bit 99

PROGRAMMABLE
POLYPHONIC
TOUCH SENSITIVE
SYNTHESIZER

OWNER'S MANUAL

CONTENTS

Introduction 2
MIDI and Audio Connections 3
Adjusting Volume and Pitch 5
Selecting Programs 6
Parameter Controls 7
Sound Generation and Control Parameters 8
The DCOs (Digitally Controlled Oscillators) 9
The VCFs (Voltage Controlled Filters) 11
The VCAs (Voltage Controlled Amplifier) 13
The LFOs (Low Frequency Oscillators) 15
MIDI Control Parameters 17
Storing Programs 21
Copying Programs Within Memory 22
"Parking" and Comparing Sounds 22
Two Synthesizers in One: Double and Split 23
Selecting Doubles and Splits 24
Programming Doubles and Splits 24
Program Chaining 26
The Cassette Interface 27
Storing Programs on Cassette 28
Loading Programs from Cassette 28

Appendix: MIDI Details and MIDI Exclusive-Codes 29
General MIDI Codes 30
MIDI System Exclusive Codes 31
Bitmaps 32
Notes 34

Owners Manual Version 1.0, June 1985
Written by Dave Th. Hutmacher and Abi Gassmann

BIT 99 - OWNERS MANUAL

Congratulations on your choice of the BIT 99! Your new instrument offers a perfect blend of advanced technology and easy operation and - above all - provides unlimited musical potential! But, before you start to explore this wonderful new world of sound, there are just three things we'd like you to do:

■ before switching on for the first time, check that the local supply voltage is the same as that given on the type plate on the rear panel.

■ fill out the warranty card, and have it stamped by your dealer. Your warranty is not valid otherwise.

■ read this manual carefully, even if you are already an experienced synthesizer user. You don't want to miss out on any of the features the BIT 99 has to offer!

And now: have fun with your BIT 99!

WARNING

Electrical energy can perform many useful functions. This unit has been designed and manufactured to assure your safety. Improper use can result in potential electric shock or fire hazards. To prevent electric shock do not expose this appliance to rain or moisture. Do not remove outer cover. No user servicable parts inside. Refer servicing to qualified personnel.

MIDI AND AUDIO CONNECTIONS

The BIT 99 is a highly versatile instrument, that can be used both as an autonomous single instrument, or as part of a MIDI music system. It has a touch sensitive keyboard, and contains all the components of a state of the art synthesizer needed for generating and modifying sound, storing programs etc. The BIT 99 can be used as a "Master Keyboard". controlling secondary instruments or "Slaves". *It can act as a "Slave" itself, if another MIDI Master Keyboard is* controlling the system, or indeed operate in both modes at once!

When the BIT 99 is hooked up and operated as the main keyboard in a MIDI system, it controls other instruments such as the **BIT 01 EXPANDER**. To do this, vast amounts of digital data are transmitted using a standardized interface (MIDI **M**usical **I**nstruments **D**igital **I**nterface), telling the slaved instrument what to do. This data includes information on which keys have been pressed, key velocity, program change data and lots more. The BIT 99 offers a number of commands unique for synthesizers of its kind, many of them normally found only in Master Keyboards! But more about that later!

The BIT 99 can act not only as a transmitter of MIDI data, but also as a receiver. It is then 'fed' with MIDI data from the MIDI OUT facility of another synthesizer, or a sequencer, or a computer with a MIDI Interface and suitable software. A further possibility is using a MIDI Master Keyboard such as the **BIT-MASTER-KEYBOARD**, to control the BIT 99 along with a number of other synthesizers and/or Expanders.

Don't worry: you don't need to know anything at all about the technical side of MIDI! You only need to manage to plug a standard *5-pin DIN cable into two instruments; MIDI will do the rest for you!*

In every MIDI system there is a transmitter - the instrument that decides just what should be played by whom. This can - as we said above - be the BIT 99 or another keyboard, but also a computer or sequencer. The transmitter sends commands to a set of other connected instruments. In order for each instrument to recognize only its 'own' commands - and noone else's - the MIDI data is provided with a code, or label: the MIDI channel number. Every receiving instrument is allocated a specific channel, so that it picks out only the data 'labelled' with its own code, and ignores all the rest.

The transmitting instrument (Master Keyboard or sequencer) sends data over its "MIDI OUT" socket. This should be connected to the "MIDI IN" socket on the receiving instrument. To transmit data to several instruments at a time, the "MIDI THRU" socket on the first receiving instrument should be connected to the "MIDI IN" socket on the second; the "MIDI THRU" socket of the second with the "MIDI IN" on the third etc etc. A number of instruments can be linked in this way without needing batteries of connections on the Master Keyboard, or creating a horrible spiders web of cables!

The major difference between the "MIDI OUT" and "MIDI THRU" connections should be noted: the "MIDI OUT" socket carries only data on events occurring in the instrument itself (e.g. keys pressed), while "MIDI THRU" passes on all data received over the "MIDI IN" connection.

Please note that MIDI standards recommend not exceeding 15 meters (50 feet) in cable length. To ensure trouble-free operation we advise you not to exceed a cable length of 5 meters (15 feet). When linking several instruments in series, chaining more than four instruments may cause minimal delays in transmission of MIDI data - these delays may become audible under some circumstances.

But you will no doubt want to hear the wonderful sounds the **BIT 99** produces too: on the rear panel are two audio connectors, marked "UPPER OUT" (left channel) and "LOWER OUT" (right channel). Connect these with two line inputs on a mixing desk, an instrument amplifier with a good frequency range, or a HiFi system. If you need a mono signal (i.e. when only one channel is available on the mixing desk) connect either one of the audio outputs.

When first switched on, the **BIT 99** sends an audio signal in mono over both output channels. For a stereo signal press the "STEREO OUT" button on the front panel - the LED in te switch will now show stereo output. You can switch back to mono output at any time by pressing the button again.

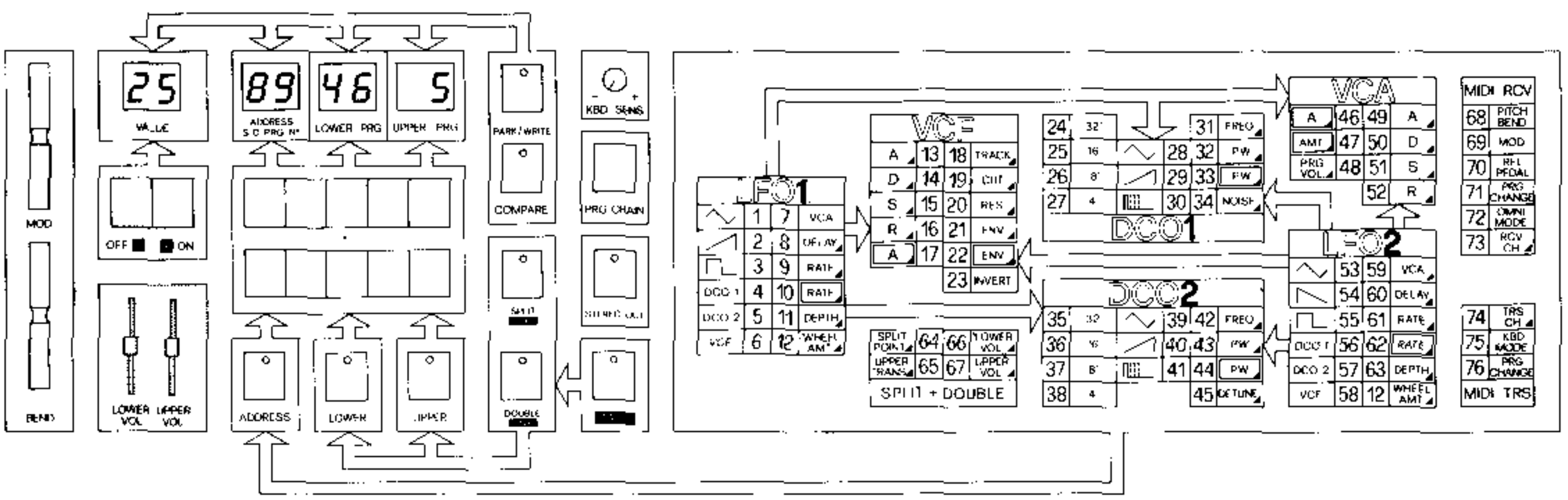
AUDIO CONNECTION

ADJUSTING VOLUME AND PITCH

Volume of the two audio outputs is controlled by two faders, marked "LOWER VOLUME" and "UPPER VOLUME", found at the lower left of the front panel. They control two different sections of the keyboard separately when using Split or Double Mode (more about that later, starting on page 23!). In Normal Mode - when the BIT 99 is used as a 6-voice polyphonic synthesizer - the two faders act as volume controls for their respective outputs. A small tip: set the faders between 3/4 and fully open, and adjust the mixer to this level. This prevents noise, which can develop in the mixing desk through over-amplification, or in other words, optimizes the signal-to-noise ratio.

The overall pitch of the BIT 99 can be tuned to other instruments using the small pot, labelled "PITCH", on the rear panel.

These controls on the front panel of the BIT 99 *influence all* selected programs. You have the opportunity for further manipulation in the form of programmable volume, programmable balance in the Split and Double Modes, as well as programmable detune (detuning the DCOs against each other). The Modulation and Pitch Bend wheels on the left side provide yet another option. In addition, Splits can be transposed, but here again: more about that later!



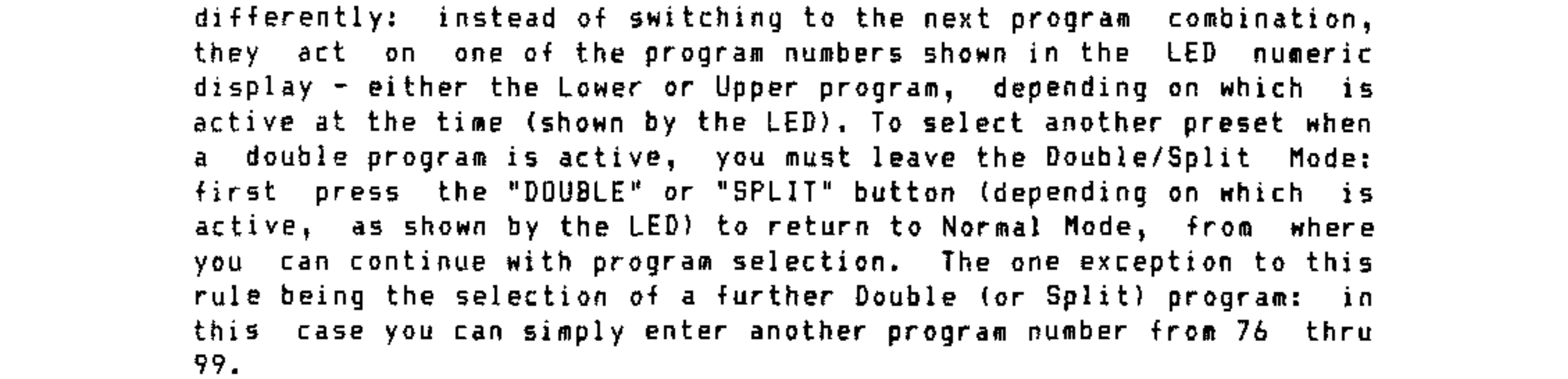
SELECTING PROGRAMS

The BIT 99 has 99 programs (or patches, or presets or whatever other weird and wonderful term might tickle your fancy). These are programmed settings that you can access at the push of a button. If you are working with a Master Keyboard, computer or sequencer that transmits program change commands, you can access these programs via MIDI without pushing any buttons.

To select programs manually, press the "LOWER" button, if its LED is not already on. Now check that none of the LEDs in the "SPLIT", "DOUBLE" or "ADDRESS" buttons are on. Should one of them be on, press that button, so that the LED goes out. Now enter a two digit program number, using the buttons numbered 0-9 under the LED numeric display. The program number will appear in the "LOWER PROG" display, and the program is now active. Another method of selecting programs is by using the "+/ON" and "-/OFF" buttons, which change the program number in the display (and thus the active program) in single steps up or down.

Programs 76 thru 99 have special functions: instead of sound settings they contain information on Splits and Doubles. When one of these is selected, the pre-set combination of a Lower and an Upper program will appear in the LED numeric display as "LOWER PROG" and "UPPER PROG", with the selected program number shown in "S.D.PRG NO". At the same time a LED in either "DOUBLE" or "SPLIT" will show which type of double program is active. We'll be telling you more about Splits and Doubles in a special section later! (See page 23 onwards).

In programs 76 thru 99 the "+/ON" and "-/OFF" buttons act differently: instead of switching to the next program combination, they act on one of the program numbers shown in the LED numeric display - either the Lower or Upper program, depending on which is active at the time (shown by the LED). To select another preset when a double program is active, you must leave the Double/Split Mode: first press the "DOUBLE" or "SPLIT" button (depending on which is active, as shown by the LED) to return to Normal Mode, from where you can continue with program selection. The one exception to this rule being the selection of a further Double (or Split) program: in this case you can simply enter another program number from 76 thru 99.



