

EUROPEAN PATENT APPLICATION

Application number: 82101801.7

Int. Cl.³: **G 07 C 15/00**
A 63 F 3/08

Date of filing: 08.03.82

Priority: 19.03.81 US 245090
18.02.82 IL 65048

Date of publication of application:
29.09.82 Bulletin 82/39

Designated Contracting States:
AT DE GB IT SE

Applicant: Gat, Yoel
32 Bialik Street
Ramat Gan 52 451(IL)

Applicant: Pasternak, Eliezer
11 Mania Shohat Street
Holon(IL)

Inventor: Gat, Yoel
32 Bialik Street
Ramat Gan 52 451(IL)

Inventor: Pasternak, Eliezer
11 Mania Shohat Street
Holon(IL)

Representative: Meddle, Alan Leonard
FORRESTER & BOEHMERT Widenmayerstrasse 4/1
D-8000 Munchen 22(DE)

Electronic game apparatus and methods of selecting combinations of events.

Gaming apparatus selects combinations of outcomes of events based on predetermined probabilities of outcomes of each event. In a Best Chance mode the most likely combinations are identified. In a Luck mode a random selection of lottery is effected using the predetermined probabilities. The apparatus (10) includes a display (15) having optical read-outs for each game whereby the selected combinations are readily identified.

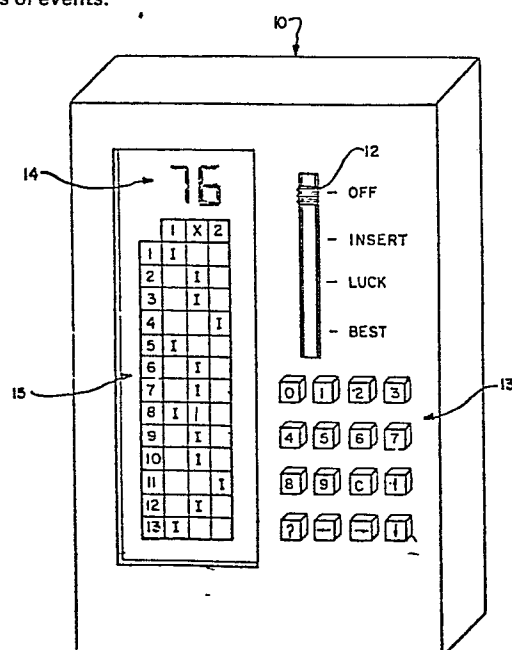


FIG. -1

This invention relates generally to gaming devices, and more particularly the invention relates to the selection of possible combinations of outcomes of a plurality of events based on probability and
5 on a random selection.

Considerable speculation is involved in the outcome of sporting events on an individual basis and on a combined basis. For example, with soccer leagues where many games are played during the
10 same time period, special rewards may be vested on an individual who can select the highest number among participants of correct outcomes of games played. It is usually common for an individual to make several selections of combinations of outcomes.
15 In arriving at the several selections, educated guesswork mixed with personal bias and some arbitrary decisions may be employed. Arriving at satisfactory combinations is usually difficult and time consuming when done manually.

20 Accordingly, an object of the present invention is means for assisting in determining combinations of outcomes of such events based on probability and on random selection.

Another object of the invention is means
25 for rapidly determining outcome selections of such events in a descending order of probability.

Still another object of the invention is means for providing random selections of outcomes with a predetermined bias.

Yet another object of the invention is a
5 method for automatically computing selections of events based upon probability data.

Briefly, in accordance with the invention, apparatus selects combinations of outcomes of events such as games based on predetermined probability
10 of outcome of each events. To achieve most and next likely combinations of outcomes the predetermined probabilities are stored in memory means, and computation means receives the predetermined probabilities and first selects the most likely
15 combination of outcomes. The computation means then sequentially selects the next most likely combinations based on the predetermined probabilities. Display means displays each combination of outcomes.

Preferably, the computation means comprises
20 a programmed microprocessor which can sequentially select the most likely combinations ("best" mode) and also can randomly select combinations of outcomes based on predetermined probabilities ("luck" mode). The random selection in a preferred embodiment is
25 achieved by a pseudorandom sequence formed by a Linear feedback shift register (LFSR).

The reading is compared with the stored probabilities to determine the selected outcome.

In a preferred embodiment, a mode switch is provided for selecting a data insertion mode of operation, a "best" mode of operation, and a random or "luck" mode of operation. A keyboard is provided for inserting data, and a columned display provides the selected combinations.

Doubles (meaning 2 outcomes of a single game) and triples (meaning all 3 outcomes of a single game) can be generated in the display from "luck" mode or from "best" mode.

The invention and objects and features thereof will be more readily apparent from the following detailed description and appended claims when taken with the drawing in which:

Fig. 1 is a perspective view of selection apparatus in accordance with one embodiment of the invention;

Fig. 2 is an electrical function block diagram of the apparatus of Fig. 1;

Fig. 3 is a general flow chart of the operation of the apparatus of Fig. 1;

Fig. 4 is a data format of a combination counter in the apparatus of Fig. 1;

Fig. 5 is a flow chart of a next likely combination algorithm;

Fig. 6 is a flow chart of a mode search program;

Fig. 7 is a flow chart for normalizing outcome probabilities;

Fig. 8 is a flow chart of a data insert program;

5 Fig. 9 is a flow chart of initializing the random number generation process formed by a LFSR (linear feedback shift register);

Fig. 10 is a flow chart of moving the LFSR;

10 Fig. 11 is a flow chart of a generation of a random number;

Fig. 12 is a flow chart of a normalization example of the random number (changing an 8 bit random number to the range of 0+99);

Fig. 13 is a flow chart of luck mode program;

15 Fig. 14 is a flow chart of a double generation from BEST mode;

Fig. 15 is a flow chart of a triple generation from BEST mode;

20 Fig. 16 is a flow chart of a double generation from LUCK mode;

Fig. 17 is a flow chart of a triple generation from LUCK mode;

25 Fig. 18 is a flow chart for data insert and normalization of the second embodiment of the invention;

Fig. 19 is a flow chart of LUCK mode program of the second embodiment of the invention;

Fig. 20 is an electrical schematic of another embodiment of the invention.

The invention will be described with reference to 2 embodiments: first, for selecting
5 combinations of outcomes of 13 soccer games in which the home team can win (designated 1), the visiting team can win (designated 2), or the game can end in a tie or a draw (designated by X). The user first formulates his guess as to the likelihood of the three
10 possible outcomes for each game, and he inputs three numbers representing the likelihood of a home team victory (1), a visiting team victory (2), or a tie (X). Each of the numbers ranges from 0 to 99 with a 0 representing the least likelihood and 99 representing
15 maximum likelihood for the event to occur. The three numbers will be normalized whereby the sum equals to 100 (representing unity), as will be described below. After all numbers are inserted for all games and the inserted numbers are normalized for each game, the
20 most likely combination of selections can be identified by selecting the outcome associated with the highest probability of each game. The selection of the next likely probability is identified by the product of probabilities for all games and sequentially
25 selecting several combinations in a descending probability order. Several combinations may have the same probability, and each is identified. Alternatively, a "Luck" mode is provided whereby the outcome of each

game is individually selected with the aid of a random number generation. The reading is compared with the stored probabilities to determine the selected outcome. Doubles and triples can be generated from "luck" and
5 from "best" modes, by keying "2" or "3" key in the keyboard.

The second embodiment relates to selecting combinations of outcome of 55 soccer games in which 8 games that would end in scored draw (2:2, 1:1 etc.)
10 need to be selected.

In this embodiment the user may formulate his guess as to the likelihood of all or several of the possible 4 outcomes: 1- home team victory, 2- visiting team victory, X- unscored draw (0:0) and
15 XX--scored draw for every game. This is done as in the first embodiment. Whenever the user does not formulate his guess, best statistical results (regarding to the last few years for example) are entered automatically. After all numbers are inserted and normalized, the most
20 likely combination can be identified by selecting the outcome associated with the highest probabilities of scored draws. In a similar matter next likely combination could be chosen. Alternatively in a "luck" mode, game numbers are individually selected with the
25 aid of the random number generator, in such a way that the probability for each possible game number (to appear on the display) is proportional to the chance estimate entered for score draw for that game.

HARDWARE

Referring now to Fig. 1, one embodiment of the invention for selecting combinations of outcomes for 13 soccer games, as above identified, is illustrated generally at 10. The apparatus includes a four position switch 12 for power off, data insert, "luck" mode operation, and "best" mode operation. A sixteen digit keyboard 13 is used for inserting data, for registering combinations of outcomes in the "best" mode, and for requesting game outcome selections during the "luck" mode of operation. An optical display 14 consisting of two seven segment light emitting diodes provides a readout of the input data, and an optical display 15 provides a readout of a combination for 13 games. Associated with each game are three light emitting diodes which are selectively illuminated (as indicated by an I) to indicate a home team victory (1), a tie (X), or a visiting team victory (2). Thus, the various combinations as determined by the apparatus are provided upon request on the display 15 for ready use by the operator.

In the insert mode of operation, the apparatus is energized and the system is ready to receive and store input data. The user enters numbers representing the chances for the three possible outcomes of each game, i.e. the numbers are proportional to the user's estimate of the probabilities of the outcomes. The numbers are entered by pressing the

keyboard 13, and are displayed on the numeric display 14.

In the luck mode and best chance mode, the various combinations are illustrated on the display 15 as a function of the numbers which have been inserted and stored during the insert mode of operation. In the luck mode, the generation of output combinations is based on an electronic lottery in which the probability for each possible outcome is proportional to the chance estimate entered for that outcome.

Each stroke of the key marked "c" in the keyboard 13 causes a lottery of a new outcome combination on the display. Therefore, after each stroke, a complete column is displayed. In the best chance mode of operation, all of the 3^{13} possible combinations may be successively displayed one after another by depressing the "↓" key, and the combinations, in a descending order of probability, are provided on display 15 in response to each depression of the down key.

Doubles and triples can be generated by a stroke on keys "2" or "3" accordingly. Each stroke of "2" key adds a second outcome to one of the games. Each stroke on "3" key selects all three outcomes of one of the games.

When the preferred embodiment is used for selecting combinations of outcome for 55 soccer games (in order to choose 8 scored-draws), it works in a similar manner.

In the insert mode of operation, the user may enter his chances estimates for each possible outcome of each game. Whenever probability assignments are not entered by the user, predetermined probabilities based on statistical knowledge are entered
5 automatically.

In the Best chance mode the game numbers are individually displayed on the display 14 in a descending probability order. Each time "c" key is
10 stroked, a new game number is entered to the display.

In the Luck mode, the generation of game numbers is based on an electronic lottery in which the probability for each possible game number is proportional to the chance estimate entered for
15 scored-draw for that game.

Referring now to Fig. 2, an electrical functional block diagram of the embodiment of Fig. 1 is illustrated. The apparatus includes a 6 volt power supply 16 which is interconnected to a microprocessor
20 17 along with the keyboard 13, the alpha numeric display 14, and the display 15 comprising three columns of light emitting diodes. In this embodiment the microprocessor 17 is an R 6500/1 microcomputer, manufactured by Rockwell International of Anaheim
25 California. This family of microcomputers contains several I.C.s that differ in the size of the program ROM, DATA RAM, I/O lines and on chip peripherals.

The R 6500/1-10 microcomputer contains 2048 bytes of Program ROM, 64 bytes of DATA RAM and 32 I/O lines.

The R 6500/1-11 microcomputer contains
5 3072 bytes of Program ROM, 192 bytes of DATA RAM and 32 I/O lines.

