1976

Digital programmable timing device for fast response instrumentation in rotating machines.

West, James Clyde

Monterey, California. Naval Postgraduate School

DIGITAL PROGRAMMABLE TIMING DEVICE FOR FAST RESPONSE INSTRUMENTATION IN ROTATING MACHINES

James Clyde West

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DIGITAL PROGRAMMABLE TIMING DEVICE
FOR FAST RESPONSE INSTRUMENTATION
IN ROTATING MACHINES

by

James Clyde West, Jr.

December 1976

Thesis Advisor: R.P. Shreeve

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Digital Programmable Timing Device for Fast Response Instrumentation in Rotating Machines

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis; December 1976

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

James Clyde West, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

December 1976

13. NUMBER OF PAGES

95

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital
Time Delay
Programmable
Phase-Locked-Loop

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The design, construction and test of an inexpensive computer peripheral device to control the acquisition of data from high response probes in periodic flows, is reported. The device (RPACE) was used with a Hewlett-Packard Model 21MX computer and 5610A A/D converter to obtain Kulite probe measurements in a transonic compressor by "synchronized sampling". A phased-locked-loop and

(20. ABSTRACT Continued)

counting circuits were used so that the moment of A/D conversion always corresponded to a programmable displacement of a stationary probe with respect to the moving rotor blades — independent of RPM. Also, the rotor speed was measured digitally in one revolution of the shaft. The results of a "survey" of a rotor blade passage at a blade passing frequency of 4500 per sec. are included.

Digital Programmable Timing Device for Fast Response Instrumentation in Rotating Machines

by

James Clyde West, Jr.
Lieutenant, United States Navy
B.S., University of Oklahoma, 1970

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1976

ABSTRACT

The design, construction and test of an inexpensive computer peripheral device to control the acquisition of data from high response probes in periodic flows, is reported. The device (RPACE) was used with a Hewlett-Packard Model 21MX computer and 5610A A/D converter to obtain Kulite probe measurements in a transonic compressor by "synchronized sampling". A phased-locked-loop and counting circuits were used so that the moment of A/D conversion always corresponded to a programmable displacement of a stationary probe with respect to the moving rotor blades — independent of RPM. Also, the rotor speed was measured digitally in one revolution of the shaft. The results of a "survey" of a rotor blade passage at a blade passing frequency of 4500 per sec. are included.

TABLE OF CONTENTS

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

B	4-bit binary counter (74193)
L	4-bit latch flip-flop (7475)
C	4-bit comparator (9324)
D	Delay flip-flop (7474)
U	AND gate (7408)
I	Inverter (7404)

ABBREVIATIONS

A/D	Analog-to-Digital
TX	Transducer

Note: Numbers in parentheses refer to Signetic Catalogue.

ACKNOWLEDGEMENT

The design and development of PRACE, as reported in this paper, would not have been possible without the guidance and moral support of my advisor, Associate Professor R. P. Shreeve. His insight and background knowledge of problem areas in turbomachine instrumentation were invaluable in this design effort.

Secondly, the design of the physical components and instrumentation used in the transonic compressor were outstanding contributions made to this work by J. E. Hammer. The design of the optical wheel was a key factor in the success of this project.

Finally, the expert knowledge of electronic circuits by M. V. Frutos were invaluable in the design of the "Wave Shaper" circuits and in incorporating the PLL into RPACE.

I. INTRODUCTION

The work reported here is part of a program at the Naval Postgraduate School to determine the behavior of the flow in a transonic compressor using fast response instrumentation. Case wall pressure signatures are to be measured using Kulite pressure transducers, followed by Kulite probe and hot-wire measurements downstream of the rotor. The stage under investigation was designed by Dr. M. H. Vavra and is located in the Turbopropulsion laboratories of the Naval Postgraduate School. The work is sponsored by Naval Air Systems Command, Code 310, through the offices of Dr. H. J. Mueller.

Until the development of miniature semiconductor transducers having large bandwidth frequency response, real-time pressure measurements in high speed compressors were not feasible. Currently, using these pressure transducers, coupled with fast analog-to-digital conversion and computer controlled time delay circuits, a variety of new real-time measurement techniques can, in principle, be applied.

For example, it is desirable to be able to determine the flow out of a first stage rotor, which is steady with respect to the rotor blades, using stationary probes downstream of the rotor. This can be done if a fixed sensor can be sampled at one per revolution intervals ("synchronized sampling") and if the sample point can be progressively delayed to, in effect, survey across a particular rotor

blade passage. Data acquired in this way can be examined to detect blade flutter frequencies, which is currently an area of concern in advanced compressor development.

The difficulties to be overcome in the design of a sampling method must include achieving timing accuracy at blade passing frequencies as high as 10 KHz, and accounting for the small variations in rotor speed that are always present. In the present work, a peripheral device was designed and constructed that can trigger data acquisition from a stationary sensor at any fixed point in the rotating rotor frame, independent of the rotor speed. The peripheral device, in effect, divides the perimeter of the rotor into a finite number of sub-intervals between each blade. Because the number of sub-intervals (126 pulses) is a constant between any two adjacent blades, the time between the sub-intervals is then inversely proportional to the speed of the rotor.

The peripheral device, called RPACE, receives one per revolution and one per blade input signals and performs two independent functions; it controls the timing for data acquisition and determines the speed of the rotor in one revolution of the wheel. A phased-locked-loop (PLL) is a key element in the design of the device. RPACE is incorporated into a high speed data acquisition system which uses a Hewlett-Packard model 21MX computer and model 5610A Analog-to-Digital (A/D) converter. In the present work, the hardware and software for the device were generated, calibration

measurements were made and compressor test data were obtained in the "synchronized sampling" mode.

In the following section, a general description of the device is given. In Section III, a description is given of the computer software used to control RPACE and in Section IV, the hardware is described. Section V reports the calibration techniques and applications of the peripheral device. Section VI contains recommendations that would improve the operations of RPACE. Appendix A contains a detailed description of the computer software and interrupt structure of the 21MX computer. The detailed circuit description of RPACE is given in Appendix B.

II. GENERAL DESCRIPTION

A schematic of the components of the system for "synchronized" high speed data acquisition is shown in Figure 1. The high speed data system itself is described in Reference 1. The new components of the system include a wheel containing holes at 1 per blade and 1 per rev. intervals, light sensitive diodes to provide wave pulses at the blade passing frequencies, wave shaper circuits to provide square pulses from the optical signals, and the computer peripheral device, RPACE. In this report, only the design and operation of RPACE is documented.

The first function of RPACE is to control the data acquisition by the A/D converter. RPACE accomplishes this by intercepting the encode command from the computer and causing a delay by counting a predetermined number of pulses, whose frequency is set by the rotor speed. The number of pulses of delay required is entered through the computer software. RPACE then generates a new encode pulse which is used to initiate data acquisition by the A/D converter at the programmed time. This sequence of events is accomplished using a phased-locked-loop circuit operating at a multiple (nf_o) of the incoming frequency. The generated pulses, nf_o, are counted by a sixteen bit binary counter and, when the programmed number of pulses have elapsed, a pulse is generated by the comparator. The function of the comparator

is to compare the number of pulses that have been counted to the number that was programmed, and to generate a pulse when the two numbers are in agreement.

The second function of RPACE is to determine the RPM of the rotor and to transfer this information to the 21MX computer. RPACE is able to determine the rotor RPM in one revolution. Determination of the RPM is accomplished by counting the number of pulses, generated by the computer's "time base generator" (TBG), between pulses of the one per rev. signal. The 21MX computer determines the revolutions-per-second by dividing the TBG clock frequency by the number of clock pulses that have elapsed.

III. COMPUTER SOFTWARE AND PROGRAMMING (OPERATING) INSTRUCTIONS

A. INTRODUCTION

The transfer of data between a computer and its peripheral device can be accomplished using either the interrupt or non-interrupt method of control. Use of the interrupt method is more efficient in the use of computer time, but is more difficult to program.

Use of the non-interrupt method of control involves a "wait-for-flag" in which the computer commands the device to operate and then waits for the device to set the flag. Use of this method forces the computer into a "halt" state while awaiting the device flag.

Use of the interrupt program structure is more complicated than the non-interrupt structure but results in more efficient use of computer time. The procedure for using interrupts requires that, initially, an instruction be issued which sets the command flip-flop and clears the device flag flip-flop. Issuance of this instruction causes the peripheral device to be activated and the computer then continues execution of the main program. Upon completion of its task, the peripheral device sets the device flag, thereby causing the computer to suspend the current program and jump to the interrupt location corresponding to the device input/output card. At this location, an instruction was previously placed by the interrupt linkage section of the subroutine (this

section is called the continuator section by Hewlett-Packard), which instructs the computer to jump to a specific location in memory where the continuator section of the subroutine is located [Ref. 2]. Upon completion of the subroutine, control is returned to the main program and the cycle repeats itself.

The peripheral device RPACE is addressed by the computer using the interrupt method. The "driver" subroutine called by the main program is "RPACE", an assembly language subroutine which is incorporated into the computer operating system when that system is generated. In the following sections, the peripheral driver program is described first, followed by the Fortran subroutine that calls the A/D converter, "R5610". The calling sequence in Basic Language appears as:

```
                    :
100 RPACE(IBLAD,IRPM,IEND)
110 R5610(LU,RBUFF,N,ICHAN,IMODE,RCHAN)
                    :
```

An example of the application of RPACE is discussed in Section V-B and the program listing is given in Appendix C.3.

B. "RPACE"

"RPACE" is the name of a subroutine that controls a peripheral device of the same name. Use of the subroutine requires the passing of one parameter to the subroutine and receiving two parameters from the subroutine. The parameters are as follows:

1. "IBLAD"

IBLAD is an integer number which has two possible ranges of values. If a number in the interval $0 < IBLAD < 255$ is selected, then the peripheral device will cause data acquisition to occur at some point between every pair of blades. However, if IBLAD is in the interval $33024 < IBLAD < 35584$, then data acquisition will occur once for every revolution. Determination of IBLAD is as follows:

$$IBLAD = 32768 + 256*L + I$$

where L = Blade pair # and I is the desired pulse location between blade pairs. The range of these parameters is such that $1 \leq L \leq 9$ and $0 < I < 255$.

2. "IRPM"

Upon return from the subroutine "RPACE", this parameter will contain the integer number of clock pulses that have elapsed between the one per rev. pulses.

3. "IEND"

The third integer parameter contains the number of readings taken by "RPACE" and, as currently programmed, is always unity.

C. "R5610"

"R5610" is a Fortran language subroutine which calls the A/D converter. It is responsible for issuing the encode command, to be intercepted by the peripheral device, and to

convert the data values returned from the A/D converter, which are integer values, into floating point numbers between -1 and +1. This subroutine is the equivalent of the Fortran program example given in Reference 3 but uses an "EXEC" input/output call to the A/D converter as required by the RTE-B operating system. The RTE-B operating system is described in Appendix A. "R5610" is incorporated into the operating system when that system is generated. Use of "R5610" requires the use of six parameters which are defined as follows:

1. Logical Unit Number (LU)

This number identifies the A/D converter. The "logical unit numbers" are assigned to all input/output devices during generation of the RTE-B operating system (see Appendix A and Reference 4). As currently configured, the number is 7.

2. "RBUFF"

This parameter identifies the base address of the array that is to receive data values from "R5610". RBUFF must be a subscripted array, containing two or more elements, dimensioned in the Basic language calling program.

3. "N"

This decimal integer is the number of samples that "R5610" is to take prior to returning to the main program.

4. "ICHANN"

The decimal number of the analog input channel which is to be sampled by the A/D converter is passed to the

subroutine in this parameter. The number must be such that $0 \leq$ ICHANN ≤ 15.

5. "IMODE"

The mode of operation of the A/D converter is specified in this parameter. IMODE can have the values 0, 1, 2, 3, 4, or 5, as described in Reference 3. With the present hardware, only modes 0, 1, 4 or 5 can be used (modes 2 and 3 require an external control device which is similar to the peripheral device discussed in this report, but interfaces to the 21MX computer in a different way).

6. "RCHANN"

Upon return from the subroutine, the channel numbers, on which the first sixteen data values were taken, are contained in RCHANN . RCHANN is a sixteen element array which is dimensioned in the Basic calling program. It is used currently only to verify that the A/D converter is operating correctly.

D. OPERATING WITH THE RTE-B SYSTEM

The RTE-B operating system for the Hewlett-Packard 21MX computer is described in Appendix A. When the system has been generated as a single "absolute" program on paper tape [Ref. 4] and loaded into the computer through the tape reader [Ref. 5], the computer can be programmed in Basic language using the CRT keyboard; or, complete programs can be entered via the tape reader.

It should be noted first that "RPACE" and "R5610" can be called in any Basic language program and will function as described in the above sections. Secondly, any Basic statement can be edited (by entering a revised statement with the same statement number) and the program re-run.

Use of the A/D converter under the control of the device RPACE requires only that the operator is familiar with:

(i) Basic language programming [Ref. 4]

(ii) "Operator Commands" for the RTE-B system [Ref. 4]

(iii) The meaning of the calling parameters in the "RPACE" and "R5610" subroutines.

An example of a basic language program for "synchronized sampling" is given in Appendix C3. The function of the program is described in Section V.

IV. COMPUTER PERIPHERAL DEVICE (RPACE) HARDWARE DESIGN

A. INTRODUCTION

The design of RPACE incorporates a phase-locked-loop circuit. The ability of the circuit to track frequency variations and to output a pulse whose frequency is equal to, or a multiple of (nf_o) the incoming frequency (f_o) has resulted in widespread use of the circuit in various applications. The features of this type of circuit and its operations are given in detail in Reference 6.

In the two sections of Figure 2 the peripheral device RPACE is shown schematically. The device logic is described in the following sections with reference to this figure. The contents and functions of the subsections are listed in Table I. The circuits for the subsections are given in Figures 3 through 7. Oscilloscope records showing the relationship between signals at various points in the circuits are given in Figures 8 through 13. Details of the circuitry and components are given in Appendix B.

B. RPACE DEVICE LOGIC

1. Trigger Signal Conditioning

The "Wave Shaper" section takes the raw signals from the optical detectors and converts the bell shaped curve into a pulse, suitable for T.T.L. (Transistor-Transistor-Logic) connections. Figure 4 shows the circuit for the "Wave Shaper" section. Oscilloscope traces showing the relationship between the input and output signals from this section are shown

in Figure 8. The output of the one-per-blade wave shaper is connected to a binary counter (B5 in Figure 2) which generates a square wave suitable for signal processing in the phase-locked-loop circuit (PLL). The conversion to a square wave is required because of the action of the phase comparator of the PLL circuit. The phase comparator requires an input wave form of the same frequency and symmetry as the waveform used for feedback. Therefore the original one per blade signal is modified into a square wave whose period is proportional to the time required for the passage of two adjacent blades. The output of the one per rev. wave shaper is connected to D1 (Figure 2) in the "Time-set" section, via a buffer amplifier.

2. "Encode Delay"

The phase-locked-loop (PLL) circuit, shown in Figure 4, requires two inputs; one input comes from B5 of the Wave Shaper Section and the other input comes from B2 of the "Encode Delay" section. Operation of the PLL circuit is such that the waveform from B2 has a 270° phase shift compared to the input from B5. This is seen in the oscilloscope record shown in Figure 12. By sensing the phase difference between the two signals, the PLL circuit generates a proportional steering voltage which is applied to correct the frequency of a voltage controlled oscillator (VCO), thereby driving the steering voltage to a zero level. The zero voltage level occurs when the two frequencies are

identical. The frequency of the VCO is $256f_o$ but, because the output of the VCO is divided by 256 (B1 and B2, Figure 4), the two signals that are compared have the same frequency (f_o).

