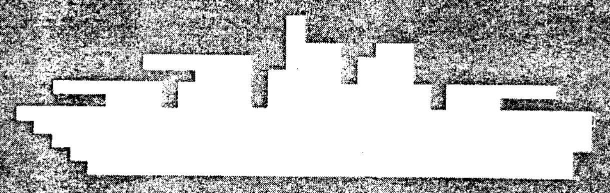
Sea Wolf



computer service manual

video game data library vol.VI

Distributed By Kush N' Stuff—60 Dillon Avenue, Unit D, Campbell, Ca. 95008 (408) 379-7180

Sea Wolf Computer Service Manual

For The Midway 8080 Microprocessor Game Series

A comprehensive analysis of the Sea Wolf game computer.

Technical Production by Laurel Publications

Table of Contents

SECTION 1 GENERAL INFORMATION

SECTION 2 THEORY OF OPERATION

SECTION 3 TROUBLESHOOTING PROCEDURES

SECTION I GENERAL INFORMATION

1.1 Introduction

1.1.1 Why Use A Microprocessor? The integrated microprocessor (often abbreviated simply as μP) has caused a revolution in the electronics industry for the contemporary μP now makes possible real computers for only a few hundred dollars. Ever since the advent of video games, designers have often wanted to use an actual, general-purpose type computer to operate their games and, in fact, a few custom games have been constructed using fairly powerful minicomputers. But it has never been possible to use such a computer for production video games simply because the cheapest available minicomputers cost thousands, rather than hundreds, of dollars. But today, with micoprocessor chips available for $20 and under, systems comparable in power to commercial minis can be built for maybe a tenth of the cost. Obviously the μP is a reality today and is rapidly changing the face of the electronics industry. But what of the reasons designers are so interested in applying them to video games? After all, video games have been successfully manufactured for a number of years using the random-logic type of architecture, so why change now?

Actually, there are quite a few reasons, but most of them boil down to two considerations: economics and design flexibility. But before getting into all of this, let's take a couple of steps backwards and talk about earlier techniques for implementing video game designs.

1.1.2 Random Logic Vs. Processor Control. Before μP systems burst upon the scene, video games were invariably designed using a dedicated "computer" custom built from random-logic elements which are assembled in such a way as to generate both the video images displayed on the screen and the various rules by which these images are controlled. The typical random-logic video game computer consists of a large number of SSI and MSI devices wired together to produce the desired effects. Since only small and medium scale device are used, this type of game computer tends to require a large amount of circuitry and PCB real estate to generate even the simplest displays and algorithms.

While the random-logic approach clearly works, it has several serious drawbacks which become particularly evident when this technique is compared with operating a game via processor control. Probably the most significant disadvantage is the tremendous amount of development time required to generate each new game since an entirely new computer is required for every distinctly different game theme. Furthermore, bugs which show up in the field at a later date can become a serious problem and one which may be extremely difficult to correct since all the boards must be recalled from all over the country (or even all over the world) and the custom printed circuit board modified by cutting traces and adding jumpers to implement the engineering change.

Now, compare this with the way a microprocessor game is developed. First, the central processing unit (CPU) is designed. Although the CPU may not contain a very large chip count as compared with a random-logic design, be aware that many of the chips found on a CPU board are LSI circuits which contain many more individual components than the simpler MSI varieties. So, although the physical size of the CPU circuitry may be smaller, the overall complexity of the system is far greater.

Microprocessor manufacturers not only build the μP itself, but they also generate associated componentry necessary to the operation of the microprocessor chip. A good example is the 8216 Bidirectional I/O Driveers used to control the 8080 system date bus. These chips are absolutely necessary to the operation of the 8080 and the manufacturer of the devices also supplies the schematic for their connection. Furthermore, almost all μP manufacturers also supply a developmental system which the engineer can often use to "breadboard" his design. In fact, sometimes all that the engineer really needs to do is to write his own custom program and eliminate any unnecessary circuitry from the developmental system. Actually, the system is already half designed before the video game designer sits down and starts work and this can obviously result in a tremendous savings of engineering time.

So, after playing with the developmental system for a while, the designer creates the actual CPU design for the video game while taking into consideration the types of functions the game system will need to perform. One of the major considerations is the fact that the CPU must work for a large number of different game themes. If the system is in fact designed according to this consideration, it should be useful for an extended period of time and this is the main reason the μP architecture is so attractive, for once the company has a viable design, a number of advantages are quickly realized. Obviously, the most important is the fact that the development costs required to get the initial system up can be spread out over the entire run of machines which might last for several years. Furthermore, only a relatively small number of parts need be stocked at a considerable savings to the manufacturer. And, since the manufacturer knows a great many CPU boards will be sold during the production run, a large number of them can be produced at once which greatly reduces the costs involved in making high quality printed circuit boards. For example, one of the larger costs associated with manufacturing PC boards is setting up for drilling. Obviously, the more boards that are drilled, the cheaper the set-up cost for each board will be.

But there are many more advantages with the μP architecture other than savings realized in engineering and the actual construction of the PCBs. Prior to shipping, all boards must be thoroughly tested and since a single CPU system is used for a great many different game themes, there is no need for test technicians to become familiar with a new random-logic design every month or so.

1.1.3 Advantages of Software Control. The foregoing examples have all been hardware-oriented advantages, but what we are really dealing with here is a software-oriented system wherein the software itself offers a great many benefits which may or may not be immediately obvious. For example, once a reliable CPU has been created, implementing another game theme mainly becomes a matter of writing new software and this can often be accomplished in a tenth of the time that would ordinarily be required to assemble a new hard-wired, random-logic design. Also, if bugs show up in the new game, these can be worked out on a software level in such a way that the program is corrected and shipped out at only a fraction of what it would take to recall a large number of random-logic PCBs and modify them to correct the problem.

Software systems are quite flexible and this aspect offers a number of advantages not possible with random-logic designs. For instance, several objects can be moved on the CRT simultaneously using a μP without increasing system complexity. However, were a designer to try to do this with a random-logic game, he would quickly discover that he needs a separate motion circuit for each different image to be moved. Were he to want very much action on the CRT, he could easily end up with a game not feasable to produce with random-logic elements.

Software controlled video games are more easily modified in the field and this could conceivably be of great significance depending on the actual circumstances. Sometimes a game looks very good in a test situation, but receives little play in the field. Were the manufacturer not able to correct this situation, that company could encounter trouble. However, if the game happened to be a software oriented system, the manufacturer could still rectify the situation even after the games had been shipped from the factory for all that company needs to do to correct the matter is to write a more acceptable game program, burn it into a number of PROMs and send them out free of charge to the disappointed operators.

And lastly, once people become used to a particular μP system, technicians everywhere will be able to repair games using that system. Also, since most μPs function similarly, there is a great deal of positive transference from one system to another meaning that technicians will already be acquainted with the general architecture of the system before they sit down and become familiar with the specifics of that system.

Now that we have discussed some of the reasons that microprocessors are so quickly gaining in popularity for video games (and a tremendous number of other systems as well), it is time to turn our attention to the specifics of the Sea Wolf game system. In the following pages, we will discuss the general architecture of the Midway 8080 CPU mother board and the Sea Wolf game board as well as the specific operation of all phases and components of the system itself. In conclusion, we will present a number of troubleshooting approaches for dealing with specific problems and problematic areas as well as providing a generous amount of actual troubleshooting data so that video game technicians with ordinary troubleshooting equipment will be able to fully explore the circuitry found in this game.

1.2 A Typical Game Sequence

1.2.1 The Attract Mode. The condition of the game prior to the start of the game is known as the attract mode and this mode is characterized by the displays in Figures 1.2.1-1 and 1.2.1-2. During the first part of this mode, wave action is visible and the words "Game Over" are flashed on and off in the center of the screen. Additionally, the CRT displays the score from the last game as well as the "High Score". After a few moments of this the display changes. At this point the player's score from the last game, the words "Sea Wolf" and "Insert Coin" appear instead. Simultaneously, a ship appears from one edge of the screen and begins to travel across the screen. After the ship has moved a bit, the computer fires a torpedo at it and sinks the ship. This last sequence is repeated until a player inserts a coin.

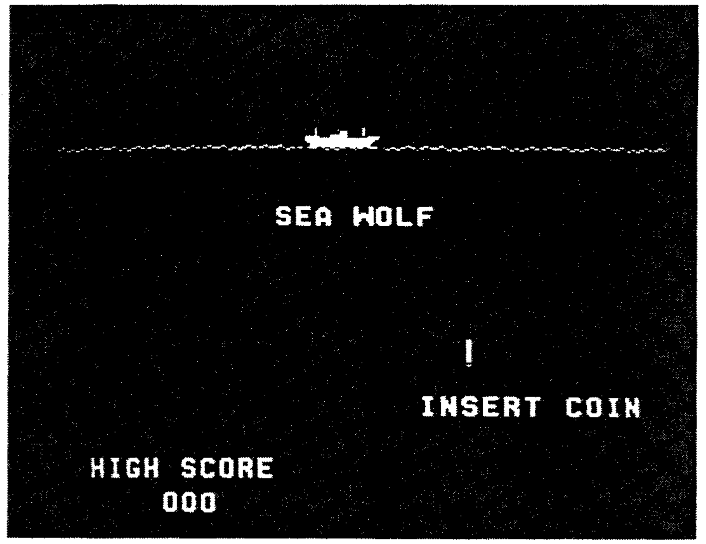


FIGURE 1.2.1-1 A Typical Attract Mode Display

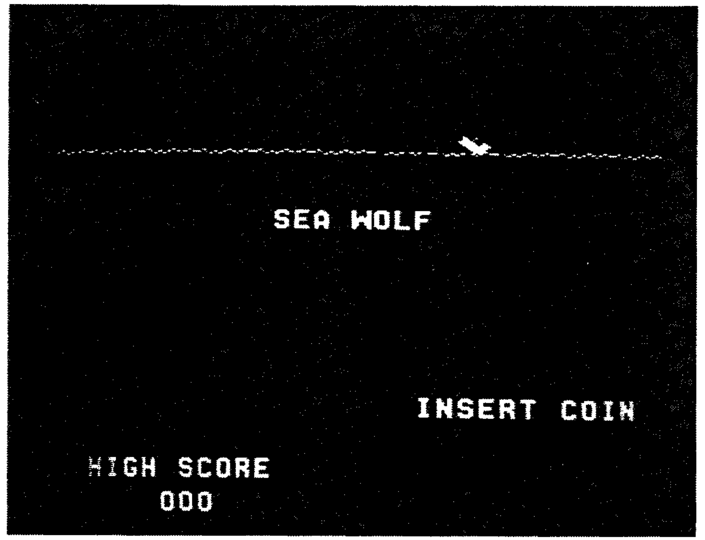


FIGURE 1.2.1-2 A Typical Attract Mode Display

1.2.2 The Play Mode. After a coin is deposited, the computer enters the play mode which is characterized by the displays in Figures 1.2.2-1 and 1.2.2-2. At this point all the displays are changed and the sonar sound commences. The game time counter is displayed in the lower portion of the CRT and it begins to count down. The player's score is displayed and it counts up in increments of 100 depending on which ship is hit. Additionally, a group of mines appears in the approximate middle of the screen and these mines drift across the screen slowly. Pressing the fire button releases a torpedo which is accompanied by a "hissing" noise. If the torpedo hits a ship or mine, the words "Zap" or "Wham" appear and that image is exploded and disappears. The player is allowed four torpedoes per volley before the torpedo tubes must be reloaded. The game continues in this fashion until the game timer reaches zero.

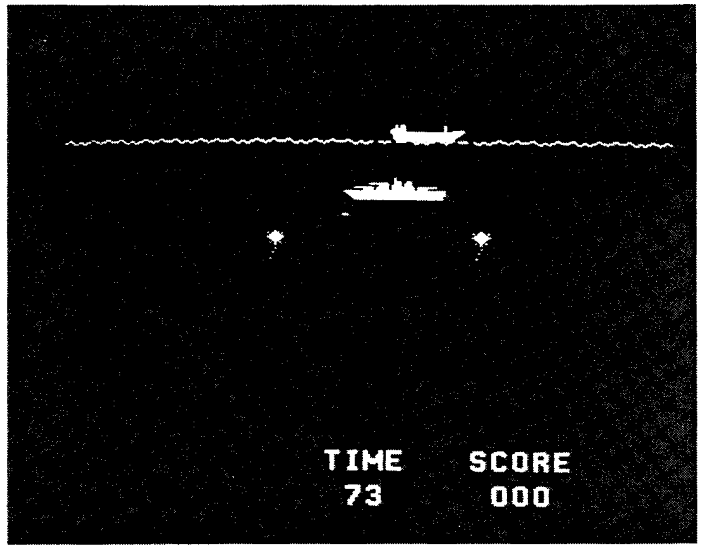


FIGURE 1.2.2-1 A Typical Play Mode Display

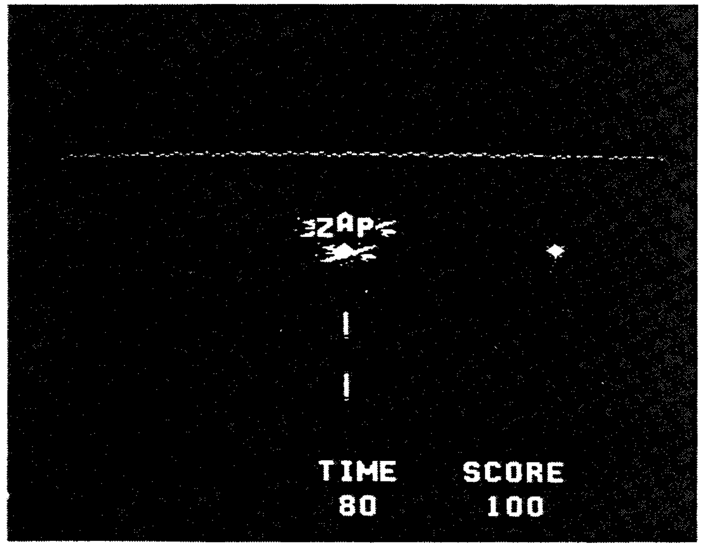


FIGURE 1.2.2-2 A Typical Play Mode Display

1.3 Electrical Adjustments

1.3.1 Introduction. In this system, there are four main categories of electrical adjustments and these are: (1) TV monitor adjustments, (2) power supply adjustments, (3) operator adjustments and (4) sound system adjustments.

1.3.2 TV Monitor Adjustments. These adjustments are located on the TV monitor itself where the four most commonly used ones (A-D) are found on a panel accessable from the rear of the monitor. Vertical linearity and height are adjusted via pots located on a printed circuit board in the monitor and the yoke is adjusted at the rear of the picture tube. More detailed information concerning the locations of these adjustments is found in the Motorola monitor manual.

A. HORIZONTAL HOLD: Adjustment is indicated only if the picture is off center horizontally, if the images appear warped or if the entire picture is broken up into a series of diagonal lines (Figure 1.3.2-1). Adjust for a stable, horizontally centered picture.

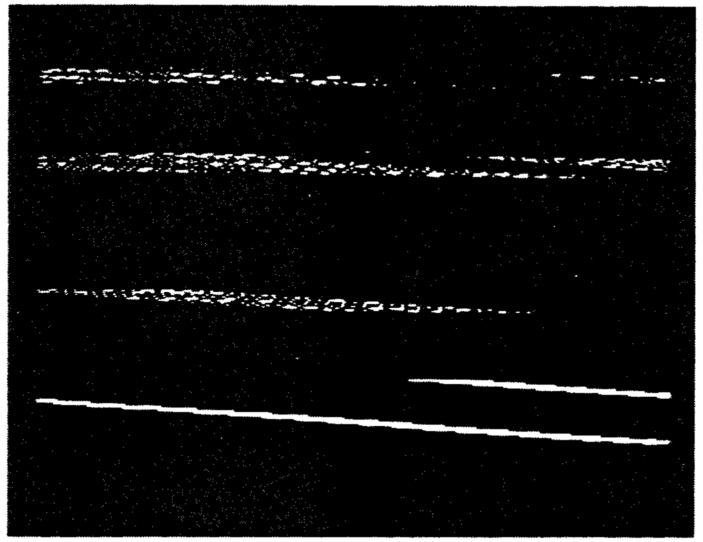


FIGURE 1.3.2-1 This Display Indicates Horizontal Hold Adjustment Is Needed

B. VERTICAL HOLD: Adjustment is indicated if the picture appears to be rolling up or down the CRT or if the picture is stable, but not centered vertically. Adjust for a stable, vertically centered picture.

C. BRIGHTNESS: Brightness is adjusted before contrast. Adjust for a dark background.

D. CONTRAST: Adjust so the white images are as bright as possible against the dark background without being blurred.

E. VERTICAL LINEARITY: Adjust only if the images at the top of the CRT appear to be compressed. If adjustment is indicated, adjust the V LIN pot so that the spacing of the raster lines is not compressed at the very top of the CRT.

F. VERTICAL HEIGHT: Adjust only if the entire picture appears compressed vertically. Be aware that there is some interdependence between the linearity and height adjustments.

G. YOKE: Yoke adjustment is indicated only if the entire picture is off center and the H HOLD adjustment has insufficient effect. Adjust both yoke rings simultaneously for optimum centering of the image area within the borders of the CRT.

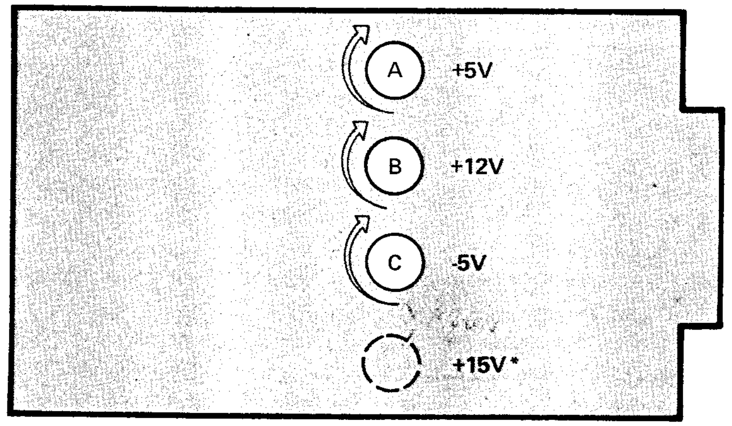
1.3.3 Power Supply Adjustments. The following power supply adjustments are performed at the power supply PCB (P.C. 80-901) and the specific adjustment locations are indicated in Figure 1.3.3-1. Arrows indicate direction of increase.

A. +5 VOLT SUPPLY: Connect a VOM or DVM to power PCB edge connector pin 11 (or 12) and ground and adjust pot A for a reading of exactly +5 volts.

B. +12VOLT SUPPLY: Connect test instrument to power PCB edge connector pin 1 (or 2) and ground and adjust pot B for a reading of exactly +12 volts.

C. -5 VOLT SUPPLY: Connect test instrument to power PCB edge connector pin 5 and ground and adjust pot C for a reading of exactly -5 volts.

D. +15 VOLT SUPPLY: Perform this adjustment only if 3235 clock driver is found at mother PCB location C5. Connect test instrument between pin 8 of the power PCB edge connector and ground and adjust pot D for a reading of exactly +15 volts.



*This adjustment performed only if I.C. 3235 is used in mother PCB location C5.

FIGURE 1.3.3-1 Power Supply Adjustment Locations

1.3.4 Operator Adjustments. These adjustments are performed by setting the individual on/off switches found in a DIP package at game PCB location G4. The following adjustments affect (A) game length, (B) coin(s) per play(s) and (C) the point at which extended play time is awarded.

A. Game Time. To adjust the total number of time units per game, set switches 1 and 2 according to the table below.

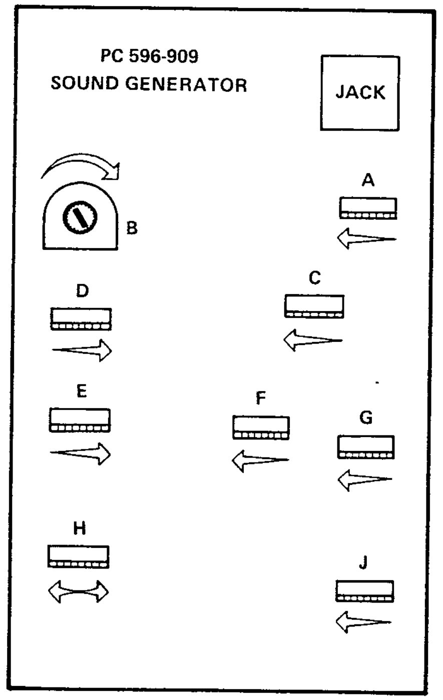
TIME	SW 1	SW 2
60 UNITS	ON	ON
70 UNITS	OFF	ON
80 UNITS	ON	OFF
90 UNITS	OFF	OFF

B. Coins/Plays. This procedure determines the number of coins required for the desired number of plays. For example, if you want the player to receive two plays for a single coin, set switches 3 and 4 to ON and OFF respectively.

COINS/PLAYS	SW 3	SW 4
1 COIN/1 PLAY	ON	ON
1 COIN/2 PLAYS	ON	OFF
2 COINS/1 PLAY	OFF	ON
2 COINS/3 PLAYS	OFF	OFF

C. Extended Play. This procedure adjusts the score at which the player is awarded extended play time. If no extended play is desired, see Note 1.

SCORE	SW 6	SW 7	SW 8
2000	OFF	ON	ON
3000	ON	OFF	ON
4000	OFF	OFF	ON
5000	ON	ON	OFF
6000	OFF	ON	OFF
7000	ON	OFF	OFF
8000	OFF	OFF	OFF



NOTE: CW AND CCW ARE DEFINED AS VIEWED FROM POT FACE.

FIGURE 1.3.5-1 Sound System Adjustment Locations

NOTE 1: With switches 6, 7 and 8 in the ON position, no extended play will be awarded.

NOTE 2. The High Score Reset Button is located on the upper left front of the cabinet. With DIP switch 5 set to the ON position, the high score may be reset by depressing the button, but only while there is not a game in progress. With switch 5 in the OFF position, the high score cannot be reset.

1.3.5 Sound System Adjustments. A number of separate adjustments located on the sound PCB affect both the volume and tone of the various game sounds. The locations of these adjustments are indicated in Figure 1.3.5-1 and these trimpots are adjusted according to the table below.

A. SONAR VOLUME: To increase, rotate pot CW.
B. MASTER VOLUME: To increase system volume, rotate CW.
C. SONAR RATE: To increase pulse rate, rotate CW.
D. TORPEDO VOLUME: To increase volume, rotate pot CW.
E. DIVE SOUND VOLUME: To increase volume, rotate pot CW.
F. SONAR TONE: To raise sonar pitch, rotate pot CCW.
G. SHIP EXPLOSION VOLUME: To increase volume, rotate CCW.
H. DIVE TONE: To vary tone, rotate pot as desired.
I. MINE EXPLOSION VOLUME: To increase volume rotate CCW.

1.4 System Architecture

1.4.1 System Subassemblies. Figure 1.4.1-1 shows the basic electrical subassemblies required for this system. The system has a separate power supply PCB which provides nearly all the regulating and other functions required to power the integrated circuits and other components. The power supply is connected by the wiring harness to the CPU mother board (also known as the mother PCB), the game board and the player's controls. The player's controls are located on an external control panel mounted to the front of the cabinet and this subassembly consists mainly of the periscope assembly and the fire button. The player's controls are connected only to the game PCB and are the main inputs to that part of the system. The game PCB is considered to be a peripheral device which does some processing of the input signal flow in terms of "smoothing" motion generated by the mother PCB. The game PCB "talks" to the mother PCB over the multiplexed data bus, however the CPU writes to the game PCB over the output data bus. Both busses are unidirectional (at least between the boards) meaning that information can pass in one direction only.

The mother PCB contains the CPU, the memory and a number of other important devices. The CPU is controlled by program information stored within the ROMs (Read Only Memories) and it uses the inputs from the players' controls entering the CPU over the multiplex data bus to change playfield images and actions according to a set of rules stored in the program. All the images including the numbers, letters, etc. are stored in the ROMs as well.

After the CPU has performed the required operation to the data, the information is stored in the RAM memory located on the mother PCB. Although some of this RAM is used as a "scratch pad" memory, most of it is used to store the next frame to be displayed on the TV screen. In other words, the RAM memory contains a pattern of bits each of which corresponds to one of the displayable points on the CRT.

The game PCB is essentially a peripheral device and as such, it gathers information (i.e. the players' control settings) to be processed by the CPU. However, this part of the system also contains the shifter, a subcircuit used to "smooth" out the jerky motion generated by the mother PCB. The shifter allows single-bit increments of motion to be generated from the larger 8-bit shifts produced by the CPU. The game PCB also produces the various sounds required by the system and controls them so they are heard only at the correct moments in the game.

1.4.2 Signal Flow. There are a number of busses and other significant signal lines which connect the mother and game PCBs to each other and enable data and control signals to be passed from one board to the other. These busses and lines are the multiplexed data bus, the output data bus, the address bus, the sample line and the reset line.

First of all, a definition of the word "bus" is in order. By bus, we mean a port of eight parallel data lines where each line represents one bit of information. The total of eight bits across the bus is known as one byte.

1.4.3 The Multiplex Data Bus. The multiplex data bus is always unidirectional and it carries information only from the game PCB to the mother PCB. Four types of information are found on this bus at different times: left player controls, right player controls, operator instructions and recirculating shifted data. When the CPU reaches a place in its program where information about the settings of the left player's controls is needed, it sends out an instruction over the address bus which is decoded by the game PCB and used to select the binary information representing the settings of the left controls.

