side the corresponding combinations of line voltage levels that result from the imposed switch configurations.

V_{AO}	V_{BO}	V_{CO}	V_{AB}	V_{BC}	V_{CA}
+	+	+	0	0	0
-	+	+	-	0	+
+	-	+	+	-	0
+	+	-	0	+	-
-	-	+	0	-	+
+	-	-	+	0	-
-	+	-	-	+	0
-	-	-	0	0	0

whenever a line voltage has one of the three possible levels (+, - or 0), the other two line voltages must have the other two possible levels, excluding the possibility for two or three line voltages to share the same level. An exception to the rule is the "all zero" combination of line voltages, for which there are two pole combinations.

Hexamorous Symmetry of the Output Cycle

Considering the fundamental component of each of the three line voltages (sine waves E_{AB} , E_{BC} and E_{CA} , Figure 2), one entire output cycle can be divided in six identical segments, such as the one comprised between the vertical axes XX and YY. In each segment, one of the line sine waves is peculiar in that it rides through its crest with a given polarity, whereas the others rise from zero or decline to zero over the segment with opposite polarity. We shall call "dominant" line voltage in any given segment the line voltage whose fundamental sine wave rides through its crest in the considered segment. The other two line voltages shall be called "complementary" line voltages. The "first complementary" line voltage is the one whose fundamental rises from zero in the considered segment. The "second complementary" line voltage is the one whose fundamental declines to zero in the considered segment. In the segment between the XX and YY axes, Figure 2, E_{AB} is the fundamental of the dominant line, and E_{CA} is the fundamental of the 1st complementary line. Because the segments repeat themselves each 60° interval, with identical relationships between dominant and complementary lines, it suffices to consider a single segment to study the modulation requirements. The relationships of interest are as follows: dominant and complementary

lines always have opposite polarity, except at the segment borderline. The dominant always exceeds either complementary in absolute value. Both complementary lines have same polarity. At any instant, the absolute values of the two complementary lines add up to the absolute value of the dominant.

Since, at any given time the three line voltages are riding through one of the segments of the output cycle, one of them is dominant at that time and the other two are complementary. The modulation mechanism must be capable of manipulating one of the line voltages through a 60° interval according to rules that confer to it the role of dominant line, and at the same time it must be manipulating the other two as required to confer to them the proper complementary role.

Pulse Polarity Consistency Rule

Even without entering into the details of the waveform harmonic analysis, one can state a sensible ground rule that the modulation strategy should follow. That is, the pulses which form the pole-to-pole waveform should have a consistent polarity throughout one half cycle of the fundamental sine-wave. This means that the components of the string of pulses which make up the positive (or negative) half wave can be of variable duration or variable spacing to suit the modulation requirements, but should all be positive (or negative). Figures 3a and 3b illustrate the concept. Both waveforms were carefully constructed to have the same fundamental amplitude, frequency and phase, and the same number of pulses, but Figure 3a conforms to the polarity consistency rule and Figure 3b does not conform. The non-consistent pulses of Figure 3b have



Fig. 3a



Fig. 3b

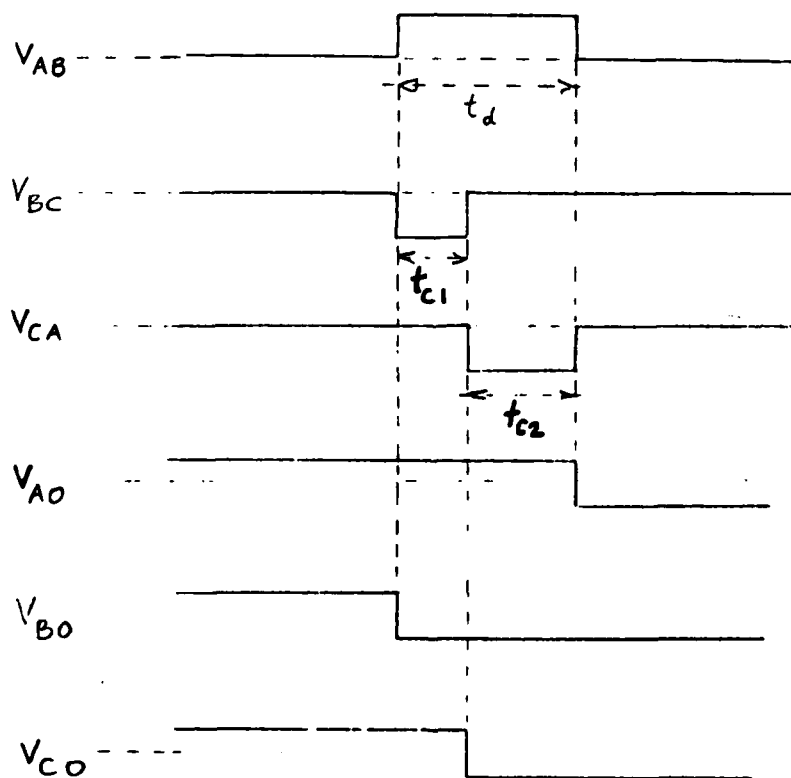


Fig. 4

no effect on the fundamental, being offset by corresponding width variations of the consistent pulses. However, they adversely affect the high order harmonic spectrum and should be avoided by using a proper modulating mechanism.

Optimal Pole Switching Sequence

The above rules and principles can be applied to find out in which order the three poles should typically change state. Assume that at a given instant line voltage V_{AB} is dominant and its fundamental is positive. V_{AB} will, therefore, be made up of a series of positive pulses, one of which is represented on Figure 4 with a width t_d .

a. During time t_d , the two complementary lines can only be negative or zero, as shown in the table of permissible levels. One could be negative for the t_d duration and the other zero for the same duration. However, to play their complementary role, they should both be negative on the average during t_d . Therefore, one of them, V_{BC} for instance, should become negative at the beginning of the t_d duration, and stay in that state for a time t_{C1} , after which V_{BC} returns to the zero level and V_{CA} becomes negative for a time t_{C2} (Figure 4), such that $t_{C1} + t_{C2} = t_d$. The ratio of times t_{C1} and t_{C2} should be the same as the ratio between the two complementary lines' fundamental voltages at the considered instant, and is, therefore, a function of the angular position of the dominant pulse within the output cycle segment.

b. Before and after time t_d , when the dominant line is zero, the two complementaries must also be zero. For the only other permissible combination of levels would be that one complementary is positive and the other negative, but this violates the pulse polarity consistency rule.

c. Before the initiation of a dominant pulse, the all-zero combination of line levels must be achieved with one of the following pole state combinations:

$$A = (+) \quad B = (+) \quad C = (+) \quad (1)$$

$$A = (-) \quad B = (-) \quad C = (-) \quad (2)$$

Assuming state (1), to generate the dominant pulse, pole B must switch from $B = (+)$ to $B = (-)$ at the beginning of the time t_d (see Figure 4) and then A and B must stay in their respective states for the duration of t_d . At some instant during t_d , determined by the t_{C1}/t_{C2} ratio requirement, pole C must follow the transition of B, i.e., must switch from $C = (+)$ to $C = (-)$. At the end of time t_d , all poles must assume the same state, and the most economical way of accomplishing that, in terms of total number of switchings, is to have pole A following the transitions of B and C, i.e., switching from $A = (+)$ to $A = (-)$.

d. The above reasoning shows that from a situation such as

$$A = (+) \quad B = (+) \quad C = (+) \quad (1)$$

a dominant positive pulse is generated by changing the state of the poles in the sequence (B-C-A), ending up with the following situation:

$$A = (-) \quad B = (-) \quad C = (-) \quad (2)$$

Similarly from situation (2), the next dominant positive pulse can be generated by changing the state of the pulses in the sequence (A-C-B) ending up with the original situation (1).

The optimal pole commutation sequence is, therefore, B-C-A, A-C-B, B-C-A, A-C-B, etc., resulting in the "nested" or "encased" pole-pole

waveform configuration of Figure 5. The sequence should continue unchanged throughout an output cycle 60° segment. At the boundary between segments, after a permutation between the role of the poles, accounting for the fact that another line becomes dominant, a similar sequence takes effect.

The following table therefore applies:

Dominant Line Voltage in the Considered Segment	Polarity of the Dominant Fundamental in the Considered Segment	Optimum Pole Commutation Sequence
V_{AB}	+	$B \searrow, C \searrow, A \searrow, A \nearrow, C \nearrow, B$
	-	$A \searrow, C \searrow, B \searrow, B \nearrow, C \nearrow, A \nearrow$
V_{BC}	+	$C \searrow, A \searrow, B \searrow, B \nearrow, A \nearrow, C$
	-	$B \searrow, A \searrow, C \searrow, C \nearrow, A \nearrow, B \nearrow$
V_{CA}	+	$A \searrow, B \searrow, C \searrow, C \nearrow, B \nearrow, A$
	-	$C \searrow, B \searrow, A \searrow, A \nearrow, B \nearrow, C \nearrow$

$B \searrow$ means: pole voltage V_{B0} commutates from the (+) level to the (-) level

$A \nearrow$ means: pole voltage V_{A0} commutates from the (-) level to the (+) level

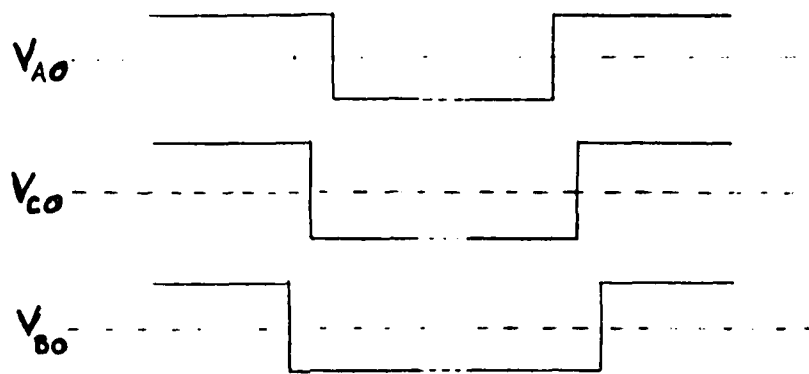


Fig. 5

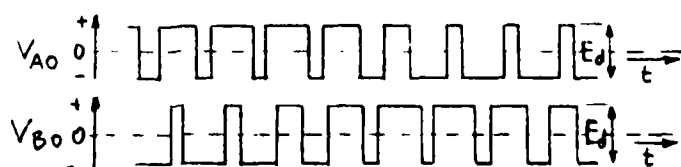


Fig 6a

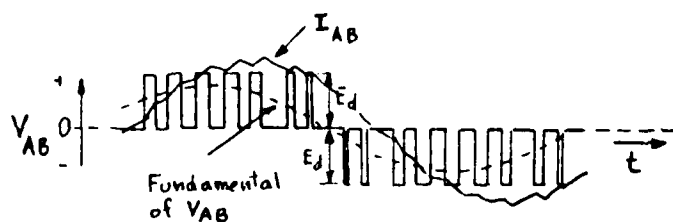


Fig 6b

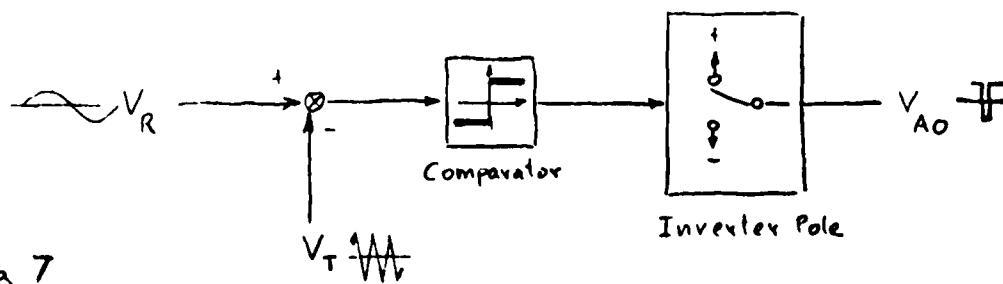


Fig 7

Assume a 60° interval such as that shown in Figure 2. Generation of the output voltage of the dominant phase will require a series of pulses. The more pulses that can be used, the closer the output current can be made to approach a sine wave. Divide the 60° interval into an arbitrary number of pulse intervals t_{pi} . There will be a pulse of length $t_{d1 \rightarrow n}$ (where t_d varies across the 60° interval) in each pulse interval t_{pi} and also the complementary pulses t_{c1} , t_{c2} for the other phases. The dominant pulse length t_d will vary such that $E_d \times t_d$ in any pulse interval t_{pi} will equal the volt time interval ($\int Vdt$) of the desired sine wave during the corresponding t_{pi} interval.

The pulse width is then a sinusoidal function of time. Similarly t_{c1} , and t_{c2} are sinusoidal functions of time across the 60° interval. To vary the amplitude of the desired output, the pulse width of all the pulses is varied by a constant multiple, thus preserving the sinusoidal relationship across the 60° interval. The maximum amplitude, while still generating a 'sine wave' of current in the output, is reached when one t_d pulse fills the whole interval and 'saturation' is reached.

It is now possible to develop techniques for generating pulse patterns that implement the generalized PWM strategy. The traditional analog technique which produces waveforms that fit the rules for optimum modulation strategy uses a triangular carrier waveform which is compared with 3 phase reference waveforms and switches each pole - or + depending on whether the triangle is greater than the reference for that phase or smaller. At one fixed frequency a fixed number of carrier cycles can be fitted inside a reference waveform half cycle. For variable frequency systems unsymmetric pulse patterns (with the effect of low frequency components in the output) arise as non-integral numbers of carrier cycles fit inside a reference half cycle.

The use of digital techniques permits an approach which guarantees a proper number of t_{pi} intervals in each 60° interval and avoids the low frequency beat problem. The steps for generating the waveforms for a digital approach are outlined below.

- o Set desired output frequency.
- o Look up (in Read Only Memory) optimum number of t_{pi} for this frequency.
- o Look up (in ROM) base value for t_d , t_{c1} t_{c2} for each t_{pi} .
- o Input desired output voltage.
- o Multiply t_d , t_{c1} , t_{c2} pulses with voltage factor.
- o Decide which phase is dominant this 60° interval. Output pulse train to power stage.

Estimating the Pole Firing Times

A Fortran program was written for a Data General Nova Computer to compute the switching times for each of the three poles. A triangle wave was simulated and each of the three phases of the sine wave was compared to it. A summary print out displayed the crossover times for the signals as well as their approximate values, switching sequence and pole and line voltages. See Figures 8-12 for a sample output.

APPENDIX C

HARMONIC ANALYSIS PROGRAM

The program appears on the following page. Note that the outputs actually published in this appendix do not print "n DB threshold for printout" and do "m integration steps per increment". Those differences from the program included herein exist because the printout shown is taken from an earlier version of the harmonic analysis program which is substantially the same as the one published. A sample of the output of the latest program is shown below.

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM, AS IN DEVELOPMENT MOTOR CONTROLLER.

9 PWM INCREMENTS PER OUTPUT CYCLE
60 HZ OUTPUT FREQUENCY
-40 DB THRESHOLD FOR PRINTOUT
 PWM INCREMENTS CENTERED

COMPONENT		ABS AMPLITUDE	REL AMPLITUDE	REL AMPL IN DB
60	HZ	0.758347572	1	0
180	HZ	0.031593768	0.041661330	-27.60533794
360	HZ	0.108020575	0.135848757	-17.33888661
480	HZ	0.242079212	0.319219341	-9.918216057
600	HZ	0.076857195	0.101348245	-19.88367535
720	HZ	0.162526080	0.214316082	-13.37890480
780	HZ	0.064951990	0.085649367	-21.34551684
840	HZ	0.074941684	0.098822344	-20.10289702
900	HZ	0.102216315	0.134788214	-17.40696161

```

10 PRINT "HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING"
20 PRINT "CONSTANT AMPLITUDE PWM WAVEFORM, AS IN DEVELOPMENT MOTOR CON..."
30 PRINT
40 PRINT
50 DIM C[100],E[100],G[100],J[100],S[100]
  FOR M=1 TO 100
    C[M]=0
    E[M]=0
    J[M]=0
    JCM=0
    SCM=0
  NEXT M
  DISP "SPEED OUT, HIGHEST HARMONIC";
  INPUT S,H
  DISP "TYPE '1' IF INCREMENTS ARE CENTRD"
  WAIT 1000
  DISP "TYPE '0' OTHERWISE";
  WAIT 1000
  INPUT Q
  DISP "NO. OF INCREMENTS/OUTPUT CYCLE";
  INPUT N
  DISP "AMPL. THRESHOLD FOR PRINTOUT(DB)";
158 INPUT Y
159 Y1=10+(Y/20)
160 X1=2*PI/N
  PRINT N;"PWM INCREMENTS PER OUTPUT CYCLE"
  PRINT S;"HZ OUTPUT FREQUENCY"
  PRINT Y;"DB THRESHOLD FOR PRINTOUT"
  IF Q=0 THEN 200
  PRINT "      PWM INCREMENTS CENTERED"
  GOTO 210
  PRINT "      PWM INCREMENTS LEFT SIDE JUSTIFIED"
  PRINT
  PRINT
  PRINT "COMPONENT","ABS AMPLITUDE","REL AMPLITUDE","REL AMPL IN DB"
  PRINT
  FOR Z1=0 TO (2*PI-X1) STEP X1
    D=(COS(Z1)-COS(Z1+X1))/X1
    FOR M=1 TO H
344 IF Q=0 THEN 349
345 Z8=Z1+(1-D)*X1/2
347 GOTO 350
    Z8=Z1
    C[M]=COS(Z8*M)/M-COS(M*D*X1+M*Z8)/M
    G[M]=SIN(M*D*X1+M*Z8)/M-SIN(Z8*M)/M
    JCM=JCM+G[M]
    ECM=ECM+C[M]
  NEXT M
  NEXT Z1
  FOR M=1 TO H
    SCM=SQR(ECM^2+JCM^2)
    IF SCM/SC[1]<Y1 THEN 440
    PRINT M;S;"HZ",SCM/4,SCM/SC[1],20*LGT(SCM/SC[1])
  NEXT M
  PRINT "-----"
  END

```

75 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 7 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
7	.784978	100	0
490	2.09044E-2	2.66306	-31.4924
504	.162348	20.6818	-13.6882
518	.152733	19.4569	-14.2185
532	.132432	16.8707	-15.4573
546	.16999	21.6554	-13.2887
560	3.17172E-2	4.04052	-27.8713
595	2.17304E-2	2.76829	-31.1558
609	.163155	20.7847	-13.6451
623	.150953	19.2303	-14.3203
637	.134044	17.0761	-15.3522
651	.169526	21.5962	-13.3124
665	3.07466E-2	3.91687	-28.1412
1001	3.36916E-2	4.29205	-27.3467
1015	9.45968E-2	12.0509	-18.3796
1029	1.72731E-2	2.20045	-33.1498
1043	5.62196E-2	7.16193	-22.8994

66 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 8 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
8	.784856	100	0
488	2.02416E-2	2.57902	-31.7709
504	.161665	20.5981	-13.7235
520	.1542	19.6469	-14.1341
536	.131124	16.7067	-15.5422
552	.170344	21.7038	-13.2693
560	2.09113E-2	2.66435	-31.4882
568	3.25119E-2	4.1424	-27.655
576	.162352	20.6856	-13.6867
592	.152735	19.4602	-14.2171
608	.132431	16.8732	-15.456
624	.169989	21.6587	-13.2874
640	3.17167E-2	4.04109	-27.87
680	2.17306E-2	2.76874	-31.1544
696	.163155	20.7879	-13.6438
712	.150954	19.2333	-14.3189
728	.134044	17.0788	-15.3509
744	.169526	21.5996	-13.3111
760	3.07466E-2	3.91748	-28.1399
1000	.032916	4.19389	-27.5477
1016	9.46747E-2	12.0627	-18.3711
1032	1.86955E-2	2.38203	-32.461
1048	5.66596E-2	7.21911	-22.8303
1064	4.98152E-2	6.34705	-23.9486
1096	8.99028E-2	11.4547	-18.8203
1112	4.52183E-2	5.76135	-24.7895
1144	3.21975E-2	4.10234	-27.7394
1160	9.44492E-2	12.034	-18.3918
1176	1.72628E-2	2. ♦INTERRUPTED♦	

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

60 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
9 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
9	.784742	100	0
495	1.96967E-2	2.50996	-32.0067
513	.16108	20.5265	-13.7537
531	.155427	19.8061	-14.064
549	.130045	16.5718	-15.6126
567	.170629	21.7434	-13.2535
585	3.31944E-2	4.22998	-27.4732

54 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
10 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
10	.784588	100	0
490	1.90428E-2	2.42711	-32.2982
510	.160366	20.4395	-13.7906
530	.157808	20.1135	-13.9302
550	.109039	13.8976	-17.1412
570	9.87375E-3	1.25846	-38.0032
590	.121404	15.4736	-16.2082
610	.133283	16.9877	-15.3973
630	.170823	21.7723	-13.2419
650	3.32023E-2	4.23181	-27.4695

48 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
11 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
11	.784373	100	0
473	1.82408E-2	2.32552	-32.6696
495	.159418	20.3242	-13.8397
517	.159642	20.3528	-13.8275
539	.108071	13.778	-17.2163
561	1.09629E-2	1.39766	-37.092
583	.12274	15.6482	-16.1107
605	.11252	14.3453	-16.8658
627	1.00941E-2	1.2869	-37.8091
649	.121394	15.4766	-16.2065
671	.133283	16.9923	-15.3949
693	.170823	21.7783	-13.2395
715	3.32022E-2	4.23295	-27.4671

45 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 12 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
12	.784232	100	0
480	1.77689E-2	2.26577	-32.8957
504	.158821	20.2518	-13.8707
516	1.82374E-2	2.32551	-32.6696
528	.159995	20.4015	-13.8068
540	.159402	20.3259	-13.839
552	.126151	16.0859	-15.8711
564	.159636	20.3558	-13.8263
576	.171551	21.875	-13.201
588	.108066	13.7799	-17.2151
600	3.56977E-2	4.55193	-26.8361
612	1.09602E-2	1.39757	-37.0925
636	.122741	15.6512	-16.1091
660	.11252	14.3477	-16.8643
684	1.00938E-2	1.28709	-37.8078
708	.121394	15.4794	-16.2049
732	.133283	16.9954	-15.3934
756	.170823	21.7822	-13.238
780	3.32022E-2	4.23372	-27.4656

39 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 13 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
13	.783847	100	0
442	1.66339E-2	2.12209	-33.4647
468	.157347	20.0738	-13.9474
494	.163516	20.8607	-13.6134
520	.106033	13.5272	-17.3758
546	1.31788E-2	1.6813	-35.4871
559	1.82412E-2	2.32714	-32.6635
572	.122731	15.6575	-16.1056
585	.159405	20.3362	-13.8346
598	.13018	16.6079	-15.5937
611	.159637	20.3659	-13.8219
624	.171836	21.9222	-13.1823
637	.108066	13.7866	-17.2109
650	3.57124E-2	4.55605	-26.8282
663	1.09599E-2	1.39822	-37.0885
689	.122742	15.6589	-16.1048
715	.11252	14.3548	-16.8601
741	1.00938E-2	1.28773	-37.8035
767	.121394	15.487	-16.2007
793	.133283	17.0037	-15.3891
819	.170823	21.793	-13.2337
845	3.32106E-2	4.23688	-27.4591

36 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 14 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
14	.783578	100	0
434	1.59462E-2	2.03505	-33.8285
462	.156375	19.9566	-13.9983
476	1.66315E-2	2.12251	-33.463
490	.164642	21.0115	-13.5508
504	.157335	20.0791	-13.9451
518	.122355	15.6149	-16.1292
532	.163511	20.8673	-13.6107
546	.172249	21.9824	-13.1585
560	.106029	13.5314	-17.3731
574	.037528	4.78932	-26.3945
588	1.31767E-2	1.68161	-35.4855
602	1.39502E-2	1.78032	-34.99
616	.122732	15.663	-16.1025
630	.159086	20.3025	-13.849
644	.13018	16.6135	-15.5908
658	.15962	20.3706	-13.8199
672	.171836	21.9297	-13.1794
686	.108066	13.7914	-17.2078
700	3.57123E-2	4.5576	-26.8253
714	1.09599E-2	1.3987	-37.0855
742	.122742	15.6642	-16.1018
770	.11252	14.3598	-16.8571
798	1.00943E-2	1.28823	-37.8001
826	.121387	15.4914	-16.1982

36 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 15 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
15	.783578	100	0
465	3.18908E-2	4.06989	-27.8083
495	.312748	39.9128	-7.97775
510	1.66291E-2	2.1222	-33.4643
525	.329278	42.0223	-7.5304
540	.157323	20.0775	-13.9458
555	.244709	31.2297	-10.1086
570	.163507	20.8667	-13.6109
585	.344501	43.9651	-7.13783
600	.106026	13.531	-17.3734
615	7.57787E-2	9.67086	-20.2907
630	1.31746E-2	1.68133	-35.4869
645	9.65923E-3	1.23271	-38.1828
660	.122733	15.6631	-16.1024
675	.158767	20.2619	-13.8664
690	.130179	16.6135	-15.5908
705	.159602	20.3684	-13.8209
720	.171836	21.9296	-13.1794
735	.108067	13.7915	-17.2078
750	3.57123E-2	4.55759	-26.8253
765	1.09599E-2	1.3987	-37.0855
795	.122741	15.6642	-16.1018
825	.11252	14.3598	-16.857

33 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 16 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
16	.783233	100	0
448	1.51523E-2	1.93458	-34.2683
480	.155202	19.8155	-14.0599
496	3.18886E-2	4.07141	-27.8051
512	.167362	21.3681	-13.4047
528	.312736	39.9289	-7.97424
544	.104034	13.2826	-17.5343
560	.329273	42.0403	-7.52668
576	1.51905E-2	1.93946	-34.2464
592	.244706	31.2431	-10.1049
608	.124083	15.8425	-16.0035
624	.344499	43.9843	-7.13406
640	.110641	14.1262	-16.9995
656	7.57779E-2	9.67501	-20.287
672	1.35366E-2	1.7283	-35.2476
688	9.65964E-3	1.2333	-38.1786
704	.122711	15.6673	-16.1001
720	.158768	20.2708	-13.8626
736	.13018	16.6209	-15.5869
752	.159602	20.3774	-13.817
768	.171836	21.9393	-13.1755
784	.108067	13.7976	-17.2039
800	3.57124E-2	4.55962	-26.8214
816	1.09603E-2	1.39936	-37.0814
848	.122736	15.6704	-16.0984

30 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 17 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
17	.782779	100	0
425	1.42317E-2	1.8181	-34.8076
459	.153748	19.6414	-14.1366
493	.169368	21.6368	-13.2961
527	8.67727E-2	11.0852	-19.1051
544	.167358	21.3799	-13.3999
561	.139962	17.8802	-14.9526
578	.104031	13.29	-17.5295
595	.288434	36.8474	-8.67185
612	1.51887E-2	1.94035	-34.2424
629	.249732	31.9032	-9.92332
646	.124084	15.8517	-15.9985
663	.344919	44.0634	-7.11843
680	.110641	14.1343	-16.9945
697	7.58048E-2	9.68406	-20.2788
714	1.35365E-2	1.72929	-35.2426
731	9.65820E-3	1.23383	-38.1749
748	.122711	15.6763	-16.0951
765	.158767	20.2825	-13.8576
782	.130181	16.6306	-15.5819
799	.159598	20.3886	-13.8122
816	.171836	21.9521	-13.1705
833	.108204	13.8231	-17.1879
850	3.57135E-2	4.56239	-26.8161

30 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 18 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
18	.782779	100	0
450	1.42317E-2	1.8181	-34.8076
486	.153748	19.6414	-14.1366
522	.169368	21.6368	-13.2961
558	.205434	26.2442	-11.6193
576	.167353	21.3794	-13.4001
594	3.28119E-2	4.19172	-27.5521
612	.104028	13.2896	-17.5298
630	.247595	31.6302	-9.99795
648	1.51869E-2	1.94013	-34.2434
666	.254757	32.5452	-9.75025
684	.124085	15.8518	-15.9984
702	.34534	44.1171	-7.10785
720	.11064	14.1343	-16.9945
738	7.58318E-2	9.6875	-20.2758
756	1.35364E-2	1.72928	-35.2427
774	9.65676E-3	1.23365	-38.1762
792	.122711	15.6763	-16.0951
810	.158767	20.2825	-13.8576
828	.130181	16.6306	-15.5818
846	.159594	20.3881	-13.8125

27 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 19 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
19	.782167	100	0
418	1.31518E-2	1.68146	-35.4863
456	.151902	19.4207	-14.2347
494	.172569	22.063	-13.1267
532	.116256	14.8634	-16.5576
570	.173017	22.1202	-13.1042
589	.205432	26.2644	-11.6126
608	.124774	15.9523	-15.9435
627	3.28104E-2	4.1948	-27.5458
646	.109589	14.0109	-17.0707
665	.247596	31.6551	-9.99113
684	1.56886E-2	2.00578	-33.9543
703	.254757	32.5707	-9.74347
722	.12405	15.8597	-15.9941
741	.34534	44.1517	-7.10106
760	.110643	14.1456	-16.9876
779	7.58347E-2	9.69547	-20.2686
798	1.35369E-2	1.7307	-35.2356
817	9.54795E-3	1.2207	-38.2678
836	.12271	15.6885	-16.0884

27 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 20 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
20	.782167	100	0
440	1.31518E-2	1.68146	-35.4863
480	.151902	19.4207	-14.2347
520	.172569	22.063	-13.1267
560	.116256	14.8634	-16.5576
600	.173017	22.1202	-13.1042
620	.205429	26.2641	-11.6127
640	8.21937E-2	10.5085	-19.5692
660	3.28088E-2	4.1946	-27.5462
680	.115149	14.7218	-16.6408
700	.247596	31.6552	-9.99111
720	1.61902E-2	2.06992	-33.6809
740	.254757	32.5706	-9.74347
760	.124014	15.8552	-15.9966
780	.34534	44.1517	-7.10105
800	.110645	14.1459	-16.9874
820	7.58377E-2	9.69585	-20.2683
840	1.35374E-2	1.73076	-35.2353

24 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 21 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
21	.781313	100	0
399	1.18724E-2	1.51954	-36.3658
441	.14949	19.1332	-14.3643
483	.176621	22.6056	-12.9157
525	.113315	14.5031	-16.7708
567	.173204	22.1683	-13.0854
609	4.47667E-2	5.72968	-24.8374
651	.211709	27.0966	-11.3417
672	8.21942E-2	10.52	-19.5597
693	3.34292E-2	4.27859	-27.374
714	.115149	14.7379	-16.6313
735	.247546	31.6833	-9.98339
756	1.61904E-2	2.07221	-33.6713
777	.254841	32.617	-9.73112
798	.124014	15.8725	-15.9871
819	.347286	44.449	-7.04275
840	.110646	14.1616	-16.9778

24 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 22 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
22	.781313	100	0
418	1.18724E-2	1.51954	-36.3658
462	.14949	19.1332	-14.3643
506	.176621	22.6056	-12.9157
550	.113315	14.5031	-16.7708
594	.173204	22.1683	-13.0854
638	4.47667E-2	5.72968	-24.8374
682	.217989	27.9004	-11.0878
704	8.21947E-2	10.5201	-19.5596
726	3.40495E-2	4.35799	-27.2143
748	.115149	14.7378	-16.6313
770	.247495	31.6769	-9.98516
792	1.61907E-2	2.07224	-33.6712
814	.254925	32.6278	-9.72825
836	.124013	15.8724	-15.9872

21 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 23 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
23	.78007	100	0
368	1.03427E-2	1.32587	-37.55
414	.146222	18.7447	-14.5424
460	.18191	23.3197	-12.6456
506	.109639	14.055	-17.0434
552	.173256	22.2104	-13.0689
598	4.75861E-2	6.10024	-24.2931
713	.218043	27.9518	-11.0718
736	8.21226E-2	10.5276	-19.5534
759	3.55942E-2	4.56296	-26.8151
782	.115156	14.7622	-16.617
805	.22757	29.1731	-10.7004
828	1.61932E-2	2.07587	-33.656

21 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 24 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
24	.78007	100	0
384	1.03427E-2	1.32587	-37.55
432	.146222	18.7447	-14.5424
480	.18191	23.3197	-12.6456
528	.109639	14.055	-17.0434
576	.173256	22.2104	-13.0689
624	4.75861E-2	6.10024	-24.2931
744	.218097	27.9587	-11.0697
768	8.20505E-2	10.5184	-19.561
792	.037139	4.76098	-26.4461
816	.115162	14.7631	-16.6165
840	.207645	26.6188	11.4962

21 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 25 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
25	.78007	100	0
400	1.03427E-2	1.32587	-37.55
450	.146222	18.7447	-14.5424
500	.18191	23.3197	-12.6456
550	.109639	14.055	-17.0434
600	.173256	22.2104	-13.0689
650	4.75861E-2	6.10024	-24.2931
775	.218151	27.9656	-11.0675
800	8.19784E-2	10.5091	-19.5687
825	3.86837E-2	4.959	-26.0921
850	.115169	14.764	-16.6159

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 26 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
26	.778161	100	0
78	8.67668E-3	1.11502	-39.0543
338	8.50392E-3	1.09282	-39.229
390	.141589	18.1953	-14.8008
442	.189096	24.3003	-12.2878
494	.104923	13.4834	-17.404
546	.173007	22.2329	-13.0601
598	5.13361E-2	6.59711	-23.6129
650	8.82124E-3	1.1336	-38.9108
754	1.72795E-2	2.22055	-33.0708
806	.307558	39.5237	-8.06285
832	8.19807E-2	10.5352	-19.5472

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 27 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
27	.778161	100	0
81	8.67668E-3	1.11502	-39.0543
351	8.50392E-3	1.09282	-39.229
405	.141589	18.1953	-14.8008
459	.189096	24.3003	-12.2878
513	.104923	13.4834	-17.404
567	.173007	22.2329	-13.0601
621	5.13361E-2	6.59711	-23.6129
675	8.82124E-3	1.1336	-38.9108
783	1.72795E-2	2.22055	-33.0708
837	.396965	51.0132	-5.84636

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 28 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
28	.778161	100	0
84	8.67668E-3	1.11502	-39.0543
364	8.50392E-3	1.09282	-39.229
420	.141589	18.1953	-14.8008
476	.189096	24.3003	-12.2878
532	.104923	13.4834	-17.404
588	.173007	22.2329	-13.0601
644	5.13361E-2	6.59711	-23.6129
700	8.82124E-3	1.1336	-38.9108
812	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 29 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
29	.778161	100	0
87	8.67668E-3	1.11502	-39.0543
377	8.50392E-3	1.09282	-39.229
435	.141589	18.1953	-14.8008
493	.189096	24.3003	-12.2878
551	.104923	13.4834	-17.404
609	.173007	22.2329	-13.0601
667	5.13361E-2	6.59711	-23.6129
725	8.82124E-3	1.1336	-38.9108
841	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 28 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
28	.778161	100	0
84	8.67668E-3	1.11502	-39.0543
364	8.50392E-3	1.09282	-39.229
420	.141589	18.1953	-14.8008
476	.189096	24.3003	-12.2878
532	.104923	13.4834	-17.404
588	.173007	22.2329	-13.0601
644	5.13361E-2	6.59711	-23.6129
700	8.82124E-3	1.1336	-38.9108
812	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 29 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
29	.778161	100	0
87	8.67668E-3	1.11502	-39.0543
377	8.50392E-3	1.09282	-39.229
435	.141589	18.1953	-14.8008
493	.189096	24.3003	-12.2878
551	.104923	13.4834	-17.404
609	.173007	22.2329	-13.0601
667	5.13361E-2	6.59711	-23.6129
725	8.82124E-3	1.1336	-38.9108
841	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 30 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
30	.778161	100	0
90	8.67668E-3	1.11502	-39.0543
390	8.50392E-3	1.09282	-39.229
450	.141589	18.1953	-14.8008
510	.189096	24.3003	-12.2878
570	.104923	13.4834	-17.404
630	.173007	22.2329	-13.0601
690	5.13361E-2	6.59711	-23.6129
750	8.82124E-3	1.1336	-38.9108
870	1.72795E-2	2.22055	-33.0708

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 31 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
31	.775015	100	0
93	.012313	1.58875	-35.9789
372	.134617	17.3697	-15.2042
434	.199398	25.7282	-11.7918
496	9.86796E-2	12.7326	-17.9016
558	.172072	22.2025	-13.072
620	5.65064E-2	7.29101	-22.7442
682	1.11796E-2	1.4425	-36.8177
713	1.36635E-2	1.76299	-35.075
775	8.57594E-2	11.0655	-19.1206
837	6.30499E-2	8.13531	-21.7925
899	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 32 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
32	.775015	100	0
96	.012313	1.58875	-35.9789
384	.134617	17.3697	-15.2042
448	.199398	25.7282	-11.7918
512	9.86796E-2	12.7326	-17.9016
576	.172072	22.2025	-13.072
640	5.65064E-2	7.29101	-22.7442
704	1.11796E-2	1.4425	-36.8177
736	1.36635E-2	1.76299	-35.075
800	8.57594E-2	11.0655	-19.1206
864	6.30499E-2	8.13531	-21.7925
928	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 33 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
33	.775015	100	0
99	.012313	1.58875	-35.9789
396	.134617	17.3697	-15.2042
462	.199398	25.7282	-11.7918
528	9.86796E-2	12.7326	-17.9016
594	.172072	22.2025	-13.072
660	5.65064E-2	7.29101	-22.7442
726	1.11796E-2	1.4425	-36.8177
759	1.36635E-2	1.76299	-35.075
825	8.57594E-2	11.0655	-19.1206
891	6.30499E-2	8.13531	-21.7925
957	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 34 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
34	.775015	100	0
102	.012313	1.58875	-35.9789
408	.134617	17.3697	-15.2042
476	.199398	25.7282	-11.7918
544	9.86796E-2	12.7326	-17.9016
612	.172072	22.2025	-13.072
680	5.65064E-2	7.29101	-22.7442
748	1.11796E-2	1.4425	-36.8177
782	1.36635E-2	1.76299	-35.075
850	8.57594E-2	11.0655	-19.1206
918	6.30499E-2	8.13531	-21.7925
986	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 35 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
35	.775015	100	0
105	.012313	1.58875	-35.9789
420	.134617	17.3697	-15.2042
490	.199398	25.7282	-11.7918
560	9.86796E-2	12.7326	-17.9016
630	.172072	22.2025	-13.072
700	5.65064E-2	7.29101	-22.7442
770	1.11796E-2	1.4425	-36.8177
805	1.36635E-2	1.76299	-35.075
875	8.57594E-2	11.0655	-19.1206
945	6.30499E-2	8.13531	-21.7925
1015	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 36 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
36	.775015	100	0
108	.012313	1.58875	-35.9789
432	.134617	17.3697	-15.2042
504	.199398	25.7282	-11.7918
576	9.86796E-2	12.7326	-17.9016
648	.172072	22.2025	-13.072
720	5.65064E-2	7.29101	-22.7442
792	1.11796E-2	1.4425	-36.8177
828	1.36635E-2	1.76299	-35.075
900	8.57594E-2	11.0655	-19.1206
972	6.30499E-2	8.13531	-21.7925
1044	7.02188E-2	9.06032	-20.8571

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 37 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
37	.769282	100	0
111	1.87308E-2	2.43484	-32.2706
333	.123258	16.0225	-15.9054
407	.215324	27.9903	-11.0599
481	9.01056E-2	11.7129	-18.6267
555	.169775	22.0693	-13.1242
629	7.31866E-2	9.51363	-20.4331
703	9.42229E-2	12.2482	-18.2386
777	8.11422E-2	10.5478	-19.5368
851	7.43328E-2	9.66263	-20.2981
925	3.84122E-2	4.99326	-26.0323
999	2.34218E-2	3.04463	-30.3293
1073	.11725	15.2415	-16.3395

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 38 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
38	.769282	100	0
114	1.87308E-2	2.43484	-32.2706
342	.123258	16.0225	-15.9054
418	.215324	27.9903	-11.0599
494	9.01056E-2	11.7129	-18.6267
570	.169775	22.0693	-13.1242
646	7.31866E-2	9.51363	-20.4331
722	9.42229E-2	12.2482	-18.2386
798	8.11422E-2	10.5478	-19.5368
874	7.43328E-2	9.66263	-20.2981
950	3.84122E-2	4.99326	-26.0323
1026	2.34218E-2	3.04463	-30.3293
1102	.11725	15.2415	-16.3395

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 39 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
39	.769282	100	0
117	1.87308E-2	2.43484	-32.2706
351	.123258	16.0225	-15.9054
429	.215324	27.9903	-11.0599
507	9.01056E-2	11.7129	-18.6267
585	.169775	22.0693	-13.1242
663	7.31866E-2	9.51363	-20.4331
741	9.42229E-2	12.2482	-18.2386
819	8.11422E-2	10.5478	-19.5368
897	7.43328E-2	9.66263	-20.2981
975	3.84122E-2	4.99326	-26.0323
1053	2.34218E-2	3.04463	-30.3293
1131	.11725	15.2415	-16.3395

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 40 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
40	.769282	100	0
120	1.87308E-2	2.43484	-32.2706
360	.123258	16.0225	-15.9054
440	.215324	27.9903	-11.0599
520	9.01056E-2	11.7129	-18.6267
600	.169775	22.0693	-13.1242
680	7.31866E-2	9.51363	-20.4331
760	9.42229E-2	12.2482	-18.2386
840	8.11422E-2	10.5478	-19.5368
920	7.43328E-2	9.66263	-20.2981
1000	3.84122E-2	4.99326	-26.0323
1080	2.34218E-2	3.04463	-30.3293
1160	.11725	15.2415	-16.3395

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 41 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
41	.769282	100	0
123	1.87308E-2	2.43484	-32.2706
369	.123258	16.0225	-15.9054
451	.215324	27.9903	-11.0599
533	9.01056E-2	11.7129	-18.6267
615	.169775	22.0693	-13.1242
697	7.31866E-2	9.51363	-20.4331
779	9.42229E-2	12.2482	-18.2386
861	8.11422E-2	10.5478	-19.5368
943	7.43328E-2	9.66263	-20.2981
1025	3.84122E-2	4.99326	-26.0323
1107	2.34218E-2	3.04463	-30.3293
1189	.11725	15.2415	-16.3395

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 42 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
42	.769282	100	0
126	1.87308E-2	2.43484	-32.2706
378	.123258	16.0225	-15.9054
462	.215324	27.9903	-11.0599
546	9.01056E-2	11.7129	-18.6267
630	.169775	22.0693	-13.1242
714	7.31866E-2	9.51363	-20.4331
798	9.42229E-2	12.2482	-18.2386
882	8.11422E-2	10.5478	-19.5368
966	7.43328E-2	9.66263	-20.2981
1050	3.84122E-2	4.99326	-26.0323
1134	2.34218E-2	3.04463	-30.3293
1218	.11725	15.2415	-16.3395

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
43 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
43	.769282	100	0
129	1.87308E-2	2.43484	-32.2706
387	.123258	16.0225	-15.9054
473	.215324	27.9903	-11.0599
559	9.01054E-2	11.7129	-18.6267
645	.169775	22.0693	-13.1242
731	7.31866E-2	9.51363	-20.4331
817	9.42229E-2	12.2482	-18.2386

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 44 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
44	.769282	100	0
132	1.87308E-2	2.43484	-32.2706
396	.123258	16.0225	-15.9054
484	.215324	27.9903	-11.0599
572	9.01056E-2	11.7129	-18.6267
660	.169775	22.0693	-13.1242
748	7.31866E-2	9.51363	-20.4331
836	9.42229E-2	12.2482	-18.2386

12 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 45 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
45	.769282	100	0
135	1.87308E-2	2.43484	-32.2706
405	.123258	16.0225	-15.9054
495	.215324	27.9903	-11.0599
585	9.01056E-2	11.7129	-18.6267
675	.169775	22.0693	-13.1242
765	7.31866E-2	9.51363	-20.4331
855	9.42229E-2	12.2482	-18.2386

9 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 46 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
46	.757151	100	0
138	.031447	4.15333	-27.6321
276	.102612	13.5524	-17.3597
368	.242866	32.0762	-9.87634
460	7.79043E-2	10.2891	-19.7524
552	.162475	21.4587	-13.3679
598	6.46243E-2	8.53519	-21.3757
644	7.46073E-2	9.85369	-20.128
690	.102706	13.5648	-17.3517
736	2.33374E-2	3.08226	-30.2226
782	8.21435E-2	10.849	-19.2922
874	3.72516E-2	4.91996	-26.1608
920	4.69544E-2	6.20146	-24.1501

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
47 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
47	.757151	100	0
141	.031447	4.15333	-27.6321
282	.102612	13.5524	-17.3597
376	.242866	32.0762	-9.87634
470	7.79043E-2	10.2891	-19.7524
564	.162475	21.4587	-13.3679
611	6.46243E-2	8.53519	-21.3757
658	7.46073E-2	9.85369	-20.128
705	.102706	13.5648	-17.3517
752	2.33374E-2	3.08226	-30.2226
799	8.21435E-2	10.849	-19.2922
893	3.72516E-2	4.91996	-26.1608
940	4.69544E-2	6.20146	-24.1501

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
48 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
48	.757151	100	0
144	.031447	4.15333	-27.6321
288	.102612	13.5524	-17.3597
384	.242866	32.0762	-9.87634
480	7.79043E-2	10.2891	-19.7524
576	.162475	21.4587	-13.3679
624	6.46243E-2	8.53519	-21.3757
672	7.46073E-2	9.85369	-20.128
720	.102706	13.5648	-17.3517
768	2.33374E-2	3.08226	-30.2226
816	8.21435E-2	10.849	-19.2922
912	3.72516E-2	4.91996	-26.1608
960	4.69544E-2	6.20146	-24.1501

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
49 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
49	.757151	100	0
147	.031447	4.15333	-27.6321
294	.102612	13.5524	-17.3597
392	.242866	32.0762	-9.87634
490	7.79043E-2	10.2891	-19.7524
588	.162475	21.4587	-13.3679
637	6.46243E-2	8.53519	-21.3757
686	7.46073E-2	9.85369	-20.128
735	.102706	13.5648	-17.3517
784	2.33374E-2	3.08226	-30.2226
833	8.21435E-2	10.849	-19.2922
931	3.72516E-2	4.91996	-26.1608
980	4.69544E-2	6.20146	-24.1501

```

9 ♦INTERRUPTED♦
140 H=INT(850/S)
45 FOR S=50 TO 60
RUN

```

78/12/22. 09.48.48.
PROGRAM HARM

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
50 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
50	.757151	100	0
150	.031447	4.15333	-27.6321
300	.102612	13.5524	-17.3597
400	.242866	32.0762	-9.87634
500	7.79043E-2	10.2891	-19.7524
600	.162475	21.4587	-13.3679
650	6.46243E-2	8.53519	-21.3757
700	7.46073E-2	9.85369	-20.128
750	.102706	13.5648	-17.3517
800	2.33374E-2	3.08226	-30.2226
850	8.21435E-2	10.849	-19.2922

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
51 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
51	.757151	100	0
153	.031447	4.15333	-27.6321
306	.102612	13.5524	-17.3597
408	.242866	32.0762	-9.87634
510	7.79043E-2	10.2891	-19.7524
612	.162475	21.4587	-13.3679
663	6.46243E-2	8.53519	-21.3757
714	7.46073E-2	9.85369	-20.128
765	.102706	13.5648	-17.3517
816	2.33374E-2	3.08226	-30.2226

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
52 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
52	.757151	100	0
156	.031447	4.15333	-27.6321
312	.102612	13.5524	-17.3597
416	.242866	32.0762	-9.87634
520	7.79043E-2	10.2891	-19.7524
624	.162475	21.4587	-13.3679
676	6.46243E-2	8.53519	-21.3757
728	7.46073E-2	9.85369	-20.128
780	.102706	13.5648	-17.3517
832	2.33374E-2	3.08226	-30.2226

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
53 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
53	.757151	100	0
159	.031447	4.15333	-27.6321
318	.102612	13.5524	-17.3597
424	.242866	32.0762	-9.87634
530	7.79043E-2	10.2891	-19.7524
636	.162475	21.4587	-13.3679
689	6.46243E-2	8.53519	-21.3757
742	7.46073E-2	9.85369	-20.128
795	.102706	13.5648	-17.3517
848	2.33374E-2	3.08226	-30.2226

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
54 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
54	.757151	100	0
162	.031447	4.15333	-27.6321
324	.102612	13.5524	-17.3597
432	.242866	32.0762	-9.87634
540	7.79043E-2	10.2891	-19.7524
648	.162475	21.4587	-13.3679
702	6.46243E-2	8.53519	-21.3757
756	7.46073E-2	9.85369	-20.128
810	.102706	13.5648	-17.3517

9 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 55 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
55	.757151	100	0
165	.031447	4.15333	-27.6321
330	.102612	13.5524	-17.3597
440	.242866	32.0762	-9.87634
550	7.79043E-2	10.2891	-19.7524
660	.162475	21.4587	-13.3679
715	6.46243E-2	8.53519	-21.3757
770	7.46073E-2	9.85369	-20.128
825	.102706	13.5648	-17.3517

9 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 56 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
56	.757151	100	0
168	.031447	4.15333	-27.6321
336	.102612	13.5524	-17.3597
448	.242866	32.0762	-9.87634
560	7.79043E-2	10.2891	-19.7524
672	.162475	21.4587	-13.3679
728	6.46243E-2	8.53519	-21.3757
784	7.46073E-2	9.85369	-20.128
840	.102706	13.5648	-17.3517

9 PWM INCREMENTS PER OUTPUT CYCLE
 5 INTEGRATION STEPS PER INCREMENT
 57 HZ OUTPUT FREQUENCY
 PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
57	.757151	100	0
171	.031447	4.15333	-27.6321
342	.102612	13.5524	-17.3597
456	.242866	32.0762	-9.87634
570	7.79043E-2	10.2891	-19.7524
684	.162475	21.4587	-13.3679
741	6.46243E-2	8.53519	-21.3757
798	7.46073E-2	9.85369	-20.128

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
58 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
58	.757151	100	0
174	.031447	4.15333	-27.6321
348	.102612	13.5524	-17.3597
464	.242866	32.0762	-9.87634
580	7.79043E-2	10.2891	-19.7524
696	.162475	21.4587	-13.3679
754	6.46243E-2	8.53519	-21.3757
812	7.46073E-2	9.85369	-20.128

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
59 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
59	.757151	100	0
177	.031447	4.15333	-27.6321
354	.102612	13.5524	-17.3597
472	.242866	32.0762	-9.87634
590	7.79043E-2	10.2891	-19.7524
708	.162475	21.4587	-13.3679
767	6.46243E-2	8.53519	-21.3757
826	7.46073E-2	9.85369	-20.128

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
60 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
60	.757151	100	0
180	.031447	4.15333	-27.6321
360	.102612	13.5524	-17.3597
480	.242866	32.0762	-9.87634
600	7.79043E-2	10.2891	-19.7524
720	.162475	21.4587	-13.3679
780	6.46243E-2	8.53519	-21.3757
840	7.46073E-2	9.85369	-20.128

APPENDIX D

Average Values of PWM Steps - Computer Output

AVERAGE VALUE OF SINE WAVE OVER SPECIFIED INTERVAL
FORMULATION:

$$G(X) = \frac{\cos(X_1) - \cos(X_2)}{(X_1 - X_2)}$$

INTERVAL (DEGREES)		AVERAGE VALUE	9.000 INTERVALS
FROM	TO		
0.000	40.000	0.335	
40.000	80.000	0.849	
80.000	120.000	0.965	
120.000	160.000	0.630	
160.000	200.000	0.000	
200.000	240.000	-0.630	
240.000	280.000	-0.965	
280.000	320.000	-0.849	
320.000	360.000	-0.335	

INTERVAL (DEGREES)		AVERAGE VALUE	15.0 INTERVALS
FROM	TO		
0.000	24.000	0.206	
24.000	48.000	0.583	
48.000	72.000	0.860	
72.000	96.000	0.987	
96.000	120.000	0.944	
120.000	144.000	0.738	
144.000	168.000	0.401	
168.000	192.000	0.000	
192.000	216.000	-0.404	
216.000	240.000	-0.738	
240.000	264.000	-0.944	
264.000	288.000	-0.987	
288.000	312.000	-0.860	
312.000	336.000	-0.583	
336.000	360.000	-0.206	

INTERVAL (DEGREES)		AVERAGE VALUE	210.00 INTERVALS
FROM	TO		
0.000	17.143	0.148	
17.143	34.286	0.432	
34.286	51.429	0.678	
51.429	68.571	0.863	
68.571	85.714	0.971	
85.714	102.857	0.993	
102.857	120.000	0.927	
120.000	137.143	0.779	
137.143	154.286	0.561	
154.286	171.429	0.294	
171.429	188.571	0.000	
188.571	205.714	-0.294	
205.714	222.857	-0.561	
222.857	240.000	-0.779	
240.000	257.143	-0.927	
257.143	274.286	-0.993	
274.286	291.429	-0.971	
291.429	308.571	-0.863	
308.571	325.714	-0.678	
325.714	342.857	-0.432	
342.857	360.000	-0.148	

INTERVAL (DEGREES)		AVERAGE VALUE	27.000 INTERVALS
FROM	TO		
0.000	13.333	0.116	
13.333	26.667	0.341	
26.667	40.000	0.548	
40.000	53.333	0.726	
53.333	66.667	0.864	
66.667	80.000	0.956	
80.000	93.333	0.996	
93.333	106.667	0.983	
106.667	120.000	0.916	
120.000	133.333	0.800	
133.333	146.667	0.641	
146.667	160.000	0.448	
160.000	173.333	0.230	
173.333	186.667	0.000	
186.667	200.000	-0.230	
200.000	213.333	-0.448	
213.333	226.667	-0.641	
226.667	240.000	-0.800	
240.000	253.333	-0.916	
253.333	266.667	-0.983	
266.667	280.000	-0.996	
280.000	293.333	-0.956	
293.333	306.667	-0.864	
306.667	320.000	-0.726	
320.000	333.333	-0.548	
333.333	346.667	0.341	
346.667	360.000	0.116	

INTERVAL (DEGREES)		AVERAGE VALUE	31,000 INTERVALS
FROM	TO		
0.000	10.909	0.095	
10.909	21.818	0.281	
21.818	32.727	0.458	
32.727	43.636	0.617	
43.636	54.545	0.755	
54.545	65.454	0.865	
65.455	76.364	0.944	
76.364	87.273	0.988	
87.273	98.182	0.997	
98.182	109.091	0.970	
109.091	120.000	0.908	
120.000	130.909	0.813	
130.909	141.818	0.689	
141.818	152.727	0.540	
152.727	163.636	0.371	
163.636	174.545	0.189	
174.545	185.454	0.000	
185.455	196.364	-0.189	
196.364	207.273	-0.371	
207.273	218.182	-0.540	
218.182	229.091	-0.689	
229.091	240.000	-0.813	
240.000	250.909	-0.908	
250.909	261.818	-0.970	
261.818	272.727	-0.997	
272.727	283.636	-0.988	
283.636	294.545	-0.944	
294.545	305.454	-0.865	
305.455	316.364	-0.755	
316.364	327.273	-0.617	
327.273	338.182	-0.458	
338.182	349.091	-0.281	
349.091	360.000	-0.095	

INTERVAL (DEGREES)		AVERAGE VALUE	NUMBER OF INTERVALS
FROM	TO		
0.000	9.231	0.000	
9.231	18.462	0.239	
18.462	27.692	0.392	
27.692	36.923	0.534	
36.923	46.154	0.662	
46.154	55.385	0.774	
55.385	64.615	0.865	
64.615	73.846	0.934	
73.846	83.077	0.979	
83.077	92.308	0.998	
92.308	101.538	0.992	
101.538	110.769	0.959	
110.769	120.000	0.902	
120.000	129.231	0.822	
129.231	138.462	0.720	
138.462	147.692	0.600	
147.692	156.923	0.464	
156.923	166.154	0.316	
166.154	175.385	0.160	
175.385	184.615	0.000	
184.615	193.846	-0.160	
193.846	203.077	-0.316	
203.077	212.308	-0.464	
212.308	221.538	-0.600	
221.538	230.769	-0.720	
230.769	240.000	-0.822	
240.000	249.231	-0.902	
249.231	258.462	-0.959	
258.462	267.692	-0.992	
267.692	276.923	-0.998	
276.923	286.154	-0.979	
286.154	295.385	-0.934	
295.385	304.615	-0.865	
304.615	313.846	-0.774	
313.846	323.077	-0.662	
323.077	332.308	-0.534	
332.308	341.538	-0.392	
341.538	350.769	-0.239	

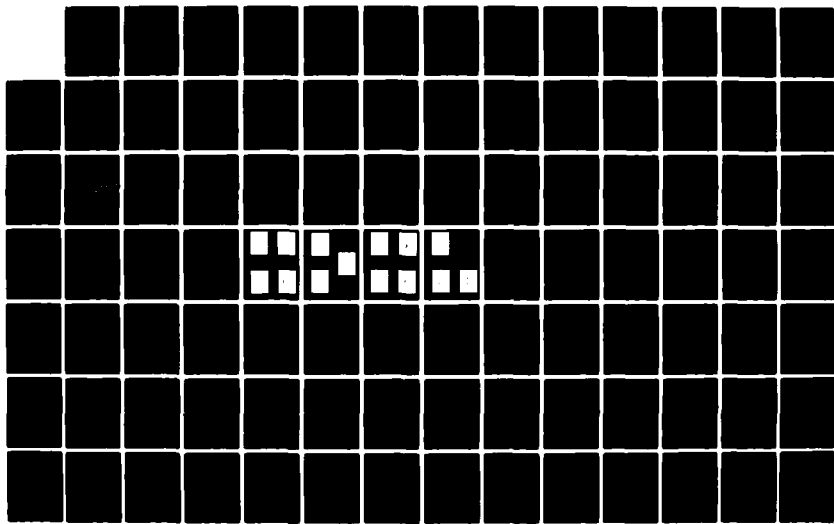
INTERVAL (DEGREES)		AVERAGE	INTERVALS
FROM	TO	VALUE	
0.000	8.000	0.070	
8.000	16.000	0.208	
16.000	24.000	0.342	
24.000	32.000	0.469	
32.000	40.000	0.587	
40.000	48.000	0.694	
48.000	56.000	0.787	
56.000	64.000	0.865	
64.000	72.000	0.926	
72.000	80.000	0.970	
80.000	88.000	0.994	
88.000	96.000	0.999	
96.000	104.000	0.984	
104.000	112.000	0.950	
112.000	120.000	0.898	
120.000	128.000	0.828	
128.000	136.000	0.743	
136.000	144.000	0.642	
144.000	152.000	0.529	
152.000	160.000	0.406	
160.000	168.000	0.275	
168.000	176.000	0.139	
176.000	184.000	-0.000	
184.000	192.000	-0.139	
192.000	200.000	-0.275	
200.000	208.000	-0.406	
208.000	216.000	-0.529	
216.000	224.000	-0.642	
224.000	232.000	-0.743	
232.000	240.000	-0.828	
240.000	248.000	-0.898	
248.000	256.000	-0.950	
256.000	264.000	-0.984	
264.000	272.000	-0.999	
272.000	280.000	-0.994	
280.000	288.000	-0.970	
288.000	296.000	-0.926	
296.000	304.000	-0.865	
304.000	312.000	-0.787	
312.000	320.000	-0.694	
320.000	328.000	-0.587	
328.000	336.000	-0.469	
336.000	344.000	-0.342	
344.000	352.000	-0.208	
352.000	360.000	-0.070	

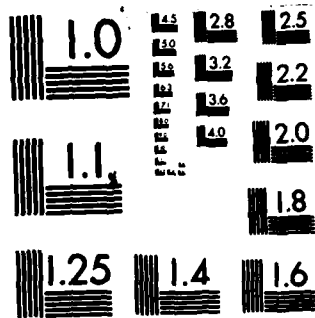
ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE
OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83
DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

2/4

F/G 9/3

N1





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INTERVAL (DEGREES)		AVERAGE VALUE	51.000 INTERVALS
FROM	TO		
0.000	7.059	0.062	
7.059	14.118	0.184	
14.118	21.176	0.303	
21.176	28.235	0.418	
28.235	35.294	0.526	
35.294	42.353	0.627	
42.353	49.412	0.717	
49.412	56.471	0.798	
56.471	63.529	0.865	
63.529	70.588	0.920	
70.588	77.647	0.961	
77.647	84.706	0.988	
84.706	91.765	0.999	
91.765	98.824	0.995	
98.824	105.882	0.976	
105.882	112.941	0.943	
112.941	120.000	0.895	
120.000	127.059	0.833	
127.059	134.118	0.759	
134.118	141.176	0.673	
141.176	148.235	0.577	
148.235	155.294	0.473	
155.294	162.353	0.361	
162.353	169.412	0.244	
169.412	176.471	0.123	
176.471	183.529	-0.000	
183.529	190.588	-0.123	
190.588	197.647	-0.244	
197.647	204.706	-0.361	
204.706	211.765	-0.473	
211.765	218.824	-0.577	
218.824	225.882	-0.673	
225.882	232.941	-0.759	
232.941	240.000	-0.833	
240.000	247.059	-0.895	
247.059	254.118	-0.943	
254.118	261.176	-0.976	
261.176	268.235	-0.995	
268.235	275.294	-0.999	
275.294	282.353	-0.988	
282.353	289.412	-0.961	
289.412	296.471	-0.920	
296.471	303.529	-0.865	
303.529	310.588	-0.798	
310.588	317.647	-0.717	
317.647	324.706	-0.627	
324.706	331.765	-0.526	
331.765	338.824	-0.418	
338.824	345.882	-0.303	
345.882	352.941	-0.184	

INTERVAL (DEGREES)		AVERAGE	51.000 INTERVALS
FROM	TO	VALUE	
0.000	6.316	0.055	
6.316	12.632	0.165	
12.632	18.947	0.272	
18.947	25.263	0.376	
25.263	31.579	0.476	
31.579	37.895	0.570	
37.895	44.211	0.656	
44.211	50.526	0.735	
50.526	56.842	0.805	
56.842	63.158	0.866	
63.158	69.474	0.915	
69.474	75.789	0.954	
75.789	82.105	0.981	
82.105	88.421	0.996	
88.421	94.737	0.999	
94.737	101.053	0.990	
101.053	107.368	0.969	
107.368	113.684	0.936	
113.684	120.000	0.892	
120.000	126.316	0.837	
126.316	132.632	0.772	
132.632	138.947	0.697	
138.947	145.263	0.614	
145.263	151.579	0.523	
151.579	157.895	0.427	
157.895	164.211	0.325	
164.211	170.526	0.219	
170.526	176.842	0.110	
176.842	183.158	0.000	
183.158	189.474	-0.110	
189.474	195.789	-0.219	
195.789	202.105	-0.325	
202.105	208.421	-0.427	
208.421	214.737	-0.523	
214.737	221.053	-0.614	
221.053	227.368	-0.697	
227.368	233.684	-0.772	
233.684	240.000	-0.837	
240.000	246.316	-0.892	
246.316	252.632	-0.936	
252.632	258.947	-0.969	
258.947	265.263	-0.990	
265.263	271.579	-0.999	
271.579	277.895	-0.996	
277.895	284.211	-0.981	
284.211	290.526	-0.954	
290.526	296.842	-0.915	
296.842	303.158	-0.866	
303.158	309.474	-0.805	
309.474	315.789	-0.735	
315.789	322.105	-0.656	
322.105	328.421	-0.570	
328.421	334.737	-0.476	
334.737	341.053	-0.376	
341.053	347.368	-0.272	
347.368	353.684	-0.165	

INTERVAL (DEGREES) AVERAGE
FROM TO VALUE

63,000 INTERVALS

0.000	5.714	0.050
5.714	11.429	0.149
11.429	17.143	0.247
17.143	22.857	0.342
22.857	28.571	0.434
28.571	34.286	0.521
34.286	40.000	0.604
40.000	45.714	0.680
45.714	51.429	0.749
51.429	57.143	0.812
57.143	62.857	0.866
62.857	68.571	0.911
68.571	74.286	0.948
74.286	80.000	0.975
80.000	85.714	0.992
85.714	91.429	0.999
91.429	97.143	0.997
97.143	102.857	0.984
102.857	108.571	0.962
108.571	114.286	0.930
114.286	120.000	0.890
120.000	125.714	0.840
125.714	131.429	0.782
131.429	137.143	0.716
137.143	142.857	0.643
142.857	148.571	0.563
148.571	154.286	0.478
154.286	160.000	0.388
160.000	165.714	0.295
165.714	171.429	0.198
171.429	177.143	0.100
177.143	182.857	0.000
182.857	188.571	-0.100
188.571	194.286	-0.198
194.286	200.000	-0.295
200.000	205.714	-0.388
205.714	211.429	-0.478
211.429	217.143	-0.563
217.143	222.857	-0.643
222.857	228.571	-0.716
228.571	234.286	-0.782
234.286	240.000	-0.840
240.000	245.714	-0.890
245.714	251.429	-0.930
251.429	257.143	-0.962
257.143	262.857	-0.984
262.857	268.571	-0.997
268.571	274.286	-0.999
274.286	280.000	-0.992
280.000	285.714	-0.975
285.714	291.429	-0.948
291.429	297.143	-0.911
297.143	302.857	-0.866
302.857	308.571	-0.812
308.571	314.286	-0.749
314.286	320.000	-0.680
320.000	325.714	-0.604
325.714	331.429	-0.521
331.429	337.143	-0.434
337.143	342.857	-0.342
342.857	348.571	-0.247
348.571	354.286	-0.149

DB

INTERVAL (DEGREES)
FROM TO

AVERAGE
VALUE

69.000 INTERVALS

0.000	5.217	0.045
5.217	10.435	0.136
10.435	15.652	0.226
15.652	20.870	0.313
20.870	26.087	0.398
26.087	31.304	0.480
31.304	36.522	0.558
36.522	41.739	0.631
41.739	46.957	0.699
46.957	52.174	0.761
52.174	57.391	0.817
57.391	62.609	0.866
62.609	67.826	0.908
67.826	73.043	0.942
73.043	78.261	0.968
78.261	83.478	0.987
83.478	88.696	0.997
88.696	93.913	0.999
93.913	99.130	0.993
99.130	104.348	0.979
104.348	109.565	0.956
109.565	114.783	0.926
114.783	120.000	0.888
120.000	125.217	0.842
125.217	130.435	0.790
130.435	135.652	0.731
135.652	140.870	0.666
140.870	146.087	0.595
146.087	151.304	0.519
151.304	156.522	0.440
156.522	161.739	0.356
161.739	166.957	0.270
166.957	172.174	0.181
172.174	177.391	0.091
177.391	182.609	-0.000
182.609	187.826	-0.091
187.826	193.043	-0.181
193.043	198.261	-0.270
198.261	203.478	-0.356
203.478	208.696	-0.440
208.696	213.913	-0.519
213.913	219.130	-0.595
219.130	224.348	-0.666
224.348	229.565	-0.731
229.565	234.783	-0.790
234.783	240.000	-0.842
240.000	245.217	-0.888
245.217	250.435	-0.926
250.435	255.652	-0.956
255.652	260.870	-0.979
260.870	266.087	-0.993
266.087	271.304	-0.999
271.304	276.522	-0.997
276.522	281.739	-0.987
281.739	286.957	-0.968
286.957	292.174	-0.942
292.174	297.391	-0.908

D9

297.391	302.609	-0.866
302.609	307.826	-0.817
307.826	313.043	-0.761
313.043	318.261	-0.699
318.261	323.478	-0.631
323.478	328.696	-0.558
328.696	333.913	-0.480
333.913	339.130	-0.398
339.130	344.348	-0.313
344.348	349.565	-0.226
349.565	354.783	-0.136

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE	75.000 INTERVALS
0.000	4.800	0.042	
4.800	9.600	0.125	
9.600	14.400	0.208	
14.400	19.200	0.289	
19.200	24.000	0.368	
24.000	28.800	0.445	
28.800	33.600	0.518	
33.600	38.400	0.588	
38.400	43.200	0.653	
43.200	48.000	0.714	
48.000	52.800	0.770	
52.800	57.600	0.821	
57.600	62.400	0.866	
62.400	67.200	0.905	
67.200	72.000	0.937	
72.000	76.800	0.963	
76.800	81.600	0.982	
81.600	86.400	0.994	
86.400	91.200	0.999	
91.200	96.000	0.998	
96.000	100.800	0.989	
100.800	105.600	0.973	
105.600	110.400	0.951	
110.400	115.200	0.922	
115.200	120.000	0.886	
120.000	124.800	0.844	
124.800	129.600	0.796	
129.600	134.400	0.743	
134.400	139.200	0.684	
139.200	144.000	0.621	
144.000	148.800	0.553	
148.800	153.600	0.482	
153.600	158.400	0.407	
158.400	163.200	0.329	
163.200	168.000	0.249	
168.000	172.800	0.167	
172.800	177.600	0.084	
177.600	182.400	-0.000	
182.400	187.200	-0.084	
187.200	192.000	-0.167	
192.000	196.800	-0.249	
196.800	201.600	-0.329	

D10

201.600	206.400	-0.407
206.400	211.200	-0.482
211.200	216.000	-0.553
216.000	220.800	-0.621
220.800	225.600	-0.684
225.600	230.400	-0.743
230.400	235.200	-0.796
235.200	240.000	0.844
240.000	244.800	-0.886
244.800	249.600	-0.922
249.600	254.400	-0.951
254.400	259.200	-0.973
259.200	264.000	-0.989
264.000	268.800	-0.998
268.800	273.600	-0.999
273.600	278.400	-0.994
278.400	283.200	-0.982
283.200	288.000	-0.963
288.000	292.800	-0.937
292.800	297.600	-0.905
297.600	302.400	-0.866
302.400	307.200	-0.821
307.200	312.000	-0.770
312.000	316.800	-0.714
316.800	321.600	-0.653
321.600	326.400	-0.588
326.400	331.200	-0.518
331.200	336.000	-0.445
336.000	340.800	-0.368
340.800	345.600	-0.289
345.600	350.400	-0.208
350.400	355.200	-0.125

INTERVAL (DEGREES)
FROM TO

AVERAGE
VALUE

81.000 INTERVALS

0.000	4.444	0.039
4.444	8.889	0.116
8.889	13.333	0.193
13.333	17.778	0.268
17.778	22.222	0.342
22.222	26.667	0.414
26.667	31.111	0.483
31.111	35.556	0.549
35.556	40.000	0.612
40.000	44.444	0.672
44.444	48.889	0.727
48.889	53.333	0.778
53.333	57.778	0.824
57.778	62.222	0.866
62.222	66.667	0.902
66.667	71.111	0.933
71.111	75.556	0.958
75.556	80.000	0.977
80.000	84.444	0.991
84.444	88.889	0.998
88.889	93.333	1.000

D11

93.333	97.778	0.995
97.778	102.222	0.985
102.222	106.667	0.968
106.667	111.111	0.946
111.111	115.556	0.918
115.556	120.000	0.885
120.000	124.444	0.846
124.444	128.889	0.802
128.889	133.333	0.753
133.333	137.778	0.700
137.778	142.222	0.643
142.222	146.667	0.581
146.667	151.111	0.517
151.111	155.556	0.449
155.556	160.000	0.378
160.000	164.444	0.305
164.444	168.889	0.231
168.889	173.333	0.154
173.333	177.778	0.077
177.778	182.222	-0.000
182.222	186.667	-0.077
186.667	191.111	-0.154
191.111	195.556	-0.231
195.556	200.000	-0.305
200.000	204.444	-0.378
204.444	208.889	-0.449
208.889	213.333	-0.517
213.333	217.778	-0.581
217.778	222.222	-0.643
222.222	226.667	-0.700
226.667	231.111	-0.753
231.111	235.556	-0.802
235.556	240.000	-0.846
240.000	244.444	-0.885
244.444	248.889	-0.918
248.889	253.333	-0.946
253.333	257.778	-0.968
257.778	262.222	-0.985
262.222	266.667	-0.995
266.667	271.111	-1.000
271.111	275.556	-0.998
275.556	280.000	-0.991
280.000	284.444	-0.977
284.444	288.889	-0.958
288.889	293.333	-0.933
293.333	297.778	-0.902
297.778	302.222	-0.866
302.222	306.667	-0.824
306.667	311.111	-0.778
311.111	315.556	-0.727
315.556	320.000	-0.672
320.000	324.444	-0.612
324.444	328.889	-0.549
328.889	333.333	-0.483
333.333	337.778	-0.414
337.778	342.222	-0.342
342.222	346.667	-0.268
346.667	351.111	-0.193
351.111	355.556	-0.116

T

INTERVAL (DEGREES)		AVERAGE VALUE	0.0000 INTERVALS
FROM	TO		
0.000	4.138	0.036	
4.138	8.276	0.108	
8.276	12.414	0.180	
12.414	16.552	0.250	
16.552	20.690	0.319	
20.690	24.828	0.387	
24.828	28.966	0.452	
28.966	33.103	0.515	
33.103	37.241	0.576	
37.241	41.379	0.633	
41.379	45.517	0.688	
45.517	49.655	0.738	
49.655	53.793	0.785	
53.793	57.931	0.828	
57.931	62.069	0.866	
62.069	66.207	0.900	
66.207	70.345	0.929	
70.345	74.483	0.953	
74.483	78.621	0.972	
78.621	82.759	0.987	
82.759	86.897	0.996	
86.897	91.034	1.000	
91.034	95.172	0.998	
95.172	99.310	0.992	
99.310	103.448	0.980	
103.448	107.586	0.963	
107.586	111.724	0.942	
111.724	115.862	0.915	
115.862	120.000	0.883	
120.000	124.138	0.847	
124.138	128.276	0.807	
128.276	132.414	0.762	
132.414	136.552	0.713	
136.552	140.690	0.661	
140.690	144.828	0.605	
144.828	148.966	0.546	
148.966	153.103	0.484	
153.103	157.241	0.420	
157.241	161.379	0.353	
161.379	165.517	0.285	
165.517	169.655	0.215	
169.655	173.793	0.144	
173.793	177.931	0.072	
177.931	182.069	0.000	
182.069	186.207	-0.072	
186.207	190.345	-0.144	
190.345	194.483	-0.215	
194.483	198.621	-0.285	
198.621	202.759	-0.353	
202.759	206.897	-0.420	
206.897	211.034	-0.484	
211.034	215.172	-0.546	
215.172	219.310	-0.605	
219.310	223.448	-0.661	
223.448	227.586	-0.713	
227.586	231.724	-0.762	
231.724	235.862	-0.807	
235.862	240.000	-0.847	

240.000	240.000	0.000
244.138	248.276	-0.915
248.276	252.414	-0.942
252.414	256.552	-0.963
256.552	260.690	-0.980
260.690	264.828	-0.992
264.828	268.966	-0.998
268.966	273.103	-1.000
273.103	277.241	-0.996
277.241	281.379	-0.987
281.379	285.517	-0.972
285.517	289.655	-0.953
289.655	293.793	-0.929
293.793	297.931	-0.900
297.931	302.069	-0.866
302.069	306.207	-0.828
306.207	310.345	-0.785
310.345	314.483	-0.738
314.483	318.621	-0.688
318.621	322.759	-0.633
322.759	326.897	-0.576
326.897	331.034	-0.515
331.034	335.172	-0.452
335.172	339.310	-0.387
339.310	343.448	-0.319
343.448	347.586	-0.250
347.586	351.724	-0.180
351.724	355.862	-0.108
355.862	360.000	-0.036

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	3.871	0.034
3.871	7.742	0.101
7.742	11.613	0.168
11.613	15.484	0.234
15.484	19.355	0.299
19.355	23.226	0.363
23.226	27.097	0.425
27.097	30.968	0.485
30.968	34.839	0.543
34.839	38.710	0.599
38.710	42.581	0.651
42.581	46.452	0.701
46.452	50.323	0.748
50.323	54.194	0.791
54.194	58.065	0.830
58.065	61.935	0.866
61.935	65.806	0.898
65.806	69.677	0.925
69.677	73.548	0.949
73.548	77.419	0.968
77.419	81.290	0.983
81.290	85.161	0.993
85.161	89.032	0.999
89.032	92.903	1.000
92.903	96.774	0.996
96.774	100.645	0.988
100.645	104.516	0.976
104.516	108.387	0.959
108.387	112.258	0.938

93.000 INTERVALS

D14

I

I

I

I

10

—

1



T

[illegible]

• •

•

1

1

112.258
116.129
120.000
123.871
127.742
131.613
135.484
139.355
143.226
147.097
150.968
154.839
158.710
162.581
166.452
170.323
174.194
178.065
181.935
185.806
189.677
193.548
197.419
201.290
205.161
209.032
212.903
216.774
220.645
224.516
228.387
232.258
236.129
240.000
243.871
247.742
251.613
255.484
259.355
263.226
267.097
270.968
274.839
278.710
282.581
286.452
290.323
294.194
298.065
301.935
305.806
309.677
313.548
317.419
321.290
325.161
329.032
332.903
336.774
340.645
344.516
348.387
352.258

116.129
120.000
123.871
127.742
131.613
135.484
139.355
143.226
147.097
150.968
154.839
158.710
162.581
166.452
170.323
174.194
178.065
181.935
185.806
189.677
193.548
197.419
201.290
205.161
209.032
212.903
216.774
220.645
224.516
228.387
232.258
236.129
240.000
243.871
247.742
251.613
255.484
259.355
263.226
267.097
270.968
274.839
278.710
282.581
286.452
290.323
294.194
298.065
301.935
305.806
309.677
313.548
317.419
321.290
325.161
329.032
332.903
336.774
340.645
344.516
348.387
352.258
356.129

0.912
0.882
0.848
0.811
0.770
0.725
0.676
0.625
0.571
0.514
0.455
0.394
0.331
0.267
0.201
0.135
0.067
-0.000
-0.067
-0.135
-0.201
-0.267
-0.331
-0.394
-0.455
-0.514
-0.571
-0.625
-0.676
-0.725
-0.770
-0.811
-0.848
-0.882
-0.912
-0.938
-0.959
-0.976
-0.988
-0.996
-1.000
-0.999
-0.993
-0.983
-0.968
-0.949
-0.925
-0.898
-0.866
-0.830
-0.791
-0.748
-0.701
-0.651
-0.599
-0.543
-0.485
-0.425
-0.363
-0.299
-0.234
-0.168
-0.100

D15

INTERVAL (DEGREES)		AVERAGE VALUE	99.000 INTERVALS
FROM	TO		
0.000	3.636	0.032	
3.636	7.273	0.095	
7.273	10.909	0.158	
10.909	14.545	0.220	
14.545	18.182	0.282	
18.182	21.818	0.342	
21.818	25.455	0.401	
25.455	29.091	0.458	
29.091	32.727	0.514	
32.727	36.364	0.567	
36.364	40.000	0.618	
40.000	43.636	0.667	
43.636	47.273	0.713	
47.273	50.909	0.756	
50.909	54.545	0.796	
54.545	58.182	0.832	
58.182	61.818	0.866	
61.818	65.455	0.896	
65.455	69.091	0.922	
69.091	72.727	0.945	
72.727	76.364	0.964	
76.364	80.000	0.979	
80.000	83.636	0.990	
83.636	87.273	0.997	
87.273	90.909	1.000	
90.909	94.545	0.999	
94.545	98.182	0.994	
98.182	101.818	0.985	
101.818	105.455	0.972	
105.455	109.091	0.955	
109.091	112.727	0.934	
112.727	116.364	0.909	
116.364	120.000	0.881	
120.000	123.636	0.850	
123.636	127.273	0.814	
127.273	130.909	0.776	
130.909	134.545	0.734	
134.545	138.182	0.690	
138.182	141.818	0.643	
141.818	145.455	0.593	
145.455	149.091	0.541	
149.091	152.727	0.486	
152.727	156.364	0.430	
156.364	160.000	0.372	
160.000	163.636	0.312	
163.636	167.273	0.251	
167.273	170.909	0.189	
170.909	174.545	0.127	
174.545	178.182	0.063	
178.182	181.818	0.000	
181.818	185.455	-0.063	
185.455	189.091	-0.127	
189.091	192.727	-0.189	
192.727	196.364	-0.251	
196.364	200.000	-0.312	

200.000	203.636	-0.372
203.636	207.273	-0.430
207.273	210.909	-0.486
210.909	214.545	-0.541
214.545	218.182	-0.593
218.182	221.818	-0.643
221.818	225.455	-0.690
225.455	229.091	-0.734
229.091	232.727	-0.776
232.727	236.364	-0.814
236.364	240.000	-0.850
240.000	243.636	-0.881
243.636	247.273	-0.909
247.273	250.909	-0.934
250.909	254.545	-0.955
254.545	258.182	-0.972
258.182	261.818	-0.985
261.818	265.455	-0.994
265.455	269.091	-0.999
269.091	272.727	-1.000
272.727	276.364	-0.997
276.364	280.000	-0.990
280.000	283.636	-0.979
283.636	287.273	-0.964
287.273	290.909	-0.945
290.909	294.545	-0.922
294.545	298.182	-0.896
298.182	301.818	-0.866
301.818	305.455	-0.832
305.455	309.091	-0.796
309.091	312.727	-0.756
312.727	316.364	-0.713
316.364	320.000	-0.667
320.000	323.636	-0.618
323.636	327.273	-0.567
327.273	330.909	-0.514
330.909	334.545	-0.458
334.545	338.182	-0.401
338.182	341.818	-0.342
341.818	345.455	-0.282
345.455	349.091	-0.220
349.091	352.727	-0.158
352.727	356.364	-0.095
356.364	360.000	-0.032

ADVANCED MOTOR CONTROLLER

PHASE II REPORT

- PULSE WIDTH MODULATION AND FEEDBACK CONTROL CIRCUITS

CONTENTS

1. Summary
2. Theoretical Basis for Control Circuit Design Approach
3. Control Circuit Design
 - 3.1 Hardware
 - 3.2 Software
4. Test and Measurement
5. Conclusion and Recommendations

APPENDIX A - Hardware

APPENDIX B - Software

ADVANCED MOTOR CONTROLLER

PHASE II REPORT

1. The Phase I study showed that it was possible to utilize an 8 bit micro-processor (M6800) to generate 3 phase sine wave output pulse width modulation waveforms for variable frequency control, via a power bridge, of an induction motor. Based on the theoretical work of Abbondanti (^(W) Research and Development Center) a preliminary look was taken at air gap flux synthesis for use as a feedback function for optimum motor control with load.

This Phase II study has taken the preliminary work a step further and has developed a design for the pulse width modulation and for the feedback utilizing separate processors for each function and connected via a parallel port.

The PWM generation utilizes volt second integral determination of each pulse width and pulse by pulse modification of pulse width is possible in response to voltage control requirements for a wide control bandwidth.

The feedback processor samples the input to the motor. Since power does not fluctuate at the supply frequency, but is "invariant", asynchronous sampling of the output is possible provided that in-phase and quadrature components of current and voltage are sampled simultaneously. The derived reactive power components are calculated and a digitally smoothed error signal is passed to the PWM processor.

The error signal is used to scale the PWM processor output voltage and also as an indication of overload. Overload causes a controlled ramp down in frequency until load is reduced or the off condition is reached.

Reversing is implemented as a ramp down to zero speed followed by a ramp up in reverse rotation to the set speed.

Controls can be analog or digital, via potentiometers and switches or a serial port. A terminal can be connected to the serial port for control and diagnostic purposes using a built in software monitor routine.

2. Theoretical Basis for Control Circuit Design Approach

The Phase I report described in detail basic modulation generation techniques and also the basic calculations leading to the Air Gap Flux synthesis approach to be employed here.

The following pages serve to consolidate these facts and summarize the basis for the control circuit design.

The control circuits assume that a three phase bridge is to be driven to produce variable frequency three phase waveforms with a sinewave fundamental component as shown in Figure 2.1. The three phase waveforms repeat every 60 degrees provided allowance is made for interchange of the dominant and complementary roles of the line to line voltages. Within a 60 degree period the waveforms are symmetrical about the centerpoint provided allowance is made for interchange of the role of the two complementary waveforms. Thus by organized use of symmetry properties the Pulse Width Modulation waveforms can be generated economically.

The number of pulses per 60° interval are chosen to be odd to ease symmetry requirements. A minimum of 3 pulse periods per 60° interval is defined at max frequency (60 Hz) hence approximate frequency at "carrier".

$f_c = 3 \times 6 \times 60 = 1080 \text{ Hz}$ for $926 \mu\text{s}$ per pulse period maximum. It should be noted that the power switches make only one transition each during a pulse period, thus their effective switching frequency is one half the carrier frequency or 540 Hz approximately; the actual number varies as the output frequency and number of pulses per 60° interval is varied.

Figure 2.2 illustrates both the volt second integral approach to pulse width determination and also the basic pulsewidths associated with a single pulse period of the several that would be present in one 60 degree interval.

Figure 2.3 extends this to the time relationship actually used in the micro-processor software. The numbers stored in memory for each frequency reflect the fact that the pulses are multiplied in amplitude by $B+$ when they are converted to the output waveform in the power stage. The scale factor (SF) likewise takes into account the variation in $B+$ compared with the assumed nominal value used to derive the stored data. Hence the multiplication factor $\frac{B+_{\text{ref}}}{B+}$. A value for $B+$ is acquired via an A/D converter as an input to the PWM microprocessor software.

More specifically and in the terms used in the software the values calculated for the pulse widths are as follows.

CONTROLLER CONFIGURATION AND 60° SYMMETRY

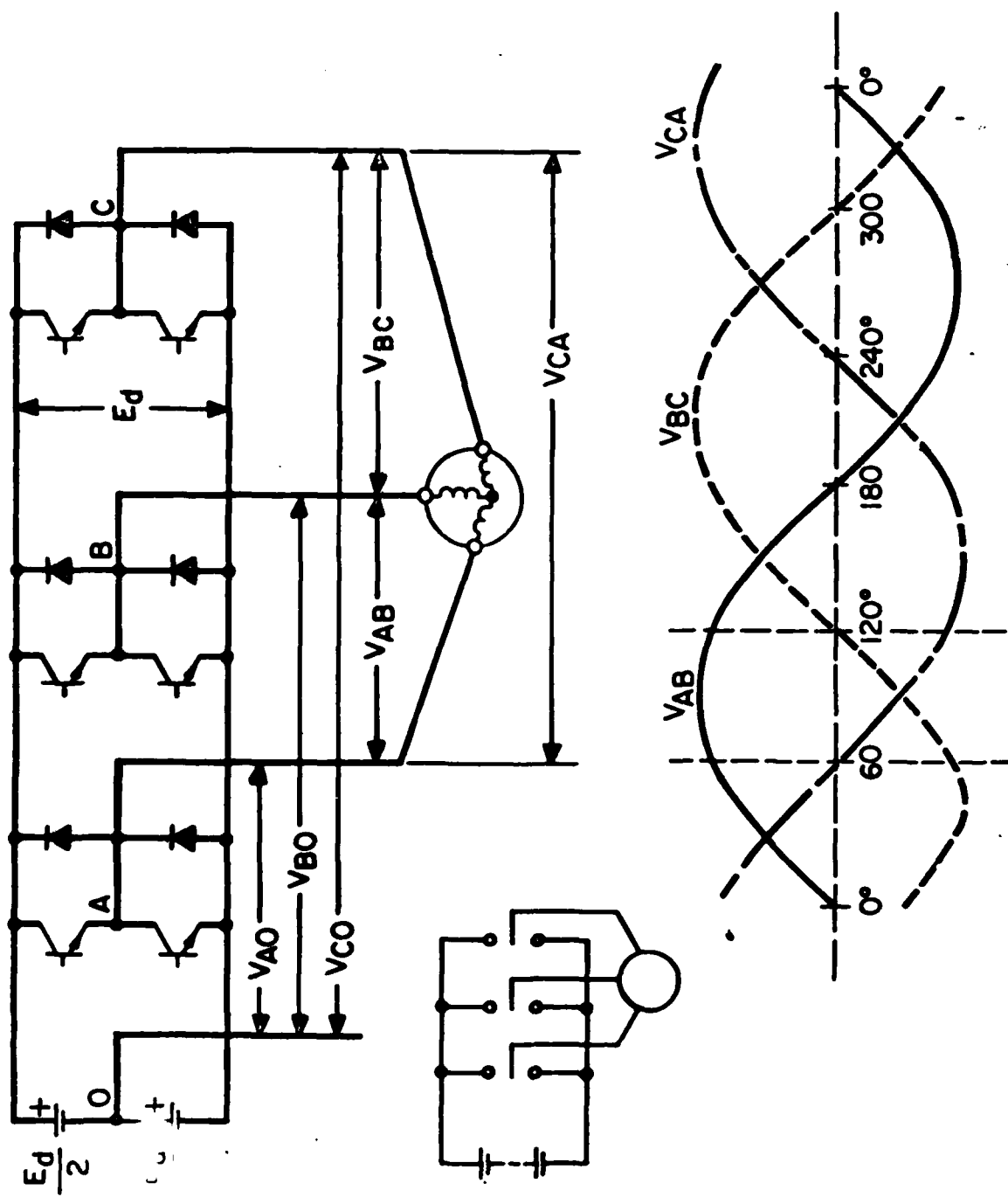
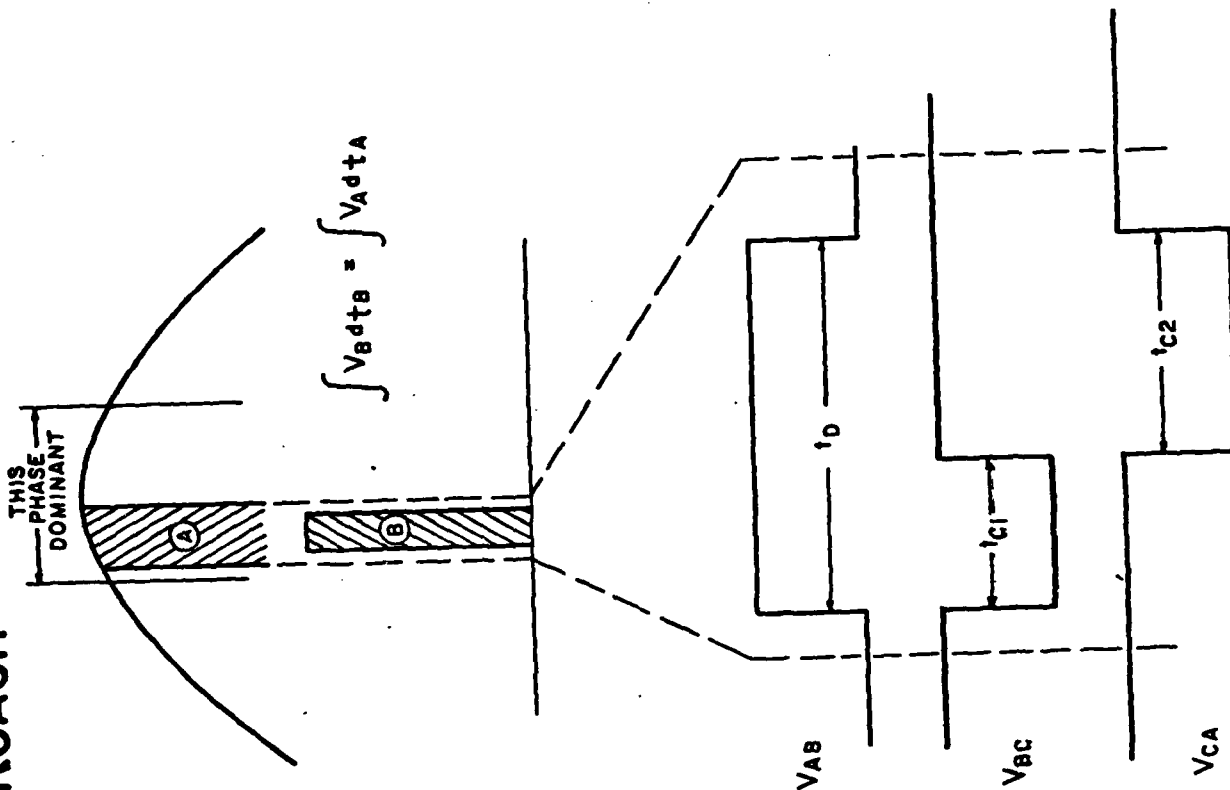
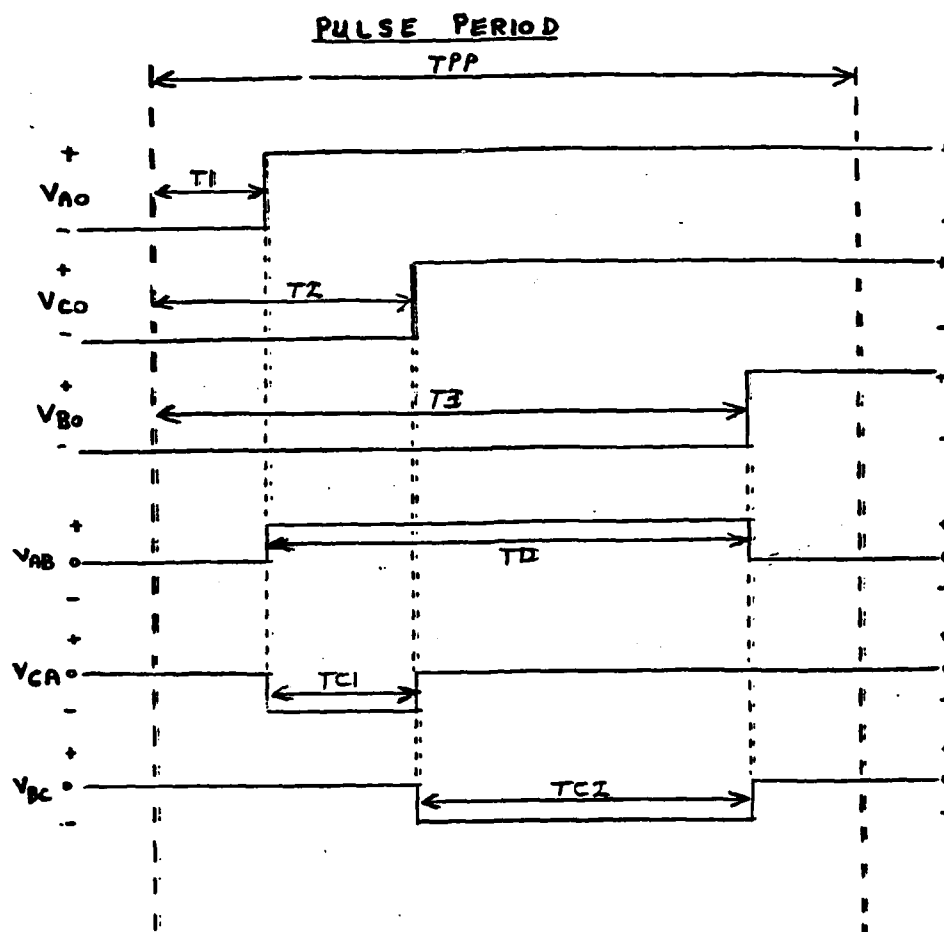


FIG. 2.1

MODULATION APPROACH





• $T_{Dref} + T_{C2ref}$ STORED IN MEMORY

$$T_{ref} = \frac{1}{B_{ref}^+} \int_T^{T+TPP} V_{ref} \cdot \sin(\omega t + \phi) dt$$

• $SF = \frac{V_{out}}{V_{ref}} \cdot \frac{B_{ref}^+}{B^+}$

$T_D = T_{Dref} \cdot SF$; $T_{C2} = T_{C2ref} \cdot SF$

• $T_1 = \frac{TPP - T_D}{2}$; $T_3 = T_1 + T_D$
 $T_2 = T_3 - T_{C2}$ (or $T_2 = T_1 + T_{C2}$)

SOFTWARE CONTROLLED TIMER OUTPUTS

$$PW_{Dom_n} = \int_{\frac{\pi}{3}(1+\frac{n-1}{N})}^{\frac{\pi}{3}(1+\frac{n}{N})} \sin \omega t \cdot d\omega t$$

$$PW_{Com_n} = \int_{\frac{\pi}{3}(\frac{n-1}{N})}^{\frac{\pi}{3}(\frac{n}{N})} \sin(\omega t + \frac{2\pi}{3}) \cdot d\omega t$$

FOR UNIT AMPLITUDE
REFERENCE, AND
N=1 TO N. ωt IS THE
ANGLE INTO THE CYCLE

Where N is number of pulse periods per 60° interval at the output frequency of interest F and N is the Nth pulse in the interval. The second complementary pulse widths are antisymmetrical with PWCOM and are not generated separately.

TOTAL PULSE PERIOD LENGTH $T_{pp} = \frac{1}{F} \cdot \frac{1}{6} \cdot \frac{1}{N}$

PERIOD TO TIMER #1 $T_1 = (T_{pp} - PW_{Dom}) / 2$

PERIOD TO TIMER #3 $T_3 = T_1 + PW_{Dom}$

PERIOD TO TIMER #2 $T_2 = T_3 - PW_{Com}, n = \text{ODD}$
 $T_2 = T_1 + PW_{Com}, n = \text{EVEN}$

If V_{BC} is dominant in the 60° period of interest then the pole centerpoint voltages with respect to B- are

$$V_B = \overline{T_1} \cdot (n = \text{ODD}) + T_3 \cdot (n = \text{EVEN})$$

$$V_A = \overline{T_2} \cdot (n = \text{ODD}) + T_2 \cdot (n = \text{EVEN})$$

$$V_C = \overline{T_3} \cdot (n = \text{ODD}) + T_1 \cdot (n = \text{EVEN})$$

There are two power switches in each pole, an upper and a lower. The base drive waveforms for upper switches (neglecting shoot through protection, etc.) have the same form as V_B, V_A, V_C . The drive for the lower switches the inverse of V_B, V_A, V_C .

The output line to line voltages are the physical combination of the pole voltages across the motor windings

$$V_{BC} = V_B - V_C$$

$$V_{AB} = V_A - V_B$$

$$V_{CA} = V_C - V_A$$

The PWM processor implements the control strategy of Fig. 2.4. A nominal $V/F = \text{constant}$ curve is chosen such that the current in the unloaded motor is a minimum at each frequency. This corresponds to optimum excitation at that load at that frequency. In response to load change as sensed by the feedback processor, the voltage is increased to maintain the excitation at the optimum level although the load is changing. Demand for excessive voltage as sensed by monitoring the level of an error signal causes the controller to attempt to shed load by reducing frequency. If load reduces bringing the error signal within bounds the controller stops at the new frequency and will return to higher frequency if the load transient is removed. Alternatively it will continue on down to zero frequency - off - if the overload condition persists. Hysteresis incorporated into the level detection provides for stable operation in this mode.

