6803 Control Board Theory of Operation

POWER SUPPLIES

The Control Board requires a regulated +5VDC for its logic circuit operation. In addition, the Power Module provides an unregulated +14 to+18VDC, the phase A and phase B voltages (11VAC),and +43VDC to the Control board.The +14 to+18VDC is used in the Valid power detector circuits; the phase A & B voltages are used for the Zero Crossing detector circuits and the +43VDC is required on the board exclusively for the flipper enable relay. Besides supplying these voltages to the Control Board and other sections of the game, the Power module also provides a regulated +190VDC to the displays and 6.3VAC for general illumination.

RESET CIRCUIT

On power up the µP (microprocessor) requires +5VDC±.25VDC be applied for 100 milliseconds before the RESET line is allowed to swing from 0 to +4.6VDC. The RESET circuit on the Control Board works with the unregulated+14 to+18VDC to prevent the RESET line from going high until the +5VDC supply has had time to stabilize after power on. The zener diode D1 and transistors Q2 and Q3 with R2 through R9 form a Valid power detector circuit that monitors the input voltage to the regulator (coming from the Power Module). This regulator requires a minimum of +7.5VDC input before it provides a +5VDC output. When this condition has been met diode D2 allows C1 to charge through R11.This RC time constant provides the initial 100 millisecond delay to allow the µP oscillator to stabilize. The voltage across C1 is monitored by Q4, Q5, D4, D5 and R12 through R16. When it has reached about +2.5VDC the RESET line snaps high to allow the µP to start program execution. In the event the output of D1 drops below +7.5VDC for an instant, the Valid power detector quickly discharges C1 through R9 and D2 to re-prime the RC time constant and insure a correct RESET cycle when power is re-applied.
The RESET signal is applied to the µP (U1) and the PIAs (U7&U8). Through the switch of Q6 it prevents false data from entering the CMOS RAM (U4) during power-up and utilizing D3, R35, U11 & U12 it sets the mode of operation for the microprocesser. The circuitry forces a 010 code on P20, P21, and P22 of the µP during RESET causing it to come up in an internal RAM external ROM, multiplexed address/data mode.

MICROPROCESSOR,BUS DEMUX,ADDRESS DECODE AND PROGRAM ROM

The Control Module uses a single chip microcomputer the MC6803 as its µP (U1). This µP provides two I/O ports,128 bytes of RAM, a multifunction timer and external ROM capibility. A bus cycle begins on the MC6803 with the Address/Data and R/W lines changing to a known state. Shortly after they are stable the Address strobe clock is output. This is used to latch the low order address lines A0-A7 from the AD0-AD7 bus via U6. After the Address strobe goes low the AD0-AD7 lines become the D0-D7 data bus. One half of a bus cycle later the E (enable) clock output goes high. The addressed device on the bus places data on the AD0-AD7 (R/W high) or takes its data from AD0-AD7 (R/W low) during the E clock . The bus cycle terminates when E goes low . Addresses are decoded by U9 & U10 to determine which external bus device the MC6803 is accessing. The E clock is used to qualify the decoding in timing the data transfer. U5 is an octal bus transceiver which controls the direction of data transmission utilizing the R/W line from the processor chip. The program is provided by U2 & U3 ROMs.These 28 pin sites accept a 128Kbit ROM giving the board a maximum of 32Kbytes of program storage.

1. Control Board Self Test:

The Control Board has as part of its program (U2 &U3) a subroutine designed to check the module each time the 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 contained in integrated circuits U2 & U3, the MPU's internal scratch pad memory, the CMOS memory (U4), the I/O chips (peripheral interface adadaptors-PIAs) U7 & U8, and the display and zero crossing interupt circuits. If the Control Module finds fault during the course of the self-test, it stops at that point in the test and will not allow game play. It should be noted that the early games using this system did not require U2 for game features and therefore the test for that chip was omitted. These games consisted of Eight Ball Champ ,Beat the Clock and Lady Luck.

A) 1st Flash

After reset the Control Module tests its µP and on chip RAM. It attempts to write -then read back all 256 patterns (0000 0000 to 1111 1111) in each of the 128 on-chip locations. If at any point in this test the µP fails to correctly read back a pattern it has written, it is deemed defective and the µP will not allow the game to come up. If the µP completes the test sucessfully, it flashes the LED and procedes to the next test.

B) 2nd Flash

Next the MPU (U1) attempts to test the ROM U2. It does a vertical checksum of the ROM contents and checks for an all ones result. If the computed checksum is not accurate, the part is deemed defective and the µP will not allow game play. If the checksum is 1111 1111 the µP flashes the LED and procedes to the next test.

