CAR POLO

EXIDY
SERVICE
MANUAL

PART TWO
TECHNICAL DATA
TABLE OF CONTENTS

1. Removal of steering control module .......... 1
2. Removal of PCB ____________________________ 1
3. Removal of monitor __________________________ 2
4. Theory of operation _________________________ 3

FIGURES

1. System Block Diagram ______________________ 7
2. Memory Map _______________________________ 8
3. Schematics ________________________________ 9
REMOVAL OF A PLASTIC STEERING CONTROL MODULE

PROCEDURE:
1. Disconnect game from AC power outlet.
2. Open coin door on control module side.
3. Unbolt the four (4) hex through bolts from the underside of the module.
4. Disconnect the harness from the module.
5. Lift the module out and up from the game cabinet.

TO REMOVE P.C.B.

1. Disconnect power to game.
2. Open door to blue car side of cabinet.
3. Remove cash box from blue car side.
4. Remove 2 hex 7/16 bolts located behind cleat at the back of the cash box compartment.
5. Slide foot box out of game. (NOTE: Foot pedal harness is still connected. Caution should be taken to protect this harness.)
6. Facing the P.C.B. mounted on the floor of the cabinet, remove the connectors from the board.
7. Pivot the two metal tabs that secure the P.C.B. in the slotted mount up and out away from the board.
8. Remove the P.C.B. from its mount by sliding it towards you.
REMOVAL OF THE MONITOR
(See Figure 14)

NOTE: Removal of the color monitor from the game cabinet is accomplished by sliding the monitor out one end of the cabinet on the wooden side rails. To gain clearance for the monitor, the top glass bezel assembly and the blue team plastic steering control module must be removed from the cabinet.

PROCEDURE:

1. Disconnect game from AC power outlet.

2. Open the coin doors at both ends of the cabinet.

3. Remove the glass view plate parts: view plate frame (C), view plate gasket (D), glass view plate (E) and bezel (F), by unbolting eight (8) through bolts from the top of the cabinet.

4. Remove the plastic steering control module (G) from the blue team end of the cabinet by unbolting the four hex bolts located through the 2 x 2 end blocks under the control box. The harness to the module is disconnected and the module is lifted out and up from the cabinet.

5. Unbolt the metal monitor bracket and rack from the cabinet side rails by removing the two hex through bolts at either end of the cabinet rails.

6. Loosen the main harness from the side wall of the monitor on the blue team end of the cabinet.

7. Two persons can now slide the monitor out of the cabinet through the blue end side of the cabinet.

8. The monitor may be placed on its bottom or back for service. CAUTION: The CRT neck is in an exposed position and is protected by the monitor rack only when the monitor is placed on a flat clean surface.
THEORY OF OPERATION (Figure 1)

CAR POLO is a microprocessor game capable of simultaneous control of four cars, a ball and a special character (referee) on a color television screen. In addition, there is a capability of writing up to 256 alphanumeric characters on the screen.

The microprocessor used in CAR POLO is a MOS TECHNOLOGY 6502, 8 bit microprocessor. For more information on this processor, refer to MOS TECHNOLOGY PUBLICATION #6500-10A, MCS 6500 Microcomputer Family Hardware Manual. It is an inherent characteristic of this processor that it treats all peripherals attached to it as if they were memory. Therefore, all memory in this system has been mapped. (See Figure 2, Memory Map). The only random access memory assigned for use by the game program is the base page (addresses 0000 - 00FF). The game program uses this memory for storage of temporary data during the course of a game (example, car horizontal position). In addition, 256 bytes of random access memory (addresses 0100 - 01FF) are assigned to the processor for use as a stack.

The CAR POLO program is just slightly less than four kilobytes long and exists in prom from addresses F000 to FFC5. The top eight addresses are used to store the interrupt and reset vectors.

The processor writes alphanumericics on the screen by writing them into a random access memory at locations 3000 - 30FF. The video display logic then reads the data out of that ram, converts it to a video wave-train and displays it on the screen. The format for an alphanumeric byte is:

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCI Character</td>
<td>Color Select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSB</td>
<td>Bit Function</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The coding for the color is as follows:

- BIT 1 0
- WHITE 0 0
- GREEN 0 1
- BLUE 1 0
- RED 1 1

Car, ball and special character horizontal and vertical positions are controlled by writing them into the ram at addresses 4000 - 400F. The functions assigned to each ram location are:
4000  Player 1 Horizontal Position
4001  Player 1 Vertical Position
4002  Player 2 Horizontal Position
4003  Player 2 Vertical Position
4004  Player 3 Horizontal Position
4005  Player 3 Vertical Position
4006  Player 4 Horizontal Position
4007  Player 4 Vertical Position
4008  Ball Horizontal Position
4009  Ball Vertical Position
400A  Special Character Horizontal Position
400B  Special Character Vertical Position
400C  Players 1 & 2 Rotation
400D  Players 3 & 4 Rotation
400E  Ball Control Bits
400F  Special Character Bits

Once, at the beginning of each frame, the same logic halts the processor, extracts the data from the position portion of the ram and uses that data to preload the position counters. It then loads the rotation data into the rotation ram to be used in selection of the image to be displayed. The processor is then allowed to run again.

To display an image the horizontal and vertical positions and the rotation angle are multiplexed and combined to select the desired line in the image prom. The data from the prom is then loaded parallel into the shift register and shifted out serially at the instant when the image is to appear on the screen.