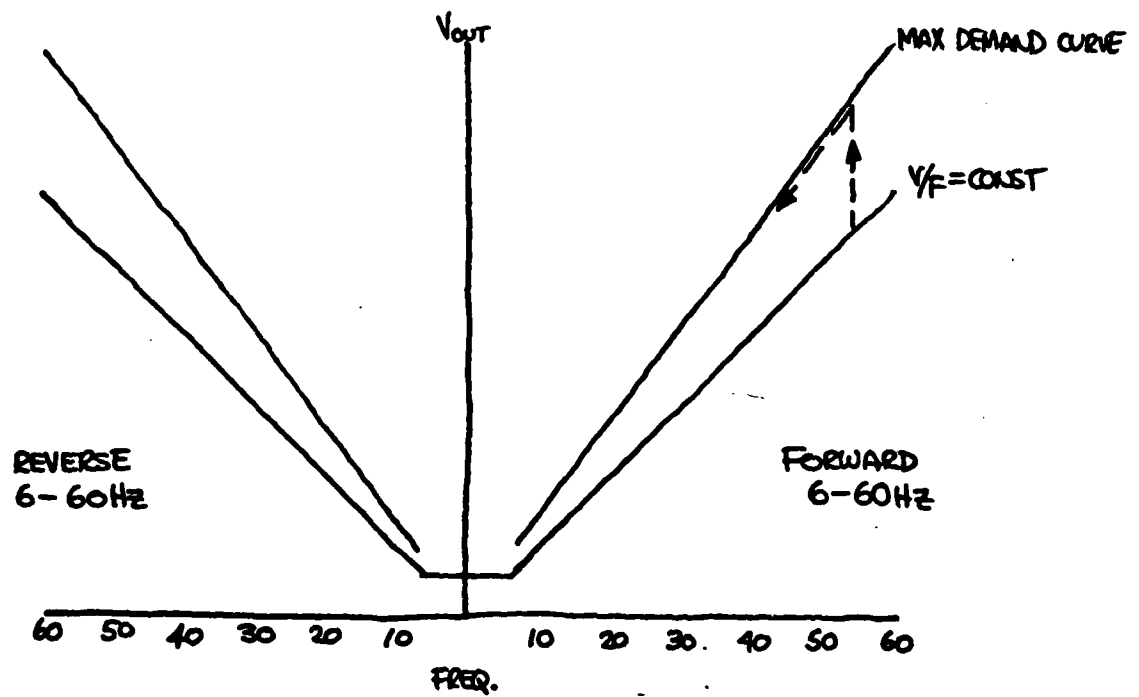


FIG. 2.4 CONTROL STRATEGY

3.0 Control System Design

3.1 Hardware

Design of the advanced motor controller was split into two basic sections: PWM Processor and Feedback Processor. An available laboratory power bridge was married with the processors. Figure 3.1 shows how each of these sections relates to the others. The power stage was employed in order to make operation with a test motor possible. The two processors are constructed on five wire-wrap boards, three being dedicated to the PWM processor, one to the feedback processor, and one to data acquisition and driver output functions.

Figure 3.2 and 3.3 show the construction of the PWM processor in block form. Analog speed command (terminal control optional) and battery B+ voltage is digitized and combined with feedback error information from the feedback processor to modify the output voltage scaling generated by the basic three phase sine wave algorithm. A monitor program to provide a terminal interface for optional keyboard control resides in PROM as well as the PWM routines. Figure 3.4 is a memory map showing locations in the PWM processor.

The configuration of the feedback processor is shown in Figure 3.5. The associated memory map appears in Figure 3.6. The feedback processor accepts the digitized form of the D-Q variables generated by the sensing and low pass filter circuits and computes the motor air gap flux using motor parameters stored in PROM. The calculated air gap flux is compared to a reference and the subsequent difference is digitally filtered and passed to the PWM processor through the V* (feedback error) latch.

Figure 3.7 is a hardware block diagram, showing the individual wire-wrap cards, their interconnections, and the System I/O. Appendix A contains schematics of the five cards and their interconnections.

3.2 Software

Figures 3.9 through 3.11 contain details of the PWM processor software. Figure 3.12 shows the feedback processor algorithm. Appendix B contains printouts of the system software.

The main control loop shown in Figure 3.9 first initializes all pointers, RAM, and flags. Next flags are set according to motor, power input, and command conditions. If no protection flags are set, the program ramps the output frequency up to the commanded frequency. Overload, B+, and reversing flags are checked each time through the loop. The PTM interrupt routine generates the dominant and complementary

times from frequency table information stored in PROM. The condition of the direction flag determines the order of output times and thus the output phase sequence. Prior to the time computations in the PIM interrupt routine, the input B+ voltage is read and the scale factor adjusted to keep the output independent of B+ variations. The quotient $B+ \text{ ref}/B+$ is stored in a table in PROM for values from below B+ min to above B+ max. The feedback error, V^* , is read during the beginning of each 60° output segment and is used to adjust the scale factor. During operation with a keyboard terminal, the monitor routine shown in Figure 3.11 is employed.

The feedback microprocessor routine containing the flux computation and filtering algorithms is shown in Figure 3.12. The D-Q variables are digitized by four A/D converters after being generated as analog signals.

The air gap flux is then computed utilizing motor parameters stored as K1 in PROM. The calculated flux, W_x , is compared to the reference. Both K1 and reference R^* are stored in PROM indexed to frequency. Thus, before calculating the flux and the error, the processor reads a word containing the output frequency from the PWM processor.

FIG. 3.1

ADVANCED MOTOR CONTROLLER BLOCK DIAGRAM

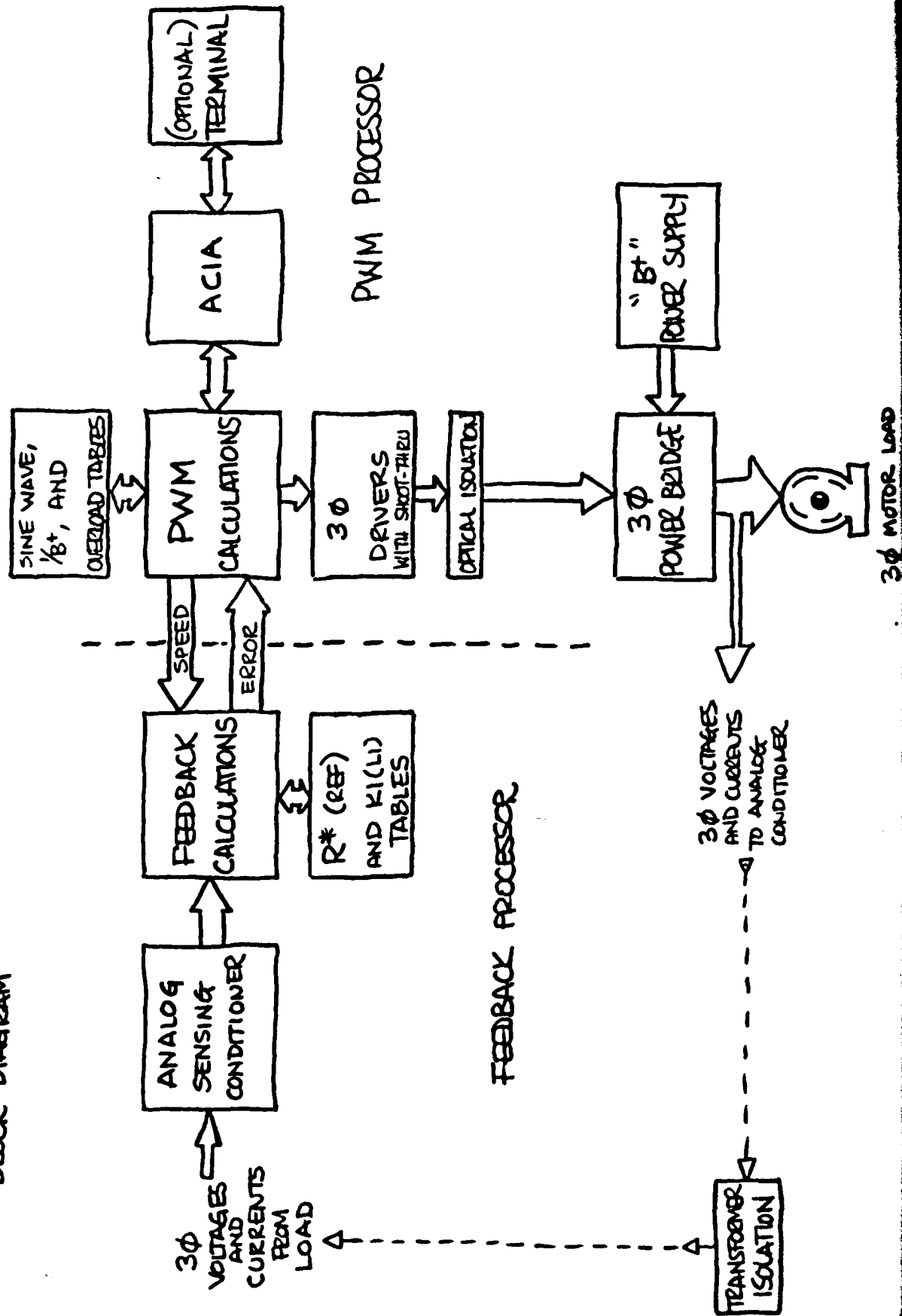


FIG. 3.2
PWM MICROPROCESSOR
BLOCK DIAGRAM

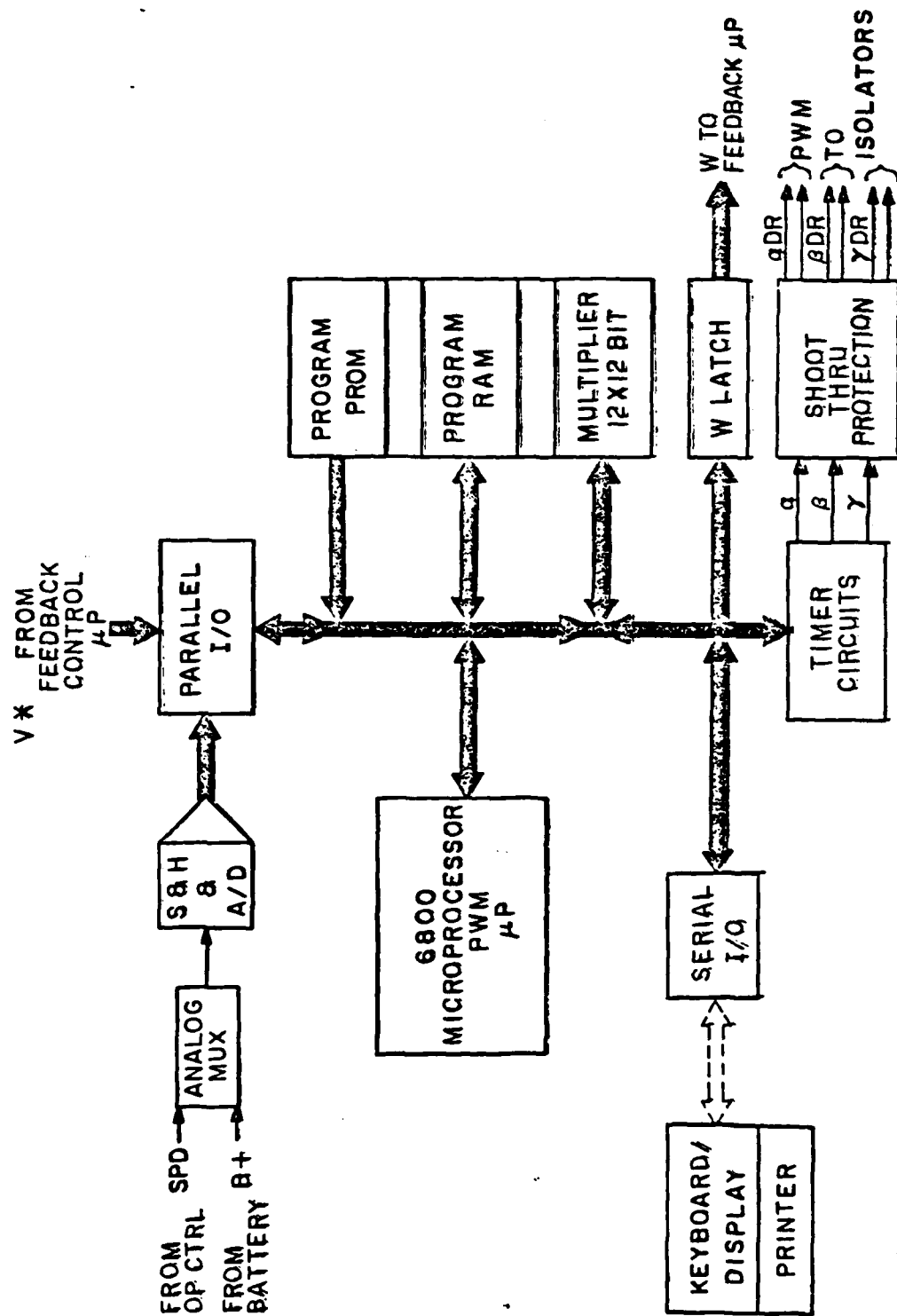
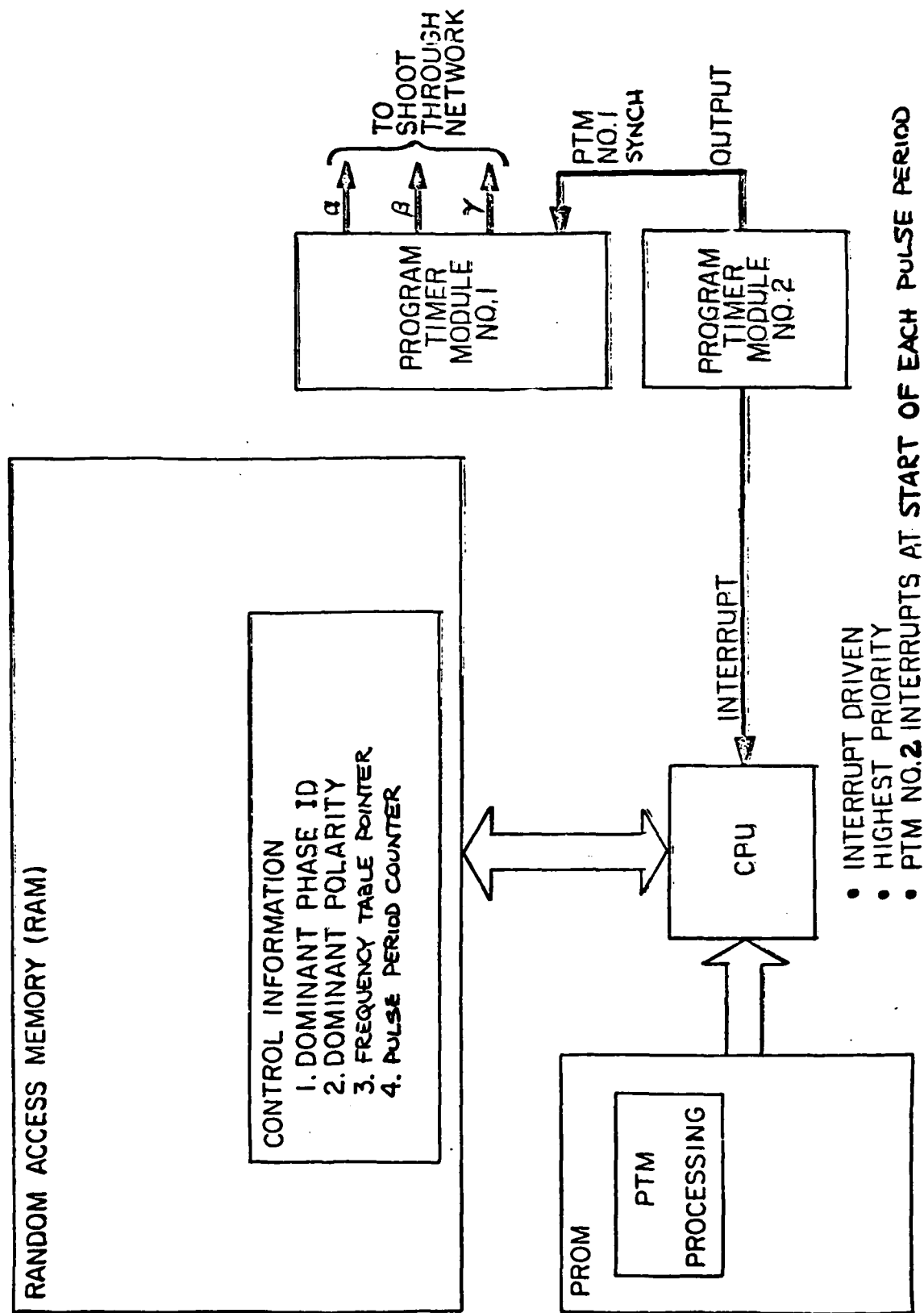


FIG. 3.3

TIMER CONTROLLER

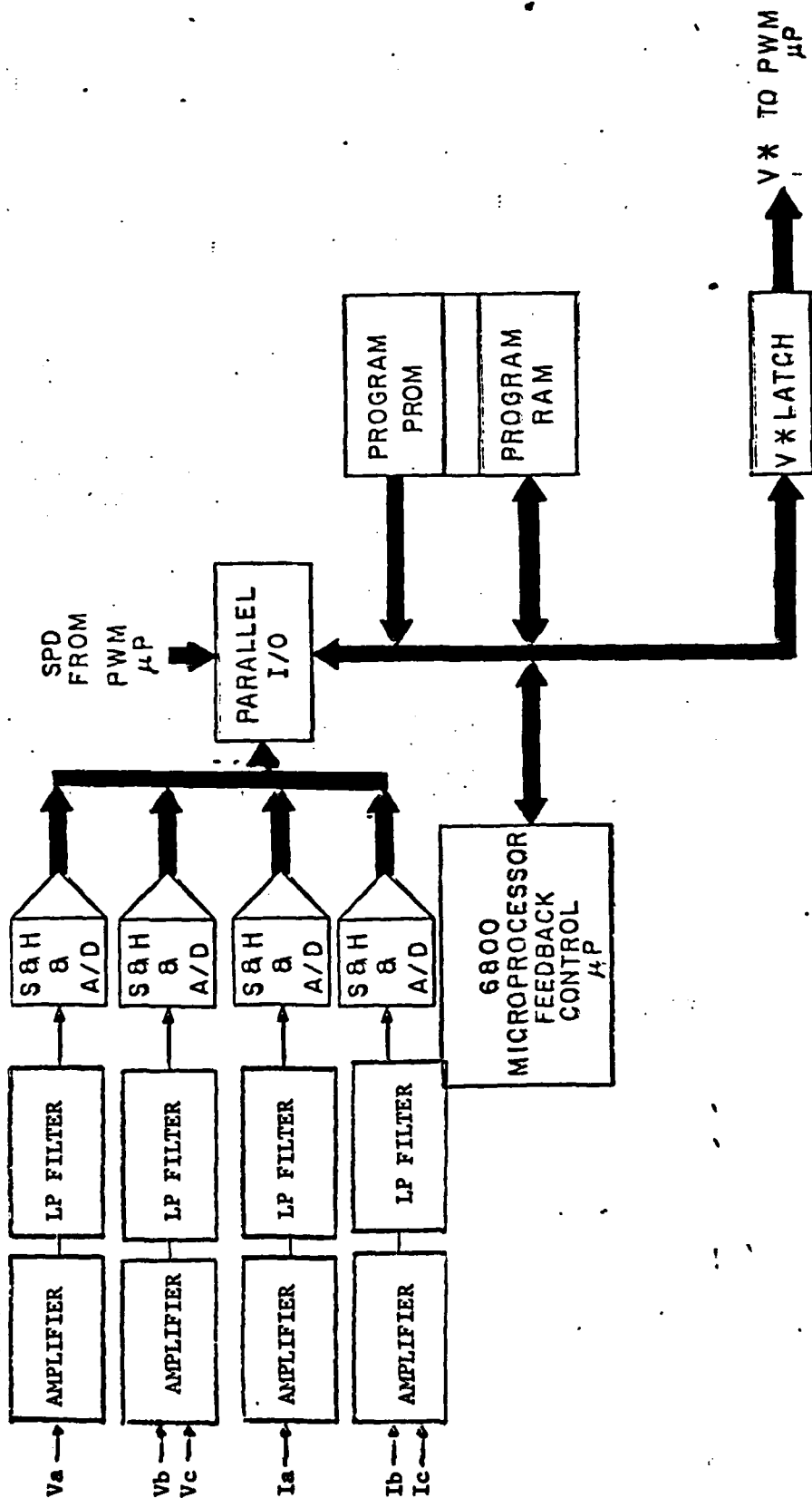


- INTERRUPT DRIVEN
- HIGHEST PRIORITY
- PTM NO. 2 INTERRUPTS AT START OF EACH PULSE PERIOD

FIG. 3.4
HARDWARE MEMORY MAP
PWM PROCESSOR

SIZE		ADDRESS	FUNCTION
2 BYTES		FFFE, FFFF	RESET VECTOR
1024 X 8	RAM	FFFD	MONITOR GLOBALS
		FC00	MCL GLOBALS
1024 X 8	RAM	FB00	
		F800	MONITOR GLOBALS
		F7FF	
2048 X 8	PROM		PMON9
		F000	
		FFFF	
2048 X 8	PROM		FREQUENCY TABLES
		E800	
		E7FF	
2048 X 8	PROM		PTIS
		E000	
		DFFF	
2048 X 8	PROM		MCL
		D800	
		D7FF	
1024 X 8	RAM		MONITOR GLOBALS
		D400	
		D3FF	
	NOT USED		
		7000	
4096 X 8	ACIA	6FFF	ACIA ADDRESSING
		6000	
4096 X 8	PIA	5FFF	PIA ADDRESSING
		5000	
4096 X 8	PTM1	4FFF	PTM1 ADDRESSING
		4000	
4096 X 8	PTM2	3FFF	PTM2 ADDRESSING
		3000	
4096 X 8	MULTIPLIER	2FFF	MULTIPLIER ADDRESSING
	'W' LATCH	2000	
		1FFF	FREQUENCY INFORMATION ADDRESSABLE LATCH
4096 X 8	NOT USED	1000	
		0FFF	
4096 X 8	NOT USED	0400	
		03FF	
1024 X 8	RAM	0000	MCL GLOBALS & TEMPORARY STORAGE

FIG. 3.5
FEEDBACK CONTROL MICROPROCESSOR BLOCK DIAGRAM

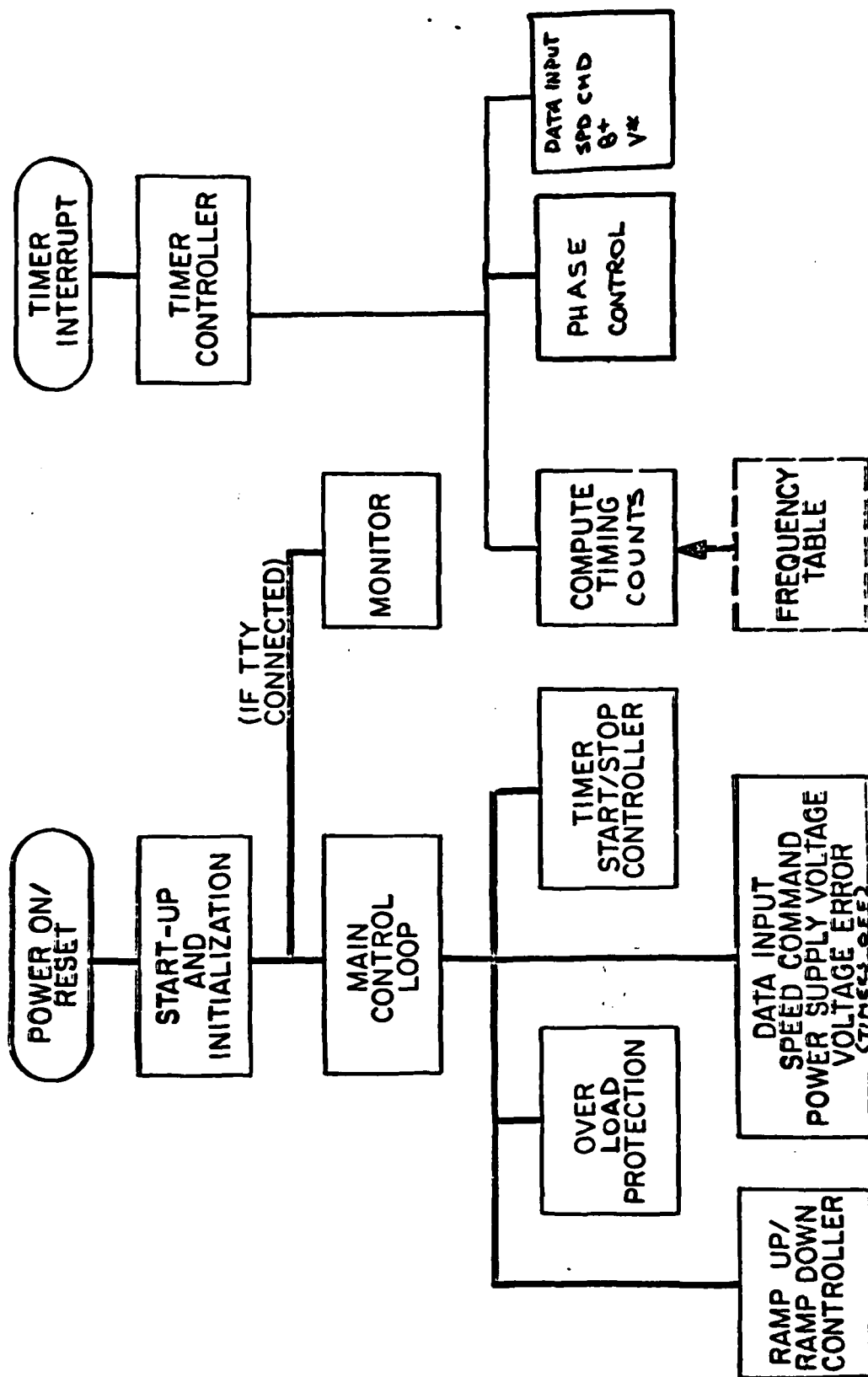


HARDWARE MEMORY MAP FEEDBACK PROCESSOR



[illegible]

FIG. 3.8
PWM PROCESSOR SOFTWARE



- TWO CONCURRENT TASKS
- INTERFACE THROUGH CONTROL PACKET

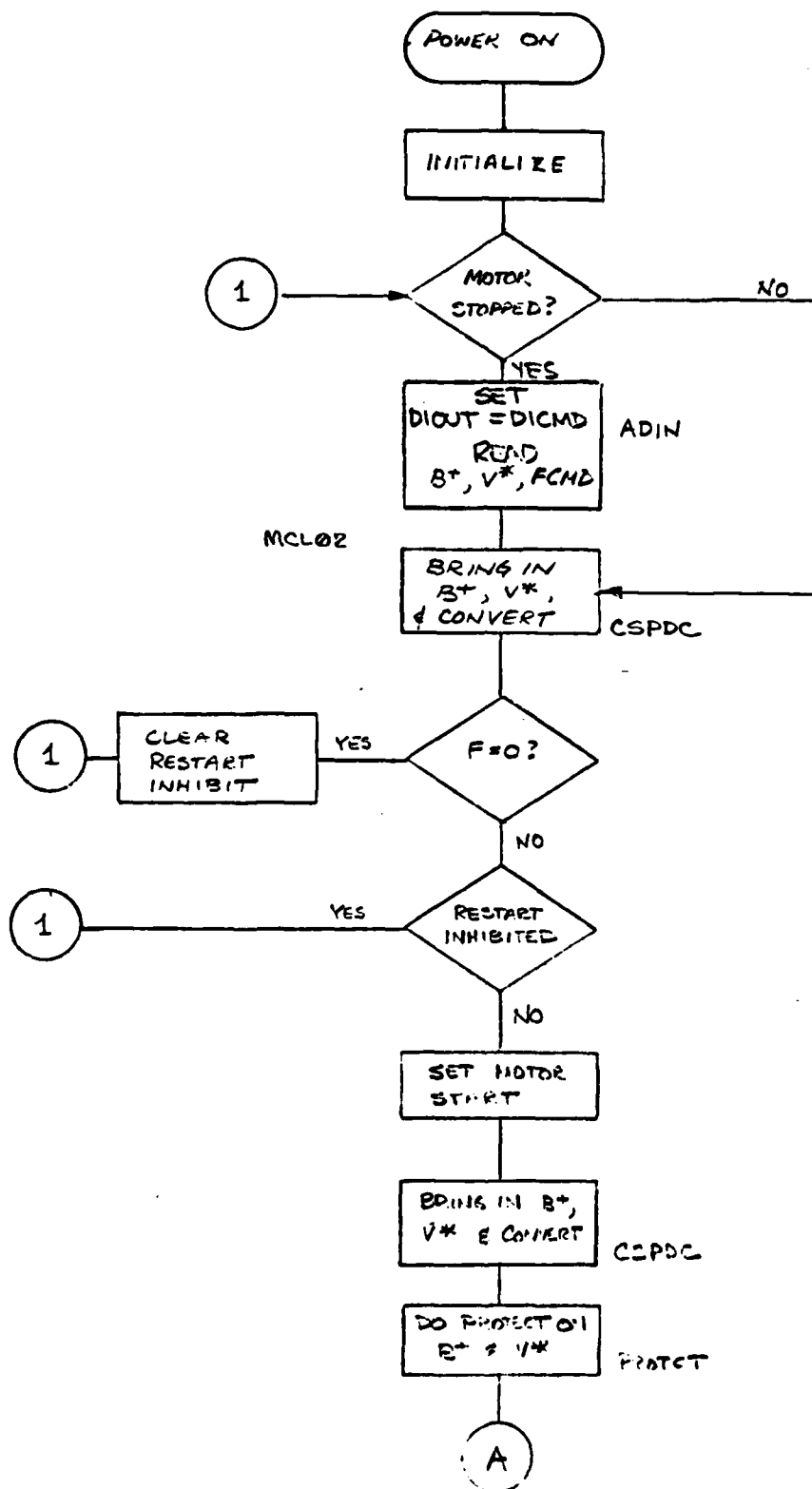


FIG 3.9 MAIN CONTROL LOOP ALGORITHM (SHEET 1)

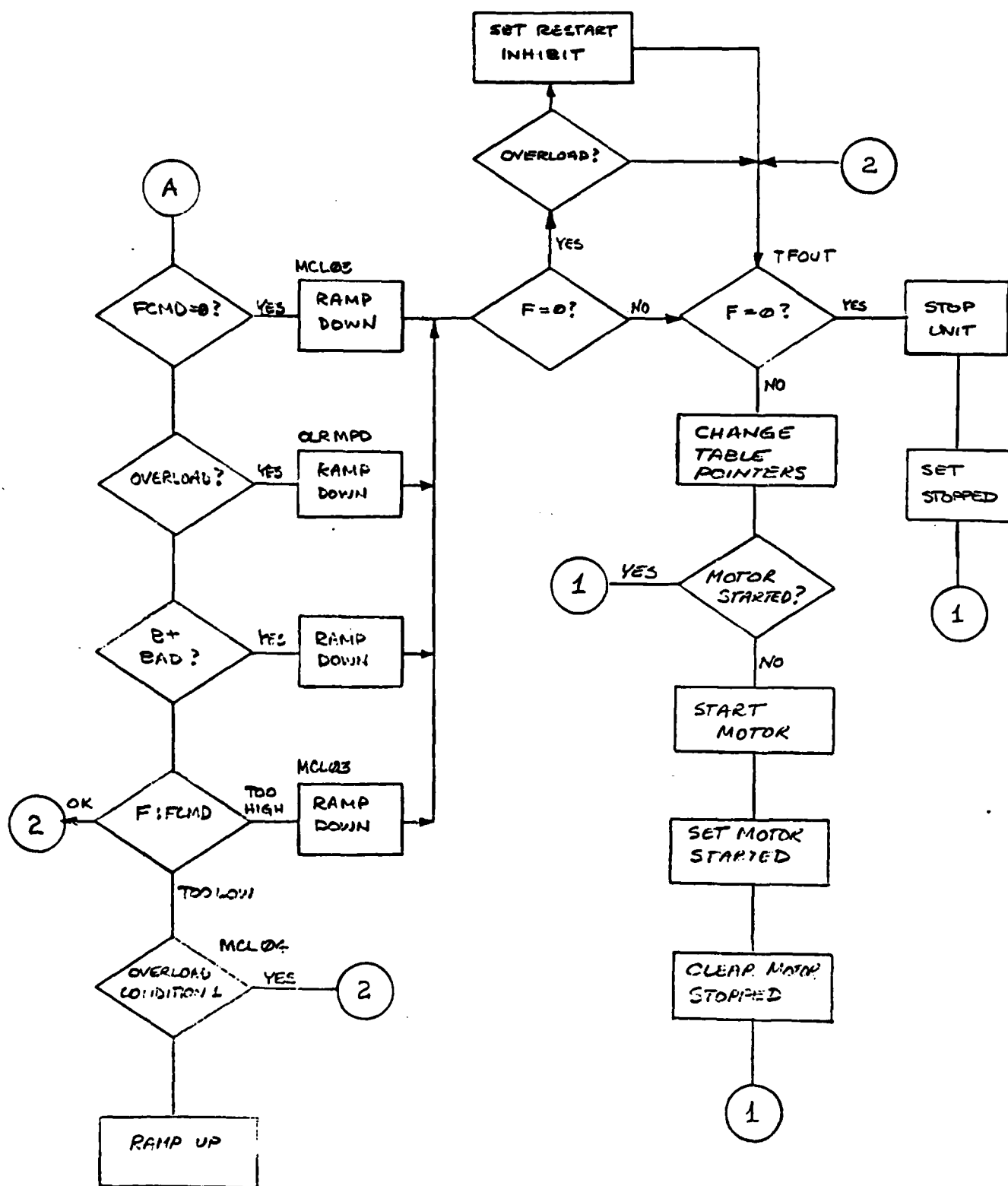


FIG 3.9

MAIN CONTROL LOOP ALGORITHM (SHEET 2)

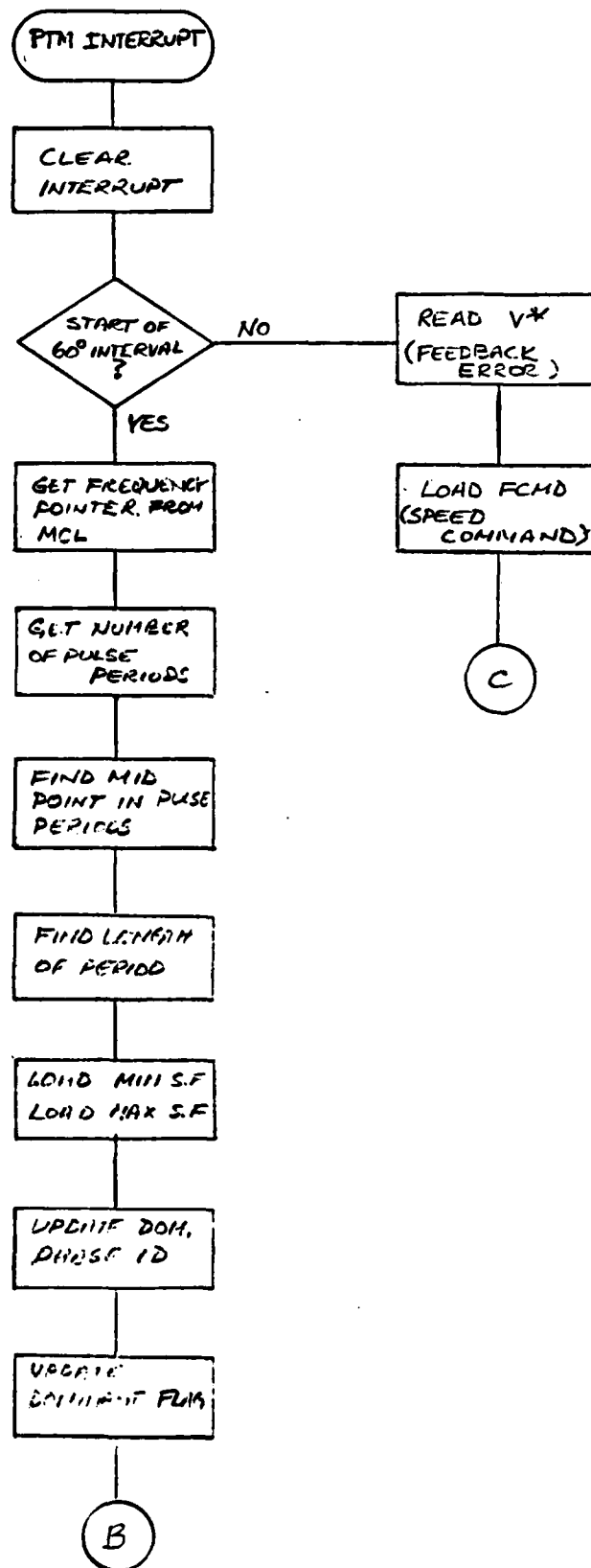
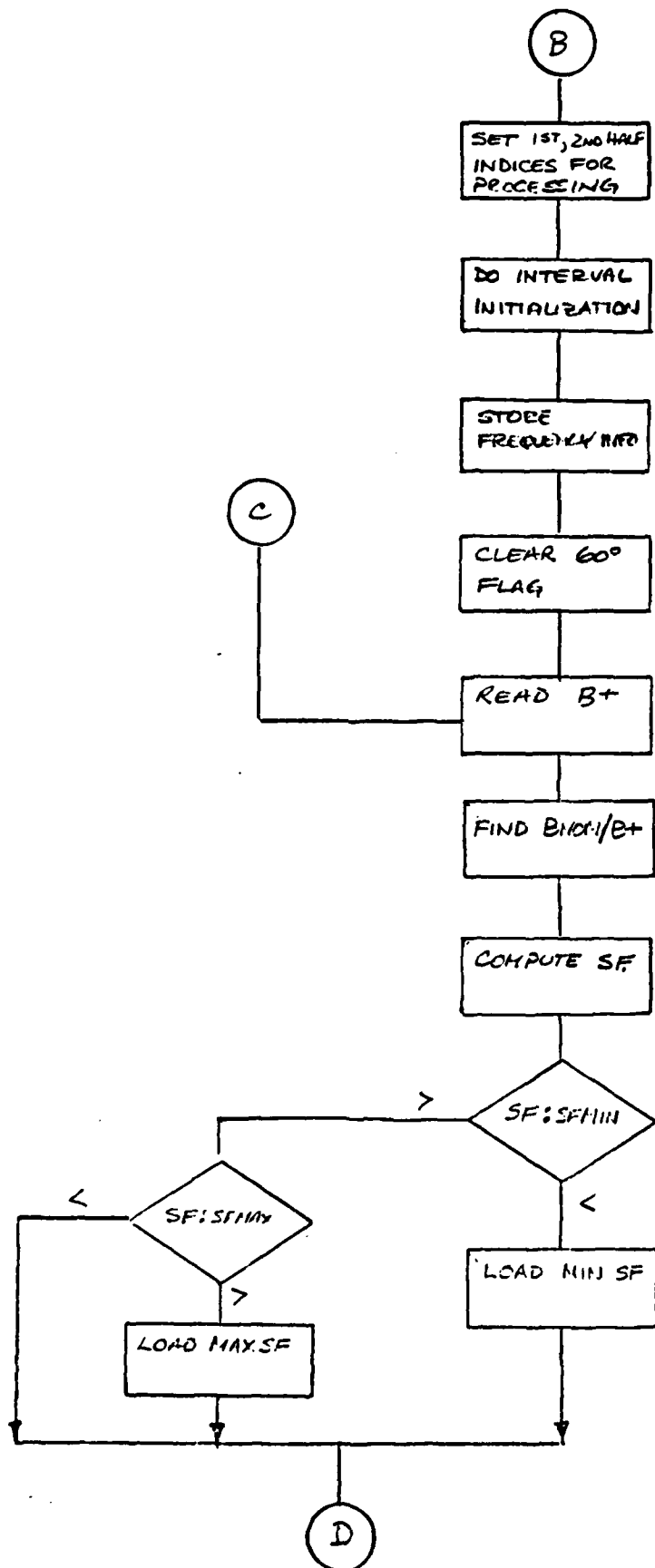
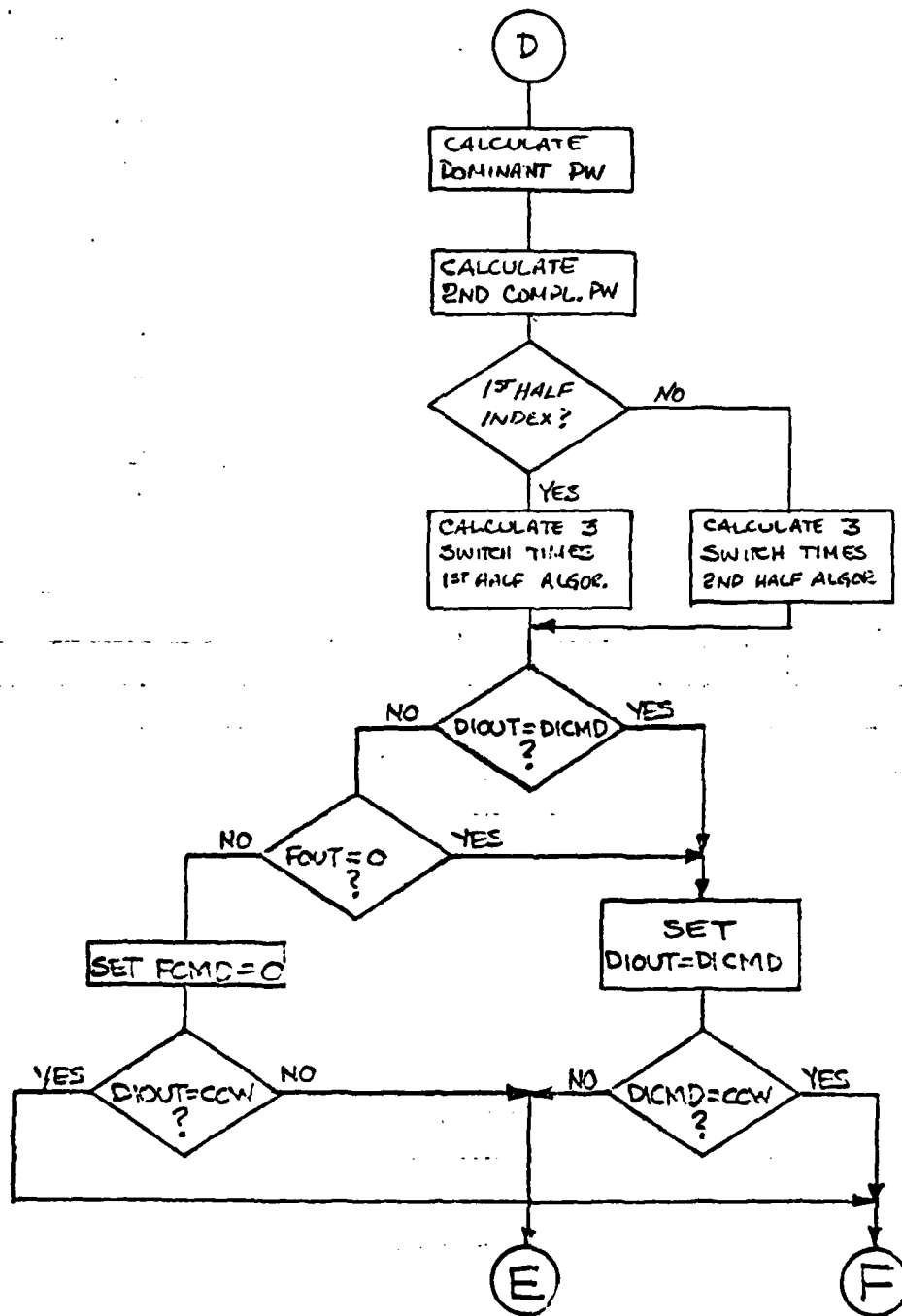
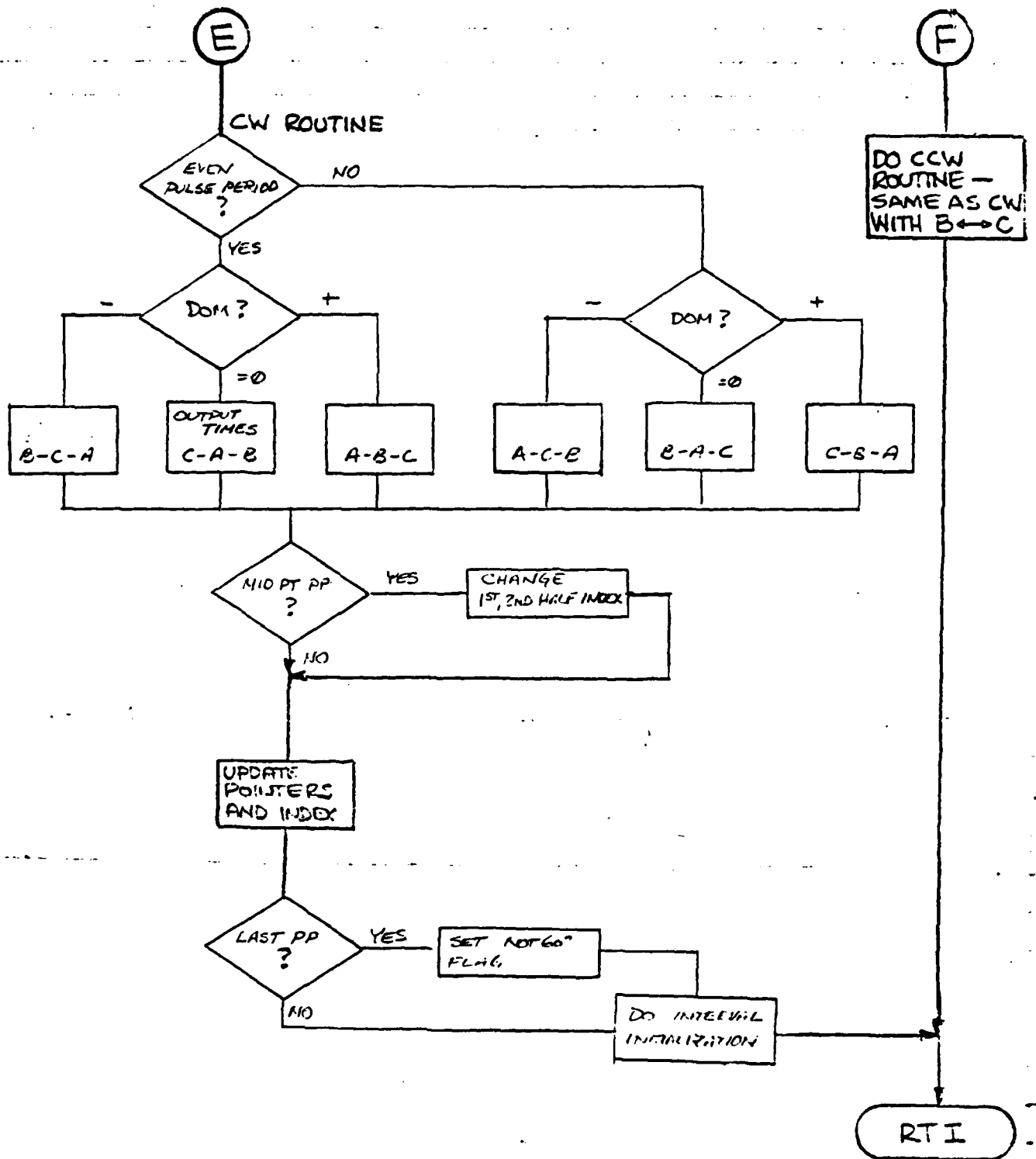


FIG 3.10 PTM INTERRUPT ROUTINE (SHEET 2)







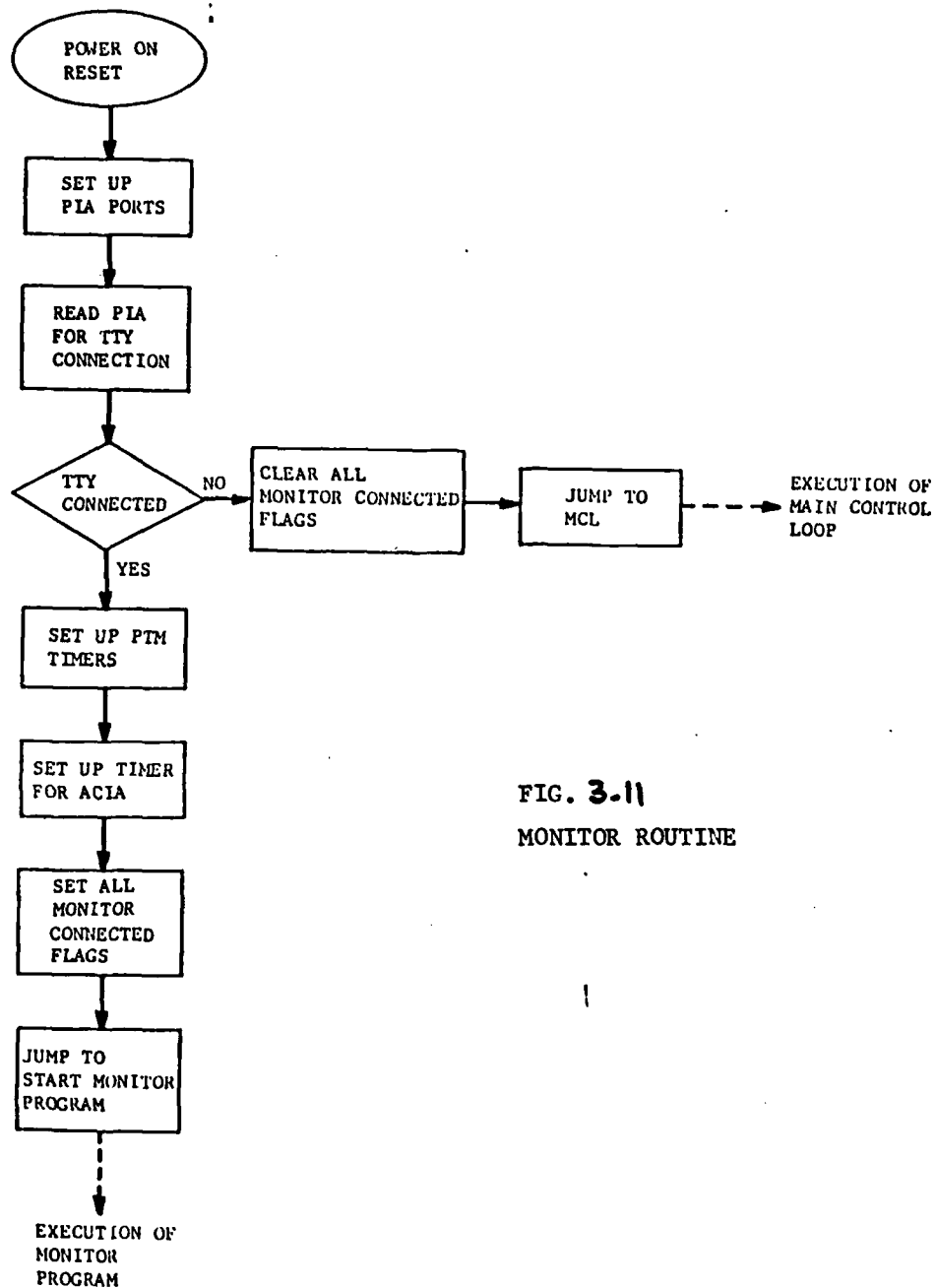


FIG. 3-11
MONITOR ROUTINE

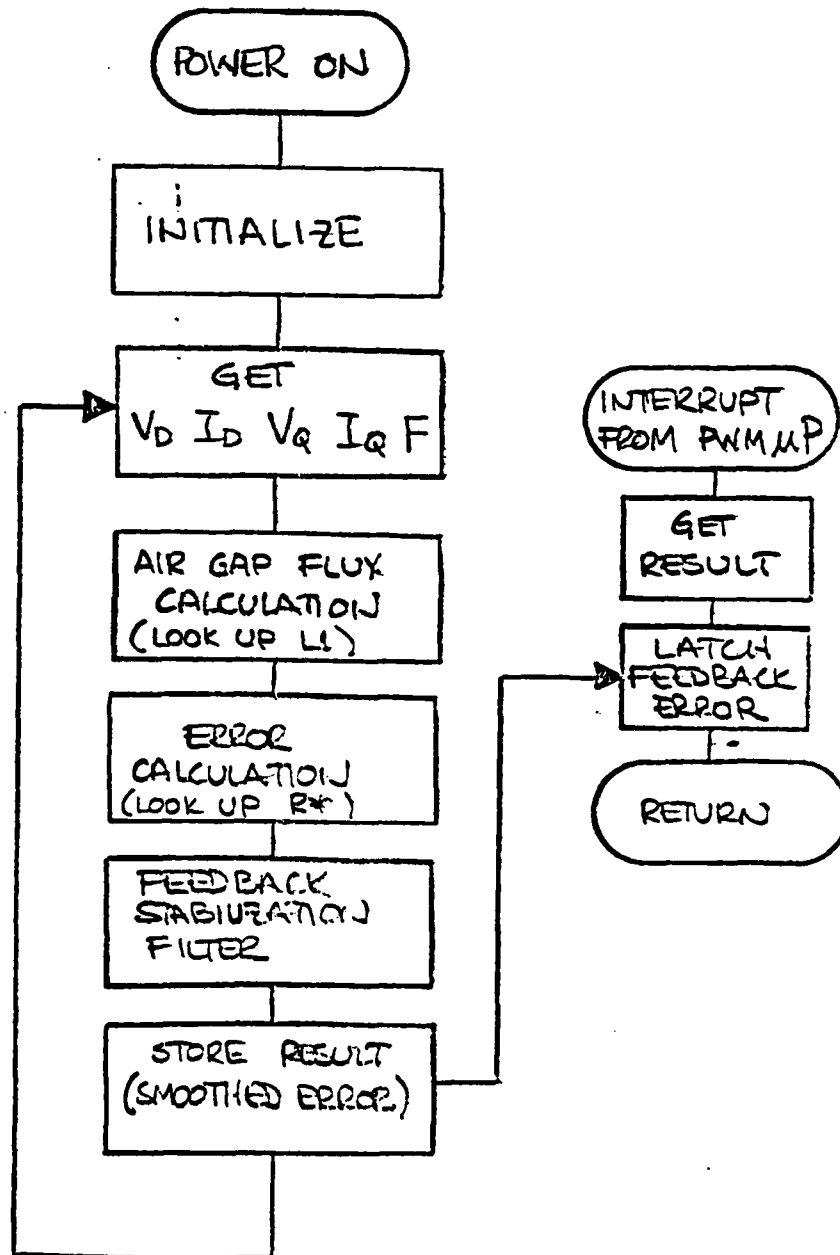


FIG. 3.12
FEEDBACK MICROPROCESSOR
ALGORITHM

4.0 Test Measurements

4.1 Flux and Control Loops

To establish an efficient operating curve, the motor was excited at frequencies from six to sixty hertz while varying the excitation voltage and observing the input current. A plot of the points of voltage required for minimum current was then made against frequency. As expected, this curve was a straight line ($v/f = \text{constant}$). The table used by the PWM processor to determine the open-loop operating point was generated using the linear v/f information.

The system operated properly under open loop conditions, responding to speed and amplitude commands from the keyboard.

Next, phase voltages and currents were sensed and the D-Q variables successfully generated, converted, and used in the flux calculation. Figure 4.2D shows the direct (bottom) and quadrature (top) currents as they exist at the analog input to their A/D converters. The D-Q voltage waveforms are similar. As desired, the amplitudes of the two components were the same and their phase differed by 90 degrees. θ , the angle between V_D and I_D , was seen to vary with the load from 20 to 62 degrees. This range of phase angle is due to the choice of an operating point near minimum excitation current. Figure 4.1 schematically shows how the D-Q variables are generated in the system.

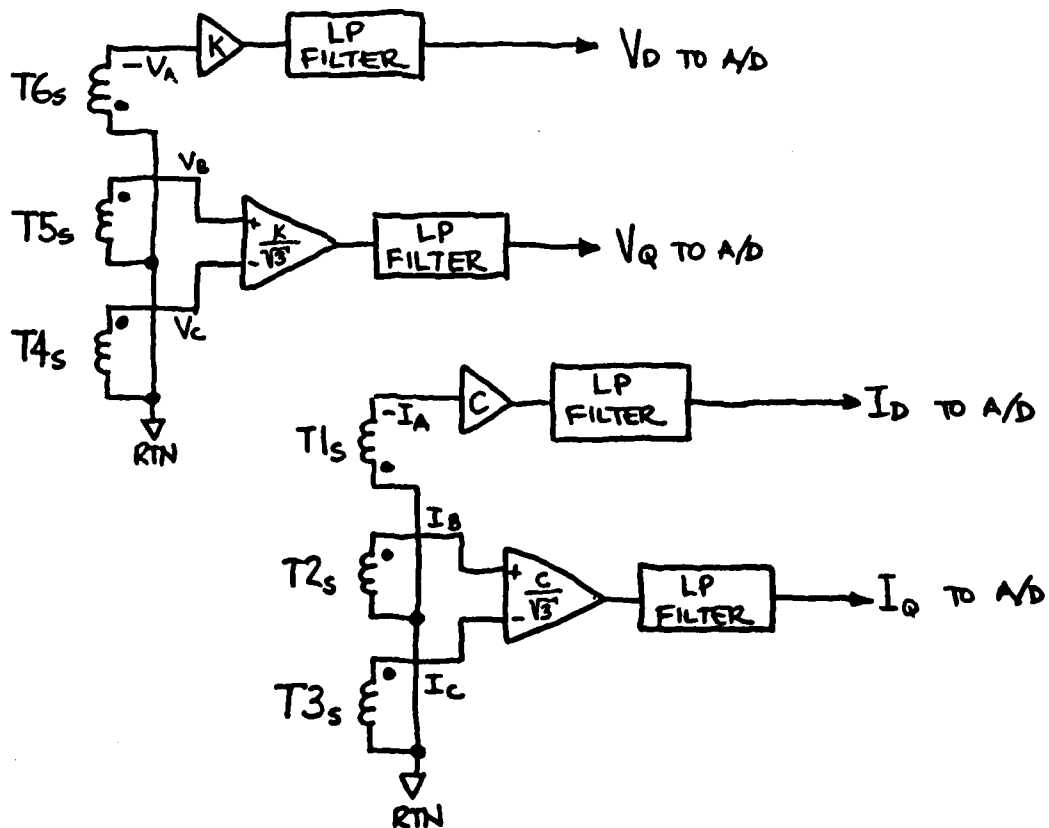
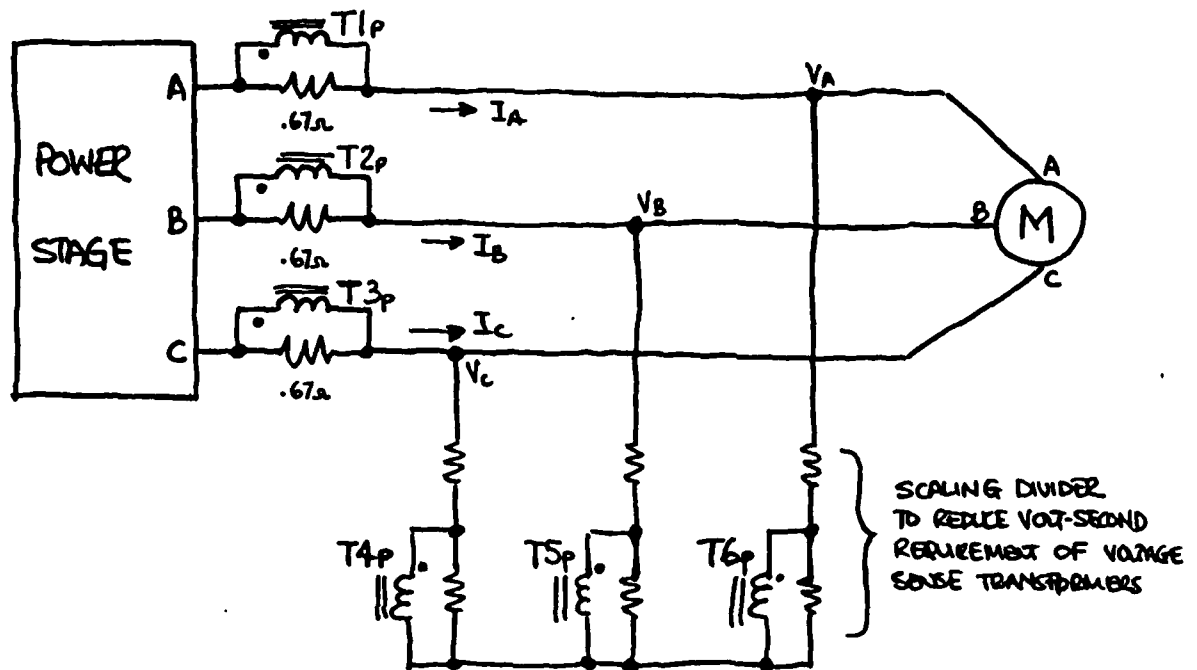
Four dedicated A/D converters digitize the analog D-Q variables. The A/D converters are sampled at a high rate compared to the output frequency, so their values may be considered as being taken simultaneously for purposes of the flux calculations.

An estimate of parameters of the test motor was made and simulation using an H.P. 9830 computer was performed to confirm that the parameters chosen would yield values of current and flux in keeping with the test motor size. These parameters were then used in the generation of the various tables required for flux calculation.

Operation in the open-loop mode yielded values of calculated flux consistent with the simulation. Table 4.1 below gives a set of measurements and calculations performed by the sense circuits and the feedback processor. A Hewlett-Packard 9611 was used to monitor operation of the microprocessor.

In closed loop operation, the calculated value of air gap flux, W_x , is subtracted from a fixed reference value, R^* , and the resulting error signal is digitally filtered and transmitted to the PWM processor. The filter is used to smooth the result of the W_x calculation and to provide a controlled pole in the loop response of the system. The filtered error signal is used by the PWM processor to modify the output amplitude so as to reduce the error and keep the calculated value of W_x constant. In open loop operation, W_x values were observed by setting the reference, R^* , to zero and observing the smoothed error.

FIG.4.1 D-Q VARIABLE GENERATION



5-2 B1
DB2

WORK SHEET
WESTINGHOUSE FORM 8295 E

INSTANTANEOUS READINGS	Iq MEAS		Vd MEAS		Vq MEAS		Id MEAS		Vd To		Vq To		Wx "PAW"	"PAW" MEANS THAT I²XL NOT CORRELATED
	HEX	DEC	H	D	H	D	H	D	CAIC	CAIC	CAIC	CAIC		
①	FC	-15A	40	64V	0D	13V	09	.45A	-9.75	585	15.6		APPROX.	
②	04	.29	29	41	38	56	0A	.50	11.89	28	16.11			
③	09	.45	E8	-24	33	51	02	.10	-10.8	5.1	15.9			
④	02	.10	32	50	32	50	0A	.50	5	25	20		CONSTANT AVG = 17.8	
⑤	FE	-.10	3E	62	1D	29	0A	.50	-6.2	14.5	20.7			
⑥	FA	-.30	3F	63	00	0	0A	.50	-18.9	0	18.9			
	CALCULATE $\sqrt{I_0^2 + I_q^2}$				FOR ABOVE CASES: ① ② ③ ④ ⑤ ⑥				.474	x 142 (XL) = 6.36 → 8.96				AVG = 10.6
									.578		8.09	8.02		
									.461		6.45	9.45		
									.510		7.14	12.86		
									.510		7.14	13.56		
[ADR 10 → 13] P2-P4 (DEC) P2-P4 (DEC)	0165 11.15	0189 12.28	0175 11.65						.583		8.16	10.74		
E* AT MPA/B INTERFACE (H) E* BEFORE ↓														
Vix AT CURT OF ADDR2 POSING = ADDR 32 ADDR 33 (H) 016X [AVG]														
P2-P4 5164	...	64 → 84 → 5.25 → 2.25												
0165	→	0001 0110 0101	→ 11.15											
0189	→	0001 1000 1001	→ 12.28											
0175	→	0001 0111 0101	→ 11.65											

Since the measurements of W_x were in accord with expectations, a reference level close to the value of W_x at no load was inserted (instead of zero) and an attempt at closed loop operation was made. The motor/controller system was successfully exercised in the closed loop mode. Under control of flux loop calculations made by the feedback processor, the controller increased the output voltage amplitude to compensate for losses when the motor was loaded and decreased the output when the load was reduced, keeping the calculated value of flux constant. The "feel" of loading the motor by hand was dramatically stiffer in this mode. Stepping up to and down from the selected frequency under closed loop conditions also functioned properly.

4.2 Spectrum of PWM Output Motor Load

A Federal Scientific model UA-500 (ubiquitous) spectrum analyzer was used to examine the voltage and current frequency spectrum of the motor controller power stage output while driving the three phase test motor. Figures 4.3 through 4.5 are photographs of the output waveforms and spectra, taken under various conditions.

Notice that the amplitudes of the unwanted frequencies in the current spectrum stay constant as the fundamental rises with increased load. Observe also that very little unwanted output energy is evident below 1 KHz under any operating circumstance. The ratio of the amplitude of output current at the fundamental frequency to that of the next highest component, about 1.3 KHz, with the motor loaded to one-fifth horsepower is in excess of 33db.

Figures 4.2 through 4.5 also show output voltages and currents at various output frequencies. The current ripple stays approximately constant in amplitude over the output frequency range due to the inverse relation between the PWM carrier frequency and the output frequency. Comparison of the unloaded currents with the corresponding loaded output currents, as in figures 4.4B and D and 4.5A and C, shows that the current ripple remains constant as the output current increases. This phenomenon is responsible for the spectral purity observed in the loaded output current photographs.

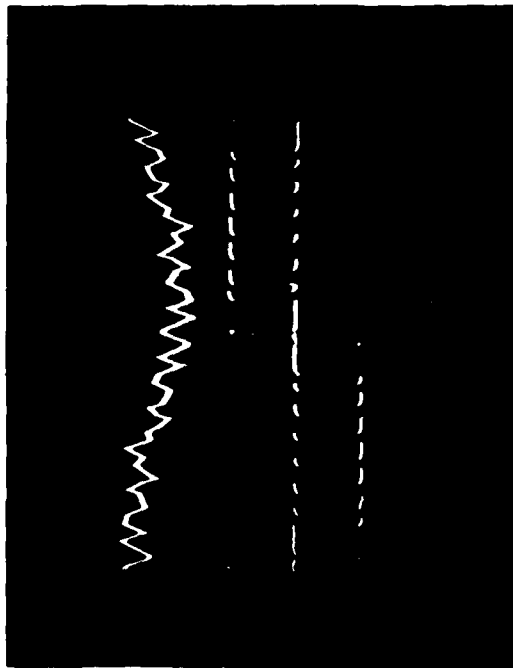
4.3 Reversing

A routine to step down to zero speed and reverse the output direction in response to the setting of a direction flag from the keyboard has been incorporated into the software and successfully tested. It remains to make some minor modifications to the flux loop D-Q inputs to properly calculate the W_x value in the reverse mode. This modification, as well as the reading of speed and direction inputs from an analog source (potentiometer), will be accomplished next.

4.4 Overload Protection

The software needed to sense overload at a preset level above the chosen operating point, to ramp down to an acceptable level, and to ramp back up when the overload is relieved has been incorporated in the control program.

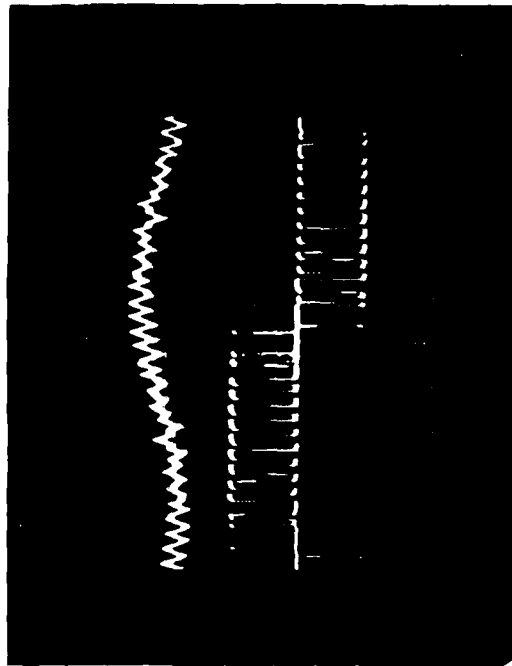
This feature will not be tested until the proposed Phase III of the program when the power stage is added and a total drive system tested.



OUTPUT
CURRENT
1A/DIV

L-L OUTPUT
VOLTAGE
100V/DIV

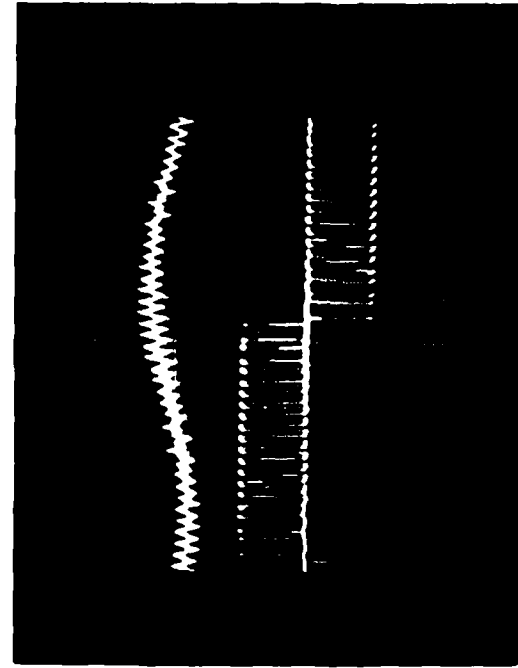
A. 60Hz, NL



OUTPUT
CURRENT
1A/DIV

L-L OUTPUT
VOLTAGE
100V/DIV

C. 55Hz, NL



OUTPUT
CURRENT
1A/DIV

L-L OUTPUT
VOLTAGE
100V/DIV

B. 41Hz, NL



I_Q
(FILTERED)
.2V/DIV

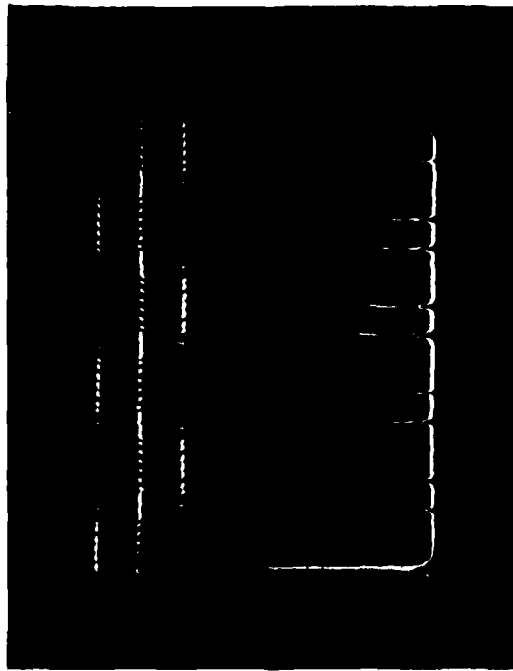
I_D
(FILTERED)
.2V/DIV

TIME,
2ms/DIV

SCALE
FACTOR
[1V=1.25A]

D. 60Hz, NL

FIGURE 4.2

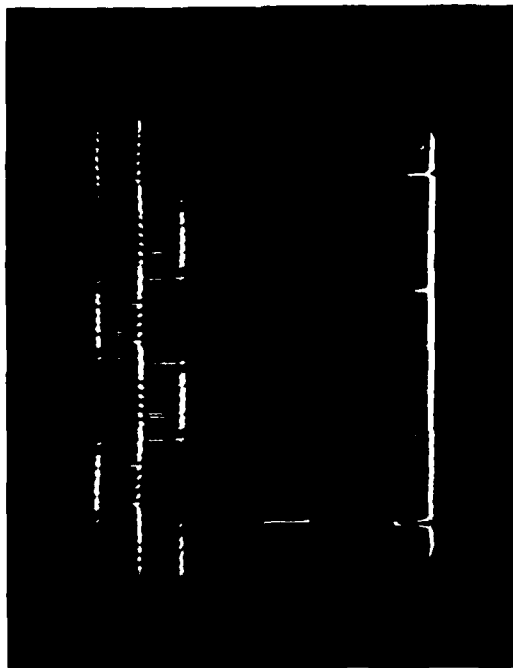


OUTPUT
PWM
VOLTAGE

OUTPUT
PWM
VOLTAGE

VOLTAGE
SPECTRUM
200Hz/DIV
REL AMP
= 1/DIV

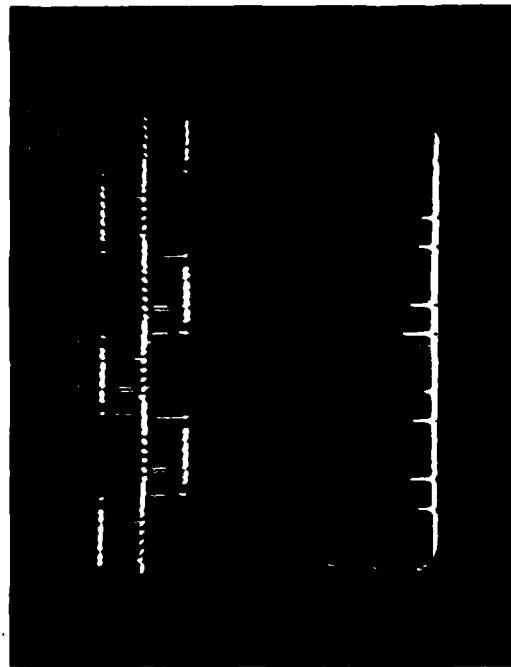
VOLTAGE
SPECTRUM
50Hz/DIV
REL AMP
= 1/DIV



A. 60Hz OUTPUT, NO LOAD ON MOTOR (NL)
18 PULSES/OUTPUT CYCLE

B. 60Hz OUTPUT, NL

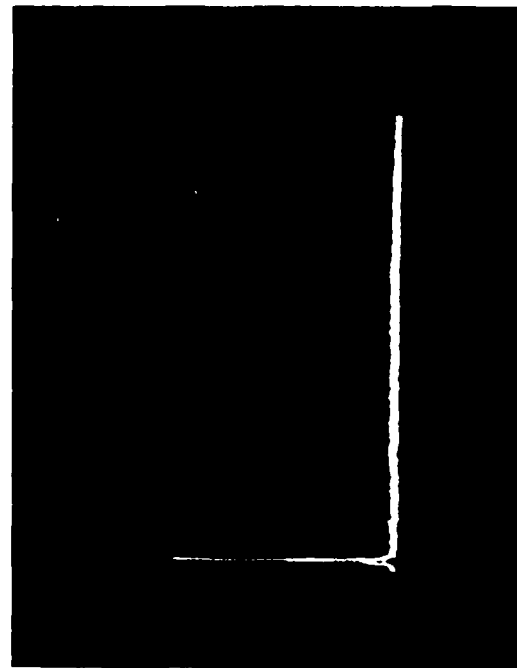
C. 60Hz, NL



OUTPUT
PWM
VOLTAGE

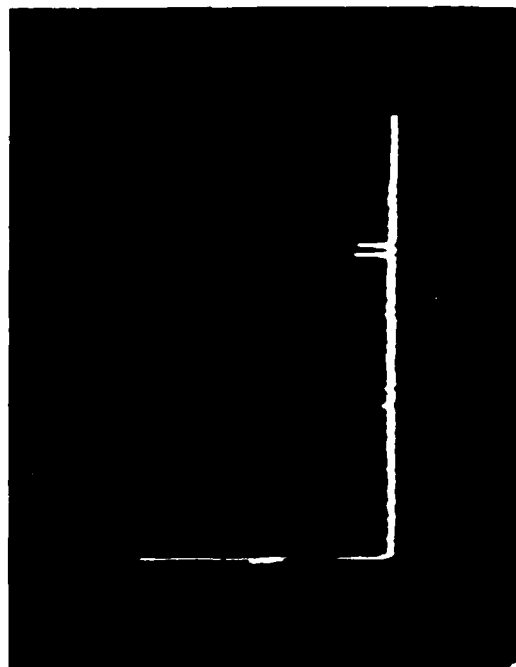
CURRENT
SPECTRUM
200Hz/DIV
REL AMP
= 1/DIV

FIGURE 4,3



CURRENT
SPECTRUM
50Hz/DIV
REL AMP
= 1/DIV

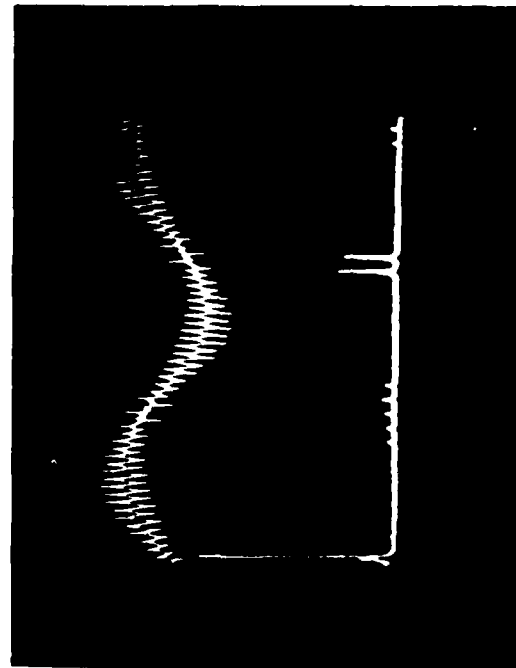
CURRENT
SPECTRUM
200Hz/CN
REL AMP
= 1/DIV



A. 10Hz, NL

C. 20Hz, NL

B. 30Hz, NL

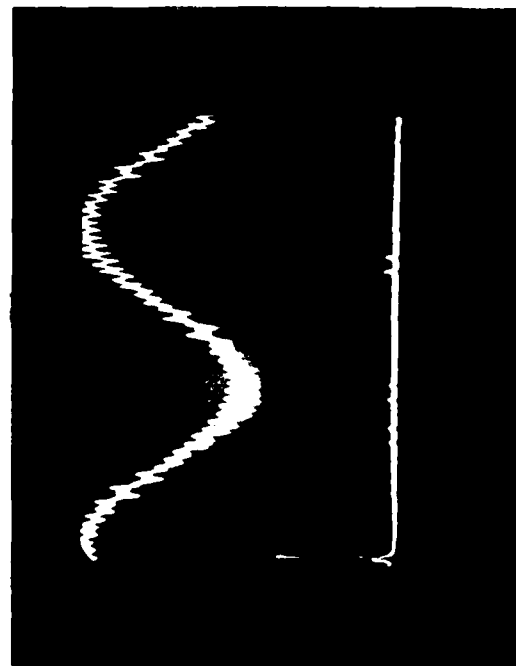


OUTPUT
CURRENT
.5A/DIV

CURRENT
SPECTRUM
200Hz/DIV
REL AMP
= 1/DIV

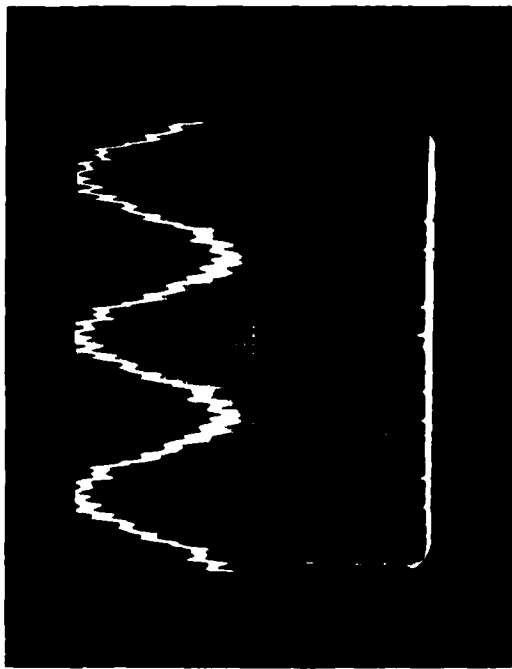
OUTPUT
CURRENT
1A/DIV

CURRENT
SPECTRUM
200Hz/DIV
REL AMP
= 3.3 /DIV



D. 30Hz, 1/5 HP LOAD

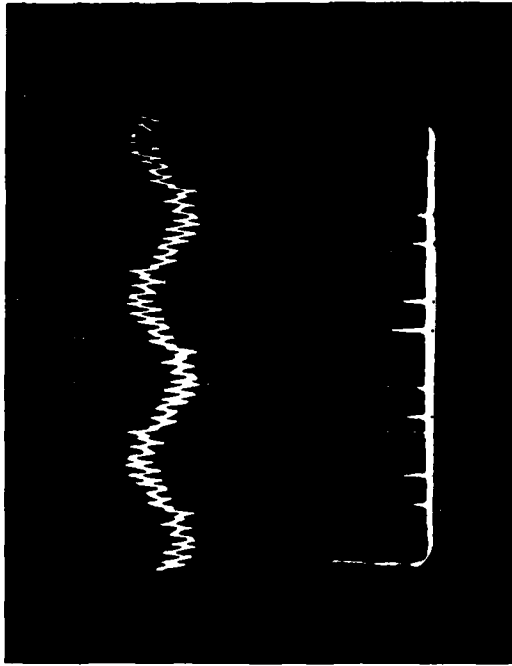
FIGURE 4.4



OUTPUT
CURRENT
1A/DIV

CURRENT
SPECTRUM
20kHz/DIV
REL AMP
= 3.3/DIV

A. 60Hz, 1/5 HP LOAD

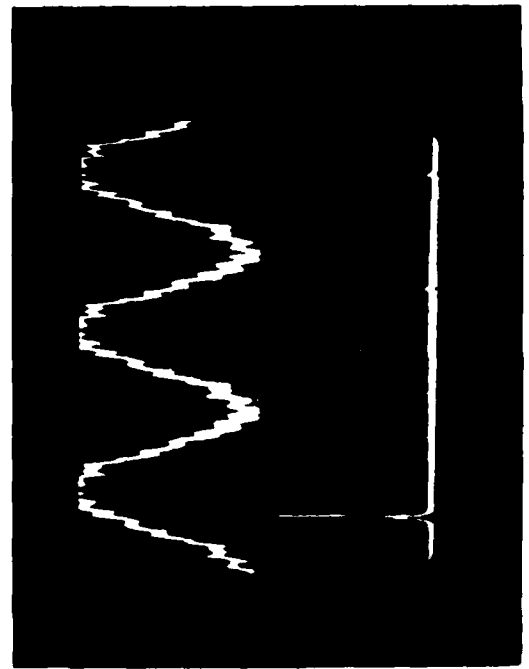


OUTPUT
CURRENT
1A/DIV

CURRENT
SPECTRUM
20kHz/DIV
REL AMP
= 1/DIV

C. 60Hz, NL

B. 60Hz, 1/5 HP LOAD



OUTPUT
CURRENT
1A/DIV

CURRENT
SPECTRUM
50kHz/DIV
REL AMP
= 3.3/DIV

5. Conclusions & Recommendations

The Phase II development of the control circuits implement the theoretical requirements of the modulation and feedback mechanisms and provide a practical control circuits for a three phase motor drive. The feedback control produces efficient off-load operation as evidenced by the wide voltage swing between no load and full load on the small test motor. The feedback processor would appear to offer a means of sensing and control of both variable frequency, variable voltage AC drives and fixed frequency variable voltage drives (e.g. phase control of fan motors).

Performance to date has been very satisfactory and development of a full power system is recommended. Use of newer 16 bit microprocessors could permit extension of the control circuits to 400 Hz operation. Development of 400 Hz control and power stages is also recommended.

