EXHIBIT 15

United States Patent [19]

Twardowski

[11] **Patent Number: 4,535,333**

[45] **Date of Patent: Aug. 13, 1985**

[54] **TRANSMITTER AND RECEIVER FOR CONTROLLING REMOTE ELEMENTS**

[75] Inventor: **Joseph W. Twardowski**, Schaumburg, Ill.

[73] Assignee: **Chamberlain Manufacturing Corporation**, Elmhurst, Ill.

[21] Appl. No.: **422,452**

[22] Filed: **Sep. 23, 1982**

[51] **Int. Cl.**[3] **H04B 9/00**; H04Q 9/00

[52] **U.S. Cl.** **340/825.69**; 340/825.22; 340/825.31; 340/825.72; 455/151

[58] **Field of Search** 340/825.69, 825.22, 340/825.31, 696, 825.72; 455/151

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,263,536	4/1981	Lee et al.	318/266
4,291,411	9/1981	Muller et al.	455/151
4,328,540	5/1982	Matsuoka et al.	364/167

FOREIGN PATENT DOCUMENTS

1423126	1/1976	United Kingdom	340/825.72

Primary Examiner—Donald J. Yusko

[57] **ABSTRACT**

Transmitters and receivers for controlling remote elements which use a synchronous serial transmission format and which allows changes in coding to be automatically made between the receiver and transmitter and wherein the code is stored in memories of the transmitter and receiver and wherein the receiver can generate and transmit a new code with a light emitting diode so as to change the code in the transmitter. The transmitter and the receiver use micro-computers which are suitably programmed and include non-volatile memories.

12 Claims, 12 Drawing Figures

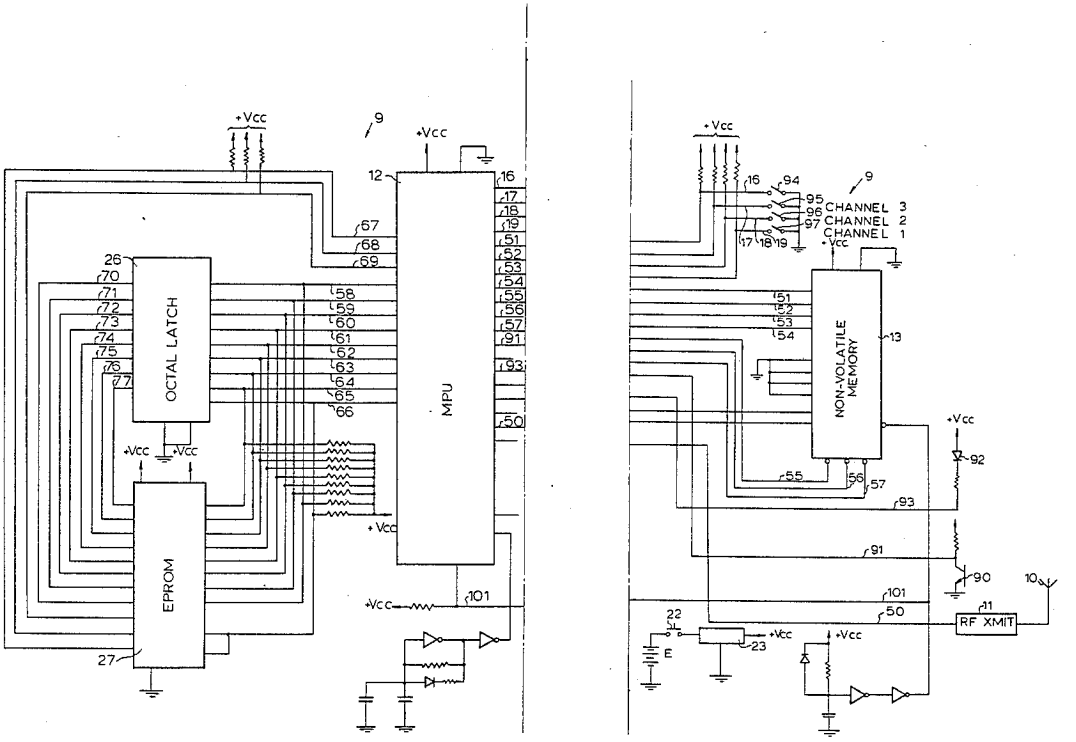


FIG. 1

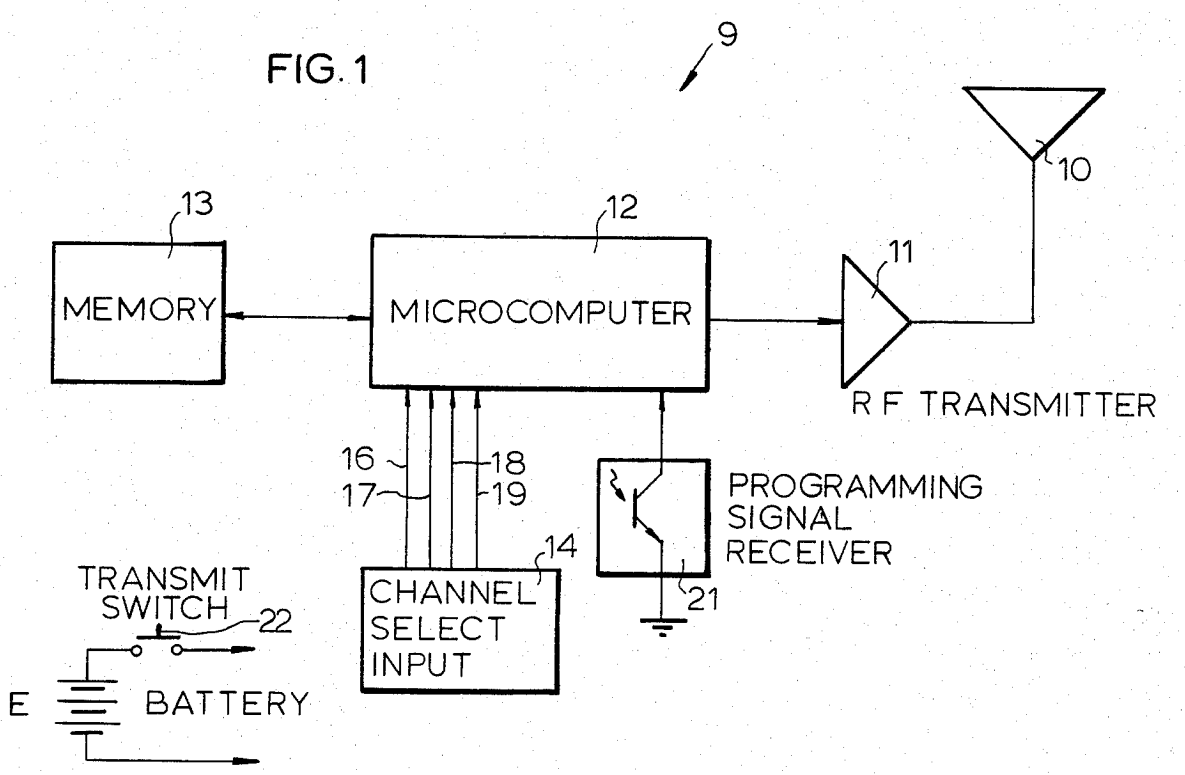

9
10
13
12
11
MEMORY
MICROCOMPUTER
R F TRANSMITTER
16
17
18
19
14
PROGRAMMING
SIGNAL
RECEIVER
21
TRANSMIT
SWITCH
22
CHANNEL
SELECT
INPUT
E
BATTERY

FIG. 3

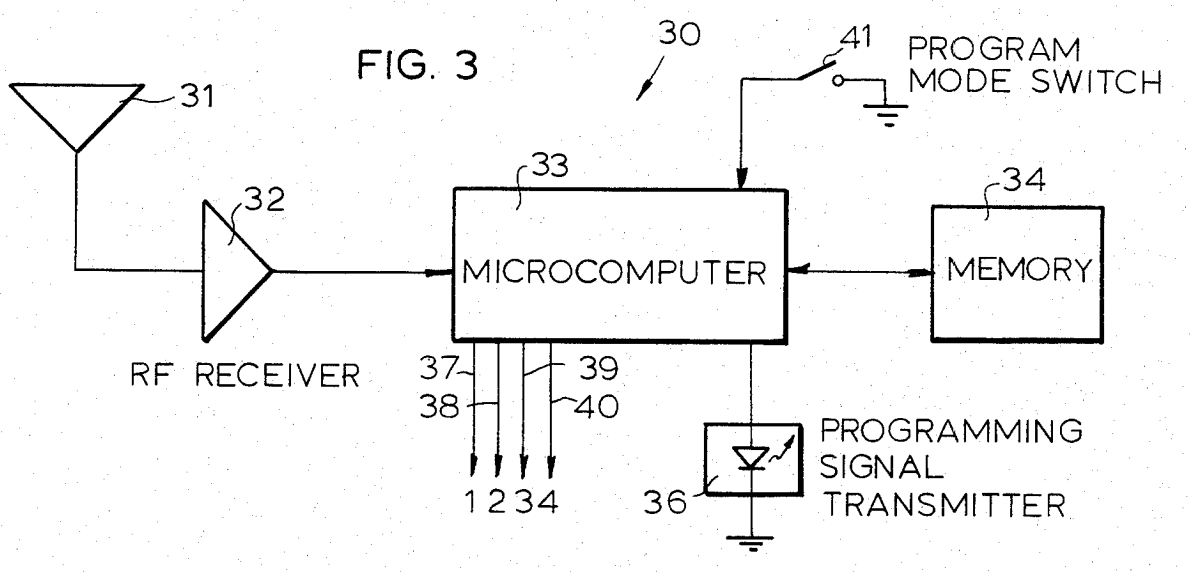

30
41
PROGRAM
MODE SWITCH
31
32
33
34
MICROCOMPUTER
MEMORY
RF RECEIVER
37
38
39
40
1 2 3 4
36
PROGRAMMING
SIGNAL
TRANSMITTER

FIG. 2

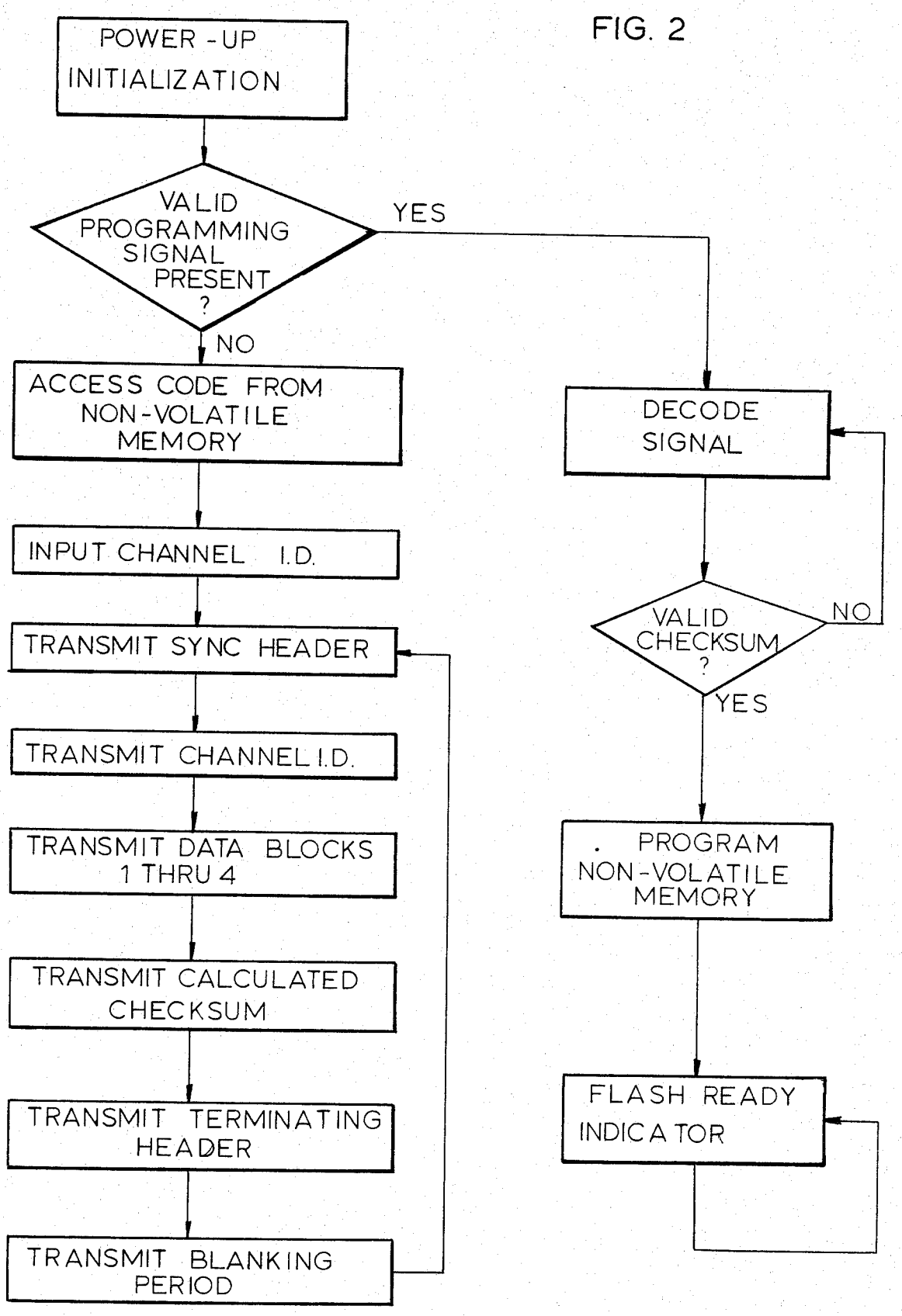

POWER - UP INITIALIZATION
VALID PROGRAMMING SIGNAL PRESENT ?
YES
NO
ACCESS CODE FROM NON-VOLATILE MEMORY
INPUT CHANNEL I.D.
TRANSMIT SYNC HEADER
TRANSMIT CHANNEL I.D.
TRANSMIT DATA BLOCKS 1 THRU 4
TRANSMIT CALCULATED CHECKSUM
TRANSMIT TERMINATING HEADER
TRANSMIT BLANKING PERIOD
DECODE SIGNAL
VALID CHECKSUM ?
NO
YES
PROGRAM NON-VOLATILE MEMORY
FLASH READY INDICATOR

