SEGA/Gremlin

Data Bus

volume 3 number 2 a service newsletter may/june 1982

SEGA/Gremlin Dedicates New Facility

SEGA/Gremlin dedicated its new Rancho Bernardo Technology Park facility May 15, with a well-attended ribbon cutting ceremony and open house. Members of the press, local dignitaries along with employees and their families toured the 125,000 square foot complex that serves as the company's executive headquarters and main manufacturing plant.

"We are particularly delighted," commented State Senator William Craven in opening remarks, "that this highly-regarded international company is in San Diego, providing capital investment and employment. SEGA/Gremlin's commitment with this facility is a tribute to the productivity and dedication of the working people of California.

Also speaking at the ceremony were California State Assemblymen Robert Frazee and Larry Stirling.

The new headquarters building consolidates many operations into a single location and incorporates many industry-leading manufacturing systems. From receiving of component parts through final game assembly and shipment, the plant provides highly mechanized methods of quality assurance and mass production.

"California is ideally suited for us," stated David Rosen, Chairman of Sega Enterprises. "There is a combination of creative and entrepreneurial talents here in the electronics industry that can't be found anywhere else in the world."

After the morning's opening ceremony, company executives held a press conference followed by a plant tour and buffet lunch.

SEGA/Gremlin currently employs in excess of 650 people at their new San Diego headquarters plant. Added to the company's four other sites in San Diego, the new building brings SEGA/Gremlin's total office and production space to more than one-quarter million square feet.

In the afternoon, employees and their families were invited to tour the new facility and to challenge the SEGA/Gremlin assortment of hit games such as ZAXXON™, Turbo™, and Frogger™, and 005™. More than 1600 people attended the day's activities.

David Rosen, Chairman of SEGA Enterprises, cuts the ribbon officially opening SEGA/Gremlin's new manufacturing facility.

tech — tips

We continue now with the last in a two-part article describing the Color X-Y Monitor and its G-80 hardware drive.

Before we go any further, you should know that the words read from the video RAM didn't get there by magic. The microprocessor put them there. All the words that represent all the characters and designs in a game are kept safely in EPROM, on the EPROM Board. When the game is powered up the CPU moves the various word groups from EPROM to RAM. Once situated in video memory, the words are made available to the Timing and Control Boards to create displays. Briefly, here is what the two X-Y boards do with the words:

After all the character words are loaded into video RAM, the Program Counter is forced to call on the first RAM address. Here, the Program Counter finds the first word in the Symbol Instructions section. One at a time, these ten words are taken out of RAM and latched into specific parts of the X-Y boards. This sequence is controlled by the Timing Generator which generates 15 active-low signals; only one signal pulses low at a time to latch the words. Next, the following sequence occurs:

1. Word 1 is latched into U52, Last Symbol block.
2. Words 2 and 3 (from the Symbol Instructions) are loaded into the X-axis Up/Down Counters.
3. Words 4 and 5 are loaded into the Y-axis Up/Down Counters.
4. Words 6 and 7 are stored in the Vector Address counter.
5. Words 8 and 9 are stored in the Symbol Angle latch.
6. Word 10 is loaded into the Serial Multiplier.

Now the first 10 words of a character we wish to display are stored somewhere. If we had not wanted to draw this character, the Program Counter would have jumped to the next set of 10 symbol words, if there were another symbol. Once the Program Counter is finished moving out the words, the Vector Address counter takes control of the memory. This switching between the Program and Vector Address Counters is governed by the Multiplexer. Through it, the G-80 Address Bus or one or other counter can address video memory. Only one device is allowed access to memory at a time. When it is the Vector Address counter's turn to get into memory, the counter addresses the first word of the Line Intructions section. The reason for this, is that the Vector counter was previously loaded with words 6 and 7 in the Symbol Instructions section. These words tell the Vector Address counter the locations of the first word in the Line Instructions section. Now the Vector counter causes the Line Instruction words to be moved out of memory, one at a time. Here is what happens to the group of four words:

1. Word 1 is stored in the Color Latch and in U52, Last Vector.
2. Word 2 is stored in the Vector Length counters.
3. Word 3 is loaded into the Vector Angle circuit.
4. Word 4 is used in the Vector Angle circuit.

Up to this point, then, 10 words that describe some symbol and 4 words that represent *one line* in that symbol have been clocked out of memory. When the Vector Address counter takes out the last group of words that represents the last line to be drawn, memory access is switched back to the Program Counter. It will now either restart the sequence as before (if there is another symbol to be drawn) or it will stop until the counter is reset to the beginning again.