The "RPACE" subroutine causes a sixteen bit word to be transferred to the peripheral device, RPACE, where it is strobed (pulsed) through the four 4-bit latches (L1 to L4, Figure 2) to one side of a digital comparator (C1 to C4, Figure 2). The computer word is compared to the output of the four binary counters (B1 to B4), which count the number of pulses generated by the VCO since the last one per rev. pulse. When the two bit patterns are the same, the comparator generates an encode pulse, thereby initiating data acquisition by the A/D converter.

The four 4-bit counters (B1 to B4) are operated in one of two modes. In mode I, one pulse is generated for every complete cycle that is fed back to the PLL circuit. Thus, when operating in this mode, the comparator (C1 to C4) will generate nine pulses per revolution. Operating in this mode requires that IBLAD be a number between 0 and 255 (i.e. $0 \leq \text{IBLAD} \leq (2^8-1)$). Operation in Mode II causes the A/D converter to acquire data once per revolution. Then, IBLAD must be a number between 33024 and 35584.

The reason for the two ranges in values of IBLAD is that bit 15 must be set either low or high to select operation in Mode I or Mode II respectively. If bit 15 is high, then Mode II operation of the binary counters (B1 to B4) is desired. Thus, the carry from B2 must be

transferred to B3, via U1 (Figure 2). Transfer of the carry permits the number of blades per revolution to be counted. If Mode I operation is selected, then bit 15 will be low and the carry from B2 will not pass through U1, and B3 and B4 will always output a low to the digital comparator. Thus, the corresponding bit on the other side of the comparator must also be zero, which means that bits 8 to 15 must be zero.

3. A/D Converter Command and Flag

When the main program calls "R5610", the subroutine sets the device command and clears the device flag flip-flop inside the input-output module (11_8). When the above events have occurred, the computer is expecting the A/D converter to respond by driving its device flag line to a low. Referring to Figure 5, the delay flip-flop D8 is used to satisfy the above requirements without triggering the A/D converter. The actual triggering of the A/D converter requires two separate events to occur; the delay flip-flop D8 must be set, and then the comparator must generate a pulse. When these two events have occurred, the A/D converter will respond by taking a data sample and, ten microseconds later, drive its device flag high. The device flag from the converter is connected to the clock input of D8. Because the clock input of D8 is a positive edge-triggered device, D8 will reset itself when the A/D converter flag has transitioned from a low to a high. When D8 resets, gate U2 is closed, and prevents any comparator pulses from triggering the converter. The above process is repeated upon receipt of another encode command from the computer.

4. RPM Clock

The determination of the RPM of the Rotor, based on one revolution, requires that the reference frequency (clock) be a crystal controlled oscillator operating at a high frequency. Ideally, the frequency of the clock divided by the revolutions per second should be close to 65535 or $(2^{16} - 1)$ without exceeding this value. However, because the rotor is a physical device, there will be fluctuations in the speed and thus, the number of pulse desired should be one less than 2^{15}, or 32767. By selecting the smaller number, any surge or overshoot in number of revolutions per second will not result in the loss of any clock pulses.

The circuit for the "RPM Clock" section is shown in Figure 6. The internal crystal controlled oscillator (or clock) in the 21MX computer is connected to the up-down binary counter B4 which divides the one megacycle clock frequency into a frequency range that is more suitable for the readings given above. The range selected depends upon the speed of the rotor. Additionally, B4 serves to buffer the clock output from the rest of the circuit, thus preventing loading and drift in the clock frequency. The output of B4 is used to clock the delay flip-flops of section 5 (D2 to D6 in Figure 7) and to drive the four 4-bit up-down counters.

At the correct time, the outputs of B5 to B8 are pulsed through four 4-bit latches (L5 to L8 in Figure 6) in preparation for transfer to the computer. The data that is present at

the input will be transferred to the output of the latches as long as the clock line is high. When the clock line goes low, the output is isolated from the latch inputs. As long as the clock line is low, input data can be allowed to change without affecting the output data. The clock input of L5 in Figure 6 is connected to D1 of the "Time-set" section shown in Figure 7.

The Master Reset line is set at the next clock pulse and clears the counters (B5 to B8 in Figure 6). The Master Reset line (MR) is connected to D4 of the "Time-set" section (Figure 7). After clearing the counters, a binary number is loaded into the counters via the parallel load (PL) line. Loading the binary number 0101 = 5_{10} compensates for the number of clock pulses that were missed while the counters were changing as shown in the timing diagram in Figure 14.

5. RPM Clock Control

The one-per-blade output of the "Wave Shaper" section is connected to the clock input of D1 (Figure 7). The positive edge of the pulse is used to initiate the gating sequence, thus transferring data through the latches of the "Time-set" section. Next, the Master Reset pulse of D4 is used to reset the binary counters (B5 to B8 in Figure 6) to zero. The sequence is terminated by D6 going low which forces the information that was stored on the parallel data lines to be preset into B5 through B8. Referring to Figure 14, the number of pulses missed is five, thus the binary number 101 is connected to the parallel data input lines. The

information is then transferred to the binary counter B3 output, when $\overline{PL}$ goes low. When $\overline{PL}$ goes high, normal counter operations resume.

5. RPACE Command and Flag

Referring to Figure 2 and Figure 7, the output of D6 is connected to the clock input of D7. When D6 outputs a positive edge, D7 sets, and Q of D7 goes high, causing an interrupt to occur. When the computer acknowledges the interrupt, it jumps to memory cell 17_8 and executes the instruction that was stored at this location when the subroutine "RPACE" was called.

After control has been returned to the main program and when subroutine "RPACE" is called again, a binary number is sent to L1 and the device control and device flag flip-flops at OCTAL I/O Slot #17_8 are set. The setting of device command causes a pulse to be transferred to the clear of D7 which sets Q to a low. When the events described in Section B.5 have occurred, the process described here is repeated.

V. CALIBRATION AND APPLICATION

A. CALIBRATION

In the application for which RPACE was intended, the requirement is that the position of a fixed probe be known with respect to the moving rotor blades at the time that the A/D converter digitizes the probe's output. This translates to the requirement that the blade location be known with respect to the pulse that is generated by the comparator of the "Encode Delay" section. However, because of errors associated with locking the phase-locked-loop circuit (Ref. 5, pp. 6-66), the location of the comparator pulse with respect to the input waveform can vary by ±5% of the period (which remains constant), depending on how the circuit was locked up. Thus, when acquiring data using RPACE, the reference signal from the one per blade trigger must be scanned together with the pressure transducer data to fix the time origin.

Hence, the timing calibration of the RPACE System is "on-line". The reference signal generated by the one per blade trigger is scanned as part of the data acquisition process. In this way, the data is referenced to the optically generated trigger pulse, which in turn is related to the position of the holes in the timing wheel (Figure 1). The orientation of the holes in the timing wheel with respect to the rotor blades can be established by inspection. A knowledge of when the one-per-blade output pulse transitions

from a low to a high locates the data conversion with respect to the position of the hole in the timing wheel.

In order to examine the relationship of the trigger pulse to the input signal wave form, and to examine whether RPACE would operate as designed, measurements were made using a 10,000 Hz motor-driven timing wheel. The wheel was equipped with a magnetic flux cutter aligned with the one per blade timing holes, in addition to the optical emitters and sensors shown in Figure 1. The magnetic flux cutter signal, the one per blade optical signal and the conditioned one per blade optical signal were input on different A/D converter channels. A program was written to "survey" the three periodic signals using RPACE. The program is given in Appendix C.3. The "survey" was carried out in the following way: IBLAD was set equal to 33024. Ten samples were taken (on successive revolutions) on one input channel and the average was stored. This was repeated for the other two channels. IBLAD was incremented by unity in steps for a total of 255 steps. At each step, the average of 10 samples was recorded on each of the three channels. The data for the stored average values are shown plotted in Figure 16 as a function of the number of the step (0-255). It can be seen that the reconstructed periodic waveforms have the appearance of time traces of the input data, which they are not. The fact that sharp increases remain after the averaging process, and that the two periods of the waveform are identical confirmed the operation and the repeatability of the device.

B. SYNCHRONIZED SAMPLING RESULTS FROM THE TRANSONIC COMPRESSOR

In a transonic compressor test, the pressure was measured at the case wall using a Kulite pressure transducer and synchronized sampling. The compressor, the transducer and the data system were as described in Ref. 1. The only difference here was that RPACE was used in acquiring the data from the single transducer channel. The method was similar to that described for the calibration measurements except that 15 data values were averaged for each data point, and the standard deviation in the samples was also recorded. The program listing is given in Appendix C.4.

The reconstructed waveform from data obtained across a single blade passage is shown in Figure 17. Also shown is an oscilloscope record that is a time trace of the single blade passing the transducer. The shapes of the waveforms and the magnitude of the voltages were similar. Additionally, the time between peaks on the oscilloscope trace was about 2.25×10^{-4} seconds, whereas 2.12×10^{-4} seconds was obtained from the reconstructed data plot.

In Figure 17, curves representing ± 1 standard deviation about the mean value (of 15 measurements) are also shown. The unsteady and noise components are therefore of the order of 1×10^{-2} volts or 2% of the mean value.

VI. CONCLUSIONS AND RECOMMENDATIONS

The equipment developed in the present work enables digitized data to be acquired from high response compressor instrumentation in a "synchronized sampling" mode. The system gave good results in two tests, the second of which was to acquire data from the transonic compressor. Use of the phased-locked-loop circuit made the timing accuracy independent of machine speed; however, it has been shown to be necessary to input the trigger signal onto one channel of the A/D converter in order to establish the time scale exactly. This presents no problem. The present system can be used routinely with the existing hardware and software (to establish wall pressure maps from eight transducer channels, for example). If a new model of the device is required, the following changes should be incorporated:

1. A phased-locked-loop circuit at a higher frequency should be used; and since this circuit is the heart of RPACE, the components should be of much higher quality than the present chip used in the system.
2. In the present equipment, the number of pulses produced by the phased-locked-loop circuit between two adjacent blades is 126. If the number of holes in the disk were doubled, then the number of pulses between adjacent blades would increase to 256, which would double the number of sample points between

blades. However, the operating frequency of the PLL circuit would also be doubled so that a higher quality chip would be required.

3. With the present configuration of RPACE, the A/D converter can not sample between the arrival of the master trigger pulse and the next positive edge of the output of B2. A circuit should be designed to eliminate this restriction.

Section	Contents	Function
1. "Wave Shaper"	Shaping & Conditioning circuits for 1 per blade and 1 per rev. signals	Converts bell-shaped optical input signals to square pulses.
2. "Encode Interrupt"	Phased-locked-loop (PLL) circuit Up-Down binary Counters (B1-B4). Comparator 4-Bit Latches (L1-L4)	Determines time delay required for encoding based on 16-bit word from IBLAD.
3. "Encode Command"	Delay Flip-Flop (D8) "AND" gates U2, U3 "Inverter" I2	Gives proper flag responses to 2IMX computer, as required by programming, while waiting for "encode command" from "Encode Interrupt".
4. "RPM Clock"	Up-Down binary counters (B5-B9) 4-bit latches (L5-L8)	Counts the number of clock pulses between one per rev. pulses.
5. "Time-Set"	Delay Flip-Flops (D1-D6)	Controls the sequencing of information from "RPM Clock".
6. "RPM Output"	Delay Flip-Flop (D7) Inverter (I1)	Gives proper flag responses to 2IMX computer, as required by programming.

TABLE I Subsections of the Device RPACE and Their Functions

Component	Schematic Number	Value or Type No.
Resistors:	R_{14}, R_{20}	22 KΩ
	R_{23}, R_{18}, R_{12}	2.7 KΩ
	R_{21}, R_{15}, R_9	5 KΩ
	R_{11}, R_{10}, R_{24}, R_{16}, R_{17}	10 KΩ
	R_{19}, R_{13}, R_7	2.2 KΩ
	R_8, R_5	50 KΩ
	R_3, R_4	4 KΩ
	R_1, R_2	1 KΩ
	R_6	166 KΩ
Capacitors:	C_2, C_6, C_8, C_9, C_{10}, C_{12}	.1 μf
	C_7, C_{11}, C_{13}	.022 μf
	C_1	.1056 μf
	C_3	.001 μf
	C_4	variable
Integrated Circuits:		
4-Bit Counter	B1 thru B10	74193
4-Bit Latch Flip-Flop	L1 thru L8	7475
4-Bit Comparator	C1 thru C4	9324
AND Gate	U1 thru U3	7408
Inverter	I1, I2	7404

