GCT GENERAL COMPUTER
TECHNOLOGY, INC.

P.O. Box 3009
Boulder, Colorado 80307
(303) 233-6800

REAL TIME CLOCK/ INTERRUPT CONTROLLER

REFERENCE MANUAL

GCT – 1979
900-0014-0

First Printing: June, 1979

The information contained herein is believed to be accurate and reliable. However, General Computer Technology, Inc. does not assume any responsibility for any injury or damages incurred in the use of this product.

900-0014-0

TABLE OF CONTENTS

Section Page

APPENDIX CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

PLEASE READ THIS DOCUMENTATION COMPLETELY AT LEAST ONCE BEFORE BEGINNING CONSTRUCTION OR ASSEMBLY!

The Interrupt Controller/Real Time Clock Board is a multi-function board designed to interface the CPU to time-dependent events. It provides a battery-operated calendar-type clock, an eight level fully maskable priority interrupt cntroller, and an interval timer/counter/generator facility for timing events, measuring frequency and generating time delays.

1.1 FEATURES

Interface: Uses 16 contiguous output and input ports; full address decoding on board. All address and data lines present 1 TTL Load. Uses 16 wire ribbon cable to CPU interrupt socket.

Clock: 1 MHz crystal timebase, stable to plus or minus .002% from -30 degrees centigrade to + 70 degrees centigrade. Calibration tolerance of plus or minus .002% at 25 degrees centigrade. Settable to plus or minus .0001% of nominal frequency. Rechargeable nickel-cadmium batteries with built in charger provide over 14 days operation per charge.

Decade frequencies available from 1 MHz to 1 second. Counter holds 2^24 seconds, readable through four input ports, six bits per port. A switch to allow setting the clock may be added.

Interrupt Controller: 8214-based 8 prioritized levels. Supports all 3 Z-80 modes and 8080 interrupt mode. Each level selectable for rising edge, falling edge, active high, or active low signals. Schmidt-triggered inputs. Each level individually maskable under software control.

Interval Timer: 8253-based timer supporting programmable one-shot rate generator, triggered strobe and counter modes. Three independent 16-bit counters (binary or BCD) with six programmable counter modes and fully software selectable inputs from eight sources for all timer clocks and gates. Two decade prescaler with 50 MHz input for frequency counter applications.

1.2 SPECIFICATIONS

Uses one I/O slot. Interfaces to Digital Group Z-80 or 8080 CPU boards.

Timebase:	1.000000 MHz crystal oscillator.
Timebase Stability:	(long term) 30 seconds per year at 25 degrees centigrade.
Timebase Stability:	(short term) + or -.002% from -30 degrees centigrade to +70 degrees centigrade.
Interrupt Inputs:	TTL compatible, 1 load.
Frequency Counter Input:	1 TTL load, 50 MHz maximun frequency.
External Counter and Gate Inputs:	TTL, 2 MHz maximum.
Power Requirements:	+5 volts at 1.2 amps. +12 volts at 40 milliamps.
Optional Power Requirements:	+4.8 volts at 0.7 to 1.3 milliamps (Ni-Cad supply included). +9 to +12 volts at 40 milliamps. (for external charger) +6.3 to +14 volts at 1.5 milliamps. (for alternate battery supply)

CHAPTER 2

CIRCUIT DESCRIPTION

The Interrupt Controller/Real Time Clock board can be broken down into three major functions. The functions are the real time clock, the interval timer with frequency counter, and the interrupt controller. The board also contains address decoders and an internal data bus to provide the system interface.

2.1 REAL TIME CLOCK

The real time clock is basically a countdown chain which divides the 1 Megahertz crystal controlled oscillator to provide timing signals down to 1 second and also a counter with capacity to store over a half year's worth of seconds. All the circuitry in this section is CMOS to allow battery operation when the computer is off or a power outage occurs.

The oscillator is of the parallel resonant type and its frequency may be pulled about + or - 30 Hz. The crystal is calibrated to within 20 Hz and has an extremely low frequency shift for temperature fluctuations about room temperature.

The oscillator output is buffered and fed through three dual-decade counters (IC's 30, 31, and 35) which provide all the decade outputs down to 1 second. Each decade output is buffered by a 4009 to reduce loading and to isolate the CMOS supply voltage from the system supply. The decade outputs are available through jumpers and a data selector (IC 15) to provide periodic interrupts or timing signals for use in other sections of the board.

The one-second signal goes to two 12-bit binary counters (IC's 9 and 32). These counters can be read by putting their output on the read bus through one of the four hex tri-state buffers (IC's 17 through 20). The counters are set by clocking each counter at a high rate (as selected by the decode selector, IC 15). A quad bi-lateral gate (IC 33) feeds the slow or fast clocking signal as controlled by the software.

Individual reset signals are also availale to help set the counter. All of these signals pass through IC 8 and thus can be inhibited by a CMOS latch (IC 7). Once the latch is in the inhibit position, it can only be activated by the time-settng switch, thus preventing the time-keeping functions from being affected by power transients, run-away software, or programming errors.

2.2 INTERVAL TIMER/FREQUENCY COUNTER

A programmable interval timer chip (8253, IC 54) forms the heart of the board's timing, counting and frequency generation applications. The chip is organized as three totally independent 16 bit counters. Each counter may work in one of six possible modes of operation. Each counter may be written or read to the internal data bus, which interfaces to the I/O bus of the computer.

The counters each have a clock input, a gate input and an output. The input and gate signals are selected by the six selectors (IC 37 through 42), which can select one of eight possible signals. One of the eight signals is fed from off the board so that each counter is accessable by external hardware. Some of the other signals are decade-divided signals from the real time clock, programmable signals and output signals from other timers. The latter feature allows timers to be cascaded, or allows one timer to trigger another.

One of the counters also has a prescaler (IC's 51 and 25) for use as a frequency counter. A selector (IC 49) controls the amount of prescaling necessary and the prescaling dividers are automatically gated and reset between signal samples.

2.3 INTERRUPT CONTROLLER

The interrupt controller is based on the 8214 (IC 5) priority interrupt controller chip. Whenever the chip detects an interrupt on a level higher than the internally stored level, it causes an interrupt. To allow flexibility, each level is maskable and edge or level triggerable due to the input flip flops. Also, the interrupt inputs can be inverted and slow wave forms are de-glitched by the schmidt triggers.

When an interrupt occurs, the 8214 generates an interrupt signal for one clock pulse. This signal is latched up by IC 24, which sends the interrupt to the CPU and activates the tri-state interrupt-level outputs so that they may be sent to the CPU. The CPU acknowledges the interrupt, but is not finished with the interrupt-level data until the acknowledge signal is removed. This removal resets the interrupt request. A separate software-controlled line can also be used to reset a pending interrupt that may occur during system power-on.

The interrupt level socket transfers the level to the CPU in the proper format by proper jumpering. Since the 8080 CPU board has eight interrupt inputs, a 7442 may be inserted in this socket to decode the interrupting level into the proper interrupt line.

2.4 UTILITY FUNCTIONS

The board performs its own port decoding for a block of 16 ports. Read and write strobes are generated by IC's 47 and 14, respectively. Data to the board is buffered by IC 46 which drives all of the internal configuration latches. This data must also be buffered by IC 45, since the timer chip's data lines are bi-directional and cannot drive the load on the previously described data lines. The low drive CMOS counter lines also appear on this bus. These outputs are buffered and sent to the CPU through IC 44.

CHAPTER 3

ASSEMBLY

3.1 BOARD CONSTRUCTION (UNASSEMBLED VERSION)

Estimated Construction Time: 3-8 hours

To build the Interrupt/Real Time Clock Board, you will need the following tools and equipment:

Fine tipped low wattage soldering iron (approximately 15 watts)
Solder - 60/40 resin wire solder, 24 gauge (approximately)
DO NOT USE ACID CORE SOLDER!
Diagonal cutters - small micro shear type preferred
Long-nosed pliers
Flux remover or alcohol
Small brush
Volt-ohm meter

Refer to the parts placement diagram (Appendix L) during construction.

Before beginning to mount and solder components, inspect the board. The side from which the components are mounted has the manufacturer's label along the left edge of the board. Compare the areas where IC sockets will be inserted with the layout to see that there are no shorts occurring between either the traces leaving the IC or the IC pads or holes in which the IC's are mounted. While plating errors like this are a rare occurrence, once the IC sockets are inserted it is very difficult to find such a problem.

The sockets should be mounted as close to the board as possible. Do not bend the leads of the IC sockets excessively before soldering, as they may break off at the base of the socket.

1. Make sure you have all of the components shown on the parts list.

2. Insert and solder the two 24-pin IC sockets for IC5 and IC54, making sure the notched end of the socket is near the top of the board, away from the edge connector.

3. Insert and solder the four 20-pin IC sockets. CAUTION: THE SOCKET TRACES ARE VERY CLOSE AND A CAREFUL JOB MAY SAVE YOU HOURS OF TROUBLESHOOTING TIME!

4. Insert and solder the thirty-three 16-pin IC sockets.

5. Insert and solder the twenty-nine 14-pin IC sockets.

6. Insert and solder the 22K 8-pin resistor pack (Z3). Note the dot near pin 1 on one end. This end must be oriented toward the top of the board.

7. Insert and solder the ten .01 mfd disc bypass capacitors as indicated on the

parts placement diagram (C1 through C6 and C10 through C13). Clip and save the excess leads.

8. Insert and solder the two 1 mfd tantalum capacitors (C7 and C14). Note that the + end must be inserted nearest the + mark on the board. The + end may be identified as the longer lead, the lead nearest a paint mark on the capacitor body, or the lead nearest a + mark on the capacitor body. Clip the excess leads.

9. Insert and solder the 220 pf mica capacitor (C9).

10. Insert and solder the 180 ohm, 1/2 watt resistor (R12). Mount the body of the resistor 1/4 inch above the surface of the board to allow efficient heat dissapation.

11. Insert and solder the 10 megohm resistor (R7).

12. Insert and solder the 100K resistor (R5).

13. Insert and solder the 22K resistor (R13).

14. Insert and solder the two 1.2K resistors (R1 and R6).

15. Insert and solder the four 2.2K resistors (R2 through R4 and R6). Clip and save the excess leads.

16. Insert and solder the IN5231 5.1 volt zener(CR3). Orient the cathode (the end marked with a black band) to the left.

17. Insert and solder seven IN4148 diodes (CR1, CR2, CR4, CR6 through CR8 and CR10). Orient the cathodes to the left. Be careful not to short the leads to the traces on the boards.

18. Insert and solder the IN4148 diode (CR9) with the cathode oriented toward the bottom of the board.

19. Carefully inspect the 5-30 pf trimmer capacitor (C8). The lead which connects to the bottom of the capacitor is oriented to the right (to ground). The round body is slightly flat at one side and is oriented to the top of the board. Insert and solder the trimmer, being careful not to allow excess solder to flow underneath it.

20. (If you do not have a crystal socket skip this section.) Trim the crystal socket's pins as shown in figure 3.1.1 to fit into the crystal holes.

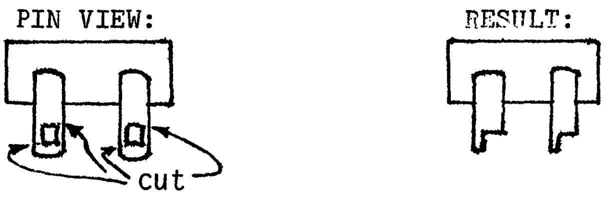


FIGURE 3.1.1 TRIMMING CRYSTAL SOCKET

Press the rear tab into the board hole provided for it. Solder the pins and the rear tab.

21. At this point, measure the resistance between pin 1 (+5V DC) and pin 2 (ground) on the 22-pin edge connector with an ohmmeter. If there is a low resistance, this indicates a bad capacitor or a solder bridge somewhere on the board.

22. Insert the board into the system at an unused I/O socket and turn the computer on. The presence of the board should not affect the operation of the system. If any difference is detected, it is probably due to a short somewhere in the I/O traces. Measure the voltage at the left lead of CR3. It should be approximately 5 volts.

23. Slide the 1 megahertz crystal into its socket or insert and solder the crystal into the board.

24. Insert all of the IC's into the proper sockets except IC21 shown on the parts placement diagram. Be sure to orient the notch on pin 1 away from the edge connectors. Again measure the resistance between connector pins 1 and 2. Reverse the meter leads and compare readings with the previous measurement. The resistance should be somewhat lower in one direction than the other, but not zero ohms. The same resistance in each direction indicates a reversed IC.

At this point, the following parts have not been installed: the 7442 decoder (IC 21), a IN4148 diode (CR5), four 2.2K resistors (R8 through R11), the IC socket headers, the ribbon cable assembly and the ni-cad batteries with associated hardware. IC21 is used with 8080-based systems and its use is explained later under Selecting Options. CR5 is omitted at this point to run the CMOS oscillator at a lower voltage, thus conserving battery life. It should be installed (cathode or band to left) if a higher degree of accuracy is needed for the clock function, since there is a slight shift in frequency without the diode (about 0.5 Hz) when operating under the computer power vs. battery.

25. Insert and solder the IN4148 diode (CR5) if it is to be used as described in the discussion above. (This step may be omitted until later).

26. A battery holder for the four AA ni-cad cells has been provided. The solder tabs at the end of each battery position should be wired as in figure 3.1.2. The molex connector should be wired in to the two end tabs. Install the batteries as shown. The molex connector will later be plugged into the wire wrap connector for the 36 pin socket at pins 29 and 30. It is important that the polarity is correct with the positive (+) side of the battery going to pin 29. Do not connect the battery now. The batteries are supplied in an uncharged state and will be totally discharged if used without charging. When the system has been debugged and is operating, the batteries may be connected and charged. The system should be powered up for about 20 hours to fully charge the batteries.

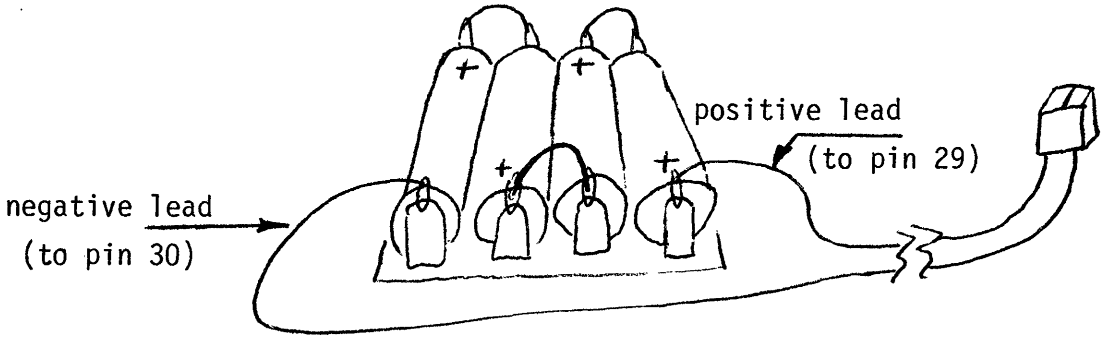


FIGURE 3.1.2 WIRING BATTERY HOLDER

27. Adjust the trimmer capacitor (C8) with a small screwdriver so that the slot is vertical and the soldered half of the capacitor is on the left and the insulator material is on the right. Turn the slot counter-clockwise approximately 1/16 of a turn.

This completes the board construction. Before plugging the board into the system, the address jumpers MUST be set up correctly. This is covered in the next section.

3.2 SELECTING OPTIONS

The Interrupt/Real Time Clock board has a number of options which may be set up according to the intended applications. This section describes each of those options. The initial setup to run the diagnostics program is also described under each option.

Since the board was designed to meet the needs of the average user, some features or options may not be available. The board is highly flexible, and can usually be modified by jumpers and cut traces to fit an individual's applications.

3.2.1 ADDRESSING JUMPERS

The board uses a contiguous 16-port block of addresses. The high four bits of the address must be set up by the address jumpers. (The lower four bits are decoded by the board logic). Select a block of addresses that are not used by any other system. The software supplied with the board uses the "E" block. In other words, it uses ports E0 through EF, as noted in hex, or 11 100 000 through 11 101 111, as noted in binary. The most significant bit (known as MSB, the leftmost bit) is shown in the diagram as A7. The other bits, A6, A5, and A4, are the following bits to the right. Each bit (A7 through A4) must be connected to one of its two associated levels (a high level or a low level). When the bit is 1, it should be jumpered to its high level (connecting A7 to 7H, A6 to 6H, etc.). Referrring to figure 3.2.1, set up the board address to the desired block. Use either component leads inserted directly into the socket or wire a header.

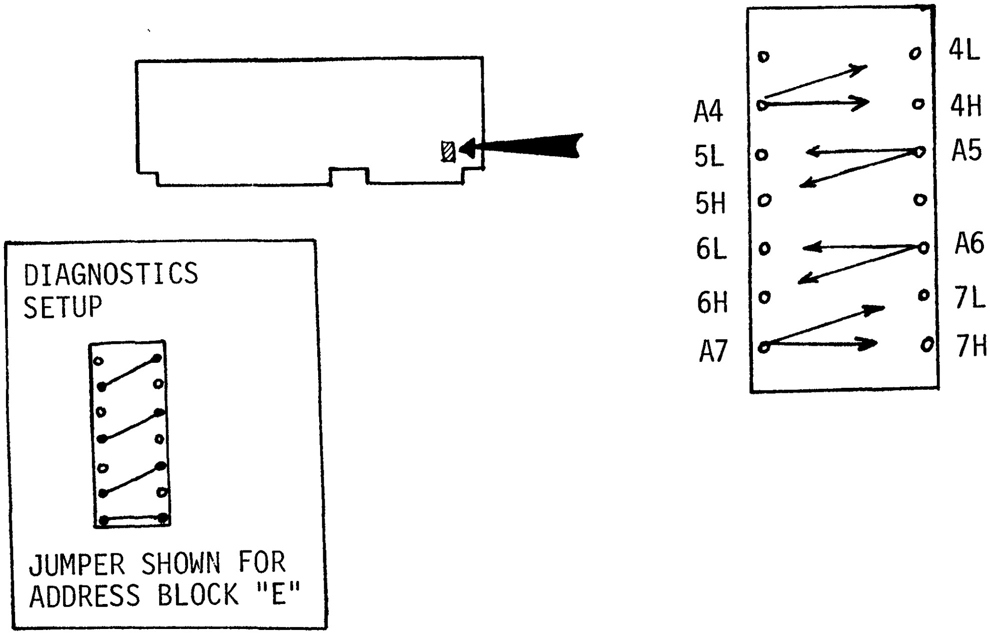


FIGURE 3.2.1 ADDRESSING JUMPERS

3.2.2 FREQUENCY JUMPERS

Two jumpers are used to select timing rates. The jumper called FINT brings the selected frequency to pin 11 on the 36 pin edge connector. The jumper called FREQ2 puts the selected signal names frequency on the internal bus so that it can be routed to the various inputs to the interval timer.

The FINT jumper is normally used to select a regular interrupt rate for timing, monitoring and other applications requiring a periodic interrupt. The name FINT stands for Frequency of INTerrupt, although this signal does not necessarily have to be used by an interrupt level.

When used to interrupt, the signal (at pin 11) should be wired to one of the interrupt inputs (pins 1 through 8) depending on the desired priority of interrupt. The signal output is an open collector TTL type.

FINT should be jumpered to one of the five frequencies located above as shown in figure 3.2.2. The two faster rates (10 micro seconds and 1 micro second) are normally too fast to allow the software overhead for interrupt processing. However, if one of these frequencies is needed off the board for other purposes, the jumper may be connected to one of them.

The second jumper, FREQ2 stands for FREQency 2. It selects a commonly-used frequency to be used by one of the on-board timers. (FREQency 1 also exists, but is software selectable).

FREQ2 should be jumpered to one of the seven frequencies located along the right side of the socket as in figure 3.2.2.

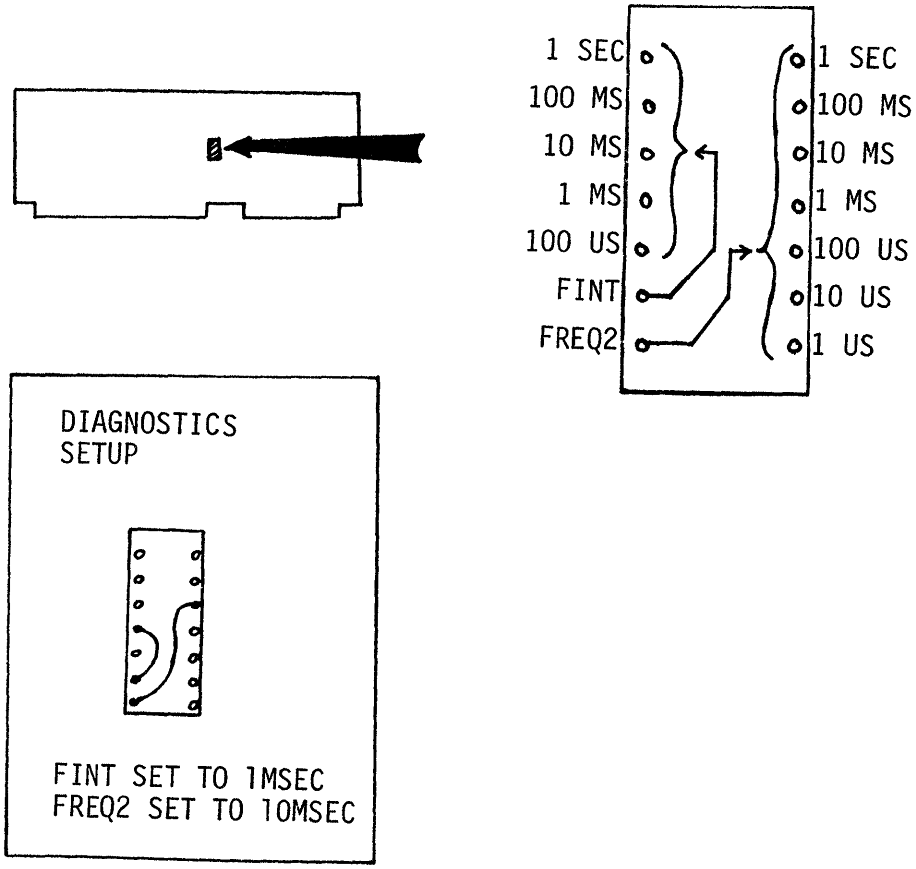


FIGURE 3.2.2 FREQUENCY JUMPERS

3.2.3 INTERRUPT TRIGGER JUMPERS

The level circuit on each of the eight interrupt levels may be set up to interrupt or trigger, on four types of conditions on its interrupt input line. The user may wish to interrupt when the line is low, high, moving from low to high, or moving from high to low.

When the interrupt level is set up to interrupt on a low or high line (static interrupt), the interrupt will be present as long as the interrupt line remains at that signal. This condition is useful when used with a device that will not toggle the interrupt line for each acknowledgement. If the device pulls the line to interrupt for a piece of information and then holds the line pulled for a second piece of information, the static interrupt will cause the controller to interrupt again to handle the second piece.

When the interrupt level is set up to interrupt on a rising or falling line (edge interrupt), the interrupt will only occur at the line transition and will be reset until the next similar transition. This condition is useful when there is no "hand-shaking" or computer reset of the interrupting signal. A slow square wave used for a time interrupt is a good example. The interrupt occurs for the high-to-low transition. If a static interrupt was used with this type of signal, the CPU would be "stuck" on the interrupt level for half the time!

Figure 3.2.3 shows an input line and the four interrupt conditions. The six pins should be connected to choose the desired condition. The lower half of the figure shows how each interrupt trigger socket allows two levels to be jumpered, using the top six pins or the bottom six pins. The two middle pins are unused. Level 0 is the highest priority level.

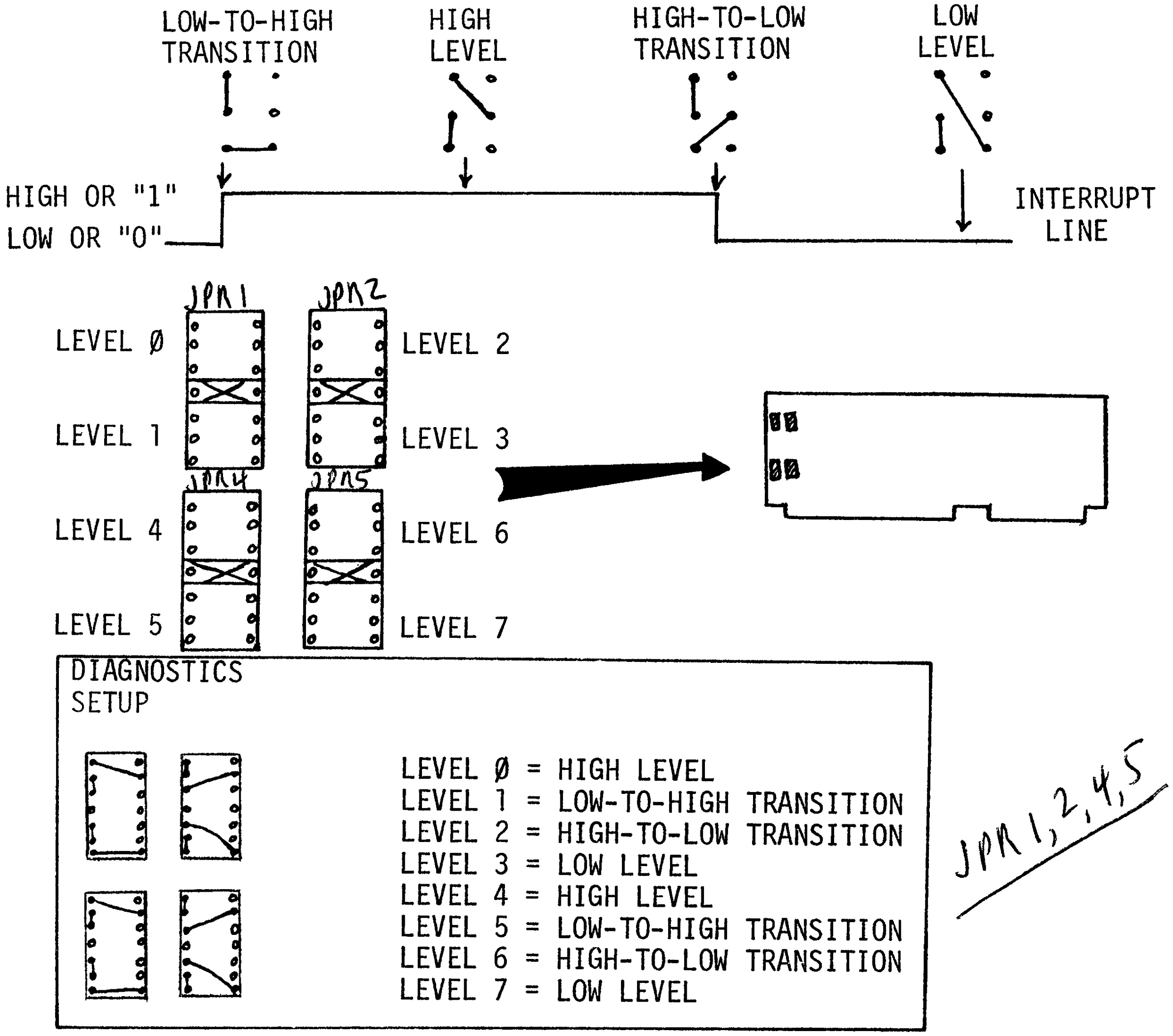


FIGURE 3.2.3 INTERRUPT TRIGGER JUMPERS

3.2.4 INTERRUPT MODE SOCKET

The Interrupt/Real Time Clock board can support all three interrupt modes of the Z-80 CPU as well as the 8080 interrupt structure.

Mode 0 allows the user to insert a restart instruction onto the data bus and allows the CPU to execute it. This mode is equivalent to the 8080 interrupt. Mode 1 causes a restart to location 0038 (hex). Mode 2 allows an indirect call to any location in memory. The address of the interrupt program is stored in a table somewhere in memory. The high byte of the table address is stored in register I (a Z-80 internal register) and the low byte is supplied by the controller.

The interrupt controller is set up for Mode 2. The low byte of the table is set up to 11 11x xx0. The xxx represents the interrupting level in binary. Thus, the table can exist in the last 16 bytes of any 256-byte section of memory. The format of the table is shown in Figure 3.2.4.1.

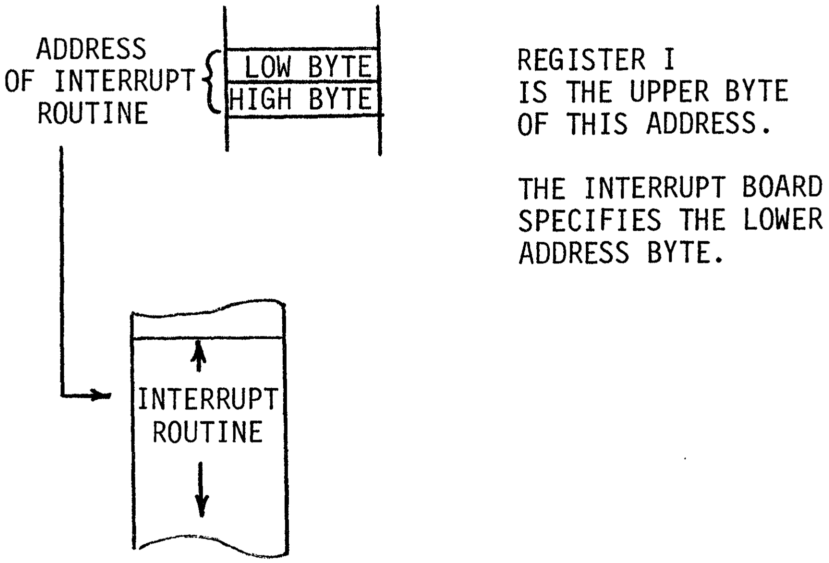


FIGURE 3.2.4.1 Z-80 MODE 2

For example, suppose the table is to be located in page 6. Furthermore, assume the interrupt routine begins at loction 0B 75 (013 165) for interrupts on level 2. Before enabling interrupts, register I should be initialized to 06 (006), using the "LD I,A" instruction. The table uses addresses 06 F0 (006 360) to 06 FF (006 377). Level 2 is the third interrupt (beginning with 0), so the address is three times two bytes into the table or address 06 F6 (006 366). At this address the low byte to address the interrupt routine is placed and at the next consecutive address the high byte is placed. Figure 3.2.4.2 shows the configuration.

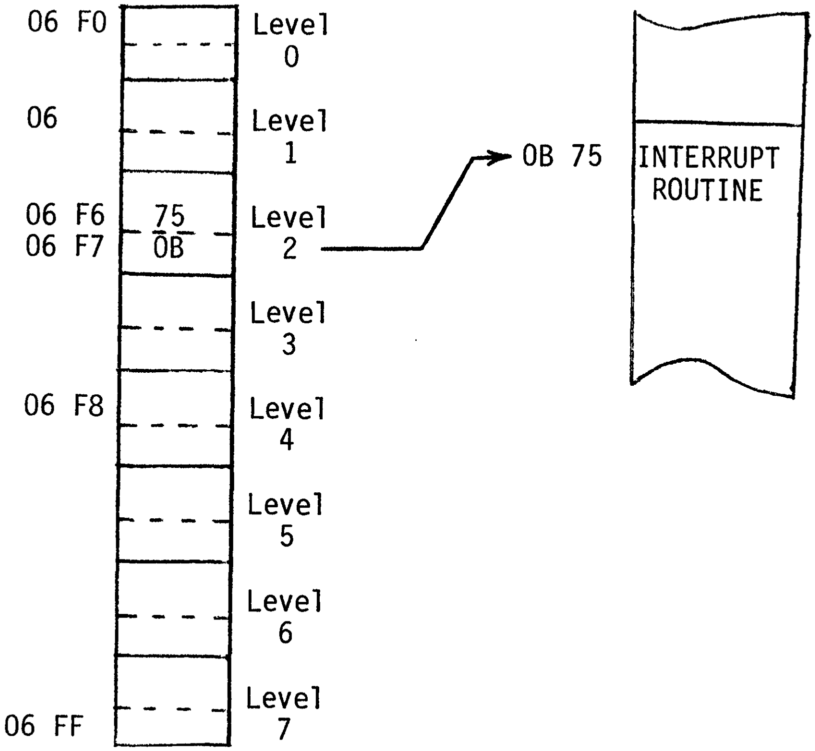


FIGURE 3.2.4.2 EXAMPLE OF Z-80 MODE 2 INTERRUPT

When an interrupt on level 2 occurs, Register I and the interrupt controller form the address 06 F6 and the address stored there becomes the address of the called routine. Execution of the interrupt routine then begins.