C) 3rd Flash

An identical test is now performed on ROM U3. It does a vertical checksum of the ROM contents and checks for an all ones result. If the computed checksum is not accurate, the part is deemed defective and the µP will not allow game play. If the checksum is 1111 1111 the µP flashes the LED and procedes to the next test.

D) 4th Flash

The MPU chip accesses the CMOS RAM U4 (read-write memory). It makes a copy of the contents of the first half of U4. This is necessary because U4 is the battery supplied, non- volitile memory location where the bookkeeping functions and game set-up are stored. It then erases the contents of the first byte of U4 (U4 contains 2Kbytes of "scratch pad" memory). It trys to read back the word 0000 0000. If it can be read back , it adds "1" to the previous word (new word , 0000 0001). It continues to write and read until it reads the word 1111 1111. When this has been done successfully, it repeats the process on the next byte and the following bytes thereafter until it completes the test on byte 1024 . It then restores the memory to the first half of the RAM and saves the contents of the second half. It repeats the process for each of the remaining 1024 bytes one byte at a time and then restores the memory to the second half of RAM. If the MPU, at the end of these tests has read back correctly each of the words it has written, the MPU causes the LED to flash the fourth time.

E) 5th Flash

The µP now tests the PIA U8. it tests each of the two full byte port initialization registers with a 256 pattern test (0000 0000 to 1111 1111). It tests each of the two full byte I/O registers, PA0-PA7 and PB0-PB7 with a 256 pattern test. it then tests the CA2 and CB2 ports. These are initialized as outputs then written into to see if they will store a "1" and a "0". When both these ports are found good, the µP flashes the LED and proceeds to the next test.....

F) 6th Flash
The MPU chips repeats the same procedure it did for the fifth flash except this time it checks PIA U7. If no faults are found it procedes to the next test.

G) 7th Flash
The MPU now monitors its internal "clock". This clock is utilized as a Display Interrupt Generator, creating a pulse once every 1.2 milliseconds. If this interrupt pulse is not detected in U1, the Program will not allow game play until the fault is corrected.
Please note: Jumper JW7, when connected, will disable the Display Interrupt Generator in the MPU chip. This option is provided exclusively for troubleshooting with an oscilloscope.

H) 8th Flash
The MPU chip now monitors PIA port CB1 U8. If transitions from high to low are detected the MPU decides that the Phase B Zero Crossing detector is working. It then causes the LED to flash the 8th time.
If U12,C15,or D9 fails and the CB1 line is stuck high or low the program will not allow game play until the problem is corrected. It is to be noted that this Zero Crossing input is the Phase B switched illumination supply. If the fuse in that line (F5 on the power module) is blown when the power is turned on, the program will not allow game play until the fault on the Phase B line is corrected.

I) 9th Flash
To complete the power up sequence, the MPU now monitors its P20 port U1 pin 8. If transitions from low to high are detected the MPU decides that the Phase A Zero Crossing detector is working. It Then causes the LED to flash the ninth time.
If U11, U12, C16, or D11 fails and the P20 line is stuck high or low the program will not allow game play until the problem is corrected. This particular line is the Phase A switched illumination supply. If the fuse on that line (F4 on the Power Module) is blown when the power is turned on, the program will not allow game play until the fault on the Phase A line is corrected.

J) Game Initialization
The MPU chip, through the program, now initializes the two PIAs, U7 and U8, assigning to each port its role as either an input or output, as required.
It then verifies the integrety of the CMOS RAM U4 information by checking certain bytes to determine if Battery failure has occured or if possibly the +5VDC supply was interupted during a previous write operation to RAM. Should an error be detected the program sets specific registers to factory default conditions.
The game now enters a selective Stuck Switch test to display to the Operator any switches that may effect normal game operation which should be open but are not. It also Resets Drop targets that may be down and kicks out balls remaining in Saucers from the last time the game was played.
The game now enters an attract mode - flashing lights, showing the score thresholds and current credits on the displays, and monitoring the coin and credit switches for closure.

2. Game Play
After completing the self test, or in 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 solenoid, lamp momentary switch scanning, and lamp update interrupts.

A)Normal Mode:

The momentary switches are arranged in a "matrix ".The MPU chip, through the program, examines a memory record of the matrix, looking for valid switch closures. If it finds a valid closure, it decodes the address associated with the closure and jumps to the appropiate subroutine.

