To the Graduate Council:

I am submitting herewith a thesis written by Dilip P. Nawathe entitled " Design of the Velocity Measurement System using a heat source and IR detectors." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science, with a major in Electrical Engineering.

R. W. Rochelle

R. W. Rochelle, Major professor

We have read this thesis
and recommend its acceptance:

Donald W. Bouldin

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

DESIGN OF THE VELOCITY MEASUREMENT SYSTEM USING A HEAT-SOURCE AND IR DETECTORS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Dilip P. Nawathe

June1987

This thesis is dedicated to

my wife Veena

and my family

as a token of the love I have for you.

ACKNOWLEDGEMENTS

The author wishes to express sincere appreciation and gratitude to Dr. R. W. Rochelle and Dr. T. W. Kerlin for their invaluable guidance during the period of this study. Sincere appreciation is also expressed to Dr. D. W. Bouldin for his helpful suggestions and review of this work. Thanks are also due to Dr. David Straight for making available HP64000 development system and various data manuals of M68000 microprocessor family and Mr. Franklin Wade for helping me in the laboratory.

Abstract

This thesis reports the design of a system to measure the velocity of moving objects or surfaces. A quartz filament lamp is used to create a hot spot on the moving object and this hot spot is detected by two infra-red detectors at a known distance from each other. The time required for the hot spot to travel this distance is measured to determine the velocity. This scheme of velocity measurement does not require the object to have any leading or trailing edge, as do most of the other velocity measurement systems. Therefore, this velocity measurement system can be used for measuring the velocity of a continuous rope or plastic film being wound on a spindle.

Three approaches are used for the design of the Velocity Measurement System. The shortcomings of the initial design are overcome in the subsequent design approaches. The first approach uses hardware components, mainly CMOS integrated circuits and no software is involved. The second approach is based on the use of bit-slice devices, like the AM2903 four-bit microprocessor slice, the AM2910 sequencer, etc. In the third approach, a Motorola MC68000, 16-bit microprocessor is used. Irrespective of the actual design approach, the basic principle of velocity measurement is the same. Schemes for accurate time measurement, which overcome problems arising from the deformation of the signal at the output of the second IR detector or due to conduction of heat from the hotspot to the rest of the body of the moving object, are implemented using the microprocessor based design.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The velocity measurement system put forward here is able to measure velocities of solids with a noncontact type of a sensor or without using any technique involving the use of optical or magnetic or any other type of transmitting device mounted on the moving object.

One of the applications of such a type of velocity measurement system is continuous measurement of velocity of a plastic film moving over two rollers or being wound on a spindle. Due to the problem of slippage, it is not possible to measure the rotational speed of the rollers and correlate it to the velocity of the plastic film. As there is no leading edge, opto-interruptive type of sensors or any other similar technique can not be used. Dr. T. W. Kerlin proposed the scheme for developing a velocity measurement scheme which can overcome these problems and Dr. R.W. Rochelle suggested several circuit designs by which the principle put forth by Dr. T. W. Kerlin can be implemented.

Principle of Operation

By focusing an intense beam of light from a quartz filament lamp or laser, a hot spot is formed on a moving object and the time required for this hot spot to travel a known distance between two infra red detectors (IR detectors) is monitored. The velocity of the moving object is the distance between the two IR detectors divided by the time measured. Out of the three design approaches discussed later, in the first two approaches it was assumed that the source of heat is powerful and efficient enough to create a hot spot on the surface of a moving object within a very short period of time. To achieve this, it is essential that the source of heat should impart heat to a spot instantly. If the object is moving at high speed, then this requirement becomes more stringent because a spot on the moving object passes in front of the source of heat before the source of heat can actually increase the temperature of that spot a couple of degrees higher than the rest of the body. At higher speeds, a spot on the object is in front of the source of heat for only a few milliseconds. In the first two designs it was essential that the source of heat should heat up the spot within that small time. This can only be done using a high-power laser. Also the temperature to which the spot should be heated depends on the physical properties of the moving object e.g. in the case of insulators, heat will be conducted to the rest of the body slowly and so the spot may remain hot enough to be detected by IR detectors.

Due to an inefficient source of heat and conduction of heat away from the heated spot, it is difficult to get a hot spot . So at the output of the IR detectors one does

not get a sharp pulse when a hot spot is detected,instead one gets waveforms as shown in figure 1.1. In the microprocessor based design approach, a special scheme is used to ensure an accurate time measurement.

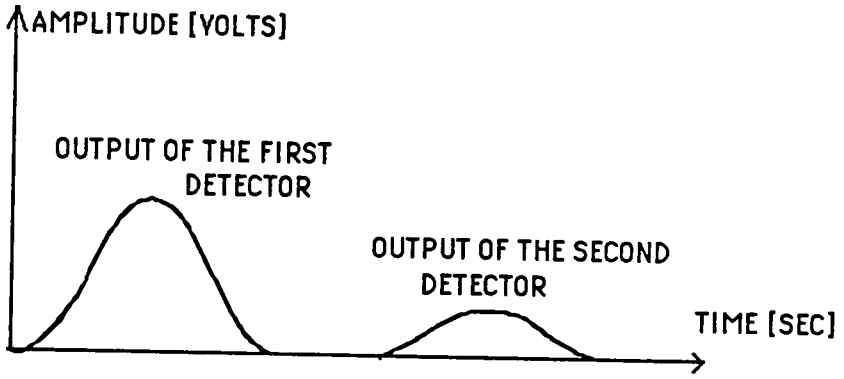


Fig. 1.1: Output signal waveform of the IR detectors.

The power and efficiency of the heat source, its distance from the moving object and also from the IR detectors depends on the physical properties of the moving object and its maximum velocity.

Data is compiled for the ropes and fibers of various materials, sizes and moving at various velocities over the pulleys driven by the motors. Optimum distance between the IR detectors and the source of heat for the various cases is determined.

Infrared Detector

Parallel-opposed dual infrared detectors manufactured by Eltec Instruments Inc. (model # P-101-3) are used in the prototype of this velocity measurement system.

The basic detector consists of a thin wafer of lithium tantalate with metal electrodes deposited on both of its faces. When infrared light is incident on the sensor, charge is generated and is gathered by the electrodes. This charge does not leak off since lithium tantalate is a very good insulator. So the pyroelectric detector can be considered as a combination of a charge generator and a capacitor. Lithium tantalate is a synthetic crystal and displays good pyroelectric properties i.e. change in temperature creates a change in its electrical polarization. Hence it produces a current only when it experiences a change in temperature. Sometimes a black coating is applied over the electrodes so as to make them good absorbers of thermal radiation or infrared light. But blackening can degrade high-frequency performance because it introduces thermal diffusion time i.e. the coating may act as a thermal cushion.

Dual-element IR detectors are used to make the system immune to extraneous signals generated in response to ambient temperature changes or rapid gross changes in the field of view. In the dual-element parallel-opposed configuration of the sensing elements these signals tend to be self cancelling.

Because of the high output impedance of the pyroelectric sensing element, most detectors have integral electronics consisting of a voltage follower or a current-

mode amplifier. Detectors with a voltage follower offer a low output impedance and high signal-to-noise ratio. The thermal time constant of the crystal in conjunction with the effective capacitance of the sensing element and the load resistor determine the detector's frequency response.

Initially parallel-opposed dual IR detectors (model number 4192) were used as they have high responsivity. But an internal load resistor of 10^{11} ohm with a 30 pF capacitance of the sensing element was found to make the detector too sluggish. The rise time was observed to be about one second and so the sensor was unsuitable for the present application.

The latest version of the IR detector introduced by Eltec lustruments (model # P-101-3) has a low load-resistor value and consequently has a low responsivity. But this problem is overcome by incorporating an on-chip integrated signal-processing circuit. This unit offers a greatly improved detection capability. The frequency response of this detector is specified as 1KHz.The signal is amplified by a factor of one hundred.

For the present application an HP7(-3) window which blocks shorter wavelengths from the sun and electric lights and gives high transmittance to the band 7.5 to 12 μm of infrared light was found suitable. Initially it was decided to determine the velocities of various plastic and other ropes with emissivities between 0.8 and 0.9 . The temperature of the hot spot was likely to be in the temperature range of 25 to 100^0C. This made the infrared emission in the above mentioned range of 7.5 to 12 μm

very important for detecting the change in temperature.

An optical head with an IR transmissive fresnel lens was used for the IR detector. This gave a 6^0 field of view and was responsible for the improved performance.

Advantages of Using a Two-detector Scheme

Initially it was decided to use a single source of heat and only one IR detector instead of two. But instead of one source of heat and a single IR detector, a scheme that uses two IR detectors with one source of heat is found to have some additional advantages particularly for improving the accuracy of the reading.

In the single IR detector scheme, the counter starts counting time at the instant when power is applied to the source of heat, such as a lamp. In practice after the lamp is turned on, it takes a finite time for the spot on the moving body to heat up. This error is negligible in the case of slow moving objects, but may create accuracy problems in high velocity measurements.

Slow response of the IR detector may also add some inaccuracy. The single IR detector scheme can give accurate readings for IR detectors with ideal high-frequency characteristics. Due to conduction of heat from the hot spot to the rest of the body of the moving object, it is not possible to get a sharp edge at the output of the IR detector. In the scheme with two IR detectors this error due to non ideal response of IR detectors gets cancelled. Refer to figure 1.2.

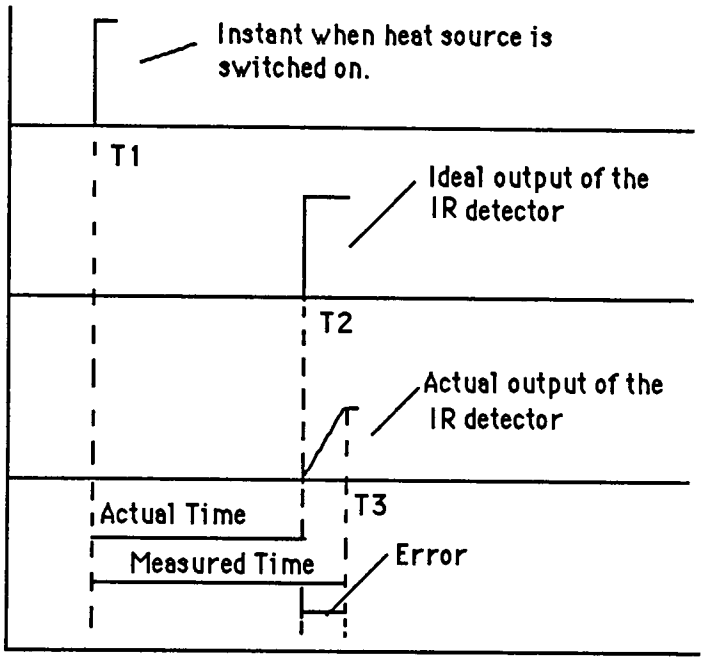


Fig.1.2: Error due to poor high-frequency response of the IR detectors and conduction of heat away from the hot spot.

As the electronic circuitery responds at the same amplitude of the output signal for both the detectors, error due to sluggish response of the IR detectors can be eliminated. Special schemes are used to eliminate further inaccuracy in time measurement due to unequal slopes of the two wave forms. Another advantage of the two-detector scheme is that if the moving object has a leading edge, it is sensed by the IR detectors due to the change in infrared emission being received by the detectors, and the velocity of the object could be measured without using the heat source. Whereas in the case of the velocity measurement of say, a rope running over motor

driven pulleys, measurement can be taken with a heat source by creating a hot spot on the rope, as there is no leading edge that can be sensed.

CHAPTER 2

DESIGN OF THE VELOCITY MEASUREMENT SYSTEM USING SSI AND MSI TYPE OF INTEGRATED CIRCUITS

Objective

The subject of this chapter is the design and manufacture of a portable velocity measuring system operating on a nine-volt D.C. power supply. It was decided that the battery and printed circuit board with most of the hardware should be encased in a box which the operator should be able to put in a bag on his shoulder.

The heat source and the IR detectors should be mounted on a handheld pistol-like structure with a 3-digit digital display on its rear side facing the operator. These two parts of the system should be connected with a four-feet-long multicore cable with a connector. A discription of the various blocks shown in the figure 2.1 follows.

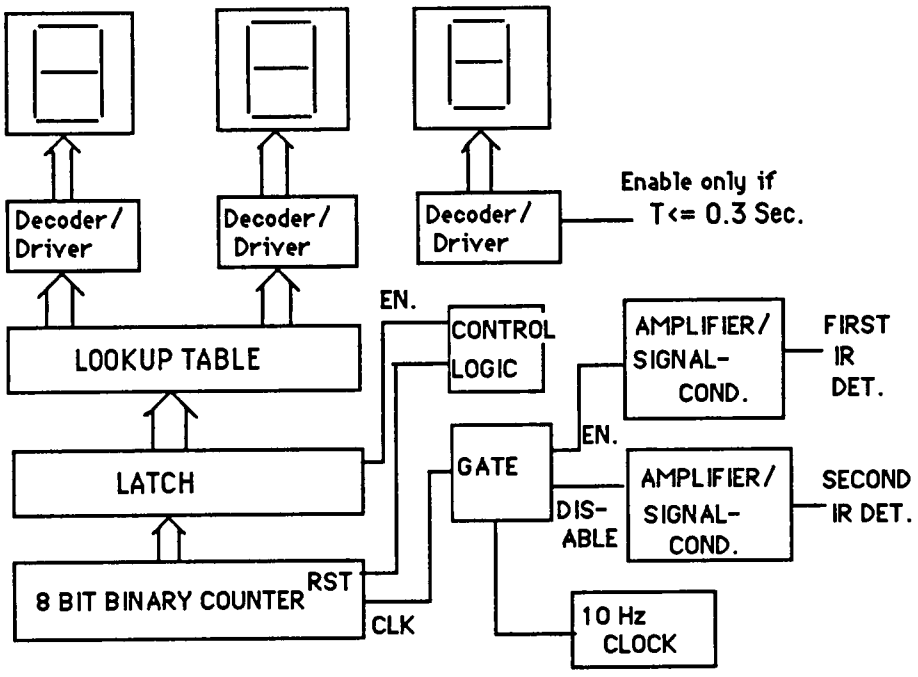


FIG.2.1: Functional block diagram

The Clock

The 10-Hz and 1-Hz clock signals are derived from the MM5369 integrated circuit which is a crystal controlled oscillator with built-in counters. The circuit diagram is shown in figure 2.2.

MM 5369 is a CMOS integrated circuit with seventeen binary divider stages which are used to generate a precise 60-Hz reference from commonly available low cost 3.58-MHz crystal. An output accuracy of 2 PPM is available.

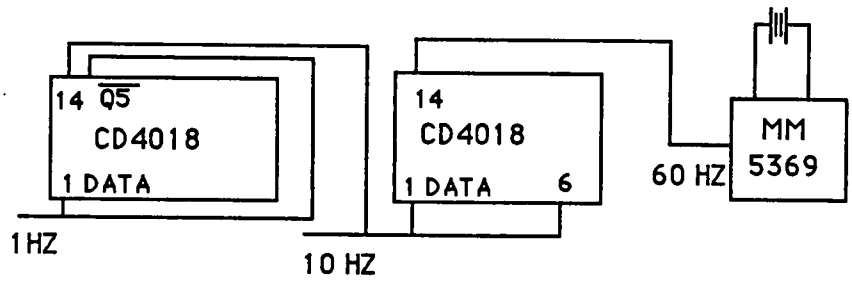


Fig 2.2: The Clock.

This 60-Hz output is first divided by six and then divided by ten to get 10 Hz (i.e. 0.1 sec.) and 1-Hz (1 sec.) signals using a CD4018 which is a presettable divide-by-N counter. If Q_3 (pin 6) is connected to the 'DATA' input, it gives a signal with a frequency six times smaller. Similarly the divide-by-ten operation is obtained by connecting Q_5 to the 'DATA' input of the second CD4018.

The 10-Hz signal which gives a rising edge after every 0.1 sec. is used as the basic clock of the system while the 1-Hz signal is used to latch the data from the counter after the hot spot is detected by the second IR detector.

Amplifier and Signal Conditioning

The circuit diagram of the amplifier and signal conditioning circuit is shown in figure 2.3. The LM324 quad-operational amplifier is used because it can be used with a single power supply. (9-V d.c.supply in the present case.)

A signal from the IR detector is fed through a buffer to the inverting

amplifier. This amplifier has a gain of 100. The IR detector requires a reference voltage of 2.5 Volts. So in the steady state condition the IR detector gives approximately 2.5 volts at its output . Depending on the temperature of the hot spot and the emissivity of the moving object, the IR detector output rises from 2.5 Volts to about 4.7 Volts (i.e. saturation voltage) when the hot spot comes in front of the detector. Initially the offset voltage of the amplifier is adjusted such that it is little higher than the d.c. output voltage of the IR detector, in the steady state condition.

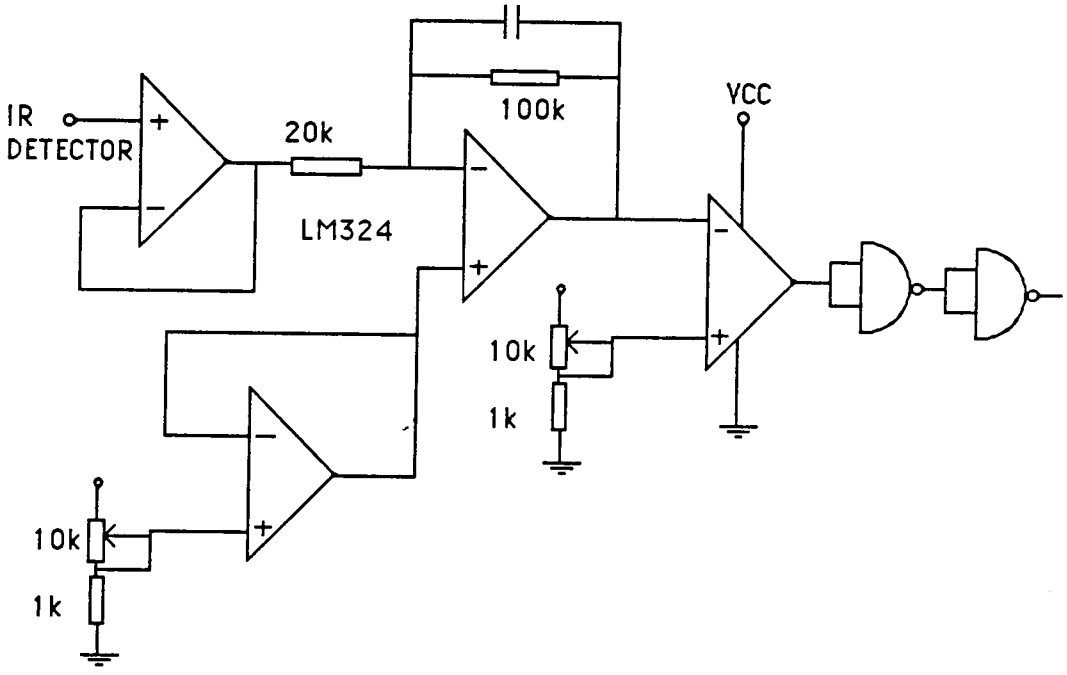


Fig. 2.3: Amplifier and signal-conditioning

When the output of the IR detector goes higher than this offset voltage, the differential signal is amplified and the output of the amplifier goes low. The output of the amplifier is connected to the comparator. When the inverting input of the

comparator (i.e. output of the amplifier) goes lower than the voltage on the non inverting input, the output of the comparator goes high.

