Design Techniques for

In Part 1 of this article we discussed basic rhythm structures and established the principles of rhythm generation using clock oscillators, counters and instrument simulators. From this theoretical discussion we now move on to practical circuits which can exploit these ideas and serve as a basis for further experimentation.

As we established in part 1, the basis for every rhythm unit is a device that generates pulses sequentially on separate output lines. One can obtain rhythm effects with as few as three or four output lines but to allow for more interesting variations 16 lines must be considered the minimum and for very fancy rhythm patterns one may need 32 lines to allow for 1/32 beats. Rather than just give one circuit, we shall discuss several approaches; a basic 16 line version, a 32 line version, and a scanning multi-pass circuit. We shall also give some diode matrix rhythm pattern examples.

One word of warning. If you decide to start experimenting with pulse generating circuits, make sure you have an oscilloscope. Without this instrument it is almost impossible to track down faults in the circuit if it does not work.

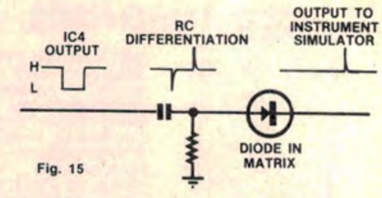
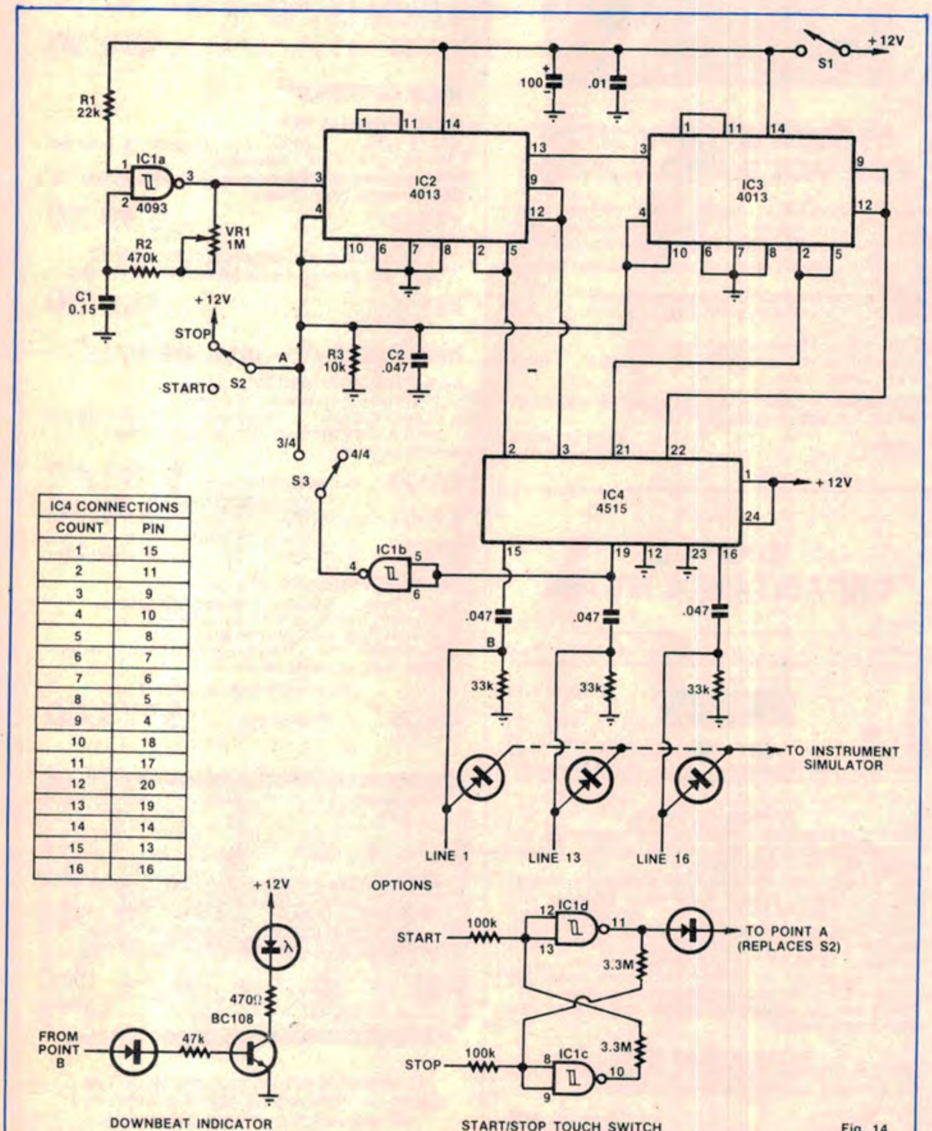