SEQ	LOC	OBJ	SOURCE	NAM	PROTO BOARD MONITOR
0001					
0002			* PMON4		
0003					
0004					
0005		0001	LSYMB	1	
0006		6000	HCIAO	EQU	\$6000 ;ACIA CONTROL REGISTER
0007		6001	HCIAO	EQU	\$6001 ;ACIA DATA REG
0008		6000	HCIAO	EQU	\$6000 ;ACIA STATUS REG
0009		0020	BLANK	EQU	\$20 ;BLANK CHAR
0010		0000	CR	EQU	\$00 ;CARRIAGE RET CHAR
0011		001B	ESC	EQU	\$1B ;ABORT CHAR
0012		0004	EOT	EQU	\$04 ;END OF MSG TO BE PRINTED
0013		FFFF	LAST	EQU	\$FFFF ;HIGHEST ROM ADDRESS
0014		000A	LF	EQU	\$0A ;LINE FEED
0015		007F	RUBOUT	EQU	\$7F ;RUBOUT CHAR
0016			*		
0017			*GLOBAL REF		
0018			* THESE LOCALS ARE NOT PROTECTED IF USER		
0019			* WRITES TO THEM IN USING MONITOR		
0020			*		
0021		0051	MONFG	EQU	\$51 ;MONITOR CONNECTED FLAG(=1)
0022		00E6	NORFG	EQU	\$E6 ;NORMAL OP FLAG =1MON,CONNECT
0023		00E7	VSFLG	EQU	\$E7 ;VSTAR TEST FLAG=1(USE MONITOR)
0024			* EXTERNALS FOR RSRSR AND FROM BURNER (NOT IMPL.)		
0025			*		
0026			* DEF MOMENT, GETRNG, NXTADR, PKISTS, RNGERR, PBADR		
0027			* DEF PORLF, OUTCH, PSPACE, SETMEM, ABORT		
0028			* DEF PROMAD, ADR, ADDL, ADDH, COUNT, MONITR		
0029			* DEF RSRSR, BURN, MOVE, READ, VFY, PINIT		
0030			*		
0031			* RSRSR ROUTINE DEFINITIONS		
0032			*		
0033			*		
0034			*		
0035			*		
0036			* MONITOR RAM		
0037			*		
0038		FF90	ORG	\$FFFE-110	**** CHANGE IF RAM USAGE CHAN
0039		FF90	BASE	EQU	* ;BASE ADDR USED WITH INDEX OPS
0040		FF8F	BOS	EQU	*-1 ;BOTTOM OF MONITOR STACK
0041			*		
0042	FF90		BUF	RMB	72 ;LINE OF TTY INPUT
0043			*		
0044		FFD8	PROMAD	EQU	* ;ADDRESS IN PROM
0045	FFD8		OFFSET	RMB	2 ;OFFSET FOR LOADER/PUNCH
0046	FFDA		ADR	RMB	2 ;PARAM. ENTERED BY USER
0047	FFDC		ADDL	RMB	2
0048	FFDE		ADDH	RMB	2
0049	FFE0		BUFPTR	RMB	2 ;POINTER TO LAST CHAR SCANNED
0050	FFE2		RECTYP	RMB	1 ;TAPE RECORD TYPE
0051	FFE3		COUNT	RMB	1 ;COUNT FIELD FROM TAPE
0052	FFE4		CKSM	RMB	1 ;CALCULATED CKSM
0053	FFE5		SHVESF	RMB	2 ;TEMP STORAGE FOR S REG
0054	FFE7		SHVEN	RMB	2 ;TEMP STORAGE FOR X REG
0055	FFE9		ECHO	RMB	1 ;1=ECHO TTY, 0=NO ECHO

SEQ	LOC	OBJ	SOURCE
0056	FFE8		TCCOUNT RMB 1 ; TEMP LOC FOR COUNT
0057			* USER REGISTERS
0058	FFEB		CREG RMB 1
0059	FFEC		BREG RMB 1
0060	FFED		HREG RMB 1
0061	FFEE		XREG RMB 2
0062	FFF0		PREG RMB 2
0063	FFF2		SREG RMB 2
0064			*
0065	FFF4		USWI RMB 2 ; USER SWI VECTOR (NOT IMPL.)
0066	FFF6		ACIAI RMB 2 ; INDIRECT POINTER TO ACIA FOR RES. SR
0067	FFF8		IRQVEC RMB 2 ; INTERRUPT REQUEST VECTOR
0068	FFFA		SWIVEC RMB 2 ; SOFTWARE INTERRUPT VECTOR
0069	FFFC		NMIVEC RMB 2 ; NON-MASKABLE INTERRUPT VECTOR
0070			*
0071			*
0072			*** MONITOR ENTRY VECTOR ***
0073			*
0074			* RESTART INTERRUPT HANDLER
0075			* INTERRUPT BREAK HANDLER
0076			*
0077			***
0078		F000	ORG \$F000
0079			* RESET VECTOR ENTRY POINT
0080			* JUMP IMMEDIATELY TO INITIAL SEQUENCE
0081	F000	7E F510	JMP PWRI ; JUMP TO PWR ON INIT SEQ
0082			*
0083		F018	ORG \$F018
0084		F018	START EQU * ; RESET INTERRUPT HANDLER
0085	F018	20 05	BRH START1
0086		F01H	BREAH EQU * ; BREAK ON INTERRUPT ROUTINE
0087	F01H	7E F0CF	JMP BREAK1
0088	F010	60 00	ACIAH FDB ACIAC ; POINTER TO ACIA
0089			*
0090		F01F	START1 EQU * ;
0091	F01F	36	PSH A ; SAVE AREG IF STACK EXISTS
0092	F020	07	TPA ; SAVE CONDITION CODES
0093	F021	B7 FFEB	STA A CREG
0094	F024	32	PUL A
0095	F025	B7 FFED	STA A AREG ; SAVE CURRENT VALUE OF REGS
0096	F028	F7 FFEC	STA B BREG
0097	F02B	FF FFEE	STX XREG ; SAVE X
0098	F02E	BF FFF2	STS SREG ; SAVE SP
0099	F031	8E FF8F	LDS #BOS ; INIT. SREG TO MON. STACK
0100	F034	CE F01H	LDX #BREAH ; BREAKPOINT ROUTINE
0101	F037	FF FFFC	STX NMIVEC ; STORE IN INTERRUPT VECTORS
0102	F03H	FF FFF8	STX IRQVEC
0103	F03D	CE F029	LDX #SWI30
0104	F040	FF FFF4	STX USWI
0105	F043	CE F006	LDX #SWIHAN ; SOFTWARE INTERRUPT HANDLER
0106	F046	FF FFFA	STX SWIVEC
0107	F049	CE F010	LDX #ACIAH ; SET UP ACIA PTR
0108	F04C	FF FFF6	STX ACIAI
0109	F04F	86 03	LDA H #3 ; RESET ACIA
0110	F051	B7 6000	STH A ACIAC

SEQ	LOC	OBJ	SOURCE
0111			*
0112			* ACIA SET-UP :CLK/16.7 BITS+ODD PARITY+1 STOP BIT
0113			*
0114	F054	86 00	LDA H #00 ;SET ACIA CNTRL REG
0115	F056	87 6000	STA H ACIAC
0116			* PRINT CR,LF, & RETURN TO MONITOR
0117			*
0118		F059	MUNENT EQU *
0119		F059	MUNEN1 EQU *
0120	F059	8D F434	JSR PINIT
0121	F05C	8D F322	JSR PURLF
0122			*
0123			* MONITOR ENTRY POINT
0124			*
0125			*
0126		F05F	MUNITR EQU *
0127	F05F	8E FF8F	LDS #B05 ;INIT MON. STACK
0128	F062	8D F389	JSR RDRUFF
0129	F065	86 6001	LDA H ACIAD ;DUMP TTY INPUT DATA
0130	F068	86 3E	LDA H #10 ;PROMPT USER
0131	F06H	8D F2FB	JSR OUTCH
0132			*
0133			* REHD TTY LINE (BUFPTR)
0134			* STORE TTY INPUT IN BUF UNTIL CR IS HIT
0135			*
0136	F06D	CE FF90	LDX #BUF ;INITIALIZE BUFPTR
0137	F070	FF FFE0	STX BUFPTR
0138	F073	8D	SEC ;SET ECHO FLAG
0139	F074	79 FFE9	ROL ECHO
0140			* BEGIN UNTIL LOOP
0141	F077	8C FFD7	RT10 CPX #BUF+71 ;TEST FOR BUF OVERFLOW
0142	F07H	26 02	BNE RT20 ;NO OVERFLOW
0143	F07C	20 47	BRH HBORI
0144	F07E	8D F41E	RT20 JSR WHITTY ;READ NEXT CHAR
0145	F081	H7 00	RT30 STH H 0,X ;INSERT CHAR INTO BUF
0146	F083	08	INX
0147			* WHILE CONDITION
0148	F084	81 00	RT30 CMP H #CR ;CARRIAGE RETURN?
0149	F086	26 EF	BNE RT10 ;NO,CONTINUE LOOP
0150			* END OF LOOP
0151			*
0152			* DECODE ONE CHAR COMMAND
0153			* COMPARE CHAR WITH TABLE OF VALID CHARS FOLLOWED BY
0154			* ADDRESSES OF APPROPRIATE ROUTINES
0155			*
0156	F088	8D F3H3	JSR PXISTS ;GET 1ST CHAR
0157	F08B	08	INX ;INC BUFPTR
0158	F08C	FF FFE0	STX BUFPTR
0159	F08F	CE F0H4	LDX #TABLE ;START OF TABLE
0160			* BEGIN LOOP
0161	F092	H1 00	DLOOP CMP A 0,X ;COMPARE
0162	F094	26 04	BNE DL10
0163			* FOUND CHAR. GET ADDRESS IMMEDIATELY FOLLOWING CHAR.
0164	F096	EE 01	LDX 1,X
0165	F098	6E 00	JMP 0,X ;GO TO PROPER ROUTINE

SEQ	LOC	OBJ	SOURCE
0166			* NO COMPARE. MOVE TO NEXT CHAR.
0167	F09A	08	DL10 INX
0168	F09B	08	INX
0169	F09C	08	INX
0170	F09D	8C F0C5	CPX #CTEND ; END OF TABLE?
0171	F0A0	26 F0	BNE DLOOP ; NO REPEAT
0172			* END LOOP
0173	F0A2	20 21	BRA ABORT ; NOT IN TABLE
0174			*
0175		F05F	MONEND EQU MONITR
0176			*
0177			* CTABLE: TABLE OF VALID 1 CHARACTER COMMANDS.
0178			* EACH ENTRY CONSISTS OF 3 BYTES. BYTE 1
0179			* CONTAINS THE ASCII CHAR. BYTES 2,3 CONTAIN THE
0180			* ADDRESS OF THE APPROPRIATE ROUTINE
0181			*
0182		F0A4	CTABLE EQU *
0183	F0A4	4C	FCB 'L
0184	F0A5	F1 08	FDB LOAD
0185	F0A7	47	FCB 'G
0186	F0A8	F1 87	FDB GO
0187	F0AA	50	FCB 'P
0188	F0AB	F3 26	FDB PUNCH
0189	F0AD	42	FCB 'B
0190	F0AE	F1 06	FDB PREGS
0191	F0B0	40	FCB 'M
0192	F0B1	F5 H2	FDB MOVE
0193	F0B3	56	FCB 'V
0194	F0B4	F4 03	FDB VFY
0195	F0B6	49	FCB 'I
0196	F0B7	F4 E0	FDB READ
0197	F0B9	53	FCB 'S
0198	F0BA	F4 01	FDB SM
0199	F0BC	44	FCB 'D
0200	F0BD	F1 4E	FDB DM
0201	F0BF	52	FCB 'R
0202	F0C0	F1 06	FDB PREGS
0203	F0C2	45	FCB 'E
0204	F0C3	F1 75	FDB EOF
0205		F0C5	CTEND EQU *
0206			*
0207			* ABORT
0208			*
0209		F0C5	ABORT EQU *
0210		F0C5	BADINP EQU *
0211	F0C5	0E F291	LDX #INQUES ; PRINT ?????
0212			*
0213			* PRINT MSG AND RETURN TO MONITOR
0214			*
0215		F0C8	MSGMON EQU *
0216		F0C8	MSGABT EQU *
0217	F0C8	5E FF8F	LDS #BOS ; S = BOTTOM OF STACK
0218	F0CB	3F	SWI
0219	F0CC	12	FCB 18 ; PMSG
0220	F0CD	20 8A	BRH MONEN1

SEQ	LOC	OBJ	SOURCE
0221			*
0222			* SWI HANDLER :
0223			* DETERMINE WHETHER SWI IS MONITOR CALL, RSRSR CALL,
0224			* OR USER SWI (NOTIMPL.)
0225			*
0226		F0CF	BREHKL EQU * ;BREAKPOINT ENTRY
0227	F0CF	01	NOP ;REPLACES JSR PINIT
0228	F0D0	01	NOP ;REPLACES JSR PINIT
0229	F0D1	01	NOP ;REPLACES JSR PINIT
0230	F0D2	86 80	LDA A #128 ;PRETEND TO BE SWI 128
0231	F0D4	20 1H	BRA SWI40 ;SAVE REGS
0232			*
0233		F0D6	SWIHAN EQU * ;
0234			* FIND INDEX BYTE (BYTE AFTER SWI THAT GOT US HERE)
0235	F0D6	30	TSX
0236	F0D7	EE 05	LDX 5,X ;X = RET ADDR
0237	F0D9	A6 00	LDH A 0,X ;H = INDEX BYTE
0238	F0D8	2B 0C	BMI SWI30 ;BREAKPOINT?
0239			* IF USER HAS ADDITIONAL (RS)**3 ADDR OF 1ST+2 MUST BE DFF4
0240	F0DD	80 18	SUB A #24 ;RSRSR CALL?
0241	F0DF	2A 03	BPL SWI20 ;NO --
0242	F0E1	7E F5BE	JMP RSRSR
0243			*
0244			* USER SWI
0245			*
0246	F0E4	FE FFF4	SWI20 LDX USWI
0247	F0E7	6E 00	JMP 0,X ;GO DO IT
0248			*
0249			* MONITOR CALL. COPY REGS FROM STACK
0250			*
0251	F0E9	30	SWI30 TSX ;INCREMENT RET ADDR
0252	F0EA	6C 06	INC 6,X
0253	F0EC	26 02	BNE SWI40
0254	F0EE	6C 05	INC 5,X
0255	F0F0	CE FFEB	SWI40 LDX #DREG ;DEST. FOR 1ST REG
0256			* BEGIN LOOP
0257	F0F3	33	SWI50 PUL B ;GET REG
0258	F0F4	E7 00	STA B 0,X ;COPY
0259	F0F6	08	INX ;MOVE TO NEXT REG
0260	F0F7	8C FFF2	CPX #DREG+7 ;END OF LOOP?
0261	F0FA	26 F7	BNE SWI50
0262			* END LOOP
0263			*
0264			* S NOW CONTAINS ITS VALUE BEFORE SWI
0265			* WAS EXECUTED. SAVE IT
0266			*
0267	F0FC	AF 00	STS 0,X
0268			*
0269			* A STILL CONTAINS SWI INDEX. TEST IT
0270			*
0271	F0FE	.. 81	CMP A #129
0272	F100	26 04	BNE PREGS ;NOT 129 : BREAK
0273	F102	80 07	BSR PR1 ;129 : SNAPSHOT
0274	F104	20 1E	BRA RESTAK ;AND RETURN TO USER PROGRAM
0275			*

SEQ	LOC	OBJ	SOURCE
0276			* PREGS : PRINT USER REGISTERS
0277			*
0278		F106	PREGS EQU *
0279	F106	8D 03	BSR PR1
0280	F108	7E F05F	JMP MONEND
0281		F10B PR1	EQU *
0282	F108	0E FFEB	LDX #CREG
0283			* PRINT 3 1-BYTE REGS
0284	F10E	06 03	LDA B #3
0285			* SET UP COUNT
0286	F110	3F	PR10 SWI
0287	F111	0F	FCB 15
0288	F112	8D F39E	JSR PSPACE
0289	F115	5A	DEC B
0290	F116	2E F8	BGT PR10
0291			*
0292			* PRINT 3 2-BYTE REGS
0293	F118	06 03	LDH B #3
0294			* SET UP COUNT
0295	F11A	8D F39A	PR20 JSR P4HEXS
0296	F11D	5A	DEC B
0297	F11E	2E FA	BGT PR20
0298			*
0299	F120	8D F322	JSR PURLF
0300	F123	39	RTS
0301			*
0302			*
0303			* RESTORE USER STATUS AND RETURN FROM MONITOR
0304			*
0305			*
0306	F124	8E FFF2	RESTAK LDS SREG
0307	F127	0E FFF1	LDX #CREG+6
0308			* BEGIN LOOP
0309	F12A	A6 00	RUS10 LDA A 0,X
0310	F12C	36	PSH A
0311	F12D	09	DEX
0312	F12E	8C FFEA	CPX #CREG-1
0313	F131	26 F7	BNE RUS10
0314			* END OF LOOP
0315	F133	38	RTI
0316			* RETURN TO USER PROG
0317			*
0318			* COMMANDS AND SUBROUTINES
0319			*
0320			* CHECKSUM (CKSM)
0321			* VALIDATE CKSM
0322			*
0323		F134	CHEKSM EQU *
0324	F134	86 FFE4	LDA A CKSM
0325	F137	36	PSH A
0326	F138	8D F2BC	JSR NEXT2D
0327	F13B	33	PUL B
0328	F13C	03	COM B
0329	F13D	11	CBA
0330	F13E	26 01	BNE CS1

SEQ	LOC	OBJ	SOURCE
0331	F140	39	RTS
0332			*
0333	F141	30	CS1 TSX ;X:= ADDR OF CALC. CKSM
0334	F142	09	DEX
0335	F143	3F	SWI
0336	F144	0F	FCB 15 ;P2HEX
0337	F145	80 F39E	JSR PSPACE
0338	F146	0E F296	LDX #MCSER ;PRINT "CKSM ERR"
0339	F146	7E F008	JMP MSGABT
0340			*
0341			* DM ADDL, ADDH COMMAND
0342			*
0343		F14E DM	EGU *
0344	F14E	8D 35	BSR GETRNG ;GET ADR RANGE FROM BUF
0345			* RETURNS ADDL, ADDH+1
0346			* BEGIN OUTER LOOP
0347	F150	0E FFDC DM10	LDX #ADDL
0348	F153	8D F39A	JSR P4HEXS ;PRINT MEM(X), SPACE, INC X
0349			* BEGIN INNER LOOP
0350	F156	FE FFDC DM20	LDX ADDL
0351	F159	3F	SWI
0352	F15A	0F	FCB 15 ;P2HEX
0353	F15B	8D F39E	JSR PSPACE
0354	F15E	FF FFDC	STX ADDL
0355	F161	8C FFDE	CPX ADDH ;ADDL=ADDH+1, END RNGE
0356	F164	27 0C	BEQ DM50 ;EXIT OUTER LOOP
0357	F166	B6 FFDD	LDA A ADDL+1 ;LSB'S OF ADDL =0, END LINE
0358	F169	84 0F	AND A #*F
0359	F16B	26 E9	BNE DM20 ;NOT END OF LINE CONTINUE
0360			* END OF INNER LOOP
0361	F16D	8D F322	JSR PCRLF ;PRINT CR, LF
0362	F170	20 DE	BRA DM10 ;EXIT INNER LOOP
0363			* END OF OUTER LOOP
0364	F172	7E F059 DM50	JMP MONEN1 ;CR, LF, BACK TO MONITOR
0365			*
0366			* PUNCH END OF FILE ANF 60 NULLS
0367			*
0368	F175	0E F2A8 EOF	LDX #MPEOF ;PUNCH EOF RECORD
0369	F178	3F	SWI
0370	F179	12	FCB 18 ;PMSG
0371			*
0372			* PUNCH 60 NULLS
0373			*
0374	F17A	06 3B	NULLS LDA B #59 ;LOAD COUNT
0375			* BEGIN LOOP
0376			*
0377	F17C	4F	NULL1 CLRA ;LOAD NULL
0378	F17D	8D F2FB	JSR OUTCH ;PRINT 1 NULL
0379	F180	5A	DEC B ;DECR LOOP CNT
0380	F181	26 F9	BNE NULL1 ;DONE ?
0381			* END OF LOOP
0382	F183	20 ED	BRA DM50 ;CR, LF, BACK TO MONITOR
0383			*
0384			*
0385			* GETRANGE (ADDL, ADDH, BUFPTR)

SEQ	LOC	OBJ	SOURCE
0386			* GET ADDRESS RANGE FROM BUF
0387			* ABORT IF INVALID
0388			* SET ADDH:= ADDH+1 TO SIMPLIFY COMPARISONS
0389			* RETURNS ADDL & ADDH+1
0390			* ALTERS ADDR, X, A, B
0391			*
0392			*
0393		F185	GETRNG EQU *
0394	F185	BD F2D6	JSR NXTADR ; GET ADDR
0395	F188	FE FFDA	LDX ADR
0396	F188	FF FFDC	STX ADDL ; STORE ADDL
0397	F18E	FF FFDE	STX ADDH ; MAY BE ONLY 1 PARAMETER
0398	F191	BD F2D6	JSR NXTADR ; GET ADDH
0399	F194	27 06	BEQ GETRG3 ; ONLY 1 PARAMETER
0400			*
0401	F196	FE FFDA	GETRG1 LDX ADR
0402	F199	FF FFDE	STX ADDH ; SAVE ADDH
0403			* THE NEXT 5 INSTR TEST ADDH-ADDL
0404	F19C	CE FF90	GETRG3 LDX #BASE ; REF W. R. T. BASE OF RAM
0405	F19F	A6 4E	LDA A ADDH-BASE, X ; MSBYTE
0406	F1A1	E6 4F	LDA B ADDH+1-BASE, X
0407	F1A3	E0 4D	SUB B ADDL+1-BASE, X
0408	F1A5	A2 4C	SBC A ADDL-BASE, X
0409	F1A7	24 06	BCC GETRG4 ; ADDH, GE, ADDL
0410	F1A9	CE F283	RNGERR LDX #RNGERR ; RANGE ERR MSG
0411	F1AC	7E F0C8	JMP MSGABT ; PRINT MSG & ABORT
0412			*
0413	F1AF	FE FFDE	GETRG4 LDX ADDH ; INC ADDH
0414	F1B2	08	INX
0415	F1B3	FF FFDE	STX ADDH
0416	F1B6	39	RTS
0417			*
0418			* GO COMMAND
0419			*
0420	F1B7	CE F900	GO LDX #F900 ; SET USER STACK VALUE
0421	F1BA	FF FFF2	STX SREG ; STORE IN USER ADDR
0422	F1BD	BD F2D6	JSR NXTADR ; GET PARAMETER
0423	F1C0	27 06	BEQ G10 ; NO PARAM, CONT EXECUT.
0424			*
0425	F1C2	FE FFDA	LDX ADR ; ADR=PARAM FROM NXTADR
0426	F1C5	FF FFF0	STX PREG
0427			*
0428	F1C8	7E F124	G10 JMP RESTAK ; (IN INTERRUPT HANDLER)
0429			*
0430			* LOAD COMMAND
0431			*
0432		F1C8	LOAD EQU *
0433	F1CB	CE 0000	LDX #0 ; INITIALIZE RANGE & OFFSET
0434	F1CE	FF FFDA	STX OFFSET ; TO 0000-FFFF, 0000
0435	F1D1	FF FFDC	STX ADDL
0436	F1D4	09	LOOPST DEX
0437	F1D5	FF FFDE	STX ADDH
0438	F1D8	BD F2D6	JSR NXTADR ; ANY OPERANDS?
0439	F1DB	27 1E	BEQ LHF2 ; NO, USE DEFAULT
0440	F1DD	FE FFDA	LDX ADR ; YES

SEQ	LOC	OBJ	SOURCE
0441	F1E0	FF FFD8	STX OFFSET ; IF ONE, IT'S OFFSET
0442	F1E3	8D F2D6	JSR NXTADR ; ANOTHER?
0443	F1E6	27 13	BEQ LHF2 ; NO
0444	F1E8	FE FFD8	LDX OFFSET ; YES, 1ST TWO ARE RANGE
0445	F1EB	FF FFDC	STX ADDL
0446	F1EE	CE 0000	LDX #0
0447	F1F1	FF FFD8	STX OFFSET
0448	F1F4	8D A0	BSR GETRG1
0449	F1F6	FE FFDE	LDX ADDH
0450	F1F9	20 D9	BRA LOOPST ; GO TRY AGAIN FOR OFFSET
0451			* BEGIN OUTER LOOP
0452	F1FB	8D F3C3	LHF2 JSR RDRON ; TURN ON READER
0453			* SHORT LOOP TO SKIP HDR RECORDS
0454	F1FE	8D 70	RDPRE BSR FINDS ; FIND START OF RECORD
0455			* SETS (ECHO):=0 ON ENTRY
0456	F200	8D F41E	JSR WAITTY ; RETURNS (A):=TTY I/P
0457	F203	81 30	CMP A #10 ; IGNORE HDR RECORDS
0458	F205	27 F7	BEQ RDPRE
0459			* END SHORT LOOP
0460	F207	B7 FFE2	STA A RECTYP ; SAVE RECORD TYPE
0461	F20A	7F FFE4	CLR CKSM
0462	F20D	8D F2BC	JSR NEXT2D ; READ BYTE COUNT FROM TAPE
0463	F210	4A	DEC A
0464	F211	4A	DEC A
0465	F212	4A	DEC A
0466	F213	B7 FFE3	STA A COUNT ; SAVE BYTE COUNT
0467	F216	8D F2BC	JSR NEXT2D ; READ HDR FIELD FROM TAPE
0468	F219	B7 FFDA	STA A ADR ; 1ST BYTE
0469	F21C	8D F2BC	JSR NEXT2D
0470	F21F	88 FFD9	ADD A OFFSET+1
0471	F222	B7 FFD8	STA A ADR+1 ; 2ND BYTE
0472	F225	B6 FFDA	LDA A ADR ; CARRY TO FIRST BYTE
0473	F228	B9 FFD8	ADC A OFFSET
0474	F22B	B7 FFDA	STA A ADR
0475	F22E	B6 FFE2	LDA A RECTYP ; GET RECORD TYPE (0,1,9)
0476	F231	81 31	LHF3 CMP A #1 ; DATA RECORD?
0477	F233	26 14	BNE LHF4 ; NO
0478			*
0479			*LOAD DATA RECORD
0480			*
0481			* BEGIN UNTIL LOOP
0482			*
0483	F235	8D F2BC	LDR10 JSR NEXT2D ; READ 2 HEX DIGITS FROM
0484			* TAPE. RETURNS IN A
0485	F238	FE FFDA	LDX ADR
0486	F23B	8D F3CD	JSR SETOFF ; STORE IN MEM(X), VERIFY
0487	F23E	08	INX
0488	F23F	FF FFDA	STX ADR
0489	F242	7A FFE3	DEC COUNT ; DOES COUNT = 0
0490	F245	2E EE	BGT LDR10 ; NO, CONT LOOP
0491			* END UNTIL LOOP
0492	F247	20 04	BRA LHF9
0493	F249	81 39	LHF4 CMP A #9 ; EOF RECORD?
0494	F24B	26 13	BNE BADTAP ; ILLEGAL RECORD TYPE
0495			*

SEQ	LOC	OBJ	SOURCE
0496	F240	80 F134	LHF9 JSR CHEKSM ; CHECK CKSM
0497	F250	86 FFE2	LDA A RECTYP ; GET RECORD TYPE
0498	F253	81 39	CMP A #'9 ; EOF RECORD?
0499	F255	26 A4	BNE LHF2 ; NO, CONT LOOP
0500			*
0501			* END OF OUTER LOOP
0502			*
0503	F257	80 F389	JSR RDROFF
0504	F25A	0E F28D	LDX #NEOF ; PRINT "EOF"
0505	F25D	7E F0C8	JMP MSGMON ; AND RETURN TO MONITOR LOOP
0506			*
0507	F260	80 F389	BADTAP JSR RDROFF
0508			*
0509	F263	0E F29F	LDX #MTAPER ; PRINT "TAPE ERR"
0510	F266	3F	SWI
0511	F267	12	FCB 18 ; PMSG
0512			*
0513			* ACCEPT NO COMMANDS UNTIL USER PRESSES ESC
0514			*
0515	F268	7C FFE9	INC ECHO ; SET ECHO
0516			*
0517	F26B	80 F41E	BT1 JSR WAITTY ; ESC CAUSES ABORT
0518	F26E	20 FB	BRA BT1
0519			*
0520			* FIND S
0521			* READ TAPE UNTIL START OF RECORD
0522			*
0523		F270	FINDS EQU *
0524	F270	7F FFE9	CLR ECHO ; NO ECHO
0525			* BEGIN LOOP
0526	F273	80 F41E	FS10 JSR WAITTY ; READ NEXT TAPE CHAR
0527	F276	81 53	CMP A #'S ; CHAR = S ?
0528	F278	26 F9	BNE FS10 ; NO
0529			* END LOOP
0530	F27A	39	RTS
0531			*
0532			* MESSAGES
0533			*
0534	F27B	42 4144	MBADR FCB /BAD ADR/
0535	F282	04	FCB \$04
0536	F283	52 414E	MRNGER FCB /RANGE ERR/
0537	F28C	04	FCB \$04
0538	F28D	45 4F46	NEOF FCB /EOF/
0539	F290	04	FCB \$04
0540	F291	3F 3F3F	MOUES FCB /????/
0541	F295	04	FCB \$04
0542	F296	43 4B53	MOGER FCB /CKSM ERR/
0543	F29E	04	FCB \$04
0544	F29F	54 4150	MTAPER FCB /TAPE ERR/
0545	F2A7	04	FCB \$04
0546	F2A8	53 3930	MPEOF FCB /S9030000FC/
0547	F2B2	04	FCB \$04
0548	F2B3	00	MCRLFS FCB CR
0549	F2B4	0A	FCB LF
0550	F2B5	00	FCB \$00

```

-- SEQ LOC OBJ SOURCE
** 0551 F2B6 00 FCB $00
0552 F2B7 00 FCB $00
0553 F2B8 00 FCB $00
0554 F2B9 53 FCB 'S
0555 F2BA 31 FCB '1
0556 F2BB 04 FCB $04
0557 *
0558 * NEXT 2 DIGITS--
0559 * READ NEXT 2 CHAR FROM TTY TAPE AND
0560 * CONVERT TO HEX NUMBER IN A REG. UPDATE CKSM.
0561 * RETURN UPDATED CKSM IN B REG.
0562 *
0563 F2BC NEXT2D EQU *
0564 F2BC BD F41E JSR WAITTY ;GET CHAR
0565 F2BF 16 TAB ;SAVE CHAR IN A
0566 F2C0 BD F41E JSR WAITTY
0567 *
0568 * SET UP PARAMS FOR CONVERSION ROUTINE,
0569 * PUSH ASCII CHARS INTO STACK, POINT X AT STACK
0570 * SET A=TYPE OF CONVERSION AND B=# OF CHARS TO CONVERT
0571 *
0572 F2C3 36 PSH A
0573 F2C4 37 PSH B
0574 F2C5 30 TSX
0575 F2C6 C6 02 LDA B #2
0576 F2C8 3F SWI
0577 F2C9 15 FCB 21 ;CONVB CNVT ASCII TO BINARY
0578 F2CA 24 94 BCC BADTAP ;IF NON-HEX CHAR. ABORT
0579 *
0580 F2CC 17 TBA ;UPDATE CHECKSUM
0581 F2CD FB FFE4 ADD B CKSM
0582 F2CE F7 FFE4 STA B CKSM
0583 F2D3 31 INS ;RESTORE STACK POINTER
0584 F2D4 31 INS
0585 F2D5 39 RTS
0586 *
0587 * NEXT ADR(BUFPTR,ADR)
0588 *
0589 * SET ADR=0 OR NEXT NUMBER STRING STARTING
0590 * AT BUFPTR
0591 * LEAVES BUFPTR AT CR, DELIMITER, OR 1ST
0592 * CHAR BETWEEN G - Z
0593 * LEAVES (A)= LAST CHAR SCANNED
0594 * LEAVES (B)= LS BYTE OF ADR
0595 *
0596 * RETURNS: CC=Z FOR NO PARAMETER
0597 * ABORTS IF NON-HEX PARAMETER
0598 *
0599 F2D6 NXTADR EQU *
0600 F2D6 7F FFDA CLR ADR ;ADR=0
0601 F2D9 7F FFDB CLR ADR+1
0602 F2DC BD F3A3 JSR PKISTS ;IS THERE A PARAMETER?
0603 F2DF 26 01 BNE NA1 ;YES
0604 F2E1 39 RTS ;RETURN W/NO PARAM CC=Z
0605 *

```

SEQ	LOC	OBJ	SOURCE
0606			* SET UP PARAMS FOR ASCII TO HEX CONVERSION
0607			*
0608	F2E2	C6 47	NA1 LDA B #71 ; MAX. CHARS TO SCAN
0609	F2E4	3F	SWI
0610	F2E5	15	FCB 21 ; CONHB
0611	F2E6	FF FFE0	STX BUFPTR
0612	F2E9	B7 FFDH	STA A ADR ; SAVE RESULT
0613	F2EC	F7 FFD8	STA B ADR+1
0614	F2EF	A6 00	LDA A 0,X ; CHECK TERMINATOR
0615	F2F1	3F	SWI
0616	F2F2	13	FCB 19 ; ALPHNUM, IS CHAR ALPHA?
0617	F2F3	25 01	BCS NA3 ; YES
0618	F2F5	39	RTS
0619			*
0620	F2F6	7E F0C5	NA3 JMP ABORT ; NO
0621			*
0622			* OUTCH = PRINT CHAR IN A
0623			* OUTCHX = PRINT CHAR AT MEM(X)
0624			* IF CHAR = 'CR', FOLLOW WITH LF & 4 NULLS
0625			*
0626	F2F9	A6 00	OUTCHX LDA A 0,X ; ENTRY1
0627			*
0628		F2FB	OUTCH EQU *
0629			* FIRST CHECK FOR ESC
0630	F2FB	37	PSH B
0631	F2FC	F6 6000	LDA B ACIAS ; ACIA INPUT STATUS
0632	F2FF	57	ASR B ; C: = RDRF
0633	F300	24 0A	BCC 0C10 ; NO INPUT
0634	F302	F6 6001	LDA B ACIAD ; READ ACIA
0635	F305	C1 1B	CMP B #ESC
0636	F307	26 03	BNE 0C10 ; NOT ESC
0637	F309	7E F0C5	JMP ABORT
0638			*
0639	F30C	3F	0C10 SWI ; PRINT CHAR
0640	F30D	11	FCB 17 ; PUTA
0641	F30E	81 0D	CMP A #CR
0642	F310	26 0E	BNE 0C20 ; NOT CR, RETURN
0643			*
0644	F312	86 0A	LDA A #LF ; PRINT LF
0645	F314	3F	SWI
0646	F315	11	FCB 17 ; PUTA
0647	F316	4F	CLR A ; PRINT 4 NULLS
0648	F317	C6 04	LDA B #4 ; SET LOOP CNT
0649			* BEGIN LOOP
0650	F319	3F	0CLOOP SWI
0651	F31A	11	FCB 17 ; PUTA
0652	F31B	5A	DEC B
0653	F31C	26 FB	BNE 0CLOOP
0654			* END LOOP
0655	F31E	86 0D	LDA A #CR ; RESTORE A
0656			*
0657	F320	33	0C20 PUL B
0658	F321	39	RTS
0659			*
0660			* PRINT CR, LF, NULL

SEQ	LOC	OBJ	SOURCE
0661			*
0662			*
0663	F322	86 0D	PCRLF LDA A #CR
0664	F324	20 05	BRA OUTCH ; OUTCH PRINTS LF AFTER CR
0665			*
0666			* PUNCH ADDL,ADDH
0667			* PUNCH MEMORY CONTENTS BETWEEN ADDL & ADDH
0668			* IN HEX FORMAT
0669			*
0670	F326	BD F185	PUNCH JSR GETRNG ; READ ADDL & ADDH+1
0671	F329	CE 0000	LDX #0
0672	F32C	FF FFD8	STX OFFSET
0673	F32F	8D A5	BSR NXTADR ; ANY OFFSET?
0674	F331	27 06	BEQ PHF15 ; NO
0675	F333	FE FFD8	LDX ADR ; YES
0676	F336	FF FFD8	STX OFFSET
0677			*
0678			* PUNCH DATA RECORD UNTIL ADDL = ADDH
0679			*
0680		F339	PHF15 EQU *
0681			* BEGIN LOOP
0682			*
0683			* CALCULATE DATA LENGTH = MIN(30,ADDH+1-ADR)
0684			*
0685	F339	F6 FFD8	PHF20 LDA B ADDH+1 ; B:=ADDH-ADDL
0686	F33C	F0 FFD0	SUB B ADDL+1
0687	F33F	B6 FFD8	LDA A ADDH
0688	F342	B2 FFD0	SBC A ADDL
0689	F345	26 04	BNE PUND10 ; DIFF GT. 256
0690	F347	C1 1E	CMF B #30 ; LS BYTE GT. 30?
0691	F349	23 02	BLS PUND20
0692			*
0693	F34B	C6 1E	PUND10 LDA B #30 ; DIFF GT. 30
0694			*
0695	F34D	5C	PUND20 INC B ; COUNT:=COUNT+3
0696	F34E	5C	INC B ; ... INCLUDES ADDR & CKSM
0697	F34F	5C	INC B
0698	F350	F7 FFE3	STA B COUNT
0699	F353	CE F2B3	LDX #MCRLF5
0700	F356	3F	SWI
0701	F357	12	FCB 18 ; PMSG
0702	F358	5F	CLR B ; B HOLDS CKSM
0703	F359	CE FFE3	LDX #COUNT
0704	F35C	8D 34	BSR PUNBYT
0705	F35E	37	PSH B
0706	F35F	FE FFD0	LDX ADDL ; COMPUTE OFFSET ADDRESS
0707	F362	B6 FFD8	LDA A OFFSET
0708	F365	F6 FFD9	LDA B OFFSET+1
0709	F368	3F	SWI
0710	F369	08	FCB 8 ; ADDR BX
0711	F36A	FF FFD8	STX ADR ; PUNCH FROM ADR
0712	F36D	CE FFD8	LDX #ADR
0713	F370	33	PUL B
0714			*
0715	F371	8D 1F	BSR PUNBYT ; (INCREMENTS X)

SEQ	LOC	OBJ	SOURCE
0716	F373	8D 1D	BSR FUNBYT
0717	F375	FE FFDC	LDX ADDL ; RESTORE X
0718			*
0719			* PUNCH BYTES FROM MEMORY UNTIL COUNT IS EXHAUSTED
0720			*
0721			* BEGIN LOOP
0722	F378	8D 18	PREC10 BSR FUNBYT ; (CC=0 IF COUNT = 0)
0723	F37A	2E FC	BGT PREC10
0724			* END LOOP
0725	F37C	FF FFDC	STX ADDL ; SAVE X
0726	F37F	0E FFE4	LDX #CKSM ; PUNCH CKSM
0727	F382	53	COM B
0728	F383	E7 00	STA B 0,X ; CKSM:=B
0729	F385	8D 0B	BSR FUNBYT
0730	F387	FE FFDC	LDX ADDL
0731	F38A	6C FFDE	CPX ADDH
0732	F38D	26 AA	BNE PHF20
0733			* END LOOP
0734	F38F	7E F055	JMP MONEN1
0735			*
0736			* FUNBYT (MEM(X), COUNT, CKSM)
0737			* PUNCH BYTE AT MEM(X) AND ADJUST COUNT AND CKSM.
0738			* CC=2 IF COUNT =0
0739			*
0740	F392	EB 00	FUNBYT ADD B 0,X ; CKSM:=CKSM+MEM(X)
0741	F394	3F	SWI
0742	F395	0F	FCB 15 ; P2HEX, PRINT MEM(X) AS 2 CHAR
0743	F396	7A FFE3	DEC COUNT
0744	F399	39	RTS
0745			*
0746			* P4HEXS: PRINT 2 BYTES AT X AS 4 HEX CHARS + 2 SPACES
0747			*
0748	F39A	3F	P4HEXS SWI
0749	F39B	10	FCB 16 ; P4HEX
0750	F39C	8D 00	BSR PSPACE
0751			*
0752			* PSPACE--- PRINT 1 BLANK SPACE
0753			*
0754	F39E	86 20	PSPACE LDA R #BLANK
0755	F3A0	3F	SWI
0756	F3A1	11	FCB 17 ; PUTA
0757	F3A2	39	RTS
0758			*
0759			* PARAM EXISTS(BUFPTR) (#BUFPTR)= BUFPTR
0760			* (X) = BUFPTR
0761			* INC BUFPTR UNTIL CHAR = ALPHA OR CR
0762			* LEAVE A = MEM(BUFPTR)
0763			* SET Z IF NO PARAMETER EXISTS
0764			*
0765		F3A3	PXISTS EQU + ; ENTRY FOR(#BUFPTR)=BUFPTR
0766	F3A3	FE FFE0	LDX BUFPTR
0767		F3A6	PXISTX EQU + ; ENTRY FOR (X)=BUFPTR
0768			* BEGIN LOOP
0769	F3A6	A6 00	PX1 LDA R 0,X ; IS CHAR ALPHANUM?
0770	F3A8	3F	SWI

SEQ	LOC	OBJ	SOURCE
0771	F3A9	13	FCB 19 ;ALPNUM
0772	F3AA	25 07	BCS PX2 ;YES, EXIT LOOP
0773	F3AC	81 0D	CMP A #CR ;IS CHAR CR?
0774	F3AE	27 03	BEQ PX2 ;YES, EXIT LOOP
0775	F3B0	08	INX ;MOVE TO NEXT CHAR
0776	F3B1	20 F3	BRA PX1
0777			* END LOOP
0778	F3B3	FF FFE0 PX2	STX BUFPTR
0779	F3B6	81 0D	CMP A #CR ;SET Z IF NO PARAMETER
0780	F3B8	39	RTS
0781			*
0782			* RDR OFF
0783			* TURNS TAPE RDR OFF:
0784			* ACIA RTS O/P HIGH
0785			* ACIA CHAR #2A (*)
0786			* OLD VERSION OUTPUT ASCII 13 (0D)
0787		F3B9 RDR OFF	EQU *
0788	F3B9	86 01	LDA A #01 ;RTS HIGH
0789	F3BB	B7 6000 RDR90	STA A ACIAC ;SET ACIA CONT REG
0790	F3BE	86 2A	LDA A #2A ;SEND TTY RDR CONT CHAR
0791	F3C0	3F	SWI
0792	F3C1	11	FCB 17 ;PUTA
0793	F3C2	39	RTS
0794			*
0795			* RDR ON:
0796			* TURNS ON TAPE READER
0797			* ACIA RTS O/P LOW
0798			* ACIA CHAR #11 (0D)
0799			*
0800		F3C3 RDR ON	EQU *
0801	F3C3	86 41	LDA A #41 ;RTS LOW
0802	F3C5	B7 6000 RDR90	STA A ACIAC ;SET ACIA CONT REG
0803	F3C8	86 11	LDA A #11 ;SEND TTY RDR CONT CHAR
0804	F3CA	3F	SWI
0805	F3CB	11	FCB 17 ;PUTA
0806	F3CC	39	RTS
0807			*
0808			* SETMEM(X)
0809			* SETS MEM(X):=H AND VERIFY
0810			*
0811		F3CD SET OFF	EQU *
0812	F3CD	36	PSH A ;FIRST CHECK RANGE
0813	F3CE	B6 FFDC	LDA A ADDL ;LOW LIMIT
0814	F3D1	F6 FFDD	LDA B ADDL+1
0815	F3D4	3F	SWI
0816	F3D5	06	FCB 11 ;SUBXAB
0817	F3D6	22 0A	BHI SETOUT ;TOO LOW
0818	F3D8	B6 FFDE	LDA A ADDH ;HIGH LIMIT
0819	F3DB	F6 FFDF	LDA B ADDH+1
0820	F3DE	3F	SWI
0821	F3DF	06	FCB 11 ;SUBXAB
0822	F3E0	24 07	BCC SETFUL ;OK
0823	F3E2	32	SETOUT ;OUTSIDE RANGE LIMITS
0824	F3E3	86 FF	LDA A #255 ;TYPE DELETE(RUBOUT)
0825	F3E5	3F	SWI

SEQ	LOC	OBJ	SOURCE
0826	F3E6	11	FCB 17 ;PUTA SIGNAL FACT TO USER
0827	F3E7	20 17	BRA SETM1 ;OTHER, IGNORE STORE REQ
0828	F3E9	32	SETPUL PUL A
0829	F3EA	A7 00	SETMEM EQU *
0830	F3EB	A1 00	STA A 0,X
0831	F3EC	27 10	CMP A 0,X ;VERIFY
0832	F3EE		BEQ SETM1 ;ERROR?
0833			* VERIFY ERROR, PRINT ADR
0834	F3F0	FF FFDH	STX ADR ;SET PARAM FOR P4HEX
0835	F3F2	0E FFDH	LDX #ADR
0836	F3F6	80 01	BSR RDRDFF
0837	F3F8	80 A0	BSR P4HEXS
0838	F3FA	0E F27B	LDX #MBADR ;PRINT "BAD ADR"
0839	F3FD	7E F008	JMP MSGABT ;PRINT MSG & ABORT
0840			*
0841	F400	29	SETM1 RTS
0842			*
0843			* SM ADR BYTE1, BYTE2, ...
0844			*
0845	F401	8D F206	SM EQU * ;ADR:= NEXT PARAM
0846	F401	8D F206	JSR NXTADR
0847			*
0848	F404	FE FFDH	STX ADR ;SAVE ADR IN ADR
0849	F407	FE FFDH	STX ADDL
0850			* BEGIN WHILE LOOP
0851	F40A	8D F206	SM10 JSR NXTADR ;ADR:= NEXT PARAM
0852	F40D	27 01	BEQ SM10 ;END OF LINE, EXIT LOOP
0853	F40F	FE FFDH	LDX ADDL ;X:= ADD TO BE SET
0854	F412	17	TEA ;A:= LS BYTE
0855	F415	80 03	BSR SETMEM ;MEM(X):=A, VERIFY
0856	F418	08	INX ;MOVE TO NEXT ADD
0857	F41B	FE FFDH	STX ADDL
0858	F41B	20 EF	BRA SM10
0859			* END OF LOOP
0860	F41B	7E F00F	SM10 JMP MONEND
0861			*
0862			* WAIT FOR TTY(CHAR, ECHO) (#ECHO)=ECHO
0863			* RETURN NEXT TTY CHAR IN A
0864			* IF (#ECHO) NOT 0, ECHO CHAR
0865			*
0866	F41E		WAITTY EQU * ;
0867			* LOOP UNTIL INPUT, NE, RUBOUT
0868	F41E	3F	W10 SWI ;READ TTY
0869	F41F	14	FCB 20 ;GETA
0870	F420	81 1B	CMP A #ESC ;ESCAPE?
0871	F422	26 03	BNE W20 ;NO
0872	F424	7E F00C	JMP ABORT ;YES, ABORT
0873	F427	81 7F	W20 CMP A #RUBOUT ;RUBOUT?
0874	F429	27 F3	BEQ W10 ;YES CONTINUE
0875			* END UNTIL LOOP
0876	F42B	7D FFE3	TST ECHO
0877	F42E	27 03	BEQ W20 ;NO ECHO
0878	F430	8D F2FB	JSR OUTCH ;ECHO A
0879	F433	39	W20 RTS
0880			* WAS ORG #F42E

SEQ	LOC	OBJ	SOURCE
0881		0001	MOVER EQU 1 ;0=MOVE ROUTINE EXCLUDED
0882		000A	DELAY EQU 10 ;POST PROG DELAY BEFORE VFY
0883			* PIA LOCATIONS
0884		5000	PIA EQU \$5000
0885		0001	V50 EQU \$5001-PIA
0886		0004	PROM EQU \$5004-PIA
0887			* STANDARD RAM BUFFER (DEFAULT)
0888		FC00	RAM EQU \$FC00
0889			*
0890			*RSRER CALL LOCATIONS
0891			*
0892			* INITIALIZE PROM BURNER PIA'S
0893		F434	PINIT EQU *
0894	F434	39	RTS ;SUB FOR ORG MONITOR
0895			* WAS ORG \$F430
0896			*
0897			* TYPE A IN BINARY, ENCLOSED BY SPACES
0898			*
0899		F435	P8BIN EQU *
0900	F435	37	PSHB ;SAVE B
0901	F436	36	PSHA
0902	F437	8D 0F	BSR PSP ;PRINT LEADING SPACE
0903	F439	32	PUL A
0904	F43A	06 08	LDA B #8 ;8 DIGIT COUNTER
0905	F43C	49	ROL A
0906	F43D	36	PSH A
0907	F43E	86 18	LDA A #24 ; (=1/2 ASCII "0")
0908	F440	49	ROL A
0909	F441	3F	SWI
0910	F442	11	FCB 17 ;PRINTA
0911	F443	32	PUL A
0912	F444	5A	DEC B
0913	F445	26 F5	BNE A8
0914	F447	33	PUL B
0915	F448	7E F39E	PSP JMP PSPACE ;PRINT ONE MORE SPACE
0916			*
0917			* RAM/ROM ADDRESS SETUP & VALIDATION
0918			*
0919	F44B	0E FC00	RASV LDX #RAM ;INITIALIZE POINTERS TO DEFAULT RAM
0920	F44E	FF FFDC	STX ADDL
0921	F451	0E FE00	LDX #RAM+512
0922	F454	FF FFDE	STX ADDH
0923	F457	7F FFE3	CLR COUNT ;SET FULL PROM FLAG
0924	F45A	8D F3A3	JSR PKISTS ;... IF NO ADDRESS
0925	F45D	27 06	BEQ AA1
0926	F45F	8D F185	JSR GETRNG
0927	F462	7D FFE3	INC COUNT
0928	F465	FE FFDC	AA1 LDX ADDL ;DEFAULT PROM ADDRESS
0929	F468	FF FFDC	STX PROMAD ;IS SAME AS START
0930	F46B	8D F206	JSR NXTADR ;TRY FOR FROM ADDRESS
0931	F46E	27 06	BEQ AA3 ;NO
0932	F470	FE FFDA	LDX ADR ;YES
0933	F473	FF FFDC	STX FROMAD
0934	F476	0E FFDC	AA1 LDX #FROMAD ;VERIFY THAT RANGE <= 512
0935	F479	0D	SEC ;(FORCE BORROW)

SEQ	LOC	OBJ	SOURCE
0936	F47A	E6 07	LDA B 7,X ;=ADDH+1
0937	F47C	E2 05	SBC B 5,X ;=ADDL+1
0938	F47E	A6 06	LDA A 6,X
0939	F480	A2 04	SBC A 4,X
0940	F482	81 02	CMP A #2 ; SHOULD BE 1 OR 0
0941	F484	2C 08	BGE AA4 ; TOO BIG
0942	F486	EB 01	ADD B 1,X ; + SHOULD NOT OVERSP FROM
0943	F488	A9 00	ADC A 0,X
0944	F48A	A8 00	EOR A 0,X
0945	F48C	84 FE	AND A #3FE
0946	F48E	26 01	BNE AA4 ; IT DOES
0947	F490	39	RTS
0948	F491	7E F1A9 AA4	JMP RANGERR ; ADDRESS RANGE ERROR
0949			*
0950			* TYPE RAM & ROM ADDRESS & DATA
0951			*
0952	F494	CE FFDC VERR	LDX #ADDL ; TYPE RAM ADDRESS
0953	F497	3F	SWI
0954	F498	10	FCB 16 ; P4HEX
0955	F499	FE FFDC	LDX ADDL ; NOW THE BYTE THERE
0956	F49C	A6 00	LDA A 0,X
0957	F49E	8D 95	BSR P8BIN
0958	F4A0	B6 5006	LDA A FROM+2+PIA ; THEN FROM DATA
0959	F4A3	8D 90	BSR P8BIN
0960	F4A5	CE FFD8	LDX #FROMAD ; NOW IF ADDRESS (LOW B)
0961	F4A8	A6 01	LDA A 1,X ; DOES NOT MATCH RAM ADDRESS
0962	F4AA	A1 05	CMP A 5,X ; =ADDL
0963	F4AC	27 02	BEQ AT
0964	F4AE	3F	SWI ; PRINT FROM ADDRESS
0965	F4AF	10	FCB 16 ; P4HEX
0966	F4B0	BD F322 AT	JSR PCRLF
0967	F4B3	86 40	LDA A #64
0968	F4B5	48	ASL A ; EXIT C=0, Z=0, V=1
0969	F4B6	39	RTS
0970			*
0971			* FROM ADDRESS SETUP & DATA READ
0972			*
0973	F4B7	CE FFD8 ADDR5	LDX #FROMAD ;
0974	F4B8	A6 01	LDA A 1,X ; LOW 8 BITS
0975	F4BC	B7 5004	STA A FROM+PIA
0976	F4BF	A6 00	LDA A 0,X ; HIGH BIT
0977	F4C1	CE 5000	LDX #PIA
0978	F4C4	48	ASL A ; POSITION IT
0979	F4C5	4C	INC A ; WITH DATA REGISTER SELECT
0980	F4C6	48	ASL A
0981	F4C7	48	ASL A
0982	F4C8	A8 07	EOR A FROM+5,X ; INSERT INTO CONTROL
0983	F4CA	84 0C	AND A #1C
0984	F4CC	A8 07	EOR A FROM+5,X
0985	F4CE	A7 07	STA A FROM+5,X
0986	F4D0	A6 05	LDA A FROM+5,X ; READ DATA
0987	F4D2	39	RTS
0988			*
0989			* FROM VERIFY
0990			*

SEQ	LOC	OBJ	SOURCE
0991	F403	80 F446	VFY JSR RASV ;GO SET UP ADDRESSES
0992	F406	80 20	AV BSR VERR ;VERIFY ONE LOCATION
0993	F408	24 02	BCC AN ;NO ERROR OR PRINTED
0994	F40A	80 3A	BSR JVER ;PRINT FIXABLE ERROR
0995	F40C	80 11	BSR INCD ;INCREMENT ADDRESSES
0996	F40E	20 F6	BRH AV
0997			*
0998			* FROM READ
0999			*
1000	F4E0	80 F446	READ JSR RASV ;SET UP POINTERS
1001	F4E1	80 02	AR BSR ADDRS ;READ ONE BYTE
1002	F4E3	FE FFDC	LDX ADDL
1003	F4E8	80 F2EH	JSR SETMEM ;STORE IN RAM
1004	F4EB	80 02	BSR INCD ;NEXT
1005	F4ED	30 F4	BRH AR
1006			*
1007			* INCREMENT RAM/ROM ADDRESS POINTERS
1008			*
1009	F4EF	FE FFDC	INCD LDX PROMAD
1010	F4F2	08	INX
1011	F4F3	FE FFDC	STX PROMAD
1012	F4F6	FE FFDC	INX LDX ADDL
1013	F4F9	08	INX
1014	F4FA	FE FFDC	STX ADDL
1015	F4FD	80 FFDE	CPX ADDH
1016	F500	26 B5	BNE ADDRS
1017	F502	7E F03F	EXIT JMP MONTR ;EXIT TO MONITOR
1018			*
1019			* FROM DATA VERIFY ONE BYTE
1020			*
1021	F505	80 80	VFY1 BSR ADDRS ;SET UP & READ A BYTE
1022	F507	FE FFDC	LDX ADDL ;COMPARE TO RAM
1023	F50A	A1 00	CMP A 0,X
1024	F50C	27 07	BEQ AX ;OK: C=0, Z=1, V=0
1025	F50E	43	COM A ;NO, IS IT FIXABLE
1026	F50F	AH 00	ORA A 0,X ;I. E. NO RAM=0, PROM=1?
1027	F511	43	COM A
1028	F512	26 02	BNE JVER ;YES. C=1, Z=0, V=0
1029	F514	40	INC A
1030	F515	39	RTS
1031	F516	7E F494	JVER JMP VERR ;NO, TYPE ERROR
1032			*
1033			* FROM BURNER ROUTINE
1034			* NOT CODED
1035			*WAS ORG #F514
1036		F519	BURN EQU *
1037	F519	7E F0C5	JMP ABORT
1038			*WAS ORG #F520
1039			* POWER ON INITIAL SEQUENCE
1040			* SETS PIA TO DETERMINE MODE
1041			* PORT "A" = INPUT
1042			* PORT "B" = OUTPUT + 1 INPUT
1043			* PB0 - PB6 = OUTPUTS
1044			* PB7 = INPUT OF TTY ENABLE (0=TTY CONNECT)
1045		D800	OPMODE EQU #D800 ;NORMAL OP MODE FROM ADDR

SEQ	LOC	OBJ	SOURCE				
1046		F510	FWRI	EDU	*		
1047	F510	86	00	LDA	A	;\$00 ;LOAD CRA WORD	
1048	F51E	87	5001	STA	A	;\$5001 ;SET CRA	
1049	F521	87	5000	STA	A	;\$5000 ;ENB DDRA ALL INPUT	
1050	F524	86	04	LDA	A	;\$04 ;SET PRA ENAB WORD	
1051	F526	87	5001	STA	A	;\$5001 ;ENAB TO READ PRA	
1052	F529	86	00	LDA	A	;\$00 ;SET CRB WORD	
1053	F52B	87	5003	STA	A	;\$5003 ;SET CRB	
1054	F52E	86	7F	LDA	A	;\$7F ;SET DORB WORD	
1055	F530	87	5002	STA	A	;\$5002 ;SET DORB	
1056	F533	86	04	LDA	A	;\$04 ;SET CRB WORD	
1057	F535	87	5003	STA	A	;\$5003 ;ENAB PRB	
1058	F538	86	60	LDA	A	;\$60 ;SET PRB WORD	
1059	F53A	87	5002	STA	A	;\$5002 ;SET PRB	
1060	F53D	86	5000	LDA	A	;\$5000 ;READ PORT A DATA	
1061	F540	84	10	AND	A	;\$10 ;MASK ALL BUT TTY	
1062	F542	27	00	BEQ		MCINIT ;BRANCH IF TTY CONNECTED	
1063	F544	7F	0051	CLR		MONFG ;RESET MONITOR CONN FLAG	
1064	F547	7F	00E6	CLR		NORFG ;RESET NORMAL FLAG	
1065	F54A	7F	00E7	CLR		V\$FLG ;VSTAR TEST FLAG	
1066	F54D	7E	D800	JMP		OPMODE ;JUMP IF NORMAL OP MODE	
1067						*	
1068						* PTM2 SET-UP . SETS CR3 AND TIMER 3	
1069						* TO GENERATE CLOCK FOR ACIA.	
1070						* TIMER 3 CONTINUOUS, NO INTERRUPT (IRQ).	
1071						* FREQUENCY = 1 MHZ/52/4 = 16*300 BAUD (4800 HZ.)	
1072						* TIMER STARTS IMMEDIATELY(G-D+R+W)	
1073	F550	86	81	MCINIT	LDA	A	;\$81 ;LOAD WORD FOR CR2
1074	F552	87	3001	STA	A	;\$3001 ;STORE IN CR2	
1075	F555	86	80	LDA	A	;\$80 ;LOAD WORD FOR CR1, CR2	
1076	F557	87	3000	STA	A	;\$3000 ;STORE IN CR1	
1077	F55A	87	3001	STA	A	;\$3001 ;STORE IN CR2	
1078	F55D	86	82	LDA	A	;\$82 ;LOAD WORD FOR CR3	
1079	F55F	87	3000	STA	A	;\$3000 ;STORE IN CR3	
1080	F562	0E	0068	LDX		;\$0068 ;LD DIV CNT FOR T3	
1081	F565	FF	3006	STX		;\$3006 ;STORE TO T3 DATA REG	
1082	F568	86	01	LDA	A	;\$01 ;SET MON CONN FLAG VALUE	
1083	F56A	97	51	STA	A	MONFG ;STORE FLAG, MON. CONNECTED	
1084	F56C	97	E6	STA	A	NORFG ;SET NORMAL OP WITH KEYBD	
1085	F56E	97	E7	STA	A	V\$FLG ;SET VSTAR TEST FLAG	
1086	F570	7E	F018	JMP		START ;JUMP TO MON START	
1087			F595	ORG		;\$F595	
1088	F595	7E	F3FA	JBAD	JMP	PBADR ;PRNT "BAD ADDRESS" & QUIT	
1089			F5A2	ORG		;\$F5A2	
1090						*	
1091						* MEMORY MOVE	
1092						*	
1093	F5A2	8D	F185	MOVE	JSR	GETRNG ;GET SOURCE ADDRESS RANGE	
1094	F5A5	8D	F2D6		JSR	NXTADR ;GET DEST STARTING ADD	
1095	F5A8	27	EB		BEQ	JBAD ;ERROR IF DONE	
1096	F5AA	FE	FFDC	AM	LDX	ADDL ;GET BYTE	
1097	F5AD	A6	00		LDA	A	;\$00
1098	F5AF	FE	FFDA		LDX	ADR ;STORE IT WITH VERIFY	
1099	F5B2	8D	F3EA		JSR	SETMEN	
1100	F5B5	08			INX	;INCREMENT POINTERS	

SEQ	LOC	OBJ	SOURCE
1101	F5B6	FF FF0A	STX ADR
1102	F5B9	BD F4F6	JSR INK ; COMPARE TO END
1103	F5BC	20 EC	BRA AM ; MORE
1104		*	
1105		*	END OF MODULE
1106		*	
1107		0018 NITEMS EQU 24	; NUMBER OF ROUTINES
1108		*	
1109		*	CALLING SEQUENCE LOC
1110		*	X SWI
1111		*	X+1 INDEX
1112		*	X+2 NEXT INSTRUCTION
1113	F5BE	ORG \$F5BE	
1114		*	
1115		*	ENTRY IS VIA LOW ORDER ADDRESS OF ROM
1116		*	<<ADDRESS IS PLACED IN SWI VECTOR ADDRESS
1117	F5BE	RSRSR EQU *	
1118		*	
1119		*	GET THE INDEX VALUE
1120		*	DOUBLE IT FOR VECTOR ADDRESS INDEX
1121		*	
1122	F5BE	30	TSX ; SP INTO X
1123		*	RESTORE STATE OF INTERRUPT AT TIME OF CALL
1124		*	
1125	F5BF	EE 05	LDX 5,X ; X HAS INDEX ADDRESS
1126		*	
1127	F5C1	4F	CLR A
1128	F5C2	E6 00	LDA B 0,X ; INDEX INTO B
1129	F5C4	58	ASL B ; DOUBLE
1130	F5C5	45	ROL A
1131		*	
1132		*	A,B HAS TWO TIMES INDEX
1133		*	
1134		*	VECTOR OF SUBROUTINE ADDRESSES IS AT 512+ROM BASE
1135		*	
1136		*	FROM HERE TO VECTOR IS 512-WHERE WE ARE
1137	F5C6	8D 00	BSR LDCVV
1138		*	
1139		*	
1140		F5C8 LDCVV EQU *	; STACK HAS WHERE WE ARE
1141	F5C8	30	TSX
1142		*	A,B WILL HAVE INDEX*2 + LOCATION(AR)
1143	F5C9	EB 01	ADD B 1,X
1144	F5CB	A9 00	ADD A 0,X
1145		*	ADD VECTOR OFFSET
1146		*	
1147	F5CD	C8 24	ADD B #24 ; LOW ORDER 8 BITS
1148	F5CF	89 01	ADD A #01 ; HIGH ORDER 8 BITS
1149		*	
1150		*	A,B NOW HAVE ADDRESS OF SUBROUTINE ADDRESS
1151		*	
1152	F5D1	A7 00	STA A 0,X ; SAVE VECTOR ADDRESS, HIGH
1153	F5D3	E7 01	STA B 1,X ; SAVE VECTOR ADDRESS, LOW
1154	F5D5	EE 00	LDX 0,X ; LOAD VECTOR ADDRESS INTO X
1155	F5D7	E8 01	ADD B 1,X

SEQ	LOC	OBJ	SOURCE
1156	F5D9	A9 00	ADD A 0,X
1157			* ADD IN OFFSET CONTAINED IN VECTOR TABLE
1158	F5DB	30	TSX
1159	F5DC	A7 00	STA A 0,X
1160	F5DE	E7 01	STA B 1,X
1161	F5E0	A6 02	LDA A 2,X
1162	F5E2	06	TAP
1163	F5E3	EE 00	LDX 0,X
1164			* STORE OLD STATE INTO CC
1165	F5E5	31	INS
1166	F5E6	31	INS
1167			* CORRECT SP
1168	F5E7	AD 00	JSR 0,X
1169			* JUMP TO SUBROUTINE
1170			* NORMAL EXIT FROM SUBROUTINE
1171			* INCREMENT RETURN ADDRESS
1172	F5E9	30	TSX
1173	F5EA	6C 06	INC 6,X
1174	F5EC	26 02	BNE **4
1175	F5EE	6C 06	INC 6,X
1176			* EXIT
1177	F5F0	3B	RTI
1178			* STACK ELEMENTS ARE STACK POINTER +2 SINCE JSR
1179			*
1181		0002 UC	EQU 2
1182		0003 UB	EQU 3
1183		0004 UA	EQU 4
1184		0005 UXH	EQU 5
1185		0006 UXL	EQU 6
1186		0007 URH	EQU 7
1187		0008 URL	EQU 8
1188			* CC RELATIVE TO SP
1189			* B RELATIVE TO SP
1190			* A RELATIVE TO SP
1191			* XH RELATIVE TO SP
1192			* XL RELATIVE TO SP
1193			* RH RELATIVE TO SP
1194			* RL RELATIVE TO SP
1195			*
1196			* PUSH ALL ONTO STACK -- REGISTERS CORRECT ON EXIT
1197			*
1198			*
1199			*
1200			*
1201			*
1202			*
1203			*
1204			*
1205	F5F1	34	DES
1206	F5F2	34	DES
1207	F5F3	34	DES
1208	F5F4	34	DES
1209	F5F5	34	DES
1210			*

SEQ	LOC	OBJ	SOURCE
1211			* MOVE STACK DOWN
1212			*
1213	F5F6	06 09	LDA B #9 ; NINE BYTES DOWN
1214	F5F8	30	TSX
1215	F5F9	A6 05	AS LDA A 5,X ; OFFSET OF 5
1216	F5FB	A7 00	STA A 0,X
1217	F5FD	08	INX
1218	F5FE	5A	DEC B
1219	F5FF	26 F8	BNE AS
1220			*
1221			* RECOPY "PUSHED" REGISTERS
1222			*
1223	F601	06 05	LDA B #5 ; FIVE BYTES TO MOVE
1224	F603	30	TSX
1225	F604	A6 02	AC LDA A UC,X
1226	F606	A7 09	STA A UC+7,X ; OFFSET BY 7
1227	F608	08	INX
1228	F609	5A	DEC B
1229	F60A	26 F8	BNE AC
1230			*
1231			* EXIT TO MAIN
1232			*
1233	F60C	39	RTS
1234			*
1235			* USER STACK IS
1236			CC B A XH XL
1237			SP
1238			*
1239			* POP ALL REGISTERS
1240			*
1241		F60D	POPALL EQU *
1242	F60D	30	TSX
1243			* CURRENT STACK
1244			* SRH SRL CC B A XH XL URH URL CC B A XH XL
1245			* RESULT STACK
1246			SRH SRL CC B A XH XL URH URL
1247			*
1248			* RECOPY "PULLED" REGISTERS
1249			*
1250	F60E	06 05	LDA B #5 ; FIVE OF THEM
1251	F610	A6 09	PC LDA A UC+7,X ; OFFSET BY 7
1252	F612	A7 02	STA A UC,X
1253	F614	08	INX
1254	F615	5A	DEC B
1255	F616	26 F8	BNE PC
1256			*
1257			* SHIFT EVERYTHING OVER
1258			*
1259	F618	06 09	LDA B #9 ; NINE BYTES
1260	F61A	A6 03	PS LDA A URL-5,X
1261	F61C	A7 08	STA A URL,X ; OFFSET 5
1262	F61E	09	DEX
1263	F61F	5A	DEC B
1264	F620	26 F8	BNE PS
1265			*

SEQ	LOC	OBJ	SOURCE
1266			* FINALLY INCREMENT SP
1267			*
1268	F622	31	INS
1269	F623	31	INS
1270	F624	31	INS
1271	F625	31	INS
1272	F626	31	INS
1273	F627	39	RTS
1274			*
1275			* TRANSFER X TO A,B
1276			*
1277	F628	30	TXAB TSX
1278	F629	A6 05	LDA A UXH,X ;X HIGH
1279	F62B	E6 06	LDA B UXL,X ;X LOW
1280	F62D	A7 04	STAB STA A UA,X ;TO A
1281	F62F	E7 03	STA B UB,X ;TO B
1282			*
1283	F631	39	RTS
1284			*
1285			* TRANSFER A,B TO X
1286			*
1287	F632	30	TABX TSX
1288	F633	A6 04	LDA A UA,X ;A
1289	F635	A7 05	STA A UXH,X ;TO X HIGH
1290			*
1291	F637	A6 03	LDA A UB,X ;B
1292	F639	E7 06	STA B UXL,X ;TO X LOW
1293			*
1294	F63B	39	RTS
1295			*
1296			* EXCHANGE A,B AND X
1297			*
1298		F63C	XABX EQU *
1299	F63C	30	TSX
1300			* CURRENT STACK
1301			* SRH SRL C B A XH XL URH URL
1302			* RESULT XL XH A B
1303			*
1304	F63D	A6 05	LDA A UXH,X ;PICK UP UX
1305	F63F	36	PSH A
1306	F640	E6 06	LDA B UXL,X
1307	F642	8D EF	BSR TABX+1 ;THEN GO TO XFER A,B TO X
1308	F644	32	PUL A
1309	F645	20 E6	BRA STAB ;TO STORE IN A,B
1310			*
1311			* PUSH X
1312			*
1313		F647	PUSX EQU *
1314			*
1315			* GET SPACE IN SPACE
1316			*
1317	F647	34	DES
1318	F648	34	DES
1319	F649	30	TSX
1320			* MOVE STACK DOWN TWO

SEQ	LOC	OBJ	SOURCE
1321	F64A	86 09	LDA A #9 ; MOVE TOTAL OF 9 BYTES
1322			*
1323	F64C	E6 02	SA LDA B 2,X
1324	F64E	E7 00	STA B 0,X
1325	F650	08	INX
1326	F651	4A	DEC A
1327	F652	26 F8	BNE SA
1328			*
1329			* STACK MOVED -- INSERT X
1330	F654	30	TSX
1331	F655	A6 05	LDA A UXH,X
1332	F657	A7 09	STA A UXH+4,X
1333	F659	A6 06	LDA A UKL,X
1334	F65B	A7 0A	STA A UKL+4,X
1335			*
1336	F65D	39	RTS
1337			* STACK ON RETURN
1338			* SRH SRL C B A XH XL URH URL XH XL
1339			* SP
1340			*
1341			* PULL X
1342	F65E	PULX EQU *	
1343			* GET X FROM STACK
1344			*
1345	F65E	30	TSX
1346	F65F	A6 09	LDA A UXH+4,X ; CURRENT X ON STACK
1347	F661	A7 05	STA A UXH,X ; REG X
1348	F663	A6 0A	LDA A UKL+4,X
1349	F665	A7 06	STA A UKL,X
1350			*
1351			* NOW MOVE UP TWO
1352	F667	86 09	LDA A #9 ; BYTE COUNT
1353		F669 PA	EQU *
1354	F669	E6 08	LDA B 8,X
1355	F66B	E7 0A	STA B 10,X
1356	F66D	09	DEX
1357	F66E	4A	DEC A
1358	F66F	26 F8	BNE PA
1359			* UPDATE SP
1360	F671	31	INS
1361	F672	31	INS
1362			*
1363	F673	39	RTS
1364			*
1365			* ADD X TO A,B
1366			*
1367		F674	ADDXAB EQU *
1368	F674	30	TSX
1369	F675	8D 06	BSR XABX+1 ; EASY WAY EXCHANGE AB & X
1370	F677	8D 03	BSR ADDABX+1 ; ADD OTHER WAY
1371	F679	2D 02	BSR XABX+1 ; THEN EXCHANGE BACK
1372			*
1373			* ADD A,B TO X
1374			*
1375	F67B	30	ADDABX TSX

SEQ	LOC	OBJ	SOURCE
1376	F67C	A6 03	LDA A UB,X
1377	F67E	E6 04	LDA B UA,X
1378			*
1379			* CODE SHARED BY ADDAX, INDEX
1380			*
1381		F680	ADDAB EQU *
1382	F680	A6 06	ADD A UXL,X ;ADD UXL TO UB
1383	F682	A7 06	STA A UXL,X ;STORE INTO UXL
1384			*
1385	F684	E9 05	ADC B UXH,X ;ADD UXH TO UA
1386	F686	07	STAUXH ;SAVE STATUS
1387	F687	E7 05	STA B UXH,X ;STORE IN UXH
1388			*
1389	F689	6D 06	TST UXL,X ;TEST LOW BYTE FOR ZERO
1390			*
1391			* CODE SHARED BY ADDABX, MUL8, MUL16
1392			*
1393		F68B	TESTZ EQU *
1394	F68B	27 02	BEQ TA ;HIGH BY STAT IS TRUE RSLT
1395	F68D	84 FB	AND A ##FB ;NO, Z BIT IS CLEARED
1396	F68F	A7 02	STA A UC,X ;SAVE STATUS
1397			*
1398	F691	39	RTS
1399			*
1400			* ADD A TO X
1401			*
1402		F692	ADDAX EQU *
1403	F692	30	TSX
1404	F693	A6 04	LDA A UA,X
1405		F695	ADDZ EQU *
1406	F695	06 00	LDA B #0
1407	F697	20 E7	BRA ADDAB
1408			* ADD B TO X
1409		F699	ADDBX EQU *
1410	F699	30	TSX
1411	F69A	A6 03	LDA A UB,X
1412	F69C	20 F7	BRA ADDZ
1413			*
1414			* SUBTRACT X FROM A, B
1415			*
1416	F69E	30	SUBXAB TSX
1417	F69F	8D 9C	BSR XABX+1
1418	F6A1	8D 03	BSR SUBABX+1
1419	F6A3	20 98	BRA XABX+1
1420			*
1421			* SUBTRACT A, B FROM X
1422			*
1423		F6A5	SUBABX EQU *
1424	F6A5	30	TSX
1425	F6A6	E6 05	LDA B UXH,X
1426	F6A8	A6 06	LDA A UXL,X
1427			*
1428	F6AA	A0 03	SUB A UB,X
1429	F6AC	A7 06	STA A UXL,X
1430			*

SEQ	LOC	OBJ	SOURCE
1431	F6AE	E2 04	SBC B UA,X
1432	F6B0	20 D4	BRA STAUXH
1433			*
1434			* SUBTRACT A FROM X
1435			*
1436		F6B2	SUBAX EQU *
1437	F6B2	30	TSX
1438	F6B3	E6 04	LDA B UA,X
1439	F6B5	A6 06	SSUB LDA A UXL,X
1440	F6B7	10	SBA
1441	F6B8	A7 06	STA A UXL,X
1442			* STORE XL
1443	F6BA	E6 05	LDA B UXH,X
1444	F6BC	C2 00	SBC B #0
1445	F6BE	20 C6	BRA STAUXH
1446			*
1447			* SUB B FROM X
1448			*
1449		F6C0	SUBBX EQU *
1450	F6C0	30	TSX
1451	F6C1	E6 03	LDA B UB,X
1452	F6C3	20 F0	BRA SSUB
1453			*
1454			* INDEX: X:=X+A+B (SAVE USER A,B)
1455			*
1456		F6C5	INDEX EQU *
1457	F6C5	80 11	BSR MPY8
1458			* A,B := USERA+USERB
1459			*
1460			* EXCHANGE A&B TO SHARE CODE WITH ADDABX
1461	F6C7	37	PSH B
1462	F6C8	16	TAB
1463	F6C9	32	PUL A
1464	F6CA	30	TSX
1465	F6CB	20 B3	BRA ADDAB
1466			*
1467			* MUL8: A,B := A*B
1468			*
1469		F6CD	MUL8 EQU *
1470	F6CD	80 09	BSR MPY8
1471	F6CF	30	TSX
1472	F6D0	E7 03	STA B UB,X
1473	F6D2	A7 04	STA A UA,X
1474	F6D4	07	TPA
1475	F6D5	50	TST B
1476	F6D6	20 B3	JMPTZ BRA TESTZ
1477		0800	ORG #0800
1478			* NOT INCLUDED IN MONITOR
1479			* MUL16--16 BIT MULTIPLY
1480			* A,B,X := A,B*X
1481			* A,B = PARTIAL PRODUCT
1482			* USERX = MULTIPLIER
1483			* USERA,USERB = MULTIPLICAND
1484			*
1485	0800	86 10	MUL16 LDA A #16
			* PUSH COUNTER INTO STACK

SEQ	LUC	OBJ	SOURCE
1486	0802	36	PSH A
1487	0803	30	TSX
1488	0804	4F	CLR A
1489	0805	5F	CLR B
1490	0806	66 06	ROR UXH+1,X
1491	0808	66 07	ROR UXL+1,X
1492			*
1493			* LOOP 16 TIMES
1494			*
1495	080A	24 04	MLOOP BCC MSHIFT
1496	080C	EB 04	ADD B UB+1,X
1497	080E	A9 00	ADC A UA+1,X
1498			*
1499	0810	46	MSHIFT ROR A
1500	0811	56	ROR B
1501	0812	66 06	ROR UXH+1,X
1502	0814	66 07	ROR UXL+1,X
1503	0816	6A 00	DEC 0,X
1504	0818	26 F0	BNE MLOOP
1505			*
1506			* END LOOP
1507			*
1508	081A	31	INS
1509	081B	30	TSX
1510	081C	E7 03	STA B UB,X
1511	081E	A7 04	STA A UA,X
1512			*
1513			* SET USER CC:= N=N(MSBYTE)
1514			* Z:=AND (Z(MSBYTE),Z(LSBYTE)):
1515			* V:= 0
1516			* CRY := 0
1517			* THE LAST ADD RTESET CRY. STA SET N=N(MSBYTE) & V=0
1518			*
1519	0820	07	TPA
1520	0821	EA 05	ORA B UXH,X
1521	0823	EA 06	ORA B UXL,X
1522			*
1523			* USER CC HAS CORRECT N,V,&C. CC HAS CORRECT Z FOR LS BYTE
1524			* GO TO END OF ADDXAB TO UPDATE REGISTERS
1525			*
1526	0825	7E F6D6	JMP JMPTZ
1527		F6D8	ORG #F6D8
1528			*
1529			* SUBROUTINE MPY8. A,B:= USERA*USERB
1530			* A = PARTIAL PRODUCT
1531			* B = MULTIPLIER & LSB'S OF PAR. PROD.
1532			* USERA = MULTIPLICAND
1533			*
1534	F6D8	86 08	MPY8 LDA A #8
1535	F6DA	36	PSH A
1536			* STACK = COUNT, R, R, R, R, C, B, A, X, X, R, R
1537	F6DB	4F	CLR A
1538	F6DD	30	TSX
1539	F6DD	E6 06	LDA B UB+3,X
1540	F6DF	36	ROR B

```

SEQ  LOC  OBJ  SOURCE

1541      *
1542      *  LOOP 8 TIMES
1543      *
1544 F6E0 24 02 BLOOP  BCC  BSHIFT  ; MULTIPLIER IS EVEN
1545 F6E2 AB 07      ADD  A  UA+3,X
1546      *
1547 F6E4 46      BSHIFT  ROR  A  ; SHIFT LSB OF A INTO B
1548 F6E5 56      ROR  B
1549 F6E6 5A 00      DEC  0,X  ; CHECK COUNT
1550 F6E8 26 F6      BNE  BLOOP
1551      *
1552      *  END OF LOOP
1553      *
1554 F6EA 31      INS      ; RESTORE SP
1555 F6EB 39      RTS
1556      *
1557      *  RELATIVE ENTRY POINTS TO SUBROUTINE VECTOR
1558      *
1559      F6EC SVECTO  EQU  *
1560      *
1561 F6EC FF 05      FDB  $FF05  ; PUSHALL--*  0
1562 F6EE FF 1F      FDB  $FF1F  ; POPALL--*  1
1563 F6F0 FF 38      FDB  $FF38  ; TXAB--*  2
1564 F6F2 FF 40      FDB  $FF40  ; TABX--*  3
1565 F6F4 FF 48      FDB  $FF48  ; XABX--*  4
1566 F6F6 FF 51      FDB  $FF51  ; PUSX--*  5
1567 F6F8 FF 66      FDB  $FF66  ; PULX--*  6
1568 F6FA FF 7A      FDB  $FF7A  ; ADDXAB--*  7
1569 F6FC FF 7F      FDB  $FF7F  ; ADDABX--*  8
1570 F6FE FF 94      FDB  $FF94  ; ADDAX--*  9
1571 F700 FF 99      FDB  $FF99  ; ADDBX--* 10
1572 F702 FF 9C      FDB  $FF9C  ; SUBXAB--* 11
1573 F704 FF A1      FDB  $FFA1  ; SUBABX--* 12
1574 F706 FF AC      FDB  $FFAC  ; SUBAX--* 13
1575 F708 FF B8      FDB  $FFB8  ; SUBBX--* 14
1576 F70A 00 17      FDB  $0017  ; P2HEX--* 15
1577 F70C 00 10      FDB  $0010  ; P4HEX--* 16
1578 F70E 00 2F      FDB  $002F  ; PRINTA--* 17
1579 F710 00 4F      FDB  $004F  ; PMESS--* 18
1580 F712 00 5D      FDB  $005D  ; VALAN--* 19
1581 F714 00 86      FDB  $0086  ; INPUTA--* 20
1582 F716 00 96      FDB  $0096  ; CONHB--* 21
1583 F718 FF AD      FDB  $FFAD  ; INDEX--* 22
1584 F71A FF B3      FDB  $FFB3  ; MUL8--* 23
1585      *
1586      *  PRINT 2/4 HEX CHARS FROM MEM(UX),UX+1
1587      *  UX IS INCREMENTED UPON OUTPUT I.E. UX = UX+2
1588      *
1589      F71C P4HEX  EQU  *
1590 F71C 30      TSX
1591 F71D EE 05      LDY  UX,X  ; USERS X
1592 F71F 8D 06      BSR  PHEX  ; PRINT MEMO)
1593      *
1594      *  PRINT 2 HEX CHARS FROM MEM(UX)
1595      F721 P2HEX  EQU  *

```

SEQ	LOC	OBJ	SOURCE
1596	F721	30	TSX
1597	F722	EE 05	LDX UXH,X ;USERS X
1598	F724	80 01	BSR PHEX ;PRINT MEM(X)
1599	F726	39	RTS
1600			*
1601			* PRINT 2 HEX CHARS FROM MEM(X)
1602			*
1603		F727	PHEX EQU *
1604	F727	A6 00	LDA A 0,X ;GET THE CHAR
1605	F729	80 29	BSR ASCIIR ;CONV RIGHT NIB & RSLT IN A
1606	F72B	36	PSH A ;SAVE IT
1607	F72C	A6 00	LDA A 0,X ;GET CHAR AGAIN
1608	F72E	80 20	BSR ASCIIL ;CONV THE LEFT NIBBLE INTO A
1609	F730	80 0E	BSR PUTAX ;PRINT A REG CHAR
1610	F732	32	PUL A ;RECOVER SAVED
1611	F733	80 10	BSR PUTA ;... THEN FALL INTO PINCX
1612			*
1613			* INCREMENT THE USER'S X IN THE STACK
1614			*
1615		F735	PINCX EQU *
1616	F735	30	TSX ;SP IS +2 SINCE 2 BSR DOWN IN CA S
1617	F736	60 08	INC UXL+2,X ;INC MEMORY X LOW
1618	F738	26 02	BNE PRTS ;OVERFLOW MEANS INC HIGH PART
1619	F73A	60 07	INC UXH+2,X ;YES--INC HIGH
1620	F73C	39	PRTS RTS ;EXIT
1621			*
1622			* PRINT THE CHAR IN USERS A
1623			*
1624		F73D	PRINTA EQU *
1625	F73D	30	TSX
1626	F73E	A6 04	LDA A UA,X ;GET THE CHAR
1627			*
1628			* PRINT CHAR IN DESIGNATED REG
1629			* ACIA ADDRESS IN X
1630			*
1631			* PRINT CHAR IN A
1632	F740	FE FFF6	PUTAX LDX ACIAI ;GET INDIR ADDR OF ACIA
1633	F743	EE 00	LDX 0,X ;GET ACTUAL ADD OF ACIA IN X
1634	F745	36	PUTA PSH A ;SAVE REGISTER
1635	F746	A6 00	PRDY LDA A 0,X ;ACIA STATUS
1636	F748	85 02	BIT A #02 ;READY?
1637	F74A	27 FA	BEQ PRDY ;NOT READY
1638	F74C	32	PUL A ;RESTORE CHAR
1639	F74D	A7 01	STA A 1,X ;PRINT CHAR
1640	F74F	39	RTS
1641			*
1642			* CONVERT A FROM BINARY TO HEXIN LEFT/RIGHT NIBBLE
1643			* LEFT PART
1644			*
1645		F750	ASCII L EQU *
1646	F750	44	LSR A ;A HAS CHAR TO BE CONVERTED
1647	F751	44	LSR A
1648	F752	44	LSR A
1649	F753	44	LSR A
1650			* IN POSITION

SEQ	LOC	OBJ	SOURCE
1651	F754	84 0F	ASCIIIR AND A ##0F ; CLEAR LEFT PART
1652	F756	88 30	ADD A ##30
1653	F758	81 39	CMP A ##39 ; 0 TO 9
1654	F75A	23 02	BLS ARTS ; YES DONE
1655	F75C	88 07	ADD A #7 ; NO THEN A TO F
1656	F75E	39	ARTS RTS
1657			*
1658			* PRINT MESSAGE POINTED TO BY X AND TERMINATED BY ETX
1659			*
1660		F75F	PMESS EQU *
1661		0004	ETX EQU #04
1662	F75F	30	TSX
1663	F760	EE 05	LDX UXH,X ; GET USERS X
1664	F762	A6 00	LDA A 0,X ; GET CHAR
1665	F764	81 04	CMP A #ETX ; IS IT TERMINATOR?
1666	F766	27 06	BEQ TRTS ; DONE
1667	F768	80 D6	BSR PUTAX ; PRINT A
1668	F76A	80 C9	BSR PINCX ; INC USERS X
1669	F76C	20 F1	BRA PMESS ; LOOP TILL DONE
1670	F76E	39	TRTS RTS
1671			*
1672			* X HAS ADDRESS OF CHAR TO BE TESTED
1673			* FOR BEING ALPHA NUMERIC
1674			* CARRY SET IF TRUE
1675		F76F	VALAN EQU *
1676	F76F	30	TSX
1677	F770	EE 05	LDX UXH,X ; GET CHAR ADDR
1678	F772	80 05	BSR ALPNUM ; TEST MEM(X)=ALPHANUMERIC
1679			*
1680			* SET USER'S CARRY = CURRENT CARRY (AND OTHER FLAGS)
1681			*
1682	F774	07	SCARRY TPA
1683			*
1684	F775	30	SETUS TSX
1685	F776	A7 02	STA A UC,X
1686	F778	39	RTS
1687			*
1688			* SET CARRY IF MEM(X) IS ALPHANUMERIC
1689			* CLEAR V IF HEX DIGIT
1690			*
1691		F779	ALPNUM EQU *
1692	F779	A6 00	LDA A 0,X ; GET THE CHAR
1693	F77B	81 41	CMP A #'A
1694	F77D	2D 0E	BLT ANUM ; TOO SMALL FOR ALPHA, = NUM
1695	F77F	81 5A	CMP A #'Z
1696	F781	2E 12	BGT ANOTOK ; BIGGER THAN "Z"
1697	F783	81 C7	CMP A ##C7 ; SET V IF >F
1698	F785	29 10	BVS NRTS ; QUIT IF NOT HEX (C=1)
1699	F787	80 07	SUB A #7 ; CONVERT LETTER TO HEX
1700	F789	84 0F	AND A #15 ; STRIP OVERBITS FROM HEX DIGIT
1701	F78B	0D	SEC ; SET C FOR VALID A/N
1702	F78C	39	RTS
1703			*
1704	F78D	81 30	ANUM CMP A #'0 ; NUMERIC TESTING
1705	F78F	2D 04	BLT ANOTOK ; NOT NUMERIC

SEQ	LOC	OBJ	SOURCE
1706	F791	81 39	CMP A #79
1707	F793	2F F4	BLE AOK ; IT IS IN 0-9
1708	F795	0C	ANOTOK ; RESET CARRY FOR NOT A/N
1709	F796	08	SEV ; SET V FOR NOT HEX EITHER
1710	F797	39	NRTS RTS
1711			*
1712	F798	20 9B	JPINCK BRA PINCK ; EXTRA BRA TO REACH PINCK
1713			*
1714			* INPUTA:
1715			* INPUT ACIA DATA INTO A REG
1716			* STRIP PARITY
1717			*
1718		F79A	INPUTA EQU *
1719	F79A	FE FFF6	LDX ACIAI ; GET INDIRECT ADDRESS
1720	F79D	EE 00	LDX 0,X ; GET ACIA ADDRESS
1721			*
1722	F79F	A6 00	IWAIT LDA A 0,X ; ACIA STATUS
1723	F7A1	47	ASR A ; CARRY:= RDRF
1724	F7A2	24 FB	BCC IWAIT ; NO INPUT, LOOP
1725			*
1726	F7A4	A6 01	LDA A 1,X ; ACIA DATA
1727	F7A6	84 7F	AND A #7F ; STRIP PARITY
1728	F7A8	30	TSX ; PUT RESULT ONTO STACK
1729	F7A9	A7 04	STA A 0A,X
1730	F7AB	39	RTS
1731			*
1732			* CONHB--CONVERT HEX TO BINARY:
1733			* SCAN UP TO B ASCII CHARS STARTING AT X
1734			* LOOKING FOR A VALID HEX NUMBER. RETURN BINARY
1735			* EQUIVALENT OF NUMBER IN A,B. IF NUMBER HAS MORE
1736			* THAN 16 BITS, IGNORE MSB'S.
1737			*
1738			* INPUT: X= ADDRESS OF 1ST CHAR TO BE SCANNED
1739			* B= MAX. # OF CHARS TO BE SCANNED
1740			*
1741			* OUTPUT: A,B=BINARY RESULT
1742			* CARRY=1 IF VALID NUMBER IS FOUND
1743			* X POINTS TO LAST CHAR SCANNED
1744			*
1745		F7AC	CONHB EQU *
1746	F7AC	30	TSX
1747	F7AD	E6 03	LDA B 0B,X ; GET MAX COUNT
1748	F7AF	6F 04	CLR 0A,X ; CLEAR USERS A,B REGS
1749	F7B1	6F 03	CLR 0B,X
1750			*
1751			* LOOP WHILE NOT ALPHANUMERIC AND COUNT > 1
1752			*
1753	F7B3	30	CLOOP1 TSX
1754	F7B4	EE 05	LDX 0A,X ; GET CHAR ADDRESS
1755	F7B6	80 C1	BSR ALPNUM ; IS MEM(X) A/N?
1756	F7B8	25 09	BOS CFOUND ; YES, STOP SCANNING
1757	F7BA	5A	DEC B ; DEC COUNT
1758	F7BB	2F 04	BLE ENDCNT ; COUNT EXHAUSTED
1759	F7BD	80 D9	BSR JPINCK ; INC USERS X
1760	F7BF	20 F2	BRA CLOOP1

```

SEQ  LOC  OBJ  SOURCE
1761      *
1762      *  END LOOP
1763      *
1764      *  COUNT EXHAUSTED WITH NO SUCCESS.
1765      *  (CARRY WAS RESET BY ALPNUM)
1766      *
1767 F7C1 20 B1  ENDCNT  BRA      SCARRY      ; RESET USER C AND RETURN
1768      *
1769      *  WHILE HEX AND COUNT >0 SHIFT MEM(X) INTO UA,UB
1770      *
1771      *  BEGIN OUTER LOOP
1772 F7C3 30      CFOUND  TSX
1773 F7C4 EE 05      LDX      UXH,X
1774 F7C6 80 B1      BSR      ALPNUM      ; CNVT MEM(X) TO HEX
1775 F7C8 29 18      BVS      NOGOOD      ; INVALID CHAR
1776 F7CA 37      PSH B      ; SAVE COUNT
1777 F7CB C6 04      LDA B      #4      ; LOOP COUNT
1778      *
1779      *  SHIFT LEFT UA,UB
1780      *
1781 F7CD 30      TSX
1782 F7CE 68 04  CSLLOOP ASL      UB+1,X      ; +1 TO COMP. FOR PUSH
1783 F7D0 69 05      ROL      UA+1,X
1784 F7D2 5A      DEC B
1785 F7D3 2E F9      BGT      CSLLOOP
1786      *
1787 F7D5 AA 04      ORA A      UB+1,X      ; OR' ON NEW CHAR
1788 F7D7 A7 04      STA A      UB+1,X
1789 F7D9 33      PUL B      ; RETRIEVE COUNT
1790 F7DA 8D BC      BSR      JPINX      ; INC USER X
1791 F7DC 5A      DEC B
1792 F7DD 2E E4      BGT      CFOUND      ; REPEAT
1793      *  END OF OUTER LOOP
1794 F7DF 00      SEC      ; VALID NUMBER
1795 F7E0 20 92      BRA      SCARRY      ; SET USER C AND RETURN
1796      *
1797      *  NON-HEX CHAR FOUND. IF CHAR=G-Z, THIS IS NOT A VALID
1798      *  HEX NUMBER OTHERWISE, CHAR IS A DELIMITER AND
1799      *  NUMBER IS VALID.
1800      *
1801 F7E2 07      NOGOOD  TPA      ; TOGGLE CARRY BIT
1802 F7E3 4C      INC A
1803 F7E4 20 8F      BRA      SETUS      ; SETUP USER STATUS & RETURN
1804      *
1805      END

```

1805 LINES ASSEMBLED, LOC = F7E6, 0000 ERRORS DETECTED.

SYMBOL	VALUE	ATTR	LOCN	LINK
ACIAC	6000	84	2AA2	
ACIAD	6001	84	2AA8	
ACIAS	6000	84	2AB4	
BLANK	0020	84	2ABD	
CR	000D	84	2AC6	
ESC	001B	84	2ACF	
EOT	0004	84	2AD8	
LAST	FFFF	84	2AE1	
LF	000A	84	2AEA	
RUBOUT	007F	84	2AF3	
MONFG	0051	84	2AFC	
NDRFG	00E6	84	2B05	
VSFLG	00E7	84	2B0C	
BASE	FF90	84	2B17	
BOS	FF8F	84	2B20	
BUF	FF90	84	2B29	
PROCRD	FFD8	84	2B32	
OFFSET	FFD8	84	2B3B	
ADR	FFDA	84	2B44	
ADDL	FFDC	84	2B4D	
ADDH	FFDE	84	2B56	
BUFPTR	FFE0	84	2B5F	
RECTYP	FFE2	84	2B68	
COUNT	FFE3	84	2B71	
CKSM	FFE4	84	2B7A	
SAVEBP	FFE5	84	2B83	
SAVEK	FFE7	84	2B8C	
ECHO	FFE9	84	2B95	
TCOUNT	FFEA	84	2B9E	
CREG	FFEB	84	2BA7	
BREG	FFEC	84	2BB0	
AREG	FFED	84	2BB9	
XREG	FFEE	84	2BC2	
PREG	FFF0	84	2BCB	
SREG	FFF2	84	2BD4	
USWI	FFF4	84	2BD0	
ACIAI	FFF6	84	2BE6	
IRQVEC	FFF8	84	2BEF	
SWIVEC	FFFA	84	2BF8	
NMIVEC	FFFC	84	2C01	
PWRI	F51C	84	2C0A	
START	F018	84	2C13	
START1	F01F	84	2C1C	
BREAK	F01A	84	2C25	
BREAK1	F0CF	84	2C2E	
ACIAR	F01D	84	2C37	
SWISO	F0E9	84	2C40	
SWIHAN	F0D6	84	2C49	
MONENT	F059	84	2C52	
MONEN1	F059	84	2C5B	
FINIT	F434	84	2C64	
PCRLF	F322	84	2C6D	
MONITR	F00F	84	2C76	
RDRDFF	F3B9	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
OUTCH	F2FB	84	2088	
RT10	F077	84	2091	
RT20	F07E	84	209A	
ABORT	F005	84	20A3	
WAITTY	F41E	84	20AC	
RT30	F081	84	20B5	
RT90	F084	84	20BE	
PXISTS	F3A3	84	20C7	
CTABLE	F0A4	84	20D0	
DL00F	F092	84	20D9	
DL10	F09A	84	20E2	
CTEND	F005	84	20EB	
MONEND	F05F	84	20F4	
LOAD	F1CB	84	20FD	
GO	F1B7	84	2D06	
PUNCH	F326	84	2D0F	
PREGS	F106	84	2D18	
MOVE	F5A2	84	2D21	
VFY	F403	84	2D2A	
READ	F4E0	84	2D33	
SM	F401	84	2D3C	
DM	F14E	84	2D45	
EOF	F175	84	2D4E	
BADINF	F005	84	2D57	
MOUES	F291	84	2D60	
MSGMON	F008	84	2D69	
MSGABT	F008	84	2D72	
SWI40	F0F0	84	2D7B	
SWI20	F0E4	84	2D84	
R5R5R	F5BE	84	2D8D	
SWI50	F0F3	84	2D96	
PR1	F10B	84	2D9F	
RESTAK	F124	84	2DA8	
PR10	F110	84	2DB1	
PSPACE	F39E	84	2DBA	
P4HEX5	F39A	84	2DC3	
PR20	F11A	84	2DCC	
RUS10	F12A	84	2DD5	
CHEKSM	F134	84	2DDE	
NEXT2D	F2BC	84	2DE7	
CS1	F141	84	2DF0	
MO5ER	F296	84	2DF9	
GETRNG	F185	84	2E02	
DM10	F150	84	2E0B	
DM20	F156	84	2E14	
DM50	F172	84	2E1D	
MPEOF	F2A8	84	2E26	
NULL5	F17A	84	2E2F	
NULL1	F17C	84	2E38	
NXTADR	F2D6	84	2E41	
GETRG3	F19C	84	2E4A	
GETRG1	F196	84	2E53	
GETRG4	F1AF	84	2E5C	
MRNGER	F283	84	2E65	

SYMBOL	VALUE	ATTR	LOCN	LINK
RNGERR	F1A9	84	2E6E	
G10	F1C8	84	2E77	
LOOFST	F1D4	84	2E80	
LHF2	F1FB	84	2E89	
RDRON	F3C3	84	2E92	
FINDS	F270	84	2E9B	
RDPRE	F1FE	84	2EA4	
LHF3	F231	84	2EAD	
LHF4	F249	84	2EB6	
LDR10	F235	84	2EBF	
SETOFF	F3C0	84	2EC8	
LHF9	F24D	84	2ED1	
BADTAP	F260	84	2EDA	
MEOF	F28D	84	2EE3	
MTAPER	F29F	84	2EEC	
BT1	F26B	84	2EF5	
FS10	F273	84	2EFE	
MBADR	F27B	84	2F07	
MCRLFS	F2B3	84	2F10	
NA1	F2E2	84	2F19	
NA3	F2F6	84	2F22	
OUTCHK	F2F9	84	2F2B	
OC10	F30C	84	2F34	
OC20	F320	84	2F3D	
OCL00P	F319	84	2F46	
PHF15	F339	84	2F4F	
PHF20	F339	84	2F58	
PUND10	F34B	84	2F61	
PUND20	F34D	84	2F6A	
PUNBYT	F392	84	2F73	
PREC10	F378	84	2F7C	
PX1STX	F3A6	84	2F85	
PX1	F3A6	84	2F8E	
PX2	F3B3	84	2F97	
RDF90	F3BB	84	2FA0	
RON90	F3C5	84	2FA9	
SETOUT	F3E2	84	2FB2	
SETPUL	F3E9	84	2FB8	
SETM1	F400	84	2FC4	
SETMEM	F3EA	84	2FCD	
PBADR	F3FA	84	2FD6	
SM5	F404	84	2FDF	
SM10	F40A	84	2FE8	
SM30	F41B	84	2FF1	
W10	F41E	84	2FFA	
W20	F427	84	3003	
W30	F433	84	300C	
MOVER	0001	84	3015	
DELAY	000A	84	301E	
PIA	5000	84	3027	
V50	0001	84	3030	
PROM	0004	84	3039	
RAM	FC00	84	3042	
P8BIN	F435	84	304B	

SYMBOL	VALUE	ATTR	LOCN	LINK
PSP	F448	84	3054	
AS	F43C	84	305D	
RASV	F448	84	3066	
RA1	F465	84	306F	
RA3	F476	84	3078	
RA4	F491	84	3081	
VERR	F494	84	308A	
AT	F480	84	3093	
ADDRS	F487	84	309C	
VFY1	F505	84	30A5	
AV	F4D6	84	30AE	
AN	F4DC	84	30B7	
JVER	F516	84	30C0	
INCD	F4EF	84	30C9	
AR	F4E3	84	30D2	
INK	F4F6	84	30DB	
EXIT	F502	84	30E4	
AX	F515	84	30ED	
BURN	F519	84	30F6	
OPMODE	D800	84	30FF	
MCINIT	F550	84	3108	
JBAD	F595	84	3111	
AM	F5AA	84	311A	
NITEM5	0018	84	3123	
LOCVV	F5C8	84	312C	
UC	0002	84	3135	
UB	0003	84	313E	
UA	0004	84	3147	
UXH	0005	84	3150	
UXL	0006	84	3159	
URH	0007	84	3162	
URL	0008	84	316B	
SRH	0000	84	3174	
SRL	0001	84	317D	
PUSHAL	F5F1	84	3186	
AS	F5F9	84	318F	
AC	F604	84	3198	
POPALL	F60D	84	31A1	
PC	F610	84	31AA	
PS	F61A	84	31B3	
TXAB	F628	84	31BC	
STAB	F62D	84	31C5	
TABX	F632	84	31CE	
XABX	F63C	84	31D7	
PUSX	F647	84	31E0	
SA	F64C	84	31E9	
PULX	F65E	84	31F2	
PA	F669	84	31FB	
ADDXAB	F674	84	3204	
ADDABX	F67B	84	320D	
ADDAB	F680	84	3216	
STAXH	F686	84	321F	
TEST2	F68B	84	3228	
TA	F68F	84	3231	

SYMBOL	VALUE	ATTR	LOCN	LINK
ADDAX	F692	84	323A	
ADDZ	F695	84	3243	
ADDBX	F699	84	324C	
SUBXAB	F69E	84	3255	
SUBABX	F6A5	84	325E	
SUBAX	F6B2	84	3267	
SSUB	F6B5	84	3270	
SUBBX	F6C0	84	3279	
INDEX	F6C5	84	3282	
MPY8	F6D8	84	328B	
MUL8	F6D0	84	3294	
JMPTZ	F6D6	84	329D	
MUL16	0800	84	32A6	
MSHIFT	0810	84	32AF	
MLOOP	080A	84	32B8	
BSHIFT	F6E4	84	32C1	
BL00P	F6E0	84	32CA	
SVECTO	F6EC	84	32D3	
P4HEX	F71C	84	32DC	
PHEX	F727	84	32E5	
P2HEX	F721	84	32EE	
ASCIIR	F754	84	32F7	
ASCIIL	F750	84	3300	
PUTAX	F740	84	3309	
PUTA	F745	84	3312	
PINCK	F735	84	331B	
PRTS	F73C	84	3324	
PRINTA	F73D	84	332D	
PRDY	F746	84	3336	
ARTS	F75E	84	333F	
PMESS	F75F	84	3348	
ETX	0004	84	3351	
TRTS	F76E	84	335A	
VALAN	F76F	84	3363	
ALPNUM	F779	84	336C	
SCARRY	F774	84	3375	
SETUS	F775	84	337E	
ANUM	F78D	84	3387	
ANOTOK	F795	84	3390	
NRTS	F797	84	3399	
AOK	F789	84	33A2	
JPINCK	F798	84	33AB	
INPUTA	F79A	84	33B4	
IWAIT	F79F	84	33BD	
CONAB	F7AC	84	33C6	
CLOOP1	F7B3	84	33CF	
CFOUND	F7C3	84	33D8	
ENDCNT	F7C1	84	33E1	
NOGOOD	F7E2	84	33EA	
CLOOP	F7CE	04	33F3	

END ASM V1.1

SEQ	LOC	OBJ	SOURCE	FTBL
0001			NAM	FTBL
0002			*	
0003			*GENERATED BY FTGN9*	
0004			* * * * *	
0005			*FREQUENCY TABLES	*
0006			* * * * *	
0007		E800	ORG	\$E800
0008			*	
0009	E800	E8 80	INDEX	FR06
0010	E802	E8 D1	FDB	FR07
0011	E804	E9 1A	FDB	FR08
0012	E806	E9 5B	FDB	FR09
0013	E808	E9 98	FDB	FR010
0014	E80A	E9 CD	FDB	FR011
0015	E80C	E9 FE	FDB	FR012
0016	E80E	EA 2F	FDB	FR013
0017	E810	EA 5C	FDB	FR014
0018	E812	EA 85	FDB	FR015
0019	E814	EA AE	FDB	FR016
0020	E816	EA D7	FDB	FR017
0021	E818	EA FC	FDB	FR018
0022	E81A	EB 21	FDB	FR019
0023	E81C	EB 42	FDB	FR020
0024	E81E	EB 63	FDB	FR021
0025	E820	EB 84	FDB	FR022
0026	E822	EB A5	FDB	FR023
0027	E824	EB C6	FDB	FR024
0028	E826	EB E7	FDB	FR025
0029	E828	EC 04	FDB	FR026
0030	E82A	EC 21	FDB	FR027
0031	E82C	EC 3E	FDB	FR028
0032	E82E	EC 5B	FDB	FR029
0033	E830	EC 78	FDB	FR030
0034	E832	EC 91	FDB	FR031
0035	E834	EC AA	FDB	FR032
0036	E836	EC C3	FDB	FR033
0037	E838	EC DC	FDB	FR034
0038	E83A	EC F5	FDB	FR035
0039	E83C	ED 0E	FDB	FR036
0040	E83E	ED 27	FDB	FR037
0041	E840	ED 40	FDB	FR038
0042	E842	ED 59	FDB	FR039
0043	E844	ED 72	FDB	FR040
0044	E846	ED 8B	FDB	FR041
0045	E848	ED A4	FDB	FR042
0046	E84A	ED B9	FDB	FR043
0047	E84C	ED CE	FDB	FR044
0048	E84E	ED E3	FDB	FR045
0049	E850	ED F8	FDB	FR046
0050	E852	EE 0D	FDB	FR047
0051	E854	EE 22	FDB	FR048
0052	E856	EE 37	FDB	FR049
0053	E858	EE 4C	FDB	FR050
0054	E85A	EE 61	FDB	FR051
0055	E85C	EE 76	FDB	FR052

SEQ LOC OBJ SOURCE

0056	E85E	EE	8B	FDB	FR053
0057	E860	EE	A0	FDB	FR054
0058	E862	EE	B5	FDB	FR055
0059	E864	EE	CA	FDB	FR056
0060	E866	EE	DB	FDB	FR057
0061	E868	EE	EC	FDB	FR058
0062	E86A	EE	FD	FDB	FR059
0063	E86C	EF	0E	FDB	FR060
0064			E880	ORG	\$E880

*FREQUENCY TABLE FOR 6 HZ.

*

0065				FR06	FDB	35	NO. -PULSE PERIODS IN 60 DEG SEG
0066							LENGTH OF PULSE PERIOD (USEC)
0067	E880	23					LENGTH OF 60 DEGREE SEG (USEC)
0068	E881	03	1A		FDB	794	SFMAX
0069	E883	6C	8E		FDB	27790	SFMIN
0070	E885	07	FF		FDB	2047	
0071	E887	00	80		FDB	128	

*PULSE WIDTHS

0072							
0073	E889	00	31		FDB	49	DOM
0074	E88B	00	30		FDB	48	
0075	E88D	00	32		FDB	50	DOM
0076	E88F	00	2F		FDB	47	
0077	E891	00	32		FDB	50	DOM
0078	E893	00	2E		FDB	46	
0079	E895	00	33		FDB	51	DOM
0080	E897	00	2D		FDB	45	
0081	E899	00	34		FDB	52	DOM
0082	E89B	00	2C		FDB	44	
0083	E89D	00	34		FDB	52	DOM
0084	E89F	00	2B		FDB	43	
0085	E8A1	00	35		FDB	53	DOM
0086	E8A3	00	2A		FDB	42	
0087	E8A5	00	36		FDB	54	DOM
0088	E8A7	00	29		FDB	41	
0089	E8A9	00	36		FDB	54	DOM
0090	E8AB	00	28		FDB	40	
0091	E8AD	00	36		FDB	54	DOM
0092	E8AF	00	27		FDB	39	
0093	E8B1	00	37		FDB	55	DOM
0094	E8B3	00	25		FDB	37	
0095	E8B5	00	37		FDB	55	DOM
0096	E8B7	00	24		FDB	36	
0097	E8B9	00	37		FDB	55	DOM
0098	E8BB	00	23		FDB	35	
0099	E8BD	00	38		FDB	56	DOM
0100	E8BF	00	21		FDB	33	
0101	E8C1	00	38		FDB	56	DOM
0102	E8C3	00	20		FDB	32	
0103	E8C5	00	38		FDB	56	DOM
0104	E8C7	00	1F		FDB	31	
0105	E8C9	00	38		FDB	56	DOM
0106	E8CB	00	1D		FDB	29	
0107	E8CD	00	38		FDB	56	DOM
0108	E8CF	00	1C		FDB	28	

*FREQUENCY TABLE FOR 7 HZ.

*

0109

0110

SEQ	LOC	OBJ	SOURCE			
0111	E801	1F	FR07	FCB	31	; NO. -PULSE PERIODS IN 60 DEG SEG
0112	E802	03 00		FDB	768	; LENGTH OF PULSE PERIOD (USEC)
0113	E804	50 00		FDB	23808	; LENGTH OF 60 DEGREE SEG (USEC)
0114	E806	07 FF		FDB	2047	; SFMAX
0115	E808	00 80		FDB	128	; SFMIN
0116			*PULSE WIDTHS			
0117	E80A	00 37		FDB	55	; DOM
0118	E80C	00 36		FDB	54	
0119	E80E	00 38		FDB	56	; DOM
0120	E8E0	00 35		FDB	53	
0121	E8E2	00 39		FDB	57	; DOM
0122	E8E4	00 34		FDB	52	
0123	E8E6	00 3A		FDB	58	; DOM
0124	E8E8	00 33		FDB	51	
0125	E8EA	00 3B		FDB	59	; DOM
0126	E8EC	00 31		FDB	49	
0127	E8EE	00 3C		FDB	60	; DOM
0128	E8F0	00 30		FDB	48	
0129	E8F2	00 3C		FDB	60	; DOM
0130	E8F4	00 2F		FDB	47	
0131	E8F6	00 3D		FDB	61	; DOM
0132	E8F8	00 2D		FDB	45	
0133	E8FA	00 3E		FDB	62	; DOM
0134	E8FC	00 2C		FDB	44	
0135	E8FE	00 3E		FDB	62	; DOM
0136	E900	00 2A		FDB	42	
0137	E902	00 3E		FDB	62	; DOM
0138	E904	00 28		FDB	40	
0139	E906	00 3F		FDB	63	; DOM
0140	E908	00 27		FDB	39	
0141	E90A	00 3F		FDB	63	; DOM
0142	E90C	00 25		FDB	37	
0143	E90E	00 3F		FDB	63	; DOM
0144	E910	00 23		FDB	35	
0145	E912	00 3F		FDB	63	; DOM
0146	E914	00 21		FDB	33	
0147	E916	00 3F		FDB	63	; DOM
0148	E918	00 1F		FDB	31	
0149			*FREQUENCY TABLE FOR 8 HZ.			
0150			*			
0151	E91A	1B	FR08	FCB	27	; NO. -PULSE PERIODS IN 60 DEG SEG
0152	E91B	03 04		FDB	772	; LENGTH OF PULSE PERIOD (USEC)
0153	E91D	51 6C		FDB	20844	; LENGTH OF 60 DEGREE SEG (USEC)
0154	E91F	07 FF		FDB	2047	; SFMAX
0155	E921	00 80		FDB	128	; SFMIN
0156			*PULSE WIDTHS			
0157	E923	00 40		FDB	64	; DOM
0158	E925	00 3E		FDB	62	
0159	E927	00 41		FDB	65	; DOM
0160	E929	00 3D		FDB	61	
0161	E92B	00 42		FDB	66	; DOM
0162	E92D	00 3B		FDB	59	
0163	E92F	00 43		FDB	67	; DOM
0164	E931	00 39		FDB	57	
0165	E933	00 44		FDB	68	; DOM

SEQ LOC OBJ SOURCE

0166	E935	00 38	FDB	56	
0167	E937	00 45	FDB	69	; DOM
0168	E939	00 36	FDB	54	
0169	E93B	00 46	FDB	70	; DOM
0170	E93D	00 34	FDB	52	
0171	E93F	00 47	FDB	71	; DOM
0172	E941	00 32	FDB	50	
0173	E943	00 47	FDB	71	; DOM
0174	E945	00 30	FDB	48	
0175	E947	00 48	FDB	72	; DOM
0176	E949	00 2E	FDB	46	
0177	E94B	00 48	FDB	72	; DOM
0178	E94D	00 2B	FDB	43	
0179	E94F	00 49	FDB	73	; DOM
0180	E951	00 29	FDB	41	
0181	E953	00 49	FDB	73	; DOM
0182	E955	00 27	FDB	39	
0183	E957	00 49	FDB	73	; DOM
0184	E959	00 24	FDB	36	

*FREQUENCY TABLE FOR 9 HZ.