The comparator at the output of the amplifier switches its output between '0'V and Vcc depending on the input signal from the IR detector. This output can easily drive the CMOS buffers and is found to be compatible.

Control Logic, Gate, Counters and Latches

From the circuit diagram shown in figure 2.4 it can be seen that the output of the first IR detector is amplified and the 1 or 0 level at the output of the comparator is fed to the buffer. When the hotspot comes in front of the first detector, a high level is fed to 'set' the input of the CD4013 which is a dual D flip-flop. This make its Q output high. This Q output is connected to one input of a 2-input nand gate (CD4011) while 10-Hz pulses from the clock are continuously fed to the other input. A high level at the Q output of CD 4013 (D1 flip-flop) enables the 10-Hz pulses to appear at the output of the NAND gate.These are fed through an inverter to a binary counter viz. CD4029. This counter is used in the binary up-counting mode. The carry out of the first counter is fed to the clock input of the next 4-bit binary counter. This arrangement gives an 8-bit binary counter. All the 'jam'input pins of these counters are grounded which in turn 'reset' the counter when the preset enable is made high.

When the hot spot comes in front of the second IR detector, the amplifier and the comparator at its output produce a 'high' level at the 'reset' input of the D1

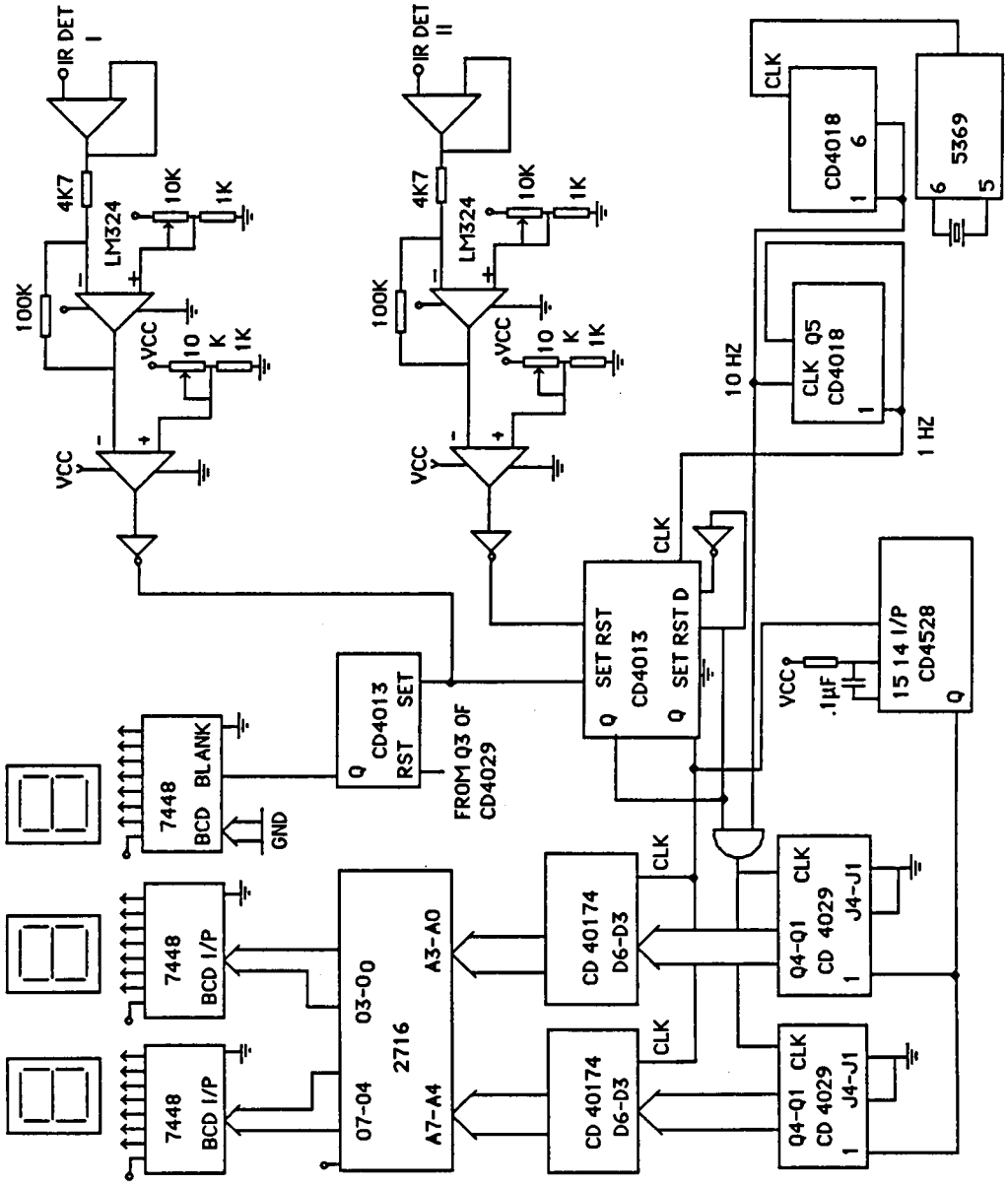


Fig. 2.4: Circuit diagram.

flip-flop.This makes its Q output go 'low', thereby disabling the 10-Hz pulses fed to the counter. So the count in the counter is incremented after every 0.1 second when the gate is enabled. At the instant when the gate is disabled, the counter holds the binary count for the time, with a resolution of 0.1 sec.

When the Q output of the above mentioned D1 flip-flop goes low, disabling the gate, the same signal is also fed to the 'reset' of the other D flip-flop, say D_2. So the Q_2 output of D_2 is low. The D input of the D_2 flip-flop is also high at this instant. The output of the 1-Hz clock is fed to the clock input of the D_2 flip-flop. Hence after the gate is disabled by D1, the first rising edge of the output of the 1-Hz clock will make the Q_2 output of the D_2 high. On this rising edge at the Q_2, the 8-bit binary count present at the output of counters is latched by the CD40174 Hex D flip-flop.

This output remains unchanged until the operator takes another reading of velocity i.e. until another hot spot passes by both the IR detectors.

Lookup Table and the Displays

The lookup table and display section of the block diagram shown in the table 2.1 is discussed in details on the following pages.

$$\text{Velocity of the moving object} = \frac{\text{Distance between the two IR detectors}}{\text{Time counted by the counter}}$$

As this division can be performed only by some complex hardware, controlled by a microprogram, the easiest solution is using a lookup table. It is assumed that the distance between the two IR detectors is 30 cm. and the time required by the hotspot to travel from one IR detector to the next may vary over the interval 0.1 sec to 5 sec. So intially the range of the velocity measurement is 30 cm./ 5 sec. = 6 cm. per sec. to 30 cm / 0.1 sec = 300 cm per sec.

This lookup table is entered into a 2716 EPROM which has a 2K memory. The 8- bit binary output of the latch, representing the time elapsed with a resolution of 0.1 sec., is connected to A0-A7 address pins of the EPROM. The BCD equivalent of the actual velocity to be displayed, corresponding to each reading of time in the counter, is stored in the memory. So the binary number representing the time is the 'address' of the memory location and the data stored at that address location is the velocity in the BCD form. Some entries of the lookup table are shown below.

For the time intervals between 0.4 sec. to 5.0 secs. , a two-digit display is used to display the velocity. The velocity is displayed as an integer and the fraction part is ignored.

Table 2.1: A section of the lookup table.

MEMORY ADDRESS(TIME)		DATA (VELOCITY)	
DECIMAL	BINARY EQUIVALENT	DECIMAL	BCD EQUIVALENT
0.1	0000 0001	300	0011 0000
0.2	0000 0010	150	0001 0101
0.3	0000 0011	100	0001 0000
0.4	0000 0100	75	0111 0101
⋮			
1.0	0000 1010	30	011 0000
⋮			
5.0	0011 0010	06	0000 0110

If the hotspot requires, for example, 0.4 secs. to reach the second IR detector after passing by the first one, the counter counts four pulses, each being fed to the counter after every 0.1 sec. The output of the counter i.e. 0000 0100 is latched. This binary count is treated as an address of the memory location, at the location addressed by the counter the correct velocity, 75 cm/sec in this case, is stored in BCD form. The memory outputs this BCD number which is fed to the BCD-to-seven segment decoder/driver. This decoder/driver activates the required segments of the LED display thereby displaying the velocity.

Control of the Least Significant Digit

If the time interval measured is less than 0.3 secs., then the velocity is 100 cms/sec or more e.g. for 0.1 sec. the velocity is 300 cms/sec and for 0.2 sec. it is 150 cms/sec. For the prototype only one EPROM was used and as each memory location has 8 bits, only two BCD numbers could be represented using these 8 bits. Now for three readings viz. 0.1, 0.2 and 0.3 secs., a third digit is required to display the least significant digit of the corresponding velocities, which is '0' . So for these three readings of time, only the first two digits of the velocity are entered in corrosponding memory locations i.e. 30, 15 and 10 for 0.1, 0.2 and 0.3 sec. readings. A simple control logic is used to display a '0' in the least significant place so that the velocities can be read as 300, 150 and 100. So this least significant digit display, displays '0' for 0.1, 0.2 and 0.1 sec. readings and is completely blanked out for all other readings as they can be displayed by only two digits. This is achieved as follows:

A separate S-R flip-flop i.e. CD4013 is set whenever the hotspot comes in front of the first IR detector. The Q output of this flip flop goes high. This is connected to the blanking input of the least significant 7448. All inputs of this decoder/driver are grounded so that when it activates the display, only a '0' is displayed. The Q3 output of the first counter is connected to the 'reset' input of the flip-flop. As soon as the fourth pulse is fed to the counter, i.e. if the time counted is 0.4 sec. or greater, Q3 goes high and resets the flip-flop. This makes the output of the flip-flop go 'low' and the decoder/driver blanks the display as its 'blank' input goes to '0' level.

Every time the hotspot comes in front of the first IR detector, the least significant digit glows for at least 0.4 secs. It continues to glow further if the count in the counter is three or less, otherwise it is blanked. Because of the way in which it is connected in the circuit, glowing of the least significant digit for a short duration also indicates that the hot spot has been sensed by the first IR detector.

Remarks

The design of the prototype for velocity measurement discussed above has the main advantage of simplicity and low cost. This instrument is easier to manufacture and maintain thereafter. As no microprograms or complex hardware is involved trouble- shooting is easier.

A major disadvantage is that the distance between the two IR detectors should be held constant and can not be changed even if the velocity of a different type of object moving at a different range of velocities is to be monitored. Changing the EPROM, for a required distance, every time may not be practical.

This prototype of the velocity measurement system sensed only the leading edge of the signal at the output of the two IR detectors. It exhibited good results while measuring the velocity of the objects having the leading edge. The velocity measurement was not satisfactory when the hotspot detection scheme was implemented, particularly when the hotspot cooled considerably before reaching the second detector.

CHAPTER 3

DESIGN OF THE VELOCITY MEASUREMENT SYSTEM USING BIT-SLICE DEVICES

In the previous design the main disadvantage was that no actual division was performed; instead a lookup table for a fixed distance between the two IR detectors was entered in an EPROM. As a result, even though the optimum distance between the two detectors vary for different objects, moving at different range of velocities, it was not possible to use the measurement system without replacing the EPROM with another one with the desired lookup table.

Actually the velocity measurement system should be capable of being used for various distances between the detectors by simply entering the data about the distance from a key board. To achieve this the instrument should be capable of performing its own arithmetic operations in addition to logical operations. In the present case as the operation involved is division which is the most complex arithmetic operation to implement using a digital system, a more powerful microprocessor has to be used.

In this design the AM-2903 from Advanced Micro Devices is used. It is a 4-bit expandable bipolar microprocessor slice. It is one of the most functionally sophisticated slices currently available. Although the specifications for the AM-2903 do not explicitly say so, the division functions of the AM-2903 were designed for the

floating-point number system. With some additional control logic, a non-restoring type algorithm for the fixed-point division is found to work well with the AM-2903.

Advantages of Using a Bit-slice Microprocessor for Designing the Present System

The design using bit-slice devices is a natural enhancement of the previous design using traditional hardware CMOS ICs, except that the EPROM with a lookup table is replaced by two 4-bit slices of the 2903 microprocessor for performing actual division and a sequencer with micro-storage for controlling the 2903's.

The advantage of using the bit slice microprocessor is that any fixed instruction set microprocessor by definition has its own predefined internal architecture and this constrains the design options available. With bit-slice devices, most of the system architecture is left to the user definition via the selected interconnections and the microprogram.

The 2900 family is expandable in multiples of four bits and so a wide variety of useful word size systems are possible (e.g. in the present system two 2903's are connected together to get an eight bit system). In the future if a more powerful system operating over a larger range of velocities is to be designed, additional slices can be used.

The instruction set that the system under design is to support has a major impact on the choice of implementation. A bit-slice design can be microprogrammed to

support any desired instruction set. In the present design only one instruction i.e. ' Distance=' is used. But if a more sophisticated system with additional instructions like '#samples=' or if readings in a desired range are to be taken then ' range=', instruction can be implemented. This is possible with little change in the hardware. It is possible to have the user's own instruction set suitable for the system. Fixed Instruction Set microprocessors are of little help where such software compatibility is necessary.

SSI/MSI using Schottky TTL and bit-slice (2900 family) can support systems with 125 nsec. cycletimes. MOS microprocessors are slower with approximate cycle times of 1-2 μ sec. The newer bit slice devices are targeted for 100 nsec. microcycle systems.

Basically where critical instruction set and high speed occur, bit-slice devices are very useful. But if design time and parts count restrictions exist or if the production volume does not warrant the efforts required to do a bit slice design, fixed instruction set microprocessors can be used.

A basic flow chart of the operations performed in sequence is shown in figure 3.1. A detailed description of each functional block of the flow chart is given in the later sections.

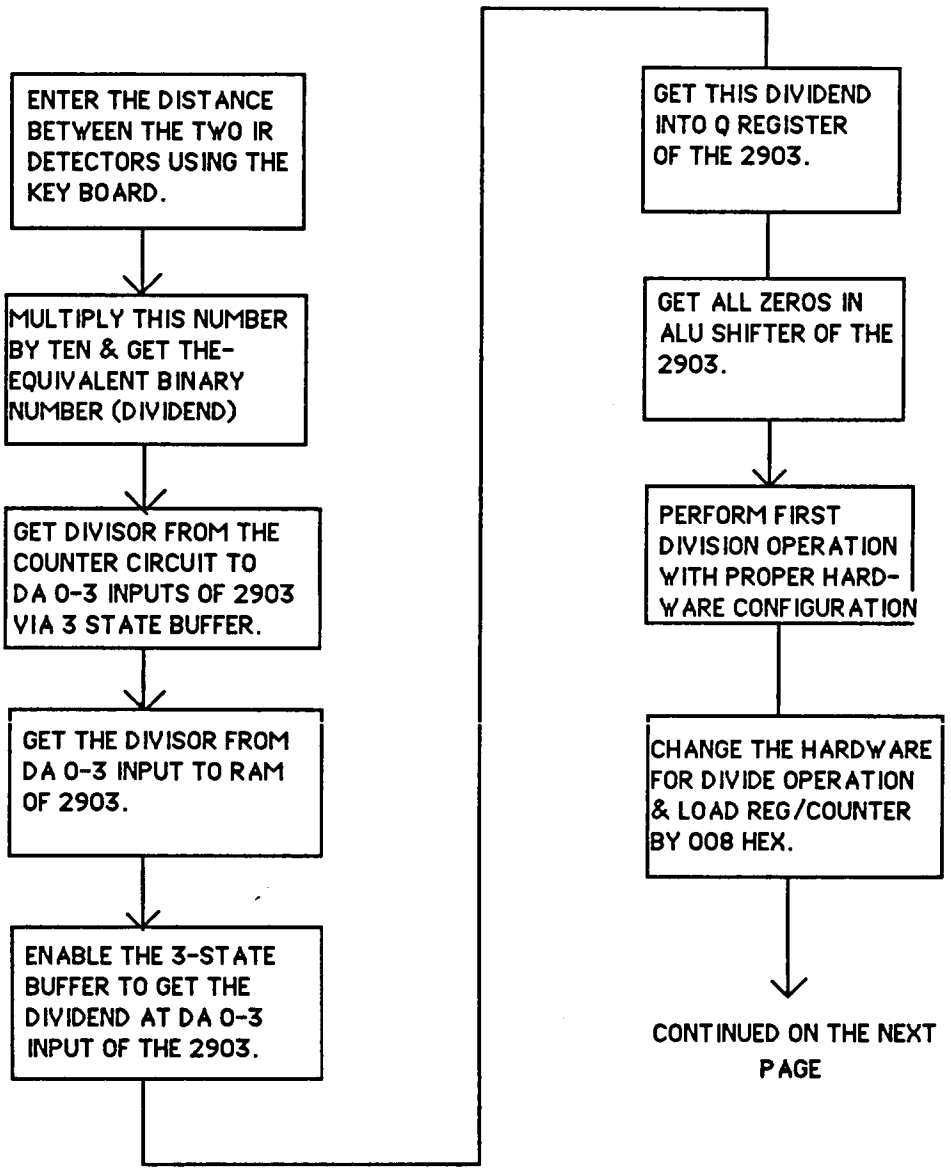
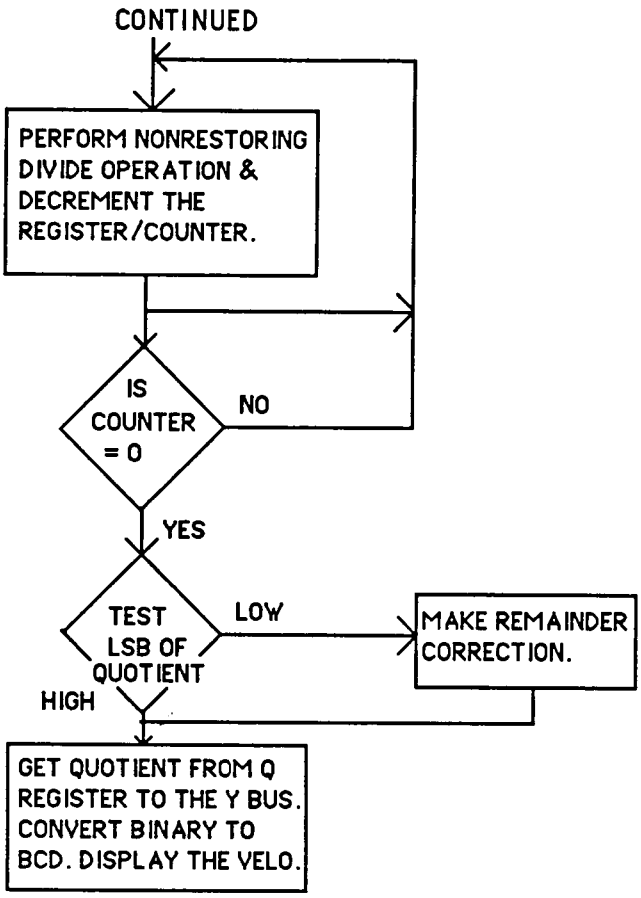


Fig. 3.1 : The flowchart.