TABLE II Component Used in RPACE

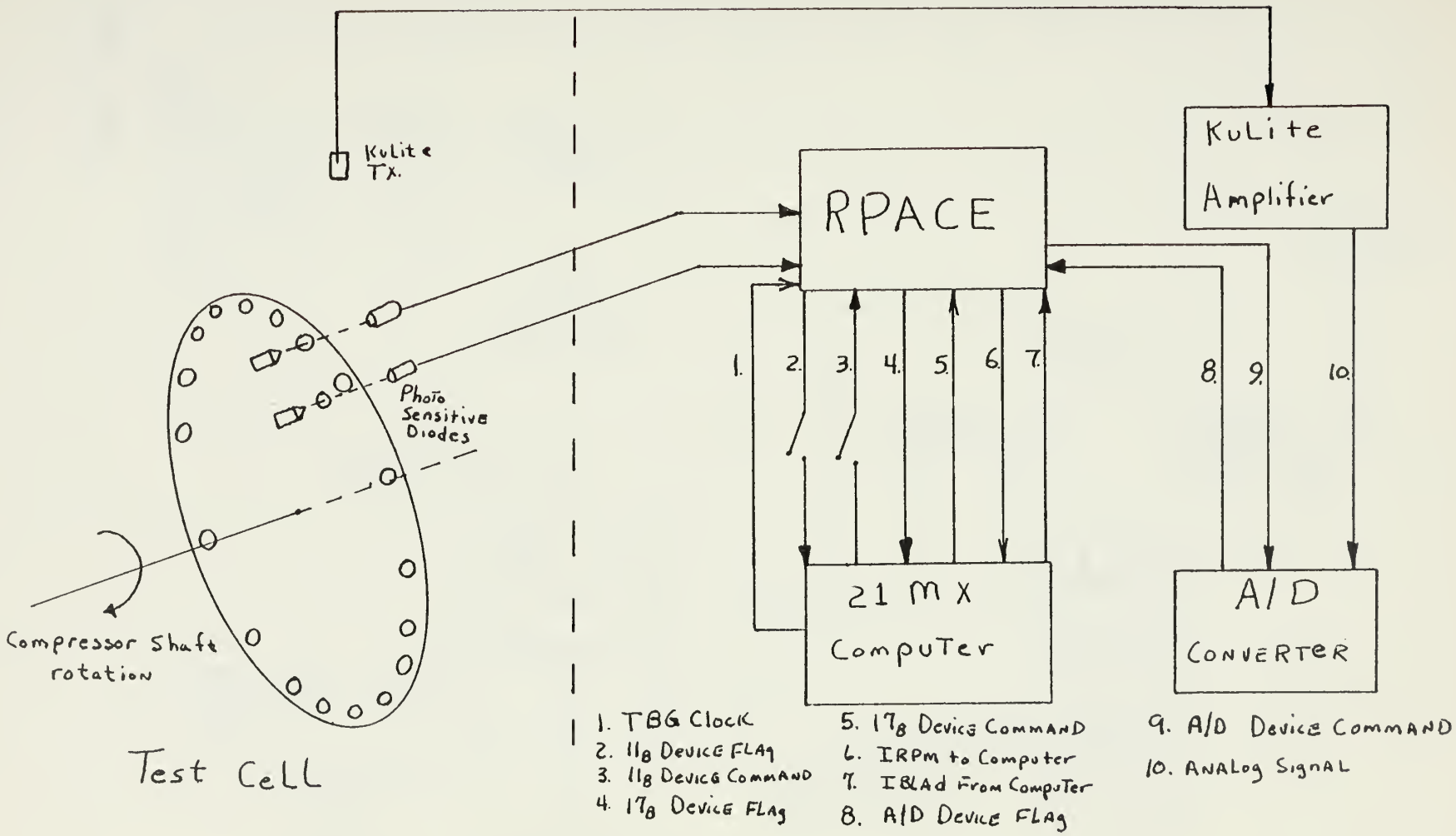


FIGURE 1. Synchronized Sampling System

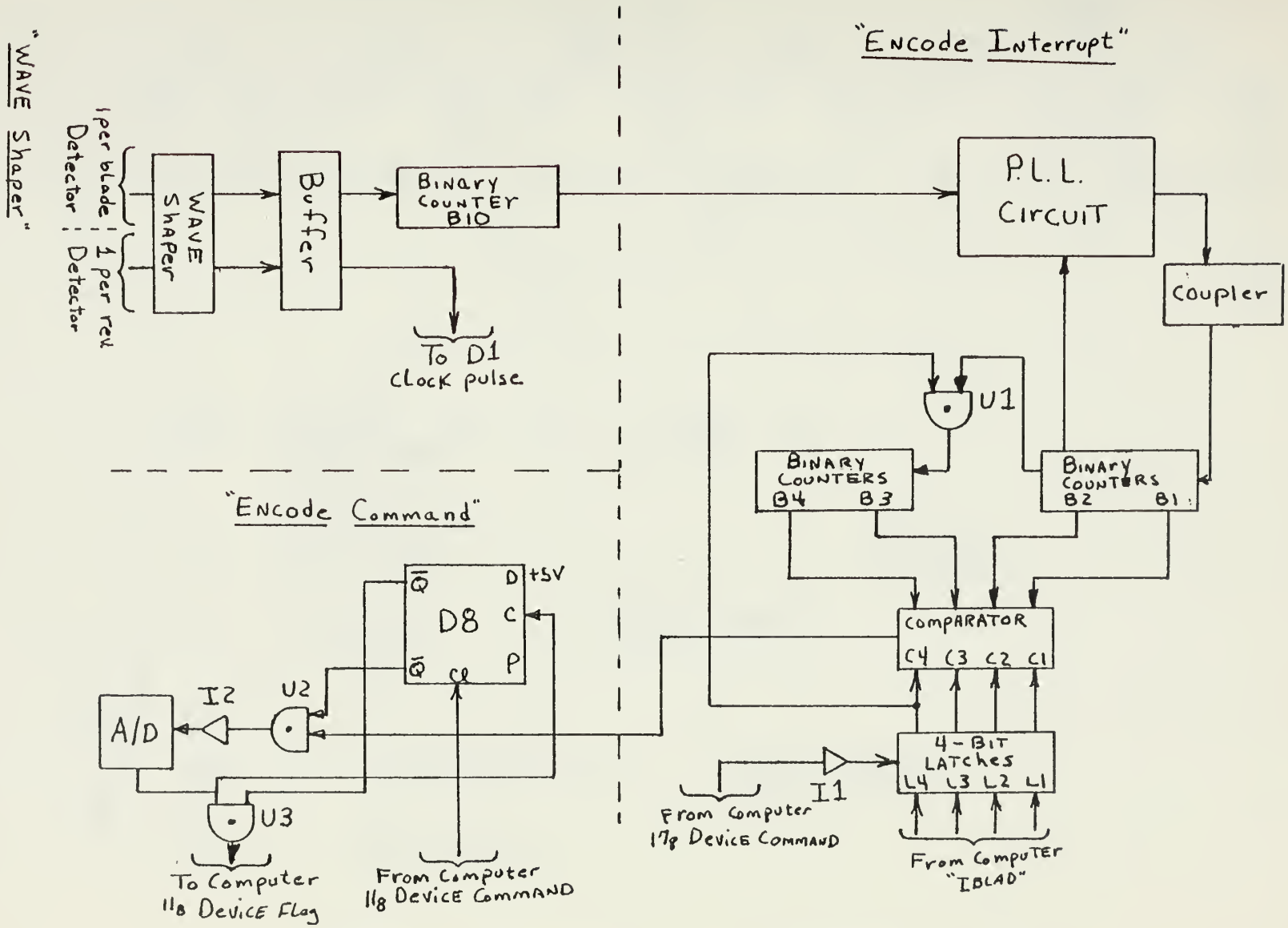


FIGURE 2. Schematic Diagram of RPACE

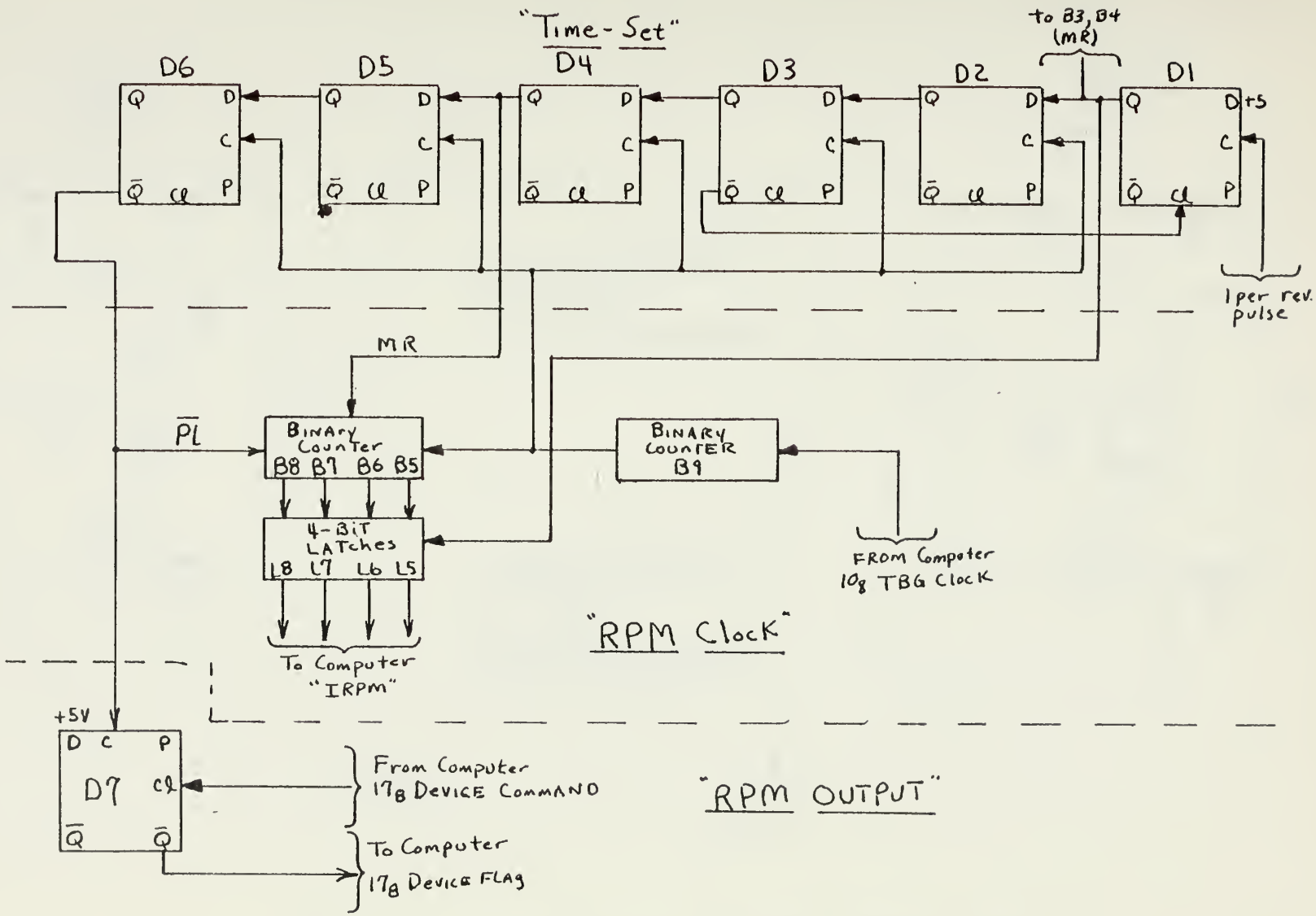


FIGURE 2 (Continued) Schematic Diagram of RPACE

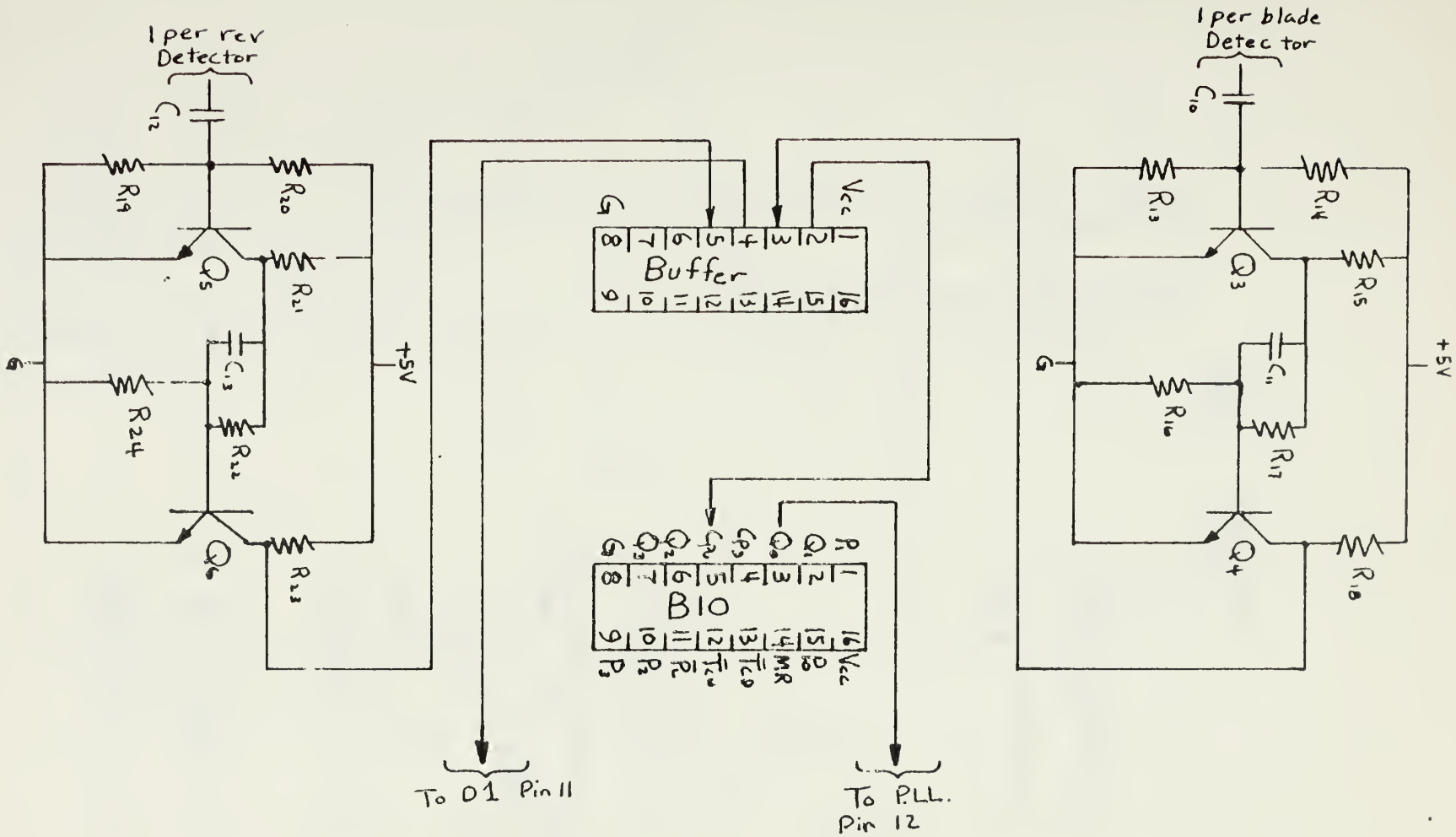


FIGURE 3. Circuit Diagram of "WAVE SHAPER" Section

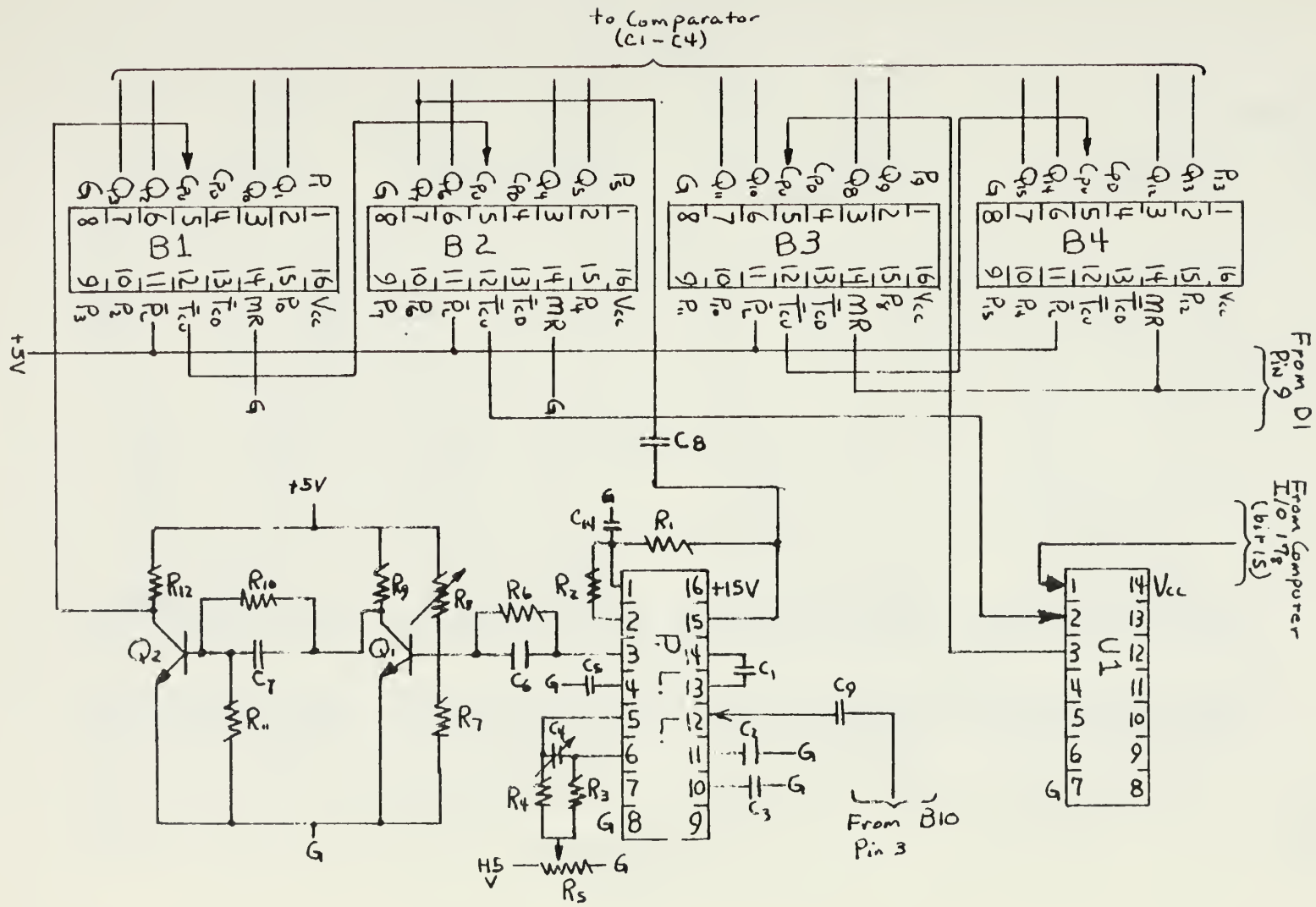


FIGURE 4. Circuit Diagram of "ENCODE INTERRUPT" Section

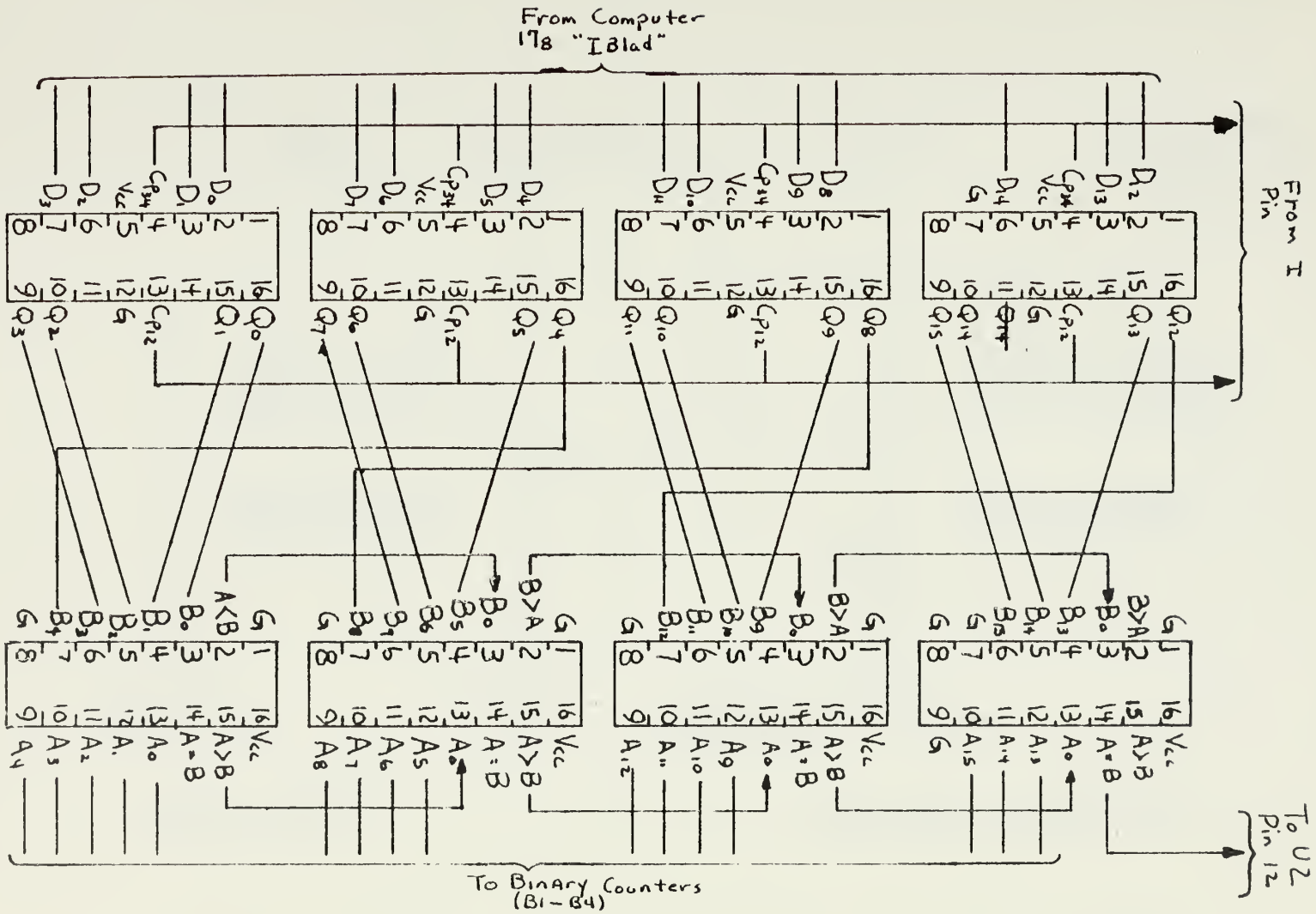