1.4.4 The Output Data Bus. The operation of this bus is slightly more complicated than that of the multiplex bus because the output data bus is bidirectional on the mother PCB, between the 8080 CPU and the bus drivers, but it is unidirectional between the mother PCB and the game PCB. On the mother PCB, this bus is used both to enter data from other circuits (i.e. the RAM memory) into the CPU so the CPU can digest the new data and also to output the digested data from the CPU back into memory or to some other device. But between the mother PCB and the game PCB, the data on this bus can flow only from the mother to the game PCB. In this case, the CPU – after "crunching" on the data – uses the output bus to write digested data into the shifter circuit of the game PCB. After being shifted, this data returns to the mother PCB via the multiplex data bus.

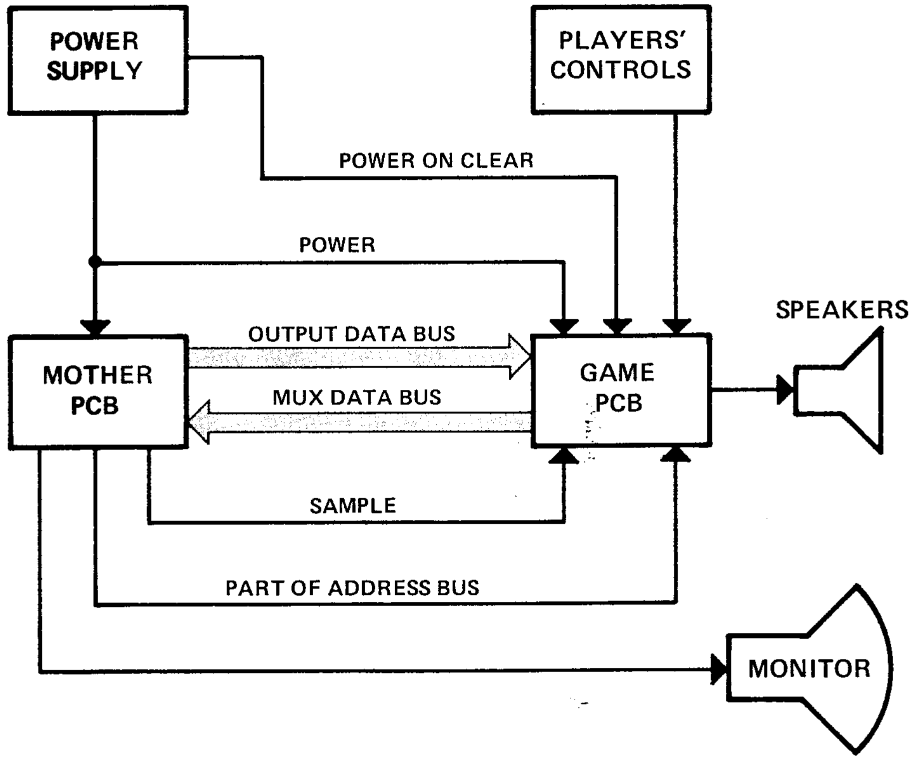


FIGURE 1.4.1-1 System Block Diagram

1.4.5 The Address Bus. Part of the address bus (address bits A^8, A^9 and A^{10}) is peeled off the 16-bit wide mother PCB address bus and sent to the game PCB and be aware that this is a unidirectional bus on both boards. On the mother PCB, the full 16-bit wide address bus is used to address memory (both RAM and ROM), however the part of the address bus taken to the game PCB is used to address that peripheral for the type of data (player control settings, operator instructions, etc.) called for by the CPU program.

1.4.6 The SAMPLE Line. The CPU uses the sample line to notify the game PCB that the CPU is writing information to that peripheral. This is necessary because the game PCB must know that the data being placed out on the output data bus is for the game PCB. The sample line is a single control line and it can only contain one bit of information.

1.4.7 The Reset Line. The reset line originates from the power PCB which has a power-on-clear circuit that outputs a signal when the game is powered up. This signal is used to clear all the latches on the game PCB so the game begins with all invalid information cleared from the latches.

1.5 Mother PCB Architecture

1.5.1 Introduction. This discussion of mother PCB and 8080 architecture is intended for orientation purposes only so these areas are covered on a very superficial basis as the main thrust of this manual is toward understanding how the game PCB works in this system. Obviously, the game PCB will make little sense if it is removed from the context of the system as a whole, so some information must be provided about the operation of both the mother PCB and the 8080 itself. However, the following information is not intended to convey complete understanding of the operations of these parts of the system. If you require more detailed information, please be aware that an entire manual has been devoted to this subject.

The mother PCB block diagram (Figure 1.5.1-1) illustrates the major subcircuits found on the mother board, the busses that interconnect these circuits and the direction of signal flow for each type of information. The major sections in this area are sync and timing, RAM memory, RAM data latches, video output, CPU input selector, I/O drivers, status latch, interrupt flip-flop, the CPU, the ROM selector and the ROM memory itself. The major busses are the address bus, the instruction bus, the bidirectional data bus, the output data bus, the RAM data bus and the multiplex data bus.

1.5.2 Sync and Timing. This section generates the master timing signal CLOCK which is processed in several ways to generate a two-phase clock ($\phi1$ and $\phi2$) for the entire system which is available in both normal TTL voltage levels for the majority of the circuitry and also in a "high voltage" (12 volt) level for the 8080 CPU. The signal CLK is also divided down by a divided chain to provide clock submultiples used both in the generation of the sync signals and to read out the contents of the RAM memory in such a way that the information is displayed coherently on the CRT.

1.5.3 The RAM Memory. All the RAMs in this section are 4K by 1-bit dynamic devices. Since they are 1-bit devices, it takes eight of them to construct a byte. And, since they are dynamic devices, they must constantly be refreshed, otherwise their contents will be lost. However, this type of RAM is constructed in such a way that reading the information out of memory is sufficient to refresh the memory. A total of 8K bytes of RAM are required by this system and the RAM is addressed both by clock submultiples as well as by the address bus. The address bus is used to address parallel data in and out of RAM for use by the CPU, however the clock submultiples read out the contents of the RAM a byte at a time into the shift register in the video output section and thence onto the CRT. In passing, we would like to mention that the stack (and a few other types of data as well) are also stored in the RAM memory on a temporary basis. These areas of the memory are reserved for use by the CPU and are not read out onto the CRT.

Note that only chip H8 is drawn as it is actually connected. All the other RAM chips are connected in the same way, but they are drawn in a "condensed" manner to make the drawing easier to read.

1.5.4 RAM Data Latches. These latches simply take RAM data and hold it stable until the CPU is ready to accept the information.

1.5.5 Video Output. This section is composed only of a single shift register which accepts a byte of parallel data from RAM and clocks it out in a serial format so each bit may be displayed sequentially as the electron beam scans the CRT.

1.5.6 CPU Input Selector. This section selects between four separate types of signals: RAM data, multiplex data bus information, instruction bus information and restart instructions where the selection process is controlled by the output of the status latch. In other words, when the CPU reaches a place in the program where control setting input data from the game PCB is required, the status latch puts out a signal which selects the multiplex data bus information in the input selector section.

1.5.7 I/0 Drivers. These devices control the direction of data flow between the CPU and the rest of the mother board circuitry under command from the CPU itself. In other words, when the CPU is writing data to a peripheral it sends out a signal to the drivers which causes them to shut off the input lines and enable only the output lines from the CPU. However, when the CPU needs to read data, it sends out a "read" instruction to the drivers which causes them to turn the output lines off and allow the inputs through.

1.5.8 Status Latch. This is just a regular hex latch used to indicate to the rest of the system the type of operation the CPU will be performing during the present machine cycle. During the very first part of the machine cycle, the status is latched and held for the entire cycle.

1.5.9 Interrupt Flip-flop. This small section provides a "real-time" (i.e. a non-software dependant) clock for functions such as the game timer.

1.5.10 The ROM Selector. This section decodes part of the address bus and uses the resulting information to turn on the proper ROM chip when the information in that chip is needed by the CPU.

1.5.11 The ROM Memory. Whereas information can be written into and read out of the RAM memory, data can only be read from the ROMs. ROMs are useful in this application because their contents are non-volatile. Were the game program (and image information) held within RAM and the power accidentally turned off, the entire program would have to be reloaded. The ROM memory stores both the game program as well as all the images used.

1.5.12 The 8080 CPU. The 8080 is the "brain" of the system for it provides all the control logic required to operate the game. Under program control, the CPU processes data which is provided both by memory and by the game PCB. The program is stored in the ROM memory and it directs the CPU by giving it an instruction directing the CPU to perform the required process to the data. Each instruction usually carries with it the address of the memory location where the data is to be found. The sequence of program steps is controlled by the program counter which is incremented at each instruction and simply steps through the program in a sequential order.

The illustration on page 6 is a more-or-less simplified block diagram of the significant internal sections and the required external devices (denoted by dotted lines). One of the most important items to keep in mind is that the data bus (D_0 – D_7) is bidirectional both in the 8080 itself and

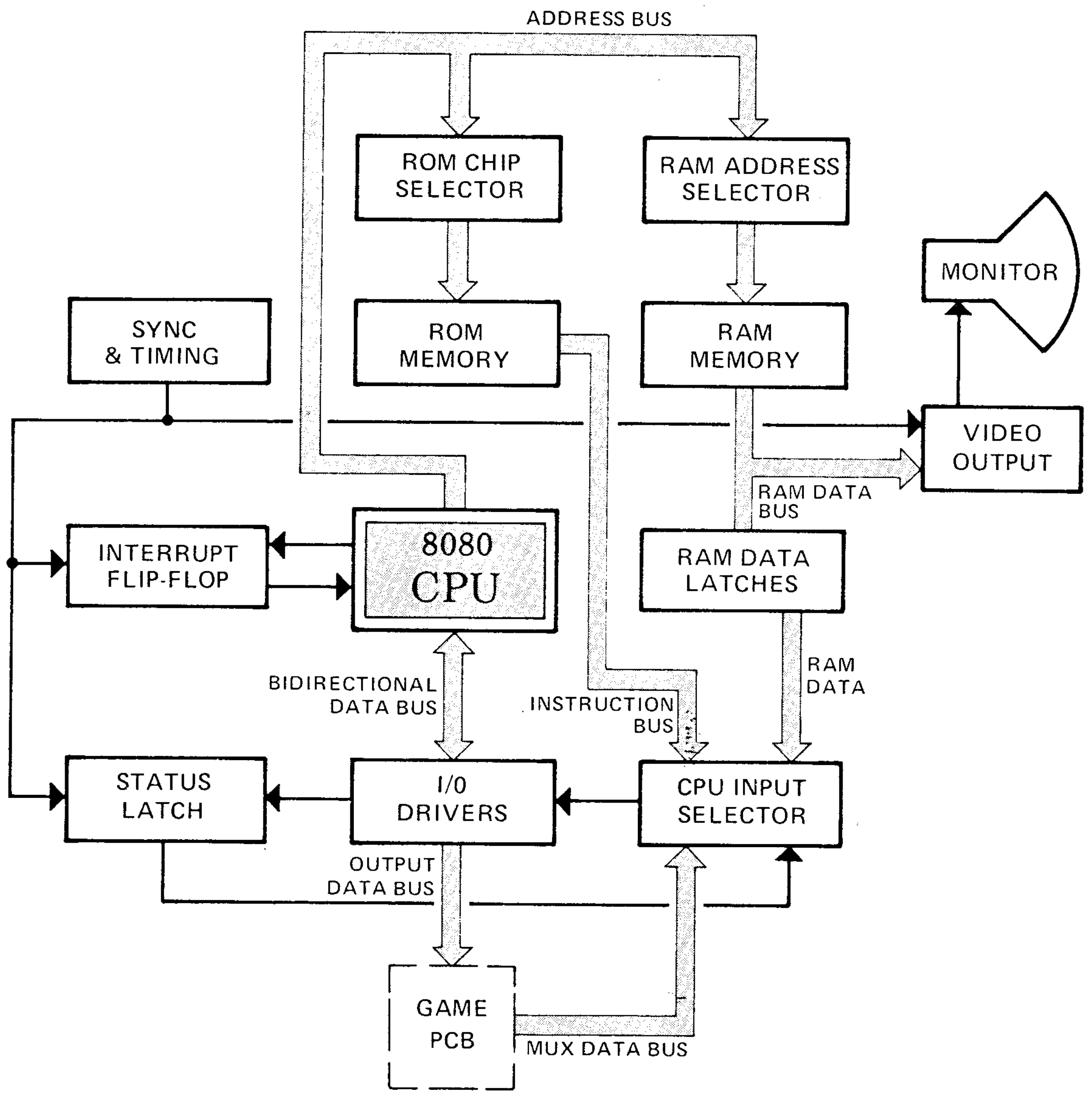


FIGURE 1.5.1-1 Mother PCB Block Diagram

between the 8080 and the bus drivers. In other words, data can flow from memory into any one of several areas (i.e. the instruction decoder or the general purpose registers) and it can also flow in the opposite direction (i.e. from the accumulator to memory). The address bus, however, is not bidirectional. The data on these lines can only go from the program counter or stack pointer to memory.

A good way to become familiar with the way a CPU chips works is to follow it as it goes through a complete cycle. For example, we want to perform the function LDA which means load accumulator. It takes one byte to define this instruction and two additional bytes to define the location in memory of the data to be loaded since the address bus is a 16-bit wide ($A_0 - A_{15}$) bus.

The most convenient point at which to begin is after the CPU has been reset. At this point, the program counter is reset to zero and it places the next name of the first program step (0000) to be executed out to memory over the address bus and fetches the instruction held at this location. The instruction returns to the CPU via the data bus and enters the instruction decoder. The decoder instructs the other parts of the 8080 what to do and when to do it by enabling and disabling various chunks of circuitry according to the decoded instruction and clock signals $\phi 1$ and $\phi 2$. At this point, the control circuitry takes over and the CPU does the function as instructed which might consist of fetching data from memory by the program counter to that memory location. Once it has acquired all the data it needs, it performs the indicated function which – in this case – consists of loading the data into accumulator A. However, it can also perform one of many other functions such as swapping internal registers or adding two sets of data from memory together. After the specified operation has been completed, the result is placed back out onto the data bus and a control signal is generated which informs the proper external device (either memory or a peripheral) that the data is meant for it. The control signal is used to dump the data into memory or into the interface unit of a peripheral device. In the case of Sea Wolf, this means entering the information into the latches of the shifter section of that peripheral (the game PCB).

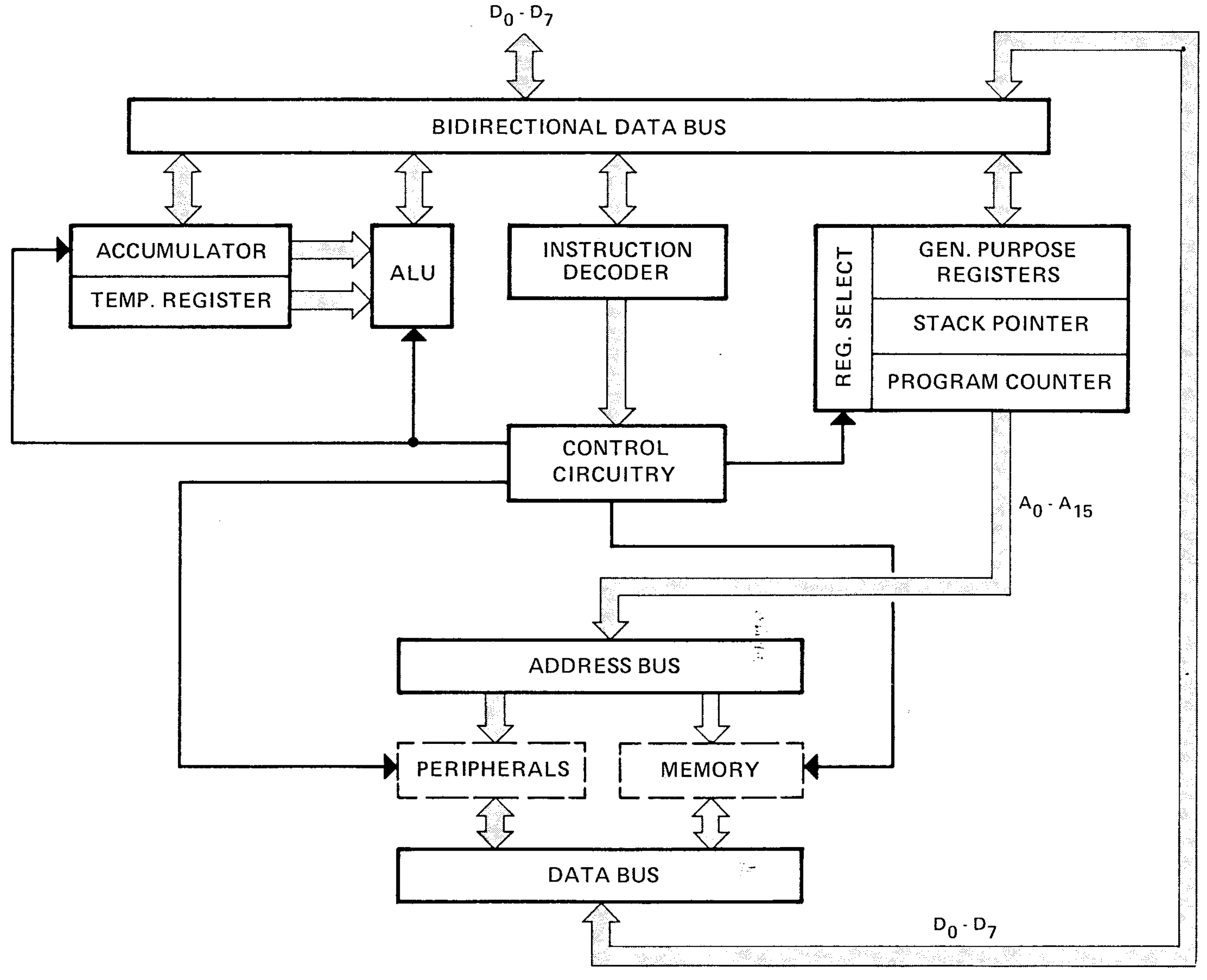


FIGURE 1.5.12-1 8080 CPU Block Diagram

So much for a typical instruction cycle. Before moving on, a few other areas of the 8080 deserve brief attention and these are the registers, the stack pointer, the program counter, the memory, the peripherals, the accumulator and the ALU. Since we have already discussed the instruction decoder and the control circuitry, these areas require no further attention.

The general purpose registers are used as a "scratch pad" memory for the CPU. These registers provide a convenient place for storing intermediate results as the CPU is solving a problem. There are six general purpose registers available to the user and two temporary registers accessable only to the machine itself.

The stack pointer is an area of memory in which the CPU may store information or the program counter so the processor knows where to return after performing a subroutine call. Since the stack is held in the RAM memory, it is essentially unlimited which means almost any number of subroutines may be nested within each other. The stack operates in a first-in, last-out manner which means that the first data entered is placed at the very bottom of the stack.

The program counter is just a regular 16-bit binary counter which outputs memory addresses to find instructions, data or whatever else is stored at the particular location in memory. The program counter is also used to address data from the interface registers of peripheral devices.

The memory in the case of Sea Wolf consists of the 8K RAM memory and 4K of ROM all located on the mother PCB. All memory is addressed the same regardless of what kind of memory chip is used. As we have already mentioned, the ROM memory holds the program and the images while the RAM memory is used to hold the next data to be displayed on the CRT as well as providing a place for the CPU to store information on a temporary basis.

The peripherals in a general purpose computing system may include many types of devices used to collect and enter or to output and display data. Some of these include printers and TV terminals which are used to output and display data or keyboards and custom electronic subsystems which are used to input information. However, in the case of Sea Wolf the only peripheral is the game PCB itself. In reality, there are two output peripherals found on the game PCB and one input peripheral. The CPU writes data into both the digital video shifter which controls the images and into the analog sound section which generates and controls the sound. The CPU receives data from the players' controls via the game PCB.

The accumulator and temporary registers are a set of ordinary registers except that they are expressly dedicated to the ALU. Since the ALU must be capable of performing an operation of two arguments (i.e. adding two numbers together), it must have two different registers.

The ALU or arithmetic logic unit is the circuit used to perform the actual arithmetic operation on the number (or numbers). The ALU can also be used to provide the equivalent of logic functions such as ORing two numbers together.

1.6 Game PCB Architecture

1.6.1 General Functions. The game PCB contains a number of input and output peripherals which collect or output data. In fact, there are three output peripherals and one input peripheral where the functions of these subsections are (1) to collect data from the players' controls and operator adjustments, (2) to provide smooth image motion, (3) to generate and control the various sounds and (4) to trip the coin counter. The general relationships of these circuits and the directions of data flow are indicated in the following block diagram (Figure 1.6.1-1). In this part of the system, all data enters via the output data bus (D_0-D_7).

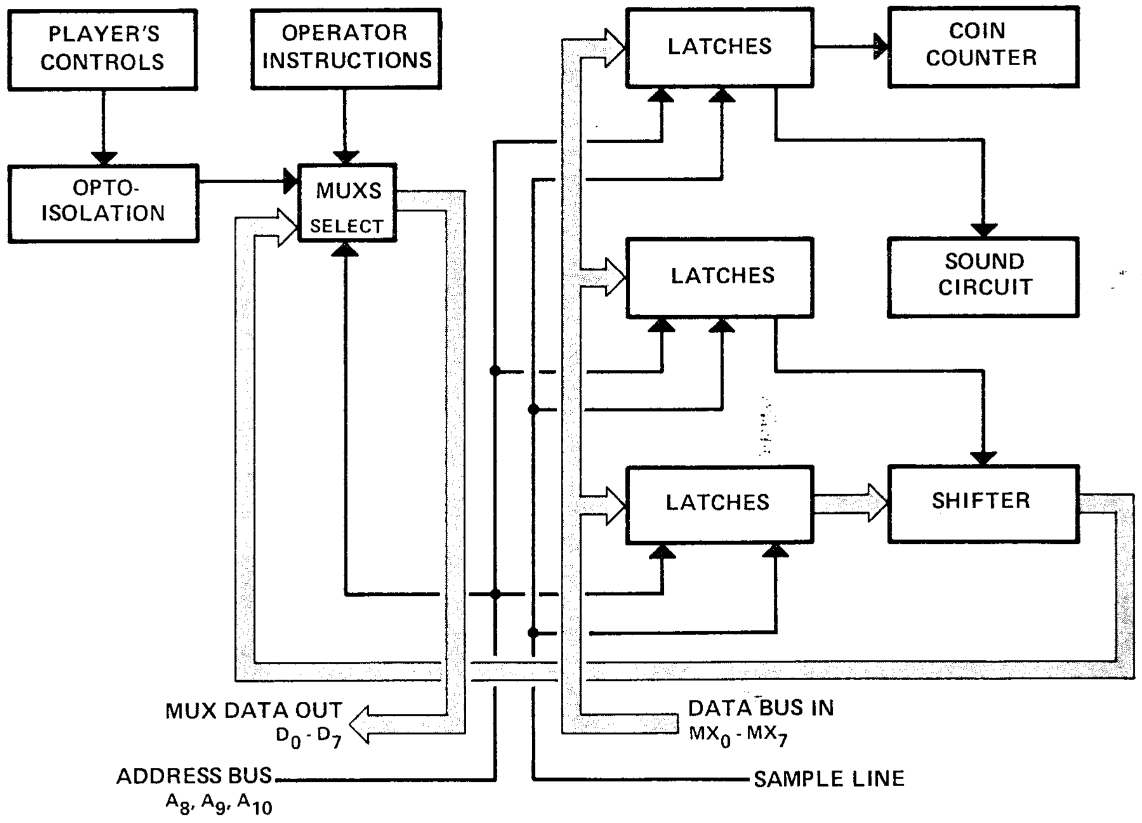


FIGURE 1.6.1-1 Game PCB Block Diagram

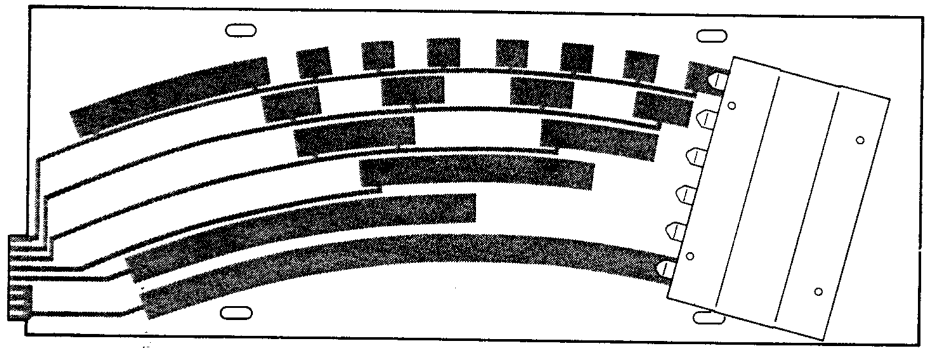


FIGURE 1.6.2-1 The Periscope Printed Circuit Board

1.6.2 Player's Controls. The player's controls comprise the main input to the system and they consist only of a fire button and a periscope assembly. The periscope is a complicated mechanical device, however it is quite simple electrically. Electrically speaking, the only relevant parts are a pc card underneath the assembly over which a wiper travels to generate a 5-bit code representing the position of the periscope. The position controls the point at which the torpedo is aimed and fired.

1.6.3 The Opto-isolators. These devices are used only to isolate the players' controls from the rest of the circuitry for protection reasons so that no possibly destructive electrical potential can occur. The opto-isolators do not affect the content of the signal line in any way. See Section 1.7.2 for more details concerning the operation of the opto-isolators.

1.6.4 Operator Instructions. These instructions are entered by the operator at the DIP switches located at game PCB position C1 and are used to define the length of the game, the number of plays per game credit and the number of coins per game credit.

1.6.5 The Multiplexers. The multiplexers are included so that the CPU on the mother PCB can request the type of information it needs when it needs it.

There are four different types of information entered into the multiplexers and these are: (1) right player control settings, (2) left player control settings, (3) operator instructions and (4) shifted data.

