

[54] COMPUTERIZED PIN BALL MACHINE

[75] Inventors: Marion F. Bracha, Chicago; William H. Englehardt, Skokie, both of Ill.

[73] Assignee: Bally Manufacturing Corporation, Chicago, Ill.

[21] Appl. No.: 633,470

[22] Filed: Nov. 19, 1975

[51] Int. Cl.² A63F 7/00

[52] U.S. Cl. 273/121 A

[58] Field of Search 273/1 E, 85 R, 54 C, 273/118 A, 119 A, 121 A, 122 A, 125 A, 126 A, DIG. 28; 235/1 B, 92 GA, 156; 445/1; 340/172.5, 323, 337

[56] References Cited

U.S. PATENT DOCUMENTS

2,806,701	9/1957	Durant	273/125 A X
3,703,288	11/1972	Vogel et al.	273/126 A X
3,874,669	4/1975	Ariano	273/DIG. 28
3,889,956	6/1975	Castle	273/1 E X
3,907,290	9/1975	Fischer et al.	235/92 GA X
3,973,244	8/1976	Lovercheck et al.	364/200
4,008,893	2/1977	Yoseloff	273/85 R
4,026,555	5/1977	Kirschner et al.	273/102.2 B X
4,069,510	1/1978	Carlow et al.	364/200
4,093,232	6/1978	Nutting et al.	273/121 A

OTHER PUBLICATIONS

MCS-4™ Microcomputer Set Users Manual; Intel Corporation; Jan. 1972; pp. 44-47, 54-58, 60, 61.

Electronics; "Shrinking Cost, Growing Reliability Make More Things Happen," Oct. 25, 1973; pp. 71-105.

Electronics; "Microcomputers Muscle in," Mar. 1, 1973; pp. 63-64.

EE/Systems Engineering Today; "Electronics Plays the Pins," Nov. 1973; pp. 37-41.

Electronic Design; "Ease Multiplexing and A/D Conversion," Apr. 12, 1973; pp. 84-89.

Process Instruments and Controls Handbook; McGraw-Hill; 1957; pp. 8-34, to 8-47.

Electronics; "Scamp Microprocessor Aims to Replace Mechanical Logic," Sept. 18, 1975; pp. 81-87.

Popular Electronics; "Altair 8800 Minicomputer, Part I," Jan. 1975; pp. 33-38.

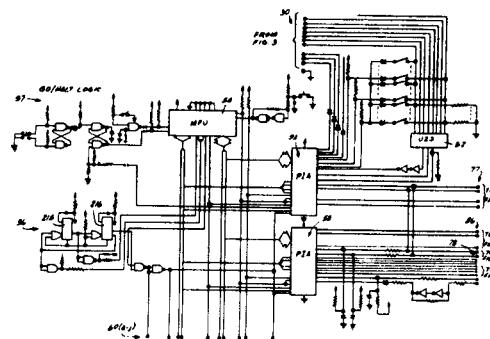
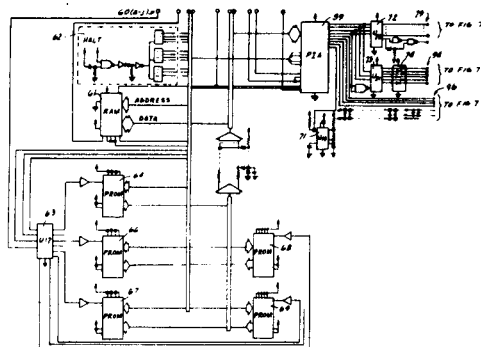
Primary Examiner—Vance Y. Hum

Attorney, Agent, or Firm—Fitch, Even & Tabin

[57] ABSTRACT

A pin ball machine which incorporates a micro processor instead of relays and hard wiring wherein the processor is programmed such that when the coin switches, the flipper switches and the various scoring switches of the machine are energized the computer accumulates and drives indicators to indicate the score as well as drives the flippers, the sling shots and other units of the playfield to provide an improved machine.

22 Claims, 32 Drawing Figures



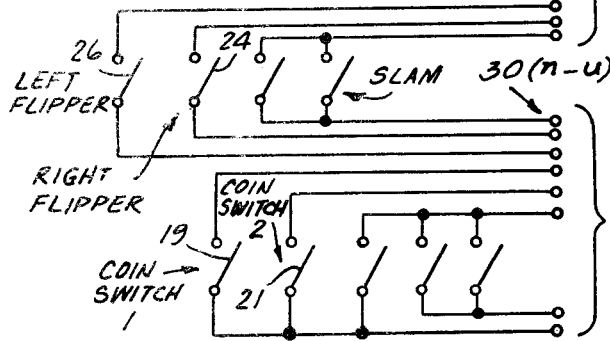
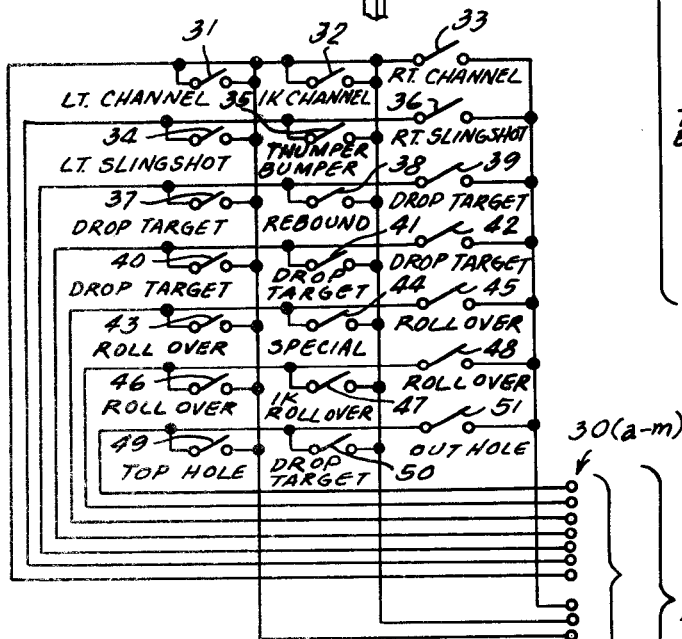
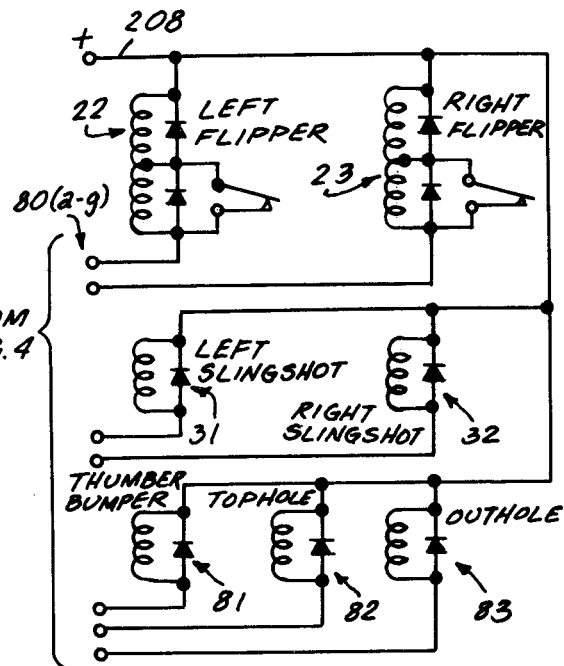
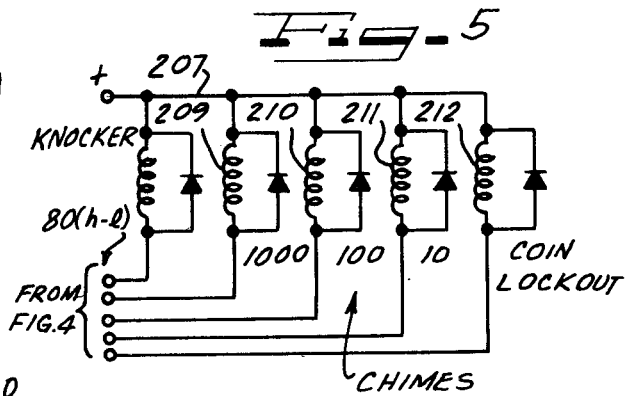
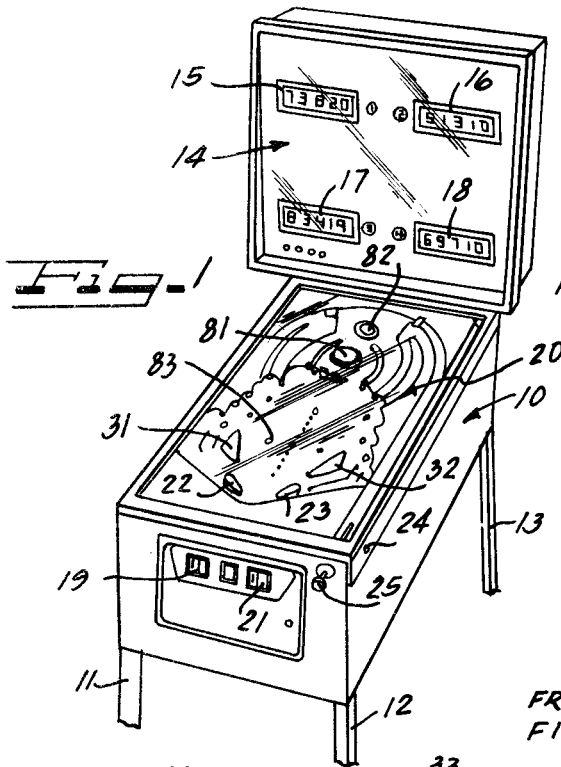
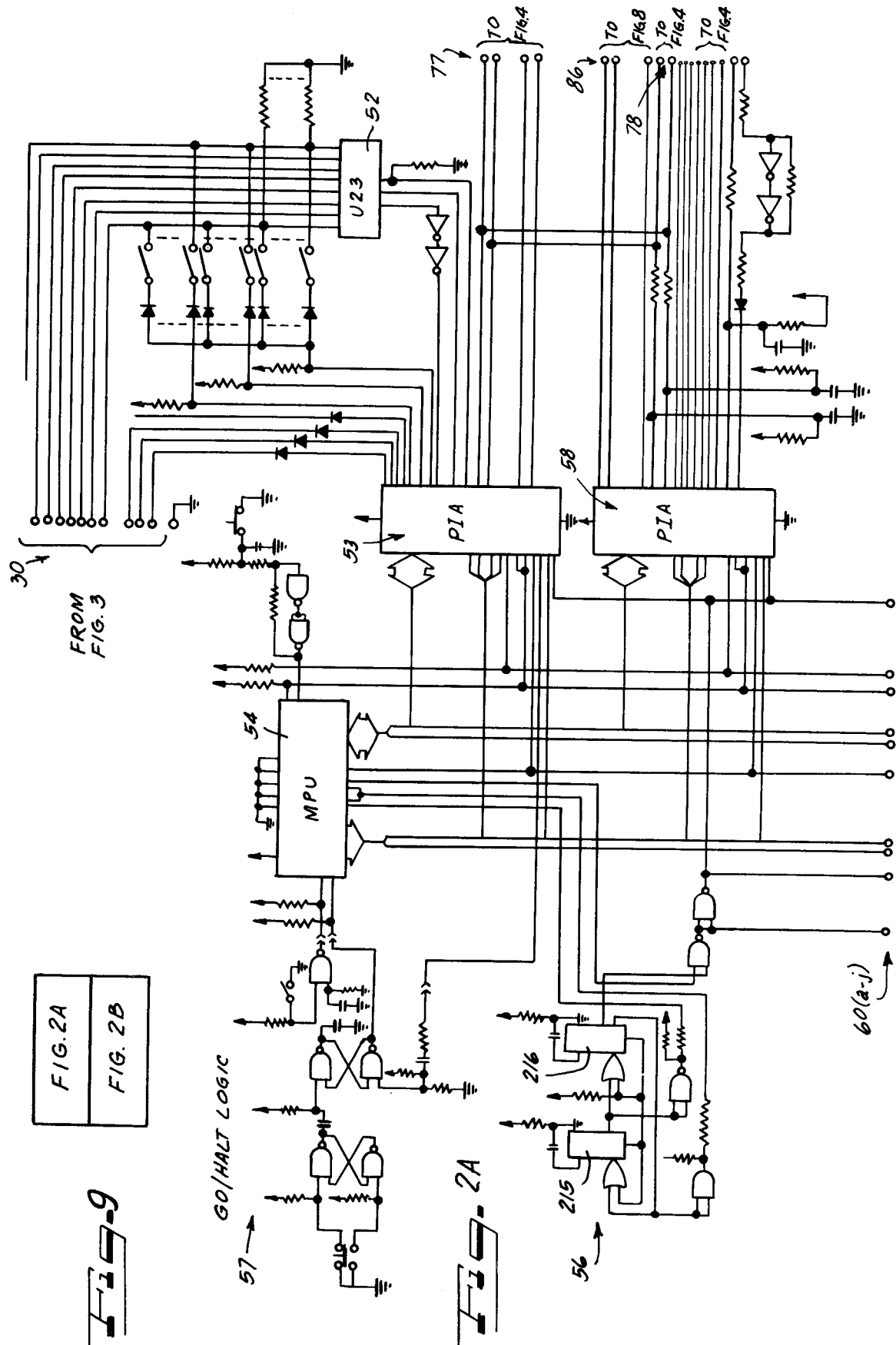
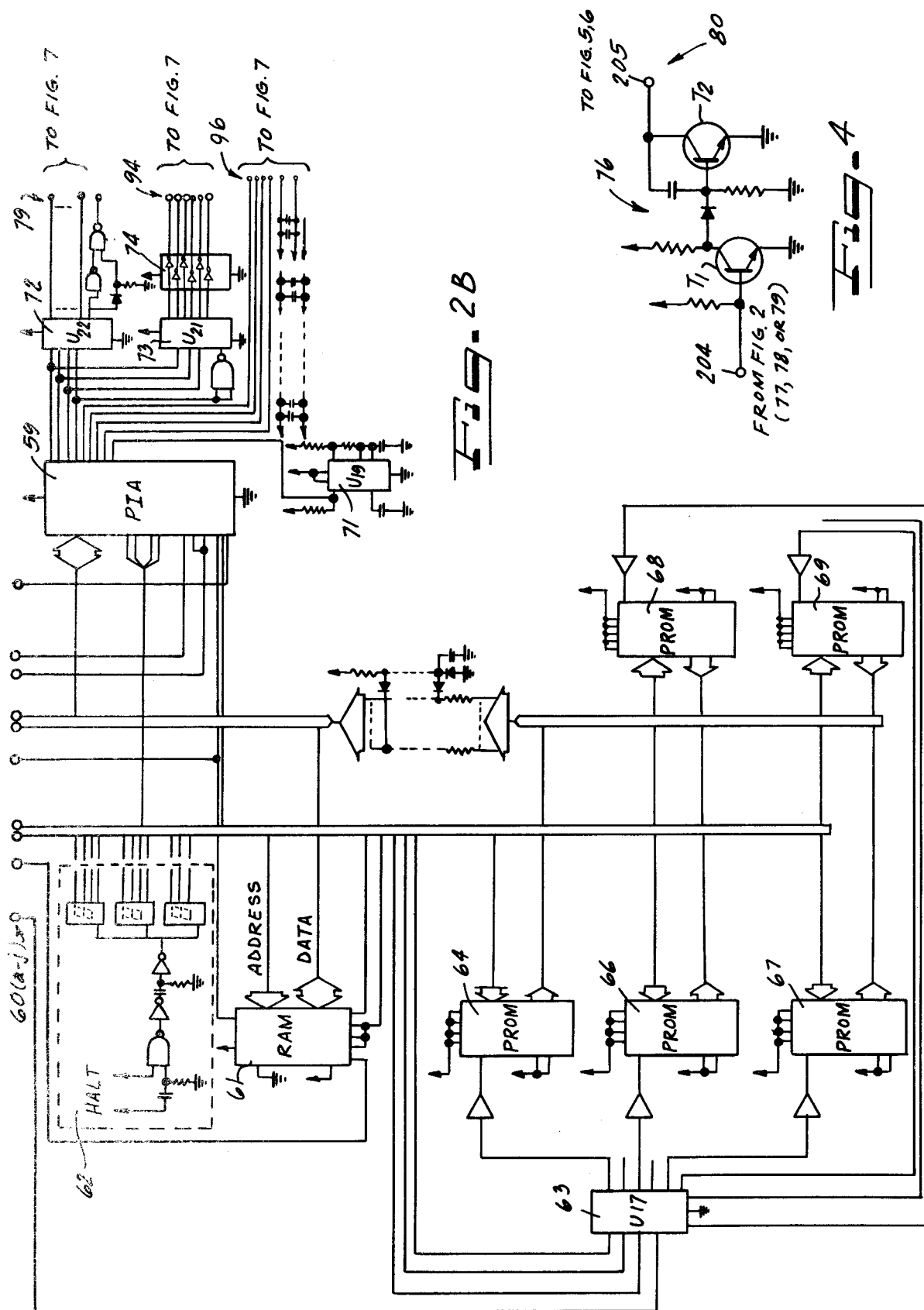
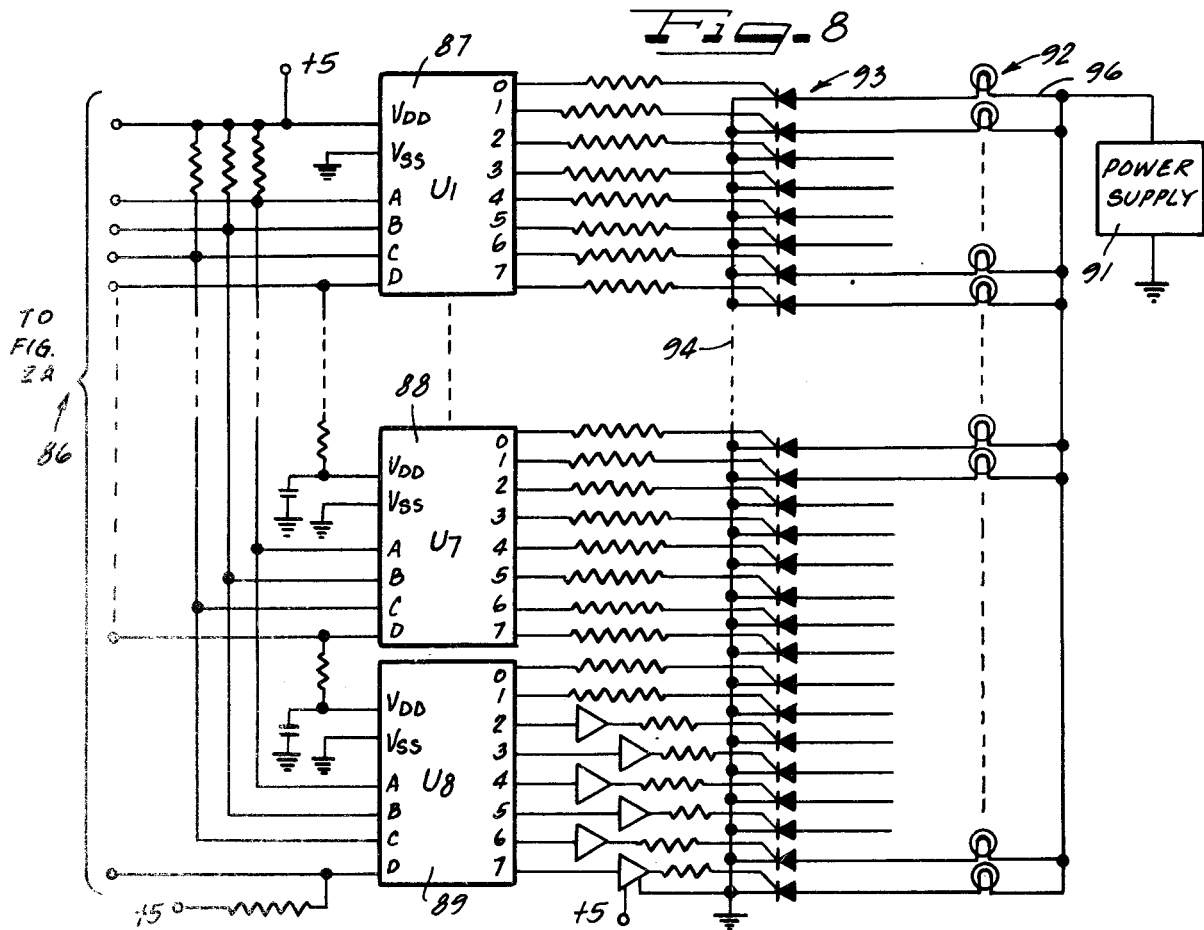
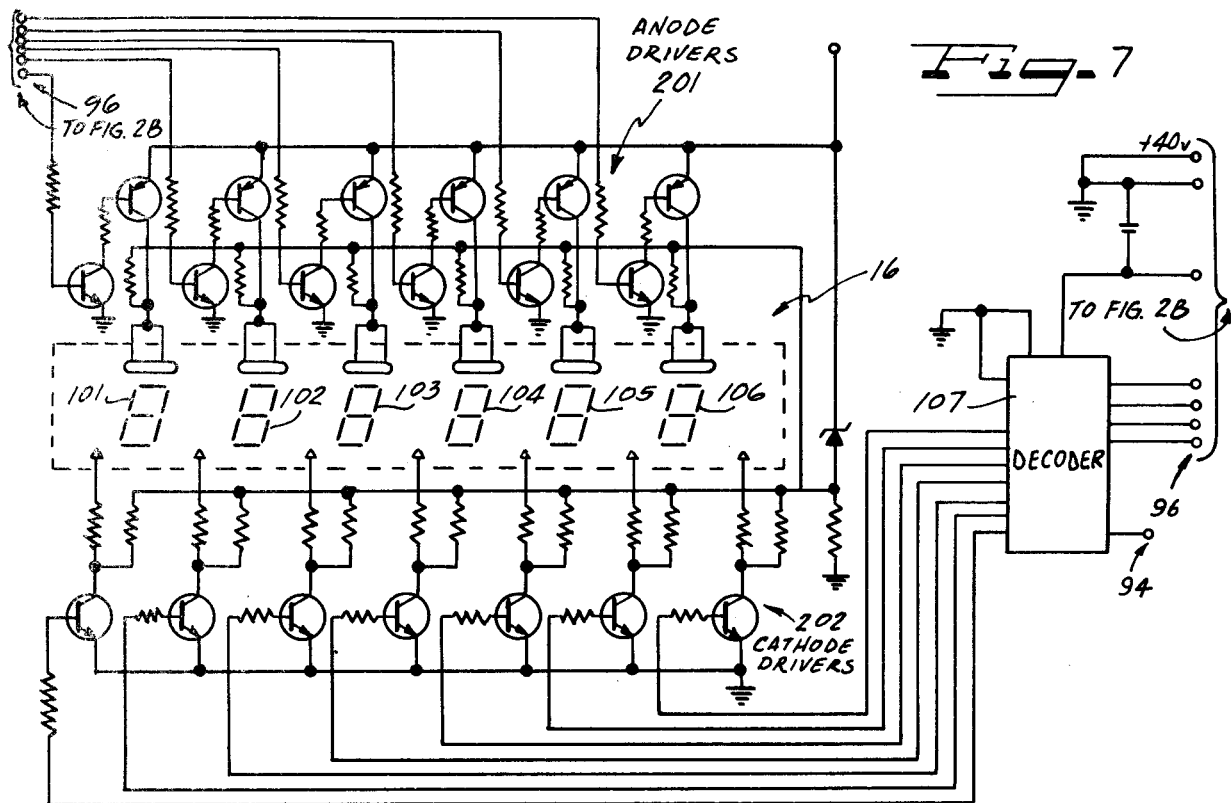
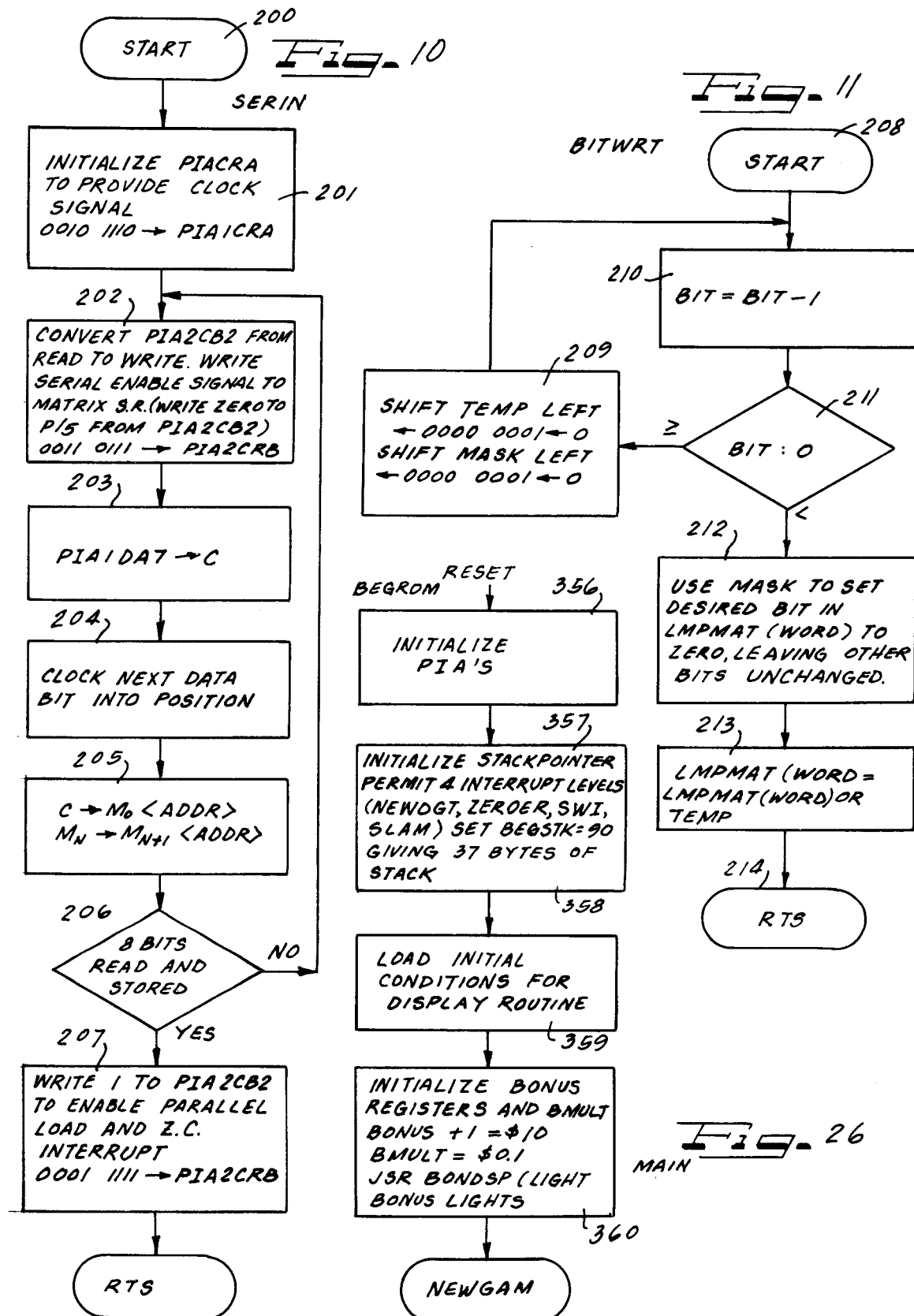