IN GENERAL: HOW TO USE THE PARAMETER CONTROLS

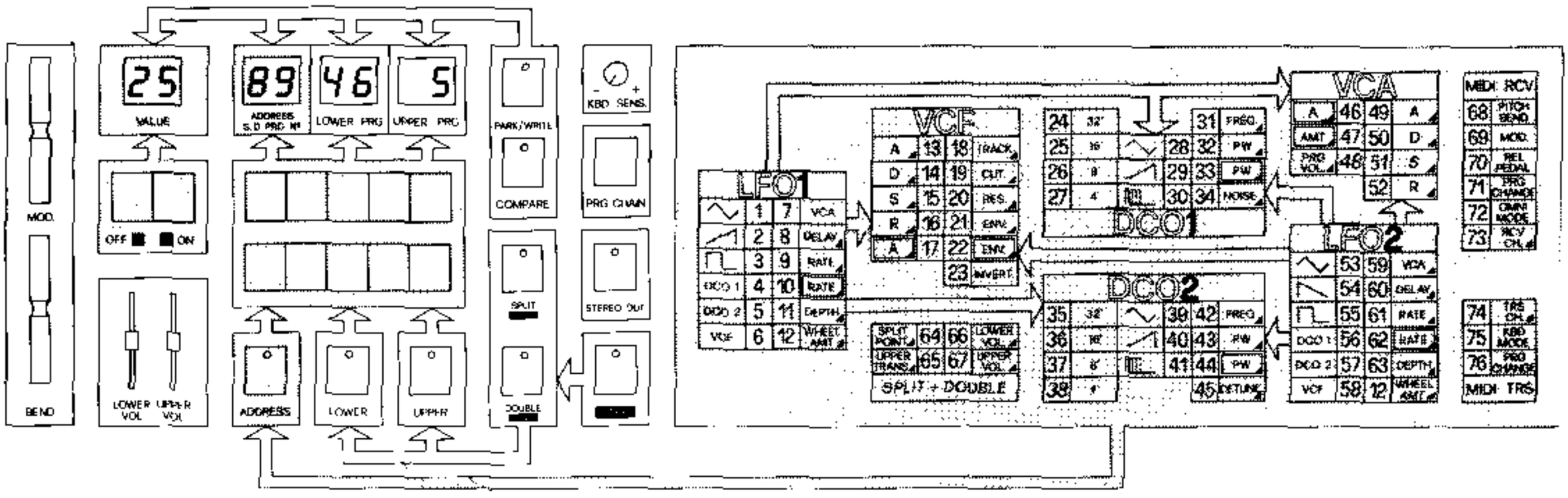
Unlike instruments of older design, the BIT 99 does not use a vast array of knobs and buttons, pots and switches to alter and set the various parameters which together shape and form a sound. Instead, modern microprocessor technology takes the heartbreak out of finding a path through the electronic jungle! The right half of the front panel is a diagram showing all the programmable parameters: the numbers shown in blue are the so-called "addresses" for each individual parameter. For example: the Attack/Sustain/Decay/Release parameters of the VCF are at addresses 13, 14, 15 and 16.

Having selected a program as described above, you can now examine and/or change its parameters. It's really simple: first press the "ADDRESS" button (its LED will go on, showing that the BIT 99 is now in Edit-Mode). Using the number buttons enter the two digit number (= address) for the parameter you want to look at and/or change. The present value of the parameter will appear in the LED numeric display "VALUE", and can be changed using the "+/ON" and "-/OFF" buttons. Pressing the buttons briefly changes the value one step up or down; the value shown in the display will start to "run" if the button is held down. The effect of any change made can be listened to immediately.

Pedestrians - and anyone else who likes their hands free - will find a socket marked "PROGRAM ADVANCE" on the rear panel, where they can connect a foot switch to work exactly like (and parallel to) the "+/ON" button.

To look at or change any other parameters, simply enter a new address using the number buttons. Any changed value remains in temporary storage as it was last displayed. ATTENTION: if the "ADDRESS" button is pressed again, you will leave the Edit-Mode. Any parameter changes made will remain in temporary storage, but will not affect the programmed patches at this time. Further changes can be made by pressing the "ADDRESS" button again, and proceeding as above. However, the temporary storage will be over-written with new data when a new program is selected by pressing "LOWER" and entering the two digit program number (this includes re-selection of the current program). A later section will explain how to store newly created or changed programs permanently in the BIT 99 memory. (See page 21).

Depending on the type of parameter addressed, the display will show either a numeric value or on/off status. Numeric values usually range from 0 thru 63, and are indicated on the front panel diagram by a small triangle in the lower righthand corner of their name field (e.g. Delay, Rate, Track). Some parameters can be either active (appears in the display as "1") or inactive (which appears as "0").



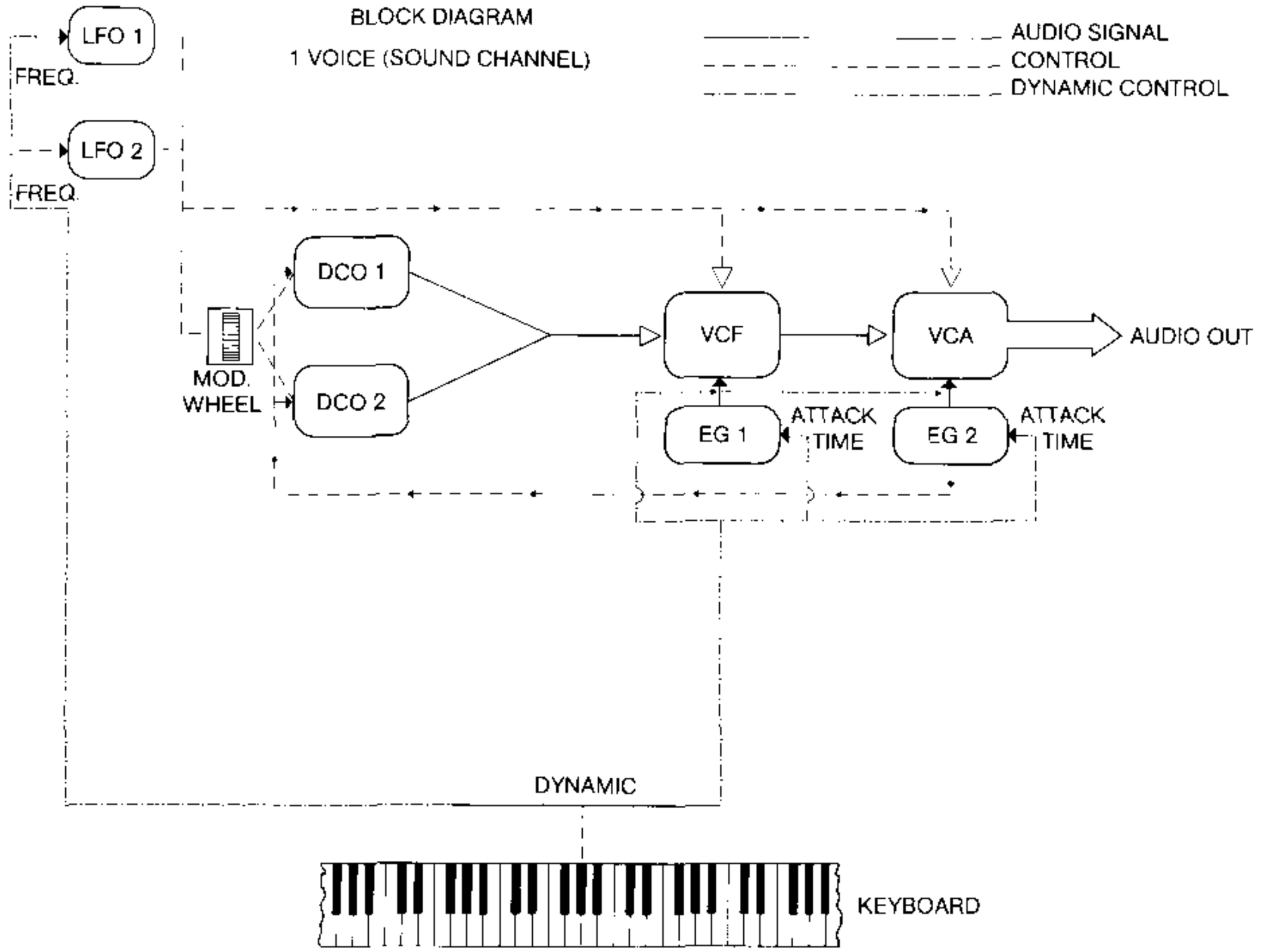
SOUND GENERATION AND CONTROL PARAMETERS

The BIT 99 is a truly 'complete' synthesizer - i.e. all parameters necessary for programming a wide variety of different sounds are at your disposal. The last section showed how to access and change parameter values. Let's take a closer look at these parameters now. You will need a good grasp of the basic principles of sound generation with analog synthesizers.

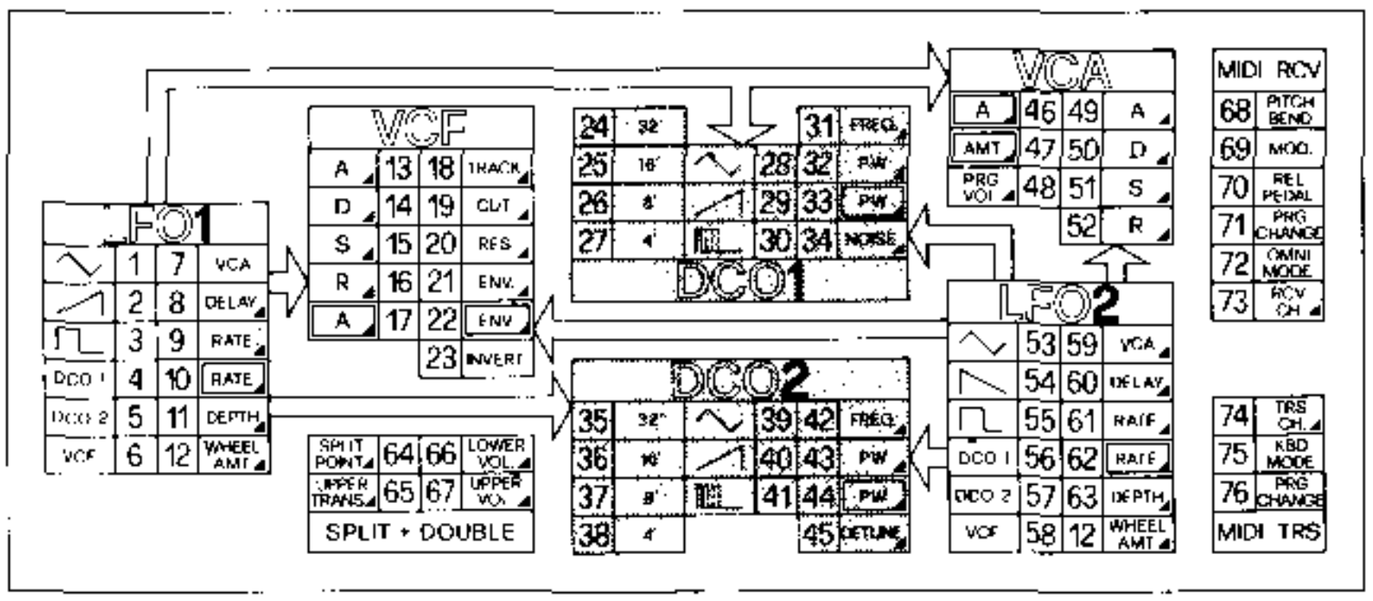
It's best to go right ahead and try everything out! Nothing permanently tragic can happen: as long as the "MEMORY PROTECT" switch on the rear panel is in the "ON" position, you cannot write over, or erase the internal program storage. Any changes made can be listened to immediately - and if it becomes too awful, just enter the program number again to hear the sound in its original state! For a quick 'reset' of all values to power up conditions, hit the "TAPE" button twice.

In addition to the two types of parameters you've already met (those with numeric values and those with on/off status), we now encounter a new 'species' of numeric value: these are the parameters shown on the front panel diagram boxed in double lines ("RATE", "AMT" etc). These are the parameters governing the key velocity (touch sensitivity, whatever) influence on the sound programmed. To use these parameters when the **BIT 99** is controlled by another keyboard, the MIDI data must obviously contain key velocity information, i.e. originate in either a touch sensitive keyboard or a computer program/sequencer providing this information. Non-touch sensitive instruments always transmit an average value (= medium key velocity).

In the following detailed sections on each parameter we will refer to each parameter as it appears on the front panel diagram, followed by its parameter number (address) in brackets. Some parameters occur twice (e.g. "DELAY" in LFO1 and LFO2), but as their functions are identical, they are only explained once.



THE DCOs (DIGITALLY CONTROLLED OSCILLATORS)



■ 32' (24 and 35), 16' (25 and 36), 8' (26 and 37), 4' (27 and 38)

The base frequency (octave) for DCO1 and DCO2. These are on/off values, which can be set to either "1" (on) or "0" (off). Only one of the four base values for a DCO can be active at any time - the other three are automatically set to "0".

