(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 795 850 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
17.09.1997 Bulletin 1997/38

(51) Int. Cl.[6]: **G10H 1/00**, G10H 1/20, G10H 7/00

(21) Application number: **97106076.9**

(22) Date of filing: **27.12.1989**

(84) Designated Contracting States:
AT BE CH DE ES FR GB IT LI LU NL SE

(30) Priority: **03.01.1989 US 292966**

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
90900583.7 / 0 452 347

(71) Applicant: **The Hotz Corporation**
Thousand Oaks, CA 91360 (US)

(72) Inventor: **Hotz,Jimmy C.**
Thousand Oaks, California 91360 (US)

(74) Representative: **BROOKES & MARTIN**
High Holborn House
52/54 High Holborn
London, WC1V 6SE (GB)

Remarks:
This application was filed on 14 - 04 - 1997 as a divisional application to the application mentioned under INID code 62.

(54) Electronic musical instrument controller

(57) A MIDI-compatible musical instrument controller includes a keyboard having a plurality of keys (12) and circuitry for electronically scanning the keyboard and for producing individual key-depression signals for each key being depressed. Each of the individual key depression signals identifies the key with which is associated, commences when the key is depressed, and terminates when the key with which it is associated is released. A plurality of note tables is provided for converting each of the individual key depression signals to MIDI note-identifying information. Each note table defines each key as one or more preselected musical notes such that no two of such tables define the plurality of keys with the same MIDI note-identifying information. One of the note tables is selected in response to a user command.

Note-start circuitry, responsive to the commencement of an individual key depression signal, is provided for generating MIDI note-one signals corresponding to the one or more musical notes defined for the individual depressed key in the note table selected at the time the individual key depression signal commences. Note-stop circuitry, responsive to the termination of an individual key depression signal, is provided for generating MIDI note-off signals corresponding to the one or more musical notes defined for the individual released key in the note table which was selected during the time when the individual released key was depressed.

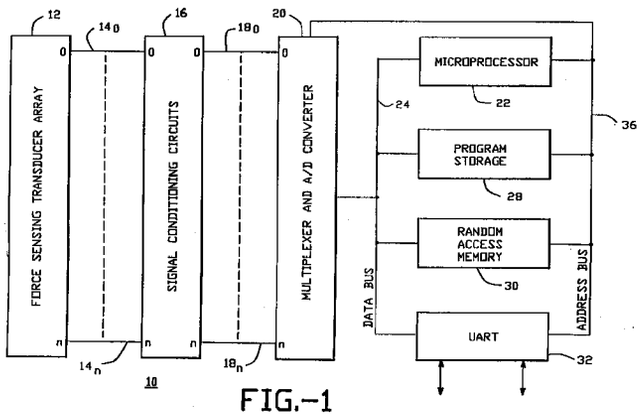


FIG.-1

EP 0 795 850 A2

Description

The present invention relates generally to electronic musical instruments. More particularly, the present invention relates to a versatile user-programmable musical instrument and more particularly to a controller for such instruments with the capability of transparently altering pitch and velocity for the user, so that only correct values relating to scale and chord value are available at any given moment.

Still more particularly; the present invention is directed to a musical controller that may be played in conjunction with the playback of a CD or similar pre-recorded sound recording on which is stored synchronously with the sound recording CHORD and/or SCALE change information which may be sent to the instrument over an interface so that the musician has creative input but does not have the option of playing an incorrect chord or note. The present invention is also directed to a method of operating such a controller.

BACKGROUND OF THE INVENTION

Electronic keyboard and other electronic musical instruments are known in the prior art. Also known are electronic musical keyboard instruments which generate tone and velocity information compatible with the MIDI (Musical Instrument Digital Interface) standard which has come into wide usage in recent years. Numerous keyboard instruments, such as those manufactured by Roland, provide a powerful measure of performance.

Electronic musical instruments which provide for an automatic accompaniment to be generated by the instrument in response to a performer playing the instrument are also known in the art. Examples of such instruments are found in US-A-4,433,601, 4,508,002 and 4,682,526.

In addition, some keyboard musical instruments provide for the automatic sharpening or flatting of a note on the white keys in response to a signal indicating that a scale is to be played in a key other than "C". An example of such an instrument is shown in US-A-4,513,650.

US-A-4 777 857 relates to a MIDI address converter and router designed to be inserted into a MIDI communication line for carrying serial data between instruments. The routing function effectively connects and disconnects MIDI cables in different ways to overcome dissimilar implementations of the MIDI standard that can occur in different manufacturers products. The address converter allows a first byte called a MIDI address following a key on key off or control change status byte to be converted allowing transposition or performance of a control increment / de-increment operation. Only bytes relating to these MIDI addresses are manipulated, and even then they may only be incremented or de-incremented by the same fixed values as preselected by the user.

While these prior art schemes and devices have been successful and have performed their intended functions, there still remains a need for improvement.

BRIEF DESCRIPTION OF THE INVENTION

In accordance with the present invention there is provided an electronic musical controller comprising:

a plurality of input device signals;

a first memory for storing a plurality of different translation tables, said translation tables relating said input device signals with corresponding control signals;

a source of translation table selection signals; and

a translator for outputting said control signals responsive to said input device signals, said translation tables and said translation table selection signals.

In accordance with the present invention there is provided a method of operating an electronic musical controller in a play-along mode with a pre-recorded sound recording, and a pre-recorded data recording, said data recording containing translation table selection signals synchronized with events in said sound recording, each said translation table corresponding to a particular chord and a particular scale, said method comprising:

storing a plurality of different translation tables, each of said translation tables corresponding to a particular chord and a particular scale;

playing the sound recording and the data recording simultaneously to form at least a translation table selection signal output containing translation table selection signals;

forming a stream of first input signals;

receiving said translation table selection signals from said translation table selection signal output; and

translating said first input signals into a stream of control signals for controlling an output of the electronic musical controller by applying said first input signals to a translation table most recently selected by a most recently received translation table selection signal so that said control signals cause the electronic musical controller to output music signals in response to said first input signals which music signals are only within said particular chord and said particular scale corresponding to said translation table selected by said most recently received translation table selection signal.

Further in accordance with the invention there is provided a method for the operation of an electronic musical controller comprising the steps of:

storing a plurality of different translation tables in a first memory, said translations tables relating to a plurality of input device signals with corresponding control signals;

providing translation table selection signals; and

outputting said control signals responsive to said input device signals, said translation tables and said translation table selection signals.

Further features of the invention are defined in the appended subsidiary claims.

According to the one presently preferred aspect of the present invention, chord and scale information may be stored along with pre-recorded music on musical media such as a CD disk and may be sent to the system of the present invention via a MIDI interface so that a musician can "play along" with pre-recorded music. Since the chord change and any scale change timing is synchronously provided by the pre-recorded media, the musician has creative input but does not have the option of playing an incorrect chord or note.

In this embodiment, the code necessary to implement chord changes requires only a fraction of the memory necessary to store melody and chord notes on a CD for playing along, thus making such an embodiment a practical reality. For instance, a typical popular music selection would require up to 500K bytes of information to reproduce the parts contained on the recording. A 10 song album could require 5M bytes or more of memory, and would not afford creative input by the listener. On the other hand, with the present invention, only the MIDI message per chord change or scale change is required. Using the present invention, chord changes for an entire album could reside in less than 100K bytes of memory. This not only reduces cost to a practical level but at the same time allows the listener to provide creative accompaniment to the pre-recorded music. As the CD plays, the chord changes appear as MIDI patch changes at the moment the CD accesses the appropriate address during its play cycle.

As appears hereinafter an embodiment of the present invention includes a set of force sensitive transducers, arranged, for example as a keyboard. The keyboard is electronically scanned and the identity of the key or keys being depressed, along with information relating to the velocity of that key depression, are stored. Stored tables in memory convert that information to MIDI standard information relating to pitch and velocity for transmission to MIDI compatible tone generators or to MIDI messages for any MIDI event. The tables may be standard tables employing chord voicing information, individual note information or other information relating to the MIDI events to be implemented. The tables switch in real time at a speed sufficient to seem transparent to the user, thus allowing dynamic reconfiguration of the keyboard during the performance of the musical composition. In a presently-preferred embodiment, the tables are arranged such that during the interval of time in which a particular chord is being played, the depression of any key will result in the generation of a "correct" note in that chord or a "correct" note in a scale which is compatible with that chord. It is thus impossible for the musician to strike a wrong note. In addition, the keyboard may be operated at 100% efficiency because the keys may be defined such that they are all utilizable at any time during the performance of the musical composition. This affords the musician the widest possible choice of correct notes and chords at any point in the performance.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a presently-preferred embodiment of the invention.

FIG. 2 is a schematic drawing of the data acquisition apparatus of a presently preferred embodiment of the invention, including the force sensing resistors and signal conditioning circuitry.

FIG. 3A is a flow diagram for the main loop executed by the software for the present invention.

FIG. 3B is a flow diagram for the output loop executed by the software for the present invention.

FIG. 4 is a presently-preferred embodiment of a layout of a force sensing resistor keyboard for use in the present invention.

FIGS. 5a-f are a flow diagram for the mapping software useful for the present invention.

FIGS. 6a-j illustrate screen contents when using a computer for editing tables.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

In a presently preferred embodiment, the MIDI standard (Musical Instrument Digital Interface) is utilized to define which note is to be played and the volume (velocity) at which that note is to be played. As those of ordinary skill in the art will appreciate, the MIDI standard allows for both note pitch and note velocity (volume) information to be transmitted to a tone generator. The MIDI standard is well known and the MIDI Specification 1.0 is hereby incorporated by reference.

Referring first to FIG. 1, a block diagram of the musical instrument system 10 of the present invention, an array of input switching devices comprising force sensing transducers 12 is used as the interface between the musician and the instrument. In its most common form, force sensitive transducer array 12 may be configured as a keyboard, having the appearance of a keyboard of a conventional musical instrument, including both white keys and black keys. Other keyboard arrangements, such as that shown in FIG. 4, may be used. Those of ordinary skill in the art will recognize that, in accordance with the present invention, the human interface may also be configured to resemble a guitar neck, a series of percussion pad inputs, or the like. For the purpose of simplicity, reference will be made herein to keys as if a keyboard is being discussed, but those of ordinary skill will realize that no limitation is intended by such usage.

Presently preferred force sensing transducers for the present invention are force sensing resistors such as those manufactured by Interlink Electronics of Santa Barbara, California. Those of ordinary skill in the art will recognize, however, that other input switching devices may be used, such as those commonly found on presently available electronic keyboard instruments and the like.

In the block diagram of FIG. 1, there are n+1 force sensing resistors, having outputs on lines 14_0 through 14_n. These output lines 14_0-14_n are connected to signal conditioning circuits 16. The function of signal conditioning circuits 16 is to convert the output of each force sensing resistor in the array 12 to a DC voltage signal having a voltage range which can be utilized by the rest of the system. The outputs from the signal conditioning circuits, shown on lines 18_0-18_n, are connected to multiplexer and analog to digital (A/D) converter circuit 20. The function of multiplexer and A/D circuit 20 is to select one of lines 18_0 through 18_n and connect it to an analog to digital converter which then converts the voltage appearing on that line to a multi-bit digital representation, as is well known in the art.

The operations of system 10 are controlled via microprocessor 22, which is connected to a data bus 24 and an address bus 26. The multi-bit digital output of the A/D converter portion of multiplexer and A/D converter 20 is connected to data bus 24. Address bus 26 is connected to the multiplexer and A/D converter circuit 20 in order to control the addressing of the multiplexer.

As is common in microprocessor-controlled circuits, a program for controlling the operation of microprocessor 22 is stored in program storage 28, which may be a read-only memory (ROM), a programmable read-only memory (PROM), or other similar means known in the art, such as EPROMs, EEPROMs, etc. Program storage 28 is connected to data bus 24 and address bus 26. In addition, random access memory 30 is also connected to data bus 24 and address bus 26.

Universal Asynchronous Receiver Transmitter (UART) 32 is also connected to data bus 24 and address bus 26. As is well understood by those of ordinary skill in the art, UART 32 is utilized to interface between the system 10 of the present invention and a series of one or more tone generators, which produce the musical sounds in response to the musician's manipulations of the keyboard containing the force sensing resistors.

A MIDI system exclusive message may be utilized via the UART for editing purposes. This message may originate from an external editing source, such as a computer, disclosed later herein, or a sequencer performing a systems exclusive data drive as is understood by those skilled in the art. MIDI patch change information is also communicated through this port.

Referring now to FIG. 2, a force sensing resistor 12_x is shown connected at one end to a source of positive voltage 50. A limiting resistor 52 is connected to the other end of the force sensing resistor 12 and at its other end to the non-inverting input of amplifier 54. Resistor 56 is shown connected between the output of operational amplifier 54 and its inverting input. Resistor 58 is connected between the output of operational amplifier 54 and ground. Resistor 60 is connected between the inverting input of amplifier 54 and ground. Resistor 62 is connected between the non-inverting input of operational amplifier 54 and ground. The node comprising the bottom end of limiting resistor 52 and the non-inverting input of operational amplifier 54 is one of the lines 14 shown in FIG. 1. The output of operational amplifier 54 is one of the lines 18 shown in FIG. 1. In a presently preferred embodiment, amplifier 54 may be an LM324 operational amplifier and resistors 52, 56, 58, 60 and 62 may be 10 kOhms.

The output of the circuit of FIG. 2 is a DC voltage between approximately 0 volts and 4 volts with a power supply voltage of 5 volts. When no pressure is applied to the force sensing resistor, its resistance may be greater than approx-

imately 2 megaOhms. When a reasonable finger pressure is applied to force sensing resistor 12_x, its resistance will decrease to a value in the neighborhood of 5 kOhms, and a fingertip impulse to the force-sensing resistor can drive its resistance down to as low as 2 to 3 kOhms or lower.

Those of ordinary skill in the art will recognize that as the number of keys increases, the total scan time necessary to read and digitize the outputs of operational amplifiers 54 will increase. At some large number of keys the time will become large enough to affect performance and require higher speed performance components. To avoid degraded performance and to avoid the need to use higher speed performance hardware, it is presently preferred to modularize the hardware into blocks handling sixteen keys. These modules may be interfaced to one another to configure systems of larger size incrementally by groups of sixteen keys.

In such an embodiment, an ADC0816 sixteen channel multiplexer and 8-bit A/D converter, manufactured by National Semiconductor of Santa Clara, California, may be utilized. An 8032 microprocessor, manufactured by Intel Corporation of Santa Clara, California, is satisfactory to drive a modular system handling sixteen keys. In such a modular embodiment, a program storage capacity of 32k is sufficient. The tables necessary for operation of the present invention may also be stored in ROM. A 32k dynamic random access memory is satisfactory for use in this preferred modular embodiment. Those of ordinary skill in the art will readily recognize that the modularity disclosed herein, while presently preferred, is not to be taken as in any way limiting the scope of the present invention.

The hardware of FIGS. 1 and 2 is driven by a software program. In a presently preferred embodiment the software includes one main loop and two interrupt-driven tasks.

Referring now to FIG. 3a, the main loop of a presently preferred software routine for use with the present invention is shown. First, at step 100, the hardware of the system is initialized and all flags are reset. The initialization process includes the scanning of all of the force sensing resistors for the purpose of determining a noise margin. The output of the A/D converter representing the output from each force sensing resistor circuit is stored and examined and a threshold, higher than the highest voltage reading, is set.

After hardware initialization, the software enters the main loop which reads the output of a timer at step 102. When the time out value has been reached the program determines which of two loops it is in at step 104. There are two loops because, in a presently preferred embodiment, the DC voltage output of each force sensing transducer driven operational amplifier 54 corresponding to a key on the keyboard is read twice. If it is determined at step 104 that the software is in the first loop, the DC voltage outputs of all of the operational amplifiers 54 are read and saved in memory at step 106. Next, at step 108, the loop 2 flag is set. The program then returns to step 102.

If, however, it has been determined that the software is in the second loop at step 104, the program again reads all of the DC voltages at the outputs of operational amplifiers at step 110. The DC voltage value read during execution of the second loop for each key on the keyboard is compared with the previously-stored DC voltage value for that key from the first loop, and the larger of the two values is selected. Next, at step 112, the loop 2 flag is reset. The software then proceeds to an output loop.

The output loop of the presently preferred embodiment is shown at FIG. 3b. First, at step 114, it is determined whether data from all keys on the keyboard have been processed. If so, the software exits from the output loop. If not, at step 116 it is determined whether the stored digitized DC voltage value for the next key on the keyboard is above the threshold determined during the initialization routine. If it is, the note-on flag for that particular key is read to see if it is set. If it is set, the program returns to step 114. If it has not been set, the note-on flag for that particular key is set at step 120. Next, at step 122 the software refers to a note value table to define the note. In the presently preferred embodiment, the note's definition will be a MIDI code. Those of ordinary skill in the art will realize that this note-on signal may be designated for any MIDI channel.

The velocity information relating to the note is also determined by reference to a table, which converts the raw digitized DC voltage value associated with each key on the keyboard to a MIDI velocity code. Those of ordinary skill in the art will recognize that use of such a table allows for expansion, compression or other volume level manipulation. At step 124, an address derived from the raw digitized DC value is used to address a velocity table to obtain a MIDI velocity value. Next, at step 126, a MIDI note-on message is sent with the calculated velocity. The program then returns to step 114.

If, at step 116, it is determined that the DC voltage value corresponding to the key on the keyboard is below threshold, at step 128 it is determined whether the note-on flag for that particular key has been reset. If it has, the program returns to step 114. If it has not, at step 130 the note-on flag for that particular key is reset. Next, at step 132 the note value table is again consulted to determine the MIDI code for the note that has been assigned to the key being depressed. Next, at step 132, a MIDI note-off message is sent with velocity = 0 and the program returns to step 114.