Fig. 15

The complete circuit of a basic single pass 16 beat pulse counter is shown in Fig. 14. The heart of the circuit is the combination of IC2, 3 and 4 which together form a counter with one clock frequency input and 16 output lines. IC2 and 3 are dual D-type flipflops (4013) which together form four flipflop stages in cascade. The Q output of each stage is used to provide the four input lines to IC4.

Depending on the state of each flipflop, the four lines carry a code of high and low voltages. These codes are accepted by IC4, a 4-to-16-line decoder, type 4515. Each input code results in one of the 16 output lines being activated. We shall therefore require a sequentially changing code to obtain the required effect of the 16 output lines to be activated in sequence.

The outputs of IC4 are normally high and will go low one by one when the ap-



IC4 CONNECTIONS	
COUNT	PIN
1	15
2	11
3	9
4	10
5	8
6	7
7	6
8	5
9	4
10	18
11	17
12	20
13	19
14	14
15	13
16	16

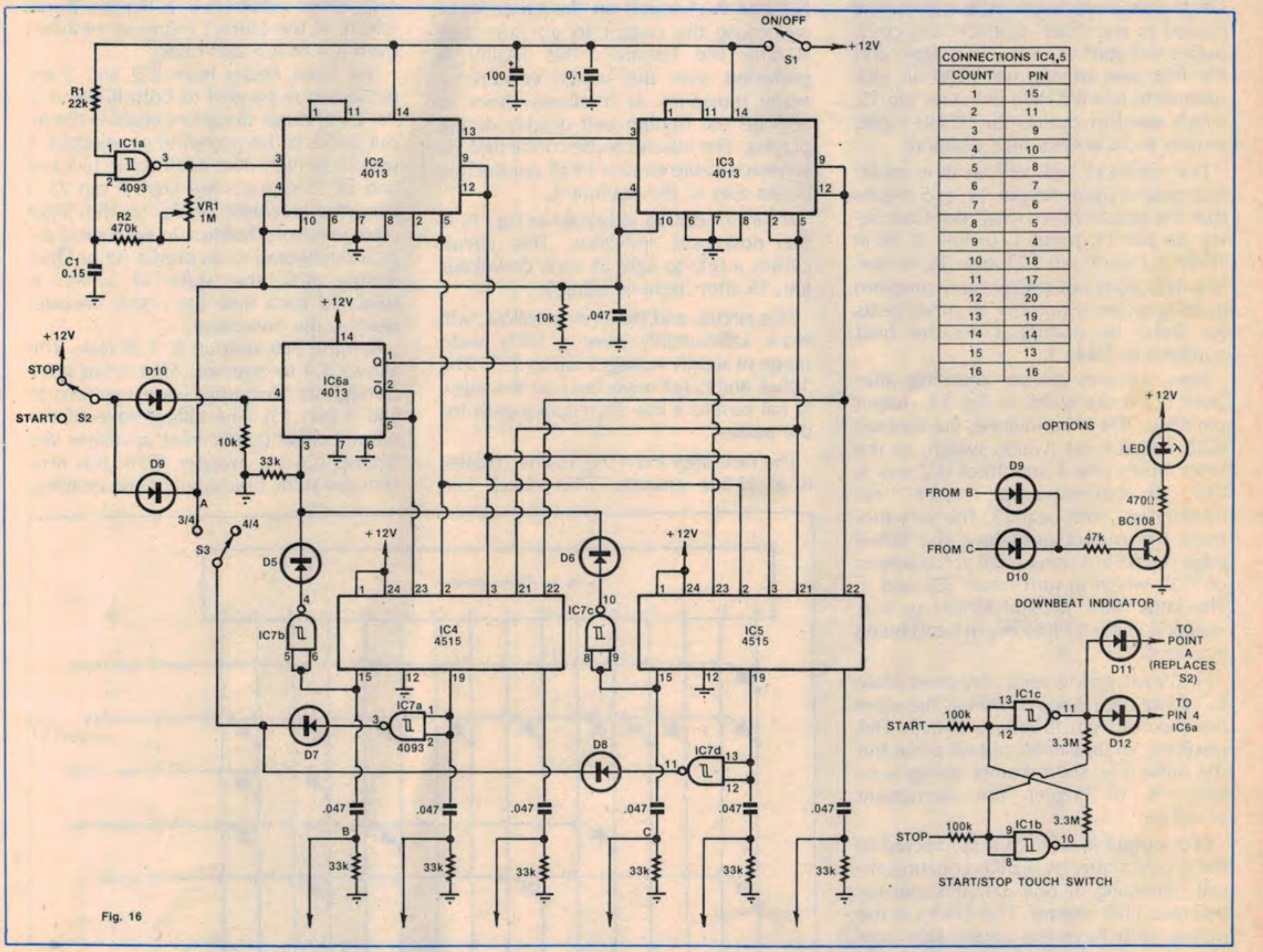
Fig. 14

TABLE 1

Input pulse number	Output IC 2 and 3 stage 1	stage 2	stage 3	stage 4	IC4 pin low
1	L	L	L	L	11
2	H	L	L	L	9
3	L	H	L	L	10
4	H	H	L	L	8
5	L	L	H	L	7
6	H	L	H	L	6
7	L	H	H	L	5
8	H	H	H	L	4
9	L	L	L	H	18
10	H	L	L	H	17
11	L	H	L	H	20
12	H	H	L	H	19
13	L	L	H	H	14
14	H	L	H	H	13
15	L	H	H	H	16
16	H	H	H	H	15

Rhythm Generators *Part 2*

by NICK LABORDUS *5/3 Durham Close, North Ryde NSW 2113*



CONNECTIONS IC4,5	
COUNT	PIN
1	15
2	11
3	9
4	10
5	8
6	7
7	6
8	5
9	4
10	18
11	17
12	20
13	19
14	14
15	13
16	16

propriate input code is presented. As we shall see later, the instrument simulators require a sharp pulse at their input. This means that the output signals of IC4 must be differentiated (Fig. 15). Differentiation of a square wave by an R/C network results in two pulses, a negative one on the falling edge of the square wave and a positive one on the rising edge. Only one of these two can be used to activate the instrument simulator and a decision must be taken before the counter circuit can be designed.