*

0187	E95B	19	FRQ9	FCB	25	; NO. -PULSE PERIODS IN 60 DEG SEG
0188	E95C	02 E5		FDB	741	; LENGTH OF PULSE PERIOD (USEC)
0189	E95E	48 5D		FDB	18525	; LENGTH OF 60 DEGREE SEG (USEC)
0190	E960	07 FF		FDB	2047	; SFMAX
0191	E962	00 80		FDB	128	; SFMIN

*PULSE WIDTHS

0192						
0193	E964	00 45	FDB	69	; DOM	
0194	E966	00 43	FDB	67		
0195	E968	00 46	FDB	70	; DOM	
0196	E96A	00 41	FDB	65		
0197	E96C	00 48	FDB	72	; DOM	
0198	E96E	00 40	FDB	64		
0199	E970	00 49	FDB	73	; DOM	
0200	E972	00 3E	FDB	62		
0201	E974	00 4A	FDB	74	; DOM	
0202	E976	00 3B	FDB	59		
0203	E978	00 4B	FDB	75	; DOM	
0204	E97A	00 39	FDB	57		
0205	E97C	00 4C	FDB	76	; DOM	
0206	E97E	00 37	FDB	55		
0207	E980	00 4D	FDB	77	; DOM	
0208	E982	00 35	FDB	53		
0209	E984	00 4E	FDB	78	; DOM	
0210	E986	00 32	FDB	50		
0211	E988	00 4E	FDB	78	; DOM	
0212	E98A	00 2F	FDB	47		
0213	E98C	00 4E	FDB	78	; DOM	
0214	E98E	00 2D	FDB	45		
0215	E990	00 4F	FDB	79	; DOM	
0216	E992	00 2A	FDB	42		
0217	E994	00 4F	FDB	79	; DOM	
0218	E996	00 27	FDB	39		

*FREQUENCY TABLE FOR 10 HZ.

*

0219
0220

SEQ	LOC	OBJ	SOURCE			
0221	E998	15	FRQ10	FCB	21	; NO. -PULSE PERIODS IN 60 DEG SEG
0222	E999	03 1A		FDB	794	; LENGTH OF PULSE PERIOD (USEC)
0223	E998	41 22		FDB	16674	; LENGTH OF 60 DEGREE SEG (USEC)
0224	E99D	07 FF		FDB	2047	; SFMAX
0225	E99F	00 80		FDB	128	; SFMIN
0226			*PULSE WIDTHS			
0227	E9A1	00 52		FDB	82	; DOM
0228	E9A3	00 50		FDB	80	
0229	E9A5	00 54		FDB	84	; DOM
0230	E9A7	00 40		FDB	77	
0231	E9A9	00 56		FDB	86	; DOM
0232	E9AB	00 48		FDB	75	
0233	E9AD	00 58		FDB	88	; DOM
0234	E9AF	00 48		FDB	72	
0235	E9B1	00 5A		FDB	90	; DOM
0236	E9B3	00 45		FDB	69	
0237	E9B5	00 58		FDB	91	; DOM
0238	E9B7	00 41		FDB	65	
0239	E9B9	00 5C		FDB	92	; DOM
0240	E9BB	00 3E		FDB	62	
0241	E9BD	00 5D		FDB	93	; DOM
0242	E9BF	00 3A		FDB	58	
0243	E9C1	00 5D		FDB	93	; DOM
0244	E9C3	00 37		FDB	55	
0245	E9C5	00 5E		FDB	94	; DOM
0246	E9C7	00 33		FDB	51	
0247	E9C9	00 5E		FDB	94	; DOM
0248	E9CB	00 2F		FDB	47	
0249			*FREQUENCY TABLE FOR 11 HZ.			
0250			*			
0251	E9CD	13	FRQ11	FCB	19	; NO. -PULSE PERIODS IN 60 DEG SEG
0252	E9CE	03 1D		FDB	797	; LENGTH OF PULSE PERIOD (USEC)
0253	E9D0	3B 27		FDB	15143	; LENGTH OF 60 DEGREE SEG (USEC)
0254	E9D2	07 6A		FDB	1898	; SFMAX
0255	E9D4	00 80		FDB	128	; SFMIN
0256			*PULSE WIDTHS			
0257	E9D6	00 5B		FDB	91	; DOM
0258	E9D8	00 58		FDB	88	
0259	E9DA	00 5E		FDB	94	; DOM
0260	E9DC	00 55		FDB	85	
0261	E9DE	00 60		FDB	96	; DOM
0262	E9E0	00 52		FDB	82	
0263	E9E2	00 62		FDB	98	; DOM
0264	E9E4	00 4E		FDB	73	
0265	E9E6	00 64		FDB	100	; DOM
0266	E9E8	00 4A		FDB	74	
0267	E9EA	00 65		FDB	101	; DOM
0268	E9EC	00 46		FDB	70	
0269	E9EE	00 66		FDB	102	; DOM
0270	E9F0	00 42		FDB	66	
0271	E9F2	00 67		FDB	103	; DOM
0272	E9F4	00 3D		FDB	61	
0273	E9F6	00 68		FDB	104	; DOM
0274	E9F8	00 39		FDB	57	
0275	E9FA	00 68		FDB	104	; DOM

SEQ	LOC	OBJ	SOURCE
0276	ESFC	00 34	FDB 52
0277			*FREQUENCY TABLE FOR 12 HZ.
0278			*
0279	E9FE	13	FR012 FDB 19 ; NO. -PULSE PERIODS IN 60 DEG SEG
0280	E9FF	02 08	FDB 731 ; LENGTH OF PULSE PERIOD (USEC)
0281	EA01	36 41	FDB 13889 ; LENGTH OF 60 DEGREE SEG (USEC)
0282	EA03	06 07	FDB 1735 ; SFMAX
0283	EA05	00 80	FDB 128 ; SFMIN
0284			*PULSE WIDTHS
0285	EA07	00 58	FDB 91 ; DOM
0286	EA09	00 58	FDB 88 ; DOM
0287	EA08	00 5E	FDB 94 ; DOM
0288	EA00	00 55	FDB 85 ; DOM
0289	EA0F	00 60	FDB 96 ; DOM
0290	EA11	00 52	FDB 82 ; DOM
0291	EA13	00 62	FDB 98 ; DOM
0292	EA15	00 4E	FDB 78 ; DOM
0293	EA17	00 64	FDB 100 ; DOM
0294	EA19	00 4A	FDB 74 ; DOM
0295	EA1B	00 65	FDB 101 ; DOM
0296	EA1D	00 46	FDB 70 ; DOM
0297	EA1F	00 66	FDB 102 ; DOM
0298	EA21	00 42	FDB 66 ; DOM
0299	EA23	00 67	FDB 103 ; DOM
0300	EA25	00 3D	FDB 61 ; DOM
0301	EA27	00 68	FDB 104 ; DOM
0302	EA29	00 39	FDB 57 ; DOM
0303	EA2B	00 68	FDB 104 ; DOM
0304	EA2D	00 34	FDB 52 ; DOM
0305			*FREQUENCY TABLE FOR 13 HZ.
0306			*
0307	EA2F	11	FR013 FDB 17 ; NO. -PULSE PERIODS IN 60 DEG SEG
0308	EA30	02 F2	FDB 754 ; LENGTH OF PULSE PERIOD (USEC)
0309	EA32	32 12	FDB 12818 ; LENGTH OF 60 DEGREE SEG (USEC)
0310	EA34	06 47	FDB 1607 ; SFMAX
0311	EA36	00 80	FDB 128 ; SFMIN
0312			*PULSE WIDTHS
0313	EA38	00 66	FDB 102 ; DOM
0314	EA3A	00 63	FDB 99 ; DOM
0315	EA3C	00 69	FDB 105 ; DOM
0316	EA3E	00 5F	FDB 95 ; DOM
0317	EA40	00 6C	FDB 108 ; DOM
0318	EA42	00 5A	FDB 90 ; DOM
0319	EA44	00 6F	FDB 111 ; DOM
0320	EA46	00 56	FDB 86 ; DOM
0321	EA48	00 70	FDB 112 ; DOM
0322	EA4A	00 51	FDB 81 ; DOM
0323	EA4C	00 72	FDB 114 ; DOM
0324	EA4E	00 4B	FDB 75 ; DOM
0325	EA50	00 73	FDB 115 ; DOM
0326	EA52	00 46	FDB 70 ; DOM
0327	EA54	00 74	FDB 116 ; DOM
0328	EA56	00 40	FDB 64 ; DOM
0329	EA58	00 74	FDB 116 ; DOM
0330	EA5A	00 3A	FDB 58 ; DOM

SEQ	LOC	OBJ	SOURCE
0331			*FREQUENCY TABLE FOR 14 HZ.
0332			*
0333	EA50	0F	FRQ14 FCB 15 ; NO. -PULSE PERIODS IN 60 DEG SEG
0334	EA50	03 1A	FDB 794 ; LENGTH OF PULSE PERIOD (USEC)
0335	EA5F	2E 86	FDB 11910 ; LENGTH OF 60 DEGREE SEG (USEC)
0336	EA61	05 01	FDB 1489 ; SFMAX
0337	EA63	00 80	FDB 128 ; SFMIN
0338			*PULSE WIDTHS
0339	EA65	00 74	FDB 116 ; DOM
0340	EA67	00 6F	FDB 111 ; DOM
0341	EA69	00 78	FDB 120 ; DOM
0342	EA6B	00 6A	FDB 106 ; DOM
0343	EA6D	00 7C	FDB 124 ; DOM
0344	EA6F	00 65	FDB 101 ; DOM
0345	EA71	00 7E	FDB 126 ; DOM
0346	EA73	00 5E	FDB 94 ; DOM
0347	EA75	00 81	FDB 129 ; DOM
0348	EA77	00 58	FDB 88 ; DOM
0349	EA79	00 82	FDB 130 ; DOM
0350	EA7B	00 51	FDB 81 ; DOM
0351	EA7D	00 83	FDB 131 ; DOM
0352	EA7F	00 49	FDB 73 ; DOM
0353	EA81	00 84	FDB 132 ; DOM
0354	EA83	00 42	FDB 66 ; DOM
0355			*FREQUENCY TABLE FOR 15 HZ.
0356			*
0357	EA85	0F	FRQ15 FCB 15 ; NO. -PULSE PERIODS IN 60 DEG SEG
0358	EA86	02 E5	FDB 741 ; LENGTH OF PULSE PERIOD (USEC)
0359	EA88	2B 6B	FDB 11115 ; LENGTH OF 60 DEGREE SEG (USEC)
0360	EA8A	05 6B	FDB 1387 ; SFMAX
0361	EA8C	00 80	FDB 128 ; SFMIN
0362			*PULSE WIDTHS
0363	EA8E	00 74	FDB 116 ; DOM
0364	EA90	00 6F	FDB 111 ; DOM
0365	EA92	00 78	FDB 120 ; DOM
0366	EA94	00 6A	FDB 106 ; DOM
0367	EA96	00 7C	FDB 124 ; DOM
0368	EA98	00 65	FDB 101 ; DOM
0369	EA9A	00 7E	FDB 126 ; DOM
0370	EA9C	00 5E	FDB 94 ; DOM
0371	EA9E	00 81	FDB 129 ; DOM
0372	EAA0	00 58	FDB 88 ; DOM
0373	EAA2	00 82	FDB 130 ; DOM
0374	EAA4	00 51	FDB 81 ; DOM
0375	EAA6	00 83	FDB 131 ; DOM
0376	EAA8	00 49	FDB 73 ; DOM
0377	EAAA	00 84	FDB 132 ; DOM
0378	EAAC	00 42	FDB 66 ; DOM
0379			*FREQUENCY TABLE FOR 16 HZ.
0380			*
0381	EAAE	0F	FRQ16 FCB 15 ; NO. -PULSE PERIODS IN 60 DEG SEG
0382	EAAF	02 B6	FDB 694 ; LENGTH OF PULSE PERIOD (USEC)
0383	EAB1	2B HA	FDB 10410 ; LENGTH OF 60 DEGREE SEG (USEC)
0384	EAB3	05 10	FDB 1296 ; SFMAX
0385	EAB5	00 80	FDB 128 ; SFMIN

SEQ LOC OBJ SOURCE

0386 *PULSE WIDTHS

0387	EA87	00	74	FDB	116	; DOM
0388	EA89	00	6F	FDB	111	
0389	EA8B	00	78	FDB	120	; DOM
0390	EA8D	00	6A	FDB	106	
0391	EA8F	00	7C	FDB	124	; DOM
0392	EA91	00	6D	FDB	101	
0393	EA93	00	7E	FDB	126	; DOM
0394	EA95	00	5E	FDB	94	
0395	EA97	00	81	FDB	129	; DOM
0396	EA99	00	58	FDB	88	
0397	EA9B	00	82	FDB	130	; DOM
0398	EA9D	00	51	FDB	81	
0399	EA9F	00	83	FDB	131	; DOM
0400	EAD1	00	49	FDB	73	
0401	EAD3	00	84	FDB	132	; DOM
0402	EAD5	00	42	FDB	66	

*FREQUENCY TABLE FOR 17 HZ.

*

0404				FRQ17	FDB	13	; NO. -PULSE PERIODS IN 60 DEG SEG
0405	EAD7	00			FDB	754	; LENGTH OF PULSE PERIOD (USEC)
0406	EAD8	02	F2		FDB	9802	; LENGTH OF 60 DEGREE SEG (USEC)
0407	EADA	26	4A		FDB	1226	; SFMAX
0408	EADC	04	0A		FDB	128	; SFMIN
0409	EADE	00	80		FDB		

*PULSE WIDTHS

0410							
0411	EAE0	00	86	FDB	134	; DOM	
0412	EAE2	00	80	FDB	128		
0413	EAE4	00	8C	FDB	140	; DOM	
0414	EAE6	00	79	FDB	121		
0415	EAE8	00	90	FDB	144	; DOM	
0416	EAEA	00	72	FDB	114		
0417	EAE0	00	93	FDB	147	; DOM	
0418	EAE6	00	69	FDB	105		
0419	EAF0	00	96	FDB	150	; DOM	
0420	EAF2	00	60	FDB	96		
0421	EAF4	00	97	FDB	151	; DOM	
0422	EAF6	00	56	FDB	86		
0423	EAF8	00	98	FDB	152	; DOM	
0424	EAF0	00	4C	FDB	76		

*FREQUENCY TABLE FOR 18 HZ.

*

0426				FRQ18	FDB	13	; NO. -PULSE PERIODS IN 60 DEG SEG
0427	EAF0	00			FDB	712	; LENGTH OF PULSE PERIOD (USEC)
0428	EAFD	02	08		FDB	9256	; LENGTH OF 60 DEGREE SEG (USEC)
0429	EAFD	24	28		FDB	1155	; SFMAX
0430	EB01	04	83		FDB	128	; SFMIN
0431	EB03	00	80		FDB		

*PULSE WIDTHS

0432							
0433	EB05	00	86	FDB	134	; DOM	
0434	EB07	00	80	FDB	128		
0435	EB09	00	8C	FDB	140	; DOM	
0436	EB0B	00	79	FDB	121		
0437	EB0D	00	90	FDB	144	; DOM	
0438	EB0F	00	72	FDB	114		
0439	EB11	00	93	FDB	147	; DOM	
0440	EB13	00	69	FDB	105		

SEQ LOC OBJ SOURCE

0441	EB15	00	96	FDB	150	/DOM
0442	EB17	00	60	FDB	96	
0443	EB19	00	97	FDB	151	/DOM
0444	EB1B	00	56	FDB	86	
0445	EB1D	00	98	FDB	152	/DOM
0446	EB1F	00	40	FDB	76	
0447				*FREQUENCY TABLE FOR 19 HZ.		
0448				*		
0449	EB21	06		FRQ19	FDB	11 /NO. -PULSE PERIODS IN 60 DEG SEG
0450	EB22	01	10	FDB	797	/LENGTH OF PULSE PERIOD (USEC)
0451	EB24	22	3F	FDB	8767	/LENGTH OF 60 DEGREE SEG (USEC)
0452	EB26	04	4F	FDB	1103	/SFMAX
0453	EB28	00	80	FDB	128	/SFMIN
0454				*PULSE WIDTHS		
0455	EB2A	00	9F	FDB	159	/DOM
0456	EB2C	00	97	FDB	151	
0457	EB2E	00	H7	FDB	167	/DOM
0458	EB30	00	80	FDB	141	
0459	EB32	00	AC	FDB	172	/DOM
0460	EB34	00	82	FDB	130	
0461	EB36	00	80	FDB	176	/DOM
0462	EB38	00	75	FDB	117	
0463	EB3A	00	B3	FDB	179	/DOM
0464	EB3C	00	68	FDB	104	
0465	EB3E	00	B3	FDB	179	/DOM
0466	EB40	00	5H	FDB	90	
0467				*FREQUENCY TABLE FOR 20 HZ.		
0468				*		
0469	EB42	06		FRQ20	FDB	11 /NO. -PULSE PERIODS IN 60 DEG SEG
0470	EB43	02	F6	FDB	798	/LENGTH OF PULSE PERIOD (USEC)
0471	EB45	20	32	FDB	8336	/LENGTH OF 60 DEGREE SEG (USEC)
0472	EB47	04	17	FDB	1047	/SFMAX
0473	EB49	00	80	FDB	128	/SFMIN
0474				*PULSE WIDTHS		
0475	EB4B	00	9F	FDB	159	/DOM
0476	EB4D	00	97	FDB	151	
0477	EB4F	00	H7	FDB	167	/DOM
0478	EB51	00	80	FDB	141	
0479	EB53	00	AC	FDB	172	/DOM
0480	EB55	00	82	FDB	130	
0481	EB57	00	80	FDB	176	/DOM
0482	EB59	00	75	FDB	117	
0483	EB5B	00	B3	FDB	179	/DOM
0484	EB5D	00	68	FDB	104	
0485	EB5F	00	B3	FDB	179	/DOM
0486	EB61	00	5H	FDB	90	
0487				*FREQUENCY TABLE FOR 21 HZ.		
0488				*		
0489	EB63	06		FRQ21	FDB	11 /NO. -PULSE PERIODS IN 60 DEG SEG
0490	EB64	02	D2	FDB	722	/LENGTH OF PULSE PERIOD (USEC)
0491	EB66	1F	06	FDB	7942	/LENGTH OF 60 DEGREE SEG (USEC)
0492	EB68	02	E3	FDB	995	/SFMAX
0493	EB6A	00	80	FDB	128	/SFMIN
0494				*PULSE WIDTHS		
0495	EB6C	00	9F	FDB	159	/DOM

AD-A140 299

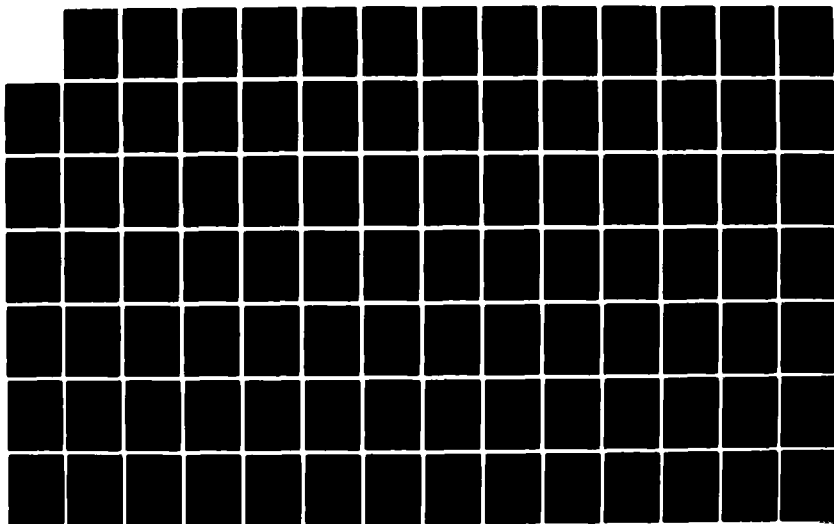
ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE
OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83
DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

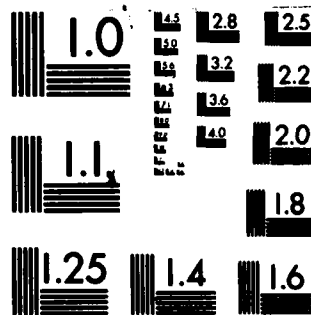
3/4

UNCLASSIFIED

F/G 9/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SEQ LOC OBJ SOURCE

0496	EB6E	00	97	FDB	151	
0497	EB70	00	A7	FDB	167	; DOM
0498	EB72	00	8D	FDB	141	
0499	EB74	00	AC	FDB	172	; DOM
0500	EB76	00	82	FDB	130	
0501	EB78	00	60	FDB	176	; DOM
0502	EB7A	00	75	FDB	117	
0503	EB7C	00	83	FDB	179	; DOM
0504	EB7E	00	68	FDB	104	
0505	EB80	00	83	FDB	179	; DOM
0506	EB82	00	5A	FDB	90	

*FREQUENCY TABLE FOR 22 HZ.

*

0509	EB84	0B		FRQ22	FCB	11	; NO. -PULSE PERIODS IN 60 DEG SEG
0510	EB85	02	61		FDB	689	; LENGTH OF PULSE PERIOD (USEC)
0511	EB87	10	98		FDB	7579	; LENGTH OF 60 DEGREE SEG (USEC)
0512	EB89	03	84		FDB	948	; SFMAX
0513	EB8B	00	80		FDB	128	; SFMIN

*PULSE WIDTHS

0514							
0515	EB8D	00	9F	FDB	159	; DOM	
0516	EB8F	00	97	FDB	151		
0517	EB91	00	A7	FDB	167	; DOM	
0518	EB93	00	8D	FDB	141		
0519	EB95	00	AC	FDB	172	; DOM	
0520	EB97	00	82	FDB	130		
0521	EB99	00	80	FDB	176	; DOM	
0522	EB9B	00	75	FDB	117		
0523	EB9D	00	83	FDB	179	; DOM	
0524	EB9F	00	68	FDB	104		
0525	EBA1	00	83	FDB	179	; DOM	
0526	EBA3	00	5A	FDB	90		

*FREQUENCY TABLE FOR 23 HZ.

*

0529	EBA5	0B		FRQ23	FCB	11	; NO. -PULSE PERIODS IN 60 DEG SEG
0530	EBA6	02	93		FDB	659	; LENGTH OF PULSE PERIOD (USEC)
0531	EBA8	10	51		FDB	7249	; LENGTH OF 60 DEGREE SEG (USEC)
0532	EBAH	03	89		FDB	905	; SFMAX
0533	EBAH	00	80		FDB	128	; SFMIN

*PULSE WIDTHS

0534							
0535	EBAE	00	9F	FDB	159	; DOM	
0536	EBB0	00	97	FDB	151		
0537	EBB2	00	A7	FDB	167	; DOM	
0538	EBB4	00	8D	FDB	141		
0539	EBB6	00	AC	FDB	172	; DOM	
0540	EBB8	00	82	FDB	130		
0541	EBBH	00	80	FDB	176	; DOM	
0542	EBBC	00	75	FDB	117		
0543	EBBE	00	83	FDB	179	; DOM	
0544	EBD0	00	68	FDB	104		
0545	EBD2	00	83	FDB	179	; DOM	
0546	EBD4	00	5A	FDB	90		

*FREQUENCY TABLE FOR 24 HZ.

*

0549	EBD6	0B		FRQ24	FCB	11	; NO. -PULSE PERIODS IN 60 DEG SEG
0550	EBD7	02	77		FDB	631	; LENGTH OF PULSE PERIOD (USEC)

SEQ	LOC	OBJ	SOURCE
0551	EB05	18 10	FDB 6941 ; LENGTH OF 60 DEGREE SEG (USEC)
0552	EB06	03 61	FDB 865 ; SFMAX
0553	EB00	00 80	FDB 128 ; SFMIN
0554			*PULSE WIDTHS
0555	EB0F	00 H0	FDB 160 ; DUM
0556	EB01	00 97	FDB 151 ; DUM
0557	EB03	00 H7	FDB 167 ; DUM
0558	EB05	00 80	FDB 141 ; DUM
0559	EB07	00 H0	FDB 172 ; DUM
0560	EB09	00 82	FDB 130 ; DUM
0561	EB0B	00 80	FDB 176 ; DUM
0562	EB0D	00 75	FDB 117 ; DUM
0563	EB0F	00 83	FDB 179 ; DUM
0564	EBE1	00 68	FDB 104 ; DUM
0565	EBE3	00 83	FDB 179 ; DUM
0566	EBE5	00 5A	FDB 90 ; DUM
0567			*FREQUENCY TABLE FOR 25 HZ.
0568			*
0569	EBE7	09	FRQ25 FCB 9 ; NO. -PULSE PERIODS IN 60 DEG SEG
0570	EBE8	02 E5	FDB 741 ; LENGTH OF PULSE PERIOD (USEC)
0571	EBEH	1H 00	FDB 6669 ; LENGTH OF 60 DEGREE SEG (USEC)
0572	EBEC	03 44	FDB 836 ; SFMAX
0573	EBEE	00 80	FDB 128 ; SFMIN
0574			*PULSE WIDTHS
0575	EBF0	00 C4	FDB 196 ; DUM
0576	EBF2	00 B7	FDB 183 ; DUM
0577	EBF4	00 CE	FDB 206 ; DUM
0578	EBF6	00 H8	FDB 168 ; DUM
0579	EBF8	00 D6	FDB 214 ; DUM
0580	EBFA	00 96	FDB 150 ; DUM
0581	EBFC	00 DA	FDB 218 ; DUM
0582	EBFE	00 83	FDB 131 ; DUM
0583	EC00	00 DB	FDB 219 ; DUM
0584	EC02	00 6E	FDB 110 ; DUM
0585			*FREQUENCY TABLE FOR 26 HZ.
0586			*
0587	EC04	09	FRQ26 FCB 9 ; NO. -PULSE PERIODS IN 60 DEG SEG
0588	EC05	02 C8	FDB 712 ; LENGTH OF PULSE PERIOD (USEC)
0589	EC07	19 08	FDB 6408 ; LENGTH OF 60 DEGREE SEG (USEC)
0590	EC09	03 22	FDB 802 ; SFMAX
0591	EC0B	00 80	FDB 128 ; SFMIN
0592			*PULSE WIDTHS
0593	EC0D	00 C4	FDB 196 ; DUM
0594	EC0F	00 B7	FDB 183 ; DUM
0595	EC11	00 CE	FDB 206 ; DUM
0596	EC13	00 H8	FDB 168 ; DUM
0597	EC15	00 D6	FDB 214 ; DUM
0598	EC17	00 96	FDB 150 ; DUM
0599	EC19	00 DA	FDB 218 ; DUM
0600	EC1B	00 83	FDB 131 ; DUM
0601	EC1D	00 DB	FDB 219 ; DUM
0602	EC1F	00 6E	FDB 110 ; DUM
0603			*FREQUENCY TABLE FOR 27 HZ.
0604			*
0605	EC21	09	FRQ27 FCB 9 ; NO. -PULSE PERIODS IN 60 DEG SEG

SEQ	LUC	OBJ	SOURCE
0606	EC22	02 AE	FDB 686
0607	EC24	18 1E	FDB 6174
0608	EC26	03 04	FDB 772
0609	EC28	06 80	FDB 128
0610			*PULSE WIDTHS
0611	EC2A	00 C4	FDB 196
0612	EC2C	00 B7	FDB 183
0613	EC2E	00 CE	FDB 206
0614	EC30	00 A8	FDB 168
0615	EC32	00 D6	FDB 214
0616	EC34	00 96	FDB 150
0617	EC36	00 DA	FDB 218
0618	EC38	00 83	FDB 131
0619	EC3A	00 DB	FDB 219
0620	EC3C	00 6E	FDB 110
0621			*FREQUENCY TABLE FOR 28 HZ.
0622			*
0623	EC3E	09	FRQ28 FCB 9
0624	EC3F	02 95	FDB 661
0625	EC41	17 3D	FDB 5949
0626	EC43	02 E6	FDB 742
0627	EC45	00 80	FDB 128
0628			*PULSE WIDTHS
0629	EC47	00 C4	FDB 196
0630	EC49	00 B7	FDB 183
0631	EC4B	00 CE	FDB 206
0632	EC4D	00 A8	FDB 168
0633	EC4F	00 D6	FDB 214
0634	EC51	00 96	FDB 150
0635	EC53	00 DA	FDB 218
0636	EC55	00 83	FDB 131
0637	EC57	00 DB	FDB 219
0638	EC59	00 6E	FDB 110
0639			*FREQUENCY TABLE FOR 29 HZ.
0640			*
0641	EC5B	09	FRQ29 FCB 9
0642	EC5C	02 7F	FDB 639
0643	EC5E	16 77	FDB 5751
0644	EC60	02 CD	FDB 717
0645	EC62	00 80	FDB 128
0646			*PULSE WIDTHS
0647	EC64	00 C4	FDB 196
0648	EC66	00 B7	FDB 183
0649	EC68	00 CE	FDB 206
0650	EC6A	00 A8	FDB 168
0651	EC6C	00 D6	FDB 214
0652	EC6E	00 96	FDB 150
0653	EC70	00 DA	FDB 218
0654	EC72	00 83	FDB 131
0655	EC74	00 DB	FDB 219
0656	EC76	00 6E	FDB 110
0657			*FREQUENCY TABLE FOR 30 HZ.
0658			*
0659	EC78	07	FRQ30 FCB 7
0660	EC79	03 1H	FDB 794

SEQ	LOC	OBJ	SOURCE
0661	EC7B	15 B6	FDB 5558 ; LENGTH OF 60 DEGREE SEG (USEC)
0662	EC7D	02 B9	FDB 697 ; SFMAX
0663	EC7F	00 80	FDB 128 ; SFMIN
0664			*PULSE WIDTHS
0665	EC81	00 FE	FDB 254 ; DOM
0666	EC83	00 E9	FDB 233 ; DOM
0667	EC85	01 0E	FDB 270 ; DOM
0668	EC87	00 CF	FDB 207 ; DOM
0669	EC89	01 17	FDB 279 ; DOM
0670	EC8B	00 B0	FDB 176 ; DOM
0671	EC8D	01 1H	FDB 282 ; DOM
0672	EC8F	00 8D	FDB 141 ; DOM
0673			*FREQUENCY TABLE FOR 31 HZ.
0674			*
0675	EC91	07	FRQ31 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0676	EC92	03 00	FDB 768 ; LENGTH OF PULSE PERIOD (USEC)
0677	EC94	15 00	FDB 5376 ; LENGTH OF 60 DEGREE SEG (USEC)
0678	EC96	02 A2	FDB 674 ; SFMAX
0679	EC98	00 80	FDB 128 ; SFMIN
0680			*PULSE WIDTHS
0681	EC9A	00 FE	FDB 254 ; DOM
0682	EC9C	00 E9	FDB 233 ; DOM
0683	EC9E	01 0E	FDB 270 ; DOM
0684	ECA0	00 CF	FDB 207 ; DOM
0685	ECA2	01 17	FDB 279 ; DOM
0686	ECA4	00 B0	FDB 176 ; DOM
0687	ECA6	01 1H	FDB 282 ; DOM
0688	ECA8	00 8D	FDB 141 ; DOM
0689			*FREQUENCY TABLE FOR 32 HZ.
0690			*
0691	ECAA	07	FRQ32 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0692	EACB	02 E8	FDB 744 ; LENGTH OF PULSE PERIOD (USEC)
0693	EACD	14 58	FDB 5208 ; LENGTH OF 60 DEGREE SEG (USEC)
0694	ECAF	02 8C	FDB 652 ; SFMAX
0695	ECB1	00 80	FDB 128 ; SFMIN
0696			*PULSE WIDTHS
0697	ECB3	00 FE	FDB 254 ; DOM
0698	ECB5	00 E9	FDB 233 ; DOM
0699	ECB7	01 0E	FDB 270 ; DOM
0700	ECB9	00 CF	FDB 207 ; DOM
0701	ECBB	01 17	FDB 279 ; DOM
0702	ECBD	00 B0	FDB 176 ; DOM
0703	ECBF	01 1H	FDB 282 ; DOM
0704	ECC1	00 8D	FDB 141 ; DOM
0705			*FREQUENCY TABLE FOR 33 HZ.
0706			*
0707	ECC3	07	FRQ33 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0708	ECC4	02 D2	FDB 722 ; LENGTH OF PULSE PERIOD (USEC)
0709	ECC6	13 BE	FDB 5054 ; LENGTH OF 60 DEGREE SEG (USEC)
0710	ECC8	02 78	FDB 632 ; SFMAX
0711	ECCA	00 80	FDB 128 ; SFMIN
0712			*PULSE WIDTHS
0713	ECCC	00 FE	FDB 254 ; DOM
0714	ECCE	00 E9	FDB 233 ; DOM
0715	ECD0	01 0E	FDB 270 ; DOM

SEQ	LOC	OBJ	SOURCE
0716	ECD2	00 CF	FDB 207
0717	ECD4	01 17	FDB 279 ; DOM
0718	ECD6	00 B0	FDB 176
0719	ECD8	01 1A	FDB 282 ; DOM
0720	ECD8	00 80	FDB 141
0721			*FREQUENCY TABLE FOR 34 HZ.
0722			*
0723	ECDC	07	FRQ34 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0724	ECDD	02 B0	FDB 700 ; LENGTH OF PULSE PERIOD (USEC)
0725	ECDF	13 24	FDB 4900 ; LENGTH OF 60 DEGREE SEG (USEC)
0726	ECE1	02 64	FDB 612 ; SFMAX
0727	ECE3	00 80	FDB 128 ; SFMIN
0728			*PULSE WIDTHS
0729	ECE5	00 FE	FDB 254 ; DOM
0730	ECE7	00 E9	FDB 233
0731	ECE9	01 0E	FDB 270 ; DOM
0732	ECEB	00 CF	FDB 207
0733	ECED	01 17	FDB 279 ; DOM
0734	ECEF	00 B0	FDB 176
0735	ECF1	01 1A	FDB 282 ; DOM
0736	ECF3	00 80	FDB 141
0737			*FREQUENCY TABLE FOR 35 HZ.
0738			*
0739	ECF5	07	FRQ35 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0740	ECF6	02 A8	FDB 680 ; LENGTH OF PULSE PERIOD (USEC)
0741	ECF8	12 98	FDB 4760 ; LENGTH OF 60 DEGREE SEG (USEC)
0742	ECFA	02 52	FDB 594 ; SFMAX
0743	ECFC	00 80	FDB 128 ; SFMIN
0744			*PULSE WIDTHS
0745	ECFE	00 FE	FDB 254 ; DOM
0746	ED00	00 E9	FDB 233
0747	ED02	01 0E	FDB 270 ; DOM
0748	ED04	00 CF	FDB 207
0749	ED06	01 17	FDB 279 ; DOM
0750	ED08	00 B0	FDB 176
0751	ED0A	01 1A	FDB 282 ; DOM
0752	ED0C	00 80	FDB 141
0753			*FREQUENCY TABLE FOR 36 HZ.
0754			*
0755	ED0E	07	FRQ36 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0756	ED0F	02 95	FDB 661 ; LENGTH OF PULSE PERIOD (USEC)
0757	ED11	12 13	FDB 4627 ; LENGTH OF 60 DEGREE SEG (USEC)
0758	ED13	02 40	FDB 576 ; SFMAX
0759	ED15	00 80	FDB 128 ; SFMIN
0760			*PULSE WIDTHS
0761	ED17	00 FE	FDB 254 ; DOM
0762	ED19	00 E9	FDB 233
0763	ED1B	01 0E	FDB 270 ; DOM
0764	ED1D	00 CF	FDB 207
0765	ED1F	01 17	FDB 279 ; DOM
0766	ED21	00 B0	FDB 176
0767	ED23	01 1A	FDB 282 ; DOM
0768	ED25	00 80	FDB 141
0769			*FREQUENCY TABLE FOR 37 HZ.
0770			*

SEQ	LOC	OBJ	SOURCE
0771	ED27	07	FRQ37 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0772	ED28	02 84	FDB 644 ; LENGTH OF PULSE PERIOD (USEC)
0773	ED2A	11 9C	FDB 4508 ; LENGTH OF 60 DEGREE SEG (USEC)
0774	ED2C	02 31	FDB 561 ; SFMAX
0775	ED2E	00 80	FDB 128 ; SFMIN
0776			*PULSE WIDTHS
0777	ED30	00 FE	FDB 254 ; DOM
0778	ED32	00 E9	FDB 233
0779	ED34	01 0E	FDB 270 ; DOM
0780	ED36	00 CF	FDB 207
0781	ED38	01 17	FDB 279 ; DOM
0782	ED3A	00 B0	FDB 176
0783	ED3C	01 1A	FDB 282 ; DOM
0784	ED3E	00 8D	FDB 141
0785			*FREQUENCY TABLE FOR 38 HZ.
0786			*
0787	ED40	07	FRQ38 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0788	ED41	02 73	FDB 627 ; LENGTH OF PULSE PERIOD (USEC)
0789	ED43	11 25	FDB 4389 ; LENGTH OF 60 DEGREE SEG (USEC)
0790	ED45	02 22	FDB 546 ; SFMAX
0791	ED47	00 80	FDB 128 ; SFMIN
0792			*PULSE WIDTHS
0793	ED49	00 FE	FDB 254 ; DOM
0794	ED4B	00 E9	FDB 233
0795	ED4D	01 0E	FDB 270 ; DOM
0796	ED4F	00 CF	FDB 207
0797	ED51	01 17	FDB 279 ; DOM
0798	ED53	00 B0	FDB 176
0799	ED55	01 1A	FDB 282 ; DOM
0800	ED57	00 8D	FDB 141
0801			*FREQUENCY TABLE FOR 39 HZ.
0802			*
0803	ED59	07	FRQ39 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0804	ED5A	02 63	FDB 611 ; LENGTH OF PULSE PERIOD (USEC)
0805	ED5C	10 B5	FDB 4277 ; LENGTH OF 60 DEGREE SEG (USEC)
0806	ED5E	02 13	FDB 531 ; SFMAX
0807	ED60	00 80	FDB 128 ; SFMIN
0808			*PULSE WIDTHS
0809	ED62	00 FE	FDB 254 ; DOM
0810	ED64	00 E9	FDB 233
0811	ED66	01 0E	FDB 270 ; DOM
0812	ED68	00 CF	FDB 207
0813	ED6A	01 17	FDB 279 ; DOM
0814	ED6C	00 B0	FDB 176
0815	ED6E	01 1A	FDB 282 ; DOM
0816	ED70	00 8D	FDB 141
0817			*FREQUENCY TABLE FOR 40 HZ.
0818			*
0819	ED72	07	FRQ40 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0820	ED73	02 53	FDB 595 ; LENGTH OF PULSE PERIOD (USEC)
0821	ED75	10 45	FDB 4165 ; LENGTH OF 60 DEGREE SEG (USEC)
0822	ED77	02 05	FDB 517 ; SFMAX
0823	ED79	00 80	FDB 128 ; SFMIN
0824			*PULSE WIDTHS
0825	ED7B	00 FE	FDB 254 ; DOM

SEQ	LOC	OBJ	SOURCE
0826	ED7D	00 E9	FDB 233
0827	ED7F	01 0E	FDB 270 ; DOM
0828	ED81	00 CF	FDB 207
0829	ED83	01 17	FDB 279 ; DOM
0830	ED85	00 B0	FDB 176
0831	ED87	01 1A	FDB 282 ; DOM
0832	ED89	00 8D	FDB 141
0833			*FREQUENCY TABLE FOR 41 HZ.
0834			*
0835	ED8B	07	FRQ41 FCB 7 ; NO. -PULSE PERIODS IN 60 DEG SEG
0836	ED8C	02 45	FDB 581 ; LENGTH OF PULSE PERIOD (USEC)
0837	ED8E	0F E3	FDB 4067 ; LENGTH OF 60 DEGREE SEG (USEC)
0838	ED90	01 F8	FDB 504 ; SFMAX
0839	ED92	00 80	FDB 128 ; SFMIN
0840			*PULSE WIDTHS
0841	ED94	00 FE	FDB 254 ; DOM
0842	ED96	00 E9	FDB 233
0843	ED98	01 0E	FDB 270 ; DOM
0844	ED9A	00 CF	FDB 207
0845	ED9C	01 17	FDB 279 ; DOM
0846	ED9E	00 B0	FDB 176
0847	EDA0	01 1A	FDB 282 ; DOM
0848	EDA2	00 8D	FDB 141
0849			*FREQUENCY TABLE FOR 42 HZ.
0850			*
0851	EDA4	05	FRQ42 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0852	EDA5	03 1A	FDB 794 ; LENGTH OF PULSE PERIOD (USEC)
0853	EDA7	0F 82	FDB 3970 ; LENGTH OF 60 DEGREE SEG (USEC)
0854	EDA9	01 F2	FDB 498 ; SFMAX
0855	EDAB	00 80	FDB 128 ; SFMIN
0856			*PULSE WIDTHS
0857	EDAD	01 69	FDB 361 ; DOM
0858	EDAF	01 3F	FDB 319
0859	EDB1	01 82	FDB 386 ; DOM
0860	EDB3	01 08	FDB 264
0861	EDB5	01 8B	FDB 395 ; DOM
0862	EDB7	00 C5	FDB 197
0863			*FREQUENCY TABLE FOR 43 HZ.
0864			*
0865	EDB9	05	FRQ43 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0866	EDBA	03 07	FDB 775 ; LENGTH OF PULSE PERIOD (USEC)
0867	EDBC	0F 23	FDB 3875 ; LENGTH OF 60 DEGREE SEG (USEC)
0868	EDBE	01 E5	FDB 485 ; SFMAX
0869	EDC0	00 80	FDB 128 ; SFMIN
0870			*PULSE WIDTHS
0871	EDC2	01 69	FDB 361 ; DOM
0872	EDC4	01 3F	FDB 319
0873	EDC6	01 82	FDB 386 ; DOM
0874	EDC8	01 08	FDB 264
0875	EDCA	01 8B	FDB 395 ; DOM
0876	EDCC	00 C5	FDB 197
0877			*FREQUENCY TABLE FOR 44 HZ.
0878			*
0879	EDCE	05	FRQ44 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0880	EDCF	02 F6	FDB 758 ; LENGTH OF PULSE PERIOD (USEC)

SEQ	LOC	OBJ	SOURCE
0881	EDD1	0E CE	FDB 3790 ; LENGTH OF 60 DEGREE SEG (USEC)
0882	EDD3	01 DA	FDB 474 ; SFMAX
0883	EDD5	00 80	FDB 128 ; SFMIN
0884			*PULSE WIDTHS
0885	EDD7	01 69	FDB 361 ; DOM
0886	EDD9	01 3F	FDB 319 ; DOM
0887	EDD8	01 82	FDB 386 ; DOM
0888	EDD0	01 08	FDB 264 ; DOM
0889	EDDF	01 88	FDB 395 ; DOM
0890	EDE1	00 C5	FDB 197
0891			*FREQUENCY TABLE FOR 45 HZ.
0892			*
0893	EDE3	05	FRQ45 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0894	EDE4	02 E5	FDB 741 ; LENGTH OF PULSE PERIOD (USEC)
0895	EDE6	0E 79	FDB 3705 ; LENGTH OF 60 DEGREE SEG (USEC)
0896	EDE8	01 CF	FDB 463 ; SFMAX
0897	EDEA	00 80	FDB 128 ; SFMIN
0898			*PULSE WIDTHS
0899	EDEC	01 69	FDB 361 ; DOM
0900	EDEE	01 3F	FDB 319 ; DOM
0901	EDF0	01 82	FDB 386 ; DOM
0902	EDF2	01 08	FDB 264 ; DOM
0903	EDF4	01 88	FDB 395 ; DOM
0904	EDF6	00 C5	FDB 197
0905			*FREQUENCY TABLE FOR 46 HZ.
0906			*
0907	EDF8	05	FRQ46 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0908	EDF9	02 D5	FDB 725 ; LENGTH OF PULSE PERIOD (USEC)
0909	EDFB	0E 29	FDB 3625 ; LENGTH OF 60 DEGREE SEG (USEC)
0910	EDFD	01 C5	FDB 453 ; SFMAX
0911	EDFF	00 80	FDB 128 ; SFMIN
0912			*PULSE WIDTHS
0913	EE01	01 69	FDB 361 ; DOM
0914	EE03	01 3F	FDB 319 ; DOM
0915	EE05	01 82	FDB 386 ; DOM
0916	EE07	01 08	FDB 264 ; DOM
0917	EE09	01 88	FDB 395 ; DOM
0918	EE0B	00 C5	FDB 197
0919			*FREQUENCY TABLE FOR 47 HZ.
0920			*
0921	EE0D	05	FRQ47 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0922	EE0E	02 C5	FDB 709 ; LENGTH OF PULSE PERIOD (USEC)
0923	EE10	0D D9	FDB 3545 ; LENGTH OF 60 DEGREE SEG (USEC)
0924	EE12	01 88	FDB 443 ; SFMAX
0925	EE14	00 80	FDB 128 ; SFMIN
0926			*PULSE WIDTHS
0927	EE16	01 69	FDB 361 ; DOM
0928	EE18	01 3F	FDB 319 ; DOM
0929	EE1A	01 82	FDB 386 ; DOM
0930	EE1C	01 08	FDB 264 ; DOM
0931	EE1E	01 88	FDB 395 ; DOM
0932	EE20	00 C5	FDB 197
0933			*FREQUENCY TABLE FOR 48 HZ.
0934			*
0935	EE22	05	FRQ48 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG

SEQ	LOC	OBJ	SOURCE
0936	EE23	02 B6	FDB 694 ; LENGTH OF PULSE PERIOD (USEC)
0937	EE25	00 8E	FDB 3470 ; LENGTH OF 60 DEGREE SEG (USEC)
0938	EE27	01 B1	FDB 433 ; SFMAX
0939	EE29	00 80	FDB 128 ; SFMIN
0940			*PULSE WIDTHS
0941	EE2B	01 69	FDB 361 ; DOM
0942	EE2D	01 3F	FDB 319 ; DOM
0943	EE2F	01 82	FDB 386 ; DOM
0944	EE31	01 08	FDB 264 ; DOM
0945	EE33	01 88	FDB 395 ; DOM
0946	EE35	00 C5	FDB 197
0947			*FREQUENCY TABLE FOR 49 HZ.
0948			*
0949	EE37	05	FRQ49 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0950	EE38	02 A8	FDB 680 ; LENGTH OF PULSE PERIOD (USEC)
0951	EE3A	00 48	FDB 3400 ; LENGTH OF 60 DEGREE SEG (USEC)
0952	EE3C	01 A8	FDB 424 ; SFMAX
0953	EE3E	00 80	FDB 128 ; SFMIN
0954			*PULSE WIDTHS
0955	EE40	01 69	FDB 361 ; DOM
0956	EE42	01 3F	FDB 319 ; DOM
0957	EE44	01 82	FDB 386 ; DOM
0958	EE46	01 08	FDB 264 ; DOM
0959	EE48	01 88	FDB 395 ; DOM
0960	EE4A	00 C5	FDB 197
0961			*FREQUENCY TABLE FOR 50 HZ.
0962			*
0963	EE4C	05	FRQ50 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0964	EE4D	02 98	FDB 667 ; LENGTH OF PULSE PERIOD (USEC)
0965	EE4F	0D 07	FDB 3335 ; LENGTH OF 60 DEGREE SEG (USEC)
0966	EE51	01 9F	FDB 415 ; SFMAX
0967	EE53	00 80	FDB 128 ; SFMIN
0968			*PULSE WIDTHS
0969	EE55	01 69	FDB 361 ; DOM
0970	EE57	01 3F	FDB 319 ; DOM
0971	EE59	01 82	FDB 386 ; DOM
0972	EE5B	01 08	FDB 264 ; DOM
0973	EE5D	01 88	FDB 395 ; DOM
0974	EE5F	00 C5	FDB 197
0975			*FREQUENCY TABLE FOR 51 HZ.
0976			*
0977	EE61	05	FRQ51 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0978	EE62	02 8E	FDB 654 ; LENGTH OF PULSE PERIOD (USEC)
0979	EE64	0C C6	FDB 3270 ; LENGTH OF 60 DEGREE SEG (USEC)
0980	EE66	01 97	FDB 407 ; SFMAX
0981	EE68	00 80	FDB 128 ; SFMIN
0982			*PULSE WIDTHS
0983	EE6A	01 69	FDB 361 ; DOM
0984	EE6C	01 3F	FDB 319 ; DOM
0985	EE6E	01 82	FDB 386 ; DOM
0986	EE70	01 08	FDB 264 ; DOM
0987	EE72	01 88	FDB 395 ; DOM
0988	EE74	00 C5	FDB 197
0989			*FREQUENCY TABLE FOR 52 HZ.
0990			*

SEQ	LOC	OBJ	SOURCE
0991	EE76	05	FRQ52 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
0992	EE77	02 81	FDB 641 ; LENGTH OF PULSE PERIOD (USE
0993	EE79	0C 85	FDB 3205 ; LENGTH OF 60 DEGREE SEG (USE
0994	EE7B	01 8F	FDB 399 ; SFMAX
0995	EE7D	00 80	FDB 128 ; SFMIN
0996			*PULSE WIDTHS
0997	EE7F	01 69	FDB 361 ; DOM
0998	EE81	01 3F	FDB 319
0999	EE83	01 82	FDB 386 ; DOM
1000	EE85	01 08	FDB 264
1001	EE87	01 8B	FDB 395 ; DOM
1002	EE89	00 C5	FDB 197
1003			*FREQUENCY TABLE FOR 53 HZ.
1004			*
1005	EE8B	05	FRQ53 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
1006	EE8C	02 75	FDB 629 ; LENGTH OF PULSE PERIOD (USE
1007	EE8E	0C 49	FDB 3145 ; LENGTH OF 60 DEGREE SEG (USE
1008	EE90	01 87	FDB 391 ; SFMAX
1009	EE92	00 80	FDB 128 ; SFMIN
1010			*PULSE WIDTHS
1011	EE94	01 69	FDB 361 ; DOM
1012	EE96	01 3F	FDB 319
1013	EE98	01 82	FDB 386 ; DOM
1014	EE9A	01 08	FDB 264
1015	EE9C	01 8B	FDB 395 ; DOM
1016	EE9E	00 C5	FDB 197
1017			*FREQUENCY TABLE FOR 54 HZ.
1018			*
1019	EEA0	05	FRQ54 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
1020	EEA1	02 69	FDB 617 ; LENGTH OF PULSE PERIOD (USE
1021	EEA3	0C 0D	FDB 3085 ; LENGTH OF 60 DEGREE SEG (USE
1022	EEA5	01 7F	FDB 383 ; SFMAX
1023	EEA7	00 80	FDB 128 ; SFMIN
1024			*PULSE WIDTHS
1025	EEA9	01 69	FDB 361 ; DOM
1026	EEAB	01 3F	FDB 319
1027	EEAD	01 82	FDB 386 ; DOM
1028	EEAF	01 08	FDB 264
1029	EEB1	01 8B	FDB 395 ; DOM
1030	EEB3	00 C5	FDB 197
1031			*FREQUENCY TABLE FOR 55 HZ.
1032			*
1033	EEB5	05	FRQ55 FCB 5 ; NO. -PULSE PERIODS IN 60 DEG SEG
1034	EEB6	02 5E	FDB 606 ; LENGTH OF PULSE PERIOD (USE
1035	EEB8	0B D6	FDB 3030 ; LENGTH OF 60 DEGREE SEG (USE
1036	EEBA	01 78	FDB 376 ; SFMAX
1037	EEBC	00 80	FDB 128 ; SFMIN
1038			*PULSE WIDTHS
1039	EEBE	01 69	FDB 361 ; DOM
1040	EEC0	01 3F	FDB 319
1041	EEC2	01 82	FDB 386 ; DOM
1042	EEC4	01 08	FDB 264
1043	EEC6	01 8B	FDB 395 ; DOM
1044	EEC8	00 C5	FDB 197
1045			*FREQUENCY TABLE FOR 56 HZ.

SEQ	LOC	OBJ	SOURCE
1046			*
1047	EECA	03	FRQ56 FCB 3 ; NO. -PULSE PERIODS IN 60 DEG SEG
1048	EECB	03 E0	FDB 992 ; LENGTH OF PULSE PERIOD (USEC)
1049	EECD	0B A0	FDB 2976 ; LENGTH OF 60 DEGREE SEG (USEC)
1050	EECF	01 79	FDB 377 ; SFMAX
1051	EED1	00 80	FDB 128 ; SFMIN
1052			*PULSE WIDTHS
1053	EED3	02 69	FDB 617 ; DOM
1054	EED5	01 F7	FDB 503
1055	EED7	02 90	FDB 656 ; DOM
1056	EED9	01 48	FDB 328
1057			*FREQUENCY TABLE FOR 57 HZ.
1058			*
1059	EEDB	03	FRQ57 FCB 3 ; NO. -PULSE PERIODS IN 60 DEG SEG
1060	EEDC	03 CF	FDB 975 ; LENGTH OF PULSE PERIOD (USEC)
1061	EEDE	0B 60	FDB 2925 ; LENGTH OF 60 DEGREE SEG (USEC)
1062	EEE0	01 72	FDB 375 ; SFMAX
1063	EEE2	00 80	FDB 128 ; SFMIN
1064			*PULSE WIDTHS
1065	EEE4	02 69	FDB 617 ; DOM
1066	EEE6	01 F7	FDB 503
1067	EEE8	02 90	FDB 656 ; DOM
1068	EEEA	01 48	FDB 328
1069			*FREQUENCY TABLE FOR 58 HZ.
1070			*
1071	EEEC	03	FRQ58 FCB 3 ; NO. -PULSE PERIODS IN 60 DEG SEG
1072	EEED	03 BE	FDB 958 ; LENGTH OF PULSE PERIOD (USEC)
1073	EEEF	0B 3A	FDB 2874 ; LENGTH OF 60 DEGREE SEG (USEC)
1074	EEF1	01 6C	FDB 364 ; SFMAX
1075	EEF3	00 80	FDB 128 ; SFMIN
1076			*PULSE WIDTHS
1077	EEF5	02 69	FDB 617 ; DOM
1078	EEF7	01 F7	FDB 503
1079	EEF9	02 90	FDB 656 ; DOM
1080	EEFB	01 48	FDB 328
1081			*FREQUENCY TABLE FOR 59 HZ.
1082			*
1083	EEFD	03	FRQ59 FCB 3 ; NO. -PULSE PERIODS IN 60 DEG SEG
1084	EEFE	03 AE	FDB 942 ; LENGTH OF PULSE PERIOD (USEC)
1085	EF00	0B 0A	FDB 2826 ; LENGTH OF 60 DEGREE SEG (USEC)
1086	EF02	01 65	FDB 357 ; SFMAX
1087	EF04	00 80	FDB 128 ; SFMIN
1088			*PULSE WIDTHS
1089	EF06	02 69	FDB 617 ; DOM
1090	EF08	01 F7	FDB 503
1091	EF0A	02 90	FDB 656 ; DOM
1092	EF0C	01 48	FDB 328
1093			*FREQUENCY TABLE FOR 60 HZ.
1094			*
1095	EF0E	03	FRQ60 FCB 3 ; NO. -PULSE PERIODS IN 60 DEG SEG
1096	EF0F	03 9E	FDB 926 ; LENGTH OF PULSE PERIOD (USEC)
1097	EF11	0A DA	FDB 2778 ; LENGTH OF 60 DEGREE SEG (USEC)
1098	EF13	01 5F	FDB 351 ; SFMAX
1099	EF15	00 80	FDB 128 ; SFMIN
1100			*PULSE WIDTHS

SEQ LOC OBJ SOURCE

1101	EF17	02 69	FDB	617	; DOM
1102	EF19	01 F7	FDB	503	
1103	EF1B	02 90	FDB	656	; DOM
1104	EF1D	01 48	FDB	328	
1105			END		

1105 LINES ASSEMBLED, LOC = EF1F, 0000 ERRORS DETECTED.

END ASM V1.1

SEQ LOC OBJ SOURCE

```

0001                                NAM      MCL59
0002                                *****
0003                                * MCL - MAIN CONTROL LOOP FOR PULSE WIDTH MODULATION *
0004                                *
0005                                *****
0006                                *
0007                                * THIS MODULE CONTAINS THE MAIN CONTROL LOOP TO
0008                                * PERFORM THE FOLLOWING FUNCTIONS
0009                                *
0010                                * . INITIALISATION AT PWR ON
0011                                * . COMPUTATION OF E*
0012                                * . OVERLOAD DETECTION OF FEEDBACK
0013                                * . KEYBOARD MONITOR FOR MANUAL CONTROL
0014                                * . RAMP UP/ RAMP DOWN IN FREQ.
0015                                * . LOW VOLTAGE B+ DETECTION
0016                                *
0017                                * THIS MODULE IS EXECUTED ONLY DURING TIME MPU IS
0018                                * / WAITING / ON AN INTERRUPT.
0019                                *
0020                                * INTERRUPTS OCCUR EVERY PULSE PERIOD AND VECTOR
0021                                * PROCESSING TO PTIS ( $E000 )
0022                                * THIS SCHEME IS ADOPTED FOR ALL PWM MODIFICATION
0023                                * EVERY PULSE PERIOD BASED ON B+
0024                                *
0025                                * MEMORY LOCATIONS USED BY THIS MODULE
0026                                *
0027                                * (GLOBAL)
0028                                0001 MSTOP EQU $0001 ; UNIT STOPPED INDIC. (1=STOPPED)
0029                                0002 MSTART EQU $0002 ; START UNIT CMND (1= TRY TO START)
0030                                0003 OVRLO EQU $0003 ; OVERLOAD FLAG(1= OVERLOAD EXISTS)
0031                                0004 INHIB EQU $0004 ; RESTART INHIBIT FLAG(1= INHIBIT)
0032                                0044 PIACMD EQU $0044 ; PIA CMD (KEEPS FF PR/CLR:SC-PB2)
0033                                0046 FCMD EQU $0046 ; COMMANDED OUTPUT FREQUENCY
0034                                004C FOUT EQU $004C ; PRESENT OUTPUT FREQUENCY
0035                                003A VSRW EQU $003A ; V* INPUT FROM FROM MPU B
0036                                004A BPRW EQU $004A ; B+ FROM A/D
0037                                0038 FCRW EQU $0038 ; SPEED CMD FROM A/D NOT DETECTED
0038                                0020 DOMID EQU $0020 ; DOMINANT'S ID (0=A, 1=B, 2=C)
0039                                0022 DPF EQU $0022 ; DOMINANT'S POLARITY ( 0=+, FF=-)
0040                                0024 J EQU $0024 ; FLAG/WAY TO USE COMPL.
0041                                00C0 F60DEG EQU $00C0 ; START 60 DEG SEG FLAG(1=START)
0042                                * (+=1ST ODD/2ND EVEN, -=1ST EVEN/2ND ODD)
0043                                00C8 FTINCR EQU $00C8 ; FREQ TAB POSN. (+4=1ST HALF, -4=2ND)
0044                                0051 MONFG EQU $0051 ; MON CONNECTED FLAG (1=CONN, 0=NO)
0045                                00C5 NPPCT EQU $00C5 ; NO. OF P. P. 'S LEFT IN SEGMENT
0046                                00E6 NORFG EQU $00E6 ; NORM OPER. FLAG (0=USE NORM, 1=KE)
0047                                00E7 VSFLG EQU $00E7 ; V* FLAG (0=USE MPB, 1=FROM KBD)
0048                                E000 PTIS EQU $E000 ; TIMER INTERRUPT SERVICE ROUTINE
0049                                00C1 FTPNW EQU $00C1 ; NEW FREQUENCY TABLE POINTER
0050                                0006 OVLD1 EQU $0006 ; OVRLO COND FLAG(1=OVLD, 10%COL<1)
0051                                00E8 INDEX EQU $E8 ; FREQ TABLE INDEX MS OF ADDRESS
0052                                00C3 FTPTR EQU $00C3 ; FREQ TAB POINTER-PRESENT POSITI
0053                                6000 ACIAC EQU $6000 ; ACIA CONTROL REGISTER

```

```

**SEQ LOC OBJ SOURCE
--0055 0009 DIRCMD EQU $0009 ;CCW DIR CMD FLG (1=CCW)
..0056 0010 DIROUT EQU $0010 ;CCW DIR OUT FLG (1=CCW)
0057 *
-0058 * (LOCALS)
0059 *
0060 0050 BPFLG EQU $0050 ;B+ FLAG (0=B+<80, 1=B+>80)
0061 0048 VSTAR EQU $0048 ;VSTAR CONTROLLED BY KEYBOARD
0062 00EA ESOC EQU $00EA ;ADDRESS OF E* OVERLOAD CALC VALUE
0063 0005 K EQU $05 ;K=32; FILTER CONSTANT, FC=XXX HZ
0064 00F0 AVE EQU $00F0 ;ADDRESS LOC FOR FILTER YN
0065 0001 ESMM EQU $01 ;E* MAX MS VALUE (195)
0066 0000 ESML EQU $00 ;E* MAX LS
0067 0000 ES1MM EQU $00 ;E* LOWER MAX MS (65)
-0068 0080 ES1ML EQU $80 ;E* " " LS
0069 003C ESTAR EQU $003C ;E* VALUE - NOW
0070 00F0 YN EQU AVE ;E* AVE PREVIOUS CALCULATION
0071 00CA FTPNT EQU $00CA ;TEMP LOC FOR NEW POINTER ADDR
0072 0040 ESRT EQU $0040 ;E* TEMP LOC FOR SF (1+V*/8)
-0073 00F0 OVCMP EQU $00F0 ;OVERLOAD COMPARISON VALUE
0074 00D8 VPADDR EQU $00D8 ;V* PEAK TABLE POINTER ADDR
0075 00DB VSREF EQU $00DB ;V* MAX FOR "F"
0076 00EF OUTCTL EQU $EF ;V* OUT OF CONTROL - LS
0077 0000 OUTCTM EQU $00 ;V* OUT OF CONTROL - MS
-0078 005A SHTDN EQU $005A ;ADDR SHUT DOWN FLAG
0079 *
0080 *
0081 * (I/O ADDRESSES)
0082 *
-0083 5000 PIA EQU $5000 ;PIA ADDRESS
0084 4000 PTM1 EQU $4000 ;PTM1 ADDRESS
-0085 3000 PTM2 EQU $3000 ;PTM2 ADDRESS
0086 2007 WLATCH EQU $2007 ;W LATCH FOR MPB
0087 6000 ACIAS EQU $6000 ;ACIA STATUS REG
0088 6001 ACIAD EQU $6001 ;ACIA DATA REG
0089 *
0090 *
0091 *
0092 0001 LSYMB 1

```


SEQ	LOC	OBJ	SOURCE
0094			*
0095			*****
0096			* PROGRAM INITIALIZATION ROUTINE *
0097			* *****
0098			*
0099		D800	ORG \$D800
0100			*
0101			* ZERO RAM -- LOCS 0 THRU \$03FF
0102			* PRESERVE MONITOR CONNECTED FLAG \$0051
0103			* NORMAL OP FLAG \$00E6
0104			* V* FLAG \$00E7
0105			*
0106	D800	0F	INIT SEI ; DISABLE INTERRUPTS
0107	D801	96 51	LDA A MONFG ; LOAD MONITOR CONN FLAG
0108	D803	DE E6	LDX NORFG ; LOAD NORM OP & V* FLAGS
0109	D805	B7 FFED	STA A \$FFED ; TEMP STORE FLAG
0110	D808	FF FFEE	STX \$FFEE ; TEMP STORE FLAGS
0111	D80B	4F	CLR A ; SET CLR VALUE
0112	D80C	CE 0000	LDX #0 ; LOAD BOTTOM ADDRESS
0113	D80F	A7 00	INIT1 STA A 00,X ; CLEAR RAM LOC
0114	D811	08	INX ; CHANGE ADDRESS
0115	D812	8C 03FF	CPX #\$03FF ; TEST FOR LAST ADDRESS DONE
0116	D815	26 F8	BNE INIT1 ; SKIP IF MORE ADDRESSES
0117	D817	B6 FFED	LDA A \$FFED ; BRING FLAG BACK
0118	D81A	CE FFEE	LDX \$FFEE ; BRING FLAGS BACK
0119	D81D	97 51	STA A MONFG ; RESTORE MON CONN FLAG
0120	D81F	DF E6	STX NORFG ; RESTORE NORM OP & V* FLAGS
0121			*
0122			* SET THE STACK POINTER
0123			*
0124	D821	8E FF00	LDS #\$FF00
0125			*
0126			* SET IRQ POINTER TO THE START OF THE PULSE TIMER
0127			* MODULE INTERRUPT SERVICE ROUTINE
0128			*
0129	D824	CE E000	LDX #PTIS ; GET VALUE
0130	D827	FF FFF8	STX \$FFF8 ; SET IRQ VECTOR
0131			*
0132			* INITIALIZE THE TIMERS AND SET THE UNIT STOPPED FLAG
0133			*
0134	D82A	BD DAB0	JSR TRESET ; TIMER RESET ROUTINE
0135	D82D	86 01	LDA A #1
0136	D82F	97 01	STA A MSTOP ; SET UNIT STOPPED FLAG
0137			*
0138			* SET OVERLOAD INDICATOR OFF(VIA PIA) BY "PMON" ROUTINE
0139			* PIA HAS BEEN SET UP PA = ALL INPUTS
0140			* PB = 6 OUTPUTS + 1 INPUT
0141			*
0142	D831	86 34	LDA A #\$34 ; SET OC WORD FORMAT
0143	D833	B7 5003	STA A PIA+3 ; TURN OL OFF THRU CRB OF PIA
0144			* ENABLE INTERRUPTS AND ENTER THE MAIN CONTROL LOOP
0145			*
0146	D836	0E	CLI
0147	D837	20 47	BRA MCL

63

SEQ	LOC	OBJ	SOURCE
0200			*
0201			*
0202			*
0203			* CALL THE OVERLOAD PROTECTION ROUTINE TO PROTECT
0204			* THE UNIT IN CASE OF OVERLOAD
0205			* TESTS V* TO DETERMINE IF OVERLOAD EXISTS
0206			* TESTS B+ TO DETERMINE IF TOO LOW (<80 VOLTS)
0207			*
0208			*
0209	D89E	BD DC00	JSR PROTCT ; OPERATION PROTECT
0210			*
0211			* TEST PROTECT FLAGS TO DETERMINE IF FREQUENCY
0212			* SHOULD CHANGE
0213			*
0214			* IF MAX OVERLOAD EXISTS ----- RAMPDOWN
0215			* IF B+ < 80 V OR B+ > 140 V -- RAMPDOWN
0216			* IF OVERLOAD CONDITION 1 ---- DON'T CHANGE FREQUENCY
0217			*
0218			* IF NO PROTECT FLAGS TEST FREQUENCY
0219			* FOUT < FCMD ----- RAMPUP
0220			* FOUT > FCMD ----- RAMPDOWN
0221			* FOUT = FCMD ----- DON'T CHANGE FREQUENCY
0222			*
0223			* IF FCMD = 0 THEN RAMP DOWN AT THE NORMAL RATE
0224			*
0225	D8A1	96 46	LDA A FCMD ; LD FREQ COMMAND
0226	D8A3	27 12	BEQ MCL03 ; SKIP IF ZERO
0227			*
0228			* MAX OVERLOAD TEST
0229			*
0230	D8A5	96 03	LDA A OVRLD ; OVERLOAD INDICATOR
0231	D8A7	26 56	BNE OLRMPD ; SKIP FOR OVERLOAD RAMPDOWN
0232			*
0233			*
0234			*TEST TO SEE IF B+ INPUT IS LOW (<80 V) OR HIGH (>140 V)
0235			* IF IT IS, WE MUST TURN OFF BEFORE THE POWER SUPPLY
0236			* CRASHES.
0237			*
0238	D8A9	96 50	LDA A BPFLG ; LD B+ FLAG
0239			* IF SET, RAMPDOWN
0240	D8AB	26 0A	BNE MCL03 ; BRANCH IF B+<80 OR >140V
0241			*
0242			* DETERMINE IF WE'RE AT COMMANDED OUTPUT FREQUENCY
0243			* COMPARE THE COMMANDED UNIT SPEED WITH THE ACTUAL
0244			* OUTPUT FREQUENCY - IF LOW RAMPUP
0245			* IF HIGH RAMPDOWN
0246			* IF THERE, DO NOTHING
0247			*
0248	D8AD	96 46	LDA A FCMD ; COMMANDED FREQUENCY
0249	D8AF	90 4C	SUB A FOUT ; SUBTRACT OUTPUT FREQUENCY
0250	D8B1	27 1C	BEQ TFOUT ; SKIP IF THERE(NO F CHANGE)
0251	D8B3	28 02	BMI MCL03 ; SKIP IF HIGH (GO DOWN IN F)
0252	D8B5	20 11	BRA MCL04 ; SKIP TO RAMP UP

- ASM V1.1

```

* SEQ LOC OBJ SOURCE
0254 *
0255 *DECISION MADE TO GO DOWN IN FREQUENCY
0256 * WE GOT HERE BECAUSE 1. FCMD < FOUT
0257 * OR 2. OVERLOAD EXISTS
0258 * OR 3. B+ < 80 VOLTS
0259 D8B7 B0 DB80 MCL03 JSR RAMPDN ; RAMP DOWN FOUT
0260 D8BA 96 4C MCL3A LDA A FOUT ; LD OUTPUT FREQ
0261 D8BC 26 11 BNE TFOUT ; SKIP IF NOT "0"
0262 *IF FOUT =0 FORCED BY OVERLOAD, SET RESTART INHIBIT FLAG
0263 *
0264 D8BE 96 03 LDA A OVRLO ; GET FLAG
0265 D8C0 27 0D BEQ TFOUT ; SKIP IF OVRLO DIDN'T CAUSE
0266 D8C2 86 01 LDA A #1 ; SET FLAG VALUE
0267 D8C4 97 04 STA A INHIB ; SET FLAG
0268 D8C6 20 07 BRA TFOUT ; SEE IF FOUT=0
0269 *
0270 *DECISION MADE TO GO UP IN FREQUENCY
0271 * WE GOT HERE BECAUSE 1. FCMD > FOUT
0272 * AND 2. OVERLOAD DOESN'T EXIST
0273 * AND 3. B+ > 80 VOLTS
0274 * MUST VERIFY PARTIAL OVERLOAD DOESN'T EXIST
0275 *
0276 D8C8 96 06 MCL04 LDA A OVRLO1 ; LD FLAG
0277 D8CA 26 03 BNE TFOUT ; SKIP IF PARTIAL OVERLOAD
0278 D8CC BD DB00 JSR RAMPUP ; ELSE, RAMP UP OUTPUT FREQ
0279 *
0280 *
0281 * NEXT STEP AFTER CHANGING FOUT IS TO ADJUST THE FREQUENCY
0282 * TABLE POINTER IF WE'RE NOT AT "0" FCMD
0283 *
0284 * TEST THE OUTPUT FREQUENCY.
0285 *
0286 *
0287 D8CF 96 4C TFOUT LDA A FOUT ; OUTPUT FREQUENCY
0288 D8D1 27 23 BEQ STUNT ; IF=0, STOP THE UNIT
0289 *
0290 * THIS SECTION FINDS & STORES THE ADDRESS OF THE FREQUENCY
0291 * TABLE INDEX CORRESPONDING TO THE OUTPUT FREQUENCY
0292 * HENCE POINTS TO LOCATION FOR PW PARAMETERS (PTIS USES)
0293 *
0294 D8D3 D6 4C CHPTR LDA B FOUT ; OUTPUT FREQ
0295 D8D5 C0 06 SUB B #6 ; GET RID OF OFFSET
0296 D8D7 2A 02 BPL AINDX ; SKIP IF > 6 HZ
0297 D8D9 C6 00 LDA B #00 ; SET TO MIN F=6
0298 D8DB 58 AINDX ASL B ; ACCOUNT FOR 2 BYTES/F
0299 D8DC 86 E8 LDA A #INDEX ; LD MS OF INDEX
0300 D8DE D7 CB STA B FTPNT+1 ; STORE LS OF ADDR
0301 D8E0 97 CA STA A FTPNT ; STORE MS OF ADDR
0302 * CHANGE ADDRESS POINTER NOW THAT ITS' GENERATED
0303 D8E2 DE CA LDX FTPNT ; LD NEW POINTER
0304 D8E4 EE 00 LDX 0,X ; GET TABLE ADDRESS
0305 D8E6 DF C1 STX FTPNW ; STORE NEW FT POINTER
0306 *
0307 * IF THE START UNIT FLAG IS SET, START THE UNIT FLAG

```

SEQ	LOC	OBJ	SOURCE
0308			*
0309	D8E8	96 02	LDA A MSTART ; START UNIT COMMAND FLAG
0310	D8EA	27 94	BEQ MCL ; IF ZERO, DON'T START
0311			*
0312			* START UNIT LOGIC
0313			* -----
0314			*
0315	D8EC	8D DA20	JSR TSTART ; TIMER STARTUP ROUTINE
0316	D8EF	4F	CLR A
0317	D8F0	97 02	STA A MSTART ; CLR START UNIT FLAG
0318	D8F2	97 01	STA A MSTOP ; CLR UNIT STOPPED INDIC
0319	D8F4	20 8A	MCLCN BRA MCL
0320			*
0321			*
0322			* STOP UNIT LOGIC
0323			* -----
0324			*
0325	D8F6	8D DAB0	STUNT JSR TRESET ; TIMER RESET ROUTINE
0326	D8F9	7E DDA6	JMP STROUT ; STOP ROUTINE
0327	D8FC	01	NOP
0328	D8FD	20 F5	BRA MCLCN ; BRA TO MCL CONNECT
0329			*
0330			* MAX OVERLOAD - RAMPDOWN SLOWLY
0331			*
0332			* IF SHUT DOWN FLAG SET - FORCE FOUT = 0
0333			*
0334			*
0335	D8FF	86 00	OLRMPD LDA A #\$00
0336	D901	27 05	BEQ OLRMP1 ; SKIP IF IN CONTRL
0337	D903	7F 004C	CLR FOUT ; SET FOUT=0
0338	D906	20 B2	BRA MCL3A ; SKIP TO CONT SHUT DOWN
0339	D908	8D DB91	OLRMP1 JSR SLWRD ; CALL RAMP DOWN SLOWLY
0340	D90B	20 AD	BRA MCL3A ; SKIP TO TEST FREQ

SEQ	LOC	OBJ	SOURCE
0342			* * * * *
0343			* TSTART--TIMER STARTUP ROUTINE *
0344			* * * * *
0345			*
0346			* THIS ROUTINE PERFORMS THE LOGIC NECESSARY TO INITIALIZE
0347			* THE TWO PULSE TIMER MODULES (PTM) AND BEGIN THE PWM OUTPUT.
0348			*
0349	DA20		ORG \$DA20
0350			*
0351	DA20 0F		TSTART SEI ;DISABLE INTERRUPTS
0352			*
0353			* SET ALL THREE PTM #1 TIMERS FOR SINGLE SHOT MODE
0354			* AND APPLY INTERNAL RESET (IR)
0355			*
0356			* CONTROL WORD: 1011001X=SINGLE-SHOT MODE, USE ENABLE CLOCK
0357			* (=B2(HEX))16 BIT MODE, RESET ON GATE .OR. RESET
0358			* INTERRUPT DISABLED/OUTPUT ENABLED
0359			*
0360	DA21 86 B2		LDA A #\$B2 ;SET THE CONTROL WORD
0361	DA23 B7 4001		STA A PTM1+1 ;INIT TIMER #2
0362	DA26 B7 4000		STA A PTM1 ;INIT TIMER #3
0363	DA29 8A 01		ORA A #1 ;SET CTRL WD TO ENB #1
0364	DA2B B7 4001		STA A PTM1+1 ;ENABLE WRITE TO CR 1
0365	DA2E B7 4000		STA A PTM1 ;INIT TIMER #1 AND APPLY IR
0366			*
0367			* LOAD PTM#1 LATCHES WITH MAX VALUE
0368			*
0369	DA31 CE FFFF		LDX #\$FFFF ;MAX COUNT
0370	DA34 FF 4002		STX PTM1+2 ;TO TIMER#1 LATCHES
0371	DA37 FF 4004		STX PTM1+4 ;TO TIMER#2 LATCHES
0372	DA3A FF 4006		STX PTM1+6 ;TO TIMER#3 LATCHES

SEQ	LOC	OBJ	SOURCE
0374			*
0375			* PULSE THE INTERNAL RESET BIT TO LOAD THE COUNTERS
0376			* FROM THE LATCHES
0377			*
0378	DA3D	84 FE	AND A #\$FE ; REMOVE IR
0379	DA3F	B7 4000	STA A PTM1 ; WRITE IT TO PTM#1
0380	DA42	8A 01	ORA A #1 ; RE-APPLY IR
0381	DA44	B7 4000	STA A PTM1 ; WRITE IT TO PTM#1
0382			*
0383			*
0384			*
0385			* INITIALIZE PTM#2 AND APPLY INTERNAL RESET TO PTM#2
0386			* TIMER #1: CONTINUOUS MODE, USE ENABLE CLOCK, 16 BIT MODE
0387			* RESET ON GATE .OR. RESET
0388			* INTERRUPT AND OUTPUT ENABLED
0389			* (USED AS INTERVAL TIMER FOR PWM)
0390			*
0391			* TIMER #2: NOT USED
0392			*
0393			* TIMER #3 CONTINUOUS MODE, USE ENABLE CLOCK, 16 BIT MODE
0394			* RESET ON GATE .OR. RESET
0395			* INTERRUPT DISABLED, OUTPUT ENABLED
0396			* (USED AS BAUD RATE GENERATOR FOR ACIA)
0397			*
0398			* CONTROL WORD : 1101001X FOR ABOVE CONDITIONS
0399			* (=D2<HEX>)
0400			* TIMER #1 - USED AS INTERVAL (PULSE PERIOD LENGTH) FOR PWM
0401			*
0402	DA47	5F	CLR B
0403	DA48	F7 3001	STA B PTM2+1 ; WRITE CR2 TO SELECT CR3
0404	DA4B	C6 92	LDA B #\$92 ; TIMER #3 CONTROL WORD
0405	DA4D	F7 3000	STA B PTM2 ; INIT TIMER #3
0406	DA50	C6 01	LDA B #1
0407	DA52	F7 3001	STA B PTM2+1 ; SELECT CR1
0408	DA55	C6 03	LDA B #\$03 ; TIMER #1 CONTROL WORD
0409	DA57	F7 3000	STA B PTM2 ; INIT TIMER#1 AND APPLY IR
0410			*
0411			* LOAD PTM #2 TIMER #1 LATCHES WITH 794 MICROSECOND COUNT
0412			* COUNT FOR TPP WITH F=6 HZ
0413			*
0414	DA5A	CE 031A	LDX #794
0415	DA5D	FF 3002	STX PTM2+2 ; COUNT TO TIMER#1 LATCHES
0416			*
0417			* PULSE THE INTERNAL RESET (IR) BIT TO LOAD PTM#2 COUNTERS
0418			*
0419			*
0420	DA60	C4 FE	AND B #\$FE ; REMOVE INTERNAL RESET
0421	DA62	F7 3000	STA B PTM2 ; WRITE IT TO CR 1
0422	DA65	CA 01	ORA B #1 ; RE-APPLY IR
0423	DA67	F7 3000	STA B PTM2 ; WRITE IT TO CR1

SEQ	LOC	OBJ	SOURCE
0425			*
0426			* SET UP RAM LOCATIONS USED BY PULSE TIMER INTERRUPT SERVICE
0427			* ROUTINE (PTIS) TO START AT BEGINNING OF 60 DEGREE SEGMENT.
0428			*
0429	DA6A	86 01	LDA A #+1 ; SET FLAG VALUE
0430	DA6C	97 C0	STA A F60DEG ; SET FLAG-60 DEG BOUND
0431	DA6E	97 20	STA A DOMID ; SET DOM ID =1=PHASE B
0432	DA70	97 22	STA A DPF ; SET LAST PULARITY= +
0433	DA72	97 24	STA A J ; SET INDEX TO 1ST HALF
0434	DA74	CE 0004	LDX #+4 ; SET INC FOR FREQ TAB(1ST HALF)
0435	DA77	DF C8	STX FTINCR ; SET IN MEM LOC
0436	DA79	86 A6	LDA A #\$A6 ; SET PR/RST FF CMD
0437	DA7B	B7 5002	STA A PIA+2 ; ENB FF'S TO PRESET
0438	DA7E	97 44	STA A PIACMD ; SAVE CMD
0439			* SECTION SETS UP FREQUENCY TABLE POINTERS
0440	DA80	CE E880	LDX #\$E880 ; TOP OF TABLE(6 HZ)
0441	DA83	DF C1	STX FTPNW ; SET NEW POINTER ADDR
0442	DA85	DF C3	STX FTPTR ; SET EXISTING POINTER
0443	DA87	CE E800	LDX #\$E800 ; TOP OF INDEX
0444	DA8A	DF CA	STX FTPNT ; SET TEMP ADDR
0445			*
0446			*
0447			* REMOVE THE INTERNAL RESETS TO ALLOW BOTH TIMERS TO OPERATE
0448			* RESTORE THE INTERRUPT FLAG AND RETURN
0449			*
0450	DA8C	86 B2	LDA A #\$B2 ; CTRL WD PTM#1 T1
0451	DA8E	C6 D2	LDA B #\$D2 ; CTRL WD PTM#2 T1
0452	DA90	B7 4000	STA A PTM1 ; REMOVE IR FROM PTM#1
0453	DA93	F7 3000	STA B PTM2 ; REMOVE IR FROM PTM#2
0454			*
0455	DA96	0E	CLI ; ENABLE INTERRUPTS
0456	DA97	39	RTS ; AND RETURN

SEQ	LOC	OBJ	SOURCE
0458			* * * * *
0459			* TRESET--TIMER RESET ROUTINE *
0460			* * * * *
0461			*
0462		DAB0	ORG \$DAB0
0463			*
0464			* THIS ROUTINE CONTAINS THE LOGIC TO RESET THE TIMERS AND
0465			* STOP THE PWM OUTPUT
0466			*
0467	DAB0	0F	TRESET SEI ; DISABLE INTERRUPTS
0468			*
0469			*
0470			* DISABLE OUTPUT OF PTM#2 TIMER #1
0471			* ALLOW TIMER #3 TO OPERATE
0472			*
0473	DAB1	86 01	LDA A #1
0474	DAB3	B7 3001	STA A PTM2+1 ; SELECT CR 1
0475	DAB6	86 12	LDA A #\$12 ; CODE TO DISABLE OUTPUT&IRQ
0476	DAB8	B7 3000	STA A PTM2 ; STOP TIMER #1
0477			*
0478			*
0479			* APPLY INTERNAL RESET TO CLEAR PTM#1
0480			*
0481	DAB8	86 B3	LDA A #\$B3
0482	DABD	B7 4001	STA A PTM1+1 ; SELECT CR1
0483	DAC0	B7 4000	STA A PTM1 ; APPLY INTERNAL RESET
0484			*
0485			*
0486			* OUTPUT FOUT=0 TO 'W' LATCH
0487			*
0488	DAC3	4F	CLR A
0489	DAC4	B7 2007	STA A WLATCH ; OUTPUT TO W LATCH
0490			*
0491			* RESTORE INTERRUPT MASK FLAG AND RETURN
0492			*
0493	DAC7	0E	CLI ; ENABLE INTERRUPTS
0494	DAC8	39	RTS

SEQ	LOC	OBJ	SOURCE
0496			*
0497			* RAMP UP ROUTINE
0498			*
0499	DB00		ORG \$DB00
0500			*
0501	DB00	D6 4D	RAMPUP LDA B FOUT+1 ; LD FOUT LS BYTE
0502	DB02	96 4C	LDA A FOUT ; LD MS OF FOUT
0503	DB04	C8 01	ADD B #01 ; INCR FOUT BY +1(WAS 2)
0504	DB06	89 00	ADC A #00 ; INCR MS
0505	DB08	81 3C	CMP A #60 ; TEST FOR FOUT>60
0506	DB0A	2A 05	BPL SMOUT ; BRANCH IF FOUT>60
0507	DB0C	D7 4D	SFOUT STA B FOUT+1 ; STORE NEW LS OF FOUT
0508	DB0E	97 4C	STA A FOUT ; STORE FOUT MS BYTE
0509	DB10	39	ERU RTS ; RETURN FROM RAMPUP
0510	DB11	5F	SMOUT CLR B ; CLEAR FOUT LS BYTE
0511	DB12	86 3C	LDA A #60 ; LD FOUT MS=60
0512	DB14	20 F6	BRA SFOUT ; BRANCH TO STORE FOUT
0513			*
0514			*
0515			* RAMP DOWN ROUTINE
0516			*
0517	DB80		ORG \$DB80
0518			*
0519	DB80	D6 4D	RAMPDN LDA B FOUT+1 ; LD FOUT LS BYTE
0520	DB82	96 4C	LDA A FOUT ; LD FOUT MS
0521	DB84	C0 01	SUB B #01 ; DECR FOUT LS BY -1(WAS 2)
0522	DB86	82 00	SBC A #00 ; DECR FOUT MS
0523	DB88	2A 02	BPL ERD ; BRANCH IF FOUT IS +
0524	DB8A	4F	CLR A ; LD 0 FOR FOUT
0525	DB8B	5F	CLR B ; LD LS OF FOUT
0526	DB8C	D7 4D	ERD STA B FOUT+1 ; STORE FOUT LS
0527	DB8E	97 4C	STA A FOUT ; STORE 0 FOR FOUT MS
0528	DB90	39	RTS ; RETURN FROM RAMPDN
0529			*
0530			* SLOW RAMP DOWN ROUTINE - USED ONLY IN OVERLOAD
0531			*
0532	DB91	D6 4E	SLWRD LDA B FOUT+2 ; LD LS OF FOUT
0533	DB93	96 4D	LDA A FOUT+1 ; LD MID BYTE OF FOUT
0534	DB95	C0 80	SUB B #\$80 ; DECR FOUT LS
0535	DB97	82 00	SBC A #\$00 ; DECR MID BYTE
0536	DB99	D7 4E	STA B FOUT+2 ; SAVE LS FOUT
0537	DB9B	97 4D	STA A FOUT+1 ; SAVE MID
0538	DB9D	D6 4C	LDA B FOUT ; LD MS OF FOUT
0539	DB9F	C2 00	SBC B #\$00 ; DECR MS OF FOUT
0540	DBA1	2A 07	BPL ESRD ; SKIP TO END SLOW RAMP
0541	DBA3	5F	CLR B ; CLR FOUT VALUE
0542	DBA4	7F 004D	CLR FOUT+1 ; CLR MID
0543	DBA7	7F 004E	CLR FOUT+2 ; CLR LS
0544	DBAA	D7 4C	ESRD STA B FOUT ; SAVE MS OF FOUT
0545	DBAC	39	RTS ; RETURN TO MAIN STREAM

```

SEQ    LOC    OBJ    SOURCE

0547      *
0548      *
0549      * * * * *
0550      * PROTECTION ROUTINE
0551      * * * * *
0552      *
0553      DC00      ORG      $DC00
0554      *
0555      * THIS SUBROUTINE CONTROLS FLAGS USED BY THE MAIN
0556      * PROGRAM CONTROL LOOP TO DETERMINE IF CONTROLLER
0557      * IS OPERATING WITHIN DESIRED CONSTRAINTS.
0558      *
0559      * OPERATIONAL CONSTRAINTS EXAMINED BY THIS SUBROUTINE :
0560      *   B+ -- INPUT VOLTAGE
0561      *       NORMAL OPERATION 80V < B+ < 140V
0562      *       LOW VOLTAGE                      B+ < 80V SET FLAG-RAMPDOWN
0563      *       HIGH VOLTAGE                     B+ > 140V SET FLAG-RAMPDOWN
0564      *
0565      *   V* -- FEEDBACK ERROR                      FLAG /OL LAMP/FOUT CTRL
0566      *   V* > ESMM                                SET      ON      RAMPDOWN
0567      *   VSREF< V* < ESMM                        RESET   ON      FOUT=FOUT
0568      *   V* < VSREF                               RESET   OFF     FOUT=FCMD
0569      *
0570      * V* IS ERROR READ IN DIGITAL INTERFACE
0571      *
0572      DC00 06 4A  PROTCT  LDA B  BPRAW      ;B+ RAW DATA
0573      DC02 C1 50      CMP B  #80          ;B+ - 80 V
0574      DC04 23 09      BLS  PVOLT          ;SKIP IF TOO LOW
0575      DC06 C1 B4      CMP B  #1B4        ;B+ - 140 V
0576      DC08 22 05      BHI  PVOLT          ;SKIP IF TOO HIGH
0577      DC0A 7F 0050    CLR  BPFLG         ;CLR B+ PROBLEM FLAG
0578      DC0D 20 04      BRA  ESPRT         ;SKIP FOR E* FILTER
0579      *
0580      * B+ PROBLEM DETECTED
0581      *   B+ IS EITHER LOW OR HIGH. IN EITHER CASE
0582      *   WANT TO CAUSE RAMPDOWN BEFORE POWER SUPPLY
0583      *   QUIT.
0584      DC0F 86 01  PVOLT  LDA A  #1          ;SET FLAG VALUE
0585      DC11 97 50      STA A  BPFLG        ;SET FLAG

```

SEQ	LOC	OBJ	SOURCE
0587			*
0588			*NEXT STEP IS V* (OVERLOAD) PROTECTION
0589			*
0590			* TEST 1ST TO SEE IF WE'RE OUT OF CONTROL
0591	DC13	96 48	ESPRT LDA A VSTAR ; LD V* MS
0592	DC15	06 49	LDA B VSTAR+1 ; LD V* LS
0593	DC17	00 EF	SUB B #OUTCTL ; SUBT LS LIMIT
0594	DC19	82 0D	SBC A #OUTCTM ; SUBT MS LIMIT
0595	DC1B	2A 51	BPL OTCT ; SKIP IF > LIMIT
0596			*DO COMPARISON OF V* TO TABLE VALUE FOR VSREF
0597	DC1D	96 4C	LDA A FOUT ; LD OUTPUT FREQUENCY
0598	DC1F	48	ASL A ; ACCOUNT FOR 2BYTES/F
0599	DC20	06 E4	LDA B #\$E4 ; LD V* PEAK MS ADDR
0600	DC22	07 DB	STA B VSREF ; STORE ADDR MS
0601	DC24	97 DC	STA A VSREF+1 ; STORE ADDR LS
0602	DC26	0E DB	LDX VSREF ; LD V* REF ADDR
0603			* NOW DO COMPARISON
0604	DC28	86 0C	LDA A #\$0C ; LD MAX CURRENT
0605	DC2A	06 00	LDA B #\$00
0606	DC2C	00 49	SUB B VSTAR+1 ; VSREF - V* LS
0607	DC2E	92 48	SBC A VSTAR ; VSREF - V* MS
0608	DC30	2A 1F	BPL NOLC ; SKIP IF NO OVR LD

SEQ	LOC	OBJ	SOURCE
0610			*
0611			* DO PROTECT USING COMPARISON VALUE
0612			*
0613	DC32	D7 F1	PRCT1 STA B OVCMP+1 ; TEMP STORE LS -COMPARISON
0614	DC34	97 F0	STA A OVCMP ; TEMP STORE MS -COMPARISON
0615			* TESTS IDENTIFY RANGE OF V* AVERAGE ERROR
0616			* 1ST TEST - IS IT > ES1MM ?
0617			*
0618	DC36	08 80	ADD B #ES1ML ; CMP - LS LOWER REF
0619	DC38	89 00	ADC A #ES1MM ; CMP - MS LOWER REF
0620	DC3A	2A 0A	BPL OLC1 ; SKIP IF COND 1
0621			* WE HAVE OVERLOAD - BUT HOW BAD
0622			* 2ND TEST - IS IT > ESMM ?
0623			*
0624			* IF FCMD=0 WITH OVERLOAD (COND 1 OR 2) - SHUTDOWN !
0625			*
0626	DC3C	D6 F1	LDA B OVCMP+1 ; RELOAD LS
0627	DC3E	96 F0	LDA A OVCMP ; RELOAD MS
0628	DC40	08 00	ADD B #ESML ; CMP - E* MAX LS
0629	DC42	89 01	ADC A #ESMM ; CMP - E* MAX MS
0630	DC44	2B 1E	BMI OLCM ; SKIP IF MAX OVERLOAD
0631			* OVERLOAD IS VSREF < V* < ESMM
0632			* TURN ON OVERLOAD LAMP, SET OVERLOAD CONDITION 1 FLAG
0633			* DON'T ALLOW FREQUENCY TO CHANGE (UP OR DOWN)
0634	DC46	86 3C	OLC1 LDA A ##3C ; LD OC LAMP ON CTRL WD
0635	DC48	B7 5003	STA A PIA+3 ; TURN LAMP ON
0636	DC4B	97 06	STA A OVLD1 ; SET OVRD COND 1 FLAG
0637	DC4D	7F 0003	OLC1A CLR OVRD ; CLR OVRD FREQ CTRL
0638	DC50	39	RTS
0639			* NO OVERLOAD EXISTS. CLEAR BOTH OVERLOAD FLAGS
0640			* AND TURN OFF OVERLOAD LAMP
0641			*
0642	DC51	96 04	NOLC LDA A INHIB ; LD RESTART INHIBIT
0643	DC53	26 0E	BNE NOLCC ; SKIP PREVENT OLD CLR
0644	DC55	86 34	LDA A ##34 ; LD OC LAMP CONTROL WD
0645	DC57	B7 5003	STA A PIA+3 ; TURN OL INDICATOR OFF
0646	DC5A	7F 0003	CLR OVRD ; CLEAR OVERLOAD FLAG
0647	DC5D	7F 0006	CLR OVLD1 ; CLR COND 1 FLAG
0648	DC60	7F 005A	CLR SHDN ; CLR SHUT DOWN FLAG
0649	DC63	39	NOLCC RTS

SEQ	LOC	OBJ	SOURCE
0651			* OVERLOAD EXCEEDS MAX. CAUSE RAMP DOWN
0652			* SET BOTH OVERLOAD FLAGS AND TURN ON
0653			* OVERLOAD LAMP
0654			*
0655	DC64	86 3C	OLCM LDA A #\$3C ; SET OL LAMP CTRL WD
0656	DC66	87 5003	STA A PIA+3 ; TURN ON OL LAMP
0657	DC69	97 03	STA A OVRLD ; SET FREQ CTRL FLAG
0658	DC6B	97 06	STA A OVLD1 ; SET COND1 FLAG
0659	DC6D	39	RTS
0660			* OUT OF CONTROL (V* > OUTCT LIMIT)
0661	DC6E	86 3C	OTCT LDA A #\$3C ; SET FLAG
0662	DC70	97 5A	STA A SHTDN ; STORE FLAG
0663	DC72	20 F0	BRA OLCM ; SKIP TO SHOW OVRLD

SEQ	LOC	OBJ	SOURCE
0665			*
0666			* * * * *
0667			* ADIN--A/D CONVERTER INPUT ROUTINE
0668			* * * * *
0669			*
0670			* THIS SUBROUTINE IS USED ONLY TO START UP
0671			* SUBROUTINES FUNCTIONS ARE :
0672			*
0673			* BRING IN B+ - BPRAW
0674			* BRING IN V* - VSRW
0675			* BRING IN SPEED CMD - FCRW
0676			*
0677			* THESE UNIT COMMANDS AND PARAMETERS ARE REQUIRED
0678			* BY "MCL" TO DECIDE WHAT TO DO IF THE MONITOR
0679			* IS NOT CONNECTED
0680			*
0681			* START CONVERT ON B+ - THEN GET V* MS & LS
0682			*
0683			* INITIAL CONDITIONS : START CONV = 0
0684			* B+ ADDRESS IS ENABLED
0685			*
0686	DCA0		ORG \$DCA0
0687			*
0688			* START A/D CONVERTER - UNIT IS STOPPED
0689			* MUST SET PTM2 TIMER #1 TO RUN GENERATING INTERVAL
0690			* MODE FOR THIS TIMER MUST NOT INTERFERE WITH STOPPED
0691			* CONDITION.
0692			*
0693			* PTM2 T1 - MODE
0694			* 16 BIT COUNTER, SINGLE SHOT, INTERRUPT DISABLED
0695			* OUTPUT ENABLED, ALLOWED TO OPERATE
0696			*
0697	DCA0	86 01	ADIN LDA A #\$01 ; SET CR2 CTRL WD
0698	DCA2	B7 3001	STA A PTM2+1 ; ENABLE WR TO CR1
0699	DCA5	86 A2	LDA A #\$A2 ; SET CR1 CTRL WD
0700	DCA7	B7 3000	STA A PTM2 ; WR CR1 WD
0701	DCA8	CE 0060	LDX #\$0060 ; SET INTERVAL TO 96US
0702	DCAD	FF 3002	STX PTM2+2 ; SET COUNT TO T1
0703			* CONTINUE WHILE A/D RUNS
0704	DCB0	86 77	LDA A #\$77 ; LD CMD TO GET V*
0705	DCB2	B7 5002	STA A PIA+2 ; ENB TO GET V*
0706	DCB5	86 5000	LDA A PIA ; LD V* MS
0707	DCB8	97 3A	STA A VSRW ; SAVE V* MS
0708	DCBA	86 75	LDA A #\$75 ; LD CMD TO GET V* LS
0709	DCBC	B7 5002	STA A PIA+2 ; ENB FOR V* LS
0710	DCBF	86 5000	LDA A PIA ; GET V* LS
0711	DCC2	97 3B	STA A VSRW+1 ; SAVE V* LS
0712	DCC4	86 76	LDA A #\$76 ; LD CMD FOR B+
0713	DCC6	B7 5002	STA A PIA+2 ; ENB B+
0714			* SET MUX ADDRESS FOR SPEED COMMAND
0715	DCC9	C6 72	LDA B #\$72 ; MUX ADDR CHANGE
0716	DCCB	86 5000	LDA A PIA ; LD B+
0717	DCCE	F7 5002	STA B PIA+2 ; ENB SPD CMD
0718	DCD1	97 4A	STA A BPRW ; SAVE B+