All of the images, including alphanumerics, are prioritized and converted into RGB color signals in the video generation circuitry, where, along with combined sync, they are sent to the color television to be displayed.

The processor detects collisions on the screen by the use of interrupts. The outputs of the shift registers are anded together in the interrupt detection and conditioning circuitry. If this circuitry detects two or more objects in the same place on the screen at the same time, (example: a car and a car collision) it sends out a pulse which interrupts the processor. The processor, at this time, knows that something has occurred, but doesn't know what. By reading memory location C000, it receives an eight bit word telling what type of interrupt it was. It then reads the interrupt data from the following locations:

A000  Ball and Border Data
A000  Ball and Goal Data
       Ball and Score Zone Data
A001  Interrupting Car Horizontal Position
A002  Interrupting Car Vertical Position
A003  Which Car Interrupted
A004  Car and Border Data
A005  Car and Ball Data
A006  Car and Goal Data
A007  Game Time Switch Setting

After servicing the interrupt, the processor clears the interrupts by writing into locations B000 to B007.

B000  Clear Ball and Goal Interrupt
B001  Clear Time Interrupt
B002  Not Used
B003  Clear Car and Car Interrupt
B004  Clear Car and Border Interrupt
B005  Clear Car and Ball Interrupt
B006  Clear Car and Goal Interrupt

Coin drops also generate an interrupt. Which player generated the coin drop is determined by reading location C000.

The processor interfaces with the outside world with two peripheral interface adaptors. These PIA's are wired to respond to addresses 5400, 5402, 5800 and 5802. The data which is interfaced through these devices is:

PIA #1, A Side (5400)

   Bit 0  Coin Counter Output
   Bit 1  Player 4 Crash (to Audio)
Bit 2  Player 3 Crash (to Audio)
Bit 3  Clear Steering Wheel Logic
Bit 4  Player 2 Crash (to Audio)
Bit 5  Score Pulse (to Audio)
Bit 6  Player 1 Crash (to Audio)
Bit 7  Ball Hit Pulse (to Audio)

PIA # 1, B Side (5402)

Bit 0  Strobe Speed Bits Out (to Audio)
Bit 1  Speed Bit 0 Out (to Audio)
Bit 2  Speed Bit 1 Out (to Audio)
Bit 3  Speed Bit 2 Out (to Audio)
Bit 4  Speed Bit 0 In (From Foot Pedals)
Bit 5  Speed Bit 1 In (From Foot Pedals)
Bit 6  Select Foot Pedal Assembly Output 0
Bit 7  Select Foot Pedal Assembly Output 1

PIA # 2, A Side (5800)

Bit 0  Player 4  Steering Input (Left or Right)
Bit 1  Player 3  Steering Input (Left or Right)
Bit 2  Player 2  Steering Input (Left or Right)
Bit 3  Player 1  Steering Input (Left or Right)
Bit 4  Player 4  Forward/Reverse Input
Bit 5  Player 3  Forward/Reverse Input
Bit 6  Player 2  Forward/Reverse Input
Bit 7  Player 1  Forward/Reverse Input

PIA # 1, B Side (5802)

Bit 0  Player 4  Active Output (to Audio)
Bit 1  Player 3  Active Output (to Audio)
Bit 2  Player 2  Active Output (to Audio)
Bit 3  Player 1  Active Output (to Audio)
Bit 4  Player 4  Steering Input (Wheel Moving)
Bit 5  Player 3  Steering Input (Wheel Moving)
Bit 6  Player 2  Steering Input (Wheel Moving)
Bit 7  Player 1  Steering Input (Wheel Moving)
<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF</td>
<td>FFFE - FFFF Interrupt Vectors</td>
</tr>
<tr>
<td>F000</td>
<td>F000 - FFC8 Game Program</td>
</tr>
<tr>
<td>EFFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>E000</td>
<td>Not Used</td>
</tr>
<tr>
<td>DFFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>D000</td>
<td></td>
</tr>
<tr>
<td>CFFF</td>
<td></td>
</tr>
<tr>
<td>C000</td>
<td>C000 Read Interrupt Vector</td>
</tr>
<tr>
<td>BFFF</td>
<td></td>
</tr>
<tr>
<td>B000</td>
<td>B000 - B00F = Clear Interrupt Data</td>
</tr>
<tr>
<td>AFFF</td>
<td></td>
</tr>
<tr>
<td>A000</td>
<td>A000 - A00F = Read Interrupt Data</td>
</tr>
<tr>
<td>9FFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>9000</td>
<td></td>
</tr>
<tr>
<td>8FFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>8000</td>
<td></td>
</tr>
<tr>
<td>7FFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>6FFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>6000</td>
<td></td>
</tr>
<tr>
<td>5FFF</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>5800 = PIA2A 5802 = PIA2B</td>
</tr>
<tr>
<td>4FFF</td>
<td>5400 = PIA1A 5402 = PIA1B</td>
</tr>
<tr>
<td>4000</td>
<td>4000 - 400F Game Data</td>
</tr>
<tr>
<td>3FFF</td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>3000 - 30FF Alphanumeric</td>
</tr>
<tr>
<td>2FFF</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>Not Used</td>
</tr>
<tr>
<td>1FFF</td>
<td>Not Used</td>
</tr>
<tr>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>0FFF</td>
<td></td>
</tr>
<tr>
<td>0100 - 01FF = Stack</td>
<td></td>
</tr>
<tr>
<td>0000 - 00FF = Base Page</td>
<td></td>
</tr>
</tbody>
</table>