■ TRIANGLE (28 and 39), SAWTOOTH (29 and 40), VARIABLE PULSE (30 and 41)

The waveforms available for DCO1 and DCO2: triangle, sawtooth and variable pulse. These are also on/off values, but here more than one waveform at a time can be active for each DCO, creating complex combined waveforms.

■ FREQ (31 and 42)

The 'fine tuning' for the DCOs in 12 semi-tone steps (0 thru 11). This transposes the base frequency (octave) up by the number of semi-tones entered.

■ PW (32 and 43)

Controls the pulse width: only active when the DCO waveform is set to "Pulse" (parameter 30 or 41 = "1"). The pulse width can be set from 3% (value "0") thru 50% (value "16", a perfect square wave) to 97% (value "30"). The percentages given represent the duty cycle of the pulse wave. A change in the pulse width alters the harmonics content dramatically, and therefore the sound.

Settings above the value "16" (50%) are mirror images of lower values: the 80% setting is the equivalent of the 20% setting and sounds identical, but its phase is reversed. Settings over 50% are of particularly useful when key velocity information is used to control the pulse width dynamically (see next section).

■ **PW (Dynamic, 33 and 44)**

Determines the influence of velocity on the pulse width. Values between "0" (minimum, no effect) and "63" (maximum) can be programmed. The values programmed under parameters 32 and 43 (above) are the maximum pulse width values, reached at maximum key velocity.

At maximum setting ("63") the following results: at pulse width settings up to 50%, the wave form will widen with increasing key velocity (becoming a square wave at maximum key velocity). Pulse width settings over 50% cause the wave form to narrow with increasing key velocity (reaching the maximum setting and thus the minimum width at maximum key velocity).

If a velocity value lower than "63" is set, the pulse width values of parameters 32 and 43 remain the maximum values reached at maximum key velocity, but the downward range is narrowed. Here's an example (it is getting rather complicated, isn't it?).

Let's assume that the pulse width parameter 32 is set to "16" (50%), and parameter 33 set to "63". Depending on the key velocity (i.e. how hard the keys are hit), the resulting pulse width will range between 3% (low velocity) and 50% (high velocity). Changing the parameter 33 value to "40" results in a pulse width between 20% and 50%, again according to key velocity. Setting the value of parameter 33 to "0" gives a pulse width independent of key velocity, between 50% and 50% - i.e. it remains the same!

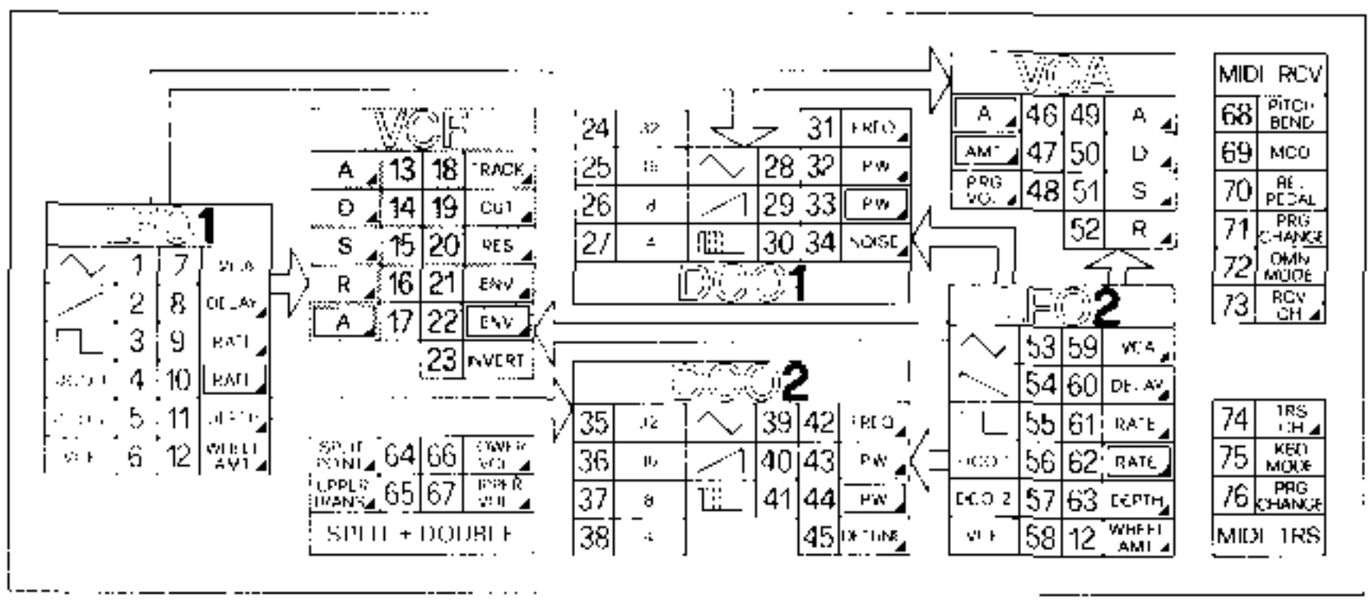
■ **NOISE (34, DCO1 section)**

Noise, a further source of sound generation, set between "0" (blissful silence) and "63" (thundering hiss), allows the programming of hisses, murmurs and rustles for such romantic effects as autumn leaves and waves breaking on the beach. More useful than it might seem at first glance: try it out!

■ **DETUNE (45, DCO2 only)**

One of the best loved features on any synthesizer: allows slight detuning of DCO2 against DCO1 in 64 miniscule (!) steps. Varying degrees of detune produce a multitude of effects, from rotating speakers over chorus to phasing. Set too high, it just sounds out of tune.... again, you have to try it!

THE VCF (VOLTAGE CONTROLLED FILTER)



The signal generated by the oscillators (DCO1 and DCO2) is still in its 'raw' form and must first be filtered in order to shape a more characteristic sound. The waveforms chosen in the oscillators have a certain harmonic content, which can be altered or even removed entirely by the filter.

The filter action is not static, but dynamic: its characteristics can be made to change over a period of time. The envelope generator (sometimes referred to as EG) is responsible for this, through its control over **A**ttack, **D**ecay, **S**ustain and **R**elease (the latter group *leading to its other common name:* ***ADSR***). *Usually the attack and* release phases come into play when a key is pressed and released. This is no different in the BIT 99 but, in addition, when controlled by MIDI data the key may be pressed on another keyboard, or a sequencer/computer can "hit the key".

■ ATTACK (13)

Determines the time taken after pressing a key for the filter to reach its greatest opening. ("0" = fast, "63" = slow).

■ DECAY (14)

Determines the time taken after the filter reaches its greatest opening for the filter to return to the opening set under "SUSTAIN". ("0" = fast, "63" = slow).

■ SUSTAIN (15)

The filter opening to be maintained following the decay phase until the key is released. ("0" = filter closed, "63" = filter open).

■ RELEASE (16)

Determines the time taken after the key is released for the filter to close completely. ("0" = fast, "63" = slow).

■ **ATTACK (Dynamic, 17)**

The attack time set under (13) can be modified further by key velocity. The harder a key is hit, the shorter the attack time will be; the setting on parameter (17) determines the extent of modification ("0" = no effect, "63" = maximum). The value programmed under (13) determines the maximum attack time at minimum key velocity.

■ **TRACKING (18)**

Depending on the value set, the filter is opened more the higher the note played lies on the keyboard. This effect is inactive at "0"; the average value of "30" is a pretty good imitation of this phenomenon found in natural instruments; at the maximum of "63" the tracking effect is most clearly audible.

■ **CUT (19)**

The cut off frequency is a frequency between 0 and 20 kHz above which the filter rolls off (i.e. 'cuts' the frequencies off, hence the name). In the above we called this the 'filter opening'. Lower frequencies pass unchanged. As the overtones or harmonics are always multiples of the base frequency of any note, their suppression can alter a sound considerably. At the lowest value ("0") all frequencies above 0 Hz are cut off, or in other words: the filter lets nothing through! So if it should ever happen that in a certain setting no sound comes through, you should check whether the cut off frequency has been set too low by mistake! At the highest value ("63") only frequencies over 20 kHz are suppressed, and as these are inaudible, the filter has no effect. The cut off frequency sets the maximum filter opening; it is modified by the envelope generator (= reaches its maximum at the ATTACK speed, goes to the SUSTAIN level at the DECAY speed, and then falls to "0" at RELEASE speed).

■ **RESONANCE (20)**

The resonance setting (from "0" = no resonance, to "63" = maximum resonance) boosts the frequencies at the cut off frequency. It is a frequency selective amplification, which again modifies the sound. Strong resonance (= values approaching "63") causes the filter to start self-oscillating, which sounds like feedback (which is in fact exactly what it is!).

■ **ENVELOPE (21)**

The VCF envelope generator (ADSR) is always active when a key is hit, but the degree of its influence on the cut off frequency can be controlled by this parameter. ("0" = no ADSR effect, "63" = maximum effect).

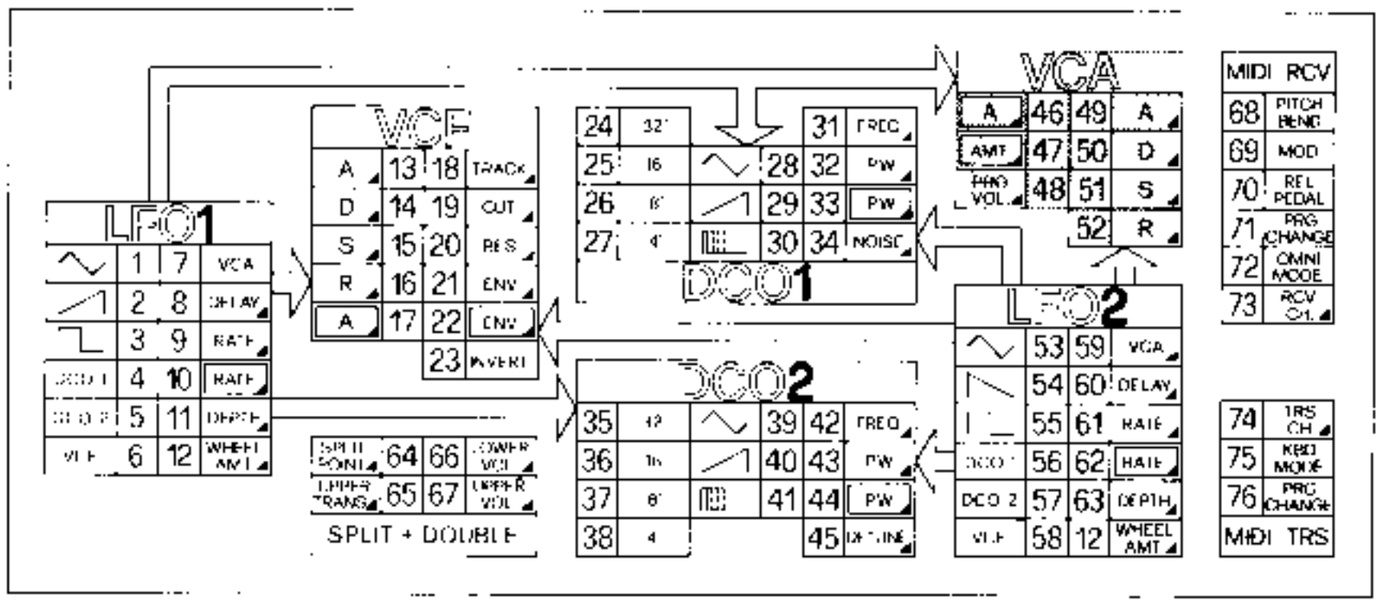
■ **ENVELOPE (Dynamic, 22)**

The value set at parameter (21) can be modified further by key velocity, so that a soft touch on the keys results in only a slight effect, while a firm touch will cause a greater effect. Parameter (22) determines the amount of modification: here too a value of "0" = no effect of key velocity, and value "63" = maximum effect of key velocity.

■ **INVERT (23)**

This last VCF parameter is an on/off switch ("1" = on, "0" = off), which exactly inverts the course of the ADSR envelope generator, and can be used for all kinds of crazy effects!

THE VCA (VOLTAGE CONTROLLED AMPLIFIER



After the sound has been generated and filtered, we want to add some volume dynamics: we do this using the voltage controlled amplifier (VCA). The attack, decay, sustain and release characteristics that we encountered in the VCF turn up here too - which is hardly surprising, seeing as the VCA is composed mainly of an ADSR!

■ **ATTACK (49)**

Determines the time taken for the VCA to reach its maximum volume ("0" = fast, "63" = slow).

■ **DECAY (50)**

Determines the time taken after the VCA reaches its maximum volume for the VCA to return to the volume set under "SUSTAIN". ("0" = fast, "63" = slow).

■ **SUSTAIN (51)**

The volume maintained following the decay phase until the the key is released.

■ **RELEASE (52)**

Determines the time taken after the key is released for the volume to return to zero. ("0" = fast, "63" = slow).

■ ATTACK (Dynamic, 46)

The VCA attack time set under (49) can also be modified by key velocity. The harder a key is hit, the shorter the attack time will be; the setting of parameter (46) determines the extent of modification ("0" = no effect, "63" = maximum). Again, the setting of parameter (49) is the longest attack time reached at minimum key velocity.

