RTL Cookbook

BY DON LANCASTER

A complete guide to the understanding and using of Resistor-Transistor Logic (RTL) digital integrated circuits.

RTL Cookbook

by

Don Lancaster

SYNERGETICS SP PRESS
3860 West First Street, Thatcher, AZ 85552 USA
(928) 428-4073   http://www.tinaja.com

Thatcher, Arizona 95552

THIRD EDITION
FIRST PRINTING—2010

International Standard Book Number: 1-882193-10-3

Created in the United States of America.

Preface

This book will help the electronics experimenter understand and use the low-cost digital integrated circuits now available for practical, everyday electronics projects. The material presented attempts to shatter the myth that digital IC's are too expensive, too complex, or too awesome to use intelligently in simple circuits.

In addition, this book shows the technician the *why* of digital IC's—how they work, how to use them, and how to design with them. It tells how digital instruments work and how to design and build your own fully integrated IC systems.

Also, this book should be valuable to the engineer who is tired of wading through a stack of application notes and pre-IC computer books to try to find realistic and reasonable designs for such things as divide-by-n scalers, low-cost decimal counter/readouts, IC monostables, synchronizers, or other circuits. The three chapters on counting flip-flops, divide-by-n counting, and decimal counting provide circuits virtually ready to drop into systems for immediate use.

The reasons this book deals entirely with Resistor-Transistor Logic (RTL) are the relatively low prices of this digital-IC line, the ease with which it can be understood, and the ease with which it can be interfaced with conventional transistor circuitry.

The book is organized into two parts, with Chapters 1 through 4 covering the more basic aspects of RTL, and Chapters 5 through 8 dealing with the more exotic RTL applications. Chapter 1 contains elementary nomenclature, and discusses power-supply considerations, mounting, construction practices, etc. Chapter 2 has to do with logic, decoders, logic functions, and the methods of coupling RTL to the outside world. Chapter 3 is on multivibrators; it tells how to build square-wave generators, pulse shapers, astables, monostables, and bistables. The next chapter concerns biasing RTL gates into their amplifying region and building such things as crystal oscillators, operational amplifiers, dc instrument amplifiers, and comparators. This chapter includes "instant-design" charts for speedy amplifier specification. Duty-cycle integration techniques, useful in tachometers and frequency discriminators, are covered in this chapter, also.

Chapter 5 has to do with the JK and Type-D flip-flops, the eight basic JK flip-flop configurations, and some counter techniques. Here, we also look at the input and output restrictions on counting flip-flops, and investigate the techniques essential for reliable and predictable operation. Chapter 6 is on divide-by-n counting and scaling—how to build reliable, frequency-independent, low-cost dividers, decoders, counters, and steppers for any desired count. Decimal counters are given thorough coverage in Chapter 7. The final chapter is on digital instruments. It shows how to tie together the circuits in the rest of the book to build frequency counters, digital voltmeters, electronic stop watches, and other complete digital systems.

I wish to extend my thanks to Billy G. Wood and Rudy O. Nonnenmann for their technical and proofing assistance in putting this book together.

DON LANCASTER

Addendum

It is rare for any technical book written in 1968 to still be in demand—especially a book on integrated circuits. Apparently, we have a classic of sorts on our hands. Your response to this book so far has generated the continuing series of Cookbooks—RTL, TTL, Active Filters, TV Typewriter, CMOS, and the related Users Guide to TTL.

Much of the information in this book is now out of date. While you can still buy and use RTL, bigger and better logic families are now available. (Check into CMOS in particular.) More critically, the problems and thought processes involved when you are working with digital IC's have matured and changed dramatically. We simply don't worry about the same things anymore.

Rather than try to update this text, we've purposely left things as they were. If for no other reason, this leaves us with an untampered historical record of the thought processes and concepts involved with early pre-MSI digital integrated circuit work.

Thanks again for your interest and support of this continuing series.

DON LANCASTER
September 1975

Contents

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 1

Some Basics

A *logic gate* is any device that obeys certain predetermined rules to turn an output off or on upon some coincidence of signals at its input. A kitchen-sink faucet is a logic gate—it provides an output if either the hot or cold input is provided with an on signal. This is an example of an "OR" gate. A garden hose is an "AND" logic gate, because both the outdoor faucet and the nozzle valve must be provided with an on command in order for the output to be turned on.

A *digital logic gate* is a logic gate whose inputs and outputs represent only a "yes" or a "no" command. (That is, either there is a voltage or there is not; things are either on or they are off—there is no halfway condition.) While a digital water source would hardly be useful, there are many advantages to digital logic. Simple elements are called for; they need only remember a "yes" or a "no." And, with enough serial combinations of yes and no outputs, practically any event or number can be represented. The right combinations of digital logic gates, along with some other slightly more complicated logic blocks, make it possible to build anything from an electronic dice game to a television pattern generator.

TYPES OF LOGIC GATES

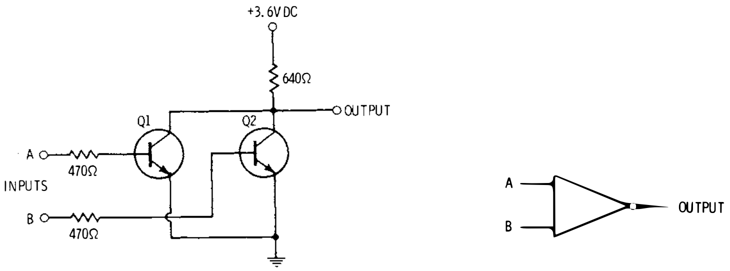
There are many forms of logic gates. They may be mechanical, hydraulic, chemical, pneumatic, optical, electromechanical, or electronic. The electronic logic is by far the most prevalent, because of the great number of gates in use by the computer industry. Integrated circuits (IC's) were called upon about a decade ago to reduce the size, cost, and power consumption of electronic computer gates. As a result, there are many different forms of integrated-circuit digital-logic gates available today. Each of these forms is called a *logic family*, and is usually identified by three letters, such as RTL, DTL, ECL, MOS, TTL, etc. There are many considerations that enter into the choice of logic family for a given application. These factors include cost, speed, complexity, availability, noise immunity, interfacing, and a dozen other more subtle considerations. An integrated-circuit logic family called RTL, short for *R*esistor *T*ransistor *L*ogic, has characteristics that make it suitable for experimental work and applications ranging from simple projects through digital instruments and complex systems.

The advantages of RTL integrated circuits include relatively low price and good availability. Many of these IC's cost less than $1.25, and since most of them are multifunction devices which include several digital logic gates or other logic blocks in one package, the per-function cost can be even lower. RTL is easy to interface with conventional npn transistor circuits, and in many cases it lends itself to *replacement* of conventional circuitry. This is particularly true of the circuits in Chapter 4. Another advantage is that RTL is easy to understand.

Compared to other logic families, RTL is relatively "slow"; it may be used only at speeds less than 10 million counts per second. It is somewhat noisy, requires considerable supply power, and is limited in drive capability. While these limitations are severe to the large-scale computer designer, they tend to be of negligible importance in simple circuits containing only a few IC's. With RTL, the total price usually is less than the cost of doing the same job (and often a poorer one) with conventional transistors.

TWO-INPUT RTL DIGITAL-LOGIC GATE

The work horse of the RTL logic line is the two-input gate. This gate and its logic symbol are shown in Fig. 1-1. (Throughout this book, we will be using a "shorthand" form of logic symbols in which the symbol for a two-input gate is the same anywhere this particular circuit is used.) Fig. 1-1 shows three resistors and two transistors. The transistors are npn and behave exactly like ordinary silicon transistors. This gate is operated, or *conditioned*, by either applying positive voltage to the inputs or grounding



(A) Equivalent schematic.

(B) Corresponding symbol.

Fig. 1-1. Two-input RTL gate.

them. The gate responds by providing either a positive or a grounded output.

RTL is a *saturated* logic family. That is, all the internal transistors are either completely off or conducting the maximum possible current (except for the brief interval during which they switch between these two states). Everything is always either on or off.

If *both* inputs to the two-input gate are grounded, neither base receives current, and both transistors remain off. Since both transistors are off, there is no collector current through either transistor, and thus there can be no current through the 640-ohm collector resistor. There can be no voltage drop across this resistor if there is no current through it, and the output terminal is therefore positive.

If a positive voltage is applied to input A, transistor Q1 saturates (turns on) and draws the maximum possible current through the 640-ohm collector resistor. The entire supply voltage (less a 0.2-volt saturation drop) appears across this resistor, and the output terminal is effectively grounded. The same thing would happen if a positive voltage were applied to terminal B, except that Q2 would saturate and ground the output. If both A and B are made positive, the output still is grounded. To make the output *positive*, *both* inputs must be *grounded.* To *ground* the output, *either* or *both* inputs must be made *positive.*

The two-input gate is used to perform *decision logic* based on the absence or coincidence of input signals. The output from one two-input gate may drive several others directly. If two two-input gates are combined back-to-back, a circuit called a multivibrator is produced; it can be made to have a memory, generate pulses, or oscillate, depending on just how the gates are connected. Further, this particular type of RTL two-input gate can be biased into its class-A region and used in low-cost crystal oscillators, input amplifiers, operational amplifiers, and comparators. Thus, the two-input gate can be made to serve many purposes in the digital-IC world.

OTHER LOGIC BLOCKS

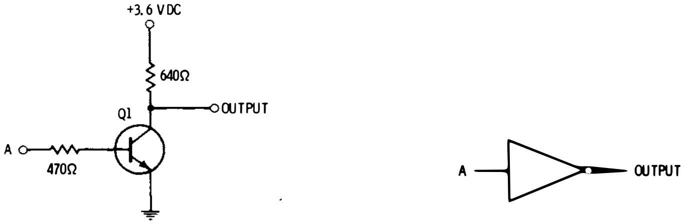
While the two-input gate is the most versatile and widely used RTL digital logic block, there are other available logic blocks which are normally used in combination to produce desired circuit functions with a minimum number of parts. Actually, we could use nothing but two-input gates in any digital circuit, but most circuits would be far more complicated. We now turn to these other RTL logic blocks to see how they function.

Inverters

A *one*-input gate (Fig. 1-2) is called an *inverter*, or a NOT gate. If the input is made *positive*, Q1 receives base current, and the output goes to ground. If the input is grounded, the output goes positive. A positive voltage and a grounded condition are opposites in digital logic. We *must* have one or the other at any logic terminal at any time. These opposites are called *complements*; the complement of a grounded terminal is a positive terminal, and vice versa.

A logic signal is called either a "1" or a "0." A 0 is a complement of a 1. The 0 is not necessarily associated with the grounded-terminal condition; in fact, 0 usually is associated with the *positive* terminal condition, as will be shown later.

An inverter is a useful device for generating the complement of a logic signal, since it automatically produces a 0 if a 1 appears at the input, and a 1 if a 0 appears at the input. An inverter also makes it possible to change the *meaning*, or the *definition*, of a 1. For certain circuits, we may want a 1 to correspond to the grounded input condition, while other circuits may operate only with a 1 corresponding to the positive input condition.



(A) Equivalent schematic.

(B) Corresponding symbol.

Fig. 1-2. One-input gate (inverter).

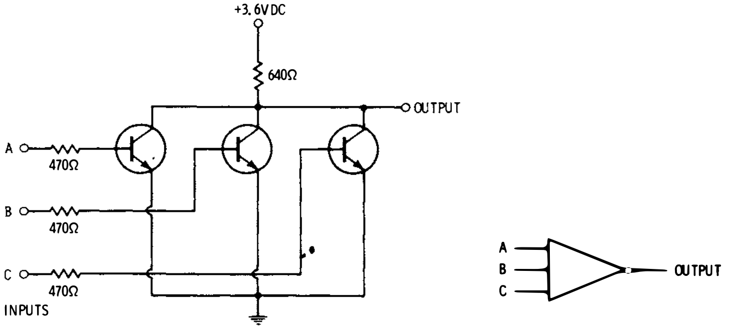
Inverters may be used back-to-back in pairs to form a latch or a memory. They may be combined with a capacitor to form a simple pulse shaper, time delay, or oscillator. They also may be biased in their class-A region for operation as a low-level amplifier or a crystal oscillator.

Gates With More Than Two Inputs

Sometimes we want a circuit to respond logically to a coincidence or absence of more than two inputs. In these cases, we can use a three-input gate, a four-input gate, or any gate with a gate expander attached to provide the necessary number of inputs. The three-input gate is shown in Fig. 1-3. This circuit has three transistors and four resistors, and its operation is just like that of the inverter and the two-input gate. When any input is positive, the corresponding transistor is saturated, and the output is grounded. When all inputs are grounded, the output swings positive.

As with the two-input gate and the inverter, the three-input gate *inverts* the logic, and produces a complementary output. Positive at the input provides ground at the output.

Adding one more resistor and transistor to the three-input gate makes a four-input gate (Fig. 1-4). Here, the output is grounded if any of the four inputs receives base current, and positive only if all four inputs do



(A) Equivalent schematic.

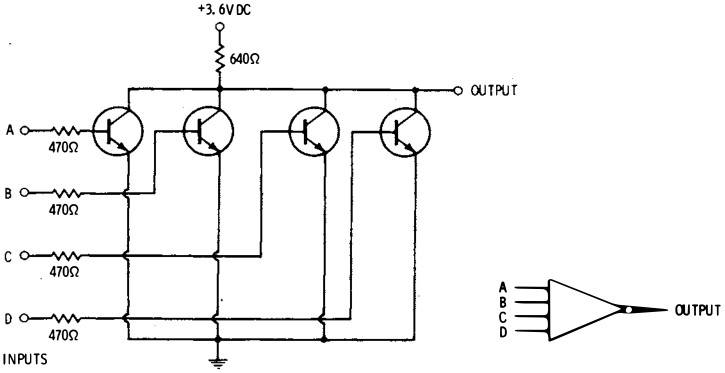
(B) Corresponding symbol.

Fig. 1-3. Three-input RTL gate.

not receive base current. The four-input gate is used exactly as the two- or three-input gate is used, except, of course, that it responds to the logic signals present on all four inputs.

Whenever more than four inputs are needed on a logic gate, a *gate expander* is necessary. This is an IC that provides enough extra input transistors to bring a gate to the desired number of inputs. Two gate expanders are shown in Fig. 1-5 along with their logic symbols. One, two- and four-input gate expanders are readily available. Fig. 1-6 shows how a four-input gate expander may be used with a four-input gate to produce an eight-input gate.

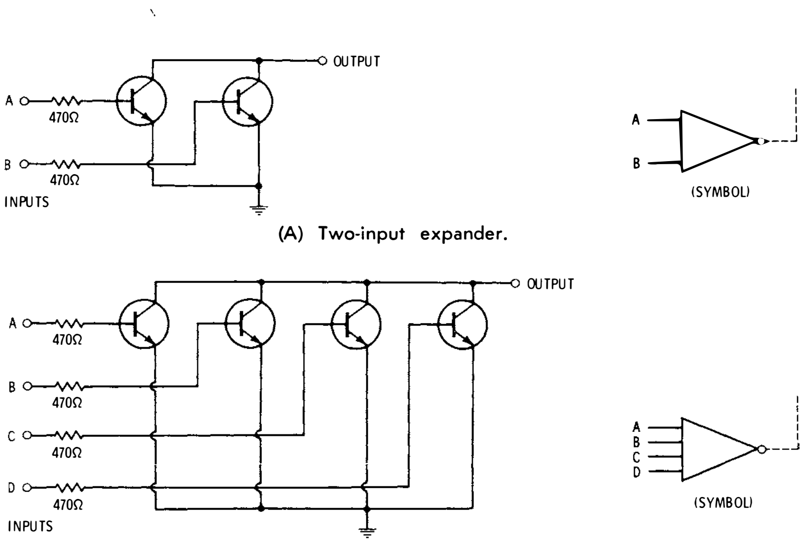
Gate expanders may be doubled up as required to obtain any reasonable number of logic inputs. They are useful when conventional circuitry is interfaced with IC's, and they also may be used to add extra "preclear" and "preset" inputs to certain flip-flops which will be discussed later.



(A) Equivalent schematic.

(B) Corresponding symbol.

Fig. 1-4. Four-input RTL gate.



(A) Two-input expander.

(B) Four-input expander.

Fig. 1-5. Gate expanders.

A four-input gate may be converted into a three-input gate by grounding one input, into a two-input gate by grounding two inputs, and into a one-input gate (inverter) by grounding three inputs. Since several gates are usually available in each IC package, the total number of IC packages needed to do a job often can be reduced. In any RTL circuit, *all unused gate inputs should always be grounded.* Failure to do so can lead to noise problems and erratic operation.

It is also possible to tie two-input gates together at their outputs to produce a single four-input gate, or to use two three-input gates to make a six-input gate. Since this connection alters both the input drive requirements and the output drive availability (it puts both collector resistors in parallel), it should be used with discretion. Connecting more than two gates together at their outputs is not recommended.

Sometimes the effect of another gate input may be gained by selectively applying supply voltage to the entire IC package. This technique is called *strobing* and, in special cases, can produce the effect of an additional gate input on each gate in the IC package.

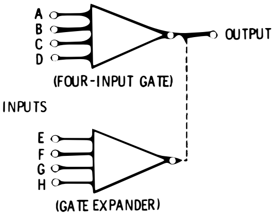
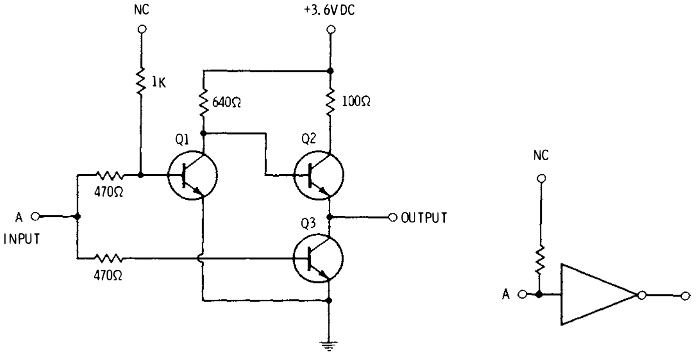


Fig. 1-6. Use of a gate expander.



(A) Equivalent schematic. (B) Corresponding symbol.

Fig. 1-7. One-input inverting buffer.

Buffers

A *buffer* is a higher-power IC used whenever many other IC's have to be driven from a single source, or whenever a more powerful output signal is needed to drive external circuitry.

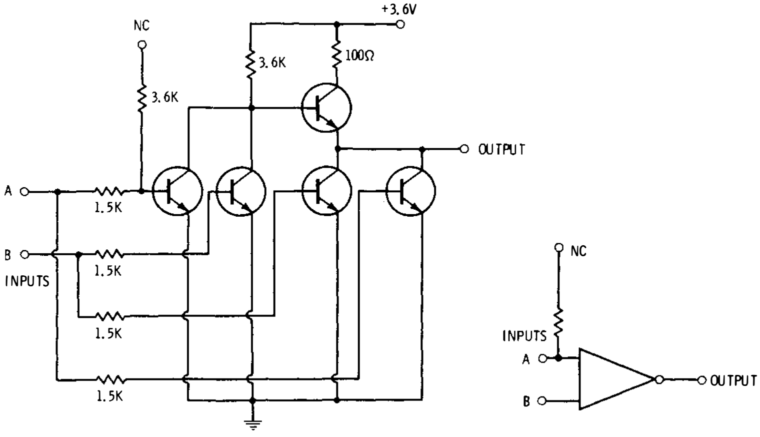
There are several types of buffers. The simplest and most common have one input and invert the signal; thus they are nothing but high-power NOT gates, or inverters. This type of buffer is shown in Fig. 1-7. If the input is grounded, Q1 and Q3 remain off. Transistor Q2 then receives base current through the 640-ohm resistor, and the output terminal swings positive. If the input is made positive, base current reaches Q1 and Q3, and Q3 turns on, grounding the output. Since Q1 is on, Q2 receives no base current and stays off. The output stage actually operates in a push-push manner, forcing the output terminal to either a positive voltage or ground, following the commands of an input signal.