FIGURE 4 (Continued) Circuit Diagram of "ENCODE INTERRUPT" Section

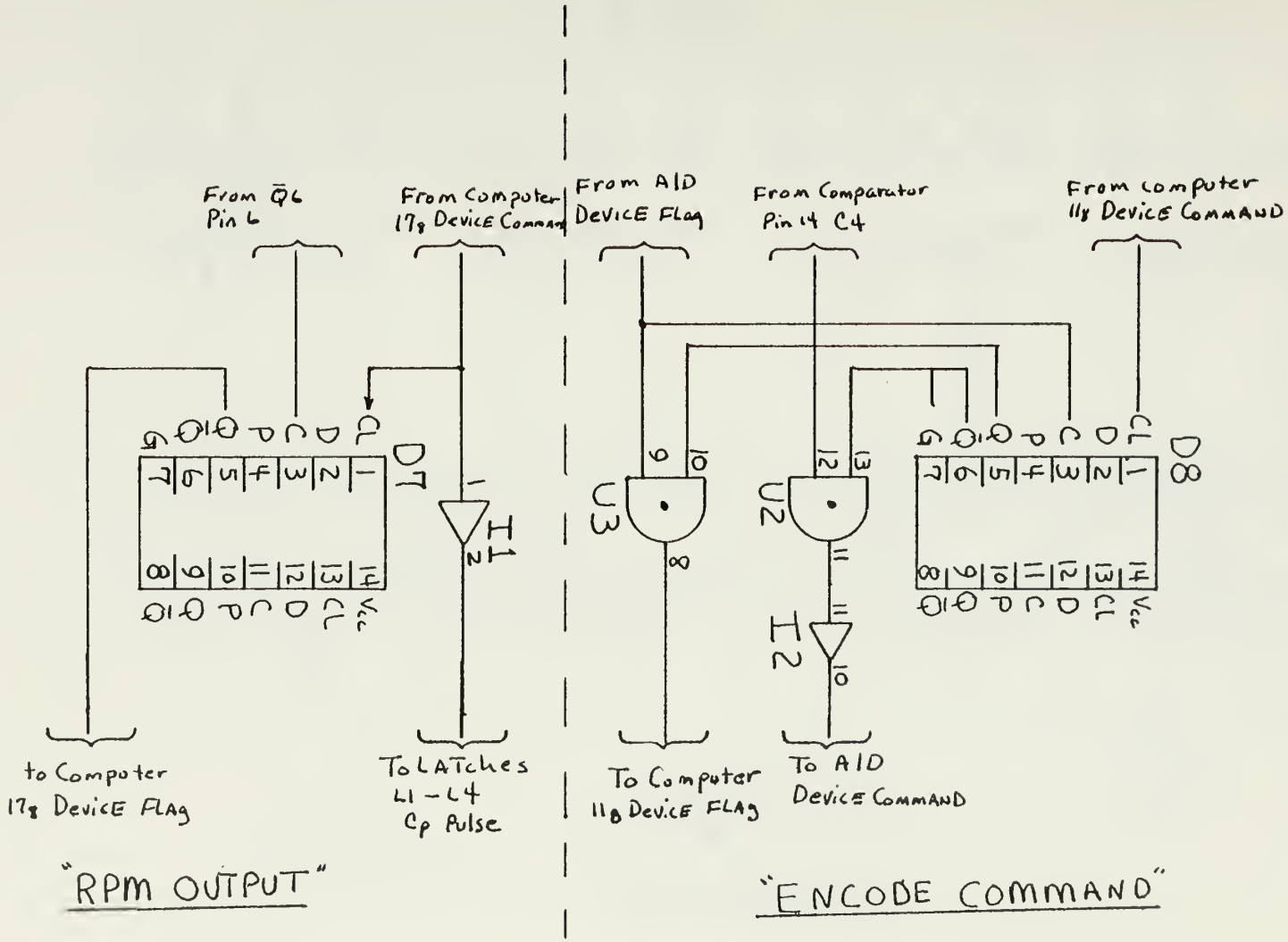


FIGURE 5. Circuit Diagrams of "ENCODE COMMAND" and "RPM OUTPUT" Sections

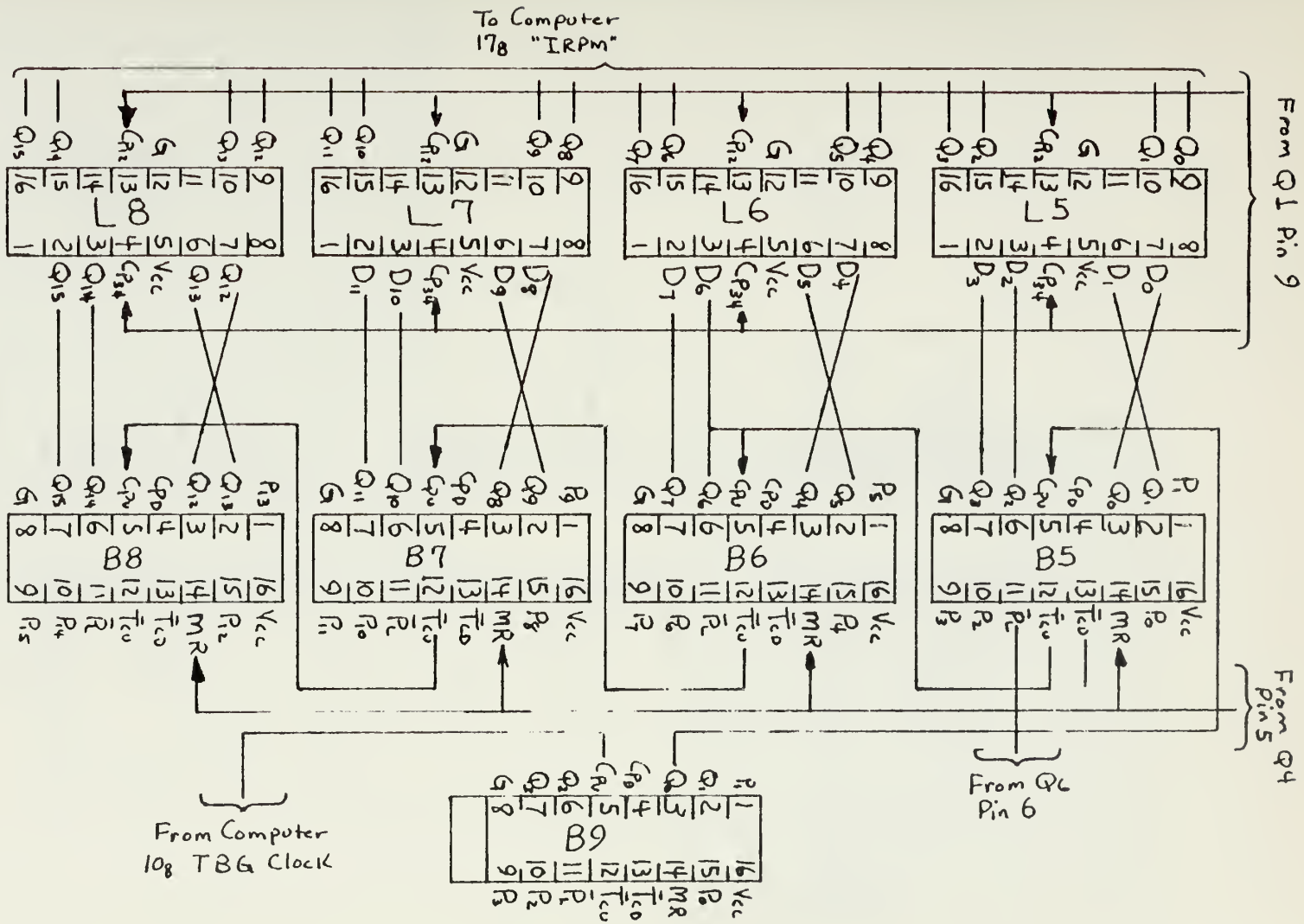


FIGURE 6. Circuit Diagram of "RPM CLOCK" Section

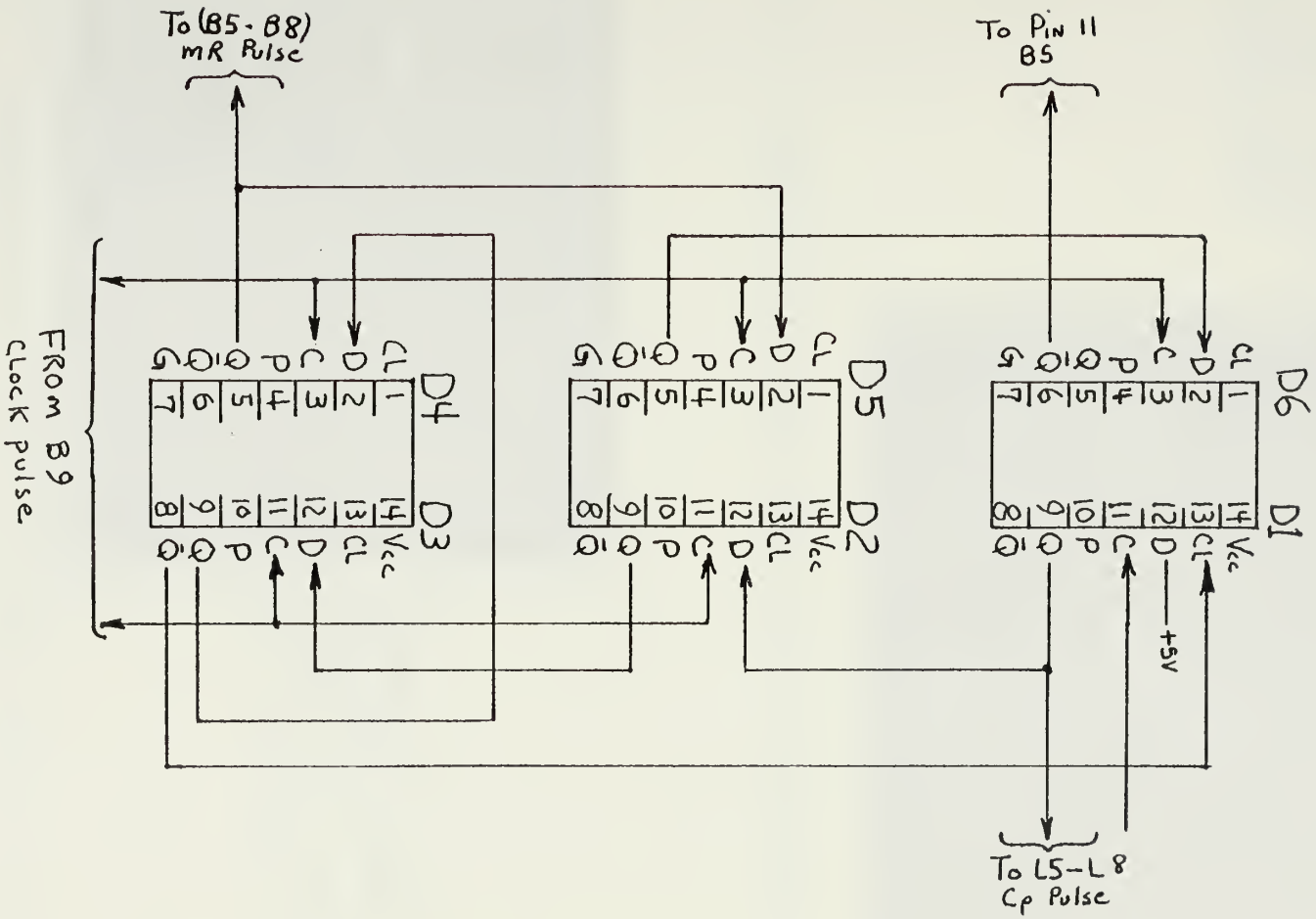


FIGURE 7. Circuit Diagram of "RPM CLOCK CONTROL" AND RESET ("TIME-SET") Section

Upper curve: Optical signal from disk (raw)
Lower curve: Output from "Wave Shaper"