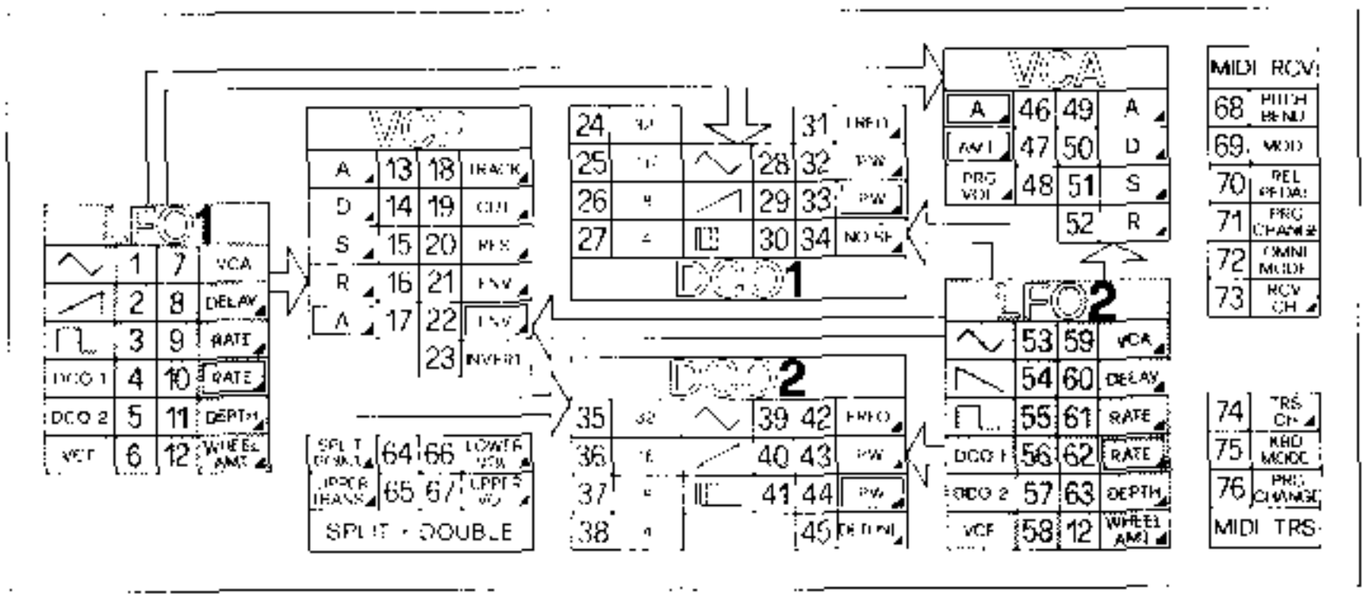
■ AMOUNT (Dynamic, 47)

The amount of ADSR influence on the VCA (parameters 49 thru 52 above) can also be controlled by key velocity. A value of "0" at parameter (47) allows no modification by key velocity; a value of "63" causes maximum modification.

■ PROGRAMMABLE VOLUME (48)

Depending on the programmed filter opening and/or the duration of each note (controlled by the filter and amplifier ADSRs), different patches can have different volume levels. The different levels are at times purely subjective, that is to say they only seem louder; some patches are in fact louder! Whatever the reason might be: volume level differences when changing from one program to the next are often annoying. The programmable volume exists to ajust the volume of different sounds to the same level. The scale ranges from "0" = no volume, to "63" = loud (!). WATCH OUT: a "0" setting really means no volume, so this is another potential trap when programming sounds!

THE LFOs (LOW FREQUENCY OSCILLATORS)



We already encountered oscillators, where the DCOs were used to generate the basic frequency of a sound; in contrast the LFOs generally operate at subsonic frequencies, that is below the audible frequency range. They generate no sounds, but are instead used for 'rhythmic modification' of other components such as the DCOs, VCF and VCA. As the exceptionally wide control range spans frequencies from 0.5 Hz (1 cycle per 2 seconds) to 250 Hz (250 cycles per second), the LFOs can even be used for certain FM (frequency modulation) effects.

The parameters for both LCOs are again identical, with the minor exception of parameters (2) and (54), which are both sawtooth waveforms - one rising and the other descending.

■ TRIANGLE (1 and 53), SAWTOOTH (2 and 54), SQUARE (3 and 55)

You can choose one of three waveforms per LFO: Triangle, Sawtooth and Square waves. The LFO1 sawtooth wave is ascending, and that of LFO2 is descending. These are on/off values which can be set to "1" (on) or "0" (off). Only one waveform can be active per LFO at any time, the other values are automatically set to "0" (off).

■ DCO1 (4 and 56), DCO2 (5 and 57), VCF (6 and 58), VCA (7 and 59)

These on/off switches determine whether either (or both) LFO should modulate any (or all) of these components (DCO1, DCO2, VCF or VCA). Modulation of a DCO alters its pitch; modulation of the VCF influences the cut off frequency; and modulation of the VCA influences volume.

■ DELAY (8 AND 60)

The LFOs are normally activated the instant a key is pressed. Using the "DELAY" parameter allows you to retard (meaning delay, as you might just have guessed) the LFOs' action; the modulation effect will then build up gradually. In a certain sense, the LFO "DELAY" parameter acts similarly to the envelope generators' "ATTACK" parameter: after pressing a key it takes a pre-programmed length of time ("0" = immediate, "63" = slow) to reach maximum modulation.

■ RATE (9 AND 61)

Here the LFO frequency (rate of oscillation) is set between 0.5 and 250 Hz. ("0" = 0,5 Hz and "63" = 250 Hz). The lower, subsonic values are most often used for modulation; higher frequencies that lie within the audible frequency range are generally saved for special effects (e.g. FM effects).

■ RATE (Dynamic, 10 and 62)

Yet another parameter that can be controlled by key velocity! This time the LFO frequency (LFO "RATE") can be modified according to touch dynamics. The higher the value of parameters (10 and 62) lie, the greater will be the effect of key velocity. The LFO rate remains unaffected when the value is "0"; at a value of "63" a soft touch on the keys results in a low LFO rate, while hammering away like a maniac will get your LFOs spinning at the maximum speed set in the "RATE" parameters (9 and 61)!

■ DEPTH (11 and 63)

Usually, we want only a subtle modulation of those components controlled by the LFOs. With the value programmed in "DEPTH" we can vary the modulation from "0" (no modulation) to "63" (maximum modulation). The effect the LFOs have on the VCF and VCA depends entirely on the above parameters. An additional factor comes into play in conjunction with the DCOs: the modulation wheel on the keyboard. All the amazing details in the very next paragraph: "WHEEL AMOUNT"!!

■ WHEEL AMOUNT (12)

The modulation wheel (or joystick, if you're using a controlling keyboard that happens to be that much fun!) gives you additional control over the LFO modulation of the DCOs. Programming a "WHEEL AMOUNT" value of "0" inactivates the modulation wheel control over the LFO. Values between "1" (low) and "63" (high) determine the proportional share of control that the modulation wheel has over the "DEPTH" parameter.

Starting to sound a bit complicated, isn't it? Something along the lines of touch sensitive parameter A modifying the modulation of parameter B which is controlled, or possibly cut off, by the cross-modulated modification in parameter XYZ, and all this and more is happening dynamically at keyboard velocity .....

It's really not that bad, as you will now see! In the following example we have used a pre-programmed "DEPTH" value of "20".

With a "WHEEL AMOUNT" value of "0" the DCO will in fact be controlled by a modulation depth of "20", and the modulation wheel will have no effect.

We now set the "WHEEL AMOUNT" value to "32" (half its maximum value, or 50%). When the modulation wheel is closed, 50% of the programmed "DEPTH" value (that is 50% of "20" = "10") will be fed to the DCO; and the remaining "DEPTH" range (between "11" and "20") is controlled by opening the modulation wheel.

Now we'll program a "WHEEL AMOUNT" value of "48" (three quarters of its maximum value, or 75%). If the modulation wheel is now closed, 25% of the "DEPTH" value (i.e. "5") reach the DCO, and the range from "6" to "20" (the remaining 75% of the control range) is controlled by the modulation wheel.

The next logical step is setting the "WHEEL AMOUNT" to "63" (or 100%). There will now be no modulation of the DCO when the wheel is fully closed: the entire "DEPTH" range from "0" thru "20" is now under the control of the modulation wheel.

All clear? In that case (but even, or perhaps especially if it isn't), it's time to put all this to a practical test on the real thing!

MIDI CONTROL PARAMETERS

The BIT 99 *can - except for the programming that we have explained* so far - be controlled entirely via MIDI data that can originate in a Master Keyboard, a synthesiser with MIDI output, or a sequencer/computer with a MIDI connection. The BIT 99 itself generates *MIDI data whenever a key is pressed on its keyboard, and* also during program changes, when the Modulation and Pitch Bend wheels are used, if the Sustain Pedal (when connected) is pressed etc. etc. MIDI itself is in fact nothing other than a sort of electronic 'language' that different instruments of different brands have in common, allowing them to communicate with one another. When a key is hit on the keyboard for example, the following information is sent out over the keyboard's MIDI-OUT socket: note number "nn" has just been hit at key velocity "vv". As the key is released, the further information that note "nn" has just been released is transmitted.

This information, or data, can be processed by every instrument with a MIDI interface, such as the BIT 99 or the BIT 01 EXPANDER. When the Note On Event data arrives at the receiving instrument it says *to itself: aha! That's interesting, let's play this note. In its* computer it then 'presses' that note with the appropriate key velocity and just look: the self-same note pressed on the keyboard emerges from the receiving instrument too! All this happens at *incredible speed - faster than the ear can hear, you might say!*

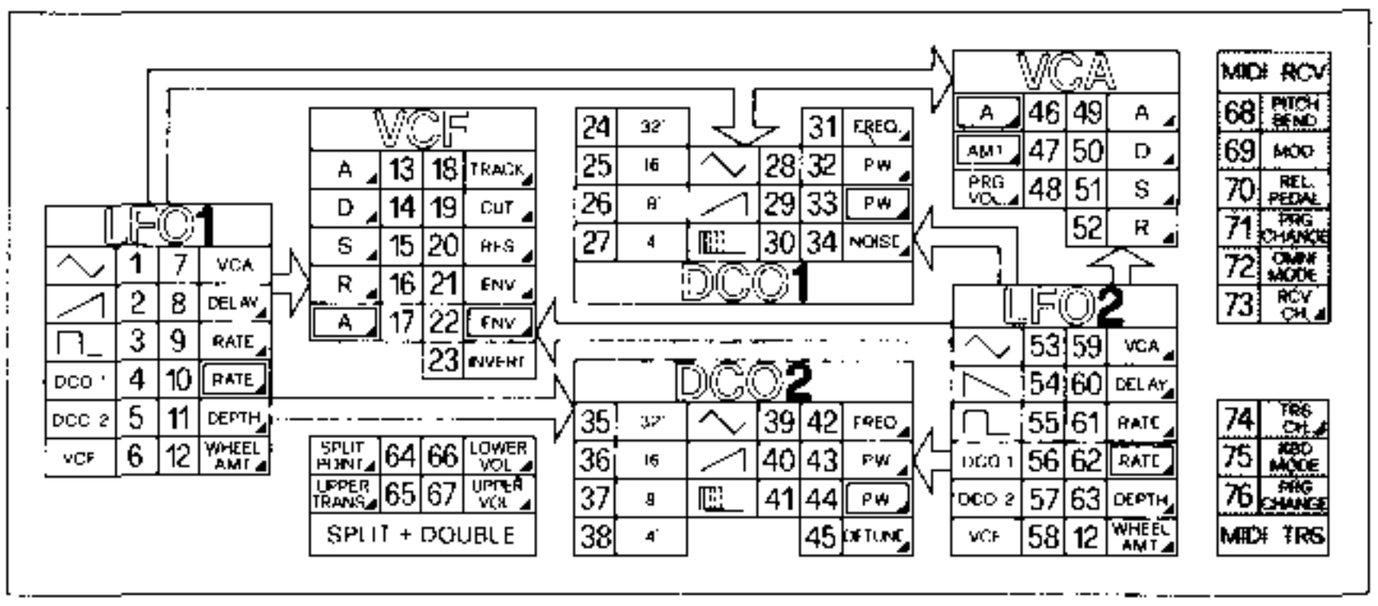
So MIDI serves, as we've seen, in data-communication. The one thing that MIDI most definitely does not do is provide any instrument with new features or capabilities it did not already possess. So although a synthesizer without touch sensitivity can receive MIDI key velocity data, it cannot process the information and will therefore simply ignore it. (Though it will process, and act on the rest of the data transmitted).

In a MIDI system built up around a MIDI sequencer (or a computer with a MIDI sequencer program), different instruments are fed with different data. The drum machine for example, needs other information than the synthesizer playing the bass-line, and a solo instrument uses different information again. So that each instrument does not need its own cable with a separate output from the sequencer, the MIDI data is transmitted with a MIDI channel number - a kind of 'address label'. Now all the MIDI data can be transmitted over one single cable, and each of the instruments hooked up only *picks out that MIDI data labelled with its own 'address'.* The drum machine might take the data on MIDI channel 1, the bass synthesizer only uses data on MIDI channel 2, the solo keyboard picks up the data on MIDI channel 3 etc., etc. There are a total of 16 MIDI channels available. *Any data not relevant for a particular* instrument is simply ignored. The BIT 99 has the parameters "RECEIVE CHANNEL" and "SEND CHANNEL", where you can choose which MIDI channel it should receive or send data through.