The 1k resistor shown is internal to the buffer and is brought out to a separate pin; this resistor is normally left unconnected. In Chapter 3 it will be shown that by connecting this resistor to the positive voltage source and by capacitor-coupling the input, the buffer may be made into a relatively high-power pulse generator, useful for waveform shaping and generating reset pulses in IC systems.

There are fancier forms of buffers that have more than one input. These allow input logic to be performed as well as providing a high-power output. A two-input inverting buffer and a three-input noninverting buffer are examples of these fancier logic blocks. They appear in Figs. 1-8 and 1-9, respectively.

JK Flip-Flop

The JK flip-flop is a more complex IC logic block and is useful for performing binary division, storage, counting, scaling, and other more com-

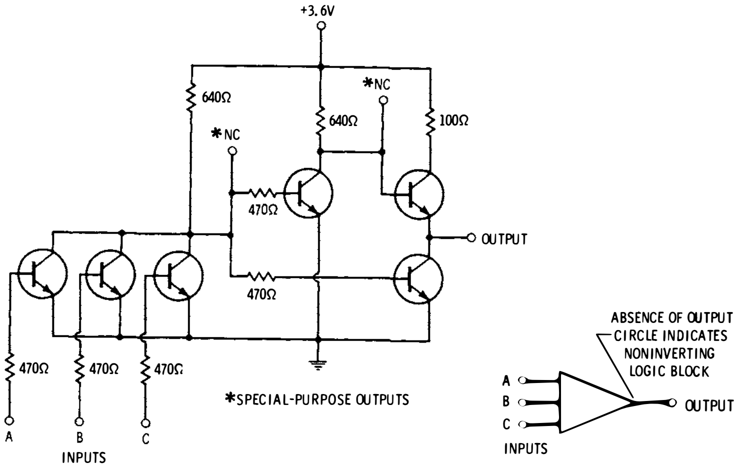


(A) Equivalent schematic.

(B) Corresponding symbol.

Fig. 1-8. Two-input buffer.

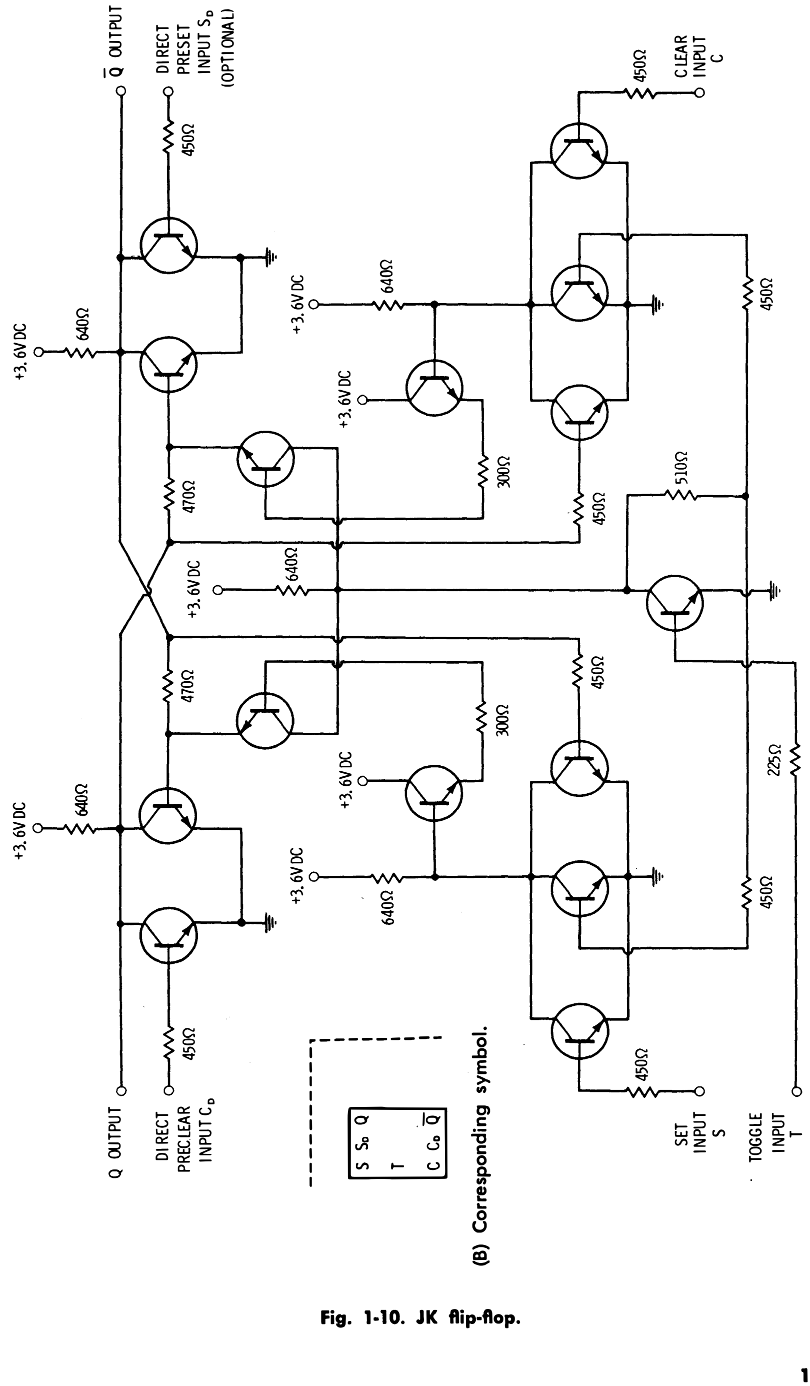
plex forms of logic. It can divide by two without the need for other logic elements. JK flip-flops have four and sometimes five inputs, called the *set*, *toggle*, *clear*, *preclear*, and *preset* inputs, and two complementary outputs called the "Q" and "$\bar{Q}$" outputs. The schematic and logic symbol of the JK flip-flop are shown in Fig. 1-10. Details of this particular logic block are contained in Chapter 5.



(A) Equivalent schematic.

(B) Corresponding symbol.

Fig. 1-9. Three-input noninverting buffer.



(A) One equivalent schematic.

(B) Corresponding symbol.

Fig. 1-10. JK flip-flop.

The essential difference between the JK flip-flop and the simpler logic blocks is that a JK flip-flop runs in a *clocked* mode. Regardless of what is done to the set and clear inputs, nothing happens until the instant the signal on the toggle input *abruptly drops* from a positive voltage to ground. We say that the JK is *edge sensitive* to the *negative-going toggle transition.* This circuit action allows us to "set up" what the JK flip-flop is supposed to do *independently* of *when* it actually does it. This feature is particularly useful in shift registers, decimal counters, and other counter circuits.

Type-D Flip-Flop

The Type-D flip-flop is another version of a clocked logic block whose input commands, or *conditioning*, may be set up prior to the actual performance of the logic. Externally, the Type-D flip-flop is very similar to a JK flip-flop, except that it lacks a clear input. It is used primarily for shift registers, arithmetic calculations, and storage circuits. The schematic and logic symbol for the Type-D flip-flop are shown in Fig. 1-11. Details of this device are contained in Chapter 5.

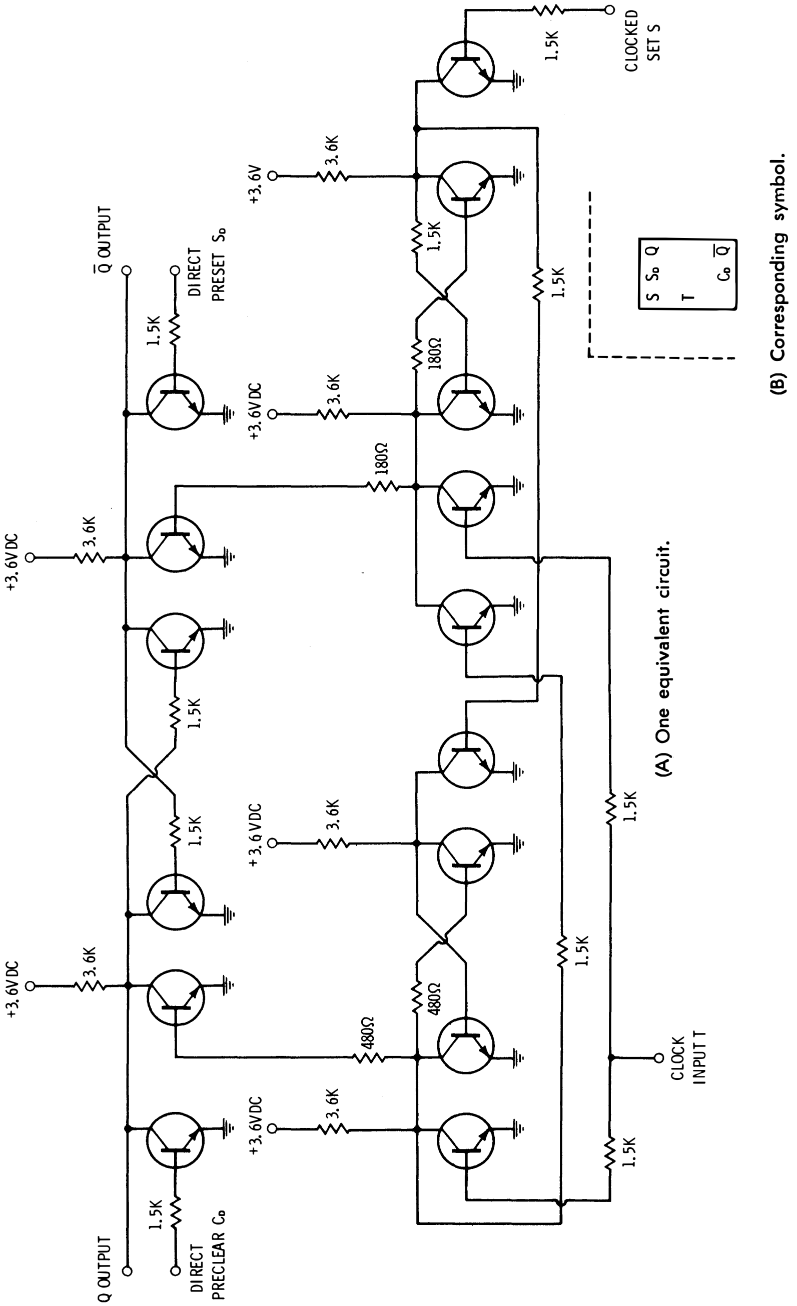
MULTIPLE-FUNCTION IC'S

Most of the available RTL integrated circuits combine two or more logic blocks in the same package, thus reducing the total number of packages needed for any particular circuit, and making maximum use of all available pins on the IC package. For instance, a dual two-input gate is an IC package with two independent two-input gates in the same package. Similarly, a quad two-input gate contains four independent two-input gates in the same package, and a dual four-input gate has two independent four-input gates in the same package. The only common connections between the gates are the positive source and ground. All the gates in a single IC may be used either together or in totally unrelated circuits. There is no crosstalk problem.

Other popular configurations are dual flip-flops, dual buffers, hex inverters (six to a package), and triple three-input gates. Some special combinations are also available, such as a JK flip-flop, an expander, and two buffers. These are particularly useful for reducing the total number of IC packages used in specialized applications.

POWER LEVELS

There are currently two types of RTL available, the *medium-power* RTL and the *milliwatt*, or *low-power*, RTL. These two types usually are similar in cost and differ primarily in the internal power dissipation. The internal power consumption is four to five times greater in the medium-power RTL than in the low-power RTL. For instance, the two-input gate of Fig. 1-1 is a medium-power gate. It has a 640-ohm collector resistor and dissipates



(A) One equivalent circuit.

(B) Corresponding symbol.

Fig. 1-11. Type-D flip-flop.

about 25 milliwatts with both inputs positive. The equivalent low-power-RTL two-input gate has a 3.6k collector resistor and dissipates about 5 milliwatts with both inputs positive.

Low-power operation is never obtained free. The low-power IC's have restricted operating speeds and very restricted drive capabilities. Often a circuit made entirely from low-power IC's requires more packages, since extra buffers may be needed to obtain sufficient drive levels. The choice of low-power versus medium-power RTL depends on the complexity of the circuit and the available power supply. The usual choice is to use the medium-power RTL and save the low-power units for use when special design problems occur.

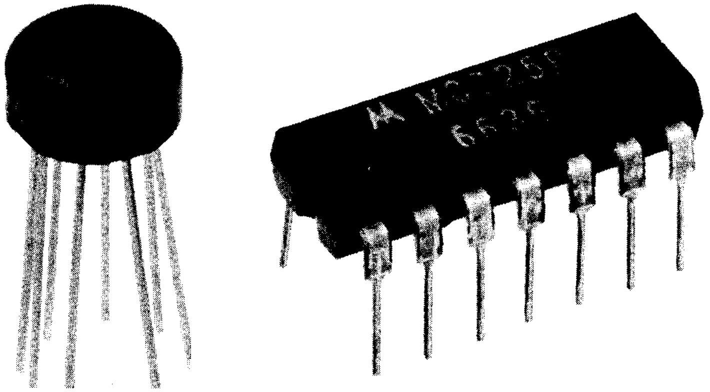
LOADING: FAN-IN AND FAN-OUT

All the transistors in the RTL logic blocks operate either in the saturated or cut-off mode; that is, the equivalent transistors are either completely conducting or completely cut off during steady-state operation. It is reasonable to expect that there is some minimum input drive level needed to provide enough base current to keep the input transistors saturated; similarly, there must be a maximum available output current for any IC. The input current requirement is called the *fan-in* of the circuit, and the output drive capability is called the *fan-out* of the circuit.

Rather than worrying about delivering so many milliamperes into a transistor with such-and-such gain, a much simpler method is used in integrated-circuit work. The manufacturer investigates the worst-case performance of his logic line and then assigns a *loading number* to each terminal. A loading number of 3 on an input means that 3 units of drive are *required.* A loading number of 15 on an output means that 15 units of drive are *available.* We can load an output with any combination of units less than or equal to the available output. For instance, one type of JK flip-flop has a fan-out of 10. This output can be used to drive any combination of gates that total 10 units or less. Three medium-power gates would require 3 units of drive each, for a total of 9; we may safely connect these to the 10-unit output. If we added a fourth gate of fan-in 3, we would be trying to get 12 units from an output with a fan-out of 10, and the circuit most likely would not perform properly.

The fan-out for any IC should not be exceeded. This is particularly true of the low-power RTL. The fan-in and fan-out numbers for the low-power and the medium-power RTL are compatible. Thus, a low-power gate with a fan-out of 3 can drive a medium-power gate with a fan-in of 3.

When the available fan-out is not enough, it can be "amplified" by an additional gate, inverter, or buffer. For instance, a low-power gate amplifies a 1 to a 3; a medium-power gate or inverter amplifies a 3 to a 16; a buffer amplifies a 5 to an 80. Buffers are particularly handy whenever several logic devices are to be driven in parallel. As an example, if ten flip-flops are to



(A) TO-5 package. (B) In-line package.

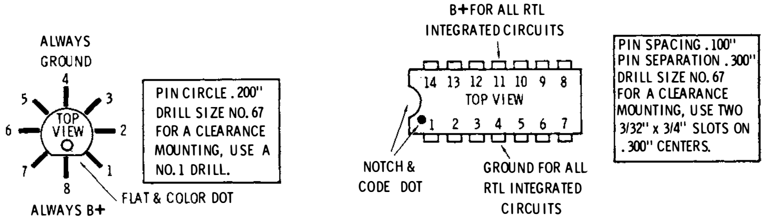
Fig. 1-12. RTL integrated-circuit packages.

be reset simultaneously, 30 units of drive are required; if they are to be toggled simultaneously, 50 drive units are needed.

There are two factors that may have to be considered when it is desired to amplify the fan-out. Almost all gates and buffers invert the input and provide its complement at the output. Inversion in the final output can be avoided either by picking an input complementary to the desired one or by reinverting the output. A second possible problem is time delay introduced as the signal goes through a gate or an inverter. While this delay may be as little as 12 nanoseconds (billionths of a second) or so, it may become important in high-speed systems.

PACKAGES AND LEAD CONVENTIONS

Modern integrated circuits come in a wide variety of packages, but the low-cost industrial RTL versions are mostly limited to one of two multilead molded plastic packages, an eight-lead, TO-5 round package and a rectan-



(A) TO-5 package. (B) In-line package.

Fig. 1-13. IC pin information.

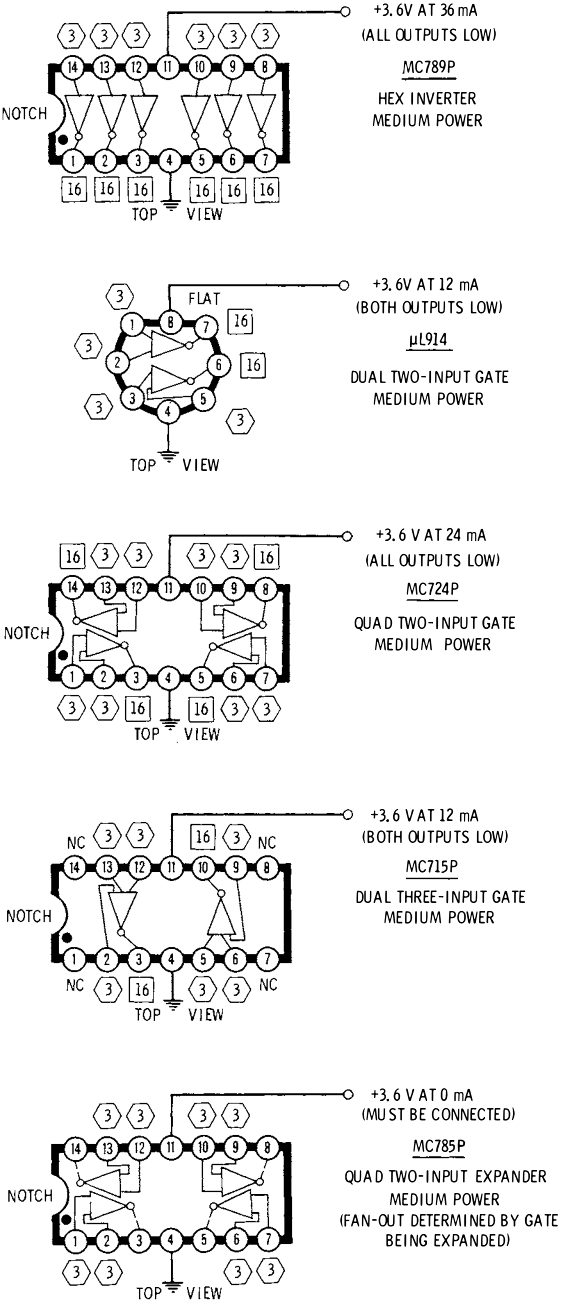
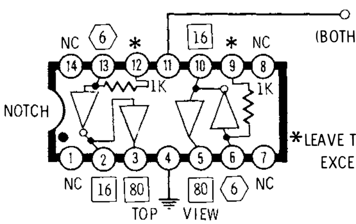


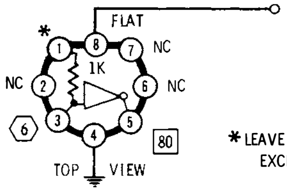
Fig. 1-14. Some available



+3.6V AT 72 mA
(BOTH OUTPUTS HIGH & LOADED)

MC799B

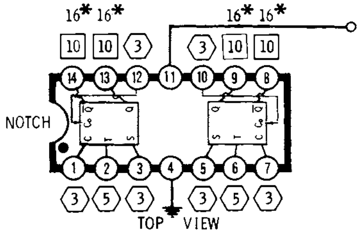
DUAL BUFFER
MEDIUM POWER
*LEAVE THIS TERMINAL UNCONNECTED
EXCEPT FOR PULSE GENERATOR



+3.6V AT 36 mA

μL900

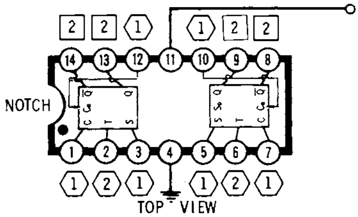
BUFFER
MEDIUM POWER
*LEAVE THIS TERMINAL UNCONNECTED
EXCEPT FOR PULSE GENERATOR.