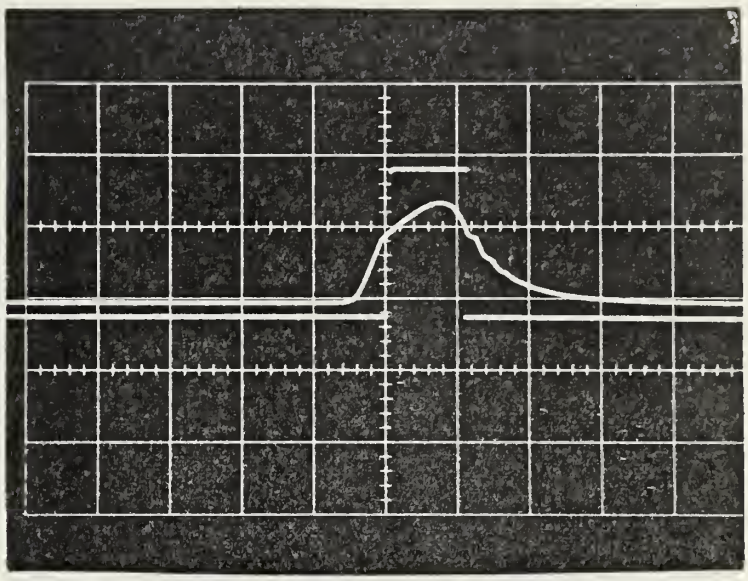
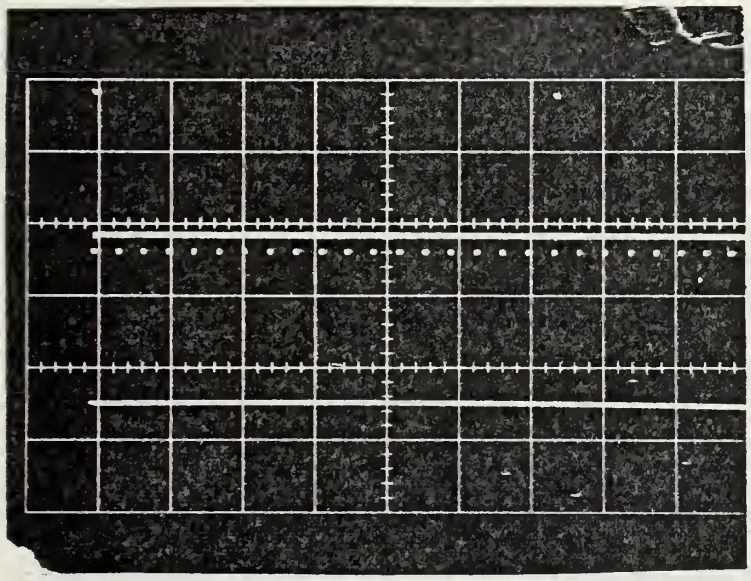


FIGURE 8. Comparison of Wave Shaper Input and Output Signal



Upper curve: Output of 1 rev. trigger shaper
Lower curve: Output of 1 per blade trigger

FIGURE 9. Relationship of One per Blade and One per Rev. Pulses

Upper curve: Output of B10
Lower curve: Output of 1 per blade "Wave Shaper"

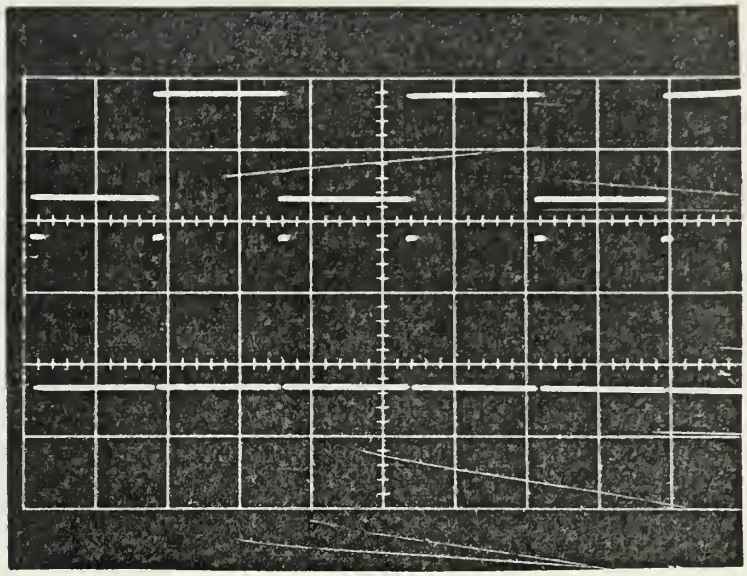
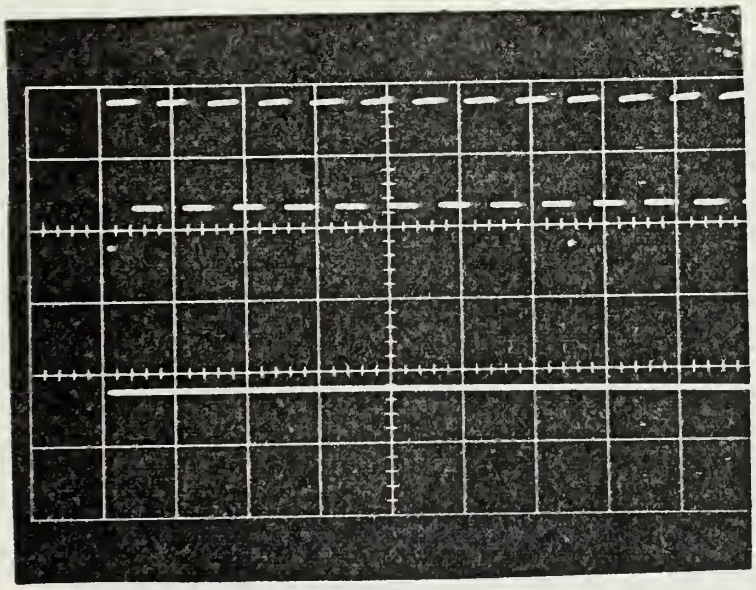


FIGURE 10. Input to and Output from B10 Showing Division by 2



Upper curve: Output of B10 (note, neg. is up)
Lower curve: Output of 1 per rev. "Wave Shaper"

FIGURE 11. Frequency Comparison of One per Rev. and Output of B10

Upper curve: Output of B10
Lower curve: Output of B2

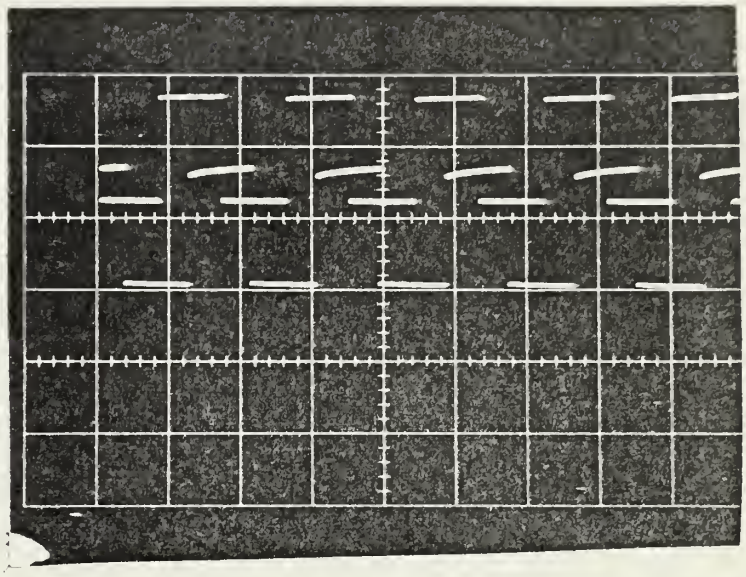
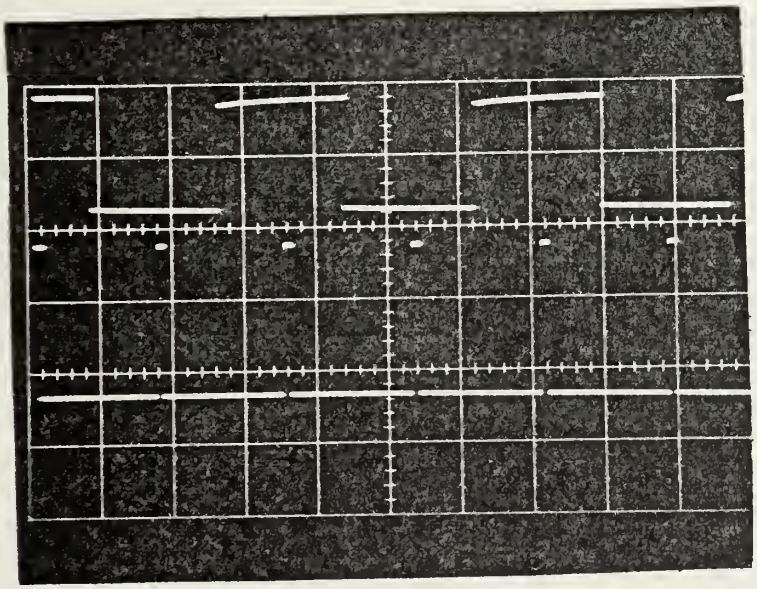


FIGURE 12. Phase Relationship Between Input Signal (B10) and Feedback Signal (B2)



Upper curve: Output of B 2
Lower curve: Output of 1 per blade trigger shaper

FIGURE 13. Phase Shift Between One per Blade and Feedback Signals (B2)

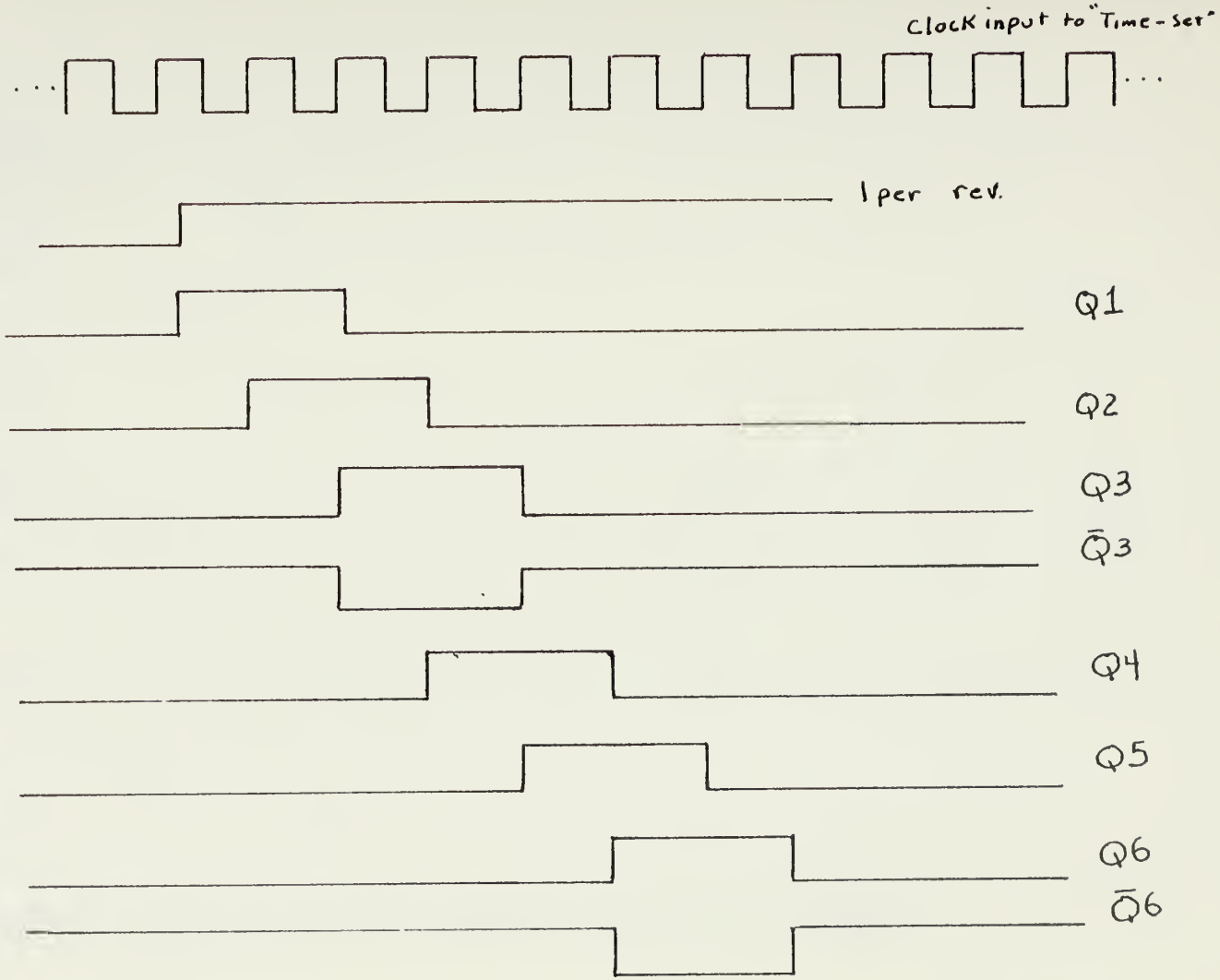


FIGURE 14. Wave Form Relationships for RPM CLOCK CONTROL and RESET ("TIME-SET")

