1988

An 8085 Microprocessor based monitor system for a 750 cc Honda motorcycle

Robert H. Leet

Rochester Institute of Technology
School of Computer Science and Technology

An 8085 Microprocessor Based Monitor System For A 750 cc Honda Motorcycle

By
Robert H. Leet

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Approved by:

Protapa Reddy
Dr. Protapa Reddy

Peter G. Andrews
Dr. Peter Anderson

Rayno Neimi
Dr. Rayno Niemi

March 24, 1988

I Robert H. Leet prefer to be contacted each time a request for reproduction is made. I can be reached at the following address:

15417 Preston Road #1152

Dallas, Texas 75248

March 25, 1988

Abstract

This system replaces the analog speedometer and tachometer gauge cluster of a 750 cc Honda motorcycle with a computerized monitor system based on the INTEL 8085A microprocessor and family of peripherals. The system adds the enhanced functionality of engine temperature, battery charging, fuel level, turn signal, and kick stand monitoring and status display. A security function is also employed as a guard against unauthorized use or theft.

The speedometer and tachometer functions utilize phototransistor sensors generating pulses derived from stock cable connections to the engine block and front wheel.

The engine temperature, fuel level, and battery charging functions are implemented via A/D conversion of external sensor inputs.

Turn signal and kick stand status functions are implemented by software polling of external switches.

The system incorporates a small keyboard for input of a five digit security code. If the code input is correct, ignition is enabled and normal operation may follow. If the code input is incorrect, ignition is disabled, and the motorcycle horn is activated as an alarm. The operator is allowed two attempts at entrance of the five digit code.

All system functions are displayed via various configurations of 7-segment and discrete LED displays for operator feedback.

The system utilizes the twelve volt motorcycle power source and is contained on three double sided circuit boards.

The system software was developed on the INTEL SDK85 system design prototype kit and transferred in final form to an on-board EPROM.

Subject code classifications taken from The New (1982) Computing Reviews Classification System published by the ACM.

Primary

C. Computer System Organization

C.3 Special Purpose and Application-Based Systems

Real Time Systems

Secondary

B. Hardware

B.4 Input/Output and Data Communications

B.4.3 Interconnections (subsystems)

Interfaces

B.7 Integrated Circuits

B.7.1 Types and Design Styles

Input/Output Circuits

Microprocessors and Microcomputers

Table of Contents

1 Introduction and Background. 1

Key Words and Phrases. 3

2 Functional Specification. 13

3 System Design. 15

Major System Components. 15

Detailed System Functions. 18

Speedometer. 18

Tachometer. 19

Engine Temperature. 19

Battery Charging Voltage. 20

Low Fuel Warning. 20

Turn Signals. 21

Kickstand Down Warning. 21

Security. 22

System Architecture. 23

Bus Structure. 23

Internal/External Bus Structures. 23

Display Structure. 25

LED/Keyboard Layout. 25

Display Component Interconnect. 25

Address Assignments. 30

Memory Mapped vs I/O Mapped I/O. 30

Address Decoding. 31

Chip Select Map. 31

System Memory Map. 33
System I/O Port Map. 33
Keyboard Map and Data Word. 36
Circuit Board Interconnect. 36
System Connector Map. 36
TTL/CMOS Interfacing. 39
Voltage Levels. 39
Fanout. 43
Filtering. 44
Multiple Voltage Handling. 45
Voltage Division. 45
Current Requirements. 47
Bus Current Requirements. 47
System Current Requirements. 49
Interrupt Structure. 49
Vectored Interrupts. 49
Polled Interrupt Mask. 52
4 Device Interfaces. 54
Display Interface. 54
8255 Device Latch Control Logic. 55
Speedometer/Tachometer Interface. 56
Phototransistor Timing. 58
Analog Signal Conditioning. 59
Interrupt Timing. 60

Pulse Count v.s. Interface Gear Ratios. 60
Security. 62
Keyboard. 62
Use of the 8279 Control/Data Words. 62
Interrupt Handling. 63
Alarm. 63
Ignition Disable. 66
Turn signals/Kick Stand. 67
Switch Sensor Interface. 67
Polled Detection Logic. 68
Led Flash Cycle/Timing. 68
A/D Conversion. 69
Temperature Sensing. 70
Kelvin to Fahrenheit Conversion. 70
Low Fuel/Battery Charging Sensing. 71
Voltage Division. 71
Addressing the ADC0808 A/D Converter. 71
ADC0808 System Precision. 72
5 Software Design. 74
Processor Modules. 74
System Initialization. 75
Left and Right Turn Signals. 81
Hexadecimal to BCD Conversion. 88
Speedometer/Tachometer Computation. 92

Keyboard/Security Computation. 96
A/D Conversion and Scaling. 103
Mathematic Routines. 106
Warning Light Logic. 121
Warning Light Flag Set Routines. 121
Warning Light Table Look-up. 122
System Delays. 126
System Monitor. 127
Databases. 131
System RAM/EPROM Usage. 131
Communications among Modules. 132
Common Access Method. 132
Value Passing. 133
Handling Context Switches 134
System Monitor Timing. 135
Variations Depending on System Conditions. 135
6 Integration and Test. 136
Module Development. 136
Module Test. 136
System Integration. 137
7 Computer Assisted Development. 140
A/D Conversion. 140
Computer Simulation. 144
Speedometer and Tachometer Functions. 144

Disk Activator Stripe Count and Placement. 144
Function Simulation For Hardware Debug. 146
8 System Design Tools. 149
Designer/Programmer Notebook. 149
SDK85 System Design Kit. 149
EPROM Programmer. 150
Use of EEPROMS in development. 150
Miscellaneous Design Tools. 151
9 Design Process. 152
Schematics. 152
CAD. 152
Fabrication..... 154
10 Complete System Testing. 156
Problems Encountered and Solved. 156
Noise. 156
Speedometer and Tachometer. 156
Effects From System Noise. 156
Phototransistor Light Noise. 157
Effects on Switching Transistor. 157
74LS373 Transient Noise. 158
Effects of A/D Fluctuations on Display. 159
Bus Timing Synchronization. 161
Need For Artificial Clock Pulse to 8254. 162
SDK85 Kit Limitations. 164

11 Discrepancies and Shortcomings of the System. 165
System Precision. 165
A/D Conversion Error. 165
Temperature Sensor Precision. 165
Simulation For Demonstration. 166
12 Final Installation Considerations. 167
Cooling Requirements. 167
LED Light Filtering. 167
Effects of Vibration. 167
Enclosures. 168
13 Conclusions. 169
Alternative Approaches for Improved System. 169
Point Closure Sensing for Tachometer Function. 169
Magnetic Sensing for Speedometer function. 169
Use of Isolated Voltage Reference For A/D. 170
Updated Technology. 171
Design Tradeoffs. 173
Hardware/Hardware. 173
Hardware/Software. 174
Suggestions for Future Extensions. 175
LCD Displays. 175
Trip Computations. 175
Speech. 176
Dual Power Sources for System and Horn. 177

Related Thesis Topics for the Future. 177

Implementing Security Algorithms in Hardware. 177

A/D Conversion of Real Time Events. 178

Hardware Synchronization and Software Control. 178

Bibliography. 179

Appendices. 182

Schematics. 221

The topic for this thesis was conceived as a result of participation in a microprocessor class during summer quarter 1982.

During completion of the curriculum, I found that my true interest in computer science lay not in software theory and programming, but in hardware and systems software at the lowest levels.

I have a mechanical background, and have always had an interest in how things work. It was a natural extension to take an interest in the electronic workings of computers, and set out to build a small system as a result. The understanding of the workings of computer hardware gained in the microprocessor course provided the impetus to begin this project.

Having no electronic background, I quickly learned that what seemed like a very feasible idea was in reality a much larger project than I had anticipated.

For successful completion, I had to cross disciplines by self-teaching basic electronics, and greatly expand my knowledge in computer system hardware, electrical characteristics, bus structures, timing, control, interfacing, and debug.

In addition, I had to master the proper use of an oscilloscope and other test equipment, acquire a prototype development system, an EPROM programmer, and a multitude of electronic parts and data books, and also learn to fabricate circuits from a bare copper clad board to a fully etched, drilled, and soldered board.

As the system grew, it became apparent that breadboarding or wire wrapping would prove inadequate, so as a final requirement, I taught myself computer aided schematic capture and printed circuit design to develop the end product.

The following document and completed firmware system represent literally more than one thousand hours of learning, experimentation, design, fabrication, and debug.

The resulting experience and knowledge gained are invaluable and will prove to be only the beginning of future endeavors.

A/D Analog to digital, as in analog to digital conversion.

Active As in an integrated circuit, the working state of the device or pin. At the TTL pin and gate level, the voltage level needed to produce the active state, such as active high (3.4 volts) or active low (0.8 volts).

Address The numerical, logical, or symbolic value representing the location of a particular device or location within a particular device in a computer circuit.

Algorithm A particular section of computer program code designed to perform a specific function.

Analog Signals that are "regulated" between different levels as opposed to signals that are "switched"; called digital. In instrumentation, the representation of a value using a dial or other non-numeric means.

Binary A number system where the weights of the digits increase or decrease by a factor of two and where each digit can have only one of two possible values; 0 or 1.

Branch To change location. In a computer system, to alter the location of program execution, usually facilitated by an altering of the contents of the program counter by a jump or call instruction.

Breadboard A hardware development tool comprised of many spring loaded electrically conductive holes where devices can be inserted and connected with wire jumpers and removed without soldering for prototype development.

Buffer Logically, an area within a computer system for temporary storage of data. Electronically, usually an intermediate device between two primary devices which retains data and often times supplies extra current drive capability.

Burn Electronically altering the contents of a memory location by applying a short high voltage pulse, as in burning a program into an EPROM.

BCD Binary Coded Decimal. The binary means of decimal representation using only the decimal digits 0-9.

Bus Two or more conductors running in parallel used for carrying electronic data information.

Calibrate To fix or alter the graduations of a particular device or scale.

Celsius A temperature scale representation based on a range of possible values where zero degrees is the freezing point of water and one hundred degrees is the boiling point.

Channel A single or multiple bus construct used for transmission or receipt of electronic signals or data.

Chip Select A signal or pin, which when activated will place a specific device in the working state.

Clock A device which produces a regular measurable pulse. Also, the act of applying a particular signal to a device or device pin.

CMOS Complimentary Metal Oxide Semiconductor

Comparator A semiconductor device which has as input two or more voltages and compares the input levels until they are the same, at which time the output of the comparator will become active.

Context Switch The saving and restoring of all the values of registers and memory locations associated with current execution when program execution is interrupted and continues in a different location in the program code.

CPU Central Processing Unit

Debounce The hardware or software means of determining a key closure by scanning the key at a given interval to filter the microscopic multiple contacts known as key "bounce".

Decode As in the 74LS154 4:16 binary decoder usage, to accept a four digit binary value and activate a specific output signal based on that value.

Demultiplex To isolate specific uses within a multi-use or shared signal line. In the case of the 8085 multiplexed bus, it contains both address and data information. Demultiplexing of the 8085 bus isolates address or data information for use by system components.

Discrete Single, individually packaged devices. As opposed to integrated circuits, examples of discrete components are resistors, capacitors, diodes etc.

Drive Pertaining to a signal, or current source, the originating device is said to "drive" the other devices which use that current source or signal for operation.

Dynamic Ever changing. Also, in memory devices, dynamic devices require a periodic refresh pulse to retain information.

EEPROM Electronically Erasable Programmable Read Only Memory. Contents of permanent memory are altered via application of a short high voltage pulse.

EPROM Erasable Programmable Read Only Memory. Contents of permanent memory locations are erased by exposure to ultraviolet light.

Fahrenheit A temperature scale representation where thirty two degrees represents the freezing point of water and two hundred twelve degrees represents the boiling point of water.

FIFO First In First Out. In application in a queue or stack, the first information placed in the queue or stack will be the first information retrieved.

Firmware The combination of hardware and software.

Flag In a computer system, a one or more bit value used to represent a specific condition.

Float A hollow, sealed structure that floats on a liquid as in a gasoline tank. The float usually has a variable resistance device attached, and as the float raises and lowers with the level of fluid, the variable resistor's resistance changes.

Gerber Code Specific numerical control data, the format of which was developed by the Gerber Scientific Company recognized as the leader in the photoplotter machine industry.

HALL Effect Named after it's inventor, Hall, the effect produced between magnetic objects.

Hex Short for hexadecimal. Also a prefix meaning six as in a hex inverter device which contains six inverters.

Hexadecimal A number system based on the number sixteen, the digits of which are valued 0-9, and A-F. The weights of these digits increase or decrease by a factor of two.

High In TTL voltage levels, a voltage level of 3.4 volts or higher.

Histogram In computer aided design systems, a graphic/numeric tool used to provide information on component and signal density within a circuit board.

Interface The means by which two compatible or non-compatible entities are united or communicate.

Interrupt Relating to a microprocessor, the existence of a high or low voltage level (>3.4 or <0.8 volts) at a particular pin, which when recognized, causes the microprocessor to branch to a particular area in memory, and execute the instructions found there.

I/O Input/Output

I/O Mapped The means of configuration of addresses within a computer system such that all devices within that system look like a series of I/O ports to the CPU.

Kelvin A temperature scale representation where zero degrees Kelvin corresponds to -273.15 degrees Celsius.

Kill Pertaining to the ignition circuitry of a motorcycle, a switch, which when opened breaks the circuit and "kills" the engine.

kHz Kilohertz. One thousand hertz or a frequency of one thousand cycles per second.

Latch A semiconductor device which when activated, will retain information until the next access.

LCD A light absorbing visual display device; Liquid Crystal Diode.

LIFO Last In First Out. As used in a queue or stack implementation, the last information stored in the queue or stack is the first information to be retrieved.

LED A light reflecting visual display device; Light Emitting Diode

Logic "0" A digital logic state represented by a TTL voltage level of 0.8 or lower. Also synonymous with a signal state of "low".

Logic "1" A digital logic state represented by a TTL voltage level of 3.4 volts or higher. Also synonymous with a signal state of "high".

Low The state of a TTL signal whose voltage level is 0.8 volts or less.

LSTTL A TTL family or complexity; Large Scale Transistor Transistor Logic.

mA Milliamp. One thousandth of an amp.

Mask In microprocessors, a variable length data word containing flags representing or affecting the states of various functions such as an interrupt mask. Also, to cover or ignore, as in mask out all but the third bit of a data word.

MHz Megahertz. One million hertz, or one million cycles per second.

Millisecond One thousandth of a second.

Memory map A graphical representation of the devices which are resident (accessible) at different locations over the addressable range of the system, where all devices appear as a series of memory locations to the CPU.

Mnemonic A programming construct of abbreviations representing executable commands used by the CPU.

Monitor A small operating system which coordinates and activates all system software execution.

Multiplexed The shared use of one or more bus lines to convey different types of data at different times.

ns Nanosecond. One billionth of a second.

Noise Voltage spikes whose levels extend above and below the given d-c reference voltage.

Nominal Middle or average as opposed to maximum or minimum.

Opcode The numeric or alphanumeric representation of an executable command of a CPU.

Operand The numeric or symbolic representation of the data to be manipulated by an executable instruction (operation).

Parallel Two or more bus lines used for the same purpose, or two or more bits of information sent or received at the same time.

Personality Module Used in an EPROM programming device, the interchangeable combinations of components used to make the programmer operate correctly with a given EPROM device.

Phototransistor A transistor whose base is activated by the application of light.

Picofarad One trillionth of a farad.

Pinout The graphical representation of the location and numbers of the pins of a given electronic device.

Pointer The symbolic or numeric representation of an address location within a computer system.

Poll The monitoring of a device through software control as opposed to the device gaining attention through an interrupt.

Port A semiconductor "doorway" that interfaces internal and external bus devices.

PPI Programmable Peripheral Interface

Pull-up A method to raise the voltage level of an electronic signal. Usually implemented by a parallel resister between the signal and a higher voltage, the signal is said to be "pulled up" to the higher voltage in operation.

RAM Random Access Memory-often called read/write memory

Register A semiconductor storage vehicle usually implemented as a memory location or area within the CPU.

Relay An electromechanical device consisting of a coil and switch mechanism. The switch opens or closes upon application of a voltage to the coil.

ROM Read Only Memory, the contents of which are not alterable by an application of a particular voltage level.

Routing In printed circuit board technology, the copper signal paths on a circuit board.

RPM Revolutions Per Minute.

Saturation Voltage The voltage, which (when applied to the base of a transistor) activates the device.

Scan To look at. As in the 8279 programmable keyboard device, the scan lines apply a voltage to the keyboard matrix or "look at" the matrix and a particular scan signal is returned via the return lines if a key has been pressed.

Serial One bus line, or the sending or receiving of a single bit of data at a time, as opposed to parallel.

Signal The symbolic name or representation of the existence of a voltage level within a circuit.

Signal Conditioning The alteration of the voltage or current characteristics of a signal, as in squaring an analog signal into a digital pulse.

Sink To accept or to terminate. In semiconductor devices, to accept or use current which has originated from another device.

Sinusoidal A slowly changing signal which resembles a sine wave as opposed to a quickly changing digital pulse.

Stack An area in memory (RAM) or CPU, which is used by the CPU for temporary storage during program execution. The stack is most often instituted as a LIFO queue.

Static Unchanging. In memory devices, information is retained without a periodic refresh signal as opposed to dynamic.

Tap A connection into or out of.

Transient A non periodic signal. As in noise, a signal that comes and goes sporadically at undefined intervals.

TTL Transistor-transistor logic

Vector A particular location within measurable bounds. In computer systems, usually a memory location.

Functional Specification

The purpose of this system is to monitor seven vital motorcycle functions (speed, rpm, temperature, charging voltage, kick stand status, fuel level and turn signals), provide a security function which controls ignition and an alarm, and provide operator feedback via LED's and the horn.

The system is comprised of three major functional subsystems: 1) Sensor input, 2) Computation logic, and 3) Display/Relay driving.

Sensor input can be further subdivided into four types: 1) Keyboard, 2) Switch, 3) Voltage sensitive, and 4) Pulse sensitive.

The security function is implemented via input from the keyboard.

The kickstand status and the left and right turn signal functions are implemented via monitoring of external switches for an on or off state.

Voltage sensitive sensors are used as input to an A/D converter for monitoring of the temperature, charging voltage, and fuel level functions.

Pulse sensitive sensors are used as input for the speed and tachometer functions.

The computational logic subsystem accepts inputs from the various external sensors via I/O interface circuitry, and through a combination of software polling and hardware interrupts utilizes specific algorithms to compute actual values for display to the operator.

The display/relay driving subsystem utilizes various display driving algorithms to relay the computed values to the proper discrete and 7-segment LED displays representing each of the systems functions, and depending on security code input, energize or de-energize the relays controlling the ignition and alarm.

Figure 1 diagrams the three major subsystems and their major internal blocks.

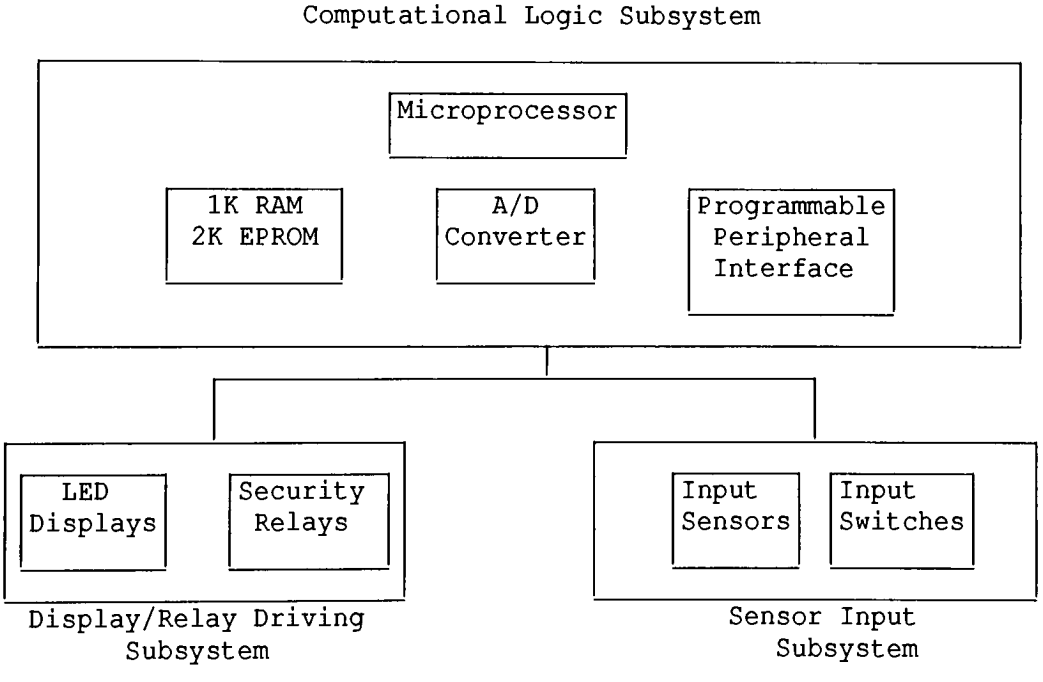


Figure 1

System Design

This system (as is true for any computer system), could have been constructed using discrete logic devices. Given the thousands of gates used in VLSI (Very Large Scale Integrated) devices such as the 8085, this approach would have been enormously complicated and would have resulted in a huge end product package.

Additionally, a 4-bit or 16-bit CPU could have been used, but would have resulted in difficult interface and increased software overhead in the case of the 4-bit CPU, and overkill in the case of the 16-bit CPU.

The 8085 microprocessor was chosen as the CPU for this system due to market popularity and vendor support. In 1983 when this project was started, INTEL microcomputers had gained wide popularity through a series of industry development leads, beginning with a 4-bit microcomputer and followed by the 8080 and 8085. Today, INTEL products have become the industry standard with their '386 technology used in most high end personal computer systems.

Vendor support has remained strong via development of a large family of support devices such as co-processors, I/O, memory, and graphic chips.

Major System Components

The sensor input subsystem utilizesa variety of individual sensors as input devices from the motorcycle's functions.

Phototransistors are used for the speedometer and tachometer functions, a precision temperature sensor for the engine temperature function, and direct interface to the motorcycle's electrical system for input from the fuel level, turn signal, and kick stand functions.

The computational logic subsystem contains components centered around an INTEL 8085 microprocessor and family of peripherals.

Included in this subsystem are the RAM, ROM, address decoding, and A/D converter components, as well as I/O components for interfacing the external sensors, keyboard and displays.

The display/relay driving subsystem contains relays to control the ignition and alarm circuitry, and two types of LEDs for display of computed values to the operator. A series of discrete LEDs are used to represent the turn signals and warning light functions (fuel level and kickstand status). Display of the speed, tachometer, temperature, and charging voltage values is accomplished via individual 7-segment LEDs.

Care was taken during design to stay within the INTEL family of components as much as possible to insure electrical and logical integrity. Non-INTEL components were chosen based on their ability to interface to the core INTEL system with a minimum of additional circuitry, and their ability to accomplish their given function in a cost effective manner.

Refer to Appendix A for a complete discussion of the major system components, their system functions and interrelations.

Figure 1a shows the expanded three functional subsystems and their major generic internal components.

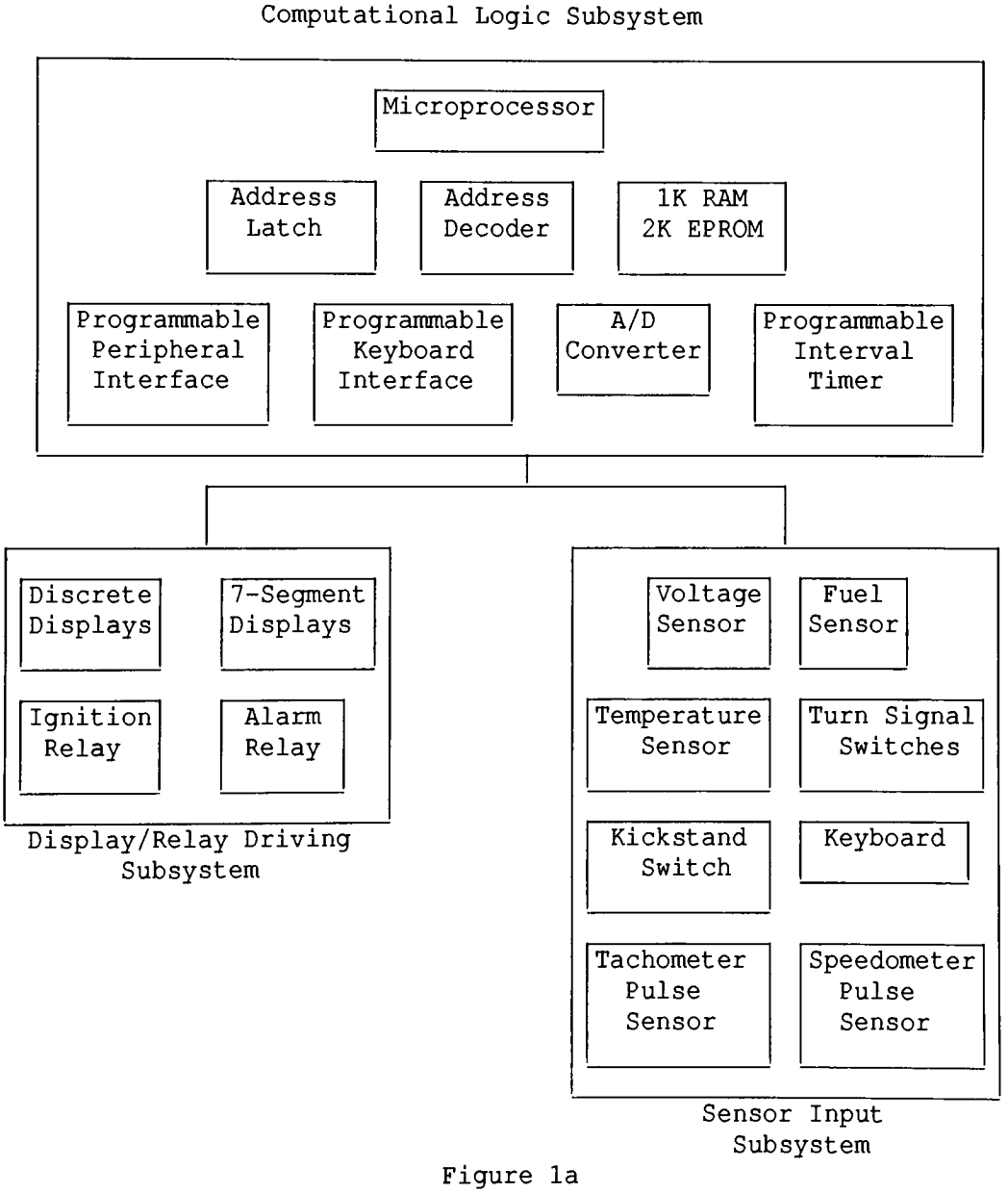


Figure 1a

Detailed System Functions

Speedometer The speedometer function measures speed derived from the number of revolutions of the front wheel in relation to a fixed time constant.

The internals of a stock analog gauge were utilized to produce the hardware interface for the speedometer function. Within the stock gauge, there was a magnetized disc which revolved in response to a cable connected to the front wheel. As the magnetized disc spun, it caused an analog pointer to register speed in response to the magnetic field which was created. The pointer hardware has been dismantled, and the disc, mounting hardware, and the cable interface retained to provide the hardware interface for this system.

The disc (which is normally shiny aluminum) was painted flat black except for strategically spaced stripes which form the reflective surfaces for activation of the FPA104 phototransistors.

An FPA104 phototransistor is placed facing the disc, at a distance of approximately one sixteenth of an inch. Functionally, as the front wheel spins, the revolutions are transferred through the cable to the disc. As the reflective stripes on the disc pass in front of the phototransistor, it causes the phototransistor to activate, and create the basic pulse which is fed through further circuitry used for counting the number of revolutions of the wheel.

The CPU periodically reads the number of pulses, and computes the speed via manipulation of the basic formula; R x T = D (Rate X Time = Distance). The result is displayed in one mile per hour resolution via three 7-segment LEDs.

Tachometer The functionality of the tachometer is nearly identical to that of the speedometer, with variations in hardware interface and pulse computation.

The stock analog gauge was identical in operation and hardware construction except that the cable interface was to the engine block rather than to the wheel. For this reason, the internals of the stock gauge were utilized in the exact manner as the tachometer except for the spacing of the reflective stripes.

The tachometer interface also uses an FPA104 phototransistor in the same manner as the speedometer, and hence uses identical signal conditioning circuitry as that of the speedometer function.

As in the speedometer function, the CPU periodically reads the number of pulses generated by the tachometer circuitry. The computation of the revolutions per minute (rpms) of the engine crank shaft is derived by relating the number of counts found to a fixed period of time. The result is displayed on two 7-segment LEDs in one revolution per minute resolution.

Engine Temperature Sensing of the engine temperature is implemented by the use of an LM335 precision temperature sensor as input to the ADC0808 A/D converter.

When powered by a +12 volt supply as in this system, the output range of the LM335 is below the maximum input range of the A/D converter (+5 volts), and no further interface circuitry is needed.

The CPU periodically polls the A/D converter channel driven by the temperature sensor, performs a Kelvin to Fahrenheit conversion, and displays the result on three 7-segment LEDs in system dependent resolution.

Battery Charging Voltage Battery charging voltage is the second function implemented using A/D conversion.

In final integration into the motorcycle electrical system, leads will be run from the battery terminals directly to the A/D converter through intermediate voltage division circuitry. Voltage division circuitry is needed to reduce the voltage range input to within the +5 volt maximum input range of the A/D converter.

The CPU will periodically poll this A/D converter channel, compute the voltage present, and display the result (to a tenth volt resolution) via three 7-segment LEDs.

Low Fuel Warning The last of three A/D converter functions comes from the fuel level sensor. Like the battery charging voltage function, the electrical interface for this function is direct from the sensor to the A/D converter via intermediate voltage conversion circuitry to bring the input voltage within the range of the A/D converter.

The CPU also periodically polls the A/D converter channel for this function and computes the level of fuel in the fuel tank.

Display of this function is via individual discrete LEDs in unique combinations according to the amount of fuel present. Three possible fuel level conditions are possible in this system: 1) Greater than one quarter tank of fuel, 2) Less than one quarter tank but greater than one eighth tank, and 3) Less than one eighth tank of fuel. No LEDs lit represent condition one, while conditions two and three result in defined combinations of LEDs being lighted accordingly.

Turn signals The turn signal function is implemented by the CPU periodically polling the status of two switches. One switch represents the left turn signal, and the other switch represents the right turn signal. If a switch closure is detected, the CPU executes the flash algorithm for the left or right turn signal.

The turn signal displays are also implemented using individual discrete LEDs. Both the left and right turn signals are displayed using rows of twenty LEDs arranged in a straight line.

The flash algorithm lights the innermost LED first, followed by the next, and so on, until the entire row of twenty LEDs are lit, then all LEDs blank, and the sequence repeats. Lighting the LEDs in this manner creates the effect of a bar of light, growing from the inside out, in the direction of the turn.

Kickstand Down Warning The kickstand down function, like the turn signal function, is implemented by polling the status of a

switch, and display of the warning is also via individual discrete LEDs. If a switch closure is detected, a unique combination of discrete LEDs light to represent the condition.

Security The security function is implemented via a hexadecimal keyboard (keys 0-9, A-F). The CPU reads the values of the keys input, in sequence, and determines whether to enable the ignition, or disable the ignition and blow the horn as a result.

The security input sequence is a five digit alphanumeric value which has been programmed to be A923B. The exact code must be entered in the exact order, for the ignition system to be enabled. The operator is allowed two attempts at correct entry of the five digit code.

In the case that the code has been correctly entered, the ignition system is enabled, and the two 7-segment LEDs used for the tachometer display flash the value "65" five times to alert the operator that a successful entry has been made, and remind him of the legal speed limit.

If the first attempt is incorrect, the LEDs used for low fuel warning will flash five times signaling incorrect entry and that one more try will be allowed. Upon successful entry on the second try, the above success sequence will be executed. If the second try is unsuccessful, the horn will blow, and the system will go into an unrecoverable (without reset) halt state.

System Architecture

Bus Structure

Internal/External Bus Structures Due to current drain on the 8085, and the fact that all components are not directly microprocessor bus compatible, large systems need some form of buffering to maintain electrical integrity. This system therefore is divided into two major bus subsections which are called the internal and external bus structures. All components deemed internal directly share the 8-bit data bus, 16-bit address bus, and control signals with the CPU. All external components are addressed by, and receive data from, the 8255 programmable peripheral device, the 8279 programmable keyboard interface device, the ADC0808 A/D converter, or the 8254 programmable interval timer, which are used as I/O buffers for CPU communications. One exception to the internal/external bus structure is the control signals from the 74LS154 address decoder to the 4511 BCD to 7-segment latch/decoder drivers.

Figure 2 shows the system organization of the internal and external bus structures. Bus loading considerations are discussed further in a the current requirements section.

Device communications within the bus structures are both one way and bi-directional. All internal devices are bi-directional except the EPROM, address latch (8212 8-bit input/output port), and address decoder (74LS154). The EPROM can only be read from, hence it's name, Electronically Programmable Read Only Memory. The address decoder and the address latch are output devices. Both follow the bus signals and

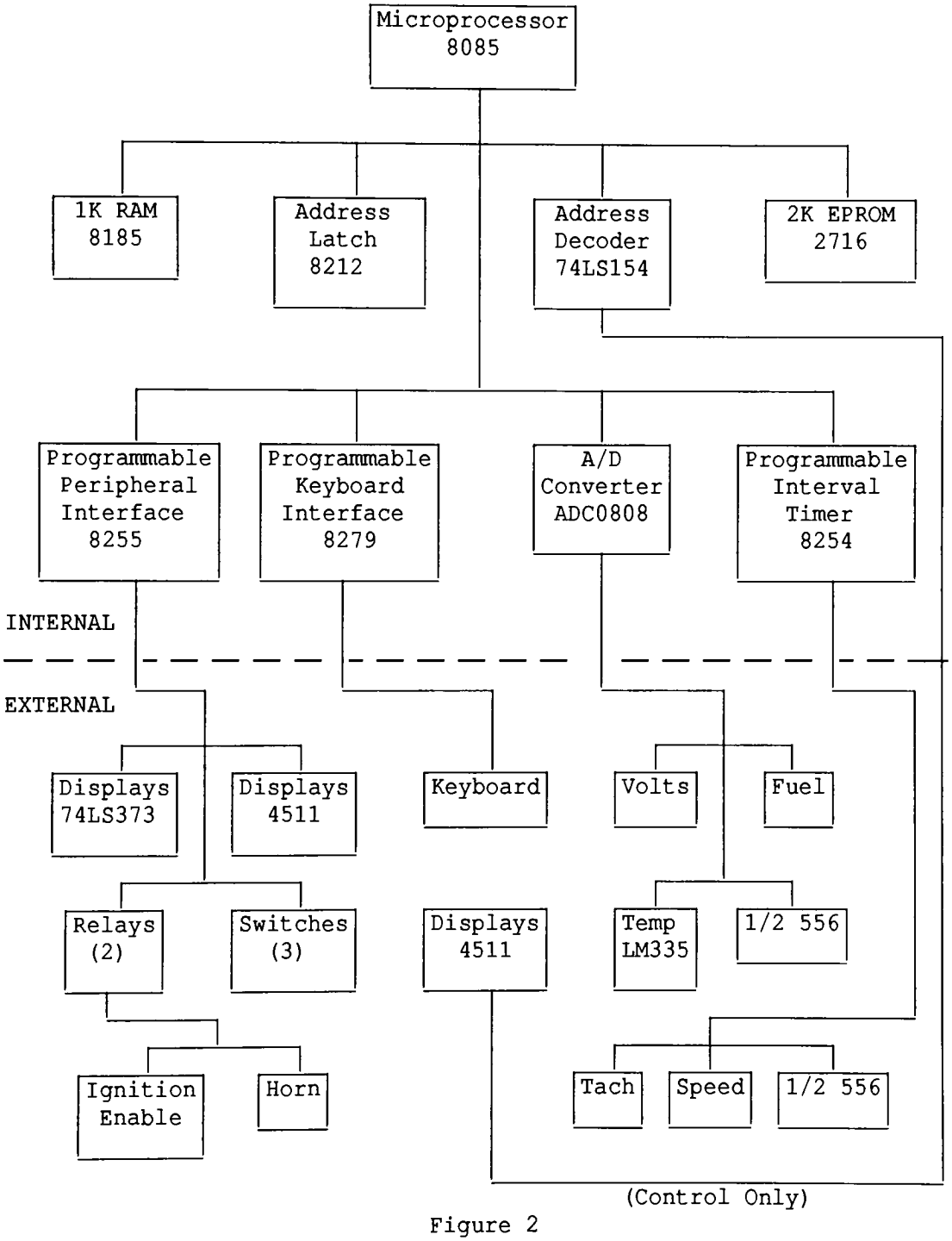

Microprocessor
8085
1K RAM
8185
Address
Latch
8212
Address
Decoder
74LS154
2K EPROM
2716
Programmable
Peripheral
Interface
8255
Programmable
Keyboard
Interface
8279
A/D
Converter
ADC0808
Programmable
Interval
Timer
8254
INTERNAL
EXTERNAL
Displays
74LS373
Displays
4511
Keyboard
Volts
Fuel
Relays
(2)
Switches
(3)
Displays
4511
Temp
LM335
1/2 556
Ignition
Enable
Horn
Tach
Speed
1/2 556
(Control Only)

Figure 2

are always active. Externally, all of the devices are used as input to the internal bus structure except for the displays and the relays, which are only used as output devices in relation to the CPU.

Display Structure

LED/Keyboard Layout The display board is used for system input/output, and contains all LEDs and the keyboard.

Figure 3 shows the layout of the display board. There are eleven 7-segment LEDs and fifty individual LEDs used for display.

The individual LEDs are packaged in groups of ten forming one unit. There are five units used for display of the turn signals and warning lights. The turn signals are comprised of two units each (left and right), and are situated at the top left and top right of the display board. The warning lights are comprised of one unit and are situated between the turn signals.

The 7-segment LEDs are used for display of the speed, tachometer, temperature, and charging voltage functions. They are arranged in three groups of three (speed, temperature, charging voltage), and one group of two (tachometer X100). The charging voltage LEDs are situated at the center left of the display board. Beneath these are the temperature LEDs. At right center of the board are the two tachometer LEDs, beneath which are the LEDs for the speed display.

The keyboard is situated in the center of the display board.

