(11) (A) No. 1 141 861

(45) ISSUED 830222

(52) CLASS 354-43

(51) INT. CL.[3] G06F 15/44

(19) (CA) CANADIAN PATENT (12)

(54) AMUSEMENT GAME MICROPROCESSOR CONTROLLER

(72) Smith, Gordon H.,
U.S.A.

(73) Granted to Rockwell International Corporation
U.S.A.

(21) APPLICATION No. 312,078

(22) FILED 780926

(30) PRIORITY DATE U.S.A. (846,247) 771028

No. OF CLAIMS 4

3120'11

FEB 22 1983 ABSTRACT 1141861

A microcomputer based pinball game controller having three distinct memory devices for control of game operation. A first memory device utilizes machine language programming instructions for control of the game independent of specific game rules thus lending itself to mass production for a large number of different amusement games. A second memory device employs a higher level language set of instructions for controlling the game in accordance with the specific rules of the game. A simple set of higher level language instructions renders this second memory device conducive to game designer programming and easy modification of the game rules, a marketing procedure traditionally used in the design of the electromechanical pin ball machine. A third memory device, employing a matrix of operator adjustable binary switches easily accessible to the exterior of the game, permits a degree of game mode control, such as the degree of playing difficulty and the number of games for a given coin denomination.

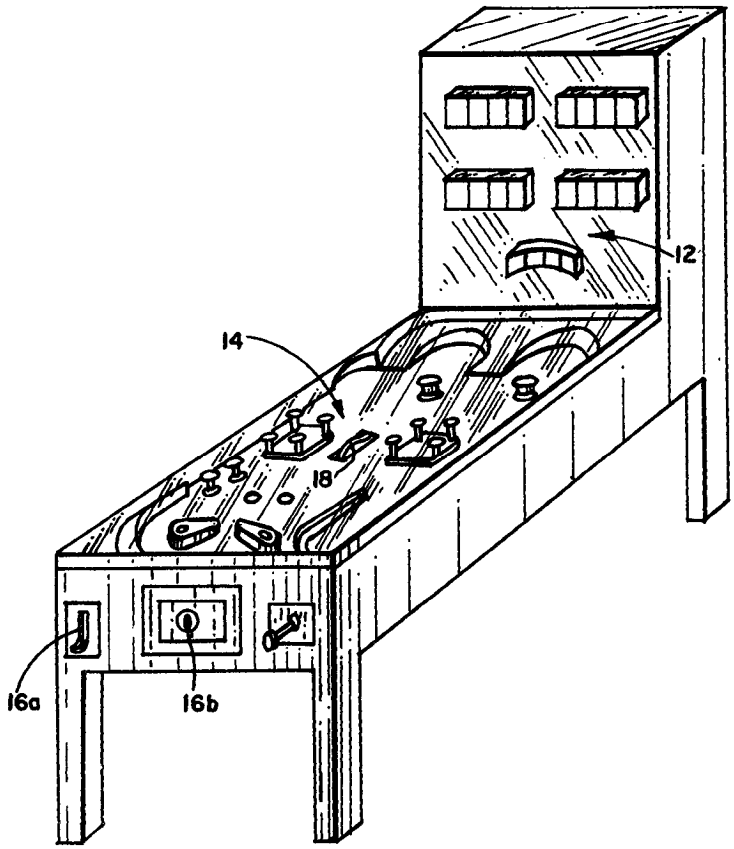

FIG. 1

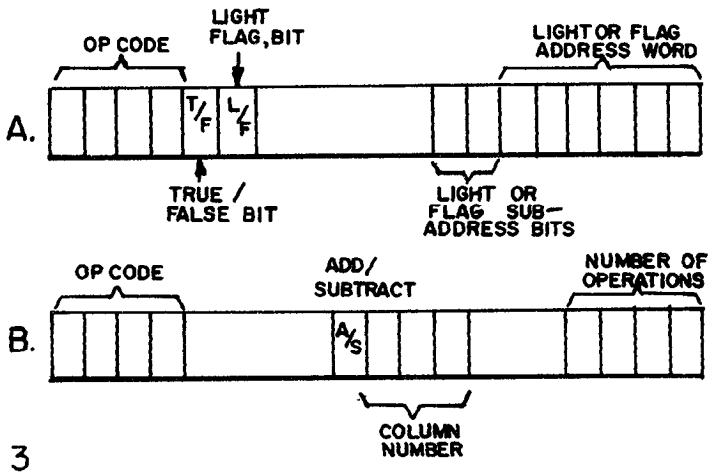

FIG. 3

Kirby Shapiro
Curphey, Eades

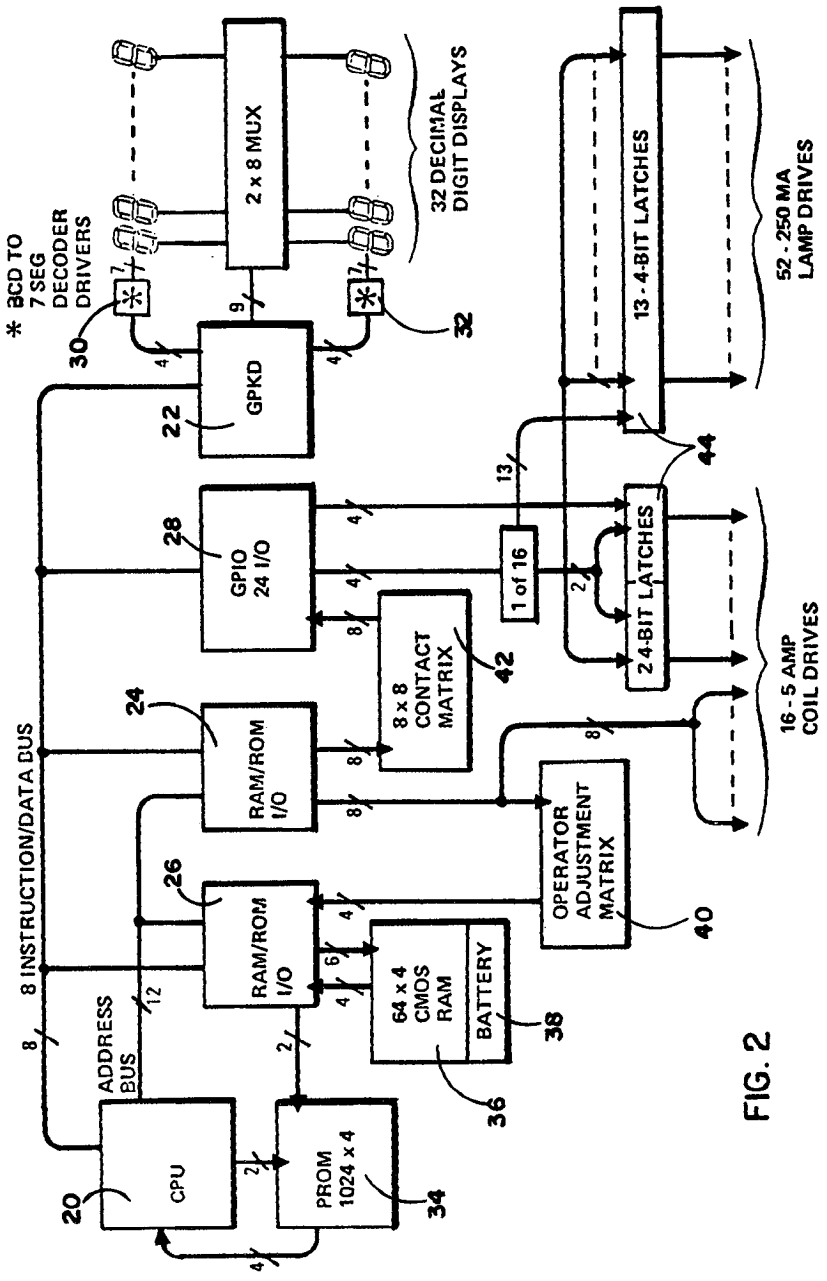

FIG. 2

Kirby Shapiro
Curphey, Eades

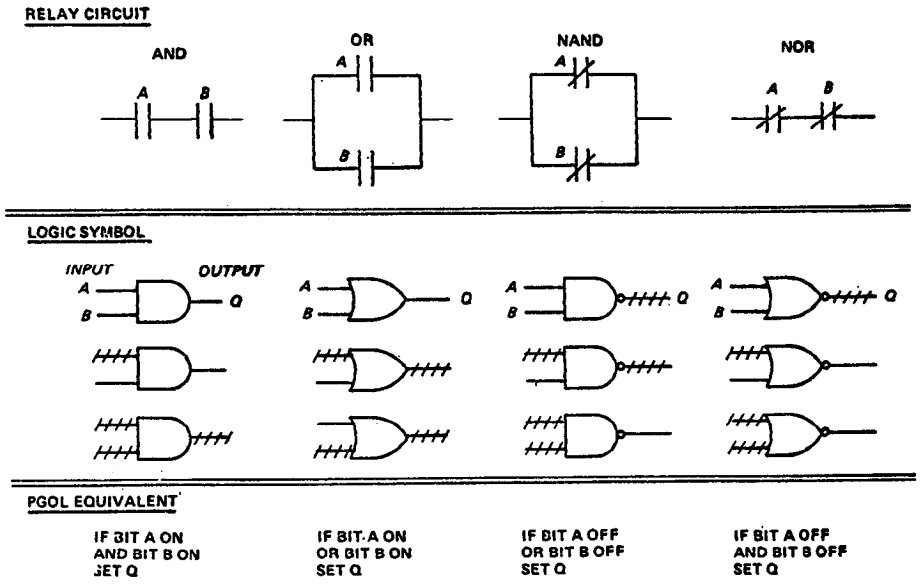

FIG. 4

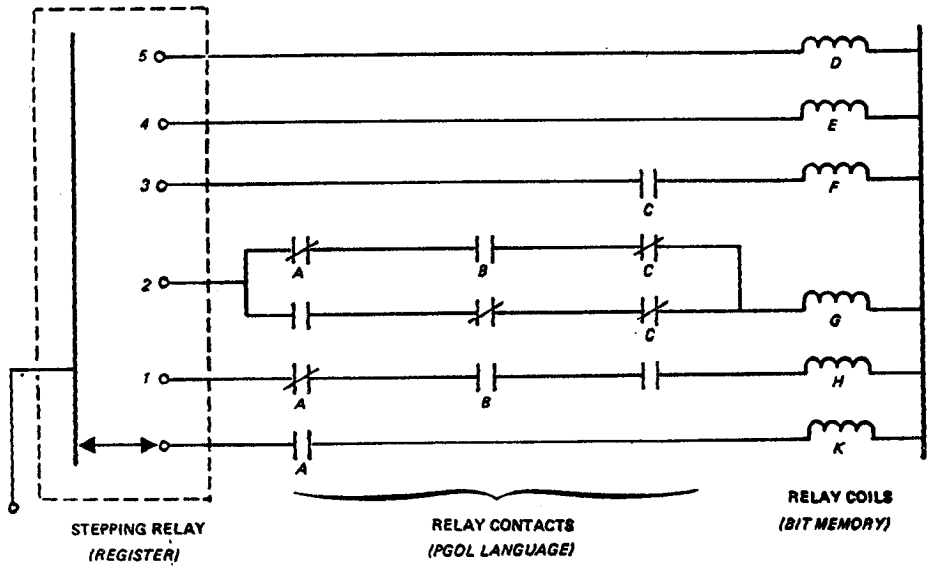

FIG. 5

Kirby Shapiro

FIELD OF THE INVENTION

This invention relates generally to a microprocessor controlled pinball game, and more specifically to a game rule memory device in combination with a pinball game controller for simplified and low cost game rule programming.

BACKGROUND OF THE INVENTION

A revolution has been taking place in a design of equipment which until now has been implemented with electromechanical devices. The logic and control functions previously performed by conventional relays, time delay relays, stepping relays, timing motors and the like are now being performed by microcomputer controlled systems. Included in this revolution are the fairly complex electromechanical devices known as pinball games.

Most of us have played electromechanical controlled pinball games, but few of us have appreciated the complexity of design that controls the flashing lights, the score, the sound and the entire pinball system. This degree of control system complexity is very suitable to microprocessor applications. In fact, microprocessor control significantly reduces the amount of material and the cost of complex pinball games at the same time increasing reliability and uptime thereby increasing the potential revenue of the arcade owner.