For example, when the CPU reaches a place in its program where it needs to know where the player is aiming the torpedo, it sends out an instruction over the address bus which is connected to the select lines of the multiplexers. The code on the address bus at this time selects the periscope control setting inputs and these signals then appear at the multiplexer outputs. The binary code representing the control setting is then passed to the CPU over the multiplex data bus for use by the CPU. By sending out slightly different codes over the address bus, the CPU can request information about the position of the fire button or the settings of the operator adjustment switches.

1.6.6 The Latches. The main function of all these latches is to hold the information coming in on the data bus stable until it can be used by one of the output devices (the shifter, coin counter or sound generator). The data is latched in two passes. During the first pass, the lowest 7 bits are latched and the next eight bits are latched in the second pass. The D_0 bit (the LSB) is dropped in the process due to the construction of the circuit and because it is not needed. The latches are clocked by a special signal which results from NANDing the SAMPLE line with address bit A^{10}. The combination of these two signals allows the CPU to define the output port (the latches in this case) to which it is writing.

1.6.7 The Shifter. The shifter is controlled by one of the latches (E6 to be specific) which simply latches up the three LSBs (D_0, D_1 and D_2) of the data word coming in over the output data bus and uses them to control the shifting process. The correct word coming in on the bus is selected by the combination of the SAMPLE line and address bit A^9 which determines the output device the CPU wants to write into.

Motion in this system is not achieved by the familiar "slipping counter" process normally associated with random-logic video games, rather it is generated by moving data words around in memory. As we have mentioned before, the RAM memory contains a reflection of the CRT in that what is in RAM at one instant will appear in the next frame of the monitor scan.

The problem is that the CPU can only move data in memory in jumps of eight-bit blocks due to the construction of the memory system. So a circuit known as the shifter is employed to convert this single eight-bit jump into a series of eight single-bit "hops." For example, let's say we have the data word 00011011 which represents part of an image stored in ROM and we want to move the image to the right on the CRT. Assume also that this word is presently stored in memory location 147 (see Figure 1.6.7-1a). Were a shifter not employed in the game PCB, the CPU would take the word 00011011 and simply dump it into the next memory location, which is 148 (Figure 1.6.7-1b).

Now, when the shifter is employed to move the image data word an entirely different process results and this is illustrated by Figure 1.6.7-1c and d. In this case, the data word is first taken from the RAM memory location 147 and entered into the shifter circuit in the game PCB which shifts it up by one bit only and the shifted word is then rewritten back into RAM memory so that the last bit of the word that was in location 147 is now the first bit in location 148. Now that you understand what this process looks like in memory, we shall illustrate the same process using representations on the TV monitor screen (following page).

1.6.7-1a 1.6.7-1b

BEFORE SHIFT		AFTER SHIFT	
RAM LOCATION	IMAGE DATA WORD	RAM LOCATION	IMAGE DATA WORD
145	XXXXXXXX	145	XXXXXXXX
146	XXXXXXXX	146	XXXXXXXX
147	00011011	147	XXXXXXXX
148	XXXXXXXX	148	00011011
149	XXXXXXXX	149	XXXXXXXX

WITHOUT DATA SHIFTER

WITH DATA SHIFTER

145	XXXXXXXX	145	XXXXXXXX
146	XXXXXXXX	146	XXXXXXXX
147	00011011	147	X0001101
148	XXXXXXXX	148	1XXXXXXX
149	XXXXXXXX	149	XXXXXXXX
BEFORE SHIFT		AFTER SHIFT	

1.6.7-1c 1.6.7-1d

FIGURE 1.6.7-1 Shifter Circuit Operation (In Memory)

Each location in memory is represented by a vertical column on the CRT. Although we have illustrated only seven columns, be aware that there are actually thirty-two columns from one side of the CRT to the other (Figure 1.6.7-2). Notice also that we have labelled two of the columns in Figures 1.6.7-3 as 147 and 148 and these numbers represent those areas of memory just as in Figure 1.6.7-1.

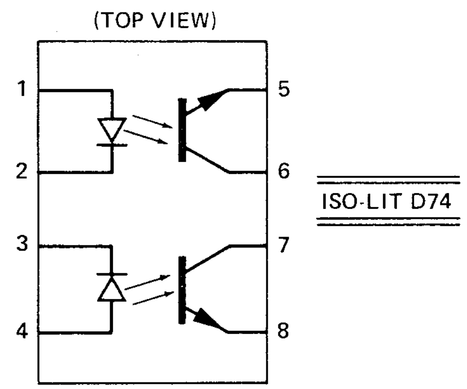
To illustrate what would happen without the shifter circuit, we have included Figures 1.6.7-3a and 3b. Since the CPU can move data in memory only in 8-bit "leaps", these two figures show that the part of the image (and hence the entire image) will jump over to the right by one whole column each time the CPU tries to move the image. However, with the shifter circuit in operation, the process is actually very smooth and this method is illustrated in Figures 1.6.7-3c and 3d. Whereas the entire word was shifted a whole column in Figures 3a and 3b, the word in Figures 3c and 3d is shifted to the right only one bit. You can easily see that the last bit of the word in column 147 in Figure 1.6.7-3c is now in the first bit position of column 148 in Figure 1.6.7-3d. The result is smooth, realistic motion.

1.7 Operation of Specific Game PCB Components

1.7.1 Introduction. Before delving into the actual theory of operation of the Sea Wolf game PCB, a few relevant comments need to be made about the function and operation of the various components from which the game PCB is constructed. Hopefully, the following information will facilitate both the comprehension and troubleshooting of the game PCB circuitry.

Fortunately, the componentry used in the game PCB is ordinary stuff, especially in comparison to the devices located on the mother PCB. Probably, you are already familiar with some of the game PCB devices (the multiplexers, latches and decoders), however a few others (the opto-isolators, Schmitt triggers and data shifters) may not be quite as familiar so we will stress these last items to a greater extent.

1.7.2 Tbe Opto-isolators. Opto-isolators do just what their name implies – they electrically isolate one component from another using the optical characteristics of a light emitting diode (LED). The reason for using these devices is to eliminate any electrical potential which might occur between the control panel and the game computer. An undesirable potential of this nature can arise from a number of causes. A well-known one stems from player accidentally or intentionally releasing large static discharges to the machine which can play havoc with the digital circuitry.



PIN NO.	FUNCTION
1	ANODE
2	CATHODE
3	CATHODE
4	ANODE
5	EMITTER
6	COLLECTOR
7	COLLECTOR
8	EMITTER

LED CHIPS ON PINS 2 AND 3
PT CHIPS ON PINS 6 AND 7

FIGURE 1.7.2-1 The Opto-isolator

Other causes include poor ground connections and operators inadvertently shorting out the wiring while repairing or collecting the machine. The opto-isolator transmits only the data from the control panel, not the actual electrical signal.

Actually, opto-isolators have been around for quite some time, even before the advent of today's remarkable semi-conductor techniques. The earliest types used either incandescent or neon lamps as light sources coupled with cadmium or lead photo-resistors as sensors. Contemporary opto-isolators generally use a gallium arsenide infrared emitting LED as the light source and a silicon photo-transistor as the sensor. The source and sensor are totally enclosed in the device package so the light emitted is not visible and the source cannot be triggered by ambient light.

The type of opto-isolator used in this application is either an ISO-LIT D74 or its equivalent, the MCT-6. This type of device contains two opto-isolators in a single package used to isolate separate control inputs. The control inputs enter the device on pins 2 & 3 or 3 & 4 where they go directly to the LED chip. The outputs come from pins 5 & 6 or 6 & 7 from the phototransistor chips. It is interesting to note that this device is not monolithic; the LED and phototransistor are mounted on different chips.

The control inputs are normally HI and when the control associated with that input is operated, the line drops to ground and illuminates the LED which emits infrared light. This light is sensed by the phototransistor, which then outputs a pulse of constant potential. A separate supply voltage (V_{LED}) is used to operate the LED and notice that this voltage is connected to the side of the LED opposite from the control input. So, when the control input drops to GND, this voltage is allowed through the LED and turns it on.

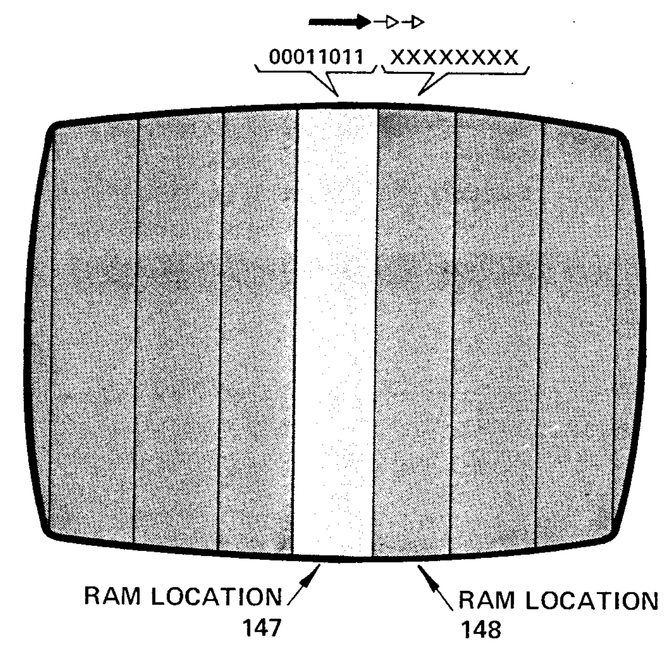


FIGURE 1.6.7-3a Without Shifter Circuit, Prior To Movement

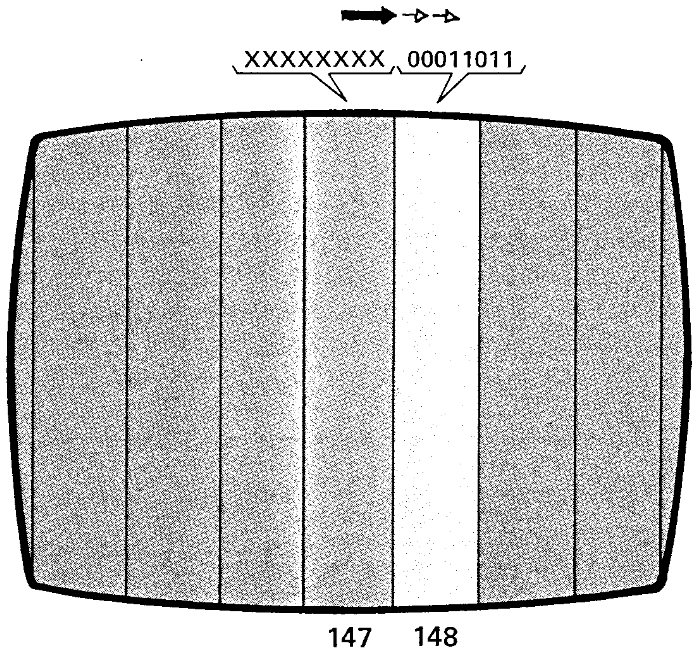


FIGURE 1.6.7-3b Without Shifter Circuit, After Movement

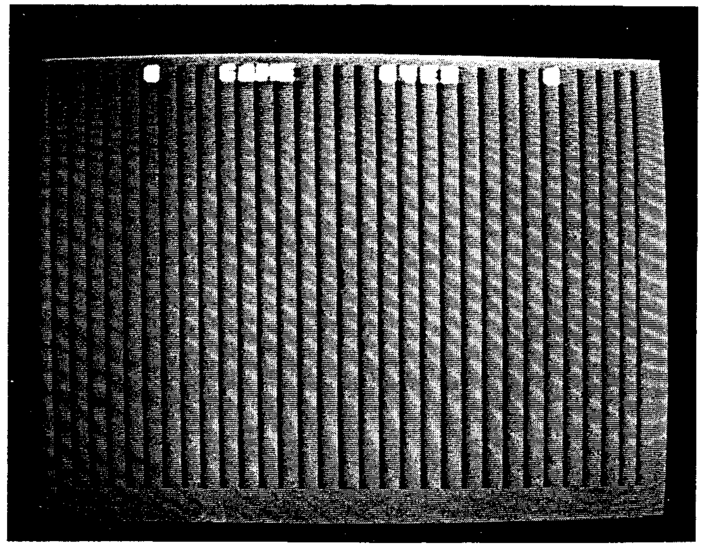


Figure FIGURE 1.6.7-2 Actual Display Columns On CRT

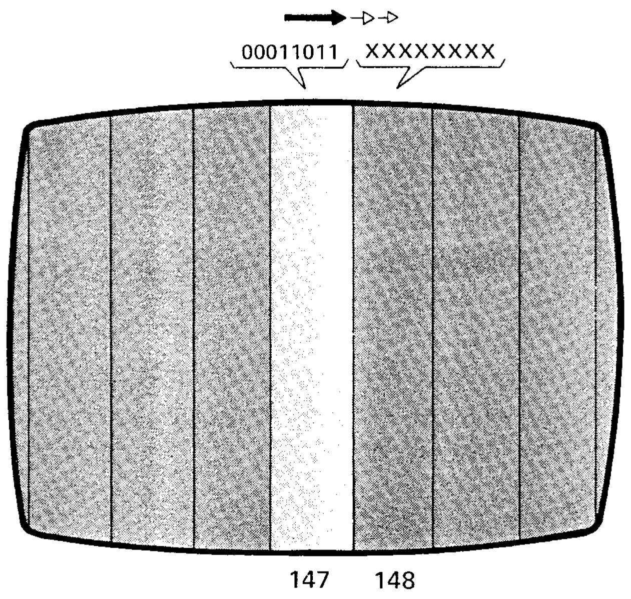


FIGURE 1.6.7-3c With Shifter Circuit, Prior To Movement

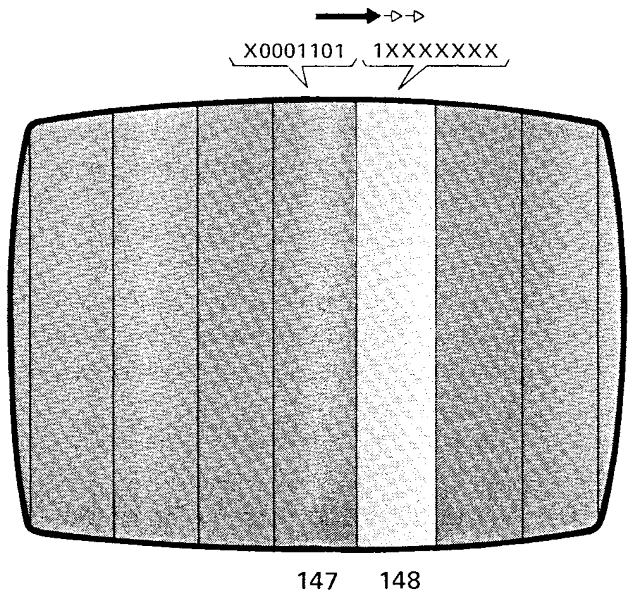


FIGURE 1.6.7-3d With Shifter Circuit, After Movement

1.7.3 The Schmitt Triggers. These devices operate essentially like 7404 inverters in that the input waveform is inverted, however Schmitt triggers also are used to square up the waveform so it becomes more acceptable to the digital circuitry. Although it is a good idea to use Schmitt triggers in this sort of application, their use is not absolutely necessary and a 7404 may be used temporarily if the 7414 Schmitt trigger is not readily available.

Figure 1.7.3-1 shows how the Schmitt Trigger cleans up a messy input waveform into a nicely squared up signal.

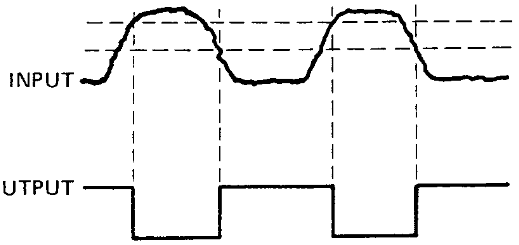


FIGURE 1.7.3-1 Schmitt Trigger Operation

1.7.4 The Multiplexers. All the multiplexers found on the game PCB are 74153 dual four-to-one devices which means there are two of the same four-input multiplexers in the same chip. In general, multiplexers are used when it is necessary to select one data line from a group of lines so the desired data can be sent to other devices for further processing.

Figure 1.7.4-1 shows the simplest multiplexer, the single two-input. In this case, input data lines A and B are selected by the state of select signal S. When S is HI, the top AND gate is enabled and it passes whatever signal is present at the selected data input. Simultaneously, the HI at S is inverted LO to disable the lower gate and lock out the other data line. But if S drops LO, the top gate is disabled and the second data line is enabled through to the output.

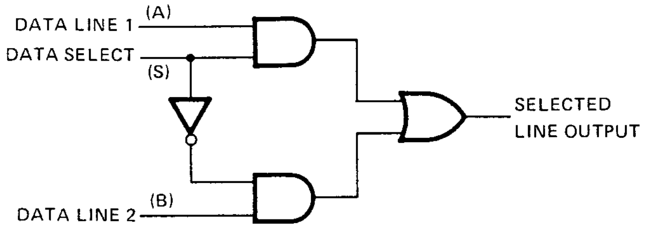


FIGURE 1.7.4-1 Basic Multiplexer Logic Diagram

A single select line can only be in one of two states: HI or LO, hence it can select only one of two signals. But if we have four signals from which we must choose one, a second select line is required. Two selects then have the capability of four states: 0 0, 0 1, 1 0 and 1 1 where each different code may be used to select one of four data lines. Figure 1.7.4-2 shows how the simplest four-input multiplexer is constructed and notice that there are two selects (S_0 and S_1) which controls the four input lines (I_0 through I_3).

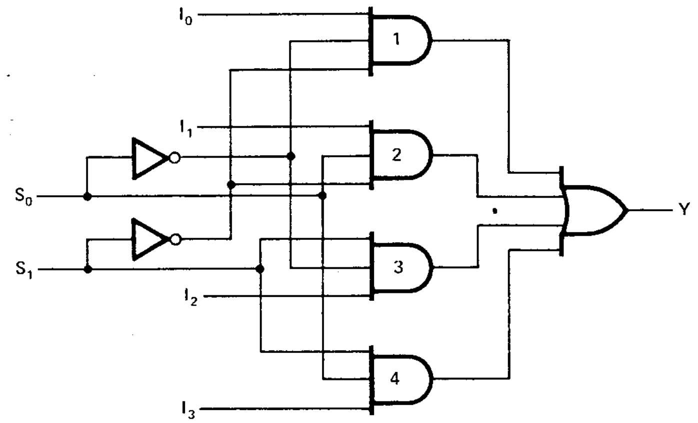


FIGURE 1.7.4-2 Four-Input Multiplexer Logic Diagram

If both S_0 and S_1 are LO, gates 2, 3 and 4 are disabled since a LO S_0 disables Gates 2 and 4 and a LO S_1 disables Gates 3 and 4. Gate 1 is receiving two HIs because the LOs at S_0 and S_1 are inverted first before being connected to it.

Consequently, Gate 1 is enabled and it passes whatever signal is at I_0. The second count occurs when S_0 and S_1 are HI and LO respectively which enables Gate 2 to pass input I_1.

Actual integrated four-input 74153 multiplexers are a bit different. First of all, there are two complete multiplexers per 16-pin DIP. Also, an additional enable input is provided which is connected to all the gates so the entire chip may be turned off. Generally, this enable input is known as a strobe and it may be used to turn off some of the multiplexers in a circuit while others are allowed to continue to operate.

1.7.5 The Latches. The latches found in this game are very simple devices which consist mainly of four flip-flops. The latches are used in this application to hold data in a stable configuration until it is needed by other devices. When the data has been used and is no longer needed, the latch is cleared. This quad latch can store up to four bits of data at a time, so it can retain a number as large as 1 1 1 1. In reality, the quad latch consists of four single-bit latches connected by common clock and clear lines. Any number of latches may be ganged together to handle larger numbers.

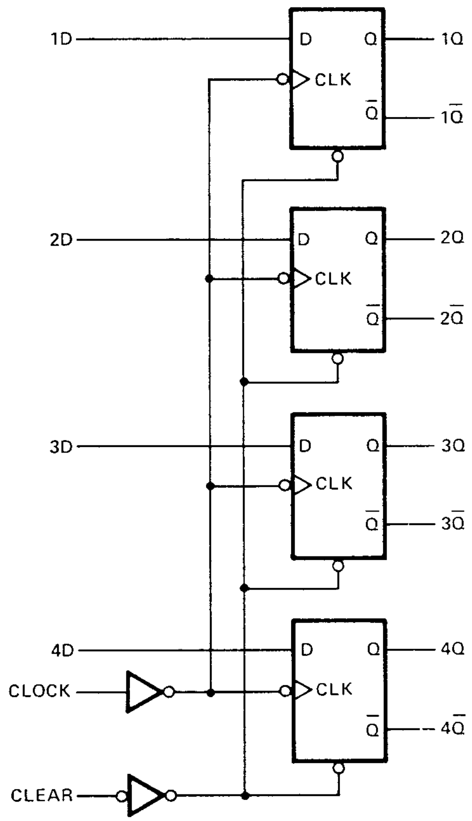
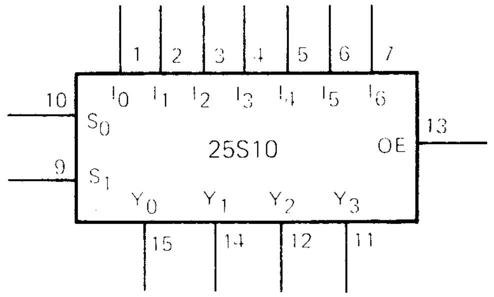


FIGURE 1.7.5-1 74175 Quad Latch

The data to be latched is entered at the D inputs to the flip-flops and clocked out when needed by the signal present at the CLK input. Since both the Q and $\overline{Q}$ outputs of the flip-flops are brought out to chip pins, both the true data and its inverse are available. A pulse on the clear line simply resets the Q outputs of all the flip-flops to zero. Since D-type edge-triggered flip-flops are used for the 74175, the data is latched into the device and appears at the devices outputs simultaneously when the flip-flop sees the rising edge of the clock pulse.

1.7.6 The Data Shifters. The data shifter is used so a four-bit section of a seven-bit word can be selected according to the states of the select lines. In the game PCB, this feature is used to enable smooth motion. By shifting the word a bit at a time, motion in increments of one clock pulse each can be achieved.

A truth table can be used to explain the operation of the device much more easily than words. In truth table 1.7.6-1, you can see that if the select code (at inputs S_0 and S_1) is 00, inputs I_6, I_5, I_4 and I_3 appear at outputs Y_0, Y_1, Y_2 and Y_3. However, if the select code is incremented to the next count (10), inputs I_5, I_4, I_3 and I_2 appear at the device



S_0	S_1	Y_0	Y_1	Y_2	Y_3
0	0	I_6	I_5	I_4	I_3
1	0	I_5	I_4	I_3	I_2
0	1	I_4	I_3	I_2	I_1
1	1	I_3	I_2	I_1	I_0

FIGURE 1.7.6-1 The Data Shifter

outputs.

Since this device has tri-state outputs to enable OR-tying of the outputs of several shifters, an input is needed so that the outputs of one device may be turned off while the outputs of the other shifter(s) are enabled to be the valid ones. Otherwise, the outputs of the devices would be trying to pull each other. So, an output enable is used for this function. When this input is at the LO logic level, the device outputs are valid. But when $\overline{OE}$ rises HI, the outputs are turned off.

1.7.7 **The Decoder.** The only digital integrated circuit not yet discussed is the 7442 one-of-ten decoder found at the very right edge of the game PCB schematic. This type of decoder simply converts a BCD (Binary-Coded-Decimal) number to a purely decimal equivalent. By binary-coded decimal, we mean a binary number which cannot be greater than decimal nine or 1 0 0 1. If the binary number increments past 1 0 0 1, it becomes 0 0 0 0 again.

For example, if the BCD number at the device inputs is 0 0 0 0 (decimal 0), output 0 is selected and rises HI. When the BCD input is incremented to 0 0 0 1 (decimal 1), output 1 is selected and it then rises HI.