+3.6V AT 44 mA
32 mA*

MC790P & MC791P*

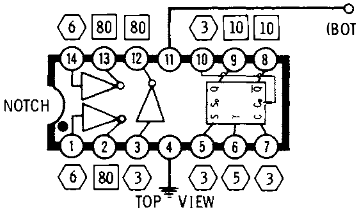
DUAL JK FLIP-FLOP
MEDIUM POWER



+3.6V AT 12 mA

MC776P

DUAL JK FLIP-FLOP
LOW POWER



+3.6V AT 100 mA
(BOTH BUFFERS HIGH & LOADED)

MC787P

JK FLIP-FLOP, INVERTER,
AND TWO BUFFERS

NOTE: POWER REQUIREMENTS ARE WORST CASE.

□ = FAN OUT

⬡ = FAN IN

RTL integrated circuits.

gular dual in-line package with 14 leads. These packages are shown in Fig. 1-12.

Both packages have their leads identified counting counterclockwise from the top, and most manufacturers use this top-view convention because it makes the schematic, layout, and logic diagrams nearly identical. From a practical standpoint, this uniformity simplifies circuit-board layout and troubleshooting.

The TO-5 package has pin 8 identified by a flat, a tab, or sometimes a color dot on the package; the count proceeds counterclockwise around the package when viewed from the top. The dual in-line package is identified by a dot beside pin 1. The count then proceeds down one side and back up the other side, returning to pin 14 directly above pin 1. A code notch between pins 1 and 14 also aids identification. These numbering conventions are shown in Fig. 1-13.

The supply-voltage connections are always the same for a particular package. On the round, eight-lead package, pin 4 is always ground; pin 8 is always B+. The 14-lead rectangular package always has lead 4 grounded and lead 11 connected to B+. This is true of all RTL IC's but is *not* true of other (non-RTL) logic families that may be supplied in the 14-lead rectangular package.

RTL is offered in several lines, differing in temperature performance and tolerance limits. For experimental work, either the hobby/experimental or the "industrial" grade RTL normally is used. These types are relatively low in cost and are the most readily available. They have a specified operating range of +50 to +130°F. Fig. 1-14 gives details on IC's referred to in the circuit descriptions in this book. The integrated circuits of Fig. 1-14 are stocked by a number of electronic distributors.

The prices of IC's vary from unit to unit. The package price might be on the order of from one to three dollars in single quantities, depending on the complexity of the internal circuitry.

Always obtain a complete set of data sheets before you begin any IC work. These sheets serve as a road map of what can and cannot be done with integrated circuits, what their temperature restrictions are, their fan-in and fan-out, their power requirements, and their pin connections.

MOUNTING TECHNIQUES

There are several practical ways to mount RTL IC's. Those of greatest interest to the technician and the economy-minded experimenter are sockets, standoffs, and PC methods.

Sockets

Sockets are handiest when the IC's have to be reused, but they are often expensive, and the close lead spacing often requires some fancy and fine wiring work. Sockets for IC's are usually a distributor stock item, and may vary in price from less than $1 to $12 or more each. Usually the more expensive ones are made of *Teflon* and are intended for miliary use. For simple systems, the cost of sockets can be a significant portion of the total project.

Standoffs

One simple way to mount an IC is to bend the leads radially outward and solder them to a grouping of eight or fourteen *Teflon*-insulated standoffs pressed into a glass-fiber or phenolic board. This method spreads the leads for easy wiring or testing, and it makes the IC easy to change. Fig. 1-15 shows this mounting technique. Both the standoffs and the IC's are easily reusable.

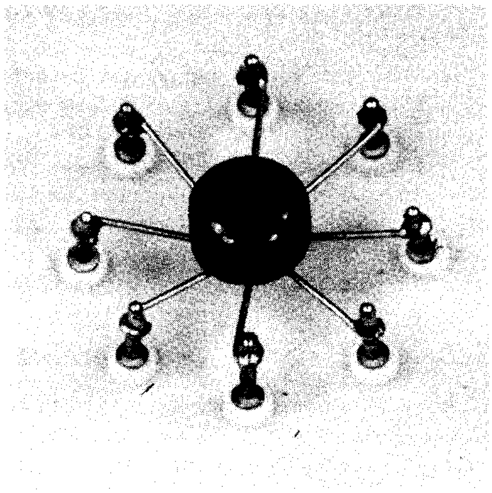


Fig. 1-15. IC mounted with insulated standoffs.

If the IC's are mounted right side up, the pin count will proceed counterclockwise, and the final wiring should correspond to the data-sheet numbering.

Printed Circuit Boards

IC's were designed for permanent insertion into printed circuit boards of the multilayer type. For simple circuits and digital instruments, single-sided PC boards with jumpers are an attractive alternate. Kits are available that permit making your own PC boards, using either the acid-resist or the photographic method.

Also, it is practical to use a "universal" PC master pattern with a grouping of IC land patterns that has only B+ and ground connected; connections can be made to the other terminals as necessary. These boards are commercially available and are reusable.

TOOLS

All of the conventional electronics hand tools carry over for IC work, although the smaller 4½-inch "jeweler's" diagonal cutting pliers and needle-nose pliers are preferable to the larger "electronic" pliers. Several additional tools are necessary for IC work:

1. Syringe-type desoldering tool for removing IC's and correcting wiring errors.
2. Wire cleaner for desoldering tool.
3. Pocket magnifier for inspecting solder connections.
4. Knife for cutting circuit-board connections, separating solder bridges, and similar uses.
5. Toenail clipper for close cutting of soldered leads on printed circuit board.
6. Machinist's scriber for general probing.
7. Small vise or other way of holding circuit boards securely for close-in work.
8. Hand drill or miniature drill press. Most IC's require a No. 67 hole for each lead; larger components require a No. 60 hole. Drills of these small sizes cannot last long in a hand-held ¼-inch electric drill, even if the chuck will close down tightly enough to center them.
9. Pin vise and No. 67 drill for PC-hole cleaning and rework.

SOLDERING

Always use a small soldering iron rated at 40 watts or less and very fine solder when soldering IC's in place. While the IC's themselves can withstand any reasonable amount of heat, problems with solder bridging, foil lifting, and poor joints increase when larger irons are used. Soldering guns are awkward for this kind of work and should not be used. A temperature-controlling iron stand is a handy accessory if a great deal of IC work is to be done.

Be certain to double-check the IC connections before soldering. It is so easy to put the IC in upside down, to rotate it a lead or two, or worse yet, to put an IC in the wrong place.

To remove an IC, use a syringe-type desoldering tool, and carefully remove *all* the solder from each lead, one at a time. Then carefully pry up the IC. It should pop loose safely with no trouble at all.

POWER REQUIREMENTS

All the IC's of Fig. 1-14 are rated to operate from a 3.6-volt dc supply. In reality, most gate-only circuits will operate properly from a supply of 1.5 to 4.5 volts dc, and the more critical circuits using flip-flops will operate from a supply of 3.1 to 4.1 volts dc. This voltage is easily obtained from batteries or from a low-voltage, high-current line-operated supply.

Whatever power source is used, it is absolutely essential that proper high-frequency bypassing be provided for the IC's. This bypassing may be accomplished by placing a 0.1-μF and a 100-μF capacitor in parallel at the IC end of the power-supply runs. The shortest possible leads should be used (Fig. 1-16). It is also important to use large leads (or wide foil runs) for both the B+ and ground connections to minimize noise effects.

The choice of batteries versus line operation depends on how many IC's are to be used, how portable the finished circuit must be, and whether the extra expense of a low-ripple power supply is justified. The first step in this decision is to calculate the current and power requirements.

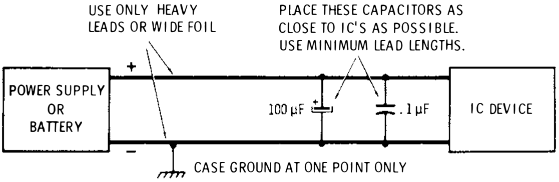

USE ONLY HEAVY LEADS OR WIDE FOIL
PLACE THESE CAPACITORS AS CLOSE TO IC'S AS POSSIBLE. USE MINIMUM LEAD LENGTHS.
+
POWER SUPPLY OR BATTERY
100 µF
.1 µF
IC DEVICE
−
CASE GROUND AT ONE POINT ONLY

Fig. 1-16. High-frequency bypassing of power leads.

CURRENT REQUIREMENTS

Table 1-1 shows the approximate average current drain for each logic block. To obtain the total current requirement, just add the currents for the individual elements. For instance, suppose we design a circuit that has three dual flip-flops, one dual buffer, and one dual two-input gate, all of which are medium-power RTL. The total current will be:

6 flip-flops at 22 mA	132 mA
2 buffers at 42 mA	84 mA
2 gates at 6 mA	12 mA
Total	228 mA

Thus, our supply would have to provide about 228 mA. To be on the safe side, we would design around a 250-mA supply. The total power requirement would be equal to the supply voltage multiplied by the supply current, or in this case 3.6 volts × 228 mA = 820 milliwatts, or, nominally, about one watt.

Note that a large number of flip-flops or buffers always means a large supply current. Thus the more complex IC circuits will be quite difficult to power with batteries.

BATTERY OPERATION

Ordinary flashlight cells, either two or three to the holder, may be used with many IC circuits, provided they are properly bypassed, and provided

Table 1-1. Current Requirements for Logic Blocks

LOGIC BLOCK	CURRENT (mA)*
Medium-Power RTL	
Inverter	6
Gate	6
Buffer	42
JK Flip-Flop	22
Expander	0
Low-Power RTL	
Gate	1
Buffer	9
JK Flip-Flop	6
Type-D Flip-Flop	6

*With 3.6-volt dc supply (approximate values)

they can supply enough current for a long enough time. If the batteries are to last longer than about five hours, the maximum current drains listed in Table 1-2 should not be exceeded. Note that for circuits that take more than about 150 mA, the heavy-duty D cells or the more expensive alkaline cells must be used. For more details on battery life versus current drain, consult any of the handbooks published by many of the leading battery manufacturers.

Table 1-2. Maximum Current Drains

Type of Cell	Current (mA)
Flashlight "AA" (carbon-zinc)	30
Flashlight "C" (carbon-zinc)	90
Flashlight "D" (carbon-zinc)	150
Heavy-Duty "D" (carbon-zinc)	200
Heavy-Duty "D" (alkaline)	300

One way to extend the life of the larger cells is to add a two-diode "battery-saver" selector switch, as shown in Fig. 1-17. When the switch is in the FRESH position, the battery will measure 4.5 volts, and the two 0.6-volt forward drops of the silicon diodes will produce a 3.3-volt output. As the battery ages, the switch is changed first to the NORMAL position, and finally to the OLD position, retaining about 3.3 volts at the output, even with the cells discharged to 1.1 volts each. With this technique, battery life may easily be doubled.

Another, somewhat expensive, alternative is to use rechargeable nickel-cadmium cells. Three 1.2-volt D cells nicely add up to 3.6 volts, and may be recharged many times. The 2000-mA-hr/200-mA-rate cell is suitable for many large projects.

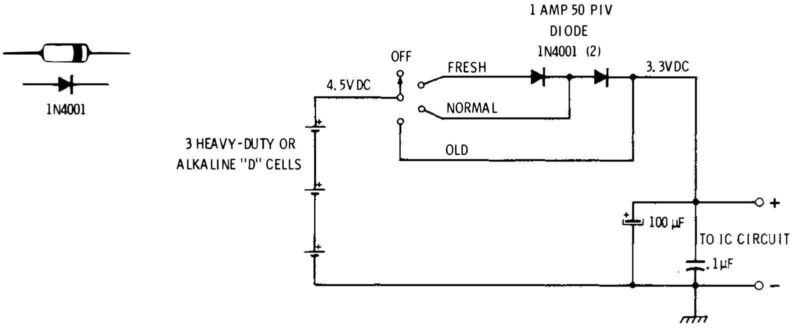


Fig. 1-17. "Battery saver" circuit.

When indicator lamps are used with the IC's, their current requirements must also be taken into account. Dry-battery operation is usually impractical in any circuit with more than three indicator lamps in addition to the IC load.

LINE OPERATION

Ac-line operation requires immense electrolytic capacitors, a regulator, or both to obtain the low ripple at high current levels required for IC work. The elegance of the required supply increases with the current requirements and the complexity of the actual IC circuit.

For a circuit using IC's only and no indicators, with a total current drain of less than 500 mA, the brute-force filtered supply of Fig. 1-18 may be used. The circuit includes a conventional filament transformer, two silicon diodes, and a large-value electrolytic capacitor. The immense capacitance is required in order to provide a low ripple at high current levels. Computer-grade electrolytic capacitors of this size are priced in the under-five-

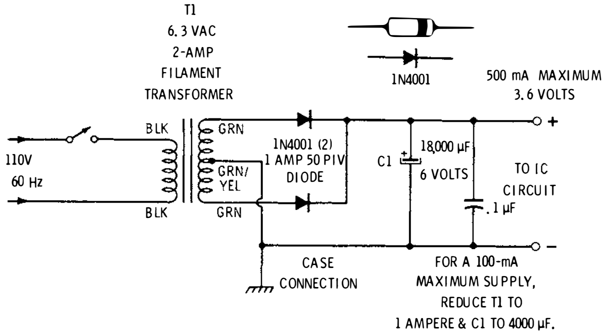


Fig. 1-18. Filtered dc power supply.

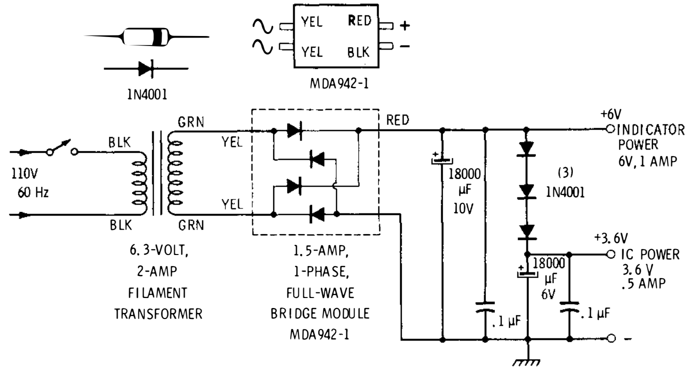


Fig. 1-19. Dual-voltage dc supply.

dollar range and measure about 1½ inches in diameter by 5 inches long.

A two-voltage, brute-force filtered supply is shown in Fig. 1-19. Here, 3.6 volts dc at 500 mA or less and 6 volts at 1 ampere are provided. This type of supply is ideal for IC projects which include indicator lamps.

Fig. 1-20 shows a regulated supply that uses relatively small-value electrolytic capacitors and that will provide 3.6 volts at 1 ampere. For applications requiring higher current, this supply may be "beefed up" with larger rectifiers and filters, and a bigger power transistor.

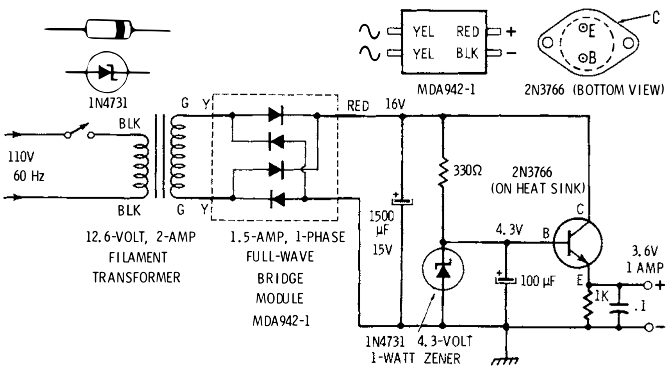


Fig. 1-20. Regulated dc power supply.

"BAD" AND "BURNED OUT" INTEGRATED CIRCUITS

There are countless examples of IC work in which, if the circuit did not operate on the first try, the IC's were immediately blamed and considered "bad" or "burned out." In reality, RTL IC's are almost indestructible, and the likelihood of getting a defective new one is very slight. The IC's are neither heat nor static sensitive and will withstand considerably more than normal soldering heat. So long as circuit voltages are less than 5 volts or so, the IC will withstand virtually any combination of shorted or reverse-polarity connections without failure.

The real gremlin is not a bad IC, but sloppy or careless workmanship—solder bridges between connections, poor soldering, reverse supply polarity, a PC layout error, the wrong IC in the socket or the IC in upside down or rotated one pin, a PC board layout done bottom-view and the IC's inserted top-view (reversing all the connections), or the omission of high-frequency supply bypassing. These are the real problems in most IC circuits and account for practically all circuit problems. Be sure to watch for them.

CHAPTER 2

Logic and Switching Circuits

Digital-logic gates often are used as high-speed electronic switches. They also are used to respond to the coincidence or absence of input signals to follow a set of logic rules. In this chapter, the logic gate as a switch will be emphasized—how to use it, what rules to follow, and how to design any desired switching function with digital-logic gates.

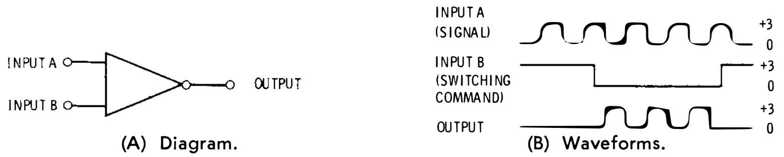
THE TWO-INPUT GATE AS A SIMPLE SWITCH

Suppose we wish to turn a train of pulses on and then turn them off again under command. For instance, we might like to allow high-frequency pulses to enter an electronic counter for precisely one second. The *readout* of the counter would then indicate the total number of pulses present during the second, or the frequency of the input in pulses per second. To do this, a two-input gate may be used as in Fig. 2-1.

So long as the switching waveform (input B) is positive, the output of the two-input gate will stay grounded. If terminal B is grounded, the two-input gate will respond to the signal waveform (input A), and an inverted replica of input A will appear at the output. Pulses will appear at the output *only* when input B is grounded. Thus the gate serves as a simple electronic switch.

If we would rather control the switch with a positive-going waveform, we may simply add an inverter to input B. This circuit is shown in Fig. 2-2. Here, the input signals at A are passed and inverted only if terminal B is positive; the signals are blocked if terminal B is grounded. In Fig. 2-3, an inverter has been added to input A also. Now, pulses are passed "right side up" only when input B is positive.

This circuit could be built with a dual two-input gate and two inverters, or with a quad two-input gate by itself. If the one-package quad two-input



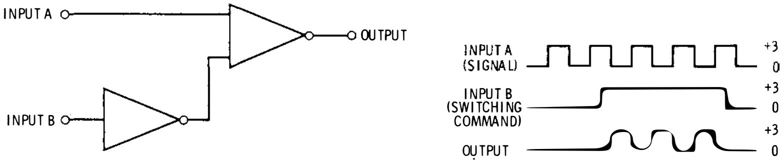
(A) Diagram.

(B) Waveforms.

Fig. 2-1. Two-input gate used as switch.

gate is used, two inputs are grounded to give the required two inverters, one gate is used "as is," and one is left over for use elsewhere.

This form of electronic switch is useful when it is desired to turn a digital signal on or off electronically, or when it is desired to *sample* a portion of a digital signal. RTL electronic switches may be operated at frequencies from dc to 10 MHz, and at speeds between 50 nanoseconds and many hours.

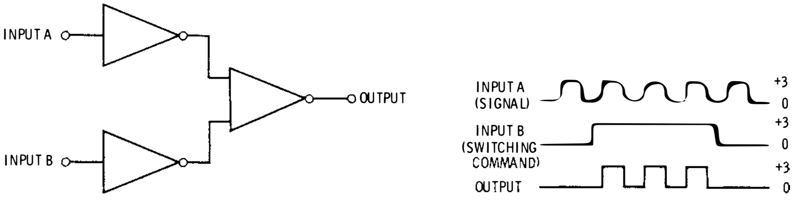


(A) Diagram.

(B) Waveforms.

Fig. 2-2. Electronic switch for positive input.