CONTINUED
PERFORM NONRESTORING DIVIDE OPERATION & DECREMENT THE REGISTER/COUNTER.
IS COUNTER = 0
NO
YES
TEST LSB OF QUOTIENT
LOW
MAKE REMAINDER CORRECTION.
HIGH
GET QUOTIENT FROM Q REGISTER TO THE Y BUS. CONVERT BINARY TO BCD. DISPLAY THE VELO.

In the following pages design of a relatively simple type of velocity measurement system using bit slice devices is discussed. For simplicity the distance between the two IR detectors which is entered using the key board is in the range of 1 to 25 cms. In actual practice this range can be increased by using a higher number of 2903's. Even with this design, velocities for distances greater than 25 cms. can be measured by simple manipulations e.g. if the actual distance between the two IR detectors is 70 cm. then enter the distance as 7 cm. on the key board instead of 70. To compensate for this, multiply the reading of the velocity by ten. Similarly if the distance between the two detectors is two meters then enter the number 2 on the key board and read the velocity on the display as meters/sec. instead of cms/sec.

A Brief Description of the AM-2903

The AM-2903 is a four-bit expandable bipolar microprocessor slice that performs various logic and arithmetic functions. It contains sixteen internal working registers arranged in a two-address architecture. The Am-2903 provides a special

having numerous applications. The nine-bit micro instruction of the AM-2903 selects the arithmetic logic unit (ALU) sources, function and destination. All data paths within the device are four-bit wide. The device consists of 16 words by 4 bits, a two-port random access memory(RAM) with latches on both output ports, a high performance ALU and shifter, a multi-purpose Q register with shifter input and a nine-bit instruction decoder.

Any two RAM words addressed at A and B address ports can be read simultaneously at respective RAM A and B output ports. EA and OE_B input pins control the output of two ports. ALU shifter output data can be enabled on to the Y I/0 port. With WE, data can be written in to RAM at the B address.

A Brief Description of the Am-2910 Sequencer

The Am 2910 is a bipolar microprogram controller intended for use in high-speed microprocessor applications. It allows addressing of up to 4K words of micro program. It contains a microprogram counter (µpc) that is composed of a twelve-bit incrementer followed by a twelve-bit register. It is used during the execution of sequential microinstructions.

Another source of address is the D input which is used for branching. The register/counter is used as a 12-bit down counter with result = zero available as a micro instruction branch test criterion. It can be used as loop termination counter.

Besides the sources discussed above additional facilities like a stack etc. are available. They are not discussed here as these are not used in the present application.

The Am-2910 is controlled by a 4-bit instruction which is supplied by one field (bits 30 to 33) of the microword format of the system. These four bits provide sixteen basic instructions. Some of the instruction which are used for designing the velocity measurement system are:

(a) Jump Zero (JZ) -- I_{3-0} = 0000 i.e. 00 $_{Hex}$ When the power is made 'on', power-on reset of pipeline register resets the pipeline. So 0 $_{Hex}$ is sent to the AM-2910 which executes JZ. This instruction resets the stack.

(b) Continue (CONT) --I $_{3-0}$ = 1110 --$E_{Hex.}$ Sequential program segments use continue statement. μ-pc register is the source of the next address . The pipeline output PL is enabled.

(C) Conditional Jump PL (CJP)-0011--I $_{3-0.}$ The jump is made to an address in the pipeline register if the CC input is low. If CC is high, the test fails and CJP behaves as a CONT instruction.

(d) Load counter and continue (LDCT)- I $_{3-0}$--1100--C_{Hex} . Whatever is gated at the D_2 input is unconditionally loaded into the register/counter, by the LDCT instruction.

(e) Repeat PL CNTR # 0 (RPCT)-I $_{0-3}$-- 1001--9 $_{Hex}$. This instruction is

used to handle a loop. It is a conditional jump pipeline instruction. The register/counter must be loaded previously via a LDCT. If the counter is not equal to zero, the jump is taken and the counter is decremented if <counter>=0, then RPCT behaves as CONT.

Out of the sixteen available instructions only the above mentioned five instructions are used. Advanced Micro Devices,Inc. manufacture a micro-program assembler called AMDASM which has the capability of using the AM-2910 instructions in the symbolic representation. AMDASM's Am-2910 instruction symbolics or mnemonics are used in the present discussion for simplicity.

A number of architectures are possible using these microprogram sequencers. The normal flow in fetching the micro-instructions is to determine the address of the next micro-instruction, fetch the contents at that address and setup this data at the input of the pipeine register such that it can be clocked into the pipeline register for execution. There are at least seven different architectures that can be defined based on placing the pipeline register in appropriate signal paths and storing data on low to high transition of the clock. The microprogram architecture used in the present design is called the Instruction Based Architecture. Here the pipeline register is placed at the output of the microprogram memory as shown in the circuit diagram.

By definition, the pipeline register contains the micro-instruction currently being executed by the machine. Simultaneously, while this microinstruction is being executed, the address of the next micro instruction is applied to the microprogram memory, and the contents of that memory word are being fetched and setup at the

inputs to the pipeline register. This technique of pipelining is used to improve the performance of the microprogram control unit. This happens because the contents of the microprogram memory word required for the next cycle are being fetched on an overlapping basis with the actual execution of the current microprogram word.

Fuse link bipolar PROM's are the only viable micro-word memory devices (micro -storage) for high speed applications. They are very fast (45 nano-sec. , maximum access time), of small size (512 x 8 in 20 pins) and less expensive than fast RAM's. It is a simple matter to alter or extend the microprogram of the system with fuse-link PROM micro-storage. For a prototype EPROMS can be used but they are very slow.

Discussion of the Circuit Design

Refer to the circuit diagram of the velocity measurement system using bit slice devices shown in figure 3.1. This circuit can be divided into three parts:

(a) Hardware for entering the distance between the two IR detectors. This number multiplied by ten is the dividend. The output of this first section is the dividend in the binary form.

(b) Circuit for amplifying the signals of the two IR detectors and digital hardware for the clock, counters for counting the time and latching the divisor at the output of this section.

dividend in the binary form.

(b) Circuit for amplifying the signals of the two IR detectors and digital hardware for the clock, counters for counting the time and latching the divisor at the output of this section.

(c) Hardware and microprogram controlled bit-slice microprocessor for performing the actual division and control logic for displaying the actual velocity.

Refer to the circuit diagram shown in figure 3.2. The operator is supposed to press the 'Dist=' key. This key is connected in the circuit such that when it is depressed the corresponding D flip-flop 74LS74 is set, thus enabling the output latches 74LS175.

Then when the operator presses any of the 10 keys numbered from 0 to 9, a corresponding binary code appears at the output of the 10 lines to 4 lines encoder viz. 74LS147. Simultaneously a rising edge at the output of the 'OR' gate triggers the mono 74LS122 and resets another 74LS74 (#4) as shown in the figure. The Q output of IC4 i.e. 'A' enables the input of the quad D Latch i.e. IC7, and after the timing of the mono is complete, it gives a pulse which latches the data at the output of 74LS174 (IC2) into IC7.

Note that the output pulse of the monostable resets the RS flip-flop (IC4) thereby making Q high and Q low to enable the second latch (IC9) for the least significant digit. The high level at Q output is held for some time by an R-C delay circuit so that the first latch (IC7) remains enabled until the monostable output pulse latches the data.

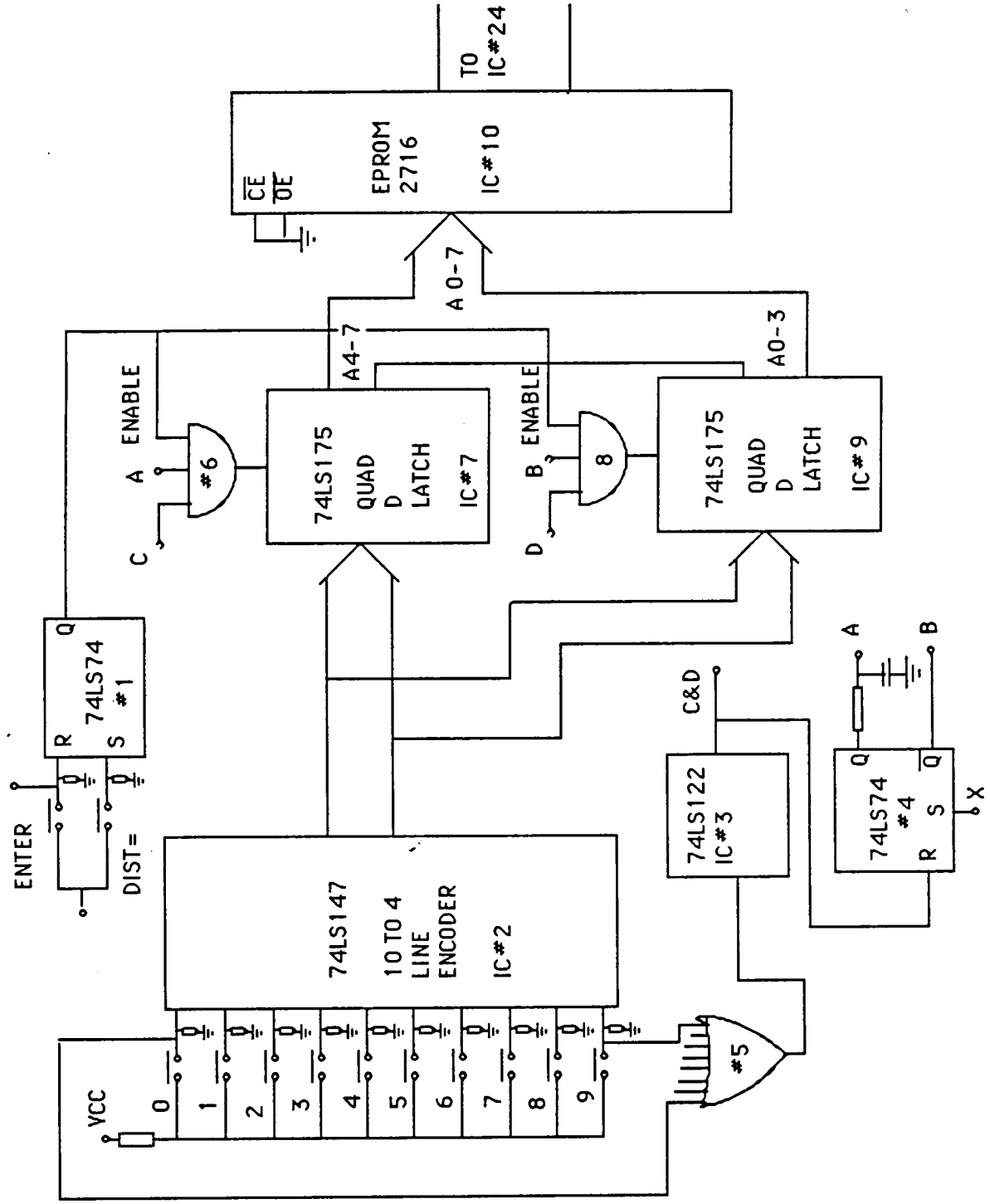
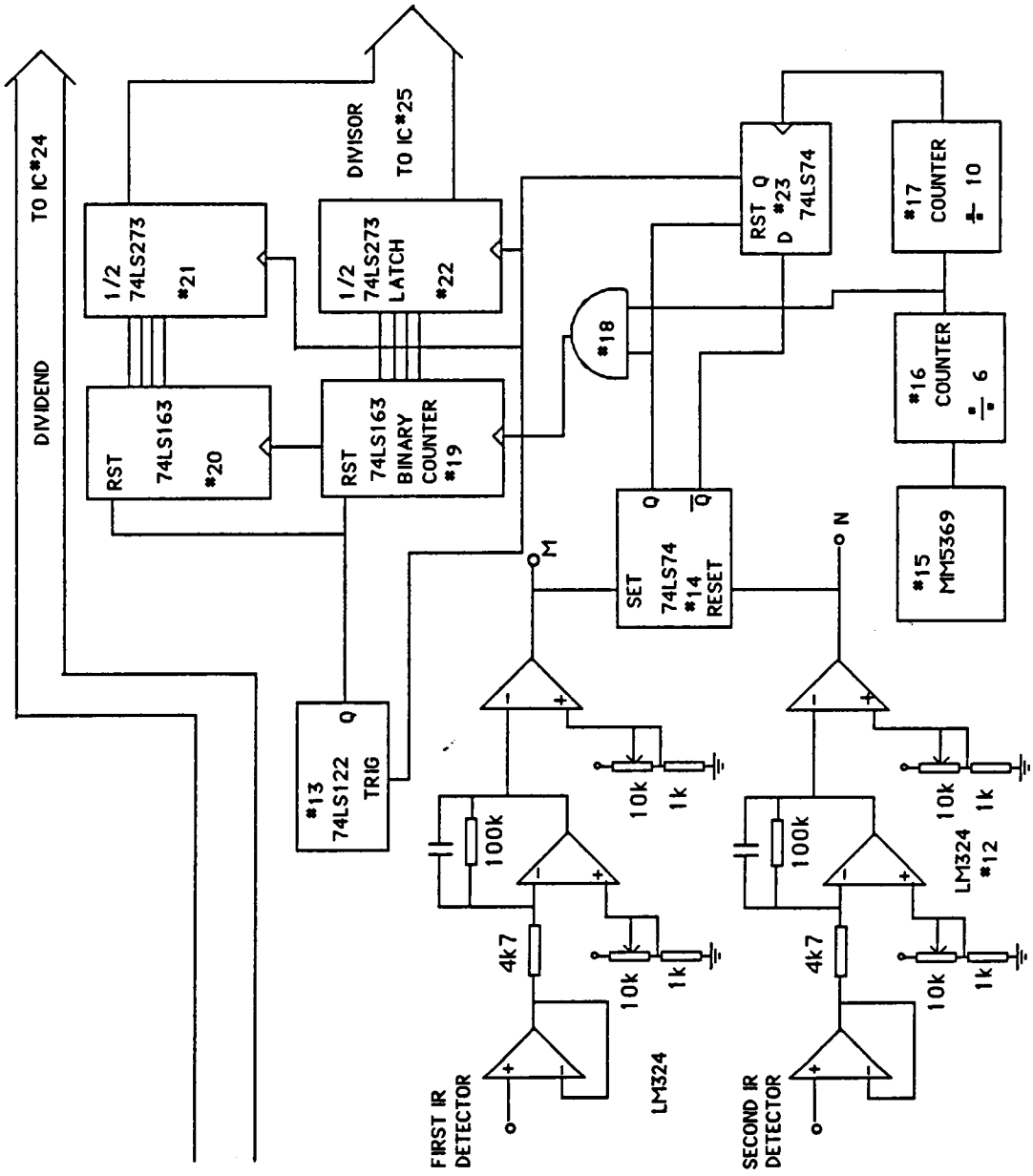
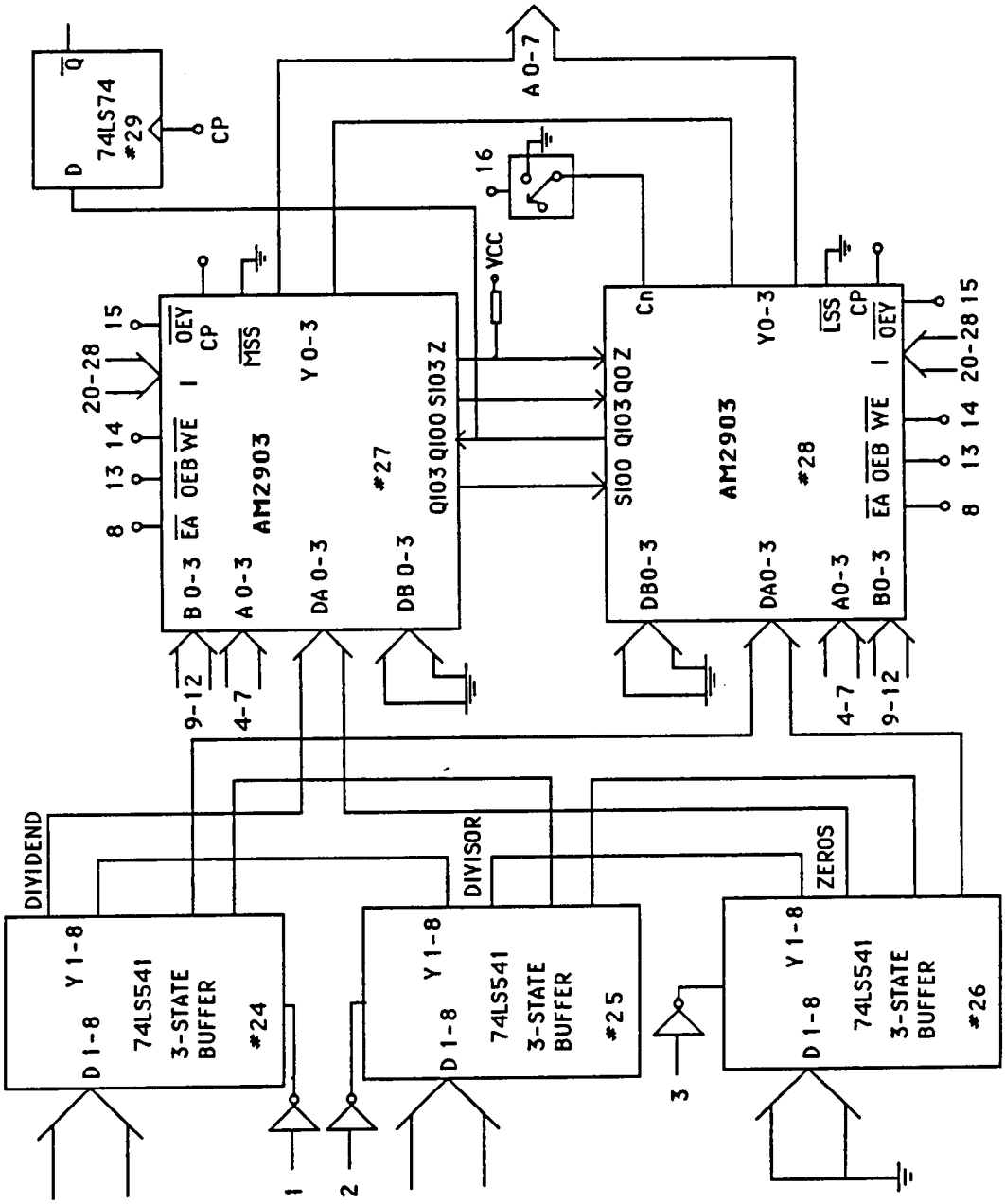
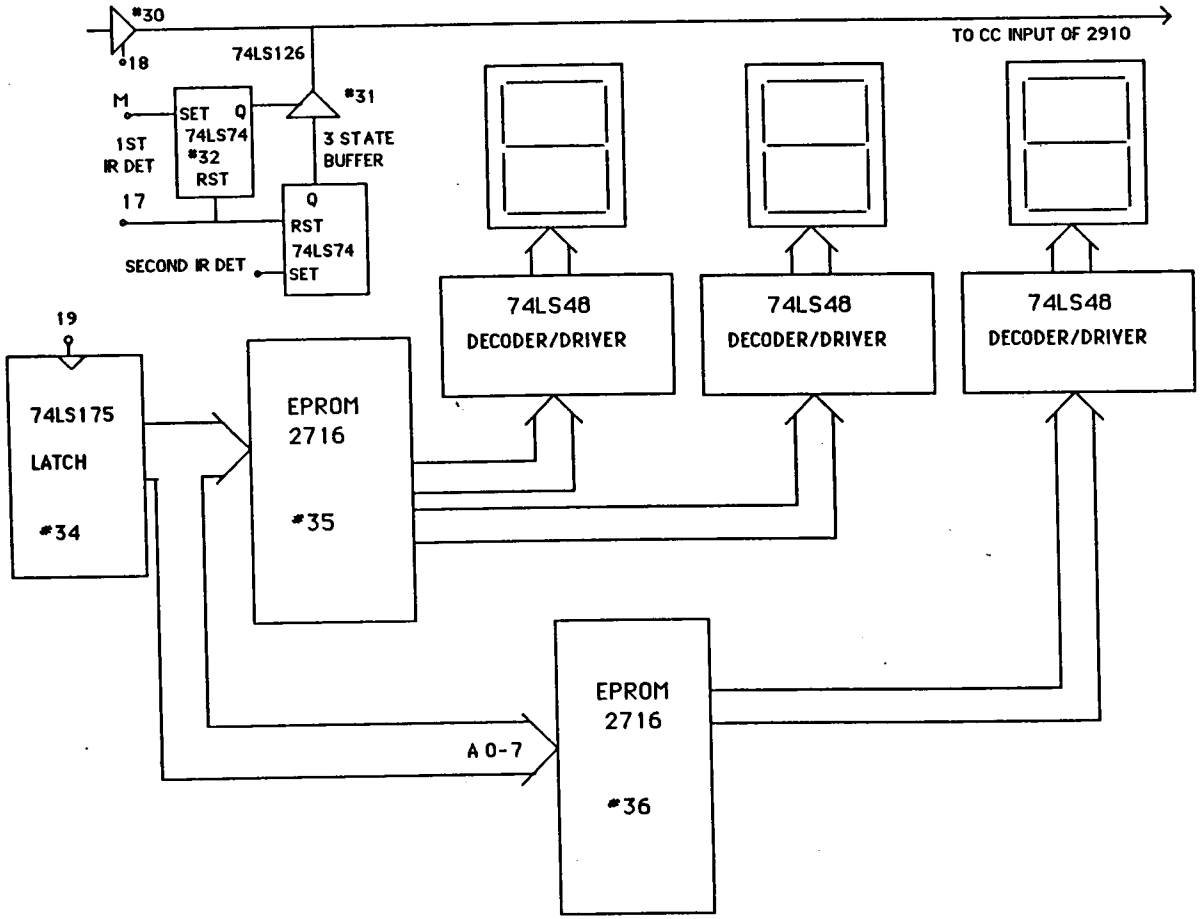
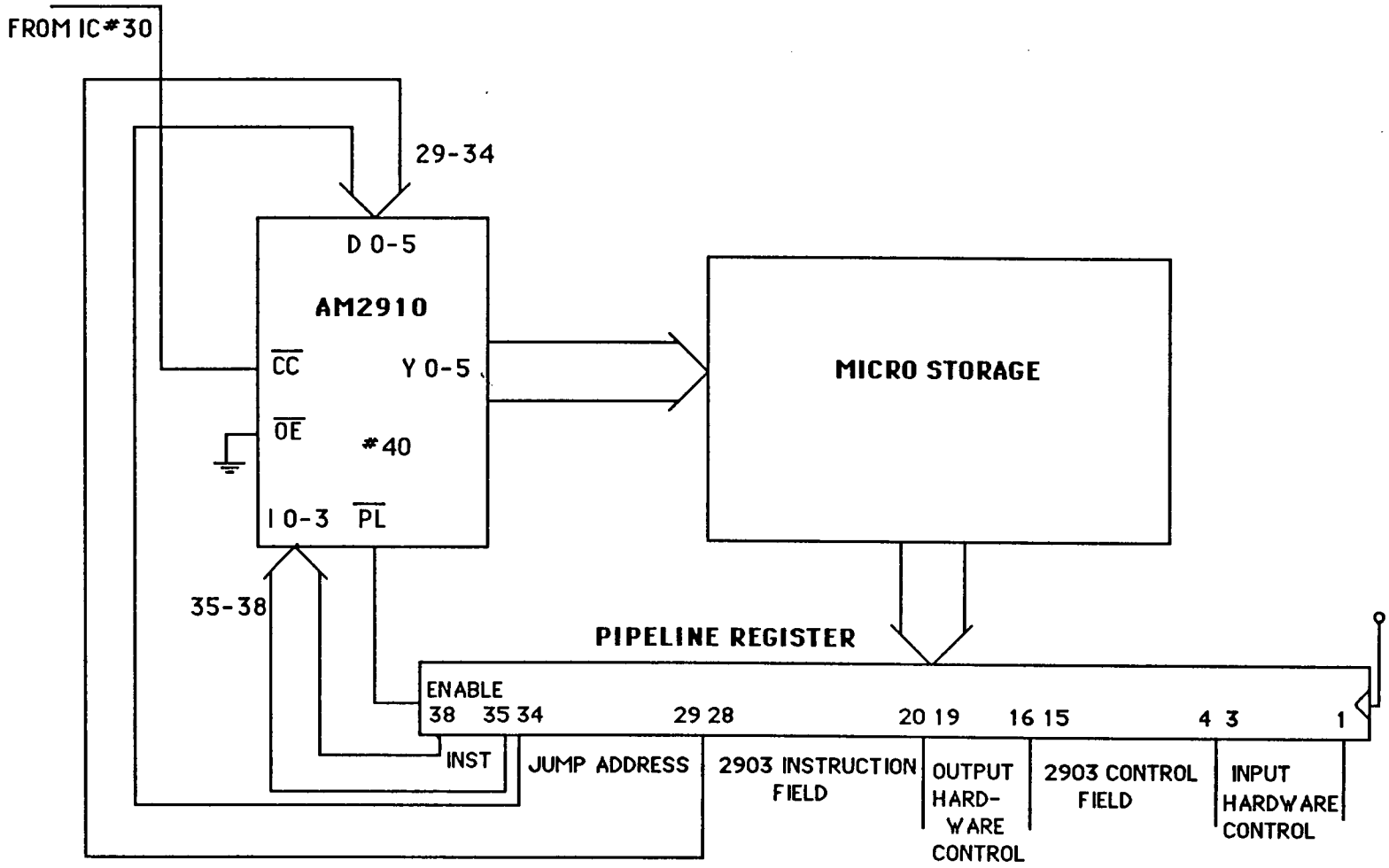