However there is an important problem inherent in the design, production, and subsequent use by parties having substantially different levels of sophistication in electronics-related systems that have traditionally been purely electromechanical in nature. The current invention substantially overcomes this problem providing in effect three different levels of sophistication in the programming of the parameters of a pinball game. The microprocessor electronics and associated interface usually manufactured by an electronics/microprocessor-oriented company includes the most sophisticated programming and design for overall control of the game. Standard memory devices such as read-only memory and random access memory may, by means of the invention described herein, be programmed for a large number of different games, each of which has different rules of operation.

The manufacturer of the pinball game per se, although perhaps less sophisticated in design of electronics having been traditionally associated with purely electromechanical devices, is still the ultimate expert on rules of the game to optimize player interest and revenue derived from the public. Accordingly, the current invention provides means for rule programming at a substantially higher language level by the pinball game manufacturer. The extent of the sophistication of the programming requirements is well below that required for the en masse programming of the ROM and RAM memory devices provided by the

electronics manufacturer, but still provides substantial leeway in allowing the game manufacturer to select a set of rules that suits each particular game.

The current invention also provides a set of discrete operator adjustments which comprise a relatively low level of programming sophistication, but which still permit the arcade owner to program certain aspects of the game which may vary as a function of where the pinball game is located. For example, these functions include the price of play, the number of games per coin, and the degree of difficulty of the game, which depends upon the sophistication of the player. Clearly, an arcade location would usually dictate a higher level of difficulty than a bus terminal location where the average level of player sophistication is lower.

An important byproduct of the current invention is a substantial reduction in the cost of overall manufacture and maintenance of the pinball game. The electronics manufacturer need not provide special electronics for each variation of game rules supplied by the pinball game manufacturer. As a result, all of the microprocessor electronics may be identical, irrespective of the type of game into which it will be installed. Consequently, the game manufacturer, in addition to paying less for the electronics, also reduces his cost for maintenance of the games by needing fewer spare parts for upkeep because virtually all of the electronics are identical for all of his games irrespective of variations in rules and modes of play.