Now the system is ready to draw its first line, to actually move the beam, because we have given it exact specifications to do so. The position of the electron-beam is now known, (Words 2 through 5 in the Symbol Instructions) so, we know where the beam will start to draw. Then, the X-Y boards calculate the length and angle of our first line. And, they know the color, if any, of this line. (In our triangle, remember, our first line, A, is the one we don't actually see, but it must be drawn. We also know that it must be drawn at a 0 degree angle — straight up.) The boards calculate the line length and angle by assigning a certain number of digital clock pulses to these qualities: The circuitry acts as a digital "ruler" and "protractor" and measures any line, in terms of how many pulses in length and angle it is. All this figuring is performed by the Fuller Adder's, 2708 EPROM and Rate Multipliers. The output of each Rate Multiplier (XCL and YCL) is a string of clock pulses that clock the respective Up/Down counter, X or Y. But the pulses are not just random pulses; they are the digital equivalents of a line with a particular length and angle. So, we clock both Up/Down counters at the same time with these meaningful pulses. Doing so, we force the counters to start counting from the beam position words previously stored in the counters. The important point is this: if we change the digital values of the beam position words, we change the position of the beam through the D/A converters. How far and at what angle we change the beam depends on the amount of pulses applied to the Up/Down counters. Just before the beam is moved, the color of the line (black, included) is sent to the monitor through the RGB D/A converters.

There you have it — one line. For more lines, the G-80 system rapidly follows the same procedure of reading the symbol and line words, latching

continued from page 5

service notes

Service Schools

Technical service training is available from SEGA/Gremlin, on an appointment basis. Classes are conducted by our Customer Service Repair Technicians, and consist of a two-part seminar; a review of digital electronic theory and general troubleshooting logic, and a more in-depth study of specific current games in use. Those interested in organizing a class may contact their distributors for more information.

ZAXXON™ Cocktail Table

We have shipped two versions of ZAXXON as it relates to the Joystick/Flying Ship interface.

The player that has played the upright version and now plays the table version, would want the ship to ascend by pulling the joystick back, and descend by pushing the joystick forward.

Both versions have been shipped to our distributors. To modify this facet of the game is quite simple and can be accomplished by following these instructions:

1. Facing the open table from the Player One position, locate the harness connector for the Player One control panel located at the left corner of the monitor, nearest you.
2. Separate the two halves of the connector.
3. On the female half of this connector, locate pin numbers 3 and 4. Numbers are stamped on the back side.
4. Reverse the wire positions of pins 3 and 4, and re-make the connector halves.
5. Locate the Player Two control panel harness connector at the far left-side of the monitor.
6. Repeat Steps 2, 3, and 4. This completes the modification.

NOTE: Removal of connector pins may require the use of pin extractor. This extractor is available from AMP Inc., part no. 458994-1-0.

ZAXXON™ Upright

A change has been made to the ZAXXON Owner's Manual (P.N. 420-0724), page 41, Drawing No. 800-3241, Assy. Control Panel. In the interests of simplicity and economy, please change your Manual as follows:

DELETE:

Item No.	Part No.	Qty. Req'd.	Description
10	510-0023	4	WICO switch
11	253-0031	8	Spacer WICO switch

ADD:

Item No.	Part No.	Qty. Req'd.	Description
10	510-0064	4	Switch, Panel Pushbutton

TURBO™

On page 5, of the TURBO Owner's Manual (P/N 420-0681) we suggest that you add the following notes:

1. When the CPU Board DIP Switch #1 is set in the "SKILLED" position, game time is automatically set to 55 seconds. Option switch settings have no effect.
2. Regardless of "GAME TIME" option selection, the displayed countdown begins at 99 seconds, but does vary its *rate* of countdown according to option selection settings of DIP Switch #2 position 1 & 2.
3. DIP Switch #1 position 5 is an option used for game testing purposes. Set in the "OFF" position, the game will operate normally. Set in the "ON" position, however, your car will not crash. In other words, DIP Switch #1 position 5 is an "endless game" option.