Display Component Interconnect Two types of components are used for driving the the individual and 7-segment LEDs on the display board. The 74LS373 octal latches drive the individual LEDs and

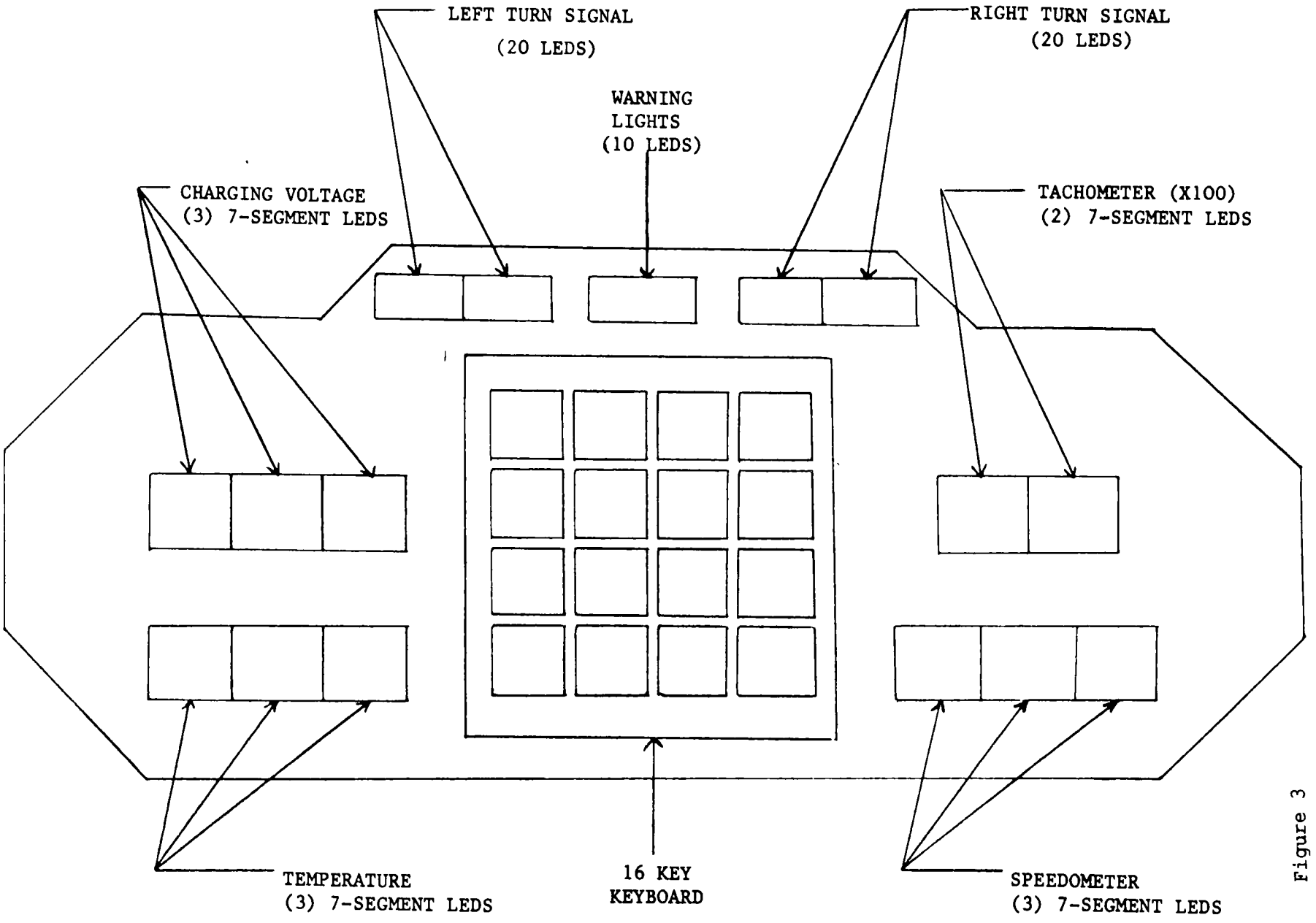

LEFT TURN SIGNAL
(20 LEDS)
RIGHT TURN SIGNAL
(20 LEDS)
WARNING
LIGHTS
(10 LEDS)
CHARGING VOLTAGE
(3) 7-SEGMENT LEDS
TACHOMETER (X100)
(2) 7-SEGMENT LEDS
TEMPERATURE
(3) 7-SEGMENT LEDS
16 KEY
KEYBOARD
SPEEDOMETER
(3) 7-SEGMENT LEDS

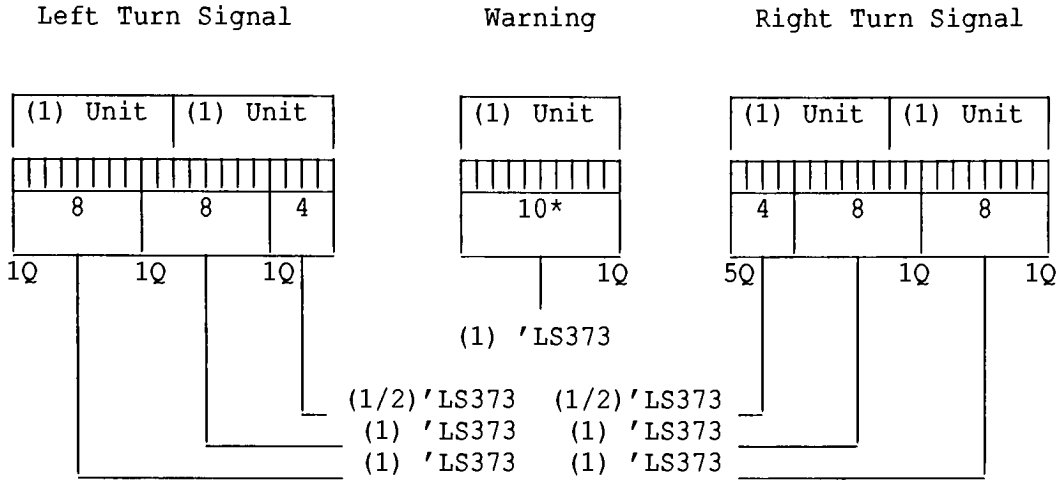
Figure 3

the 4511 BCD to 7-segment latch/decoder drivers drive the 7-segment LEDs. Both types of components have mutually exclusive control signals, but share an 8-bit data bus generated by the 8255 programmable peripheral interface.

The 74LS373 octal latches have their eight input pins (1D-8D) connected to the eight bits of the data bus (D0-D7). The eight corresponding outputs (1Q-8Q) are connected to one LED each. In this manner, each 74LS373 can drive eight individual LEDs of one ten LED unit. As stated, each turn signal uses two units, for a total of twenty individual LEDs for each signal, which is not evenly divisible by eight. Because of this correspondence, the turn signals are implemented by using two and a half 74LS373s for each turn signal for a total of five. The fifth 74LS373 uses four of it's outputs for the left turn signal, and four for the right turn signal, resulting in three 74LS373s being addressed by software for each turn signal algorithm.

A sixth 74LS373 is used to drive the warning light unit. Since there are eight 74LS373 outputs and ten LEDs, LEDs 4 and 5, and six and seven have been tied together. Figure 4 shows the configuration of the fifty individual LEDs.

BCD (Binary Coded Decimal) numbers require 4-bits to represent the entire range (numbers 0-9). Each 4511 correspondingly has four input pins which are connected to four data bus lines. Since the data bus is eight bits wide, two 4511 devices can occupy eight distinct data bus lines; one 4511 connected to data lines D0-D3, and one 4511 connected to data lines D4-D7. This interconnect method has been

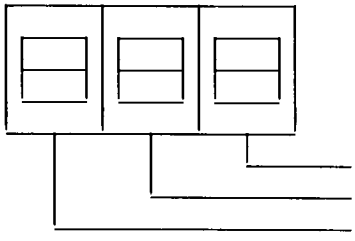


* LED'S 4 & 5 and 6 & 7 tied together

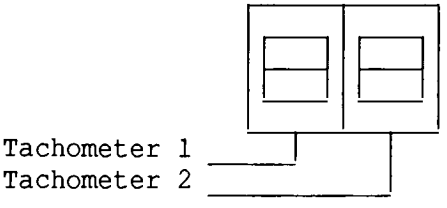
Figure 4

utilized for all of the 7-segment displays in this system to reduce software overhead. By connecting the components in this manner and tying the latch control pins of two devices together, one 8-bit word representing two BCD numbers can be written to two 7-segment displays in one address (WRITE) cycle. For the temperature, speed, and charging voltage displays (groups of three 7-segment LEDs), two writes are required to display three digits. For the tachometer (one group of two LEDs) only one write is required to display two digits. This also reduces the number of individual control signals (latch enable) from a possible total of eleven (one for each 4511), to only seven.

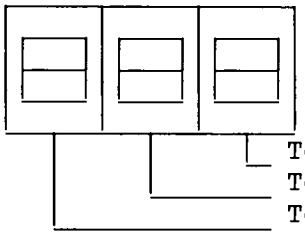
Figure 5 shows the functional configuration of the 7-segment LEDs and their interconnect to the 8-bit data bus.



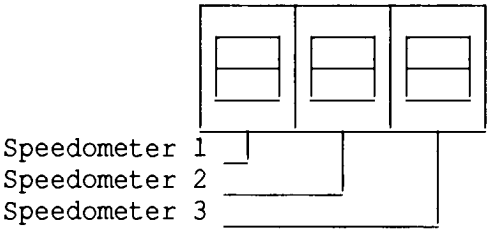
(Voltage 2 & 3 Latch Controls Tied)



(Tach 1 & 2 Latch Controls Tied)



(Temp 2 & 3 Latch Controls Tied)



(Speed 2 & 3 Latch Controls Tied)

Seven Segment LED Data Bus Connect Map

Temperature 1	Data Bus D0 - D3
Temperature 2	Data Bus D4 - D7
Temperature 3	Data Bus D0 - D3
Tachometer 1	Data Bus D4 - D7
Tachometer 2	Data Bus D0 - D3
Voltage 1	Data Bus D0 - D3
Voltage 2	Data Bus D4 - D7
Voltage 3	Data Bus D0 - D3
Speedometer 1	Dsts Bus D0 - D3
Speedometer 2	Data Bus D4 - D7
Speedometer 3	Data Bus D0 - D3

Figure 5

Address Assignments

Memory Mapped vs I/O Mapped I/O INTEL discriminates between memory mapped I/O, and I/O mapped I/O, based on the use of the IO/M pin on the 8085. This method pertains only to device selection within their family of multiplexed bus compatible chips. By convention, if the IO/M pin of the 8085 is connected to the IO/M pins of these devices, the system is said to be I/O mapped. Conversely, if the IO/M pin is not used in this manner, the system is said to be memory mapped. This system uses only one multiplexed bus compatible device; the 8185 RAM. The IO/M pin of the 8085 is used in conjunction with the 8185, but only as input to one of it's chip enable pins. Therefore, by definition, this system is said to be a memory mapped I/O system.

One drawback of configuring a system as an I/O mapped system, is that the I/O instructions are then limited to IN and OUT instructions. By configuring the system as a memory mapped system, the IN and OUT instructions can be used in addition to several move, load and store, arithmetic, and logical instructions which operate on memory locations.

The timing of the READ and WRITE cycles for memory and I/O reads and writes are identical. The difference, as stated previously, is in what appears on the bus during these cycles. For memory reads and writes, the 16-bit address value appears on address lines AD0-A15, for I/O reads and writes, the 8-bit port address is duplicated on address

lines AD0-AD7 and A8-A15. This must be taken into consideration for address decoding purposes to avoid simultaneous device selection.

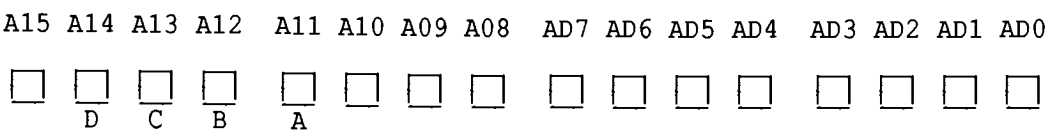
Address Decoding Part of the most significant byte of the address bus (A11-A14) is used for device decoding in this system. Address lines A11-A14 are used as input to the four input pins of the 74LS154 4 to 16 binary decoder (A-D). Address line A15 is left unused and is not connected to any of the system devices.

Each device in the circuit needs at least two input signals from the CPU for selection; either a read or write signal, and/or one or more chip select signals (also called chip enable/device enable/device selects). By decoding four binary inputs (A11-A14) the 74LS154 can provide one of sixteen possible chip select signals, which will enable any particular device in the system according to the value on those address lines. Like any other device in the system, the 74LS154 needs a chip select signal for operation. This is achieved by gating the READ and WRITE signals via the 74LS08 quad 2-input AND device and applying the output to the G1 (chip select) pin of the 74LS154. Gating the signals in this manner not only provides the 74LS154 chip select signal but also synchronizes the device with the 8085 READ/WRITE control signals.

Refer to figure 6 and schematic sheet 2 for the 74LS154 interconnect.

Figure 7 shows the resulting chip select map for this system. Thirteen of the possible sixteen outputs of the 74LS154 are utilized in this system. The first six outputs (Y0-Y5) are used for selection

16 Bit Address Bus



(D, C, B, A are connections to 74LS154)

Figure 6

of the six major internal bus devices. The last seven outputs (Y6-Y12) are used as chip selects for the seven 4511 BCD to 7-segment latch/ decoder driver devices.

CS0 ..... 2716 EPROM

CS1 ..... 8185 RAM

CS2 ..... 8255 Programmable Peripheral Interface

CS3 ..... 8279 Programmable Keyboard Interface

CS4 ..... 8254 Programmable Interval Timer

CS5 ..... ADC0808 8 Channel A/D Converter

CS6 ..... 7-segment Display (Temperature 1

CS7 ..... 7-segment Display (Temperature 2 & 3)

CS8 ..... 7-segment Display (Voltage 1)

CS9 ..... 7-segment Display (Voltage 2 & 3)

CS10..... 7-segment Display (Speedometer 1)

CS11 ..... 7-segment Display (Speedometer 2 & 3)

CS12 ..... 7-segmant Display (Tachometer 1 & 2)

Figure 7

System Memory Map By decoding address lines A11-A14 in this manner, each time that an address is generated which enables a new 74LS154 output signal, that address will fall on a 2k word boundary. This effectively partitions system memory into a series of 2k word areas which are reserved for the various system components. Since A15 is left unconnected, it is removed from the system for possible address decoding/generation purposes, and cuts the addressable memory space to 32k. The resulting 32k system memory map is shown in figure 8.

System I/O Port Map Although there are 2k words of reserved space for each device in the system memory map, there are only two memory devices used in this system; the 8185 RAM and the 2716 EPROM, and are accessed as such. Because of the nature of the other devices, they are accessed not as memory devices, but as a series of I/O ports.

These devices are addressed by using the 8085 IN and OUT (PORT#) instructions, which cause a duplicate of the port address to appear on both address lines AD0-AD7, and A8-A15 simultaneously. In conjunction with the binary value that enables a particular 74LS154 chip select output, some of the system components (8254, 8255, and 8279) utilize values on the A0 and or A1 address lines to discriminate between unique ports within themselves. These A0 and or A1 values combined with the device select values, together, make up the unique 8-bit port address(s) of the rest of the devices in the system.

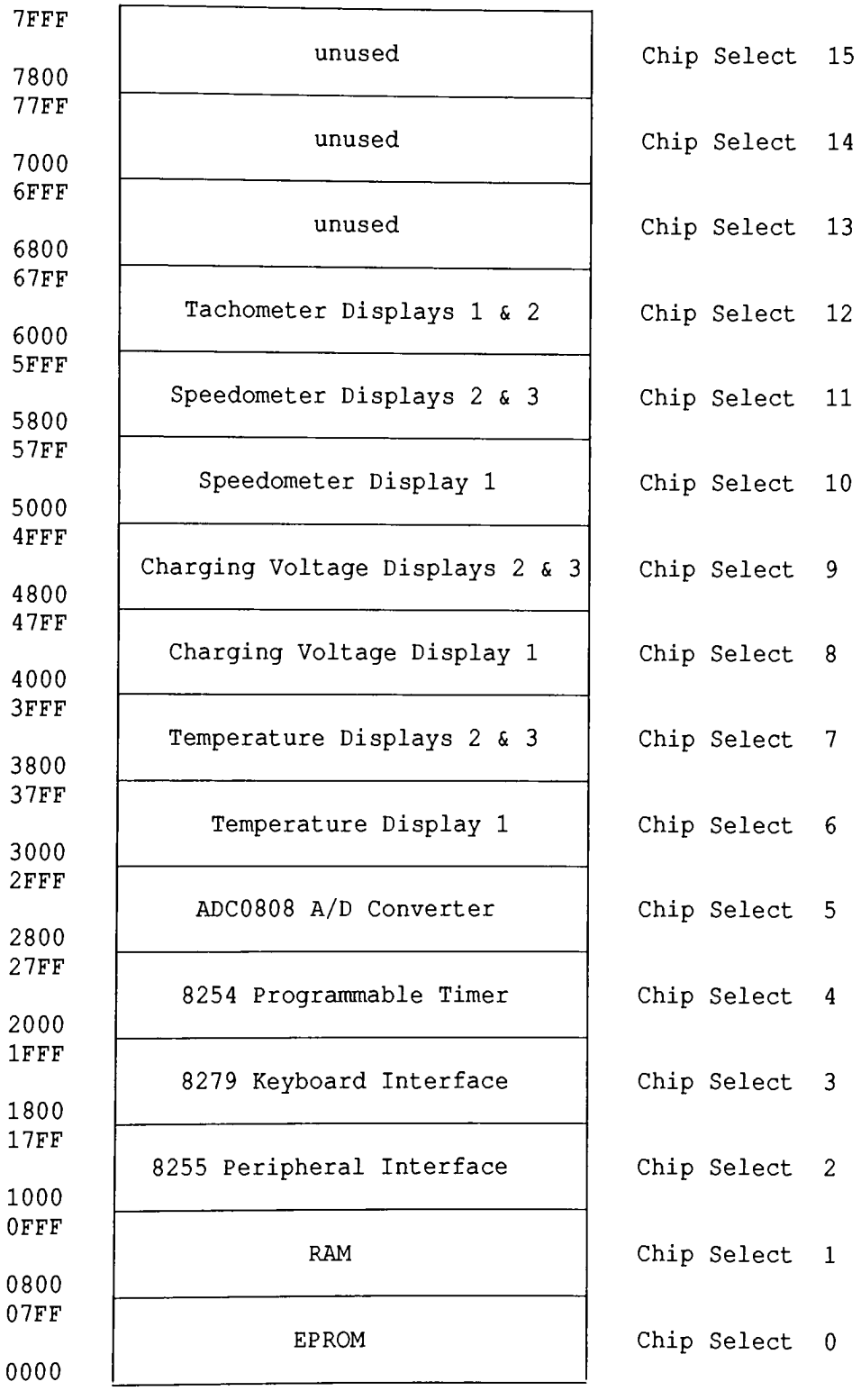

7FFF
7800
77FF
7000
6FFF
6800
67FF
6000
5FFF
5800
57FF
5000
4FFF
4800
47FF
4000
3FFF
3800
37FF
3000
2FFF
2800
27FF
2000
1FFF
1800
17FF
1000
0FFF
0800
07FF
0000
unused
unused
unused
Tachometer Displays 1 & 2
Speedometer Displays 2 & 3
Speedometer Display 1
Charging Voltage Displays 2 & 3
Charging Voltage Display 1
Temperature Displays 2 & 3
Temperature Display 1
ADC0808 A/D Converter
8254 Programmable Timer
8279 Keyboard Interface
8255 Peripheral Interface
RAM
EPROM
Chip Select 15
Chip Select 14
Chip Select 13
Chip Select 12
Chip Select 11
Chip Select 10
Chip Select 9
Chip Select 8
Chip Select 7
Chip Select 6
Chip Select 5
Chip Select 4
Chip Select 3
Chip Select 2
Chip Select 1
Chip Select 0

Figure 8

Figure 9 is the resulting system I/O port address map.

The far right column lists the usage of the A0 or A0 and A1 pins of the particular devices for the corresponding I/O READ or WRITE operations. The last eight I/O port addresses are for devices that do not employ A0 or A1 pins. The A0 and A1 values (AD0 and AD1 bus lines) are therefore zero by default.

Function	I/O Port	A0/A1
8255 Command Register	13	A0,A1=1
8255 Port A Read/Write	10	A0,A1=0
8255 Port B Read Write	11	A0=1, A1=0
8255 Port C Read Write	12	A0=0, A1=1
8254 Command Register	23	A0,A1=1
8254 Counter 0 Read/Load	20	A0,A1=0
8254 Counter 1 Read Load	21	A0=1,A1=0
8254 Counter 2 Read/Load	22	A0=0,A1=1
8279 Command Register	19	A0=1
8279 Read/Write	18	A0=0
ADC0808 Read/Start	28	Unused (Default = 0)
Temperature Display 1	30	Unused (Default = 0)
Temperature Displays 2 & 3	38	Unused (Default = 0)
Charging Voltage Display 1	40	Unused (Default = 0)
Charging Voltage Displays 2 & 3	48	Unused (Default = 0)
Speedometer Display 1	50	Unused (Default = 0)
Speedometer Displays 2 & 3	58	Unused (Default = 0)
Tachometer Displays 1 & 2	60	Unused (Default = 0)

Figure 9

Keyboard Map and Data Word The keyboard in this system is a plastic membrane hexadecimal device configured as a four row by four column matrix. The 8279 programmable keyboard interface device is capable of interface to a sixty four key keyboard if all of the scan and return line pins are used in conjunction with a 3 to 8 decoder. In this system, the four scan lines were utilized, but only four of the return lines were connected to the keyboard, effectively turning it into a four-by-four matrix interface device.

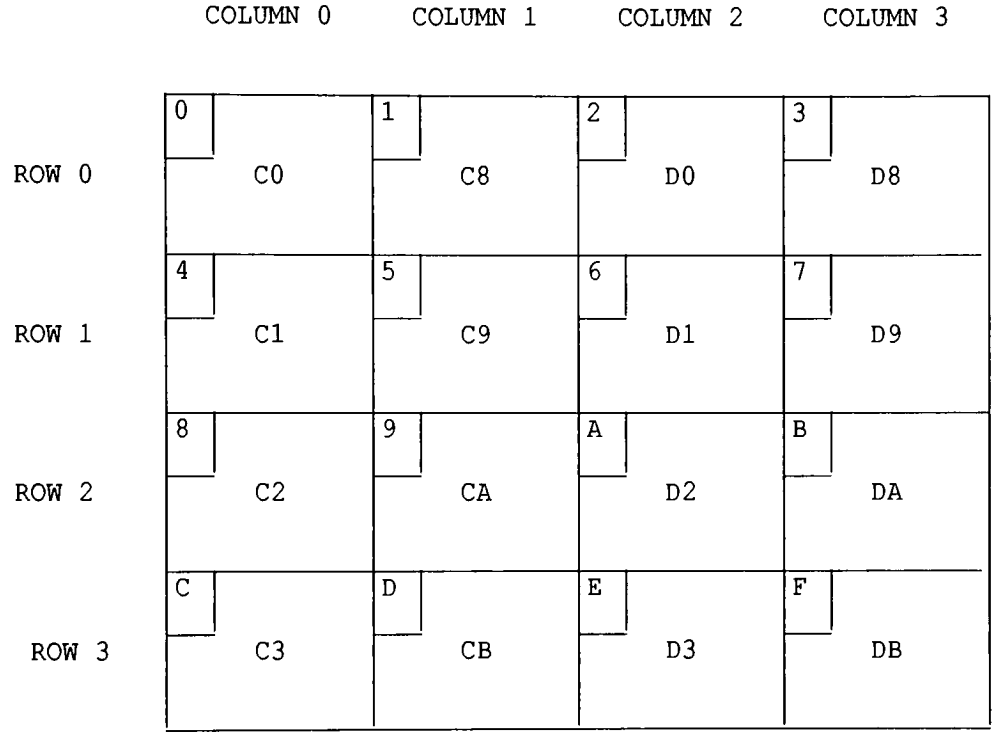
When a key closure is sensed, as well as entering the key into it's RAM, and outputting an interrupt signal, the 8279 enters the value of the row-column intersection into a data word. This data word can then be read by the CPU, the individual key closure decoded, and then be manipulated by the application program. The data word consists of 8-bits which uses 3-bits for representation of the rows, 3-bits for the columns, and 2-bits to represent whether or not the shift and or control keys have been closed.

The hexadecimal values representing the row-column intersections of the keyboard returned by the 8279 in this system form the keyboard map. The keyboard map and data word format are shown in figure 10. The smaller boxes in the keyboard map are the actual key markings on the hexadecimal keyboard.

Circuit Board Interconnect

System Connector Map There are three individual circuit boards that comprise the total system. These boards are

8279-5 Keyboard Map



8279-5 Data Word Format

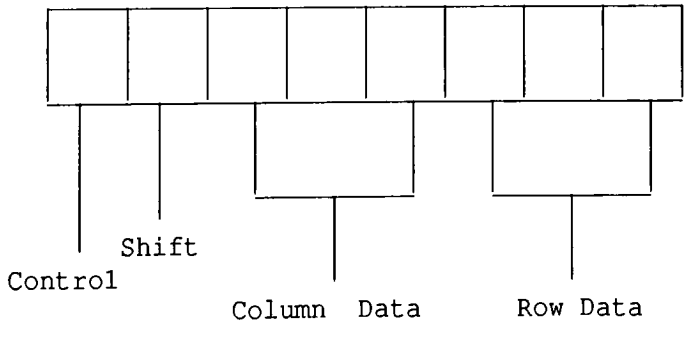


Figure 10

stacked one above each other in a "sandwich" fashion, separated by one inch spacers.

The top circuit board contains all of the LED displays, the keyboard, and the associated LED driving circuitry.

The center board contains the CPU, EPROM, RAM, 8255, address latch, 8279, address decoder, 8254, ADC0808, 556 timer, and interface circuitry for A/D conversion, speed and tachometer, ignition enable, and alarm functions.

The bottom circuit board contains the voltage regulator, noise filtering circuitry, calibration and voltage division circuitry, and miscellaneous interface circuitry for connection to the actual motorcycle subsystems.

There are nine connectors used to link the three boards together and to the outside world. The top board contains connectors J1 and J2 which are fifteen and fourteen pin straight-inline-packages respectively. Also on the top board is J3, which is a ten pin right angle header connector used to supply the power and ground to the display board.

On the second board are connectors J4, J5, and J6. J4 and J5 mate with the connectors J1 and J2 from the top board. Connector J6 is used to bring both signals and power and ground from the bottom board, and is a twenty pin dual-inline-package.

Connectors J7-J9 reside on the bottom board. Connector J7 mates with J6 on the second board, while J8 supplies power and ground to J3 on the top board. Connector J9 is a twenty four pin dual-inline-package, and is used to bring in the signals from the motorcycle sub-

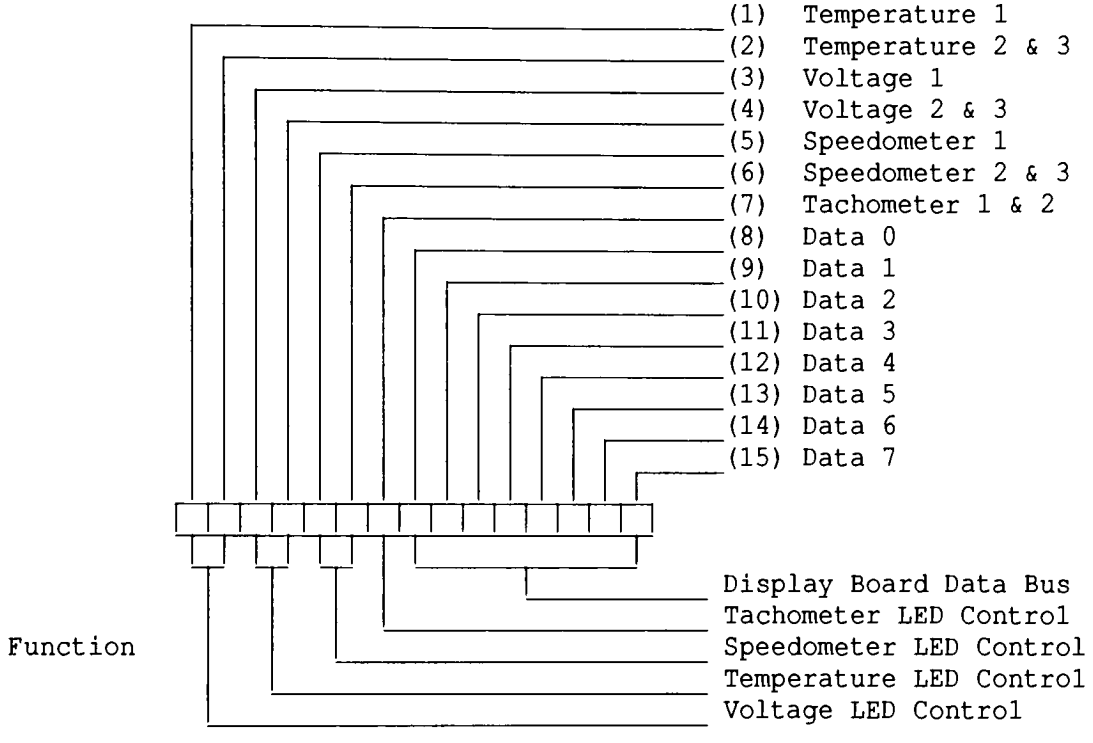
systems. Figure 11 shows the connector pinout map for connectors J1/J4, and J2/J5. Figure 12 shows the connector pinout map for connectors J3/J8, and J6/J7. Figure 13 shows the connector pinout map for connector J9. Connectors are shown in pairs since both ends of a mating pair of connectors have the same pinout in this system. All dual-in-line package connectors in this system are connected to each other via ribbon cable.

TTL/CMOS Interfacing

Voltage Levels All of the integrated circuits in this system are from the TTL family except the ADC0808 A/D converter. When designing within a family of devices (all TTL or all CMOS) or between families (TTL/CMOS), special attention must be given to the current requirements and voltage levels utilized by those devices.

Attention to voltage levels within the same family of devices is not as critical as when mixing families, since all devices within the same family will have compatible voltages, provided the ability of those devices to supply the voltage has not been compromised. Typically, TTL devices operate from a minimum supply voltage of 4.5 volts to a maximum of 5.5 volts, with a nominal supply voltage of 5 volts. A high level signal (logic 1) is in the range from a minimum 2 volts to a maximum equal to the supply voltage with a nominal value of 3.4 volts, while a low level signal (logic 0) is a minimum 0 volts to a maximum of 0.8 volts. The range between 0.8 volts to 2 volts is unrecognized by the TTL devices.

Connector J1/J4



Connector J2/J5

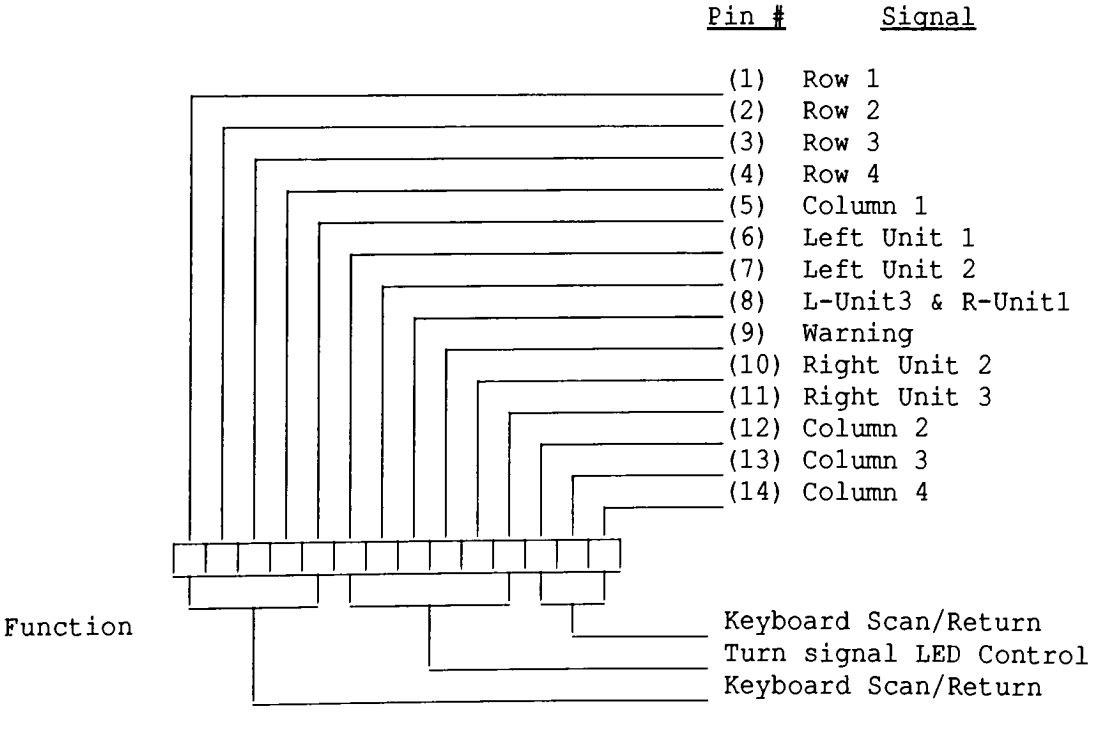
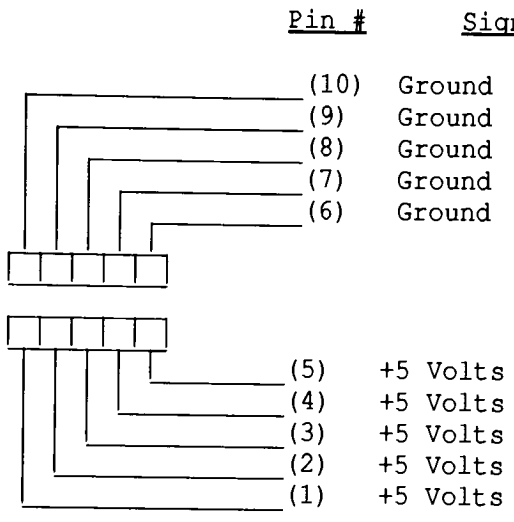


Figure 11

Connector J3/J8



Connector J6/J7

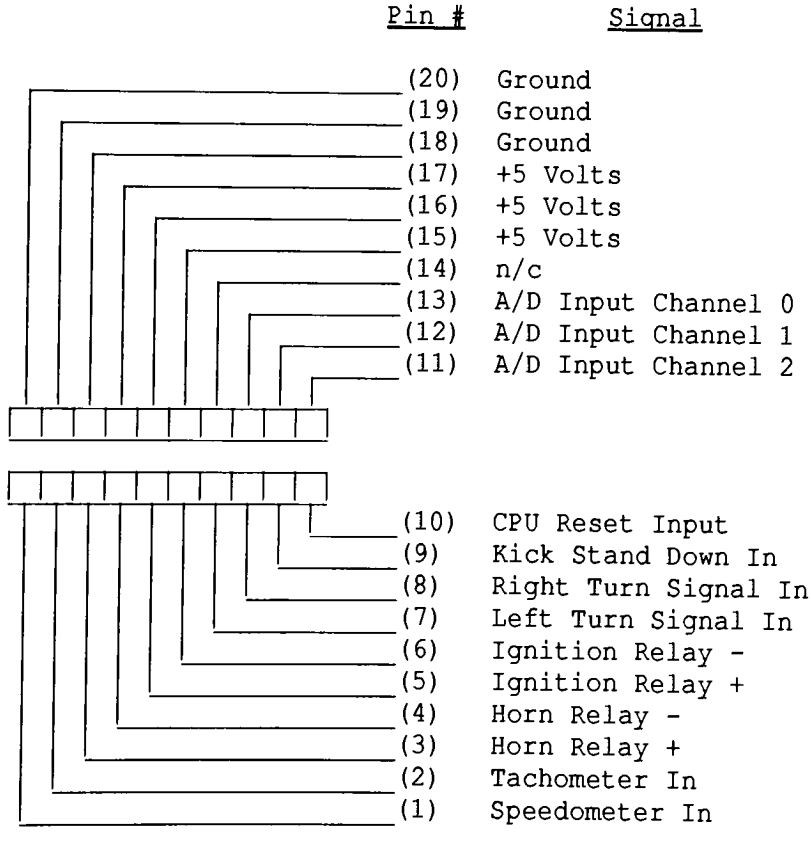


Figure 12

Connector J9

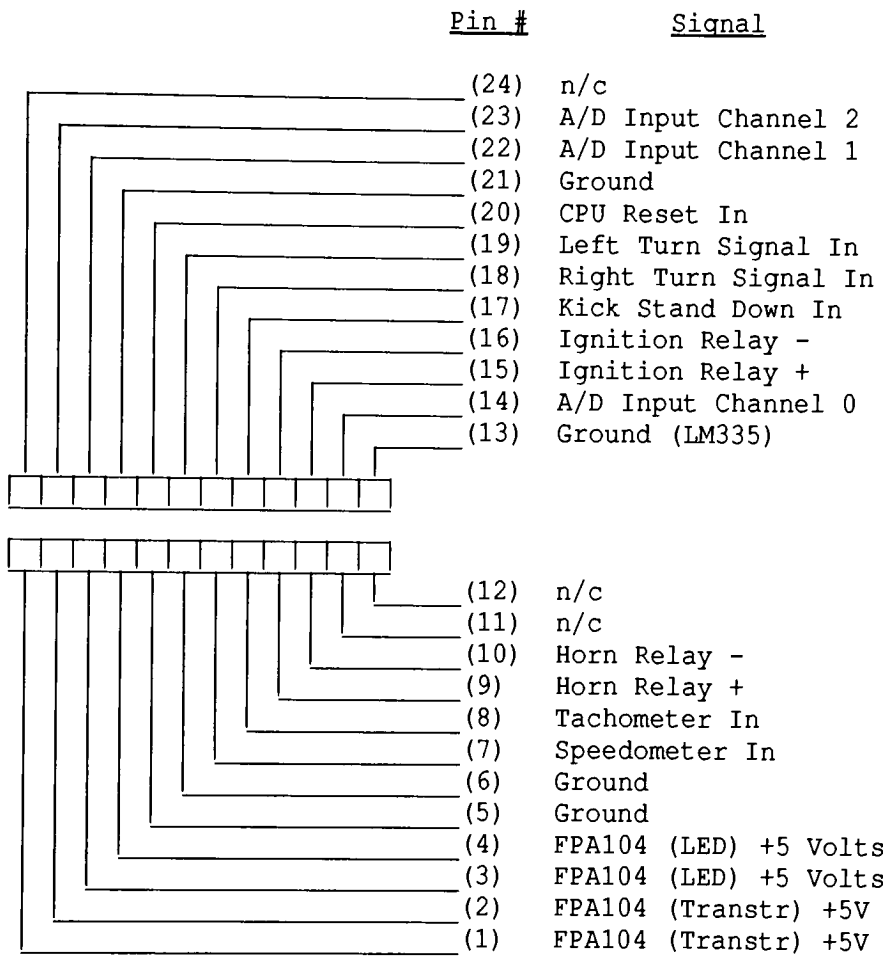


Figure 13

CMOS devices operate over a larger supply voltage range, from a minimum of 3 volts to a maximum of 15 volts. High level output voltage and low level input voltage (logic 1/logic 0) vary according to the supply voltage used (this circuit operates from a 5 volt supply, so the following CMOS parameters are based on that voltage level). A high level signal is in the range of a minimum 3.5 volts to a maximum equal to the supply voltage. A low level signal is in the range of 0 volts

to a maximum of 1.5 volts. The range between 1.5 volts and 3.5 volts is unrecognized by CMOS devices. Given the above parameters for the two families of devices, one can see that the following general "rule of thumb" is applicable: "with a supply voltage of 5 volts, TTL can drive CMOS devices, but CMOS devices cannot drive TTL without additional interface circuitry".