This selection of data from a particular MIDI channel can also be suppressed, so that the BIT 99 reacts to all the incoming MIDI commands that it recognises and can carry out. This is done by using a control parameter to select MIDI OMNI Mode.

The parameters in the following section, which all govern MIDI functions, are slightly different from the other parameters that you have encountered so far: they are global definitions. That is, they are not stored as part of a program (or preset), but are valid for all BIT 99 programs. They are stored in a specially reserved memory area, and remain in force until changed (re-programmed), or until the next reset of the BIT 99 to power up defaults. (Either by actually switching on, or when the "TAPE" button is pressed twice).

The MIDI parameters are divided into two logical blocks: those for the reception of incoming MIDI data ("MIDI RCV" in the block diagram), and those for the transmission of MIDI data to other instruments ("MIDI TRS" in the diagram).



■ PITCH BEND (68)

This parameter determines whether pitch bend information from the controlling instrument will be processed ("1") or ignored ("0") by the BIT 99. When value "1" is set, moving the pitch wheel on the master keyboard will change the pitch of notes played on the BIT 99.

■ MODULATION (69)

The same applies to modulation wheel information: the BIT 99 will process the information when a value of "1" is set, and ignore it if the value is "0". Please note that the parameter "WHEEL AMOUNT" (12), described in the LFO section, can only be used when the "MODULATION" parameter is set to "1"!

■ RELEASE PEDAL (70)

Here you can determine whether the BIT 99 should process ("1"), or ignore ("0") release pedal information from the controlling instrument. The BIT 99 also has its own release pedal input. If MIDI release pedal information (from a master keyboard etc.) is being processed (value "1"), the BIT 99's release pedal is inactivated. Whenever value "0" is programmed, any MIDI release pedal data is ignored, and instead the BIT 99's release pedal is active. In both cases, MIDI release pedal data will be generated by the BIT 99, and transmitted through "MIDI OUT" when the pedal connected to the rear panel is used.

■ PROGRAM CHANGE (71)

You can change programs on the BIT 99 automatically via MIDI. If you want program changes on the controlling instrument (or sequencer, or computer) to have the same effect on the BIT 99, set this parameter to "1". This means that selecting program number XY on the controlling instrument will also automatically select program XY on the BIT 99. If you don't want to do this, just set parameter (71) to "0"!

This automatic program change function is particularly useful both when a series of program changes are needed in one song (e.g. different sounds for the intro, verse and chorus); and also where the BIT 99 is 'slaved' to the controlling instrument, allowing additional doubling effects (e.g. extending your modest string quartet to a full-fledged string section of a not-so-modest symphony orchestra!). In whichever case you ought to make sure that programs with the same number are musically compatible. Thus if program 38 on the controlling synthesizer is Mantovani-type strings, program 38 on the BIT 99 should most probably not be a pneumatic drill!

Present MIDI standards allow a total of 128 programs. Every manufacturer operates with slightly different configurations of the program selection keys: if, for example, a YAMAHA DX-7™ is controlling the BIT 99, the internal programs numbered 1-32 correspond to programs 1-32 on the BIT 99, while cartridge programs 1-32 correspond to BIT 99 programs 33-64.

One of the special features of the BIT 99 is that programs 76 thru 99 contain not sounds, but pre-programmed Split and Double program combinations. These can be accessed in exactly the same way as 'normal' programs, by selecting a program number between 76 and 99 on the Master Keyboard.

■ OMNI MODE (72)

Here you can choose whether the BIT 99 should react to all incoming MIDI commands ("1", OMNI ON), or only process data from one selected MIDI channel ("0", OMNI OFF). Selecting a particular channel is dealt with in the next section!

■ RECEIVE CHANNEL (73)

If you have programmed a "0" value in the "OMNI MODE" parameter (i.e. OMNI OFF), you can now determine in "RECEIVE CHANNEL" which of the MIDI channels 1-16 contains control data for the BIT 99. The controlling instrument must obviously transmit these commands over the appropriate channel, in order for the BIT 99 to execute them! This feature lets several instruments receive different MIDI data over one cable, using one to play the bass line, another the lead or solo, and one (or possibly several) others to provide backing - all using different programs with different commands from different sequencer tracks.

■ TRANSMIT CHANNEL (74)

Just as the "RECEIVE CHANNEL" parameter allows you to determine which MIDI channel contains data for the BIT 99, this parameter allows you to select which of the 16 MIDI channels the BIT 99 should use to transmit MIDI data to another synthesizer or expander. As we saw above, MIDI data is sent through the "MIDI OUT" socket on the rear panel, and read in through the "MIDI IN" socket of the other instrument. As you might expect, it's important that the the receiving instrument is set to the same MIDI channel selected at the "TRANSMIT CHANNEL" parameter of the BIT 99. Alternatively the receiving instrument can be operating in Omni Mode, where all incoming MIDI data is processed, regardless of channel number.

■ KEYBOARD MODE (75)

The BIT 99 is equipped with a very useful feature found, as a rule, only in Master Keyboards : the "KEYBOARD MODE" parameter enables a MIDI-Keyboard-Split (not to be confused with the 'normal' keyboard split, which we'll be explaining in detail a bit further on!). When "KEYBOARD MODE" is on (Value "1"), key information from each half of the keyboard is transmitted over two different MIDI channels. The key information from the Lower section of the keyboard goes out over the channel selected in "TRANSMIT CHANNEL", also referred to as 'base channel', while key information from the Upper section is transmitted on the base channel plus one. That is to say that if you've chosen MIDI channel 6 in parameter (74), Lower section data will be transmitted over that channel, and Upper section data on channel 7. MIDI channel numbers wrap around automatically from 16 back to 1. "KEYBOARD MODE" allows you to play the BIT 99 just as usual (with one sound over the whole keyboard), while controlling two instruments (with different sounds) with each half of the keyboard.

The value last programmed under "SPLIT POINT" (parameter 64, see detailed description page24) determines where the keyboard will be divided. That is, it sets which key (numbered 1 thru 61) will be the first key of the Upper section (the right hand half of the keyboard). The BIT 99 sets this value to "25" when switched on, corresponding to middle C. The Upper section can be transposed against the Lower section using parameter 65 "UPPER TRANSPOSE" (see page 24). Here again, only the notes transmitted via MIDI are affected, not those played by the BIT 99 itself!

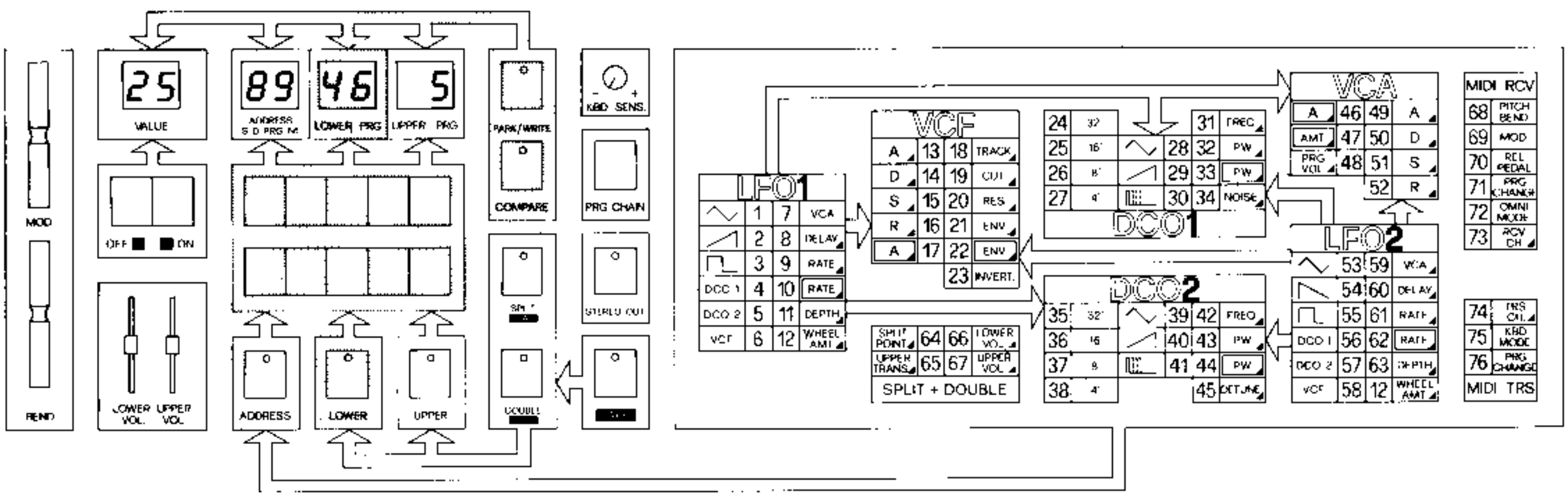
Changing the "KEYBOARD MODE" value from "0" to "1" generates an "Omni Mode Off, All Notes Off" MIDI command. This ensures that the receiving instrument is not in Omni Mode (i.e. is not receiving data on all MIDI channels, which would make no sense when using the MIDI split!). When the "KEYBOARD MODE" value is switched back to "0", an "Omni Mode On, All Notes Off" MIDI command is transmitted.

The "KEYBOARD MODE" feature, or MIDI split, can also be used in combination with a 'real' Split on the BIT 99. In Split Mode a "KEYBOARD MODE" value of "0" allows the BIT 99 to use two different programs for each section of the keyboard, while transmitting note values via MIDI as though no Split had taken place.

■ PROGRAM CHANGE 76

Parameter 71 allowed you to decide whether to implement program change data from another instrument received via MIDI IN - here at Parameter 76 you determine whether the BIT 99 should pass its own program change data via MIDI OUT to other instruments ("0"), or not ("1"). Caution: for once we have a value where "0" means yes, and "1" *no* - the reverse of the rule! The reason behind this is that we can also program a "PROGRAM CHANGE" value of "2" - whereupon entering a program number will have no effect on the BIT 99. What will happen is that the program change data will be transmitted via MIDI OUT to any connected instruments! The "LOWER PROG" display will show the entered program number briefly, before returning to the original program number. This gives you 'remote control' over program selection in instruments connected via MIDI!

STORING PROGRAMS



So far we've covered altering and constructing sounds in the Edit Mode. This happens in a temporary 'working storage area'; all changes disappearing as soon as a new program is selected. We will now deal with the permanent storage of patches in the program memory of the BIT 99. When storing a program under a certain program number, you should always keep in mind that the original content of this memory location will be over-written, i.e. lost.

In an earlier section we mentioned the "MEMORY PROTECT" switch on the rear panel of the BIT 99: its function is to protect the memory from accidental over-writing or erasure. To store a new or changed program, the "MEMORY PROTECT" switch must be in the "OFF" position (memory not protected), as it is otherwise impossible to write to the memory! We advise you to return the switch to the "ON" position (memory protected) immediately after storing new data, to prevent accidents and subsequent nasty surprises!

We will first look at memory locations 1-75, those containing program information (and not Split or Double combinations, which will be dealt with in a later section, see page 24).

Programs can only be stored when the BIT 99 is in the Lower Mode ("LOWER" LED on). For actual storage we use the "PARK/WRITE" button, but let's proceed step by step:

1) Put the BIT 99 in Lower Mode ("LOWER" LED on), after checking that the rear panel "MEMORY PROTECT" switch is "OFF".

2) Press the "PARK/WRITE" button; its LED will come on.

3) Select the program number you want to store your new sound under (if the correct number is not already shown in the display), first pressing "LOWER" and then the number buttons.

4) Play a couple of notes to check, for safety's sake, that the sound is exactly as you want it.

5) Now press the "PARK/WRITE" button a second time. The LED will go out, and your new sound is now stored!

6) Return the "MEMORY PROTECT" switch on the rear panel to the "ON" position.

COPYING PROGRAMS WITHIN MEMORY

Thanks to step 3) above, it is also possible to move programs unchanged to other locations (program numbers) in memory. This is especially useful when re-organising memory so that the **BIT 99** programs are musically compatible with those of the controlling instrument, in order to use the MIDI automatic program change function (parameter "PROGRAM CHANGE" = "1").

1) Use the number buttons to select the program you want to move to another memory location. Again, first make sure that the BIT 99 is in Lower Mode, and that the rear panel "MEMORY PROTECT" switch is "OFF".

2) Press the "PARK/WRITE" button, the LED will come on.

3) Now select the new program location (number), again pressing the "LOWER" button and then the number buttons, where the sound should be stored in future.

4) Press the "PARK/WRITE" button again, and the sound is now stored under the new program number. The 'original' is still stored at its initial location, and you can over-write it at any time.

5) As usual, don't forget to turn the "MEMORY PROTECT" switch back "ON"!