The R 6500/1-10 microcomputer suits the first embodiment and the R 6500/1-11 suits the second embodiment (as well as the first).

10 The keyboard 13 is a 4 by 4 matrix with 4 input lines I_1-I_4 and 4 output lines O_1-O_4 interconnecting the matrix with the microprocessor 17. The input lines are connected to pins 35-38 of the microprocessor, and the keyboard outputs are
15 connected to the microprocessor pins 31-34. The keyboard output lines are read after each "high" setting of one of the four keyboard input lines. After all output lines have been successively set high a unique identification of any single key
20 depression is accomplished. Resistors 18 are pull-down resistors to ensure "low" state in the non-activated line.

The two digit display 14 is provided to display the numbers entered in the "insert" mode and
25 to display the serial number (modulo 100) of each combination displayed in the "best chance" mode. The two digit display may comprise a seven segment LED

display such as the HP 5082-7740 of Hewlett Packard.

The display is driven by a BCD to seven segment
latch-decoder-drivers 8 such as an RCA CD 4511B.

Pins 4-9 of the microprocessor 17 control the drivers
5 8 with the digits being separately set by the setting
of a BCD code at pins 6-9 and then temporarily setting
pins 4 and 5 low according to the digit selected to
be updated.

The light emitting diode array 15 consists
10 of 39 LED's arranged in 13 rows with 3 LED's per row,
assuming 13 games in each column with each game having
three possible outcomes. Each LED displays one possible
outcome for each game. The LED's are driven by an
array of low power logic inverters 20 such as the
15 Texas Instruments SN74LS05, which is a hex inverter,
open collector LS-TTL device. Each inverter output
pin is connected to one row of LED's via a resistor
21 for limiting the current. The anodes of the LEDs
are each connected to a collector of one of PNP
20 transistors 22, 23 or 24. The transistors are 2N2904
devices. An LED is on when the interconnected
transistor is conducting and the inverter's output
line is in a "low" state (corresponding to a "high"
output at the appropriate microprocessor output pin).
25 Thus, the microprocessor 17 controls the three
transistors 22-24 whereby only one transistor can
conduct at a time and a combination of "high" and
"low" levels appears on the microcomputer output

pins 13-29 and one column with any combination of LED states is displayed at one time. The three columns are successively selected through transistors 22-24 at a high rate whereby the combinations can be
5 visually observed.

The three operational modes of the apparatus are selected via the switch 12. Two microprocessor input lines are connected to the switch and to the pull-up resistors 26. The switch can connect one of the
10 lines to ground thus providing the distinguishable combinations necessary to define the three operational modes: insert, best chance, and luck. A crystal oscillator 27 provides a clock frequency of four megahertz to the microprocessor 17. Power is turned on
15 and off by the switch 28, and a silicon diode 29 such as a 1N4001 connects the V_{BB} line to the V_{CC} line to provide a voltage drop of about 0.8 volt at 100 milliamps and thus provides V_{CC} in the allowable range for the microcomputer and the logic devices. The reset pin 39
20 of the microprocessor is connected to the V_{CC} line through resistor 30 and diode 31. When the switch 28 is turned on after a period of being off, the reset line is held "low" by the charge on capacitor 32. This forces the reset procedure in the microprocessor.
25 Resistor 30 charges the capacitor to release the reset condition. Diode 31 provides a fast discharge pass for the capacitor, once the switch 28 has been turned off,

0061052

and thus ensures a new reset condition even after a short off period. Capacitor 33 is provided to reduce noise in the V_{CC} line.

For the second embodiment of the apparatus
5 slight changes in the hardware improve the man-machine interface. Four seven segment LEDs are provided, two for the probability assignments and two for the game number. Instead of the 30 discrete LEDs, four are adequate in order to distinguish between outcome 1,2,X
10 and XX.

Memory Allocation for the first embodiment
(13 soccer games)

	<u>Function</u>	<u>No. of Bytes</u>	<u>Name</u>	<u>Comments</u>
15	1) Counter	4	$C_1 + C_4$	Checking all possible 3^{13} combinations
	2) Probabilities	26	$P_1 + P_{26}$	Outcome probability [1]
20	3) Virtual Prob.	2	LV	Current probability [2]
	4) Next Prob.	2	LN	Probability at next counter state [3]
25	5) Current Display Buffer	5	CDB	
	6) Old Display Buffer	4	ODB	[4]
	7) New Display Buffer	4	NDB	
30	8) Temporary Prob.	2	LPC	Probability of the combination represented by the counter
35	9) Subroutine	6		Stack Memory for Subroutine [5]
	10) Temporary	9		Temporary Cells

Notes

- [1] Only 26 outcome probabilities are needed for 13 games since the third probability for each game can be calculated by: $P_2 = 100 - P_1 - P_x$
- 5 [2] Virtual probability - 16 bit number (2 bytes) representing the logarithmic probability of the column now being displayed. The probability of a column is the product of all 13 probabilities associated with it. The logarithmic probability of a
- 10 column is the sum of all 13 probabilities associated with it after converting them to logarithmic numbers.
- [3] Next probability - 2 bytes of a logarithmic probability representing the next virtual probability.
- [4] Display buffers - RAM cells contain all necessary
- 15 data for the display subroutine. The display includes the two digit numeric display 14 of Fig. 1 and the 39 LED's of display 15. Four bytes are required for the 39 LED's and one byte for the two digit numeric display. The two digit numeric display data is not retained
- 20 for the "new" and "old" display buffers, thus, only four bytes are required for each.
- [5] Subroutine - Six bytes are required for storage of main CPU registers when the interrupt subroutine is initiated.
- 25 ROM The ROM size of the microprocessor is 2048 x 8 bit or 2K bytes. 1.5K bytes will be dedicated to software, 1/4 K bytes to the logarithmic

table and 1/4K bytes to incrementing the column counter through all 3^{13} possible outcome columns.

SOFTWARE (for the first embodiment)

Having now described the hardware of the apparatus, consider now the software for controlling the apparatus in implementing the several modes of operation. Fig. 3 is a general flow diagram of the apparatus operation.

When switch 12 is turned on (from off to insert) the apparatus goes through a reset and initialization procedure and then enters a "Cold Insert" mode. In this mode all of the 39 chance values (3 values for each one of the 13 games) can be entered by the user.

When all the chance values have been entered, the program is looking for a change in the main panel keys and jumps to the appropriate section as follows:

1) WARM INSERT: If the main switch 12 remains in the INSERT mode the user can modify the 39 chance values by means of the 4 x 4 keyboard 13.

2) LUCK: If the main switch 12 is moved to the "LUCK" mode the program first normalizes the 39 chance values into probabilities. The normalization maintains the relationships between the three chance values for each game but converts the values so that the sum of the three numbers equals 100. The three new numbers now represent the probabilities for each of the three possible outcomes of each game. The user

can now perform the weighted lottery by pressing the "C" key of keyboard 13. Each key stroke can result in a lottery of a game or new row in the display. Thus, after 13 strokes, a complete combination for
5 13 games are displayed. However in a preferred embodiment all 13 games are simultaneously displayed after one key stroke.

3) BEST CHANCE: Again, as in the "LUCK" mode the chance values are normalized into probabilities,
10 then the maximum likelihood (ML) combination is identified and displayed and its probability calculated. Then, by repeatedly pressing the down key the user can display all 3^{13} possible combinations in a descending order of probability.

15 The probability of each combination can be calculated by multiplying all 13 outcome probabilities associated with it. Since the microprocessor inherently performs add and subtract operations it is more efficient to convert all possible probabilities in the range of
20 0 to 99 to logarithmic numbers of 16 bits and perform summation rather than multiplication. The range of numbers in the table will allow summation of the 13 logarithmic probabilities without overflow.

In order to check all possible combinations
25 (3^{13}) a counter format is defined as shown in Fig. 4. Each 2 bit combination in Fig. 4 represents the outcome of a specific game in the following manner:

	<u>Outcome</u>	<u>Bit Designation</u>
	1	00
	x	01
	2	10
5	Invalid	11

Since the microprocessor cannot count modulo 3 the counter is advanced through a ROM table. The entry to the ROM table is one of the 4 x 2 bit cells ($C_1 - C_4$), and the output of the table is the
 10 binary difference needed to increment the counter modulo 3. This difference is added to the current cell content. The status of the carry bit determines whether the next significant byte of the counter should be advanced.

15 Next Likely Algorithm

This algorithm is based on the fact that many combinations will be defined for any given probability. The total number of possible combinations is 1,594,323 (3^{13}) while only 65,536 different
 20 probabilities may exist since the logarithmic sum is arbitrarily limited to 16 bits. Thus, there are 2^{16} possible values. If the probabilities are distributed uniformly then about 24 combinations will have the same probability. When the combinations counter is
 25 manipulated the microprocessor searches another combination with the same probability and displays it next. While running through the counter possibilities,

the microprocessor also searches for the next (lower) probability that would be the virtual probability of the next counter cycle (LN). It will be appreciated that there is a trade-off between the time required
5 for a complete cycle and the accuracy of the comb. probabilities. A 13 bit accuracy per logarithmic probability for each game (and 16 bit accuracy for the combination probabilities) will result in several seconds for identifying the next combination to be
10 displayed, with a 2MHz clock.

Next Most Likely Combination

The following description relates to the numbered blocks on the Next Most Likely Combination flowchart diagram of Fig. 5.

15 1) After entering the "BEST CHANCE" mode by means of the main panel switch 12 the maximum likelihood (ML) combination is found by selecting the outcome with the highest probability for each game. Several ML combinations may exist if the same probability
20 is assigned to two outcomes of a given game. It is sufficient to identify one of these ML combinations and calculate its probability. This probability is inserted to LV.

25 2) The combination counter is cleared and the next probability is loaded into LN. The program returns to point A after incrementing the counter.

 3) Calculation is then made of the probability of the combination that is represented by

the counter. All 13 logarithmic probabilities of the outcomes associated with this combination are summed and the results reside in LPC. It should be emphasized that logarithmic scaling is so designed that the
5 higher the probability the lower the logarithmic representation.

4) LPC is compared with LV. If they are equal then this combination is displayed next. If the new display buffer is full (5) the program waits until
10 the user requests the next combination (7), (8), and only then are all buffers updated and the new combination is inserted to the new display buffer (9). The program then increments the counter (12). If the new display buffer is empty NDB is updated (6) and then
15 the program increments the counter (12). This approach is used to save time. The CPU identifies the next combination before it is requested by the user. If the buffer is full when the column is found, the waiting loop (7) goes into effect.

20 10) If LPC is greater than LV, check if it is necessary to change LV, if it is ($LPC < LV$), change LV to LPC (11). Go then to increment the counter (12).

12) The counter is incremented by 1.
Incrementing the counter is performed by the ROM table
25 which executes the modulo 3 operation.

13) After incrementing the counter a check is made to determine if the cycle through all 3^{13} possible combinations is over. If all 3^{13} combinations

had been scanned it is necessary to define the new probability for the counter's next cycle. This is done by moving LN to LV and initiallizing LN (14). The program then returns to a new cycle starting from a
5 0 state of the counter.

Notes

- 1) The two digits in the display section will show the number of combinations between the present displayed combination and the first ML combination,
10 modulo 100.
- 2) When the user asks for the next combination, but this combination is not yet ready, the display will go off until it is ready in the current display buffer. This is accomplished by searching the keyboard for
15 "new combination request" every 20 millisecond by means of the timer interrupt.

SEARCH FLOWCHART

The mode search flowchart is shown in Fig. 6 and is provided in two separate locations. After "COLD
20 INSERT" the mode search determines the mode requested by the user. If the main panel switch 12 has not been moved to the "BEST CHANCE" or "LUCK" modes a "WARM
INSERT" (modifications of input data) is assumed. The interrupt subroutine, after displaying the present
25 buffer, checks if a change has been made in one of the main panel keys.