The ADC0808 is a specially designed CMOS device with outputs being TTL compatible for direct interface to microprocessor busses. Conversely, the inputs to the ADC0808 must be run at CMOS levels.

All input signals to the ADC0808 in this circuit are derived from TTL family devices. Although the above rule of thumb states that this is an allowable condition, special attention was given to these inputs to insure integrity.

The critical parameter when driving CMOS by TTL is the high level input voltage to the CMOS device. As outlined above, The high level input voltage for CMOS at 5 volts is a minimum 3.5 volts, while the high level output voltage from the TTL device operates at a nominal 3.4 volts. The four TTL sourced input signals to the ADC0808 in this circuit were further conditioned by adding 5k pull-up resistors. These resistors were added to the CLK, ALE, START, and OE pins to insure that the high level input voltage would be "pulled up" to nearly 5 volts for an added margin of safety.

Fanout As with voltage levels, the TTL and CMOS families differ in respect to fanout capabilities. Fanout is defined as the number of inputs a single particular component or gate can drive.

Fanout discussions are generally based on current source and sink capabilities of a particular device or gate within a device. Source current applies to the level of current a device or gate can generate as an output. Sink current applies to the level of current a device or gate can accept as input from other gates or devices.

With TTL, the source current is typically 1.6 mA per gate, with a sink current of 16 mA per gate. In this example, one TTL gate can drive 10 TTL inputs at a logic 0, for a fanout of ten.

CMOS devices are relatively low power devices, with low power source current abilities. The current sink ability is conversely high, and allows a CMOS fanout of as high as 50.

Fanout was not a major area of consideration in this system. Only in the interface circuitry to the A/D converter are multiple inputs to a particular component pin utilized. The source of these signals are from the 8255, and the inputs are to TTL devices which can handle a fanout of ten. Several signals from the 8085 terminate at multiple pins of other devices within the system such as the data, READ, WRITE, and control signals. Although they terminate at multiple locations, no more than the permissible number of devices at these terminations are active (accepting these signals) at any one time insuring electrical integrity. The 8085 bus loading specifications are discussed at more detail in the current requirement section.

Filtering The gates within TTL devices are inherently noisy due to the operation characteristics of the transistors that make up the device. When the transistors within a device turn off and

on (changing logic states of the outputs) they create d-c fluctuations and current transients.

Several methods for minimizing this noise have been accepted, one being the method used in this system: Distribute decoupling capacitors throughout the system between VCC (+5 volts) and ground.

The general rule of thumb when using TTL is to provide one .01 microfarad capacitor per gate, or at least one .1 microfarad capacitor per twenty gates. This capacitance may be lumped in one area of the circuit board, but is most effective if it is distributed throughout. Ceramic disk capacitors are usually the choice for this type of filtering, and have been used in this system.

Capacitors are inexpensive and were not a factor in board component density, so a decoupling "overkill" was used in this system. One .1 microfarad capacitor was used per TTL and CMOS integrated circuit (IC) throughout most of the system regardless of the number of gates within the IC, which were most often less than twenty. Each of the capacitors were placed as close as possible to each IC thus effectively distributing the capacitance throughout the entire system.

Multiple Voltage Handling

Voltage Division It is often desirable to provide multiple voltages within a circuit derived from one power source. Voltage division is an economical means to achieve this situation by "dividing" the main voltage level into one or more different voltage levels.

Also, as in this system, it is often desirable to divide higher voltage levels down to specific working ranges of various components. All input voltage levels from the motorcycle are +12 volts, while the working voltages of the components that interface with these input signals are +5 volts or lower. The left and right turn signal, kick stand down, fuel level, and charging voltage inputs have been conditioned by voltage division circuitry to provide a +5 volts or less input range.

Voltage division is implemented by the following formula;

$$Vout = (Rout/Rtotal) \times Vsupply$$

where Vout is the voltage output between any two points of a voltage divider, Rout is the series resistance of that point, Rtotal is the total series resistance of the circuit, and Vsupply is the supply voltage level. More simply put, the voltage between any two points of a voltage divider equals the ratio of the resistance of these two points times the supply voltage.

The voltage dividers for the left and right turn signals and the kick stand down inputs were implemented using fixed 1k and 2k resistors. Substituting into the above formula, we have;

$$Vout = (1K/(1K+2K) \times 12 \text{ Volts}$$

Simplifying:

$$Vout = (1K/3K) \times 12$$

$$Vout = 4 \text{ Volts}$$

This provides a +4 volt maximum input to the interface devices which is used a high level (logic 1) input, and is within the TTL logic 1 voltage range.

Voltage division is implemented in a slightly different manner for the A/D conversion interfaces and will be discussed in greater detail in the interface specification section.

Current Requirements

Bus Current Requirements Two bus current parameters must be acknowledged when determining the bus loading capacities of a microprocessor system; d-c loading, and capacitive loading.

D-c loading refers to the source and sink capabilities of the particular components as discussed in the fanout section.

Capacitive loading refers to the line capacitance seen by any one output as it sources an output signal to the rest of the devices in the system.

In respect to d-c loading, the 8085 can source four hundred microamps and sink twenty milliamps. This allows a fanout of five LSTTL devices or one TTL and one LSTTL device at a logic 0 voltage level.

Capacitive loading is critical to maintaining the timing parameters of the microprocessor system. Even though the d-c loading may not be exceeded, the capacitance of the system may be. As a signal is output and travels down the signal line to each of the active devices on that line, each of the devices is seen as a capacitance by the signal source. Each device that the signal encounters must be charged by the source signal for operation. Electronically, the signal is delayed at each capacitance until charging is complete and is then fed to the next destination. If there are many devices on a signal

line, the time for the signal to arrive at the last device may be longer than the timing specifications of the source signal, causing degradation of system timing.

The timing specifications of the 8085 are guaranteed as long as the capacitance of the line is one hundred fifty picofarads or less. If this specification is exceeded, the 8085 can handle a capacitance of three hundred picofarads with a degradation of timing of .13 nanoseconds per picofarad. Conversely, if the capacitance is held below one hundred fifty picofarads, the timing will be increased by .1 nanoseconds per microfarad.

Capacitive loading pertains to a large fanout of signals, found primarily when using CMOS devices. The 8085 can drive up to forty CMOS devices from one signal. As this system is primarily of the TTL family, capacitive loading was not a critical factor.

The d-c loading of the TTL family was a critical factor in this system and caused the addition of buffering to insure integrity and forced the creation of the internal and external bus structures.

None of the shared signals within the internal bus structure violate the loading specifications of the 8085. There are however, not enough output signals present on the 8085 to avoid adding additional address decoding circuitry to drive the external bus components and the display board. For this purpose the 8255 programmable peripheral device was added to the system which provided the additional number of signals and drive current needed to interface the external bus components.

System Current Requirements The current draw in this system will vary greatly depending on many factors. Factors such as the state of the output pins of system devices (1 or 0), how many outputs are active, or whether or not devices are three-stated will cause dynamic fluctuations in current draw according to the state of the system activity at any one time.

The largest current draw in this system is from the LED displays. The number of total LEDs lit (both individual and 7-segment) and the number of 7-segment LED segments lit contribute to LED display current draw.

Because of the dynamic nature of the current draw whithin the system, an attempt was made to define the maximum current requirements by lighting all individual LEDs, displaying the number eight on all 7-segment LEDs, and causing CPU activity at the same time.

The current draw at this level of activity was 2.7 amps, .3 amps below the maximum output of the 3 amp voltage requlator. There is a margin of safety present with the 2.7 amp requirement, since at no time during normal operation will all the LEDs be lit, nor will the 7-segment LEDs all display the number eight.

Interrupt Structure

Vectored Interrupts As outlined in the 8085 component section, there are five available hardware interrupts on the 8085; INTR, TRAP, RST 5.5, RST 6.5, and RST 7.5. The last three of these interrupts are utilized in this system, with the priority being highest for RST 7.5 and lowest for RST 5.5.

In addition to the priority of these interrupts, they differ in two other major functional areas; their trigger mechanisms, and their hardware branch vector locations.

Interrupt RST 7.5 utilizes a flip-flop trigger mechanism and is edge sensitive. This interrupt is set the instant it receives the rising edge of a low to high transition signal. The flip-flop will remain set until the 8085 takes one of three possible actions: 1) It responds to the interrupt and clears the flip-flop internally, 2) It receives an external RESET IN signal, or 3) It clears the flip-flop by issuance of the SIM instruction.

The RST 6.5 and RST 5.5 interrupts do not utilize a flip-flop, and must be maintained at a high level until sampled. Since there is no flip-flop, there is no clearing that needs to take place. When the high level signal is removed, the acknowledgement of an interrupt is also terminated by the 8085.

The 8085 samples interrupts on the descending edge of CLK one cycle prior to completion of the instruction in progress when the interrupt was received, and must be valid for at least one hundred fifty nanoseconds prior to sampling.

Each of the interrupts has a predefined vector address in ROM that is branched to upon acknowledgement of a valid interrupt. Interrupt RST 7.5 branches to location 3CH, RST 6.5 branches to 34H, and RST 5.5 branches to 2CH.

Each branch is handled in the same manner internally as an execution of a software CALL instruction. When an interrupt is acknowledged, the 8085 places the present contents of the program counter on the stack, reloads the program counter with the address of the interrupt vector, and proceeds with execution at the new location. It is imperitive that the software routine that is branched to by a response to an interrupt include an RET (return) instruction as the last command, or the addresses pushed on the stack will never be popped, and may eventually cause stack overflow problems as a result.

There are a limited number of memory locations available at the vector branch locations before running into other reserved locations for both software and hardware branchs. Depending on the application, it is usually impossible to begin service routine code at these locations. Because of the limited, shared, vector locations, the method most usually employed is to place a three byte jump (JMP) instruction at these locations which transfers control to the service routines elsewhere in RAM or ROM.

Figure 14 shows the hardware and software RST branch locations.

In this system, interrupt RST 7.5 is connected to the 8254 interval timer used in the speed and tachometer interface, RST 6.5 is connected to the ADC0808 A/D converter, and RST 5.5 is connected to the 8279 keyboard interface device. An interrupt from the 8254, or the 8279, causes an immediate branch to their respective vector locations through enabling by the SIM (Set Interrupt Mask) instruction. The RST

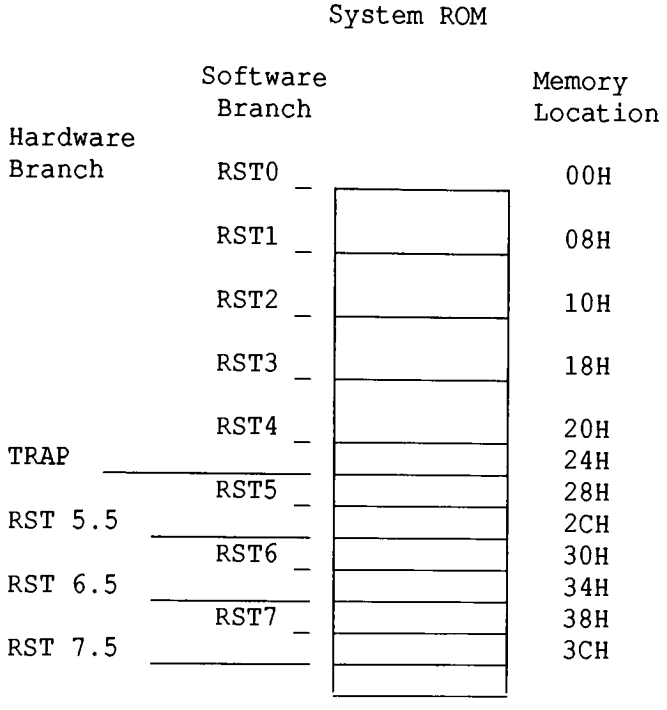


Figure 14

6.5 interrupt is polled (not enabled via SIM) and does not result in a branch to it's respective vector.

Polled Interrupt Mask The 8085 returns information on the status of all interrupts through the issuance of the RIM (Read Interrupt Mask) command. Bits 0-2 of the 8-bit status word represent which of the interrupt masks have been set by prior use of the SIM (Set Interrupt Mask) command, bit 3 shows whether or not the interrupts have been enabled via the EI (Enable Interrupts) command, bits 4-6 show the status of pending interrupts, and bit 7 shows the status of the SID (Serial Input Data) pin.

Bits 4-6 are important for implementation of polled interrupts. A 1 in any of these bit locations represents a hardware interrupt that has been received but not serviced. By masking out a particular

interrupt by the use of the SIM instruction, and checking the status of that particular interrupt's pin via the RIM instruction, the interrupt may be acknowledged at any time via software polling rather than through an immediate hardware branch.

In this system, only the status of RST 6.5 is polled. The system monitor periodically checks the status of the interrupt by reading the interrupt mask. The 8-bit word is ANDed with a hex 20 to zero out all but bit 5. The software then compares the resulting word to a hex 20 and acts upon the result. The 8085 compare command performs a subtract of the two values, so if the result is zero, the interrupt was pending, and control branches to a service routine.

Polled interrupts give the designer the flexibility of timing, priority, and choice, in resonding to an interrupt through software control.

The decision in this system of whether or not to use polled interrupts or hardware interrupts was based on timing and need. The conversion time of the A/D converter is very fast, but it is measuring slowly changing analog signals. Had RST 6.5 been configured as a hardware interrupt, the system monitor would spend most of it's time responding to interrupts from the A/D converter, when in reality, the slowly changing signals being measured need only to be checked on a much slower basis. Conversely, the signals from the speed and tachometer interface must be checked in real time due to the high rate of change of these signals, to avoid grosely erroneous values. For this purpose the RST 7.5 interrupt was configured as a hardware serviced interrupt.

Device Interfaces

In addition to subtopic discussions, each major heading within the interface specification section will include a high level, comprehensive discription of the interface circuit from input to output. All references to particular components or circuitry deal with schematic sheets 1-4 found on the last four pages of this document.

Display Interface The display board contains all 7-segment and individual LEDs, 74LS373 octal latches (used to drive the individual LEDs, the 4511 BCD latch/decoder drivers, current limiting resistors for each LED, various capacitors, and a keyboard.

Signals to the display portion of the display board originate from the internal bus structure via the 8255 programmable peripheral device, and the 74LS154 address decoder.

Control signals for the 74LS373 devices reach the display board via connector J5, while the data bus signals and control signals for the 4511 devices are interfaced via connector J4.

The 8-bit bus for the display devices is implemented through the 8255 programmable peripheral device as well as device select control signals for the 74LS373 devices. Device select control signals for the 4511 devices originate from the 74LS154 address decoder attached to the internal bus.

Each individual LED and each segment of each 7-segment display has one 93 ohm resister in series (used as a current limiter) for maximum brightness.

Each latch enable signal (pin 11) of each 74LS373 device contains one .1 microfarad capacitor to filter noise transients.

Operation of the display interface is quite straightforward. An 8-bit data value is received by the 8255 via the CPU and output to the display board creating the 8-bit display bus word. A control signal to latch the value into a particular display device is then provided via the 8255 or the 74LS154.

8255 Device Latch Control Logic Bus loading necessitated that the 4511 BCD to 7-segment latch/decoder drivers and the 74LS373 octal latches be interfaced through the 8255 programmable peripheral interface for both data and control (data only for the 4511s). Software development necessitated that these devices be configured as though they were connected to an 8-bit data bus and driven by the micrpoprocessor control signals. The result, is a software controlled simulation of microprocessor bus and control signals.

The timing sequence of display bus operations is slightly different than the multiplexed bus timing of the 8085. With the 8085, the data word appears after the address. Since the 74LS154 address decoder is attached to the upper bits of the address bus (A11-A14), the device that is being written to is already selected by the time data appears on the bus. In the data write sequence of the display board, the data word is placed on the display data bus first, then the select clock signals for the particular device being written to are initiated. Since data appearance and clock operations must happen

sequentially via the 8255 on the display board rather than concurrently as on the 8085 bus, this method was chosen for ease of software implementation.

When producing device select clock signals via software control, one must be aware of the type of clock pulse required to activate a particular device. The 74LS373 outputs will follow the inputs when the latch enable pin (11) is brought high. The data value is latched when the latch enable pin is brought low. Due to these requirements, the normal state of the latch enable pin signals via the 8255 is low. The entire write cycle to a 74LS373 device on the display board therefore is as follows: 1) Put the data value on the display board data bus, 2) Bring the latch enable signal high, and 3) Bring the latch enable pin back low to latch the value.

Speedometer/Tachometer Interface The FPA104 phototran- sistors are powered from the third circuit board via the LM323k 5 volt voltage regulator. Power is supplied to the diode sections via pins three and four of connector J9, and to the transistor sections via pins one and two of connector J9. Connector pins five and six of J9 are the ground returns for the diodes.

The input signals from the FPA104s originate in through pins seven and eight of connector J9 and pass to the second circuit board via pins one and two of connectors J7 and J6 respectively.

On the second circuit board, the signals pass through variable resistors R140 and R144 which act as bias adjustments to the bases of

transistors Q2 and Q4. Transistors Q2 and Q4 act as switching transistors for input to the 5413 dual four-input, positive-NAND, schmitt trigger device via some intermediate current limiting/ pull-up circuitry; resistors R137 & R138, and R141 & R142.

The 5413 device conditions the analog wave form inputs and outputs a digital pulse to the 74LS86 quadruple two-input exclusive OR device which enables clock pulses used for loading the 8254 programmable interval timer device while the motor cycle is stopped. The use of and need for the 74LS86 device will be discussed in more detail in the problems encountered and solved section.

The 74LS86 outputs signals to the 8254 programmable interval timer which counts the number of FPA104 activations, and is read by the CPU to determine the speed and engine revolutions per second.

The input signal from the speedometer interface enters the 8254 via pin 15 (CLK1), while the input signal from the tachometer interface enters via pin 18 (CLK2).

The 8254 is programmed in mode 0, which is used for event counting as in this system. In mode 0, the state of the GATE pins determine whether counting is enabled (GATE=1) or disabled (GATE=0). To enable counting, the three GATE pins (11,14 and 16) are all tied high (+5 volts).

Pin 9 (CLK0) is used as the clock input from the dual 556 timer device. This input drives counter 0 which is used as the master clock for the speed and tachometer interfaces. When counter 0's initial

count decrements to zero, the OUT0 pin (10) of the 8254 goes high and generates an interrupt to the 8085 (RST 7.5).

Upon receipt of the interrupt, the 8085 branches to the interrupt service routine for the speedometer and tachometer. There, it first reads the counts that have registered on counters one and two since the last interrupt, computes the speed and revolutions per second, and writes the values to the proper 4511 displays. It then the reinitializes all three counters and interrupt by writing an initial count to each counter, and lastly branches back to the routine which was under execution prior to receipt of the 8254 interrupt.

Phototransistor Timing The speedometer disc contains four reflective activator stripes, and the tachometer disc contains six. For each revolution of a disc, four and six pulses (FPA104 activations) are produced respectively. In designing such an interface, one must be aware of the possibility that at high speeds or engine rpms, the discs used in the interfaces may spin at a higher rate than the phototransistors are able to respond to.

Because of this possibility, careful attention was given to the rise and fall timing capabilities of the FPA104 phototransistors. The rise and fall times are the times required for the current within the transistor to rise from 10% to 90% of peak value and down again to 10%. Simply stated, this is the time for the transistor to turn on and off again in response to the turn on and turn off activation of the photodiode. The rise and fall time for the FPA104 is 100 microseconds,

which means that the device is capable of turning on and off again ten thousand times a second.

Given the gear ratios present on the motorcycle, at one hundred twenty miles per hour, the speedometer disc will spin 75.32 times a second producing 301.28 FPA104 activations per second (75.32 X 4 stripes).

The tachometer disc, at an engine speed of eleven thousand revolutions per minute will spin 55 times per second producing 275 FPA104 activations per second (45.83 X 6 stripes).

Both the speed and tachometer values are well above safe limits for operation of the motorcycle, and well below the capability threshold of the FPA104 devices, demonstrating that the FPA104 is more than suitable for the interface device of choice in this system.

Analog Signal Conditioning Interfacing analog signals with digital devices can cause several problems: 1) If the digital device is edge sensitive, the analog pulse may rise or fall too slowly to activate the device, 2) If the analog signal is noisy, false activations of the digital device may result.

The signals from the FPA104 devices are sinusoidal in nature and require further conditioning for interfacing to the accompanying digital circuitry. For this purpose, the input signals are input to the 5413 device.

The 5413 is a special type of digital device employing Schmitt trigger internal circuitry and hysteresis. The Schmitt trigger

circuitry is used to define a very precise trip point as the slowly changing analog signal is applied. When this point is achieved, the output of the device changes state quickly, in a digital manner effectively "squaring up" the analog signal into a digital, square wave form. The hysteresis is used as a type of noise immunity further guarding against multiple triggering around the logic state change threshold point of the 5413.

Interrupt Timing The 556 dual timer device outputs a constant clock pulse at a rate of two milliseconds or 2/1000th of a second. Counter 0 is programmed with an initial count of 200, which decrements each time a clock pulse is received by the 556. Given these two values, the 8254 will decrement the count to zero in 400 milliseconds or 4/10ths of a second, which is the interrupt rate of the speedometer and tachometer interface.

Pulse Count v.s Interface Gear Ratios The internals of the stock analog speedometer and tachometer gauges were utilized in this system to facilitate ease of integration into the actual motorcycle system. The stock gauge dials and internal gears were discarded. The remaining parts consisted of the discs and housings which supported the discs, and cable connections.

By using the internals of the stock gauges, the cable interfaces could be maintained. The speedometer cable connects the gauge internals to a housing on the front wheel. The tachometer cable connects the gauge internals to a housing on the top of the engine.

Both the engine and wheel connections use internal gear assemblies to drive the cables. The engine connection is geared from the camshaft, while the wheel connection is geared from the front axle.

Because of these gear assemblies, it was necessary to determine the ratios of disc spins to revolutions of the front wheel and revolutions of the crankshaft of the engine. By marking a reference point on each disc, and slowly spinning the front wheel and engine crankshaft, the following ratios were established; the speedometer disc revolved twenty six times to every nine revolutions of the front wheel for a ratio of 9:26, while the tachometer disc revolved two times to every eight revolutions of the crankshaft for a ratio of 8:2, or 4:1. These ratios could then be used further to compute miles per hour and revolutions per minute, and also to determine the number of reflective stripes to put on each disc to produce the desired count per given time units.

For the speedometer interface it was also necessary to determine travel of the front wheel for application in the rate = distance/time formula. The front wheel was found to have a circumference of 81 inches meaning that for every 81 inches traveled, the front wheel will revolve once. At 63,360 inches in one mile, the front wheel will revolve 782.2 times and the speedometer disc will revolve X number of times. By applying this ratio to the known ratio of 9:29, cross multiplying, and solving for X, it was found that the speedometer disc

will revolve 2346.6 times in one mile.

Final computations and methods used to determine disc stripe count and placement for the speedometer and tachometer interfaces are discussed in greater detail under the computer aided development section.

Security The security interface is a function of input values from the keyboard, and output values to two relays. The 8279 programmable keyboard interface is an internal bus device and communicates directly with the CPU. Both the 8279 and the two security relays reside on the second circuit board.

One relay controls ignition enable/disable, while the other controls the horn blow circuit. Both relays receive their energize/de-energize signals from the 8255 programmable peripheral device.

The keyboard resides on the first (display) circuit board and is interfaced to the 8279 via connector pair J2/J5.

All four scan lines of the 8279 are utilized (pins 32-35) while only four of eight return lines are used (pins 1, 2, 38, and 39). This configuration produces the eight lines needed to interface a 16-key four-by-four matrix. The remaining return lines (pins 5-8) are left unconnected. Internal pull-up resistors keep theses pins at a logic 1 state.

Keyboard

Use of the 8279 Control/Data Words The 8279 is set up prior to use by writing a series of control words to the command

register. The first control word sets up the 100 kHz clock. The second control word issued sets up the mode of the device, which in this system is the decoded scanned keyboard, 2-key lockout. The final control word issued sets up the 8279 for the first read of the FIFO RAM. The 8279 remains in the read ready state until another command is written to the command register. Figure 15 shows the format, and sequence of the three control words used to set up the 8279 for this system.

Subsequent reads of the FIFO RAM return row/column data in a data word format. In this system the correct security code values and their hexadecimal equivalents (from the keyboard map) in the correct order of entry are as follows; "A"(D2), "9"(CA), "2"(D0), "3"(D8), "B"(DA). Figure 16 shows the format and sequence of the data words returned from a correct attempt at entering the security code.

Interrupt Handling The issuance of the control words to the 8279 command register clear the interrupt (pin 4) for initial use. Each time a key is pressed, the 8279's interrupt pin goes high and generates an RST 5.5 interrupt at the 8085's pin 9. Program execution branches to interrupt vector 2CH, where a jump instruction further directs the 8085 to the starting address of the keyboard software routine at location 0079H. The keyboard routine is the first routine executed in this system and no other system interrupts are enabled during this execution.

Alarm The alarm is implemented by tapping into the

motorcycle horn circuit and passing the 12 volt horn signal through relay RL1 which acts as a switch to enable or disable this function. The 12 volt horn circuit enters the system on the third circuit board via pin 9 of connector J9, runs to the second circuit board via

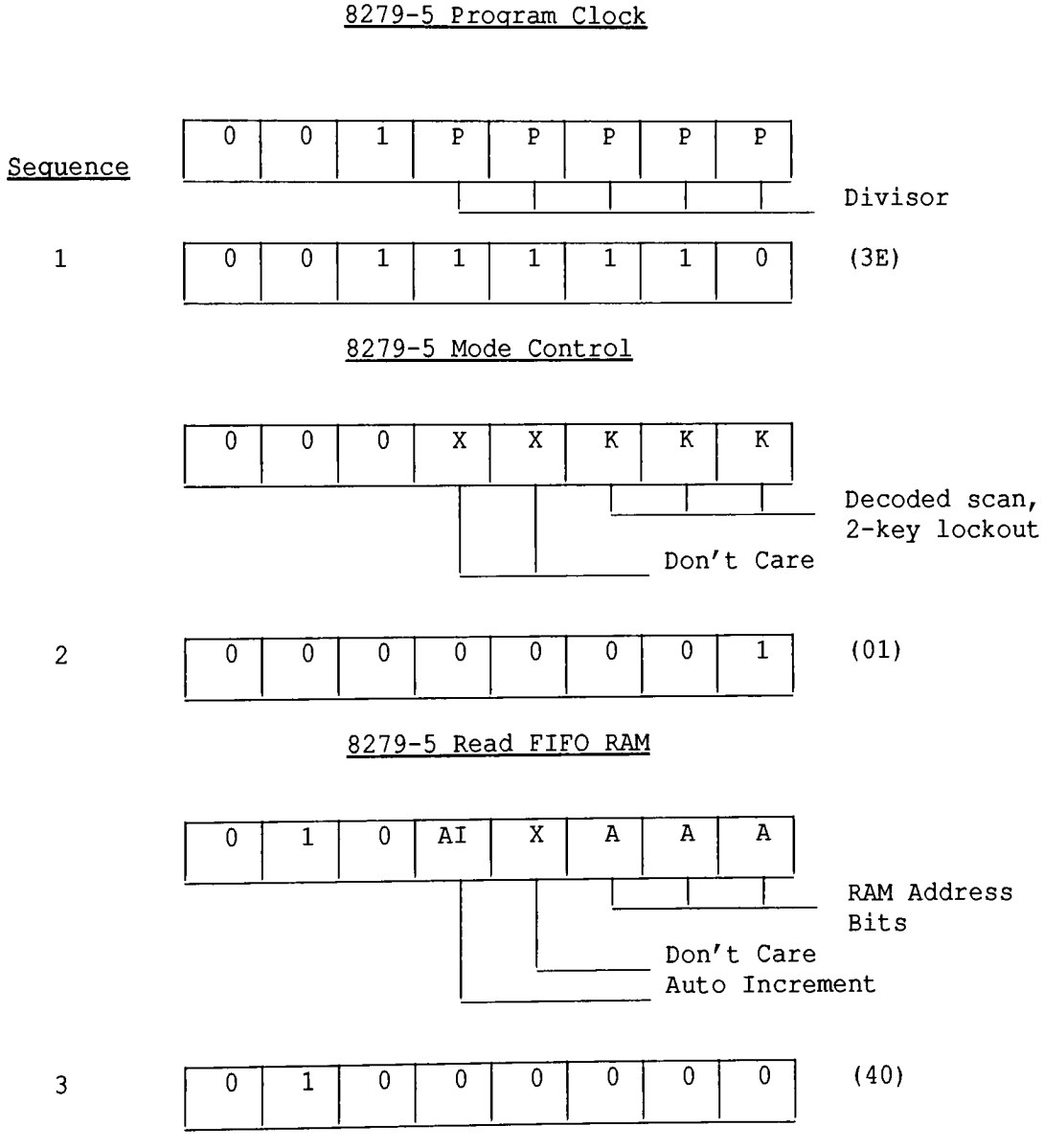
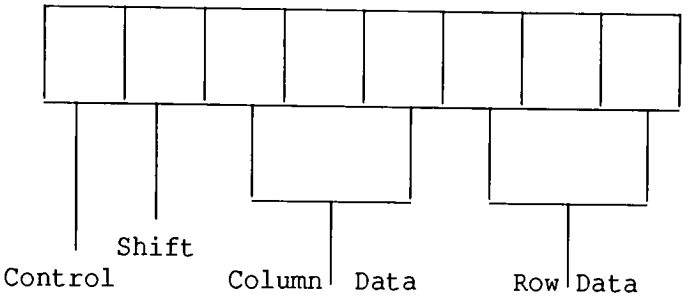


Figure 15

8279-5 Data Word Format



Sequence

1	1	1	0	1	0	0	1	0	(D2)
2	1	1	0	0	1	0	1	0	(CA)
3	1	1	0	1	0	0	0	0	(D0)
4	1	1	0	1	1	0	0	0	(D8)
5	1	1	0	1	1	0	1	0	(DA)

Figure 16

connector pin 3 of J7, and J6, runs through the normally open section of the relay and returns via the same connectors, pins 4 of J6/J7, and 10 of J8.

The relay is energized via a signal from the 8255 programmable peripheral device if an incorrect code has been entered. Upon system initialization, a low signal is applied to the relay as it is in a normally open state and the horn is disabled as a result. If the security code is incorrect, an 8255 high signal is applied to the base of the relay driving transistor Q1, which energizes the relay, closes the switch, and the horn blows.

Ignition disable The implementation of the ignition enable/disable circuit is essentially the same as the alarm circuit. The kill switch circuit in the motorcycle will be tapped and the 12 volt signal run through relay RL2. The signal enters the system via board 3 and runs to the second board through the same connectors as the alarm circuit. The relay driving circuitry is identical to the alarm function, as is the activation by a signal from the 8255 programmable peripheral device.

Connections at the relay however are different from the alarm circuit. The 12 volt kill circuit signal runs through the normally closed section of the relay so that the ignition function will be active when the relay is not energized. Upon system initialization, the relay is energized, the switch opens, and the ignition function is disabled. Only upon correct entry of the security code is the signal from the 8255 withdrawn enabling the start of the motorcycle.

The relay connections were implemented in this manner to provide normal functionality of the alarm and ignition enable/disable systems

when the relays were in a deactivated state for greater reliability and longer life of the relays.

Turn Signals/Kick stand These three functions employ identical interface circuitry and software detection logic, and each function will tap into the existing motorcycle circuitry for it's input signal to this system. In each case, the sensor interface is a switch.

The closure of any of these three switches will provide a 12 volt signal into the system via connector J9 pins 17-19. Pin 17 is the input signal pin for the kick stand down function, pin 18 is the input signal pin for the right turn signal, and pin 19, the left turn signal. From connector J9, the three signals pass through identical voltage division circuitry, and pass to the second circuit board via pins 7-9 of connectors J7/J6. On the second circuit board, these signals are fed as input to pins 11-13 (PC6-PC4) of the 8255 programmable peripheral controller.

Switch Sensor Interface The left and right turn signal functions will take advantage of the existing switch on the motorcycle. The kick stand down switch is not a stock item, and will be installed to provide this functions signal.

Each of the three signals pass through voltage division circuitry implemented with two fixed 1/4 watt resistors. Each pair of resistors consists of one 2K and one 1K resistor. A lead tapped from between these resistor pairs yields a 4 volt logic 1 signal to the 8255 when a 12 volt signal is applied by the closure of a switch. The second

resistor of each pair is grounded and "pulls down" the voltage to an active low logic state when the switches are in an open state.

Polled Detection Logic Detection of a switch closure is similar to reading the interrupt mask of the 8085. The 8255 is periodically polled from the monitor routine, and the 8-bit word representing the status of PORT C is read. PORT C is programmed to enable the four low order bits as output (PC0-PC3), and the four high order bits (PC4-PC7) as input. When the 8-bit PORT C word is read by the 8085, only bits PC4-PC6 are tested. All other bits are masked out while each of the three bits is tested one at a time for a logic 1.

If a particular bit is found to be a logic 1, a software controlled branch is made to the particular service routine.

LED Flash Cycle/Timing The turn signal flash software routine is the most CPU intensive routine in the system. The flash routine computes which LED of which LED unit to light, lights the LED, branches to a delay routine, then returns and repeats the process until all twenty LEDs are lit. After all LEDs have been lit, they are all blanked, and the 8085 branches back to the monitor routine where it again checks for switch closures. If the turn signal switch is found to be closed, it repeats the branch to the turn signal routine.

Lighting of the entire twenty LEDs takes approximately one second. It therefore is possible that the turn signal switch will be opened while the flash routine is executing. Since the state of the switches are not checked again until return from the flash routine to

the monitor, this implementation results in the effect that the flash routine will never be interrupted by the opening of a switch, and will always complete the lighting of all twenty LEDs once a switch closure is sensed.

Upon final implementation of the monitor routine, is was necessary to insert a delay loop to avoid adverse effects on the A/D function displays (discussed further in the problems encountered and solved section). This delay directly effects the timing of the blank interval between turn signal display flashes since it must be executed prior to the next check for a closed turn signal switch. By adjusting this delay an attempt was made to balance the time that the turn signals were off in ratio to the time they were on, without degrading the A/D functions displays.

A/D Conversion Three signals from external analog sensors are input to this system for conversion to a digital value. The first sensor changes in relation to the amount of fuel in the fuel tank, the second in relation to the temperature of the engine, and the third in relation to the charging voltage of the battery.

Pin 14 of connector J9 on the third circuit board supplies 12 volts to the LM335 precision temperature sensor which will reside behind the heads of the engine. Pin 13 of connector J9 is the ground return for the LM335. The temperature sensor signal is a tap taken off the 12 volt line going to pin 13 of connector J9, and runs to the second circuit board via pin 13 of connectors J7 and J6. This signal

is then fed directly to pin 26 of the ADC0808 A/D converter and becomes logical channel 1.

Pins 22 and 23 of connector J9 are the input signal pins for the charging voltage and fuel level functions respectively. These signals receive identical variable voltage division implemented via one 1K fixed 1/4 watt resistor, and one 5K 1/4 watt variable resistor each. The signals pass to the second board via pins 11 and 12 of connectors J7 and J6 and are fed directly to pins 27 and 28 of the ADC0808 to become logical channels 2 and 3 of the A/D converter.

The ADC0808 is an internal bus device and is accessed directly by the CPU sharing the internal 8-bit data bus.

The second half of the 556 dual timer supplies the clock input to the ADC0808 at pin 10 (CLK) and provides the A/D conversion clock rate of 100 kHz (10 microseconds).

The 74LS02 (quadruple 2-input positive-NOR) and the 74LS04 (hex inverter) chips gate the READ, WRITE, and chip select signals to the ADC0808 for proper operation.

Pins 23-25 of the ADC0808 control addressing of a particular channel (1-8), and are wired as part of the internal data bus as D0-D2.

Temperature Sensing

Kelvin To Fahrenheit Conversion The LM335 parametric data is referenced in degrees Kelvin by the manufacturer. Although the scales are different for Kelvin and centigrade, the size of a

degree in either scale is the same. For calibration purposes, a reference voltage was obtained at a fixed temperature and treated as a centigrade degree for conversion to a Fahrenheit value. For the centigrade to Fahrenheit conversion, the centigrade value is multiplied by a factor of 1.8 and 32 is added to the result.

Low Fuel/Battery Charging Sensing

Voltage Division The signals from the low fuel sensor and the battery charging voltage pass through voltage division circuitry implemented by using fixed and variable resistors rather than fixed resistors only.

The fuel sensor installation will mandate alterations to the fuel tank, and is as yet unchosen. Since the sensor is non existent, the output voltage range is also unknown. By using variable voltage division circuitry, the output range can be varied for calibration purposes upon final installation.

Similarly, the charging voltage input range can be precisely calibrated during final installation into the motorcycle circuit by adjusting the variable resistor for this signal.

Addressing The ADC0808 A/D Converter A read of the ADC0808 by the CPU first causes the READ and chip select signals to go low. These signals are NOR gated via the 74LS02 and cause a high level output signal to be applied at pin 9 (OE) of the ADC0808. The address placed on the bus is latched via pins 23-25 of the ADC0808. The

ADC0808 then puts the data from the addressed channel on the data bus and it is read by the 8085.

A write operation (which starts the conversion for a particular channel) is executed by NOR gating the chip select and WRITE signals of the 8085 via the 74LS02 which produces a high level signal at pin 6 (START) of the ADC0808. The high level output of the 74LS02 (pin 4) also feeds the 74LS04 hex inverter input (pin 1) which produces a low level output to pin 22 (ALE) of the ADC0808. These two signals (START and ALE) must both be applied to start a conversion. The address is latched by pins 23-25 of the ADC0808, and the conversion process begins.