SEQ	LOC	OBJ	SOURCE
0719			* BRING IN SPEED COMMAND
0720	DCD3	86 60	LDA A #60 ; SET CMD
0721	DCD5	B7 5002	STA A PIA+2 ; START CONV ON SPD CMD
0722	DCD8	C6 66	LDA B #66 ; SET MUX ADDR CHANGE
0723	DCDA	B6 5000	LDA A PIA ; LD SPD CMD
0724	DCDD	F7 5002	STA B PIA+2 ; ENB B+ AT MUX
0725	DCE0	97 38	STA A FCRAW ; SAVE SPD CMD
0726			* ANALOG MUX ADDRESS HAS BEEN LEFT AT B+ ENABLED(PB2=1)
0727			* AND START CONVERT=0. REMEMBER THIS SINCE START
0728			* CONVERT IS INITIATED BEFORE ANY MORE ADDRESS CHANGES
0729			*
0730			* HAVE ALSO LEFT PR/RST IN NO CHANGE CONDITION
0731	DCE2	39	RTS

SEQ	LOC	OBJ	SOURCE
0733			*
0734			* THIS SUBROUTINE ESTABLISHES FCMD AND E* FOR USE BY
0735			* THE MAIN CONTROL LOOP. TO ACCOMPLISH THIS FUNCTION
0736			* - THE FOLLOWING PROCESSES MUST HAPPEN
0737			* FCMD - GENERATION
0738			* DETERMINE IF KEYBOARD OR REAL INPUT IS USED
0739			* DOING THE CORRECT PROCESS FOR THE SOURCE OF
0740			* THIS DATA
0741			*
0742			* E* - GENERATION
0743			* DETERMINE IF KEYBOARD OR REAL INPUT IS USED
0744			* DOING CORRECT PROCESS FOR DATA SOURCE
0745			*
0746			* BOTH THESE PROCESSES REQUIRE MPA TO KNOW IF THE
0747			* KEYBOARD IS CONNECTED.
0748			*
0749			* IT IS KNOWN THAT THE RAW DATA IS STORED AT :
0750			* BPRAW = B+ FROM A/D
0751			* FCRW = SPEED COMMAND FROM A/D
0752			* VSRW = V* FROM MPB
0753			*
0754	DCE3	96 51	CSPDC LDA A MONFG ; IS MONITOR CONNECTED
0755	DCE5	27 46	BEQ NPROC ; SKIP IF NO KBD AVAILABLE
0756	DCE7	20 17	BRA RACIA ; SKIP FOR KBD INPUT
0757			*
0758			* SECTION READS KEYBOARD TO SEE IF THERE'S ANY NEW
0759			* INSTRUCTIONS FOR THE MANUAL OPERATION OF THE UNIT
0760			*
0761			* AVAILABLE MANUAL CONTROLS :
0762			*
0763			* "+" = ADVANCE FREQ CMD (FCMD) BY 1 HZ.
0764			* "-" = DECREMENT FREQ CMD (FCMD) BY 1 HZ.
0765			* "A" = INCREASE OUTPUT VOLTAGE(SF) BY (XXX) VOLTS
0766			* "D" = DECREASE OUTPUT VOLTAGE(SF) BY (XXX) VOLTS
0767			* "V" = USE V* FROM MPB TO OPERATE (CLOSE LOOP)
0768			* "R" = RAMP DOWN TO 0 SPEED AND REVERSE
0769			* MOTOR DIRECTION FLAG
0770			*
0771			* "R" MODE CANCELED BY HITTING "+", "-", "A", OR "D"
0772			* "V" MODE CANCELED BY HITTING "A" OR "D"
0773			*
0774		DD00	ORG \$DD00
0775			*
0776	DD00	B6 6000	RACIA LDA A ACIAS ; LD ACIA STATUS
0777	DD03	47	ASR A ; CARRY=RDRF
0778	DD04	24 1D	BCC RTRA ; BRANCH IF NO CHANGE
0779	DD06	B6 6001	LDA A ACIAD ; LD CHAR
0780	DD09	84 7F	AND A #\$7F ; STRIP PARITY
0781	DD0B	81 52	CMP A #'R ; TEST FOR REVERSE
0782	DD0D	27 54	BEQ NRMO ; BRANCH IF REVERSE
0783	DD0F	81 2B	CMP A #'+
0784	DD11	27 5A	BEQ IFCD ; BRANCH TO INCR FREQ
0785	DD13	81 2D	CMP A #'-
0786	DD15	27 5C	BEQ DFCD ; BRANCH TO DECR FREQ

SEQ	LOC	OBJ	SOURCE
0787	DD17	81 56	CMP A #1V ; TEST FOR CLOSED LOOP
0788	DD19	27 62	BEQ UVSTAR ; SKIP TO USE MPB-V*
0789	DD1B	81 41	CMP A #1A ; TEST FOR ADV V*
0790	DD1D	27 63	BEQ AVSTAR ; SKIP TO ADV V*
0791	DD1F	81 44	CMP A #1D ; TEST FOR DECR V*
0792	DD21	27 71	BEQ DVSTAR ; SKIP TO DECR V*
0793			* CHARACTER NOT VALID OR NO CHANGE FROM KEYBOARD
0794			* MUST ALWAYS DO E* CALCULATION
0795			*
0796	DD23	96 E6	RTRA LDA A NORFG ; LD NORM OP FLAG
0797	DD25	27 06	BEQ NPROC ; SKIP FOR NORM OP
0798	DD27	96 E7	LDA A VSFLG ; LD V* FLAG
0799	DD29	27 16	BEQ VSPROC ; SKIP FOR CLOSED LOOP OP
0800	DD2B	20 1D	BRA ESC ; COMPUTE E*
0801			*
0802			* NORMAL PROCESSING - KEYBOARD IS NOT CONNECTED
0803			*
0804			* 1ST OPERATION - DETERMINE FREQUENCY COMMAND
0805			* THIS BRANCH MAY ALSO BE REACHED IF NORMAL MODE
0806			* "N" IS CHOSEN FROM KEYBOARD
0807			*
0808			*
0809			* SPEED INPUT IS DIGITAL AT PIA
0810			* FORMAT IS B7 B6 B5 B4 B3 B2 B1 B0
0811			* REV UP DN MON X X X X
0812			*
0813			* TOGGLE SWITCH INPUT FOR RAMPING UP SPEED, RAMPING DOWN
0814			* SPEED, OR REVERSING MOTOR DIRECTION
0815			*
0816	DD2D	96 46	NPROC LDA A FCMD ; GET COMMANDED FREQ
0817	DD2F	91 4C	CMP A FOUT ; COMPARE WITH ACTUAL FREQ
0818	DD31	26 0E	BNE VSPROC ; BRANCH IF NOT AT COMMANDED SPD
0819	DD33	D6 38	LDA B FCRAW ; LD FREQ CMD-TOGGLE SWITCHES
0820	DD35	C5 80	BIT B #80 ; TEST FOR REVERSE
0821	DD37	27 2A	BEQ NRMO ; IF BIT 7=0 REVERSE MOTOR
0822	DD39	C5 40	BIT B #40 ; TEST FOR RAMP-UP SPEED
0823	DD3B	27 2A	BEQ IFCD1 ; IF BIT 6=0 INCREMENT SPD CMD
0824	DD3D	C5 20	BIT B #20 ; TEST FOR RAMP-DOWN SPEED
0825	DD3F	27 32	BEQ DFCD ; IF BIT 5=0 RAMP-DOWN SPEED

SEQ	LOC	OBJ	SOURCE
0827			*
0828			* FREQ CMD DONE - NEXT IS CALCULATE E*
0829			*
0830			* $E* = 1 + V*/B+NOM = 1 + V*/128$
0831			* V* FORMAT IS TWO BYTES AS :
0832			* B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0 0 0 0 0
0833			* 1 . . 5 . 25 . 125
0834			*
0835			* WILL BE USED WITH SF CALCULATION
0836			* SF SCALING :
0837			* B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0838			* X X X X S 4 2 1 . 5 X X X X X X X 2E-5
0839			*
0840			* ORIGINALLY V* HAD A SIGIFICANCE(MEASURED BY OPERATION)
0841			* OF THE LSB = 0.125 VOLTS
0842			*
0843			* WILL SCALE V* SO THAT A "1" VOLT CHANGE AFFECTS S. F.
0844			* THE SAME AS A "1" VOLT B+ CHANGE (MEANS LS 2 OF V*
0845			* ARE OF NO IMPORTANCE SINCE THEY FALL OFF
0846			*
0847			* V* SCALING *-----A-----* *-----B-----*
0848			* VSRW S S S 32 16 8 4 2 1 . 5 . 25 X 0 0 0 0
0849			* E* S S S S 8 4 2 1 . 5 . 25 X X X X X X
0850			*
0851			* V*/128 S32 16 8 4 2 1 . 5 S S S S S S S
0852			*
0853	DD41	DE 3A	VSPROC LDX VSRW ;LD V*
0854	DD43	DF 48	STX VSTAR ;STORE V* FOR CALC
0855	DD45	CE 0200	LDX #0200 ;V/F MOD 5-3-83
0856	DD48	20 16	BRA VOLTHZ
0857	DD4A	96 48	ESC LDA A VSTAR ;RELOAD MS V*
0858	DD4C	D6 49	LDA B VSTAR+1 ;RELOAD LS V*
0859	DD4E	CE 0003	LDX #0003 ;LD SCALING CNT
0860	DD51	47	SVS ASR A ;SHIFT MS
0861	DD52	56	ROR B ;SHIFT LS
0862	DD53	09	DEX ;DECR SHIFT CNT
0863	DD54	26 FB	BNE SVS ;SKIP IF MORE SCALING
0864			* V* IS ORIGINAL VALUE INPUT / BY 8
0865			* NOW DO 1 + V*/8
0866	DD56	CB 00	FESTAR ADD B #00 ;ADD LS + 0.0
0867	DD58	89 01	ADC A #01 ;ADD MS + 1
0868	DD5A	D7 41	STA B ESRT+1 ;SAVE LS E*
0869	DD5C	97 40	STA A ESRT ;SAVE MS E*
0870	DD5E	DE 40	LDX ESRT ;RELOAD NEW E*
0871	DD60	DF 3C	VOLTHZ STX ESTAR ;STORE E* FOR INTRP PROG
0872	DD62	39	RTS

SEQ	LOC	OBJ	SOURCE
0874			*
0875			* "R" - REVERSE DIRECTION KEY
0876			*
0877	DD63	73 0009	NRMO COM DIRCMD ; COMPLEMENT THE DIRECTION FLAG
0878	DD66	39	RTS ; RETURN
0879			*
0880			* "+" ADVANCE FREQ KEY
0881			*
0882	DD67	96 46	IFCD1 LDA A FCMD ; TEST FOR MAX FREQ
0883	DD69	81 20	CMP A ##20 ; MAX = 32 HZ FOR NSRDC
0884	DD6B	27 05	BEQ RETURN ; IF AT MAX, RETURN
0885	DD6D	97 E6	IFCD STA A NORFG ; STORE FLAG
0886	DD6F	7C 0046	INC FCMD ; INCR FCMD
0887	DD72	39	RETURN RTS
0888			*
0889			* "-" DECREMENT FREQUENCY KEY
0890			*
0891	DD73	97 E6	DFCD STA A NORFG ; STORE FLAG
0892	DD75	96 46	LDA A FCMD ; LD FREQ TO BE OUTPUT
0893	DD77	27 03	BEQ EX1 ; BRANCH IF 0 FREQ
0894	DD79	7A 0046	DEC FCMD ; DECR FREQ CMD
0895	DD7C	39	EX1 RTS ; RETURN
0896			*
0897			* "V" USE REAL V* -CLOSE LOOP ON MPB ERROR
0898			*
0899	DD7D	7F 00E7	UVSTAR CLR VSFLG ; CLR V* CLOSED LOOP FLG
0900	DD80	20 BF	BRA VSPROC ; DO NORM V* PROC
0901			*
0902			* "A" ADVANCE V* KEY
0903			*
0904	DD82	97 E7	AVSTAR STA A VSFLG ; SET KBD CTRL OF V*
0905	DD84	97 E6	STA A NORFG ; RESET NORM FLAG
0906	DD86	96 48	LDA A VSTAR ; V* MS
0907	DD88	D6 49	LDA B VSTAR+1 ; V* LS
0908	DD8A	CB 40	ADD B ##40 ; INCR LS
0909	DD8C	89 00	ADC A ##00 ; INCR MS
0910	DD8E	97 48	STA A VSTAR ; SAVE NEW V* MS
0911	DD90	D7 49	STA B VSTAR+1 ; SAVE NEW V* LS
0912	DD92	20 B6	BRA ESC ; SKIP TO USE V*
0913			*
0914			* "D" DECREMENT V* KEY
0915			*
0916	DD94	97 E7	DVSTAR STA A VSFLG ; SET KBD CTRL OF V*
0917	DD96	97 E6	STA A NORFG ; RESET NORM FLAG
0918	DD98	96 48	LDA A VSTAR ; V* MS
0919	DD9A	D6 49	LDA B VSTAR+1 ; V* LS
0920	DD9C	C0 40	SUB B ##40 ; DECR LS
0921	DD9E	82 00	SBC A ##00 ; DECR MS
0922	DDA0	97 48	STA A VSTAR ; SAVE NEW V* MS
0923	DDA2	D7 49	STA B VSTAR+1 ; SAVE NEW V* LS
0924	DDA4	20 A4	BRA ESC ; SKIP TO USE V*

SEQ	LOC	OBJ	SOURCE
0926			*
0927			* STOP ROUTINE
0928			*
0929	DDA6	96 09	STRUT LDA A DIRCMD ; LOAD NEW DIRECTION
0930	DDA8	97 10	STA A DIROUT ; STORE NEW DIRECTION
0931	DDAA	86 01	LDA A #\$01
0932	DDAC	97 01	STA A MSTOP ; SET UNIT STOPPED FLAG
0933	DDAE	7E D8F4	JMP MCLCN ; MOTOR CONNECT MCL
0934			END

0934 LINES ASSEMBLED, LOC = DDB1, 0000 ERRORS DETECTED.

SYMBOL	VALUE	ATTR	LOCN	LINK
MSTOP	0001	84	2AA2	
MSTART	0002	84	2AA8	
OVRD	0003	84	2AB4	
INHIB	0004	84	2ABD	
PIACMD	0044	84	2AC6	
FCMD	0046	84	2ACF	
FOUT	004C	84	2AD8	
VSRAW	003A	84	2AE1	
BPRAW	004A	84	2AE8	
FCRAW	0038	84	2AF3	
DOMID	0020	84	2AFC	
DPF	0022	84	2B05	
J	0024	84	2B0E	
F60DEG	00C0	84	2B17	
FTINCR	00C8	84	2B20	
MONFG	0051	84	2B29	
NPPCT	00C5	84	2B32	
NORFG	00E6	84	2B38	
VSFLG	00E7	84	2B44	
PTIS	E000	84	2B4D	
FTPNT	00C1	84	2B56	
OVL1	0006	84	2B5F	
INDEX	00E8	84	2B68	
FTPTR	00C3	84	2B71	
ACIAC	6000	84	2B7A	
DIRCMD	0009	84	2B83	
DIROUT	0010	84	2B8C	
BPFLG	0050	84	2B95	
VSTAR	0048	84	2B9E	
ESOC	00EA	84	2BA7	
K	0005	84	2BB0	
AVE	00F0	84	2BB9	
ESMM	0001	84	2BC2	
ESML	0000	84	2BCB	
ES1MM	0000	84	2BD4	
ES1ML	0080	84	2BD0	
ESTAR	003C	84	2BE6	
YN	00F0	84	2BEF	
FTPNT	00CA	84	2BF8	
ESRT	0040	84	2C01	
OVCMP	00F0	84	2C0A	
VPADDR	00D8	84	2C13	
VSREF	00DB	84	2C1C	
OUTCTL	00EF	84	2C25	
OUTCTM	000D	84	2C2E	
SHTDN	005A	84	2C37	
PIA	5000	84	2C40	
PTM1	4000	84	2C49	
PTM2	3000	84	2C52	
WLATCH	2007	84	2C5B	
ACIAS	6000	84	2C64	
ACIAD	6001	84	2C6D	
INIT	D800	84	2C76	
INIT1	D80F	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
TRESET	D860	84	2C88	
MCL	D880	84	2C91	
MCL02	D89B	84	2C9A	
ADIN	DCA0	84	2CA3	
CSPDC	DCE3	84	2CAC	
MCL01	D893	84	2CB5	
PROTCT	DC00	84	2CBE	
MCL03	D8B7	84	2CC7	
OLRMPD	D8FF	84	2CD0	
TFOUT	D8CF	84	2CD9	
MCL04	D8C8	84	2CE2	
RAMPDN	D880	84	2CEB	
MCL3A	D8BA	84	2CF4	
RAMPUP	D800	84	2CFD	
STUNT	D8F6	84	2D06	
CHPTR	D8D3	84	2D0F	
AINDX	D8D8	84	2D18	
TSTART	DA20	84	2D21	
MCLCN	D8F4	84	2D2A	
STROUT	DDA6	84	2D33	
OLRMP1	D908	84	2D3C	
SLWRD	DB91	84	2D45	
SMOUT	DB11	84	2D4E	
SFOUT	DB0C	84	2D57	
ERU	DB10	84	2D60	
ERD	DB8C	84	2D69	
ESRD	DBAA	84	2D72	
PVOLT	DC0F	84	2D7B	
ESPT	DC13	84	2D84	
OTCT	DC6E	84	2D8D	
NOLC	DC51	84	2D96	
PRTCT1	DC32	84	2D9F	
OLC1	DC46	84	2DA8	
OLCM	DC64	84	2DB1	
OLC1A	DC4D	84	2DBA	
NOLCC	DC63	84	2DC3	
NPROC	DD2D	84	2DCC	
RACIA	DD00	84	2DD5	
RTRA	DD23	84	2DDE	
NRMO	DD63	84	2DE7	
IFCD	DD6D	84	2DF0	
DFCD	DD73	84	2DF9	
UVSTAR	DD7D	84	2E02	
AVSTAR	DD82	84	2E0B	
DVSTAR	DD94	84	2E14	
VSPROC	DD41	84	2E1D	
ESC	DD4A	84	2E26	
IFCD1	DD67	84	2E2F	
VOLTHZ	DD60	84	2E38	
SVS	DD51	84	2E41	
FESTAR	DD56	84	2E4A	
RETURN	DD72	84	2E53	
EX1	DD7C	04	2E5C	

END ASM V1.1

SEQ LOC OBJ SOURCE

```

0001      * PIM INTERRUPT SERVICE ROUTINE
0002      NAM      PTMS6
0003      ****
0004      * PIM-PULSE TIMER INTERRUPT ROUTINE *
0005      ****
0006      * THIS MODULE CONTROLS THE OPERATION OF 2 PULSE TIMER MODULES
0007      * IN ORDER TO SYNTHESIZE THE PULSE WIDTH MODULATED (PWM)
0008      * WAVEFORMS REQUIRED TO CONTROL A UNIT. PIM#1 OUTPUTS ARE
0009      * CONNECTED TO THE THREE PHASE INPUTS OF THE CONTROLLER.
0010      * PIM#2 ESTABLISHES THE PULSE PERIOD INTERVAL AND GENERATES
0011      * INTERRUPTS WHICH USE THIS MODULE.
0012      *
0013      *
0014      * DEFINITION OF MEMORY LOCATIONS USED BY THIS ROUTINE
0015      *
0016      * (LOCAL)
0017      0000 F60DG EQU $0000 ; START OF 60 DEG SEGMENT FLAG
0018      0001 F1PFW EQU $0001 ; NEW FREQ TABLE POINTER
0019      0003 F1PIR EQU $0003 ; POINTER FOR VALUES IN USE
0020      0006 NPFMD EQU $0006 ; COUNT FOR MID-POINT OF P.P.
0021      0002 BSTHR EQU $0002 ; B* FOR SF COMPUTATION
0022      0000 SF EQU $0000 ; SCALE FACTOR
0023      0072 PWDUM EQU $0072 ; DOMINANT'S PULSE WIDTH
0024      0074 PWCUM EQU $0074 ; COMPLEMENT'S PULSE WIDTH
0025      0070 TPF EQU $0070 ; WIDTH OF PULSE PERIOD
0026      0078 T1 EQU $0078 ; 1ST SWITCHING TIME
0027      007A T2 EQU $007A ; 2ND SWITCHING TIME
0028      007C T3 EQU $007C ; 3RD SWITCHING TIME
0029      0008 VPADD EQU $0008 ; V PEAK TABLE ADDRESS
0030      *
0031      *
0032      0030 SFMAX EQU $0030 ; MAX SF FOR PP
0033      0032 SFMIN EQU $0032 ; MIN SF FOR PP
0034      *
0035      *
0036      * (GLOBAL)
0037      0044 F1RCMD EQU $0044 ; COMMANDED STATE OF PIA-B
0038      0020 DOMID EQU $0020 ; DOMINANT'S ID (0=A,1=B,2=C)
0039      0022 DPF EQU $0022 ; DOM'S POLARITY (0=+ ,FF=-)
0040      0024 J EQU $0024 ; FLAG (+=1ST,ODD/ -=1ST EVEN)
0041      0008 F1INCR EQU $0008 ; FREQ TAB (+4=1ST HALF/-4=2ND)
0042      0051 MUNFG EQU $0051 ; MUN CONN FLAG ( 0= NOT CONN)
0043      003A VSRAM EQU $003A ; V* FROM PIA (RAW DATA)
0044      00E7 VSLG EQU $00E7 ; V* IN FLG(0=IN MUPA,1=USE KBD)
0045      004A BFRAM EQU $004A ; B+ (RAW INPUT)
0046      0038 FCRAM EQU $0038 ; FREQ CMD FROM A/D (RAW DATA)
0047      003C ESTHR EQU $003C ; E* = V*/B+NOM
0048      0005 NPPCT EQU $0005 ; NO. OF P.P.'S LEFT IN SEG
0049      004C FOUT EQU $004C ; OUTPUT FREQUENCY
0050      0046 FCMD EQU $0046
0051      0009 DIRCMD EQU $0009
0052      0010 DIROUT EQU $0010

```


SEQ	LOC	OBJ	SOURCE
0054			*
0055			*
0056			* (I/O ADDRESSES)
0057	2000	MP1ER	EQW \$2000 ; MULTPLR 12 BITS (HANDLE AS 1
0058	2002	MCAND	EQW \$2002 ; MULTPLND 12 BITS (" " "
0059	2005	PRGD	EQW \$2005 ; PRODUCT (MS 8 + 2 MORE BYTES)
0060	5000	PIH	EQW \$5000 ; PROGRAMMABLE INTERFACE ADAPT
0061	4000	PTM1	EQW \$4000 ; PULSE TIMER MODULE #1
0062	3000	PTM2	EQW \$3000 ; PULSE TIMER MODULE #2
0063	2007	WLATCH	EQW \$2007 ; W LATCH ADDRESS (FOR MUPB)
0064			*
0065			*
0066			*
0067			*
0068	0001		LSYMB 1
0069	E000		URG \$E000
0070			* INTERRUPT SERVICE
0071			*
0072			* IMMEDIATELY CLEAR INTERRUPT FLAG IN TIMER
0073			* THIS DONE VIA FOLLOWING ADDRESSING IF TIMER #2
0074	E000	B6	3001 PTIS LDH H PTM2+1 ; READ TIMER STATUS REG
0075	E003	B6	3002 LDH H PTM2+2 ; READ TIMER #1 COUNT
0076			*
0077			* CHECK IF THIS IS 60 DEG SEGMENT BOUNDARY
0078			*
0079	E006	7D	0000 IST F60DG ; TEST FOR BEGINNING OF 60 DE
0080	E009	27	4A BEW NOT60 ; BRANCH IF NOT BEGINNING
0081			* 60 DEG INTERVAL - SET UP COUNTERS, FLIP-FLOPS, ETC
0082			*
0083			* FIRST THING IS SET MUX ADDRESS AND START CONVERT TO
0084			* SPEED CMD "0"
0085			*
0086	E00B	86	66 LDH H #\$66 ; ENAB B+ & SET SC=0
0087	E00D	B7	0002 STH H PIH+2 ; START CONV TO CLR SEQ
0088			* THE ABOVE H/D CONVERSION IS USED ONLY TO GET START CONVERT
0089			* BACK IN SEQUENCE. CONVERTER RESULT IS NOT USED !!
0090			*
0091	E010	DE	01 LDX FTPNW ; GET NEW FREQ TABLE POINTER
0092	E012	DF	03 STX FTPTR ; SET POINTER TO USE NEW VALU
0093	E014	A6	00 LDH A 00.X ; LOAD NO. OF P. P. 'S IN SEG
0094	E016	97	05 STA H NPPCT ; ESTABLISH BEGIN. PP COUNT
0095			*
0096			* CALCULATE MID-POINT OF SEGMENT
0097			*
0098	E018	4D	INC H ; NPP+1
0099	E019	47	ASR H ; (NPP+1)/2=MID-POINT COUNT
0100	E01A	97	06 STA H NPPMD ; STORE IN MID-POINT LOCAL
0101			*
0102			* FULL PULSE PERIOD PARAMETERS FROM TABLE
0103			*
0104	E01C	EE	01 LDX 01.X ; GET PP LENGTH
0105	E01E	FF	0002 STX PTM2+2 ; ESTAB TIME BETWEEN INTERRU
0106	E021	DF	70 STX TFP ; STORE NEW PP LENGTH
0107	E023	DE	03 LDX FTPTR ; RELOAD TABLE TOP ADDRESS

SEQ	LOC	OBJ	SOURCE
0108	E025	EE 05	LDX 05,X ; LOAD MAX SF
0109	E027	DF 30	STX SFMAX ; STORE HS MAX SCALE FACTOR
0110	E029	DE 03	LDX FTPTX ; RELOAD TABLE TOP
0111	E02B	EE 07	LDX 07,X ; LOAD MIN SF
0112	E02D	DF 32	STX SFMIN ; STORE HS MIN SCALE FACTOR
0113			*
0114			* UPDATE DOMINANT WAVEFORM ID
0115			* DOMINANT ID = DOMID = (A=0, B=1, C=2)
0116			*
0117	E02F	96 20	LDA H DOMID ; LOAD DOM ID FLAG
0118			*
0119			* ID ALWAYS CHANGES B, A, C, B, A, C, ETC
0120			* AT 60 DEG BOUNDARY DOMID=B !
0121			*
0122	E031	4H	DEC H ; DECREMENT ID
0123	E032	2H 02	BPL AH ; BRANCH IF AOR B
0124	E034	86 02	LDA H #02 ; SET FLAG VALUE FOR "C"
0125	E036	97 20	STA H DOMID ; STORE NEW ID

SEQ	LOC	OBJ	SOURCE
0127			* CHANGE DOMINANT POLARITY FLAG (DPF=0/+ , FF/-)
0128			* ALSO SET VALUES OF A,B,C FLIPFLOPS. AT 60 DEG BOUNDARY DPF=0
0129			* LHS1 START CONVERT ADDR. -PB2=0 BEFORE HERE
0130			*
0131	E038	86 A6	LDA A #A6 ;LD CMD TO SET FLIP FLOP
0132	E03A	73 0022	COM DPF ;CHANGE DOMINANT POLARITY
0133	E03D	2B 02	BMI A2 ;BRANCH IF NEGATIVE
0134	E03F	86 06	LDA A #06 ;LD CMD TO RESET FLIP FLOP
0135	E041	97 44 A2	STA A PIACMD ;STORE COMMAND FOR LATER USE
0136			*
0137			* INITIALIZE INDICES FOR PROCESSING
0138			*
0139	E043	7F 0024	CLR J ;CLEAR 1ST/2ND HALF FLAG
0140	E046	CE 0004	LDX #+4 ;SET 1ST HALF VALUE
0141	E049	DF 08	STX FTINCR ;ESTABLISH LS OF ADDRESS
0142			*
0143			* SEND OUTPUT FREQUENCY TO MUPB
0144			*
0145	E048	96 4C	LDA A FOUT ;LOAD PRESENT OUTPUT FREQUENCY
0146	E04D	B7 2007	STA A WLATCH ;STORE FOUT TO LATCH
0147	E050	7F 00C0	CLR F60DG ;CLEAR 60 DEG BOUNDARY FLAG
0148	E053	20 28	BRA BPIN ;DONE 60 DEG SEG, DO B+ INPUT
0149			*
0150			* CODE IF NOT A 60 DEG BOUNDARY (INSIDE A 60 DEG SEGMENT)
0151			* USE EXTRA LINE TO INPUT V* AND FCMD
0152			* SPD CMD AND V* ARE INPUT AND STORED FOR USE BY MAIN PROGRAM
0153			* A/D START CONVERT IS HIGH TO BEGIN WITH
0154			* ADDRESS FOR SPD CMD ALREADY SET
0155			*
0156	E055	86 62 NO160	LDA A #62 ;SET IN RESET/PRESET - NO CHANGE
0157	E057	06 63	LDA B #63 ;SET FOR V* INPUT
0158	E059	B7 5002	STA A PIA+2 ;START A/D CONVERSION
0159			*
0160			* CONVERSION FOR SPEED COMMAND IS STARTED
0161			* GET V* WHILE CONVERSION IS WORKING
0162			* START CONVERSION - PIA-PB4 SET=1
0163			*
0164	E05C	F7 5002	STA B PIA+2 ;OUTPUT COMMAND TO PIA (ENB V*
0165	E05F	F6 5000	LDA B PIA ;BRING IN V* MSB
0166	E062	86 61	LDA A #61 ;SELECT V* LS, OTHERS NO CHANGE
0167	E064	B7 5002	STA A PIA+2 ;OUTPUT COMMAND TO PIA
0168	E067	B6 5000	LDA A PIA ;LOAD V* LS
0169	E06A	97 3B	STA A VSRW+1 ;SAVE V* LS
0170	E06C	D7 3A	STA B VSRW ;SAVE V* MS
0171			* ENABLE ANALOG MUX FOR B+ INPUT
0172	E06E	86 60	LDA A #60 ;SET ENAB WD FOR B+
0173	E070	B7 5002	STA A PIA+2 ;ENAB B+
0174			*
0175			* FINISH A/D CONVERSION FOR SPEED COMMAND
0176			*
0177	E073	01	NOP ;FILL KEEP PROG LENGTH
0178	E074	01	NOP ;FILL KEEP PROG LENGTH
0179	E075	01	NOP ;FILL KEEP PROG LENGTH
0180	E076	01	NOP ;FILL KEEP PROG LENGTH

AMI 6800
ASM V1.1

PTM56

PAGE 0005

SEQ LOC OBJ SOURCE

0181 E077 01

NOF

; FILL KEEP PROG LENGTH

0182 E078 F6 5000

LDA B PIA

; LOAD SPEED COMMAND

0183 E07B D7 38

STH B PCRAW

; STORE RAW FREQUENCY COMMAND

SEQ	LOC	OBJ	SOURCE
0185			*
0186			* V* IS STORED - EITHER BY ABOVE OR VIA KEYBOARD CONTROL
0187			* CONTINUE SEGMENT PROCESSING
0188			*
0189			* BRING IN V PEAK DURING EVERY SEGMENT, CALCULATE NEW
0190			* SCALE FACTOR AND APPLY TO ADJUST PULSE WIDTHS.
0191			*
0192			* V PEAK MUST BE READ IN AND SAVED IN BPAW. ALSO SET AND
0193			* RESET FF'S DEPENDING ON POLE VOLTAGE STATE FOR 60 DEG
0194			* SEGMENT
0195			* INPUT V PEAK FIRST
0196			* START CONVERSION PIA-PB4 SET=0
0197			*
0198	E07D	86 76	BPIN LDA A #76 ;SAVE SET/RST,ENB A/D; SET S.C.
0199	E07F	B7 5002	STA A PIA+2 ;ENABLE FLIPFLOPS, START CONVE
0200	E082	C6 06	LDA B #06 ;LD A/D TIMING CNT(36 US TOTAL)
0201	E084	5A	CBPC DEC B ;DECREMENT A/D COUNT
0202	E085	26 FD	BNE CBPC ;BRANCH WAIT FOR CONVERTER
0203	E087	B6 5000	LDA A PIA ;LOAD B+
0204	E08A	97 4A	STA A BPAW ;STORE B+(VOLT-SEC) (RAW INPUT
0205	E08C	C6 72	LDA B #72 ;SET SPD CMD ENAB WD
0206	E08E	F7 5002	STA B PIA+2 ;ENAB SPD CMD
0207			*
0208			* V PEAK HAS NOW STORED AT BPAW
0209			* , PIA IS SET UP SO THAT FLIPFLOPS
0210			* WILL BE SET, RESET OR NOT AFFECTED BY PULSE PERIOD INTERRUPT
0211			*
0212			* NEXT STEP COMPUTES SCALE FACTOR BASED ON NEW V PEAK
0213			*
0214			* MUST FIND VPEAK NOM/V PEAK FROM LOOK UP TABLE
0215			* FIRST GENERATE TABLE ADDRESS FROM V PEAK
0216	E091	86 E3	LDA A #E3 ;LD MS OF TABLE ADDR
0217	E093	D6 4A	LDA B BPAW ;LD V PK FROM MEM
0218	E095	58	ASL B ;ADJ LS FOR 2 BYTES/V PK
0219	E096	24 02	BCC SVPA00 ;SKIP FOR <128 SAVE ADDR
0220	E098	86 E6	LDA A #E6 ;SET MS AT 128 VOLT POS
0221			* ADDRESS FOUND, SAVE IT AND LOOK UP VPEAK NOM/V PEAK
0222	E09A	97 D8	SVPA00 STA A VPADD ;SAVE MS
0223	E09C	D7 D9	STA B VPADD+1 ;SAVE LS
0224	E09E	DE D8	LDX VPADD ;RELOAD VPEAK ADDR
0225			*
0226			* SCALE FACTOR COMPUTATION
0227			*
0228			* SF = (1+E*/128) * VPEAK NOM/V PEAK
0229			* FROM MCL FOUND BY THIS SUBROUTINE
0230			*
0231	E0A0	EE 00	SFCHL LDX 0,X ;LD X V PEAK NOM/V PEAK
0232	E0A2	FF 2000	STX MPIER ;STORE AS MULTIPLIER
0233	E0A5	DE 3C	LDX ESTAR ;LD (1+E*/128)
0234	E0A7	FF 2002	STX MCAND ;STORE AS MULTIPLICAND
0235	E0AA	B6 2005	LDA A PROD ;LD MS OF SF
0236	E0AD	F6 2006	LDA B PROD+1 ;LD LS OF SF
0237	E0B0	47	ASR A ;ADJ FOR MULTIP TRUNC
0238	E0B1	56	ROR B ;ADJ LS FOR TRUNC

SEQ	LOC	OBJ	SOURCE
0239			*
0240			* SCALE FACTOR NOW EXISTS IN H & B REGISTERS
0241			* SCALING IS :
0242			* B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0243			* X X X X S 4 2 1.5 D D D D D D 2E-8
0244			*
0245	E082	D7 D1	STH B SF+1 ; SAVE SF LS
0246	E084	97 D0	STH A SF ; SAVE SF MS
0247			* TEST FOR NEGATIVE FIRST
0248	E086	81 00	CMP A #000 ; CMP TO ZERO
0249	E088	26 12	BMI SFMNL ; SKIP TO LD MIN SF

SEQ	LOC	OBJ	SOURCE
0251			*
0252			*
0253			* TEST MS OF SCALE FACTOR
0254			* SCALE FACTOR NOMINAL IS 0.5 (MIN E* & MIN B+ERR/B+NOM)
0255			* 2.996 (MAX E* & MAX B+ERR/B+NOM)
0256			*
0257	E0BA	27 00	BEQ SFMSZ ; BRANCH IF MS=0
0258			*
0259			* SF>=1.0000
0260			*
0261	E0BC	D0 31	SUB B SFMAX+1 ; SF-SFMAX LS
0262	E0BE	92 30	SEC A SFMAX ; SF-SFMAX MS
0263	E0C0	28 0E	BHI SFOK ; BRANCH IF <3.00
0264	E0C2	DE 30	LDX SFMAX ; LOAD MAX SF
0265	E0C4	DF D0	STX SF ; SET SF = SFMAX IN MEM
0266	E0C6	20 08	BRA SFOK ; BRANCH TO SFOK
0267			*
0268			* SF<1.000
0269			*
0270	E0C8	D1 33	SFMSZ CMP B SFMIN+1 ; SF-SFMIN LS
0271	E0CA	22 04	BHI SFOK ; BRANCH IF >0.75
0272	E0CC	DE 32	SFMINL LDX SFMIN ; LOAD MIN SF
0273	E0CE	DF D0	STX SF ; SET SF = SFMIN
0274			*
0275			* NEXT STEP IS TO COMPUTE PULSE WIDTHS BASED ON WHERE YOU ARE
0276			* COMPUTE DOMINANT FIRST
0277			*
0278	E0D0	DE C3	SFOK LDX FTPTR ; LOAD WORK FREQ TABLE PT
0279	E0D2	EE 09	LDX 09,X ; GET PWDOM BEFORE SCALING
0280	E0D4	FF 2000	STX MPIER ; STORE PWDOM AS MULTIPLIER
0281	E0D7	DE D0	LDX SF ; LOAD SCALE FACTOR
0282	E0D9	FF 2002	STX MCAND ; STORE SF AS MULTIPLICAND
0283			*
0284			* PWDOM = SF * TC1 (PWDOM BEFORE SCALING)
0285			*
0286	E0DC	B6 2005	LDA A PROD ; INPUT SIGN + PR1 TO PR7
0287	E0DF	F6 2006	LDA B PROD+1 ; INPUT PR8 TO PR15
0288	E0E2	47	ASR A ; POSITION ANSWER MS
0289	E0E3	56	ROR B ; POSITION LS
0290			*
0291			* PWDOM NOW HAS RESOLUTION IN MICROSECONDS AS
0292			*
0293			* B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0294			* S S 8K 4K 2K 1K 512 256 128 64 32 16 8 4 2 1*
0295			*
0296	E0E4	97 72	STA A PWDOM ; STORE MS
0297	E0E6	D7 73	STA B PWDOM+1 ; STORE LS

```

-- SEQ LOC OBJ SOURCE
--
*
*0299 *
*0300 * NEXT STEP IS TO COMPUTE THE COMPLEMENT PULSE WIDTH
*0301 *
*0302 E0E8 DE 03 LDW FTPTR ; RELOAD THE TABLE POINTER
*0303 E0EA EE 0B LDW 11,X ; GET TC - COMPLEMENT
*0304 E0EC FF 2000 STX MPIR ; STORE AS MULTIPLIER
*0305 *
*0306 * DON'T HAVE TO RELOAD SCALE FACTOR SINCE IT IS LATCHED AS
*0307 * MULTIPLICAND
*0308 *
*0309 E0EF B6 2005 LDA A PROD ; LOAD MS 8
*0310 E0F2 F6 2006 LDA B PROD+1 ; LOAD LS 8
*0311 E0F5 47 ASR A ; POSITION MS
*0312 E0F6 56 ROR B ; POSITION LS
*0313 E0F7 97 74 STA A PWCOM ; STORE MS
*0314 E0F9 D7 75 STA B PWCOM+1 ; STORE LS
*0315 *
*0316 * RESOLUTION & SCALING OF PWCOM IS IDENTICAL TO PWDM

```


SEQ	LOC	OBJ	SOURCE
0318			*
0319			* CALCULATE THE THREE SWITCHING TIMES FOR THIS PULSE PERIOD
0320			* TEST THE 'J' INDEX TO DETERMINE WHICH OF TWO ALGORITHMS TO
0321			* USE DIFFERENCE IN ALGORITHM IS RESULT OF POSITION IN SEGME
0322			* AND WHETHER PP NO. IS ODD OR EVEN
0323			*
0324			* T1= 1ST TO TIME OUT
0325			* T2= 2ND TO TIME OUT
0326			* T3= 3RD TO TIME OUT
0327			*
0328			* IN 1ST HALF OF SEGMENT T3= ODD P. P'S EVEN P. P. :
0329			* IN 2ND HALF OF SEGMENT T3= T1+PWCOM T1+PWCOM
0330			* J=(-) IF 1ST HALF EVEN PP, OR IF 2ND HALF ODD PP
0331			* J=(+) IF 1ST HALF ODD PP, OR IF 2ND HALF EVEN PP
0332			*
0333	E0FB 7D 0024		TST J ; IS J +/- ?
0334	E0FE 2B 20		BMI MINDX ; BRANCH FOR (-) INDEX
0335			*
0336			* ALGORITHM 1
0337			* CALCULATE SWITCHING TIMES FOR:
0338			* PVS#DPF AND PWCOM IS 2ND COMPLEMENT
0339			* OR PVS=DPF AND PWCOM IS 1ST COMPLEMENT
0340			*
0341			* T1=(TPP-PWDOM)/2
0342			*
0343	E100 96 70		LDA A TPP ; LOAD TOTAL PP COUNT MS
0344	E102 06 71		LDA B TPP+1 ; LOAD " " " LS
0345	E104 00 73		SUB B PWDOM+1 ; TPP-PWDOM LS
0346	E106 92 72		SBC A PWDOM ; TPP-PWDOM MS
0347	E108 47		ASR A ; DIVIDE DIFFERENCE
0348	E109 56		ROR B ; BY 2
0349	E10A 97 78		STA A T1 ; STORE MS
0350	E10C 07 79		STA B T1+1 ; STORE LS
0351			*
0352			* T3=T1+PWDOM
0353			*
0354	E10E 08 73		ADD B PWDOM+1 ; T1+PWDOM LS
0355	E110 99 72		ADC A PWDOM ; T1+PWDOM MS
0356	E112 97 7C		STA A T3 ; STORE MS
0357	E114 07 7D		STA B T3+1 ; STORE LS
0358			*
0359			* T2=T3-PWCOM
0360			*
0361	E116 00 75		SUB B PWCOM+1 ; T3-PWCOM LS
0362	E118 92 74		SBC A PWCOM ; T3-PWCOM MS
0363	E11A 97 7A		STA A T2 ; STORE T2 MS
0364	E11C 07 7B		STA B T2+1 ; STORE T2 LS
0365	E11E 20 22		BRA OTCNTS ; BRANCH TO OUTPUT COUNTS

SEQ	LOC	OBJ	SOURCE
0367			*
0368			* CALCULATE THE SWITCHING TIMES FOR:
0369			* PVS#DPF AND PWCOM IS THE 1ST COMP
0370			* OR PVS=DPF AND PWCOM IS THE 2ND COMP
0371			*
0372			* T1=(TPP-PWDOM)/2
0373			*
0374	E120	96 70	MINDX LDA A TPP ;LOAD TOTAL PP MS
0375	E122	06 71	LDA B TPP+1 ;LOAD " " LS
0376	E124	00 73	SUB B PWDOM+1 ;TPP-PWDOM LS
0377	E126	92 72	SBC A PWDOM ;TPP-PWDOM MS
0378	E128	47	ASR A ;DIVIDE DIFFERENCE
0379	E129	56	ROR B ; BY 2
0380	E12A	97 78	STA A T1 ;STORE MS
0381	E12C	07 79	STA B T1+1 ;STORE LS
0382			*
0383			* T3=T1+PWDOM
0384			*
0385	E12E	08 73	ADD B PWDOM+1 ;T1+PWDOM LS
0386	E130	99 72	ADC A PWDOM ;T1+PWDOM MS
0387	E132	97 70	STA A T3 ;STORE MS
0388	E134	07 70	STA B T3+1 ;STORE LS
0389			*
0390			* T2=T1+PWCOM
0391			*
0392	E136	96 78	LDA A T1 ;LOAD T1 MS
0393	E138	06 79	LDA B T1+1 ;LOAD T1 LS
0394	E13A	08 75	ADD B PWCOM+1 ;T1+PWCOM LS
0395	E13C	99 74	ADC A PWCOM ;T1+PWCOM MS
0396	E13E	97 7A	STA A T2 ;STORE MS
0397	E140	07 7B	STA B T2+1 ;STORE LS
0398	E142	96 09	OTCNTS LDA A #09 ;GET DIR CMD FLAG
0399	E144	91 10	CMP A DIROUT ;COMPARE WITH DIR OUT
0400	E146	27 0E	BEQ DAN1
0401	E148	96 4C	LDA A FOUT ;GET OUTPUT FREQ
0402	E14A	27 0A	BEQ DAN1
0403	E14C	86 00	LDA A #00
0404	E14E	97 46	STA A FCMD ;SET FREQ CMD TO ZERO
0405	E150	91 10	CMP A DIROUT ;IS DIRECTION CW?
0406	E152	27 0A	BEQ DAN2
0407	E154	20 0B	BRA DAN3
0408	E156	96 09	DAN1 LDA A DIRCMD
0409	E158	97 10	STA A DIROUT
0410	E15A	96 09	LDA A DIRCMD
0411	E15C	26 03	BNE DAN3
0412	E15E	7E E242	DAN2 JMP CWRT
0413	E161	7E E342	DAN3 JMP CCWRT

SEQ	LOC	OBJ	SOURCE
0415		E242	ORG \$E242
0416			*
0417			* CLOCKWISE ROUTINE
0418			*
0419	E242	96 05	CWRT LDA A NPPCT ; GET NO. OF PP COUNTED
0420	E244	84 01	AND A #1 ; IS IT ODD?
0421	E246	27 3A	BEQ EVPP1 ; BRANCH FOR EVEN PP
0422			*
0423			* DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE
0424			* DETERMINE THE DOMINANT PHASE (INIT: 0=A, 1=B, 2=C)
0425			*
0426	E248	96 20	LDA A DOMID ; LOAD DOMINANT PHASE ID
0427	E24A	4A	DEC A ; DECREMENT PHASE (B-A-C ALWAYS)
0428	E24B	27 13	BEQ PHBD01 ; BRANCH B PHASE DOMINANT (0)
0429	E24D	2A 22	BPL PHCD01 ; BRANCH C PHASE DOMINANT (00)
0430			*
0431			* PHASE A IS DOMINANT
0432			* DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE A-C-B
0433			*
0434	E24F	DE 78	LDX T1 ; LOAD T1 COUNT
0435	E251	FF 4002	STX PTM1+2 ; T1 TO PHASE A (DOM/1ST ON)
0436	E254	DE 7A	LDX T2 ; LOAD T2 COUNT
0437	E256	FF 4006	STX PTM1+6 ; T2 TO PHASE C (2ND ON/1ST OFF)
0438	E259	DE 7C	LDX T3 ; LOAD T3 COUNT
0439	E25B	FF 4004	STX PTM1+4 ; T3 TO PHASE B (DOM/2ND OFF)
0440	E25E	20 5A	BRA TMDPT1 ; BRANCH TO TEST FOR MID-POINT
0441			*
0442			* PHASE B IS DOMINANT B-A-C
0443			*
0444	E260	DE 78	PHBD01 LDX T1 ; LOAD T1 COUNT
0445	E262	FF 4004	STX PTM1+4 ; T1 TO PHASE B (DOM/1ST ON)
0446	E265	DE 7A	LDX T2 ; LOAD T2 COUNT
0447	E267	FF 4002	STX PTM1+2 ; T2 TO PHASE A (2ND ON/1ST OFF)
0448	E26A	DE 7C	LDX T3 ; LOAD T3 COUNT
0449	E26C	FF 4006	STX PTM1+6 ; T3 TO PHASE C (DOM/2ND OFF)
0450	E26F	20 49	BRA TMDPT1 ; BRANCH TO TEST FOR MID-POINT
0451			*
0452			* PHASE C IS DOMINANT C-B-A
0453			*
0454	E271	DE 78	PHCD01 LDX T1 ; LOAD T1 COUNT
0455	E273	FF 4006	STX PTM1+6 ; T1 TO PHASE C (DOM/1ST ON)
0456	E276	DE 7A	LDX T2 ; LOAD T2 COUNT
0457	E278	FF 4004	STX PTM1+4 ; T2 TO PHASE B (2ND ON/1ST OFF)
0458	E27B	DE 7C	LDX T3 ; LOAD T3 COUNT
0459	E27D	FF 4002	STX PTM1+2 ; T3 TO PHASE A (DOM/2ND OFF)
0460	E280	20 38	BRA TMDPT1 ; BRANCH TO TEST FOR MID-POINT

SEQ	LOC	OBJ	SOURCE
0462			*
0463			* DOMINANT POLARITY EQUALS POLE VOLTAGE STATE
0464			* DETERMINE THE DOMINANT PHASE
0465			*
0466	E282	96 20	EVFP1 LDA A DOMID ;LOAD DOMINANT ID
0467	E284	4A	DEC A ;DECREMENT PHASE (--A, 0=B, +=C)
0468	E285	27 13	BEQ PHBDE1 ;PHASE B DOMINANT - EVEN PP
0469	E287	2H 22	BPL PHCDE1 ;PHASE C DOMINANT - EVEN PP
0470			*
0471			* PHASE A IS DOMINANT B-C-A
0472	E289	DE 78	LDX T1 ;LOAD T1 COUNT
0473	E28B	FF 4004	STX PTM1+4 ;T1 TO PHASE B (DOM/2ND ON)
0474	E28E	DE 7A	LDX T2 ;LOAD T2 COUNT
0475	E290	FF 4006	STX PTM1+6 ;T2 TO PHASE C
0476	E293	DE 7C	LDX T3 ;LOAD T3 COUNT
0477	E295	FF 4002	STX PTM1+2 ;T3 TO PHASE A
0478	E298	20 20	BRA TMDPT1 ;BRANCH TO TEST FOR MID-POINT
0479			*
0480			* PHASE B IS DOMINANT C-A-B
0481			*
0482	E29A	DE 78	PHBDE1 LDX T1 ;LOAD T1 COUNT
0483	E29C	FF 4006	STX PTM1+6 ;T1 TO PHASE C
0484	E29F	DE 7A	LDX T2 ;LOAD T2 COUNT
0485	E2A1	FF 4002	STX PTM1+2 ;T2 TO PHASE A
0486	E2A4	DE 7C	LDX T3 ;LOAD T3 COUNT
0487	E2A6	FF 4004	STX PTM1+4 ;T3 TO PHASE B
0488	E2A9	20 0F	BRA TMDPT1 ;BRANCH TO TEST FOR MID-POINT
0489			*
0490			* PHASE C IS DOMINANT A-B-C
0491			*
0492	E2AB	DE 78	PHCDE1 LDX T1 ;LOAD T1 COUNT
0493	E2AD	FF 4002	STX PTM1+2 ;T1 TO PHASE A
0494	E2B0	DE 7A	LDX T2 ;LOAD T2 COUNT
0495	E2B2	FF 4004	STX PTM1+4 ;T2 TO PHASE B
0496	E2B5	DE 7C	LDX T3 ;LOAD T3 COUNT
0497	E2B7	FF 4006	STX PTM1+6 ;T3 TO PHASE C

SEQ	LOC	OBJ	SOURCE
0499			*
0500			* TEST IF THIS IS THE MID-POINT IN THE 60 DEG SEGMENT
0501			*
0502	E2BA	96 C6	TM0PT1 LDA A NPPMD ;GET NO. FOR MID-POINT P.P.
0503	E2BC	91 C5	CMP A NPPCT ;COMPARE TO PULSE PERIOD COUNTED
0504	E2BE	26 08	BNE NMDPT1 ;BRANCH IF NOT MID-POINT
0505			*
0506			* MID-POINT OF 60 DEG SEGMENT ENCOUNTERED
0507			* COMPLEMENT 'J' INDEX (COUNT DOWN -) TO ACCOUNT FOR
0508			* FOR COMPLEMENTARY PULSE WIDTH BECOMING 1ST COMP
0509			* SET FTPTR INDEX TO COUNTDOWN
0510			*
0511	E2C0	73 0024	COM J ;COMPLEMENT J INDEX
0512	E2C3	CE FFFC	LDX #-04 ;SET FTPTR INDEX
0513	E2C6	DF C8	STX FTINCR ;STORE NEW INCREMENT VALUE
0514			*
0515			* UPDATE POINTERS AND INDICES AND TEST FOR END OF 60 DEG SEC
0516			*
0517	E2C8	73 0024	NMDPT1 COM J ;J = NOT J
0518	E2CB	96 C3	LDA A FTPTR ;LOAD PRESENT POSITION MS
0519	E2CD	D6 C4	LDA B FTPTR+1 ; " " " LS
0520	E2CF	D8 C9	ADD B FTINCR+1 ;MOVE IN TABLE MS
0521	E2D1	99 C8	ADC A FTINCR ; " " " LS
0522	E2D3	97 C3	STA A FTPTR ;STORE IN NEW POSITION MS
0523	E2D5	D7 C4	STA B FTPTR+1 ; IN TABLE LS
0524			*
0525			* TEST FOR END OF PULSE PERIODS
0526			*
0527	E2D7	7A 00C5	DEC NPPCT ;DECR TOTAL NO. OF PP
0528	E2DA	26 03	BNE NPPTD1 ;SKIP IF MORE PP TO GO
0529	E2DC	73 00C0	COM F60DG ;SET 60 DEG FLAG
0530			*
0531			* EVERY PULSE PERIOD WILL OUT SET/PRESET OR NULL
0532			* OPERATION COMMAND TO PIA TO SET UP FLIP FLOPS
0533			*
0534	E2DF	96 44	NPPTD1 LDA A PIACMD ;GET PIA CMD'D STATE
0535	E2E1	87 5002	STA A PIA+2 ;START CONV TO CLEAR DETECT
0536	E2E4	86 72	LDA A #\$72 ;PR/RST=1, NEW MUX ADDR-SPD CMD
0537	E2E6	97 44	STA A PIACMD ;CLEAR PIA CMD
0538			*
0539			* DISMISS THE INTERRUPT
0540			*
0541	E2E8	3B	RTI
0542		E342	ORG #E342
0543			*
0544			* COUNTERCLOCKWISE ROUTINE
0545			*
0546	E342	96 C5	CCWRT LDA A NPPCT ;GET NO. OF PP COUNTED
0547	E344	84 01	AND A #1 ;IS IT ODD?
0548	E346	26 3A	BNE EVPP2 ;BRANCH FOR EVEN PP
0549			*
0550			* DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE
0551			* DETERMINE THE DOMINANT PHASE (INIT: 0=A, 1=B, 2=C)
0552			*

SEQ	LOC	OBJ	SOURCE
0553	E348	96 20	LDA A DOMID
0554	E34A	4A	DEC H
0555	E34B	27 24	BEQ PHC002
0556	E34D	2A 11	BPL PHB002
0557			*
0558			* PHASE A IS DOMINANT
0559			* DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE A-C-B
0560			*
0561	E34F	DE 78	LDX T1
0562	E351	FF 4002	STX PTM1+2
0563	E354	DE 7A	LDX T2
0564	E356	FF 4006	STX PTM1+6
0565	E359	DE 7C	LDX T3
0566	E35B	FF 4004	STX PTM1+4
0567	E35E	20 5A	BRA TMDPT2
0568			*
0569			* PHASE B IS DOMINANT B-A-C
0570			*
0571	E360	DE 78	PHB002 LDX T1
0572	E362	FF 4004	STX PTM1+4
0573	E365	DE 7A	LDX T2
0574	E367	FF 4002	STX PTM1+2
0575	E36A	DE 7C	LDX T3
0576	E36C	FF 4006	STX PTM1+6
0577	E36F	20 49	BRA TMDPT2
0578			*
0579			* PHASE C IS DOMINANT C-B-A
0580			*
0581	E371	DE 78	PHC002 LDX T1
0582	E373	FF 4006	STX PTM1+6
0583	E376	DE 7A	LDX T2
0584	E378	FF 4004	STX PTM1+4
0585	E37B	DE 7C	LDX T3
0586	E37D	FF 4002	STX PTM1+2
0587	E380	20 38	BRA TMDPT2

SEQ	LOC	OBJ	SOURCE
0589			*
0590			* DOMINANT POLARITY EQUALS POLE VOLTAGE STATE
0591			* DETERMINE THE DOMINANT PHASE
0592			*
0593	E382	96 20	EVPP2 LDA A DOMID ; LOAD DOMINANT ID
0594	E384	4A	DEC A ; DECREMENT PHASE (--A, 0=B, +=C)
0595	E385	27 24	BEQ PHCDE2 ; PHASE B DOMINANT - EVEN PP
0596	E387	2A 11	BPL PHBDE2 ; PHASE C DOMINANT - EVEN PP
0597			*
0598			* PHASE A IS DOMINANT B-C-A
0599	E389	DE 78	LDX T1 ; LOAD T1 COUNT
0600	E38B	FF 4004	STX PTM1+4 ; T1 TO PHASE B (DOM/2ND ON)
0601	E38E	DE 7A	LDX T2 ; LOAD T2 COUNT
0602	E390	FF 4006	STX PTM1+6 ; T2 TO PHASE C
0603	E393	DE 7C	LDX T3 ; LOAD T3 COUNT
0604	E395	FF 4002	STX PTM1+2 ; T3 TO PHASE A
0605	E398	20 20	BRA TMDPT2 ; BRANCH TO TEST FOR MID-POINT
0606			*
0607			* PHASE B IS DOMINANT C-A-B
0608			*
0609	E39A	DE 78	PHBDE2 LDX T1 ; LOAD T1 COUNT
0610	E39C	FF 4006	STX PTM1+6 ; T1 TO PHASE C
0611	E39F	DE 7A	LDX T2 ; LOAD T2 COUNT
0612	E3A1	FF 4002	STX PTM1+2 ; T2 TO PHASE A
0613	E3A4	DE 7C	LDX T3 ; LOAD T3 COUNT
0614	E3A6	FF 4004	STX PTM1+4 ; T3 TO PHASE B
0615	E3A9	20 0F	BRA TMDPT2 ; BRANCH TO TEST FOR MID-POINT
0616			*
0617			* PHASE C IS DOMINANT A-B-C
0618			*
0619	E3AB	DE 78	PHCDE2 LDX T1 ; LOAD T1 COUNT
0620	E3AD	FF 4002	STX PTM1+2 ; T1 TO PHASE A
0621	E3B0	DE 7A	LDX T2 ; LOAD T2 COUNT
0622	E3B2	FF 4004	STX PTM1+4 ; T2 TO PHASE B
0623	E3B5	DE 7C	LDX T3 ; LOAD T3 COUNT
0624	E3B7	FF 4006	STX PTM1+6 ; T3 TO PHASE C

```

SEQ  LOC  OBJ  SOURCE
0626      *
0627      * TEST IF THIS IS THE MID-POINT IN THE 60 DEG SEGMENT
0628      *
0629 E3BA 96 06  TMDPT2  LDA A  NPPMD      ;GET NO. FOR MID-POINT P. P.
0630 E3BC 91 05      CMP A  NPPCT      ;COMPARE TO PULSE PERIOD COUNT
0631 E3BE 26 08      BNE     NMDPT2      ;BRANCH IF NOT MID-POINT
0632      *
0633      * MID-POINT OF 60 DEG SEGMENT ENCOUNTERED
0634      *   COMPLEMENT "J" INDEX (COUNT DOWN - ) TO ACCOUNT FOR
0635      *   FOR COMPLEMENTARY PULSE WIDTH BECOMING 1ST COMP
0636      * SET FTPTR INDEX TO COUNTDOWN
0637      *
0638 E3C0 73 0024      COM     J          ;COMPLEMENT J INDEX
0639 E3C3 0E FFFC      LDX     #-04      ;SET FTPTR INDEX
0640 E3C6 0F 08      STX     FTINCR      ;STORE NEW INCREMENT VALUE
0641      *
0642      * UPDATE POINTERS AND INDICES AND TEST FOR END OF 60 DEG SEG
0643      *
0644 E3C8 73 0024  NMDPT2  COM     J          ;J = NOT J
0645 E3CB 96 03      LDA A  FTPTR      ;LOAD PRESENT POSITION MS
0646 E3CD 06 04      LDA B  FTPTR+1    ; " " " " " " LS
0647 E3CF 0B 09      ADD B  FTINCR+1  ;MOVE IN TABLE MS
0648 E3D1 99 08      ADC A  FTINCR    ; " " " " " " LS
0649 E3D3 97 03      STA A  FTPTR      ;STORE IN NEW POSITION MS
0650 E3D5 07 04      STA B  FTPTR+1    ; IN TABLE " " " " LS
0651      *
0652      * TEST FOR END OF PULSE PERIODS
0653      *
0654 E3D7 7A 00C5      DEC     NPPCT      ;DECR TOTAL NO. OF PP
0655 E3DA 26 03      BNE     MPPTD2     ;SKIP IF MORE PP TO GO
0656 E3DC 73 00C0      COM     F60DG     ;SET 60 DEG FLAG
0657      *
0658      * EVERY PULSE PERIOD WILL OUT SET/PRESET OR NULL
0659      * OPERATION COMMAND TO PIA TO SET UP FLIP FLOPS
0660      *
0661 E3DF 96 44  MPPTD2  LDA A  PIACMD    ;GET PIA CMD'D STATE
0662 E3E1 B7 5002      STA A  PIA+2      ;START CONV TO CLEAR DETECT
0663 E3E4 86 72      LDA A  #$72      ;PR/RST=1, NEW MUX ADDR-SPD CMD
0664 E3E6 97 44      STA A  PIACMD    ;CLEAR PIA CMD
0665      *
0666      * DISMISS THE INTERRUPT
0667      *
0668 E3E8 3B      RTI
0669      END

```

0669 LINES ASSEMBLED, LOC = E3E9, 0000 ERRORS DETECTED.

SYMBOL	VALUE	ATTR	LOCN	LINK
F60DG	00C0	84	2AA2	
FTPNI	00C1	84	2AA8	
FTPTR	00C3	84	2AB4	
NPPMD	00C6	84	2ABD	
BSTAR	00D2	84	2AC6	
SF	00D0	84	2ACF	
PWCOM	0072	84	2AD8	
PWCOM	0074	84	2AE1	
TPP	0070	84	2AEA	
T1	0078	84	2AF3	
T2	007A	84	2AFC	
T3	007C	84	2B05	
VPADD	00D8	84	2B0E	
SFMAX	0030	84	2B17	
SFMIN	0032	84	2B20	
PIACMD	0044	84	2B29	
DOMID	0020	84	2B32	
DFF	0022	84	2B3B	
J	0024	84	2B44	
FTINCR	00C8	84	2B4D	
MONFG	0051	84	2B56	
VSRW	003A	84	2B5F	
VSFLG	00E7	84	2B68	
BPRAW	004A	84	2B71	
FCRAW	0038	84	2B7A	
ESTAR	003C	84	2B83	
NPPCT	00C5	84	2B8C	
FOUT	004C	84	2B95	
FCMD	0046	84	2B9E	
DIRCMD	0009	84	2BA7	
DIROUT	0010	84	2BB0	
MPIER	2000	84	2BB9	
MCAND	2002	84	2BC2	
PROD	2005	84	2BCB	
PIA	5000	84	2BD4	
PTM1	4000	84	2BD0	
PTM2	3000	84	2BE6	
WLATCH	2007	84	2BEF	
PTIS	E000	84	2BF8	
NOT60	E055	84	2C01	
AA	E036	84	2C0A	
A2	E041	84	2C13	
BPIN	E07D	84	2C1C	
CBPC	E084	84	2C25	
SVFADD	E09A	84	2C2E	
SFCAL	E0A0	84	2C37	
SFMINL	E0CC	84	2C40	
SFMSZ	E0C8	84	2C49	
SFOK	E0D0	84	2C52	
MINDX	E120	84	2C5B	
OTCNTS	E142	84	2C64	
DAN1	E156	84	2C6D	
DAN2	E15E	84	2C76	
DAN3	E161	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
CWRT	E242	84	2C88	
CCWRT	E342	84	2C91	
EVPP1	E282	84	2C9A	
PHB001	E260	84	2CA3	
PHC001	E271	84	2CAC	
TMDPT1	E2BA	84	2CB5	
PHB0E1	E29A	84	2CBE	
PHC0E1	E2AB	84	2CC7	
NMDPT1	E2C8	84	2CD0	
MPPTD1	E2DF	84	2CD9	
EVPP2	E382	84	2CE2	
PHC002	E371	84	2CEB	
PHB002	E360	84	2CF4	
TMDPT2	E3BA	84	2CFD	
PHC0E2	E3AB	84	2D06	
PHB0E2	E39A	84	2D0F	
NMDPT2	E3C8	84	2D18	
MPPTD2	E3DF	C4	2D21	

END ASM V1.1

SEQ LOC OBJ SOURCE

```

0001      NAM      FEEDBACK CONTROL MPU PROGRAM
0002      * MP917
0003      * NSKDC
0004      FFF8      ORG      $FFFF
0005      0001      LSYMB    1
0006      FFF8 FC 00      IRUV      FDB      $FC00      ; IRQ VECTOR, E*OUTPUT
0007      FFFA F8 00      SWIV      FDB      $F800      ; SWI VECTOR
0008      FFFC F8 00      NMIV      FDB      $F800      ; NMI VECTOR
0009      FFFE F8 00      RSTV      FDB      $F800      ; RESET VECTOR
0010      FCD0      ORG      $FCD0
0011      **CONSTANTS FOR FLTCL
0012      *
0013      * FILTER OF FORM - (S+W1)/(S+W2)
0014      * - WHERE W1 = "ZERO" AT 180 HZ.
0015      * -           W2 = "POLE" AT 0.18 HZ.
0016      *
0017      * VALUES CALCULATED FOR DELTA T = 6.0 MS.
0018      FCD0 3F EE      RADD      FDB      16366      ; A CONSTANT (0.998920583)
0019      FCD2 D0 7D      BADD      FDB      ~8835      ; B CONSTANT (-0.539275)
0020      FCD4 22 80      CADD      FDB      08832      ; C CONSTANT (0.53906317)
0021      * DEFINITIONS REQUIRED FOR SUBROUTINES& PROGRAM
0022      *
0023      0001      FREQ      EQU      $0001      ; FREQUENCY ADDRESS
0024      0002      IQAD      EQU      $0002      ; IQ ADDRESS
0025      0003      VQAD      EQU      $0003      ; VQ ADDRESS
0026      0004      VQAD      EQU      $0004      ; VQ ADDRESS
0027      0005      IDAD      EQU      $0005      ; ID ADDRESS
0028      0012      ERMT      EQU      $0012      ; ERROR MS BYTE
0029      0013      ERLT      EQU      $0013      ; ERROR LS BYTE
0030      C000      LLS      EQU      $0C000      ; LATCH ERROR LS BYTE
0031      B000      LMS      EQU      $0B000      ; LATCH ERROR MS BYTE
0032      0012      MPRM      EQU      $12      ; MULTIPLIER MS BYTE
0033      0013      MPRL      EQU      $13      ; MULTIPLIER LS BYTE
0034      0010      MPDM      EQU      $10      ; MULTIPLICAND MS BYTE
0035      0011      MPDL      EQU      $11      ; MULTIPLICAND LS BYTE
0036      0011      ADDL      EQU      $11      ; ADDEND LS BYTE
0037      0010      ADDM      EQU      $10      ; ADDEND MS BYTE
0038      0013      SUML      EQU      $13      ; SUM LS BYTE
0039      0012      SUMM      EQU      $12      ; SUM MS BYTE
0040      00FD      RADD      EQU      $0FD      ; R* CORRECTION K-MS ADDR
0041      00FE      KADD      EQU      $0FE      ; 2*PI*F*L1 ADDR MS BYTE
0042      0010      PROD      EQU      MPDM      ; MS BYTE-PRODUCT
0043      *PRODUCT IS 4 BYTES FROM PROD TO PROD+3
0044      0010      DIFF      EQU      MPDM      ; DIFFERENCE MS BYTE
0045      * DIFFERENCE IS 2 OR 4 BYTES FROM DIFF TO DIFF+3
0046      0032      ESTAR      EQU      $32      ; E* AFTER FLXCL 032,33
0047      0021      AXMS      EQU      $21      ; A*KN BYTES(21-25)
0048      0026      BUN1M      EQU      $26      ; B*UN1 MS BYTES
0049      0028      BUN1LM      EQU      $28      ; B*UN1 LS MID & LS
0050      002A      CUNM      EQU      $2A      ; C*UN MS BYTES
0051      002C      CUNLM      EQU      $2C      ; C*UN LS MID & LS
0052      0032      UN1      EQU      $32      ; UN+1 2 BYTES
0053      0047      XN      EQU      $47      ; XN - 3 BYTES
0054      0034      UN      EQU      $34      ; UN - 2 BYTES
0055      002A      CMB      EQU      CUNM      ; C*UN-B*XN1

```

SEQ LOC OBJ SOURCE

0056		0026	XN1T	EQU	BUN1M	;XN1 TEMPORARY LOC
0057		0040	ULDF	EQU	\$40	;OLD FREQ VALUE
0058		0041	NEWF	EQU	\$41	;NEW FREQ VALUE
0059		0043	TMONT	EQU	\$43	;TIME INTERVAL TWEEN ERRST'S

SEQ	LOC	OBJ	SOURCE
0061			**** SCHLING COMMENTS FOR PROGRAM ****
0062			* EXISTING LOAD TRIAL
0063			*WORD SCHLING INTERNAL TO PROGRAM
0064			*
0065			*****FORMATS*****
0066			*
0067			* B B B B B B B B B B B B B B B
0068			*31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
0069			* B B B B B B B B B B B B B B B
0070			*15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0071			*
0072			*VD & VQ
0073			*5 64 32 16 8 4 2 1 0 0 0 0 0 0 0
0074			*ID &IQ
0075			*5 128 64 32 16 8 4 2 0 0 0 0 0 0 0
0076			*

SEQ	LOC	OBJ	SOURCE
0078			*
0079		F800	URG \$F800
0080			* INITIALIZATION SEQUENCE (ALL VECTORS POINT HERE)
0081			* SEQUENCE ACCOMPLISHES THE FOLLOWING :
0082			*
0083			* STACK POINTER SET TO TOP OF RAM (ADDR = 03FF)
0084			* PIA IS SET UP
0085			* PA = INPUTS (A/D'S OR FREQ)
0086			* PB = OUTPUTS
0087			* PB INITIAL VALUES PB7-PB4 = 0; NO A/D INPUTS ENABLED
0088			* PB1 = 1 NO FREQ INPUT ENABLED
0089			* PB0 = 0 NO START CONV COMMAND
0090			* ERROR LATCH SET AT ALL "0'S"
0091			*
0092			*
0093			*
0094	F800	8E 03FF	ISTART LDS #\$03FF ; SET TOP OF STACK
0095	F803	86 00	PAS LDA A #\$00 ; SET CRA WORD; DDRA CHOSEN
0096	F805	CE 0000	LDX #\$0000 ; START ADDR
0097	F808	A7 00	PAS1 STA A 0,X ; CLEAR RAM BYTE
0098	F80A	08	INX ; INCR BYTE COUNT
0099	F80B	8C 0100	CPX #\$0100 ; TEST FOR LAST ADDR
0100	F80E	26 F8	BNE PAS1 ; BRANCH FOR MORE ADDR
0101	F810	B7 D001	STA A \$D001 ; STORE IN CRA; DDRA ENABLED
0102	F813	B7 D000	STA A \$D000 ; MAKE PA ALL INPUTS
0103	F816	86 04	LDA A #\$04 ; SET CRA WD, PRA CHOSEN
0104	F818	B7 D001	STA A \$D001 ; ENABLE PRA FOR READ
0105	F81B	86 00	PBS LDA A #\$00 ; SET CRB WD, DDRB CHOSEN
0106	F81D	B7 D003	STA A \$D003 ; SET CRB, DDRB ENABLED
0107	F820	86 FF	LDA A #\$FF ; SET DDRB WD
0108	F822	B7 D002	STA A \$D002 ; SET DDRB TO ALL OUTPUTS
0109	F825	86 04	LDA A #\$04 ; SET CRB WD, PRB CHOSEN
0110	F827	B7 D003	STA A \$D003 ; STORE IN CRB, PRB ENABLED
0111	F82A	86 02	LDA A #\$02 ; LD PRB WD
0112	F82C	B7 D002	STA A \$D002 ; SET PRB (NOTHING GOING)
0113	F82F	86 00	EVS LDA A #\$00 ; SET INIT ERROR VALUE
0114	F831	B7 B000	STA A LMS ; STORE ERROR WD MS 6
0115	F834	B7 C000	STA A LLS ; STORE ERROR WD LS 6
0116	F837	86 FE	LDA A #KADD ; LD CONSTANT ADDR MSB
0117	F839	97 00	STA A \$00 ; STORE ADDR
0118	F83B	86 01	LDA A #\$01 ; LD TIME INTERVAL LS
0119	F83D	97 44	STA A TMCNT+1 ; STORE LS OF INTERVAL
0120	F83F	0E	CLI ; CLEAR INT MASK

SEQ	LOC	OBJ	SOURCE
0122			* MAIN SEQUENCE PROGRAM
0123			* INPUT OF DATA (F, VD, ID, VO, IQ) IS FIRST
0124			* FLUX CALCULATION IS NEXT
0125			* FILTER CALCULATION IS THIRD
0126			* OUTPUT OF DATA WORD IS FINAL OPERATION
0127			* LOOP BACK TO INPUT NEXT SAMPLE
0128			*
0129	F840	BD F85F	INPUT JSR INP ;JSR INPUT ROUTINE
0130	F843	BD F841	JSR RECTFY ;JSR RECTIFY ID FOR OVERLOAD PROG
0131	F846	BD F9CA	JSR FLTCL ;JSR FILTER CALCULATION
0132	F849	BD FAD7	JSR ERRST ;JSR OUTPUT ROUTINE
0133	F84C	7E F840	JMP INPUT ;JUMP & CONTINUE
0134			*
0135			*
0136			* GLOBALS ARE :
0137			* INP - INPUT ROUTINE
0138			* FLXCL - FLUX ROUTINE
0139			* FLTCL - FILTER ROUTINE
			* ERRST - ERROR WORD OUTPUT ROUTINE

SEQ	LOC	OBJ	SOURCE
0141			*
0142			***** SUBROUTINES *****
0143			*
0144			* INP (INPUT OF FREQ, VOLTAGE, & CURRENT)
0145			* (F, VD, VQ, ID, IQ)
0146			* READS FREQ IN PA
0147			* DOES A/D CONVERSION
0148			* STORES ALL VALUES DIRECT
0149			* MEMORY WHERE VALUES STORED IS :
0150			* ADDR CONTENTS
0151			* 0000 "FE"
0152			* 0001 F
0153			* 0002 IQ
0154			* 0003 VD
0155			* 0004 VQ
0156			* 0005 ID
0157			*
0158			* CONTROL OF PIH IS VIA PORT B
0159			* THESE WORDS ARE RECOGNIZED *
0160			* B7 B6 B5 B4 B3 B2 B1 B0
0161			* 0 0 0 0 X X 1 0 RESET VALUE
0162			* 0 0 0 0 X X 1 1 START CONVERSION
0163			* 0 0 0 1 X X 1 0 READ IQ
0164			* 0 0 1 0 X X 1 0 READ VD
0165			* 0 1 0 0 X X 1 0 READ VQ
0166			* 1 0 0 0 X X 1 0 READ ID
0167			* 0 0 0 0 X X 0 0 READ FREQUENCY
0168			*
0169			*F AT INPUT IS B6 B5 B4 B3 B2 B1 X X
0170			***
0171	F85F		ORG \$F85F
0172	0055	FZFG	EQU \$55 ; FREQ =0 FLAG
0173	F85F	86 00	INP LDA A #\$00 ; LD READ FREQ WD
0174	F861	B7 D002	STA A \$D002 ; STORE IN PRB
0175	F864	B6 D000	LDA A \$D000 ; READ FREQ IN PRA
0176	F867	43	COM A ; F WAS INVERTED
0177	F868	44	LSR A ; SCALE FREQUENCY
0178	F869	44	LSR A ; SCALE FREQUENCY
0179			*TEST FOR FREQ=0, IF YES HOLD E*=0 TO FILTER
0180	F86A	81 00	CMP A #\$00 ; TEST FOR F=0
0181	F86C	26 06	BNE CNFI ; BRANCH IF NOT 0
0182	F86E	C6 01	LDA B #\$01 ; ESTABLISH FLAG VALUE
0183	F870	D7 55	STA B FZFG ; SET FLAG
0184	F872	20 03	BRA CNFI1 ; BRANCH TO CONT INP
0185	F874	7F 0055	CNFI CLR FZFG ; RESET F=0 FLAG
0186	F877	97 41	CNFI1 STA A NEWF ; STORE NEW FREQ
0187	F879	80 06	SUB A #06 ; GEN F OFFSET
0188	F87B	2A 02	BPL GT6 ; BRANCH IF >6
0189	F87D	86 00	LDA A #\$00 ; LD MIN OFFSET=6HZ
0190	F87F	48	GT6 ASL A ; POSITION OFFSET
0191	F880	8B 06	ADD A #06 ; FREQ POINTER
0192	F882	97 01	STA A FREQ ; STORE VALUE
0193	F884	86 03	LDA A #\$03 ; SET START CONV WD
0194	F886	C6 02	LDA B #\$02 ; SET SC TURN OFF WD

SEQ LOC OBJ SOURCE

0195	F888	B7 D002	STA H \$D002	SET SC IN PRB
0196	F888	F7 D002	STA B \$D002	TURN OFF SC
0197	F88E	86 12	LDA H ##12	LD READ IQ WD
0198	F890	C6 18	LDA B ##18	LD LOOP CNT(MUST > 80MS)
0199	F892	5A	DEC B	DECR LOOP CNT(WAIT ON A/D)
0200	F893	26 FD	BNE ADLOOP	JUMP IF A/D NOT DONE
0201	F895	B7 D002	STA A \$D002	ENABLE IQ AT PRA
0202	F898	F6 D000	LDA B \$D000	LD IQ
0203	F89B	C8 7F	EOR B ##7F	2'S COMPL IQ
0204	F89D	D7 02	STA B IQAD	STORE IQ
0205	F89F	86 22	LDA A ##22	LD READ VQ WORD
0206	F8A1	B7 D002	STA A \$D002	ENAB VQ AT PRA
0207	F8A4	F6 D000	LDA B \$D000	LD VQ
0208	F8A7	C8 7F	EOR B ##7F	2'S COMPL VQ
0209	F8A9	D7 03	STA B VQAD	STORE VQ
0210	F8AB	86 42	LDA A ##42	LD READ VQ WORD
0211	F8AD	B7 D002	STA A \$D002	ENAB VQ AT PRA
0212	F8B0	F6 D000	LDA B \$D000	LD VQ
0213	F8B3	C8 7F	EOR B ##7F	2'S COMPL VQ
0214	F8B5	D7 04	STA B VQAD	STORE VQ
0215	F8B7	86 82	LDA A ##82	LD READ ID WORD
0216	F8B9	B7 D002	STA A \$D002	ENAB ID AT PRA
0217	F8BC	F6 D000	LDA B \$D000	LD ID
0218	F8BF	C8 7F	EOR B ##7F	2'S COMPL ID
0219	F8C1	D7 05	STA B IDAD	STORE ID
0220	F8C3	86 02	LDA A ##02	LD RESET VALUE TO A/D
0221	F8C5	B7 D002	STA A \$D002	RESET A/D ENABLES
0222	F8C8	86 FE	LDA A ##FE	LD K1 MS ADDR
0223	F8CA	97 00	STA A \$0000	STORE K1 MS ADDR
0224	F8CC	39	RTS	

SEQ LOC OBJ SOURCE

```

0226 *
0227 ***** FLXCL *****
0228 * TRIES TO USE COMMON SECTION OF RAM
0229 * FOR CALCULATION
0230 * SEQUENCE OF OPERATIONS IS :
0231 * LOOK UP 2*P1*L*F
0232 * K1*IQ = P1
0233 * VD-P1 = S1
0234 * S1*IQ = P2 STORED AT 0020,0021
0235 * K1*IQ = P3
0236 * VQ+P3 = S2
0237 * S2*IQ = P4
0238 * P2-P4 = 2WX
0239 * WXM=(VD-K1*IQ)IQ-(VQ+K1*IQ)IQ
0240 * R-WX = ERR STORED AT 0010,0011
0241 *
0242 * THIS SECTION DOES P1, S1, P2
0243 F8D0 ORG $F8D0
0244 F8D0 DE 00 FLXCL LDX $0000 ;LD K1 ADDRESS
0245 F8D2 A6 00 LDA H 0,X ;LD K1 MS
0246 F8D4 E6 01 LDA B 1,X ;LD K1 LS
0247 *
0248 *K1 FORMAT (RANGE F=6-60, 0.03769-0.3769)
0249 * L1 ASSUMED = 1.0 MILLIHENRIES
0250 *B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0251 * S 1 . 5 .25 X X X X X X X X X X
0252 *
0253 F8D6 36 AB PSH A ;TEMP STORE K1 MS
0254 F8D7 37 PSH B ;TEMP STORE K1 LS
0255 *P1 CALCULATION
0256 F8D8 97 10 STA A MPDM ;STORE K1 FOR MULT
0257 F8DA 07 11 STA B MPDL ;STORE K1 LS FOR MULT
0258 F8DC 06 00 LDA B #$00 ;LD IQ LS BYTE
0259 F8DE 96 02 LDA A IQAD ;LD IQ MS
0260 F8E0 97 12 PIQ STA A MPRM ;STORE IQ FOR MULT
0261 F8E2 07 13 STA B MPRL ;STORE IQ LS
0262 F8E4 BD FB58 AC JSR MPY16 ;DO K1*IQ
0263 *
0264 * K1*IQ FORMAT
0265 * B31 B30 B29 B28 B27 B26-----B01 B00
0266 * S S S S 64 32 . . . . .
0267 *
0268 *P1 DONE, SCALE USES PROD AND PROD+1 FOR P1 AFTER SCALING
0269 F8E7 86 02 LDA A #02 ;LD LOOP COUNT
0270 F8E9 78 0012 SCP1 ASL PROD+2 ;SHIFT LS
0271 F8EC 79 0011 ROL PROD+1 ;SHIFT MID BYTE
0272 F8EF 79 0010 ROL PROD ;SHIFT MS
0273 F8F2 4A DEC A ;DECR LOOP CNT
0274 F8F3 26 F4 BNE SCP1 ;BRANCH FOR MORE SCALING
0275 F8F5 96 10 LDA A PROD ;LD P1 MS
0276 F8F7 06 11 LDA B PROD+1 ;LD P1 LS, TRUNCATES
0277 F8F9 43 COM A ;SET -K1*IQ MS
0278 F8FA 53 COM B ;SET -K1*IQ LS
0279 *

```

SEQ	LOC	OBJ	SOURCE
0280			* K1*IQ FORMAT-ADJUSTED
0281			* S 128 64 32 16 8 4 2 1 . . 5 . 25 X X X X---->
0282			* RANGE (0.0--->96. X)
0283			*
0284	F8FB	97 10	STA A ADDM ; STORE P1 MS, FOR ADD
0285	F8FD	D7 11	STA B ADDL ; STORE P1, LS
0286	F8FF	C6 00	LDA B #\$00 ; LD VD LS BYTE
0287	F901	96 03	LDA A VDDA ; LD VD MS
0288			* SCALING OF VD, MOVE RIGHT 1 PLACE, K*ID MATCH
0289	F903	47	CVD ASR A ; MOVE MS 1 PLACE
0290	F904	56	ROR B ; MOVE LS 1PLACE
0291	F905	00	SEC ; SET CRY FOR SUBT
0292			*
0293			* VD FORMAT ADJUSTED
0294			* S 5 64 32 16 8 4 2 1 . . 5 . 25 X X X X
0295			* RANGE (+/- 128)
0296			*
0297			*DO (VD-K1*IQ); TRUNCATES TO 16 BITS
0298	F906	BD FB8D	AD JSR ADD16 ; DO VD+P1
0299			*S1 DONE
0300			* S1 FORMAT SAME AS VD ADJUSTED
0301	F909	C6 00	LDA B #\$00 ; LD IQ LS
0302	F90B	96 02	LDA A IQAD ; LD IQ MS FOR ADD
0303	F90D	97 10	SIC STA A MPDM ; STORE VD MS FOR MULT
0304	F90F	D7 11	STA B MPDL ; STORE VD LS
0305			*DO (VD-K1*IQ)*IQ
0306	F911	BD FB58	JSR MPY16 ; DO IQ*S1
0307			*
0308			* (VD-K1*IQ)*IQ FORMAT - 32 BITS
0309			* S X X X X X X X 8 4 2 1 . . 5 . 25 X X X---->LSB
0310			*
0311			*P2 DONE, STORE P2 FOR LATER USE
0312	F914	96 10	LDA A PROD ; LD P2 MS
0313	F916	D6 11	AE LDA B PROD+1 ; LD P2 LS
0314	F918	97 20	STA A \$20 ; STORE P2 TEMP
0315	F91A	D7 21	STA B \$21 ; STORE P2 MS MID TEMP
0316	F91C	96 12	LDA A PROD+2 ; LD P2 LS MID
0317	F91E	D6 13	LDA B PROD+3 ; LD P2 LS
0318	F920	97 22	STA A \$22 ; STORE P2 LS MID
0319	F922	D7 23	STA B \$23 ; STORE P2 LS
0320			* SECTION DOES P3, S2, & P4 CALCULATION
0321	F924	33	AF PUL B ; PULL K1 LS
0322	F925	32	PUL A ; PULL K1 MS
0323			*P3 CALCULATION
0324	F926	97 10	STA A MPDM ; STORE K1 FOR MULT
0325	F928	D7 11	STA B MPDL ; STORE K1 LS FOR MULT
0326	F92A	C6 00	LDA B #\$00 ; LD ID LS
0327	F92C	96 05	LDA A IDAD ; LD ID MS
0328	F92E	97 12	PID STA A MPRM ; STORE ID FOR MULT
0329	F930	D7 13	STA B MPRL ; STORE ID LS
0330	F932	BD FB58	JSR MPY16 ; DO K1*ID
0331			*
0332			* K1*ID FORMAT
0333			* B31 B30 B29 B28 B27 B26-----B01 B00
0334			* S S S S 64 32

SEQ	LOC	OBJ	SOURCE
0335			*
0336			*P3 DONE, SCALE USES PROD+1 AND PROD+2 FOR P3
0337	F935	86 02	LDA A #02 ;LD LOOP COUNT
0338	F937	78 0012	SCPS ASL PROD+2 ;SHIFT LS
0339	F93A	79 0011	ROL PROD+1 ;SHIFT MID BYTE
0340	F93D	79 0010	ROL PROD ;SHIFT MS
0341	F940	4A	DEC A ;DECR LOOP CNT
0342	F941	26 F4	BNE SCP3 ;BRANCH FOR MORE SCALING
0343	F943	96 10	LDA A PROD ;LD P3 MS
0344	F945	06 11	LDA B PROD+1 ;LD P3 LS, TRUNCATES
0345			*
0346			* K1*ID FORMAT-ADJUSTED
0347			* S 128 64 32 16 8 4 2 1 . . 5 . 25 X X X X---->
0348			* RANGE (0.0--->96.X)
0349			*
0350	F947	97 10	STA A ADDM ;STORE P3 MS, FOR ADD
0351	F949	07 11	STA B ADDL ;STORE P3, LS
0352	F94B	06 00	LDA B #00 ;SET V0 LS= 0
0353	F94D	96 04	LDA A V0AD ;LD V0 MS
0354			* SCALING OF V0, MOVE RIGHT 1 PLACE, K*ID MATCH
0355	F94F	47	CV0 ASR A ;MOVE MS 1 PLACE
0356	F950	56	ROR B ;MOVE LS 1PLACE
0357	F951	0C	CLC ;CLEAR CRY FOR ADD
0358			*
0359			* V0 FORMAT ADJUSTED
0360			* S S 64 32 16 8 4 2 1 . . 5 . 25 X X X X
0361			* RANGE (+/- 128)
0362			*
0363			*DO (V0+K1*ID), TRUNCATES TO 16 BITS
0364	F952	BD FB8D	JSR ADD16 ;DO V0+P3
0365			*S2 DONE
0366	F955	06 00	LDA B #00 ;LD ID LS
0367	F957	96 05	LDA A IDAD ;LD ID MS FOR ADD
0368	F959	97 10	SID STA A MPDM ;STORE V0 MS FOR MULT
0369	F95B	07 11	STA B MPDL ;STORE V0 LS
0370			*DO (V0+K1*ID)*ID
0371	F95D	BD FB58	JSR MPY16 ;DO ID*S2
0372			*
0373			* (V0+K1*ID)*ID FORMAT - 32 BITS
0374			* S X X X X X X X 8 4 2 1 . . 5 . 25 X X X X---->LSB
0375			*
0376			* SECTION DOES P2-P4 (32 BITS)
0377	F960	96 23	AK LDA A #23 ;LD P2 LS
0378	F962	06 22	LDA B #22 ;LD P2 LS MID
0379	F964	90 13	SUB A DIFF+3 ;P2LS-P4LS
0380	F966	02 12	SBC B DIFF+2
0381	F968	97 13	AL STA A DIFF+3 ;STORE P2LS-P4LS
0382	F96A	07 12	STA B DIFF+2 ;STORE P2LS, MID-P4LS, MID
0383	F96C	96 21	LDA A #21 ;LD P2 MS MID
0384	F96E	06 20	LDA B #20 ;LD P2 MS
0385	F970	92 11	SBC A DIFF+1 ;P2MS, MID-P4MS, MID
0386	F972	02 10	SBC B DIFF ;P2MS-P4MS
0387	F974	97 11	AM STA A DIFF+1 ;STORE P2MS, MID-P4MS, MID
0388	F976	07 10	STA B DIFF ;STORE P2MS-P4MS
0389			* WX IS 32 BITS NOW AT 0010 THRU 0013

SEQ	LOC	OBJ	SOURCE
0390			*
0391			* SECTION ADJUSTS 2WX FOR 16 BITS OF ACCURACY
0392			* PRIOR TO "R" CORRECTION
0393			* WX SIGNIFICANCE IS DIFF AND DIFF+1
0394			* R* & WX FORMAT IS 16 BITS
0395			* S 16K 8K 4K 2K 1K 512 256 128 64 32 16 8 4 2 1
0396			*
0397			*** WX IS POSITIVE *****
0398			*
0399			*** TEST2 SCALING OF WX: MOVE LEFT 2 PLACES
0400	F978	96 55	WXCOR LDA A FZFG ;LD F=0 FLAG
0401	F97A	27 0E	BEQ SCWX ;BRANCH IF F NOT 0
0402	F97C	4F	CLR A
0403	F97D	97 32	STA A ESTAR ;CLR ESTAR MS
0404	F97F	97 33	STA A ESTAR+1 ;CLR ESTAR LS
0405	F981	97 34	STA A UN
0406	F983	97 35	STA A UN+1
0407	F985	97 47	STA A XN ;CLR XN MS
0408	F987	97 48	STA A XN+1 ;CLR XN LS
0409	F989	39	RTS ;RETURN FROM SUBR
0410	F98A	06 01	SCWX LDA B #01 ;SET BIT COUNT FOR LOOP
0411	F98C	78 0013	SCWX1 ASL DIFF+3 ;MOVE LS LEFT 1
0412	F98F	79 0012	ROL DIFF+2 ;MOVE MID-LS LEFT 1
0413	F992	79 0011	ROL DIFF+1 ;MOVE MID-MS LEFT 1
0414	F995	79 0010	ROL DIFF ;MOVE MS LEFT 1
0415	F998	5A	DEC B ;DECR BIT COUNT
0416	F999	26 F1	BNE SCWX1 ;BRANCH FOR MORE SCALING
0417			*
0418			* MODIFIED 3-30-83
0419			* (FILTERS WX BEFORE SUBTRACTING R*) CLAMPS WX)
0420			*
0421	F99B	96 10	LDA A DIFF ;GET WX MS
0422	F99D	06 11	LDA B DIFF+1 ;GET WX LS
0423	F99F	43	COM A ;COMPLEMENT
0424	F9A0	53	COM B
0425	F9A1	81 21	CMP A #21 ;CLAMP TO PREVENT FILTER O/F
0426	F9A3	20 02	BLT NOCLMP
0427	F9A5	86 20	LDA A #20
0428	F9A7	97 32	NOCLMP STA A ESTAR ;STORE FOR FILTER
0429	F9A9	07 33	STA B ESTAR+1
0430	F9AB	39	RTS
0431			*
0432			* END OF MODIFICATION
0433			*

SEQ LOC OBJ SOURCE

```

0435 ***** FLTCL *****
0436 * FILTER CALCULATION
0437 * PERFORMS S+A/S+B FILTER ALGORITHM
0438 * DONE FOR EVERY INPUT VALUE
0439 * INPUT IS 16 BITS STORED AT 0010,0011
0440 * OPERATIONS INCLUDE :
0441 * XN1 = A*X - B*UN1 + C*U
0442 * YN1 = UN1 + XN1
0443 * CONSTANTS A,B,C ARE STORED IN PROM
0444 * X,UN1 ARE STORED IN RAM AT 0030-0031,0032-0033
0445 * UN1= E* FROM FLXCL
0446 * U IS STORED AT 0034,0035
0447 *
0448 * FORMAT FOR DATA THIS SECTION
0449 * CONSTANTS A,B,C HAVE 16 BITS AS
0450 * B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0451 * S 1 . . 5 . 25 X X X X X X X X X X
0452 *UN & UN+1 HAVE FORM
0453 * S 16K 8K 4K 2K 1K 512 X X 64 32 16 8 4 2 1
0454 *XN FORM
0455 * S 4160K X X X X X X X X X X X X X X 256
0456 *XN+1 HAS FORM
0457 * S 4160K X X X X X X X X X X X X X X 256
0458 *YN+1 HAS FORM
0459 * S 4160K X X X X X X X X X X X X X X 256
0460 *
0461 F90H ORG $F90H
0462 F90A FE F000 FLTCL LDX HADD ; GET H
0463 F90D DF 10 STX MPDM ; AND SAVE FOR MPY
0464 F90F DE 47 LDX XN ; GET XN
0465 * SPECIAL OP FOR - XN'S
0466 F9D1 96 47 LDA H XN ; LD XN MS
0467 F9D3 49 ROL A ; GET SIGN
0468 F9D4 24 01 BCC CXP ; BRANCH FOR +XN
0469 F9D6 08 INX ; INCR MS BYTE
0470 F9D7 DF 12 CXP STX MPRM ; AND SAVE FOR MPY
0471 F9D9 BD FB58 JSR MPY16 ; DO A*XN
0472 * A*XN DONE, STORE 4 BYTES OF PRODUCT
0473 * STORE A*XN (40BITS) FOR ADD
0474 F9DC DE 10 LDX PROD ; GET A*XN MS 2
0475 F9DE DF 21 STX AXMS ; STORE MS 2
0476 F9E0 DE 12 LDX PROD+2 ; STORE 2 LS BYTES
0477 F9E2 DF 23 STX AXMS+2 ; STORE 2 LS
0478 * DOING 24 BY 16 BIT MULTIPLY
0479 F9E4 FE F000 LDX HADD ; LD A
0480 F9E7 DF 10 STX MPDM ; STORE FOR MULT
0481 F9E9 96 47 LDA H XN ; LD XN
0482 F9EB 49 ROL A ; GET SIGN
0483 F9EC 86 00 FXN LDA A #$00 ; LD + MS BYTE
0484 F9EE 24 02 BCC CHXN ; BRANCH FOR +XN
0485 F9F0 86 FF CHXN LDA A #$FF ; LD - MS BYTE
0486 F9F2 06 49 CHXN LDA B XN+2 ; LD XN LS BYTE
0487 F9F4 97 12 STH H MPRM ; STORE MS FOR MULT
0488 F9F6 07 13 STH B MPRM+1 ; STORE LS

```

SEQ LOC OBJ SOURCE

```

0489 F9F8 BD FB58 JSR MPY16 ; DO LS MULTIPLY
0490 * COMBINE PARTIAL PRODUCTS OF A*XN
0491 F9FB 96 13 LDA A PROD+3 ; LD LS BYTE
0492 F9FD 97 20 STH A AXMS+4 ; STORE LS BYTE
0493 F9FF 96 12 LDA A PROD+2 ; LD LS MID
0494 FA01 D6 11 LDA B PROD+1 ; LD MS MID
0495 FA03 96 24 ADD A AXMS+3 ; ADD LS MID
0496 FA05 D9 23 ADD B AXMS+2 ; ADD MS MID
0497 FA07 97 24 STA A AXMS+3 ; STORE LS MID
0498 FA09 D7 23 STA B AXMS+2 ; STORE MS MID
0499 FA0B D6 10 LDH B PROD ; LD MS OF PROD
0500 FA0D 96 10 LDH A PROD ; LD MS OF PROD
0501 FA0F 99 22 ADD H AXMS+1 ; ADD MS
0502 FA11 D9 21 ADD B AXMS ; ADD MS OF AXN
0503 FA13 97 22 STA A AXMS+1 ; STORE MS MID
0504 FA15 D7 21 STA B AXMS ; STORE MS BYTE
0505 * HAVE DONE A*XN, NEXT OP IS B*UN1
0506 FA17 FE F0D2 LDX BADD ; GET B
0507 FA1H DF 10 STX MPDM ; SAVE FOR MPY
0508 FA1C DE 32 LDX UN1 ; GET UN1
0509 FA1E DF 12 STX NPRM ; SAVE FOR MPY
0510 FA20 BD FB58 JSR MPY16 ; DO B*UN1
0511 FA23 DE 10 LDX PROD ; SAVE B*UN1 IN LOCS 26, 27, 28, 29
0512 FA25 DF 26 STX BUN1M
0513 FA27 DE 12 LDX PROD+2
0514 FA29 DF 28 STX BUN1LM
0515 * HAVE COMPLETED B*UN1, NEXT OP IS C*U
0516 FA2B FE F0D4 LDX CADD ; GET C
0517 FA2E DF 10 STX MPDM ; SAVE FOR MPY
0518 FA30 DE 34 LDX UN ; GET UN
0519 FA32 DF 12 STX NPRM ; SAVE FOR MPY
0520 FA34 BD FB58 JSR MPY16 ; DO C*U
0521 FA37 DE 10 LDX PROD ; SAVE C*UN IN LOCS 2A, 2B, 2C, 2D
0522 FA39 DF 2H STX CUNM
0523 FA3B DE 12 LDX PROD+2
0524 FA3D DF 2C STX CUNLM
0525 * HAVE COMPLETED MULT FOR XN1, NEXT OP IS XN1 CALC
0526 * DOING A*X-B*UN1+C*N.
0527 *
0528 * 1ST OPERATION - DO C*UN-B*UN1
0529 * RESOLUTION NEEDED = 32 BITS
0530 FA3F 96 2D LDR A CUNLM+1 ; GET C*UN
0531 FA41 D6 2C LDA B CUNLM
0532 FA43 90 29 SUB A BUN1LM+1 ; SUBTRACT: C*UN-B*UN1
0533 FA45 D2 28 SBC B BUN1LM
0534 * DON'T NEED LS BYTE DUE TO SCALING 8->
0535 FA47 97 2D STA A CMB+3 ; STORE LS (ALSO SCALES)
0536 FA49 D7 2C STA B CMB+2 ; STORE LS MID
0537 FA4B 96 28 LDH A CUNM+1 ; LD C*UN MS MID
0538 FA4D D6 2H LDA B CUNM ; LD C*UN MS
0539 FA4F 92 27 SBC A BUN1M+1 ; DO C*UN MS MID
0540 FA51 D2 26 SBC B BUN1M ; DO C*UN MS
0541 FA53 97 28 STA A CMB+1 ; STORE CUN-BUN1 LS MID
0542 FA55 D7 2H STA B CMB ; STORE CUN-BUN1 MS MID
0543 * C*UN - B*UN1 DONE IN ADDR 2H->2D