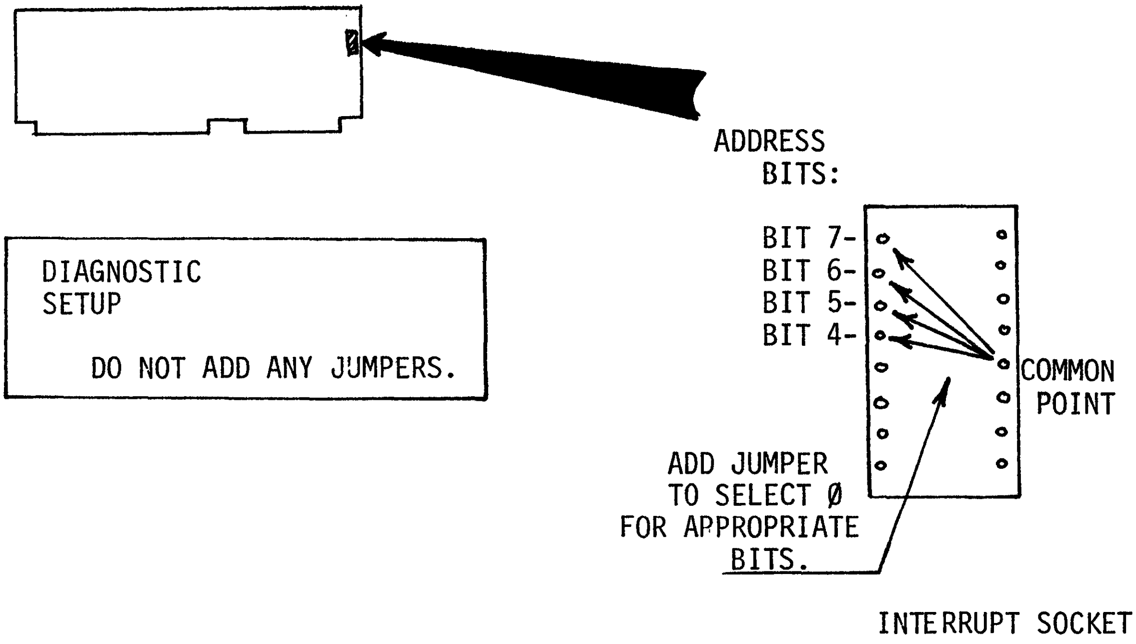


FIGURE 3.2.4.3 MOVING Z-80 MODE 2 VECTOR ADDRESS

Connections may be made to move the 16-byte table to other 16-byte areas in the same 256-byte page. To select the upper 4 bits in the lower byte of the table address, connect any bit which should be a zero to the common point as shown in Figure 3.2.4.3. This selects the 16-byte section where the interrupt vector addresses are stored.

NOTE: To select 8080 mode Z-80 Mode 0, the following traces must be cut:

> Turn the board over to the solder side and locate the four jumpers under the interrupt socket. Cut the traces as shown below, but DO NOT CUT the traces for the diagnostics, which will use Mode 2.

8080 Mode

Cut the four traces. Insert the 7442 (IC 21) into the interrupt socket.

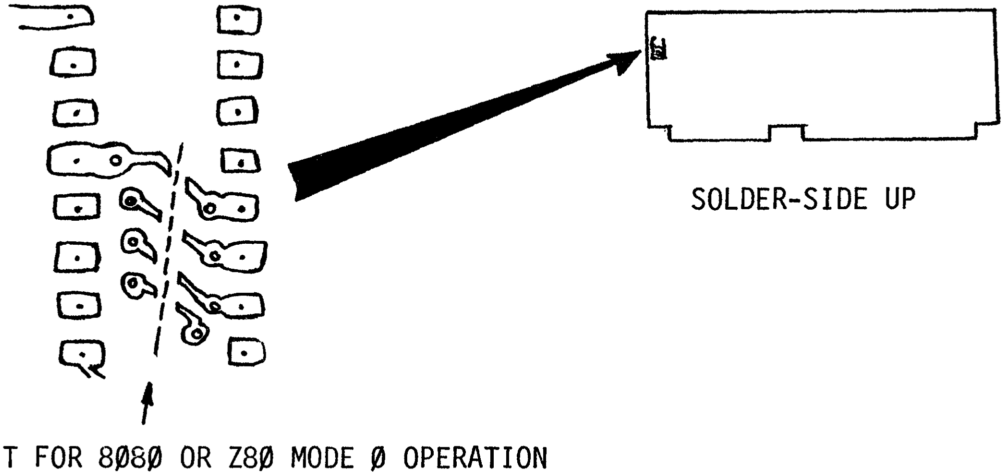


FIGURE 3.2.4.4 CUTTING TRACES

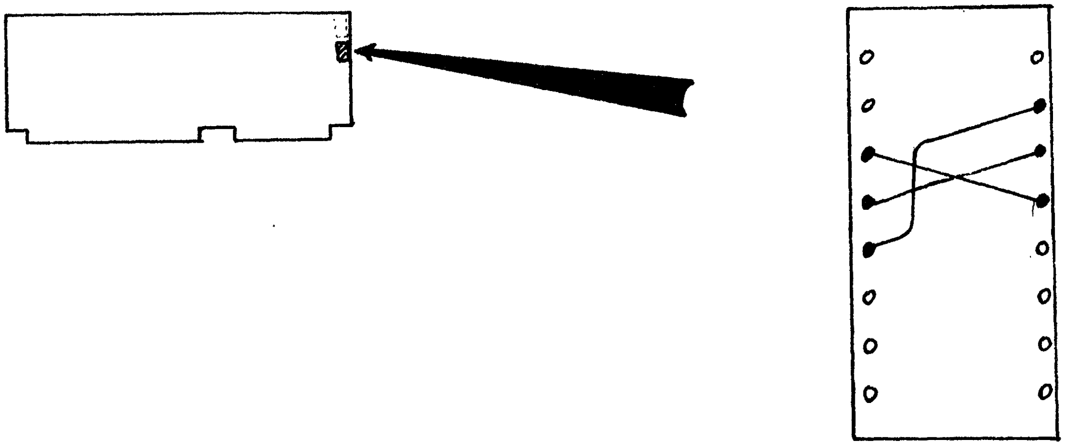


FIGURE 3.2.4.5 Z-80 MODE 0 JUMPER

Z-80 Mode 0

Cut the four traces. Wire pin 10 of the board's dual 36 pin connector to pin AU (opposite pin 39) on the CPU card. This connects INT to the IRQ input on the CPU card. Wire a header as shown in Figure 3.2.4.5 and insert it into the interrupt socket.

Z-80 Mode 1

DO NOT CUT the traces. If the traces have been cut, install a header to re-connect them as described for Mode 2.

Z-80 Modes 1 and 2 with cut traces

If Mode 1 or Mode 2 operation is desired after the traces have been cut, a header may be inserted to restore the cut traces as shown in Figure 3.2.4.6. Different addressing may also be added to this header (refer to the section on moving the Z-80 Mode 2 vector address).

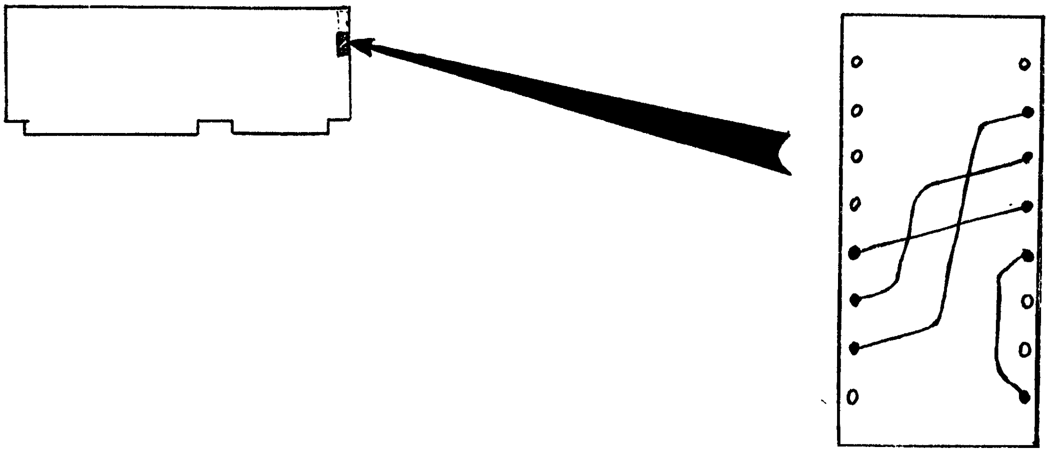


FIGURE 3.2.4.6 RESTORING CUT TRACES FOR Z-80 MODES 1 AND 2

3.2.5 OUTPUT PULLUPS

The timer outputs (TOUT0, TOUT1, and TOUT2) are of an open collector configuration. Pullup resistors may be installed if needed.

To pull up TOUT0, insert and solder R10 (2.2k).
To pull up TOUT1, insert and solder R9 (2.2k).
To pull up TOUT2, insert and solder R8 (2.2k).

A one second timer gate (on one second, off one second) is also available (called 1SG) and is also open collector.

To pull up 1SG, insert and solder R11 (2.2k).

3.2.6 TIME-SET SWITCH

To set the time, a momentary switch (not supplied) must be pressed in order to allow processor control of the real time clock. This is done so that powering up and down will not normally affect the clock unless this switch has been pressed.

The switch should be mounted in an out of the way location within the cpu cabinet. It will rarely be used, and may be omied entirely if a temporary jumper wire to ground is acceptable.

Run a wire from SETIME (pin 33 of the 36 pin edge connector) to one side of a momentary normally open switch. Wire the other side of the switch (or the common terminal) to ground.

3.2.7 EXTERNAL BATTERY OR CHARGER

The ni-cad batteries supplied will normally run the clock for more than two weeks on a full charge. If a battery with more capacity is needed, it may replace the supplied batteries at the 4.8 Volt battery input (BATTERY) if it is also rechargeable and about 4.8 Volts, or it may be connected to the external battery input (BAT2) if it is between 6.3 V and 14 V.

If the external battery is not intended to also charge the ni-cad batteries, an external power supply may be connected to BAT2 in order to charge the ni-cads without having to run the entire system. This would be necessary if the system was turned on less than an average of ten hours per week. This power supply should supply 9 to 12 volts and provide at least 40 milliamps of current. Several commercial "battery eliminator" modules will satisfy these requirements. Connect the negative lead of the power supply to ground. Connect the positive lead to BAT2 (pin 26 on the 36 pin edge connector).

3.2.8 EXTERNAL TIME BASE

In order to achieve an extremely accurate clock through crystal ovens, the TV network rubidium clock signals, the National Bureau of Standards time satellite, or other means, an external signal at 1 MHz may be used instead of the built-in crystal oscillator.

The signal should be TTL-compatible and brought in at FSTAN (pin 35). The board should be modified to accept this signal as follows:

Remove the 1 MHz crystal. Locate the two slightly larger feed through holes between IC 14 and IC 15. Note that the traces come to each hole but do not go elsewhere. Insert and solder a 10K resistor (not supplied) between these holes, mounting the resistor upright.

To protect the CMOS, install a IN5231, 5.1 volt zener diode (not supplied) between the top lead of R7 and the right lead of CR3. The cathode or band should be nearest R7.

3.2.9 FREQUENCY PRESCALER

The maximum usable range of the frequency counter is limited by the maximum speed of the input counter (74196, IC 51) which is about 50 MHz. The signal required on pin 19 must be TTL compatable, which is a further limitation.

A frequency prescaler may be used to extend the maximum frequency and improve the sensitivity of the computer. 500 MHz prescalers which divide by 10 to 50 MHz can be found or built to provide a versatile frequency-counting system.

The following circuit may be used to increase the sensitivity at low frequencies:

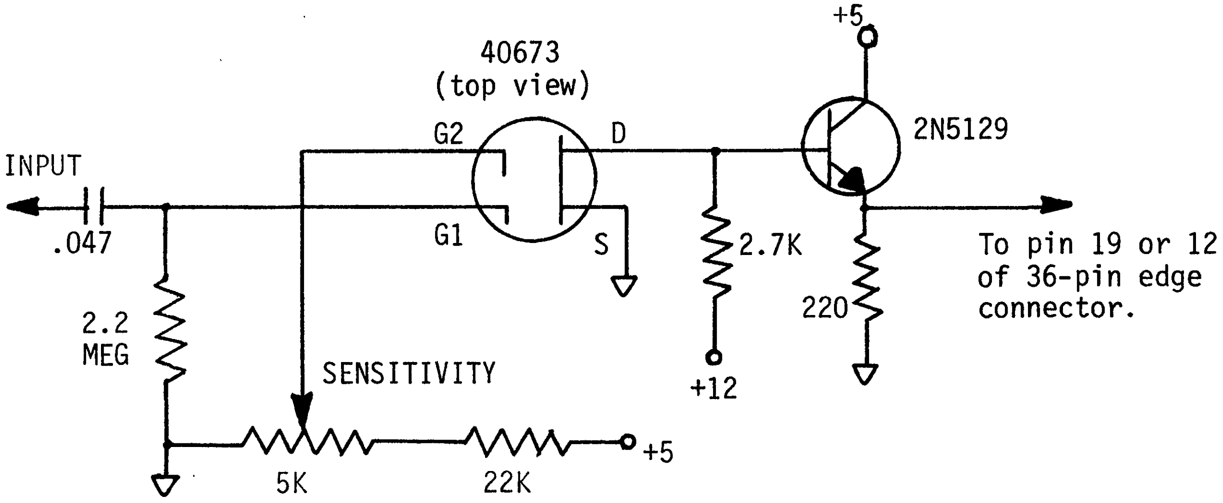


FIGURE 3.2.9 FREQUENCY COUNTER INPUT STAGE

3.3 CABLE INSTALLATION

The interrupt cable supplied connects the interrupt controller to the interrupt socket on the CPU card. It may be routed in two ways, depending on the card-supporting hardware in the system. The interrupt cable socket is located at the extreme top right of the board. The cable leaving the DIP socket at both the controller card and the CPU card should leave in the same direction, either both to the left, or both to the right.

Connect the cable in one of the two ways shown below:

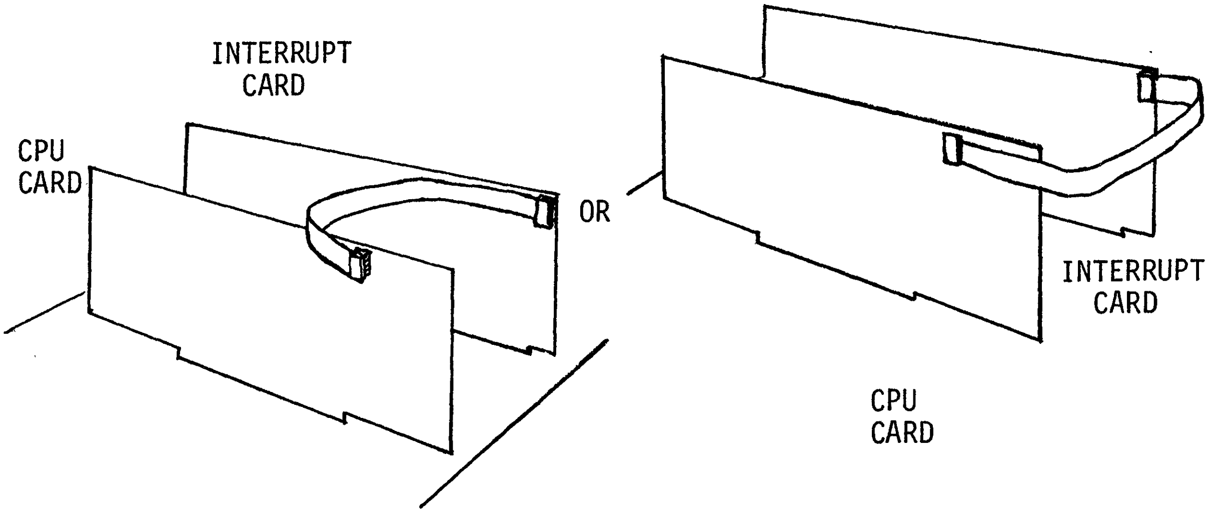


FIGURE 3.3 ROUTING OF THE INTERRUPT CABLE

CHAPTER 4

DIAGNOSTICS

The Interrupt/Real Time Clock board is basically a simple board to debug or check out, but a total check-out is a somewhat laborious process. This is due to the many options avaailable for configuring the board. A small solder short between two normally unused lines may not show up until the board has been in use for some time and a new application is implemented. In certain cases, the short may never show up at all. Thus, the amount of checkout needed depends on the projected uses of the board. If its total use cannot be defined, then a total checkout is recommended.

4.1 TEST PROCEDURES

The diagnostic routines are supplied in the first program of the cassette. The routines tie in very closely with the written descriptions of each test.

To run the test follow the written description for the preliminary measurements, if any, and then select the proper option. Most routines simply set up registers and decoders since no feedback is available from the board to automate the test. Thus, although several tests may be run, the signals on the board must actually be measured in order to achieve the true results.

In general, the tests are broken into various phases, and the space bar should be pressed to go from one phase to the next phase. If an error is found by the routine, it will loop on the error until the problem is cured (or RESET is pressed). Some routines will end through the space bar and others require the RESET switch.

A high impedence volt meter may be used to measure many of the levels. An oscilloscope is ideal since some of the reading requires either a frequency measurement or the observation of strobe phases. The second best instrument for these types of readings is an event counter (or a frequency counter that doesn't need a symmetrical waveform). A logic probe can also be used to detect whether pulses are present as well as levels, although it must have a high impedence input to work on any of the CMOS integrated circuits.

The test descriptions are written to point out correct levels and waveforms. If a problem is detected, the problem should be fixed before proceeding. Although some comments are usually made to help isolate the problem, a general knowledge of digital circuitry would be helpful. Measurements are usually made at the final destination of a signal and often the signal can be traced back to its source to find the IC or board trace that is causing the problem.

These routines will not find all of the problems on the board. The other programs on the tape and the BASIC listings will help test more of the function and should be used to test the board.

Since the interval timer chips can be set up to read or write two values in sequence, any glitches on the bus may cause false reads or writes. The glitches will get the internal logic of the timer chip out of sequence. A bus terminator may be needed to suppress any excess ringing on the I/O strobes and address lines. The circuit in Figure 4.1 may be used at the ends of the I/O and memory buses.

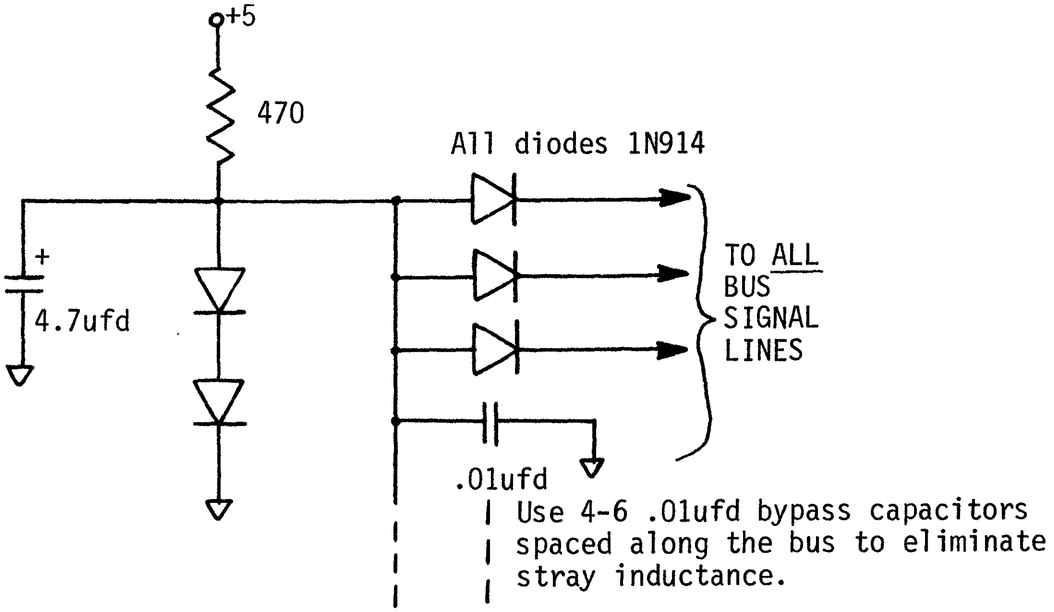

+5
470
All diodes 1N914
+
4.7ufd
TO ALL BUS SIGNAL LINES
.01ufd
Use 4-6 .01ufd bypass capacitors spaced along the bus to eliminate stray inductance.

FIGURE 4.1 BUS TERMINATOR SCHEMATIC

4.2 INITIAL TESTING

If power has never been applied to the board, the procedures described here may be used to bring the board up and to isolate any problems. Insert the board into an unused I/O slot. Do not connect either end of the interrupt cable. All jumpers on the board should be set up as described in the options section under set-up.

Power up the computer. If anything unusual happens so that the computer does not power up, and the supply voltages are within minimum limits, then remove IC's 44, 46, 48, 56. and 59. If the problem persists, it is probably in the races between the 22 pin connector and the sockets of the IC's just removed. If the problem goes away, refer to Section 4.4 on Addressing and I/O Operations.

Measure the signal at pin 11 of IC 7. Its frequency should be approximately 1 MHz. If no signal is present, turn the trimmer (C8) until oscillation starts. If no signal appears, check that all components are installed properly in this area, check the power supply and check that there are no shorts or opens in the traces. If no success, remove the crystal and IC 7 and verify that the capacitors are not shorted and the resistors are not open. Try replacing IC 7 or the crystal.

Momentarily short pin 5 of IC 7 to ground. Verify that the signal on pin 1 of IC 55 alternately is on for 1 second and off for 1 second. If not, refer to section 4.3 on the Count Chain.

Load the Diagnostic Routines from cassette into the memory. Select option 3 for initial testing to read the low byte of the time. If the test fails, refer to section 4.4 on Addressing and I/O Operations. Press the space bar on the keyboard for the next test. Check the following levels: High at pin 13 of IC 1, pin 1 of IC 2, pin 1 of IC 22, and pin 13 of IC 23. Press the space bar of the keyboard and check for a low at pin 1 of IC 1, pin 13 of IC 22, pin 13 of IC 23. If only one of the levels is wrong, check for shorts and opens (and possibly a bad IC). If more are wrong, refer to section 4.4 on Addressing and I/O Operations. Again, press the space bar on the keyboard. Verify that pins 13, 14, and 15 at the interrupt mode socket (IC 21) are low. If the signals are not all correct and the interrupt jumpers are set up as in Figure 3.2.3, then refer to section 4.5 on Interrupts.

This completes the initial testing. Press the space bar to return to the menu. If all the tests in this section worked, the board is basically working. At this point the routines described under Programming, in this chapter should eventually be used to complete check-out.

4.3 COUNT CHAINS

The count chain consists of the decade dividers and the seconds counter (calendar) as well as the time setting and disable circuitry.

To begin testing, momentarily ground pin 5 of IC 7 to allow the clock to count. Make sure pin 10 of IC 7 is low. Pins 3, 4, 10 and 11 of IC 8 should also be low. Check that all the proper frequencies are present at the frequency jumper socket. Pin 8 should be a 1 microsecond signal and the periods should lengthen by a factor of 10 on each pin to pin 14, which has a period of 1 second. Check for the selected signal for periodic interrupts (FINT) at pin 11 of the 36 pin edge connector. Also check that pin 22 of that connector is alternately on and off for 1 second each.

Run option 4 of the diagnostics. Momentarily ground pin 33 of the 36 pin edge connector (SETIME) to allow testing of the time-set function. Press space. If the message "STATUS AND RESET FUNCTIONS OK" appears, procede with the next paragraph. If a status failure occurs, and if either IC 16 pin 10 is low or IC 7 pin 5 is low, then refer to Addressing and I/O Operations, Section 4.4. Otherwise, make sure pin 8 of IC 7 goes low when grounding SETIME and that pin 10 of IC 7 goes high.

Once again test the frequencies at the frequency jumper socket. All should now be a high level except for the 1 microsecond signal.

If a reset failure occurred, make sure the data inputs (pins 1, 3, 6, 10, 13 and 15) of IC's 17 through 20 are all low. Make sure pins 11 of IC 9 and IC 32 are high. If no problem is found, refer to Addressing and I/O Operations, section 4.4.

Press space again. If a SETIME failure occurs, make sure that pins 9, 10, and 11 of IC 15 are high, low, and low levels respectively. Check that pin 5 of IC 15 is a 1 millisecond signal and that this signal is also on pins 10 of IC 9 and IC 32. Also check that at IC 33, pins 5, 6, 12 and 13 are all high. Pins

11 of IC 9 and IC 32 should be low and the two IC's should each be counting up in binary as checked at the data inputs of IC's 17 through 20. Press space bar to end the test.

4.4 ADDRESSING AND I/O OPERATIONS

The Interrupt Controller/Real Time Clock board has many latches and uses many I/O addresses. It is important that the addressing is unique and that all of the internal bus structure is functioning correctly.

The test program begins with inputs and outputs to all devices whose lowest 4 bits are "six". Thus it will output to ports 06, 16, 26, etc. (It will skip E6). If any devices at these addresses would be damaged due to the inputs and outputs, then skip this test and proceed to the next paragraph. If they won't be damaged or can be removed, run options 5 of the diagnostics. Check that there are no pulses on pins 4 through 7 and 10 through 12 of IC's 14 and 47. They should all be high levels. Press reset to end the test.

The second test checks that all inputs and outputs are decoding properly. Run option 6 of the diagnostics. This puts 1 strobe pulse on each latch or tri-state gate about once every millisecond. These strobe pulses should be observed at the following locations:

IC 17, pin 4
IC 18, pin 4
IC 19, pin 4
IC 20, pin 4
IC 26, pin 9
IC 27, pin 9
IC 28, pin 9
IC 29, pin 9

IC 4, pins 4 and 13
IC 6, pins 4 and 13
IC 12, pins 4 and 13
IC 13, pins 4 and 13
IC 43, pins 1 and 19
IC 5, pin 23

This test also writes low level to the latches. Measure voltages at the following locations to insure that all levels are as specified (low can be as high as about 1 volt on CMOS):

IC's	PIN(S)	LEVEL
49	10,11	low
1,2,22,23	1,13	high
8	1,5,9,12	low
7	5	high
15,37-42	9,10,11	low
37,39,41	3	low
55	5	low

Press the space bar and check that all of the above locations are now the opposite level from the level specified. Press space bar to exit from the test.

4.5 INTERRUPTS

The interrupt tests are run with the interrupt cable disconnected at the CPU card. This isolates problems to the Interrupt/Real Time Clock board. The interrupt trigger sockets should be set up as described in the set-up section.

Run option 1 of the diagnostics. Momentarily ground pins 1 through 8 of the 36 pin edge connector (IRQ). Then check that pins 15 through 22 on IC 5 are all at a low level. Press space and re-check those pins. All should be at a high level. Make sure pin 9 on the 36 pin edge connector (INT) is a high level. Press space again and now make sure that INT (pin 9) is a low level.

Press space. Make sure INT (pin 9) remains low. If a priority failure occurs, check the lines going into pins 1 through 4 of IC 5 as well as the strobe on pin 23 and the proper masking of interrupts on pins 20 through 23 of IC 5.

Press space. INT (pin 9) should go high, and should be reset by grounding one of pins 9 through 16 of the interrupt socket where the cable is attached. Pressing space will repeat this test in order to test all pins. To exit the test press RESET.

4.6 SELECTORS AND DECODERS

Many different signals are selectively routed from point to point on the Interrupt/Real Time Clock board. This test will check the signal lines for shorts or opens, the correct decode operation and also the dual decade prescaler.

Connect pin 12 of the frequency jumper socket to pin 19 of the 36 pin edge connector. Run option 8 of the diagnostics. Follow the test procedures for each test number.

				COMPARE TO:		
TEST#	IC	PIN	SIGNAL OR LEVEL	IC	PIN	REMARKS
1	54	9	--	*	13	Compare with edge pin open and ground
1	54	11	--	*	14	"
1	54	15	--	*	16	"
1	54	14	--	*	17	"
1	54	18	--	*	12	"
1	54	16	--	*	20	"
1	15	5	--	*	36	"
2	54	9	1 microsecond			
2	54	11	0			
2	54	15	1 microsecond			
2	54	14	0			
2	54	18	1 microsecond			
2	54	16	0			

* = 36 pin edge connector

2	15	5	1 second		
3	54	11	1		
3	54	14	1		
3	54	16	1		
4	54	9	2 seconds		
4	54	11	2 seconds		
4	54	15	2 seconds		
4	54	14	2 seconds		
4	54	18	2 seconds		
4	54	16	2 seconds		
4	15	5	100 milliseconds		
5	54	9	10 milliseconds		
5	54	11	10 milliseconds		
5	54	15	10 milliseconds		
5	54	14	10 milliseconds		
5	54	18	10 milliseconds		
5	54	16	10 milliseconds		
5	15	5	10 milliseconds		
6	54	9	--	48	2
6	54	11	--	48	2
6	54	15	--	48	2
6	54	14	--	48	2
6	54	18	--	48	2
6	54	16	--	48	2
6	15	5	1 millisecond		
7	54	9	--	54	10
7	54	11	--	54	10
7	54	15	--	54	10
7	54	14	--	54	10
7	54	18	--	54	10
7	54	16	--	54	10
7	15	5	100 microseconds		
8	54	9	--	54	13
8	54	11	--	54	13
8	54	15	--	54	13
8	54	14	--	54	13
8	54	18	--	54	13
8	54	16	--	54	13
8	15	5	10 microseconds		
9	54	9	--	54	17
9	54	11	--	54	17
9	54	15	--	54	17
9	54	14	--	54	17
9	54	18	--	54	17
9	54	16	--	54	17

9	15	5	1 microsecond	
10	54	18	10 milliseconds	Signal present every other second
11	54	18	100 milliseconds	"
12	54	18	1 second	"

4.7 CALIBRATION

To achieve an accuracy of about 30 seconds per year, the crystal oscillator must be set to within 1 cycle. Without any adjustment whatsoever, the accuracy will be about 1 minute per month worst case.

Measure the frequency at IC 56, pin 12 to reduce loading effects. Since the oscillator frequency is slightly higher when operating from the 12 volt supply than on battery, the frequency may be set to compensate for the anticipated usage of the computer. Since the components will also age and thus change frequency slightly, these effects can also be compensated.

If an extremely accurate frequency counter is available, it may be used to set the frequency. Many computers, however, do not have sufficient accuracy to allow this method to be useful for a highly accurate setting.

A simple, but lengthy process to calibrate the oscillator is to let the clock run and make slight adjustments after a period of several days. One second per two weeks would be a realistic goal. A BASIC program to read the clock is listed in APPENDIX D. A probe on the 1 second signal could be used to better show a slight drift over a shorter period of time.

A third method uses a WWV receiver with a BFO. WWV broadcasts time signals at 5, 10 and 15 MHz from the National Bureau of Standards, and the oscillator produces harmonics at these frequencies.

Tune in WWV at 10 MHz and set the carrier beat note at a comfortable frequency. (Be careful not to confuse the tone frequency with the carrier frequency: the tone will go away during the vocal time identification). Now mix the signal from the oscillator so that both can be heard. Adjust the trimmer to "beat" the notes against each other. If the two tones are 1 Hertz apart, the "quality" of the tones will appear to cycle once per second. This will set the oscillator frequency to within a tenth of a Hertz, or about 3 seconds per year. Component aging, temperature, and varying supply voltages will cause further error.

When using this method, it is often easier to stop the CPU or turn off the computer altogether, since the number of various harmonics generated by the computer may make finding the right frequency difficult. Also, the components should be run for a time at their normal operating temperature and voltage before setting the oscillator frequency.

CHAPTER 5

SOFTWARE INTERFACE

This section describes the readable and writeable latches on the board, as well as their applicability. A more concise representation of this information may be found in the Programmers Guide, Appendix A.

5.1 REAL TIME CLOCK

The calendar clock may be read by inputting x0 (hex) through x3 (hex). (NOTE: the x represents the upper 4 bits of the user-defined address). Also, all values will be given in hex. Input x0 is the lowest byte of the counter, while input x3 is the highest byte. Only the low six bits are valid in each byte. The counter can only count in binary.