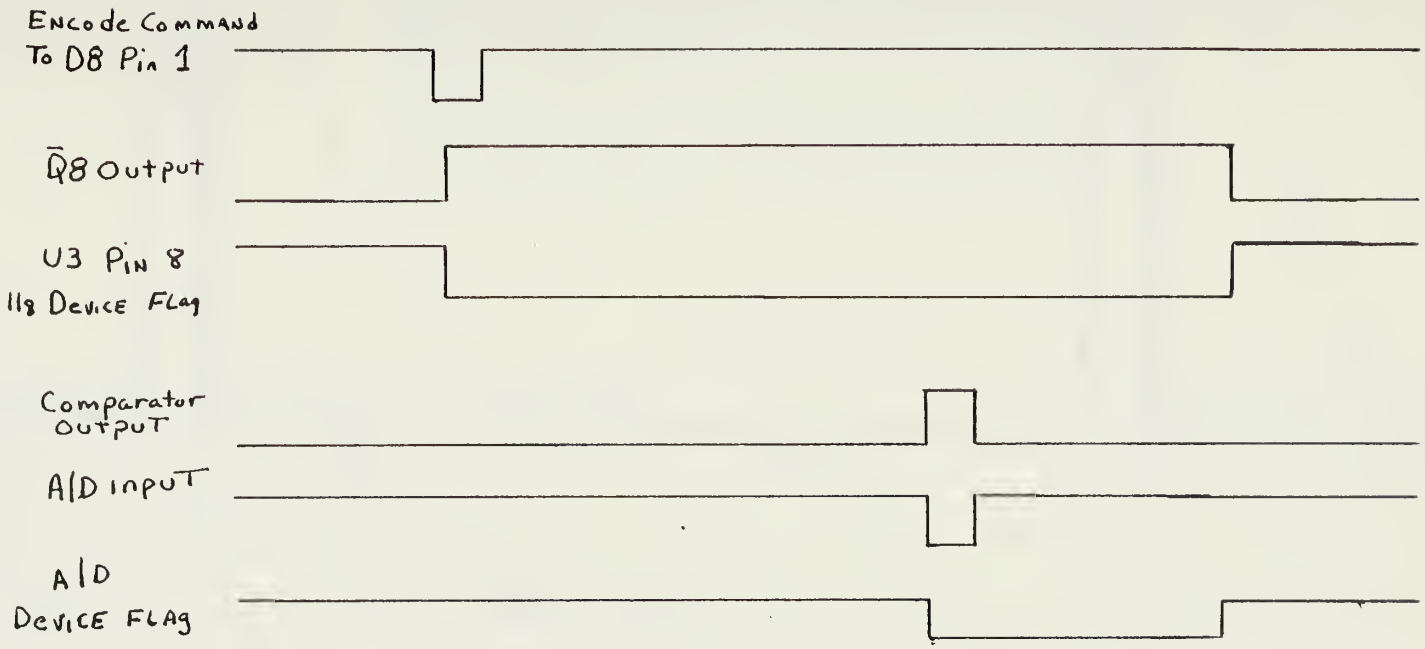


FIGURE 15. Waveform Relationships for ENCODE COMMAND

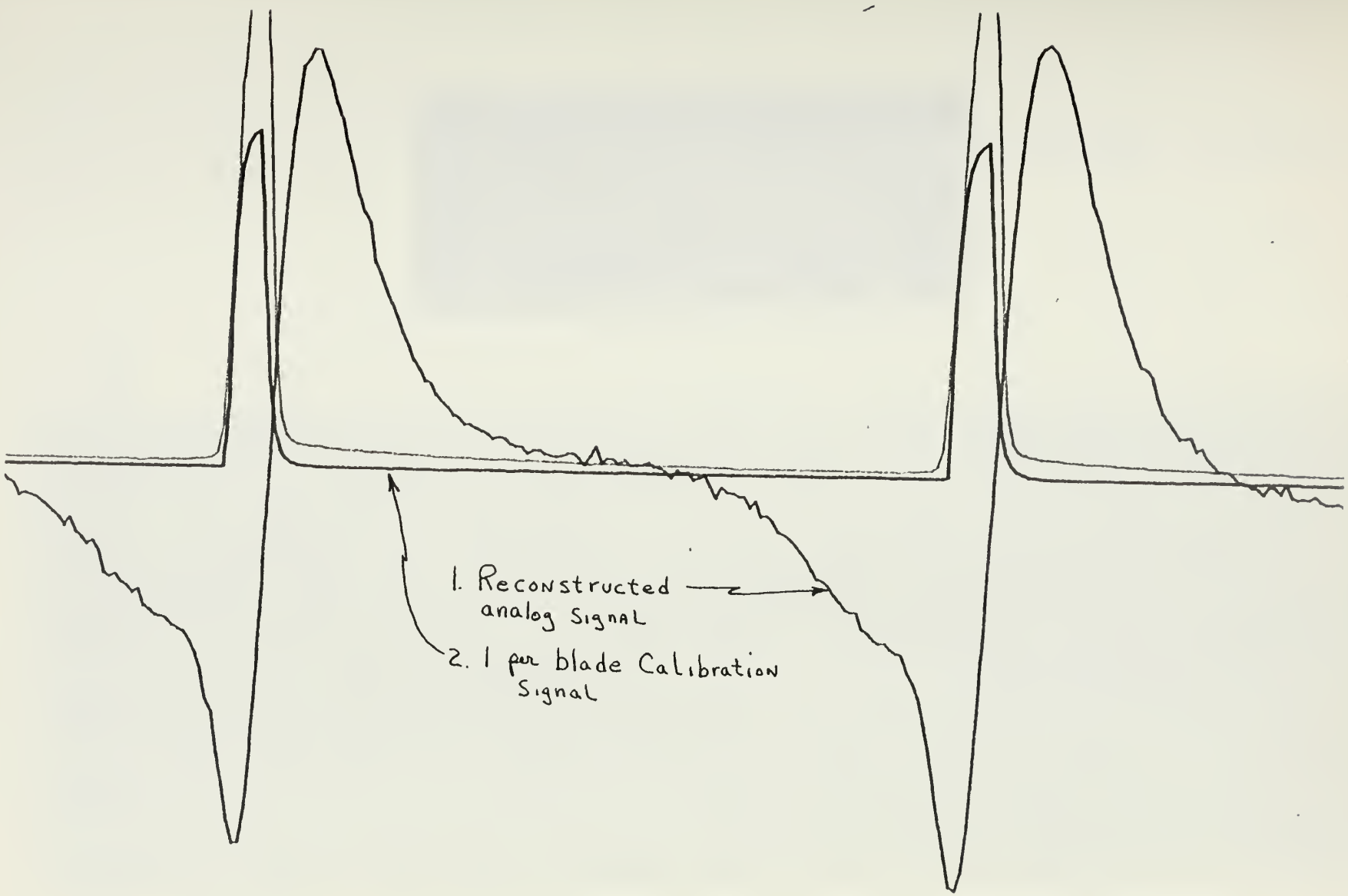
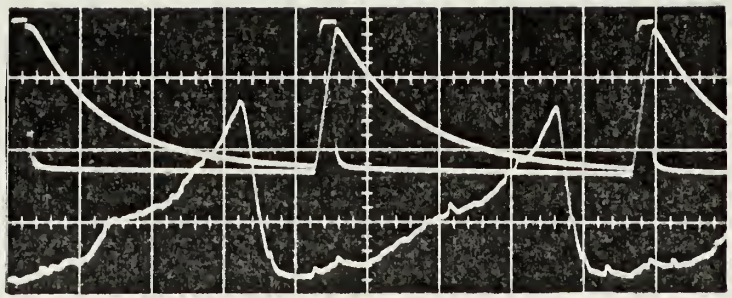


FIGURE 16. Reconstructed Data Taken in a Calibration Test of the Synchronized Sampling System



Time Scale: 50 μsec/DIV

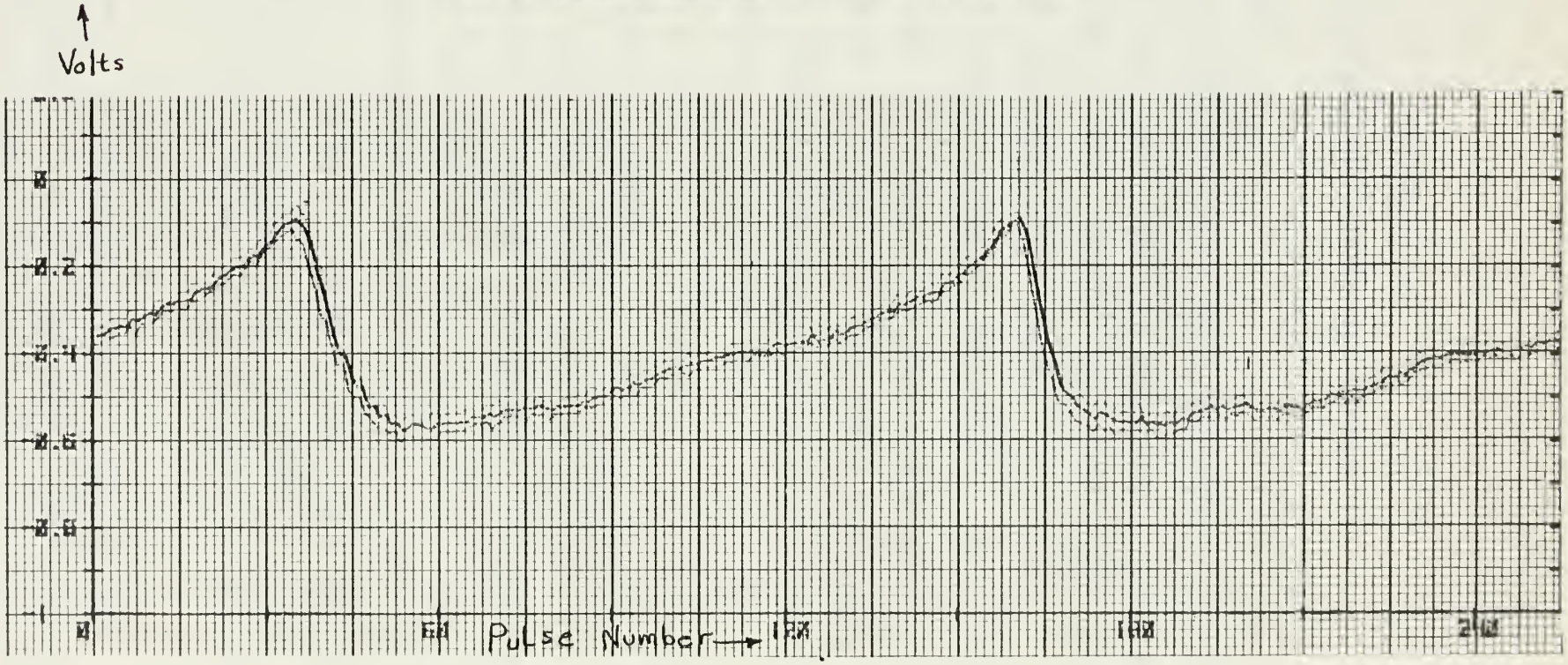


FIGURE 17. Synchronized Sampling Results from the Transonic Compressor

APPENDIX A

RTE-B COMPUTER OPERATING SYSTEM

A1. SYSTEM FEATURES

As presently configured, the RTE-B operating system for the Hewlett-Packard 21MX computer used the following devices: HP2640A CRT Display, Fast paper tape punch, Paper Tape reader, ASR-33 Teletype Unit, HP9830 calculator, HP5610A Analog-to-Digital converter and RPACE. Detailed operating instructions for the individual components can be found in References 3, 4 and 5. The RTE-B operating system is a master program which must first be generated on paper tape. Details of the generation of the RTE-B operating system are contained in Reference 4, Section 7. A record of the system generation is listed in Appendix C.4.

Formally, the operating system used (as in Ref. 1) was the BCS system. However, because of problems encountered in using this paper-tape based system, the worst of which was the time-consuming procedures involved in generating or modifying programs, a change was made to the RTE-B system.

The only major problem associated with RTE-B is the actual generation of the system. Some of the difficulties are as follows:

1. Subroutines: no subroutine can be called other than those that are configured into the operating system at system generation time. A new subroutine must first be configured

into the operating system before it can be called in a program. Reference 4, Section 7 contains instructions for incorporating subroutines into the operating system.

2. Arrays: The arrays that are passed between the main program and the subroutines must be floating point. However, the data that is returned to the subroutine, as a result of calling EXEC (Ref. 3) is an integer array and must be converted to a floating point array. "R5610" is the subroutine that does the array conversion prior to returning the A/D converter data values to the main program.

3. System Generation: The order in which the binary tapes are entered when generating the operating system is important. (The present tapes have been marked in the correct order.)

A2. DATA ACQUISITION

The A/D converter has provisions for recording up to sixteen different channels of analog signals. The method of acquiring data depends upon the control word that the 21MX computer supplies to the A/D converter.

A.2.1. A/D Converter

The HP5610A Analog-to-Digital converter is designed to convert analog voltages into digital values, of ten bits. The maximum data rate at which the A/D converter can operate is 100,000 samples/sec, with an aperture window of 50 nanoseconds (Ref. 3). The A/D converter can be operated in one of two possible modes: in SEQUENTIAL mode all sixteen

channels are sampled in numerical order, in RANDOM mode a single preselected channel is scanned repeatedly. Variations of these two basic modes are provided which allow the A/D converter to be controlled by the 21MX computer or by an external device. Presently the A/D converter is operated in the RANDOM mode under computer control (mode 0) or in free run (mode 4). Reference 3 contains a complete description of the operating instructions.

A.2.2. HP21MX Computer

The HP21MX is a micro-programmable computer having 128 basic instructions and 32K of memory. Complete specifications for the computer and a complete listing of the 128 assembly language instructions are contained in Reference 7.

To the computer, any device attached to its input/output structure is a peripheral device. Associated with each peripheral device is either an HP input/output software "driver" or an assembly language subroutine to control the peripheral devices. Presently, all of the peripheral components, except for RPACE, have a "driver" associated with their input/output location. The driver is assigned to the particular octal location during RTE-B system generation time.

A significant difference between the A/D converter and the other peripheral devices is the speed with which data can be transferred. In order to transfer data at 100

KHz, the data is transferred via Direct Memory Access (DMA), to successive memory locations in the computer as they are received from the A/D converter. However, when using the computer with RPACE to control encoding, this high data transfer rate is not needed because sampling is performed once per revolution, which is less than 500 Hz.

A3. SYSTEM SUBROUTINES FOR DATA ACQUISITION

A.3.1. Introduction

The two subroutines that are used for data acquisition are "R5610" and "RPACE". For listings of the two programs, see Appendix C.1 and C.2. The purpose of "RPACE" is to output a number which can be used to position a pulse anywhere on the perimeter of the rotor. "RPACE" also returns the number of clock pulses that have elapsed between 1 per rev. trigger pulses. "R5610" is used to call and to control the A/D converter and to return to the main program voltage values from the analog input channels.

A.3.2. "RPACE"

"RPACE" has three calling parameters. IBLAD is passed to the subroutine from the main program and IRPM is passed from the subroutine to the main program. IEND is provided as a test parameter. Appendix C.2 contains a program listing of "RPACE". Figure A1 is a flow chart of "RPACE", and illustrates how interrupt subroutines are used in conjunction with RTE-B. As indicated in Figure A1, "RPACE" consists of two parts. The first part, called the

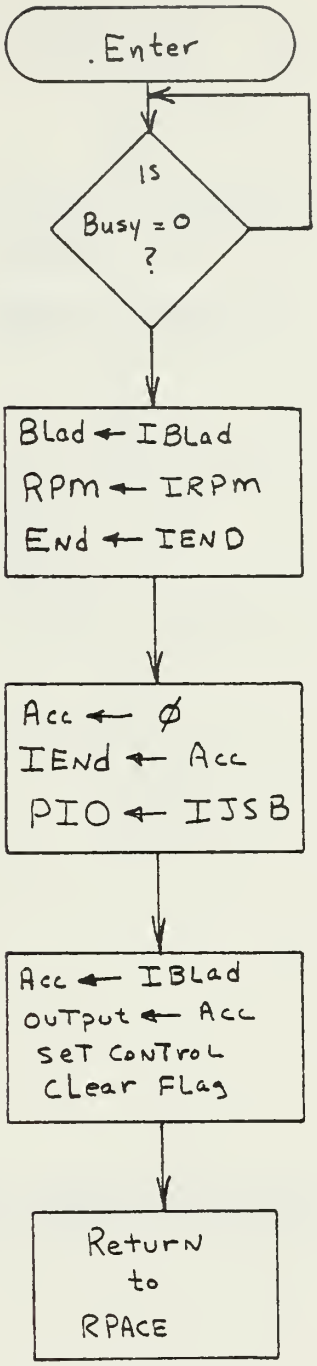


Fig. A1, Flow Chart of "RPACE", Initiator Section

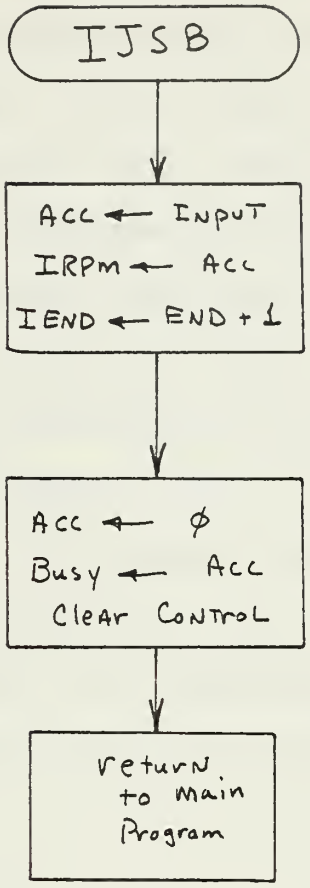


Fig. A1 (Cont'd), Flow Chart of "RPACE", (Continuator Section)

initiator section, functions to output the desired pulse location, as indicated by IBLAD, to the RPACE hardware. It also sets flag called "Busy", which prevents the subroutine from being called again until the present interrupt has been cleared by the continuator section. The second part of "RPACE" is called the continuator section and has the necessary linkage required to jump from the trap cell, octal location 17, to the continuator section of "RPACE". When an interrupt does occur on octal location 17_8, RPACE hardware is signaling to the computer that data is available and is waiting to be transferred to the computer. In addition to accepting the data, the continuator section also clears the "Busy" flag which allows the subroutine "RPACE" to be called again.