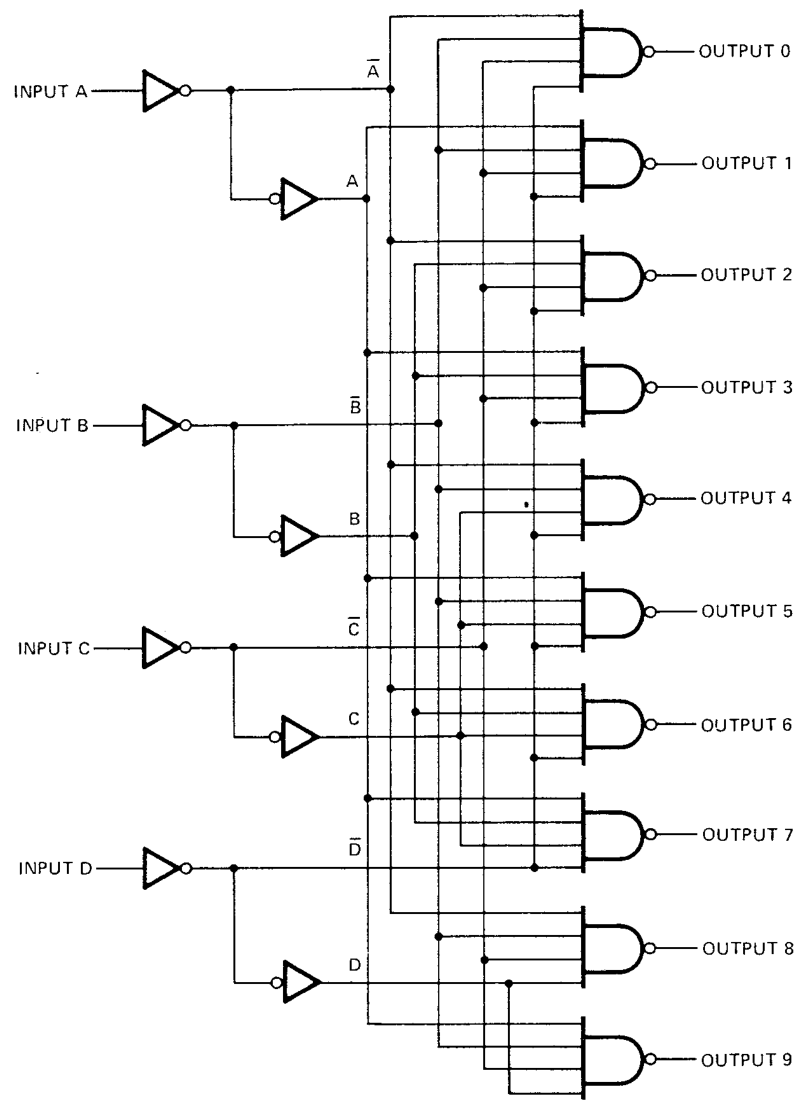


FIGURE 1.7.7-1 One-Of-Ten Decoder Logic Diagram

SECTION 2 THEORY OF OPERATION

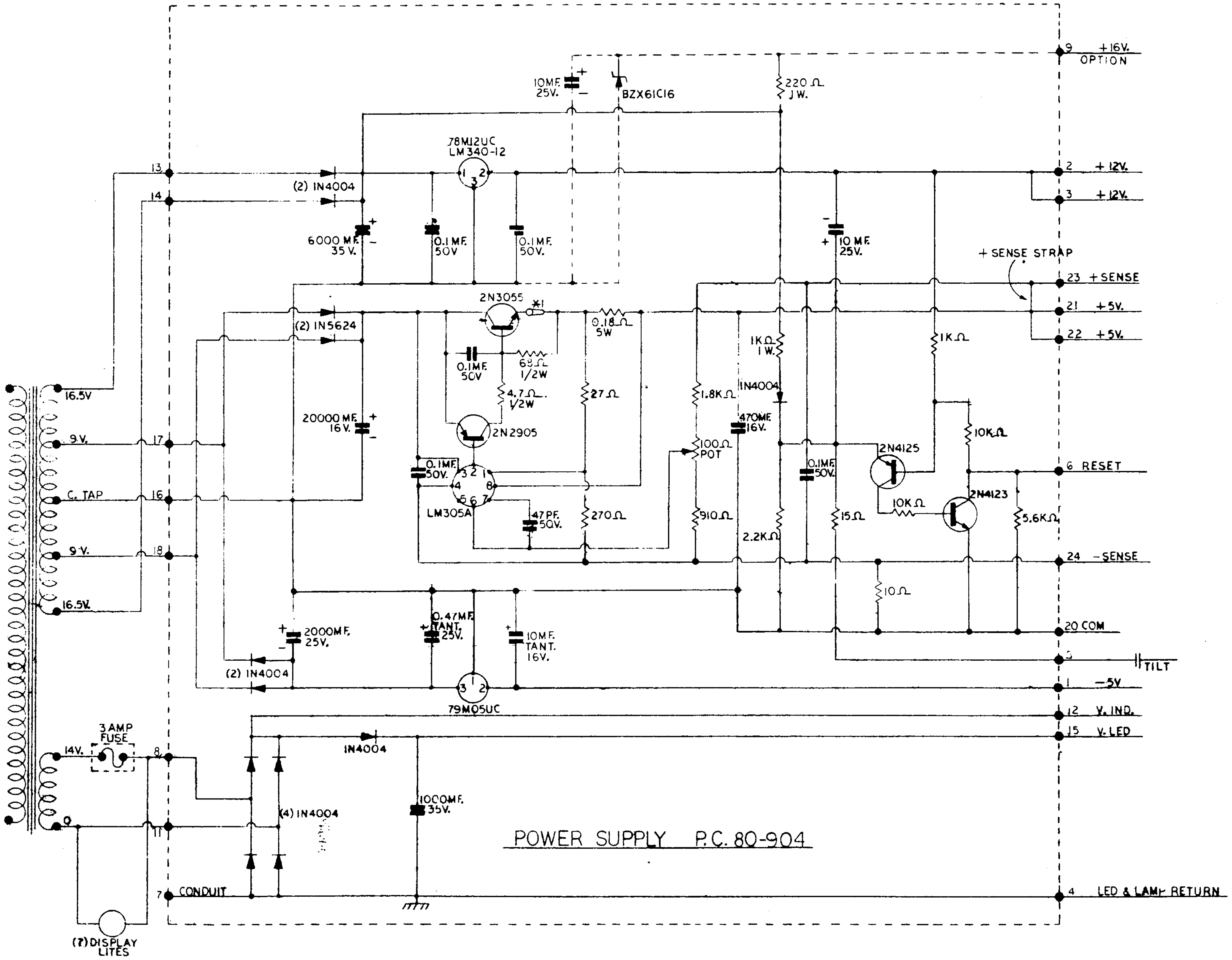
2.1 The Power Supplies

2.1.1 **Introduction.** This power supply has been designed in such a way that it can be used for all the games in the series for which the CPU mother PCB was designed. Although this game does not draw the full amount of power these supplies are capable of generating, other games in the series do so the supply must be designed to operate the biggest power hog. The power PCB generates quite a few different voltages, both regulated and unregulated, used to power a number of different types of componentry. The +15V, +12V and -5V supplies utilize integrated voltage regulators in a configuration common to video games to generate closely regulated voltages used to power the integrated circuits and other components as well. The +5V supply, however, is a relatively high current source carefully regulated by a standard fold-back, current-limited circuit design which compensates both for line voltage and load fluctuations. In addition to all these regulated supplies, two unregulated supplies are also created to power incandescent lamps (not used in this game) and the LEDs of the opto-isolators.

2.1.2 **SENSE and COM Lines.** One interesting feature of the power PCB which is not found in other games is the use of the SENSE and COM lines to detect any IR drop which might occur in the ground line between the power PCB and the mother PCB. IR drops across a ground line can cause several problems, including the annoying hum bar rolling up the TV screen which is so infuriating to operators and players. Essentially, both the SENSE and COM lines are ground lines. Notice in the wiring diagram that the SENSE and COM lines are connected together just before they enter Pins 20-21 of the mother PCB. Since the SENSE line is connected to the COM line at this point, it can be used to sense any IR drop which might occur between Pin 14 of the power PCB and Pin 15 of the mother PCB. Since the SENSE line is used as the GND reference for the entire power PCB, any IR drop which occurs across the COM line simply offsets the entire power supply system by that amount and thereby eliminates any problematic conditions which might otherwise occur.

2.1.3 **The Transformer.** The transformer reduces the 117VAC line voltage down to two center tap voltages: 9V and 16.5V. The 19.5V winding is a "high tap" used only if local power is consistently too low to operate the system. In this case, the wires from the 16.5V winding are unsoldered and placed on the 19.5V winding. The transformer also has a 14V secondary winding used directly to light a number of display lamps as well as being further processed for V_{IND} and V_{LED}.

2.1.4 **The +12 Source.** The 16.5VAC waveform from the transformer secondary is full-wave rectified by the two 1N4004 diodes and filtered by the 6000μF capacitor before the waveform is placed at the input of the LM340 T-12 integrated voltage regulator. The sense line is used as the GND reference for the regulator to compensate for any IR drop in the ground line as discussed previously. Adjustment provisions have been made by incorporating a voltage divider network composed of the 2700Ω resistor and the 100Ω trimpot. By adjusting the trimpot, the regulator can be further offset from SENSE GND to compensate for any minor deviation. The resulting voltage is further filtered by the 0.1μF capacitor to eliminate fast transients. The resulting closely regulated +12V source is made available to the rest of the machine through Pins 1 and 2 of the power PCB edge connector.



NOTE *1 FERRITE BEAD FERROXCUBE K5-001-00/38

*3 POWER SUPPLY TO BE USED WITH
MOTHERBOARD P.C. 80-900.

WARNING ⏚ IS COMMON CENTER TAP AND IS
NOT CONNECTED TO CONDUIT GROUND CIRCUITS.
⊥ IS CONDUIT GROUND AND IS NOT
CONNECTED TO COMMON CIRCUITS.
⏚ & ⊥ ARE CONNECTED TOGETHER AT
X FORMER FRAME WITH SEPERATE LUGS

FIGURE 2.1.1-1 The Power PCB Schematic

2.1.5 The +5V Source. The 9V center-tap winding of the transformer is first full-wave rectified by the two 1N5624 diodes and then filtered by the 20,000μF capacitor. This waveform is placed at the collector of the 2N3055 pass transistor and more or less of this voltage is allowed through depending on load and other factors. The actual regulator in this circuit is the LM305 which operates a 2N2905 amplifier transistor. The LM305 senses the output voltage across the 270Ω resistor and uses this waveform to control the base of the amplifier transistor which is necessary in this circuit to provide sufficient current to operate the pass transistor. If the LM305 senses a drop in voltage, it turns on the amplifier transistor which in turn activates the pass transistor. When the pass transistor is activated, it allows more voltage through to the output to compensate for the drop in voltage. The circuit also senses the amount of current across the .18Ω resistor and, if the current exceeds the safe limit determined by the value of this resistor, the amplifier transistor is shut off which turns the pass transistor off, thereby limiting current to a safe level.

2.1.6 The -5V Supply. The 9V winding is again full-wave rectified, but since the cathodes of the two 1N4001 diodes are wired to the transformer secondary this waveform is negative with respect to GND. This negative waveform is filtered by the 2000μF capacitor and notice that the other side of this and all the other filter caps are tied to COM ground. This waveform is then placed at the input pin of the LM320 T-5 integrated voltage regulator which can be further offset from GND by adjusting the 100Ω trimpot to compensate for any minor deviation from the specified -5V level. The resulting fully regulated voltage is further filtered by a 10ΩF capacitor to prevent load fluctuations from disturbing the operation of the regulator.

2.1.7 The RESET Line. This part of the power PCB provides a power-on-clear signal when the game is first turned on. Since random information loaded into many parts of the computer when power is first applied, this signal is necessary to clear this meaningless data away so the computer can start operating with a "clean slate."

The operation of this part of the circuit is really quite simple. When the game is first turned on and all the supplies come up, the 10μF capacitor connected to the emitter of the 2N4125 starts charging through the 1K resistor. Until the cap is fully charged (which takes about 1ms), there is a higher voltage at the base of this transistor than at the emitter so the transistor remains shut off. Since this transistor is shut off, the 2N4123 is also turned off and the reset line is pulled up to the HI logic level which resets the 8080 and clears the latch array on the Game PCB. But, when the cap has been charged to such an extent that the emitter is more positive than the base, the transistor is turned on which activates the 2N4123 so the reset line drops LO and allows the system to begin operating.

2.1.8 The Lamp Supplies. This area of the power PCB generates V_{IND} which may be used to power certain lamps and also V_{LED} which is the voltage necessary to power the infrared LEDs in the opto-isolators found in the game PCB. The V_{IND} voltage is developed by full-wave bridge-rectification of the 14V secondary winding and this pulsating waveform is taken directly to the lamps when used. Although this source is not used in Sea Wolf , it is used in other games of this series where certain lamps need to be controlled by the logic. Since a pure AC voltage is not efficiently switched by a transistor, it must first be rectified. The V_{LED} voltage is developed by the same initial process used to create V_{IND} however it is further filtered by the 1000μF capacitor before it is sent to the opto-isolators.

2.2 Operation of Game PCB Digital Circuitry

2.2.1 Introduction. In this section, the operation of the digital circuitry found on the game PCB is discussed. The sound system is a completely separate system and although the inputs to the sound system are discussed since these originate in the game PCB digital circuitry, the actual operation of the sound circuit is analyzed in its own section (Section 2.3). The primary functions of this section are to describe the control signals which affect data flow in the game PCB, the operation of the shifter circuit and the multiplexing of the input data from the player's controls.

2.2.2 The Game PCB As An I/O Port. The explosion lights, periscope lights, sound generators and coin counter may be thought of as separate from the main processing system. These areas are considered peripheral devices and circuits. Therefore, the section of circuitry illustrated in the Figure 2.2.1-1 is said to contain three I/O ports which output data words to the peripherals at times selected by the game program. The data word output dertermines which light or sound is activated and for how long it is to remain active. The control signals for this section of circuitry insure that these data words are gated off the data bus and out of the proper port at the appropriate time. These control signals are derived primarily from three of the processor address bits and are manipulated by the programmer to turn the various lights on and off. The data words are constructed by the processor itself in such a way as to select only the desired light or sound from several possible ones at each port.

2.2.2A Control Signals. Latching signals for all the data latches on the game PCB are derived at decoder F4. This device is a one-of-ten decoder and it operates in the following manner. The four input lines (pins 12 through 15) form a 4-bit binary number ranging from 0 to 15 which addresses, at most, one of the five connected output lines. For example, an input address of four selects the fourth output and, since all the outputs are internally inverted, this line drops LO while all the others remain HI. Invalid input addresses (10 through 15) will not select any output and all output pins will remain HI. The five connected output lines are used to clock a bank of latches (E1, F1, F3 and F5) and the selection of one of these latches in an important preparatory step in capturing data off the data bus. It is considered a preparatory step because the data latches are clocked and will accept data only during a transistion from a LO to a HI level at the clock input, an event that only after one of the decoder output pins has been selected to drop LO and then allowed to return HI again. During the time that one of the decoder outputs has been selected to drop LO, data intended for one of the game PCB latches is supplied by the processor via the data bus to the latch input. At the conclusion of the decoder clock select (after the selected output returns HI), the data word on the bus will be the one intended for the enabled set of latches.

A^8	A^9	A^{10}	SAMPLE	LO OUTPUT
0	0	0	1	NONE
1	0	0	1	LATCH EXPLOSION LIGHT
0	1	0	1	LATCH PERISCOPE LIGHTS
1	1	0	1	LATCH VIDEO DATA
0	0	1	1	LATCH SHIFT SELECT
1	0	1	1	LATCH SOUND WORD
OTHERS				NONE

FIGURE 2.2.2A-1 Decoding Game PCB Latch Clocks

Selection of which latch is to be clocked (and when) is a function of processor address bits A^8, A^9 and A^{10} combined with the processor signal SAMPLE. The address lines are simply higher-order memory locations which—in this system—are program manipulated to effect data flow control as well as normal memory address. To distinguish between times when these lines are addressing memory and when they are being used as control signals, the SAMPLE signal is generated by the CPU board as the fourth decoder input line. This signal goes to a HI level under program control to indicate that the address bits are set in control states. Whenever SAMPLE is set LO, the most significant bit of the decoder input word is at a HI level and consequently no output pin less than the one associated with a decimal 8 may be selected. Since the only decoder outputs used in clocking the latches are ones corresponding to decimal 1 through decimal 5, only unused decoder pins can be selected as long as SAMPLE is LO and no latching of data can occur. It is only when this signal is active that the three processor address bits are capable of selecting one of the latch clocks and latching the data word off the bus. Figure 2.2.2A-1 shows which latch in the bank is clocked.

2.2.2B Data Flow And Interface. The output data bus is the only means the computer has for moving data from one place to another. Since a large number of data word transfers are required of the processor during even the simplest of game functions, the data bus cannot be tied up with one function any longer than necessary. It would be impractical to place a word on the bus indicating a DIVE sound for example and leave it there until the DIVE sound was no longer needed. To tie up the data bus for a period of time with a single function—even if that time is in the range of a few milliseconds—would cause gross malfunctions in the processing flow. The idea is to put data words—output data words in this case—on the data bus and get them off again as quickly as possible so the bus may be used for the next manipulation in the processing flow (quick

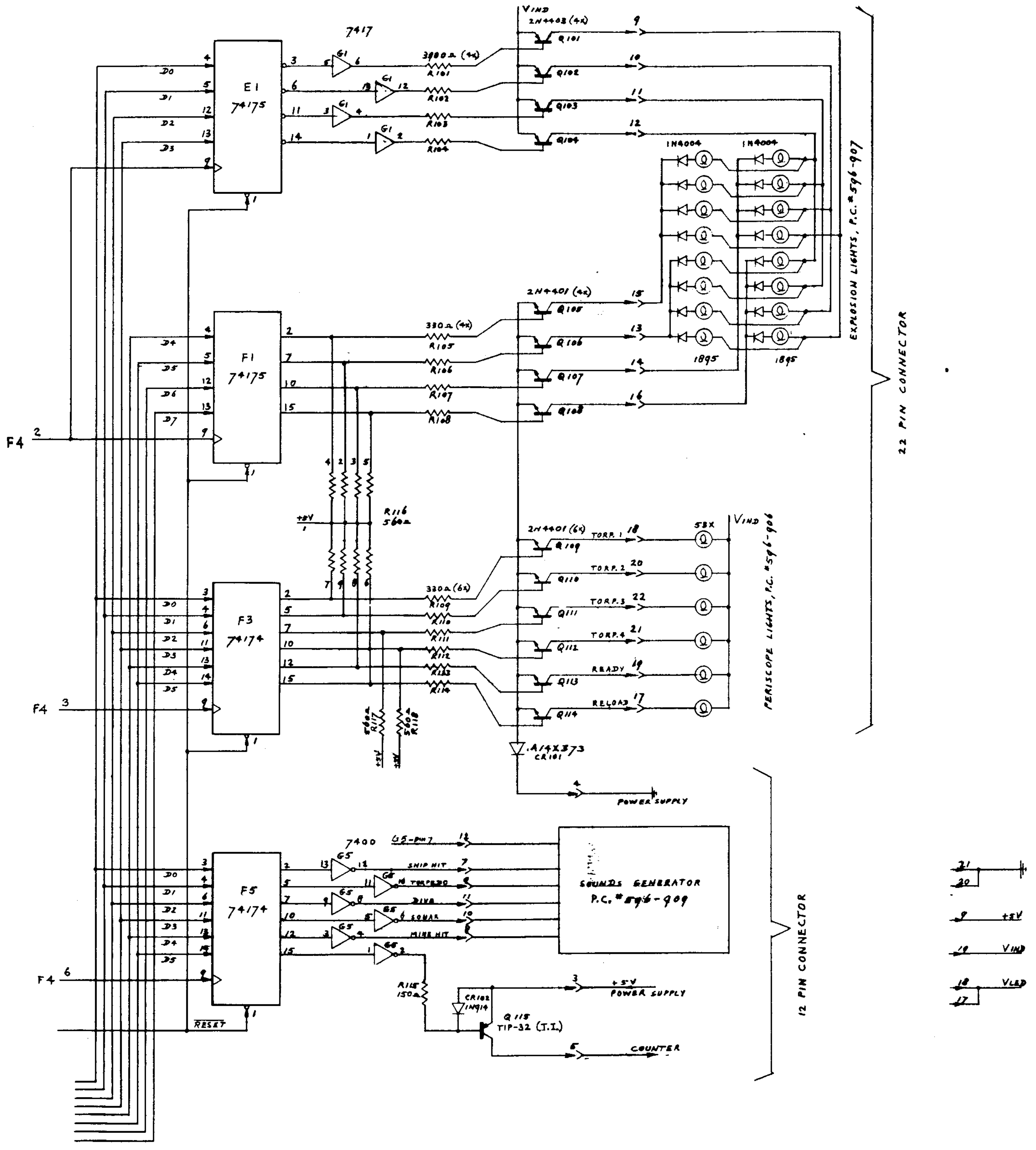


FIGURE 2.2.1-1 Game PCB Output Circuitry

in this case is about half a microsecond). In order to accomplish this, (and still output a word to the sound generator or light back for the time desired), data words are captured off the bus during selected latching times and temporarily stored in the output latches E1, F1, F3 and F5. In this way the same word may be held in the latches and presented to the peripheral devices for as long a period as desired while only tying up the bus for a brief period.

For example, by properly programming the three address bits relevant to the game PCB, F4-6 may be selected to go to a LO level, indicating that data on the bus will be clocked into latch F5 for presentation to the sound board. During the time that the clocking signal is selected and LO, the processor will retrieve an 8-bit word from memory. We may, for the purpose of illustration, assume that all the bits of this word are zeroes except bit 3. When the processor continues in the program and alters the state of either one of the processor address bits or the SAMPLE line, the decoder output pin will cease being selected and will again rise to its normal HI level. While the rising transistion is occurring, the word on the data bus will be captured by latch F5. The word will remain latched in an unchanged form until the clock signal at pin 9 again rises HI. Since all the bits in the word were assumed to be zeroes except bit 3, all the latch output pins of F5 will be set to a LO level except for pin 10 which will be active. Chip G5 inverts this signal to a LO level to provide the LO required to initiate the SONAR sound. This sound will continue to be heard until such time as the processor latches a new data word into F5 with a zero in bit position 3. The processor may elect to latch the new word at any time the program specifies (even as long as several seconds after the initiating word was latched) and thus allow the sound to continue for a relatively long period of time even though the data bus will only be tied up handling output sound data for a total of about a microsecond (one latch sequence lasts about 500ns).

The sound generator board may be considered as a separate modular device with active-LO TTL compatible inputs, a configuration which requires no level changing or other interfacing. However, this is not the case with the other output ports. The light bulbs and the coin counter coil require higher current levels than that standard TTL load and are therefore non-compatible with the logic devices of the game PCB. This necessitates an interface to provide current amplification from the standard TTL levels to the much higher current requirements of the indicators and counter coil. This conversion is provided by driving transistors Q101 through Q115, however we will discuss only one of these current boosters since all of them operate in exactly the same manner.

Chip F3 is a latch where words intended to turn on the periscope lamps are held when these lights are to be illuminated. The chip itself is the open collector type, meaning that while in the LO state the output pin is essentially shorted to ground and while in the HI state the output pin is an open or "floating" level. If the processor were to initiate a latching sequence which left the latch output F3-15 at a LO level (bit 5 on the data bus LO when clocked), the apparent short to ground at the latch output would be passed through R114 to the base of Q114, where the near ground base voltage level would prevent an emitter-base drop and hold the transistor in its OFF state. This in turn would prevent the RELOAD lamp from illuminating since there would be no ground path for current flowing through the lamp. If, however, a HI were to be latched the apparent "open" at the latch output would cause the base of Q114 to see no voltage level except that positve level provided by resistor R116. The provision of positive voltage would create an emitter-base drop sufficient to turn the transistor to the ON state and provide a current path to ground through the RELOAD lamp. The lamp would illuminate and remain illuminated until the processor conducted another output latching sequence and latched a LO into F3-15.

The same sequence of events also enables and disables the explosion lights but here the presence of sixteen possible lights requires the use of a decoding matrix to select which of the sixteen lights will illuminate when only eight data bits are available to activate them. The latch outputs at E1 apply positive current to the lamps through driver transistors Q101 through Q104. For example, a HI latched into E1-13 will appear at output pin 14 as a LO (E1 employs inverting outputs), will be buffered by the open collector G1 and will provide a LO level voltage to the base of Q104. This will turn the transistor on and supply a current source for four of the sixteen explosions lights. Inversely, if a LO level were latched into the same latch bit, the open collector output would be forced into an apparent open condition. With no voltage applied to the base of the transistor, it assumes the OFF state since there is no emitter-base voltage drop (a dribble of current from the emitter to the base serves to hold the base at the same voltage level as the emitter).

With four of the sixteen possible lights provided with a current source, latch F1 and its associated driving transistors and buffers similarly supply a ground return to four different lamps. Only one of the set of four provided with a ground path will also be in a set provided with a source of current, hence the latching of a single word at E1 and F1 will illuminate one explosion lamp if one bit is held in each latch (note that F1 outputs are non-inverting). More than one light may be illuminated at once by latching more than one HI bit into F1 and more than one LO into E1.

2.2.3 Video Data and CRT Motion. Before delving into a detailed discussion of the shifter circuit, it would be well worth the time to briefly review some of the fundamentals of the processor's handling of video data and, specifically, its ability to simulate motion on the TV screen.

Video patterns, such as the ships, mines and words (i.e. "ZAP") are permanently stored in the ROMs in the form of 8-bit words. A block of 8-bit words, with the proper bits programmed HI or LO, may be made to generate a pattern resembling a real-world object when transferred to the monitor CRT in an orderly, sequential way and Figure 2.2.3-1 illustrates how this is done.

At the initialization of the program, these blocks of data words are transferred from the ROM memory and loaded into the RAM memory where most are stored throughout the game. The initial address in RAM, hence the initial position in which the image will appear, is fixed by the program. Consequently, the images will always appear in the same locations each time the game is started or powered up. Some images (i.e. the words "Time" and "Score") retain their initial address location in RAM for as long as they appear on the screen and are simply read out each frame at the same location. Others (i.e. the ships and mines) are movable by changing the location of the video patterns in RAM so their positions on the CRT changes with successive frames.

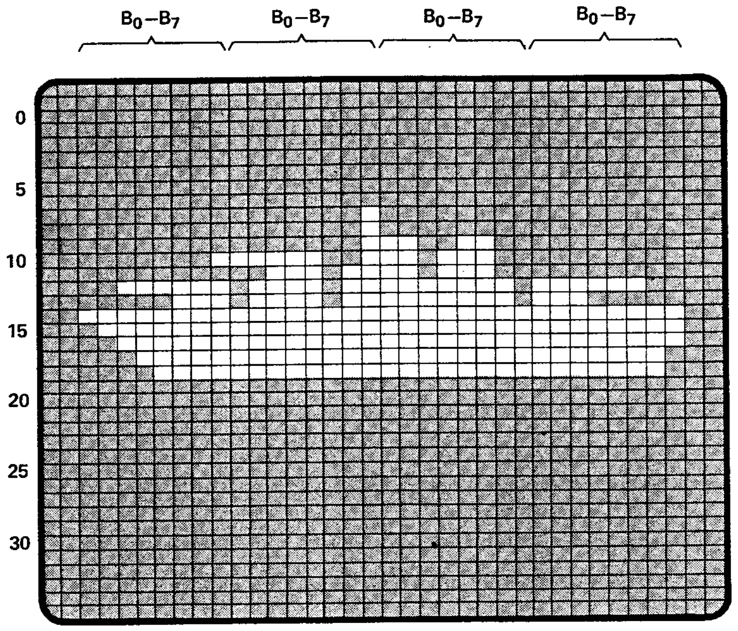


FIGURE 2.2.3-1 Video Image Generation

Vertical motion is a relatively simple process, at least when compared to horizontal movement of the images. When reading from or writing to RAM, processor addresses A^5 through A^{12} may be thought of as a vertical address for lines on the TV screen, where address 00000000 is the extreme top edge of the CRT. If the processor takes a word from a given location in RAM and rewrites it back into RAM at a different address location, the video pattern corresponding to that word will appear to have shifted up or down on the CRT when read out by the electron beam. The direction of the shift is a function of whether the new address location in memory is greater or smaller than the old one. Since the line numbers of the raster scan increase toward the bottom of the screen, greater memory addresses produce a downward shift. The amount of position shift is one raster line for each unit change in the vertical address.