BIT POSITION:

		7	6	5	4	3	2	1	0
INPUT:	x0			5	4	3	2	1	0
	x1			11	10	9	8	7	6
	x2			17	16	15	14	13	12
	x3			23	22	21	20	19	18

FIGURE 5.1.1 BITS OF CALENDAR CLOCK

A utility output byte is used to set the calendar clock. Bits 3 through 7 of output x6 facilitate setting the time. Bits 3 through 5 reset the decade countdown chain, the first 12 bit counter, and the second 12 bit counter, respectively. Bit 6 controls the clocking (or set) speed and bit 7 disables bits 3 through 6 until the time set switch is manually pressed.

BIT POSITION:

	7	6	5	4	3	2	1	0
OUTPUT: x6	Disable Time Set	Fast Clock	Reset 2^12 Second	Reset Second	Reset Milli-Second	UTILITY BYTE		

FIGURE 5.1.2 TIME SETTING FUNCTIONS

The time may be set or represented in several ways. The simplest is storing to the exact second the time and date the counter is set on disk or tape. The counter can then start at zero and the stored value plus the elapsed seconds is the current date and time.

A second method is a little more involved, but sets up the counters to a preset value normally representing the time from a fixed reference point as the first day of the first or seventh month of the year. The fast clock is used to set the counters to the proper values. When the fast clock bit is a 1, both 12 bit counters in the calendar section are set to count simultaneously by the fast clock. (The fast clock is FREQ1 and so its actual clocking speed can be selected as described in a later section).

In the sequence of events, one of the 12 bit counters is held reset while the other is clocked to its intended value minus the intended value of the other 12 bit counter. Then both counters are clocked simultaneously until the count in both counters become the intended value. The countdown chain is then reset to inhibit counting, the fast clock turned off, and the computer waits for a command to disable the time set mode, thus de-activating bits 3 through 6 and proceding with its timekeeping.

It should be noted that the disable time set signal may be a 0 and the time-setting bits will NOT be re-enabled.

In fact, when the computer is powered up, those bits may be initially disabled and the set time switch must be pressed to enable them. Also, once the disable set time bit is turned on it is remembered as long as the CMOS has battery power (unless reset by the switch).

To determine whether bits 3 through 6 are active the time enabled status, bit 6 of input x4, can be read. If the bit is a 1, the clock can be set. If the bit is a 0, the set time button must be pressed to set the time.

		BIT POSITION: 7	6	5	4	3	2	1	0
INPUT:	x4		Time Enabled						

STATUS BYTE

FIGURE 5.1.3 TIME ENABLED BIT

Note that if reset microseconds is on (when active), the decade frequency divider is reset and FREQ1 is stopped.

5.2 INTERVAL TIMER

The 8253 interval timer has several functional modes, the particulars of which are best seen in the manufacturer's data sheets. A general description will be given here which will be sufficient to begin using the device in most applications.

The timer has three 16-bit counters and thus three control words to configure each counter. The information in the control word selects the operational mode, the selection of Binary or BCD counting, and the loading sequence of the count.

Each counter is a presettable down counter, and has an input, gate and output which are configured by the selected mode.

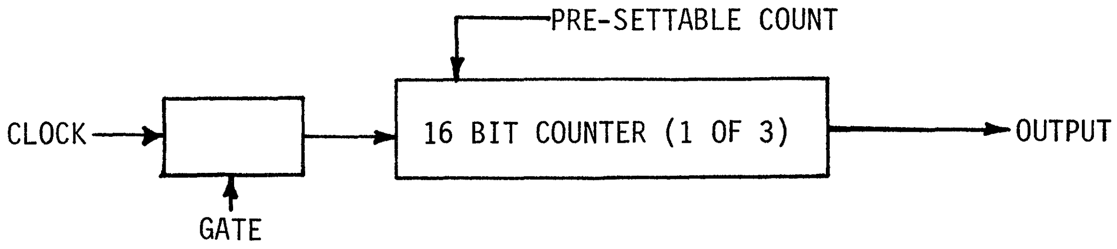


FIGURE 5.2.1 CLOCK, GATE AND OUTPUT PER COUNTER

The counters may also be read and a special command allows the logic to "capture" the count so that the counters may be read while in operation.

The internal timer must be programmed by the systems software to set it up for proper operation. Each of the counters must first be set up with a control word before data is entered. The control word is given by outputting to port x0 or xC. (Both are equivalent, due to the board's mapping of the timer's I/O addresses).

		BIT POSITION: 7	6	5	4	3	2	1	0
OUTPUT:	x8 or xC	SC1	SC0	RL1	RL0	M2	M1	M0	BCD

FIGURE 5.2.2 CONTROL WORD FORMAT

SC1 and SC0 select the counter for which the rest of the control word applies.

SC1	SC0	
0	0	SELECT COUNTER 0
0	1	SELECT COUNTER 1
1	0	SELECT COUNTER 2
1	1	ILLEGAL

FIGURE 5.2.3 SELECT COUNTER FORMAT

RL1 and RL0 control the reading and loading sequence for the counter data. They can set up sequence dependent operations and also latch the count for reading "on the fly".

RL1	RL0	
0	0	Counter latching operation
1	0	Read/Load most significant byte only
0	1	Read/Load least significant byte only
1	1	Read/Load least significant byte, most significant byte.

FIGURE 5.2.4 READ/LOAD BITS

BCD controls the type of counting.

BCD	
0	Binary Counter (16 bits)
1	BCD Counter (4 Decades)

FIGURE 5.2.5 BCD BIT

M2, M1, and M0 select the mode of the counter. The Programmer Guide in Appendix A describes the configuration of the CLOCK, GATE, and OUTPUT signals for the various modes.

M2	M1	M0	
0	0	0	Interrupt on terminal count
0	0	1	Programmable One-Shot (retriggerable)
0	1	0	Rate Generator (pulse output)
0	1	1	Rate Generator (square wave output)
1	0	0	Software triggered strobe (pulse output)
1	0	1	Hardware triggered strobe (pulse output)

FIGURE 5.2.6 MODE FORMAT

The data read or load sequence is selected by RL1 and RL0. When RL1=RL0=1, the sequenced operation MUST be performed before the control word is again written for that counter. Half of the operation is not allowed.

The data or count for counter 0 is input/output to port xB or xF. Counter 1 is input/output to port xA or xE, and counter 2 is input/output to port x9 or xD. NOTE: WHEN DATA IS INPUT FROM THOSE COUNTERS, THE BITS WILL BE INVERTED.

5.3 SIGNAL ROUTING FOR THE INTERVAL TIMER

Each of the three 16 bit counters has a clock input, a gate input and an output. Each input can have any of several signals routed to it. The outputs to select the appropriate signals are shown in Figure 5.3.1.

		BIT POSITION:							
		7	6	5	4	3	2	1	0
OUTPUT:	x2			Gate 0 Select			Clock 0 Select		
				C	B	A	C	B	A
	x3			Gate 1 Select			Clock 1 Select		
				C	B	A	C	B	A
	x4			Gate 2 Select			Clock 2 Select		
				C	B	A	C	B	A

FIGURE 5.3.1 GATE AND CLOCK ROUTING

The decode of each three bit value to the selected signal is shown in Figure 5.3.2.

C	B	A	
0	0	0	See EXTERNAL Table
0	0	1	Programmable value for gate; 1 MHz for clock
0	1	0	1 Second
0	1	1	FREQ1 (see selection chart)
1	0	0	FREQ2 (jumper selected)
1	0	1	Output of counter 0
1	1	0	Output of counter 1
1	1	1	Output of counter 2

FIGURE 5.3.2 GATE/CLOCK SELECTS

When CBA=000 external signals are selected from a pin on the 36 pin edge connector. The table in Figure 5.3.3 associates the counters to the external pins.

Gate 0	Pin 14
Clock 0	Pin 13
Gate 1	Pin 17
Clock 1	Pin 16
Gate 2	Pin 20
Clock 2	Prescaler

FIGURE 5.3.3 EXTERNAL CONNECTIONS

The prescaler is one of four signals and is selected through the utility byte.

BIT POSTION:

		7	6	5	4	3	2	1	0
OUTPUT:	x6							FS1	FS0

UTILITY BYTE

where:

FS1	FS0	
0	0	External input, pin 12
0	1	Gated/1 from pin 19
1	0	Gated/10 from pin 19
1	1	Gated/100 from pin 19

FIGURE 5.3.4 PRESCALER FUNCTION

When CBA of the GATE/CLOCK selects = 001, the gate value may be directly output through output port x5.

BIT POSITION:

		7	6	5	4	3	2	1	0
OUTPUT:	x5						P2	P1	P0

FIGURE 5.3.5 PROGRAMMABLE GATES

P2, P1 and P0 go to gate 2, gate 1 and gate 0, respectively. Thus, each gate can be turned on or off by software control.

This same byte selects FREQ1.

BIT POSTION:

		7	6	5	4	3	2	1	0
OUTPUT:	x5			FC	FB	FA			

FIGURE 5.3.6 FREQ1 ROUTING

FREQ1 selects one of the internal frequencies as given by the table in Figure 5.3.7.

FC	FB	FA	
0	0	0	External frequency, pin 36
0	0	1	1 second
0	1	0	100 Milleseconds
0	1	1	10 Milleseconds
1	0	0	1 Millesecond
1	0	1	100 Microseconds
1	1	0	10 Microseconds
1	1	1	1 Microsecond

FIGURE 5.3.7 FREQ1 SELECTS

Some of the signals may be monitored from status byte.

		BIT POSITION:							
		7	6	5	4	3	2	1	0
INPUT:	x4				1SEC	FREQ1	OUT2	OUT1	OUT0

STATUS BYTE

FIGURE 5.3.8 STATUS BYTE SIGNALS

OUT0, OUT1, and OUT2 are the outputs of the counters, BUT INVERTED! (If the actual output is a 1, the bit would be a 0). FREQ1 is the FREQ1 signal inverted and 1SEC is the 1 second gate signal, also inverted.

The one second gate signal is on for one second and off for 1 second. It gates the prescaler to allow a 1 second burst of counts. When the status bit is a 0, the prescaler is counting and when the bit (1SEC) is a 1, the frequency may be read. Note that only counter 2 is used with the prescaler.

5.4 INTERRUPT CONTROLLER

The interrupt controller is based on the 8214 chip. It is necessary to write the current interrupt level to the chip so that it can determine if the incoming interrupt signal is of higher or lower priority.

		BIT POSITION:							
		7	6	5	4	3	2	1	0
OUTPUT:	x0					LEN	L2	L1	L0

FIGURE 5.4.1 INTERRUPT LEVEL

L2 through L0 is the current level of interrupt on which the programming is executing. Level 0 is the highest level; level 7 is the lowest. LEN is the level-enable bit and should be 0 to enable L2 through L0. If no interrupt is active, LEN should be a 1.

The interrupt mask byte is accessed by an output to port x1.

		BIT POSITION: 7	6	5	4	3	2	1	0
OUTPUT:	x1	MSK0	MSK1	MSK2	MSK3	MSK4	MSK5	MSK6	MSK7

FIGURE 5.4.2 INTERRUPT MASK

When MSK0 is a 1, level 0 is masked and any interrupt which occurs on that level will be ignored. Likewise for each of the other masks.

To process interrupts, the interrupt controller must be initialized. When the system is powered up, an interrupt may be pending. To reset that interrupt without going through the actual interrupt routine, bit 2 of the utility byte must be written to a 1, and then reset back to a 0.

		BIT POSITION: 7	6	5	4	3	2	1	0
OUTPUT:	x6						Reset Int		

UTILITY BYTE

FIGURE 5.4.3 INTERRUPT INITIALIZATION

Also, the current level must be initialized and all pending interrupts must be reset. To reset pending interrupts, the mask byte is written with all 1's so that all levels are masked. Then the reset interrupt bit may be momentarily set to 1 and the level byte may be written with 08 (base level). Finally, the active interrupt levels are unmasked.

Appendix H shows a sample interrupt program. Two bytes from main memory are used to save the current level and the current mask. When the interrupt occurs, the program saves the accumulator, puts the interrupted level into the stack, and stores the new level into the controller's level byte as well as into the main storage byte (called CURLEV).

When returning from the interrupt, the interrupt request is reset (by turning the mask bit for that level on and off) and the previous level is fetched from the stack and output to both the controller and the CURLEV byte in main memory. The accumulator is then restored and the interrupt routine returns.

CHAPTER 6

PROGRAMMING

The software for the Interrupt/Real Time Clock board can be simple or complex, depending on the application involved. But even the most complex applications can be broken down into smaller and smaller pieces, each of which provides a simpler function to be integrated into the whole. This section illustrates sample programs which use the board's facilities, illustrating some of the programming steps used to accomplish the functions.

The assembly level programs can be found on the tape supplied with the board. The BASIC programs are not on the tapes, since many different versions of BASIC and BASIC files exist. The user should enter the programs from the listing, tailored to the particular version of BASIC. The listings themselves were taken and run on The Digitl Group's MAXI BASIC, although almost any BASIC will work as long as it can be made to do INPUTs and OUTPUTs to the comptuer's I/O ports.

6.1 SETTING THE CLOCK

There are several ways to set the calendar clock up for day and date timekeeping. Some of the methods were outlined in the Software Interface Section. One of the methods consisted of resetting the counter to all zeros. In this method, the starting day and time is recorded on a non-volatile media, such as cassette tape or diskette. The counter then becomes the number of seconds after that recorded time.

The second method uses a commonly known starting point, such as the first day of the half-year, so that a day and time might be calculated without resorting to automatic storage on a file medium.

As described in the Software Interface, the method holds one half of the counter reset while toggling the other half, and then counts simultaneously to the desired count. More care is needed, however, to prevent false clocking of the calendars.

This false clocking occurs when switching between a fast and normal clock, or changing clock speeds. Unless enough care is used to prevent unwanted signal transitions, extra counts can be created.

The program example of setting the clock, Appendix C, allows for these transitions by comparing for a value less than the matching value and then slowly controlling the clock until the desired value is reached. Also, if a counter value is read during the time the counter is incrementing, a false value may be read in. Therefore, at key points note that the counters are read twice to assure the correct value.

To use the program load the second cassette program on the demonstration tape. Using some sort of memory editor load the locations TIME 1 through TIME 7 (07DD to 07E3) to the time desired. TIME 1 holds a binary value for unit seconds; TIME 2 = 10 seconds; TIME 3 = unit minutes; TIME 4 = ten minutes; TIME 5 = unit hours; TIME 6 = ten hours (note that hours is in 24 hour time) and TIME 7 = the day of the half year in binary. The eleventh day (0B) would be January 11 or July 11 (assuming the second half year begins July 1).

Thus, the time 2:49:05 P.M., August 2, would appear as the sequence 05,00,09,04,04,01,21, in hex, or 005,000,011,004,004,001,041, in octal. These values may also be located by program control.

Select the menu number (7) and the little program CALL will invoke the SETIME subroutine to set up the correct time. If the program returns immediately, then it is possible that the time setting function is disabled. SETIME returns with a 1 in the accumulator if this error condition occurs. To enable the timesetting functions, momentarily press the time-enable switch or momentarily ground pin 33 on the connector. Then run the program again.

The short calling program may be replaced by a user-provided program as long as the seven time bytes are set up beforehand. A return accummulator of 0 indicates successful completion. SETIME does not disable the time setting function so a new time may still be set.

6.2 READING THE CLOCK

To determine if the clock setting is correct, the counters may be read by a sequence of input commands. A BASIC program (Appendix D) may be typed in from this listing to read the clock and display the day and time in a readable format.

When the four counter bytes are read, they should be read starting from the lowest, least significant byte to the most significant byte. Then the least significant byte is read again to make sure it hasn't changed. This method obtains a stable count by determining that no clock pulses had occurred between or during the reading of values.

Once the clock is read, arithmetic operations are performed to extract the day and time. If the time is correct, the clock can be started by typing CONT to continue the program and disable the time setting function. This should be done at the moment the entered time and the current time are the same. The clock is now running and can be read again by re-running the program.

6.3 PERPETUAL TIMING

Perpetual timing is merely a scheme to keep track of the correct time, date, year, etc. As was mentioned earlier, this technique consists of keeping a reference date and counting the seconds since that reference. Thus some form of non-volatile storage is needed.

Since the counters will overflow after about half a year, a method is needed to count the overflows and store them in the same permanent storage. This update process should be done on the average of once every three months.

Since is it likely that the computer may not be turned on at the moment the counters overflow, some flag must be set in order to determine that an overflow did occur. If this flag is to be turned on and stored when the high bit of the counter is a "one", then there is a period of three months during which the computer can "notice" the condition and set the flag. Then, after the counters have overflowed, the computer has three months to recognize that although the high bit is a "zero", the flag is turned on and therefore an overflow occurred. It then increments the overflow counter and resets the flag so that the overflow counter won't be incremented again (until the next overflow).

To keep the scheme from vulnerability to power blackouts two sets of flags, dates and overflows should be kept so that when the permanent storage is beng updated, a power failure at that critical time can be recoverable by using the alternate set of values.

Over such a long time span, the clock will gain or lose time. Since the calendar is not settable without pushing a button to enable the set electronics and since this also produces a vulnerability to power outages, it is best to let the counter keep counting. To make the adjustment, the base date and time which was originally stored is altered by the appropriate amount and restored.

6.4 MEASURING INTERVALS

Appendix E shows a BASIC program which measures intervals. The three 16 bit counters are cascaded to count up to 48 bits of the 1 MHz clock. Accuracy is to 1 microsecond, although the BASIC input shown in this example cannot demonstrate the accuracy.

To run the program, type a value (any value) after starting the program: this starts the timer. Type another value at the end of the interval and the timer is stopped and the value is read out.

Note that when the counters are loaded with the maximum count, the clock inputs at that time are all 1 MHz for each counter. The reason is that the data loaded into a counter does not become active until that counter receives a clock pulse on its clock input.

Also for simplicity of programming, the counters are counting at 1 less that the maximum count of each counter. Since the data read from the counters is inverted, a maximum count of all ones will read as zero and as the counters count down, the data will appear to count up.

6.5 CREATING TIME DELAYS

Several of the modes can be used to create a time delay. If the intent of the delay is to create an interrupt, modes 4 and 5 will generate a one-clock-width pulse. Mode 0 and 1 will change output states permanently when the time delay is complete.

Modes 0 and 4 do not depend on the gate input to initiate counting, although the gate input should be high to enable counting. Mode 1 and 5 begin counting with the rising gate transition.

One counter may be loaded during one count to change the period of the counter on the next count. However, if the time of the count is too short to allow the software to load the new values, a second counter may be prestored with the required value. Thus, tying a pair of counters together to gate each other is one way to produce delays when one of the delays is too short for software interaction.

6.6 FREQUENCY COUNTER

Counter number 2 has a two decade prescaler to facilitate frequency measurements. Appendix F contains a BASIC program to make measurements.

The counter is set up for mode 0 in order to count the number of clocks per second. The clock input is automatically gated on for exactly 1 second. This allows the counter to be read directly in Hz. However, various stages of the prescaler may be used to allow higher frequencies.

The frequency counter shown here has an upper limit of about 6.5 MHz, since a larger count would overflow the counter. The use of a second counter configured in a similar manner to the example of an interval measurement counter would extend the maximum frequency limit.

Overflow is determined by monitoring the output signal from counter 2. If this signal is ever low, then overflow is detected.

The counter also autoranges. This is done by selecting the appropriate output from the prescaler. If the frequency count is low, less stages of the prescaler are used to increase the count and accuracy. If the count gets too high or overflows, then more stages are used to divide the count down to a more manageable number.

6.7 PLAYING MUSIC

The final program in assembly language in Appendix G, is a routine that plays music. This routine demonstrates the square wave generator counter mode plus interrupt programming.

The circuit of figure 6.7.1 should be built to interface the computer to an audio amplifier. Each counter will be used to play a note or frequency and these notes are added together and fed to the amplifier.

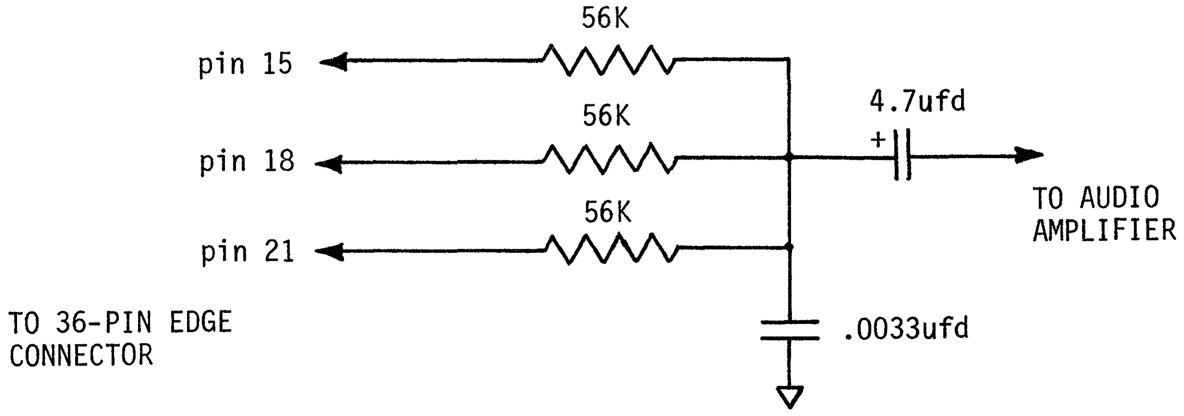


FIGURE 6.7.1 INTERFACE CIRCUIT TO PLAY MUSIC

The interrupt rate of 1 millisecond should be selected and levels 2 and 4 should be set up for high-to-low transition and high level inerrupts, respectively.

The program executes from a command table. The Table is set up to specify the length of each note or rest, the counter which will play it and the coded frequency of the note.

In a command byte of this list, the low two bits select the counter, the next three select the duration of the note, and bit 6 selects between a rest or a playing note. The data byte, which is only used with a playing note, selects the frequency by means of a table lookup. The lower four bits select the notes on the scale and the next three bits select the octave. The data byte immediately follows the playing note byte.

	LENGTH	POSITION
0=rest	000=stop	01=counter 2
1=note	001=64th note	10=counter 1
	010=32nd note	11=counter 0
	011=16th note	
	100=8th note	
	101=quarter note	
	110=half note	
	111=whole note	

FIGURE 6.7.2 COMMAND FORMAT FOR MUSIC

The program starts at INIT and shows some of the counter and interrupt initialization steps. Each note value for the counter and length is set up on the base or non-interrupt level. This information is passed to the level 4 interrupt routine, which transfers the information to the counters at the appropriate time. This time is determined by the level 2 routine, which counts milliseconds and timing beats and knows when the next note is due. The level 2 routine causes a level 4 interrupt when level 4 has something to do.

Load the last program from the cassette and choose option 3. After a few moments, the music will begin.

6.8 INTERRUPT PROGRAMMING

The music routine and the sample interrupt-handling excerpts in Appendix H illustrate interrupt controller programming. Basically, the interrupt level must be given to the controller to determine priority of interrupts. The previous interrupt level must be saved in order to restore the previous priority. Before returning, the interrupt request latch must be reset (through the mask byte) to keep from re-interrupting. This must be done before the old interrupt level is output or else a second interrupt will be remembered for the current level.

Since the interrupts can occur anywhere, any common bytes used both by the base and interrupt levels or between interrupt levels should be carefully analyzed for possible timing problems. For example, the base level may wish to add a value to a register and the interrupt level may wish to reset the register. Between the time the value is read, added and stored again, the interrupt may occur, causing the register to be reset. Then the base routine regains control and stores back the added value. Thus the interrupt routine's reset of the register is lost.

These problems are commonly solved by the use of the DI and EI instructions on the lower level. They keep the interrupt from occuring during critical processing times. Another method is a set of flags which indicate the routine that has control of the register in question. Still another method gives each routine read-only registers and exclusive-write registers. Thus, only the base level can write a particular register. Interrupt problems usually stem from such conflicts and careful timing analysis of the program can usually resolve such problems.

APPENDIX CONTENTS

APPENDIX A - PROGRAMMER GUIDE

CONTROL WORD FORMAT: SC1 SCØ RL1 RLØ M2 M1 MØ BCD (A1, AØ = ØØ) WRITE ONLY

SC1	SCØ	SELECT COUNTER	RL1	RLØ	READ/LOAD	BCD	MODE
Ø	Ø	Select Counter Ø	Ø	Ø	Counter Latch	Ø	Binary
Ø	1	Select Counter 1	Ø	1	Least Significant Byte		
1	Ø	Select Counter 2	1	Ø	Most Significant Byte	1	BCD
1	1	Illegal	1	1	Least/Most Sequence		

M2	M1	MØ	DEFINITION	MODE SELECT OUTPUT	LOAD 1st BYTE	LOAD 2nd BYTE	LOAD DURING COUNT	LOW OR FALLING GATE	RISING GATE	HIGH GATE	OUT CHANGE	OUT INITIALIZE
Ø	Ø	Ø	Interrupt on Terminal Count	Ø	Stop	New Count	New Count(same)	Disable Counting	---	Enable Counting	Final Count (Hi)	Load Mode
Ø	Ø	1	Retriggerable One-shot	Same	----	----	Active for next trigger	---- ----	Initial Count and Reset out after Clock	----	Hi on final Count	Low on Trigger
X	1	Ø	Rate Generator (÷N) Pulse	1	----	Start Count (Gate=1)	Active for next period	Stop Counting; Out goes Hi	Initiate Counting	Enable Counting	Lo for 1 clock after Count	------
X	1	1	Rate Generator (Square Wave)	1	----	Start Count (Gate=1)	Active after Transition	Stop Counting; out goes Hi	Initiate Counting	Enable Counting	Lo for last ½ Count	------
1	Ø	Ø	Software Trigger Strobe (pulse)	1	----	Start Counting	Next period (Gate=1), New Count (Gate=Ø)	Stop Count	-----	Enable Counting	Lo for 1 clock after Count	------
1	Ø	1	Strobe (pulse)	----	----	----	----	----	Initiate Counting	----	Lo for 1 Clock after Count	Retrigger

A3 = 1, A2 = Don't Care

DATA BYTES (A1,AØ): 11 = Counter Ø 1Ø = Counter 1 Ø1 = Counter 2

NOTE: Timer date inverted when read

Complete entire read or write operation before doing other (with respect to each Counter)

				WRITE (OUTPUT)								READ (INPUT)							
A3	A2	A1	AØ	7	6	5	4	3	2	1	Ø	7	6	5	4	3	2	1	Ø
Ø	Ø	Ø	Ø					INT.-LEVEL		(CURRENT)						CMOS TIME			
				X	X	X	X	EN---	L2	L1	LØ	X	X	5	4	3	2	1	Ø
Ø	Ø	Ø	1	BIT 7 = HIGHEST PRIORITY (INT Ø); BIT Ø = LOWEST												CMOS TIME			
				INTERRUPT MASK (1 = MASK)								X	X	11	1Ø	9	8	7	6
Ø	Ø	1	Ø			GATE Ø SELECT			CLOCK Ø SELECT							CMOS TIME			
				X	X	GC	GB	GA	CC	CB	CA	X	X	17	16	15	14	13	12
Ø	Ø	1	1			GATE 1 SELECT			CLOCK 1 SELECT							CMOS TIME			
				X	X	GC	GB	GA	CC	CB	CA	X	X	23	22	21	2Ø	19	18
Ø	1	Ø	Ø			GATE 2 SELECT			CLOCK 2 SELECT						STATUS				
				X	X	GC	GB	GA	CC	CB	CA	INT	TEN		1SEC	F1	OUT 2	OUT 1	OUT Ø
Ø	1	Ø	1			FREQUENCY 1 SELECT PROGRAM GATES													
				X	X	FC	FB	FA	P2	P1	PØ								
Ø	1	1	Ø	DISABLE TIME SET WHEN Ø	FAST CLOCK (CMOS)	RESET 212 SEC	RESET SEC	RESET MICROSECOND	RESET INT.	FREQUENCY COUNTER FS1	FSØ								

GATE SELECT

GC	GB	GA	
Ø	Ø	Ø	XGAn, where n=Ø, 1, or 2
Ø	Ø	1	Pn, where n=Ø, 1, or 2
Ø	1	Ø	1 SEC
Ø	1	1	FREQ 1
1	Ø	Ø	FREQ 2 (JUMPER SELECTED)
1	Ø	1	OUTØ
1	1	Ø	OUT1
1	1	1	OUT2

CLOCK SELECT

CC	CB	CA	
Ø	Ø	Ø	XCLn where n=Ø, 1, or 2 (see frequency counter select XCL2)
Ø	Ø	1	1 MHz
Ø	1	Ø	1 SEC
Ø	1	1	FREQ 1
1	Ø	Ø	FREQ 2 (JUMPER SELECTED)
1	Ø	1	OUTØ
1	1	Ø	OUT1
1	1	1	OUT2

FREQ 1 SELECT

FC	FB	FA	
Ø	Ø	Ø	XF1 (EXT. CLK FREQ.)
Ø	Ø	1	1 SEC CLK
Ø	1	Ø	100 MSEC
Ø	1	1	10 MSEC
1	Ø	Ø	1 MSEC
1	Ø	1	100 USEC
1	1	Ø	10 USEC
1	1	1	1 USEC

FREQUENCY COUNTER SELECT

FS1	FSØ	
Ø	Ø	FXCL2 FAST CLOCK
Ø	1	EXCL2 ÷ 1, GATED
1	Ø	EXCL2 ÷ 10, GATED
1	1	EXCL2 ÷ 100, GATED

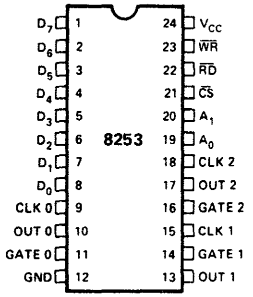
8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- **MCS—85^{TM} Compatible 8253-5**
- **3 Independent 16-Bit Counters**
- **DC to 2 MHz**
- **Programmable Counter Modes**
- **Count Binary or BCD**
- **Single +5V Supply**
- **24-Pin Dual In-Line Package**

The Intel® 8253 is a programmable counter/timer chip designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

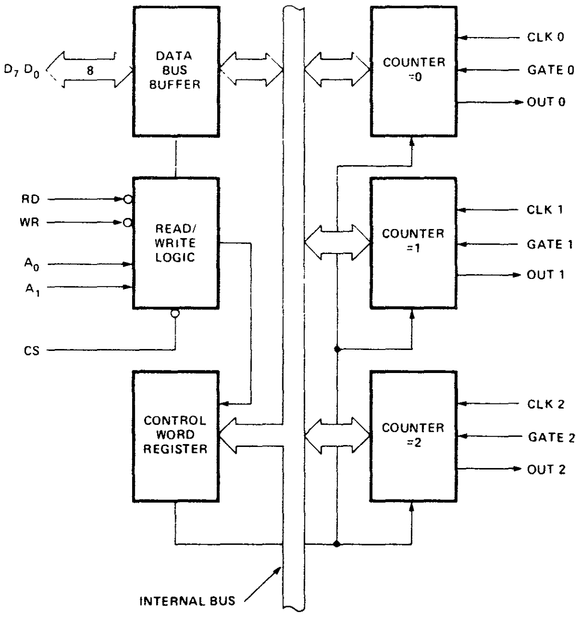
PIN CONFIGURATION



PIN NAMES

D_7 D_0	DATA BUS (8 BIT)
CLK N	COUNTER CLOCK INPUTS
GATE N	COUNTER GATE INPUTS
OUT N	COUNTER OUTPUTS
RD	READ COUNTER
WR	WRITE COMMAND OR DATA
CS	CHIP SELECT
A_0 A_1	COUNTER SELECT
V_{CC}	+5 VOLTS
GND	GROUND

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

APPENDIX B

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

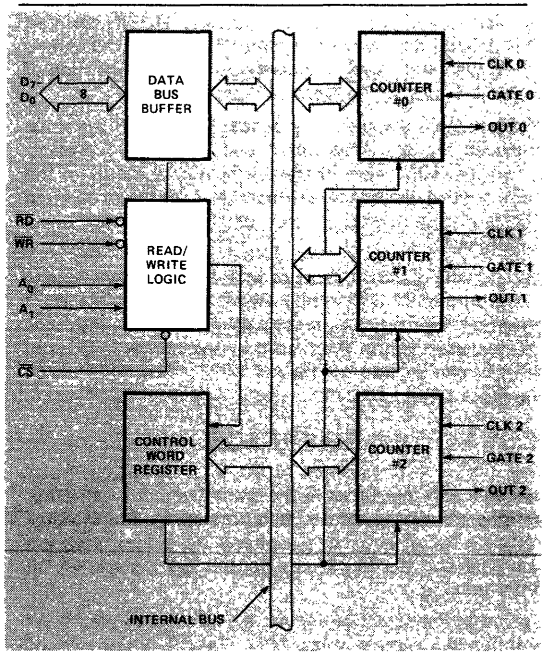


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel® 8205 for larger systems.