TURBO™

There have been some errors identified in the TURBO Owner's Manual (P/N 420-0681). Please make the following corrections on the pages indicated:

1. Pg. 15, ITEM NO. 15, PART NO. *SHOULD READ* 834-0220
2. Pg. 31, ITEM NO. 11, PART NO. *SHOULD READ* 834-0229
3. Pg. 32, ITEM NO. 23, PART NO. *SHOULD READ* 834-0090
   ITEM NO. 24, PART NO. *SHOULD READ* 834-0091
4. Pg. 49, ITEM NO. 45, PART NO. *SHOULD READ* 834-0234
   ITEM NO 47, PART NO. *SHOULD READ* 834-0233
5. Pg. 58, ITEM NO. 17, PART NO. *SHOULD READ* 834-0142
   ITEM NO. 18, PART NO. *SHOULD READ* 834-0230
   ITEM NO. 19, PART NO. *SHOULD READ* 834-0231
   ITEM NO 20, PART NO. *SHOULD READ* 834-0232

Service

In the interests of improved "turnaround" time, as well as a more thorough troubleshooting procedure, we request that in the future, all monitor repair shipments to SEGA/Gremlin include the *entire* assembly, less possibly the CRT. Although it is understood that shipping costs will be higher, the practice of sending only the EHT Board (for example) proves to be a false-economy when the "real" problem is on the 'neck board', and another round of shipping and receiving is the result. Additionally, when we receive a complete monitor assembly, we can verify/align/repair the entire system, thus adding a facet of preventive maintenance to the exchange. We believe this policy to be in the best interest of all concerned, and would like to thank you for your cooperation.

the ROM line

FIRST OF A TWO PART ARTICLE

Random Illogic And Elegant Creeps

". . . I just figured out that a 'flip-flop' is not a stomach disorder, and now you expect me to fix a microprocessor game . . ."

Lane Hauck, Gremlin Industries

For the service technician, the electronic game art might be advancing a little too rapidly.

The last few years have seen a switch from mechanical steppers, relays and lamps to sophisticated logic systems with a whole new list of rules and symbols. With barely enough time to warm up his logic probe, the service technician is now faced with still another set of problems bearing such exotic names as ROM, PROM, RAM, CPU, Bus, Tri-State and Two/phase-clock/generator.

To fix it, of course, you have to understand it and the enterprising serviceman finds that there is an apparent wealth of information on the microprocessor. It is scarcely possible to pick up any electronic publication these days which does not contain something about the microprocessor. Unfortunately, a lot of this information consists of statements like, "The microprocessor is here to stay" and other generalizations which sound nice, but do nothing for the guy who has to get his game working.

This article is intended to remove some of the haze created by overzealous manufacturers and "general purpose" articles concerned with the microprocessor. Using detailed hardware examples, we'll explore the following important properties of the microprocessor.

1. The microprocessor is nothing more than a collection of logic gates. The only difference between the processor and random logic design is in the organization — the microprocessor is organized in a manner which allows changeability for different applications.
2. The microprocessor accepts inputs from the outside world, manipulating them according to a stored list of instructions.
3. The microprocessor can only perform one operation at a time. But it does things so rapidly that it appears to be doing things simultaneously.

RANDOM LOGIC DESIGN

Today's digital integrated circuits offer a rich variety of specific functions: counters, shift registers, decoders, flip-flops, to name a few. To implement a complex electronic game, the designer's task is to combine many of these specific functional blocks into a system. A look at any modern electronic game will show you the great variety of integrated circuits (IC) types required in the logic system — each performs a specific function of the game.

For example, if a scoring system is needed, a simple IC counter can be used to accumulate scoring pulses and a decoder is available to convert the counter outputs into a suitable display format. If it is necessary to store a piece of information temporarily (such as the fact that the game is in progress), a latch is available. And so goes the list — each requirement is satisfied by a logic block (integrated circuit) designed for a specific purpose.

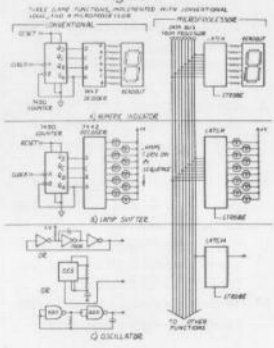
This might be called a distibuted intelligence approach since all parts perform a specific task without which the system could not operate.

ENTER THE MICROPROCESSOR

To illustrate how the microprocessor's approach to an electronic game differs from the above "conventional" approach, the three game circuits are shown in Figure One, implemented both ways (conventional and micro-processor). While these circuits are not associated with any game in particular, they provide elements that are common to most games; a scorekeeping circuit, an animation circuit (which turn indicators on in a sequence) and an oscillator.

Let's investigate the readout in detail. The display format is called "seven-segment" because of the seven bars which form the numbers. A counter accepts pulses from the game logic to advance the score once for each pulse. A reset line into the counter sets the score back to zero at the beginning of a game. Another logic element, a seven-segment

Figure 1



the ROM line

decoder, takes the counter output and translates it into the seven bar format necessary to display the numbers.