While the actual mechanics of moving a complete video image up or down the CRT in a continuous and orderly way is a bit more complex than the simplified description above and requires a great deal of processor attention as well as program instructions, the principle involved should be fairly clear. A video image is nothing more than a coherent block of words. In order to move the image as a block, each individual word must be read out of RAM and written back into RAM in a new location. As long as each word in the block is moved the same number of address units, the image will remain coherent and, as long as the amount each image is moved is not excessive or unsteady, the image will appear to move in a smooth manner.

Vertical motion is conveniently produced by the CPU itself and requires no other external circuitry. Horizontal motion, on the other hand, is a more complex process party because additional circuitry is required to effect smooth, realistic motion. However, the basic principle of horizontal motion remains the same. Processor address bits A^0 through A^4 may be thought of as horizontal screen addresses where each new address number represents one of sixteen word locations across the scan line. However, each word location identifies eight bits of data in RAM and hence eight "spots" on the screen stretching across the scan line for about ½ inch. If the processor were to take a word out of RAM and rewrite it back in at a new address, the word will not appear to have moved one bit location across the screen but rather eight bit locations or almost ½ inch. Again, the direction of the shift is dependent on whether the location of the new address in memory is greater or less than the old address location. In this case, greater address locations cause shifting to the right because the electron beam scans the raster starting from the left and proceeding to the right.

In principle, there is nothing wrong with this method of data transfer and in fact this is what the processor does do to move the image. However, if the image actually moved in eight-position jumps, the effect would be jerky, uneven motion. So, a block of circuitry has been designed to take the processor's eight-position location jump and generate eight in-between shifts of one position each and thereby smooth out the jerky, discontinuous motion produced by the processor.

2.2.4 Shifter Circuit Logic. The shifter circuit is one of the most interesting areas of the Sea Wolf computer. It consists of three subsections which are: (1) a 15-bit latch composed of A6, B6, C6 and D6, (2) an addressable shifter at A3, B3, A5 and B5 and (3) an address latch at E6.

The 15-bit latch operates in very much the same way as the output data latch described in Section 2.2.2. The control signal is generated at H5-11 by the NANDing processor address bit A^{12} and the SAMPLE line. When both these signals are HI, data destined for the first eight bits of the 15-bit latch are placed on the data bus lines. The subsequent fall of SAMPLE drives the NAND output HI and the next rising edge admits the 8-bit word on the bus into the first eight locations of the latch. The second set of latches consist of seven bits and do not connect to the data bus. Rather, the data to be latched into these positions is the word written into the first eight bits of the latch on the previous latching cycle. During the first latching cycle, an 8-bit word – call it "word 0" – is latched into the first eight bits of the latch. During the next latching cycle, a new word – call it "word 1" – will be admitted into the first eight positions of the latch while seven bits of word 0 will be transferred to the other seven locations in the latch. Note that the least significant bit – bit 0 – of word 0 is not transferred to the second part of the latch. It is simply lost. Similarly, on the next latching cycle word 2 will be latched into the first word position of the latch while word 1 is transferred into the second set of seven latches. Bit 0 of word 1 is also dropped.

The outputs of this 15-bit latch are then placed at the inputs of the

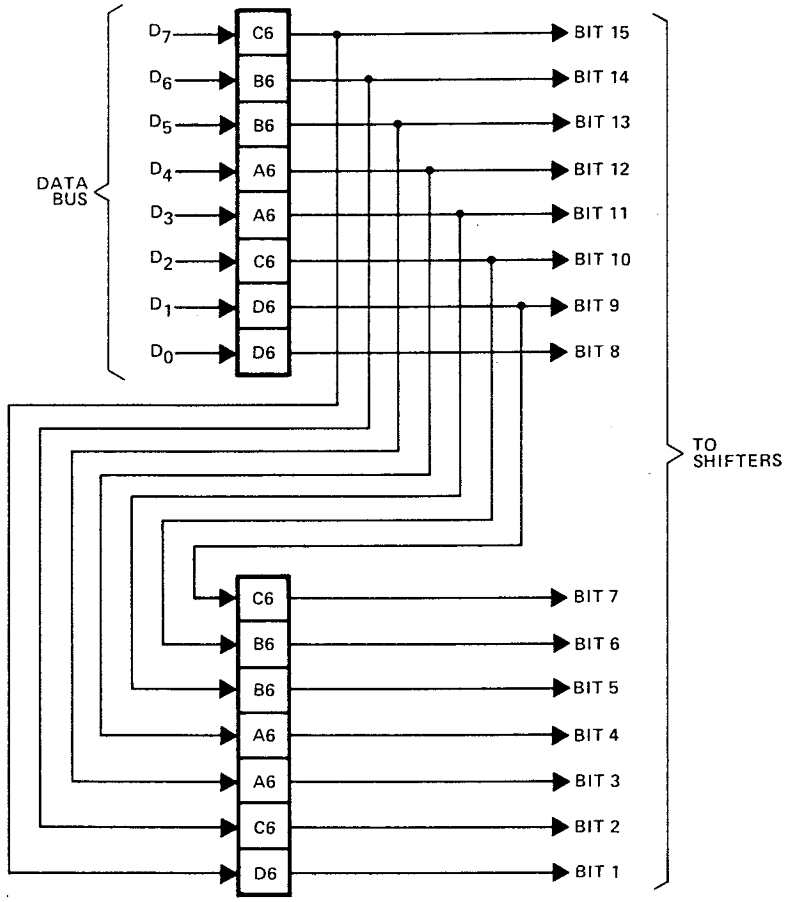


FIGURE 2.2.3B-1 Shifter Circuit Flow Diagram

SHIFTER ADDRESS			WORD OUTPUT							
S_0	S_1	S_2								
0	0	0	8	9	10	11	12	13	14	15
1	0	0	7	8	9	10	11	12	13	14
0	1	0	6	7	8	9	10	11	12	13
1	1	0	5	6	7	8	9	10	11	12
0	0	1	4	5	6	7	8	9	10	11
1	0	1	3	4	5	6	7	8	9	10
0	1	1	2	3	4	5	6	7	8	9
1	1	1	1	2	3	4	5	6	7	8

FIGURE 2.2.3B-2 Shifter Truth Table

25S10 shifter chips (see Section 1.7.6). These devices function in much the same way as a multiplexer with one significant difference. Instead of the address and enable lines determining which input is reflected by the output, the address lines and enable of the shifter determine which four of the seven inputs are output from the four shifter output lines. Any combination of four adjacent inputs may be selected by one of the four possible states of S_0 and S_1. Since two of the shifters are enabled by S2 and the other two shifters by S2, two of the four shifters will always be enabled at any given time so that an 8-bit word will be generated from eight of the fifteen bits held in the latch. The latch outputs are wired to the shifters in such a way as to allow any eight adjacent bits to be output as the shifted word, depending on the particular address at S_0 and S_1. Figure 2.2.3B-2 shows which eight bits of the latch are output for any particular address and enable combination.

The address latch E6 captures three bits off the data bus (D_0, D_1 and D_2) when its clock input (pin 9) rises HI. These three bits are under program control and determine the address code to be presented to the shifters. The clock for the device is output from H5-8 as the NANDed signal of processor address bit A^9 and SAMPLE. As before, data intended for use as shifter address code will be placed on the first three bits of the data bus during the time A^9 and SAMPLE are HI (H5-8 is LO). When either of these two signals returns LO, the signal at pin 9 of the latch will rise HI and the data will be latched into E6. The data thus latched will select a new address code for the shifters.

2.2.5 A Typical Shifter Sequence. The foregoing describes the operation of the individual shifter circuit sections but it does not relate the shifting process to the CPU or the other processor circuitry. In order to tie these areas together, we will analyze the following hypothetical example and take a step by step walk through a typical shifting process.

Suppose we have an 8-bit word stored in a certain RAM location, say horizontal address 3. This 8-bit word is displayed on the CRT by being read out of RAM once each frame. Since it is in RAM location 3 it will appear at the far left of the screen where bit 0 is the LSB and occupies the leftmost position in the word. We will also assume that we need to transfer this data word one position to the right, or to horizontal address 4. We could do this simply by reading the word out of RAM location 3 and writing it back in at location 4, but this would appear as a discontinuous eight bit jump when viewed on the screen. Instead, we will use the shifter circuit to move the data word a bit at a time to the right during each frame and accomplish the entire transfer in eight frames of a single position shift each.

STEP 1. Read word 4, the location we wish to transfer into, out of RAM and into a temporary holding register. This register may be internal to the CPU or some unused location in RAM.

At this point we have the two words involved in the transfer operation being held where they may be read at any time but where they will not be altered by subsequent operations. Word 3, the word we wish to transfer, is held in ROM while word 4, the word occupying the place word 3 is expected to move into, is safely held in the temporary register. This is absolutely necessary since the CPU must keep track of where in the RAM memory the video image is held. When the processor attempts to move that image, it must first be able to locate its image data in RAM and thence onto the screen. However, the processor is capable only of locating data in 8-bit words, so video image data must always be transferred from one whole word location to another. Partial word transfers, such as three bits to the right, would place the image data in a location overlapping two words and the processor would not be able to address the beginning or end of the image. So what must be done is to hold the image data and word 4 in an unchanged state until we have shifted the image through the eight intermediate single position shifts. Once this has been done, the word will be written into the new location and the processor informed that the data pattern can now be found at the new location.

STEP 2. Read word 2 (the word immediately to the left of the word to be transferred) out of RAM and latch it into the first eight bits of the 15-bit latch.

STEP 3. Read the data word out of ROM and latch it into the first eight bits of the 15-bit latch. Note that at the same time this occurs the data already in the first eight bits of the latch (word 2) will be transferred back into the second seven bits of the latch.

STEP 4. Set the data lines to the following values:

$$\boxed{\begin{array}{l} D_0 = 1 \\ D_1 = 0 \\ D_2 = 0 \end{array}}$$

and latch this 3-bit word into the shifter selector latch E6. This will set the shifters to output a new word where the LSB is bit 7 of the 15-bit latch and the MSB is bit 14 of the same latch. This new word will essentially be the data word shifted to the right one bit, its MSB being shifted out of the word and its LSB being moved up from word 2.

STEP 5. Take the output word from the shifter and place it in RAM for TV display at location 3.

STEP 6. Read word 4 out of its temporary register and latch it into the first eight bits of the 15-bit latch. Note that this will also take the data already in the first eight bits (the data word) and transfer them to the second set of seven bits in the latch.

STEP 7. Without changing the shifter select controls, again read the output word from the shifter and place it in RAM location 4 for TV display. The new word thus generated will also be built in such a way as to place the MSB in the least significant position of word 4. In effect we have moved word 4 one bit to the right, losing the MSB and taking one bit from the data word as its LSB.

This completes one pass in the transfer operation. At this point, we have the original data word still held in ROM, the original word 4 in a temporary register while words 3 and 4 in RAM are the data words shifted one bit to the right. We may keep this arrangement for as long as we wish, depending on how we want the image to move. If we desire slow motion, we may keep everything the same for 8, 16 or any other number of TV frames. On the other hand, we can just as easily produce rapidly-appearing motion merely by shortening the period of time the data is kept unchanged. However, at some time we will have to continue the transfer operation by making another pass with the shifter (8 to 18).

STEP 8. Read word 2 (the word immediately to the left of the original data word location) out of RAM and latch it into the first eight bits of the 15-bit latch.

STEP 9. Read the data word to be moved out of ROM and latch it into the first eight bits of the 15-bit latch. Note that this simultaneously transfers word 2 into the second seven bits of the latch.

STEP 10. Set the shifter address select lines to the following values by latching the appropriate 3-bit word into E6:

$$\boxed{\begin{array}{l} S_0 = 0 \\ S_1 = 1 \\ S_2 = 0 \end{array}}$$

STEP 11. Write the word output from the shifters into RAM location 3 for video display. This new word must be constructed in the following manner. The LSB of the word will be the second MSB of word 2, while the MSB of the new word will be the second most significant bit of the data word. In effect the new word is the data word shifted two positions to the right, the two most significant bits being shifted out of the word altogether while the two least significant bits are moved up from word 2.

STEP 12. Read word 4, the new location for the shifted data, out of its temporary holding register and write it into the first eight bits of the 15-bit latch. This will transfer the data word to the second set of latch bits.

STEP 13. Without changing the shifter select control lines, read the output word from the shifters into RAM location 4. Again, the new word will be the original word 4 shifted two bits to the right, the rightmost two bits being shifted out of the word and the leftmost two bits being moved up from the data word.

At this point, we still have the original data word stored in ROM while the original word 4 is similarly still being held in the temporary register. The data in RAM that is being displayed on the screen at word locations 3 and 4 is the original data word shifted two bits into word 4. Again, we may keep these RAM words unchanged for as long as desired, but in practice they are held in the 2-bit shift configuration for as long as they were held in the 1-bit shift positions. As long as data spends equal amounts of time in each of the eight shift configurations, the motion across the CRT will be smooth and continuous.

It is not necessary to repeat the entire latching and shifting procedure again to explain how the rest of the transfer is accomplished since each pass of the shifter will be the same as those described previously except that a new shifter address code will be latched to select right shifts of 3, 4, 5, 6 or 7 bits. After the 7-bit shift has been selected by an address code of 1 1 1, the program changes to the following operation.

STEP 14. Read word 2 out of RAM and latch it into the first eight bits of the 15-bit latch.

STEP 15. Set the latch E6 to select a shifter address of 0 0 0.

STEP 16. Read the word output from the shifters (the unaltered word 2) and write in into RAM location 3.

STEP 17. Read the data word we are moving out of RAM and latch it into the first eight bits of the 15-bit latch.

STEP 18. Word 4 in its original form may now be cleared out of its temporary register. We must now increment the horizontal accounting scheme in the CPU to indicate that the data pattern now resides in location 4 in RAM. The processor must log all such transfers so that it can find the data in RAM should it become desirable to effect another transfer.

The above is a simplified description of what the shifter and processor must do to move a single data word one location to the right. By expanding on the principles outlined above, more complex motion problems may be solved. Consider, for example, the case where the video image we wish to move one word location to the right is not a single data word, but a block of data words four words wide by sixteen lines tall. To move such a block merely requires repetition of the above steps for the required number of word locations and the required number of lines. In the example above, only three words were actually sent to the shifters to be involved in the transfer: (1) the word immediately to the left of the data word, (2) the data word itself and (3) the new location word. To move a longer block of words, on each pass we would send to the shifter the word immediately to the left of the first data word, the first data word, the second, third, fourth data words and finally, the new location of the rightmost data word. For each pass we need only set the shifter addresses once for a one, two, etc. bit shift to the right and we need store only the last word of the string in the holding register. If we performed such a pass across five or six word locations on one raster line, the other lines of the image could similarly be shifted by simply conducting an identical shifter pass on each line in the image. In this manner a block of video data of any size can be moved one (or more) word locations to the right in a series of eight passes over the block.

Leftward motion is accomplished similarly by eight consecutive passes over the data words. Suppose that our example above had been to move the single data word one location to the right or into word 2. On our first pass, we need to transfer the Least Significant Bit of the data word into the most significant position of word 2. This condition is easily satisfied by selecting a shifter address code of 1 1 1 for the first pass and essentially generating all the shift codes in the opposite sequence as in the previous example. By running through the shifter codes in reverse, we can generate left motion in an almost identical manner as rightward motion.

2.2.6 Game Data Multiplexing. The circuitry responsible for the acquisition and multiplexing of game data to the CPU for processing is outlined in Figure 2.2.6-1 and this circuitry may be conveniently divided into two sections. The first is the opto-isolation of the player's control and coin data. Since all the opto-isolators operate similarly, only one will be covered in the following discussion. The second section contains the game data multiplexers which multiplex the input game data back to the CPU for processing.

2.2.6A Isolation of Player Controls. We will discuss the operation of the trigger switch circuit (opto-isolator A1) as an example for the operation of the entire group of opto-isolators. When the trigger switch is open (i.e. any time a torpedo is not being fired by the player), A1-3 is an open circuit. Internally, an infrared light-emitting-diode is placed across pins 3 and 4 so that an open circuit at pin 3 will result in no current flow through this diode. The output pin of the chip (pin 6) is the collector of a photo-transistor whose emitter is available at pin 5. With no current through the LED, the transistor will be held in its OFF state by the optical detector in its base circuit. As long as the transistor is turned off, the only voltage source for output pin 6 is the +5V supplied by the resistor in the resistor pack at game PCB location C-D-1.

When the player wishes to fire a torpedo and closes the trigger switch, A1-3 is shorted to ground. With a ground path supplied at pin 3, the LED will conduct and emit infrared light inside the chip. This is detected at the base of the output transistor and turns on the transistor which, for all practical purposes, shorts the collector output (pin 6) to the emitter ground. In this way, the opto-isolator acts as a non-inverting buffer which completely isolates the control panel from the delicate logic.

The opto-isolators themselves are linear devices, meaning that the voltage appearing at the output pin varies with the voltage at the input in a linear fashion. Consequently, the output signal may rise or fall at a relatively slow rate or may even waver in its transistion. While this presents no real problem in slow speed circuitry, it can cause serious problems in a microprocessing system which depends on its signal lines to be in clearly defined HI or LO states at the instant those lines are sampled. Slow transitions or unsteady logic states may cause the data represented by the signal to be misinterpreted or misread. For this reason, Schmitt-triggered inverting buffers are used after the outputs of each opto-isolator. A Schmitt device is characterized by a very narrow range of input threshold voltages for switching the output between states. This has the effect of creating output pulses with very sharp rising and falling edges regardless of the speed of the input transistion. The resulting squared-up output is presented to the multiplexers as a steady logic level with sharp rise and fall characteristics.

2.2.6B Game Data Multiplexing. Although the processing activity of the game is essentially set by the program and generally progresses in an inflexible manner, there are certain areas where the programmer may elect

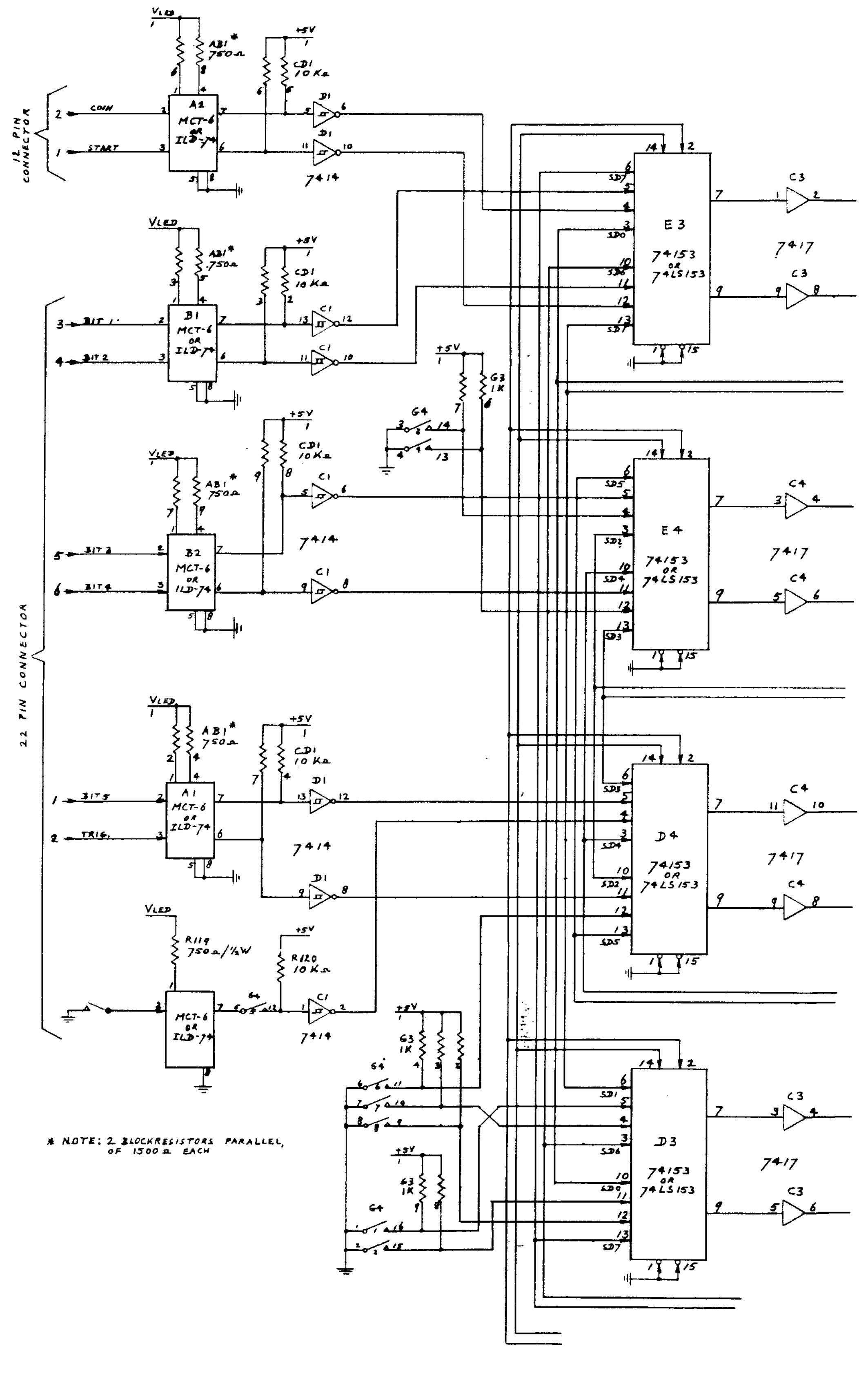


FIGURE 2.2.6-1 Game PCB Input Circuitry

to leave a blank which will later be filled in by the processor itself at the actual time of play. For example, the CPU must constantly scan the coin switch to detect when a coin is deposited so that it may jump out of its attract mode operation and begin processing that part of the program related to the playing of the game. It must also scan a number of other similar variables in the program in order to allow the game to proceed in the proper manner. The processor, however, can only receive such information in the form of an 8-bit word placed on the data bus during a time in the program when player control information in expected to be present on those lines. For this reason, two functions are required to insure that the program parameters are sampled properly: (1) a control function and (2) a data multiplexing function.

The control function is handled by two of the processor address bits which come to the game board and these are A^8 and A^9. By means of program manipulation of these address lines, one of four possible select codes may be presented to select pins 2 and 14 of multiplexers E3, E4, D3 and D4. In this manner, the time at which each one of the four possible 8-bit words are selected to be output from the multiplexers becomes a programming function dependent on processing activity in areas of the program characterized by the four combinatorial states of address bits A^8 and A^9.

The actual composition of the four words selected is shown in Figure 2.2.6B-1 and each of the multiplexers operate in the following manner to construct these words. For a select code of 00 at multiplexer pins 2 and 14, pins 6 and 10 of the two input sets will be output at pins 7 and 9 respectively. If a different address select code were presented by placing a HI at pin 14, the logic states of the signals at input pins 5 and 11 would be reflected the the respective output pins. In this way, any four of the inputs in each set may be selected to be output at the output pin of that particular set.

A^8	A^9	MX_0	MX_1	MX_2	MX_3	MX_4	MX_5	MX_6	MX_7
0	0	SD_7	SD_6	SD_5	SD_4	SD_3	SD_2	SD_1	SD_0
0	1	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	TRIG	SW 1	SW 2
1	0	COIN	START	SW 3	SW 4	SW 5	SW 6	SW 7	SW 8
1	1	SD_0	SD_1	SD_2	SD_3	SD_4	SD_5	SD_6	SD_7

FIGURE 2.2.6B-1 Composition of Game Data Words

The words selected by address codes 01 and 10 require no special attention since the data word chosen is constructed on a bit-by-bit basis from the player's controls. However, codes 00 and 11 do need slight explanation. These two codes select the output word from the shifter circuit for placement on the game data bus. This word returns to the processor via the multiplex data bus for restorage in RAM and eventual screen display after having been shifted left or right by the shifter circuit on the game board. Note that a single game data bit, MX_0 for example, may serve to return either of two shifted data bits: SD_0 or SD_7. In other words, by processor control of the two address lines, data sent to the shifter may be returned to the processor either in the normal configuration or as a mirror image of the original word. By using this capability, only one image orientation need be stored in ROM to provide dual orientation on the screen. An aircraft carrier, for instance, may be stored in ROM as the permanent copy of the image in a left-facing orientation and may, by beihg run through the shifter and game data multiplexer, be stored in RAM for screen display in either a left-facing or right-facing orientation. Therefore, a image may be placed on the screen as a left or right moving object while only taking up a single-orientation block of ROM for permanent image storage.

2.3 The Sound Circuitry

2.3 Sound Generation. Five separate audio signals are generated on the sound board and these are the two explosion sounds, a torpedo "hiss", the sonar and the dive sound. Each sound is triggered by a separate TTL compatible input which is timed by its own timing and switching elements. The two explosion sounds are almost identical except for component values and both will be discussed together, however references will be made to circuit dissimilarities where they exist. The remaining three sounds are very different in nature and will be discussed separately.

In addition to the five sound generation circuits, the sound board also contains a separate power regulation section and a "raw"noise generator. Since these two sections of the system are common to nearly all the sound circuits, they will also be discussed together. Figure 2.3.1-2 shows the sound board schematic with the five sound sections, the power supply and the noise generator indicated separately.