There are some features of the simple electronic switches discussed so far that might not be desirable in some special applications. The switch may be turned on either just before or just after the instant a pulse arrives on input A. If we watch the switch output from sample to sample, we would obtain a roundoff error of plus and minus one count. In an electronic counter, this would make the last digit fluctuate. Also, the switch might be turned on *during* an input pulse; the output would then contain part of a pulse. If the time *width* of each pulse at input A is critical later on in the circuit, the switching will introduce a timing error. To get around these problems, it is necessary to resort to a *synchronized* electronic switch, a slightly more complex circuit described in Chapter 5.



(A) Diagram.

(B) Waveforms.

Fig. 2-3. Noninverting electronic switch.

COMPUTER LOGIC

The rate at which signals appear at either input A or B has nothing to do with the switch operation. There could just as easily be a long gating waveform on terminal A and a series of short pulses on input B, or the pulses appearing at A and B could have nearly the same time duration.

In the latter case, the two-input gate responds to the coincidence or absence of the input pulses. For instance, if either input is made positive, the output goes to ground. If both inputs are grounded, the output goes positive. We say that the output goes positive on the *coincidence* of grounded input signals, and it goes to ground on the presence of either input positive. This type of circuit crops up over and over again in computers, and a whole set of rules, called *computer logic*, has been established to allow you to design any circuit you want that will respond to the presence or coincidence of inputs in any desired manner. We cannot change what the two-input RTL gate does—its output *always* goes positive on coincident grounded inputs, and always goes to ground with either input positive. What we can do is *interconnect* enough gates and inverters in a more complex circuit to get the desired result.

ONES AND ZEROS

So far, we are dealing only with digital input signals. A gate input must be either positive or grounded. In other words, any gate can have only one of two possible *states* on one of its inputs; similarly, at its output it can produce only one of two possible states. To identify these two possibilities, one state is called a *one* and the other a *zero.*

Now, we have a choice: Do we call the grounded state a zero and the positive state a one, or do we call the grounded state a one and the positive state a zero? It all depends. It is standard to assign the designation one to *something we want* and the designation zero to the *absence of something we want.* Rules and names for the rules are established *only* on what combination of ones at the inputs is required to get a desired output. Then, the definition of what a one is (grounded or positive) is chosen to "fit" the easiest possible way to make the circuit using RTL.

Logic functions are arranged according to what we want the ones to do. A suitable definition of a one is chosen to get the two-input RTL gates to give the desired results. Before we do this, however, it is necessary to take a closer look at the actual computer logic functions.

THE NOT CIRCUIT

A NOT gate is a one-input gate, or inverter. This circuit always automatically converts a one to a zero and vice versa, regardless of our definitions. This process is called *generating the complement.* Sometimes the more complex logic blocks have both an output and the complement of that output already available. If this is the case, the NOT circuit is already built in, and it is only necessary to pick the proper output to do the job.

People who work with computers have a way of building up a chart that illustrates how each logic function works. To do this, they simply list all possible combinations of inputs and then indicate for each combination the only permissible output that fits the logic function they are trying to generate. Such a chart for the NOT logic function is made up as follows: The chart is begun by listing all possible conditions on all possible inputs. With a NOT function, there is only one input, and it can only be a 1 or a 0 (Fig. 2-4A). The chart is completed by listing the outputs the NOT function produces: one produces a zero, and zero produces a one (Fig. 2-4B).

INPUT	OUTPUT
1	
0	

(A)

INPUT	OUTPUT
1	0
0	1

(B)

Fig. 2-4. Logic chart for NOT function.

TWO-INPUT LOGIC FUNCTIONS

There are a number of two-input logic elements, each of which has a descriptive name. Just what each circuit does is most easily shown with the aid of charts similar to the one just developed for the NOT function.

OR Circuit

The OR circuit provides a 1 output if *either* input is a 1. The chart becomes a bit more complicated; now there are two inputs with two possible states on each input, giving four possible combinations. The chart for this logic function is shown in Fig. 2-5. This type of circuit is used whenever it is desired that *any* input produce an output.

NOR Circuit

NOR is shorthand for NOT OR. This circuit gives a 0 output if *either* input is a 1, and its chart looks like Fig. 2-6. The NOR circuit may be used any time it is necessary that *any* input *block* an output.

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	1

Fig. 2-5. Logic chart for OR function.

Fig. 2-6. Logic chart for NOR function.

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	0

AND Circuit

The AND logic function gives a 1 output only if *both* inputs are 1's (Fig. 2-7). This circuit is quite popular. It is used any time we want only the *coincidence* of input 1's to produce an output.

Fig. 2-7. Logic chart for AND function.

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	0
0	1	0
1	1	1

NAND Circuit

NAND is shorthand for NOT AND. This logic function gives a 0 output *only* if *both* inputs are 1's (Fig. 2-8). The NAND function *blocks* an output only when coincident 1's appear at its input.

Fig. 2-8. Logic chart for NAND function.

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	1
0	1	1
1	1	0

Half Adder (EXCLUSIVE OR) Circuit

The half adder, or EXCLUSIVE OR, logic function gives a 1 at the output if a 1 is present at *either* input, but *not both* (Fig. 2-9). What makes this logic function so useful is that it obeys the laws of *binary* (or base-two) *addition.* This provides the basic means for a computer to add two numbers together.

Fig. 2-9. Logic chart for EXCLUSIVE OR.

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	0

In the base-two number system, there are only two digits, 1 and 0. The binary addition rules are:

$$0 + 0 = 0$$
$$1 + 0 = 1$$
$$0 + 1 = 1$$
$$1 + 1 = 0 \text{ and carry } 1$$

Just as in base-ten arithmetic, we add one column of two numbers together, and, if there is a carry, add it to the next column to the left. Thus, decimal 2 in the binary system is 0010; decimal 6 is 0110; and their sum is 1000, or decimal 8.

The reason that the EXCLUSIVE OR is called a half adder is that a carry may be left over from the previous column. A first half adder must add this possible carry to the first of the two numbers to be added; a second half adder must then add this *result* to the second number in each column. Two cascaded EXCLUSIVE OR circuits are called a *full adder*, and are the key to practically all computer calculations.

Once we have addition, we can add repeatedly to perform multiplication. If definitions are changed slightly to borrow instead of carry, we can subtract. And repeated subtraction is division. More complicated operations can be carried out easily by repeated sequential combinations of the basic addition-subtraction process.

Correlator, or EXCLUSIVE NOR

The correlator circuit provides a 1 output if the inputs are identical, and a 0 output if the inputs are different (Fig. 2-10). The correlator logic function is used to *verify* that two binary numbers are identical, or to check to be sure that the same thing is happening at two different places in a circuit at the same time. This configuration is particularly handy for error detection and for electronic recognition, code, and coincidence circuits.

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	1

Fig. 2-10. Logic chart for EXCLUSIVE NOR.

Other Two-Input Logic Functions

There are 16 possible two-input logic functions if a function is designed arbitrarily to fit every possible combination of 1's and 0's on the two-input chart. Six have just been discussed. The remaining ten functions are either worthless or so specialized that they would rarely, if ever, be used.

MULTIPLE-INPUT LOGIC

There are 256 possible logic functions with a three-input gate, 65,536 with a four-input gate, and so on. Generally speaking, the six functions already talked about for the two-input gate are often the only ones that are needed or that are really useful.

The rules carry over directly. On a three-input gate the OR function provides a 1 at the output for a 1 at any input; the AND function must have a 1 at each and every input to produce a 1 at the output, and so on.

BUILDING LOGIC FUNCTIONS WITH RTL

How do we actually *build* these logic circuits using RTL integrated gates? Remember, the only thing an RTL gate can do is provide a grounded output when any input is positive, and a positive output only if all inputs are grounded. The first step is to decide what a 1 is to be.

Suppose our definition says a 1 is the grounded state. The six two-input logic circuits are built as shown in Fig. 2-11. Here, the single two-input gate serves directly as a NAND circuit (Fig. 2-11A), for both inputs must have 1's (be grounded) for there to be a 0 (+) output. For an AND circuit (Fig. 2-11B), the output is inverted with a NOT gate. Both inputs are inverted to get an OR gate (Fig. 2-11C), and both inputs and the output are inverted to get a NOR gate (Fig. 2-11D).

The EXCLUSIVE NOR circuit (Fig. 2-11E) combines an OR circuit with a NAND circuit. The OR circuit works normally unless simultaneous 1's appear at the input. If this happens, the NAND circuit takes over and *inhibits*, or blocks, the inverting gate at the OR output. The EXCLUSIVE OR circuit (Fig. 2-11F) simply contains another inverter on the end of an EXCLUSIVE NOR.

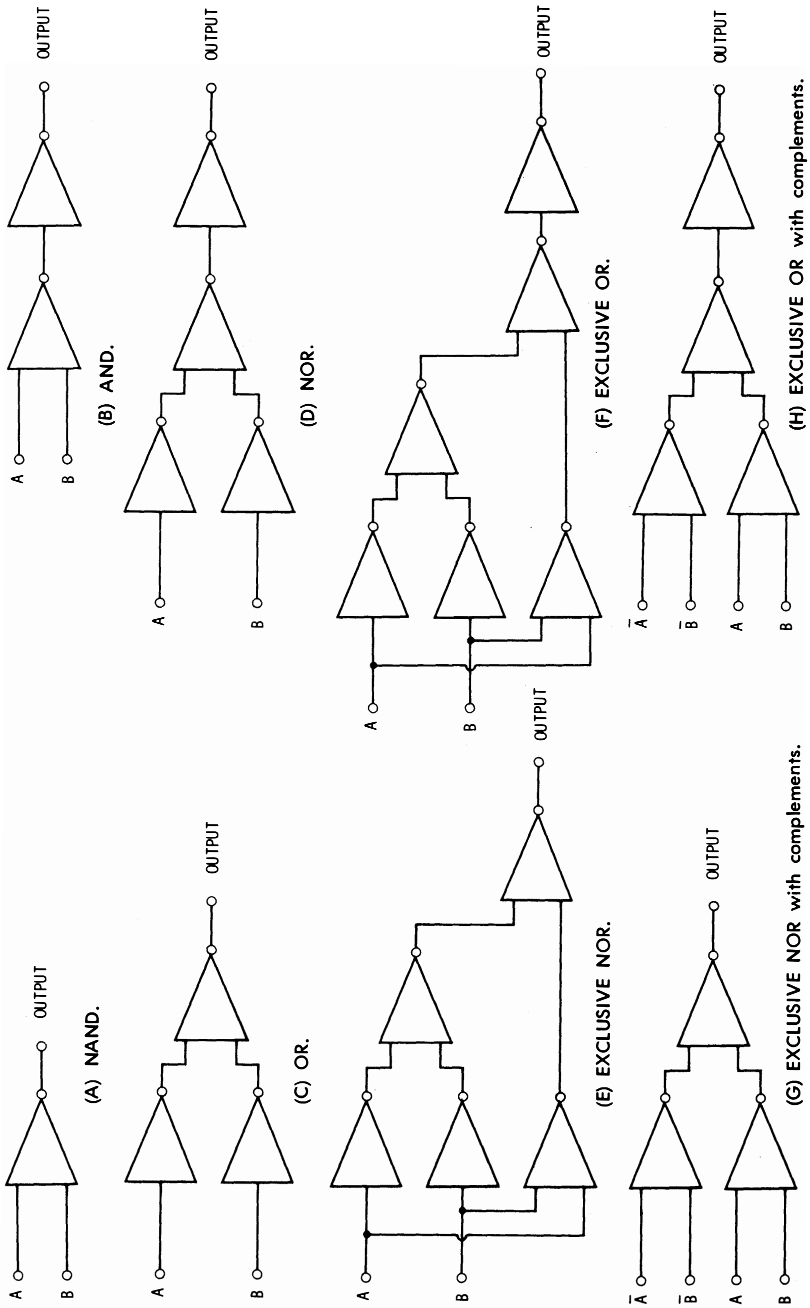
Sometimes both the complement and the actual input signal are available. If this is the case, the last two circuits may be simplified as shown in Figs. 2-11G and 2-11H, nicely eliminating two inverters from the circuit.

Remember, all the circuits of Fig. 2-11 are good only if 1 is defined as ground and 0 is defined as +. Suppose we do the opposite. The gates all have to be changed around if 1 is defined as + and a 0 is defined as ground. For this *new definition*, the logic circuits are built up as shown in Fig. 2-12. Now the two-input gate by itself is a NOR gate (Fig. 2-12A). Invert the output for an OR gate (Fig. 2-12B), and both inputs for an AND gate (Fig. 2-12C). Invert the inputs and the output for the NAND (Fig. 2-12D).

The EXCLUSIVE OR (Fig. 2-12E) combines a NOR with an AND and an inhibiting inverter. A final inversion produces the EXCLUSIVE NOR (Fig. 2-12F). And once again, if inputs and their complements are readily available, two inverters may be saved, as shown in Figs. 2-12G and 2-12H.

CHOOSING LOGIC DEFINITIONS

There are special names for the "what-a-one-is" definitions. If a 1 is grounded, we are using *negative* logic; if a 1 is positive, we are using *positive* logic.



(A) NAND.

(B) AND.

(C) OR.

(D) NOR.

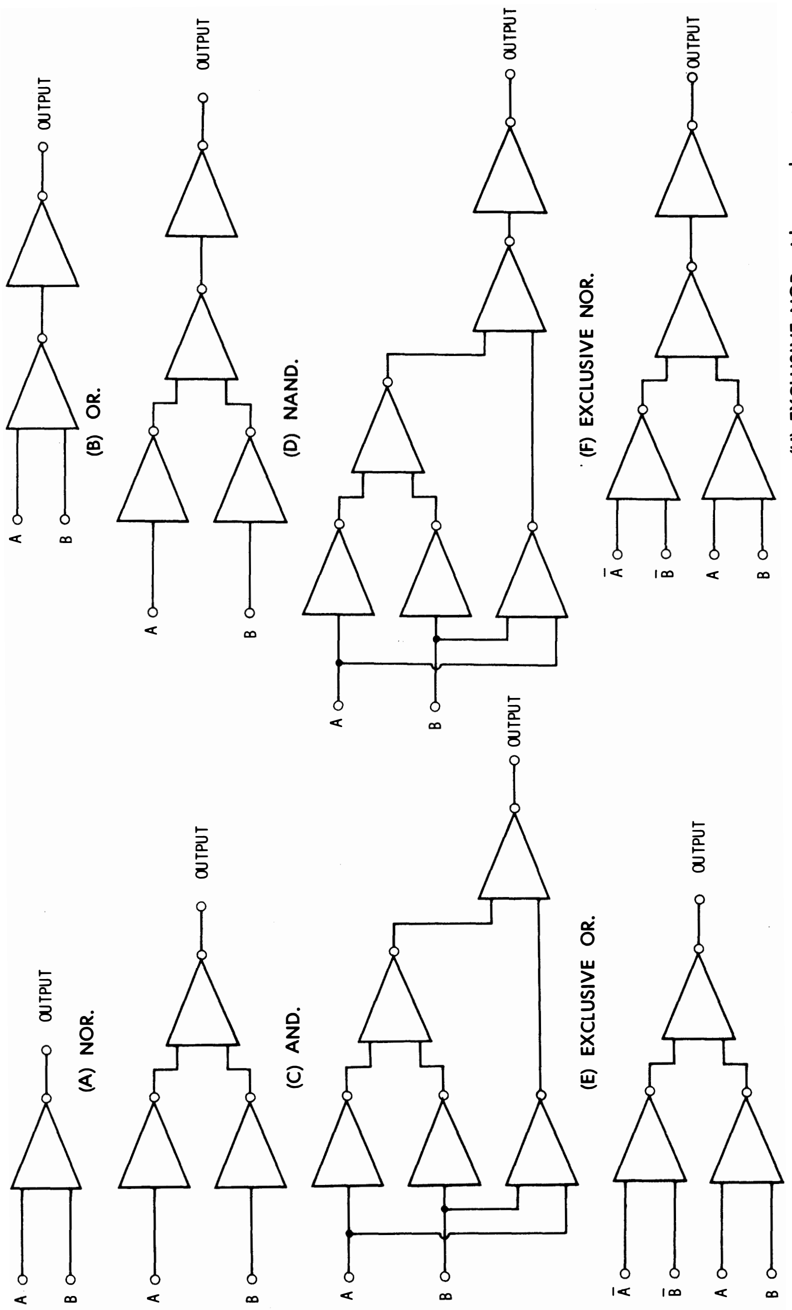
(E) EXCLUSIVE NOR.

(F) EXCLUSIVE OR.

(G) EXCLUSIVE NOR with complements.

(H) EXCLUSIVE OR with complements.

Fig. 2-11. Negative logic functions with RTL.



(A) NOR.

(B) OR.

(C) AND.

(D) NAND.

(E) EXCLUSIVE OR.

(F) EXCLUSIVE NOR.

(G) EXCLUSIVE OR with complements.

(H) EXCLUSIVE NOR with complements.

Fig. 2-12. Positive logic functions with RTL.

It appears there are two ways to realize any of the six basic logic functions. We can use negative logic, call a 1 the grounded state, and use the circuits of Fig. 2-11; or we can call a 1 positive, use positive logic, and use the circuits of Fig. 2-12. But which one do we choose? We pick the circuit that requires the fewest parts. If an entire system requires ten NAND gates, only ten two-input gates are needed with negative logic, but forty gates are needed with positive logic.

All of the gates in RTL are basically inverters, so when RTL is used the NAND and NOR circuits are always easier to build and require fewer parts. So, we usually bend our thinking to "fit" what RTL "likes" to do, and avoid the AND and OR functions wherever possible. In fact, it happens that practically everything done with RTL can be centered around the negative logic definition and the basic NAND gate. Later, it will be shown that the operation of counting flip-flops is *defined* only in terms of negative logic.

The choice of a logic definition boils down to this: Always use the negative logic definition *unless* you are absolutely certain that parts can be saved by using the positive logic definition. If there are any flip-flops at all in the circuit, you are likely to be forced into using the negative-logic circuits in any event. Above all, learn to think out RTL in negative logic—it is the logic you will use over 90% of the time. If a "1" being ground seems bothersome, try thinking of it as a saturated output transistor.

WHAT GOOD IS LOGIC?

We now have all the rules for the two-input logic functions, and a choice of ways to build each circuit. What good is all this? Plenty.

Look again at the simple switch of Fig. 2-1; it is really a NAND gate. As will be shown in the next chapter, when two NAND gates are connected back-to-back, they acquire such features as memory, time delay, or pulse shaping, or they may serve as oscillators.

We have already seen how the EXCLUSIVE OR circuit directly performs binary arithmetic and forms the basis of all digital computer calculations. Also, it is possible to use NAND gates to make a binary counting chain count by any desired number, through a feedback process. More NAND gates can be used to *decode* the same counter, and produce a unique output for each and every counter state.

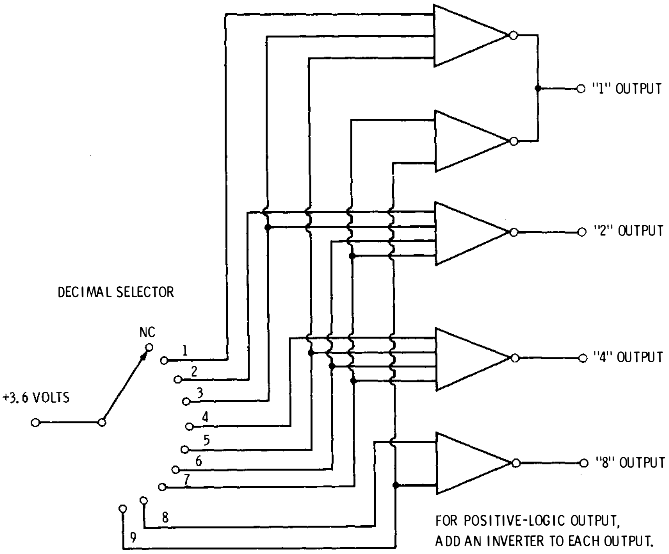
DECIMAL-TO-BINARY ENCODER

Gates may be used to convert from one number system to another, or to convert one code to another. Everyday arithmetic is done in base 10; computers work in base 2, coded in any of several dozen possible ways. Gates make it possible to convert decimal numbers to binary numbers, convert binary numbers to decimal numbers, and convert the various binary codes to each other.

Fig. 2-13. Decimal-binary equivalents.