PROBABILITY NORMALIZATION

The user can input the chances of the three

outcomes for each game by means of 2 digit decimal numbers. The relative chance for each outcome is determined by the ratio between the chance numbers associated with the appropriate outcomes. To convert
5 these chances into probabilities, the numbers have to be normalized so that the sum of the three probabilities is 100 (when the probabilities are represented in percentage form).

If the chances given by the user are X, Y, Z,
10 the normalized probabilities are x, y, z:

$$x = \frac{100 X}{X+Y+Z}$$

$$y = \frac{100 Y}{X+Y+Z}$$

15
$$z = \frac{100 Z}{X+Y+Z}$$

We define $S = X + Y + Z$ where S is represented in memory by a 16 bit word (4 BCD digits). The 8 leftmost bits are defined as SUM high and the
20 8 rightmost bits are defined as SUM low. The value of x (or y or z) is calculated by the equation:

$$Sx = 100 X.$$

Fig. 7 is a flow chart for normalizing outcome probabilities. Prior to the actual normalization
25 a rough normalization is performed. This step reduces the value of X, Y and Z by a factor of 2 until their sum does not exceed 100.

INSERT MODE PROGRAM

Fig. 8 is a flow diagram of the Insert Mode Program. This mode deals with the initial insertion of data or chances for the 13 games (COLD INSERT), or
5 changes to the data which were already input (WARM INSERT). As used therein LINK is a flag used to indicate whether the digit just inserted is the first or second in order to determine whether the game number pointer should be advanced. WARM/COLD Insert
10 is a flag indicating which part of the software should be used. This flag is erased after the INSERT routine is completed. When the "?" key is pressed, some fixed preset probabilities are entered to the active game indicated by the pointer.

15 For example, assuming $P(1) = 35$, $P(x) = 40$, and $P(2) = 25$, the program will update the game's probabilities and then increment the pointer to the next game.

When a given LED is blinking and waiting for
20 the associated probability to be inserted, pressing the appropriate keys of the keyboard will enter the data to the right location in memory and display buffer, shifting prior information in this byte to the left by 4 bits (1 decimal digit). If it is the second byte,
25 the pointer is incremented for inserting or modifying data for the next game.

There are two possibilities to exit this INSERT routine, as follows:

- (a) WARM INSERT - changing the main panel switch to "LUCK" or "BEST CHANCE".
- (b) COLD INSERT - completing the entry of all 39 chance values.

5 The pointer state is not affected by a transfer to another software section to allow convenient return through "WARM INSERT".

RANDOM NUMBER GENERATION

 In order to make lotteries, a random number
10 generator (RNG) is implemented in the preferred embodiment. The RNG comprises of 32 bits "Linear feedback shift register" (LFSR). The series generated by LFSR are called M-series or pseudo random series. The feedback is formed by XORing the 1,2,22,32 bits,
15 entering the results in the LSB of the LFSR after rotating the whole LFSR 1 bit left. Generation of random number is done by parallel sampling $n < 32$ bits from the LFSR forming a number with the range $0 + (2^n - 1)$. The randomness of such a process is discussed in
20 mathematical and technical books.

 Fig. 9 illustrates the initialization of the LFSR. For initial conditions 21 random bits are placed in the LFSR in order to distinguish between up to 3^{13} simultaneous users. Those random bits can be generated
25 in many ways. One example is by letting the μP timer free-run, and each time a probability is entered (in the INSERT mode), the LSB of the timer is being sampled and entered to the LFSR.

4 bytes of RAM (inside the μ P) form the LFSR, and the 21 random bits are entered, say, in the first 21 bits of the LFSR.

Fig. 10 illustrates the method of moving the LFSR. Bits 1,2,22,32 are being XORed and the result is stored in the carry bit. The 4 bytes are successively rotated left through the carry bit. A check of all zero is made (illegal state) and if by mistake such a situation arises, 1 is entered to the LSB (least significant bit).

Fig. 11 illustrates the process of a random number generation. The LFSR is moved 32 times. n bits are sampled. The number is normalized to the appropriate value.

An example of such normalization is shown on Fig. 12. Whenever a number in the range of 0 to 99 need to be generated, 8 bits are sampled (forming a number in the range 0 to 255). This number is 16 bits added to itself 99 times (forming a multiplication by 99), and the upper byte (most significant) of the 16 bit number contains the normalized result.

LUCK MODE FLOW DIAGRAM

Fig. 13 is a flow diagram of the Luck mode. When the user strokes "c" key 13 random numbers in the range of 0 to 99 are generated. For each number, the value (R) is compared to the probabilities of row I, i.e. $P(1)$ is the probability of outcome 1, $P(X)$ is the probability of outcome X, and $P(2)$ is the probability of outcome 2.

If $R < P(1)$ then "1" is the lottery outcome, else
If $R < P(1) + P(X)$ then "X" is the lottery outcome,
otherwise "2" is the lottery outcome.

The display buffer is being updated each time an
5 outcome is selected.

Double and Triple

Double - means two possible outcomes of a single game.

Triple - means three possible outcomes of a single game.

Doubles and Triples can be entered from within Luck and
10 Best modes, by stroking keys 2 and 3 respectively.

After an output combination is being selected the user
can add as many doubles and triples as he wishes, until
the display is full.

Fig. .14 illustrates the generation of a double
15 from Best mode. LP is the maximal double probability
at a certain stage, and L is a temporary probability.
I is the game number and J is the double number (there
are 3 in a game: 1,X; X,2; 1,2). After an initialization,
the double probability is being calculated as a multi-
20 plication between the two probability assignments.

If the probability exceeds LP, and the double
is not yet displayed, then this is a preferred double.
The loop is made on all J and I, and the best available
double is displayed,

25 The generation of a triple (Fig. 15) from
Best mode is similar. L is the multiplication of all
3 outcome probabilities in a game, and the loop is
on I (the game number) alone. The best available triple
is displayed.

Fig. 16 illustrates the generation of a double from Luck mode. First, SD is calculated as the sum of all 26 available doubles. P_{IM} is the probability of the outcome that has been selected for the I^{th} game in this current outcome combination. P_{IJ} represents, one of the two probabilities of an outcome that has not been selected: (one of the two available doubles). SD is calculated as the sum of the two available doubles in each game summed for all games.

A random number R is generated in the range $0 \div SD$. A summing procedure as in the first stage of Fig. 16, first on J (double number) and then on I (game number) takes place until the result is equal or greater (in the first time) than R. The number K_2 represents the double number (1 or 2), and K_2 represents the game number.

The display is updated so as to light the appropriate LED.

The generation of a triple from Luck mode is similar (Fig. 17).

ST is calculated as the sum of all available triples. A triple probability is the multiplication of all three normalized probabilities of a single game.

A random number R is generated in the range $0 \div ST$. A summing procedure takes place until the sum equals or is greater than R for the first time. K represents the game number in which the triple is generated.

Memory Allocation (for the second embodiment,
8 score-draws of 55 games)

	<u>Function</u>	<u>No. of bytes.</u>	<u>Name</u>	<u>Comments</u>
5	1) Score draw probabilities	55	$P(XX)_1 + P(XX)_{55}$	
10	2) Temporary game probability	4	$P(1), P(2), P(X)$ $P(XX)$	$P(X)$ -prob. for unscored draw $P(XX)$ -prob. for scored draw
	3) LFSR	4	$SR_1 + SR_4$	random number generator
	4) Virtual probability	2	IV	current prob.
15	5) Temporary for normalization	4	$N_1 + N_4$	
	6) Old display buffer	5	ODB	
	7) Current display buffer		CDB	
	8) New display buffer	5.	NDB	
20	9) Subroutine	6		
	10) Temporary	20		

More than 110 bytes of RAM are required, so a microcomputer with at least 128 bytes of internal RAM need to be chosen. A good example as in the preferred embodiment is Rockwell's R 6500/1-11 which contains 192 bytes of internal RAM.

Software (for the 55 soccer league game example)

The implementation of an apparatus for selecting several scored draws out of all games (in the example - 8 scored draws out of 55 games) is quite similar in principle to the example shown before. Most

of the changes are self evident. Several main differences will be emphasized in this section.

The hardware of the apparatus was mentioned earlier.

5 All the basic modes exist. INSERT mode (cold and warm) is shown on Fig. 19. When a game number is selected, predetermined probabilities obtained by best statistical result (of the last several years) are entered automatically to $P(1), P(2), P(X), P(XX)$. The
10 user may enter his chance estimates to all, part of, one or none of the available outcomes. For each possibility the value of $P(XX)_I$ is calculated. If for example, the user has changed only $P(1)$, then the new $P(XX)_I$ is

15
$$P'(XX)_I = \frac{P(XX)}{P(XX) + P(X) + P(2)} (100 - P(1)).$$

$P'(XX)$ is inserted to the appropriate place of the $P(XX)_I$ array (the I^{th} place). After the user finishes entering his chance estimates, he can activate Best or
20 Luck mode. In Best mode, game numbers are successively displayed in a descending probability order (of $P(XX)$). With each stroke of "C" key a new game number appears on the display.

Luck mode is shown on Fig. 19. At first
25 SE is calculated as the sum of all $P(XX)_I$. A random number R is generated (each time "c" key is stroked) in the range 0 to SE. A summing procedure takes place until for the first time the sum is equal or greater than R. K is the generated game number for that stroke.

Fig. 20 is an electrical schematic of another embodiment of the invention in which electronic means randomly selects outcomes of each game based on predetermined probabilities. The probabilities for each game are set by means of three potentiometers. Consecutive lotteries for each game can then be effected by pressing a button. Each button depression results in a lottery based on the potentiometers setting, and one of three LED's is turned on to indicate the selected outcome. The potentiometers are then modified to represent the probabilities for the next game.

In the circuit of Fig. 20 three CMOS NOR gates 80, 82, 84 have interactive feedback which create a ring counter. Each gate in the ring counter goes high in its turn for a period proportional to the resistance of the associated potentiometer 81, 83, and 85 while the two remaining gates are in the "low" state.

When the switch 88 is depressed, one of the LED's 100, 102, 104 associated with the gate in the high state is turned on. The three gates and their possible outcomes (i.e. 1, 2, X). Thus, a lottery based on the predetermined probabilities is performed by pressing the switch 88 with the potentiometers set according to the desired probabilities.

More specifically, if the output of gate 84 is high, then a high is also initially forced at the input to gate 82 through capacitor C_3 thus

resulting in a low output of gate 82. After a charging period proportional to $(R_{83} + R_{83'}) C_3$ the voltage at the input to gate 82 will go to "low". The other input to gate 82 is in low since the output of gate 84 imposes low at the output of gate 80 which in turn is connected to the input of gate 82.

When C_3 is sufficiently charged to force low the gate 82 input, the output of gate 82 goes high and forces the output of gate 84 to low state. At this point a similar process begins through capacitor C_2 which forces gate 80 to go high. Thus, the three gates comprise a recycling ring counter. The total rate of oscillations is designed to be higher than human response to ensure a random lottery.

When switch 88 is depressed the oscillations cease and the LED associated with the gate in the high state is turned on as long as the switch 88 is depressed. During the oscillations the third input of each gate is connected to "low" through resistors 91, 93, 95 and resistor 98. When the switch 88 is pressed the collectors of the transistors 90, 92, 94 receive high states but only the one associated with the gate in high is conducting and the associated LED is turned on. The conducting transistor also forces low the third input of the gate to which it is connected. The two other transistors being in the non-conducting state force high the inputs of their associated gates or low at the gate outputs. This

arrangement latches the state of all gates as long as the switch 88 is depressed. When the switch 88 is released the third input to each gate goes low to restart oscillations.