Either version is possible but as there is a time shift between the positive and the negative pulse, the choice has some effect on certain reset connections in the counter circuit. In our example we have selected instrument simulators with a positive input pulse.

In Fig. 14, IC1a and its associated components form the clock pulse generator, the frequency of which is determined by the values of C1, R2 and VR1. The clock produces square wave pulses which are fed into four flipflop circuits in cascade. Both IC2 and IC3 consist of two flipflops each. Each flipflop stage divides its input pulse frequency by two. At a clock frequency f, the respective output frequencies of the stages 1 to 4 are: Stage 1 (pins 2, 5 IC2) f/2; stage 2 (pins 9, 12 IC2) f/4; stage 3 (pins 2,5 IC3) f/8; stage 4 (pins 9, 12 IC3) f/16.

Table 1 shows the behaviour of the four output lines as new pulses from the clock generator arrive at the input of flipflop 1. Table 1 also shows which of the 16 output lines of IC4 (indicated by pin number) goes from high to low at what input code. Note that the flipflops return automatically to the first code after having reached the 16th. This suits our aim perfectly for all non-waltz rhythm patterns. For waltz pattern we need only 12 lines and we shall have to reset the counter after it has produced the 12th output pulse.

But before considering this function in detail we will have to modify the beat (input pulse) numbers of Table 1. This is due to the need to always start the rhythm generator on beat 1.

