



**ELECTRONIC PINBALL GAMES
THEORY OF OPERATION**

F.O. 601-2

REVISED MAY, 1982

C.E.W.

BALLY ELECTRONIC PINBALL GAMES

THEORY OF OPERATION

The following discussion is intended for the serious reader as an aid to increase his understanding of the Electronic game. It is, essentially, the same material presented during the Bally Service Schools and is a direct response to the requests for a written record of this material. It is not intended to replace Repair Procedure F.O. 560, but rather to give greater insight into the functioning of the game. It is hoped that this insight will be of assistance in servicing the game.

There are several modes of operation designed into the Electronic Pinball games. These are as follows:

	PAGE
1) MPU Module Self-Test on Power-Up	2
2) Game Play	
A) Normal Mode	6
B) Interrupt Modes	7
3) Self-Diagnostic Tests	
A) MPU Module Resident	10
B) AID Module Controlled	10

It is recommended that the reader have a set of game schematics at hand for reference purposes.

C.E.W.

- 1. MPU MODULE SELF-TEST:** The MPU module has, as part of integrated circuit E 720-XX (U6), a program designed to test the module each time power is turned on. No action is required on the Operator's part to initiate the test. The program causes the MPU chip to test itself, the program containing integrated circuits U1 thru U6, the scratch pad memory U7, the non-volatile scratch pad memory U8, each of the input-output chips (peripheral interface adaptors: PIA) U10 and U11, and the display and zero crossing interrupt generator circuits. If the MPU finds all circuits in proper operating order, it initializes the game and makes it ready for play. If the MPU module finds a fault during the course of the Self-Test, it stops at that point in the test and does not allow game play.

The accuracy of the MPU Self-Test is about 99.5%. The MPU module catches all faults except buffer amplifier problems ('B' ports) in the PIA chips. These can easily be detected later by use of the AID1 module.

The interesting idea behind the MPU Self-Test is that it not only prevents game play when faults are detected, but also helps localize these faults. This is illustrated by means of page 3, borrowed from F.O. 560. The LED on the MPU module flashes once for each successfully completed test. Simply counting the number of flashes of the LED after power-up localizes the fault to the offending circuit on the module.

A) 1ST FLICKER:

On power-up, the MPU chip (U9), requires that $+5 \pm .25\text{VDC}$ be applied *before* the reset line is allowed to swing from 0 to $+4.8\text{VDC}$. It also requires the presence of a two-phase, non-overlapping clock pulse. If these conditions are met, and if the MPU chip itself is good, the LED on the module flickers briefly. F.O.560, page 29, deals with fault localization if the LED indicates a fault (LED always 'on' or 'off').

The brief flicker indicates the operation is proper. The MPU has gone out to memory. It has obtained the starting address of the Self-Test from memory. The flicker indicates that it then went to that address and started to execute the Self-Test program.

The Valid Power Detector circuit on the MPU module works with the $+5\text{VDC}$ regulator, Q20, on the Solenoid Driver/Voltage Regulator module to prevent the reset line from going high until $+5\text{VDC}$ is proper at the MPU chip. Q20 is guaranteed by the manufacturer to go into regulation when $+7.5\text{VDC}$ is applied to its input. This means that when the game is turned on and a sufficient period of time (milliseconds) has passed so that C23 (11,700mf) has charged to a $+7.5\text{VDC}$ level, Q20 switches into regulation and supplies $+5\text{VDC}$ to the MPU chip. However, Q1 in the Valid Power Detector circuit does not allow the MPU chip to turn 'on' until some time later. The Zener Diode (VR1), in series with the base of Q1 delays application of the reset voltage until C23 charges to an 8.9VDC level ($+8.2\text{V}$ across VR1, and $+ .7\text{V}$ base-emitter junction drop across Q1). At that point in time, Q1 and Q5 go into conduction, and the reset line at the MPU is caused to go high ($+4.8\text{VDC}$). Only then is the MPU chip 'on'.

The importance of the Valid Power Detector circuit can be appreciated when the following fact is known: Should the reset line be allowed to go high before the $+5\text{VDC}$ is applied and proper, or should the 5VDC supply fail and go out of regulation, the MPU can jump out of the program. The reason that this happens is that the MPU goes out to the program memory bank U1-U6 for instructions. The logic levels are wrong because the $+5\text{VDC}$ is not proper. The MPU misinterprets the data, jumps out of the program, and writes its own program! The MPU at that point in time is like a train that has left the tracks. It can end up anywhere. The difference is that the train eventually stops. The MPU may continue as long as the clock circuit continues to run.

If the MPU chip jumps out of the program, it is said to be in 'run-away'. While it is creating its own program by going out for instructions and executing anything that it gets from memory that looks like an instruction, it invariably overwrites the Bookkeeping functions in U8, the non-volatile scratch pad RAM. An indication of 'run-away', then, can be false data in the bookkeeping functions. Probable cause: faulty Q20, leaky C23 (high ripple) or leaky zener diode, (VR1, MPU module).

B) FIRST FLASH:

The MPU chip next goes out to program memory bank U1 thru U6 (Read Only Memory). It tests each chip, one at a time. When it finds the bank correct, it flashes the LED the first time to indicate its finding. A fault in the memory bank then is indicated by the absence of the first flash.

