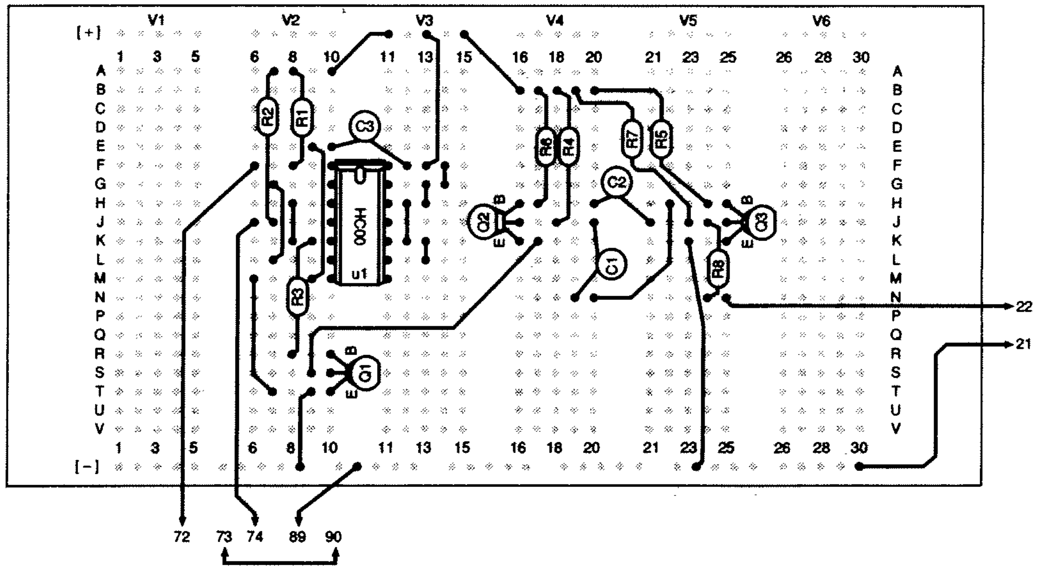
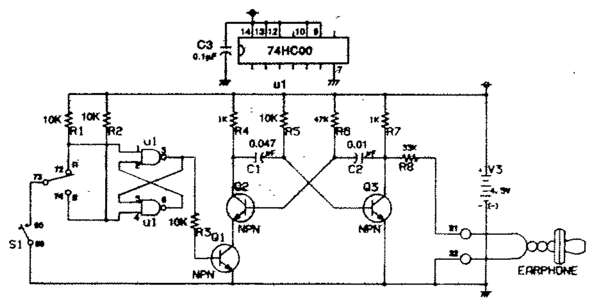
PROJECT 160. SET/RESET BUZZER



U1	74HC00	R1	10KΩ	R5	10KΩ	C1	0.047μF
Q1	NPN	R2	10KΩ	R6	47KΩ	C2	0.01μF
Q2	NPN	R3	10KΩ	R7	1KΩ	C3	0.1μF
Q3	NPN	R4	1KΩ	R8	33KΩ		

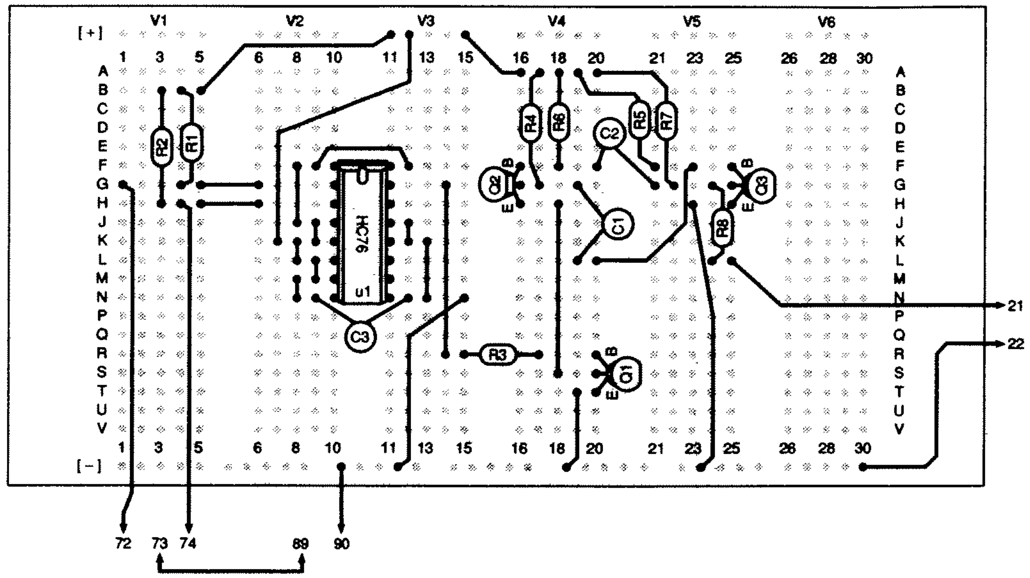
Anything look familiar about the schematic for this project? If not, take a look back at the schematic for project 158. You'll see that this circuit uses an R-S flip-flop circuit made from NAND gates.



After you finish wiring this project, set the **select switch** up and press **S1**. You'll hear a sound through the earphone. Try pressing **S1** several times. You'll still hear a sound in your earphone. Now move the **select switch** down and press **S1** one more time. What happens now?

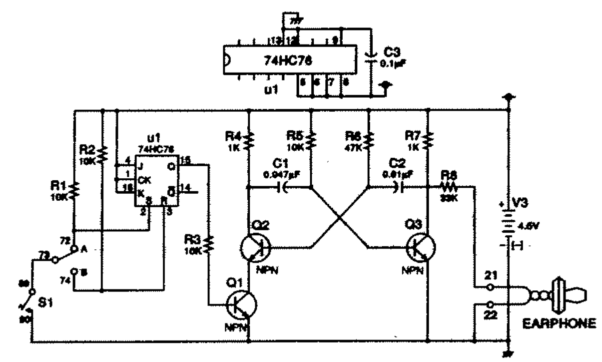
Circuits like this can be used in alarms. They're especially useful since intruders often can't figure out how to make the sound stop. You might also want to experiment using **LEDs** instead of sound to alert you that the circuit has been set or reset.

PROJECT 161. SET/RESET BUZZER II



U1	74HC76	R1	10KΩ	R5	10KΩ	C1	0.047μF
Q1	NPN	R2	10KΩ	R6	47KΩ	C2	0.01μF
Q2	NPN	R3	10KΩ	R7	1KΩ	C3	0.1μF
Q3	NPN	R4	1KΩ	R8	33KΩ		

Here's another version of our last projects, this time using a transistor multivibrator and a J-K flip-flop.

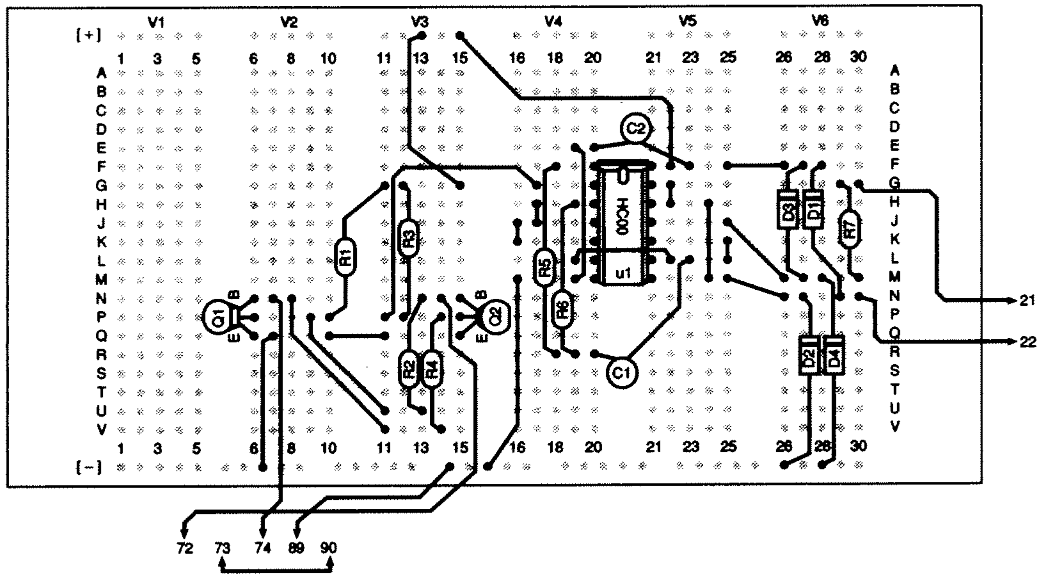


Set the **select switch** up and press **S1**. You'll hear a sound in your earphone. Just as in our last project, you'll continue to hear sound in the earphone no matter how many times you press **S1** again. Set the **select switch** down and press **S1** one more time. You'll hear the sound stop.

You can see from the schematic that the transistor multivibrator is controlled by the Q output of the J-K flip-flop. When you have the **select switch** at the up position and press **S1**, Q becomes 1 and the transistor multivibrator keeps on working regardless of how many more times you press the key. But if you press the **S1** when the **select switch** is at the down position, Q becomes 0. This output goes to emitter of the left transistor in the multivibrator and prevents the multivibrator from working.

(We've mentioned this before - but we'll say it again. Aren't transistors terrific in the way they can be switched on and off so easily?).

PROJECT 162. SET/RESET BUZZER III

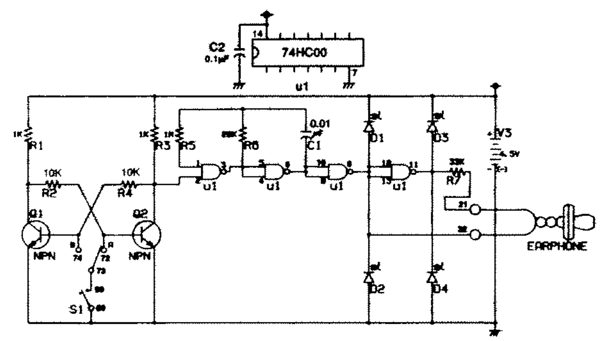


U1	74HC00	R1	1KΩ	R4	10KΩ	R7	33KΩ
Q1	NPN	R2	10KΩ	R5	1MΩ	C1	0.01μF
Q2	NPN	R3	1KΩ	R6	22KΩ	C2	0.1μF

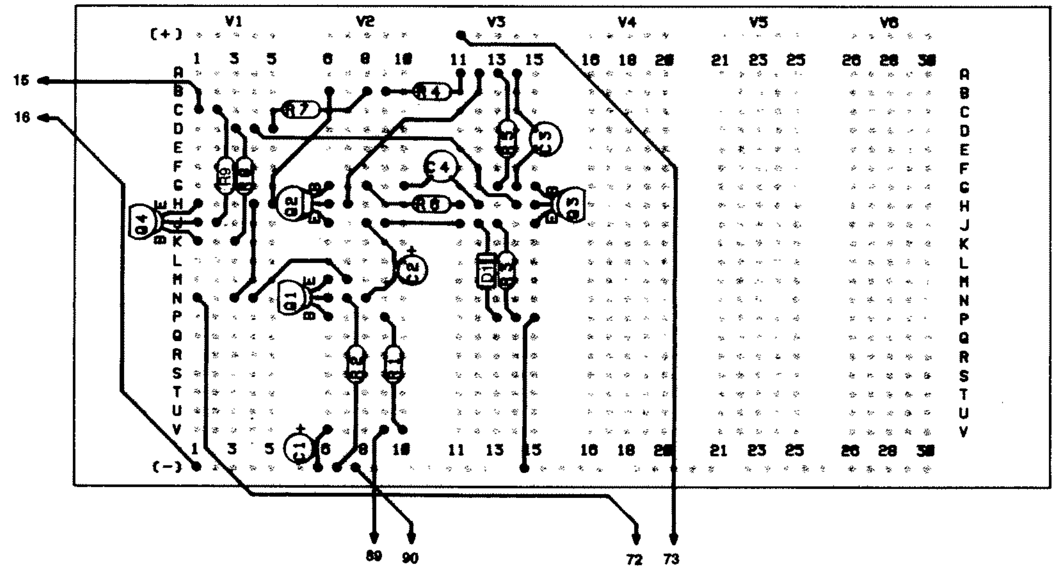
Here's another version of our last two project. This time we used a NAND multivibrator and an R-S flip-flop by transistors.

This circuit works like our last two. When you set the **select switch** up and press **S1**, you'll hear a sound in the earphone. You'll continue to hear the sound no mater how many times you press the key again. Set the **select switch** down and press **S1** and you'll hear the sound stop.

You might want to compare the operation of these last three projects. Is there any difference between the three circuits? Can you think of some situations where one circuit might be better suited than the other two? Be sure to make some notes about what you found out.



PROJECT 163. TRANSISTORIZED TOGGLE FLIP-FLOP



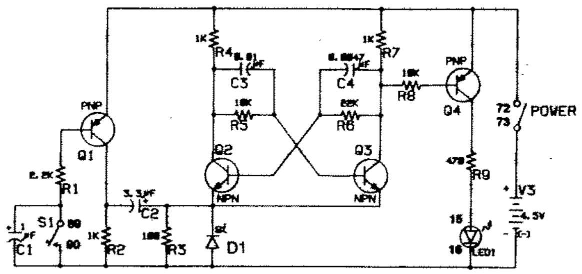
Q1	PNP	R1	2.2KΩ	R4	1KΩ	R7	1KΩ	C1	1μF
Q2	NPN	R2	1KΩ	R5	10KΩ	R8	10KΩ	C2	3.3μF
Q3	NPN	R3	100Ω	R6	22KΩ	R9	470Ω	C3	0.01μF
Q4	PNP					D1	Si	C4	0.0047μF

Do you know what a "toggle" is? (No, it's not someone from the nation of Togg!) A toggle switch is used in electronics to turn circuits on and off... and that should give you a big hint about how toggle flip-flop circuits work!

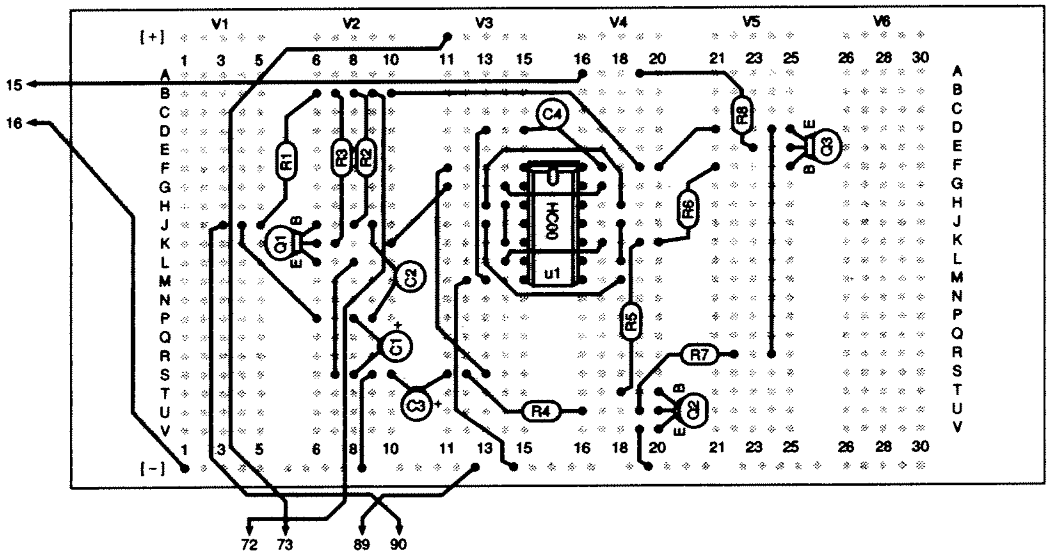
After you build this circuit, turn power ON. Is there any change in **LED 1**? Now press **S1** and see what happens to **LED 1**. Can you explain what is happening?

As you might have guessed by now, a toggle flip-flop is one that sets or resets only when a signal has been received for it to do so, no matter what the inputs tell the circuit to do. This is very handy when we're working with flip-flop circuits. Suppose we have several flip-flop circuits receiving inputs at different times, but we only want them to set or reset together. Using toggled flip-flops means that we can make sure that note of them set or reset until we give them the signal.

We already mentioned that flip-flop circuits can "remember" things thanks to their set and reset abilities. That means several flip-flop circuits controlled by a single toggle signal can operate together to "remember" more and more things. (That's basically how big computers "remember" all they do.)



PROJECT 164. NAND TOGGLE FLIP-FLOP



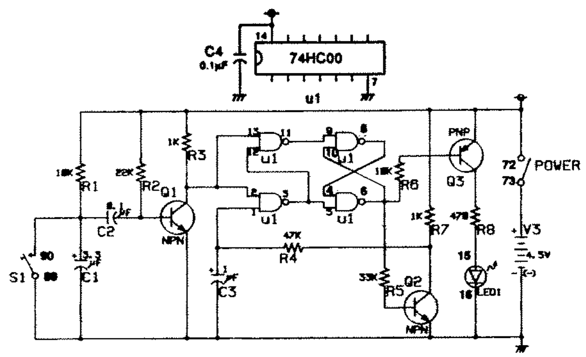
U1	74HC00	R1	10KΩ	R5	33KΩ	C1	3.3μF
Q1	NPN	R2	22KΩ	R6	10KΩ	C2	0.1μF
Q2	NPN	R3	1KΩ	R7	1KΩ	C3	1μF
Q3	PNP	R4	47KΩ	R8	470Ω	C4	0.1μF

If you're starting to suspect that the NAND gate is a very versatile circuit, you're right! Here's a toggle flip-flop circuit made from four NAND gates.

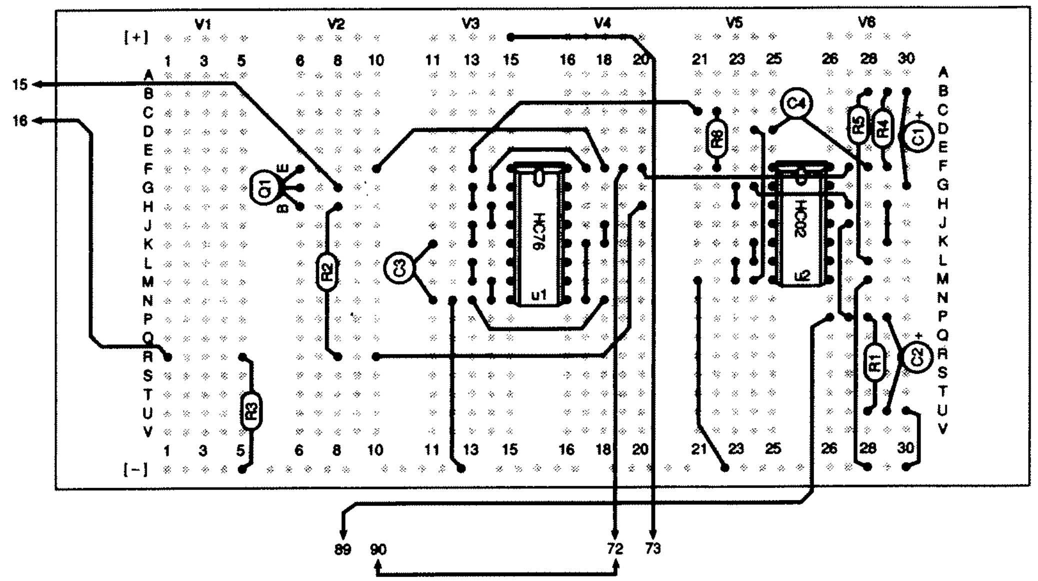
Turn power ON after you finish building this circuit. Press **S1** several times. You'll see that **LED 1** turns on and off each time you press **S1**.

Put on your thinking cap and try to trace what happens from **S1** input to **LED 1**. Of the four NANDs, two function as an R-S flip-flop. See if you can figure out what the remaining NANDs are doing.

This circuit is also a sort of an inverter, since it does take the input and "reverse" it.



PROJECT 165. J-K TOGGLE FLIP-FLOP



U1	74HC76	R1	47KΩ	R4	47KΩ	C1	1μF	C3	0.1μF
U2	74HC02	R2	10KΩ	R5	1KΩ	C2	1μF	C4	0.1μF
Q1	PNP	R3	470Ω	R6	10KΩ				

The two flip-flop circuits contained in the Dual J-K Flip-Flop IC in your kit can be used in "toggled" application. This project shows you how it's done.

You can easily see how this circuits works by examining the schematic diagram. When you press **S1**, you input a clock signal to the flip-flop, allowing it to quickly set and reset. And, as you can also see on the schematic, the output at $\overline{Q}$ is used to control **LED 1**.

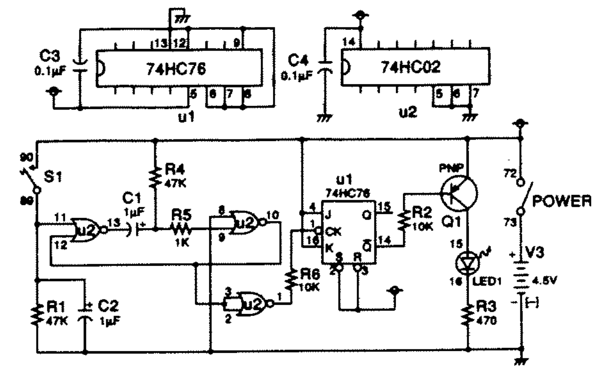
You're probably wondering how a J-K flip-flop circuit is different from an R-S flip-flop. The difference is in the J and K inputs - they're another way of controlling the flip-flop! Here's how J and K control the flip-flop: when both J and K are 0, $\overline{Q}$ stays at its last value (either 0 or 1) regardless of what the clock input is. When this happens, we say the flip-flop is latched. (For this operation, both R and S inputs must be at 0.)

But suppose that J is 0 and K is 1. When this happens, the flip-flop resets when the clock changes from 0 to 1. When J is 1 and K is 0 the opposite happens... flip-flop.

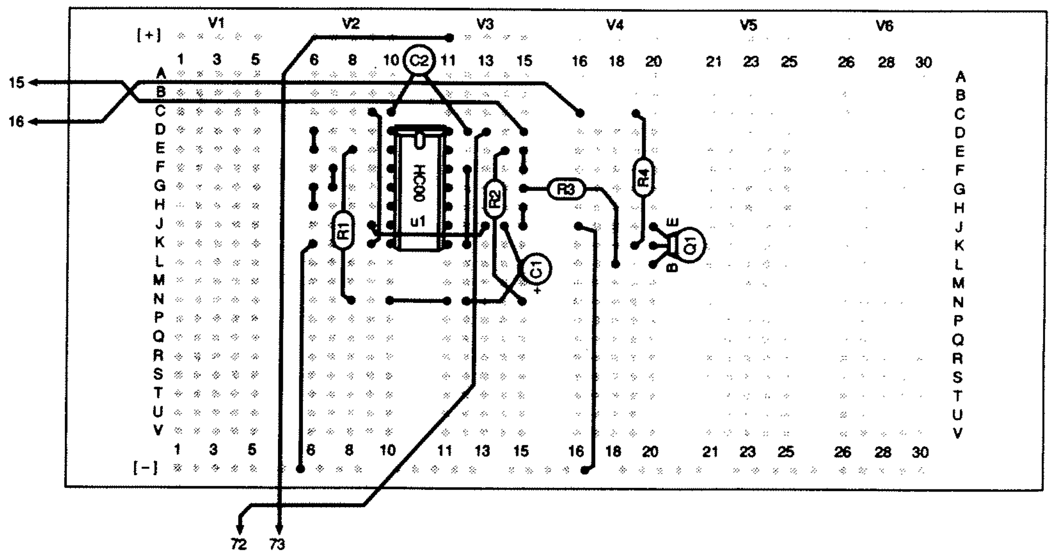
When both J and K are at 1, every time the clock input changes from 0 to 1, it resets and sets alternately.

One interesting use for J-K flip-flops is in "master and slave" circuits (no, you don't use these circuits to build pyramids!). A "master and slave" arrangement takes the output of one flip-flop (the master) and uses it for the J and K inputs of a second flip-flop (the slave). Both the master and slave flip-flops use the same clock signal.

This a little bit about what could be done with a "master and slave" J-K flip-flop arrangement. Can you figure out a why to make such an arrangement "count?" Be sure to make some notes about what you think!



PROJECT 166. C-MOS ASTABLE MULTIVIBRATOR



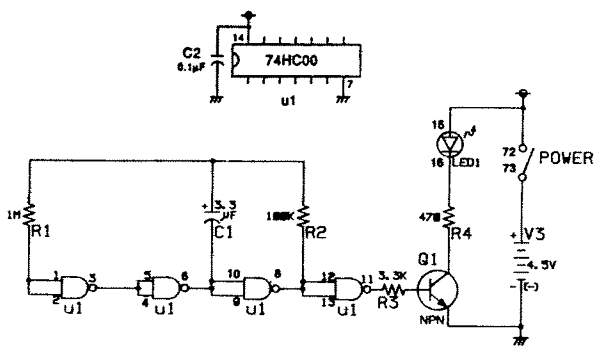
U1	74HC00	R1	1MΩ	R3	3.3KΩ	C1	3.3µF
Q1	NPN	R2	100KΩ	R4	470Ω	C2	0.1µF

Even multivibrator circuits can be made from NAND gates. This project is an example of an astable multivibrator - can you guess what that means? Make a guess, and then build this project to find out.

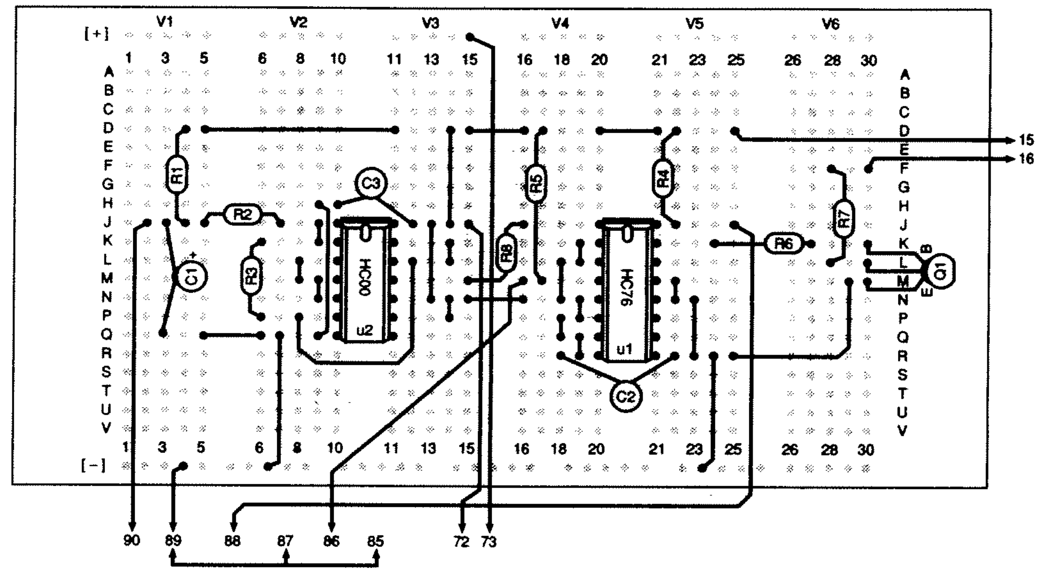
Turn power ON after you finish making all the wiring connections. You'll see **LED 1** flashing on and off. This is like the operation of many of the multivibrators we've used in earlier circuits. Astable means the multivibrator keeps switching back and forth between 0 and 1. As you remember, that's what most of the multivibrators you've built so far do.

You shouldn't have much trouble figuring out how this particular circuit works... and yes, a 3.3µF capacitor makes it all possible. Try using the other electrolytic capacitors in place of the 3.3µF values and see what effect they have on **LED 1** (be sure to observe correct polarity). If you want to see the output of this circuit in a different way, ... it can be done, but we'll let you figure out how.

By now you can see why NAND gate ICs are so popular. The Quad 2-input NAND IC in your kit is one of the most widely used electronic components in the world, mainly because it can be used in so many different ways (and you can probably think of many more!).



PROJECT 167. C-MOS J-K FLIP-FLOP



U1	74HC76	R1	10KΩ	R4	10KΩ	R7	470Ω	C1	1µF
U2	74HC00	R2	10KΩ	R5	10KΩ	R8	10KΩ	C2	0.1µF
Q1	NPN	R3	33KΩ	R6	10KΩ			C3	0.1µF

In project 163, you saw how a flip-flop circuit can be "toggled" so that we can have additional control over it.

You can see by the schematic that you'll provide the clock signal for this circuit each time you press **S1**. The signals for the J and K inputs are provided by **S2** and **S3**.

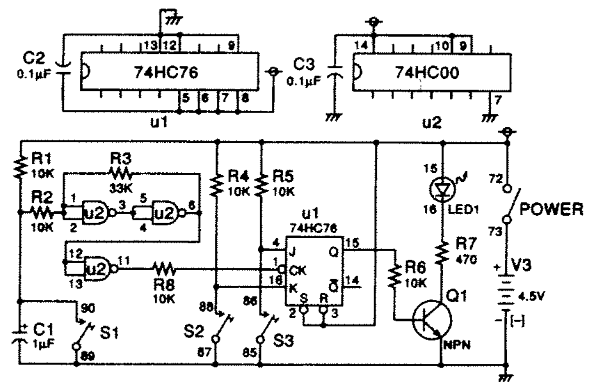
After you build this project, turn power ON and press **S1** a few times. You'll see **LED 1** go on and off. Now press **S3**. While you pressing it, press **S1** a few times. What happens to **LED 1** ? Release **S3** and press **S1** again. Is there any difference in what **LED 1** does?

Now press **S1** while pressing **S2** several times. What does **LED 1** do now? Release **S2** and press **S1** again. What happens to **LED 1** ?

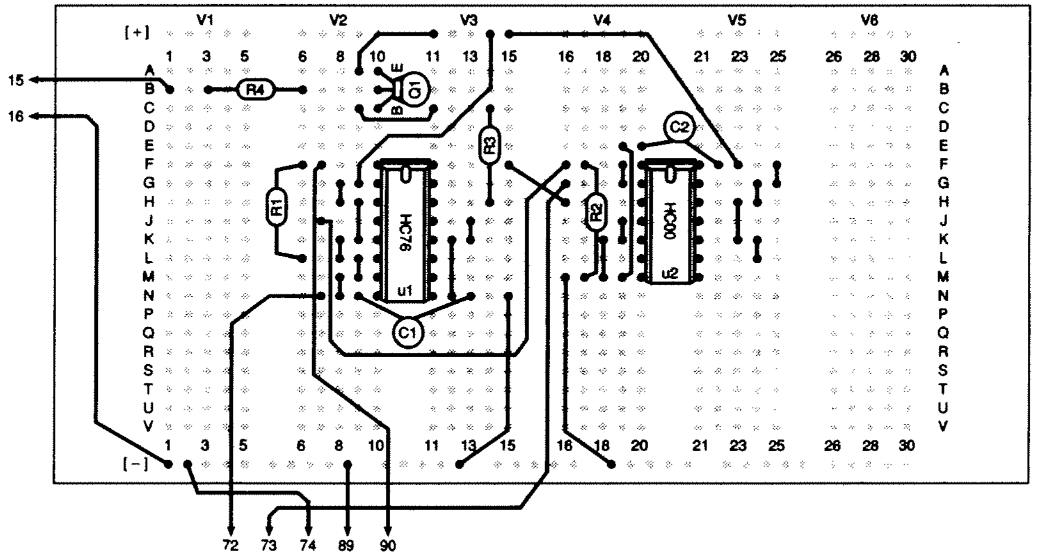
As you may have figured out, if both the J and K inputs are 1, the flip-flop sets and resets each time you press **S1** (which inputs a clock signal). This means **LED 1** goes on and off. But if you press **S3**, this makes **LED 1** goes out when the flip-flop reverses a clock signal and stays out.

When you release **S3** andpress **S2**, the J input is 1 while the K input is 0. This means **LED 1** lights with a clock signal and stays lit.

Time to put on your thinking cap - can you think of some of the interesting things and could be done of you used different types of multivibrator circuits for the clock, J and K inputs? Try whipping up some circuits to see if you're right.



PROJECT 168. C-MOS D FLIP-FLOP

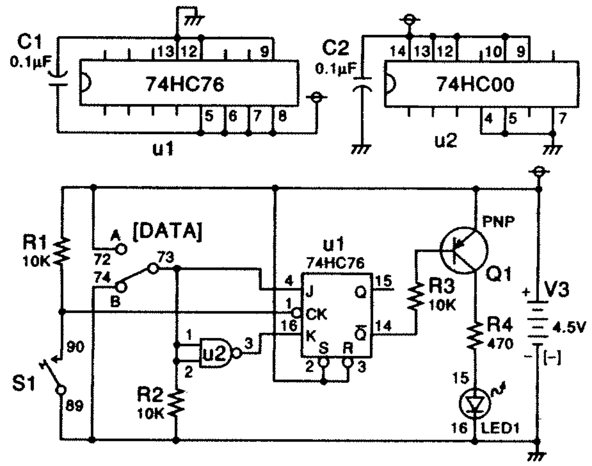


U1	74HC76	Q1	PNP	R1	10KΩ	R3	10KΩ	C1	0.1μF		
U2	74HC00			R2	10KΩ	R4	470Ω	C2	0.1μF		

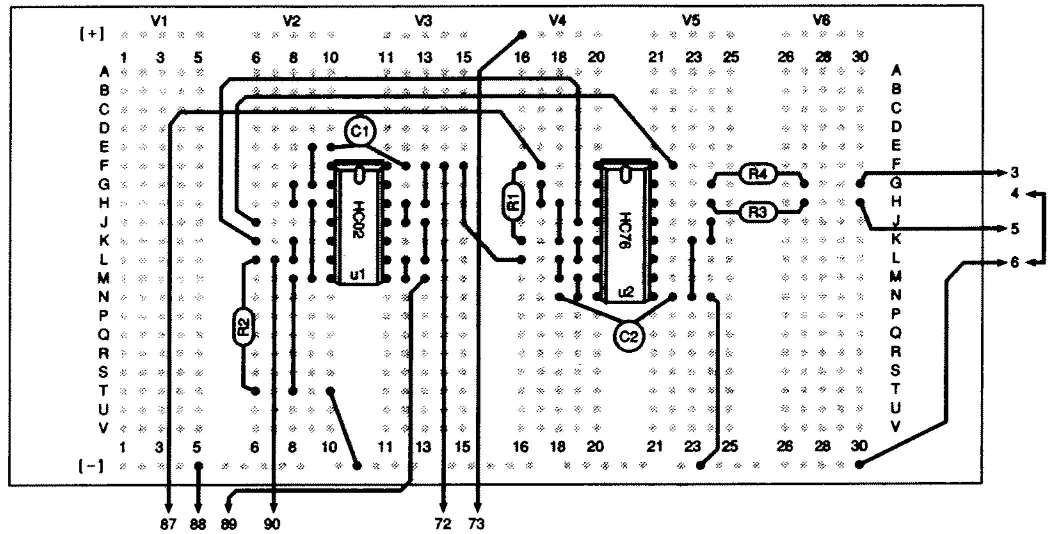
Can you guess what the "D" in C-MOS D Flip-Flop stands for? Make a quick mental note because you're about to find out.

As you build this experiment, set the **select switch** either up or down. Press **S1** and release it. If you set the **select switch** up, **LED 1** lights up. If the **select switch** is down, **LED 1** goes out. Now try changing the position of the **select switch** up and down, without pressing **S1**. Can you get **LED 1** to change without pressing **S1**?

As you can see on the schematic, pressing **S1** inputs a clock signal. There's also an input signal provided by setting the **select switch** up or down. The "D"- in this circuit's name comes from "delay" no matter how the J or K input changes, the output changes is delayed until clock signal is received.



PROJECT 169. C-MOS D FLIP FLOP II



U1	74HC02	R1	100KΩ	R3	470Ω	C1	0.1μF
U2	74HC76	R2	100KΩ	R4	470Ω	C2	0.1μF

We're now going to experiment with a D flip-flop circuit to find out how it works. The D flip-flop is performed by the section surrounded by a dotted line in the schematic. Note it has two input terminals (D terminal and CLK terminal) and two output terminals (Q and Q̄).

Figure 1 shows you that if you turn D terminal to 1 by pressing **S1** and send a clock pulse to CLK terminal by pressing **S2**, the 1 and 0 levels are inverted at the output terminals Q and Q̄, and Q becomes 1. The output level doesn't change even if you apply 0 to D terminal ,or 1 to the CLK terminal. This means that the pulse sent to the CLK terminal is "remembered" by Q. If you press **S2** and send another pulse to CLK terminal, Q and Q̄ are inverted again, and this is also "remembered."

Assemble the project and turn power ON, and you'll see that either **LED 6** to **LED 7** lights up. If **LED 7** lights up, press **S2**, and this results in **LED 6** lighting up and **LED 7** going out. If **LED 6** lights up, press **S2** while pressing **S1**, and you'll notice **LED 7** lights up and **LED 6** goes out. Now press **S2** again and see what happens this time. **LED 6** lights up and **LED 7** goes out.

Repeat this experiment while looking at Figure 1 so you can obtain a good understanding of the D flip-flop function.

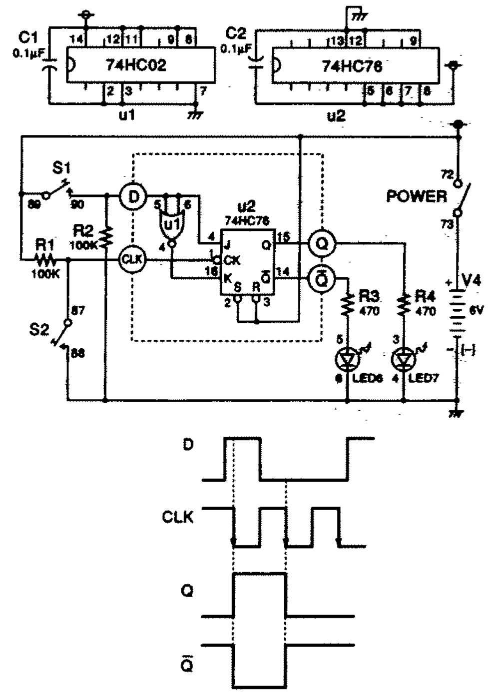
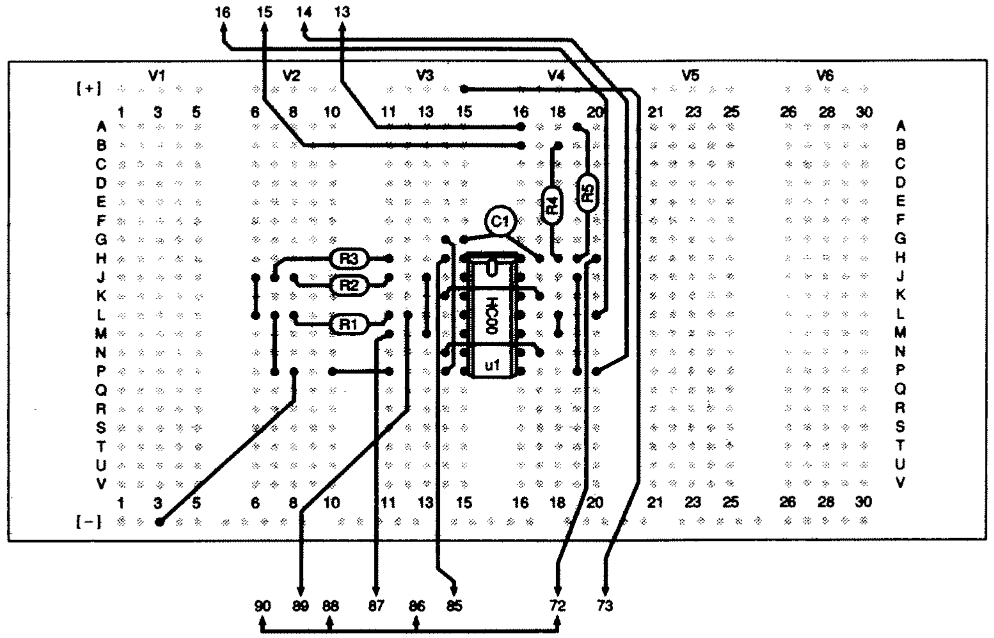


Figure 1

PROJECT 170. R-S-T FLIP FLOP



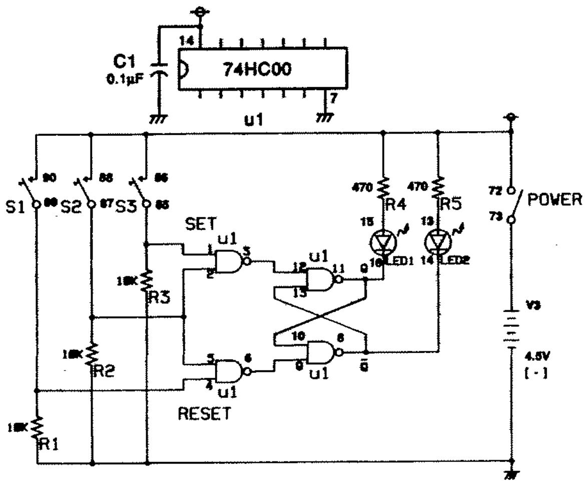
U1	74C00	R2	10KΩ	R4	470Ω	C1	0.1μF
R1	10KΩ	R3	10KΩ	R5	470Ω		

The R-S-T flip-flop circuit we're going explore in this project can reverse any seesaw-like action applied to it, and keep that reversed state.

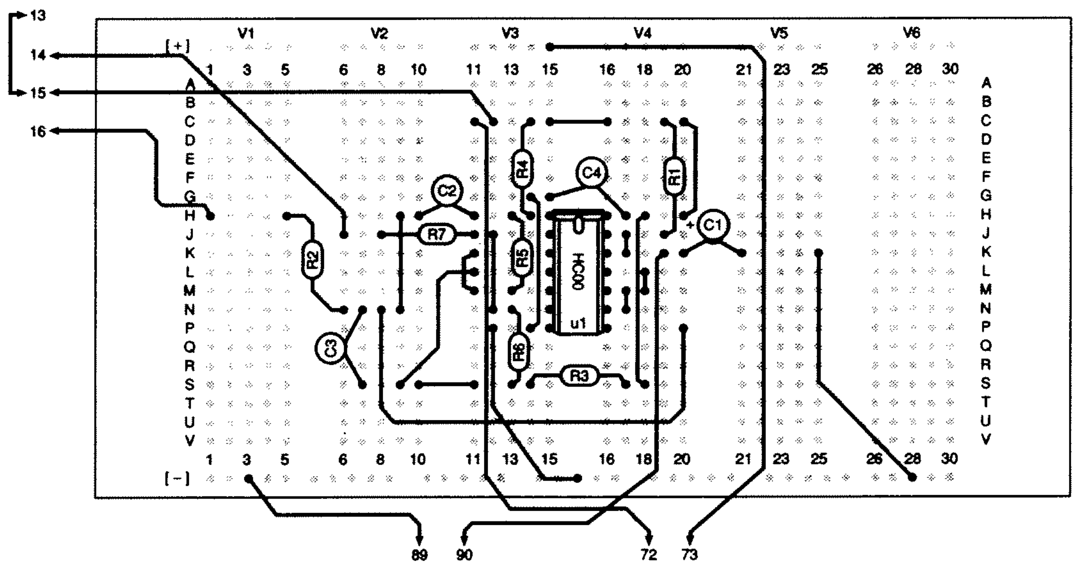
The schematic diagram shows that this circuit has three inputs (Set, Reset and Trigger) and two outputs (Q and $\overline{Q}$). The diagram indicates the relationship between these inputs and outputs.

Now, let's see how it works. Turn power ON and press **S1** and **S2**. **LED 1** lights up and **LED 2** goes out. When you press **S2** and **S3**, **LED 1** goes out and **LED 2** lights up, and here you'll notice something.... yes, the circuit maintains this state.

This means that the flip-flop can "remember" things. So, it's used in many different electronic devices such as memories, counters and so on.



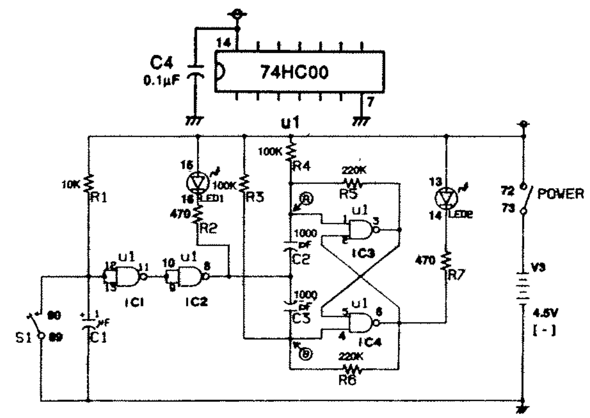
PROJECT 171. T TYPE FLIP FLOP



U1	74HC00	R3	100KΩ	R6	220KΩ	C1	1μF	C3	1000pF
R1	10KΩ	R4	100KΩ	R7	470Ω	C2	1000pF	C4	0.1μF
R2	470Ω	R5	220KΩ						

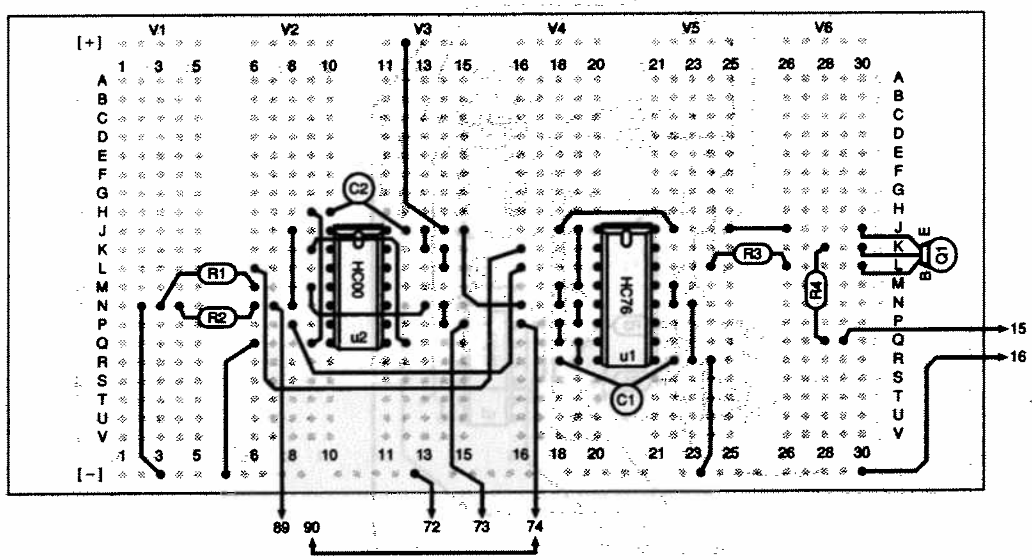
We're going experiment with a T flip-flop. It has one input terminal and two output terminals. What do you suppose T stands for? You've already met... it's a "toggle" flip-flop.

When you finish the wiring for the project, turn power ON, tap **S1** a few times and see what happens to **LED 2**. It blinks ON and OFF each time you tap **S1**. When you press **S1** once, a pulse is input and the toggle flip-flop changes its state. When you press it again, another pulse is input and this causes the flip-flop to return to the former state.



The two output terminals of this circuit are pin 3 of IC3 and pin 6 of IC4. We are using only pin 6 of IC4 in this experiment, but note that these terminals have an inverse relationship with one other. When one of them is ON, the other is OFF, and vice versa.

PROJECT 172. C-MOS LATCH



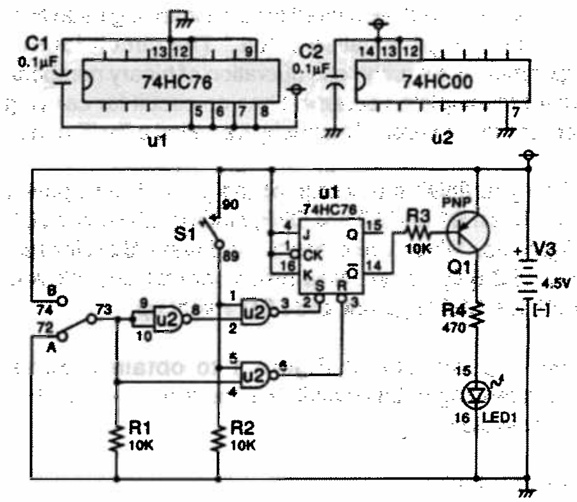
U1	74HC76	Q1	PNP	R1	10KΩ	R3	10KΩ	C1	0.1µF
U2	74HC00			R2	10KΩ	R4	470Ω	C2	0.1µF

This is another circuit you can see how it works by its name - so make a guess before building this project and finding out for sure.