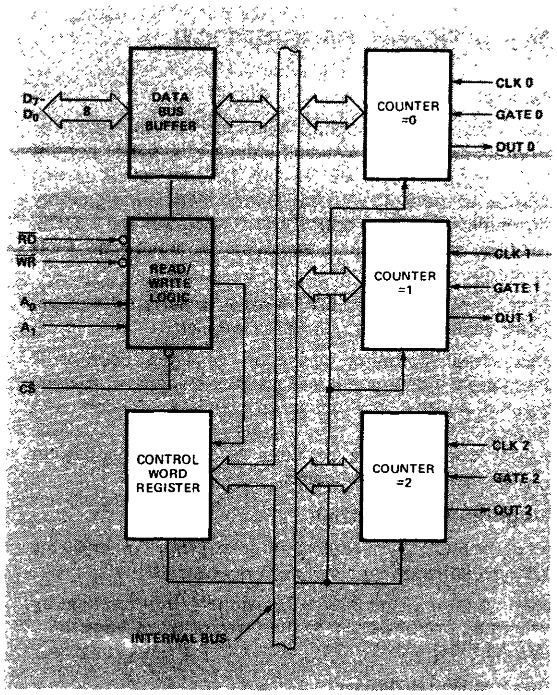


Figure 2. Block Diagram Showing Control Word Register and Counter Functions

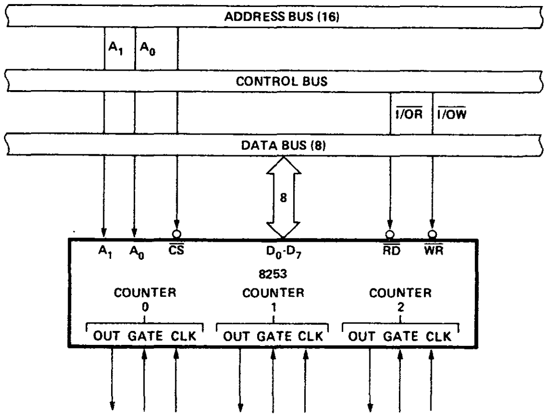


Figure 3. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. These control words program the MODE, Loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register. (A0, A1 = 11)

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL — Read/Load:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section)
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
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

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE Definition

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

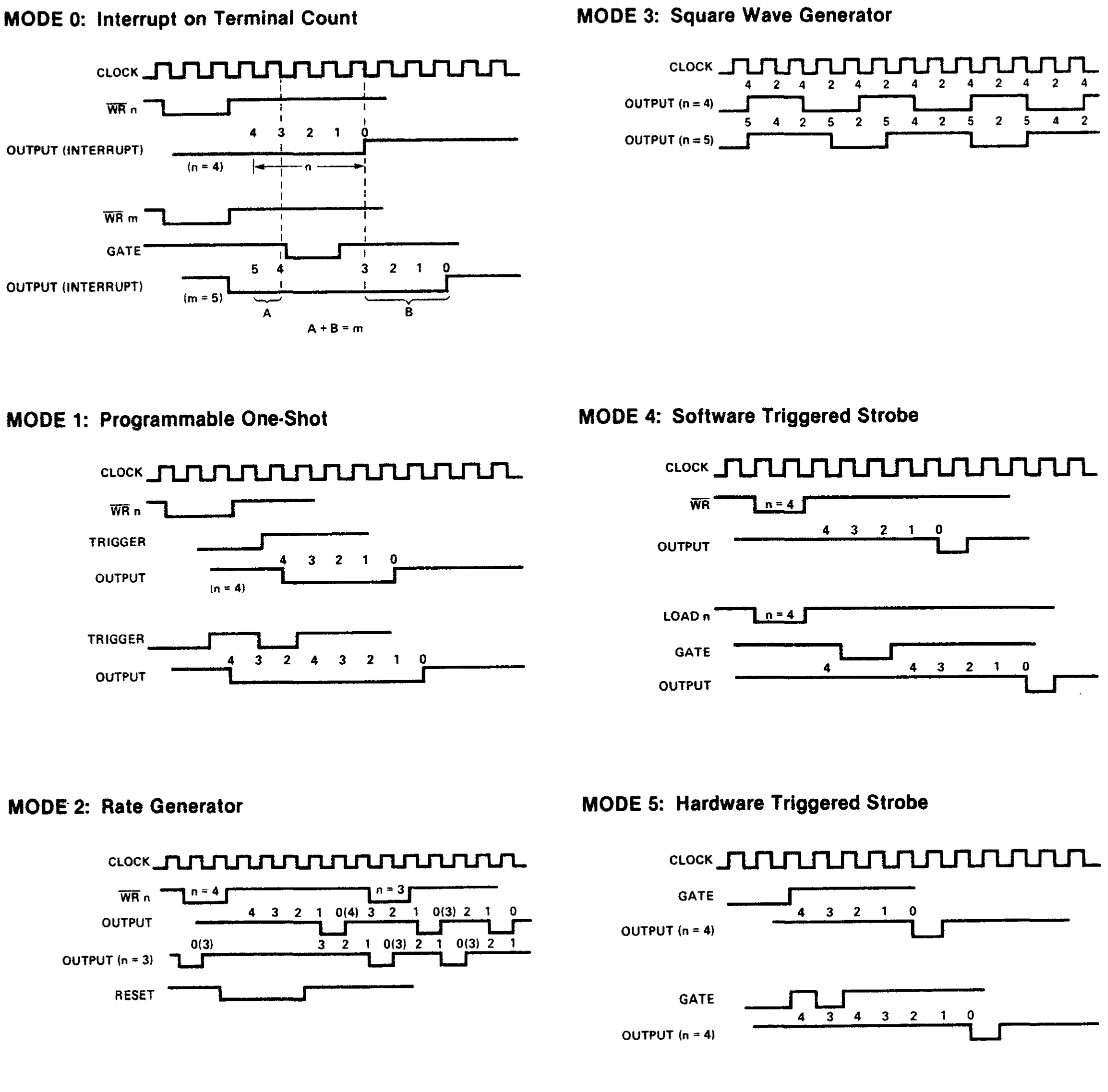
The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for (N + 1)/2 counts and low for (N – 1)/2 counts.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.


MODE 0: Interrupt on Terminal Count

MODE 1: Programmable One-Shot

MODE 2: Rate Generator

MODE 3: Square Wave Generator

MODE 4: Software Triggered Strobe

MODE 5: Hardware Triggered Strobe

Figure 5. 8253 Timing Diagrams

If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is low. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	——	Enables counting
1	——	1) Initiates counting 2) Resets output after next clock	——
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	——	Enables counting
5	——	Initiates counting	——

Figure 4. Gate Pin Operations Summary

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 6. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 7. Alternate Programming Formats

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input The contents of the counter selected will be available as follows.

first I/O Read contains the least significant byte (LSB)

second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity The programmer then issues a normal read command to the selected counter and the contents of the latched register is available

MODE Register for Latching Count

A0, A1 = 11

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 — specify counter to be latched

D5,D4 — 00 designates counter latching operation

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	−65°C to +150°C
Voltage On Any Pin With Respect to Ground	−0.5 V to +7 V
Power Dissipation	1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±5%)

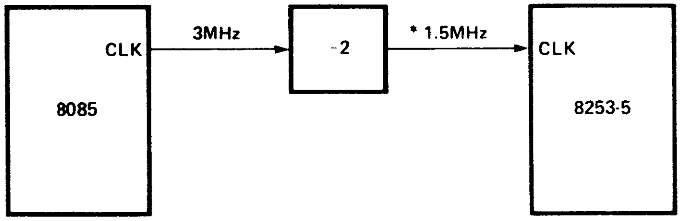
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 2
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V
I_{CC}	V_{CC} Supply Current		140	mA	

Note 1: 8253, I_{OL} = 1.6 mA; 8253-5, I_{OL} = 2.2 mA.

Note 2: 8253, I_{OH} = −150 μA; 8253-5, I_{OH} = −400 μA.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 8. MCS-85™ Clock Interface*

A.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Bus Parameters (Note 1)

Read Cycle:

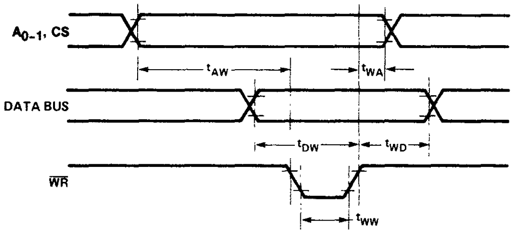
SYMBOL	PARAMETER	8253		8253-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{AR}	Address Stable Before $\overline{\text{READ}}$	50		30		ns
t_{RA}	Address Hold Time for $\overline{\text{READ}}$	5		5		ns
t_{RR}	$\overline{\text{READ}}$ Pulse Width	400		300		ns
t_{RD}	Data Delay From $\overline{\text{READ}}$[2]		300		200	ns
t_{DF}	$\overline{\text{READ}}$ to Data Floating	25	125	25	100	ns
t_{RV}	Recovery Time Between READ and Any Other Control Signal	1		1		μs

Write Cycle:

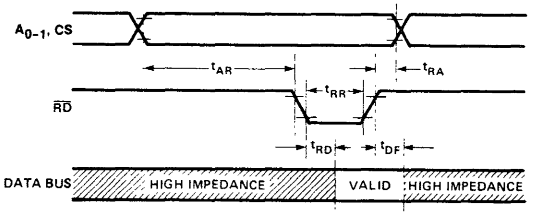
SYMBOL	PARAMETER	8253		8253-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{AW}	Address Stable Before $\overline{\text{WRITE}}$	50		30		ns
t_{WA}	Address Hold Time for $\overline{\text{WRITE}}$	30		30		ns
t_{WW}	$\overline{\text{WRITE}}$ Pulse Width	400		300		ns
t_{DW}	Data Set Up Time for $\overline{\text{WRITE}}$	300		250		ns
t_{WD}	Data Hold Time for $\overline{\text{WRITE}}$	40		30		ns
t_{RV}	Recovery Time Between $\overline{\text{WRITE}}$ and Any Other Control Signal	1		1		μs

Notes: 1. AC timings measured at V_{OH} = 2.2, V_{OL} = 0.8
2. Test Conditions: 8253, C_L = 100pF; 8253-5: C_L = 150pF.

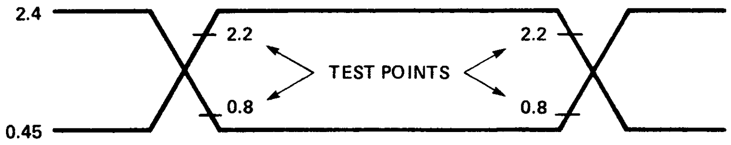
Write Timing:



Read Timing:



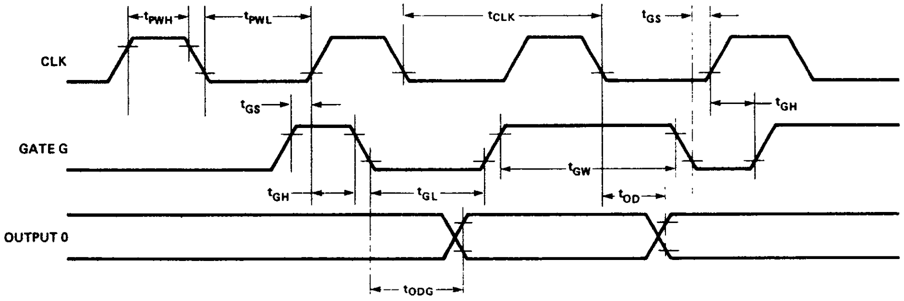
Input Waveforms for A.C. Tests:



Clock and Gate Timing:

SYMBOL	PARAMETER	8253		8253-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{CLK}	Clock Period	380	dc	380	dc	ns
t_{PWH}	High Pulse Width	230		230		ns
t_{PWL}	Low Pulse Width	150		150		ns
t_{GW}	Gate Width High	150		150		ns
t_{GL}	Gate Width Low	100		100		ns
t_{GS}	Gate Set Up Time to CLK↑	100		100		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		ns
t_{OD}	Output Delay From CLK↓[1]		400		400	ns
t_{ODG}	Output Delay From Gate↓[1]		300		300	ns

Note 1: Test Conditions: 8253: C_L = 100pF; 8253-5: C_L = 150pF.



```
4DA0                0090 *** SETIME ***
05F8                0100        ST    5370
05F8                0110 *ZERO 4 OR N BYTES
05F8                0120 ** H+L POINTS TO LOW BYTE
05F8                0130 ** C=# OF BYTES TO ZERO
05F8 0E 04          0140 ZERO4  LD    C,4H
05FA 06 00          0150 ZERON  LD    B,0H
05FC E5             0160 SETN   PUSH HL
05FD 70             0170 ZEROA  LD    (HL),B
05FE 23             0180        INC   HL
05FF 0D             0190        DEC   C
0600 C2 FD 05       0200        JP    NZ,ZEROA
0603 E1             0210        POP   HL
0604 C9             0220        RET
0605                0230 * ADD 4 OR N-WIDE
0605                0240 ** H+L=ACCUMULATED VALUE
0605                0250 ** D+E=ADDED VALUE
0605                0260 ** C=# OF BYTES PER ARGUEMENT
0605 0E 04          0270 ADD    LD    C,4H
0607 E5             0280 ADDN   PUSH HL
0608 D5             0290        PUSH DE
0609 97             0300        SUB   A
060A EB             0310 ADDA   EX    DE,HL
060B 7E             0320        LD    A,(HL)
060C 23             0330        INC   HL
060D EB             0340        EX    DE,HL
060E 8E             0350        ADC   (HL)
060F 77             0360        LD    (HL),A
0610 23             0370        INC   HL
0611 0D             0380        DEC   C
0612 C2 0A 06       0390        JP    NZ,ADDA
0615 D1             0400        POP   DE
0616 E1             0410        POP   HL
0617 C9             0420        RET
0618                0430 *MLTADD MULTIPLY BY SUCCESSIVE ADDITIONS
0618                0440 ** C=# OF BYTES IN ARGUEMENTS
0618                0450 ** B=# OF TIMES DE IS ADDED TO HL
0618 0E 04          0460 MLTADD LD    C,4H
061A D5             0470 MLTADN PUSH DE
061B C5             0480        PUSH BC
061C 11 2F 06       0490        LD    DE,TEMP
061F CD 39 06       0500        CALL MOVEN
0622 C1             0510        POP   BC
0623 05             0515        DEC   B
0624 C5             0520 MLTADA PUSH BC
0625 CD 07 06       0530        CALL ADDN
0628 C1             0540        POP   BC
0629 05             0550        DEC   B
062A C2 24 06       0560        JP    NZ,MLTADA
062D D1             0570        POP   DE
062E C9             0580        RET
062F                0590 TEMP   DS    8D
0637                0600 * MOVE FROM HL TO DE
0637                0610 ** C=# OF BYTES TO MOVE
0637 0E 04          0620 MOVE4  LD    C,4H
0639 E5             0630 MOVEN  PUSH HL
063A D5             0640        PUSH DE
063B 7E             0650 MOVEA  LD    A,(HL)
063C EB             0660        EX    DE,HL
063D 77             0670        LD    (HL),A
063E EB             0680        EX    DE,HL
```

```
063F 23          0690         INC  HL
0640 13          0700         INC  DE
0641 0D          0710         DEC  C
0642 C2 3B 06    0720         JP   NZ,MOVEA
0645 D1          0730         POP  DE
0646 E1          0740         POP  HL
0647 C9          0750         RET
0648             0760 * SHFTR SHIFT RIGHT (HL)
0648             0770 ** C=# OF BYTES IN ARGUEMENT
0648             0780 ** B=AMOUNT OF SHIFT
0648 0E 04       0790 SHFTR  LD   C,4H
064A E5          0800 SHFTRN PUSH HL
064B C5          0810        PUSH BC
064C 23          0820 SHFT1  INC  HL
064D 0D          0830        DEC  C
064E C2 4C 06    0840        JP   NZ,SHFT1
0651 2B          0850        DEC  HL
0652 C1          0860        POP  BC
0653 97          0865        SUB  A
0654 C5          0870        PUSH BC
0655 7E          0880 SHFTRB LD   A,(HL)
0656 1F          0890        RRA
0657 77          0900        LD   (HL),A
0658 2B          0910        DEC  HL
0659 0D          0920        DEC  C
065A C2 55 06    0930        JP   NZ,SHFTRB
065D C1          0940        POP  BC
065E E1          0950        POP  HL
065F 05          0960        DEC  B
0660 C2 4A 06    0970        JP   NZ,SHFTRN
0663 C9          0980        RET
0664             0990 * SUB DE FROM HL
0664             1000 ** C=# OF BYTES IN ARGUEMENT
0664 0E 04       1010 SUB    LD   C,4H
0666 E5          1020 SUBN   PUSH HL
0667 D5          1030        PUSH DE
0668 97          1040        SUB  A
0669 7E          1050 SUBA   LD   A,(HL)
066A EB          1060        EX   DE,HL
066B 9E          1070        SBC  (HL)
066C 23          1080        INC  HL
066D EB          1090        EX   DE,HL
066E 77          1100        LD   (HL),A
066F 23          1110        INC  HL
0670 0D          1120        DEC  C
0671 C2 69 06    1130        JP   NZ,SUBA
0674 D1          1140        POP  DE
0675 E1          1150        POP  HL
0676 C9          1160        RET
0677             1170 * SETIME
0677             1180 ** THIS PROGRAM SETS THE REAL TIME CLOCK
0677             1190 ** THE TIME SHOULD BE SET INTO THE TABLE 'TIME'
0677             1200 ** IN THE FOLLOWING FORMAT
0677             1210 **   BYTE1: SEC
0677             1220 **   BYTE2: TEN SEC
0677             1230 **   BYTE3: MIN
0677             1240 **   BYTE4: TEN MIN
0677             1250 **   BYTE5: HR
0677             1260 **   BYTE6: TEN HR
0677             1270 **   BYTE7: DAY OF 1/2 YEAR (BINARY)
0677             1280 **
0677             1290 ** THE PROGRAM RETURNS WITH A CONDITION CODE IN THE ACC:
0677             1300 **   0 = NO ERROR
```

```
0677                 1310 **   1 = SETTING OF CLOCK DISABLED
0677                 1320 **
0677                 1330 ***********************************************************
1340 **
0677                 1340 **
0677                 1350 * NOTE CONVERT TIME TO BINARY
0677                 1360 * ZERO T1, T2, AND T3
0677 21 EC 07        1370 SETIME LD   HL,T3
067A CD F8 05        1380        CALL ZERO4
067D 21 E8 07        1390        LD   HL,T2
0680 CD F8 05        1400        CALL ZERO4
0683 21 E4 07        1410        LD   HL,T1
0686 CD F8 05        1420        CALL ZERO4
0689                 1430 * SET T1 TO 1/2 YEAR
0689 3A E3 07        1440        LD   A,(TIME7)
068C 77              1450        LD   (HL),A
068D                 1460 * SET MULT TO 24
068D 06 18           1470        LD   B,24D
068F                 1480 * CALL MLTADD FOR T1
068F CD 18 06        1490        CALL MLTADD
0692                 1500 * SET T2 TO TEN HR
0692 3A E2 07        1510        LD   A,(TIME6)
0695 21 E8 07        1520        LD   HL,T2
0698 77              1530        LD   (HL),A
0699                 1540 * SET MULT TO TEN
0699 06 0A           1550        LD   B,10D
069B                 1560 * CALL MLTADD FOR T2
069B CD 18 06        1570        CALL MLTADD
069E                 1580 * SET T3 TO HR
069E 11 EC 07        1590        LD   DE,T3
06A1 3A E1 07        1600        LD   A,(TIME5)
06A4 EB              1610        EX   DE,HL
06A5 77              1620        LD   (HL),A
06A6 EB              1630        EX   DE,HL
06A7                 1640 * CALL ADD (T3 TO T2)
06A7 CD 05 06        1650        CALL ADD
06AA                 1660 * CALL ADD (T2 TO T1)
06AA EB              1670        EX   DE,HL
06AB 21 E4 07        1680        LD   HL,T1
06AE CD 05 06        1690        CALL ADD
06B1                 1700 *SET MULT TO 6
06B1 06 06           1710        LD   B,6H
06B3                 1720 * CALL MLTADD FOR T1
06B3 CD 18 06        1730        CALL MLTADD
06B6                 1740 * SET T2 TO TEN MIN
06B6 EB              1750        EX   DE,HL
06B7 CD F8 05        1760        CALL ZERO4
06BA 3A E0 07        1770        LD   A,(TIME4)
06BD 77              1780        LD   (HL),A
06BE                 1790 * CALL ADD (T2 TO T1)
06BE EB              1800        EX   DE,HL
06BF CD 05 06        1810        CALL ADD
06C2                 1820 * SET MULT TO 10
06C2 06 0A           1830        LD   B,10D
06C4                 1840 * CALL MLTADD FOR T1
06C4 CD 18 06        1850        CALL MLTADD
06C7                 1860 * SET T2 TO MIN
06C7 EB              1870        EX   DE,HL
06C8 CD F8 05        1880        CALL ZERO4
06CB 3A DF 07        1890        LD   A,(TIME3)
06CE 77              1900        LD   (HL),A
06CF                 1910 * CALL ADD (T2 TO T1)
06CF EB              1920        EX   DE,HL
```

```
06D0 CD 05 06      1930          CALL ADD
06D3               1940 * SET MULT TO 6
06D3 06 06         1950          LD   B,6D
06D5               1960 * CALL MLTADD FOR T1
06D5 CD 18 06      1970          CALL MLTADD
06D8               1980 * SET T2 TO TEN SEC
06D8 EB            1990          EX   DE,HL
06D9 CD F8 05      2000          CALL ZERO4
06DC 3A DE 07      2010          LD   A,(TIME2)
06DF 77            2020          LD   (HL),A
06E0               2030 * CALL ADD (T2 TO T1)
06E0 EB            2040          EX   DE,HL
06E1 CD 05 06      2050          CALL ADD
06E4               2060 * SET MULT TO 10
06E4 06 0A         2070          LD   B,10D
06E6               2080 * CALL MLTADD FOR T1
06E6 CD 18 06      2090          CALL MLTADD
06E9               2100 * SET T2 TO SEC
06E9 EB            2110          EX   DE,HL
06EA CD F8 05      2120          CALL ZERO4
06ED 3A DD 07      2130          LD   A,(TIME1)
06F0 77            2140          LD   (HL),A
06F1               2150 * CALL ADD (T2 TO T1)
06F1 EB            2160          EX   DE,HL
06F2 CD 05 06      2170          CALL ADD
06F5               2171 * SUBTRACT 1 SEC TO ALLOW SWITCHING CLOCK
06F5 EB            2172          EX   DE,HL
06F6 CD F8 05      2173          CALL ZERO4
06F9 3E 01         2174          LD   A,1
06FB 77            2175          LD   (HL),A
06FC EB            2176          EX   DE,HL
06FD CD 64 06      2177          CALL SUB
0700               2180 * MOVE LOWER 12 BITS OF T1 TO T2
0700 CD 37 06      2190          CALL MOVE4
0703 EB            2200          EX   DE,HL
0704 23            2210          INC  HL
0705 7E            2220          LD   A,(HL)
0706 E6 0F         2230          AND  0FH
0708 77            2240          LD   (HL),A
0709 23            2250          INC  HL
070A 0E 02         2260          LD   C,2D
070C CD FA 05      2270          CALL ZERON
070F EB            2280          EX   DE,HL
0710 06 0C         2290          LD   B,12D
0712               2300 * SHIFT T1 12 BITS TO RIGHT
0712 CD 48 06      2310          CALL SHFTR
0715               2320 * CALL SUB (T2 FROM T1)
0715 11 E8 07      2330          LD   DE,T2
0718 CD 64 06      2340          CALL SUB
071B               2350 * TRUNCATE T1 TO 12 BITS
071B E5            2360          PUSH HL
071C 23            2370          INC  HL
071D 7E            2380          LD   A,(HL)
071E E6 0F         2390          AND  0FH
0720 77            2400          LD   (HL),A
0721 23            2410          INC  HL
0722 0E 02         2420          LD   C,2D
0724 CD FA 05      2430          CALL ZERON
0727 E1            2440          POP  HL
0728               2450 * READ STATUS
0728 DB E4         2460          IN   A,ECH
072A               2470 * IF T-EN = 0
072A 07            2480          RLCA
```

```
072B 07             2490        RLCA
072C                2500 * THEN
072C                2510 * . RETURN WITH ERROR 1
072C 3E 01          2520        LD   A,1D
072E D0             2530        RET  NC
072F                2540 * ELSE
072F                2550 * . SET T3 TO T1
072F 11 EC 07       2560        LD   DE,T3
0732 CD 37 06       2570        CALL MOVE4
0735                2580 * . SHIFT T3 6 BITS TO RIGHT
0735 EB             2590        EX   DE,HL
0736 06 06          2600        LD   B,6D
0738 CD 48 06       2610        CALL SHFTR
073B                2620 * . SET F1 TO 100 MICROSEC AND FAST CLOCK
073B 3E 2F          2630        LD   A,2FH
073D D3 E5          2640        OUT  EDH,A
073F                2650 * . SET L AND H, RESET M
073F 3E 50          2660        LD   A,50H
0741 D3 E6          2670        OUT  EEH,A
0743                2680 * . REPEAT
0743                2690 * . . CHECK CTIME3
0743 DB E3          2700 HG     IN   A,EBH
0745 C6 02          2710        ADD  2D
0747 E6 3F          2715        AND  3FH
0749 BE             2720        CP   (HL)
074A                2730 * . UNTIL CTIME3=T3-2
074A C2 43 07       2740        JP   NZ,HG
074D                2750 * . WAIT UNTIL CTIME3=T3-1
074D DB E3          2760 HG2    IN   A,EBH
074F C6 01          2770        ADD  1D
0751 E6 3F          2775        AND  3FH
0753 BE             2780        CP   (HL)
0754 C2 4D 07       2790        JP   NZ,HG2
0757                2800 * . SET F1 TO MSEC
0757 3E 27          2810        LD   A,27H
0759 D3 E5          2820        OUT  EDH,A
075B                2830 * . REPEAT
075B                2840 * . . IF CTIME3=T3
075B DB E3          2850 HG3    IN   A,EBH
075D BE             2860        CP   (HL)
075E C2 5B 07       2870        JP   NZ,HG3
0761 EB             2880        EX   DE,HL
0762                2890 * . . THEN
0762                2900 * . . . IF CTIME2=T1 (LOW)
0762 7E             2910        LD   A,(HL)
0763 E6 3F          2915        AND  3FH
0765 47             2920        LD   B,A
0766 DB E2          2925 HF     IN   A,EAH
0768 E6 3F          2930        AND  3FH
076A B8             2935        CP   B
076B C2 66 07       2940        JP   NZ,HF
076E DB E2          2945        IN   A,EAH
0770 E6 3F          2950        AND  3FH
0772 B8             2955        CP   B
0773 C2 66 07       2960        JP   NZ,HF
0776                2970 * . . . THEN
0776                2980 * . . . . SET M
0776                2990 * . . . ENDIF
0776                3000 * . . ENDIF
0776                3010 * . UNTIL TIME REACHED
0776                3020 * . SET F1 TO 100 MICROSEC
0776 3E 40          3030        LD   A,40H
0778 D3 E6          3040        OUT  EEH,A
```

```
077A 3E 2F         3050          LD   A,2FH
077C D3 E5         3060          OUT  EDH,A
077E               3070 * . SET T3 TO T2
077E 21 E8 07      3080          LD   HL,T2
0781 CD 37 06      3090          CALL MOVE4
0784               3100 * . SHIFT T3 6 BITS RIGHT
0784 EB            3110          EX   DE,HL
0785 06 06         3120          LD   B,6D
0787 CD 48 06      3130          CALL SHFTR
078A               3140 * . REPEAT
078A               3150 * . . CHECK CTIME1
078A DB E1         3160 LG       IN   A,E9H
078C C6 02         3170          ADD  2D
078E E6 3F         3175          AND  3FH
0790 BE            3180          CP   (HL)
0791               3190 * . UNTIL CTIME1=T3-2
0791 C2 8A 07      3200          JP   NZ,LG
0794               3210 * . REPEAT
0794               3220 * . . CHECK CTIME1
0794 DB E1         3230 LG2      IN   A,E9H
0796 C6 01         3240          ADD  1D
0798 E6 3F         3245          AND  3FH
079A BE            3250          CP   (HL)
079B               3260 * . UNTIL CTIME1=T3-1
079B C2 94 07      3270          JP   NZ,LG2
079E               3280 * . SET F1 TO MSEC
079E 3E 27         3290          LD   A,27H
07A0 D3 E5         3300          OUT  EDH,A
07A2               3310 * . REPEAT
07A2               3320 * . . IF CTIME1=T3
07A2 DB E1         3330 LG3      IN   A,E9H
07A4 BE            3340          CP   (HL)
07A5 C2 A2 07      3350          JP   NZ,LG3
07A8 EB            3360          EX   DE,HL
07A9               3370 * . . THEN
07A9               3380 * . . . CHECK CTIME0
07A9 7E            3390          LD   A,(HL)
07AA E6 3F         3395          AND  3FH
07AC 47            3400          LD   B,A
07AD DB E0         3405 LF       IN   A,E8H
07AF E6 3F         3410          AND  3FH
07B1 B8            3415          CP   B
07B2 C2 AD 07      3420          JP   NZ,LF
07B5 DB E0         3425          IN   A,E8H
07B7 E6 3F         3430          AND  3FH
07B9 B8            3435          CP   B
07BA C2 AD 07      3440          JP   NZ,LF
07BD               3450 * . . ENDIF
07BD               3451 * . UNTIL TIME REACHED
07BD               3452 * . SET TO NORMAL CLOCK
07BD 3E 00         3453          LD   A,00H
07BF D3 E6         3454          OUT  EEH,A
07C1               3455 * . WAIT UNTIL CTIME0=T1+1
07C1 DB E0         3456 FL       IN   A,E8H
07C3 D6 01         3457          SUB  1D
07C5 E6 3F         3458          AND  3FH
07C7 B8            3459          CP   B
07C8 C2 C1 07      3460          JP   NZ,FL
07CB DB E0         3461          IN   A,E8H
07CD D6 01         3462          SUB  1D
07CF E6 3F         3463          AND  3FH
07D1 B8            3464          CP   B
07D2 C2 C1 07      3465          JP   NZ,FL
```

```
07D5                  3470 * . STOP TIME
07D5 3E 08            3480          LD   A,08H
07D7 D3 E6            3490          OUT  EEH,A
07D9                  3500 * . RETURN WITH NO ERROR
07D9 3E 00            3510          LD   A,00H
07DB C9               3520          RET
07DC                  3530 * ENDIF
07DC                  3540 TIME0   DC    0
  00
07DD                  3550 TIME1   DC    0
  00
07DE                  3560 TIME2   DC    0
  00
07DF                  3570 TIME3   DC    0
  00
07E0                  3580 TIME4   DC    0
  00
07E1                  3590 TIME5   DC    0
  00
07E2                  3600 TIME6   DC    0
  00
07E3                  3610 TIME7   DC    0
  00
07E4                  3620 T1      DS    4
07E8                  3630 T2      DS    4
07EC                  3640 T3      DS    4
07F0                  3650 E8H     EQU   0E0H
07F0                  3660 E9H     EQU   0E1H
07F0                  3670 EAH     EQU   0E2H
07F0                  3680 EBH     EQU   0E3H
07F0                  3690 ECH     EQU   0E4H
07F0                  3700 EDH     EQU   0E5H
07F0                  3710 EEH     EQU   0E6H
07F0 3E 04            3722 CALL    LD    A,04H
07F2 D3 E6            3724         OUT   EEH,A
07F4 CD 77 06         3730         CALL  SETIME
07F7 FE 00            3740         CP    0
07F9 C2 FD 07         3750         JP    NZ,ER
07FC C7               3760         RST   0
07FD C7               3770 ER      RST   0



READY
LTABL
ADD     0605   ADDA    060A   ADDN    0607   CALL    07F0
E8H     00E0   E9H     00E1   EAH     00E2   EBH     00E3
ECH     00E4   EDH     00E5   EEH     00E6   ER      07FD
FL      07C1   HF      0766   HG      0743   HG2     074D
HG3     075B   LF      07AD   LG      078A   LG2     0794
LG3     07A2   MLTADA  0624   MLTADD  0618   MLTADN  061A
MOVE4   0637   MOVEA   063B   MOVEN   0639   SETIME  0677
SETN    05FC   SHFT1   064C   SHFTR   0648   SHFTRB  0655
SHFTRN  064A   SUB     0664   SUBA    0669   SUBN    0666
T1      07E4   T2      07E8   T3      07EC   TEMP    062F
TIME0   07DC   TIME1   07DD   TIME2   07DE   TIME3   07DF
TIME4   07E0   TIME5   07E1   TIME6   07E2   TIME7   07E3
ZERO4   05F8   ZEROA   05FD   ZERON   05FA
FILE  3000  4D7F
READY
```