A.3.3. "R5610"

Appendix C.1 contains a program listing of "R5610". The calling parameters are analogous to those described in Reference 3 for "R5610". When a subroutine "R5610" is called, it in turn calls another subroutine called "EXEC". The purpose of "R5610" is to convert the integer array IBUFF, which contains the data returned from "EXEC", into a floating point array as required by RTE-B. If it were not for this requirement "EXEC" could be called direct from the Basic program.

As indicated in the listing, RBUFF and RCHAN are both subscripted variables and must be dimensioned in

the main program. Use of the subroutine follows the procedures given in Reference 4 for Basic programs in RTE-B.

APPENDIX B

RPACE CIRCUIT DETAILS

B.1 INTRODUCTION

RPACE was designed to perform two functions; control of the A/D converter and determination of the speed of a rotor based on one revolution. The design is illustrated in the two sections of Figure 2 and in Figures 3 through 7.

The underlying idea in the approach is simply to count a specified number of pulses and to take a sample. The number of pulses generated by the PLL circuit is always a constant but the frequency of the PLL is not. The frequency at which the PLL will operate is given by

$$f_{vco} = 256 * (\frac{f_o}{2})$$

where f_o is the blade passing frequency and f_{vco} is the frequency of the voltage controlled oscillator. The two in the denominator is required because B10 divides the blade passing frequency (f_o) in half.

B.2 WAVE SHAPER

The wave shaper circuit is shown in Fig. 3. The transistors Q_3 and Q_4 are associated with the one per blade optical signal. Transistors Q_5 and Q_6 are used in the one per rev. channel. The two channels are identical: only one channel will be described.

The input to transistor Q_3 is a bell-shaped curve (see Fig. 8) which is transformed by the circuit into the digital pulse shown in Figure 8. Examination of Fig. 8 shows that the output of the optical detector must be greater than .3V (the maximum amplitude of the input waveform is .5V) before transistor Q_3 is gated on. When Q_3 is gated on, it immediately goes into saturation, driving the voltage at R_{15} to approximately .5 volt. This low voltage causes Q_4 to be cut off, thus the output of Q_4 goes to +5V. The reverse of the description occurs when the bell-shaped input waveform drops below approximately .4V and Q_3 is turned off. When Q_3 is gated off, no current flows through the transistor and the entire +5V is dropped across Q_3 (i.e. the junction of R_{15} and R_{17} is at +5V). This +5V potential is supplied to the base of Q_4 through the biasing resistors R_{17} and R_{16}. The bias level for Q_4 is adjusted by the values of R_{16} and R_{17} so as to allow Q_4 to conduct when Q_3 is cutoff. Thus, the voltage out of Q_4 is at a zero level.

The above circuit was necessary to convert the analog, low voltage signal into a digital pulse with sharp edges suitable for TTL connections.

The output of the two channels is connected to a buffer/amplifier so as to isolate the optical detectors from the rest of the circuit. From the buffer, the 1 per rev. signal is fed to D1 of the "Time-set" section and the 1 per blade is fed to B10. B10 then converts the 1 per blade pulse into

a square wave with one period for every 2 blades (Fig. 10). This conversion is necessary because the analog phase comparator of the PLL requires two waveforms of similar shape and frequency for stable and reliable operation. The phase comparator of the PLL circuit will operate with a pulse input from B10 and a square wave from B2, but it is then very unstable and will not track input frequency variations resulting from rotor speed changes.

B.3 ENCODE INTERRUPT

This section is composed of the components required to compute the time delay requested by the program. The delay is accomplished by counting the number of pulses that have elapsed following the start of the 1 per rev. signal. The unique feature of this circuit is that the number of pulses between any two adjacent input pulses is a constant. If the speed of the rotor is 100 revolutions per second, then the number of pulses between adjacent blades is 126. If the speed of the rotor is 500 revolutions per second, then the number of pulses will still be 126. Thus the frequency of the VCO is tracking the blade passing frequency (f_o), and is always 126 f_o.

The two inputs into the PLL circuit are the feedback from B2 and the blade passing frequency ($f_o/2$). Internal circuitry in the PLL is able to generate an analog signal, based on a phase comparison between the two inputs, which is used to control an oscillator (VCO). The frequency of

the VCO is proportional to the analog voltage (steering voltage) generated and will track the incoming frequency across a limited frequency range (see Reference 6 for a detailed circuit description).

The output of the VCO, which is at a multiple of the input frequency, is fed to a binary counter via a coupling circuit. The coupling circuit functions in the same manner as the one described for the wave shaper circuit. Again, the purpose of the circuit is to convert the output of the VCO, which varies between 0 and +15V, into a TTL compatible signal whose voltage range is between 0 and +5V.

The binary counters (B1 and B2) are used to divide the incoming VCO frequency by 256 (2^8) and feed back a signal whose frequency is the same as that input from B10, but shifted in phase by 270 degrees. Figure 12 shows the 270 degree phase shift that occurs between the two inputs. Figure 4 is the circuit showing the various connections required and used by the binary counters (B1 to B4). Cascade operation of the counters requires that the "Terminal Count Up" line (T_{cu}) be connected to the "Clock Pulse Up" (C_{pu}) line of the next unit. This line is called the "Carry Line" and information on it results from an overflow condition in the counter. If all four carry lines are connected as described above, then the four 4-bit counters have been cascaded and they can count up to (2^{16}-1) before an overflow condition will occur. Connection of the carry line is not required for normal operation, only for cascade

operations. Referring to Figure 4, the carry line from B2 is connected to B3 via an "AND" gate. The other input to the "AND" gate is bit 15 from the 21MX computer. Thus, by programming bit 15 high, the carry will be connected to B3 and normal cascade operation of the binary counters will result. However, if bit 15 is low, then the carry is blocked and the two counters (B3 and B4) will never count because their C_{pu} lines never go low (see Reference 6 for detailed input requirements for normal operation). It should be noted that B3 and B4 are only associated with counting the number of blades that have passed. The number of blades that have passed is the information that is transferred via the C_{pu} line through U1 to B3. Thus, if U1 is open, every pulse generated between 1 per rev. pulses will be counted by the counters. If U1 is closed, only the pulses between any two adjacent blades will be counted, with B1 and B2 resetting to zero for the next set of blades. Thus, in this mode, B1 and B2 will reset nine times; once for each set of blades.

Referring to Figure 4, the computer output card is connected to latches L1 to L4. When directed by the software, the binary number representing IBLAD is strobed through the latches to the digital comprator (C1 to C4). The purpose of the comparator is to compare the two words, A and B, and output pulses when a certain relationship exists between the two words. Possible outputs from the comparator are

A > B, A < B and A = B. The circuit design used in RPACE uses only A = B. Thus, when the binary counters B1 to B4 have counted the programmed number of pulses (i.e. both sides of the comparator are the same), a pulse is generated. The location of this pulse is determined by the mode of operation of the counters. If bit 15 is high, then the pulse will occur once for every revolution of the wheel. If bit 15 is low, then the pulse will occur nine times per revolution of the wheel. The output of the comparator is connected to the input of U2 of the "Encode Command" section.

B4. ENCODE COMMAND

Referring to Figure 5, the comparator output, when "Anded" with the output of D8, is used to trigger the A/D converter. Initially, the 21MX computer commands that a data sample be taken via the device command line. This command can occur at any point on the perimeter of the wheel unless additional steps are taken to ensure that the samples are acquired only where desired.

Thus, in light of the above, the encode pulse from the computer (which has negative logic) is intercepted by D8. The negative pulse on the clear line of D8 causes the $\overline{Q}$ output of D8 to be set to a high and gates U2 on. It should be noted that normally U2 is gated off, thus allowing no comparator pulses to be supplied to the A/D converter and preventing inadvertent data acquisition before the computer

is ready to accept the data. When $\overline{Q}$ goes high, then Q, which is the complement of $\overline{Q}$, must go low. This low signal is then "Anded" with the data available line of the A/D converter, which is always high unless it is working. In this manner, the computer is tricked into believing that the A/D has been set, and is busy taking a data sample. But, in reality the A/D converter is still awaiting an encode command and its data ready flag is still high. Meanwhile, the computer has set up an interrupt linkage which links the A/D converter I/O board to the subroutine that will process the interrupt. The computer then continues execution of the main program.

In the interim, the comparator has been continually comparing the binary counter to the programmed word. When they agree, the comparator generates a pulse which is connected to the A/D converter via U2. The inverter (I2) converts the positive logic pulse output of the comparator to a negative logic pulse as required by the A/D converter. Upon receipt of the encode command, the data ready flag of the A/D converter goes negative. The data ready flag is connected to U3 and to the clock input of D8 (see Reference 6 for a detailed description of delay flip-flops). Ten microseconds later, the A/D has completed conversion of the analog signal and is ready to transfer data. At this time, the data ready flag goes to a high, causing D8 to change states (D8 is a positive edge-triggered device) and driving Q high. U3

now has two inputs that are high causing its output to also go high, thereby generating an interrupt. The computer then services this interrupt and clears its flags. The cycle is now ready to repeat itself.

B5. RPM CLOCK

The RPM clock of RPACE is designed to receive four inputs and to generate two output signals. Referring to Figures 2 and 6, one of the input signals comes from the 21MX/TBG clock and the other three come from the "Time-set" section. The physical location of the 21MX (1MC) clock is on the Time Base Generator (TBG) card, which is located in Octal slow 10_8 of the 21MX computer. Binary counter (B9) allows the selection of 1/2, 1/4, 1/8 or $1/16^{th}$ of the input frequency. It thus divides the TBG clock frequency & provides flexibility in selecting the optimum frequency for the particular rotor speed, such that all 16 bits will be used without causing overflow. The higher the speed of the rotor, the higher the clock frequencies needed.

The optimum clock frequency for a given rotor speed can be calculated using

$$f_c = S_r * (2^{16} - 1) \qquad B(1)$$

where

f_c = optimum clock frequency for the binary counters, cps (cycles per second)

S_r = rotor speed, rps (revolutions per second)

and

$2^{16}-1$ = maximum value in the computer word for the 21MX.

If the rotor speed was 500 c.p.s., for example then the value for f_c would be 32.7×10^6 cps. But, because the TBG clock frequency is set at 1×10^6 cps, then the TBG clock input would be connected directly to B5, and B9 would not be used.

However, in the case where S_r is 5 cps, the optimum clock frequency from Eq. B(1) would be $.33 \times 10^6$ cps, and the TBG clock frequency must be reduced. The frequency selected for the present range of rotor speeds to input to B5 was 250,000 cps, which was obtained by selecting 1/4 of the input frequency at B9.

The binary counters (B5 to B8) are cascaded in the same manner as is described in the "Encode Delay" section. The three other inputs required to the RPM clock and their functions are as follows:

<u>To the Binary Counters</u> (B5-B8)

Master Reset (MR): A positive pulse on this line causes the binary counters (B5-B8) to reset their data out lines to zero.

Parallel Load ($\overline{P}_L$): A negative pulse (negative logic) on this line transfers the information stored on the parallel data lines to the data out lines.

To the Latches (L5-L8)

The latch requires a positive pulse on its clock input line to transfer data to its data out line. While the clock input is high, the latch outputs will follow the inputs and be disconnected when the clock goes low (Refs. 6 and 8 contain detailed descriptions of these devices).

Naturally, the binary counters (B5 through B8) are not allowed to reset prior to the information being transferred to the latch outputs. When the information has been transferred to the latch outputs and the latch clock input is again low, then the counters are cleared. After the counters have been cleared, they are preset by the $\overline{P}_L$ line to the pulse count that would have resulted had not the counter been reset. This sequence control is the function of the "Time-set" section.

B6. TIME-SET

The waveforms from each delay flip-flop are shown in Figure 14. The master pulse can occur at any point in the clock cycle. Because the delay flip-flops are positive-edge sensitive, they will only change states when their clock inputs go from a low to a high. Thus, D1 will change states when the master pulse arrives. As indicated in

Figure 7, all of the data inputs are connected to the Q outputs of the previous stage. When Q1 is low, which occurs prior to the arrival of the Master pulse, the Q's for the other delay flip-flops will also be low.

By careful study of the waveforms, it can be shown that the sequence of events, as described, does occur. The reason for using six separate delay flip-flops was to ensure that the three functions performed by this section did not overlap. Namely, the clock pulse input of the latches (L5-L8) must not overlap the MR pulse of the binary counters (B5-B9). If overlap did occur, then the pulse count of the TBG clock would be lost, because the latch outputs will follow their inputs during the overlap portion of the two waveforms. Figure 14 shows that one clock pulse is used to separate the MR and latch clock input pulses. The MR and $\overline{PL}$ do have some overlap. However, this overlap is not critical, as long as the MR occurs prior to the $\overline{PL}$, and that the MR line is low during the latter portion of the $\overline{PL}$ pulse. Again, Figure 14 waveforms indicate that the two waveforms do overlap. The output Q1 is used by latches (L5-L9), and Q4 does the clearing of the counters. Because the $\overline{PL}$ line to the counters uses negative logic, $\overline{Q6}$ is used to preset the counters, and is also used to indicate to the 21MX computer, via D7 of "RPM Output" section, that data is available for transfer.

B7. RPM OUTPUT

This section is used to interface the command and control flags of the 21MX computer to RPACE. When the subroutine "RPACE" is called by the main program, IBLAD is sent to the latches (L1 to L4). The data remains on the input lines until a device command instruction is issued by the computer, at which time it is strobed (pulsed) through the latches (L1 to L4) via I1. The device command pulse also sets Q7 (Figures 4 and 5) to a low, thus setting an interrupt flag. Nothing else happens to D7 until the occurrence of another master blade pulse from the "wave shaper" section. Upon arrival of this pulse, the events described in Section B6 occur resulting in a negative pulse being transmitted to the clock input of D7. The pulse from D6 causes Q7 to go high which, in turn, forces the computer to service the interrupt, as described in the previous sections.