DECIMAL	BINARY (8421)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

One popular binary code is called the *binary-coded decimal*, or 1-2-4-8, code. In this code, a decimal number corresponds to its binary equivalent up to count nine (Fig. 2-13). Each decimal decade is treated as a new binary number; when ten is reached, the conversion is repeated, one column to the left. Thus 24 = 0010 0100; 395 = 0011 1001 0101, and so on. The binary equivalents 10 through 15 are not used; these are simply "thrown away."



FOR POSITIVE-LOGIC OUTPUT, ADD AN INVERTER TO EACH OUTPUT.

Fig. 2-14. Decimal-to-binary encoder.

This decimal-like grouping of binary numbers makes decimal-to-binary conversion easy, particularly for larger numbers. While this is a rather popular binary code, there are many others.

The process of going from decimal to binary numbers is called *encoding*, and an RTL decimal-to-binary encoder is shown in Fig. 2-14. A ten-position selector switch sets up the proper gates to produce the binary equivalent of the selected decimal number. Each gate is basically a NOR gate. As

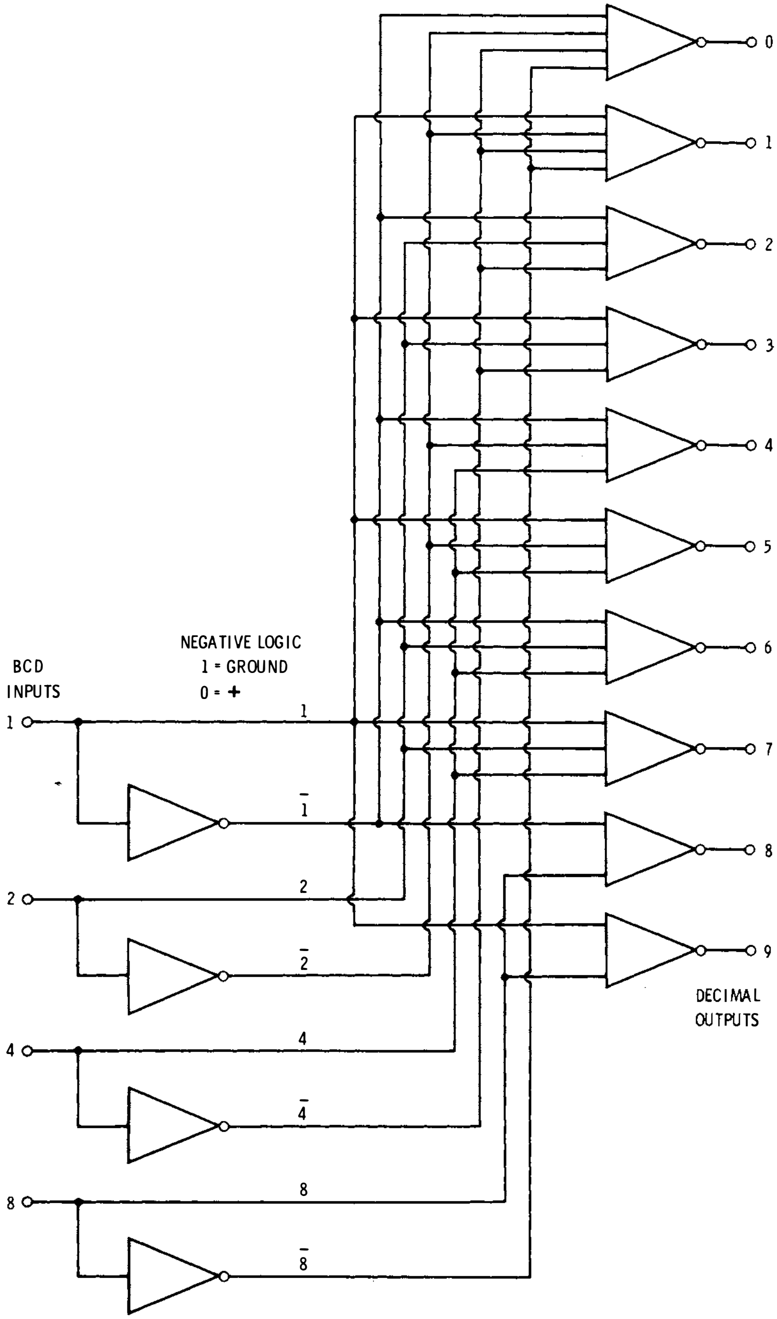


Fig. 2-15. Binary-to-decimal decoder.

the selected gates receive base current, their outputs go to ground, producing negative-logic 1's at the output. If positive-logic 1's are desired, the outputs are simply inverted.

BINARY-TO-DECIMAL DECODER

The opposite of encoding is called *decoding.* As Fig. 2-15 shows, ten gates are necessary to accept the 1-2-4-8 inputs, NAND them together, and produce the equivalent decimal output. A positive output voltage appears at the correct decimal output. This voltage then may be used to drive a lamp, a printer hammer, or something else in the outside world. If both the binary input and its complement are available, the four input inverters can be omitted.

Remember, the circuits in Figs. 2-14 and 2-15 are useful only for converting decimal to binary-coded-decimal, or 1-2-4-8, coding, and vice versa. There are many other binary codes (some will be covered later), and all the other codes require different encoding and decoding.

INDICATING STATES: DRIVING THE OUTSIDE WORLD WITH RTL

Although a low-current, low-voltage light bulb can be operated directly from the output of an inverter, gate, or buffer, the brightness will be very low and the results highly nonuniform. There are several good ways to determine what state any RTL logic block is in, and several ways to transfer this information to the outside world. We can use a voltmeter, any lamp with a proper choice of drive transistor or SCR, a relay, printer hammer, or indicator coil, or an audio-tone module.

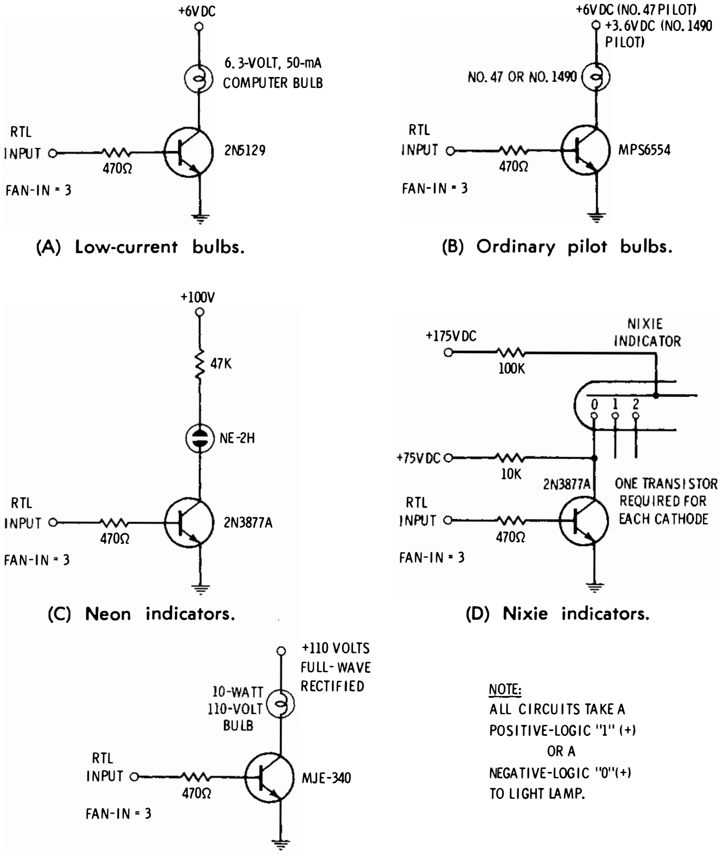
Voltmeters

An ordinary VOM, a VTVM, or a transistor voltmeter can be used to check the output and input levels on any gate. The *grounded* state measures 0.2 volt or less. The positive-state voltage depends on the loading and the supply voltage; it will vary from 1.3 to 3.6 volts from a 3.6-volt supply. Just set the VOM to the 3-volt scale. If the pointer barely moves, the output is in the grounded state; if the pointer swings well up scale, the output is in the + state.

Indicator Lamps

Either transistors or SCR's can be used as lamp drivers. Npn *silicon* transistors are the best choice—their levels are immediately compatible with RTL, and they need the same supply polarity. Because saturated transistors have a very low input impedance, always add a series base resistor to any lamp driver, unless it is the only load presented to a gate. If you always make this resistor 470 ohms, you can be assured of at least 1 milliampere of base-current drive, and an equivalent RTL fan-out requirement of approximately 3. Then select an npn transistor with enough gain and enough breakdown voltage to handle the lamp to be driven.

If low-current bulbs are used, they typically require 50 milliamperes of drive; to insure saturation, the drive transistor should have a gain of at least 50 at the 50-milliampere level. The 2N5129 transistor is suitable and may be used to drive lamps requiring up to 28 volts. The circuit is shown in Fig. 2-16A.



(A) Low-current bulbs.

(B) Ordinary pilot bulbs.

(C) Neon indicators.

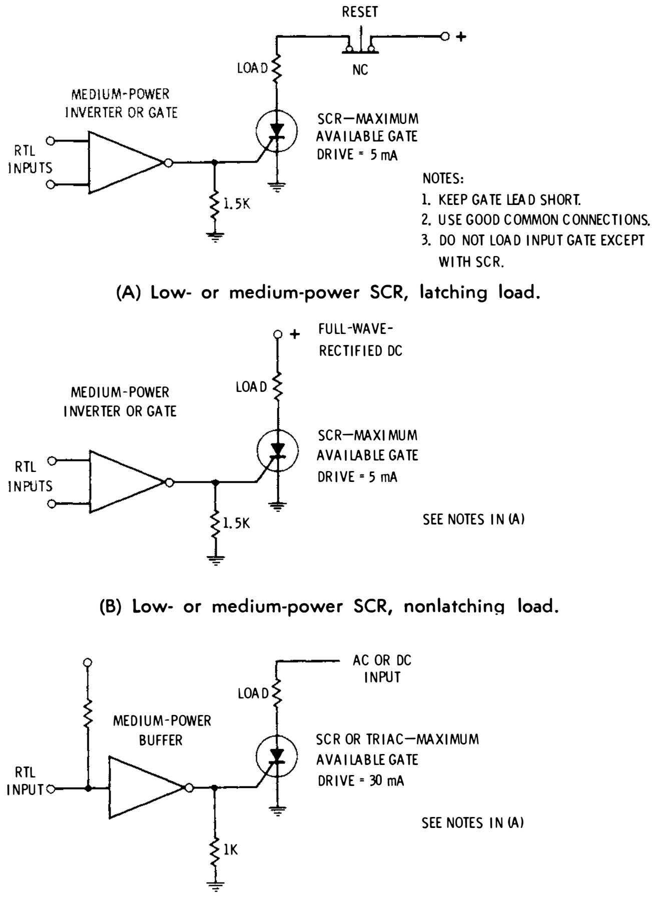
(D) Nixie indicators.

(E) 110-volt lamps.

Fig. 2-16. Transistor lamp drivers.

The bulbs may be driven from dc, full-wave rectified dc, or half-wave rectified dc. With half-wave rectified dc, use double the normal bulb voltage. This will make the brightness the same for any type of supply. If there are many bulbs in the IC system, avoid using filtered dc on them, since it only adds expensive electrolytic capacitors to the circuit and makes the ripple worse for the rest of the circuit.

Ordinary pilot lamps need more current, so a transistor with a high gain at the 150-milliampere current level is needed. The MPS6554 may be used; the circuit is shown in Fig. 2-16B.



(A) Low- or medium-power SCR, latching load.

(B) Low- or medium-power SCR, nonlatching load.

(C) High-power SCR or Triac, ac or dc supply.

Fig. 2-17. SCR or Triac load control.

Nixie indicators, neon bulbs, and other numerical readouts require little current, but a high-voltage transistor is needed, as the circuits of Figs. 2-16C and 2-16D show. Two suitable transistors are the 80-volt 2N3877A and the 120-volt 2N4410. You can directly drive a 10-watt, 100-volt light bulb by using a 200-volt, high-gain power transistor. One transistor to use is the MJE-340, in the circuit of Fig. 2-16E.

SCR's may be used to drive outside-world loads, as the circuits in Fig. 2-17 show. Ordinary low-current SCR's may be driven directly from an inverter or a gate, but the high-power *Triacs* and SCR's must be driven from a buffer. By using these high-power semiconductors, you can directly control several kilowatts of power with RTL. If an SCR or *Triac* is dc powered, it will latch on until it is reset manually. If these devices are powered from full-wave rectified dc, half-wave rectified dc, or ac, they will automatically shut off shortly after the input signal does. Be sure to keep the gate-lead lengths very short and use good grounds and common connections to prevent any noise or interaction problems; in particular, prevent the SCR load current from affecting the RTL ground. This is particularly true of the new low-cost, plastic SCR's with their extremely sensitive gates.

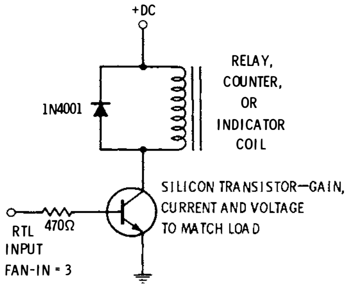


Fig. 2-18. Driver for inductive loads.

More complex RTL circuits will allow proportional control of ac-powered semiconductor switching devices without the use of trigger diodes and the usual phase-control circuitry. These techniques are most useful when zero-point switching, feedback, or dc voltage control is essential.

Relays and Driver Coils

Fig. 2-18 shows the techniques for driving inductive loads. The problems are the same as in driving lamps; pick a transistor with enough gain, current, and breakdown voltage to handle the load. When driving coils, be sure to add a damper diode as shown in Fig. 2-18; this protects the tran-

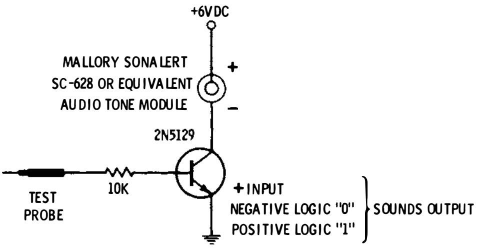


Fig. 2-19. IC "logic-state checker."

sistor or SCR from a voltage spike if the semiconductor is turned off suddenly. If SCR's are used to drive relays, make sure the gate signal is present long enough to give the SCR a chance to turn on fully. Several milliseconds are usually adequate.

Audio-Tone Modules

Some types of audio-tone modules may be directly driven with almost any npn silicon transistor in the circuit of Fig. 2-19. This arrangement provides an audio tone output, and serves as a useful "state-checker" for use in troubleshooting IC circuits.

CHAPTER 3

Multivibrators

When two inverting gates are connected back-to-back, a *multivibrator* is formed. Depending on the connections, the multivibrator may have zero, one, or two stable states. Multivibrators are used for oscillators, memories, latches, pulse shapers, frequency multipliers and dividers, reset generators, time delays, signal conditioners, synchronizers, and level detectors. In this chapter, we will take a closer look at this back-to-back gate connection and see how to design any desired type of RTL multivibrator.

TWO INVERTERS BACK-TO-BACK

Suppose, as in Fig. 3-1, we connect two inverters back-to-back so that the output of one inverter drives the input of the other inverter, and vice versa. The circuit has two stable states. If inverter A happens to be receiving no base current, it stays off. If inverter A is off, its output is positive and, in turn, delivers base current to inverter B. Therefore, inverter B will be on. If inverter B is on, its output is grounded; this condition is just what inverter A needs to stay off. If undisturbed, the two inverters will remain this way indefinitely.

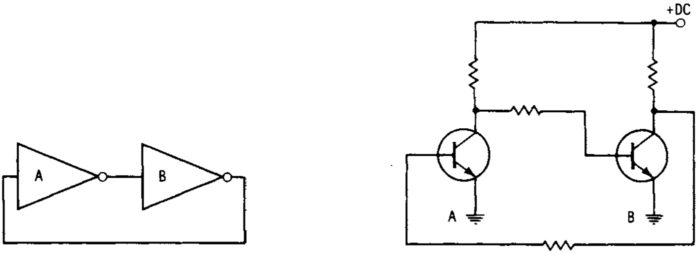
We could just as easily have inverter B off and inverter A on; this state is just as stable. The circuit can be in one of two possible states; if undisturbed it will remain in either state. This action is somewhat similar to the latching of a mechanical relay on its own "make" contact.

TRIGGERING: THE BISTABLE MULTIVIBRATOR

The circuit in Fig. 3-1 would be far more useful if it could be forced to go into a desired one of the two possible states on command. The process of forcing the two back-to-back inverters into a desired state is called *trig-*

gering; a triggerable pair of back-to-back inverters is called a *bistable multivibrator* (often shortened to "bistable").

Suppose, once again, that inverter A is off, and its output is shorted to ground momentarily. This removes base current from inverter B and turns B off. The output of B swings positive and provides base current for A; A turns on. When the momentary short is removed, A stays on. By shorting the output of inverter A, we have forced the bistable multivibrator to go into the state with the output of inverter A grounded and the output of inverter B positive. If the output of inverter B is shorted momentarily, the circuit will flip back to the state in which the output of B is grounded and the output of A is positive.



(A) Logic-symbol diagram.

(B) Discrete equivalent.

Fig. 3-1. Inverters back-to-back.

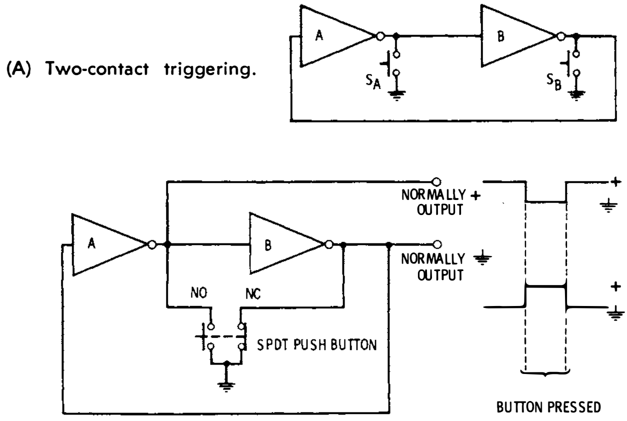
There are three ways to trigger a bistable multivibrator and force it to go into a desired state. It can be triggered mechanically with a contact or a switch, electronically with another gate input, or by capacitor coupling a negative pulse into the output of the off inverter.

Mechanical Triggering

In Fig. 3-2A, momentarily closing contact S_A grounds the output of inverter A. Momentarily closing contact S_B grounds the output of inverter B. Either contact closure forces the flip-flop to assume and *stay in* the desired state.

One major application of this simple circuit is to make a push button bounceless. Ordinary push buttons generate noise and contact bounce both on contact closure and on opening. If this noisy operation is fed directly to any high-speed IC circuitry, very erratic multiple counting and triggering is almost a certainty. To get around this problem, the circuit of Fig. 3-2B may be used with a spdt push button. Whenever the button is pressed, the normally open contact momentarily "makes," bounces for a while, and then finally comes to rest. The very first momentary closure triggers the bistable into one state; it remains in that state throughout the bouncing and settling time. When the button is released, the very first closure of the normally closed contact triggers the bistable back into its initial state. The result is a high-speed square wave, lasting from initial make to final release, that is bounce and noise free—and that has rise and fall times fast enough to drive reliably the fancier counting flip-flops to be discussed later.

Other possible applications of the mechanically triggered bistable are in latches and memory circuits. This circuit could be used in an annunciator or an alarm, or perhaps as the memory of a tick-tack-toe computer, reflex tester, or some other electronic game.



(A) Two-contact triggering.

(B) Bounceless push button.