FIG. 4

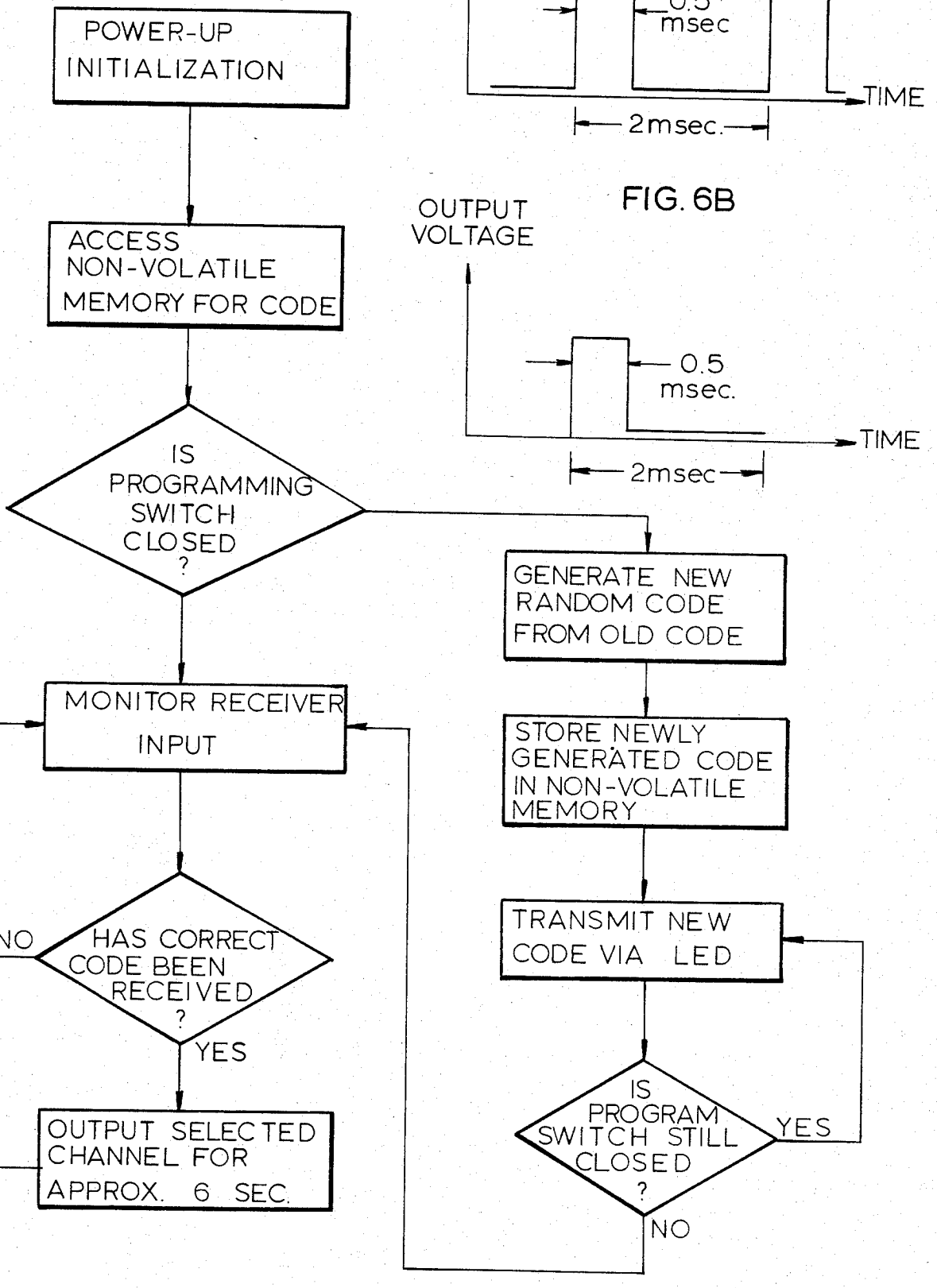


FIG. 6A

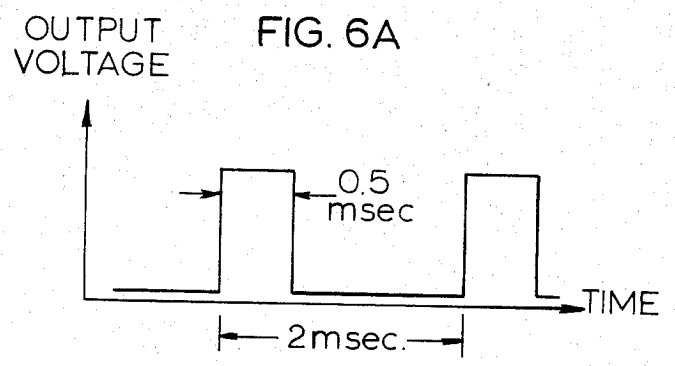


FIG. 6B

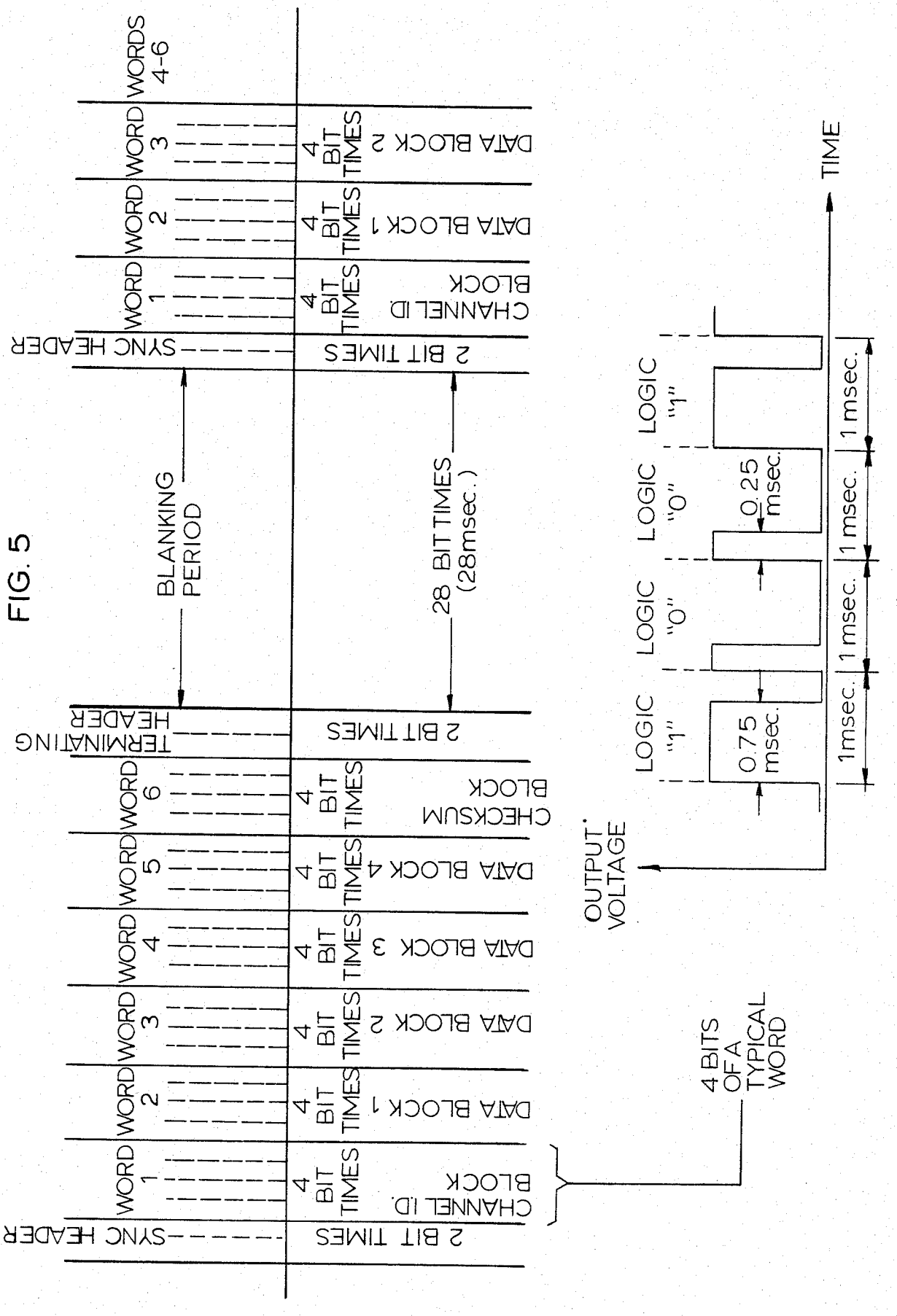

SYNC HEADER
2 BIT TIMES
CHANNEL I.D. BLOCK
4 BIT TIMES
WORD 1
DATA BLOCK 1
WORD 2
DATA BLOCK 2
WORD 3
DATA BLOCK 3
WORD 4
DATA BLOCK 4
WORD 5
CHECKSUM BLOCK
WORD 6
TERMINATING HEADER
BLANKING PERIOD
28 BIT TIMES (28msec.)
WORDS 4-6
4 BITS OF A TYPICAL WORD
OUTPUT VOLTAGE
LOGIC "1"
LOGIC "0"
LOGIC "0"
LOGIC "1"
0.75 msec.
0.25 msec.
1msec.
1 msec.
TIME

FIG. 5

FIG. 7A

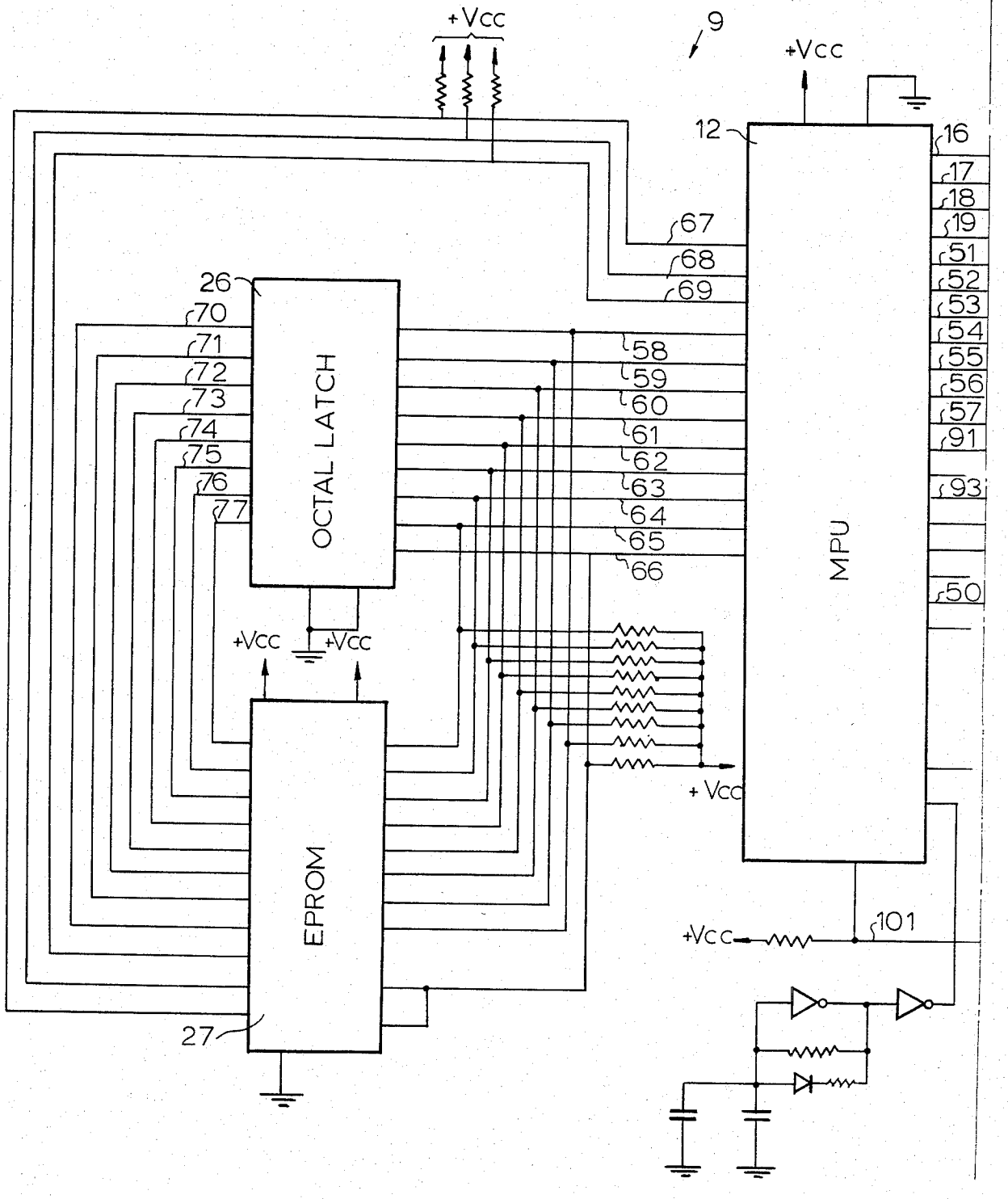

+Vcc
9
+Vcc
12
16
17
18
19
51
52
53
54
55
56
57
91
93
50
67
68
69
26
70
71
72
73
74
75
76
77
OCTAL LATCH
58
59
60
61
62
63
64
65
66
MPU
+Vcc
+Vcc
+Vcc
EPROM
27
+Vcc
101