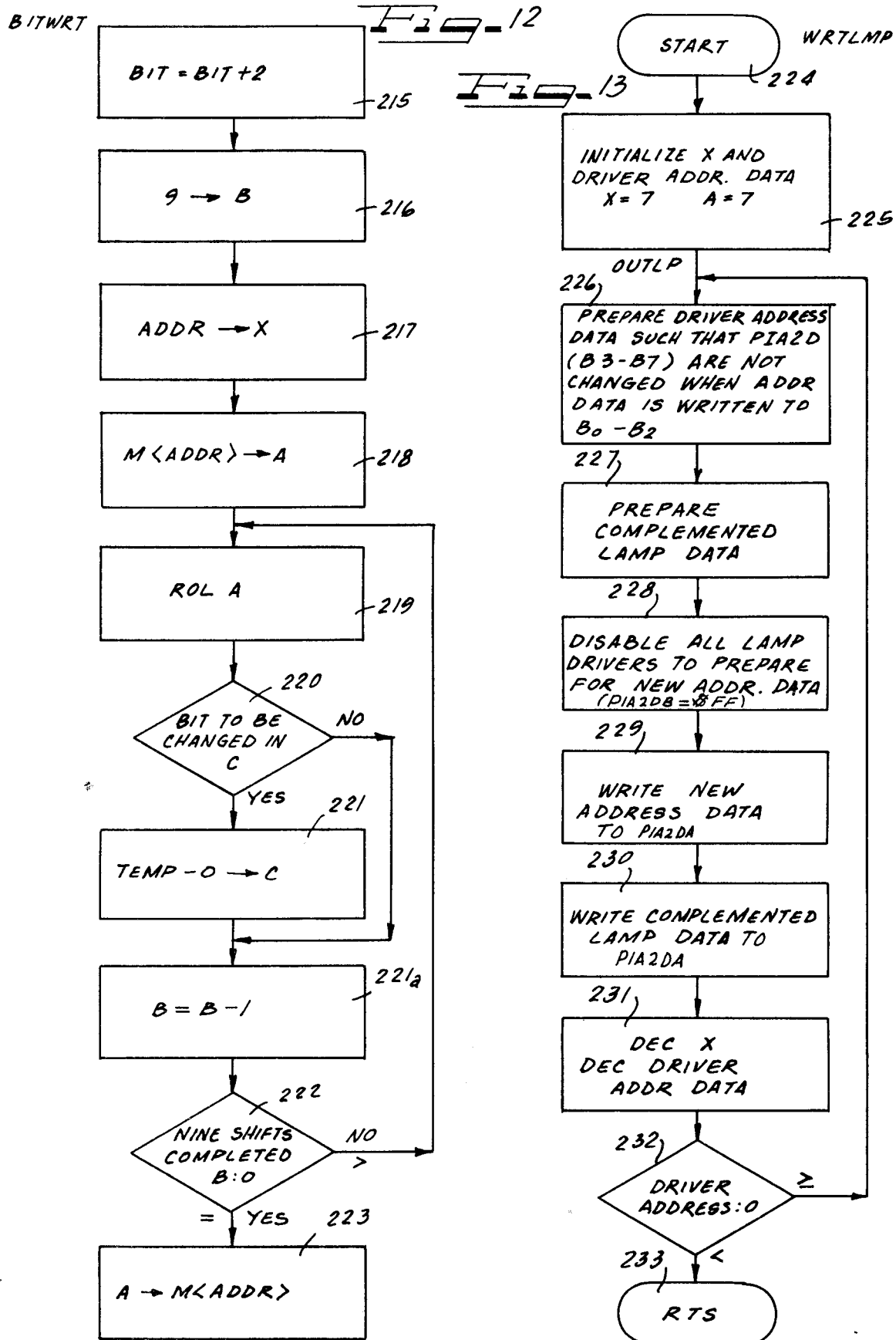
Fig. 3

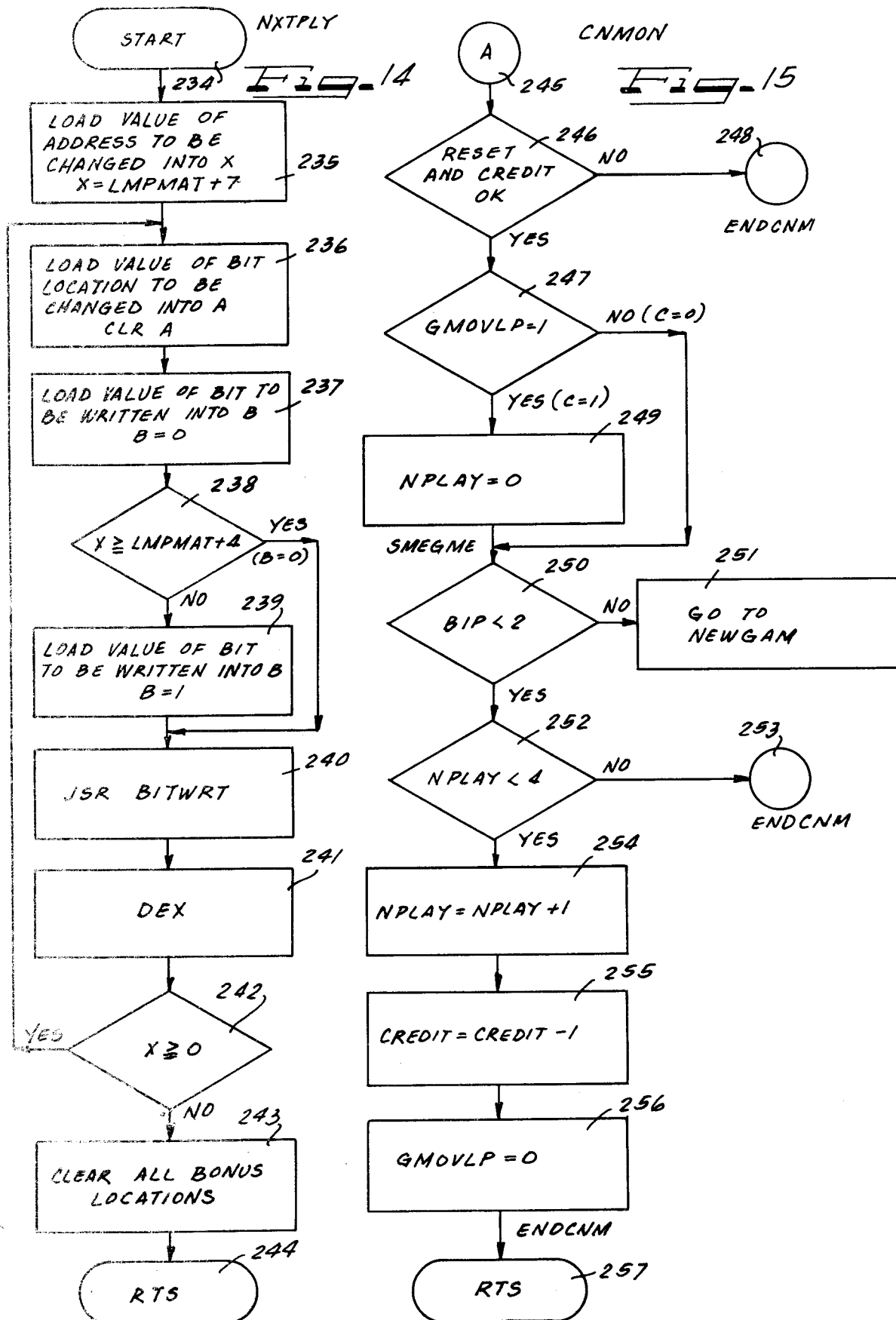


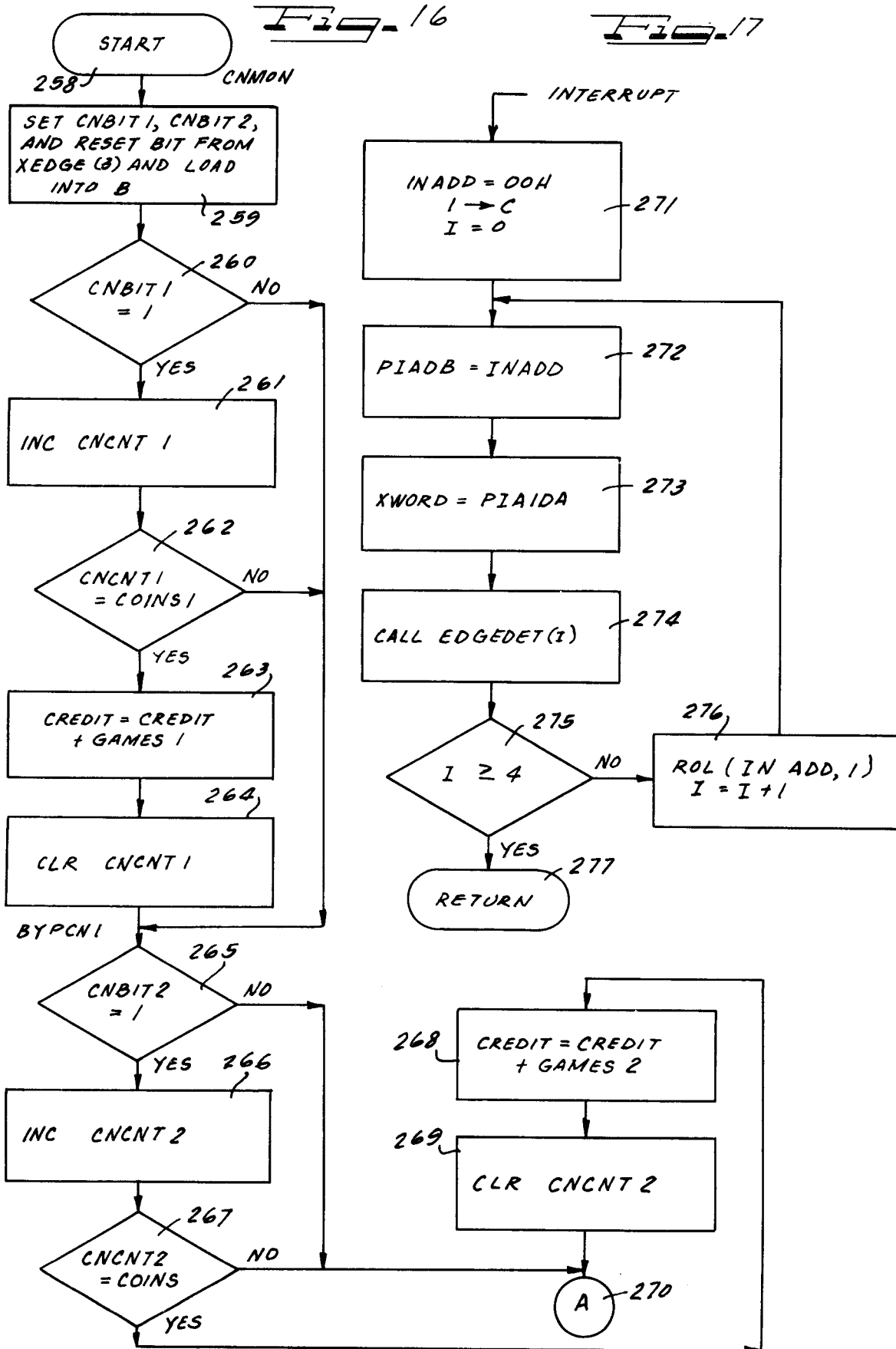












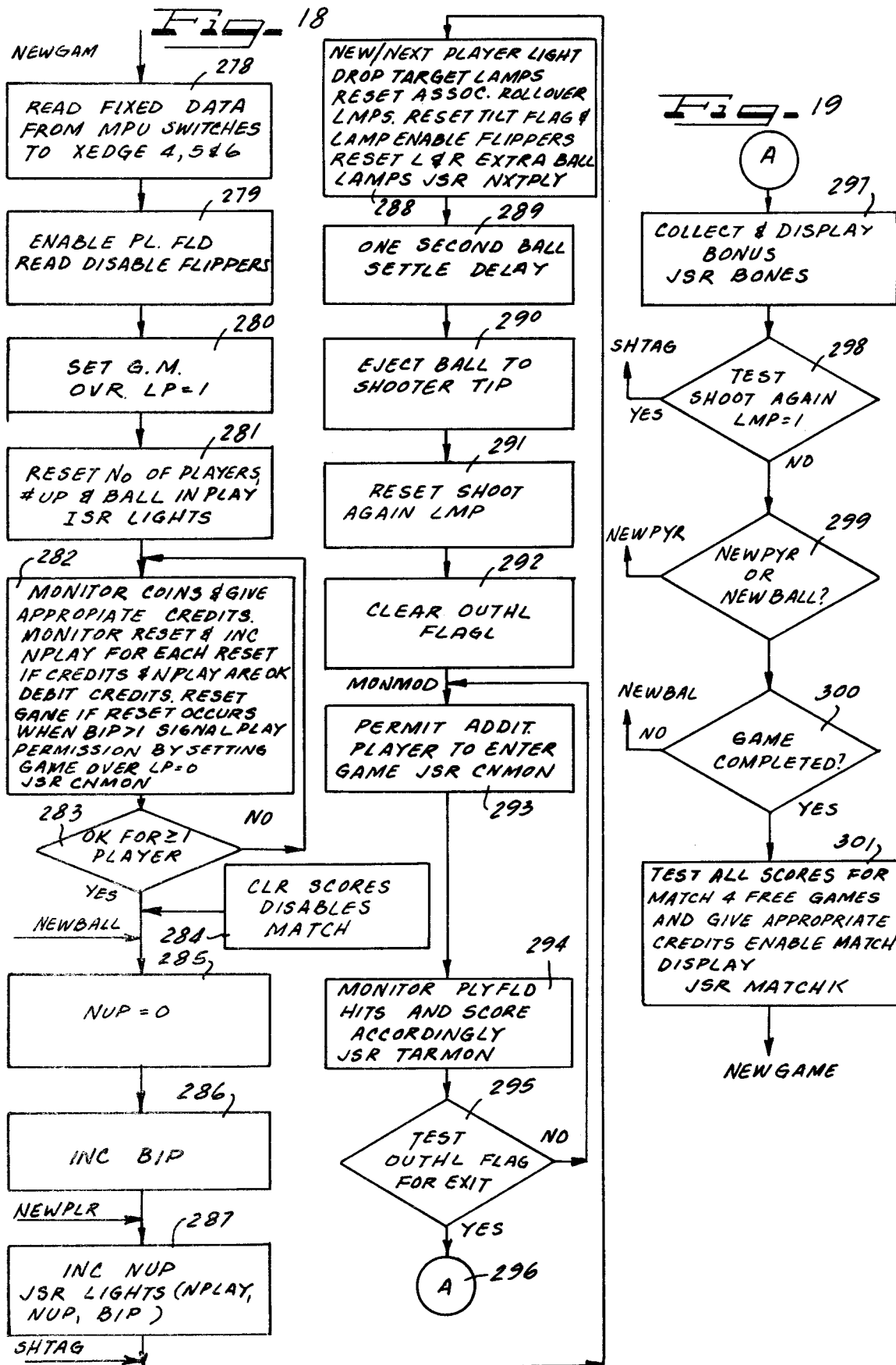
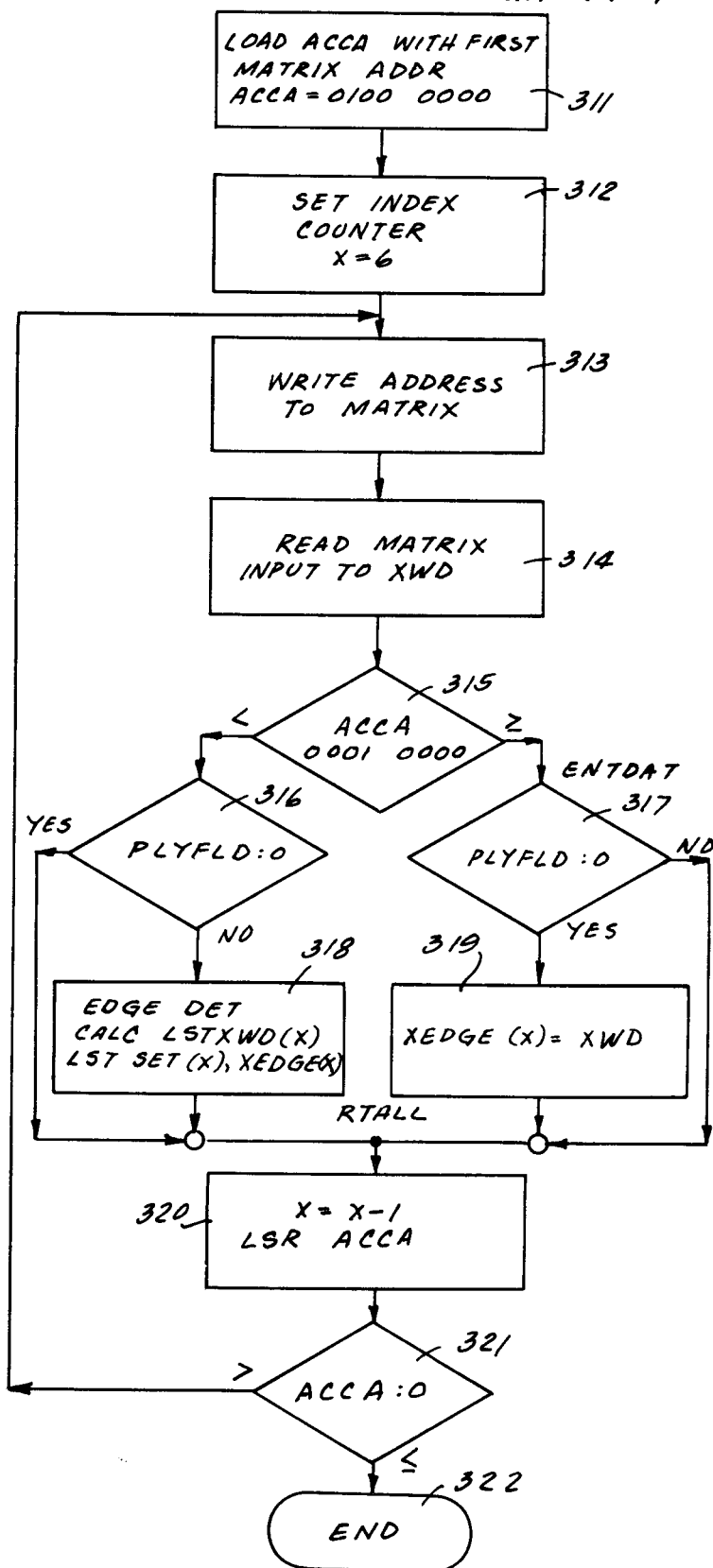
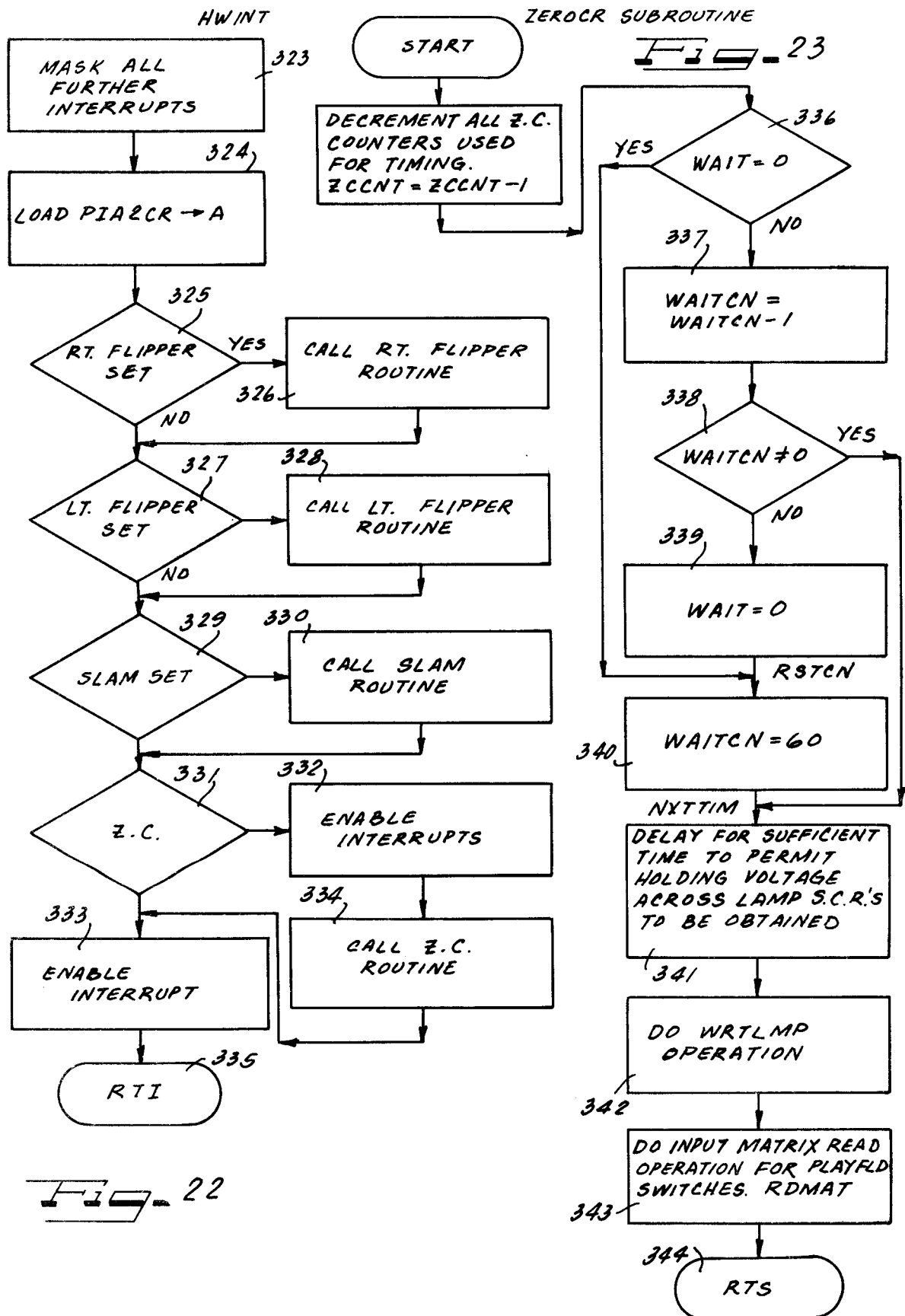
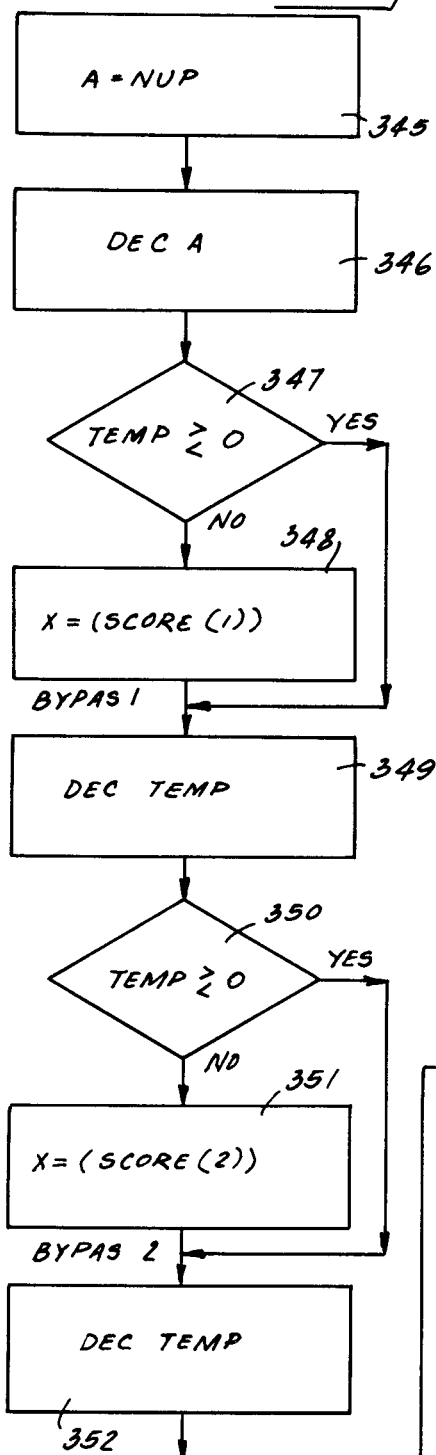
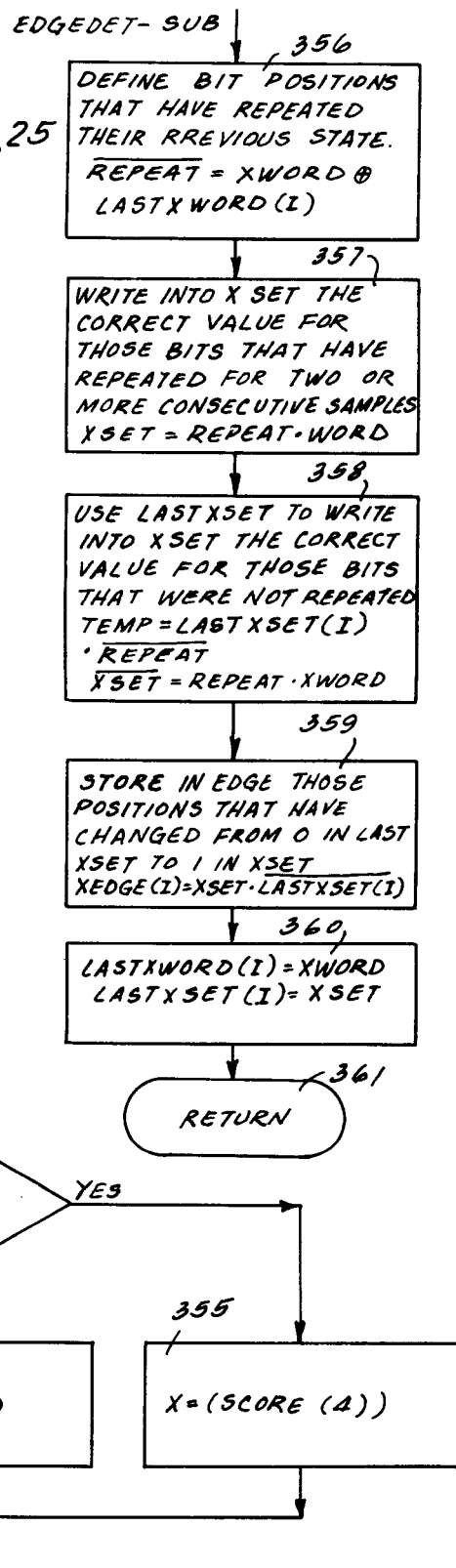


Fig. 21RDMAT
INTERRUPT



SCRPL Fig. 24Fig. 25

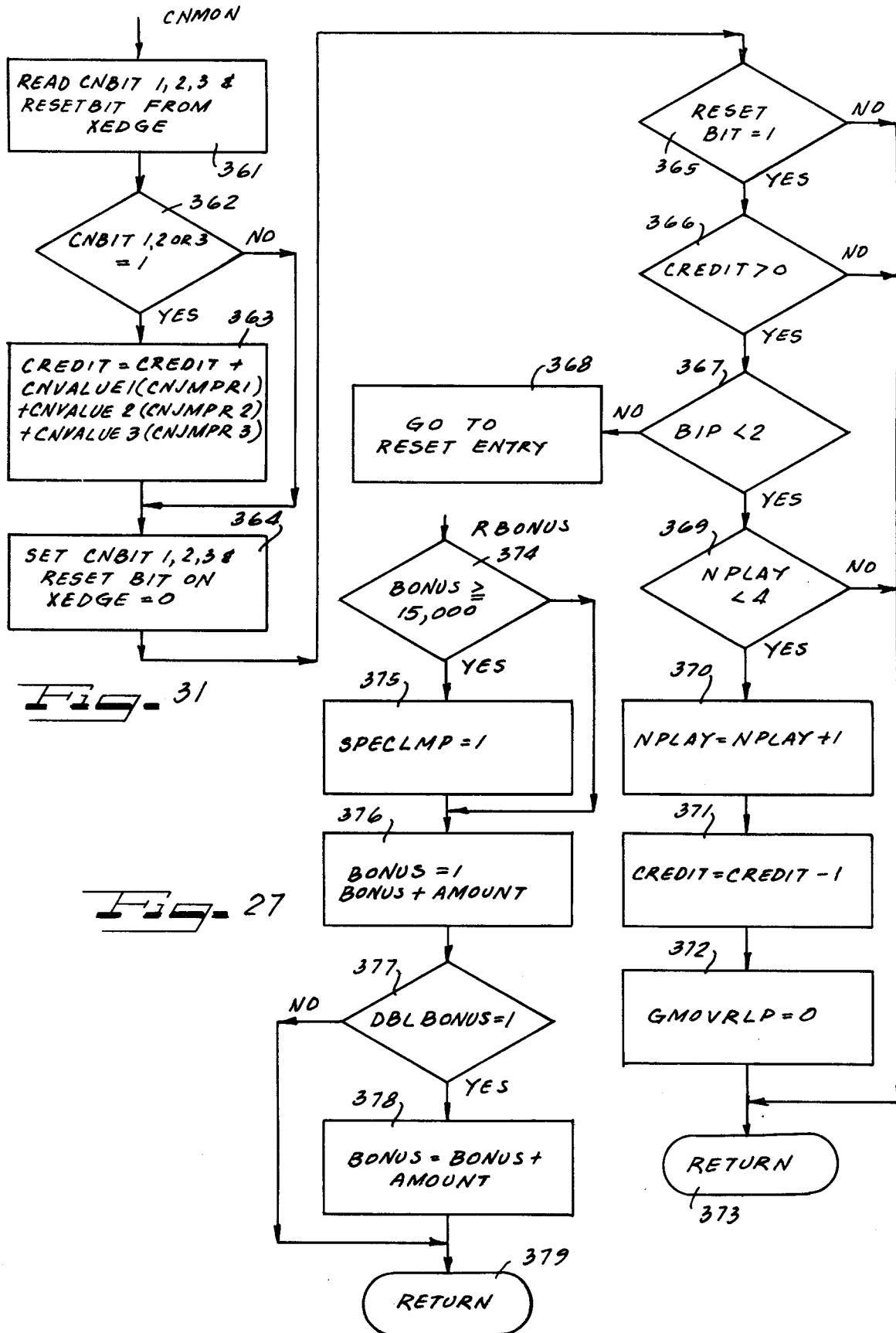
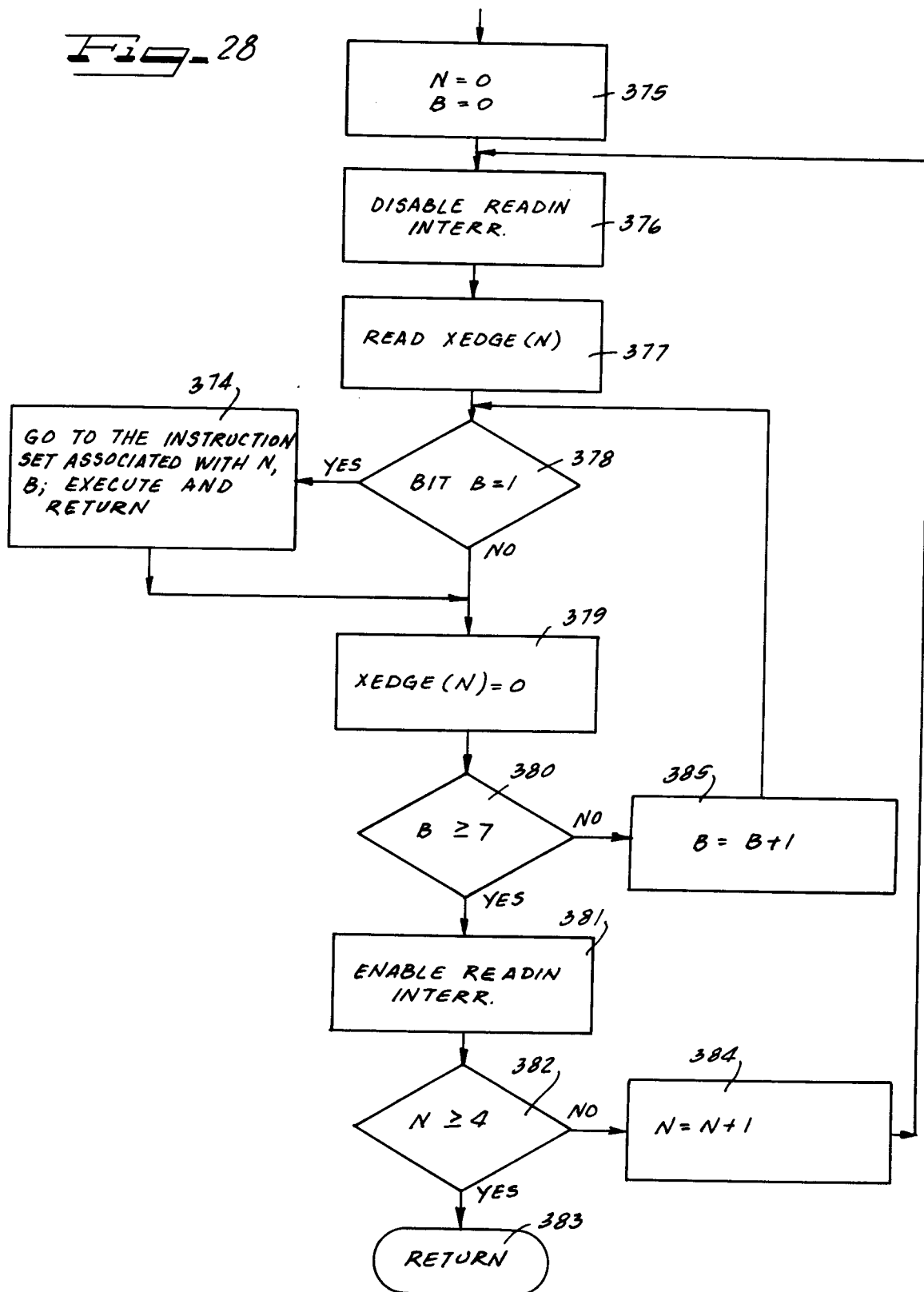