Fig. 3.2: Circuit diagram of the bit-slice design.


DIVIDEND
TO IC*24
1/2
74LS273
*21
DIVISOR
TO IC*25
1/2
74LS273
LATCH
*22
RST
74LS163
*20
RST
74LS163
BINARY
COUNTER
*19
*18
RST Q
D *23
74LS74
*17
COUNTER
÷ 10
*16
COUNTER
÷ 6
*15
MM5369
SET
74LS74
*14
RESET
Q
Q
M
N
*13
74LS122
TRIG
Q
100k
4k7
10k
1k
FIRST IR
DETECTOR
LM324
LM324
*12
SECOND IR
DETECTOR


74LS74
#29
D
Q
CP
A 0-7
16
VCC
AM2903
#27
AM2903
#28
20-28
15
13
14
8
OEY
CP
I
WE
OEB
EA
MSS
LSS
Y 0-3
Cn
A 0-3
B 0-3
DA 0-3
DB 0-3
Q103 Q100 S103 Z
S100 Q103 Q0 Z
DB0-3
DA0-3
AO-3
BO-3
9-12
4-7
DIVIDEND
DIVISOR
ZEROS
74LS541
3-STATE
BUFFER
#24
#25
#26
D 1-8
Y 1-8
1
2
3


#30
#18
74LS126
TO CC INPUT OF 2910
M
SET Q
74LS74
#32
RST
1ST
IR DET
#31
3 STATE
BUFFER
17
Q
RST
74LS74
SET
SECOND IR DET
74LS48
DECODER/DRIVER
74LS48
DECODER/DRIVER
74LS48
DECODER/DRIVER
19
74LS175
LATCH
#34
EPROM
2716
#35
EPROM
2716
#36
A 0-7


FROM IC#30
29-34
D 0-5
AM2910
CC
Y 0-5
OE
#40
I 0-3
PL
MICRO STORAGE
35-38
PIPELINE REGISTER
ENABLE
38
35 34
29 28
20 19
16 15
4 3
1
INST
JUMP ADDRESS
2903 INSTRUCTION FIELD
OUTPUT HARD-WARE CONTROL
2903 CONTROL FIELD
INPUT HARDWARE CONTROL

When the operator presses another key to enter the least significant digit of the distance, the monostable is again triggered by the pulse at the output of OR gate IC4 i.e. the RS flip-flop is already reset so Q is high and the least significant latch, IC9, is already enabled. The output pulse of the monostable latches the binary code for the key pressed into this latch. Then the operator presses the 'Enter' key. This key, as shown in the circuit, resets the RS flip-flop, IC1, thereby disabling both the latches. No further data will be entered into latches if any number key is accidently pressed. Enter key also sets the RS flip-flop, IC4 so that further data, if required, can be entered in to the most significant latch i.e. IC7.

The binary numbers at the output of both the latches act as an address to the EPROM (IC10). The actual data stored in a particular memory location is the binary code for the address multiplied by 10. So if the distance entered by the operator is 15 then the actual output of EPROM will be the binary equivalent of 150.

This multiplication by 10 is done to avoid division of fractions. The measurement of time is done with a resolution of 0.1 secs. So the divisor which is the time measured, is not an integer but in the form like 2.3 or 0.4 etc. Multiplying both dividend and divisor by 10 give integers.

The (b) part of the circuit, i.e. the one which gives the divisor is similar to the hardware design discussed in the first chapter, so it is discussed very briefly here.

The signal from the IR detectors are amplified and the corrosponding digital signal sets or resets the flip-flop (IC 14). The output of this flip flop controls the

input to the binary counters which is fed from a 0.1 Hz clock. A signal from the second IR detector resets the RS flip-flop (IC14). Therefore input to the counters is disabled. The counters hold the binary count for the time required by the hot spot to travel from the first to the second IR detector. As the first RS flip-flop is reset by the second IR detector signal, its Q output goes high which makes the D input of second RS flip-flop (IC23) high. The next positive edge of the 1-Hz clock makes the output of this RS flip-flop (IC23) go high which latches the data from the binary counter and via a monostable resets the counters. These latches hold the divisor.

The (c) part of the circuit comprises two 2903's which form an eight-bit processor, the sequencer 2910, the microstorage and associated hardware for displaying the results. This part of the circuit is mainly controlled by the microprogram in the microstorage. In the circuit diagram the control points controlled by the microprogram are numbered in green .

A microword can be divided into 5 fields in the present design.

Bits 1--3 . Input hardware control:

These bits when in the high state enable the corrosponding three state buffer. One of these three bits is in a 'high' state enabling either dividend, divisor or all zeros and feeding this data to the DA0-3 input of the 2903.

Bits 4--15 . 2903 Control field.

Bits 4-7 select the register in the RAM whose data should be output on 'A' port or R input of the ALU

Bit 8 specifies whether the input to R port of ALU shall be from A or DA bus.

Bits 9-12 again select a register in the array (RAM) to be designated as B register i.e. the contents of this register are presented to the B port.

Bit 13 controls OEB pins of the 2903s and this pin designate the input to the ALU 'S' port (B,Q or th DB bus)

Bit 14 controls WE. If WE is 0, it causes data on the Y bus to be gated into the B register (when the clock input is low). The WE of the most significant slice (MSS)should be connected to the WRITE output of the least significant slice(LSS).

Bits 16--19, Output hardware control:

Bit 16 when '0' configures the 2903's for the first divide operation by connecting Cn of LSS to ground. If it is '1' the configuration is good for normal non-restoring division.

Bit 18 control a hardware which checks if the LSB of the quotient is 0 or 1 and accordingly the 2910 decides if the remainder is to be corrected or not. Bit 17 controls the 3- state input to CC of the AM-2910 which uses it to decide if program execution is to be started or not.

Bit 19 latches the quotient which after necessary code conversion is displayed as the velocity.

Bits 20--28, These control the nine I lines of the ALU:

If I_0 to I_5 are all Os, pins I_5 to I_8 control special functions of the 2903 like a 2's complement divide etc.

I_5--I_8 control routing of data (ALU and shifter outputs). These pins control the operation of both shifters (arithmetic as well as logical). Pins I_1 to I_4 control various logical and arithmetic operations like AND, OR, Exclusive OR as well as addition, subtraction, etc.

Bits 29--34, Jump address:

These bits of the microword are fed to the D_2 input of the 2910 sequencer. When a conditional jump PL (CJP) instruction is executed by the sequencer, the program starts executing from the address present at the Di input.

Bits 35--38 :

These 4 bits control the operation of the sequencer by 16 instructions discussed earlier.

Description of the Microprogram Control

When power is switched 'ON', the power-on reset circuitry of the pipeline register resets it. As all bits at the output of PL are now '0', the I_{0-3} input of the 2910 sequencer is 0000 which is the code for JZ, i.e. jump to zero instruction. This instruction resets the stack pointer.

All instructions cause the next address value to be switched and to be incremented by the incrementer.

The next micro-instruction has in the 2910 instruction field (bits 35-38) code 1100 i.e. C_{Hex} which is the LDCT instruction for 2910. In the jump address field (bits 29-34) the value 8_{Hex} is present. Therefore the register/counter gets loaded by this value. The purpose of this is to keep the register/counter ready for performing the division loop eight times. LDCT behaves as a continue statement besides loading the register/counter and therefore the incrementer in the 2910 increments the counter.

The third micro instruction is CJP

```
I CJP I 000000 I xx---0 0 xxx---x--1 I
38  35        29      18 17          0
```

So the instruction field for the 2910 has the code 0011 which stands for the conditional jump PL (CJP). If CC of the 2910 is low , a jump is made to the address in the jump field of the PL register. Now the hardware is arranged such that when the hot spot comes in front of the first detector, the 3-state buffer, IC31, is enabled as the M input of IC32 is high. Now it is desired that the program should not be executed further until the hot spot comes in front of the second detector after which we will get the dividend and the divisor ready at the input for further processing. So as long as the hot spot is not detected by the second IR detector the CC input of the 2910 remains low and as the address in the jump address field of the PL register is 0000 the program

remains in the initial loop and does not procced.

After the hot spot is detected by the second IR detector the CC input of the 2910 goes high. The CJP test fails and this instruction acts as a CONT instruction and the incrementer increments and the micro-PC register is the source of the next address.

The next instruction of the micro-program is CONT

CONT	XXXX---XXXI	XX----XXX	I---1	1----I	X-----1--I
38 35	29	20	18 17	15	2

The instruction to the sequencer is CONTINUE which is a sequential program statement. So the micro-PC is the source of the next address. Bit 17 is made '1' which resets the flip-flop IC's 32 and 33, and the 3-state buffer IC 32 is disabled. Also bit 18 is made high enabling the 3-state buffer IC 30, thereby connecting Q of IC 29 to CC of the 2910. Bit 2 is made high so that the divisor (time counted) is presented at the DA_{0-3} input of the 2903.

The next task is to get the divisor into Ro register of the RAM. For achieving this, in the first microinstruction get the divisor at DA_{0-3} input to F_{0-3} output of ALU. This is done by making EA high so that the source of the 'R' input of the ALU is DA_{0-3}. Then the source of the 'S' input of the ALU is selected as DB_{0-3} which is all 0s, by making OE_B 'high' and Io 'low'. So at the R input of the ALU we have the divisor and at S input we have all O's. This instruction is represented as shown in the figure.

```
I   CONT   I-----XX-----I-----0----1    1----1----1----1--I
                                  20   18 17  13    8     2
                                              OEB   EA
```

In the next microinstruction R EOR S i.e. exclusive-OR operation is performed so that the divisor at the R appears at the F_1 output of the ALU unchanged. For R EOR S code on I_4-I_1 i.e. bits 21-24, should be HLHH which is $B_{Hex.}$

Also the divisor on the F bus has to be moved to the Y bus. So let the code on I_8-I_5 be HHLL i.e. C_{Hex} which moves F --> Y and Q register is hold. Then as this divisor is to be moved to Ro of RAM, the code 0000 should appear on B_{0-3} input of the 2903 i.e. bits 9 to 12 of the micro-word should be 0000.

In the next micro-instruction make WE of LSS low so that the divisor which has been brought to the Y bus will be stored in the register whose address is present at B_{0-3} i.e. 0000 in our case. Therefore the divisor is stored in the Ro.