FIG. 7B

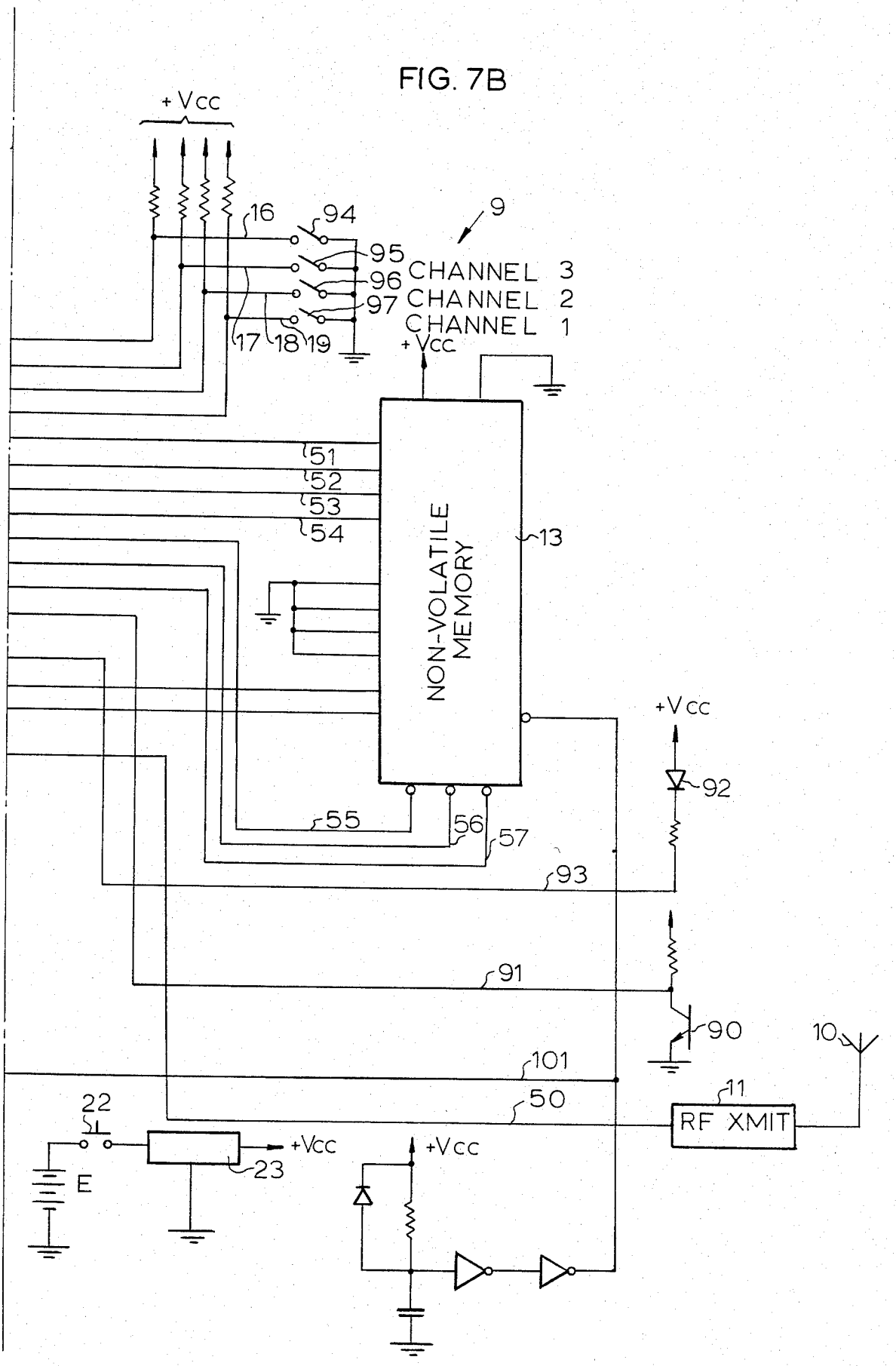

+Vcc
16
94
9
95
96
97
CHANNEL 3
CHANNEL 2
CHANNEL 1
17
18
19
+Vcc
51
52
53
54
NON-VOLATILE MEMORY
13
+Vcc
92
55
56
57
93
91
90
10
101
50
11
RF XMIT
22
+Vcc
23
E
+Vcc

FIG. 8A

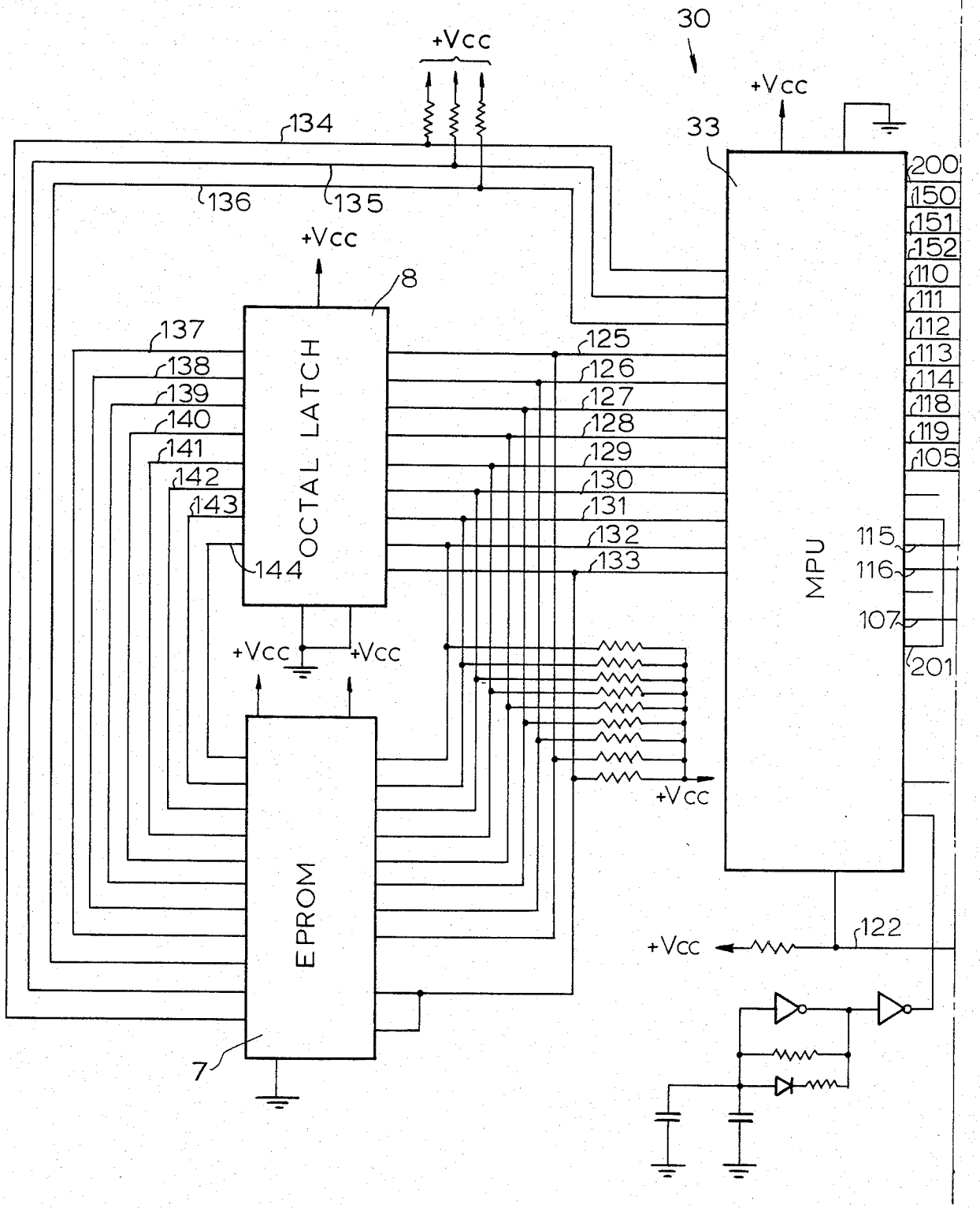

30
+Vcc
134
136
135
33
+Vcc
200
150
151
152
110
111
112
113
114
118
119
105
8
137
138
139
140
141
142
143
OCTAL LATCH
144
125
126
127
128
129
130
131
132
133
MPU
115
116
107
201
+Vcc
+Vcc
+Vcc
EPROM
7
+Vcc
122

FIG. 8B

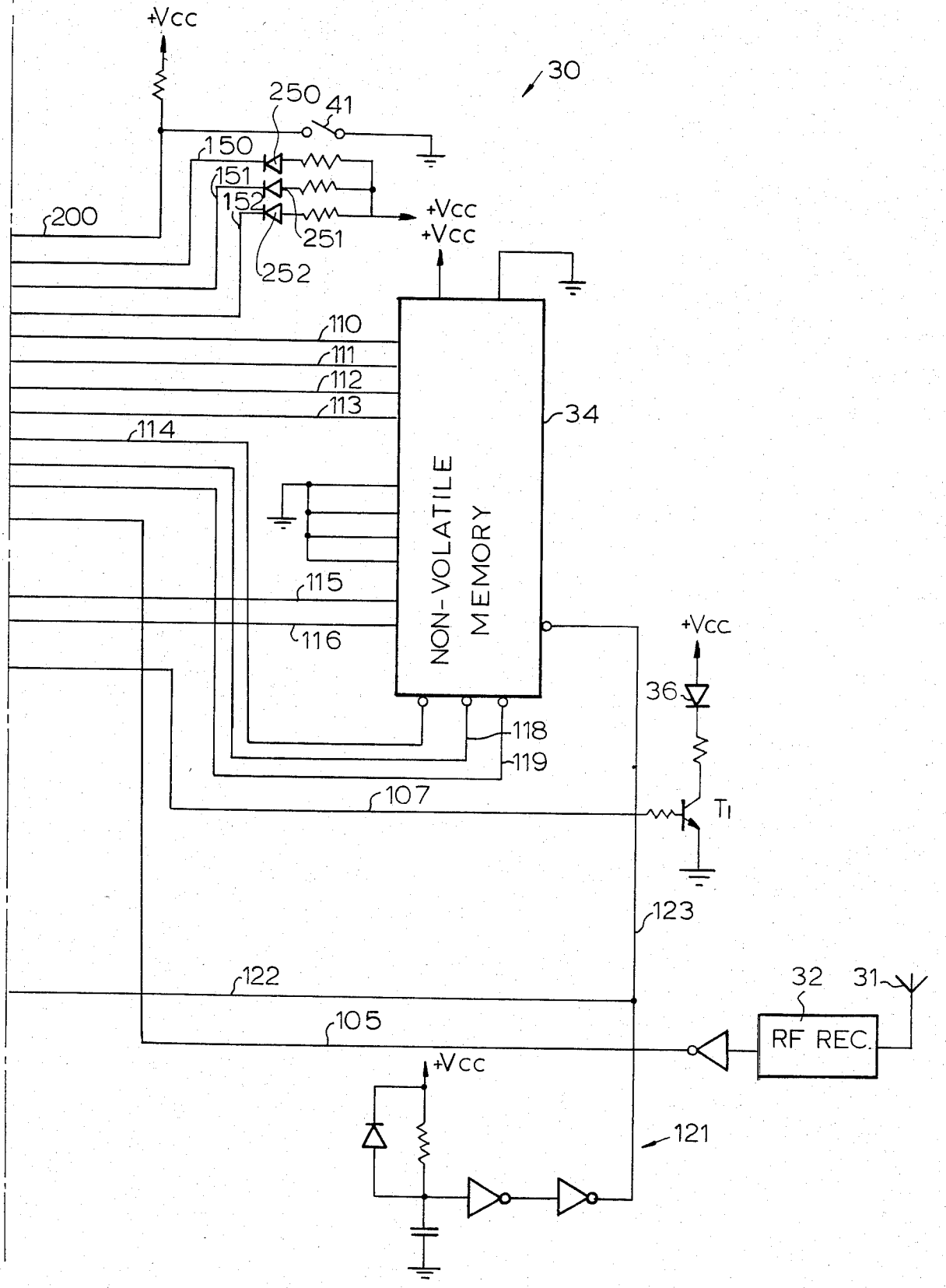
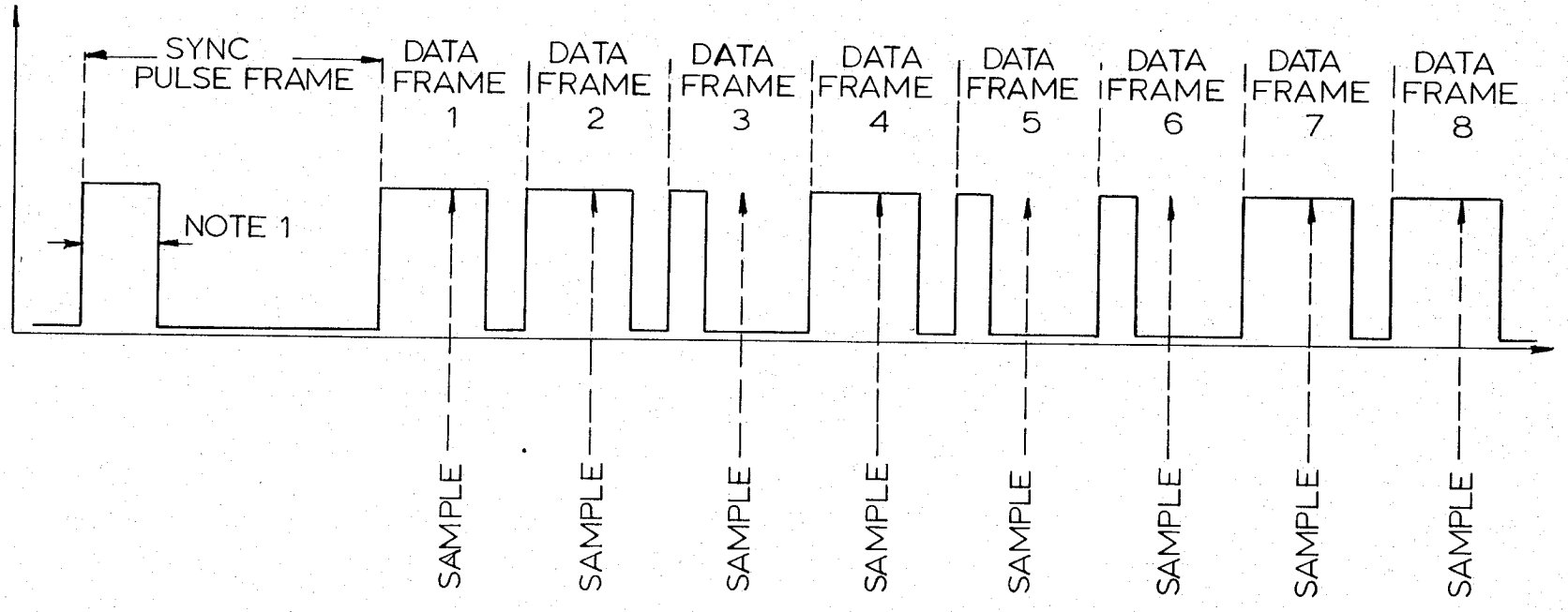

+Vcc
30
250
41
150
151
152
200
+Vcc
+Vcc
251
252
110
111
112
113
34
114
NON-VOLATILE MEMORY
115
116
+Vcc
36
118
119
107
T1
123
122
32
31
105
RF REC.
+Vcc
121

FIG. 9



NOTE 1. Pulse width of sync pulse determines sampling time interval starting from the leading edge of each pulse.