Fig. 28

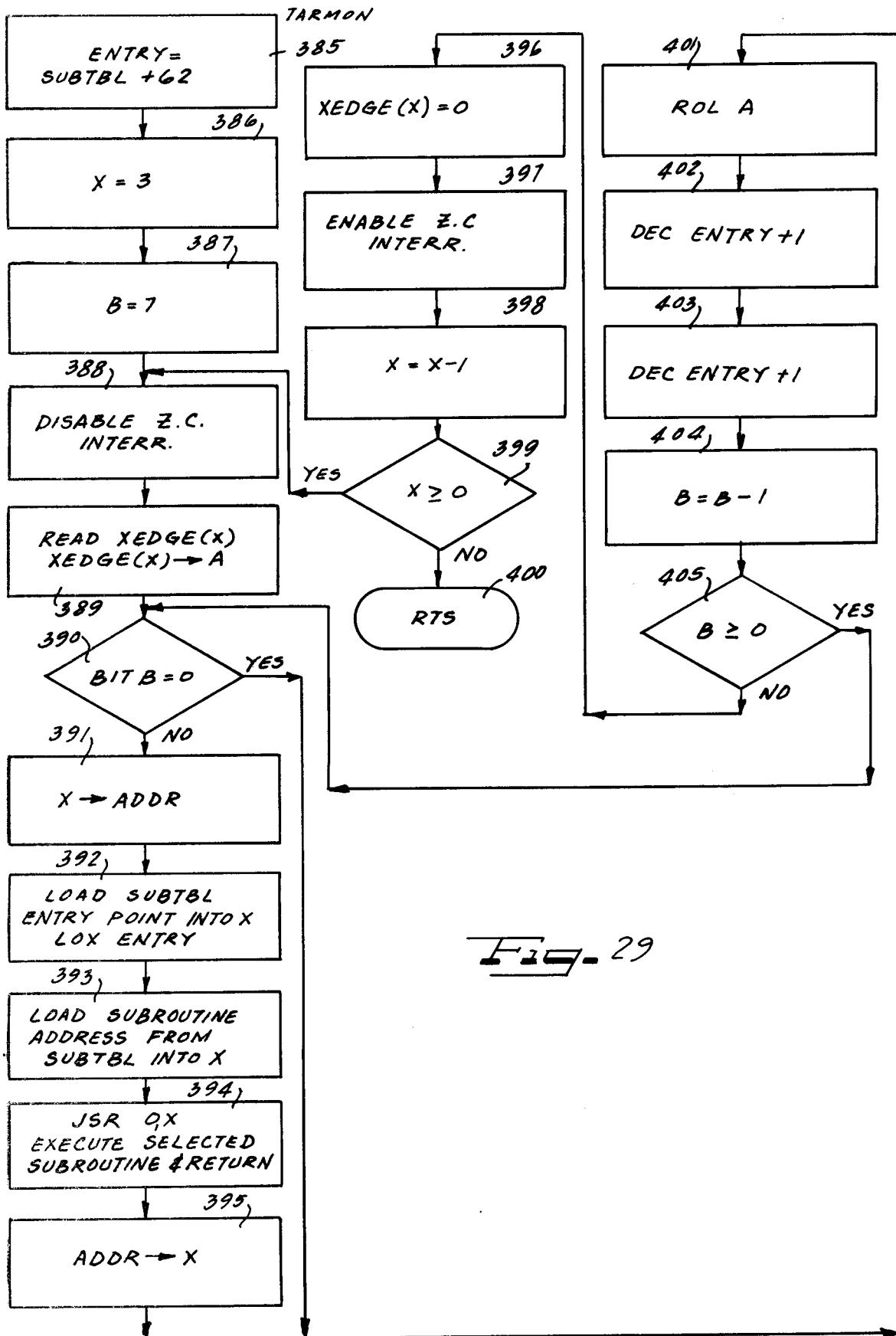
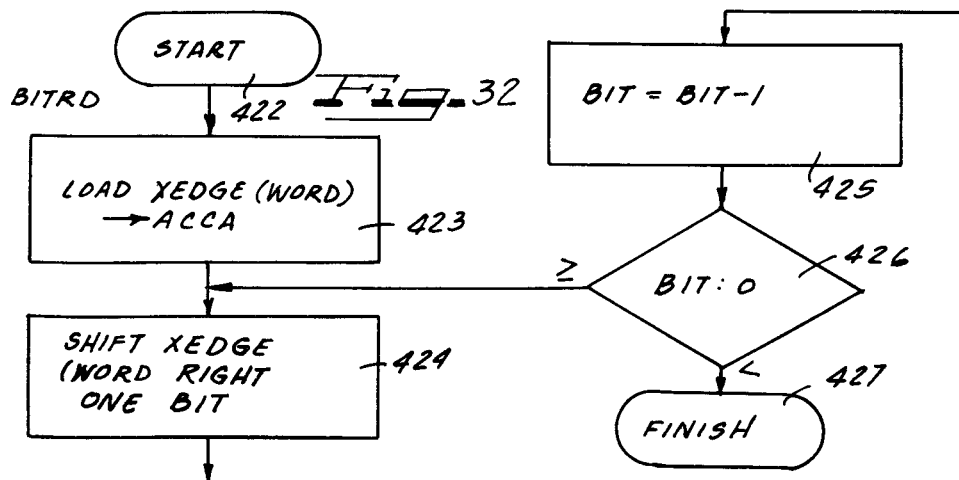
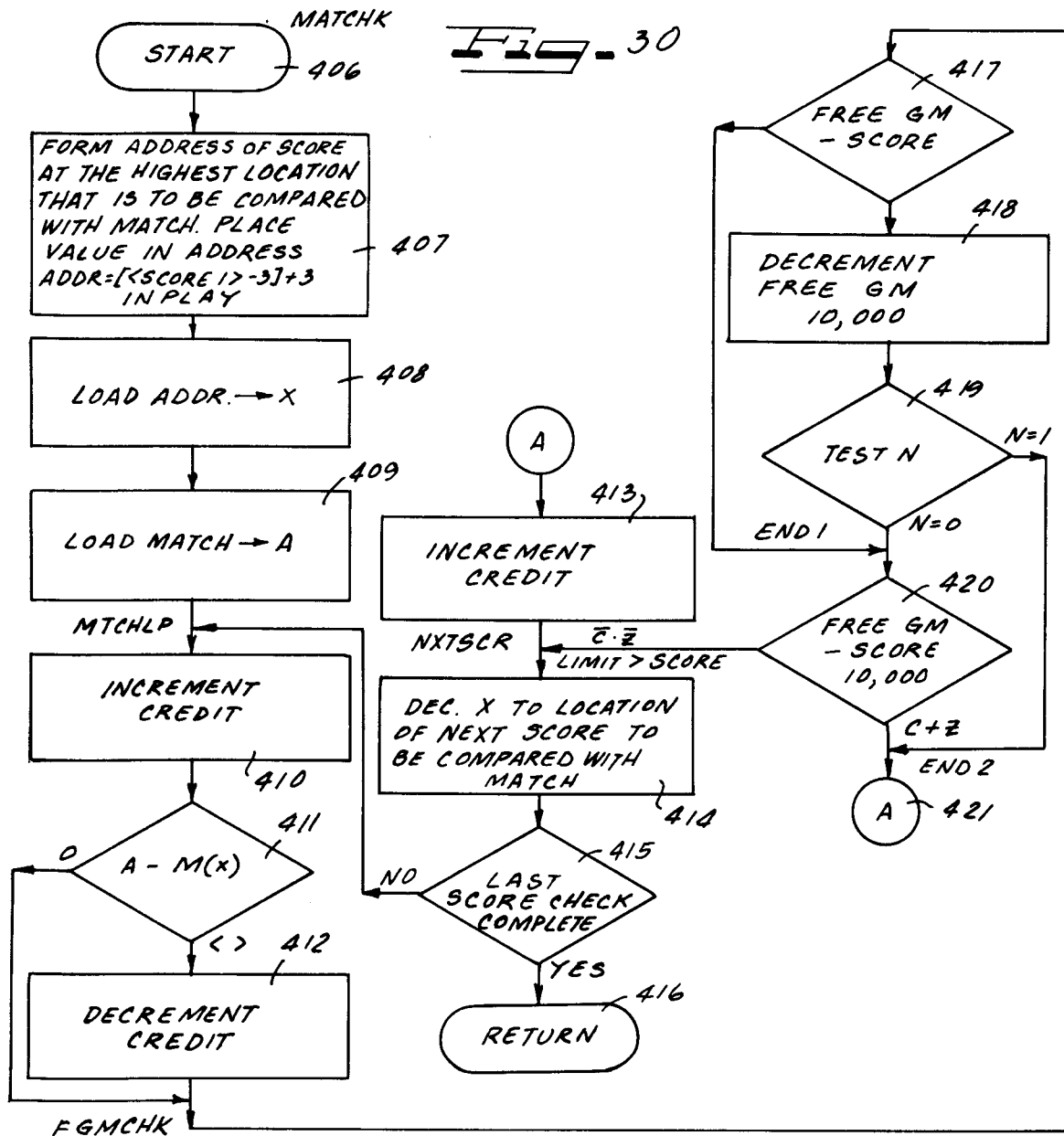


Fig. 29



COMPUTERIZED PIN BALL MACHINE

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates in general to amusement machines of the pin ball type and in particular to an improved computerized pin ball machine.

2. Description of the Prior Art

Pin ball machines of the prior art have utilized switches which actuate relays so as to accumulate and drive indicators, chimes, lights and other units of the machine.

SUMMARY OF THE INVENTION

The present invention incorporates a micro processor which is operated by a suitable program according to the invention and receives inputs from the playfield wiring of a pin ball machine and provides outputs to drive solenoids on the playfield such as the right flipper, the left flipper, the right sling shot, the left sling shot, the thumper, a top hole, an out hole and a knocker as well as indicator lamps to indicate the score and wherein the micro processor is programmed to produce the proper operation of the units on the playfield as well as indicate the score of each player.

Other objects, features and advantages of the invention will be readily apparent from the following description of certain preferred embodiments thereof taken in conjunction with the accompanying drawings although variations and modifications may be effected without departing from the spirit and scope of the novel concepts of the disclosure, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a prospective view of the invention;
FIGS. 2A and 2B are schematic views of the computer with its inputs and outputs;

FIG. 3 illustrates the input switches to the computer;

FIG. 4 illustrates a single solenoid drive circuit;

FIG. 5 illustrates units driven by solenoid drive circuits;

FIG. 6 illustrates units driven by solenoid drive circuits;

FIG. 7 illustrates digital score indicators driven by the computer;

FIG. 8 illustrates various lights driven by the computer; and

FIG. 9 illustrates the relationship of FIGS. 2A and 2B of the drawings.

FIGS. 10 through 32 illustrate flow charts for various programs and subprograms of the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention comprises a novel computerized pin ball machine wherein the conventional relays which are actuated by switches on the playfield are energized so as to store information as to which of the particular switches has been actuated by the balls as they roll on the playfield of the machine are replaced by a minicomputer which has been programmed by the novel program of the invention to control the operation and scoring and other functions of the machine without the hard wiring and relays of the prior art. The micro computer of the invention is connected to switches and indicators on the playfield of the machine and to the indicators on an indicating support member as well as to

indicator lights mounted on the playfield or other portions of the machine. The micro computer also receives inputs from the coin receiving switches of the machine. The micro computer comprises a single micro processing unit, a clock and a number of read only memories, random access memories and peripheral interface adapters.

The program for the micro computer is incorporated in this application and provides the soft ware for operation, control address and output of the micro computer.

FIG. 1 illustrates a pin ball machine 10 according to the invention which is mounted on legs 11, 12, 13 and on a fourth leg. A display region 14 at one end of the machine has display area 15, 16, 17 and 18 for indicating the scores of 1 to 4 players. A pair of coin slots 19 and 21 are provided. The playing field 20 of the machine 10 is provided with a number of switches and actuators so as to interact with balls on the playfield. The balls are propelled by a ball shooter 25, actuated by a player onto the playfield 20 and due to the momentum of the ball and the tilting surface of the playfield, the ball moves about the playfield to engage various switches, flippers, slingshots and Thumper Bumpers on the playfield. It should be realized that the arrangement of the playfield may differ. The machine illustrated in FIG. 1 has the Boomerang playfield manufactured by the Bally Manufacturing Corporation.

A tophole 82 is provided with an opening with a switch therein which will energize the scoring and further is provided with a solenoid so as to kick the ball out of the hole. The Thumper Bumper 81 can be energized by the ball as can the left Sling Shot 31 and the right Sling Shot 32. A pair of pivoted flippers 22 and 23 can be actuated under control of flipper switches 26 and 24, respectively. Numerous lights, chimes and action mechanisms are mounted on the playing field and are well known to those skilled in the art.

In prior art devices the various switches located on the playing field 20 have directly energized relays and switches so as to illuminate various lights, chimes and drum indicators, but in the present invention the switches on the playfield 20 provide inputs to a micro computer such as illustrated in FIGS. 2A and 2B which includes a microprocessing unit 54 which might be of a type M 6800 unit manufactured by Motorola Semiconductor Products, Inc. For example, the Micro Processing Unit 54 receives inputs from the playfield through the peripheral interface adapter 53 and provides output to drive various indicating lights 92 through the peripheral interface adapter 58. Also, the indicator units 15, 16, 17 and 18 are driven by the computer 54 through the peripheral interface adapter 59. The computer 54 also drives a plurality of solenoid drivers 76 illustrated in FIG. 4. The solenoid drivers 76 are connected to various actuated devices illustrated in FIGS. 5 and 6. It should be realized that there is a solenoid driver for each of the elements of FIGS. 5 and 6. The indicator units 15, 16, 17 and 18 are illustrated in FIG. 7 and are driven by the output of the computer 54 through the peripheral interface adapter 59. A plurality of indicator units 92 illustrated in FIG. 8 are driven by the computer through the peripheral interface adapter 58 by the computer.

The inputs of the computer are illustrated in FIG. 3 and comprise a first coin switch 19 and a second coin switch 21 and also comprise a left flipper switch 26 which can be actuated by the operator of the pin ball

machine and a right flipper switch 24. The coin switches 19 and 21 actuate the machine upon the deposit of a coin in the coin slots of the machine so as to reset the indicators 15 through 18 to 0 and enable the machine for a new game. Upon the deposit of a coin credit is given to the player for a number of balls which are automatically supplied to the ball ejector 25 one after the other so that they can be shot onto the playfield.

Mounted on the playfield are a number of ball actuated switches such as the switches 31 through 51 illustrated in FIG. 3. For example, the switch 31 is the LT 500 point channel switch. The switch 32 is the 1,000 point channel. The switch 33 is the right 500 point channel switch. The switch 34 is the left Sling Shot switch. The switch 35 is the Thumper Bumper switch. The switch 36 is the right Sling Slot switch. The switch 37 is the drop target "A" switch. The switch 38 is the rebound 10 switch. The switch 39 is the drop target "D". The switch 40 is the drop target "B". The switch 41 is the drop target 100 point switch. The switch 42 is the drop target "C", the switch 46 is the roll over "A" switch and the switch 48 is a special switch. The switch 45 is a roll over "D". The switch 46 is a roll over "B". The switch 47 is a 1,000 roll over switch and the switch 48 is a roll over "C". The switch 49 is a tophole, 3,000 point switch, the switch 50 is a drop target 500 point switch and switch 51 is an outhole switch.

The switches 19, 21, 24, 26 and 31 through 51 are connected to output terminals 30a through U which are supplied to the central processor 54 through the peripheral interface adapter 53 and the unit 52 identified as U23 and which may be a type CD 4,021 AE unit manufactured by Motorola. The input of switches of FIG. 3 are fed into the computer which stores, processes and provides control outputs to light indicators and solenoid drive switches to actuate various devices in the machine.

For example, the plurality of indicator lights 92 shown in FIG. 8 are driven by the computer and have first sides 96 connected to a power supply 92 which has its other side grounded and the lights have their second sides connected to ground through a plurality of SCR's 93 as shown in FIG. 8 such that if the SCR's 93 are gated to the "on" condition the SCR's will conduct allowing current to pass from the power supply through the lights 92 to ground to complete the circuit thus illuminating the lights. The gates of the SCR's 93 are connected to a plurality of one of eight decoders 87, 88 and 89 which receive inputs from terminals 86 which are also shown in FIG. 2A and are connected to the peripheral interface adapter 58. Thus, the provision of the peripheral decoder adapters 87, 88 and 89 allow a large number of lights 92 to be driven by a smaller number of input leads connected to the terminals 86.

FIG. 7 illustrates one of the indicators such as 15, 16, 17 and 18 which might be a Burrough's type BR08571 indicator having a plurality of indicator units 101 through 106 and having a plurality of anode drivers 201 which are respectively connected to anodes of one of the indicators 101 through 106 and receive inputs at terminals 79 which are connected to the output of a decoder 72 whose input is connected to the output of the peripheral interface adapter 59 illustrated in FIG. 2B. The plurality of cathode drivers 202 are connected through a decoder unit 107 which might be a type MC14543 CP type and which receives inputs from terminals 96 illustrated in FIG. 2B as connected to the

peripheral interface adapter 59 of the computer. The decoder unit 107 also receives an input from a terminal 94 illustrated in FIG. 2b as connected through inverter 74 to the output of a decoder 73 which has inputs connected to the output of the peripheral interface adapter 59. It is to be realized, of course, that the other four indicators are also driven by the output of the computer in the same fashion as the indicator 16.

FIG. 4 illustrates a single solenoid driver 76 which has an input terminal 204 which receives inputs from terminals 77, 78 or 79 of the computer illustrated in FIGS. 2A and 2B. The solenoid driver has a pair of transistors T1 and T2 and has an output terminal 205. It is to be realized that there is a solenoid driver such as 76 for each of the units illustrated in FIGS. 5 and 6. For example, in FIG. 5 the terminal 80h through k are each connected to the output terminal 205 of a solenoid driver 76 such as shown in FIG. 4 such that when the transistor T2 is turned on terminal 205 is connected to ground thus supplying power from positive terminal 207 through the energizing winding of the solenoids of the respective units. For example, terminal 80k is connected to a Klocker winding to energize a Klocker. Terminal 80i is connected to a 1,000 solenoid 209. Terminal 80j is connected to a 100 solenoid 210. Terminal 80k is connected to a 10 solenoid 211 and terminal 80l is connected to a coin lockout solenoid 212.

FIG. 6 illustrates terminals 80a through g with terminal 80a through g, respectively, connected to an output terminal of a solenoid driver such as 76 in FIG. 4 which has an output terminal 205 such that if the particular solenoid driver is energized, ground is applied to terminal 205 so as to complete the circuit from power lead 208 through the solenoids and transistors T2.

The solenoid 22 controls the left flipper. The solenoid 23 controls the right flipper. The solenoids 31 and 32, respectively, control the left and right Slingshots. Solenoid 81 controls the Tumper Bumper, the solenoid 82 controls the tophole solenoid and the solenoid 83 controls the outhole.

The computer illustrated in FIGS. 2A and 2B include a random access memory which may be a type MCM 6810 and designated by 61 in FIG. 2B. The computer also includes a number of programmable read only memories 64, 66, 67, 68 and 69 which might be type 1702A. A unit 71 connected to the unit 59 may be the type NE555. The unit 63 connected to the memory 64 through 69 may be type CD4028B/MC1402B. Components of the clock 56 indicated by 215 and 216 may be type 9602.

Attached and made a part hereof is the program for the computer so as to provide software for the computer such that it actuates the proper output indicators and solenoids to operate the pinball machine.

FIGS. 10 through 30 illustrate various flow charts for the program and subprograms of the computer.

FIG. 10 illustrates a subroutine program which reads serial data presented at peripheral interface adapter 58 and clocks 8 bits of data. PIA2CA2 is the shift register clock signal. After the start signal 200 is received the peripheral interface adapter 58 receives a clock signal in step 201. Steps 202 through 207 complete the flow chart.

FIG. 11 is a routine for writing a bit to a specified position in LMPMAT (word). Steps 208 through 214 define this program.

FIG. 12 illustrates a routine which writes a bit to M(ADDR) at bit position bit. TEMP holds the bit to be

written in position 0. Only the specified bit is affected, all other bits in M(ADOR) are unchanged, inputs are bit (07), ADDR (0-64K), and TEMP (0 or 1). This program is represented by blocks 215 through 223.

FIG. 13 illustrates a subroutine which writes the LMPMAT to the peripheral interface adapter 58. The output data is formatted for use by 8 channel demultiplexer. This subroutine is represented by blocks 224 through 233.

FIG. 14 is a subroutine for NXTPLY and is represented by blocks 234 through 244.

FIG. 15 illustrates a subroutine used for checking the credit when a coin is deposited in the machine.

FIG. 16 is a subroutine used to monitor coins and give appropriate credit.

FIG. 17 illustrates a subroutine for interrupt which is initiated by a 120 cycle per second signal which reads the 5 bit byte by 8 bite input matrix and processes the input data using EDGEDET.

FIG. 18 is a new game routine.

FIG. 19 illustrates a routine for collecting display of bonus.

FIG. 21 illustrates the routine which reads data from 7 by 8 input matrixes.

FIG. 22 illustrates the routine which identifies the active interrupt port and transfers control to an appropriate routine.

FIG. 23 illustrates the zero credit subroutine.

FIG. 24 illustrates the subroutine for scoring.

FIG. 25 is a subroutine for checking various values that have changed states.

FIG. 26 illustrates a routine for lighting bonus lights.

FIG. 27 is a bonus amount subroutine which is used to register the amount of bonus after a target is hit.

FIG. 28 is a subroutine for monitoring the target hits and scores accordingly.

FIG. 29 is a routine for monitoring the target hits. This routine scans each bit of the words jumping to a designated subroutine when a bit is set.

FIG. 30 illustrates a routine to determine free game threshold.

FIG. 31 is an alternative subroutine for monitoring coins and giving appropriate credit.

FIG. 32 is a routine to shift a specified bit to the carry flag position.

Although this invention has been described with respect to preferred embodiments, it is not to be so limited as changes and modifications may be made which are within the full intended scope as defined by the appended claims.