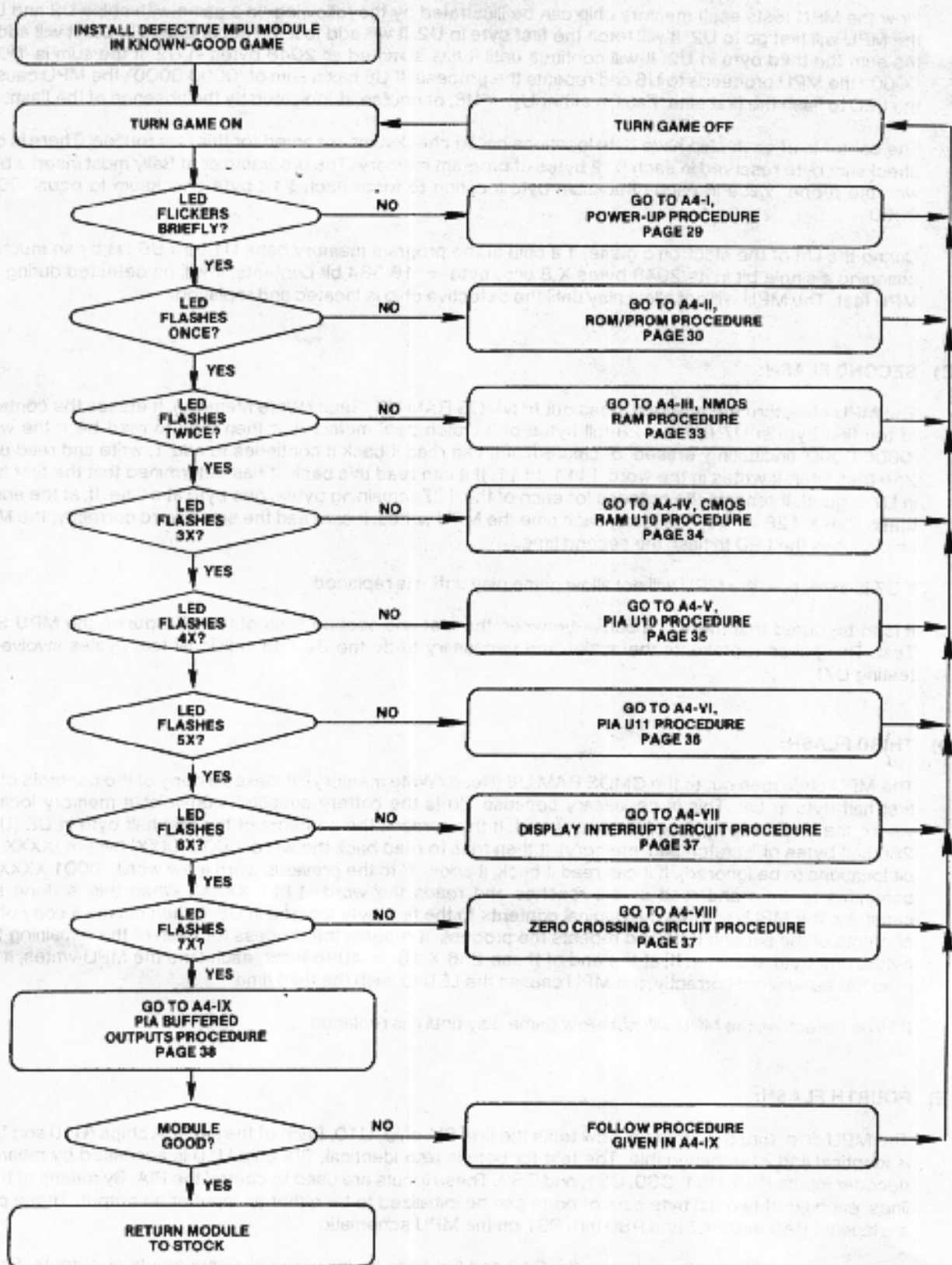


FIGURE A4-1 MPU MODULE TEST FLOW CHART

How the MPU tests each memory chip can be illustrated by the following: In a game with chips U2 and U6, the MPU will first go to U2. It will fetch the first byte in U2, it will add to it the second byte in U2. It will add to the sum the third byte in U2. It will continue until it has summed all 2048 bytes in U2. If the sum is '0000 0000', the MPU proceeds to U6 and repeats the process. If U6 has a sum of '0000 0000', the MPU causes the LED to flash the first time. Fault in either U2 or U6, of course, is indicated by the absence of the flash.

The contents of each chip have byte locations called checksums reserved for this test routine. There is one checksum byte reserved in each 512 bytes of program memory. The programmer at Bally must insert a byte with the proper value in each checksum byte location to force each 512 byte checksum to equal '0000 0000'.

During the life of the electronic game, if a chip in the program memory bank U1 thru U6 fail by so much as changing a single bit in its 2048 bytes $\times$ 8 bits/byte = 16,384 bit contents, it will be detected during the MPU test. The MPU will not allow play until the defective chip is located and replaced.

C) SECOND FLASH:

The MPU chip, thru the program, goes out to NMOS RAM U7 (Read/Write Memory). It erases the contents of the first byte in U7 (U7 is 128 full bytes of 'scratch pad' memory). It then tries to read back the word '0000 0000' (indicating 'erased' or cleared). If it can read it back it continues to add 1, write and read until, 256 tries later, it writes in the word 1111 1111. If it can read this back, it has determined that the first byte in U7 is good. It repeats the process for each of the 127 remaining bytes, one byte at a time. If, at the end of these $256 \times 128 = 32,768$ tests, each time the MPU writes, it can read the same word correctly, the MPU chip causes the LED to flash the second time.

If U7 is defective, the MPU will not allow game play until it is replaced.

It is to be noted that there is a pause between the first and second flash of the LED during the MPU Self-Test. This pause represents the actual time necessary to do the 32,768 individual test cycles involved in testing U7!

D) THIRD FLASH:

The MPU chip goes out to the CMOS RAM U8 (Read/Write memory). It makes a copy of the contents of the first half byte in U8. This is necessary because U8 is the battery supplied, non-volatile memory location where the bookkeeping functions are stored. It then erases the contents of the first half byte in U8 (U8 is 256 half bytes of 'scratch pad' memory). It then tries to read back the word '0000 XXXX' (where 'XXXX' are bit locations to be ignored). If it can read it back, it adds '1' to the previous word (new word, '0001 XXXX'). It continues to write and read until it reaches and reads the word '1111 XXXX'. When this is done successfully, the MPU restores the original contents to the first byte located in U8. It then makes a copy of the contents of the second byte and repeats the process. It repeats the process for each of the remaining 254 bytes, one byte at a time. If, at the end of these $256 \times 16 = 4096$ tests, each time the MPU writes, it can read the same word correctly, the MPU causes the LED to flash the third time.

If U8 is defective, the MPU will not allow game play until it is replaced.

E) FOURTH FLASH:

The MPU chip, thru the program, now tests the first PIA chip, U10. Each of the two PIA chips (U10 and U11) is identical and interchangeable. The test for both is also identical. PIA chip U10 is accessed by means of decoder inputs RS0, RS1, CS0, CS1, and CS2. These inputs are used to control the PIA. By means of these lines, each bit of two full byte sets of ports can be initialized to be either an input or an output. These ports are labeled PA0 thru PA7 and PB0 thru PB7 on the MPU schematic.

Also by means of these lines, two ports, CA2 and CB2 can be initialized as either inputs or outputs. Finally, two additional ports, CA1 and CB1, designed to be used as inputs only, can be initialized to trigger on a positive or negative going edge. These ports on PIA's U10 and U11 are used as interrupt inputs, to be discussed later.

To determine if each of the two PIA chips is good, the MPU chip, thru the program, does the following:

- 1) It accesses, by means of inputs RS0, RS1, CS0, CS1 and CS2 each of the two full byte registers used to store the port initialization information. It does this, one register at a time. After it completes the first register, it repeats for the second. It goes thru 256 tests similar to that used to check each byte in U7 (Second Flash). If, each time the MPU writes a word into the register, it can read the same word back, it continues the test to completion. If the MPU finds a fault in either of these two full byte registers, it stops the test and will not allow game play until the PIA chip is replaced.
- 2) It accesses, by means of inputs RS0, RS1, CS0, CS1 and CS2, each of two full byte registers used as data output registers when PA0 thru PA7, PB0 thru PB7 are used as outputs. It does the same type of test on each register described in 1) above. Again, if no faults are found, the test is continued to completion. A fault detected stops the test. Game play is not permitted until the PIA is replaced.
- 3) It then accesses, by means of inputs RS0, RS1, CS0, CS1 and CS2, first the CA2 and then the CB2 port. The port is initialized as an output. The port is then written into to see if it can store a '1' and then a '0'. When both ports are found good, the MPU causes the LED to flash the Fourth Time.

The CA1 port on U10 is tested manually each time the Self-Test switch on the inside of the front door is activated. The CB1 port is tested later as the Seventh Flash.

A total of $4 \times 256 + 4 = 1028$ test steps are required to test each of the two PIA chips. However, there are internal buffer amplifiers used with the PB0 thru PB7 output register and CB2 port register which cannot be tested by the MPU (access is only to the register — if the buffer is open, it does not interfere with the register's ability to be written into and read from by the MPU). It is this uncertainty that reduces the accuracy of the MPU Self-Test to 99.5%. Fortunately, open buffer amplifiers are easily detected and quickly located by means of the AID1 module.

F) FIFTH FLASH:

Identical to the test procedure and results detailed for the Fourth Flash except:

- 1) The CA1 port on U11 is tested later as the Sixth Flash.
- 2) The CB1 port on U11 is tested by the AID1 module. If the game cannot be made to go into the AID mode, the port is defective. (See discussion under AID1 Module Self-Diagnostic Tests)

G) SIXTH FLASH:

The MPU chip now monitors PIA2, port CA1 (U11). If transitions from high to low are detected, the MPU decides the Display Interrupt Generator is working. It causes the LED to flash the Sixth time. The MPU then enters the part of the program that deals with the initialization of the game prior to play.

If U12, a 555 timer, or any associated circuit component, fails, the MPU chip will not allow game play until the fault is corrected.