Using the "on-off" switch to start the unit will result in the counter starting in a random fashion rather than on beat 1 (the down beat). This can be overcome by introducing a separate switch for the "start-stop" action, leaving the "on-off" switch in the "on" position so that all cir-

Design Techniques for Rhythm Generators

cuits are functional.

In the "stop" position, S2 connects a high to reset pins 4 and 10 of IC2 and 3. This forces the four selected outputs to show HHHH, independent of the arriving clock pulses. As soon as the switch is moved to the "start" position, the clock pulses will start to take effect again and the first one arriving will send all four outputs to low (LLLL). In doing so, pin 15, which was low due to the HHHH input, returns to its normal high position.

This return to high will result in an active output pulse on pin 15, and means that the actual "down beat" does not occur on pin 11 (pulse 1, or line 1), as in Table 1, but on pin 15 (pulse 16, or line 16). This does not present any problem as long as we re-number all other beats (or lines) by adding 1 to the beat numbers in Table 1.

Now we can discuss resetting after beat 12 for the waltz. In Fig. 14, output pin 19 of IC4 is connected, via inverter IC1b and the 3/4 (waltz) switch, to the reset inputs pins 4 and 10 of IC2 and 3. Pin 19 corresponds, after "re-numbering", with beat 13. The very moment this output goes low, the falling edge results in a rising edge at the output of IC1b which in turn resets IC2 and 3. The latter now present HHHH to IC4, resulting in pin 15 (the down beat) being activated.

Pin 19 will go low for a very short while to initiate the reset. As soon as the reset has occurred, pin 19 returns to high. This results in an unwanted output pulse but this pulse is so short that its energy is insufficient to trigger the instrument simulator.

Each output line of IC4 is connected to the diode matrix by a differentiating circuit consisting of one .047μF capacitor and one 33kΩ resistor. The diodes in the matrix must have the anode side connected to the output lines so that only the positive pulses will arrive at the instrument simulators.

Having two spare gates in IC1, we can add a touch control to our rhythm unit without much effort. Touching one of the inputs with the finger will produce a high output, acting as "stop" by resetting IC2 and 3. A touch on the other input will cause the output to go low, thus starting the counter. This facility is perferred over the switch version by many musicians as it allows them to activate the rhythm unit quickly during playing. The inputs can be connected via screened cable to two small conducting plates next to the keyboard.

A further option indicated in Fig. 14 is the downbeat indicator. This circuit causes a LED to light at each downbeat (pin 15 after "re-numbering").

This circuit, and the one to follow, will work satisfactorily over a fairly wide range of supply voltages, up to 15V. The 100μF and 0.1μF capacitors on the supply rail ensure a low impedance path for the pulses.

The next step from the 16-line counter is a 32-line counter. This circuit can create fancy rhythm patterns with 1/32 beats or act as a double-pass 16-beat counter, allowing some reduction of monotony. The full circuit is shown in Fig. 16. The heart of the circuit is the same as that of the 16 beat counter. In this case there are two 4-to-16-line decoders to obtain 32 output lines. The only other addition is a flipflop (IC6a) which, at the correct moment, switches from IC4 to IC5 and back.

The input codes from IC2 and 3 are presented in parallel to both IC4 and 5. Pin 23 of these decoders enables the input codes to be accepted or rejected. If pin 23 is high, the decoder is blocked and all 16 outputs are high. If pin 23 is low, the decoder reacts to the input code in normal fashion. In each case, pin 23 is connected to an output (Q or $\overline{Q}$) of flipflop IC6, the state of which is switched each time the active decoder reaches the downbeat.

Assume IC6 output 2, 5 is low. This allows IC4 to operate. After it has gone through its 16 output lines it returns to line 1 (pin 15). The falling edge of this output pulse (going low), activates the flipflop IC6 via inverter IC7b. IC6 now changes state, blocking IC4 and enabling