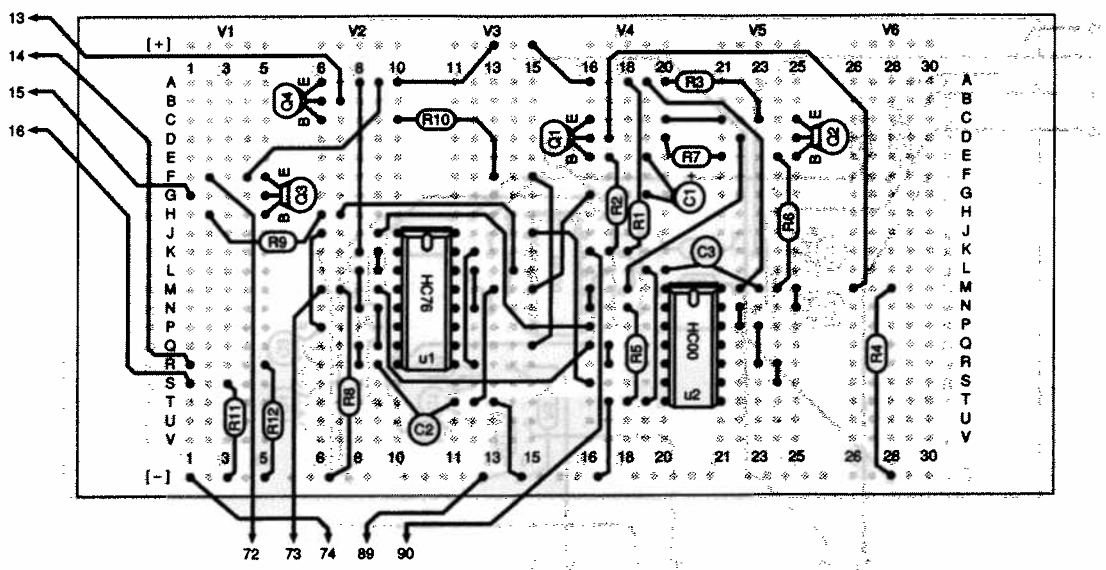
As you build this project, set the **select switch** up. Then, press **S1**. You'll see **LED 1** light up. Now set **select switch** down. What does **LED 1** do?

Now with the **select switch** down, press **S1** again. You'll see that **LED 1** goes out. Move the **select switch** up and back again. Is there any change in **LED 1**?

You can easily see where the latch circuit gets its name. The circuit can "latch" in a certain condition and keep the same output regardless of changes in the input. We can use latch circuits to control the operations of electronic devices since it maintains a certain output until we send a signal to cause a change - we don't have to keep sending an input signal to the latch circuit continuously. This means just a touch can activate a latch circuit... just as you saw in this project.



PROJECT 173. SHIFT REGISTER



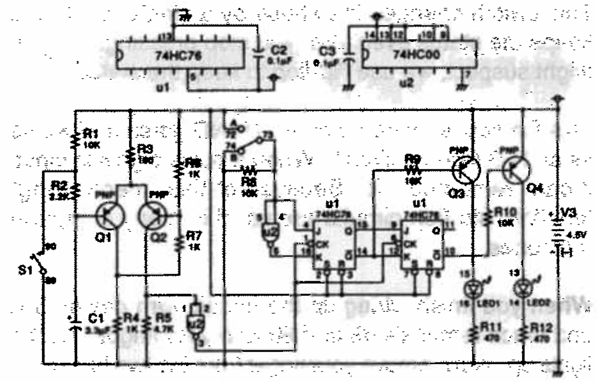
U1	74HC76	Q4	PNP	R5	4.7KΩ	R9	10KΩ	C1	3.3µF
U2	74HC00	R1	10KΩ	R6	1KΩ	R10	10KΩ	C2	0.1µF
Q1	PNP	R2	2.2KΩ	R7	1KΩ	R11	470Ω	C3	0.1µF
Q2	PNP	R3	100Ω	R8	10KΩ	R12	470Ω		
Q3	PNP	R4	1KΩ						

What's a shift register? If you've used an electronic calculator, you're already seen one in action. This project is a simple shift register that lets you discover how the circuit works.

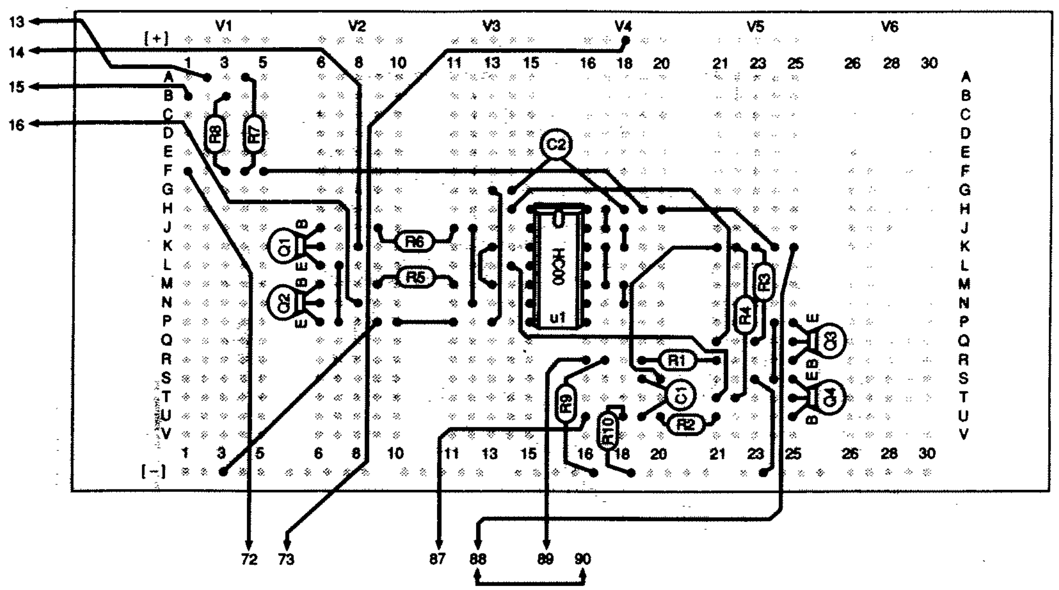
There's a lot of connections to be made in wiring this project, so take your time and double-check your work as you go along. Set the **select switch** down. Now press **S1** several times, you'll see that both **LED 1** and **LED 2** are off. Now move the **select switch** up and press **S1** once. **LED 1** lights. Now press **S1** again and **LED 2** lights as well. Both stay lit no matter how many more times you press **S1**.

Now set the **select switch** down and press **S1** once. What happens? Press **S1** again. Is there any change? Keep pressing **S1** a few more times and see if anything else happens.

A shift register is a circuit used to store information by shifting it from one flip-flop to another. You saw this shift takes place as you press **S1** and the **LEDs** lit or went off. In calculators, a more complex version of this circuit causes the number "10" to appear on the calculator display when you add 1 to 9.



PROJECT 174. TOUCH SWITCH USING NAND GATE

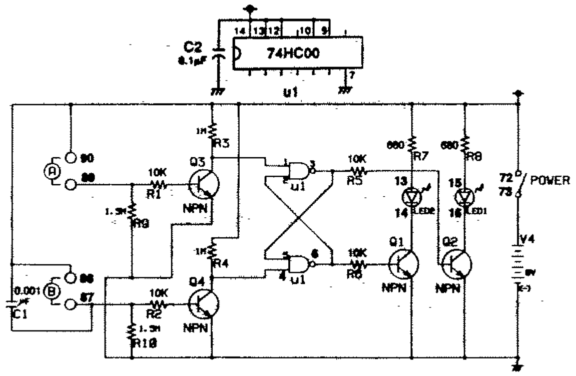


U1	74HC00	Q4	NPN	R4	1MΩ	R8	680Ω	C1	0.001μF
Q1	NPN	R1	10KΩ	R5	10KΩ	R9	1.5MΩ	C2	0.1μF
Q2	NPN	R2	10KΩ	R6	10KΩ	R10	1.5MΩ		
Q3	NPN	R3	1MΩ	R7	680Ω				

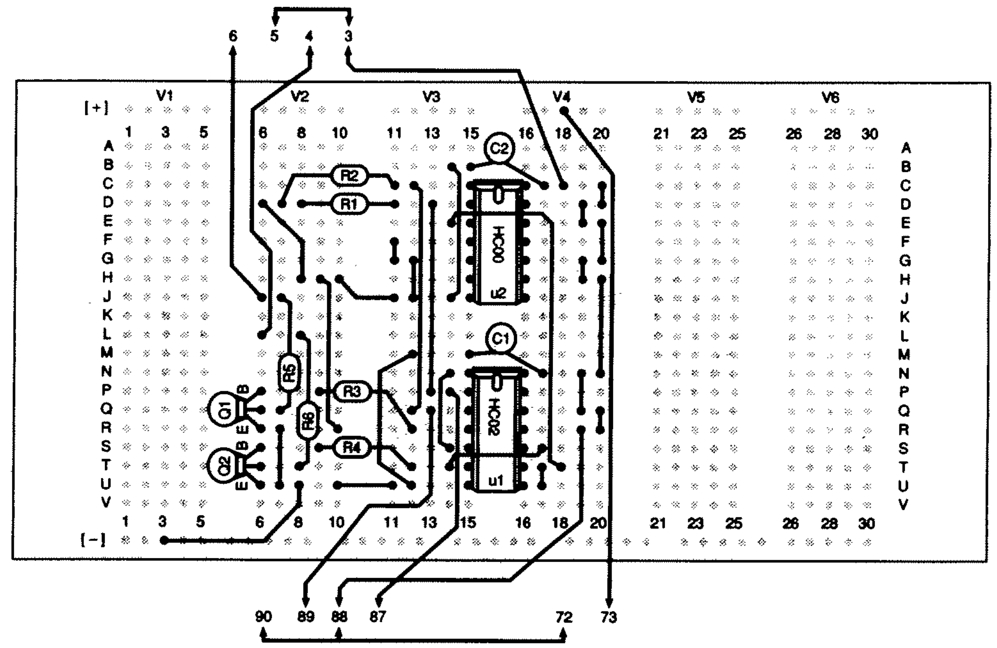
This switch changes the status by a single touch and keeps the status even after you stop pressing. As you might suspect, we use flip-flop to keep the status.

This flip-flop is made up of two NAND gates and works as an R-S flip-flop circuit. When a pulse signal is input, it can "remember" it. Because of this "memorizing" function, flip-flop circuits are used in many computer memories.

When you finish wiring up the circuit, turn power ON, and touch terminals **89** and **90** with your finger. **LED 1** lights up. Now remove your finger from the two terminals, and you'll notice **LED 1** stays ON. Now touch terminals **87** and **88** and see what happens. This time **LED 1** goes out and **LED 2** lights up. Do you see **LED 2** still ON?



PROJECT 175. HALF ADDER



U1	74HC00	Q1	NPN	R1	47KΩ	R4	10KΩ	C1	0.1μF
U2	74HC02	Q2	NPN	R2	47KΩ	R5	680Ω	C2	0.1μF
				R3	10KΩ	R6	680Ω		

A half adder is a device used for the binary digit addition. The binary digit is called a "bit." The circuit for this experiment is for arithmetic operation of binary numbers, but it's incomplete because it has no circuit for carrying from the lower order. That's why it's called a "half" adder.

When you finish the wiring for the project, turn power ON. When **S1** and **S2** are OFF, neither **LED 6** nor **LED 7** lights up. When you press **S1** and leave **S2** released, only **LED 6** lights up. Now press both **S1** and **S2**, and you'll see **LED 6** go out and **LED 7** light up.

Take a good look at Figure 1 to obtain a better understanding of this circuit. Remember, 1 + 1 is 10 in binary system.

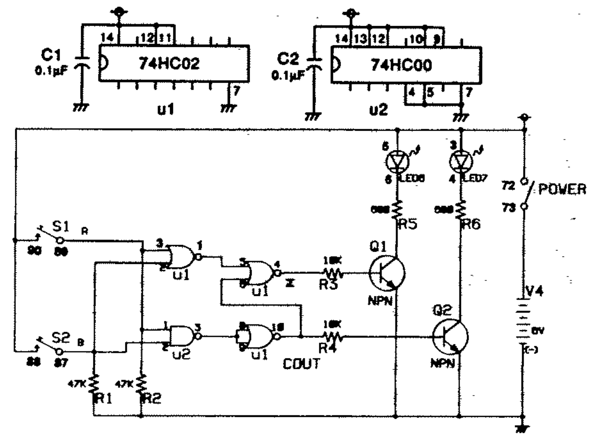


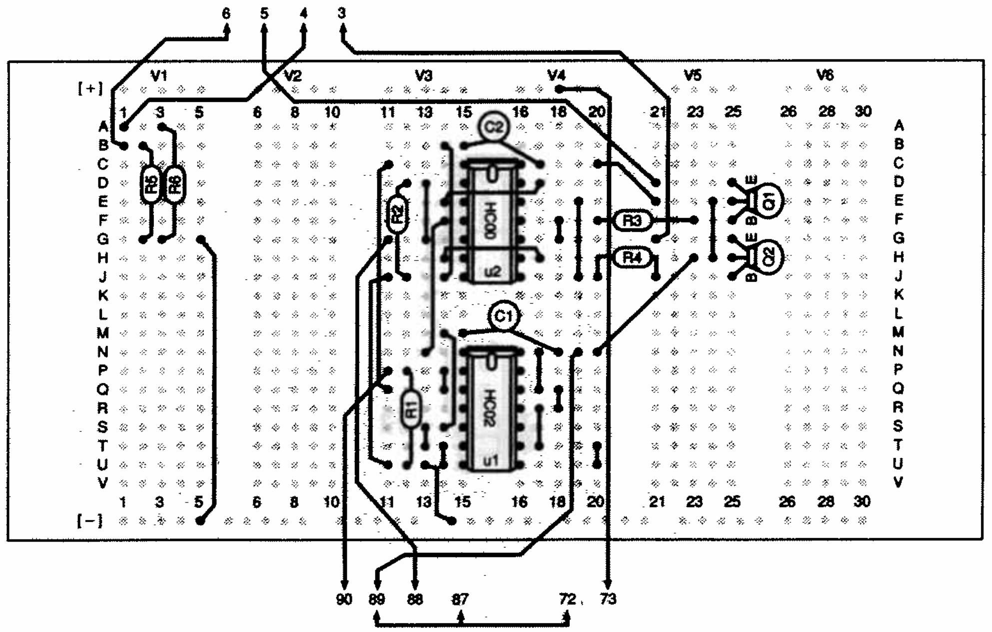
Table conteuts:
A, B : Inputs to be added
Σ : Added result
Cout : Carry out for the upper digit

A	B	Σ	C out
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

LED ON : 1
LED OFF: 0

Figure 1

PROJECT 176. D-LATCH



U1	74HC02	Q1	NPN	R1	100KΩ	R4	10KΩ	C1	0.1μF
U2	74HC00	Q2	NPN	R2	100KΩ	R5	680Ω	C2	0.1μF
				R3	10KΩ	R6	680Ω		

Let's experiment with a D flip-flop circuit with a latch function. The latch is a kind of memory function, so this circuit can store data produced at a certain moment. Just in case you forget, D flip-flop stands for "delay flip-flop."

As you'll see from block B of the schematic, this D flip-flop circuit is made up of D (data terminal). T (clock terminal), and two output terminals Q and $\overline{Q}$. It works by the ON/OFF operation of **S1** and **S2** as shown in Figure 1.

In this project, the output level is 1 when the **LED** is ON and 0 when it is OFF. When you finish wiring the circuit, turn power ON, and you'll notice **LED 6** or **LED 7** light up.

a) Turn **S1** ON and OFF with **S2** released, and the **LED** stays ON.

b) Turn **S1** ON and OFF while pressing **S2**, and **LED 6** and **LED 7** take turns blinking.

c) Now, release **S2** after **LED 6** is lit, then turn **S1** ON and OFF, and you'll see **LED 6** stay ON.

d) Now press **S2**, and **LED 6** goes out and **LED 7** lights up.

Where in the above operation do you think data is stored? It's stored when you come to step c). The circuit doesn't accept the signal from **S1** once it stores data at this step. Try to figure out why it works this way by checking its function with Figure 1.

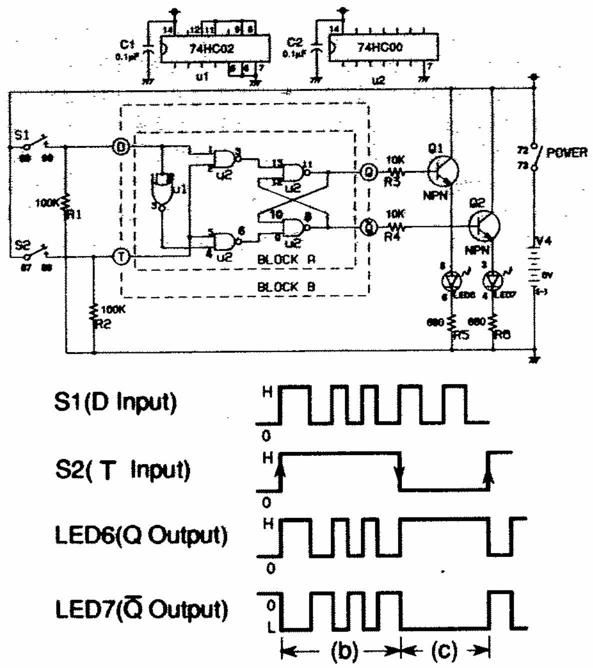
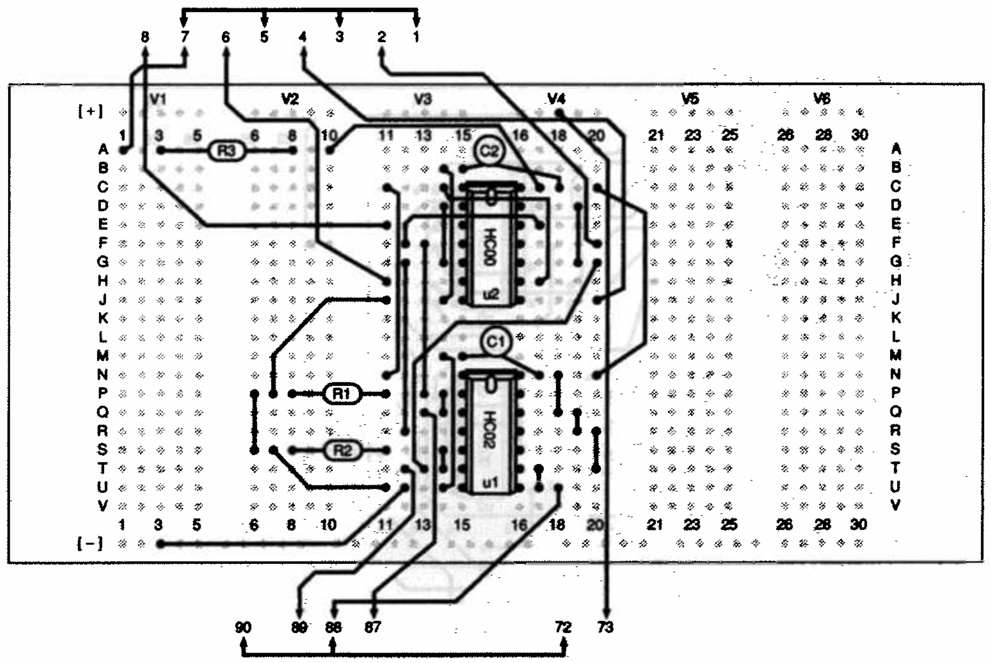


Figure 1

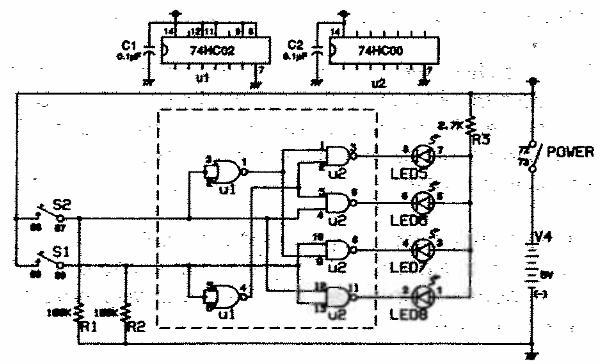
PROJECT 177. 2-LINE TO 4-LINE DECODER



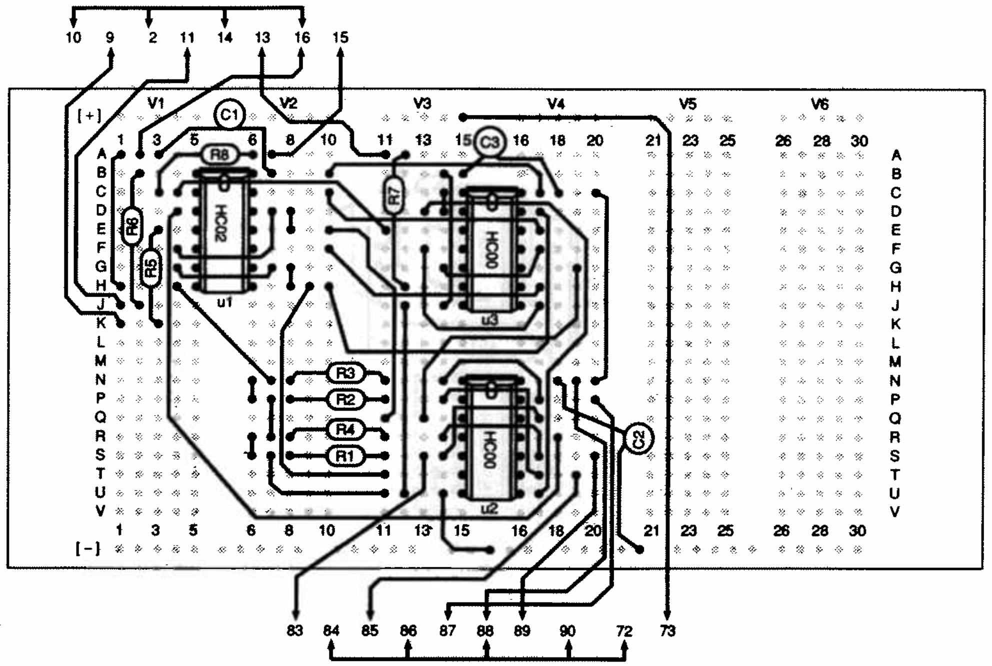
U1	74HC02	R1	100KΩ	R3	2.7KΩ	C1	0.1μF
U2	74HC00	R2	100KΩ			C2	0.1μF

Here's a circuit that has two terminals with four different combinations of 1 and 0 levels (2x2 = 4). The 2-line to 4-line decoder we're going to experiment with here can output according to each of these four input levels. With this decoder, we can find out the kind of input applied simply by watching the **LEDs**.

Wire the project, turn power ON, and see how this decoder works when you press and release **S1** and **S2**. You'll notice that **LED 5** lights up when **S1** and **S2** are both OFF. Press **S2**, and **LED 6** lights up. Now press **S1**, and **LED 7** lights up. When you press both **S1** and **S2**, **LED 8** lights up. Did you notice that only one **LED** lighted whichever switch you pressed? If you didn't, press **S1** and **S2** while watching the **LEDs** closely, and you'll see that it's true.



PROJECT 178. MULTIPLIER



U1	74HC02	R1	100KΩ	R4	100KΩ	R7	1KΩ	C1	0.1μF		
U2	74HC00	R2	100KΩ	R5	1KΩ	R8	1KΩ	C2	0.1μF		
U3	74HC00	R3	100KΩ	R6	1KΩ			C3	0.1μF		

Here's a multiplier that can multiply of two numbers.

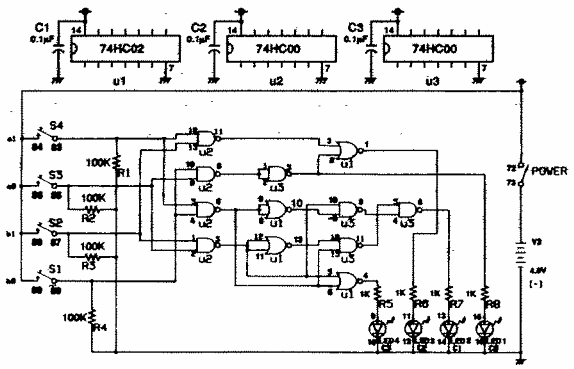
You'll see from the schematic that the two numbers that can be multiplied by this circuit are input by **S1 - S4** - one of them by **S1** and **S2** and the other by **S3** and **S4**. This means that each number is expressed by two bits, so the input numbers are limited to 0 - 3.

Figure 1 shows the input number combinations and the results of multiplication. Output is expressed by 4-bit binary numbers produced by C0-C3 and displayed by **LED 1 - LED 4**.

When you finish assembling, turn power ON and see how this multiplier works. Remember that 1 is indicated when any of **S1 - S4** is ON and 0 when OFF.

An example of multiplication: When you input 1 (a1 = 0, a0 = 1) to A and 3 (b1 = 1, b0 = 1) to B, only **LED 1** and **LED 2** light up. Check this ON-OFF combination in Figure 1 and you'll find out that it means 3 which is the right answer.

See how this multiplier circuit functions? Now input a1 = 0, a0 = 1, b1 = 1 and b0 = 0, and write down the answer in the blank columns in Figure 1.



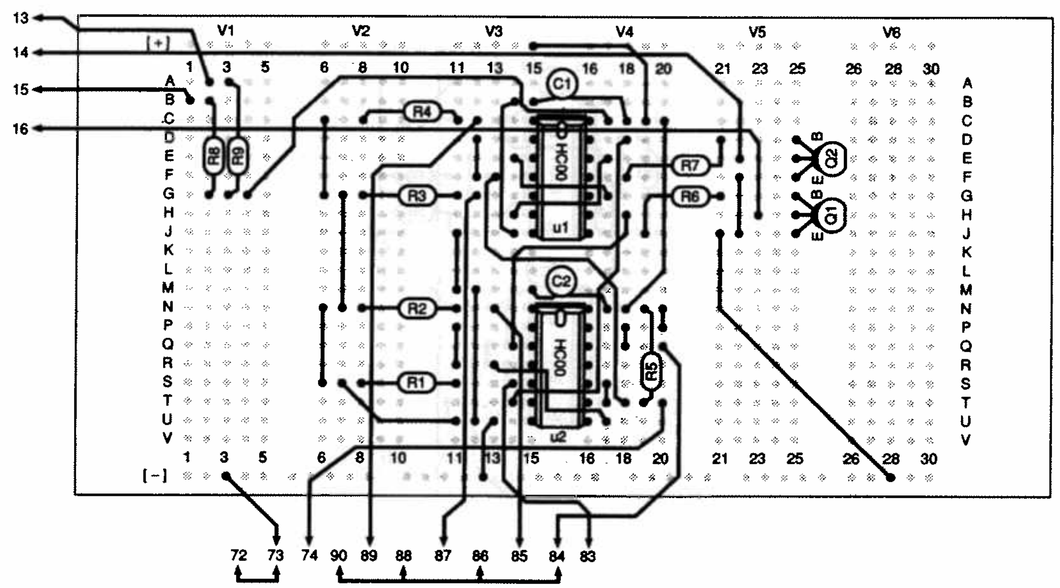
A			B			C				
Hexa	Binary		Hexa	Binary		Hexa	Binary			
	a_1	a_0		b_1	b_0		C_3	C_2	C_1	C_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0	0
0	0	0	3	1	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	0	0	1
1	0	1	2	1	0	2	0	0	1	0
1	0	1	3	1	1	3	0	0	1	1

A			B			C				
Hexa	Binary		Hexa	Binary		Hexa	Binary			
	a_1	a_0		b_1	b_0		C_3	C_2	C_1	C_0
2	1	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	2	0	0	1	0
2	1	0	2	1	0	4	0	1	0	0
2	1	0	3	1	1	6	0	1	1	0
3	1	1	0	0	0	0	0	0	0	0
3	1	1	1	0	1	3	0	0	1	1
□	1	1	□	1	0	□	□	□	□	□
3	1	1	3	1	1	9	1	0	0	1

LED ON : 1 or H
LED OFF: 0 or L

Figure 1

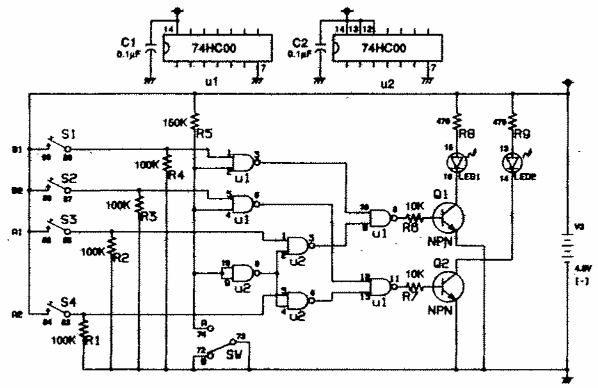
PROJECT 179. DUAL 2-INPUT MULTIPLEXER



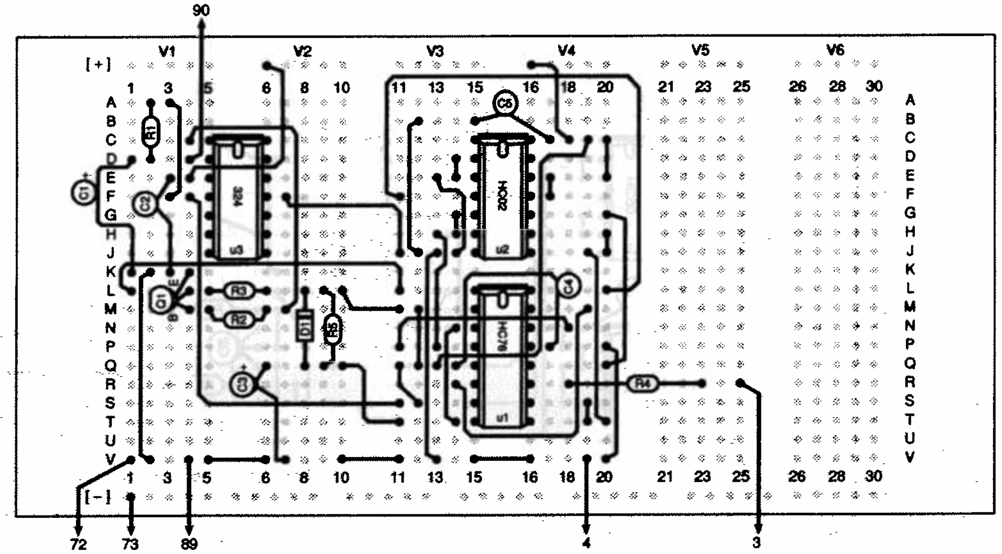
U1	74HC00	R1	100KΩ	R4	100KΩ	R7	10KΩ	C1	0.1μF
U2	74HC00	R2	100KΩ	R5	150KΩ	R8	470Ω	C2	0.1μF
Q1	NPN	R3	100KΩ	R6	10KΩ	R9	470Ω		
Q2	NPN								

Did you know that a switch that works just like a double pole toggle switch can be made by combining NAND gates? That's what we're going to build right now. Switches of this kind are used widely for digital signal switching.

When you finish assembling the project, set **select switch** up, and press **S1** and **S2** at the same time. This results in **LED 1** and **LED 2** lighting up. Now set the **select switch** down and press **S3** and **S4** at the same time. **LED 1** and **LED 2** blink ON and OFF again, but this time they do not light up even if you press **S1** and **S2**. Can you guess what this means? It means that the A1, A2 circuits and the B1, B2 circuits are switched over.

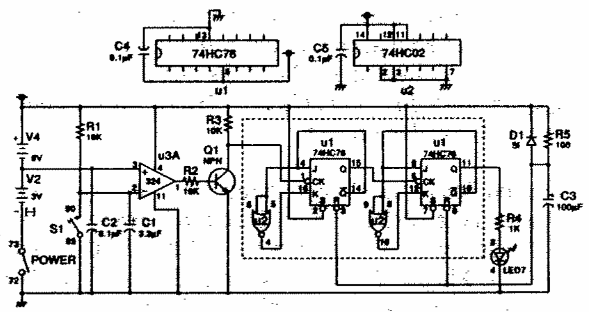


PROJECT 180. TWO-STAGE FREQUENCY DIVIDER



Q1	NPN	R3	10KΩ	C1	3.3μF	C4	0.1μF	D1	Si
R1	10KΩ	R4	1KΩ	C2	0.1μF	C5	0.1μF		
R2	10KΩ	R5	100Ω	C3	100μF				

A flip-flop generates one output when it receives two input pulses. In this project, we're going to make use of this flip-flop function to make a frequency divider. We'll use two flip-flop circuits to divide a frequency to one fourth.

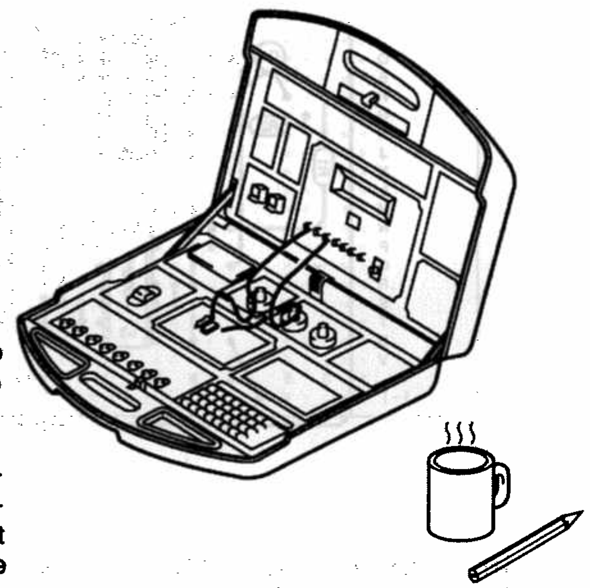


Take a look at the schematic for the project. A J-K flip-flop IC is combined with a NOR gate circuit to make up the D flip-flop circuit (surrounded by a dotted line). Since we use two D flip-flop circuits each capable of dividing a frequency to its half, we can make a 1/4 frequency division (1/4 = 1/2 x 1/2).

When you finish wiring up, turn power ON and press **S1**. **LED** 7 stays off. Press **S1** again, and this time **LED 7** lights up. Press **S1** for the third time. **LED** 7 stays lit. Press **S1** for the fourth time, and **LED 7** goes out. This means you need to send four pulses to let the **LED** go one cycle of operation --1/4 frequency division.

Coffee Break

Edison, the great inventor



Do you know the great American inventor, Thomas Edison (1847-1931)? He is the inventor of the incandescent lamp. When a current is run through a filament inside a vacuum, in an incandescent lamp, the heat generated by the filament turns into light. After trying out many types of filament materials, Edison's experiment finally succeeded when he used Japanese bamboo, according to his legend.
Surprisingly, Edison attended school for only 3 months in his entire life. The rest of his formative education came from his mother. Now, don't let that alarm you to think you are over educated. There will always be much left to be invented for as long as you live, and no matter how long you live.

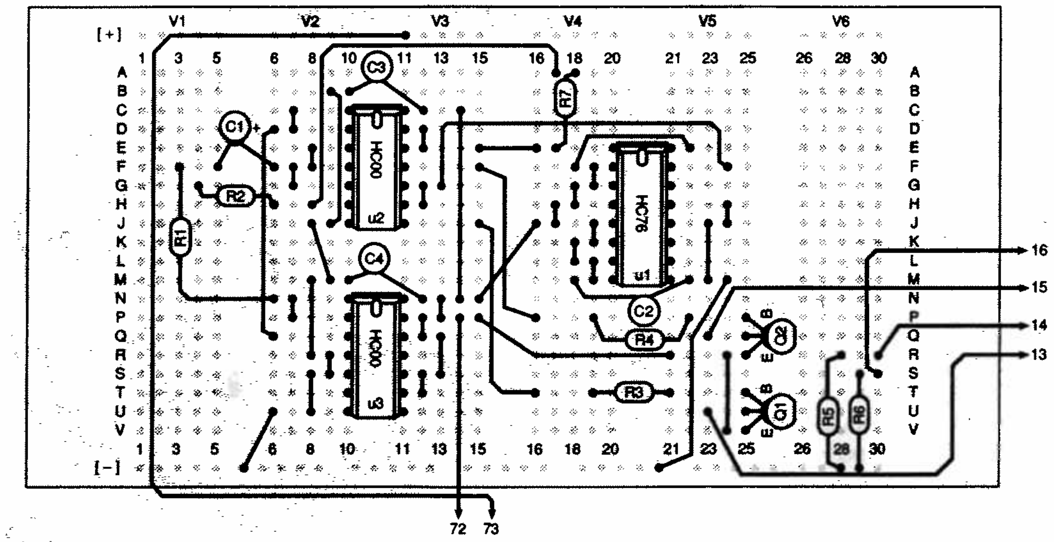
Edison once worked as a temporary telegraph assistant, and it is said that his first invention was an automatic telegraph relay. He also made the world's first commercial power generator, and is known to have started the business of supplying power.
The list of his inventions go on and on. After all, he ended up with over 1,000 patents. There was even the "Edison Battery." Many of his inventions were involved with motion pictures, such as the foundations that made talking-films possible. Before talking-films became available, we only had silent films (like Charlie Chaplin movies).
Also, the thermionic emission phenomenon (Edison Effect), discovered by Edison, was later used in the vacuum tube, which is why we have radios and TV sets today (although vacuum tubes have been replaced by transistors). Isn't that enough to make you feel like we owe a lot to Mr. Edison?

It was over 400 years ago when William Gilbert studies static charges and coined them "electrical." Back then, physicians and physicists were the principle people involved in advancing the science of electricity, and progress was slow and tedious. Edison might be the first inventor that we can call an "electrician" in the true sense of that word.

♦Right hand or left hand?
Do you use your right hand or your left hand when you change a light bulb? Depends on whether you are right handed or left handed? Well, that is probably a sensible answer from a practical standpoint, but you should probably use your right hand.
Do you know why? The reason is because your heart is on your left side. So, you can protect your heart from electric shock by taking advantage of your right hand.

♦Is the same AC voltage used everywhere in the world?
The voltage of alternating currents (AC) varies from country to country, such as 115V, 220V, 230V, 240V, and 100V, etc.
So, it is probably a bad idea to take your hair drier to a country that uses a higher voltage. Needless to say, your drier's heater will burn out.

PROJECT 181. BASIC COUNTER

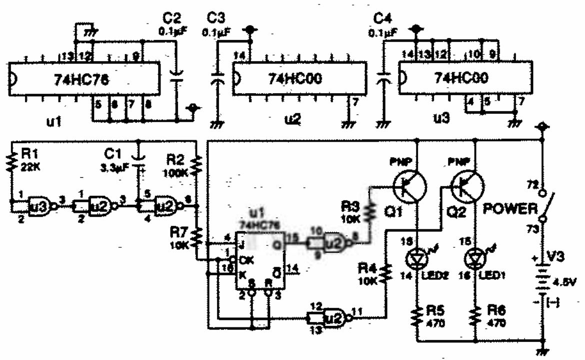


U1	74HC76	Q2	PNP	R4	10KΩ	C1	3.3μF
U2	74HC00	R1	22KΩ	R5	470Ω	C2	0.1μF
U3	74HC00	R2	100KΩ	R6	470Ω	C3	0.1μF
Q1	PNP	R3	10KΩ	R7	10KΩ	C4	0.1μF

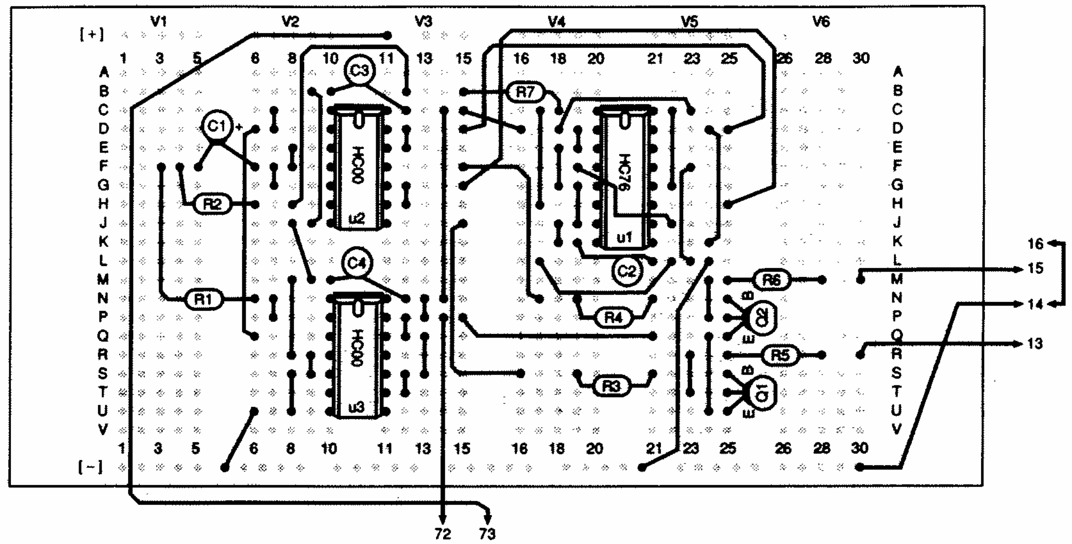
We've mentioned from time to time in earlier projects that digital circuits can "count." Here's an example project of a very simple counter circuit.

You'll see a NAND multivibrator circuit in the schematic. The output of the multivibrator serves a double purpose here - it turns **LED 1** on and off and also serves as a clock impulse for the flip-flop. Turn power ON. **LED 1** and **LED 2** flash on and off in turn.

Think about this circuit for a moment: you'll realize that it is actually counting the pulses made by the multivibrator. Of course, since there's only two pulses from the multivibrator, this counter circuit doesn't count very high. Yet the same principle is used for other counter circuits. Can you figure out how to use flip-flops to make counters which can count higher? Keep notes on what you think... because we're going to find out how in the next few projects.



PROJECT 182. SYNCHRONOUS COUNTER

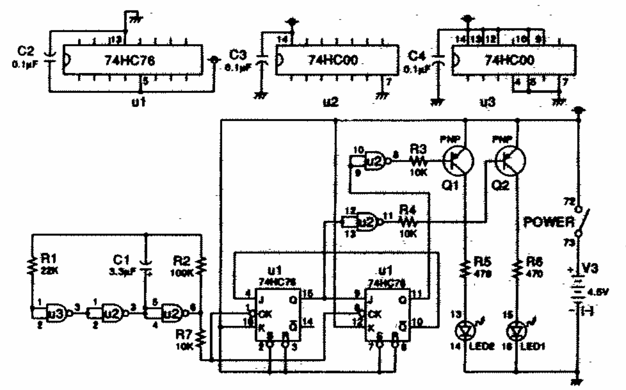


U1	74HC76	Q2	PNP	R4	10KΩ	C1	3.3μF
U2	74HC00	R1	22KΩ	R5	470Ω	C2	0.1μF
U3	74HC00	R2	100KΩ	R6	470Ω	C3	0.1μF
Q1	PNP	R3	10KΩ	R7	10KΩ	C4	0.1μF

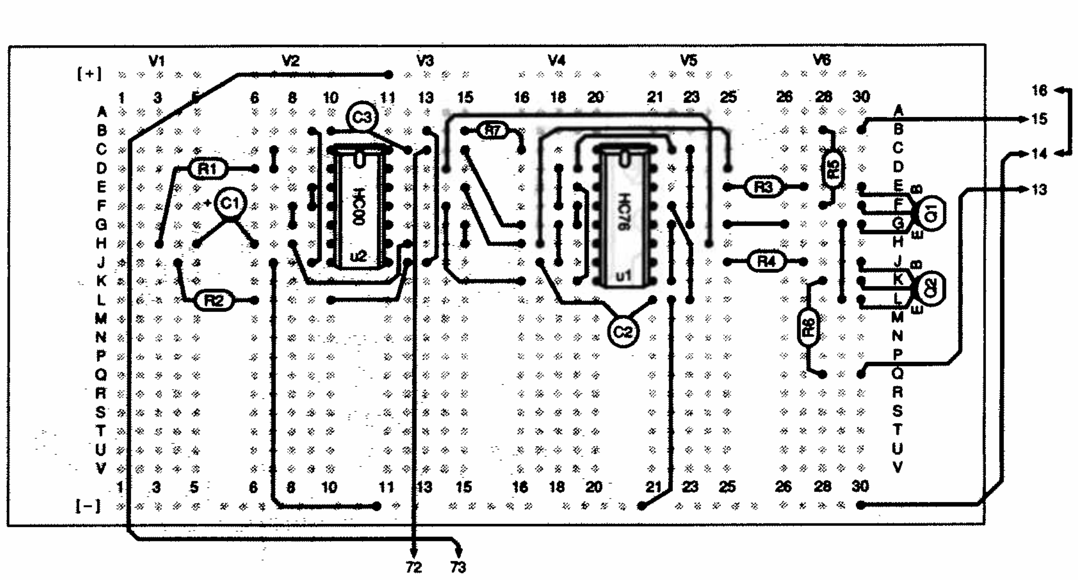
Here's another counter circuit which can count up to three. It's called a synchronous counter - do you have any idea what that means? And can you guess how it can count to three using only two **LEDs**? Take a guess before building this project.

Once you finish the wiring connections, turn power ON and watch the **LEDs**. Both **LEDs** are blinking. When both **LEDs** are off, that stands for 1. Then **LED 1** lights while **LED 2** is off. That's the same as 2. Finally, **LED 2** lights while **LED 1** is off. That's 3 so this circuit can really count up to three. You'll see the **LEDs** both go out after three is reached and the entire cycle starts over.

A synchronous counter is one where the clock signal is input to all flip-flops at the same time. You can see from the schematic that each flip-flop gets a clock signal directly from the multivibrator. This is how we manage to keep the two flip-flops "in step" with each other - something very important in a counting circuit.



PROJECT 183. ASYNCHRONOUS COUNTER



U1	74HC76	R1	22KΩ	R5	470Ω	C1	3.3μF
U2	74HC00	R2	100KΩ	R6	470Ω	C2	0.1μF
Q1	PNP	R3	10KΩ	R7	10KΩ	C3	0.1μF
Q2	PNP	R4	10KΩ				

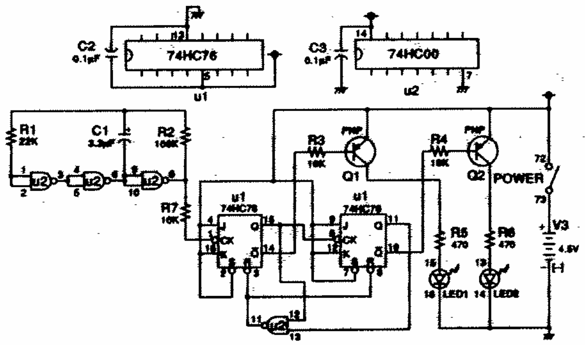
Since we just found out about synchronous counters, can you guess what an asynchronous counter is and how it works? Make it quick - you're about to build one.

You'll find this project works almost exactly like the counter in the previous project. When you turn power ON, you'll see that the **LEDs** operates as follows. After both **LEDs** are off, then **LED 1** lights while **LED 2** stays off. The **LED 2** lights while **LED 1** goes out. Then both **LEDs** are off and the cycle starts all over again.

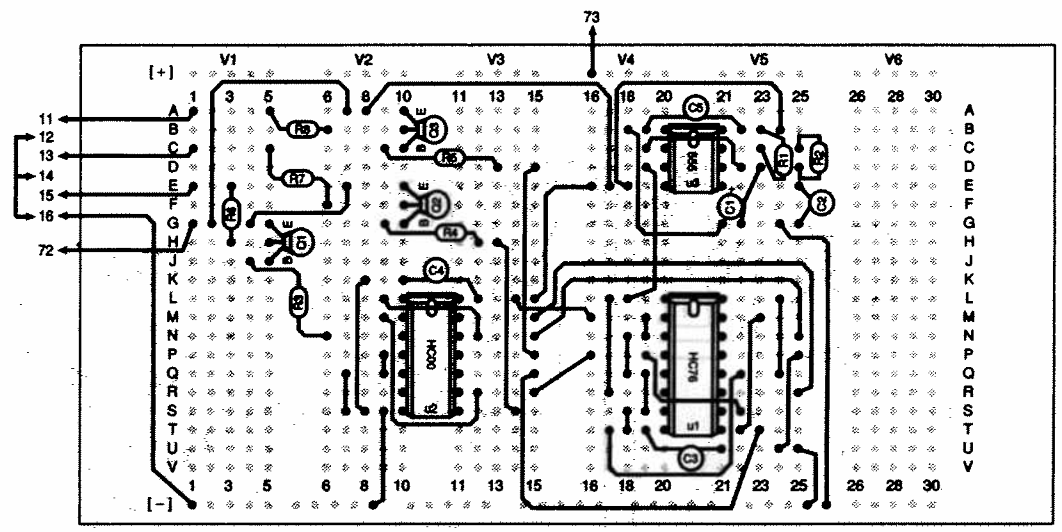
The key difference between synchronous and asynchronous counters can be found by comparing the schematic diagram of this project and the last one. In a synchronous counter each flip-flop has its own clock signal input. But in an asynchronous counter such as this one, the clock input for the second flip-flop comes from one of the outputs of the first flip-flop. It comes from the Q output of the first flip-flop in this project.