For example: If the game is in a game over status and the MPU finds that the left coin switch has a valid closure in memory, it jumps to the coin /credit handling routine in the program. This routine reviews the memory record in bookeeping to determine if the maximum credits have been reached. If they have, the coin will not be acknowledged. TheMPU goes back to monitoring the record of switch closures. If the maximum credits have not been reached, the memory record in bookkeeping 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 through the left chute are increased by one .

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

The memory record of valid switch closures is a qualified memory record.The MPU,as discussed under "Interupts", 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 game 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 momentry switches in the matrix are the "eyes" and "ears" of the MPU.It is only by means of sensing closures, and reacting to valid closures (during normal operation) that the MPU,through the program, knows what 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 the PIA-U8,CA1 or CB1 or from the peripheral port on the µP-U1, P20, 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 to service the interrupt .When the interrupt 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 interupted.

Interrupts are used for two types of activity in the Bally game. The first is the periodic lamp, solenoid, and momentary switch status update U8, CB1 & U1, P20 and the Display update internal to U1.The second is the signal to go into Self-Diagnostic tests U8, CA1.

The periodic interrupts are generated by the Phase A & Phase B Zero Crossing detectors and the Display interrupt generator on the Control module.The former occurs at a rate of 120 times per second or once each power line zero crossing (60 for Phase A and 60 for Phase B). The second occurs internally in U1 at a rate of approximately 830 times per second.

1)Zero Crossing interrupts:120 times per second, or once each 8.3 milliseconds, the MPU chip senses a zero crossing, time delayed by U12 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 counter 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.
At the start of this routine the MPU looks at the contents of several general purpose timers. 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 nine, 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 *solinoid 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 the ball clears the saucer.
The MPU next enters the *feature lamp update* part of the program.Their are 90 single bit entries in U4, the CMOS scratch pad memory. 45 bits corresponding to Phase A and 45 Bits corresponding to Phase B. This is utilized to form a memory "picture" (lamp matrix) of the status (on or off) of each feature lamp in the game.
The MPU will now aquire the first 3 bits from memory, combine it with an appropiate half byte address and send this address and data to the lamp decoders via PIA U8, ports PA0 thru PA7. The low order ports carry the first decode address (0000) generated by the MPU chip thru the program. The high order ports, PA5 thru PA7 contain lamp data from the first 3 bits in memory. The address (0000) goes to each of the three "one of sixteen" decoder chips in the lamp driver section via the lines labeled PA00, PA01, PA02, and PA03 . 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. PA5 goes to U15, PA6 goes to U16 and PA7 goes to U17.
It is to be noted that pin 11 of chip U15 drives SCR Q23, U16 drives SCR Q70, and U17 drives SCR Q55. Conclusion: the first 3 bits in the lamp matrix in memory chip U4 is a picture of the status of the lamps driven by these three SCRs.
The MPU chip, thru its program, now causes the strobe line (CB2, PIA U8) to go high and low, thereby presenting the first 3 bits of update information to the gates of SCRs Q23, Q70, and Q55. A low (0) at the gate leaves the SCR and it's 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 stay off unless the next update again drives the gate high.
The MPU fetches the second 3 bits from memory, generates an address (0001) and repeats the process. It is now addressing the gates of SCR Q24, SCR Q41, and SCR Q56. It causes the strobe to go high and then low, driving the gates of these SCRs and thereby updating their corresponding lamps.
The MPU fetches the third 3 bits from memory, generates an address (0010) etc. It is now addressing the gates of SCR Q25, SCR Q42, and SCR Q57. It repeats the strobe pulse and the appropriate lamps are updated.
Twelve more quick passes thru the subroutine and each of the 45 SCRs in the lamp section are updated. (Note that <u>not</u> all 45 SCRs are necessarily used in a given game.)The SCRs 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 (1/60 of a second).
The next step in the lamp update program is to strobe an address (1111) into the chips U15, U16, and U17. This is a "rest" address and frees the PA0 thru PA7 lines for other purposes. This completes the sequence for updating Phase A lamps and now the whole process is repeated for the Phase B lamps.