2.3.1 Power Supply And Noise Generator. The initial unregulated AC is supplied to the sound board via edge connector pins 1 and 2. This voltage is full-wave rectified by two 1N4001 diodes and filtered to some extent by the 1000μF capacitor C5. Some of the current available at the positive side of this capacitor is trickled through the 16.8V Zener diode at the base of Q28 to maintain a well regulated 16.8 volts. The .6V base-emitter drop across the transistor provides a very steady 16V at the transistor's emitter which is also the main power bus for the sound system. Current required to maintain the 16V level with changing sound system demands is supplied by the collector-emitter current of Q28. The additional filtering provided by the 470μF capacitor on the power bus helps to buffer any high frequency current requests made by the system.

Q27, along with the Zener diode on its base, acts as a noise generator for those sounds using random white noise as an audio source. Zener diodes are characteristically noisy when conducting in their "knee" region since the amount of current flowing through them is highly erratic and unstable. This instability is amplified by Q27 to provide a source of wide frequency random signals for capacitive coupling into the sound amplifiers.

2.3.2 Explosion Sounds. The two explosion sounds operate identically in general with only minor differences in component values. Consequently,

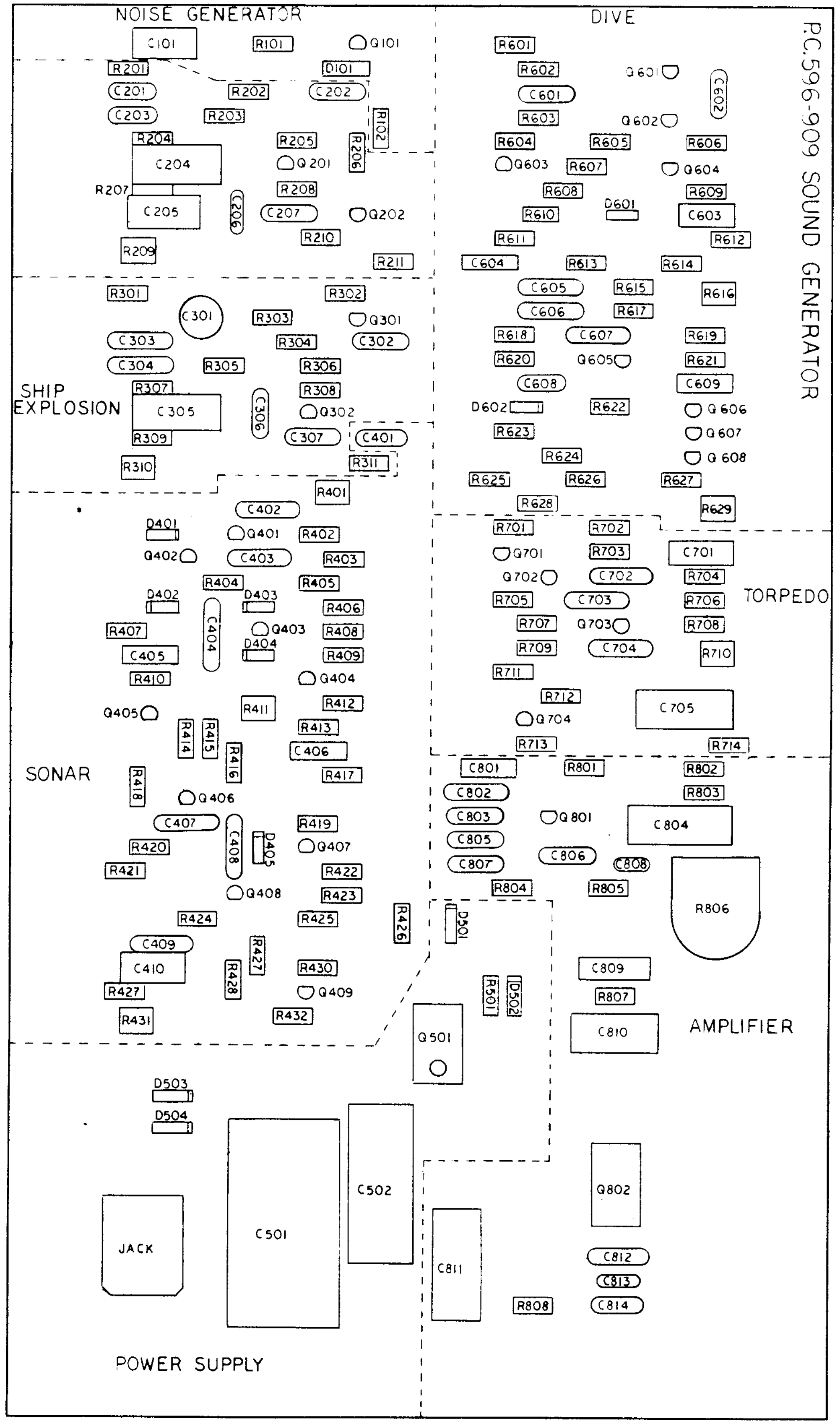
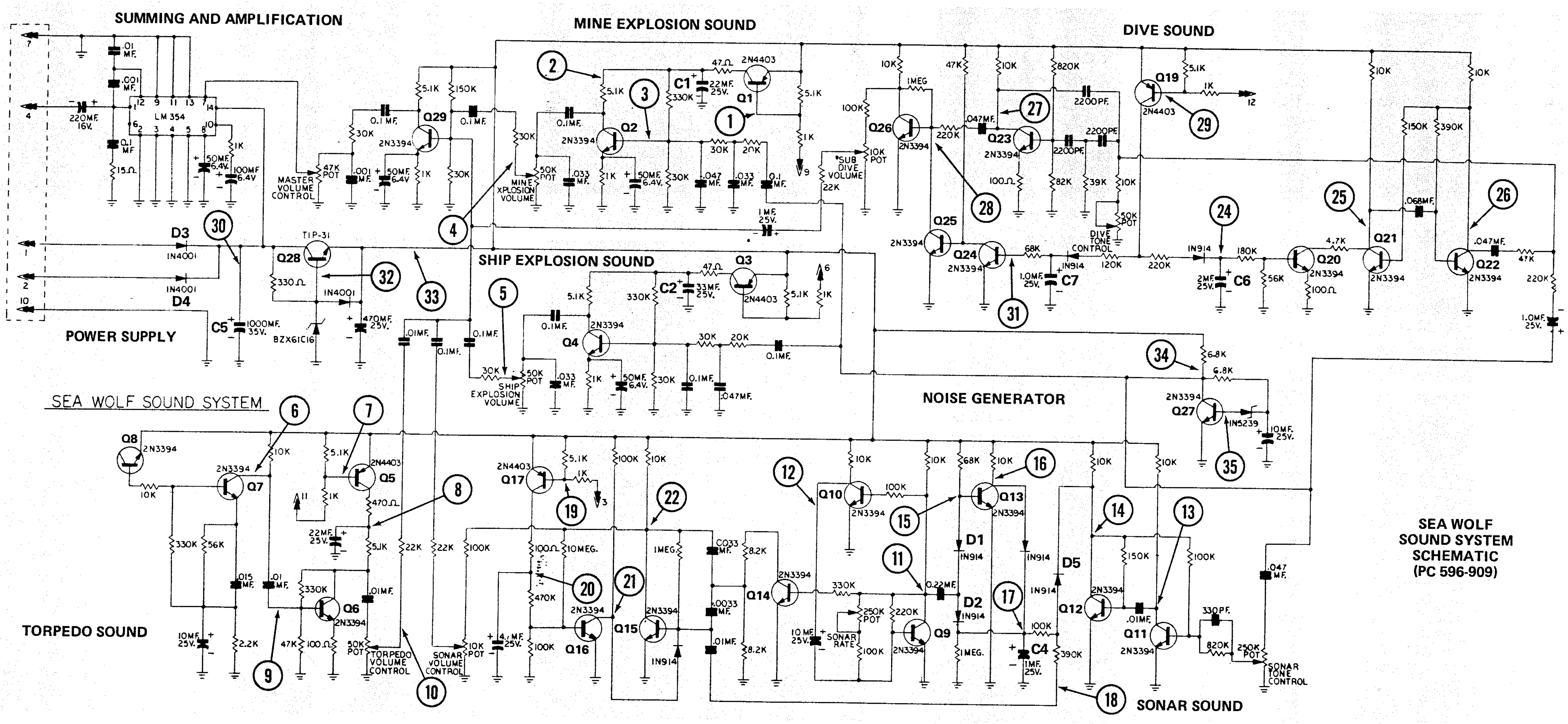


FIGURE 2.3.1-1 Sound PCB Component Diagram



SEA WOLF
SOUND SYSTEM
SCHEMATIC
(PC 596-909)

FIGURE 2.3.1-2 Sound System Schematic

we will discuss only the mine explosion sound and note any significant differences with the ship explosion sound where they exist.

Edge connector pin 9 of the sound board is the TTL trigger signal from the game board. The output device which supplies this signal is an open collector and, for our purposes, may be considered either as a ground or an open. The sound trigger input is active LO. While the TTL input is in its open state, the only voltage that will appear at the base of Q1 will be the Vss level supplied by the 5.1K resistor and a slight trickle of base-emitter current. No voltage drop will exist between the emitter and base of Q1 and the transistor will be in its off state. No current will be supplied at the collector. When the TTL input drops LO, the base of Q1 will be held at a lower voltage with respect to the emitter, the transistor will turn on and current will flow from the collector of the transistor and into capacitor C1. This capacitor is provided with a low resistance charging path and a high resistance discharging path once Q1 again turns off at the conclusion of the trigger signal. The voltage found at this capacitor will therefore be characterized by a very sharp rise, followed by a slow decay at the conclusion of the trigger signal. This characteristic provides the explosion noise with a sharp attack and slow decay, a phenomenon associated with noises such as explosions where the initial bang is followed by a period of fading rumbles and reverberations. Note the capacitor value corresponding to C1 in the ship explosion circuit (C2) is larger than C1 which causes the ship explosion sound to last a bit longer than the mine explosion.

C1 is actually the power supply for the common emitter amplifier centered at Q2. The gain of this stage is directly proportional to the voltage present at the capacitor so that the amplifying characteristic of the stage is a sharply switched high gain followed by a period of slowly falling amplification. The signal amplified is the output of the noise generator coupled into the base circuit through the 0.1μF capacitor. The two other capacitors and resistors in the base circuit for a two-stage passive low-pass filter used to dump the higher frequency noises to ground and leave only the deeper sounds inherent in an explosion. Note that the filter values for the ship explosion are different than those of the mine. The larger values of capacitance will dump more of the higher frequencies to ground, creating a purer deep tone for the ship explosion. The effect is a deeper, more realistic tone.

The filtered, amplified and pulse-shaped noise is capacitively coupled out of the sound amplifier from the collector of Q2 and sent to the volume pot where its relative amplitude may be changed.

2.3.3. Torpedo Sound. The torpedo sound circuit is a unique circuit for it contains its own noise generator (Q7). The noise source in this case is the base-emitter diode of Q8 which is filtered by the band-pass network at the collector of Q7. A separate noise generator is used since the noise needed for the torpedo sound is a constantly pitched hiss of relatively high frequency. The tone is almost musical in its consistency and does not contain as much of the wide frequency random noise found in the explosion sound waveforms.

Switching the sound on and off proceeds in essentially the same manner as described in the explosion sounds. Q6 acts as an amplifier for the high frequency hiss coupled into its base, the stage gain being dependent on the voltage found on the stage power capacitor (22μF). As before, this capacitor is charged by turning on Q5 with the application of a TTL LO trigger signal at edge connector pin 11 and is discharged at a relatively slow rate by the removal of that same input. In this case, the sharp attack and slow decay of the Q6 stage simulates the firing of the torpedo and the gradual fading of the sound with increased distance. The signal at the collector of Q6 is filtered, switched and pulse shaped and only then is it coupled into the volume adjustment pot and into the summing amp.

2.3.4 Sonar Sound. The sonar circuitry is much more complex than that of the other sounds just described because it consists of four sound components rather than two. There is a 1Hz tone signal, a timing factor which determines the frequency at which the tone will be pulsed (and the echo generated), an exponential decay factor for the fade-out of the tone and an emphasizing component which further sharpens the pulse attack.

Q11 and Q12 form the tone generator for the sonar sound. These transistors basically act as a 1Hz square wave oscillator where the frequency is determined by the bias voltage level on the base of Q11. With the sonar tone control set to ground, the oscillator output is a steady 1Hz. By rotating the tone control away from ground, the output of the noise generator is coupled to the base of Q11 and alters the frequency of the circuit in a random way. With the pot fully rotated away from ground, the oscillator is a variable frequency device operating randomly between 1KHz and about 500Hz.

Q9 and Q10 form a second oscillator and output a square wave pulse approximately once per second. The frequency of this circuit varies with the setting of the sonar rate pot. The output of this oscillator is capacitively coupled from the collector of Q9 and into the circuitry surrounding Q13, where diodes D1 and D2 act as rising and falling edge detectors. The falling edge of the pulses are detected and sent to Q13 where they are amplified into large current spikes for charging the 1μF capacitor C4. The rising edges of the pulses are detected and taken in an unamplified form to recharge C4 a short time later. Since no amplification is provided for the rising edge, the effect that they have is to recharge the capacitor to about one-third the level to which it was charged by the amplified falling edge spikes.

Diode D5 acts as a gate for the 1KHz tone signal. With C4 completely discharged, the diode will be reverse biased and the tone signal will not appear through the device. With the input of a falling edge into the Q13 circuit, C4 will quickly charge to its high voltage level and the 1KHz tone will flow through the diode. As the capacitor discharges exponentially, the amount of current available for conduction through D5 will also drop off and the peak-to-peak level of the AC signal will fade similarly. After a short period of time, a rising edge will occur at the input to the Q13 circuit and C4 will be recharged to about one-third its original value, sharply increasing the peak-to-peak AC signal momentarily before resuming its decay. Therefore, output across the 390K resistor is the switched-on tone fading for a time, then the reoccurrence of the tone at a new high level (the echo effect) before continued fading at an exponential rate.

This complex signal is then capacitively coupled into the base of Q15 where it is added to the output from Q14 and switched on and off in response to the trigger signal. Q14 acts merely to couple the rising and falling edge of the slow pulse generator into amplifier Q15 which further accentuates the sharpness of the attack by increasing the AC signal strength at the beginning of the pulse. The diode at the base of Q15 acts as a current switch to dump Vss from the 100K resistor into the base circuit in response to the state of Q16. When this transistor is in the off state, its collector has no effect on the Vss supply line from the 100K resistor and the voltage supplied to the base of Q15 is sufficient to saturate it and prevent sound amplification. Then Q16 is turned on, the Vss supply line is held to ground by the apparent collector-emitter short and the diode on the base of Q15 becomes reverse-biased. Bias voltage for Q15 is provided by the 1M resistor which places Q15 conduction in a region correct for output amplified audio.

Q16, like the other circuits, is switched on and off by the charging and discharging of C4, which is charged to its full level by turning on Q17. The discharging of C4 proceeds through the 470K and 100K resistors. Q16 will be switched on and noise will result from the time the capacitor is initially charged until the discharge voltage falls so low that the threshold of the switching transistor Q16 is reached and, at this point, this device will turn off sharply. In this manner, C4 and its two resistors provide a timer for switching on the sound, allowing it to occur for the time required to discharge C4, then turning it off suddenly. The sound itself is conducted through the volume pot and capacitively coupled into the summing amp.

2.3.5 Dive Sound. The dive sound generator is also a three component circuit employing a pulse shaper and two oscillators. The digital switch centered at Q19 operates in exactly the same way as the other sound switches and acts to charge two timing capacitors (C6 and C7) when edge connector pin 12 is pulled to ground.

Transistors Q21 and Q22 form a square wave pulse generator with a frequency normally of approximately 50Hz. The frequency of this generator closes to about 25Hz with the charging of C6. Charging requires about a second, hence the overall effect is to drop the frequency of the generator an octave, then open it back up to 50Hz in a two-second cycle.

The square pulse output is taken from the collector of Q22 by the .047μF capacitor and mixed with random noise from the noise generator. The combined signal is wired to the base circuit of Q23 where it acts to modulate a 1.8KHz sine wave generated at this transistor. The amount of modulation is variable using the dive tone control pot. The modulation effect may be seen in Photograph 3-10-4.1 on page 27.

The closing of digital switch Q19 also charges capacitor C7 which acts as the timer for the duration of the sound. The time required for charging and discharging this timer is about one second. As soon as the voltage across the capacitor rises above .6V, Q24 will turn on and ground the base of Q25. With Q25 turned off in this manner, bias voltage on the base of Q26 is provided by two biasing resistors (1M and 10K). Q26 is thus set in a conduction region proper for amplifying the dive sound. As the voltage on C7 discharges to a level near ground, Q24 suddenly turns off and allows Vss to reach the base of Q25 through the 47K resistor. Turned on in this manner, Q25 holds the base of Q26 near ground and renders it useless as an amplifier. Therefore, the sound output from the collector of Q33 is switched on and off in response to the charging and discharging of C7.

2.3.6 Summing And Power Amplifiers. In this circuit Q29 is the sound summing amp which capacitively couples the various sounds together onto a common output line to the power amp. Gain is provided at this point to boost the signal level to a value compatible with the input of the device and the single master volume control allows the game operator to select a common gain setting for all the various sounds. The additional power required to drive the speakers in the system is provided by the LM354 integrated audio amplifier. This device is powered by the unregulated DC at the collector of Q28. AC ripple is not heard since the LM354 has provisions for supply decoupling. For more information regarding the operation of this device, consult the manufacturer's data sheet.

SECTION 3 TROUBLESHOOTING PROCEDURES

3.1 Introduction

3.1 Introduction. In this section each of the circuit blocks found on the game PCB will be discussed in terms of actual signal descriptions and timing. Whenever possible, the exact values of the timing configurations will be given, however—due to the nature of the circuitry—this will not always be possible. The processor address lines and both sets of data busses are under program control and—unless one has an actual program to read and follow—the signals will appear more or less random. If possible, the activity appearing at a particular test point will be described in terms of a known game condition (i.e. the credit mode) and compared with the associated TV monitor display. In some cases, patterns of signals that would ordinarily appear random may be made to take on a predictable structure once the number of program parameters and interrupts has been reduced to a minimum. In other cases, blocks of circuitry may be tested by forcing inputs to known states and watching for predictable results in other areas of this system. Wherever such techniques are usable, they will be given.

3.2 General Test Procedures

3.2.1 Data Bus Procedures. As we have mentioned several times before, there are two sets of data busses used to carry information between the mother and game PCBs. The multiplex data bus carries information from the game PCB to the mother PCB while the output data bus carries information in the reverse direction. Both data busses are tri-state meaning that their signal levels may be HI, LO or "floating." The activity of both sets of busses is highly complex at all times during the course of the game. An oscilloscope applied to any data line will show only general chaos and a logic probe will merely reveal HI, LO and PULSING for all game modes and states. While this may be helpful information in a few cases (i.e. if a line stays LO constantly), the logic probe will, in general, not yield conclusive information.

3.2.2 Game Data. Troubleshooting the flow of game data over the multiplex data bus involves testing the opto-isolators, the Schmitt triggers and the multiplexers. In the case of the opto-isolators and Schmitts, the game may be in any mode or configuration. Each input pin to the isolators must show about 12 volts when the corresponding player control switch not being activated. Operating a switch (torpedo trigger switch for example) will cause the corresponding isolator output to drop to ground (A1-6 in this case). The isolator output will respond by outputting the same level as applied to the input, but at TTL logic levels. An open switch will force a TTL HI from the output while a closed switch will force the output LO. Opposite states must be detected at the outputs of the appropriate Schmitt triggers, as these devices also invert the signal level.

Troubleshooting the multiplexers is a more difficult task and it may be less time consuming in some cases to simply replace a suspected device. However, these devices may be tested in the following manner if desired. Processor bits A^8 and A^9 may be isolated from the CPU and placed in a floating condition by removing the game PCB from the mother PCB edge connector and taping edge connector pins K and L. Then re-insert the game PCB. A multiplexer address of 00 can be selected by grounding both select lines simultaneously. The signal states at output pins 7 and 9 must match the input signal levels at pins 6 and 10 respectively. For example, the input pins may be forced HI while the output pins are monitored to see that they also output HI logic levels. After all the multiplexers have been tested with an address select of 00, the grounding clip lead on pin 14 of the chip may be removed. The input signal will be regarded as a HI and the multiplexer address will increment to 01. Again, output pins 7 and 9 may be observed for a matching state with respect to input pins 5 and 11 respectively. In a similar manner, all four binary address select combinations may be constructed at the multiplexer select inputs and the outputs checked to see that they are being properly selected.

3.2.3 Shifter Procedures. In general, game PCB problems are those associated with sound, coin counting and the player's controls. This last category includes proper game timing, credit score value, torpedo aim position, triggering of the torpedo and ship/mine motion. Problems which involve objects which are supposed to move but do not may very likely be centered in the shifter circuitry.

An extremely detailed discussion of how to test the shifter circuit components will not be presented here since most of this information has already been discussed previously. The procedure for testing latches has been thoroughly discussed and the shifters themselves are tested in very much the same way as the multiplexers except that the data is shifted instead of merely being reflected. However, there is one area which does merit an in-depth discussion and this is deciding whether it is the shifter or the CPU circuitry which is the problematic area.

The shifter is active only for images which undergo horizontal motion. This includes the ships and also the underwater mines. All the other images (i.e. words) do not go through shifter modification so problems with motionless or vertically moving images do not involve this area of circuitry. In addition, the game board generally will not cause gross distortions in the images. If the entire picture rolls (either vertically or horizontally) or is broken up diagonally, the TV monitor may need adjustment or sync from the CPU may be malfunctioning.

Problems that arise in the shifter circuitry generally fall into two categories: data and control. Data bits to the shifter make up the actual image on the screen and if any of them are missing or incorrect, the effect will be very obvious. Ship images with black or bright white vertical stripes, broken up arrays of dots where a coherent image should be or even solid blocks of white (or black) may very well turn out to be shifter problems involving the flow of shifted data. These malfunctions will appear with the above symptoms while everything else on the screen appears perfectly normal and, in many cases, even the motion of the distorted image will be normal.

Control problems in the shifter will most likely result in jerky or incorrect motion of an otherwise correct and coherent image. Shifter codes which do not change will result in motion of eight bit jumps which will be very noticeable on the CRT. Single held bits in the shifter codes will cause slightly jerky motion or even a wavering of the image and this effect may be evidenced by a "three steps forward, two steps back" sort of motion. In some cases, motion may be completely absent and we have even seen instances where the image left replicas of itself side by side in its wake as it moved across the screen. These replicas may appear as whole images, parts of the image or even as a solid white bar across the CRT.

3.3 Control and Load Signals

3.3.1 Test Point 118 (Sound Load); F5-9. A normally HI level signal, this line pulses LO twice each time it becomes active. Pulse width when LO is 500ns and spacing between pulses is about .5 sec. The signal will exhibit a pair of pulses under the following conditions:

1. Each time the coin switch is closed.
2. In the game mode, each time the topedo trigger switch is closed. There is about a half second delay from the time the switch is closed to the occurrence of the first pulse.
3. Each time in the game mode that a torpedo strikes a mine or ship.
4. Following the closing of the coin switch, and during the reload mode, several pairs of pulses may be seen. These pairs activate the sonar sound. Time between pulse pairs is about a second and the number of pairs occuring at a particular sounding is a program variable.
5. In the game mode, each time the cruiser appears on the screen three sets of pulse pairs will occur. These three pairs activate the dive sound three times. Time between pulse pairs is about a second.

3.3.2 Test Point 119 (Periscope Light Load); F3-9. Normally HI, TP-119 pulses LO only in the game mode and each time the torpedo switch is closed. Again, the pulses occur in pairs, each pulse being 500ns wide and .5 sec. apart. On the firing of the fourth torpedo in a set, a third pulse will appear. This third pulse will occur 2 seconds after the initial pulse of the fourth pulse pair. Closing the torpedo switch between the first member of the fourth pulse pair and the third pulse in this set will not result in the signal line dropping LO, as the system is in the reload mode and will not accept any trigger inputs until the occurrence of the third pulse. There will also be some activity on this line when the cruiser appears on the screen during the game mode.

3.3.3 Test Point 120 (Explosion Light Load); F1-9. A normally HI signal, this line will pulse LO only in the game mode when a torpedo strikes a ship. Pulses will occur in pairs where the width of each pulse is 500ns and the spacing between them is about a second. There will also be some activity on this line when the cruiser appears on the screen during the game mode.

3.3.4 Test Point 121 (Shifter Address Load); E5-9. The attract mode is the best time to examine this signal as the presence of only one horizontally moving object makes the signal activity much more patterned and consequently much easier to distinguish. This line is normally HI, becoming active only when a horizontally moving object appears on the screen. The exception to this occurs in the attract mode when the torpedo is also sent through the shifter. For a single horizontally moving ship on the screen, LO pulses will be 500ns wide and spaced 32.5ms apart. As additional left/right moving objects appear on the screen, the signal activity will become increasingly more complex.

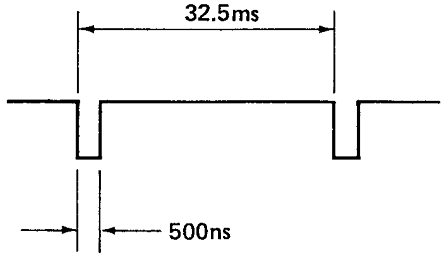


FIGURE 3.3.4-1 Test Point 121

3.3.5 Test Point 122 (15-Bit Latch Load); C5-9. Normally HI, this signal becomes active only when horizontally moving objects appear on the screen. Again, the attract mode is the simplest time to verify the operation of this signal. LO pulses are gated in five pulse groups, each pulse being 500ns wide with 30μs between pulses. Groups of five pulses occur every 200μs. The activity at this point becomes more complicated as more objects appear on the screen.

