Basic Electronics

Build this simple electronic die

Here's a simple electronic die that has a seven segment readout to display the result of a throw. The design overcomes the problem often associated with an electronic die — bias towards a particular result. The distribution of results for this design is remarkably even and appears to have no bias.

by GERALD COHN

Even with all the variety available to us today, particularly in the way of electronic games, there is still a strong attraction to games of chance, in particular those employing a die or a pair of dice. However, many arguments have been caused by the penchant for normal dice to give apparently biased or ambiguous results, particularly when the fever of the game is running high.

The readout of our electronic die is in the form of a seven segment display which is used to display the digits one to six inclusive, representing the six sides of a conventional die. In all, three ICs are employed in the circuit and the construction is made simple by the use of a printed circuit board.

Now, let's take a look at the circuit and see how it works. The circuit can be broken up into three separate sections, an oscillator, a counter, and the display circuits.

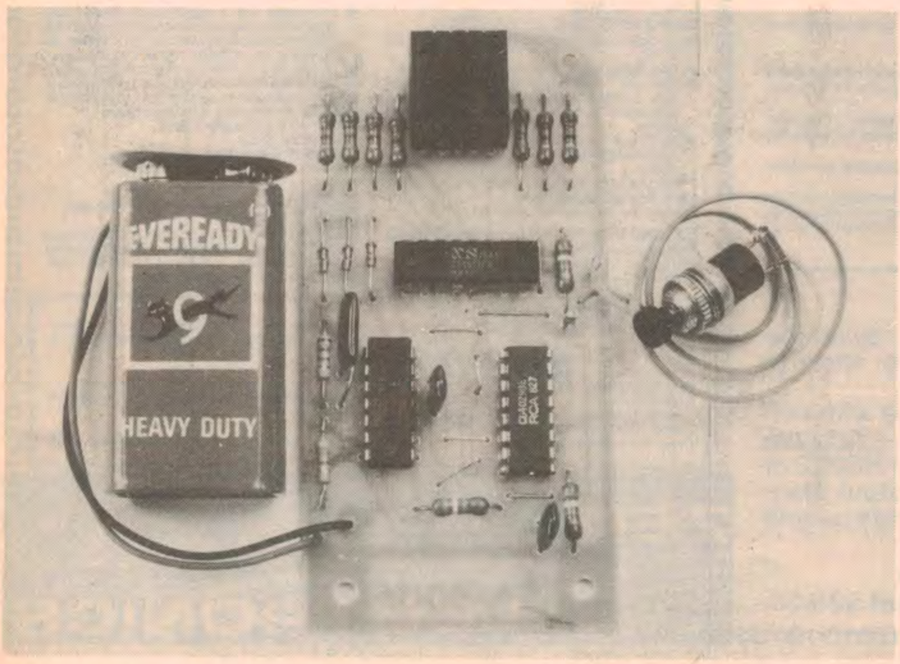
The oscillator is a typical three-gate type, in this case made up using three of the available six inverters in a 4069 IC package. Operation of the oscillator circuit is simple: if we assume the output of the oscillator to be at a logical high level, then the input to last inverter in the chain must be low. The capacitor charges up until the voltage across it is equal to the required voltage at the input to the first inverter to cause the outputs of all the inverters to change state; the capacitor now begins to charge up in the opposite direction. This cycle repeats itself for as long as power is applied to the circuit.

The counter following the oscillator is the CMOS 4029, a four bit counter with the capacity to count from zero to 15. But, there are some special control inputs to the counter that allow us to use the device in a number of different modes. We can for example choose whether the counter is to count up from zero, or down to zero. A single pin on the IC controls the direction of the count.

Since we are using the counter as a die, we only require it to count up to six and then reset to start from one. This is accomplished using only the first three least significant outputs from the counter, the fourth not being needed. We have left the fourth output unconnected, while the other three are connected to the inputs of a display decoder/driver, 4511. This IC takes the binary output of the counter and decodes it to drive a seven segment display. Also connected to the counter outputs is a three diode AND gate.