TRANSMITTER AND RECEIVER FOR CONTROLLING REMOTE ELEMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to co-pending application of Joseph W. Twardowski and F. J. Liotine entitled "Method and Apparatus For Controlling the Coding In A Transmitter and Receiver".

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates in general to control transmitters and receivers and in particular to a novel control system.

2. Description of the Prior Art

Remote control transmitters and receivers are known as, for example, for garage door openers and other devices. Initially, a different carrier frequency was utilized for each pair of transmitters and receivers so as to isolate them from other units. Also, various coding schemes have been utilized to encode data into binary form. Certain of such transmitters and receivers include a plurality of two position switches which control the coding for the transmitter and receiver and in such systems the codes can be changed by manually changing the positions of the switches to different positions to assure that the position of the switches in the transmitter and receiver are the same.

SUMMARY OF THE INVENTION

The present invention comprises a novel multi-channel transmitter and receiver for controlling a plurality of functions and includes the feature of changing the code in the receiver and transmitter to one of a large number of codes in an automatic manner. A pulse length binary code is utilized.

When it is desired to change the identification code, a program mode switch is closed in the receiver and the micro-computer recalls from the non-volatile memory the last stored code. Using this code as a start, it performs a random number generation algorithm and stores the newly generated code in the non-volatile memory and immediately transmits the new code through a light emitting diode. The transmission format with the light emitting diode at the receiver continues until the program mode switch is turned off. During the energization of the light emitting diode in the receiver, the transmitter is placed in close proximity to the receiver so that it detects the code from the light emitting diode and the new code is then stored in the memory of the transmitter which then produces a flashing ready signal to indicate to the operator that the programming cycle has been completed.

It is seen that the present invention provides an improved remote control system that can be used for a number of channels and allows for automatic change of the address coding between the transmitter and receiver.

Another object of the invention is to provide transmitters and receivers which have a large number of possible codes so as to eliminate interference between closely spaced transmitters and receiver systems.

Yet another object of the invention is to provide an improved transmitter and receiver system for a remote control device.

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

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** comprises a block diagram of the transmitter;

FIG. **2** comprises a flow chart for the transmitter;

FIG. **3** comprises a block diagram for the receiver;

FIG. **4** comprises a flow chart for the receiver;

FIG. **5** illustrates a transmission signal format;

FIG. **6A** illustrates a sync header waveform;

FIG. **6B** illustrates a terminating header waveform;

FIGS. **7A** and **7B** comprise a schematic diagram of the transmitter;

FIGS. **8A** and **8B** comprise a schematic diagram of the receiver; and

FIG. **9** illustrates a typical pulse train.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. **1** illustrates in block form the transmitter of the invention which comprises an antenna **10**, an RF transmitter section **11** connected to the antenna and a micro-computer **12** supplying an input signal to the RF transmitter **11**. The micro-computer is connected to a memory **13** which may be a non-volatile type memory and a number of channel select inputs **16**, **17**, **18** and **19** are connected to a channel selector unit **14** and supply inputs to the micro-computer **12**. A power supply comprises a battery E and a transmit switch **22** such that when the transmit switch **22** is closed the transmitter is energized by applying power to the various units of the transmitter. A programming signal receiver **21** is connected to the micro-computer and provides means for selecting the code in the transmitter.

FIG. **2** comprises the transmitter flow chart and when power is turned on the micro-computer **12** determines whether a valid programming signal is present.

FIG. **3** is a block diagram of the receiver **30** which comprises an antenna **31** for receiving radiation from the transmitter **9**. The receiver **30** includes an RF section **32** which is connected to the output of the antenna **31** and the RF receiver section **32** supplies an input to a micro-computer **33**. A memory **34** such as a non-volatile type is connected to the micro-computer **33**. A program mode switch **41** is connected to the micro-computer and output channel leads **37**, **38**, **39** and **40** supply operating signals for various apparatus or functions which are to be controlled as, for example, channel 1 might comprise a garage door opener. Channel 2 might comprise a security control channel. A programming signal transmitter **36** is connected to the micro-computer **33** for programming the transmitter **9**.

FIG. **4** comprises a flow chart for the receiver.

The transmitter and receiver of the invention eliminate the dip switches for code selection which are required in prior art devices and allows the expansion of channels so that a number of channels can be utilized to control different functions. Faster response times are obtained than prior art control transmitters and receivers. A specific embodiment of the invention was constructed wherein a four-bit single chip micro-computer was utilized rather than custom discrete logic integrated circuit for performing the encoding and decoding of the

algorithm. In addition, a non-volatile memory is used rather than a multiple three position switch for storing the custom code for each transmitter and receiver system.

The use of a single chip micro-computer rather than a discrete logic integrated circuit allows system flexibility for additional expansion and for various other radio controlled applications in addition to garage door opener systems without the requirement of major and exhaustive redesign efforts or custom integrated circuits. For such subsequent changes, a simple micro-program change in the self-contained mask ROM is all that is required and thus only software changes are necessary.

By using non-volatile memories rather than the dip switches used in the transmitters and receivers of the prior art devices requires that the randomly selected code be supplied from the receiver to the transmitter. Because of Federal Communication Commission rules and regulations, the transmission of radio frequency signals for this purpose cannot be used since the transmission of a coding signal for defining the code in the transmitter would not be within the Rules for actuating a garage door opener. This would comprise the transmission of a message containing information. This means that (1) during the programming mode transfer of code information from the receiver to the transmitter, the transmitter and receiver would have to be hard wired together or (2) the transfer of such data occurs by using infrared transmitters and receivers. The use of infrared transmitting and receiving means requires no physical contact between the systems.

In the present invention a synchronous serial transmission data format is utilized because (1) the equivalent replacement of the prior art nine pole three-position switch with a non-volatile memory requires that the electrical inputs be binary and (2) the present design allows additional channel expansion and identification.

In a particular embodiment constructed according to the invention, the maximum number of channels was selected to be sixteen and allow 2^{16} possible code combinations or 65,536.

The transmission format used in the invention utilizes security and privacy and is binary and uses pulse position modulation as the decoding format for data transmission. FIGS. **5** and **6A** and **6B** illustrate the data format used. As shown in FIG. **5**, a synchronization header frame of two bits is used for synchronization at the receiver. The first word **1** is a channel identification block of four-bits in length which contains the binary coded information that identifies the transmitting channel and this selection limits the maximum number of channels to sixteen.

Words **2** through **5** are data blocks and comprise four words each of four-bits containing binary coded information that can represent the code for a particular channel (2^{16} possible code combinations or 65,536). Alternatively, other forms of digital information as, for example, the output of a transducer can be included in these words.

Word **6** is a checksum block and is an error checking format which is derived by the binary addition of the identification block with data blocks **1** through **4** and eliminates any carry bits. For example:

BLOCK	MSB Bit 4	Bit 3	Bit 2	LSB Bit 1
Channel Identification Block	0	1	1	0
Data Block 1	1	1	0	1
Data Block 2	1	0	0	1
Data Block 3	1	1	1	0
Data Block 4	1	1	0	1
Checksum Block = binary sum of all blocks less any carry bits	0	1	1	1

Then a termination header which is two-bits in length indicates to the receiver that the current information transmission train has terminated. Then there is a blanking period of 28 bits which in a specific embodiment comprises 28 msec and then the data format is repeated again.

An example of word **1** is shown in exploded form in FIG. **5** comprising four-bits of a typical word and a logic 1 comprises a pulse of 0.75 msec and a 0.25 period of no signal. A logic 0 comprises a signal of 0.25 and then no signal for 0.75 msec.

FIG. **3** illustrates the receiver block diagram and the software flow chart for the receiver is illustrated in FIG. **4**. When the power is turned on, the receiver software first turns on the complete hardware system. It first interrogates the program mode switch input. If the program mode switch **41** is closed, the micro-computer **33** proceeds to access the non-volatile memory **34** to recall the last stored code. Using this code as a start, it then performs a random number generation algorithm and stores the newly generated code in the non-volatile memory and immediately transmits this new code through the light emitting diode **36**. The transmitter **9** is placed in close proximity to the receiver **30** such that the programming signal receiver **21** receives the information from the light emitting diode **36**. The transmission signal format of the receiver is as shown in FIG. **5** except that it does not need the channel identification block and uses a shorter blanking time equal to 5 msec. The receiver continues to transmit the code until the program mode switch **41** is opened after which the receiver monitors the receiver input port from the RF section and antenna.

The receiver algorithm contains a software phase lock loop to lock it on the receiver sync header. All timing information required to perform the remainder of the algorithm is contained in the pulse width of the sync pulse. A software timing loop times out the pulse and stores this value in the memory. For each consecutive negative to positive transition, the micro-computer samples the input at the time interval it calculated from the sync pulse, as illustrated in FIG. **9**. After all of the bits are sampled and stored in the memory, a comparison is made with the code stored in the non-volatile memory for a valid match. If a match is found, the appropriate channel output is identified by an appropriate light emitting diode to identify that particular channel.

FIG. **1** comprises a block diagram of the transmitter and FIG. **2** illustrates the software flow chart of the transmitter. The transmitter upon power up interrogates the input photo-transistor **21** for a period of about 10 msec for indication of a valid programming signal. If no programming signal is available within the first ten milliseconds, the transmitter software assumes that the

presently stored code is accurate and the transmitter proceeds to transmit such code. It accesses the stored code from the non-volatile memory, reads the channel identification number, computes the checksum and then transmits all the information using the format illustrated and described.

If a programming signal is received, the transmitter decodes the incoming information and if the checksum is correct stores the new code in its non-volatile memory **13** and outputs a flashing ready signal to indicate that the programming cycle has been completed.

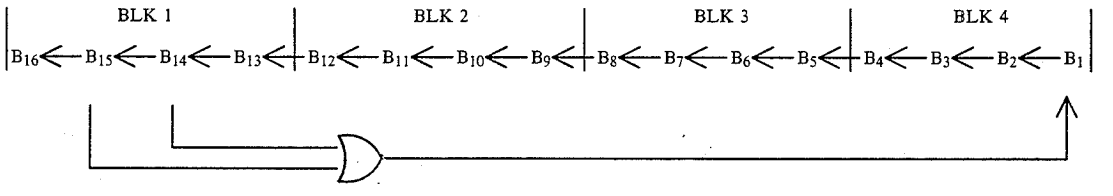
All output transmission timing is based on an ideal instruction execution time of 20 msec. Since the software is fixed, the only parameters that affect output timing are the resistor capacitor tolerances and any input tolerance variations between different micro-computers.

A software pseudorandom number generator is utilized at the receiver to generate the different codes.

The use of software to generate random values results in a paradox. The fact that an algorithm exists for a process implies that the process outputs are not truly random because the algorithm can be used to predict the output sequence. True random values can only be generated by the use of systems such as "memory garbage" or "human reaction time". The use of human reaction time requires additional hardware and expense which is undesirable in the high volume electronic industry. In the present invention, the use of "memory garbage" to start the system "initiation" or starting value is used on a one time basis.

In the algorithm used every time a random number is required a new sixteen bit configuration will result from the seed or initiation value used. Continuous recall for sufficient number of times will result in all the possible sixteen bit configurations. However, the outputs will appear random if the sequence of outputs are considered and it is impossible to prove that the program is not producing true random numbers. The distribution of outputs is uniform over the range of possible outputs although all possible sixteen bit values appear before any repetition occurs. In the present invention 65,536 outputs will occur before any repetition occurs.

The algorithm used works as follows. The random code is stored in four blocks of memory each four-bits wide for a sixteen bit word. This allows a binary representation of 65,536 discrete numbers. However, for the random number generator algorithm to work, the all zero state must not be used therefore there are only 65,535 numbers that can be used.



Whenever the program calls for random number, the previous value or "seed" is recalled. Each bit is shifted left one position. Bits **14** and **15** are exclusive or-ed and the result is shifted into the first position of block **4**. In this manner, all possible 65,535 combinations will result before the pattern repeats.

The program for the transmitter micro-processor **12** and the program for the receiver micro-processor **33** are attached.

FIGS. 7A and 7B illustrate the electrical schematic of the transmitter **9**, the antenna **10** is connected to the RF transmitter **11** which receives an output on lead **50** from output terminal SO of the micro-computer **12**. The micro-computer **12** may be a National type 404LP, for example. The non-volatile memory **13** may be a XICOR type X-2210 and is connected by leads **51** through **57** to the micro-processor **12** as illustrated. An octal latch **26** is connected to the micro-computer **12** by leads **58** through **66** and might be a type 74C373. A EPROM **27** might be a type 2716 available from INTEL and is connected by leads **58** through **69** to the micro-computer **12** and is further connected to the octal latch **26** by leads **70** through **77**. The power supply E and transmit switch **22** are connected to a regulator **23** which produces the drive voltage +Vcc. Infrared sensor **90** is connected by lead **91** to the micro-computer **12**. A ready indicator **92** is connected by lead **93** to the micro-computer **12**. Channel selector switches **94** through **97** are connected to channel selector leads **16, 17, 18** and **19** which are connected to the micro-computer **12**. A lead **101** is connected from the memory **13** to the reset terminal of the micro-computer **12**.