APPENDIX D

```
10 A=14*16
20 I0=INP(A)
30 I1=INP(A+1)
40 I2=INP(A+2)
50 I3=INP(A+3)
60 I9=INP(A)
70 IF I0=I9 THEN 80 ELSE 20
80 T0=((64^3)*I3)+((64^2)*I2)+(64*I1)+I0
90 T1=INT(T0/(3600*24))
100 T9=T0-(T1*3600*24)
110 T2=INT(T9/3600)
120 T8=T9-(T2*3600)
130 T3=INT(T8/60)
140 T4=T8-(T3*60)
150 PRINT "DAY OF YEAR IS ";%3I;T1
160 # "HOUR IS ";%2I;T2
170 # "MINUTE IS ";%2I;T3
180 # "SECOND IS ";%2I;T4
190 STOP
200 OUT A+6,128
210 END
```

APPENDIX E

```
100 REM INITIALIZE THE COUNTER CONFIGURATION
110 A=14*16
120 REM INPUT OF CTRS=1 MHZ
130 REM GATE OF CTRS=PROG
140 OUT A+4,(1*8)+1
150 OUT A+3,(1*8)+1
160 OUT A+2,(1*8)+1
170 REM TURN GATES OFF
180 OUT A+5,0
190 REM SET CTRS 0-2 TO MODE 2
200 OUT A+8+0,(0*64)+(3*16)+(2*2)+0
210 OUT A+8+0,(1*64)+(3*16)+(2*2)+0
220 OUT A+8+0,(2*64)+(3*16)+(2*2)+0
230 REM LOAD CTRS WITH DATA
240 D=255
250 OUT A+8+3,D
260 OUT A+8+3,D
270 OUT A+8+2,D
280 OUT A+8+2,D
290 OUT A+8+1,D
300 OUT A+8+1,D
310 REM INPUT OF CTR 2=CTR 1
320 OUT A+4,(1*8)+6
330 REM INPUT OF CTR 1=CTR 0
340 OUT A+3,(1*8)+5
350 REM WAIT FOR INPUT TO START COUNTING
360 INPUT I
370 REM START COUNTERS
380 OUT A+5,7
390 REM WAIT FOR INPUT TO STOP COUNTING
400 INPUT I
410 REM STOP COUNTERS
420 OUT A+5,0
430 REM READ COUNTERS
440 L0=INP(A+8+3)
450 H0=INP(A+8+3)
460 L1=INP(A+8+2)
470 H1=INP(A+8+2)
480 L2=INP(A+8+1)
490 H2=INP(A+8+1)
500 REM DISPLAY TIME
510 T2=L2+(256*H2)
520 T1=L1+(256*H1)
530 T0=L0+(256*H0)
540 T=T0+(((256^2)-1)*T1)+((((256^2)-1)^2)*T2)
550 PRINT T/1000000,"SECONDS"
560 END
```

APPENDIX F

```
10 A=14*16
20 OUT A+4,8
30 OUT A+5,4
40 OUT A+6,128+3
50 I0=INP(A+4)
60 GOSUB 1000
70 IF I0=1 THEN 50
80 I0=INP(A+4)
90 GOSUB 1000
100 IF I0=0 THEN 80
105 OUT A+8,128+48
110 M=2
115 OUT A+5,0
120 OUT A+8+1,255
122 OUT A+8+1,255
126 IF M=0 THEN 136
128 OUT A+6,1
130 OUT A+5,4
132 OUT A+5,0
134 OUT A+6,M+1
135 GOTO 145
136 OUT A+6,2
138 OUT A+5,4
140 OUT A+5,0
142 OUT A+6,1
145 OUT (A+5),4
150 O=0
155 GOSUB 1200
160 I0=INP(A+4)
170 GOSUB 1000
180 IF I0=1 THEN 160
190 I0=INP(A+4)
200 GOSUB 1000
210 IF I0=0 THEN 190
220 IF O=0 THEN 250
230 PRINT "OVERFLOW"
235 OUT A+6,3
240 M=2
242 F=0
244 GOTO 115
250 I1=INP(A+8+1)
260 I2=INP(A+8+1)
270 IF I1=0 THEN 280 ELSE 400
280 IF I2=0 THEN 290 ELSE 400
290 IF M=0 THEN M=3
300 M=M-1
310 OUT (A+6),M+1
320 F=0
330 GOTO 115
400 C0=I1+(256*I2)
410 F=C0*(10^M)
420 IF I2>12 THEN 442
430 IF M=0 THEN 445
440 M=M-1
441 GOTO 445
442 IF I2<50 THEN 445
443 IF M=2 THEN 445
444 M=M+1
445 OUT A+5,0
446 OUT A+6,M+1
447 OUT A+5,4
```

```
450 GOTO 115
1000 FOR N=7 TO 5 STEP -1
1010 IF (2^N)<(IO+1) THEN IO=IO-(2^N)
1020 NEXT N
1030 I9=IO
1040 IF IO<16 THEN IO=0 ELSE IO=1
1050 FOR N=4 TO 3 STEP -1
1060 IF (2^N)<(I9+1) THEN I9=I9-(2^N)
1070 NEXT N
1080 IF I9<4 THEN O=1
1090 RETURN
1200 IF F>=1000000 THEN 1300
1210 IF F>=1000 THEN 1280
1220 PRINT %8I,F,"HZ"
1230 RETURN
1280 PRINT %8F4,F/1000,"KHZ"
1290 GOTO 1230
1300 PRINT %8F4,F/1000000,"MHZ"
1310 GOTO 1230
3000 OUT 2,24
3010 OUT 1,128+31
3020 OUT 1,255-32
3030 STOP
```

```
554F                0100 *** I/RTC MUSIC ROUTINES VERSION 1.0
5900                0110            ORG   5900H
0900                0115            ST    900H
0900 31 EF 0F       0120 INIT       LD    SP,STACK
0903 3E FF          0130            LD    A,0FFH MASK INTERRUPTS
0905 D3 E1          0140            OUT   0E1H,A
0907 3E 84          0150            LD    A,84H RESET INT ACKNOWLEDGE
0909 D3 E6          0160            OUT   0E6H,A
090B 3E 00          0170            LD    A,0
090D D3 E6          0180            OUT   0E6H,A
090F 3E 08          0190            LD    A,08H SET INT TO BASE LEVEL
0911 D3 E0          0200            OUT   0E0H,A
0913 3E DF          0210            LD    A,0DFH SET MASK TO ALLOW LEVEL 2
0915 32 2A 0D       0215            LD    (CURMSK),A
0918 D3 E1          0220            OUT   0E1H,A
091A 3E 0F          0230            LD    A,0FH HIGH BYTE OF INTERRUPT ADDRESS
091C ED 47          0240            LD    I,A SET INT MODE 2
091E ED 5E          0250            IM    2
0920 C3 23 09       0260            JP    BASEC INTERRUPTS NOW INITIALIZED
0923                0270 STACK      EQU   0FEFH
0923 3E F8          0280 BASEC      LD    A,0F8H INITIALIZE PROGRAMMABLE GATES OFF
0925 32 1C 0D       0290            LD    (PGATES),A
0928 D3 E5          0300            OUT   0E5H,A
092A 3E 09          0310            LD    A,09H INIT CLOCK AND GATE SOURCES
092C D3 E2          0320            OUT   0E2H,A
092E D3 E3          0330            OUT   0E3H,A
0930 D3 E4          0340            OUT   0E4H,A
0932 3E 36          0350            LD    A,36H INIT COUNTERS
0934 D3 E8          0360            OUT   0E8H,A
0936 3E 76          0370            LD    A,76H
0938 D3 E8          0380            OUT   0E8H,A
093A 3E B6          0390            LD    A,0B6H
093C D3 E8          0400            OUT   0E8H,A
093E DD 21 00 0D    0410            LD    IX,BASE SET UP INDEX TO ALL BYTES
0942 2A 25 0D       0420            LD    HL,(PIECE) INIT LOCATION OF MUSIC COMMANDS
0945 ED 4B 27 0D    0430            LD    BC,(FLAGS) INIT FLAGS
0949 11 00 08       0440            LD    DE,FREQLU INIT NOTE-TO-COUNT XLATE TABLE PTI
094C FB             0450            EI
094D CD D2 0B       0460 LOOP       CALL  INKBD LOOK FOR STOP (SPACE BAR)
0950 FE A0          0470            CP    0A0H
0952 CA C2 09       0480            JP    Z,STOP
0955 7E             0490            LD    A,(HL) GET COMMAND
0956 E6 03          0500            AND   03H
0958 DD 21 FF 0C    0510            LD    IX,BASE-1
095C DD 23          0520 LP1        INC   IX SET INDEX TO CMD'S TIMER BITS
095E 3D             0530            DEC   A
095F C2 5C 09       0540            JP    NZ,LP1
0962 DD CB 0D 46    0550            BIT   0,(IX+RDY0-BASE) WAIT TILL PREV. NOTE PASSE
0966 C2 4D 09       0560            JP    NZ,LOOP
0969 7E             0570            LD    A,(HL) RDY FOR THIS NEW CMD
096A CB 77          0580            BIT   6,A SET UP GATE (NOTE OR REST)
096C C2 76 09       0590            JP    NZ,B1
096F DD CB 04 8E    0600            RES   1,(IX+MNXT0-BASE)
0973 C3 7A 09       0610            JP    B2
0976 DD CB 04 CE    0620 B1         SET   1,(IX+MNXT0-BASE)
097A CB 6F          0630 B2         BIT   5,A
097C C2 86 09       0640            JP    NZ,B3
097F DD CB 04 86    0650            RES   0,(IX+MNXT0-BASE)
0983 C3 8A 09       0660            JP    B4
0986 DD CB 04 C6    0670 B3         SET   0,(IX+MNXT0-BASE)
098A CB 0F          0680 B4         RRC   A
```

```
098C CB 0F       0690        RRC  A GET LENGTH OR TIME OF NOTE (REST)
098E E6 07       0700        AND  07H
0990 FE 00       0710        CP   0 IF LENGTH = 0, DONE
0992 CA C2 09    0720        JP   Z,STOP
0995 47          0730        LD   B,A
0996 3E 80       0740        LD   A,80H
0998 CB 07       0750 LP2    RLC  A CONVERT LENGTH CODE TO A REAL VALUE
099A 05          0760        DEC  B
099B C2 98 09    0770        JP   NZ,LP2
099E DD 77 01    0780        LD   (IX+BTLEN0-BASE),A
09A1 DD CB 04 4E 0790        BIT  1,(IX+MNXT0-BASE)
09A5 CA B6 09    0800        JP   Z,B5 IF NOT A REST, THEN:
09A8 23          0810        INC  HL
09A9 7E          0820        LD   A,(HL) GET NOTE NUMBER
09AA CB 07       0830        RLC  A
09AC 5F          0840        LD   E,A TRANSLATE TO COUNTER VALUES
09AD 1A          0850        LD   A,(DE) AND GET READY
09AE DD 77 13    0860        LD   (IX+FNXTH0-BASE),A
09B1 13          0870        INC  DE
09B2 1A          0880        LD   A,(DE)
09B3 DD 77 10    0890        LD   (IX+FNXTL0-BASE),A
09B6 23          0900 B5     INC  HL
07B7 DD CB 04 E6 0910        SET  4,(IX+MNXT0-BASE) FLAG NEW NOTE
09BB DD CB 0D C6 0920        SET  0,(IX+RDY0-BASE) SET READY
07BF C3 4D 09    0930        JP   LOOP
09C2 C7          0940 STOP   RST  0
09C3             0945 ***************************************
09C3 F5          0950 INT2   PUSH AF TIMER INTERRUPT ROUTINE (1 MILLISECOND)
09C4 3A 29 0D    0960        LD   A,(CURLEV) SAVE INTERRUPTED LEVEL
09C7 F5          0970        PUSH AF
09C8 3E 02       0975        LD   A,02H TELL CONTROLLER THIS LEVEL
09CA D3 E0       0980        OUT  0E0H,A
07CC 32 29 0D    0990        LD   (CURLEV),A
09CF FB          1000        EI
09D0 3A 23 0D    1010        LD   A,(MSCNT) INCREMENT MILLISECOND COUNTER
09D3 3C          1020        INC  A
09D4 32 23 0D    1030        LD   (MSCNT),A
09D7 C5          1040        PUSH BC
09D8 47          1050        LD   B,A
09D9 3A 1F 0D    1060        LD   A,(BKCNT)
09DC B8          1070        CP   B
09DD C2 F0 09    1080        JP   NZ,I2A
09E0 3A 20 0D    1090        LD   A,(IFLAG)
09E3 E6 08       1100        AND  BKACTN
09E5 CA F0 09    1110        JP   Z,I2A
09E8 3A 20 0D    1120        LD   A,(IFLAG)
09EB F6 04       1130        OR   BREAKI
09ED 32 20 0D    1140        LD   (IFLAG),A
09F0 3A 1D 0D    1150 I2A    LD   A,(BTTIME) DECREMENT COUNTER OF MSEC/BEAT
09F3 3D          1160        DEC  A
09F4 32 1D 0D    1170        LD   (BTTIME),A
09F7 C2 1F 0A    1180        JP   NZ,I2B IF ONE BEAT'S TIME (1/64TH NOTE)
09FA 3A 24 0D    1190        LD   A,(BEAT)
09FD 32 1D 0D    1200        LD   (BTTIME),A
0A00 3A 22 0D    1210        LD   A,(BTCNT) THEN INCREMENT BEAT COUNTER
0A03 3C          1220        INC  A
0A04 32 22 0D    1230        LD   (BTCNT),A
0A07 47          1240        LD   B,A
0A08 3A 21 0D    1250        LD   A,(INTCNT) INTCNT=BEAT COUNT OF NEXT NOTE
0A0B B8          1260        CP   B
0A0C C2 1F 0A    1270        JP   NZ,I2B
0A0F 3A 20 0D    1280        LD   A,(IFLAG)
0A12 E6 10       1290        AND  INTACN
```

```
0A14 CA 1F 0A      1300         JP    Z,I2B
0A17 3A 20 0D      1310         LD    A,(IFLAG) SET UP FOR BEAT INTERRUPT
0A1A F6 02         1320         OR    OBEATI
0A1C 32 20 0D      1330         LD    (IFLAG),A
0A1F 3A 20 0D      1340 I2B     LD    A,(IFLAG)
0A22 E6 06         1350         AND   BNB IF INTERRUPT NEEDED
0A24 CA 2B 0A      1360         JP    Z,I2C
0A27 DD CB 2A 9E   1370         RES   3,(IX+CURMSK-BASE)
0A2B C1            1410 I2C     POP   BC
0A2C F1            1420         POP   AF RESTORE PREVIOUS INTERRUPT LEVEL
0A2D F3            1430         DI    AND RESTORE MASK
0A2E 32 29 0D      1440         LD    (CURLEV),A
0A31 3A 2A 0D      1451         LD    A,(CURMSK)
0A34 CB EF         1452         SET   5,A
0A36 D3 E1         1453         OUT   0E1H,A
0A38 CB AF         1454         RES   5,A
0A3A D3 E1         1455         OUT   0E1H,A
0A3C 3A 29 0D      1457         LD    A,(CURLEV)
0A3F D3 E0         1458         OUT   0E0H,A
0A41 F1            1460         POP   AF
0A42 FB            1470         EI
0A43 C9            1480         RET   RETURN FROM INTERRUPT
0A44               1485 *******************************************
0A44 F5            1490 INT4    PUSH AF NOTE INTERRUPT ROUTINE
0A45 3A 29 0D      1500         LD    A,(CURLEV)
0A48 F5            1510         PUSH AF
0A49 3E 04         1520         LD    A,04H
0A4B D3 E0         1530         OUT   0E0H,A
0A4D 32 29 0D      1540         LD    (CURLEV),A
0A50 C5            1550         PUSH BC
0A51 DD CB 2A DE   1552         SET   3,(IX+CURMSK-BASE) TURN OFF INT
0A55 3A 2A 0D      1554         LD    A,(CURMSK)
0A58 D3 E1         1556         OUT   0E1H,A
0A5A 3A 23 0D      1560         LD    A,(MSCNT) SET UP CURRENT VALUES
0A5D 47            1570         LD    B,A BEFORE UNMASKING INTERRUPTS
0A5E 3A 22 0D      1580         LD    A,(BTCNT)
0A61 4F            1590         LD    C,A
0A62 D5            1600         PUSH DE
0A63 3A 20 0D      1610         LD    A,(IFLAG)
0A66 57            1620         LD    D,A
0A67 E6 F9         1630         AND   BNBN RESET INTERRUPT FLAGS
0A69 32 20 0D      1640         LD    (IFLAG),A
0A6C FB            1650         EI
0A6D E5            1660         PUSH HL
0A6E 7A            1670         LD    A,D
0A6F E6 04         1680         AND   BRKIN
0A71 CA A7 0A      1690         JP    Z,I4A
0A74 DD E5         1700 BREAK   PUSH IX
0A76 DD 21 00 0D   1710         LD    IX,BASE
0A7A 1E 03         1720         LD    E,3
0A7C CB 52         1730 BRK0    BIT   2,D
0A7E CA 9F 0A      1740         JP    Z,BRK1
0A81 DD CB 07 56   1750         BIT   2,(IX+MODE0-BASE)
0A85 CA 9F 0A      1760         JP    Z,BRK1
0A88 DD CB 07 8E   1780         RES   1,(IX+MODE0-BASE)
0A8C DD CB 07 DE   1790         SET   3,(IX+MODE0-BASE)
0A90 DD 34 0A      1800         INC   (IX+BTCNT0-BASE)
0A93 DD CB 07 96   1810         RES   2,(IX+MODE0-BASE)
0A97 3A 20 0D      1812         LD    A,(IFLAG)
0A9A CB 9F         1814         RES   3,A
0A9C 32 20 0D      1816         LD    (IFLAG),A
0A9F DD 23         1820 BRK1    INC   IX
0AA1 1D            1830         DEC   E
```

```
0AA2 C2 7C 0A      1840          JP    NZ,BRK0
0AA5 DD E1         1850          POP   IX
0AA7 CB 4A         1860 I4A      BIT   1,D
0AA9 CA 4D 0B      1870          JP    Z,I4B
0AAC DD E5         1880 BEET     PUSH  IX
0AAE DD 21 00 0D   1890          LD    IX,BASE SET UP INDEXING
0AB2 1E 03         1900          LD    E,3
0AB4 DD 7E 0A      1910 BEAT0    LD    A,(IX+BTCNT0-BASE)
0AB7 B9            1920          CP    C BEAT READY FOR CURRENT COUNTER?
0AB8 C2 45 0B      1930          JP    NZ,BEAT1
0ABB DD CB 07 4E   1940          BIT   1,(IX+MODE0-BASE)
0ABF CA E2 0A      1950          JP    Z,BEAT3
0AC2 DD CB 07 46   1960          BIT   0,(IX+MODE0-BASE)
0AC6 C2 E2 0A      1970          JP    NZ,BEAT3
0AC9 78            1980          LD    A,B
0ACA C6 10         1990          ADD   20
0ACC DD 77 1F      2000          LD    (IX+BKCNT-BASE),A
0ACF DD CB 07 D6   2010          SET   2,(IX+MODE0-BASE)
0AD3 DD CB 07 9E   2020          RES   3,(IX+MODE0-BASE)
0AD7 3A 20 0D      2022          LD    A,(IFLAG)
0ADA CB DF         2024          SET   3,A
0ADC 32 20 0D      2026          LD    (IFLAG),A
0ADF C3 45 0B      2030          JP    BEAT1
0AE2 DD CB 0D 46   2040 BEAT3    BIT   0,(IX+RDY0-BASE) YES. NOTE SET UP?
0AE6 CA 3D 0B      2050          JP    Z,BEAT2
0AE9 DD CB 0D 86   2060          RES   0,(IX+RDY0-BASE) YES. TAKE NOTE
0AED DD 7E 10      2070          LD    A,(IX+FNXTL0-BASE)
0AF0 DD 77 16      2080          LD    (IX+FREQL0-BASE),A
0AF3 DD 7E 13      2090          LD    A,(IX+FNXTH0-BASE)
0AF6 DD 77 19      2100          LD    (IX+FREQH0-BASE),A
0AF9 DD 7E 04      2110          LD    A,(IX+MNXT0-BASE)
0AFC DD 77 07      2120          LD    (IX+MODE0-BASE),A
0AFF DD 7E 01      2130          LD    A,(IX+BTLEN0-BASE)
0B02 81            2140          ADD   C CALCULATE COUNT WHEN NOTE SHOULD BE DONE
0B03 DD 77 0A      2150          LD    (IX+BTCNT0-BASE),A
0B06 DD CB 07 DE   2160          SET   3,(IX+MODE0-BASE)
0B0A DD CB 07 4E   2170          BIT   1,(IX+MODE0-BASE)
0B0E CA 3A 0B      2180          JP    Z,BEAT4
0B11 DD CB 07 46   2190          BIT   0,(IX+MODE0-BASE)
0B15 C2 3A 0B      2200          JP    NZ,BEAT4
0B18 DD 7E 0A      2210          LD    A,(IX+BTCNT0-BASE)
0B1B D6 01         2220          SUB   1
0B1D DD 77 0A      2225          LD    (IX+BTCNT0-BASE),A
0B20 B9            2230          CP    C
0B21 C2 3A 0B      2240          JP    NZ,BEAT4
0B24 78            2250          LD    A,B
0B25 C6 10         2260          ADD   20
0B27 32 1F 0D      2270          LD    (BKCNT),A
0B2A DD CB 07 9E   2280          RES   3,(IX+MODE0-BASE)
0B2E DD CB 07 D6   2290          SET   2,(IX+MODE0-BASE)
0B32 3A 20 0D      2292          LD    A,(IFLAG)
0B35 CB DF         2294          SET   3,A
0B37 32 20 0D      2296          LD    (IFLAG),A
0B3A C3 45 0B      2300 BEAT4    JP    BEAT1
0B3D DD 7E 07      2310 BEAT2    LD    A,(IX+MODE0-BASE)
0B40 E6 F5         2320          AND   0F5H NOTE NOT READY. TURN OFF
0B42 DD 77 07      2330          LD    (IX+MODE0-BASE),A
0B45 DD 23         2340 BEAT1    INC   IX REPEAT UNTIL ALL COUNTERS SERVICED
0B47 1D            2350          DEC   E
0B48 C2 B4 0A      2360          JP    NZ,BEAT0
0B4B DD E1         2370          POP   IX
0B4D DD E5         2380 I4B      PUSH  IX
0B4F DD 21 00 0D   2390          LD    IX,BASE
```

```
0B53 1E 03         2400          LD   E,3
0B55 0E E9         2410          LD   C,0E9H
0B57 DD CB 07 66   2420 OPT0     BIT  4,(IX+MODE0-BASE) IF NEW NOTE,
0B5B CA 6C 0B      2430          JP   Z,OPT1
0B5E DD CB 07 A6   2440          RES  4,(IX+MODE0-BASE)
0B62 DD 46 16      2450          LD   B,(IX+FREQLO-BASE)
0B65 ED 41         2460          OUT  (C),B THEN OUTPUT NEW VALUE
0B67 DD 46 19      2462          LD   B,(IX+FREQH0-BASE)
0B6A ED 41         2464          OUT  (C),B
0B6C 53            2470 OPT1     LD   D,E
0B6D 3A 1C 0D      2480          LD   A,(PGATES)
0B70 CB 0F         2490 OPT4     RRC  A OUTPUT REST OR NOTE
0B72 15            2500          DEC  D
0B73 C2 70 0B      2510          JP   NZ,OPT4
0B76 DD CB 07 4E   2520          BIT  1,(IX+MODE0-BASE)
0B7A C2 82 0B      2530          JP   NZ,OPT2
0B7D CB BF         2540          RES  7,A
0B7F C3 84 0B      2550          JP   OPT3
0B82 CB FF         2560 OPT2     SET  7,A
0B84 53            2570 OPT3     LD   D,E
0B85 CB 07         2580 OPT5     RLC  A
0B87 15            2590          DEC  D
0B88 C2 85 0B      2600          JP   NZ,OPT5
0B8B 32 1C 0D      2605          LD   (PGATES),A
0B8E D3 E5         2610          OUT  0E5H,A
0B90 DD 23         2620          INC  IX REPEAT UNTIL ALL 3 COUNTERS SERVICED
0B92 0C            2630          INC  C
0B93 1D            2640          DEC  E
0B94 C2 57 0B      2650          JP   NZ,OPT0
0B97 DD E1         2660          POP  IX
0B99 0E 7F         2670          LD   C,7FH
0B9B 3A 21 0D      2680          LD   A,(INTCNT)
0B9E 47            2690          LD   B,A
0B9F DD E5         2700 I4C      PUSH IX
0BA1 DD 21 00 0D   2710          LD   IX,BASE
0BA5 1E 03         2720          LD   E,3
0BA7 DD CB 07 5E   2730 I4E      BIT  3,(IX+MODE0-BASE) IF NOTE ACTIVE,
0BAB CA BD 0B      2740          JP   Z,I4D
0BAE DD 7E 0A      2750          LD   A,(IX+BTCNT0-BASE) THEN SEE IF NEXT EVENT T
0BB1 90            2760          SUB  B
0BB2 B9            2770          CP   C
0BB3 F2 BD 0B      2780          JP   P,I4D
0BB6 4F            2790          LD   C,A
0BB7 DD 7E 0A      2800          LD   A,(IX+BTCNT0-BASE)
0BBA 32 21 0D      2810          LD   (INTCNT),A IF CLOSER, MAKE THIS COUNTER NEX
0BBD DD 23         2820 I4D      INC  IX REPEAT UNTIL CLOSEST IS MADE NEXT
0BBF 1D            2830          DEC  E
0BC0 C2 A7 0B      2840          JP   NZ,I4E
0BC3 DD E1         2850          POP  IX
0BC5 E1            2860          POP  HL
0BC6 D1            2870          POP  DE
0BC7 C1            2880          POP  BC
0BC8 F1            2890          POP  AF
0BC9 F3            2900          DI   RETURN FROM INTERRUPT
0BCA 32 29 0D      2910          LD   (CURLEV),A
0BCD D3 E0         2920          OUT  0E0H,A
0BCF F1            2930          POP  AF
0BD0 FB            2940          EI
0BD1 C9            2950          RET
0BD2               2955 *******************************************
0BD2 ED 78         2960 INKBD    IN   A,(00H) KEYBOARD ROUTINE TO GET CHARACTERS
0BD4 FE 00         2970          CP   0
0BD6 F0            2980          RET  P
```

```
0BD7 ED 78         2990 INK1   IN   A,(00H)
0BD9 FE 00         3000        CP   0
0BDB FA D7 0B      3010        JP   M,INK1
0BDE F6 80         3020        OR   80H
0BE0 C9            3030        RET
5E00               3031        ORG  5E00H JUMP HERE TO TURN INTERRUPTS OFF
0E00               3032        ST   0E00H
0E00 F3            3033        DI
0E01 C7            3034        RST  0
5D00               3035        ORG  5D00H
0D00               3036        ST   0D00H
0D00               3040 BASE   DS   1
0D01               3050 BTLEN0 DS   3
0D04               3060 MNXT0  DB   0,0,0
  00   00   00
0D07               3070 MODE0  DB   0,0,0
  00   00   00
0D0A               3080 BTCNT0 DB   0,0,0
  00   00   00
0D0D               3090 RDY0   DB   0,0,0
  00   00   00
0D10               3100 FNXTL0 DB   0,0,0
  00   00   00
0D13               3110 FNXTH0 DB   0,0,0
  00   00   00
0D16               3120 FREQL0 DB   0,0,0
  00   00   00
0D19               3130 FREQH0 DB   0,0,0
  00   00   00
0D1C               3140 PGATES DB   0F8H
  F8
0D1D               3150 BTTIME DB   20H
  20
0D1E               3160 BTNXT  DB   18H
  18
0D1F               3170 BKCNT  DB   18H
  18
0D20               3180 IFLAG  DB   10H
  10
0D21               3190 INTCNT DB   0
  00
0D22               3200 BTCNT  DS   1
0D23               3210 MSCNT  DB   0
  00
0D24               3220 BEAT   DB   20H
  20
0D25               3230 PIECE  DW   1000H
  00 10
0D27               3240 FLAGS  DW   0000H
  00 00
0D29               3250 FREQLU EQU  0800H
0D29               3260 CURLEV DB   08H
  08
0D2A               3265 CURMSK DB   0FFH
  FF
0D2B               3270 BKACTN EQU  08H
0D2B               3280 BREAKI EQU  04H
0D2B               3290 BNB    EQU  06H
0D2B               3300 BRKIN  EQU  04H
0D2B               3310 BEATI  EQU  01H
0D2B               3320 INTACN EQU  10H
0D2B               3330 OBEATI EQU  02H
0D2B               3340 BNBN   EQU  0F9H
```

```
5800                    3350        ORG   5800H
0800                    3355        ST    800H
0800                    3360        DB    77H,72H,70H,0BEH,6AH,6AH,64H,71H
  77    72    70    BE    6A
  6A    64    71
0808                    3370        DB    5EH,0CEH,59H,7CH,54H,76H,4FH,0B8H
  5E    CE    59    7C    54
  76    4F    B8
0810                    3380        DB    4BH,3FH,47H,06H,43H,09H,3FH,46H
  4B    3F    47    06    43
  09    3F    46
0818                    3390        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
0820                    3400        DB    3BH,0B9H,38H,5FH,35H,35H,32H,38H
  3B    B9    38    5F    35
  35    32    38
0828                    3410        DB    2FH,67H,2CH,0BEH,2AH,3BH,27H,0DCH
  2F    67    2C    BE    2A
  3B    27    DC
0830                    3420        DB    25H,9FH,23H,83H,21H,85H,1FH,0A3H
  25    9F    23    83    21
  85    1F    A3
0838                    3430        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
0840                    3440        DB    1DH,0DDH,1CH,2FH,1AH,9AH,19H,1CH
  1D    DD    1C    2F    1A
  9A    19    1C
0848                    3450        DB    17H,0B3H,16H,5FH,15H,1DH,13H,0EEH
  17    B3    16    5F    15
  1D    13    EE
0850                    3460        DB    12H,0D0H,11H,0C1H,10H,0C2H,0FH,0D2H
  12    D0    11    C1    10
  C2    0F    D2
0858                    3470        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
0860                    3480        DB    0EH,0EEH,0EH,18H,0DH,4DH,0CH,8EH
  0E    EE    0E    18    0D
  4D    0C    8E
0868                    3490        DB    0BH,0DAH,0BH,2FH,0AH,8FH,09H,0F7H
  0B    DA    0B    2F    0A
  8F    09    F7
0870                    3500        DB    09H,68H,08H,0E1H,08H,61H,07H,0E9H
  09    68    08    E1    08
  61    07    E9
0878                    3510        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
0880                    3520        DB    07H,77H,07H,0CH,06H,0A7H,06H,47H
  07    77    07    0C    06
  A7    06    47
0888                    3530        DB    05H,0EDH,05H,98H,05H,47H,04H,0FCH
  05    ED    05    98    05
  47    04    FC
0890                    3540        DB    04H,0B4H,04H,70H,04H,31H,03H,0F4H
  04    B4    04    70    04
  31    03    F4
0898                    3550        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
08A0                    3560        DB    03H,0BCH,03H,86H,03H,53H,03H,24H
```

```
  03    BC    03    86    03
  53    03    24
08A8                 3570        DB    02H,0F6H,02H,0CCH,02H,0A4H,02H,7EH
  02    F6    02    CC    02
  A4    02    7E
08B0                 3580        DB    02H,5AH,02H,38H,02H,18H,01H,0FAH
  02    5A    02    38    02
  18    01    FA
08B8                 3590        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
08C0                 3600        DB    01H,0DEH,01H,0C3H,01H,0AAH,01H,92H
  01    DE    01    C3    01
  AA    01    92
08C8                 3610        DB    01H,7BH,01H,66H,01H,52H,01H,3FH
  01    7B    01    66    01
  52    01    3F
08D0                 3620        DB    01H,2DH,01H,1CH,01H,0CH,00H,0FDH
  01    2D    01    1C    01
  0C    00    FD
08D8                 3630        DB    0,0,0,0,0,0,0,0
  00    00    00    00    00
  00    00    00
08E0                 3640        DB    00H,0EFH,00H,0E1H,00H,0D5H,00H,0C9H
  00    EF    00    E1    00
  D5    00    C9
08E8                 3650        DB    00H,0BEH,00H,0B3H,00H,0A9H,00H,9FH
  00    BE    00    B3    00
  A9    00    9F
08F0                 3660        DB    00H,96H,00H,8EH,00H,86H,00H,7FH
  00    96    00    8E    00
  86    00    7F
08F8                 3665        DB    00H,77H
  00    77
5FF0                 3670        ORG   5FF0H INTERRUPT BRANCH TABLE
0FF0                 3675        ST    0FF0H
0FF0                 3680        DW    0000H
  00 00
0FF2                 3690        DW    0000H
  00 00
0FF4                 3700        DW    INT2
  C3 09
0FF6                 3710        DW    0000H
  00 00
0FF8                 3720        DW    INT4
  44 0A
0FFA                 3730        DW    0000H
  00 00
0FFC                 3740        DW    0000H
  00 00
0FFE                 3750        DW    0000H
  00 00
```

```
LTABL
B1       0976    B2       097A    B3       0986    B4       098A
B5       09B6    BASE     0D00    BASEC    0923    BEAT     0D24
BEAT0    0AB4    BEAT1    0B45    BEAT2    0B3D    BEAT3    0AF2
BEAT4    0B3A    BEATI    0001    BEET     0AAC    BKACTN   0008
BKCNT    0D1F    BNB      0006    BNBN     00F9    BREAK    0A74
```

```
BREAKI 0004    BRK0   0A7C    BRK1   0A9F    BRKIN  0004
BTCNT  0D22    BTCNT0 0D0A    BTLEN0 0D01    BTNXT  0D1E
BTTIME 0D1D    CURLEV 0D29    CURMSK 0D2A    FLAGS  0D27
FNXTH0 0D13    FNXTL0 0D10    FREQH0 0D19    FREQL0 0D16
FREQLU 0800    I2A    07F0    I2B    0A1F    I2C    0A2B
I4A    0AA7    I4B    0B4D    I4C    0B9F    I4D    0BBD
I4E    0BA7    IFLAG  0D20    INIT   0900    INK1   0BD7
INKBD  0BD2    INT2   09C3    INT4   0A44    INTACN 0010
INTCNT 0D21    LOOP   094D    LP1    095C    LP2    0998
MNXT0  0D04    MODE0  0D07    MSCNT  0D23    OBEATI 0002
OPT0   0B57    OPT1   0B6C    OPT2   0B82    OPT3   0B84
OPT4   0B70    OPT5   0B85    PGATES 0D1C    PIECE  0D25
RDY0   0D0D    STACK  0FEF    STOP   07C2
```