ADC0808 System Precision The precision of the ADC0808 is determined by two factors: 1) The bit resolution of the chip, and 2) The working voltage range set between the REF+ and REF- inputs. The ADC0808 is an 8-bit device which means that the working voltage range must be represented by eight bits. Eight binary bits yields two hundred fifty six possible divisions of the working voltage range.

In this system the REF+ pin is tied to 5 volts and the REF- pin is tied to ground, or "ground referenced". The resulting working voltage range is a 5 volt span.

Division of the 5 volt range by the two hundred fifty six possible divisions of that range yields a single division resolution of .0195 volts, or 19.5 millivolts.

For ease of mathematical computation in this system, the 19.5 millivolt value is rounded to 20 millivolts resulting in a "built in" .5 millivolt conversion error. System software therefore interprets each divisional change to be in 20 millivolt increments.

Software Design

Processor Modules The following sections outline the program code for each function in this system. There was no assembler used in the generation of code for this system. The code was developed on a modular basis for each function, and entered into an EEPROM by hand.

Due to this development method, there are no labels or symbolic values used for addresses or data.

The complete contiguous program listing can be requested by contacting the author.

Noticeable in several sections of the complete program listing are areas of contiguous memory space not occupied by executable code or data. A lesson quickly learned, was to leave memory space open before and after the module in case modifications needed to be made. Nothing is more frustrating than to have to move entire sections of code following a particular module, further in memory (by hand) if a section of code needed to be altered and expanded.

Another reason for the empty space was that an effort was made to generate the most efficient code possible to save CPU cycles and time. Often a particular segment of code was "optimized" after it was debugged, and re-written in less space than allotted for the original module.

In the following processor module sections, labels have been added to aid understanding and help in following the program flow wherever possible.

Each of the processor module code sections is divided into three

parts. The left most column shows the actual memory locations of the code. The next column shows the hand assembled hexadecimal code for the instructions, and the last columns show the mnemonic instructions and operand fields for the code. The discussion of the workings of the code follows to the right.

System Initialization There are actually two parts to the system initialization code. The first system initialization takes place at system cold start prior to the entry of the security code by the operator. The second initialization section takes place if a successful attempt has been made at entering the code. The following two system initialization sections are presented in this order.

```
0000    31FF0F      LXI  SP, 0FFFH      When the system is initially
0003    F3          DI
0004    3E88        MVI   A, 88H       powered up (cold start), or
0006    D313        OUT   13H
0008    3EFF        MVI   A, 0FFH      receives a reset pulse (warm
000A    D310        OUT   10H
000C    D330        OUT   30H          start), the CPU begins execu-
000E    D338        OUT   38H
0010    D340        OUT   40H          tion at location 0. The first
0012    D348        OUT   48H
0014    D350        OUT   50H          instruction in ROM initializes
0016    D358        OUT   58H
0018    D360        OUT   60H          the stack pointer to an area
001A    2F          CMA
001B    D310        OUT   10H          of available memory for that
001D    3E3F        MVI   A,3FH
001F    D311        OUT   11H          use. The next instruction (DI),
0021    3E00        MVI   A,00
0023    D311        OUT   11H          disables all interrupts to
0025    3E02        MVI   A,02
0027    D312        OUT   12H          guard against spurious inter-
0029    C33F00      JMP   003FH
002C    C37900      JMP   0079H        rupt signals from system de-
003C    C3CA01      JMP   01CAH
003F    21000C      LXI   H,0C00H      vices prior to their initial-
0042    3ED2        MVI   A,0D2H
```

```
0044    77          MOV   M,A
0045    23          INX   H
0046    23          INX   H
0047    3ECA        MVI   A,0CAH
0049    77          MOV   M,A
004A    23          INX   H
004B    23          INX   H
004C    3ED0        MVI   A,0D0H
004E    77          MOV   M,A
004F    23          INX   H
0050    23          INX   H
0051    3ED8        MVI   A,0D8H
0053    77          MOV   M,A
0054    23          INX   H
0055    23          INX   H
0056    3EDA        MVI   A,0DAH
0058    77          MOV   M,A
0059    3E3E        MVI   A,3EH
005B    D319        OUT   19H
005D    3E01        MVI   A,01
005F    D319        OUT   19H
0061    3E40        MVI   A,40H
0063    D319        OUT   19H
0065    0600        MVI   B,00
0067    48          MOV   C,B
0068    50          MOV   D,B
0069    3E3E        MVI   A,3EH
006B    30          SIM
006C    21010C      LXI   H,0C01H
006F    FB          EI
0070    76          HLT
```

ization. Locations 03 and 06 initialize the 8255 PORTS A-C by writing a hex 88 to the 8255 command register at PORT 13H. Locations 8 and 0AH, load the accumulator with the hex value to blank the 7-segment displays and write the value (FF) to the display board data bus via the 8255 port address 10H. The instructions at locations 0CH-18H clock the blank display value into all of the 4511 devices. The blank display value for the 74LS373 devices is 00, so the value in the accumulator is complemented via the CMA instruction at 1AH, and this new value is put on the display data bus. Locations 1DH-23H provide the simulated high to low transition signals via the 8255 to clock the value into the 74LS373s. A hex 3F is used to address all 74LS373 chip selects and is written to PORT 11H which effectively raises all the chip selects to a logic 1 state. The value 00 is then written to PORT 11H to bring all chip selects low again.

The above blanking sequence insures that the entire display board will be dark upon system initialization or reset.

The instructions at locations 25H and 27H write the value 02 to the 8255 PORT 12H which addresses the two relays and disables the horn blow and ignition circuits prior to entry of the security code.

The jump instruction at location 29H causes continuation of the initialization code at location 3FH by jumping over the reserved locations used for hardware and software interrupt vectors.

The jump instruction at location 2CH resides in the interrupt vector for the RST 5.5 hardware interrupt from the 8279 keyboard controller. This jump instruction branches control, upon interrupt, to the entry point of the keyboard routine.

Similarly, the jump instruction at 3CH branches control upon receipt of the RST 7.5 interrupt to the entry point of the speedometer and tachometer routine.

The instruction (LXI) at location 3FH loads the H&L register pair with the starting RAM address for storage of the security key values. The H&L registers are used for indirect addressing via the MOV instruction which stores the value of the accumulator at the memory locations pointed to by the address in those registers.

The instructions at locations 42H-58H perform successive loads and stores of the five security key values. Between each load and store, the H&L registers are incremented twice via the INX command. This sequence causes each new key value to be stored in every other

memory location rather than in contiguous memory locations. In the keyboard routine, the "in between" locations are used to store the values input from the keyboard. The memory locations are then compared in contiguous pairs for security code key matches.

The instructions at locations 59H-63H initialize the 8279 clock, mode, and set-up values for input of the security code via the keyboard.

These instructions are followed by a series of register initializations for use in the keyboard routine. Register B, C, and D are set to zero. These registers are used as counters for the number of key closures sensed, number of entry attempts, and number of LED flashes respectively.

Location 6BH sets the interrupt mask via the SIM instruction to only allow interrupt RST 5.5 (8279 keyboard controller).

Location 6CH sets the address stored in the H&L registers to point to the first "in between" memory location (0C01) as the location to store the first security code key value input from the keyboard.

Lastly, at locations 6FH and 70H, 8085 hardware interrupts are enabled via the EI instruction, and the system then sits and waits for an interrupt from the 8279 by executing the HLT instruction which is the 8085 equivalent of an idle instruction. Since all other interrupts have been disabled, the system will remain at this point in execution indefinitely until it receives an entry from the keyboard.

The second initialization is used for system set-up following a successful entry of the security code. The first instructions at 25CH and 25EH reset the two security relays to enable ignition and disable the horn. At locations 260H and 262H, the interrupt mask is set for RST 7.5 which is used for the speedometer and tachometer interfaces. The keyboard interrupt is not enabled since the keyboard is only scanned during security code input. Locations 263H-26EH set the modes of the 8254 interval timer counters to mode zero by writing the appropriate codes to the 8254's command register at port address 23H.

```
025C    3E00        MVI   A,00
025E    D312        OUT   12H
0260    3E3B        MVI   A,3BH
0262    30          SIM
0263    3E50        MVI   A,50H
0265    D323        OUT   23H
0267    3E90        MVI   A,90H
0269    D323        OUT   23H
026B    3E10        MVI   A,10H
026D    D323        OUT   23H
0270    3E78        MVI   A,78H
0272    D321        OUT   21H
0274    3E63        MVI   A,63H
0276    D322        OUT   22H
0278    3EC0        MVI   A,0C0H
027A    D311        OUT   11H
027C    3E00        MVI   A,00
027E    D311        OUT   11H
0280    3EC8        MVI   A,0C8H
0282    D320        OUT   20H
0284    3E00        MVI   A,00
0286    32500C      STA   0C50H
0289    32580C      STA   0C58H
028C    32590C      STA   0C59H
028F    325A0C      STA   0C5AH
0292    CD1603      CALL  0316H
0295    3E00        MVI   A,00
0297    D328        OUT   28H
0299    FB          EI
```

Locations 270H through 27BH load the initial counts of counters one and two which are used for the speedometer and tachometer counters respectively. After the initial counts are written to each of the counters (PORT 21H and PORT 22H), the values are clocked into the individual counters by applying a pulse via the 74LS86. This pulse, like the pulse used to clock data values into the 74LS373s, is

simulated via the 8255 by bringing the clock signal high and back low. The 74LS86 is addressed by writing the high to low transition codes to PORT 11H of the 8255.

The final counter access takes place at location 280H, where the initial count (decimal 200) is written to counter 0. This write starts the speedometer and tachometer interface decrement sequence which will interrupt the CPU when the terminal count of zero is reached.

The instruction at location 0284H loads the value (00) which is used as the flag for A/D conversion channel set and verification. The value 00 is used since this is the value to activate the first channel to be converted; channel one. The value is stored at RAM location 0C50H, which is the location to be accessed by the verification and channel set software.

The instructions at locations 289H through 291H initialize the RAM memory locations used to store the status of the fuel level and kick stand down functions accessed by the warning light module. The initialization is accomplished by using the zero value already in the accumulator and writing this value to RAM locations 0C58H, 0C59H, and 0C5AH.

The CALL at location 292H calls the routine that stores the flags which represent all the possible warning conditions in RAM. These flags will be manipulated by warning light computation software as per the flag values stored in the above initialized status RAM locations.

The first A/D conversion process is started by the instructions

at locations 295H and 297H. The value 00 is written to the A/D converter at PORT 28H which begins the conversion for channel one.

The final initialization instruction at location 299H enables interrupts. This will allow the interrupt from the 8254 (RST 7.5) which was set via the SIM instruction to be recognized by the 8085 when the count decrements to zero.

At this point, the speedometer and tachometer interface is activated and running, and the conversion of the first A/D channel (engine temperature) is in progress.

Software execution now continues by entering the system monitor which will be discussed in a later section.

Left and Right Turn Signals There are two main functional areas within the turn signal code. The first area is the rotation code which sets up the proper LED to be lit, and the second area is the loop counter checks that insure that the proper ten LED unit is addressed. Both the left and right turn signal routines share the same code for timing delays and LED blanking.

The growing bar of light effect is implemented by addressing the first LED of a two unit, twenty LED series, and sequentially lighting all twenty, one at a time, in the direction of the turn.

A series of data, select, and offset values are stored in ROM at the end of the two routines, which are accessed by the algorithm.

Upon entry to the turn signal routines, the H&L registers are loaded with the address value which points to the first data word

stored in ROM. In locations 113H-119H and 13EH-144H, the C register is initialized with the value of the loop counter, the D&E registers are then loaded with the address of the value of the first offset, the accumulator is loaded via the address in H&L with the first data value, and the H&L registers are incremented to point to the first LED unit select value in ROM.

This initialization sequence sets up all registers and address values for the next instruction which is a CALL to the rotate, display and delay code.

The same code for the rotate and display functions are shared by the left and right turn signal routines except for the processing of the first four LEDs of the right turn signal.

The 74LS373s which drive the turn signal LEDs are wired such that the same direction of rotation of the data word in the accumulator will provide right to left LED lighting of the left signal, and left to right lighting of the right signal. Recall, although, that one 74LS373 is used by both the left and right turn signal LEDs, half (four LEDs) for the left, and half (four LEDs) for the right. This double duty of one 74LS373 causes the inconsistency of use of the rotate and display code. For the first four rotations of the data word for the right turn signal, the direction of rotation must be opposite to create LED lighting in the proper direction. From this point on, the routine uses the shared rotate and display code.

Upon entry to the rotate and display code, the value of the

accumulator, which is the data word sent to the 74LS373s, is placed on the display board data bus via PORT 10H of the 8255. The word is then rotated one position right and stored in register B. The offset value pointed to by the D&E registers is then loaded into the accumulator, the rotated value in B added to the offset, and the result stored back in the B register.

The offset values are always the same as the initial data word value. The adding of the offset value back into the data word always resets the bit position that was shifted one place to the right, to a logic 1. Figure 17 shows the effect of seven rotates and additions of the seven offsets.

1	0	0	0	0	0	0	0	Data Word (80H)
1	1	0	0	0	0	0	0	Rotate, Add Offset (80H)
1	1	1	0	0	0	0	0	Rotate, Add Offset (80H)
1	1	1	1	0	0	0	0	Rotate, Add Offset (80H)
1	1	1	1	1	0	0	0	Rotate, Add Offset (80H)
1	1	1	1	1	1	0	0	Rotate, Add Offset (80H)
1	1	1	1	1	1	1	0	Rotate, Add Offset (80H)
1	1	1	1	1	1	1	1	Rotate, Add Offset (80H)

Figure 17

This data word in register B will always contain the result of the rotations, and is set up for the next placement on the display data bus prior to exit of the routine.

The remainder of the routine loads the preset LED unit select value and clocks the data word into the proper LED unit, then calls the delay sequence and returns to the calling routine.

```
0110    219801      LXI   H,0198H
0113    0E15        MVI   C,15H
0115    119E01      LXI   D,019EH
0118    7E      L1  MOV   A,M
0119    23          INX   H
011A    CD7801  NXT CALL  RODE
011D    0D          DCR   C
011E    79          MOV   A,C
011F    FE01        CPI   01
0121    CA8A01      JZ    BALL
0124    FE09        CPI   09
0126    CA3501      JZ    ALIT
0129    FE11        CPI   11H
012B    CA3501      JZ    ALIT
012E    78          MOV   A,B
012F    C31A01      JMP   NXT
0135    23    ALIT  INX   H
0136    13          INX   D
0137    C31801      JMP   L1
013B    21A101      LXI   H,01A1H
013E    0E15        MVI   C,15H
0140    11A701      LXI   D,01A7H
0143    7E          MOV   A,M
0144    23          INX   H
0145    D310  NXT1  OUT   10H
0147    07          RLC
0148    47          MOV   B,A
0149    1A          LDAX  D
014A    88          ADC   B
014B    47          MOV   B,A
014C    7E          MOV   A,M
014D    D311        OUT   11H
014F    3E00        MVI   A,00
0151    D311        OUT   11H
0153    CDAB01      CALL  DEL
0156    0D          DCR   C
```

The entry point for the left turn signal routine is 0110H. The instructions at locations 110H-11AH perform the initialization and call to the rotate and delay code. The code at locations 11DH-131H comprise the second major functional area of the code; the loop counter checks. As the loop counter in register C is decremented, it is checked against three values; 01 at location 11FH, 09 at location 124H, and and 11H (decimal 16) at location 129H. If the C register equals 01, then all LEDs have been lit and control branches

```
0157    79          MOV   A,C
0158    FE11        CPI   11H
015A    CA6101      JZ    ALIT1
015D    78          MOV   A,B
015E    C34501      JMP   NXT1
0161    23    ALIT1 INX   H
0162    13          INX   D
0163    7E          MOV   A,M
0164    23          INX   H
0165    CD7801  L2  CALL  RODE
0168    0D          DCR   C
0169    79          MOV   A,C
016A    FE09        CPI   09
016C    CA6101      JZ    ALIT1
016F    FE01        CPI   01
0171    CA8A01      JZ    BALL
0174    78          MOV   A,B
0175    C36501      JMP   L2
0178    D310  RODE  OUT   10H
017A    0F          RRC
017B    47          MOV   B,A
017C    1A          LDAX  D
017D    88          ADC   B
017E    47          MOV   B,A
017F    7E          MOV   A,M
0180    D311        OUT   11H
0182    3E00        MVI   A,00
0184    D311        OUT   11H
0186    CDAB01      CALL  DEL
0189    C9          RET
018A    3E00  BALL  MVI   A,00
018C    D310        OUT   10H
018E    3E37        MVI   A,37H
0190    D311        OUT   11H
0192    3E00        MVI   A,00
0194    D311        OUT   11H
0196    C9          RET
0198    08  (Left signal data 1)
0199    04  (Left signal select 1)
019A    80  (Left signal data 2)
019B    02  (Left signal select 2)
019C    80  (Left signal data 3)
019D    01  (Left signal select 3)
019E    08  (Left signal offset 1)
019F    80  (Left signal offset 2)
01A0    80  (Left signal offset 3)
```

to the blank-all code at location 18AH. If the C value is 09, then one full unit of ten LEDS has been lit, control branches to location 135H where the H&L registers are incremented to point to the next data word, the pointer in the D&E register is incremented to point to the next offset, and control branches back to the initial register set-up at location 118H via the jump at location 137H to continue the process. If the C value is found to be 16H, then another (the first) LED unit has been lit, and the same branch sequence is followed as in the C = 09 case by jumping to location 135H. Note the use of the H&L registers at locations 118H and the data organization at

```
01A1    10  (Right signal data 1)
01A2    04  (Right signal select 1)
01A3    80  (Right signal data 2)
01A4    10  (Right signal select 2)
01A5    80  (Right signal data 3)
01A6    20  (Right signal select 3)
01A7    10  (Right signal offset 1)
01A8    80  (Right signal offset 2)
01A9    80  (Right signal offset 3)
01AB    E5      DEL PUSH  H
01AC    2AB701      LHLD  01B7H
01AF    2B      L3  DCX   H
01B0    7C          MOV   A,H
01B1    B5          ORA   L
01B2    C2AF01      JNZ   L3
01B5    E1          POP   H
01B6    C9          RET
01B7    11
01B8    11
```

locations 198H-1A9H. The values stored at 198H-19DH alternate between data and select values. The H&L registers are used to address both data and select values via the instructions at locations 118H and 119H. At location 118H, the data value is loaded into the accumulator, and at location 119H the H&L registers are incremented to point to the next select value. The H&L registers remain pointing to the select value until the next increment at 135H, while the data value is preserved by storing the value in the C register in the rotate and delay routine. The values at 19EH-1A0H are the three consecutive offset values for the left turn signal routine and are accessed by the increments of the D&E registers at location 0136H.

The entry point for the right turn signal routine is at location 013BH. The code for the opposite rotation gave rise to two choices for implementation: 1) Perform the opposite rotation in the main right turn signal code and utilize the existing rotate and delay code by calling the module at a different entry point (after the rotate), or 2) Duplicate the rotate and delay code for the first four rotations.

The choice was made to duplicate the code to avoid the confusion and poor programming technique of using multiple entry points to the same subroutine.

Because of this choice, the code at locations 13BH-160H contains the initialization as in the left turn signal routine, as well as the duplicate code for the rotate and check logic and delay call.

Normal execution begins after fallout from the first four LED loop at location 161H. The code at locations 161H-177H is identical to the left turn signal routine in code and execution of shared subroutines.

Organization of data, select, and offset values are identical to that of the left turn signal, and reside at locations 1A1H-1A9H.

The shared blank all code at locations 18AH-196H places a data value of zero on the display data bus and clocks all 74LS373 devices high, and then low to turn off all lighted LEDs. This is the last module executed by either the left or right turn signal routines, so the RET (return) instruction at location 196H returns execution to the calling module, in this case the system monitor.

The shared delay routine begins at location 1ABH. The value of the H register is preserved upon entry (via the PUSH H command) by placing the value on the stack. The H&L registers are then loaded with the delay value at locations 1B7H and 1B8H, then a simple loop of decrements and ORs is performed until the value is decremented to

zero, whereby the H value is popped off the stack and control branches to the calling program via the RET instruction at location 1B6H.

Hexadecimal to BCD Conversion All modules which produce values displayed on the 7-segment displays via the 4511 devices must first call the hex-to-BCD conversion routine to convert the hexadecimal value to a BCD number. The reason for this is twofold; first, the display for the operator must be in an understandable decimal format, and second, the 4511s are only capable of displaying the numbers 0-9.

The hex-to-BCD conversion in this system is able to convert numbers as large as 63H (99 decimal). If the number is larger than 99, prior prescaling must be done to reduce it to 99 or less.

Since it takes only four bits to represent one BCD digit, it is possible to pack two BCD digits into one 8-bit word, which is what is done in the conversion module in this system.

```
0220    C5          PUSH  B          The routine is called at entry
0221    D5          PUSH  D
0222    E5          PUSH  H          point 0220H with the number to
0223    0E0A        MVI   C,0AH
0225    1601        MVI   D,01       be converted already contained
0227    47          MOV   B,A
0228    B9      NXT CMP   C          in the accumulator. The first
0229    FA3802      JM    LESS
022C    CA4C02      JZ    EQUAL      three instructions save the
022F    79          MOV   A,C
0230    C60A        ADI   0AH        current register values on the
0232    4F          MOV   C,A
0233    14          INR   D          stack so the registers can be
0234    78          MOV   A,B
```

```
0235    C32802      JMP   NXT
0238    79    LESS  MOV   A,C
0239    D60A        SUI   0AH
023B    4F          MOV   C,A
023C    15          DCR   D
023D    78          MOV   A,B
023E    91          SUB   C
023F    47          MOV   B,A
0240    3E00        MVI   A,00
0242    7A          MOV   A,D
0243    17          RAL
0244    17          RAL
0245    17          RAL
0246    17          RAL
0247    80          ADD   B
0248    E1          POP   H
0249    D1          POP   D
024A    C1          POP   B
024B    C9          RET
024C    3E00  EQUAL MVI   A,00
024E    7A          MOV   A,D
024F    17          RAL
0250    17          RAL
0251    17          RAL
0252    17          RAL
0253    E1          POP   H
0254    D1          POP   D
0255    C1          POP   B
0256    C9          RET
```

reinitialized and used within the routine. The algorithm is implemented by a succession of comparisons and counter increments. The comparisons are made to numbers beginning with ten, and increasing by a factor of ten for each comparison. Each time the number being compared is found to be larger than the factor, the counter is incremented by one, until the number is found to be less than the factor. The C register is used to contain the factor and is initialized at location 223H. The comparison loop increment counter is stored in the D register and initialized at location 225H.

The instructions at locations 227H-237H perform the compare and increment logic. At location 227H, the accumulator value is preserved by storing it in register B. The value is then compared against the contents of the C register at location 228H. If the number is found to be less than the comparison factor, control branches to location 238H for further processing and packing. If the number is found to be equal to the comparison factor, control branches to location 24CH. If the

number was larger than the comparison factor, the value in register C is incremented by a factor of ten, the comparison loop counter in register D is incremented by one at location 233H, the original value in register B is restored at location 234H, and control branches back to location 228H for the next compare.

This algorithm is implemented in such a manner that when the value in the accumulator is found to be less than the comparison factor, the loop counter will be incremented one more than the actual value. In addition, the difference between the accumulator value and the factor value must be computed to provide the unit difference between the two numbers.

The algorithm decrements the factor value by ten and subtracts it from the accumulator value to provide the units difference and decrements the loop counter via the instructions at the branch point at location 238H-23FH.

As an illustration, if the value passed in was 23H (decimal 35), the less than condition would be achieved when 35 was compared to the incremented C register value of 40. The loop would be executed three times, and with an initial value of 1, the loop counter would contain the value 4 at the time of the branch. At locations 238H-23FH, the loop counter is decremented by one to form the tens value of 3, and the factor (40) is then decremented by ten, to 30, and subtracted from the accumulator value (35) to form the units value of 5. The resulting BCD representation of 23H is therefore 35.

The instructions at locations 240H-247H pack the two 4-bit BCD values into one 8-bit word in the accumulator for return to the calling module. The accumulator value is first cleared by loading a zero at location 240H. The tens value (loop counter in register D) is then loaded into the accumulator and rotated four positions to the left to form the first BCD value as the most significant four bits. The B register which now contains the units value is then added to the value in the accumulator to form the second BCD value as the least significant four bits.

Locations 24CH-252H are the pack instructions branched to if the value compared was found to be an even factor of ten (10, 20, ... 90). The branch to this pack code takes place prior to the loop counter increment at 233H, so the value of the loop counter is actual and does not need to be decremented. Since the value fell on an even factor of ten, there is also no units value to be computed. Because of this condition, the loop counter is the only value rotated into the accumulator.

The packing takes place at locations 24EH-252H by loading the register D value into the accumulator and rotating the value four places to the left to form the first BCD value as the most significant four bits.

The rotate instructions cause the most significant bit of the accumulator to be rotated into the least significant bit. Since the accumulator was set to zero at location 24CH, as the loop counter

value is rotated four positions left, four zeros are rotated into the four least significant bits and no further addition takes place to prepare the value for return to the calling module.

At the end of each pack code segment, the values placed on the stack are restored via the POP instructions, and control returns via the RET instructions to the calling module.

Speedometer/Tachometer Computation Computation and software flow for the speedometer and tachometer algorithms are straightforward.

An interrupt is received from the 8254 when the initial count in counter 0 has decremented to zero. Upon entering the speedometer and tachometer subroutine, the values of counters 1 and 2 are latched, read, converted to BCD format, and displayed. The last last sequence of instructions reinitializes all three counters, and control branches back to the module where the CPU was executing instructions prior to the interrupt.

The strategically placed reflective stripes on the speedometer and tachometer discs were arranged to provide a one-to-one correspondence between actual speed/tachometer rates and count values. Therefore, the values read from counters 1 and 2 need only simple subtraction computations to derive actual speed and tachometer rates. How this scheme was developed is discussed in detail in the computer assisted development section.

```
01CA    F5          PUSH  PSW
01CB    C5          PUSH  B
01CC    D5          PUSH  D
01CD    E5          PUSH  H
01CE    3EDC        MVI   A,0DCH
01D0    D323        OUT   23H
01D2    DB21        IN    21H
01D4    47          MOV   B,A
01D5    3E78        MVI   A,78H
01D7    90          SUB   B
01D8    47          MOV   B,A
01D9    FE63        CPI   63H
01DB    FAEB01      JM    LESS
01DE    78          MOV   A,B
01DF    D664        SUI   64
01E1    47          MOV   B,A
01E2    3E01        MVI   A,01
01E4    D310        OUT   10H
01E6    D350        OUT   50H
01E8    C3F101      JMP   OVER
01EB    3E00  LESS  MVI   A,00
01ED    D310        OUT   10H
01EF    D350        OUT   50H
01F1    78    OVER  MOV   A,B
01F2    CD2002      CALL  BCD
01F5    D310        OUT   10H
01F7    D358        OUT   58H
01F9    DB22        IN    22H
01FB    47          MOV   B,A
01FC    3E63        MVI   A,63H
01FE    90          SUB   B
01FF    CD2002      CALL  BCD
0202    D310        OUT   10H
0204    D360        OUT   60H
0206    3E78        MVI   A,78H
0208    D321        OUT   21H
020A    3E63        MVI   A,63H
020C    D322        OUT   22H
020E    3EC0        MVI   A,0C0H
0210    D311        OUT   11H
0212    3E00        MVI   A,00
0214    D311        OUT   11H
0216    3EC8        MVI   A,0C8H
0218    D320        OUT   20H
021A    E1          POP   H
021B    D1          POP   D
```

Upon entering the subroutine at location 01CAH, the current contents of the registers from the interrupted routine are preserved by placing them on the stack. Unlike the BCD subroutine, there is no value passed in via the accumulator, so the accumulator must also be preserved. The PUSH PSW command places the Program Status Word on the stack which contains the value of the accumulator and all associated flags. The value of counters 1 and 2 are latched by the issuance of a hex DC to to PORT 23H at locations 1CEH-1D1H. Writing this hexidecmal value to PORT 23H issues the previously mentioned READ BACK command which latches both counters simultaneously and sets the 8254 up for a read. At

```
021C    C1          POP   B             location 1D2H, PORT 21H which
021D    F1          POP   PSW
021E    FB          EI                  is the address of counter 1, is
021F    C9          RET
```

read, and the result is stored in register B. To derive the actual speed value, the counter 1 value in register B is subtracted from the maximum speed value of 78H (120 decimal) which represents the maximum speed of the motorcycle.

Since it is possible that the current speed of the motorcycle may be greater than one hundred miles per hour, and the since the BCD routine can only handle values as large as ninety nine, the value must be checked and prescaled to ninety nine or less before submission to the BCD subroutine.

This prescaling also determines the value of the first digit of the three digit speed display. If the value is greater than or equal to one hundred, the first digit is set to one. If the value is less than one hundred, the first digit is set to zero. For example, if the actual speed was one hundred fifteen miles per hour, the first digit would be set to 1 and the value 0FH (15) would be sent to the BCD routine. The three speed displays would display: 115. If the actual speed was sixty five miles per hour, the first digit would be set to zero and the value 41H would be sent to the BCD routine. The three speed displays would then display: 065.

Both the prescaling and the lighting of the first digit are done prior to sending the remaining value (representing the second and

third digits) to the BCD routine. This prescaling and lighting are accomplished via instructions at locations 1D4H-1F1H.

Note the difference in the addressing of the 4511 device at location 1E6H over the previous method of supplying a high and low signal to the 74LS373s. Since the 4511 devices are driven by the 74LS154, only a simple write to a port address is needed to clock the value on the data bus into the device.

The CALL to the BCD routine takes place at location 1F2H. The packed BCD value returns in the accumulator and is placed on the display data bus at location 1F5H.

Since the 4511 devices which drive digits two and three of the speed display are wired to the four most significant and four least significant bus lines respectively, and their address latch pins wired together, a single write to PORT 58H will latch the 8-bit value into the two displays in one operation at location 1F7H.

At location 1F9H, counter 2 is read and the processing of the tachometer value continues in much the same manner as that of the speed.

Since the tachometer is only a two digit display, and the maximum tachometer value is ninety nine, no first digit prescaling is necessary. Only a simple subtract from the maximum value, 63H (decimal 99) is needed prior to submission to the BCD routine.

The read, subtract, and CALL instructions are executed at locations 1F9H-201H.

Upon return from the BCD routine, the accumulator value is placed on the display board data bus at location 202H, and clocked into the display in the same manner as digits two and three of the speed by one write operation to PORT 60 at location 205H.

The instructions at locations 206H-219H reinitialize the three counters in the same fashion as the system initialization sequence previously discussed at locations 263H-283H.

Reinitialization is accomplished by writing the maximum counts 78H and 63H to counters 1 and 2 respectively, clocking the values into the counters through the 74LS86 via the 8255, and writing the master count C8H to counter 0.

The final instructions of the routine restore the register values and program status word by popping them off the stack at locations 21AH-21DH, enable interrupts for acceptance of the next 8254 interrupt via the EI instruction at location 21EH, and return to the interrupted code via the RET instruction at location 21EH.

Keyboard/Security Computation The keyboard and security code is divided into several functional subsections: 1) The keyboard input loop, 2) The key comparison loop, 3) The flash success loop, 4) The flash unsucessful loop, and 5) System shutdown.

```
0071    FE05        CPI   05
0073    CA5C02      JZ    START
0076    C36F00      JMP   006FH
0079    DB18        IN    18H
007B    77          MOV   M,A
007C    23          INX   H
007D    23          INX   H
007E    04          INR   B
```

Although the code at locations 71H-78H is included in the keyboard routine, the actual entry point of the routine is

```
007F   78          MOV   A,B
0080   D310        OUT   10H
0082   D360        OUT   60H
0084   FE05        CPI   05
0086   CA8A00      JZ    GO
0089   C9          RET
008A   21000C  GO  LXI   H,0C00
008D   0600        MVI   B,00
008F   7E      NXT MOV   A,M
0090   23          INX   H
0091   BE          CMP   M
0092   C2BA00      JNZ   BAD
0095   04          INR   B
0096   78          MOV   A,B
0097   FE05        CPI   05
0099   CA0A00      JZ    GOOD
009C   23          INX   H
009D   C38F00      JMP   NXT
00A0   3E65  GOOD  MVI   A,65H
00A2   D310        OUT   10H
00A4   D360        OUT   60H
00A6   CDEE00      CALL  DEL
00A9   3EAA        MVI   A,0AAH
00AB   D310        OUT   10H
00AD   D360        OUT   60H
00AF   CDEE00      CALL  DEL
00B2   14          INR   D
00B3   7A          MOV   A,D
00B4   FE05        CPI   05
00B6   C2A000      JNZ   GOOD
00B9   C9          RET
00BA   OC          INR   C
00BB   79          MOV   A,C
00BC   FE02        CPI   02
00BE   CAFB00      JZ    BAD
00C1   0600        MVI   B,00
00C3   3E18  NXT1  MVI   A,18H
00C5   D310        OUT   10H
00C7   3E08        MVI   A,08
00C9   D311        OUT   11H
00CB   3E00        MVI   A,00
00CD   D311        OUT   11H
00CF   CDEE00      CALL  DEL
00D2   3E00        MVI   A,00
```

location 79H. The instructions at locations 71H-78H are used for memory management. The keyboard code execution is a result of an interrupt from the 8279. Each vector branch from an interrupt places the return address on the stack prior to the branch. Since the 8279 generates an interrupt for each key closure, and the routine allows the entrance of ten keys (two five key attempts), there is the possibility that ten return addresses could be put on the stack and never popped. Because of this, each subsection of the keyboard code is treated like a called subroutine and is terminated with an RET instruction to pop the address off the stack. The 8085

```
00D4    D310        OUT   10H
00D6    3E08        MVI   A,08
00D8    D311        OUT   11H
00DA    3E00        MVI   A,00
00DC    D311        OUT   11H
00DE    CDEE00      CALL  DEL
00E1    14          INR   D
00E2    7A          MOV   A,D
00E3    FE04        CPI   04
00E5    C2C300      JNZ   NXT1
00E8    1600        MVI   D,00
00EA    21010C      LXI   H,0C01H
00ED    C9          RET
00EE    2AF800 DEL  LHLD  00F8H
00F1    2B       L1 DCX   H
00F2    7C          MOV   A,H
00F3    B5          ORA   L
00F4    C2F100      JNZ   L1
00F7    C9          RET
00F8    FF
00F9    FF
00FB    3E03        MVI   A,03
00FD    D312        OUT   12H
00FF    76          HLT
```

stores the address of the next memory location as the return address when an interrupt or CALL instruction is invoked. Since the CPU idles at location 70H following system initialization, the return address following an interrupt will be location 71H. The instructions 71H-78H allow the use of the RET instructions to keep the stack clean, and direct program execution upon return.

There are three possible conditions that will be present upon return to location 71H: 1) The keyboad input loop is still executing and five keys have not yet been entered, 2) Execution is returning from the first unsuccessful attempt at entering the security code, and 3) Execution is returning from a successful attempt at entering the security code.

The decision as to which condition is valid is accomplished by the comparison at location 71H. The value of the accumulator will be five only upon return from the code that processed a successful entry of the five digit code. If the accumulator value is five, control

branches out of the keyboard routine to the system initialization at location 25CH. If the value is not five, a jump is made to location 6FH where interrupts are re-enabled, and control falls through to location 70H (HLT) where the CPU idles and waits for the next interrupt resulting from a key closure.

Locations 79H-89H process the security code input. The first instruction reads the FIFO RAM of the 8279 which contains the value of the key closure. Location 7BH stores the value in the first of the "in between" locations left open in RAM between the stored values of the actual key codes representing the actual security code. Locations 7CH and 7DH increment the H&L registers twice to point to the address of the next open location in RAM for storage of the next key input.

The instructions at locations 7EH-83H were added to keep the operator aware of the number of keys entered. These instructions load the B register (value of the loop counter), put it on the display board data bus and clock it into the two tachometer digits.

The final instructions compare the loop counter value to five to determine whether or not all keys have been entered and either jump over the RET instruction at location 89H to begin the key comparison code if all keys have been entered, or execute the RET instruction which branches control back up to location 71H.

If all keys were entered, the jump instruction branches control to location 8AH to begin the comparison routine. The instruction at location 8AH resets the H&L registers to point to the RAM location

where the first actual security key code was stored (0C00H). The instruction at location 8DH resets the register B value (loop counter) to keep track of the number of comparisons. The instructions at locations 8FH-91H read the contents of the location pointed to by the H&L registers, increment the H&L registers to point to the next location in RAM, and compare the first value with the second. In this manner, the stored actual key codes from system initialization are compared to the stored values of the keys input from the keyboard.

Locations 92H-9FH determine program control based on whether the comparison was or was not successful. If the comparison was not successful, control branches to location BAH via the JNZ instruction at location 92H. If the comparison was successful, the value of the loop counter is compared to five to see whether all RAM key pairs have been tested. If all keys pairs have been tested, control branches to A0H to the success routine. If all key pairs have not been tested, execution fall through to the jump instruction at location 9DH which transfers control back to location 8FH for the next compare.

The instructions at locations A0H-B9H constitute the success routine. These are the instructions that place the value "65" on the display board data bus and flash the value on the tachometer LEDs to show the operator that the correct code was entered and remind him of the legal speed limit. The instructions at locations A0H-A8H load the value, put it on the display board data bus, clock it into the tachometer LEDs and call the delay routine to leave the value

displayed for approximately 1/2 second. The instructions at locations A9H-B9H load the value to blank the display, put the value on the display board data bus, clock the blank value into the tachometer LEDs, call the delay routine to leave the LEDs blank for 1/2 second, then increment and check the value of the D register to determine whether the tachometer display has flashed fives times.

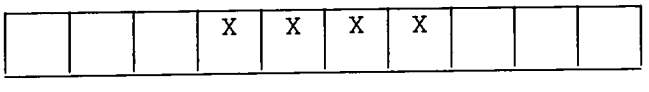
If the display has not flashed five times, control branches back to location A0H for the next cycle via the JNZ instruction at location B6H. If the display has flashed five times, the RET instruction is executed at location B9H which sends control back to location 71H with the value five in the accumulator. This is the value checked at 71H which sends control to the system initialization routine and enables the normal operation of the motorcycle.

The instructions at locations BAH-EDH handle the unsuccessful entrance of the security code. The value of the C register (previously initialized to zero at location 48H is incremented and tested against the value of two at locations BAH-BDH. This is the value which determines whether or not it is the first or second unsuccessful attempt at the security code. If the value equals two, no more attempts are allowed an the execution branches to the unrecoverable halt instructions at location FBH via the JZ instruction at location BEH. If the value is one, one more attempt is allowed, but control first falls through to the flash routine that signals to the operator that the attempt was not correct.