One of the features of the present invention which sets it apart from anything known in the prior art is the use of the tables which give the system the ability to assign any key to any note value or any other MIDI event. Table switching is performed in real time at a speed sufficient to render the process undetectable to the ear. The two primary types of tables are chord tables and scale tables.

Chord tables are normally accessed via a specified section of the keyboard (such as all black notes in a conventional keyboard.) These tables contain all possible notes within a given chord and may be assigned in any manner

desired. For example, a C major chord consists of the three notes C,E, and G. In ascending pitch, the notes might be assigned to a particular key or group of keys in any, including, but not limited to the following: E-G-C, E-C-G, C-E-G, C-E-G-C, etc., with the ability to assign various octaves and instrument voices.

Scale tables contain all notes within a given scale and are likewise accessed by a specified section of the keyboard. Chord tables and scale tables may be switched independently of one another either in real time by the user or via predetermined computer control such as via a sequencer or the like.

In one embodiment of the present invention, chord and scale information may be stored along with prerecorded music on musical media such as a CD disk and may be sent to the system of the present invention via a MIDI interface so that a musician can "play along" with prerecorded music. Since the chord change and any scale change timing is synchronously provided by the prerecorded media, the musician has creative input but does not have the option of playing an incorrect chord or note.

In this embodiment, the code necessary to implement chord changes requires only a fraction of the memory necessary to store melody and chord notes on a CD for playing along, thus making such an embodiment a practical reality. For instance, a typical popular music selection would require up to 500K bytes of information to reproduce the parts contained on the recording. A 10 song album could require 5M bytes or more of memory, and would not afford creative input by the listener. On the other hand, with the present invention, only one MIDI message per chord change or scale change is required. Using the present invention, chord changes for an entire album could reside in less than 100K bytes of memory. This not only reduces cost to a practical level, but at the same tame allows the listener to provide creative accompaniment to the prerecorded music. As the CD plays, the chord charges appear as MIDI patch changes at the moment the CD accesses the appropriate address during its play cycle.

The number of tables which may be associated with the system of the present invention is limited only by the size of the memory which is utilized with the system. For instance, with a memory size of 64K, scale tables and chord tables for sixteen of the most common chords for each root note for 128 different keyboard keys can be provided.

By the arrangement of and switching of the tables, the user is presented with an incredibly flexible musical instrument that contains all of the correct musical choices at any given point in time, but only those correct choices. Since only correct choices are presented, the musician is freed from the complex and sometimes tedious task of performing the mathematical calculations necessary to execute correct chord, melody and harmony structures.

Referring now to FIG. 4, a presently preferred layout for a keyboard for use as part of the present invention is shown. A first group of two sets of seven function keys, indicated by even reference numerals 202-228, preferably relate to chord tables. Depressing any one of keys 202-228 will result in the generation of MIDI codes representing a component note of a desired musical chord.

The section of keys just above keys 202-228 is arranged much like two octaves of a conventional keyboard. Keys bearing even reference numerals 230-256 are the white keys of the keyboard and keys bearing even reference numerals 258-276 are the black keys of the keyboard. Note that the particular layout permits playing black keys only by running a finger across the keyboard, since white keys do not extend all of the way between black keys.

The white keys 230-256 are assigned to individual notes from scale tables. However, unlike a conventional keyboard, keys 230-256 are all utilized in the playing of each scale. If the scale is arranged so that key 230 is always the root note, the keyboard may be arranged such that key 232 is always the second, key 234 is always the third, key 236 is always the fourth, key 238 is always the fifth, key 240 is always the sixth, key 242 is always the seventh, key 244 is octave and so on.

Those of ordinary skill in the art will recognize that this arrangement results in a 100% efficient keyboard in that all keys are proper notes and are used in any scale. Furthermore, the key/note assignments may be made such that the root note always resides at the memory location corresponding to key 230 so that playing a scale, major or minor, in any key is as simple as playing a C major scale on a conventional keyboard. This arrangement frees the musician from having to count notes and intervals and memorize musical keys and scales which require the use of the black keys on a conventional keyboard to implement sharps and flats. When compared with a conventional keyboard which has an efficiency of approximately from 25% to 60% from three note chords to a seven note scale, and which requires the musician to be ever mindful of flatted intervals and other peculiarities of certain musical chords, keys and scales, the power of and simplicity of use of the musical instrument of the present invention is readily discernible.

As examples of possible note/key assignments, Table 1 shows the note assignments to keys 230-256 for the C major, C# minor, D# major, and F minor scales respectively.

TABLE 1

	C Major	C# Minor	D# Major	F Minor
Key 230	C	C#	D#	F
Key 232	D	D#	F	G
Key 234	E	E	G	G#
Key 236	F	F#	G#	A#
Key 238	G	G#	A#	C
Key 240	A	A#	C	D
Key 242	B	B	D	D#
Key 244	C	C#	D#	F
Key 246	D	D#	F	G
Key 248	E	E	G	G#
Key 250	F	F#	G#	A#
Key 252	G	G#	A#	C
Key 254	A	A#	C	D
Key 256	B	B	D	D#

From the examples given in Table 1, those of ordinary skill in the art will easily configure the key/note map for any musical scale. In systems employing tone generators which are capable of outputting non-standard intervals, such as are found in certain Arabic and Oriental musical structures, key/note maps to implement these otherwise difficult musical systems are easily developed. It will additionally be noted from Table 1 that the playing of a scale in a different key is easily accomplished on the same keys which would produce the scale of C major on a conventional keyboard, thus illustrating the elimination of the need to constantly calculate sharps and flats when in keys other than C major.

The black keys, even reference numerals 258-276 may be configured as the notes which are components of selected chords. For example, Table 2 note assignments to keys 258-276 for a C major and D# minor chord respectively.

TABLE 2

	C Major	D# Minor
Key 258	C	D#
Key 260	E	F#
Key 262	G	A#
Key 264	C	D#
Key 266	E	F#
Key 268	G	A#
Key 270	C	D#
Key 272	E	F#
Key 274	G	A#
Key 276	C	D#

Since the two sets of seven horizontal keys, even reference numerals 202-228, are also assigned to chord component notes, MIDI note-on messages from keys 258-276 may be sent on a different MIDI channel to drive a voice different from that associated with keys 202-228.

Two sets of 16 vertical keys, even reference numerals 278-308 and even reference numerals 310-340 respectively may be used for numerous functions. In a presently preferred embodiment, keys with reference numerals 278-308 are used to cause MIDI program commands which will transform the rest of the unit to an entire window of corresponding scale and chord information and may also sound a chord if desired. Keys bearing reference numerals 310-340 may be configured to be a scale. Since white keys 230-256 are already configured as a scale, the two sets of scale keys can be used with different voices to create two scales of two different instruments. Those of ordinary skill in the art will readily recognize that the scales played on keys 230-256 and 310-340 respectively could even be different scales.

The two sets of three keys 342, 344 and 346, to the left of the double group of seven horizontal seven keys and 348, 350 and 352 to the right of the double group of seven horizontal keys may be used as MIDI control signals as positive pitch bend, negative pitch bend, modulation, etc.

The two sets of 15 keys above keys 230-276 may be used for any MIDI function. In a presently preferred embodiment they may be used for any of the computer controlled functions disclosed with respect to FIGS. 6a-i.

While the embodiment of FIG. 4 has been discussed in terms of specific key functions, those of ordinary skill in the art will readily recognize that any key may be assigned any MIDI function and that the embodiment of FIG. 4 is merely a practical illustrative and presently preferred arrangement.

The set of 16 vertical keys shown at even reference numerals 278-308, may be configured to cause MIDI program commands which will change the chord configured on keys 202-228 and 258-276. Depressing these program change keys can optionally sound the chord which they select. In this manner, the musician may play a song and with one finger redefine the chord keys at the appropriate times so that the song may be played without the possibility of striking an incorrect note in a chord. Optionally one or more of these keys may also cause one or both banks of scale keys 230-256 and 310-340 to define a different scale if it is desired.

Another computer, either integral with the system of FIG. 1, or an external computer may be used as a mapping tool to manipulate other MIDI compatible musical instruments as well as the musical instrument of the present invention. One computer which has been found to be particularly suitable for use with the present invention is the Atari 1040 ST computer, which comes with a built-in MIDI interface.

FIGS. 5a-f show a flow diagram for the mapping software in a presently preferred embodiment. Referring first to FIG. 5a, the main loop begins at step 400 where all the tables and indices are initialized as is well understood by those of ordinary skill in the art. Next, at step 402 it is determined whether a MIDI byte has been received. If a MIDI byte has been received, the program proceeds to the MIDI processing loop disclosed with respect to FIG. 5b. If not, at step 404 a determination is made whether one of the mouse buttons has been clicked. If not, the loop returns to step 402. If a mouse button has been clicked, the program proceeds to block 406 where the tables and indices are edited by user interface. After the user has edited the desired tables, a determination is made at step 408 whether it is desire to quit the program. If so, the program is ended and if not, it returns to step 402.

Referring now to FIG. 5b, the MIDI processing routine is disclosed. First, at step 410 the received MIDI bytes are assembled into MIDI events. Next, at step 412, it is determined whether a complete MIDI event has been assembled. If not, the program returns to the main processing loop. If, however, a complete MIDI event has been assembled, if the event is a note-off, at step 414 a pseudo note-off command is changed to a real note-off command. Then, a determination is made at step 416 whether the events channel matches either the upper or the lower bank. If not, at step 420 the event is transmitted to the other MIDI units in the system and then at step 421 a determination is made regarding whether the program is in zoom mode. If not, the program returns to the main processing loop. If so, the program returns to zoom processing.

If, at step 416 the events channel has matched one of the two banks, a determination of what kind of MIDI event has been assembled is made at step 418. If it is a note-on event, the program proceeds to note-on processing described with respect to FIG. 5c. If the event is a note-off event, the program proceeds to note-off processing described with respect to FIG. 5d. If the event is a patch change, the program proceeds to patch change processing described with respect to FIG. 5e.

Referring now to FIG. 5c, the note-on processing routine begins at step 424 where the computation of what value this note is mapped into is made. At step 426, the value is examined to see if it is less than zero. If the note is mapped into a value of less than zero, it indicates zoom processing and the program proceeds to zoom processing as disclosed with respect to FIG. 5f.

If, however, the note has a mapped value of greater than zero at step 428 the events channel may optionally be changed if desired. At step 430 the mapped value for the incoming note number is substituted for the incoming note number. Next, at step 432 this map value is stored in a table which indicates the note and channel on which it came in and the note and channel on which it went out. This table is used later to identify the note to be turned off in the event of an intervening patch change. Next, at step 434 the mapped note-on event is transmitted over the MIDI channel.

Referring to FIG. 5d, the note-off processing routine is disclosed. First, at step 436, the note and channel out and note and channel in information are retrieved from the table in which they were stored. Next, the mapped value of the stored note to be turned off is substituted for the incoming note number at step 438. At step 440, the table channel is substituted for the event's channel and at step 442 the mapped note-off event is transmitted.

Referring to FIG. 5e the patch change processing routine is described. First, at step 444 it is determined to which bank the patch change refers. If there is no match, the program returns to the main processing loop. If there is a match, at step 446 it is determined whether the current bank number is the same as the current channel number. If it is, at step 448 the channel indices are updated. Step 450, is performed after step 446 if there is no match between the channel number and the bank number, and after step 448 if there has been a match. After step 450, the patch change MIDI event is transmitted at 452.

Referring now FIG. 5f, zoom processing begins at step 454 where a zoom index is computed from the current map number. Next, at step 456 the zoom index and the map index are swapped. Next, at step 458, a loop is performed relating to the zoom depth. If the count is not completed, at step 462 a MIDI event is built from the map index and the loop. Next, a step 464, the event is broken into bytes and the program proceeds to MIDI processing according to FIG. 5b. When the loop for zoom depth has been completed, the zoom index and map index are swapped in step 460 and the program returns to the main loop.

Since in the MIDI standard, there are 128 possible notes, the tables which are used with the present invention may be conveniently divided into 256 eight-bit bytes. The first set of 128 eight bit bytes define the 128 possible MIDI notes. The second 128 eight bit bytes define the MIDI channels over which the notes will be transmitted.

The tables are switched by a dynamic table allocation process. The tables are arranged in two banks of 128 tables each. Each table has 128 bytes. Each location in a table may hold a value of indicating one of 128 possible MIDI notes. A MIDI note which comes into the UART is directed to either the upper or the lower bank of tables depending on the channel number assigned to that incoming MIDI note. Which table in the bank is selected by the position of a mouse used in conjunction with the computer. Alternatively, the table can be selected by a MIDI patch change over a MIDI channel reserved for patch changes. The value of the incoming note (between zero and 127) determines the address to look at within the table. The contents of the table gives the note and channel number to be transmitted.

Note-off information, on the other hand, may not be related to the table from which the note-on information was obtained because of the possibility that a patch change will change the table to be referenced before that particular key on the keyboard is released. To avoid the problem of stuck notes, a second table, transparent to the user, is used to enter the note-on information. When the transducer circuitry senses that a key has been released, the system looks to this user transparent table to determine which note to turn off to avoid errors due to patch changes.

The previously described zoom function is a powerful function which allows the musician greatly enhanced flexibility when composing and playing compositions. It allows a single pad to play many notes, as in a chord, in place of a single note, and further optionally allows patch change information to be sent to co-ordinate chord changes among a plurality of MIDI instruments.

In order to better understand the zoom function, FIGS. 6a - H, show what the computer screen will show at various points in the zoom process.

In FIG. 6a, two tables are shown. The upper table is from the first bank of tables and the lower table is from the second bank of tables both previously described. In a presently preferred embodiment, to conserve memory space, the upper and lower bank of tables each contain 16 tables which are zoomable. These tables are found as the last two columns of eight entries in each of the upper and lower banks. Each table of structures contains 128 structures. Each structure has six bytes. The first byte defines which of the 256 table of both the upper and lower bank to address. The second byte contains a start address from zero to 127 within that table. The third byte contains two nibbles. The high nibble contains an all/white/black mask which allows either all keys, white keys only or black keys only to be selected. The low nibble decides how deep to zoom. The depth of the zoom is the number of notes in an upward direction from the start note. The fourth byte may contain an optional patch change which may be sent to other devices. The fifth byte contains information defining a channel for the patch change to sent over. Byte six is currently reserved for a function to be defined later. The zoom function is enabled as follows. Normally, the content of the note tables will be a note number. However, if the contents of the note table is minus one, a zoom table instead of a note table is referred to.

FIGS. 6a-j, illustrate the use of a computer to perform editing on the tables of the present invention. FIGS. 6a-j are printouts showing the screen configurations of a computer at various steps in the editing process.

Referring first to FIG. 6a, the screen shows an upper matrix of 16 x 8 table positions and a lower matrix of 16 x 8 table positions. Note that in the upper matrix, the chord B minor in the second row of the twelfth column appears in reverse video, having been selected by a mouse. Likewise, in the bottom of matrix, the chord D in the eighth row of the fifteenth column has been selected. In particular configuration, the sixteen zoomable tables have been located in the last two columns of both the upper and lower matrices. Thus, the selection of B minor in the upper matrix is not the selection of a zoomable table, but the selection of the D chord in the lower matrix is from a zoomable table.

Referring now to FIG. 6b, the table for a B minor chord has been brought up. Note that in the far left-hand column, outside of the rectangle, a list of the 12 chromatic scale notes, beginning with C and ending with B, represents the key positions on a conventional keyboard corresponding to those notes. In the first row of the table outside of the rectangle, the numbers -2 through 8 signify the octaves spanning by MIDI. Within the rectangle, there are 128 entries, corresponding to the 128 possible keys of a keyboard addressed by the invention. Note that the screen contains three print styles, normal, bold, and reverse video as will be readily recognizable by those of ordinary skill in the art. In the fields below

the rectangle, the indication "notes" has been selected by mouse, and thus appears in reverse video, indicating that this is a note table. It will be recognized that the notes which appear in bold representation on screen indeed represent the notes from an extended B minor scale. The notes appearing in normal video are unselected. It will be noted that seven notes in the third octave in the +3 octave column, seven notes in the +4 octave, and two notes in the +5 column have been displayed in reverse video. These 16 notes have been selected and assigned to keyboard keys by placing the mouse their locations and engaging the mouse button or by selection through MIDI input.

It should be understood that for the purposes of all of FIGS. 6a-j, all 128 positions in the rectangle are always active and any one or group of these positions may be simultaneously selected for manipulation by the edit screen, for example for the purpose of downloading a group of 16 keyboard keys in a modular unit. This may be accomplished through a systems exclusive MIDI message as a single table, group of tables, or even by individual notes, channels or other MIDI events.

It will also be noted that in the field under the rectangle column, the indication "white" has been selected by the mouse and thus is shown in reverse video, indicating that white keys from the conventional keyboard notation in the first column have been selected. Thus, the 16 notes of the B minor scale shown will be played only when corresponding MIDI values, relating to white keys designated in the column to the left of the rectangle, are received in the appropriate octave designated by the note's position in the rectangle.

Referring now to FIG. 6c, a third screen is shown, differing from the second screen in that the indication "black" has been selected by the mouse in the field under the rectangle. The 16 notes in the reverse video within the rectangular field have been selected by the mouse and correspond to the black key notations in the first column outside the rectangle. Those of ordinary skill in art will recognize that the selected notes are all contained within a B minor chord. This examples of white and black notes are not intended to indicate any limitation on the intermingling of white and black notes for any manipulation, as shown by the availability of the choice "all" in the "white/black/all" field under the rectangle.

Referring now to FIG. 6d, the indication "channels" has been selected by the mouse and appears in reverse video, indicating that the portion of the tables dealing with the channels over which the notes are to be sent has been accessed. The screen shown in FIG. 6d corresponds to the screen shown in FIG. 6c, the reverse video images showing the channels over which the notes comprising the B minor chord shown in FIG. 6c are to be sent. Those of ordinary skill in the art will recognize that any one of the 16 MIDI channels could be selected for any of these notes, thus allowing a single keyboard to play any chord or scale in one or more of several voices.