H) SEVENTH FLASH:

The MPU chip, thru the program, monitors PIA1, port CB1 (U10). If transitions from high to low are detected, the MPU decides that the zero crossing detector is working. It then causes the LED to flash the Seventh time.

If U14 fails and the CB1 line is stuck high or stuck low the MPU chip will not allow game play until the chip is replaced. It is to be noted that the zero crossing detector circuit input is the + 43VDC line to the solenoid common. If the fuse in that line (F4 on the Power Transformer module) is blown when the power is turned on, the MPU will not allow game play until the fault on the + 43VDC line is corrected.

1) GAME INITIALIZATION:

The MPU chip, thru the program, now initializes the two PIA's, U10 and U11, assigning to each port its role as either an input or an output, as required.

It then clears out U7 (Read/Write memory), which is like erasing the blackboard prior to school.

The MPU now 'takes a picture' of the settings of the fixed switches S1 thru S32 on the MPU board. It stores this 'picture' in memory (four bytes) in chip U7. With this record safely in memory, PIA U10 is now free to do other things during the course of play. The MPU need only go to memory to determine if the game is set for three ball or five ball, 1 game per quarter or three games per half dollar!

The MPU next jumps to a subroutine which turns on the 'Game Over' feature light, lights the 'Ball in Play' light and the 'Credit Indicator' light if there are credits stored in memory. It resets drop targets and activates saucer kickers or any kicker associated with a playfield device that can trap the ball and keep it out of the outhole. It then energizes the coin lock-out solenoid to allow the game to accept coins (unless credit maximum was stored in memory). Playfield and backbox feature lights associated with and appropriate to animation effects are now turned on.

With the game tested and initialized, the MPU now divides its time between monitoring momentary switches for closures (coin switch, credit button) and updating displays (lamps and score registers).

- 2. GAME PLAY:** After completing the MPU Self-Test, or between games, the MPU spends approximately 40% of its time monitoring the memory record of the momentary switches on the playfield and in the cabinet. The other time is divided between servicing the display update interrupts and the lamp, momentary switch scanning, and solenoid update interrupts.

A) NORMAL MODE:

The momentary switches are arranged in a 'matrix' (See Schematic W-1186, Sheets 2 & 3 of 3). The MPU chip, thru the program, examines a memory record of the matrix, looking for valid switch closures. If it finds a valid closure, decodes the address associated with the closure and jumps to the appropriate subroutine.

For example: If the game is in a game over status and the MPU finds that the #3 coin switch has a valid closure recorded in memory, it jumps to the coin/credit handling routine in the program. This routine reviews the memory record of the setting of switches S17, 18 and 19 on the MPU module to determine if the maximum credits have been reached. If they have, the coin lock-out solenoid has been deenergized and the coin is returned to the player. The MPU chip goes back to monitoring the record of switch closures. If maximum credits have not been reached, the memory record of switches S9 thru S13 is reviewed to determine how many credits are to be awarded per coin. These credits are added to the credit register in memory. The record of the number of coins thru coin chute #3 in the bookkeeping functions is increased by one. The credits are examined to determine if the maximum number of credits allowable by the memory record of switches S17, 18 and 19 on the MPU module have now been reached. If yes, the coin lock-out solenoid is deenergized.

The MPU chip, thru the program, now returns to its task of monitoring the memory record of valid switch closures, ready to jump to the appropriate subroutines that deal with the player pressing the credit button, etc.

The memory record of valid momentary switch closures is a qualified memory record. The MPU, as discussed under 'Interrupts', looks at each switch several times before it makes a decision as to whether or not a closure is valid. This multiple-look is a debounce mechanism that prevents the electronic game from giving away points on noise pulses or stuck switches. The debounce criteria is: When the MPU chip reviews the history of a switch to determine if a closure is valid, it must see an 'open' in the 'oldest' record. There must be a 'closed' in an 'old' record and a 'closed' in the current reading. Only when this criteria is satisfied will it make an entry in the memory record of valid closures that a switch is closed. If it saw a 'closed', 'closed', 'closed', the MPU would assume a stuck switch, and do nothing. 'Open', 'closed', 'open', or 'closed', 'open', 'open', are likewise rejected.

The momentary switches in the matrix are the 'eyes' and 'ears' of the MPU chip. It is only by means of sensing closures, and later, reacting to valid closures (during normal operation) that the MPU, thru the program, knows what it is to do next!

B) INTERRUPTS:

An interrupt is a signal to the MPU chip to stop what it is doing and do something else. When the MPU senses an interrupt from PIA-U10, CA1 or CB1 or from PIA-U11, CA1 or CB1, it completes the instruction it is working on, and makes a memory record of its contents and its place in the program so that it can get back to what it was doing before it jumps off to service the interrupt. When the interrupt task is completed, control is relinquished to normal operation. The MPU goes to the memory record of its pre-interrupt contents and methodically refills itself. It then goes about its business as if it had never been interrupted.

Interrupts are used for two types of activity in the Bally Electronic game. The first is the periodic lamp, solenoid, and momentary switch status update, U10, CB1 and the display update, U11, CB1. The second is to signal the MPU to go into the Self-Diagnostic tests, U10, CA1 or that the AID1 module has assumed control, U11, CA1. The second type is discussed under "3). Self-Diagnostic Tests".

The periodic interrupts are generated by the Zero Crossing Detector and the Display Interrupt Generator on the MPU module. The former occurs at a rate of 120 times per second, or once each power line zero crossing. The second occurs at a rate of approximately 320 times per second as determined by R21, R22 and C16.

- 1) *Zero Crossing Interrupts:* 120 times per second, or once each 8.3 milliseconds, the MPU chip senses a zero crossing, time-delayed by U14 just enough to allow a voltage to appear at the anodes of the silicon controlled rectifiers that drive the feature lamps before that portion of the interrupt routine begins.

Lamps are updated near the zero crossing to minimize the inrush current associated with a cold filament and hence extend their life. DC powered solenoids, likewise, exhibit a far smaller back EMF, if turned off near a zero crossing. This helps extend the life of the solenoid driver transistors and other circuit components by keeping large voltage spikes out of the system.

A period of time of about 3.7 milliseconds is required by the interrupt routine. The MPU looks at the contents of several general purpose timers at the start of this routine. If it finds them active, it subtracts 'one' from their remaining period. In passing thru, it adds 'one' to the random number generator used for the 'Match Feature' (unless the contents are already equal to 9, in which case, it resets the generator to zero). The random number generator, then, counts from 0 to 9, twelve times a second. This makes it virtually impossible to cheat, and truly random.

The MPU examines the status of the *solenoid period counter*. If it is zero, it turns off all momentary solenoids, and branches to the feature lamp update routine. If it is not zero, it subtracts 'one' from the contents of the counter. In general, momentary solenoids (thumper bumpers, slingshots, etc.) are energized for 3 zero crossings (26 milliseconds). Saucer kickers are energized longer to make sure that the ball clears the saucer.

The MPU next enters the *feature lamp update* part of the program*. There are fifteen, one half byte entries in U8, the CMOS scratch pad memory. This is equal to $15 \times 4 = 60$ bits, enough to form a memory 'picture' (lamp matrix) of the status (on or off) of each feature lamp in the game!

The MPU will now acquire the first half byte from memory, add to it an appropriate half byte address and send this address and data to the Lamp Driver module via PIA U10, ports PA0 thru PA7. The low order ports, PA0 thru PA3 carry the first decode address (0000) generated by the MPU chip thru the program. The high order ports, PA4 thru PA7 contain lamp data from the first half byte in memory.

The address (0000) goes to each of the four 'one of sixteen' decoder chips on the Lamp Driver module via the lines labeled AD0, AD1, AD2 and AD3. This is the address of the '0' port (pin 11) of each of these chips. The data is routed to the chips by the foil on the printed circuit board, i.e., PD0 goes to U1, PD1 goes to U2, PD2 goes to U3 and PD3 goes to U4.