- 5 Selection apparatus in accordance with the invention is useful in quickly determining combinations of outcomes of sporting events such as soccer games. However, while the invention has been described with reference to two embodiments the description is
- 10 illustrative of the invention and not limiting of the invention. For example, low power consumption can be realized through use of a CMOS processor and LCD display, and the visual display can be replaced by or used with a printer or sound output. Improved accuracy can be
- 15 realized with a 16 bit microprocessor or by increasing computation time. Thus, various modifications and applications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

CLAIMS

1. Apparatus for selecting combinations of outcomes of events such as games based on a predetermined probability of outcome of each event comprising:

means for storing predetermined probabilities of outcomes of each event,

means for selecting outcomes of events based on said predetermined probabilities, and

means for displaying said outcomes.

2. Apparatus as defined by Claim 1 wherein the selection of combinations of outcomes include random selection means.

3. Apparatus as defined by Claim 2 wherein said random selection means comprises a multistage ring counter.

4. Apparatus as defined by Claims 1-3 wherein said means for selecting combination of outcomes include means for receiving said predetermined probabilities and selecting the most likely combination of outcomes based on said predetermined probabilities, and

means for sequentially selecting the next most likely combinations of outcomes based on said predetermined probabilities.

5. Apparatus as defined in Claim 4 wherein said means for selecting the most likely combinations of outcomes and next most likely combinations of outcomes includes a micro-processor.

6. Apparatus as defined by Claims 2-5 wherein said random selection means includes counter means for cyclically generating a count and comparator means for comparing said count to said predetermined probabilities of outcome for each game.

7. Apparatus as defined by Claim 2-6 and further including a mode switch for selecting a data insertion mode of operation, a best combination mode of operation, and a random selection mode of operation; a keyboard for inserting data, and display means for displaying each combination of outcomes.

8. The apparatus as defined in Claim 2 wherein said random selection means includes linear feedback shift register means.

9. The apparatus of the prior claims wherein said display means includes:

- a plurality of columns,
- a plurality of rows,
- each of said columns indicating a different possible outcome of the event signified by the row, and means for visually indicating the selected outcome by activating indicator means in the selected column.

10. The apparatus of Claim 9 including means for generating multiple selections along individual ones of said rows.

11. The apparatus of Claim 9 including means for generating multiple selections along a plurality of said rows.

12. A method of selecting combinations of outcomes of events such as games comprising the steps of
storing predetermined probabilities of outcomes of each event,
establishing the most likely combinations of outcomes of all events based on said stored predetermined probabilities,
sequentially establishing the next most likely combinations of outcomes based on said predetermined probabilities, and
visually displaying each combination of outcomes.

13. The method of selecting combinations of outcomes of events as defined by Claim 12 and further including establishing random combinations of outcomes of events.

14. The method of selecting combinations of outcomes of events as defined by Claim 12 or 13 wherein said step of establishing random combinations includes comparing said predetermined probabilities for each event with the count from recycling counter.

15. The method of selecting combinations of outcomes of events as defined by Claim 12, 13 and 14 wherein said steps of establishing most likely combinations and next most likely combinations include normalizing said predetermined probabilities, determining the logarithm of the normalized predetermined probabilities, and adding the logarithms for the normalized predetermined probabilities of combinations of outcomes.

16. A method of selecting combinations of outcomes of events such as games comprising the steps of
 storing predetermined probabilities of outcomes of each event, and
 randomly selecting combinations of outcomes of events with the random selections being weighted in accordance with said predetermined probabilities.

17. The method in accordance with Claim 16 wherein said step of randomly selecting comprises random stopping of a multi-stage ring counter, and said step of storing predetermined probabilities comprises adjusting the count duration of stages of said ring counter.

18. The method of Claim 16 including the steps of using equal predetermined probabilities of outcomes for each event thereby obtaining a random combination of outcomes.

19. The method of Claims 15 - 18 including the steps of establishing next most likely outcomes for at least one of the events, and displaying multiple selections of outcomes for said at least one of the events based on the most likely outcome and the next most likely outcome.

20. The apparatus and methods as described herein by way of example and with reference to the accompanying drawings.

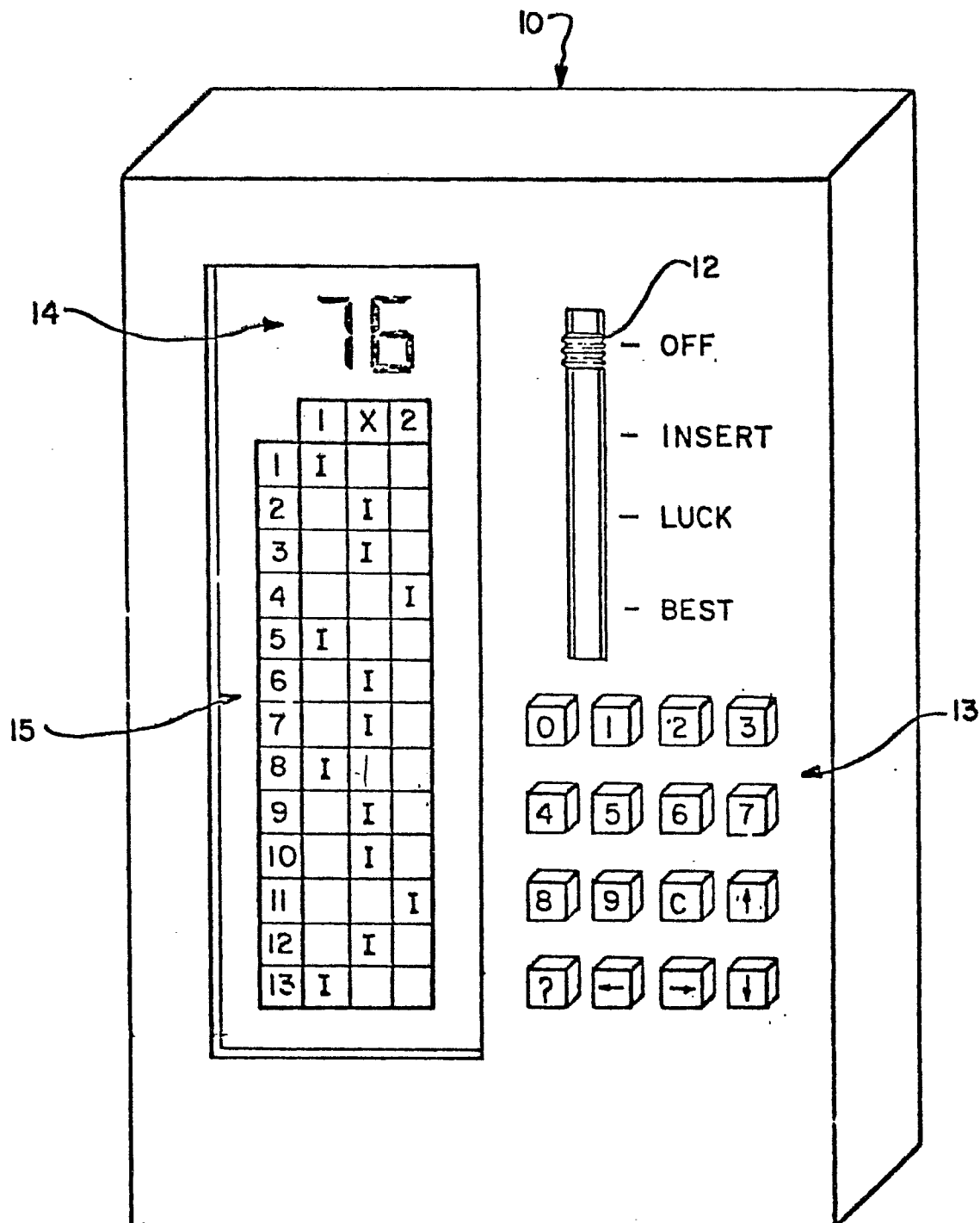
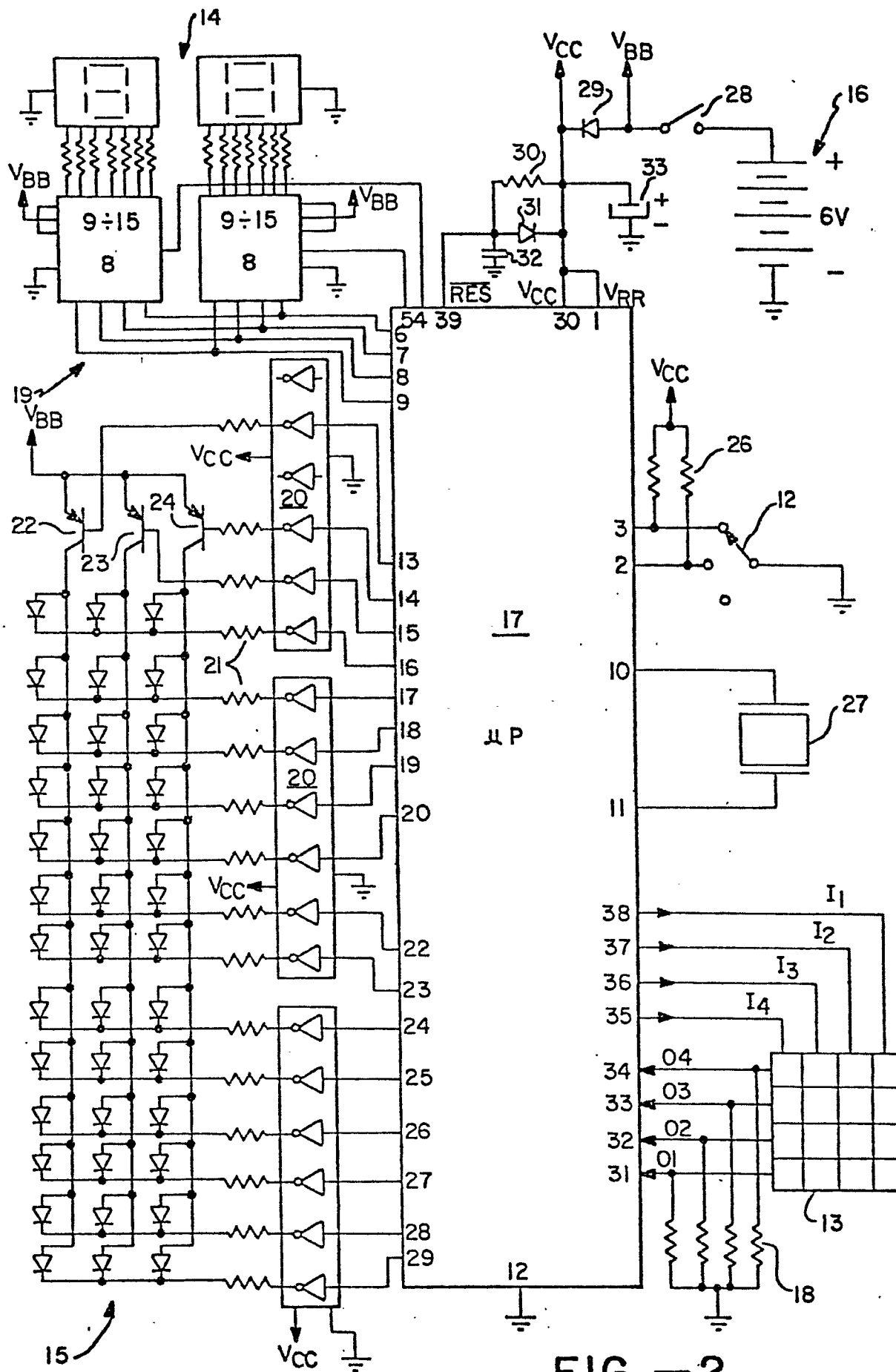