This difference might seem to be trivial but it's not. It takes time for the first flip-flop to set or reset and produce a clock signal for a second flip-flop. When many flip-flop circuits are used (such as in a computer), this delay from the first flip-flop to the second can produce quite an error. That's why synchronous counters are always preferred in circuits that are elaborate or where accuracy is important.

In fact, you might be able to notice one side effect of an asynchronous circuit now. Look carefully at the **LEDs** in this project as they light on and off - can you spot any occasion where both **LEDs** are lit (even if only briefly)?



PROJECT 184. COUNTER WITH LINE DECODER



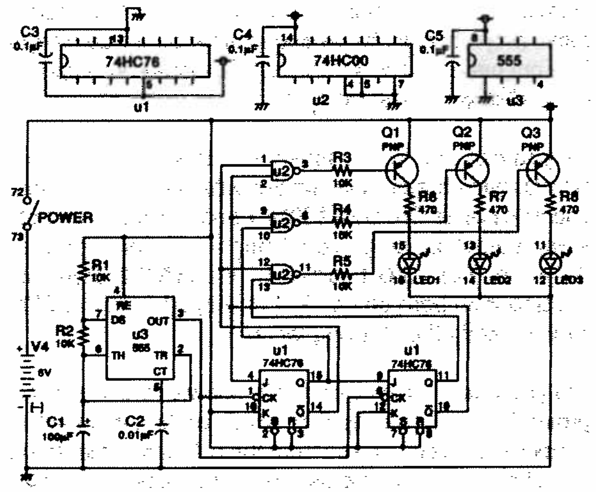
U1	74HC76	Q2	PNP	R3	10KΩ	R7	470Ω	C3	0.1μF
U2	74HC00	Q3	PNP	R4	10KΩ	R8	470Ω	C4	0.1μF
U3	555	R1	10KΩ	R5	10KΩ	C1	100μF	C5	0.1μF
Q1	PNP	R2	10KΩ	R6	470Ω	C2	0.01μF		

Here's a more refined counter circuit that counts up to three and uses three **LEDs** to indicate the count. It's a bit more difficult to count to three using three **LEDs** instead of two, so we have to add another circuit known as a line decoder to the counter circuit. You can locate the line decoder in the schematic - it's the group of NAND gates connected to the three **LEDs**.

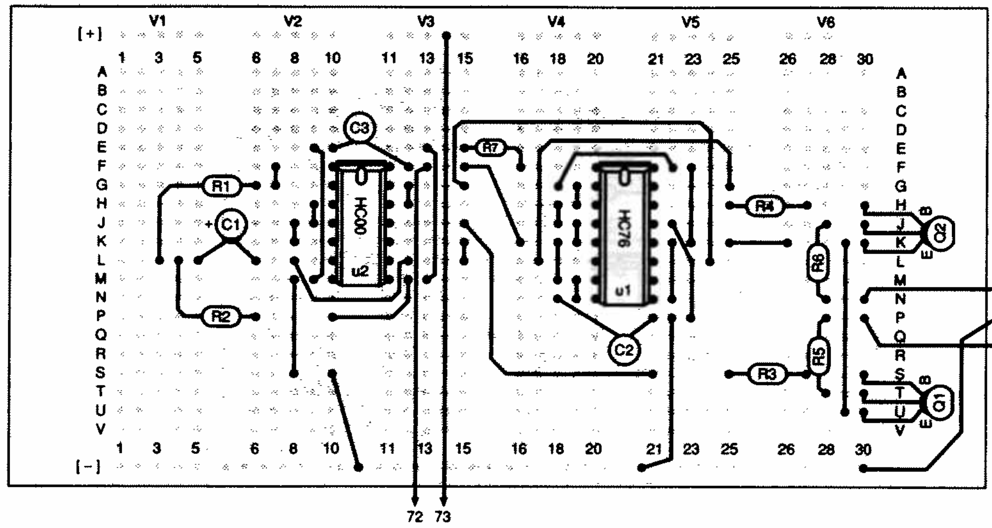
You probably recognized the oscillator U3 in the schematic. When you turn power ON, you'll see **LEDs 1, 2** and **3** light on and off in order. The counter circuit is ... (can you guess? Don't peek.)

It's a synchronous counter. Notice how each flip-flop has its own clock input signal from the oscillator??

Here's a brain exercise that'll really keep you going for a while ... can you trace the 0 and 1 inputs from the oscillator and follow them through the counter to the **LEDs**? It's not difficult to trace, and it gives you a good idea how a counter circuit like this works. Try it and see.



PROJECT 185. DIVIDE BY 4 COUNTER



U1	74HC76	Q1	PNP	R2	100KΩ	R5	470Ω	C1	3.3µF
U2	74HC00	Q2	PNP	R3	10KΩ	R6	470Ω	C2	0.1µF
		R1	22KΩ	R4	10KΩ	R7	10KΩ	C3	0.1µF

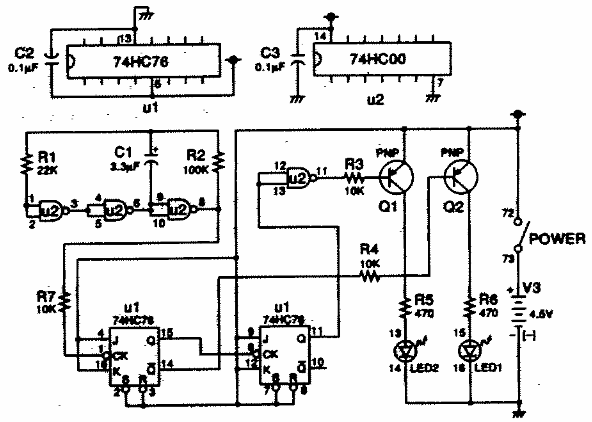
Here's a counter circuit that can count up to four. (Gee, these projects must be getting smarter as we go along!) Like our previous counters, it uses flip-flops - but a special kind. Can you guess which?

As you can see in the schematic, this project uses a NAND multivibrator to set up the pulses to count. When you turn power ON, both **LEDs** 1 and 2 blink on and off. After some blinking, you'll begin to see a pattern develops. First **LED** 1 comes on, then **LED** 2, then both on, and then both off - 1, 2, 1 and 2, off ... both LED's blink on and off in this sequence.

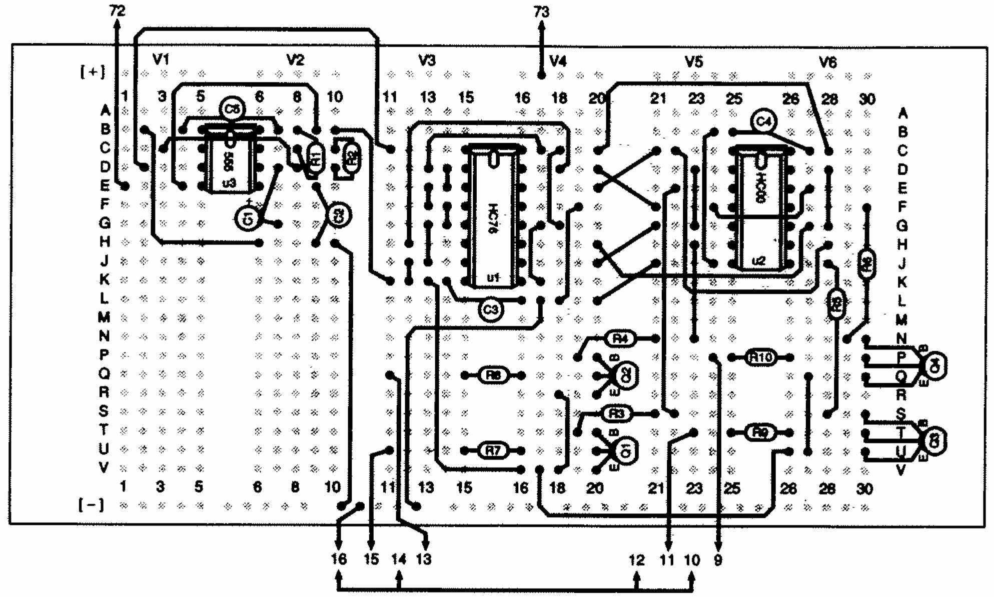
Did you guess which type of flip-flop we're using in this project? It's a toggle flip-flop, which we first met back in project 163. You can see from the schematic that the Q output of the first flip-flop serves as the clock input for the second flip-flop, meaning that this counter circuit is ...

Asynchronous, of course!

Back in project 165 we saw how "master and slave" flip-flop arrangements worked. This is the same arrangement used in this project. The first flip-flop controls the operation of the second. Try tracing the input of the multivibrator through the flip-flops to see how the **LEDs** are turned off and on!



PROJECT 186. DIVIDE BY 4 COUNTER WITH LINE DECODER



U1	74HC76	Q1	PNP	R1	10KΩ	R6	10KΩ	C1	100µF
U2	74HC00	Q2	PNP	R2	10KΩ	R7	470Ω	C2	0.01µF
U3	555	Q3	PNP	R3	10KΩ	R8	470Ω	C3	0.1µF
		Q4	PNP	R4	10KΩ	R9	470Ω	C4	0.1µF
				R5	10KΩ	R10	470Ω	C5	0.1µF

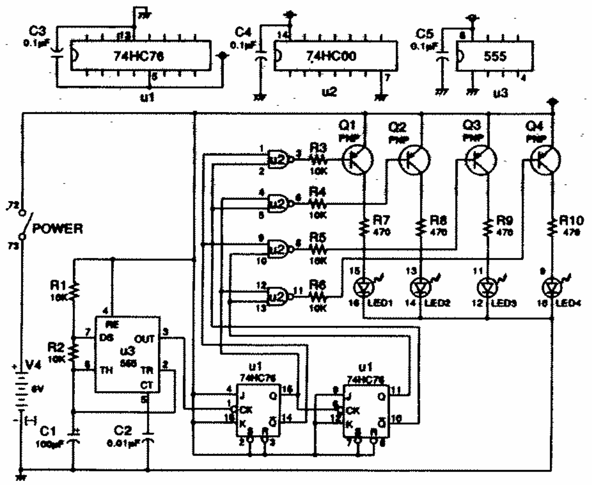
We can use the same line decoder arrangement that we used project 184 with a divide by 4 counter circuit. This project lets you see how it's done.

You'll notice that the oscillator U3 is used in this project. When you turn power ON, the four **LEDs** light up one after another as the circuit counts each pulse from the oscillator.

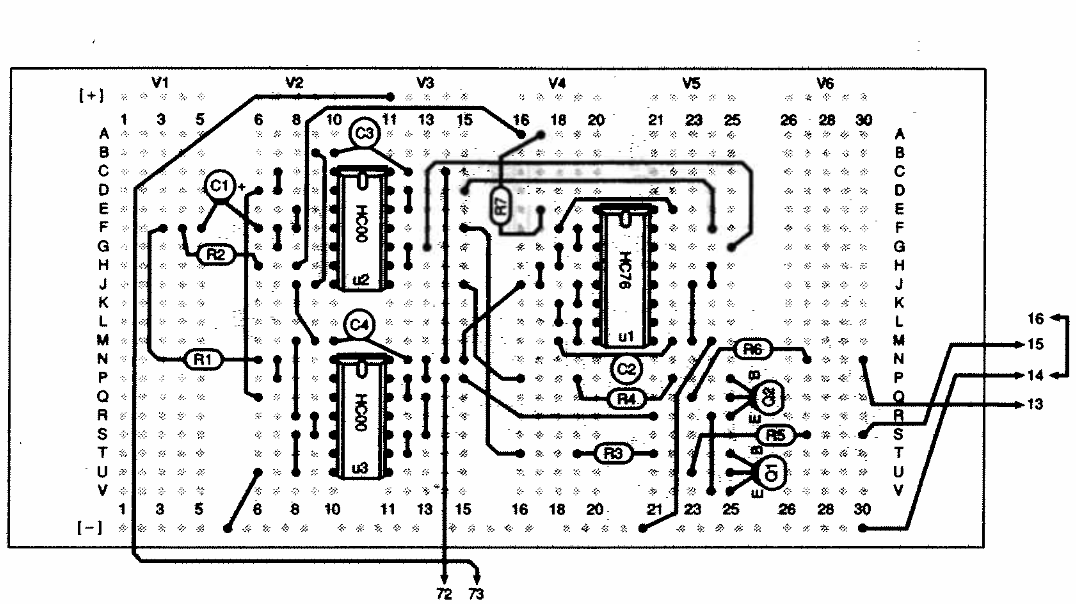
The counter is an asynchronous type and the Q and Q̄ outputs of both are input into the line decoder arrangement. You'll notice that each output serves as one of the inputs for two different NANDs. And the Q output of the first flip-flop is also the clock input for the second flip-flop.

Ever wonder what would happen if you added another flip-flop to this circuit? The first flip-flop "divides" the input into two outputs (Q and Q̄). The second flip-flop adds two more outputs. Do you think the third flip-flop would give you a total of six outputs? Guess again ... it would give you eight outputs! When flip-flops are connected in a "master and slave" arrangement, the outputs, aren't added together - they're multiplied. In a counter with two flip-flops, multiplying the outputs of both gives 2 x 2, or 4. With three outputs, the result of multiplying the outputs of all three is 2 x 2 x 2, or 8. If we were to connect four flip-flops together in a circuit like this, we would have 16 different outputs (you know why, don't you?).

And since the counter could have up to 8 or 16 different outputs, it could count up to 8 or 16. Today's large computers are able to count and handle large numbers using more complex versions of the circuit in the project. (Of course, the counters in computers are synchronous so the all the flip-flops can "march in step!")



PROJECT 187. HOW A LINE DECODER WORKS



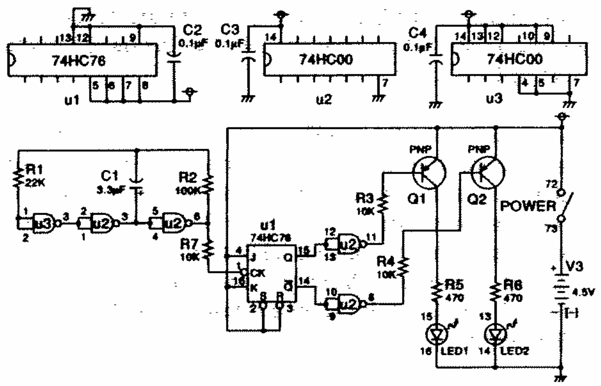
U1	74HC76	Q1	PNP	R1	22KΩ	R5	470Ω	C1	3.3μF
U2	74HC00	Q2	PNP	R2	100KΩ	R6	470Ω	C2	0.1μF
U3	74HC00			R3	10KΩ	R7	10KΩ	C3	0.1μF
				R4	10KΩ			C4	0.1μF

Line decoder can be a bit confusing when you're working with many flip-flops and their outputs. Let's back up a moment and take a look at a simpler counter and line decoder circuit.

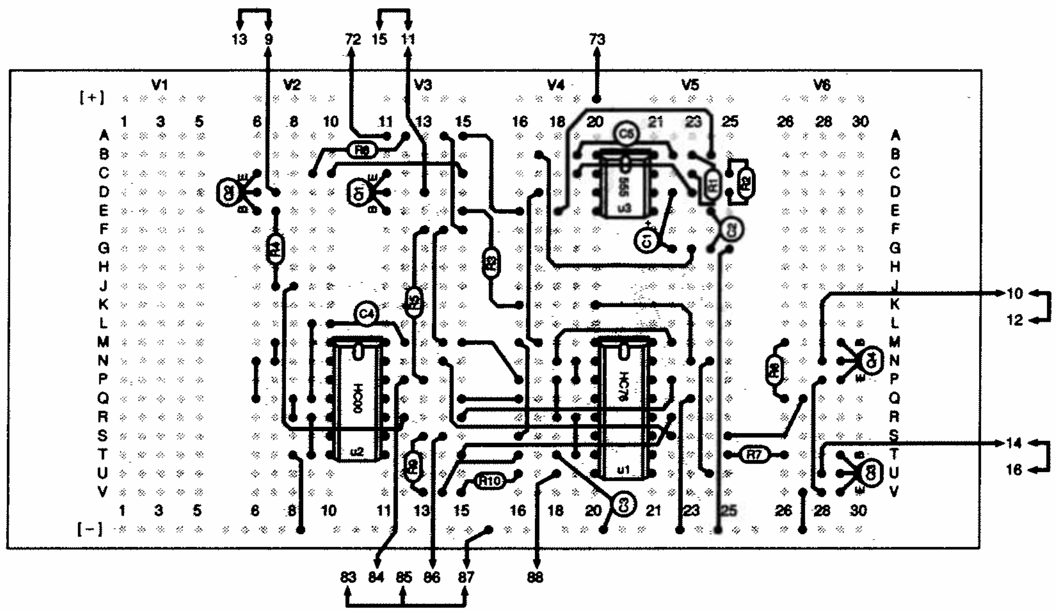
In this project we have a NAND multivibrator inputting a clock signal into a single flip-flop. You can easily guess what happens ... each time the clock signal is input, the flip-flop either sets or resets. Suppose that output Q is 1 while $\overline{Q}$ is 0. Both of these outputs go to NAND gates, where they're used for both inputs. Turn power ON. If Q is 1, the output of the NAND is 0 - and **LED 1** lights. $\overline{Q}$ is 0 at the same time, so the NAND it is connected to have 1 for an output ... and **LED 2** goes out. The next clock pulse from the multivibrator causes this situation to reverse (or flip-flop!).

If you'll look back at the schematics for the counters with line decoder we've played with in previous projects, you'll notice that each NAND gets one input from the first flip-flop and another input from the second flip-flop. These combined inputs turn the **LEDs** on or off as the two flip-flops set and reset.

We use NANDs exclusively in counter projects. But do you think that we could make counter circuits out of AND, OR, NOR or XOR gates? You might want to get some scratch paper and try to figure out how those digital circuits could combine the outputs of the flip-flop in this project. Do you think NAND gates are the easiest way to make a line decoder? Can you come up with a simpler method?



PROJECT 188. MULTIPLE COUNTER



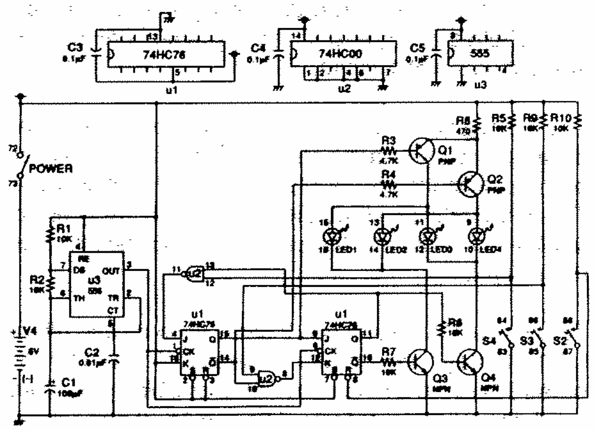
U1	74HC76	Q1	PNP	R1	10KΩ	R6	10KΩ	C1	100μF
U2	74HC00	Q2	PNP	R2	10KΩ	R7	10KΩ	C2	0.01μF
U3	555	Q3	NPN	R3	4.7KΩ	R8	470Ω	C3	0.1μF
		Q4	NPN	R4	4.7KΩ	R9	10KΩ	C4	0.1μF
				R5	10KΩ	R10	10KΩ	C5	0.1μF

It's not very handy having a separate counter for counting by 2, 3, 4 or some other number. You might be wondering if there's some way to have a counter circuit capable of counting up to several different numbers. Yes, there is such a circuit and we're going to build it right now.

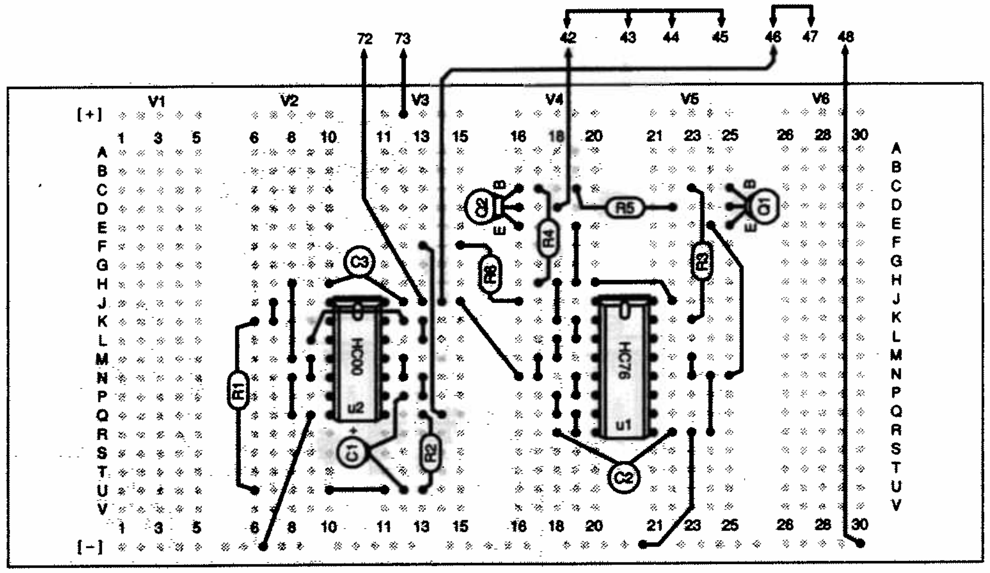
You can see from the schematic that the pulses to count are generated by the oscillator U3. The counter itself is synchronous, since each flip-flop receives a clock signal directly from the oscillator. Turn power ON and press **S2**. You'll see that **LED 1** and **LED 2** take turns lighting, indicating that this project is functioning as a "divide by 2" (or binary) counter.

Next, release **S2** and then press **S3**. Now you can see that **LED 1, LED 2** and **LED 3** light up one after the other. This indicates that the circuit is now a divide by 3 counter. Finally, release **S3** and press **S4** so that you'll see all four (4) **LEDs** light in turn. This indicates the project is now a divide by 4 counter.

This project shows how counters can be made to cover a wide range of counting rates. More complex counters using this same basic circuit are very important pieces of electronic equipment.



PROJECT 189. BINARY COUNTER WITH DISPLAY



U1	74HC76	Q1	NPN	R1	22KΩ	R4	10KΩ	C1	3.3μF
U2	74HC00	Q2	PNP	R2	100KΩ	R5	1KΩ	C2	0.1μF
				R3	10KΩ	R6	10KΩ	C3	0.1μF

Counters aren't terribly exciting circuits when they just flash **LEDs** ... but add the digital display and it's a different story altogether. Here's a binary counter that lets you see it in action as it counts.

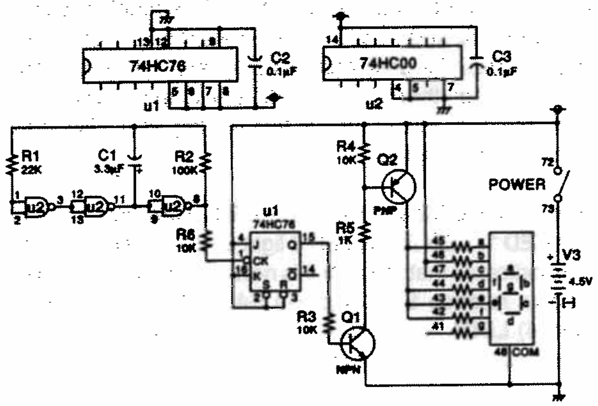
You'll recognize this circuit as being very similar to the binary counter you built back in project 181. The source of the pulses for the counter is a NAND multivibrator. The output of the multivibrator provides clock-signals for the flip-flop. Note that the output of the flip-flop is taken from output Q.

Turn power ON. You'll see 0 and 1 flash on and off the **LED display**. Why 0 and 1 instead of 1 and 2? We called this circuit a binary counter ... and binary numbers are those made up entirely of 0s and 1s. 0 and 1 mean the same thing in the binary number system that they do in our normal (called decimal) number system.

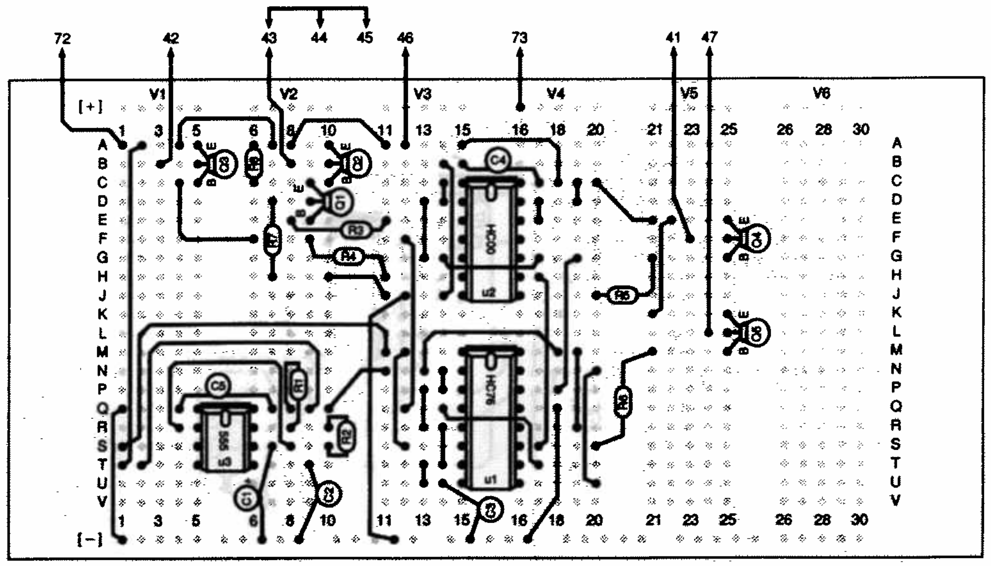
But what happens when we want to express the number 2 in the binary system? We can't write "2" since we can only use 0 and 1 in the binary system. We do in this case what we do when we reach 9 in our decimal system ... we write " 10?"

That's right, 10 in binary numbers is actually 2 in the decimal system. How do you suppose we write 3 in binary numbers? That's right ... it's " 11."

Ask your math teacher for more information about the binary number system. Your library will also have some math books covering the subject. Computers add, subtract, multiply and divide numbers in binary form.



PROJECT 190. DIVIDE BY 3 COUNTER WITH DISPLAY

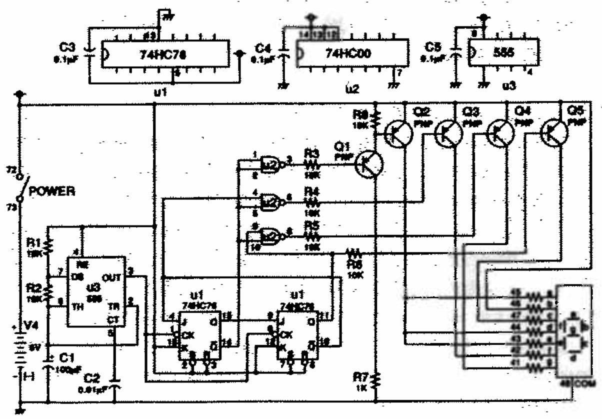


U1	74HC76	Q1	PNP	R1	10KΩ	R6	10KΩ	C1	100μF
U2	74HC00	Q2	PNP	R2	10KΩ	R7	1KΩ	C2	0.01μF
U3	555	Q3	PNP	R3	10KΩ	R8	10KΩ	C3	0.1μF
		Q4	PNP	R4	10KΩ			C4	0.1μF
		Q5	PNP	R5	10KΩ			C5	0.1μF

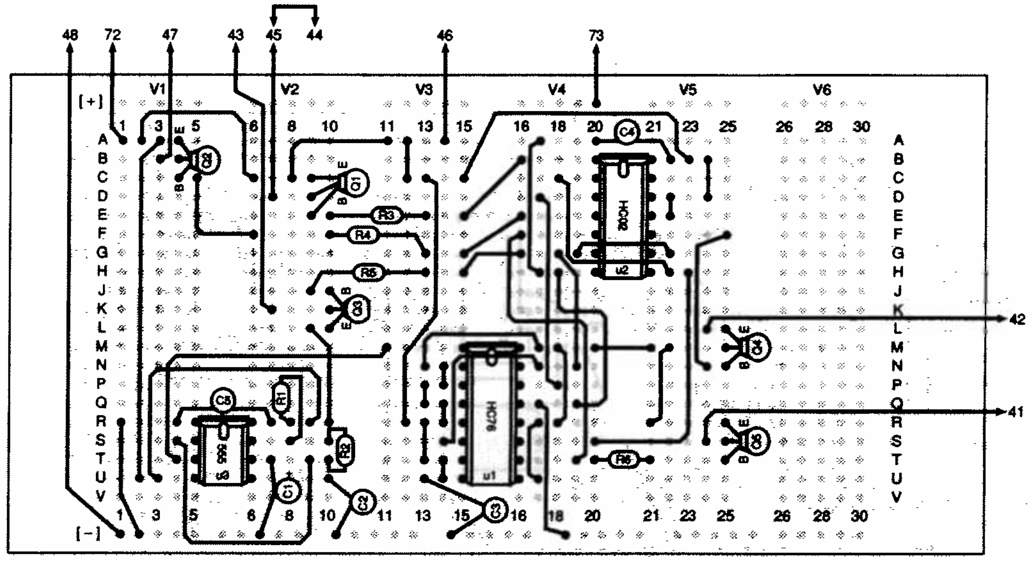
In project 184, you built a divide by 3 counter with a line decoder. After our last project, you might be wondering if we can add the display to other counter circuits we've built. The answer is yes - as this project demonstrates.

This is basically the same circuit you built back in project 184, but with some alternations to the line decoder to allow use of the **LED display**. Compare the schematics for these and find out how and why the line decoder was changed.

This circuit uses the oscillator U3 to supply pulses for the synchronous counter. The line decoder takes the outputs from the counter and combines them using the NAND gates to turn the transistors ON and OFF. These transistors turn the various **LED** segments of the **LED display** on and off. When you turn power on, you can see 0, 1 and 2 displayed in a repeating pattern on the **LED display**.



PROJECT 191. DIVIDE BY 4 COUNTER WITH DISPLAY



U1	74HC76	Q1	PNP	R1	10KΩ	R6	10KΩ	C1	100µF
U2	74HC02	Q2	PNP	R2	10KΩ			C2	0.01µF
U3	555	Q3	PNP	R3	10KΩ			C3	0.1µF
		Q4	NPN	R4	10KΩ			C4	0.1µF
		Q5	PNP	R5	10KΩ			C5	0.1µF

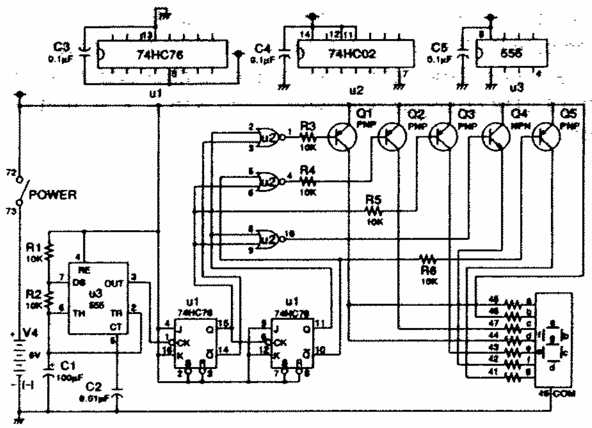
Here's the divide by 4 counter you built back in project 186 however we're using NOR gate IC against NAND gate IC. By using NOR gate IC, you'll need less electronic parts rather than NAND gate IC. As in the previous project, most changes are in the line decoder to allow use of the **LED display**.

As in other counters you've played around with, the pulses for the counter are generated by the oscillator U3. The counter circuit is an asynchronous type, since the Q output of the first flip-flop provides the clock signal for the second flip-flop. When you turn power ON, you'll see 0, 1, 2 and 3 light up on the **LED display** as the counter keeps track of the pulses from the oscillator.

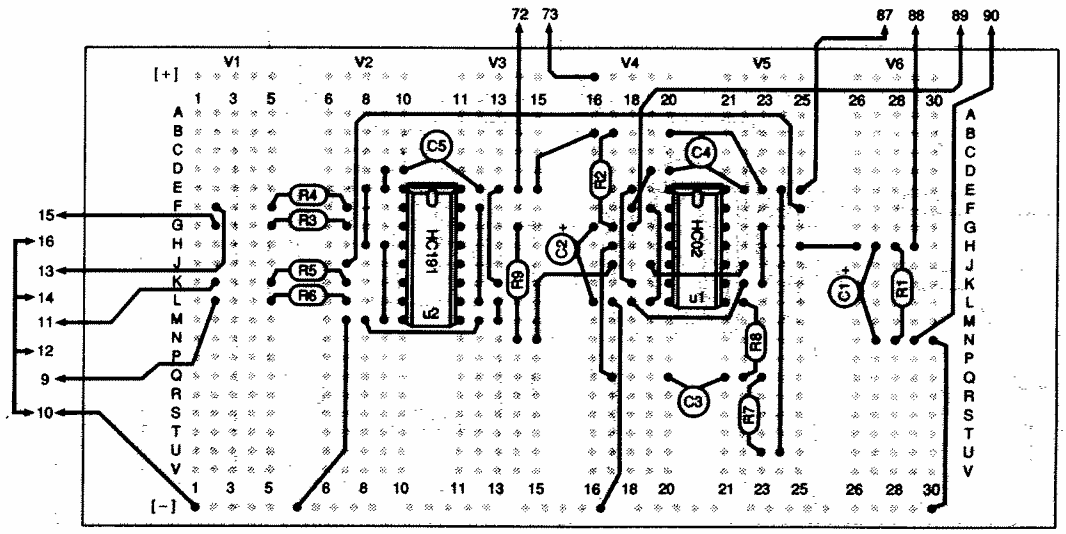
We've used oscillator with these counter circuits simply because they're a handy source of pulses. You could just as easily do away with the oscillator part of this circuit altogether and uses the key to generate the pulses to be counted! Try it yourself - let your hand do the work instead of the oscillator. (After all, we've been using that circuit so much it deserves vacation!)

You can also make connections to the **LED display** so that the counter indicates other numbers, small letters or capital letters as it counts the pulses. Can you figure out how to do this? Check back at those notes you made for projects 21, 22 and 23 - we told you those notes would come in handy for you.

When you first turn on the power the counter might not start with 0 and then 1, 2 and 3. Don't let that bother you. Most counters that you'll come across in electronic circuitry are "reset" as a result always starts at 0. (By the way, the "R" terminal on the 74HC76 IC is for "Reset" or "Clear" which is used to reset the count to zero or clear the count.)

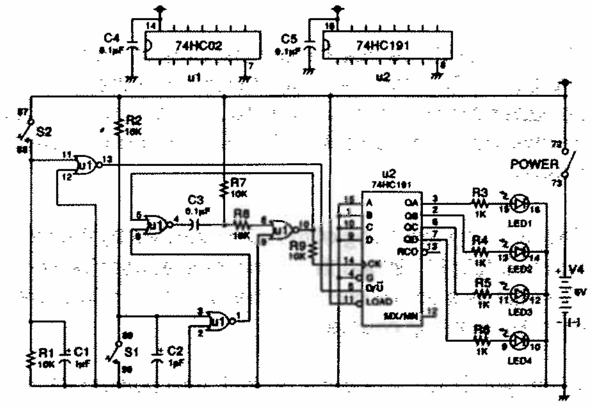


PROJECT 192. UP/DOWN COUNTER



U1	74HC02	R3	1KΩ	R7	10KΩ	C2	1µF
U2	74HC191	R4	1KΩ	R8	10KΩ	C3	0.1µF
R1	10KΩ	R5	1KΩ	R9	10KΩ	C4	0.1µF
R2	10KΩ	R6	1KΩ	C1	1µF	C5	0.1µF

Here's an up/down counter that uses a counter IC with an up/down function. Output of this counter is displayed by four **LEDs**.



S2 is used for up/down switching. This counter works as an down counter when **S2** is OFF, and as a up counter when **S2** is ON. **S1** is used to send out the pulses to count.

Wire the project and turn power ON. **S2** is OFF at this time, so the counter works as a down counter, and starts counting from 15 to 14, 13, 12 and so on as you send the pulses by pressing **S1**.

What do you think happens if you press **S1** to send out the pulses while pressing **S2**? You certainly can figure that out. Repeat experimenting with this counter and get the knack of it.

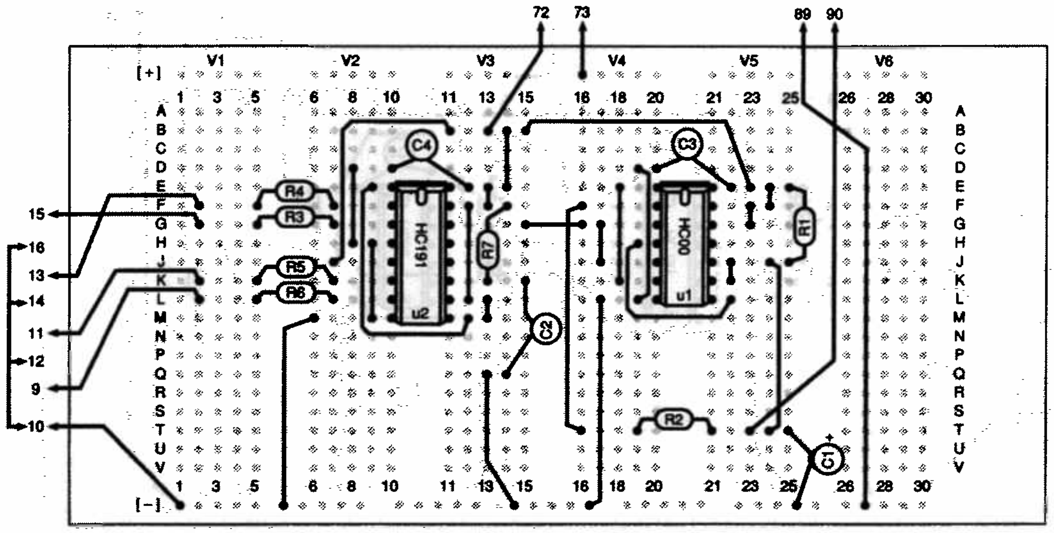
Figure 1 shows the decimal number for each of the counter outputs displayed by different ON-OFF combinations of **LED 1 - LED 4**.

Decimal	Binary				S2	
	Q3	Q2	Q1	Q0	ON UP	OFF DOWN
0	0	0	0	0		
1	0	0	0	1		
2	0	0	1	0		
3	0	0	1	1		
4	0	1	0	0		
5	0	1	0	1		
6	0	1	1	0		
7	0	1	1	1		
8	1	0	0	0		
9	1	0	0	1		
10	1	0	1	0		
11	1	0	1	1		
12	1	1	0	0		
13	1	1	0	1		
14	1	1	1	0		
15	1	1	1	1		

Decimal number
"1" when LED is ON
"0" when LED is OFF
QA: LED1
QB: LED2
QC: LED3
QD: LED4

Figure 1

PROJECT 193. DOWN COUNTER

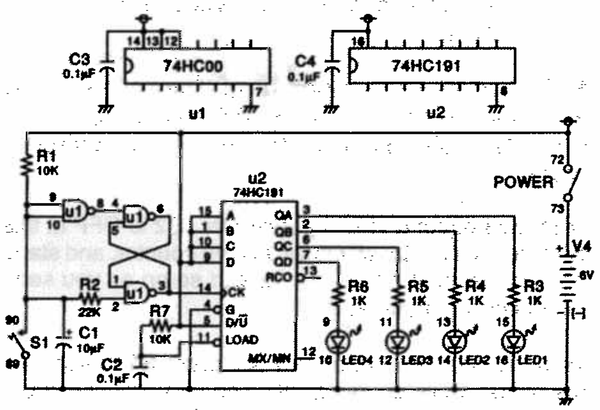


U1	74HC00	R1	10KΩ	R4	1KΩ	R7	10KΩ	C3	0.1µF
U2	74HC191	R2	22KΩ	R5	1KΩ	C1	10µF	C4	0.1µF
		R3	1KΩ	R6	1KΩ	C2	0.1µF		

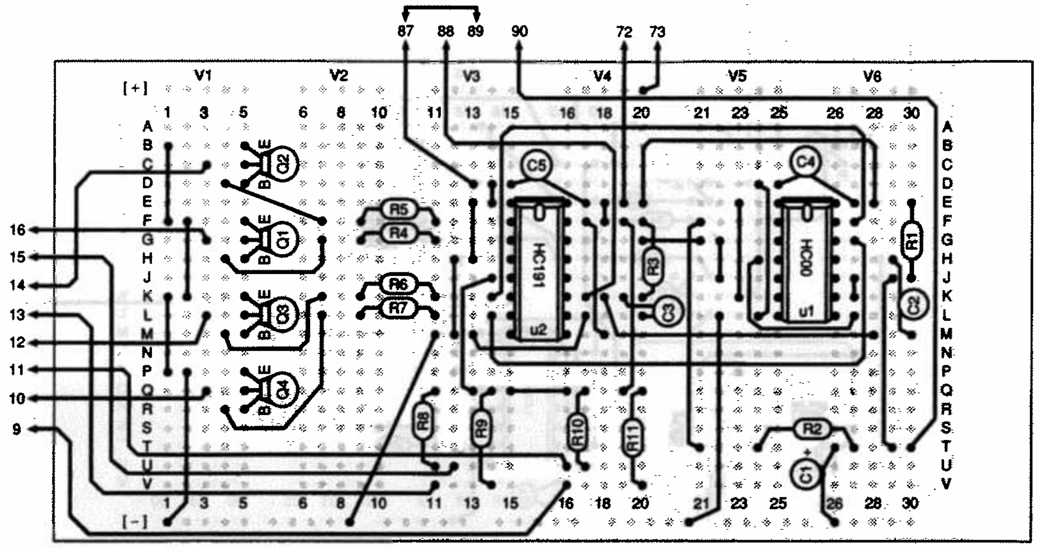
The counter IC we used in our last project is called an "Up/Down Counter" because it can count the number upward from 0 to 1, 2, 3..and so on (up counter) and downward from 15 to 14,13, 12... and so on (down counter).

The up/down switching of the IC is handled by the D/$\overline{U}$ terminal (pin 5). When this terminal is at positive level, IC works as a down counter. When it is at negative level, it works as an up counter.

When you finish wiring the project, turn power ON and operate the counter while looking at Figure 1 in previous project. What number does it display first? It's 15. Now press **S1** to input the pulses and watch the **LEDs**, and observe how the counter works.



PROJECT 194. DECADE DOWN COUNTER

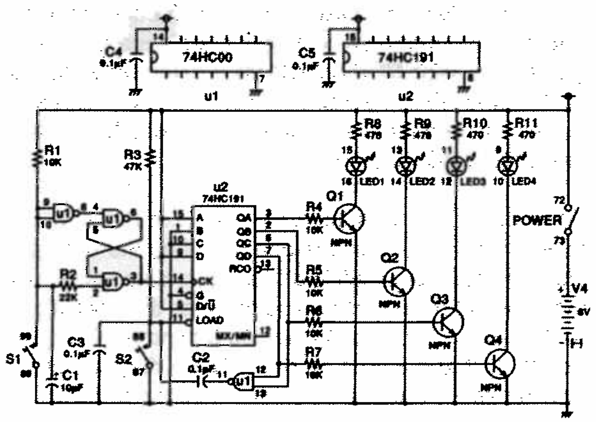


U1	74HC00	Q3	NPN	R4	10KΩ	R9	470Ω	C1	10µF
U2	74HC191	Q4	NPN	R5	10KΩ	R10	470Ω	C2	0.1µF
Q1	NPN	R1	10KΩ	R6	10KΩ	R11	470Ω	C3	0.1µF
Q2	NPN	R2	22KΩ	R7	10KΩ			C4	0.1µF
		R3	47KΩ	R8	470Ω			C5	0.1µF

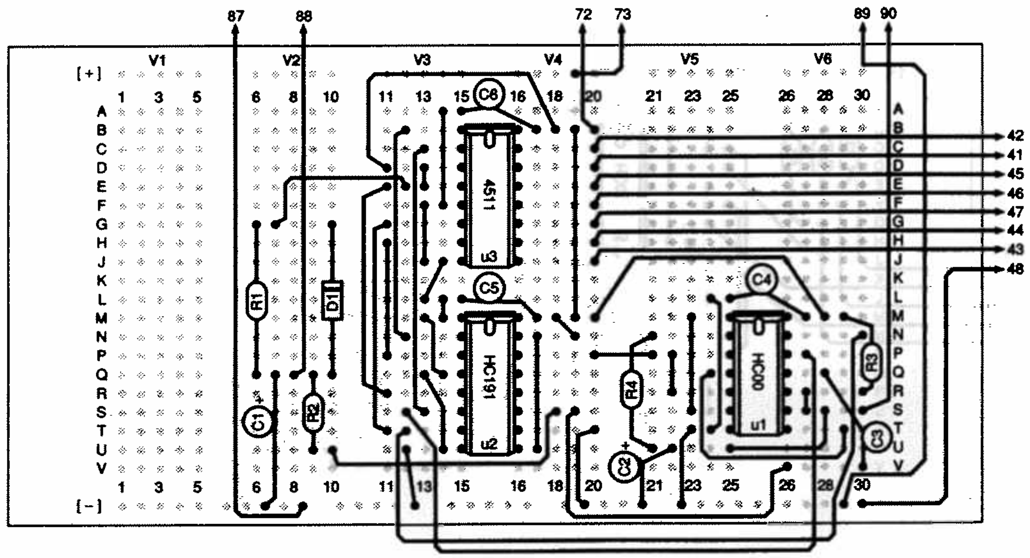
We're now going to make a decade counter using a counter IC. It can display the numbers 0 - 9 using four **LEDs**.

As you can see by the schematic, **S1** sends the pulses to CK terminal of the IC. The output of the counter is taken from QA - QD. **S2** is used as a reset switch for resetting the count to 9.

When you finish assembling, turn power ON and press **S2**. What happens to the **LEDs**? They all go out and indicate 9. Now flip **S1**, and you'll see **LED 4** lights up to indicate 8. (Note this is a down counter.) Now press **S1** again a few times and see how the number displayed by the **LEDs** changes downward from 9 to 8, 7, 6..and so on.



PROJECT 195. DECADE DOWN COUNTER WITH DISPLAY



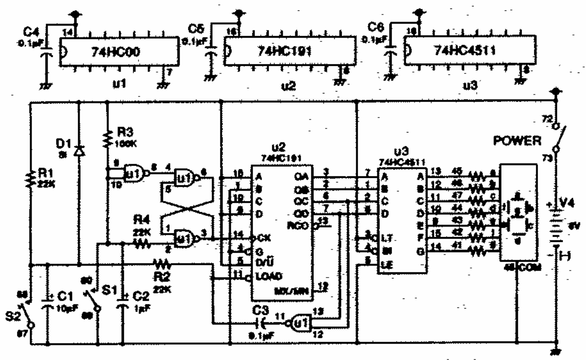
U1	74HC00	R1	22KΩ	R4	22KΩ	C3	0.1μF	C6	0.1μF
U2	74HC191	R2	22KΩ	C1	10μF	C4	0.1μF	D1	Si
U3	74HC4511	R3	100KΩ	C2	1μF	C5	0.1μF		

In this project, we're going to make a down counter that can display the numbers **0 - 9** on the **LED display**, using a counter IC and a decoder IC. As we found in our project 194, a down counter can count the numbers downward from **9** to **8**, **7**, **6**..and so on when the pulses are input.

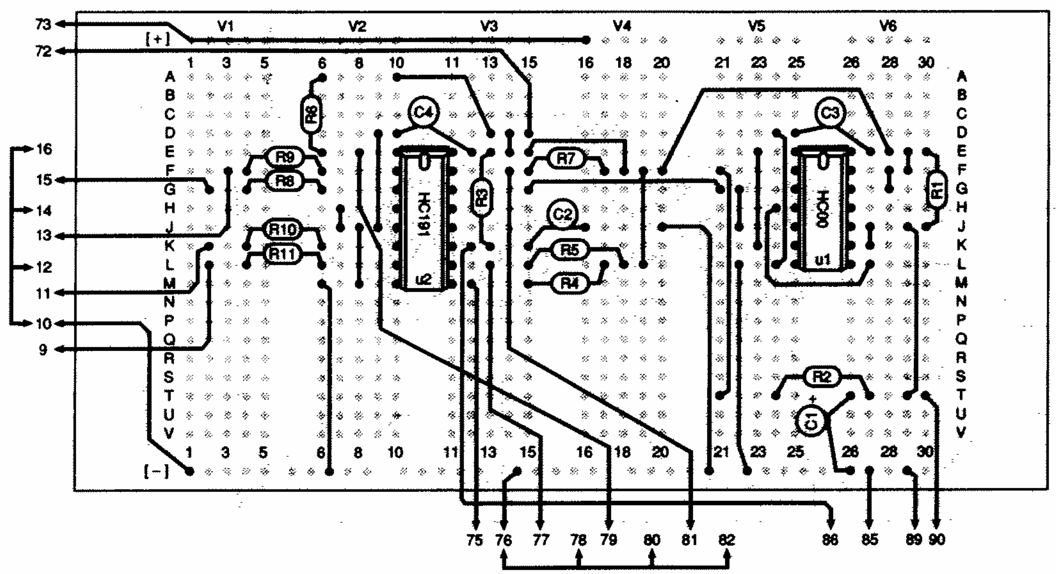
The schematic lets you see that the counter IC output is taken from QA - QD. This output is decoded by the decoder IC and displayed on the **LED display**.