The majority of lamps in the game each have one lead commoned to an 11VAC line (Phase A or Phase B). In addition each lamp has a blocking diode tied in series with its phase line to preserve the integrety of that phase.Since the common lines are 90 degrees out of phase, the "B"lamps are all off when the "A" lamps are being updated and conversely the "A" lamps are off when the "B" lamp status is renewed. Since these updates occur so rapidly an observer could believe that both lamps appear to be on at the same time. This procedure allows a single SCR to control the state of two lamp circuits, one for each phase.
In games utilizing the Phase C & D lines the same principles apply with the following rules :
The Phase C line corresponds in timing to the Phase A line.
The Phase D line corresponds in timing to the Phase B line.
Each of these phases supply 24VAC for the bright light circuits.
And only the large SCR drivers (MCR 106-1) may be used in firing these lamps.
The last portion of the zero crossing interrupt routine is to read the *momentary switches* and look for valid closures. PIA U8, ports PB0 thru PB7 are initialized as inputs, PA0 thru PA7 as outputs.The MPU chip, thru the program, sends a pulse down the ST4 strobe line. If a switch is closed the pulse will return 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 a valid closure. Thus the 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" line 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 scaned and a memory record of the switches previous and current history is filed together with a record of valid switch 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 hit 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 Party Animal, for example, if diodes were not used: when the MPU sends the group strobe pulse down the ST1 line, and a coin is dropped through the right coin chute and both bottom and middle drop targets are down the strobe pulse will be returned down the "I1" line. The game will assume that a coin was dropped in both the left and right hand chutes and award the appropriate credits.
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 had never been interrupted.

2)Display Update Interrupt: 830 times per second the MPU senses an internal display update interrupt. The MPU makes a memory record of its contents and then jumps to service the interrupt.
There is also a memory record for each digit on each of the two displays in the CMOS memory, U4. 14 x 2=28 one byte memory locations are reserved for retaining this data. Each byte is capable of storing 256 states, representing all possible combinations of the segments that may be lit. Because many of these combinations would be unintelligible, we use an abbreviated version of an ASCII lookup table to represent the numbers 0 thru 9 and the letters A thru Z. When we refer to this table we are able to extract exactly what bits should be turned on to illuminate the corresponding segments in a particular digit.
The displays in the Bally Pinball 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" in both display driver modules.
The multiplex rate is fast enough that humans do not see the flicker. The advantage of multiplexing is that it minimizes the number of leads necessary to control the displays. For example without multiplexing you would need 8 leads for each segment on a digit, times14 digits, times 2 displays equals 224 leads and thats not even including commons or power leads.
The segment data lines (PA10-PA17) for both displays are commoned. The same is true for each of the Binary Digit Select lines (PA04-PA07) and the blanking line. Only a separate display latch strobe line is required for each of the display driver modules. It should also be noted that the comma information is time shared with PA16 & PA17 for Players1and 2 and PA14 &PA15 for Players 3 & 4.
The MPU 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 chip goes into memory and obtains the segment data for the fifth digit, for the Player 3 & 4 Display. It places this information on the PA10 thru PA17 lines and strobes it into U1 on the display module.
The MPU goes back to memory, obtains the segment data for the fifth digit, for the Player 1 & 2 Display. It places this information on the PA10 thru PA17 lines and strobes it into U1 on the Player 1 & 2 display module at the same time sending the Binary digit select for digit #5 into U2 of both displays.
With this process complete, it removes the blanking pulse thereby enabling digit #5 and returns from the interrupt to whatever it was doing previously.

Assume that the MPU chip is to enable digit #5. The Binary Digit select information 0101 (DCBA) is sent to U2 on both displays. When U2 is strobed, the base of level shifter transistor Q11 is made high. This causes the collector of Q11 to drop from a high, positive voltage (+190 volts DC less the leakage current drop across R31, 100k ohms) to the saturation voltage, approximately .3VDC. The voltage across R31 is now approximately 189VDC (190-.3(VCE SAT, Q11)-.7VDC (VBE, Q23).). The collector of Q23, which was clamped to the +80VDC bus, now rises to approximately 189.7 VDC (190-.3(VCE SAT, Q23). Digit #5 is now enabled.
Assume that the fifth digit of the 1st & 2nd Display module is to display the character "H". The MPU chip looks in the ASCII table and obtains the segment data 0111 0110. It then places this information into chip U1on the display via the PA10 thru PA17 lines. U1, when strobed, latches this input, and as soon as the MPU chip removes the blanking pulse, the bases of transistors Q13, Q2, Q15, Q26 and Q14 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 "b", "c", "e", "f" and "g" segments in the display panel are enabled.

Both of the actions of the previous example result in turning on the character "H" in the 5th digit position on the Player 1 & 2 Display.
It is interesting to note that the 6803 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 and 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.

Bally MIDWAY·SENTE MANUFACTURING COMPANY

SERVICE BULLETIN BOOK - 1987

Bally MIDWAY™
10601 W. Belmont Ave. • Franklin Park, Illinois 60131 • U.S.A.
Telephone: (312) 451-9200 • Fax No.: 312-451-4150
Cable Address: MIDCO • Telex No.: 72-1596