FIG. 6e is included to show that any randomly chosen notes can be assigned to the selected black keys. Although not shown in FIG. 6e, the same is true for the white keys, which may have assigned to them any random note or other MIDI event.

The zoom functioning of the present invention is shown with respect to FIGS. 6f-j.

The lower matrix of FIG. 6a had the D in the last row of column 15 selected. The screen shown in FIG. 6f is brought up to edit the zoom function. The event which equals the position in the matrix i.e., C# in the -2 octave (C# -2), will cause anything selected in the zoom edit page shown in FIG. 6g to be output, including patch change, note information or other MIDI events. It will be noted that the first column within the rectangle of the screen of FIG. 6f contains chord information. In bold video the chords G, F, E minor, and C have been selected and are highlighted because the filters allowing zoom only are active, indicated by the indication "zoom" in reverse video. It should be understood that any one of the 128 positions within the rectangle on the screen of FIG. 6f are zoomable. The A minor chord is shown in reverse video to indicate that it has been selected.

Referring now to FIG. 6g, the reverse video indications of "notes" and "black" show that the notes of A minor chord indicated by the five reverse video notes in the rectangular field have been selected to be played when the MIDI values corresponding to the black key (C# -2 as selected in FIG. 6f) has been received. This is indicated at the top of FIG. 6g.

In FIG. 6g, the A minor chord has been composed of the five notes shown in reverse video and will play. Anytime that the key indicated at the top of this edit screen is depressed, when its host edit page (here FIG. 6f) has been selected, whatever is selected in the zoom rectangle will be output as indicated by the reverse video indication "black" in the field below the rectangle. The "depth" of "05" appearing in the field under the rectangle indicates how many notes are to be played in the chord and/or scale. This number is user selectable. The information "patch 026 16" in the field under the rectangle are user selectable and indicate that a MIDI patch chance 026 will be sent out on channel 16. The MIDI patch numbers are shown in FIG. 6h. Comparing the position 026 in FIG. 6h to the corresponding position in FIG. 6a confirms that the patch relates to the A minor chord.

Referring now to FIG. 6i, the definitions of the patch changes are defined by the user. FIG. 6i shows that patch changes for the lower matrix are transmitted on channel 16 and patch changes for the upper matrices are received on channel 16. Also shown in FIG. 6i is the transpose function, allowing a global transpose relative to the note C-3. If the note identifiers appearing in the upper and lower boxes are equal to C-3 no transposing will take place. Otherwise, all notes will be transposed up or down by the difference between the note C-3 and the contents of the upper and lower transpose boxes, allowing exploration of various keys without the need to reconfigure the tables being utilized.

FIG. 6j illustrates the map filling function which allows filling the upper and lower matrices automatically. The

reverse video indications show that the upper matrix from positions 18 to 32 in the matrix are to be filled with the table at MIDI number 17 in the upper matrix. It further indicates that the successive positions in the matrix are incremented by half steps and are displayed by names. Both chord and scale tables are changed; the selection of "all" "white", or "black" allows selection of chords, scales or both. Also, the name itself may be automatically transposed, thus avoiding the need to manually enter a new name in each corresponding table which has been transposed. Likewise, channel information may be selected so that a voicing arrangement may be placed on pre-existing tables without altering their note values. Alternatively, both note and channel information may be altered by selecting "both".

While a presently preferred embodiment of the invention has been disclosed, those of ordinary skill in the art will, from an examination of the within disclosure and drawings be able to configure other embodiments of the invention. These other embodiments are intended to fall within the scope of the present invention which is to be limited only by the scope of the appended claims.

Claims

1. An electronic musical controller comprising:

   a plurality of input device signals;
   a first memory for storing a plurality of different translation tables, said translation tables relating said input device signals with corresponding control signals;
   a source of translation table selection signals; and
   a translator for outputting said control signals responsive to said input device signals, said translation tables and said translation table selection signals.

2. An electronic musical controller according to claim 1, further comprising an electronic signal generator responsive to said control signals.

3. An electronic musical controller according to claim 1, wherein said source of translation table selection signals is a manually operated device.

4. An electronic musical controller according to claim 1, wherein said source of translation table selection signals includes a second memory for outputting said translation table selection signals in synchronization with musical events in a musical performance.

5. An electronic musical controller according to claim 4, wherein said musical performance is a live performance.

6. An electronic musical controller according to claim 4, further comprising a third memory in which said musical performance is prerecorded.

7. An electronic musical controller according to claim 1, wherein said source includes a second memory for storing translation table selection signals time-indexed to musical events in a musical performance.

8. An electronic musical controller according to claim 7, further comprising a third memory for storing musical signals representative of music containing musical events.

9. An electronic musical controller according to claim 7 further comprising:

   a first sound reproduction device responsive to said musical signals for generating a first audio signal representative of said music, said first audio signal having at any given moment a predetermined chord and scale,
   wherein said electronic signal generator generates a second audio signal responsive to said input device signals, notes of said second audio signal at each said given moment being within said predetermined chord and scale.

10. An electronic musical controller according to claim 1, wherein said source includes a second memory for storing translation table selection signals time-indexed to musical events in a musical performance; and further comprising:

    a third memory for storing musical signals representative of music containing musical events, said second and third memories synchronized so as to output said translation table selection signals responsive to said musical events.

11. An electronic musical controller according to claims 4, 5, 6, 7, or 10 wherein said musical events include chord changes.

12. An electronic musical controller according to claims 4, 5, 6, 7, or 10 wherein said musical events include scale changes.

13. An electronic musical controller according to claims 4, 5, 6, 7, or 10 wherein said musical events include chord and scale changes.

14. An electronic musical controller according to claim 13 wherein said control signals include CHORD signals specifying a chord and SCALE signals specifying a scale.

15. An electronic musical controller according to claim 14 wherein said electronic signal generator, responsive to said control signals, generates signals within a chord identified by a most recently received CHORD signal and within a scale identified by a most recently received SCALE signal in response to said input device signals.

16. An electronic musical controller according to claim 14 wherein said means for generating a plurality of input device signals includes a plurality of force sensitive keys and wherein said electronic signal generator, responsive to said CHORD and SCALE signals, will generate signals within a chord and scale, respectively, identified by a most recently received pair of CHORD and SCALE signals in response to depressions of said plurality of force sensitive keys.

17. An electronic musical controller according to claim 1, further comprising a source of a musical signal, said musical signal containing a plurality of phrases, each said phrase having a chord and a scale associated therewith.

18. An electronic musical controller according to claim 17 wherein said source of translation table selection signals includes a sequence of CHORD and SCALE signals, said CHORD and SCALE signals identifying said chords and said scales associated with said phrases of said musical signal.

19. An electronic musical controller according to claim 18 wherein said translator further comprises:

a CHORD table memory for retrieval of one of a plurality of CHORD tables;
a SCALE table memory for retrieval of one of a plurality of SCALE tables;
a table memory loader for receiving said CHORD tables and said SCALE tables from an external source and storing said CHORD tables and said SCALE tables in said CHORD table memory and said SCALE table memory, respectively;
said translator responsive to said CHORD and SCALE signals for selecting one of said CHORD tables to be an ACTIVE CHORD TABLE and one of said SCALE tables to be an ACTIVE SCALE TABLE during the playing of said musical signal; and wherein
said translator further responsive to said input device signals, and said electronic signal generator limited to generating signals within said ACTIVE CHORD TABLE and said ACTIVE SCALE TABLE.

20. An electronic musical instrument according to claims 17, 18, or 19 wherein said source of a musical signal is a compact disc.

21. An electronic musical instrument according to claim 17 further comprising:

a mixer responsive to said source of a musical signal and to said electronic signal generator for generating and amplifying a mixed audio signal.

22. A method of operating an electronic musical controller in a play-along mode with a pre-recorded sound recording, and a pre-recorded data recording, said data recording containing translation table selection signals synchronized with events in said sound recording, each said translation table corresponding to a particular chord and a particular scale, said method comprising:

storing a plurality of different translation tables, each of said translation tables corresponding to a particular chord and a particular scale;
playing the sound recording and the data recording simultaneously to form at least a translation table selection signal output containing translation table selection signals;

forming a stream of first input signals;

receiving said translation table selection signals from said translation table selection signal output; and

translating said first input signals into a stream of control signals for controlling an output of the electronic musical controller by applying said first input signals to a translation table most recently selected by a most recently received translation table selection signal so that said control signals cause the electronic musical controller to output music signals in response to said first input signals which music signals are only within said particular chord and said particular scale corresponding to said translation table selected by said most recently received translation table selection signal.

23. A method according to claim 22, wherein said stream of first input signals is formed by manually activating a plurality of force sensitive devices.

24. A method for the operation of an electronic musical controller comprising the steps of:

storing a plurality of different translation tables in a first memory, said translation tables relating a plurality of input device signals with corresponding control signals;

providing translation table selection signals; and

outputting said control signals responsive to said input device signals, said translation tables and said translation table selection signals.

25. A method according to claim 24, further comprising the steps of generating signals responsive to said control signals.

26. A method according to claim 24, wherein said step of providing includes a step of manually operating a device to provide said translation table selection signals.

27. A method according to claim 24, wherein said step of providing further includes a step for outputting said translation table selection signals in synchronization with musical events in a musical performance.

28. A method according to claim 27, wherein said step of outputting includes a step of performing live said musical performance.

29. A method according to claim 27, further comprising a step of providing said musical performance in a prerecorded form.

30. A method according to claim 24, wherein said step of providing includes a step of storing said translation table selection signals time-indexed to musical events in a musical performance.

31. A method according to claim 30, further comprising a step of storing musical signals representative of music containing musical events.

32. A method according to claim 30, further comprising the steps of:

generating a first audio signal representative of said music, said first audio signal having at any given moment a predetermined chord and scale; and

wherein said step of generating signals generates a second audio signal responsive to said input device signals, notes of said second audio signal at each said given moment being within said predetermined chord and scale.

33. A method according to claim 24, wherein said step of providing includes a step of storing translation table selection signals time-indexed to musical events in a musical performance in a second memory; and further comprising a step of:

storing musical signals representative of music containing musical events in a third memory, synchronizing said second and third memories so as to output said translation table selection signals responsive to said musical events.

34. A method according to claims 27, 28, 29, 30, or 33 wherein said musical events include a step of changing chords.

35. A method according to claims 27, 28, 29, 30, or 33 wherein said musical events include a step of changing scales.

36. A method according to claims 27, 28, 29, 30, or 33 wherein said musical events include a step of changing scales and a step of changing chords.

37. A method according to claim 36, wherein said control signals include CHORD signals specifying a chord and SCALE signals specifying a scale.

38. A method according to claim 37, wherein said step of generating signals is responsive to said control signals and generates said signals within a chord identified by a most recently received CHORD signal and within a scale identified by a most recently received SCALE signal in response to said input device signals.

39. A method according to claim 37, wherein said step of generating a plurality of input device signals includes operating a plurality of force sensitive keys and wherein said step of generating signals is responsive to said CHORD and SCALE signals, and generates signals within a chord and scale, respectively, identified by a most recently received pair of CHORD and SCALE signals in response to said step of operating said plurality of force sensitive keys.

40. A method according to claim 24, further comprising the step of providing a musical signal, said musical signal containing a plurality of phrases, each said phrase having a chord and a scale associated therewith.

41. A method to claim 40, wherein said step of providing includes providing a sequence of CHORD and SCALE signals, said CHORD and SCALE signals identifying said chords and said scales associated with said phrases of said musical signal.

42. A method according to claim 41, wherein said step of outputting further includes the steps of:

retrieving one of a plurality of CHORD tables from a CHORD table memory;
retrieving one of a plurality of SCALE tables from a SCALE table memory;
receiving said CHORD tables and said SCALE tables from an external source and storing said CHORD tables and said SCALE tables in said CHORD table memory and said SCALE table memory, respectively;
selecting one of said CHORD tables to be an ACTIVE CHORD TABLE and one of said SCALE tables to be an ACTIVE SCALE TABLE during the playing of said musical signal; and wherein
said step of selecting responsive to said input device signals, and said step of generating signals generates said signals within said ACTIVE CHORD TABLE and said ACTIVE SCALE TABLE.

43. An electronic musical instrument according to claims 40, 41, 42 wherein said step of providing a musical signal includes a compact disc.

44. A method according to claim 40, further comprising the steps of:

mixing said musical signal and said audio signal generated by said step of generating signals, resulting in a mixed signal; and
amplifying said mixed signal.

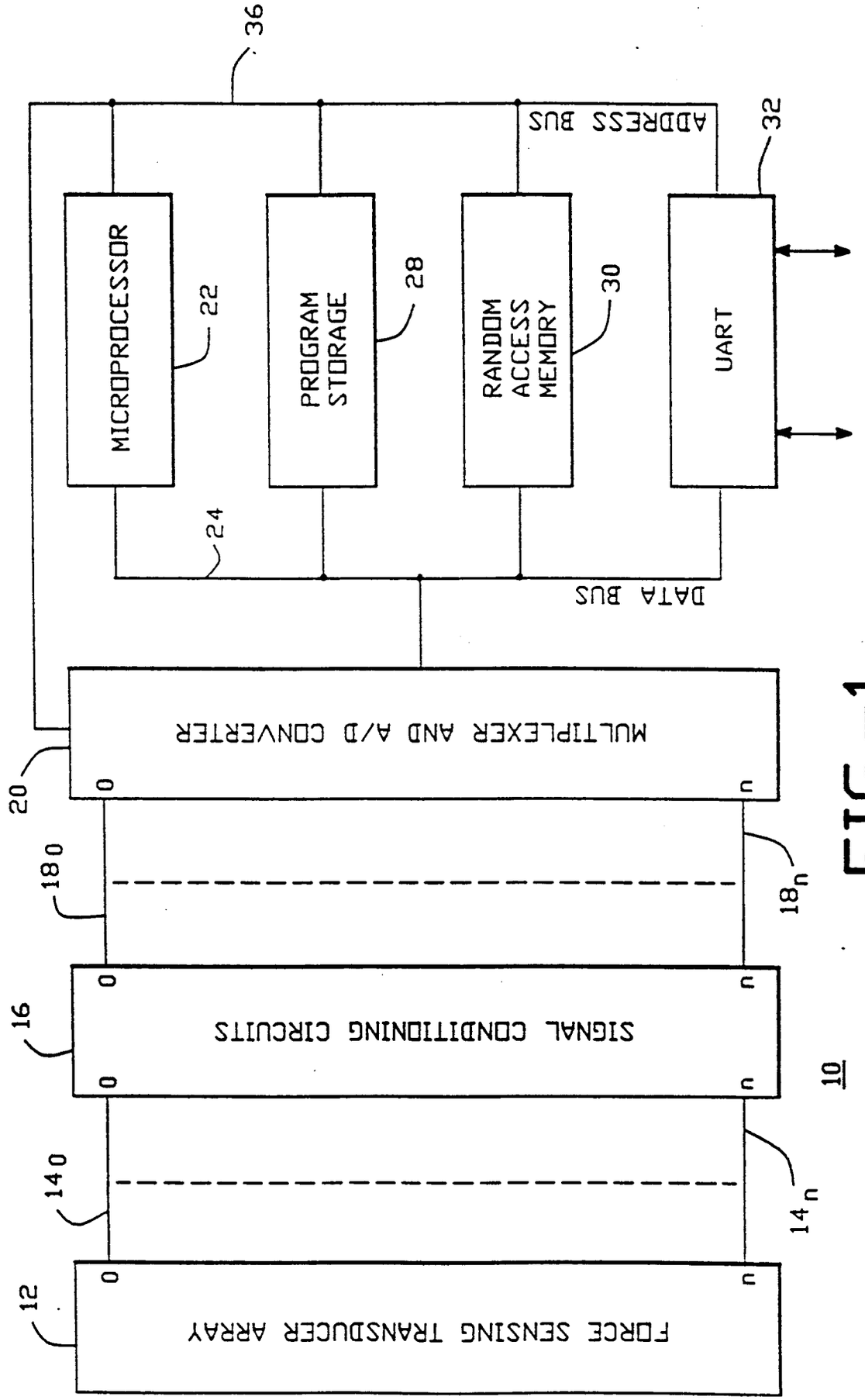

FORCE SENSING TRANSDUCER ARRAY
12
0
n
14 0
14 n
SIGNAL CONDITIONING CIRCUITS
16
18 0
18 n
MULTIPLEXER AND A/D CONVERTER
20
DATA BUS
24
MICROPROCESSOR
22
PROGRAM STORAGE
28
RANDOM ACCESS MEMORY
30
UART
32
ADDRESS BUS
36
10