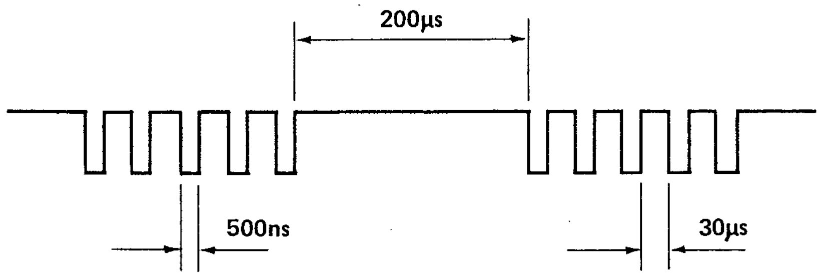


FIGURE 3.3.5-1 Test Point 122

3.3.6 Test Points 123, 124 & 125 (Address Bits A^8, A^9 & A^{10}). The activity on these lines is completely under program control and will be difficult to distinguish from random signals. There is a basic processing pattern to each of the address bits, however, and these are shown in Figures 3.3.6-1, 3.3.6-2 and 3.3.6-3. These photographs were taken while the game was in the attract mode and processing activity was relatively simple. Any deviation from the attract mode serves to complicate the processing and hence the complexity of the address line activity. These photographs are included as test points to indicate that the activity is in fact patterned, although complex, and each pattern bears features distinguishable from the other two. However, the only generalization which can be made about these points is that rapid, nearly random activity will appear as soon as the game is powered up.

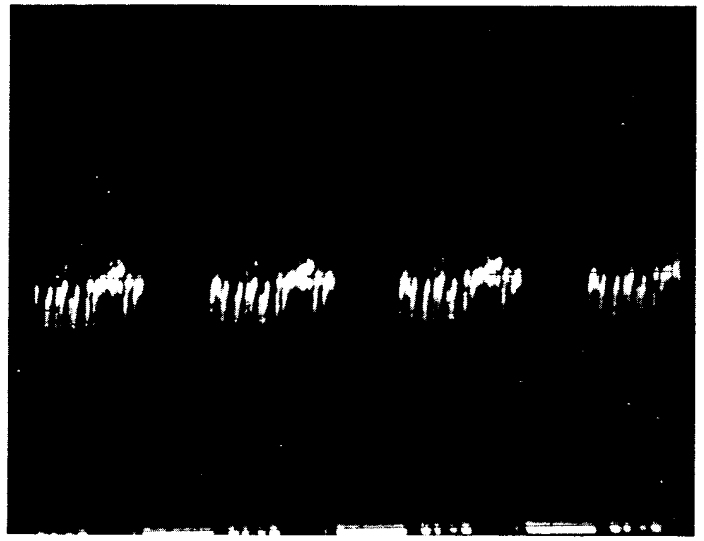


FIGURE 3.3.6-1 Test Point 123

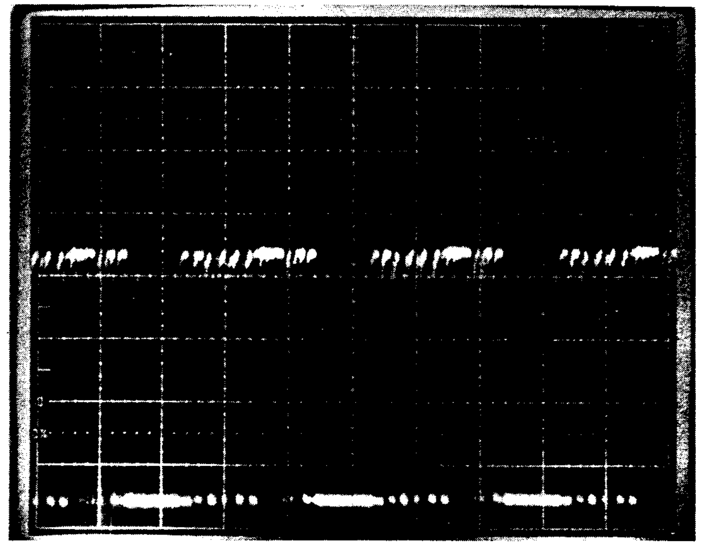


FIGURE 3.3.6-2 Test Point 124

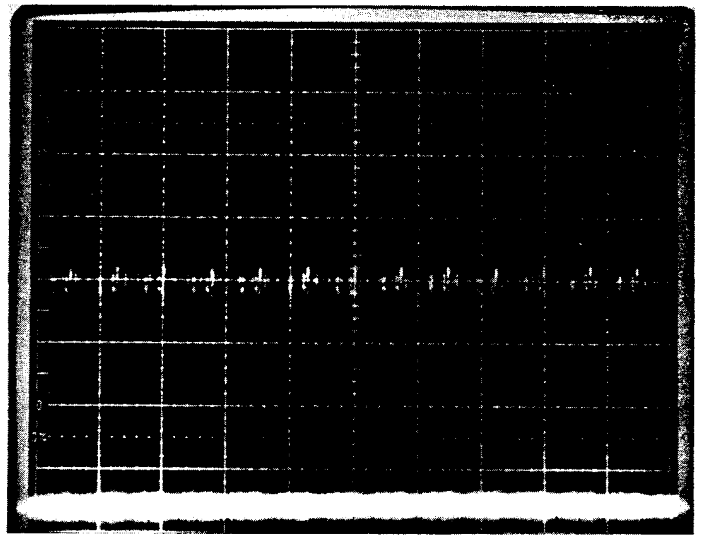


FIGURE 3.3.6-3 Test Point 125

3.3.7 Test Point 126 (The Sample Line); D1-3. SAMPLE is the CPU write control to the game board and is an active HI signal. As with the previous test point, the activity on this line will be difficult to distinguish from random as the number of game parameters increases. Activity in the attract mode with only a single ship on the screen is identical to TP-122.

3.4 Output Data Signals

3.4.1 Test Point 101 (Mine Sound Trigger); G5-3. Normally a TTL LO level signal, this line will go HI for approximately one-half second whenever a torpedo hits a mine.

3.4.2 Test Point 102 (Sonar Sound Trigger); G5-5. Also an active HI signal, the sonar trigger will activate the sonar sound whenever the CPU enters the game mode, and during the reload pause following the firing of a volley of four torpedoes. Duration of the activity is a program variable and ranges from about one-half second to four or five seconds.

3.4.3 Test Point 103 (Dive Sound Trigger); G5-9. This line will pulse HI for a half second each time the dive sound is to occur (i.e. whenever the cruiser appears on the screen). The dive sound occurs three times in rapid succession. Frequency of these HI pulses is just under one second.

3.4.4 Test Point 104 (Torpedo Sound Trigger); G5-11. One-half second is the pulse duration of this active HI signal and the activity occurs each time the torpedo is fired during the game mode. This pulse may be cut short if the torpedo strikes an object before the half second interval has elapsed.

3.4.5 Test Point 105 (Ship Explosion Sound Trigger); G5-13. Normally LO, this signal becomes active for about half a second each time a torpedo strikes a ship during the game mode.

3.4.6 Test Point 106 (Coin Switch Activation); G5-1. Normally LO, this line pulses HI for one-half second whenever the coin switch is operated.

3.4.7 Test Point 107 (Torpedo Light Control Bits); F3-2, 5, 7 &10. These lines will all be at a HI level during the attract mode when no torpedo lights are to be illuminated. During the game, pin 2 (torpedo light 1) will drop LO after the first torpedo is fired and will remain LO until the completion of the reload cycle. Similarly, pin 5 will go LO after the second torpedo is fired, pin 7 after the third and so on. When these signals go LO, the base of the associated lamp driving transistor is grounded. All lines return to the HI level when the reload cycle is completed.

3.4.8 Test Points 108 & 109 (Ready & Reload Light Controls); F3-12 & F3-15. These lines are complementary in logic state; one being HI necessarily implies that the other is LO. However, this is not true during the attract mode when both lines are in their inactive LO levels. During the game mode, F3-15 will rise HI and F3-12 will drop to ground after the firing of the fourth torpedo. This configuration will remain for about two seconds during which time torpedo firings will not be recognized by the processor. At the end of the two second reload period, the line at TP-109 will again go LO while TP-108 will again rise to its HI level.

3.4.9 Test Points 110–117 (Explosion Light Control Bits); E1-3, 6, 11 14; F1-2, 7, 10 & 15. TPs 110-113 are active HI level signals which turn on their associated driving transistor and complete a ground path for four of the sixteen lights. Figure 3.3.3-1 outlines which lights are turned on by the particular combination of bit states at E1 and F1. Activity in each case lasts for one second.

The decoding of the explosion lights is governed by two factors: (1) the position of the periscope when the ship hit occurred and (2) the direction the ship was moving when hit. There are eight possible positions used to decode the light matrix, position 1 being on the extreme left side of the screen. These eight positions, along with the two directions possible for ship movement, decode to produce the sixteen possible explosion light combinations outlined in the table.

PERISCOPE POSITION	LEFT/RIGHT SHIP MOVEMENT	TEST POINTS ACTIVE
1	L	112, 113, 114
1	R	111, 112, 114
2	L	112, 113, 115
2	R	111, 112, 115
3	L	112, 113, 116
3	R	111, 112, 116
4	L	112, 113, 117
4	R	111, 112, 117
5	L	112, 114
5	R	110, 112, 114
6	L	112, 115
6	R	110, 112, 115
7	L	112, 116
7	R	110, 112, 116
8	L	112, 117
8	R	110, 112, 117

FIGURE 3.3.3-1 Explosion Light Decode

3.5 Input Data Signals

3.5.1 Test Points 127 & 128. TPs 127 and 128 describe the operation of a single opto-isolator. Be aware that all the others operate identically. TP-127 will normally be at a V_{LED} level (about 12V) and will drop to ground whenever the torpedo trigger switch is closed. It will remain grounded until the switch is released and V_{LED} again appears at the isolator input. At the output test point (TP-128), the signal is non-inverted however it is converted to TTL levels so that the V_{LED} at the input appears as a TTL HI at the output. This level drops LO whenever the torpedo switch is closed.

3.5.2 Periscope PC Board. The periscope position code requires a bit of explanation. The periscope position is encoded in a 5-bit word derived from a wiper connector to ground which slides over a printed circuit card containing a binary pattern of contacts. This card is illustrated in Figure 3.3.2-1. Each row of printed contacts is either shorted to ground by the wiper or held to V_{LED} by the voltage on the input pin of its associated opto-isolator. Each row of contacts represents a single bit in the 5-bit periscope position word. Each bit is either HI or LO depending on the position of the wiper. The five rows are input to their respective opto-isolators where their operation becomes identical to the trigger input just discussed.

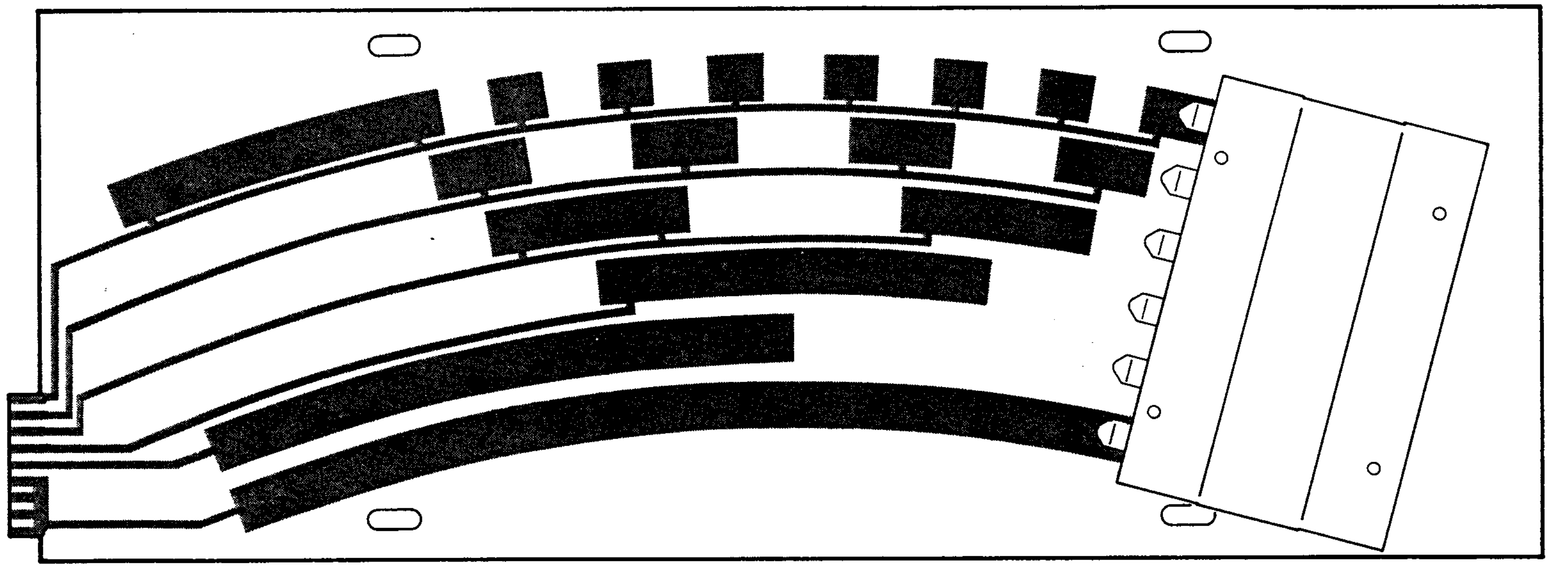


FIGURE 3.3.2-1 The Periscope PCB

SWITCHES 1 & 2	NUMBER OF TIME UNITS
SWITCH 3	NUMBER OF COINS
SWITCH 4	NUMBER OF PLAYS
SWITCH 5	ERASE HIGH SCORE
SWITCHES 6, 7 & 8	SCORE ADJUSTMENTS

FIGURE 3.5.3-1 Function Of The DIP Switches

3.5.3 Operator Switches. The DIP switches in this part of the circuit are set by the operator of the game to govern parameters such as the number of plays per coin, etc. The action governed by each switch is listed in Figure 3.5.3-1. If the switch is in the off position, the voltage appearing at the input of the multiplexers from this switch will be provided solely by the pull-up resistor in the resistor pack at PCB location G3 and will be interpreted as a TTL HI level. Closing the switch grounds the associated multiplexer input and generates a TTL LO level.

3.6 Sound PCB Power & Noise Generation

3.6.1 Test Point 31 (Filtered AC). Unregulated DC voltage level at the (+) side of the large 1000μF capacitor must be approximately 28 or 29 volts with an AC component of about 40mV.

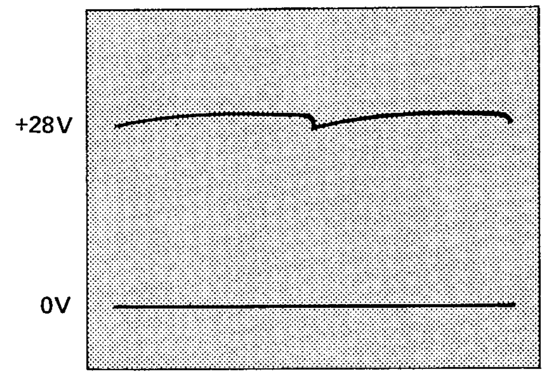


FIGURE 3.6.1-1 Test Point 31

3.6.2 Test Point 32 (Zener Diode Voltage). DC voltage level across the Zener diode must be the Zener voltage of the device which is about 16.6 to 16.8 volts. There should be no measurable AC component.

3.6.3 Test Point 33 (Power Supply Bus). Regulated DC voltage for use by the sound board is about 16V with no measurable AC component.

Whenever more than one sound in the system is not working, the first test point to check is the Vss supply bus. If this test point is functioning properly, the power supply may be assumed to be fully functional. If TP-32 does not correspond to the description above, the next step is to check TP-30 to insure that AC power is properly connected to the board and that the rectification diodes and filter capacitor are working properly. If these areas check out, TP-31 will verify the operation of the Zener diode and limit the possible fault locations to Q28 or the diode across the base-emitter junction of this device.

3.6.4 Test Point 34 (Base Of The Noise Generator). The DC level at this point will be in the neighborhood of the .6V diode drop across the base-emitter junction. There is a slight AC component of random white noise, however it is immeasurable.

3.6.5 Test Point 35 (Noise Generator Output). Figure 3.6.5-1 shows the AC component at the collector of Q27. Peak-to-peak level should be on the 400mV range. The DC bias level at the collector of Q7 must be about 9V.

If the sound problem encountered is characterized by the proper switching action but improper or generally poor tone, the output of the noise generator is a logical place to check. This is not true of the torpedo sound, however, since this sound does not employ noise from Q27.

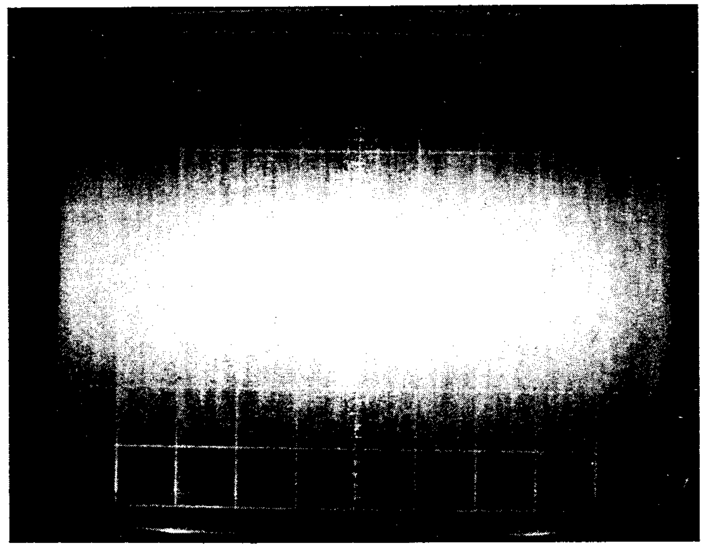


FIGURE 3.6.5-1 Test Point 35

3.7 Explosion Sound Signals

3.7 Explosion Sounds. Test point information is given here only for the mine sound. Voltages and timing are assumed to be identical for the explosion unless otherwise noted.

3.7.1 Test Point 1 (Base Of The Digital Switch). With no digital input, the edge connector pin to the switching transistor floats and shows only a Vss DC level. With a digital signal provided, or with the edge connector pin held to ground with a clip lead, the DC level on the base of the transistor Q1 (Q3 for ship explosion) should be about .6V below Vss.

3.7.2 Test Point 2 (Timing Capacitor). The positive side of C1 with no digital input is normally at ground. The voltage at this point will rise to Vss with the application of a ground at the sound edge connector pin and will remain there until the pin is again allowed to float. At this point, the DC level across the capacitor will discharge to ground. Charging time for the mine capacitor C1 is about 5ms, however discharging requires several seconds. These times are different for the ship explosion, in which case less time is required to charge the cap and more time is required to discharge it. The difference is approximately 50%.

3.7.3 Test Point 3 (Base Of The Noise Amplifier). With no LO at the edge connector pin, the DC level at the base of Q2 (Q4 in the ship explosion circuit) will be about 20mV above ground with an AC component of random white noise approximately 40mV peak-to-peak. With the application of a momentary ground to the edge connector pin, the DC level will rise to about 1.1V at the timing capacitor charging rate. It will fall back to its 20mV level as the capacitor discharges. AC component values remain constant.

3.7.4 Test Point 4 (Explosion AC Output). With no noise being output, this test point will be at ground potential. Following the application of a momentary ground at the edge connector pin, the AC sound output will be as shown in Figure 3.7.4-1.

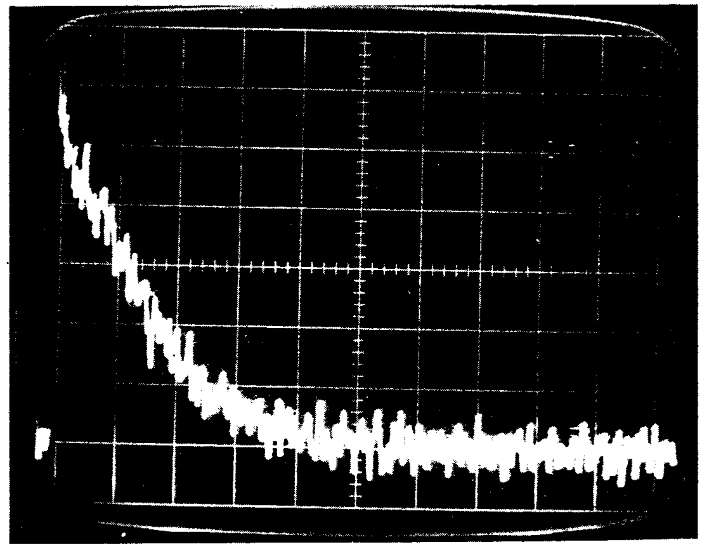


FIGURE 3.7.4-1 Test Point 4

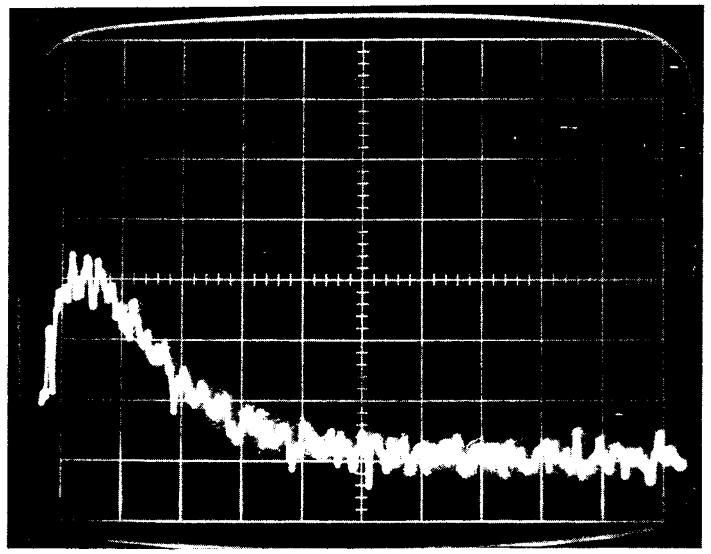


FIGURE 3.7.4-2 Test Point 4

3.8 Torpedo Sound Signals

3.8.1 Test Point 6 (Torpedo Noise Generator). The collector of Q7 will always show a DC level of nearly 8V with a high frequency random noise component of 20mV peak-to-peak. The base of this transistor operates at a bias voltage of about 13.6V above ground with a similar noise signal. Noise on the base is immeasurable.

3.8.2 Test Points 7, 8 & 9 (Switching And Pulse Shaping). The torpedo sound circuit is identical to the explosion circuits except that the frequency of the noise is different. As a result, TPs 7, 8 and 9 will be identical to TPs 1, 2 and 3 respectively.

3.8.3 Test Point 10 (Torpedo AC Output). With no LO at edge connector pin 11, this test point will remain at ground potential. When the pin 11 is grounded, the AC output will be as shown in Figure 3.8.3-1. This photograph was taken with the torpedo sound set at maximum volume, as were all the sound output photographs.

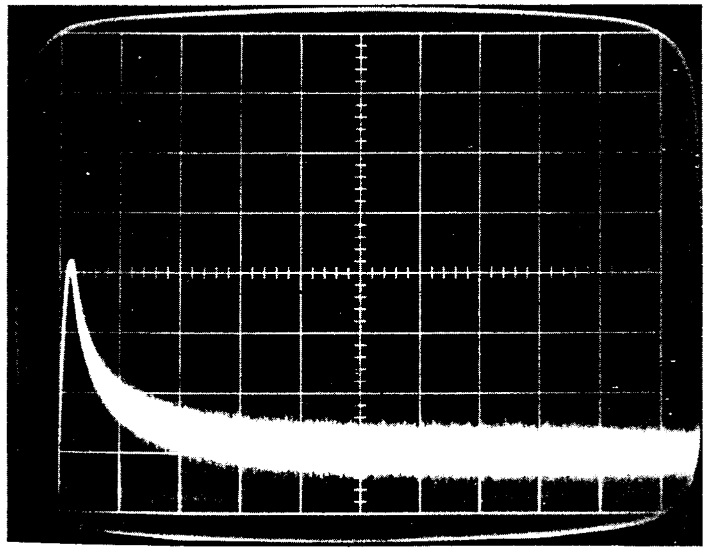


FIGURE 3.8.3-1 Test Point 10

3.9 Sonar Sound Signals

3.9.1 Test Points 11 & 12 (Sonar Pulse Generator). The output at the collector of Q9 is normally a Vss level which drops to .6V above ground for about one-third of a second. The rate at which these LO-going pulses occur may be adjusted via the sonar rate pot. This rate may be adjusted from a frequency of two-thirds of a second to about one second. TP-12 will show identical characteristics but will be 180° out of phase.

3.9.2 Test Point 13 (Collector of Q11). With the sonar tone control set to its fully grounded position, the signal at this point will be .6V above ground and pulsing to Vss for .4ms every 1ms. By rotating the tone pot away from ground, the frequency of the pulses may be closed up to about 500Hz (frequency is determined by noise level). The amplitude will similarly decrease on the positive pulses, closing down to about one-half the Vss level as a minimum. Figure 3.9.2-1 displays the behavior at this point with a mid-range tone pot setting.