The next task is to get the divisor into the Q register. Make bit 1 high which will feed the dividend to DA_{0-3} of the 2903 via the 3 state buffer IC24. Again perform R EOR S i.e. exclusive-OR operation of the dividend with all zeros to get the dividend on the F bus of the ALU. For R EOR S use code 1011 on $I_{4-1.}$ Now the dividend has to be stored in the Q register, so use the code 0110 on I_{8-5} or bits

28-25 which performs F --> Q.

To get all the O's in the ALU shifter, make bit 3 high so that all the inputs at DA_{0-3} and DA_{0-3} are O's. Perform R EOR S to get the O's in F and the ALU shifter. In this way divisor is obtained at Ro, dividend at Q register and ALU shifter is cleared.

The next step is to perform the first divide operation. The first divide operation consists of the following steps.

F <-- B+Cn

B <-- 2F Q <-- 2Q

The hardware connections for the first divide operation are shown in figure 3.3.

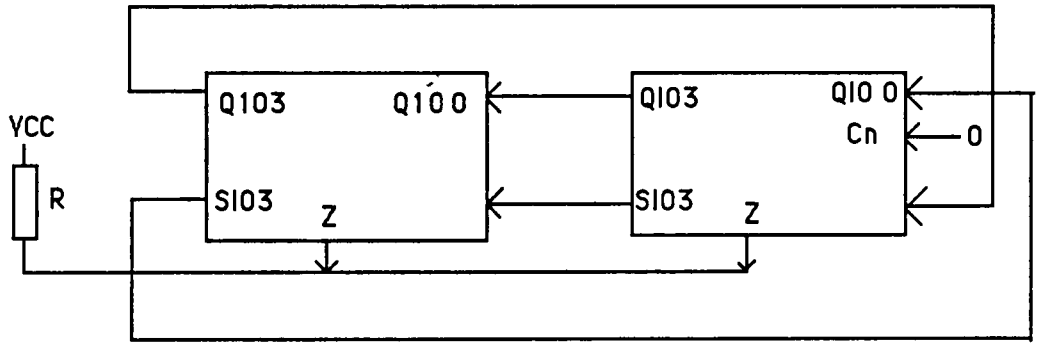


Fig. 3.3 : Hardware connections for the first divide operation.

For this configuration make bit 16 'Low' so that the Cn input of the least significant 2903 is connected to the ground. Make bits 4 to 7, i.e. address at $A_{0-3,}$ 0000 so that Ro which contains the divisor comes to R input of the ALU. Make EA low so that the divisor is present at the 'R' input. Keep bit 13 i.e. OE_B high so that on the 'S' input of the ALU we get all zeros.

To perform F <-- S+Cn use 4_{Hex} or 0100 on I_{4321} (bits 24-21) of the 2903. Simultaneously on the bits 12-9 i.e. B_{0-3} get the code 0001 so that R_1 of the RAM is addressed. In the same micro-instruction to perform 2F use B_{Hex} or 1011 on I_{8765} , i.e. bits 28-25 which perform Log 2F --> y and Log 2Q --> Q.

In the next microinstruction make WE low so that contents on the Y bus will be written into R_1; keep the code on B_{0-3} the same so that the contents of R_1 always appear at S. This concludes the 'first divide' operation.

Then the divide instruction of the 2903 has to be executed 8 times. For division which is an extended instruction of the 2903, the hardware contiguration is as follows.

The extended divide instruction does the following operations

F <-- B+A+Cn if Z=0

F <-- B-A+Cn if Z=1

B <-- 2F Q<--2Q

Note that to perform this division eight times, the register/counter is loaded by 8_{Hex} in the initial micro-instruction. The LDCT sequencer instruction is used. Refer to figure 3.4.

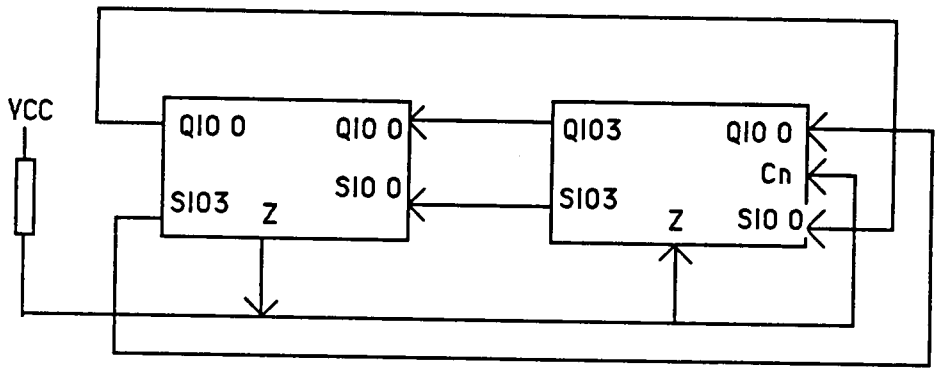


Fig. 3.4 : Hardware configuration for the divide instruction.

To perform division, set bits 20 to 24 to 00000. i.e. I_{0-4} are all zeros. Under this condition I_{8-5} decide the extended operation. For the divide operation use code 1100 i.e. C_{Hex} on I_{8-5} i.e. bits 28 to 25. After the division is performed, in the next micro instruction use the RPCT instruction for the 2910 and puts the address of

the divide instruction in the jump address field. RPCT is for handling the loop with the start address of the loop in the jump address field of the pipeline register. The register/counter is decremented after each jump is taken. If the counter is not equal to 0 the jump is taken; otherwise RPCT behaves as a CONT instruction.

After the division instruction is performed eight times, we get the quotient in the Q register and the remainder in the ALU shifter. How exactly this happens is described later using an example of the non-restoring division algorithm.

If the least significant bit of the quotient is zero, then the remainder needs to be corrected by adding the divisor else no correction is needed. For this some additional hardware is used. At every rising edge of the clock, IC 29 latches the status of Qo. The Q output of this IC presents the opposite state of Qo to CC of the 2910 via a 3-state buffer IC 30. Therefore if Qo i.e. LSB of quotient is 0, the CC input of the 2910 is high.

Therefore after the division is performed, the next microinstruction uses the CJP instruction for the 2910. If CC is high this acts as a CONT instruction. Therefore the micro- instructions following the above mentioned instruction to the remainder correction by adding divisor on the R input of the 2903 to the contents of the R_1 on S input of the 2903. This is done by using the code 3_{Hex} or 0011 on $I_{4-1,}$ i.e. bits 24 to 21. To get F --> Y use 1100 i.e. C_{Hex} on I_{8-5} (bits 28 to 25). Note that though the correct remainder is obtained for further simplicity of the hardware, it is not used for the final velocity reading.

The final part of the micro-program is for displaying the quotient. First to get the quotient from the Q register to the F_{0-3} bus make Io i.e. 20 high and OE_B i.e. 13 also high so that the S input of the ALU is connected to the Q register. In addition, make 3 high and 1 and 2 low so that all zeros are present at DA_{0-3}. Make EA i.e. 8 high so that we get zeros at the R input of the ALU and the quotient at Q into the S input. To get the quotient at the output, in the next micro- instruction perform R EOR S and F --> Y using 1011 i.e. B_{Hex} on I_{4-1} or bits 24 to 21 and 1100 on I_{8-5} or bits 28 to 25 so that the quotient is present at the input of the latch (IC 34)

In the next micro-instruction make 19 high so that the quotient is latched into the 74LS175 latch. The quotient is an eight-bit binary number. Two EPROM's are used with a lookup table to convert binary to BCD and 74LS48 decoder/driver display the actual velocity. The last micro-instruction is jump to zero and all the controls used in the program are reset.

The actual non restoring division is performed as follows:

For simplicity a five-bit divisor and a five-bit dividend is considered in figure 3.5. example. The same algorithm also holds true for an eight-bit operation.

Divisor (in A port). A = 00101 ; -A = 11011 ; Dividend = 01101 in Q register.

OPERATION	ALU OPERATION	F BUS B+A	B 2F--B	Q	SCFF F3 R3	Z ON NEXT CYCLE
START	X	X	00000	01101	X	X
1stDiv.	S+Cn	00000	00000	11010	0	1
C (Hex.)	B-A	11011	10111	10100	1	0
C (Hex.)	B+A	11100	11001	01000	1	0
C (Hex)	B+A	11110	11100	10000	1	0
C (Hex.)	B+A	00001	00011	00001	0	1
C (Hex.)	B-A	11110	11100	00010	1	0
RESTORE	B+A	00011	F--B	00010	X	X

Fig. 3.5: Non-restoring division.

Note: SCFF is the sign correction flip flop of 2903

SCFF<--F_3 EOR R_3, Z on next cycle is F_3 EOR R_3

If Z = 1 perform B-A. If Z = 0 perform B+A.

Then shift F and Q left. Transfer shifted F to B.

CHAPTER 4

MICROPROCESSOR-BASED VELOCITY MEASUREMENT SYSTEM

A velocity measurement system with the following features was found desirable at the design stage.

(a) Time measurement (for measuring the elapsed time between the instants when the hot spot is detected by the two IR detectors) with a resolution of 10 milli-seconds or better.

(b) Distance between the two IR detectors should be operator selectable and the velocity measurement system should make all the necessary computations by using the data regarding the distance, fed in by the operator.

(c) The system should be capable of giving the correct velocity irrespective of the dissimilarity in the output signals of the two IR detectors, which is caused due to the conduction of the heat from the hot spot.

(d) The system should be flexible enough to incorporate the new design ideas, to improve the system performance.

All the above mentioned tasks were sucessfully achieved by using a Motorola MC68000 microprocessor based system. It is a 16-bit microprocessor and has a very powerful instruction set. Besides its capabilities one of the major reasons for selecting this microprocessor was, the availability of a

MEX68KECB/D2 microcomputer board and a HP64000 development system which has a powerful editor,assembler, linker, emulator, etc.

A Brief Description of the MC68000

MC68000 utilizes VLSI technology and is a fully implemented 16-bit microprocessor with 32-bit registers, a rich basic instruction set and versatile addressing modes.

It has a 24-bit address bus and a 16-bit data bus. It contains seventeen 32-bit data and address registers and has fifty-six powerful instruction types. It has fourteen addressing modes and uses memory-mapped input /output. It also has a 32-bit program counter and a 16-bit status register.

The first eight registers DO-D7 are used as data registers for byte, word (16-bit) and long word (32-bit) operations. The second set of seven registers (A0-A6) and the system stack pointer may be used as software stack pointers and base address registers. All of the seventeen registers may be used as the index registers.

The status register contains interrupt mask as well as condition codes viz. extended (x), negative (N), Zero (Z), overflow (V) and carry (C). Also status bits, indicating that the processor is in a trace (T) mode and in a supervisor (S) mode, are available.

Five basic data types viz. Bits, BCD digits (4-bits) Bytes (8-bits), Words (16 bits) and Long Words (32 bits) are available.

The basic address modes Register Direct, Register Indirect, Absolute, Program Counter Relative, Immediate and Implied are provided.

The instruction set of the MC 68000 can be found in any instruction manual. The instructions for multiplication and division were very useful in the development of software for the present application. Instructions are available for data movement operations, integer arithmetic operations, logical operations, shift and rotate operations, bit manipulation, BCD operations, program control operations, system control operations, etc.

A Brief Description of the Parallel Interface/Timer (PI/T)

The MC 68230 Parallel Interface/Timer (PI/T) provides versatile buffered parallel interfaces and a system-oriented timer for MC 68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes either 8- or 16-bits wide. In the unidirectional mode, an associated data direction register determines whether each port pin is an input or output. In the bidirectional mode four handshake pins are used. The PI/T ports allow use of vectored or auto-vectored interrupts and DMA requests. The PI/T timer contains a 24-bit wide counter and a 5-bit prescalar. The timer may be clocked by the system clock (PI/T CLK pin) or by an external clock (Tin pin) and a 5-bit prescalar can be used. PI/T is used in the present design, to measure the elapsed time.

The main features of the PI/T are

(1) M 68000 BUS compatible

(2) Port modes include- Bit Input/Output (I/O)

- Unidirectional 8-bit and 16-bit.
- Bidirectional 8-bit and 16-bit.

(3) 24-bit programmable Timer Modes.

(4) Registers are Read/Write and directly addressable.

(5) Registers are addressed for MOVEP (Move Peripheral) compatibility

The PI/T consists of two logically independent sections: the ports and the timer.

The port section consists of port A (PA 0--PA 7) , port B (PB 0--PB 7), four hand shake pins (H_1, H_2, H_3, and H_4), two general I/O pins and six dual function pins.

The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescalar. Three pins provide complete timer I/O : PC2/Tin, PC3/Tout and PC7/TIACK. Only the ones needed for the given configuration perform the timer function, while the others remain port C I/O.

The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (DO-D7).

Port modes of the PI/T

The primary focus in the present application is on port A, port B, and the handshake pins. These are controlled in the following way:

The port general control register contains a 2-bit field that specifies one of the four operation modes. These govern the overall operation of the ports and determine their inter-relationships. For each mode there are some submodes which are decided by a 2-bit submode field in the port control register. Each of the ports i.e. A, B and C have their own port control registers.

In the present design, Mode 0 (unidirectional 8 bit mode) is used in conjunction with the submode 1X; for both the ports i.e. port A and port B. Submode 1X gives pin-definable Single Buffered Output or Non-Latched Input, and the handshake pins H_1, H_2 and H_3, H_4 act as the status/ interrupt generating inputs.

The detail description of various registers in PI/T and their functions is given in the later sections while explaining the software. A brief description of each is given here.

The port A and port B control registers decide the submode by means of bits 7 and 6. Bits 5, 4 and 3 control operation of H_2 (H_4) handshake pin and bit 2 determines if an interrupt will be generated when the H2S (H4S) status bit goes to one. Bit 1 and 0 are for H_1 (H_3) and H1S (H3S) control.

Port A and B data direction registers determine the direction of each of the port A and B pins. A zero indicates that the pin is used as an input and a one level indicates that it is used as an output. Port A and B data registers are holding registers for moving data to and from port A and B pins. Data direction registers determine whether each pin is an input (zero) or an output (one) and are used in

configuring the actual data paths.

Port status register contains information about handshake pin activity. Bits 7--4 show the instantaneous level of the respective handshake pin. Bits 3--0 are corrosponding status bits which are very frequently used in the software design of this velocity measurement system.

The timer control register determines all the operations of the timer. Bits 7--5 configure the PC3/TOUT and PC7/TIACK pins for port C, square wave, vectored interrupt, etc. Bit 4 specifies whether the counter receives data from the counter preload register or continues counting when zero detect is reached. Bit 3 is unused, bits 2 and 1 configure the path from CLK and Tin pins to the counter controller. Bit 0 enables the counter.

The counter preload registers are a group of three 8-bit registers used for storing data to be transferred to the counter. In the present case these registers are loaded with 1s. The count register (high, medium and low) is a group of three registers at which the counter can be read.

Besides the registers described above there are some other registers available in PI/T, but as these are not used in the design, they are not mentioned. The velocity measurement system can be divided into three parts.

(a) The MEX 68KECB/D2 microcomputer board (with MC68000, MC68230 chips and associated hardware already assembled)

(b) External hardware (Two IR detectors, amplifiers, comparators, a 100 Hz. clock as well as latches and driver/decoders for displaying the velocity .)

(c) The power supply.

For convenience, a description of the external hardware is given first.

External Hardware For The Velocity Measurement System

The microprocessor interacts with the external circuit via MC68230 i.e. Parallel Interface /Timer chip. As explained before it has ports A and B. Port A is used as an output where as Port B is used as input.

So, port B which has eight input pins, is used for getting the data regarding the distance between the two IR detectors. For simplicity it is assumed that this 8-bit data is in the binary form and is always present at port B. In reality, if inputting data in binary form poses any problem then a set of three BCD type thumb wheel switches and a BCD to binary lookup table can be provided which will enable the operator to input a decimal number for the distance while at port B an EPROM with a lookup table can present a crresponding binary number.

Port A is used as an output and it outputs the data to be displayed. Now, port A also has only eight pins which therefore at an instant can output only eight bits. But we have to display a four digit velocity, and for outputting four BCD numbers a port with 16-bits will be required. This problem is solved by multiplexing. For holding a BCD number a CD 40174, four-bit latch is provided for each display. This latch latches the data present at its D3-D6 inputs on the rising edge of the clock input. On this rising edge of the clock , data present at the D inputs appears at the Q outputs.

The lower four bits PA0--PA3 of port A, always output a BCD number.

Pins PA4--PA7 are connected to the clock inputs of the four latches. Depending on which of the four displays should be presented with the BCD number on PA0--PA3, one of the four clock or strobe pins (PA4--PA7) will go high for a while, thereby latching the data in the desired latch. Note that the BCD number is fed simultaneousely to all the latches. But this number gets latched into only one of the four latches.

After the microprocessor under the control of the software program, latches the desired BCD count in each of the four latches, the 74LS48 BCD to seven -segment decoders/drivers illuminate the desired segments of the seven-segment displays. The hardware for this is shown in figure 4.2.

The internal registers of the PI/T MC68230 are programmed such that a negative edge on the handshake pins H_1,H_2, H_3 and H_4 set the corrosponding bits H1S, H2S, H3S, and H4S in the port status register. The microprocessor makes extensive use of the status of these status bits and the internal timer measures various time intervals. H1, H2, H3 and H4 pins are asserted at a predetermined voltage level on the leading and the trailing edges of the output signal of the two IR detectors. Hardware configuration is shown in figures 4.1 and 4.3.

From the circuit diagram in figure 4.1 it can be seen that the output of the IR detectors is first fed to an inverter which offers a high input impedance.