FIG. -1



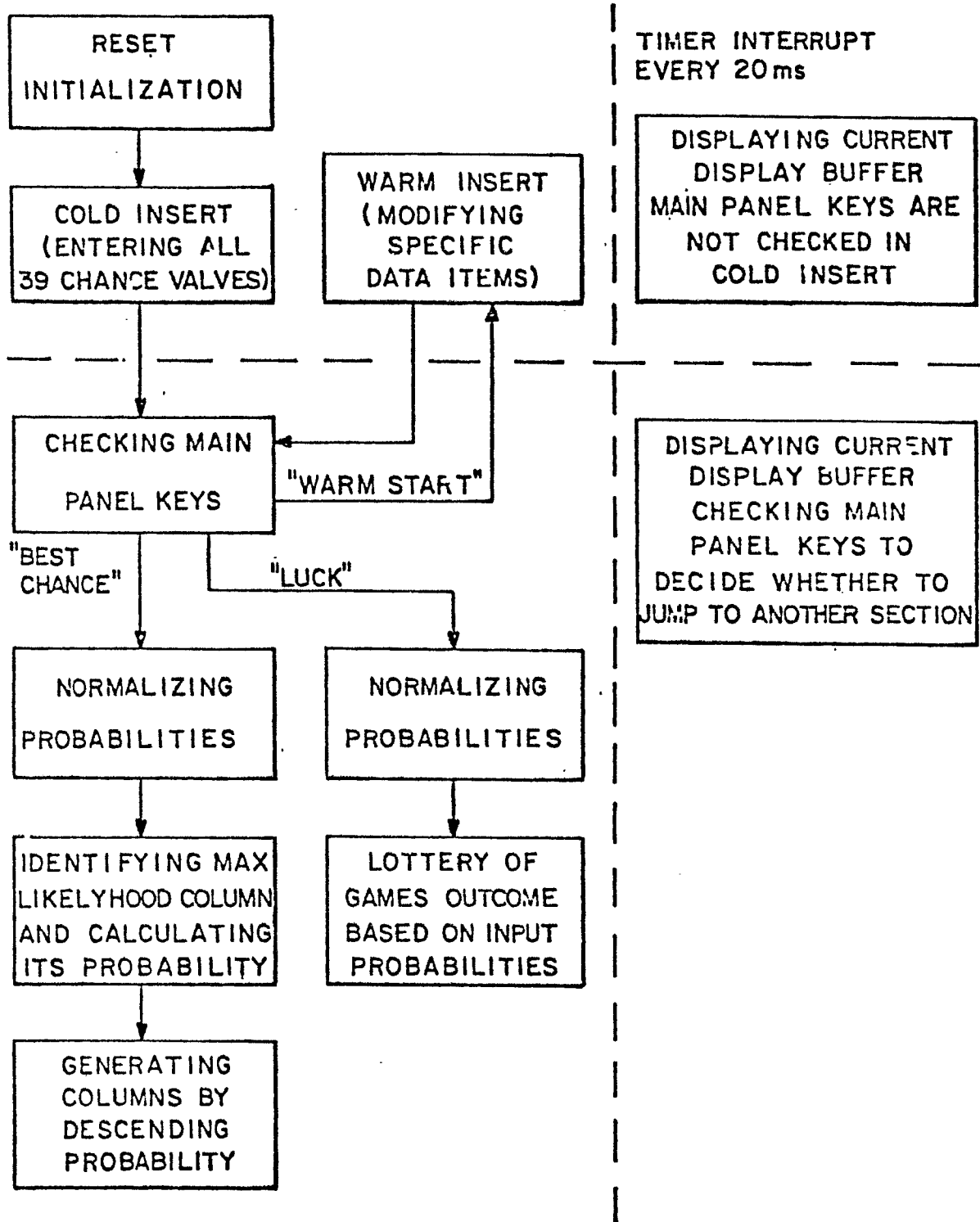


FIG.—3

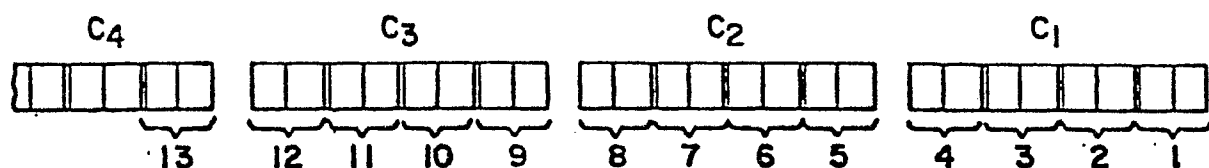


FIG.—4

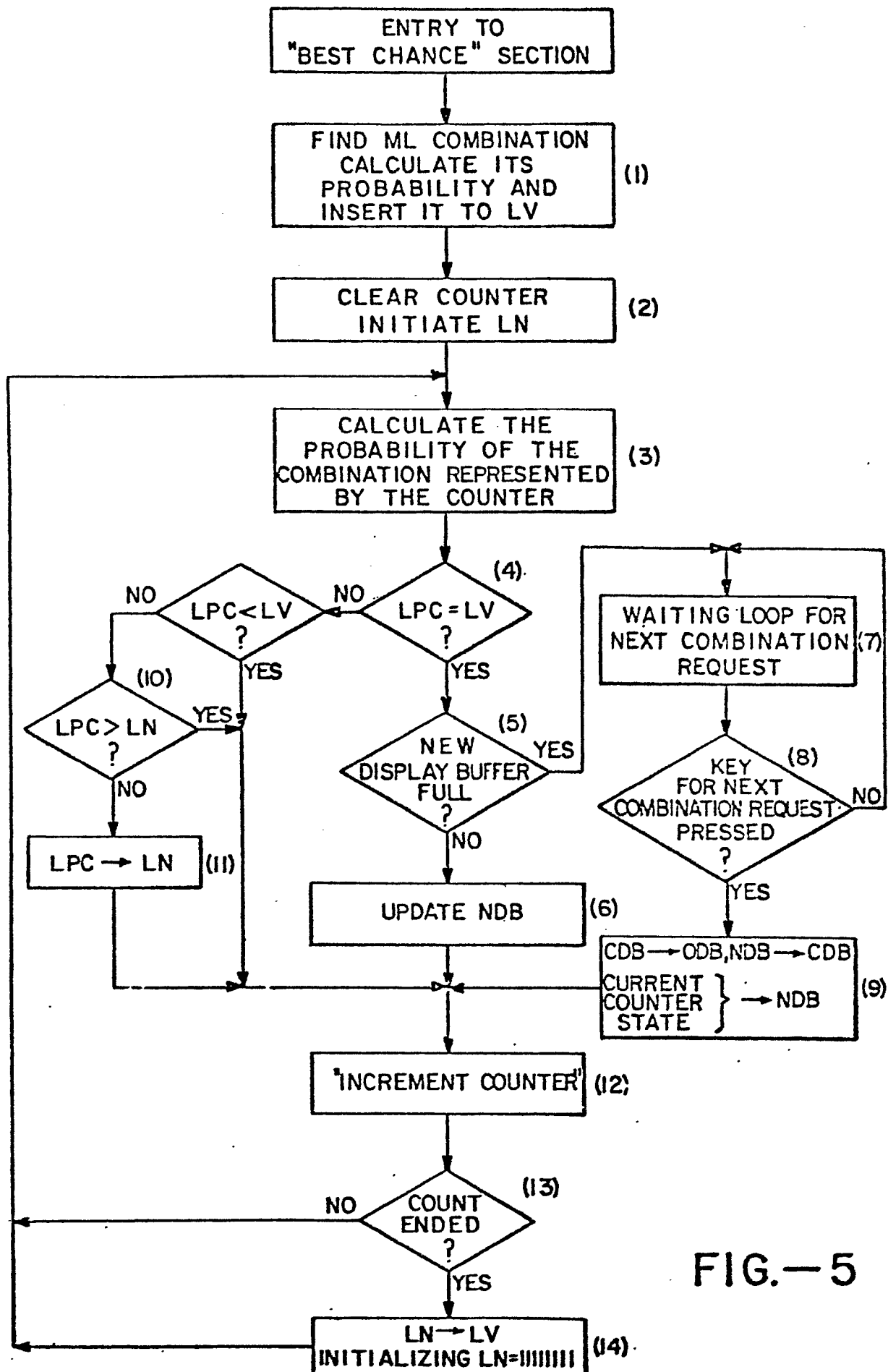


FIG.-5

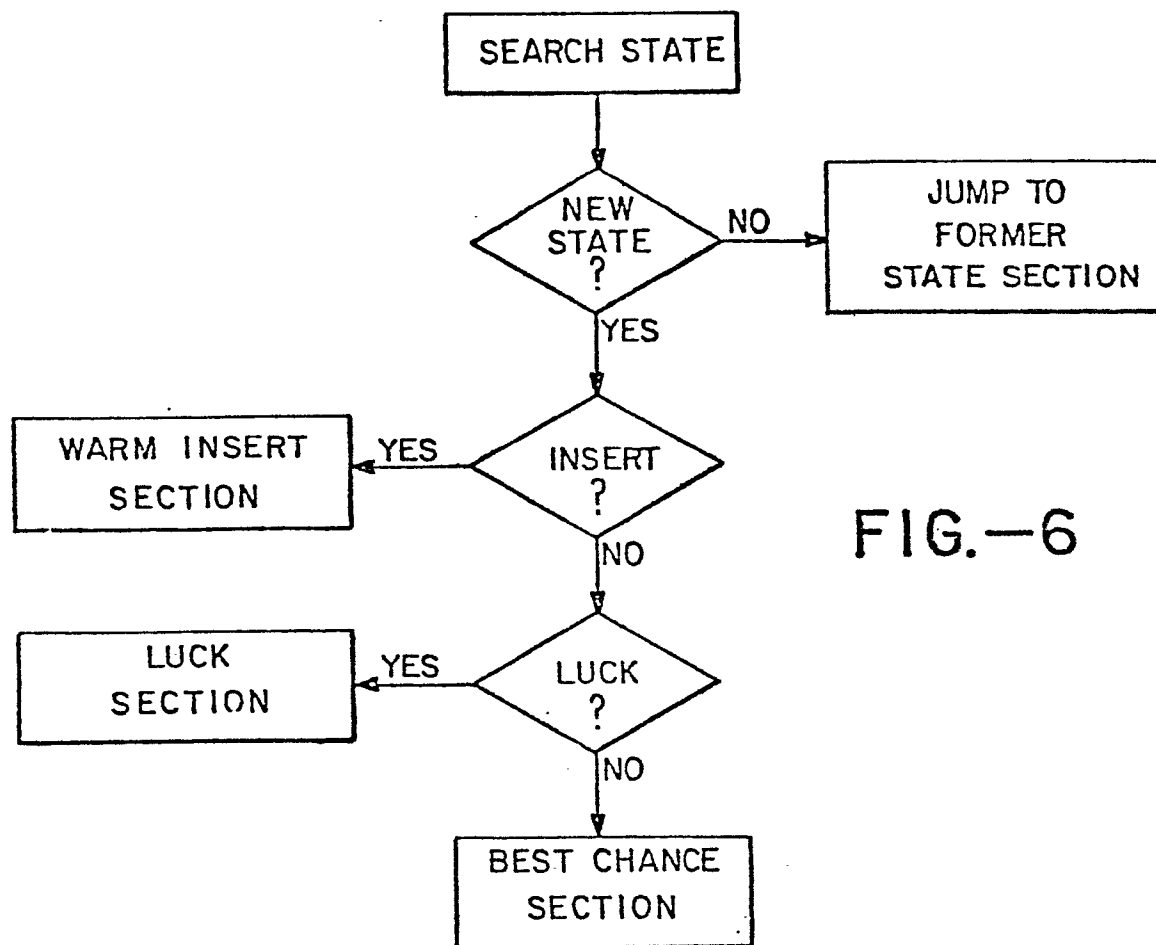


FIG.-7

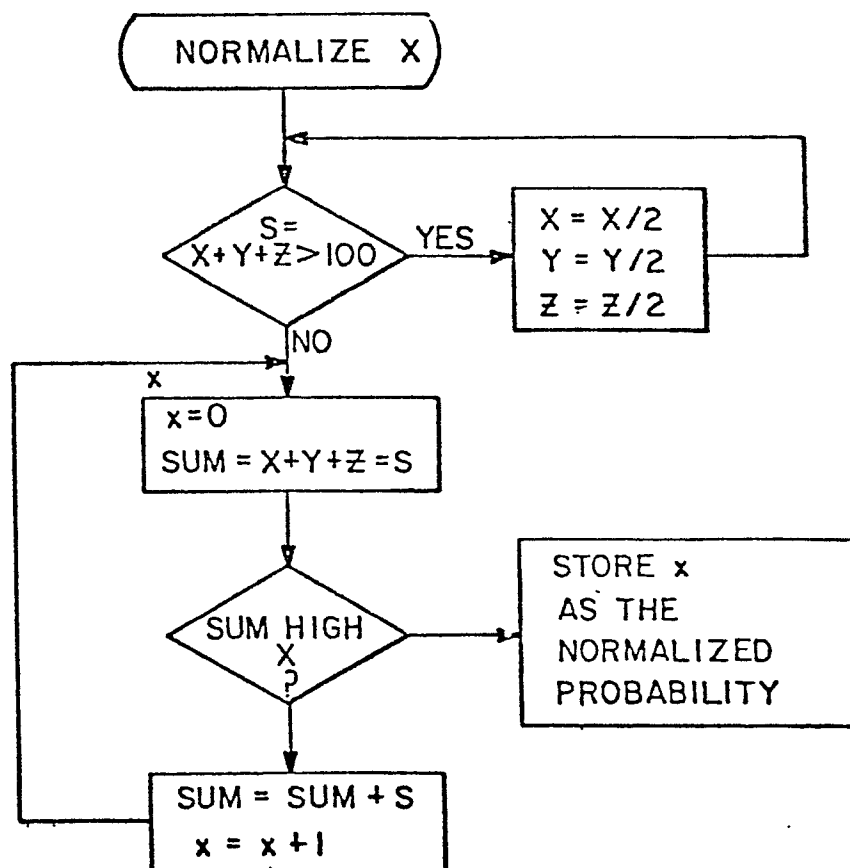