LATCH ONTO THIS

To understand how a microprocessor implements the same functions, we first need to understand a basic logic block called a latch. A latch has three elements: a set of input lines, a corresponding set of output lines and a control line which regulates the transfer of data from the input lines to the output lines.

The reason for the name, "latch," is that the control lines freeze the data on the output lines even though the input data can be changing. The control line is normally called a strobe in a microprocessor system.

While the two score implementations shown in Figure One produce the same result, they achieve it quite differently. The conventional logic circuit requires that the information reaching the latch is already in the seven segment format.

Consider an example where the readout changes from 6 to 7. The processor must do two things:

1. Place the proper combination of ON and OFF signals on the latch inputs.
2. Pulse the stobe line.

Suppose that there are two readouts in the game. The second readout is implemented the same way as the first even to the extent that the corresponding inputs of both latches are tied together. The only thing that distinguishes one latch from the other is that they are controlled by separate strobe lines. The group of eight lines feeding the latches is called a bus. To change readout A to 5 and readout B to 9, the control logic feeding the latches would do the following:

1. Put the (seven segment) code for 5 on the bus.
2. Pulse the A strobe line.
3. Put the code for 9 on the bus.
4. Pulse the B strobe line.

VERSATILITY PERSONIFIED

It might seem at first glance that the microprocessor approach is overly complex. After all, the random logic design required only a pulse on a single line to advance the readout count by one, while the microprocessor approach required eight input lines and a strobe line.

But now suppose you wish to have the readout do more than simply count by one. For example, you might have it do the following:

1. At the beginning of a game perform a 10, 9, 8 . . . 1 countdown.
2. Add any number to the score (not just one) based on game bonus situations.
3. In a self-check maintenance mode indicate failed components by number.
4. Flash rapidly for certain game situations.

To do any of these would be impossible with the conventional logic implementation since the circuits used are not designed to do anything more than count by one and decode. The processor, on the other hand, by having direct access to all output lines can make each latch function as required by the system.

Using the bus and strobe technique, all sytem outputs can be implemented in this manner. The guiding rule in this type of organization is that only one output latch can be strobed at a time. This insures that the lamp shifter in Figure one, for example, does not receive information which was intended for the score readouts.

In the next issue, we will continue this article which delves further inside the microprocessor to discuss its elements and outline its use in a typical game.

continued from page 2

tech — tips

them, calculating line values, and then forcing the beam line-by-line to form the complete symbols.

Notice that throughout our discussion, we have mentioned the microprocessor's role just once: it loads the character words into memory. During an actual game, however, it does more than act as a "loader". Whenever we want to move our symbols on the screen, for example, rotating the triangle, we need some way of changing all those words to represent new lines and angles. Well, the CPU is told when and how to alter these words to create a whole range of dynamic displays — different colored lines, new angles, longer or shorter lines, bigger or smaller characters, whatever the game play calls for. Also, the microprocessor governs all game functions such as player control and coin inputs, or sound and speech outputs. The CPU then, gives us the variety and color in the X-Y games.

Data Bus is a service newsletter published by SEGA/Gremlin Customer Service, 16250 Technology Drive, San Diego, California 92127.

Editor - Richard Cortez
Writer - Jim Bender
Photographer - Ron Stein
Layout - Carol Johnson

Parts Order - (800) 854-1900
Technical Assistance - (800) 854-1938
TLX - 910-335-1621

input port

Continuing our question and answer column, we would like to thank you for your interest and urge you to "keep those cards and letters coming in". Though urgent service questions should be directed to the toll-free technical assistance line, we do answer technical questions in this column, both written and those called in. If writing, please direct your questions to:

Editor — Data Bus
SEGA/Gremlin Customer Service
16250 Technology Drive
San Diego, California 92127-1985

Q: Although FROGGER has a very attractive music program, it's often not heard in my store above the other background noise. Is there any way for me to "up the volume" any more than it is? — K. Kanai, Aiea, HI.

A: Yes. The following modification will result in a substantial sound increase without risking the speaker:

1. Remove the existing 100 ohm potentiometer from the volume control block, and replace it with a 10K ohm pot (P/N 475-0007).
2. Remove the 1K ohm resistor from R-42 to ground, and replace it with a 4.7K ohm resistor in parallel with R-42 (P/N 475-0472).
3. Correct your Owner's Manual page 100, Drawing No. 834-0085, Zone B-1 to reflect the change.

Customer Service
16250 Technology Drive
San Diego, California 92127

BULK RATE
U.S. POSTAGE
PAID
PERMIT #248
CARLSBAD, CA

SEGA®/Gremlin®
Technology the Cause . . . Entertainment the Effect.