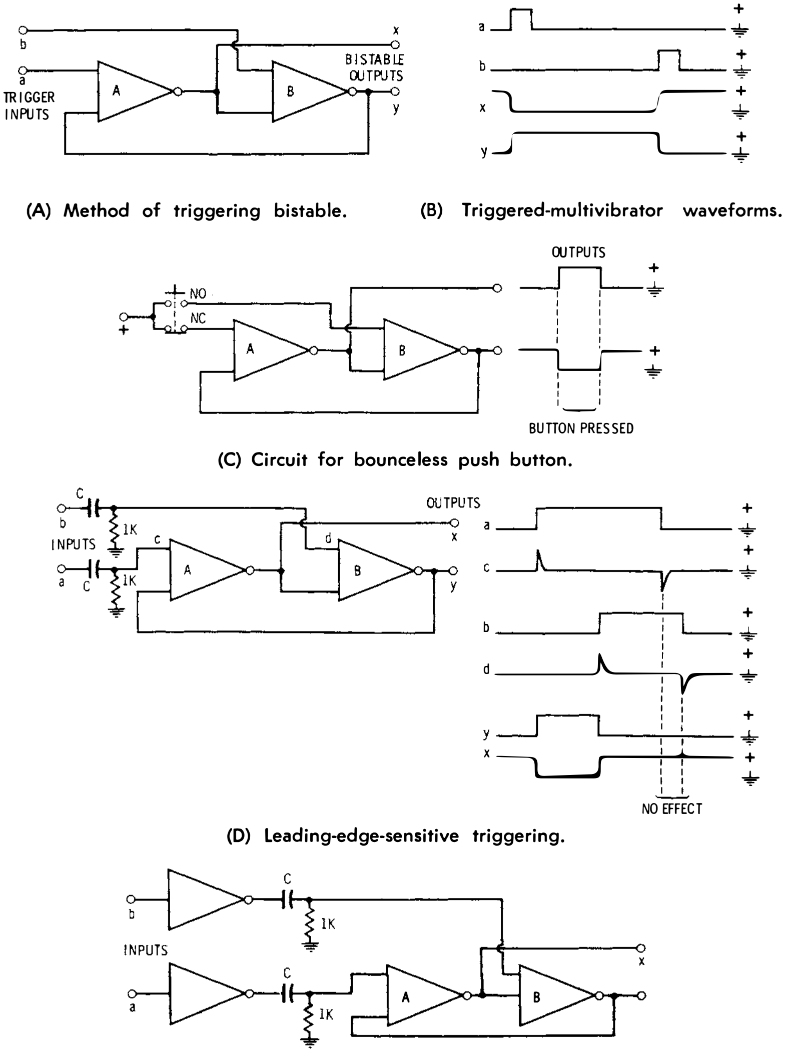
Fig. 3-2. Mechanical triggering of bistable multivibrator.

Electronic Triggering

Instead of a one-input gate, Fig. 3-3A shows a two-input gate on each side of the bistable, and the outputs are cross-coupled to the inputs so that only one input is used on each gate. This connection leaves one input available on each side. When one of these inputs is made positive, the bistable flips into the state that grounds the output of the side whose input was made positive. Fig. 3-3B shows how narrow pulses are delivered to the desired input anytime it is desired to trigger the bistable. These pulses may be obtained from previous RTL circuits in the system. The first pulse must end before the second one arrives.

Fig. 3-3C shows another bounceless push button. This time, the contacts are used to deliver base current to the input of the desired gate. The only advantage of this circuit over the previous one is that you can buy a dual two-input gate, but inverters normally come six to a package. You can mount the dual two-input gate directly on the push button if you like, to make a bounceless button in practically the same space required by an ordinary one.

A *leading-edge-sensitive* bistable is shown in Fig. 3-3D. Here, if input *a* suddenly goes positive and stays positive, a brief triggering pulse is delivered to the bistable. This pulse is obtained by capacitor-resistor differentiation of the leading edge of the input signal. The same thing happens if input *b* suddenly goes positive. The bistable is now sensitive only to the *start* of the input signals, and not to their *duration.* The trailing edge, or end, of each input pulse has no effect since it delivers a *negative* turn-off pulse to an input that is already off.



(A) Method of triggering bistable.

(B) Triggered-multivibrator waveforms.

(C) Circuit for bounceless push button.

(D) Leading-edge-sensitive triggering.

(E) Trailing-edge-sensitive triggering.

Fig. 3-3. Electronic triggering of bistable multivibrator.

A use of electronic triggering would be in a ballistic chronometer. Two normally conductive screens are placed a known distance apart in front of a rifle. The bullet breaks the screens in sequence, causing the bistable to flip to one state the instant the bullet crosses the first screen, and flip back the instant the bullet crosses the second screen. The bistable output is used to turn on a reference-frequency source, the output pulses of which are counted. The counter output then indicates the time the bullet took to traverse the known distance between the two screens. When the time and distance are known, a simple division gives the velocity of the bullet.

The choice of resistor and capacitor values depends highly on the circuit. The RC time constant has to be long enough to pass the rise of the input signal, but not so long as to interfere with the later change of state caused by the other input. A resistance of 1k is usually a good value for this circuit. With a 1k resistor, the capacitance in microfarads is equal to the time constant in milliseconds. With normal, high-speed RTL transitions, a good value for C is often 0.001 microfarad (1000 pF), with a resultant 1.0-microsecond time constant. Slower events, such as the breaking of the screens by the bullet, may require correspondingly longer time constants.

A *trailing-edge-sensitive* bistable (Fig. 3-3E) is obtained simply by inverting both inputs before differentiation. Now the leading edge has no effect, and the trailing edge produces a positive trigger pulse that flips the bistable into the desired state.

Pulse Triggering

In Fig. 3-4, a negative turn-off pulse is coupled into the output of the off inverter. The negative pulse robs the on inverter of its base current long enouɥh to allow the bistable to flip into its proper state. This triggering requires careful control of pulse amplitude and shape. It also depends on the output loading of the bistable, and a portion of the trigger amplitude may appear in the output.

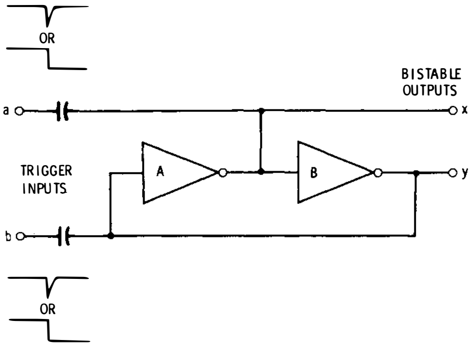
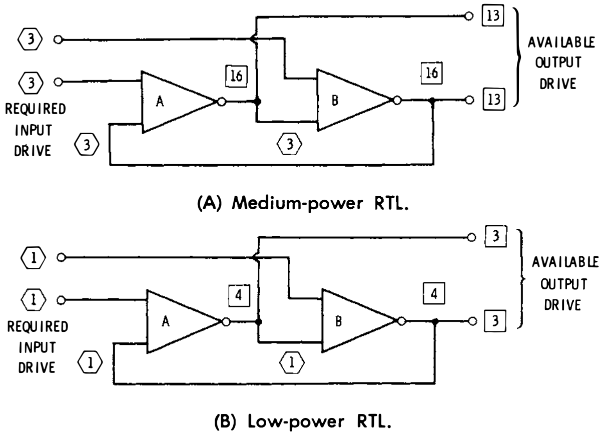


Fig. 3-4. Pulse triggering of bistable multivibrator.

While this connection was quite popular with discrete transistor multivibrators, there is seldom, if ever, any need for it with RTL integrated circuits. Sometimes, however, if you need a sequence of cascaded time delays, this triggering mode may be of value.

LOADING

There are no loading restrictions on the bistable multivibrators except those of the gates or inverters themselves. In the case of a bistable built with medium-power RTL, the total fan-out per output is 16, of which 3 is needed to run the other input. This leaves 13 units of fan-out for outside use. A low-power gate starts with a fan-out of 4; 1 unit is needed to run the other gate, and an output fan-out of 3 remains. Fig. 3-5 shows the details. Two buffers may be used as a bistable, but unless an exceptional amount of fan-out is needed, the supply-power consumption favors the use of lower-power devices.



(A) Medium-power RTL.

(B) Low-power RTL.

Fig. 3-5. Available output of a bistable multivibrator.

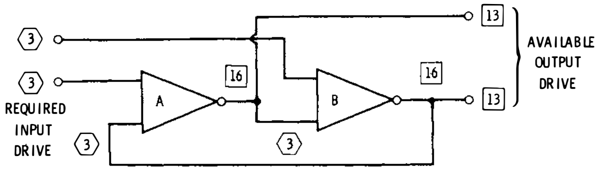
The available output *voltage* is usually of no concern in any RTL circuit, so long as there is fan-out left. If you need to measure it or want to use it in the outside world, this voltage is around half the supply voltage for light loading and around one-third the supply voltage for heavy loading. When the output is "grounded," the actual terminal voltage will range from 0.1 to 0.2 volt positive; this small saturation drop may be subtracted from the normal "high" voltage output to obtain the output voltage *swing* available.

The fact that the output voltage in the grounded state never exceeds 0.2 volt guarantees that a grounded output cannot provide base current for the next stage—the next stage needs at least 0.6 volt to overcome the base-emitter turn-on voltage drop.

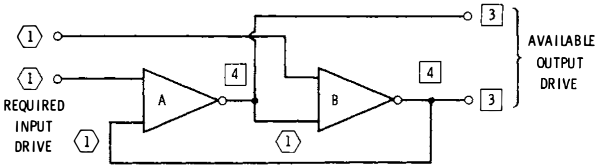
While this connection was quite popular with discrete transistor multivibrators, there is seldom, if ever, any need for it with RTL integrated circuits. Sometimes, however, if you need a sequence of cascaded time delays, this triggering mode may be of value.

LOADING

There are no loading restrictions on the bistable multivibrators except those of the gates or inverters themselves. In the case of a bistable built with medium-power RTL, the total fan-out per output is 16, of which 3 is needed to run the other input. This leaves 13 units of fan-out for outside use. A low-power gate starts with a fan-out of 4; 1 unit is needed to run the other gate, and an output fan-out of 3 remains. Fig. 3-5 shows the details. Two buffers may be used as a bistable, but unless an exceptional amount of fan-out is needed, the supply-power consumption favors the use of lower-power devices.



(A) Medium-power RTL.



(B) Low-power RTL.

Fig. 3-5. Available output of a bistable multivibrator.

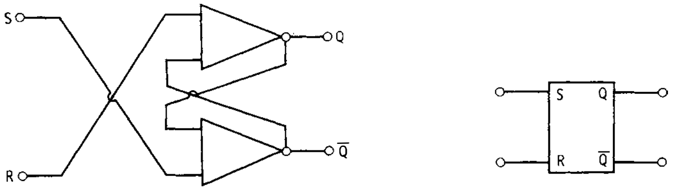
The available output *voltage* is usually of no concern in any RTL circuit, so long as there is fan-out left. If you need to measure it or want to use it in the outside world, this voltage is around half the supply voltage for light loading and around one-third the supply voltage for heavy loading. When the output is "grounded," the actual terminal voltage will range from 0.1 to 0.2 volt positive; this small saturation drop may be subtracted from the normal "high" voltage output to obtain the output voltage *swing* available.

The fact that the output voltage in the grounded state never exceeds 0.2 volt guarantees that a grounded output cannot provide base current for the next stage—the next stage needs at least 0.6 volt to overcome the base-emitter turn-on voltage drop.

THE RESET-SET FLIP-FLOP

The circuit of Fig. 3-3A can wear any of several hats in the IC world, and thus is called by several names: bistable multivibrator, latch, back-to-back gates, or memory. We can also call this circuit a *reset-set flip-flop*, or *RS flip-flop* for short, and treat it as a computer-logic element just as we did the logic gates by themselves. We will now look at the bistable multivibrator in this new light, rearranged and drawn schematically either of the ways shown in Fig. 3-6.

The RS flip-flop has two inputs, the set and reset inputs, and two outputs, a Q and a $\bar{Q}$ output. Since one output is normally positive and the other normally grounded, these outputs are complementary to each other, as indicated by the Q and $\bar{Q}$ notation.



(A) Diagram.

(B) Logic symbol.

Fig. 3-6. Bistable multivibrator, or RS flip-flop.

The most important thing to notice is that this logic block possesses *history.* We must take into account the last thing that happened to it before we can find out whether any change in output will take place. This is an important difference between flip-flops in general and the simpler logic blocks.

Since there are two digital inputs, there are four possible input combinations to consider:

1. If both inputs are *grounded*, nothing will happen, and the flip-flop will remain in its previous state.
2. If the S input is made positive and the R input is grounded, the flip-flop goes into the state that has the Q output *positive*, regardless of what state the flip-flop was in before.
3. If the R input is made positive and the S input is grounded, the flip-flop goes into the state that has the $\bar{Q}$ output positive, regardless of the previous state.
4. If both inputs are made positive, *both* outputs go to ground. Obviously, in this condition, the Q and $\bar{Q}$ outputs are no longer complementary. The state the flip-flop assumes is determined by the last input to go to ground. Although there are rare instances in which you would want both outputs grounded, this mode of operation is not

normally considered useful by the computer people; they call this a *disallowed state*, or an *undefined* output condition. Thus in an RS flip-flop, only three of the four possible input conditions are normally considered useful.

Converting all this information to a negative-logic chart results in Fig. 3-7. In this chart, the history appears as Q_n. This is the state the flip-flop was in before the new conditions arrived. If Q_n stays the same, the flip-flop did not change state with the new conditioning.

A positive-logic chart may be built up in the same manner, but it would only lead to confusion later when we talk about the JK and Type-D flip-flops that are defined only in terms of the negative-logic definition.

With the RS flip-flops, we have added to our arsenal a new logic weapon —one with a history, or memory, capability.

INPUT S	INPUT R	OUTPUT Q	OUTPUT $\overline{Q}$
1	1	Q_n	$\overline{Q}_n$
0	1	0	1
1	0	1	0
0	0	(1)	(1)
		(UNDEFINED STATE)	

Fig. 3-7. Negative-logic chart, RS flip-flop.

TRIGGERING RESTRICTIONS ON THE RS FLIP-FLOP

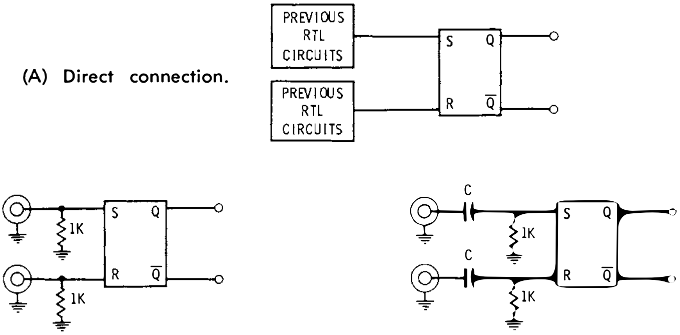
Brief trigger pulses obtained from other RTL circuits may be routed directly to the inputs of an RS flip-flop. The rise times and fall times are not at all critical, but unless the undefined state is specifically desired, there must be some guarantee that the set input disappears before the reset input appears. Should you apply simultaneous 0's to both the set and reset inputs, the *last* 0 to change to a 1 determines which state the flip-flop will assume.

Input signals from +1 to +4 volts are ideal for triggering. Input voltages above +4 volts should be attenuated with suitable resistors before they reach the flip-flop. Input trigger pulses must be positive-going; a negative voltage applied to an npn transistor that is already off will have no effect.

If desired, more than one input on either side may be obtained; just use a gate with three or four inputs instead of two. This provides multiple points of control for the RS flip-flop. Keep all inputs 1 (grounded); a 0 (positive voltage) on any input will then trigger the flip-flop into the desired state. In addition, any desired combination of triggering may be used, such as mechanical on one side and trailing-edge sensitive on the other.

The input to the RS flip-flop *must* have a dc return path to ground for the input base current. Fig. 3-8 shows several ways to guarantee a path for the base current. In Fig. 3-8A, the ground-return path is provided automatically by a previous RTL stage. In Fig. 3-8B, 1k resistors have been added to each input for outside-world signals, thus guaranteeing the return path no matter how the input signal is derived.

The input resistors become particularly important if capacitor coupling is used at the input (Fig. 3-8C), regardless of whether the original signal comes from more RTL or from the outside world. If the resistors were omitted in this case, the base-emitter junction of the input transistor would act as a dc restorer and charge the input capacitor. After one or two cycles of operation, the capacitor would stay fully charged and pass on no more input signals. The 1k resistors eliminate this problem.



(A) Direct connection.

(B) External resistors.

(C) With RC coupling.

Fig. 3-8. Dc return paths for input base current.

COUNTING WITH RS FLIP-FLOPS

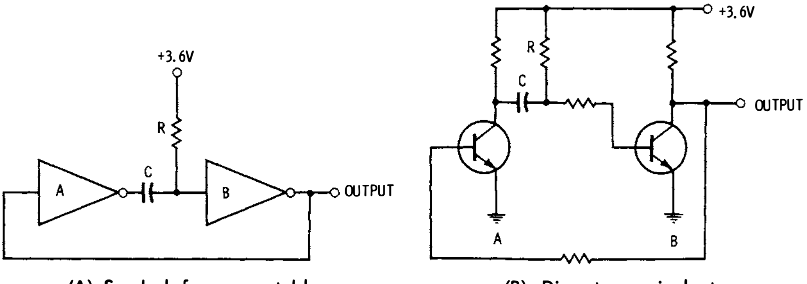
A single RS flip-flop cannot count by itself, for simply connecting the two inputs together will only bring about the undefined state. To get an RS flip-flop to count, you have to add a *steering* and *sensing* mechanism that senses what state the flip-flop is in and automatically routes the input pulse to the *other* side. Secondly, you must provide either *storage* or *time delay* just after the flip-flop changes state, for otherwise you would have either an oscillator or a device that always flips into a preferential state. And when this is done, you still have to make sure that there is no way for one input to *ripple through* an entire cascaded group of flip-flops, changing all of them instead of just the first one.

Getting reliable counting operation from an RS flip-flop is difficult when discrete-transistor circuits are used. You must be careful about the input pulse shapes and durations; watch the loading, supply, and temperature variations; and often select gain-matched transistor pairs. These factors, however, are not problems with RTL IC's. Two types of RTL counting flip-flops are available, the *JK flip-flop* and the *Type-D flip-flop.* Both types automatically and reliably perform all the steering, time-delay, storage, and anti-ripple-through functions. Only ½ or sometimes 1 integrated circuit is needed for each stage, with no outside components. These flip-flops will be covered in detail in Chapter 5. The important point here is that you need never attempt to convert an RS flip-flop into anything more elaborate; suitable commercial packages are available for such requirements.

THE MONOSTABLE MULTIVIBRATOR

Suppose we again connect two inverters back-to-back, but this time we insert a series capacitor and a resistor recharging network in place of one of the back-to-back connections (Fig. 3-9). At a given instant, this multivibrator circuit could be in either of two states, but eventually capacitor C will become charged, and the circuit will flip into the state in which inverter B is receiving base current and has a grounded output. This circuit has a *preferential* state that it will eventually assume. There is one stable state and one unstable state, and the circuit is called a *monostable multivibrator,* or a time-delay generator. The term monostable multivibrator is often shortened to "monostable."

Usually, the monostable multivibrator is triggered into the unstable state. It then flips back by itself at a later time determined by the choice of R and C. Monostables are used as time-interval generators, time-delay circuits, pulse generators, digital frequency discriminators, and tachometers—in fact, anywhere there is need for a reasonably accurate output pulse of a constant duration under command.



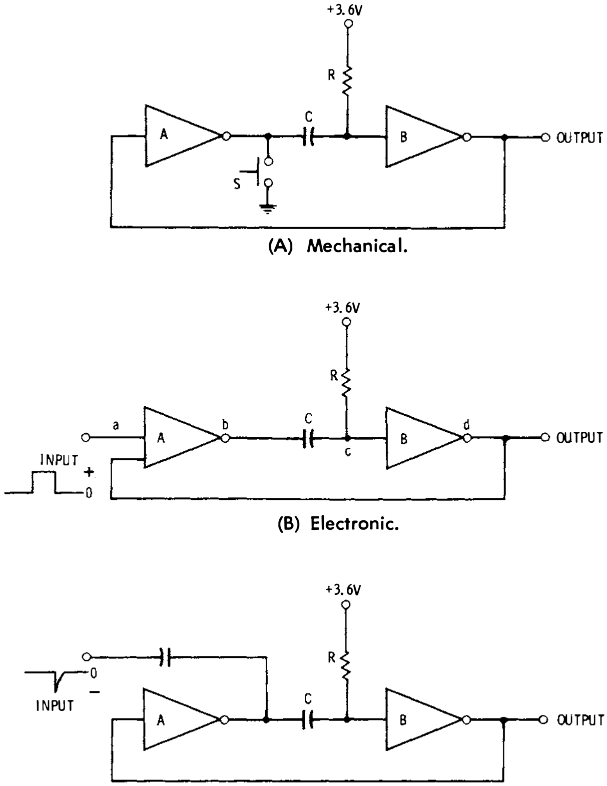
(A) Symbol for monostable.