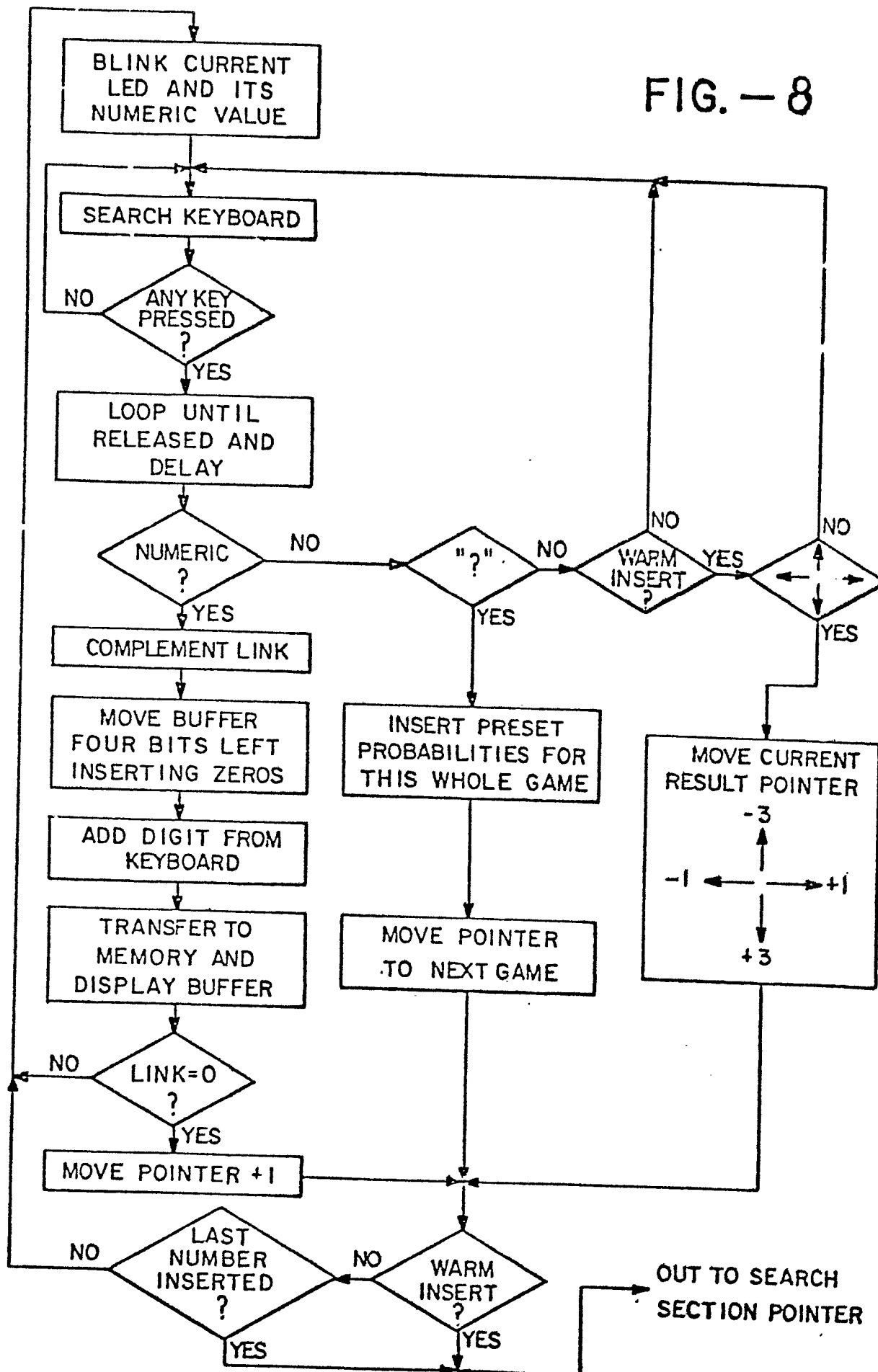


FIG.-8



RANDOM NO. GENERATION

initialization

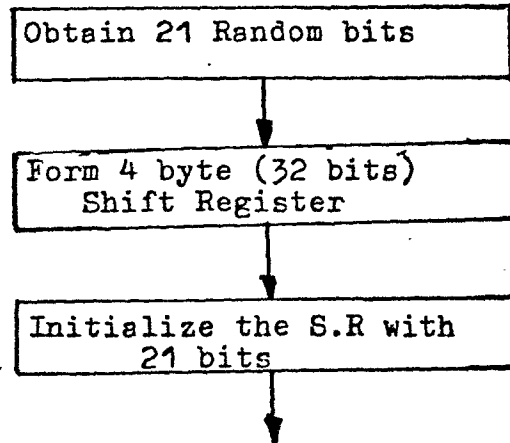
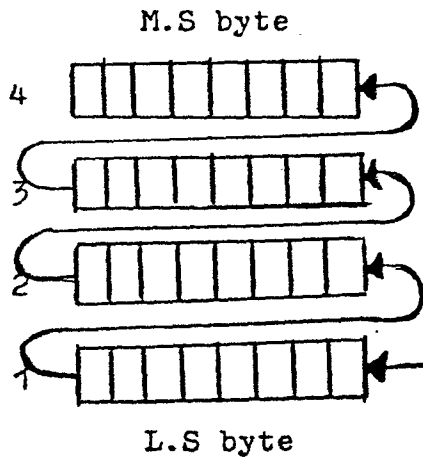
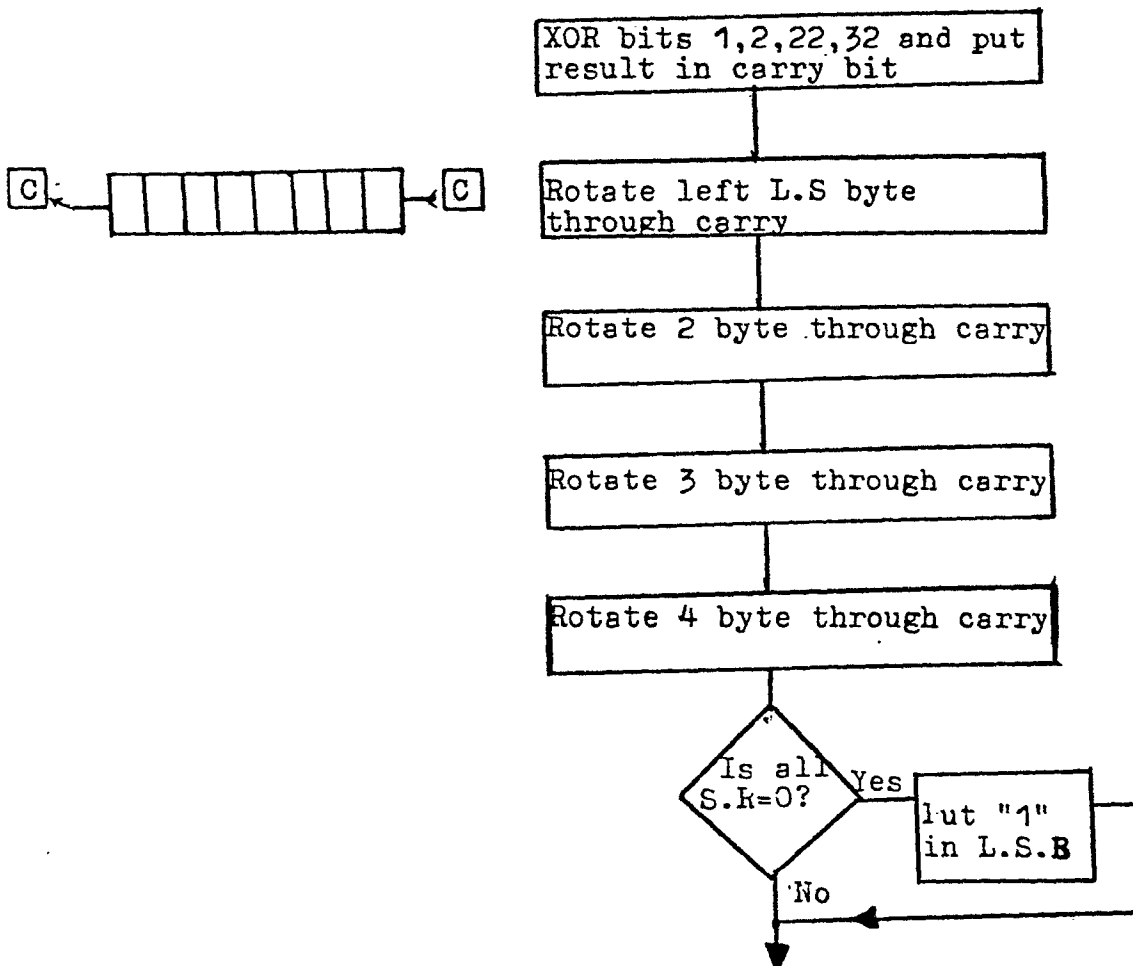