FIG.-1

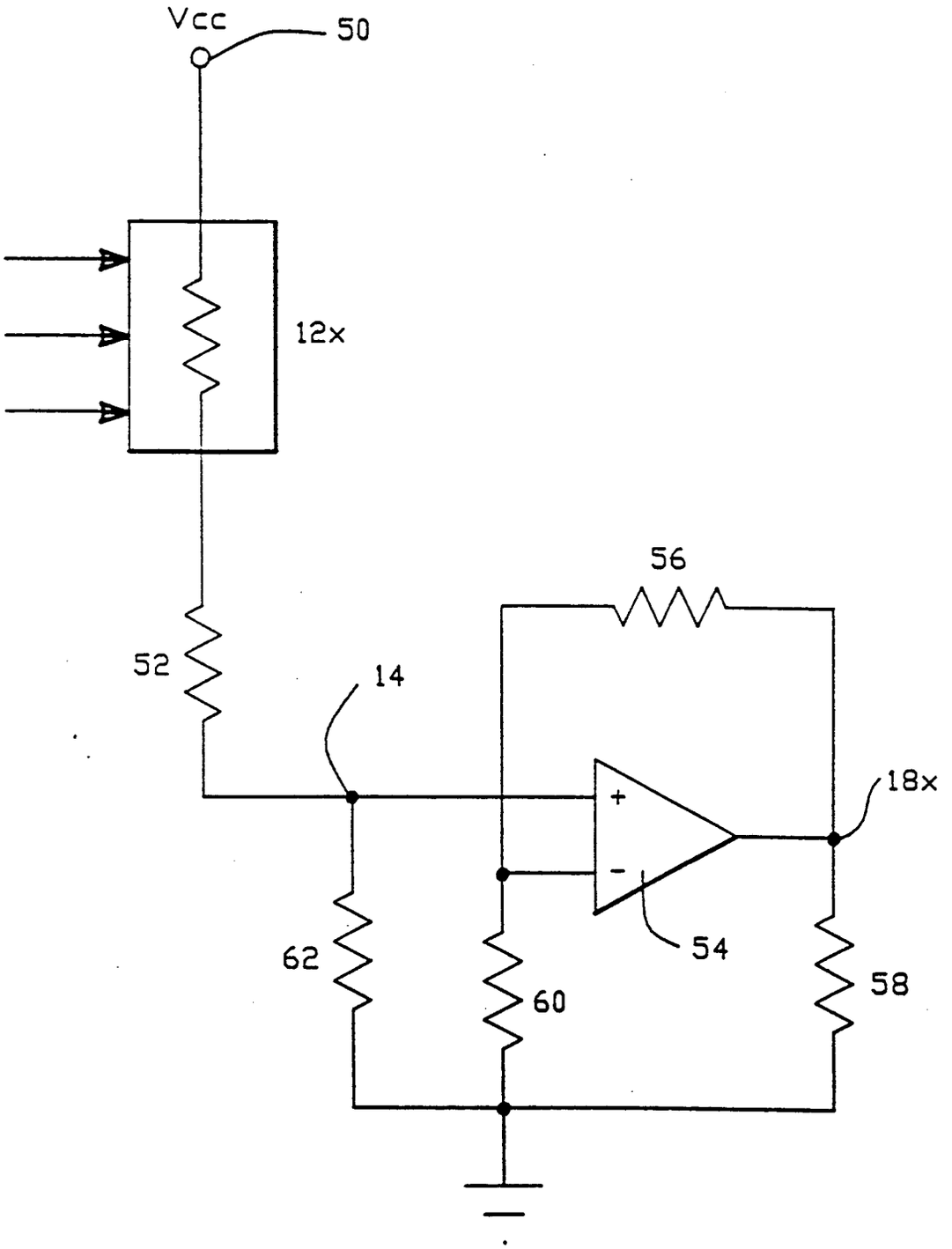


FIG.-2

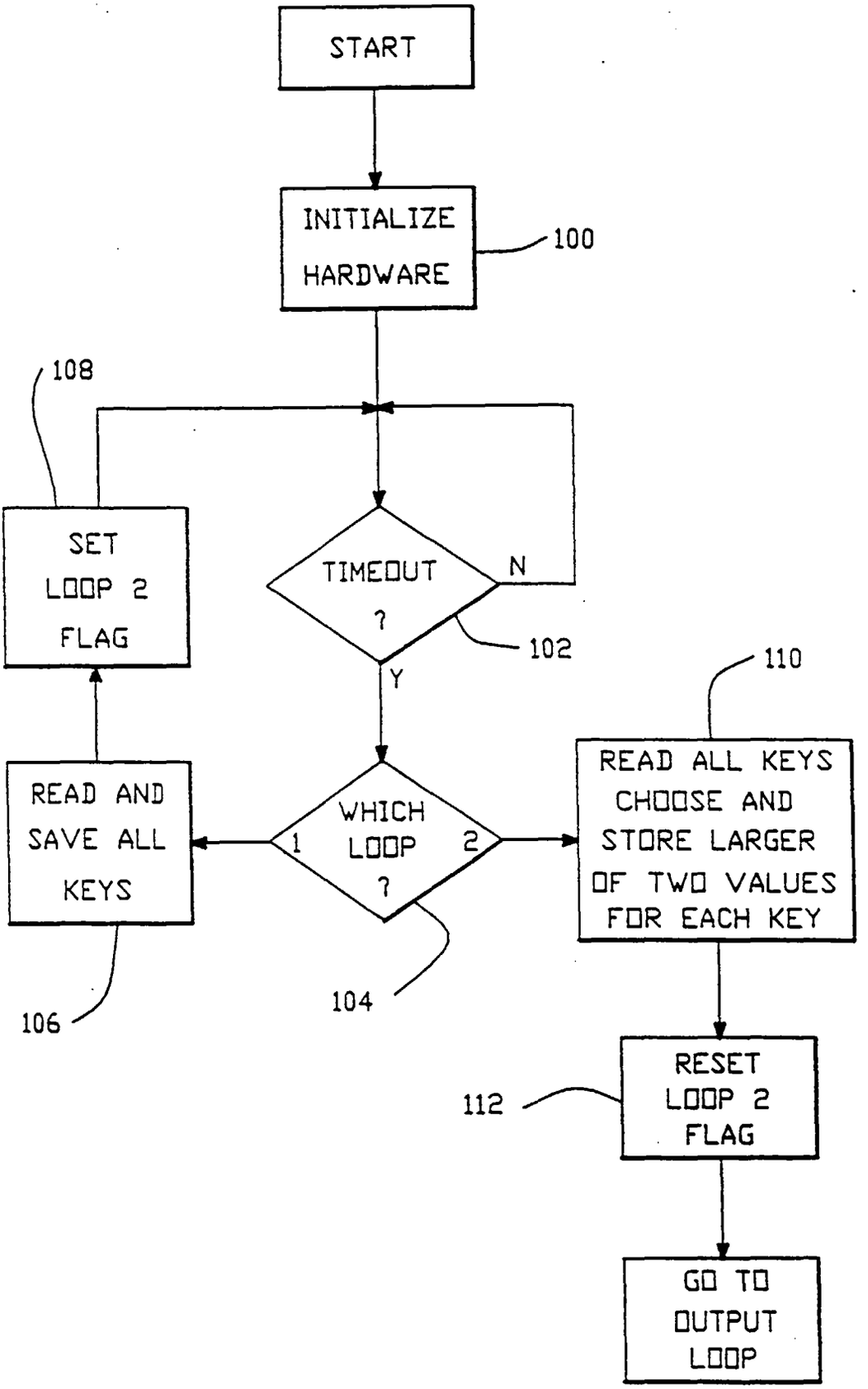

START
INITIALIZE HARDWARE
100
108
SET LOOP 2 FLAG
TIMEOUT ?
N
102
Y
110
READ ALL KEYS CHOOSE AND STORE LARGER OF TWO VALUES FOR EACH KEY
READ AND SAVE ALL KEYS
WHICH LOOP ?
1
2
106
104
RESET LOOP 2 FLAG
112
GO TO OUTPUT LOOP

FIG-3a

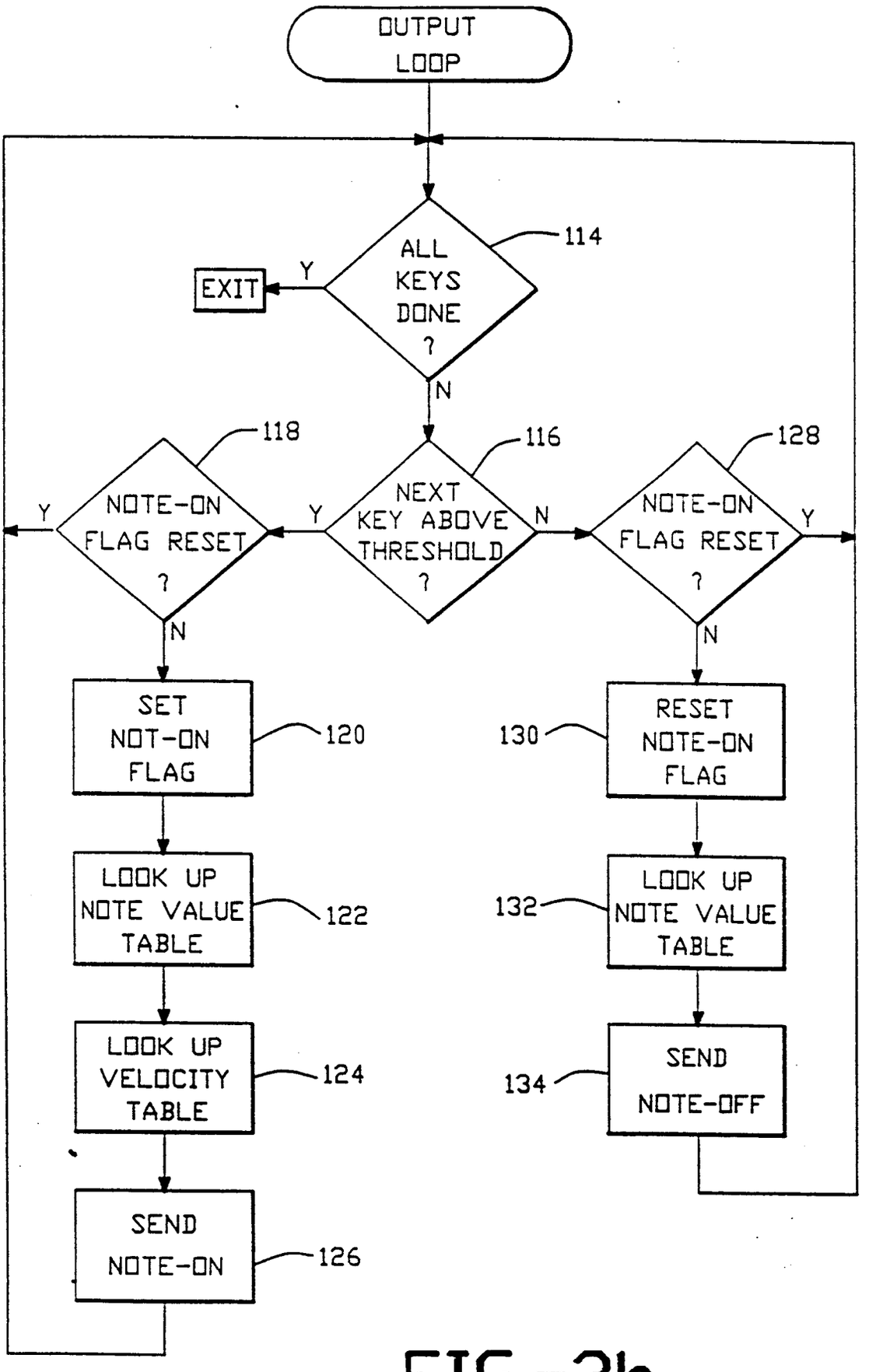

OUTPUT LOOP
ALL KEYS DONE ?
114
Y
EXIT
N
NEXT KEY ABOVE THRESHOLD ?
116
Y
N
NOTE-ON FLAG RESET ?
118
Y
N
SET NOT-ON FLAG
120
LOOK UP NOTE VALUE TABLE
122
LOOK UP VELOCITY TABLE
124
SEND NOTE-ON
126
NOTE-ON FLAG RESET ?
128
Y
N
RESET NOTE-ON FLAG
130
LOOK UP NOTE VALUE TABLE
132
SEND NOTE-OFF
134

FIG.-3b

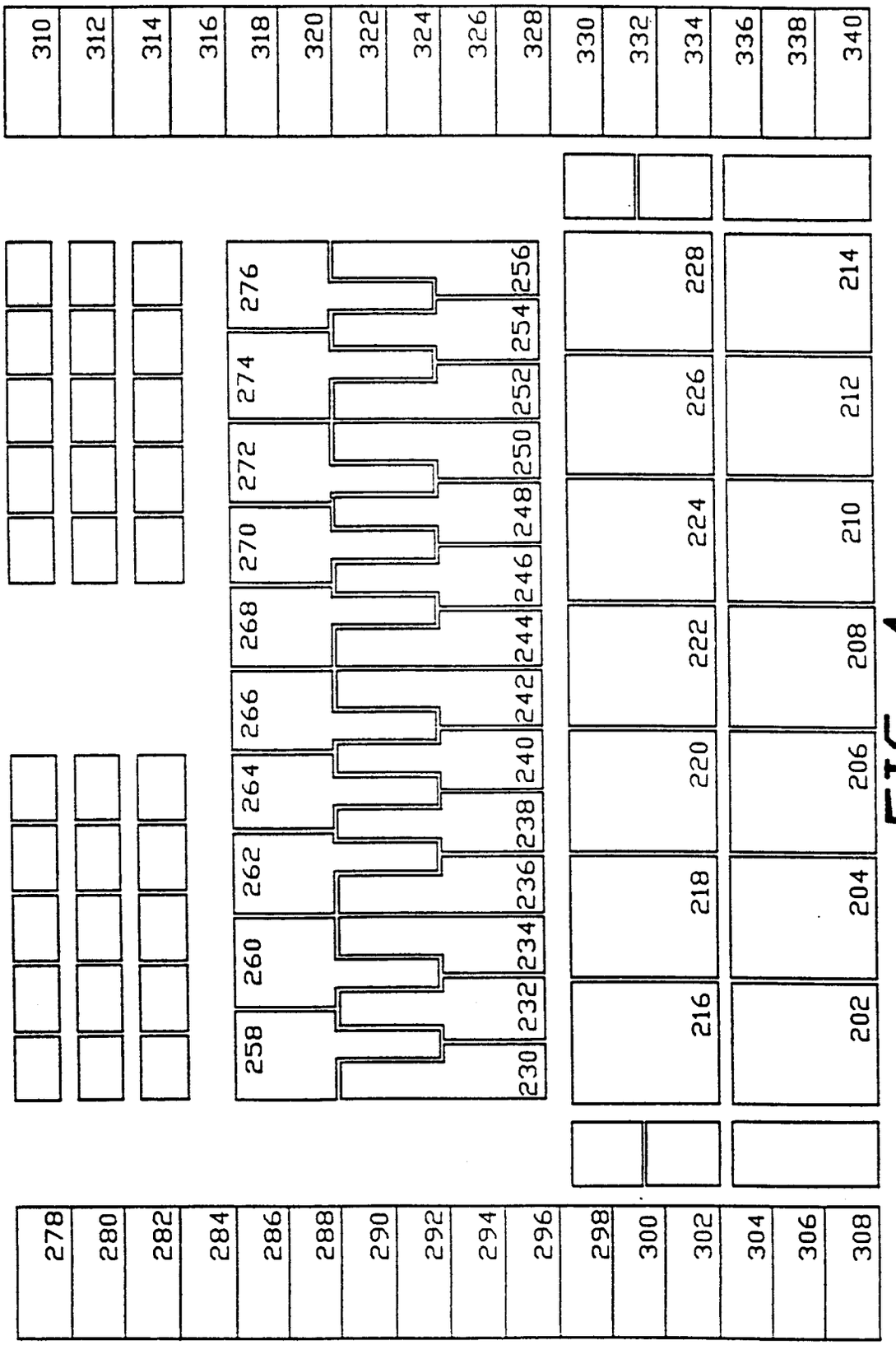

310
312
314
316
318
320
322
324
326
328
330
332
334
336
338
340
228
226
224
222
220
218
216
214
212
210
208
206
204
202
276
274
272
270
268
266
264
262
260
258
256
254
252
250
248
246
244
242
240
238
236
234
232
230
278
280
282
284
286
288
290
292
294
296
298
300
302
304
306
308

FIG.-4

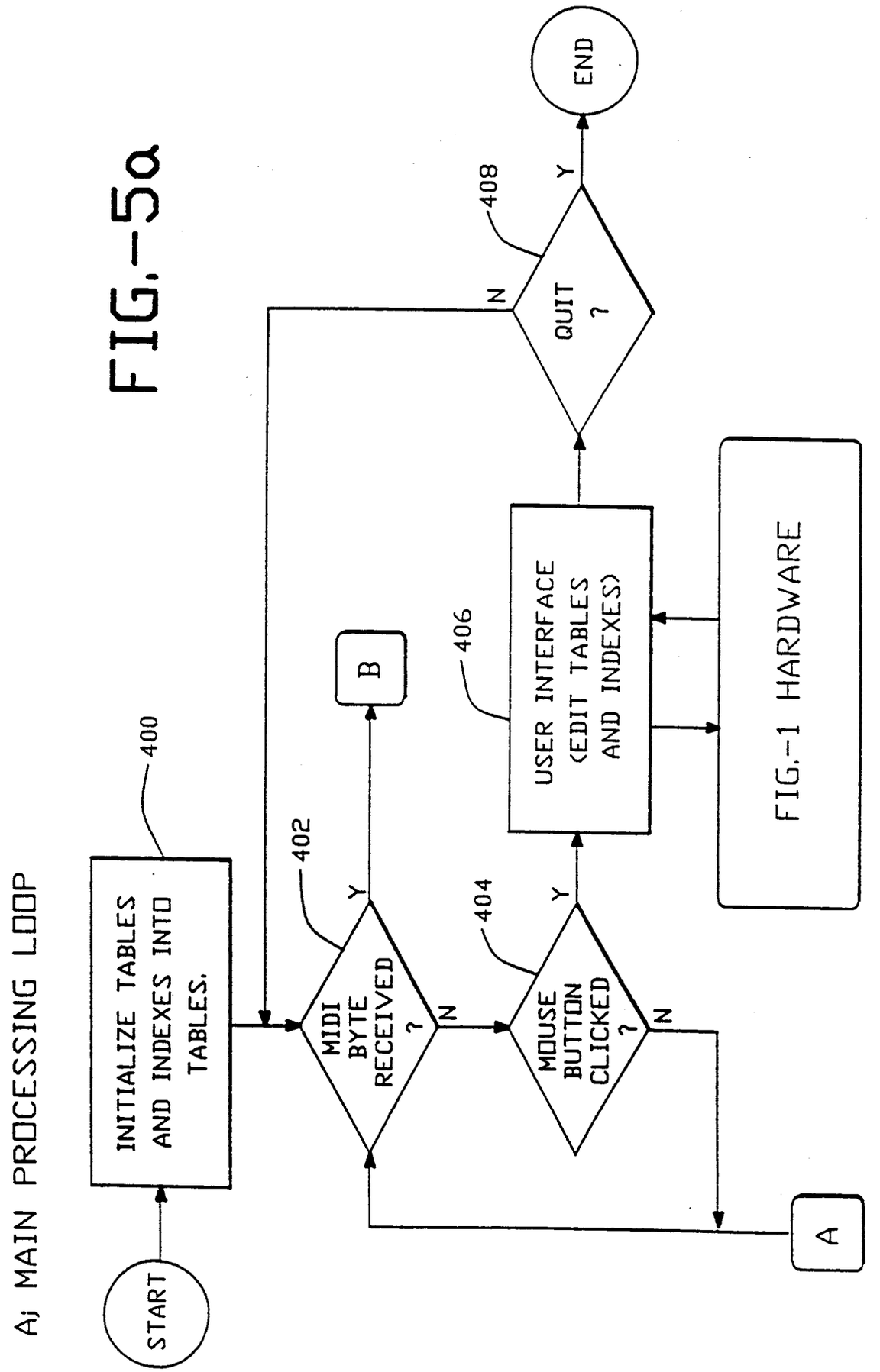

A; MAIN PROCESSING LOOP
START
INITIALIZE TABLES AND INDEXES INTO TABLES.
400
MIDI BYTE RECEIVED ?
402
Y
N
B
MOUSE BUTTON CLICKED ?
404
Y
N
A
USER INTERFACE (EDIT TABLES AND INDEXES)
406
QUIT ?
408
Y
N
END
FIG.-1 HARDWARE