```

SEQ LOC OBJ SOURCE

```

0544      * DO A*AN+(C*UN-B*UN1)
0545      *DOING SUM 40 BITS LS OF AXN =LS OF CMB
0546      *
0547      * FIRST MAKE CUN-BUN1 SAME LENGTH AS AX
0548      *
0549      0000 CMB1 EQU $50 ;SIGN OF CMB
0550 FA37 36 2n LDA A CMB ;LD CMB MS
0551 FA39 49 ROL A ;GET SIGN
0552 FA3A 86 00 LDA A #$00 ;SET + SIGN
0553 FA5C 24 02 BCC SPCMB ;BRANCH FOR +CMB
0554 FA5E 86 FF LDA A #$FF ;SET - SIGN
0555 FA60 97 50 SPCMB STA A CMBT ;STORE SIGN BYTE
0556      *SIGN BYTE READY DO ADD
0557      *
0558 FA62 96 25 LDA A AXMS+4 ;LD MS BYTE CUN-BUN1
0559 FA64 98 20 ADD A CMB+3 ;ADD LS BYTES
0560 FA66 97 25 STA A AXMS+4 ;TEMP STORE XN1 LS
0561 FA68 96 24 LDA A AXMS+3 ;LD A*XN LS
0562 FA6A 06 23 LDA B AXMS+2 ;LD A*XN LS MID
0563 FA6C 99 20 ADD A CMB+2 ;ADD AX +CU-BU1 LS
0564 FA6E 09 26 ADD B CMB+1
0565 FA70 97 29 STA A XN1T+3 ;STORE XN1 LS TEMP
0566 FA72 07 28 STA B XN1T+2
0567 FA74 96 22 LDA A AXMS+1 ;LD A*X MS MID
0568 FA76 06 21 LDA B AXMS ;LS A*XN MS
0569 FA78 99 2A ADD A CMB ;ADD MS BYTES
0570 FA7A 09 50 ADD B CMBT ;ADD SIGN CMB
0571 FA7C 97 27 STAN1 STA A XN1T+1 ;STORE XN1 MS MID
0572 FA7E 07 26 STA B XN1T ;STORE MS TEMP
0573      * XN1 DONE NEED TO SCALE TO MATCH UN1 & STORAGE
0574      * TO XN POSITION(SHIFT <- 2)
0575 FA80 86 02 LDA A #$02 ;LD SHIFT COUNT
0576 FA82 78 0025 CXSCL ROL AXMS+4 ;SCALE XN1 LS
0577 FA85 79 0029 ROL XN1T+3 ;SCALE XN1
0578 FA88 79 0028 ROL XN1T+2 ; IN
0579 FA8B 79 0027 ROL XN1T+1 ; TEMP
0580 FA8E 79 0026 ROL XN1T ;LOCAL
0581 FA91 4A DEC A ;DECR SHIFT COUNT
0582 FA92 26 EE BNE CXSCL ;BRANCH FOR MORE SCALE
0583      *READY FOR YN+1
0584      * STORE UN & XN VALUE FOR NEXT CALC
0585 FA94 0E 26 LDX XN1T ;LD XN1 MS + MID
0586 FA96 96 28 LDA H XN1T+2 ;LD LS MID=LS AFTER SCALE
0587 FA98 0F 47 STA XN ;STORE XN1 AS XN NEXT CALC
0588 FA9A 97 49 STA H XN+2 ;STORE XN1 LS AS XN
0589 FA9C 0F 27 STX XN1T+1 ;STORE XN1 SCALING
0590 FA9E 97 29 STA H XN1T+3 ;STORE XN1 LS SCALED
0591 FAA0 96 33 LDA H UN1+1 ;LD LS UN+1
0592 FAA2 06 32 LDA B UN1 ;LD MS UN+1
0593 FAA4 97 35 STA A UN+1 ;STORE UN1 AS UN NEXT CALC
0594 FAA6 07 34 STA B UN ;STORE UN1 AS UN MS
0595      *DOING YN+1=XN+1 + UN+1
0596 FAAB 55 ROL B ;ROLL MS FOR SIGN
0597 FAAD 06 32 LDA B UN1 ;RESTORE UN1
0598 FAAB 24 00 BCC F0N1 ;BRANCH FOR + UN1

```


SEQ	LOC	OBJ	SOURCE
0609	FH00	98 29	ADD H XN11+1
0600	FH00	09 28	ADD B XN11+2
0601			*DON'T NEED LS OF YN+1
0602	FH01	07 13	STH B ERLT
0603	FH03	86 FF	LDA H #FF
0604	FH05	99 27	ADD H XN11+1
0605	FH07	97 12	STH H ERMT
0606	FH09	39	RIS
0607	FH0A	98 29	PUN1 ADD A XN11+3
0608	FH0C	09 28	ADD B XN11+2
0609			*DON'T NEED LS OF YN+1
0610	FH0E	07 13	STH B ERLT
0611	FH00	86 00	LDA H #00
0612	FH02	99 27	ADD H XN11+1
0613	FH04	97 12	STH H ERMT
0614	FH06	39	RIS

REV. VI. 1

ADDR	LOC	OPR	SOURCE

0616			***** ERROR *****
0617			* ERROR WORD OUTPUT SUBROUTINE
0618			* OUTPUTS 12 BITS OF ERROR WORD TO MPH
0619			* TAKES 16 BITS & ADJUSTS FORMAT AS FOLLOWS:
0620			*ERROR WORD
0621			*B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0622			* TWO BYTES OUTPUT TO LATCHES
0623			* ADDR B07 B06 B05 B04 B03 B02 B01 B00
0624			* ERHI X X B15 B14 B13 B12 B11 B10 STORE TO C004
0625			* ERLO X X B09 B08 B07 B06 B05 B04 STORE TO B000
0626			*
0627			*****
0628		005H	ES01R EQU \$3H ;E* MS VALUE FOR OUTPUT
0629		005B	ES01L EQU \$3B ;E* LS VALUE FOR OUTPUT
0630			*
0631		FHD7	ORG \$FHD7
0632	FHD7	96 41	ERRST LDA H NEWF ;LD NEW FREQ WND
0633	FHD9	98 40	EUR H ULDF ;JUMP TO OLD FREQ
0634	FHD6	27 0H	BEW NFCHG ;BRANCH NO FREQ CHANGE
0635	FHD0	0E 0001	LDX #0001 ;LC MAX TIME INTRVL
0636	FHE4	96 41	SP10 LDA H NEWF ;RELOAD NEW FREQ
0637	FHE2	97 40	STH H ULDF ;STORE AS OLD FREQ
0638	FHE1	DF 43	STNT STX INCHG ;STORE MAX INTRVL
0639	FHE6	39	RIS ;RETURN
0640	FHE7	DE 43	NFCHG LDX INCHG ;LOAD TIME INT
0641	FHE9	05	DEX ;DECR TIME
0642	FHEH	26 F8	BNE STNT ;BRANCH IF MORE TIME
0643	FHE0	0E 0001	LDX #0001 ;SET TIME INTRVL TO MIN
0644	FHEF	DF 43	STX INCHG ;STORE MIN TIME
0645	FHF1	96 12	LDA H ERMT ;LOAD ERROR MS 8
0646	FHF3	D6 13	LDA B ERLO ;LOAD ERROR LS 8
0647	FHF3	7E F830	JMP BYPASS ;BYPASS THE GP 3 ROUTINE
0648			*
0649			*ALTERED CODE
0650			*PROVIDES FOR A VARIABLE GAIN FOR WND
0651			*
0652	FHF6	86 FF	LDA H #FF ;GET GAIN FROM TABLE
0653	FHF8	97 00	STH H 00
0654	FHFC	DE 00	LDX 0000
0655	FHFE	H6 00	LDA H 0X
0656	FB00	E6 01	LDA B 1X
0657	FB02	97 10	STH H \$10 ;STORE FOR MULTIPLY
0658	FB04	D7 11	STB B \$11
0659	FB06	BD F838	JSR MPY16 ;GAIN X ESTAR
0660	FB07	0E 0003	LDX #0003 ;SCALE DOWN PRODUCT
0661	FB0C	77 0010	LSR \$10 ;SCALE
0662	FB0F	76 0011	ROR \$11
0663	FB12	76 0012	ROR \$12
0664	FB15	76 0013	ROR \$13
0665	FB18	05	DEX
0666	FB19	26 F1	BNE SCALE
0667	FB1B	DE 12	LDX \$12 ;GET RESULT
0668	FB1D	DF 10	STX \$10 ;STORE FOR ADD
0669			*

```

SEQ    LOC    OBJ    SOURCE

0670 FB1F 36 FD          LDA H    #RADD          ;LD R MS OF ADDR
0671 FB21 37 00          STA H    #00          ;CORRECT ADDR
0672 FB23 0E 00          LDX      #0000          ;LD R ADDRESS
0673 FB25 H6 00          LDA H    0,X          ;LD R MS
0674 FB27 E6 01          LDA B    1,X          ;LD R LS
0675 FB29 ED FB8D        JSR      ADD16          ;DO R-(+WX)
0676                      *
0677                      * END OF MODIFICATION 3-30-83
0678                      *
0679                      ** E* FOR FLICL NOW EXISTS
0680                      * SCALING 5 16K 8K 4K 2K 1K ----- 16 8 4 2 1
0681 FB2C 36 12          LDA H    SUMM          ;LD HNS MS(R-WX)
0682 FB2E 06 13          LDA B    SUML          ;LD HNS LS(R-WX)
0683                      *FORMAT E*OUT FOR LATCH
0684 FB30 CE 0002        BYPASS LDX      #0002          ;LD SHIFT COUNT
0685 FB32 47            CES          HSR H          ;SCALE MS
0686 FB34 36            ROR B          ;ROTATE LS RIGHT 1
0687 FB36 09            DEX          ;DECR LOOP CNT
0688 FB38 26 FB        BNE      CES          ;BRANCH FOR MORE SCALING
0689                      * MS BYTE NOW CORRECT
0690                      * FINISH FORMAT FOR ERL1 TO LATCH
0691 FB38 57            HSR B          ;ROTATE LS RIGHT 1
0692 FB39 57            HSR B          ;ROTATE LS RIGHT 1
0693 FB3A 0F            SEI          ;SET MASK OF IRQ
0694 FB3B 97 3H        STA H    ESUM          ;STORE MS TO RAM
0695 FB3D 07 3B        STA B    ESUTL         ;STORE LS TO RAM
0696 FB3F 0E            CLI          ;CLEAR INT MASK
0697 FB40 39            RIB
0698                      *
0699                      * ID RECTIFIER ROUTINE FOR OVERLOAD PROTECTION 5-3-83
0700                      *
0701 FB41 06 05        RECTFY LDA B    IDAD          ;GET DIRECT CURRENT
0702 FB43 2H 01        BPL      POS          ;BRANCH IF POSITIVE
0703 FB45 50          NEG B          ;NEGATE IF NEGATIVE
0704 FB46 07 33        POS          STA B    ESTAR+1 ;STORE IDAD
0705 FB48 7F 0032      CLR      ESTAR        ;CLEAR ESTAR MS BYTE
0706 FB4B CE 0004      LDX      #0004        ;SCALE RECTIFIED ID FOR FILTER
0707 FB4E 78 0033      GAIN1   ASL      ESTAR+1
0708 FB51 79 0032      ROL      ESTAR
0709 FB54 05          DEX
0710 FB55 26 F7        BNE      GAIN1
0711 FB57 39          RTS
0712                      *
0713                      *END OF RECTIFIER ROUTINE 5-3-83
0714                      *

```


SEQ	LUC	OBJ	SOURCE
0758			***** ADD16 *****
0759			* 16 BIT ADD SUBROUTINE
0760			* ASSUMES ADDEND IS IN A,B REGISTERS
0761			* ADDER LOCATED IN 0010,0011 PRIOR TO JSR
0762			* SUM IS STORED IN 0012,0013
0763			* A = N MSB
0764			* B = N LSB
0765			* 0010 = A MSB
0766			* 0011 = A LSB
0767			* AFTER ADD,
0768			* 0012 = SUM MSB
0769			* 0013 = SUM LSB
0770			*
0771	FB80	09 11	ADD16 ADC B ADDL ; ADD TWO LS BYTES
0772	FB8F	99 10	ADC A ADDM ; ADD TWO MS BYTES
0773	FB91	07 13	STA B SUML ; STORE LS BYTE
0774	FB93	97 12	STA A SUMM ; STORE MS BYTE
0775	FB95	39	RTS

SEQ LOC OBJ SOURCE

```

0777          ***** E+ OUTPUT ROUTINE *****
0778          * WRITES TO E+ LATCH IN RESPONSE TO INTERRUPT
0779          * INTERRUPT FROM MPA HAPPENS APPROX. 200MICROSEC
0780          * PRIOR TO V+ INPUT INTO MPA
0781          *
0782 FB36 96 3H EQU1    LDA A  ES01N    ;LD E+ MS FOR LATCH
0783 FB36 06 3B        LDA B  ES01L    ;LD E+ LS FOR LATCH
0784 FB3A B7 8000      STA H  LMS      ;STORE MS 6 TO LATCH
0785 FB3D F7 0000      STA B  LLS      ;STORE LS 6 TO LATCH
0786 FBA0 3B          RTI              ;RETURN FROM INT
0787          END

```

0787 LINES ASSEMBLED, LOC = FBA1, 0000 ERRORS DETECTED.

SYMBOL	VALUE	ATTR	LOCN	LINE
IRQV	FFFF	84	2882	
SWIV	FFFF	84	2886	
NMIV	FFFF	84	2884	
RSTV	FFFF	84	2880	
RADD	FCD0	84	28C6	
BADD	FCD2	84	28CF	
CADD	FCD4	84	28D8	
FREQ	0001	84	28E1	
IQAD	0002	84	28EH	
VQAD	0003	84	28F3	
VQAD	0004	84	28FC	
IDAD	0005	84	2805	
ERMT	0012	84	280E	
ERLT	0013	84	2817	
LLS	0000	84	2820	
LMS	8000	84	2829	
MPRM	0012	84	2832	
MPRL	0013	84	283B	
MPDM	0010	84	2844	
MPDL	0011	84	284D	
ADDL	0011	84	2856	
ADDH	0010	84	285F	
SUML	0013	84	2868	
SUMH	0012	84	2871	
RADD	00FD	84	287H	
KADD	00FE	84	2883	
PROD	0010	84	288C	
DIFF	0010	84	2895	
ESTAR	0032	84	289E	
AKMS	0021	84	28A7	
BUN1M	0026	84	28B0	
BUN1LM	0028	84	28B9	
CUNM	002A	84	28C2	
CUNLM	002C	84	28CB	
UN1	0032	84	28D4	
XN	0047	84	28DD	
UN	0034	84	28E6	
CMB	002A	84	28EF	
XN1T	0026	84	28F8	
OLDF	0040	84	2C01	
NEWF	0041	84	2C0A	
TMCNT	0043	84	2C13	
ISTART	F800	84	2C1C	
PA5	F803	84	2C25	
PA51	F808	84	2C2E	
PBS	F81B	84	2C37	
EVS	F82F	84	2C40	
INP	F85F	84	2C49	
INPUT	F840	84	2C52	
RECTFY	F841	84	2C5B	
FLTCL	F90A	84	2C64	
ERRST	FAD7	84	2C6D	
FZFG	0055	84	2C76	
CNFI	F874	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
CNFI1	F877	84	2C88	
GT6	F87F	84	2C91	
ADLOOP	F892	84	2C9A	
FLXCL	F8D0	84	2CA3	
AB	F8D6	84	2CAC	
PIQ	F8E0	84	2CB5	
MPY16	FB58	84	2CBE	
AC	F8E4	84	2CC7	
SCP1	F8E9	84	2CD0	
CVD	F903	84	2CD9	
ADD16	FB80	84	2CE2	
AD	F906	84	2CEB	
SIG	F90D	84	2CF4	
AE	F916	84	2CFD	
AF	F924	84	2D06	
PID	F92E	84	2D0F	
SCP3	F937	84	2D18	
CVD	F94F	84	2D21	
SID	F959	84	2D2H	
AK	F960	84	2D33	
HL	F968	84	2D3C	
AM	F974	84	2D45	
WACOR	F978	84	2D4E	
SCWA	F98A	84	2D57	
SCWA1	F98C	84	2D60	
NOCLMP	F9A7	84	2D69	
CKP	F9D7	84	2D72	
PKN	F9EC	84	2D7B	
CAN	F9F2	84	2D84	
CH8T	0050	84	2D8D	
SPOMB	FA60	84	2D96	
STXN1	FA7C	84	2D9F	
CKSCL	FA82	84	2DAH	
PUN1	FABA	84	2DB1	
ES0TM	003A	84	2DBH	
ES0TL	003B	84	2DC3	
NFCHG	FAE7	84	2DCC	
SFTC	FAE0	84	2DD5	
STCNT	FAE4	84	2DDE	
BYPASS	FB30	84	2DE7	
SCALE	FB0C	84	2DF0	
CES	FB33	84	2DF9	
POS	FB46	84	2E02	
GHIN1	FB4E	84	2E0B	
MPY2	0014	84	2E14	
SMY2	FB6F	84	2E1D	
SMY1	FB67	84	2E26	
SMY3	FB71	84	2E2F	
SMY4	FB88	84	2E38	
EOUT	FB96	U4	2E41	

END ASM V1.1


```

0001 REM PROGRAM GENERATES VALUES FOR V PEAK NOM / V PEAK
0002 REM THIS IS TO NORMALIZE AFFECT OF B+ VARIATIONS
0003 REM ON THE OUTPUT VOLTAGE.
0004 REM
0005 LINE= 0
0010 PRINT #7, "    NAM V PEAK TABLE"
0020 PRINT #7, "    ORG $E000 "
0030 PRINT #7, "* "
0040 PRINT #7, "*    SCALING OF VNOM/VPEAK * 256 "
0050 PRINT #7, "*    GIVES SAME SCALING AS SCALE FACTOR"
0060 FOR B=0 TO 80 STEP 1
0070   B1=(128/80)*256
0080   B1=INT(B1)
0090 GOSUB 500
0100 NEXT B
0110 FOR B=81 TO 150 STEP 1
0120   B1=(128/B)*256
0130   B1=INT(B1)
0140 GOSUB 500
0150 NEXT B
0160 FOR B=151 TO 255 STEP 1
0170   B1=(128/150)*256
0180   B1=INT(B1)
0190 GOSUB 500
0200 NEXT B
0210 GOTO 800
0500 PRINT #4, "VP=", B, "128/V PEAK * 256="; B1
0501   K=B
0502 IF B<80 THEN K=80
0503 IF B>150 THEN K=150
0510 PRINT #7, "VP"; B, "FDB   "; B1, "; FACTOR="; 128/K
0520 RETURN
0800 PRINT #7, "    END"
0810 END

```

SEQ	LOC	OBJ	SOURCE
0001			NAM V PEAK TABLE
0002		E500	ORG #E500
0003			*
0004			* SCALING OF VNOM/VPEAK * 256
0005			* GIVES SAME SCALING AS SCALE FACTOR
0006	E500	02 00	VP0 FDB 512 ; FACTOR =2
0007	E502	02 00	VP1 FDB 512 ; FACTOR =2
0008	E504	02 00	VP2 FDB 512 ; FACTOR =2
0009	E506	02 00	VP3 FDB 512 ; FACTOR =2
0010	E508	02 00	VP4 FDB 512 ; FACTOR =2
0011	E50A	02 00	VP5 FDB 512 ; FACTOR =2
0012	E50C	02 00	VP6 FDB 512 ; FACTOR =2
0013	E50E	02 00	VP7 FDB 512 ; FACTOR =2
0014	E510	02 00	VP8 FDB 512 ; FACTOR =2
0015	E512	02 00	VP9 FDB 512 ; FACTOR =2
0016	E514	02 00	VP10 FDB 512 ; FACTOR =2
0017	E516	02 00	VP11 FDB 512 ; FACTOR =2
0018	E518	02 00	VP12 FDB 512 ; FACTOR =2
0019	E51A	02 00	VP13 FDB 512 ; FACTOR =2
0020	E51C	02 00	VP14 FDB 512 ; FACTOR =2
0021	E51E	02 00	VP15 FDB 512 ; FACTOR =2
0022	E520	02 00	VP16 FDB 512 ; FACTOR =2
0023	E522	02 00	VP17 FDB 512 ; FACTOR =2
0024	E524	02 00	VP18 FDB 512 ; FACTOR =2
0025	E526	02 00	VP19 FDB 512 ; FACTOR =2
0026	E528	02 00	VP20 FDB 512 ; FACTOR =2
0027	E52A	02 00	VP21 FDB 512 ; FACTOR =2
0028	E52C	02 00	VP22 FDB 512 ; FACTOR =2
0029	E52E	02 00	VP23 FDB 512 ; FACTOR =2
0030	E530	02 00	VP24 FDB 512 ; FACTOR =2
0031	E532	02 00	VP25 FDB 512 ; FACTOR =2
0032	E534	02 00	VP26 FDB 512 ; FACTOR =2
0033	E536	02 00	VP27 FDB 512 ; FACTOR =2
0034	E538	02 00	VP28 FDB 512 ; FACTOR =2
0035	E53A	02 00	VP29 FDB 512 ; FACTOR =2
0036	E53C	02 00	VP30 FDB 512 ; FACTOR =2
0037	E53E	02 00	VP31 FDB 512 ; FACTOR =2
0038	E540	02 00	VP32 FDB 512 ; FACTOR =2
0039	E542	02 00	VP33 FDB 512 ; FACTOR =2
0040	E544	02 00	VP34 FDB 512 ; FACTOR =2
0041	E546	02 00	VP35 FDB 512 ; FACTOR =2
0042	E548	02 00	VP36 FDB 512 ; FACTOR =2
0043	E54A	02 00	VP37 FDB 512 ; FACTOR =2
0044	E54C	02 00	VP38 FDB 512 ; FACTOR =2
0045	E54E	02 00	VP39 FDB 512 ; FACTOR =2
0046	E550	02 00	VP40 FDB 512 ; FACTOR =2
0047	E552	02 00	VP41 FDB 512 ; FACTOR =2
0048	E554	02 00	VP42 FDB 512 ; FACTOR =2
0049	E556	02 00	VP43 FDB 512 ; FACTOR =2
0050	E558	02 00	VP44 FDB 512 ; FACTOR =2
0051	E55A	02 00	VP45 FDB 512 ; FACTOR =2
0052	E55C	02 00	VP46 FDB 512 ; FACTOR =2
0053	E55E	02 00	VP47 FDB 512 ; FACTOR =2
0054	E560	02 00	VP48 FDB 512 ; FACTOR =2
0055	E562	02 00	VP49 FDB 512 ; FACTOR =2

SEQ	LOC	OBJ	SOURCE				
0056	E564	02 00	VP50	FDB	512	, FACTOR	=2
0057	E566	02 00	VP51	FDB	512	, FACTOR	=2
0058	E568	02 00	VP52	FDB	512	, FACTOR	=2
0059	E56A	02 00	VP53	FDB	512	, FACTOR	=2
0060	E56C	02 00	VP54	FDB	512	, FACTOR	=2
0061	E56E	02 00	VP55	FDB	512	, FACTOR	=2
0062	E570	02 00	VP56	FDB	512	, FACTOR	=2
0063	E572	02 00	VP57	FDB	512	, FACTOR	=2
0064	E574	02 00	VP58	FDB	512	, FACTOR	=2
0065	E576	02 00	VP59	FDB	512	, FACTOR	=2
0066	E578	02 00	VP60	FDB	512	, FACTOR	=2
0067	E57A	02 00	VP61	FDB	512	, FACTOR	=2
0068	E57C	02 00	VP62	FDB	512	, FACTOR	=2
0069	E57E	02 00	VP63	FDB	512	, FACTOR	=2
0070	E580	02 00	VP64	FDB	512	, FACTOR	=2
0071	E582	02 00	VP65	FDB	512	, FACTOR	=2
0072	E584	02 00	VP66	FDB	512	, FACTOR	=2
0073	E586	02 00	VP67	FDB	512	, FACTOR	=2
0074	E588	02 00	VP68	FDB	512	, FACTOR	=2
0075	E58A	02 00	VP69	FDB	512	, FACTOR	=2
0076	E58C	02 00	VP70	FDB	512	, FACTOR	=2
0077	E58E	02 00	VP71	FDB	512	, FACTOR	=2
0078	E590	02 00	VP72	FDB	512	, FACTOR	=2
0079	E592	02 00	VP73	FDB	512	, FACTOR	=2
0080	E594	02 00	VP74	FDB	512	, FACTOR	=2
0081	E596	02 00	VP75	FDB	512	, FACTOR	=2
0082	E598	02 00	VP76	FDB	512	, FACTOR	=2
0083	E59A	02 00	VP77	FDB	512	, FACTOR	=2
0084	E59C	02 00	VP78	FDB	512	, FACTOR	=2
0085	E59E	02 00	VP79	FDB	512	, FACTOR	=2
0086	E5A0	02 00	VP80	FDB	512	, FACTOR	=2
0087	E5A2	01 F9	VP81	FDB	505	, FACTOR	=1. 97530864
0088	E5A4	01 F3	VP82	FDB	499	, FACTOR	=1. 95121951
0089	E5A6	01 ED	VP83	FDB	493	, FACTOR	=1. 92771084
0090	E5A8	01 E7	VP84	FDB	487	, FACTOR	=1. 9047619
0091	E5AA	01 E1	VP85	FDB	481	, FACTOR	=1. 88235294
0092	E5AC	01 DC	VP86	FDB	476	, FACTOR	=1. 86046511
0093	E5AE	01 D6	VP87	FDB	470	, FACTOR	=1. 83908045
0094	E5B0	01 D1	VP88	FDB	465	, FACTOR	=1. 81818181
0095	E5B2	01 CC	VP89	FDB	460	, FACTOR	=1. 7977528
0096	E5B4	01 C7	VP90	FDB	455	, FACTOR	=1. 77777777
0097	E5B6	01 C2	VP91	FDB	450	, FACTOR	=1. 75824175
0098	E5B8	01 BD	VP92	FDB	445	, FACTOR	=1. 73913043
0099	E5BA	01 B8	VP93	FDB	440	, FACTOR	=1. 7204301
0100	E5BC	01 B3	VP94	FDB	435	, FACTOR	=1. 70212765
0101	E5BE	01 AF	VP95	FDB	431	, FACTOR	=1. 68421052
0102	E5C0	01 AA	VP96	FDB	426	, FACTOR	=1. 66666666
0103	E5C2	01 A6	VP97	FDB	422	, FACTOR	=1. 64948453
0104	E5C4	01 A1	VP98	FDB	417	, FACTOR	=1. 63265306
0105	E5C6	01 9D	VP99	FDB	413	, FACTOR	=1. 61616161
0106	E5C8	01 99	VP100	FDB	409	, FACTOR	=1. 6
0107	E5CA	01 95	VP101	FDB	405	, FACTOR	=1. 58415841
0108	E5CC	01 91	VP102	FDB	401	, FACTOR	=1. 56862745
0109	E5CE	01 8D	VP103	FDB	397	, FACTOR	=1. 55339805
0110	E5D0	01 89	VP104	FDB	393	, FACTOR	=1. 53846153

SEQ	LOC	OBJ	SOURCE			
0111	E5D2	01 86	VP105	FDB	390	. FACTOR =1. 52380952
0112	E5D4	01 82	VP106	FDB	386	. FACTOR =1. 50943396
0113	E5D6	01 7E	VP107	FDB	382	. FACTOR =1. 4953271
0114	E5D8	01 78	VP108	FDB	379	. FACTOR =1. 48148148
0115	E5DA	01 77	VP109	FDB	375	. FACTOR =1. 4678899
0116	E5DC	01 74	VP110	FDB	372	. FACTOR =1. 45454545
0117	E5DE	01 71	VP111	FDB	369	. FACTOR =1. 44144144
0118	E5E0	01 6D	VP112	FDB	365	. FACTOR =1. 42857142
0119	E5E2	01 6A	VP113	FDB	362	. FACTOR =1. 4159292
0120	E5E4	01 67	VP114	FDB	359	. FACTOR =1. 40350877
0121	E5E6	01 64	VP115	FDB	356	. FACTOR =1. 39130434
0122	E5E8	01 61	VP116	FDB	353	. FACTOR =1. 37931034
0123	E5EA	01 5E	VP117	FDB	350	. FACTOR =1. 36752136
0124	E5EC	01 58	VP118	FDB	347	. FACTOR =1. 3559322
0125	E5EE	01 58	VP119	FDB	344	. FACTOR =1. 34453781
0126	E5F0	01 55	VP120	FDB	341	. FACTOR =1. 33333333
0127	E5F2	01 52	VP121	FDB	338	. FACTOR =1. 32231404
0128	E5F4	01 4F	VP122	FDB	335	. FACTOR =1. 3114754
0129	E5F6	01 4D	VP123	FDB	333	. FACTOR =1. 300813
0130	E5F8	01 4A	VP124	FDB	330	. FACTOR =1. 29032258
0131	E5FA	01 47	VP125	FDB	327	. FACTOR =1. 28
0132	E5FC	01 45	VP126	FDB	325	. FACTOR =1. 26984126
0133	E5FE	01 42	VP127	FDB	322	. FACTOR =1. 25984251
0134	E600	01 40	VP128	FDB	320	. FACTOR =1. 25
0135	E602	01 3D	VP129	FDB	317	. FACTOR =1. 24031007
0136	E604	01 3B	VP130	FDB	315	. FACTOR =1. 23076923
0137	E606	01 38	VP131	FDB	312	. FACTOR =1. 22137404
0138	E608	01 36	VP132	FDB	310	. FACTOR =1. 21212121
0139	E60A	01 33	VP133	FDB	307	. FACTOR =1. 20300751
0140	E60C	01 31	VP134	FDB	305	. FACTOR =1. 19402985
0141	E60E	01 2F	VP135	FDB	303	. FACTOR =1. 18518518
0142	E610	01 2D	VP136	FDB	301	. FACTOR =1. 17647058
0143	E612	01 2A	VP137	FDB	298	. FACTOR =1. 16788321
0144	E614	01 28	VP138	FDB	296	. FACTOR =1. 15942028
0145	E616	01 26	VP139	FDB	294	. FACTOR =1. 15107913
0146	E618	01 24	VP140	FDB	292	. FACTOR =1. 14285714
0147	E61A	01 22	VP141	FDB	290	. FACTOR =1. 13475177
0148	E61C	01 20	VP142	FDB	288	. FACTOR =1. 12676056
0149	E61E	01 1E	VP143	FDB	286	. FACTOR =1. 11888111
0150	E620	01 1C	VP144	FDB	284	. FACTOR =1. 11111111
0151	E622	01 1A	VP145	FDB	282	. FACTOR =1. 10344827
0152	E624	01 18	VP146	FDB	280	. FACTOR =1. 09589041
0153	E626	01 16	VP147	FDB	278	. FACTOR =1. 08843537
0154	E628	01 14	VP148	FDB	276	. FACTOR =1. 08108108
0155	E62A	01 12	VP149	FDB	274	. FACTOR =1. 0738255
0156	E62C	01 11	VP150	FDB	273	. FACTOR =1. 06666666
0157	E62E	01 0F	VP151	FDB	271	. FACTOR =1. 05960264
0158	E630	01 0D	VP152	FDB	269	. FACTOR =1. 05263157
0159	E632	01 0B	VP153	FDB	267	. FACTOR =1. 04575163
0160	E634	01 09	VP154	FDB	265	. FACTOR =1. 03896103
0161	E636	01 08	VP155	FDB	264	. FACTOR =1. 03225806
0162	E638	01 06	VP156	FDB	262	. FACTOR =1. 02564102
0163	E63A	01 04	VP157	FDB	260	. FACTOR =1. 01910828
0164	E63C	01 03	VP158	FDB	259	. FACTOR =1. 01265822
0165	E63E	01 01	VP159	FDB	257	. FACTOR =1. 0062893

SEQ	LOC	OBJ	SOURCE			
0166	E640	01 00	VP160	FDB	256	FACTOR =1
0167	E642	30 FE	VP161	FDB	254	FACTOR =0.993788819
0168	E644	00 FC	VP162	FDB	252	FACTOR =0.98765432
0169	E646	00 FB	VP163	FDB	251	FACTOR =0.981595092
0170	E648	00 F9	VP164	FDB	249	FACTOR =0.975609756
0171	E64A	00 F8	VP165	FDB	248	FACTOR =0.969696969
0172	E64C	00 F6	VP166	FDB	246	FACTOR =0.963855421
0173	E64E	00 F5	VP167	FDB	245	FACTOR =0.958083832
0174	E650	00 F3	VP168	FDB	243	FACTOR =0.952380952
0175	E652	00 F2	VP169	FDB	242	FACTOR =0.946745562
0176	E654	00 F0	VP170	FDB	240	FACTOR =0.94117647
0177	E656	00 EF	VP171	FDB	239	FACTOR =0.935672514
0178	E658	00 EE	VP172	FDB	238	FACTOR =0.930232558
0179	E65A	00 EC	VP173	FDB	236	FACTOR =0.924855491
0180	E65C	00 EB	VP174	FDB	235	FACTOR =0.919540229
0181	E65E	00 EA	VP175	FDB	234	FACTOR =0.914285714
0182	E660	00 E8	VP176	FDB	232	FACTOR =0.909090909
0183	E662	00 E7	VP177	FDB	231	FACTOR =0.903954802
0184	E664	00 E6	VP178	FDB	230	FACTOR =0.898876404
0185	E666	00 E4	VP179	FDB	228	FACTOR =0.893854748
0186	E668	00 E3	VP180	FDB	227	FACTOR =0.888888888
0187	E66A	00 E2	VP181	FDB	226	FACTOR =0.8839779
0188	E66C	00 E1	VP182	FDB	225	FACTOR =0.879120879
0189	E66E	00 DF	VP183	FDB	223	FACTOR =0.874316939
0190	E670	00 DE	VP184	FDB	222	FACTOR =0.869565217
0191	E672	00 DD	VP185	FDB	221	FACTOR =0.864864864
0192	E674	00 DC	VP186	FDB	220	FACTOR =0.860215053
0193	E676	00 DB	VP187	FDB	219	FACTOR =0.855614973
0194	E678	00 D9	VP188	FDB	217	FACTOR =0.851063829
0195	E67A	00 D8	VP189	FDB	216	FACTOR =0.846560846
0196	E67C	00 D7	VP190	FDB	215	FACTOR =0.842105263
0197	E67E	00 D6	VP191	FDB	214	FACTOR =0.837696335
0198	E680	00 D5	VP192	FDB	213	FACTOR =0.833333333
0199	E682	00 D4	VP193	FDB	212	FACTOR =0.829015544
0200	E684	00 D3	VP194	FDB	211	FACTOR =0.824742268
0201	E686	00 D2	VP195	FDB	210	FACTOR =0.82051282
0202	E688	00 D0	VP196	FDB	208	FACTOR =0.81632653
0203	E68A	00 CF	VP197	FDB	207	FACTOR =0.812182741
0204	E68C	00 CE	VP198	FDB	206	FACTOR =0.808080808
0205	E68E	00 CD	VP199	FDB	205	FACTOR =0.8040201
0206	E690	00 CC	VP200	FDB	204	FACTOR =0.8
0207	E692	00 CC	VP201	FDB	204	FACTOR =0.8
0208	E694	00 CC	VP202	FDB	204	FACTOR =0.8
0209	E696	00 CC	VP203	FDB	204	FACTOR =0.8
0210	E698	00 CC	VP204	FDB	204	FACTOR =0.8
0211	E69A	00 CC	VP205	FDB	204	FACTOR =0.8
0212	E69C	00 CC	VP206	FDB	204	FACTOR =0.8
0213	E69E	00 CC	VP207	FDB	204	FACTOR =0.8
0214	E6A0	00 CC	VP208	FDB	204	FACTOR =0.8
0215	E6A2	00 CC	VP209	FDB	204	FACTOR =0.8
0216	E6A4	00 CC	VP210	FDB	204	FACTOR =0.8
0217	E6A6	00 CC	VP211	FDB	204	FACTOR =0.8
0218	E6A8	00 CC	VP212	FDB	204	FACTOR =0.8
0219	E6AA	00 CC	VP213	FDB	204	FACTOR =0.8
0220	E6AC	00 CC	VP214	FDB	204	FACTOR =0.8

SEQ	LOC	OBJ	SOURCE
0221	E68E	00 CC	VP215 FDB 204 ; FACTOR =0.8
0222	E690	00 CC	VP216 FDB 204 ; FACTOR =0.8
0223	E692	00 CC	VP217 FDB 204 ; FACTOR =0.8
0224	E694	00 CC	VP218 FDB 204 ; FACTOR =0.8
0225	E696	00 CC	VP219 FDB 204 ; FACTOR =0.8
0226	E698	00 CC	VP220 FDB 204 ; FACTOR =0.8
0227	E69A	00 CC	VP221 FDB 204 ; FACTOR =0.8
0228	E69C	00 CC	VP222 FDB 204 ; FACTOR =0.8
0229	E69E	00 CC	VP223 FDB 204 ; FACTOR =0.8
0230	E6A0	00 CC	VP224 FDB 204 ; FACTOR =0.8
0231	E6A2	00 CC	VP225 FDB 204 ; FACTOR =0.8
0232	E6A4	00 CC	VP226 FDB 204 ; FACTOR =0.8
0233	E6A6	00 CC	VP227 FDB 204 ; FACTOR =0.8
0234	E6A8	00 CC	VP228 FDB 204 ; FACTOR =0.8
0235	E6AA	00 CC	VP229 FDB 204 ; FACTOR =0.8
0236	E6AC	00 CC	VP230 FDB 204 ; FACTOR =0.8
0237	E6AE	00 CC	VP231 FDB 204 ; FACTOR =0.8
0238	E6B0	00 CC	VP232 FDB 204 ; FACTOR =0.8
0239	E6B2	00 CC	VP233 FDB 204 ; FACTOR =0.8
0240	E6B4	00 CC	VP234 FDB 204 ; FACTOR =0.8
0241	E6B6	00 CC	VP235 FDB 204 ; FACTOR =0.8
0242	E6B8	00 CC	VP236 FDB 204 ; FACTOR =0.8
0243	E6BA	00 CC	VP237 FDB 204 ; FACTOR =0.8
0244	E6BC	00 CC	VP238 FDB 204 ; FACTOR =0.8
0245	E6BE	00 CC	VP239 FDB 204 ; FACTOR =0.8
0246	E6E0	00 CC	VP240 FDB 204 ; FACTOR =0.8
0247	E6E2	00 CC	VP241 FDB 204 ; FACTOR =0.8
0248	E6E4	00 CC	VP242 FDB 204 ; FACTOR =0.8
0249	E6E6	00 CC	VP243 FDB 204 ; FACTOR =0.8
0250	E6E8	00 CC	VP244 FDB 204 ; FACTOR =0.8
0251	E6EA	00 CC	VP245 FDB 204 ; FACTOR =0.8
0252	E6EC	00 CC	VP246 FDB 204 ; FACTOR =0.8
0253	E6EE	00 CC	VP247 FDB 204 ; FACTOR =0.8
0254	E6F0	00 CC	VP248 FDB 204 ; FACTOR =0.8
0255	E6F2	00 CC	VP249 FDB 204 ; FACTOR =0.8
0256	E6F4	00 CC	VP250 FDB 204 ; FACTOR =0.8
0257	E6F6	00 CC	VP251 FDB 204 ; FACTOR =0.8
0258	E6F8	00 CC	VP252 FDB 204 ; FACTOR =0.8
0259	E6FA	00 CC	VP253 FDB 204 ; FACTOR =0.8
0260	E6FC	00 CC	VP254 FDB 204 ; FACTOR =0.8
0261	E6FE	00 CC	VP255 FDB 204 ; FACTOR =0.8
0262	E700	00 CC	VP256 FDB 204 ; FACTOR =0.8
0263			END

0263 LINES ASSEMBLED, LOC = E702, 0000 ERRORS DETECTED.

END ASM V1.1

```

0005 REM *****FTGN0*****
0006 REM *** VALUES FROM VAC MEASURED - UNLOADED ***
0010 REM PROGRAM CALCULATES & LISTS VALUES FOR FTBL
0020 REM THIS FOR NEW MODU SCHEME (EVERY P. P. -B+)
0030 REM F      = FREQ OF INTEREST
0040 REM N      = NO. OF PULSE PERIODS IN 60 DEGREES
0050 REM D(1)   = DOMINANT PULSE LENGTH - TPOUM
0060 REM C(1)   = 2ND COMPL PULSE LENGTH- TPCUM
0070 REM M1     = MAX SCALE FACTOR
0080 REM M2     = MIN SCALE FACTOR
0090 REM T      = PULSE PERIOD LENGTH   - TPF
0100 REM L      = LENGTH OF 60 DEG SEG
0110 REM B(F)   = NO. OF P. P. IN 60 DEG - NPP
0120 REM
0125 REM "26 ON LINE 770 IS SHOOT THROUGH TIME"
0130 REM ESTABLISH NPP FOR EACH "F" (ALWAYS ODD)
0131 DIM B(60)
0132 INPUT "DISK OUTPUT ? YES=1, NO=0", Q1
0133 IF Q1=0 GOTO 1400
0140 B(6)=35
0150 B(7)=31
0160 B(8)=27
0170 B(9)=25
0180 B(10)=21
0190 B(11)=19
0200 B(12)=19
0210 B(13)=17
0220 B(14)=15
0230 B(15)=15
0240 B(16)=15
0250 B(17)=13
0260 B(18)=13
0270 FOR A1=19 TO 24
0280 B(A1)=11
0290 NEXT A1
0300 FOR A1=25 TO 29
0310 B(A1)=9
0320 NEXT A1
0330 FOR A1=30 TO 41
0340 B(A1)=7
0350 NEXT A1
0360 FOR A1=42 TO 55
0370 B(A1)=5
0380 NEXT A1
0390 FOR A1=56 TO 60
0400 B(A1)=3
0410 NEXT A1
0420 REM SET UP TOP OF DISK FILE TO BE 6800 FORMAT
0421 LINE= 0
0422 DIGITS= 0
0430 PRINT #7, "    NAM FTBL"
0431 PRINT #4, "F", TAB(5), "VOPEN", TAB(20), "NPP", TAB(25), "TPP",
0432 PRINT #4, TAB(30), "L-60", TAB(40), "SFMAX"
0433 PRINT #7, "*"
0434 PRINT #7, "*GENERATED BY FTGN0*"
0435 GOTO 1200
0440 PRINT #7, "    ORG $E880"
0450 REM NO. OF P. P. IS ESTABLISHED FOR EACH "F"
0460 REM NEXT DO DOMINANT AND COMPLEMENTARY
0470 REM PULSE WIDTHS FOR EACH P. P. IN "F"
0480 P1=3.14159
0490 DIM D(60), C(60), E(60), F(60)
0500 GOTO 1400
0510 IF Q1=0 GOTO 1080

```

```

0520 L1=1E+06/6/F
0530 T1=L1/N
0540 T=INT(T1+0.5)
0550 L=N+T
0600 FOR K=1 TO (N+1)/2 STEP 1
0610 K1=1/(2*P1+F)
0620 K2=(60/N)*(K-1)
0630 K3=(60/N)+K
0640 K7=(K2+60)*2*P1/360
0650 K8=(K3+60)*2*P1/360
0660 K9=(K2+120)*2*P1/360
0670 G1=(K3+120)*2*P1/360
0680 E(K)=K1*(COS(K7)-COS(K8))*1E+06+0.5
0690 F(K)=K1*(COS(K9)-COS(G1))*1E+06+0.5
0700 REM OLD TP=VOPEN/B+NOM= NEW TP
0710 D(K)=INT(E(K)+0/160)
0720 C(K)=INT(F(K)+0/160)
0730 NEXT K
0740 REM HAVE COMPLETED PULSE WIDTHS FOR "F". DO SCALE FACTORS
0750 K4=(N+1)/2
0760 K5=D(K4)
0770 M=(T-26)/K5
0790 M1=INT(M+256+0.5)
0791 IF K5>(T-26) THEN M1=INT((T-26)*256/K5+0.5)
0792 IF M1>2047 THEN M1=2047
0795 PRINT #4, F, TAB(5); D, TAB(20); N, TAB(25); T, TAB(30); L, TAB(40); M1
0800 REM MIN SF = 0.50 ALWAYS
0810 M2=128
0820 REM OUTPUT VALUES TO TABLE ON DISK
0830 REM FTBL HAS FORM :
0840 REM N = NPP
0850 REM T = TPP
0860 REM L = LENGTH OF 60 DEG (USEC)
0870 REM SFMAX
0880 REM SFMIN
0890 REM TPD0M #1 (USEC)
0900 REM TPD0M #1 (USEC)
0910 REM TPD0M #2
0920 REM TPD0M #2
0930 REM "
0940 REM "
0950 REM TPD0M (N+1)/2
0960 REM TPD0M (N+1)/2
0970 PRINT #7, "+FREQUENCY TABLE FOR "; F; "HZ."
0980 PRINT #7, "*"
0990 PRINT #7, "F0"; F; " FCB "; N; " ; NO. -PULSE PERIODS IN 60 DEG SEG"
1000 PRINT #7, " FDB "; T; " ; LENGTH OF PULSE PERIOD (USEC)"
1010 PRINT #7, " FDB "; L; " ; LENGTH OF 60 DEGREE SEG (USEC)"
1020 PRINT #7, " FDB "; M1; " ; SFMAX"
1030 PRINT #7, " FDB "; M2; " ; SFMIN"
1040 PRINT #7, "+PULSE WIDTHS"
1050 FOR I=1 TO (N+1)/2
1060 PRINT #7, " ", "FDB", D(I); " ; D0M"
1070 PRINT #7, " ", "FDB", C(I)
1080 NEXT I
1081 RETURN
1085 PRINT #4, "F="; F, "O="; O
1086 RETURN
1100 IF O1=0 GOTO 1800
1101 PRINT #7, " END"
1110 GOTO 1800
1200 REM SECTION WRITES FREQ INDEX CHARACTERS FOR DISK FILE
1210 PRINT #7, "*****"
1220 PRINT #7, "+FREQUENCY TABLES *"
1230 PRINT #7, "*****"

```



```
1250 PRINT #7, "*"
1255 PRINT #7, "INDEX", "FDB", "FRQ6"
1260 FOR I=7 TO 60
1270 PRINT #7, " ", "FDB", "FRQ", I
1280 NEXT I
1290 GOTO 440
1400 FOR F=6 TO 60
1410 O=1.9*F
1420 GOSUB 510
1430 NEXT F
1440 GOTO 1100
1800 END
```

F=6	O=11.4
F=7	O=13.3
F=8	O=15.2
F=9	O=17.1
F=10	O=19
F=11	O=20.9
F=12	O=22.8
F=13	O=24.7
F=14	O=26.6
F=15	O=28.5
F=16	O=30.4
F=17	O=32.3
F=18	O=34.2
F=19	O=36.1
F=20	O=38
F=21	O=39.9
F=22	O=41.8
F=23	O=43.7
F=24	O=45.6
F=25	O=47.5
F=26	O=49.4
F=27	O=51.3
F=28	O=53.2
F=29	O=55.1
F=30	O=57
F=31	O=58.9
F=32	O=60.8
F=33	O=62.7
F=34	O=64.6
F=35	O=66.5
F=36	O=68.4
F=37	O=70.3
F=38	O=72.2
F=39	O=74.1
F=40	O=76
F=41	O=77.9
F=42	O=79.8
F=43	O=81.7
F=44	O=83.6
F=45	O=85.5
F=46	O=87.4
F=47	O=89.3
F=48	O=91.2
F=49	O=93.1
F=50	O=95
F=51	O=96.9
F=52	O=98.8
F=53	O=100.7
F=54	O=102.6
F=55	O=104.5
F=56	O=106.4
F=57	O=108.3
F=58	O=110.2
F=59	O=112.1
F=60	O=114

```

0005 REM ***** VSR5G *****
0006 REM ***** THIS VERSION GENERATES LINEAR RELATIONSHIP
0007 REM FOR ALL VALUES. ACTUAL TABLE WAS MODIFIED BY HAND
0010 REM PROGRAM GENERATES REFERENCE VALUES
0020 REM FOR V* TABLE USED BY MPA TO DO
0030 REM OVERLOAD DETECTION.
0040 REM VALUES COME FROM MEASUREMENTS MADE
0050 REM ON MACHINE AT 117 V B+
0060 REM CREATE LEAD IN ON DISK
0070 PRINT #7, "  NAM  VSREF"
0080 PRINT #7, "  ORG  $E400"
0085 PRINT #4, "F", "V* REF"
0090 FOR F=0 TO 10
0100   V=0
0110   GOSUB 1006
0120 NEXT F
0130 FOR F=11 TO 15
0140   V=(144-0)*(F-10)/5+0
0150   GOSUB 1000
0160 NEXT F
0170 FOR F=16 TO 18
0180   V=(336-144)*(F-15)/3+144
0190   GOSUB 1000
0200 NEXT F
0210 FOR F=19 TO 20
0220   V=(544-336)*(F-18)/2+336
0230   GOSUB 1000
0240 NEXT F
0250   F=21
0260   V=688
0270   GOSUB 1000
0290 FOR F=22 TO 24
0300   V=(1264-688)*(F-21)/3+688
0310   GOSUB 1000
0320 NEXT F
0330 FOR F=25 TO 26
0340   V=(1584-1264)*(F-24)/2+1264
0350   GOSUB 1000
0360 NEXT F
0370 FOR F=27 TO 29
0380   V=(1888-1584)*(F-26)/3+1584
0390   GOSUB 1000
0400 NEXT F
0410 FOR F=30 TO 31
0420   V=(2032-1888)*(F-29)/2+1888
0430   GOSUB 1000
0440 NEXT F
0450 FOR F=32 TO 37
0460   V=(2480-2032)*(F-31)/6+2032
0470   GOSUB 1000
0480 NEXT F
0490 FOR F=38 TO 55
0500   V=(3024-2480)*(F-37)/23+2480
0510   GOSUB 1000
0520 NEXT F
0530 FOR F=56 TO 60
0540   V=3583+(56-F)*(3583-3375)/(60-56)
0550   GOSUB 1006
0560 NEXT F
0570 GOTO 1500
1000 REM SUBROUTINE WRITES EACH LINE OF VSREF
1002 REM V* IN MPA AFTER SHIFTING IS 2 TIMES E*
1003 REM IN MPB AT FAF1
1005   V=1.25*V

```

1007 V1=16*V1+15
1008 IF FC11 GOTO 1010
1009 IF V1>V+8 THEN V1=V1-16
1010 PRINT #7, "VS"; F; " FDB "; V1; " , V*="; V
1020 PRINT #4, F, V
1030 RETURN
1500 PRINT #7, " END"
1510 END

```

0005 REM ***** R*GN *****
0010 REM PROGRAM MAKES TABLE OF R* VALUES
0015 REM VALUES OF R* ESTABLISHED BY MEASUREMENTS
0016 REM ON ACTUAL MACHINE
0020 PRINT #7, "  NAM  R* VALUES"
0030 PRINT #7, "  ORG $F008"
0035 PRINT #7, "R* VALUES ESTABLISHED BY MEASUREMENTS ON MACHINE"
0040 DIM R(60)
0050 R(6)=32
0055 R(7)=29
0060 R(8)=27
0065 R(9)=24
0070 R(10)=22
0075 R(11)=19
0080 R(12)=16
0085 R(13)=14
0090 R(14)=11
0095 R(15)=9
0100 R(16)=8
0105 R(17)=6
0110 R(18)=5
0115 R(19)=3
0120 R(20)=2
0125 R(21)=2
0130 R(22)=9
0135 R(23)=3
0140 R(24)=4
0145 R(25)=12
0150 R(26)=11
0155 R(27)=12
0160 R(28)=14
0165 R(29)=15
0170 R(30)=23
0175 R(31)=25
0180 R(32)=28
0185 R(33)=30
0190 R(34)=33
0195 R(35)=34
0200 R(36)=36
0205 R(37)=38
0210 R(38)=39
0215 R(39)=42
0220 R(40)=44
0225 R(41)=51
0230 R(42)=64
0235 R(43)=69
0240 R(44)=71
0245 R(45)=72
0250 R(46)=69
0255 R(47)=66
0260 R(48)=64
0265 R(49)=62
0270 R(50)=59
0275 R(51)=57
0280 R(52)=54
0285 R(53)=52
0290 R(54)=49
0295 R(55)=46
0300 R(56)=94
0305 R(57)=93
0310 R(58)=96
0315 R(59)=79
0320 R(60)=86
0330 FOR K=6 TO 60

```

0350 PRINT #4, "F="; K, "R="; R(K)

0360 NEXT K

0370 PRINT #7, " END"

0380 END

```

0005 REM **** K1GN ****
0010 REM PROGRAM GENERATES K1(F) TABLE
0020 REM F=FREQUENCY
0030 REM K=DESIRED CONSTANT(2*PI*F*L1)
0040 REM L1=SERIES INDUCTANCE
0042 REM SCALING ON K HAS 12BITS AFTER DECIMAL
0050 INPUT "L1",L1
0060 F1=3.1415927
0062 GOSUB 200
0070 FOR F=6 TO 60 STEP 1
0080 K=2*F1*F*L1
0090 PRINT #4,F,K
0092 PRINT #7,"K",F,"FDB ";INT(K*8192);" ";K=")K
0100 NEXT F
0110 PRINT #7," END"
0115 GOTO 800
0200 PRINT #4,"F","K1"
0210 PRINT #7," NAM K1 VALUES"
0220 PRINT #7," ORG $FE06"
0221 PRINT #7,"* TABLE GENERATED USING L1 ="L1
0230 RETURN
0800 END

```


1000
1000

1000

1000

1000

1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000

1000

1000

```

0010 DIGITS= 0
0015 LINE= 0
0050 REM PROGRAM CALCULATES FILTER CONSTANTS
0055 REM FORM OF EQUATION IS  $A+X-B+U1+U+U$ 
0100 INPUT "W1=H, W2=L, D1="; W1, W2, D
0130 A=EXP(-W2+D)
0135 B1=(W1-W2)/((D+W2+W2)
0136 B2=(1-W2+D-A)
0137 C1=1-(1+W2+D)*A
0140 B=B1+B2
0150 C=B1+C1
0160 PRINT "A="; A, "B="; B, "C="; C
0165 PRINT #4, "A="; A, "B="; B, "C="; C
0170 K=1/16384
0180 A1=A/K
0190 A2=B/K
0200 A3=C/K
0210 PRINT A1, A2, A3
0211 PRINT #4, A1, A2, A3
0220 X=INT(A1)
0221 PRINT X
0230 GOSUB 500
0240 PRINT #4, "A="; M4, M3, M2, M1
0250 X=INT(A2)
0251 PRINT X
0260 GOSUB 500
0270 PRINT #4, "B="; M4, M3, M2, M1
0280 X=INT(A3)
0281 PRINT X
0290 GOSUB 500
0300 PRINT #4, "C="; M4, M3, M2, M1
0310 GOTO 1000
0500 M4=INT(X/4096)
0510 X=INT(X-M4*4096)
0520 M3=INT(X/256)
0530 X=INT(X-M3*256)
0540 M2=INT(X/16)
0550 X=INT(X-M2*16)
0560 M1=X
0570 RETURN
1000 END

```

AD-A140 299

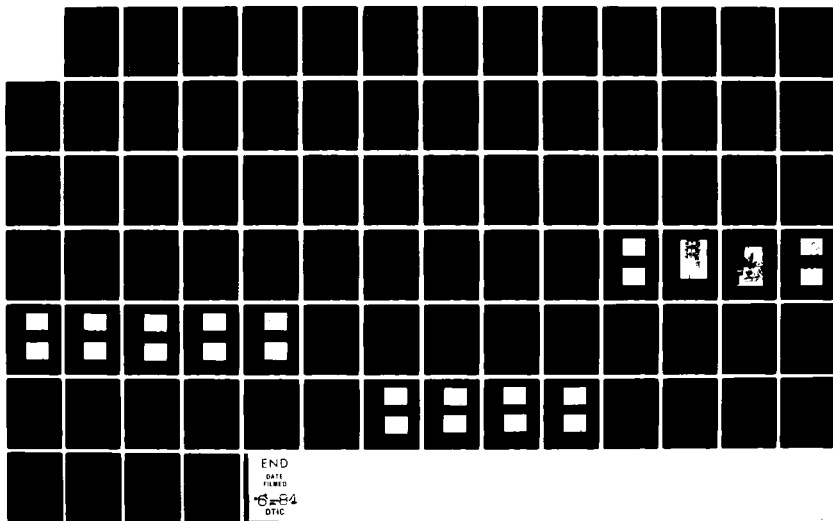
ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE
OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83
DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

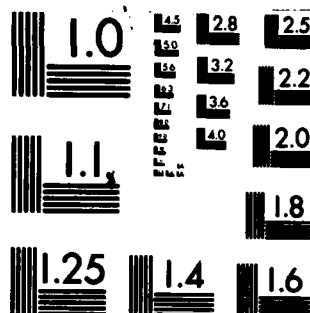
4/4

UNCLASSIFIED

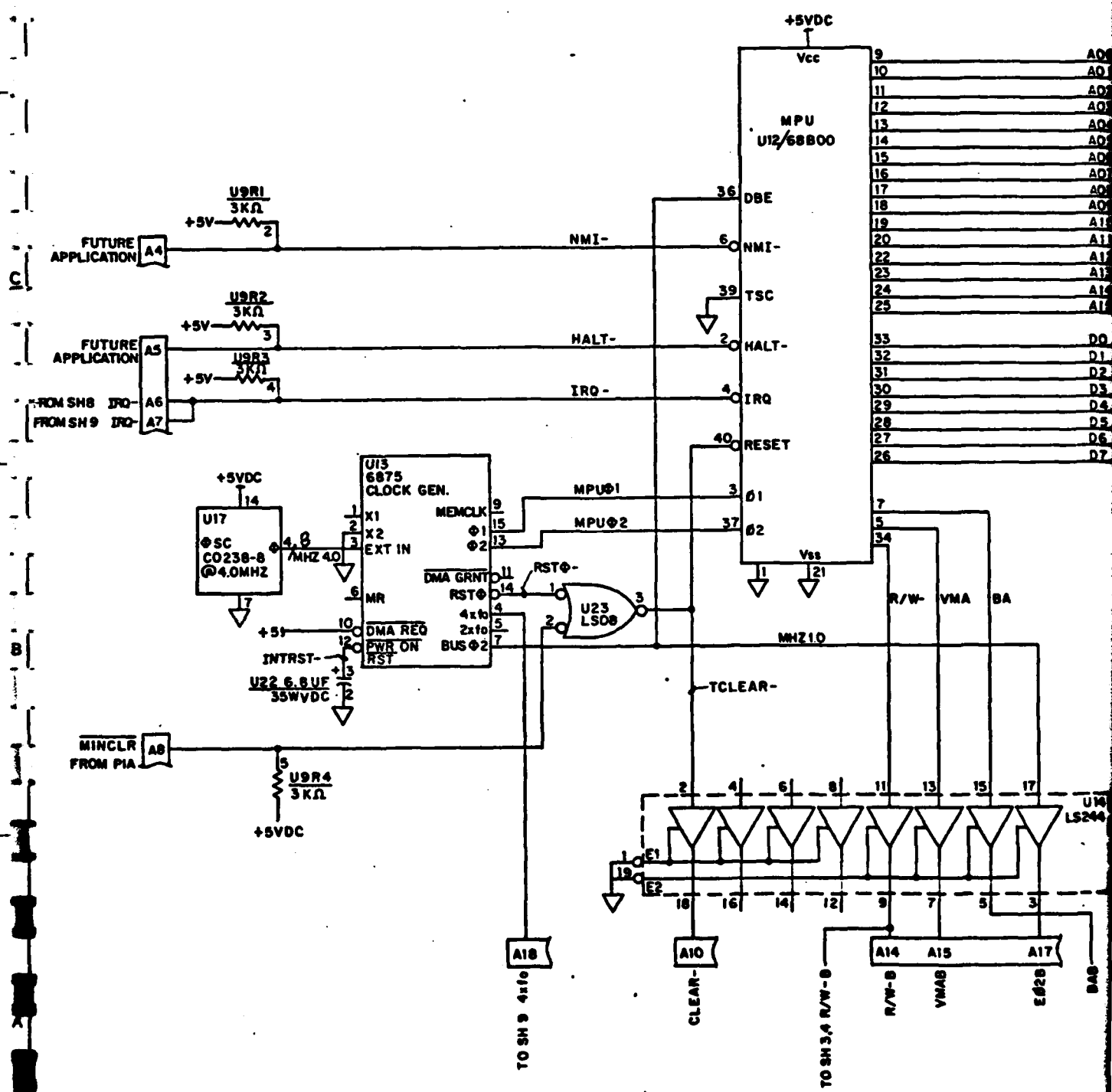
F/G 9/3

NL



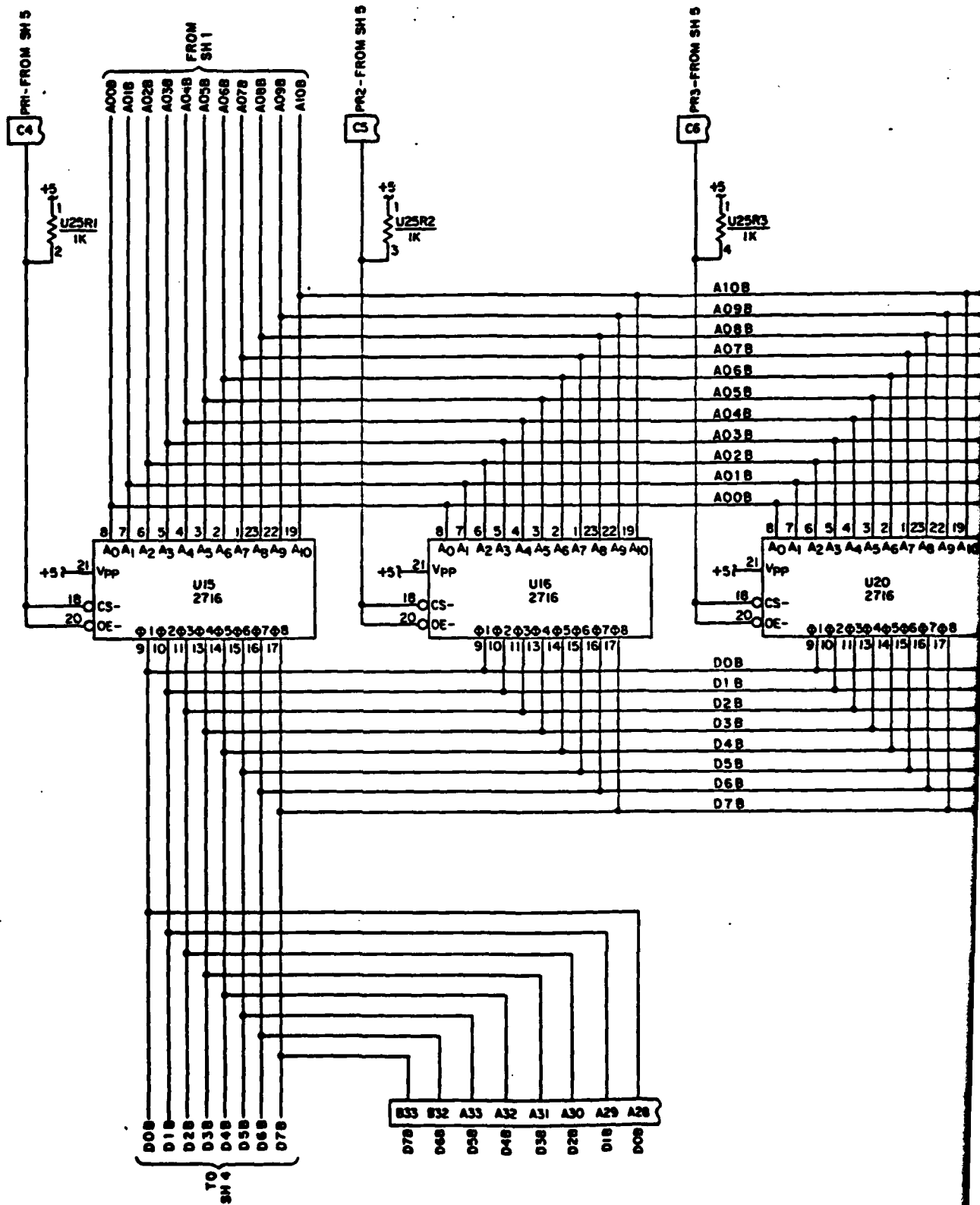


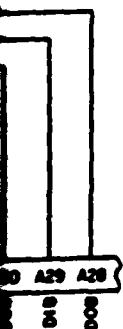
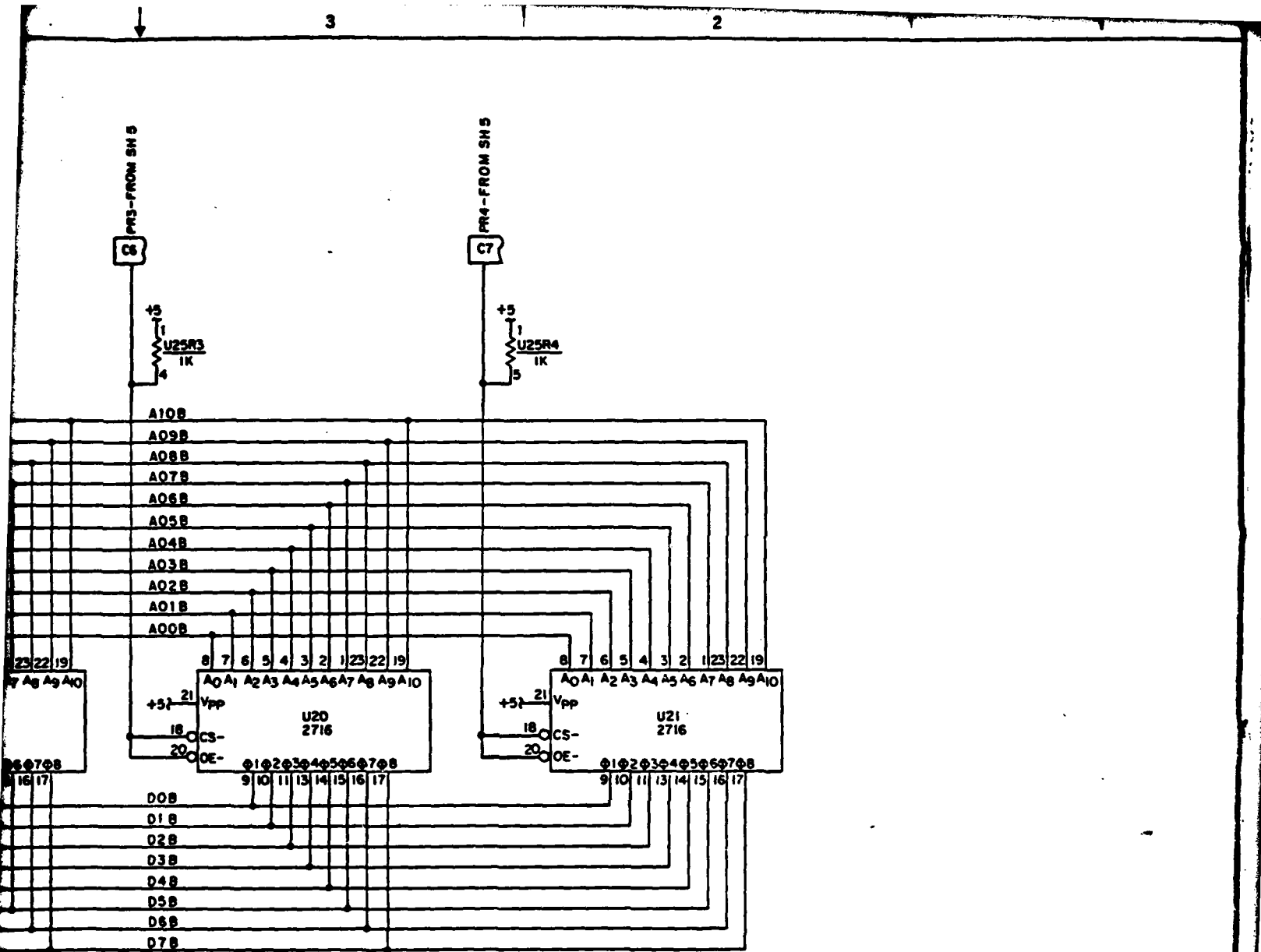
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A





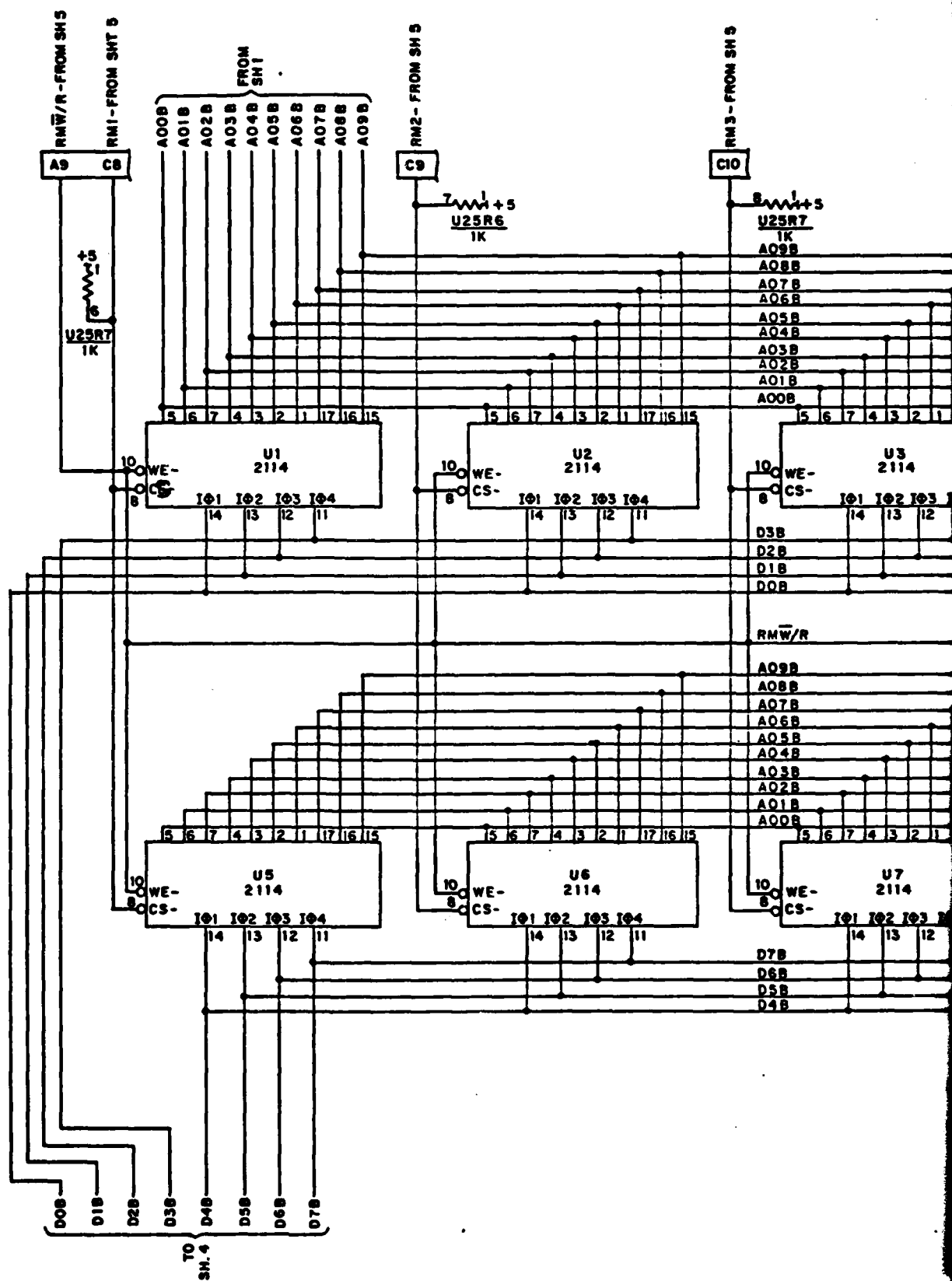
SHEET REVISION STATUS																				A1 MOD MPU					
20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	SHEET NO	2/18/81	MPU & ADDR/CTRL BUFFERS			
																				LATEST	0 PENNONI	MICROPROCESSOR LOGIC DIAGRAM			
																				REV. LTR					
																						B			
																				NEXT ASSY	USED ON	SCALE	1 - 1 - 10000	(SHEET)	OF
																				APPLICATION					

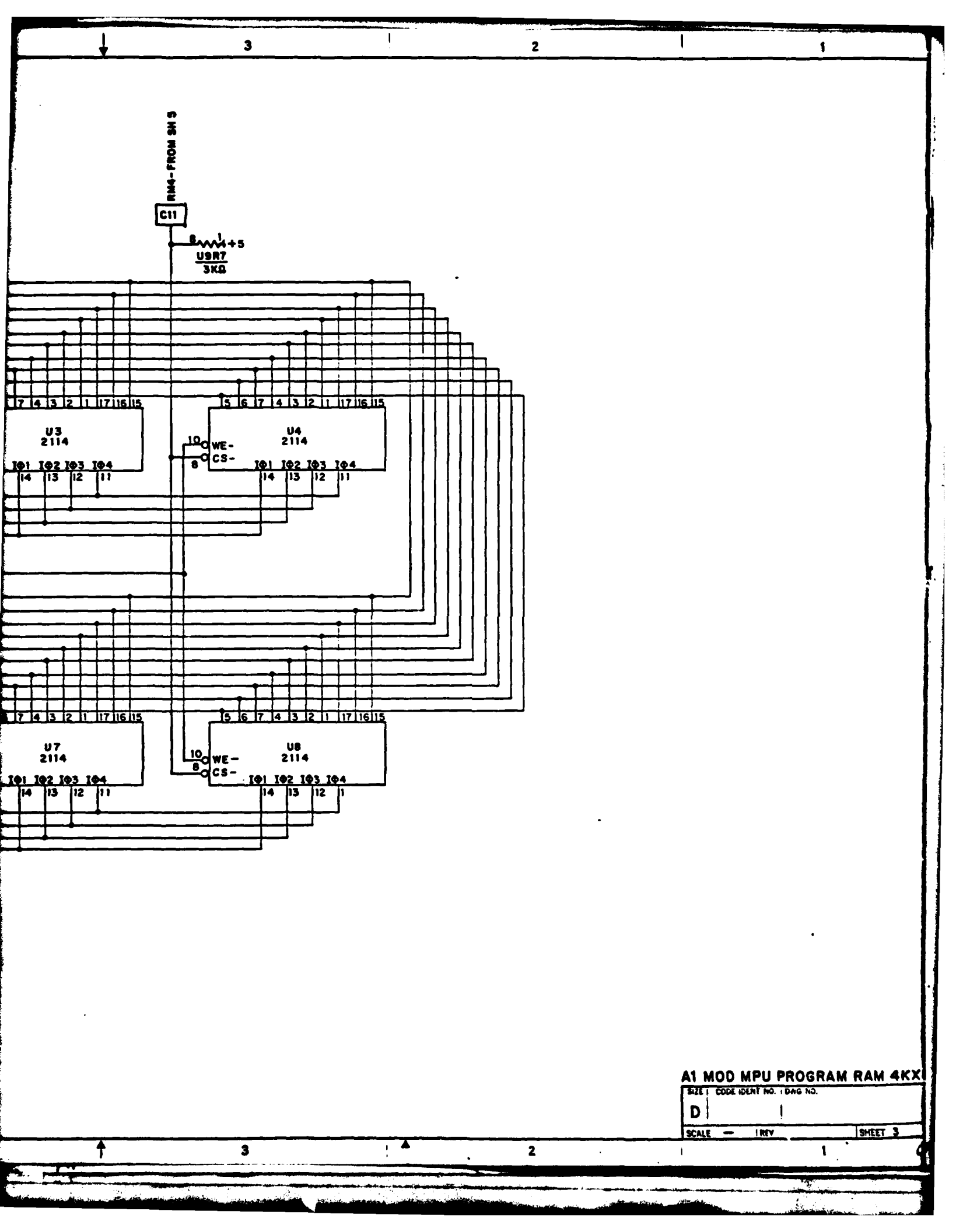




A1 MOD MPU PROGRAM PRM 8KX1

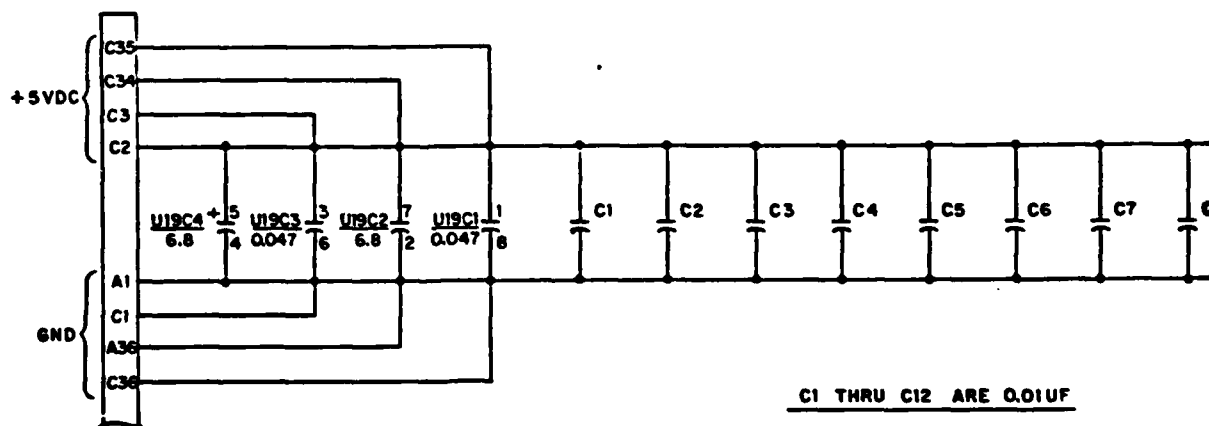
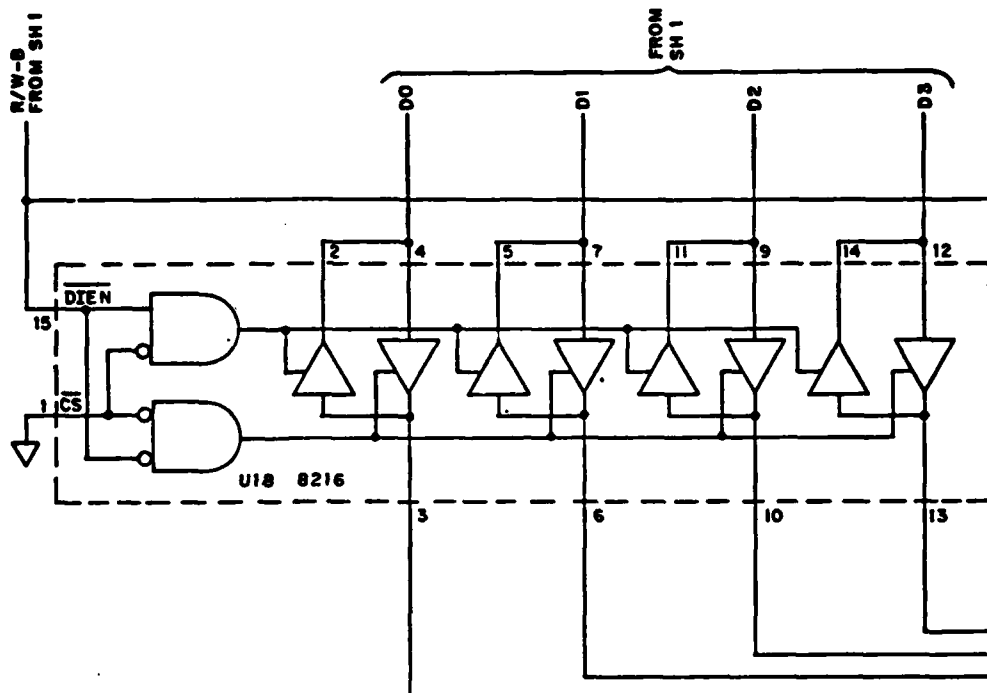
SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE	REV	SHEET 2

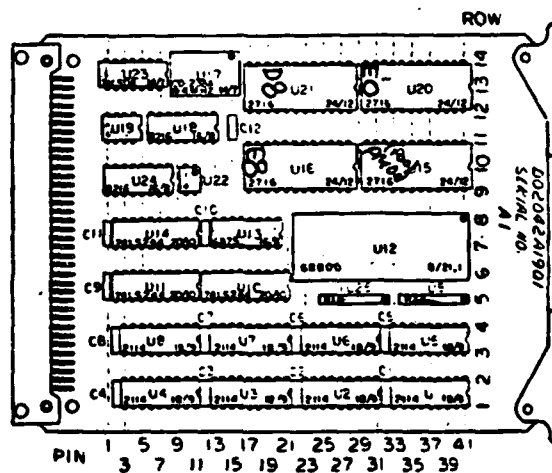
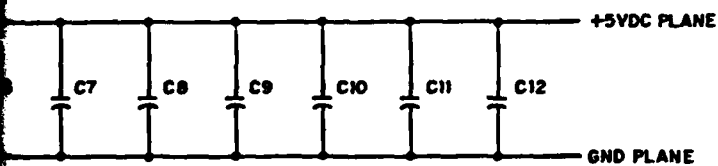
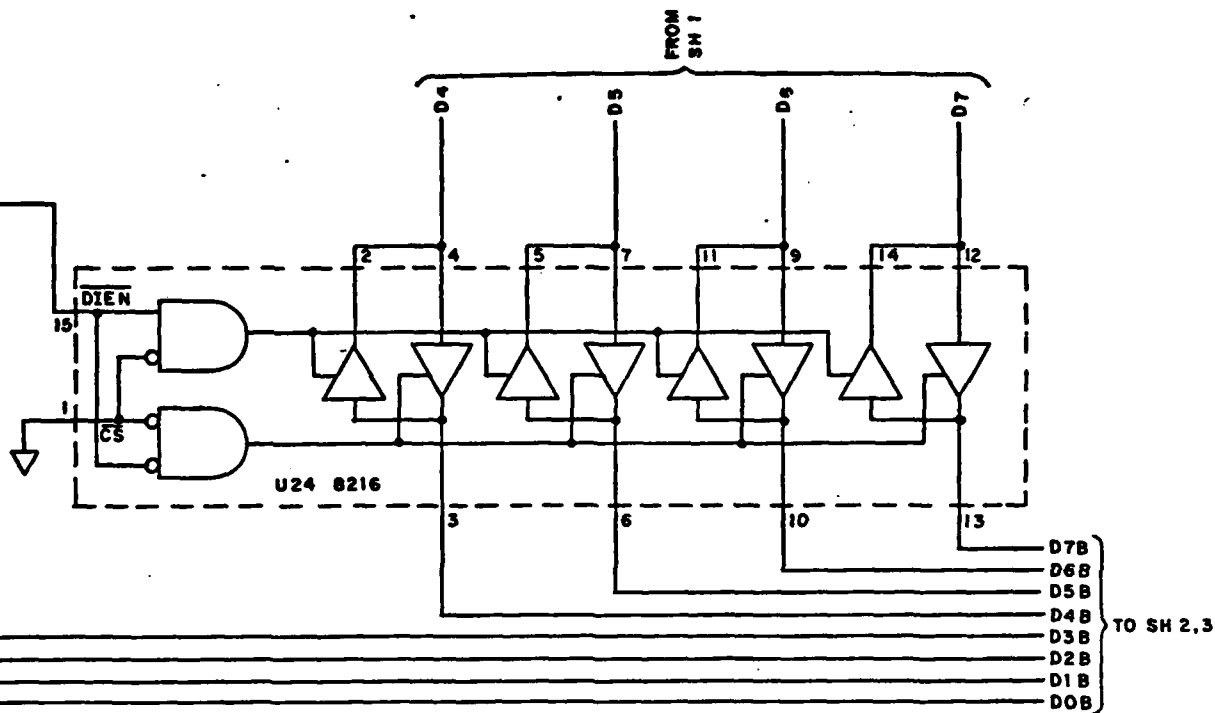




A1 MOD MPU PROGRAM RAM 4KX1

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE	REV	SHEET 3





A1 MOD MPU

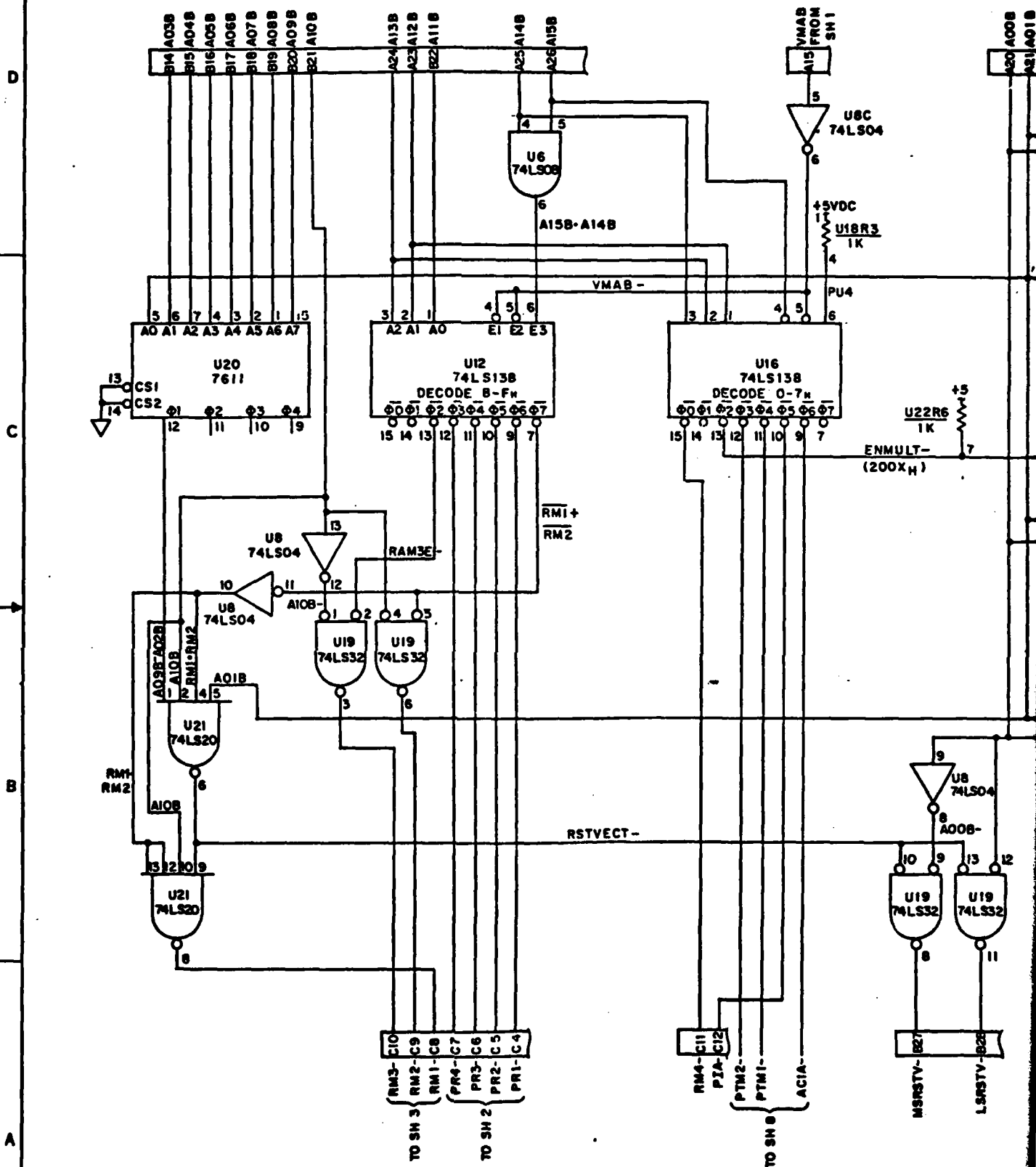
SIZE CODE IDENT NO. 10000 N.

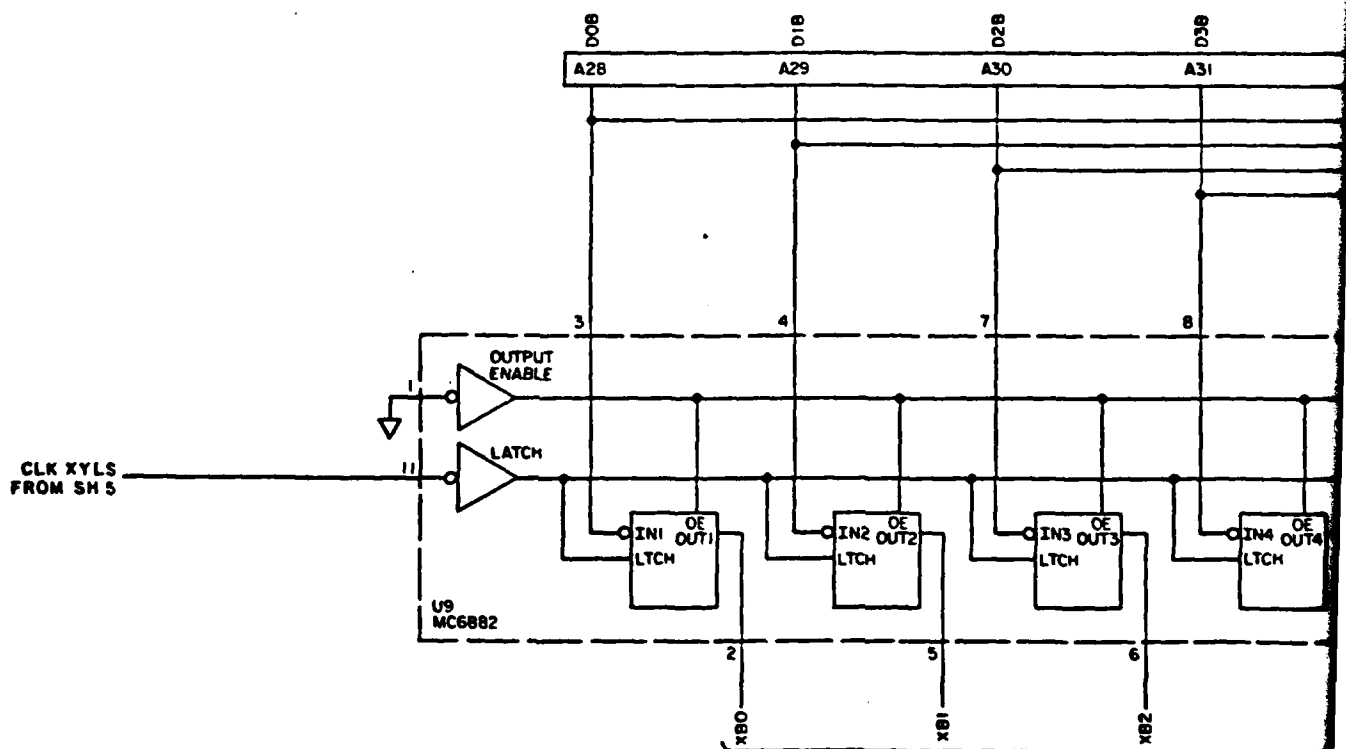
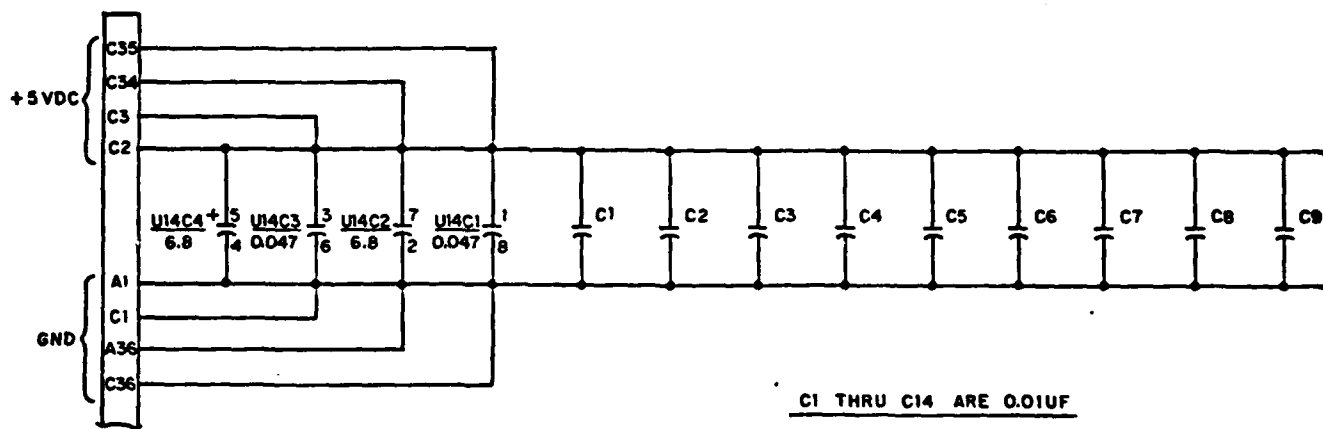
D

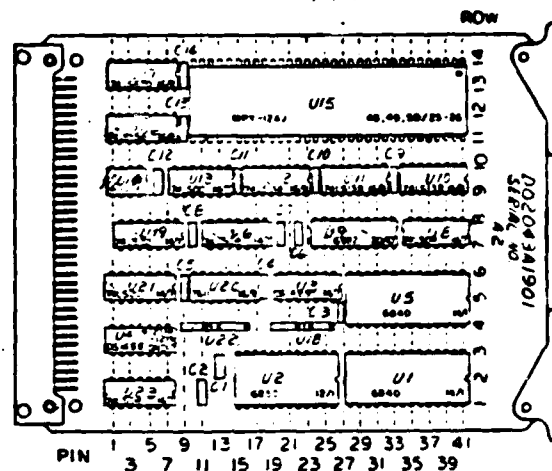
SCALE 1 REV

1 SHEET

2





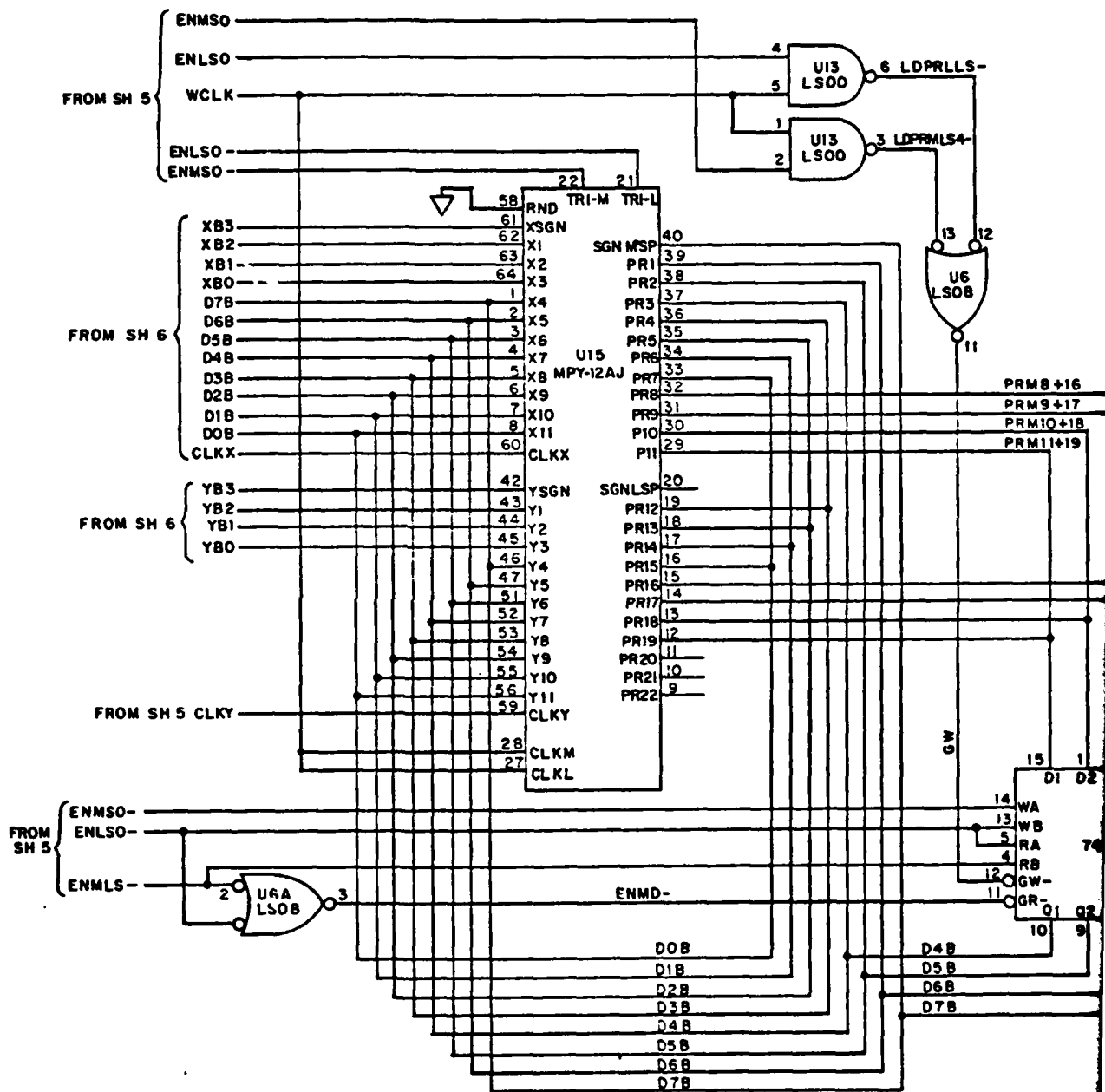


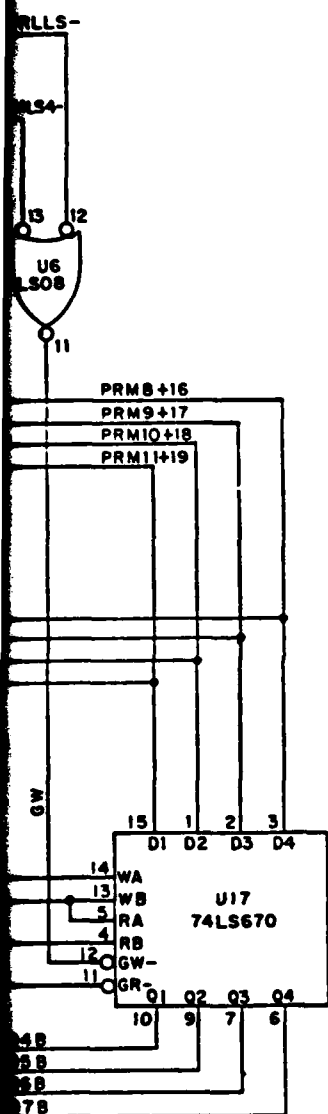
SUB CODE IDENT NO. 1 DWG A.