The code at location C1H resets the B register to use as the loop counter for the number of flashes. The instructions at locations C3H-E7H comprise a flash routine and incorporate the same logic as the flash routine for the success condition; the flash code value is put on the display board data bus, clocked, delayed, blanked, delayed, the loop counter incremented, and the sequence is either repeated or control branches back to 71H if the loop counter equals four.

The difference in this routine is that the flash value is written to the warning light unit. Because the warning lights are driven by the 74LS373s, the high to low simulated signal code is incorporated to clock both the light and blank values into the LED units. These additional instructions are at locations C7H-CCH and D2H-DDH. By writing a value 18H to the display board data bus the middle four LEDs of the warning light unit are flashed as in figure 18.

Warning Light Unit



X= LEDs Flashed

Figure 18

The instructions at locations E8H-EDH are executed as control falls though following the unsuccessful flash routine completion. Here the D register is reinitialized to zero, the H&L registers are reset to point to the first RAM location to store the input key codes, and

control branches back to location 71H via the RET instruction to enable the second and last attempt at entrance of the security code.

The instructions at locations EEH-F9H are the delay for the keyboard flash routines. This delay sequence is identical to that of the turn signal routine except for the values at locations F8H and F9H. The delay must be longer for the flash routines, so the values were increased to FFH as compared to the values 11H used in the turn signal routines.

The instructions at locations FBH-FFH are executed if two unsuccessful attempts were made at entering the security code. The instructions at locations FBH-FEH write to and energize the horn relay. The relay contacts close, complete the horn circuit, and the horn blows. The HLT instruction at location FFH puts the system into the unrecoverable halt state. Since control never branched back to location 6FH to enable interrupts, the only way to break the system out of this state is to turn the system off and on again or reset it via the reset switch which will be concealed during final installation.

A/D Conversion and Scaling The A/D conversion routines are the most complex of the system. They are comprised of two main processing routines, with each calling an additional five subroutines.

The difference between the two main processing subroutines is in their manipulation of the mathematical subroutines. The channel one routine accesses the math routines to scale its' temperature value and

produce a Fahrenheit result. Channel two and three process the battery charging voltage and fuel level respectively, and share the same math scaling values.

```
02CE    3A500C      LDA   0C50H       Entrance to a particular sub-
02D1    FE00        CPI   00
02D3    CAF402      JZ    02F4H       routine is decided by the code
                                      segment at locations 2CEH-2DEH.
02D6    CD0304      CALL  0403H
02D9    DB28        IN    28H         This code is branched to when
02DB    5F          MOV   E,A
02DC    3E06        MVI   A,06        the system monitor detects a
02DE    57          MOV   D,A
02DF    CD3703      CALL  0337H       converted A/D channel after in-
02E2    CD8103      CALL  0381H
02E5    CD3704      CALL  0437H       terrogating the polled inter-
02E8    CD7004      CALL  0470H
02EB    C39F02      JMP   029FH       rupt mask. The instruction at
                                      location 2CEH loads the channel
02F4    CD0304      CALL  0403H
02F7    DB28        IN    28H         flag which tells which channel
02F9    D69F        SUI   9FH
02FB    5F          MOV   E,A         has been converted. The compare
02FC    3E12        MVI   A,12H
02FE    57          MOV   D,A         instruction at location 2D1H
02FF    CD3703      CALL  0337H
0302    7B          MOV   A,E         checks for zero (channel one).
0303    C620        ADI   20H
0305    5F          MOV   E,A         If the value is zero, control
0306    CD8103      CALL  0381H
0309    CD3704      CALL  0437H       branches to the channel one
030C    CD7004      CALL  0470H
030F    C39F02      JMP   029FH       main subroutine at location
```

2F4H. If the value was not zero, control falls through to the entry point of the main channel two and three subroutine at location 2D6H.

The channel two and three subroutine first calls the math setup subroutine which initializes memory values and loads the divisor value for this A/D function prior to the CALL to the division math routine.

At location 2D9H, the A/D converter is read by an input (IN) instruction to PORT 28H. This value is then moved to register E which will serve as the multiplicand upon entrance to the multiplication math routine. At locations 2DCH-2DEH the accumulator is loaded with the value six and is moved to the D register. This value will serve as the multiplier upon entrance to the multiplication math routine.

The call to the math routine routine at location 2DFH performs the multiplication and division necessary to convert the value input from the A/D channel to a value that can be further manipulated and displayed.

The CALL to the channel routine at location 2E2H further decodes which channel has been converted, performs the necessary prescaling, calls the BCD routine, displays the value on the proper displays, and sets the next channel convert flag.

The call to the warning light routine at location 2E5H lights or blanks the proper warning light LEDs according to the value of the A/D input.

The CALL to the delay routine slows down the overall A/D and system monitor timing to avoid adverse display affects which will be further discussed in the problems encountered and solved section.

The final instruction at location 2EBH returns control to the system monitor via the JMP to location 29FH.

The channel one routine is essentially the same as the channel two and three routine except for some prescaling and the values which

it passes to the math routines. At location 2F9H the value 9FH (decimal 159) is subtracted from the A/D value to provide the proper base offset for conversion of celsius to Fahrenheit. It then loads the value 12H (decimal 18) as the multiplier at location 2FCH prior to the CALL to the math routine. Upon return from the math routine, the value 20H (decimal 32) is added to the value in the accumulator to complete the conversion from celsius to Fahrenheit at location 303H. The remaining CALLs to the channel, warning light, and delay routines at locations 306H-30EH are the same as in the channel two and three routine, as is the final jump back to the system monitor at location 30FH.

Mathematic Routines Two math routines are utilized in this system; a multiplication and a division routine. The multiplication routine is an 8-bit by 8-bit routine and returns a 16-bit result. The division routine is a 16-bit by 16-bit routine and returns a 16-bit result and 16-bit remainder.

The 8085 has no multiplication operations in it's instruction set. Multiplication can be accomplished by successive additions or by successive bit test algorithms. Successive additions are wasteful of CPU cycles as compared to a bit test algorithm so the latter was chosen for this system.

Both A/D main subroutines call the math set up at location 403H.

```
0403    010A00      LXI   B,000AH      This routine sets up the divis-
0406    21550C      LXI   H,0C55H
```

```
0409    71          MOV   M,C        or for the division routine and
040A    23          INX   H
040B    70          MOV   M,B        stores it in RAM at locations
040C    23          INX   H
040D    3611        MVI   M,11H      403H-40BH. RAM location 0C55 is
040F    C9          RET
```

loaded with the value 0AH (10 decimal) and RAM location 0C56H is loaded with zero for a 16-bit divisor of 000AH or decimal ten. The loop counter value of 11H (decimal 17) is stored at the next RAM location 0C57H.

This code was used since it appeared during development that there would be a need for different divisor values for the two main A/D routines. As it turned out, both routines require a divisor of ten but the set-up routine was left rather than merge it into the division code for modularity.

```
0337    010000      LXI   B,0000     At the entry point of the mult-
033A    2E08        MVI   L,08
033C    7A     NEXT MOV   A,D        iplication routine (location
033D    1F          RAR
033E    57          MOV   D,A        337H) the B&C registers are
033F    D24503      JNC   NOCAR
0342    78          MOV   A,B        initialized to zero. This reg-
0343    83          ADD   E
0344    47          MOV   B,A        ister pair will contain the 16-
0345    78     NOCAR MOV  A,B
0346    1F          RAR              bit result. At location 33AH,
0347    47          MOV   B,A
0348    79          MOV   A,C        the content of the L register
0349    1F          RAR
034A    4F          MOV   C,A        used for the loop counter is
034B    2D          DCR   L
034C    C23C03      JNZ   NEXT       loaded with eight which repre-
```

sents the number of bits in the multiplier. The rest of the instructions form the main logic of the routine.

The logic of this routine is based on the fact that the result of the multiplication of two numbers is based not only on the value of the digits of the multiplier, but also on their weights as they increase by powers of ten as they move from left to right. For example, if the number 213 was multiplied times 111, it would be the equivalent of adding 3 X 111, plus 10 X 111, plus 200 X 111. The result would be:

```
    111
    213
    333          (Zeros implied)
  111
 222
-----
  23643
```

To produce this effect in code, the value of the multiplier is loaded into the accumulator, rotated right one position and stored back in register D at locations 33CH-33EH. The rotate causes the ones position value of the multiplier to be rotated into the carry bit of the accumulator where it is tested for a one or zero value. If there was a one present, the value of the multiplicand is added to the value of the B register which contains the accumulating result at locations 342H-344H. Execution then falls through to locations 345H-34AH where the entire 16-bit word in registers B&C are rotated one position to the right. If the carry check was found to be zero in the JNC instruction at location 33FH, control branches to location 345H where

only the rotate of the 16-bit word will be executed without the addition of the multiplicand.

The final instructions at locations 34B-34E decrement the loop counter, test whether or not all 8-bits of the multiplier have been tested, and control either branches back to location 33CH for the next test, or falls through to the division routine.

By testing only the number of bits in the multiplier and amassing an accumulating result by additions and rotations, this loop has only to be executed eight times for any multiplication of two 8-bit numbers. If successive addition had been used, the loop for the 111 X 213 example would have been executed either one hundred eleven or two hundred thirteen times depending on which number was used as the loop counter.

Even though the multiplier may only have three bits at logic 1, the loop must be executed the full eight times to ensure that the 16-bit result value will be rotated eight bits right into the proper places of magnitude.

As in the case of multiplication instructions, the 8085 does not utilize any division instructions in it's instruction set. Division can be accomplished by successive subtractions which like the multiplication successive additions cause lenghty CPU loop cycles, or it too can be accomplished by a bit testing algorithm which executes the test loop only sixteen times, which is the number of bits in the

divisor. The latter is the algorithm used in this system.

```
034F    59          MOV   E,C
0350    50          MOV   D,B
```

When the multiplication algorithm has completed, control falls through to locations 34FH and 350H where the multiplication result in register pair B&C is moved to register pair D&E to become the value of the dividend for the division routine.

The division algorithm is implemented by successive subtracts and tests. Initially, the divisor is subtracted from the first bit of the dividend and a test for a borrow is performed.

If a borrow is generated, the dividend value was less than the divisor so the divisor is added to the result of the division to restore the dividend and the divisor is shifted one position to the right for subtraction and borrow test against the next bit of the dividend. If a borrow is generated, the carry bit of the accumulator will be zero as a result. The carry bit is then rotated into the D&E register pair to form part of a building quotient.

If the result of the subtraction did not generate a borrow, then the value of the dividend was greater than the divisor. In this case the divisor is not added to the result of the subtraction to restore the dividend. The result of the subtraction now becomes the new dividend, and the divisor is shifted one position to the right for the next subtract and test operation. If the result of the subtraction did not generate a borrow, the value of the carry bit will be one, and

this value rather than zero, is rotated into the D&E registers to form part of the building quotient.

Upon completion of the division routine, the 16-bit quotient will be contained in the D&E register pair, and the remainder will be contained in the B&C register pair.

```
0351    010000       LXI   B,0000       At the entry point of the div-
0354    21570C LOOP  LXI   H,0C57H
0357    7B           MOV   A,E          ision routine at location 351H,
0358    17           RAL
0359    5F           MOV   E,A          the B register is initialized
035A    7A           MOV   A,D
035B    17           RAL                to zero for use in containing
035C    57           MOV   D,A
035D    35           DCR   M            the partial dividend throughout
035E    CA7F03       JZ    BACK
0361    79           MOV   A,C          the process. At location 354H
0362    17           RAL
0363    4F           MOV   C,A          the H&L register pair is loaded
0364    78           MOV   A,B
0365    17           RAL                with the value of the loop
0366    47           MOV   B,A
0367    2B           DCX   H            counter that was stored in the
0368    2B           DCX   H
0369    79           MOV   A,C          math set-up routine. The
036A    96           SUB   M
036B    4F           MOV   C,A          instructions at locations 357H-
036C    23           INX   H
036E    9E           SBB   M            366H rotate the 16-bit divi-
036F    47           MOV   B,A
0370    D27B03       JNC   COMP         dend, check the loop counter to
0373    2B           DCX   H
0374    79           MOV   A,C          see whether all bits have been
0375    86           ADD   M
0376    4F           MOV   C,A          tested, and either jump to the
0377    23           INX   H
0378    78           MOV   A,B          RET instruction at location
0379    8E           ADC   M
037A    47           MOV   B,A          37FH, or rotate the 16-bit par-
037B    3F      COMP CMC
037C    C35403       JMP   LOOP         tial dividend one position to
037F    C9      BACK RET
```

the left to set it up for subtraction. As the 16-bit dividend in register pair D&E is rotated left one position in the instructions at locations 357H-35CH, the most significant bit is shifted into the carry bit of the accumulator by the RAL instruction. If the loop has not been exhausted, the 16-bit partial dividend is rotated at locations 361H-366H. Since the carry bit contained the most significant bit from the dividend in the D&E registers, when the B&C registers are rotated left the carry bit is rotated into the least significant bit of the partial dividend. The repeated rotation of the dividend in the D&E registers followed by the rotation of the B&C registers cause a continual building of the partial dividend in B&C and effectively shift the D&E value one bit at a time into the B&C registers for manipulation.

The instructions at locations 367H-36FH decrement the H&L registers to point to the divisor value in memory and subtract the 16-bit divisor value from the partial dividend in registers B&C.

If the carry bit is set to zero as a result of the subtraction, the divisor was less than the partial dividend, control branches to 37BH, and the new value of the partial dividend remains the result of the subtraction.

If the carry bit is set to one, a borrow has been generated meaning that the value of the divisor was greater than the dividend, so instructions at locations 373H-37AH must be executed. These instructions add the value of the divisor in memory to the results of

the subtraction in register pair B&C to restore the value of the partial divisor for the next execution of the loop.

At location 37BH, the value of the carry bit is complemented via the CMC instruction. If the result of the subtraction was zero (no borrow), the bit is set to one. If a borrow occurred and the partial dividend was restored, the carry bit (one) would be reset to zero.

The jump instruction at location 37CH returns control to location 354H for the next execution of the loop.

The last instruction (RET) returns control to the calling main A/D subroutine.

The carry bit is complemented at location 37BH, since, when control returns to 354H and the loop is repeated, the first RAL instruction at location 358H causes the carry bit to be shifted into the least significant bit of the E register as the most significant bit of the E register is being shifted out. In this way, as the dividend value in registers D&E is being shifted out to form the partial dividend in register pair B&C, the carry bits as the result of the subtractions are being shifted into register D&E to become the quotient value of the division.

When the entire process has completed, the quotient value will be contained in register pair D&E, and the remainder, which is the final value of the partial dividend left over, will be contained in register pair B&C.

Both the multiplication and division routines were discovered

during reading. Since both routines were efficient and well written they were borrowed and modified for this system. The routines were found in the 8080/8085 Software Design Book 1 by Larsen, Titus and Titus. These two routines and the delay routines comprise the only software in the entire system which is not totally original in design.

Upon return from the math routines, the channel one and channel two and three main subroutines call the main channel routine at location 381H. This channel routine further conditions the values returned by the math routines by rounding and prescaling prior to the CALLs to the BCD routine and subsequent display.

It then performs a channel check to determine which channel value is being manipulated, and branches control to the subroutine for the particular channel.

```
0381    37          STC
0382    3F          CMC
0383    3E04        MVI   A,04
0385    91          SUB   C
0386    D28D03      JNC   NRND
0389    7B          MOV   A,E
038A    C601        ADI   01
038C    5F          MOV   E,A
038D    7B      NRND MOV  A,E
038E    37          STC
038F    3F          CMC
0390    FEC7        CPI   0C7H
0392    D29D03      JNC   L200
0395    D6C8        SUI   0C8H
0397    4F          MOV   C,A
0398    3E02        MVI   A,02
039A    C3B003      JMP   CHK1
039D    37      L200 STC
```

The entry point of the channel routine is location 381H. The instructions at locations 381H-3B0H perform the rounding and prescaling. Incorporated into these instructions, like the math instructions, is the extensive use of the carry bit for testing the results of sub-

```
039E    3F          CMC
039F    FE63        CPI   63H
03A1    D2AD03      JNC   L100
03A4    D664        SUI   64H
03A6    4F          MOV   C,A
03A7    3E01        MVI   A,01
03A9    47          MOV   B,A
03AA    C3B103      JMP   CHK2
03AD    4F     L100 MOV   C,A
03AE    3E00        MVI   A,00
03B0    47     CHK1 MOV   B,A
03B1    3A500C CHK2 LDA   0C50H
03B4    FE00        CPI   00
03B6    CA5A04      JZ    CHN1
03B9    FE01        CPI   01
03BB    CAED03      JZ    CHN2
```

traction. This was an experiment on my part to hone my skills using this flag for branching purposes. At location 381H the carry bit is initialized to one, and complemented to zero at location 382H to begin the tests. Locations 383H -38DH perform the round function. The value four is loaded into the accumulator at location 383H and is used as the test value for the round. If the value subtracted from four generates a carry, then the value is larger than four (five or greater). If a carry was not generated, the value is four or less.

The value subtracted is the remainder from the division routine contained in register C. This operation is performed at location 385H. If the carry bit was not set, the remainder value is four or less, no rounding takes place, and control branches via the JNC instruction at location 386H to location 38DH.

If a carry was generated, the quotient passed in register E is loaded into the accumulator, incremented by one and stored back in register E at locations 389H-38CH. The remainder value is lost after these instructions and only the value in register E is manipulated.

Location 38DH begins the prescale function code. Since the value passed in by the temperature A/D function may be two hundred or more (greater than or equal to two hundred degrees Fahrenheit), the first prescale check is against the value C7H (decimal 199) at location 390H.

The compare instruction performs a subtract, so if the value of the carry bit is one, the value being subtracted is greater than or equal to, two hundred. Control branches via the JNC (Jump No Carry) to the greater than or equal to one hundred test if no carry was generated.

If a carry was generated, two hundred is subtracted from the accumulator and the value is stored in register C at locations 395H-397H. The value two is then loaded into the accumulator at location 398H. This is the value of the first digit of a three digit 7-segment display, and will be passed to the proper channel routine for display.

Control then branches via the JMP instruction at location 39AH to the channel check which will pass control to that proper channel routine.

At locations 39DH-39EH, the carry bit is set and reset for the greater than or equal to one hundred test at locations 39FH-3ACH. The logic of this test sequence is identical to that of the greater than or equal to two hundred test, except that the value is compared against ninety nine, and if the condition is true, the first digit value in the accumulator is set to one. If the test is not true, then

the value is less than one hundred, and control branches to location 3ADH where the value of the first display digit in the accumulator is set to zero. In either case, the test value is stored in register C.

The final instruction of the test code sequences stores the value of the first display digit in register B at location 3B0H.

The code at locations 3B1H-3BDH check for the current channel whose value is being manipulated by loading the current channel flag which was stored in memory location 0C50H. It then performs three compares for either a zero, one, or two, which correspond to logical channels one, two, or three, and branches control via JZ instructions to the proper channel routine.

The channel one (temperature) and two (charging voltage) routines are identical in logic. They retrieve and display the value of the first digit of the three digit display stored in register B, call the BCD routine with the value stored in register C, clock the returning value into the second and third displays, load the flag value of the next A/D channel to be converted in RAM location 0C50H, and finally, address the A/D converter via the same value to begin conversion of the next channel, and return to the calling subroutine.

```
045A    78      CHN1 MOV   A,B          The channel one routine moves
045B    D310         OUT   10H
045D    D338         OUT   38H          the value of the first digit of
045F    79           MOV   A,C
0460    CD2002       CALL  BCD          the display into the accumula-
0463    D310         OUT   10H
0465    D330         OUT   30H          tor from register B, writes to
0467    3E01         MVI   A,01
```

```
0469    32500C      STA   0C50H       the display board data bus, and
046C    D328        OUT   28H
046E    C9          RET               clocks the first temperature
```

display at PORT 38H at locations 45AH-45DH. The value in register C is moved to the accumulator, the BCD routine called, the value is written to the display board data bus, and clocked into digits two and three of the temperature display via PORT 30H at locations 45FH-465H.

The value of the next channel to be converted (01 for channel two) is loaded and stored as the current channel flag at locations 467H-469H. The write to PORT 28H (A/D converter) at location 46CH starts the conversion of the next A/D channel (channel two). The RET instruction at location 46EH sends the control back to the calling main subroutine while the conversion of the next A/D channel is in progress.

```
03ED    78      CHN2 MOV   A,B         The channel two routine is
03EE    D310         OUT   10H
03F0    D340         OUT   40H         identical except for port
03F2    79           MOV   A,C
03F3    CD2002       CALL  BCD H       addresses written to for access
03F6    D310         OUT   10H
03F8    D348         OUT   48H         to the displays, and the value
03FA    3E02         MVI   A,02
03FC    32500C       STA   0C50H       stored in RAM location C50H as
03FF    D328         OUT   28H
0401    C9           RET               the current channel flag.
```

The value 02 is used as the current channel flag for channel three, and is also used to address the A/D converter to begin the conversion for channel three.

The channel three routine (fuel level) uses entirely different logic since the value passed into the routine does not have to be displayed on 7-segment LEDs. Instead, the value must be checked against predefined limits to determine the level of fuel in the tank and jump to flag set routines which set the fuel level flags accordingly.

The channel three function uses the same voltage input range to the A/D converter as the channel two function. It is therefore known that if the value passed in from the prescaling routine created a one in the B register, that the value in the C register is well above range for fuel level testing, that is, there is lots of fuel in the tank. As a shorthand method, the B register is checked first upon

```
03BE    3E01         MVI   A,01
03C0    B8           CMP   B
03C1    CADC03       JZ    LOTS
03C4    3E27         MVI   A,27H
03C6    B9           CMP   C
03C7    FADC03       JM    LOTS
03CA    3E14         MVI   A,14H
03CC    B9           CMP   C
03CD    FAD603       JM    SOME
03D0    CD1104       CALL  L18
03D3    C3DF03       JMP   CHAN
03D6    CD1804 SOME  CALL  L14
03D9    C3DF03       JMP   CHAN
03DC    CD1F04 LOTS  CALL  SHUT
03DF    3E00   CHAN  MVI   A,00
03E1    32500C       STA   0C50H
03E4    D328         OUT   28H
03E6    C9           RET
```

entry to the channel three routine at location 3BEH. If the value is one, a jump is made to location 3DCH and the level checks are bypassed, At location 3C4H, the value 27H is loaded which represents 1/4 tank of fuel or less. A jump on minus is made to the same

location as the register B check if the value in register C is greater than 27H, meaning that there is greater than or equal to 1/4 tank of fuel remaining.

If the value is less than 27H, then there is less than 1/4 tank of fuel remaining, and the value is then compared, as control falls through, to 14H at location 3CCH which is the value representing 1/8 tank of fuel.

If the register C value is greater 14H, a jump on minus is made at location 3CDH to location 3D6H, which means that the level of fuel in the tank is less than 1/4 but greater than or equal to 1/8.

If the compare at location 3CCH did not generate a jump on minus, control falls through to the CALL at location 3D0H.

Each of the jumps on minus generate a CALL to a routine which sets flags according to the level of fuel in the tank. These flags are further manipulated by the warning light LED lighting routine which will be discussed in the warning light table look-up section.

The jumps made to location 3DCH (if there is a lot of fuel in the tank) cause a CALL to the warning light shut-off flag set routine which sets the needed flag (to shut off the fuel level warning light LEDs) used by the warning light shut-off routine which is also discussed in a later section.

By means of these checks and calls, the status of the fuel level flags are updated each time the channel three subroutine is executed.

The final instructions at locations 3DFH-3E6H load and store the current channel flag in RAM location 0C50H, start the next A/D conversion, and return to the calling subroutine.

Since this is the channel three routine, the flag (00) for the first channel is loaded and stored, and the channel one conversion is started prior to return to the calling subroutine.

Warning Light Logic The warning light code manipulates flag values which are reset upon system initialization, and set or reset throughout system execution depending on the level of fuel in the fuel tank and the status of the kick stand (either up or down).

Warning Light Flag Set Routines The CALLs in the channel three routine at locations 3D0H, 3D6H, and 3DCH, call the 1/8 tank flag set, the 1/4 tank flag set, and the warning light shut-off flag set routines respectively.

The checks for the status of the kick stand are performed in the system monitor and call the kick stand down flag set routine at location 429H, or the kickstand up flag set at 430H depending on the status.

```
0411    3E01        MVI   A,01        Each mini-subroutine loads the
0413    32590C      STA   0C59H
0416    C9          RET               status condition flag value
```

```
0418    3E01        MVI   A,01
041A    32580C      STA   0C58H
041D    C9          RET

041F    3E00        MVI   A,00
0421    32580C      STA   0C58H
0424    32590C      STA   0C59H
0427    C9          RET

0429    3E01        MVI   A,01
042B    325A0C      STA   0C5AH
042E    C9          RET

0430    3E00        MVI   A,00
0432    325A0C      STA   0C5AH
0435    C9          RET
```

into the proper RAM memory location for that flag. A flag store with a "1" value denotes that the function represented is valid such as the kick stand is down, there is less than 1/4 tank of fuel etc. The storage of a zero value is used as a clear, meaning that the function is not valid, such as greater than 1/4 tank of fuel, or the kick stand is up.

Each of the above mini-routines loads the flags into the proper memory locations, and returns to the calling program.

The routines at locations 411H and 418H set the flags for less than 1/4 tank of fuel and less than 1/8 tank of fuel respectively.

The routine at location 41FH resets the flags for the less than 1/4 and less than 1/8 tank of fuel conditions. This routine is executed when there is greater than or equal to 1/4 tank of fuel.

The last two routines at locations 429H and 430H set and reset the kick stand down conditions respectively.

Warning Light Look-Up Table There are two functions whose status is displayed via the warning light LEDs: 1) Fuel level, and 2) Kick Stand. By further defining the possible warning light statuses of these functions at any one time, we get six possible

permutations. In the following example, it is understood that an "on" condition for fuel means less than, and an "off" condition means greater than or equal to. For the kick stand, Ksd stands for kick stand down, and Ksu stands for kick stand up.

The six possible permutations are as follows:

1) 1/4 tank off, 1/8 tank off, Ksu

2) 1/4 tank off, 1/8 tank off, Ksd

3) 1/4 tank on, 1/8 tank off, Ksu

4) 1/4 tank on, 1/8 tank off, Ksd

5) 1/4 tank on, 1/8 tank on, Ksu

6) 1/4 tank on, 1/8 tank on, Ksd

Note that the condition where the 1/8 tank is on and the 1/4 tank is off is impossible in this system since the 1/4 tank warning light must come on before the 1/8 tank light, and it stays lit as the level of fuel in the tank is lowered through the 1/8 tank condition and beyond.

By assigning a "1" for the on condition and a "0" for the off

condition of each function, we get the following binary combinations: (decimal equivalents are given in parenthesis)

(0)	000	1/4 tank off, 1/8 tank off, Ksu
(1)	001	1/4 tank off, 1/8 tank off, Ksd
(4)	100	1/4 tank on, 1/8 tank off, Ksu
(5)	101	1/4 tank on, 1/8 tank off, Ksd
(6)	110	1/4 tank on, 1/8 tank on, Ksu
(7)	111	1/4 tank on, 1/8 tank on, Ksd

```
0316    3E00        MVI   A,00        The instructions of the routine
0318    32600C      STA   0C60H
031B    3E24        MVI   A,24H       at locations 316H-334H store
031D    32610C      STA   0C61H
0320    3E18        MVI   A,18H       unique codes corresponding to
0322    32640C      STA   0C64H
0325    3E3C        MVI   A,3CH       the activation of the warning
0327    32650C      STA   0C65H
032A    3EDB        MVI   A,0DBH      light LEDS for the above con-
032C    32660C      STA   0C66H
032F    3EFF        MVI   0FFH        ditions in RAM memory locations
0331    32670C      STA   0C67H
0334    C9          RET               C60H, C61H, C64H, C65H, C66H
```

and C67H respectively. This set-up routine is called from the second system initialization code segment at location 292H. The boundaries of these RAM locations C60H-C67H form the warning light look-up table.

As execution progresses, three RAM memory locations are manipulated for flag storage areas according to the status of the particular functions. Location C58H is the location used to store the 1/4 tank status flag. Location C59H is used to store the 1/8 tank status flag, and location C5AH is used to store the kick stand status flag. These three locations are accessed by the warning light status

routine and their contents are used to generate the warning light look-up table address to retrieve the proper warning light LED activation code.

```
0437    21580C      LXI   H,0C58H     At location 437H, the H&L pair
043A    7E          MOV   A,M
043B    07          RLC               is loaded with the address of
043C    47          MOV   B,A
043D    23          INX   H           the first of the three RAM flag
043E    7E          MOV   A,M
043F    80          ADD   B           storage locations. The instruc-
0440    07          RLC
0441    47          MOV   B,A         tions at locations 43AH-445H
0442    23          INX   H
0443    7E          MOV   A,M         load the contents of the first
0444    80          ADD   B
0445    47          MOV   B,A         location (C58H), rotate the
0446    21600C      LXI   H,0C60H
0449    7D          MOV   A,L         value one position to the left,
044A    80          ADD   B
044B    6F          MOV   L,A         store the rotated value in the
044C    7E          MOV   A,M
044D    D310        OUT   10H         B register, increment the H&L
044F    3E08        MVI   A,08
0451    D311        OUT   11H         register pair to point to the
0453    3E00        MVI   A,00
0455    D311        OUT   11H         next address (C59H), load the
0457    C9          RET
```

value, add the value in register B, rotate the value one position to the left, store the result in register B, increment the H&L registers to point to the last location (C5AH), then load the value, add the value in register B and restore the result back into in register B.

This sequence builds a unique value from zero to seven based on the values of the flags at the RAM memory locations and is used as the offset to the base address of the look-up table.

At location 446H, the value of the base address of the look-up table (C60H) is loaded into the H&L registers. At location 449H, the least significant byte of the 16-bit address in register L is moved to the accumulator, and the value that was built up in register B is added to this value in location 44AH.

The H&L registers now contain the base address and offset used to access a particular address within the look-up table.

At location 44CH, the contents of the look-up table address is loaded into the accumulator, and at location 44DH, that value is put on the display board data bus via an OUT instruction to PORT 10H.

The warning light unit is clocked with this value at locations 44FH-456H which provides the high to low transition needed by the warning light 74LS373.

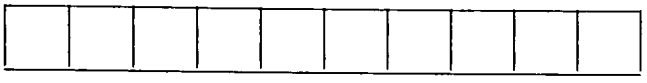
At location 457H, control returns to the calling program via the RET instruction.

The Warning light unit LED lighting configurations for the six possible conditions are found in figure 19.

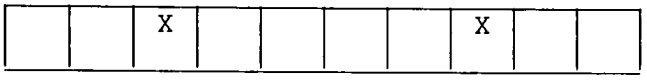
System Delays The last instructions in the channel one and channel two and three main subroutines are calls to the system delay routine at location 470H. This delay routine as noted, was added to slow

```
0470    E5          PUSH  H
0471    2A7E04      LHLD  047EH
0474    2B    LOOP  DCX   H
0475    7C          MOV   A,H
0476    B5          ORA   L
0477    C27404      JNZ   LOOP
047A    E1          POP   H
047B    C9          RET
047C    FF
047D    FF
```

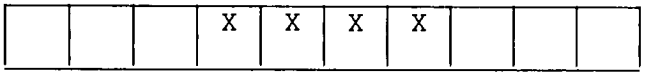
Warning Light Unit



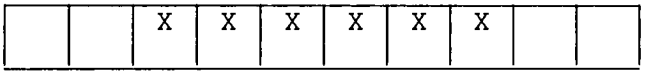
1/4 tank off, 1/8 tank off, Ksu



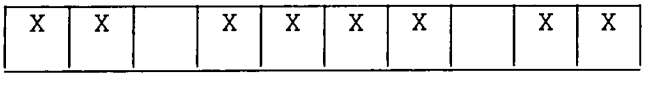
1/4 tank off, 1/8 tank off, Ksd



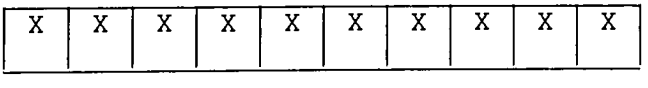
1/4 tank on, 1/8 tank off, Ksu



1/4 tank on, 1/8 tank off Ksd



1/4 tank on, 1/8 tank on, Ksu



1/4 tank on, 1/8 tank on, Ksd

X = LED Lit

Figure 19

down the execution of the system monitor timing to avoid adverse affects on the displayed values of the functions from the A/D converter. This routine is identical to the delay routine used in the turn signal delay call. The values at locations 47CH and 47DH were manipulated to find the best results. The final delay values chosen (FFH), were identical to the turn signal delay values. Although the two delay routines and their values are identical, due to the modular development of the system code, the calls from the two main subroutines and the delay routine were in place, so the code was left as is, rather than directing the CALL to the turn signal delay routine.

System Monitor The system monitor is the operating system loop which monitors all system functions and sequences all calls to service subroutines. PORT C is used to input the status of the left and right turn signal and kick stand switches. Execution of the system monitor begins at location 29FH, where the value of PORT C of the 8255 programmable peripheral interface is read into the accumulator via an IN instruction to PORT 12H. This value is stored for later retrieval into regis-

```
029F    DB12      IN    12H
02A1    47        MOV   B,A
02A2    E620      ANI   20H
02A4    FE20      CPI   20H
02A6    CC1001    CZ    0110H
02A9    78        MOV   A,B
02AA    E640      ANI   40H
02AC    FE40      CPI   40H
02AE    CC3B01    CZ    013BH
02B1    DB12      IN    12H
02B3    E610      ANI   10H
02B5    FE10      CPI   10H
02B7    C2C002    JNZ   02C0H
02BA    CC2904    CZ    0429H
02BD    C3C302    JMP   02C3H
02C0    CD3004    CALL  0430H
02C3    20        RIM
02C4    E620      ANI   20H
02C6    FE20      CPI   20H
02C8    CACE02    JZ    02CEH
02CB    C39F02    JMP   029FH
```

ter B at location 2A1H. The polled interrupt mask logic of the system monitor is implemented by reading the PORT C value (interrupt mask), and ANDing (masking) out all but the bits which represent the turn signals and the kick stand.

A value 20H denotes the closure of the left turn signal switch, a 40H the right turn signal switch, and a 10H the kick stand switch. An AND of the accumulator with the same value, 20H, 40H, or 10H, will mask out all but the bit in question. If the bit result following the AND is one, the switch is closed, while if the value of the result is zero, the switch is open.

The instructions at 2A2H-2B0H test for the left and right turn signals. At locations 2A2H-2A8H, the mask is ANDed with 40H to mask out all but the left turn signal bit, a compare is performed with the same value, and control branches to the left turn signal routine at location 110H if the result of the compare is zero. If the result was not zero, the original value of the interrupt mask read from PORT 12H is restored from register B at location 2A9H.

The same process of AND, compare and CALL is re-executed using the 20H value as the value for the right turn signal at locations 2AAH-2B0H. If the value of the compare at 2ACH was non zero, the CALL instruction is not executed, and control falls through to the next read of the interrupt mask at location 2B1H.

The additional read of the interrupt mask at location 2B1H is executed to handle the condition that the kick stand switch may be

closed during the turn signal routine. If the read were not inserted, the original value of the mask in register B would be interrogated (which did not reflect the kick stand down) upon exit from the turn signal routine and the kickstand would be missed. The execution of the system monitor would loop back and check the turn signals first, and if one was found to be active, execute the turn signal flash routine a second time before the kick stand was ever recognized.

At locations 2B3H-2B6H, the turn signal bit is masked out and compared to the turn signal valid value; 10H. Since the warning light routine executed from the channel one and channel two and three main A/D conversion subroutines activates the warning light LEDs, the calls resulting from the kick stand bit value comparison are to the flag setting routines for the particular status.

If the value of the bit compare at location 2B5H is zero (kick stand up), a jump is made to location 2C0H where a CALL is made to the kick stand up flag set routine at location 430H. If the value of the compare is one, control falls through to the CALL at location 2BAH which calls the flag set routine at location 429H which stores the value for the kick stand down condition.

Upon return from this CALL, the jump at location 2BDH is executed to jump over the CALL instruction to the kick stand up flag set routine.

Regardless of which set of instructions were executed as a result of the kick stand check, system monitor execution resumes at location

2C3H where the value of the CPU polled interrupt mask is read via a RIM (Read Interrupt Mask) instruction. It is this mask which is interrogated for the presence of a finished A/D channel conversion.

Only one bit of the interrupt mask is checked at this time for the existence of a converted channel since the A/D conversion subroutines further decode the current channel flags for branch to the proper service routines.

If the A/D converter has finished converting a channel and sent an interrupt to the CPU via RST 6.5, the value 20H will exist following the masking operation at location 2C4H.

If the result of the compare to 20H at location 2C6H is zero (converted channel interrupt pending), control branches to the channel check at location 2CEH and the A/D converter software routines are executed.

If the result of the compare was not zero, a channel conversion is still in progress, so control branches via the jump instruction at location 2CBH to the beginning of the system monitor at location 29FH and the system monitor loop execution continues.

Databases

System RAM/EPROM Usage There are no high level file structures or databases created in this system. Since this is an application specific firmware system, the EPROM represents the main system input file and executable code, and the RAM represents the database construct.

The contents of RAM are dynamic during system execution but static in it's segmentation of available work area. Figure 20 shows the partitioned RAM system usage map.

Communications Among Modules Communication among system routines is implemented by value passing and the use of common access areas.