In accordance with one aspect of the invention there is provided a microprocessor amusement game controller adapted to be programmed to control any of a

series of games employing a moving object in a prescribed type of game environment common to the series of games, each of the games having a different set of game play rules for controlling game play in response to input signals, comprising: (a) a central processing unit; (b) a display responsive to said unit; (c) first memory means programmed in hardware in a first word format and connected to said processing unit for controlling game operation according to said common type of game environment for all of said series of games, said first memory means being programmed substantially independently of the specific rules of play for a given game; (d) second memory means programmed in firmware in a second word format different from the first and connected to said processing unit and to the first memory means for controlling game operation dependent upon the specific rules of play for the given game; and (e) said unit and the program of said first memory means interpreting the program of the second memory means for execution thereof in response to said input signals during game play.

In accordance with another aspect of the invention there is provided a method of controlling a series of microprocessor controlled games employing at least one moving object, the series of games having a common, prescribed type of game environment but different rules of specific game play, the method of controlling being responsive to input signals representing the interaction of the moving object with objects of the game environment and comprising the steps of: (a) programming in hardware a first memory in a first word format for controlling game operation of the entire series of games

according to said common type of game environment substantially independently of the specific rules of play of an individual game; (b) programming in firmware a second memory in a different word format for controlling game operation dependent upon the specific rules of play for the given game; and (c) interpreting during game play the programming of the second memory by the programming of the first memory for execution thereof in response to said input signals.

The present invention is a microprocessor controlled pinball game having means for three levels of control program sophistication including overall game action control in response to mass produced and commonly programmed memory storage devices, a higher level language

game rule memory device which permits relatively simple variations in the electronics to accommodate each set of game rules, and a third level of control in the form of binary switch memory accessible to the ultimate consumer for convenient control of simple mode game parameters.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an isometric drawing of a typical pinball game.

Figure 2 is a block diagram of the microcomputer based pinball controller of the invention.

Figure 3 is a diagrammatic illustration of the instruction formats used in conjunction with the game rule memory device of the invention.

Figure 4 is an illustration used to explain the logical equivalencies of a game rule memory device of the invention.

Figure 5 is an illustration of examples of electromechanical logic.

DESCRIPTION OF A PREFERRED EMBODIMENT

Before proceeding to a description of the microcomputer implemented pinball machine, a brief description of the electromechanical version of the pinball game is provided in conjunction with Figure 1.

Electromechanical pinball games may be divided into three major subsystems: the vertical display 12, play field 14, and the credit subsystems. The vertical display 14 shows the player and ball status and each of the player's scores. The electromechanical design for the display is essentially the same for all pinball games except for the art work which ties the system to the scheme of each particular game.

The playing field contains the contact and lights which define the play of the game and electromechanical assemblies for kicking the ball to provide action in the game.

The credit subsystem, accepts information from the mechanical coin mechanism 16a and 16b and displays credits for playing games based upon the price per play and the coin values accepted. The credit system also accepts inputs from the play field logic to award credits for additional games based on achieving specific goals established by the game designer and settings made by the operator of the game. These game credits are stored in a reversible stepping relay which decrements the credit count as credits are spent to play additional games. The credit information is displayed on the vertical display 14.

The basic electromechanical building blocks consist of a number of standard elements which are interconnected to provide a proper score when a play field contact, such as roll over switch 18, closes. The contact closure also provides signals to solenoid-operated chimes or bells to latch relays, control lights, and enable circuit paths through contacts which allow varying scores depending on play field action as well as on bonus scores for extra games. The entire system is synchronized by means of a multicam contact timing motor so that no race conditions will occur. A race condition occurs when uncertainty in relay operation time causes unpredictable circuit paths to be established because different relays may "win the race" to closure in different situations.

The typical electromechanically controlled pinball game contains about 70 coils, of which 20 to 25 are stepping relays, 35 are logic related, and 10 to 15 are used for mechanical lockouts, ball movement or chime solenoids. In one embodiment of a microcomputer controlled pinball game, all of the above-mentioned coils, except those used for chime solenoids, are replaced by the microcomputer.

The typical electromechanical pinball game is developed and produced over an extended period of time. The game designer usually

spends several months defining the location of the play field contacts, any special play field action items, such as solenoid operated bumpers, the scoring for each contact, and any optional scoring sequences under which the bonus system operated. The game designer strives to develop a game which is interesting, having a considerable amount of action, plays for an acceptable period of time, and which has the capability of awarding the typical player approximately 30 to 35 percent free games. These parameters have been empirically established over the approximately 40 year history of pinball games to insure game interest and coin revenue to the machine owners.

Once the game is shipped, the microcomputer-based game must still have a level of programmability by the game/owner operator to achieve the desired play time, player acceptability, free game characteristics and adjustable price as the electromechanical version. The microcomputer based pinball game must give the game designer the same level of design creativity available to him in the traditional electromechanical version. The game designer must still be able to implement variations in the scoring, in the contact logic, and in the general play of the game. Furthermore, changes in these parameters by the game designer must be possible as a result of the method by which pinball games are usually marketed.

Approximately two months before a pinball game is scheduled to go into production, 200 to 300 units are typically put into the field in a test marketing situation. The machines are placed with knowledgable distributors who maintain detailed records relating to the action of players, the income gathered from the coin collection box, and other pertinent information relating to the general acceptance of the pinball game. This information is fed to the manufacturer so that the game designer can make minor modifications to the game before putting it into production. By means of the current invention, a microcomputer version of the pinball game

permits fast design development, because of the ease of making the changes of the game rule parameters simply by programming a read-only memory device. As a result, unlike the electromechanical versions, in the computer-based games changes in the play of the game can be implemented even as the machine is being readied for shipment.

Referring now to Figure 2, there is shown therein a simplified block diagram of a game controller of the current invention for use in a microcomputer based pinball game. The controller includes a central processing unit 20, a general purpose keyboard display circuit 22, two memory/input-output devices 24 and 26, and one general purpose input-output device 28. These devices provide all of the control capabilities, a total of 4,000 eight bit words of program memory, 1,000 bits of data memory, and 85 input-output lines. The total system is capable of displaying up to 32 decimal digits for scoring purposes and the like and can switch up to 68 discrete power devices such as lamps and coils.

The general purpose keyboard display circuit device 22 provides internal memory for the 32 binary coded decimal digits, and outputs the information in sequential pairs along with digital identification lines. Sequential binary coded decimal codes are supplied to a pair of binary coded decimal-to-7 segment decoded driver units 30 and 32. This configuration of the controller allows CPU 20 to load the internal memory once and then the general purpose keyboard display circuit 22 provides continuous refresh of the display information until it is commanded to change. The display system and the associated digital displays replace the scoring drums of the electromechanical systems. The scoring drums are basically stepping relays typically with several sets of contacts, with a drum around the outer periphery which rotates and shows a different digit for each position of the stepping relay.

The central processor 20 employs a TV crystal base clock oscillator (not shown) and 12 input-output lines. The input-output lines are

used to address the game rule memory device 34. RAM/ROM devices 24 and 26 each have 2,048 eight bits of read-only memory which contains the main controller program. They each also have 128 four-bit words of random access memory for the player scores, status bit memory, play field contact memory and other alterable memory. These devices also have 16 input-output lines each.

The general purpose input-output circuit 22 has a total of 24 input-output lines. The general purpose keyboard display circuit is used to buffer, refresh and control the 32 decimal digits of display. The digit strobe signals are multiplexed in two banks of eight digits. The CMOS random access memory 36 is a device which keeps data available even when power is lost or turned off, and is powered by a battery system 38 to keep the memory active. The operator adjustable matrix 40 is an array of diodes which may be independently switched in or out of the circuit so that operator selection of preprogrammed game options may be made. Operator adjustable matrix 40 shares strobe lines with 8 of the 5 amp coil drivers. These coil drivers may be shared because the matrix information may be read in a few microseconds which is too short for the solenoids or relay armatures to react. The 8x8 contact matrix 42 is scanned by signals from one of the RAM/ROM devices and return signals are read into the general internal memory to prevent bouncing error and appropriate action is initialized. Because up to 16 coil drives and up to 52 lamp drives are required in a pinball game, the output signals from the 15 four bit latches 44 go to two types of drivers: 5 amps and 250 milliamps respectively.