FIG.-5a

FIG.-5b

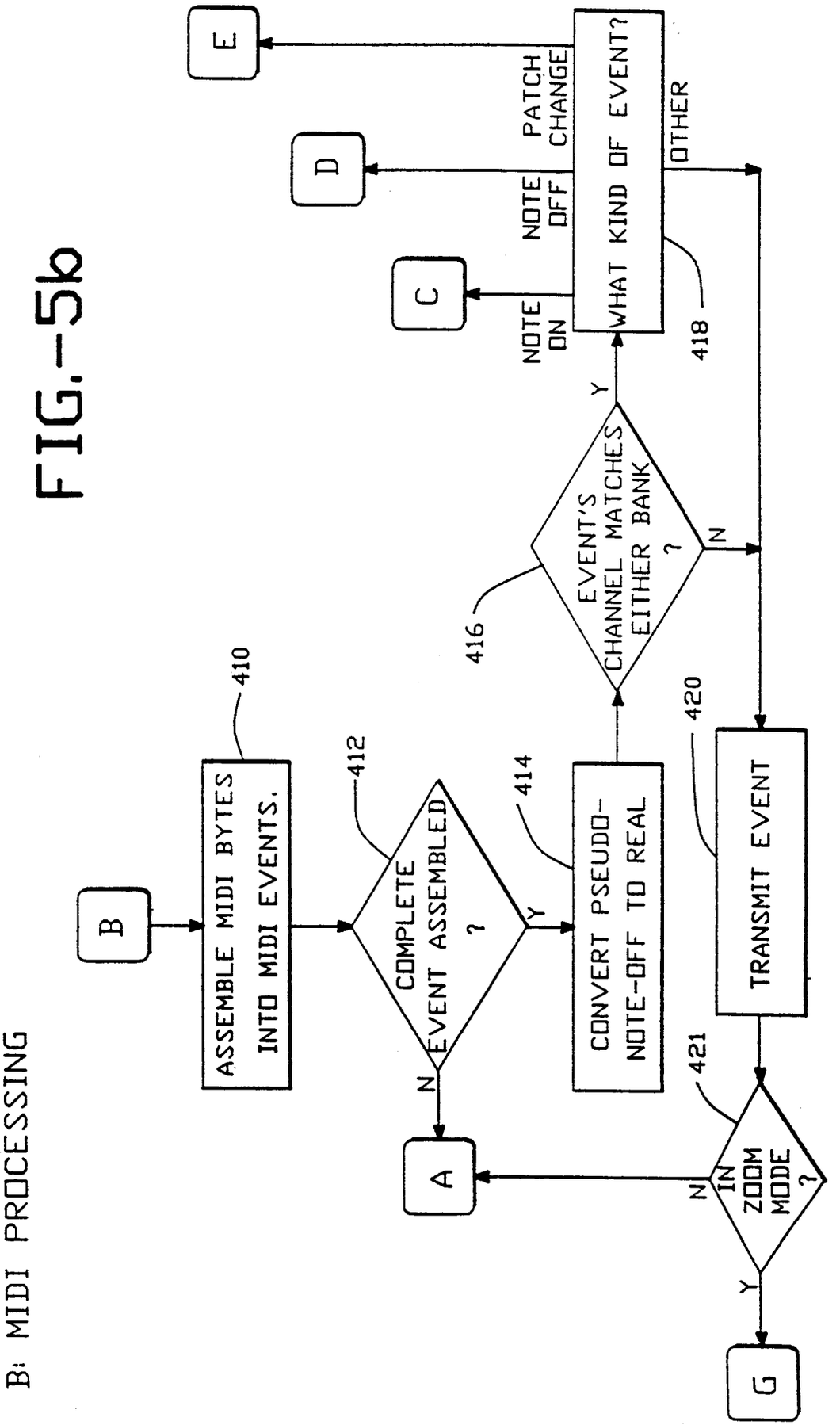

B: MIDI PROCESSING
B
ASSEMBLE MIDI BYTES INTO MIDI EVENTS.
410
COMPLETE EVENT ASSEMBLED ?
412
N
Y
A
CONVERT PSEUDO-NOTE-OFF TO REAL
414
EVENT'S CHANNEL MATCHES EITHER BANK ?
416
Y
N
WHAT KIND OF EVENT?
418
NOTE ON
NOTE OFF
PATCH CHANGE
OTHER
C
D
E
TRANSMIT EVENT
420
IN ZOOM MODE ?
421
N
Y
G

FIG.-5C

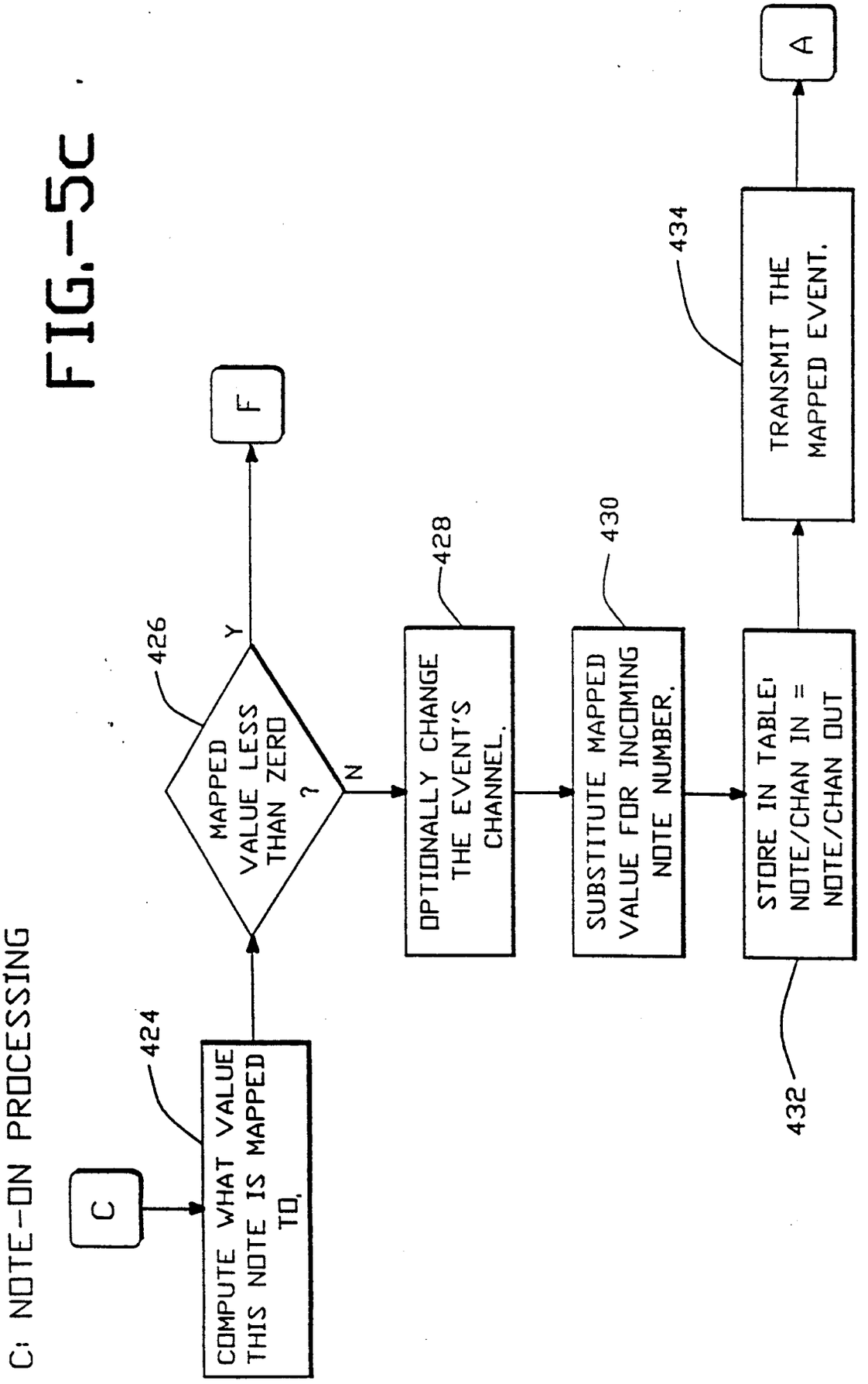
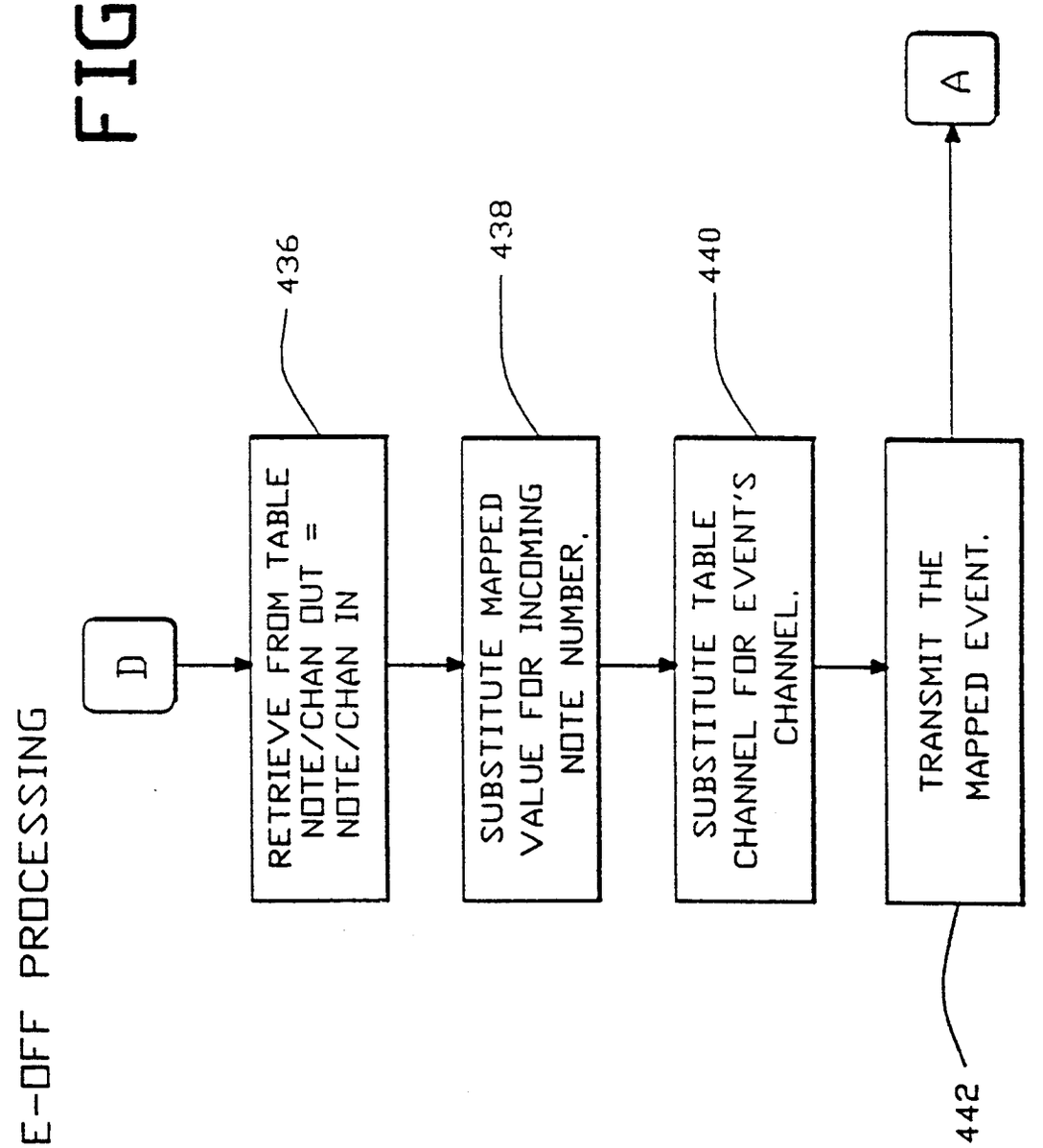

C: NOTE-ON PROCESSING
C
424
COMPUTE WHAT VALUE THIS NOTE IS MAPPED TO.
426
MAPPED VALUE LESS THAN ZERO ?
Y
F
N
428
OPTIONALLY CHANGE THE EVENT'S CHANNEL.
430
SUBSTITUTE MAPPED VALUE FOR INCOMING NOTE NUMBER.
432
STORE IN TABLE: NOTE/CHAN IN = NOTE/CHAN OUT
434
TRANSMIT THE MAPPED EVENT.
A

FIG.-5d


D: NOTE-OFF PROCESSING
D
RETRIEVE FROM TABLE NOTE/CHAN OUT = NOTE/CHAN IN
436
SUBSTITUTE MAPPED VALUE FOR INCOMING NOTE NUMBER.
438
SUBSTITUTE TABLE CHANNEL FOR EVENT'S CHANNEL.
440
TRANSMIT THE MAPPED EVENT.
442
A