Fig-2

Moving the Shift Register



GENERATION A RANDOM NO. $0+R$

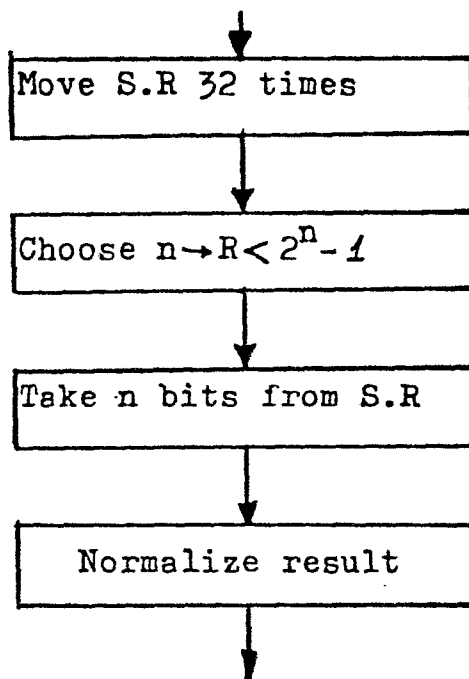


Fig-11

An example for normalization for $0+99$

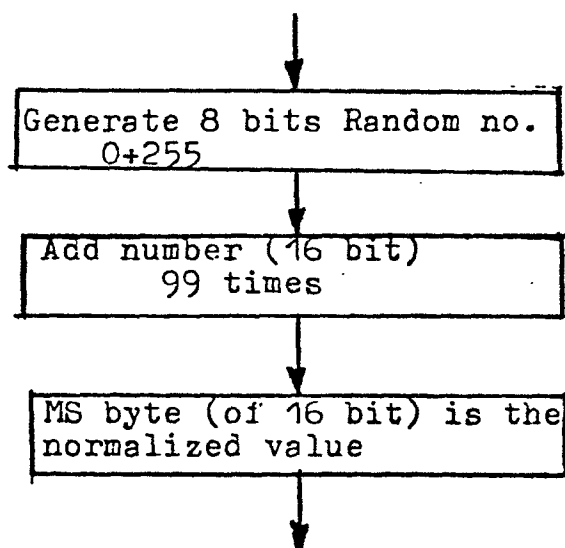
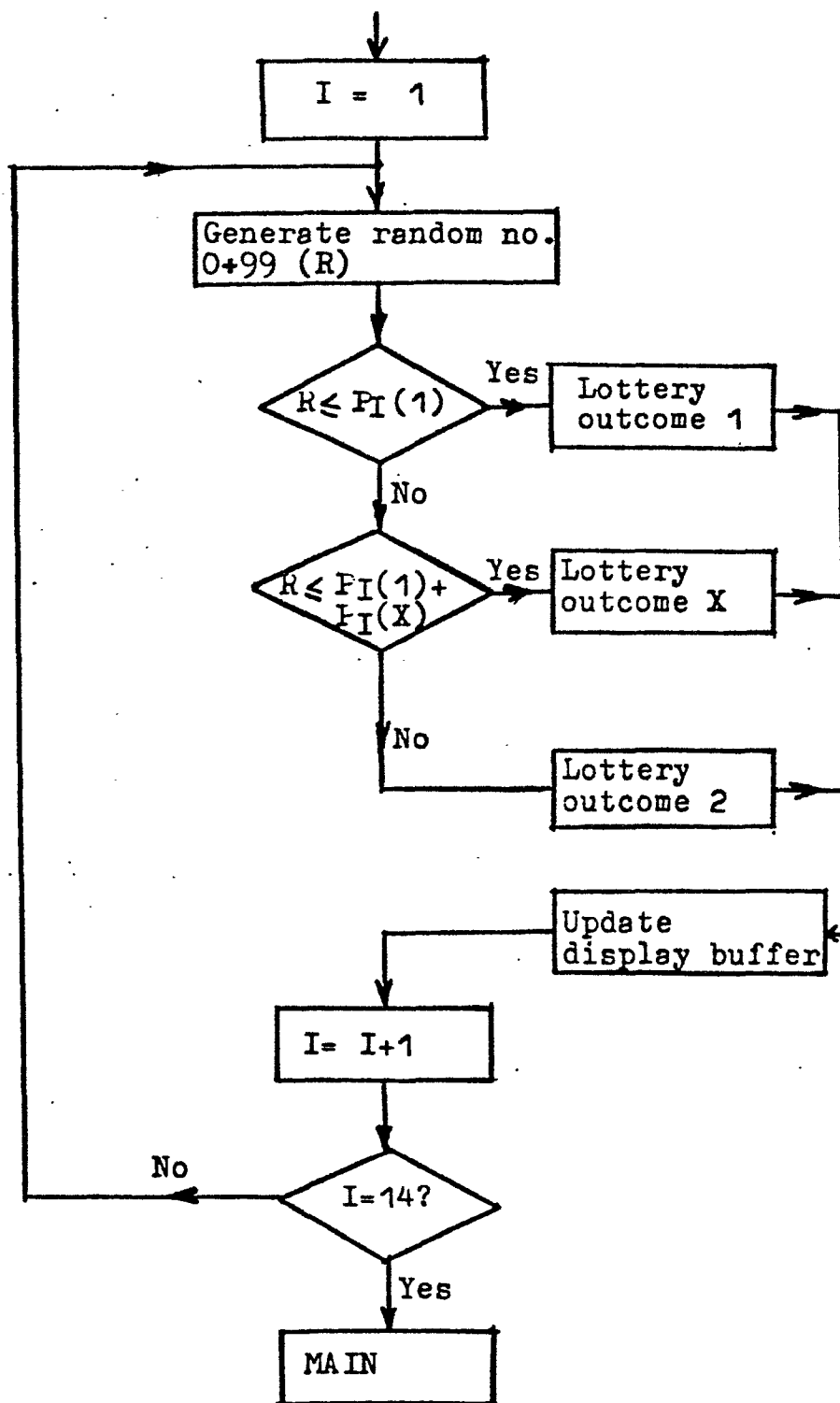


Fig-12

LUCK MODE FLOW PROGRAM



Double from BEST

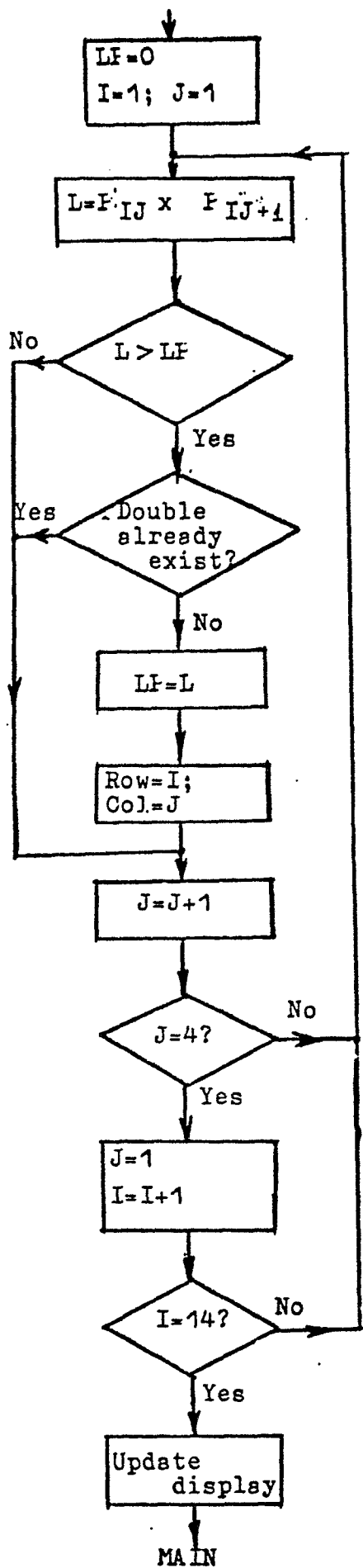


Fig-14.