PROGRAM FOR MPU

OF INVENTION

		NAM	FOURTH	
		OPT	DB16, MEM, SYMBOL	
00100				
00110				
01001		* F R O N T		
01002		*RAM LABEL ASSIGNMENTS		
01005	005A	BEGSTK EQU	90	LOAD ADDR. FOR BEGIN STACK.
01010	0048	RESRAM EQU	BEGSTK-18	STRT.RESV.RAM AREA
01015		*STARTING AREA		
01035		*PIA LABEL ASSIGNMENTS		
01040	0034	PIAIDA EQU	\$84	
01045	0035	PIAICA EQU	PIAIDA+1	
01050	0086	PIA1DB EQU	PIAIDA+2	
01055	0087	PIA1CB EQU	PIAIDA+3	
01060	0088	PIA2DA EQU	\$88	
01065	0089	PIA2CA EQU	PIA2DA+1	
01070	008A	PIA2DB EQU	PIA2DA+2	
01075	008B	PIA2CB EQU	PIA2DA+3	
01079	0090	PIA3DA EQU	\$90	
01080	0091	PIA3CA EQU	PIA3DA+1	
01081	0092	PIA3DB EQU	PIA3DA+2	
01082	0093	PIA3CB EQU	PIA3DA+3	
01083		*ROM LABEL ASSIGNMENTS		
01085	0800	BEGROM EQU	\$0800	
01090		*LAMP SYMBOL ASSIGNMENTS		
01095	0000	DRLPA0 EQU	%00000000	A TARGET OFF
01100	0080	DRLPA1 EQU	%10000000	A TARGET ON
01105	0001	DRLPB0 EQU	%00000001	B TARGET OFF
01110	0081	DRLPB1 EQU	%10000001	B TARGET ON
01115	0002	DRLPC0 EQU	%00000010	C TARGET OFF
01120	0082	DRLPC1 EQU	%10000010	C TARGET ON
01125	0003	DRLPD0 EQU	%00000011	D TARGET OFF
01130	0083	DRLPD1 EQU	%10000011	D TARGET ON
01135	0004	RVLPA0 EQU	%00000100	A ROLLOVER OFF

4,198,051

7

8

01140	0084	RVLPA1 EQU	%10000100	A ROLLOVER ON
01145	0005	RVLPB0 EQU	%00000101	B ROLLOVER OFF
01150	0085	RVLPB1 EQU	%10000101	B ROLLOVER ON
01155	0006	RVLPC0 EQU	%00000110	C ROLLOVER OFF
01160	0086	RVLPC1 EQU	%10000110	C ROLLOVER ON
01165	0007	RVLPD0 EQU	%00000111	D ROLLOVER OFF
01170	0087	RVLPD1 EQU	%10000111	D ROLLOVER ON
01175	0010	BL01K0 EQU	%00010000	1K BONUS OFF
01180	0090	BL01K1 EQU	%10010000	1K BONUS ON
01185	0011	BL02K0 EQU	%00010001	2K BONUS OFF
01190	0091	BL02K1 EQU	%10010001	2K BONUS ON
01195	0012	BL03K0 EQU	%00010010	3K BONUS OFF
01200	0092	BL03K1 EQU	%10010010	3K BONUS ON
01205	0013	BL04K0 EQU	%00010011	4K BONUS OFF
01210	0093	BL04K1 EQU	%10010011	4K BONUS ON
01215	0014	BL05K0 EQU	%00010100	5K BONUS OFF
01220	0094	BL05K1 EQU	%10010100	5K BONUS ON
01225	0015	BL05K0 EQU	%00010101	6K BONUS OFF
01230	0095	BL06K1 EQU	%10010101	6K BONUS ON
01235	0016	BL07K0 EQU	%00010110	7K BONUS OFF
01240	0096	BL07K1 EQU	%10010110	7K BONUS ON
01245	0017	BL08K0 EQU	%00010111	8K BONUS OFF
01250	0097	BL08K1 EQU	%10010111	8K BONUS ON
01255	0020	BL09K0 EQU	%00100000	9K BONUS OFF
01260	00A0	BL09K1 EQU	%10100000	9K BONUS ON
01265	0021	BL10K0 EQU	%00100001	10K BONUS OFF
01270	00A1	BL10K1 EQU	%10100001	10K BONUS ON
01275	0022	DBLBNO EQU	%00100010	DOUBLE BONUS OFF
01280	00A2	DBLBN1 EQU	%10100010	DOUBLE BONUS ON
01285	0023	BL010 EQU	%00100011	INCREMENT BONUS 1 OFF
01290	00A3	BL011 EQU	%10100011	INCREMENT BONUS 1 ON
01295	0024	BL020 EQU	%00100100	INCREMENT BONUS 2 OFF
01300	00A4	BL021 EQU	%10100100	INCREMENT BONUS 2 ON
01305	0025	BL030 EQU	%00100101	INCREMENT BONUS 3 OFF
01310	00A5	BL031 EQU	%10100101	INCREMENT BONUS 3 ON
01315	0026	BL040 EQU	%00100110	INCREMENT BONUS 4 OFF
01320	00A6	BL041 EQU	%10100110	INCREMENT BONUS 4 ON
01325	0027	BL050 EQU	%00100111	INCREMENT BONUS 5 OFF
01330	00A7	BL051 EQU	%10100111	INCREMENT BONUS 5 ON
01335	0030	BL060 EQU	%00110000	INCREMENT BONUS 6 OFF
01340	00B0	BL061 EQU	%10110000	INCREMENT BONUS 6 ON
01345	0031	BL070 EQU	%00110001	INCREMENT BONUS 7 OFF
01350	00B1	BL071 EQU	%10110001	INCREMENT BONUS 7 ON
01355	0032	BL080 EQU	%00110010	INCREMENT BONUS 8 OFF
01360	00B2	BL081 EQU	%10110010	INCREMENT BONUS 8 ON
01365	0033	BL090 EQU	%00110011	INCREMENT BONUS 9 OFF
01370	00B3	BL091 EQU	%10110011	INCREMENT BONUS 9 ON
01375	0034	BL100 EQU	%00110100	INCREMENT BONUS 10 OFF
01380	00B4	BL101 EQU	%10110100	INCREMENT BONUS 10 ON
01385	0035	LXBLP0 EQU	%00110101	LT. EXTRA BALL OFF
01390	00B5	LXBLP1 EQU	%10110101	LT. EXTRA BALL ON
01395	0036	RXBLP0 EQU	%00110110	RT. EXTRA BALL OFF
01400	00B6	RXBLP1 EQU	%10110110	RT. EXTRA BALL ON
01405		* EQU %00110111		
01410		* EQU %10110111		
01415	0040	UP1LP0 EQU	%01000000	1 UP OFF
01420	00C0	UP1LP1 EQU	%11000000	1 UP ON
01425	0041	UP2LP0 EQU	%01000001	2 UP OFF
01430	00C1	UP2LP1 EQU	%11000001	2 UP ON
01435	0042	UP3LP0 EQU	%01000010	3 UP OFF
01440	00C2	UP3LP1 EQU	%11000010	3 UP ON
01445	0043	UP4LP0 EQU	%01000011	4 UP OFF
01450	00C3	UP4LP1 EQU	%11000011	4 UP ON
01455	0044	PLY1N0 EQU	%01000100	1 NO. OF PLAYERS OFF
01460	00C4	PLY1N1 EQU	%11000100	1 NO. OF PLAYERS ON
01465	0045	PLY2N0 EQU	%01000101	2 NO. OF PLAYERS OFF

4,198,051

9

10

01470	00C5	PLY2N1 EQU	%11000101	2 NO. OF PLAYERS ON
01475	0046	PLY3N0 EQU	%01000110	3 NO. OF PLAYERS OFF
01480	00C6	PLY3N1 EQU	%11000110	3 NO. OF PLAYERS ON
01485	0047	PLY4N0 EQU	%01000111	4 NO. OF PLAYERS OFF
01490	00C7	PLY4N1 EQU	%11000111	4 NO. OF PLAYERS ON
01495	0050	MTLP00 EQU	%01010000	00 MATCH OFF
01500	00D0	MTLP01 EQU	%11010000	00 MATCH ON
01505	0051	MTLP10 EQU	%01010001	10 MATCH OFF
01510	00D1	MTLP11 EQU	%11010001	10 MATCH ON
01515	0052	MTLP20 EQU	%01010010	20 MATCH OFF
01520	00D2	MTLP21 EQU	%11010010	20 MATCH ON
01525	0053	MTLP30 EQU	%01010011	30 MATCH OFF
01530	00D3	MTLP31 EQU	%11010011	30 MATCH ON
01535	0054	MTLP40 EQU	%01010100	40 MATCH OFF
01540	00D4	MTLP41 EQU	%11010100	40 MATCH ON
01545	0055	MTLP50 EQU	%01010101	50 MATCH OFF
01550	00D5	MTLP51 EQU	%11010101	50 MATCH ON
01555	0056	MTLP60 EQU	%01010110	60 MATCH OFF
01560	00D6	MTLP61 EQU	%11010110	60 MATCH ON
01565	0057	MTLP70 EQU	%01010111	70 MATCH OFF
01570	00D7	MTLP71 EQU	%11010111	70 MATCH ON
01575	0060	MTLP80 EQU	%01100000	80 MATCH OFF
01580	00E0	MTLP81 EQU	%11100000	80 MATCH ON
01585	0061	MTLP90 EQU	%01100001	90 MATCH OFF
01590	00E1	MTLP91 EQU	%11100001	90 MATCH ON
01595	0062	BIPL10 EQU	%01100010	1 BALL IN PLAY OFF
01600	00E2	BIPL11 EQU	%11100010	1 BALL IN PLAY ON
01605	0063	BIPL20 EQU	%01100011	2 BALL IN PLAY OFF
01610	00E3	BIPL21 EQU	%11100011	2 BALL IN PLAY ON
01615	0064	BIPL30 EQU	%01100100	3 BALL IN PLAY OFF
01620	00E4	BIPL31 EQU	%11100100	3 BALL IN PLAY ON
01625	0067	BIPL40 EQU	%01100111	4 BALL IN PLAY OFF
01630	00E5	BIPL41 EQU	%11100111	4 BALL IN PLAY ON
01635	0066	BIPL50 EQU	%01100110	5 BALL IN PLAY OFF
01640	00E6	BIPL51 EQU	%11100110	5 BALL IN PLAY ON
01645	0067	GNILL1 EQU	%01100111	
01650	00E7	GNILLO EQU	%11100111	
01655		* EQU %01110000		
01660		* EQU %11110000		
01665		* EQU %01110001		
01670		* EQU %11110001		
01675		* EQU %01110010		
01680		* EQU %11110010		
01685		* EQU %01110011		
01690		* EQU %11110011		
01695	0074	SPCLP0 EQU	%01110100	LT. AND RT. SPECIAL OFF
01700	00F4	SPSLP1 EQU	%11110100	LT. AND RT. SPECIAL ON
01705	0075	STAGN0 EQU	%01110101	SHOOT AGAIN OFF
01710	00F5	STAGN1 EQU	%11110101	SHOOT AGAIN ON
01715	0076	TLTLPG EQU	%01110110	TILT OFF
01720	00F6	TLTLP1 EQU	%11110110	TILT ON
01725	0077	GMOVRO EQU	%01110111	GAME OVER OFF
01730	00F7	GMOVRI EQU	%11110111	GAME OVER ON
01735		*RAM ALLOCATIONS		
01740	0000	ORG	00	
01741	0000 0002	ADDR2 RMB	2	
01745	0002 0002	ADDR RMB	2	BITRD,BITWRT
01751	0004 0001	CHIME RMB	1	
01753	0005 0002	SCRWRD RMB	2	
01755	0007 0001	XWD RMB	1	RDMAT
01756	0008 0001	QLOOP RMB	1	
01760	0009 0001	XST RMB	1	RDMAT
01762	000A 0001	CURENT RMB	1	
01763	000B 0001	DESI RD RMB	1	
01765	000C 0001	REPEAT RMB	1	RDMAT
01766	000D 0001	SEARCH RMB	1	

01757	000E	0001	REPLCE	RMB	1	
01768	000F	0001	DISPLY	RMB	1	
01770	0010	0001	PLYFLD	RMB	1	RDMAT
01771	0011	0002	ADDOFF	RMB	2	
01772	0013	0001	BMULT	RMB	1	
01775	0014	0001	PCOUNT	RMB	1	
01775	0015	0008	LMPMAT	RMB	8	WRTLMP, BITWRT
01780	001D	0001	TEMP	RMB	1	BITWRT
01790	001E	0002	ENTRY	RMB	2	TARMON
01795	0020	0001	ZCCNT	RMB	1	DELAY,ZEROCR
01800	0021	0001	BIP	RMB	1	MAIN
01805	0022	0001	NUP	RMB	1	MAIN
01810	0023	0001	NPLAY	RMB	1	
01815	0024	0001	BITNUM	RMB	1	
01820	0025	0003	BONUS	RMB	3	
01822	0028	0001	MATCH	RMB	1	
01825	0029	0003	ADDEND	RMB	3	
01835	002C	0001	TILT	RMB	1	
01840	002D	0001	OTHLFG	RMB	1	OUTHLM,MAIN
01845	002E	0001	WAIT	RMB	1	ZEROCR
01850	002F	0001	WAITCN	RMB	1	SEROCR
01880	0030	0001	CNCNT1	RMB	1	CNMON
01884	0031	0001	DECADD	RMB	1	
01885	0032	0001	CHCNT2	RMB	1	CNMON
01895	0033	0004	LSTXWD	RMB	4	
01897	0037	0004	LSTXST	RMB	4	
01899	003B	0007	XEDGE	RMB	7	
01905	0042	0003	SCORE1	RMB	3	
01906	0045	0003	SCORE2	RMB	3	
01907	0046	0003	SCORE3	RMB	3	
01908	004B	0003	SCORE4	RMB	3	
01909	004E	0001	CREDIT	RMB	1	
01910	004F	0001	DGTADD	RMB	1	
01911	0050	0001	MATCHD	RMB	1	
01912	0051	0002	XSTART	RMB	2	
01913	0053	0001	SHFTFG	RMB	1	
01920			*			
01925			*			
01930			*			
01935			*			
01940			*PIA INITIALIZATION			
01945	0800		ORG	BEGROM		
01950			*BEGIN ACTIVE PROGRAM, ENTERED FROM RESTART INT.			
01958	0800	86 7F	BEGROM	LDA A	##01111111	
01960	0802	97 84		STA A	PIA1DA	
01965	0804	73 0086		COM	PIA1DB	
01970	0807	73 0088		COM	PIA2DA	
01975	080A	73 008A		COM	PIA2DB	
01976	080D	73 0090		COM	PIA3DA	
01977	0810	73 0092		COM	PIA3DB	
01980			*			
01985	0813	86 2E		LDA A	##00101110	
01990	0815	97 85		STA A	PIA1CA	
01995	0817	86 36		LDA A	##00110110	
02000	0819	97 87		STA A	PIA1CB	
02005	081B	86 16		LDA A	##00010110	
02010	081D	97 89		STA A	PIA2CA	
02012	081F	86 0D		LDA A	##00001101	
02013	0821	97 8B		STA A	PIA2CB	
02014	0823	86 25		LDA A	##00100101	
02015	0825	97 91		STA A	PIA3CA	
02016	0827	86 24		LDA A	##00100100	
02017	0829	97 93		STA A	PIA3CB	
02015			*			
02027			*INIT STACK POINTER			
02028			*(SWI.Z.C.,DISPLAY,SLAM)			

```

02030 082B 8E 007F      LDS      #127
02035
02040
02042 082E FC           LDA A    #%11111100
02044 0830 97 86       STA A    PIA1DB
02045 0832 86 FF       LDA A    #$OFF
02046 0834 97 8A       STA A    PIA2DB
02050
02055
02056 0836 CE 007D     *CLEAR ALL RAM <=125
02057 0839 BD 0BD1     LDX      #125
02061
02062 083C 86 0E       JSR      RAMZR
02063 083E 97 52       *LOAD INITIAL CONDITIONS FOR NEWDOT
02064 0840 86 06       LDA A    #14
02065 0842 97 4F       STA A    XSTART+1
02066 0844 7F 0051     LDA A    #6
02067 0847 7F 0053     STA A    DGTADD
02070
02075 084A 86 10       CLR      XSTART
02078 084C 97 26       CLR      SHFTFG
02080 084E 86 01       *INITIALIZE BONUS REGISTERS & MULTIPLIER
02082 0850 97 13       LDA A    #$10
03001
03050
03065 0852 BD 0E7C     STA A    BONUS+1
03090
03095 0855 7F 0010     LDA A    #1
03100 0858 0E          STA A    BMULT
03103 0859 3E          * M A I N
03105
03106
03110
03115
03120
03125
03130 085A 7F 0010     *LIGHT BONUS LIGHTS
03137 085D C6 01       JSR      BONDSP
03139 085F BD 0C9E     *READ FIXED DATA
03140
03141 0862 86 01       CLR      PLYFLD
03142 0864 97 10       CLI
03143
03144 0866 86 FF       WAI
03145 0868 97 86
03147
03150 086A F7
03160 086C BD 0C7D     *WAIT FOR NEW GAME
03161
03162 086F 4F
03164 0870 97 23
03166 0872 97 22
03167 0874 97 21
03169 0876 BD 0EDC     *READ FIXED DATA FROM INPUT MATRIX TO XEDGE
03170
03175 0879 BD 0CA5     *4,5,6. (FREE GAME THRESHOLD, SELECT GAMES1&2
03180
03185
03190
03197 087C C6 07
03200 087E CE 001C     *SELECT COINS1&2, SELECT BALLS/GAME, AND MATCH)
03205 0881 BD 0D38
03210 0884 25 F3
03211
03212
03213
03214
03215
03216
03217
03218
03219
03220
03221
03222
03223
03224
03225
03226
03227
03228
03229
03230
03231
03232
03233
03234
03235
03236
03237
03238
03239
03240
03241
03242
03243
03244
03245
03246
03247
03248
03249
03250
03251
03252
03253
03254
03255
03256
03257
03258
03259
03260
03261
03262
03263
03264
03265
03266
03267
03268
03269
03270
03271
03272
03273
03274
03275
03276
03277
03278
03279
03280
03281
03282
03283
03284
03285
03286
03287
03288
03289
03290
03291
03292
03293
03294
03295
03296
03297
03298
03299
03300
03301
03302
03303
03304
03305
03306
03307
03308
03309
03310
03311
03312
03313
03314
03315
03316
03317
03318
03319
03320
03321
03322
03323
03324
03325
03326
03327
03328
03329
03330
03331
03332
03333
03334
03335
03336
03337
03338
03339
03340
03341
03342
03343
03344
03345
03346
03347
03348
03349
03350
03351
03352
03353
03354
03355
03356
03357
03358
03359
03360
03361
03362
03363
03364
03365
03366
03367
03368
03369
03370
03371
03372
03373
03374
03375
03376
03377
03378
03379
03380
03381
03382
03383
03384
03385
03386
03387
03388
03389
03390
03391
03392
03393
03394
03395
03396
03397
03398
03399
03400
03401
03402
03403
03404
03405
03406
03407
03408
03409
03410
03411
03412
03413
03414
03415
03416
03417
03418
03419
03420
03421
03422
03423
03424
03425
03426
03427
03428
03429
03430
03431
03432
03433
03434
03435
03436
03437
03438
03439
03440
03441
03442
03443
03444
03445
03446
03447
03448
03449
03450
03451
03452
03453
03454
03455
03456
03457
03458
03459
03460
03461
03462
03463
03464
03465
03466
03467
03468
03469
03470
03471
03472
03473
03474
03475
03476
03477
03478
03479
03480
03481
03482
03483
03484
03485
03486
03487
03488
03489
03490
03491
03492
03493
03494
03495
03496
03497
03498
03499
03500
03501
03502
03503
03504
03505
03506
03507
03508
03509
03510
03511
03512
03513
03514
03515
03516
03517
03518
03519
03520
03521
03522
03523
03524
03525
03526
03527
03528
03529
03530
03531
03532
03533
03534
03535
03536
03537
03538
03539
03540
03541
03542
03543
03544
03545
03546
03547
03548
03549
03550
03551
03552
03553
03554
03555
03556
03557
03558
03559
03560
03561
03562
03563
03564
03565
03566
03567
03568
03569
03570
03571
03572
03573
03574
03575
03576
03577
03578
03579
03580
03581
03582
03583
03584
03585
03586
03587
03588
03589
03590
03591
03592
03593
03594
03595
03596
03597
03598
03599
03600
03601
03602
03603
03604
03605
03606
03607
03608
03609
03610
03611
03612
03613
03614
03615
03616
03617
03618
03619
03620
03621
03622
03623
03624
03625
03626
03627
03628
03629
03630
03631
03632
03633
03634
03635
03636
03637
03638
03639
03640
03641
03642
03643
03644
03645
03646
03647
03648
03649
03650
03651
03652
03653
03654
03655
03656
03657
03658
03659
03660
03661
03662
03663
03664
03665
03666
03667
03668
03669
03670
03671
03672
03673
03674
03675
03676
03677
03678
03679
03680
03681
03682
03683
03684
03685
03686
03687
03688
03689
03690
03691
03692
03693
03694
03695
03696
03697
03698
03699
03700
03701
03702
03703
03704
03705
03706
03707
03708
03709
03710
03711
03712
03713
03714
03715
03716
03717
03718
03719
03720
03721
03722
03723
03724
03725
03726
03727
03728
03729
03730
03731
03732
03733
03734
03735
03736
03737
03738
03739
03740
03741
03742
03743
03744
03745
03746
03747
03748
03749
03750
03751
03752
03753
03754
03755
03756
03757
03758
03759
03760
03761
03762
03763
03764
03765
03766
03767
03768
03769
03770
03771
03772
03773
03774
03775
03776
03777
03778
03779
03780
03781
03782
03783
03784
03785
03786
03787
03788
03789
03790
03791
03792
03793
03794
03795
03796
03797
03798
03799
03800
03801
03802
03803
03804
03805
03806
03807
03808
03809
03810
03811
03812
03813
03814
03815
03816
03817
03818
03819
03820
03821
03822
03823
03824
03825
03826
03827
03828
03829
03830
03831
03832
03833
03834
03835
03836
03837
03838
03839
03840
03841
03842
03843
03844
03845
03846
03847
03848
03849
03850
03851
03852
03853
03854
03855
03856
03857
03858
03859
03860
03861
03862
03863
03864
03865
03866
03867
03868
03869
03870
03871
03872
03873
03874
03875
03876
03877
03878
03879
03880
03881
03882
03883
03884
03885
03886
03887
03888
03889
03890
03891
03892
03893
03894
03895
03896
03897
03898
03899
03900
03901
03902
03903
03904
03905
03906
03907
03908
03909
03910
03911
03912
03913
03914
03915
03916
03917
03918
03919
03920
03921
03922
03923
03924
03925
03926
03927
03928
03929
03930
03931
03932
03933
03934
03935
03936
03937
03938
03939
03940
03941
03942
03943
03944
03945
03946
03947
03948
03949
03950
03951
03952
03953
03954
03955
03956
03957
03958
03959
03960
03961
03962
03963
03964
03965
03966
03967
03968
03969
03970
03971
03972
03973
03974
03975
03976
03977
03978
03979
03980
03981
03982
03983
03984
03985
03986
03987
03988
03989
03990
03991
03992
03993
03994
03995
03996
03997
03998
03999
04000
04001
04002
04003
04004
04005
04006
04007
04008
04009
04010
04011
04012
04013
04014
04015
04016
04017
04018
04019
04020
04021
04022
04023
04024
04025
04026
04027
04028
04029
04030
04031
04032
04033
04034
04035
04036
04037
04038
04039
04040
04041
04042
04043
04044
04045
04046
04047
04048
04049
04050
04051
04052
04053
04054
04055
04056
04057
04058
04059
04060
04061
04062
04063
04064
04065
04066
04067
04068
04069
04070
04071
04072
04073
04074
04075
04076
04077
04078
04079
04080
04081
04082
04083
04084
04085
04086
04087
04088
04089
04090
04091
04092
04093
04094
04095
04096
04097
04098
04099
04100
04101
04102
04103
04104
04105
04106
04107
04108
04109
04110
04111
04112
04113
04114
04115
04116
04117
04118
04119
04120
04121
04122
04123
04124
04125
04126
04127
04128
04129
04130
04131
04132
04133
04134
04135
04136
04137
04138
04139
04140
04141
04142
04143
04144
04145
04146
04147
04148
04149
04150
04151
04152
04153
04154
04155
04156
04157
04158
04159
04160
04161
04162
04163
04164
04165
04166
04167
04168
04169
04170
04171
04172
04173
04174
04175
04176
04177
04178
04179
04180
04181
04182
04183
04184
04185
04186
04187
04188
04189
04190
04191
04192
04193
04194
04195
04196
04197
04198
04199
04200
04201
04202
04203
04204
04205
04206
04207
04208
04209
04210
04211
04212
04213
04214
04215
04216
04217
04218
04219
04220
04221
04222
04223
04224
04225
04226
04227
04228
04229
04230
04231
04232
04233
04234
04235
04236
04237
04238
04239
04240
04241
04242
04243
04244
04245
04246
04247
04248
04249
04250
04251
04252
04253
04254
04255
04256
04257
04258
04259
04260
04261
04262
04263
04264
04265
04266
04267
04268
04269
04270
04271
04272
04273
04274
04275
04276
04277
04278
04279
04280
04281
04282
04283
04284
04285
04286
04287
04288
04289
04290
04291
04292
04293
04294
04295
04296
04297
04298
04299
04300
04301
04302
04303
04304
04305
04306
04307
04308
04309
04310
04311
04312
04313
04314
04315
04316
04317
04318
04319
04320
04321
04322
04323
04324
04325
04326
04327
04328
04329
04330
04331
04332
04333
04334
04335
04336
04337
04338
04339
04340
04341
04342
04343
04344
04345
04346
04347
04348
04349
04350
04351
04352
04353
04354
04355
04356
04357
04358
04359
04360
04361
04362
04363
04364
04365
04366
04367
04368
04369
04370
04371
04372
04373
04374
04375
04376
04377
04378
04379
04380
04381
04382
04383
04384
04385
04386
04387
04388
04389
04390
04391
04392
04393
04394
04395
04396
04397
04398
04399
04400
04401
04402
04403
04404
04405
04406
04407
04408
04409
04410
04411
04412
04413
04414
04415
04416
04417
04418
04419
04420
04421
04422
04423
04424
04425
04426
04427
04428
04429
04430
04431
04432
04433
04434
04435
04436
04437
04438
04439
04440
04441
04442
04443
04444
04445
04446
04447
04448
04449
04450
04451
04452
04453
04454
04455
04456
04457
04458
04459
04460
04461
04462
04463
04464
04465
04466
04467
04468
04469
04470
04471
04472
04473
04474
04475
04476
04477
04478
04479
04480
04481
04482
04483
04484
04485
04486
04487
04488
04489
04490
04491
04492
04493
04494
04495
04496
04497
04498
04499
04500
04501
04502
04503
04504
04505
04506
04507
04508
04509
04510
04511
04512
04513
04514
04515
04516
04517
04518
04519
04520
04521
04522
04523
04524
04525
04526
04527
04528
04529
04530
04531
04532
04533
04534
04535
04536
04537
04538
04539
04540
04541
04542
04543
04544
04545
04546
04547
04548
04549
04550
04551
04552
04553
04554
04555
04556
04557
04558
04559
04560
04561
04562
04563
04564
04565
04566
04567
04568
04569
04570
04571
04572
04573
04574
04575
04576
04577
04578
04579
04580
04581
04582
04583
04584
04585
04586
04587
04588
04589
04590
04591
04592
04593
04594
04595
04596
04597
04598
04599
04600
04601
04602
04603
04604
04605
04606
04607
04608
04609
04610
04611
04612
04613
04614
04615
04616
04617
04618
04619
04620
04621
04622
04623
04624
04625
04626
04627
04628
04629
04630
04631
04632
04633
04634
04635
04636
04637
04638
04639
04640
04641
04642
04643
04644
04645
04646
04647
04648
04649
04650
04651
04652
04653
04654
04655
04656
04657
04658
04659
04660
04661
04662
04663
04664
04665
04666
04667
04668
04669
04670
04671
04672
04673
04674
04675
04676
04677
04678
04679
04680
04681
04682
04683
04684
04685
04686
04687
04688
04689
04690
04691
04692
04693
04694
04695
04696
04697
04698
04699
04700
04701
04702
04703
04704
04705
04706
04707
04708
04709
04710
04711
04712
04713
04714
04715
04716
04717
04718
04719
04720
04721
04722
04723
04724
04725
04726
04727
04728
04729
04730
04731
04732
04733
04734
04735
04736
04737
04738
04739
04740
04741
04742
04743
04744
04745
04746
04747
04748
04749
04750
04751
04752
04753
04754
04755
04756
04757
04758
04759
04760
04761
04762
04763
04764
04765
04766
04767
04768
04769
04770
04771
04772
04773
04774
04775
04776
04777
04778
04779
04780
04781
04782
04783
04784
04785
04786
04787
04788
04789
04790
04791
04792
04793
04794
04795
04796
04797
04798
04799
04800
04801
04802
04803
04804
04805
04806
04807
04808
04809
04810
04811
04812
04813
04814
04815
04816
04817
04818
04819
04820
04821
04822
04823
04824
04825
04826
04827
04828
04829
04830
04831
04832
04833
04834
04835
04836
04837
04838
04839
04840
04841
04842
04843
04844
04845
04846
04847
04848
04849
04850
04851
04852
04853
04854
04855
04856
04857
04858
04859
04860
04861
04862
04863
04864
04865
04866
04867
04868
04869
04870
04871
04872
04873
04874
04875
04876
04877
04878
04879
04880
04881
04882
04883
04884
04885
04886
04887
04888
04889
04890
04891
04892
04893
04894
04895
04896
04897
04898
04899
04900
04901
04902
04903
04904
04905
04906
04907
04908
04909
04910
04911
04912
04913
04914
04915
04916
04917
04918
04919
04920
04921
04922
04923
04924
04925
04926
04927
04928
04929
04930
04931
04932
04933
04934
04935
04936
04937
04938
04939
04940
04941
04942
04943
04944
04945
04946
04947
04948
04949
04950
04951
04952
04953
04954
04955
04956
04957
04958
04959
04960
04961
04962
04963
04964
04965
04966
04967
04968
04969
04970
04971
04972
04973
04974
04975
04976
04977
04978
04979
04980
04981
04982
04983
04984
04985
04986
04987
04988
04989
04990
04991
04992
04993
04994
04995
04996
04997
04998
04999
05000
05001
05002
05003
05004
05005
05006
05007
05008
05009
05010
05011
05012
05013
05014
05015
05016
05017
05018
05019
05020
05021
05022
05023
05024
05025
05026
05027
05028
05029
05030
05031
05032
05033
05034
05035
05036
05037
05038
05039
05040
05041
05042
05043
05044
05045
05046
05047
05048
05049
05050
05051
05052
05053
05054
05055
05056
05057
05058
05059
05060
05061
05062
05063
05064
05065
05066
05067
05068
05069
05070
05071
05072
05073
05074
05075
05076
0507
```



```

03215 0886 CF 000C      LDX      #12
03217 0889 6F 41 NOSCR  CLR      SCORE 1-1,X
03219 088B 09          DEX
03220 088C 26 FB      BNE      NOSCR
03221 088E
03222 088F BD 0F00      JSR      TASK4SUPPR. LEADING ZEROES
03223 0892 86 FF      LDA A    #$FF
03224 0894 97 50      STA A    MATCHD
03240
03245 0896 7C 0021 *NEXT BALL NEWBAL INC
03250 0899 7F 0022      CLR      NUP
03255
03260 089C 7C 0022 *NEXT PLAYER NEWPYR INC NUP
03266 *DISPLAY NO. OF PLAYERS, N). UP, AND BALL IN PLAY
03267 089F BD 0EDC      JSR      LIGHTS
03268 *ENTRY FOR SHOOT AGAIN FEATURE,
03269 08A2 BD 0BD9      SHTAG JSR      NXTPLY
03270 *DELAY .25E SECOND
03275 08AK C6 1E      EJECT LDA B    #30
03280 08A7 BD 0C9E      JSR      DELAY
03285
03290 *PULSE BALL EJECT SOLENOID
03295 08AA CE 0086      LDX      #PIA1DB LOAD ADDRESS AND BIT LOCAT
03300 08AD 86 06      LDA A    #6
03305 08AF BD 0AD0      JSR      SLNPLS
03310 *RESET SHOOT AGAIN LMP.
03315 08B2 86 75      LDA A    #STACNO
03320 08B4 BD 0C7D      JSR      LPCTRL
03335 *RESET OUTHL FLAG P BIT AFTER .5 SEC DELAY
03340 08B7 C6 3F      LDA B    #63
03341 08B0 BD 0C9E      JSR      DELAY
03342 08BC 7F 003D      CLR      XEDGE+2
03343 08BF 7F 002D      CLR      OTHLFG
03345
03345 *MONITOR COINS AND RESET
03350 08C2 BD 0CA5      MONMOD JSR      CNMON
03355
03356
03357
03358 0805 01 0101
03360 *MONITOR PLAYFIELD HITS AND SCORE ACCORDINGLY
03365 08C8 BD 0A8E      JSR      TARMON
03370 *TEST FOR BALL EXIT AT OUTHOLE(BIT 7-0) SET
03380 08CB 7D 002D      LDA A    OTHLFG
03385 08CE 27 F2      BMI      MONMOD IF NO EXIT. GO MONMOD
03390
03440
03445 *THIS PLAYERS TURN COMPLETE. COLLECT BONUS
0345 08D0 BD 0E20      JSR      BONES
03452 *TEST FOR SHOOT AGAIN
03454
03454 08D3 CE 001A      LDX      #LMPMAT+5 GET SHOOT AGAIN LMP.BIT
03455 08D6 C6 07      LDA B    #7
03456 08D8 BD 0D38      JSR      BITRD
03457 08DB 25 C5      BCS      SHTAG GO TO SHOOT AGAIN ENTRY
03490 *NEXT PLAYER OR NEXT BALL?
03495 08DD 96 22      BYPSHT LDA A    NUP
03500 08DF 91 23      CMP A    NPLAY
03505 08E1 26 B9      BNE      NEWPYR GO TO NEXT PLAYER
03515
03520 *ALL PLAYERS HAVE COMPLETED THIS BALL.TEST
03525 *FOR REMAINING BALLS IN GAME.
03530 08E3 86 C0      LDA A    #%11000000 LOAD THE MAX BALLS CONT
03535 08E5 94 40      AND A    XEDGE+5