"PARKING" AND COMPARING SOUNDS

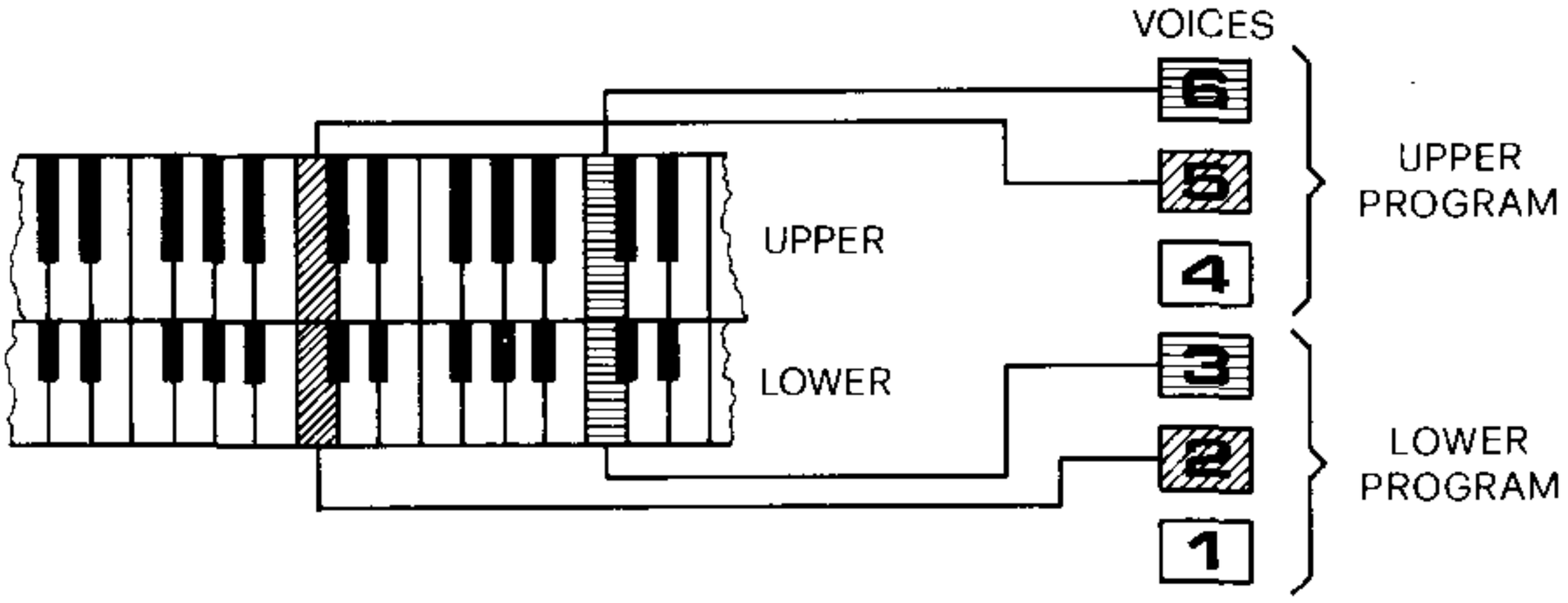
Before permanently storing an altered sound, it can be useful to compare it with the original. The BIT 99 provides the "COMPARE" function specially for this! If you press the "COMPARE" button while changing values, the "ADDRESS" and "VALUE" LEDs will go out, and the original sound can be heard again. To get back to your changed sound, just press "COMPARE" again. You can now continue to work on the sound, or store the new setting as described above.

The "COMPARE" button has another useful function: while storing sounds you press the "PARK/WRITE" button (as you will already know, if you read the above section). This puts your changed sound in the "PARK" area - a temporary storage area - but does not yet affect the permanent program memory (this only occurs when you press "PARK/WRITE" the second time). If you now change your mind, and decide not to store the sound permanently, you can abort the storage process by pressing the "COMPARE" button. Cute, huh?!

TWO SYNTHESIZERS IN ONE: DOUBLE AND SPLIT

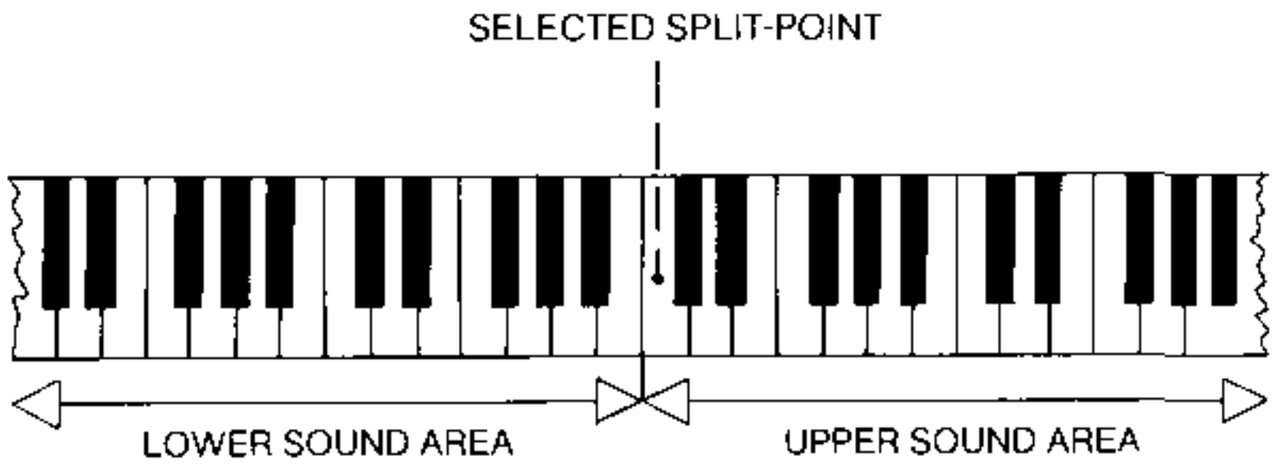
We have touched on the Double and Split Modes many times, without ever explaining exactly what they were. Well, we're now going to do just that, in great detail!

The **DOUBLE MODE** allows two programs (two sounds) to be played simultaneously. You can visualize this as two synthesizers inside the BIT 99, both played by the same keyboard. A key pressed on the keyboard results in a note played by each of these synthesizers in their selected programs. This way you can 'layer' sounds, e.g. have strings and a piano sound on top of each other, and create a much fuller, richer sound. Well, every silver lining has a cloud lurking somewhere on the horizon, and Double Mode is no exception: the BIT 99's voices (of which there are 6) are shared out between the two synthesizers, each getting three voices. The result being that you can no longer use six individual voices (meaning hit six keys at a time), but only three, each key then calling two voices.



These 'imaginary' synthesizers inside the BIT 99 are designated "LOWER" and "UPPER" sections. When the BIT 99 is operating in Stereo Mode (with both audio outputs connected and the front panel stereo LED on), the "LOWER" and "UPPER" voices go through their respective audio outputs. In Mono Mode the Lower and Upper voices are mixed together, and can then be taken from either of the two audio outputs.

In the **SPLIT MODE** the keyboard is divided at a certain point, the so-called Split Point, each section then using a different program. The keys below the Split Point use the Lower program; and the keys from the Split point upward, the Upper program (the names start to make sense, don't they?). Here again the voices are shared out: the Lower section getting three and the Upper section three. And again, each section uses its own audio output when the BIT 99 is in Stereo Mode. The Split Mode is ideal for playing a bass line with the left hand (and the appropriate bass sound), and a solo with the right (with another sound).
The features available in Split Mode can be extended in combination with the MIDI-Split option in the BIT 99's Keyboard Mode (as described above under MIDI parameters).

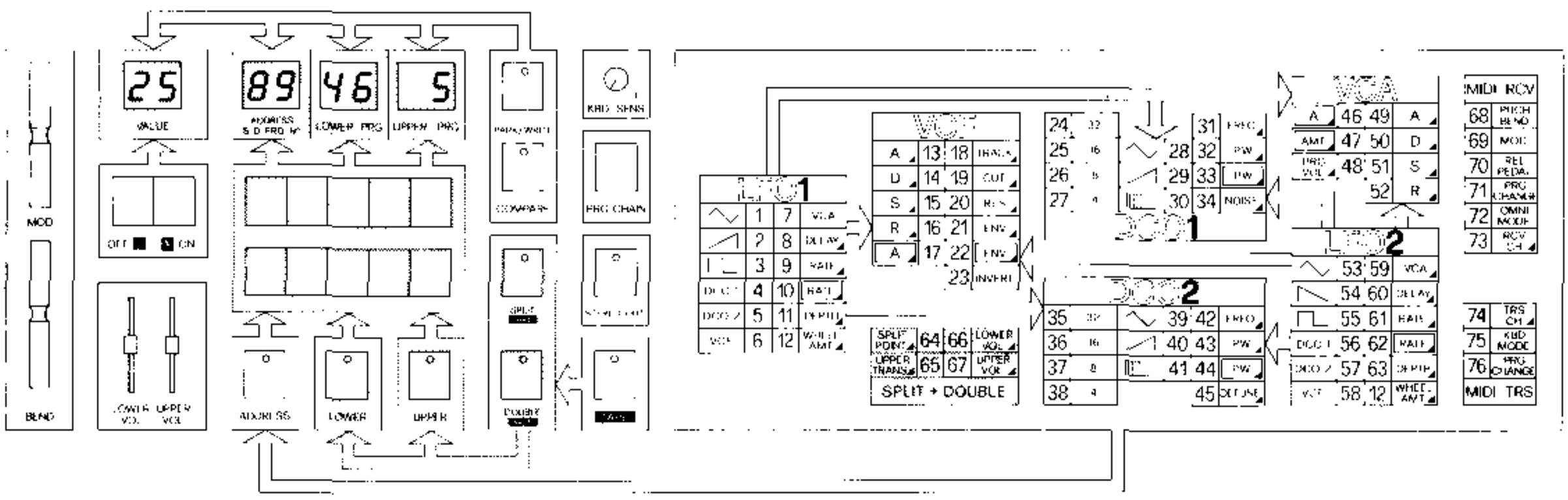


SELECTING DOUBLES AND SPLITS

Really simple: press the "DOUBLE" or "SPLIT" button (whichever you want to use). Its LED will come on, showing the selected Mode. Now press the "LOWER" button and enter the number of the program you want to use on the Lower section. Then press the "UPPER" button, and select the program number for this section. And that's it!

The selected program numbers are now shown in the "LOWER" and "UPPER" LED displays. In addition the "VALUE" display shows the Split Point (when in Split Mode). This is a key number between 1 and 61 (counting from the bottom), and not the MIDI key code (this notice for MIDI-freaks only!). The Split Point can be changed at any time at parameter (64), by entering the key number of the first key in the Upper section.

The two selected programs can also be changed at any time, by first pressing either the "LOWER" or "UPPER" button (if its LED isn't already on), then entering a new program number. You can also use the "+/ON" and "-/OFF" buttons to change program numbers.



PROGRAMMING DOUBLES AND SPLITS

In the sections on selecting and storing programs we noted that program numbers 76 thru 99 had a special purpose: they are used to store Double and Split data. Not only the two program numbers, but also a range of other relevant parameters are stored here (and now come all the details we've been promising you for so long!).

■ SPLIT POINT (64)

This parameter determines which key (numbered from "1" to "61") should be the first key of the Upper program (i.e. the right section of the keyboard). For fairly obvious reasons this parameter is only effective when the keyboard is split .....

■ UPPER TRANSPOSE (65)

When in Split Mode, the pitch of the Upper program can be determined independently. At "SPLIT POINT" (64) you have decided where the Upper section of the keyboard should start; "UPPER TRANSPOSE" (65) now lets you choose the pitch of the lowest key of that section by setting a value from 1 to 61, corresponding to one of the keys on the keyboard. If the pitch of both sections of the keyboard (Lower and Upper) should correspond to the physical location of their keys, the "UPPER TRANSPOSE" value must be the same as the value set under "SPLIT POINT". This somewhat involved explanation meaning simply that if (for example) both "SPLIT POINT" and "UPPER TRANSPOSE" are set to key 25, pitch values will be contiguous over the whole keyboard, with the sound changing at key 25. However, the keyboard can also be configured so that the starting note of both sections corresponds to the lowest note on the keyboard, ("UPPER TRANSPOSE" = "1"); or the starting pitch of the Upper section can be set anywhere from there to "61". (As we said to start off with!).

■ LOWER VOLUME (66), UPPER VOLUME (67)

These two parameters allow you to program a volume balance between the Upper and Lower programs. NOTE: the value programmed at parameter (48) "PROGRAMMABLE VOLUME" is not changed.

To store both program numbers for the Upper and Lower sections, and also the double program parameters (64) thru (67), follow the steps below:

1) Select a program number (free or over-writeable) between 76 and 99.

2) Enter - as described above - the program numbers for the Upper and Lower programs.

3) Program parameters (64) thru (67) as needed.