Common Access Method The warning light flag set routines, the warning light LED activation/table look-up routines and the A/D

System RAM Map

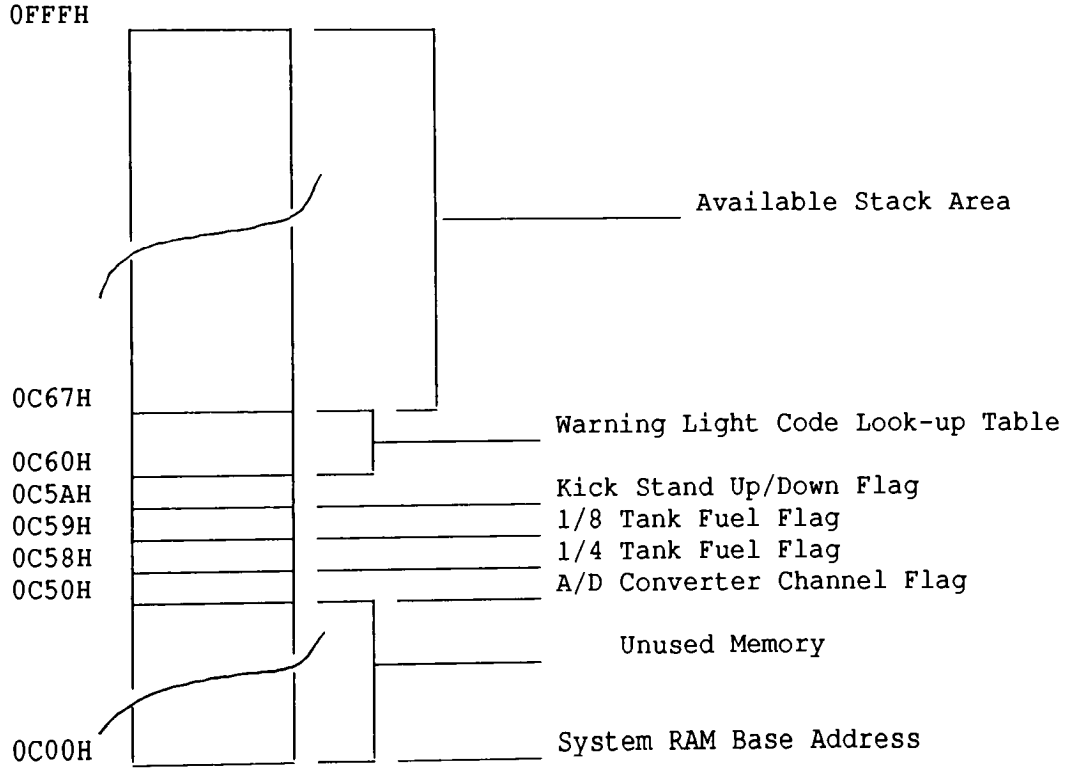


Figure 20

channel logic routines are implemented using the common access method.

In this communications scheme, two or more modules access the same memory locations to store updated data or for retrieval of data as input to a module which interprets the data within it's logic process.

Memory location 0C50H is used for storage of the current A/D channel status flag, and it's contents are updated at the end of each channel's A/D conversion display routine. The value of location 0C50H is retrieved and used for decision making from within the system monitor and A/D conversion prescaling routines.

Memory locations 0C58H-0C5AH are similarly used for the 1/4 tank and 1/8 tank fuel level flags, and the kick stand status flag respectively. These locations are updated via calls from the system monitor and the A/D conversion main subroutines. The values in these locations are accessed as input to the the warning light LED activation/table look-up routine.

Value Passing The second communication method used within this system is value passing. In this method, the value to be passed is loaded into a temporary storage area, control branches to the manipulation routine, and the new value is passed back to the calling routine, again, within the temporary storage area.

This method can be implemented using RAM memory locations, or as in this system, by using the 8085's internal registers.

The hexadecimal to BCD conversion routine utilizes the value passed into the module via the accumulator (register A), and returns the BCD equivalent to the calling module via the same vehicle.

Handling Context Switches Neither the software invoked CALL instruction nor the CALL operation as a result of an interrupt automatically save any of the contents of the current environment present within the 8085 internal registers. Context switches therefore are an overhead operation maintained by the programmer.

Storage of the current contents of the 8085's registers can be accomplished upon entry to a subroutine by moving the contents to RAM memory locations via the move or store instructions, or they can be put on the stack via the PUSH instruction as in this system. The PUSH operation was chosen since it requires less CPU cycles for execution and there existed ample RAM area reserved for stack operations.

Additionally, the programmer must be particularly cognizant of the use of the registers within the subroutines so that only the values in the proper registers are saved and restored.

Context switches are used in this system upon entry to the speedometer and tachometer, and hexadecimal to BCD conversion subroutines.

Upon entry to the speedometer and tachometer subroutine, all of the internal registers are saved on the stack as well as the value of the accumulator which is pushed via the PUSH PSW (Push Program Status Word) instruction.

Since the accumulator is used as the transfer vehicle for the value returned from the hexadecimal to BCD conversion, all registers except the accumulator are saved upon entry to the routine.

Upon exit from these routines, the environments are restored via the POP counterpart to the PUSH instructions. The POP operations are executed in reverse order from the PUSH operations since the stack is implemented using Last-In-First-Out (LIFO) sequencing.

System Monitor Timing

Variations Depending on System Conditions Speed of execution of the system monitor is affected by the delay routine executed from within the two A/D conversion main subroutines as well as the delay executed from within the turn signal routines.

If there are no turn signal flash routines operating and an A/D channel conversion has not completed, the monitor runs at maximum speed operating as a small closed loop.

Slowest operation of the system monitor is evident when both the turn signal and A/D conversion subroutines are executed within one cycle of the monitor loop.

Since the slowly changing A/D converter values are not critical, and the real time speedometer and tachometer values are captured via the hardware interrupt service, the system monitor can effectively operate at varying execution speeds without loss of system integrity.

Integration and Test

Module Development Most of the system modules were developed using the SDK85 system design kit. The SDK85 kit offers 1/4K user memory for storage of application programs, utlities for examining registers and memory locations, and a single step function.

As the hardware was developed and breadboarded, application modules were developed to drive the hardware. Using the SDK85 utilities, programs were entered and executed. Modifications, if needed, were implemented in the software and or hardware to achieve the desired results.

The modules were developed in the sequence in which they appear in the final system.

Module Test Module test was performed by running the application programs and observing the final results. Most modules operate on very defined input values such as mathematical values for the speedometer and tachometer, temperature and charging voltage, and fuel level functions.

Full scale output value representations were developed via computer program to define expected outputs for any given input over the entire operating range of a given function. Verification and validation of the modules for these functions consisted of comparison between actual results of a module's outputs and the predefined values in the computer program listings.

Full scale output listings were developed for and used to test; the speedometer, tachometer, temperature, charging voltage and fuel level functions.

The keyboard input module is subject to variable input values and was tested with various keyboard closure input values, both valid and non valid, to assure that the code could handle all exceptions and branch to the proper subroutine on any given input.

The turn signal modules were tested and validated by observing that the correct loop and rotation values were used to light the proper LEDs and access the proper LED units in the proper sequence, and also by observing that a switch closure resulted in a software branch to the correct routine.

System delay routines were tested and observed, and the hard coded delay input values were adjusted accordingly until the desired delay times were achieved.

The kick stand down function was tested by observing that a switch closure resulted in the correct pattern of LED lighting within the warning light unit.

System Integration System integration was achieved by migrating all code developed in the SDK85 system into the 2816 EEPROM upon completion of all hardware functional circuitry. In addition, the A/D conversion handling modules were developed at this time using the actual hardware configuration rather than via the SDK85 interface devices.

During SDK85 to actual circuit migration, care was taken to change address values and code segments to the actual values in the EEPROM.

Development on the SDK85 allows application program input beginning at RAM memory location 2000H. Consequently, all code developed on the SDK85 contained address values for branches, moves, loads and stores which had to be changed to match the new EEPROM actual locations.

All access to the 4511 BCD to 7-segment latch/decoder driver devices during development on the SDK85 was via 8155 and 8355 I/O devices which mandated that the high to low transition signals be simulated throughout. System integration was the first time that advantage could be taken of the 74LS154 address decoder for access to these devices via a simple OUT instruction. During migration, all access code to the 4511 devices was transcribed substituting the OUT instruction accordingly.

For ease of integration and correctness, the migration process was accomplished in two steps. First, the SDK85 code was transcribed with the needed changes to new programming sheets with actual memory locations beginning at location 0000. The code was then transferred into the EEPROM for system integration test.

System integration was the first opportunity to actually observe total system interaction. Until that time it was unknown what effects total system timing would have on the system displays, and contingency plans for alternate coding had been made, if needed, for critical

areas such as turn signal flash if the speedometer and tachometer interrupt service routine caused visual delay during the flash cycle.

Moving the input of the polled interrupt mask for the kick stand down function was one result of system integration, as it was observed that the turn signal flash could be executed twice before the kick stand down warning lights were activated.

As it turned out, the speedometer and tachometer subroutines executed fast enough that the effects on the turn signal or any other routine were not visible to the eye.

Changes, if any, were made by removing the EEPROM from the system, recoding, and then reinserting it for further tests of those changes.

Computer Assisted Development

By writing adjunct programs, it was possible to derive full range input and output values for all functions within the system for verification and calibration. Programs were also written for simulation purposes for hardware debug of the display board and generation of the speedometer and tachometer hardware interfaces.

A/D Conversion The charging voltage and fuel level interfaces share the same functional input values from their respective input sensors. A program was constructed to observe the complete zero to five volt input range and the resulting effects on various system values over that range. Figure 21 is a segment of the output listing from that program. The complete listing can be found in Appendix E.

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
1	20	0	.00	0.6	1	0.1
2	40	0	.00	1.2	1	0.1
3	60	0	.00	1.8	2	0.2
4	80	0	.00	2.4	2	0.2
5	100	0	.10	3.0	3	0.3
6	120	0	.10	3.6	4	0.4
7	140	0	.10	4.2	4	0.4
8	160	0	.10	4.8	5	0.5
9	180	0	.10	5.4	5	0.5
10	200	0	.20	6.0	6	0.6

Figure 21

By using such a listing one can see a correlation between actual input values and the expected output values for any given value within the range.

Since the system precision of the A/D converter function is 20 millivolts, column two of the listing shows a running total of actual millivolts over the entire range in system precision units.

Column one lists the millivolt increments which are the actual digital output of the A/D converter for the corresponding functional millivolt input in column two.

Columns three and four are the actual whole and tenth volt value representations of the millivolt inputs.

Column five values are derived from a formula based on several factors. The 6X portion of the 6X/10 factor represents a scaling factor to generate actual millivolt representations of the true input scale.

The fuel level and charging voltage inputs are voltage divided to one third their actual voltage so the input voltage range is within the A/D converter input range (0-5 volts). Since the output value must represent the true input range of 0-12 volts, the millivolt increment (X) is multiplied by 3. In addition, the X value (millivolt increments) is one twentieth of the actual millivolt range so the X value must be multiplied by 20. The factor 2 was chosen arbitrarily so the ensuing division would be accomplished by a factor of ten rather than 100.

The 6X factor is derived from the fact that 2X3(X) is the same as 6(X).

The resulting value is then rounded in column six, and the actual displayed value is shown in column seven.

Programatically, the actual value manipulation within the subroutines for these functions is as depicted in columns five and six. The value input from the A/D converter is multiplied by six and then divided by ten. This value is passed to the code which performs the round function, and the actual final value to be displayed is as in column seven.

Since the actual voltage value in column six is still a factor of ten too large, the value must be displayed as if it had been divided again by ten.

The decimal point is hardwired between the second and third digit of the charging voltage display (X.X). When the whole number in column seven is sent to the displays, the resulting display of the digits 77, for example, would display as 7.7. Because of this hardware configuration, no further division of the column six value is needed to produce the proper displayed value. The integrity of this display method is maintained throughout the entire range of values, since the prescaling code strips off the most significant digit of column six values of more than two places (100-150), and the remaining two digit value is displayed on display digits two and three.

This voltage range listing also enabled choice of the values for 1/4 and 1/8 tank of fuel reference checks used in the channel three routine. One quarter of the full scale A/D increment value (column one) yields 37.5 or the rounded value 38. The check value used in code is 39. Any value less than 39 equates to less than 1/4 tank of fuel. One eighth of the column one maximum value yields 18.75 or a rounded

19. The check value used in the channel three routine for 1/8 tank of fuel is therefore 20.

A partial listing of the results of the program to compute the range results of the temperature function is shown in figure 22. The full listing for the temperature function can be found in Appendix F.

The base millivolt increment value in column one (159) was found by freezing the sensor in ice to produce the zero degree celsius condition (thirty two degrees Fahrenheit), then inserting it into the working system and reading the actual value returned from the A/D converter.

Since the A/D precision is twenty millivolts, and the LM335 sensor precision is ten millivolts per degree centigrade, for every twenty millivolt increment (column two), the the celsius value returned by the sensor increases by two degrees (column four).

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 3-+5V	DEGREES CELSIUS 0C=3V	FAHRENHEIT DEGREES (((X-159)1.8)+32)	DISPLAY VALUE
159.0	3180	3	0.0	32.00	32
160.0	3200	3	2.0	35.60	36
161.0	3220	3	4.0	39.20	39
162.0	3240	3	6.0	42.80	43
163.0	3260	3	8.0	46.40	46
164.0	3280	3	10.0	50.00	50
165.0	3300	3	12.0	53.60	54
166.0	3320	3	14.0	57.20	57
167.0	3340	3	16.0	60.80	61
168.0	3360	3	18.0	64.40	64
169.0	3380	3	20.0	68.00	68
170.0	3400	3	22.0	71.60	72
171.0	3420	3	24.0	75.20	75

Figure 22

The Fahrenheit value in column five is derived by subtracting the base offset (159) from the actual millivolt value returned from the A/D converter (column one), then multiplying this value by 1.8 and adding 32.

The result of the celsius to Fahrenheit conversion is found in column five which is then sent to the rounding code, which produces the final values which are displayed, and found in column six.

Since the manufacturer claims linearity of the sensor over the entire range, once the base offset was computed it was not necessary to generate this listing for calibration purposes, although it is valuable for observing the behavior of the displayed values over the entire range of possible temperature conditions and does point out system precision variances.

Computer Simulation

Speedometer and Tachometer Functions

Disc Activator Stripe Count and Placement One of the most useful results of computer simulation in this system was applied to the speedometer and tachometer functions. An interactive program was written which allowed varied inputs for interrupt timing and reflective phototransistor activator stripe counts to be entered. This allowed the opportunity to play "what if" with different input combinations to determine the best interrupt time/stripe count correlation for the speedometer interface.

The same program was modified for the tachometer interface and used to find the stripe count and interrupt rate for that function.

Appendix G and H are the listings for the speedometer and

tachometer program results respectively.

Figure 23 is a partial listing of the speedometer results, and figure 24 shows a partial listing of the tachometer results.

By using four disc activator stripes and an interrupt rate of four tenths of a second, it was found that a one to one correspondence could be achieved between speed and the resulting count returned from the 8254 programmable counter.

Fortunately, by using the same interrupt rate as in the speedometer function, it was found that the same one to one correspondence could be achieved by using six activator stripes for the tachometer function.

MPH	PULSES/SEC	PULSES/READ
1	2.51	1.00
2	5.02	2.01
3	7.53	3.01
4	10.04	4.02
5	12.55	5.02
6	15.06	6.03
7	17.58	7.03
8	20.09	8.03
9	22.60	9.04

Figure 23

RPM	PULSES/SEC	PULSES/READ
100	2.50	1.00
200	5.00	2.00
300	7.50	3.00
400	10.00	4.00
500	12.50	5.00
600	15.00	6.00
700	17.50	7.00
800	20.00	8.00
900	22.50	9.00

Figure 24

Functional Simulation For Hardware Debug During development of the turn signal interface, unexpected results occurred with respect to the behavior of the 74LS373 devices during the flash routine. As a full unit of LEDs (8) became lit, instead of the next LED of the next unit lighting as expected, the entire next unit of LEDs was activated. The problem did not occur during the transition between the first four LED activations (1/2 74LS373) and the first LED of the next unit, instead, it occurred only between the transition of the first full unit and the first LED of the next unit.

The problem occurred in the same manner for either the left or right turn signal.

Since the chip selects are all unique, it was thought that the activation code sent to the display board was in some way being corrupted and activating the other units due to some unforeseen logic error in the code segment.

The code segment appeared correct, and no adverse conditions were observed on the oscilloscope, so to verify or disprove the code inconsistency, a program was written to simulate the turn signal flash routine.

The partial listing in figure 25 is a result of the simulation program.

The simulation followed the values of all registers, the data bus values, values used for activation of the 74LS373s (Select bus) and all address locations including branches throughout one complete

ADDRESS	DATA BUS	SELECT BUS	A-REG	B-REG	C-REG	H&L	H&LCONT	D&E	D&ECONT
201C	0	0	0	0	0	2062	0	0	0
2020	0	0	0	0	21	2062	0	0	0
2023	0	0	0	0	21	2062	0	2068	0
2024	0	0	8	0	21	2062	8	2068	0
2025	0	0	8	0	21	2063	4	2068	0
2027	8	0	8	0	21	2063	4	2068	0
2028	8	0	4	0	21	2063	4	2068	0
2029	8	0	4	4	21	2063	4	2068	0
202A	8	0	8	4	21	2063	4	2068	8
202B	8	0	12	4	21	2063	4	2068	8
202C	8	0	12	12	21	2063	4	2068	8
202D	8	0	4	12	21	2063	4	2068	8
202F	8	4	4	12	21	2063	4	2068	8
2031	8	4	0	12	21	2063	4	2068	8
2033	8	0	0	12	21	2063	4	2068	8
CALL THE DELAY ROUTINE FROM LOCATION 2034H, RETURN TO 2037H									
2037	8	0	0	12	20	2063	4	2068	8
2038	8	0	20	12	20	2063	4	2068	8
203D	8	0	20	12	20	2063	4	2068	8
2042	8	0	20	12	20	2063	4	2068	8
2046	8	0	20	12	20	2063	4	2068	8
2048	8	0	12	12	20	2063	4	2068	8
204B	8	0	12	12	20	2063	4	2068	8
2027	12	0	12	12	20	2063	4	2068	8
2028	12	0	6	12	20	2063	4	2068	8
2029	12	0	6	6	20	2063	4	2068	8
202A	12	0	8	6	20	2063	4	2068	8
202B	12	0	14	6	20	2063	4	2068	8
202C	12	0	14	14	20	2063	4	2068	8
202D	12	0	4	14	20	2063	4	2068	8
202F	12	4	4	14	20	2063	4	2068	8
2031	12	4	0	14	20	2063	4	2068	8
2033	12	0	0	14	20	2063	4	2068	8
CALL THE DELAY ROUTINE FROM LOCATION 2034H, RETURN TO 2037H									
2037	12	0	0	14	19	2063	4	2068	8
2038	12	0	19	14	19	2063	4	2068	8
203D	12	0	19	14	19	2063	4	2068	8

Figure 25

execution of the turn signal routine. (Note the address values used for development using the SDK85 design kit)

As a result of using this simulation method, it was determined that the code segment for the turn signal flash routine was operating as expected, and that the problem was hardware related. The problem

was finally identified as a voltage transient not apparent on the oscilloscope.

As a further benefit, the results of the simulation provided an interesting graphic of the internal workings of the microprocessor and system bus values during an actual execution.

System Design Tools

Designer/Programmer Notebook

A programmer notebook was maintained throughout the entire development process as a means of coordinating all needed information.

The notebook contained sections on all functional areas, a section of data sheets on representative and actual devices, and a note section to keep track of the history of observations throughout design.

The notebook was invaluable for coordination purposes upon final system migration/integration.

SDK85 System Design Kit As stated the SDK85 system design kit offers 1/4K RAM for entry of applications programs, utilities for examination and alteration of internal registers and a single step function for debug.

The system was expanded to 3/4K RAM and 9 I/O ports, and use was made of the large wire wrap area for the connectivity interface to the breadboarded hardware functional circuits.

The SDK85 kit offers a matched chip family system that allowed development of code that was directly migratable with only a few modifications to compensate for electronic and address differences in the final system.

All programming of the SDK85 is accomplished by loading hexadecimal opcodes in sequential RAM locations and executing the codes beginning at location 2000H.

The SDK85 kit proved a worthy development tool, although it was somewhat limited in providing easy interface to prototypes which do not make use of the I/O ports as a means of access.

EPROM Programmer An EPROM programmer was constructed for use in burning the code into final chip form. The EPROM programmer allowed entry of the code by toggling the hexadecimal values into a set of eight switches.

There are two onboard groups of 7-segment displays which display the data and address values.

The EPROM programmer makes use of "personality modules" which are interchanged to match the particular type of EPROM being programmed. Additional provisions are made for copying one EPROM to another, and by setting the data switches to FFH and invoking the copy function, an EEPROM can be erased in minutes.

Use of EEPROMS in Development As a development aid, an EEPROM was chosen rather than an EPROM for intermediate code generation.

By using the proper personality module, and taking advantage of the EPROM programmer's ability to access and program any sequential location, it was possible to enter and modify code in real time rather than wait a lengthy period for an EPROM to erase under ultraviolet light if an error was made.

A zero insertion force socket was temporarily wired to the system board, and is also incorporated on the EPROM programmer. By means of

these sockets, it was possible to program the EEPROM, then insert it directly into the system under development and observe the results.

If modifications were needed, it was possible to instantly return the EEPROM to the EPROM programmer socket, make changes, and return the EEPROM to the system for verification.

Final integration was achieved by copying the EEPROM to an EPROM, disconnecting the temporary zero insertion force socket, and inserting the EPROM in the designated permanent system socket.

Miscellaneous Design Tools For hardware development, heavy use was made of an oscilloscope and volt ohm meter for testing system wave forms, frequencies, and voltage levels.

For software development, a program was written to produce programming sheets with sequential locations representing the SDK85 and system memory. The program sheets were invaluable in providing numbered sequential locations which saved time and potential error of hand numbering each time a piece of code was developed.

At the bottom of the code sheets, a note section was incorporated which was used to record observations and particulars to remember during final integration.

The importance of organization cannot be stressed enough when dealing with the integration of multiple interacting hardware and software modules into a complete working system.

Design Process

Schematics The schematics for this project were hand generated and later entered into PCCAPS software, which is a schematic capture package marketed by the Personal CAD Company, better known as PCAD, and was run on the Texas Instruments Business Pro computer.

PCCAPS requires the generation of symbols which represent the particular parts within a system. Symbols were generated for all integrated circuits, discretes, and connectors in this system.

Once the symbols are generated, they are interactively placed, and connections are manually entered. As the connections are entered, they are named and become "nets" which the program tracks as it constructs the connectivity database. Nets may be merged, deleted, renamed, or highlighted as a design aid.

Once the schematic database is constructed, it can be post processed into a database for circuit board layout.

CAD Computer aided design tools were used in the development of the display and main logic circuit boards in this system. The first board (display board) was laid out on the SCICARDS® CAD design system in 1983 when I joined Texas Instruments. The company at that time was involved in evaluation of the SCICARDS system and used my design as a test. I generated the "netlist" (connectivity information) from the hand drawn schematic and T.I. personnel used this information to design the board on the CAD system.

The second board was designed in 1987 by myself using the PCAD software package. The schematic captured using the PCCAPS software was

postprocessed into a database for use by the PCCARDS portion of the PCAD software.

The counterpart of symbols in the PCCAPS software is the PCCARDS "part". All parts used in the system were generated and used by the PCCARDS software for final layout database creation.

Once the parts and resulting database are created, the parts are interactively placed within the coordinates of the board outline which is an actual representation of the size and shape of the circuit board to be designed.

PCCARDS offers several tools which greatly automate the design process and were taken advantage of during the creation of this board. Once the parts were entered within the board outline, the "histogram" function was invoked which graphically and numerically gives feedback to the designer on density values. Using the histogram function, the parts were moved within the board outline to provide the optimum placement for "routing" the connections.

Rather than connecting all the routing channels by hand, the PCCARDS autorouter was used. PCCARDS offers several algorithms for autorouting which will result in various degrees of routing success depending on the needs of the designer. For this board, the slowest most comprehensive algorithm was used which resulted in an autoroute run completion in one and one-half hours, with eight signals left unrouted out of a total of three hundred eighty six. These eight signals (nets) were then entered interactively to complete the design.

The final design tool used was the design rule checker (DRC)

which performs a spacing tolerance check of all features on the board and will also flag shorts.

In this design, the line to line, line to pad, pad to pad, and pad or line to board outline tolerances were checked.

Fabrication Texas Instruments is very supportive of educational achievement. For each board, T.I. allowed the generation of photographic film from the output of the CAD system so that fabrication of the boards could be achieved.

The output of the CAD design is "Gerber" code and consists of numerical control data instructions to drive a photoplotter machine. The Gerber code was sent to the artwork department and plotted, and both negative and positive film of exact tolerance was returned for fabrication.

The fabrication process was performed by myself at home. The chemicals and miscellaneous equipment were obtained through retail mail order.

The fabrication process is a six step process. In order of sequence the steps are: 1) Coat the circuit board with photoresist, 2) Expose the design, 3) Develop the image, 4) Etch the image, 5) Drill the board, and 6) Assemble the parts.

In the first step, the copper clad circuit board is cut to dimension, cleaned with steel wool to remove impurities, and coated with an ultraviolet light sensitive photoresist solution.

In step two, the negative film is placed over the coated board and exposed to ultraviolet light for a period of time. Negative film

is opaque in all places except for the pads, signal connections and outline of the design, which are transparent. As the ultraviolet light shines through the transparent areas of film, it causes a chemical reaction with the photoresist and hardens it in only the transparent areas.

In step three, the image is developed. The exposed copper clad circuit board is immersed for a period of time in a developing solution which rinses off all but the hardened resist from step two.

In step four, the developed board is immersed in an acid solution where all copper is etched away except that which is covered by the hardened resist. At the end of the etch step, all copper is removed from the board except for the image of the actual design.

The design must then be drilled for part insertion and through board signal interconnect (these are double sided boards) and have the parts and through board connections soldered permanently in steps five and six which completes the design.

The third circuit board was not dense enough to use a CAD system, so the board was laid out (designed) by hand. Working from a schematic, the pads and lines were transferred to the copper clad circuit board via "rub on transfers" which are resistant to the acid etch solution. Once the transfer of the design to the circuit board is complete, steps four through six are followed as in the photographic process.

Complete System Testing

Problems Encountered and Solved

Noise Noise can be classified as the most prevalent and bothersome problem encountered when developing this system. Noise appeared in several of the interface circuits and within the system proper and accounted for untold hours of problem solving debug activity.

Speedometer and Tachometer The effects from noise became apparent in the first interface circuitry developed which was the speedometer and tachometer circuitry. It appeared in three distinct areas of the interface and was corrected in three separate ways.

Effects From System Noise The initial development of the speedometer and tachometer circuitry was via breadboarding of the circuit, and interface for data and control signals was achieved through a ribbon cable to the SDK85. Power for the circuit was supplied from a separate power source.

As the activator disks were spun, multiple pulses were entered into the 8254 device rather than the much fewer expected result.

After much investigation, a check with the oscilloscope showed that the signal to the 8254 was extremely noisy. As the threshold of the 5413 schmitt trigger device was approached, the spikes from the noise began to trigger the device multiple times, resulting in erroneous counts being fed to the 8254 counting device.

The source of the noise was tracked to the power supply, and

large electrolytic capacitors were added between power and ground to correct the situation.

The filtering of the power supply was maintained on the final circuit board assembly. On the third board which contains the power input for the system and the rectification for the five volt devices, very large electrolytic capacitors were added as filtering for the power and ground run throughout the entire circuits.

Phototransistor Light Noise The phototransistors used in this system operate within the infrared spectrum. When exposed to incandescent or indirect sun light as from a light bulb or a room with the curtains open, the phototransistors react to these spectrums and will produce erroneous count pulses.

The incandescent light is especially troublesome since the phototransistors react to the 60 Hertz cycling of the bulbs and this 60 Hertz frequency appears as noise on the output signal of the FPA104s.

A simple solution was used to rectify the problem by covering the phototransistors and causing them to operate in full darkness. This must in some manner be transferred to the packaging of the final design when installed on the motorcycle. The phototransistor and activator disc sections must be encased to avoid exposure to sunlight.

Effects On The Switching Transistor The final area of the speedometer and tachometer affected by noise was the input signal to the switching transistor.

Due to a small amount of noise still present on the input signal to the transistor, the saturation level of the base was being prematurely reached and causing the the transistor to turn on and off erroneously.

The saturation voltage of the transistor used in this circuit is .6 volts. When the input to the base of the transistor was low, due to system current draw the low level voltage was higher than zero volts. The noise on the input signal to the base of the transistor had just enough amplitude to reach the .6 volt threshold at the highest point of the noise spikes, causing erroneous transistor activation.

The problem was overcome by placing a 20K resistor in series with the input signal to the transistor base which acted as a current limiter and lowered the input voltage to the base of the transistor accordingly. The noise was still evident on the signal line, but the amplitude of the highest spikes now fall below the .6 volt threshold voltage of the transistor.

74LS373 Transient Noise As previously stated, by the generation of a computer simulation program, a noise problem was discovered involving the chip select signal lines to the 74LS373 octal latch devices.

By running the computer simulation program it was discovered that the problem was noise related and not due to the program code. The simulation was then used to track the contents of the system bus throughout the turn signal flash cycle and showed that the final value

put on the bus prior to erroneous activation of the other LED units was FFH.

As the final rotated value of one full unit (FFH) was put on the data bus to light the last LED in a full unit, the current draw was at the peak for the flash cycle since a hexadecimal FF is the condition of all eight bus lines being a logic 1 or "on".

The noise created by this current draw created the transient effect within the display board circuitry and prematurely activated the other 74LS373 devices.

The problem was rectified by the insertion of .01 microfarad capacitors on the chip select input lines of each 74LS373 device which effectively filtered the transients and corrected the problem.

Effects of A/D Fluctuations On Display Upon final system integration, it was discovered that the 7-segment displays which displayed the results of the A/D conversions did not display a stable number. Instead, they flickered wildly and were in reality displaying varying numbers at a very high rate.

The problem was traced to current draw on the system power supply due to the normal system execution activity. As the system operates, the current demands change according to the particular system activities at the moment. This current draw has the effect of raising and lowering the five volt and ground voltage levels of the system accordingly.

In this system, the A/D converter is referenced to ground, and the upper range input reference is five volts. As the system voltage

levels for ground and power change during normal execution, so do the reference values of the A/D converter. Since the precision of the A/D converter is twenty millivolts, only a small change in the reference voltage levels is needed to change the A/D converted values.

The values being displayed were in actuality these changing A/D values as a result of the changes in the reference value inputs to the A/D converter device.

The flicker occurred due to the speed of execution of the A/D converter and the governing system monitor software. The channels were being converted so quickly, and the system monitor was servicing the polled interrupt mask (set by the converter upon completion of a converted channel) so quickly, that the values were being displayed faster than the human eye could discern.

The solution to the problem was twofold. The clock rate supplied by one half of the 556 timer device to the A/D converter was slowed down considerably to slow the speed of conversion of the analog input value. In addition, a delay routine was added in the program code to slow the service speed of the system monitor.

The reference voltages to the A/D converter still fluctuate, but the conversion and monitor service times are slow enough to create a filtering effect where most of the fluctuations are missed by the A/D converter and the resulting display is much more stable.

The displays still change occasionally in relation to the changing reference voltage levels, but the change is so slow as to not be annoying to the operator.

Bus Timing Synchronization All of the 7-segment displays are activated via the 74LS154 4:16 decoder, which takes it's input from the system address bus. Upon system integration, the 7-segment displays rhythmically pulsed erroneous values and would not display any of the expected values.

The problem was traced to fluctuations caused by changing values on the system address bus during several 8085 internal operations when the contents of the address bus are not guaranteed and are listed as undefined by the manufacturer. The changing values on the address bus during this system activity caused activation of the 74LS154 outputs to the 4511 devices which drive the 7-segment displays.

This was a normal effect in this system and did not create a problem with other system devices since activation of all other devices needed at least two signals; a chip select from the 74LS154 in conjunction with a READ or WRITE signal. The other devices in the system also received the spurious chip select signals, but it had no effect on them in the absence of the needed READ or WRITE signal present only when they were addressed.

The problem was corrected by the addition of a 74S08 positive AND device which gates the READ and WRITE signals together. The original G1 chip select pin of the 74LS154 was connected to the A15 address line which activated the device whenever the A15 line was low. This signal was replaced with the output from the 74S08 gated READ and WRITE signals. The 74LS154 was then effectively synchronized via activation only during a READ or WRITE sequence. Spurious signals to

other system components were also eliminated since the activation signal to the A15 address line was disconnected and the 74LS154 no longer receives signals during the undefined bus content periods.

Need For Artificial Clock Pulse To The 8254 To load a count into the 8254 programmable timer, the CPU writes the count and the value is latched upon receipt. The count, although, is not transferred into the internal counting element until the rising edge of a clock pulse to the particular counter.

In this system, the pulses from the speedometer and tachometer interfaces act as the clock pulses to counter one and two respectively. Until a pulse is received from either function's phototransistors, the initial count written to the counters will "sit at the front door" and not be loaded into the counting element where the actual decrementing of the count is executed.

The existence of a pulse from either of the two interfaces result from either the engine running or the motorcycle moving. Upon initial startup of the system the engine will not be running nor will the motorcycle be moving when the first display of the counter values is executed (upon successful entry of the security code). The count returned from the counters when no clock pulses have been received to enter the count into the counting elements is undefined and results in random numbers being displayed on the speed and tachometer 7-segment displays.

To correct this problem experimentation was made with "motion sensing" by taking a tap off the input lines to the clock pins of the

8254 counters one and two and using flip-flops which would be set if a pulse had been received meaning that the engine was running or the motorcycle was moving. The flip-flops would be reset by the CPU if either condition existed. If a flip-flop was not set, the CPU would ignore the count returned from the "unloaded" counters and instead display zeros on the speed and tachometer displays. This method proved ungainly in software and hardware overhead, so the final solution was to add an artificial pulse via the 8255 programmable interface device to clock the count values into the counters when no pulse from the interfaces existed.

A 74LS86 exclusive OR device was used to gate one input signal from each interface with one artificial clock pulse for each counter from the 8255 together. The output of each of these gatings feed one of the inputs of each counter.

An exclusive OR device was used to guard against the situation that the signals from the artificial clock and the input from the interface may be active at the same time. With an exclusive OR gate, if the input from the interface is high at the same time the artificial clock pulse is high, the output from the gate is low. If either of the of the signals is low while the other is high, the clock signal will be accepted. Once the value is clocked into the 8254 counter via the artificial clock pulse from the 8255, the 8255 signal is brought low to allow normal operation via the interface pulse signal throughout the remainder of the count decrement to interrupt cycle.

SDK85 Kit Limitations The SDK85 is limited in provision of access to it's system bus. All access to breadboard prototypes is achieved via the onboard I/O chips which causes a designer to have to simulate many of the normal microprocessor signals such as the simulated chip select signals used in this system.

There came a need to use the SDK85 and still have access to its system bus for testing of the 74LS154 device and the 8279 keyboard device.

The problem was overcome by wirewrapping custom taps into the system data bus, address, and control signals. Devices were removed from the system and prototype devices were interfaced via wirewrap to strategic pins of the vacated sockets. In this way the programs could be entered into the SDK85 and manipulated at the address of a previously different device.

This scheme effectively "fooled" the system and gave access to all internal signals of the SDK85.

Discrepancies And Shortcomings of the System

System Precision

A/D Conversion Error The computations for the A/D conversion function have a built in round-off error. The A/D converter is an 8-bit device and can represent the scale between the maximum and minimum reference voltages (REF+/REF-) in two hundred fifty six distinct divisions.

Since the A/D converter is referenced between five volts and ground, the range to be expressed in two hundred fifty six divisions is five volts. Five volts divided by two hundred fifty six results in a value of .0195 volts or 19.5 millivolts.

The value per division used in this system is rounded to 20 millivolts for a built in error of .5 millivolts per A/D conversion increment.

Temperature Sensor Precision One LM335 temperature sensor is capable of ten millivolt per degree celsius precision. The A/D converter in this system has a precision of 20 millivolts per increment.

When the two devices are mated with these precision levels, the result is a two degree celsius precision level since the temperature sensor will register a two degree change per one unit change of the A/D converter.

When the celsius value is converted to Fahrenheit and rounded as in this system, the resulting scale is not in one degree increments. As can be seen in the temperature scale listing on page 210, the resulting values can be plus four or minus three degrees in accuracy.

Simulation For Demonstration The system will not be installed at the time of demonstration, necessitating that the functionality be "bench" demonstrated for the thesis defense.

The speed and tachometer interfaces will be demonstrated utilizing the actual cables used for final installation, connected to a variable speed drill to simulate changes in speed and engine RPM.

On board variable resistors will be manipulated to simulate changes in charging voltage and fuel level. A buzzer will be used to simulate the motorcycle horn, and toggle switches will be used to simulate turn signal and kickstand down switch closures.

The temperature sensor will be manipulated using a heat source to simulate changes in engine temperature and the system will be powered by a 12 volt power supply to simulate the 12 volt motorcycle battery.

Although the circuit will be demonstrated using simulated inputs, all circuitry and sensors will be actual and ready for installation with one exception; the fuel sensor. Actual fuel sensors are "float activated" variable resistance devices. Retrofiting one of these will cause alteration of the existing fuel tank, so this sensor will not be chosen until final installation. The circuitry however, will be ready to accept whatever sensor is finally chosen.

Final Installation Considerations

Cooling Requirements Temperature should pose no problem for system operation. All of the integrated circuits in the system are rated seventy degrees celsius (one hundred fifty eight degrees Fahrenheit) or above as the upper temperature operating limit, although I believe that care should be taken not to leave the motorcycle in the direct sunlight as absorbent temperatures could surpass these limits.

A consideration upon final installation would be to make provisions in the enclosure for a small cooling fan.

The enclosure should also incorporate ventilation openings for both passive and active cooling while the motorcycle is at rest or moving.

LED Light Filtering The effects of bright light or direct sunlight greatly reduce the efficiency of light emitting diodes.

Final installation should incorporate a filter which reflects light and allows the underlying LEDs to be visible. These filters are usually red plastic although an amber color would be just as effective.

Another consideration would be to recess the displays at an indirect angle within the enclosure creating a shading effect in conjunction to the filtering.

Light emitting diodes perform best in low or no light, and will pose no problem for night time operation.

Effects of Vibration The effects of vibration in this system

should be minimized as much as possible. The effects on the solder joints of the components and the tightness of the connectors could be major since the automotive environment is notoriously hostile to electronic equipment.

The stock analog gauges which are replaced by this system were mounted on rubber shock absorbing mounts. The replacement system should utilize the same mounting interfaces wherever possible.

Enclosures The enclosure should be constructed to provide the previously mentioned shading, venting, and shock absorbing characteristics.

Consideration should also be given to the enclosure material which will have to withstand the effects of weather and road related contaminants such as grease and oil.

Since motorcycles are not always fair weather transportation vehicles, the enclosure must also be watertight in the event the motorcycle is operated in the rain.

Conclusions

Alternative Approaches for Improved System

Point Closure Sensing for the Tachometer Function A method for interfacing electronic tachometer equipment currently being used is to run an input lead from one side of the coil to the digital system.

This high voltage line is then rectified, filtered, and squared up for use with digital components.

By using such an approach, the mechanical interface of cable and spinning disc as used in this system would be replaced by solid state components with immunity to light noise and provide potentially longer life.

Magnetic Sensing for the Speedometer Function Another possible solution for eliminating the FPA104 phototransistors from the system all together, would be the use of magnets and magnetic pick-ups for the speedometer function.