When you finish the wiring, turn power ON and you'll notice that **9** is displayed on the **LED display**. Tap **S1**, and **9** changes to **8** ..Tap **S1** a few times, and the number changes to **7**, **6**, **5**..and so on.

S2 is a reset switch for resetting the count to **9**. Press it and see if the number on the display changes to **9**.



PROJECT 196. PRESETTABLE COUNTER



U1	74HC00	R3	100KΩ	R6	4.7KΩ	R9	1KΩ	C1	10μF
U2	74HC191	R4	4.7KΩ	R7	4.7KΩ	R10	1KΩ	C2	0.1μF
R1	10KΩ	R5	4.7KΩ	R8	1KΩ	R11	1KΩ	C3	0.1μF
R2	22KΩ							C4	0.1μF

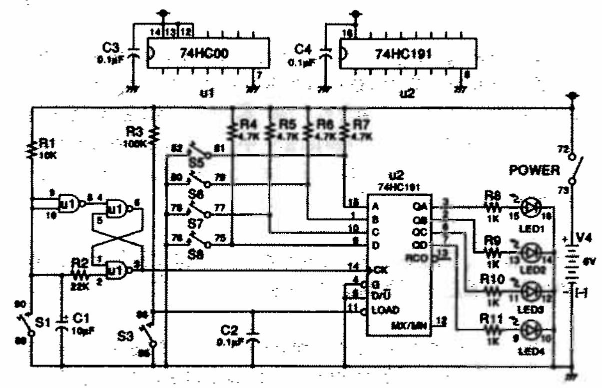
We're going to make a counter that can start counting from any preset number, using a counter IC with a preset function. The output of this counter is displayed by four **LEDs**. Figure 1 shows the decimal number for each of the outputs displayed in binary numbers.

Wire the project and turn power ON. At this time, the counter is preset to 15 and all **LEDs** light up; when you press **S1** to send out the pulses, it starts counting from 0 to 1, 2, 3 and so on. Let it go on counting. (When it is set to 0, all **LEDs** OFF).

Now try presetting the counter. You use **S5 - S8** to input the presetting the number. Use binary -- see table and press the key(s) that correspond to 0 in the table. For example, to input 6 as the preset number, press and hold **S5** and **S8** and then press **S3**. The **LED** display changes **6**, as shown in example 3.

With the number **6** is preset this way, press **S1** to send out the pulses. The counter starts counting from **7** to **8**, **9**, **10** and so on. Preset the counter to other numbers and see how it works.

You probably noticed something different in this project. We have been using the four **LEDs** to count up to **10**, but in this project we counted up to **15**. We'll explain about this in next project.



Decimal	Binary(4 Bits) P3 QD	P2 QC	P1 QB	P0 QA	Decimal	Binary(4 Bits) P3 QD	P2 QC	P1 QB	P0 QA
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

LED ON: 1
LED OFF: 0
QA : LED5 QB : LED6 QC : LED7 QD : LED8

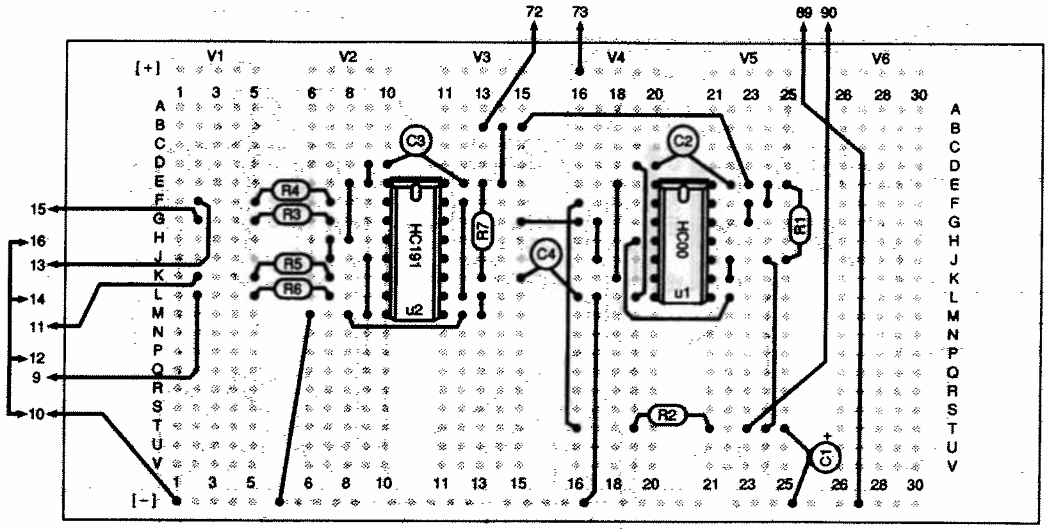
Figure 1

	LED4	LED3	LED2	LED1	
EX1	●	●	●	●	Displays 0.
EX2	S8 (ON)	S7 (OFF)	S6 (OFF)	S5 (ON)	Setting 6 (ON/OFF)
EX3	●	○	○	●	Displays 6.

LED ON: ○
LED OFF: ●
For setting EX2, press S3 with pushing S5 and S8.

Figure 2

PROJECT 197. HEXADECIMAL COUNTER

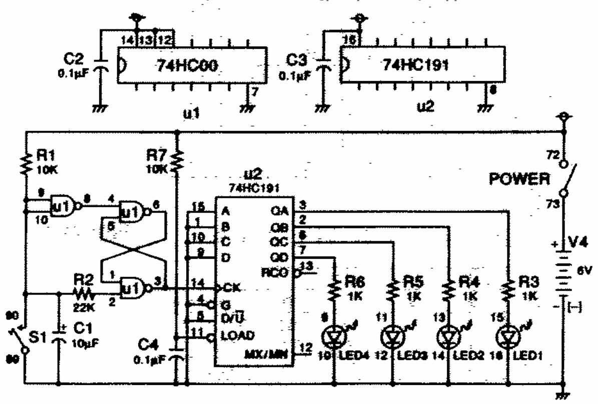


U1	74HC00	R1	10KΩ	R4	1KΩ	C1	10μF
U2	74HC191	R2	22KΩ	R5	1KΩ	C2	0.1μF
		R3	1KΩ	R6	1KΩ	C3	0.1μF
				R7	10KΩ	C4	0.1μF

We're now going to find out how to express the hexadecimal numbers **0 - 15** in four-digit binary numbers, using a counter IC. The counter IC outputs hexadecimal numbers when clock pulses are input. The relationship between the hexadecimal numbers and the binary numbers is shown in Figure 1.

Before using this project, remember that 1 is expressed when the **LED** is ON and 0 when it is OFF.

Wire the project and turn power ON, and see what the **LEDs** are doing. Do they display 0 ? Now, give a flip to **S1** and see what number is displayed. It's 1. Press **S1** while watching Figure 1, and see how this counter works.



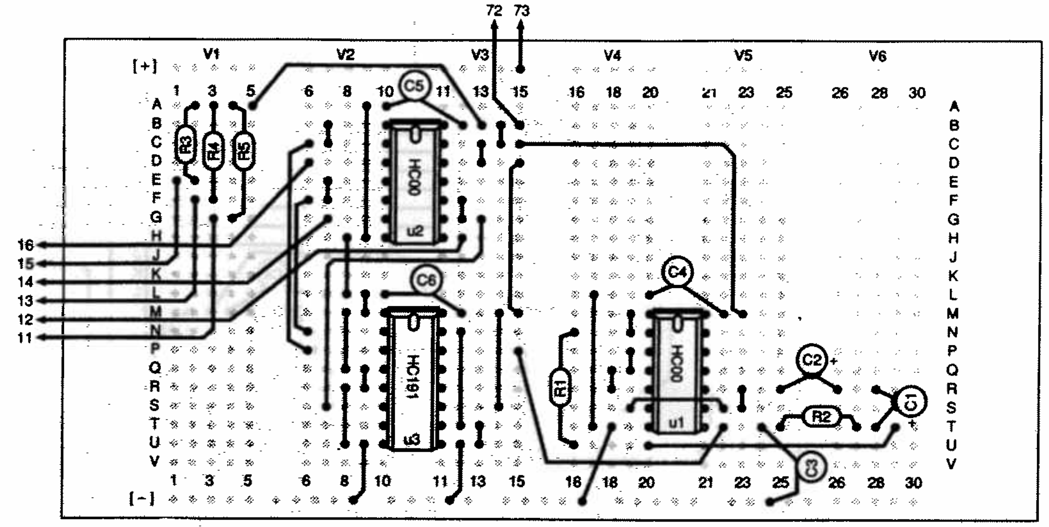
LED4	LED3	LED2	LED1	Hex.
●	●	●	●	0
●	●	●	○	1
●	●	○	●	2
●	●	○	○	3
●	○	●	●	4
●	○	●	○	5
●	○	○	●	6
●	○	○	○	7
○	●	●	●	8
○	●	●	○	9
○	●	○	●	A
○	●	○	○	B
○	○	●	●	C
○	○	●	○	D
○	○	○	●	E
○	○	○	○	F

↑ 2^3 ↑ 2^2 ↑ 2^1 ↑ 2^0

Returns to "0" state.

LED OFF: ● (0)
LED ON : ○ (1)

PROJECT 198. OCTAL COUNTER

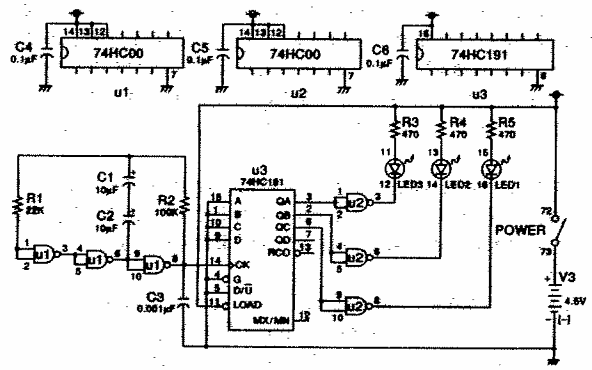


U1	74HC00	R1	22KΩ	R4	470Ω	C1	10μF	C4	0.1μF
U2	74HC00	R2	100KΩ	R5	470Ω	C2	10μF	C5	0.1μF
U3	74HC191	R3	470Ω			C3	0.001μF	C6	0.1μF

Here's an octal counter that can display the numbers **0 - 7** in three **LEDs**, using a counter IC. Figure 1 shows the relationship between the decimal numbers **0 - 7** and the counter outputs displayed by different ON-OFF combinations of **LED 1 - LED 3.**

The schematic shows you see that the astable multivibrator using a NAND gate generates clock pulses. U3 counts the pulses from the multivibrator, and its output is sent to QA - QC. A NAND gate circuit is used to display this output with the **LEDs**.

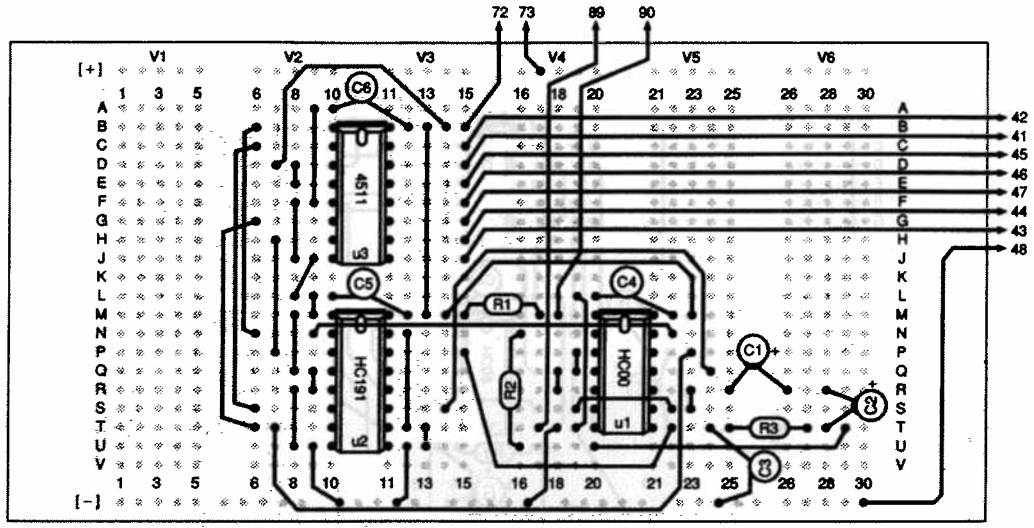
Wire the project, turn power ON and see what happens to the **LEDs**. You'll see them start blinking ON and OFF. Watch the **LEDs** while looking at Figure 1, and you'll notice that the counter is counting from **0** to **1, 2, 3...** and so on.



Octal	LED3	LED2	LED1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

LED ON : 1 or H
LED OFF: 0 or L

PROJECT 199. RANDOM ACCESS DISPLAY

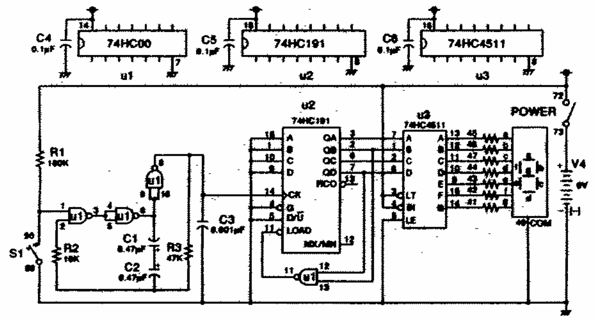


U1	74HC00	R1	100KΩ	C1	0.47μF	C4	0.1μF
U2	74HC191	R2	10KΩ	C2	0.47μF	C5	0.1μF
U3	74HC4511	R3	47KΩ	C3	0.001μF	C6	0.1μF

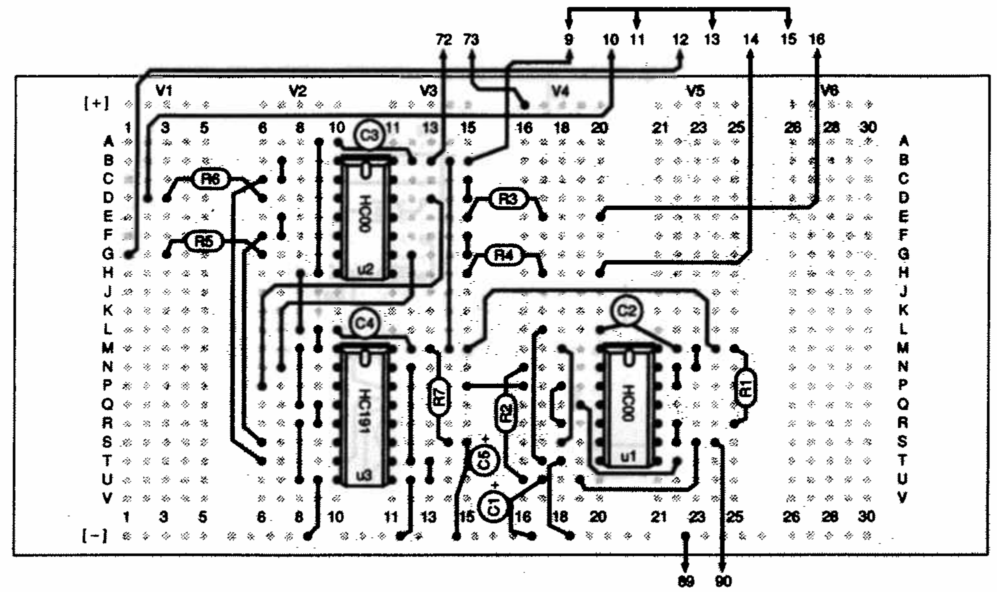
Here's a random access display. It indicates the numbers **0 - 9** rapidly in order, and stops at one of the numbers when **S1** is pressed. You can use it as for roulette or dice.

When you finish wiring, turn power ON. The numbers on the **LED display** changes rapidly and looks like "8." Now press **S1** and see what happens. The display stops at one of the numbers **0 - 9**. Now release **S1** and the number begins changing rapidly again. Press and release **S1** a few times, and you'll notice that the number on the **LED display** changes each time you release **S1**.

As you can see by the schematic, the astable multivibrator sends out clock pulses to the counter IC. It stops generating the clock pulses when **S1** is pressed. The counter IC is a decade-down counter, and its output is decoded by the decoder IC and displayed on the **LED display**.



PROJECT 200. DECADE COUNTER

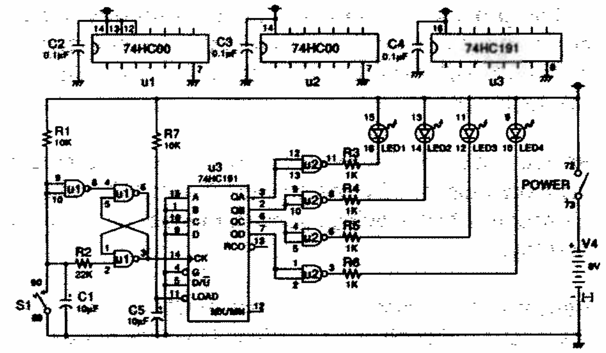


U1	74HC00	R1	10KΩ	R4	1KΩ	R7	10KΩ	C3	0.1μF
U2	74HC00	R2	22KΩ	R5	1KΩ	C1	10μF	C4	0.1μF
U3	74HC191	R3	1KΩ	R6	1KΩ	C2	0.1μF	C5	10μF

Here's another decade counter using a counter IC, but this one is an up counter unlike the one we built in our last project. It can display the numbers **0 - 9** by four **LEDs**. Figure 1 shows the decimal number for each of the counter outputs displayed by different ON-OFF combinations of the four **LEDs**.

The schematic shows you that IC 74HC191 is the decade up counter, and **S1** sends the pulses to the CK terminal. The output of the counter is taken from QA - QD. A NAND gate circuit is used to display this output using the **LEDs**.

When you finish assembling, turn power ON and press **S1**. The counter starts counting the numbers upward from **0** to **1, 2, 3, 4** ... and so on. Use this project while looking at Figure 1 so you can get a better understanding of the counter operation.

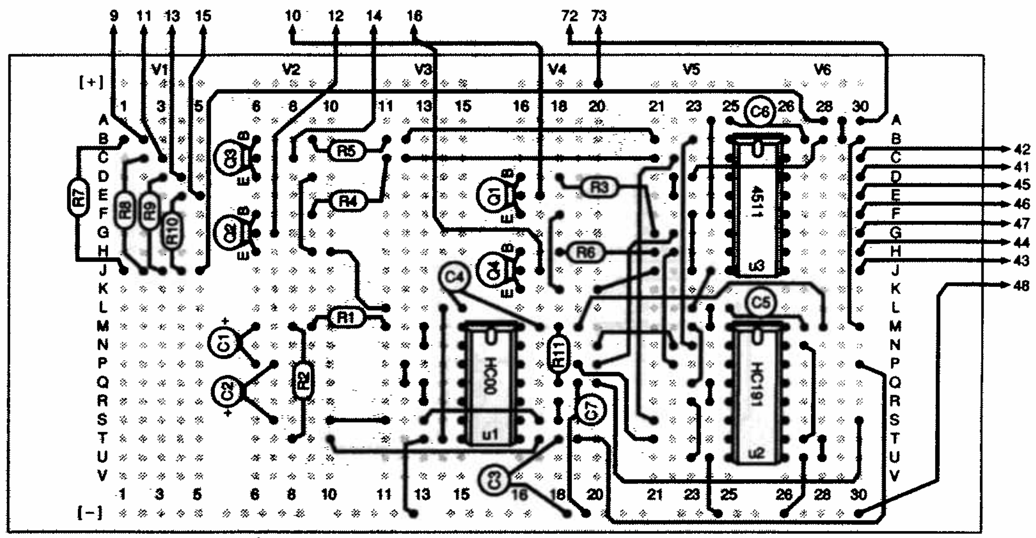


Decimal	LED4	LED3	LED2	LED1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

LED ON : 1 or H
LED OFF: 0 or L

Figure 1

PROJECT 201. BCD COUNTER WITH DISPLAY

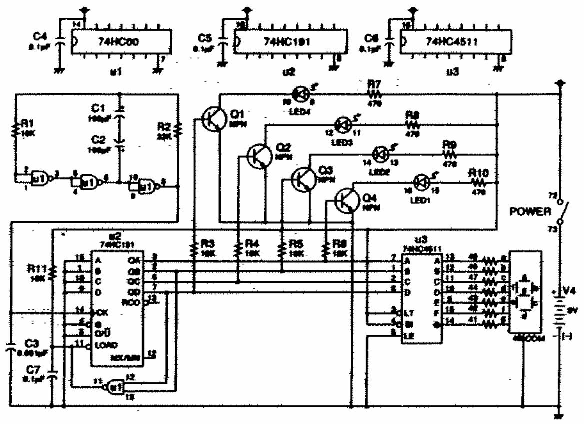


U1	74HC00	Q3	NPN	R4	10KΩ	R9	470Ω	C3	0.001µF
U2	74HC191	Q4	NPN	R5	10KΩ	R10	470Ω	C4	0.1µF
U3	74HC4511	R1	10KΩ	R6	10KΩ	R11	10KΩ	C5	0.1µF
Q1	NPN	R2	33KΩ	R7	470Ω	C1	100µF	C6	0.1µF
Q2	NPN	R3	10KΩ	R8	470Ω	C2	100µF	C7	0.1µF

In this project, we compare the BCD (binary coded decimal) display on the **LEDs** with the decimal display on the **LED display**. A slow clock pulse is counted by the counter IC.

As you can see in the schematic, U1 is used to generate the clock pulses, U2 is the counter IC, and U3 works as a decoder which displays decimal number on the **LED display**.

Assemble the project and turn power ON. At this time, the **LED display** shows "0". When the clock pulses are generated, it starts counting from **1** to **2, 3, 4....** an so on, as shown in **Figure 1**, and at the same time **LED 1 - LED 4** begin blinking ON and OFF. Can you see the relationship between the four **LEDs** and the **LED display**? Look at Figure 1. It helps you understand the relationship.

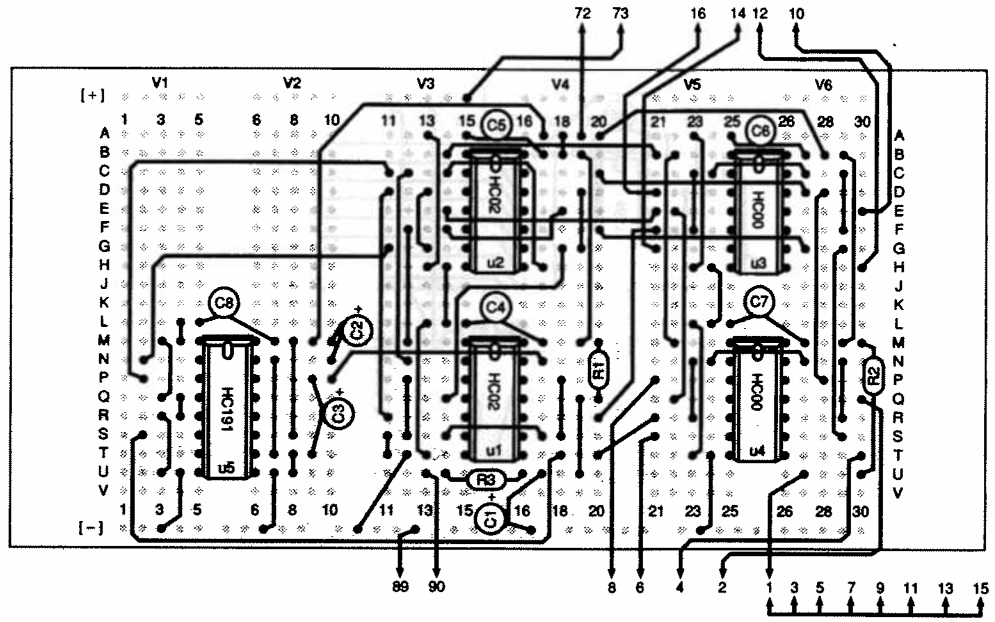


Decimal	BCD Code				7-Segment Element						
	LED4	LED3	LED2	LED1	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

LED ON : 1 or H
LED OFF: 0 or L

Figure 1

PROJECT 202. OCTAL COUNTER WITH LINE DECODER

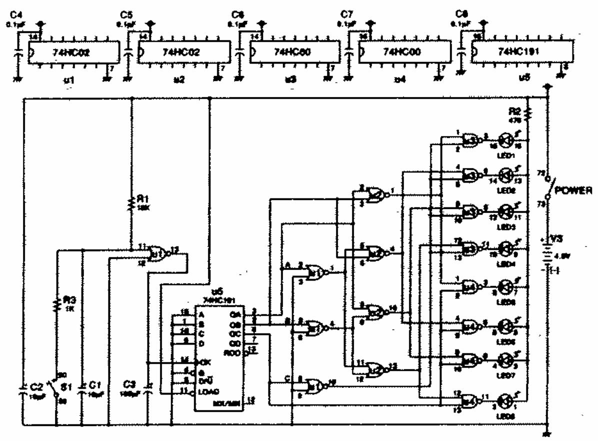


U1	74HC02	U5	74HC191	C1	10µF	C5	0.1µF
U2	74HC02	R1	10KΩ	C2	10µF	C6	0.1µF
U3	74HC00	R2	470Ω	C3	100µF	C7	0.1µF
U4	74HC00	R3	1KΩ	C4	0.1µF	C8	0.1µF

In this project, we're going to light up one of the eight **LEDs**, using an octal counter and an octal line decoder.

Look at the schematic for the project. You'll see that the counter's output is taken from QA - QC. This output is decoded by the octal line decoder made up of a NAND gate and a NOR gate to light up the eight **LEDs**. Figure 1 shows the relationship between the outputs QA - QC (A - C) and the **LEDs**.

When you finish wiring, turn power ON and see which **LED** is ON. Take a look at Figure 1, and you'll see that you can tell about the outputs QA - QC from the **LED** which is lit. Now press **S1** to send out the pulses. The counter IC counts the pulses and lights up the **LEDs** one by one as shown in Figure 1.

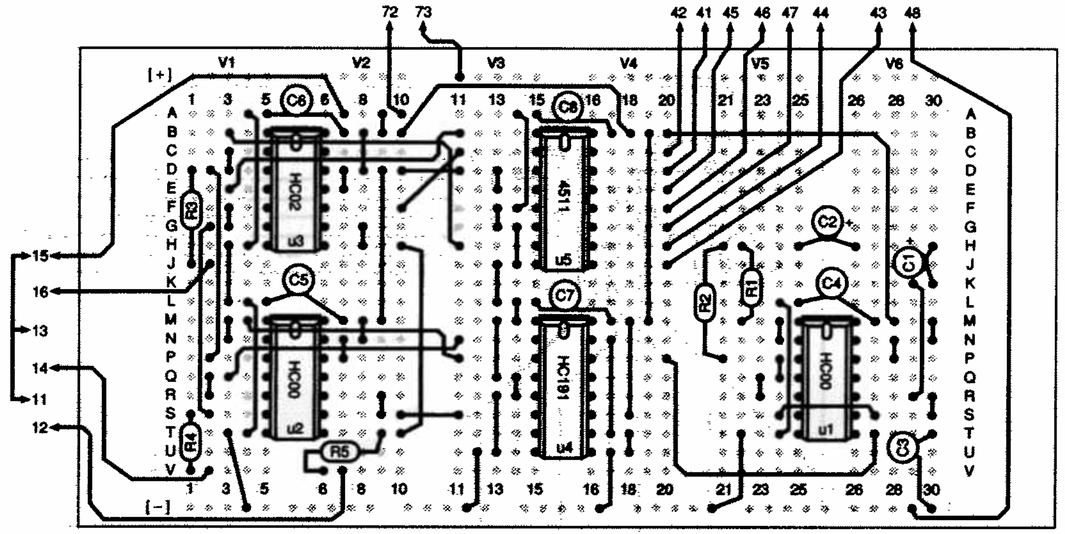


C	B	A	LED to be turned ON
0	0	0	LED1
0	0	1	LED2
0	1	0	LED3
0	1	1	LED4
1	0	0	LED5
1	0	1	LED6
1	1	0	LED7
1	1	1	LED8

1 = H
0 = L

Figure 1

PROJECT 203. OCTAL COUNTER WITH DISPLAY

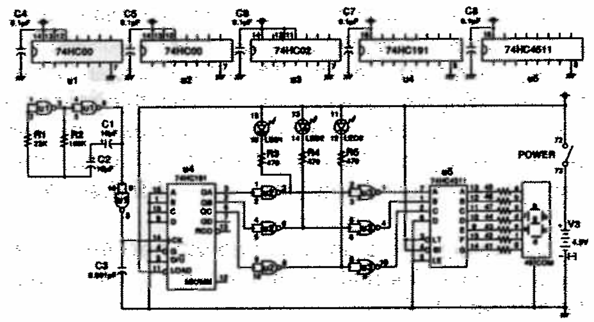


U1	74HC00	U5	74HC4511	R4	470Ω	C1	10μF	C5	0.1μF
U2	74HC00	R1	22KΩ	R5	470Ω	C2	10μF	C6	0.1μF
U3	74HC02	R2	100KΩ			C3	0.001μF	C7	0.1μF
U4	74HC191	R3	470Ω			C4	0.1μF	C8	0.1μF

Here's another octal counter whose outputs (QA - QC) light up **LEDs** and are also displayed in decimal numbers on the **LED display**.

As you'll see in the schematic for this project, an astable multivibrator made up of two NAND gates generates the clock pulses to count. The outputs of the counter, QA - QC, are decoded by the decoder IC and displayed on the **LED display**.

When you finish assembling, turn power ON. You'll notice that the **LED display** begins displaying the numbers from **1** to **2**, **3**..and so on up to **7**, and returns to **0** and repeats this counting operation over and over again.

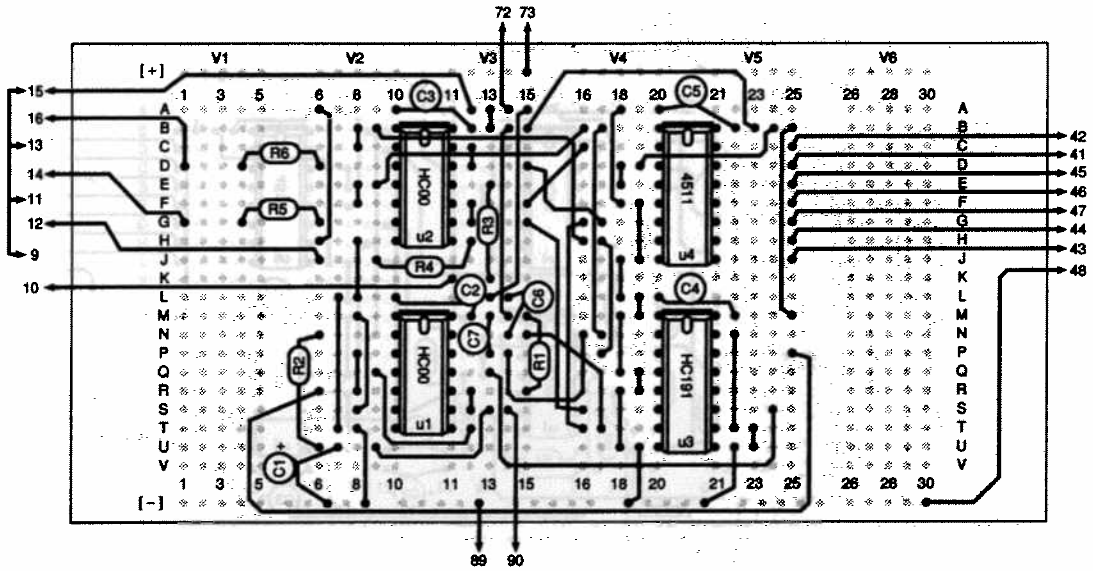


Display LED3	LED2	LED1	7-Segment Display
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

LED ON : 1 or H
LED OFF: 0 or L

Figure 1

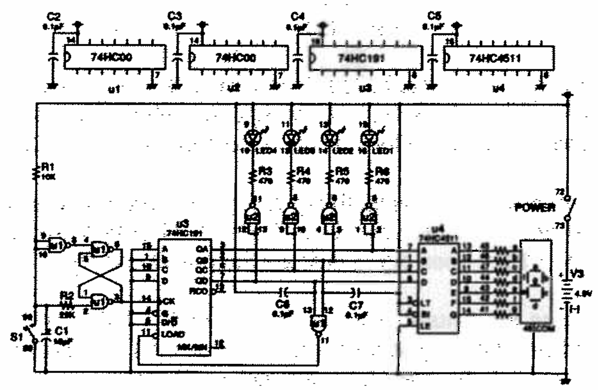
PROJECT 204. DECADE COUNTER WITH DISPLAY



U1	74HC00	R1	10KΩ	R4	470Ω	C1	10μF	C4	0.1μF
U2	74HC00	R2	22KΩ	R5	470Ω	C2	0.1μF	C5	0.1μF
U3	74HC191	R3	470Ω	R6	470Ω	C3	0.1μF	C6	0.1μF
U4	74HC4511							C7	0.1μF

Let's make a decimal counter using a counter IC. You can see by the schematic that the counter IC output is decoded with a decoder IC and display the result in decimal numbers on the **LED display**.

When you finish assembling, turn power ON. At this time, all **LEDs** are OFF and **0** is shown on the **LED display**. Press **S1** to send the pulses to the counter IC. The display changes to **1**, **2**, **3**..and so on. At the same time, you'll notice the four **LEDs** blinking ON and OFF as shown in Figure 1. Keep pressing **S1**, and the number reaches **9** and return to **0**, then change again to **1**, **2**, **3**, ... and so forth. This counting operation repeats over and over again so long as you keep pressing **S1**.

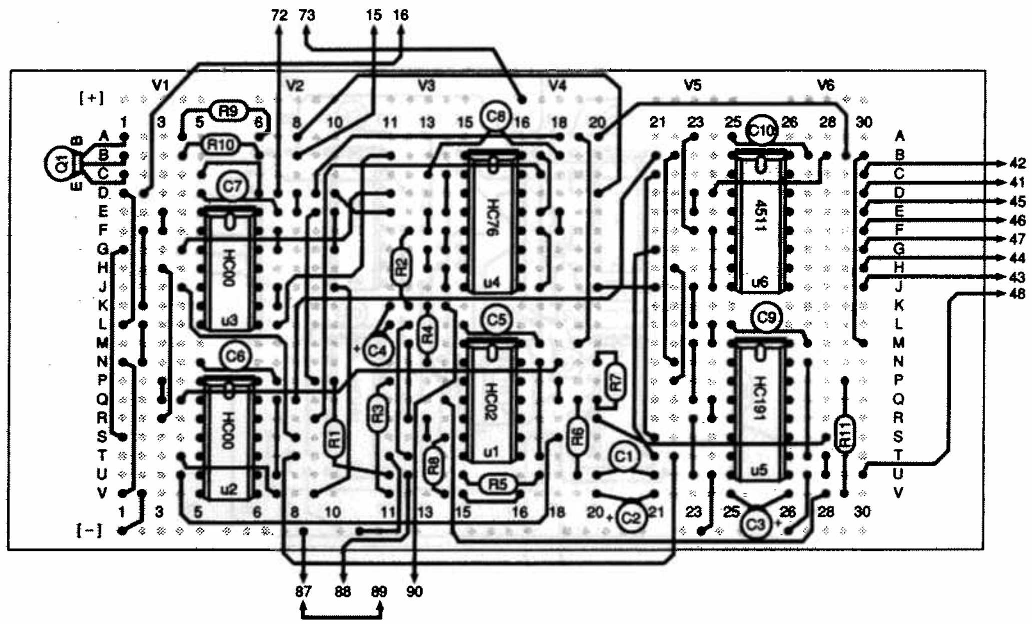


Decimal	LED4	LED3	LED2	LED1
9	1	0	0	1
8	1	0	0	0
7	0	1	1	1
6	0	1	1	0
5	0	1	0	1
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0

LED ON : 1 or H
LED OFF: 0 or L

Figure 1

PROJECT 205. DECADE COUNTER WITH DISPLAY II

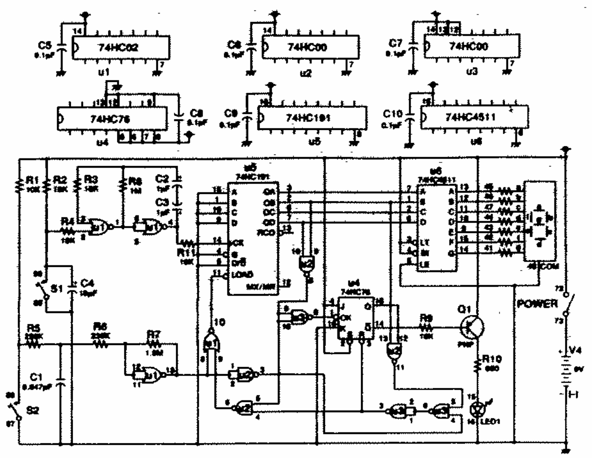


U1	74HC02	R1	10KΩ	R7	1.5MΩ	C1	0.047µF	C6	0.1µF
U2	74HC00	R2	10KΩ	R8	1MΩ	C2	1µF	C7	0.1µF
U3	74HC00	R3	10KΩ	R9	10KΩ	C3	1µF	C8	0.1µF
U4	74HC76	R4	10KΩ	R10	680Ω	C4	10µF	C9	0.1µF
U5	74HC191	R5	220KΩ	R11	10KΩ	C5	0.1µF	C10	0.1µF
U6	74HC4511	R6	330KΩ						

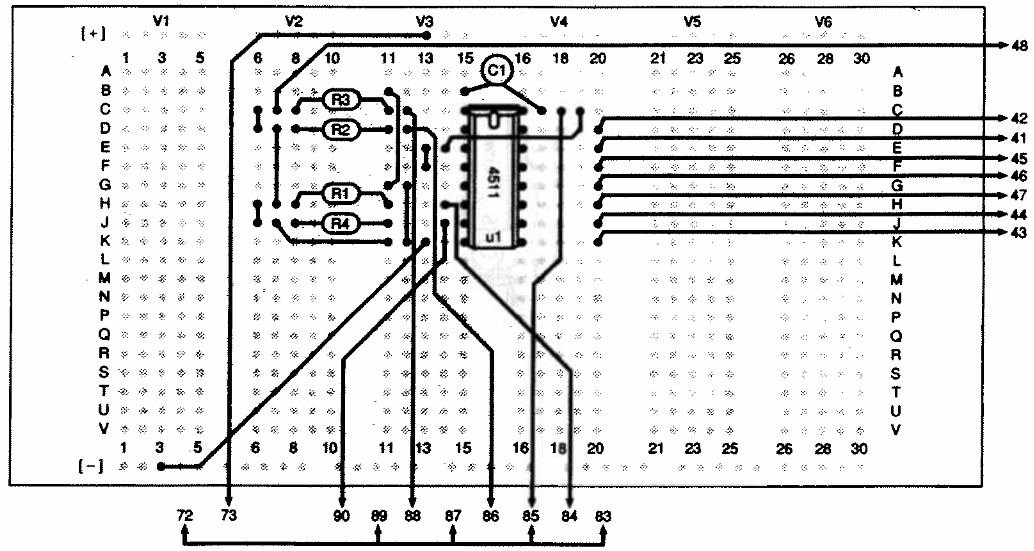
We're now going to build a decade counter that can count **12** numbers from **0** to **11**, using one **LED display** and one **LED** (**LED 1**) for indicating the carry from **9** to **10**.

When you finish assembling, turn power ON, and press **S1** and **S2** at the same time. The **LED display** shows **0** and **LED 1** goes out to indicate the preset condition.

Press **S1** a number of times, and **LED 1** lights up to indicate the carry when the number **9** on the **LED display** changes to **0**. Press **S1** again, and you'll see the number **1** on the **LED display**, which actually means **11**. If you press **S1** once again, **LED 1** goes out and the **LED display** returns to **0**.



PROJECT 206. BCD TO 7-SEGMENT DECODER

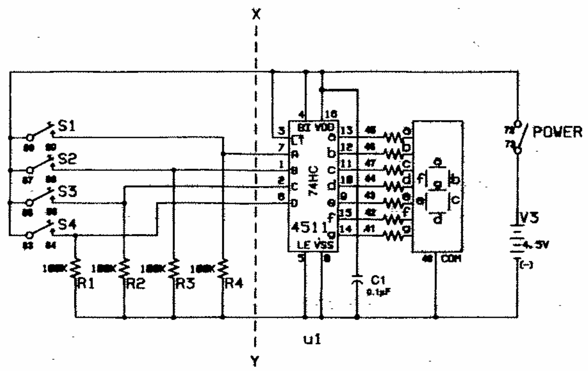


U1	74HC4511	R1	100KΩ	R3	100KΩ	C1	0.1µF
		R2	100KΩ	R4	100KΩ		

In the digital world of computers, data is processed using combinations of 4-bit, 8-bit, 16-bit, or 32-bit pulses. But these are a machine language hard to understand for us humans. So, it's necessary to convert this machine language to numbers understandable to us. This can be realized by the BCD to 7-Segment Decoder that displays the decoded notations on the **LED display**.

In this project, we're going to find out how 4-bit BCD (Binary Coded Decimal) notations can be converted to the numbers 0 - 9.

Keys **S1 - S4** are used for input of A - D, as shown in Figure 1. When any of these switches is pressed, it gives a "1." When it is turned OFF, a "0" is input, Try producing different numbers on the display by manipulating these keys.



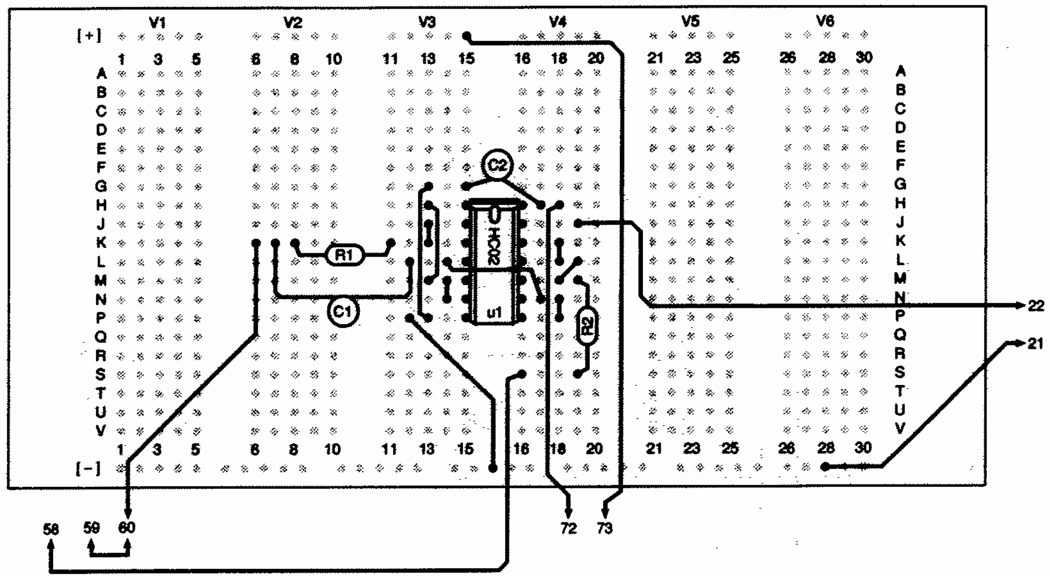
BCD to 7-Segment Decoder

S4	S3	S2	S1	DISPLAY
D	C	B	A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

S1-S4 ON : 1
S1-S4 OFF: 0

Figure 1

PROJECT 207. VCO BY NOR GATE

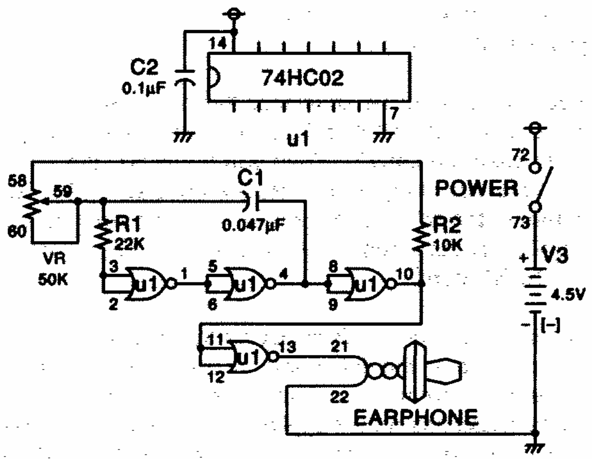


U1	74HC02	R1	22KΩ	C1	0.047µF
		R2	10KΩ	C2	0.1µF

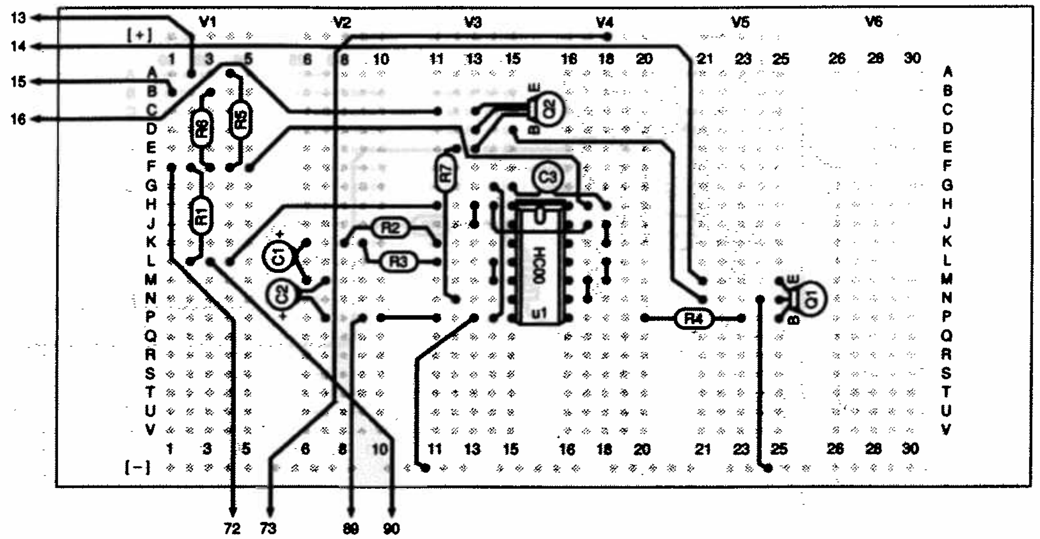
Now, we're going to assemble a tone generator that emits a tone in varying height of pitch as you turn the **control volume**. Tone generators of this type are used as the sound source of music synthesizers and electronic organs.

This tone generator uses the VCO made up by NOR gate. The oscillation frequency is determined by the value "C1 + (R + **control volume**)". So, the oscillation frequency varies when the **control volume** is turned.

Now, let's see how this VCO works. Turn power ON first, put the earphone in you ear, and turn the **control volume**. You'll note that the tone becomes lower when you turn the **control volume** clockwise, and higher when you turn the **control volume** counterclockwise.



PROJECT 208. PULSE-DELAYED CIRCUIT

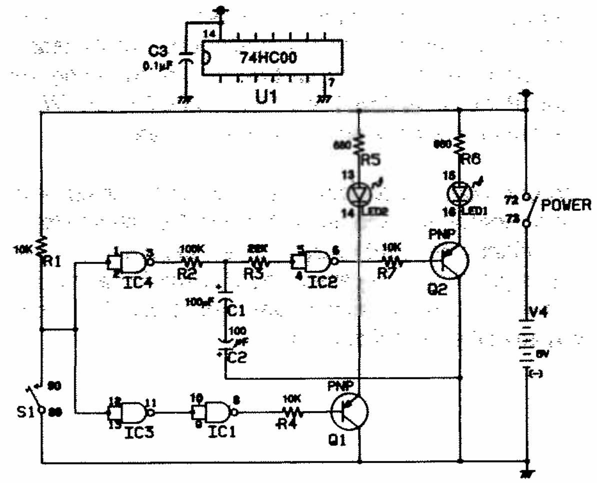


U1	74HC00	R1	10KΩ	R4	10KΩ	R7	10KΩ	C1	100µF		
Q1	PNP	R2	100KΩ	R5	680Ω			C2	100µF		
Q2	PNP	R3	22KΩ	R6	680Ω			C3	0.1µF		

We're now going to make a pulse delayed circuit using a NAND gate. This circuit does not produce an output immediately when it is turned ON, but only after the lapse of a pre-determined time. For comparison with this circuit, we'll make another circuit that generates an output immediately when it is switched ON.