APPENDIX C

COMPUTER PROGRAM LISTINGS

Program listings are given for the following:

C.1. "R5610"

C.2. "RPACE"

C.3. CALIBRATION TEST PROGRAM

C.4. TRANSONIC COMPRESSOR PROGRAM

C.5. RTE-B OPERATING SYSTEM GENERATION.

```
0001  FTN,L
0002        SUBROUTINE R5610(IDRT,RBUF,N,ICHAN,ICODE,RCHAN)
0003  C
0004  C
0005  C     DATE: 761022      TIME: 1800
0006  C
0007  C
0008        DIMENSION RBUF(1),RCHAN(1),ITEMP(2),IREG(2)
0009        EQUIVALENCE (RTEMP,ITEMP(1)),(REG,IREG(1))
0010  C
0011  C
0012  C     CHECK ICHAN.   IF ICHAN < 0, THEN CREATE TEST DATA ONLY.
0013  C     OTHERWISE PASS ICHAN THROUGH TO THE 5610 DRIVER.
0014  C
0015        IF (ICHAN)  200,10,10
0016  C
0017  C     CALL THE 5610.
0018  C
0019  10    REG = EXEC(1,IDRT,RBUF,N,ICHAN,ICODE)
0020  C
0021  C     ON RETURN FROM THIS EXEC CALL, THE A REGISTER CONTAINS THE DEVICE
0022  C     STATUS AND THE B REGISTER CONTAINS THE TRANSMISSION LOG.
0023  C     THE NUMBER OF READINGS REQUESTED (N) SHOULD EQUAL THE TRANSMISSION
0024  C     LOG.   IF THESE VALUES ARE UNEQUAL, RCHAN(1) IS SET TO 99, RCHAN(2
0025  C     IS SET TO THE STATUS(A REGISTER).   RCHAN(3) IS SET TO THE
0026  C     TRANSMISSION LOG (B REGISTER).  RCHAN(4) IS SET TO "N".
```

```
0029          IF (IREG(2)-N) 20,300,20
0030    20    CONTINUE
0031    C     RCHAN(1) = 99.0
0032    C     RCHAN(2) = FLOAT(IREG(1))
0033    C     RCHAN(3) = FLOAT(IREG(2))
0034    C     RCHAN(4) = FLOAT(N)
0035          GO TO 300
0036    C
0037    C
0038    C     CREATE DUMMY DATA FOR USE IN TESTING.
0039    C
0040    200   ITEMP(1) = 140001B
0041          ITEMP(2) = 140002B
0042          DO 220 I = 1,N/2+1
0043    220   RBUF(I) = RTEMP
0044    C
0045    C
0046    C
0047    C     CONVERT THE CHANNEL ADDRESSES  AND PLACE IN THE RCHAN ARRAY.
0048    C
0049    C
0050    C
0051    300   J = 1
0052          DO 330 I = 1,8
0053    310   RTEMP =RBUF(I)
0054          RCHAN(J) = FLOAT(IAND(ITEMP(1),17B))
0055          J = J + 1
0056          IF (N - J) 400,320,320
```

```
0057  320   RCHAN(J) = FLOAT(IAND(ITEMP(2),17B))
0058        J = J + 1
0059        IF (N - J) 400,330,330
0060  330   CONTINUE
0061  C
0062  C
0063  C     NOW CONVERT THE DATA READINGS.
0064  C
0065  C
0066  C
0067  C
0068  400   J = N
0069        IF (((N/2)*2)-N) 410,420,410
0070  410   RTEMP = RBUF(N/2+1)
0071        RBUF(N) = (1.0) * FLOAT(IAND(ITEMP(1),177700B))/32768.
0072        J = N-1
0073  420   M = J/2
0074        DO 500 I = 0,M-1
0075        RTEMP = RBUF(M-I)
0076        RBUF(J-(I*2)) = 1.0 * FLOAT(IAND(ITEMP(2),177700B))/32768.
0077        RBUF(J-(I*2)-1)=1.0 * FLOAT(IAND(ITEMP(1),177700B))/32768.
0078  500   CONTINUE
0079        RETURN
0080        END
```

** NO ERRORS** PROGRAM = 00310 COMMON = 00000

```
ASMB,R,B,L,T
      NAM   RPACE
      EXT   .ENTR
      ENT   RPACE
PIO   EQU   17B
IBLAD NOP
IRPM  NOP
IEND  NOP
RPACE NOP
      JSB   .ENTR
      DEF   IBLAD
      LDA   BUSY
      SZA
      JMP   *-2
      ISZ   BUSY
      LDA   IBLAD
      STA   BLAD
      LDA   IRPM
      STA   RPM
      LDA   IEND
      STA   END
      CLA
      STA   END,I
      LDA   IJSB
      STA   PIO
      LDA   BLAD,I
      OTA   PIO
      STC   PIO,C
      JMP   RPACE,I
```

```
IJSB   JSB   LINK,I
       ORB
LINK   DEF   CONT
       ORR
CONT   NOP
       STA   SAVEA
       LIA   PIO
       STA   RPM,I
       OTA   01B
       ISZ   END,I
       CLA
       STA   BUSY
       LDA   SAVEA
       CLC   PIO
       JMP   CONT,I
BUSY   NOP
BLAD   NOP
RPM    NOP
END    NOP
SAVEA  NOP
       END
       END$
```

C.3 CALIBRATION TEST PROGRAM

```
10   DIM A[3,255],B[3,255],C[255],D[255]
20   FOR J=1 TO 3
25   FOR I=1 TO 255
30   LET A[J,I]=0
35   LET B[J,I]=0
40   NEXT I
45   NEXT J
50   FOR I=1 TO 255
55   LET N1=10
60   LET A1=8
65   LET A8=33024+I
70   FOR K=1 TO 3
85   FOR J=1 TO N1
90   PPACE(7,A2,A3)
100  PSE10(7,C[I],1,A1,0,D[I])
105  LET A[K,I]=C[I]+A[K,I]
110  LET B[K,I]=A8+B[K,I]
115  NEXT J
117  LET A1=A1+1
120  NEXT K
122  NEXT I
125  FOR I=1 TO 255
130  FOR K=1 TO 3
135  LET A[K,I]=A[K,I]/N1
140  LET B[K,I]=2500PGX(B[K,I]/N1)
141  PRINT# 8;A[K,I]
150  PRINT# 8;B[K,I]
170  NEXT K
180  NEXT I
190  GOTO 20
```

C.4 TRANSONIC COMPRESSOR PROGRAM

```
5   DIM A[101,16],B[101,16],C[3],D[3],E[4,255]
10   PRINT "ENTER---TEST#,  MONTH,DAY,YEAR"
15   INPUT I1,I4,I3,I5
20   PRINT "ENTER---RUN#,EXPERIMENT#"
25   INPUT I2,I9
30   PRINT "ENTER---MODE#, CHANNEL#, TRANSDUCER#"
35   INPUT I6,A1,T1
40   PRINT# 8;I1
45   PRINT# 8;I2
50   PRINT# 8;I3
55   PRINT# 8;I4
60   PRINT# 8;I5
65   PRINT# 8;I6
70   PRINT# 8;I9
75   PRINT# 8;T1
80   PRINT# 8;A1
85   IF I6=0 THEN 150
90   IF I6=4 THEN 100
95   PRINT "WRONG MODE#"
98   GOTO 30
100   PRINT "ENTER---SAMPLES/CHANNEL(NOT>1616)"
105   INPUT I7
110   PRINT# 8;I7
115   R5610(7,A[1,1],I7,A1,I6,B[1,1])
120   FOR I=1 TO 16
125   FOR J=1 TO 101
130   PRINT# 8;A[J,I]
135   NEXT J
140   NEXT I
```

```
145  GOTO 40
150  PRINT "ENTER BLADE#, SAMPLES/POINT"
155  INPUT N2,N1
157  PRINT# 8;N1
158  PRINT# 8;N2
160  INPUT N1,N2
170  LET A3=0
180  LET A6=32768+N2*256
190  FOR I=1 TO 255
200  LET A3=A6+I
215  LET R=0
220  FOR J=1 TO N1
225  RPACE(A3,A4,A5)
230  R5610(7,C[1],1,A1,0,D[1])
240  LET B[J,1]=C[1]
245  R5610(7,C[1],1,15,0,D[1])
247  LET R=R+C[1]
250  NEXT J
255  LET R=R/N1
260  LET S=0
270  LET M=0
280  FOR J=1 TO N1
290  LET M=M+B[J,1]
300  NEXT J
310  LET M=M/N1
320  FOR J=1 TO N1
330  LET S=S+((B[J,1]-M)*(B[J,1]-M))
340  NEXT J
350  LET S=SQR(S/(N1-1))
```

```
360  LET E[1,I]=M
370  LET E[2,I]=S
380  LET E[3,I]=A4
385  LET E[4,I]=R
390  NEXT I
400  FOR J=1 TO 4
410  FOR I=1 TO 255
411  PRINT E[J,I]
412  GOTO 430
420  PRINT# 8;E[J,I]
430  NEXT I
440  NEXT J
470  GOTO 20
```

```
* TURN ON PUNCH *

# OF 2313'S?
-
0

6940 ?
-
NO

* CONFIGURATION CONSTANT PHASE

* ENTER INSTRUMENT CONFIGURATION CONSTANTS
-
/E

* TEAR OFF INSTRUMENT TABLE TAPE *

* PUT B&M TAPE IN READER *

* THEN PRESS RUN *

LIST B&M TABLE?
-
NO

* DELETE SUBROUTINES *
-
NOPE
```

```
-
PCALU(
-
SCALU(
-
PACEP(
-
ALRDV(
-
ALSQV
NAME NOT FOUND
-
ALSQV(
-
/E

* DELETE FUNCTIONS *
-
/E

* DELETE DEVICES *
-
/E

* ADD SUBROUTINES *
-
IBSET(I,I,T(
IBSET(I,I,P),SUB=PRSET
-
IEOR(I,I,P),SUB=PLOP
-
IOR(I,I,P),SUB=PLOP
```

```
IAND(I,I,I),SUB=IAND
IAND(I,I,I),SUB=IAND
-
INOT(I,I),SUB=INOT
-
ISHFT(I,I,I),SUB=ISHFT
-
IBTST(I,I,R),SUB=IBTST
-
IBCLR(I,I,R),SUB=IBCLR
-
ISETC(R,R),SUB=ISETC
-
B5610(I,R,I,I,I,R),SUB=B5610
-
TRACE(I,V,V),SUB=TRACE
-
/E

* ADD FUNCTIONS *
-
/E

* ADD DEVICES *
-
CRT=1,SUB=CRT1
-
PUNCH=4,SUB=PNCH1
-
TTY=6,SUB=TTY1
-
/E
```

```
WHAT IS LIST IQ VICI LIPE LINCTIP
-
??

LIST B&M TABLE?
-
YES

*LIST OF SUBROUTINES *
TIME( R), SUB=TIME
SETP( I,I), SUB=SSETP
START(I,R), SUB=SSTRT
DSABL(I), SUB=DSABL
ENABL(I), SUB=ENABL
TRNON(I,R), SUB=TRNON
IBSET(I,I,R), SUB=BRSET
IEOR( I,I,R), SUB=BEOR
IOR(I,I,R), SUB=BIOR
IAND( I,I,R), SUB=BAND
INOT( I,R), SUB=BNOT
ISHFT(I,I,R), SUB=BSHFT
IBTST(I,I,R), SUB=BBTST
IBCLR(I,I,R), SUB=BBCLR
ISETC(R,R), SUB=ISETC
[illegible](I,R,I,I,I,R), SUB=B5610
[illegible](I,V,V), SUB=BPAGI
```

```
* LIST OF FUNCTIONS *
TAN , SUB=ETAN
SIN ,ERROR, SUB=SIN
COS ,ERROR, SUB=COS
TAH ,ERROR, SUB=TAN
ATN , SUB=ATAN
LN,ERROR, SUB=ALOG
EXP ,ERROR, SUB=EXP
ABS , SUB=ABS
SQR ,ERROR, SUB=SQRT
INT , SUB=FINT
MOD , SUB=FMOD
SGN , SUB=FSGN
SWP , SUB=ESWP
IFDR, SUB=IFDR

* LIST OF DEVICES *
TAPE = 5, SUB=INPUT1
CRT = 1, SUB=OUT1
PUNCH = 4, SUB=PNCH1
TTY = 6, SUB=TTY1

* END FTN-D TABLE GEN *
```


C.5

C.5

```
  [illegible]                    04761    [illegible]    [illegible]    00117
-
RELOCATE
  DVR56                          05771    06222    00120    00125
-
DISPLAY UNDEFS
NO UNDEFS.
-
END

STARTING ADDRESS 00002
NO UNDEFS

ECT TRL.


ECT   1 =?
11,DVR56,[illegible]

ECT   2 =?
[illegible]1 ,[illegible] 01,T=0000

ECT   3 =?
12,DVR 02,[illegible],T=0000

ECT   4 =?
[illegible]01 00,[illegible],T=[illegible]

ECT   5 =?
1[illegible],DVR00,[illegible]
```


C.5

```
[illegible]
IC, [illegible] 02, L, I = [illegible]

IOT 7 =?
MID [illegible]

INT, TDL
LU# [illegible]

 1 = IOT #?
 ; 1

 2 = IOT #?
 0

 3 = IOT #?
 0

 4 = IOT #?
 3, 0

 5 = IOT #?
 3, 4

 6 = IOT #?
 4 [illegible]

 7 = IOT #?
 1
```

```
9 = 1[? #?
END
INT TNL
11,POT,1
12,EQT,9
13,POT,3
14,POT,4
15,EQT,5
16,POT,6
END

"ID SVC?
1
```

C.5

```
[illegible]
0

DEL [illegible] LIB
-
RELOCATE
  TELIB          06434   06713   00127   00131
-
RELOCATE
  ..COM          06714   06713   00132   00131
-
END

STARTING ADDRESS 00000
DO FILES

# [illegible] IN CORE?


DEL USER[illegible]
-
RELOCATE
  BASIC          06714   15057   00132   00650
-
RELOCATE
  MEM            15040   15345   00051   00053
  [illegible]TEL         15346   15534   00051   00735
```

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SEARCH				
RS610	15606	16273	00756	00763
SEARCH				
BSCHL	16274	17453	00764	01016
RSHPU	17454	21561	01017	01210
CMHES	21562	22063	01211	01227
[illegible]	22064	22224	01230	01231
LITCT	22225	22415	01232	01231
FTVAP	22414	22457	01232	01233
[illegible]	22460	22500	01234	01253
INDCT	22501	22506	01234	01[illegible]
ISCH	22507	22509	01234	01[illegible]
FIFP	[illegible]	22547	01234	01[illegible]
TAE	[illegible]	22571	01234	01[illegible]
FLCOM	22572	22722	01241	01244
LITFY	22723	23042	01245	01253
WAIT	23043	23075	01254	01255
CALLS	23076	23245	01256	01275
.STOP	23246	23247	01276	01275
SEARCH				
TAU	23250	23413	01276	01277
PZT	23414	23556	01300	01277
CICCO	23557	23[illegible]	01300	01277
SCHT	23[illegible]	24051	01300	01312

C.5

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
PHEL	24340	24367	01311	01311
FLIB	24370	24472	01314	01314
..FCM	24473	24505	01315	01315
CHELY	24506	24604	01316	01315
MANT	24605	24620	01316	01315
.ENTP	24621	24710	01316	01315
.PACK	24711	25024	01316	01315
..PLC	25025	25041	01316	01315
IAND	25042	25051	01316	01315
.ZPLR	25052	25112	01316	01316

[illegible] ENT
.DST

.FAU.	25113	25162	01317	01316

-
DISPLAY UNDEFS,
NO UNDEFS
-
[illegible]

STARTING ALLOCATION [illegible]
[illegible]

[illegible] BLOCKS
0

FHL USED BLOCK[illegible]

C.5

LIST OF REFERENCES

1. Paige, G.C., Measurement of Case Wall Pressure Signatures in a Transonic Compressor using Real-Time Digital Instrumentation, Naval Postgraduate School M.S. Thesis, June 1976.

2. Hewlett-Packard Driver Manual, Published by Hewlett-Packard, Palo Alto Division, November 1969.

3. Hewlett-Packard Operating and Service Manual, High Speed Data Acquisition Subsystem 2311A, HP02311-90001, March 1970.

4. Hewlett-Packard Programming and Operating Manual, Real-Time Executive Basic Software System, HP29102-93001, March 1974.

5. Hewlett-Packard Operators Manual, HP21MX Computer Series, HP02108-90004, March 1975.

6. Signetics Catalog, Signetics Corporation, copyright 1972.

7. Hewlett-Packard Reference Manual, HP21MX Computer Series HP02108-90002, December 1974.

8. Fairchild Semiconductor, TTL Data Handbook, June 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Assoc. Professor R. P. Shreeve, Code 67Sf Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Lt. J. C. West, USN 6229 Hanley Corpus Christi, Texas 78412	1
6.	Mr. J. E. Hammer, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
7.	Turbo-Propulsion Laboratory, Code 67 Naval Postgraduate School Monterey, California 93940	8
8.	Dr. H. J. Mueller Naval Air Systems Command, Code 310A Navy Department Washington, D.C. 20361	1
9.	Mr. Karl H. Guttman Naval Air Systems Command, Code 330C Navy Department Washington, D.C. 20361	1
10.	Mr. Eric Lister R & T Division Naval Air Propulsion Test Center Trenton, New Jersey 08628	1