This system, in use in automotive applications, uses fixed magnets usually attached to the drive shaft and HALL effect switches mounted at close proximity which sense the magnetic field and activate as the drive shaft turns.

This arrangement would provide a solid state interface for the speedometer function, although mounting of the components may be difficult as they would have to be mounted to the wheel in the absence of a drive shaft.

Use of an Isolated Voltage Reference for A/D Several of the A/D precision issues existent in the present system would be resolved by the incorporation of a precision voltage reference.

The most widely used precision voltage references are rated at 2.5 volts. This cuts the voltage range to be expressed by the 8-bit precision of the A/D converter from five volts to two and one half volts and effectively increases the precision of the A/D converter by a factor of twenty.

A two hundred fifty six increment representation of a two and one half volt range now yields a single A/D reference division of .0097 millivolts instead of the .0195 currently in use in this system.

The benefits of incorporation of a precision voltage in the system would be enhanced if the power source to the reference were isolated from the digital system power source. The isolation of power sources would provide immunity to the current fluctuations in the present system which cause the voltage reference range to float and cause fluctuations in the A/D converter outputs accordingly.

By increasing the precision of the A/D converter, the representation of the temperature values returned from the LM335 temperature sensor would also be enhanced. Presently, the A/D converter is only able to represent two degree changes in temperature since the temperature sensor precision is ten millivolts per degree while the precision of the A/D converter is twenty millivolts per increment. By increasing the precision of the A/D converter to .0097,

the A/D converter could represent the changes in temperature in one degree precision.

Updated technology The conception of this system and the beginnings of development began in 1983. In the nearly five year time span from then to the present it is acknowledged that technology has changed to such an extent that alternate devices could be used today throughout the system to achieve the same results.

Although technology advances have been many, I believe that if the system was redesigned today the only change I would make would be in the area of component reduction.

All of the components used in the system are still widely in use today and readily available. It would be possible to use a sixteen or thirty two bit microprocessor, although the benefits from the increased addressing capabilities and speed would be overkill in this system.

As demonstrated, the amount of RAM and ROM, and the speed of the 8085 and peripheral components are more than adequate in the present system.

The 8085 instruction set is also fully capable and techniques such as segmentation are not needed. Benefit would although be derived from today's enhanced microprocessor mathematical capabilities. The software overhead of the system could be greatly reduced by a microprocessor that was capable of multiplication and division.

The greatest benefit from using present updated technology would

be gained by using a single chip microcomputer for component reduction and overall miniaturization.

For example, Motorola offers an extensive microcomputer family of chips, one of which, would be ideally suited for use in this system.

The MC6805R3 8-bit device offers twenty four bi-directional I/O lines configurable to three 8-bit I/O ports, eight input only lines, a four channel A/D converter, three thousand seven hundred seventy six bytes of EPROM, one hundred twelve bytes of RAM, an 8-bit timer and a 7-bit prescaler in addition to a broad instruction set in one forty pin package.

This one component could, with the addition of a few support devices, replace the entire internal bus configuration of this system.

Use of a device of this nature would in effect condense almost the entire second circuit board used in this system to a single chip. Since the third circuit board is not very dense, and the display board could be redesigned for more efficient placement of discrete components, it is conceivable that by altering the dimensions of the board size slightly, and using this microcomputer, that it may be possible to condense the entire three board system to a single dense circuit board.

The circuit board technology used in this system could also be greatly updated if one could afford the expense of commercial fabrication. The technology of the present boards is standard technology, through hole, double sided configuration.

By changing to a multilayer design (multiple internal routing layers) the system would definitely fit on one board.

By changing to state-of-the-art surface mount technology, which places components on both sides of the board and eliminates through holes, the system would not only fit on one board, but the size of the board could actually shrink from the present dimensions!

Design Tradeoffs There are a number of times within the design cycle of a firmware project when tradeoff decisions must be made between parts or parts and software based on the cost and or benefit of the change to the overall system. Two major tradeoff decisions were made in this system and are worthy of note.

Hardware/Hardware The original design of this system called for the use of a 74LS138 3 to 8 binary decoder as the chip select device for all system components. The chip select signals for all devices on the display board (4511 and 74LS373) were to be originated via the 8255 peripheral interface and the simulation of the high to low and low to high signals would be simulated in software.

After feasibility testing with the 4511 devices, it was found that they could be activated via a pulse from a binary decoder in an operating system and the signals could be derived from simple OUT instructions from the CPU.

By using one more unoccupied address line as input to a 74LS154 4 to 16 binary decoder, the functionality of the 74LS138 was maintained, and an additional eight chip selects were obtained.

These additional chip selects were in turn used to drive all of the 4511 devices on the display board and resulted in a decrease in software overhead of the instructions for simulation of the 4511 latch signals.

The software benefit more than outweighed the additional pin count of swapping a sixteen pin device (74LS138) for the larger twenty four pin device (74LS154).

Hardware/Software In the original design, the plan was for the CPU to poll an unintelligent device such as an I/O port connected directly to the keyboard for input from key closures. The CPU would be directly responsible for sensing key closures and debounce of the closures when detected.

This particular scenario is particular costly in CPU cycles and greatly complicates system interaction and integration where there are interrupts and multiple functions being monitored.

One forty pin device (8279) at a monetary cost of two dollars and forty nine cents, was chosen which provided the entire interface to the keyboard and completely off-loaded the CPU from the keyboard polling task.

The result now demands only a periodic check of the 8279 by the CPU for contents of a sensed key closure and the resulting simple input instruction to read the value if a closure has been detected.

As an added benefit, this chip could enable the keyboard to be active throughout full system operation for expanded keyboard

operations (as a future enhancement) at the cost of very few additional CPU cycles and little software overhead.

Suggestions for Future Extensions/Enhancements

LCD Displays The addition of Liquid Crystal Displays (LCDs) would solve the problem of LED brightness during daytime operation.

LCDs absorb light for operation, so the brighter the light source, the easier it is to read the displays. For nighttime operation the method used for illumination is to provide back lighting from a common white light source.

LCDs also operate more slowly at higher temperatures. This would definitely be a consideration before incorporating them for rapidly changing displays such as the speedometer and tachometer functions in high temperature summer conditions.

Trip Computations An easy extension of the existing system would be to provide feedback on trip or mileage computations. The information basics to provide this type of data are already built into the system, since it now manipulates rate, time, and distance values in present operation.

The counter for the speed and tachometer interfaces could be manipulated to provide a time source, and the system knows values for speed and distance. Periodically, the values could be stored and updated in RAM and compared against an input value representing total estimated miles to complete a trip, and provide miles traveled, miles

to go, total elapsed time, time of arrival at present speed and time of arrival at an altered speed.

If compared to the fuel level sensor values, the system could also return average fuel usage per mileage information.

Speech It was originally planned to incorporate a speech function into this system, but the function was dropped to expedite development.

The National Semiconductor Co. manufactures a microprocessor bus compatible speech processor and speech ROM set called DIGITALKER. The complete ROM set provides a two hundred plus word vocabulary which is "spoken" in either a male, female, or child's voice depending on the external frequency setting of the processor clock.

Sentences are developed by successive activations of the speech processor, accessing different words from the vocabulary. Words may be split and or joined to form words not in the vocabulary by interrupting a word in progress with a new word access via custom timing delays.

The DIGITALKER vocabulary is ideally suited to automotive applications and could be easily accessed to give audio feedback of all functions within this system.

Consideration of the speech function must include the inclusion of a speaker into system. In the motorcycle environment, unlike a closed automobile interior, outside noise and the effects of weather would have to be overcome in some way for the use of a speaker to be successful in this application.

A possible solution would be to use a small speaker for feedback while the motorcycle is at rest and the engine is not running. For over the road operation, an earphone jack could be used, and the operator could wear a small earphone within the helmet.

Dual Power Sources for the System and Horn Power to the system and initial reset are achieved by turning on the ignition via the ignition key.

The security function in this system has one drawback in that when the system goes into an unrecoverable halt, the reset can be accomplished by turning off the ignition and then turning it on again.

If the unauthorized user understands this, then the system can be turned off and the security horn blast reset quickly upon unsuccessful entry of the security code.

An enhancement of the system would be to power the horn and system from an alternate source other than the stock ignition system. When the system went into the halt state, and the horn sounded, the only way to reset the system then would be via the hidden reset switch.

Related Thesis Topics for the Future

Implementing Security Algorithms in Firmware The implementation of the security function in this system was one of the most interesting portions of the project to me. The combinations of software security algorithms and hardware resultant reactions are limitless.

Future theses could lend themselves to original methods of encryption, decryption, code recognition, and hardware security system configuration and activation systems.

A/D Conversion of Real Time Events The topic of conversion of non-digital real time events can be further explored. The topics encountered in this system of A/D precision, full scale value representation of the function, functional multiplexing, and timing in A/D conversion systems are worthy of investigation.

Hardware Synchronization and Software Control The activities of a microprocessor bus, the effects of software instructions on the bus, and the synchronization of the bus activities and internal processor functioning are also interesting topical areas. These areas lend themselves to simulation studies, waveform studies, concurrency and theoretical processing systems.

Artwick, Bruce A. Microcomputer Interfacing. Prentice-Hall, 1980.

Bedard, P. "Seeing Red [Instrument Lights]," Car and Driver (April, 1986), pp. 182.

Fairchild. Optoelectronics Handbook. Fairchild, 1980.

Foster, Claxton C. Computer Architecture. Litton Educational Publishing, 1976.

Hayes, John P. Computer Architecture and Organization. McGraw-Hill, 1978.

Hordeski, Michael F. Handbook of Microprocessor Applications. Tab Books, 1979.

INTEL. MCS-8085 Family Users Manual. INTEL, 1979.

INTEL. Microsystem Components Handbook. Vol.I, INTEL, 1986.

INTEL. Microsystem Components Handbook. Vol.II, INTEL, 1986.

Keuchen, John A. Microprocessor Cookbook. Tab Books, 1980.

Lamm, M. "Dashboards: From Dial to Digital," Popular Mechanics (January, 1984), pp. 94-95+.

Lee, A. "Microprocessors in Cars," Computers and Electronics (December, 1984), pp. 72-77+.

Larson, Titus, & Titus. The 8080 Bugbook: Microcomputer Interfacing and Programming. Sams Publications, 1981.

Larson, Titus, & Titus. 8080/8085 Software Design. Sams Publications, 1981.

Larson, Titus, & Titus. 8085A Cookbook. Sams Publications, 1982.

National Semiconductor. CMOS Data Book. National Semiconductor, 1981.

National Semiconductor. Linear Data Book. National Semiconductor, 1980.

Stevens, D. "Of Computers and Hardware [Kawasaki Voyager]," Cycle Magazine (November, 1983), pp. 49+.

Texas Instruments. DC Circuits-Circuit Analysis Methods. Vol.I, Texas Instruments, 1979.

Texas Instruments. DC Circuits-Circuit Analysis Methods. Vol.II, Texas Instruments, 1979.

Texas Instruments. Designing With TTL Integrated Circuits. McGraw-Hill, 1971.

Texas Instruments. The TTL Data Book For Design Engineers. 2nd. Edition, Texas Instruments, 1981.

Texas Instruments. Understanding Digital Electronics. Texas Instruments, 1974.

Texas Instruments. Understanding Microprocessors. Texas Instruments, 1979.

Texas Instruments. Understanding Optronics. Texas Instruments, 1972.

Texas Instruments. Understanding Solid State Electronics. Texas Instruments, 1972.

Titus, Titus, & Larson. Microcomputer-Analog Converter Software and Hardware Interfacing. Sams Publications, 1978.

Tracton, Ken. Display Electronics. Tab books, 1977.

Wirth, Niklaus. Algorithms and Data Structuters=Programs. Prentice-Hall, 1976.

Wobschall, Darold. Circuit Design For Electronic Instrumentation. McGraw-hill, 1979.

Woolley, S. "Car Alarms Are Getting Smarter All The Time," Business Week (October 19, 1987), pp. 150.

Major System Components

8085 Central Processing Unit The INTEL 8085 microprocessor is an 8-bit parallel processor designed as the successor to the 8080A, and was introduced in the mid 1970's.

Major internal functions of the 8085 include multiplexed bus timing, clock generation, system bus control, interrupt priority selection, serial input/output, the addressability of eight 8-bit registers, and a two hundred forty six operation instruction set.

In order to conserve overall chip pin count, the 8085 employs an 8-bit bi-directional 3-state bus (AD0-AD7), which serves as both the data bus and also the lower 8-bits of the address bus. An additional eight bus lines, A8-A15, expand the addressing capability to 16-bits, allowing 64k bytes of memory to be addressed without further decoding circuitry.

Multiplexed Bus Timing The 8085 accesses the system bus via READ and WRITE operations. Each READ and WRITE operation is referred to as a machine cycle. Depending on the type of READ or WRITE instruction being executed, there will be a minimum of three to a maximum of six machine cycles involved. The type of READ or WRITE instruction being executed also dictates when data or address information is present on bus lines AD0-AD7 during a particular machine cycle.

The first machine cycle always consists of a MEMORY READ (OPCODE FETCH). During the OPCODE FETCH, the 16-bit address (contents of the

program counter) first appears on address lines AD0-A15. Coincident with the address appearance, the READ and the ALE (Address Latch Enable) pins go low and high respectively, signifying that a READ operation is taking place. The 8085 then interprets the opcode, and determines how many machine cycles will be needed to complete the instruction.

Demultiplexing of bus lines AD0-AD7 is accomplished by use of the ALE signal and another component known as an address latch. Via the transition of the ALE signal, the AD0-AD7 bits are latched and held during the remainder of the machine cycle and thus the lower 8-bits of the address are preserved during the time that data is present on bus lines AD0-AD7. Address lines A8-A15 are usually valid throughout all machine cycles. The state of the READ or WRITE pins indicate whether a READ or WRITE instruction is being executed. Both the READ and WRITE pins are active low. Figure A1 shows the timing of the multiplexed bus signals during an OPCODE FETCH cycle. Notice address lines A8-A15 are valid through all time periods except T4. The opcode at this time has been received from memory and is being internally interpreted. Depending on the opcode received, the CPU may either enter time periods T5 and T6, or begin T1 of another machine cycle.

The 8085 employs two types of READ and WRITE operations; MEMORY READ and WRITE, and I/O READ and WRITE. The difference between the two types is discernable through the state of the IO/M pin. During I/O READS and WRITES, the IO/M pin is active high, while during MEMORY

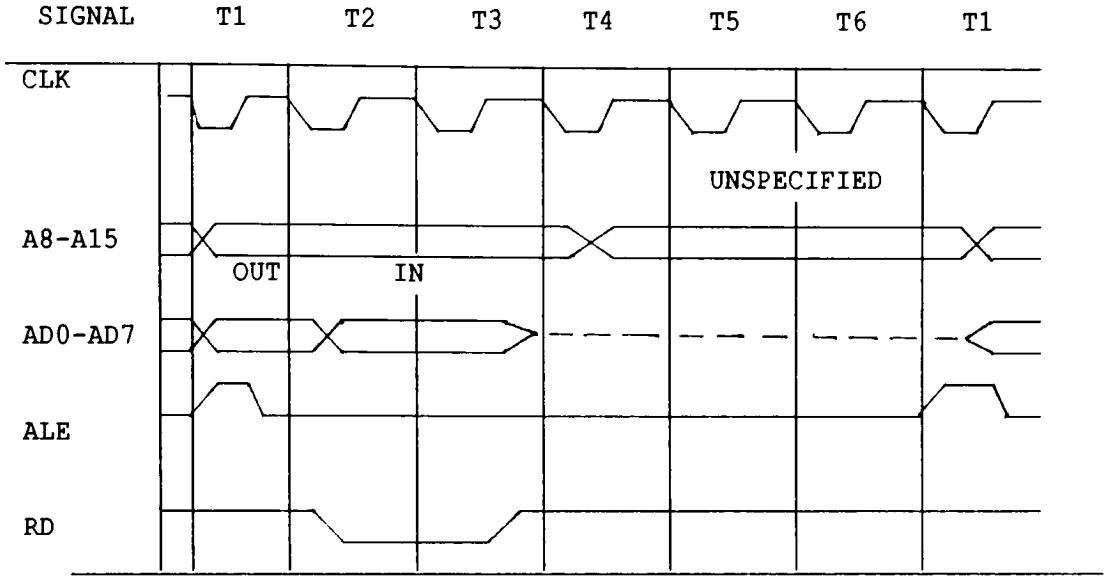


FIGURE A1

READS and WRITES, the IO/M pin is active low. The timing of both types of READS and WRITES is identical, although there is a difference as to what appears on the multiplexed bus between the the two methods. During a MEMORY READ or WRITE, the 16-bit address appears on bus lines AD0-A15 and requires an address latch to preserve the lower order 8-bits of the address. During an I/O READ or WRITE the I/O "port" 1ddress is duplicated on bus lines AD0-AD7, and A8-A15. Since the address is duplicated, there is no need for an address latch if system components are to be addressed as a series of I/O port locations rather than a series of memory addresses. The two methods of address are known as memory mapped I/O and I/O mapped I/O. This system employs memory mapped I/O. Functionality of the memory mapped I/O method will be further discussed in the system architecture section.

Internal Clock Generator The 8085 incorporates clock generator circuitry, and requires an external clock source of 6.25 MHZ or less. The 8085A-2 can accept a 10 MHz input. Clock input is usually derived from a suitable microprocessor crystal. Via the internal clock generator circuitry, the 8085 generates a system clock signal output at its CLK pin. CLK is always half the frequency of the input clock source. This system uses a 6.144 MHz microprocessor crystal, yielding a 3.072 MHz (320 ns) system clock speed appearing at the CLK pin. The instruction cycle speed is 1.3 microseconds.

System Bus Control In addition to AD0-AD7, A8-A15, ALE, RD, WR, and IO/M functions, the 8085 makes bus control provisions for Direct Memory Access (DMA), slow speed peripherals, and system interrupt via the READY, HOLD, HLDA, INTR, INTA, RESET IN and RESET OUT functions.

The READY function is used for interface to slower peripherals, usually memory devices. If the READY pin is low prior to a READ or WRITE, the CPU will generate wait states, until the READY pin turns high. If the READY pin is sampled and found to be high, the signaling device is ready to provide or receive data.

The HOLD and HLDA functions are used for DMA. When a DMA device needs access to the system bus, it will provide a high level signal at the HOLD pin. Upon receipt of a HOLD request, the CPU will relinquish the system bus upon completion of the bus transfer in progress and may continue internal processing. When the CPU receives the HOLD request, the HLDA (HOLD Acknowledge) pin goes high to signal that the HOLD

request has been acknowledged, and the bus will become available.

INTR (Interrupt Request) is used as a general purpose interrupt. The INTA (Interrupt Acknowledge) pin goes low as a response to the receipt of a high level signal being received at the (INTR) pin. Upon acknowledgement of an INTR signal, the program counter is inhibited, and a RESTART or CALL instruction may be placed on the bus to transfer control to a service routine. These functions are usually used in conjunction with an interrupt controller device.

The RESET IN and RESET OUT functions are used as general purpose system resets. A low level signal at the RESET IN pin will stop execution and reset the program counter to 0000. In response to the RESET IN signal, the RESET OUT pin will go high. This pin is usually used for the system reset pulse.

Interrupt Priority Selection The 8085 provides five hardware interrupts which differ in priority and functionality.

The previously mentioned INTR is maskable; it can be enabled by the Enable Interrupt (EI), and disabled by the Disable Interrupt (DI) software instructions. This interrupt causes the CPU to fetch an RST (RESTART) or CALL instruction from the system bus which has been placed there by a system component. There are eight valid RST instructions which cause the CPU to vector the branch to one of eight possible fixed memory locations. If a CALL instruction is encountered, the branch can be made to any location in memory.

The RST 5.5, RST 6.5, and RST 7.5 interrupts are maskable through the use of the SIM (Set Interrupt Mask) software instruction, and also

subject to enable and disable through the EI and DI instructions. Any one, all, or none of the three RST interrupts can be selected (enabled) by setting particular flag bits within the interrupt mask. Figure A2 shows the format of the 5-bit Set Interrupt Mask word.

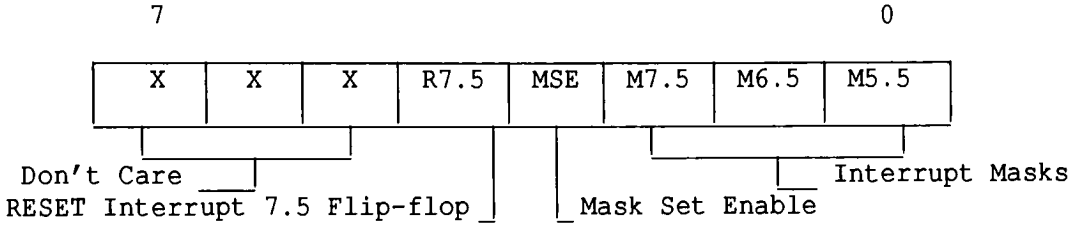


Figure A2

The status of the interrupt mask previously set by the SIM instruction can be read by the use of the RIM (Read Interrupt Mask) instruction. The format of the 6-bit word returned by the Read Interrupt Mask instruction is shown in figure A3.

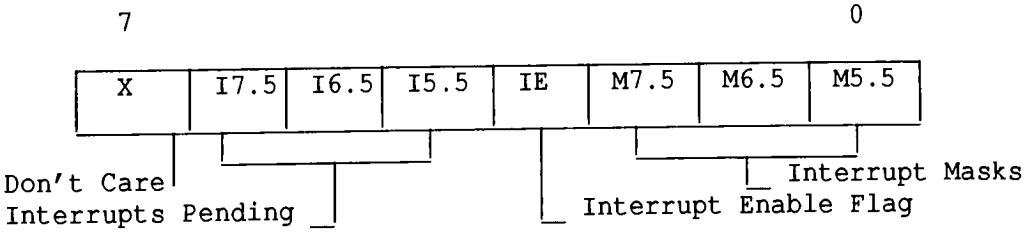


Figure A3

The interrupts pending bits (4-6) allow the recognition of interrupts without actual branching as a response to the receipt of an interrupt. This scheme is used in this system to enable the polling of interrupts and will be discussed in greater detail in the system architecture section.

The last available hardware interrupt is the TRAP. This interrupt is not subject to any masking instruction, and cannot be enabled or disabled by the EI or DI instructions.

The priority of interrupt recognition is as follows: 1) TRAP, 2) RST7.5, 3) RST6.5, 4) RST5.5, 5) INTR.

Serial Input/Output The 8085 provides two pins for serial input/output; Serial Input Data (SID), and Serial Output Data (SOD). Data at the SID pin is read by issuing the RIM instruction. The data at the SID pin is then read into bit 7 of the accumulator for processing. Conversely, the issuance of a SIM instruction latches bit 7 of the accumulator out to the SOD pin for submittal to the serial bus.

Addressability of Eight 8-bit Registers The 8085 has eight 8-bit internal registers available for manipulation or interrogation by the programmer. Six of these registers can be used as individual 8-bit registers or three pairs of 16-bit registers. In addition to the eight registers, the 8085 contains a stack pointer and a flag register. The eight registers, the stack pointer and flag register descriptions and functions are as follows;

The accumulator (the "A" register) is the main working register of the 8085. All arithmetic, logic, load and store, and I/O takes place via the accumulator. The accumulator can be used only as an 8-bit single register.

The program counter is a 16-bit pointer which always points to the next executable instruction in memory. The contents of the program

counter is a 16-bit address and cannot be directly manipulated by software instructions.

The BC, DE, and HL registers are the six registers that can be manipulated as six individual 8-bit registers or as three 16-bit register pairs. The HL register pair is used extensively by the 8085 instruction set as an address pointer for indirect addressing. The BC and DE registers can also be used by a smaller subset of the instruction set for indirect addressing.

The stack pointer is a 16-bit pointer to the top of the stack in system memory. The stack pointer cannot be manipulated other than increasing, decreasing, or changing it's contents to point to a different location in memory. No arithmetic or logic instructions are allowed to be executed on the stack pointer register. The instructions for putting data on the stack cause the stack pointer to decrease, and conversely, the instructions for removing items from the stack cause the pointer to increase. Therefore, the 8085 stack area is usually placed high in physical memory and grows down as items are place on the stack.

The flag register contains 4-bits which reflect the status of the accumulator following arithmetic and logic operations, and 1-bit for parity. The 4-bits are for carry, auxiliary carry (for BCD operations), sign, and a zero flag.

Instruction Set The 8085 is driven by a comprehensive two hundred forty six operation instruction set. The instruction set is divided into five functional groups of operations as follows;

The arithmetic group operations add, subtract, increment, and decrement data which may be contained either in memory or in internal registers.

The logic group is capable of comparisons, rotations, complements, ANDs, ORs, and XORs between memory data or data in registers.

The data transfer group includes the move, exchange, load, and store operations which manipulate data between memory locations or between registers.

The branch group contains operations for subroutine access (call and return), restarts, and conditional or unconditional jumps. All branch instructions operate on memory address locations.

The stack, I/O, and machine control group include operations for maintaining a LIFO stack, reading from input ports, writing to output ports, setting and clearing of flags, and the setting and reading of interrupt masks. The pinout for the 8085 is found in Appendix B.

The 8185 Random Access Memory chip is a 1k static RAM with 8192-bits configured as one thousand twenty four 8-bit words. The 8185 is one of a special family of integrated INTEL chips that interfaces directly to the 8085 multiplexed bus without the need for address acquisition via an address latch. This is accomplished by using two special pins for the chip enable functions called CE1 and CE2. The status of CE1 and CE2 as well as the address information on pins AD0-AD7, A8 and A9 are latched at the falling edge of the ALE signal generated by the 8085. If the status of both pins is active (CE1 = low and CE2 = high), the 8185 powers itself up and readies itself for data

transfer. No data transfers can take place until the CS and either a RD (read) or WR (write) pin is activated. This system uses the 8085 IO/M signal connected to the CE1 pin of the 8185, which keeps the 8185 powered down except during memory access cycles for minimum power consumption. The pinout of the 8185 can be found in Appendix B.

The 2716 Electronically Programmable Read Only Memory (EPROM) is a static, 2k, 16,384-bit memory chip configured as two thousand forty eight 8-bit words. The 2716 used in this system has a memory access time of 350 nanoseconds. The 2716 is not a part of the INTEL family of chips that can be directly interfaced to the 8085 multiplexed bus, and therefore must take it's address information from an address latch. The chip operates from a single +5 volt supply, and is programmed by applying a 25 volt, 50 millisecond pulse to pin 21 (VPP) while holding pin 20 (OE) at a high level. The 2716 has a low power standby mode which is enabled whenever a high level signal is present at pin 18 (CE). Since this pin is the chip enable pin, the 2716, by default, will be in the standby mode whenever it is not being directly addressed. The pinout for the 2716 is listed in Appendix B.

The 8255 Programmable Peripheral Interface is a general purpose programmable input/output device which contains 24 I/O pins which may be individually programmed as input or output pins, and grouped in a large number of configurations for ease of system application.

The 8255 has three programmable modes; mode 0, mode 1, and mode 2. Mode 0 allows the 24 pins to be grouped in sets of four, as either input or output. This is the general purpose mode, and is used by this

system. Modes 1 and 2 are most often used for communications applications allowing 8-bit bidirectional bus capability with pins specially designated for handshaking and control.

The 8255 is programmed by writing a special mode control word to it's command register prior to use. The control word sets up the configuration of the ports, establishes how the pins will be used (input/output) within the configurations, and establishes the mode in which the ports will be used.

Addressing of the 8255 is accomplished via normal access read, write, and chip select signals from the CPU in conjunction with two special pins; A0 and A1. When A0 and A1 are high, and the CPU writes to the 8255, the command register will be addressed. When A0 and A1 are low, PORT A is enabled for either a read or write. Similarly, PORT B is accessed when A0 is high and A1 is low, and PORT C is accessed when A0 is low and A1 is high. A read of the command register (A0 and A1 high) is illegal.

Figure A4 shows the control word and resulting configuration that is used in this system by programming the 8255 with the control word = 88H. This chip also uses the address latch as it is not capable of directly interfacing to the 8085 multiplexed bus. The pinout of the 8255 can be found in Appendix B.

The 8279 Programmable Keyboard/Display Interface is a general purpose programmable interface which greatly reduces software overhead and offers a wide variety of capabilities and operating modes. The keyboard section can scan a 64-contact keyboard or be interfaced to an array of sensors. Keyboard entries are debounced and

stored in an on-chip 8-character FIFO queue, and can be decoded using the n-key rollover or 2-key lockout methods. Key entries set the interrupt output, and there is an overflow flag if the FIFO receives more than 8-characters before a read takes place. The display section of the chip can be interfaced to numeric, alphanumeric, or discrete LEDs. The 8279 contains a 16x8 display RAM which can be configured into dual 16x4, and offers right entry, left entry, and calculator display formats.

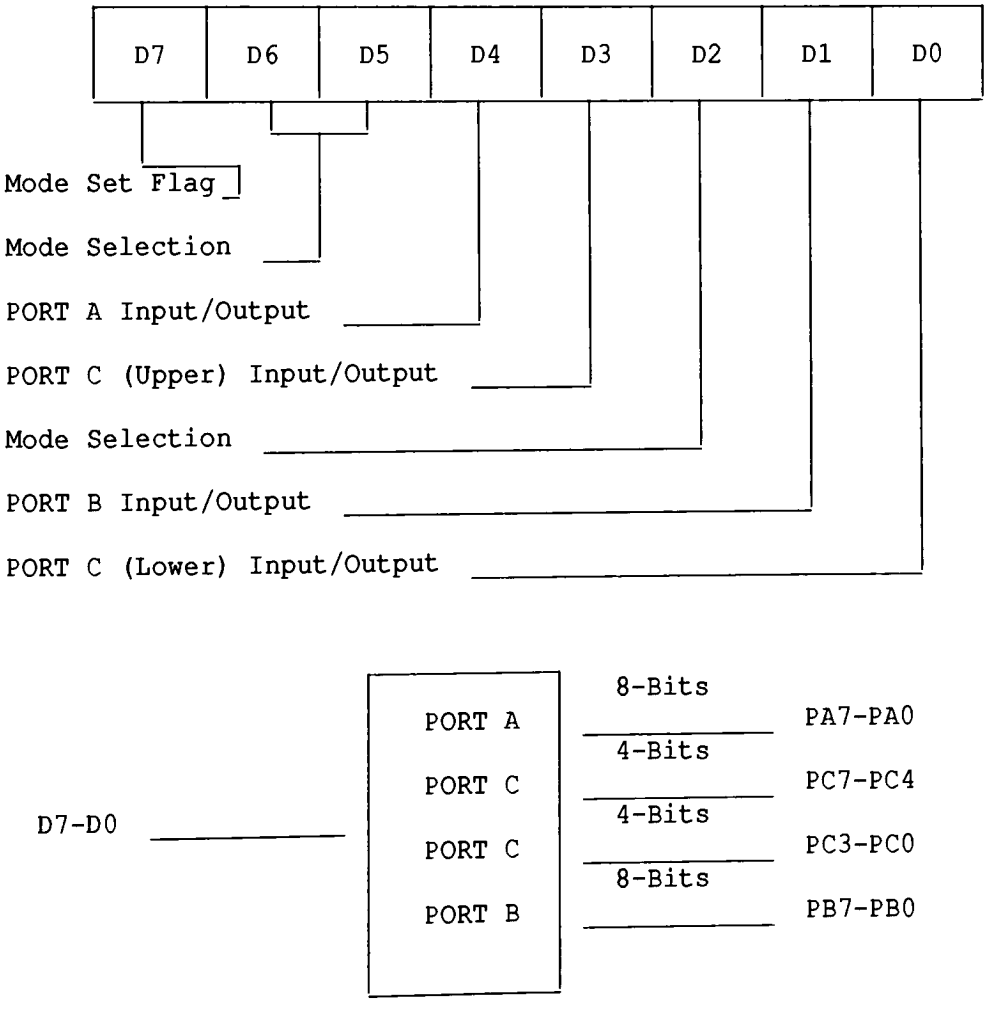


Figure A4

Like the 8255, the 8279 is set up by writing an 8-bit control word to it's command register prior to use, and is not 8085 multiplexed bus compatible. The 8279 needs only one pin to differentiate whether the command register or the other sections of the chip are being addressed. The A0 pin is used for this purpose. A high signal at the A0 pin indicates that the command register is being addressed, while a low level signal indicates that other sections of the chip are being addressed.

Debounce and key matrix scan times are programmable via an 8-bit control word. The 8279 contains a prescaler which divides the external clock frequency by the divisor contained in the control word. INTEL recommends that the correct divisor be used which will yield a 100 kHz internal frequency. This system uses a 3 MHz clock, so the divisor used in the control word is 30 to yield the proper 100 kHz. This frequency in turn is used by the 8279 to yield a 5.1 millisecond keyboard scan time and a 10.3 millisecond debounce time.

This system utilizes the 2-key lockout method of scan logic for deciding which key(s) has been depressed. In the 2-key lockout method, when the first key closure is detected, the debounce logic is set, and other key closures are looked for during the next two scan cycles. If no other closures are detected, it was a single key closure, and the key is entered into the FIFO RAM. If more than one closure is detected within the next two scan cycles, no entries will be made into the FIFO. If all keys are released before the first, then the first key

will be entered into the FIFO. If the first key is released before the others, then the last key found to be depressed will be entered into the FIFO and the first key will be ignored. No key will be entered into the FIFO until it remains depressed alone.

The n-key rollover method allows multiple key closures to be entered into RAM, as each key closure is treated independently of others. The keys are entered into the FIFO RAM in the order in which they are found.

The 2-key lockout method was deemed more reliable since multiple closures are ignored, and the key held the longest (if multiple closures occur) has the highest probability of being the key that was intended to be pressed.

Further discussion on the use of the 8279 control words and interrupts can be found in the interface specification section. The pinout of the 8279 is found in Appendix B.

The 8212 8-Bit Input/Output Port is an 8-bit parallel data register and buffer, and used in this system as the address latch for those chips that are not directly compatible with the 8085 multiplexed bus. The 8212 in this system is wired so the chip will always be active. The eight data-in pins are connected directly to the AD0-AD7 bus lines of the 8085. As discussed in the 8085 section, the data-out values of the 8212 are driven by the 8085's ALE signal. During the time period that the address appears on the 8085 AD0-AD7 pins, the 8085 also pulses the ALE pin high to low. It is this pulse which

latches the data into the 8212 and preserves that lower 8-bits of the address. This latched value will remain as the output of the 8212 until another ALE pulse is received from the 8085. This chip has high bus driving capability and is also often used as a buffer in systems where the bus driving limits of the microprocessor are being neared. The pinout of the 8212 can be found in Appendix C.

The 8254 Programmable Interval Timer is a counter/timer device used for timing control applications in microprocessor systems. The 8254 is used in this system in the speed and tachometer interfaces to keep real time and offload the CPU from timing delays and the associated software overhead. The 8254 provides three independent 16-bit counters (each of which is capable of handling input clock rates of up to 10 MHz), six programmable modes, and either BCD or binary counting.

Like the 8255, the 8254 uses A0 and A1 pins for differentiating between the four possible addressable portions of the chip. In the 8254 they are; counters one through three, and the command register.

The 8254 uses a control word which sets up the operation of the counters prior to use called the mode control word, and is written to the 8254 command register in the same manner as the 8255 and 8279. The mode control word defines which counters will be used, in what mode, and also defines the type of counting to be used; BCD or binary. The mode control word format is shown in figure A5.

SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC1 SC0 (Select Counter)

0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	READ BACK Command

BCD

0	Binary Counter 16-Bits
1	BCD Counter: 4 Decades

RW1 RW0 (READ/WRITE)

0	0	COUNTER LATCH Command
0	1	READ/WRITE LSB Only
1	0	READ/WRITE MSB Only
1	1	READ/WRITE LSB First Then MSB

M2 M1 M0 (MODE)

0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Figure A5

The 8254 allows three different methods of reading a counter while a count is is progress; a simple read, a COUNTER LATCH command followed by a read, and the issuance of a READ BACK command followed by a read.

The simple read returns the value of the counter provided that the clock input to the counter has been inhibited. If the clock signal is still active at the time of a read, the value may be invalid due to a count transition from the clock during the read. This method also requires additional circuitry/control signals to inhibit the clock.

The COUNTER LATCH command latches the count into a buffer at the moment the command is received, and the counter continues uninhibited. This value is then read by the CPU via a normal read operation.

The third method, issuance of a READ BACK command, is much like the COUNTER LATCH command but more powerful, and is the method used in this system. This command effectively takes the place of multiple COUNTER LATCH commands, by allowing the specification of from one to three counters to be latched simultaneously, in one operation. In addition, the READ BACK command can be issued to return the status of the OUT pin, the programmed mode, and the Null Count flag of the selected counter(s). At the moment an interrupt is received from the speed and tachometer circuitry, the READ BACK command is issued which captures the count values of both the speed and tachometer counters thus returning more accurate real time values, and conserving software overhead.

Programming (writing to) the 8254 is straight forward. The control word must be issued first, which sets the configuration and mode of the counter, then the initial count may be written to the counter according to the parameters specified in the control word. This dual operation must be repeated for each counter, although the sequence of the control word write/counter load is flexible. For example, all control words may be written to the 8254 in succession followed by the initial count loads, rather than a control word write then an initial load for each counter.

The five possible modes of operation in the 8254 are as follows: 1) Interrupt on terminal count, 2) Hardware Retriggerable One Shot, 3) Rate generator, 4) Square Wave Mode, 5) Software Triggered Strobe.

Mode 1 is typically used for event counting, and is the mode used in this system. In mode 1, an initial count is loaded in the 8254, and decrements according to the rate of the clock input, in this case, the pulses from 1/2 of the 556 dual timer IC circuitry. When the counter has decremented to zero, the OUT pin (initially low) switches to high and serves as the interrupt to the 8085.

The pinout of the 8254 is listed in Appendix C.

The 74LS154 4 to 16 Binary Decoder is a TTL decoder and demultiplexer, and used in this system as the address decoder. It accepts four binary input bits, and outputs a low signal on one of 16 possible output pins according to the value of the binary input received. These outputs are wired to the Chip Select (CS) pins of the various system components and enable (select) the specific devices during CPU reads and writes.

The pinout of the 74LS154 is found in Appendix C.

The ADC0808 8-Bit 8 Channel Analog to Digital Converter accepts analog signals from external sensors, and converts the signal to digital values capable of being read by the CPU. This device is used for the temperature, charging voltage, and fuel level interfaces of this system. The pinout for the ADC0808 is found in Appendix C.

The ADC0808 offers eight channels, each of which is an input for

an analog signal, eliminating the need for external multiplexing circuitry.