It is to be noted that Pin 11 of chip U1 drives SCR Q14 (Bonus 1K), U2 drives SCR Q29 (Bonus 2K), U3 drives SCR Q36 (Bonus 3K) and U4 drives Q57 (Bonus 4K). Conclusion: the first half byte in the lamp matrix in memory chip U8 is a picture of the status of the Bonus 1K, 2K, 3K, and 4K lamps!

*Ref: Schematics W-1181, MPU, and W-1182, Lamp Driver Modules.

The MPU chip, thru the program, now causes the strobe line (CB2, PIA U10 thru J1 Pin 11 to J4 Pin 13, Lamp Driver Module), to go high and low, thereby presenting the first half byte of lamp update information to the gates of SCR's Q14, Q29, Q36 and Q57. A low (0) at the gate leaves the SCR and its associated lamp 'off', a high (1) turns it 'on'. When an SCR is turned on, it will stay on for the remainder of the supply line alternation (1/120 second) and turn off at the next zero crossing. It will then stay off, unless the next update again drives the gate high.

The MPU fetches the second half byte from memory, generates an address '0001' (pin 9 on U1, U2, U3 and U4) and repeats the process. It is now addressing the gates of SCR Q12 (Bonus 5K), SCR Q27 (Bonus 6K), SCR Q38 (Bonus 7K) and SCR Q50 (Bonus 8K). It causes the strobe to go high and then low, driving the gates of these SCR's and thereby updating the feature lamps Bonus 5 thru 8K.

The MPU fetches the third half byte, generates an address '0010', etc. It is now addressing the gates of SCR Q13 (Bonus 9K), SCR Q28 (Bonus 10K), SCR Q44 (Bonus 20K) and SCR Q51 (refer to Lamp Driver schematic for assignment). It repeats the strobe pulse and the Bonus 9, 10, 20, etc. lamps are updated.

Twelve more quick passes thru the subroutine and each of the 60 SCR's on the Lamp Driver module are updated. (Note that not all 60 SCR's are used in a given game.) The SCR's then can be thought of as a type of memory or storage. When the MPU updates the SCR, if it is turned on, it will stay on for the rest of the cycle.

The last step in the lamp module update program is to strobe an address '1111' into the chips U1, U2, U3 and U4. This is a rest address and frees the PA0 thru PA7 lines for other purposes.

The feature lamps each have one lead commoned to the +5.4VDC supply leg. The associated SCR, when turned 'on' completes the circuit to ground, thereby turning the lamp 'on'.

Lamp Strobe #2, U11, CA2, is provision for expansion. Should a game requiring more than 60 feature lamps be designed, the address and data lines will be commoned to a second lamp driver module and lamp strobe #2 will be used to select and control this module.

The last portion of the zero crossing interrupt routine is to read the *momentary switches* and look for valid closures. PIA U10, ports PBO thru PB7 are initialized as inputs, PA0 thru PA4 as outputs. The MPU chip, thru the program, sends a pulse down the ST4 strobe line (refer to MPU schematic W-1182 and wiring diagrams W-1186 sheets 2 and 3 of 3). If a switch is closed, the pulse will return down the corresponding 'I' line. The MPU chip examines the past history (in memory) of the switch and if it finds that the switch was 'open', 'closed' and is now 'closed', it makes a memory record of the valid closure. The reaction to this valid closure was discussed previously under 'NORMAL MODE of OPERATION'.

It is to be noted that stuck switches, a 'closed', 'closed' and currently 'closed' condition is ignored and does not result in a memory record of valid closure. Thus, the electronic game ignores stuck switches. Also, noise conditions such as 'open', 'closed' and currently 'open' do not satisfy the valid closure criteria, and are ignored.

The MPU chip sends a strobe pulse down the ST3 line and monitors the 'I' lines for returns. It repeats the process of evaluating the previous history of the switches from memory and makes a record of any valid closures. The process is repeated for the ST2, ST1 and ST0 lines. At the end of the time period, the entire switch matrix has been scanned and a memory record of the switches previous and current history is filed together with a record of valid closures.

It is to be noted that this multiple reading of a switch takes time, i.e., it must be done over several zero crossings before a valid closure can be verified and recorded. This procedure would spoil the response time to a hit on a thumper bumper or slingshot or any electromechanical device that must react quickly. To overcome this difficulty, a special, quick reaction subroutine exists in the program dealing with 'normal operation'. This routine takes place immediately after the memory record of valid closures is reviewed. It consists of a review of the previous and current history of *just* the solenoids that require a quick reaction. If an 'open', 'closed' record is found, the solenoid is energized. No scoring is involved in this routine. The net result is slingshots and thumper bumpers respond 'instantaneously'. They are not allowed to score until a valid closure is detected later. Because of this quick reaction subroutine, a noise pulse may cause a solenoid to pull (very-very infrequent). However, no points will be added to the players score. If the pull and scoring ever occur for no apparent reason, it is most probably because of improperly adjusted switch contacts.

The diodes in the switch matrix are steering diodes that prevent sneak paths and subsequent false decodes. On Freedom, for example, if diodes were not used: when the MPU sends the group strobe pulse down the ST0 line, if drop target B is down and the ball goes thru the spinner and strikes the GP10 rebound switch at the top of the playfield while the spinner is still rotating, the strobe pulse will be returned on the I6 line and decoded as a 'Tilt'!

All switches in the switch matrix are driven or scanned at very low current levels. The life expectancy of these switches is excellent. The material necessary to make reliable contact at low current levels is gold. The contacts are gold plated. Dirt and grease can interfere with good contact. A TV tuner contact cleaner-lubricant is excellent for cleaning contaminated switches. They may also be serviced dry by inserting a piece of clean, lint-free paper (a business card), for example, between the contacts, gently closing them and working the paper back and forth until the contacts are wiped clean.

Burnishing tools and contact files are not to be used. These will break the contacts' gold plated finish. Corrosion and unreliable operation will follow. This can only be remedied by replacing the switch contact assembly.

The Zero Crossing Interrupt is now completed. The MPU chip goes into memory and replenishes itself with its place in the program and the data it was processing prior to the interrupt. It then begins and continues in the program as if it never had been interrupted.

- 2) **DISPLAY UPDATE INTERRUPT:** 320 times per second the MPU senses a display update interrupt thru CA1, U11. The MPU again makes a memory record of its contents and then jumps off to service the interrupt.

There is also a memory record for each digit on each of the five displays in the CMOS memory, U8. $8 \times 5 = 30$, 1/2 byte memory locations are reserved for retaining this data. The data is stored in BCD (binary coded decimal). Each half byte is capable of storing 16 states, the numbers 0 thru 15 could thusly be represented and stored in a half byte of memory. BCD uses only ten of these states, i.e., the numbers 0 thru 9 only are stored.

The displays in the Bally Electronic Pinball games are multiplexed. This means that only one digit per display is on at a given point in time. If a picture of the backbox were taken with a high speed camera, the result might show that at the time the shutter opened, the #6 digit was 'on' on all five display driver modules.

The multiplex rate is fast enough that humans do not see the flicker resulting from the fact that each digit is actually on slightly less than 17% of the time! Multiplexing minimizes the number of leads necessary to control the displays, just as arranging the momentary switches in a matrix cut the required leads from $8 \times 5 + 1$ (return) = 41 to $8 + 5 = 13$ leads. The true advantage is the number of PIA chips required by the system are minimized.

The BCD lines for all five displays are commoned. The same is true for each of the six digit enable lines and the blanking line. Only a separate display latch strobe line is required for each of the five display driver modules.

The MPU chip begins the update by determining which digit was updated last. Assume that this was digit #4. The MPU, thru the program, adds one to this number and makes a memory record of this fact for future reference. It causes the blanking line to go high and blank the displays. This keeps each digit clean and crisp looking by preventing flicker during the update.

