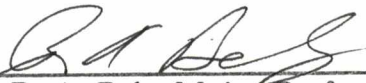


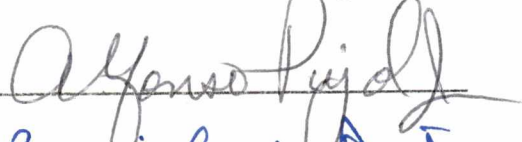
To the Graduate Council:

I am submitting herewith a thesis written by Sanjay Shreeniwas Date entitled ' Microprocessor Based Turbine Engine Image Controller '. I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science and Mechanics.



R. A. Belz, Major Professor

We have read this thesis
and recommend its acceptance:




Basil V. Ant

Accepted for the Council:



Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under the rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be given by my major professor when the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Sunny S. Walsh

Date JUNE 1988

**MICROPROCESSOR BASED TURBINE
ENGINE IMAGE CONTROLLER**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Sanjay Shreeniwas Date

June 1988

ACKNOWLEDGMENT

The work described herein was supported and performed at Sverdrup Technology at the Arnold Engineering Development Center.

The author wishes to express his gratitude and appreciation to the many people who have offered assistance during the course of the thesis. The author sincerely thanks his major professor, Dr. R. A. Belz for his invaluable guidance and help during the work of the thesis. He also thanks the other members of his committee, Dr. Alfonso Pujol and Dr. Basil Antar for their guidance and suggestions. He also acknowledges all the industry technicians who helped him in the project.

Very special thanks are extended to author's spouse, Jaya Date, for her dedicated support, editorial efforts and excellent keying of the text of thesis.

The author is grateful to the Dean of UTSI for giving him the opportunity to study here. Thanks are also due to the computer center for the use of the computer.

ABSTRACT

In this thesis a microprocessor based turbine engine image controller is presented. The controller uses a Z-80 CPU based microprocessor development system as a command module. The circuit developed to control the image is driven by this module. The software is written in Z-80 assembly language. Being a microprocessor driven circuit, it allows the user to gain full control over the image produced on the monitor. A detailed discussion of the project has been done in the first four chapters.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
II. DESIGN PHILOSOPHY	5
2.1. Modes	7
2.2. Input-Output Hardware	8
III. CIRCUIT OPERATION	13
3.1. The Strobe Generator Circuit	16
3.1.1. Mode A	16
3.1.2. Modes B,C,D	18
3.2. Rpm Measurement Circuit	19
IV. SOFTWARE	22
4.1. Initialization	25
4.2. Rpm Display	27
4.3. Status Switch Update	30
4.4. Mode Routines	30
4.4.1. Mode A	33
4.4.2. Mode B	35
4.4.3. Mode C	37
4.4.4. Mode D	39

V. RESULTS	42
VI. SUMMARY, CONCLUSION AND RECOMMENDATIONS	55
6.1. Summary	55
6.2. Conclusion	55
6.3. Recommendations	56
LIST OF REFERENCES	57
APPENDIX	59
VITA	72

LIST OF TABLES

TABLE	PAGE
1. Control Words for Different Modes	17
2. Initialization Words for PIO and CTC	17
3. Subroutine Memory Map	24
4. Binary Interpretation of Status Switches	32
5. Experimental Data-I	44
6. Experimental Data-II	45

LIST OF FIGURES

FIGURE	PAGE
1. Turbine Engine Inlet Viewing System	2
2. General System Layout	6
3. Block Diagram of CTC and PIO	10
4. Bit Designation of Port A and Port B	11
5. General Block Diagram of MDS	14
6. Strobe Generator Circuit	15
7. Rpm Measurement Circuit	20
8. Overall Software Flowchart	23
9. Flowchart For Initialization	26
10. Flowchart For Rpm	28
11. Flowchart for Status Update	31
12. Flowchart For Mode A	34
13. Flowchart For Mode B	36
14. Flowchart For Mode C	38
15. Flowchart For Mode D	40
16. Mode A Test Results	46
17. Mode B Test Results	47

18. Mode C Test Results	50
19. Mode D Test Results	53

CHAPTER I

INTRODUCTION

A turbine engine must pass a series of tests under simulated altitude conditions at the Arnold Engineering Development Center, AEDC, before it is considered to be qualified for flight by the U. S. Air Force. One of these tests consists of simulating conditions experienced by the aircraft when it passes through a cloud of supercooled water droplets. These droplets solidify as they hit the inlet components of the engine. This accumulation of ice on the spinner, fan blades and stator blades results in changes in the inlet air flow conditions and this, in turn, can affect the performance of the engine.

Turbine engines are tested under simulated icing conditions by spraying water from a bank of spray nozzles located in the plenum chamber into a supercooled air stream (normally at temperatures ranging from -4 to 23° F). To improve the understanding of ice accretion and ingestion on engine performance, a viewing system was developed which serves as a monitor of the inlet icing conditions Ref. (1). This system consists of a strobed light source synchronized to engine rotation and a low light level video camera for viewing the engine inlet components. Figure 1 shows those components. The light strobe was triggered on one of these pulses each revolution. In addition the light source was also strobed by an external signal generator set to the rotation frequency of the engine; since the engine generates a fixed number of pulses per revolution called the engine pulses. A slight misadjustment of the signal generating frequency from the actual frequency of the engine pulses caused the image of the turbine blades generated by a video camera to rotate

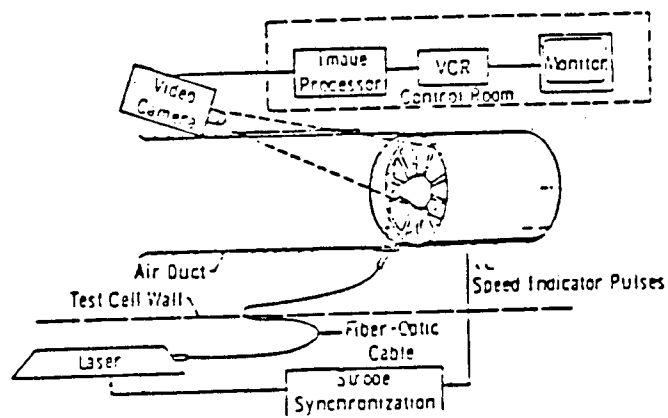


Figure 1. Turbine Engine Inlet Viewing System.

through the field of view. Likewise, by setting the frequency different from the engine pulses, the blades could be made to blur out and the objects behind them became visible.

Even though this system was effective for viewing ice build up on the fan blades of the engine, it had the following disadvantages:

1. The light source could only be strobed on an engine pulse and not between two engine pulses;
2. Preselection of the engine pulse which strobed the light source was not possible;
3. Manual adjustment of the strobing frequency could not be easily controlled;
4. Changes in engine speed caused the image to rotate.

These drawbacks led to the concept of controlling the strobing of the light source with a microprocessor. The strobe control system was designed around a Z-80 based microprocessor development system (MDS) Ref. (2). With the help of assembly language programs written for the MDS, the hardware enabled the user to vary the strobing of the light source in a variety of ways. The system was developed using standard TTL components.

Chapter II discusses design philosophy. It gives a general outline of the project. It also presents a brief discussion of the circuit architecture.

In chapter III, the details of the hardware are presented. The subsections of this chapter discuss the individual circuits such as the strobe generator circuit and the rpm measuring circuit.

Chapter IV describes the software written for the MDS to drive the various hardware circuits. Various subroutines have been categorized in respective subsections.

Chapter V gives the results obtained from the tests.

Chapter VI presents the summary, conclusion and recommendations for future developments.

CHAPTER II

DESIGN PHILOSOPHY

This chapter presents an introduction to the various sections of the project. The different modes of operation of the light strobing circuit were primarily designed to support the requirements of the user in viewing the ice build up on the turbine engine fan blades. The Input/Output hardware section explains all the necessary equipment and software needed for communication between the MDS and the hardware.

The basic approach for ice visualization was based on the concept of stopping rotating blade motion by strobing a light source in synchronization with the turbine engine speed thereby generating a stationary image on a video monitor. Figure 2 illustrates this approach. The approach taken in this effort was to process engine and clock pulses using the MDS software and specially designed digital hardware which contains counters, timers and logic circuitry. The MDS software also determines the appropriate mode of operation via a thumbwheel switch (TWS) and toggle and rotary switches. The engine pulses are collected and analyzed by the MDS and stored back into a counter. When the preselected engine pulse is sensed, the strobe is fired. During operation, the user can decide when the strobe is to be fired by setting the input switches accordingly.

The system was developed so that four operating modes could be selected by the user. These modes essentially allow the user to change the time of the strobe in order to effectively view the rotating component. The following sections briefly describe these modes and their operation.

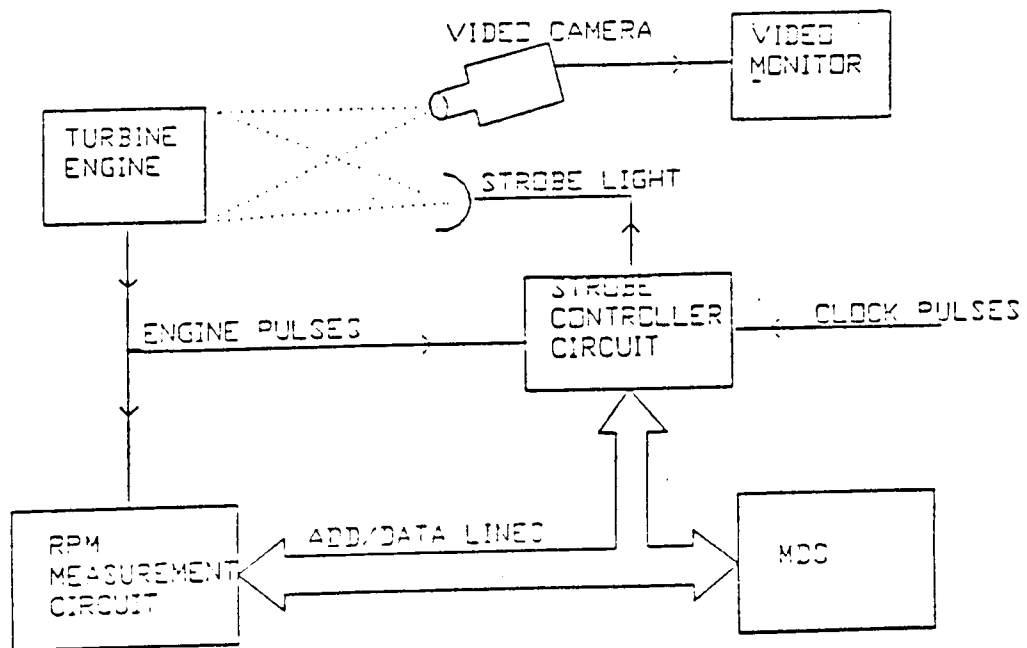


Figure 2. General System Layout

2.1 Modes

The first of the four modes identified for generating the strobe is mode A. This mode was designed to fire the strobe on any engine pulse during a revolution. If the engine generates n pulses per revolution, then the strobe can be fired on every pulse, during each revolution by the user by changing the TWS setting.

Sometimes a situation may arise where a clear view of the rotor blade under observation is not possible by strobing the light source on an engine pulse and increasing or decreasing the pulse number does not position the blade into the field of view. Therefore a second mode (mode B) was designed to fire the strobe in between two consecutive engine pulses. This mode gives better time resolution of the strobe signal than that produced by mode A. Implementation of mode B was based on the assumption that it would be selected after mode A and therefore the strobe would be fired sometime following the engine pulse selected in mode A. This engine pulse selected in mode A is called the target pulse. The strobe can be fired between the target pulse and the following engine pulse. In addition the time between the target pulse and firing of the strobe can be increased or decreased by means of thumbwheel switches on the input panel. The MDS updates the time delay of the strobe pulse on every engine rotation when the push button is released and pushed again. On zero time delay the strobe is fired on the target pulse; hence the MDS functions as if it were in mode A i.e. the strobe is fired on the engine pulse.

For the case in which the user wants the image to rotate, mode C is selected. This allows all the blades to be continuously viewed as they rotate through the field of view. In addition if the actual engine speed fluctuates, mode C will compensate for these engine speed fluctuations in order to keep the image rotating at a constant rate. In mode B changing engine speed will be manifested in the viewed image as a

rotation. However in mode D the image rotation speed will change when the engine speed changes. This mode (and mode B) allow engine speed changes to be observed in the image.

Setting up of the strobe pulse time is done from the input switches. These switches consist of a rotary switch which indicates the mode of operation. The thumbwheel switch indicates the number of engine or clock pulses by which the timing of the strobe is incremented or decremented. This results in delaying or speeding up the time between the target pulse and the strobe by causing the time of firing of the strobe to change on every revolution. A rocker switch is used to enter the new value set by the TWS into the MDS. Besides entering the new value the rocker switch is also used to make the image rotate. This function is activated only during modes C and D. In modes C and D, when the rocker switch is held ON, it causes the strobe to be continuously changed so that the image on the monitor rotates. The direction of rotation is determined by the toggle switch. If the toggle switch is held towards increment, the image rotates clockwise and when held towards decrement, it rotates counterclockwise.

2.2 Input-Output Hardware

The MDS was linked with the specially designed counting and strobe generating hardware through two eight-bit I/O ports. Engine and high frequency clock pulse counting was carried out by a counter timer chip which was coupled to the MDS through the address and data bus. The output of this chip was fed to specific digital logic gates. To accomplish the input, output and the counting functions, two Z-80 peripheral devices were used. They are the programmable counter timer chip Z-80-CTC (CTC) and the programmable input/output chip Z-80-PIO (PIO). The block

diagrams of these chips are illustrated in Figure 3.

The CTCs were used to count engine and clock pulses. The pulse counts were then used to calculate various parameters such as the engine rpms, the engine pulse reference for modes C and D and the time delay of the strobe after the target pulse has occurred. This chip has four independent channels which can be programmed as timers or counters. Each channel has a separate input/output line for clock pulse counting on trigger input and output line to indicate a zero count condition or a time off. Each channel can be monitored independently. Since all the channels were used to count the engine or clock pulses, all the channels were programmed as counters and not as timers.