The play field system, which is visible to the eye of the player, is virtually unchanged in the microcomputer version of the pinball game. The mechanical devices which propel the ball are still necessary for exciting field action. Scoring contacts are closed to provide signals to the microcomputer rather than to operate conventional relays and stepping

relays found in the electromechanical games. Indicator lights showing the play field scoring status are implemented in the microcomputer game by drivers which receive a control signal from conventional TTL latches. The microcomputer selects the information to be sent to the latch and outputs it while at the same time identifying which latch is to receive the information. The contact closure information is obtained in the microcomputer system by providing a sequence of scanning signals which selects sequential groups of contacts. The microcomputer inputs the information from each group and performs the debounce function.

Because of implementation with a microcomputer, the pinball game can be provided with a capability which is not achievable in the electromechanical version. The microcomputer can be put into a special mode when the coin collector comes to collect the coins. This mode tests the overall operation and identifies by means of the display system, the contact identification information for every contact which is stuck. During this test mode the microcomputer can also output various display patterns to check the display system. It can also drive the lights to check operation of all the light bulbs and can individually drive all the solenoids of the play field and coin system to check their operation. In addition, the microcomputer can provide various bookkeeping and status information to the coin collector not possible with the electromechanical system. In the microcomputer system the bookkeeping information is stored in low power dissipation CMOS memory 36 which maintains its information even when system power is off, by means of small battery system 38.

The central processing unit 20 of Figure 2, may by way of example be a Rockwell model PPS 4/2 CPU which is described in more detail in data sheet Document No. 29000D02 published by Rockwell International Corporation in 1975 and Revised March 1976.

The ROM/RAM devices 24 and 26 of Figure 2 may by way of example be a Rockwell Model A17XX described in more detail in data sheet Document

No. 29000D28 published by Rockwell International Corporation in 1975 and Revised in October 1976.

The GPIO (General Purpose Input/Output Device) 28 of Figure 2 may by way of example be a Rockwell Model GPIO chip described in more detail in data sheet Document No. 29000D06 published by Rockwell International in 1975.

The GPKD (General Purpose Keyboard and Display Circuit) 22 of Figure 2 may by way of example be a Rockwell GPKD circuit chip described in more detail in data sheet Document No. 10788N40 published by Rockwell International in 1975.

The PROM utilized for storing the higher level language program may by way of example be a programmable read only memory Model 6351 manufactured by Monolythic Memories Incorporated.

The requirement for stepping relays for bonus advancing or for timing motors to eliminate race conditions is eliminated from the microcomputer system. The position of a stepping relay can be implemented in the microcomputer by storing a number in memory representing the stepper position. The microcomputer can use this number as part of a program logic sequence to implement the desired function for that position. The microcomputer can provide sequence information much more readily and because of the sequential operation of the microcomputer, the timing motor is not required since race conditions are impossible.

The software approach used in the implementation of the microcomputer pinball game of the current invention is actually implemented in three levels: the first programming level includes those basic control functions that every pinball game controller is expected to perform. These are the necessary power on, sequencing and control, display operation, player and ball counting operations, play field input computations, general play field control operations for functions such as "tilt" and functions which occur when the ball leaves the playing field, and similar operations

which do not change from game to game. This program is generated by the controller designer to allow the system to be a general pinball game controller, and is the program stored in the microcomputer fixed read-only memory contained in the RAM/ROM devices 24 and 26.

The next level of programming is generated by the game designer and is accomplished in an interpretative program format. This means that the game controller program in the RAM/ROM devices interprets programs written in a higher level language oriented toward pinball game rules. Using this higher level language, the game designer selects the operation response to each contact in easy to use sentence descriptions of the rules of the game. The instructions used in this higher level pinball-game oriented language (PGOL) are indicated in Table I and the instruction format for two types of instructions are presented in Figure 3.

As indicated in Figure 3 there are basically two types of instruction format. Format A is used for copying or setting the logic state on certain lights, flags, or solenoids as the game proceeds. Instruction format B is used to control the score of the game. As indicated in Figure 3, each instruction format includes an OP code comprising four bits. This OP code identifies the instruction generated. The Format A instruction also includes a true-false bit which indicates whether the flag, light, or solenoid, the state of which is being copied or set, should be copied or set in its current state or in an inverted state. A light-flag bit indicates whether the instruction will have an effect on a light or on a flag or solenoid. The right-most eight bits of the Format A instruction include a light or flag address word of six bits and a light or flag subaddress comprising two bits. In combination these eight bits designate the specific light, flag or solenoid the instruction operates on.

The Format B instruction, which is specifically oriented to effecting scoring changes, includes an add-subtract bit which indicates whether an addition or subtraction should be made to a score. It also includes a

TABLE I

PGOL INSTRUCTION SET

MNEMONIC	OP CODE	ADDRESS		FUNCTION
*COPY	0	TLBB	CCCC	COPY LAMP/FLAG STATE TO CONTROL BIT
*SCORE	1	ANNN	VVVV	ADD/SUBTRACT TO/FROM SCORE
*SET	2	TLBB	CCCC	SET/RESET LAMP/FLAG/SOL
*CGOTO	3	QQQQ	RRRR	CONDITIONAL GO TO LOCATION/QQQQ RRRR
RFLG	4	-	-	ROTATE FLAGS (11-20) LEFT ONE POSITION
*RBUP	5	-	-	BONUS UP
*RBDN	6	-	-	BONUS DOWN
*REQL	7	MMMM	-	IF REG ≠ MMMM, CONTROL BIT = 0
*DECR	8	-	-	DECREMENT INDEX REGISTER
*INCR	9	-	-	INCREMENT INDEX REGISTER
OR	A	TLBB	CCCC	OR LAMP/FLAG STATE WITH CONTROL BIT
FLAG	B	-	-	SET CONTROL BIT = 1
STOP	C	-	-	RETURN TO MAIN PROGRAM
TOGL	D	-	-	TOGGLE CONTROL BIT
GOTO	E	QQQQ	RRRR	GO TO LOCATION QQQQ RRRR
DELAY	F	-	-	150 MILLISECONDS DELAY

*ONLY EXECUTED IF THE CONTROL BIT = 1

column number comprising three bits which indicate which column of the score display is being affected by the current instruction. Finally, the instruction contains four bits indicating the number of operations to effect the score change. For example, if 5,000 is to be added to the score, the add bit would be TRUE, the column number would correspond to five, so that one would be added five times to the thousands column of the score.

When a contact on the play field closes in response to the play of the game, the controller program stored in the RAM/ROM devices causes the instructions in a particular section of the higher level language memory to be executed. The sequence of instructions starting at a particular program location corresponding to the contact closure, indicates exactly what the game is to do when the contact closes. For example, one contact may simply cause the system to score 100 points for the player who is operating the game. In this case, the higher level language program consists of two instructions: 1) score 100 points and 2) stop. The stop code indicates the completion of the operation related to this particular contact closure. If another contact is closed, the controller may cause the instruction sequence for that contact to be executed; for example, the contact instructions may be as follows: when contact 1 closes, if contact 3 and contact 11 have been closed, score 1,000, otherwise score 100 and stop. This requires six higher level language instructions to implement. In both of these examples, the controller program interprets what the game designer defined by the higher level language instructions and executes a sequence of machine language instructions to accomplish each instruction and to continue to the next one.