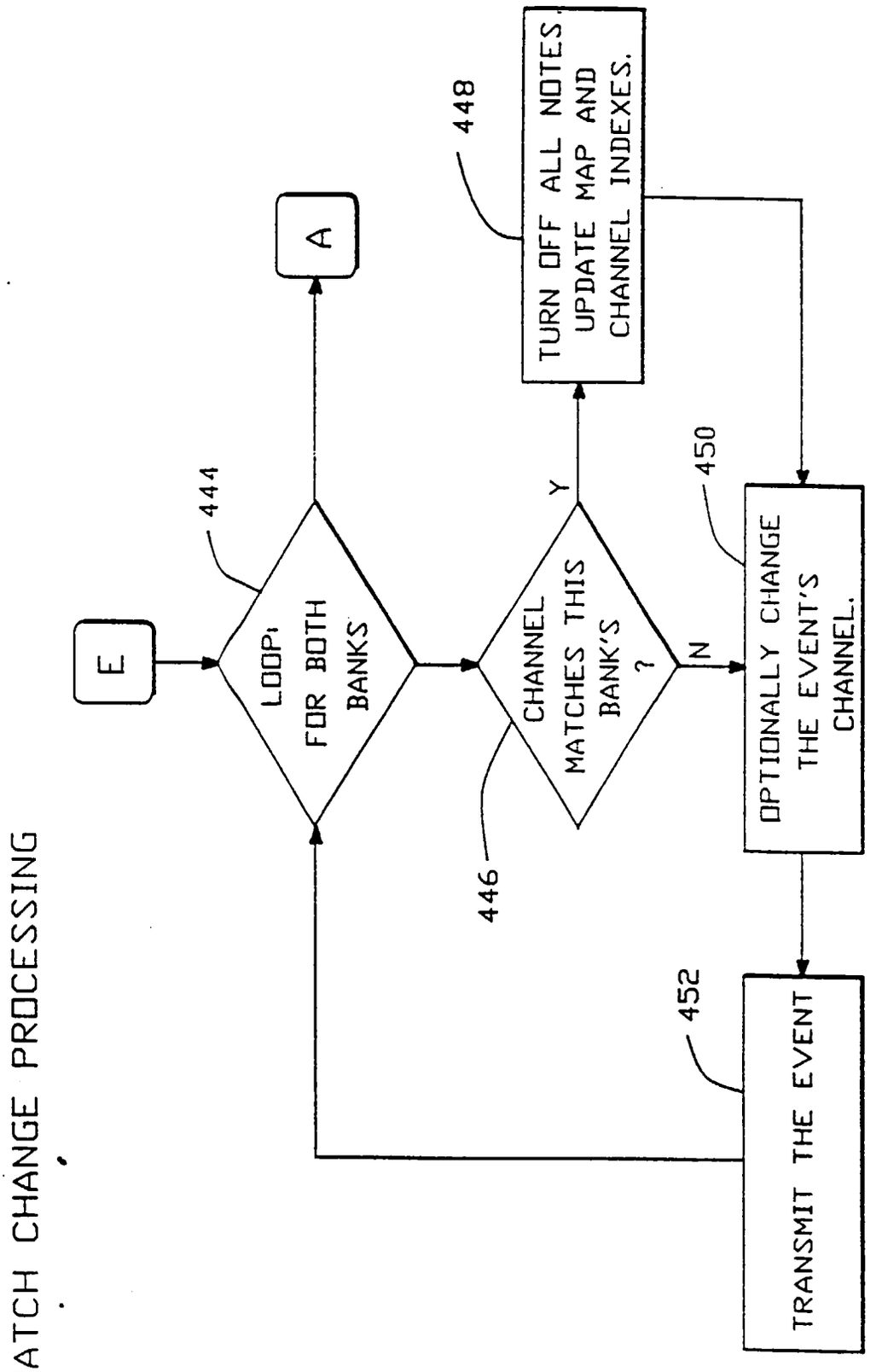
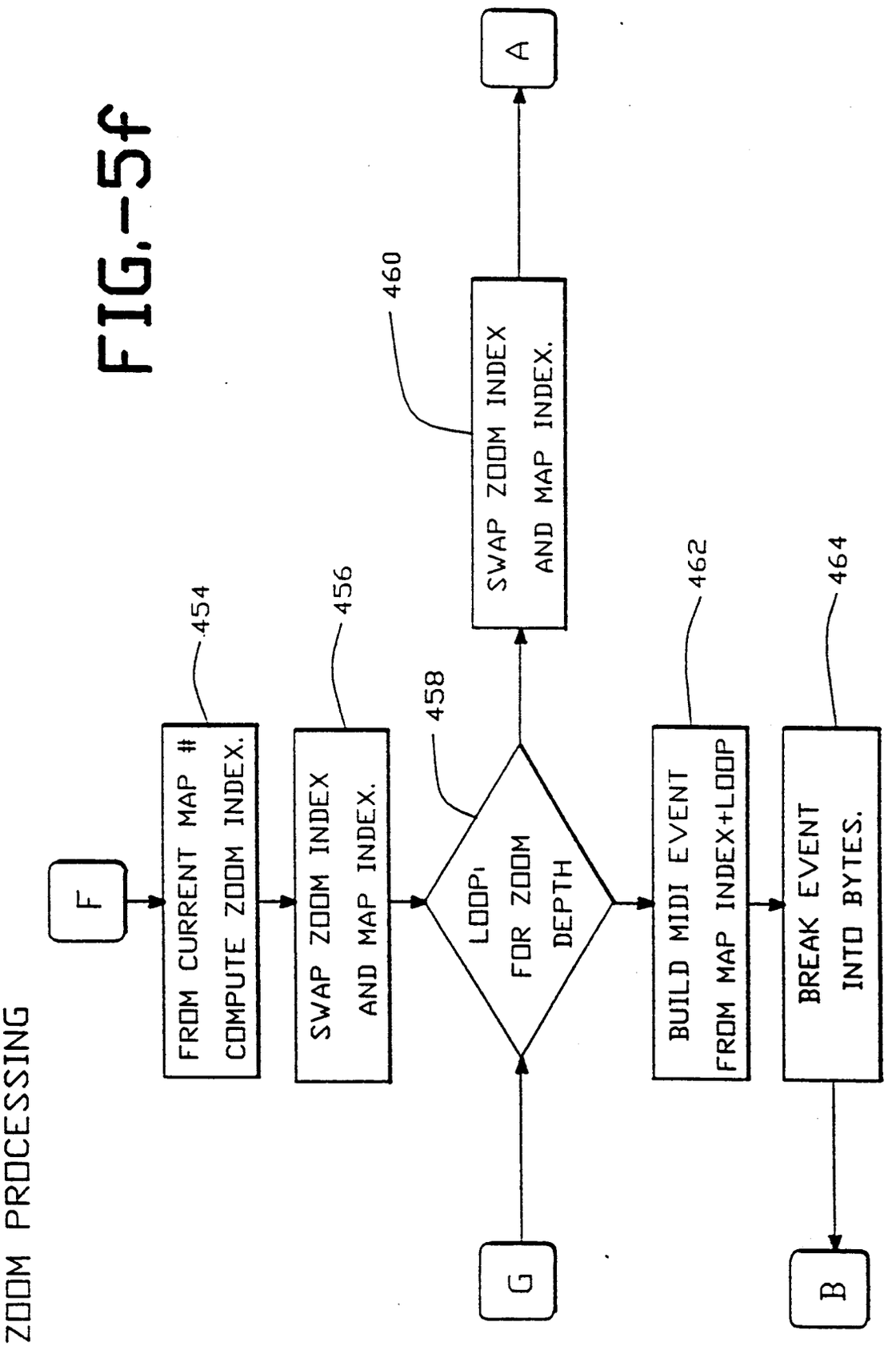

E' PATCH CHANGE PROCESSING
E
LOOP, FOR BOTH BANKS
444
A
CHANNEL MATCHES THIS BANK'S ?
Y
N
446
TURN OFF ALL NOTES, UPDATE MAP AND CHANNEL INDEXES.
448
OPTIONALLY CHANGE THE EVENT'S CHANNEL.
450
TRANSMIT THE EVENT
452

FIG.-5e

FIG.-5f


F: ZOOM PROCESSING
F
FROM CURRENT MAP # COMPUTE ZOOM INDEX.
454
SWAP ZOOM INDEX AND MAP INDEX.
456
LOOP, FOR ZOOM DEPTH
458
SWAP ZOOM INDEX AND MAP INDEX.
460
A
G
BUILD MIDI EVENT FROM MAP INDEX+LOOP
462
BREAK EVENT INTO BYTES.
464
B

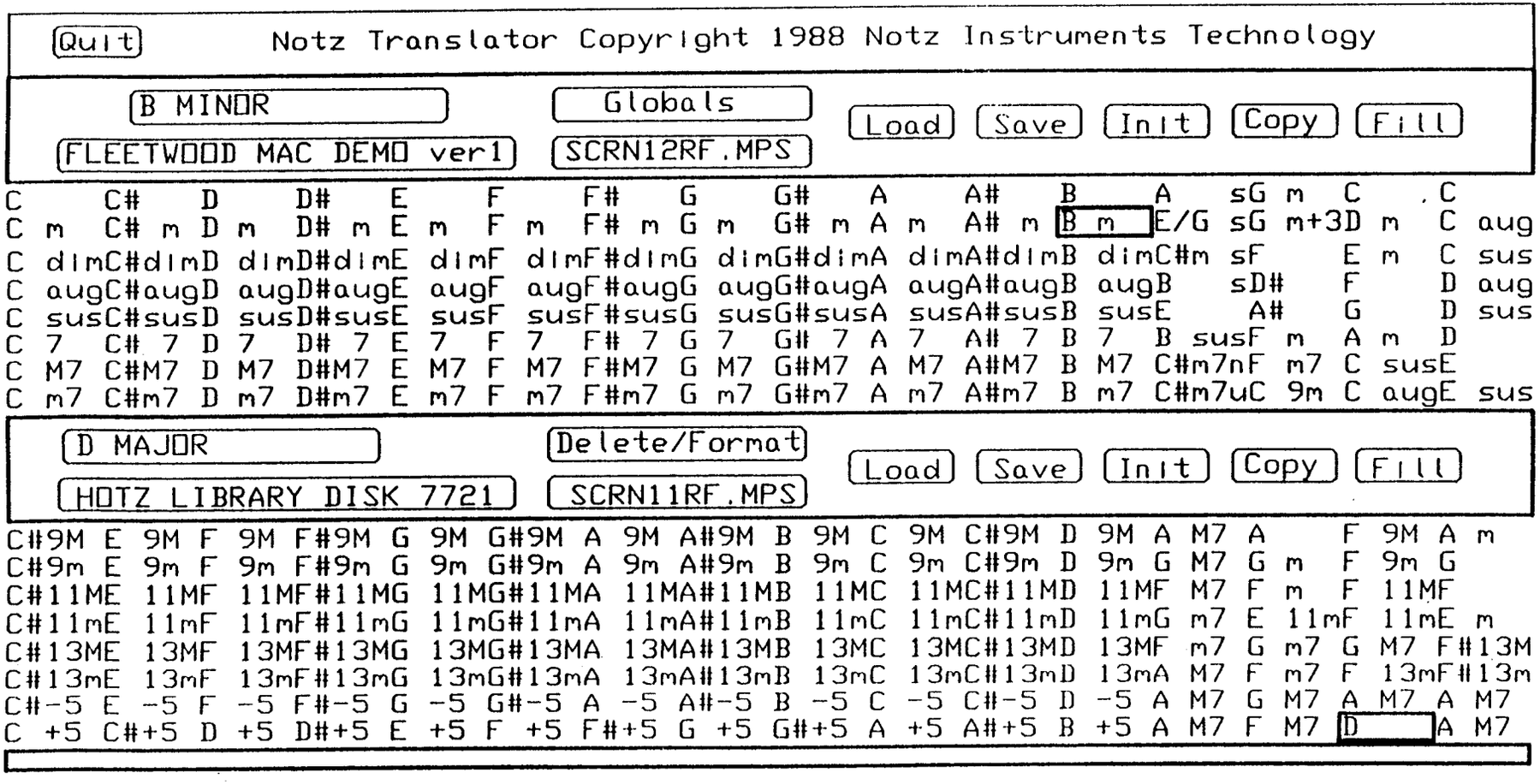

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
B MINOR
Globals
Load
Save
Init
Copy
Fill
FLEETWOOD MAC DEMO ver1
SCRN12RF.MPS
C C# D D# E F F# G G# A A# B A sG m C C
C m C# m D m D# m E m F m F# m G m G# m A m A# m B m E/G sG m+3D m C aug
C dimC#dimD dimD#dimE dimF dimF#dimG dimG#dimA dimA#dimB dimC#m sF E m C sus
C augC#augD augD#augE augF augF#augG augG#augA augA#augB augB sD# F D aug
C susC#susD susD#susE susF susF#susG susG#susA susA#susB susE A# G D sus
C 7 C# 7 D 7 D# 7 E 7 F 7 F# 7 G 7 G# 7 A 7 A# 7 B 7 B susF m A m D
C M7 C#M7 D M7 D#M7 E M7 F M7 F#M7 G M7 G#M7 A M7 A#M7 B M7 C#m7nF m7 C susE
C m7 C#m7 D m7 D#m7 E m7 F m7 F#m7 G m7 G#m7 A m7 A#m7 B m7 C#m7uC 9m C augE sus
D MAJOR
Delete/Format
Load
Save
Init
Copy
Fill
HOTZ LIBRARY DISK 7721
SCRN11RF.MPS
C#9M E 9M F 9M F#9M G 9M G#9M A 9M A#9M B 9M C 9M C#9M D 9M A M7 A F 9M A m
C#9m E 9m F 9m F#9m G 9m G#9m A 9m A#9m B 9m C 9m C#9m D 9m G M7 G m F 9m G
C#11ME 11MF 11MF#11MG 11MG#11MA 11MA#11MB 11MC 11MC#11MD 11MF M7 F m F 11MF
C#11mE 11mF 11mF#11mG 11mG#11mA 11mA#11mB 11mC 11mC#11mD 11mG m7 E 11mF 11mE m
C#13ME 13MF 13MF#13MG 13MG#13MA 13MA#13MB 13MC 13MC#13MD 13MF m7 G m7 G M7 F#13M
C#13mE 13mF 13mF#13mG 13mG#13mA 13mA#13mB 13mC 13mC#13mD 13mA M7 F m7 F 13mF#13m
C#-5 E -5 F -5 F#-5 G -5 G#-5 A -5 A#-5 B -5 C -5 C#-5 D -5 A M7 G M7 A M7 A M7
C +5 C#+5 D +5 D#+5 E +5 F +5 F#+5 G +5 G#+5 A +5 A#+5 B +5 A M7 F M7 D A M7

FIG.-6a

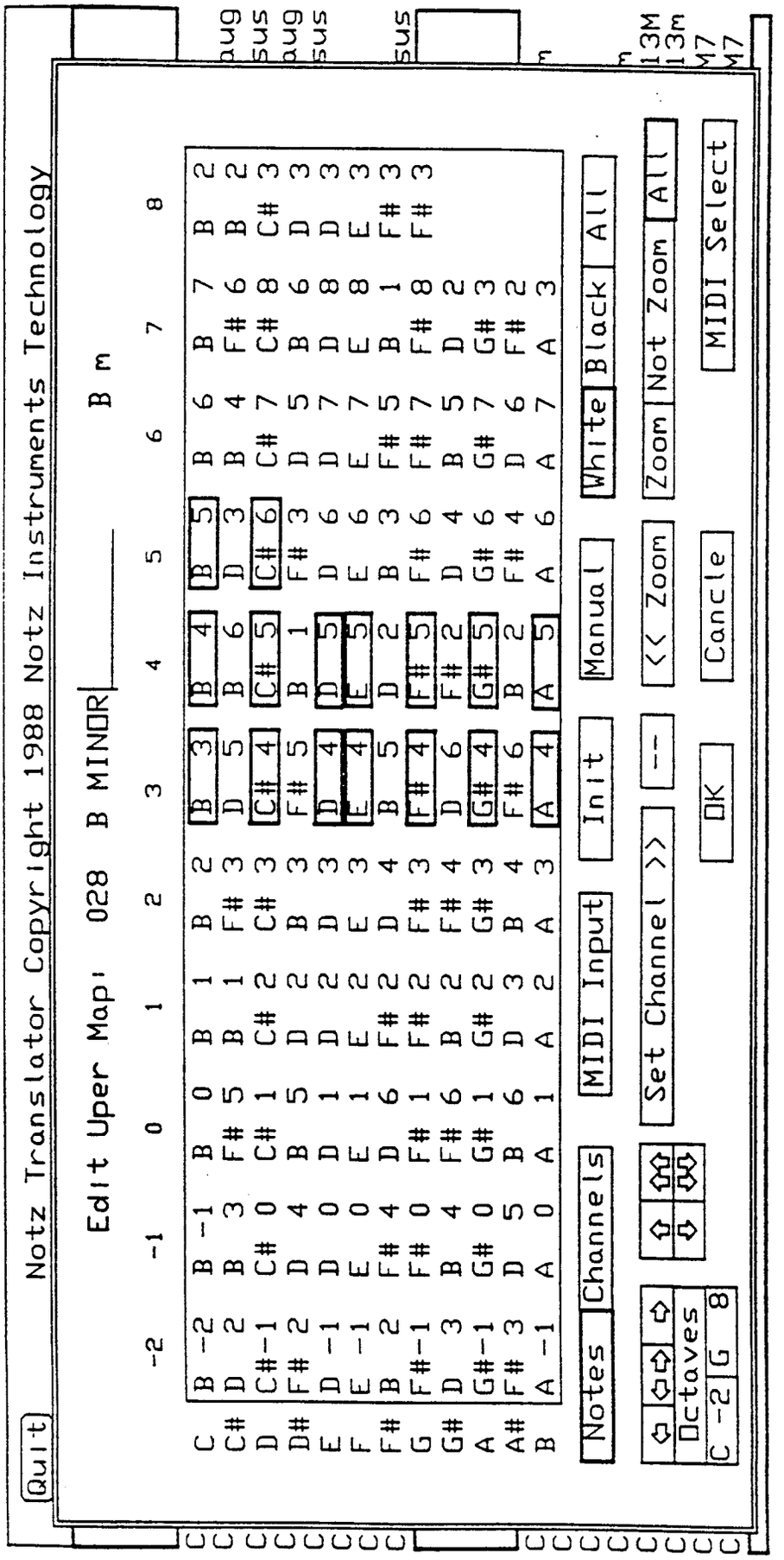

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
Edit Uper Map: 028 B MINOR
B m
-2 -1 0 1 2 3 4 5 6 7 8
C B -2 B -1 B 0 B 1 B 2 B 3 B 4 B 5 B 6 B 7 B 2
C# D 2 B 3 F# 5 B 1 F# 3 D 5 B 4 D 3 B 4 F# 6 B 2
D C#-1 C# 0 C# 1 C# 2 C# 3 C# 4 C# 5 C# 6 C# 7 C# 8 C# 3
D# F# 2 D 4 B 5 D 2 B 3 F# 5 B 1 F# 3 D 5 B 6 D 3
E D -1 D 0 D 1 D 2 D 3 D 4 D 5 D 6 D 7 D 8 D 3
F E -1 E 0 E 1 E 2 E 3 E 4 E 5 E 6 E 7 E 8 E 3
F# B 2 F# 4 D 6 F# 2 D 4 B 5 D 2 B 3 F# 5 B 1 F# 3
G F#-1 F# 0 F# 1 F# 2 F# 3 F# 4 D 6 F# 2 D 4 B 5 D 2
G# D 3 B 4 F# 6 B 2 F# 4 D 6 F# 5 D 4 B 5 D 2
A G#-1 G# 0 G# 1 G# 2 G# 3 G# 4 G# 5 G# 6 G# 7 G# 3
A# F# 3 D 5 B 6 D 3 B 4 F# 6 B 2 F# 4 D 6 F# 2
B A -1 A 0 A 1 A 2 A 3 A 4 A 5 A 6 A 7 A 3
Notes
Channels
MIDI Input
Init
Manual
White
Black
All
Octaves
C -2
G 8
Set Channel >>
--
<< Zoom
Zoom
Not Zoom
All
OK
Cancle
MIDI Select
aug
sus
aug
sus
sus
m
m
13M
13m
M7
M7