The PIO has two eight bit I/O ports which can be programmed for byte input, byte output, bidirectional data transfer or bit I/O. Only one PIO was required for all of the I/O operations. Port A of this chip was designated as the user input port. The status of the panel switches was monitored through this port by the MDS. Port B was designated as the system control port. Through this port the MDS communicates with the hardware, hence this port was set up for bit I/O where three of the eight bits were for input and five were output bits. Figure 4 indicates the designated bit directions and bit operation for ports A and B. Bits 2, 5 and 7 of port B were designated as input bits. Bit 2 is set when channel A0 of CTC reaches zero count and it is called the 'count complete' indicator bit. Bit 5 is connected to the RPM counter circuit. This bit is set by a 10 Hz clock which is used for the RPM calculations. Output bits 1, 3, 4 and 6 control the strobe and the speed counter circuits. Bit 3 opens and closes a logic gate which allows the engine pulses to enter channel A0 for modes B, C and D. Bit 1 sets a flipflop which allows the engine pulses to be counted by channel B0.

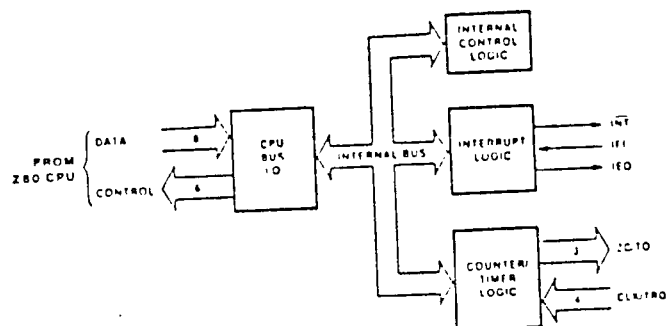
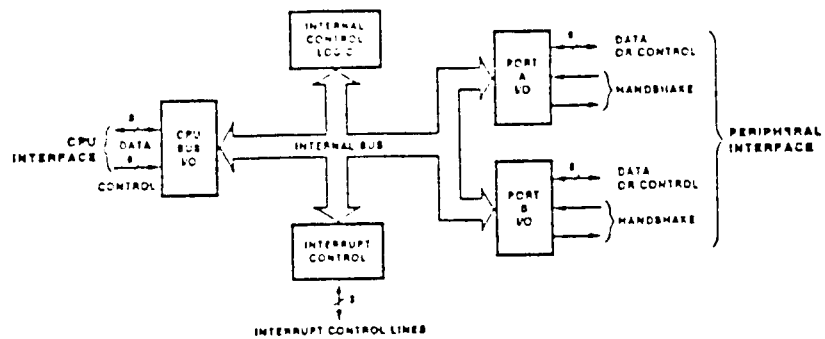


Figure 3. Block Diagram of CTC and PIO

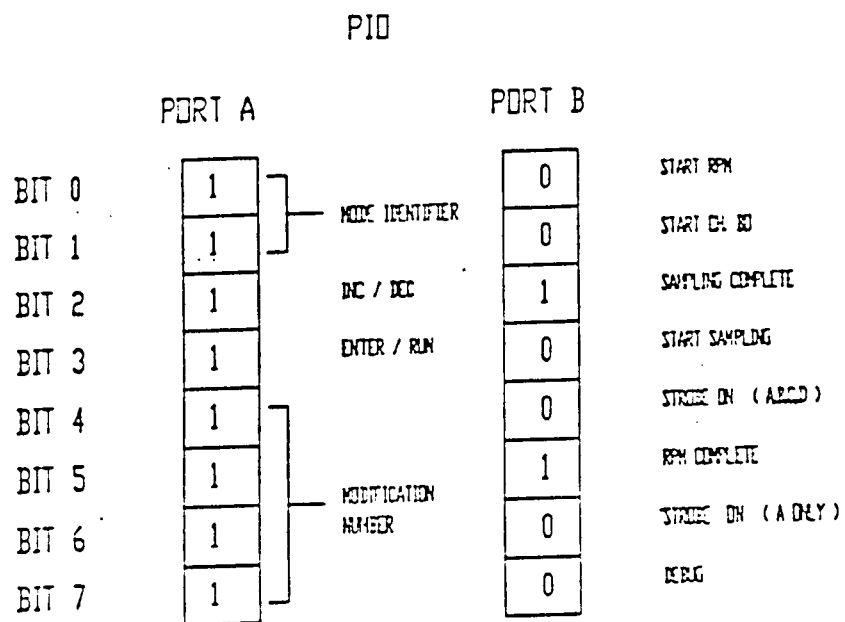


Figure 4. Bit Designation of Port A and Port B

This bit is used to start the counting process and is reset after every revolution. Bit 4 controls the strobe firing gate. The zero count output of CTC channel A2 is the strobe pulse. When bit 4 is set the output pulse from the channel passes through a gate and generates the strobe pulse. Bit 6 controls the strobe for mode A by means of a gate which allows the channel output to trigger the strobe line.

Both of these chips are Z-80 CPU peripheral chips and are synchronized to the Z-80 clock. They do not require any buffers for interfacing. They are TTL compatible and therefore the setup and propagation delay times also fall in the TTL range.

Strobing was performed under interrupt control. On an interrupt the CPU exits the main program which calculates and displays the engine RPM and services the interrupting device .

The Z-80 uses three different kinds of interrupts . They are Bus request, general purpose and vectored interrupts and are called IM 0, IM 1 and IM 2 respectively. In this project only the last two interrupts were used. A vectored interrupt IM 2 is generated when one engine revolution finishes and a new one begins. Generation of the vectored interrupt is done by CTC channel B3 which contains the total number of pulses per revolution. Engine pulses are input to the CLK/TRG input of this CTC channel. When the channel downcounts to zero, the vectored interrupt is generated and the CPU services the interrupt by jumping to the selected mode handling routine.

The general purpose interrupt is generated when the hardware fires the strobe. The purpose of this interrupt is to indicate to the CPU that the strobe generation process is now complete. The CPU, in servicing this interrupt , resets the strobe generation circuit to an inactive stage and reads the input status switches.

CHAPTER III

CIRCUIT OPERATION

In this chapter detailed circuit operation of the hardware is explained. The hardware is composed of two circuits; the strobe generator circuit and the rpm measurement circuit. The first section explains circuit behaviour for strobe generation in the four different modes. The second section explains the rpm measurement circuitry.

The hardware consisted of two boards, 1) the microprocessor board and 2) the strobe generator and speed counter board. The microprocessor which is a commercially available unit, Ref. (3) was used as a controller for the second board which consisted of a CTC, the PIO and TTL gates. Figure 5 illustrates the general block diagram of the microprocessor, the strobe generator circuit and the speed counter circuit. Details of the microprocessor can be found in the schematics of Ref. (3). In general however, it contains a random access memory of 2 K. It also has an on-board PIO and CTC and has a keyboard which was used for programming. This keyboard was used for the input of the number for programming. This keyboard was used for. A six digit LED display was used to display the engine rpm calculations.

3.1 The Strobe Generator Circuit

Figure 6 shows the circuit diagram of the strobe generator circuit that was designed. The entire circuit uses a total of 8 logic gates and 3 T-type flipflops. At reset the circuit is disconnected from the incoming engine pulses and from the 180 KHz high frequency clock pulses.

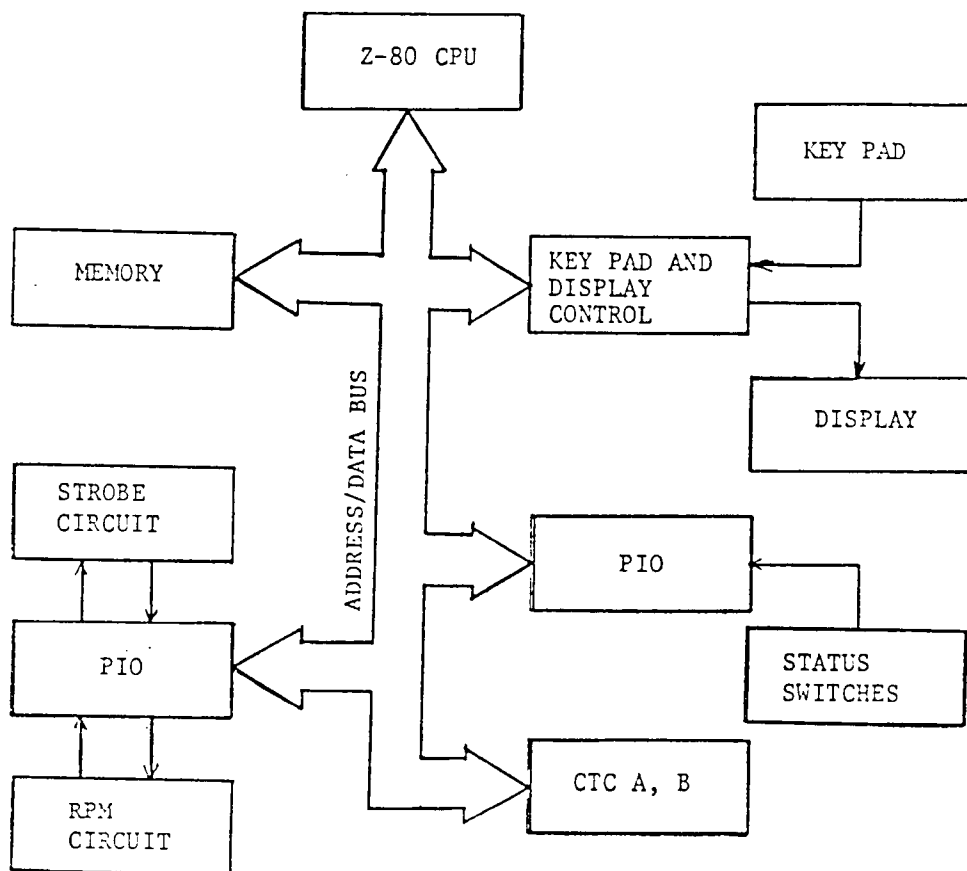


Figure 5. General Block Diagram of MDS

For simplicity this is called the Stop All Action (SAA) stage of the circuit. SAA can be changed only by having the CPU write an eight bit word into port B of the PIO. The specific eight bit hex words for each mode are given in Table 1 and the initialization hex words for PIO and CTC are presented in Table 2. In Figure 6, CTC channels A0 and B0 count the engine pulses and CTC channels A1 and A2 count the high frequency clock pulses. The starting point of counting is set by the software.

3.1.1 Mode A

In this mode flipflops F1 and F3 and gates 2 and 5 are used. Because, in this mode, the strobe pulse is generated on the engine pulse itself, only one CTC channel (B0) is required. In order to get the circuit in mode A, an eight bit word (41) is written to port B of the PIO which resets flipflop F2. This results in disabling channels A1 and A2 and enabling flipflop F3 and gate 5. Channel B0 is loaded by the CPU with the engine pulse number on which a strobe is to be generated. CTC channel B3 which counts the number of pulses per revolution generates an interrupt when its downcount reaches zero. After the interrupt, port B (bit 1) is pulsed low by the CPU to set the Q output of F1 which in turn opens gate 2 and allows engine pulses to be counted down by CTC channel B0.

The CTC zero output is connected to the clock input of flipflop F3 and drives the strobe line high. This flipflop is initialized by port B, bit 4, and drives the strobe line high when channel B0 goes to zero. The Q output of this flipflop is connected to the CPU interrupt line through gate 12. At interrupt 1 which is a general purpose interrupt the circuit is put into the SAA stage and is protected from stray pulses or spikes which may otherwise fire the strobe pulse.

Table 1. Control Words for Different Modes

	<u>Start Process</u>	<u>Reset (SAA)</u>	<u>Start Sampling</u>	<u>Start Strobe</u>
Mode A	41	03	-	41
Mode B	01	03	-	01
Modes C/D	01	03	08	11

Table 2. Initialization Words for PIO and CTC

	<u>Reset (SAA)</u>	<u>Initialization Word</u>
Ch B0	57	05
CH B1	57	FF
CH B2	57	FF
CH B3	57	N
CH A0	57	B0+1
CH A1	57	FF
CH A2	57	FF
Port A	03	FF
Port B	03	24

3.1.2 Modes B,C,D

The circuit in Figure 6 for modes B, C and D is more complicated than for mode A because the high frequency clock pulses are counted before firing the strobe. To understand how this circuit in Figure 6 supports these modes assume that the strobe is to be generated after the n^{th} engine pulse. The CPU first loads CTC channel B0 with a number that is two counts less than the source pulse and the engine pulse preceding the strobe pulse. This is required so that the number of high frequency clock pulses between two engine pulses can be measured prior to the generation of the strobe on the target engine pulse. Secondly, channel A0 is loaded with a number that is one count less than the target pulse.

The zero output of both of these channels is connected to the clock input of flipflop F2 through an OR gate (gate 4). F2 is cleared during circuit initialization through bit 3 of PIO port B. The output of this flipflop is connected to one input of an AND gate (gate 6). The second input of this gate is connected to the 180 KHz high frequency clock. The output of this gate is the input to a 16 bit counter formed by CTC channels A1 and A2. When interrupt 2 occurs channels A0 and B0 start to count on every incoming engine pulse which is passed to the channels by gates 1 and 2 by bringing bit 3 of port B high. On the $n-2$ pulse, B0 generates an output pulse which clocks flipflop F2 and opens gate 6. Channels A1 and A2 start counting down from FFFF on every 180 KHz clock pulse. On the $n-1$ engine pulse, channel A0 generates an output pulse which again clocks flipflop F2, closing gate 6 and completing the sampling operation. Since the next engine pulse is the target pulse; the number 01 is loaded into channel B0. The clock pulse count in channels A1 and A2 is now read and is processed. The resultant data are loaded back into channels A1 and A2. Up to this time bit 4 of port B is held low. This serves a dual

purpose. One, it holds flipflop F3 low and two, it holds the output of gate 7 low. This is done to ensure that no stray strobing occurs when the sampling operation is conducted. Bit 4 is brought high when channels A1 and A2 are loaded with the new data. On the next engine pulse channel B0 reaches zero and generates an output pulse which clocks flipflop F2 and again opens gate6. 180 KHz clock pulses are counted down by channels A1 and A2. In addition bit 3 of port B is brought low which closes gate 1 and stops channel A0 from accepting any further engine pulses. When channel A2 reaches zero, the strobe pulse is generated. The output of F7 is also connected to the CPU interrupt line, so that when the strobe is generated interrupt 1 is set, causing the CPU to jump back to the mode handling routine. The software then resets the flipflops to the SAA stage. The CPU then exits the interrupt routine and reads the status switches for the next set of instructions. As far as the hardware operation is concerned, the circuit operates in the same fashion for modes C and D, the difference being in the software handling of the samples collected in channels A1 and A2 during the sampling operation.