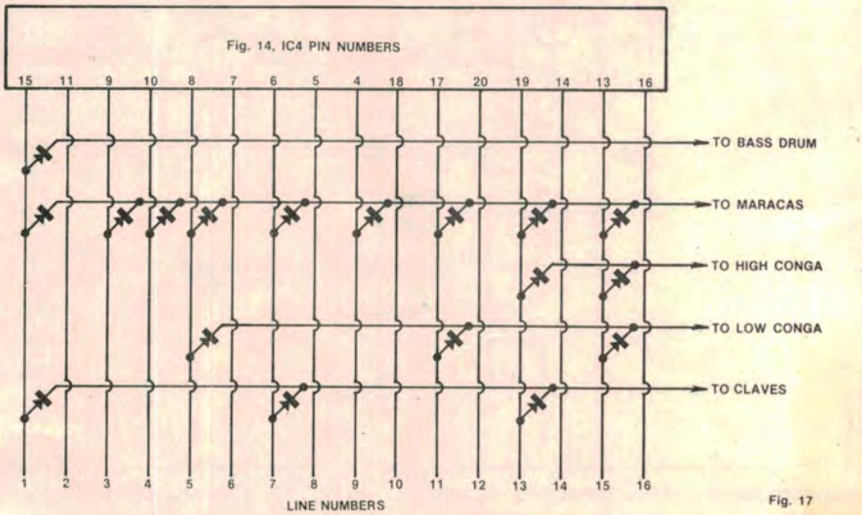
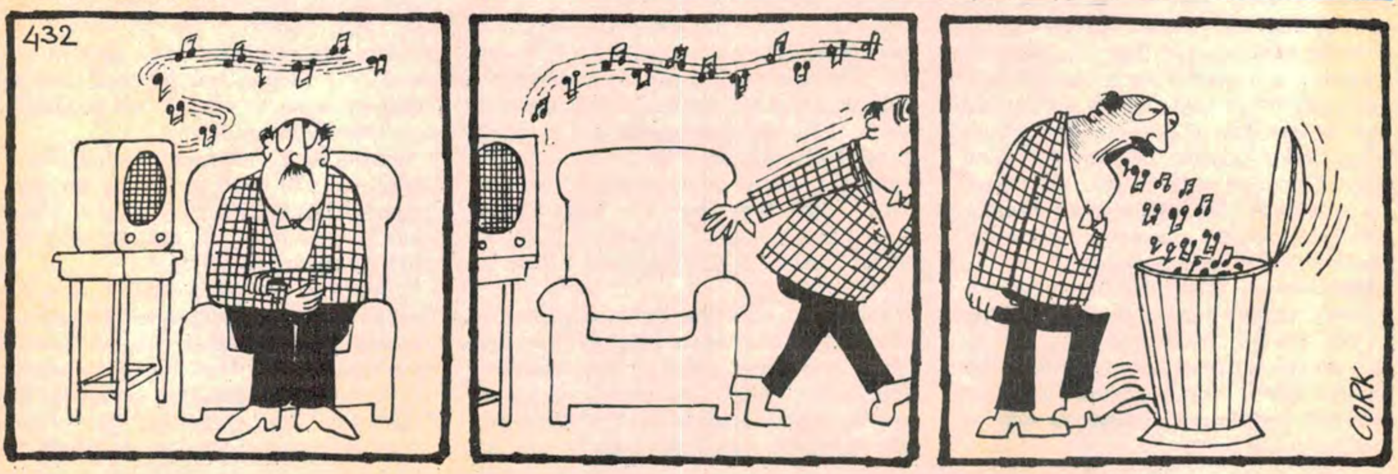


Fig. 17



IC5. The latter now reacts to the available HHHH input by sending pin 15 low, this pulse becoming the down beat for the IC5 cycle of line pulses.

(Note that there is an active output pulse on pin 15 of IC4 at this moment, as this pin returns to high. This pulse is so short, however, that the energy content is insufficient to activate the instrument simulator.)

This low pulse on pin 15, IC5, also appears, inverted by IC7c, at the input to IC6a (pin 3), but does not have any additional effect because it appears at this point simply as a continuation of the pulse which began on pin 15 of IC4. After IC5 has been through a full cycle of 16 beats and pin 15 goes low again, IC6 switches again and enables IC4.

The "start-stop" and "3/4" actions are the same as described before but require more connections to control more ICs. The circuit also shows the options suggested for the previous circuit, the down beat indicator and the start-stop touch switch.