FIG.-6b

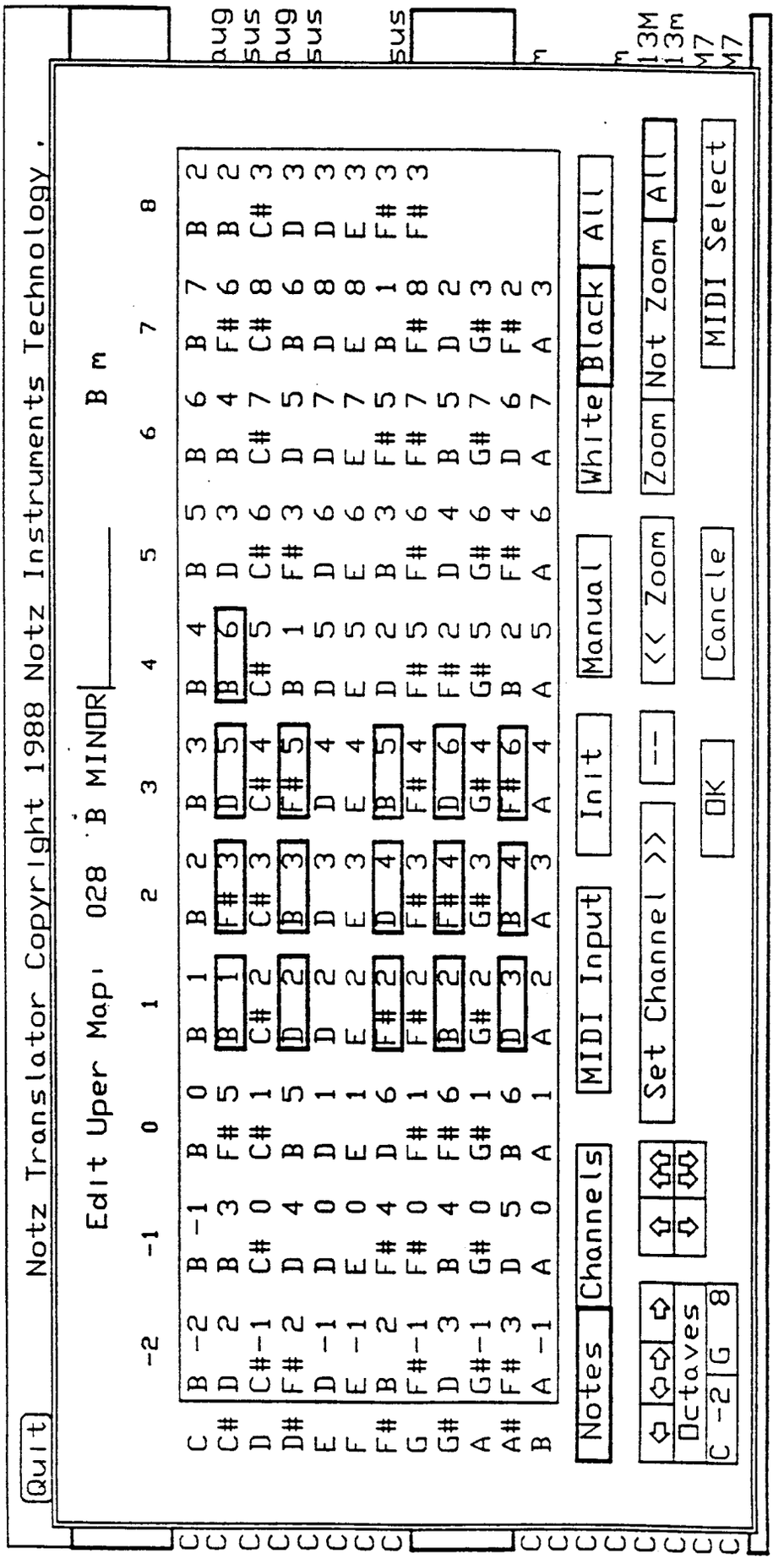

Quit
Notz Translator Copyright 1988 Notz Instruments Technology.
Edit Uper Map: 028 B MINOR
B m
-2 -1 0 1 2 3 4 5 6 7 8
C B -2 B -1 B 0 B 1 B 2 B 3 B 4 B 5 B 6 B 7 B 2
C# D 2 B 3 F# 5 B 1 F# 3 D 5 B 6 D 3 B 4 F# 6 B 2
D C#-1 C# 0 C# 1 C# 2 C# 3 C# 4 C# 5 C# 6 C# 7 C# 8 C# 3
D# F# 2 D 4 B 5 D 2 B 3 F# 5 B 1 F# 3 D 5 B 6 D 3
E D -1 D 0 D 1 D 2 D 3 D 4 D 5 D 6 D 7 D 8 D 3
F E -1 E 0 E 1 E 2 E 3 E 4 E 5 E 6 E 7 E 8 E 3
F# B 2 F# 4 D 6 F# 2 D 4 B 5 D 2 B 3 F# 5 B 1 F# 3
G F#-1 F# 0 F# 1 F# 2 F# 3 F# 4 F# 5 F# 6 F# 7 F# 8 F# 3
G# D 3 B 4 F# 6 B 2 F# 4 D 6 F# 2 D 4 B 5 D 2
A G#-1 G# 0 G# 1 G# 2 G# 3 G# 4 G# 5 G# 6 G# 7 G# 3
A# F# 3 D 5 B 6 D 3 B 4 F# 6 B 2 F# 4 D 6 F# 2
B A -1 A 0 A 1 A 2 A 3 A 4 A 5 A 6 A 7 A 3
Notes
Channels
MIDI Input
Init
Manual
White
Black
All
Octaves
C -2
G 8
Set Channel >>
--
<< Zoom
Zoom
Not Zoom
All
OK
Cancle
MIDI Select
sus
aug
sus
aug
sus
m
m
13M
13m
M7
M7

FIG.-6C

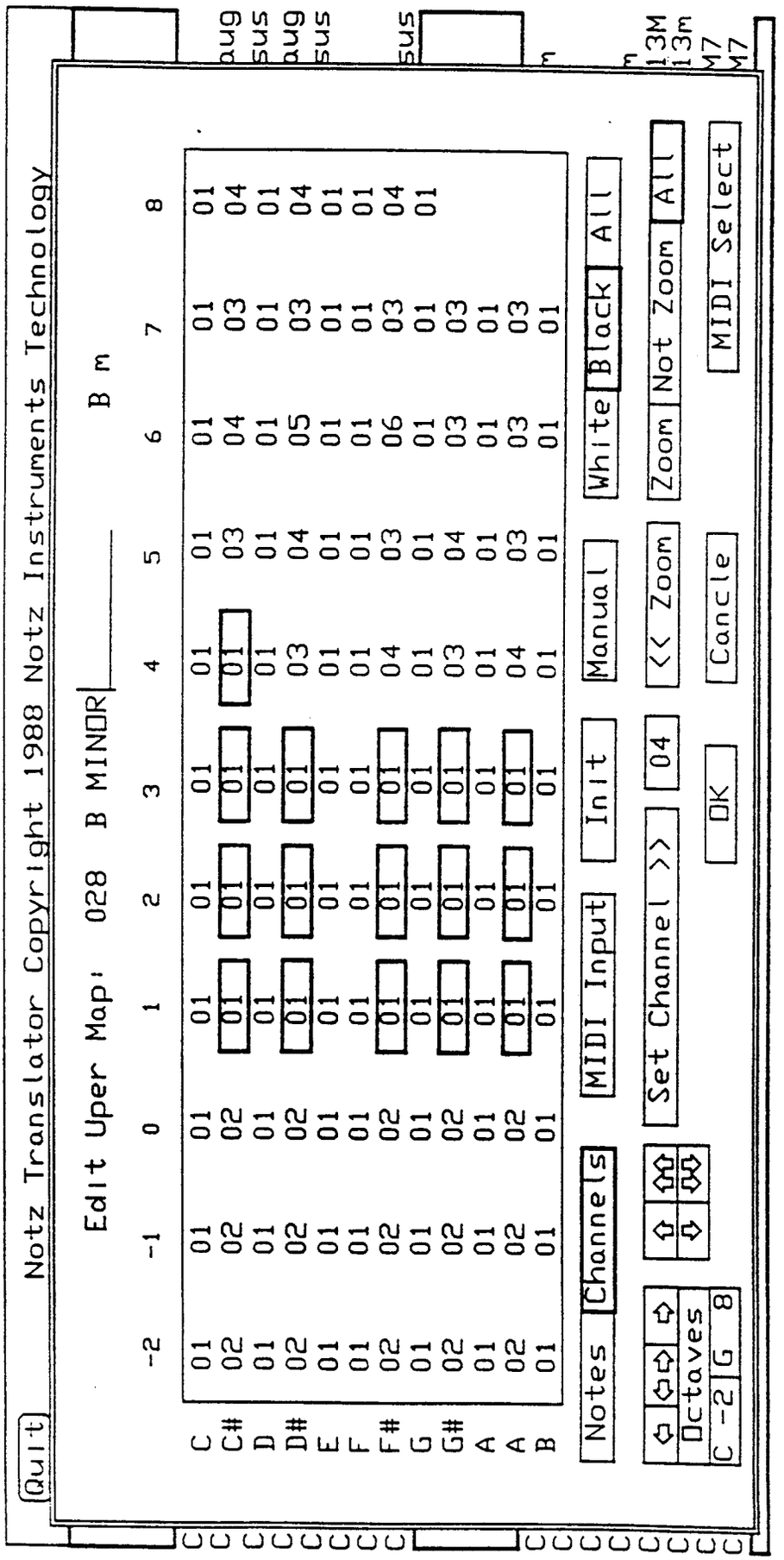

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
Edit Uper Map: 028 B MINOR
B m
-2 -1 0 1 2 3 4 5 6 7 8
C C# D D# E F F# G G# A A# B
Notes
Channels
Octaves
C -2 G 8
MIDI Input
Init
Manual
Set Channel >>
04
<< Zoom
OK
Cancle
White
Black
All
Zoom
Not Zoom
All
MIDI Select
aug sus aug sus sus m m
13M 13m M7 M7

FIG.-6d

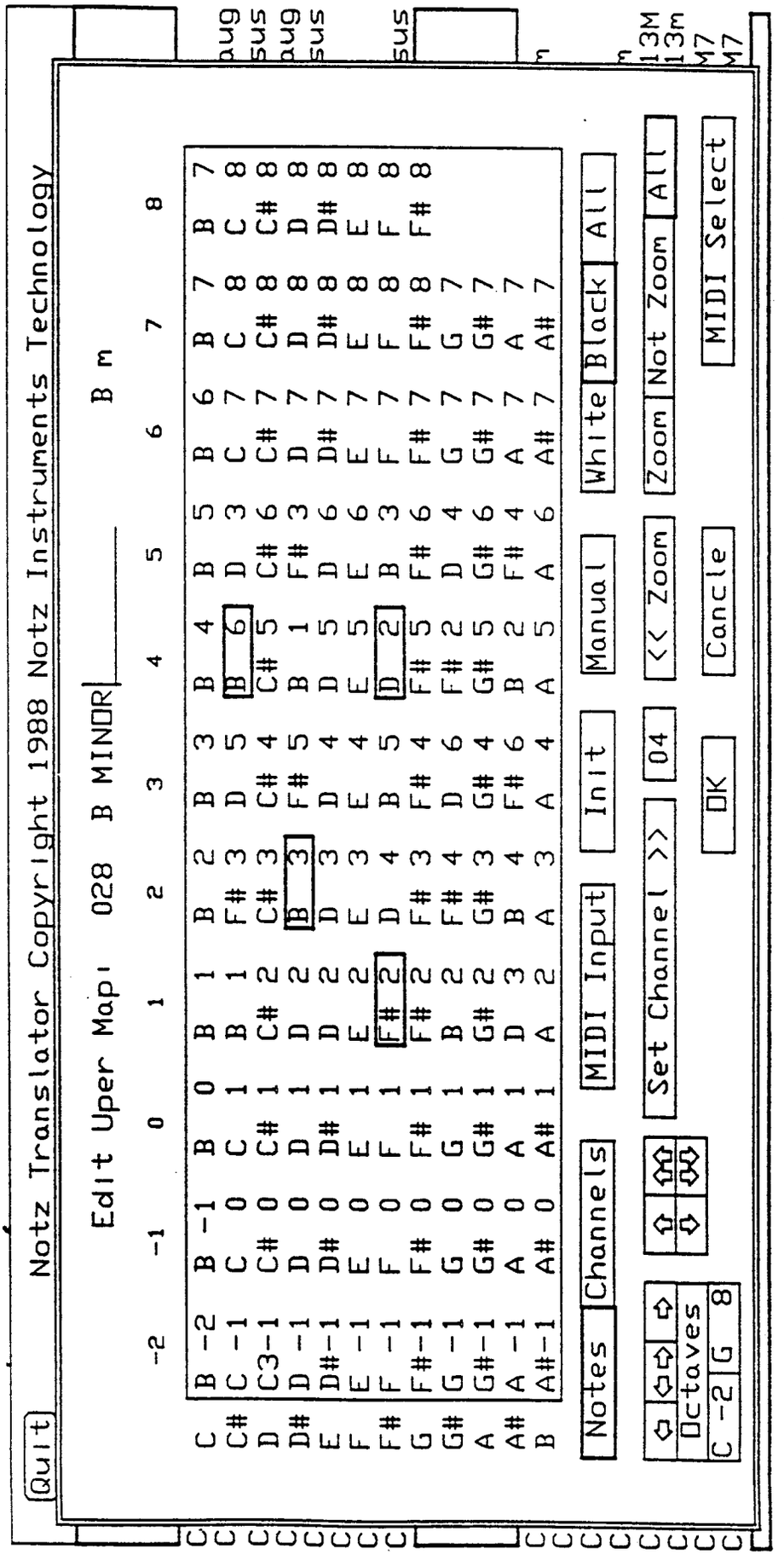

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
Edit Uper Map: 028 B MINOR
B m
-2 -1 0 1 2 3 4 5 6 7 8
C B -2 B -1 B 0 B 1 B 2 B 3 B 4 B 5 B 6 B 7 B 7
C# C -1 C 0 C 1 B 1 F# 3 D 5 B 6 D 3 C 7 C 8 C 8
D C3-1 C# 0 C# 1 C# 2 C# 3 C# 4 C# 5 C# 6 C# 7 C# 8 C# 8
D# D -1 D 0 D 1 D 2 B 3 F# 5 B 1 F# 3 D 7 D 8 D 8
E D#-1 D# 0 D# 1 D 2 D 3 D 4 D 5 D 6 D# 7 D# 8 D# 8
F E -1 E 0 E 1 E 2 E 3 E 4 E 5 E 6 E 7 E 8 E 8
F# F -1 F 0 F 1 F# 2 D 4 B 5 D 2 B 3 F 7 F 8 F 8
G F#-1 F# 0 F# 1 F# 2 F# 3 F# 4 F# 5 F# 6 F# 7 F# 8 F# 8
G# G -1 G 0 G 1 B 2 F# 4 D 6 F# 2 D 4 G 7 G 7
A G#-1 G# 0 G# 1 G# 2 G# 3 G# 4 G# 5 G# 6 G# 7 G# 7
A# A -1 A 0 A 1 D 3 B 4 F# 6 B 2 F# 4 A 7 A 7
B A#-1 A# 0 A# 1 A 2 A 3 A 4 A 5 A 6 A# 7 A# 7
Notes Channels
MIDI Input
Init
Manual
White Black All
Octaves
Set Channel >>
04
<< Zoom
Zoom Not Zoom All
C -2 G 8
OK
Cancle
MIDI Select
aug sus aug sus sus
13M 13m M7 M7

FIG.-6e

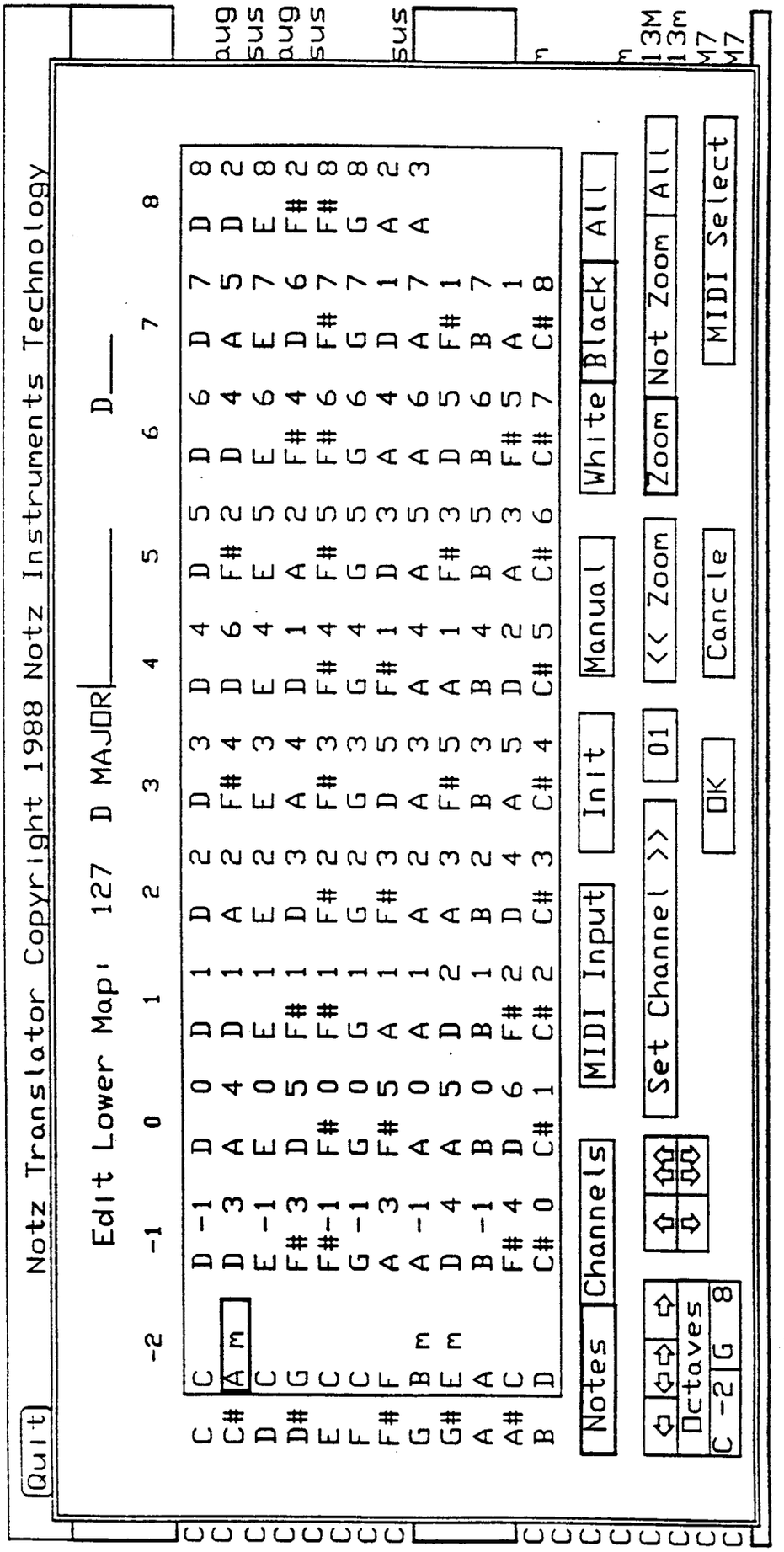

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
Edit Lower Map: 127 D MAJOR
Notes
Channels
Octaves
C -2 G 8
MIDI Input
Init
Manual
Set Channel >>
01
<< Zoom
White
Black
All
Zoom
Not Zoom
All
OK
Cancle
MIDI Select