Triple from BEST

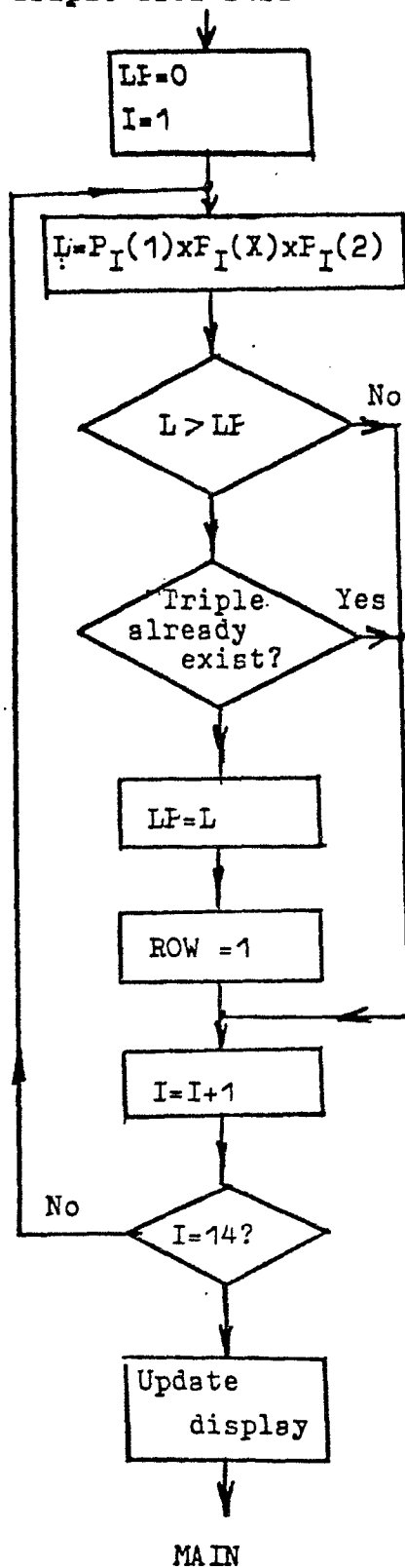


Fig-15

Double from LUCK

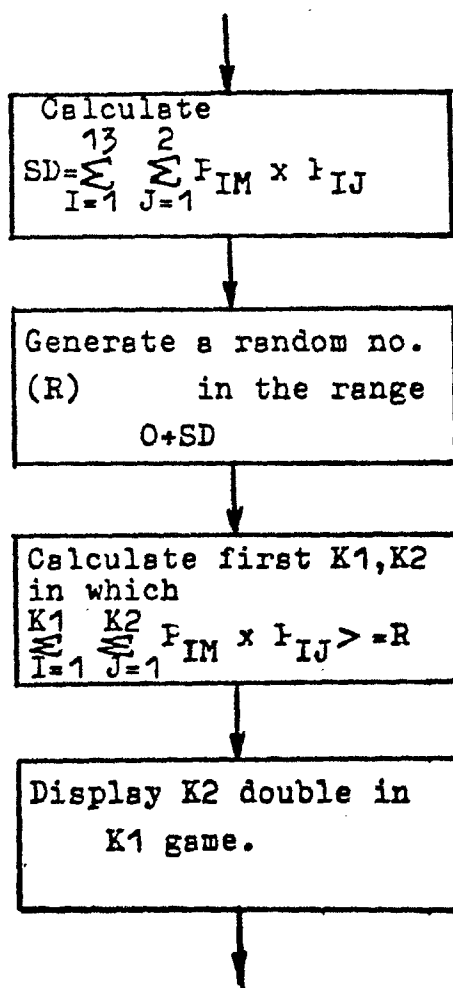


Fig-16

Triple from LUCK

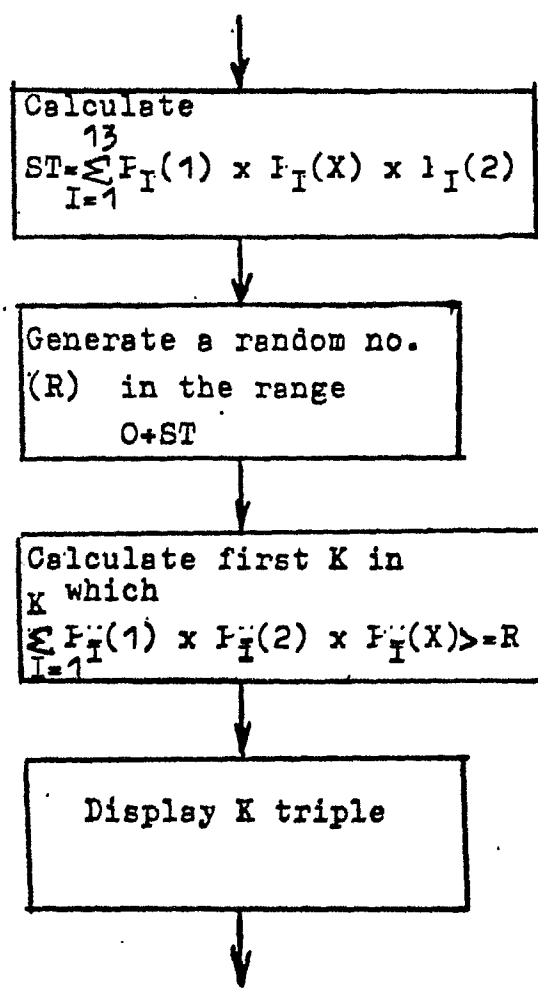
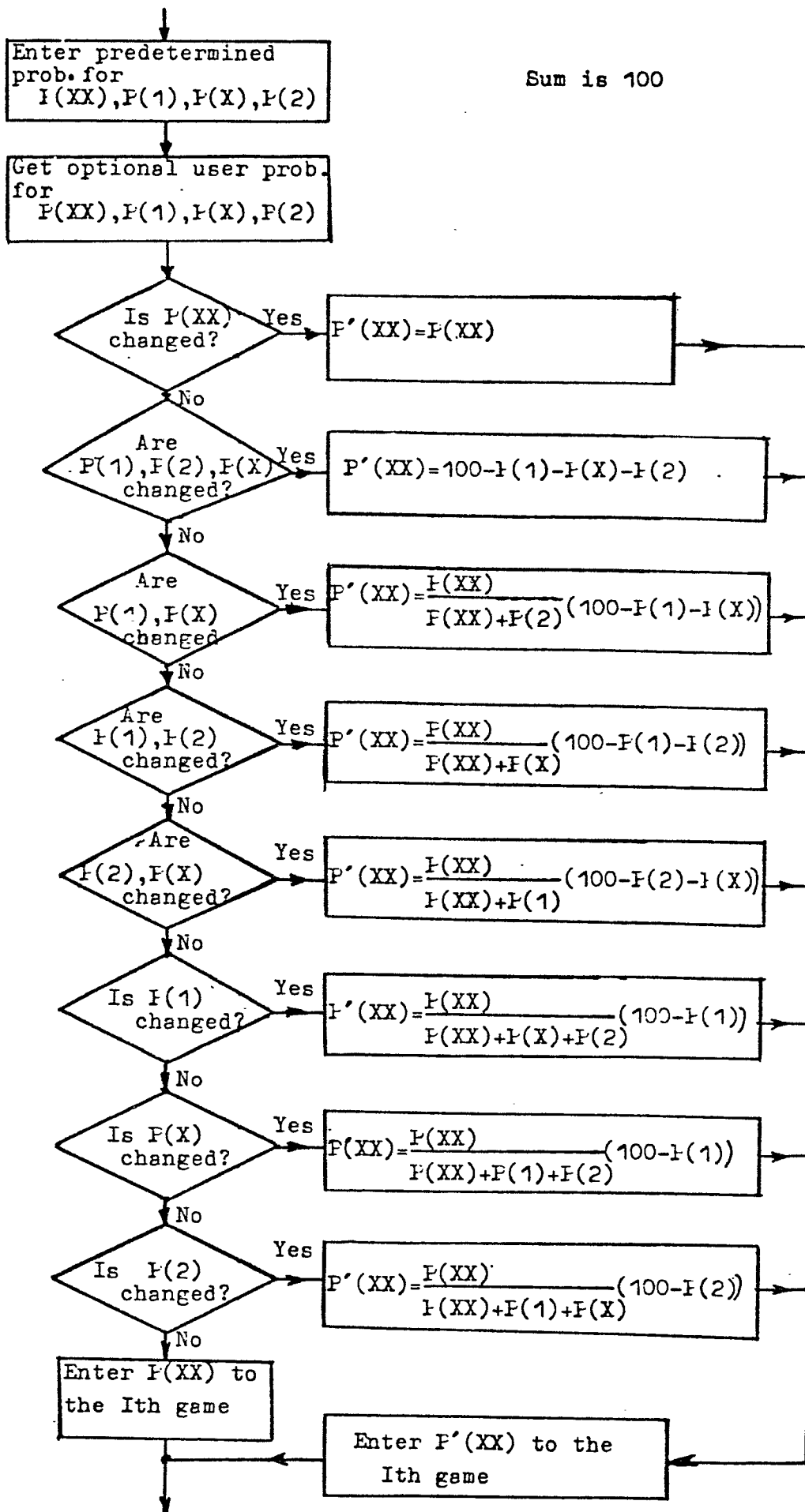


Fig-17



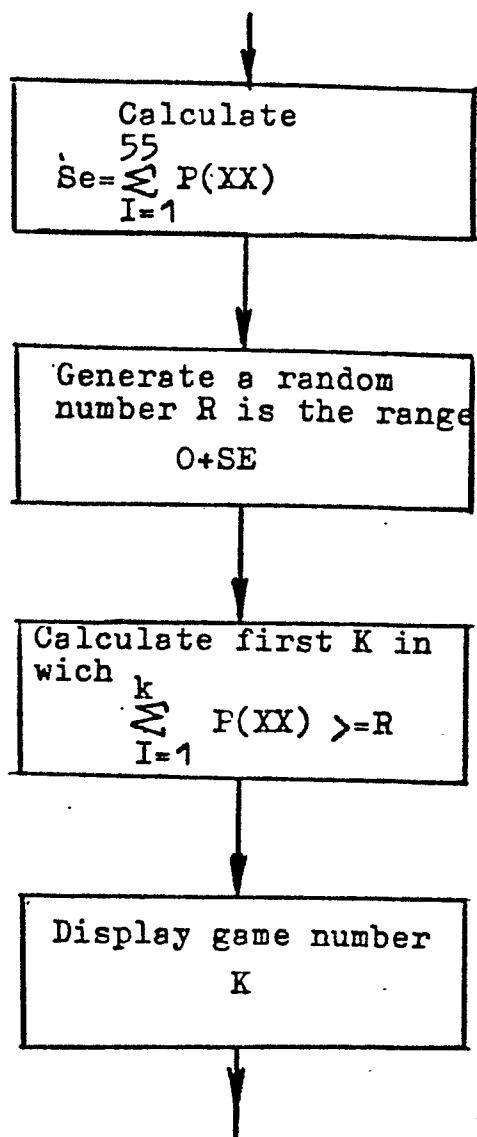


Fig--19--

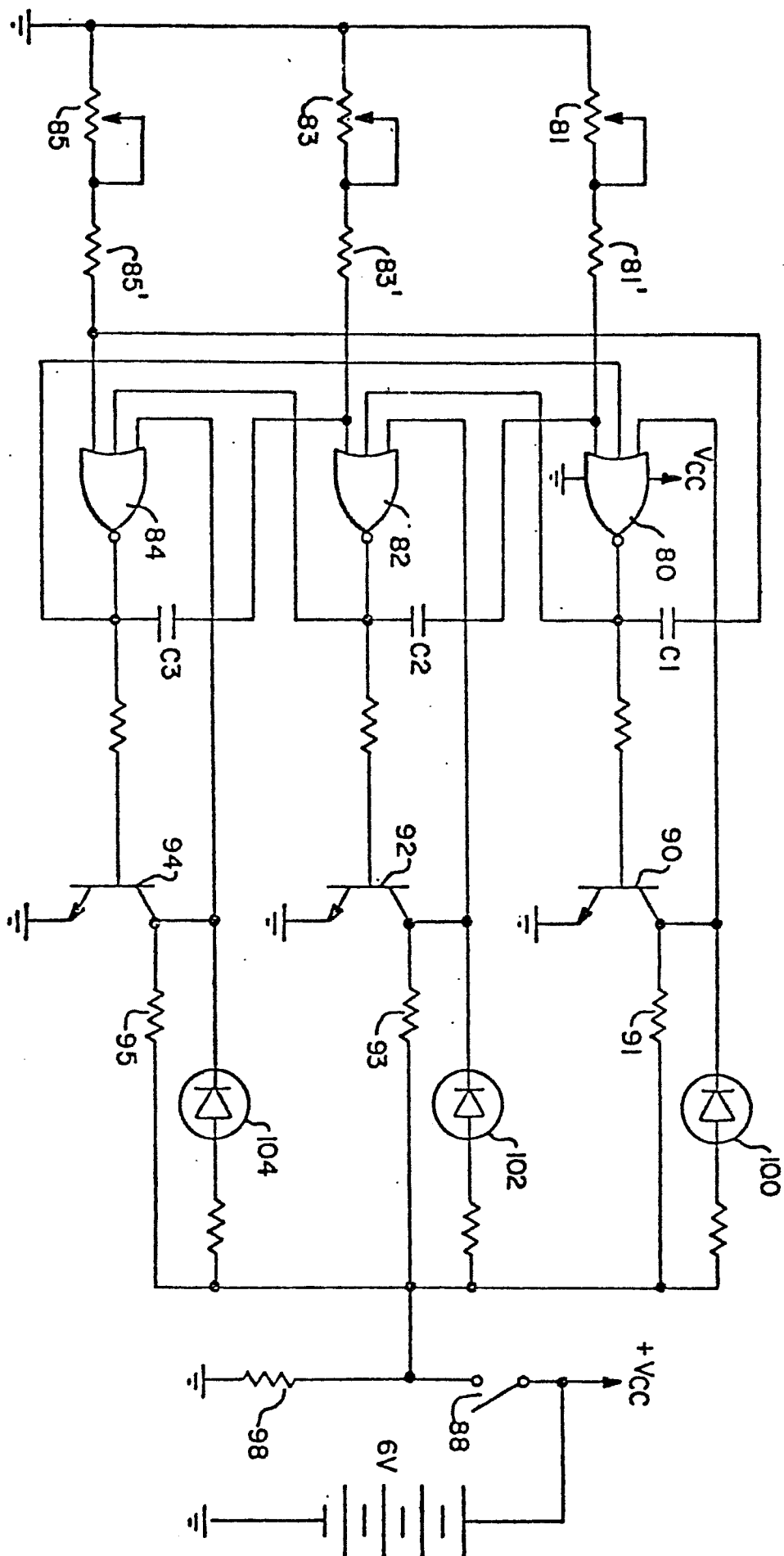


FIG.—20