APPENDIX H

```
0100 *********************************************************
0110 *  INITIALIZE INTERRUPTS
0120 *
0130 INIT   LD   SP,STACK
0140        LD   A,0FFH
0150        OUT  0E1H,A      CLEAR ALL PENDING INTERRUPTS
0160        LD   A,84H
0170        OUT  0E6H,A      RESET PROCESSOR INTERRUPT
0180        LD   A,0
0190        OUT  0E6H,A
0200        LD   A,08H       SET INTERRUPT TO BASE LEVEL
0210        OUT  0E0H,A
0220        LD   A,(CURMSK)  UNMASK INTERRUPTS
0230        OUT  0E1H,A
0240        LD   A,>TABLE    LOAD THE INTERRUPT VECTOR TABLE
0250        LD   I,A         SET INTERRUPT MODE 2
0260        IM   2
0270        EI
0280 PROG   JP   PROG        INITIALIZATION DONE
0290 STACK  EQU  57EFH
0300 *
0310 *********************************************************
0320 *  SET UP THE INTERRUPT HANDLER VECTORS
0330 *
0340        ORG  57F0H
0350 TABLE  EQU  $
0360        DW   INT0
0370        DW   INT1
0380        DW   INT2
0390        DW   INT3
0400        DW   INT4
0410        DW   INT5
0420        DW   INT6
0430        DW   INT7
0440 *  NOTE:  INT0 THROUGH INT7 ARE THE STARTING ADDRESSES OF
0450 *         THE 8 INTERRUPT ROUTINES, REPRESENTED IN THE
0460 *         FOLLOWING SAMPLE INTERRUPT HANDLER BY "INTN".
0470 *
0480 *********************************************************
0490 *  SAMPLE INTERRUPT HANDLER
0500 *
0510 INTN   PUSH AF          SAVE ACCUMULATOR AND FLAGS
0520        LD   A,(CURLEV)  PUT INTERRUPTED LEVEL IN STACK
0530        PUSH AF
0540        LD   A,N         WHERE N IS THE VALUE OF THIS LEVEL
0550        OUT  0E0H,A      TELL THE CONTROLER THE NEW LEVEL
0560        LD   (CURLEV),A  PUT THIS LEVEL INTO CURLEV
0570        EI               CONTINUE WITH INTERRUPT ROUTINE
0580 *
0590 *********************************************************
0600 *  THE FOLLOWING ROUTINE IS USED TO END INTERRUPT HANDLER
0610 *
0620 INTOUT POP  AF          REPLACE CURLEV WITH PREVIOUS LEVEL
0630        DI
0640        LD   (CURLEV),A
0650        LD   A,(CURMSK)  TOGGLE MASK BIT OF THIS LEVEL
0660        OR   MLVL        WHERE MLVL=2**(7-N), N=THIS LEVEL
0670        OUT  0E1H,A      OUTPUT TO MASK
0680        LD   A,(CURMSK)  RESTORE CURRENT MASK
0690        OUT  0E1H,A
```

```
0700        LD   A,(CURLEV)
0710        OUT  0E0H,A       TELL CONTROLLER PREVIOUS LEVEL
0720        POP  AF           RESTORE ACCUMULATOR AND FLAGS
0730        EI
0740        RET
0750 CURLEV DB   08H
0760 CURMSK DB   00H
0770        END
0780 *  EXAMPLE OF "N" AND "MLVL" FOR LEVEL 4:
0790 N      EQU  4D
0800 MLVL   EQU  8D

FILE  3000  3823
READY
```

APPENDIX I

```
6000              0100          ORG   140000
0660              0105          ST    6140
0660 21 0D 0A     0110 DIAGS    LD    HL,LST1
0663 CD 3D 06     0130          CALL  CLR
0666 26 00        0140          LD    H,0
0668 6C           0150          LD    L,H
0669 5C           0155          LD    E,H
066A DB E0        0160          INP   IRTC+0
066C 47           0170 DIAG1    LD    B,A
066D DB E0        0180          INP   IRTC+0
066F B8           0190          CP    B
0670 CA 6C 06     0200          JP    Z,DIAG1
0673 DB E0        0210          INP   IRTC+0
0675 47           0220          LD    B,A
0676 2C           0230 DIAG1A   INC   L
0677 C2 76 06     0240          JP    NZ,DIAG1A
067A DB E0        0250          INP   IRTC+0
067C B8           0260          CP    B
067D C2 91 06     0270          JP    NZ,DIAG1D
0680 1C           0272          INC   E
0681 C2 76 06     0274          JP    NZ,DIAG1A
0684 24           0280          INC   H
0685 C2 76 06     0290          JP    NZ,DIAG1A
0688 21 2F 0A     0300 DIAG1B   LD    HL,LST2
068B CD 08 06     0320 DIAG1C   CALL  MSG
068E C3 A2 06     0330          JP    DIAG2
0691 7C           0340 DIAG1D   LD    A,H
0692 FE 02        0350          CP    2D
0694 FA 88 06     0360          JP    M,DIAG1B
0697 FE 04        0370          CP    4D
0699 F2 88 06     0380          JP    P,DIAG1B
069C 21 3C 0A     0390          LD    HL,LST3
069F C3 8B 06     0400          JP    DIAG1C
06A2 CD 24 06     0410 DIAG2    CALL  KBD
06A5 21 49 0A     0420          LD    HL,LST4
06A8 CD 3D 06     0440          CALL  CLR
06AB 3E 96        0450          LD    A,96H
06AD D3 E1        0460          OUT   IRTC+1
06AF CD 24 06     0470          CALL  KBD
06B2 21 99 0A     0480          LD    HL,LST5
06B5 CD 3D 06     0500          CALL  CLR
06B8 CD 24 06     0510          CALL  KBD
06BB 3E 00        0520 DIAG3    LD    A,0
06BD D3 E1        0530          OUT   IRTC+1
06BF 3E 08        0540          LD    A,8H
06C1 D3 E0        0550          OUT   IRTC+0
06C3 21 E8 0A     0560          LD    HL,LST6
06C6 CD 3D 06     0580          CALL  CLR
06C9 CD 24 06     0590          CALL  KBD
06CC C7           0600          RST   0
06CD 21 19 0B     0610 CNTCH    LD    HL,LST7 COUNT CHAIN TEST
06D0 CD 3D 06     0620          CALL  CLR
06D3 3E 00        0630          LD    A,0
06D5 D3 E6        0640          OUT   IRTC+6
06D7 CD 24 06     0650 CNTCH1   CALL  KBD
06DA 3E 78        0660          LD    A,78H
06DC D3 E6        0670          OUT   IRTC+6
06DE DB E4        0680          IN    IRTC+4
06E0 CB 77        0690          BIT   6,A
06E2 C2 EE 06     0700          JP    NZ,CNTCH2
06E5 21 2B 0B     0710          LD    HL,LST8  STATUS FAILURE
```

```
06E8 CD 08 06        0720         CALL MSG
06EB C3 D7 06        0730         JP   CNTCH1
06EE 06 04           0740 CNTCH2 LD   B,4
06F0 0E E0           0750         LD   C,IRTC
06F2 ED 78           0760 CNTCH3 IN   A,(C)
06F4 E6 3F           0770         AND  3FH
06F6 FE 00           0780         CP   0
06F8 C2 09 07        0790         JP   NZ,CNTCH4
06FB 0C              0800         INC  C
06FC 05              0810         DEC  B
06FD C2 F2 06        0820         JP   NZ,CNTCH3
0700 21 3B 0B        0830         LD   HL,LST9  STATUS AND RESET OK
0703 CD 08 06        0840         CALL MSG
0706 C3 12 07        0850         JP   CNTCH5
0709 21 5A 0B        0860 CNTCH4 LD   HL,LST10  RESET FAILURE
070C CD 08 06        0870         CALL MSG
070F C3 D7 06        0880         JP   CNTCH1
0712 CD 24 06        0890 CNTCH5 CALL KBD
0715 3E 20           0900         LD   A,40
0717 D3 E5           0910         OUT  IRTC+5
0719 3E 40           0920         LD   A,40H
071B D3 E6           0930         OUT  IRTC+6
071D 0E 48           0940         LD   C,48H
071F DB E0           0950 CNTCH6 IN   IRTC
0721 FE 01           0960         CP   1
0723 CA 33 07        0970         JP   Z,CNTCH8
0726 0D              0980         DEC  C
0727 C2 1F 07        0990         JP   NZ,CNTCH6
072A 21 69 0B        1000 CNTCH7 LD   HL,LST11  TIME SET FAILURE
072D CD 08 06        1010         CALL MSG
0730 C3 57 07        1020         JP   CNTCHB
0733 DB E2           1030 CNTCH8 IN   IRTC+2
0735 FE 01           1040         CP   1
0737 C2 2A 07        1050         JP   NZ,CNTCH7
073A 0E 48           1060         LD   C,48H
073C DB E0           1070 CNTCH9 IN   IRTC
073E FE 02           1080         CP   2
0740 CA 4A 07        1090         JP   Z,CNTCHA
0743 0D              1100         DEC  C
0744 C2 3C 07        1101         JP   NZ,CNTCH9
0747 C3 2A 07        1103         JP   CNTCH7
074A DB E2           1105 CNTCHA IN   IRTC+2
074C FE 02           1107         CP   2
074E C2 2A 07        1110         JP   NZ,CNTCH7
0751 21 7B 0B        1120         LD   HL,LST12  TIME SET OK
0754 CD 08 06        1130         CALL MSG
0757 CD 24 06        1140 CNTCHB CALL KBD
075A C7              1150         RST  0
075B                 1160 ********************************************
075B 21 88 0B        1170 ADIO1  LD   HL,LST13  ADDR + I/O TEST 1
075E CD 3D 06        1180         CALL CLR
0761 0E F6           1190         LD   C,0F6H
0763 79              1200 ADIO1A LD   A,C
0764 C6 10           1210         ADD  10H
0766 4F              1220         LD   C,A
0767 FE E6           1230         CP   0E6H
0769 CA 63 07        1240         JP   Z,ADIO1A
076C ED 79           1250         OUT  (C),A
076E ED 78           1260         IN   A,(C)
0770 C3 63 07        1270         JP   ADIO1A
0773                 1280 ********************************************
0773 16 00           1290 ADIO2  LD   D,0
0775 21 02 0A        1291         LD   HL,CLEAR
```

```
0778 CD 08 06       1292        CALL MSG
077B 21 A0 0B       1293        LD   HL,LST14  ADDR+I/O CHK LO LEVELS
077E CD 08 06       1294        CALL MSG
0781 06 08          1300 ADIO2A LD   B,8D
0783 3E DF          1310 ADIO2B LD   A,IRTC-1
0785 80             1320        ADD  B
0786 4F             1330        LD   C,A
0787 7A             1340        LD   A,D
0788 ED 79          1350        OUT  (C),A
078A ED 78          1355        IN   A,(C)
078C DB 00          1360        IN   KB
078E 17             1370        RLA
078F DA 9F 07       1380        JP   C,ADIO2D
0792 05             1390        DEC  B
0793 C2 83 07       1400        JP   NZ,ADIO2B
0796 0E 40          1410        LD   C,40H
0798 0D             1420 ADIO2C DEC  C
0799 C2 98 07       1430        JP   NZ,ADIO2C
079C C3 81 07       1440        JP   ADIO2A
079F CD 24 06       1450 ADIO2D CALL KBD
07A2 7A             1460        LD   A,D
07A3 FE 00          1470        CP   0
07A5 C2 B3 07       1480        JP   NZ,ADIO2E
07A8 21 CF 0B       1490        LD   HL,LST15  CHECK FOR HIGH LEVELS
07AB CD 08 06       1500        CALL MSG
07AE 16 FF          1510        LD   D,0FFH
07B0 C3 81 07       1520        JP   ADIO2A
07B3 C7             1530 ADIO2E RST  0
07B4                1540 ******************************************
5800                8000        ORG  130000
0600                8005        ST   6000
0600 3E A0          8010 TVSPC  LD   A,SPC
0602 D3 00          8020 TVOUT  OUT  TV
0604 97             8030        SUB  A
0605 D3 00          8040        OUT  TV
0607 C9             8050        RET
0608                8060
0608                8070 *****************************************
0608 7E             8080 MSG    LD   A,(HL)
0609 FE 00          8090        CP   0
060B C8             8100        RET  Z
060C 17             8110        RLA
060D DA 1C 06       8120        JP   C,MSG2
0610 1F             8130        RRA
0611 47             8140        LD   B,A
0612 CD 00 06       8150 MSG1   CALL TVSPC
0615 05             8160        DEC  B
0616 C2 12 06       8170        JP   NZ,MSG1
0619 C3 20 06       8180        JP   MSG3
061C 1F             8190 MSG2   RRA
061D CD 02 06       8200        CALL TVOUT
0620 23             8210 MSG3   INC  HL
0621 C3 08 06       8240        JP   MSG
0624                8250 *******************************************
0624 DB 00          8260 KBD    IN   KB
0626 17             8270        RLA
0627 D2 24 06       8280        JP   NC,KBD
062A 47             8300 KBD1   LD   B,A
062B DB 00          8310        IN   KB
062D 17             8320        RLA
062E DA 2A 06       8330        JP   C,KBD1
0631 78             8340        LD   A,B
0632 1F             8350        RRA
```

```
0633 C9           8360         RET
0634              8370 *********************************************
0634 D3 E2        8380 OUTP    OUT  IRTC+2
0636 D3 E3        8390         OUT  IRTC+3
0638 D3 E4        8400         OUT  IRTC+4
063A D3 E5        8410         OUT  IRTC+5
063C C9           8420         RET
063D              8430 ********************************************
063D EB           8440 CLR     EX   DE,HL
063E 21 02 0A     8450         LD   HL,CLEAR
0641 CD 08 06     8460         CALL MSG
0644 EB           8470         EX   DE,HL
0645 CD 08 06     8480         CALL MSG
0648 C9           8490         RET
0649              8500 **********************************************
6800              8980         ORG  150000
0A00              8990         ST   12000
0A00              9000 SPC     EQU  0A0H
0A00              9010 TVLSTP  DS   2
0A02              9012 CLEAR   DB   255D
  FF
0A03              9013         DB   127D
  7F
0A04              9014         DB   127D
  7F
0A05              9015         DB   127D
  7F
0A06              9016         DB   127D
  7F
0A07              9017         DB   127D
  7F
0A08              9018         DB   127D
  7F
0A09              9019         DB   127D
  7F
0A0A              9020         DB   127D
  7F
0A0B              9021         DB   8D
  08
0A0C              9022         DB   0
  00
0A0D              9023 LST1    DC   'TESTING REAL TIME CLOCK COUNTING'
  D4   C5   D3   D4   C9
  CE   C7   A0   D2   C5
  C1   CC   A0   D4   C9
  CD   C5   A0   C3   CC
  CF   C3   CB   A0   C3
  CF   D5   CE   D4   C9
  CE   C7
0A2D              9025         DB   32D
  20
0A2E              9030         DB   0
  00
0A2F              9040 LST2    DC   'TEST FAILED'
  D4   C5   D3   D4   A0
  C6   C1   C9   CC   C5
  C4
0A3A              9050         DB   53D
  35
0A3B              9060         DB   0
  00
0A3C              9070 LST3    DC   'TEST PASSED'
  D4   C5   D3   D4   A0
  D0   C1   D3   D3   C5
  C4
0A47                9080        DB   53D
  35
0A48                9090        DB   0
  00
0A49                9100 IRTC   EQU  0E0H
0A49                9110 KB     EQU  0  KBD PORT
0A49                9111 TV     EQU  0  TV OUTPUT PORT
0A49                9118 LST4   DC   'CHECK FOR A HIGH LEVEL ON:'
  C3   C8   C5   C3   CB
  A0   C6   CF   D2   A0
  C1   A0   C8   C9   C7
  C8   A0   CC   C5   D6
  C5   CC   A0   CF   CE
  BA
0A63                9120        DB   38D
  26
0A64                9130        DC   'IC 1, PIN 13'
  C9   C3   A0   B1   AC
  A0   D0   C9   CE   A0
  B1   B3
0A70                9140        DB   52D
  34
0A71                9150        DC   'IC 2, PIN 1'
  C9   C3   A0   B2   AC
  A0   D0   C9   CE   A0
  B1
0A7C                9160        DB   53D
  35
0A7D                9170        DC   'IC 22, PIN 1'
  C9   C3   A0   B2   B2
  AC   A0   D0   C9   CE
  A0   B1
0A89                9180        DB   52D
  34
0A8A                9190        DC   'IC 23, PIN 13'
  C9   C3   A0   B2   B3
  AC   A0   D0   C9   CE
  A0   B1   B3
0A97                9200        DB   51D
  33
0A98                9201        DB   0
  00
0A99                9222 LST5   DC   'CHECK FOR A LOW LEVEL ON:'
  C3   C8   C5   C3   CB
  A0   C6   CF   D2   A0
  C1   A0   CC   CF   D7
  A0   CC   C5   D6   C5
  CC   A0   CF   CE   BA

0AB2                9230        DB   39D
  27
0AB3                9240        DC   'IC 1, PIN 1'
  C9   C3   A0   B1   AC
  A0   D0   C9   CE   A0
  B1
0ABE                9250        DB   53D
  35
0ABF                9260        DC   'IC 2, PIN 13'
  C9   C3   A0   B2   AC
  A0   D0   C9   CE   A0
  B1   B3
```

```
0ACB              9270        DB   52D
  34
0ACC              9280        DC   'IC 22, PIN 13'
  C9   C3   A0   B2   B2
  AC   A0   D0   C9   CE
  A0   B1   B3
0AD9              9290        DB   51D
  33
0ADA              9300        DC   'IC 23, PIN 1'
  C9   C3   A0   B2   B3
  AC   A0   D0   C9   CE
  A0   B1
0AE6              9310        DB   52D
  34
0AE7              9320        DB   0
  00
0AE8              9430 LST6   DC   'CHECK THAT PINS 13,14, AND 15 OF IC 21'
  C3   C8   C5   C3   CB
  A0   D4   C8   C1   D4
  A0   D0   C9   CE   D3
  A0   B1   B3   AC   B1
  B4   AC   A0   C1   CE
  C4   A0   B1   B5   A0
  CF   C6   A0   C9   C3
  A0   B2   B1
0B0E              9440        DC   ' ARE LOW.'
  A0   C1   D2   C5   A0
  CC   CF   D7   AE
0B17              9450        DB   15D
  0F
0B18              9460        DB   0
  00
0B19              9470 LST7   DC   'COUNT CHAIN TEST'
  C3   CF   D5   CE   D4
  A0   C3   C8   C1   C9
  CE   A0   D4   C5   D3
  D4
0B29              9480        DB   48D
  30
0B2A              9490        DB   0
  00
0B2B              9500 LST8   DC   'STATUS FAILURE'
  D3   D4   C1   D4   D5
  D3   A0   C6   C1   C9
  CC   D5   D2   C5
0B39              9510        DB   50D
  32
0B3A              9520        DB   0
  00
0B3B              9530 LST9   DC   'STATUS AND RESET FUNCTIONS OK'
  D3   D4   C1   D4   D5
  D3   A0   C1   CE   C4
  A0   D2   C5   D3   C5
  D4   A0   C6   D5   CE
  C3   D4   C9   CF   CE
  D3   A0   CF   CB
0B58              9540        DB   35D
  23
0B59              9550        DB   0
  00
0B5A              9560 LST10  DC   'RESET FAILURE'
  D2   C5   D3   C5   D4
  A0   C6   C1   C9   CC
  D5    D2    C5
OB67              9570          DB    51D
  33
OB68              9580          DB    0
  00
OB69              9590 LST11    DC    'TIME SET FAILURE'
  D4    C9    CD    C5    A0
  D3    C5    D4    A0    C6
  C1    C9    CC    D5    D2
  C5
OB79              9600          DB    48D
  30
OB7A              9610          DB    0
  00
OB7B              9620 LST12    DC    'TIME SET OK'
  D4    C9    CD    C5    A0
  D3    C5    D4    A0    CF
  CB
OB86              9630          DB    53D
  35
OB87              9640          DB    0
  00
OB88              9650 LST13    DC    'ADDRESS AND I/O TEST 1'
  C1    C4    C4    D2    C5
  D3    D3    A0    C1    CE
  C4    A0    C9    AF    CF
  A0    D4    C5    D3    D4
  A0    B1
OB9E              9660          DB    42D
  2A
OB9F              9670          DB    0
  00
OBA0              9680 LST14    DC    'ADDRESS AND I/O TEST 2'
  C1    C4    C4    D2    C5
  D3    D3    A0    C1    CE
  C4    A0    C9    AF    B0
  A0    D4    C5    D3    D4
  A0    B2
OBB6              9690          DB    42D
  2A
OBB7              9700          DC    'CHECK FOR GIVEN LEVELS'
  C3    C8    C5    C3    CB
  A0    C6    CF    D2    A0
  C7    C9    D6    C5    CE
  A0    CC    C5    D6    C5
  CC    D3
OBCD              9710          DB    42D
  2A
OBCE              9720          DB    0
  00
OBCF              9730 LST15    DC    'CHECK FOR OPPOSITE LEVELS'
  C3    C8    C5    C3    CB
  A0    C6    CF    D2    A0
  CF    D0    D0    CF    D3
  C9    D4    C5    A0    CC
  C5    D6    C5    CC    D3

OBE8              9740          DB    39D
  27
OBE9              9750          DB    0
  00
OBEA              9760 LST16    DC    'CHECK INTERRUPT REQUESTS LOW'
  C3    C8    C5    C3    CB
  A0    C9    CE    D4    C5
  D2    D2    D5    D0    D4
  A0    D2    C5    D1    D5
  C5    D3    D4    D3    A0
  CC    CF    D7
0C06                 9770         DB   36D
  24
0C07                 9780         DB   0
  00
0C08                 9790 LST17   DC   'CHECK INTERRUPT REQUESTS HIGH'
  C3    C8    C5    C3    CB
  A0    C9    CE    D4    C5
  D2    D2    D5    D0    D4
  A0    D2    C5    D1    D5
  C5    D3    D4    D3    A0
  C8    C9    C7    C8
0C25                 9800         DB   35D
  23
0C26                 9810         DB   0
  00
0C27                 9820 LST18   DC   'CHECK "INT" LOW'
  C3    C8    C5    C3    CB
  A0    A2    C9    CE    D4
  A2    A0    CC    CF    D7

0C36                 9830         DB   49D
  31
0C37                 9840         DB   0
  00
0C38                 9850 LST19   DC   'PRIORITY FAILURE'
  D0    D2    C9    CF    D2
  C9    D4    D9    A0    C6
  C1    C9    CC    D5    D2
  C5
0C48                 9860         DB   48D
  30
0C49                 9870         DB   0
  00
0C4A                 9880 LST20   DC   'PRIORITY OK'
  D0    D2    C9    CF    D2
  C9    D4    D9    A0    CF
  CB
0C55                 9890         DB   53D
  35
0C56                 9900         DB   0
  00
0C57                 9910 LST21   DC   'INTERRUPT ACKNOWLEDGE TEST'
  C9    CE    D4    C5    D2
  D2    D5    D0    D4    A0
  C1    C3    CB    CE    CF
  D7    CC    C5    C4    C7
  C5    A0    D4    C5    D3
  D4
0C71                 9920         DB   38D
  26
0C72                 9930         DB   0
  00
0C73                 9940 LST22   DC   'SELECTORS AND DECODERS TEST #'
  D3    C5    CC    C5    C3
  D4    CF    D2    D3    A0
  C1    CE    C4    A0    C4
  C5    C3    CF    C4    C5
  D2    D3    A0    D4    C5
  D3    D4    A0    A3
0C90                  9941          DB   35D
  23
0C91                  9942          DC   '1 '
  B1    A0
0C93                  9943          DB   0
  00
0C94                  9946 LST23    DC   '2 '
  B2    A0
0C96                  9948          DB   0
  00
0C97                  9950 LST24    DC   '3 '
  B3    A0
0C99                  9952          DB   0
  00
0C9A                  9954 LST25    DC   '4 '
  B4    A0
0C9C                  9956          DB   0
  00
0C9D                  9958 LST26    DC   '5 '
  B5    A0
0C9F                  9960          DB   0
  00
0CA0                  9962 LST27    DC   '6 '
  B6    A0
0CA2                  9964          DB   0
  00
0CA3                  9966 LST28    DC   '7 '
  B7    A0
0CA5                  9968          DB   0
  00
0CA6                  9970 LST29    DC   '8 '
  B8    A0
0CA8                  9972          DB   0
  00
0CA9                  9974 LST30    DC   '9 '
  B9    A0
0CAB                  9976          DB   0
  00
0CAC                  9978 LST31    DC   '10 '
  B1    B0    A0
0CAF                  9980          DB   0
  00
0CB0                  9982 LST32    DC   '11 '
  B1    B1    A0
0CB3                  9984          DB   0
  00
0CB4                  9986 LST33    DC   '12 '
  B1    B2    A0
0CB7                  9988          DB   0
  00




LTABL
ADIO1   075B   ADIO1A  0763   ADIO2   0773   ADIO2A  0781
ADIO2B  0783   ADIO2C  0798   ADIO2D  079F   ADIO2E  07B3
CLEAR   0A02   CLR     063D   CNTCH   06CD   CNTCH1  06D7
CNTCH2  06EE   CNTCH3  06F2   CNTCH4  0709   CNTCH5  0712
CNTCH6  071F   CNTCH7  072A   CNTCH8  0733   CNTCH9  073C
CNTCHA  074A   CNTCHB  0757   DIAG1   066C   DIAG1A  0676
```

```
DIAG1B 0688   DIAG1C 068B   DIAG1D 0691   DIAG2  06A2
DIAG3  06BB   DIAGS  0660   IRTC   00E0   KB     0000
KBD    0624   KBD1   062A   LST1   0A0D   LST10  0B5A
LST11  0B69   LST12  0B7B   LST13  0B88   LST14  0BA0
LST15  0BCF   LST16  0BEA   LST17  0C08   LST18  0C27
LST19  0C32   LST2   0A2F   LST20  0C44   LST21  0C51
LST22  0C6D   LST23  0C8E   LST24  0C91   LST25  0C94
LST26  0C97   LST27  0C9A   LST28  0C9D   LST29  0CA0
LST3   0A3C   LST30  0CA3   LST31  0CA6   LST32  0CAA
LST33  0CAE   LST4   0A49   LST5   0A99   LST6   0AE8
LST7   0B19   LST8   0B2B   LST9   0B3B   MSG    0608
MSG1   0612   MSG2   061C   MSG3   0620   OUTP   0634
SPC    00A0   TV     0000   TVLSTP 0A00   TVOUT  0602
TVSPC  0600
```

```
6000                0100        ORG  140000
0800                0110        ST   10000
0800 21 EA 0B       1550 INTST  LD   HL,LST16  CHK INT REQUESTS LOW
0803 CD 3D 06       1560        CALL CLR
0806 3E 00          1570        LD   A,0
0808 D3 E6          1580        OUT  IRTC+6
080A 3E FF          1590        LD   A,0FFH
080C D3 E1          1600        OUT  IRTC+1
080E 3E 08          1602        LD   A,08H
0810 D3 E0          1604        OUT  IRTC
0812 3E 00          1610        LD   A,0
0814 D3 E1          1620        OUT  IRTC+1
0816 CD 24 06       1630        CALL KBD
0819 3E FF          1640        LD   A,0FFH
081B D3 E1          1650        OUT  IRTC+1
081D 21 08 0C       1660        LD   HL,LST17  CHK INT REQUESTS HI
0820 CD 08 06       1670        CALL MSG
0823 CD 24 06       1680        CALL KBD
0826 3E 04          1690        LD   A,4
0828 D3 E6          1700        OUT  IRTC+6
082A 3E 00          1702        LD   A,0
082C D3 E6          1704        OUT  IRTC+6
082E 21 27 0C       1710        LD   HL,LST18  CHK INT LOW
0831 CD 08 06       1720        CALL MSG
0834 CD 24 06       1730        CALL KBD
0837 3E 00          1740 INTST0 LD   A,0
0839 D3 E6          1750        OUT  IRTC+6
083B 3E 03          1760        LD   A,3
083D D3 E0          1770        OUT  IRTC
083F 3E F0          1780        LD   A,0F0H
0841 D3 E1          1790        OUT  IRTC+1
0843 DB E4          1800        IN   IRTC+4
0845 CB 7F          1810        BIT  7,A
0847 C2 56 08       1820        JP   NZ,INTST1
084A 21 38 0C       1830        LD   HL,LST19  PRIORITY FAILURE
084D CD 08 06       1840        CALL MSG
0850 CD 24 06       1850        CALL KBD
0853 C3 37 08       1860        JP   INTST0
0856 21 4A 0C       1870 INTST1 LD   HL,LST20  PRIORITY OK
0859 CD 08 06       1880 INTST2 CALL MSG
085C CD 24 06       1890 INTST3 CALL KBD
085F 21 57 0C       1900        LD   HL,LST21  INT ACKNOWLEDGE TEST
0862 CD 08 06       1910        CALL MSG
0865 3E FF          1920        LD   A,0FFH
0867 D3 E1          1930        OUT  IRTC+1
0869 3E 08          1932        LD   A,08H
086B D3 E0          1934        OUT  IRTC
086D 3E 00          1940        LD   A,0
086F D3 E1          1950        OUT  IRTC+1
0871 C3 5C 08       1960        JP   INTST3
0874                1970 ***********************************************
0874 21 73 0C       1980 SELD   LD   HL,LST22 SEL AND DEC TEST 1
0877 CD 3D 06       1990        CALL CLR
087A 3E 00          2000        LD   A,0
087C CD 34 06       2010        CALL OUTP
087F CD 24 06       2020        CALL KBD
0882 21 94 0C       2030        LD   HL,LST23  2
0885 CD 08 06       2040        CALL MSG
0888 3E 09          2050        LD   A,11
088A D3 E2          2060        OUT  IRTC+2
088C D3 E3          2070        OUT  IRTC+3
```

```
088E D3 E4          2080           OUT   IRTC+4
0890 3E 08          2090           LD    A,10
0892 D3 E5          2100           OUT   IRTC+5
0894 CD 24 06       2110           CALL  KBD
0897 21 97 0C       2120           LD    HL,LST24   3
089A CD 08 06       2130           CALL  MSG
089D 3E 0F          2140           LD    A,17
089F D3 E5          2150           OUT   IRTC+5
08A1 CD 24 06       2160           CALL  KBD
08A4 21 9A 0C       2170           LD    HL,LST25   4
08A7 CD 08 06       2180           CALL  MSG
08AA 3E 12          2190           LD    A,22
08AC CD 34 06       2200           CALL  OUTP
08AF CD 24 06       2210           CALL  KBD
08B2 21 9D 0C       2220           LD    HL,LST26   5
08B5 CD 08 06       2230           CALL  MSG
08B8 3E 1B          2240           LD    A,33
08BA CD 34 06       2250           CALL  OUTP
08BD CD 24 06       2260           CALL  KBD
08C0 21 A0 0C       2270           LD    HL,LST27   6
08C3 CD 08 06       2280           CALL  MSG
08C6 3E 24          2290           LD    A,44
08C8 CD 34 06       2300           CALL  OUTP
08CB CD 24 06       2310           CALL  KBD
08CE 21 A3 0C       2320           LD    HL,LST28   7
08D1 CD 08 06       2330           CALL  MSG
08D4 3E 2D          2340           LD    A,55
08D6 CD 34 06       2350           CALL  OUTP
08D9 CD 24 06       2360           CALL  KBD
08DC 21 A6 0C       2370           LD    HL,LST29   8
08DF CD 08 06       2380           CALL  MSG
08E2 3E 36          2390           LD    A,66
08E4 CD 34 06       2400           CALL  OUTP
08E7 CD 24 06       2410           CALL  KBD
08EA 21 A9 0C       2420           LD    HL,LST30   9
08ED CD 08 06       2430           CALL  MSG
08F0 3E 3F          2440           LD    A,77
08F2 CD 34 06       2450           CALL  OUTP
08F5 CD 24 06       2460           CALL  KBD
08F8 21 AC 0C       2470           LD    HL,LST31   10
08FB CD 08 06       2480           CALL  MSG
08FE 3E 00          2490           LD    A,0
0900 D3 E4          2500           OUT   IRTC+4
0902 3E 01          2510           LD    A,1
0904 D3 E6          2520           OUT   IRTC+6
0906 CD 24 06       2530           CALL  KBD
0907 21 B0 0C       2540           LD    HL,LST32   11
090C CD 08 06       2550           CALL  MSG
070F 3E 02          2560           LD    A,2
0911 D3 E6          2570           OUT   IRTC+6
0913 CD 24 06       2580           CALL  KBD
0916 21 B4 0C       2590           LD    HL,LST33   12
0919 CD 08 06       2600           CALL  MSG
091C 3E 03          2610           LD    A,3
091E D3 E6          2620           OUT   IRTC+6
0920 CD 24 06       2630           CALL  KBD
0923 C7             2640           RST   0
0924                2650 *******************************************
5800                8000           ORG   130000
0600                8005           ST    6000
0600 3E A0          8010 TVSPC     LD    A,SPC
0602 D3 00          8020 TVOUT     OUT   TV
0604 97             8030           SUB   A
```

```
0605 D3 00        8040        OUT  TV
0607 C9           8050        RET
0608              8060
0608              8070 *******************************************
0608 7E           8080 MSG    LD   A,(HL)
0609 FE 00        8090        CP   0
060B C8           8100        RET  Z
060C 17           8110        RLA
060D DA 1C 06     8120        JP   C,MSG2
0610 1F           8130        RRA
0611 47           8140        LD   B,A
0612 CD 00 06     8150 MSG1   CALL TVSPC
0615 05           8160        DEC  B
0616 C2 12 06     8170        JP   NZ,MSG1
0619 C3 20 06     8180        JP   MSG3
061C 1F           8190 MSG2   RRA
061D CD 02 06     8200        CALL TVOUT
0620 23           8210 MSG3   INC  HL
0621 C3 08 06     8240        JP   MSG
0624              8250 *********************************************
0624 DB 00        8260 KBD    IN   KB
0626 17           8270        RLA
0627 D2 24 06     8280        JP   NC,KBD
062A 47           8300 KBD1   LD   B,A
062B DB 00        8310        IN   KB
062D 17           8320        RLA
062E DA 2A 06     8330        JP   C,KBD1
0631 78           8340        LD   A,B
0632 1F           8350        RRA
0633 C9           8360        RET
0634              8370 *********************************************
0634 D3 E2        8380 OUTP   OUT  IRTC+2
0636 D3 E3        8390        OUT  IRTC+3
0638 D3 E4        8400        OUT  IRTC+4
063A D3 E5        8410        OUT  IRTC+5
063C C9           8420        RET
063D              8430 ********************************************
063D EB           8440 CLR    EX   DE,HL
063E 21 02 0A     8450        LD   HL,CLEAR
0641 CD 08 06     8460        CALL MSG
0644 EB           8470        EX   DE,HL
0645 CD 08 06     8480        CALL MSG
0648 C9           8490        RET
0649              8500 **********************************************
6800              8980        ORG  150000
0A00              8990        ST   12000
0A00              9000 SPC    EQU  0A0H
0A00              9010 TVLSTP DS   2
0A02              9012 CLEAR  DB   255D
  FF
0A03              9013        DB   127D
  7F
0A04              9014        DB   127D
  7F
0A05              9015        DB   127D
  7F
0A06              9016        DB   127D
  7F
0A07              9017        DB   127D
  7F
0A08              9018        DB   127D
  7F
0A09              9019        DB   127D
```

```
  7F
0A0A                 9020        DB   127D
  7F
0A0B                 9021        DB   8D
  08
0A0C                 9022        DB   0
  00
0A0D                 9023 LST1   DC   ´TESTING REAL TIME CLOCK COUNTING´
  D4   C5   D3   D4   C9
  CE   C7   A0   D2   C5
  C1   CC   A0   D4   C9
  CD   C5   A0   C3   CC
  CF   C3   CB   A0   C3
  CF   D5   CE   D4   C9
  CE   C7
0A2D                 9025        DB   32D
  20
0A2E                 9030        DB   0
  00
0A2F                 9040 LST2   DC   ´TEST FAILED´
  D4   C5   D3   D4   A0
  C6   C1   C9   CC   C5
  C4
0A3A                 9050        DB   53D
  35
0A3B                 9060        DB   0
  00
0A3C                 9070 LST3   DC   ´TEST PASSED´
  D4   C5   D3   D4   A0
  D0   C1   D3   D3   C5
  C4
0A47                 9080        DB   53D
  35
0A48                 9090        DB   0
  00
0A49                 9100 IRTC   EQU  0E0H
0A49                 9110 KB     EQU  0  KBD PORT
0A49                 9111 TV     EQU  0  TV OUTPUT PORT
0BEA                 9750        ST   13352
0BEA                 9760 LST16  DC   ´CHECK INTERRUPT REQUESTS LOW´
  C3   C8   C5   C3   CB
  A0   C9   CE   D4   C5
  D2   D2   D5   D0   D4
  A0   D2   C5   D1   D5
  C5   D3   D4   D3   A0
  CC   CF   D7
0C06                 9770        DB   36D
  24
0C07                 9780        DB   0
  00
0C08                 9790 LST17  DC   ´CHECK INTERRUPT REQUESTS HIGH´
  C3   C8   C5   C3   CB
  A0   C9   CE   D4   C5
  D2   D2   D5   D0   D4
  A0   D2   C5   D1   D5
  C5   D3   D4   D3   A0
  C8   C9   C7   C8
0C25                 9800        DB   35D
  23
0C26                 9810        DB   0
  00
0C27                 9820 LST18  DC   ´CHECK INT LOW   ´
  C3   C8   C5   C3   CB
  A0    C9    CE    D4    A0
  CC    CF    D7    A0    A0

 0C36               9830          DB    49D
  31
 0C37               9840          DB    0
  00
 0C38               9850 LST19    DC    'PRIORITY FAILURE'
  D0    D2    C9    CF    D2
  C9    D4    D9    A0    C6
  C1    C9    CC    D5    D2
  C5
 0C48               9860          DB    48D
  30
 0C49               9870          DB    0
  00
 0C4A               9880 LST20    DC    'PRIORITY OK'
  D0    D2    C9    CF    D2
  C9    D4    D9    A0    CF
  CB
 0C55               9890          DB    53D
  35
 0C56               9900          DB    0
  00
 0C57               9910 LST21    DC    'INTERRUPT ACKNOWLEDGE TEST'
  C9    CE    D4    C5    D2
  D2    D5    D0    D4    A0
  C1    C3    CB    CE    CF
  D7    CC    C5    C4    C7
  C5    A0    D4    C5    D3
  D4
 0C71               9920          DB    38D
  26
 0C72               9930          DB    0
  00
 0C73               9940 LST22    DC    'SELECTORS AND DECODERS TEST #'
  D3    C5    CC    C5    C3
  D4    CF    D2    D3    A0
  C1    CE    C4    A0    C4
  C5    C3    CF    C4    C5
  D2    D3    A0    D4    C5
  D3    D4    A0    A3
 0C90               9941          DB    35D
  23
 0C91               9942          DC    '1 '
  B1    A0
 0C93               9943          DB    0
  00
 0C94               9946 LST23    DC    '2 '
  B2    A0
 0C96               9948          DB    0
  00
 0C97               9950 LST24    DC    '3 '
  B3    A0
 0C99               9952          DB    0
  00
 0C9A               9954 LST25    DC    '4 '
  B4    A0
 0C9C               9956          DB    0
  00
 0C9D               9958 LST26    DC    '5 '
  B5    A0
 0C9F               9960          DB    0
```

```
  00
0CA0              9962 LST27  DC   '6 '
  B6   A0
0CA2              9964        DB   0
  00
0CA3              9966 LST28  DC   '7 '
  B7   A0
0CA5              9968        DB   0
  00
0CA6              9970 LST29  DC   '8 '
  B8   A0
0CA8              9972        DB   0
  00
0CA9              9974 LST30  DC   '9 '
  B9   A0
0CAB              9976        DB   0
  00
0CAC              9978 LST31  DC   '10 '
  B1   B0   A0
0CAF              9980        DB   0
  00
0CB0              9982 LST32  DC   '11 '
  B1   B1   A0
0CB3              9984        DB   0
  00
0CB4              9986 LST33  DC   '12 '
  B1   B2   A0
0CB7              9988        DB   0
  00




LTABL
CLEAR  0A02   CLR    063D   INTST  0800   INTST0 0837
INTST1 0856   INTST2 0859   INTST3 085C   IRTC   00E0
KB     0000   KBD    0624   KBD1   062A   LST1   0A0D
LST16  0BEA   LST17  0C08   LST18  0C27   LST19  0C38
LST2   0A2F   LST20  0C4A   LST21  0C57   LST22  0C73
LST23  0C94   LST24  0C97   LST25  0C9A   LST26  0C9D
LST27  0CA0   LST28  0CA3   LST29  0CA6   LST3   0A3C
LST30  0CA9   LST31  0CAC   LST32  0CB0   LST33  0CB4
MSG    0608   MSG1   0612   MSG2   061C   MSG3   0620
OUTP   0634   SELD   0874   SPC    00A0   TV     0000
TVLSTP 0A00   TVOUT  0602   TVSPC  0600
FILE  3000  40D7
READY
```