3.2 Rpm Measurement Circuit

The rpm measurement circuit shown in Figure 7 counts engine pulses for the calculation of engine speed. It consists of 4 logic gates and two T type flipflops. The circuit was designed to be fully dependent on the MDS. Bit 0 of port B is the circuit controller bit and bit 5 is the indicator bit.

To start the measurement process bit zero of port B is pulsed by the CPU. This clears F4 and sets F5.

As a result gate 8 and 9 are closed and gate 14 is opened. On the rising edge of a 10 Hz clock, both flipflops are clocked. This sets Q of F4 high and Q of F5 low. The Q of F4 opens gates 8 and 10 and Q of 9 closes gate 9. With gate 8 open, the engine pulses propagate through to channels B1 and B2 which start counting down. On the next rising edge of the 10 Hz clock the Q output of F4 is clocked low. This closes gate 8 and prevents engine pulses from being counted by channels B1 and B2. At the same time gate 10 prevents further engine pulses from toggling of F4. Bit 5 of port B is connected to Q of F4 and is monitored by the MDS. When this bit goes low after it has gone high, it indicates that the counting of engine pulses is complete and the channels are ready to be read. The rpm measurement is repeated by the CPU pulsing bit 0 again.

CHAPTER IV

SOFTWARE

In this chapter all software routines which control and communicate with the hardware are explained. The mode subroutines for the different modes will also be individually explained. Figure 8 illustrates the overall software flowchart for the strobe controller. The diagram starts with the initialization program where all the PIO ports and CTC channels are initialized and the number of engine pulses generated per revolution are input.

Interrupt mode 2 is enabled as soon as the initialization process is completed and the rpm calculation program begins execution. On the first interrupt (IM 2) which occurs when the engine pulse counting channel (B3) reaches zero (i.e. the revolution is completed) the CPU jumps out of the rpm calculation program and enters the mode A strobe generating routine. After executing the strobe subroutine, interrupt 2 is set and software branches back to the rpm calculation. A second interrupt (IM 1) is executed when the strobe pulse is generated. During the execution of the interrupt subroutine the CPU reads the status switches to determine the mode to be executed during the next revolution. After the update, the CPU once again jumps to the rpm program until IM 2 is generated. At IM 2 the CPU jumps to the preselected mode routine and starts executing it. After the execution is complete, the CPU once again jumps to the rpm program. The explanation of individual blocks is as follows. Table 3 illustrates the memory map of the subroutines and the starting addresses for each.

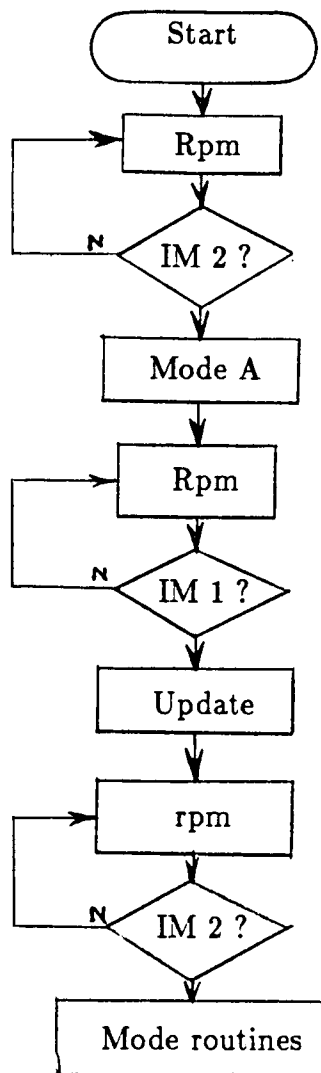


Figure 8. Overall Software Flowchart

Table 3. Subroutine Memory Map

ADDRESS	PROGRAM
180A H	Initialization
1840 H	RPM
1880 H	Mode A
18E0 H	Mode B
1900 H	Mode C
1A00 H	Mode D
1B00 H	Status Update
1D00 H	Multiplication
1D20 H	Division
1D40 H	Display
1E00 H	Vectored Interrupt
1FEE H	General Interrupt

4.1 Initialization

Figure 9 shows the flowchart of the initialization routine. During system initialization the number of engine pulses per revolution is requested for input by displaying a lower case ' N ' on the alphanumeric displays of the MDS. The two digit hex number corresponding to the number of engine pulses per revolution is input by the user from the MDS keypad. The maximum value of ' N ' which can be input is FF hex (256 decimal). and upon entry of the engine pulse number the ' GO ' key is pushed to indicate that the number should be read. Following the input the I/O ports of the PIO and the two CTC chips are initialized. The seven CTC channels which are used in this project are initialized in groups according to their tasks because the initialization procedure for the CTC channels counting the engine pulses is different than those counting the clock pulses. The number of pulses per revolution is loaded into channel B3 and the interrupt mode is activated. Therefore, after every revolution this channel generates an interrupt. CTC channels B1 and B2 were designated to count engine pulses for display of engine speed while CTC channels A0, A1, A2 and B0 were used for strobe generation as was explained in chapter 3.

In the initialization program mode A is initially selected. A default value 05 is loaded into channel B0. It generates the strobe pulse on the fifth engine pulse when the count equals zero. Appendix A gives the source listing of the initialization routine. As discussed earlier ports A and B of the PIO are designated as input and bit control respectively.

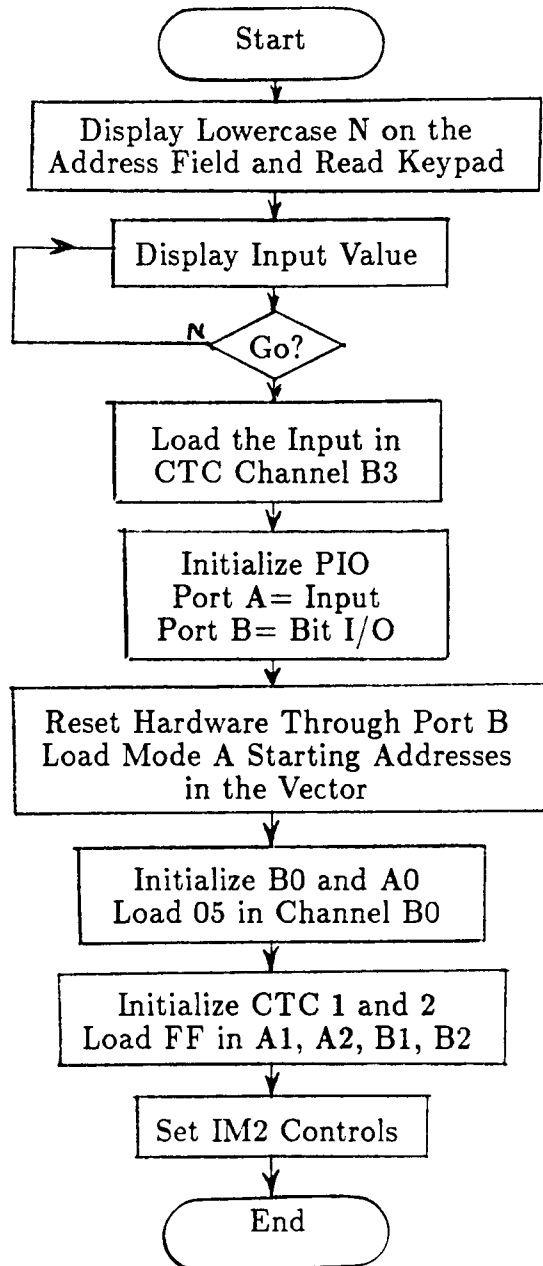


Figure 9. Flowchart for Initialization

4.2 Rpm Display

Figure 10 shows the flowchart of this program. The engine rpm is calculated by counting the number of engine pulses over a predetermined time period set by an external low frequency 10 Hz clock. The number of pulses per revolution (N) were input and stored in memory during the initialization. The number of pulses (P) counted in during one clock period (one tenth of a second) are stored in the CTC channels B1 and B2. Engine speed (Q) is calculated as follows.

$$Q = \frac{P \times 600}{N}$$

Eventhough P can be a sixteen bit number since the two engine pulse counting CTCs' form a sixteen bit register, there is a possibility of Q having a value greater than 65,536. Since P is first multiplied by 600 before dividing by N an overflow check was set up after the multiplication. If an overflow occurs then division is performed before multiplication. Multiplication is performed first because the higher the value of the multiplication the greater is the rpm calculation resolution. The calculated rpm is displayed on the MDS display. If the turbine engine revolves so slowly that even after dividing P by N, Q results in an overflow and then MDS display will flash indicating that the engine rpm are lower than the lower limit of the system.

This program repeats itself and if no strobe is fired, it continues displaying the engine speed at 10 Hz intervals. When an interrupt occurs the CPU stores the contents of the registers in the stack and jumps to the mode routine servicing the interrupt. After servicing the interrupt it comes back to the rpm program and resumes execution.

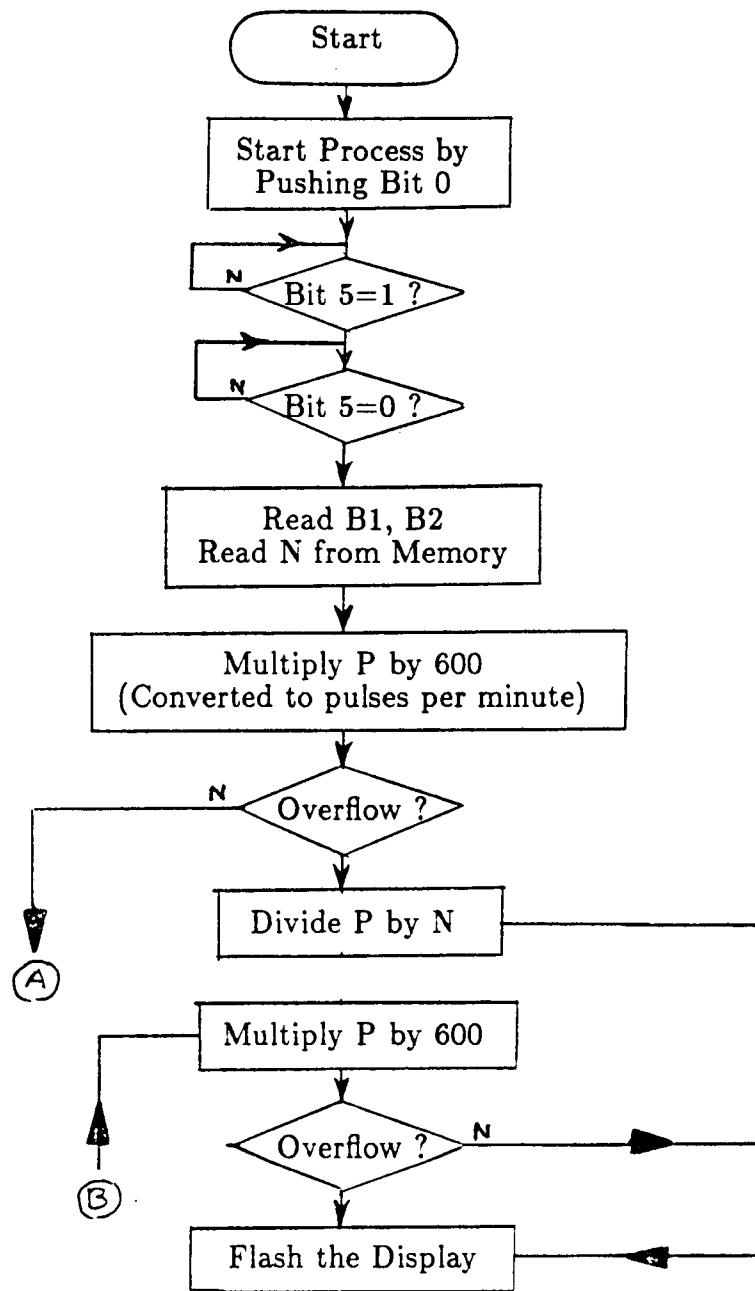


Figure 10. Flowchart for Rpm

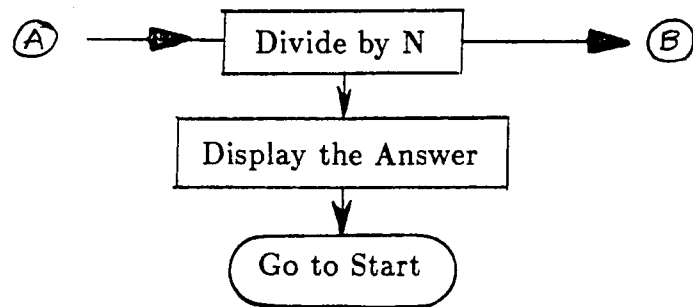


Figure 10. (cntd.)

4.3 Status Switch Update

This program is the IM 1 service routine. The flowchart of this routine is given in Figure 11. In this routine the decision as to which mode subroutine to execute during the next revolution is made based on the status switch inputs. The status switches are read through port A of the PIO. When the CPU reads this port a binary equivalent of the mode is read in the least significant two bits of the port bits 0 and 1. This provides the starting address of the specific mode routine to be executed on the next revolution. Table 4 shows the binary representation of the status switches.

Bit 3 of port A is now read to determine the position of the ENTER/RUN rocker switch. If the switch is not pressed then the status of the input switches is ignored and the previously selected mode routine and the value of the strobe which is generated was executed. If the rocker switch is pressed then the current mode, the thumbwheel switch value and the increment/decrement status are accepted. The value set by the TWS is either added or subtracted from the strobe number depending upon the position of the toggle switch. If the newly calculated value goes below zero or goes beyond the maximum value allowed for the specific mode, measures are taken to restrict it to an acceptable value. The acceptable region and the mode capacity differs from mode to mode. These parameters are discussed in the respective mode subsections. Software initiates interrupt mode 2 and exits this routine to resume the rpm calculation.