```

```

03540 08E7 49      ROL A
03545 08E8 49      ROL A
03550 08E9 49      ROL A
03555 08EA 8B 03    ADD A      #3 CONVERT CONTROL BIT TO MAX.
03560 08EC 91 21    CMP A      BIP
03565 08EE 22 A6    BHI        NEWBAL GO TO NEXT BALL
03570                *NO REMAINING BALLS GAME IS OVER
03575                *TEST ALL SCORES FOR MATCH AND FREE GAME.
03577                *GIVE APPROPRIATE CREDITS.
03580 08F0 BD 0COA    JSR        MATCHK
03585                *BRANCH TO NEW GAME
03590 08F3 7E 085A    JMP        NEWGAM
03595                *
03600                *
03605                *
03610                * ENTERS UBRoutines
05000                * PLY SUB

05005                *NAM TOPHOL
05010                *THIS ROUTINE INCREMENTS CURRENT PLAYERS SCORE
05015                *BY PULSES BALL EJECT SOLENOID AFTER 300MS DELAY
05020                *AND DISABLES THE TOP HOLE SENSOR FOR 0.5SEC
05025                *AFTER BAL
05030                *
05035                *TEST FLAG WAIT FOR 0.5 SEC DISABLE CONDITION
05040                *AND SKIP ROUTINE IF WAIT=1.
04034 08F6 7D 002E    TOPHOL TST      WAIT
05050 08F9 26 13      BNE        BYPSHL
05055                *INCREMENT SCORE BY 3000
05060 08FB 86 83      LDA A      #$83
05070 08FD BD 0DDC    JSR        SCOREX
05075                *WAIT 300MS FOR BALL TO SETTLE INTO POSITION
05080 0900 C6 24      LDA B      #36
05085 0902 BD 0C9E    JSR        DELAY
05090                *PULSE TOP HOLE EJECT SOLENOID FOR 25MS
05095 0905 CE 0086    LDX        #PIA1DB LOAD ADDRESS AND BIT LOCAT
05100 0908 86 05      LDA A      #5
05105 090A BD 0ADO    JSR        SLNPLS
05110                *DISABLE TOP HOLE SENSOR FOR 0.5SEC. INITIALIZE
05115                *WAIT (DECREMENTED BY Z.C.)
05120 090D 43      LDA A      #60
05125 090E 39      STA A      WAIT
05135                *NAM THUBPR
05140                *THIS ROUTINE IS CALLED BY TARMON WHEN THE
05145                *THUMPER BUMPER SENSOR IS HIT. SCORE IS INCREASED
05150                *BY 100 A THE SOLENOID PULSED FOR 25MS.
05155                *THE ROUTINE IS BYPASSED IF
05160                *TILT=1
05165 090F 7D 002C    THUBPR TST      TILT
05170 0912 26 0D      BNE        ENDTH
05175                *
05180                *
05185                *PULSE SOLENOID
05186                LDX        #PIA1D8 LOAD ADDRESS AND BIT LOCAT
05186 0917
05188 0919
05189                *INCREASE SCORE BY 100
05190 091C 86 41      LDA A      #$41
05200 091E BD 0DDC    JSR        SCOREX
05221 0921 39      ENDTH RTS

```

```

05223      *NAM RTILT
05224      *THIS ROUTINE SETS TILT LIGHT AND FLAG AND
05225      *INHIBITS FLIPPER ACTION. ROUTINE INHIBITS
05226      *SCORING AND BONUS COLLECTION, THUMPER BUMPED
05227      *AND SLINGSHOT ACTION THRU THE TILT FLAG.
05228 0922 86 01  RTILT LDA A      #1 SET TILT FLAG
05229 0924 97 2C      STA A      TILT
05230 0926 86 F6      LDA A      #TILTP1
05231 0928 BD 0C7D    JSR        LPCTRL LIGHT TILT LAMP.
05232 092B 86 FF      LDA A      #$OFF
05233 092D 97 86      STA A      PIA1DB DISABLE FLIPPERS.
05234 092F 39      RTS

```

```

05236      *NAM RLVSX
05240      *THIS ROUTINE IS CALLED BY THE ROLLOVER SWITCH
05245      *A,B D SUBROUTINE. IT ACCEPTS A SWITCH
05250      *IDENTIFIER (A,B D), IN THE FORM OF A WORD
05255      *LOCATION FOR THE CORRES A,B,C,D ROLLOVER
05260      *LAMP BIT. THE ADDR. IS IN X FROM CALLING
05265      *PROGRAM. INCREMENTS THE SCORE BY 1000. BO
05270      *IS INCREMENTED BY 1000 IF THE CORRESPONDING
      *ROLLO IS LIT.
      *
      *
05275      *INCREMENT BONUS BY 1000 IF CORRESPONDING ROLL
05280      *OVER LAMP IS LIT. ADDRESS OF WORD HOLDING LAMP
05285      BIY IN X
05290
05295 0930 5F      RLVSX CLR B      LOAD BIT LOCATION AND GET B
05300 0931 BD 0D38    JSR        BITRD
05305 0934 24 07      BCC        PYPSWX BRANCH IF NO BONUS
05310 0936 86 10      LDA A      #00010000 INCREMENT BONUS
05315 0938 97 2A      STA A      ADDEND+1
05320 093A BD 0 B8A    JSR        RBONUS
05325      *INCREMENT SCORE BY 1000
05330 093D 86 81      BYPSWX LDA A  #$81
05340 093F BD 0DDC    JSR        SCOREX
05345 0942 39      RTS

```

```

05355      *NAM RLVSX
05360      *THIS ROUTINE IS ACTIVATED BY THE D ROLLOVER BUTT
05365      *THE ROUTINE LOADS X WITH THE WORD LOCATION OF ROL
05370      *OVER SWITCH LAMP D AND CALLS RLVSX (SUPPLIES
05375      *THE BIT LOCATION) TO INCREMENT SCORE BY 1000 AND
05380      *ALSO BONUS IF CORRESPONDING LAMP IS LIT.
05385      *
05390      *
05395 0943 CE 001C    RLVSX LDS      #LMPMAT+7 WORD LOCATION OF ROLLOVE
05400      *SW. LMP. D.
05405 0946 20 E8      BRA        RLVSX

```

```

05420      *NAM RLVSX
05425      *THIS ROUTINE IS ACTIVATED BY THE C ROLLOVER BUTT
05430      *THE ROUTINE LOADS X WITH THE WORD LOCATION OF ROL
05435      *OVER SWITCH LAMP C AND CALLS RLVSX (SUPPLIES
05440      *THE BIT LOCATION) TO INCREMENT SCORE BY 1000 AND

```

```

05445      *ALSO BONUS IF CORRESPONDING LAMP IS LIT.
05450      *
05455      *
05460 0948 CE 001B RLVSWC LDX      #LMPMAT+6 WORD LOCATION OF ROLLOVE
05465      *                      SW. LMP. C.
05470 094B 20 E3      BRA      RLVSWX

05485      *NAM RLVSWB
05490      *THIS ROUTINE IS ACTIVATED BY THE B ROLLOVER BUTT
05495      *THE ROUTINE LOADS X WITH THE WORD LOCATION OF ROL
05500      *OVER SWITCH LAMP B AND CALLS RLVSWX (SUPPLIES
05505      *THE BIT LOCATION) TO INCREMENT SCORE BY 1000 AND
05510      *ALSO BONUS IF CORRESPONDING LAMP IS LIT.
05515      *
05520      *
05525 094D CE 001A. RLVSWB LDX      #LMPMAT+5 WORD LOCATION OF ROLLOVE
05530      *                      SW. LMP. B.
05535 0950 20 DE      BRA      RLVSWX
0550      *NAM RLVSWA
05555      *THIS ROUTINE IS ACTIVATED BY THE A ROLLOVER BUTT
05560      *THE ROUTINE LOADS X WITH THE WORD LOCATION OF ROL
05565      *OVER SWITCH LAMP A AND CALLS RLVSWX (SUPPLIES
05570      *THE BIT LOCATION) TO INCREMENT SCORE BY 1000 AND
05575      *ALSO BONUS IF CORRESPONDING LAMP IS LIT.
05580      *
05585      *
05590 0952 CE 0019 RLVSWA LDX      #LMPMAT+4 WORD LOCATION OF
05595      *                      ROLLOVER SW. LMP.A.
05600 0955 20 D9      BRA      RLVSWX

05615      *NAM RLVSW
05620      *THIS ROUTINE IS ACTIVATED BY THE FOUR 1K ROLLOVER
05625      *SWITCHES, NOT INCLUDING THOSE ASSOCIATED WITH THE
05630      *A,B,C & D 1K DROP TARGETS. THE ROUTINE FORMS
05635      *SCORE(NUP)=SCORE(NUP)+1000 AND BONUS=BONUS+1000.
05640      *THE ROUTINE IS ALSO ACTIVATED BY THE 1K CHANNEL
05645      *SWITCHES BECAUSE THEY HAVE THE SAME SCORING
05650      *REQUIREMENTS.
05655      *ADD 1000 TO SCORE
05660 0957 86 81 RLVSW LDA A      #$81
05670 0959 BD ODDC      JSR      SCOREX
05675      *INCREMENTS BONUS BY 1000
05680 095C 86 10      LDA A      #%00010000 (10 BCD)
05685 095E 97 2A      STA A      ADDEND+1
05690 0960 BD 0B8A      JSR      RBONUS
05695 0963 39      RTS

05705      *NAM REBOUN
05710      *THIS ROUTINE IS CALLED BY TARMON WHEN THE REBOUND
05715      *SENSOR IS HIT. THE ROUTINE INCREASES SCORE AND
05720      *MATCH BY 10. MATCH CONSISTS OF TWO HALF BYTES FOR
05725      *THE UNITS AND TENS POSITIONS. THE REGISTER IS INCR
05730      *BY 10 FROM 0 TO 90 WITH NO CARRIES GENERATED TO

```

```

05735      *A FOLLOWING BYTE.
05740      *
05745      *
05750      *INCREMENT SCORE BY 10
05755 0964 86 01 REBOUN LDA A    #$01
05765 0966 BD ODDC      JSR      SCOREX
05770      *ADD 10 TO MATCH (BCD)
05775 0969 86 10      LDA A    #%00010000
05780 096B 9B 28      ADD A    MATCH
05785 096D 19      DAA
05786 096E 97 28      STA A    MATCH
05790 0970 39      RTS
05800      *NAM SPECIAL
05805      *THIS ROUTINE IS CALLED BY TARMON WHEN THE
05810      *SPECIAL IS HIT.THE ROUTINEINCREMENTS CREDIT
      *IF BONUS=>
05815      *
05820      *
05825      *CHECK FOR BONUS=>15,000.TEST SPECIAL LAMP
05830 0971 CE 0019 SPECIAL LDX #LMPMAT+4  LOAD ADDRESS & BIT LOCAT
05835 0974 C6 07      LDA B #7
05840 0976 BD 0D38      JSR      BITRD
05845 0979 24 0B      BCC      BYPSPL BRANCH IS SPECIAL LAMP=0
05850 097B 7C 004E      LDA A    #1
      ADD A    CREDIT
      DAA
      STA A    CREDIT
05851      *OPERATE KNOCKER
05852 097E CE 0086      LDX      #PIA1DB
05853 0981 86 07      LDA A    #7
05854 0983 BD 0A00      JSR      SLNPLS
05855 0986 39      *BYPSPS RTS
05865      *NAM RSLGST
05870      *THIS ROUTINE IS CALLED BY TARMON WHEN THE RIGHT
05875      *SLING SHOT SENSOR IS HIT.THE ROUTINE IS
05880      *BYPASSED IF TILT=1.
05885      *
05890      *
05895      *BYPASS IF TILT=1
05900 0987 7D 002C RSLGST TST      TILT
05905 098A 26 0D      BNE      ENDRSL
05908      *PULSE RIGHT SLINGSHOT SOLENOID
05909 098C      LDX      #PIA1DB  LOAD ADDR. P BIT LOCATION
05910 098F      LDX A    #2
05911 0991      JSR      SLNPLS
05912      *ADD 10 TO SCORE
05915 0094 86 01      LDA A    #$01
05925 0996 BD ODDC      JSR      SCOREX
05950 0999 39      ENDRSL RTS

05960      *NAM LSLGST
05965      *THIS ROUTINE IS CALLED BY TARMON WHEN THE LEFT
05970      *SLINGSHOT SENSOR IS HIT.
05975      *
05980      *
05985      *BYPASS IF TILT=1
05990 099A 7D 002C LSLGST TST      TILT
05995 099D 26 0D      BNE      ENDSL
05998      *PULSE LEFT SLINGSHOT SOLENOID
05999 099F      LDX      #PIA1DB  LOAD ADDR. P BIT LOCATION
06000 09A2 86 03      LDA A    #3
06001 09A4      JSR      SLNPLS
06002      *ADD 10 TO SCORE
06005 09A7 86 01      LDA A    #$01

```

```

06015 09A9 BD ODDC      JSR      SCOREX
06040 09AC 39          ENDSL  RTS

```

```

06050          *NAM OUTHL
06055          *THIS ROUTINE IS CALLED BY TARMON WHEN THE OUTHOLE
06060          *SENSOR IS HIT. THE FLAG OTHLFG, BIT7, IS SET TO ZERO
06065          *TO INDICATE TO THE MAIN PROGRAM THAT THE BALL
          *HAS EXITED.
06070          *
06075          *
06080 09AD 86 01  OUTHL  LDA  A      OTHLFG
          AND  A      #$7F
06085 09AF 97 2D          STA  A      OTHLFG THE FLAG IS REST UPON ENTE
06090 09B1 39          RTS

```

```

06100          *NAM CHANIK
06105          *THIS ROUTINE IS CALLED BY TARMON WHEN THE 1K CHAN
06110          *SWITCH IS HIT. INCREMENTS SCORE AND BONUS BY 1000
06115          *CALL RLVSW WHICH HAS THE SAME SCORING REQUIREMENT
06120 09B2 7E 0957 CHANIK JMP      RLVSW

```

```

06135          *NAM CHANNX
06140          *THIS ROUTINE IS EXECUTED WHENEVER THE LEFT
06145          *OR RIGHT 500 CHANNELS ARE HIT. THE ENTRY POINT
06150          *IS DEPENDENT UPON WHETHER THE LEFT OR RIGHT
06155          *CHANNEL IS HIT. THIS ROUTINE SETS THE SHOOT
06160          *AGAIN LAMP WHEN THE LEFT EXTRA BALL LAMP IS
06165          *SET AND THE LEFT CHANNEL IS HIT OR WHEN THE
06170          *RIGHT EXTRA BALL LAMP IS SET AND THE RIGHT
06175          *CHANNEL IS HIT. SCORE IS INCREMENTED BY 500
06180          *WHENEVER EITHER CHANNEL IS HIT.
06185          *
06190          *
06195          *LOAD ADDRESS OF RIGHT EXTRA BALL LAMP.
06200 09B5 CE 001B CHANNR LDX      #LMPMAT+6
06205 09B8 DF 02          STX      ADDR
06210 09BA 20 03          BRA      CHANNX
06215          *LOAD ADDRESS OF LEFT EXTRA BALL LAMP.
06220 09BC CE 001A CHANNL LDX      #LMPMAT+5
06225          *LOAD BIT POSITION OF RIGHT EXTRA BALL
06230          *LAMP=LEFT
06235 09BF C6 03  CHANNX LDA  B      #3
06240          *TEST SELECTED EXTRA BALL LAMP, BIT, IF 1
06245          *THEN LIGHT SHOOT AGAIN LAMP.
06250 09C1 BD 0D38          JSR      BITRD
06255 09C4 24 05          BCC      BYPSET
06260 09C6 86 F5          LDA  A      #STAGN1  LIGHT SHOTT AGAIN LIGHT, RE
06265          *          AFTER BALL IS DELIVERED TO
06270          *          SHOOTER TIP.
06280 09C8 BD 0C7D          JSR      LPCTRL
06285          *ADD 500 TO SCORE
06290 09CB 86 45  BYPSET LDA  A      #$45
06300 09CD BD ODDC          JSR      SCOREX
06305 09D0 39          RTS
06315          *NAM DROPX
06320          * THIS ROUTINE IS ACTIVATED THRU TARMON BY
06325          *A HIT ON ANY OF THE A,B,C,OR D DROP TARGET

```

```

06330      *SWITCHES. THE ROUTINE ACTIVATES THE CORRES.
06335      *ROLLOVER LAMP AND RESETS THE ASSOC.
06340      *DROP TARGET LAMP. ADDITIONALLY, THE ROUTINE
06345      *ADDS 1000 TO THE SCORE AND CHECKS FOR THE
06350      *DOUBLE BONUS CONDITION(ALL FOUR DROP TARG.
06355      *HIT). IF DOUBLE BONUS EXISTSTHE DOUBLE
06360      *BONUS LAMP(DBLBN) AND DOUBLE BONUS FLAG(BMULT)
06365      *ARE SET.
06370      *   THE ROUTINE IS ENTERED AT THE POINT
06375      *CORRES. TO THE TARGET HIT.
06380      *
06385      *LOAD DATA FOR DROP TARGET AND ROLLOVER LAMPS.
06390 09D1 86 00  DROPX LDA A   #DRLPA0
06395 09D3 C6 84      LDA B   #RVLPA1
06400 09D5 20 10      BRA      DROPX
06405 09D7 86 01  DROPB LDA A   #DRLPBO
06410 09D9 C6 85      LDA B   #RVLPB1
06415 09DB 20 0A      BRA      DROPX
06420 09DD 86 02  DROPX LDA A   #DRLPC0
06425 09DF C6 86      LDA B   #RVLPC1
06430 09E1 20 04      BRA      DROPX
06435 09E3 86 03  DROPD LDA A   #DRLPDO
06440 09E5 C6 87      LDA B   #RVLPD1
06445      *CHANGE LAMPS PER ABOVE DATL.
06450 09E7 BD 0C7D  DROPX JSR      LPCTRL
06455 09EA 17      TBA
06465 09EB BD 0C7D  JSR      LPCTRL
06470      *ADD 1000 TO SCORE
06475 09EE 86 81      LDA A   #$81
06485 09F0 BD 0DDC  JSR      SCOREX
06490      * ACTIVATE DOUBLE BONUS LAMP IF ALL FOUR DROP TARG
06495      * ARE HIT. RESET BY BONES,LOAD FOUR DROP TARGET
06500      * LAMP BITS INTO A.
06505 09F3 CE 0018  AGAIN LDX      #LMPMAT+3   LOAD LAMP ADDRESS AND BI
06510 09F6 5F      CLR B
06515 09F7 BD 0D38  JSR      BITRD   GET SELECTED BIT, ROTATE IN
06520 09FA 46      ROR A
06525 09FB 09      DEX          INC. LAMPMAT ADDR. TO GET N
06530 09FC 8C 0014  CPX      #LMPMAT-1   BRANCH UNTIL 4 BITS LOAD
06535 09FF 26 F5      BNE      AGAIN
06540 0A01 4D      TST A          ARE ALL 4 BITS ZERO?
06545 0A02 26 09      BNE      BYPDA
06550 0A04 86 A2      LDA A   #DBLBN1   SET DOUBLE BONUS LAMP ON.
06560 0A06 BD 0C7D  JSR      LPCTRL
06561 0A09 86 02      LDA A   #$02     SET BONUS MULT
06562 0A0B 97 13      STA A   BMULT
06565 0A0D 39      BYPDA RTS
06830      *NAM DRP500
06835      *THIS ROUTINE IS CALLED BY TARMON WHEN THE 500 POIN
06840      *DROP TARGETS ARE HIT.THE ROUTINE INCREMENTS THE
06845      *SCORE BY 500
06850      *
06855      *
06860      *INCREMENT SCORE BY 500
06865 0A0E 86 45  DRP500 LDA A   #$45
06875 0A10 BD 0DDC  JSR      SCOREX
06880 0A13 39      RTS

06890      *NAM DRP100
06895      *THIS ROUTINE IS CALLED BY TARMON WHEN THE 100 DRO
06900      *TARGETS ARE HIT.
06905      *

```

```

06910
06915
06920 0A14 86 41
06930 0A16 BD 0DDC
06935
06940 0A19 86 01
06945 0A1B 97 2A
06950 0A1D BD 0B8A
06955 0A20 39

```

```

*
*INCREMENT SCORE BY 100
DRP100 LDA A    #$41
        JSR      SCOREX
*INCREMENTS BONUS BY 100
        LDA A    #%00000001
        STA A    ADDEND+1
        JSR      RBONUS
        RTS

```

```

06990
06995
07000
07005
07010
07015
07020
07025
07030
07035
07040
07045
07050
07055
07060 0A21 36
07065 0A22 96 24
07070 0A24 CE 0086
07075 0A27 01
07080 0A28 DB 0AD0
07085 0A2B C6 06
07090 0A2D BD 0C9E
07095 0A30 01 01
07100 0A32 01 0101
07105
07110 0A35 86 7F
07115 0A37 94 3C
07120 0A39 97 3C
07125 0A3B 32
07130 0A3C 39
0001
0005
10010
10015
10020
10025
10030
10035
10040
10045
10050
10055
10060
10065
10070 0A3D 7A 0020
10075
10080
10085
10090
10095
10010 0A40 7D 002E
10105 0A43 27 08
10110 0A45 7A 002F

```

```

*NAM QUICK
*THIS ROUTINE ENERGIZES EITHER THE T.B.,
*RSLGSHT,OR LSLGSHT SOLENOIDS FOR 75 MS. THE
*ROUTINE IS ACTIVATED BY A 1 INBIT 7 OF XEDGE+1
*WHICH SHALL BE LOADED BY RDMAT WHENEVER A 1
*IS DETECTED AT THE CORRESPONDING SOLENOID SWITCH.
*THE SWITCH DEBOUNCE ROUTINE IS BYPASSED WHEN
*LOADING XEDGE+1, BIT 7.THE ROUTINE ALSO CLEARS
*THIS BIT AFTER THE 75 MS.SOLEN. ACTIVATE
*TIME.
*INPUT DATA CONSISTS OF THE BIT POSITION OF THE
*SOLENOID TO BE OPERATED AND IS INPUT IN BITNUM.
*
*
QUICK    PSH A
        LDA A BITNUM BITNUM HOLDS SOLEN. BIT#.
        LDX #PIA1DB LOAD SOLEN. WORD LOCATION.
        JSR
        LDA B #6 WAIT 50 MS.
        JSR DELAY
*CLEAR BIT 7,XEDGE+1 SINCE THE JOB IS DONE.
        LDA A    #$7F
        AND A    XEDGE+1
        STA A    XEDGE+1
        PUL A
        RTS
* P R G S U B
*NAM ZEROCR
*THIS ROUTINE IS CALLED BY THE Z.C. INTERRUPT AND
*INITIATES ALL Z.C. RELATED ROUTINES
*THE ROUTINE:
* 1)DECREMENTS COUNTERZCCNT FOR THE ROUTINE
* 2)TESTS AND RESETS WAIT FLAG .5 SEC AFTER IT
* SET TO GENERATE A DEAD TIME FOR THE TOP HOLE
* SUBROUTINE
* 3) GENERATE 1446 US.TIME DELAY FROM Z.C. BEFORE
* ENABLING SCR WRITE OPERATION
* 4)CALLS SCR LAMP TRIGGER ROUTINE WRTLMP
* 5)READS MOMENTARY SWITCH DATA FROM PLAYFIELD
* USING RDMAT
10070 ZEROCR DEC      ZCCNT  USED BY SUBROUTINE DELAY
*
*
*HOLD OF ROUTINE USED TO MASK TOP HOLE SENSOR FOR
*.5 SEC. AFTER SOLENOID OPERATION.DECREMENTS WAIT
*IF WAIT NOT EQUAL TO ZERO.