The philosophy of operation of the higher level language program is that a controlled flag bit in a microcomputer memory is initially set to a ONE state. A higher level language instruction inspects the state

of that bit and performs the operation specified if the bit continues to be in the ONE state. Many of the logic type higher level language instructions control the state of the bit to cause blocks of logic to be ignored or executed as the situation demands. The unconditional instructions control the flow of the logic and always execute regardless of the state of the control bit or flag bit. The higher level language is a general logic language with some special instructions which relate to pinball operations. Special pinball instructions are the "score" instructions, the "increment-decrement bonus" instructions, the "increment-decrement" register instruction and the "register equal" instructions.

From the instructions listed in Table I, it can be seen that the full capability for sequential logic is provided by AND functions and OR functions with TRUE or FALSE states. Figure 4 illustrates the equivalent high level language statement for the four basic logic operations. These four basic logic operations are shown in relay circuit, logic gate symbols, and higher level language statement forms. In the logic gate illustrations, the input lines on the left are active when the line is shaded. If the gate passes a signal, the output is shaded. The shaded inputs are then equivalent to the relay being activated in the relay logic, or to the bit memory being in the ON state in the higher level language equivalent.

The higher level language program provides a capability for counting events and making decisions based upon the actual value of the count. Also provided is the capability for inserting time delays and for setting, resetting, and testing individual status bits in a bit memory in the microprocessor system to store the status of contact closures, light bulb drives, coil drives and logic information.

Using the higher level language program, the game designer can easily and quickly configure the logical options for a particular game and modify them as he develops information about the general play of the game and its level of difficulty. In one embodiment of the invention,

this higher level language program is implemented in programmable read-only memory PROM, even in the production system, so that the game modifications which must be made from feedback from the field operations can be implemented right up to the last moment in the production line.

A general example for implementing a block of mechanical logic with the higher level language programming of the current invention is illustrated in Figure 5. The higher level language program statements corresponding to the electromechanical logic of Figure 5 would be as indicated below:

If register equals 5, set bit D, always.
If register equals 4, set bit E, always.
If register equals 3, and if bit C is on, set bit F, always.
If register equals 2, and if bit A is off, and if bit B is on, and if bit C is off, set bit G always.
Or if register equals 2,and if bit A is on, and bit B is off, and bit C is off, set bit G always.
If register equals 1, and if bit A is off and if bit B is on and if bit C is on, set bit H always.
If register equals 0, and if bit A is on, set bit K.
STOP

Table II is a program listing of a programmable read only memory unit used in a preferred embodiment of the invention to store the higher level language used for game rule control. The column on the left is the input address to the PROM, expressed in hexadecimal format. The next four columns, labeled W1, W2, W3 and W4 respectively, are the hexadecimal representations of the output of the PROM generated in response to the input address. The next column to the right is an instruction number used for reference in the listing. The next column to the right is a label used to identify each step in the listing for GoTo operations. The next column to the right is the name of the OP code for each instruction in

TABLE II

PBA 06/10.76

```
ADDR   W1 W2 W3 W4     ISN    LABEL       OP        OPERAND
                         1    *                 PGOL PROGRAM LISTING - GAME #409
                         2    *  FLAGS        F11-F20 KICKOUT HOLE-SPOT ROLLOVER INDICATORS
                         3    *               F1  ALL ROLLOVERS DOWN
                         4    *               F2  ALL DROP TARGETS DOWN
                         5    *               F3  ALL DROP TARGETS DOWN  1ST PASS ONLY
                         6    *               F4  SPECIAL AWARD
                         7    *               L4  SAME PLAYER SHOOTS AGAIN
                         8    *               F5  SPECIAL HIT
                         9    *               F6  EXTRA BALL HIT
                        10    *               F38 5BALL=0 3BALL=1
                        11    *               F7  TEMP FLAG -(10 POINTS)
                        12    *               F9  TEMP FLAG
                        13    *               F21 EXTRA BALL
                        14    *
0100   E  4  1  5       15                    GOTO    INIT
0104   E  2  6  B       16                    GOTO    BONUS
0108   F  F  F  F       17                    FILL    24
0120   E  3  C  R       18    S10P            GOTO    10P       10 POINT
0124   E  3  C  0       19    SCPB            GOTO    CPB       CENTER POP BUMPER
0128   E  3  A  3       20    SRIG            GOTO    RIG       RIGHT TARGET
0120   E  2  F  D       21    SCTG            GOTO    CIG       CENTER TARGET
0130   E  3  8  6       22    SCTG            GOTO    LTG       LEFT TARGET
0134   F  F  F  F       23                    FILL    12
0140   E  1  C  0       24    SBRO            GOTO    BR01      BLUE ROLLOVER
0144   E  1  D  1       25    SGRO            GOTO    GR01      GREEN ROLLOVER
0148   1  A  5          26    SRRB            SCORE   A1000,5   RIGHT ROLLOVER BUTTON
014B   C                27                    STOP
014C   E  2  0  C       28    SWDT            GOTO    WDT       WHITE DROP TARGET
0150   E  2  0  2       29    SYDT            GOTO    YDT       YELLOW DROP TARGET
0154   F  F  F  F       30                    FILL    12
0160   E  2  2  A       31    SGDT            GOTO    GDT       GREEN DROP TARGET
0164   E  1  B  2       32    SRRO            GOTO    RR01      RED ROLLOVER
0168   E  2  1  6       33    SRDT            GOTO    RDT       RED  DROP TARGET
0160   E  1  9  0       34    SYRO            GOTO    YR01      YELLOW ROLLOVER
0170   E  1  A  1       35    SWRO            GOTO    WR01      WHITE ROLLOVER
```

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
0174	F	F	F	F	36		FILL	12	
0180	E	3	7	E	37	SRPB	GOTO	POP	RIGHT POP BUMPERS
0184	E	3	1	2	38	SRHL	GOTO	RHL	RIGHT HOLE
018B	E	3	0	H	39	SLHL	GOTO	LHL	LLEFT HOLE
018C	E	2	2	0	40	SBDT	GOTO	BDT	BLUE DROP TARGET
					41	*			
					42	* YELLOW ROLLOVER			
0190	O	C	7		43	YR01	COPY	TL28	
0193	2	4	7		44		SET	FL28	
0196	2	6	6		45		SET	FL22	
0199	2	0	2		46		SET	TL8	
0190	9				47		INCR		
019D	E	1	D	E	48		GOTO	ROV	
					49	*			
					50	* WHITE ROLLOVER			
					51	*			
01A1	O	D	8		52	WR01	COPY	TL29	
01A4	2	5	8		53		SET	FL29	
01A7	2	7	6		54		SET	FL23	
01AA	2	E	2		55		SET	TL6	
01AD	9				56		INCR		
01AE	E	1	D	E	57		GOTO	ROV	
					58	*			
					59	* RED ROLLOVER			
					60	*			
01B2	O	E	8		61	RR01	COPY	TL30	
01B5	2	6	8		62		SET	FL30	
01B8	2	D	2		63		SET	TL5	
01BB	9				64		INCR		
01BC	E	1	D	E	65		GOTO	ROV	
					66	*			
					67	* BLUE ROLLOVER			
					68	*			
0100	0	F	8		69	BR01	COPY	TL31	
0103	2	7	8		70		SET	FL31	
0106	2	4	6		71		SET	FL24	