The MPU goes into memory and obtains the BCD information for the fifth digit, 1st player up display. It places this information on the BCD lines and strobes it into U1 on the 1st player up display driver module.

The MPU goes back to memory, obtains the BCD information for the fifth digit, 2nd player up display. It places this information on the BCD lines and strobes it into U1 on the 2nd player up display driver module.

The process is repeated for the 3rd, 4th player up display drivers and then for the match/ball in play, credit display. With this complete, it enables the #5 digit, disables all other digit lines, removes the blanking pulse and returns from the interrupt to whatever it was doing previously. And all this in a period of 500 microseconds!

Assume that the MPU chip is to enable digit #5. The base of level shifter transistor Q5 is made high on each of the five display driver modules. This causes the collector of the Q5 to drop from a high, positive voltage (+ 190 volts DC less the leakage current drop across R9, 100k ohms) to the saturation voltage, approximately .3VDC. The voltage across R9 is now approximately 189 VDC (190 - .3 (VCE SAT, Q5) - .7VDC (VBE, Q11)). The collector of Q11, which was clamped to the + 80 VDC bus, now rises to approximately 189.7 VDC (190 - .3(VCE SAT, Q11)). Digit #5 is now enabled.

Assume that the fifth digit of the 1st player up display driver module is to display the number '3'. The MPU chip loads the BCD lines with '0011' (D3, D2, D1, D0). It then strobes this information into chip U1 on the display driver module, where it is stored for the period of the update. U1 decodes the input, and as soon as the MPU chip removes the blanking pulse, the bases of transistors Q13, Q14, Q15, Q16 and Q19 are made high by the outputs of U1. The emitter-collector voltage of these transistors, previously at + 80VDC due to the blanking pulse, now falls to VCE SAT, approximately + .3VDC. The result is the 'a', 'b', 'c', 'd' and 'g' segments in the display panel are enabled.

Both of the actions in the above example result in turning on the number '3' in the 5th digit position on the 1st player up display.

It is interesting to note that the 6800 MPU is capable of being interrupted while it is servicing an interrupt. All that is necessary when this happens is for the MPU chip to make a record of where it was in the program and of its contents. It can then jump off to service the interrupt. At the completion of this task, it returns, completes and finally returns to normal operation. An example of this action is a zero crossing interrupt being interrupted by the Display Interrupt Generator.

3) SELF DIAGNOSTIC TESTS

- a) The Bally Electronic Pinball games have resident (built-in) module self-diagnostic tests. These test routines can be entered at any time, provided that the MPU module is capable of passing its own Self-Test. The module test is illustrated on page 10, borrowed from F.O.560. The test is initiated by pressing the Self-Test switch button located inside the cabinet door. The switch closure is detected by the MPU chip thru CA1, U10, as an interrupt. Each test continues until the next press of the Self-Test button. The operator or service technician can 'walk thru' these tests as slowly or rapidly as is desired. A minimum period of about one second is required by the program between presses of the Self-Test button, however.

The same Self-Test button is used to enter the threshold review/reset entries in memory. These are entered immediately after the stuck switch and are the next four presses of the button. The bookkeeping features are next. It is to be noted that thresholds or bookkeeping entries can be set to zero by means of S33 on the MPU module. Thresholds are then changed to the desired level by holding in the credit button and releasing it at the appropriate time. Bookkeeping entries, however, cannot be reestablished.

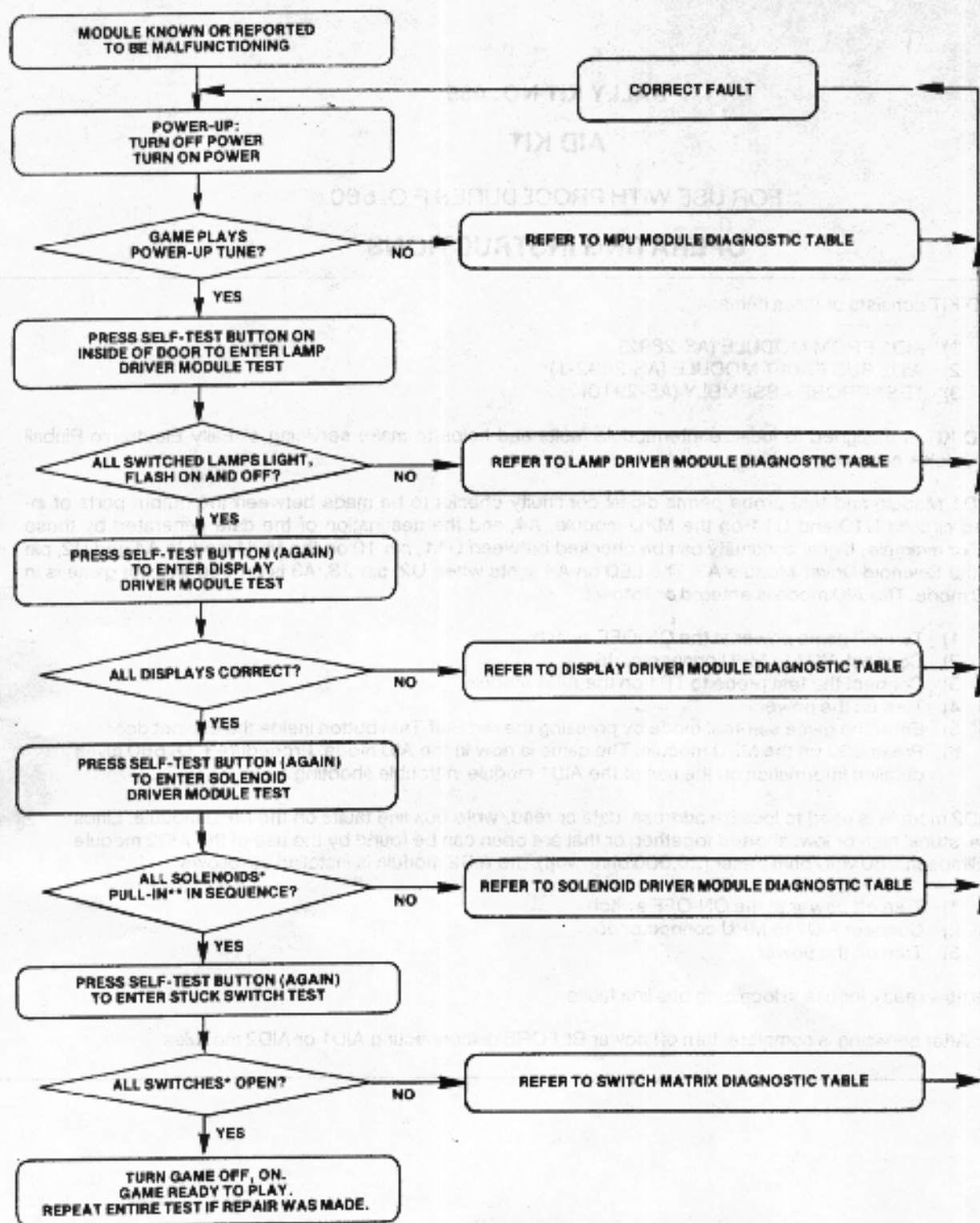
The Self-Test threshold/bookkeeping interrupt routine ends with the sixteenth press of the button or can be cut short by turning the game 'off' and then 'on'. In either case, the MPU does not return to its previous place in the program, but rather jumps to the power-up routine and reinitializes the game.

- b) The AID program is a special subset of the Self Diagnostic Interrupt. If the AID1 module is inserted on MPU module connector J5, and the game is turned 'on', the AID mode can be entered from any of the Self Diagnostic Tests, i.e., the first four presses of the Self-Test button. Entry is by means of S33 on the MPU module. Exit is by means of the game 'on-off' switch.