The diode matrix represents the "recipe book" of the rhythm unit. It is not the intention of this article to provide the reader with a complete list of recipes. The reason for experimenting with rhythm units is to develop new and interesting patterns. The way to do this is to listen to a record, analyse the rhythm and to try to simulate it in the diode matrix. This takes time and patience but is great fun, especially when one plays an instrument like a piano or an organ.

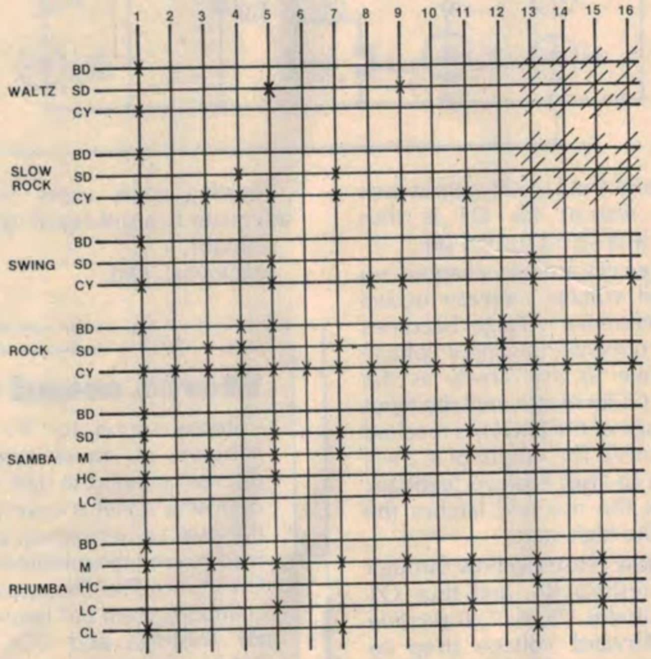
In order to set up the rhythm unit with some test patterns, Table 2 gives some recipes for a 16-beat counter. Fig. 17 shows the actual diode matrix for the "rhumba" pattern in Table 2.

Rhythm selector switches are required to connect certain diode matrices to instrument simulators. For each rhythm, one multi-pole switch is needed. As explained in Part 1, such switches are difficult to buy, and expensive. For that reason it was decided to use single pole switches which control type 4016 quadruple bilateral switches. These ICs are readily available and as the four CMOS gates in each IC are individually accessible, total freedom is obtained in determination of the number of poles per selector switch.

Fig. 18 shows the "waltz" selector switch. The number of switch elements is equal to the number of instrument simulators to be activated (plus one acting as the 3/4 switch in this case). With the selector switch in the "off" position, the control inputs of the CMOS switches are kept low, causing them to be turned off. In the "on" position, the control inputs go high and each CMOS switch element now connects the appropriate diode pattern to the instrument simulators.

Continued next month

TABLE 2



BD : BASS DRUM SD : SNARE DRUM CY : CYMBALS M : MARACAS HC : HIGH CONGA LC : LOW CONGA CL : CLAVES

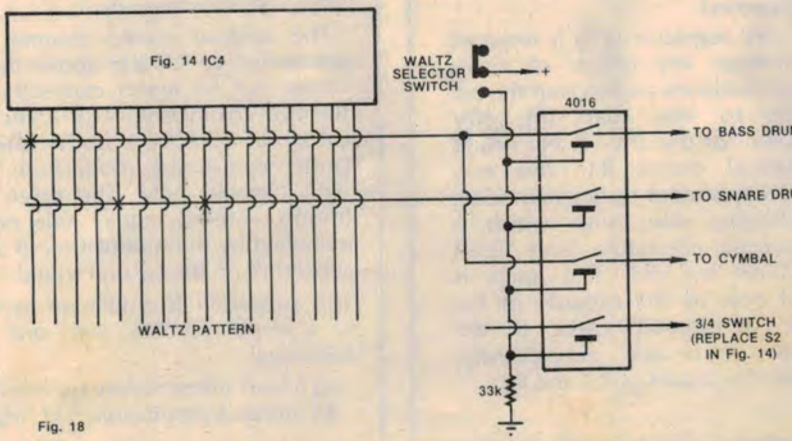


Fig. 18