4.4 Mode Routines

Each of the modes has its own subroutine. The routine is executed when one revolution of the engine is completed. This subsequently produces an interrupt

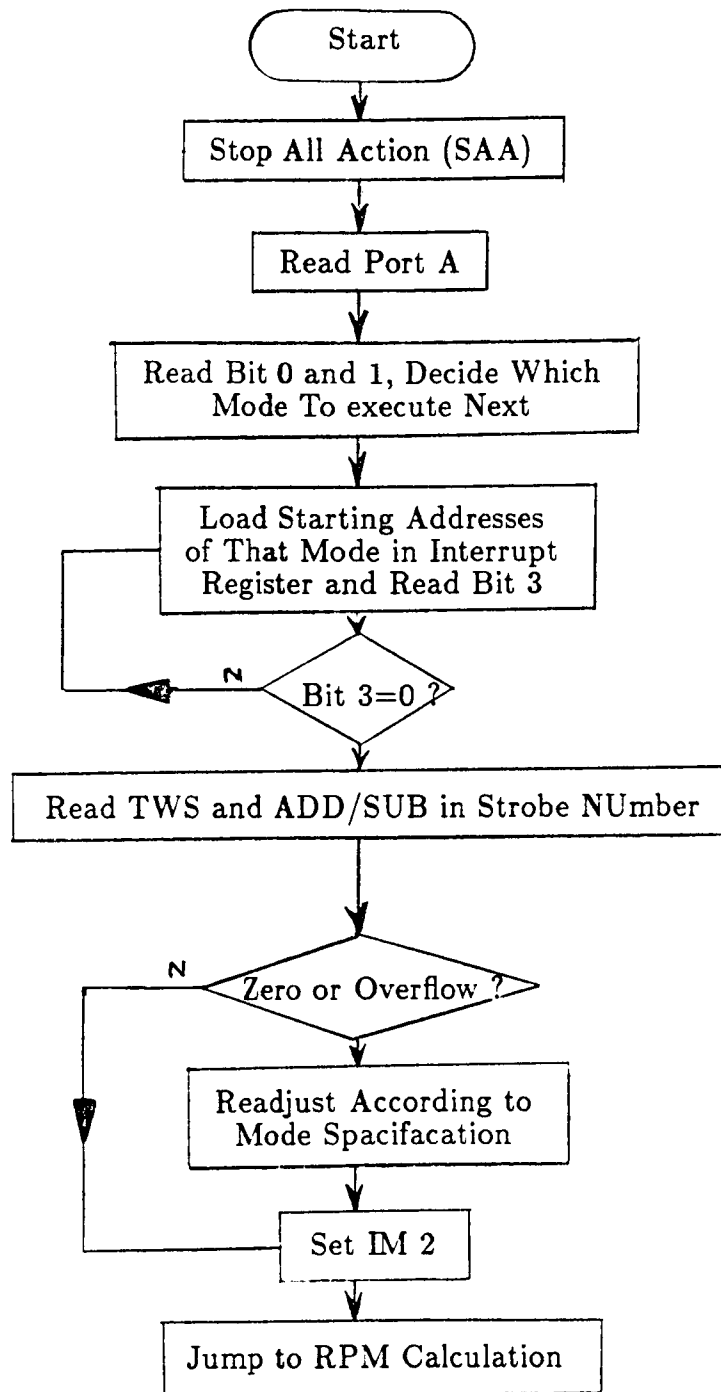


Figure 11. Flowchart for status update

Table 4. Binary Interpretation of Status Switches

Rotary Switch	Toggle Switch	Rocker Switch	TWS	REMARKS
00	0	1/0	x	Mode A subtract x from last value/Mode A use last value
01	0	1/0	x	Mode B subtract x from last value/Mode B use last value
10	0	1/0	x	Mode C decrease phase by x/Mode C synchronize
11	0	1/0	x	Mode D decrease phase by x/Mode D use last value
00	1	1/0	x	Mode A add x in last value/Mode A use last value
01	1	1/0	x	Mode B add x in last value/Mode B use last value
10	1	1/0	x	Mode C increase phase by x/Mode C synchronize
11	1	1/0	x	Mode D increase phase by x/Mode D use last value

(IM 2) that is received by the CPU. The specific starting addresses of the mode subroutines and their assembly listings are given in appendix B.

4.4.1 Mode A

Figure 12 shows the flowchart of mode A. As stated earlier this mode consumes the least CPU time of all. This mode is automatically selected following the initialization program. Since all other modes are based upon the results of this mode it is necessary that it be initiated first. In this mode the time resolution of the strobe is poor because it is fired only on an engine pulse.

Software operation in this mode is fairly simple. The pulse number which is the number of pulses per revolution is loaded into CTC channel B0. An eight bit word which sets the strobe generation circuit for mode A is written into port B of the PIO. CTC channel B3 contains the number of pulses per revolution and generates interrupt 2 when its count reaches zero. At this interrupt the program starts a count down of CTC channel B0 on each engine pulse and exits the routine. At the zero count of this channel the strobe pulse is generated. Interrupt 1 is also generated along with the strobe pulse. To service this interrupt the CPU jumps to a status switch update routine and sends an eight bit word to port B which stops the CTC channel and all other flipflops from getting clocked by further engine pulses. During operation of this mode the user may increase or decrease the pulse number on which the strobe is generated by inputting a new value into the TWS. If the pulse number on which the strobe is fired becomes higher than the total number of pulses per revolution then the program sets this number equal to one count less than the total number of pulses per revolution and if the pulse number on which

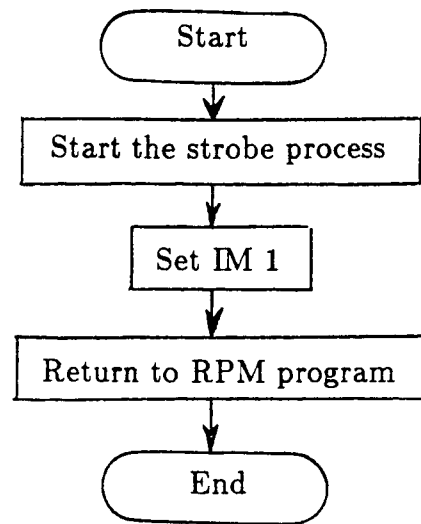


Figure 12. Flowchart for Mode A

the strobe is to be fired falls below or equal to zero, the program sets it back to 01 before loading it into the CTC channel.

4.4.2 Mode B

This is a modified version of mode A and is entered when the mode switch is set to the position. On interrupt 2 the software reads the binary output of the switch and subsequently jumps to the mode B routine. The flowchart of this routine is shown in Figure 13. As stated earlier a total of three CTC channels are used in this mode. Channel B0 is loaded with the target pulse count. The software writes the 'start' word in PIO port B which starts the strobe generation process. On the engine target pulse (i.e. when channel B0 reaches zero) CTC channels A1 and A2 start downcounting the 180 KHz clock. When these channels reach zero a strobe pulse and interrupt (IM 1) are generated. This interrupt causes the CPU to jump to the status switch update routine.

The software jumps to mode B handling routine within this status switch update routine. In the mode handling routine if the ENTER/RUN switch is pressed the TWS pulses are added or subtracted from the old strobe value depending upon the position of the toggle switch. The newly calculated number is loaded into channels A1 and A2 where the most significant bits are loaded into A2.

If the newly calculated number is less than or equal to zero, the number is automatically reset to 1 and then loaded in channels A1 and A2. If the number goes beyond 65536 (FFFF in hex) then the number is allowed to roll over to 01. Mode B doesn't deviate from the target pulse set by mode A and therefore the value in channel B0 is not updated.

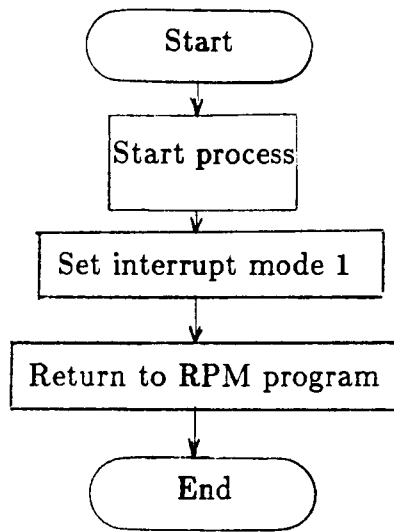


Figure 13. Flowchart for Mode B

After this calculation interrupt mode 2 is set and the interrupt register is loaded with the starting address of the mode B strobe generation routine. The software then jumps to the RPM calculation program and resumes the program where it had left off.

4.4.3 Mode C

In mode C the image can be rotated at a rate irrespective of the fluctuations in the engine speed. The flowchart of mode C is shown in Figure 14. The appropriate values for the engine pulses are loaded into the CTC channels B0 and A0. The number of 180 KHz clock pulses which occur between two consecutive engine pulses are read and placed in memory. The CPU now calculates the percentage of time between the engine pulses when the strobe was generated on the last rotation and when it should be generated on the current rotation. The calculation is done as follows.

The number, M, of clock pulses between two engine pulses on the previous rotation is stored into one particular register. A second register contains the pulse number, P, on which the strobe was fired. Now, if the number of pulses in the current revolution is Q, the clock pulse S on which the strobe should be fired, is calculated from the equation.

$$S = \frac{P \times Q}{M}$$

The image can be rotated by adding a fixed constant value to S. For this work an arbitrary value of ten was chosen to be added to the newly calculated S for every revolution.

The new value of S and Q are loaded back into memory for latter use. The program determines whether to increase the number in channels B0 and A0 by 1

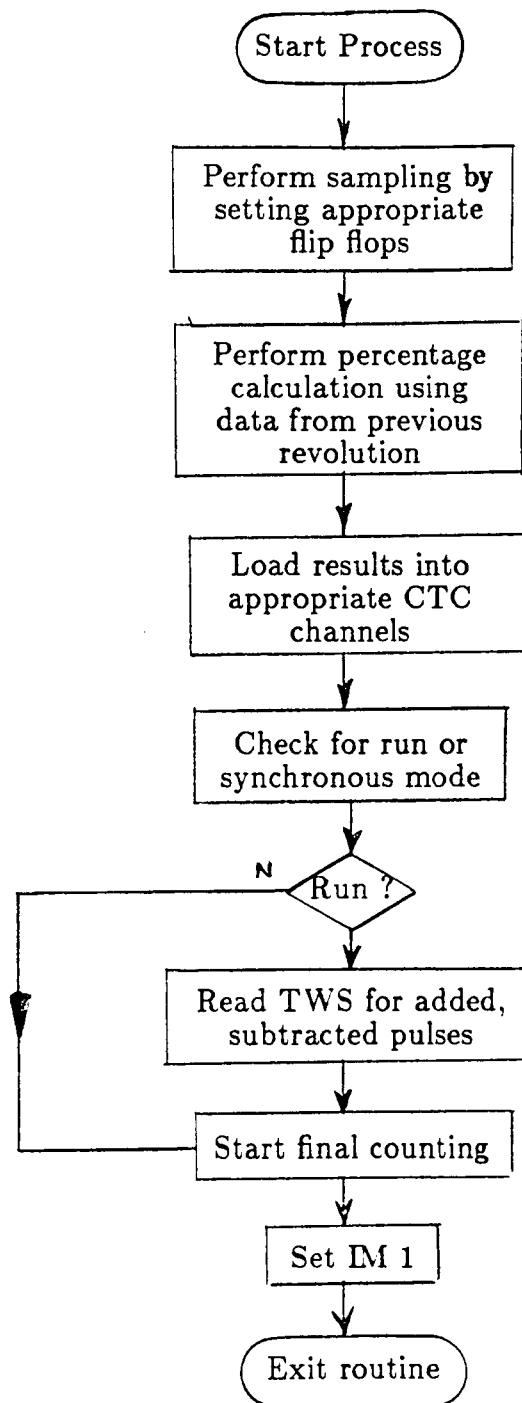


Figure 14. Flowchart for Mode C

by testing S against Q. If the pulse number, S, is greater than the high frequency pulses between two engine pulses, Q, then FFFF is loaded into channels A1 and A2. and the values of B0 and A0 are increased by 1 to advance the target pulse to the next engine pulse. When the value for B0 is equal to N (where N is the total number of pulses per revolution), then B0 and A0 are loaded with 1 and 2 respectively. This way the image continues to rotate with a constant rate after the target pulse has exceeded the number of engine pulses in one revolution. The program then activates interrupt 1 and returns to the main program. At the interrupt (when the strobe is fired) the CPU jumps back into the status switch service routine again and resets the circuit to a Stop All Action stage.

4.3.4 Mode D

The flowchart of this mode is shown in Figure 15. As with mode C the time of the strobe pulse can be changed on each blade rotation so that the image of the blade is made to rotate. In this mode the strobe pulse is generated on the same number of high frequency clock pulses and therefore if the engine rpm changes then the image rotation speed will also change.

Addition or subtraction of pulse count values is done based on the value of the thumbwheel switch settings and the status of the increase/decrease toggle switch. In this mode the speed of rotation of the image seen on the monitor is proportional to the change in the engine speed. The actual engine speed is displayed on the MDS display by the rpm calculation program. As in mode C this mode allows the user to pulse the strobe between any two engine pulses; whereas in mode B only one pair of pulses can be selected.

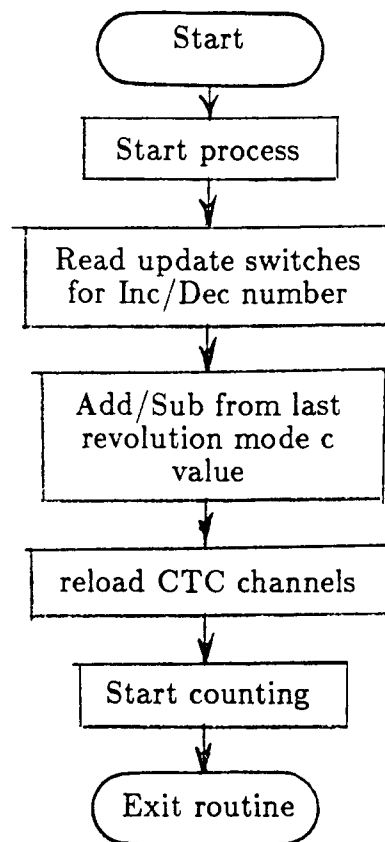


Figure 15. Flowchart for Mode D

In this mode the clock pulses are collected from channels A1 and A2 exactly the same way they are collected in mode C. Then they are added to, or subtracted from, the number specified by the user through the status switches. The result is again fed back in channels A1 and A2 and the countdown of the strobe starts with this number. The engine speed is not taken into account when the strobe is fired. Therefore if the engine speed changes, the rotation of the image on the monitor also changes

The advantage of this mode over mode C is that the engine speed changes can be observed on the monitor. After channels A1 and A2 are set to count down for the strobe, the software sets IM 1 and exits the routine. At the interrupt the CPU jumps to the status switch update routine and if the user has selected mode D, it jumps to the mode D handling routine. The operation of this routine is exactly like mode C except if the rocker switch is off the program ignores the TWS value and exits the program. If the switch is ON the TWS value is added or subtracted depending upon the toggle switch position. If the new count exceeds 65536, it is not allowed to roll over to 1; but instead is held at FFFF. The CPU then sets interrupt 2 and loads the starting address into an index register and exits the routine to resume rpm calculation.