```



```

10115 0A48 26 07
10120 0A4A 7F 002E      LDA A WAIT
                        BEQ RSTCH
                        DEC WAIT

10125 0A4D 86 3C
10130 0A4F 97 2F
10135 0A51 01
10140
10145
10150
10155
10160 0A52 C6 19      *GENERATE 600US.TIME DELAY FROM Z.C. TO PROVIDE
10165 0A54 5A          *SUFFICIENT VOLTAGE ACROSS THE LAMP DRIVER SCR
10170 0A55 26 FD      RSTCN LDA B #25 START 300 US. TIMER (.5MHZ.)
                        TIMLP 2 DEC B
                        BNE TIMLP2

10175
10180
10185
10190 0A57 BD 0D6F    *TRIGGER ALL LAMP DRIVER SCR THAT SHOULD BE ON
                        JSR WRTIMP
10195
10200
10205
10220 0A5A BD 0B27    *READ MOMENTARY SWITCH DATA FROM PLAYFIELD
                        JSR RDMAT
10225 0A5D 3B          RTI
10345
10350
10355
10360
10365
10370
10375
10380
10385
10390
10395
10400
10405
10410
10415 0A5E 09BC
10420 0A60 099 A
10425 0A62 09D1
10430 0A64 09D7
10435 0A66 0952
10440 0A68 094D
10445 0A6A 08F6
10450 0A6C 0AB8
10455 0A6E 09B2
10460 0A70 090F
10465 0A72 0964
10470 0A74 0A14
10475 0A76 0971
10480 0A78 0957
10485 0A7A 0A0E
10490 0A7C 0A21
10495 0A7F 09B5
10500 0A80 0987
10505 0A82 09E3
10510 0A84 09DD
10515 0A86 0943
10520 0A88 0948
10525 0A A 09AD
10530 0A8C 0922
10575
10580
10585
10590 0A8E CE 0A8C    *NAM TARMON
10595 0A91 DF 1L      *TARMON-THIS ROUTINE MONITORS TARGET HITS AS
                        *REPRESENTED BY XEDGE(0) THRU XEDGE (2).
                        *THE ROUTINE SCANS EACH BIT OF THE FOUR WORDS,
                        *JUMPING TO A DESIGNATED SUBROUTINE WHENEVER A BIT
                        *IS SET. THE SUBROUTINE ADDRESS ASSOCIATED WITH
                        *EACH BIT OF EACH WORD IS STORED IN ARRAY SUBTBL.
                        *BIT 7 OF XEDGE(2) CORRESPONDS TO THE SUBROUTINE
                        *ADDRESS GIVEN AT SUBTBL+46 AND BIT 0 OF XEDGE(0)
                        *CORRESPONDS TO SUBTBL(0) NOTE THAT SUBTBL
                        *MUST BE LOCATED SUCH THAT THE BINARY ADDITION OF
                        *46 TO SUBTBL MUST NOT CAUSE A CARRY INTO THE
                        *HIGHER ORDER BYTE.
                        *DEFINE SUBROUTINE ADDRESS TABLE SUBTBL.
SUBTBL FDB CHANNL LT.500PT.CHANNEL 10,PA0
FDB LSLGST LT.SLING SHOT 11,PA0
FDB DROPB DROP TARGET(A) 12,PA0
FDB DRPPB DROP TARGET(B) 13,PA0
FDB RLVSWA ROLL OVER (A) 14,PA0
FDB RLVSWB ROLL OVER (B) 15,PA0
FDB TOPHOL TOP HOLE(3000PT.) 16,PA0
FDB BYPSUB 17,PA0
FDB CHAN1K 1K CHANNEL 10,PA1
FDB THUBPR THUMPER BUMPER 11,PA1
FDB REBOUN REBOUND(10PT.) 12,PA1
FDB DRP100 DROP TARGET(100PT) 13,PA1
FDB SPICAL SPECIAL 14,PA1
FDB RLVSW 1K ROLL OVER 15,PA1
FDB DRP500 DROP TARGET(500PT) 16,PA1
FDB QUICK -----
FDB CHANNR RT.500PT.CHANNEL 10,PA2
FDB RSLGST RT.SLING SHOT 11,PA2
FDB DROPD DROP TARGET(D) 12,PA2
FDB DROPC DROP TARGET(C) 13,PA2
FDB RLVSWD ROLL OVER(D) 14,PA2
FDB RLVSWC ROLL OVER(C) 15,PA2
FDB OUTHL OUT HOLE 16,PA2
FDB RTILT TILT 17,PA2
10575
10580
10585
10590 0A8E CE 0A8C    *LOAD VALUES FOR INTIAL SUBTBL ENTRY POINT (ENTRY
10595 0A91 DF 1L      *FIRST WORD OF XEDGE TO BE READ, AND FIRST BIT
                        *TO BE READ.
TARMON LDX #SUBTBL+46 FIRST SUB. POINTER TABL
STX ENTRY

```

```

10600 0A93 CE 0003      LDX      #3          FIRST VALUE TO GET XEDGE+2
10605 0A96 C6 07      WORDLP LDA B      #7
10610                      *DISABLE Z.C. INTERRUPT
10615 0A98 86 05      LDA A      #%00000101
10620 0A9A 97 8B      STA A      PIA2CB
10625                      *READ XEDGE AND THEN CLEAR.
10627 0A9C A6 3A      LDA A      XEDGE-1,X
10632 0A9E 6F 3A      CLR       XEDGE-1,X
10634                      *ENABLE Z.C. INTERRUPT.
10635 0AA0 36          PSH A
10636 0AA1 86 0D      LDA A      #%00001101
10637 0AA3 97 8B      STA A      PIA2CB
10638 0AA5 32          PUL A
10642                      *TEST XEDGE WORD AND BYPASS BIT TESTS IF ALL
10643                      *BITS = 0.
10644 0AA6 4D          TST A
10645 0AA7 27 1D      BEQ       FAST
10646                      *TEST EACH BIT AND GO TO SUB IF 1.
10647 0AA9 4D      BITLP1 TST A
10648 0AAA 2A 0E      BPL       SKIP3
10649 0AAC DF 00      STX       ADDR2      STORE X TEMPORARILY
10650 0AAE DE 1E      LDX       ENTRY     LOAD THE SUBTBL ENTRY POINT
10655 0AB0 EE 00      LDX       0,X       LOAD THE ADDR. OF THE SUBR.
10662 0AB6 33          PUL B
10663 0AB7 32          PUL A
10665 0AB8 DE 00      BYPSUB LDX      ADDR2      RESTORE X
10670                      *
10675                      *ROTATE NEXT BIT TO N FLAG, DEC. BIT COUNTER AND
10680                      *SUBTBL ENTRY POINTER, AND TEST FOR LAST BIT IN WO
10685 0ABA 49      SKIP3 ROL A
10690 0ABB 7A 001F      DEC      ENTRY+1      DECREMENT POINTER BY 2.
10695 0ABE 7A 001F      DEC      ENTRY+1
10700 0AC1 5A          DEC B      BIT COUNTER
10705 0AC2 2A E5      BPL      BITLP1      BRANCH IF ALL BITS NOT READ
10721 0AC4 20 06      BRA      FASTRT
10722 0AC6 86 F0      FAST LDA A      #$0F0      SUBT. 16 FROM ENTRY+1
10723 0AC8 9B 1F      ADD A      ENTRY+1
10724 0ACA 97 1F      STA A      ENTRY+1
10735                      *DEC WORD COUNTER (X), CHECK FOR LAST WORD, AND GO
10740                      *TO WORDLP TO CONTINUE.
10745 0ACC 09      FASTRT DEX
10750 0ACD 26 C7      BNE      WORDLP
10755 0ACF 39          RTS
10765                      *NAM SLNPLS
10770                      *THIS ROUTINE PROVIDES A 25MS PULSE TO THE M ADDRE
10775                      *AND BIT LOCATION LOADED INTO X AND A RESPECTIVELY
10785 0AD0 5F      SLNPLS CLR B
10790 0AD1 BD 0D1C      JSR      BITWRT      OPERATE SOLENOID
10795 0AD4 C6 03      LDA B      #3          HOLD FOR 25MS
10800 0AD1 BD 0C9E      JSR      DELAY
10810 0AD9 C6 01      LDA B      #1
10815 0ADB BD 0D1C      JSR      BITWRT
10820 0ADE 39          RTS
10830                      *NAM SERIN
10835                      * THIS ROUTINE READS SERIAL DATA PRESENTED AT
10840                      * PIA2AD7 AND CLOCKS 8 BITS OF DATA TO
10845                      *XWD. THE FIRST BIT READ IS LOADED INTO
10850                      * MEMORY AT BIT POSITION 7 AND THE LAST INTO BIT 0
10855                      *PIA2AC-CA2 IS THE SHIFT REGISTER
10860                      * CLOCK SIGNAL AND REQUIRES THAT CRA-5=
10865                      *CRA-3=1,CRA-4=0. A &X ARE UNCHANGED
10870                      *
10875                      *WRITE A ONE TO PIA2CB2 TO ENABLE PARALLEL
10876                      *S.R. LOAD.
10877 0ADF 36      SERIN  PSH A
10878 0AE0 86 3D      LDA A      #%00111101

```

```

10879 OAE2 97 8B      STA A    PIA2CB
10880                *150 US. ADDR. SETTLE TIME.
10881 OAE4 C6 0C      LDA B    #12
10882 OAE6 5A        TIME3 DEC B
10883 OAE7 26 FD      BNE      TIME3
10884                *WRITE A ZERO TO ENABLE SERIAL TRANSF.
10885 OAE9 86 35      LDA A    #%00110101
10886 OAEB 97 8B      STA A    PIA2CB
10920                * LOAD PIA1DA7 INTO FLAG C AND THEN TO M(0)<INADDR
10923 OAE8 C6 07      LDA B    #7
10925 OAEF 96 84      SERLP    LDA A    PIA1DA
10930 OAF1 49        ROL A
10935 OAF2 79 0007    ROL      XWD
10940 OAF5 5A        DEC B
10945 OAF6 2A F7      BPL      SERLP    REPEAT UNTIL ALL 8 BITS HAV
10950                *BEEN STORED AT XWD.
10951                *RESET PIA2CRB TO ENABLE RECEIPT OT ZC
10952 OAF8 86 0D      LDA A    #%00001101
10953 OAF9 97 8B      STA A    PIA2CB
10954 OAF0 32        PUL A
10955 OAFD 39        RTS
10965                *NAM SCRPLS
10970                *THIS SUB. FORMS SCORE (NUP)=SCORE(NUP)+ADDEND
10975                *ADDEND AND NUP ARE RAM VARIABLES THAT SHALL BE
10980                *SET BEFORE EXECUTION. ADDEND CONSISTS OF SIX HALF
10985                *BYTES IN BCS. THE LS HALF BYTE OF ADDEND CORRESPONDS
10990                *TO THE UNITS DECIMAL LOCATION. ADDEND IS RESET TO
10995                *ZERO AT END OF BCD TO INITIALIZE FOR NEXT SCRPLS
11000                *CALL
11005                *
11010                *
11015                *
11020                *DETERMINE THE STARTING ADDRESS OF SCORE(NUP) AND
11025                *LOAD ITS VALUE TO X.
11030 OAFE          SCRPLS
11031 OAFF BD        OF97    JSR      TASK40  DEBLANK SCORE LOCATIONS.
11032 OB02 96 22      LDA A    NUP
11035 OB04 4A        DEC A
11040 OB05 26 03      BNE      BYPAS1
11045 OB07 CF 0042    LDX      #SCORE1
11050 OB0A 4A        BYPAS1 DEC A
11055 OB0B 26 03      BNE      BYPAS2
11060 OB0D CE 0045    LDX      #SCORE2
11065 OB10 4A        BYPAS2 DEC A
11070 OB11 26 03      BNE      BYPAS3
11075 OB13 CE 0048    LDX      #SCORE3
11080 OB16 4A        BYPAS3 DEC A
11083 OB17 26 03      BNE      NEXT
11085 OB19 CE 004B    LDX      #SCORE4
11090                *CALL THE ADDITION ROUTINE NOW THAT X HAS BEEN
11095                *LOADED WITH THE STARTING ADDRESS OF SCORE(NUP).
11100 OB1C BD 0D41    NEXT    JSR      BCD
11102 OB1F          JSR
11103 OB20 BD 0F00    JSR      TASK4    BLANK SCORE LOCATIONS.
11105 OB23 39        RTS
11118                *NAM RDMAT
11121                *THIS ROUTINE IS CALLED BY THE Z.C. INTERRUPT
11124                *SERVICE ROUTINE(ZERO CR) OR THE MAIN AND READS DA
11127                *THE 7 X 8 INPUT MATRIX . THE DATA READ MAY BE
11130                *EITHER OF TWO GROUPS , PLAYFIELD MOMENTARY
11133                *SWITCH DATA OR FIXED DATA FROM CONTROL SWITCHES
11136                *FOR PLAYFIELD = 1 THE ROUTINE READS
11139                *MOMENTARY SWITCH DATA LOCATED AT MATRIX ADDRESS
11142                *01,02,04,08. FOR PLAYFIELD = 0 THE FIXED DATA
11145                *SWITCHES AT 10,20,AND 40, ARE READ. THE MATRIX A
11148                *ARE WRITTEN TO PIA1DA0-6 AND MATRIX WORDS ARE

```

```

11151
11154
11157
11160
11163
11166
11169
11172
11199
11202
11203 0B24 03
      0B25 04
      0B26 02
11205 0B27 96 10
11208 0B29 27 4A
11211
11214
11217 0B2B CE 0004
11220 0B2E 86 08
11223
11226 0B30 97 84
11229 0B32 BD 0ADF
11232 0B35 D6 07
11235
11238
11241 0B37 E8 32
11242 0B39 53
11244 0B3A D7 0C
11247
11250
11253 0B3C D4 07
11256 0B3E D7 09
11259
11262
11265 0B40 D6 0C
11268 0B42 53
11271 0B43 E4 36
11274 0B45 DA 09
11277 0B47 D7 09
11280
11283
11286 0B49 E6 36
11289 0B4B 53
11292 0B4C D4 09
11293 0B4E EA 3A
11295 0B50 E7 3A
11298
11299
11300
11301
11302
11303
11304
11305
11306
11307
11308
11309 0B52 8C 0004
11310 0B55 27 10
11311 0B57 D6 07
11312 0B59 56
11313 0B5A 56
11314 0B5B 24 0A
11315 0B5D C6 80
11316 0B5F DA 3C
11317 0B61 D7 92

*READ SERIALY AT PIA1DA7. MOMENTARY DATA IS
*DE-BOUNCED AND WRITTEN TO XEDGE WHENEVER TWO
*CONSECUTIVE ONES ARE SAMPLED. THESE BITS ARE NOT
*RESET TO ZERO BY THIS ROUTINE. FIXED DATA
*IS WRITTEN DIRECTLY TO XEDGE EXACTLY AS THE DATA
*IS RECEIVED FROM THE INPUT MATRIX. INPUT DATA
*REQUIRED IS PLAYFIELD WHICH MUST BE SET BY THE
*CALLING PROGRAM.
*TEST PLYFLD AND BRANCH TO FIXED IF 0 ELSE
*DO MOMENTARY.
BITTBL FCB      3,4,2

RDMAT LDA A      PLYFLD
      BEQ        FIXED
*INITIALIZE ROUTINE FOR READING MOMENTARY
*PLAYFIELD SWITCH DATA
      LDX        #4
      LDA A      #%00001000
*READ INPUT WORD TO XWD
MOMLP STA A      PIA1DA
      JSR        SERIN
      LDA B      XWD
*GENERATE REPEAT (DEFINE BIT POSITIONS THAT
*HAVE REPEATED THEIR PREVIOUS STATE)
      EOR B      LSTXWD-1,X
      COM B
      STA A      REPEAT
*FOR THOSE BITS WHICH DID REPEAT, ENTER THE
*REPEATED VALUE TO XST
      AND B      XWD
      STA B      XST
*FOR THOSE BITS WHICH DID NOT REPEAT, RETAIN
*THE PREVIOUS VALUE OF XST
      LDA B      REPEAT
      COM B
      AND B      LSTXST-1,X
      ORA B      XST
      STA B      XST
*WRITE 1 TO XEDGE FOR THOSE POSITIONS THAT
*HAVE CHANGED FROM 0 IN LSTXST TO 1 IN XST
      LDA B      LSTXST-1,X
      COM B
      AND B      XST
      ORA B      XEDGE-1,X
      STA B      XEDGE-1,X WRITE FINISHED XEDGE TO
*TEST TO DETERMINE IF T.B.,LSLGSHT, OR RSLGSHT
*CONTROL SWITCH IS A 1. IF YES THEN SET
*XEDGE +1,BIT 7=1,BYPASSING THE DEBOUNCE
*ROUTINE. BIT 7 IS A FLAG TO TARMON FOR
*A SOLENOID PULSE. LOAD THE BIT LOCATION OF THE
*DETECTED SOLEN. TO BITNUM FOR USE BY QUICK
*(SOLENI ROUTINE CALLED BY TARMON).
*
*
*TEST FOR A 1 AT TB, LSLGSHT, OR RSLGSHT SWITCH
* =1.
      CPX        #4
      BEQ        NOSOLN
      LDA B      XWD
      ROR B
      ROR B
      BCC        NOSOLN BRA IF CONTROL SWT.<>1
      LDA B      #$80 SET BIT 7 OF XEDGE+1=1.
      ORA B      XEDGE+1
      STA B      XEDGE+1

```

```

11318 *LOAD BITNUM WITH BIT POS. OF SOLEN. COIL
11319 *THAT IS TO BE ACTIVATED.
****ERROR 210 1-1,X OF74: CPX #3 8C 00 03
11302 OB63 1E 0F LDA B BITTBL-1,X REQ PIS2 27 0A
11321 OB65 14 01 STA B BITNUM CPX #2 8C 00 02
11322 OB67 01 NOSOLN NOP REQ B154 27 C9
11323 *RETAIN THE XWD CPX #1 18C 00 01
11324 OB68 D6 07 LDA B XWD REQ B153 27 08
11325 OB6A E7 32 STA B LSTXWD-B152 LDA B#2 C6 02
11326 *RETAIN XST 1,X BRA LOAD 20 06
11327 OB6C D6 09 LDA B XST B154 LDA B#4 C6 04
11328 OB6F E7 36 STA B LSTXST01,X BRA LOAD 20 02
B153 LDA B#3 C6 03

11329 *DEC X AND SHIFT ADDR
11330 OB70 09 DEX LOAD STA B BITNUM D7 24
11331 OB71 44 LSR A
11332 OB72 22 BC BHI MOMLP JMP OB67 7E OB 67
11333 OB74 39 RTS
11334 *INITIALIZE ROUTINE FOR READING FIXED DATA
11337 OB75 CE 0007 FIXED LDX #7
11340 OB78 86 40 LDA A #%01000000
11343 *READ INPUT WORD TO XWD
11346 OB7A 97 84 FIXLP STA A PIA1DA
11349 OB7C BD 0ADF JSR SERIN
11352 OB7F D6 07 LDA B XWD
11355 OB81 E7 3A STA B XEDGE-1,X
11358 OB83 09 DEX
11361 OB84 44 LSR A
11364 OB85 81 08 CMP A #%00001000
11367 OB87 26 F1 BNE FIXLP
11370 OB89 39 RTS
11450 *NAM RBONUS
11455 *THIS ROUTINE IS USED TO ADD THE AMOUNT OF THE BON
11460 *INCREMENT TO BONUS. THE ROUTINE FORMS BONUS=BONUS
11465 *+ADDEND WHERE ADDEND IS SET PRIOR TO EXECUTION.
11470 *ADDEND CONSISTS OF SIX HALF BYTES IN BCD. THE LS
11475 *BYTE OF ADDEND CORRESPONDS TO THE UNITS DECIMAL
11480 *LOCATION. ADDEND IS RESET TO ZERO AT END OF BCD T
11485 *INITIALIZE FOR NEXT EXECUTION. THE BONUS VALUE IS
11490 *CHECKED AFTER EACH ENTRY AND SPCLP (SPECIAL LAMP)
11495 *=1 IF BONUS => 15,000.
11496 *THE ROUTINE LIGHTS THE LEFT AND RIGHT EXTRA BALL
11498 *SCORES => 8000. LEFT ON EVEN SCORES AND RIGHT ON
11500 *BOTH LAMPS ARE ENABLED FOR => 15,000.
11502 *THE MAXIMUM BONUS VALUE IS LIMITED TO 15,000
11504 *
11506 *
11508 *LOAD BONUS STARTING ADDRESS AND INCREMENT BONUS
11510 OB8A CE 0025 RBONUS LDX #BONUS
11512 OB8D BD 0D41 JSR BCD
11514 *TEST BONUS FOR VALUE => 15,000. IF YES LIMIT BON
11516 *ENABLE SPECIAL LAMP, AND ENABLE BOTH LEFT AND RI
11518 *EXTRA BALL LAMPS.
11520 OB90 96 27 LDA A BONUS+2
11522 OB92 27 14 BEQ LESS15K
11524 OB94 86 50 LDA A #$50
11526 OB96 91 26 CMP A BONUS+1
11528 OB98 2A 0E BHI LESS15K
11530 OB9A 86 F4 LDA A #SPCLP1 LIGHT SPECIAL LAMP.
11532 OB9C BD 0C7D JSR LPCTRL
11534 OB9F 86 50 LDA A #$50 SET BONUS 10E3=5
11536 OBA1 97 26 STA A BONUS+1
11538 OBA3 86 B5 LDA A #LXBLP1 LIGHT LEFT EXTRA BALL LAMP
11540 OBA5 BD 0C7D JSR LPCTRL
11542 *LIGHT LEFT OR RIGHT EXTRA BALL LAMPS FOR BONUS>
11544 OBA8 96 27 *LESS 15 LDA A BONUS+2

```

```

11546 OBAA 26 06      BNE      EVENX
11548 OBAC 96 26      LDA A      BONUS+1
11550 OBAE 81 80      CMP A      #$80
11552 OBB0 2B 1B      BMI      ENDENS
11554                *IS BONUS 1000 EVEN OR ODD AND LIGHT EXTRA BALL L
11556 OBB2 48      EVENX ASL A
11558 OBB3 48      ASL A
11560 OBB4 48      ASL A
11562 OBB5 2B 0C      BMI      ODD
11564 OBB7 86 B5      EVEN  LDA A      #LXBLP1
11566 OBB9 BD 0C7D      JSR      LPCTRL
11568 OBBC 86 36      LDA A      #RXBLP0
11570 OBBE BD 0C7D      JSR      LPCTRL
11572 OBC1 20 0A      BRA      ENDBNS
11574 OBC3 86 B6      ODD   LDA A      #RXBLP1
11576 OBC5 BD 0C7D      JSR      LPCTRL
11578 OBC8 86 35      LDA A      #LXBLP0
11580 OBCA BD 0C7D      JSR      LPCTRL
11582                *CALL BONUS LAMP DISPLAY
11584 OBCE BD 0E7C      ENDBNS JSR      BONDSP
11586 OBDO 39      RTS
11591                *NAM RAMZR
11592                *THIS ROUTINE CLEARS ALL RAM MEM.AT X AND
11593                *BELOW TO ZERO. X IS LOADED BY THE CALLER.
11594 OBD1 6F 00      RAMZR CLR      0,X
11595 OBD3 09      DEX
11596 OBD4 26 FB      BNE      RAMZR
11597 OBD6 6F 00      CLR      0,X
11598 OBD8 39      RTS
11610                *NAM NXTPLY
11615                *THIS ROUTINE PREPARES THE GAME FOR THE NEXT PLAYE
11620                *AFTER THE PREVIOUS PLAYER HAS COMPLETED HIS TURN.
11625                *THE ROUTINE LIGHTS THE FOUR DROP TARGET LAMPS,
11630                *EXTINGUISHES THE ASSOCIATED ROLLOVER LAMPS, AND
11640                *RESETS THE TILT FLAG AND LAMP,ENABLES FLIPPER
11641                *ACTION AFTER TILT, RESETS SPECIAL LAMP,
11645                *RESET EXTRA BALL LAMPS.
11650                *RESET ROLLOVER LAMPS & LIGHT DROP TARGET LMPS.
11653 OBD9 CE 001C      NXTPLY LDX      #LMPMAT+7      LOAD WORD LOCATION
11655 OBDC 4F      CLR A      LOAD BIT POSITION
11657 OBDD 5F      CLR B      LOAD BIT VALUE
11660 OBDE BD 0D1C      LOOPLP JSR      BITWRT      BITWRT RETURNS A,B,X UNCHAN
11663 OBE1 8C 0019      CPX      #LMPMAT+4      ALL ROLLOVRS. COMPLETED
11665 OBE4 26 02      BNE      BYPAS5
11667 OBE6 C6 01      LDA B      #1      CHANGE BIT VALUE TO 1 FOR D
11670 OBE8 09      BYPAS5 DEX
11673 OBE9 8C 0014      CPX      #LMPMAT-1      DRP.TSRG. &ROLLVRS. COMP
11675 OBEC 26 F0      BNE      LOOPLP      REPEAT UNLESS COMPLETED.
11700                *RESET TILT LAMP,TILT FLAG AND ENABLE FLIPPER
11701 OBEE 86 76      LDA A      #TLTLPO
11702 OBFO BD 0C7D      JSR      LPCTRL
11703 OBF3 7F 002C      CLR      TILT
11704 OBF6 86 FC      LDA A      #$0FC
11705 OBF8 97 86      STA A      PIA1DB
11706                *RESET SPECIAL LAMP.
11707 OBFA 86 74      LDA A      #SPCLPO
11708 OBFC BD 0C7D      JSR      LPCTRL
11709                *RESET L&R EXTRA BALL LAMPS.
11715 OBF5 86 35      LDA A      #LXBLP0
11720 OC01 BD 0C7D      JSR      LPCTRL
11725 OC04 86 36      LDA A      #RXBLP0
11730 OC06 BD 0C7D      JSR      LPCTRL
11735 OC09 39      RTS
11785                *NAM MATCHK
11790                *THIS ROUTINE COMPARES THE SCORES OF ACTIVE PLAYER
11795                *WITH MATCH AND THE FREE GAME THRESHOLD(FREEGM)

```

```

11800      *CREDIT IS INCREMENTED FOR EACH MATCH AND EACH
11805      *SCORE>=FREEGM,ENABLES MATCH DISPLAY (RESET MAIN)
11810      *
11815      *FORM ADDRESS OF SCORE (NPLAY).
11820      *THE COMPUTED VALUE IS PLACED IN ADDR. ADDR=
11825      *<SCORE>-3+3*NPLAY.
11830      OCOA      MATCHK
11831      OCOB BD    OF97      JSR      TASK40  DEBLANK ALL SCORE LOC.
11832      OCOE 96 23      LDA A      NPLAY
11835      OC10 48        ASL A
11840      OC11 9B 23      ADD A      NPLAY
11845      OC13 8B 3F      ADD A      #SCORE1-3
11850      *LOAD COMPUTED ADDR. TO X AND MATCH TO A
11855      OC15 7F 0002    CLR      ADDR
11857      OC1A DE 02      MTCHLP LDX      ADDR
11865      OC1C 96 28      LDA A      MATCH
11870      *TEST FOR MATCH; INC CREDIT & PULSE KNOCKER
11872      OC1E A1 00      CMP A      0,X
11874      OC20 26 0F      BNE      FGMCHK
11876      OC22 C6 01      LDA B      #1      INC CREDIT
11878      OC24 DB 4E      DAA
11882      OC27 D7 4E      STA B      CREDIT
11884      OC29 CE 0086    LDX      #PIA1DB      PULSE KNOCKER
11886      OC2E BD 0AD0    JSR      SLNPLS
11895      *NOW THAT MATCH HAS BEEN CHECKED, CHECK SAME PLAYER
11900      *SCORE AGAINST THE FREE GAME THRESHOLD.
11905      *
11910      *LOAD TEMP WITH THE 1000 PLACE OF SCORE, SETTING
11915      *THE 10000 PLACE TO ZERO
11920      OC31 DE 02      FGMCHK LDX      ADDR      RESTORE X
11922      OC33 E6 01      LDA B      1,X
11925      OC35 54        LSR B
11930      OC37 54        LSR B
11935      OC37 54        LSR B
11940      OC38 54        LSR B
11945      OC39 D7 1D      STA B      TEMP
11950      *LOAD A WITH THE 1000 PLACE OF THRESHOLD,
11955      *MASKING THE 10000 PLACE TO ZERO
11960      OC3B 86 0F      LDA A      #%00001111
11965      OC3D 94 3F      AND A      XEDGE+4 (FREEGM)
11970      *COMPARE THRESHOLD-SCORE (1000 PLACE)
11975      OC3F 91 1D      CMP A      TEMP
11980      OC41 24 0A      BCC      END1
11985      *TEST FOR A 1 IN 10,000 PLACE TO BORROW FROM
11986      *AND DO IF YES
11988      OC43 86 F0      LDA A      #$0F0
12000      OC45 94 3F      AND A      XEDGE+4      (FREEGM)
12002      OC47 4D      TST A
12004      OC48 27 0F      BEQ      END2
12006      OC4A 4A      DEC A
12008      OC4B 84 0F      AND A      #$0F MASK EXTRA BITS FROM SUBT.
12010      *LOAD 10000 PLACE OF SCORE
12015      OC4D E6 02      END1      LDA B      2,X
12020      OC4F 58      ASL B
12025      OC50 58      ASL B
12030      OC51 58      ASL B
12035      OC52 58      ASL B
12040      OC53 D7 1D      STA B      TEMP
12045      *COMPARE THRESHOLD-SCORE (10000 PLACE)
12050      OC55 91 1D      CMP A      TEMP
12055      OC57 22 11      BHI      NXTSCR
12060      *INC. CREDIT
12065      OC59 86 01      END2      LDA A      #1
12070      OC5B 9B 4E      ADD A      CREDIT
12075      OC5D 19      DAA
12076      OC5E 97 4E      STA A      CREDIT

```

```

12077 0C60 CE 0086      LDX      #PIA1DB      PULSE KNOCKER
12078 0C63 86 07        LDA A      #7
12079 0C65 BD 0AD0      JSR      SLNPLS
12080 0C68 DE 02        LDX      ADDR      RESTORE X
12081          .        *DEC X TO LOCATION OF NEXT SCORE TO BE COMPARED WI
12085          *MATCH AND TEST FOR LAST SCORE CHECK COMPLETED.
12090 0C6A 09          NXTSCR DEX
12095 0C6B 09          DEX
12100 0C6C 09          DEX
12102 0C6D DF 02        STX      ADDR
12105 0C6F 8C 003F      CPX      #SCORE1-3
12110 0C72 26 A6        BNE      MTCHLP      DROP THRU AFTER SCORE1
12111          *ENABLE MATCH DISPLAY
12112 0C74 96 28        LDA A      MATCH
12113 0C76 97 50        STA A      MATCHD
12114 0C78
12115 0C79 BD 0F00      JSR      TASK4      BLANK ALL SCORE LOC.
12116 0C7C 39          RTS
12125          *NAM LPCTRL
12130          *THIS ROUTINE IS USED TO TURN LAMPS ON AND OFF. TH
12135          *INPUT DATA IS A CONTROL WORD STORED IN A BY
12140          *THE CALLING PROGRAM. THE CONTROL WORD FORMAT:
12145          *      BIT      FUNCTION
12150          *      0-2      POS OFFSET FROM LMPMAT(DECODER LE
12155          *      3        DISREGARDED BY LPCTRL
12160          *      4-6      LMP BIT POSITION(DECODER NO.)
12165          *      7        1 EQUALS LAMP ON
12170          *
12175          *      B IS UNCHANGED
12180          *
12200          *READ LMPMAT OFFSET VALUE AND ADD TO <LMPMAT>
12202 0C7D 37          LPCTRL PSH B
12203 0C7E 5F          CLR B
12204 0C7F 97 1D        STA A      TEMP
12205 0C81 86 07        LDA A      #%00000111      MASK
12210 0C83 94 1D        AND A      TEMP
12215 0C85 8B 15        ADD A      #LMPMAT
12220          *LOAD <LMPMAT> + OFFSET TO X
12223 0C87 7F 0002      CLR      ADDR
12225 0C8A 97 03        STA A      ADDR+1
12230 0C8C DE 02        LDX      ADDR
12235          *LOAD THE LAMP CONDITION, 1/0, INTO B.
12240 0C8E 79 001D      ROL      TEMP
12245 0C91 59          ROL B
12250          *LOAD THE LMPMAT BIT LOCATION CODE INTO A.
12255 0C92 86 E0        LDA A      #%11100000      MASK
12260 0C94 94 1D        AND A      TEMP
12265 0C96 49          ROL A
12270 0C97 49          ROL A
12275 0C98 49          ROL A
12280 0C99 49          ROL A
12283 0C9A 33
12285 0C9B 7E 0D1C      JSR      BITWRT
                        PUL B
                        RTS
12300          *NAM DELAY
12310          *DELAY(SUBROUTINE)-THIS SUBROUTINE CREATES A
12315          *DELAY OF 8.33N,+8.33,-OMS. THE NUMBER OF DELAY
12320          *PERIODS N SHALL BE ENTERED INTO B PRIOR TO EXECUT
12325          *
12330          *
12335 0C9E D7 20        DELAY STA B      ZCCNT
12340 0CA0 D6 20        TIMLP LDA B      ZCCNT
12345 0CA2 26 FC        BNE      TIMLP
12350 0CA4 39          RTS