TABLE II (CONTINUED)

```
PBA 06/10/76
ADDR  W1 W2 W3 W4     ISN   LABEL OP     OPERAND
0109  2  F  2        72                 SET      TL7
010C  9              73                 INCR
010D  E  1  D  E     74                 GOTO     ROV
                     75    *
                     76    *      GREEN ROLLOVER
                     77    *
01D1  0  C  8        78    GR01         COPY     TL32
01D4  2  4  8        79                 SET      FL32
01D7  2  5  7        80                 SET      FL25
01DA  2  D  3        81                 SET      TL9
01DD  9              82                 INCR
                     83    *
                     84    *      ROLLOVER
                     85    *
01DE  0  D  2        86    ROV          COPY     TL5       IF ALL ROLLOVERS DOWN
01E1  0  E  2        87                 COPY     TL6
01E4  0  F  2        88                 COPY     TL7
01E7  0  C  2        89                 COPY     TL8
01EA  0  D  3        90                 COPY     TL9
01ED  2  9  0        91                 SET      TF1       F1=1
01F0  2  B  0        92                 SET      TF3        1ST PASS FLAG
01F3  B              93                 FLAG
01F4  0  0  1        94                 COPY     FF8
01F7  1  B  5        95                 SCORE    A100,5
01FA  B              96                 FLAG
01FB  2  0  1        97                 SET      FF8
01FE  E  3  D  2     98                 GOTO     RODT
                     99    *
                     100   *      YELLOW DROP TARGET
                     101   *
0202  0  5  4        102   YDT          COPY     FL13      NEED 'COPY' FOR 1ST PASS FLAG
0205  2  D  4        103                SET      TL13
0208  E  2  3  0     104                GOTO     DT
                     105   *
                     106   *      WHITE DROP TARGET
0200  0  7  3        107   WDT          COPY     FL11
```

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND
020F	2	F	3		108		SET	TL11
0212	E	2	3	0	109		GOTO	DT
					110	*		
					111	*	RED DROP TARGET	
0216	0	6	3		112	RDT	COPY	FL10
0219	2	E	3		113		SET	TL10
021C	E	2	3	0	114		GOTO	DT
					115	*		
					116	*	BLUE DROP TARGET	
0220	0	4	3		117	BDT	COPY	FL12
0223	2	C	3		118		SET	TL12
0226	E	2	3	0	119		GOTO	DT
					120	*		
					121	*	GREEN DROP TARGET	
022A	0	6	4		122	GDT	COPY	FL14
022D	2	E	4		123		SET	TL14
					124	*		
					125	*	DROP TARGETS - EXTRA BALL AND SPECIAL LIGHTS	
					126	*		
0230	0	E	3		127	DT	COPY	TL10
0233	0	F	3		128		COPY	TL11
0236	0	C	3		129		COPY	TL12
0239	0	D	4		130		COPY	TL13
023C	0	E	4		131		COPY	TL14
023F	2	A	0		132		SET	TF2
0242	2	B	0		133		SET	TF3
0245	2	E	7		134		SET	TL26 CENTER TARGET
					135	*		
					136	*	DROP TARGETS - SCORE	
					137	*		
0248	B				138		FLAG	
0249	7	1			139		REQL	1
024B	1	A	1		140		SCORE	A1000,1
024E	B				141		FLAG	
024F	7	2			142		REQL	2
0251	1	A	2		143		SCORE	A1000,2

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
0254	B				144		FLAG		
0255	7	3			145		REQL	3	
0257	1	A	3		146		SCORE	A1000,3	
025A	B				147		FLAG		
025B	7	4			148		REQL	4	
025D	1	A	4		149		SCORE	A1000,4	
0260	B				150		FLAG		
0261	7	5			151		REQL	5	
0263	1	A	5		152		SCORE	A1000,5	
0266	B				153		FLAG		
0267	E	3	D	2	154		GOTO	RODT	
					155	*			
					156	*	BONUS COUNTDOWN		
					157	*			
					158	*	GREEN BONUS		
					159	*			
026B	2	F	5		160	BONUS	SET	TL19	
026E	0	D	3		161		COPY	TL9	
0271	0	E	4		162		COPY	TL14	
0274	1	A	1		163		SCORE	A1000,1	
0277	F	F			164		FILL	2	
0279	0	C	5		165		COPY	TL20	
027C	0	D	6		166		COPY	TL21	
027F	1	A	1		167		SCORE	A1000,1	
0282	F	F			168		FILL	2	
0284	B				169		FLAG		
0285	2	7	5		170		SET	FL19	
					171	*			
					172	*	YELLOW BONUS		
					173	*			
0288	2	E	5		174		SET	TL18	YELLOW
028B	0	C	2		175		COPY	TL8	
028E	0	D	4		176		COPY	TL13	
0291	1	A	2		177		SCORE	A1000,2	
0294	F	F			178		FILL	2	
0296	0	C	5		179		COPY	TL20	

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND
0299	0	D	6		180		COPY	TL21
029C	1	A	2		181		SCORE	A1000,2
029F	F	F			182		FILL	2
02A1	B				183		FLAG	
02A2	2	6	5		184		SET	FL18
					185	*		
					186	*	BLUE BONUS	
02A5	2	D	5		187		SET	TL17
02A8	0	F	2		188		COPY	TL7
02AB	0	C	3		189		COPY	TL12
02AE	1	A	3		190		SCORE	A1000,3
02B1	F	F			191		FILL	2
02B3	0	C	5		192		COPY	TL20
02B6	0	D	6		193		COPY	TL21
02B9	1	A	3		194		SCORE	A1000,3
02BC	F	F			195		FILL	2
02BE	B				196		FLAG	
02BF	2	5	5		197		SET	FL17
					198	*		
					199	*	WHITE BONUS	
02C2	2	C	4		200		SET	TL16
02C5	0	E	2		201		COPY	TL6
02C8	0	F	3		202		COPY	TL11
02CB	1	A	4		203		SCORE	A1000,4
02CE	F	F			204		FILL	2
02D0	0	C	5		205		COPY	TL20
02D3	0	D	6		206		COPY	TL21
02D6	1	A	4		207		SCORE	A1000,4
02D9	F	F			208		FILL	2
02DB	B				209		FLAG	
02DC	2	4	4		210		SET	FL16
					211	*		
					212	*	RED BONUS	
02DF	2	F	4		213		SET	TL15
02E2	0	D	2		214		COPY	TL5
02F5	0	E	3		215		COPY	TL10