FIG.-6f

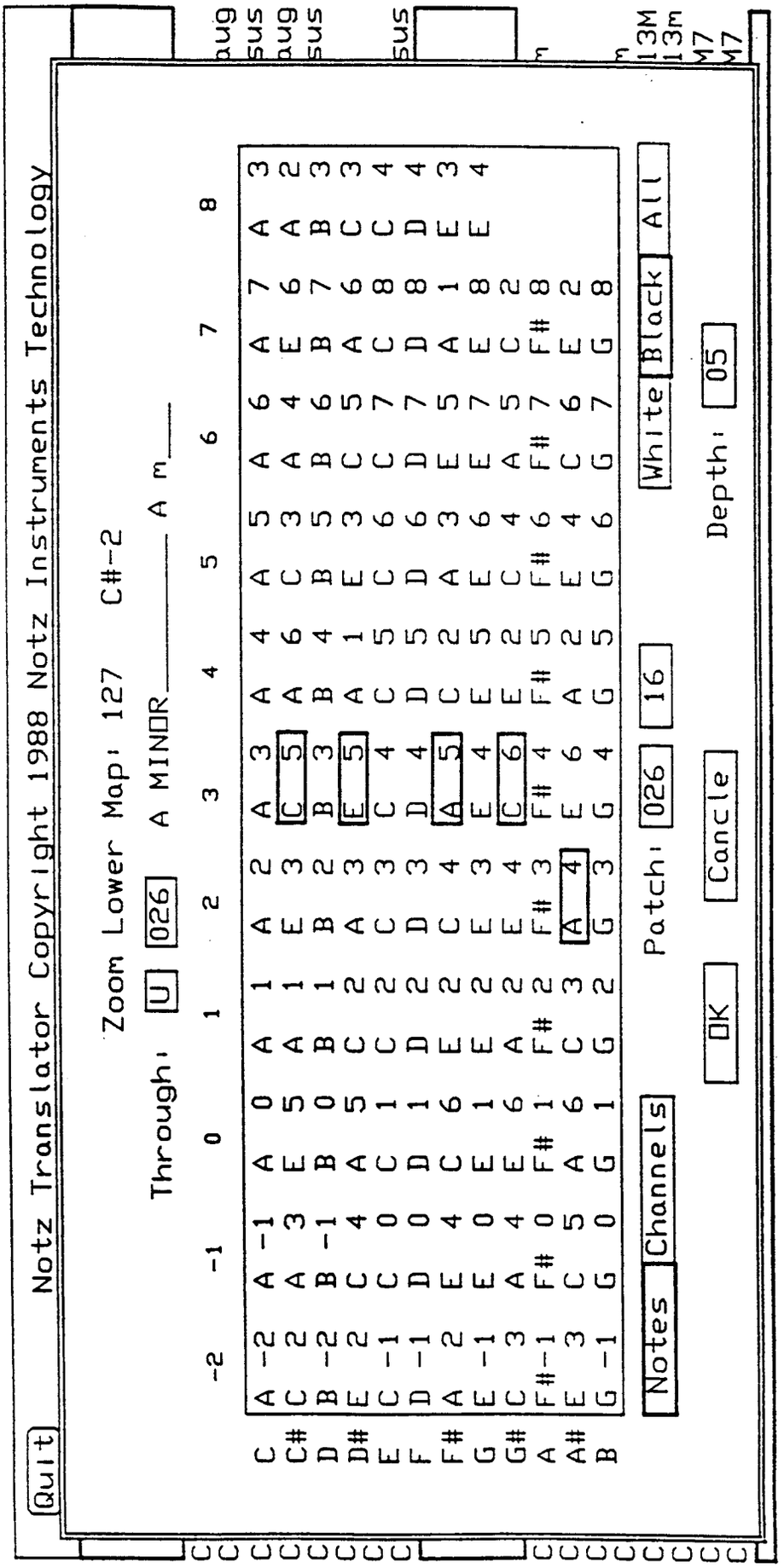

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
Through: U 026
Zoom Lower Map: 127
C#-2
A MINOR
A m
Notes
Channels
OK
Cancle
Patch: 026
16
White
Black
All
Depth: 05
aug
sus
aug
sus
sus
m
m
13M
13m
M7
M7

FIG.-69

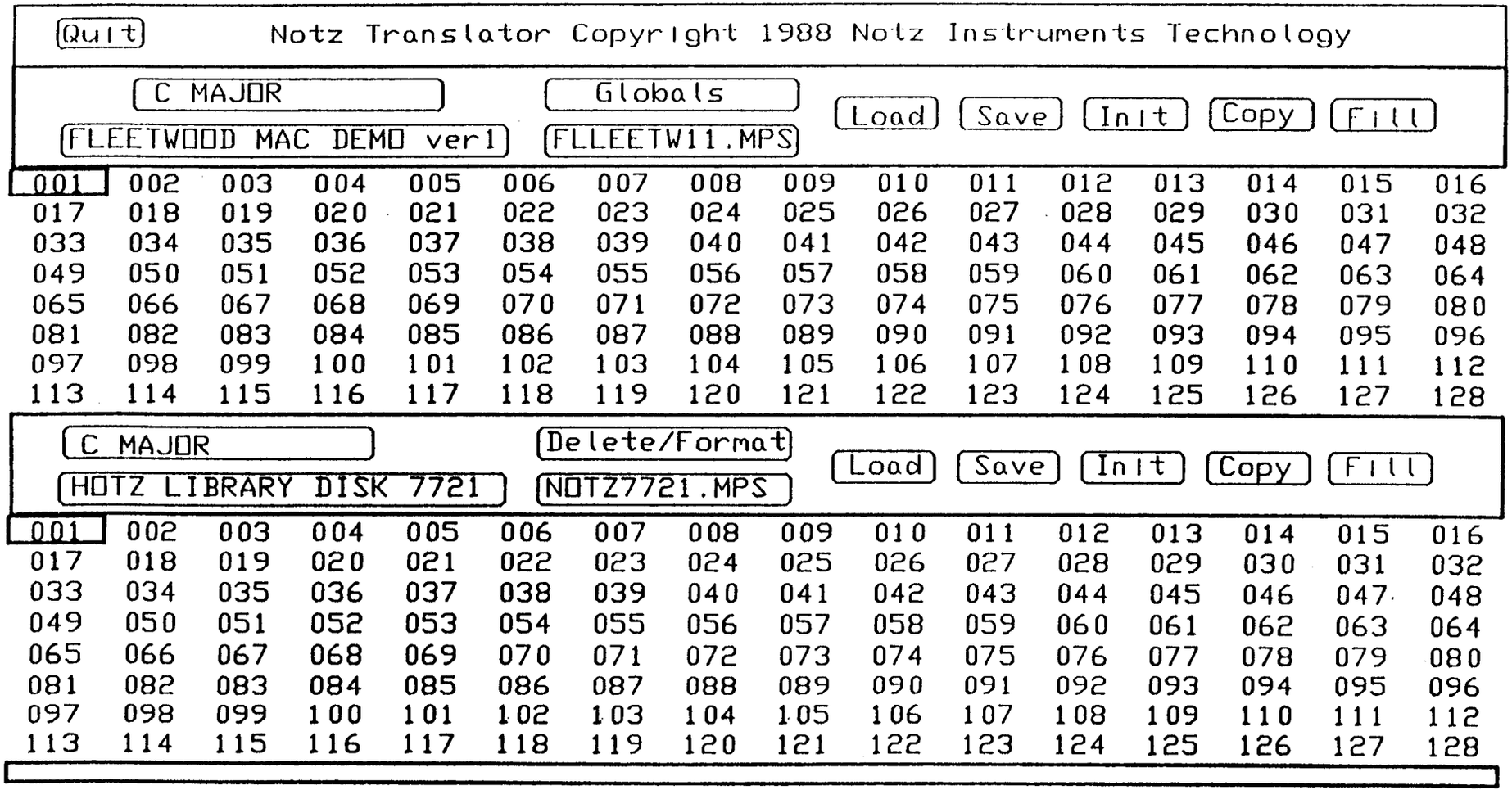


FIG.-6h

EP 0 795 850 A2

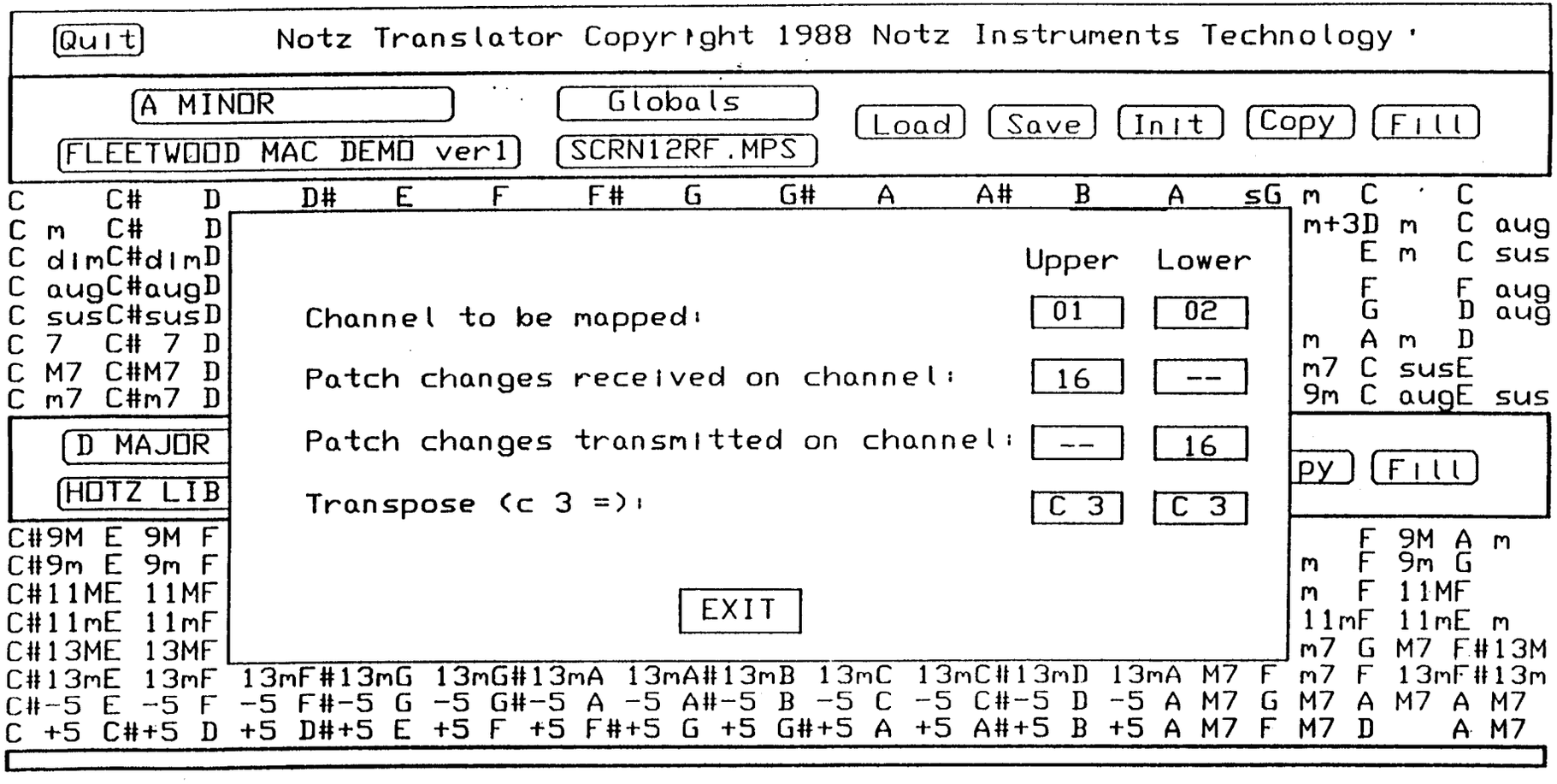

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
A MINOR
Globals
Load
Save
Init
Copy
Fill
FLEETWOOD MAC DEMO ver1
SCRN12RF.MPS
Upper
Lower
Channel to be mapped:
01
02
Patch changes received on channel:
16
--
Patch changes transmitted on channel:
--
16
Transpose (c 3 =):
C 3
C 3
EXIT
D MAJOR
HOTZ LIB

FIG.-61

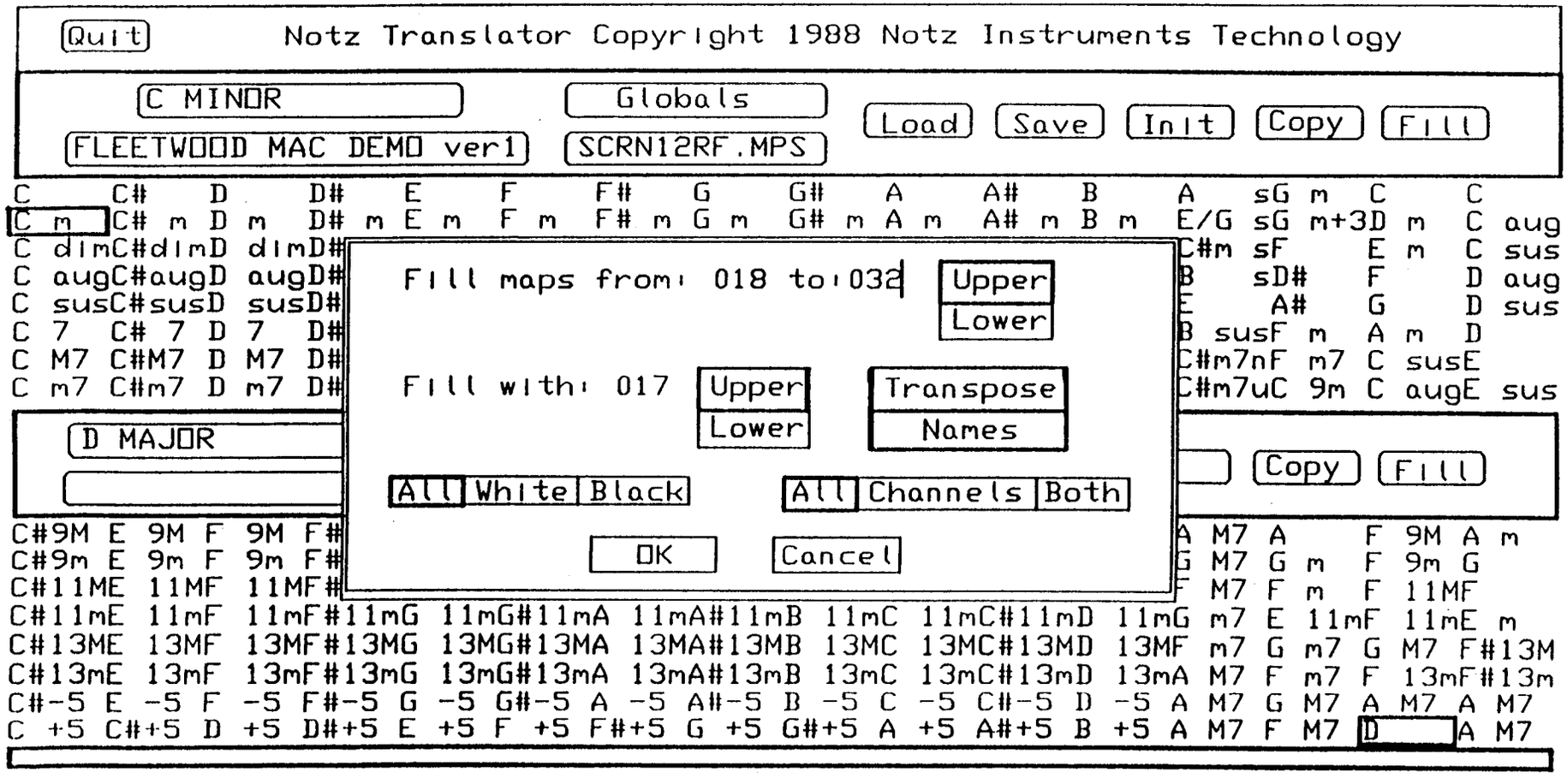

Quit
Notz Translator Copyright 1988 Notz Instruments Technology
C MINOR
Globals
Load
Save
Init
Copy
Fill
FLEETWOOD MAC DEMO ver1
SCRN12RF.MPS
Fill maps from: 018 to: 032
Upper
Lower
Fill with: 017
Upper
Lower
Transpose
Names
All White Black
All Channels Both
OK
Cancel
D MAJOR
Copy
Fill

FIG.-6J