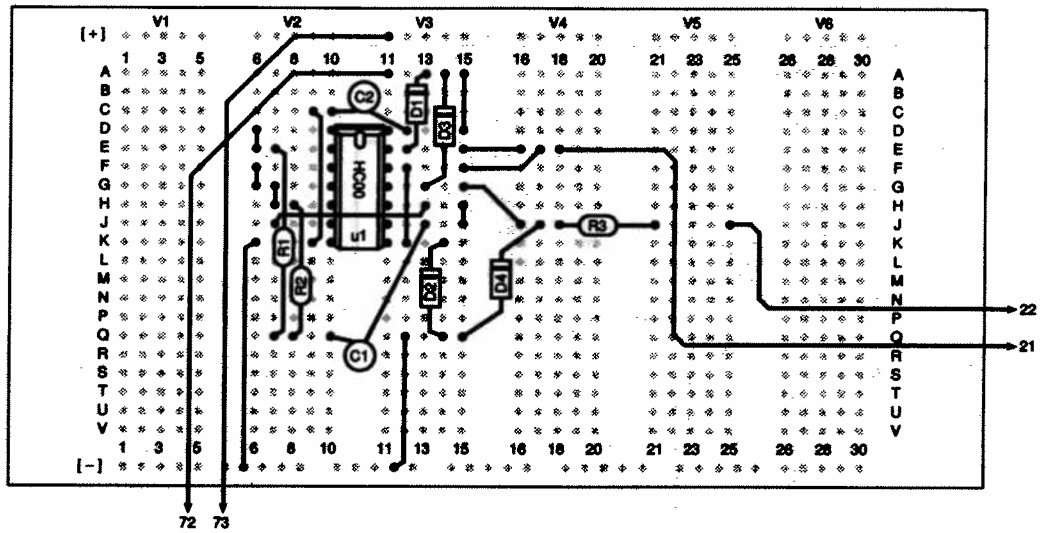
Take a look at the schematic for this project. You'll see that the circuit made up of IC1 and Q1 generates an output immediately when **S1** is pressed, so **LED 2** also lights up immediately when **S1** is pressed.

This doesn't happen with the other circuit made up of IC2 and Q2; it gives out an output after the lapse of a delay time determined by C (combined capacitance of C1 and C2 with 50µF) and R2.



Turn power ON and press **S1** to see the delayed operation.

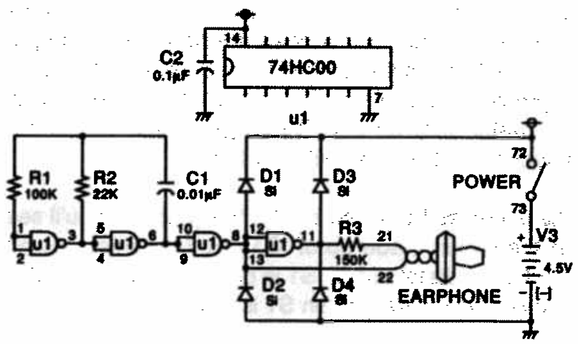
PROJECT 209. NAND GATE TONE GENERATOR



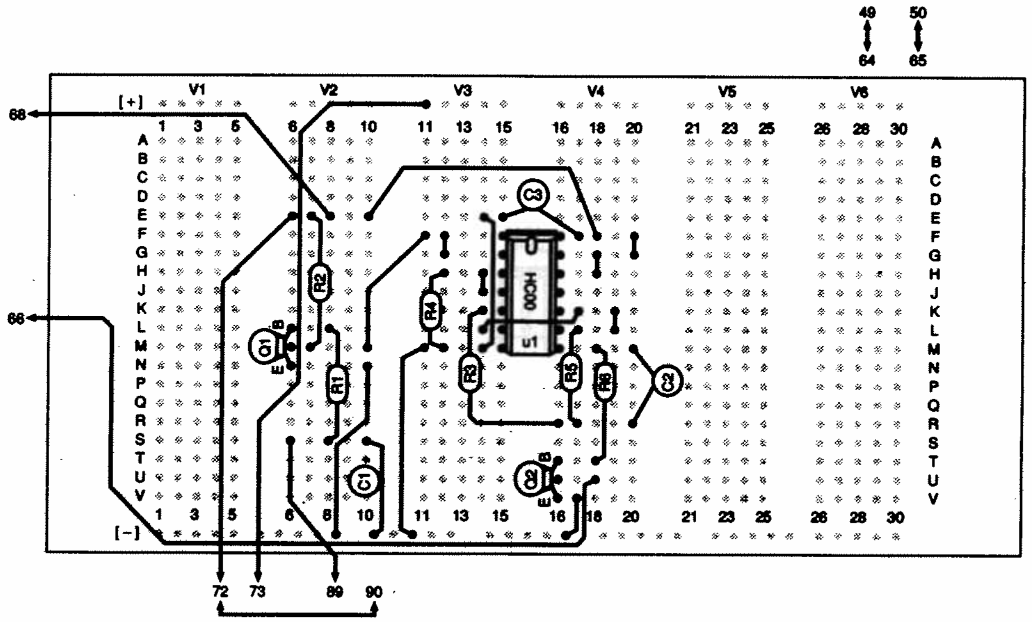
U1	74HC00	R2	22KΩ	C1	0.01μF	D1	Si	D3	Si
R1	100KΩ	R3	150KΩ	C2	0.1μF	D2	Si	D4	Si

When you finish the wiring, connect the earphone to terminals 21 and 22 and turn power ON. You'll hear a tone produced by the multivibrator. Try changing the value of the capacitors from 0.01μF to 0.047μF. What effect does this have on the sound you hear?

Try some sort of arrangement so you can switch different value capacitors in and out of this circuit to vary the tone. You might also want to try different capacitors in this project (don't try using any of the electrolytic capacitors, however). Can you think of any way to use this circuit with any other digital circuits?



PROJECT 210. TRANSISTOR TIMER



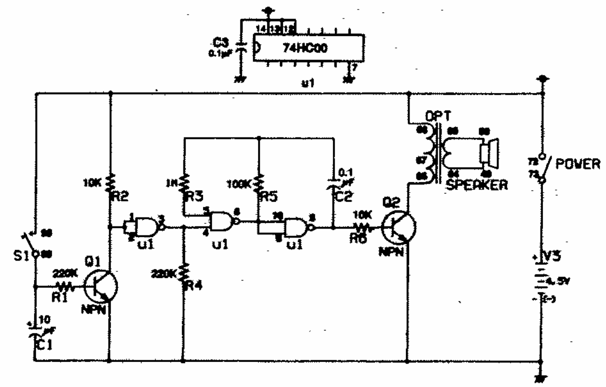
U1	74HC00	R1	220KΩ	R3	1MΩ	R5	100KΩ	C1	10μF
Q1	NPN	R2	10KΩ	R4	220KΩ	R6	10KΩ	C2	0.1μF
Q2	NPN							C3	0.1μF

Here's another type of "one shot" circuit, but this time you hear the results.

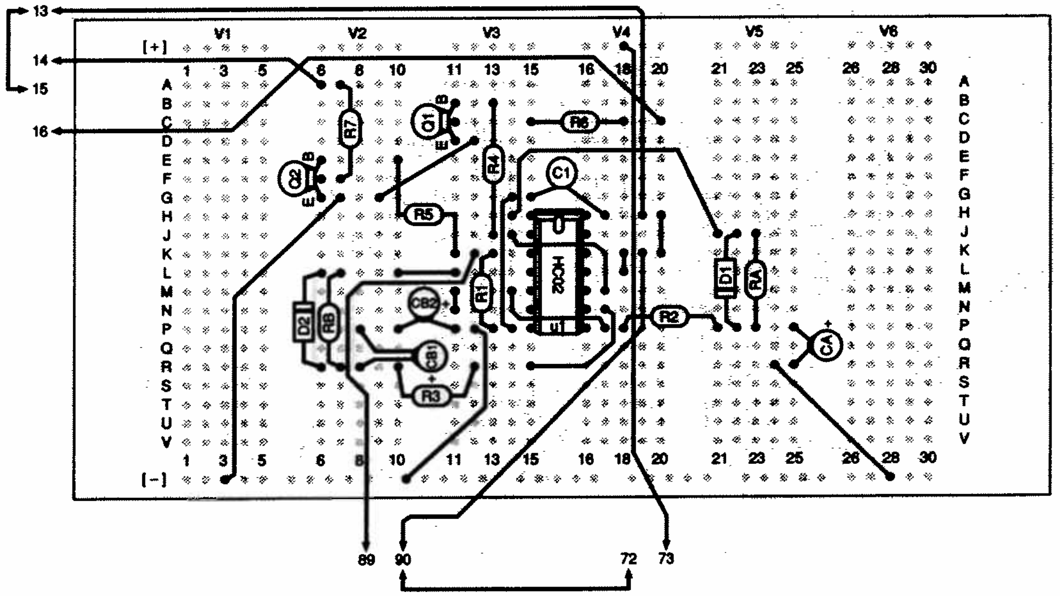
You can see from the schematic that this project uses a combination of digital and analog electronics. (Forgot what analog means?) When you press S1, the 10μF capacitor are charged and lets the transistor in the left corner of the schematic operate. You can see that the collector of this transistor serves as both inputs for the first NAND gate.

The digital portion of this circuit controls the operation of the transistor in the right corner of the schematic. Turn power ON. When the output of the digital portion is 1, the transistor has current applied to its base and operates ... and you hear a sound from the speaker. When S1 is released and the 10μF capacitor discharges, the first transistor can no longer operate. The output of the digital section becomes 0, and the second transistor can no longer operate.

Try to change the value of C1 and see what happens.



PROJECT 211. NOISE-SIGNAL DISCRIMINATOR



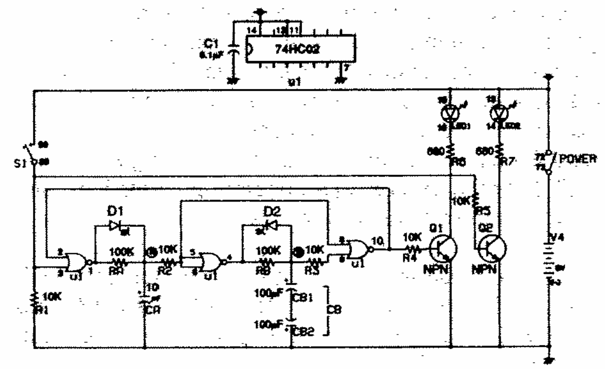
U1	74HC02	RA	100KΩ	R3	10KΩ	R7	680Ω	C1	0.1μF
Q1	NPN	RB	100KΩ	R4	10KΩ	CA	10μF	D1	Si
Q2	NPN	R1	10KΩ	R5	10KΩ	CB1	100μF	D2	Si
		R2	10KΩ	R6	680Ω	CB2	100μF		

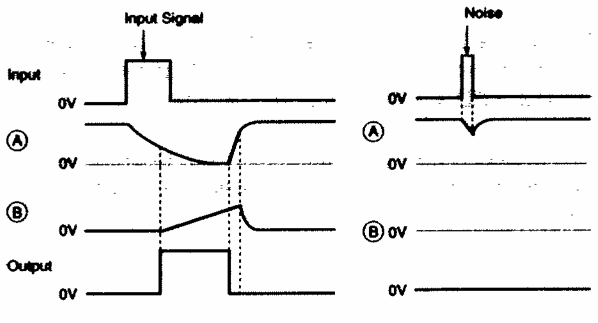
When listening to an AM radio, we often hear a scratching or grating noise which is very offensive to the ear. In this project, we're going to make a circuit that can discriminate between noise and signals using a NOR gate. The discrimination can be made by the signal width, because noises are small in width but signals are large in width.

The schematic shows you that the **LEDs** are used for noise and signal indication. **S1** is used to generate signals, and **LED 1** lights up when a noise is generated. Noise and signal discrimination is made by NOR gate IC - it doesn't light up **LED 1** unless the signal received has a certain width (see Fig. 1).

When you've wired the project, turn power ON and press **S1** just for a moment (less than 1 second). You'll notice that **LED 2** lights up but **LED 1** doesn't. This is because a noise is generated. Now press **S1** for more than 1 second. This time **LED 2** lights up immediately, and **LED 1** stays ON for about 5 seconds after **S1** is released.

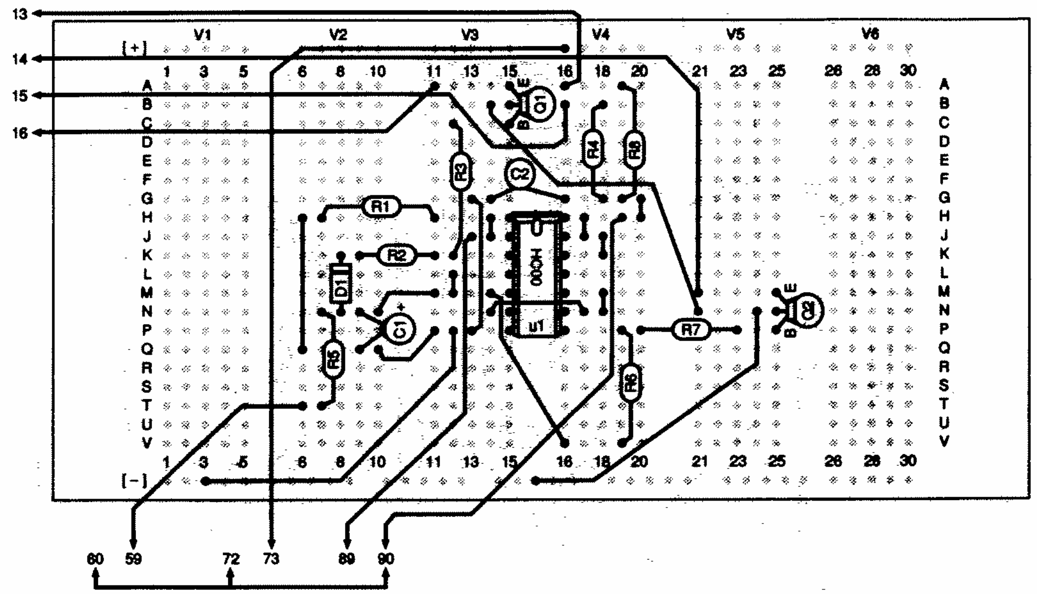
Finally, press **S1** for more than 1 second and release it, then press it again for 1 - 2 seconds. **LED 2** lights up immediately, but **LED 1** indicates only the first input signal (generated by pressing **S1** for more than 1 second), but not the second input signal (generated by pressing **S1** for 1 - 2 seconds).





< Fig-1(A) > < Fig-1(B) >

PROJECT 212. PULSE STRETCHER

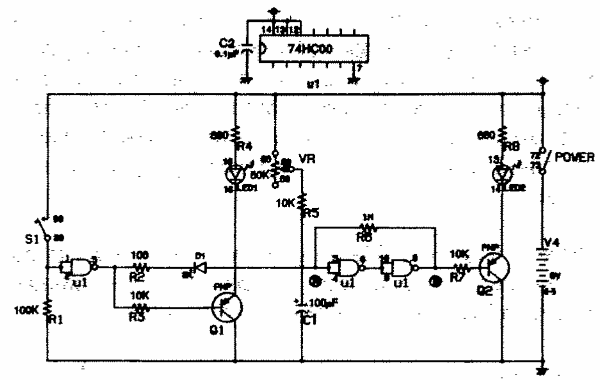


U1	74HC00	R1	100KΩ	R4	680Ω	R7	10KΩ	C1	100μF
Q1	PNP	R2	100Ω	R5	10KΩ	R8	680Ω	C2	0.1μF
Q2	PNP	R3	10KΩ	R6	1MΩ				

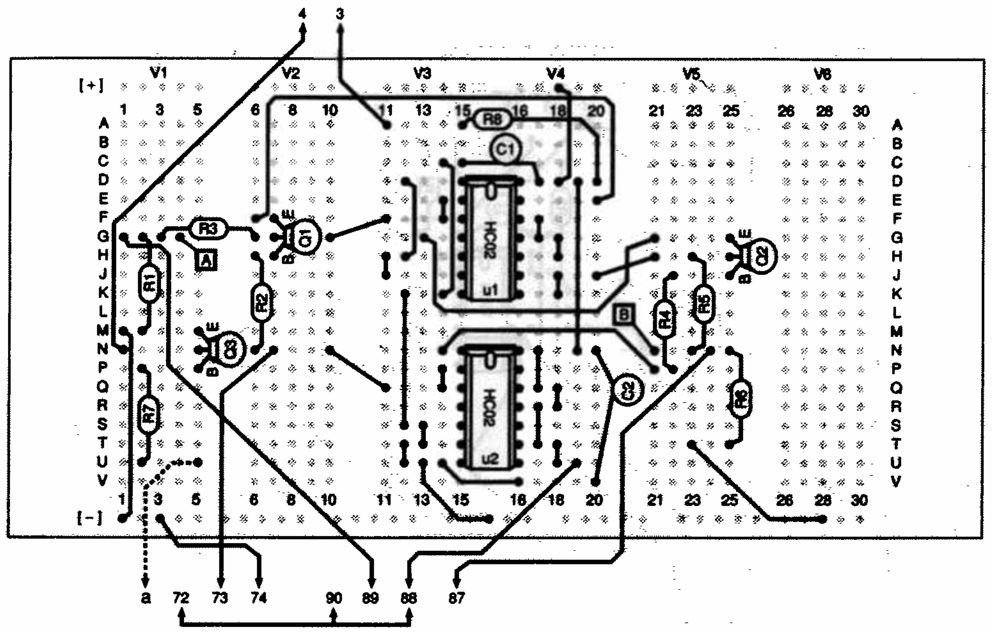
We're now going to make a pulse stretcher that can generate a pulse with a large or small width, just as we want when a trigger pulse is input.

As you'll see in the schematic, **S1** is used to generate a trigger pulse. It is sent to the Schmitt circuit that generates a new pulse determined by R (R5 and the **control volume**). The width of this new pulse can be changed by turning the **control volume**.

When you finish assembling, rotate the **control volume** fully clockwise and turn power on. Then, flip **S1**, and **LED 1** goes out immediately and **LED 2** also goes out about a second later. Now turn the **control volume** counterclockwise, and again flip **S1**. What happens now? It takes a longer time, a few seconds, for **LED 2** to goes out this time. Set the **control volume** in different positions and see how this time lag (which is the pulse width) changes.

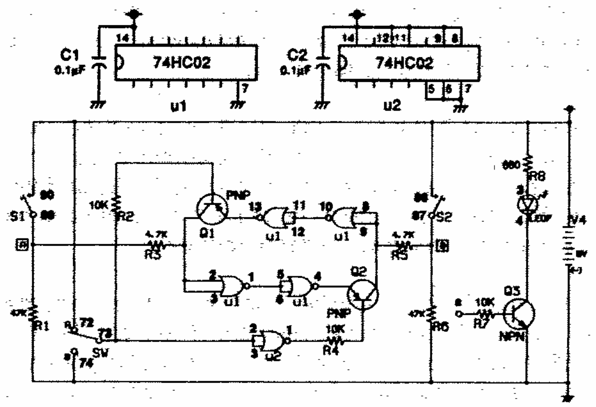


PROJECT 213. BIDIRECTIONAL BUFFER



U1	74HC02	Q2	PNP	R2	10KΩ	R5	4.7KΩ	R8	680Ω
U2	74HC02	Q3	NPN	R3	4.7KΩ	R6	47KΩ	C1	0.1µF
Q1	PNP	R1	47KΩ	R4	10KΩ	R7	10KΩ	C2	0.1µF

Bus lines play an important role in exchanging signals between many elements and circuits in a microcomputer. In this project, we're going to experiment with a bidirectional buffer used as a common bus for both input data and output data. Microcomputers generally use 8-bit, 16-bit or 32-bit bus lines, but we're going to use a 1-bit bus.

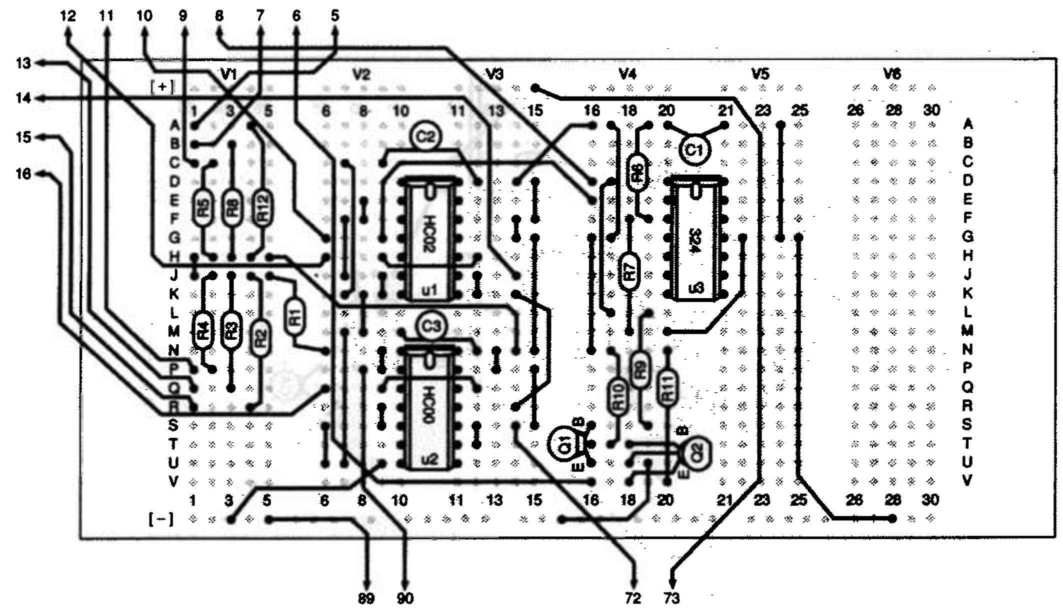


What is bus? You ride on a bus when you want to go somewhere. In computers, the data or control signals ride on a bus to go somewhere to do the job.

When you finish assembling, set the **select switch** up and press **S1**. Then connect terminal a to B. **LED 7** lights up. This means that the signal data is flowing from A to B.

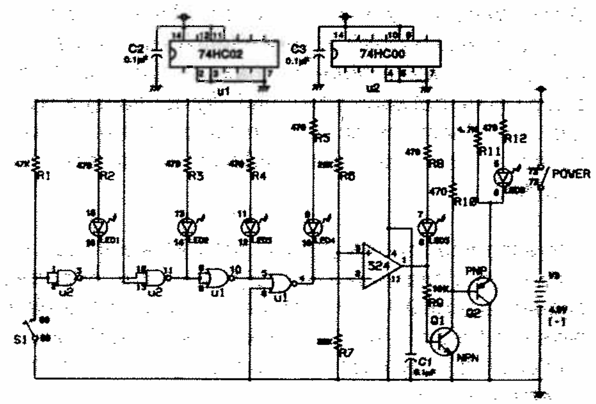
Now set the **select switch** down, press **S2**, and connect terminal a to A. You'll notice **LED 7** lights up again. This indicates that the signal data is flowing from B to A. This is how the 1-bit bidirectional buffer works.

PROJECT 214. VARIOUS INVERTERS



U1	74HC02	Q2	PNP	R4	470Ω	R8	470Ω	R12	470Ω
U2	74HC00	R1	47KΩ	R5	470Ω	R9	10KΩ	C1	0.1µF
U3	324	R2	470Ω	R6	22KΩ	R10	470Ω	C2	0.1µF
Q1	NPN	R3	470Ω	R7	22KΩ	R11	4.7KΩ	C3	0.1µF

Inverters are made in many different ways, and we're going to experiment with a few of them in this project. Look at the schematic for this project. **LED 1 - LED 6** are used to display the output of each inverter. **LED 1 - LED 2** are used for the inverter using a NAND gate circuit, **LED 3 - LED 4** for the one using a NOR gate circuit, **LED 5** for the one using an operational amplifier, and **LED 6** for the one using a transistor.

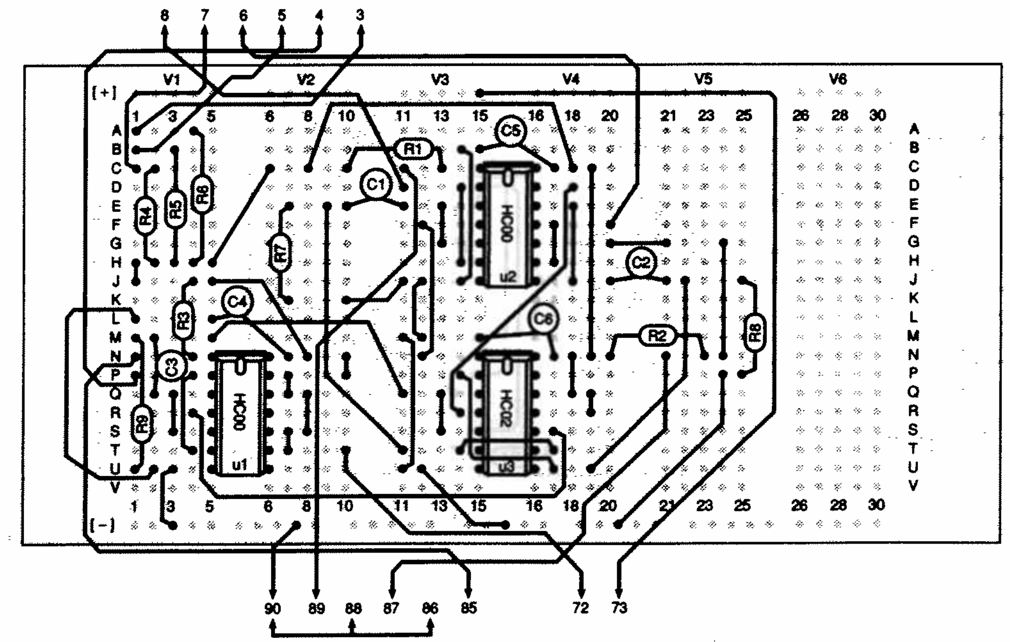


When you finish assembling, turn power ON and you'll see that the odd numbered **LEDs** light up (**LED 1, LED 3, LED 5**). This is because the input of the first inverter is 1 when **S1** is OFF.

Now press **S1** to make the input to the first inverter 0. What happens to the **LEDs**? **LEDs** with an even number (**LED 2, LED 4, LED 6**) light up this time.

The project shows you that every other **LED** lighted up, and this means that each **LED** shows the action of each inverter.

PROJECT 215. ELECTRONIC SWITCH

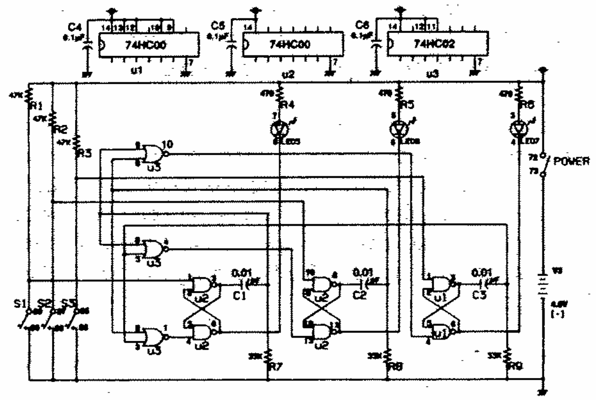


U1	74HC00	R1	47KΩ	R5	470Ω	R9	33KΩ	C4	0.1μF
U2	74HC00	R2	47KΩ	R6	470Ω	C1	0.01μF	C5	0.1μF
U3	74HC02	R3	47KΩ	R7	33KΩ	C2	0.01μF	C6	0.1μF
		R4	470Ω	R8	33KΩ	C3	0.01μF		

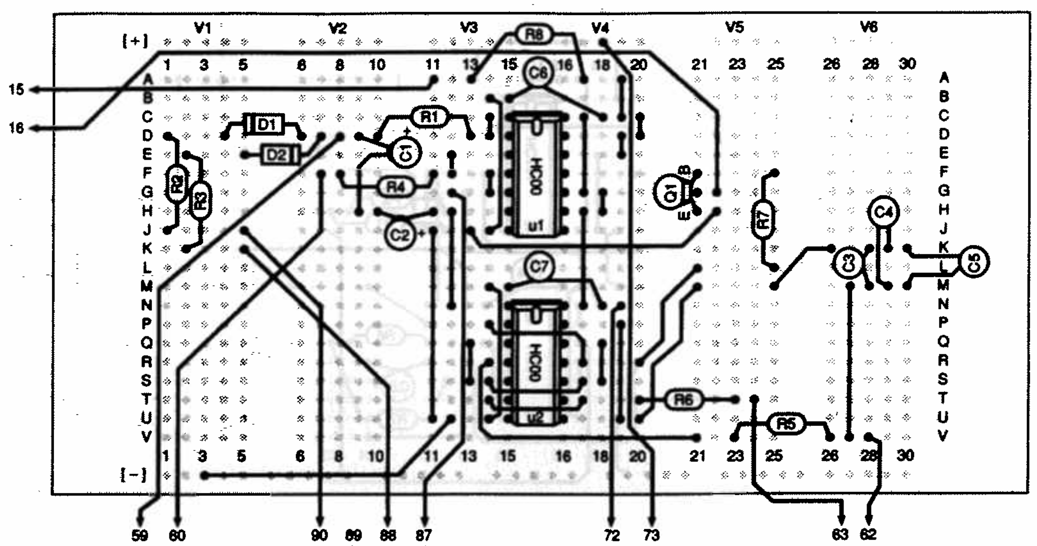
We're now going to make of an interesting experiment - electronic switch. Turn power ON and press **S1** just for a moment. Then **LED 5** stays ON and **LED 6** and **LED 7** stay OFF. This means that the electronic switch of **LED 5** is ON.

You can observe the same performance by pressing **S2** or **S3**. When any of these switches is pressed to turn ON any electronic switch, all other electronic switches are automatically turned OFF.

This electronic switch is made up of three sets of R-S flip-flop circuit. If you press any one switch and the corresponding electronic switch is turned ON, a reset signal is sent to all other electronic switches. TV channel switches and stereo control switches are examples of this electronic switch.



PROJECT 216. TONE BURST GENERATOR



U1	74HC00	R2	47KΩ	R6	33KΩ	C2	10μF	C6	0.1μF
U2	74HC00	R3	47KΩ	R7	10KΩ	C3	0.1μF	C7	0.1μF
Q1	NPN	R4	33KΩ	R8	470Ω	C4	0.1μF	D1	Si
R1	220KΩ	R5	47KΩ	C1	10μF	C5	0.1μF	D2	Si

A tone generator is an oscillator that sends out signals repeated at regular intervals, as shown in Figure 1. As its name suggests, a speaker is usually used to let you hear the tone it makes. But in this project, we're going to use an **LED** to find out how it works.

You can see from the schematic, IC U2 is a tone generator whose frequency can be changed by 100K **control volume**. IC U1 is another generator which controls the start and stop of the tone generator. Its frequency can also be changed by turning 50K **control volume**, and its duty ratio can be adjusted using **S1** and **S2**. Remember what the duty ratio is? Refer back to project 53. The duty ratio is about 24% when **S1** is ON, and about 76% when **S2** is ON. This project can be used as a logic circuit whose waveform ends in an integer cycle.

When you finish assembling the project, turn power ON and see what the **LED** is doing. Does it blink ON and OFF as shown in Figure 1? Then, turn 50K **control volume** and press **S1** and **S2**, and see how the **LED** changes its blinking intervals.

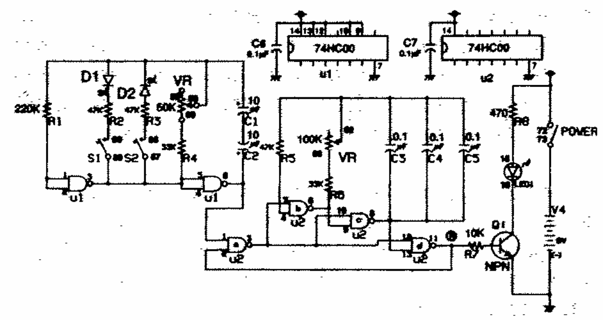
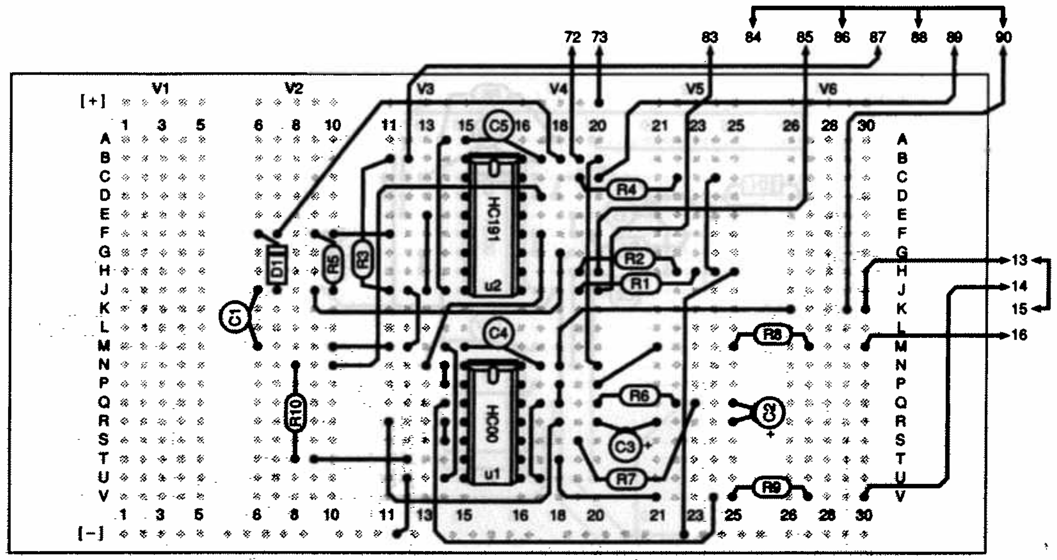


Figure 1

PROJECT 217. DIGITAL TIMER



U1	74HC00	R3	10KΩ	R7	330KΩ	C1	0.1μF	C5	0.1μF
U2	74HC191	R4	10KΩ	R8	470Ω	C2	10μF	C6	0.1μF
R1	10KΩ	R5	220KΩ	R9	470Ω	C3	10μF	D1	Si
R2	10KΩ	R6	47KΩ	R10	47KΩ	C4	0.1μF		

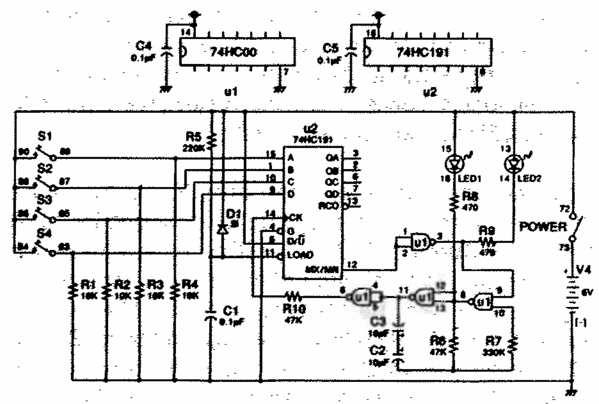
We're now going to build a digital timer using a counter IC. Since the digital timer works by counting the number of clock pulses, it is much more accurate than analog timers so long as the clock pulse repetition period (time) is maintained constant and free from any change.

Since the counter IC works as a presettable binary down counter, you can preset the count in 4-bit binary numbers, as Figure 1 shows. The clock pulses are generated by a NAND gate, fed to the CK terminal of the counter IC, and displayed by **LED 1**.

Now, set **S1 - S4** to the count you've selected from Figure 1 (if the count you've chosen is 11, it is equivalent to a binary number of 1011, so turn **S1**, **S2** and **S4** ON, and turn **S3** OFF).

Then turn power ON. **LED 1** starts flashing on and off during counting the preset number. **LED 2** lights up after the preset number has been counted.

The setting of this digital timer can be changed by varying the clock pulse repetition period (time). You can make this period longer by using a larger value for C2 or R6. So try changing the timer setting.

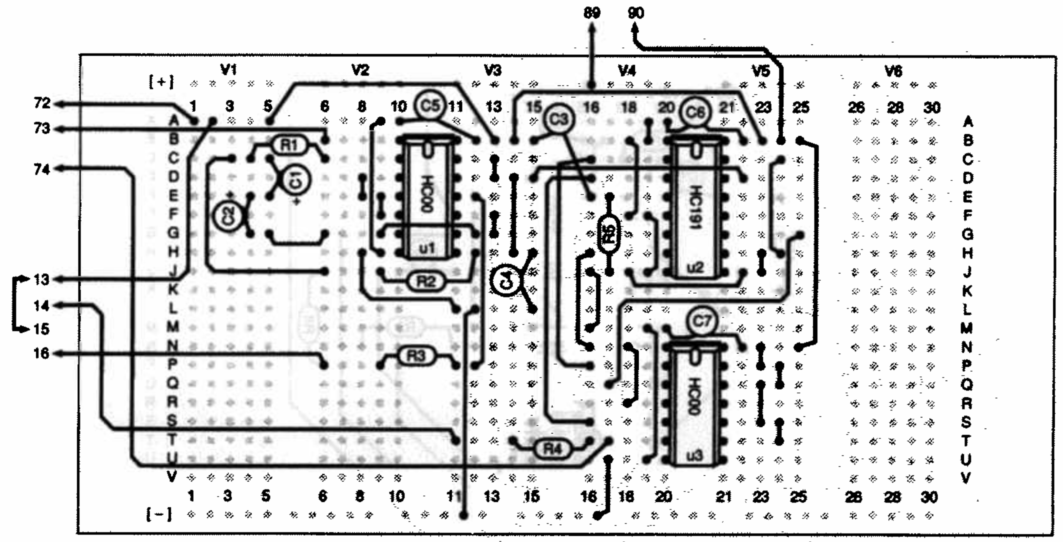


Switch No. 4 3 2 1	Number of LED1 Count	Switch No. 4 3 2 1	Number of LED1 Count
0 0 0 0	0	1 0 0 0	8
0 0 0 1	1	1 0 0 1	9
0 0 1 0	2	1 0 1 0	10
0 0 1 1	3	1 0 1 1	11
0 1 0 0	4	1 1 0 0	12
0 1 0 1	5	1 1 0 1	13
0 1 1 0	6	1 1 1 0	14
0 1 1 1	7	1 1 1 1	15

Switch ON : 1 or H
Switch OFF: 0 or L

Figure 1

PROJECT 218. DIGITAL TIMER II

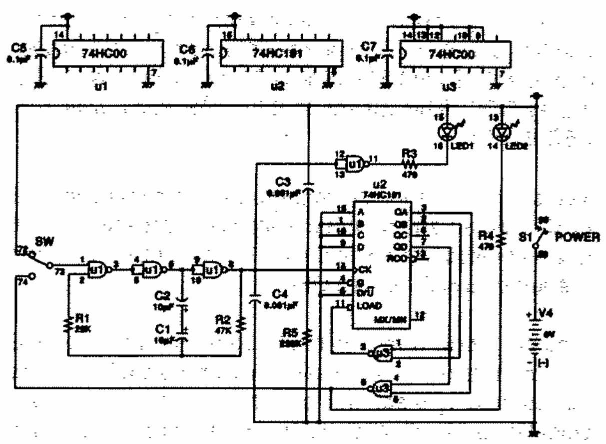


U1	74HC00	R1	22KΩ	R4	470Ω	C2	10μF	C5	0.1μF
U2	74HC191	R2	47KΩ	R5	220KΩ	C3	0.001μF	C6	0.1μF
U3	74HC00	R3	470Ω	C1	10μF	C4	0.001μF	C7	0.1μF

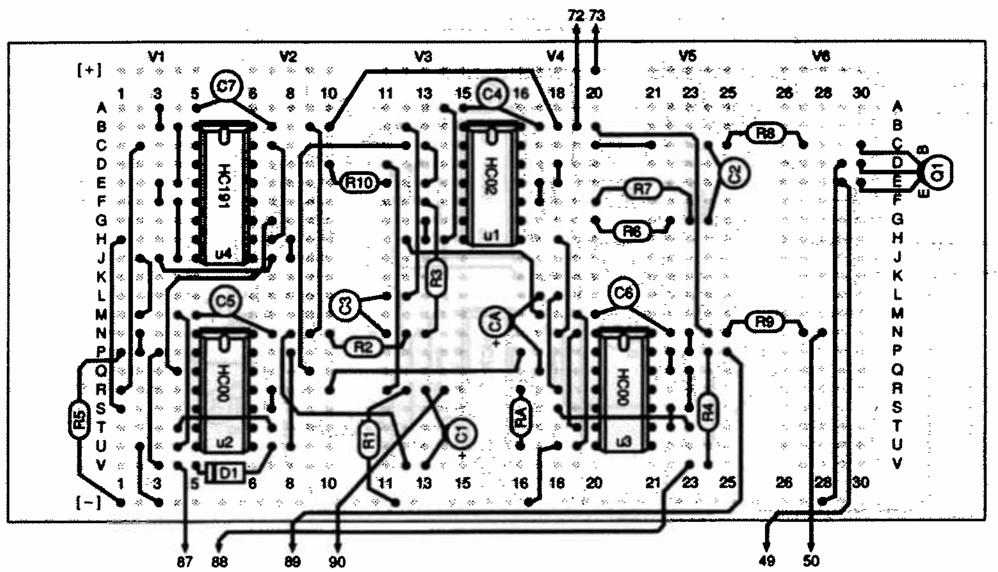
Here's a digital timer using a counter IC. It can count clock pulses with a separation of one second and give an output when the 10 pulses are counted up.

The schematic shows you that U1 is the oscillator that generates the clock pulses at intervals of one second, and U2 is the counter whose output is displayed by **LED 2**. When you set the **slide switch** up, the counter keeps counting the pulses continuously. When it is set to down, the counter lights up **LED 2** on counting the 10th pulse, then stops.

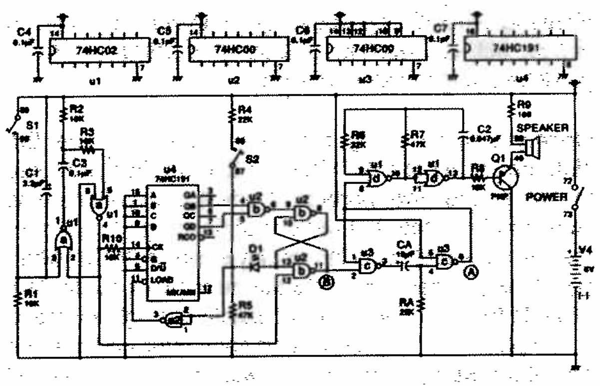
When you finish the wiring, turn power ON and press **S1** to operate the timer. If you keep pressing **S1** with the **slide switch** up, **LED 1** blinks ON and OFF at intervals of one second and **LED 2** lights up in 10 seconds (when the 10th pulse is counted). The timer repeats this operation over and over again so long as you keep pressing **S1**.



PROJECT 219. TEN COUNT BUZZER



U1	74HC02	R1	10KΩ	R6	22KΩ	RA	22KΩ	C5	0.1µF
U2	74HC00	R2	10KΩ	R7	47KΩ	C1	3.3µF	C6	0.1µF
U3	74HC00	R3	10KΩ	R8	10KΩ	C2	0.047µF	C7	0.1µF
U4	74HC191	R4	22KΩ	R9	100Ω	C3	0.1µF	CA	10µF
Q1	PNP	R5	47KΩ	R10	10KΩ	C4	0.1µF	D1	Si



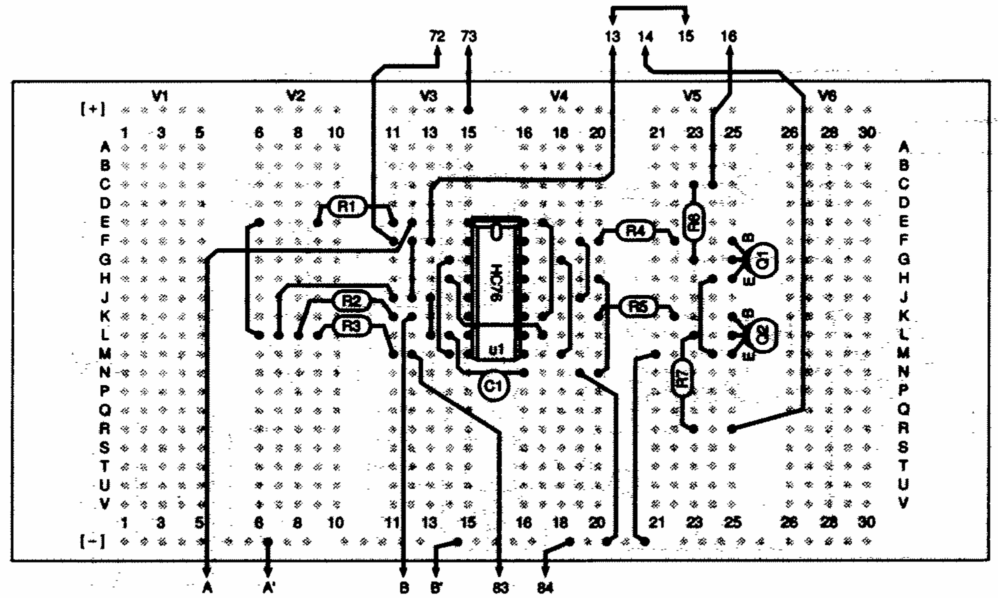
How would you like to make a buzzer that sounds only if you press the key 10 times, not 8 times or 12 times. You need to press the key exactly 10 times to make this buzzer sound, and we'll find out why in this project.

This buzzer is a ten count buzzer that counts the number of pulses using the counter IC. You can see by the schematic that the pulses are generated by **S1** and U1 "a". IC2 is a counter that generates an output from "b" after receiving 10 input pulses. **S2** is a reset switch for resetting the count to zero.

"c" is a circuit that causes the buzzer to sound for the preset time when the output is produced. "d" is an oscillator which generates the buzzer sound. Its output signal is amplified by Q1 and activates the **speaker**.

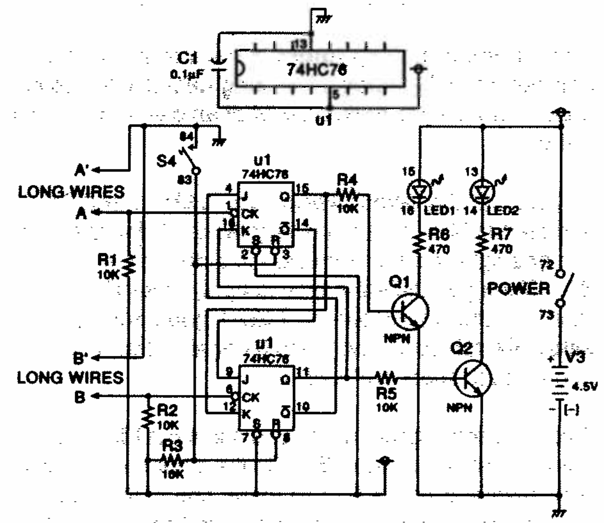
Let's see how the project works. Turn power ON and press **S2** to reset the counter. Now you can check the buzzer performance. Press **S1** 10 times. Do you hear a buzzer sound from the **speaker**? Be sure to press **S2** if you want to use this project again.

PROJECT 220. PRESS FIRST



U1	74HC76	R1	10KΩ	R4	10KΩ	R7	470Ω		
Q1	NPN	R2	10KΩ	R5	10KΩ	C1	0.1µF		
Q2	NPN	R3	10KΩ	R6	470Ω				

Who's the fastest draw in the West? (Or even the East?) Now you can find out with this project! You and your opponent face each other like Western gunfighters and "go for it" - fortunately, you'll be reaching for a pair of long wires instead of a sixshooter.