AID1 module includes a program chip. This program reinitializes both PIA's, turning all ports into outputs. Each port in U10 and U11 is pulsed in sequence. Probing any of these lines examines the line for defective PIA buffer amplifiers, shorts and opens.

The AID1 module has built-in noise immunity. It looks for ten consecutive 'hits' before it makes a decision that a pulse is present. It therefore ignores random noise pulses.

The AID1 module can be used to detect adjacent line shorts. If two adjacent lines are shorted (ex: U10 PA0 and PA1), the AID1 module will detect a return pulse twice as long as normal. It will not allow the LED on the MPU module to light when either line is probed.



*SOLENOID AND STUCK-SWITCH NUMBER IS FLASHED ON CREDIT REGISTER TO AID IN TROUBLESHOOTING.

**HOLD FLIPPER BUTTONS IN DURING TEST.

FIGURE III SELF DIAGNOSTIC TEST

BALLY KIT NO. 485

AID KIT

FOR USE WITH PROCEDURES F.O. 560

OPERATING INSTRUCTIONS

The AID KIT consists of three items:

- 1) AID1 PROM MODULE (AS-2892)
- 2) AID2 BUS FAULT MODULE (AS-2892-1)
- 3) TEST PROBE ASSEMBLY (AS-2910)

The AID KIT is designed to localize intermodule faults and helps to make servicing of Bally Electronic Pinball games quick & efficient.

The AID1 Module and test probe permit digital continuity checks to be made between the output ports of integrated circuits U10 and U11 on the MPU module, A4, and the destination of the data generated by those ports. For example, digital continuity can be checked between U11, pin 10 on the MPU module A4 and U2, pin 23 on the Solenoid Driver Module A3. The LED on A4 lights when U2, pin 23, A3 is probed and the game is in the AID mode. The AID mode is entered as follows:

- 1) Turn off game power at the ON-OFF switch.
- 2) Connect AID1 to MPU connector J5.
- 3) Connect the test probe to TP1 on the AID1 module.
- 4) Turn on the power.
- 5) Enter the game self-test mode by pressing the red Self-Test button inside the cabinet door.
- 6) Press S33 on the MPU module. The game is now in the AID mode. Procedure F.O. 560 gives detailed information on the use of the AID1 module in trouble shooting.

The AID2 module is used to localize address, data or read/write bus line faults on the MPU module. Lines that are 'stuck' high or low, shorted together, or that are open can be found by the use of the AID2 module and a Simpson 260 volt/ohm meter (20,000 ohm/volt). The AID2 module is installed as follows:

- 1) Turn off power at the ON-OFF switch.
- 2) Connect AID2 to MPU connector J5.
- 3) Turn on the power.

AID2 is now ready for use in localizing bus line faults.

NOTE: After servicing is complete, turn off power BEFORE disconnecting AID1 or AID2 modules.

The AID1 module has built-in high voltage protection. Touching the 115VAC or 234VAC lines in the game cannot damage the module.

The AID2 module is used to locate faults on address and data lines. It makes use of the fact that if the HLT (HALT) line on the 6800 MPU chip is grounded, the MPU relinquishes control of the address and data lines on the MPU module. The AID2 module can then be used to test these lines on a static basis.

Detailed procedures for using the AID1 and AID2 modules are given in F.O.560.

4) MISCELLANEOUS:

a) SOLENOID DRIVER MODULE

- 1) The MPU chip energizes continuous duty solenoids (coin lock-out, flipper enable relay coil) by means of dedicated lines PB4 thru PB7, PIA U11. For example, to energize the coin lock-out coil, the MPU chip, thru the program, causes PB5, U11, to go low. This pulls down the base of the transistor connected to pin 13 in chip U4 on the solenoid driver module. The collector, pin 14, goes high and Q19 is turned on, thereby completing the coin lock-out solenoid return to ground. When the MPU chip allows PB5, U11 to go high, the process is reversed and the coin lock-out is deenergized.

It is to be noted that each solenoid in the game has one lead connected to the +43VDC common. The other lead is controlled by a transistor (Q1 thru Q19). The transistor when turned on, completes the circuit to ground and turns the solenoid 'on'. The diodes across the solenoids short any voltage induced by the collapsing electromagnetic field on turn-off, preventing transient voltage spikes from damaging the game.

- 2) Momentary solenoids are controlled by means of lines PB0 thru PB3, U11. The MPU chip, thru the program, must generate the address of the solenoid to be energized, and present it to U2 on the solenoid driver module. Note that four lines control the selection of any one of fifteen solenoids by means of this decoder chip (U2).

The schematic shows that if all four decoder input lines (PB0 thru PB3) are low, the output of U2, pin 1 will go low and the solenoid driven by transistor Q2 will be energized. An address '1111' causes the decoder U2 to make pin 17 low. This is the rest address — all fifteen solenoids are 'off' for this condition.

- 3) All solenoids are controlled by identical driver circuits. Fifteen of these circuits work with one decoder chip U2 — the other four are driven directly from PIA U11.
- 4) The 190VDC regulator is a simple feed back amplifier/series regulator arrangement. The potentiometer RT1 is adjusted initially so that the voltage at TP2 is 190VDC. At that point, the base-emitter voltage of Q23 is about +.7VDC, the collector is at 193VDC. The base-emitter voltage across Q22 is about +1.5VDC, across Q21, an additional +1.5VDC. If the input voltage is +230 at this time, the collector emitter-voltage drop across the series regulator Q21 is 40VDC, across the control amplifier Q22, +37VDC.

If the input voltage were to increase to +240VDC, Q23 would conduct more, the voltage at the collector of Q23 would decrease slightly, +50VDC would be dropped across the series regulator Q21 and the net result would be that the output voltage at TP2 is still approximately +190VDC.

If the input voltage were to decrease to +220VDC, Q23 would conduct less, the voltage at the collector of Q23 would increase slightly, +30VDC would be dropped across the series regulator Q21 and the net result would be that the output voltage at TP2 is still approximately +190VDC.

b) NON-VOLATILE MEMORY

- 1) U8 on the MPU module is used to store data overnight when the power to the game is disconnected. The batteries BT1, 2 & 3 at the upper right hand corner of the schematic are charged thru CR5 anytime the game is on. These batteries are connected to U8 only. They supply the very small amount of power required by U8 whenever the game is off. The batteries, when fully charged, can maintain the bookkeeping and score threshold information stored in U8 for many months. U8 is a CMOS RAM. It draws about .7 microamps in the stand-by mode. This results in a power consumption, typically of only 2.5 microwatts!

If the batteries were to fail, or if the charging diode, CR5 were leaky, memory retention would be affected. Also, if the MPU were to jump out of the program, the result could be loss of the data stored in U8.

c) NOISE:

Before you buy, a good test for any electronic game is to shuffle your feet as you cross a nylon carpet on a dry winter's day. Hold a key in your hand to eliminate the pain, and discharge the static electricity you picked up into various, player-accessible metal parts of the game. If the game jumps out of the program or is actually damaged on the distributor's floor, the same thing (or worse) will happen on location. Put your money elsewhere!

Bally went thru an extensive period of design to achieve noise immunity. Every player-accessible, external metal part (legs, door, cabinet trim, etc.) is grounded thru the third lead on the power plug. The back box has a metal liner, also grounded, to keep high frequency pulses out of the electronic circuitry. Every line into or out of the MPU module is decoupled by means of a resistor or capacitor. Finally, the line cord ends at an EMI (electromagnetic interference) filter designed to keep noise pulses on the power line out of the game.

The net result is Bally Electronic games work on locations where other manufacturers' electronic games become problems. The third lead in the power plug must be connected to conduit ground in the location before these results can be achieved, however. Owner-operators must be educated that this is a necessary condition for reliable operation of any electronic game.