APPENDIX J

CONNECTOR PINOUT

DUAL 36 PIN CONNECTOR:

1	IRQ0	19	EXCL2
2	IRQ1	20	XGA2
3	IRQ2	21	TOUT2
4	IRQ3	22	1SG
5	IRQ4	23	
6	IRQ5	24	GND
7	IRQ6	25	
8	IRQ7	26	BAT2
9	INT	27	
10	INT	28	
11	FINT	29	BATTERY
12	FXCL2	30	GND
13	XCL0	31	
14	XGA0	32	
15	TOUT0	33	SETIME
16	XCL1	34	
17	XGA1	35	FSTAN
18	TOUT1	36	XFI

DUAL 22 PIN CONNECTOR:

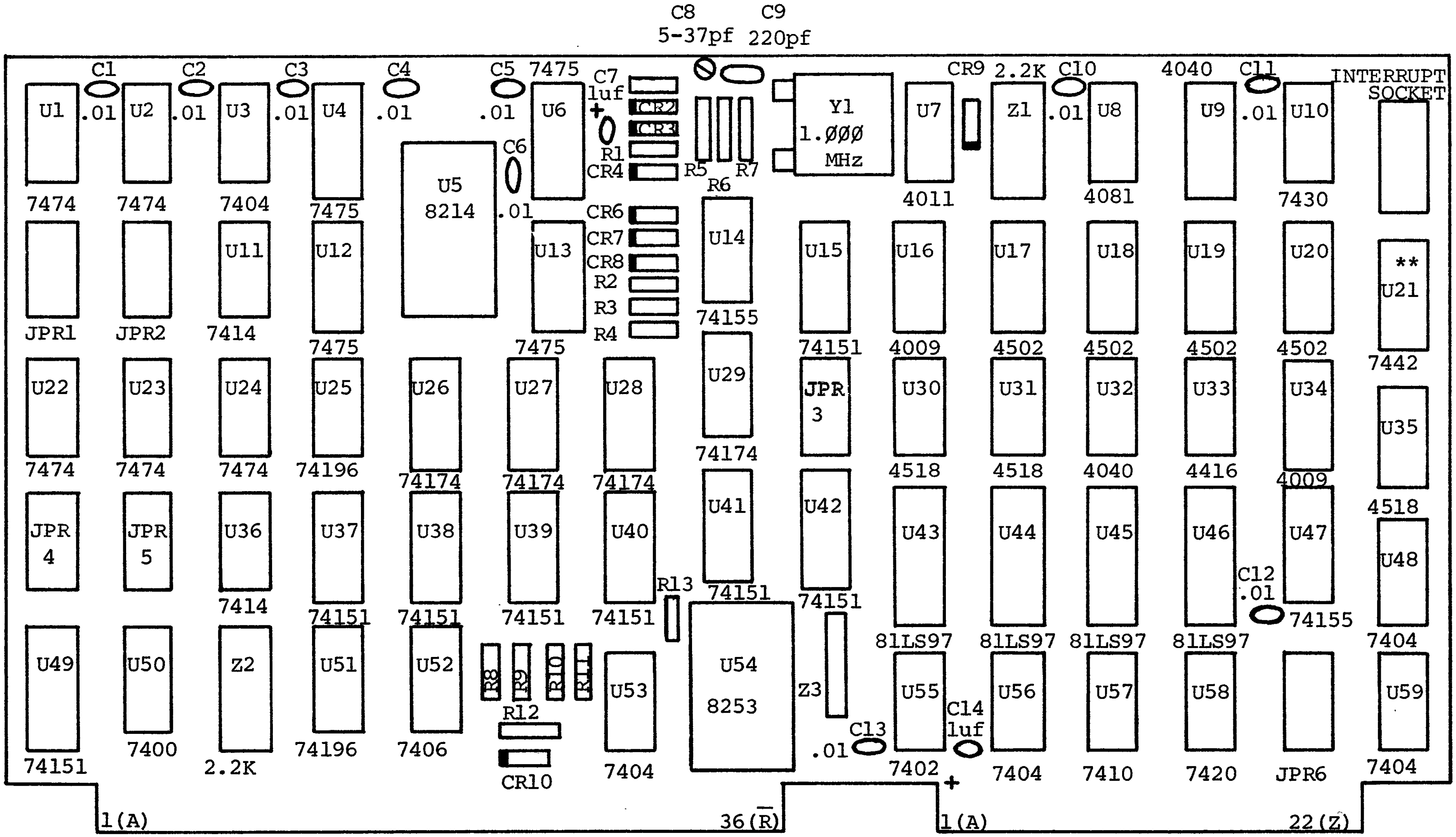
1.	+5V		A.	+5V	
2.	GROUND		B.		
3.	MSB		C.	MSB	
4.	MSB-1		D.	MSB-1	
5.	MSB-2	DATA TO CPU	E.	MSB-2	DATA FROM CPU
6.	MSB-3		F.	MSB-3	
7.	LSB+3		H.	LSB+3	
8.	LSB+2		J.	LSB+2	
9.	LSB+1		K.	LSB+1	
10.	LSB		L.	LSB	
11.	INPUT STROBE		M.		
12.			N.	LSB	
13.			P.	LSB+1	
14.			R.	LSB+2	PORT ADDRESS LINES
15.			S.	LSB+3	
16.			T.	LSB+4	
17.			U.	LSB+5	
18.			V.	LSB+6	
19.			W.	LSB+7	
20.			X.	OUTPUT STROBE	
21.			Y.		
22.	+12V		Z.		

APPENDIX K

PARTS LIST

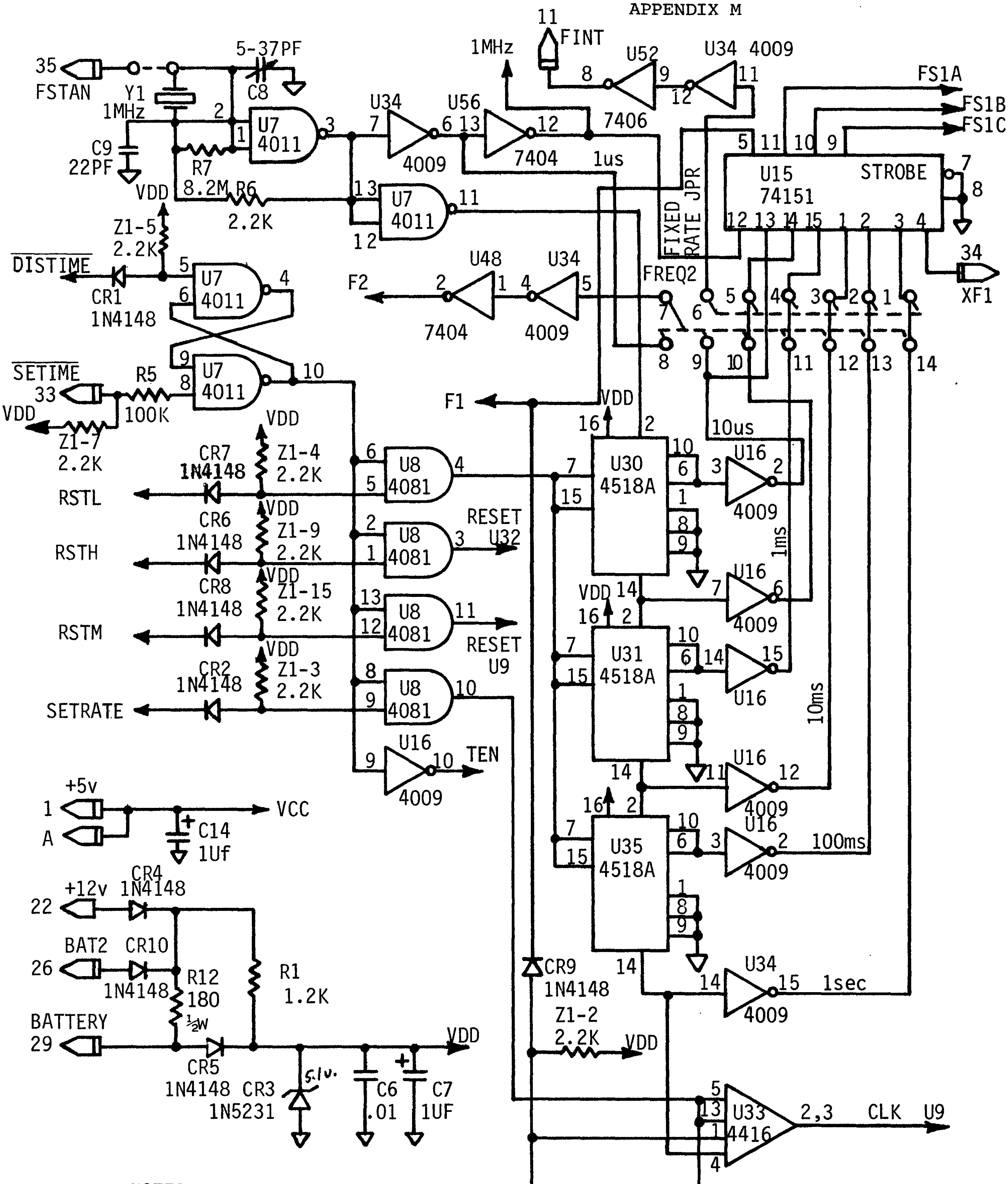
DESCRIPTION	QUANTITY	CIRCUIT REFERENCE
7400	1	IC50
7402	1	IC55
7404	5	IC3,48,53,56,59
7406	1	IC52
7410	1	IC57
7414	2	IC11,36
7420	1	IC58
7430	1	IC10
7442	1	IC21
7474	5	IC1,2,22,23,24
7475	4	IC4,6,12,13
74151	8	IC15,37,38,39,40,41,42,49
74155	2	IC14,47
74174	4	IC26,27,28,29
74196	2	IC25,51
4009A	2	IC16,34
4011A	1	IC7
4040A	2	IC9,32
4081A	1	IC8
4416A	1	IC33
4502A	4	IC17,18,19,20
4518A	3	IC30,31,35
81LS97	4	IC43,45,46,44
8214	1	IC5
8253	1	IC54
1N4148	9	CR1,2,4,5,6,7,8,9,10
1N5231	1	CR3
5.6 K R-PAK (8 pin sip)	1	Z3
2.2K R-PAK (16 pin dip)	2	Z1,2
180 Ohm 1/2 W	1	R12
1.2K Ohm 1/4 W 5%	1	R1
2.2K Ohm 1/4 W 5%	8	R2,3,4,6,8,9,10,11
5.6K Ohm 1/4 W 5%	1	R13
100K Ohm 1/4 W 5%	1	R5
8.2 MEG Ohm 1/4 W 5%	1	R7
5-37 PF TRIMMER CAPACITOR	1	C8
220 PF AG MICA CAPACITOR	1	C9
0.01 MFD CAPACITOR	10	C1,2,3,4,5,6,10,11,12,13
1.0 MFD TANTULUM	2	C7,14
CRYSTAL SOCKET	1	(MAY BE OPTIONAL)
1.00000 MHz CRYSTAL	1	Y1
24 PIN DIP SOCKET	2	
20 PIN DIP SOCKET	4	
16 PIN DIP SOCKET	33	
14 PIN DIP SOCKET	29	
14 PIN HEADERS	6	
16 PIN HEADERS	1	
16 PIN DIP TO 16 PIN DIP RIBBON CABLE ASSEMBLY	1	

APPENDIX L



** U21 USED WITH
8080 ONLY

APPENDIX M



NOTES:

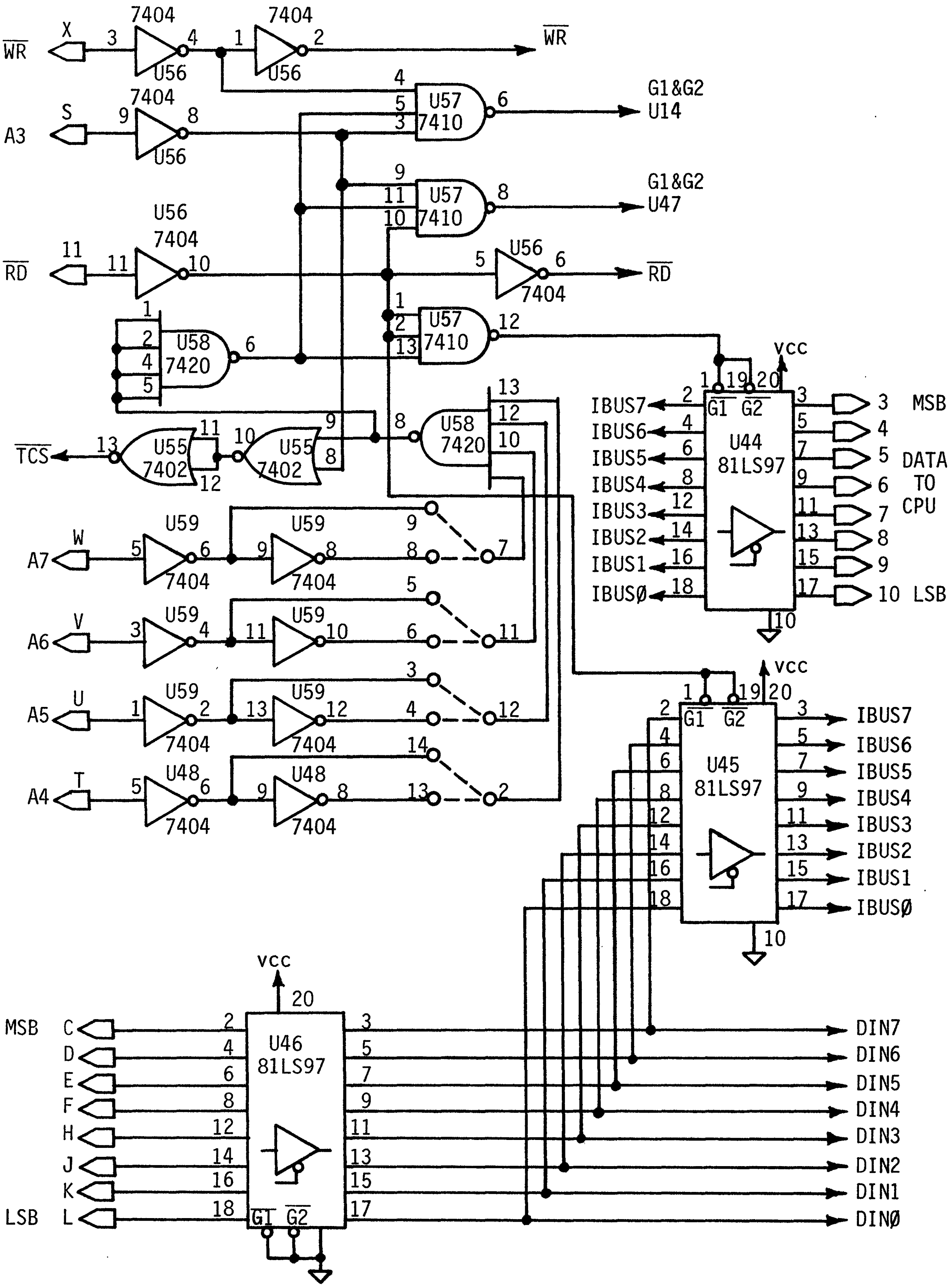
22 Pin Edge Connector

36 Pin Edge Connector

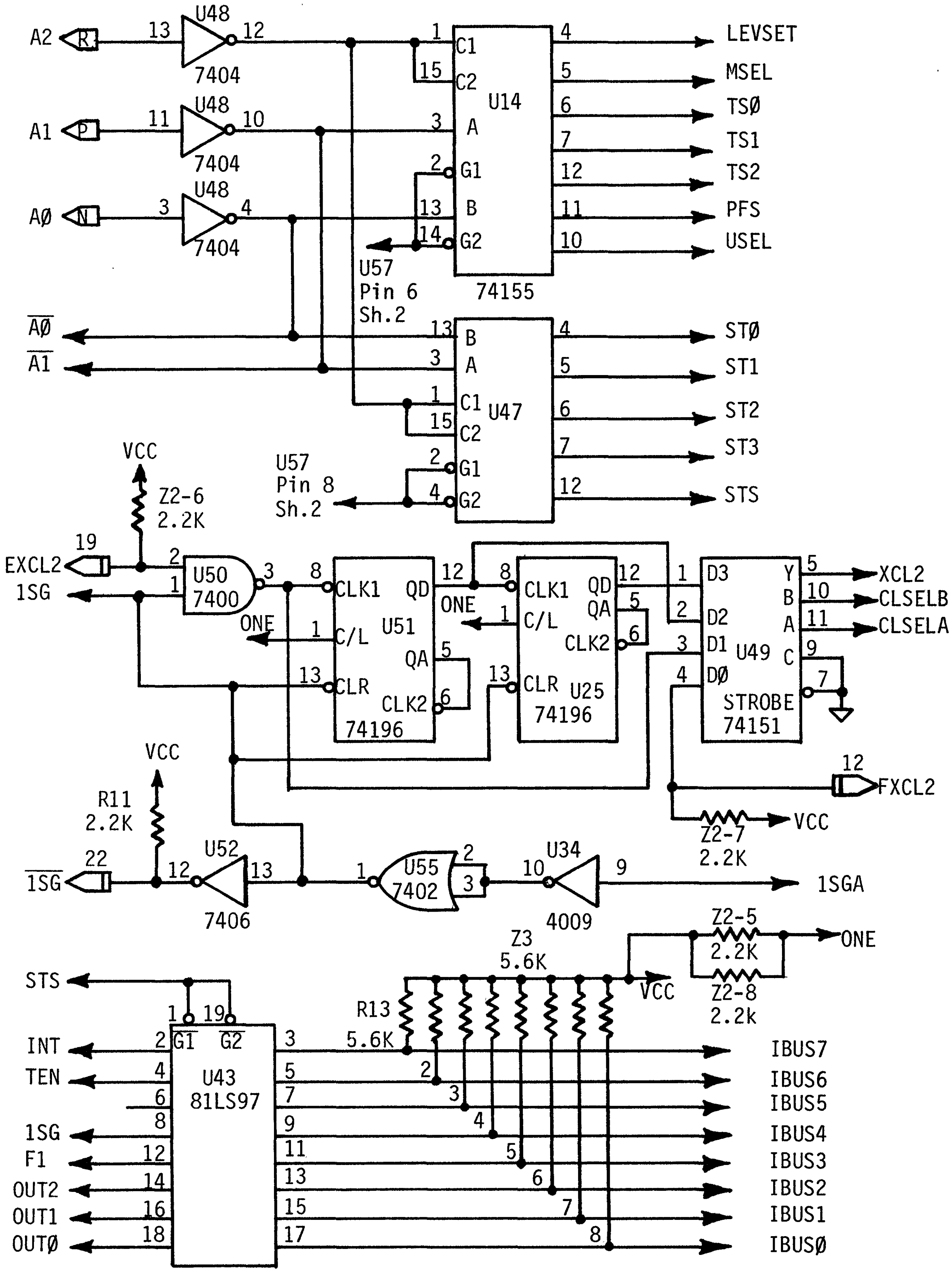
Signal Runs To Other Points

Jumper

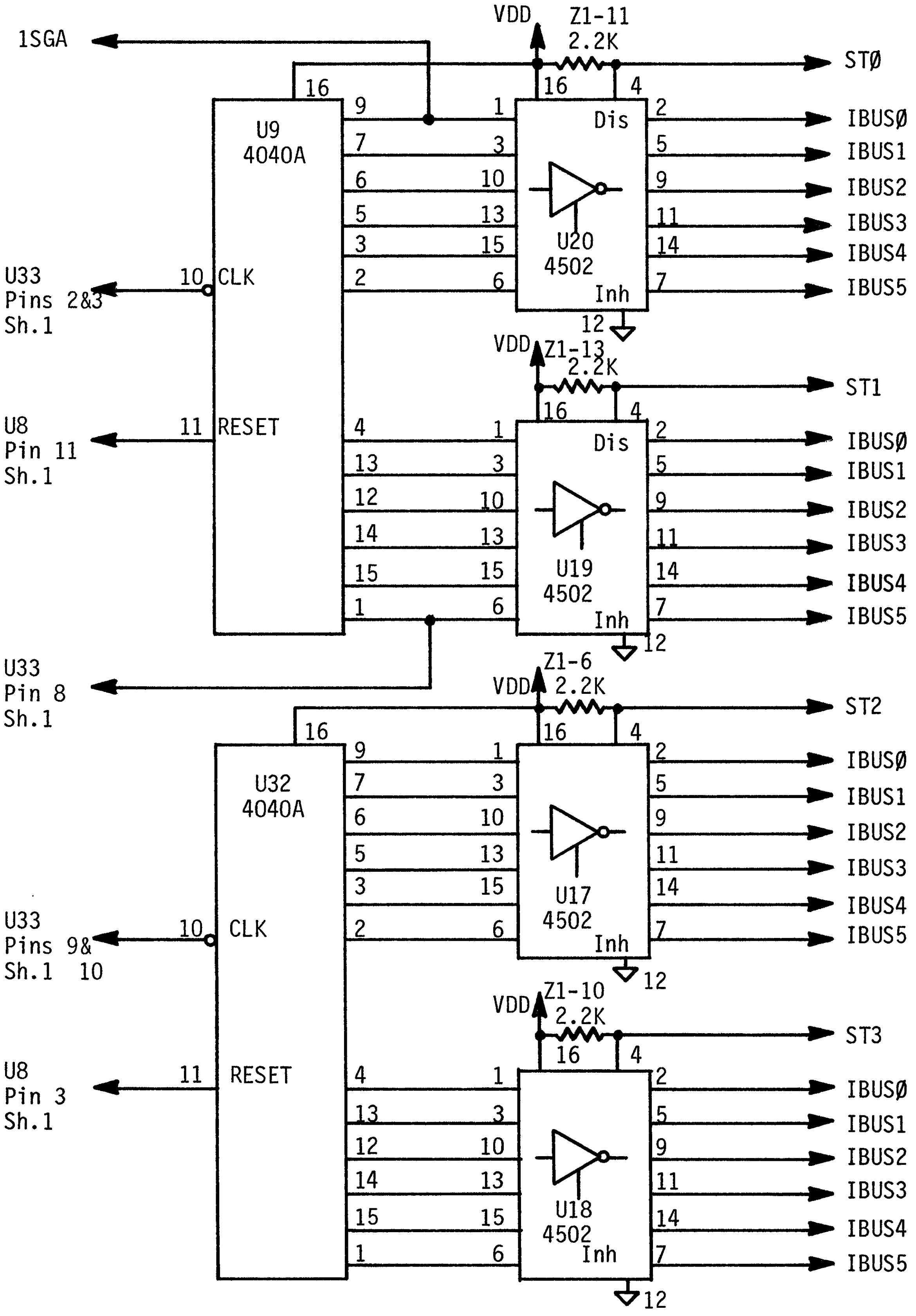
SHEET 1


WR
X
3
7404
U56
4
1
7404
U56
2
WR
A3
S
9
7404
U56
8
4
5
3
U57
7410
6
G1&G2
U14
9
11
10
U57
7410
8
G1&G2
U47
RD
11
11
U56
7404
10
5
U56
6
7404
RD
1
2
4
5
U58
7420
6
1
2
13
U57
7410
12
TCS
13
U55
7402
11
12
10
U55
7402
9
8
8
U58
7420
13
12
10
vcc
1
19
20
G1
G2
U44
81LS97
10
IBUS7
IBUS6
IBUS5
IBUS4
IBUS3
IBUS2
IBUS1
IBUSØ
2
4
6
8
12
14
16
18
3
5
7
9
11
13
15
17
3
4
5
6
7
8
9
10
MSB
DATA
TO
CPU
LSB
A7
W
5
U59
6
7404
9
U59
8
7404
9
8
7
A6
V
3
U59
4
11
U59
10
5
6
11
A5
U
1
U59
2
7404
13
U59
12
7404
3
4
12
A4
T
5
U48
6
7404
9
U48
8
7404
14
13
2
vcc
1
19
20
G1
G2
U45
81LS97
10
2
4
6
8
12
14
16
18
3
5
7
9
11
13
15
17
IBUS7
IBUS6
IBUS5
IBUS4
IBUS3
IBUS2
IBUS1
IBUSØ
vcc
20
U46
81LS97
MSB
C
D
E
F
H
J
K
LSB
L
2
4
6
8
12
14
16
18
G1
G2
3
5
7
9
11
13
15
17
DIN7
DIN6
DIN5
DIN4
DIN3
DIN2
DIN1
DINØ