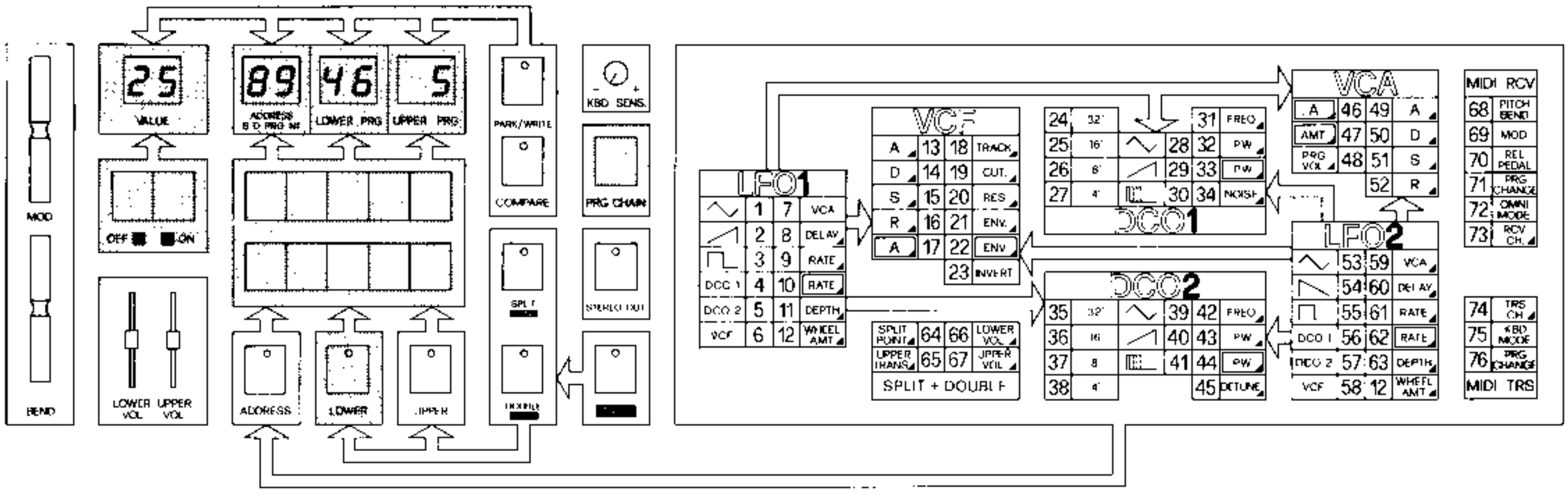
4) Press the "PARK/WRITE" button. Again, the "MEMORY PROTECT" switch on the rear panel must be "OFF" at this point.

5) You now have the chance to think over your choice of programs at 1) above, and choose another program number between 76 and 99 if you should reconsider!

6) Now press "PARK/WRITE" a second time, and your combination program is stored. Returning the "MEMORY PROTECT" switch to the "ON" position will make sure that it stays that way!

SPECIAL NOTE: the "COMPARE" function that you encountered while storing 'normal' programs can not be used when storing combination programs!

PROGRAM CHAIN FUNCTIONS



Now that we've learned how to store programs, let's take a look at a special function on the BIT 99 that allows programs to be called up in a pre-determined order. The "PRG CHAIN" button and the rear panel foot pedal socket marked "PROGRAM ADVANCE" control this function. Visualise the BIT 99 as holding three 'lists', each with your chosen sequence of 33 programs: at the touch of a button you can step through the programs on the 'lists'. Which means that you no longer have to concentrate on remembering program numbers in a certain order, but have your head free for making great music! Which is what it's all about, right?

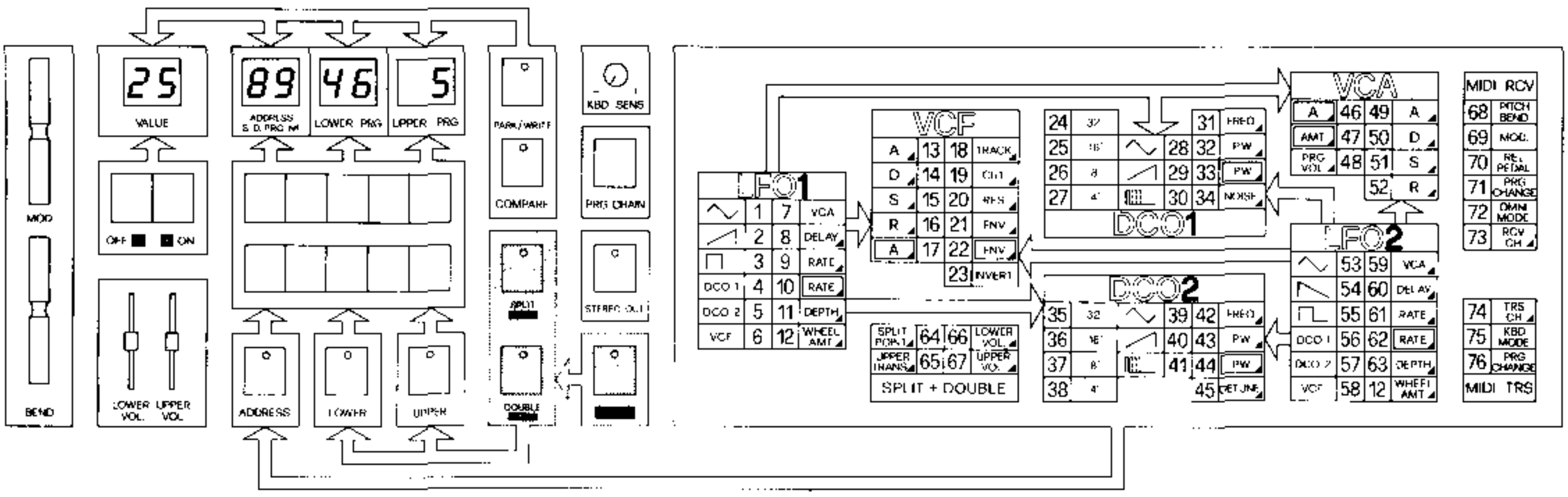
The "VALUE" display shows which of the three available program chains (containing 33 programs each) you've selected: the first time you press the "PRG CHAIN" button the number "1" will appear. Pressing the "PRG CHAIN" button repeatedly will switch through to program chains 2 and 3, and then return you to the Mode last displayed (Lower, Split or Double). Pressing the "LOWER" button will also exit the PROGRAM CHANGE function.

The "ADDRESS" display shows your location within the chain (i.e. the position in the list of 33), and the "LOWER PRG" display shows the program number programmed in this position. Combination programs (numbers 76 thru 99) also indicate the type of double program selected, using the LED in the "SPLIT" or "DOUBLE" button.

Within each program chain, the "+/ON" and "-/OFF" buttons step up and down through the chain, the "ADDRESS" display showing the position within the chain at any time. After the last program in a chain ("33" in the "ADDRESS" display), the BIT 99 will switch to the first program in the next chain. The "PROGRAM ADVANCE" socket on the rear panel allows you to step upwards using a foot pedal, each switch on the foot pedal corresponding to pressing the "+/ON" button once, moving to the next program in the chain.

How to program program chains? (No misprint, could you think of a better way to say it?) Whatever, there's nothing easier! First select a program chain, as described above. Use the number buttons to enter a program number for each position in the chain, using the "+/ON" and "-/OFF" buttons to step forward (or back!) through the chain - done! The program numbers selected are stored in sequence in the BIT 99 until you change them; they are not affected by power up resets.

THE CASSETTE INTERFACE



The **BIT 99**, like almost every modern, microprocessor controlled instrument, is equipped with a cassette interface that allows all stored data (programs!) to be saved onto a cassette, and re-loaded at a later time. In this way the storage capacity of the **BIT 99** can be extended almost indefinitely: you program new sounds, store them on cassettes, and end up with a whole library of sounds that can be re-loaded anytime!

The "TAPE IN" and "TAPE OUT" sockets on the rear panel are used to connect the **BIT 99** to a cassette recorder. We recommend the use of a simple portable recorder, that has a microphone (or AUX) input and a headphone output (PHONES OUT). HiFi recorders with the usual line levels (-10 dB) are as a rule not suitable; however the **BIT 99** comes equipped with a very tolerant cassette interface, which can handle considerable level differences.

There are now various 'Data Recorders' from different manufacturers on the market, specially designed for use with Home Computers (and similar machines, like the **BIT 99**), and usually reasonably priced. Please note that most such machines are designed to use Normal tapes (not the Chrome or Metal types); using the wrong tape can cause data-transmission problems. Talking of cassettes: though the cassette interface does not require HiFi quality (the frequency range required is modest), do use good quality, reliable cassettes, which provide a certain guarantee against tape drop-outs.

You should switch off any noise suppression systems (Dolby, dbx etc., etc.). If the recorder has a tone control (normally a treble cut), this should be fully open. The frequency range of the recorder should be in no way altered. Following these points will ensure trouble-free operation when using the cassette interface.

Exiting Tape Mode resets the **BIT 99** to its power up settings. The "TAPE" button can therefore be used as a 'soft' reset at any time: pressing it twice does the trick!

STORING PROGRAMS ON CASSETTE

Connect the "TAPE IN" socket on the BIT 99 with the headphones output of the cassette recorder, and the "TAPE OUT" socket with the microphone or Aux input on the cassette recorder. Now press the "TAPE" button on the BIT 99. The BIT 99 is now in Tape Mode, shown by the LED in the "TAPE" button, and is ready for action!

1) Switch the cassette recorder to record.

2) Press the "SAVE" button on the BIT 99 (same button as "DOUBLE"). First a sine wave is transmitted, to ensure the correct level in machines with automatic recording level adjustment. Recorders with manual recording level should be adjusted so that the VU meter reads between -5 and 0 VU.

3) After around 20-30 seconds of continuous signal the data will be transmitted.

4) When the data transfer is over, the "SAVE" LED will go out, and in the "UPPER PRG" display a "0" will confirm completion of the storage process. Pressing "TAPE" again will return the BIT 99 to Normal Mode.

LOADING PROGRAMS FROM CASSETTE

1) Loading programs from cassette into the BIT 9901 over-writes the memory: CHECK that the programs now in storage have been saved onto cassette themselves, or are no longer needed!!! Press the "TAPE" button; its LED will come on. As usual, check that the "MEMORY PROTECT" switch is "OFF".

2) Press the "LOAD" button on the BIT 99 (same button as "SPLIT"), and the PLAY button on the cassette recorder. The level on the cassette recorder (if under manual control) should be set to 3/4 of the maximum level. This level varies from machine to machine, so you may have to try several times before finding the ideal level. (Marking this on the cassete recorder saves endless fiddling).

3) At the end of the data transfer the BIT 99 will inform you if it was successful (showing "0" in the "UPPER PRG" display). A value higher than "0" indicates unsuccessful operation, caused by one (or several!) of the following reasons:
   - bad connection; check your connecting cables/jacks/sockets
   - noise suppression active (Dolby, dbx etc.); these must be switched off!
   - wrong level from recorder
   - *drop-outs on the cassette.*

After a successful transfer press the "TAPE" button again to return the BIT 99 to its Normal Mode, with a memory full of new programs from the tape! And yes, as you might already have guessed, we are now going to remind you to put the "MEMORY PROTECT" switch back to "ON" - protecting all these precious programs from sundry disasters!

Errors when reading in data can be caused by phase problems arising in the cassette recorder. Some recorders designed for use with computers have a phase reverse switch (sometimes just labelled PHASE), to eliminate such problems. As far as possible you should use the same recorder for both recording and re-loading program data; any differences in tape head alignment (Azimuth angle) can cause phase problems, these will be eliminated automatically if the same recorder is used for recording and playback.

APPENDIX: MIDI DETAILS AND MIDI EXCLUSIVE-CODES

This section is highly technical and intended only for computer freaks and other such fiends planning on writing their own MIDI software for controlling the BIT 99. Normal mortals, who become faint when confronted with esoteric codes and are frightened of being bitten by all these bytes, can cheerfully ignore the following, and get on with making music!!! Roll over Beethoven!

As for you freaks: before we get down to the dirty details, we'd like to point out that you will shortly be passing the limits of the "MEMORY PROTECT" switch. Program data manipulation via MIDI bypasses this safety device! You are therefore strongly advised to dump all your precious voice data onto cassette BEFORE exploring the weird and wonderful world of MIDI code sequences!

All MIDI data is freely exchangeable between the **BIT 99** and the **BIT 01 EXPANDER**, as they work with the same internal data structure. This includes the system exclusive codes which contain instrument specific Bitmaps. Slight differences occur in areas where the BIT 99 is sending MIDI data, as the BIT 01 can not transmit, but only receive MIDI data. These differences do not affect the actual MIDI data, but do change the Power Up Defaults (see below), where the **BIT 99** can select a Send Channel and send Release Pedal information etc. The Power Up Defaults reflect the fact that the **BIT 99**, in contrast to the **BIT 01 EXPANDER**, is primarily used as a transmitting keyboard. This info mainly for BIT 01 owners wondering about the differences!