REFERENCE: M6800 MICROCOMPUTER SYSTEM DESIGN DATA
MOTOROLA SEMICONDUCTOR PRODUCTS, INC.
BOX 20912
PHOENIX, ARIZONA 85036

AI: DISPLAY DRIVER MODULE

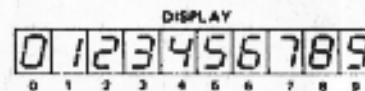
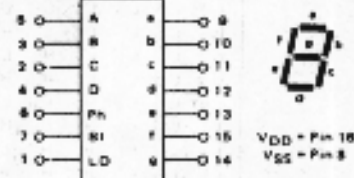
UI: LATCH/DECODER DRIVER

14543B

BCD-TO-SEVEN SEGMENT LATCH/DECODER/DRIVER for LIQUID CRYSTALS

The 14543B BCD-to-seven segment latch/decoder/driver is designed for use with liquid crystal readouts, and is constructed with complementary MOS (CMOS) enhancement mode devices. The circuit provides the functions of a 4-bit storage latch and an 8421 BCD-to-seven segment decoder and driver. The device has the capability to invert the logic levels of the output combination. The phase (Ph), blanking (Bl), and latch disable (LD) inputs are used to reverse the truth table phase, blank the display, and store a BCD code, respectively.

- Logic Circuit Quiescent Current = 5.0nA/package typical @ 5 Vdc
- Latch Storage of Code
- Blanking Input
- Readout Blanking on All Illegal Input Combinations
- Direct LED (Common Anode or Cathode) Driving Capability
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-power TTL Loads, One Low-power Schottky TTL Load or Two HTL Loads Over the Rated Temperature Range



TRUTH TABLE

INPUTS				OUTPUTS						
LD	Bl	Ph	DCBA	a	b	c	d	e	f	Display
X	1	0	X X X X	0	0	0	0	0	0	Blank
1	0	0	0 0 0 0	1	1	1	1	1	1	0
1	0	0	0 0 0 1	0	1	1	0	0	0	1
1	0	0	0 0 1 0	1	1	0	1	1	0	2
1	0	0	0 0 1 1	1	1	1	0	0	1	3
1	0	0	0 1 0 0	0	1	1	0	0	1	4
1	0	0	0 1 0 1	1	0	1	1	0	1	5
1	0	0	0 1 1 0	1	0	1	1	1	1	6
1	0	0	0 1 1 1	1	1	1	0	0	0	7
1	0	0	1 0 0 0	1	1	0	0	1	1	8
1	0	0	1 0 0 1	1	1	1	0	1	1	9
1	0	0	1 0 1 0	0	0	0	0	0	0	Blank
1	0	0	1 0 1 1	0	0	0	0	0	0	Blank
1	0	0	1 1 0 0	0	0	0	0	0	0	Blank
1	0	0	1 1 0 1	0	0	0	0	0	0	Blank
1	0	0	1 1 1 0	0	0	0	0	0	0	Blank
1	0	0	1 1 1 1	0	0	0	0	0	0	Blank
0	0	0	X X X X	**	**	**	**	**	**	**
1	1	1	1	Inverted Output Combinations Allowed					Display as above	

X = Don't care
 1 = Above Combinations
 * = For liquid crystal readouts, apply a square wave to Ph.
 For common cathode LED readouts, select Ph = 0.
 For common anode LED readouts, select Ph = 1.
 ** = Depends upon the BCD code previously applied when LD = 1.

A3: SOLENOID DRIVER MODULE

U2: 4-LINE-TO-16-LINE DECODER

Each of these monolithic, 4-line-to-16-line decoders utilizes TTL circuitry to decode four binary-coded inputs into one of sixteen mutually exclusive outputs when both the strobe inputs, G1 and G2, are low.

74L154

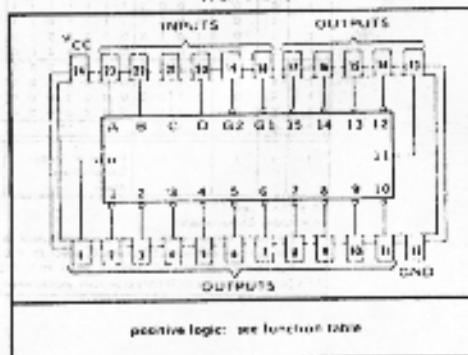
logic

FUNCTION TABLE

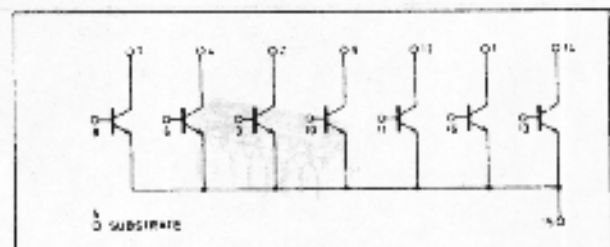
INPUTS					OUTPUTS																
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	L	H	L	H	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H															

H = high level, L = low level, X = irrelevant

(TOP VIEW)



3081



3081 pin connections

A5: LAMP DRIVER MODULE

U1-U4: DECODER SCR DRIVER

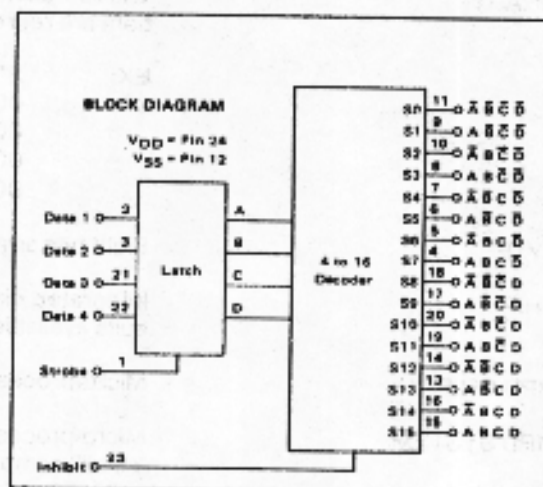
14514B

4-BIT LATCH/4-TO-16 LINE DECODER

The 14514B and 14515B are two output options of a 4 to 16 line decoder with latched inputs. The 14514B (output active high option) presents a logical "1" at the selected output, whereas the 14515B (output active low option) presents a logical "0" at the selected output. The latches are R-S type flip-flops which hold the last input data presented prior to the strobe transition from "1" to "0". These high and low options of a 4-bit latch/4 to 16 line decoder are constructed with N-channel and P-channel enhancement mode devices in a single monolithic structure. The latches are R-S type flip flops and data is admitted upon a signal incident at the strobe input, decoded, and presented at the output.

These complementary circuits find primary use in decoding applications where low power dissipation and/or high noise immunity is desired.

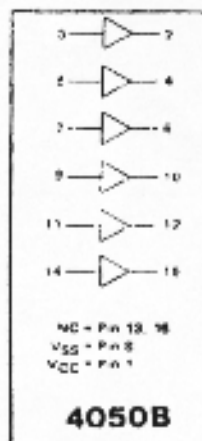
- Quiescent Current = 5.0mA package typical @ 5 Vdc
- Noise Immunity = 45% of VDD typical
- Supply Voltage Range = 3.0 Vdc to 18Vdc
- Capable of Driving Two Low-power TTL Loads, One Low-power Schottky TTL Load or Two HTL Loads Over the Rated Temperature Range
- Single Supply Operation — Positive or Negative
- Input Impedance = 10¹¹ ohms typical



DECODE TRUTH TABLE (Strobe = 1)

INHIBIT	DATA INPUTS				SELECTED OUTPUT 14514 = Logic "1"
	D	C	B	A	
0	0	0	0	0	S0
0	0	0	0	1	S1
0	0	0	1	0	S2
0	0	0	1	1	S3
0	0	1	0	0	S4
0	0	1	0	1	S5
0	0	1	1	0	S6
0	0	1	1	1	S7
0	1	0	0	0	S8
0	1	0	0	1	S9
0	1	0	1	0	S10
0	1	0	1	1	S11
0	1	1	0	0	S12
0	1	1	0	1	S13
0	1	1	1	0	S14
0	1	1	1	1	S15
1	X	X	X	X	All Outputs = 0, 14514

X = Don't Care



GLOSSARY OF TERMS

'1's and '0's

For the 6800 MPU chip, a '1' is defined as +2.4VDC or greater, a '0' is 0.4VDC or less.