FIG. **8** illustrates the receiver in schematic form. The micro-computer **33** may be a type 404LP available from National Corporation. The antenna **31** is connected to the RF receiver **32** and by lead **105** to the micro-computer **33**. The programming LED **36** is connected through a resistor and a transistor T1 to lead **107** which is connected to the micro-computer **33**. A non-volatile memory **34** which might be a type X2210 available from XICOR is connected by leads **110** through **119** to the micro-computer **33**. A reset circuit **121** is connected by leads **122** and **123** to the reset of the micro-computer **33** and the memory **34**. An octal latch **8** which might be type 74C373 is connected by leads **125** through **133** to the micro-computer **33**. An EPROM **7** which may be a type 2715 is connected to the octal latch **8** and to the computer **33** by leads **125** through **136**. The EPROM **7** and octal latch **8** are connected together by leads **137** through **144**. The program switch **41** is connected to the micro-computer **33** by lead **200**. The channel indicator lights **250, 251** and **252** are connected to the micro-computer by leads **150, 151** and **152** and illustrate which channel is energized.

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

APPENDIX

TRANSMITTER SOFTWARE LISTING

```
1   001F        CKSUM = 1,15        ;CHECKSUM
2   001E        BLK4 = 1,14         ;INFORMATION BLOCK 4
3   001D        BLK3 = 1,13         ;INFORMATION BLOCK 3
4   001C        BLK2 = 1,12         ;INFORMATION BLOCK 2
5   001B        BLK1 = 1,11         ;INFORMATION BLOCK 1
6   001A        IDNUM = 1,10        ;I. D  NUMBER
8   002E        COUNT0 = 2,14       ;GP COUNTER 0
9   002D        COUNT1 = 2,13       ;GP COUNTER 1
10  002C        COUNT2 = 2,12       ;GP COUNTER 2
11  003F        SCRATC = 3,15       ;SCRATCH PAD REGISTER
12  003E        BRPNT = 3,14        ;BR MEMORY POINTER
13  003D        BDPNT = 3,13        ;BD MEMORY POINTER
14  003C        COMP1 = 3,12        ;COMPARE REGISTER 1
15  003B        COMP2 = 3,11        ;COMPARE REGISTER 2
16  003A        CNTROL = 3,10       ;CONTROL WORD REGISTER
17  0030        NOUSE = 3,0         ;SCRATCH PAD REGISTER
18  000F        FLAG1 = 0,15        ;FLAG REGISTER 1
19          ;INITIALIZATION
20 000 00       CLRA
21 001 3E       LBI SCRATC
22 002 7F       STII 15
23 003 40       COMP
24 004 3E       LBI SCRATC
25 005 333A     OMG                 ;INITIALIZE G PORT
26 007 333C     CAMQ                ;INITIALIZE L PORT
27 009 3361     LEI 1               ;TRISTATE L PORT
28 00B 50       CAB                 ;SET SI AS BINARY COUNTER
29 00C 333E     OBD                 ;SET SO = 0
30                                  ;INITIALIZE D PORT
```

```
31          ;CHECK FOR VALID PROGRAMMING SIGNAL ON INPUT GO
32          ;FOR A PERIOD OF APPROX. 10 MSEC.
33 00E 2E   START:  LBI IRIN
34 00F 70           STII 0
35 010 2E           LBI IRIN
36 011 3301         SKGBZ 0                 ;IS INPUT A LOGIC 1 ?
37 013 D5           JP BR1                  ;YES
38 014 E5           JP BEGIN                ;NO, ACTIVE INPUT ,CHECK
39 015 00   BR1:    CLRA                    ;FOR SYNC PULSE
41 017 3301 BR3:    SKGBZ 0                 ;IS INPUT STILL A LOGIC 1 ?
42 019 D8           JP BR2                  ;YES
43 01A E5           JP BEGIN                ;NO, ACTIVE INPUT, CHECK
44 01B 51   BR2:    AISC 1                  ;FOR SYNC PULSE
45 01C D7           JP BR3
46 01D 06           X 0
47 01E 51           AISC 1
48 01F 06           X 0
49 020 05           LD 0
50 021 5A           AISC 10
51 022 D5           JP BR1
52 023 60D0         JMP STRTX               ;NO INDICATION OF A PROGRAMMING
53                                          ;SIGNAL WITHIN 10 MSEC.
54 025 3C   BEGIN:  LBI BDPNT               ;INITIALIZE RECEIVED DATA
55 026 7B           STII 11                 ;MEMORY POINTER
56 027 2D           LBI COUNT0              ;RESET COUNTER IRIN
57 028 70           STII 0
58 029 70           STII 0
59 02A 19           LBI IDNUM
61 02C 70           STII 0
62 02D 70           STII 0
63 02E 70           STII 0
64 02F 70           STII 0
65 030 70           STII 0
66 031 2D           LBI COUNT0
67 032 3301         SKGBZ 0                 ;IS INPUT A LOGIC 1 ?
```

```
68 034 F6          JP LOOK1              ;YES
69 035 E5          JP BEGIN              ;NO
70 036 00    LOOK1 CLRA
71 037 51          AISC 1
72 038 3301  BR5   SKGBZ 0               ;IS INPUT STILL A LOGIC 1 ?
73 03A FC          JP BR4                ;YES
74 03B E5          JP BEGIN              ;NO
75 03C 51    BR4   AISC 1
76 03D F8          JP BR5
77 03E 06          X 0
78 03F 51          AISC 1
79 040 06          X 0
80 041 05          LD 0
81 042 5D          AISC 13
82 043 6036        JMP LOOK1
83 045 2E          LBI IRIN              ;VALID 5 MSEC. BLANK TIME
84 046 00          CLRA
85 047 3301        SKGBZ 0               ;IS INPUT A LOGIC 0 ?
86 049 C7          JP .-2                ;NO
87 04A 51    BR6:  AISC 1                ;YES
88 04B 3301        SKGBZ 0               ;IS INPUT STILL A LOGIC 0 ?
89 04D CF          JP BR7                ;NO, RISING TRANSITION DETECTED
90 04E CA          JP BR6                ;YES
91 04F 06    BR7:  X 0                   ;STORE COUNT IN IRIN
93 051 5E          AISC 14               ;FOR SAMPLE VALUE
94 052 44          NOP
95 053 2C          LBI COUNT1            ;STORE SAMPLE VALUE
96 054 06          X 0                   ;IN COUNTER 1
97 055 2E          LBI IRIN              ;CHECK FOR OVERFLOW
98 056 00          CLRA
99 057 21          SKE
100 058 DB         JP REPEAT             ;NO OVERFLOW
101 059 600E       JMP START             ;OVERFLOW EXISTS
102 05B 3301 REPEAT SKGBZ 0              ;CHECK FOR FALLING TRANSITION
```

```
103 05D DB          JP .-2          ;INPUT STILL HIGH
104 05E 6974        JSR SAMPLE      ;FALLING TRANSITION DETECTED
105 060 3C          LBI BDPNT       ;POINT TO RECEIVE DATA POINTER
106 061 25          LD 2
107 062 50          CAB
108 063 20          SKC
109 064 6067        JMP BR8
110 066 4B          SMB 3
111 067 3301 BR8:   SKGBZ 0         ;LOOK FOR FALLING TRANSITION
113 06A E7          JP BR8          ;INPUT STILL LOW
114 06B 3301 BR9:   SKGBZ 0
115 06D EB          JP .-2          ;INPUT STILL HIGH
116 06E 6974        JSR SAMPLE      ;FALLING TRANSITION DETECTED
117 070 3C          LBI BDPNT       ;POINT TO RECEIVE DATA POINTER
118 071 25          LD 2
119 072 50          CAB
120 073 20          SKC
121 074 F6          JP BR10
122 075 46          SMB 2
123 076 3301 BR10:  SKGBZ 0         ;LOOK FOR FALLING TRANSITION
124 078 FA          JP BR11         ;INPUT STILL HIGH
125 079 F6          JP BR10         ;INPUT STILL LOW
126 07A 3301 BR11:  SKGBZ 0
127 07C FA          JP  -2          ;INPUT STILL HIGH
128 07D 6974        JSR SAMPLE      ;FALLING TRANSITION DETECTED
129 07F 3C          LBI BDPNT       ;POINT TO RECEIVE DATA POINTER
130 080 25          LD 2
131 081 50          CAB
132 082 20          SKC
133 083 85          JP BR12
134 084 47          SMB 1
135 085 3301 BR12:  SKGBZ 0         ;LOOK FOR FALLING TRANSITION
136 087 89          JP BR13         ;INPUT STILL HIGH
137 088 85          JP BR12         ;INPUT STILL LOW
138 089 3301 BR13:  SKGBZ 0
```

```
139 08B 89          JP .-2              ;INPUT STILL HIGH
140 08C 6974        JSR SAMPLE          ;FALLING TRANSITION DETECTED
141 08E 3C          LBI BDPNT           ;POINT TO RECEIVE DATA POINTER
142 08F 25          LD 2
143 090 50          CAB
145 092 94          JP BR14
146 093 4D          SMB 0
147 094 3C    BR14. LBI BDPNT           ;INCREMENT BD POINTER
148 095 05          LD 0
149 096 51          AISC 1              ;LAST MEMORY LOCATION FILLED ?
150 097 99          JP BR15             ;NO
151 098 9F          JP BR16             ;YES
152 099 06    BR15  X 0
153 09A 3301  BR17: SKGBZ 0             ;LOOK FOR FALLING TRANSITION
154 09C 605B        JMP REPEAT          ;INPUT STILL HIGH
155 09E 9A          JP BR17             ;INPUT STILL LOW
156         ;CALCULATE RECEIVED DATA CHECKSUM
157 09F 00    BR16: CLRA
158 0A0 1A          LBI BLK1
159 0A1 31          ADD
160 0A2 1B          LBI BLK2
161 0A3 31          ADD
162 0A4 1C          LBI BLK3
163 0A5 31          ADD
165 0A7 31          ADD
166 0A8 1E          LBI CKSUM
167 0A9 21          SKE                 ;CHECKSUM CORRECT
168 0AA 600E        JMP START           ;CHECKSUM INCORRECT
169         ;PROGRAM NON-VOLATILE MEMORY
170 0AC 0E          LBI FLAG1           ;SET WRITE TO NVRAM FLAG
171 0AD 4C          RMB 0
172 0AE 39          LBI CNTROL          ;STORE CONTROL WORD
173 0AF 73          STII 3
174 0B0 6983        JSR NVRAM           ;STORE DATA IN NVRAM
175 0B2 3F          LBI NOUSE           ;INITIATE A STORE COMMAND
```

```
176 0B3 7D          STII 13             ;TO NVRAM
177 0B4 3F          LBI NOUSE
178 0B5 333A        OMG
179 0B7 3F          LBI NOUSE
180 0B8 7F          STII 15             ;TERMINATE STORE TO NVRAM
181 0B9 3F          LBI NOUSE
182 0BA 333A        OMG
183 0BC 3A    BR18: LBI COMP2           ;INITIALIZE COMPARE
184 0BD 7F          STII 15             ;REGISTERS
185 0BE 7F          STII 15
186 0BF 2C          LBI COUNT1          ;RESET COUNTER REGISTERS
187 0C0 70          STII 0
188 0C1 70          STII 0
189 0C2 0A          LBI 0,11            ;TURN ON READY INDICATOR
190 0C3 333E        OBD
191 0C5 69B2        JSR TIMER
192 0C7 2C          LBI COUNT1          ;RESET COUNTER REGISTERS
193 0C8 70          STII 0
194 0C9 70          STII 0
195 0CA 0E          LBI 0,15            ;TURN OFF INDICATOR
197 0CD 69B2        JSR TIMER
198 0CF BC          JP BR18
199           ;START TRANSMIT SEQUENCE
200 0D0 3387  STRTX: LBI 0,7            ;INSTITUTE AN ARRAY RECALL
201 0D2 333E        OBD                 ;CYCLE TO NVRAM
202 0D4 0E          LBI 0,15
203 0D5 333E        OBD
204 0D7 0E          LBI FLAG1           ;SET READ FROM NVRAM FLAG
205 0D8 4D          SMB 0
206 0D9 39          LBI CNTROL          ;STORE CONTROL WORD
207 0DA 77          STII 7
208 0DB 6983        JSR NVRAM
209 0DD 19          LBI IDNUM           ;READ CHANNEL ID NUMBER
210 0DE 332E        INL
211           ;CALCULATE CHECKSUM OF DATA BLOCKS AND ID NUMBER
```

```
212 0E0 00          CLRA
213 0E1 19          LBI IDNUM
214 0E2 31          ADD
215 0E3 1A          LBI BLK1
217 0E5 1B          LBI BLK2
218 0E6 31          ADD
219 0E7 1C          LBI BLK3
220 0E8 31          ADD
221 0E9 1D          LBI BLK4
222 0EA 31          ADD
223 0EB 1E          LBI CKSUM           ;STORE CHECKSUM IN CKSUM
224 0EC 06          X 0
225         ;TRANSMIT SYNC PULSE
226 0ED 3B  INITX   LBI COMP1           ;INITILIZE COMPARE REGISTER
227 0EE 78          STII 8
228 0EF 00          CLRA
229 0F0 3B          LBI COMP1
230 0F1 3369        LEI 9               ;SET ENG=1, SO=1
231 0F3 51  BR19:   AISC 1              ;TRANSMIT LOGIC 1
232 0F4 21          SKE                 ;FOR 500 USEC.
233 0F5 F3          JP BR19
234 0F6 3361        LEI 1               ;SET ENG=0, SO=0
235 0F8 3B          LBI COMP1           ;TRANSMIT LOGIC 0
236 0F9 7D          STII 13             ;FOR 1.5 MSEC.
237 0FA 00          CLRA
238 0FB 3B          LBI COMP1
239 0FC 51  BR20:   AISC 1
240 0FD 44          NOP
241 0FE 21          SKE

242 0FF 60FC        JMP BR20

243 101 44          NOP
244 102 44          NOP
245         ;TRANSMIT DATA BLOCK
```

```
246 103 3C          LBI BDPNT       ;INITIALIZE BD POINTER TO
248 105 3A          LBI COMP2       ;DATA TO BE TRANSMITTED
249 106 70          STII 0
250 107 3C    NXBYTE LBI BDPNT
251 108 25          LD 2
252 109 50          CAB
253 10A 3361        LEI 1           ;SET EN3=0, S0=0
254 10C 13          SKMBZ 3         ;CHECK LOGIC STATUS OF BIT 3
255 10D D0          JP BR21         ;RAM B3=1
256 10E 32          RC              ;RAM B3=0
257 10F D1          JP BR22
258 110 22    BR21: SC
259 111 69CB  BR22: JSR TXMIT
260 113 3E          LBI SCRATC      ;160 USEC. DELAY
261 114 7E          STII 14
262 115 3E          LBI SCRATC
263 116 06          X 0
264 117 51          AISC 1
265 118 D7          JP .-1
267 11A 3C          LBI BDPNT
268 11B 25          LD 2
269 11C 50          CAB
270 11D 3361        LEI 1           ;SET EN3=0, S0=0
271 11F 03          SKMBZ 2         ;CHECK LOGIC STATUS OF BIT 2
272 120 E3          JP BR23         ;RAM B2=1
273 121 32          RC              ;RAM B2=0
274 122 E4          JP BR24
275 123 22    BR23: SC
276 124 69CB  BR24: JSR TXMIT
277 126 3E          LBI SCRATC      ;160 USEC. DELAY
278 127 7E          STII 14
279 128 3E          LBI SCRATC
280 129 06          X 0
281 12A 51          AISC 1
282 12B EA          JP .-1
```

```
283 12C 44          NOP
284 12D 3C          LBI BDPNT
285 12E 25          LD 2
286 12F 50          CAB
287 130 3361        LEI 1               ;SET EN3=0,SO=0
288 132 11          SKMBZ 1             ;CHECK LOGIC STATUS OF BIT 1
289 133 F6          JP BR25             ;RAM B1=1
290 134 32          RC                  ;RAM B1=0
291 135 F7          JP BR26
292 136 22    BR25: SC
293 137 69CB  BR26: JSR TXMIT
294 139 3E          LBI SCRATC
295 13A 7E          STII 14
296 13C 3E          LBI SCRATC
297 13C 06          X 0                 ;160 USEC  DELAY
298 13D 51          AISC 1
301 140 3C          LBI BDPNT
302 141 25          LD 2
303 142 50          CAB
304 143 3361        LEI 1               ;SET EN3=0,SO=0
305 145 01          SKMBZ 0             ;CHECK LOGIC STATUS OF BIT 0
306 146 C9          JP BR27             ;RAM B0=1
307 147 32          RC                  ;RAM B0=0
308 148 CA          JP BR28
309 149 22    BR27: SC
310 14A 69CB  BR28: JSR TXMIT
311 14C 3C          LBI BDPNT           ;INCREMENT BD POINTER
312 14D 05          LD 0
313 14E 51          AISC 1
314 14F 44          NOP
315 150 06          X 0
316 151 05          LD 0
317 152 3A          LBI COMP2           ;LAST MEMORY LOCATION
319 154 6107        JMP NXBYTE          ;NO
320 156 44          NOP                 ;YES
```

```
321 157 3361          LEI 1
322 159 3B            LBI COMP1
323 15A 78            STII 8
324 15B 00            CLRA
325 15C 3B            LBI COMP1
326 15D 44            NOP
327 15E 44            NOP
328             ;TRANSMIT TERMINATOR
329 15F 3B            LBI COMP1
330 160 78            STII 8
331 161 00            CLRA
332 162 3B            LBI COMP1
333 163 3369          LEI 9                 ;SET EN3=1, S0=1
334 165 51      BR29  AISC 1                ;TRANSMIT LOGIC 1 FOR 500 USEC.
335 166 21            SKE
336 167 E5            JP BR29
337 168 3361          LEI 1                 ;SET EN3=0, S0=0
338             ;TRANSMIT BLANK TIME OF 28 MSEC.
339 16A 3A            LBI COMP2             ;INITIALIZE COMPARE REGISTERS
340 16B 72            STII 2
341 16C 76            STII 6
342 16D 2C            LBI COUNT1            ;RESET COUNTER REGISTERS
343 16E 70            STII 0
344 16F 70            STII 0
345 170 69B2          JSR TIMER
346 172 60ED          JMP INITX             ;RETRANSMIT DATA
347             ;SUBROUTINES
348             ;           SUBROUTINE SAMPLE
349 174 2C      SAMPLE: LBI COUNT1
350 175 00            CLRA                  ;SETUP SAMPLING VALUE