The ADC0808 is a CMOS device with TTL compatible outputs, and employs an 8-bit parallel output latch allowing direct interface to microprocessor bus systems.

Three input (address) pins allow direct addressing of the eight channels. Conversion is initiated by writing a 3-bit value specifying which channel is to be converted, followed by a signal at the ALE (Address Latch Enable) pin to latch the address, and lastly a signal at the START pin to begin the conversion. Using a 640 kHz clock, conversion is complete in 100 microseconds. Upon completion of the conversion, the End Of Conversion (EOC) pin (initially low) switches to a high state and can be used as an interrupt signal to the CPU.

The conversion value is calculated using an internal comparator implemented by use of a successive approximation algorithm. The successive approximation technique tests the unknown value against the weight of each of the eight binary bits beginning with the most significant bit, and testing down to the least significant bit. In an 8-bit converter like the ADC0808, there will be at most eight tests to arrive at the binary equivalent of the unknown value.

The 4511 BCD to 7-segment Latch/Decoder Driver is a CMOS 7-segment LED driver used in this system for driving the speed, tachometer, battery charging voltage, and temperature displays.

The 4511 has four input pins (A-D) which accept a 4-bit Binary Coded Decimal (BCD) value. The decoder converts this BCD value and

outputs to any or all of the seven output pins (a-g) which correspond to the individual segments of a 7-segment LED.

A latch/decoder driver was chosen over a decoder/driver to relieve the burden of display from the CPU. The value is latched into the 4511 by the CPU, and will continue to retain the value displayed until the CPU latches in a new value. The truth table for the 4511 and the segment labels for a 7-segment LED are shown in figure A6. The pinout for the 4511 is found in Appendix D.

D	C	B	A		a	b	c	d	e	f	g	Displayed Value
0	0	0	0		1	1	1	1	1	1	0	0
0	0	0	1		0	1	1	0	0	0	0	1
0	0	1	0		1	1	0	1	1	0	1	2
0	0	1	1		1	1	1	1	0	0	1	3
0	1	0	0		0	1	1	0	0	1	1	4
0	1	0	1		1	0	1	1	0	1	1	5
0	1	1	0		0	0	1	1	1	1	1	6
0	1	1	1		1	1	1	0	0	0	0	7
1	0	0	0		1	1	1	1	1	1	1	8
1	0	0	1		1	1	1	0	0	1	1	9

* For a-g values; 1 = segment lit

7-Segment LED
(Segment Labels)

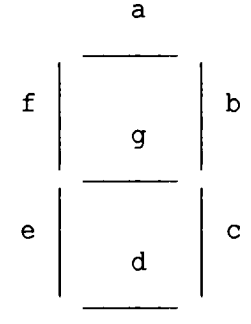


Figure A6

The 74LS373 Octal Latch is used in this system to drive the discrete LEDs of the turn signals and the warning lights.

The 74LS373 is arranged internally as eight D-type flip-flops and activated by pins 11 (latch enable) and 1 (output control). Provided the output control is held low, if the latch enable pin is high, the outputs will follow the inputs. When the latch enable pin is brought

low, the outputs will be latched at the level of the data that was set up at the time of transition.

In this system, the output control is hardwired low so the chips are always active. The data and latch enable signals for the 74LS373 are provided by the 8255 Programmable Peripheral Interface device.

Like the 4511 BCD to 7-segment Latch/Decoder Drivers, the 74LS373 chips were chosen for their latching ability thus freeing the CPU from costly display driving cycles. The 74LS373 chips also drive (source) the LEDs like the 4511's. A high output from the 74LS373 will therefore light the corresponding LED.

The pinout for the 74LS373 can be found in Appendix D.

The LM335 Precision Temperature Sensor is a three terminal integrated circuit operating as a two terminal Zener diode, with one terminal for calibration. The LM335 has an operating voltage in this circuit of 12 volts. Resolution with one device is +10 millivolts per degree Kelvin and operates over the -10 to +100 degrees centigrade temperature range.

The LM335 was chosen for it's near linear output. When calibrated, the device has less than one degree centigrade error over the entire temperature range. The pinout for the LM335 can be found in Appendix D.

The FPA104 Infrared Phototransistor is utilized in this circuit as the sensor for the speedometer and tachometer interfaces.

The FPA104 consists of a Gallium Arsenide infrared-emitting diode, and an NPN phototransistor packaged together in one device. The diode and phototransistor are arranged atop each other and perpendicular to the face of the device, therefore, activation must result from a reflective surface being placed in front of the face of the sensor. Figure A7 shows the packaging of the FPA104. The pinout of the FPA104 can be found in Appendix D.

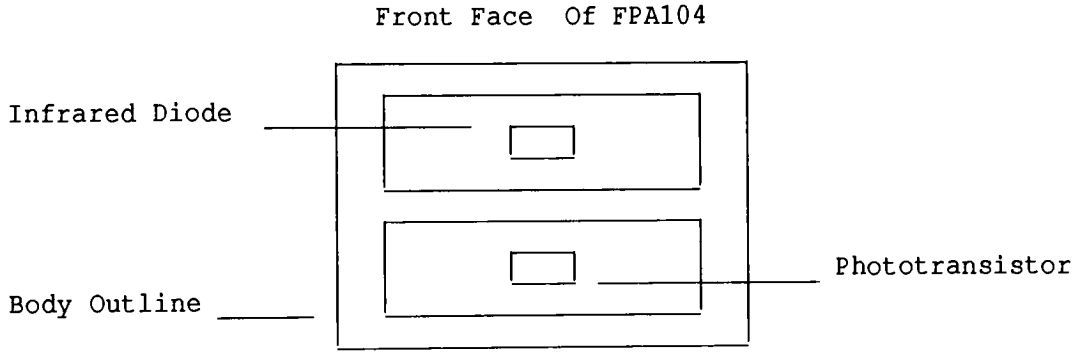
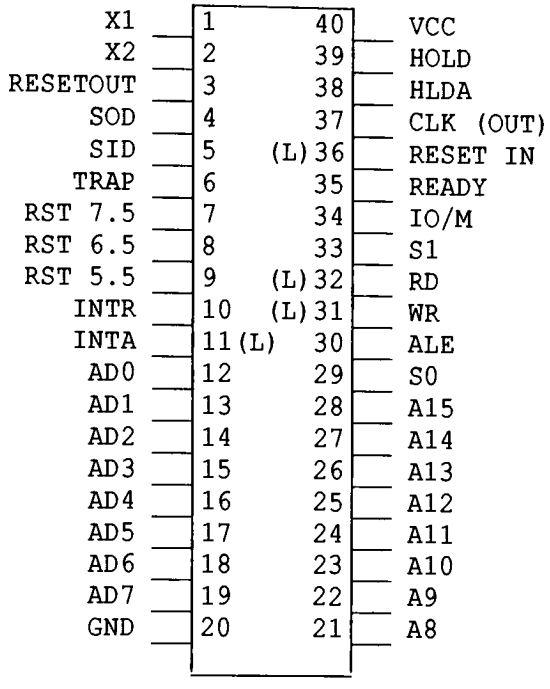


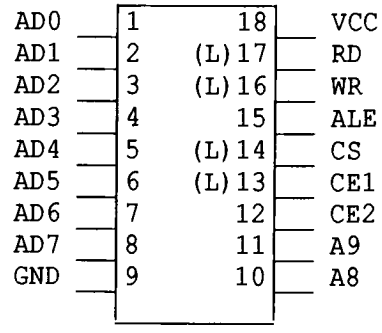
Figure A7

8085 Microprocessor

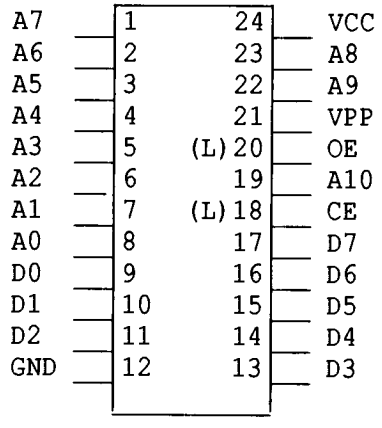


(L) Denotes Active Low

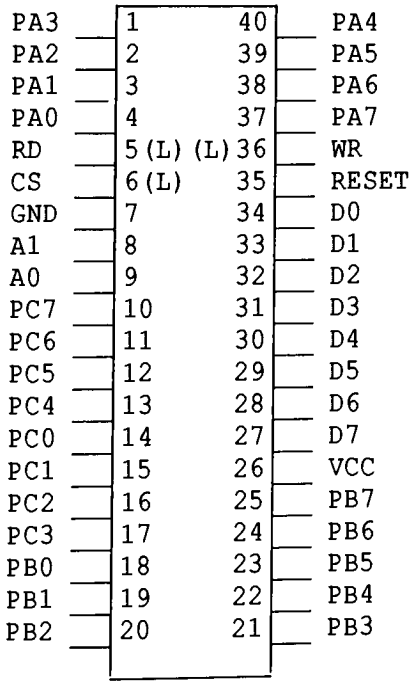
8185 1K Static RAM



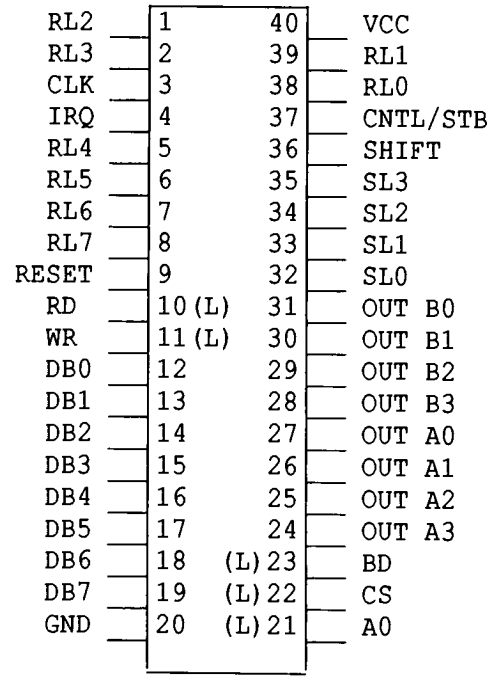
2716 2K EPROM



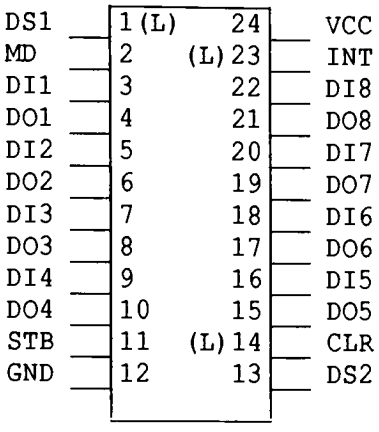
8255 Programmable
Peripheral Interface



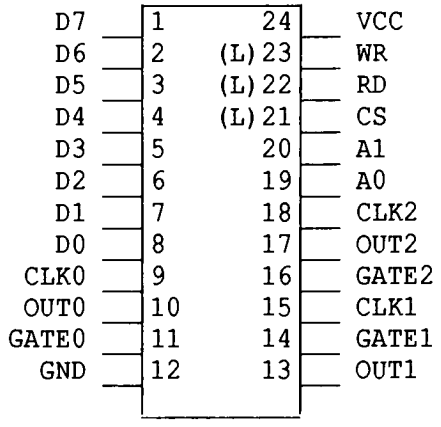
8279 Programmable
Keyboard Interface



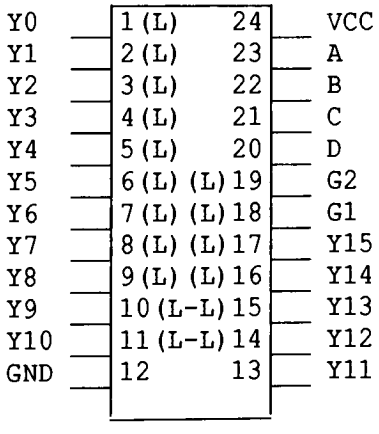
8212 8-Bit
Input/Output Port



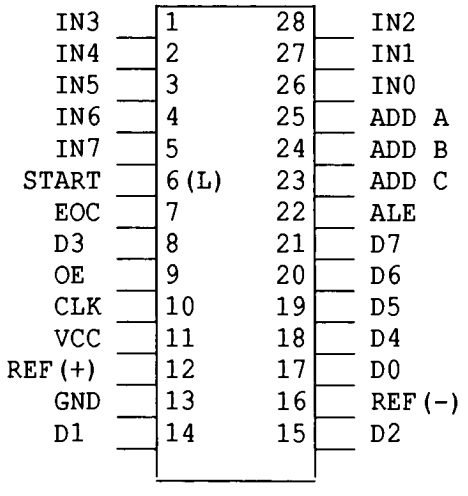
8254 Programmable
Timer



74LS154 4:16
Decoder/Demultiplexer

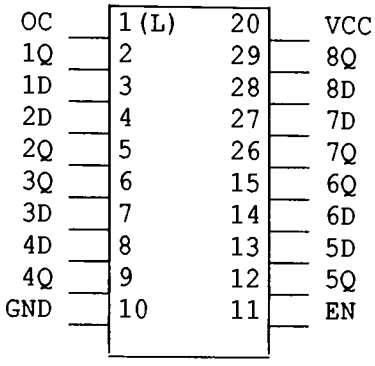


ADC0808 8-Bit 8-Channel
A/D Converter

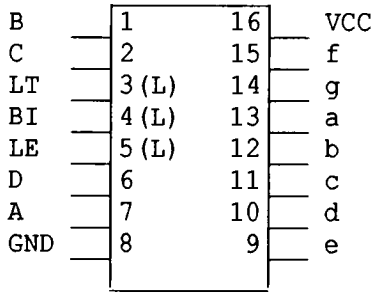


(L) Denotes Active Low

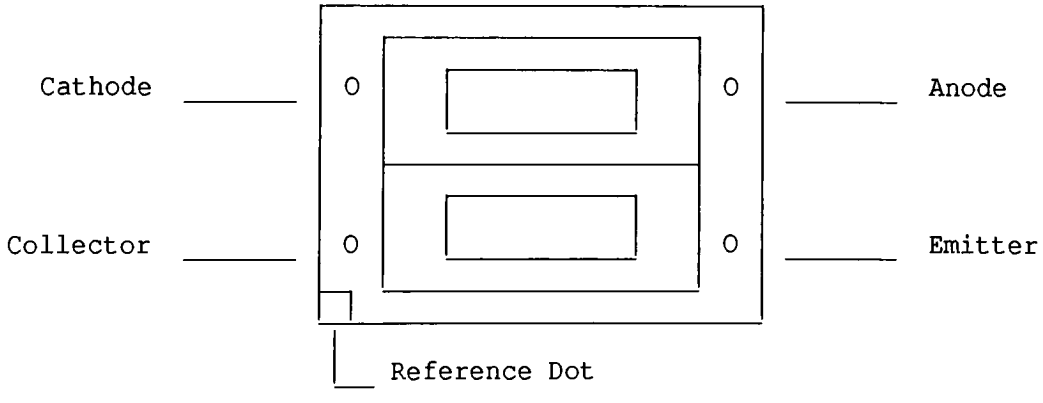
74LS373
Octal Latch



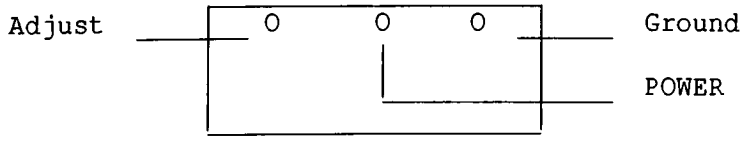
4511 Binary to BCD
Latch/Decoder Driver



FPA104 Infrared
Phototransistor



LM335 Precision
Temperature Sensor



(TO92 Pagkage)

(L) Denotes Active Low

Charging Voltage Calibration/Test Listing

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
1	20	0	.00	0.6	1	0.1
2	40	0	.00	1.2	1	0.1
3	60	0	.00	1.8	2	0.2
4	80	0	.00	2.4	2	0.2
5	100	0	.10	3.0	3	0.3
6	120	0	.10	3.6	4	0.4
7	140	0	.10	4.2	4	0.4
8	160	0	.10	4.8	5	0.5
9	180	0	.10	5.4	5	0.5
10	200	0	.20	6.0	6	0.6
11	220	0	.20	6.6	7	0.7
12	240	0	.20	7.2	7	0.7
13	260	0	.20	7.8	8	0.8
14	280	0	.20	8.4	8	0.8
15	300	0	.30	9.0	9	0.9
16	320	0	.30	9.6	10	1.0
17	340	0	.30	10.2	10	1.0
18	360	0	.30	10.8	11	1.1
19	380	0	.30	11.4	11	1.1
20	400	0	.40	12.0	12	1.2
21	420	0	.40	12.6	13	1.3
22	440	0	.40	13.2	13	1.3
23	460	0	.40	13.8	14	1.4
24	480	0	.40	14.4	14	1.4
25	500	0	.50	15.0	15	1.5
26	520	0	.50	15.6	16	1.6
27	540	0	.50	16.2	16	1.6
28	560	0	.50	16.8	17	1.7
29	580	0	.50	17.4	17	1.7
30	600	0	.60	18.0	18	1.8
31	620	0	.60	18.6	19	1.9
32	640	0	.60	19.2	19	1.9
33	660	0	.60	19.8	20	2.0
34	680	0	.60	20.4	20	2.0
35	700	0	.70	21.0	21	2.1
36	720	0	.70	21.6	22	2.2
37	740	0	.70	22.2	22	2.2
38	760	0	.70	22.8	23	2.3
39	780	0	.70	23.4	23	2.3
40	800	0	.80	24.0	24	2.4
41	820	0	.80	24.6	25	2.5
42	840	0	.80	25.2	25	2.5
43	860	0	.80	25.8	26	2.6
44	880	0	.80	26.4	26	2.6
45	900	0	.90	27.0	27	2.7

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
46	920	0	.90	27.6	28	2.8
47	940	0	.90	28.2	28	2.8
48	960	0	.90	28.8	29	2.9
49	980	0	.90	29.4	29	2.9
50	1000	1	.00	30.0	30	3.0
51	1020	1	.00	30.6	31	3.1
52	1040	1	.00	31.2	31	3.1
53	1060	1	.00	31.8	32	3.2
54	1080	1	.00	32.4	32	3.2
55	1100	1	.10	33.0	33	3.3
56	1120	1	.10	33.6	34	3.4
57	1140	1	.10	34.2	34	3.4
58	1160	1	.10	34.8	35	3.5
59	1180	1	.10	35.4	35	3.5
60	1200	1	.20	36.0	36	3.6
61	1220	1	.20	36.6	37	3.7
62	1240	1	.20	37.2	37	3.7
63	1260	1	.20	37.8	38	3.8
64	1280	1	.20	38.4	38	3.8
65	1300	1	.30	39.0	39	3.9
66	1320	1	.30	39.6	40	4.0
67	1340	1	.30	40.2	40	4.0
68	1360	1	.30	40.8	41	4.1
69	1380	1	.30	41.4	41	4.1
70	1400	1	.40	42.0	42	4.2
71	1420	1	.40	42.6	43	4.3
72	1440	1	.40	43.2	43	4.3
73	1460	1	.40	43.8	44	4.4
74	1480	1	.40	44.4	44	4.4
75	1500	1	.50	45.0	45	4.5
76	1520	1	.50	45.6	46	4.6
77	1540	1	.50	46.2	46	4.6
78	1560	1	.50	46.8	47	4.7
79	1580	1	.50	47.4	47	4.7
80	1600	1	.60	48.0	48	4.8
81	1620	1	.60	48.6	49	4.9
82	1640	1	.60	49.2	49	4.9
83	1660	1	.60	49.8	50	5.0
84	1680	1	.60	50.4	50	5.0
85	1700	1	.70	51.0	51	5.1
86	1720	1	.70	51.6	52	5.2
87	1740	1	.70	52.2	52	5.2
88	1760	1	.70	52.8	53	5.3
89	1780	1	.70	53.4	53	5.3
90	1800	1	.80	54.0	54	5.4
91	1820	1	.80	54.6	55	5.5

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
92	1840	1	.80	55.2	55	5.5
93	1860	1	.80	55.8	56	5.6
94	1880	1	.80	56.4	56	5.6
95	1900	1	.90	57.0	57	5.7
96	1920	1	.90	57.6	58	5.8
97	1940	1	.90	58.2	58	5.8
98	1960	1	.90	58.8	59	5.9
99	1980	1	.90	59.4	59	5.9
100	2000	2	.00	60.0	60	6.0
101	2020	2	.00	60.6	61	6.1
102	2040	2	.00	61.2	61	6.1
103	2060	2	.00	61.8	62	6.2
104	2080	2	.00	62.4	62	6.2
105	2100	2	.10	63.0	63	6.3
106	2120	2	.10	63.6	64	6.4
107	2140	2	.10	64.2	64	6.4
108	2160	2	.10	64.8	65	6.5
109	2180	2	.10	65.4	65	6.5
110	2200	2	.20	66.0	66	6.6
111	2220	2	.20	66.6	67	6.7
112	2240	2	.20	67.2	67	6.7
113	2260	2	.20	67.8	68	6.8
114	2280	2	.20	68.4	68	6.8
115	2300	2	.30	69.0	69	6.9
116	2320	2	.30	69.6	70	7.0
117	2340	2	.30	70.2	70	7.0
118	2360	2	.30	70.8	71	7.1
119	2380	2	.30	71.4	71	7.1
120	2400	2	.40	72.0	72	7.2
121	2420	2	.40	72.6	73	7.3
122	2440	2	.40	73.2	73	7.3
123	2460	2	.40	73.8	74	7.4
124	2480	2	.40	74.4	74	7.4
125	2500	2	.50	75.0	75	7.5
126	2520	2	.50	75.6	76	7.6
127	2540	2	.50	76.2	76	7.6
128	2560	2	.50	76.8	77	7.7
129	2580	2	.50	77.4	77	7.7
130	2600	2	.60	78.0	78	7.8
131	2620	2	.60	78.6	79	7.9
132	2640	2	.60	79.2	79	7.9
133	2660	2	.60	79.8	80	8.0
134	2680	2	.60	80.4	80	8.0
135	2700	2	.70	81.0	81	8.1
136	2720	2	.70	81.6	82	8.2
137	2740	2	.70	82.2	82	8.2

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
138	2760	2	.70	82.8	83	8.3
139	2780	2	.70	83.4	83	8.3
140	2800	2	.80	84.0	84	8.4
141	2820	2	.80	84.6	85	8.5
142	2840	2	.80	85.2	85	8.5
143	2860	2	.80	85.8	86	8.6
144	2880	2	.80	86.4	86	8.6
145	2900	2	.90	87.0	87	8.7
146	2920	2	.90	87.6	88	8.8
147	2940	2	.90	88.2	88	8.8
148	2960	2	.90	88.8	89	8.9
149	2980	2	.90	89.4	89	8.9
150	3000	3	.00	90.0	90	9.0
151	3020	3	.00	90.6	91	9.1
152	3040	3	.00	91.2	91	9.1
153	3060	3	.00	91.8	92	9.2
154	3080	3	.00	92.4	92	9.2
155	3100	3	.10	93.0	93	9.3
156	3120	3	.10	93.6	94	9.4
157	3140	3	.10	94.2	94	9.4
158	3160	3	.10	94.8	95	9.5
159	3180	3	.10	95.4	95	9.5
160	3200	3	.20	96.0	96	9.6
161	3220	3	.20	96.6	97	9.7
162	3240	3	.20	97.2	97	9.7
163	3260	3	.20	97.8	98	9.8
164	3280	3	.20	98.4	98	9.8
165	3300	3	.30	99.0	99	9.9
166	3320	3	.30	99.6	100	10.0
167	3340	3	.30	100.2	100	10.0
168	3360	3	.30	100.8	101	10.1
169	3380	3	.30	101.4	101	10.1
170	3400	3	.40	102.0	102	10.2
171	3420	3	.40	102.6	103	10.3
172	3440	3	.40	103.2	103	10.3
173	3460	3	.40	103.8	104	10.4
174	3480	3	.40	104.4	104	10.4
175	3500	3	.50	105.0	105	10.5
176	3520	3	.50	105.6	106	10.6
177	3540	3	.50	106.2	106	10.6
178	3560	3	.50	106.8	107	10.7
179	3580	3	.50	107.4	107	10.7
180	3600	3	.60	108.0	108	10.8
181	3620	3	.60	108.6	109	10.9
182	3640	3	.60	109.2	109	10.9
183	3660	3	.60	109.8	110	11.0

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
184	3680	3	.60	110.4	110	11.0
185	3700	3	.70	111.0	111	11.1
186	3720	3	.70	111.6	112	11.2
187	3740	3	.70	112.2	112	11.2
188	3760	3	.70	112.8	113	11.3
189	3780	3	.70	113.4	113	11.3
190	3800	3	.80	114.0	114	11.4
191	3820	3	.80	114.6	115	11.5
192	3840	3	.80	115.2	115	11.5
193	3860	3	.80	115.8	116	11.6
194	3880	3	.80	116.4	116	11.6
195	3900	3	.90	117.0	117	11.7
196	3920	3	.90	117.6	118	11.8
197	3940	3	.90	118.2	118	11.8
198	3960	3	.90	118.8	119	11.9
199	3980	3	.90	119.4	119	11.9
200	4000	4	.00	120.0	120	12.0
201	4020	4	.00	120.6	121	12.1
202	4040	4	.00	121.2	121	12.1
203	4060	4	.00	121.8	122	12.2
204	4080	4	.00	122.4	122	12.2
205	4100	4	.10	123.0	123	12.3
206	4120	4	.10	123.6	124	12.4
207	4140	4	.10	124.2	124	12.4
208	4160	4	.10	124.8	125	12.5
209	4180	4	.10	125.4	125	12.5
210	4200	4	.20	126.0	126	12.6
211	4220	4	.20	126.6	127	12.7
212	4240	4	.20	127.2	127	12.7
213	4260	4	.20	127.8	128	12.8
214	4280	4	.20	128.4	128	12.8
215	4300	4	.30	129.0	129	12.9
216	4320	4	.30	129.6	130	13.0
217	4340	4	.30	130.2	130	13.0
218	4360	4	.30	130.8	131	13.1
219	4380	4	.30	131.4	131	13.1
220	4400	4	.40	132.0	132	13.2
221	4420	4	.40	132.6	133	13.3
222	4440	4	.40	133.2	133	13.3
223	4460	4	.40	133.8	134	13.4
224	4480	4	.40	134.4	134	13.4
225	4500	4	.50	135.0	135	13.5
226	4520	4	.50	135.6	136	13.6
227	4540	4	.50	136.2	136	13.6
228	4560	4	.50	136.8	137	13.7
229	4580	4	.50	137.4	137	13.7

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 0-+5V	TENTH VOLTS	ADJUSTED VALUE (6X/10)	ROUNDED VALUE	DISPLAY VALUE (ROUND/10)
230	4600	4	.60	138.0	138	13.8
231	4620	4	.60	138.6	139	13.9
232	4640	4	.60	139.2	139	13.9
233	4660	4	.60	139.8	140	14.0
234	4680	4	.60	140.4	140	14.0
235	4700	4	.70	141.0	141	14.1
236	4720	4	.70	141.6	142	14.2
237	4740	4	.70	142.2	142	14.2
238	4760	4	.70	142.8	143	14.3
239	4780	4	.70	143.4	143	14.3
240	4800	4	.80	144.0	144	14.4
241	4820	4	.80	144.6	145	14.5
242	4840	4	.80	145.2	145	14.5
243	4860	4	.80	145.8	146	14.6
244	4880	4	.80	146.4	146	14.6
245	4900	4	.90	147.0	147	14.7
246	4920	4	.90	147.6	148	14.8
247	4940	4	.90	148.2	148	14.8
248	4960	4	.90	148.8	149	14.9
249	4980	4	.90	149.4	149	14.9
250	5000	5	.00	150.0	150	15.0

Temperature Calibration/Testing Listing

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 3-+5V	DEGREES CELSIUS 0C=3V	FARENHEIT DEGREES (((X-159)1.8)+32)	DISPLAY VALUE
159.0	3180	3	0.0	32.00	32
160.0	3200	3	2.0	35.60	36
161.0	3220	3	4.0	39.20	39
162.0	3240	3	6.0	42.80	43
163.0	3260	3	8.0	46.40	46
164.0	3280	3	10.0	50.00	50
165.0	3300	3	12.0	53.60	54
166.0	3320	3	14.0	57.20	57
167.0	3340	3	16.0	60.80	61
168.0	3360	3	18.0	64.40	64
169.0	3380	3	20.0	68.00	68
170.0	3400	3	22.0	71.60	72
171.0	3420	3	24.0	75.20	75
172.0	3440	3	26.0	78.80	79
173.0	3460	3	28.0	82.40	82
174.0	3480	3	30.0	86.00	86
175.0	3500	3	32.0	89.60	90
176.0	3520	3	34.0	93.20	93
177.0	3540	3	36.0	96.80	97
178.0	3560	3	38.0	100.40	100
179.0	3580	3	40.0	104.00	104
180.0	3600	3	42.0	107.60	108
181.0	3620	3	44.0	111.20	111
182.0	3640	3	46.0	114.80	115
183.0	3660	3	48.0	118.40	118
184.0	3680	3	50.0	122.00	122
185.0	3700	3	52.0	125.60	126
186.0	3720	3	54.0	129.20	129
187.0	3740	3	56.0	132.80	133
188.0	3760	3	58.0	136.40	136
189.0	3780	3	60.0	140.00	140
190.0	3800	3	62.0	143.60	144
191.0	3820	3	64.0	147.20	147
192.0	3840	3	66.0	150.80	151
193.0	3860	3	68.0	154.40	154
194.0	3880	3	70.0	158.00	158
195.0	3900	3	72.0	161.60	162
196.0	3920	3	74.0	165.20	165
197.0	3940	3	76.0	168.80	169
198.0	3960	3	78.0	172.40	172
199.0	3980	3	80.0	176.00	176
200.0	4000	4	82.0	179.60	180
201.0	4020	4	84.0	183.20	183
202.0	4040	4	86.0	186.80	187
203.0	4060	4	88.0	190.40	190

MILLIVOLT INCREMENTS (X)	ACTUAL MILLIVOLTS (20X)	VOLTAGE RANGE 3-+5V	DEGREES CELSIUS 0C=3V	FARENHEIT DEGREES (((X-159)1.8)+32)	DISPLAY VALUE
204.0	4080	4	90.0	194.00	194
205.0	4100	4	92.0	197.60	198
206.0	4120	4	94.0	201.20	201
207.0	4140	4	96.0	204.80	205
208.0	4160	4	98.0	208.40	208
209.0	4180	4	100.0	212.00	212
210.0	4200	4	102.0	215.60	216
211.0	4220	4	104.0	219.20	219
212.0	4240	4	106.0	222.80	223
213.0	4260	4	108.0	226.40	226
214.0	4280	4	110.0	230.00	230
215.0	4300	4	112.0	233.60	234
216.0	4320	4	114.0	237.20	237
217.0	4340	4	116.0	240.80	241
218.0	4360	4	118.0	244.40	244
219.0	4380	4	120.0	248.00	248
220.0	4400	4	122.0	251.60	252
221.0	4420	4	124.0	255.20	255
222.0	4440	4	126.0	258.80	259
223.0	4460	4	128.0	262.40	262
224.0	4480	4	130.0	266.00	266
225.0	4500	4	132.0	269.60	270
226.0	4520	4	134.0	273.20	273
227.0	4540	4	136.0	276.80	277
228.0	4560	4	138.0	280.40	280
229.0	4580	4	140.0	284.00	284
230.0	4600	4	142.0	287.60	288
231.0	4620	4	144.0	291.20	291
232.0	4640	4	146.0	294.80	295
233.0	4660	4	148.0	298.40	298
234.0	4680	4	150.0	302.00	302
235.0	4700	4	152.0	305.60	306
236.0	4720	4	154.0	309.20	309
237.0	4740	4	156.0	312.80	313
238.0	4760	4	158.0	316.40	316
239.0	4780	4	160.0	320.00	320
240.0	4800	4	162.0	323.60	324
241.0	4820	4	164.0	327.20	327
242.0	4840	4	166.0	330.80	331
243.0	4860	4	168.0	334.40	334
244.0	4880	4	170.0	338.00	338
245.0	4900	4	172.0	341.60	342
246.0	4920	4	174.0	345.20	345
247.0	4940	4	176.0	348.80	349
248.0	4960	4	178.0	352.40	352
249.0	4980	4	180.0	356.00	356
250.0	5000	5	182.0	359.60	360

Speed Calibration/Test Listing

MPH	PULSES/SEC	PULSES/READ
1	2.51	1.00
2	5.02	2.01
3	7.53	3.01
4	10.04	4.02
5	12.55	5.02
6	15.06	6.03
7	17.58	7.03
8	20.09	8.03
9	22.60	9.04
10	25.11	10.04
11	27.62	11.05
12	30.13	12.05
13	32.64	13.06
14	35.15	14.06
15	37.66	15.06
16	40.17	16.07
17	42.68	17.07
18	45.19	18.08
19	47.70	19.08
20	50.22	20.09
21	52.73	21.09
22	55.24	22.09
23	57.75	23.10
24	60.26	24.10
25	62.77	25.11
26	65.28	26.11
27	67.79	27.12
28	70.30	28.12
29	72.81	29.12
30	75.32	30.13
31	77.83	31.13
32	80.34	32.14
33	82.85	33.14
34	85.37	34.15
35	87.88	35.15
36	90.39	36.15
37	92.90	37.16
38	95.41	38.16
39	97.92	39.17
40	100.43	40.17
41	102.94	41.18
42	105.45	42.18
43	107.96	43.18
44	110.47	44.19
45	112.98	45.19
46	115.49	46.20
47	118.01	47.20
48	120.52	48.21

MPH	PULSES/SEC	PULSES/READ
49	123.03	49.21
50	125.54	50.22
51	128.05	51.22
52	130.56	52.22
53	133.07	53.23
54	135.58	54.23
55	138.09	55.24
56	140.60	56.24
57	143.11	57.25
58	145.62	58.25
59	148.13	59.25
60	150.65	60.26
61	153.16	61.26
62	155.67	62.27
63	158.18	63.27
64	160.69	64.28
65	163.20	65.28
66	165.71	66.28
67	168.22	67.29
68	170.73	68.29
69	173.24	69.30
70	175.75	70.30
71	178.26	71.31
72	180.77	72.31
73	183.29	73.31
74	185.80	74.32
75	188.31	75.32
76	190.82	76.33
77	193.33	77.33
78	195.84	78.34
79	198.35	79.34
80	200.86	80.34
81	203.37	81.35
82	205.88	82.35
83	208.39	83.36
84	210.90	84.36
85	213.41	85.37
86	215.92	86.37
87	218.44	87.37
88	220.95	88.38
89	223.46	89.38
90	225.97	90.39
91	228.48	91.39
92	230.99	92.40
93	233.50	93.40
94	236.01	94.40
95	238.52	95.41
96	241.03	96.41
97	243.54	97.42

MPH	PULSES/SEC	PULSES/READ
98	246.05	98.42
99	248.56	99.43
100	251.08	100.43
101	253.59	101.43
102	256.10	102.44
103	258.61	103.44
104	261.12	104.45
105	263.63	105.45
106	266.14	106.46
107	268.65	107.46
108	271.16	108.46
109	273.67	109.47
110	276.18	110.47
111	278.69	111.48
112	281.20	112.48
113	283.72	113.49
114	286.23	114.49
115	288.74	115.49
116	291.25	116.50
117	293.76	117.50
118	296.27	118.51
119	298.78	119.51
120	301.29	120.52

RPM Calibration/Test Listing

RPM	PULSES/SEC	PULSES/READ
100	2.50	1.00
200	5.00	2.00
300	7.50	3.00
400	10.00	4.00
500	12.50	5.00
600	15.00	6.00
700	17.50	7.00
800	20.00	8.00
900	22.50	9.00
1000	25.00	10.00
1100	27.50	11.00
1200	30.00	12.00
1300	32.50	13.00
1400	35.00	14.00
1500	37.50	15.00
1600	40.00	16.00
1700	42.50	17.00
1800	45.00	18.00
1900	47.50	19.00
2000	50.00	20.00
2100	52.50	21.00
2200	55.00	22.00
2300	57.50	23.00
2400	60.00	24.00
2500	62.50	25.00
2600	65.00	26.00
2700	67.50	27.00
2800	70.00	28.00
2900	72.50	29.00
3000	75.00	30.00
3100	77.50	31.00
3200	80.00	32.00
3300	82.50	33.00
3400	85.00	34.00
3500	87.50	35.00
3600	90.00	36.00
3700	92.50	37.00
3800	95.00	38.00
3900	97.50	39.00
4000	100.00	40.00
4100	102.50	41.00
4200	105.00	42.00
4300	107.50	43.00
4400	110.00	44.00
4500	112.50	45.00
4600	115.00	46.00
4700	117.50	47.00

RPM	PULSES/SEC	PULSES/READ
4800	120.00	48.00
4900	122.50	49.00
5000	125.00	50.00
5100	127.50	51.00
5200	130.00	52.00
5300	132.50	53.00
5400	135.00	54.00
5500	137.50	55.00
5600	140.00	56.00
5700	142.50	57.00
5800	145.00	58.00
5900	147.50	59.00
6000	150.00	60.00
6100	152.50	61.00
6200	155.00	62.00
6300	157.50	63.00
6400	160.00	64.00
6500	162.50	65.00
6600	165.00	66.00
6700	167.50	67.00
6800	170.00	68.00
6900	172.50	69.00
7000	175.00	70.00
7100	177.50	71.00
7200	180.00	72.00
7300	182.50	73.00
7400	185.00	74.00
7500	187.50	75.00
7600	190.00	76.00
7700	192.50	77.00
7800	195.00	78.00
7900	197.50	79.00
8000	200.00	80.00
8100	202.50	81.00
8200	205.00	82.00
8300	207.50	83.00
8400	210.00	84.00
8500	212.50	85.00
8600	215.00	86.00
8700	217.50	87.00
8800	220.00	88.00
8900	222.50	89.00
9000	225.00	90.00
9100	227.50	91.00
9200	230.00	92.00
9300	232.50	93.00
9400	235.00	94.00
9500	237.50	95.00

RPM	PULSES/SEC	PULSES/READ
9600	240.00	96.00
9700	242.50	97.00
9800	245.00	98.00
9900	247.50	99.00
10000	250.00	100.00
10100	252.50	101.00
10200	255.00	102.00
10300	257.50	103.00
10400	260.00	104.00
10500	262.50	105.00
10600	265.00	106.00
10700	267.50	107.00
10800	270.00	108.00
10900	272.50	109.00
11000	275.00	110.00