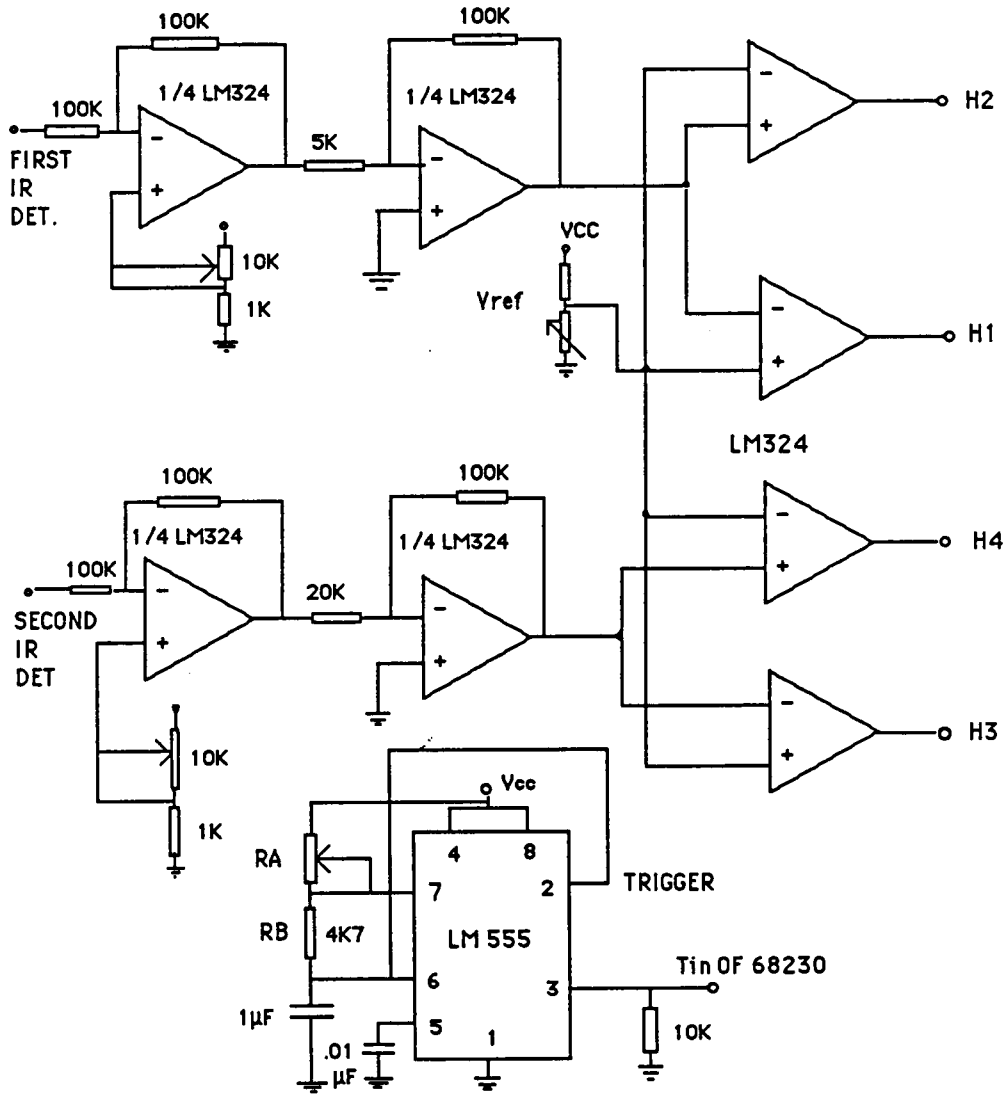


Fig. 4.1 : Threshold detection circuit and 100 Hz. clock

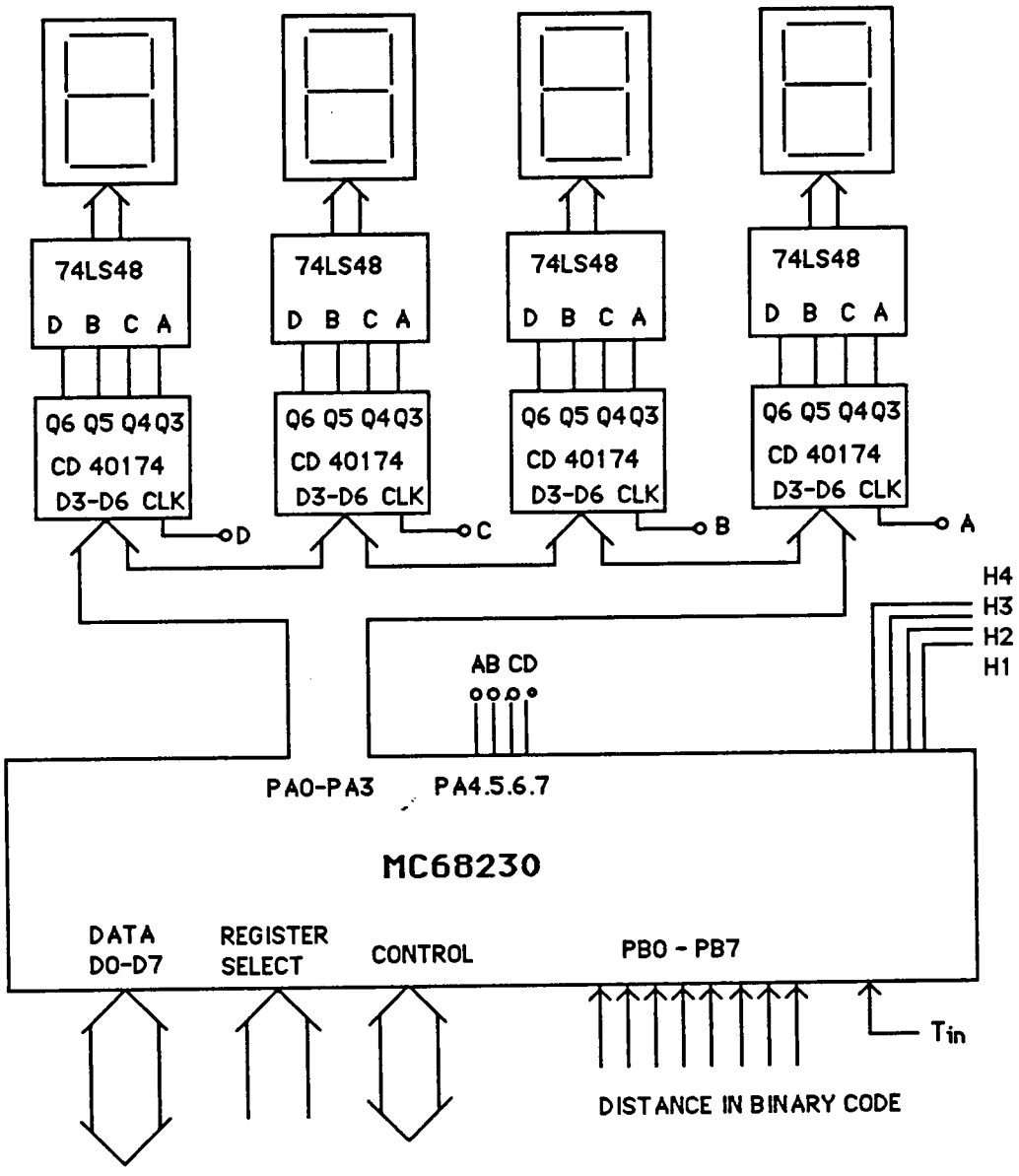


Fig. 4.2 : External hardware for velocity display.

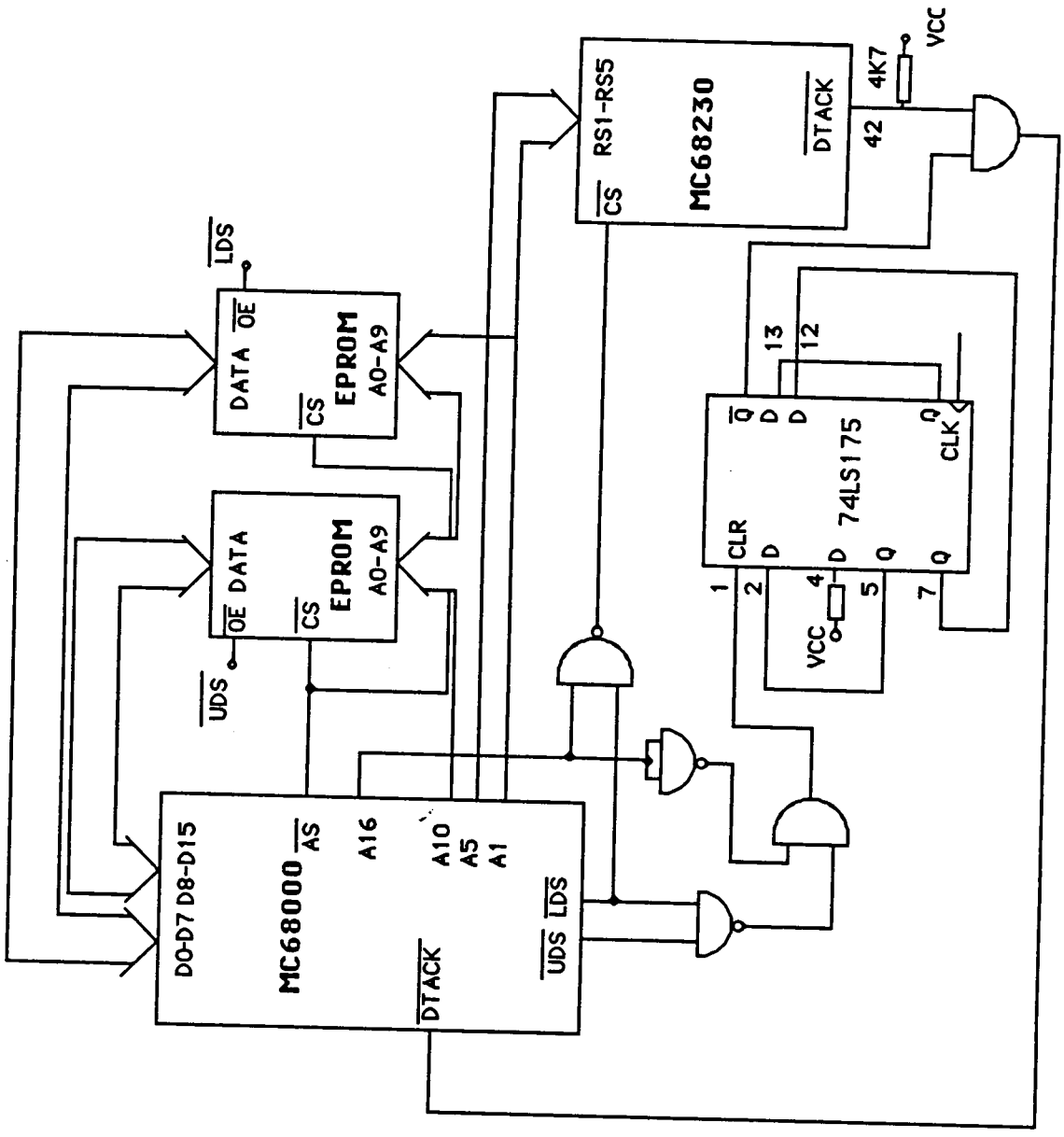


Fig. 4.3: Hardware Configuration for Read/Write operation of MC68000.

As this output of the IR detector is superimposed on a d.c. voltage, the d.c. level is offset by a positive voltage applied at the non-inverting input of the first operational amplifier. The output of the first stage is amplified by a factor of 100 for both the IR detectors.

For exact measurement of the elapsed time, it is desired that H1 and H3 be asserted on the leading edges of the input signals from the two IR detetectors when the amplitude of the signal rises to Vref. Also H_2 and H_4 should be asserted at instants when the amplitudes of the signals decrease to the level of Vref. All the handshake pins are driven by the corresponding comparators with their inverting and non-inverting pins connected as shown in figure 4.1 . When the amplitude of the signal at the inverting end becomes slightly greater than the voltage at the non-inverting end, the output of comparator goes from a high to a low state thereby setting the status bit of the corrosponding handshake pins (i.e. H1 and H3). Note that in practice it is seen that the status bits are set when a negative going signal is applied to the corresponding hand shake pin.

It is also desired that the system should measure the elapsed time with a resolution of 10 milli-seconds. To achieve this, an external clock is used for the counter/timer in the MC68230. In the present design a LM555 is used to generate a clock signal of 100 Hz . This can be replaced by a crystal oscillator if better stability and accuracy is desired. The circuit is shown in figure 4.1 .

A one micro-farad capacitor is charged through the resistors $R_A$and R_B. The voltage across the capacitor is monitored by the IC and when this voltage

reaches 2/3 of Vcc, the capacitor is discharged through R_B via pin 7 viz. discharge pin of the timer IC. This negative transition of the voltage across the capacitor is coupled by connecting pin 6 to the trigger input i.e. pin 2 of the timer IC. At this instant the discharge path via. pin 7 of the IC is removed and the capacitor starts charging again.

The total time interval of a cycle is $T= 0.637 (R_A+2R_B) C$. In the present design, component values are selected such that a 100-Hz square-wave signal is obtained at the output

A Scheme for Accurate Velocity Measurement Which Overcomes Problems Due to the Degraded Output From the Second IR Detector

It is difficult to get sharp pulses at the output of the two IR detectors because unless a very efficient source of heat like a high power laser is used, one cannot heat a particular point on the moving object to create a hotspot . Besides, due to the conduction of the heat from the hotspot to the rest of the body of the moving object, the output signals of the two IR detectors appear as shown in figure 4.4. If the scheme stated in the chapter 2 is used for the elapsed-time measurement , it gives inaccurate results as the slopes of the two leading edges of

the two output signals are different. Flexibility and the tremendious capabilities of the MC 68000 microprocessor allowed us to implement a relatively simple scheme for solving this problem which otherwise cannot be solved easily by the circuits using the SSI/MSI chips.

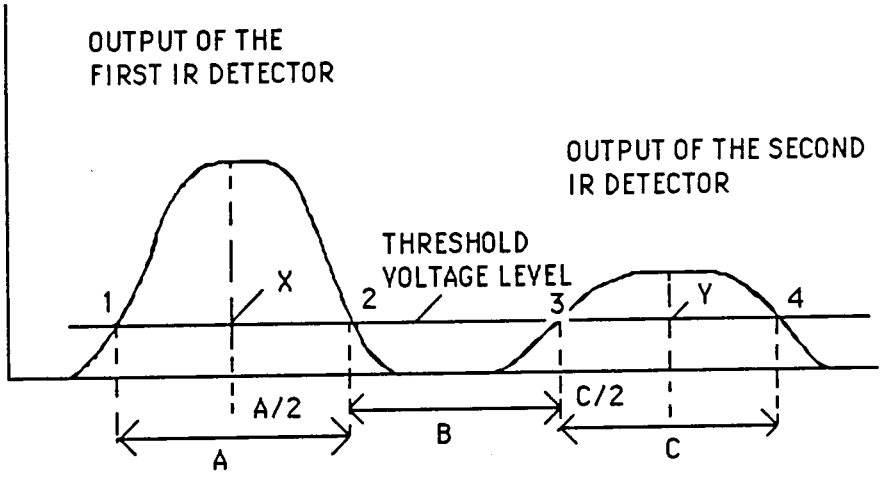


Fig. 4.4 : Signal-conditioning scheme.

In this approach, instead of taking measurements on the leading edges of the two IR detector outputs, the pulse width of both the output pulses are measured. It is obvious that due to heat conduction, the output of the second detector will be more spread out and of lower amplitude. As the conduction of heat is almost uniform in all the directions on the moving body, the center point about which the pulse spreads out remains unchanged.

In this approach both the pulse widths are measured at a predetermined threshold voltage. So the measurement of the interval 'A' starts at the instant (1) and stops at (2). The microprocessor also measures intervals B and C. The central

point from which points (1) and (2) are equidistant is X. Therefore X is A/2 seconds away from the point (2), on the time axis. Similarly the time interval between (3) and (Y) is C/2. Therefore the exact time interval between X and Y is A/2 +B + C/2 secs. As long as the threshold voltage is less than the peak voltage level of the second IR detector output signal, accurate results can be obtained.

The program written for this approach is explained in the next section. The correct elapsed time is obtained irrespective of the pulse-width and amplitude of the signals at the output of the two detectors.

Description of the Software for the Microprocessor-based System

The software program with the file name " ACCUVELO1" was written in the assembly language code for the Motorola MC 68000 microprocessor. Though various signal conditioning schemes for accurate velocity measurement can be employed, this program has been developed using the signal conditioning scheme described in the last section.

The initial part of the program is for initializing various registers of the Parallel Interface/Timer viz. MC68230. This chip is very flexible and can handle a lot of functions. It supports a wide range of operating modes through the programming of its twenty-three internal registers.

The base address of the PT / T is 10001 Hex. The registers are addressed on the lower half of the MC 68000 data bus i.e. odd address locations. The " EQU "

assembler directive is used to assign a label to the address of each register so that this label can be used more easily to refer to the corresponding register in the PI/T, e.g. PGCR EQU 10001 H, this assembler directive enables us to use a label PGCR whenever the address of the 'Port General Control Rigister' of the MC 68230 is to be refered.

The first instruction of the initialization program is MOVE. B # 0FH, PGCR. Now the lower four bits of the Port General Control Register decide the sense of the handshake pins. A positive sense i.e. a pin getting asserted for a 0 to 1 transition, is desired. So the lower four bits of the PGCR are all 1's . Bits 7 and 6 of the PGCR decide the port mode, which in the present case is mode 0. Therefore bits 7 and 6 are kept low. Bits 5 and 4 are H_{34} and H_{12} enable bits. Initially till the execution begins, these control bits are also kept low thereby disabling all the handshake pins.

All the bits of the port A data direction register are kept 'high' so that port A is used as an output port, while all the bits of the port B data direction register are kept low so that all the eight pins of port B can be used as inputs.

Now MOVE. B # 87 H, PACR actually moves bits 1000 0111 to the port A control register. The two most significant bits decide the port A submode. The submode selected is 1X. The code for this submode is 10 for bits 7 and 6. Now bits 5,4 and 3 are all zero. These control the H2 handshake pin. With 5,4 and 3 at low levels, the H2S bit of the status register is set on an asserted edge at H_2. Bit 2 is for H_2 interrupt enable. This bit is kept high so that H2S is set on the asserted

edge at H_2. Bits 1 and 0 are also initialized to the 1 level so that H1S is set on a positive pulse at H1. The port B control register is also initialised like the port A control register to obtain identical operations at port B.

The instruction MOVE . B # 06 H, TCR initialises the ' Timer Control Register' (TCR), bits 7,6 and 5 of TCR are for TOUT/TIACK control. As these pins of the MC68230 are not used in the present design, bits 7,6 and 5 of TCR are kept low. Bits 2 and 1 are for clock control and are kept high. With this configuration the Tin pin serves as the timer input and the prescalar is not used. The 24-bit counter is decremented and is loaded from the counter preload register following the rising edge at the Tin pin after being synchronised with the internal clock. Bit 0 is the timer-enable bit and is initially kept low to keep the timer in the halt state. Bits 4 and 3 of the timer control register are not used.

The next three instructions starting from the instruction MOVE. B # 0FFH, TCPRH load the 24-bit preload register with all ones. Note that all the instructions discussed above use the immediate mode of addressing.

The last instruction of the initialization routine is MOVE. B # 3FH, PGCR. This enables the handshake pins. The address of timer the count register is stored in the address register A5 for further use in the program.