(B) Discrete equivalent.

Fig. 3-9. Monostable multivibrator.

Triggering

Once again, there are three methods of triggering the circuit—mechanically, electronically, or with pulses—as Fig. 3-10 shows. In Fig. 3-10A, suppose the circuit is undisturbed for a long time. Capacitor C eventually will become charged, and inverter B will receive base current and have its output grounded. At this time, inverter A will be receiving no base current, and its output will be positive. The left end of C is at +3.6 volts, and the right end is at +0.6 volt, caused by the forward base-emitter drop of the on transistor in inverter B. The net charge on C is 3 volts, positive on the



(A) Mechanical.

(B) Electronic.

(C) Pulse.

Fig. 3-10. Triggering of monostable multivibrator.

left. Now, contact S is closed momentarily. The output of inverter A immediately goes to ground. But the charge on a capacitor, or the voltage across it, *can not* change instantaneously. If the left end of the capacitor suddenly is grounded, the capacitor still has to have 3 volts across it. The right end *must* drop to minus 3 volts. This voltage reverse biases inverter B, turning it off. The output of inverter B swings positive and provides base current for inverter A; inverter A stays on. When the contact is opened, inverter A stays on, just as it did in the bistable multivibrator, but only for a while.

Resistor R slowly charges the right end of the capacitor in a positive direction. Eventually, the voltage at the right end of the capacitor reaches +0.6 volt, the value needed to provide enough base current to turn inverter B on. When inverter B is turned on, its output is grounded and removes base current from inverter A. Inverter A then flips back into its original state.

The monostable is triggered into the unstable state. It goes there instantly, but stays there only as long as the RC time constant lets it. Then it swings back to the original state.

In Fig. 3-10B, another input has been added to the stable side, just as was done to the RS flip-flop to allow electronic triggering. Detailed waveforms of the operation of the circuit in Fig. 3-10B appear in Fig. 3-11. In Fig. 3-10C, negative trigger pulses are coupled to the output of inverter A. This connection is not used often.

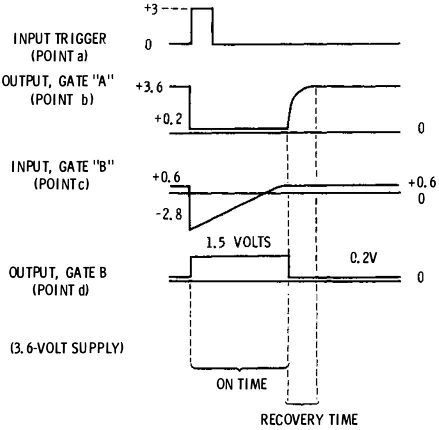


Fig. 3-11. Waveforms of monostable multivibrator circuit.

Regeneration

The transistions in any multivibrator, be it astable, monostable, or bistable, are always very rapid, because once the multivibrator has started to change state, both gates are in their linear regions and act as high-gain amplifiers that *force* the state transition in the desired direction. This process is called *regeneration*, or *positive feedback*, and is the key to having sharp rise and fall times with any multivibrator.

Controlling the On Time

The on time of the monostable may be controlled with either R or C, for either or both varies the recharging time constant. The formula is simply:

$$T_{on} = 0.8\ RC$$

where,

T_{on} is the on time in milliseconds,
R is in kilohms (thousands of ohms),
C is in microfarads.

For instance, a 1-microfarad capacitor and a 2.2k resistor have a time constant of 2.2 milliseconds. The on time is 0.8 × 2.2, or 1.8 milliseconds. Fig. 3-12 is a chart for rapidly determining any desired on time with a variation in either R or C.

The on time is very nearly independent of the supply voltage. It changes only a few percent for a 3-to-4.5 volt supply range. This behavior is caused by two effects which nicely cancel each other: As the supply voltage goes up, the initial capacitor charge goes up, but the available recharging current also goes up at the same time. The on time is slightly temperature dependent, but again it varies only a few percent over a fairly wide temperature range. Because of these effects, a 10-percent on-time stability is easy to obtain. However, for a 1-percent stability, a regulated supply and calibration at the operating temperature are needed.

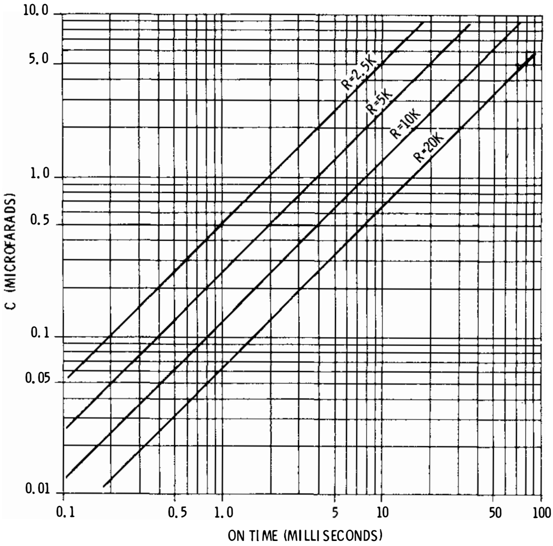
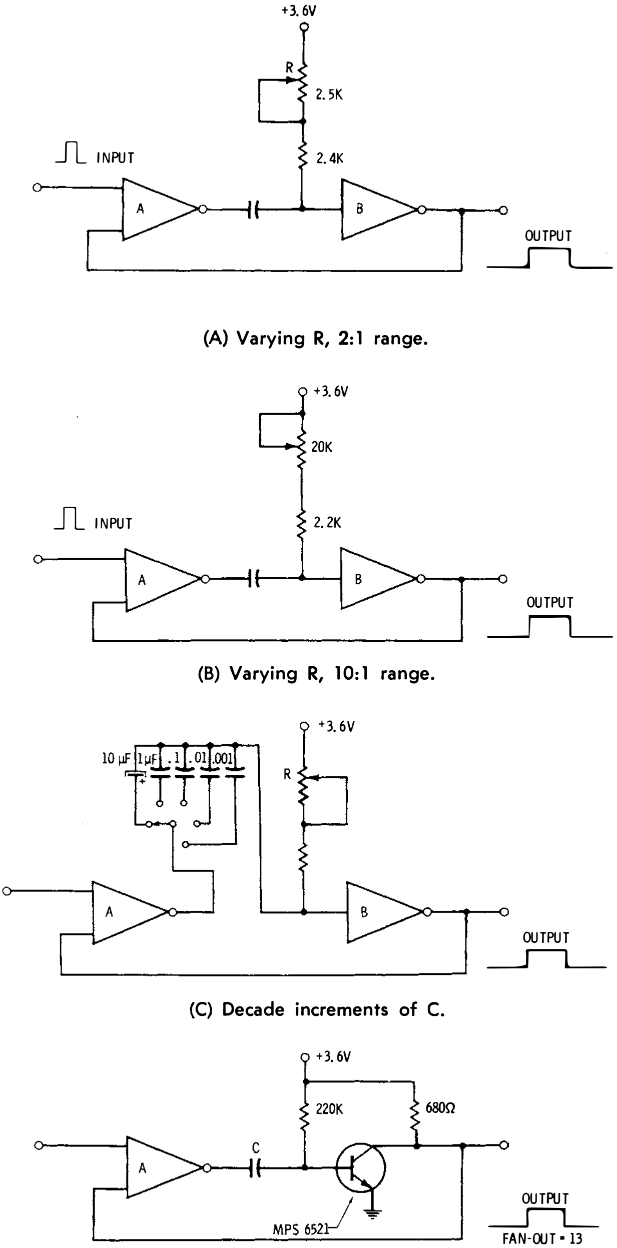


Fig. 3-12. RC chart for monostable multivibrators.

Control and Component Ranges

Capacitor C (Fig. 3-10) may range from 330 picrofarads to several thousand microfarads. When an electrolytic capacitor is being used, the positive end should go to the *output* of gate A. The range for R is more limited; its value should lie between 1 and 25k when a 3.6-volt supply is used. The upper resistance limit is determined by the base current required to saturate the second gate. The lower limit is determined by the series base resistor already present inside the gate; the external resistor should be at least three times this value. Thus the minimum resistance for a buffer should be around 1k; for a medium-power gate, 2k; and for a low-power gate, 4k.

Fig. 3-13 shows how to vary the on time. In Fig. 3-13A, a fixed capacitor, a series resistor, and a potentiometer are used. Varying the resistance of the potentiometer varies the on time. The fixed resistor allows the circuit



(A) Varying R, 2:1 range.

(B) Varying R, 10:1 range.

(C) Decade increments of C.

(D) Hybrid monostable.

Fig. 3-13. Methods of controlling on time.

to remain operative at the minimum potentiometer setting. The ratio of total to fixed resistance determines the range of operation. In Fig. 3-13A, a 2.4k resistor and a 2.5k potentiometer give slightly over a 2:1 adjustment range. In Fig. 3-13B, a 2.2k resistor and a 20k potentiometer give slightly over a 10:1 adjustment range.

In Fig. 3-13C, capacitors are switched to cause large changes in on time. A combination of a 10:1 resistance change and decade-selected capacitors (1 μF, 0.1 μF, 0.01 μF, 0.001 μF, etc.) will give many decades of on-time coverage. Circuits of this type are useful in frequency meters and pulse generators.

On times greater than a second or so are hard to achieve using only IC's because of the relatively small recharging resistors. One way around this problem is to build a "hybrid" transistor-IC multivibrator, and use a very-high-gain transistor with a high-value recharging resistor. Fig. 3-13D is a typical circuit; on times of several tens of seconds may be obtained with this circuit.

Duty-Cycle Restrictions

After the monostable multivibrator has flipped back into its stable state, capacitor C must be recharged fully. This charging takes a certain amount of time which is called the *recovery time* of the circuit.

The ratio of on time to retrigger time is called the *duty cycle* of the monostable. In RTL monostables, a duty cycle of less than 75 percent is essential for any operation at all, and a duty cycle of less than 30 percent is desirable if the retriggering is not to adversely affect the on time. For instance, a 1-millisecond monostable cannot be retriggered faster than once every 1.33 milliseconds. To insure a constant on time independent of how often the circuit is retriggered, it could not be retriggered faster than once every 3.3 milliseconds. If at all possible, keep the duty cycle below the 30-percent range at all times.

Somewhat higher duty cycles may be obtained with larger values (10k-20k) of R, and by the addition of a 470-ohm resistor in parallel with the collector resistor of gate A.

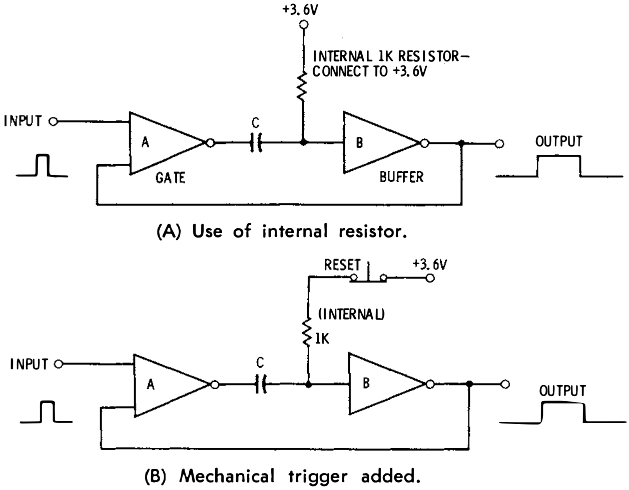
Triggering Restrictions

The input trigger pulse must be of short duration compared to the on time, in order to allow the monostable to turn off again. Pulses of 1.5 or 4.5 volts in amplitude are required for triggering, and any of the electronic triggering circuits of Fig. 3-3 may be used, including the leading- and trailing-edge trigger circuits. Once again, a dc return path for input base current must be provided, as in Fig. 3-8.

Loading Restrictions

Gate A (Fig. 3-10) may not be loaded at all without changing the on time. Usually a loading of 3 on a medium-power gate, or 1 on a low-power gate, may be used if the on time is not particularly critical. The output waveform at this point will have a poor trailing edge, caused by the recovery of capacitor C through the collector resistor of gate A. The output gate (gate B) may be loaded to whatever fan-out is left; this is 13 for a medium-power gate and 3 for a low-power gate. This side produces a positive waveform for the on time; an inverter or buffer may be added to the B output to get a waveform that goes to ground during the on time. This buffered output will not affect the on time and its waveform will have a sharp trailing edge.

Buffers may be used instead of gates where higher fan-out is required. For fixed on times, the internal 1k resistor of the buffer may be used, connected as shown in Fig. 3-14A. The monostable shown in Fig. 3-14B may be either electronically triggered at the input, or mechanically triggered by opening the normally closed RESET push button. This connection is often used to reset electronic counters and other digital instruments. For electronic-only triggering, the 1k resistor is connected directly to +3.6 volts; for electronic plus mechanical triggering, the 1k resistor goes to +3.6 volts through a normally closed contact.



(A) Use of internal resistor.

(B) Mechanical trigger added.

Fig. 3-14. Buffers used for high fan-out.

Applications

A monostable multivibrator is normally used only when a fair amount of on-time stability is sufficient—perhaps a 10-percent on-time stability with fixed values, and perhaps 1 percent with adjustable values and calibration at the operating temperature and with a regulated power supply. For circuits that require greater than 1-percent stability, a solid-state phantastron or Miller integrater circuit is called for. For better than 0.1-percent stability, you must go to time-base techniques using digital dividers such as those discussed in Chapter 8.

CONTACT CONDITIONERS

Mechanical contacts must be conditioned to "make" noiselessly and without bouncing if they are to be used with any high-speed electronic counter. Circuits to do this are called *contact conditioners* and take the form of either a monostable multivibrator or a half-monostable type of circuit.

Fig. 3-15 shows several contact-conditioning circuits. In Fig. 3-15A, a single make contact is conditioned with a 10-millisecond monostable. At first, the circuit behaves as a "real" monostable, but only the output inverter returns to its normal (grounded) state until the contact is broken. The circuit thus produces a sharp fall at output A and a sharp rise at output B. The trailing edge of the pulse at B does not fall sharply, and the trailing edge at output A does not occur until the contact at the input of inverter A is released. The choice of a 10-millisecond on time is adequate for many mechanical contacts and practically all push buttons.

In Fig. 3-15B, larger values of R and C are used to obtain a 1-second on time. Such long time constants are required if the mechanical contact actuation is slow; a slot car completing a lap is a good example of this.

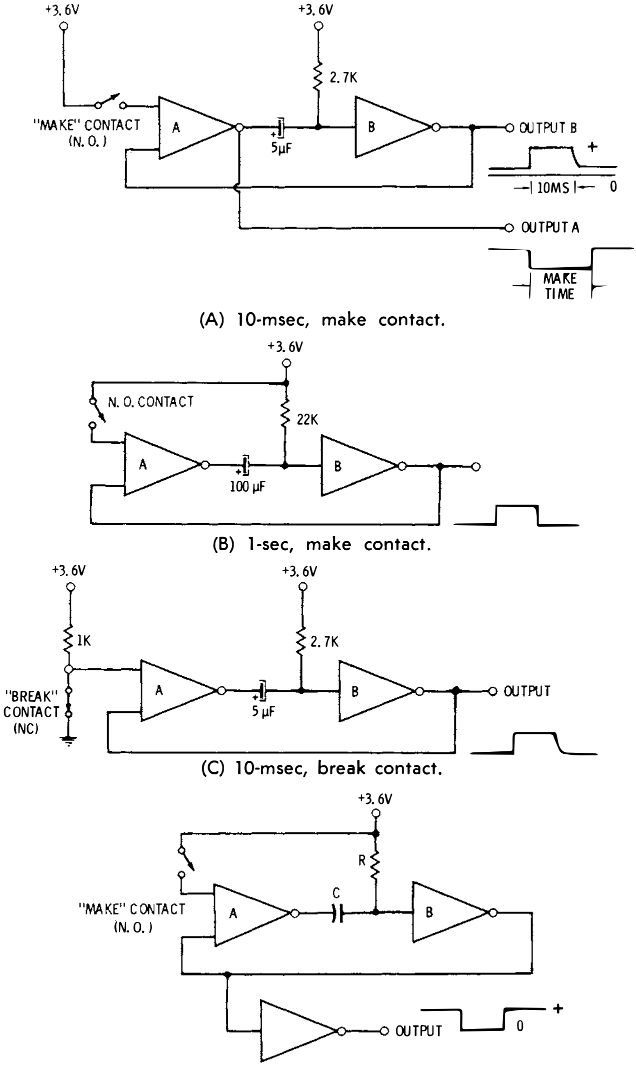
To operate from a "break" contact, the circuit of Fig. 3-15C may be used. In this circuit, the contact normally holds one input of inverter A at ground.

Finally, to obtain a sharp waveform that is grounded for the on time of the monostable instead of the entire contact closure time, an inverter may be added to form a new output as shown in Fig. 3-15D.

THE HALF MONOSTABLE

Sometimes it is possible to use a single inverter circuit that has monostable-like properties. This is called a half-monostable circuit and is shown in Fig. 3-16A. There are several restrictions to this simple circuit, but when these restrictions can be met, the half monostable is one of the simplest and most useful RTL pulse generators.

Just as in the monostable circuit, the half monostable relies on the fact that you cannot instantaneously change the charge on a capacitor. In the half monostable, the input must be positive during the time before triggering occurs (Fig. 3-16B). During this time, capacitor C charges up to this positive input voltage, less the 0.6-volt base-to-emitter forward drop inside the inverter. To trigger the half monostable, the input positive voltage is abruptly brought to ground and held there. Since the charge on C cannot change instantaneously, the right end of C must swing *negative* by the same amount the left end did, placing a *negative* voltage on the inverter. This action reverse-biases the inverter, turns it off, and lets the output immediately swing positive. Conditions remain this way until R can charge C back up to the 0.6 volt positive necessary to produce base-emitter current and turn the inverter on, grounding the output. The net result is a short positive output pulse that appears the instant a longer input waveform drops to ground and stays there.



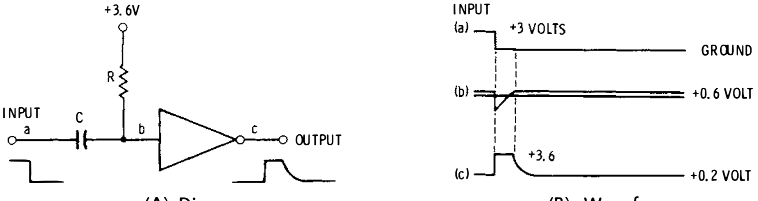
(A) 10-msec, make contact.

(B) 1-sec, make contact.

(C) 10-msec, break contact.

(D) Inverter added to output.

Fig. 3-15. Conditioners for mechanical contacts.



(A) Diagram. (B) Waveforms.

Fig. 3-16. Half monostable.

Restrictions on the Half Monostable

The restrictions and limitations of the half monostable are much more severe than on a "real" monostable. The input signal must start at a positive voltage and drop sharply to ground in a time that is short compared to the on time; it must stay at ground for the entire on time. Once the input swings positive again, it must stay positive until the next retriggering.

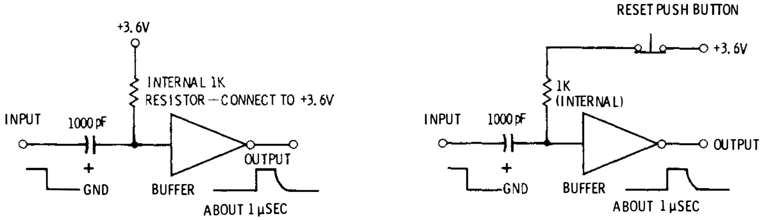
There is no feedback or regeneration in this circuit. The trailing edge of the output pulse will always be slow to fall, usually being from 1/10 to 1/5 of the on time in duration. Only the positive-going output is available.