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
02E8	1	A	5		216		SCORE	A1000,5	
02EB	F	F			217		FILL	2	
02ED	0	C	5		218		COPY	TL20	
02F0	0	D	6		219		COPY	TL21	
02F3	1	A	5		220		SCORE	A1000,5	
02F6	F	F			221		FILL	2	
02F8	8				222		FLAG		
02F9	2	7	4		223		SET	FL15	
02FC	C				224		STOP		
					225	*			
					226	*	CENTER TARGET		
					227	*			
02FD	0	6	7		228	CTG	COPY	FL26	CENTER TARGET=1 ?
0300	1	B	5		229		SCORE	A100,5	
0303	D				230		TOGL		
0304	1	A	5		231		SCORE	A1000,5	
0307	C				232		STOP		
					233	*			
					234	*	LEFT HOLE		
					235	*			
0308	2	C	5		236	LHL	SET	TL20	LEFT HALF DOUBLE BONUS=1
030B	2	2	5		237		SET	FF22	
030E	E	3	1	8	238		GOTO	HOLE	
					239	*			
					240	*			
					241	*	RIGHT HOLE		
0312	2	D	6		242	RHL	SET	TL21	RIGHT HALF DOUBLE BONUS=1
0315	2	A	5		243		SET	TF22	
0318	0	A	9		244	HOLE	COPY	TF38	3 BALL ?
031B	1	A	5		245		SCORE	A1000,5	SCORE 5000
031E	D				246		TOGL		OTHERWISE
031F	1	A	3		247		SCORE	A1000,3	SCORE 3000
0322	B				248		FLAG		
0323	0	A	5		249		COPY	TF22	
0326	2	A	D		250		SET	TS6	
0329	D				251		TOGL		

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
032A	2	B	D		252		SET	TS7	
032D	B				253		FLAG		
032E	2	A	8		254		SET	TF34	
0331	2	8	1		255		SET	TF8	
0334	0	8	2		256		COPY	TF12	
0337	0	A	9		257		COPY	TF38	3 BALL
033A	A	B	2		258		OR	TF11	
033D	E	1	9	0	259		GOTO	YR01	SPOT YELLOW ROLLOVER
0341	B				260		FLAG		
0342	0	A	3		261		COPY	TF14	
0345	0	A	9		262		COPY	TF38	
0348	A	9	3		263		OR	TF13	
034B	E	1	A	1	264		GOTO	WR01	SPOT WHITE ROLLOVER
034F	B				265		FLAG		
0350	0	8	3		266		COPY	TF16	
0353	0	A	9		267		COPY	TF38	
0356	A	B	3		268		OR	TF15	
0359	E	1	B	2	269		GOTO	RR01	SPOT RED ROLLOVER
035D	B				270		FLAG		
035E	0	A	4		271		COPY	TF18	
0361	0	A	9		272		COPY	TF38	
0364	A	9	4		273		OR	TF17	
0367	F	1	C	0	274		GOTO	BR01	SPOT BLUE ROLLOVER
036B	B				275		FLAG		
036C	0	8	4		276		COPY	TF20	
036F	0	A	9		277		COPY	TF38	
0372	A	B	4		278		OR	TF19	
0375	E	1	D	1	279		GOTO	GR01	SPOT GREEN ROLLOVER
0379	B				280		FLAG		
037A	2	0	1		281		SET	FF8	
037D	C				282		STOP		
					283	*			
					284	*	LEFT OR RIGHT POP BUMPERS		
					285	*			
037E	1	B	1		286	POP	SCORE	A100,1	
0381	4				287		LBDN		MOVE ROLLOVER FLAG ONE POSITION

TABLE II (CONTINUED)

```
PBA 06/10/76
ADDR  W1 W2 W3 W4     ISN    LABEL        OP       OPERAND
                      288    *
                      289    *     EXTRA BALL LITES (L41.L40)
                      290    *
0382  E  4  5  9      291           GOTO     POP01
                      292    *
                      293    *     LEFT TARGET
                      294    *
0386  1  B  5         295    LTG    SCORE    A100,5
0389  0  F  9         296           COPY     TL35      SPECIAL ?
038C  2  7  9         297           SET      FL35      L35=0
038F  2  9  1         298           SET      TF5       DISABLE SPECIAL
0392  2  8  0         299           SET      TF4       CREDIT FLAG
0395  B               300           FLAG
0396  0  E  9         301           COPY     TL34      EXTRA BALL=1 ?
0399  2  6  9         302           SET      FL34      L34=0
039C  2  A  1         303           SET      TF6       DISABLE EXTRA BALL
039F  2  9  5         304           SET      TF21       EXTRA BALL FLAG
03A2  C               305           STOP
                      306    *
                      307    *     RIGHT TARGET
                      308    *
03A3  1  B  5         309    RTG    SCORE    A100,5
03A6  0  C  9         310           COPY     TL36      SPECIAL ?
03A9  2  4  9         311           SET      FL36      L36=0
03AC  2  9  1         312           SET      TF5
03AF  2  8  0         313           SET      TF4       CREDIT FLAG
03B2  B               314           FLAG
03B3  0  D  9         315           COPY     TL33      EXTRA BALL ?
03B6  2  5  9         316           SET      FL33      L40=0
03B9  2  A  1         317           SET      TF6
03BC  2  9  5         318           SET      TF21      EXTRA BALL FLAG
03BF  C               319           STOP
                      320    *
                      321    *     CENTER POP BUMPER
                      322    *
03C0  0  A  9         323    CPB    COPY     TF38      3 BALL ?
```

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
03C3	1	A	1		324		SCOPE	A1000,1	SCORE 1000
03C6	D				325		TOGL		OTHERWISE
03C7	1	B	1		326		SCORE	A100,1	SCORE 100
03CA	C				327		STOP		
					328	*			
					329	*	10 POINTS		
					330	*			
03CB	1	C	1		331	10P	SCORE	A10,1	
03CE	2	8	8		332		SET	TF36	ALTERNATE SPECIAL AND EXTRA BALL LIGHTS
03D1	C				333		STOP		
					334	*			
					335	*	ROLLOVER AND DROP TARGET EXIT ROUTINE		
03D2	0	8	0		336	RODT	COPY	TF3	1ST PASS ?
03D5	0	6	9		337		COPY	FL34	
03D8	0	5	9		338		COPY	FL33	((L33.L34)=0 ?
03DB	0	2	1		339		COPY	FF6	EXTRA BALL NOT HIT ?
03DE	0	7	9		340		COPY	FL35	
03E1	0	4	1		341		COPY	FL4	
03E4	2	E	9		342		SET	TL34	
03E7	B				343		FLAG		
03E8	0	B	0		344		COPY	TF3	1ST PASS ?
03EB	0	9	0		345		COPY	TF1	ALL ROLLOVERS ?
03EF	0	A	0		346		COPY	TF2	ALL DROP TARGETS ?
03F1	0	6	9		347		COPY	FL34	L34=0 ?
03F4	0	1	1		348		COPY	FF5	
03F7	2	F	9		349		SET	TL35	L35=1 (SPECIAL)
03FA	B				350		FLAG		
03FB	0	B	0		351		COPY	TF3	1ST PASS ?
03FE	0	9	0		352		COPY	TF1	ALL ROLLOVERS ?
0401	0	A	0		353		COPY	TF2	ALL DROP TARGETS ?
0404	0	5	9		354		COPY	FL33	L33=0 ?
0407	0	7	9		355		COPY	FL35	
040A	0	1	1		356		COPY	FF5	
040D	2	C	9		357		SET	TL36	L36=1 (SPECIAL)
0410	B				358		FLAG		
0411	2	3	0		359		SET	FF3	