In the beginning of the execution of the main program, the 8-bit data containing the distance between the two IR detectors, which is fed into port B by the operator, is fetched into the data register D1. This data is assumed to be in a binary form. This distance is multiplied by 100. To accomplish this # 64 Hex (

equivalent of 100 $_{10}$) is moved into D_2 and the unsigned multiply instruction i.e. MULU is used. Availability of the multiplication and the division instructions is one of the biggest advantages of using the MC 68000, 16-bit microprocessor over most of its 8-bit counterparts.

The reason for multiplying the distance by 100 is as follows.

Velocity = Distance between two IR detectors divided by the time required for the hot spot to travel from one detector to the other.

The time elapsed is measured with a clock of 100-Hz. i.e. in one second the counter counts 100 pulses. The time is measured with a resolution of 100 th of a second. By multiplying the numerator and the denominator by 100 the fraction in the denominator (i.e. time) can be eliminated. This facilitates the division of the integer values. So the actual distance multiplied by 100, becomes the dividend in this case. This dividend is stored in the address register A1.

BTST # 00H, PSR. This instruction instructs the microprocessor to test bit 0 of the port status register which is H1S. This bit sets when the handshake pin H1 is asserted. The instruction BEQ BEGIN, keep the program in the initial loop until H1S is set by the external hardware. This happens when the hot spot is detected by the detector.

When the output voltage of the IR detector after amplification reaches the level Vref., H1 is asserted. Now the instruction BSET # 00H, TCR sets the bit '0' of the timer control register thereby enabling the timer. On every rising pulse at the Tin the counter starts counting down from the initial value loaded from the

preload register.

After this the program enters the DET2 loop. In this loop the microprocessor keeps on testing bit 1 i.e. H2S of the status register. At the same time the counter continues to count down.

The handshake pin H_2 is asserted by the comparator circuit, when the amplitude of the signal of the first IR detector decreases to slightly less than the Vref. This sets the H2S bit i.e. bit 1 of port status register to '1' level. This is detected by the microprocessor as the instruction BTST # 01H, PSR is executed and bit 0 of the status register in the MC 68000 is changed accordingly. The next instruction BEQ DET2 instructs the microprocessor to check bit 0 of the status register and if the bit has a low level, the program counter branches back to DET2. But if this bit is set to '1', no branch is taken and the next sequential instruction is executed. In this program the next instruction BCLR # 01H, PSR clears the H2S bit of the port status register of the MC 68230. The program now disables the counter by executing the instruction BCLR # 00H, TCR which clears bit '0' of the timer control register. After the counter is halted, the contents of the timer count registers are moved to the D1 register by using the instruction MOVEP. L 0 (A5) , D1. This instruction is for moving the data to and from the peripheral registers which are at alternate memory addresses. The indirect-address register mode of addressing is used. So the D1 data register contains a count for the time interval A. Immediately the timer is enabled by setting bit '0' of the timer control register and counting of the elapsed time interval B starts.

The program now remains in the DET3 loop till H3S is set by the output signal of the second IR detector. Using the same technique described above, the counting is stopped when H3S is set and the contents of the counter representing interval B are moved to the D2 register. Again, the counter is enabled immediately and counting for the third interval C continues.

After H4S is set the counter is disabled and the count for the interval C is moved to the D3 data register.

Note that while measuring each of the time intervals the counter counts down from 00FFFFFF Hex. To get the actual count, the contents of the counter should be subtracted from the original count i.e. FFFFFF Hex. This is done by using the instructions SUB. D1,D4 etc. to get the actual elapsed time intervals A, B and C in D4, D5 and D6.

From the previous discussion of the signal conditioning scheme employed here, it follows that the actual time-interval between the two pulses obtained at the output of the two IR detectors is equal to A/2 + B + C/2. The count for the interval 'A' is in the data register D4. This is divided by two, by using the instruction DIVU D1,D4. Note that initially the number 02 (hex) is loaded into the register D1 before executing the division. Similarly, the time-interval C/2 is obtained. Addition is performed to get the sum A/2 + B + C/2 using the instruction ADD.L and the final sum is obtained in the D4 register.

In this algorithm it should be noted that, though the time intervals A, B and C are portions of a continuous time interval, the counter is stopped. After storing the count for an interval, it is restarted to measure the next interval.

Some time elapses in doing this and is not taken into account in this design because the time measurement is done with a resolution of 10 milliseconds while the time not being counted is only a few micro-seconds. This is because the system clock has a frequency of 4 MHz. The inaccuracy resulting due to the above mentioned reason is negligible.

Now, Velocity = Distance travelled divided by the time elapsed.

This division is performed by the instruction DIVU D2, D5 where D5 has the dividend and the time elapsed is in D2. After the execution the quotient is available in the lower word of D5 while the remainder is in the upper half and both are in the binary form. The next task is a binary to BCD conversion and displaying each digit.

The initial portion of this program for binary to BCD conversion, deals with the quotient part, in otherwords the integer part of the velocity reading. The velocity is displayed in the form XXX.X so the task is to convert the quotient in the binary form to BCD and outputting the BCD data for each of the three interger digits to be displayed. The program should first output a BCD code for the number to be displayed in the unit position and give a pulse which will latch this data into a latch provided for this digit. Then the program should compute the appropriate BCD code for the digit in the ten's position and finally for the digit in the hundred's position.

The algorithm used is as follows. Divide the binary number by 10 (or 'A' Hex). The remainder is the digit in the unit position. Divide the quotient in the above division by ten to get the BCD number in the ten's position. Repeat the

procedure for the third and the final time to get a number in the 100's position.

In the loop labeled BCD1, the velocity in the binary form in D2 is divided by A Hex contained in D1 register. This is done by the instruction DIVU D1, D2. This gives the quotient in the lower word of D2 while the remainder is in the upper word. The remainder is the BCD code for the number to be displayed in the unit's position. To output the remainder in the upper word, first the contents of D2 are moved to D6. Then a SWAP D6 instruction actually swaps the lower and the upper words in D6. Now the remainder is in the lower half of D6.

This remainder is moved to the output of port A by the instruction MOVE. B D6, PADR. The 4-bit BCD number appears at PA3, PA2, PA1 and PA0 bits of port A. These four bits are simultaneousely connected to the four latches each meant for holding the data to be displayed via the corrosponding 7-segment display.

The program now enters the WAIT1 loop. Initially D7 is loaded with a count FF Hex. The instruction DBRA D7, WAIT1 keep on decrementing D7 and branching to the same address (i.e. WAIT1) till the count in D7 becomes negative. In doing this some time is deliberately elapsed by the program before latching, to allow for the setup time of the latches.

After this, a correct bit of the port A register, bit 5 in this case, is set by the instruction BSET D4, PADR. Note that initially # 05 is moved into D4. This pin i.e. PA5 of port A is connected to the clock (or latch input) of the latch in the unit's position. This pin is held high till the delay introduced by the loop WAIT2 is over, after which this bit is cleared making PA5 low. This is used to generate the

software controlled clock pulse for the external latches.

At the end of the BCD1 loop, the contents of D4 are incremented by one so that the procedure discussed above can be repeated for the next i.e. the ten's digit and eventually for the last or the third digit. After three executions of the BCD1 loop, it is terminated as the count in the D3 reaches a negative value and the instruction DBRA D3, BCD1 fails to branch. The remaining portion of the program is for displaying the fractional binary part of the velocity.

For conversion of the fraction in the binary form to a BCD number, the algorithm is different. First the fraction is multiplied by 10 and then the product is divided by the original divisor (i.e. binary distance between the two IR detectors). The quotient is then displayed. It can be clearly seen that the last portion of the program follows the algorithm mentioned above. The BCD number is output and the strobe pulse is generated just like in the display of the integer part discussed earlier.

Instructions For Operating the Microprocessor Based Velocity Measurement System

Determine the distance between the two IR detectors. If this distance is too large, the hot spot may get cooled off by the time it reaches the second IR detector and may not be detected. Because the time is measured by the system with a fixed resolution of ten milli-seconds, the larger the distance between the two IR

detectors, the better is the accuracy.

Input this data regarding the distance to the system in the binary code. Note that the least significant switch inputs the distance = 2^0 = 1 and the most significant switch presents a value = 2^7 =128. The intermediate switches have a value which is a power of two depending on their positions. By using the appropriate switches one can present any number between 0 to 256 to the system. Then depress the 'reset' button which initializes the system. Every time when a new distance is presented using the set of switches, the 'reset' button should be pressed.

The input signal-conditioning circuitary may need to be adjusted for the offset and gain depending on the level of the output signal from the IR detectors. The d.c. component in the output of the IR detectors should be offset using the trimpots on the printed circuit board. Then adjust the gain such that the comparator circuit starts feeding negative going pulses to the hand-shake pins of the MC68000 on the desired leading or trailing edges of the signal. If the velocity measurement system is to be used in the same environment, the signalconditioning circuit may not require frequent adjustments.

To operate this velocity measurement system go through the following steps:

1. Switch on the 5 Volt D. C. power supply.

2. Determine the distance between the two IR detectors and enter this distance in the binary code using a set of eight switches on the circuit bord.

3. Press the 'reset' switch.

4. When the hot spot passes in front of the two IR detectors the velocity is displayed on the LED displays.

Occassionally it may be necessary to adjust the signal conditioning circuit in the following manner:

1. Adjust the offset trimpots to compensate for the 2.5 volt D.C. voltage at the output of each of the detectors. The voltage level can be checked on pin 5 of both the LM324's.

2. Make sure that when a hot spot is detected by the IR detectors the output of the second amplifier (pin 7 of LM324) changes by atleast a volt. If the output is lower than a volt, decrease the distance between the two IR detectors or increase the intensity of the hot spot.

3. Adjust the reference voltage trimpot to half the amplitude of the output signal of the second amplifier.

Features to be Implemented in the Next Version of the Velocity Measurement System

Following are some of the features which will be taken in to consideration while upgrading the design of the velocity measurement system:

1. Averaging: It is desired to add this capability to the next version of the velocity measurement system. The number of readings to be considered for computing the average can be determined by the operator using a set of switches.

These switches can be used to feed the binary data to the microprocessor via port 'C' of MC68230, which is partially idle in the present design. This is the only hardware modification that is required to achieve this task of averaging. The software can be modified to compute the average of a desired number of readings. The softwarecan be written such that a reading far off from the rest of the readings, can be ignored while taking the average.

2. Ability to accept data in the decimal code: The data regarding the distance or the number of readings to be taken for computing the average velocity, can be presented in decimal. It is awkward to determine the binary code for a decimal number every time when a decimal number is to be fed to the system. To achieve this capability either of the software or hardware can be modified. Thumbwheel switches and an EPROM lookup table can be used or the function of the lookup table can be performed by additional software.

3. Automatic offset and gain adjustment: A resistor-capacitor combination can be used with an operational amplifier to compensate for drifts in the D.C. output of the IR detectors. Logarithmic amplifiers can also be used to avoid the need for frequent gain adjustments.

4. Time measurement with a better resolution: In the present system the time required by the hot spot to travel from the first IR detector to the second one is measured with a resolution of ten milli-seconds. For a better accuracy, this resolution can be improved to one milli-second by increasing the frequency of the external clock to 1000Hz. A very minor modification is required to be done in the software. For a better stability a crystal oscillator can be used to replace the

LM555 oscillator used in the present design.

Operational Experience of the Prototype

The prototype of the microprocessor based velocity measurement system was tested using different experimental setups. To test the performance of the digital part of the instrument, the IR detectors and analog signal conditioning circuit at the input of the digital circuit were decoupled. The output of the signal conditioning circuit was simulated by using four debounced micro-switches which fed negative going pulses to the handshake pins of the MC68230 interface chip. This test setup enabled the simulation of the output of the signal conditioning circuit for various conditions such as sharp pulses at the output of the IR detectors, pulses with large pulse-width, output pulses of the two IR detectors having different pulse-widths as well as different time intervals between the output pulses of the two detectors. Under all such conditions the velocity measurement system displayed the desired readings. This test setup was useful to conclude that the performance of the microprocessor based design is satisfactory.

In the other experimental setup the entire velocity measurement system was tested by interrupting the field view of the two IR detectors, at a regular frequency, by a plain card of paper. This piece of paper was attached to a rotating disc and it passed in front of the two detectors once every revolution of the disc. It was found that the velocity readings displayed by the prototype conformed with the

velocity measured by the personal computer monitoring the signal from the same detectors. However at a faster rotational speed of the disc, the velocity displayed was inconsistant because the resolution of the time measurement (10 milli-secs) was inadequate.

In the third experimental setup, hot spots were created on a black PVC cable moving over two pulleys driven by an electrical motor. A high intensity quartz filament lamp was used for generating hot spots. The speed of the cable was monitored using the prototype as well as the personal computer. The readings on both the systems were consistant up to 100 cms/sec which was the maximum speed at which the cable could be run using this setup. The distance between the IR detectors was kept 10 cms. The thyristor speed controller used for controlling the speed of the motor and the motor itself generated electrical noise which adversely affected the performance of the prototype. A surge suppressor was used to suppress this noise. While performing this experiment the switching time of the heat source was carefully controlled to ensure that only a small portion of the heater is heated. It avoided overlapping of the pulses at the output of the two IR detectors. It was necessary for the proper operation of the prototype that the output of the second IR detector does not go high unless the output of the first IR detector goes low. If a larger portion of the cable is heated, both the detectors see it simultaneously and two distinct pulses can not be obtained.

CHAPTER 5

A COMPARISON OF THE THREE DESIGN APPROACHES AND THE CONCLUSION

It is important to note at the outset that initially it was not intended to design three velocity measurement systems with three different design approaches. However, the shortcomings of the initial design and also the decisions to upgrade the initial specifications were responsible for taking the subsequent design approaches.

The first design using the SSI/MSI chips using the CMOS technology and the EPROM besides the operational amplifiers used in the signal conditioning stage was good only for a fixed distance between the two IR detectors. No attempt was made in this design to compensate for the inaccuracy resulting from the degraded output of the second IR detector. For the relatively relaxed specifications, the advantage of this design was simplicity and low cost.

Eventually it was found that the best results are obtained if the distance between the two IR detectors is changed to suit the type and the velocity of the moving object and the type of the heat source. It was not feasible to extend the initial design approach to meet these new specifications. Initially for a fixed and a predetermined distance between the two IR detectors, it was practical to calculate the velocity to be displayed for an elapsed time over an interval from 0.1 to 10

seconds with a resolution of 0.1 second. An EPROM was then used to generate a lookup table where for an elapsed time measured (used as an address) the velocity to be displayed was precalculated and stored as the data. But for the operator selectable distance between the two IR detectors, different EPROMS with a lookup table for that particular distance were required to be used. Also for better resolution and wider range of the elapsed time, the size of the lookup table and hence the EPROM was found to be increasing too much to make this approach feasible.

A serious disadvantage of the first design was that it was difficult to do any mathemetical computations. There was very little flexibility, i.e. any modification in design was difficult to implement as it involved a lot of change in the original wiring. Some times addition of new IC's was necessary which made the problem more complex.

As the need was felt to design a circuit which could perform mathemetical operations such as division, addition etc. , an attempt was made to use the AM 2903 chip manufactured by Advanced Micro Divices. This 4-bit microprocessor slice is capable of doing all the basic mathemetical computations besides the logical functions. Any number of slices can be connected together to improve the data handling capabilities.

Though a design for the velocity measurement system was prepared using the AM 2903, AM 2910 etc., a lot of time had to be spent on figuring out the suitable algorithm for performing the 'integer' divisions. This is because the

internal architecture of the AM 2903 is suitable for floating-point arithmetic. The design complexity increases enormously as one always has to think at the machine level, keeping in mind functions performed by various devices due to a change in a logical level at a control point. Keeping track of the timing considerations is also a difficult task. One of the problems in implementing this design concept was the lack of any development system like a micro-assembler which would have made the task simpler. Even with the availability of the development system, the implementation is difficult considering the hardware involved such as 10 to 4 line encoders, various latches, counters, tristate buffers, EPROMS for lookup tables, micro storage and the pipeline register.

The major advantage of designing the Velocity Measurement System using bit- slice devices is in addition to giving a good understanding of the hardware and timing considerations, this design gives one a very good insight into the extent to which we take things for granted when using a powerful VLSI microprocessor chip like MC68000 or a ready made MC68000 based microcomputer board with peripheral I/0 chips like MC 68230, of tremendous flexibility. Therefore unless a very high-speed programmable velocity measurement system with its own instruction set is to be designed, it would not be practical to use the bit-slice design approach.

It is concluded that the best design approach for making a Velocity Measurement System is using a microprocessor. The MC 68000 which was used in the final design has instructions for unsigned division, multiplication, data movement, bit manipulation which makes design of the software very simple. The

flexibility of the MC68230, parallel Interface/Timer add to the simplicity of the design. The availability of powerful development systems such as HP64000 is one of the biggest advantages. One can edit, assemble, link, emulate and then debug the program step by step. The size of the external hardware reduces tremendously. Use of a powerful development system enables new design idea to be be implemented easily, as most of the changes are generally required to be done in the software with little or no change required in the trivial external hardware. Because of the reduced costs of the micro-computer board, it is also economically viable to manufacture the Velocity Measurement Systems using these micro-computer boards. Reduced hardware will lead to ease of maintenance and higher reliability. In the future more sophisticated versions can be designed by modifying the software, therefore the chances of the hardware inventory becoming useless due to a change in the design are very low.