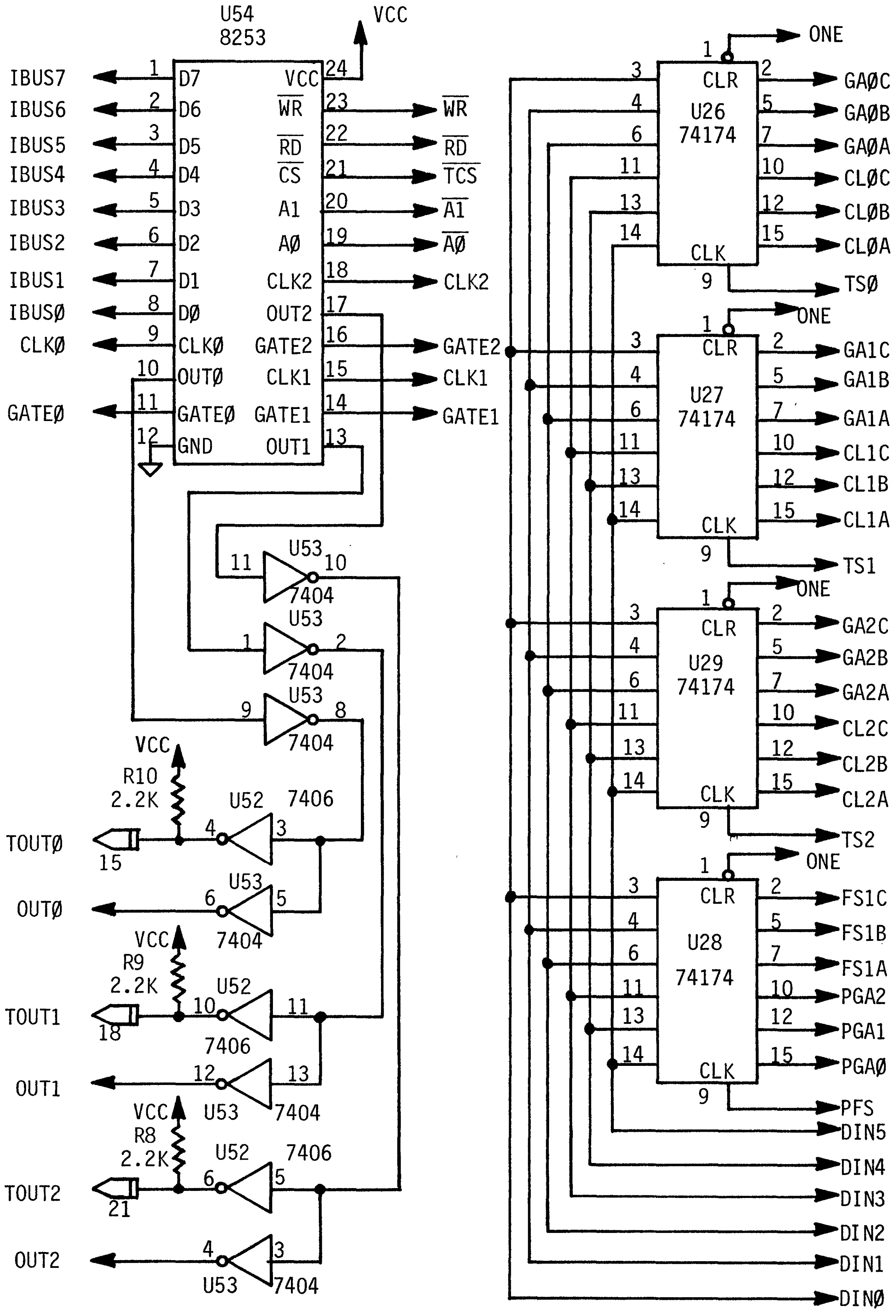
SHEET 2


A2
R
13
U48
7404
12
A1
P
11
U48
7404
10
AØ
N
3
U48
7404
4
1
C1
15
C2
U14
3
A
2
G1
13
B
14
G2
U57
Pin 6
Sh.2
74155
4
LEVSET
5
MSEL
6
TSØ
7
TS1
12
TS2
11
PFS
10
USEL
$\overline{A\emptyset}$
$\overline{A1}$
13
B
3
A
1
C1
U47
15
C2
2
G1
4
G2
U57
Pin 8
Sh.2
4
STØ
5
ST1
6
ST2
7
ST3
12
STS
VCC
Z2-6
2.2K
19
EXCL2
1SG
2
1
U50
7400
3
8
CLK1
QD
12
ONE
1
C/L
U51
QA
5
13
CLR
CLK2
6
74196
8
CLK1
QD
12
QA
5
ONE
1
C/L
CLK2
6
13
CLR
U25
74196
1
D3
2
D2
3
D1
4
DØ
U49
STROBE
74151
Y
5
XCL2
B
10
CLSELB
A
11
CLSELA
C
9
7
12
FXCL2
Z2-7
2.2K
VCC
VCC
R11
2.2K
$\overline{1SG}$
22
12
U52
7406
13
1
U55
7402
2
3
10
U34
4009
9
1SGA
Z2-5
2.2K
ONE
Z2-8
2.2k
Z3
5.6K
VCC
R13
5.6K
STS
1
19
$\overline{G1}$
$\overline{G2}$
INT
2
TEN
4
6
1SG
8
F1
12
OUT2
14
OUT1
16
OUTØ
18
U43
81LS97
3
5
7
9
11
13
15
17
2
3
4
5
6
7
8
IBUS7
IBUS6
IBUS5
IBUS4
IBUS3
IBUS2
IBUS1
IBUSØ

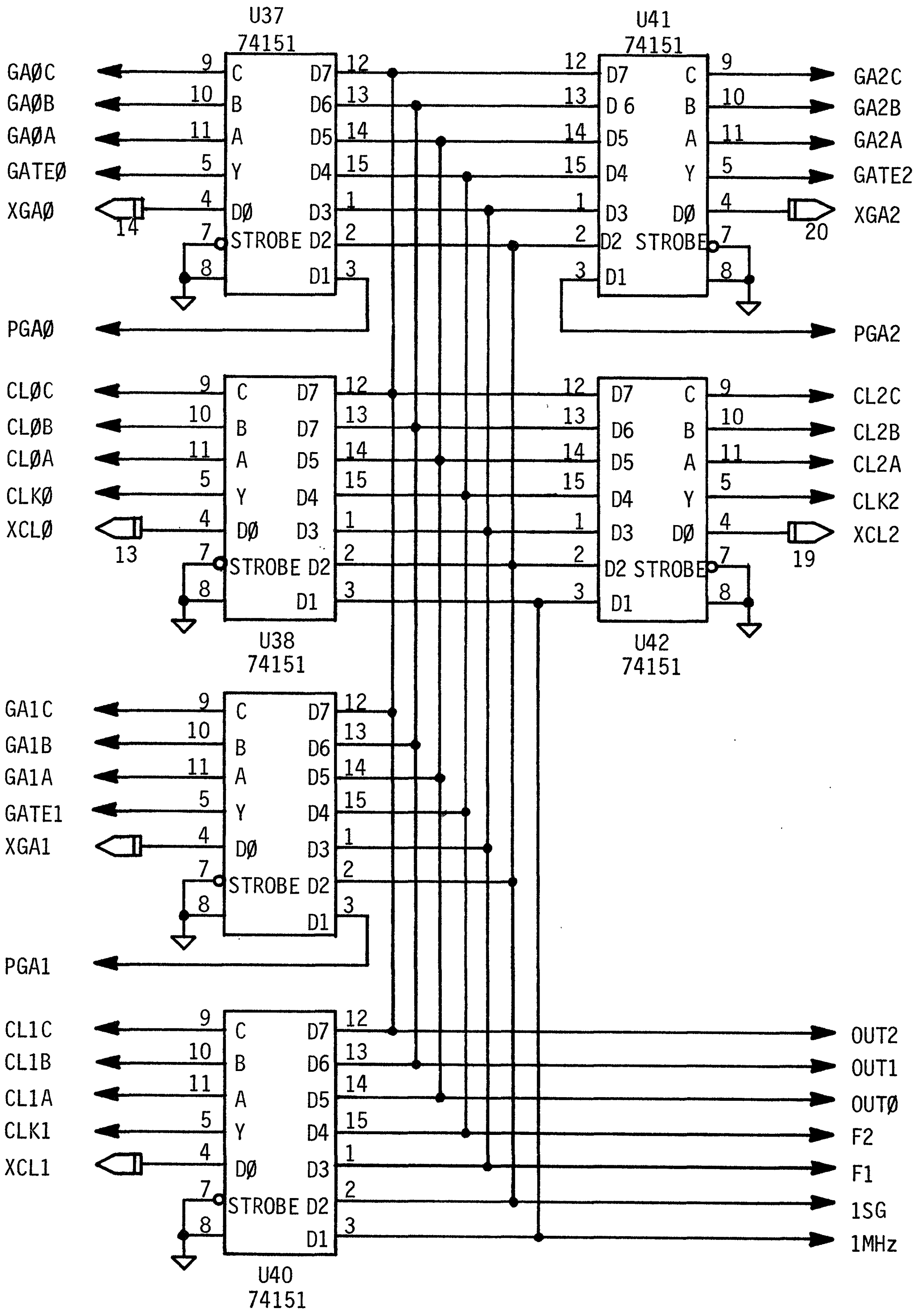
SHEET 3

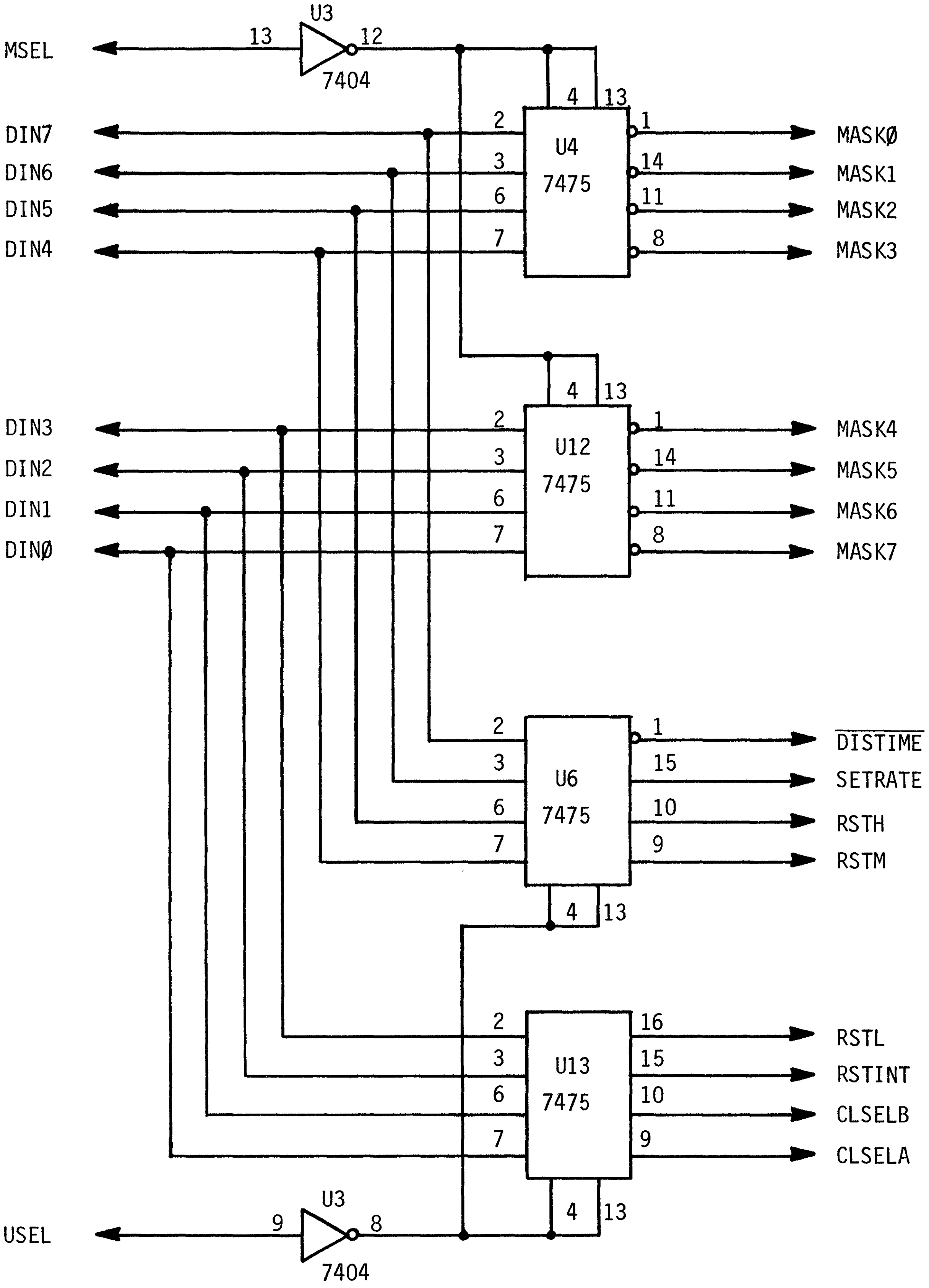

1SGA
VDD
Z1-11
2.2K
STØ
16
U9
4040A
9
7
6
5
3
2
1
3
10
13
15
6
16
4
Dis
2
5
9
11
14
7
IBUSØ
IBUS1
IBUS2
IBUS3
IBUS4
IBUS5
U20
4502
Inh
12
U33
Pins 2&3
Sh.1
10
CLK
VDD
Z1-13
2.2K
ST1
16
4
Dis
U8
Pin 11
Sh.1
11
RESET
4
13
12
14
15
1
U19
4502
Inh
12
U33
Pin 8
Sh.1
Z1-6
VDD
2.2K
ST2
16
U32
4040A
U17
4502
Inh
12
U33
Pins 9&
Sh.1 10
10
CLK
Z1-10
VDD
2.2K
ST3
U8
Pin 3
Sh.1
11
RESET
U18
4502
Inh
12

SHEET 4

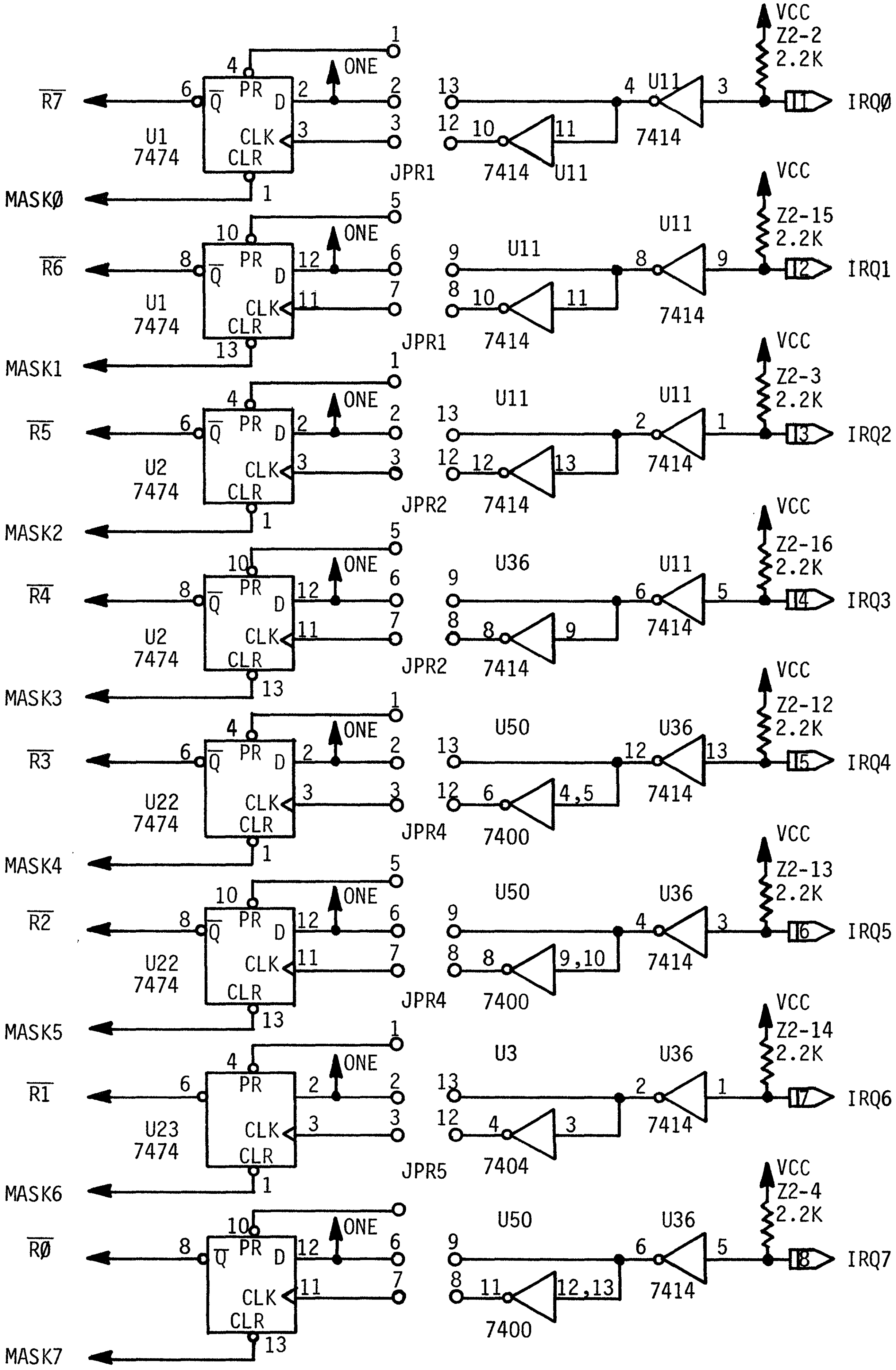

U54
8253
VCC
IBUS7
IBUS6
IBUS5
IBUS4
IBUS3
IBUS2
IBUS1
IBUSØ
CLKØ
GATEØ
1 D7
2 D6
3 D5
4 D4
5 D3
6 D2
7 D1
8 DØ
9 CLKØ
10 OUTØ
11 GATEØ
12 GND
VCC 24
WR 23
RD 22
CS 21
A1 20
AØ 19
CLK2 18
OUT2 17
GATE2 16
CLK1 15
GATE1 14
OUT1 13
WR
RD
TCS
A1
AØ
CLK2
GATE2
CLK1
GATE1
U53
7404
11
10
1
2
9
8
VCC
R10
2.2K
U52
7406
4
3
TOUTØ
15
U53
6
5
OUTØ
7404
VCC
R9
2.2K
U52
10
11
TOUT1
18
7406
12
13
OUT1
U53
7404
VCC
R8
2.2K
U52
7406
6
5
TOUT2
21
4
3
OUT2
U53
7404
ONE
U26
74174
CLR
CLK
GAØC
GAØB
GAØA
CLØC
CLØB
CLØA
TSØ
U27
74174
GA1C
GA1B
GA1A
CL1C
CL1B
CL1A
TS1
U29
74174
GA2C
GA2B
GA2A
CL2C
CL2B
CL2A
TS2
U28
74174
FS1C
FS1B
FS1A
PGA2
PGA1
PGAØ
PFS
DIN5
DIN4
DIN3
DIN2
DIN1
DINØ

SHEET 5

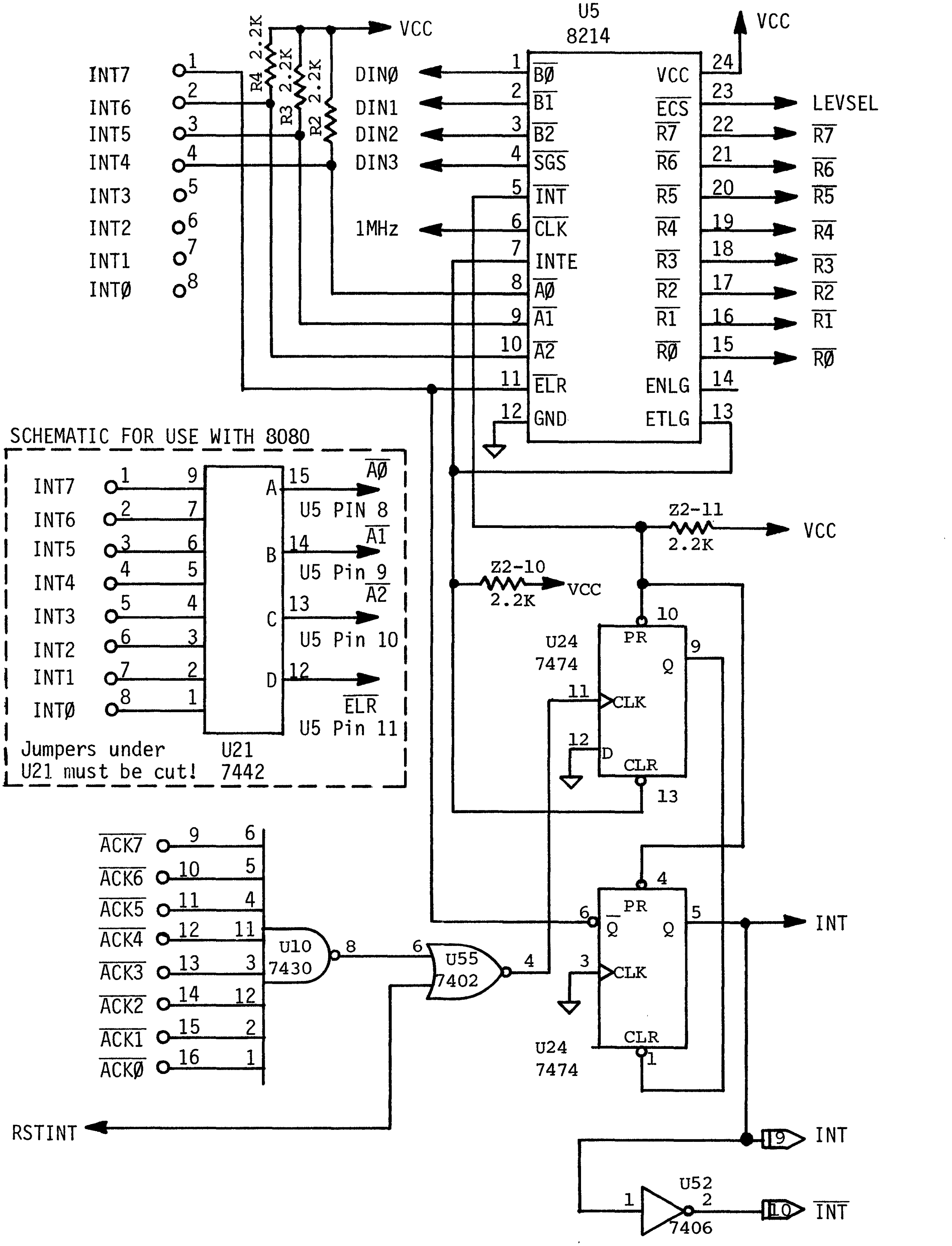

U37
74151
U41
74151
GAØC
GAØB
GAØA
GATEØ
XGAØ
14
PGAØ
C
B
A
Y
DØ
STROBE
D7
D6
D5
D4
D3
D2
D1
GA2C
GA2B
GA2A
GATE2
XGA2
20
PGA2
CLØC
CLØB
CLØA
CLKØ
XCLØ
13
CL2C
CL2B
CL2A
CLK2
XCL2
19
U38
74151
U42
74151
GA1C
GA1B
GA1A
GATE1
XGA1
PGA1
CL1C
CL1B
CL1A
CLK1
XCL1
U40
74151
OUT2
OUT1
OUTØ
F2
F1
1SG
1MHz


U3
13
12
7404
MSEL
4
13
2
3
6
7
U4
7475
1
14
11
8
DIN7
DIN6
DIN5
DIN4
MASKØ
MASK1
MASK2
MASK3
4
13
U12
7475
DIN3
DIN2
DIN1
DINØ
2
3
6
7
1
14
11
8
MASK4
MASK5
MASK6
MASK7
U6
7475
2
3
6
7
1
15
10
9
DISTIME
SETRATE
RSTH
RSTM
4
13
U13
7475
2
3
6
7
16
15
10
9
RSTL
RSTINT
CLSELB
CLSELA
4
13
U3
9
8
7404
USEL

SHEET 7


VCC
Z2-2
2.2K
1
4
ONE
PR
2
2
13
4
U11
3
R7
6
Q
D
I1
IRQØ
U1
7474
CLK
3
3
12
10
11
7414
CLR
JPR1
7414
U11
MASKØ
1
VCC
5
Z2-15
10
ONE
2.2K
U11
U11
R6
8
Q
PR
12
D
6
9
8
9
I2
IRQ1
U1
7474
CLK
11
7
8
10
11
7414
CLR
13
JPR1
7414
MASK1
VCC
1
Z2-3
4
ONE
U11
U11
2.2K
R5
6
Q
PR
2
D
2
13
2
1
I3
IRQ2
U2
7474
CLK
3
3
12
12
13
7414
CLR
JPR2
7414
MASK2
1
VCC
5
Z2-16
10
ONE
U36
U11
2.2K
R4
8
Q
PR
12
D
6
9
6
5
I4
IRQ3
U2
7474
CLK
11
7
8
8
9
7414
CLR
JPR2
7414
MASK3
13
VCC
1
Z2-12
4
ONE
U50
U36
2.2K
R3
6
Q
PR
2
D
2
13
12
13
I5
IRQ4
U22
7474
CLK
3
3
12
6
4,5
7414
CLR
JPR4
7400
MASK4
1
VCC
5
Z2-13
10
ONE
U50
U36
2.2K
R2
8
Q
PR
12
D
6
9
4
3
I6
IRQ5
U22
7474
CLK
11
7
8
8
9,10
7414
CLR
JPR4
7400
MASK5
13
VCC
1
Z2-14
4
ONE
U3
U36
2.2K
R1
6
PR
2
2
13
2
1
I7
IRQ6
U23
7474
CLK
3
3
12
4
3
7414
CLR
JPR5
7404
MASK6
1
VCC
Z2-4
U50
U36
2.2K
10
ONE
RØ
8
Q
PR
12
D
6
9
6
5
I8
IRQ7
CLK
11
7
8
11
12,13
7414
CLR
7400
MASK7
13

SHEET 8


U5
8214
VCC
VCC
R4 2.2K
R3 2.2K
R2 2.2K
INT7
INT6
INT5
INT4
INT3
INT2
INT1
INTØ
DINØ
DIN1
DIN2
DIN3
1MHz
B̅Ø̅
B̅1̅
B̅2̅
S̅G̅S̅
I̅N̅T̅
C̅L̅K̅
INTE
A̅Ø̅
A̅1̅
A̅2̅
E̅L̅R̅
GND
VCC
E̅C̅S̅
R̅7̅
R̅6̅
R̅5̅
R̅4̅
R̅3̅
R̅2̅
R̅1̅
R̅Ø̅
ENLG
ETLG
LEVSEL
R̅7̅
R̅6̅
R̅5̅
R̅4̅
R̅3̅
R̅2̅
R̅1̅
R̅Ø̅
SCHEMATIC FOR USE WITH 8080
INT7
INT6
INT5
INT4
INT3
INT2
INT1
INTØ
A
B
C
D
A̅Ø̅
U5 PIN 8
A̅1̅
U5 Pin 9
A̅2̅
U5 Pin 10
E̅L̅R̅
U5 Pin 11
Jumpers under
U21 must be cut!
U21
7442
Z2-11
2.2K
VCC
Z2-10
2.2K
VCC
U24
7474
PR
Q
CLK
D
CLR
A̅C̅K̅7̅
A̅C̅K̅6̅
A̅C̅K̅5̅
A̅C̅K̅4̅
A̅C̅K̅3̅
A̅C̅K̅2̅
A̅C̅K̅1̅
A̅C̅K̅Ø̅
U10
7430
U55
7402
PR
Q̅
Q
CLK
CLR
U24
7474
INT
RSTINT
INT
U52
7406
I̅N̅T̅