CHAPTER V

RESULTS

This chapter presents the results obtained from a test simulating real conditons. The following equipment was used to simulate test conditions.

1) A Hewlett Packard signal generator model HP 3314A was used to simulate engine pulses.

2) A Hewlett Packard Plotter model HP 7047A was used to simulate the recorder/VCR.

3) A Nicolett storage digital oscilloscope model 3091 was used in place of the camera.

Proper operation of the hardware design was verified by using the signal generator and generation of the strobe pulse was monitored on a two channel oscilloscope.

The pulses obtained from the signal generator at various frequencies which simulated different engine speeds were input to the hardware as the engine pulses. The same signal generator pulses were input to channel A of the oscilloscope. Channel B of the scope was connected to the strobe line and the output of CTC channel B3 which counts down engine pulses was used as an external trigger.

After verification of system operation different tests were conducted to analyze the circuit performance. These were 1) analysis of the maximum rpm that can be handled by each mode. 2) Number of clock pulses accumulated in an engine pulse for different rpm. 3) Deviation of degrees per one clock pulse. and 4) Resolution of the image's angular position at different rpm.

The maximum engine speed at which the system can operate correctly in each

mode is mainly dependent on the hardware speed and the speed which the CPU executes the software. Table 5 lists the maximum rpm that could be experimentally handled by the system. Table 6 shows the number of degrees (rightmost column) that could be changed by incrementing or decrementing one clock pulse while strobing the light. The deviation of the number of degrees per clock pulse was considered to be good even at 34,000 rpm. Mode A was clearly faster than the other modes because of less circuitry involved. For modes B, C, D the total propagation delay of the hardware was calculated to be 95 ns. Hence at the worst case the hardware was able to handle an engine pulse frequency of 10.5 MHz. In terms of rpm, at 20 pulses per revolution the worst case rpm handled by the hardware were 31.5×10^6 rpm. A significant decrease in speeds was introduced by the software written for each mode.

Mode A was tested first. In Figure 16 the pulses shown on the top are the strobe pulses and the pulses shown on the bottom are the engine pulses. The testing was done at an equivalent engine speed of 10,000 rpm with 12 pulses per revolution and the strobe was fired on every twelfth engine pulse. Channel B0 was programmed to generate a strobe on second engine pulse.

The target pulse number was increased to 14 (which is larger than the total number, 12, of pulse per revolution) and the test was again run. The strobe was generated on the eleventh engine pulse. This shows that the MDS correctly adjusted the target pulse number to one less than the total pulses per revolution.

Figures 17(a) and 17(b) show the test results for mode B. Figure 17(a) shows the channel output pulse and a single engine pulse. This plot was taken to verify the operation of the clock pulse counting CTC channels A1 and A2.

Table 5. Experimental Data-I

MODES	NO. OF CYCLES	EXECUTION TIME*	MAXIMUM RPM**
A	50.4	90	33333.33
B	64.2	120	25535.25
C	168.0	300	10351.69
D	134.4	240	12561.85

*In μ sec., the hardware propagation delay included.
 **At 20 pulses per revolution

Table 6. Experimental Data-II

ENG. RPM	CLK/1 ENG PLS	DEG/CLK PLS
100.000	5357.000	.003
200.000	2678.000	.007
300.000	1785.000	.010
400.000	1339.000	.013
500.000	1071.000	.017
600.000	892.000	.020
700.000	765.000	.024
800.000	669.000	.027
900.000	595.000	.030
1000.000	535.000	.034
2000.000	267.000	.067
3000.000	178.000	.101
4000.000	133.000	.134
5000.000	107.000	.168
6000.000	89.000	.202
7000.000	76.000	.235
8000.000	66.000	.269
9000.000	59.000	.302
10000.000	53.000	.336
11000.000	48.000	.370
12000.000	44.000	.403
13000.000	41.000	.437
14000.000	38.000	.470
15000.000	35.000	.504
16000.000	33.000	.538
17000.000	31.000	.571
18000.000	29.000	.605
19000.000	28.000	.638
20000.000	26.000	.672
21000.000	25.000	.706
22000.000	24.000	.739
23000.000	23.000	.773
24000.000	22.000	.806
25000.000	21.000	.840
26000.000	20.000	.874
27000.000	19.000	.907
28000.000	19.000	.941
29000.000	18.000	.974
30000.000	17.000	1.008
31000.000	17.000	1.042
32000.000	16.000	1.075
33000.000	16.000	1.109
34000.000	15.000	1.142

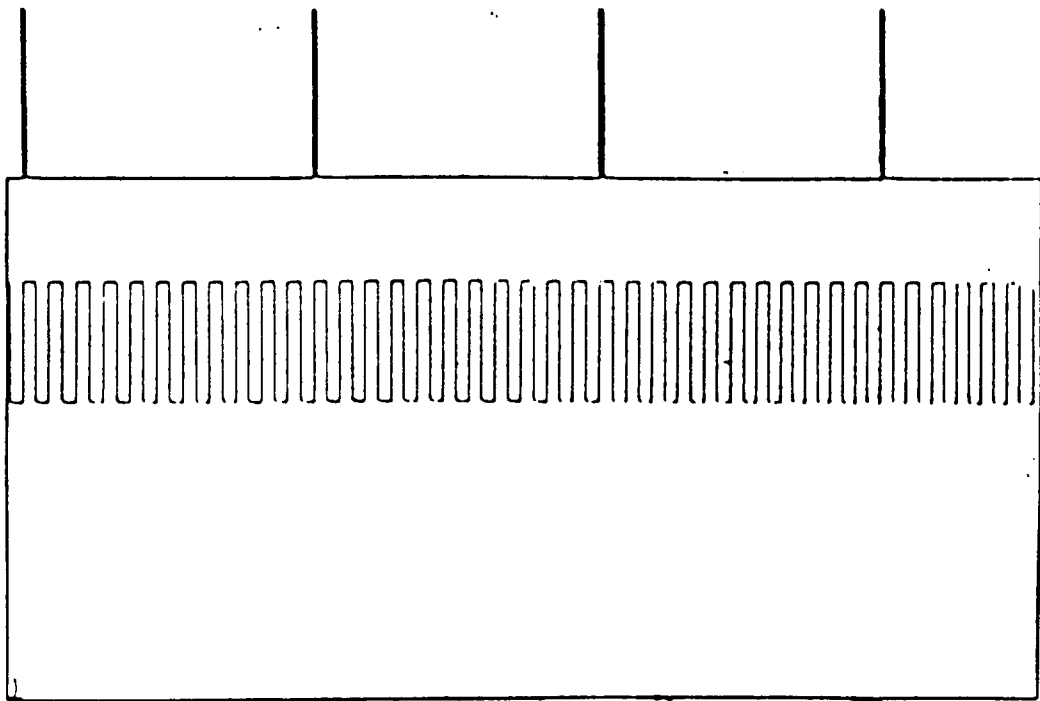


Figure 16. Mode A Test Results

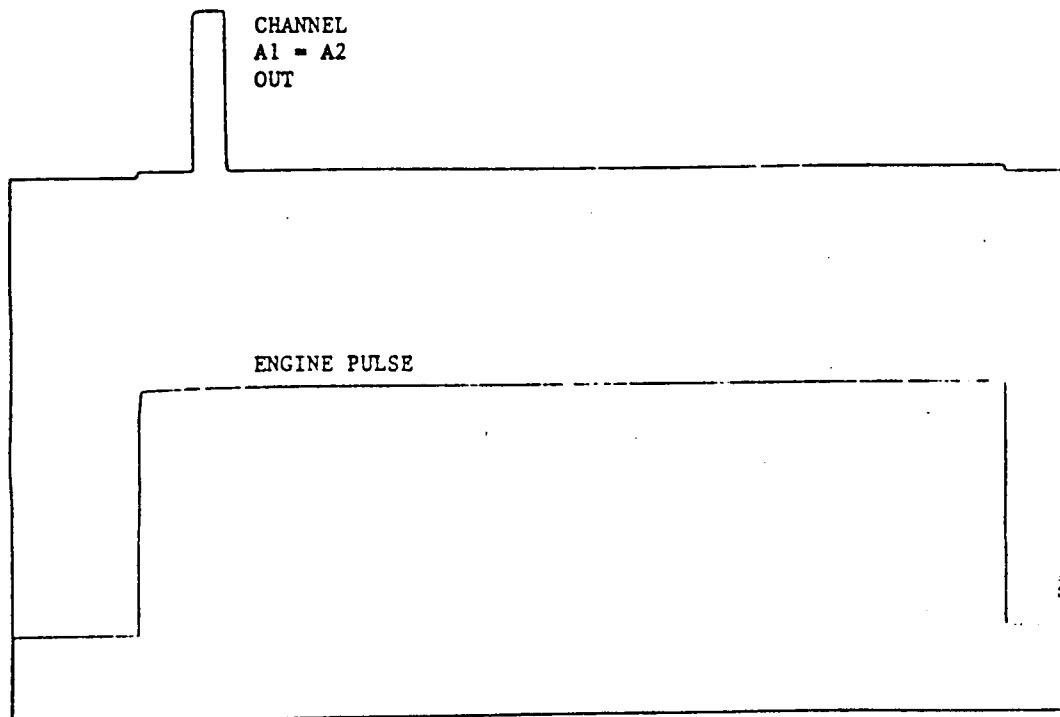


Figure 17. Mode B Test Results

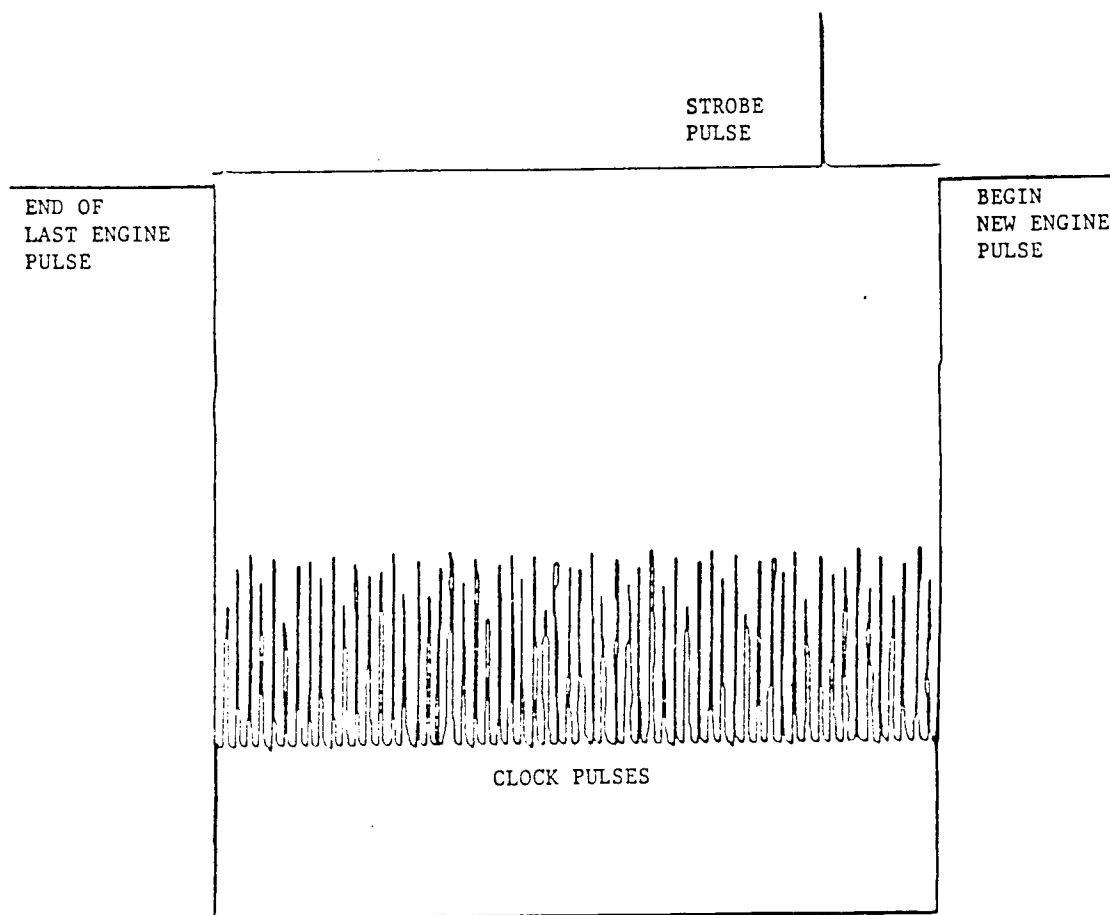


Figure 12. (contd.)

The circuit being a rising edge trigger, responded on the rising edge of the engine pulse and hence the plot is showing the channel output pulse arriving after the rising edge of the engine pulse. Figure 17(b) is actually a mixture of three plots. The topmost waveform is the strobe pulse. The darkened pulse is the engine pulse and the cluster of pulses inside the engine pulse consists of clock pulses. This plot indicates that there are 60 clock pulses between the target pulse and the next engine pulse and that the strobe is generated on the 50th clock pulse.

Mode C was first tested for synchronous operation. In this operation a percent value on which the strobe is to be fired is calculated. It is based on the strobe pulse number and the total number of clock pulses counted during the last revolution.

Figure 18(a) and 18(b) show the operation of mode C in synchronous mode. Figure 18(a) shows the strobe being fired on the 62nd pulse where the number of clock pulses between engine pulses were 141 (43.76 percent). The signal generator was then increased to simulate an increment in the engine speed and the test was rerun. The results obtained are shown in Figure 18(b). The strobe width was increased for this figure by keeping bit 3 high longer than a software delay routine in order to illustrate the ability to adjust the strobe width to meet speed constraints of slower circuitry. It shows that the strobe is being fired on the 35th clock pulse where the total number of pulses generated were only 80 (43.75 percent). This indicated the satisfactory operation for mode C in the synchronization mode.

Next the thumbwheel switch was set to 05 (decimal) and the rocker switch was pressed. The strobe pulse moved towards the right. This indicated that the number set by the thumbwheel switch was being added to the calculated strobe value.