```



```

12360 *NAM CNMON
12365 *THIS ROUTINE MONITORS COIN CHUTE 1 AND COIN CHUTE
12370 *AND INCREMENTS CREDIT AS DEFINED BY THE INPUT
12375 *DATA SWITCH SETTINGS,THESE DATA SWITCHS PERMIT
12380 *TWO VARIABLES TO BE ASSOCIATED WITH EACH SWITCH.
12385 *THE VARIABLES ARE GAMES1,WHICH DEFINES THE NUMBER
12390 *OF CREDITS ISSUED WHEN THE NO. OF COINS ENTERED
12395 *EQUALS THE SECOND VARIABLE,COINS1.ADDITIONALLY,
12400 *THE ROUTINE MONITORS THE RESET BUTTON AND
12405 *INCREMENTS NO. OF PLAYERS IF CREDIT AND NPLAY
12410 *ARE OK.THE EXIT TO NEWGAM IS USED WHENEVER THE
12415 *RESET SWITCH IS SENSED AFTER THE FIRST BALL HAS
12420 *BEEN PLAYED AND CREDIT>0.
12425 *THE ROUTINE READS CNBIT1,CNBIT2,COINS1,
12430 *COINS2,GAMES1,GAMES2,AND CREDIT FROM XEDGE
12435 *
12440 *MONITOR COIN CHUTES AND GIVE CREDITS A/R
12445 *READ CNBIT1 AND INCREMENT CNCNT1 IF PRESENT.
12450 OCA5 D6 3E CNMON LDA B XEDGE+3
12455 OCA7 2A 1A BPL BYPCN1 BRANCH IF COIN BIT 0
12460 OCA9 7C 0030 INC CNCNT1
12465 *TEST CNCNT1 AND CREDIT=CREDIT+GAMES1 IF CNCNT1
12470 *=COINS1.
12465 OCAC 86 38 LDA A #%00111000 MASK FOR COINS1
12480 OCAF 94 40 AND A XEDGE+5
12481 OCB0 44 LSR A
12482 OCB1 44 LSR A
12483 OCB2 44 LSR A
12485 OCB3 91 30 CMP A CNCNT1
12490 OCB5 26 0C BNE BYPCN1 BRANCH IF CNCNT1><COINS1
12495 OCB7 86 07 LDA A #%00000111 MASK TO LOAD GAMES1
12500 OCB9 94 40 AND A XEDGE+5
12505 OCB8 9B 4E ADD A CREDIT
12510 OCB0 19 DAA
12513 OCBE 97 4E STA A CREDIT
12515 *RESET THE CHUTE1 COIN COUNTER
12520 OCC0 7F 0030 CLR CNCNT1
12525 *READ CNBIT2 AND INCREMENT CNCNT2 IF PRESENT
12530 OCC3 59 BYPCN1 ROL B
12535 OCC4 2A 1A BPL BYPCN2
12540 OCC6 7C 0032 INC CNCNT2
12545 *TEST CNCNT2 AND CREDIT=CREDIT+GAMES2
12550 *IF CNCNT2=COINS2
12555 OCC9 86 38 LDA A #%00111000 MASK FOR COINS2
12560 OCCB 94 41 AND A XEDGE+6
12561 LSR A
12562 OCCE 44 LSR A
12563 OCCF 44 LSR A
12565 OCD0 91 32 CMP A CNCNT2 BRANCH IF CNCNT2><COIN1
12566 OCD2 26 0C BNE BYPCN2
12570 OCD4 86 07 LDA A #%00000111 MASK FOR GAMES2
12575 OCD6 94 41 AND A XEDGE+6
12580 OCD8 9B 4E ADD A CREDIT
12585 OCDA 19 DAA
12587 OCDB 97 4E STA A CREDIT
12590 *RESET CHUTE2 COIN COUNTER
12595 OCDD 7F 0032 CLR CNCNT2
12597 *CLEAR CNBIT1,CNBIT2,AND RESET FROM XEDGE(3)
12598 *SINCE THIS DATA NOT NEEDED.
12599 OCE0 7F 003E BYPCN2 CLR XEDGE+3
12600 *MONITOR GAME RESET AND INCREMENT NPLAY AS REQD.
12605 *RESET AND CREDIT OK?
12610 OCE3 59 ROL B
12615 OCE4 2A 35 BPL ENDCNM BRANCH IF RESET<>1
12620 OCE6 7D 004E TST CREDIT
12625 OCE9 27 30 BEQ ENDCNM

```

```

12645
12650
12655 OCFB
12660 OCEB
12665 OCF0
12670 OCF3
12675
12680 OCF5
12685
12690
12695 OCF8 96 21
12700 OCFA 81 02
12705 OCFC 2A 1D
12710
12715 OCFE 96 23
12720 OD00 81 04
12725 OD02 27 17
12730
12735 OD04 7C 0023
12736 OD07 BD 0EDC
12738 OD0C BD 0AD0
12740 OD0F 86 99
12742 OD0F 86 99
12743 OD13 19
12744 OD14 97 4E
12745 OD16 36 77
12755 OD18 BD 0C7D
12760 OD18 39

*TEST FOR RESET RECEIVED AFTER SECOND BALL
*STARTED AND BRANCH TO NEWGAM IF SO.
SMEGME LDA A BIP
      CMP A #2
      BPL GONWGM
*TEST THAT NPLAY < 4 AND BYPASS IF NOT
      LDA A NPLAY
      CMP A #4
      BEQ ENDCNM
*INC NPLAY, CHIME1--, CREDIT, RESET GM.OVR.LMP
      INC NPLAY
      JSR LIGHTS LDX #PIA2DB CHIME 100
      JSR SLNPLS
      LDA A #$99
      ADD A CREDIT
      DAA
      STA A CREDIT
      LDA A #GMVRO
      JSR LPCTRL
      ENDCNM RTS
      GONWGM TSX
      DEX
      LDX

12737 OD0A 86 05 LDA A #5

12770
12775
12780
12785
12790
12800 OD1C D7 1D
12801 OD1E 36
12802
12803 OD20 84 07
12806
12812
12815
12818
12821 OD22 C6 01
12824 OD24 4C
12827 OD25 4A
12830 OD26 27 06
12833 OD28 58
12836 OD29 78 001D
12839 OD2C 20 F7
12842
12845
12848
12851
12852 OD2E 53
12854 OD2F E4 00
12857
12860
12863 OD31 DA 1D
12866
12869 OD33 E7 00
12872

*NAM BITWRT
*THIS ROUTINE WRITES A BIT TO MEMORY WITHOUT ALTER
*ADJACENT BITS. INPUT DATA IS WORD ADDRESS(X)
*BIT POSITION (A), AND NEW BIT VALUE(B)
*X,B, AND A ARE UNCHANGED AFTER EXEC.
BITWRT STA B TEMP STORE NEW BIT VALUE
      PSH A
      PSH B
      AND A #%00000111 MASK OFF ANY SUPERFLOWS
*BITS IN A (BIT POSITION)
*GENERATE MASK TO SET SELECTED BIT IN ORIGINAL
*WORD TO ZERO. SIMULTANEOUSLY SHIFT NEW BIT VALUE
*(0 OR 1) IN TEMP TO SELECTED BIT POSITION.
      LDA B #%00000001
      INC A INC.BIT POSITION FOR LOOPIN
      MSKLP DEC A DEC. BIT POSITION INDICATOR
      BEQ MKSET SKIP SHIFT OPERATION IF BIT
      ASL B SHIFT MASK BIT
      ASL TEMP SHIFT POSITION OF NEW BIT V
      BRA MSKLP RETURN TO MSKLP TO DEC. POS
*
      COUNTER & TEST IF NEXT SHIFT REQ.
*MASK AND NEW BIT IN CORRECT POSITION
*COMP. MASK AND SET SELECTED BIT POS. IN SEL.
*WORD TO ZERO.
      MSKSET COM B
      AND B 0,X
*SET SELECTED BIT TO THE VALUR SPECIFIED BY THE
*SHIFTED TEMP BIT.
      ORA B TEMP
*WRITE MODIFIED WORD BACK TO MEM.
      STA B 0,X
*RESTORE BIT POSITION TO A.

```

```

12874 0D35 33      PUL B
12875 0D36 32      PUL A
12880 0D37 39      RTS
12905              *NAM BITRD
12910              *SHIFT A SPECIFIED BIT(BIT) IN M<X>
12915              *TO THE CARRY FLAG POSITION.INPUTS REQUIRED
12920              *ARX BIT(0-7) IN B AND ADDR. TO BE READ
12925              *PRIOR TO EXECUTION, THE VALUE OF B IS ALTERED
12930              *BUT X AND A ARE UNCHANGED
12935 0D38 36      BITRD PSH A
12940              *
12845 0D39 A6 00      LDA A      0,X
12950 0D3B 44      SHFTLP LSR A
12955 0D3C 5A      DEC B
12960 0D3D 2A FC      BPL      SHFTLP
12965 0D3F 32      PUL A
12970 0D40 39      RTS
12980              *NAM BCD
12983              *THIS SUBR. ADDS TWO OPERANDS THAT ARE EACH IN BCD
12985              *AND CONSIST OF 3 BYTES. THIS FIRST OPERAND P HAS
12990              * STARTING ADDRESS PASSED THRU X WHILE THE SECOND
12995              *OPERAND Q HAS A STARTING ADDRESS OF ADDEND. BOTH
13000              *P&Q MUST BE DEFINED BEFORE EXECUTION AND THE RESU
13005              *P&Q IS RETURNED TO M<X>. THE LSB OF Q (UNITS) SHA
13010              *NORMALLY BE ZERO, HOWEVER A 2 OR 1 MAY BE ENTERED
13015              *IN THIS CASE THE ROUTINE IS FOR ESHORTENED TO A 2
13020              *1 BYTE ADDITION PROCESS. Q IS RESET TO ZERO AFTER
13025              *THE OPERATION.
13030              *THE OPERATION. THE ADDITION DOES NOT TAKE
13035              *PLACE IF TILT=1.
                  THE ROUTINE ALSO SETS BIT 0 OF OTGLFG TO ZERO
13040              *CHECK TILT AND SKIP
13045              *ADDITION PROCESS IF SET.
13050 0D41 7D 002C BCD TST TILT
13055 0D44 26 20      BNE RSTADD
12050              *LOAD THE UNITS HALF BYTE OF THE LSB OF Q TO B TO
13065              *CHECK FOR THE FORESHORTENED 2 OR 1 BYTE ADDITION
13070              *OPTION
13075 0D46 C6 0F      LDA B      #%00001111
13080 0D48 D4 29      AND B      ADDEND
13085              *CLEAR C AND ADD FIRST BYTE
13090 0D4A 0C      CLC
13095 0D4B 96 29      LDA A      ADDEND
13100 0D4D AB 00      ADD A      0,X
13105 0D4F 19      DAA
13110 0D50 A7 00      STA A      0,X
13115              *CHECK FOR ADDITON OPTION
13120 0D52 5A      DEC B
13125 0D53 27 11      BEQ RSTADD
13130              *ADD SECOND BYTE
13135 0D55 96 2A      LDA A      ADDEND+1
13140 0D57 A9 01      ADC A      1,X
13145 0D59 19      DAA
13150 0D5A A7 01      STA A      1,X
13155              *CHECK FOR ADDITION OPTION
13160 0D5C 5A      DEC B
13165 0D5D 27 07      BEQ RSTADD
13170              *ADD THIRD BYTE
13175 0D5F 96 2B      LDA A      ADDEND+2
13180 0D61 A9 02      ADC A      2,X
13185 0D63 19      DAA
13190 0D64 A7 02      STA A      2,X
13195              *RESET Q TO ZERO
13200 0D66 CE 0003 RSTADD LDX      #3
13205 0D69 6F 28      LOOP2 CLR      ADDEND-1,X
13210 0D6B 09      DEX

```

```

13215 0D6C 26 FB      BNE
                      *RESET BIT 0 OF OTHLFG. SET BY MAIN
                      LDA A      OTHLFG
                      AND A      #$FE
                      STA A      OTHLFG
                      RTS
13220 0D6E 39          *NAM WRTLMP
13225                  *THIS SUB. WRITES THE LAMP MATRIX LMPMAT TO
13230                  *PIA2. THE OUTPUT DATA IS FORMATTED FOR
13235                  *USE BY 8 CHANNEL DEMULTIPLEXERS.
13240                  *
13245                  *
13250                  *
13255                  *
13260                  *INITIALIZE X (LMPMAT OFFSET) AND DECADD (DECODER
13265                  *ADDRESS DATA)
13270 0D6F CE 0007 WRTLMP LDX      #7
13275 0D72 86 07      LDA A      #7
13280 0D74 97 31      STA A      DECADD
13285                  *PREPARE DRIVER ADDRESS DATA SUCH THAT PIA2DB3
13290                  *-B7 ARE NOT CHANGED WHEN ADDRESS DATA IS WRITTEN
13295                  *TO B0-B2.
13300 0D76 C6 F8      OUTLP LDA B      #%11111000
13305 0D78 D4 8A      AND B      PIA2DB
13310 0D7A DA 31      ORA B      DECADD B CONTAINS PREPARED ADDR.DA
13315                  *PREPARE COMPLEMENTED LAMP DATA WORD FROM LMPMAT
13320 0D7C A6 15      LDA A      LMPMAT,X
13325 0D7E 43      COM A      A CONTAINS COMP. LAMP DATA
13330 0D7F 36      PSN A      STORE
13335                  *DISABLE ALL LMP. DRIVERS IN PREP. FOR NEW
13340                  *ADDR DATA
13345 0D80 86 FF      LDA A      #$FF
13350 0D82 97 88      STA A      PIA2DA
13355                  *WRITE NEW DECODER ADDRESS DATA PREPARED ABOVE.
13360 0D84 D7 8A      STA B      PIA2DB
13365                  *WRITE NEW COMPLEMENTED LMP.DATA.
13370 0D86 33      PUL B
13375 0D87 D7 88      STA B      PIA2DA
13380                  *DECREMENT LMPMAT OFFSET AND DRIVER ADDRESS DATA
13385 0D89 09      DEX
13390 0D8A 7A 0031    DEC      DECADD
13395                  *RETURN TO OUTLP IF NOT LAST DRIVER ADDR. ELSE
13400                  *RETURN
13405 0D8D 2A E7      BPL      OUTLP
13410 0D8F 39      RTS
13420                  *NAM NEWDGT
13425                  *
13430                  *THIS INTERRUPT ROUTINE IS ACTIVATED BY THE 360
13435                  *HZ.DISPLAY TIMER ON PIA3CA1. THE ROUTINE SENDS
13440                  *BCD DATA REPRESENTING THE DESIRED DISPLAY
13445                  *CHARACTER TO EACH OF FIVE DISPLAYS. DISPLAYS
13450                  *1-4 CORRESPOND TO THE PLAYERS SCORES.
13455                  *DISPLAY 5 RECEIVES CREDIT DATA FOR
13460                  *DIGITS 1&2 AND MATCH DATA FOR 5&6. THE FIRST
13465                  *TIME THE PROGRAM IS EXECUTED,FIVE BYTES OF BCD
13470                  *DATA ARE FED TO THE FIVE DIFFERENT DISPLAY
13475                  *MODULES.THE BCD DATA IS FOR DIGIT 6 OF EACH
13480                  *DISPLAY.ON FOLLOWING EXECUTIONS THE BCD DATA
13485                  *IS FOR DIGIT 5,4,3 ECT.AFTER THE EXECUTION
13490                  *THAT TRANSMITS THE DATA FOR DIGIT 1 THE DIGIT
13495                  *COUNTER IS SET TO 6.NOTE THAT INITIALIZATION
13500                  *OF THE ROUTINE IS ACCOMPLISHED AS THE SYSTEM
13505                  *COMES FROM RESET.XSTART&SHFTG=0,
13510                  *XSTART+1=14 AND DGTADD=6.
13515                  *
13520                  *
13525                  *OUTPUT DATA FOR CURRENT DIGIT ADDRESS TO EACH

```

```

13530      *OF FIVE DISPLAY ADDRESSES BEGINNING WITH DISP.
13535      *ADDR.=5.DGTADD=DIGIT ADDR.
13540      *
13545  OD90 C6 05  NEWDGT LDA B      #5      INIT.DISPLAY NO. COUNTER
13550  OD92 DE 51      LDX      XSTART  LOAD OFFSET FROM<SCORE1>
13555                                OF THE RAM WORD.
13560      *THIS SECTION OBTAINS BCD DATA FROM SCORE1,2
13565      *3,4,MATCH,AND CREDIT FOR TRANSMISSION
13570      *TO THE DISPLAYS. INPUTS ARE THE NO. OF THE
13575      *DISP.(B), THE DIGIT NO.(DGTADD) FOR WHICH THE
13580      *DATA IS REQUIRED AND THE DISTANCE FROM
13585      *THE BASE ADDRESS<SCORE1> OF THE WORD CONTAIN.
13590      *THE DATA(X). THE HALF BYTE OUTPUT CONTAINING
13595      *THE BCD DATA IS PLACED IN THE M.S. HALF BYTE
13600      *OF A WITH ZERO IN THE LOWER ORDER FOUR BITS.
13605      *MEMORY MUST BE ARRANGED SEQUENTIALLY WITH
13610      *SCORE1 AT THE LOWEST NEMERICAL ADDRESS AND
13615      *SCORE4+2 LOCATED AT SCORE1+11, MATCH MUST
13620      * BE LOCATED AT SCORE1+12 AND CREDIT AT SCORE1+14
13625      *
13630      *
13635  OD94 A6 42  DSPLP LDA A      SCORE1,X    READ RAM WORD CONT. BCD D
13640      *TEST SHFTFG TO DETERMINE IF DESIRED BCD
13645      *DATA IS IN LOWER ORDER BITS OF RAM WORD
13650      *SHFTFG INDICATES WHETHER THE CURRENT DIGIT
13655      *ADDRESS IS ODD(BCD DATA IN LOWER ORDER BITS)
13660      *OR EVEN (DATA IN HIGHER ORDER BITS).
13665      *
13670  OD96 7D 0053      TST      SHFTFG
13675      *SHIFT RAM WORD LEFT IF DATA IN LOWER ORDER
13680      *BITS,ELSE MASK OFF LOWER ORDER BITS.
13685  OD99 27 04      BEQ      SKPASL
13690  OD9B 48      ASL A
13695  OD9C 48      ASL A
13700  OD9D 48      ASL A
13705  OD9E 48      ASL A
13710  OD9F 84 F0  SKPASL AND A      #%11110000
13715      *BCD DATA NOW LOCATED IN HIGHER ORDER BITS
13720      *OF A.LOWER ORDER 4 BITS ARE ZERO
13725      *OUTPUT DISPLAY NO. TO L.E. DECODER(PIA3DO-3)
13730  ODA1 D7 90      STA B      PIA3DA
13735      *OUTPUT BCD DATA TO DISPLAYS (PIA3DA4-7)
13740  ODA3 9A 90      ORA A      PIA3DA
13745  ODA5 97 90      STA A      PIA3DA
13746  ODA7 84 F0      AND A      #%11110000
13747  ODA9 97 90      STA      PIA3DA
13750  ODAB 09      DEX
13755      *          DEC.COUNTERS IN PREPARATION
13760  ODAC 09      DEX          FOR NEXT DISPLAY ADDR.
13765  ODAD 09      DEX
13770  ODAE 5A      DEC B
13775  ODAF 26 E3      BNE      DSPLP      B HOLDS THE DISPLAY ADDR.
13780                                CONTINUE LOOPING UNTIL ALL
13785                                DISPLAYS ARE LOADED
13790      *
13795      *ALL 5 DISPLAYS ARE LOADED WITH BCD INFO.
13800      *SEND CODE FOR BLANKING PULSE TO LATCH ENABLE
13805      *DECODER (OUTPUT LEAD,7)VIA PIA3DAO-3.
13810  ODB1 86 07      LDA A      #7
13815  ODB3 9A 90      ORA A      PIA3DA
13820  ODB5 97 90      STA A      PIA3DA
13825      *OUTPUT DIGIT ENABLE CODE TO DIGIT
13830      *SELECT DECODER VIA PIA3DAO-3.
13835      *
13840  ODB7 86 0F      LDA A      #$0F
13841  ODB9 97 90      STA A      PIA3DA

```

```

13842 ODBB 86 08      LDA A      #8
13845 ODBD 9B 4F      ADD A      DGTADD  DIGIT ADDR.CODE FORMED
13850 ODBF 97 90      STA A      PIA3DA
13855
13860      *
13865      *DO HOUSEKEEPING IN PREPARATION FOR THE NEXT
13870 ODC1 96 53      LDA A      SHFTFG
13875 ODC3 27 03      BEQ      BYPDEC
13880 ODC5 7A 0052     DEC      XSTART+1
13885 ODC8 73 0053     BYPDEC COM  SHFTFG
13890 ODCB 7A 004F     DEC      DGTADD
13895
13900      *IF EACH OF THE SIX DIGITS HAS BEEN SCANNED
13905      *THEN RESET TO DIGIT NO.6.
13905 ODCE 96 4F      LDA A      DGTADD
13910 ODD0 26 08      BNE      BYPRES
13915 ODD2 86 06      LDA A      #6
13920 ODD4 97 4F      STA A      DGTADD
13925 ODD6 86 0E      LDA A      #14
13930 ODD8 97 52      STA A      XSTART+1  RESET OFFSET FRM.SCORE1
13935 ODDA 3B
14000      BYPRES RTI
14005      *NAM SCOREX
14010      *SCOREX IS A SUBROUTINE WHICH IS CALLED
14015      *TO ADD THE DESIGNATED TARGET OR BONUS
14020      *VALUE TO THE CURRENT PLAYER'S CORE.
14025      *THE ROUTINE SORTS THE TARGET OR BONUS
14030      *VALUE INTO INCREMENTAL MULTIPLES OF
14035      *TEN, ONE-HUNDRED, & ONE-THOUSAND.
14040      *CONCURRENTLY, THE APPROPRIATE CHIME
14045      *IS SOUNDED WITH EACH INCREMENT.
14050      *ARGUMENTS FOR THE ROUTINE:
14055      *THE APPROPRIATE SCORING CODEWORD IS
14060      *LOADED INTO ACCUM "A" JUST PRIOR
14065      *TO THE CALL. THE CODEWORD CONSISTS
14070      *OF A BCD# IN THE LOW ORDER 1/2 BYTE,
14075      *WHICH CORRESPONDS TO THE INCREMENTAL MULTIPLIER.
14080      *BIT#7=1 INDICATES THOUSANDS;
14085      *BIT#6=1 INDICATES HUNDREDS;
14090      *BITS 6&7 INDICATES TENS.
14095      *SCOREX USES RAM VARIABLES "QLOOP","CHIME",
14100      *"SCRWRD", AND A ONE BYTE CONSTANT "GETLO".
14105      *SCOREX CALLS "DELAY","SLNPLS","SCRPLS",
14110      *ALSO USES RAM VAR "ADDEND"
14115
14120 ODDB 0F      GET LO FCB      $0F
14125 ODDC 7D 002C  SCOREX TST      TILT
14130 ODDF 26 3F      BNE      KAREN
14135 ODE1 16
14140 ODE2 F4 ODDB  AND B      GETLO  SAVE CODEWRD IN B
14145 ODE5 D7 08      STA B      QLOOP  PICK OFF LO-ORD MULTIPLIER
14150 ODE7 49      ROL A
14155 ODE8 25 11      BCS      N1000  C=1 MEANS A K
14160 ODEA 49      ROL A
14165 ODEB 25 07      BCS      N100  C=1 MEANS A 100
14170 ODED 86 04      N10  LDA A      #$04  SETUP FOR 10'S CHIME
14175 ODEF CE 1000  LDX      #$1000  PREP FOR SCRPLS
14180 ODF2 20 0C      BRA      VICKI  DO IT
14185 ODF4 86 05      N100  LDA A      #$05  PREP4 CHIME
14190 ODF6 CF 0001  LDX      #$0001  PREP4 SCRPLS
14195 ODF9 20 05      BRA      VICKI
14200 ODFB 86 06      N1000  LDA A      #$06  PREP4 CHIME
14205 ODFD CE 0010  LDX      #$0010  PREP4 SCRPLS
14210 OE00 97 04      VICKI  STA A      CHIME  SAVE CHIME BIT #
14215 OE02 DF 05      STX      SCRWRD  SAVE CURRENT SCOREWORD
14220 OE04 96 04      SUSIE  LDA A      CHIME  PREP4 SLNPLS BIT#
14225 OE06 CE 008A  LDX      #PIA2DB M IN X
14230 OE09 BD 0A00  JSR      SLNPLS
14235 OE0C DE 05      LDX      SCRWRD  PREP4 SCRPLS

```

```

15230 0E0E DF 29      STX      ADDEND
15235 0E10 BD 0AFE      JSR      SCRPLS  INCREMENT SCORE
15240 0E13 7A 0008      DEC      QLOOP
15245 0E16 27 07        BEQ      KAREN   QLOOP=0 MEANS DONE
15250 0E18 C6 10        LDA B    #$10    NOT ZERO MEANS DELAY; RETRY
15255 0E1A BD 0C9E      JSR      DELAY
15260 0E1D 20 E5        BRA      SUSIE   DONE WAITING; DO MORE
15265 0E1F 39          KAREN RTS      THATS ALL FOLKS
15505                  INAM BONES
15510                  *BONES IS A SUBROUTINE WHICH INCREMENTALLY ADDS
15515                  *THE BONUS POINTS TO THE CURRENT PLAYER'S SCORE
15520                  *THIS ROUTINE IS CALLED AFTER THE BALL EXITS.
15525                  *ARGUMENTS FOR THE ROUTINE ARE:
15530                  *****
15535                  *BONUS+1, BONUS+2, AND BMULT
15540                  *BONUS+2 IS A 1-BYTE RAM VARIABLE CONTAINING
15545                  ***THE BCD# OF 10K'S IN THE LOW-ORDER BITS
15550                  *"BONUS+1" IS A 1-BYTE RAM VARIABLE WHICH CONTAINS
15555                  ***THE BCD# OF 1K'S IN THE HI-ORDER, AND THE BCD#
15560                  ***OF 100'S IN THE LO-ORDER BITS
15565                  *"BMULT" IS A 1-BYTE RAM VARIABLE WHICH CONTAINS
15570                  ***THE CURRENT BONUS MULTIPLIER IN THE LO-ORDER.
15575                  *THIS ROUTINE CALLS "SCOREX", WHICH DOES THE
15580                  *SCORING AND CHIMING.
15585                  *UPON EXIT, THE ROUTINE INITIALIZES THE BONUS
15590                  *TO "BMULT" =1, NAD 1K OF BONUS
15595 0E20 86 F0        BONES LDA A    #$F0    SETUP BIT MASK
15600 0E22 94 26        AND A      BONUS+1 GET # OF 1K'S IN HI-ORDER
15605 0E24 27 24        BEQ      LPASO
15610 0E26 D6 13        ALPHA LDA B    BMULT   LOAD PAS COUNTER=BMULT
15615 0E28 D7 14        STA B      PCOUNT
15620 0E2A C6 10        LDA B      #$10    PREP FOR DECREMENT
15625 0E2C 10          SBA          DECREMENT # OF 1K'S
15630 0E2D 97 26        STA A      BONUS+1 RESTORE TO RAM
15633 0E2F BD 0E7C      JSR      BONDSP GO UPDATE THE DISPLAY
15635 0E32 86 81        QAGAIN LDA A    #$81   PREP4 SCOREX AT 1K
15645 0E34 BD 0DDC      JSR      SCOREX
15650 0E37 7A 0014      DEC      PCOUNT DECREMENT PASSCOUNT
15651 0E3A 27 07        BEQ      HOWIE   GO SETUP LONG DELAY
15652 0E3C C6 10        LDA B      #$10    DO SHORT DELAY
15653 0E3E BD 0C9E      JSR      DELAY
15654 0E41 20 EF        BRA      QAGAIN DO MORE
15655 0E43 C6 25        HOWIE LDA B    #$25   LONG DELAY
15665 0E45 BD 0C9E      JSR      DELAY GO DO IT
15670 0E48 20 D6        BRA      BONES GO BACK AND CHECK FOR MORE
15675 0E4A 96 27        LPASO LDA A    BONUS+2 GET # OF 10K'S
15680 0E4C 26 12        BNE      TOMB
15685 0E4E CE 1000      LDX      #$1000 PREP4 INTIAL BONUS
15690 0E51 DF 26        STX      BONUS+1 DO IT
15695 0E53 86 01        LDA A      #$01   PREP FOR INITIAL BMULT
15700 0E55 97 13        STA A      BMULT   DO IT
15702 0E57 BD 0E7C      JSR      BONDSP UPDATE IT
15703 0E5A 86 22        LDA A      #DBLBNO
15704 0E5C BD 0C7D      JSR      LPCTRL
15705 0E5F 39          RTS          RETURN FROM SUB
15710 0E60 7A 0027      TOMB DEC      BONUS+2 BORROW A 10K
15715 0E63 86 A0        LDA A      #$A0    SET UP 1K'S
15720 0E65 97 26        STA A      BONUS+1 RESTORE
15725 0E67 20 B7        BRA      BONES DO SOME MORE
16005                  *NAM BONDSP BONUS DISPLAY
16010                  *BONDSP IS A ROUTINE WHICH EXAMINES THE CONTENTS 0
16015                  *"BONUS+1" & "BONUS+2", AND MAKES THE APPROPRIATE
16020                  *BIT CHANGES IN LAMP MATRIX. BONDSP SHOULD BE
16025                  *CALLED WHENEVER A BONUS SCORING UPDATE OCCURS.
16030                  ****TABLE ASSIGNMENTS