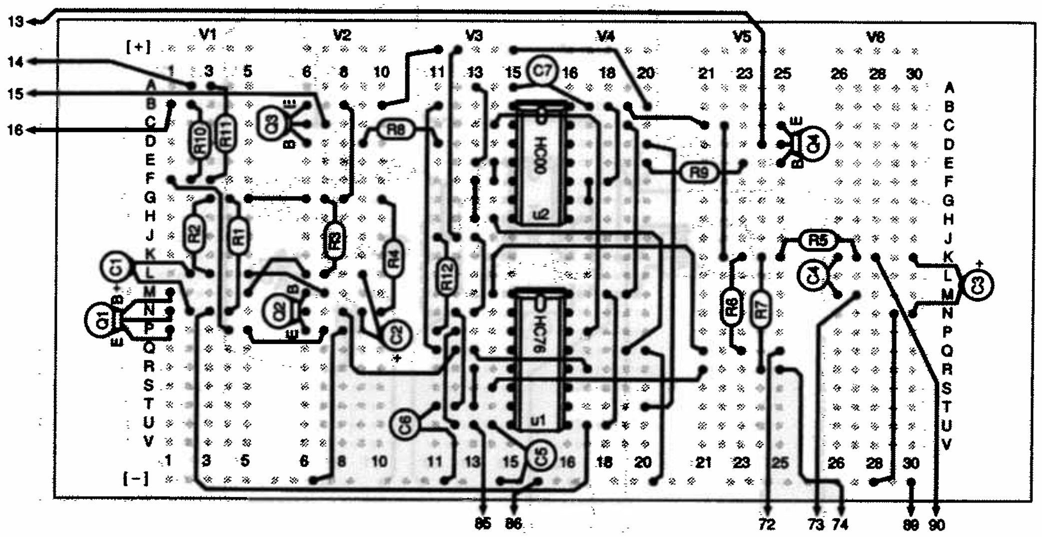


To play this game - first you have to build it - then you take the long wires from A' and A while your opponent takes the ones from B' and B. Turn power on and press **S4** once. Have someone else give the signal to draw. When this happens, try to touch the exposed ends of your long wires together before your opponent does. If you touched your wires first, **LED 1** lights up. If your opponent touched first, **LED 2** lights. If it's a draw, both **LEDs** light.

You can play again just by pressing **S4**. This turns off the **LEDs** and reset the circuit. Don't hold the ends of the long wires together after they touch - if you do, both **LEDs** might come on even if the game wasn't a draw.

Be sure to use different colored long wires when building this project (or you could wind up with your gun in your opponent's holster!).

PROJECT 221. TARGET RANGE



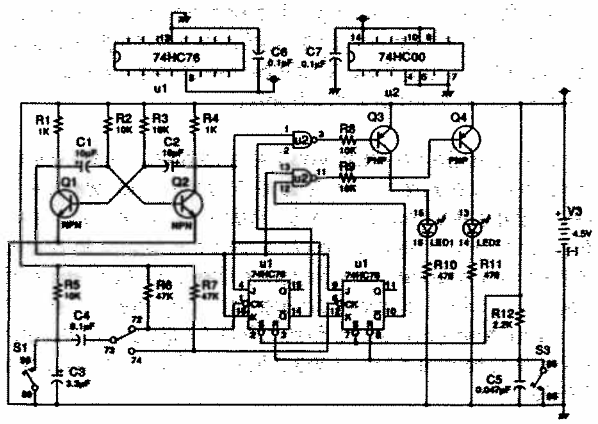
U1	74HC76	Q4	PNP	R5	10KΩ	R10	470Ω	C3	3.3µF
U2	74HC00	R1	1KΩ	R6	47KΩ	R11	470Ω	C4	0.1µF
Q1	NPN	R2	10KΩ	R7	47KΩ	R12	2.2KΩ	C5	0.047µF
Q2	NPN	R3	10KΩ	R8	10KΩ	C1	10µF	C6	0.1µF
Q3	PNP	R4	1KΩ	R9	10KΩ	C2	10µF	C7	0.1µF

If you think a moving target is hard to hit, just wait until you try to hit a pair of blinking **LEDs**! This project tests your ability to anticipate what an electronic circuit will do.

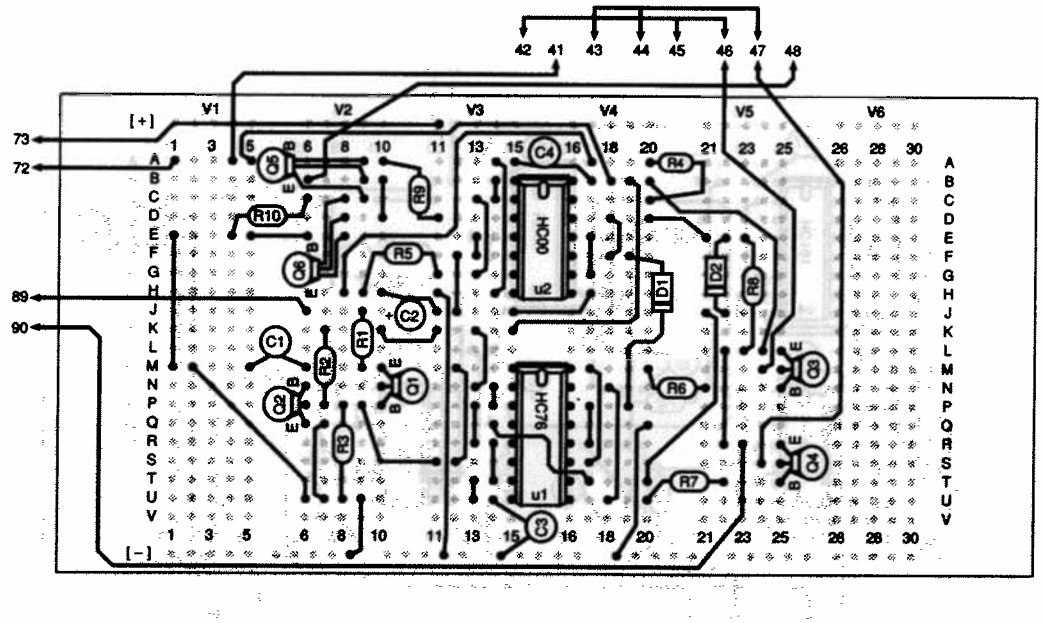
To play, set the **select switch** up. **LEDs 1** and **2** light and start to blink off and on very rapidly (it looks almost like one light moving from side to side). Your "rifle" is **S1**. When you think the time is right, press **S1**. If you "fired" at the right moment, **LED 1** goes out. Set the **select switch** down and press **S1** again. If your "aim" is good, you'll "shoot out" **LED 2**.

If you're a really good "shot", you might get **LED 1** and **2** out with just one press of **S1**. On the other hand, it might take several press of **S1** just to get one of the **LEDs** to go out. Be sure to set the **select switch** down before trying to "shoot" **LED 2**. If you don't, you'll discover that ... (hold on, we shouldn't tell you everything. Try if for yourself and see what happens.)

When the both **LEDs** are off, press the **S3** key. This means the circuit stays at the start position and you are ready to start again.



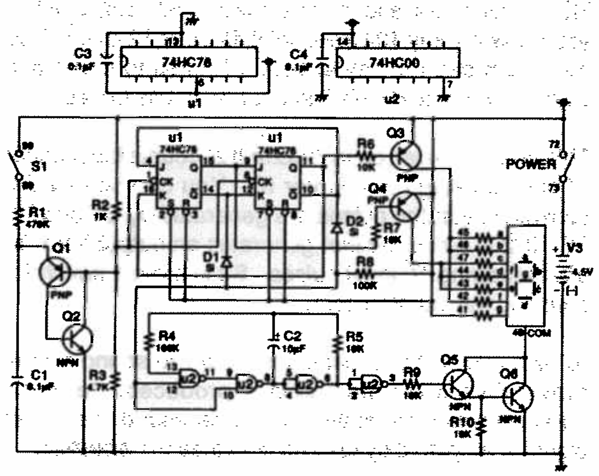
PROJECT 222. CATCH THE EIGHT



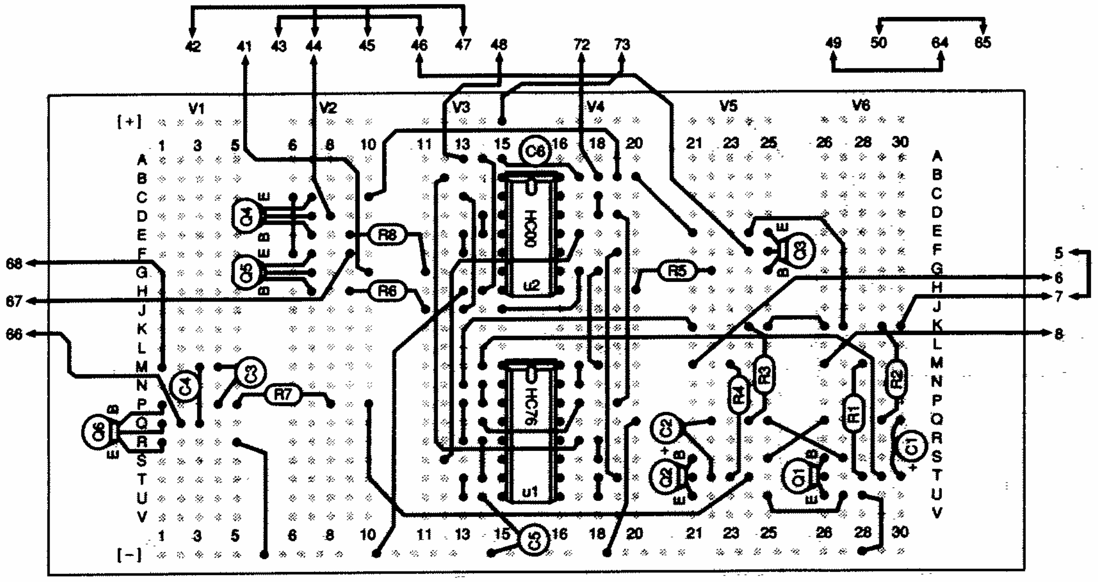
U1	74HC76	Q4	PNP	R3	4.7KΩ	R8	100KΩ	C3	0.1µF
U2	74HC00	Q5	NPN	R4	100KΩ	R9	10KΩ	C4	0.1µF
Q1	PNP	Q6	NPN	R5	10KΩ	R10	10KΩ	D1	Si
Q2	NPN	R1	470KΩ	R6	10KΩ	C1	0.1µF	D2	Si
Q3	PNP	R2	1KΩ	R7	10KΩ	C2	10µF		

Here's another IC and display game to test your wits. In this project, you won't be "shooting" at **LEDs**... instead, you'll be trying to catch an eight!

Once you finish wiring, turn power ON. Then set the slide switch **S1** to ON. On the **LED display** you'll see the upper and lower parts of "8" are flashing alternately. Now carefully note the pattern of flashing. You'll note there's a moment when the display shows nothing - when you think you can anticipate the moment of no display, slide **S1** to OFF. If you do it at exactly the right moment, a complete **8** is "captured" and flash over and over on the **LED display**. If you don't - well, slide **S1** to ON and try again.



PROJECT 223. SOS ALERT

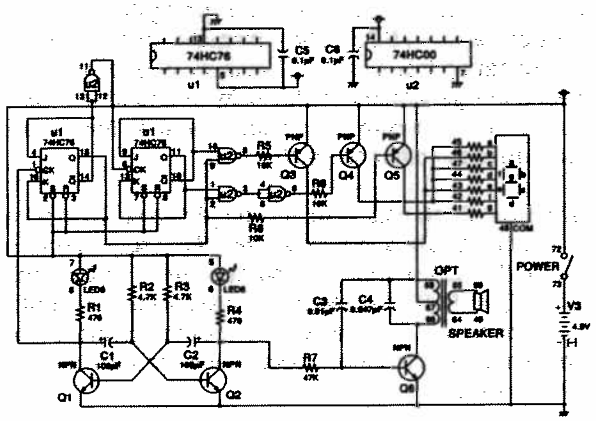


U1	74HC76	Q4	PNP	R3	4.7KΩ	R8	10KΩ	C5	0.1µF
U2	74HC00	Q5	PNP	R4	470Ω	C1	100µF	C6	0.1µF
Q1	NPN	Q6	NPN	R5	10KΩ	C2	100µF		
Q2	NPN	R1	470Ω	R6	10KΩ	C3	0.01µF		
Q3	PNP	R2	4.7KΩ	R7	47KΩ	C4	0.047µF		

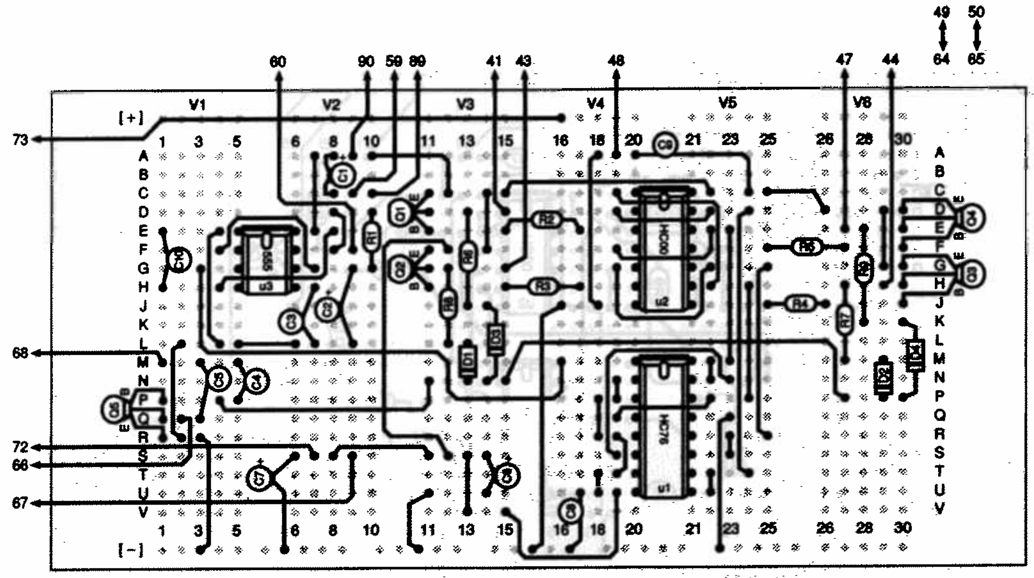
We've already used the ICs in your kit and the **LED display** to create some unusual circuits. This project also uses the ICs and the display to create an automatic SOS alert system.

Turn power ON. You'll hear sound from the **speaker** and see the letters "SOS" on the display. You'll also see **LEDs 5** and **6** light and go off as if they're "taking turns" being lit.

Notice how the display changes. It goes to multivibrator circuit. The project uses both a starting over at "S". It almost seems like the circuit is counting "1,2,3,4 ... 1,2,3,4..." over and over. It turns out that's pretty close to what is actually going on.



PROJECT 224. WHEEL OF FORTUNE



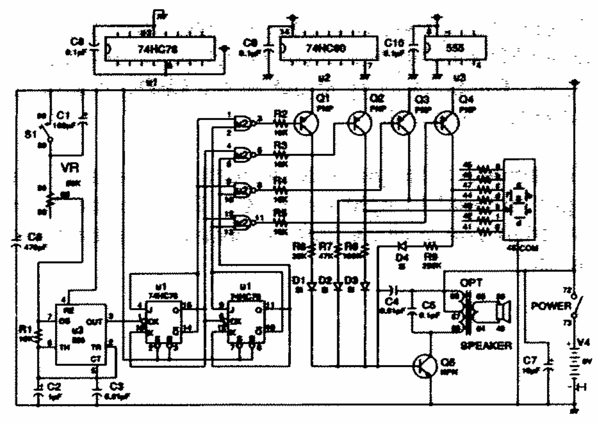
U1	74HC76	Q4	PNP	R5	10KΩ	C2	1µF	C8	0.1µF
U2	74HC00	Q5	NPN	R6	10KΩ	C3	0.01µF	C9	0.1µF
U3	555	R1	10KΩ	R7	47KΩ	C4	0.01µF	C10	0.1µF
Q1	PNP	R2	10KΩ	R8	100KΩ	C5	0.1µF	D1	Si
Q2	PNP	R3	10KΩ	R9	220KΩ	C6	470µF	D2	Si
Q3	PNP	R4	10KΩ	C1	100µF	C7	10µF	D3	Si
								D4	Si

You've probably seen a roulette wheel, or "wheel of fortune" type game in operation. You know how it works ... players try to guess where the wheel stops and they win if they guess right. we couldn't find room in this kit for the real thing, but we've included an electronic version!

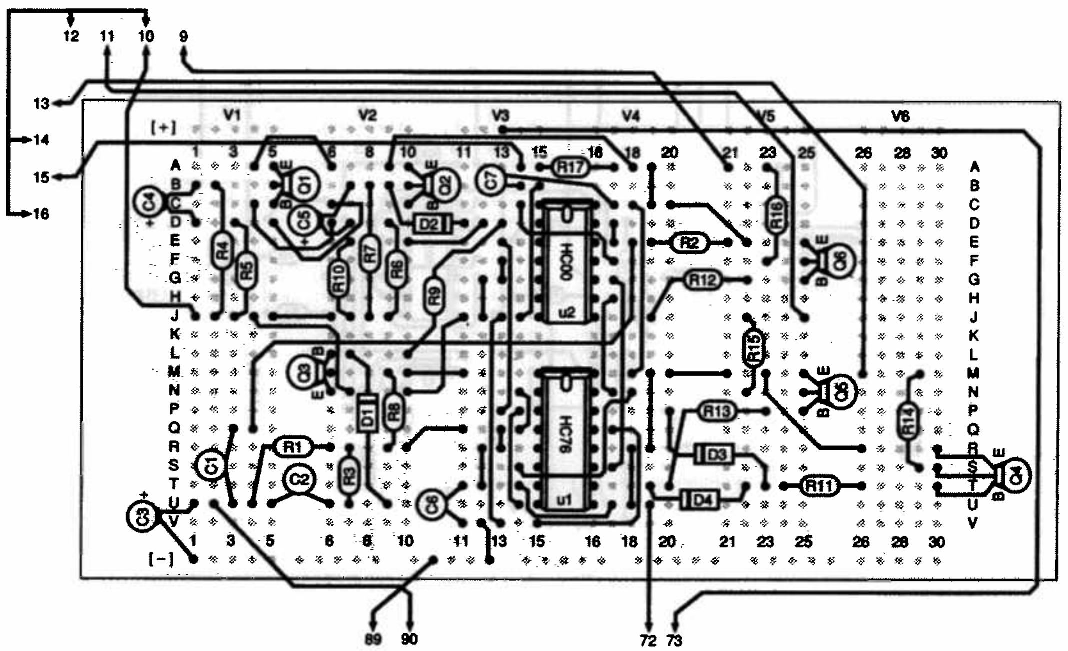
Using this electronic "wheel of fortune" is supersimple. Turn power ON and press **S1** down. You'll notice the lower half of the **LED display** lights up and seem to "spin" around. (It's not really spinning, of course - the different segments are just rapidly blinking on and off, one after the other.) During the "spinning" you'll hear a funny sound from the **speaker**. After a few moments both the "spinning" and sound slows down. Eventually, it stops with just one segment lit and a steady sound coming from the **speaker**.

You'll notice a couple of interesting things about this circuit. Each segment of the display has its own sound. And the speed at which the display "spins" depends upon the **control volume**. Try moving the **control volume** while the "wheel" is "spinning" ... notice how you can make it slow down or speed up.

You can use this project as a game by guessing which segment will be lit when the "wheel" finally stops "spinning." Or you can try to make the "wheel" stop at a certain segment by adjusting the **control volume** while it is still "spinning."



PROJECT 225. LEAPIN' LEDS

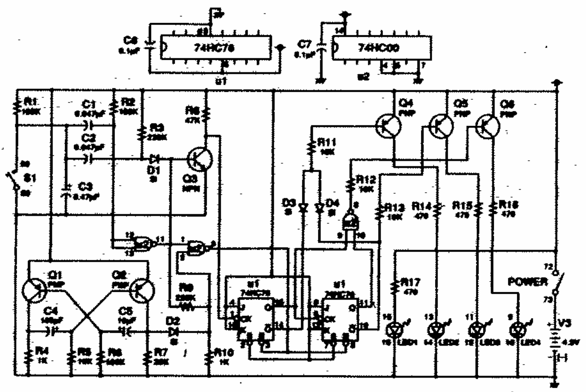


U1	74HC76	Q6	PNP	R7	33KΩ	R14	470Ω	C4	100µF
U2	74HC00	R1	100KΩ	R8	47KΩ	R15	470Ω	C5	10µF
Q1	PNP	R2	100KΩ	R9	220KΩ	R16	470Ω	C6	0.1µF
Q2	PNP	R3	220KΩ	R10	1KΩ	R17	470Ω	C7	0.1µF
Q3	NPN	R4	1KΩ	R11	10KΩ	C1	0.047µF	D1	Si
Q4	PNP	R5	10KΩ	R12	10KΩ	C2	0.047µF	D2	Si
Q5	PNP	R6	100KΩ	R13	10KΩ	C3	0.47µF	D3	Si
								D4	Si

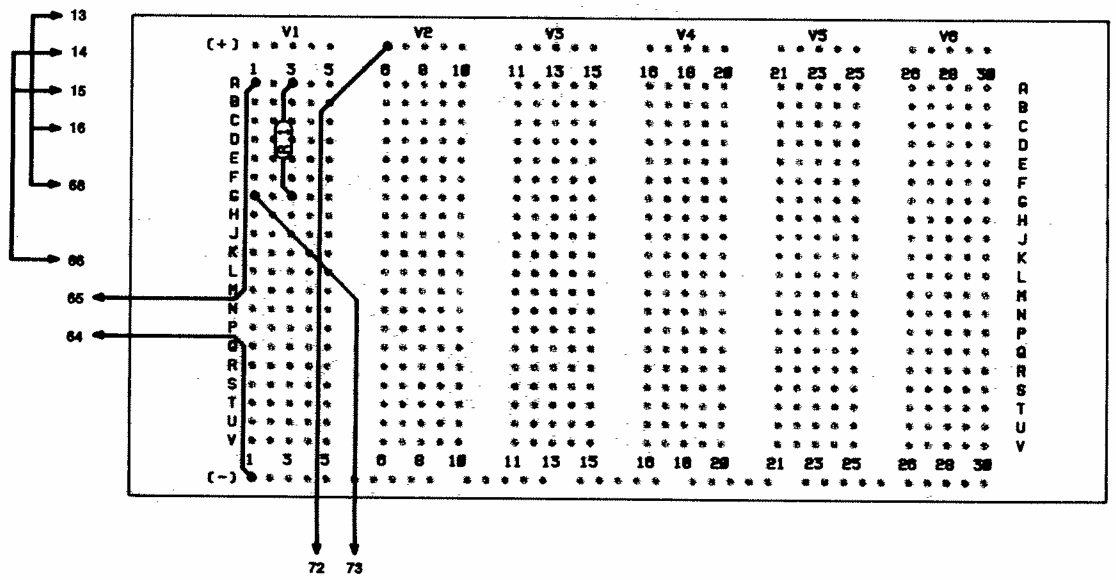
Here's game to see how fast you are on the trigger (or at least the key!). The object is to light **LEDs** 1 through 4 as quickly as you are able or with as few presses of **S1** as you can.

To play, turn power ON. **LED 1** lights. Now press **S1** until **LED 2** lights. But, if you're not lucky, only **LED 1** lights. Continue to try to get all the **LEDs** to light up (**LED 1, 2, 3** and **4**).

The secret of this game is to press **S1** at exactly the right moment to light the **LED**. Timing's been an important part of all the digital circuits we've played with so far.



PROJECT 226. EXPERIMENT OF ELECTROMAGNETIC INDUCTION

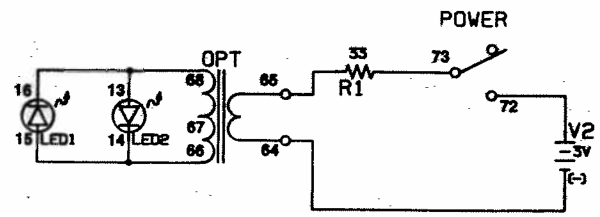


R1 33Ω

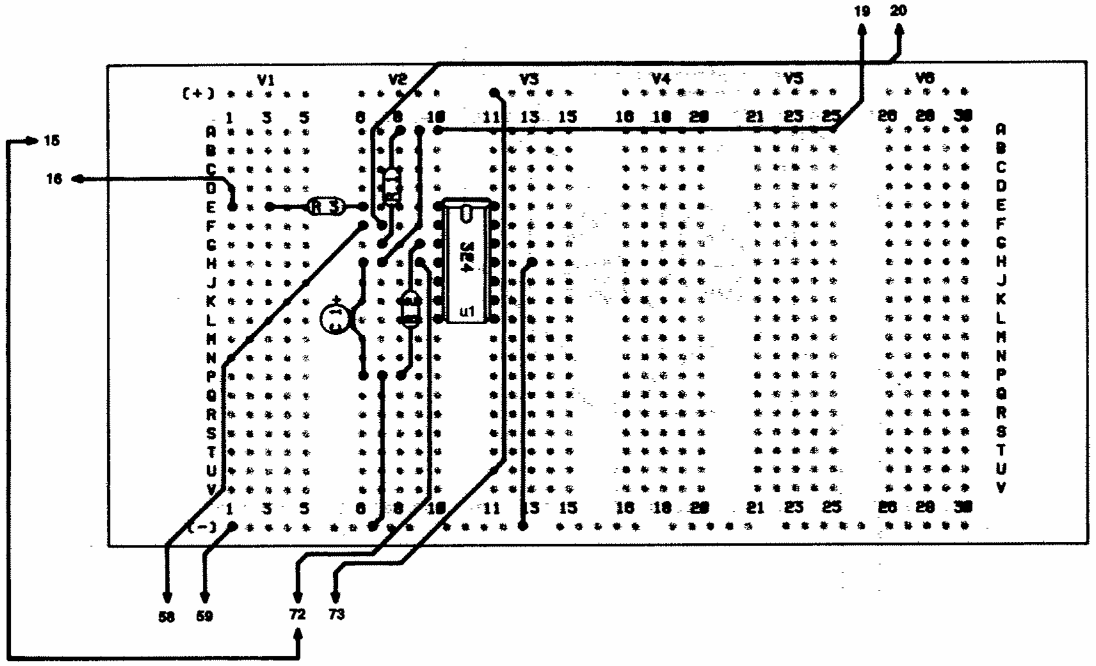
In this project, we're going to find out what electromagnetic induction links, using the battery, transformer and **LEDs**. The schematic shows you see how to set up this project. Electromagnetic induction means the generation of electricity in one of two coils wound on an iron core that occurs only when the current supplied to the other coil is changed.

When you finish setting up this project keep your eyes on the two **LEDs** and turn power ON. Did you notice one of the **LEDs** flash the moment when you turned power ON? That's the electromagnetic induction. But if you keep power ON, the current doesn't change, so no electromagnetic induction occurs and the **LED** doesn't light up either.

Now, watch the two **LEDs** closely again and turn power OFF. Which **LED** flashed this time? Turn the power switch ON and OFF and see how the electromagnetic induction occurs.



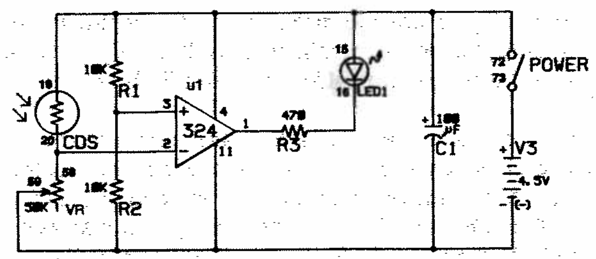
PROJECT 227. ELECTRONIC CANDLE



U1	324	R1	10KΩ	R3	470Ω
		R2	10KΩ	C1	100μF

Blowing out a candle is easy - but have you ever tried blowing out an electric light? You can with this project! This little bit of electronic trickery keeps your friends guessing how you do it.

Turn power ON and carefully adjust the **control volume** to the point where the **LED** is about to turn on. Since this trick works best in a dark room, darken the room and tell your friends you must "light" the electronic candle with - what else? - a flashlight. Shine the flashlight on the **CdS cell**; the **LED** turns on.

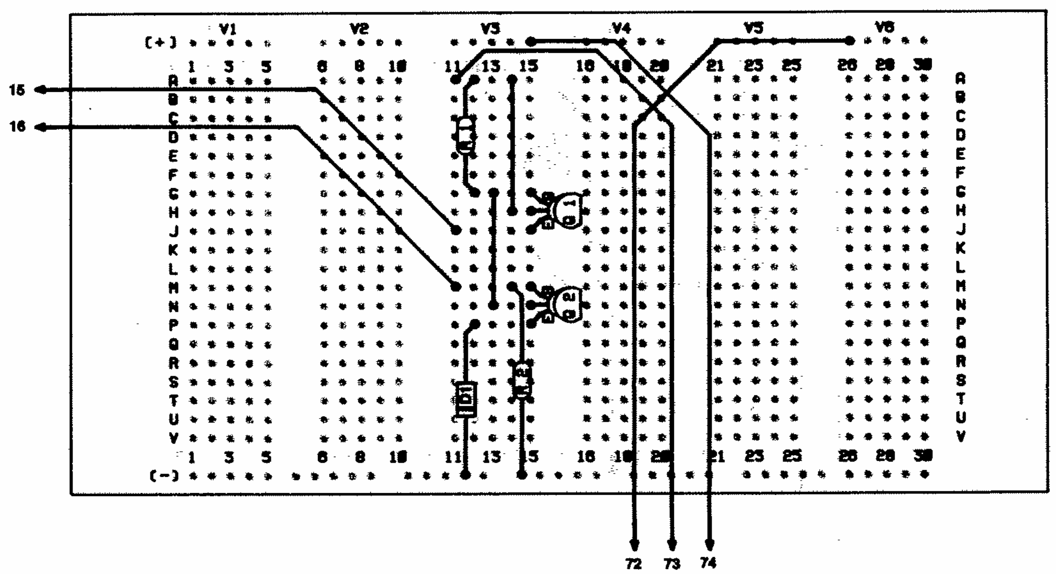


Now you're ready to "blow" your electronic candle out. Cup your hand around the **LED** and blow on it. Slightly move your hand so that you cover up the **CdS cell**. Presto! The **LED** goes out! To "re-light" it, simply shine the flashlight on it again.

Like all good magicians, you'll have to rehearse this trick a few times before performing it for others. It's important that you only try this trick in a dark room. And you'll find that adjusting the **control volume** to the point to let you do this trick takes some practice.

Have fun with this project - and please don't give away the secret of how you do it!

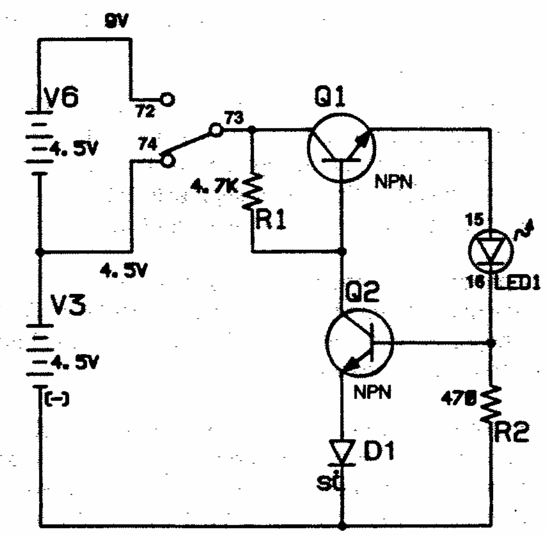
PROJECT 228. CONSTANT CURRENT CIRCUIT



Q1	NPN	R1	4.7KΩ	D1	Si
Q2	NPN	R2	470Ω		

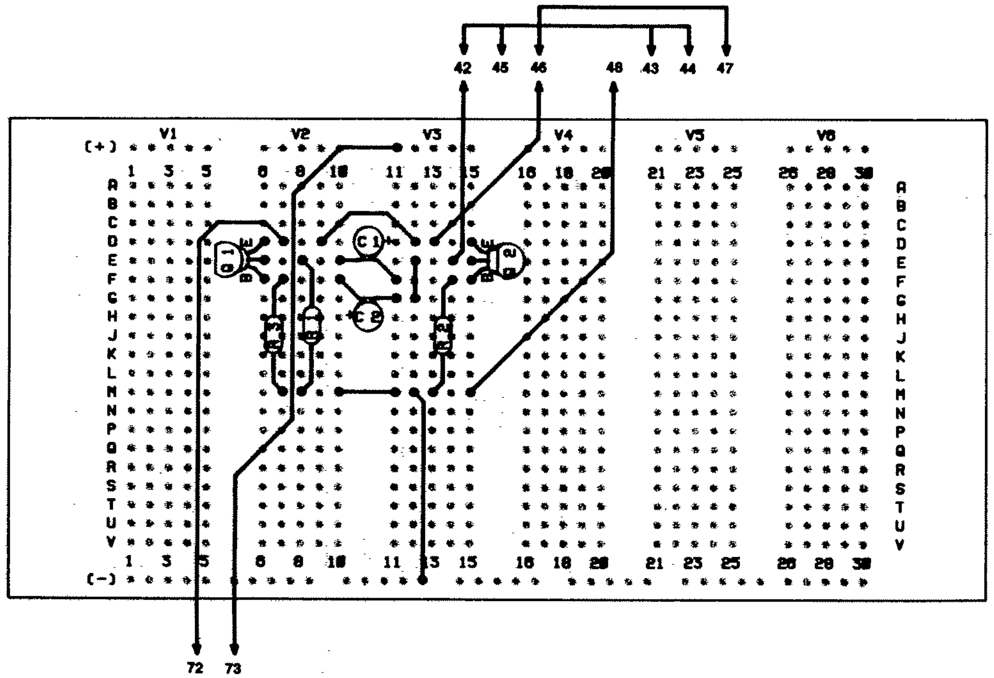
In the world of electric engineering, everybody knows that when the voltage is raised, the current flow increases proportionally. This is what they call Ohm's law. But transistors make it possible for us to build a circuit that can maintain a constant current flow even if the voltage is increased or decreased.

In this project, we're going to make this constant current circuit using two transistors. The schematic shows you that the current flowing through the **LED** can be controlled with Q1, and when the voltage becomes higher and the current tends to increase, Q2 detects it and holds down the increases of the Q1 current.



Now, let's see how this circuit works. Change the **select switch** setting and see if the **LED** brightness shows any difference between 4.5 V and 9.0 V. It doesn't show any difference. Of course you know why! Yes, it's because a constant current is supplied to the **LED**.

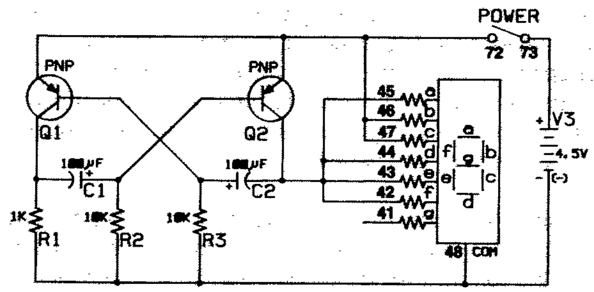
PROJECT 229. A PHONY COUNTER



Q1	PNP	R1	1KΩ	C1	100µF
Q2	PNP	R2	10KΩ	C2	100µF
		R3	10KΩ		

Here's chance to make a multivibrator circuit imitate a counter.

Turn power ON when you finish the wiring. You'll see the numbers **0** and **1** appear in turn on the **LED display**. You'll notice that two of the segments on the **LED display** stay lit all the time. Four other segments light up and go off according to the output of the multivibrator.

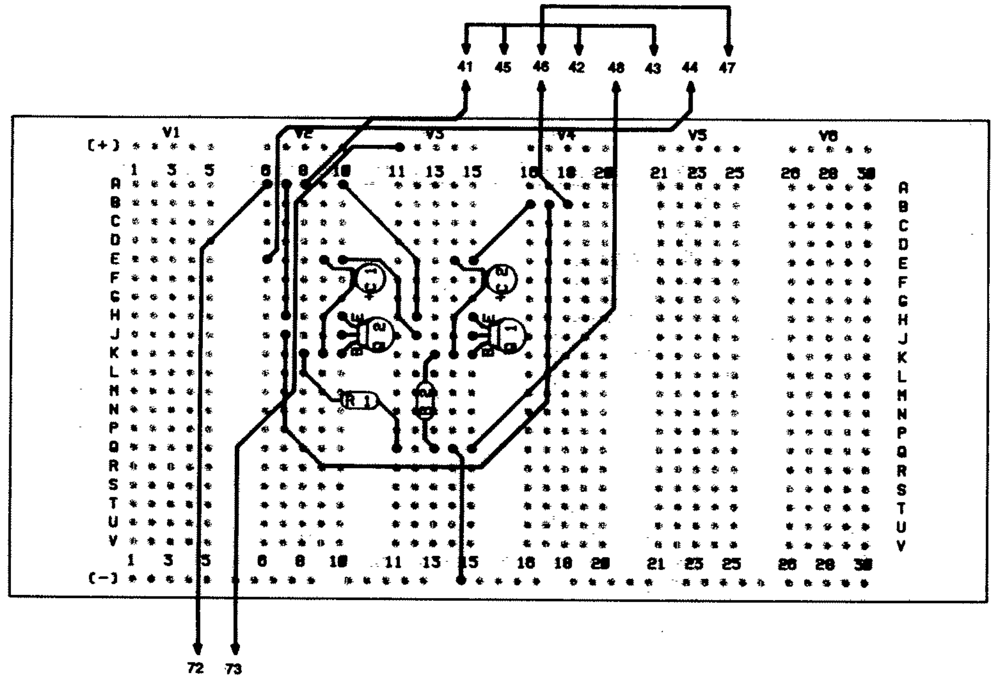


This project produces an effect similar to project 189. But are both projects the same? Think about it a minute (you might want to compare schematics for them) and then check your answer below.

They're not the same. Project 189 is a digital circuit that actually counts. This project does not count the pulses from the multivibrator - the circuit only switches the display back and forth each time the multivibrator changes states. It might look like it's counting, but it's not.

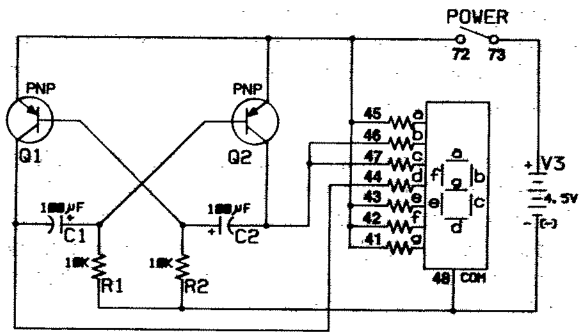
Before going on to the next project, see if you can produce some different indications on the **LED display**. Keep notes of what you managed to come up with - because you'll soon see how to produce different indications.

PROJECT 230. ALPHABET FLASHER



Q1	PNP	R1	10KΩ	C1	100µF
Q2	PNP	R2	10KΩ	C2	100µF

Did you figure ways to make the digital display in our last project indicate something other than 0 or 1 ? This circuit flashes the capital letters **A** and **E** at you.

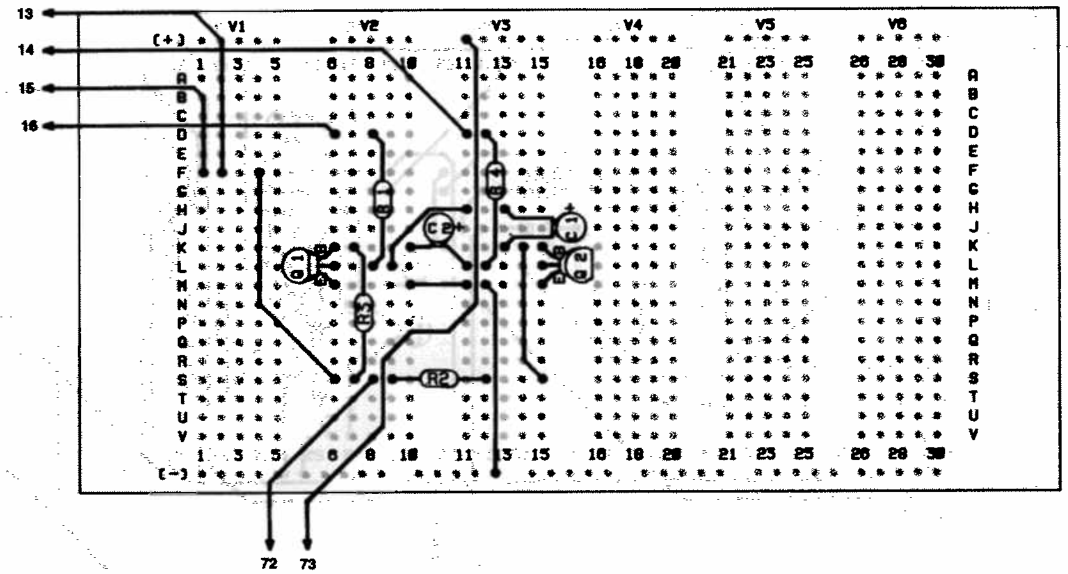


This circuit is the same astable multivibrator used in our last project. By altering the connections to the display, **A** and **E** are flashed over and over instead of 0 and 1. But **A** and **E** aren't the only letters that you can show on the **LED display** - **C, F, H, J, L, P** and **U** can also be displayed!

You can dream up some other interesting circuits the multivibrator by altering the values of ... C'mon, by now we don't have to tell you which parts to change! (We're right, aren't we?)

You can dream up some other interesting circuits using a multivibrator on your own.

PROJECT 231. WINKING LEDS



Q1	NPN		R1	470Ω		R3	10KΩ		C1	100µF
Q2	NPN		R2	10KΩ		R4	470Ω		C2	100µF

If you have been making up some of your own circuits, chances are you've come up with something similar to this project. Multivibrators and **LEDs** seem to naturally go together.

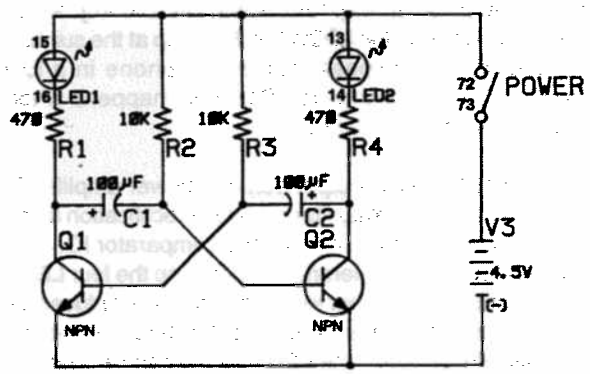
The schematic lets you see how this project works. Do you think that the two **LEDs** light up and go off together or do they take turns lighting? Try to figure out what happens before you build the circuit. (And, if you need some help, look back at the notes you made for other multivibrator circuits ...)

You can vary the speed of the pulses from the multivibrator by using different values for the 100µF capacitors and the 10K resistor. You should be able to predict what happens as you change values by reviewing the notes you've been keeping.

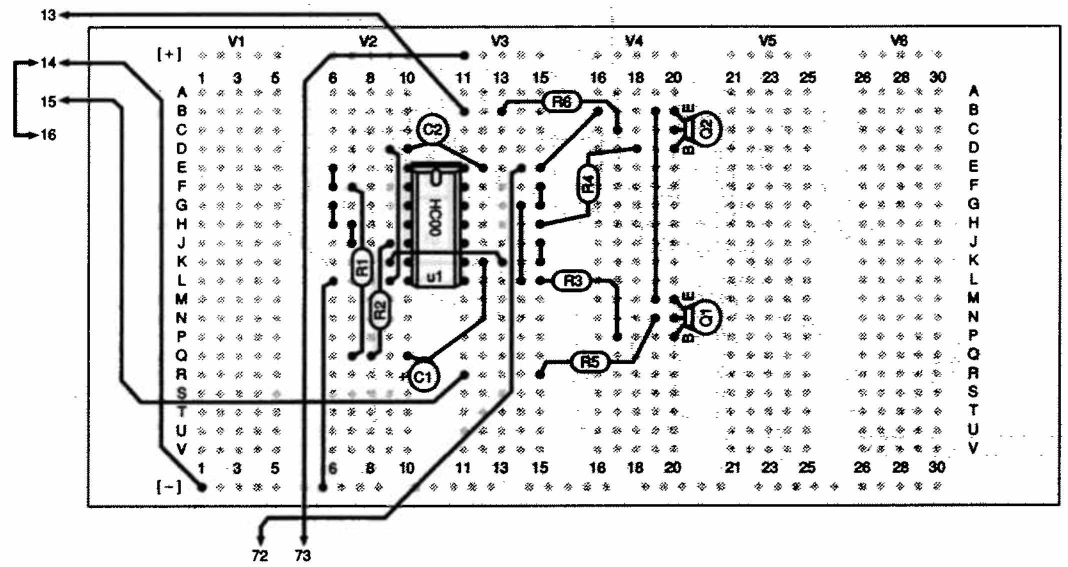
You can use this project in many of your own circuit creations. Try adding some method of changing the operation of the multivibrator by changing the setting of the **control volume**. You might also want to add the **speaker** in some way so you can hear and see the multivibrator operating. Keep schematics of any circuits you create ...

One final thing - what kind of multivibrator circuit is this project? (Try to answer without peeking at your notes or this Manual.)

It's an astable multivibrator.



PROJECT 232. WINKING LEDS II



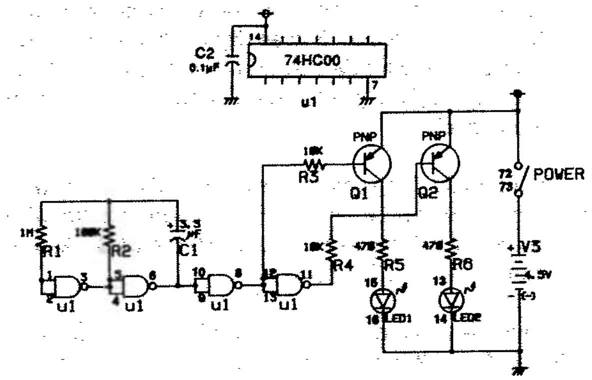
U1	74HC00		R1	1MΩ		R4	10KΩ		C1	3.3µF
Q1	PNP		R2	100KΩ		R5	470Ω		C2	0.1µF
Q2	PNP		R3	10KΩ		R6	470Ω			

Here's a circuit similar to the one you built in our last project, but this time the multivibrator circuit is made using the C-MOS IC. If you're a little bit rusty about how C-MOS multivibrators work, look back at project 166 and the notes you made for it.

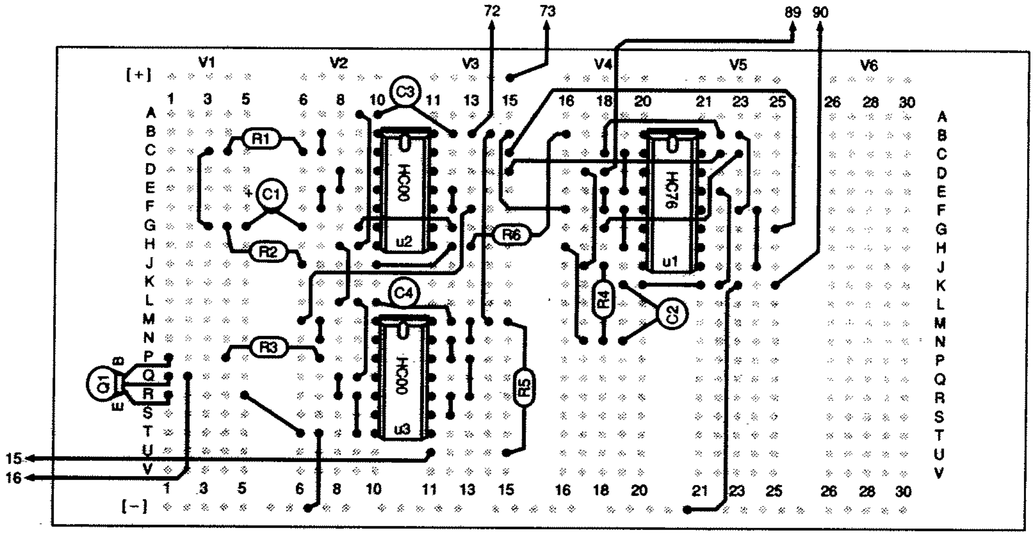
When you turn power ON, you'll see both **LED 1** and **LED 2** take turns going on and off. You can change the speed of the pulses by substituting different values for the 3.3µF capacitors. You can use this C-MOS multivibrator in place of the transistor multivibrator in such projects as 229, 230 and 231.

C-MOS multivibrators are becoming more widely used today in place of transistor multivibrators. Can you think of some reasons why? Think about it for a minute and then turn this manual upside down.

C-MOS multivibrators take up much less space than transistor multivibrators. C-MOS ICs also use less current than similar transistor arrangements.