CLOCK PULSES

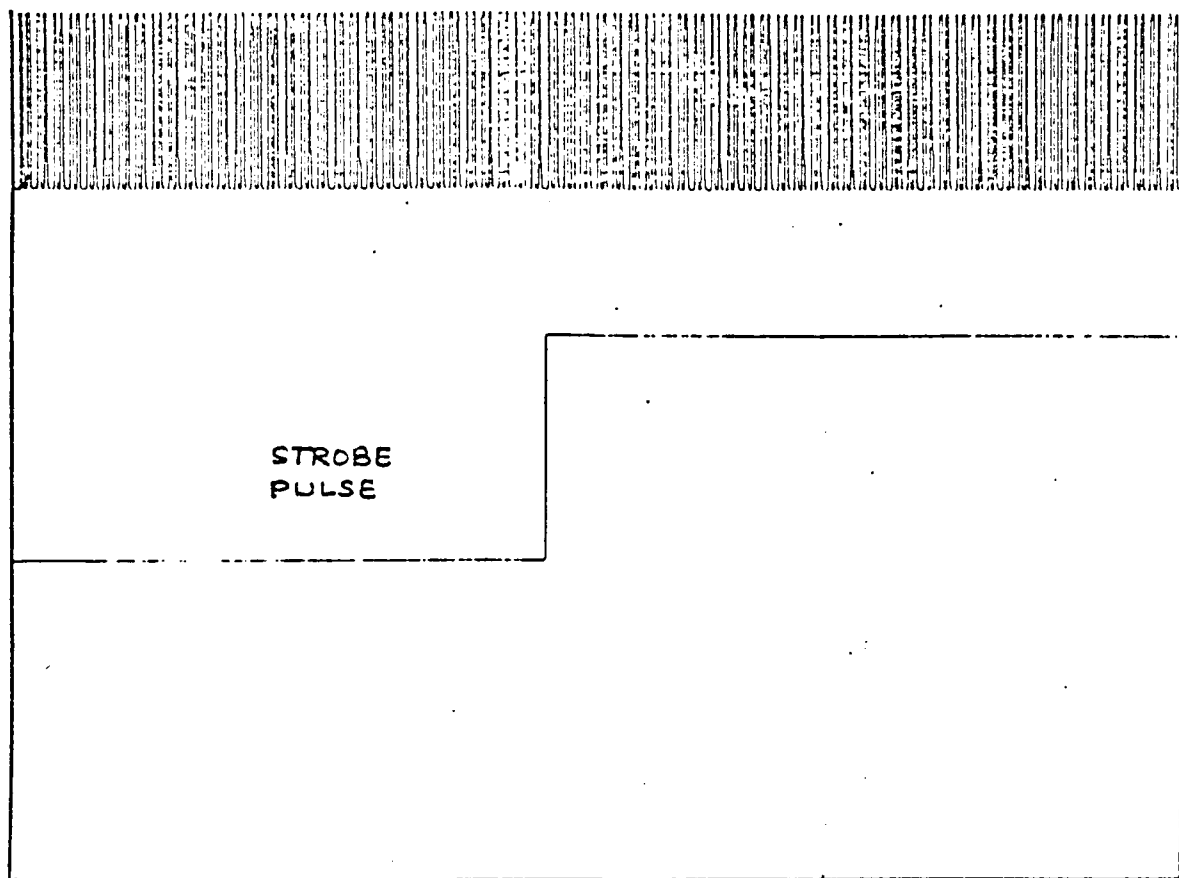


Figure 18 Mode C Test Results

CLOCK PULSES

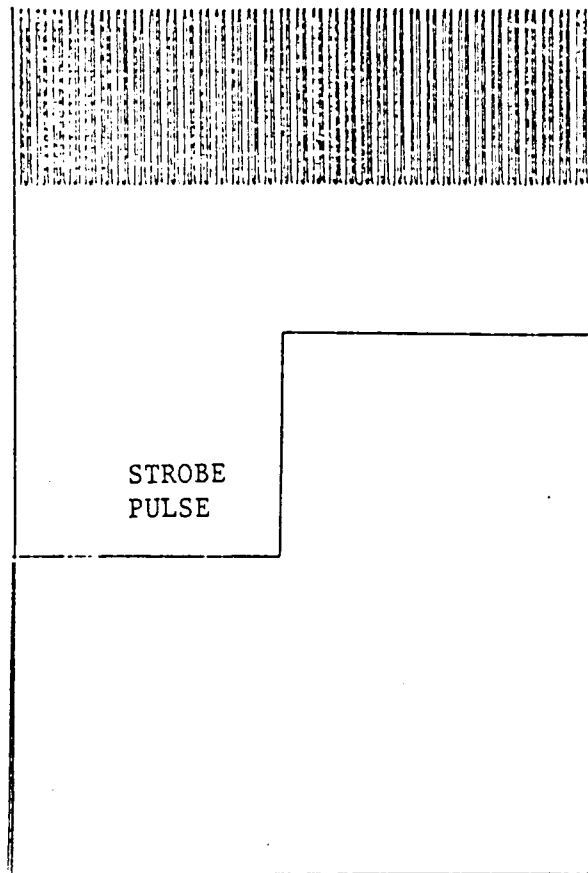


Figure 18. (contd.)

Figures 19(a) and 19(b) show the results obtained for mode D. Figure 19(a) shows the strobe was occurring on the 32nd clock pulse when the total number of clock pulses were 71. The data that was acquired after decreasing the signal generator value is shown in Figure 19(b). It shows that when the engine slowed down, the strobe was still generated on the 32nd clock pulse. Thus in mode D the percent value is not calculated and therefore the change in engine speed will directly reflect on the state of the image because the strobe number is totally independent of the speed of the engine. In addition to the data for angular deviation per clock pulse was compiled with 20 pulses per revolution and 360 degrees in one revolution. Each engine pulse occurs on 18 degrees of engine rotation. Table 5 illustrates the number of clock cycles per clock pulse at different rpm. IT starts at 10 rpm and goes upto 34, 000 rpm. The number of pulses collected in one engine pulse are from 53, 571 to 15. Even at 34, 000 rpm a deviation of 1 degree per pulse is possible. This gives a tremendous amount of control of the image rotation to the user.

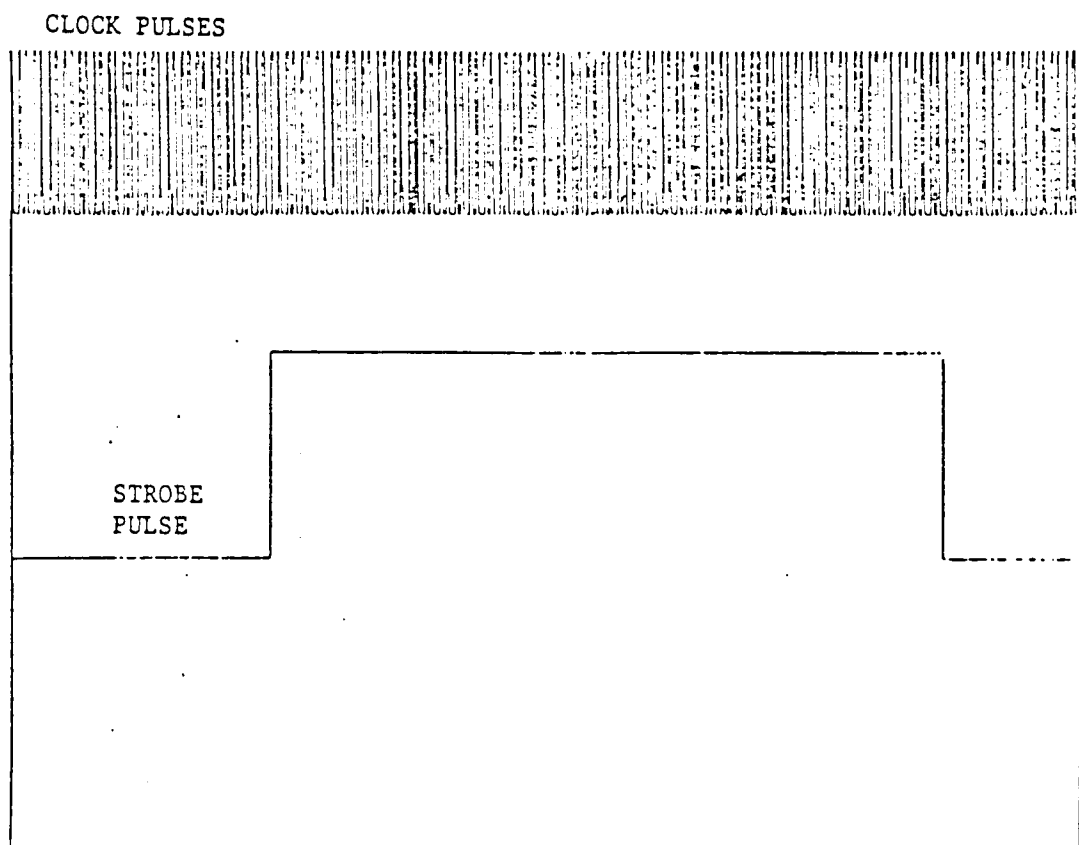


Figure 19 Mode D Test Results

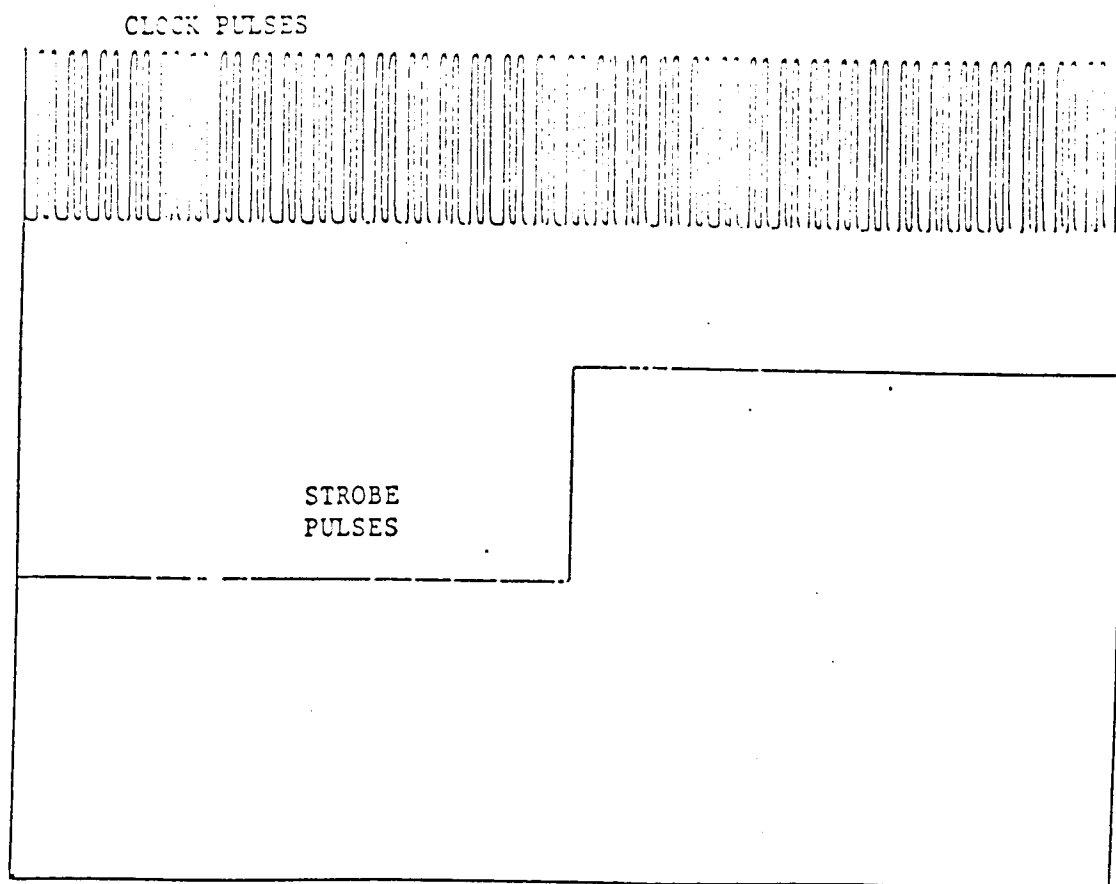


Figure 19. (Contd.)

CHAPTER VI

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

In order to view the rotor blades of a turbine engine the concept of strobe light visualization was developed. This concept lead to the fabrication of a microprocessor controlled hardware called the 'Turbine Engine Image Controller'. The strobe light was strobed in synchronization with turbine engine blade rotation and a video camera viewed the fan blades. The image of the turbine engine appeared to be stationary on the video monitor. Due to the varsitility of the microprocessor, strobing of the light source could be done on any selected engine pulse and also in between any two engine pulses. The strobe frequency controller was designed to support four different operating modes. Strobing 1) on the engine pulse, 2) in between two engine pulses, 3) for stationary image, 4) for image rotation with a preselected speed even if the turbine engine frequency fluctuateds and 5) for changing the image rotation with changing engine speed. The results obtained from the laboratory testings indicate the circuit operates correct for each of these modes.

6.2 Conclusion

The hardware and the software worked very efficiently to give a wide range of engine speeds that could be controlled. At 20 pulses per revolution, the maximum engine speed at which the strobe light could efficiently be strobed repetitively was 33,333.33 (in mode A). The two interrupts provided by the Z-80 CPU were utilized in such a way that one always occurred only after the other had completely

been serviced. The interrupt generating circuitry and the interrupt handling software routine made it possible to service the interrupts in a repeated manner. The hardware execution times of each mode were listed in table 5. A typical Z-80 response time was 1.2 to 2.8 microseconds. The CTC and PIO chips were quick in response. Their architecture made it very easy to read from or write into them. Both the chips being Z-80 peripherals, did not require any buffering or signal conditioning circuitry. Although the controller could effectively operate at 34,000 RPM in mode A, in mode C it could only operate to 10,000 rpm because of the large software programs. The MDS used in this project did not have any means of permanent memory storage. Hence the system had to be kept powered ON at all times while in the development stage. That resulted in excessive heating of the chips and extra careful handling of the board while plugging or unplugging new or damaged components. Also the MDS did not have an assembler built in the operating system. As a result the software had to be input in hexadecimal equivalent codes of assembly language.