351 177 44            NOP
353 178 21            SKE
354 179 F6            JP SUB1
355 17A 3301          SKGBZ 0               ;READ INPUT
356 17C 6181          JMP SUB2
```

```
357 17E 22          SC                  ;INPUT IS A LOGIC 0

358 17F 6182        JMP SUB3

359 181 32    SUB2: RC                  ;INPUT IS A LOGIC 1
360 182 48    SUB3: RET
361           ;           SUBROUTINE NVRAM
362 183 3B    NVRAM: LBI COMP1          ;INITIALIZE BD MEMORY POINTER
363 184 7F          STII 15             ;AND COMPARE REGISTER
364 185 7B          STII 11
365 186 3C    SUB6: LBI BDPNT
366 187 25          LD 2                ;OUTPUT ADDRESS POINTED TO
367 188 51          AISC 1              ;BY COMPARE REGISTER
369 18A 333E        OBD
370 18C 39          LBI CNTROL          ;OUTPUT CONTROL WORD TO
371 18D 333A        OMG                 ;MEMORY
372 18F 0E          LBI FLAG1           ;CHECK FOR READ OR WRITE
373 190 01          SKMBZ 0               ;FLAG
374 191 DB          JP SUB4             ;READ FLAG
375 192 3C          LBI BDPNT           ;WRITE FLAG
376 193 25          LD 2
377 194 50          CAB
378 195 00          CLRA
379 196 333C        CAMQ                ;WRITE TO NVRAM
380 198 3365        LEI 5               ;ENABLE L DRIVERS
381 19A E1          JP SUB5
382 19B 3C    SUB4: LBI BDPNT
383 19C 25          LD 2
384 19D 50          CAB
385 19E 332E        INL                 ;READ FROM NVRAM
386 1A0 06          X 0
387 1A1 3F    SUB5: LBI NOUSE           ;INITIATE A DESELECT
388 1A2 7F          STII 15
389 1A3 3F          LBI NOUSE
390 1A4 333A        OMG
```

```
391 1A6 3361          LEI 1               ;DISABLE L DRIVERS
392 1A8 3C            LBI BDPNT
393 1A9 05            LD 0
394 1AA 51            AISC 1
395 1AB 06            X 0                 ;STORE NEW VALUE FOR
396 1AC 05            LD 0                ;MEMORY POINTER
397 1AD 3B            LBI COMP1           ;LAST MEMORY LOCATION
398 1AE 21            SKE                 ;ACCESSED?
399 1AF 6186          JMP SUB6            ;NO
400 1B1 48            RET                 ;YES
402 1B2 2D    TIMER   LBI COUNT0
403 1B3 00            CLRA
404 1B4 51            AISC 1
405 1B5 F4            JP -1               ;640 USEC. DELAY LOOP
406 1B6 05            LD 0                ;COUNT0 CONTENTS TO ACCU
407 1B7 51            AISC 1              ;ADD 1 TO ACCU
408 1B8 FD            JP LP1
409 1B9 07            XDS                 ;STORE ZERO IN COUNT0
410 1BA 05            LD 0                ;DECREMENT TO COUNT1
411 1BB 51            AISC 1              ;ADD 1 TO COUNT1
412 1BC 44            NOP
413 1BD 06    LP1     X 0
414 1BE 3A            LBI COMP2           ;COMPARE COUNTERS TO
415 1BF 05            LD 0                ;COMPARE REGISTERS

416 1C0 2C            LBI COUNT1
417 1C1 21            SKE                 ;SKIP IF MOST SIGN  COUNT
418 1C2 61B2          JMP TIMER           ;EQUALS COMP2
419 1C4 3B            LBI COMP1
420 1C5 05            LD 0
422 1C7 21            SKE                 ;EQUALS COUNT0
423 1C8 61B2          JMP TIMER           ;NO, INSTITUTE ANOTHER CYCLE
424 1CA 48            RET
425           ;               TRANSMIT SUBROUTINE
426 1CB 3B    TXMIT   LBI COMP1
```

```
427 1CC 73          STII 3
428 1CD 00          CLRA
429 1CE 3B          LBI COMP1
430 1CF 44          NOP
431 1D0 3369        LEI 9               ;SET EN3=1,SO=1
432 1D2 51    INC1  AISC 1              ;250 USEC. LOOP
433 1D3 21          SKE
434 1D4 D2          JP INC1
435 1D5 20          SKC                 ;TRANSMIT A LOGIC 1, SO=1
436 1D6 DF          JP TRANSO           ;TRANSMIT A LOGIC 0, SO=0
437 1D7 3B          LBI COMP1
438 1D8 74          STII 4
439 1D9 3B          LBI COMP1
440 1DA 00          CLRA
441 1DB 51    INC2: AISC 1              ;240 USEC. LOOP
442 1DC 21          SKE
443 1DD DB          JP INC2
444 1DE 48          RET
445 1DF 3361  TRANSO: LEI 1             ;SET EN3=0
446 1E1 3B          LBI COMP1
447 1E2 73          STII 3
448 1E3 3B          LBI COMP1
449 1E4 00          CLRA
450 1E5 51    INC3  AISC 1              ;180 USEC. LOOP
451 1E6 21          SKE
452 1E7 E5          JP INC3
453 1E8 48          RET
454                 END
```

```
BDPNT  003D   BEGIN  0025   BLK1  001B   BLK2  001C
BLK3   001D   BLK4   001E   BR1   0015   BR10  0076
BR11   007A   BR12   0085   BR13  0089   BR14  0094
BR15   0099   BR16   009F   BR17  009A   BR18  00BC
BR19   00F3   BR2    001B   BR20  00FC   BR21  0110
BR22   0111   BR23   0123   BR24  0124   BR25  0136
BR26   0137   BR27   0149   BR28  014A   BR29  0165
```

```
BR7    004F    BR8    0067    BR9    006B    BRPNT  003E *
CKSUM  001F    CNTROL 003A    COMP1  003C    COMP2  003D
COUNT0 002E    COUNT1 002D    COUNT2 002C *   FLAG1  000F
IDNUM  001A    INC1   01D2    INC2   01DB    INC3   01E5
INITX  00ED    IRIN   002F    LOOK1  0036    LP1    01BD
MOUSE  0030    NVRAM  0183    NXBYTE 0107    REPEAT 005B
SAMPLE 0174    SCRATC 003F    START  000E    STRTX  00D0
SUB1   0176    SUB2   0181    SUB3   0182    SUB4   019B
SUB5   01A1    SUB6   0186    TIMER  01B2    TRANS0 01DF
TXMIT  01CB

NO ERROR LINES

  489  ROM WORDS USED

COP   420  ASSEMBLY

SOURCE CHECKSUM = 50C2

OBJECT CHECKSUM = 0CB2

INPUT FILE    GENUSER.JANN SRC  VN:    1

OBJECT FILE   GENUSER.JANN LM

 1                ;            RECEIVER SOFTWARE LISTING
 2                ;
 3                ;
 4                ;            "COPYRIGHT 1982
 5                ;             CHAMBERLAIN MANUFACTURING CORPORATION
 6                ;             ALL RIGHTS RESERVED"
 7                ;
 8                ;
 9                ;
10      001F              CKSUM = 1,15              ;CHECKSUM
11      001E              BLK4 = 1,14               ;INFORMATION BLOCK 4
12      001D              BLK3 = 1,13               ;INFORMATION BLOCK 3
13      001C              BLK2 = 1,12               ;INFORMATION BLOCK 2
14      001B              BLK1 = 1,11               ;INFORMATION BLOCK 1
15      001A              IDNUM = 1,10              ;I. D. NUMBER
```

```
16      002F            IRIN = 2,15             ;INFRARED INPUT COUNTER
17      002E            COUNT0 = 2,14           ;GP COUNTER 0
18      002D            COUNT1 = 2,13           ;GP COUNTER 1
19      002C            COUNT2 = 2,12           ;GP COUNTER 2
20      003F            SCRATC = 3,15           ;SCRATCH PAD REGISTER
21      003E            BRPNT = 3,14            ;BR MEMORY POINTER
22      003D            BDPNT = 3,13            ;BD MEMORY POINTER
23      003C            COMP1 = 3,12            ;COMPARE REGISTER 1
24      003B            COMP2 = 3,11            ;COMPARE REGISTER 2
25      003A            CNTROL = 3,10           ;CONTROL WORD REGISTER
26      0030            NOISE = 3,0             ;SCRATCH PAD REGISTER
27      000F            RCKSUM = 0,15
28      000E            RBLK4 = 0,14
29      000D            RBLK3 = 0,13
30      000C            RBLK2 = 0,12
31      000B            RBLK1 = 0,11
32      000A            RIDNUM = 0,10
33      0009            FLAG1 = 0,9
34              ;INITIALIZATION
35 000 00               CLRA
36 001 3E               LBI SCRATC
37 002 7F               STII 15
38 003 40               COMP
39 004 3E               LBI SCRATC
40 005 333A             OMG                     ;INITIALIZE G PORT
41 007 333C             CAM0                    ;INITIALIZE L PORT
42 009 3360             LEI 0                   ;TRISTATE L PORT
43 00B 50               CAB                     ;SET SI AS SHIFT REGISTER
44 00C 333E             OBD                     ;SET SO = 0
45                                              ;INITIALIZE D PORT
46 00E 08               LBI FLAG1
47 00F 4D               SMB 0                   ;SET READ FROM NVRAM FLAG
48 010 39               LBI CNTROL              ;STORE CONTROL WORD
49 011 77               STII 7
50 012 3387             LBI 0,7                 ;INSTITUTE AN ARRAY RECALL
```

```
51 014 333E          OBD                ; CYCLE
52 016 0E            LBI 0,15
53 017 333E          OBD
54 019 69D7          JSR NVRAM
55 01B 3F    START:  LBI NOUSE          ; CHECK FOR PROGRAM SWITCH
56 01C 332E          INL                ; CLOSURE
57 01E 13            SKMBZ 3
58 01F 60F1          JMP BEGIN          ; NO, MONITOR RECEIVE INPUT
59 021 08    NGEN:   LBI FLAG1          ; YES, PROGRAM SWITCH CLOSED
60 022 47            SMB 1              ; SET PROGRAM FLAG
61 023 3F            LBI NOUSE          ; CHECK FOR ALL ZERO INPUTS
62 024 00            CLRA
63 025 06            X 0
64 026 1A            LBI BLK1
65 027 21            SKE                ; BLK1=0
66 028 F8            JP BR1
67 029 1B            LBI BLK2
68 02A 21            SKE                ; BLK2=0
69 02B F8            JP BR1
70 02C 1C            LBI BLK3
71 02D 21            SKE                ; BLK3=0
72 02E F8            JP BR1
73 02F 1D            LBI BLK4
74 030 21            SKE                ; BLK4=0
75 031 F8            JP BR1
76 032 0A            LBI 0,11           ; ALL BLOCKS EQUAL 0
77 033 7F            STII 15            ; STORE DEFAULT NUMBER
78 034 7E            STII 14
79 035 7D            STII 13
80 036 7C            STII 12
81 037 FA            JP BR2
82 038 6A3D  BR1:    JSR RANDOM         ; GENERATE NEW RANDOM NUMBER
83 03A 0A    BR2:    LBI 0,11           ; TRANSFER GENERATED NUMBER
84 03B 15            LD 1               ; TO BLOCKS 1 THRU 4
85 03C 06            X 0
```

```
 86 03D 0B              LBI 0,12
 87 03E 15              LD 1
 88 03F 06              X 0

 89 040 0C              LBI 0,13
 90 041 15              LD 1
 91 042 06              X 0
 92 043 0D              LBI 0,14
 93 044 15              LD 1
 94 045 06              X 0
 95 046 08              LBI FLAG1               ;PROGRAM NEW CODE
 96 047 4C              RMB 0                   ;TO NVRAM
 97 048 39              LBI CNTROL
 98 049 73              STII 3
 99 04A 69D7            JSR NVRAM
100 04C 3F              LBI NOUSE               ;INSTITUTE A STORE COMMAND
101 04D 7D              STII 13
102 04E 3F              LBI NOUSE
103 04F 333A            OMG
104 051 3F              LBI NOUSE
105 052 7F              STII 15                 ;TERMINATE STORE COMMAND
106 053 3F              LBI NOUSE
107 054 333A            OMG
108             ;CALCULATE DATA BLOCK CHECKSUM
109 056 00              CLRA
110 057 1A              LBI BLK1
111 058 31              ADD
112 059 1B              LBI BLK2
113 05A 31              ADD
114 05B 1C              LBI BLK3
115 05C 31              ADD
116 05D 1D              LBI BLK4
117 05E 31              ADD
118 05F 1E              LBI CKSUM
119 060 06              X 0
```

```
120                 ;TRANSMIT SYNC PULSE
121 061 3B      INITX:  LBI COMP1           ;INITILIZE COMPARE REGISTER
122 062 78              STII 8
123 063 00              CLRA
124 064 3B              LBI COMP1
125 065 3369            LEI 9               ;SET EN3=1, SO=1
126 067 51      BR19:   AISC 1              ;TRANSMIT LOGIC 1
127 068 21              SKE                 ;FOR 500 USEC.
128 069 E7              JP BR19
129 06A 3361            LEI 1               ;SET EN3=0, SO=0
130 06C 3B              LBI COMP1           ;TRANSMIT LOGIC 0
131 06D 7D              STII 13             ;FOR 1.5 MSEC
132 06F 00              CLRA
133 06F 3B              LBI COMP1
134 070 51      BR20:   AISC 1
135 071 44              NOP
136 072 21              SKE
137 073 6070            JMP BR20
138 075 44              NOP
139 076 44              NOP
140                 ;TRANSMIT DATA BLOCK
141 077 3C              LBI BDPNT           ;INITIALIZE BD POINTER TO
142 078 7B              STII 11             ;FIRST AND LAST LOCATION OF
143 079 3A              LBI COMP2           ;DATA TO BE TRANSMITTED
144 07A 70              STII 0
145 07B 3C      NXBYTE: LBI BDPNT
146 07C 25              LD 2
147 07D 50              CAB
148 07E 3361            LEI 1               ;SET EN3=0,SO=0

149 080 13              SKMBZ 3             ;CHECK LOGIC STATUS OF BIT 3
150 081 84              JP BR21             ;RAM B3=1
151 082 32              RC                  ;RAM B3=0
152 083 85              JP BR22
153 084 22      BR21:   SC
```

```
154 085 6A1F   BR22:   JSR TXMIT
155 087 3E             LBI SCRATC              ;160 USEC. DELAY
156 088 7E             STII 14
157 089 3E             LBI SCRATC
158 08A 06             X 0
159 08B 51             AISC 1
160 08C 8B             JP -1
161 08D 44             NOP
162 08E 3C             LBI BDPNT
163 08F 25             LD 2
164 090 50             CAB
165 091 3361           LEI 1                   ;SET ENB=0,SO=0
166 093 03             SKMBZ 2                 ;CHECK LOGIC STATUS OF BIT 2
167 094 97             JP BR23                 ;RAM B2=1
168 095 32             RC                      ;RAM B2=0
169 096 98             JP BR24
170 097 22     BR23:   SC
171 098 6A1F   BR24:   JSR TXMIT
172 09A 3E             LBI SCRATC              ;160 USEC. DELAY
173 09B 7E             STII 14
174 09C 3E             LBI SCRATC
175 09D 06             X 0
176 09E 51             AISC 1
177 09F 9E             JP -1
178 0A0 44             NOP
179 0A1 3C             LBI BDPNT
180 0A2 25             LD 2
181 0A3 50             CAB
182 0A4 3361           LEI 1                   ;SET ENB=0,SO=0
183 0A6 11             SKMBZ 1                 ;CHECK LOGIC STATUS OF BIT 1
184 0A7 AA             JP BR25                 ;RAM B1=1
185 0A8 32             RC                      ;RAM B1=0
186 0A9 AB             JP BR26
187 0AA 22     BR25:   SC
188 0AB 6A1F   BR26:   JSR TXMIT
```

```
189 0AD 3E          LBI SCRATC
190 0AE 7E          STII 14
191 0AF 3E          LBI SCRATC
192 0B0 06          X 0                 ;160 USEC. DELAY
193 0B1 51          AISC 1
194 0B2 B1          JP -1
195 0B3 44          NOP
196 0B4 3C          LBI BDPNT
197 0B5 25          LD 2
198 0B6 50          CAB
199 0B7 3361        LEI 1               ;SET EN3=0, SO=0
200 0B9 01          SKMBZ 0             ;CHECK LOGIC STATUS OF BIT 0
201 0BA BD          JP BR27             ;RAM BO=1
202 0BB 32          RC                  ;RAM BO=0
203 0BC BE          JP BR28
204 0BD 22    BR27: SC
205 0BE 6A1F  BR28: JSR TXMIT

206 0C0 3C          LBI BDPNT           ;INCREMENT BD POINTER
207 0C1 05          LD 0
208 0C2 51          AISC 1
209 0C3 44          NOP
210 0C4 06          X 0
211 0C5 05          LD 0
212 0C6 3A          LBI COMP2           ;LAST MEMORY LOCATION
213 0C7 21          SKE                 ;TRANSMITTED
214 0C8 607B        JMP NXBYTE          ;NO
215 0CA 44          NOP                 ;YES
216 0CB 33A1        LEI 1
217 0CD 3B          LBI COMP1
218 0CE 78          STII 8
219 0CF 00          CLRA
220 0D0 3B          LBI COMP1
221 0D1 44          NOP
222 0D2 44          NOP
```

```
223             ;TRANSMIT TERMINATOR
224 0D3 3B              LBI COMP1
225 0D4 78              STII 8
226 0D5 00              CLRA
227 0D6 3B              LBI COMP1
228 0D7 3369            LEI 9                   ;SET EN3=1, SO=1
229 0D9 51      BR29:   AISC 1                  ;TRANSMIT LOGIC 1 FOR 500 USEC
230 0DA 21              SKE
231 0DB D9              JP BR29
232 0DC 3361            LEI 1                   ;SET EN3=0,SO=0
233             ;TRANSMIT BLANK TIME OF  5 MSEC.