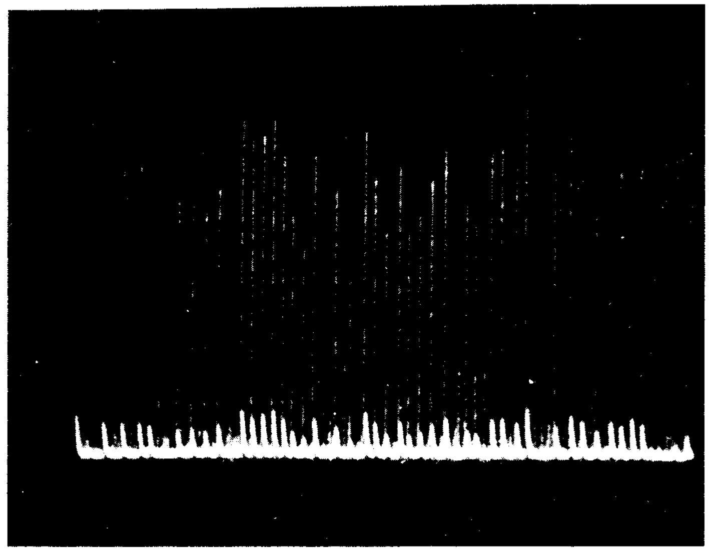


FIGURE 3.9.2-1 Test Point 13

3.9.3 Test Point 14 (Collector of Q12). With the tone control set to its fully grounded setting, the signal at this test point will be at a Vss level and pulsing to .6V above ground for about .4ms every 1ms. By rotating the tone control away from ground, the frequency of this signal can be closed up to about 500Hz. A mid-range setting of the tone control results in a signal of constant amplitude and a frequency varying randomly between 1KHz and 500Hz.

3.9.4 Test Point 15 (Base Of Q13). Normal DC bias level is the .6V base-emitter drop. This point will pulse sharply to a voltage below ground (i.e. as low as -7 volts) for about 8ms. The negative spikes correspond to the falling transitions coupled in from TP-11. The waveform of the spike is basically a ramp.

3.9.5 Test Point 16 (Amplified Falling Transistion). This point is normally at a level near ground and pulses to about 7.5V for 8ms. Pulses correspond to falling transitions at TP-11. The waveform is basically a ramp and is shown in Figure 3.9.5-1.

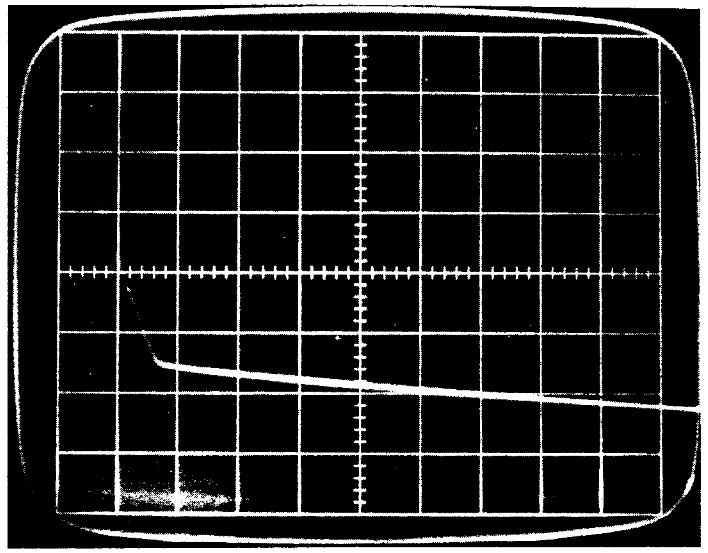


FIGURE 3.9.5-1 Test Point 16

3.9.6 Test Point 17 (Combined Transistion Voltages). A normally ground level signal, capacitor C4 is charged initially to 7.5V by the amplified falling transitions, then recharged a short time later by a 2.5V ramp representing the rising edge of TP-11. The time between initial charge and recharge is fixed, while the time between successive 7.5V charges is variable by adjusting the rate pot.

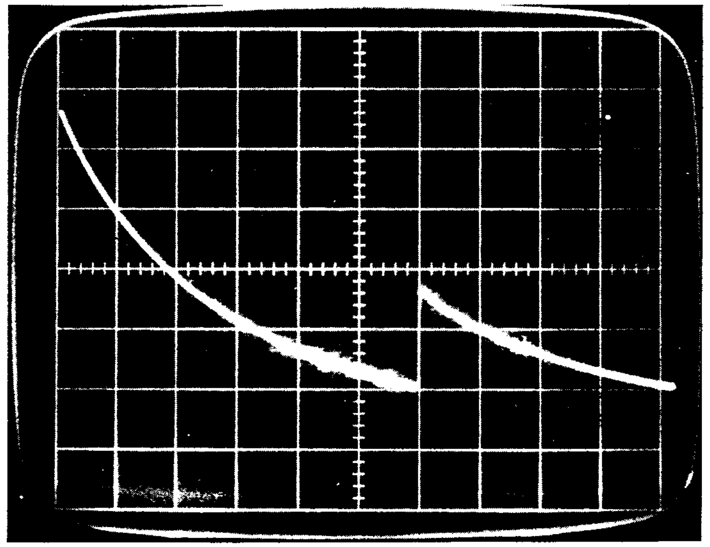


FIGURE 3.9.6-1 Test Point 17

3.9.7 Test Point 18 (Pulsed 1KHz Tone). The signal at this point is simply a periodically-pulsed 1 kilohertz tone.

3.9.8 Test Points 19 & 20 (Digital Switch & Timing Cap). TP-19 is identical to TP-1 in responding to a LO at edge connector pin 3. TP-20 is similar to TP-2, the only difference being slight variations in the charge and discharge rates. TP-20 requires several milliseconds to charge and several seconds to discharge.

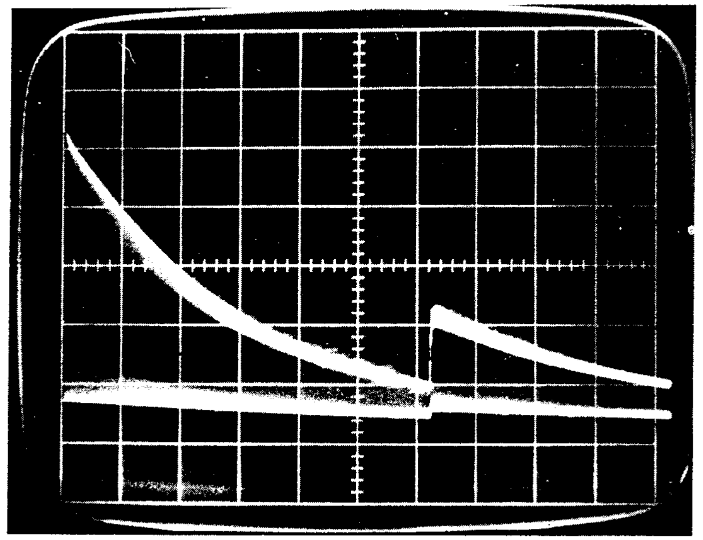


FIGURE 3.9.8-1 Test Points 19 & 20

3.9.9 Test Point 21 (Collector of Q6). With no digital input applied to the edge connector pin, this level will be about 1V, turning on Q15 and grounding the audio bus. With edge connector pin 3 pulled to ground, the collector of Q16 will drop to about .6V above ground and remain there for the entire discharging time of the capacitor. After the cap times out, the signal at TP-21 will jump back to its 1V level.

3.9.10 Test Point 22 (Sonar AC Output). Normally held nearly to ground, the application of a momentary ground to edge connector pin 3 will cause the sonar sound pulses to occur as they are shown in the photograph below. These shaped tone pulses will continue to occur until the timing capacitor discharges, at which point they will suddenly cease.

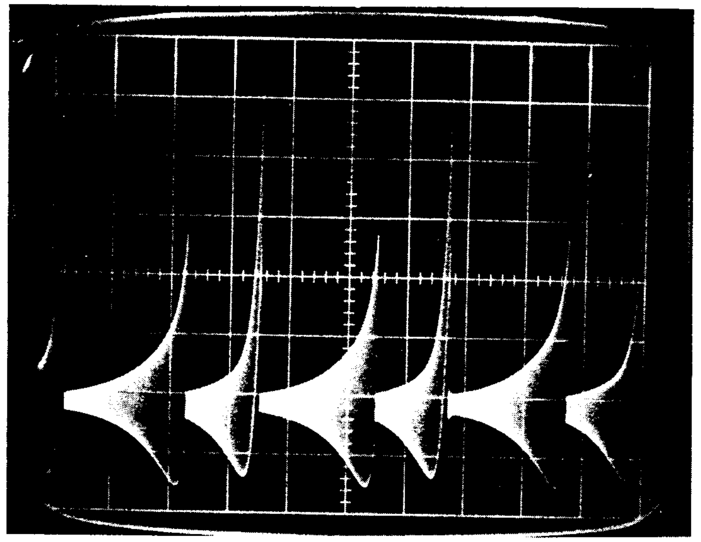


FIGURE 3.9.10-1 Test Point 22

3.10 Dive Sound Signals

3.10.1 Test Point 24 (Timing Capacitor C6). Normally at a ground potential, the voltage across this device rises to about 6.8V with the application of a LO at edge connector pin 12. Charging time is about a second and the digital signal must be maintained for this period of time. Upon removal of the LO, the capacitor discharges to ground ayain and discharge time is also approximately a second.

3.10.2 Test Point 25 (Collector of Q21). Q21 outputs square wave pulses every 20ms. The signal is normally at a .6V level above ground, the pulses rise nearly to Vss and they last for 4ms. With the application of a digital signal at the edge connector pin. The amplitude of the pulses begins to fall in response to the rising voltage on C6. The frequency of these pulses also begins to close up as C6 charges. With C6 charged to its maximum value, pulse amplitude is nearly zero and the frequency just before close-off of pulse amplitude is just about 10ms.

3.10.3 Test Point 26 (Collector of Q26). At this test point, the normal DC level is about 14V pulsing to .6V above-ground for 4ms every 20ms. When the edge connector pin for the dive sound is grounded, the frequency of the LO-going pulses closes up with the charging of C6 and reaches a minimum value of 10ms with C6 fully charged. Amplitude of the pulses remains constant.

3.10.4 Test Point 27 (Collector of Q23). The DC level at this point is about 3V. The AC component riding on this DC level is a 5V peak-to-peak sine wave combined with sharp 10V spikes corresponding to the rising and falling transistions coupled in from the pulse generator. The AC at this test point is clearly defined by the oscilloscope photo below.

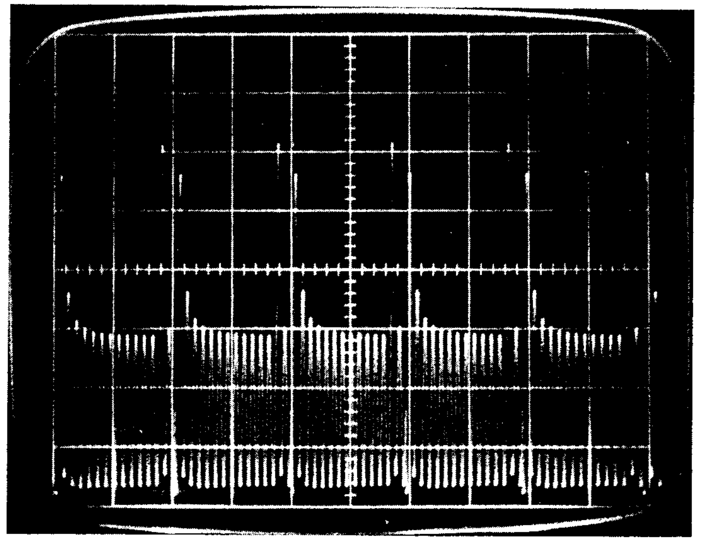


FIGURE 3.10.4-1 Test Point 27

3.10.5 Test Point 28 (Dive Sound Prior To Final Amplification). The signal at this point rides on a DC level of .6V above ground, or the test point will be grounded depending on the state of Q25. The AC signal when this point is not grounded is shown in Figure 3.10.5-1.

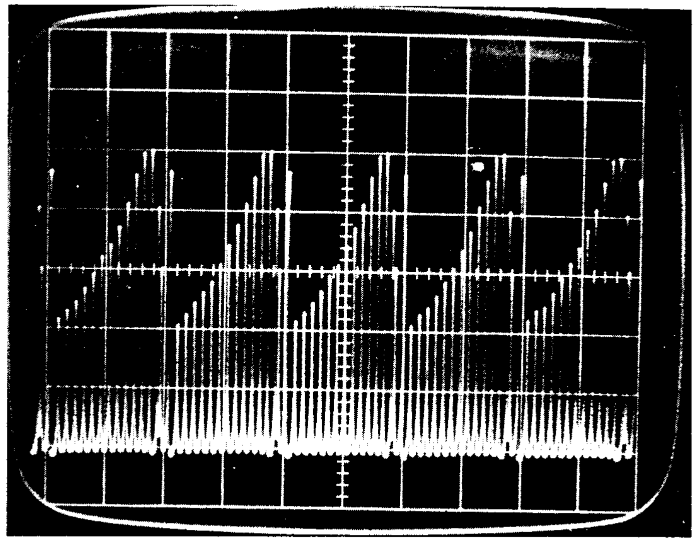


FIGURE 3.10.5-1 Test Point 28

3.10.6 Test Point 29 (Digital Switch). The characteristics at this test point are identical to those described for TP-1.

3.10.7 Test Point 30 (Timing Capacitor C7). This capacitor is normally held at ground potential and rises to about 5.5V with the application of a digital input at the edge connector pin. Charging and discharging time is about a half second each which result in the sound being output for one second following the grounding of the edge connector pin. Note that the LO level must be maintained at the sound input for the full half second charging time of C7.

3.11 Summing & Power Amplifier Signals

3.11 Summing And Audio Amplifier Signals. The summing and audio amplifier essentially provides only AC gain, the AC waveforms found in these regions of the circuit being the same as those output from the separate sound generators. The summing amp is an emitter feedback amplifier, hence the bias levels are quite critical. DC normally found on the base of Q29 is about 2V while that found on the collector of this transistor is typically 8V. AC signals input to the amplifier are at a peak-to-peak level dependent on the particular sound volume setting, but 20mV or 30mV on the base of the amplifier is about maximum. The transistor circuit will boost this to a 6 or 8 volt peak-to-peak level. Capacitive coupling and attentuation onto the master volume control will reduce this to about 2.5V for input to the audio amplifier (master volume set to the maximum level). Maximum allowable AC swing-out of the amplifier is fixed by the integrated Vss or about 28V peak-to-peak before clipping sets in. The oscilloscope photograph (Figure 3.11-1—next page) shows the sonar sound at full volume being input to the audio amplifier and is included here as an example of typical AC peak-to-peak levels as well as AC symmetry with respect to ground after capacitive coupling out of the sound generator.

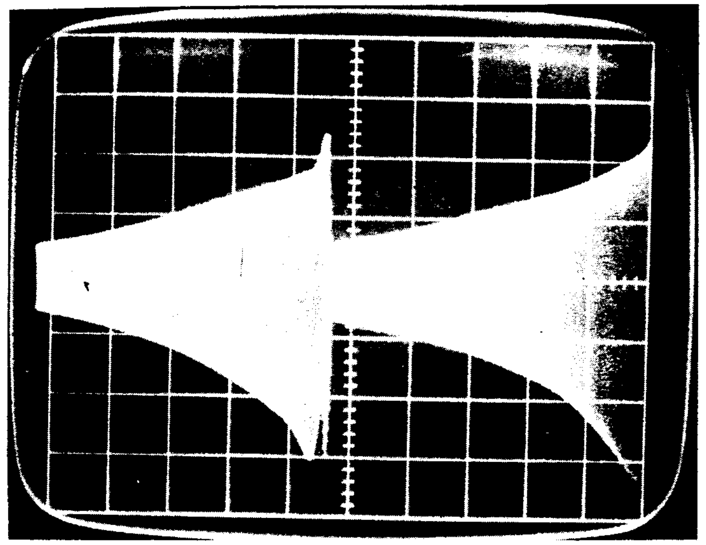


FIGURE 3.11.1-1 Summing And Power Amplifier

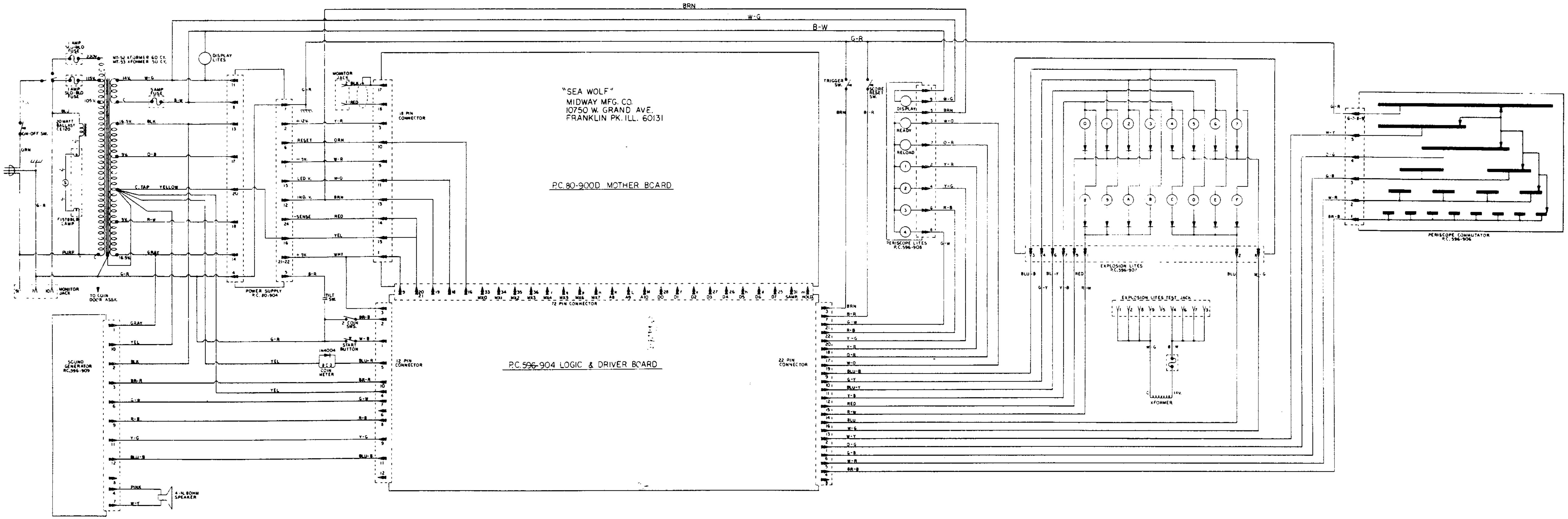


FIGURE 3.12.1-1 Sea Wolf Wiring Diagram

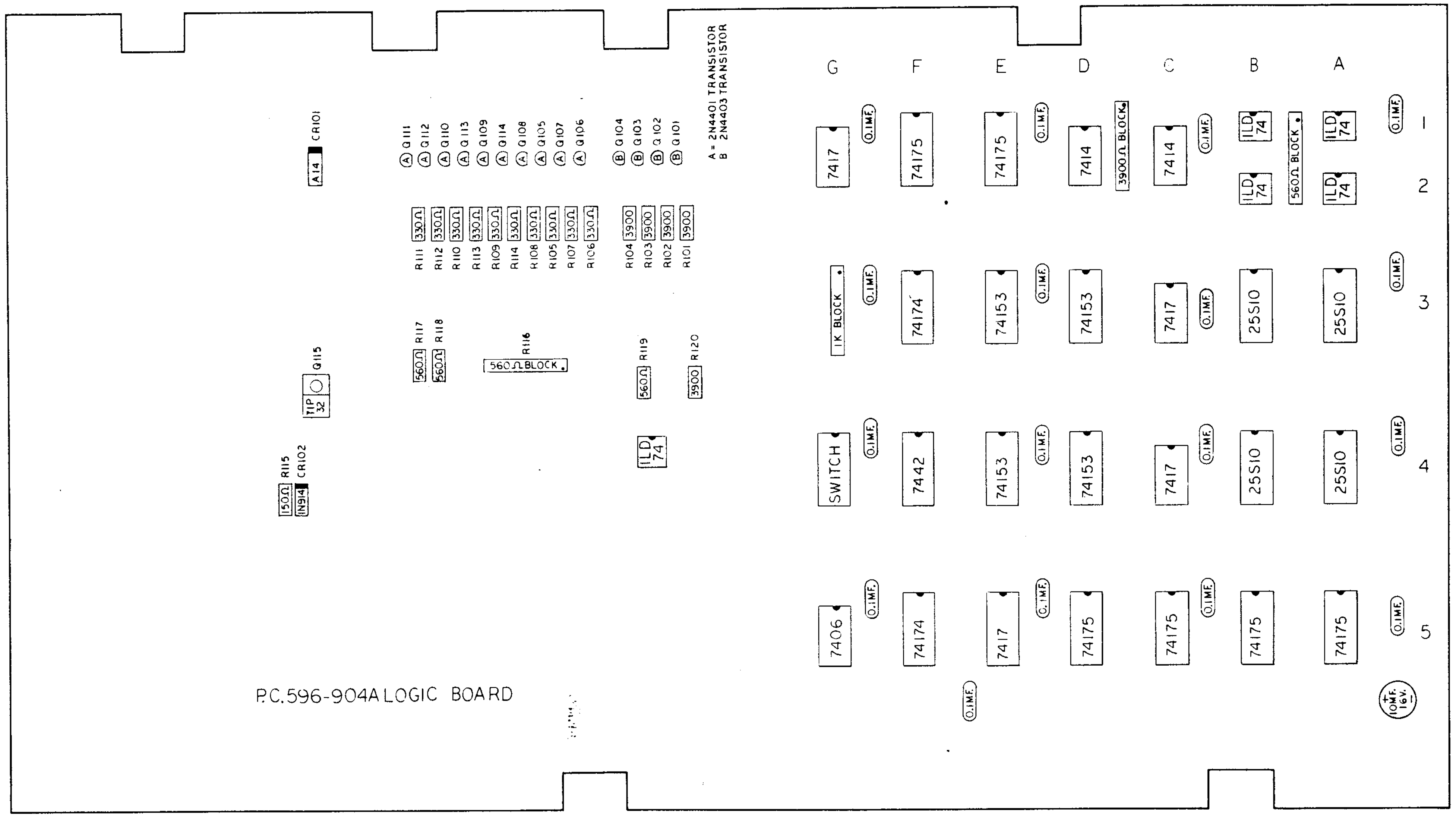


FIGURE 3.12.1-2 Sea Wolf Digital PCB Component Diagram

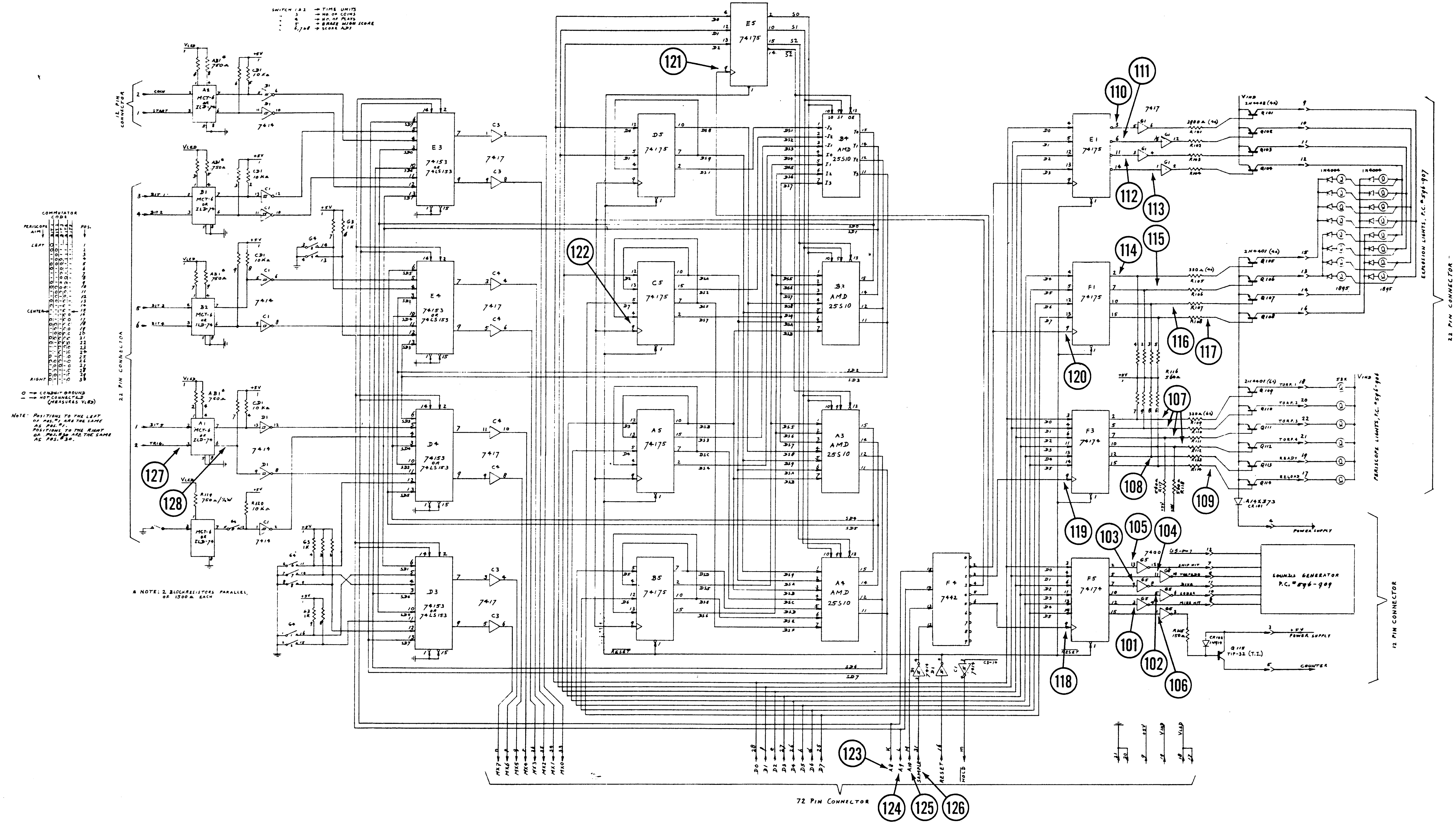


FIGURE 3.12.1-3 Sea Wolf Digital Schematic