PROJECT 233. DELAYED TIMER II



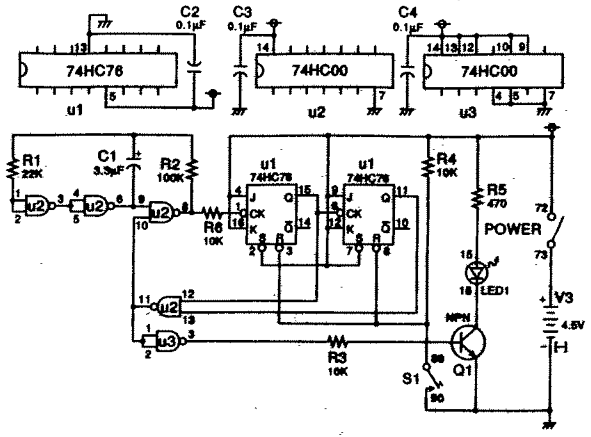
U1	74HC76	R1	22KΩ	R4	10KΩ	C1	3.3μF
U2	74HC00	R2	100KΩ	R5	470Ω	C2	0.1μF
U3	74HC00	R3	10KΩ	R6	10KΩ	C3	0.1μF
Q1	NPN					C4	0.1μF

We've built some timer circuits before (like project 107) but this one is different - it uses digital electronics. The other timers have made use of the discharging rate of capacitors. Can you guess how this project works from the schematic?

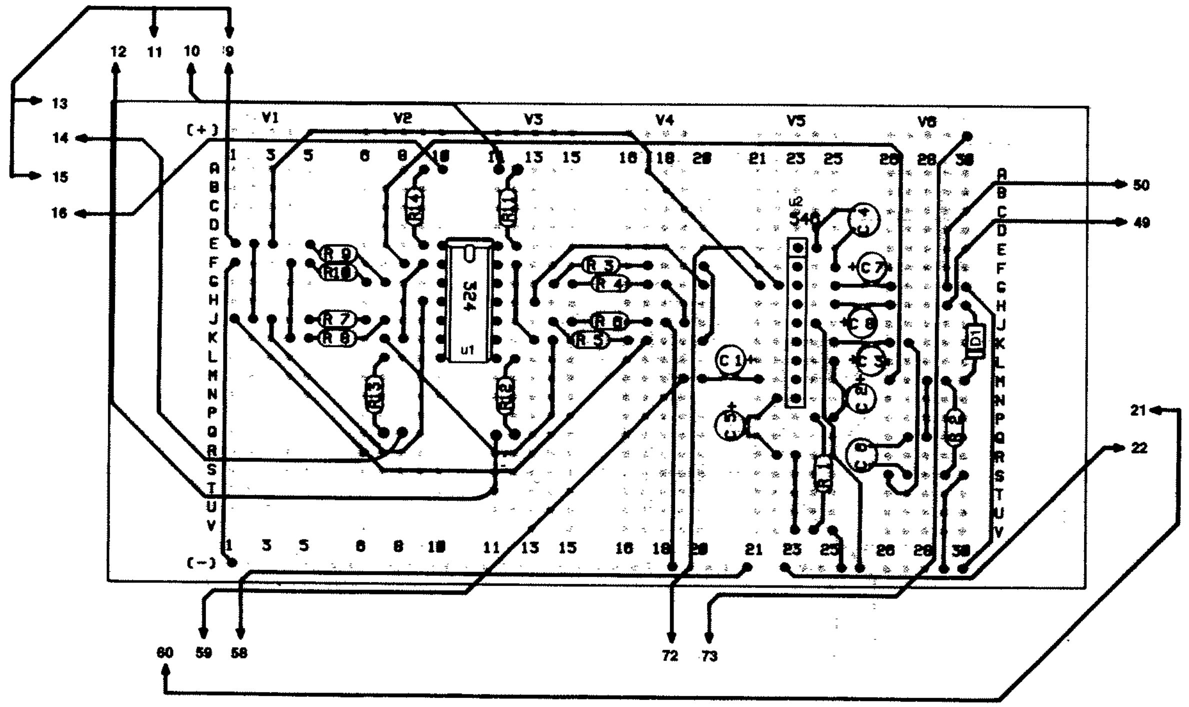
After you build this project, turn power ON. The **LED** might or might not be on. Press **S1** once and the **LED** should go out. Release **S1** and watch the **LED**. After a few moments, the **LED** lights up.

You can see this project uses a NAND multivibrator and a J-K flip-flop. When you press **S1**, the counter is "reset" and the **LED** is turned off. When you release **S1**, pulses from the mulfivibrator are fed into the clock input of the first J-K flip-flop. After three pulses are input, the Q outputs of both flip-flops are 1. This 1 output goes to a NAND gate, producing a 0 output. The 0 output goes to another NAND gate, where it produces a 1 output. This enables the transistor to operate, lighting up the **LED**. You'll also notice that the 0 output also goes to the NAND multivibrator, where it stops the multivibrator from operating.

Try altering the operation of the multivibrator and see what effect it has on the timer. Can you think of any advantages this type of timer would have compared to circuits like project 107?



PROJECT 234. VOICE LEVEL METER

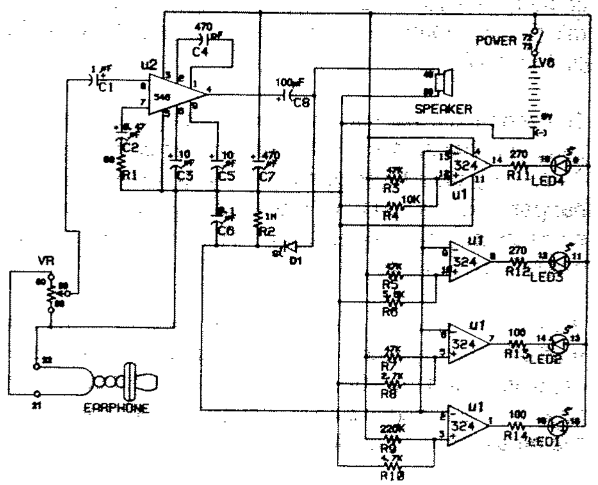


U1	324	R4	10KΩ	R9	220KΩ	R14	100Ω	C5	10μF
U2	546	R5	47KΩ	R10	4.7KΩ	C1	1μF	C6	0.1μF
R1	68Ω	R6	5.6KΩ	R11	270Ω	C2	0.47μF	C7	470μF
R2	1MΩ	R7	47KΩ	R12	270Ω	C3	10μF	C8	100μF
R3	47KΩ	R8	2.7KΩ	R13	100Ω	C4	470pF	D1	Si

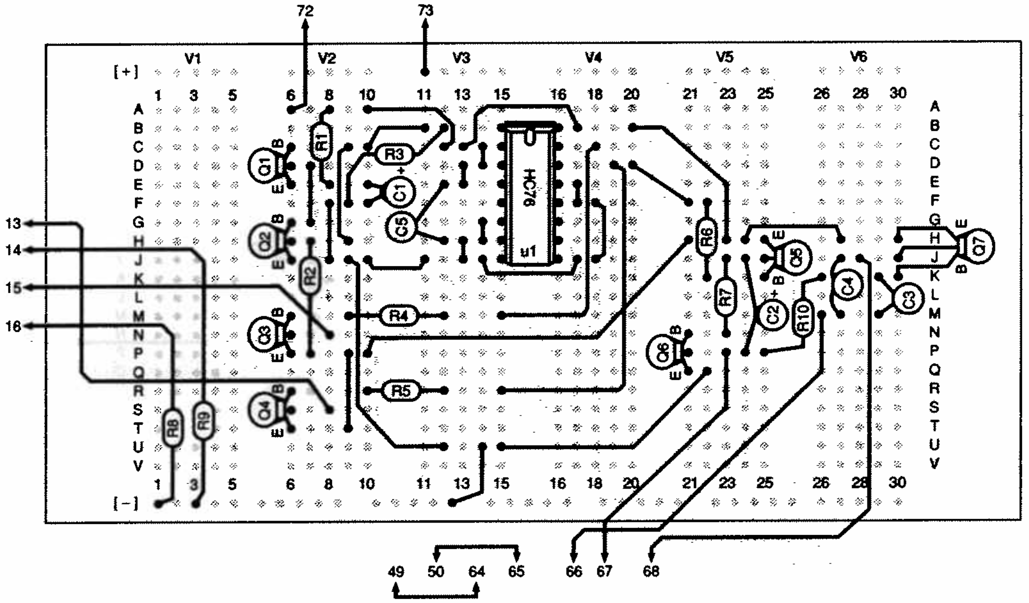
Here's an **LED** level meter that can tell you the loudness of your voice. Use the earphone as a microphone and speak into it, or input any other audio signal.

When you finish wiring, turn power on and adjust the **control volume** so the **LEDs** 1 to 4 light up at the suitable voice level. Now speak into the earphone in a low voice..only **LED 1** lights up. See what happens when you yell into the earphone.

In this circuit, the AC output from the power amplifier is converted to a DC voltage by half-wave rectification using the Si diode. Then it is input to the comparator IC. The comparator has four reference voltages, so the four **LEDs** light up one after another when the voltage produced by half-wave rectifier goes beyond each reference voltage. The ON/OFF speed of the **LED** flashing can be changed by changing the R2 and C6. Try it!



PROJECT 235. CROSSING SIGNAL



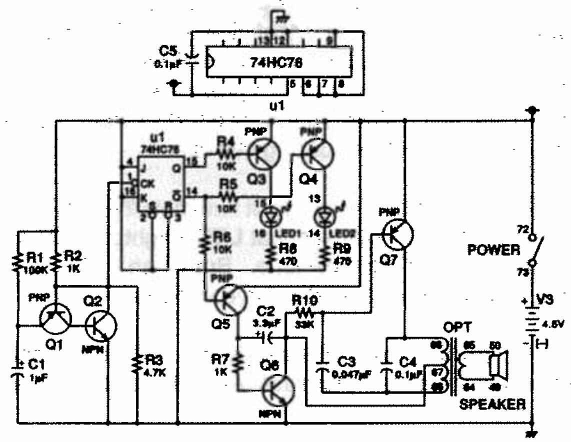
U1	74HC76	Q5	PNP	R3	4.7KΩ	R8	470Ω	C3	0.047µF
Q1	PNP	Q6	NPN	R4	10KΩ	R9	470Ω	C4	0.1µF
Q2	NPN	Q7	PNP	R5	10KΩ	R10	33KΩ	C5	0.1µF
Q3	PNP	R1	100KΩ	R6	10KΩ	C1	1µF		
Q4	PNP	R2	1KΩ	R7	1KΩ	C2	3.3µF		

This project is an all-electronic version of the signal you've seen many times at train crossings. You can also use this circuit as an alarm or "surprise" project as well.

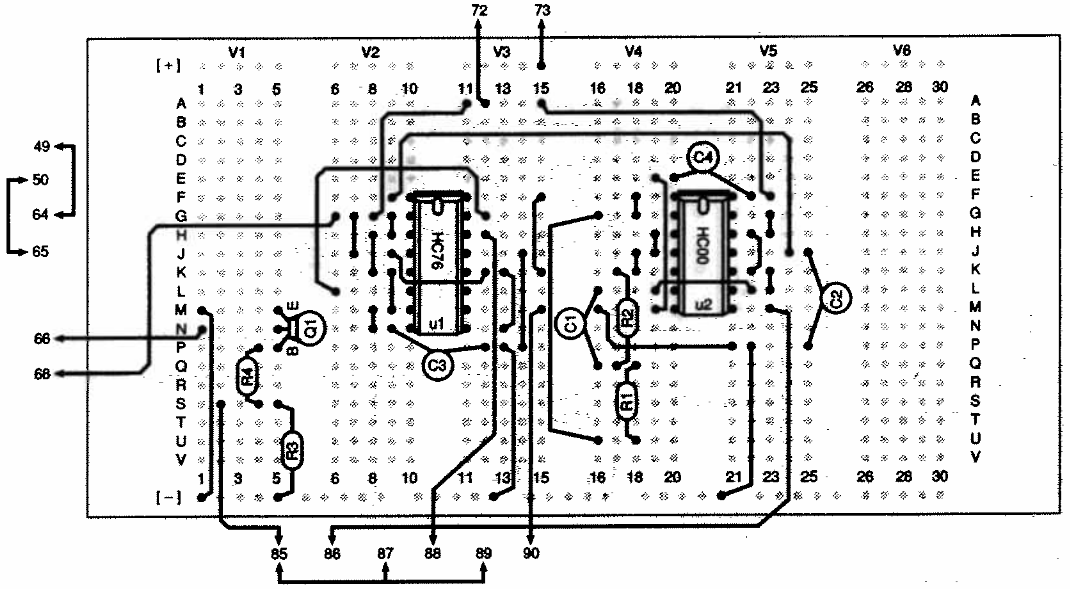
You can figure out how this project works by looking at the schematic. **LED 1** and **LED 2** are connected to the outputs of a J-K flip-flop, and they go on off as the outputs switch between 0 and 1. An audio oscillator supplies sound to the **speaker**.

The circuit that generates the clock signal for the J-K flip-flop is a Schmitt trigger. A Schmitt trigger is a bistable pulse generator (bistable means the circuit has two steady operating conditions) that takes a slowly changing signal and converts it into one with sharp changes (much like the on-off output of an astable multivibrator).

Turn power ON and you'll hear a sound of train crossing together with flashing **LEDs**.



PROJECT 236. OCTAVE GENERATOR



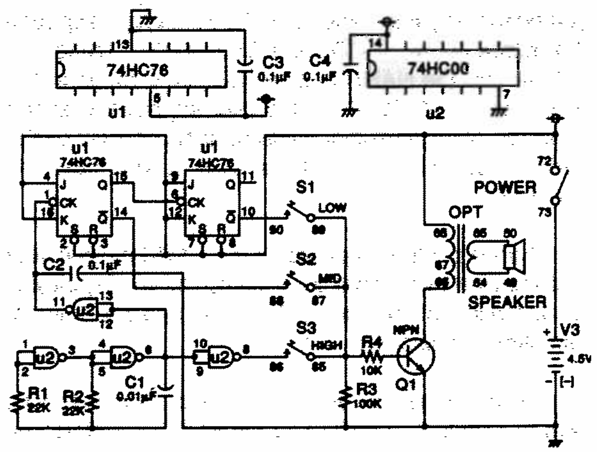
U1	74HC76	R1	22KΩ	R4	10KΩ	C3	0.1µF
U2	74HC00	R2	22KΩ	C1	0.01µF	C4	0.1µF
Q1	NPN	R3	100KΩ	C2	0.1µF		

Have you ever heard some electronic music or read something about it? You probably have - electronics is becoming very common in music today. Electronic circuits are being used to imitate the sound of various musical instruments or to create entirely new sounds. Here's a project that lets you see how some electronic musical instruments work.

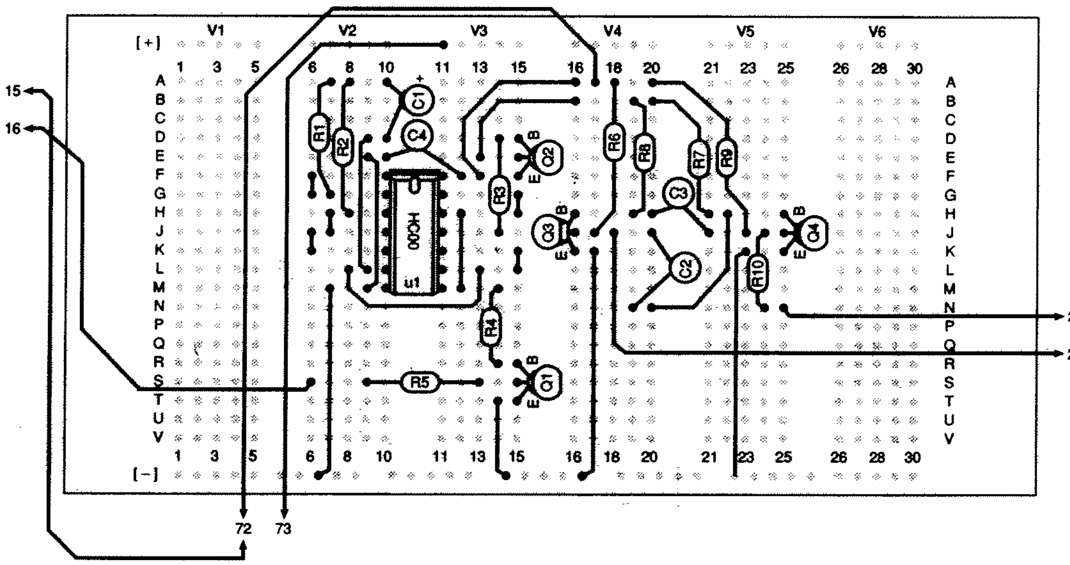
You can see from the schematic that this project is made up of a NAND multivibrator and a J-K flip-flop divider circuit. Using **S1**, **S2** and **S3**, your "sample" the output from the multivibrator at three different points: from the multivibrator itself, after it's been divided once and after it's been divided twice. Turn power ON and press **S1**, **S2** and **S3** in turn. You'll notice that each time the multivibrator's output has been divided the sound you hear gets lower.

You can further change the sound you hear by substituting a different capacitor for the 0.01µF one used in the NAND multivibrator.

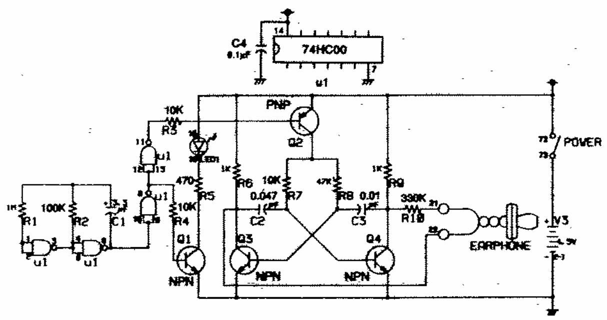
More complex electronic musical devices make use of several different multivibrator and divider circuits. The output from the multivibrators and dividers are combined in several different ways to produce unusual musical effects. Can you think of any other circuits you've played with that could be used to produce music?



PROJECT 237. BUZZIN' LED



U1	74HC00	Q4	NPN	R4	10KΩ	R8	47KΩ	C1	3.3μF
Q1	NPN	R1	1MΩ	R5	470Ω	R9	1KΩ	C2	0.047μF
Q2	PNP	R2	100KΩ	R6	1KΩ	R10	330KΩ	C3	0.01μF
Q3	NPN	R3	10KΩ	R7	10KΩ			C4	0.1μF



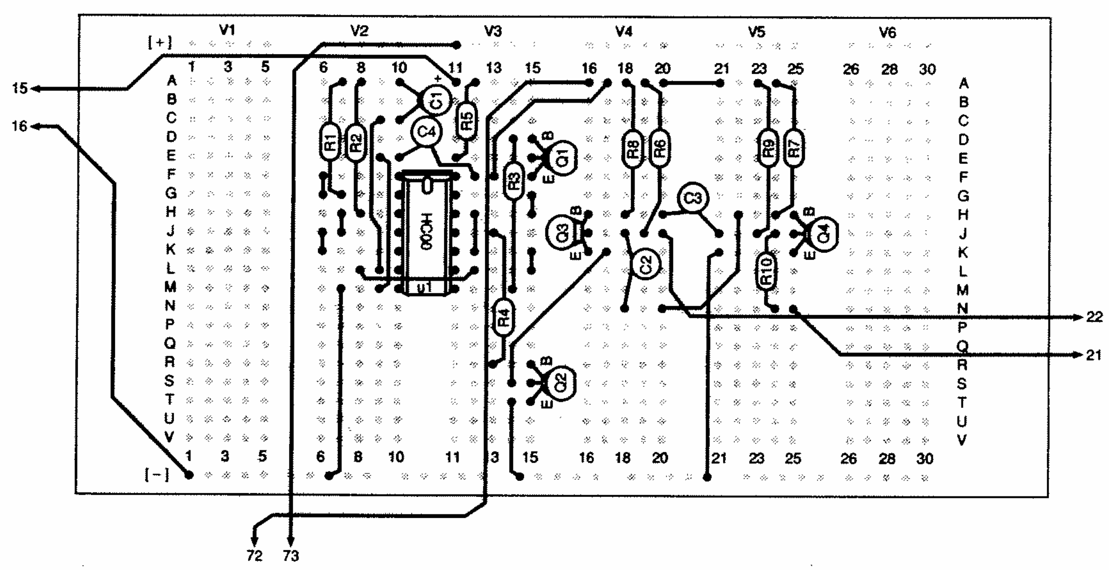
Here's another circuit making use of both transistor and NAND multivibrators. Together they light up **LED 1** at the same time you hear a sound through the earphone.

When you finish wiring, connect the earphone to terminals **21** and **22** and turn power ON. You'll hear a pulse in the earphone each time the **LED** lights up. Can you guess why this happens? Here's a clue - suppose the output of the NAND multivibrator is 0. Trace that output from the NAND multivibrator on through to the transistor multivibrator. Do you suppose that the operation of the transistor multivibrator is affected by the NAND multivibrator? (If the answer is yes, how is it affected?)

Try change a 3.3μF capacitor in the NAND multivibrator and see what effect it has on circuit operation. Try playing around with the transistor multivibrator and see how you can alter its operation.

It's possible to use the **speaker** instead of the earphone with this project. You'll need the PNP transistors, the output transformer and maybe a resistor or two. Try adding the **speaker** - be sure to make a note of the circuit you finally come up with.

PROJECT 238. SON OF BUZZIN' LED



U1	74HC00	Q4	NPN	R4	10KΩ	R8	47KΩ	C1	3.3μF
Q1	PNP	R1	1MΩ	R5	470Ω	R9	1KΩ	C2	0.047μF
Q2	NPN	R2	100KΩ	R6	1KΩ	R10	330KΩ	C3	0.01μF
Q3	NPN	R3	10KΩ	R7	10KΩ			C4	0.1μF

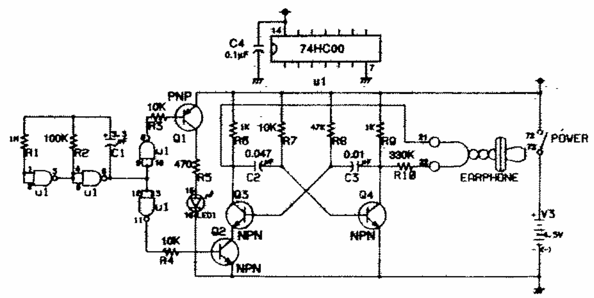
Carefully compare the schematic for this project with the schematic for the last project. They're alike in many way's, but there's an important difference. Can you spot what it is? Better still, can you tell what effect this difference has on the way this project works compared to the last one? Make your best effort to answer those two questions before building this circuits.

Connect the earphone to output terminals **21** and **22** and turn power ON. You'll see that **LED 1** lights up, but you hear nothing in the earphone. But when **LED 1** goes out, then you hear sound in the earphone!

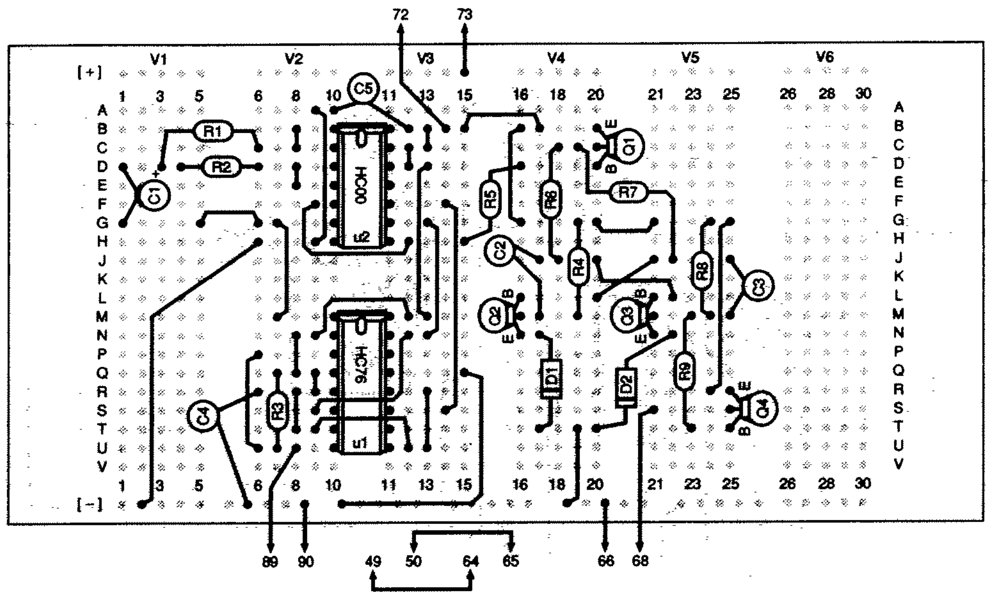
Try to figure out why this happened. When you think you have the answer, turn this Manual upside down and find out!

When the output of the NAND multivibrator is 1, the remaining of two NAND gates output becomes 0, so the PNP transistor turns ON and current can flow through **LED 1** to light up but the transistor multivibrator won't work because the NPN transistor is OFF thus the transistor on the left side of multivibrator is OFF.

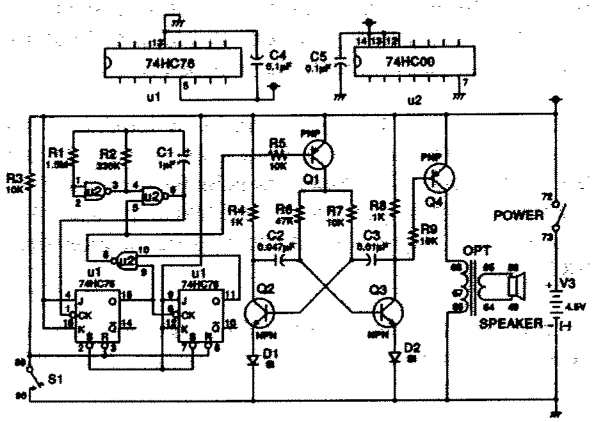
When the output of the NAND multivibrator is 0, **LED 1** won't light but a signal is applied to turn the NPN transistor on. The multivibrator can work, and you can hear the sound in your earphone.



PROJECT 239. SOUND OUT TIMER



U1	74HC76	Q4	PNP	R5	10KΩ	C1	1μF	D1	Si		
U2	74HC00	R1	1.5MΩ	R6	47KΩ	C2	0.047μF	D2	Si		
Q1	PNP	R2	330KΩ	R7	10KΩ	C3	0.01μF				
Q2	NPN	R3	10KΩ	R8	1KΩ	C4	0.1μF				
Q3	NPN	R4	1KΩ	R9	10KΩ	C5	0.1μF				



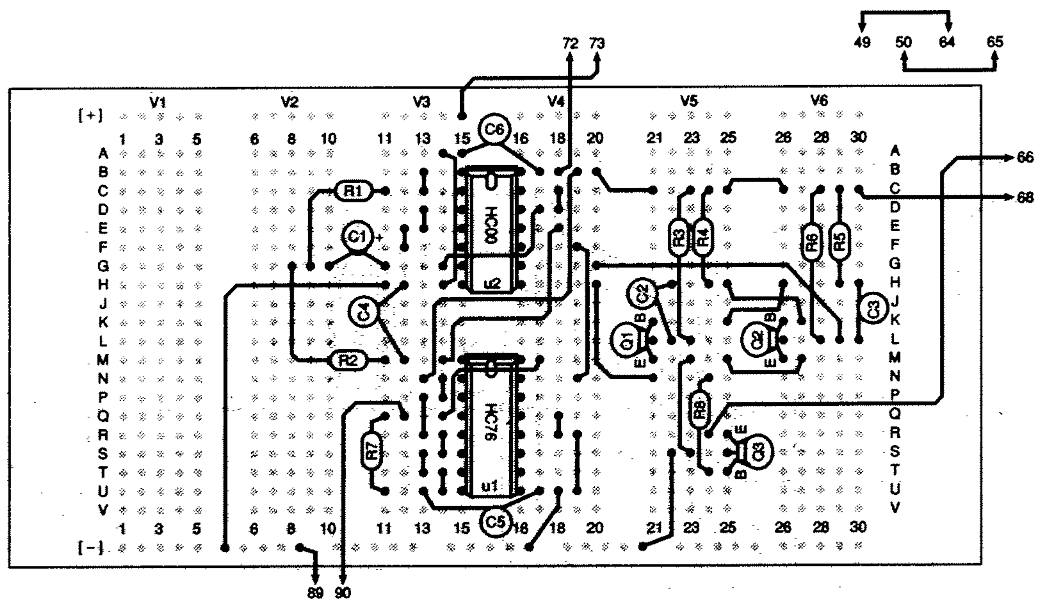
Look closely at the schematic for this project notice anything familiar about it? If not, check back to project 185. This circuit uses a divide-by-4 counter just like the project. That should give you a BIG hint how this circuit works before you build it!

Turn for power ON when you finish wiring. You'll hear a sound from the **speaker**. Now press and release **S1**. The sound from the **speaker** stops ... only to resume a short time later!

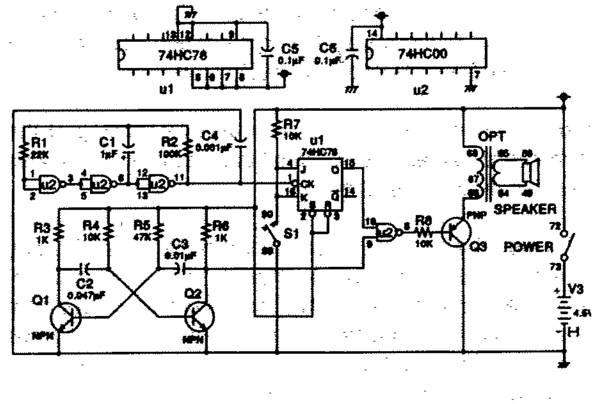
You can easily spot the astable multivibrator in this project. You might have also spotted the multivibrator made up of NAND gates (if you didn't, take a look back at project 166). Have any idea how these two multivibrators help this circuit operate? Here's a hint: when the Q output of both J-K flip-flops is 1, the NAND multivibrator stops working. And you can also see that the Q outputs of both J-K flip-flops also leads back to the transistor multivibrator through a NAND gate.

Try changing a 1μF capacitor in the NAND multivibrator circuit. What effect does this have on circuit operation? Did you except this to happen?

PROJECT 240. SOUND STOP



U1	74HC76	Q3	PNP	R4	10KΩ	R8	10KΩ	C4	0.001μF		
U2	74HC00	R1	22KΩ	R5	47KΩ	C1	1μF	C5	0.1μF		
Q1	NPN	R2	100KΩ	R6	1KΩ	C2	0.047μF	C6	0.1μF		
Q2	NPN	R3	1KΩ	R7	10KΩ	C3	0.01μF				



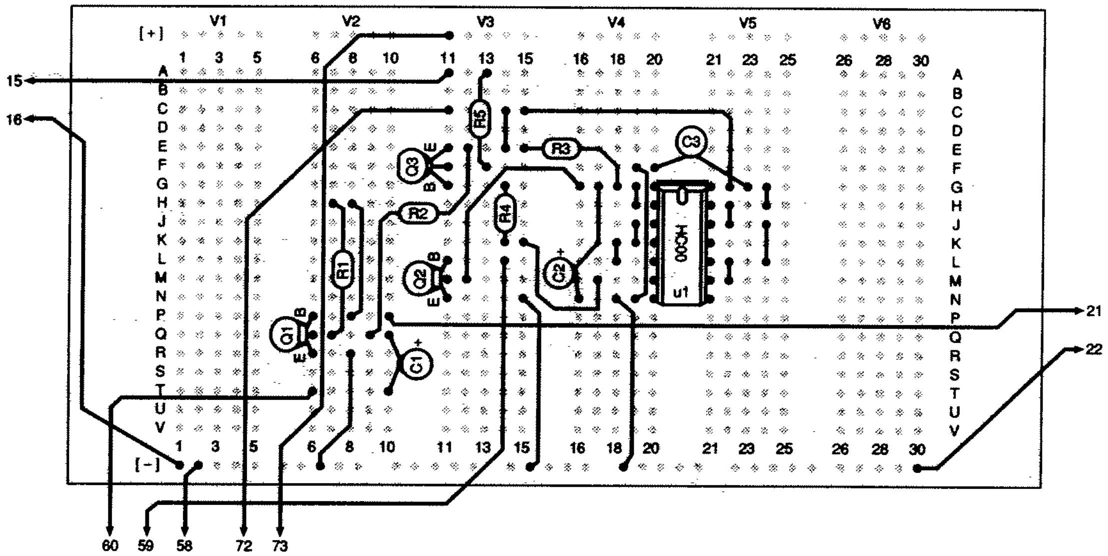
Here's another circuit that lets you see how well you can anticipate whether a digital circuit is 0 or 1. If you guess right, you can turn off the sound from the **speaker**.

You'll notice that this project uses two multivibrators. The transistor multivibrator produces the pulses you hear from the **speaker**. The J-K flip-flop plays an important part in the circuit ... can you guess why? (Try reviewing your notes.) You change the output of the J-K flip-flop by pressing **S1**. If the output at Q is 1, the sound does not stop. If the output is 0, the sound stops.

To play this game, turn for power ON and you'll hear a sound from the **speaker**. Press **S1**. Does the sound stop? If not, try pressing **S1** again and again until you finally manage to turn off the sound.

After you play the game a few times, you might suspect that there is some sort of pattern to pressing **S1** at the right moment to stop the sound. Keep careful notes of when you press **S1** and what happens - try using a clock or watch with a second hand to make your notes as accurate as possible. Is there really a pattern?

PROJECT 241. BIG MOUTH!



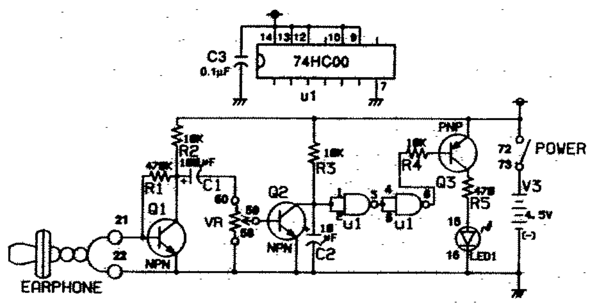
U1	74HC00	Q1	NPN	R1	470KΩ	R4	10KΩ	C1	100μF
		Q2	NPN	R2	10KΩ	R5	470Ω	C2	10μF
		Q3	PNP	R3	10KΩ			C3	0.1μF

Know someone's who's a big mouth? (Or have you been accused of being one?) This project lets you and your friends see who's got the loudest voice.

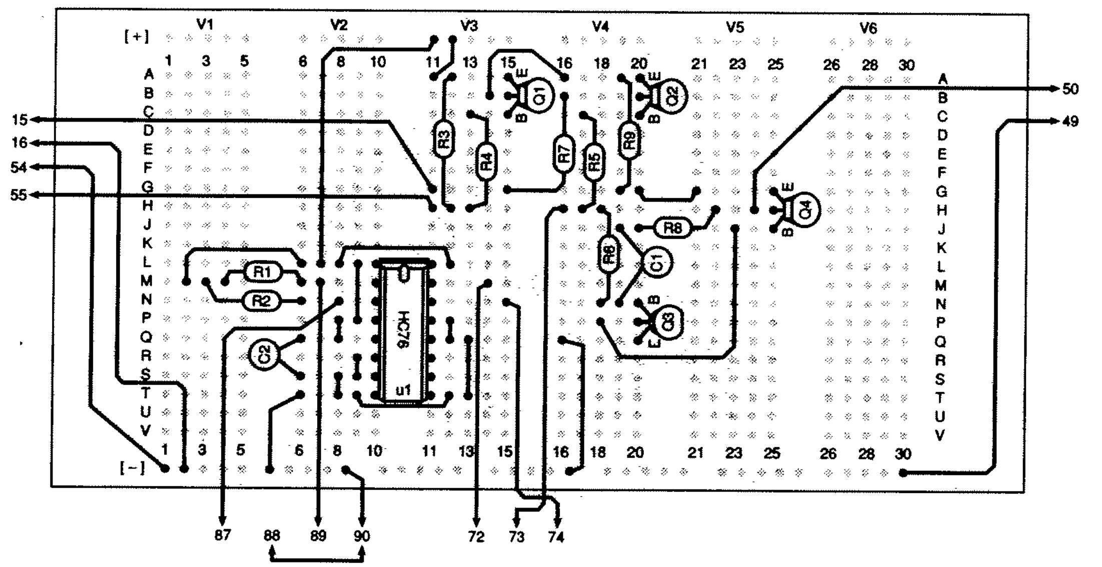
You can see how this project works by looking at the schematic. When you yell into the earphone your voices creates electrical energy by ...

The electrical energy from the earphone is amplified by the two-transistor circuit. You'll notice the **control volume** ... that lets you control how much the signal from the earphone is amplified. Next come two NAND gates in series. They control whether or not **LED 1** lights. (Just to keep in practice, trace how the 0's and 1's change from input to output.)

To play this game, turn power ON and set the **control volume** to the middle of its range. Yell into the earphone and watch **LED 1**. It probably lights. Try turning the **control volume** counterclockwise to make it more difficult to light **LED 1**. (Try adjusting it just a tiny bit each time.) See how far you can reduce the gain of the amplifier and still light the **LED 1**.



PROJECT 242. LIGHT OR SOUND



U1	74HC76	Q3	NPN	R3	10KΩ	R7	470Ω	C1	0.047μF
Q1	PNP	Q4	PNP	R4	10KΩ	R8	270Ω	C2	0.1μF
Q2	PNP	R1	10KΩ	R5	10KΩ	R9	100Ω		
		R2	10KΩ	R6	470KΩ				

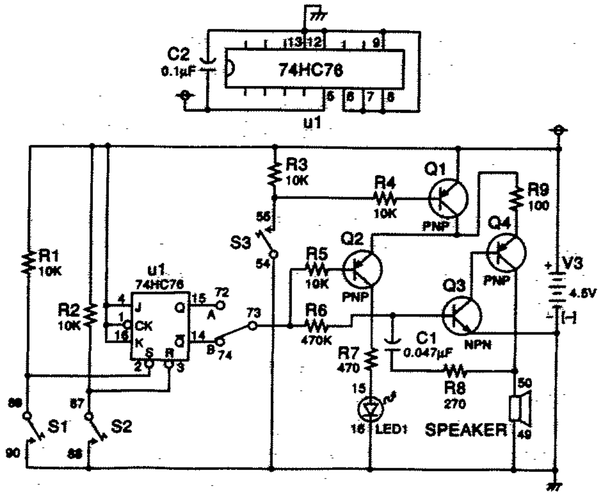
Flip-flop circuits can be used in game circuits since there's no way to tell if the flip-flop circuit is set or reset just by looking at it. Trying to outwit a flip-flop circuit can be frustrating ... as this project demonstrates.

You can see how this circuit works by looking at the schematic. You set or reset the flip-flop using **S1** and **S2**. The Q and $\overline{Q}$ outputs are selected by the **select switch**. Depending on whether the two outputs are 0 or 1, either **LED 1** lights or the audio oscillator sounds when **S3** is pressed.

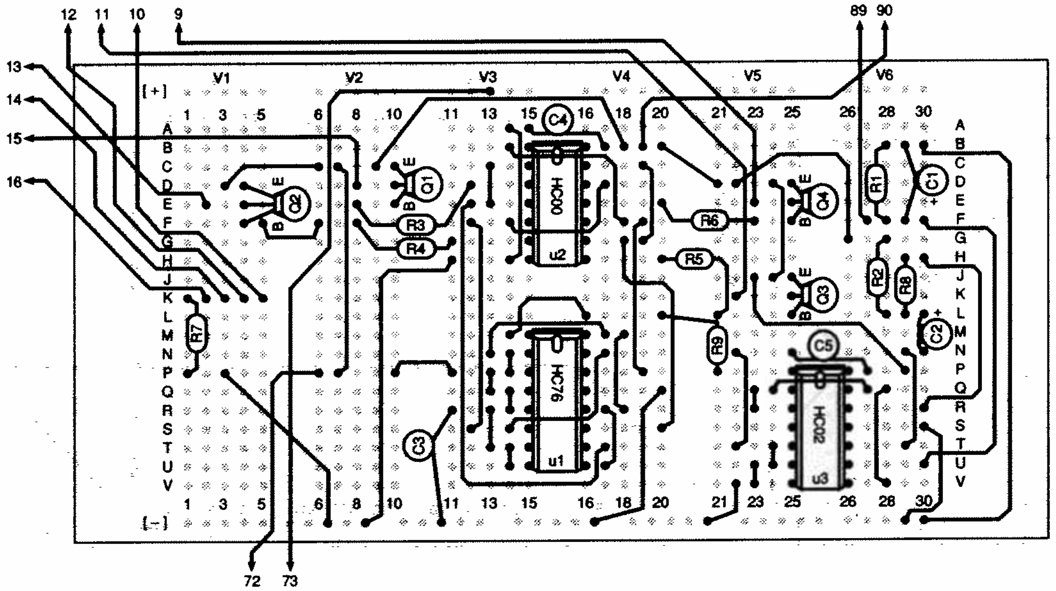
To play with this circuit, press **S1** or **S2**. Now set the **select switch** to either up or down position. Press **S3** - what happens?

You either got a sound from the **speaker** or **LED 1** lit. Continue to hold **S3** down and press **S1** or **S2**. You'll see the opposite now happen - if the **LED** is lit, it goes out and you'll hear a sound from the **speaker**.

This shouldn't too much of a surprise to you ... as you can see on the schematic **S1** is the set input and **S2** is the reset input. You can use this project as a game by pressing **S1** or **S2**, and then trying to figure out whether the **LED** lights or a sound comes from the **speaker**. What are the combinations of Q and $\overline{Q}$ levels (0 or 1) and **select switch** positions to let each happen?



PROJECT 243. BE YOUR OWN MULTIVIBRATOR



U1	74HC76	Q1	PNP	R1	47KΩ	R6	10KΩ	C1	1µF
U2	74HC00	Q2	PNP	R2	47KΩ	R7	470Ω	C2	1µF
U3	74HC02	Q3	PNP	R3	10KΩ	R8	1KΩ	C3	0.1µF
		Q4	PNP	R4	10KΩ	R9	10KΩ	C4	0.1µF
				R5	10KΩ			C5	0.1µF

Notice anything familiar about the schematic for this project? If you get the feeling that you've seen this circuit before, you're right. Take a look back at the schematic for project 186 - you'll see that it's the same counter and line decoder circuit without the multivibrator. This project lets you be your own multivibrator.

As you can see from the schematic, each time you press **S1** you send a clock signal to the first flip-flop. The counter circuit is asynchronous since the Q output of the first flip-flop provides the clock signal for the second flip-flop.

Turn power ON and press **S1** a few times. You'll see **LEDs** 1 through 4 light up in order and then go off. This cycle is repeated a few times depending on how many times you press **S1**.

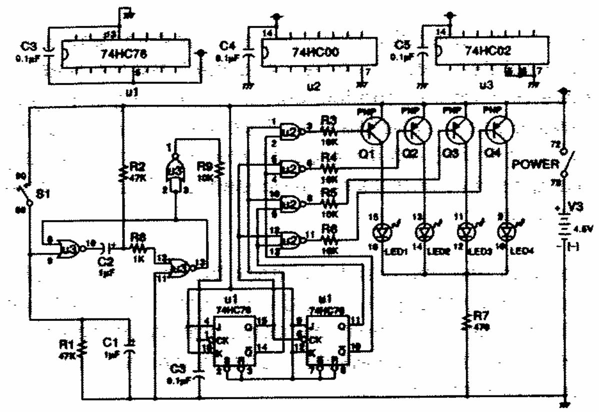
Of course, there's several ways to vary this circuit. You could supply the set or reset signal instead of the clock ... or you could let the multivibrator supply some of the inputs while you supply the others.

Back in project 191 we told you that you could substitute a **S1** for a multivibrator circuit in a counter. If you tried to do it, did your results look anything like this circuit?

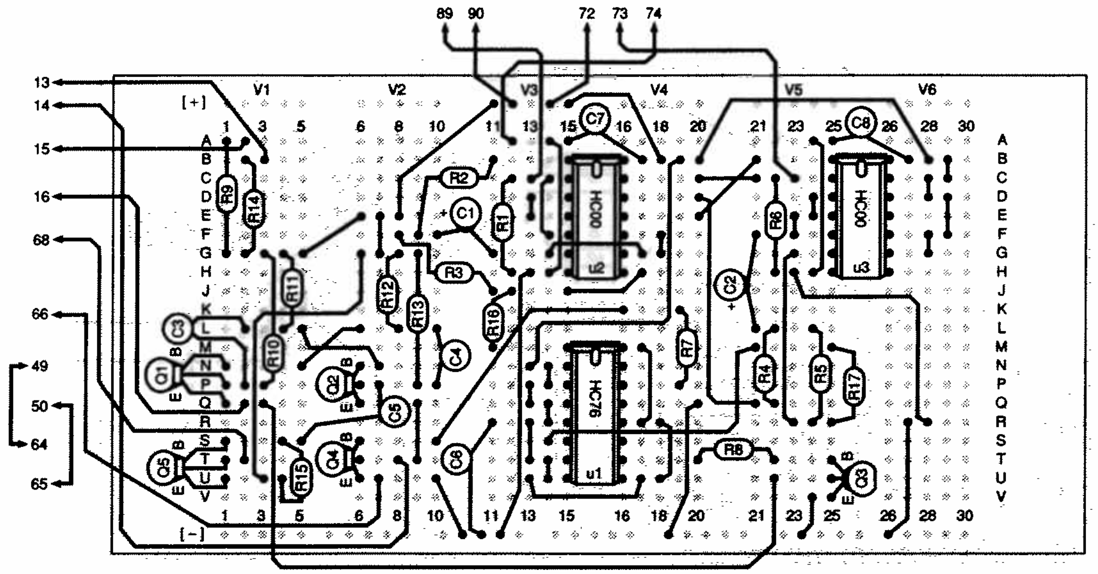
NOTES :

1. When pressing **S1** don't do it too quickly (fast pressing might result in what we call switch "bounce", i.e. "chattering" - which means very, very fast un-intentional switch action).

2. Be sure you're using fresh batteries (or you might end up with some strange results).



PROJECT 244. ANTICIPATION

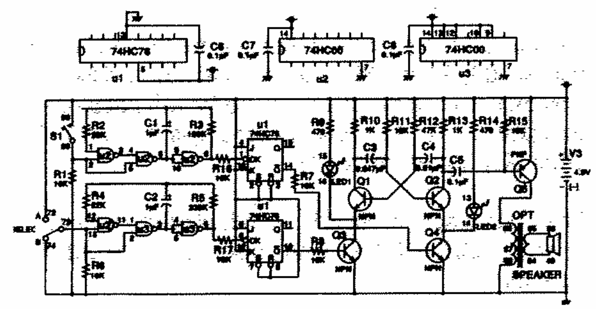


U1	74HC76	Q5	PNP	R7	10KΩ	R14	470Ω	C3	0.047µF
U2	74HC00	R1	10KΩ	R8	10KΩ	R15	10KΩ	C4	0.01µF
U3	74HC00	R2	22KΩ	R9	470Ω	R16	10KΩ	C5	0.1µF
Q1	NPN	R3	100KΩ	R10	1KΩ	R17	10KΩ	C6	0.1µF
Q2	NPN	R4	22KΩ	R11	10KΩ	C1	1µF	C7	0.1µF
Q3	NPN	R5	330KΩ	R12	47KΩ	C2	1µF	C8	0.1µF
Q4	NPN	R6	10KΩ	R13	1KΩ				

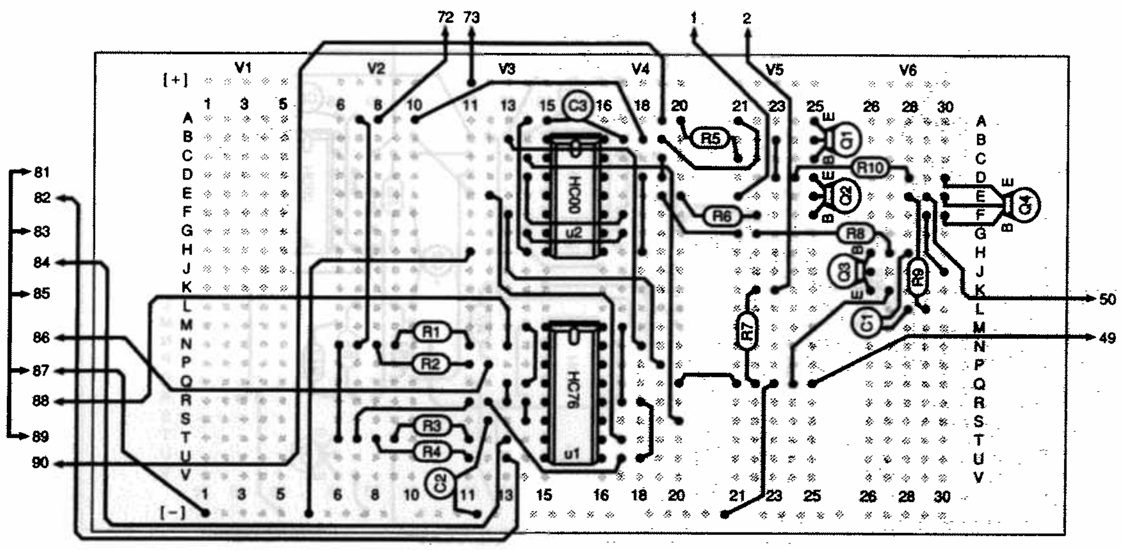
Think you have a good sense of timing? If you're not sure, here's a project that lets you quickly find out.