234 0DE 3A              LBI COMP2               ;INITIALIZE COMPARE REGISTERS
235 0DF 70              STII 0
236 0E0 78              STII 8
237 0E1 2C              LBI COUNT1              ; RESET COUNTER REGISTERS
238 0E2 70              STII 0
239 0E3 70              STII 0
240 0E4 6A06            JSR TIMER
241 0E6 3F              LBI NOUSE               ;CHECK FOR PROGRAM
242 0E7 332E            INL                     ;SWITCH CLOSURE
243 0E9 13              SKMBZ 3
244 0EA F1              JP BEGIN                ;SWITCH NOT CLOSED
245 0EB 08              LBI FLAG1               ;SWITCH CLOSED
246 0EC 11              SKMBZ 1                 ;IS PROGRAM FLAG SET?
247 0ED 6061            JMP INITX               ;YES RETRANSMIT DATA
248 0EF 6021            JMP NGEN                ;NO GENERATE NEW RANDOM NUMBE
249 0F1 08      BEGIN:  LBI FLAG1
250 0F2 45              RMB 1
251 0F3 3362            LEI 2
252 0F5 3C              LBI BDPNT               ;INITIALIZE RECEIVED DATA
253 0F6 7A              STII 10                 ;MEMORY POINTER
254 0F7 2D              LBI COUNTO              ;RESET COUNTER IRIN
255 0F8 70              STII 0
256 0F9 70              STII 0
257 0FA 09              LBI RIDNUM
```

```
258 0FB 70              STII 0
259 0FC 70              STII 0
260 0FD 6104            JMP JT4
261 0FF 44              NOP
262 100 3360            LEI 0
263 102 601B            JMP START
264 104 70      JT4:     STII 0
265 105 70              STII 0
266 106 70              STII 0
267 107 70              STII 0
268 108 2D              LBI COUNTO
269 109 3301            SKGBZ 0                 ;IS INPUT A LOGIC 1 ?
270 10B CE              JP LOOK1                ;YES
271 10C 60F1            JMP BEGIN                ;NO
272 10F 00      LOOK1:  CLRA
273 10F 51              AISC 1
274 110 3301    BR5:    SKGBZ 0                 ;IS INPUT STILL A LOGIC 1 ?
275 112 D5              JP BR4                  ;YES
276 113 60F1            JMP BEGIN                ;NO
277 115 51      BR4:    AISC 1
278 116 D0              JP BR5
279 117 06              X 0
280 118 51              AISC 1
281 119 06              X 0
282 11A 05              LD 0
283 11B 5A              AISC 10
284 11C 610E            JMP LOOK1
285 11E 2E              LBI IRIN                ;VALID 10 MSEC. BLANK TIME
286 11F 00              CLRA
287 120 3301            SKGBZ 0                 ;IS INPUT A LOGIC 0 ?
289 123 51      BR 6    AISC 1
290 124 3301            SKGBZ 0                 ;IS INPUT STILL A LOGIC 0 ?
291 126 E8              JP BR7                  ;NO, RISING TRANSITION DETECTI
292 127 E3              JP BR6                  ;YES
293 128 06      BR7:    X 0                     ;STORE COUNT IN IRIN
```

```
294 129 05              LD 0                    ;SUBTRACT CORRECTION FACTOR
295 12A 5E              AISC 14                 ;FOR SAMPLE VALUE
296 12B 44              NOP
297 12C 2C              LBI COUNT1              ;STORE SAMPLE VALUE
298 12D 06              X 0                     ;IN COUNTER 1
299 12E 2E              LBI IRIN                ;CHECK FOR OVERFLOW
300 12F 00              CLRA
301 130 21              SKE
302 131 F4              JP REPEAT               ;NO OVERFLOW
303 132 601B            JMP START               ;OVERFLOW EXISTS
304 134 3301  REPEAT:   SKGBZ 0                 ;CHECK FOR FALLING TRANSITION
305 136 F4              JP -2                   ;INPUT STILL HIGH
306 137 69C8            JSR SAMPLE              ;FALLING TRANSITION DETECTED
307 139 3C              LBI BDPNT               ;POINT TO RECEIVE DATA POINTEI
308 13A 35              LD 3
309 13B 50              CAB
310 13C 20              SKC
311 13D 6140            JMP BR8
312 13F 4B              SMB 3
313 140 3301  BR8:      SKGBZ 0                 ;LOOK FOR FALLING TRANSITION
314 142 C4              JP BR9                  ;INPUT STILL HIGH
315 143 C0              JP BR8                  ;INPUT STILL LOW
316 144 3301  BR9:      SKGBZ 0
317 146 C4              JP -2                   ;INPUT STILL HIGH
318 147 69C8            JSR SAMPLE              ;FALLING TRANSITION DETECTED
319 149 3C              LBI BDPNT               ;POINT TO RECEIVE DATA POINTE
320 14A 35              LD 3
321 14B 50              CAB
322 14C 20              SKC
323 14D CF              JP BR10
324 14E 4A              SMB 2
325 14F 3301  BR10:     SKGBZ 0                 ;LOOK FOR FALLING TRANSITION
326 151 D3              JP BR11                 ;INPUT STILL HIGH
327 152 CF              JP BR10                 ;INPUT STILL LOW
328 153 3301  BR11:     SKGBZ 0
```

```
329 155 D3              JP  -2                  ;INPUT STILL HIGH
330 156 69C8            JSR SAMPLE              ;FALLING TRANSITION DETECTED
331 158 3C              LBI BDPNT               ;POINT TO RECEIVE DATA POINTE
332 159 35              LD  3
333 15A 50              CAB
334 15B 20              SKC
335 15C DE              JP  BR12
336 15D 47              SMB 1
337 15E 3301    BR12:   SKGBZ 0                 ;LOOK FOR FALLING TRANSITION
338 160 E2              JP  BR13                ;INPUT STILL HIGH
339 161 DE              JP  BR12                ;INPUT STILL LOW
340 162 3301    BR13:   SKGBZ 0
341 164 E2              JP  -2                  ; INPUT STILL HIGH
342 165 69C8            JSR SAMPLE              ;FALLING TRANSITION DETECTED
343 167 3C              LBI BDPNT               ;POINT TO RECEIVE DATA POINT
344 168 35              LD  3
345 169 50              CAB
346 16A 20              SKC
347 16B ED              JP  BR14
348 16C 4D              SMB 0
349 16D 3C      BR14:   LBI BDPNT               ;INCREMENT BD POINTER
350 16E 05              LD  0
351 16F 51              AISC 1                  ;LAST MEMORY LOCATION FILLED
352 170 F2              JP  BR15                ;NO
353 171 F8              JP  BR16                ;YES
354 172 06      BR15:   X   0
355 173 3301    BR17:   SKGBZ 0                 ;LOOK FOR FALLING TRANSITION
356 175 4134            JMP REPEAT              ;INPUT STILL HIGH
357 177 F3              JP  BR17                ;INPUT STILL LOW
358             ;CALCULATE RECEIVED DATA CHECKSUM
359 178 00      BR16:   CLRA
360 179 09              LBI RIDNUM
361 17A 31              ADD
362 17B 0A              LBI RBLK1
363 17C 31              ADD
```

```
364 17D 0B          LBI RBLK2
365 17E 31          ADD
366 17F 0C          LBI RBLK3
367 180 31          ADD
368 181 0D          LBI RBLK4
369 182 31          ADD
370 183 0E          LBI RCKSUM
371 184 21          SKE                 ;CHECKSUM CORRECT
372 185 601B        JMP START           ;CHECKSUM INCORRECT
373 187 08          LBI FLAG1
374 188 4D          SMB 0               ;SET READ FROM NVRAM FLAG
375 189 39          LBI CNTROL          ;STORE CONTROL WORD
376 18A 77          STII 7
377 18B 3387        LBI 0,7             ;INSTITUTE AN ARRAY RECALL
378 18D 333E        ORD                 ;CYCLE
379 18F 0E          LBI 0,15
380 190 333E        OBD
381 192 69D7        JSR NVRAM
382 194 1A          LBI BLK1            ;CHECK FOR RANDOM
383 195 15          LD 1                ;NUMBER MATCH
384 196 21          SKE
385 197 601B        JMP START
386 199 1B          LBI BLK2
387 19A 15          LD 1
388 19B 21          SKE
389 19C 601B        JMP START
390 19E 1C          LBI BLK3
391 19F 15          LD 1
392 1A0 21          SKE
393 1A1 601B        JMP START
394 1A3 1D          LBI BLK4
395 1A4 15          LD 1
396 1A5 21          SKE
397 1A6 601B        JMP START
398 1A8 09          LBI RIDNUM
```

```
399 1A9 4B              SMB 3
400 1AA 05              LD 0
401 1AB 40              COMP
402 1AC 333C            CAM0
403 1AE 3364            LEI 4
404 1B0 3A              LBI COMP2           ;INITILIZE COMPARE REGISTERS
405 1B1 7F              STII 15
406 1B2 7F              STII 15
407 1B3 2F              LBI IRIN
408 1B4 70              STII 0
409 1B5 2C      JT1:    LBI COUNT1          ;INTTIALIZE COUNTER REGISTERS
410 1B6 70              STII 0
411 1B7 70              STII 0
412 1B8 6A06            JSR TIMER
413 1BA 2E              LBI IRIN
414 1BB 06              X 0
415 1BC 51              AISC 1
416 1BD 61C1            JMP JT2
417 1BF 61C4            JMP JT3

418 1C1 06      JT2:    X 0
419 1C2 61B5            JMP JT1
420 1C4 3360    JT3:    LEI 0
421 1C6 601B            JMP START
422             ;SUBROUTINES
423             ;           SUBROUTINE SAMPLE
424 1C8 2C      SAMPLE: LBI COUNT1
425 1C9 00              CLRA                ;SETUP SAMPLING VALUE
426 1CA 51      SUB1:   AISC 1
427 1CB 44              NOP
428 1CC 21              SKE
429 1CD CA              JP SUB1
430 1CE 3301            SKGBZ 0             ;READ INPUT
431 1D0 61D5            JMP SUB2
432 1D2 22              SC                  ;INPUT IS A LOGIC 0
```

```
433 1D3 61D6          JMP SUB3
434 1D5 32      SUB2: RC                    ;INPUT IS A LOGIC 1
435 1D6 48      SUB3: RET
436           ;              SUBROUTINE NVRAM
437 1D7 3B      NVRAM: LBI COMP1            ;INITIALIZE BD MEMORY POINTER
438 1D8 7F            STII 15               ;AND COMPARE REGISTER
439 1D9 7B            STII 11
440 1DA 3C      SUB6: LBI BDPNT
441 1DB 25            LD 2                  ;OUTPUT ADDRESS POINTED TO
442 1DC 51            ATOC 1                ;BY COMPARE REGISTER
443 1DD 50            CAB
444 1DE 333E          OBD
445 1E0 39            LBI CNTROL            ;OUTPUT CONTROL WORD TO
446 1E1 333A          OMG                   ;MEMORY
447 1E3 08            LBI FLAG1             ;CHECK FOR READ OR WRITE
448 1E4 01            SKMBZ 0                 ;FLAG
449 1E5 EF            JP SUB4               ;READ FLAG
450 1E6 3C            LBI BDPNT             ;WRITE FLAG
451 1E7 25            LD 2
452 1E8 50            CAB
453 1E9 00            CLRA
454 1EA 333C          CAMD                  ;WRITE TO NVRAM
455 1EC 3344          LEI 4                 ;ENABLE L DRIVERS
456 1EE F5            JP SUB5
457 1EF 3C      SUB4: LBI BDPNT
458 1F0 25            LD 2
459 1F1 50            CAB
460 1F2 332E          INI                   ;READ FROM NVRAM
461 1F4 06            X 0
462 1F5 3F      SUB5: LBI NOUSE             ;INITIATE A DESELECT
463 1F6 7F            STII 15
464 1F7 3F            LBI NOUSE
465 1F8 333A          OMG
466 1FA 3360          LEI 0                 ;DISABLE L DRIVERS
467 1FC 3C            LBI BDPNT
```

```
468 1FD 05              LD 0
469 1FE 51              AISC 1
470 1FF 06              X 0                     ;STORE NEW VALUE FOR

471 200 05              LD 0                    ;MEMORY POINTER
472 201 3B              LBI COMP1               ;LAST MEMORY LOCATION
473 202 21              SKE                     ;ACCESSED?
474 203 A1DA            JMP SUB6                 ;NO
475 205 48              RET                     ;YES
476             ;                   SUBROUTINE TIMER
477 206 2D      TIMER:  LBI COUNT0
478 207 00              CLRA
479 208 51              AISC 1
480 209 C8              JP  -1                  ;640 USEC. DELAY LOOP
481 20A 05              LD 0                    ;COUNT0 CONTENTS TO ACCU.
482 20B 51              AISC 1                  ;ADD 1 TO ACCU.
483 20C D1              JP LP1
484 20D 07              XDS                     ;STORE ZERO IN COUNT0
485 20E 05              LD 0                    ;DECREMENT TO COUNT1
486 20F 51              AISC 1                  ;ADD 1 TO COUNT1
487 210 44              NOP
488 211 06      LP1:    X 0
489 212 3A              LBI COMP2               ;COMPARE COUNTERS TO
490 213 05              LD 0                    ;COMPARE REGISTERS
491 214 2C              LBI COUNT1
492 215 21              SKE                     ;SKIP IF MOST SIGN. COUNT
493 216 6206            JMP TIMER                ;EQUALS COMP2
494 218 3B              LBI COMP1
495 219 05              LD 0
496 21A 2D              LBI COUNT0              ; SKIP IF LEAST SIGN. COUNT
497 21B 21              SKE                     ;EQUALS COUNT0
498 21C 6206            JMP TIMER                ;NO, INSTITUTE ANOTHER CYCLE
499 21E 48              RET
500             ;                   TRANSMIT SUBROUTINE
501 21F 3B      TXMIT:  LBI COMP1
```

```
502 220 73          STII 3
503 221 00          CLRA
504 222 3B          LBI COMP1
505 223 44          NOP
506 224 3369        LEI 9                   ;SET EN3=1,SO=1
507 226 51    INC1: AISC 1                  ;250 USEC. LOOP
508 227 21          SKE
509 228 E6          JP INC1
510 229 20          SKC                     ;TRANSMIT A LOGIC 1, SO=1
511 22A F3          JP TRANSO               ;TRANSMIT A LOGIC 0, SO=0
512 22B 3B          LBI COMP1
513 22C 74          STII 4
514 22D 3B          LBI COMP1
515 22E 00          CLRA
516 22F 51    INC2: AISC 1                  ;240 USEC. LOOP
517 230 21          SKE
518 231 EF          JP INC2
519 232 48          RET
520 233 3361  TRANSO: LEI 1                 ;SET EN3=0
521 235 3B          LBI COMP1
522 236 73          STII 3
523 237 3B          LBI COMP1
524 238 00          CLRA
525 239 51    INC3: AISC 1                  ;180 USEC. LOOP
526 23A 21          SKE
527 23B F9          JP INC3
528 23C 48          RET
529           ;RANDOM NUMBER GENERATOR
530           ;SEED CONTAINED IN (1,11) THRU (1,14) IS USED AS A BASIS
531           ;FOR A NEW RANDOM NUMBER GENERATED IN AND STORED IN
532           ;(0,11) THRU (0,14)
533 23D 3C     RANDOM: LBI BDPNT
534 23E 7E          STII 14
535 23F 1A          LBI BLK1                ;XOR BITS 1 AND 2 OF BLK1
536 240 05          LD 0                    ;USING INTERNAL SHIFT REGISTER
```

```
537 241 4F              XAS
538 242 4F              XAS
539 243 02              XOR
540 244 3F              LBI NOUSE
541 245 06              X 0
542 246 03              SKMBZ 2
543 247 625E            JMP SUB10
544 249 6259            JMP SUB11
545 24B 3C      SUB12:  LBI BDPNT       ;SHIFT LEFT BLOCKS 1 THRU
546 24C 25              LD 2            ;4 USING XOR OF BLOCK
547 24D 50              CAB             ;1 BITS 1 AND 2 AS
548 24E 15              LD 1             ;ENTRY FOR THE LEAST
549 24F 4F              XAS             ;SIGNIFICANT BIT OF BLK4
550 250 4F              XAS
551 251 17              XDS 1
552 252 4E              CBA
553 253 3C              LBI BDPNT
554 254 26              X 2
555 255 50              CAB
556 256 13              SKMBZ 3
557 257 625E            JMP SUB10
558 259 1A      SUB11:  LBI 1,11        ;XOR OPERATION = 0
559 25A 333E            ORD             ;FORCE LEAST SIGN. BIT ON
560 25C 6261            JMP SUB13        ;SHIFT REGISTER TO 0
561 25E 1E      SUB10:  LBI 1,15        ;XOR OPERATION = 1
562 25F 333E            ORD             ;FORCE LEAST SIGN. BIT ON
563 261 3C      SUB13:  LBI BDPNT       ;SHIFT REGISTER TO 1
564 262 05              LD 0
565 263 55              AISC 5
566 264 E7              JP SUB14
567 265 624B            JMP SUB12        ;CONTINUE UNTIL FINISHED
568 267 48      SUB14:  RET             ;TERMINATE PROCESS
569             ;
570             ;
571             ;
```

```
572                ;          "COPYRIGHT 1982
573                ;           CHAMBERLAIN MANUFACTURING CORPORATION.
574                ;           ALL RIGHTS RESERVED"
575                ;
576                ;
577                          END
```