We are in fact using the AND gate to detect the count of seven. Since it is desired to display the digit "six", we have had to actually count up to seven before the reset takes place. Here's how it works: the AND gate detects the count of seven (all outputs from the counter high). When this condition occurs, the output of the AND gate goes high and after buffering by two inverters applies a pulse to the preset enable input, loading "one" into the counter registers (via the preset enable, pin four). The counter begins to count from one again, but the thing to remember is that the counter resets almost instantly when the preset pulse appears at the preset enable input, so "seven" never appears.

We have used the latching function of the 4511 decoder/driver for the "throw" pushbutton. When the button is depressed, the latch enable (LE) input is taken low,



Above is a view of the completed electronic die. It's easy to built, with all components (except for the battery and switch) mounted on a PC board.

We estimate that the current cost of parts for this project is approximately

$10.00

This includes sales tax.

thus allowing the data at the inputs to pass straight through to the outputs. When the button is now released however, the level at the LE input changes to high, and the data that last appeared at the inputs the moment that the LE input was taken high is latched into the flipflop memory. This latched data will be displayed until the button is again depressed.

We used the blanking input of the 4511 to conserve the battery. This is done by connecting the clock output (from the oscillator) to the blanking input and thus turning the display on and off at the rate of about 10kHz. Since the clock output has a duty cycle of about 50% (ie, it is a square wave) it cuts the battery drain due to the display by about half. The total current drain is actually 23 milliamps.

The display we used is the commonly available FND-500. This has a common cathode line, the anodes being driven by the outputs of the decoder IC via 560 ohm resistors to limit the current through the segments to approximately 10mA.

CONSTRUCTION

The unit is constructed on a printed circuit board (80d6) measuring 48 × 94mm which accommodates all components with the exception of the battery and the push button.

When assembling the PCB refer to the component overlay diagram taking special note of the orientation of the ICs, the display and the diodes. Start by first inserting the seven wire links and soldering these to the board, followed by the resistors and the capacitors.

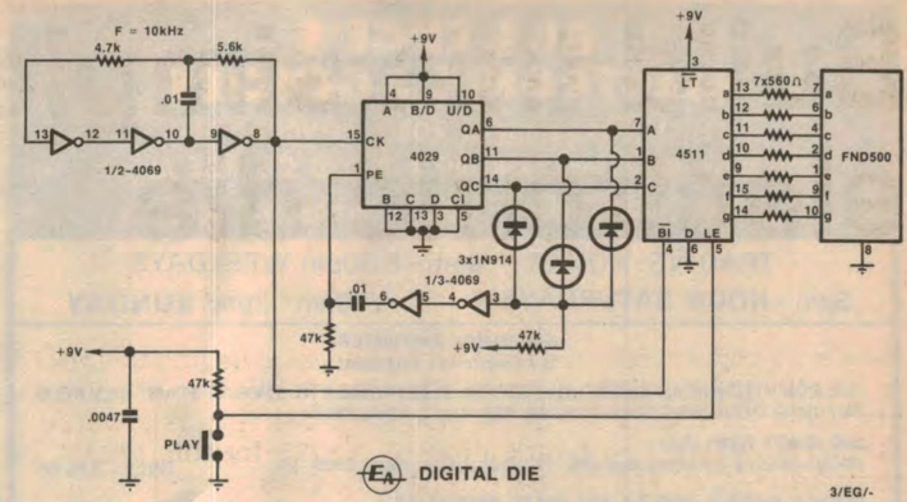
The last components to go on the board are the semiconductors: first the diodes and then the ICs. When you solder the ICs to the board, start by soldering the supply pins first, and then proceed to solder the remaining pins to the board. The reason for this is that the CMOS ICs are susceptible to damage from static discharges, and soldering the supply pins first goes a long way to preventing any damage.