D

SCALE — 1 REV

15477 6





A2 MULTIPLIER

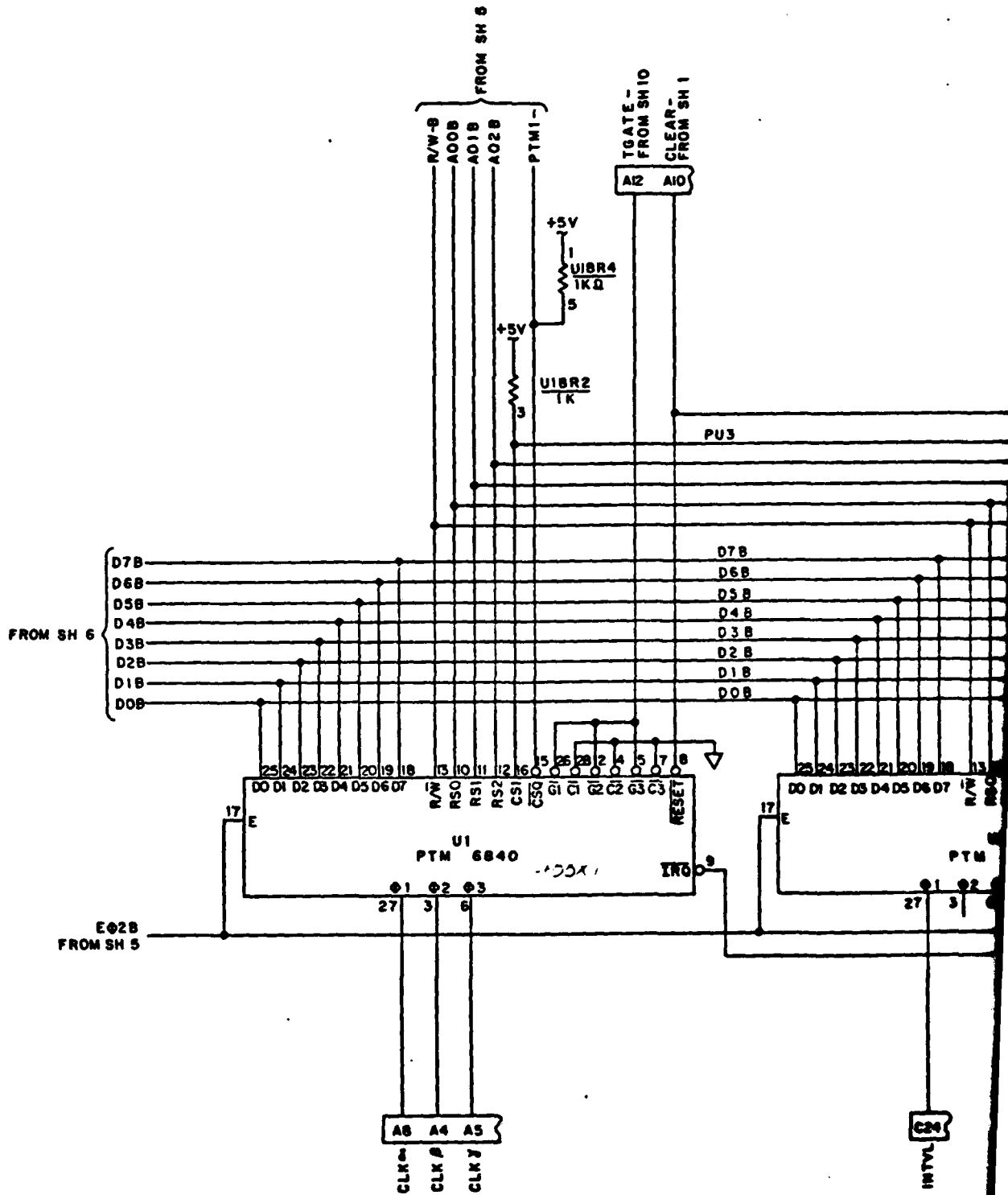
SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE	REV	SHEET
		7

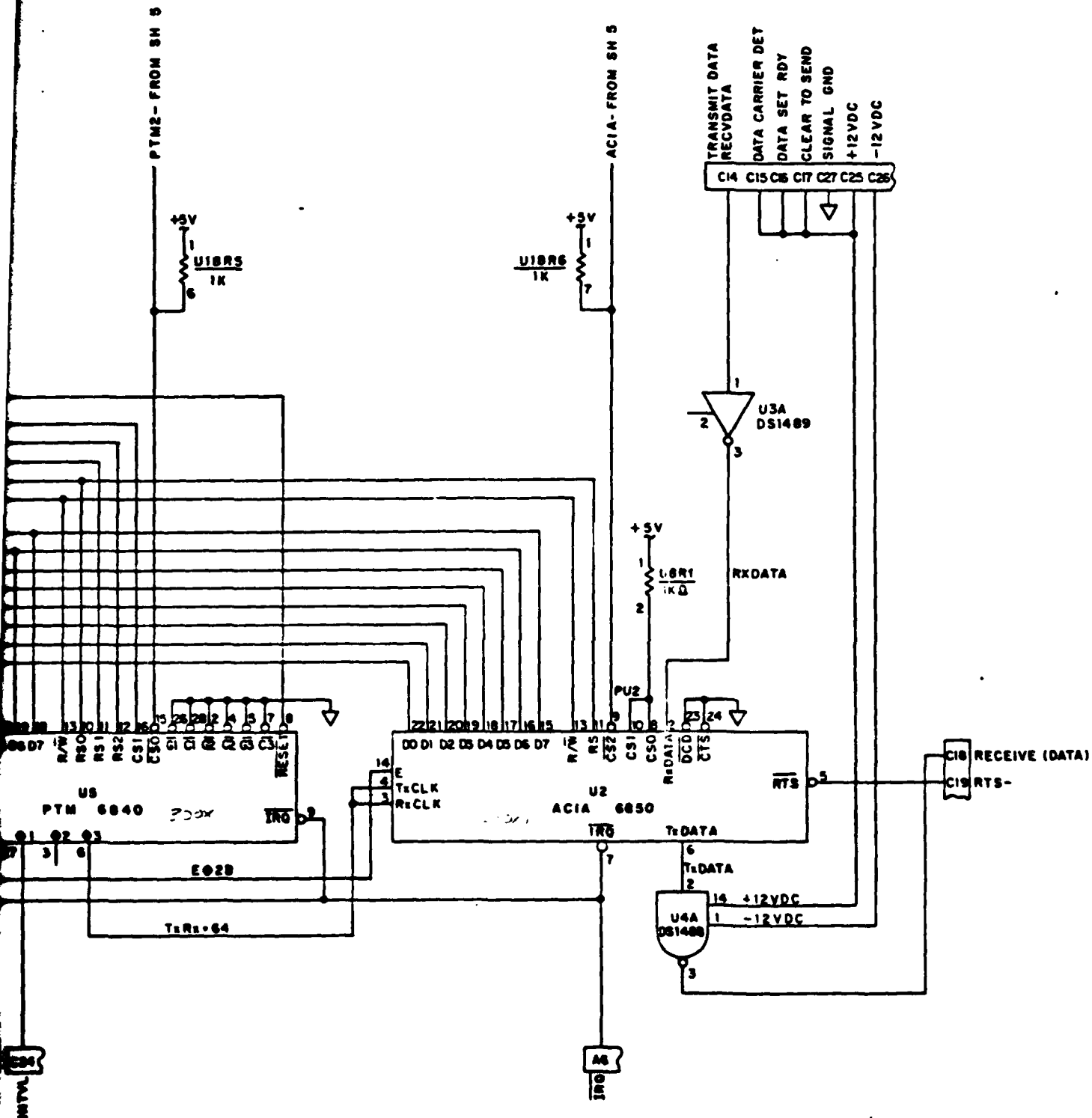
3

2

2

1



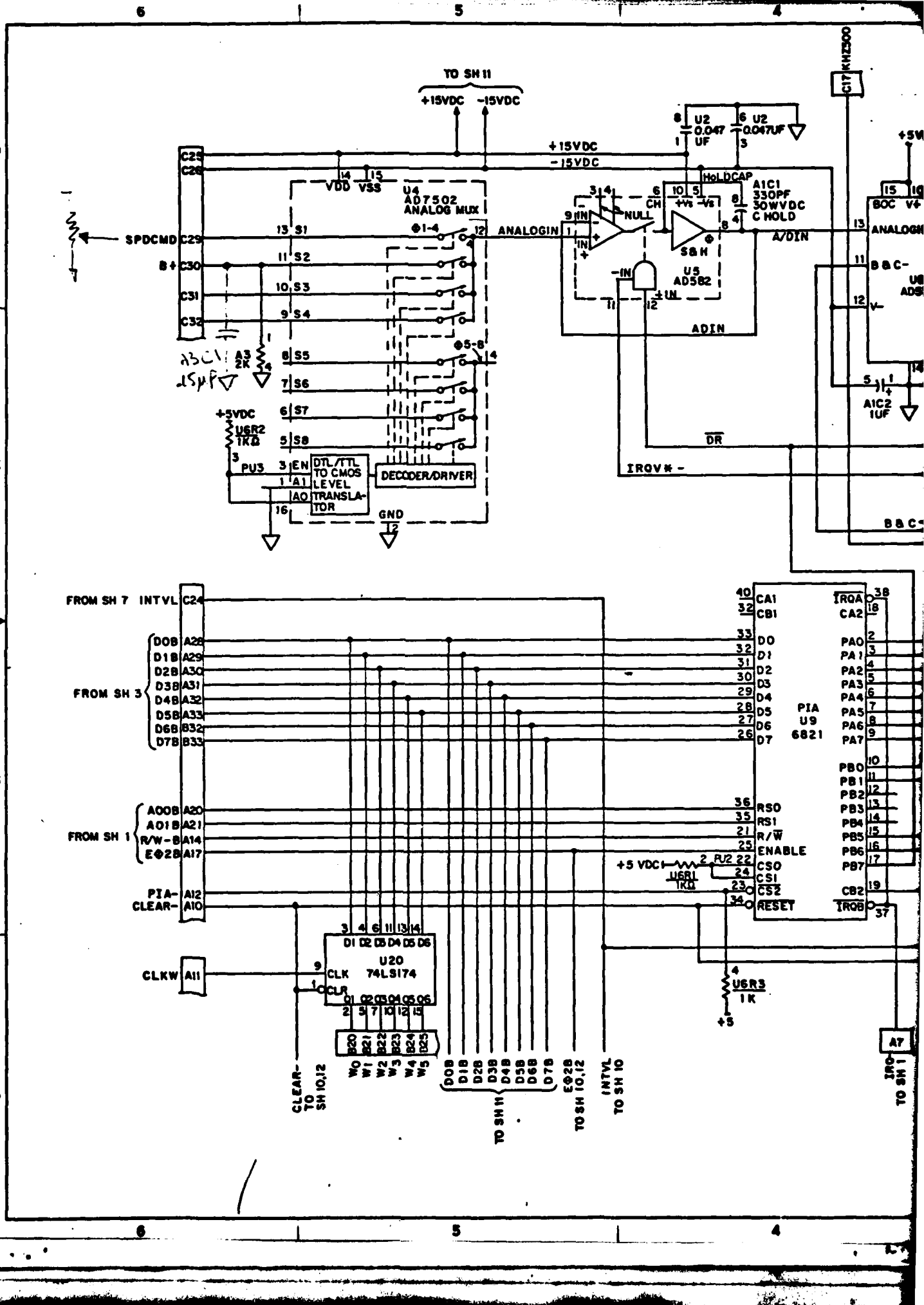


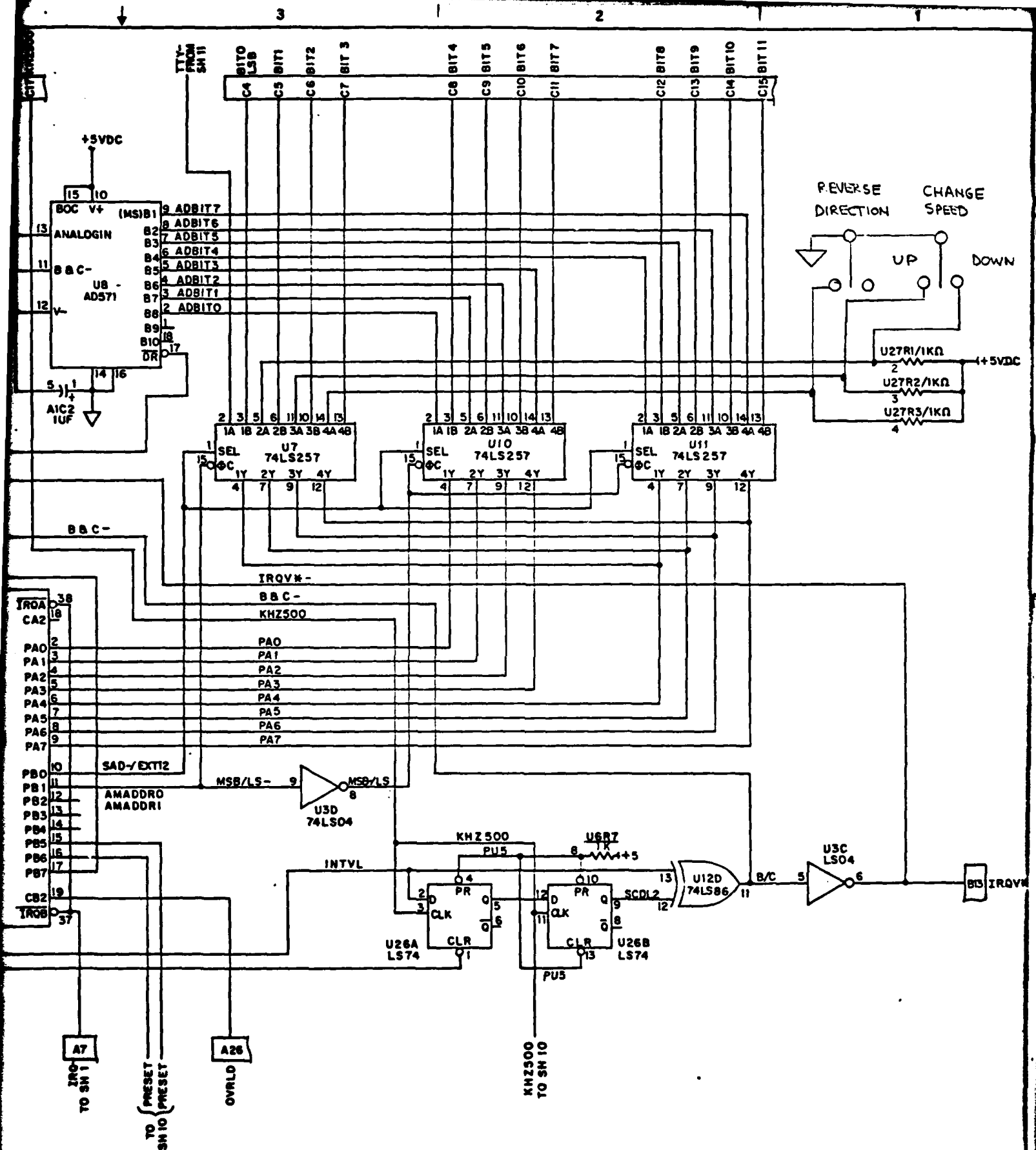
A2 MULTIPLIER PTM'S & ACIA

DATE: CODE IDENT NO. DWG NO.

D

SCALE: REV SHEET 2

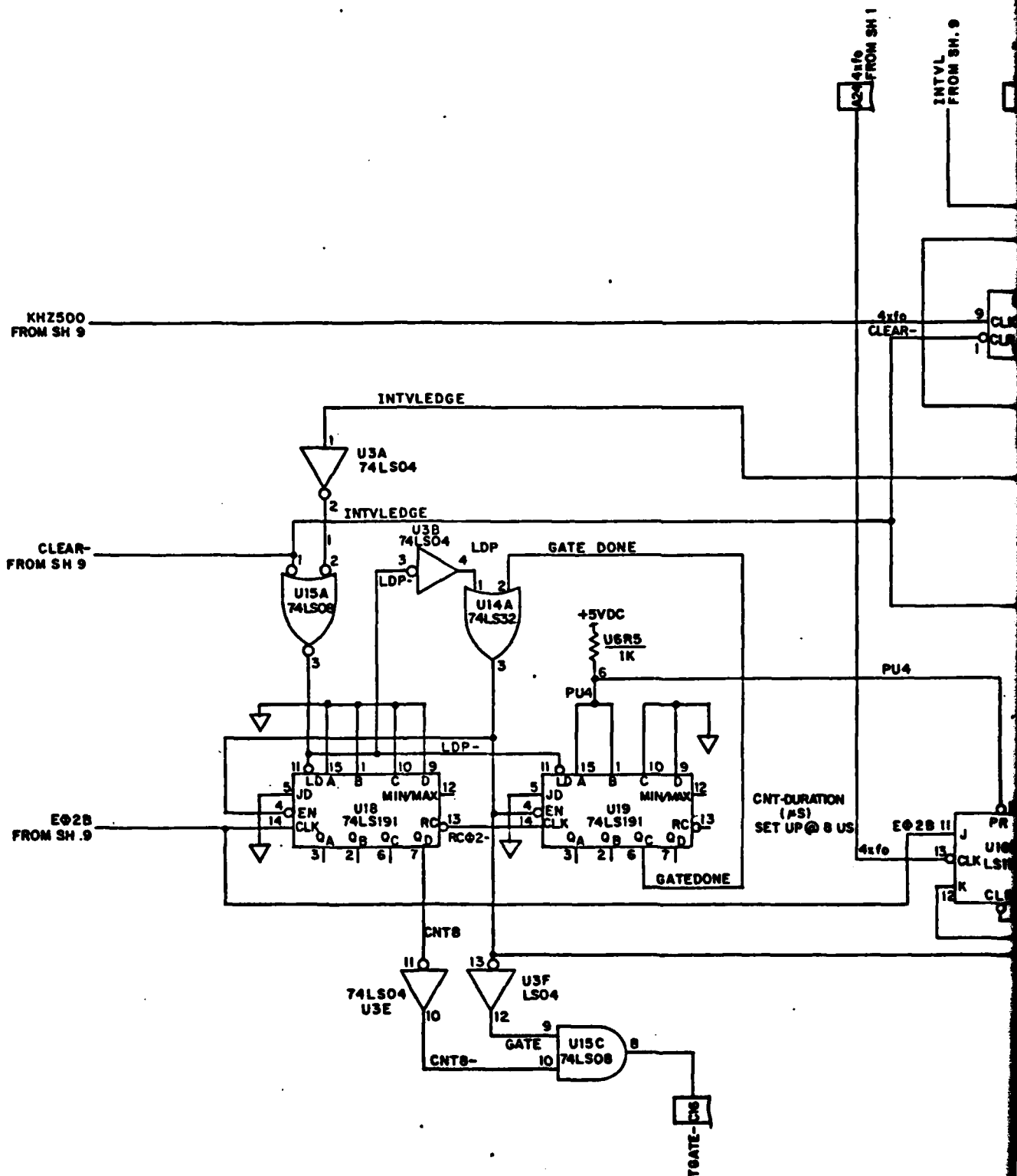


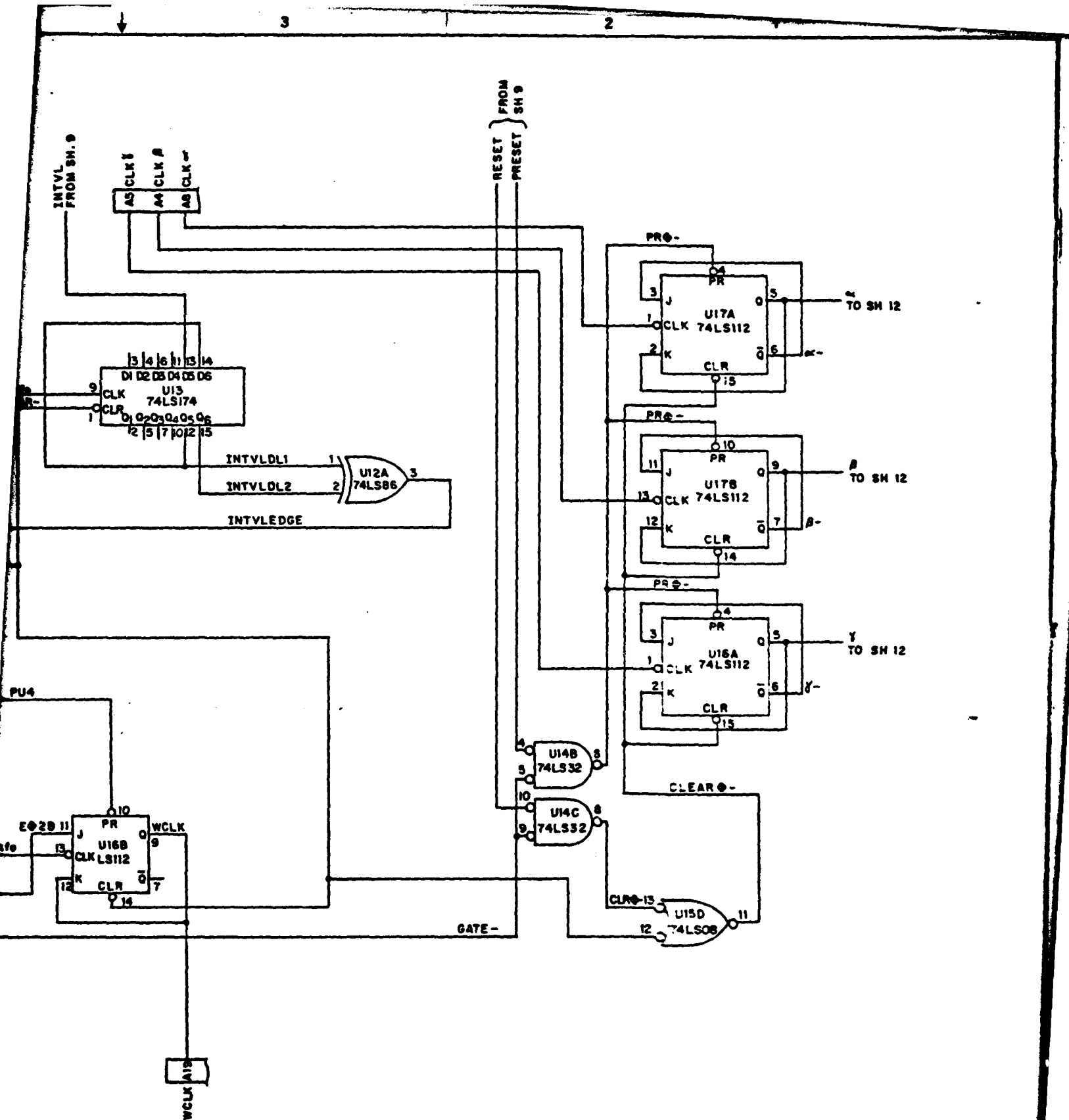


A3 PIA; DATA ACQUISITION

SIZE	CODE IDENT NO.	DWG NO.
D	1	
SCALE	REV	SHEET 9

2 1





A3 PIA

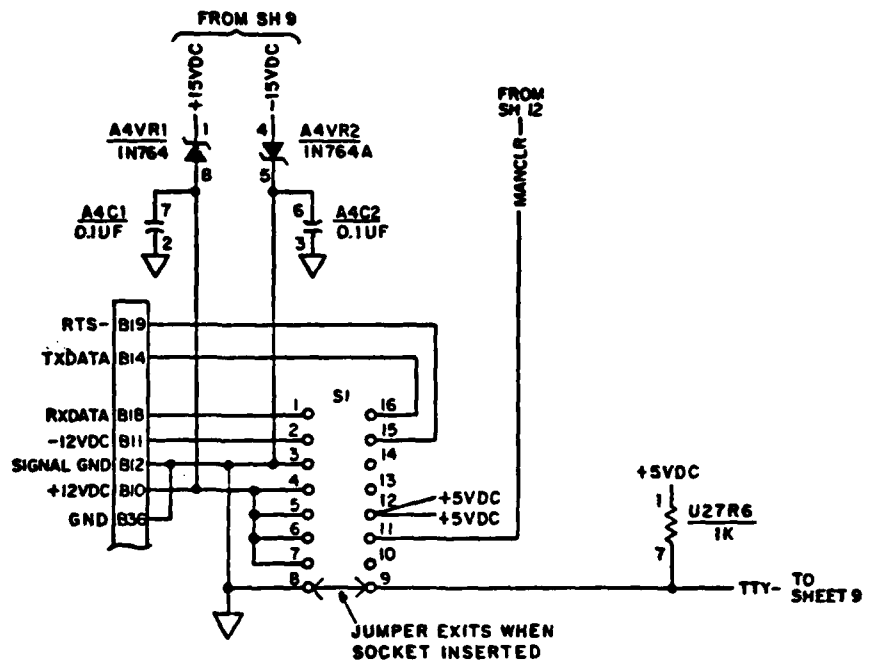
SIZE CODE IDENT NO. 1 DAG NO.

D

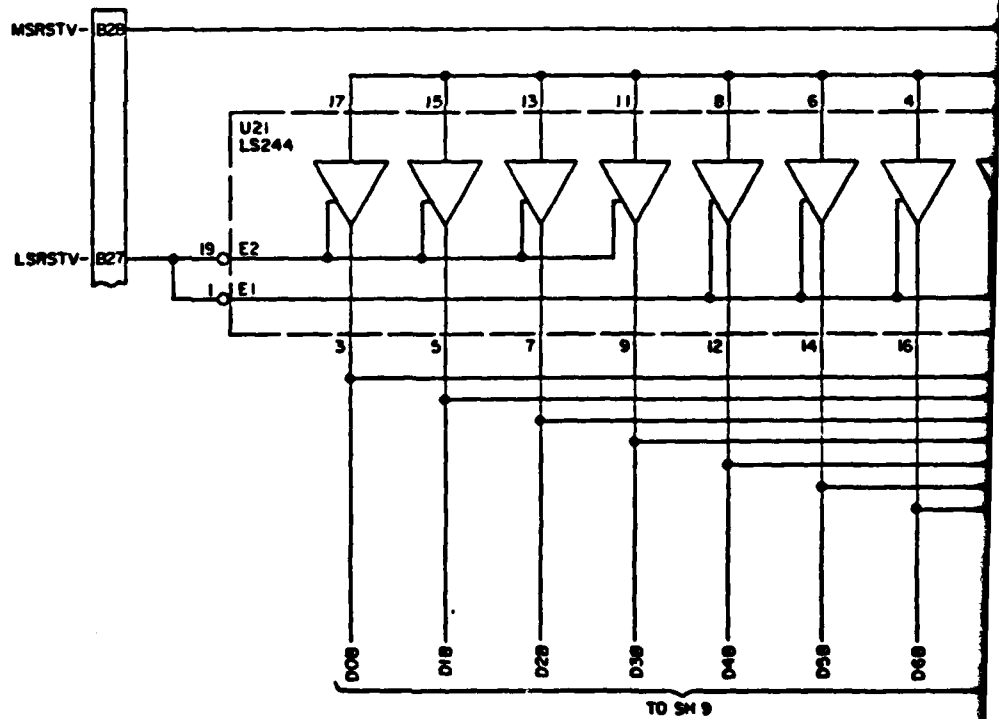
SCALE - 1 REV

SHEET 1

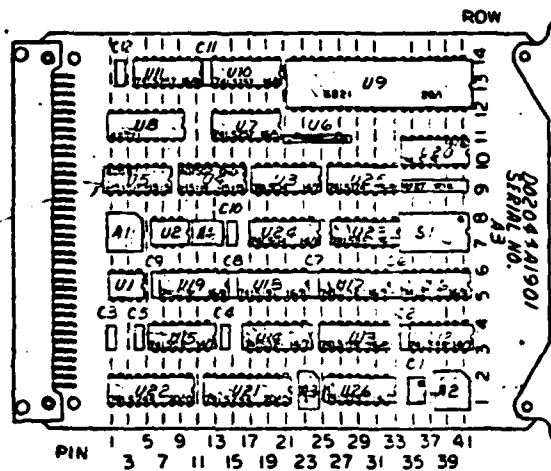
2



(Signals presence of terminal)

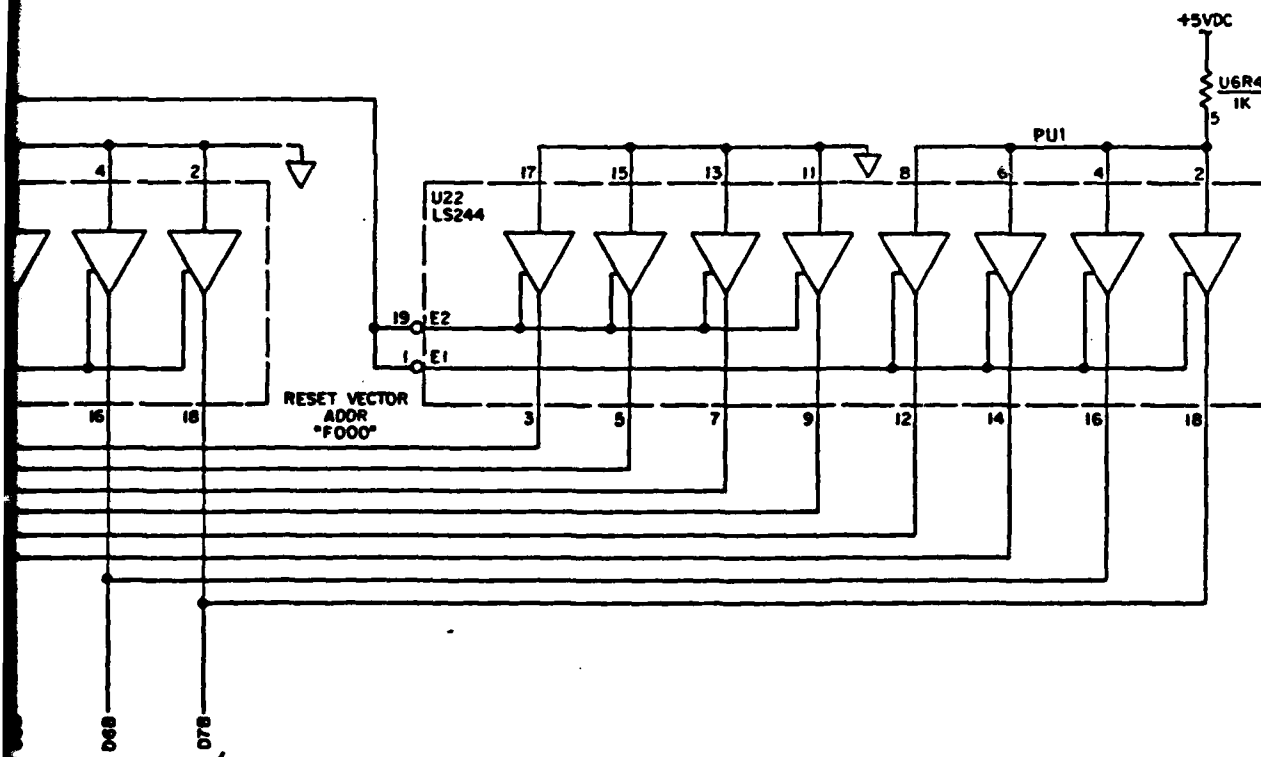


terminal)



15

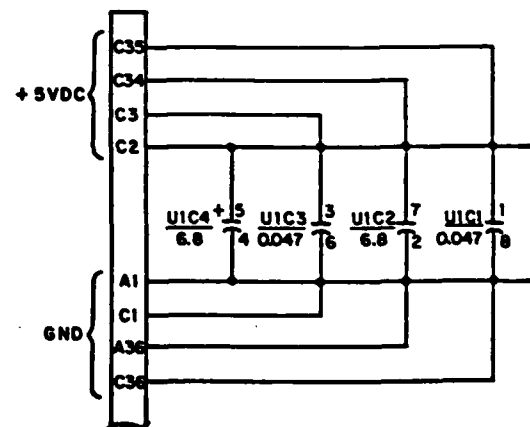
U1 & U2 = SPRAGUE
939Z685X0035DE

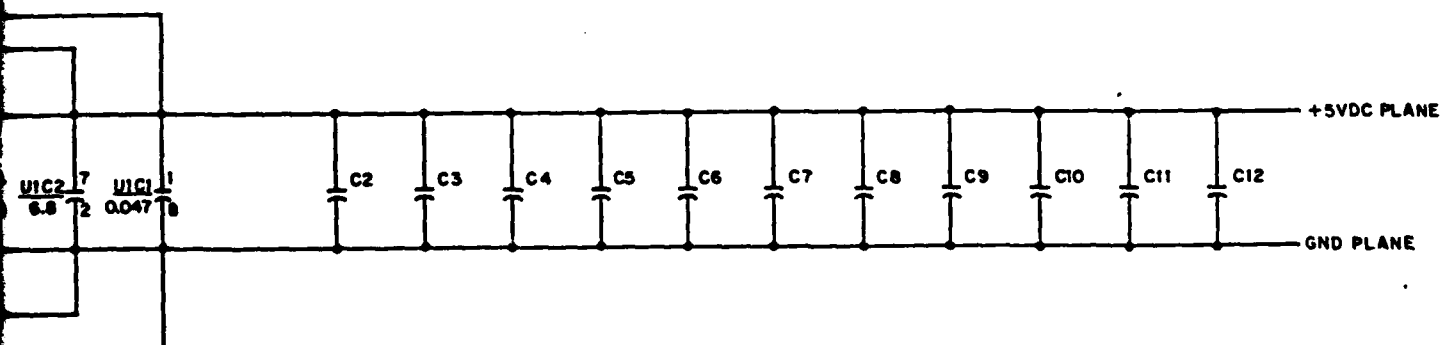
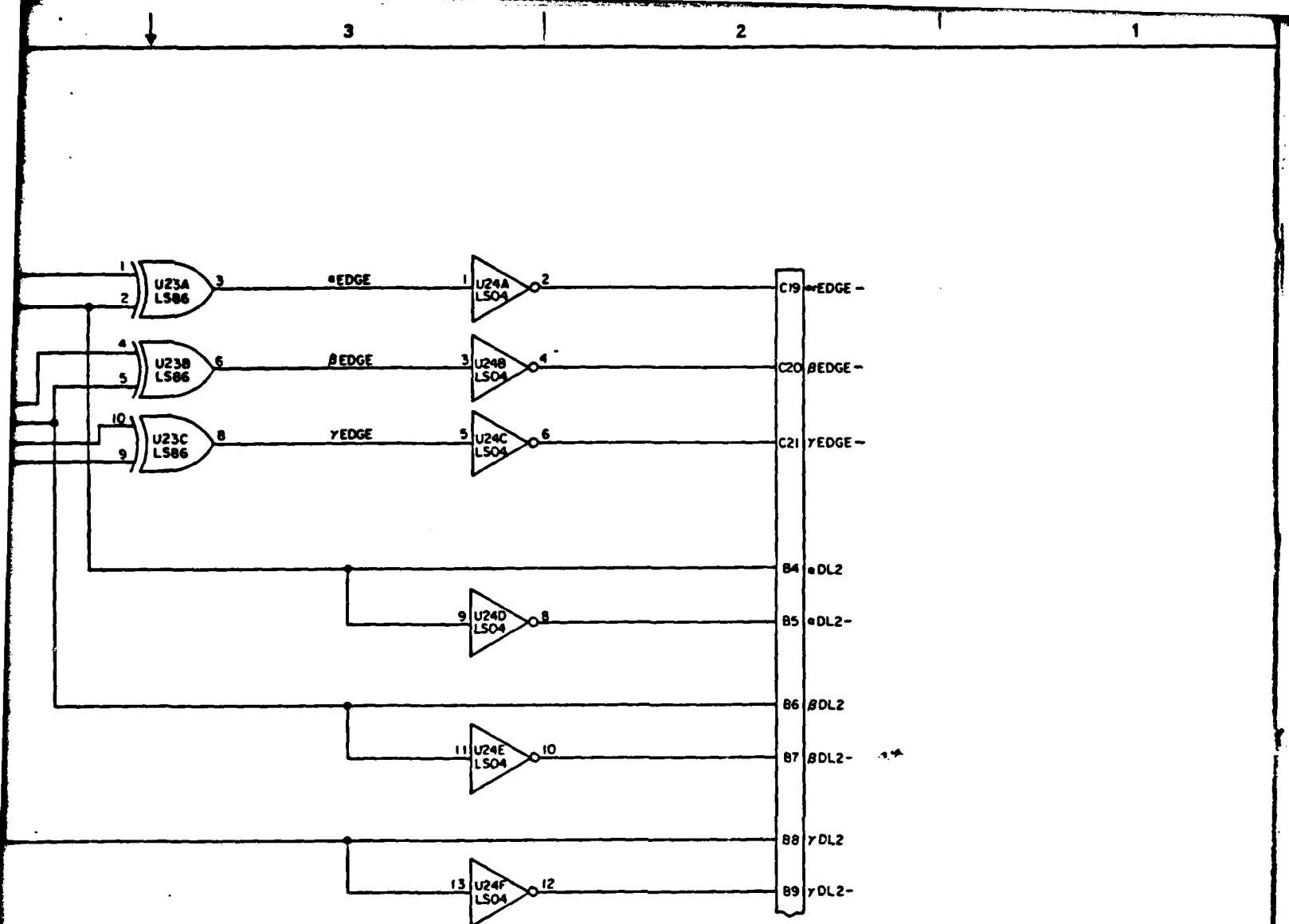


A3 PIA

SIZE	CODE IDENT NO.	DWG NO.
D1		
SCALE	REV	SHEET 11

2





C2 THRU C12 ARE 0.01UF

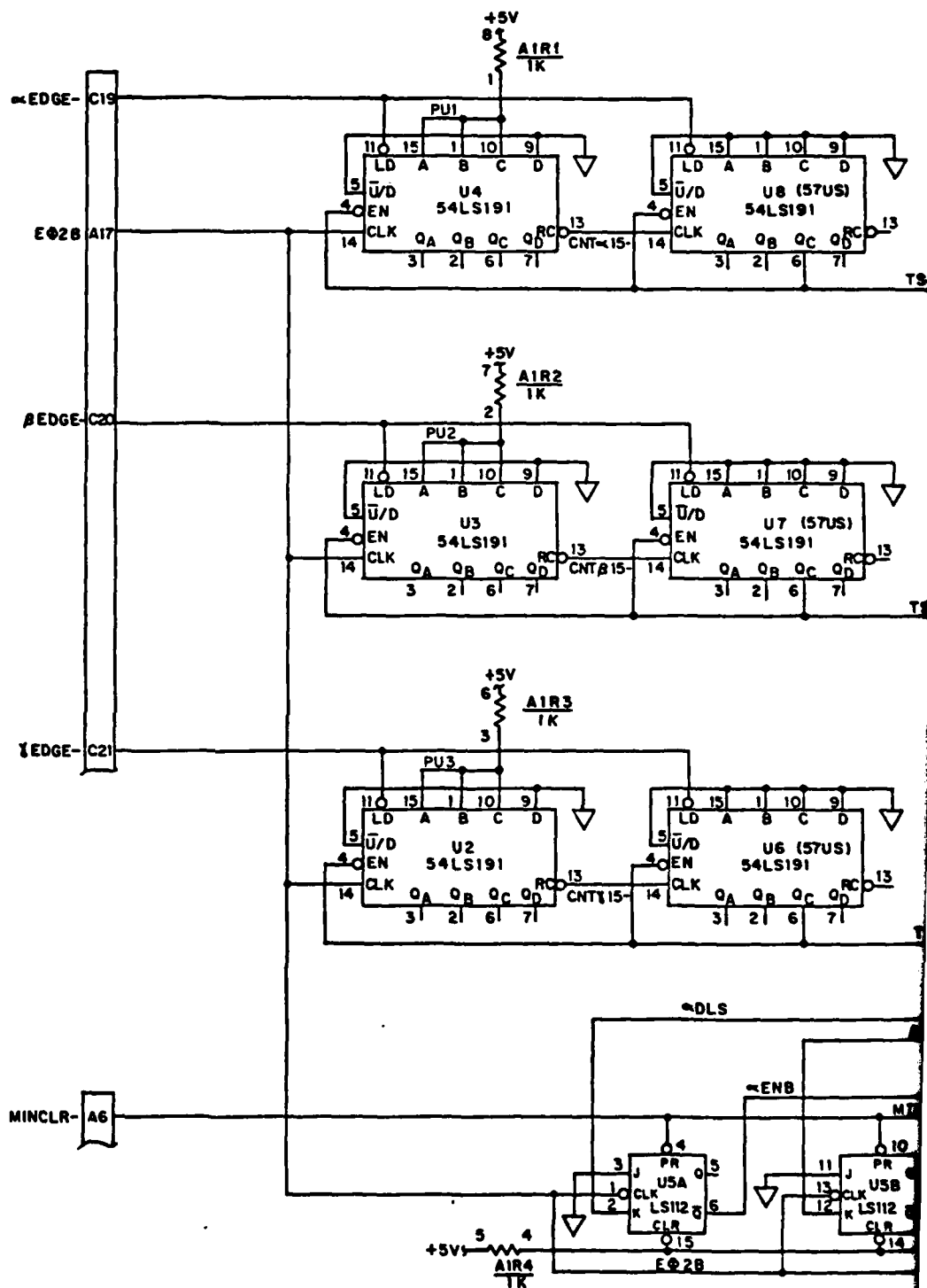
A3 PIA-ST LOGIC (PART OF A3)

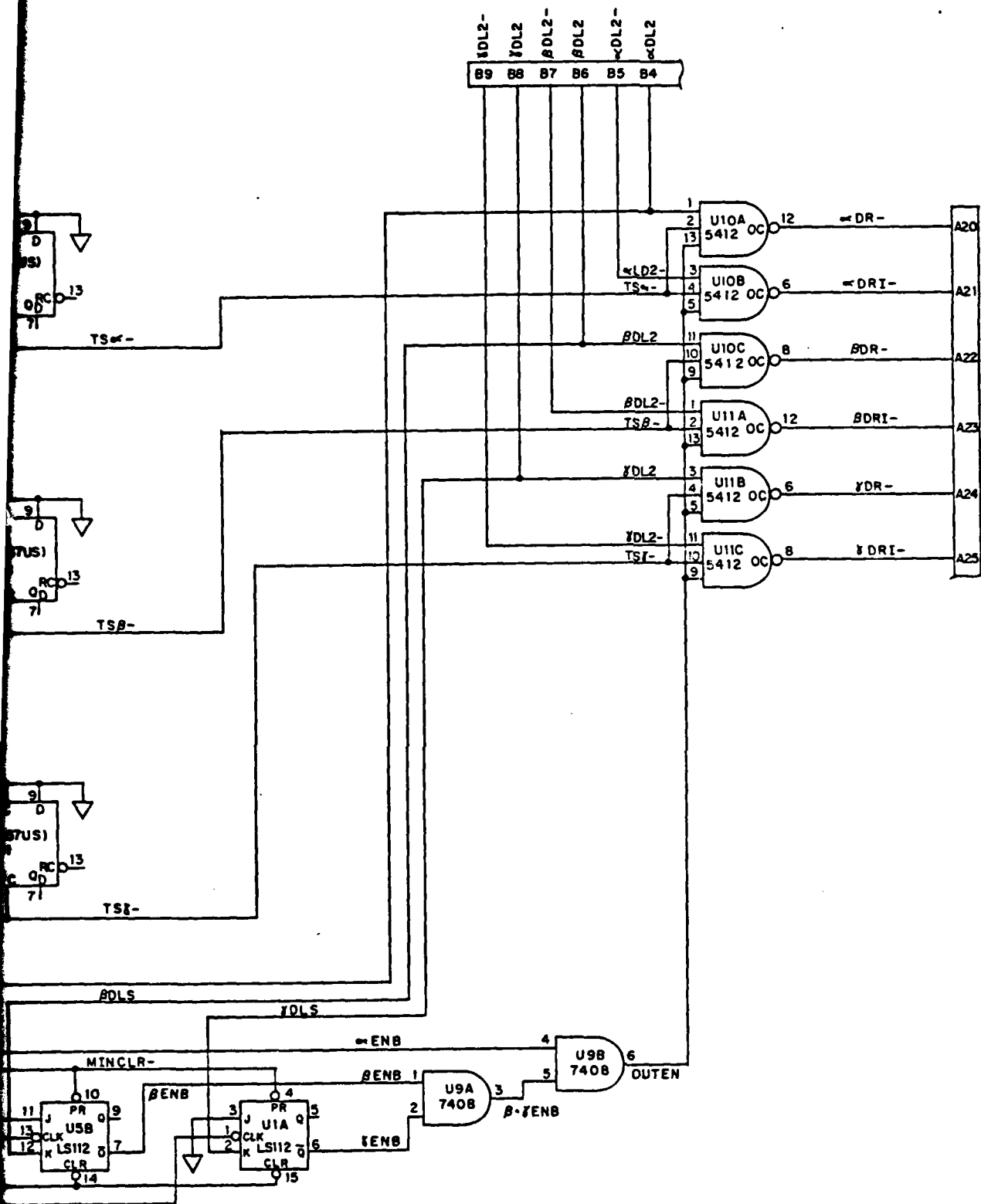
SIZE: CODE IDENT NO. (CAG NO.)

D

SCALE - REV

SHEET 12





A4 S.T. & FILTERS

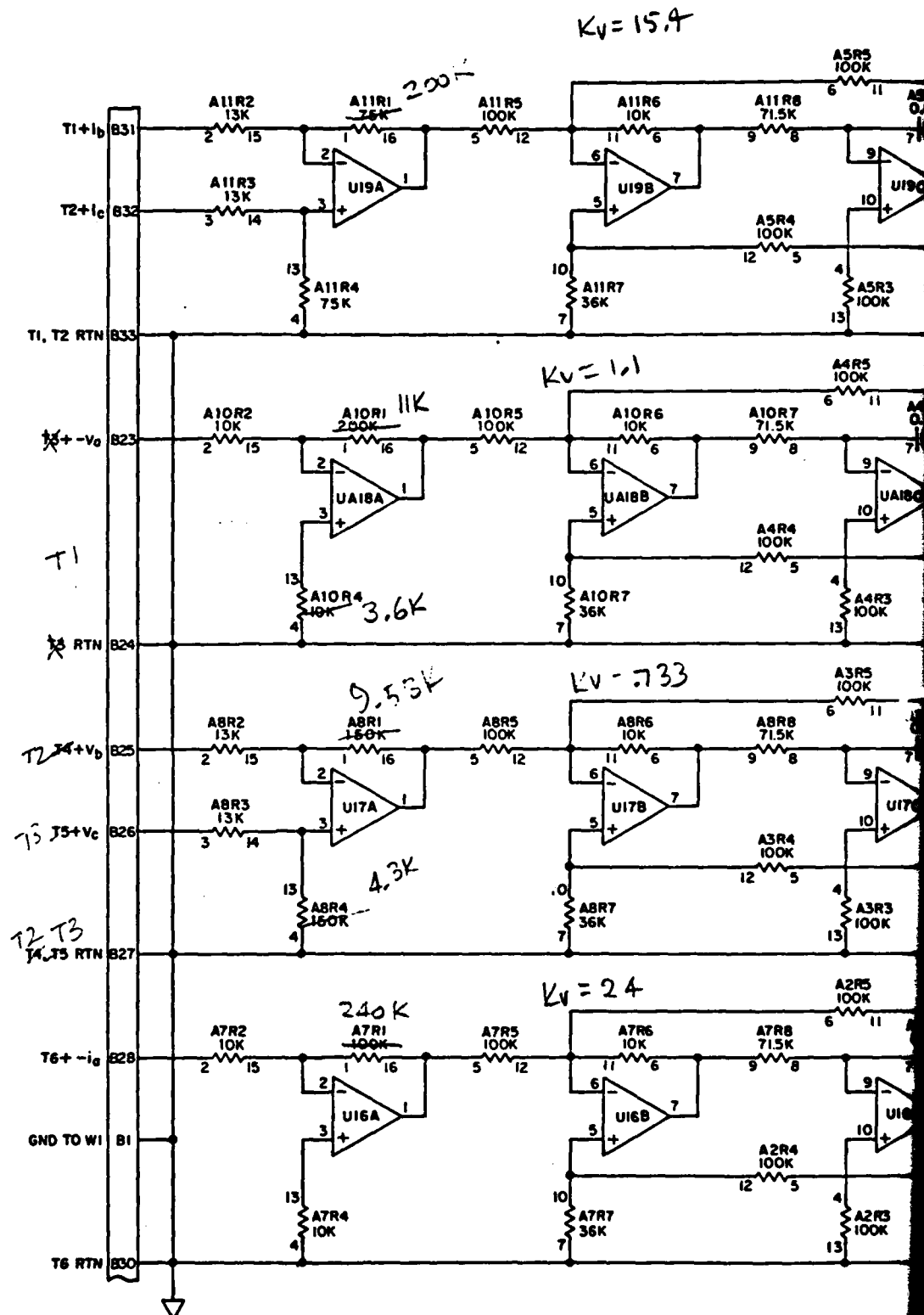
SIZE: CODE IDENT NO. DATE NO.

D

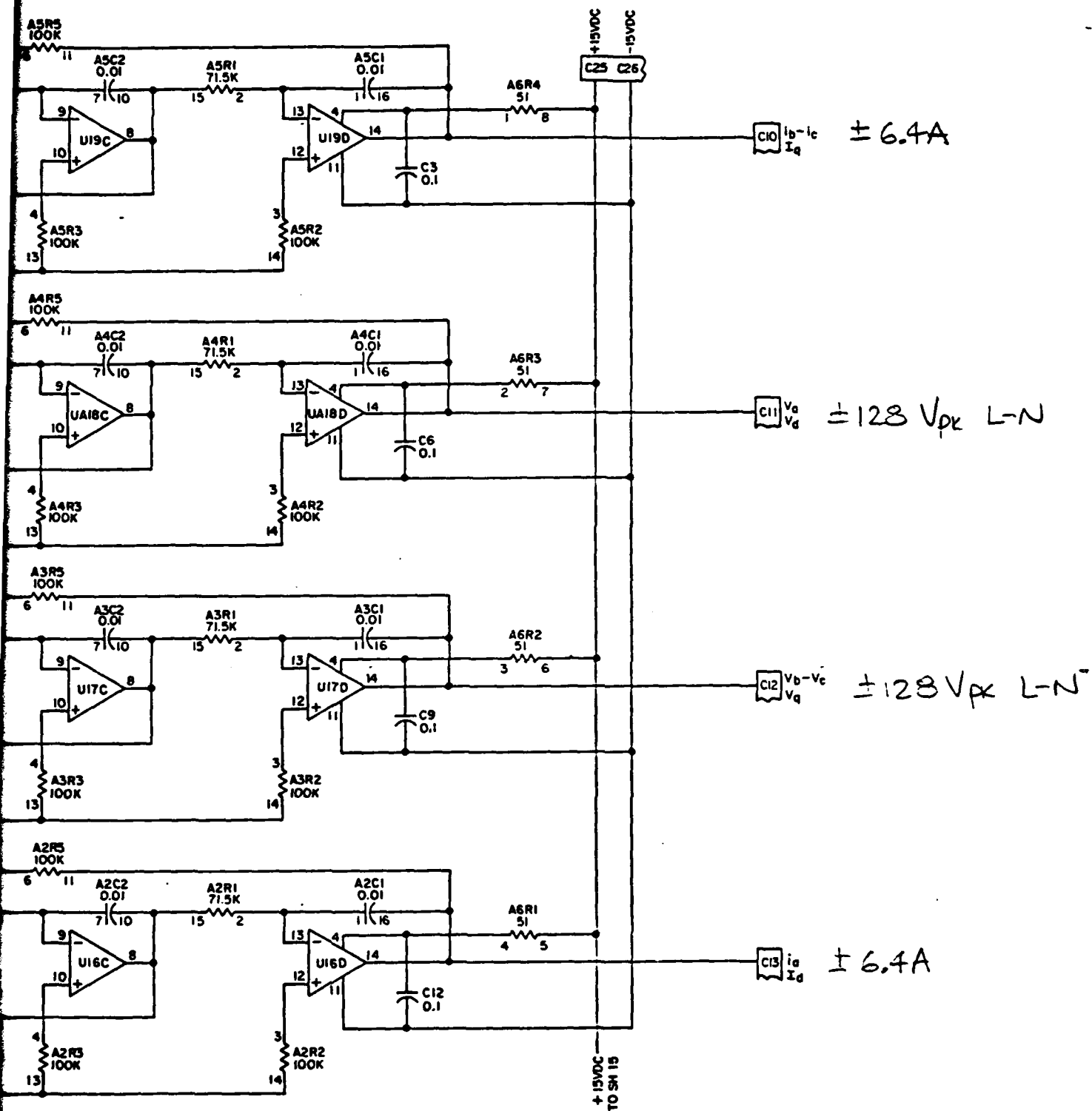
SCALE: 1 REV

SHEET 12

2



NOTE: $A7R1$, $A8R1$, $A10R1$, $A11R1$ ESTABLISH FEEDBACK SCALING.



46. (APPLICATION DEPENDENT)

A4 S.T. & L.P.F.

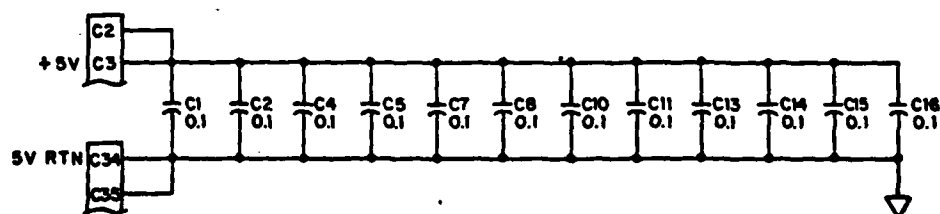
SIZE CODE IDENT NO. DWG NO.

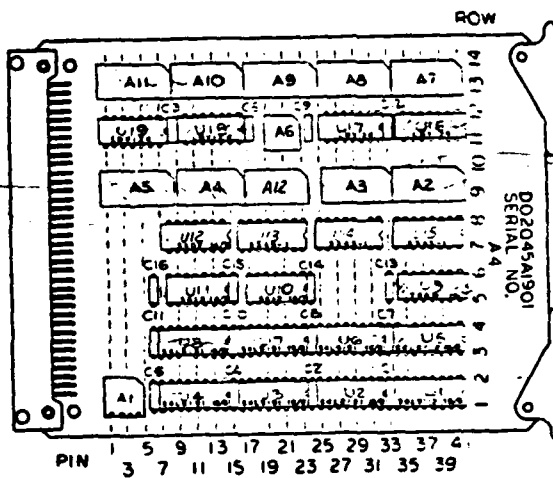
D

SCALE 1 REV

SHEET 14

2





NOTE: ON LM124 PIN 4 = V+ (+15VDC)
PIN 11 = V- (-15VDC)

A4 S.T. & LPF

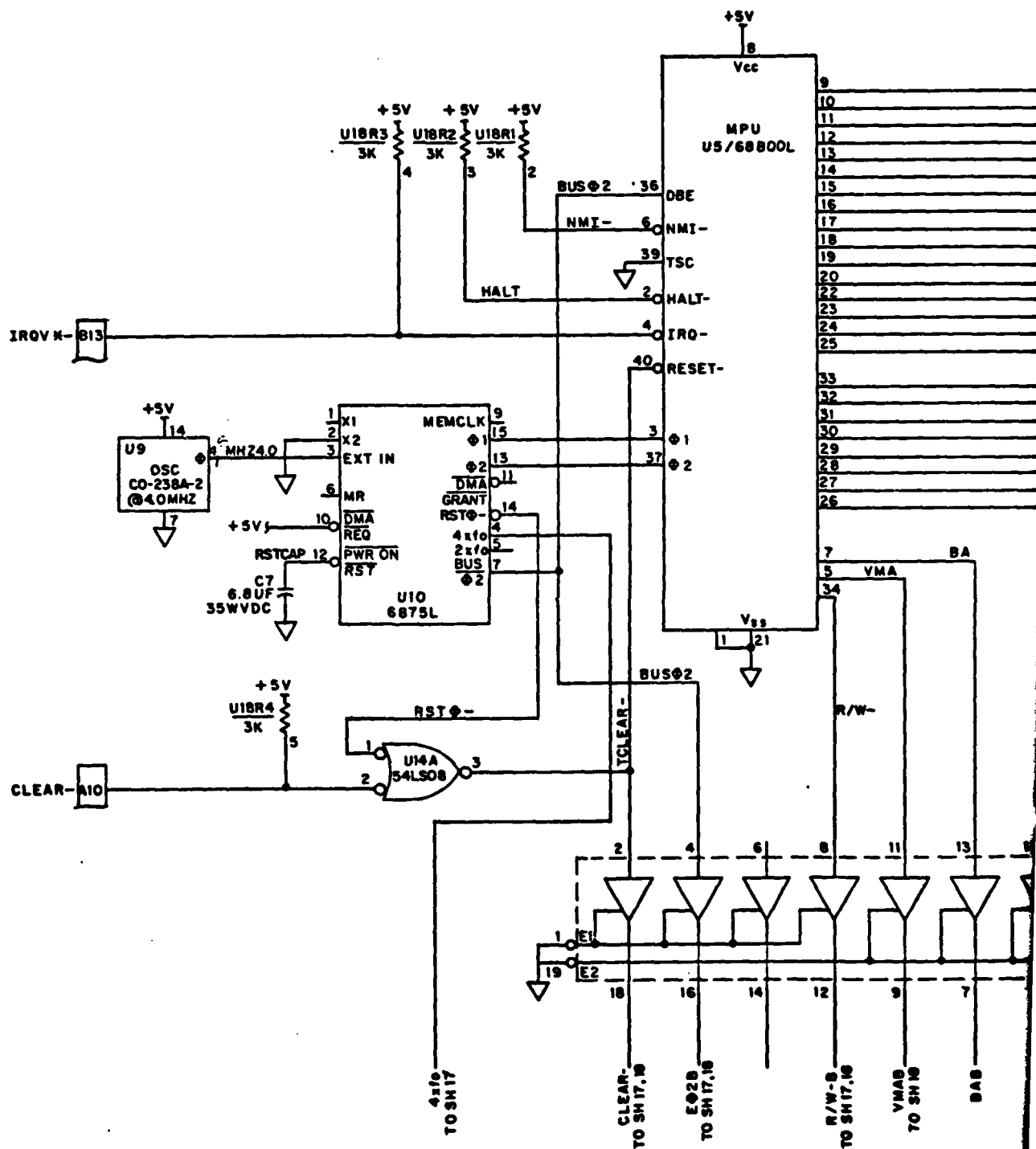
SIZE CODE IDENT NO. DWG NO.

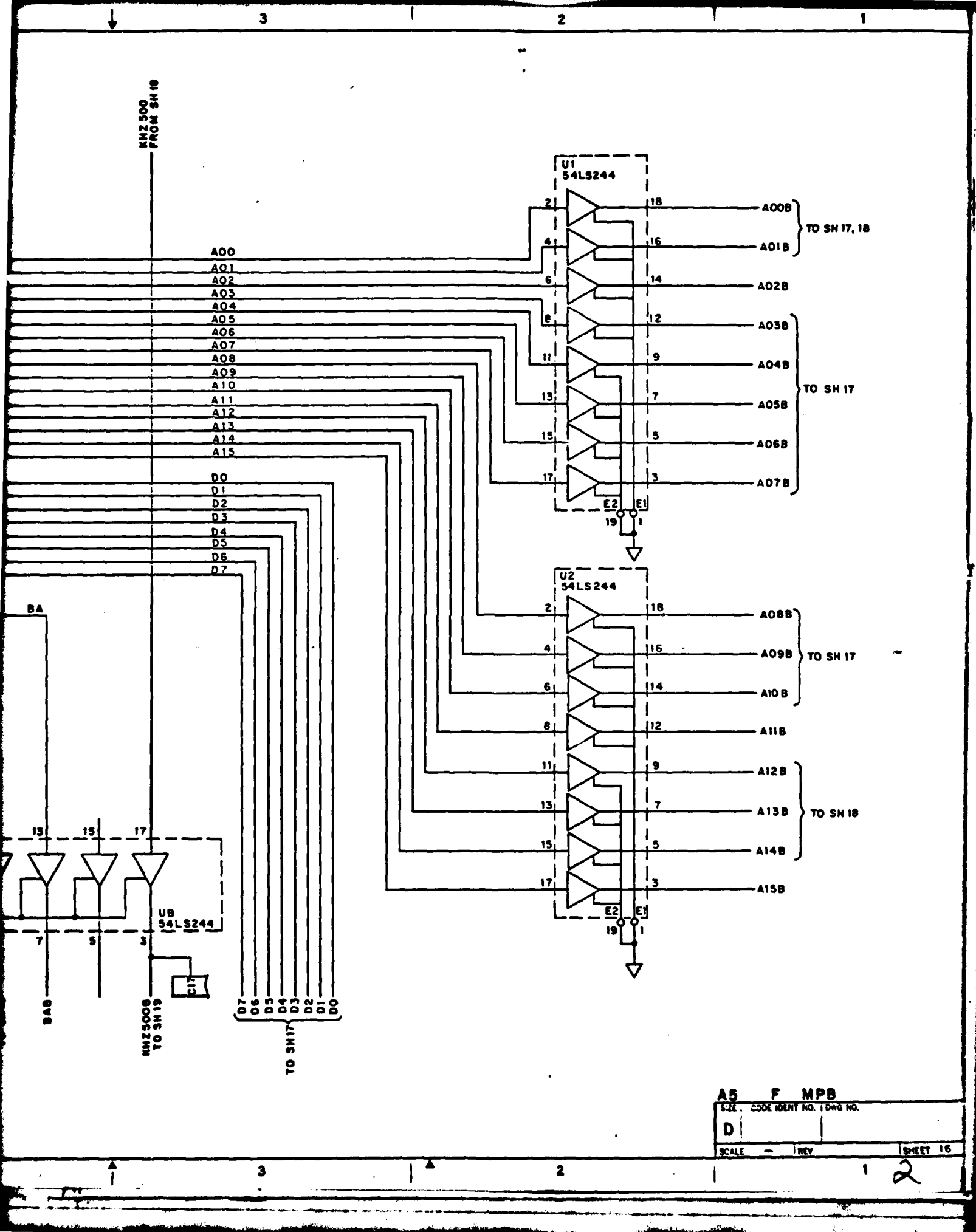
D

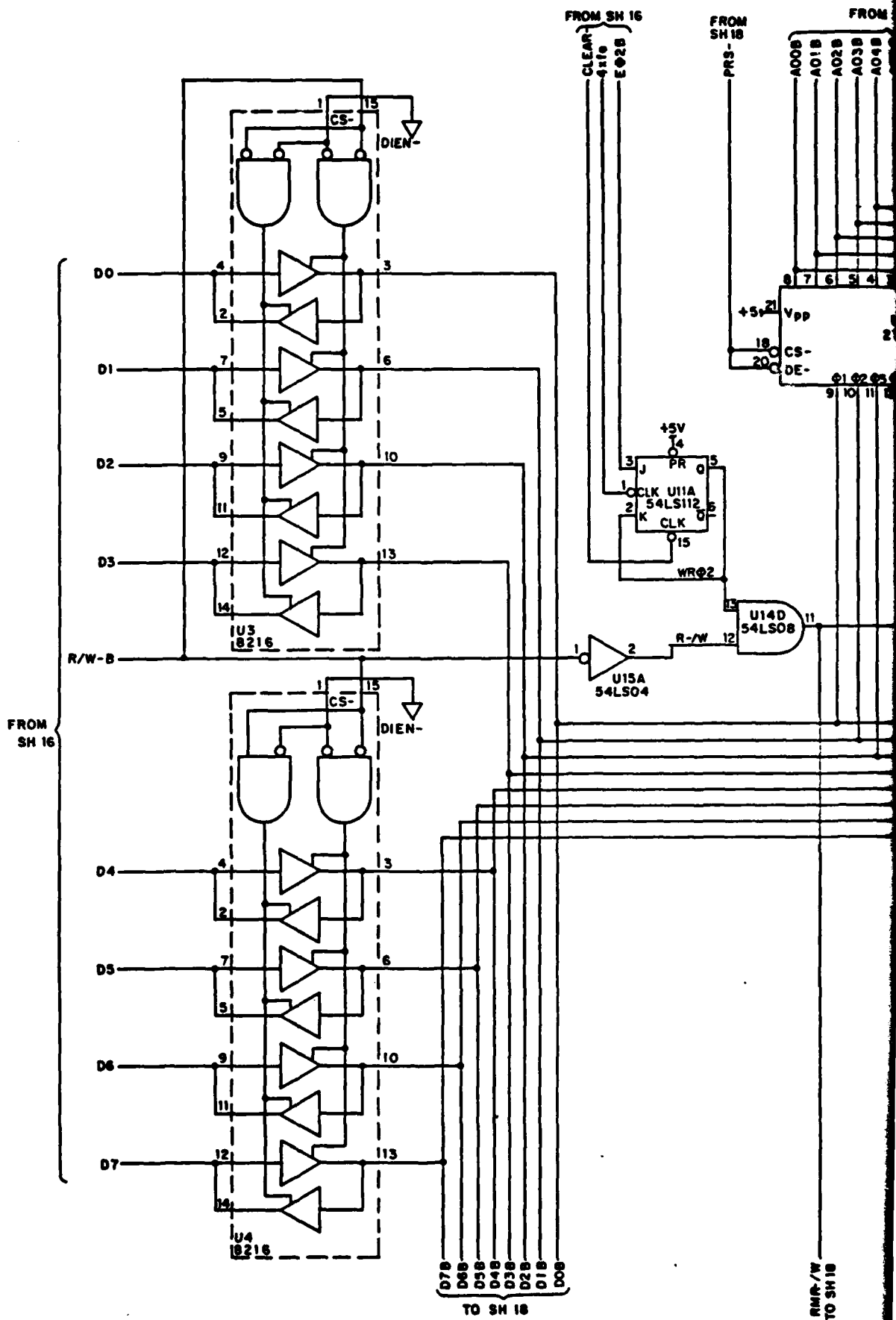
SCALE - REV

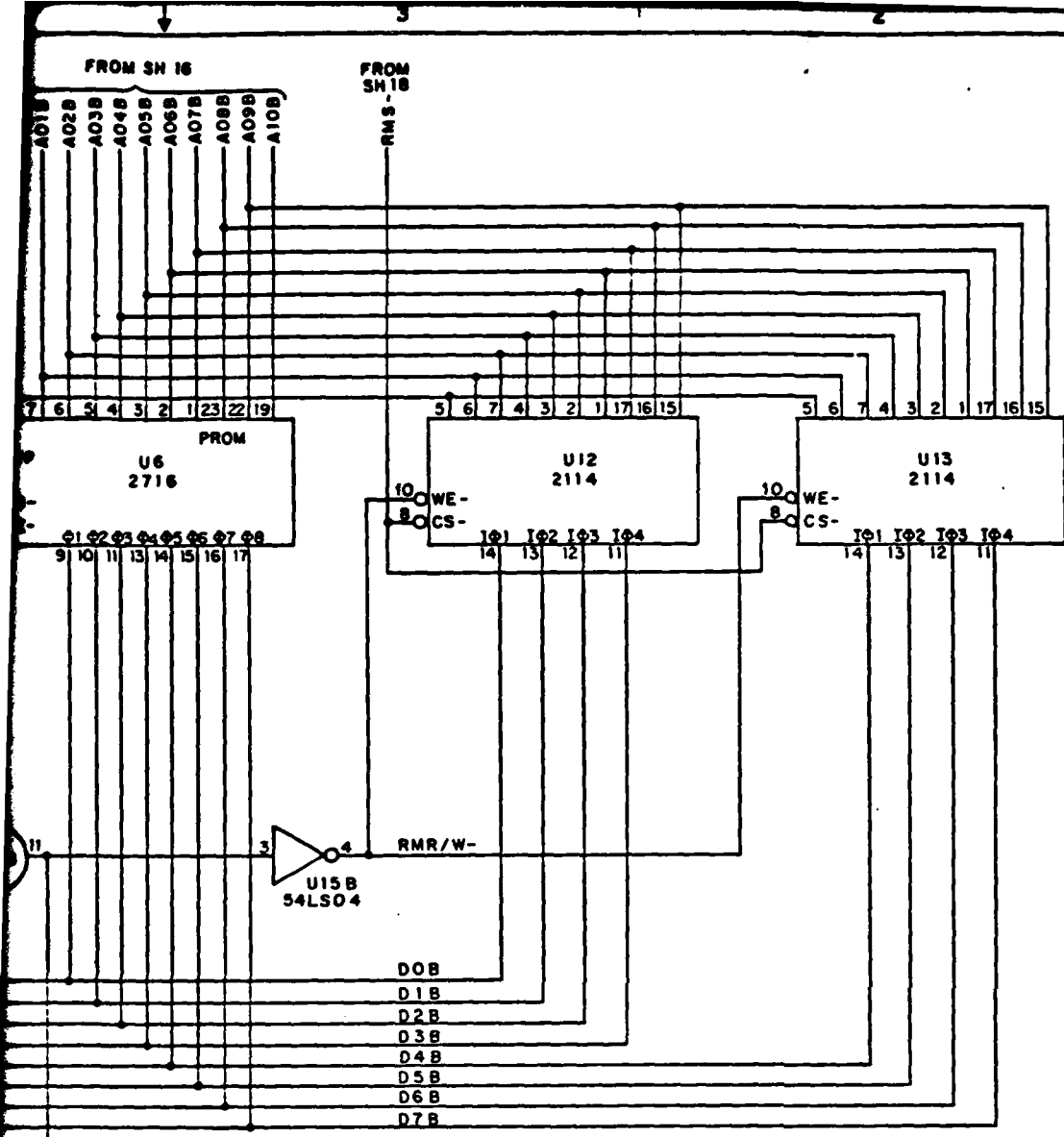
SHEET 15

2



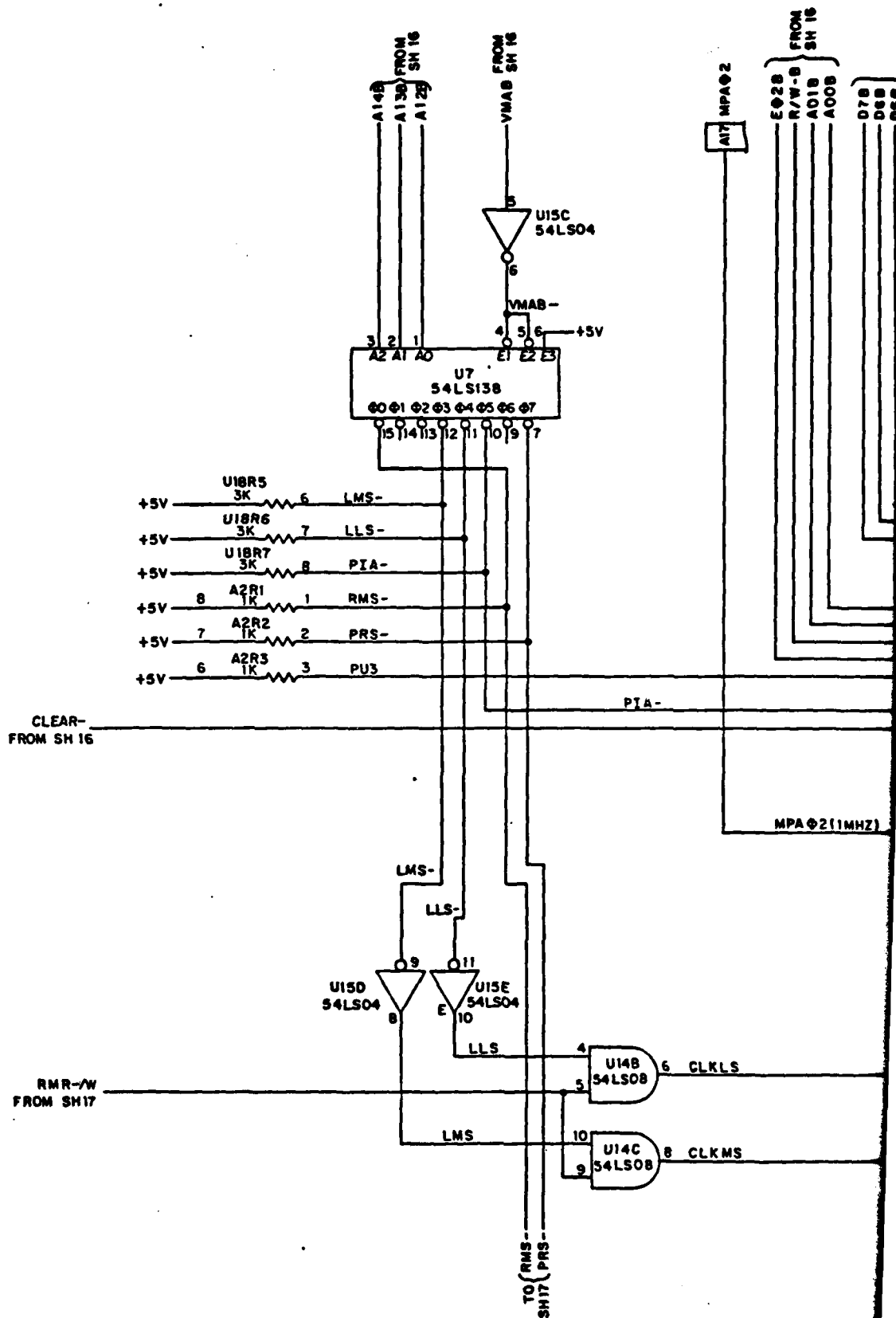


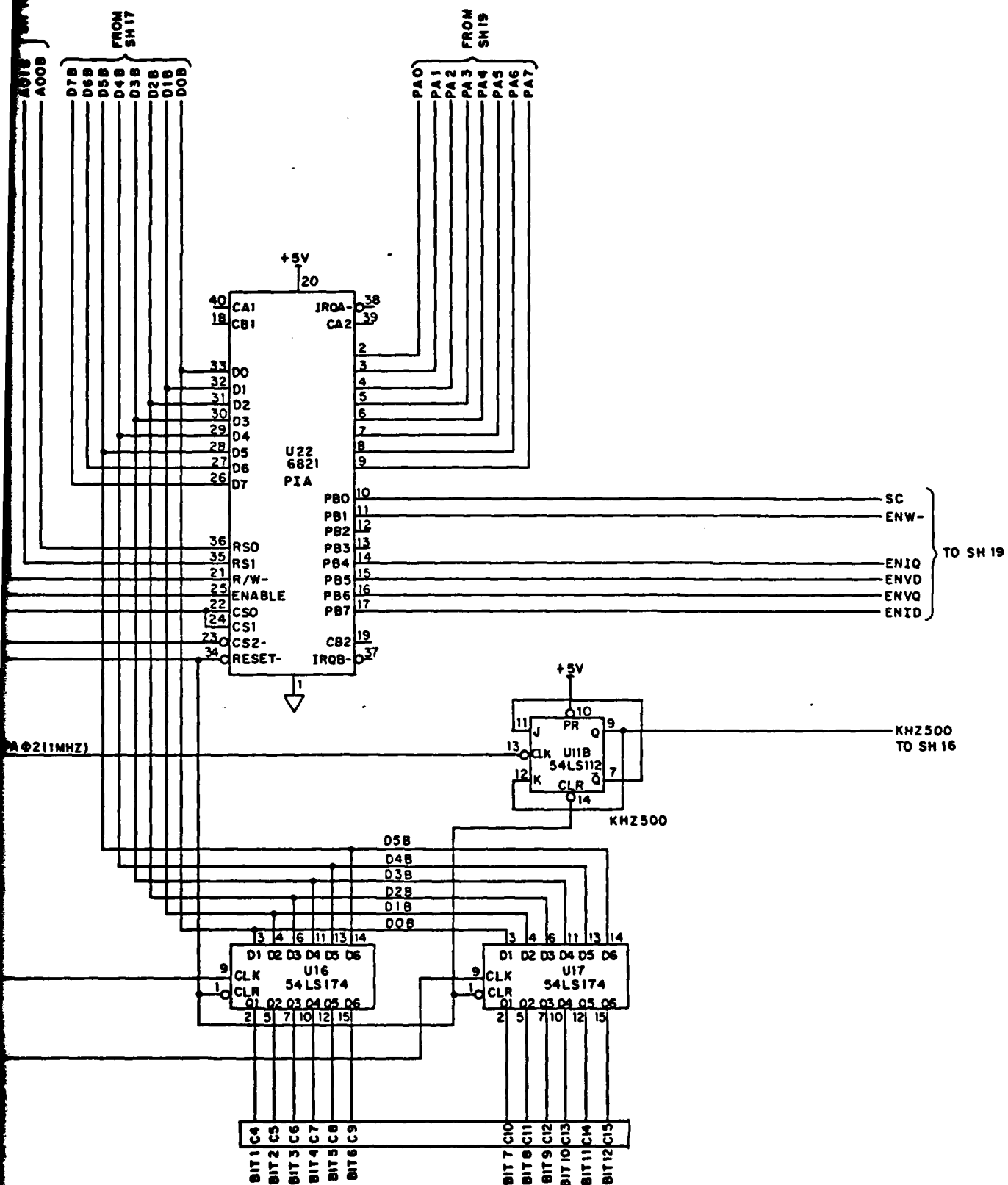




A5 F MPB	
SIZE 1	CODE IDENT NO. 1 DWG NO.
D	
SCALE -	REV
SHEET 17	

2

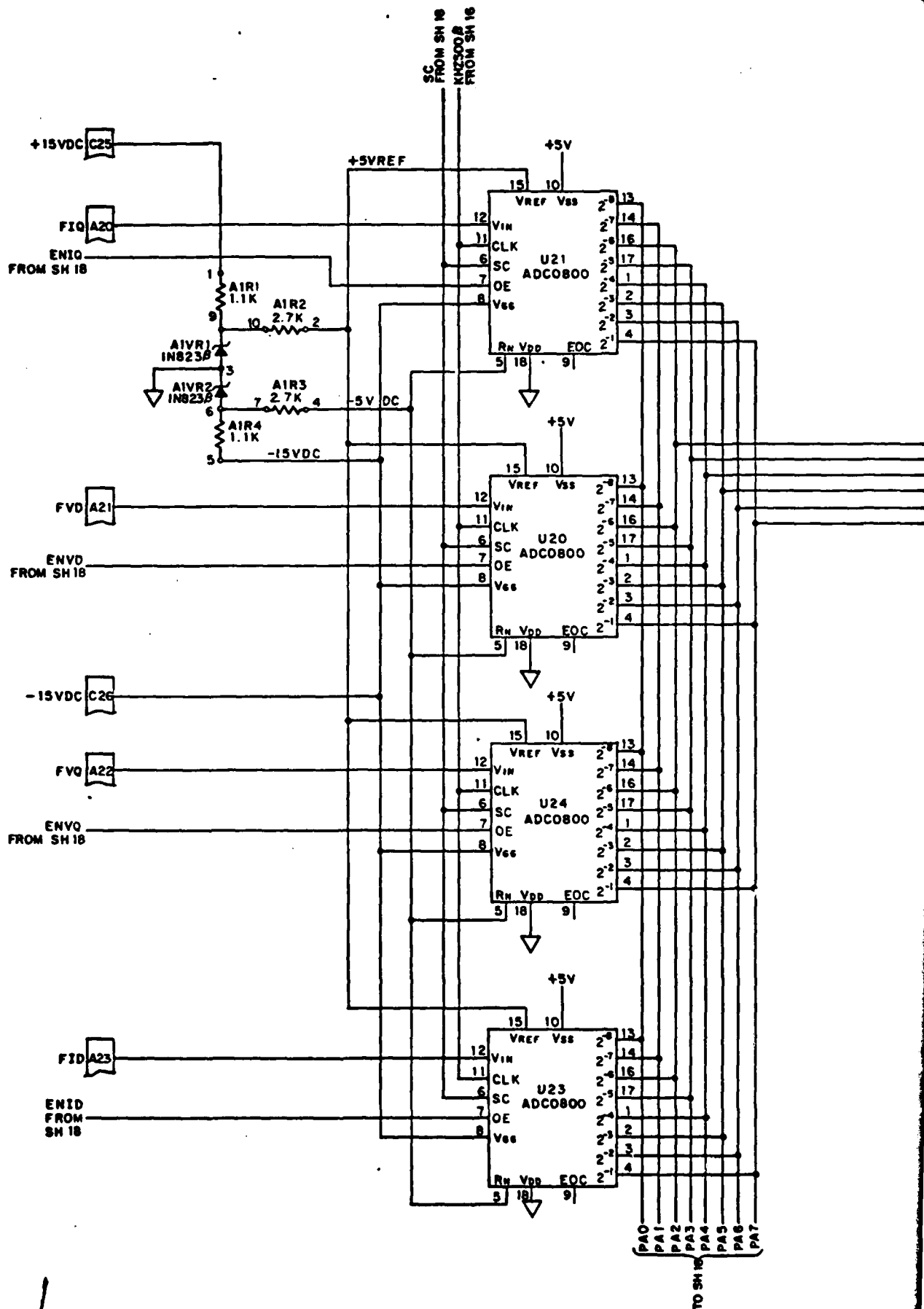


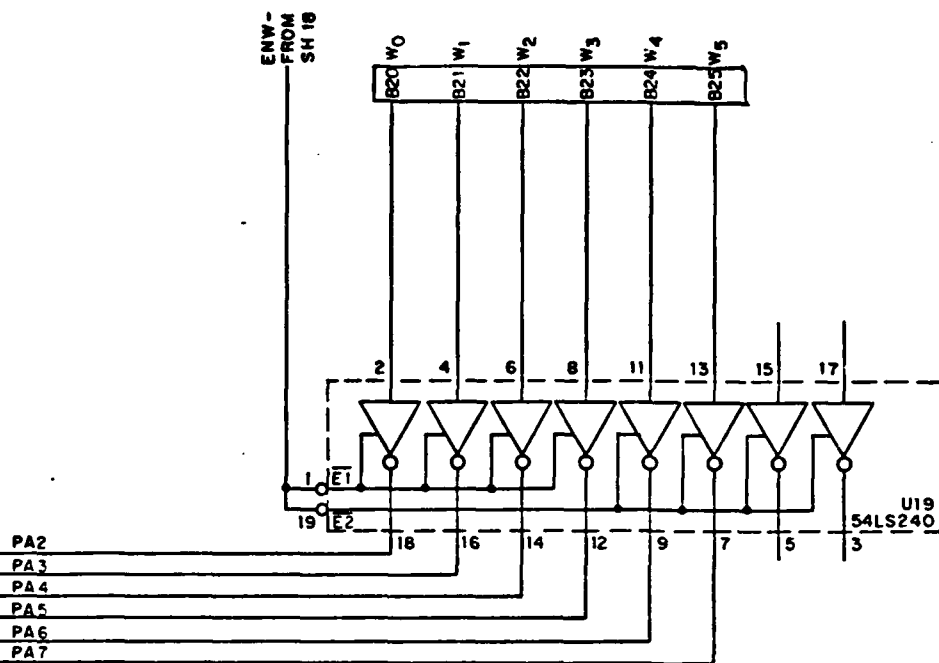


A5 F MPB

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE	REV	SHEET 18

2





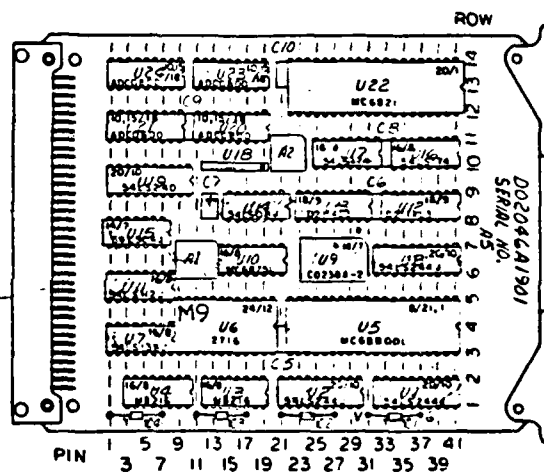
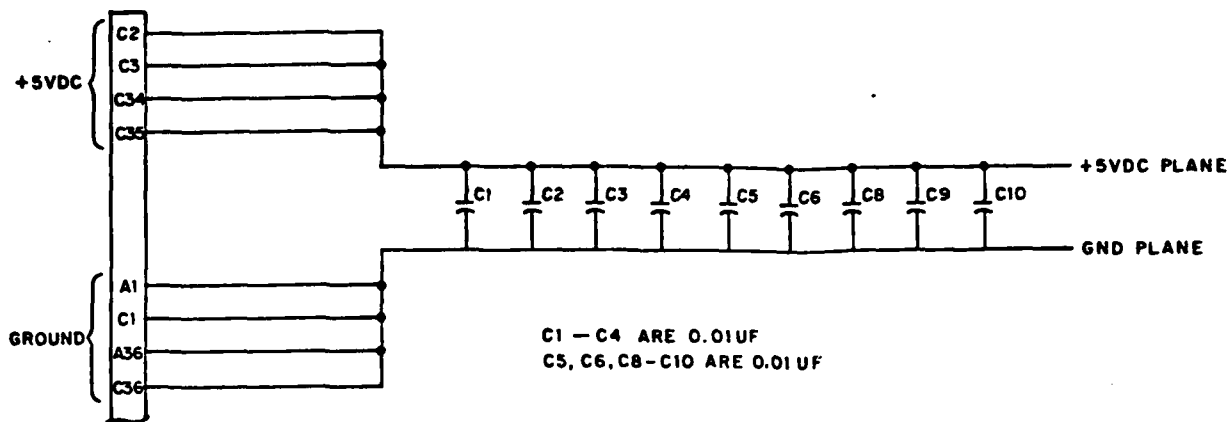
A5 F MPB

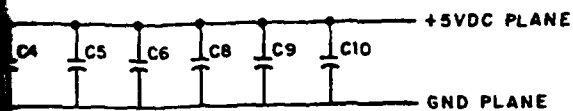
SHEET CODE IDENT NO. DWG NO.

D

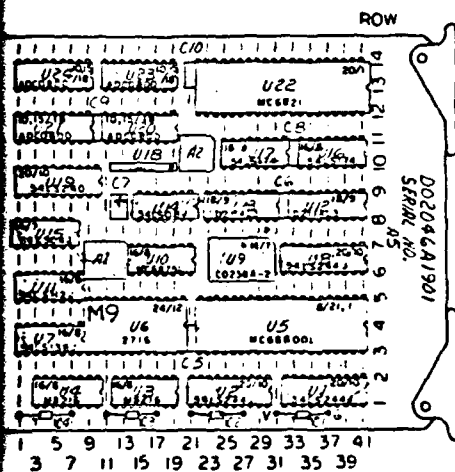
SCALE - 1 REV

1 SHEET





ALL 0.01UF
C10 ARE 0.01UF



2

A5	F	MPB
SIZE: CODE IDENT NO. DWG NO.		
D		
SCALE	REV	SHEET 20 OF 2

**Motor Controller Phase II
Task B Report**

Table of Contents

- I. Introduction
Summary
- II. Power Bridge Circuit Description
 - A. Power Switch Unit Description
 - B. Transistor Switch Module Description
 - C. "Catch" Diode Description
 - D. Switch Driver Circuit Description
 - 1. Input Isolation
 - 2. On-Drive Timer
 - 3. Current Limit Disable
 - 4. Over-Current Detector
 - 5. Off-Drive Timer
 - E. Low Voltage Power Supply
 - F. Shunt Resistor
- III. Power Bridge Test Results
- IV. Motor Controller Test Results

I. Introduction

This Phase II, Task B report describes the results of the final of three developmental stages of the advanced motor controller. Phase I developed the approach for the digital synthesis of 3-phase, variable frequency, variable voltage, sinusoidal waveforms using an 8-bit microprocessor. The Phase I effort also analysed a method of deriving an inductive motor air-gap-flux related signal useful for optimally controlling motor voltage with varying load. Phase II, Task A developed the design for the digital and microprocessor circuitry for synthesizing the 3-phase sinusoidal waveforms and the control circuitry for deriving the air-gap-flux related feedback signal. The Phase II, Task B effort developed a 3-phase power-bridge design and breadboard, combined the power-bridge with the waveform synthesis and feedback control circuitry developed in Task A and tested the combined system.

Summary

The power-bridge and PWM control circuits were combined and tested to the design goal of 7.5 hp with excellent performance. Motor voltage control under varying load using the air-gap-flux related feedback signal was demonstrated at frequencies above 15 Hz, however, control was less than optimal. Other motor-controller features including speed ramp-down under overload conditions, power switch over-current protection, and the maintenance of constant AC output voltage under conditions of varying DC input voltage were successfully tested.

II. Power Bridge Circuit Description

The three-phase power bridge circuit consists of 6 identical power switch units connected as shown in Figure 1. The switches are driven by the control circuits to produce pulse width modulated variable-frequency, variable-voltage, three phase power. The Phase II, Task A report describes the control circuits

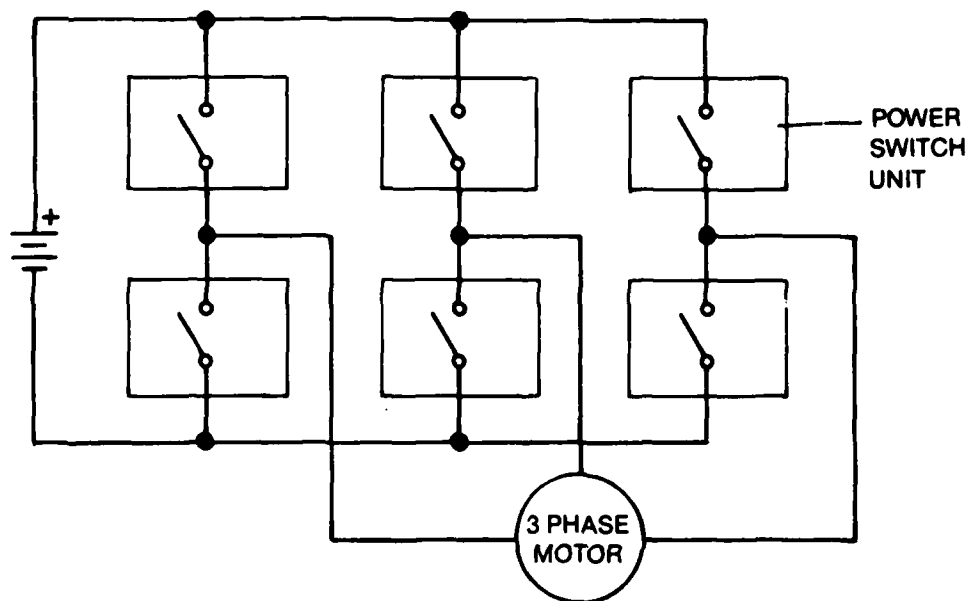


Figure 1. 3-Phase Power Bridge

and the method for generating the proper control signals for the power switches. Each of the power switches is optically isolated from the control signal source and each has individual current limit protection.

Power Switch Unit Description

The power switch units consist of a transistor switch module, a "catch" diode, a switch driver circuit, a low-voltage power supply, and a series current shunt resistor connected as shown in Figure 2.

The transistor switch module conducts the load (motor) current when in the on state and the "catch" diode conducts the load current when the switch module is off. The switch driver circuit provides the proper current pulse to drive the switch module on or off according to the control signal. The low voltage power supply provides isolated power for the switch driver circuit. The shunt provides a current proportional signal that is used for switch-module over current protection.

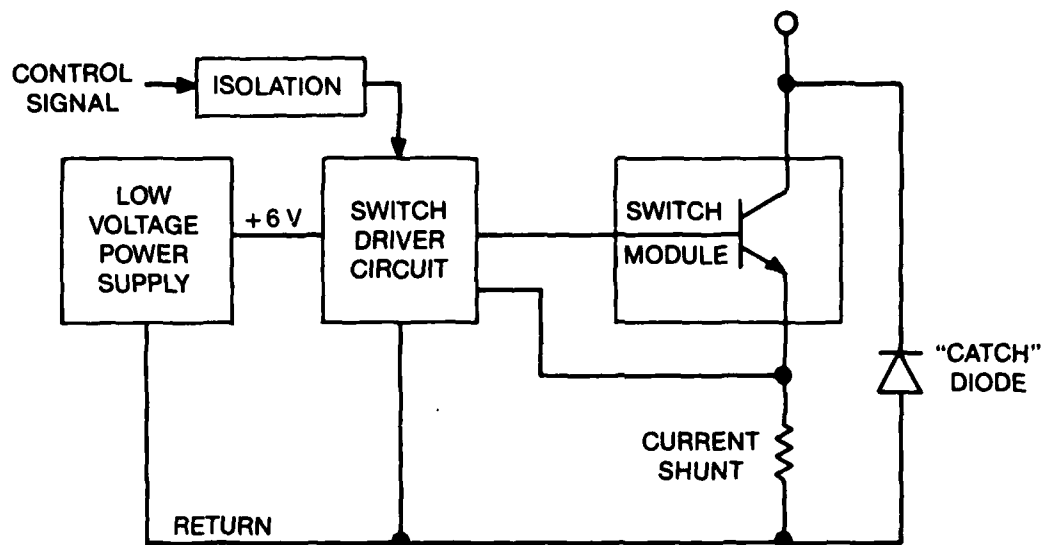


Figure 2. Power Switch Unit

Transistor Switch Module Description

The switch module consists of a matched set of transistors connected in a Darlington arrangement as shown in Figure 3. The transistors are matched by the manufacturer and housed in a 3.25 x 2.75 x 1.25 inch module. Each individual transistor is a PT-7511 (Power Tech, Inc.) (See Data Sheet, Appendix A). The switch module characteristics are as follows:

1. Collector-Emitter Breakdown Voltage $BV_{CEO} = 200 \text{ min at } I_C = 200 \text{ ma.}$
2. Emitter Cut-Off Current $I_{EBO} = 15\text{ma max at } V_{EB} = 8\text{V } I_{CB} = 0.$
3. Collector Cut-Off Current $I_{CIS} = 10\text{ma max at } V_{CB} = 200\text{V } R_{BE} = 0.$
4. D.C. Current Gain $h_{fe} = 100 \text{ min at } V_{CE} = 5\text{V } I_C = 250\text{A.}$
5. Base-Emitter Voltage $V_{BE} = 3.0\text{V max at } I_C = 250\text{A } V_{CE} = 4\text{V.}$

The modules used in the motor controller were tested by the manufacturer to the specification.
(See Test Data, Appendix A).

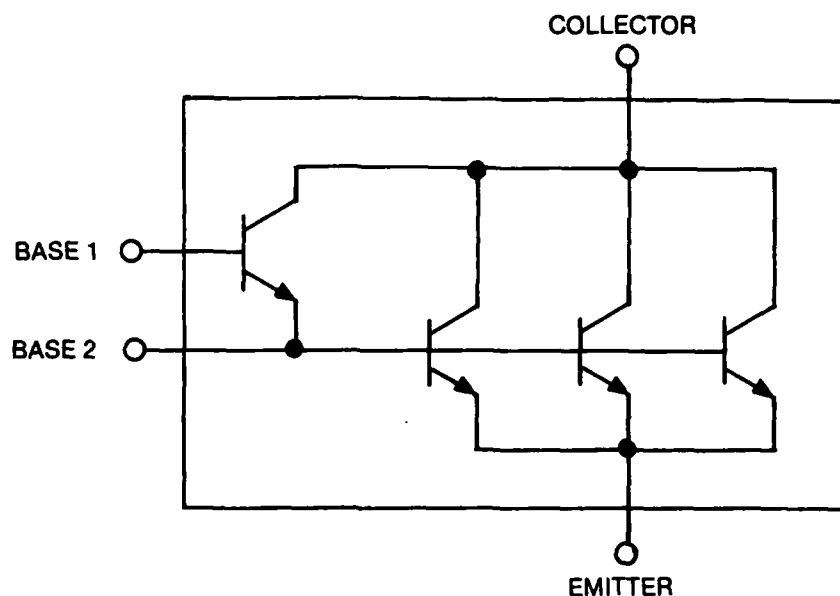


Figure 3. Switch Module

Diode Description

The "catch" diode provides a return path for the load current when the associated switch module is turned off. The manufacturer is Semtech Corp., part number SCSF4R. The important characteristics are as follows:

Peak Inverse Voltage	400V
Forward Voltage	$V_f = 1.40$ max at $I_f = 100A$
Recovery Time	$t_{rr} = 200$ nanoseconds
	(measured when rectifier recovers to 0.25 amp from a 0.5 amp forward current)

Reverse polarity diodes were used in the breadboard such that the cathode could be connected to the collector of its associated switch module using the heat sink as the conductor.

Switch Driver Circuit Description

The switch driver circuit inputs a control signal from the control electronics and outputs corresponding turn-on and turn-off drives to the base of its associated transistor switch module. A block diagram of the circuit is shown in Figure 4. The circuitry represented by the blocks to the left of the vertical dotted line in Figure 4 is located on a 4 x 4 inch Printed Wiring Assembly and the remaining circuitry is located on the switch module heat sink assembly. The circuitry operates as follows: (See Driver Board Schematic Diagram, Appendix B).

1. Input Isolation

Input isolation consists of an optical isolator, U1, that receives a TTL logic signal from the control circuit. A logic "0" turns the switch module on and a logic "1" turns it off. The optical isolator permits the power switch unit and the control circuit to operate at different reference voltages.