LIST OF REFERENCES

1. Thomas L. Harman, Barbara Lawson, The Motorola MC68000 Microprocessor Family, Prentice Hall, 1985.

2. Technical Information Center, Motorola Microprocessor Data Manual, Motorola Inc. , 1981.

3. Glenford J. Myers, Digital System Design With LSI Bit-Slice Logic, Wiley, 1980.

4. Donnamaie White, Bit-slice Design: Controllers and ALU's, Gardland STPM press, 1981.

5. John Mick, James Brick, Bit-slice Microprocessor Design,McGraw-Hill,1980.

APPENDIXES

APPENDIX A

```
"68000"
                ORG     1010H
*THIS PROGRAM IS FOR VELOCITY MEASUREMENT USING TWO IR DETECTORS.THE
*SIGNAL CONDITIONING SCHEME USED,GIVES ACCURATE RESULTS IRRESPECTIVE OF
*THE SLOPE OF THE LEADING EDGE OF IR DETECTOR OUTPUTS.THE EXTERNAL ELECTRONIC
*CIRCUIT GIVES TWO NEGATIVE GOING PULSES,AT PREDETERMINED VOLTAGE LEVELS,
*ON THE LEADING EDGE OF EACH IR DETECTOR OUTPUT.THESE PULSES ARE FED TO
*H1,H2,H3 & H4 INPUTS OF MC 68230 PI/T
*THE 68230 REGISTERS*
PGCR            EQU             10001H ;
PSRR            EQU             10003H ;
PADDR           EQU             10005H ;
PBDDR           EQU             10007H ;
PCDDR           EQU             10009H ;
PACR            EQU             1000DH ;
PBCR            EQU             1000FH ;
PADR            EQU             10011H ;
PBDR            EQU             10013H ;
PAAR            EQU             10015H ;
PBAR            EQU             10017H ;
PCDR            EQU             10019H ;
PSR             EQU             1001BH ;
TCR             EQU             10021H ;
TCPRH           EQU             10027H ;
TCPRM           EQU             10029H ;
TCPRL           EQU             1002BH ;
TCRH            EQU             1002DH ;
*INITIALISATION OF 68230 REGISTERS
*PORT A =O/P; PORT B = I/P; USE MODE 0 AND SUBMODE 1X
*LOAD TIMER PRELOAD REG WITH ALL 1'S.INITIALLY DISABLE TIMER
*USE 100 HZ EXT CLOCK AT Tin OF 68230

                MOVE.B  #0FH,PGCR       ;MODE 0, H1234 DISABLED
*                                       ;H1234 POSITIVE SENSE
                MOVE.B  #0FFH,PADDR      ;PORT A O/P, PORT B= I/P
*                                       ;AS PBDDR HAS ALL 0'S WHEN
*                                       ;RESET
                MOVE.B  #87H,PACR       ;SUBMODE 1X,H1S-H2S SET ON
                                        ;ASSERTED EDGES OF H1-H2
                                        ;INTERRUPTS DISABLED.
                MOVE.B  #87H,PBCR       ;       *
                MOVE.B  #06H,TCR        ;Tin USED WITHOUT PRESCALAR.
                                        ;TIMER DISABLED
                MOVE.B  #0FFH,TCPRH      ;LOAD TIMER PRELOAD REG
                MOVE.B  #0FFH,TCPRM      ;WITH ALL 1'S
                MOVE.B  #0FFH,TCPRL      ;
                MOVE.B  #3FH,PGCR       ;ENABLE H12 AND H34.

*THE EXTERNAL HARDWARE PRESENTS A BINARY NUMBER FOR THE DISTANCE  *
*BETWEEN THE TWO IR DETECTORS ,AT PORT B. *
* FOLLOWING *
*PROGRAM FETCHES THE BINARY COUNT,MULTIPLIES IT BY 100 & STORES IT IN A1  *

                CLR.L   D1
                MOVE.L  #TCRH,D1
                MOVEA.L  D1,A5          ;STORE ADDRESS OF TCRH IN A5 FOR FURTHER USE

BEGIN           CLR.L   D1
                CLR.L   D2
                CLR.L   D3
                MOVE.B   PBDR,D1        ;GET DISTANCE BETWEEN TWO DETECTORS
                MOVE.B  #64H,D2         ;GET 100 IN TO D2
```

```
                MULU  D2,D1              ;DIVIDEND = DISTANCE * 100
                MOVEA.L  D1,A1          ;STORE DIVIDEND IN A1

                CLR.L  D1
                CLR.L  D2
*NOW CHECK IF HOT SPOT IS DETECTED BY  DETECTOR1,THE ASSOCIATED CIRCUITARY OF WHICH SETS H1S

                BTST  #00H,PSR          ;TEST IF H1S IS SET
                BEQ   BEGIN             ;ELSE GET LATEST VALUE OF DIVIDEND IN TO A1
                BCLR  #00H,PSR          ;RESET H1S
                BSET  #00H,TCR          ;ENABLE THE TIMER.THE FIRST RISING EDGE-
                                        ;OF CLOCK AT Tin LOADS THE COUNTER FROM
                                        ;PRE LOAD REGISTER & SUBSEQUENT CLK PULSES
                                        ;DECREMENT THE COUNTER

DET2            BTST  #01H,PSR          ;KEEP CHECKING IF THE TRAILING EDGE IS
                BEQ   DET2              ;REACHED OF THE O/P SIGNAL OF FIRST IR DETECTOR
                                        ;WHICH SETS H2S
                BCLR  #01H,PSR          ;CLR H2S
                BCLR  #00H,TCR          ;DISABLE TIMER
                MOVEP.L  0[A5],D1       ;D1 CONTAINS COUNT FOR FIRST INTERVAL i.e.A
                BSET  #00H,TCR          ;START COUNTING SECOND INTERVAL viz.B2
DET3            BTST  #02H,PSR          ;TEST IF THE LEADING EDGE OF SECOND IR
                BEQ   DET3              ;DETECTOR IS REACHED, WHICH SETS H3S
                BCLR  #02H,PSR          ;CLEAR H3S
                BCLR  #00H,TCR          ;DISABLE COUNTER
                MOVEP.L  0[A5],D2       ;D2 CONTAINS DOWN  COUNT FOR SECOND INTERVAL viz.B
                BSET  #00H,TCR          ;START COUNTING THE THIRD TIME INTERVAL i.e. C
DET4            BTST  #03H,PSR          ;KEEP TESTING H4S
                BEQ   DET4
                BCLR  #03H,PSR          ;CLEAR H3S
                BCLR  #00H,TCR          ;DISABLE COUNTER
                MOVEP.L  0[A5],D3       ;D3 CONTAINS DOWN COUNT FOR THIRD INTERVAL viz.C

*AS THE COUNTER COUNTS DOWN FROM FFFFFFH, TO FIND THE ACTUAL TIME ELAPSED, IN EACH OF THE
*THREE INTERVALS ,SUBTRACT THE DOWN COUNT FROM FFFFFFH

                MOVE.L  #00FFFFFFH,D4   ;INITIALISE D4, D5 & D6
                MOVE.L  #00FFFFFFH,D5
                MOVE.L  #00FFFFFFH,D6

                SUB.L  D1,D4            ;FIRST INTERVAL viz. A
                SUB.L  D2,D5            ;SECOND INTERVAL viz.B
                SUB.L  D3,D6            ;THIRD INTERVAL viz. C


*THE DURATION OF THE OUTPUT PULSES OF THE IR DETECTORS AT A PREDETERMINED VOLTAGE *
*LEVEL viz. A & C IS NOW KNOWN. THE PEAK OF THESE PULSES IS AT A/2 & C/2.THE EXACT*
*DURATION OF THE TIME ELAPSED = A/2 + B + C/2

                CLR.L  D1
                MOVE.B  #02H,D1
                DIVU  D1,D4             ;DIVIDE INTERVAL A BY 2
                ANDI.L #0000FFFFH,D4    ;KEEP ONLY QUOTIENT
                DIVU  D1,D6             ;DIVIDE INTERVAL C BY 2
                ANDI.L   #0000FFFFH,D6  ;KEEP QUOTIENT
                ADD.L  D6,D4
                ADD.L  D5,D4            ;A/2 + B + C/2, EXACT TIME ELAPSED IN D4


                CLR.L  D1
                CLR.L  D2
                CLR.L  D3

                MOVE.L  D4,D2           ;PRESERVE TIME ELAPSED IN D2
                MOVE.L  D2,A2
                CLR.L   D4
                CLR.L   D5
                CLR.L   D6
                MOVE.L  A1,D5           ;GET DIVIDEND IN D5

*VELOCITY = DISTANCE  TRAVELLED / TIME ELAPSED *

                DIVU  D2,D5             ;GET VELOCITY IN  BINARY
                                        ;LOWER WORD OF D5 IS QUOTIENT
                                        ;& UPPER ONE IS REMAINDER.
```

```
* CONVERT BINARY QUOTIENT TO BCD & DISPLAY 3 MOST SIGNIFICANT DIGITS *

                MOVEQ   #02H,D3          ;SET COUNTER FOR 3 DIGITS
                CLR.L   D4
                MOVE.B  #05H,D4          ;ENABLE STROBE OF L.S.DIGIT

*NOTE: FOUR MSB OF PADR ARE STROBES FOR LATCHES FOR EACH DISPLAY  *
*BITS PA7,PA6,PA5 ARE STROBES FOR 3 INTEGER DIGITS & PA4 IS STROBE *
*FOR FRACTION.                                                     *

                CLR.L D1
                CLR.L D2
                MOVE.B  #0AH,D1          ;MOVE 10 TO D1
                MOVE.W  D5,D2            ;GET ONLY QUOTIENT IN LOWER
                                         ;WORD OF D2
BCD1            DIVU  D1,D2              ;DIVIDE QUOTIENT OF VELO.
                                         ;BY 10
                MOVE.L  D2,D6            ;PRESERVE D2
                SWAP  D6                 ;GET REMAINDER IN LOWER WORD
                ANDI.L  #0000000FH,D6      ;KEEP ONLY 4 LSB. BCD VALUE
                ANDI.L  #0000FFFFH,D2      ;CLEAR REMAINDER & USE ONLY
                                         ;QUOTIENT FOR NEXT DIVISION
                MOVE.B  D6,PADR          ;OUTPUT BCD  TO LATCHES
                CLR.L   D7
                MOVE.W  #00FFH,D7          ;WAIT1 IS FOR ALLOWING
WAIT1           DBRA    D7,WAIT1         ;SET UP TIME FOR LATCHES
                BSET  D4,PADR            ;SET STROBE BIT OF LATCH
                CLR.L D7
                MOVE.W  #0FFFFH,D7         ;WAIT2 DECIDES TIME FOR WHICH
WAIT2           DBRA  D7,WAIT2           ;STROBE IS HIGH
                BCLR  D4,PADR            ;CLEAR STROBE BIT
                ADDQ.B  #1,D4              ;FOR STROBING NEXT LATCH.
                DBRA D3,BCD1             ;REPEAT LOOP 3 TIMES

*BCD CONVERSION & DISPLAY OF FRACTION *

                MOVE.L  D5,D2            ;GET VELOSITY IN BINARY
                SWAP  D2                 ;REMAINDER IN LOWER WORD
                ANDI.L  #0000FFFFH,D2    ;KEEP ONLY REMAINDER
                MULU    D1,D2            ;MULTIPLY BY  10

                CLR.L  D1
                MOVE.L  A2,D1            ;GET TIME ELAPSED
                DIVU    D1,D2            ;AFTER MULTIPLYING ORIGINAL FRACTION-
                                         ;BY 10 DIVIDE IT BY DIVISOR i.e.TIME
                ANDI  #0000000FH,D2      ;BCD FRACTION
                MOVE.B  #0AH,D1            ;RESTORE D1

                MOVE.B  D2,PADR          ;BCD FRACTION TO LATCH


                CLR.L   D7
                MOVE.W  #00FFH,D7        ;SETUP TIME
WAIT3           DBRA  D7,WAIT3

                CLR.L D7
                MOVE.B  #04H,D7

                BSET.B    D7,PADR        ;ENABLE STROBE

                MOVE.W  #0FFFFH,D7         ;STROBE HIGH
WAIT4           DBRA    D7,WAIT4
                CLR.L D7

                MOVE.B  #04H,D7
                BCLR.B    D7,PADR        ;CLEAR STROBE
                MOVE.L  A5,D5            ;RESTORE ORIGINAL DIVIDEND
                BRA   BEGIN
END
```

APPENDIX B

Alternative Approach for Signal Conditioning

There are a couple of problems in creating a sharp hot spot on the moving object. First of all, the source of heat cannot impart heat to a spot on the moving object instantaneously. It needs some finite time to do it. As such, instead of a spot, a line on the object gets heated. Due to conduction, the heat from this portion gets conducted to the rest of the body. As a result the first IR detector has the output waveform of figure 6.1 instead of a sharp pulse.

As the moving object travels ahead and as more time elapses, more heat gets conducted. So the output of the second IR detector is more sluggish and lower in amplitude.

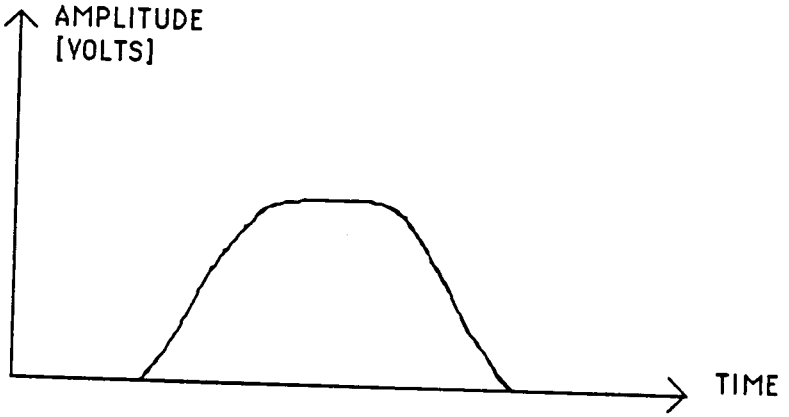


Fig. 6.1: Output of the IR detector.

The slopes of the leading edges of the output signals vary depending on velocity, distance and intensity of the heat source, etc. Hence it is very important to use a signal conditioning scheme which will enable an accurate velocity measurement to be made, irrespective of the output signal of the IR detectors.

The basic principle is highlighted in the following paragraphs:

From figure 6.2 it can be seen that, in Δ^{s} OPQ and ORS,

SR/QP = OR/OQ But OR = A' and OQ = X

$\therefore$ SR/QP = A'/X ------(1)

But QP = QT-PT But SR = QT

= SR - PT ---------(2)

Now in Δ^{s} ORS and UTP

SR/OR = PT/UT

$\therefore$ SR/A' = PT/A (as Δ^{s} UTV and VQW are concurrent $\therefore$ UT = A)

$\therefore$ PT = A/A' SR.

Substituting in (2)

QP = SR (1- A/A')--- (3)

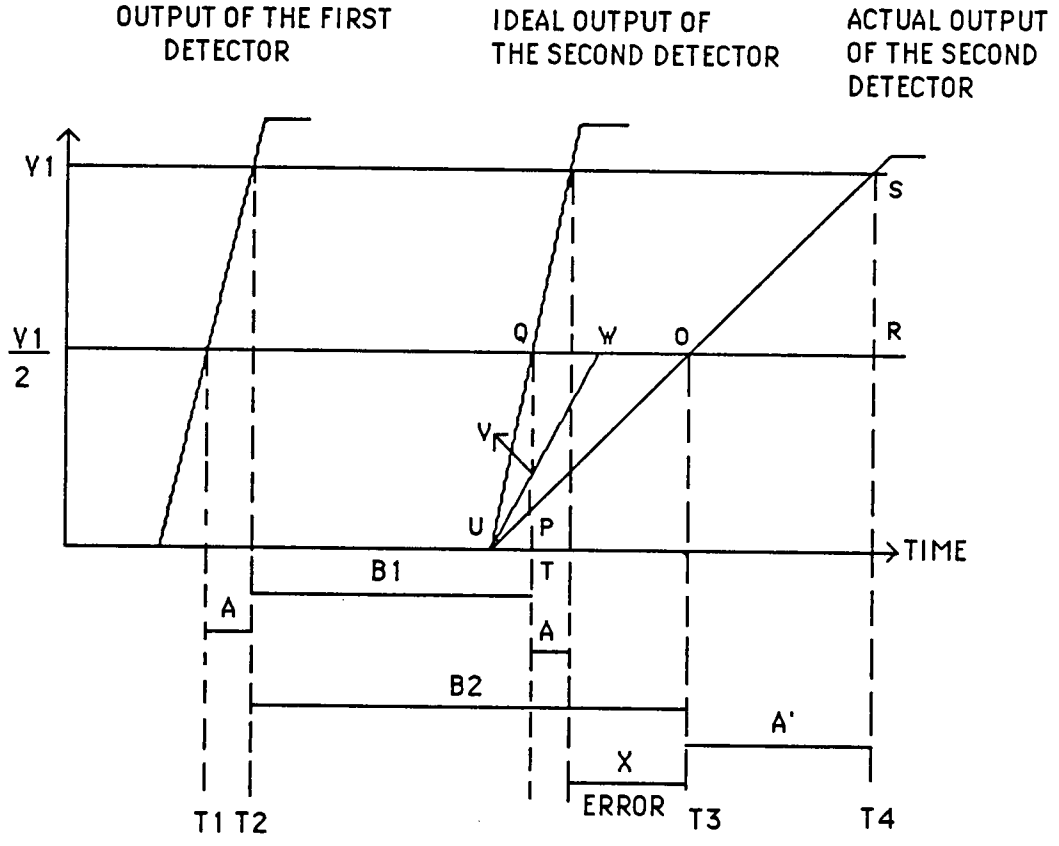


Fig. 6.2 : The alternative signal-conditioning scheme.

Substituting (3) in (1) we get

$X = A' (1-A/A')$

$\therefore$ $\underline{X = ERROR = A' - A}$

Implementation of the Alternative Scheme

From figure 6.2, it can be seen that the actual time elapsed is A+B1 if the signals at the output of both the IR detectors are identical. But in reality the output of the second IR detector is sluggish compared to the first one.

Two comparators are used in conjunction with each IR detector to give pulses at the triggering voltage levels V1 and V1/2 . It is essential to have one triggering voltage level exactly half of the other for X to be equal to A'-A. This can be seen from a trigonometric derivation.

The triggering pulses at the output of the comparators are obtained at instants T_1, T_2 and T_3, T_4. These are coupled to the handshake pins of the micro-computer, which in turn measures the time intervals A, B_2 and A'.

Now, if the output of the second IR detector were ideal, the actual time required for the hotspot to travel from the first IR detector to the second one would be A + B1 as can be seen from the diagram.

However due to the decrease in the slope of the second IR detector instead of B1 the time measured is B2 giving an error 'X' . To measure exact time interval B1, this error 'X' has to be subtracted from B2.

From the previous trigonometric derivation it is seen that the error

X = A' - A.

As both the time intervals A' and A are known, the error or extra time measured, i.e. X, is calculated using the microprocessor.

The actual time required by the hotspot to travel from the first to the second IR detector is

$= A + B1$

$= A + (B2 - X)$

This measurement of the three time intervals viz. A, B2 and A', calculating the error X and the final answer is all done by the MC68000-based system. The correct elapsed time is obtained irrespective of the slopes of signals at the output of the IR detectors. The program written for this approach is very similar to the program for the previously discussed approach. Only the design of the comparators change a little as in this case we have to detect the voltage levels only on the leading edges of the signals.

VITA

Dilip P. Nawathe was born in Pune, India on June 30, 1956. He graduated from the University of Poona in 1980. He received a Bachelor of Engineering degree in Electronics and Telecommunication with a distinction.

After working in a research organisation for six months, he floated a small scale industry in June, 1981 and started manufacturing a variety of electronic instruments and controllers. In addition to designing the instruments he was also responsible for the marketing and sales. After successfully managing this company till January 1985 he decided to come to U. S. A. for higher education. He accepted a teaching assistantship at The University of Tennessee, Knoxville and began study towards a Master's degree.