A quick glance at the schematic shows this project uses both a J-K flip-flop and an astable multivibrator. The astable multivibrator produces a sound from the **speaker** and you'll also notice **LEDs 1** and **2** light up at different rates. Both **S1** and **select switch** are used here to control the inputs to the J-K flip-flop circuits. Your job is to select that combination of switch settings that lets both **LEDs** light and also produce a sound from the **speaker**.

Set the **select switch** up to generate pulses for the J-K flip-flop. You see **LED 1** flicker but hear no sound. Set the **select switch** down: the **LED 1** stays in either condition, on or off. Now try press **S1**. You'll now see **LED 2** flicker. Release it, and the **LED 2** stays either on or off. If you manage to set the switches to the correct positions at the right time you'll light both **LEDs** and hear a sound from the **speaker**.



PROJECT 245. SET/RESET MATCH



U1	74HC76	Q3	NPN	R3	10KΩ	R7	470Ω	C1	0.047µF
U2	74HC00	Q4	PNP	R4	10KΩ	R8	470KΩ	C2	0.1µF
Q1	PNP	R1	10KΩ	R5	10KΩ	R9	470Ω	C3	0.1µF
Q2	PNP	R2	10KΩ	R6	10KΩ	R10	100Ω		

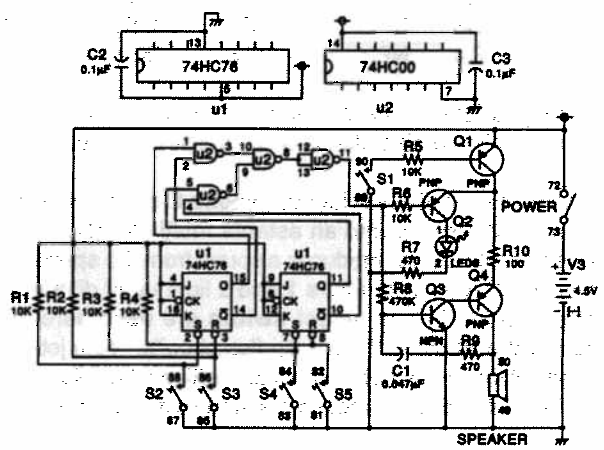
You can't tell if a J-K flip-flop is set or reset just by looking at it. This project takes that simple fact and makes a game out of it.

As you can see from the schematic for this project, you use **S2, S3, S4** and **S5** to set or reset both J-K flip-flops. You can use this circuit as a game for two people. First turn power ON. Next one player press **S2** or **S3**. The other player press **S4** or **S5**.

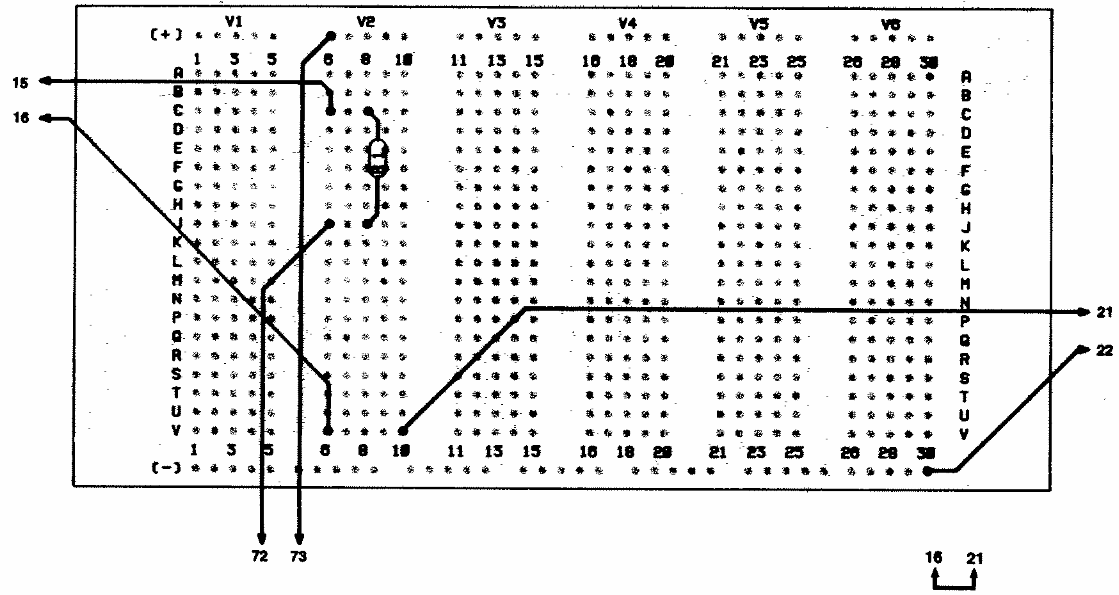
Then one player press **S1**. If both players managed to "match", "**LED 8**" lights. If not, a sound comes out of the **speaker**.

Be sure to play with this circuit a few times and find out whether the **LED** lights when both flip-flops are set or reset (or both). You can also use this circuit to "test" the compatibility of two people. If the **LED** lights, the two people are compatible; if you hear sound from the **speaker**, then the people aren't compatible!

Try some variations of this basic circuit - see if you can add a multivibrator, for example.



PROJECT 246. CIRCUIT CONTINUITY CHECKER



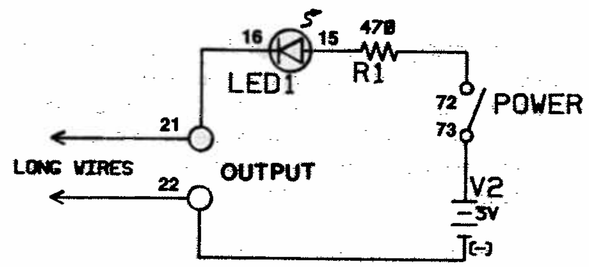
R1 470Ω

We're now going to look at how we test electronic circuits. One of the simplest things we might want to find out is whether or not a part of a circuit is open or closed. This project is simple but it does the job.

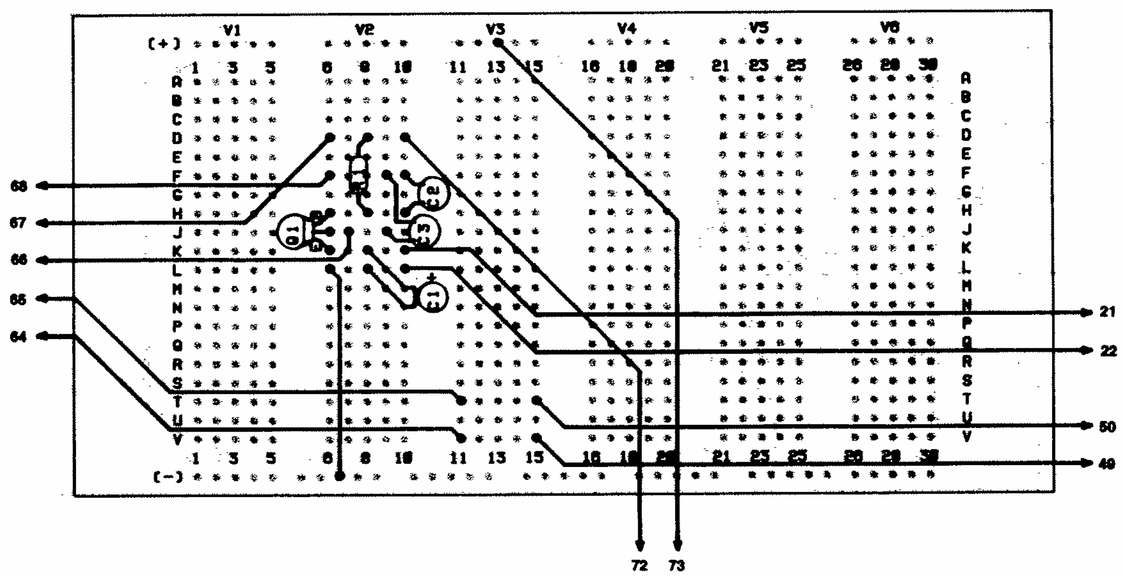
Apparently you figured out how to use this project just by looking at the schematic. Turn power ON and place the two wires at opposite ends of the circuit part you want to test. If the current can flow through the circuit, the **LED** lights. If it does not light at all, you'll know the circuit portion is open.

This is an especially handy circuit for testing items such as insulated wire or cable. You can't tell if a plastic-covered wire can conduct current by looking at it - you have to use a tester like this. This device is also helpful in trying to find out if there's an electric current flowing along a path where it's not supposed to. (This is called a short circuit.)

You can use this circuit to check the conductivity of a wide variety of object around your home. But don't use this project to check the conductivity of anything connected to a wall electric outlet (such as an electric range or record player).



PROJECT 247. ACOUSTIC OHMMETER



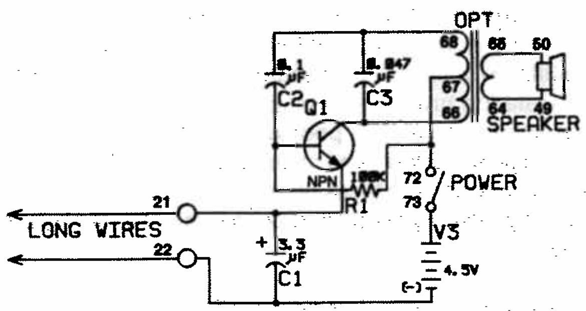
Q1	NPN	C1	3.3µF
R1	100KΩ	C2	0.1µF
		C3	0.047µF

An ohmmeter is a device that measures the resistance of a circuit or a part (remember that resistance is measured in ohms.). Most ohmmeters have a meter of LED display to indicate the resistance. This project is a bit difference in resistance between various parts.

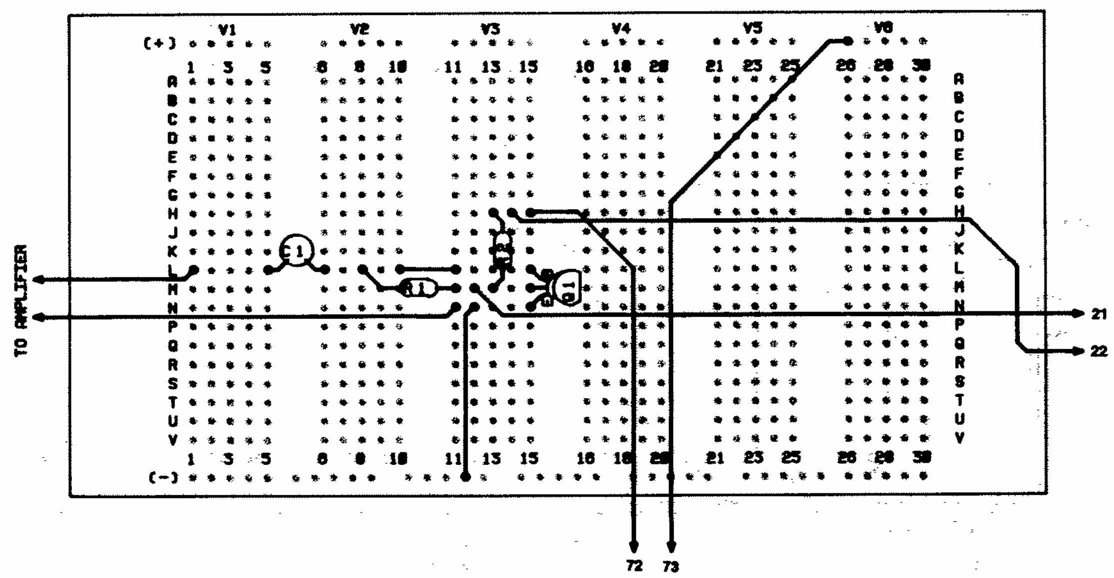
This project is actually a version of the audio oscillator circuit we've used in many other projects. When a resistance is connected across the two long wires, it provides negative feedback so the circuit can oscillate. After you finish the wiring connections, turn power On. Measure the resistance of several different resistors in your kit.

What differences in sound do you notice when you measure different value resistors? Is there a pattern to these changes? (You'll probably find it helpful here to review the notes you've made on audio oscillator projects.)

Try touching the end of the two long wire together. What do you hear now from the **speaker**? Can you explain why this happens?



PROJECT 248. AUDIO SIGNAL TRACER



Q1	NPN	R2	4.7KΩ
R1	470KΩ	C1	0.1µF

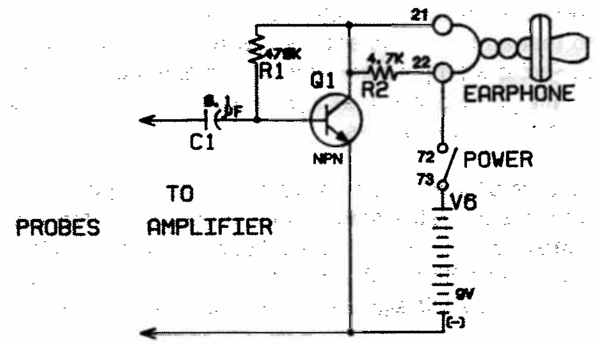
This project is a simple one transistor audio amplifier which is used as an audio signal tracer. With this amplifier you can troubleshoot transistor audio equipment. You do this by connecting the probes across the circuit from stage to stage until you find the stage or component which is not passing the signal along.

No volume control is used with this amplifier because you can use the volume control on the equipment being checked to adjust signal levels when necessary.

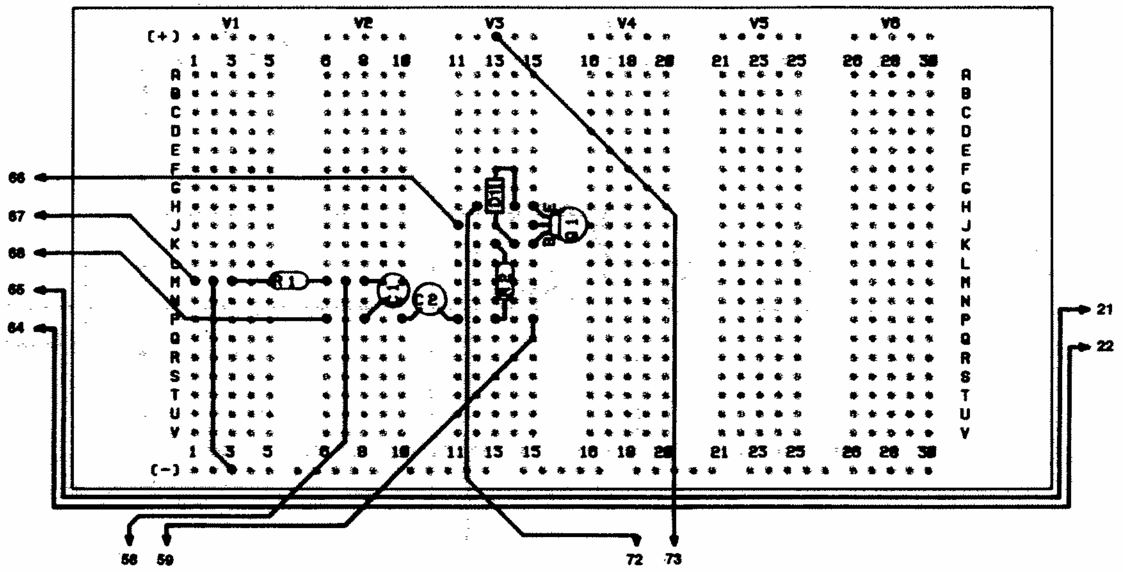
The 0.1µF input capacitor blocks DC so you can probe around circuits without worrying about the effects of DC voltage on the circuit.

The amplifier circuit is the common-emitter type. The bias current is the self current type. That is, the bias current (through the 470K) is obtained from the collector voltage, providing some stabilizing negative DC feedback. This is a very simple and popular circuit for silicon transistors such as the one in your kit.

Use the amplifier to probe around on any transistor radio or amplifier you have that needs fixing. You can also use this circuit to test some of the other projects in this kit. The only requirement is that they do not have any common components. Have fun!



PROJECT 249. AUDIO SIGNAL GENERATOR

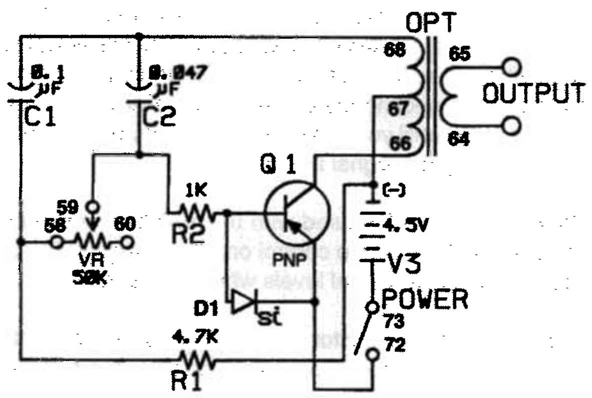


Q1	PNP	R1	4.7KΩ	C1	0.1μF	D1	Si
		R2	1KΩ	C2	0.047μF		

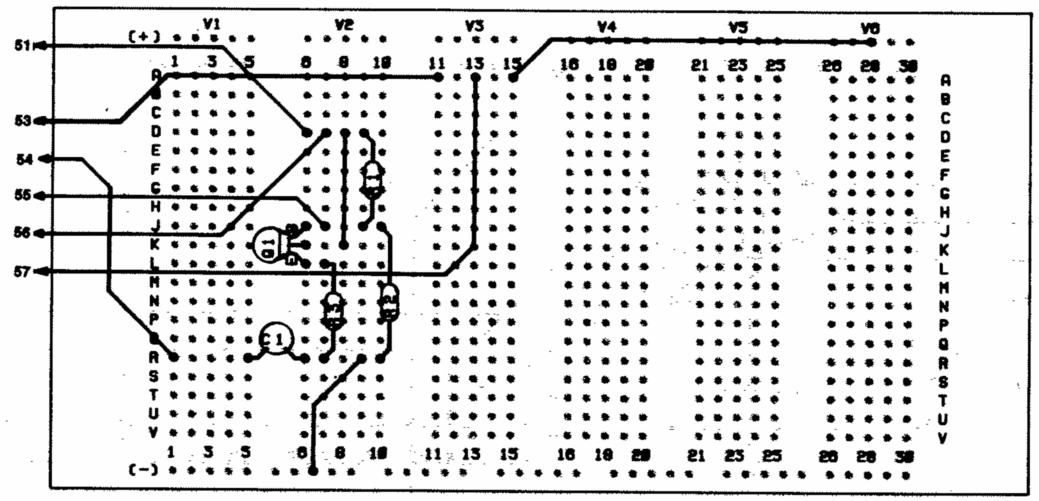
Very often electronic technicians need an audio signal to test amplifiers, speakers, headphones, etc. This project is a simple audio signal generator that operates on the same principle as the ones used in professional electronics laboratories.

This circuit is an audio oscillator, and the output of this circuit is available at terminals 16 and 15. Here you connect the circuit or component you want to have tested, like an amplifier or the speaker. You can adjust the tone of the audio signal using the control volume.

Audio signal generators are often used to adjust stereo equipment. They help make sure the tone that goes into a stereo amplifier is the same tone that comes out of the amplifier. If the amplifier changes the tone of the signal, this is called distortion.



PROJECT 250. METAL DETECTOR



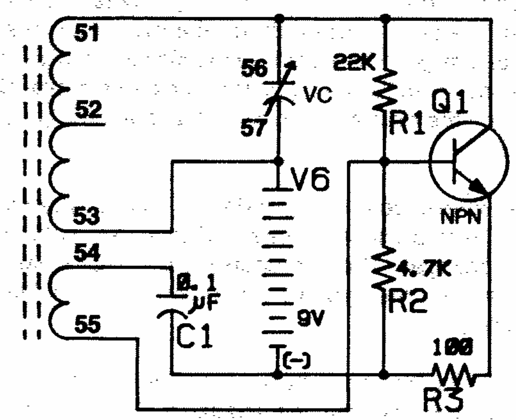
Q1	NPN	R1	22KΩ	R3	100Ω
		R2	4.7KΩ	C1	0.1μF

This project is a demonstration of a metal detector. When it comes near to any metallic object, the coil of an oscillator changes the frequency of the oscillator to indicate the presence of metal. These types of metal detectors are used by people to locate lost treasures, buried pipes, hidden land mines and much more. During war time especially these have been used to save many lives by locating mines and booby traps set out by the enemy.

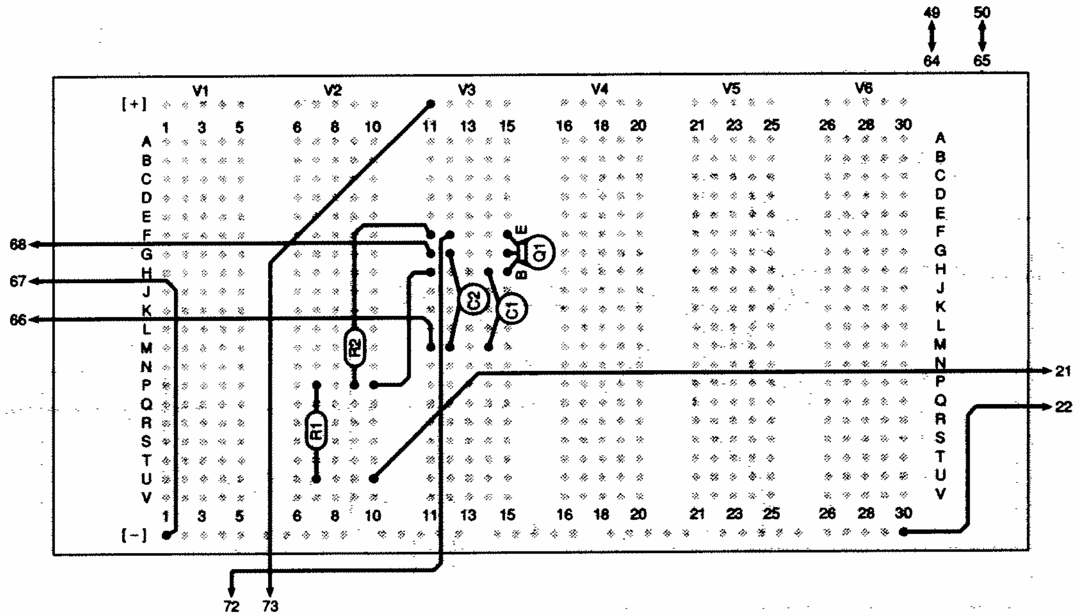
This circuit is a low distortion oscillator that draws only one milliamp from the 9V supply. Low power is desirable to allow the nearby metal to have maximum affect on oscillation frequency.

Use a small transistor radio tuned to a weak AM broadcast station as the detector for this oscillator signal. Tune this oscillator until a low-frequency beat-note is heard. This beat-note is the difference between the broadcast station signal and this oscillator signal. Do not bring the radio closer than necessary. Just enough to obtain about the equal level on the two signals (the radio station and this oscillator). This is the best sensitivity setting.

Try using keys, plastic objects, coins, etc. as samples of what to expect when using a metal detector like this. Of course a real metal detector dose not have a small ferrite coil like this. It is usually an air core coil which is shielded with a aluminum electrostatic shield called a "Faraday electrostatic shield". This project at least gets the point across.



PROJECT 251. RAIN DETECTOR



Q1	PNP	R1	100KΩ	C1	0.01μF
		R2	100KΩ	C2	0.1μF

One way to tell if it's raining is to stand outside without an umbrella... if you get wet, it's raining! Fortunately, there is a better way - our rain detector circuit.

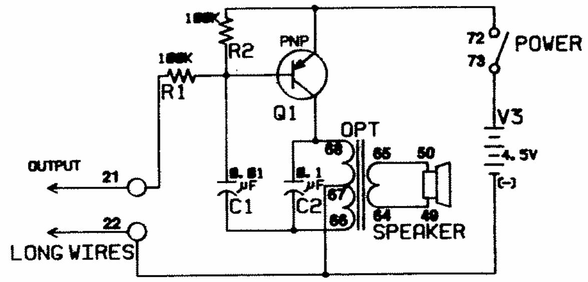
After you finish the wiring for this project, carefully place the exposed ends of the two long wires side-by-side on a piece of plastic or wood. Place the two exposed ends as close as you can without having them touch. Use tape to hold the two long wire in place (make sure their exposed ends don't touch and that the exposed ends aren't covered by the tape.)

Using this project is a snap. Just turn power ON and place the two long wires wherever you want to detect rain or water. When water falls on the ends of the long wires, you'll hear a sound from the **speaker**.

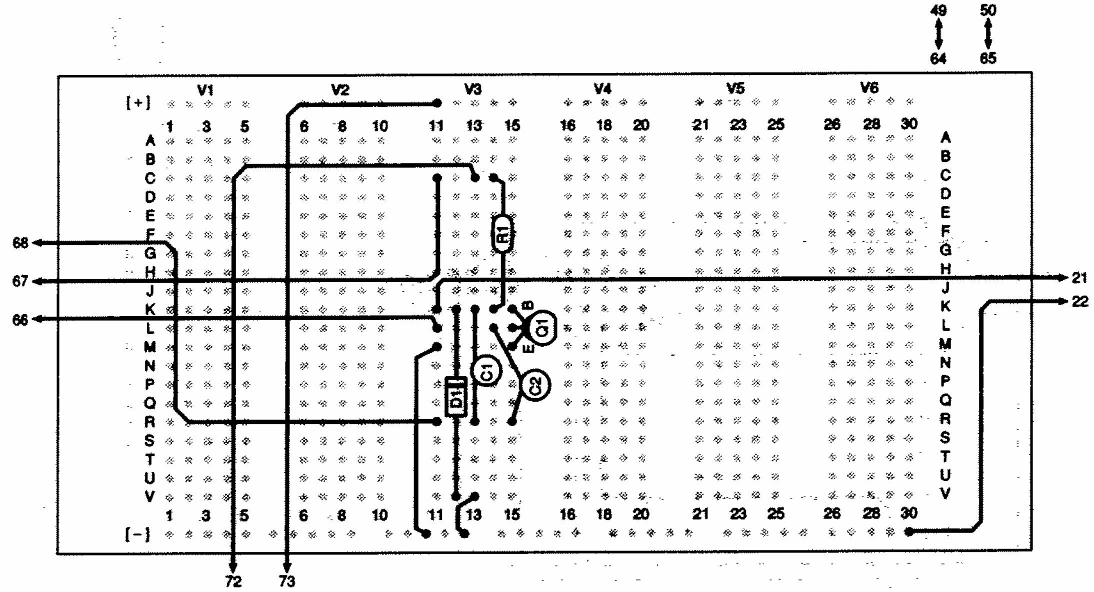
You can use this project for more than just detecting rain, of course. You can use it to check if a faucet is dripping, or if the water in the bathtub or aquarium is getting too high.

This project works because water can conduct current, just like a wire. When we say something can "conduct" current, we mean current can flow through it. When water falls on the exposed ends of the two long wires, it "completes" the circuit and you hear a sound from the **speaker**. But water doesn't conduct current as well as a wire, which is why we have the exposed ends so close together.

But does all water conduct alike? Try this - remove the two long wires from the wood or plastic you taped them to. Get two glasses of water. In one glass dissolve as many spoons of salt as you can. In the fresh water glass place the exposed ends of the two long wires as far apart as you can. Move one long wire toward the other. Note when you hear a sound from the **speaker**. Now try the same thing with the glass of salt water. In which one did the detector sound with the wires farthest apart? That's the glass with the water that conducts current best!



PROJECT 252. BURGLAR ALARM



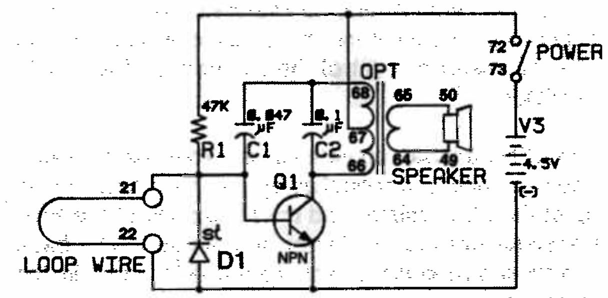
Q1	NPN	R1	47KΩ	C1	0.047μF	D1	Si
				C2	0.1μF		

Here's a burglar alarm project that's really sneaky! You can locate it away from the object you want to protect - and no one can know about the alarm until after they've set it off.

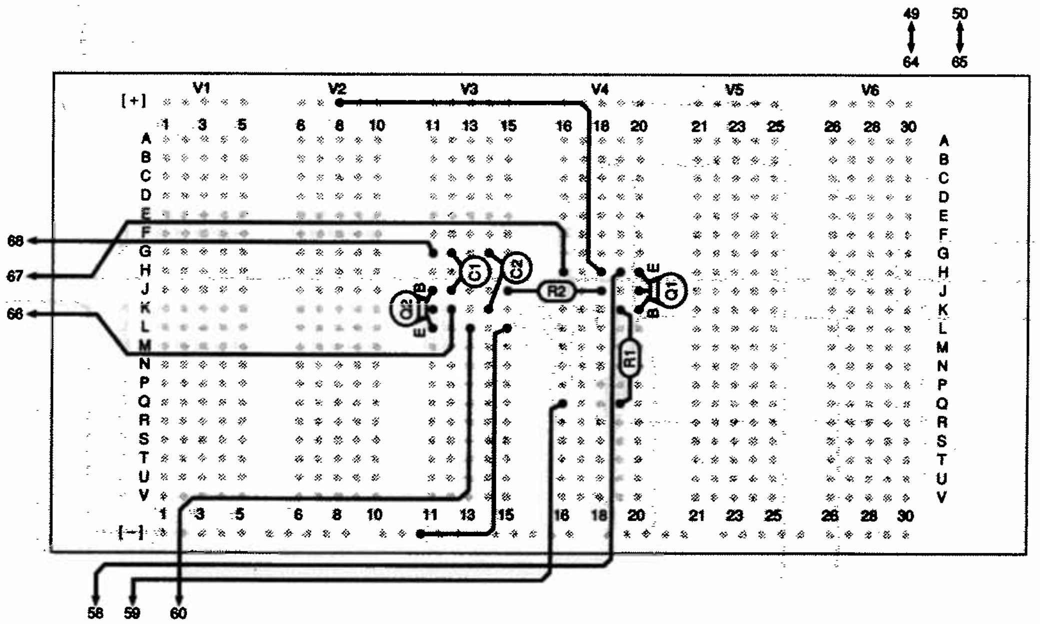
This project uses the long length of wire included with your kit. Normally, this wire is used for an antenna in radio projects. But in this project it's used to trip the alarm.

Before using this project, make sure that the long green wire is connected between terminals 21 and 22. Now turn power ON, and pull the wire out of either the terminal 21 or 22. What happens?

You probably got your ears blasted! Whenever the wire is disconnected, the alarm sounds. This means you could tape the green wire to a door, window or drawer so that the alarm would sound if one of those objects were moved, causing the wire to be pulled out of one of the two terminals. (Now you can find out who's been messing around your closet or drawers!)



PROJECT 253. TEMPERATURE-SENSITIVE AUDIO AMPLIFIER

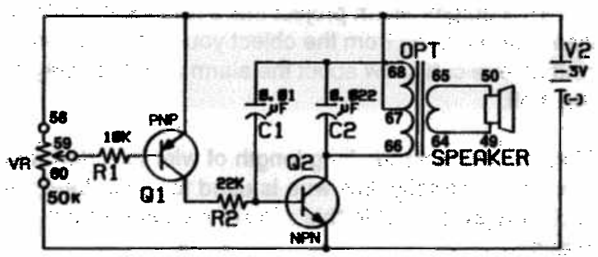


Q1	PNP	R1	10KΩ	C1	0.01μF
Q2	NPN	R2	22KΩ	C2	0.022μF

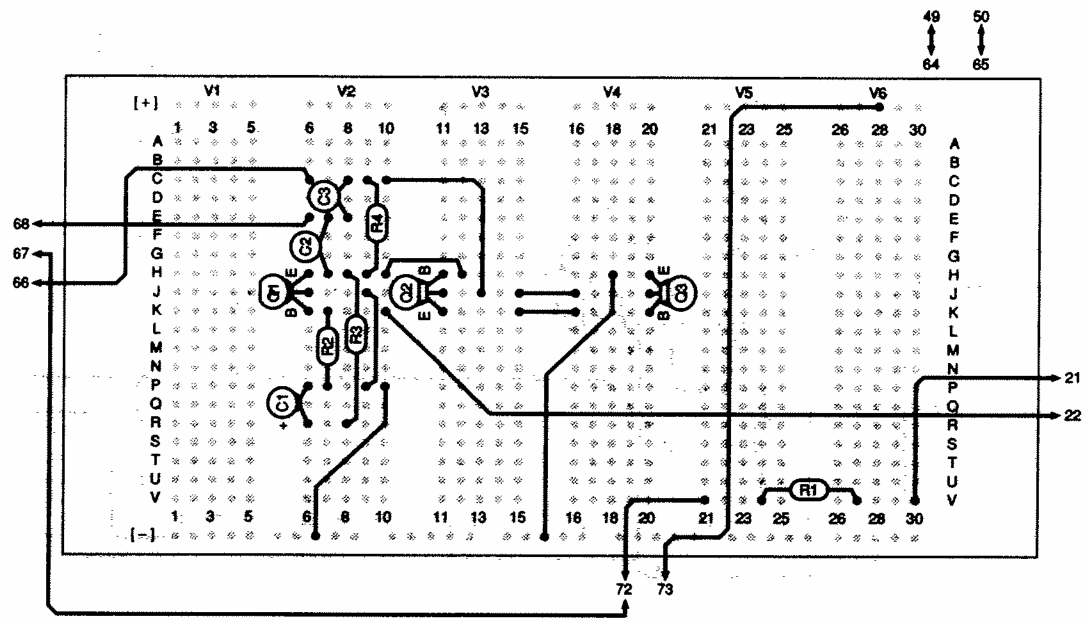
This project demonstrates how transistor collector current depends on temperature, by using the leakage current of a transistor to control the frequency of an audio oscillator.

The NPN transistor operates as a pulse-type oscillator. The bias voltage is controlled by the series circuit made up of the 22K resistor and the resistance of the PNP transistor between collector and emitter. The base current and collector current of a transistor vary with the temperature.

Adjust the 50K **control volume** so that the **speaker** output is a low sound or pulses. Now warm up the PNP transistor by holding it between your fingers. You should hear the tone become higher as the transistor temperature is increased.



PROJECT 254. WATER LEVEL DETECTOR

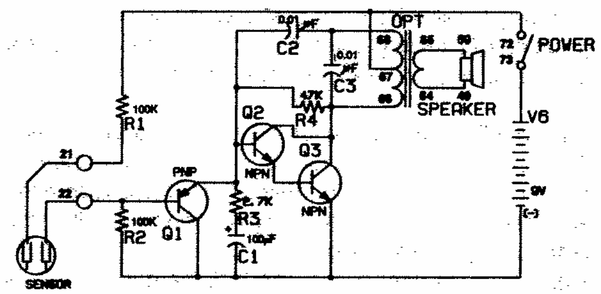


Q1	PNP	R1	100KΩ	R4	47KΩ	C3	0.01μF
Q2	NPN	R2	100KΩ	C1	100μF		
Q3	NPN	R3	2.7KΩ	C2	0.01μF		

We're now going to make a water level detector that gives out a "peep" generated by Q2 and Q3 oscillator. This oscillator is controlled by switching transistor Q1, which turns ON and OFF by the water sensor.

When the wires from terminal 21 and 22 do not contact any water, Q1 stays ON and Q2 and Q3 don't work. When the water contacts the wires, Q1 turns OFF and Q2 and Q3 begin working.

When you finish wiring, turn power ON and bring wires into contact with water. You'll hear a "pip" at this time. You might want to tape the wires to the bathtub so you don't have to worry if the tub is overflowing....



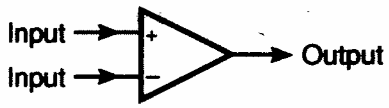
Note : The arrow sign points ("→") to related words.

[A] **A-D conversion (Analog-to-Digital conversion)**
The conversion of analog signals into digital signals. In this kit, for example, the illuminance meter uses an A-D conversion circuit. The amount of light (an analog quantity value) received by the phototransistor is converted into digital data, and its degree of brightness is numerically displayed. (→ D-A conversion)

AC (Alternating Current)
A current that changes in value and direction in constant cycles. Usually, it has a sine waveform. Regular households use AC of 115V, 220V, and 240V, etc. (→ DC, Sine wave)

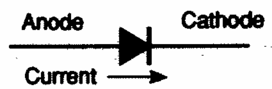
AGC (Automatic Gain Control) circuit
A circuit that automatically controls its gains. For example, let's suppose you are taping a discussion in your classroom. The AGC circuit will automatically raise the gain, thus raising the volume of the sound being recorded, for people speaking from far away, while the voice of someone nearby (a louder voice) will be recorded with a smaller sound by lowering the gain.

Amplifier
Amplifiers amplify input signals. In this kit, the "inverting amplifier," "non-inverting amplifier," and the "differentiating amplifier" are introduced. Additionally, various types of amplifiers are used. Different types of amplifiers are made by specific uses operational amplifiers.



Analog
A value that can be expressed as a continuously changing physical quantity (such as the amount of voltage). (→ Digital)

Anode
One of the poles of a diode. Electric currents flow from the anode to the cathode. That means you can find out the direction in which electric current is flowing by closely observing the diode's symbol.
(→ Cathode)



Audio frequency
Frequency that is audible (that humans can hear). (Frequencies between approximately 20Hz to 20kHz) With this kit, the sound comes out of a speaker, and sounds with audio frequencies of around 1kHz are principally used for it.

[B] **BCD (Binary Coded Decimal)**
Numerical notation that expresses 1 digit decimal number as a 4 digit binary number. It can be understood as a numerical notation that bridges computers and humans.

Bias
To apply voltage or current to a device, such as a transistor, beforehand, so the device will operate correctly. You will come across phrases like "apply a bias voltage to the transistor," and "flow some amount of current through the Zener diode." When you tell a friend that his "thinking is biased," you mean that he has already formed an opinion "beforehand," which is like there is already a voltage applied to his head before he needs to think (operate).

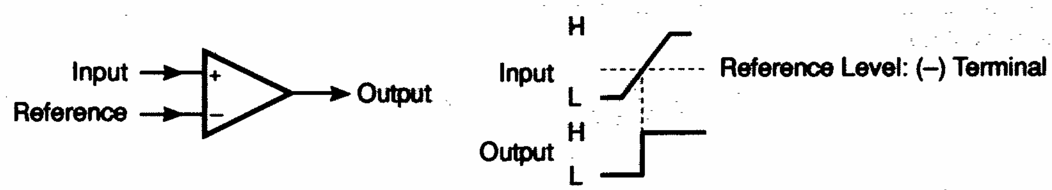
Binary notation
Numerical notation used for arithmetic operations carried out in computers. Humans, on the other hand, count and calculate in the decimal notation.

[C] **Cathode**
One of the poles of a diode. Electric currents flow from the anode to the cathode. (→ Anode)

Charge
The storing of electricity ("electric charges," to be accurate) in capacitors. (→ Discharge)

Chattering
When a key is pressed or released, a short cycle of ON and OFF is repeated until the key's contact comes to rest. This phenomenon is called chattering. As you can imagine, chattering is not desirable for circuits. For example, even if you thought you pressed a key once, the circuit might interpret that it was pressed several times. For this reason, an additional circuit that absorbs chattering is installed to the key's input circuit when necessary.

Comparator
An operational amplifier used, not as a signal amplifier, but as a signal comparator. The output of a comparator is either "low" or "high." It is low when the level of a comparator's positive (+) terminal is lower than that of its negative (-) terminal, and high when the positive is higher than the negative. Since the output of a comparator is digital, it can be used as the input for a logic circuit.



Count down
To reduce the counter's (numerical) value.

Count up
To increase the counter's (numerical) value.

Cycle
An interval of time in which one set of phenomena is completed. (→ Period)

[D] **D-A conversion (Digital-to-Analog conversion)**
The conversion of digital signals into analog signals. In this kit, digitally processed outputs are converted to analog voltage, by means of a resistance network (a circuit assembled with resistors), to create sounds that come out of the speaker. In short, digitally processed data you need to hear or see as a physical quantity must first go through D-A conversion. (→ A-D conversion)

DC (Direct Current)
An electric current that flows in one direction at a constant value. You probably know that batteries are DC power supplies. (→ AC)

DC-DC conversion (Direct current to direct current conversion)
This term is normally used for the process of turning a low voltage into a high voltage (boosting). High DC voltage cannot be created directly from a low DC voltage. So, it must first be converted to AC, and boosted to a higher voltage, then returned to DC.

[D] Decode
To decipher significant (or meaningful) signals from received signals. You probably read about decoding in spy novels. In this kit, binary coded decimal (BCD) inputs are decoded into the decimal notation that we commonly use.

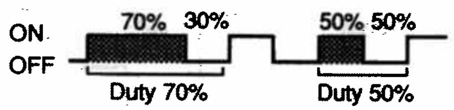
Detector
A device that detects the presence of light and other matters. Detectors that use for light are called photo-detectors.

Digital
Values that are expressed numerically.
(→ Analog)

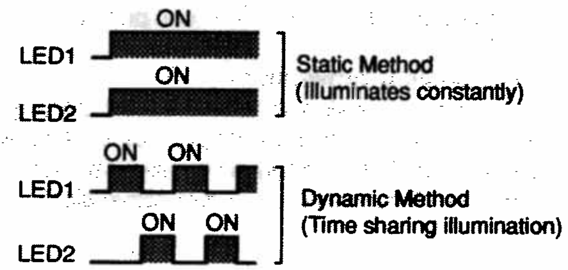
Discharge
To release (discharge) electricity ("electric charge" to be accurate) that is charged to a capacitor.
(→ Charge)

Dividing
To reduce the frequency of a pulse waveform to 1/2, 1/4, 1/8 ... and so forth. The desired pulse frequency can be obtained by dividing the oscillator's output.

Duty
The proportion of significant (meaningful) "ON" signals in 1 period of a pulse waveform. It is expressed as a percentage.

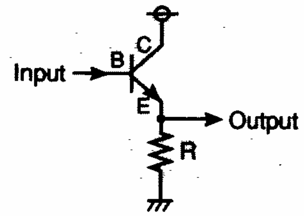


Dynamic method
The dynamic method is the opposite of the static method. Let's see how the two differ with examples of LED illumination. With the static method, there is an exclusive circuit for each LED. With the dynamic method, on the other hand, the circuit is commonly used for all LEDs, and the LEDs are illuminated on a time-sharing basis. Since the circuit can be made compact with this method, the dynamic method has an advantage when there are many display items. Calculators use the dynamic method for their displays. The displays blink rapidly so the human eye's attention is not compromised by time sharing illumination.



[E] Earth
Also called "ground." The negative (-) terminal of this kit is its reference electric potential (its earth). The reference electric potential in an equipment is usually called its earth. The word "earth," as well as "ground," are used for this convention because it was originally meant to equalize an equipment's reference electric potential with the electric potential of the planet's surface by connecting it with the earth (or ground).

Emitter follower
A current amplifier that creates a "1" voltage gain using a transistor. It is also called a buffer. As shown here, the collector is connected directly to the power supply. (→ Voltage follower)



[F] Feedback
To return (feedback) an amplifier's output signal back to its input signal. The word feedback will probably appear frequently in your daily conversations, too. Example : "What happened with that matter? Don't forget to give me a feedback on its results." (→ Positive feedback, Negative feedback)

Frequency
Numerical value that shows how many times a sine wave or a pulse wave is repeated in 1 second. For example, the frequency of alternating currents in regular households is 50Hz or 60Hz. With a 50Hz frequency current, its sine waveform appears 50 times in 1 second. In the project, "Light Source Sensing Circuit by Sound," 50Hz and 60Hz sounds can be heard.
The frequency is expressed as an inverse number of the period.
Frequency = 1/period (→ Period)

[G] Gain
The ratio between the voltage, current, or electric power, of an amplifier's input and its output. If 1 Volt is input to an amplifier and its output is 2 Volts, for instance, the gain is 2.

[H] Hexadecimal number
Notation used when handling computer data and commands. It consists of the 16 alphanumeric characters, 0 through 9, A, B, C, D, E, and F. Hexadecimal numbers are expressed with the suffix "H," as in "2FH," for instance. 2FH is the 2 digit hexadecimal number for 2F.

[L] Logic level
Either "1 and 0," or "H and L." Since it is used in a logic circuit, there is no intermediate (or half measure) level. It is either 1 or 0.

[M] Mixing
The process of mixing 2 or more signals.

Modulation
To alter a signal's amplitude, frequency, or phase, etc. (While this description is not entirely accurate, it will be convenient to understand this term in this way for now.)

[N] Negative feedback
Feedback in which the amplifier's output component is returned (fed back) in a phase that makes the input signal smaller. Since amplifiers tend to generate unnecessary oscillations, operations can be stabilized by giving negative feedbacks to the amplifier so the gain is held at a constant value.
(→ Positive feedback)

[O] Oscillation
The generation of waveforms. In analog circuits sine waveforms are generated, while in digital circuits, pulse waveforms are generated using astable multivibrators, for instance.

[P] Period

The time it takes to complete 1 cycle of a sine wave or pulse wave. For example, the period of an alternating current of 50Hz frequency is 1/50 = 0.02 seconds (20msec.).
The period is expressed as an inverse number of the frequency.
Period = 1/frequency

Phase

This term is used to show the positional relation between sine waves and pulse waves, etc. where 1 cycle is expressed as 360 degrees. For example, you will come across phrases like "a waveform that has a 180 degrees delayed phase with respect to the reference point." Imagine a three-legged race at a sporting festival in electric terms. If the two partners move their tied legs in "matching phase," (or "in-phase"), they can run as if they were one runner. Surely, the partners who are in-phase will win the contest. It's even O.K. to have a slight "phase difference." Well, of course, it would be a great problem if they were in "opposite phase" (phases shifted by 180 degrees), would it not?

Positive feedback

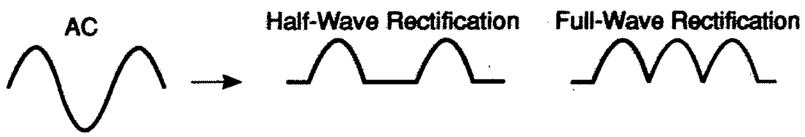
The feedback of an amplifier's output component that is matched to the input signal's phase. Since positive feedbacks are made to enlarge input signals, output signals also become larger. For this reason, positive feedbacks are used when structuring oscillators. (→ Negative feedback)

[R] Random

It means to be "free or aimless." A typical use of this word would be something like "press the key, and one of the 5 LEDs will be randomly illuminated."

Rectification

To convert an alternating current to a direct current. Diodes are used for rectification, and in this kit, a full-wave rectification project is also introduced. The diagram shows an alternating current rectified to a half-wave waveform and a full-wave waveform. (→ AC, DC)

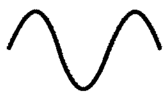


Resonance

A phenomenon in which an object (such as a swing) starts to vibrate (or swing) substantially, even though only a small force is applied to it, when the cycle of that force is synchronous with the object's. Recall how a heavy church bell can be moved with just one finger. In electric circuits, resonating phenomena are generated by capacitors and coils. This phenomenon is used to tune into broadcast frequencies of radio stations.

[S] Sine waveform

The most basic waveform. For example, if red paint is painted on 1 location of a bicycle's front wheel, the visual trail the red paint forms as the bicycle moves forward is a sine waveform.



Square wave

A pulse waveform that has even portions of ON and OFF (50% of duty).

Static method

The opposite of dynamic method. (→ Dynamic method)

[T] Threshold

The "threshold value" to distinguish whether an input signal is logic 1 or 0. The threshold of CMOS ICs used in this kit is approximately 1/2 of the power supply voltage. Input signals that are higher than the threshold level are logic 1 and input signals that are lower are 0.

Time constant

With pulse circuits, for example, the smoothness of a CR differentiating circuit's waveform is determined by "C (capacitor) x R (resistance)." In other words, it is a time constant of "C x R." The larger the time constant, the smoother the waveform's rate of change.

Trigger

You probably already guessed what this means from the trigger of a pistol. Pull the trigger, and "BANG!" In other words, it initiates (or triggers) an operation. Trigger pulses are pulses that initiate circuit operations.

[V] VCO (Voltage-controlled Oscillator)

The kit uses various types of VCOs to display LEDs and make sounds. The frequency generated by a VCO changes with the voltage level that is applied to the input.

Voltage follower

Current amplifier with a voltage gain of "1" using an amplifier. It is also called a buffer. Voltage followers are connected in the following way. (→ Emitter follower)