6.3 Recommendations

This prototype strobe controller has a lot of potential for engine ice visualization. Its utility can be improved by implementing it on a dedicated system. This is best accomplished using a logic development system such as the Hewlett-Packard 64100. A coprocessor such as Advanced Micro Devices (AMD) 9512 would give 32 bit floating point results and would increase the resolution and speed considerably. Finally a full fledged practice test of the controller using a light source and a small fan is recommended to prove the operation of the system.

LIST OF REFERENCES

LIST OF REFERENCES

1. The Turbine Engine Inlet Viewing System, R. A. Belz Et al., AIAA/ ASME/ SAE/ ASEE 22nd Joint Propulsion Conference, June 16-18, 1986, Huntsville, Alabama., pp.1-8.
2. MPF-1 Experiment Manual (Software-Hardware), Multitech Industrial Corporation, 1981.
3. MPF-1 User's Manual, Multitech Industrial Corporation, 1981.
4. Programming The Z80, Rodney Zaks, Sybex Inc., 1982
5. The TTL Data Book, Vol. 2, Texas Instruments Inc., 1985.
6. User's manual, Hewlett Packard Logic Development System (64000).

APPENDIX
SOFTWARE SOURCE LISTING

FILE: INITIALIZ:DATE

HEWLETT-PACKARD: Z80 Assembler 01.11

LOCATION OBJECT CODE LINE SOURCE LINE

```

1  "Z80"
2  *****
3  *          INITIALIZATION          *
4  *****
5  *
6  CONUT DEFS 15
7  SCAN  DEFS 15
8  *
9  *
10 ORG 1825H
1825 DD2A1D40 11 START      LD IX,[1D40H]      ; LOAD DISPLAY BUFF.
1829 3E3D      12          LD A,3DH
182B 321D45     13          LD [01D45H],A      ; DISPLAY " N "
182E CD000F     14          CALL SCAN
1831 FE00       15          CP 00H
1833 32180D     16          LD [0180DH],A
1836 CD0000     17          CALL CONUT
1839 321D41     18          LD [01D41H],A      ; SHOW FIRST VALUE
183C CD000F     19          CALL SCAN
183F FE00       20          CP 00H
1841 320180     21          LD [0180H],A
1844 CD0000     22          CALL CONUT
1847 321D42     23          LD [01D42H],A      ; SHOW SECOND VALUE
184A 3E00       24          LD A,00H
184C 321D43     25          LD [01D43H],A
184F 321D44     26          LD [01D44H],A      ; BLANK REMAINING DISP
1852 CD000F     27          CALL SCAN
1855 FE12       28          CP 12H      ; SHOW UNTIL "GO KEY"
1857 C21825     29          JP NZ,START ; IS PRESSED
185A 2A180D     30          LD HL,[180DH]
185D 4E         31          LD C,[HL]
185E 3A180E     32          LD A,[180EH]
1861 CB27       33          SLA A
1863 CB27       34          SLA A
1865 CB27       35          SLA A
1867 CB27       36          SLA A
1869 81         37          ADD A,C
186A 321801     38          LD [1801H],A      ; LOAD N IN 1801
39  *
40  *
186D 3E05       41          LD A,05H
186F 321802     42          LD [1802H],A      ; MODE A DEFAULT
43  *
44  *
45  * INITIALIZE THE CTC CHANNELS AND PIO PORTS
46  *
47  * PIO  PORT A
1872 3E00       48          LD A,00H
1874 D392       49          OUT (082H),A
1876 3E0F       50          LD A,0CFH      ; DEFINE BIT I/O
1878 D362       51          OUT (082H),A
187A 3EFF       52          LD A,0FFH      ; DEFINE ALL INPUTS
187C D362       53          OUT (082H),A
187E 3E03       54          LD A,03H
1880 D392       55          OUT (082H),A
1882 3E03       56          LD A,03H
1884 D392       57          OUT (082H),A

```

FILE: INITIALIZ:DATE HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE SOURCE LINE

```

58 *
59 * PIO PORT B
60 *
1886 3E00 61 LD A,00H
1888 D383 62 OUT [83H],A
188A 3ECF 63 LD A,0CFH
188C D383 64 OUT [83H],A
188E 3EA4 65 LD A,0A4H
1890 D383 66 OUT [83H],A ; BITS 2,5,7 AS I/P
1892 3E03 67 LD A,03H ; BITS 0,1,3,4,6 O/P
1894 D383 68 OUT [83H],A
1896 3E03 69 LD A,03H
1898 D383 70 OUT [83H],A
189A 3E03 71 LD A,03H
189C D381 72 OUT [081H],A ; RESET BOTH CIRCUITS
73 *
74 * CTC
75 *
189E 3E00 76 LD A,00H
18A0 D341 77 OUT [041H],A ; STROBE/SAMPLING
18A2 D342 78 OUT [042H],A ; CHANNELS ( A1,A2 )
18A4 3E45 79 LD A,45H
18A6 D341 80 OUT [041H],A
18A8 D342 81 OUT [042H],A
18AA 3EFF 82 LD A,0FFH
18AC D341 83 OUT [041H],A ; INITIAL VALUE
18AE D342 84 OUT [042H],A
85 *
86 *
18B0 3E00 87 LD A,00H
18B2 D3C1 88 OUT [0C1H],A ; RPM COUNTERS
18B4 D3C2 89 OUT [0C2H],A ; CHANNELS ( B1,B2 )
18B6 3E45 90 LD A,45H
18B8 D3C1 91 OUT [0C1H],A
18BA D3C2 92 OUT [0C2H],A
18BC 3EFF 93 LD A,0FFH
18BE D3C1 94 OUT [0C1H],A
18C0 D3C2 95 OUT [0C2H],A
96 *
97 *
18C2 3E00 98 LD A,00H
18C4 D3C0 99 OUT [0C0H],A
18C6 3E45 100 LD A,45H
18C8 D3C0 101 OUT [0C0H],A ; CHANNEL B0
18CA 3A1B02 102 LD A,[1B02H]
18CD D3C0 103 OUT [0C0H],A ; LOAD N
104 *
105 *
18CF 3E00 106 LD A,00H
18D1 D340 107 OUT [040H],A ; CHANNEL A0
18D3 3E45 108 LD A,45H
18D5 D340 109 OUT [040H],A
18D7 3A1B02 110 LD A,[1B02H]
18DA 3C 111 INC A
18DB D340 112 OUT [040H],A
113 *
114 *

```

FILE: INITIALIZ:DATE HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION	OBJECT CODE	LINE	SOURCE LINE
18DD	3EDF	115	LD A,0DFH
18DF	D3C3	116	OUT [0C3H],A ; CHANNEL B3
18E1	3A1801	117	LD A,[1801H]
18E4	D3C3	118	OUT [0C3H],A
18E6	3E00	119	LD A,00H
18E8	D3C0	120	OUT [0C0H],A
		121 *	
		122 *	
18EA	3E1E	123	LD A,1EH ; LOAD VECTORED INTR
18EC	ED47	124	LD I,A ; REGISTER
18EE	211AC0	125	LD HL,1AC0H
18F1	221FEE	126	LD [1FEEH],HL ; LOAD IM 1 ADDRESS
		127 *	
		128 *	
18F4	210505	129	LD HL,0505H
18F7	221803	130	LD [1803H],HL ; DEFAULT MODE B
18FA	221808	131	LD [1808H],HL ; DEFAULT MODE D
18FD	210A0A	132	LD HL,0A0AH
1900	221807	133	LD [1807H],HL ; DEFAULT MODE C
1903	3E01	134	LD A,01H
1905	321815	135	LD [1815H],A ; MODE EXECUTION CHECK
1908	3E5E	136	LD A,5EH
190A	321818	137	LD [1818H],A ; IM2 SET
190D	ED5E	138	IM 2
190F	FB	139	EI
		140 *	
		141 *	THIS CONCLUDES THE INITIALIZATION OF THE PARAMETERS
		142 *	
		143	END

Errors= 0

FILE: RPM:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE SOURCE LINE

```

1  "Z80"
2  *****
3  *          RPM          *
4  *****
0000 5 CONUT DEFS 15
000F 6 DIV  DEFS 10
0019 7 MULT DEFS 10
0023 8 SCAN1 DEFS 25
9  *
10 *
11 *
12  ORG 01910H
1910 3E02 13 BEGIN  LD A,02H          ; START RPM COUNTING
1912 D381 14      OUT [81H],A
1914 3E03 15      LD A,03H
1916 D381 16      OUT [81H],A
1918 00   17      NOP
1919 00   18      NOP
191A 00   19      NOP
191B DB91 20 CHK1  IN A,[81H]      ; CHECK FOR COUNT  COMPLETE
191D CB6F 21      BIT 5,A
191F CA191B 22      JP Z,CHK1
1922 DB81 23 CHK2  IN A,[81H]
1924 CB6F 24      BIT 5,A
1926 C21922 25      JP NZ,CHK2
1929 DEC1 26      IN A,[0C1H]      ; READ CHANNEL B1
192B 2F   27      CPL
192C 321811 28      LD [1811H],A      ; READ CHANNEL B2
192F DBC2 29      IN A,[0C2H]
1931 2F   30      CPL
1932 321812 31      LD [1812H],A
1935 3A1801 32      LD A,[1801H]
1938 4F   33      LD C,A
1939 0600 34      LD B,00H
193B ED5B1811 35      LD DE,[1811H]
193F CD000F 36      CALL DIV          ; DIVIDE BY N
1942 EB   37      EX DE,HL
1943 3E3C 38      LD A,3CH
1945 CD0019 39      CALL MULT        ; MULTIPLY BY 60
1948 220713 40      LD [1811],HL
194B DD2A1D40 41      LD IX,[1D40H]
194F 3A1811 42      LD A,[1811H]
1952 4F   43      LD C,A
1953 CD0000 44      CALL CONUT        ; CONVERT INTO DISPLAY FORMAT
1956 321D40 45      LD [1D40H],A
1959 79   46      LD A,C
195A CB7F 47      SRL A
195C CB7F 48      SRL A
195E CB7F 49      SRL A
1960 CB7F 50      SRL A
1962 CD0000 51      CALL CONUT
1965 321D41 52      LD [1D41H],A
1968 3A1812 53      LD A,[1812H]
196B 4F   54      LD C,A
196C CD0000 55      CALL CONUT
196F 321D42 56      LD [1D42H],A
1972 79   57      LD A,C

```

FILE: RPM:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION	OBJECT	CODE	LINE	SOURCE LINE
1973	CB3F		58	SRL A
1975	CB3F		59	SRL A
1977	CB3F		60	SRL A
1979	CB3F		61	SRL A
197B	CD0000		62	CALL CONUT
197E	321D43		63	LD [1D43H],A
1981	3E00		64	LD A,00H
1983	321D44		65	LD [1D44H],A
1986	321D45		66	LD [1D45H],A
1989	06FF		67	LD B,0FFH
198B	CD0023		68	CONTI CALL SCAN1 ; DISPLAY ON MPF DISPLAY
198E	05		69	DEC B
198F	C2198B		70	JP NZ,CONTI
1992	C31910		71	JP BEGIN
			72	*
			73	*
			74	* THIS CONCLUDES THE RPM PROGRAM
			75	END

Errors= 0

FILE: MODE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE SOURCE LINE

```

1  "Z80"
2  *****
3  *                               *
4  *                               *
5  *
6  LDSTB DEFS 20
7  DIV  DEFS 20
8  MULT DEFS 20
9  *
10
11      ORG 19A0H
12 *
13      LD A,41H
14      OUT [081H],A
15      LD A,53H
16      OUT [081H],A      ; START PROCESS
17      LD A,56H
18      LD [190EH],A
19      RETI
20 *
21 * THIS CONCLUDES MODE A EXECUTION ROUTINE
22 *
23 *
24      LD A,01H
25      OUT [081H],A
26      LD A,1BH
27      OUT [081H],A      ; START EXECUTION
28      LD A,56H
29      LD [190EH],A
30      RETI
31 *
32 * THIS CONCLUDES MODE B EXECUTION ROUTINE
33 *
34 *
35      DI
36      LD A,01H
37      OUT [081H],A
38      LD A,0BH
39      OUT [081H],A      ; START SAMPLING
40      NOP
41      NOP
42      NOP      ; TIMING DELAY
43      CHKC1 IN A,[081H]
44      BIT 2,A      ; CHECK FOR SAMPLING
45      JP Z,CHKC1    ; PROCESS TO FINISH
46      LD A,57H
47      OUT [0C0H],A      ; RESET B0
48      OUT [040H],A      ; RESET A0
49      IN A,[041H]      ; READ A1
50      CPL
51      LD [1809H],A      ; STORE
52      IN A,[042H]      ; READ A2
53      CPL
54      LD [180AH],A      ; STORE
55      OUT [041H],A      ; RESET A1
56      OUT [042H],A      ; RESET A2

```

FILE: MODE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE

SOURCE LINE

```

58 * PREPARE FOR MULT AND DIV ROUTINES
59 *
19E7 ED5B1809 60 LD DE,[1809H]
19EB ED4B1807 61 LD BC,[1807H]
19EF CD0014 62 CALL DIV ; PERFORM P/M (SEE TEXT)
19F2 7D 63 LD A,L
19F3 C600 64 ADD A,00H
19F5 CA1A48 65 JP Z,REVRS
19F8 7C 66 LD A,H
19F9 C600 67 ADD A,00H
19FB CA1A48 68 JP Z,REVRS ; IF P/M = 0 THEN GOTO
19FE EB 69 CONTIN EX DE,HL ; REVRS ELSE CONTINUE
19FF 3A1806 70 LD A,[1806H]
1A02 CD0028 71 CALL MULT ; MULTIPLY BY 1ST BYTE
1A05 37 72 SCF
1A06 3F 73 CCF
1A07 CB23 74 SLA E ;SHIFT THE ANSWER LEFT
1A09 CB12 75 RL D ; BY 1 FOR SECOND BYTE
1A0B 3A1805 76 LD A,[1805H]
1A0E CD0028 77 CALL MULT ; MULTIPLY BY SECOND BYTE
1A11 221805 78 STRBC LD [1805H],HL
1A14 221811 79 LD [1811H],HL ; GET READY FOR LDSTB
1A17 2A1809 80 LD HL,[1809H]
1A1A 221807 81 LD [1807H],HL ; GET READY FOR NEXT REVRS
82 *
1A1D 3A181B 83 LD A,[181BH]
1A20 D601 84 SUB 01H
1A22 CA1A6F 85 JP Z,RUNC ; THE RUN MODE IS SELECTED
1A25 3E01 86 LD A,01H
1A27 32181D 87 LD [181DH],A ; CHECK POINT FOR SYNCHRO
88 *
1A2A CD0000 89 RUNEND CALL LDSTB ; MODE
1A2D 3E45 90 LD A,45H
1A2F D3C0 91 OUT [0C0H],A
1A31 3A181D 92 LD A,[181DH] ; LOAD 1 IN CHANNEL B0
1A34 3E01 93 LD A,01H
1A36 D3C0 94 OUT [0C0H],A
95 *
1A38 3E01 96 LD A,01H
1A3A D3B1 97 OUT [0B1H],A
1A3C 3E1B 98 LD A,1BH
1A3E D3B1 99 OUT [0B1H],A ; START FINAL COUNTDOWN
1A40 3E56 100 LD A,56H
1A42 32190E 101 LD [190EH],A
1A45 FB 102 EI
1A46 ED4D 103 RETI
104 *
1A48 7C 105 REVRS LD A,H
1A49 C600 106 ADD A,00H
1A4B C219FE 107 JP NZ,CONTIN
1A4E ED5B1809 108 LD DE,[1809H]
1A52 3A1806 109 LD A,[1806H]
1A55 CD0028 110 CALL MULT ; MULTIPLY P BY 1ST BYTE
1A58 37 111 SCF
1A59 3F 112 CCF
1A5A CB23 113 SLA E ; SHIFT DE FOR NEXT BYTE
1A5C CB12 114 RL D