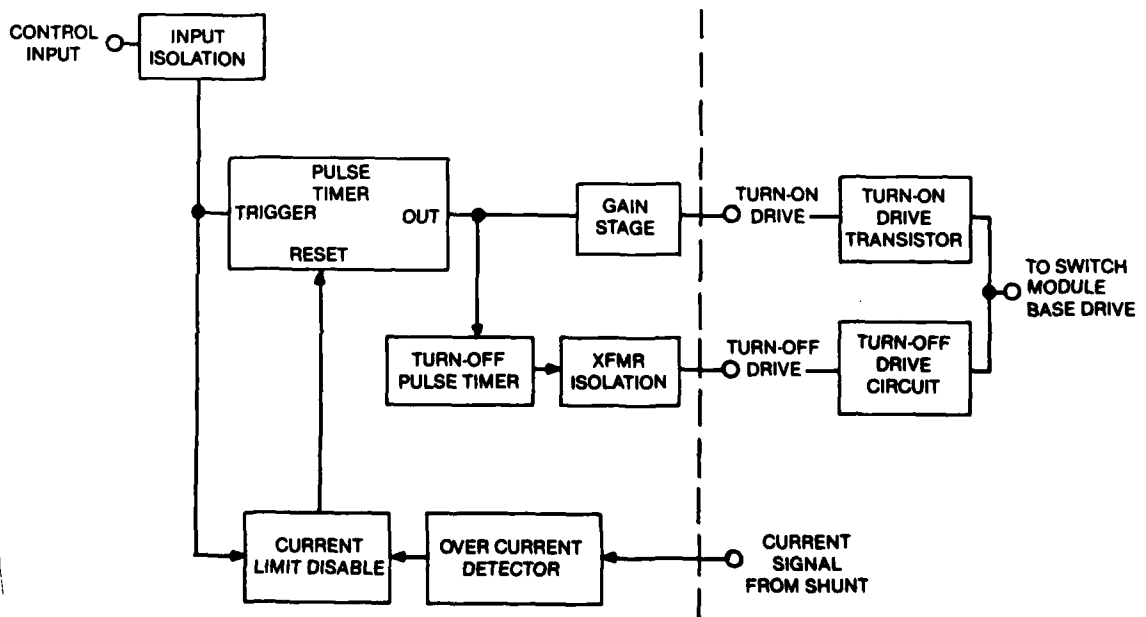


Figure 4. Switch Driver Block Diagram

2. On-Drive Timer

The pulse on-drive timer, U2A, is a protective circuit that limits the maximum on-time of the switch module. The timer turns off the switch module should the control circuit fail to do so within the prescribed time. The timer is triggered on by a logic "0" and reset by a logic "1" on the control input. The timer can also be reset by a signal from the overcurrent protection circuit.

3. Current Limit Disable

The current limit disable circuit disables the overcurrent protection circuit for the initial 6 microseconds of each power switch turn-on cycle. The circuit prevents the detection of overcurrents due to the recovery current of the "catch" diode or $L di/dt$ voltages in the shunt circuit that normally occur when a switch is turned-on. The overcurrent-disable time period is established by comparing the turn-on pulse from the input optical isolator, differentiated by C2 and R7, with a reference voltage generated by R19 and VR1. The output of the comparator U3A is wired in an "OR" arrangement with the current limit comparator U3B such that the overcurrent limit is disabled. The waveforms are diagrammed in Figure 5.

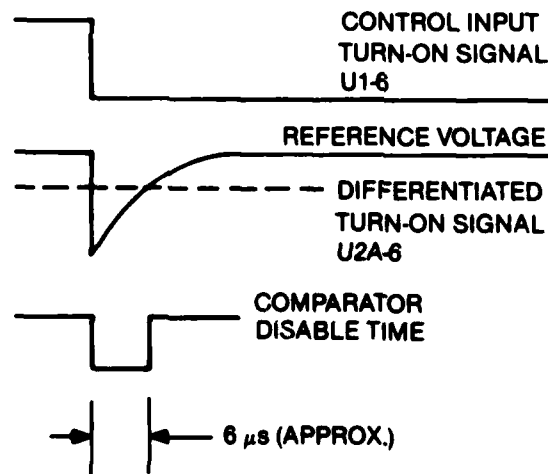


Figure 5. Current Disable Waveforms

4. Overcurrent Detector

The overcurrent detector compares the switch module current with a reference and resets the pulse time if the reference is exceeded. The voltage across the current shunt is sensed and amplified by a differential op amp, U6A, such that common mode rejection of the op amp reduces the effects of the $L di/dt$ induced voltage in the shunt circuit. (See schematic diagram, Appendix D). The current signal is compared with a reference, U6B, and if the reference is exceeded the pulse timer is reset through U3B. The shunt resistance is 0.002Ω and the differential op amp gain is 5 such that the reference voltage (pin 5 of U6) may be established by the following formula:

$$V = 0.01 I_{LIMIT} + 0.7$$

The reference is presently set at 2.00 volts for current limiting at approximately 125 amps.

5. Off-Drive Timer and Isolation Transformer

The overall purpose of the turn-off circuit is to minimize the turn-off time of the transistor switch modules by reverse biasing the base-emitter junctions to rapidly remove the stored charge in the base regions of the transistors. The off-drive timer supplies a $40 \mu s$ pulse that establishes the period during which the reverse voltage is applied to the switch module base-emitter junction. The timer is triggered by the termination of the turn-on drive and its output drives the primary winding of an isolation transformer T1. The isolated pulse from the secondary winding of T1 drives a second transformer (See Power Stage Schematic Diagram, Appendix B) that generates the reverse voltage. The actual base drive pulse is shown in photos 1 and 2.

Low Voltage Power Supply

Each of the 6 power switch units has an associated low-voltage power supply for powering the switch driver circuits. The supplies are standard commercial AC to DC types rated at 5VDC 3 amps.

Shunt Resistor

The shunt resistor conducts the power switch emitter current and is used for current sensing. The shunts are made of strips of monel metal cut to dimensions that produce a 0.002Ω resistance between terminals. The voltage generated across the shunts is connected to the differential op amp in the overcurrent detector circuit. Monel is used because of its low temperature coefficient.

Hardware Protective Circuits

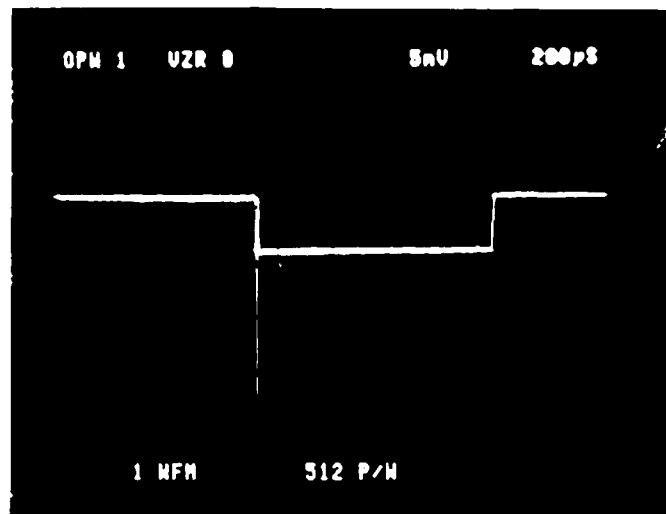
Each power switch unit has been designed to turn-on only from a command from the control circuits but will turn off under three conditions: (1) a control circuit command; (2) an overcurrent condition; or (3) in the event of no control circuit turn-off command within a prescribed period. Refer to the Switch Driver circuit description for details in the operation of these circuits. In addition, shoot-through protection has been designed in the control circuit such that a 26 microsecond off-period occurs between the time a power switch unit turns-off and its complementary power switch turns-on. This off-period insures that the conducting periods for complementary power switch units do not overlap.

III. Power Bridge Test Results

The Motor Controller Breadboard is shown in Photo 3 and the motor and dynamometer used for test purposes shown in Photo 4. The motor is a Franklin, oil-cooled type, rated at 7.5 hp, 600 rpm, 87V, 32 Hz. The dynamometer is rated at up to 125 hp. The power bridge was tested over the range of frequencies 6 to 60 Hz at various loads and was specifically tested to the motor ratings of 7.5 hp, 600 Rpm, 32 Hz.

Performance of the power bridge was excellent in all respects. Photos 5 through 12 show the phase to phase voltage and phase current waveforms under medium load at 15, 30, 45, and 60 Hz. Photo 13 shows the collector-emitter voltage of one of the switch modules and photo 14 shows the waveform of the current into one power switch unit including the "catch" diode current. Photo 15 shows the unfiltered phase to phase voltage for a high output voltage (wide pulse width) and photo 16 shows the same waveform for a low output voltage (narrow pulse width).

The switching waveform for the power switch modules were such that snubber networks were not required. The improvement in switching waveform would not compensate for the additional power dissipation and circuit complexity of snubber networks. This was a result of two design decisions: (a) to use relatively slow power transistors; and (2) to use a careful layout of the bus bars and filter capacitor to insure clamping of the bus voltage, in conjunction with fast "catch" diodes.

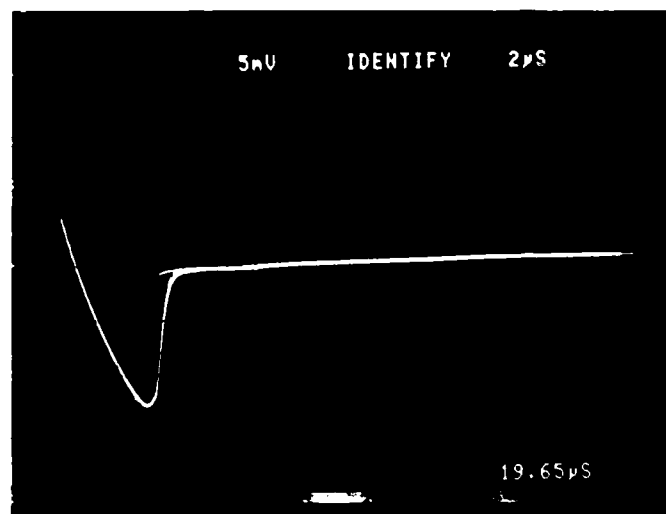


1

ON

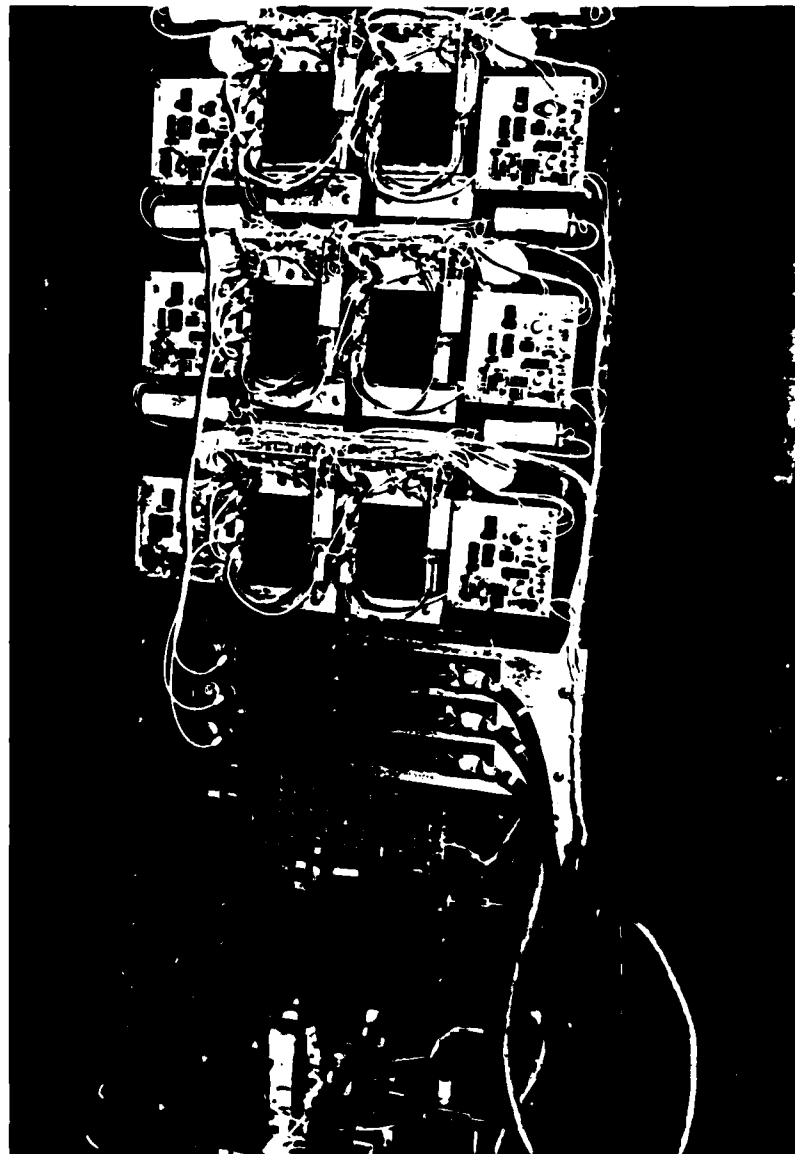
OFF

Switch Module Base Drive Current
2 Amp/Div



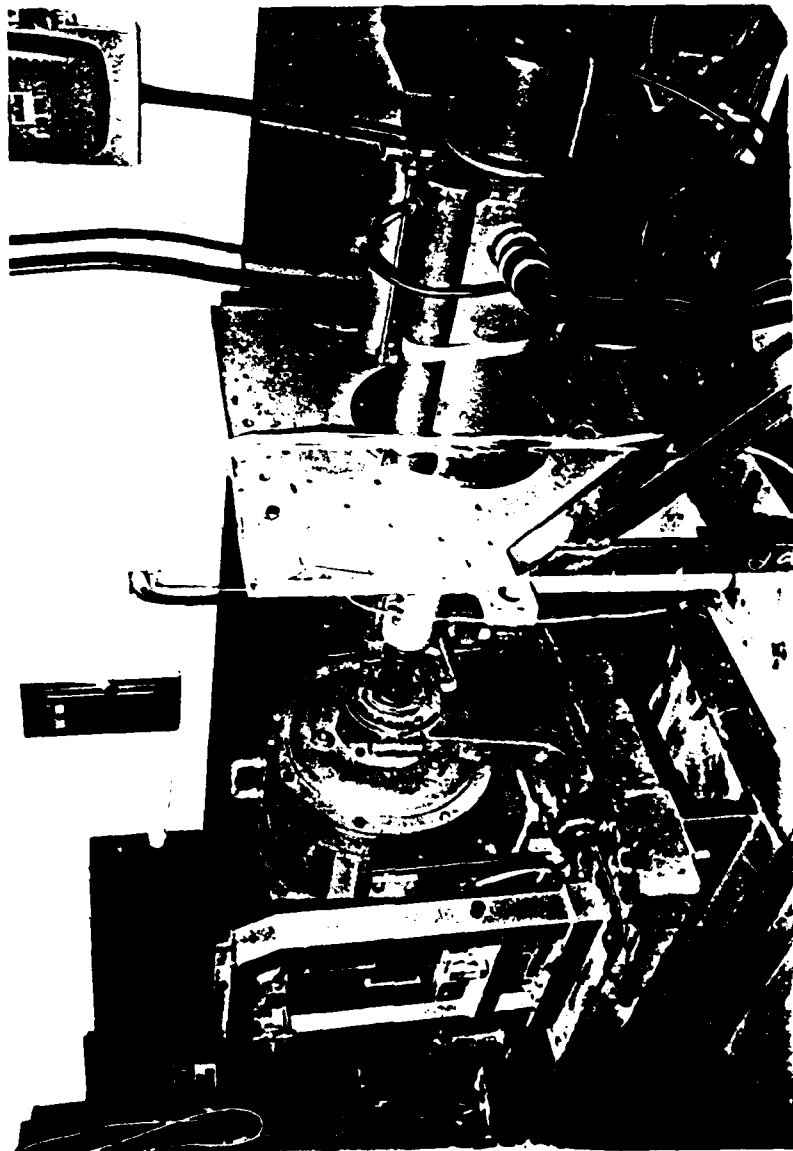
2

Switch Module Base Drive Expanded to
Show Turn Off Reverse Current
2 Amp/Div

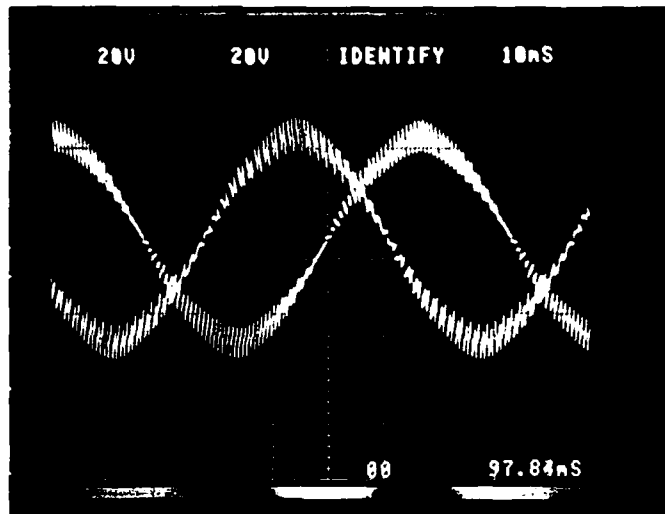


3

Motor Controller Breadboard

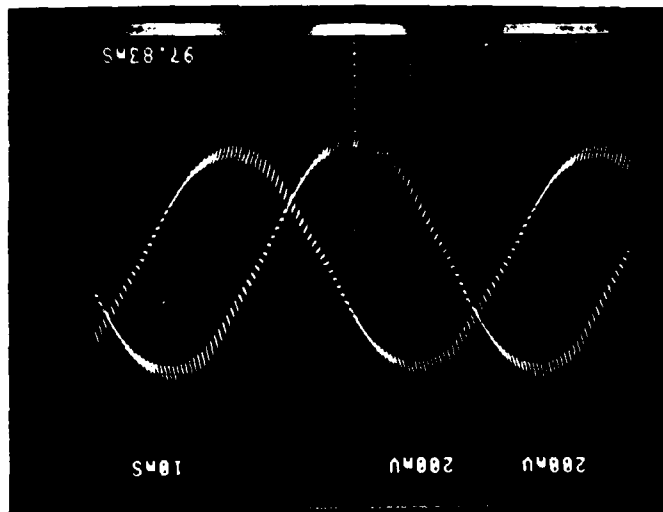


Motor and Dynamometer



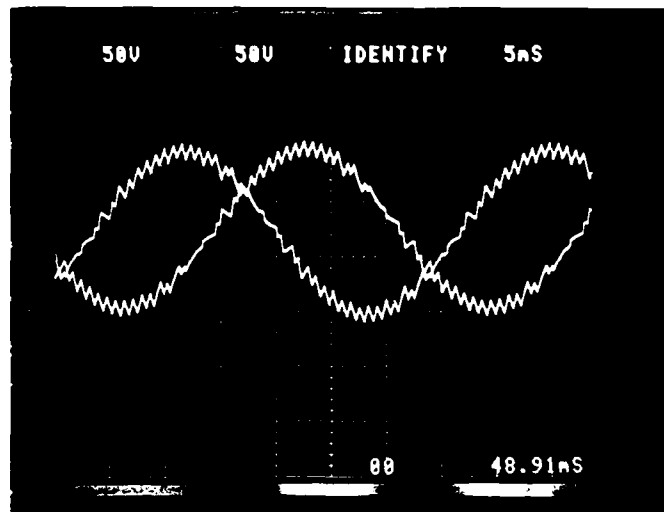
5

⌀ - ⌀ Voltage (2 Phase) 15 Hz (20 V/Div)
Filtered (100 Hz Cutoff)



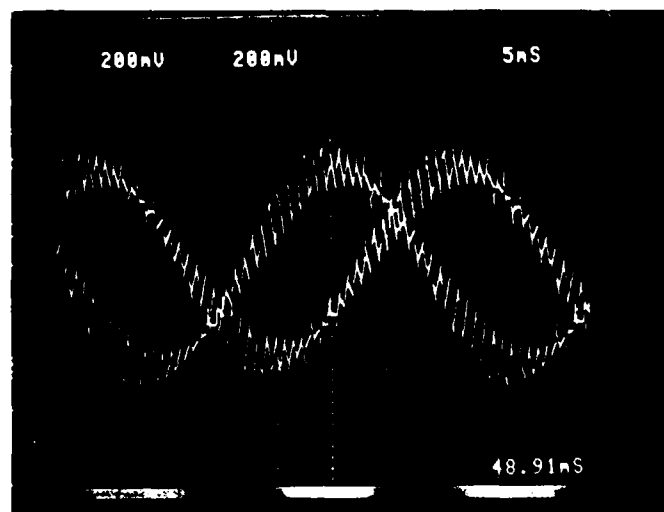
6

⌀ Current (2 Phase) 15 Hz (20 A/Div)
Unfiltered



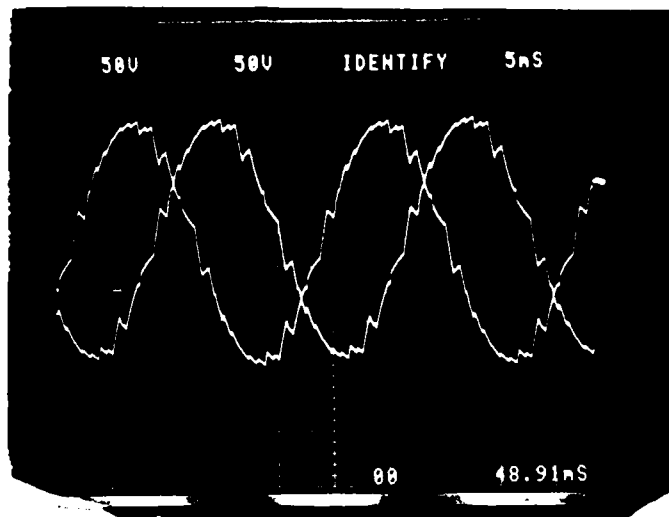
7

$\phi - \phi$ Voltage (2 Phase) 30 Hz (50 V/Div)
Filtered (100 Hz Cutoff)



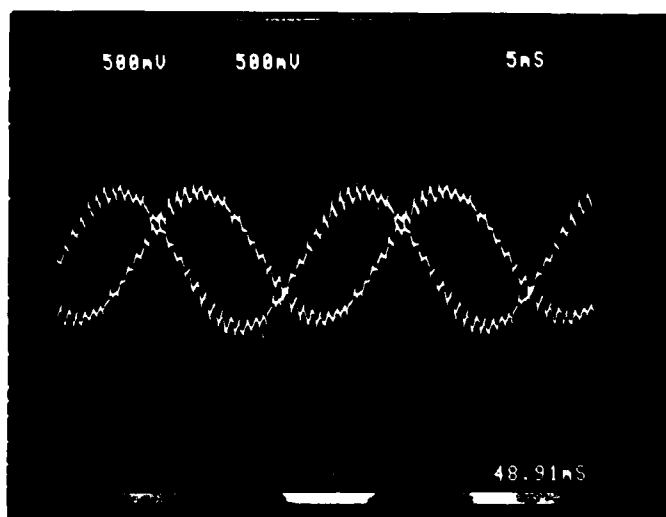
8

ϕ Current (2 Phases) 30 Hz (20 A/Div)
Unfiltered



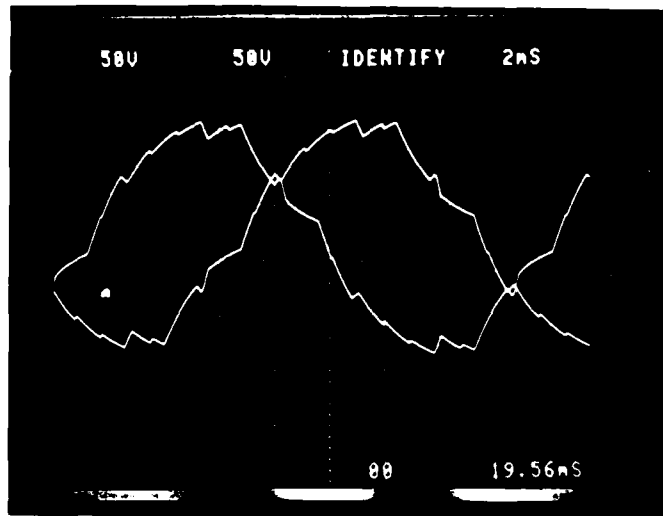
9

$\phi - \phi$ Voltage (2 Phase) 45 Hz (50 V/Div)
(Filtered — Hz Cutoff)



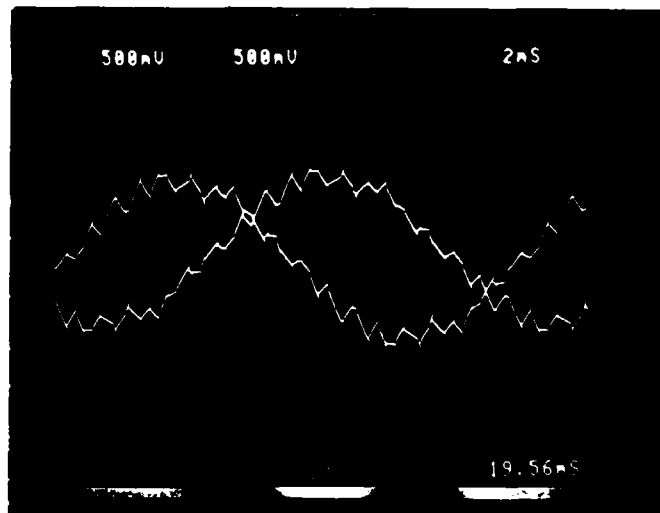
10

ϕ Current (2 Phases) 45 Hz (50 A/Div)



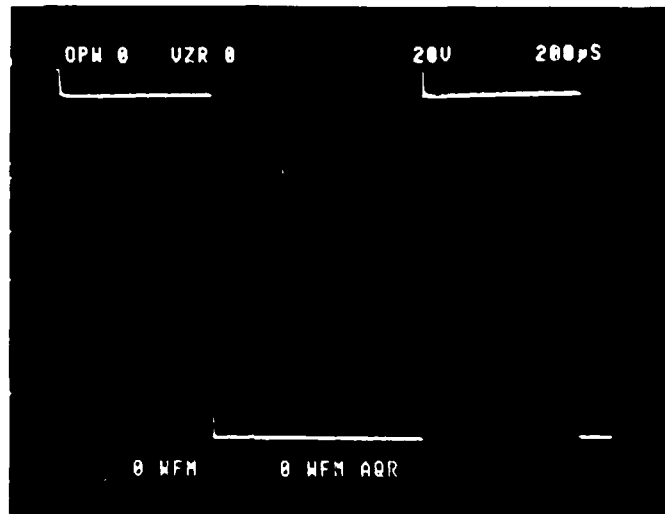
11

$\phi - \phi$ Voltage (2 Phase) 60 Hz (50 V/Div)
(Filtered — Hz Cutoff)



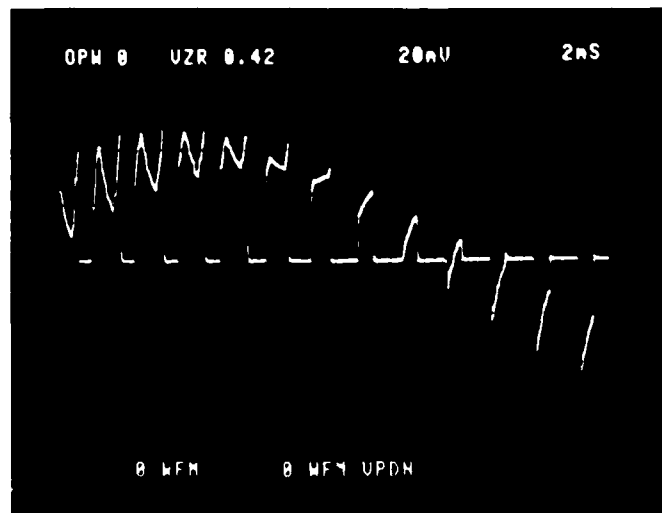
12

ϕ Current (2 Phase) 60 Hz (50 A/Div)



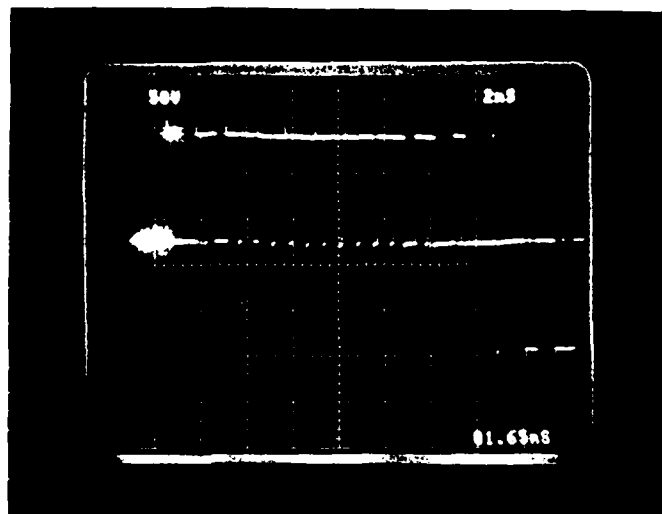
13

Vce (One Switch Module) (20 V/Div)



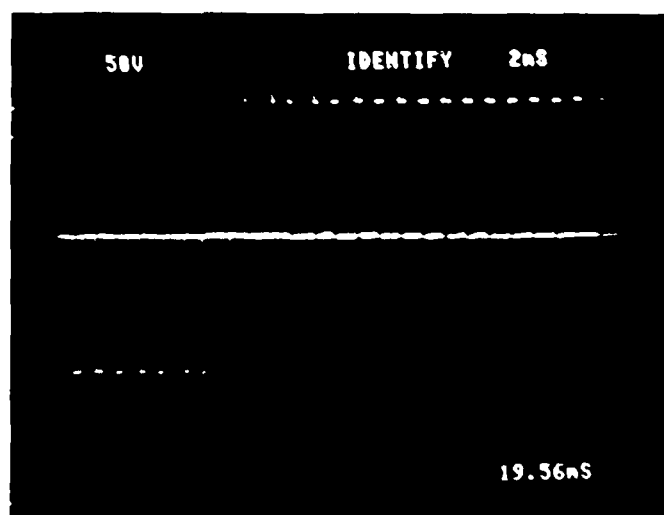
14

Power Switch Unit Collector Current Including
"Catch" Diode Current



15

Unfiltered $\phi - \phi$ Voltage (30 Hz) High Output Voltage



16

Unfiltered $\phi - \phi$ Voltage (30 Hz) Low Output Voltage

IV. Motor Controller Test Results

The power-bridge and feedback control circuits were connected to test the hardware and software required for motor-controller closed-loop operation. The resistance and inductance values for the test motor were provided by NSRDC as follows:

$$R_s = 0.0605\Omega$$

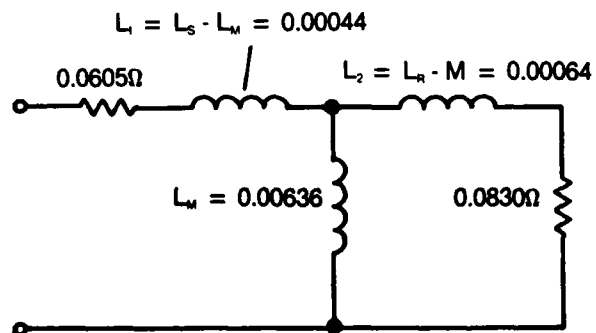
$$L_s = 0.00680\text{h}$$

$$R_r = 0.0830\Omega$$

$$L_r = 0.00700\text{h}$$

$$M = 0.00636\text{h}$$

The equivalent circuit values were derived as follows:



The control circuits and software were tailored to the test motor as follows:

- (1) The stator inductance value L (Refer to Phase I report) used for the feedback equation was calculated as follows:

$$L' = L_1 + L_M L_2 / (L_M + 2L_2)$$

$$L' = 0.00044 + (0.00636)(0.00064) / [0.00636 + (2)(0.00064)] = 9.73 \times 10^{-4} \text{h}$$

A table of K_i values was computed and entered into EPROM memory as follows:

$$K_i = 2\pi f L' \times \text{scaling factor}$$

A single K_i value (16 bits) for each integer frequency from 6 to 60 Hz was entered into the table. (PWM microprocessor program location FE06 Hex).

- (2) The direct and quadrature currents and voltages were scaled by installing the appropriate resistors in the scaling circuits located on the A4 PWA in the control electronics. Scaling is as follows:

$$V_d \text{ and } V_q \text{ scaled for } \pm 128 \text{V peak}$$

$$I_d \text{ and } I_q \text{ scaled for } \pm 256 \text{ amps peak}$$

- (3) The nominal (open-loop) output voltage table (V_{open}) was established for the test motor. The no-load motor voltage at which the motor begins to stall was measured at several excitation frequencies. A linear $V/F = \text{table}$ for the full range of frequencies was established from these measurements and the table was entered into the PWM processor memory.

An initial attempt to close the air gap flux feedback loop was partially successful. For excitation frequencies above 15 Hz at light to medium motor loads the feedback loop varied the motor voltage proportionally with load variations as expected. At high loads the feedback error calculated in software exceeded the maximum allowed by the software filter subroutine and an overflow occurred. At low excitation frequencies 6 to 15 Hz the feedback loop did not correctly respond to the load variations.

Several minor changes were made to the feedback processor software to eliminate the overflow problems. The software flow was rearranged and a variable gain (table loop-up) algorithm was added to the feedback error signal, E^* , as shown in Figure 6. Implementation of the selectable gain algorithm on the feedback error signal resulted in a smaller magnitude error signal in the filter subroutine, eliminating the overflow problem. The variable gain feature also added flexibility in establishing the optimal voltage boost for a given feedback error.

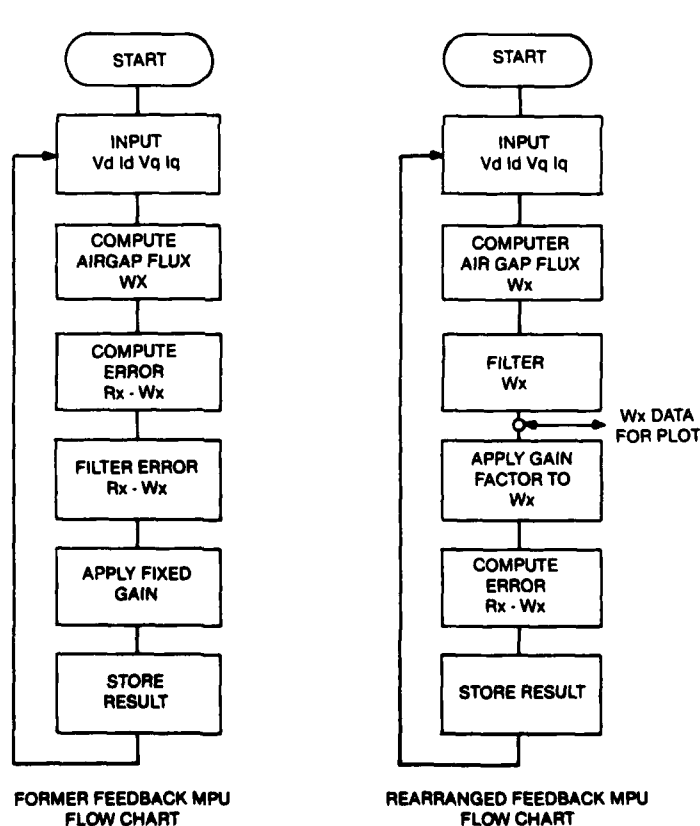


Figure 6. Revision to Feedback Flowchart

Closed-loop testing with these software modifications enabled adequate operation at excitation frequencies above 15 Hz. The motor voltage responded proportionally with load variations over the entire load range. Motor voltage regulation for a given load, however, tended to be on the underexcited side of the optimal motor excitation. Attempts at achieving optimal motor excitation voltage resulted in a runaway condition, where the excitation voltage increased to its maximum regardless of the load. A set of data was taken to examine the flux feedback signal at various excitation frequencies from 6 to 60 Hz and various motor loads. The data is tabulated and plotted in Figures 7 through 12. The air gap flux related signal, W_{λ} , was recorded from feedback microprocessor program location "FABE" hex.

The feedback control loop as presently implemented requires the W_{λ} versus motor voltage plot to have a negative slope. Regulation occurs in this negative slope area to the left of the minimum "valley" W_{λ} value shown on the W_{λ} plot. The calculated W_{λ} curves for frequencies above 15 Hz have the necessary negative slope areas that allow the loop to regulate. Below 15 Hz, the curves do not have the required shape. Regulation was attained at frequencies above 15 Hz and the motor voltage increased with load as expected. Regulation was not achieved at excitation frequencies below 15 Hz. In all cases, the motor was slightly under-excited during closed-loop regulation. The approximate optimal regulation points on the W_{λ} versus Motor Voltage plots are shown by a small circle on each of the plots. These points were estimated from the attached data through an interpretation of the slip speed and input power. Photos 17 through 24 show the filtered feedback signals used to calculate W_{λ} . Pictures of V_d , V_q , I_d , and I_q at 4 frequencies are shown. In all cases the "direct" quantity leads the "quadrature" quantity. The waveform photos were taken at the A/D Converter inputs in the control circuit.

Problems that affected proper motor voltage regulation using the air-gap-flux related signal included: (1) obvious distortion in the filtered I_d , I_q , V_d , and V_q signals at low frequencies (see photos 23 and 24); (2) general difficulty with the complexity of tailoring the feedback equation to the test motor parameters; and (3) control problems related to the double valued nature of the air gap flux related feedback signal.

In order to test the various features of the motor controller without relying on the Wx error quantity, the motor controller software was modified to provide a constant voltage per frequency output in the closed-loop mode. This change involved installing a new open-loop voltage table in the PWM processor software and using a constant for determining the PWM Scale factor rather than the error signal from the feedback microprocessor. (Constant is located at DD46 and DD47 Hex). The feedback microprocessor software was also rearranged slightly to rectify and filter the motor direct current value previously used in the air gap flux calculation and pass the result to the PWM microprocessor. The rectified current value is used by the PWM processor for software overload protection in the same manner as the Wx error signal was previously used. Software listings containing these changes are provided.

The overload, ramp-down feature was tested and the overload threshold set to approximately 100 amps peak motor current. Motor controller operation using toggle switches for motor speed changing and direction reversing rather than using the CRT Terminal was also demonstrated.

OUTPUT	INPUT		Wk. at "FASE"		MOTOR RPM	DYNAMOMETER %	FLD. EXCIT.	GHE DATA			
	Vac PEAK	Iac PEAK	Voc	Ioc Amper	HEX	DEC					
25	45	125.6	3.4	0750	118	2	}	LIGHT LOAD			
20	37	125.6	2.4	0490	117	2					
17.5	32	125.7	2.0	0329	116	2					
15	27	125.8	1.8	0241	114	2	}	LIGHT LOAD			
12.5	25	125.8	1.6	0196	112	2					
10	22	125.8	1.4	0118	107	2					
7.5	20	125.9	1.2	00A3	091	2	}	LIGHT LOAD			
6	21	125.9	1.2	0085	044	2					
30	60	125.7	5.8	0800	116	6	}	MEDIUM LOAD			
25	47.5	125.7	4.4	0766	114	6					
20	40	125.3	3.6	0480	111	6					
17.5	37.5	125.3	3.2	0380	108	6	}	MEDIUM LOAD			
15	40	125.4	3.0	0296	103	6					
12.5	37.5	125.4	2.8	01C0	091	6					
10	35	125.5	2.6	0160	060	6	}	MEDIUM LOAD			
30	65	124.8	8.0	0C40	110	10	}	HEAVY LOAD			
25	55	124.8	6.4	0785	107	10					
20	50	124.9	5.8	04E7	098	10					
17.5	50	124.9	5.4	03D0	087	10	}	HEAVY LOAD			
15	50	124.9	5.2	0320	070	10					
12.5	50	125.1	4.4	0286	040	10					

FIGURE 7A

OUTPUT		INPUT		Wx at 'FABE'		MOTDR	DYNAMOMETER	FLD EXCIT.		10 HZ DATA			
VAC PEAK	IAC PEAK	Vdc	Ioc AMS	HEX	DEC	RPM	%						
50	80	124.7	8.6	1F9A		198	2						
40	50	125.0	4.6	0E90		198	2						
30	37.5	125.2	3.2	0730		195	2						
25	32	125.4	2.6	04A0		193	2						
20	27.5	125.4	2.2	0300		190	2						
17.5	26	125.5	2.2	026A		187	2						
15	25	125.5	2.0	01E8		181	2						
12.5	26	125.6	2.0	0180		165	2						
10	27.5	125.6	2.0	0186		119	2						
7.5	26	125.6	1.4	0125		STALL	2						
40	52	124.8	6.8	0E38		193	6						
30	45	124.9	5.6	07CC		186	6						
25	42	124.9	5.2	0567		177	6						
20	47.5	124.9	5.2	044C		162	6						
17.5	47.5	124.9	5.2	0425		135	6						
15	50	125.0	5.0	0445		98	6						
12.5	47.5	125.1	3.8	038F		48	6						
50	75	123.8	15.4	1890		186	10						
40	65	124.0	12.2	0E8A		179	10						
30	67	124.0	11.6	0A3D		158	10						
27.5	72	124.0	11.6	0900		135	10						
25	75	124.0	11.4	090F		118	10						
20	70	124.3	8.8	086E		59	10						

FIGURE 7B

OUTPUT		INPUT		Wk at "FABE"		MOTOR	DYNAMOMETER FLD. EXCT.	20 HZ DATA			
Vac PEAK	Iac PEAK	Vdc	Idc AMPS	HEX	DEC	RPM	%				
60	44	124.9	6.0	1398		394	2	}			
50	37.5	124.9	5.2	0090		392	2				
40	35	125.0	4.6	0824		386	2				
30	35	125.0	4.4	0560		375	2		}		
25	35	125.1	4.4	0485		356	2				
20	42	125.0	4.6	0500		293	2				
17.5	45	125.1	4.4	0790		152	2				
60	55	123.9	14.2	1408		378	6	}			
50	60	123.8	14.0	1115		366	6				
40	65	123.7	14.2	1097		338	6		}		
37.5	75	123.6	14.8	1400		290	6				
35	85	123.7	14.2	1850		175	6				
80	82	122.0	31.2	280A		370	10	}			
70	82	122.0	31.0	2660		362	10				
68	85	121.8	31.8	2590		354	10		}		
65	87.5	121.9	31.8	2560		352	10				
60	95	121.8	32.2	2490		337	10				
57.5	100	121.8	32.2	2770		318	10				

HEAVY LOAD

MEDIUM LOAD

LIGHT LOAD

FIGURE 7C

27

FIGURE 7C

WORK SHEET
WESTINGHOUSE FORM 8295 E

FIGURE 7 D

WORK SHEET
WESTINGHOUSE FORM 8295 E

OUTPUT		INPUT		Wk at "FAIR"	MOTR	DYNAMOMETER RD. EXCIT.	45 HZ DATA
Vac PEAK	IAC PEAK	Voc	Ixc AMPS	HEX	DEC	RPM	%
100	45	123.8	13.8	26AE		886	2
80	40	123.8	13.2	1A80		877	2
70	40	123.8	13.0	16D0		869	2
60	42	123.8	13.0	1520		851	2
55	47	123.7	13.2	1630		833	2
50	55	123.7	13.4	1A40		798	2
100	45	123.0	19.6	2380		1174	2
90	45	123.0	19.4	1FD0		1166	2
80	47	123.0	19.4	1E68		1154	2
70	55	123.0	19.8	2080		1129	2
60	75			4800		833	2
* INSUFFICIENT VOLTAGE AVAILABLE FOR FULLY LOADING THE MOTOR AT FREQUENCIES > 32 HZ.							

} LIGHT LOAD *

} LIGHT LOAD *

GO HZ DATA

82

WX CALCULATED

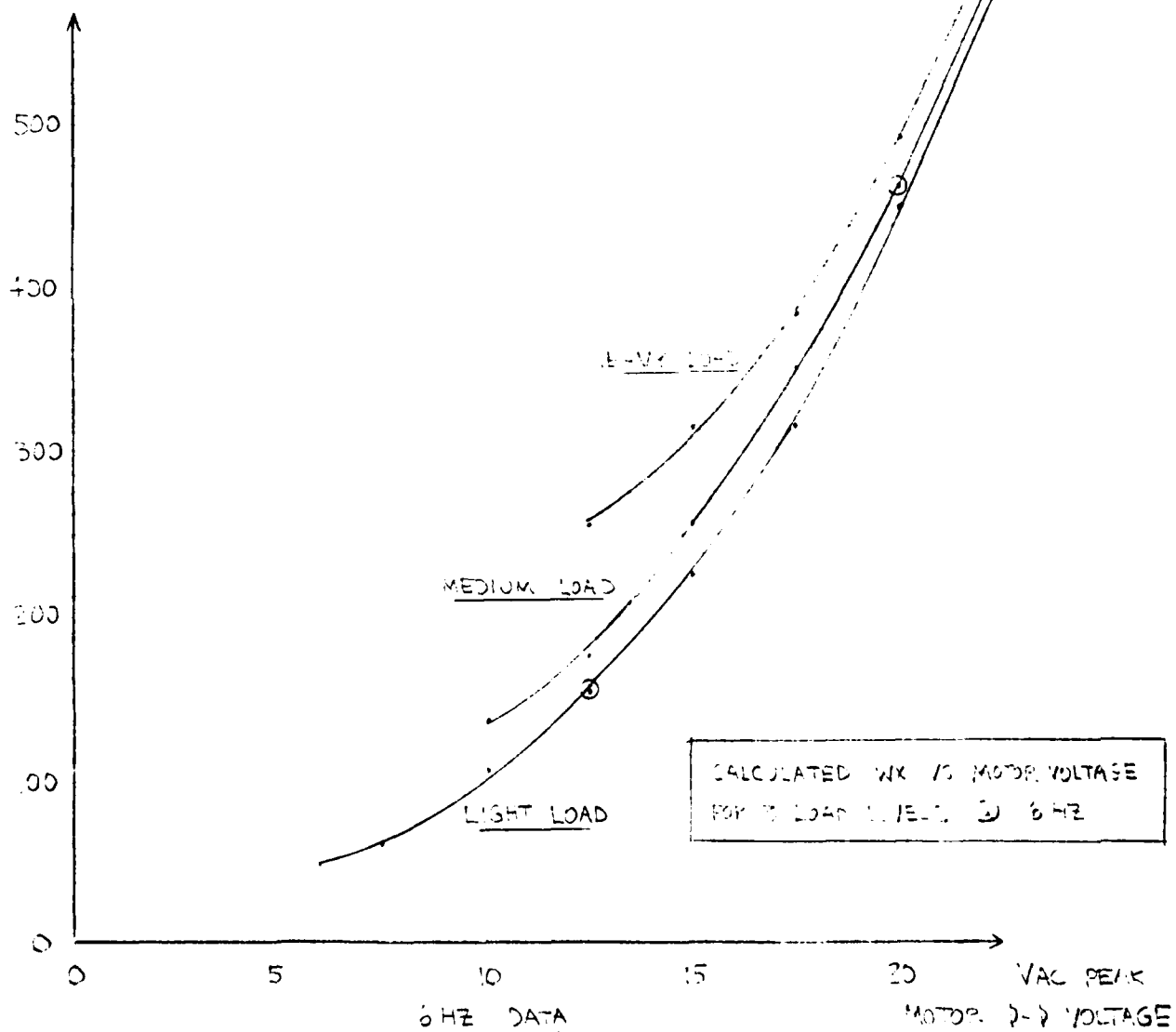
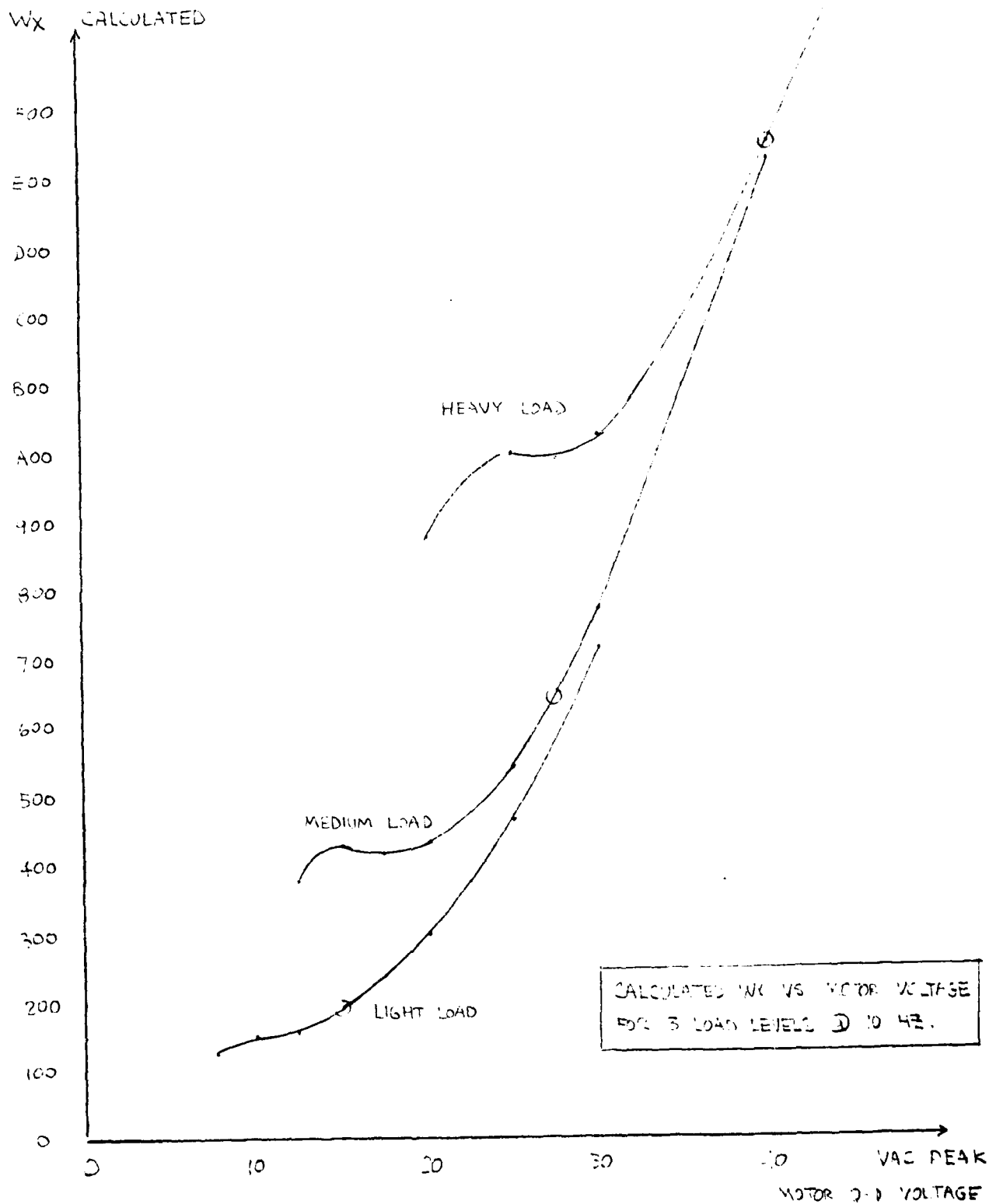


Figure 8



10 HZ DATA

FIGURE 9

31

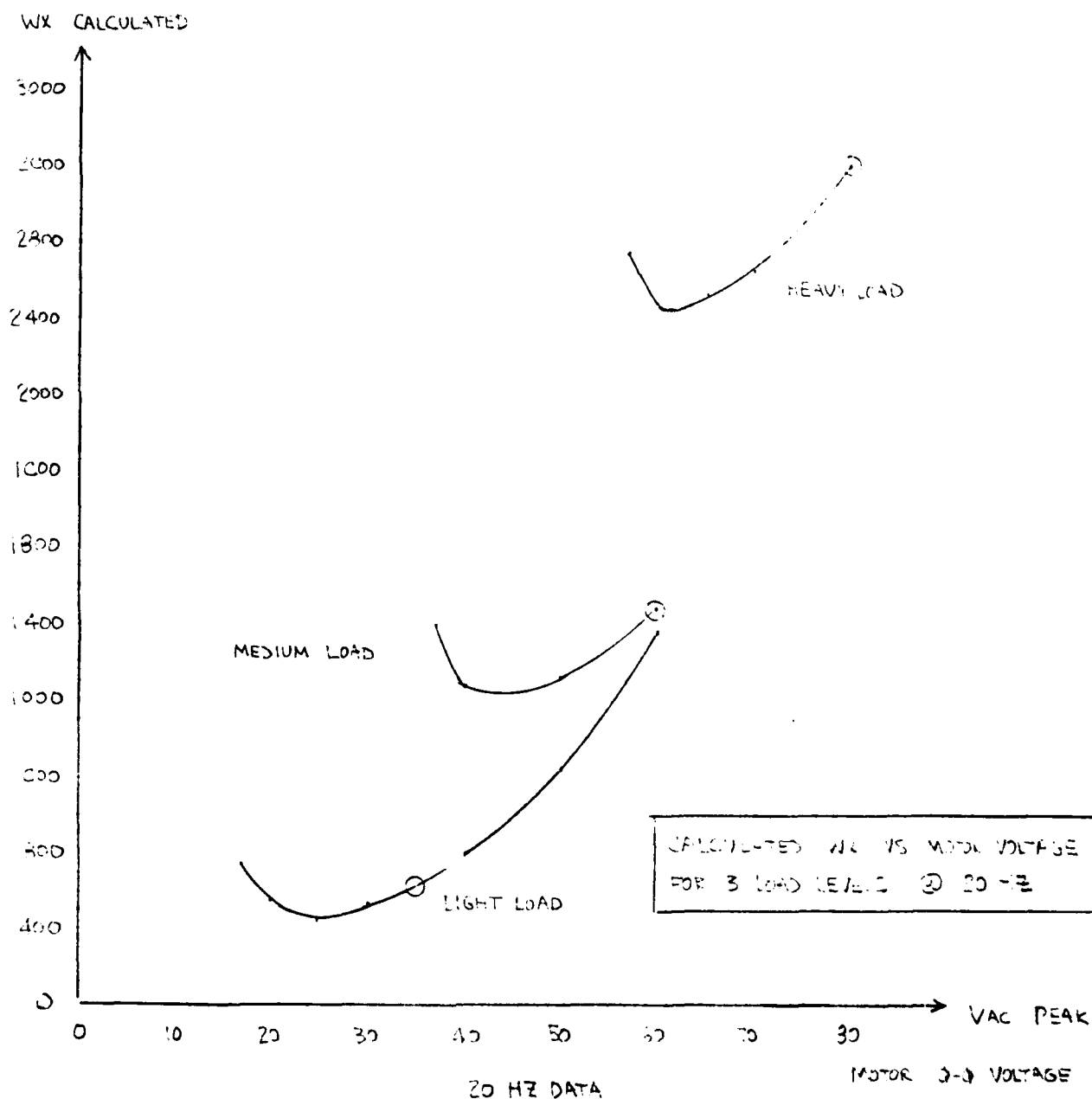
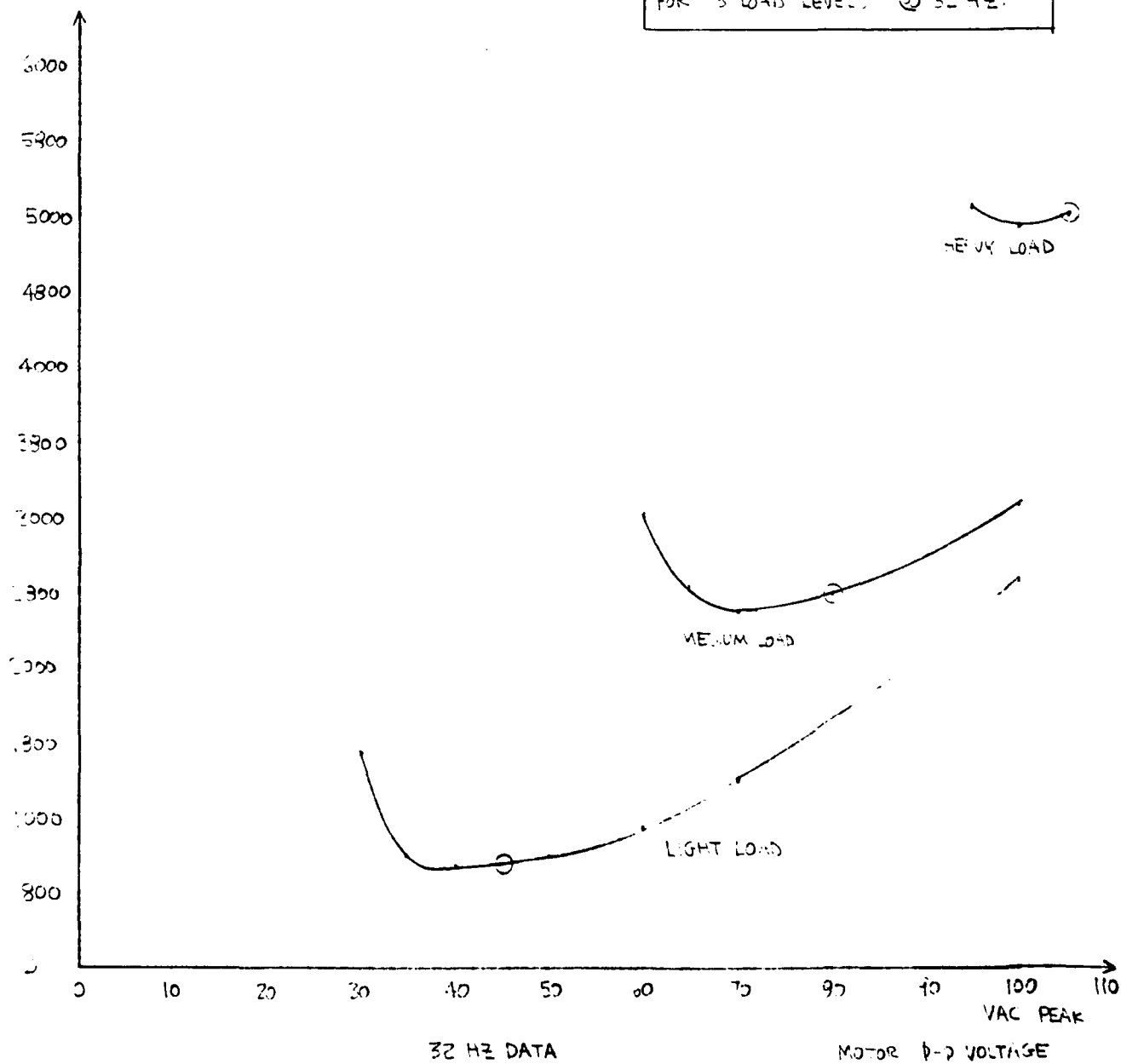


FIGURE 10

WX CALCULATED

CALCULATED WX vs. MOTOR VOLTAGE
FOR 3 LOAD LEVELS @ 32 HZ.



32 HZ DATA

MOTOR V-V VOLTAGE

Figure 11

CALCULATED WK VS MOTOR VOLTAGE
FOR LIGHT LOAD @ 45HZ & 60HZ

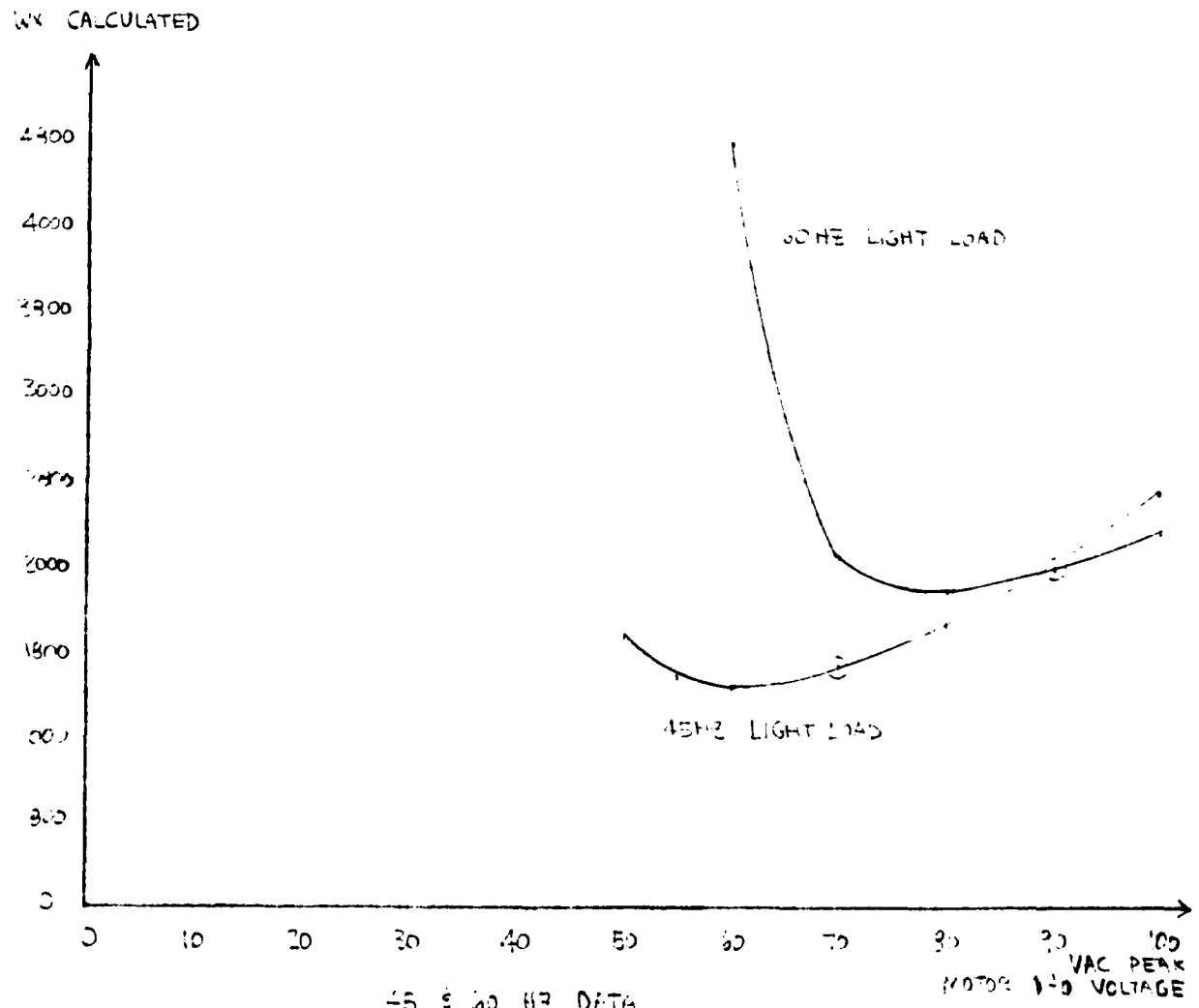
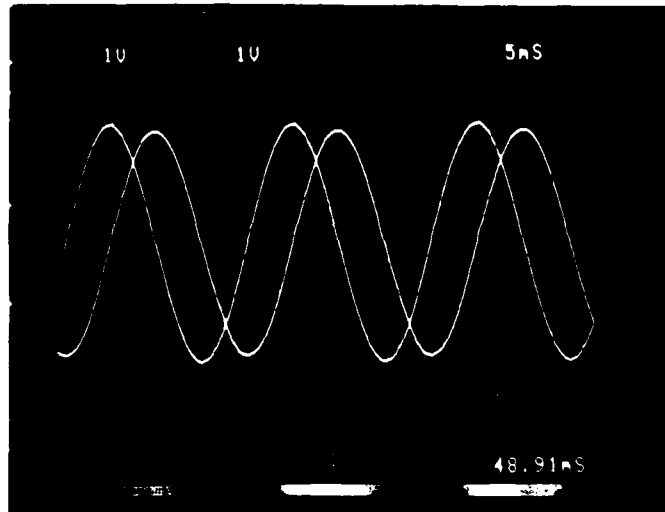
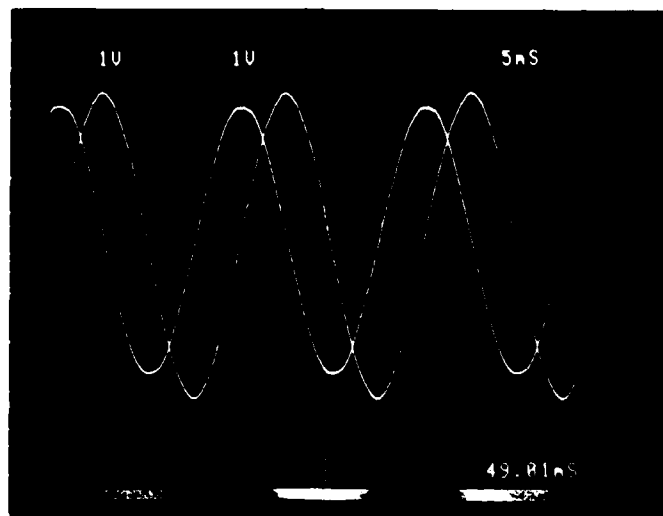


FIGURE 12
34



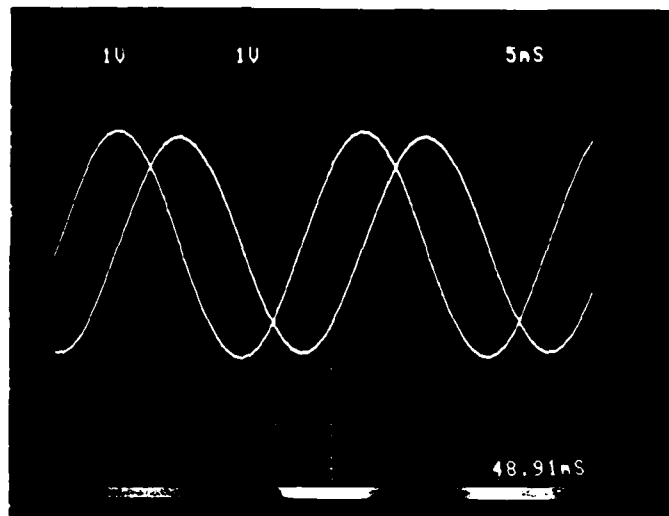
17

Vd, Vq at 60 Hz



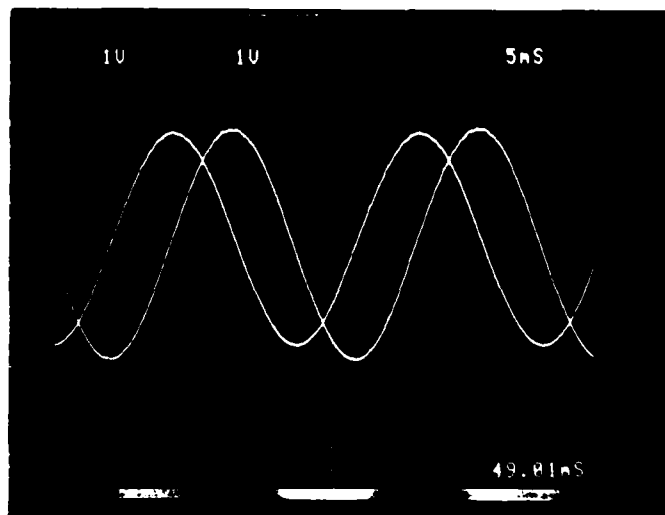
18

Id, Iq at 60 Hz



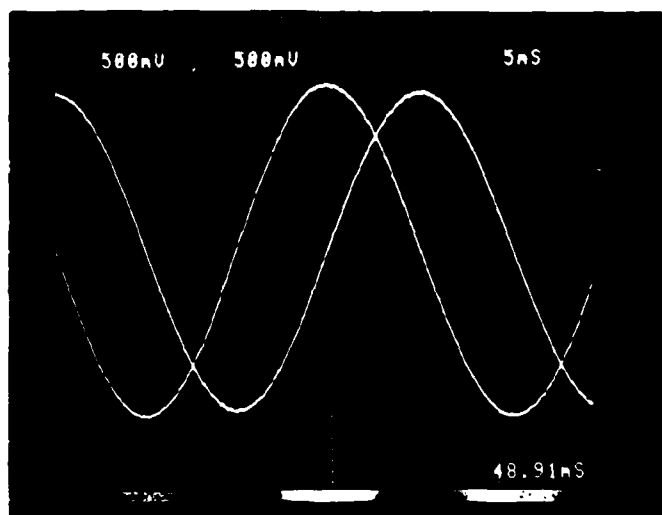
19

V_d, V_q at 45 Hz



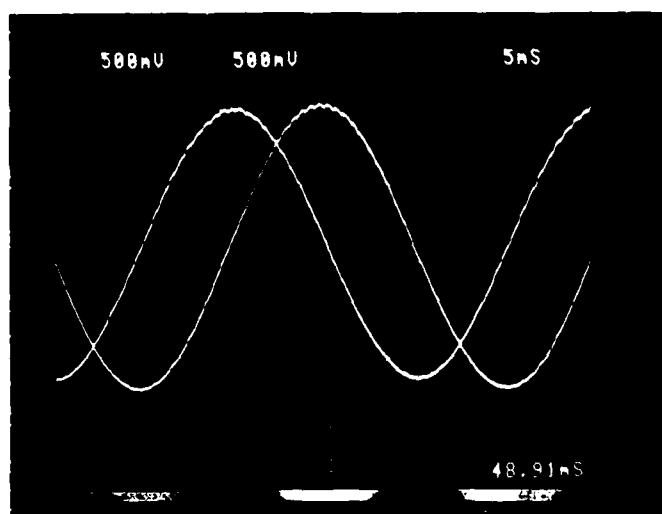
20

I_d, I_q at 45 Hz



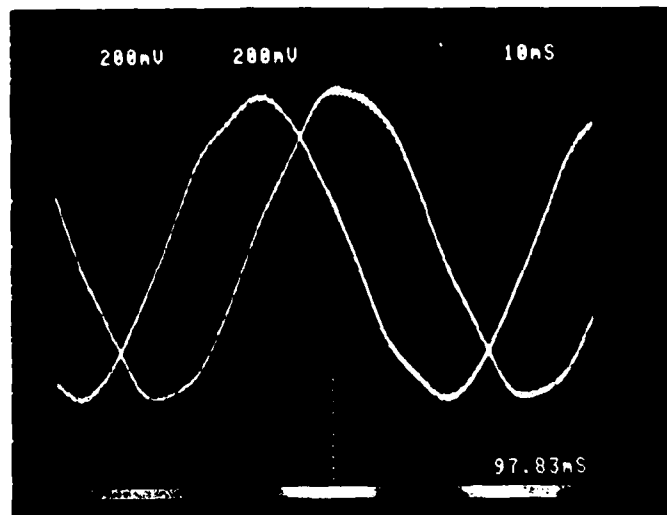
21

Vd, Vq at 30 Hz



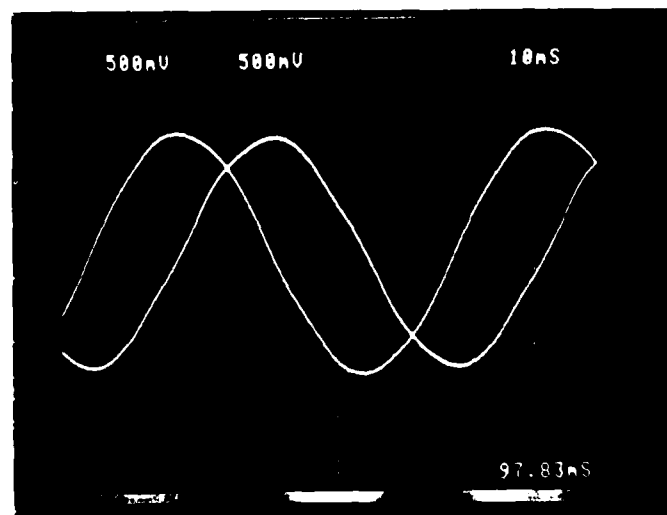
22

Id, Iq at 30 Hz



23

Vd, Vq at 15 Hz



24

Id, Iq at 15 Hz

APPENDIX A

90 AMPERES

PT-7511

SILICON NPN TRANSISTOR**MAXIMUM RATINGS**

	SYMBOL	PT-7511
Collector-Base Voltage	V_{CBO}	200V
Collector-Emitter Voltage	V_{CEO}	200V
Emitter-Base Voltage	V_{EBO}	10V
Peak Collector Current	I_{CM}^*	90A
D.C. Collector Current	I_C	50A
Power Dissipation at 25°C Case Temperature	P_D	350W
Power Dissipation at 100°C Case Temperature	P_D	200W
Operating Junction Temperature Range	T_J	-65 to 200°C
Storage Temperature Range	T_A	-65 to 200°C
Thermal Resistance	θ_{JC}	0.5° C/W
Package		TO-63

ELECTRICAL CHARACTERISTICS (at 25°C unless noted)

TEST	SYMBOL	LIMITS		UNIT	TEST CONDITIONS
		PT-7511			
		MIN.	MAX.		
D.C. Current Gain*	h_{FE}	10	40		$I_C=50A, V_{CE}=2V$
D.C. Current Gain*	h_{FE}	5	—		$I_C=90A, V_{CE}=4V$
Collector Saturation Voltage*	$V_{CE(sat)}$	—	0.6	V	$I_C=50A, I_B=5A$
Collector Saturation Voltage*	$V_{CE(sat)}$	—	1.5	V	$I_C=90A, I_B=18A$
Base Emitter Voltage*	V_{BE}	—	1.5	V	$I_C=50A, V_{CE}=2V$
Base Emitter Voltage*	V_{BE}	—	2.5	V	$I_C=90A, V_{CE}=4V$
Collector-Emitter Breakdown Voltage*	$V_{CEO(sus)}$	200	—	V	$I_C=200mA, I_B=0$
Collector Cut-off Current	I_{CBO}	—	2.0	mA	$V_{CB}=200V, I_{EB}=0$
Collector Cut-off Current @ 150°C	I_{CBO}	—	10	mA	$V_{CB}=100V, I_{EB}=0$
Emitter Cut-off Current	I_{EBO}	—	1.0	mA	$V_{EB}=8V, I_{CB}=0$
Gain Bandwidth Product Typ.	f_t	1.0	—	MHz	$I_C=5A, V_{CE}=10V$ $f=100KHz$
Collector Capacitance	C_{obo}	—	1800	pf	$V_{CB}=10V, f=1MHz$
Switching Speed Typ.	t_r	—	2.5	μs.	
(PowerTech Test Circuit)	t_s	—	3	μs.	$I_C=50A$
	t_f	—	2.5	μs.	$I_{B1}=10A, I_{B2}=5A$

*PW ≤ 300μs., D.C. ≤ 2%



PowerTech, Inc.

8 BAKER COURT CLINTON N.J. 07011

TEST DATA

CUSTOMER W. J. Higgins

P.O.# 8764041000 PRINT #

GROUP MT-1223

SUBGROUP

TEST CONDITIONS: Tc = 25°C

LIMITS OR END POINTS:

UNIT #	② 200mA @ 8V			③ 250mA @ 5V				
	<u>Flow</u>	<u>I_{sc}</u>	<u>I_{rs}</u>	<u>IFE</u>	<u>I_{sc}</u>	<u>I_{rs}</u>		
	<u>min</u>	<u>max</u>	<u>max</u>	<u>min</u>	<u>max</u>	<u>max</u>		
	<u>300V</u>	<u>15mA</u>	<u>10mA</u>	<u>100</u>	<u>3.0V</u>			
001	254V	<100µA	<100µA	160	2.17			
002	252	<100	<100	175	2.07			
003	252	<100	<100	175	1.98			
004	248	<100	<100	167	2.03			
005	254	<100	<100	205	2.03			
006	248	<100	<100	154	2.14			
007	255	<100	<100	183	2.08			
008	255	250µA	<100	160	2.05			

SIGNED

DATE 1/2/73

A-Z

MT-1223

250 AMPERE POWERBLOCK POWER SYSTEM

MAXIMUM RATINGS

Collector-Base Voltage
Collector-Emitter Voltage
Emitter-Base Voltage
Peak Collector Current
D. C. Collector Current
Power Dissipation at 25°C Case Temperature
Power Dissipation at 100°C Case Temperature
Operating Junction Temperature Range
Storage Temperature Range
Package:
Thermal Resistance

SYMBOL

V_{CBO}
 V_{CE}
 V_{EBO}
 I_{CM}^*
 I_C
 P_D
 P_D
 T_J
 T_A
 θ_{JC}

MT-1223

200V
200V
10V
250A
250A
975W
600W
-65 to 200°C
-65 to 150°C
PPS-500
0.17°C/W

ELECTRICAL SPECIFICATIONS (at 25°C unless otherwise noted)

TEST	SYMBOL	MIN.	MAX.	UNITS	TEST CONDITIONS
D. C. Current Gain*	h_{FE}				$I_C = \quad V_{CE} =$
D. C. Current Gain*	h_{FE}	100			$I_C = 250A, V_{CE} = 5V$
Collector Saturation Voltage*	$V_{CE(sat)}$		2.5	V	$I_C = 250A, I_B = 5A$
Collector Saturation Voltage*	$V_{CE(sat)}$			V	$I_C = \quad I_B =$
Base Emitter Voltage*	V_{BE}			V	$I_C = \quad V_{CE} =$
Base Emitter Voltage*	V_{BE}		3.0	V	$I_C = 250A, V_{CE} = 4V$
Collector-Emitter Breakdown Voltage*†	$V_{CE(sus)}$	200		V	$I_C = 50mA$
Collector Cutoff Current**	I_{CES}		10	mA	$V_{CB} = 200V, R_{BE} = 0$
Emitter Cutoff Current***	I_{EBO}		15	mA	$V_{EB} = 8V, I_C = 0$

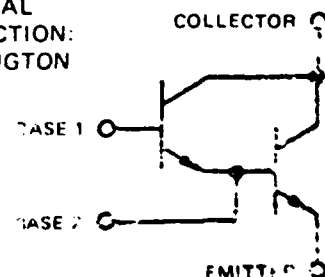
* $\leq 300\mu\text{sec}$, DC $\leq 2\%$

** Base #1 connected to Base #2

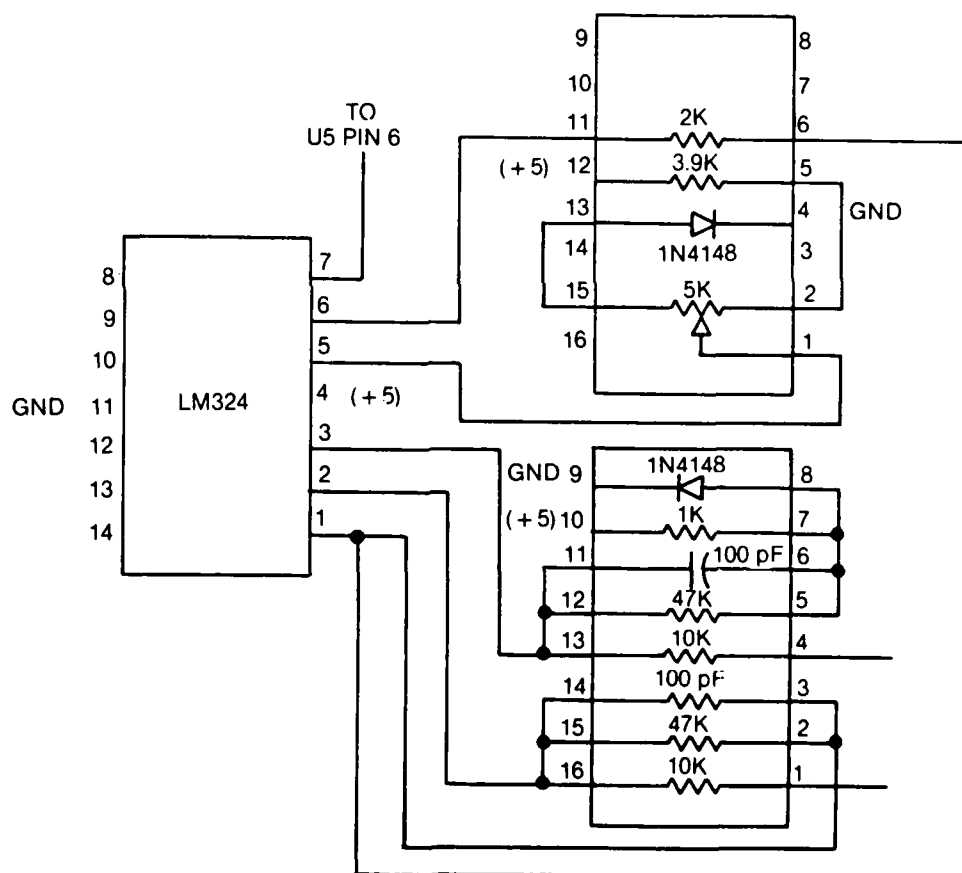
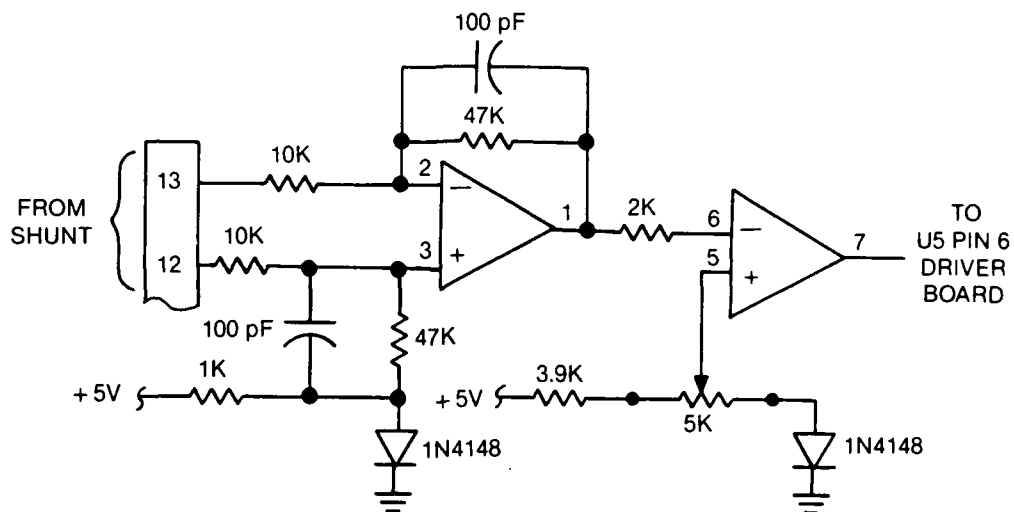
*** Base #2 open circuit

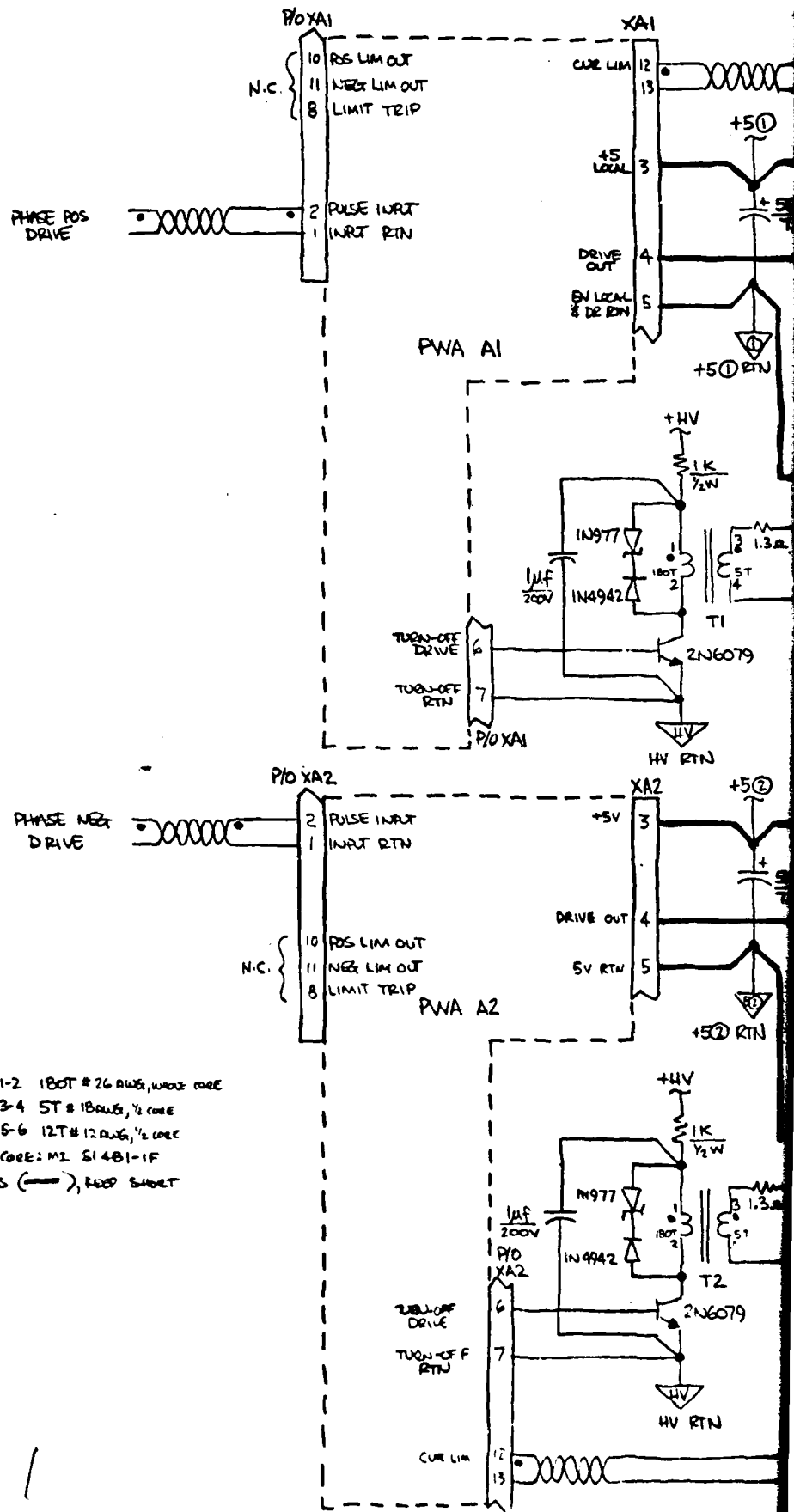
† $R_{B1B2} = 100\text{ ohms}$, $R_{B2E} = 10\text{ ohms}$

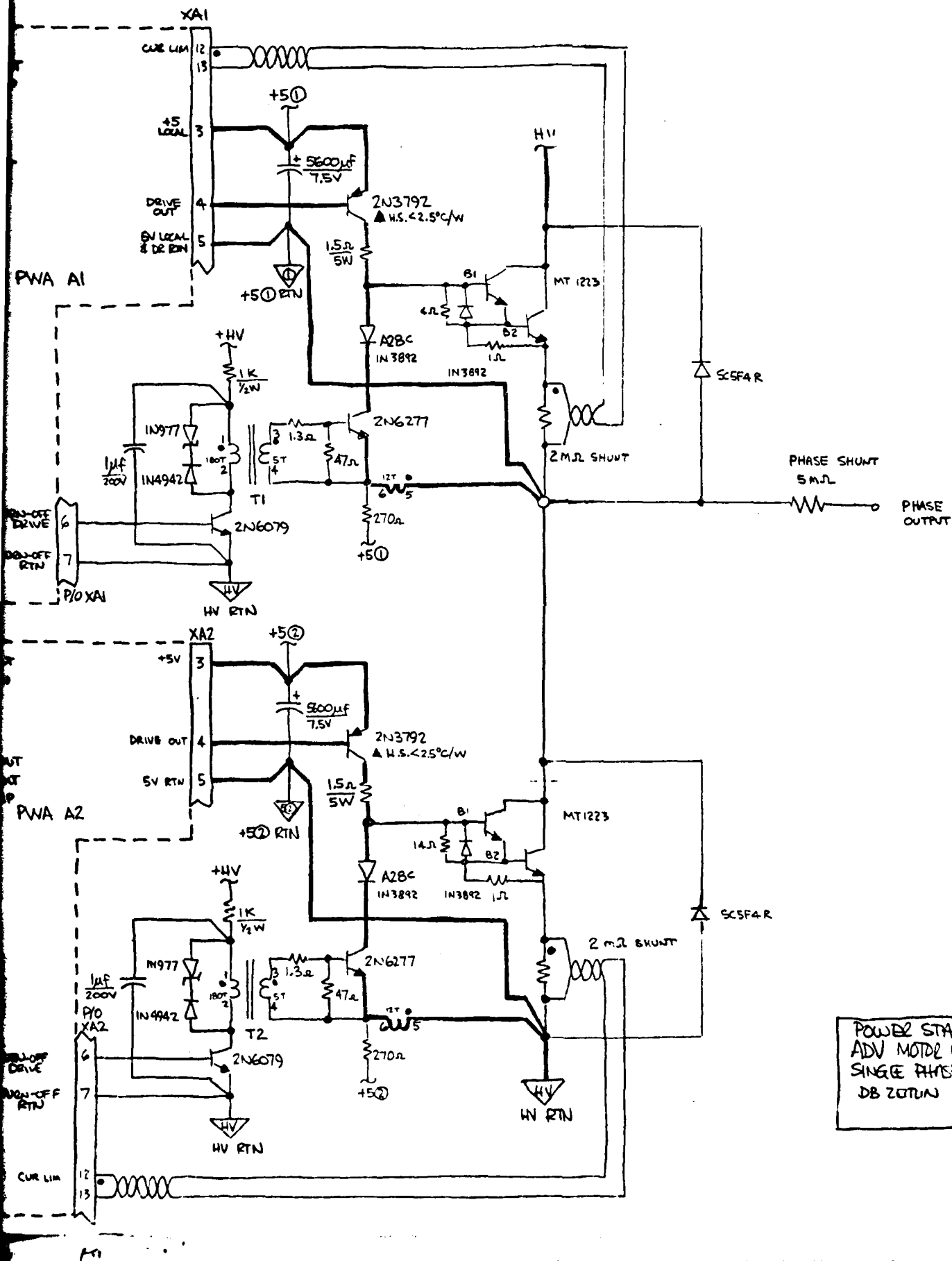
INTERNAL
CONNECTION:
DARLINGTON



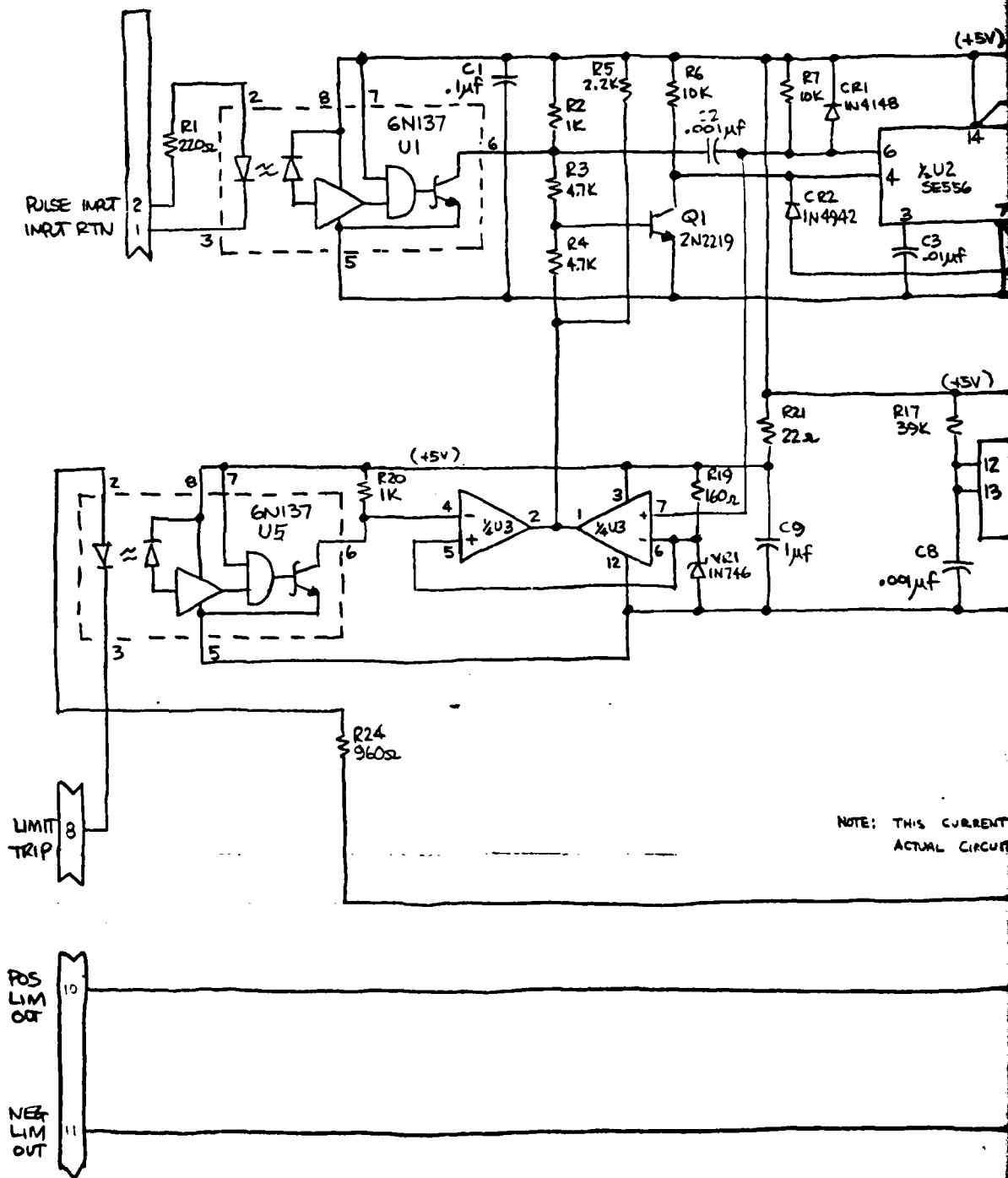
APPENDIX B





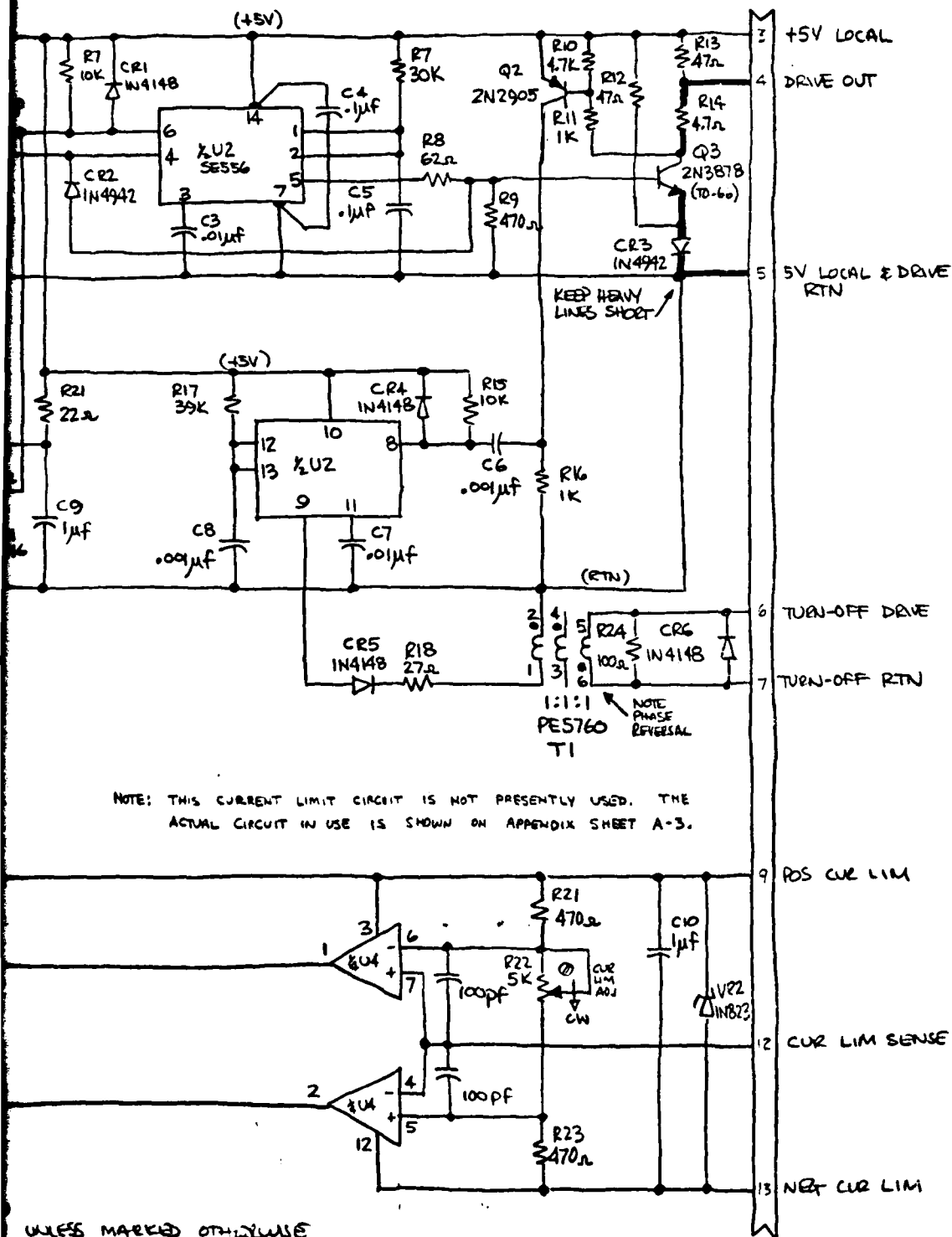


POWER STAGE
ADV MOTOR CONTROLLER (1RED)
SINGLE PHASE OF THREE
DB 207UN



NOTES:

1. 13 CONN. PINS USED
2. ALL RESISTORS RCOT UNLESS MARKED OTHERWISE
3. ALL CAPACITORS CK06 UNLESS MARKED OTHERWISE
4. Q1 IS TD-66 STYLE
5. U1 & U5 ARE 8-PIN DIP - 6N137
6. U2 IS 14-PIN DIP - SE556
7. U3 & U4 ARE 14-PIN DIP - LM139
8. CUR LIM IS SET UP FOR 25A MINIMUM ADJUSTMENT (470K)



UNLESS MARKED OTHERWISE
 10K UNLESS MARKED OTHERWISE
 1N DIP - 6N137
 1N DIP - SE556
 1N DIP - LM137
 2.5A MINIMUM ADJUSTMENT (470Ω-5K-470Ω)

IR & D MOTOR CONTROL
 DRIVER BOARD
 DB ZETUN

2

**DAT
FILM**

6

—