I claim as my invention:

1. Apparatus for controlling a receiver with a remote radio frequency transmitter comprising a first micro-processor and a first memory means in said receiver for storing at least one address code, non-radio frequency transmitting means in said receiver, switch means for energizing said non-radio frequency transmitting means to transmit an address code, a non-radio frequency receiving means in said transmitter for receiving said address code, a second micro-processor and second memory means in said transmitter for storing said address code, radio frequency radiating means in said transmitter for radiating said address code, receiving means in said receiver for receiving said radio frequency radiated address code, said first micro-processor in said receiver comparing the received address code with the address code stored in said first memory, an output circuit energized by said comparing means when said addresses are the same, wherein said first memory means comprises a non-volatile memory and a programmable read only memory and said second memory means comprises a non-volatile memory and a programmable read only memory.

2. Apparatus according to claim 1 wherein said first micro-processor is programmed to operate as a pseudo random number generator to generate a plurality of different address codes to allow the address codes in said transmitter and receiver to be changed.

3. Apparatus according to claim 2 wherein said non-radio frequency transmitting means is a light radiator.

4. Apparatus according to claim 3 wherein said non-radio frequency receiving means in said transmitter is a light detector.

5. Apparatus according to claim 2 wherein said non-radio frequency transmitting means in said receiver is an electrical conductor.

6. Apparatus according to claim 4 wherein said address code comprises a serial binary code of pulse length modulated form of a plurality of word lengths.

7. Apparatus according to claim 6 wherein said address code contains a binary check block for indicating whether a correct signal has been received.

8. Apparatus according to claim 6 wherein said address code contains a binary synchronizing block for synchronizing the transmitter and receiver.

9. Apparatus according to claim 6 wherein said address code contains a terminating block.

10. Apparatus according to claim 6 wherein said address code is repeated and each address code is separated by a blanking period.

11. Apparatus according to claim 2 including a first octal latch in said receiver which is connected to said first micro-processor.

12. Apparatus according to claim 2 including a second octal latch in said transmitter.

* * * * *