```

```

16035 0E69 23      TABE2  FCB      $23,$24,$25,$26,$27,$30,$31,$32,$33,
      0E6A 24
      0E6B 25
      0E6C 26
      0E6D 27
      0E6E 30
      0E6F 31
      0E70 32
      0E71 33
      0E72 B4
16040 0E73 10      TABE3  FCB      $10,$11,$12,$13,$14,$15,$16,$17,$A0
      0E74 11
      0E75 12
      0E76 13
      0E77 14
      0E78 15
      0E79 16
      0E7A 17
      0E7B A0

16045      ****
16050      *DO BONUS 100'S (TABE2)
16055 0E7C CE 0E69 BONDSP LDX      #TABE2  PREP4 ABSUB THIS TABLE
16060 0E7F C6 0F      LDA B      #$0F    GET # OF 100'S FROM BONUS+1
16065 0E81 D4 26      AND B      BONUS+1 100'S IN B; PREP4 TABSUB
16067 0E83 5C      INC B
16070 0E84 BD 0EA6    JSR      TABSUB  DO IT
16075      *DO BONUS 1K'S (TABE3)
16080 0E87 96 26      LDA A      BONUS+1 GET # OF 1K'S
16085 0E89 BD 0EA1    JSR      GETHI   IN "B"
16090 0E8C 16      TAB      PREP4 TABSUB
16095 0E8D CE 0E73    LDX      #TABE3  PREP4 TABSUB THIS TABLE
16100 0E90 BD 0EA6    JSR      TABSUB  DO IT
16105      *DO 10K'S OF BONUS
16110 0E93 96 27      LDA A      BONUS+2 GET # OF 10K'S IN BONUS
16115 0E95 27 04      BEQ      EX32   IF ZERO, TURN OFF
16120 0E97 86 A1      LDA A      #$A1   NOT ZERO, TURN ON
16125 0E99 20 02      BRA      EX32A
16130 0E9B 86 21      EX32  LDA A      #$21   OFF
16135 0E9D BD 0C7D    EX32A JSR      LPCTRL  DOIT
16140 0EA0 39      RTS
16334      *NAM GETHI
16335      *GETHI IS A SUBROUTINE WHICH TAKES THE HIGH-ORDER
16340      *BITS OF "A" AND SHIFTS THEM TO LOW-ORDER, INSERTI
16345      *HI-ORDER ZEROS.  RESULT STORED IN "A".
16350 0EA1 44      GETHI  LSR A
16355 0EA2 44      LSR A
16360 0EA3 44      LSR A
16365 0EA4 44      LSR A
16370 0EA5 39      RTS
16405      *NAM TABSUB
16410      *TABSUB IS A ROUTINE WHICH TAKES A NUMBER IN "B"
16415      *AND A TABLE REFERENCED BY X, AND TURNS ON
16420      *THE LAMP ASSOCIATED WITH B & X, WHILE TURNING
16425      *OFF ALL OTHER LAMPS IN THIS TABLE. "X" CONTAINS
16430      *THE STARTING ADDRESS OF THE CURRENT TABLE;
16435      *"B" CONTAINS THE NTH ENTRY TO BE SWITCHED ON
16440      *THE LAST WORD OF THE CURRENT TABLE MUST HAVE
16445      *BIT 7=1; ALL OTHER ENTRIES BIT 7=0.  THE BALANCE
16450      *OF THE WORDS CONTAINS THE WORD# &BIT# THIS CODEWO
16455 0EA6 DF 11      TABSUB STX      ADDOFF  SAVE START ADDRESS
16460 0EA8 D7 0B      STA B      DESIRD  SAVE DESIRED ENTRY POSITION
16465 0EAA C6 01      LDA B      #$01   START WITH ENTRY 1
16470 0EAC D7 0A      STA B      CURENT  SAVE IT (CURENT ENTRY)
16475 0EAE D6 0B      KPASA  LDA B      DESIRD  COMPARE DESIRED ENTRY
16480 0EB0 D1 0A      CMP B      CURENT  WITH CURENT ENTRY
16485 0EB2 27 06      BEQ      GTON   SAME, THEN GO TURN ON

```


4,198,051

63

64

```

16490 OEB4 86 7F  GTOFF  LDA A  #$7F  MASK OFF BIT 7=0
16495 OEB6 A4 00      AND A  X      FOR THIS CODEWORD*
16500 OEB8 20 04      BRA    EX7A  DO LPCTRL
16505 OEBA 86 80  GTON   LDA A  #$80  PUT A 1 IN BIT 7
16510 OEBC AA 00      ORA A  X      FOR THIS CODEWORD ON
16515 OEBE BD 0C7D EX7A  JSR    LPCTRL GO CHANGE MATRIX
16520 OEC1 DE 11      LDX    ADDOFF EXAMINE CURENT CODEWORD
16525 OEC3 E6 00      LDA B  X      FOR END OF TABLE
16530 OEC5 2B 08      BMI    GUDBYE MINUS MEANS DONE
16535 OEC7 08      INX      SET UP NXT CODEWRD  ADDRESS
16540 OEC8 DF 11      STX    ADDOFF IN ADDOFF
16545 OECA 7C 000A    INC    CURENT PREP4 NXT ENTRY CHECK
16550 OECF 20 DF      BRA    KPASA  DO NEXT ENTRY
16555 OECF 39      GUDBYE RTS
17140      *NAM LIGHTS
17145      *ACTIVATE THE CORRECT BITS IN LMPMAT TO ILLUMINAT
17150      *PLAYERS AND NO.OF PLAYER UP,AND BALL IN PLAY NO
17155      *
17160      *
17165      *ILLUMINATE THE NUMBER OF PLAYERS BASED UPON THE
17170      *LOAD TABLE OF LAMP CODEWORDS
17175 OED0 44      TNPLAY FCB  PLY1NO,PLY2NO,PLY3NO,PLY4N1
      OED1 45
      OED2 46
      OED3 C7
17177 OED4 40      TNUP   FCB    UP1LP0,UP2LP0,UP3LP0,UP4LP1
      OED5 41
      OED6 42
      OED7 C3
17178 OED8 62      TBIP   FCB    BIPL10,BIPL20,BIPL30,BIPL40,BIPL51
      OED9 63
      OEDA 64
      OEDB E5
17180      *LOAD INPUT PARAMETERS FOR TABSUB
17190 OEDC D6 23  LIGHTS LDA B  NPLAY LOAD NO. OF ON LIGHT.
17195 OEDE CE OED0    LDX    #TNPLAY STARTING ADDRESS
17200 OEE1 BD OEA6    JSR    TABSUB LIGHT LAMPS
17205      *
17210      *ILLUMINATE NO. OF PLAYER UP
17225      *LOAD INPUT PARAMETERS FOR TABSUB
17235 OEE4 D6 22      LDA B  NUP    NO.OF PLAYER THAT IS UP
17240 OEE6 CE OED4    LDX    #TNUP  STARTING ADDRESS
17245 OEE9 BD OEA6    JSR    TABSUB
17250      *LIGHT BALL IN PLAY LAMP
17260      *LOAD INPUT VALUES
17270 OEEC D6 21      LDA B  BIP
17275 OEEE CE OED8    LDX    #TBIP
17280 OEF1 BD OEA6    JSR    TABSUB
17285 OEF4 39      RTS
17295      *NAM SLAM
17300      *INHIBIT FLIPPERS,THUMPER BUMPER,SLINGSHOT
17305      *INHIBIT SCORE & BONUS COLLECTION
17310      *LIGHT GAME OVER LAMP
17315      *DELAY 30 SEC. AND RESTART AT NEWGAM
17320      *
17325 OEF5 BD 0922  SLAM   JSR    RTILT
17330 OEF8 86 F7      LDA A  #GMOVR1
17335 OEFA BD 0C7D    JSR    LPCTRL
17340 OEFD 20 F6      BRA    SLAM
17345 OEFF 39      RTS
17475 OF00 CE 004B  TASK4 LDX    #SCORE4
17477 OF03 8D 10      BSR    DOIT
17480 OF05 CE 0048  TASK3 LDX    #SCORE3
17482 OF08 8D 0B      BSR    DOIT
17490 OF0A CE 0045  TASK2 LDX    #SCORE2
17492 OF0D 8D 06      BSR    DOIT

```

```

17495 0F0F CE 0042 TASK1 LDX      #SCORE1
17497 0F12 8D 01      BSR      DOIT
17498 0F14 39      RTS
17499 0F15 24 06      DOIT     BCC      DBLANK
17500 0F17 86 00      BLANK    LDA A    #$00
17505 0F19 C6 FF      LDA B    #$FF
17510 0F1B 20 04      BRA      SANDR    SANDR SEARCHES AND REPLACES
17515 0F1D 86 FF      DBLANK   LDA A    #$FF
17520 0F1F C6 00      LDA B    #$00
17525 0F21 97 0D      SANDR    STA A    SEARCH    SAVE SEARCH WORD
17530 0F23 D7 0E      STA B    REPLCE   SAVE REPLACE WORD
17535 0F25 08      INX      SETUP CURRENT SCORE FOR MS
17540 0F26 08      INX
17545 0F27 86 02      LDA A    #$02    DO ONLY 2 BYTES (MS 4 DIGIT
17550 0F29 97 0F      STA A    DISPLY   SAVE IT
17555 0F2B 86 F0      HIORD    LDA A    #$F0    SEE IF HIORDER OF SEARCH =
17560 0F2D 16      TAB      HIORDER OF THIS WORD. NOT =
17565 0F2E A4 00      AND A    X        MEANS NO ACTION: SO EXIT
17570 0F30 D4 0D      AND B    SEARCH
17575 0F32 11      CBA
17580 0F33 26 21      BNE      ADIOS
17585 0F35 A6 00      REPHI    LDA A    X        SAME: REPLACE HIORDER THIS
17590 0F37 84 0F      AND A    #$0F      WITH REPLACEMENT VALUE
17595 0F39 D6 0E      LDA B    REPLCE
17600 0F3B C4 F0      AND B    #$F0
17605 0F3D 1B      ABA
17610 0F3E A7 00      STA A    X
17615 0F40 86 0F      LOORD    LDA A    #$0F    NOW CHECK LOORDER THIS WORD
17620 0F42 16      TAB      WITH LOORDER SEARCH WORD
17625 0F43 A4 00      AND A    X        NOT = MEANS EXIT
17630 0F45 D4 0D      AND B    SEARCH
17635 0F47 11      CBA
17640 0F48 26 0C      BNE      ADIOS
17645 0F4A 96 0E      REPLO    LDA A    REPLCE   SAME: REPLACE HI & LO WITH
17650 0F4C A7 00      STA A    X        REPLACEMENT WORD.
17655 0F4E 7A 000F      DEC      DISPLY
17660 0F51 27 03      BEQ      ADIOS
17665 0F53 09      DEX
17670 0F54 20 D5      BRA      HIORD    GO BACK FOR MORE IF NOT DON
17675 0F56 39      ADIOS    RTS
20001      * B A C K
20002      * INTERRUPT TARGETS
20003      *
20005      *
20020      *
20055      *
20060      *
20065      *
20070      * HARDWARE INTERRUPT SERVICE ROUTINE-THIS ROUTINE
20075      * IDENTIFIES THE ACTIVE INTERRUPT PORT AND TRANSFER
20080      * CONTROL TO AN APPROPRIATE ROUTINE. THE PORTS CHECK
20100 0F57 96 91      HWINT    LDA A    PIA3CA
20105 0F59 2A 05      BPL      Z1        CALL NEWDGT IF CRA7=1
20110 0F5B 96 90      LDA A    PIA3DA    CLEAR CRA7
20115 0F5D 7E 0D90      JMP      NEWDGT
20120 0F60 96 8B      Z1      LDA A    PIA2CB
20125 0F62 2A 06      BPL      Z2        CALL SLAM IF CRB7=1
20130 0F64 96 8A      LDA A    PIA2DB    CLEAR CRB6&7
20135 0F66 0E      CLI      PERMIT NEW INTERRUPTS
20140 0F67 7E 0EF5      JMP      SLAM
20145 0F6A 49      Z2      ROL A
20150 0F6B 2A 06      BPL      Z3        CALL ZEROOCR IF CRB6=1
20155 0F6D 96 8A      LDA A    PIA2DB    CLEAR CRB6&7
20160 0F6F 0E      CLI      PERMIT NEW INTERRUPTS
20165 0F70 7E 0A3D      JMP      ZEROOCR
20170 0F73 3B      Z3      RTI

```

```

20187      *
20200      *
20201      *
20210      *INTERRUPT VECTORS
20215      *   SOFTWARE INTERRUPT VECTOR
20220 OFFA      ORG      $OFFA
20230      *   HARDWARE INTERRUPT VECTOR
20235 OFF8      ORG      $OFF8
20240 OFF8 OF57 FDB      HWINT
20245      *   RESTART INTERRUPT VECTOR
20250 OFFE      ORG      $OFFE
20255 OFFE 0800 FDB      BEGROM
20256      END

```

SYMBOL TABLE

ADDEND	0029	ADDOFF	0011	ADDR	0002	ADDR2	0000	ADIOS	OF56
AGAIN	09F6	ALPHA	0E26	BCD	0D41	BEGROM	0800	BEGSTK	005A
BIP	0021	BIPL10	0062	BIPL11	00E2	BIPL20	0063	BIPL21	00E3
BIPL30	0064	BIPL31	00E4	BIPL40	0067	BIPL41	00E5	BIPL50	0066
BIPL51	00E6	BITLP1	0AA9	BITNUM	0024	BITRD	0D38	BITTBL	0B24
BITWRT	0D1C	BL010	0023	BL011	00A3	BL01K0	0010	BL01K1	0090
BL020	0024	BL021	00A4	BL02K0	0011	BL02K1	0091	BL030	0025
BL031	00A5	BL03K0	0012	BL03K1	0092	BL040	0026	BL041	00A6
BL04K0	0013	BL04K1	0093	BL050	0027	BL051	00A7	BL05K0	0014
BL05K1	0094	BL060	0030	BL061	00B0	BL06K0	0015	BL06K1	0095
BL070	0031	BL071	00B1	BL07K0	0016	BL07K1	0096	BL080	0032
BL081	00B2	BL08K0	0017	BL08K1	0097	BL090	0033	BL091	00B3
BL09K0	0020	BL09K1	00A0	BL100	0034	BL101	00B4	BL10K0	0021
BL10K1	00A1	BLANK	0F17	BMULT	0013	BONDSP	0E7C	BONES	0E20
BONUS	0025	BYPAS1	0B0A	BYPAS2	0B10	BYPAS3	0B16	BYPAS5	0BE8
BYPEN1	0CC3	BYPEN2	0CE0	BYPDA	0A0D	BYPDEC	0DC8	BYPRES	0DDA
BYPSET	09CB	BYPSTL	090E	BYPSTL	0986	BYPSTL	0A88	BYPSTL	093D
CHAN1K	09B2	CHANNL	09BC	CHANNR	09B5	CHANNX	09BF	CHIME	0004
CNCNT1	0030	CNCNT2	0032	CNMN	0CA5	COINS	0879	CREDIT	004E
CURRENT	000A	DBLANK	0F1D	DBLBNO	0022	DBLBNI	00A2	DECADD	0031
DELAY	0C9E	DESIRD	000B	DGTADD	004F	DISPLY	000F	DOIT	0F15
DRLPA0	0000	DRLPA1	0080	DRLPB0	0001	DRLPB1	0081	DRLPC0	0002
DRLPC1	0082	DRLPDO	0003	DRLPD1	0083	DROPA	09D1	DROPB	09D7
DROPC	09DD	DROPD	09E3	DROPX	09E7	DRP100	0A14	DRP500	0A0E
DSPLP	0D94	END1	0C4D	END2	0C59	ENDBNS	0BCD	ENDCNM	0D1B
ENDLSL	09AC	ENDRSL	0999	ENDTH	0921	ENTRY	001E	EVEN	0BB7
EVENX	0BB2	EX32	0E9B	EX32A	0E9D	EX7A	0EBE	FAST	0AC6
FASTRT	0ACC	FGMCHK	0C31	FIXED	0B75	FIXLP	0B7A	GETHI	0EA1
GETLO	0DD8	GMOVRO	0077	GMOVRI	00F7	GNILLO	00E7	GNILL1	0067
GTOFF	0EB4	GTON	0EBA	GUDBYE	0ECF	HIORD	0F2B	HOWIE	0E43
HWINT	0F57	KAREN	0E1F	KPASA	0EAE	LESS15	0BA8	LIGHTS	0EDC
LMPMAT	0015	LOOP2	0D69	LOOPLP	0BDE	LOORD	0F40	LPASO	0E4A
LPCTRL	0C7D	LSLGST	099A	LSTXST	0037	LSTXWD	0033	LXBLPO	0035
LXBLP1	00B5	MATCH	0028	MATCHD	0050	MATCHX	0C0A	MOMLP	0B30
MONMOD	08C2	MSKLP	0D25	MSKSET	0D2E	MTCHLP	0C1A	MTLP00	0050
MTLP01	00D0	MTLP10	0051	MTLP11	00D1	MTLP20	0052	MTLP21	00D2
MTLP30	0053	MTLP31	00D3	MTLP40	0054	MTLP41	00D4	MTLP50	0055
MTLP51	00D5	MTLP60	0056	MTLP61	00D6	MTLP70	0057	MTLP71	00D7
MTLP80	0060	MTLP81	00E0	MTLP90	0061	MTLP91	00E1	N10	0DED
N100	0DF4	N1000	0DFB	NEWBAL	0896	NEWDGT	0D90	NEWGAM	085A
NEWPYR	089C	NEXT	0B1C	NOSCR	0889	NOSOLN	0B67	NPLAY	0023
NUP	0022	NXTPLY	0BD9	NXTSCR	0C6A	NXTTIM	0A51	ODD	0BC3
OTHLFG	002D	OUTHL	09AD	OUTLP	0D76	PCOUNT	0014	PIA1CA	0085
PIA1CB	0087	PIA1DA	0084	PIA1DB	0086	PIA2CA	0089	PIA2CB	008B
PIA2DA	0088	PIA2DB	008A	PIA3CA	0091	PIA3CB	0093	PIA3DA	0090
PIA3DB	0092	PLY1N0	0044	PLY1N1	00C4	PL2N0	0045	PLY2N1	00C5
PLY3N0	0046	PLY3N1	00C6	PLY4N0	0047	PLY4N1	00C7	PLYFLD	0010
QAGAIN	0E32	QLOOP	0008	QUICK	0A21	RAMZR	0BD1	RBONUS	0B8A
RDMAT	0B27	REBOUN	0964	REPEAT	000C	REPHI	0F35	REPLCE	000E

REPLO	0F4A	RESRAM	0048	RLVSW	0957	RLVSWA	0952	RLVSWB	094D
RLVSWC	0948	RLVSWD	0943	RLVSWX	0930	RSLGST	0987	RSTADD	0D66
RSTCN	0A4D	RTILT	0922	RVLPA0	0004	RVLPA1	0084	RVLPB0	0005
RVLPB1	0085	RVLPC0	0006	RVLPC1	0086	RVLPD0	0007	RVLPD1	0087
RXBLP0	0036	RXBLP1	00B6	SANDR	0F21	SCORE1	0042	SCORE2	0045
SCORE3	0048	SCORE4	004B	SCOREX	0DDC	SCRPLS	0AFE	SCRWRD	0005
SEARCH	000D	SERIN	0ADF	SERLP	0AEF	SHFTFG	0053	SHFTLP	0D3B
SHTAG	08A2	SKIP3	0ABA	SKPASL	0D9F	SLAM	0EF5	SLNPLS	0AD0
SMEGME	0CF8	SPCIAL	0971	SPCLP0	0074	SPCLP1	00F4	STAGNO	0075
STAGN1	00F5	SUBTBL	0A5E	SUSIE	0E04	TABE2	0E69	TABE3	0E73
TABSUB	0EA6	TARMON	0A8E	TASK1	0F0F	TASK2	0F0A	TASK3	0F05
TASK4	0F00	TBIP	0ED8	TEMP	001D	THUBPR	090F	TILT	002C
TIME3	0AE6	TIMLP	0CA0	TIMLP2	0A54	TLTLP0	0076	TLTLP1	00F6
TNPLAY	0ED0	TNUP	0ED4	TOMB	0E60	TOPHOL	08F6	UP1LP0	0040
UP1LP1	00C0	UP2LP0	0041	UP2LP1	00C1	UP3LP0	0042	UP3LP1	00C2
UP4LP0	0043	UP4LP1	00C3	VICKI	0E00	WAIT	002E	WAITCN	002F
WORDLP	0A96	WRTLMP	0D6F	XEDGE	003B	XST	0009	XSTART	0051
XWD	0007	Z1	0F60	Z2	0F6A	Z3	0F73	ZCCNT	0020
ZEROGR	0A3D								

TOTAL ERRORS 1

We claim as our Invention:

1. A pinball machine having visual indicators and a playing field with a plurality of ball responsive switches and a digital computer means, including one or more input ports and output ports, for receiving input signals in response to said ball responsive switches through an input port, for supplying switch address signals corresponding to selected ball responsive switches through an output port, and for supplying output signals corresponding to selected visual indicators through an output port in response to the input signals, said machine further comprising a first control circuit including interface means having an output port and an input port and being operatively connected to the ball responsive switches for supplying switch addressing test signals to said selected switches in response to the computer means switch address signals and for supplying switch input signals from select switches to said input port, and a second control circuit separate from said first control circuit having an output port and being operatively connected to the visual indicators for supplying visual indicator address signals and visual indicator data signals to activate said visual indicators in response to the computer means visual indicator output signals.

2. The machine of claim 1 wherein the ball responsive switches are operably connected as a plurality of sets of switches in a matrix, said first control circuit having a multiple bit data bus operably connected to an output port of the interface means with each line of the data bus being connected to a set of switches in the matrix and the switches being operably connected to the input port of the interface means, and means for placing an addressing test signal on each of the data lines so that an addressing test signal placed on a data line is conducted by the closed switches of the set of switches connected thereto and received as data by the input port whereby data identifying the closed switches is supplied to said computer means.

3. A pinball machine according to claim 1 further including at least one manually operated switch mounted on said pinball machine operably connected to said computer means and at least one actuating solenoid, said computer means for further receiving input signals in response to said manually operated switch and for supplying output signals to activate the solenoid.

4. A pinball machine according to claim 1 wherein said computer means comprises a microprocessor for storing said input signals, controlling said output signals and addressing said ports.

5. A pinball machine according to claim 4 wherein said computer means further comprises a memory means in which the microprocessor stores the input signals.

6. A pinball machine according to claim 4 wherein said microprocessor includes a source of timing signals.

7. A pinball machine according to claim 5 wherein said memory means comprises a random access memory.

8. A pinball machine according to claim 4 wherein said computer means further comprises a memory means for storing instructions for controlling the microprocessor wherein the microprocessor controls the output signals in response to the input signals and the instructions stored in the memory means.

9. A pinball machine according to claim 8 wherein said memory means comprises a read only memory.

10. A pinball machine according to claim 8 wherein said memory means comprises a programmable read only memory.

11. A pinball machine according to claim 4 wherein said interface means further comprises a peripheral interface adapter having said ports through which the input signals to and output signals from said microprocessor pass.

12. A pinball machine according to claim 1 wherein said visual indicators include scoring display means comprising digital display devices, said computer means having means for supplying output signals corresponding to selected digits in response to said input signals.

13. A pinball machine according to claim 12 including binary coded decimal decoder means operably connected between said digital display means and the digit output signal means.

14. A pinball machine according to claim 1 further having a plurality of solenoids for moving the ball on the playfield, said computer means further having means for supplying output signals corresponding to selected solenoids, said machine further comprising a plurality of solenoid driver circuits operably connected between said computer means and said solenoids to

activate said solenoids with one of said solenoid driver circuits connected to each of said solenoids.

15. A machine according to claim 14 further comprising a second interface means, operably connected between said driver circuits and the solenoid signal means, separate from the first control circuit interface means and having an output port for supplying solenoid activation signals to the solenoid driver circuits in response to the computer means solenoid output signals.

16. A pinball machine according to claim 1 further having coin operated switch means, said computer means having means for receiving and storing input signals in response to said coin operated switch means, and for supplying output signals corresponding to the stored input signals.

17. The machine of claim 1 wherein said second control circuit includes a second interface means having said control circuit output ports through which the visual indicator address and data signals pass and said second control circuit further comprises decoder means for decoding said visual indicator address signals to supply said visual indicator data signals to said selected visual indicators and said visual indicators having switching means associated therewith for maintaining associated visual indicators in an on or off condition in response to the decoded address signals and visual indicator data signals.

18. The machine of claim 17 wherein a plurality of said visual indicators have controlled rectifiers respectively associated therewith for conducting power to illuminate the visual indicators, said machine further having a plurality of decoders, each having multiple input lines operably connected to a control circuit address port, multiple output lines, each operably connected to a controlled rectifier, and an enable line operably connected to a control circuit data output port, each decoder being responsive to a visual indicator data signal from the second interface means to decode the visual indicator address signals supplied by said second interface means and provide a signal on an output line to a controlled rectifier in accordance with the address signal, whereby a particular visual indicator for each addressed decoder may be illuminated.

19. The machine of claim 17 wherein the visual indicators include multiple digit score indicators, and the machine comprises a third control circuit having means for supplying score indicator address signals and score indicator data signals in response to the visual indicator output signals.

20. The machine of claim 19 wherein the means for supplying score indicator address and data signals includes a third interface means having one or more output ports separate from said second interface means through which the score indicator address and data signals pass.

21. The machine of claim 1 wherein said visual indicators include a plurality of multiple digit score indicators, each having digit enable inputs and data inputs, and the machine comprises a third control circuit having an output port for supplying latch enable address signals, an output port for supplying digit enable address signals operably connected to the digit enable inputs of the multiple digit score indicators, and an output port for supplying binary score data signals; said machine having a plurality of decoder/latches each having multiple input lines operably connected to the binary score data output port, a latch enable line operably connected to the latch enable output port and multiple output lines operably connected to the data inputs of a score indicator, each of said decoder/latches being responsive to a latch enable signal from said latch enable output port to decode the binary score signals supplied by the binary score output port and supply latched signals representative of a particular digit to a score indicator, whereby latched data representing a digit to be displayed in accordance with the binary score data signals is supplied to a multiple digit score indicator by a particular decoder/latch in accordance with the latch enable signals for a particular digit position in accordance with the digit enable signals.

22. A pinball machine having a playing field with a plurality of ball responsive switches and indicator lights, a digital computer comprising means for supplying switch address signals corresponding to selected ball responsive switches, means for receiving input signals in response to said ball responsive switches and means for supplying output signals corresponding to selected indicator lights in response to said input signals, and a plurality of controlled switches respectively associated with said indicator lights for conducting power to illuminate the lights, said means for supplying output signals having an output port for supplying indicator light data signals and an output port for supplying indicator light address signals, said machine further having a plurality of decoders each having multiple input lines operably connected to the address signal output port, multiple output lines each being operably connected to a controlled switch, and an enable line operably connected to the data signal output port, each of said decoders being responsive to a data signal from the data signal output port to decode the address signals from the address signal output port and provide a signal to a controlled switch in accordance with the address from the address signal output port whereby a selected indicator light may be illuminated, and said machine further having control means including interface means having an output port and an input port and being operatively connected to the ball responsive switches for supplying switch addressing test signals to said selected switches in response to the computer means switch address signals and for supplying switch input signals from said selected switches to said input port.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,198,051
DATED : April 15, 1980
INVENTOR(S) : Marion Frank Bracha and William Harmon Englehardt

It is certified that error appears in the above—identified patent and that said Letters Patent is hereby corrected as shown below:

Column 3, line 41, change "92" to --91--.

Column 4, line 11, change "illutrated" to --illustrated--.

Column 4, line 26, change "801" to --80ℓ--.

Column 4, line 38, change "Tumper" to --Thumper--.

Signed and Sealed this

Eighteenth Day of November 1980

[SEAL]

Attest:

SIDNEY A. DIAMOND

Attesting Officer

Commissioner of Patents and Trademarks