The **BIT 99** can receive MIDI commands on any of the 16 MIDI channels. Executable commands are detailed below. The following are the power up defaults:

- MIDI Mode: OMNI ON (Parameter 72 = "1")
- MIDI Channel: 1 (Parameters 73 and 74 = "1")
- Pitch Bend: inactivated (Parameter 68 = "0")
- Modulation: inactivated (Parameter 69 = "0")
- Release Pedal: inactivated (Parameter 70 = "0")
- Program Change: inactivated (Parameters 71 and 76 = "0")
- Keyboard Mode: normal (Parameter 75 = "0")

Power up generates an internal "All Notes Off" command. This command is also generated whenever Omni Mode (Parameter 72) changes status, when Tape Mode is activated, and when changing programs. This eliminates 'hung up' notes, which could occur if the BIT 99 were switched on in the middle of a sequence, or in Split Mode by changing the "Upper Transpose" value while pressing a key. There are various popular keyboards on the market which, for no apparent reason, produce 'random' data (or in other words: electronic garbage!) from time to time - leaving notes 'hanging', because a randomly generated Note On Command was not followed by a Note Off command!

In the following text we use the binary representation of MIDI codes and data where appropriate. Other values are given in hexadecimal; these are identified by a trailing "H" (e.g. 29H).

GENERAL MIDI CODES

STATUS	DATA BYTE(S)		DESCRIPTION
1000 nnnn	0kkk kkkk	0vvv vvvv	Note Off
1001 nnnn	0kkk kkkk	0vvv vvvv	Note On
1011 nnnn	0ccc cccc	0xxx xxxx	Control Change
1100 nnnn	0ppp pppp		Program Change
1110 nnnn	0www wwww	0www wwww	Pitch Wheel Change

VALUES

nnnn = MIDI Channel Number 0-15

kkkk = MIDI Key Number (24H thru 60H, corresponding to key 1-61).
The BIT 99 automatically transposes values outside this range to the nearest octave within the range

vvvv = MIDI Velocity Value. A velocity of 0 in a "Note On" command is the equivalent of a "Note Off" command.

cccc = Control Value or MIDI Controller
01H = Modulation Wheel; value in xxxx
40H = Release Pedal; value in xxxx
7BH = All Notes Off
7CH = Omni Mode Off and All Notes Off
7DH = Omni Mode On and All Notes Off

xxxx = Controller Value
for Modulation Wheel: 00H - 7FH
for Release Pedal: 00H = off, 7FH = on

pppp = Program Number. 00H - 62H call programs 1-99.
63H - 7FH call programs 1-29.

wwww = Pitch Wheel Value; least significant byte (LSB) first, followed by most significant byte (MSB). The BIT 99 uses only the second (MSB) byte. To ensure future compatibility the LSB should be set to 00H.

MIDI SYSTEM EXCLUSIVE CODES

STATUS	DATA BYTE(S)	DESCRIPTION
1111 0000	0010 0101	Manufacturer ID (BIT, 25H)
	0iii nnnn	iii = 002 = BIT 99 (001 = BIT 01) nnnn = MIDI Channel No.
	0ccc cccc	Exclusive data until EOX
	0www wwww 0www wwww ...	cccc = Command wwww = Data Byte(s)
1111 0111		EOX (End System Excl.)

Depending on the command (cccc), a variable number of data bytes (wwww) follow.

cccc = 00H — Activate Split Mode; two data bytes follow.
1st byte wwww: Split Point 00H - 3CH
2nd byte wwww: Upper Transpose 00H - 3CH
(00H - 3CH correspond to keys 1-61)

cccc = 01H — Inactivate Split Mode; no data bytes follow.

cccc = 02H — Activate Double Mode; no data bytes follow.

cccc = 03H — Inactivate Double Mode; no data bytes follow.

cccc = 05H — Lower Program Change (see Upper Program Change below).

cccc = 06H — Upper Program Change; one data byte follows.
wwww = Program Number 00H - 7FH.
00H - 4AH correspond to programs 1-75
4BH - 7FH correspond to programs 1-53

cccc = 07H — Single Program Dump (transfer one program to the BIT 99).
1st byte: Program Number 00H - 62H (for 1-99)
+ 74 bytes wwww for program data or
+ 14 bytes wwww for Double/Split data
(data format see "Bitmaps" below)

cccc = 09H — Request for Single Program Dump. It transfers all data relative to one program of BIT 99 to to another BIT 99 or BIT 01 Exp. or to a Computer
1st byte: 0jjj - dddd. 0jjj = identifier device to which BIT 99 has to transfer program (001 = BIT 01 Exp. - 010 = BIT 99). dddd = MIDI channel to which BIT 99 send program.
2nd. byte: 0ppp pppp = Program N° requested 0-98

BIT 99 - BITMAPS

The following parameter values can range between 00H and FFH (0-255 decimal). As MIDI standards require the leading bit (MSB) of System Exclusive data to be 0, the maximum value that can be represented in one single byte is reduced to 7FH (127 decimal). Thus, all data bytes (written as 0wwww wwww) must be split into two bytes for transmission. The first byte contains the least significant four bits of the value (LSB) in its low order nibble (right hand four bits); the second byte contains the most significant four bits of the value (MSB) in its low order nibble. An example: a value of 9FH is transmitted as 0FH (first byte) plus 09H (second byte).

The BIT 99 does not use the full resolution of 256 steps that can be represented in a value; the full resolution of 256 steps might be used in future expansions. The resolution (step size) used for each parameter is shown in the right-most column of the bit maps below. Where step size = 4, the value of the right-most two bits is ignored; with step size = 8, the value of the right-most three bits is ignored; where step size = 1, the full resolution as indicated by RANGE is used.

PROGRAM BIT MAP

BYTE	PARAMETER	RANGE	DESCRIPTION	STEP SIZE
00	12	00H-FFH	Wheel Amount	4
02	11	"	LFO1 Depth	4
04	10	"	LFO1 Dynamic Rate	4
06	63	"	LFO2 Depth	4
08	62	"	LFO2 Dynamic Rate	4
10	45	"	Detune	(Note 1)
12	48	"	Volume	4
14	34	"	Noise	4
16	33	"	DCO1 Dynamic Pulse Width	4
18	44	"	DCO2 Dynamic Pulse Width	4
20	**	***	DCO1 Octave/Freq. (Note 2)	n/a
22	**	***	DCO2 Octave/Freq. (Note 2)	n/a
24	32	00H-FFH	DCO1 Pulse Width	8
26	43	"	DCO2 Pulse Width	8
28	19	"	VCF Cut Off Frequency	4
30	20	"	VCF Resonance	4
32	15	"	VCF Sustain	4
34	21	"	VCF Envelope	4
36	18	"	VCF Tracking	4
38	17	"	VCF Dynamic Attack	4
40	22	"	VCF Dynamic Envelope	4
42	51	"	VCA Sustain	4
44	46	"	VCA Dynamic Attack	4
46	47	"	VCA Dynamic Volume	4

BYTE	PARAMETER	RANGE	DESCRIPTION	STEP SIZE
48	13	00H-3FH	VCF Attack	1
50	14	"	VCF Delay	1
52	16	"	VCF Release	1
54	49	"	VCA Attack	1
56	50	"	VCA Delay	1
58	52	"	VCA Release	1
60	08	"	LFO1 Delay	1
62	60	"	LFO2 Delay	1
64	09	"	LFO1 Rate	1
66	61	"	LFO2 Rate	1
68	**	***	LFO Flags byte 1 (Note 3)	n/a
70	**	***	LFO Flags byte 2 (Note 4)	n/a
72	**	***	DCO Flags (Note 5)	n/a

SPLIT/DOUBLE BIT MAP

BYTE	PARAMETER	RANGE	DESCRIPTION	STEP SIZE
00	n/a	01H-4BH	Lower Program 1-75	1
02	n/a	"	Upper Program 1-75	1
04	n/a	01H-02H	Mode (1=Split, 2=Double)	n/a
06	64	00H-3CH	Split Point, key 1-61	1
08	65	"	Upper Transpose, key 1-61	1
10	66	00H-FFH	Lower Volume	4
12	67	"	Upper Volume	4

Notes:

1) The value for Detune is interpreted as follows:
00H - 7FH : 00H
80H - FFH : subtract 80H and divide the remainder by 2

2) DCO Octaves and Frequencies are coded together in a single hex value. Octaves are defined as: 32'=0, 16'=1, 8'=2 and 4'=3. To calculate the hex value, use the following formula: Octave * 12 + Frequency. Frequency can range between 00H and 0BH (0-11 decimal). An example: Octave 8', Frequency = 5 is calculated as 2 * 12 + 5 = 29 (1DH).

3) The single bits in LFO Flag byte 1 contain the on/off status for LFO control.

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
VCA	VCF	DCO 2	DCO 1	VCA	VCF	DCO 2	DCO 1
----controlled by LFO2----				----controlled by LFO1----			

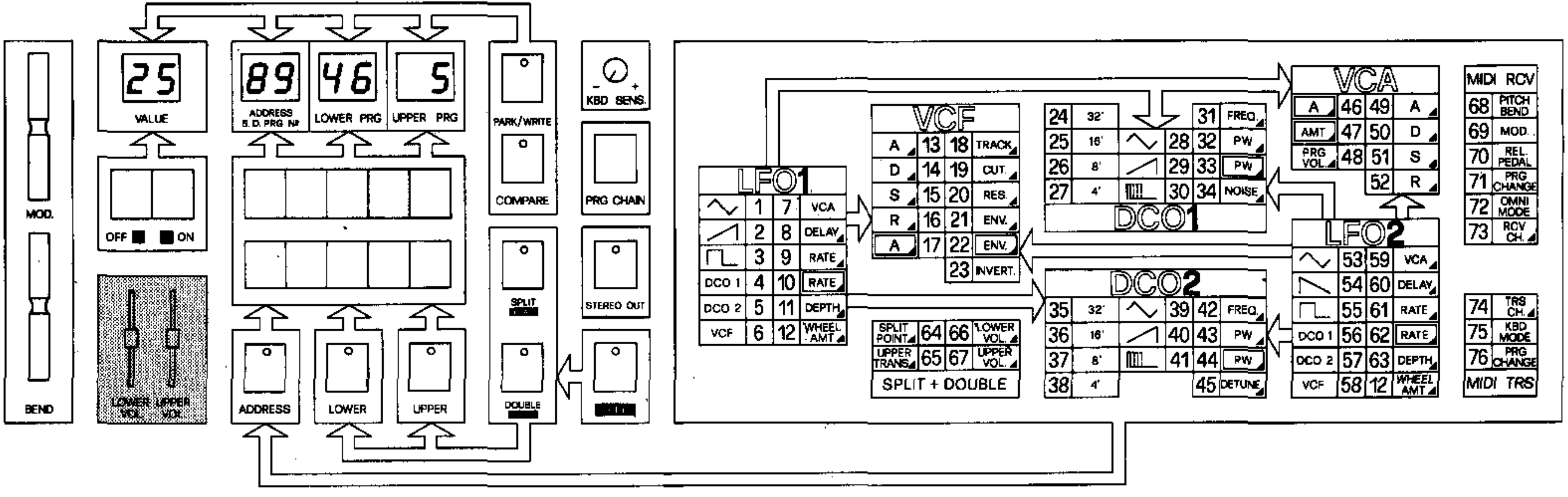
4) The single bits in LFO Flag Byte 2 define the Invert Status of the VCF-ADSR and the waveforms of the LFOs.

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Invert	N/A	N/A	N/A	LFO2-wavefrm		LFO1-wavefrm	

waveforms: 00 = no LFO
01 = triangle
10 = sawtooth
11 = pulse

5) The DCO flags determine the DCO waveforms or combinations thereof.

Bit 1	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
N/A	N/A	DCO 2	DCO 1	DCO 2	DCO 1	DCO 2	DCO 1
		--Triangle--		--Sawtooth--		---Pulse----	


MOD.
BEND
25
VALUE
OFF
ON
LOWER VOL
UPPER VOL
89
ADDRESS S.D. PRG N°
46
LOWER PRG
5
UPPER PRG
ADDRESS
LOWER
UPPER
PARK/WRITE
COMPARE
DOUBLE
KBD SENS.
PRG CHAIN
SPLIT
STEREO OUT
LFO1
1 7 VCA
2 8 DELAY
3 9 RATE
DCO 1 4 10 RATE
DCO 2 5 11 DEPTH
VCF 6 12 WHEEL AMT
VCF
A 13 18 TRACK
D 14 19 CUT.
S 15 20 RES.
R 16 21 ENV.
A 17 22 ENV.
23 INVERT.
SPLIT POINT 64 66 LOWER VOL.
UPPER TRANS. 65 67 UPPER VOL.
SPLIT + DOUBLE
24 32' 31 FREQ.
25 16' 28 32 PW
26 8' 29 33 PW
27 4' 30 34 NOISE
DCO1
DCO2
35 32' 39 42 FREQ.
36 16' 40 43 PW
37 8' 41 44 PW
38 4' 45 DETUNE
VCA
A 46 49 A
AMT 47 50 D
PRG VOL. 48 51 S
52 R
LFO2
53 59 VCA
54 60 DELAY
55 61 RATE
DCO 1 56 62 RATE
DCO 2 57 63 DEPTH
VCF 58 12 WHEEL AMT
MIDI RCV
68 PITCH BEND
69 MOD.
70 REL. PEDAL
71 PRG CHANGE
72 OMNI MODE
73 RCV CH.
74 TRS CH.
75 KBD MODE
76 PRG CHANGE
MIDI TRS

challenging technology