BIT

A BIT is one data element. This bit may be either a '1' or a '0'.

BINARY

The representation of a system of numbers having 2 as its base. The numbers are represented as '1's or '0's.

EX:	2^7	2^4	2^3	2^0	
	0000		0000		= 0
	0000		0001		= 1
	0000		0010		= 2
	0000		0011		= 3

BYTE

Eight bits arranged as in example shown in BINARY above.

CHIP

Integrated circuit. An MPU chip is among the most complex integrated circuits available.

MPU CHIP

Micro-processor unit — 'small' computer on a chip.

MPU SYSTEM

Micro-processor or unit, memory banks, input/output circuits — complete 'small' computer.

SUPPLEMENT TO

THEORY OF OPERATION, BALLY ELECTRONIC PINBALL GAMES

GENERAL

The sound module accepts address signals from the MPU to select one of the notes stored in its tone memory (U3). Depending on its operating mode (A or B), determined by whether "A" or "B" jumpers are installed on the sound module PC board, the module will either decode the address information and generate its own trigger, or it accepts a trigger signal from the MPU via the solenoid bank select line. The trigger starts an exponentially decaying envelope which results in sound output at a frequency determined by the addressed note. If the rate of addressing/triggering is low, the sound has a chime like quality. Increasing this rate produces special effect sounds (noises). Address and triggering data information and the rate of data transmission are under MPU program control.

Operating mode B is the standard configuration and fully utilizes the sound module capability to produce chime and special effect sounds. Operating mode A is intended as a replacement of the electro-mechanical four chime unit with four equivalent electronic chime sounds where chime volume control is desired. No special effect sounds are feasible in the A operating mode, however, when so configured the sound module can be interfaced directly with MPU modules programmed for electro-mechanical chime games.

OPERATING MODE B

Operating mode B is selected when all four "B" jumpers on sound module P.C. board are installed and none of the "A" jumpers are in place. In addition, sound module tone memory U3, Bally P/N E-729-18 must be used.

In this configuration the sound module may be viewed as a bank of 30 chimes, tuned to cover a chromatic scale of two and half octaves from C through F2. Selection of a particular chime depends on the address data present at the tone memory (U3) address inputs A, B, C, D and E. Striking of the selected chime occurs when solenoid bank select signal makes a low to high transition. In addition to the 30 chime notes there are two nulls, corresponding to address locations 15 and 31. When struck, the null chime frequency is outside the audio range and thus no sound is produced. Details of the signal flow are discussed in the following paragraphs.

Address data A, B, C, D received from the MPU are shared with the momentary solenoid address bus. However, since auxiliary solenoid bank is selected (solenoid bank select signal high) immediately preceding and during sound address transmission, the momentary solenoids are not addressed at this time. Conversely, when primary solenoid bank is selected (bank select signal low) to operate momentary solenoids, the sound module is not addressed.

The solenoid bank select signal is capacitively coupled to the tone trigger generator one-shot U7, which is started when the signal makes a low to high transition. When triggered, U7 output goes positive for about 5 milliseconds. This pulse is applied to the clock input of the address data latch U2. With its clock input positive, the latch becomes transparent. The address data are inverted at the latch outputs and applied to the four buffer-inverters U1A, U1D, U1E and U1F which drive the tone memory U3 address inputs A, B, C and D. Due to the double inversion, the address data at the tone memory are the same as the address data input to the latch at this time. When at the end of the one-shot pulse the clock input goes low, the address data are latched by U2 and held frozen until the cycle repeats when the next trigger action takes place.

Address signal E is dedicated to the sound module and is held at the required level by the MPU.

The address inputs A, B, C, D and E applied to the tone memory U3 (32X8 PROM) select one of 32 possible 8-bit combinations at its data outputs D01 through D08. The eight bits represent the binary code for a divisor N and are applied to the programmable frequency divider consisting of U4 and U5. D08 is the *least* significant bit, D01 the *most* significant.

A square wave signal at a typical frequency $f_0 = 500$ KHZ is produced by the clock frequency generator U1B and U1C, and applied to the frequency divider. The output of the frequency divider is a series of 1 microsecond pulses at a frequency f_1 , determined by the divisor N, such that $f_1 = f_0 \div N$. The frequency divider output is further divided by 2 and by 8 in the 4-bit binary counter U11 to produce two square wave signals at U11, pins 9 and 11. The square wave at pin 9 has a frequency $f_2 = f_1 \div 2 = f_0 \div 2N$, while the square wave at pin 11 has a frequency $f_3 = f_1 \div 8 = f_0 \div 8N$. The two square wave signals are shaped and combined in the low pass filter, formed by R2, R3, R4, C3 and C4. The filter output is a simulated chime waveform at a fundamental frequency f_3 , modulated by its fourth harmonic f_2 .

The divisors (N) for successive address locations (except addresses 15 and 31) are so chosen, that chime frequencies are chromatically ascending. Thus if the lowest chime note at address 0 is C, the next note at address 1 is C sharp, the note at address 2 is D, etc. For addresses 15 and 31 the divisor $N = 1$, therefore the fundamental chime frequency $f_3 = 500\text{KHZ} \div (8 \times 1) = 62.5\text{KHZ}$. This is beyond the range of human hearing, and furthermore the signal is completely attenuated by the low pass filter. These null addresses are used when an abrupt cut-off of sound is desired, as in some of the special effect noise sequences.

The low pass filter signal is applied to U8A, configured as a voltage controlled attenuator. The output of U8A is amplitude modulated by the envelope shaping control voltage applied to pin 3. The control voltage has a sharp attack and an exponential decay and is started by the tone trigger generator U7, which in turn is triggered by the low to high transition of the solenoid bank select, as discussed above. When installed, sustain control RT2 works like a DC pedestal on the envelope shaping control voltage waveform. Turning RT2 clockwise raises the DC level of the control voltage baseline and allows longer decay times before U8A is shut off. If RT2 is turned too far clockwise, the baseline DC level becomes high enough to turn on U8A continuously. When this happens, a sustained note is heard, unless a null address has been latched. Care therefore should be exercised to assure that RT2 is properly adjusted for the desired chime decay.

Output of the voltage controlled attenuator U8A is preamplified by U9 and applied to the audio amplifier U10, whose output drives the loudspeaker.

OPERATING MODE A

Operating mode A is selected when all four "A" jumpers on the sound module PC board are installed and the "B" jumpers are removed. Sound module tone memory U3, Bally P/N E-725-22 must be used.

Operating in mode A is similar to mode B, except for the following differences.

Solenoid bank select and address E inputs from the MPU are not used. Tone memory U3 address input E is grounded and thus permanently low. The tone memory differs from that used in mode B in that it has all null divisors ($N = 1$), except at address locations 1, 2, 3 and 4. The divisors at these four addresses are such that chime frequencies corresponding to the four mechanical chimes are produced. The tone trigger generator is started each time solenoid address D makes a high to low transition. This occurs whenever one of the first 8 momentary solenoids is addressed by the MPU. However, since tone memory contains null divisors in all locations except the four occupied by the chimes, sound is heard only when the MPU addresses a chime solenoid.