```

FILE: MODE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION	OBJECT	CODE	LINE	SOURCE	LINE
1A5E	3A1805		115	LD A,[1805H]	
1A61	CD0028		116	CALL MULT	
1A64	EB		117	EX DE,HL	
1A65	ED4B1807		118	LD BC,[1807H]	; PERFORM (P*Q)/M
1A69	CD0014		119	CALL DIV	
1A6C	C31A11		120	JP STRBC	
			121 *		
1A6F	AF		122	RUNC XOR A	
1A70	2A1811		123	LD HL,[1811H]	; FINAL STROBE VALUE
1A73	ED5B1813		124	LD DE,[1813H]	; TWS VALUE
1A77	1600		125	LD D,00H	
1A79	ED5A		126	ADC HL,DE	; ADD TWS TO STROBE
1A7B	221811		127	LD [1811H],HL	; CHECK FOR S>M
1A7E	ED5B1809		128	LD DE,[1809H]	
1A82	ED52		129	SBC HL,DE	
1A84	D21A93		130	JP NC,ENGADU	; IF YES THEN INCREASE
1A87	3E01		131	LD A,01H	; TARGET PULSE BY ONE
1A89	32181D		132	LD [181DH],A	
1A8C	ED531807		133	LD [1807H],DE	
1A90	C31A2A		134	JP RUNEND	
1A93	ED531811		135	ENGADU LD [1811H],DE	
1A97	3E02		136	LD A,02H	
1A99	32181D		137	LD [181DH],A	
1A9C	ED531807		138	LD [1807H],DE	
1AA0	C31A2A		139	JP RUNEND	
			140 *		
			141 *	THIS CONCLUDES MODE C EXECUTION ROUTINE	
			142 *		
			143 *		
1AA3	3E01		144	LD A,01H	
1AA5	D381		145	OUT [081H],A	
1AA7	3E1B		146	LD A,1BH	
1AA9	D381		147	OUT [081H],A	; START EXECUTION
1AAB	3E56		148	LD A,56H	
1AAD	32190E		149	LD [190EH],A	
1AB0	ED4D		150	RET	
			151 *		
			152 *		
			153 *	THIS CONCLUDES THE MODE ROUTINES	
			154 *		
			155	END	

Errors= 0

FILE: UPDATE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE

SOURCE LINE

```

1  "Z80"
2  *****
3  *          STATUS UPDATE          *
4  *****
5  *
0000 6 BOA      DEFS 25      ; LOAD CHANNEL B0 FOR MODE A
0019 7 LDSTB    DEFS 25      ; LOAD STROBE VALUE IN A1,A2
0032 8 AOC      DEFS 25      ; LOAD CHANNEL A0 FOR SAMPLING
0048 9 A1A2C    DEFS 25      ; PREPARE CHANNELS A1,A2 FOR SAMPL
10 *
11 *
12  ORG 01AC0H
13 *
1AC0 3E57     14      LD A,57H      ; RESET ALL THE CHANNELS
1AC2 D340     15      OUT [040H],A
1AC4 D3B0     16      OUT [0B0H],A
1AC6 D341     17      OUT [041H],A
1AC8 D342     18      OUT [042H],A
1ACA AF       19      XOR A
1ACB DB50     20      IN A,[80]      ; READ THE STATUS SWITCHES
1ACD CB3F     21      SRL A
1ACF CB3F     22      SRL A
1AD1 CB3F     23      SRL A
1AD3 CB3F     24      SRL A
1AD5 4F       25      LD C,A      ; STORE TWS VALUE
1AD6 321813   26      LD [1813H],A
1AD9 3A1800   27      LD A,[1800H]  ; MODE IDENTIFY
1ADC E603     28      AND 03H
1ADE D602     29      SUB 02H
1AE0 CA1BAC   30      JP Z,UPDTC      ; GO TO MODE C
1AE3 3A1800   31      LD A,[1800H]
1AE6 CB5F     32      BIT 3,A
1AE8 CA1B09   33      JP Z,GOBAC
1AEB 3A1816   34      LD A,[1816H]
1AEE D601     35      SUB 01H
1AF0 CA1B09   36      JP Z,GOBAC
1AF3 3E01     37      LD A,01H
1AF5 321816   38      LD [1816H],A
1AF8 3F       39      CCF
1AF9 3A1800   40      LD A,[1800H]
1AFC E603     41      AND 03H
1AFE D601     42      SUB 01H
1B00 CA1B69   43      JP Z,UPDTB      ; GO TO MODE B
1B03 DA1B21   44      JP C,UPDTA      ; GO TO MODE A
1B06 C31BED   45      JP UPDTD      ; GO TO MODE D
1B09 3E00     46  GOBAC  LD A,00H      ; IF THE ENTER/RUN SWITCH IS NOT
1B0B 321816   47      LD [1816H],A  ; PRESSED THEN GO BACK AND
1B0E 3A1815   48      LD A,[1815H]      ; EXECUTE THE LAST MODE
1B11 3A1800   49      LD A,[1800H]
1B14 E603     50      AND 03H
1B16 D601     51      SUB 01H
1B19 CA1B60   52      JP Z,GOBB      ; GOBACK TO MODE B
1B1B DA1B3E   53      JP C,GOBA      ; GOBACK TO MODE A
1B1E C31C0A   54      JP GOBD      ; GOBACK TO MODE D
55 *
56 *
57 * THIS CONCLUDES THE MODE IDENTIFICATION:

```

FILE: UPDATE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION OBJECT CODE LINE SOURCE LINE

```

58 *
59 * NOW FOLLOWS THE ACTUAL UPDATE OF EACH MODE*
60 *
1821 3E01      61 UPDTA LD A,01H
1823 321815    62      LD [1815H],A
1826 3A1800    63      LD A,[1800H]
1829 CB57      64      BIT 2,A
182B CA1857    65      JP Z,DECA      ; THIS MEANS THE TWS IS TO ADDED
182E 3A1802    66      LD A,[1802H]
1831 81        67      ADD A,C
1832 47        68      LD B,A
1833 3A1801    69      LD A,[1801H]
1836 90        70      SUB B
1837 DA1B4E    71      JP C,NM1A      ; THIS MEANS STROBE VALUE IS > N
183A 78        72      LD A,B
183B 321802    73 AEND  LD [1802H],A
183E CD0000    74 GOBA  CALL BOA      ; LOAD CHANNEL BO WITH THE UPDATE
1841 3E5E      75      LD A,5EH
1843 32190E    76      LD [190EH],A      ; LOAD THE IM2 BYTE
1846 2A0758    77      LD HL,[1880]
1849 221E06    78      LD [1E06H],HL
184C ED4D      79      RETI
184E 0601      80 NM1A  LD B,01H
1850 3A1801    81      LD A,[1801H]
1853 90        82      SUB B      ; LOAD STROBE = N-1
1854 C31B3B    83      JP AEND
1857 3A1802    84 DECA  LD A,[1802H]
185A 91        85      SUB C
185B CA1B61    86      JP Z,ONEA
185E DA1B61    87      JP C,ONEA      ; IF THE NEW VALUE IS <= 0 THEN
1861 3E01      88 ONEA  LD A,01H      ; RESET IT BACK 01
1863 321802    89      LD [1802H],A
1866 C31B3B    90      JP AEND
91 *
92 *
93 * THIS CONCLUDES THE UPDATE OF MODE A
94 *
1869 3E02      95 UPDTB LD A,02H
186B 321815    96      LD [1815H],A
186E 3A1800    97      LD A,[1800H]
1871 CB57      98      BIT 2,A
1873 CA1B99    99      JP Z,DECB      ; THIS MEANS DECREASE STROBE VALUE
1876 2A1803   100      LD HL,[1803H]
1879 0600     101      LD B,00H
187B ED4A     102      ADC HL,BC
187D 221803   103 BEND  LD [1803H],HL
1880 CD0000   104 GOBB  CALL BOA
1883 2A1803   105      LD HL,[1803H]
1886 221811   106      LD [1811H],HL
1889 CD0019   107      CALL LDSTB      ; LOAD CHANNELS A1,A2 FOR STROBE
188C 3E5E     108      LD A,5EH
188E 32190E   109      LD [190EH],A
1891 211880   110      LD HL,1880H
1894 221E06   111      LD [1E06H],HL      ; LOAD STARTING ADDRESS OF MODE B
1897 ED4D     112      RETI
1899 2A1803   113 DECB  LD HL,[1803H]
189C 0600     114      LD B,00H

```

FILE: UPDATE:DATE

HEWLETT-PACKARD: 280 Assembler V1.11

LOCATION OBJECT CODE LINE

SOURCE LINE

```

189E ED42      115      SBC HL,BC
18A0 DA18A6    116      JP C,SET1      ; IF THE STROBE VALUE IS <= 0
18A3 C31E7D    117      JP BEND      ; THEN SET IT TO 01
18A6 21C101    118 SET1  LD HL,0101H
18A9 C31E7D    119      JP BEND
120 *
121 *
122 * THIS CONCLUDES THE UPDATE OF MODE B
123 *
18AC 3E03      124 UPDTC LD A,03H
18AE 321815    125      LD [1815H],A
18B1 3A1800    126      LD A,[1800H]
18B4 CE5F      127      BIT 3,A      ; CHECK FOR SYNCRO OR RUN MODE
18B6 C21ECF    128      JP NZ,RUN      ; USER HAS SELECTED RUN MODE
18B9 211900    129      LD HL,1900H
18BC 221E06    130      LD [1E06H],HL      ; LOAD STARTING ADDRESS OF SYNCRO
18BF CD0000    131 CDEND  CALL BOA
18C2 CD0032    132      CALL AOC
18C5 CD004B    133      CALL A1A2C
18C8 3E5E      134      LD A,5EH
18CA 32190E    135      LD[190EH],A
18CD ED4D      136      RETI
18CF 3A1800    137 RUN   LD A,[1800H]
18D2 CE57      138      BIT 2,A
18D4 CA18E5    139      JP Z,DECC      ; ROTATE THE IMAGE COUNTER CLOCK
18D7 3E01      140      LD A,01H
18D9 321818    141      LD [1818H],A
18DC 211900    142 ENDC  LD HL,1900H
18DF 221E06    143      LD [1E06H],HL
18E2 C318BF    144      JP CDEND
18E5 3E01      145 DECC  LD A,01H
18E7 321818    146      LD [1818H],A
18EA C318DC    147      JP ENOC
148 *
149 *
150 * THIS CONCLUDES UPDATE OF MODE C
151 *
18ED 3E04      152 UPDTC LD A,04H
18EF 321815    153      LD [1815H],A
18F2 AF        154      XOR A
18F3 3A1800    155      LD A,[1800H]
18F6 CB57      156      BIT 2,A
18F8 CA1C1D    157      JP Z,DECD      ; STROBE VALUE IS TO BE DECREASED
18FB 2A180B    158      LD HL,[180BH]
18FE 0600      159      LD B,00H
1C00 09        160      ADD HL,BC
1C01 DA1C30    161      JP C,FIXU
1C04 22180B    162 DEND  LD [180BH],HL
1C07 221811    163      LD [1811H],HL
1C0A CD0000    164 GOBO  CALL BOA
1C0D CD0019    165      CALL LDSTB
1C10 3E5E      166      LD A,5EH
1C12 32190E    167      LD [190EH],A
1C15 211A00    168      LD HL,1A00H
1C18 321E06    169      LD [1E06H],A
1C1B ED4D      170      RETI
1C1D 2A180B    171 DECD  LD HL,[180BH]

```

FILE: UPDATE:DATE

HEWLETT-PACKARD: Z80 Assembler V1.11

LOCATION	OBJECT CODE	LINE	SOURCE LINE
1C20	0600	172	LD B,00H
1C22	ED42	173	SBC HL,BC
1C24	DA1C2A	174	JP C,ONED
1C27	C31C04	175	JP DEND
1C2A	210101	176	ONED LD HL,0101H
1C2D	C31C04	177	JP DEND
1C30	21FFFF	178	FIXU LD HL,0FFFFH
1C33	C31C04	179	JP DEND
		180	*
		181	*
		182	* THIS CONCLUDES THE STATUS UPDATE
		183	END

Errors= 0

VITA

Sanjay Shreeniwas Date was born in San Jose, California, on April 17, 1963. At a young age he went to India and stayed in Phaltan, India until he completed high school. He then moved to Pune, India, where he received a Bachelor of Science degree in Physics in June, 1983. He then entered University of Poona, Pune, India where he received his second Bachelor of Science Degree in Advanced Electronics in July 1984. He came to the U.S. and joined Tennessee Technological University at Cookeville, TN. in September 1984. He transferred to University of Tennessee Space Institute in Tullahoma, Tennessee in September 1985 and received the Master of Science Degree in Engineering Science and Mechanics in November, 1987. He has started working at the Redstone Arsenal, Huntsville, Alabama in the TMDE Support Group of the U. S. Army as an electronic engineer. He is married to the former Jayashree Vaidya of Pune, India.