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
0414	C				360		STOP		
					361	*			
					362	*	INITIALIZATION		
					363	*			
0415	2	C	7		364	INIT	SET	TL28	
0418	2	D	8		365		SET	TL29	
041B	2	E	8		366		SET	TL30	
041E	2	F	8		367		SET	TL31	
0421	2	C	8		368		SET	TL32	
0424	2	E	6		369		SET	TL22	
0427	2	F	6		370		SET	TL23	
042A	2	C	6		371		SET	TL24	
042D	2	D	7		372		SET	TL25	
0430	2	B	2		373		SET	TF11	
0433	2	8	D		374		SET	TS8	
0436	2	8	D		375		SET	TS8	
0439	2	8	D		376		SET	TS8	
043C	0	8	6		377		COPY	TF27	
043F	0	9	6		378		COPY	TF25	LAST BALL ?
0442	2	C	5		379		SET	TL20	
0445	2	D	6		380		SET	TL21	DOUBLE BONUS=1
0448	B				381		FLAG		
0449	0	A	6		382		COPY	TF26	
044C	0	9	6		383		COPY	TF25	LAST BALL ?
044F	0	A	9		384		COPY	TF38	3 BALL ?
0452	2	C	5		385		SET	TL20	
0455	2	D	6		386		SET	TL21	DOUBLE BONUS=1
0458	C				387		STOP		
					388	*			
					389	*	ALTERNATE EXTRA BALL LITE WITH 5 BALL		
					390	*			
0459	0	9	0		391	POPO1	COPY	TF1	ALL R.O.'S
045C	A	A	0		392		OR	TF2	OR DT.HS
045F	E	4	6	4	393		GOTO	POP1	
0463	C				394		STOP		
0464	0	2	1		395	POP1	COPY	FF6	EXTRA BALL HIT ?

TABLE II (CONTINUED)

PBA 06/10/76

ADDR	W1	W2	W3	W4	ISN	LABEL	OP	OPERAND	
0467	E	4	6	C	396		GOTO	POP2	NO
046B	C				397		STOP		YES
046C	0	2	9		398	POP2	COPY	FF38	3 BALL ?
046F	E	4	7	4	399		GOTO	POP3	NO
0473	C				400		STOP		
0474	0	D	9		401	POP3	COPY	TL33	L33=1 ?
0477	E	4	A	2	402		GOTO	POP4	YES
047B	B				403		FLAG		
047C	0	E	9		404		COPY	TL34	L34=1 ?
047F	E	4	A	9	405		GOTO	POP5	YES
0483	B				406		FLAG		
0484	0	C	9		407		COPY	TL36	L36=1?
0487	E	4	B	0	408		GOTO	POP6	
0488	B				409		FLAG		
048C	0	F	9		410		COPY	TL35	L35=1?
048F	E	4	B	4	411		GOTO	POP7	
0493	B				412		FLAG		
0494	0	B	5		413		COPY	TF23	
0497	2	D	9		414		SET	TL33	
049A	B				415		FLAG		
049B	0	3	5		416		COPY	FF23	
049E	2	E	9		417		SET	TL34	
04A1	C				418		STOP		
04A2	2	5	9		419	POP4	SET	FL33	L33=O
04A5	2	B	5		420		SET	TF23	F23=1
04A8	C				421		STOP		
04A9	2	6	9		422	POP5	SET	FL34	L34=0
04AC	2	3	5		423		SET	FF23	F23=0
04AF	C				424		STOP		
04B0	2	E	9		425	POP6	SET	TL34	L34=1
04B3	C				426		STOP		
04B4	2	D	9		427	POP7	SET	TL33	
04B7	C				428		STOP		
				***	429				

0

the listing. The next column to the right is a name given to the OPERAND of the instruction indicating what the instruction will act on. The final column to the right is a brief explanation of the instruction in the listing.

By way of an example in using the program listing of Table II, instruction number 50 is labeled White Rollover. This label corresponds to a series of higher level language instructions which occurs in response to a contact closure when a Rollover switch, such as switch 18 of Figure 1, is activated by the pinball. As indicated at instruction number 52, the input address to the higher level language PROM is 01A1 in hexadecimal form and the output is 0D8 in hexadecimal form. The label given to this instruction and to the five instructions that follow and that together comprise this subroutine designated White Rollover is WR01. The first step is a copy instruction which calls for copying TRUE light 29. The next step, with input address 01A4, output 258, is a set instruction to set FALSE light 29. After two more set instructions, the subroutine calls for an increment of an index register in the PROM by means of input address 01AD and output 9. The final instruction in this subroutine White Rollover, is a GoTo instruction where the address of the destination of the GoTo instruction is ROV which as shown as instruction number 86 is the name of another subroutine clled Rollover.

An example of a Score instruction is shown at instruction number 26 of Table II. The input address of the PROM is 0148, and the output is 1A5 which as indicated by the OPCODE and OPERAND, is an instruction which causes an increase in the score by 5,000 which is accomplished by adding to the 1,000th column a total of five times.

SUMMARY

It will now be understood that what is described herein is a microcomputer based pinball machine controller having a means for three different levels of programming and control. The first level is a machine language program that may be provided in mass production quantities by the controller designer and is capable of accomodating all of the anticipated variations for which the controller may be used irrespective of the particular rules of a pinball game. The second level is a higher level language interpretive routine having a high level language flexible instruction set permitting pinball game designers to utilize their creativity in the design of the rules of the game without requiring large amounts of programming time ordinarily needed to establish the rules of a particular game. The third level, the least sophisticated in terms of an actual knowledge of the detailed electronics of the controller, permits operator control by means of binary switches of general game mode operations, such as the difficulty of play and the number of plays for each coin.

Although a specific embodiment has been described it will be understood that the invention is not limited to the particular implementation utilized and that the invention could be implemented in other forms of logic including other types of hardware and software to accomplish the operations described herein. However, all such alternative embodiments are contemplated within the scope of the invention.

The invention has been described in more than sufficient detail to enable one skilled in the art to make and use the invention. For purposes of brevity and to avoid inadvertent obfuscation of the important elements of the invention, certain trivial aspects have not been described in specific detail. By way of example, specific time relationships of clock signals have not been specifically delineated. However, these

aspects of the invention will now be readily apparent to those having skill in the applicable art and having the teaching of the applicants before them.

The invention described herein may be employed in many ways different from that specifically set forth and many variations may be made therein within the scope of the appended claims.

I claim:

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A microprocessor amusement game controller adapted to be programmed to control any of a series of games employing a moving object in a prescribed type of game environment common to the series of games, each of the games having a different set of game play rules for controlling game play in response to input signals, comprising:

(a) a central processing unit;

(b) a display responsive to said unit;

(c) first memory means programmed in hardware in a first word format and connected to said processing unit for controlling game operation according to said common type of game environment for all of said series of games, said first memory means being programmed substantially independently of the specific rules of play for a given game;

(d) second memory means programmed in firmware in a second word format different from the first and connected to said processing unit and to the first memory means for controlling game operation dependent upon the specific rules of play for the given game; and

(e) said unit and the program of said first memory means interpreting the program of the second memory means for execution thereof in response to said input signals during game play.

2. The game controller according to claim 1 wherein said first memory means includes means which are not responsive to said second word format.

3. A method of controlling a series of microprocessor controlled games employing at least one moving object, the series of games having a common, prescribed type of game environment but different rules of specific game play, the method of controlling being responsive to input signals representing the interaction of the moving object with objects of the game environment and comprising the steps of:

(a) programming in hardware a first memory in a first word format for controlling game operation of the entire series of games according to said common type of game environment substantially independently of the specific rules of play of an individual game;

(b) programming in firmware a second memory in a different word format for controlling game operation dependent upon the specific rules of play for the given game; and

(c) interpreting during game play the programming of the second memory by the programming of the first memory for execution thereof in response to said input signals.

4. The method according to claim 3 wherein said step of programming a first memory includes programming the first word format in machine language and said step of programming a second memory includes programming the different word format in a higher level language which will not access the machine language.

C