When you have finished assembling the board, go back over it, looking for solder bridges between adjacent copper tracks, dry joints, and unsoldered components. Then turn the PCB over and do a final check to see that all components have, in fact, been properly placed. When you have satisfied yourself that all is OK, solder the battery clip and pushbutton wires to the board.

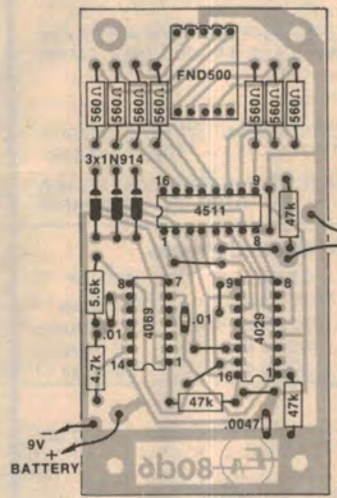
Testing is easy – just clip the battery in place and press the pushbutton. The display will show the figure "eight". The reason for this is due, as you will remember from earlier discussion, to the fact that when the latch enable input is taken low, the data at the inputs to the decoder flows straight through to the outputs to be displayed. The "eight" is due to all the segments turning on and off at a 10kHz count rate. When you now release the button, the display will show any of the digits between one and six inclusive.

All that needs be done now is to house the PCB in some sort of box. We have left this up to the constructor to do since he may have a special way of wanting to house it. The board was sized to fit into one of the plastic utility boxes, and it can be housed in one of these or it may be mounted into the "family" gaming table.

Anyway, whichever way you choose to house it, one thing is certain – no one can say the die is biased. Lot's of luck, and good betting!



The circuit consists of three sections: an oscillator, a counter and the display.



Solder the power supply pins first when mounting the CMOS ICs. These are pins 7 and 14 for the 4069, and pins 8 and 16 for the 4029 and 4511.

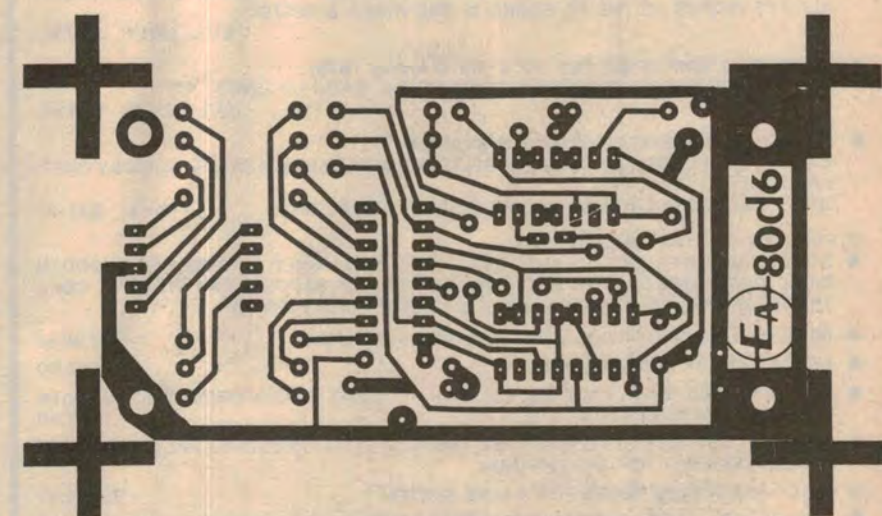
PARTS LIST

SEMICONDUCTORS
1 x 4029 4-bit counter
1 x 4069 hex inverter
1 x 4511 seven-segment decoder/driver
1 x FND-500 seven-segment display
3 x 1N914 diodes

CAPACITORS
2 × .01uF greencap (metallised polyester)
1 x .0047uF greencap

RESISTORS
3 x 47k, 1 x 5.6k, 1 x 4.7k, 7 x 560 ohm

OTHER
1 printed circuit board 80d6, 48 x 94mm
1 pushbutton (normally open)
1 type 216 nine volt battery and clip to suit



Here is an actual size reproduction of the PC pattern.