The on-time accuracy will be only 20 percent or so and will vary directly with the amplitude of the input signal. If a half monostable is driven from a counting flip-flop, 0.4 RC may be used as a rough approximation to the on time. You can obtain suitable resistor and capacitor sizes simply by using Fig. 3-12 and *doubling* the final capacitor size.

Buffers as Half Monostables

The half monostable is simplified further by using a buffer and its internal 1k resistor, just as was done in Fig. 3-14. Either a mechanical or electronic input may be used.

A reset-pulse generator for an electronic counter is shown in Fig. 3-17. Since this circuit requires only one buffer and one capacitor, it is a very attractive circuit that sees wide use. There are numerous important applications, such as resetting a frequency counter at the beginning of a 1-second time gate, clearing the old number in a digital voltmeter before a new



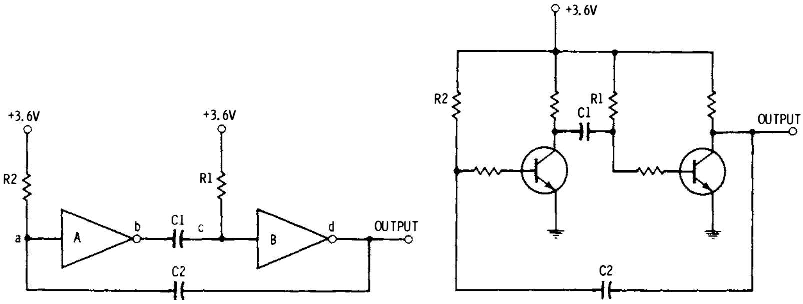
(A) Generator electronically triggered. (B) Electronic or mechanical triggering.

Fig. 3-17. Reset-pulse generator.

measurement is made, taking a 15,750-Hz square wave and producing a 5-microsecond television horizontal-sync pulse once each line, producing a time delay long enough for other gates and flip-flops in the system to catch up and get back in the right order, and many more.

THE ASTABLE MULTIVIBRATOR

Suppose we once again return to the back-to-back inverter connection, except that this time we insert series timing capacitors and recharging resistors in *both* feedback paths. Now *both* states will be unstable, and we have an *astable multivibrator* that keeps flipping back and forth by itself, generating its own square- or rectangular-wave output. Astable multivibrators (often called "astables") are used as oscillators, signal generators, and audio-tone generators, and for novelty audio devices such as sirens, "panic buttons," and back-up alarms. Fig. 3-18 shows the RTL astable connection and its discrete equivalent.



(A) Diagram for RTL.

(B) Discrete equivalent.

Fig. 3-18. Astable multivibrator.

Suppose we momentarily ground the input to inverter B. The output of this inverter swings positive, and capacitor C2 charges, positive to the right. Inverter A is on, receiving base current through resistor R2; the output of inverter A is grounded. Now, we remove the short. Capacitor C1 charges through resistor R1 until the right end of C1 reaches +0.6 volt; at this time inverter B turns on. The output of inverter B suddenly goes to ground. Once again, a charged capacitor cannot discharge instantaneously—the left end of C2 must swing *negative*, turning off inverter A. The circuit has changed state.

The charge on C2 cannot hold inverter A off forever, for the left end of C2 starts charging in a positive direction through resistor R2. Meanwhile, capacitor C1 is charging to the supply voltage, positive to the left. When the left end of C2 reaches +0.6 volt, inverter A turns on, and its output goes to ground. The circuit has changed state, for now the charge on C1 cannot change instantaneously, and inverter B is forced off by the reverse bias caused when the right end of C1 suddenly swings negative.

The foregoing sequence of events repeats during every cycle, and no input signals are required to make an astable continuously flip from side to side. The frequency of the back-and-forth flipping is determined by the RC products. Fig. 3-19 shows the waveforms in detail.

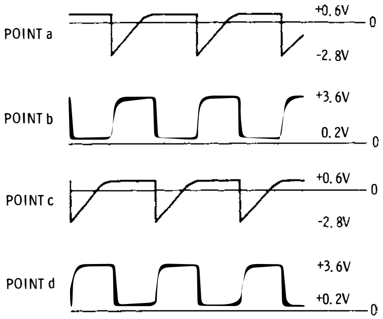


Fig. 3-19. Waveforms of astable multivibrator.

Astables normally find use anytime we need a signal source adjustable over a wide frequency range, or anytime we need a fixed, low-cost square-wave oscillator circuit that does not have to be extremely accurate or stable. We also can use external voltages to control the frequency of an astable. If we do this on a continuous basis, we form a *voltage-controlled oscillator.* If we use a two-step input voltage, we obtain a *two-tone* frequency source.

Controlling Frequency

We can adjust R and C, either together or separately, to control the frequency of an astable multivibrator. Values of C may range from 330 pF to several thousand microfarads, and R may range from 3k to 30k with a low-power gate, from 2k to 25k with a medium-power gate, and from 1k to 10k with a buffer. The usable frequency range extends from one cycle every few seconds to over one megahertz.

The formula for the period of an astable multivibrator is simply:

$$T = 1.4\,RC$$

where,

T is the period in milliseconds,
R is the resistance in kilohms (thousands of ohms),
C is the capacitance in microfarads,
$R = R1 = R2$,
$C = C1 = C2$.

This formula may be used whenever $R1 = R2$ and $C1 = C2$. (R1, R2, C1, and C2 are identified in Fig. 3-18.)

The frequency of an astable is given by the inverse of the period, or:

$$f = \frac{1}{1.4\,RC}$$

where,

f is the frequency in kilohertz,
R is the resistance in kilohms (thousands of ohms),
C is the capacitance in microfarads,
$R = R1 = R2$,
$C = C1 = C2$.

Fig. 3-20 is a chart that permits finding any desired frequency, period, or component value, once again assuming that R1 = R2 and C1 = C2. For instance, with two 10k resistors and two 0.1-microfarad capacitors, the period would be about 1.4 milliseconds, and the corresponding frequency would be about 700 Hz.

Loading

Any loading at all on either output of the astable multivibrator will cause the frequency to change, and heavy loading will surely stall the oscillator. There are several ways to obtain load isolation, some of which are shown in Fig. 3-21. In Fig. 3-21A, a low-power gate is added directly to a medium-power astable; if only medium-power RTL is available, a series 2.2k resistor can be added to insure light loading (Fig. 3-21B). Either of

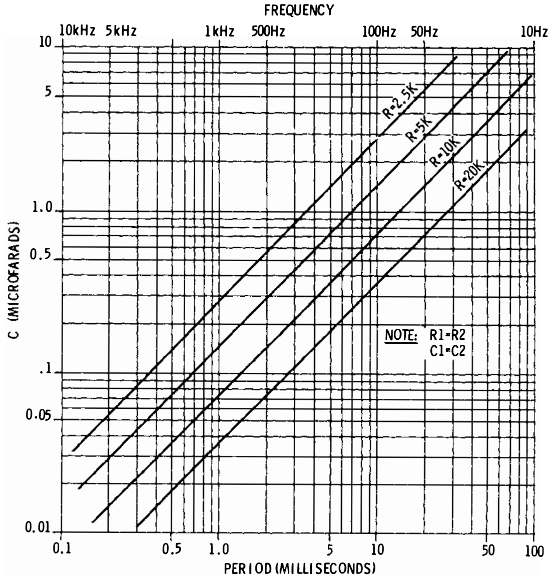
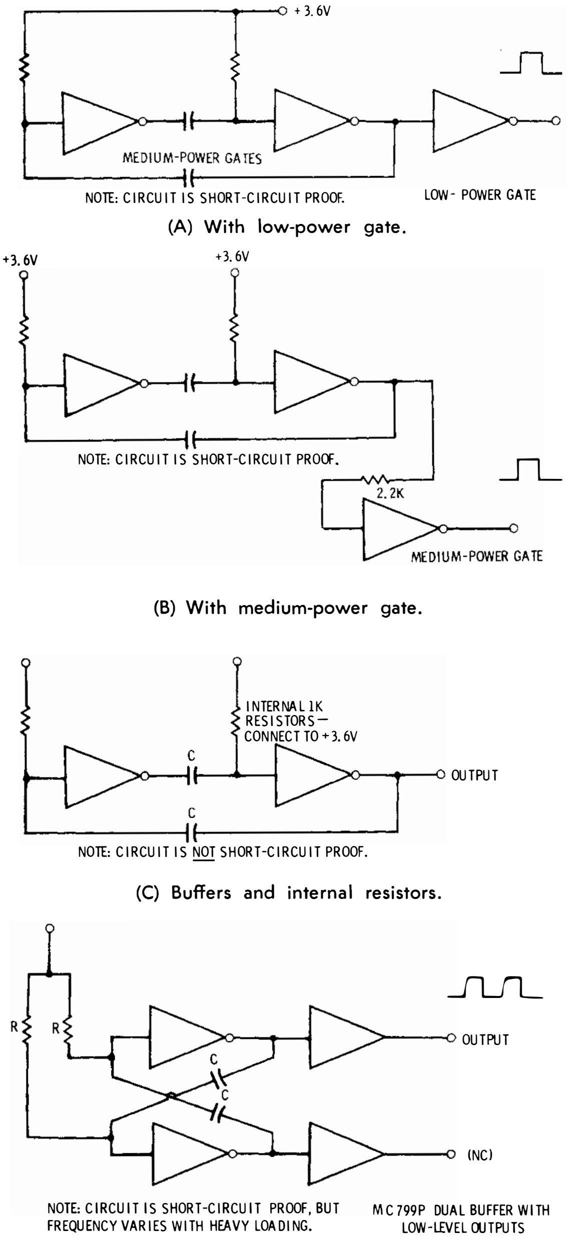


Fig. 3-20. RC chart for astable multivibrators.



NOTE: CIRCUIT IS SHORT-CIRCUIT PROOF.

(A) With low-power gate.

NOTE: CIRCUIT IS SHORT-CIRCUIT PROOF.

(B) With medium-power gate.

NOTE: CIRCUIT IS NOT SHORT-CIRCUIT PROOF.

(C) Buffers and internal resistors.

NOTE: CIRCUIT IS SHORT-CIRCUIT PROOF, BUT FREQUENCY VARIES WITH HEAVY LOADING.

(D) Dual two-output buffer.

Fig. 3-21. Load-isolation techniques.

these two connections will "square up" the trailing edge of the output waveform considerably. Also, any varying load on the output of the isolating gate will have practically no effect on the frequency of the astable.

Another approach is shown in Fig. 3-21C. Here, buffers are used as one or both of the inverters in the astable. This arrangement makes available some fan-out, perhaps 8 to 12 on a buffer with a normal fan-out of 80, and allows the use of the internal 1k resistors to save two external parts.

One interesting way of "having your cake and eating it, too" is to use a buffer that has an internal lead brought out (Fig. 3-21D). The capacitors are cross-coupled from the low-level outputs, and the high-level outputs are still available for load drive. This particular circuit results in a relatively long rise time and a sharp fall time of the output waveform; also, heavy loading changes the frequency. The circuit is short-circuit proof; that is, shorting the high-level output terminals will not stall the astable.

Just as with the monostable, the frequency of the astable is not highly dependent upon the supply voltage or the temperature, although a 10-percent frequency variation may be expected as the supply voltage goes from 1.5 to 4.5 volts.

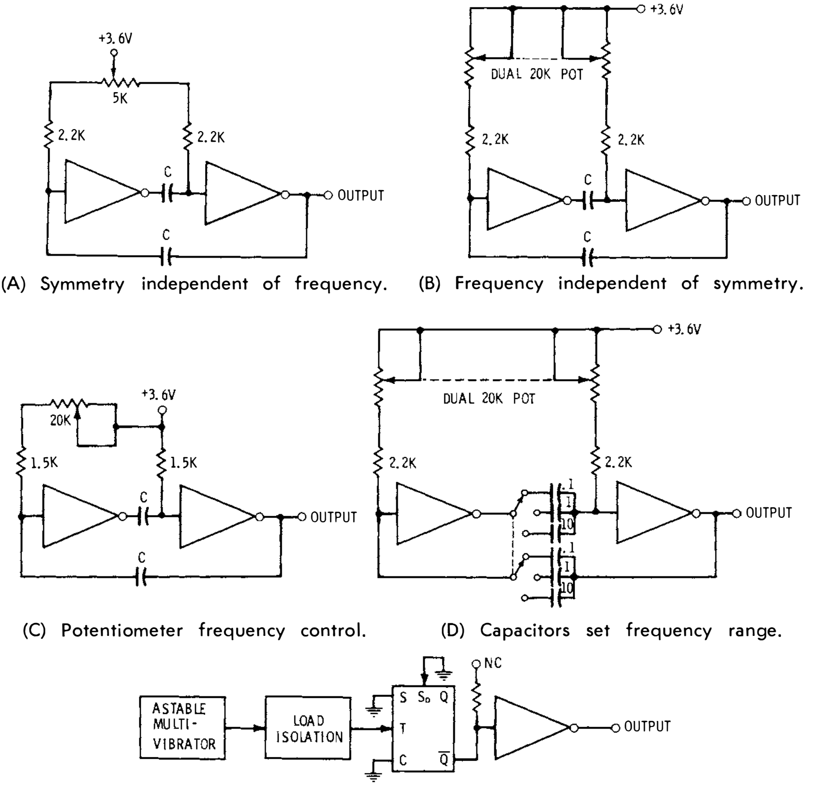
Symmetry and Timing Adjustments

The frequency of the astable can be controlled by changing any combination of the four resistors and capacitors. If equal-value resistors and equal-value capacitors are used, the output waveshape will be "up" for as long as it is "down." The duty cycle is 50 percent, and the output waveform is *symmetric.* If the resistors or capacitors differ in value, the output will be nonsymmetric and have a duty cycle greater or less than 50 percent. Symmetry and frequency adjustments can interact drastically on an astable; you have to investigate the effects on both caused by a change in any component.

The capacitors should always be nearly equal in value, or at least within a factor of three of each other. If a more asymmetric output waveshape is desired, it may be obtained by varying either R1 or R2.

Several techniques for controlling frequency and symmetry are shown in Fig. 3-22. In Fig. 3-22A, identical resistors, identical capacitors, and a symmetry potentiometer are shown. This connection permits adjusting the symmetry without greatly affecting the frequency. The relative values of fixed and variable resistance determine the available asymmetry. For instance, with 2.2k fixed resistors and a 5k potentiometer, we can vary the effective resistance on either side from 2.2k to 7.2k, smoothly varying the asymmetry from approximately 3:1 through 1:1 to 1:3.

For a symmetric output waveform over a wide frequency range, the dual-potentiometer circuit of Fig. 3-22B may be used. Here, the effective resistances of the two variable elements are always the same for a given setting, and the symmetry stays nearly constant. The choice of fixed versus variable resistance determines the range of frequency adjustment available.



(A) Symmetry independent of frequency.

(B) Frequency independent of symmetry.

(C) Potentiometer frequency control.

(D) Capacitors set frequency range.

(E) JK flip-flop used as divider.

Fig. 3-22. Symmetry and frequency adjustments for astable.

If a single potentiometer is used, there will be a wide frequency variation, but the output symmetry will vary drastically. With the circuit shown in Fig. 3-22C a 10:1 range may be obtained. Frequency calibration versus knob rotation will be very nonlinear, but helped somewhat if a special-taper potentiometer is used. This behavior is typical of any wide-range circuit of this type.

We can also switch the capacitors to change frequency. It is usually best to have identical values for both capacitors at any given time (Fig. 3-22D). Decade (×10) changes in capacitance will cause decade changes in output frequency. It is possible to combine a continuous resistance control with a switched bank of decade capacitors to obtain continuous coverage from subaudio frequencies to almost 2 MHz.

There is one sure-fire way to guarantee a 50-percent duty cycle. Just start with an astable producing twice the frequency you want, and divide its output frequency by two with a JK flip-flop. It will be shown later that a JK flip-flop only responds to negative toggle transitions and can be told to change its output states once for each negative transition. This provides a perfectly symmetrical waveform with excellent rise and fall times. We can even go one step further and add a buffer to the flip-flop output; the result is a load-independent, short-circuit proof, symmetrical astable multivibrator with sharp rise and fall times. Fig. 3-22E shows the details. The multipurpose MC787P combines an inverter, two buffers, and a flip-flop—just what is needed to handle this job.

Starting and Stalling Problems

If we short the unisolated output of an astable multivibrator for a while, there may result a state in which there is very little charge on either capacitor and both outputs are grounded. The circuit will not oscillate in this condition, and we say the astable has *stalled.* Also, at least in theory, if we bring the supply voltage up too slowly, there is no reason to expect the circuit to ever start oscillating.

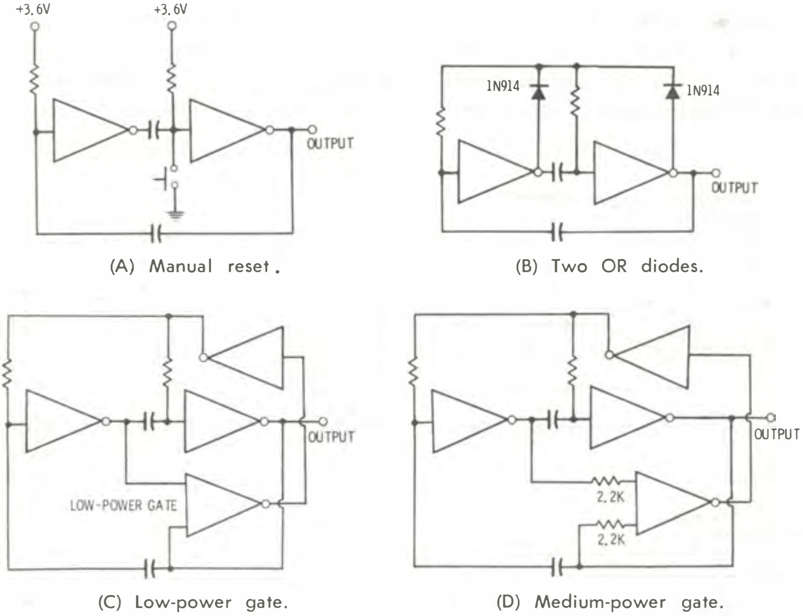
Stalling and failure to start were often severe problems in conventional astable circuits, but the newer load-isolated RTL circuits rarely have a stalling problem. The load isolation effectively eliminates any possibility of an output short. Starting is also no problem with normal RTL circuits. An "unprotected" astable made of RTL can be counted on to start 99 percent of the time.

If the astable absolutely must start every time, suitable antistall and protection circuitry is sometimes justified. There are several antistall protection circuits; some of the more common ones are shown in Fig. 3-23. In Fig. 3-23A, a push button has been added to the input of one inverter. If the oscillator ever does stall, momentarily depressing the button will force inverter B off and allow oscillation to resume.

All-electronic protection requires that the output of one inverter be positive at any given time. One simple way to accomplish this is to OR the output waveforms and allow them to provide a source of charging current for the timing resistors. As long as either output is positive, charging current and base current are provided the astable. Should both outputs go to ground, all base current is removed, automatically forcing operation to resume. In Fig. 3-23B, two conventional diodes have been added to OR the outputs and provide a source of charging current. In Fig. 3-23C, the same thing is done with a low-power gate and an inverter, once again providing base and recharging current only when the output of one inverter is positive. Fig. 3-23D shows the use of a medium-power gate and an inverter. The brief transition times between states do not markedly affect the OR-circuit output; during transition, base current is provided by positive feedback, and the momentary lack of current through the recharging resistors is not significant. If desired, a small capacitor may be added to the output of the OR circuit.

VCO Operation

Fig. 3-24 shows several ways to attain electronic remote control of the frequency of an astable. In Fig. 3-24A, a positive dc control voltage of 0-5 volts is applied to the recharging resistors as shown. As the voltage varies, the recharging current varies, and the frequency of the astable changes accordingly. As the voltage increases, the frequency increases.



(A) Manual reset.

(B) Two OR diodes.

(C) Low-power gate.

(D) Medium-power gate.

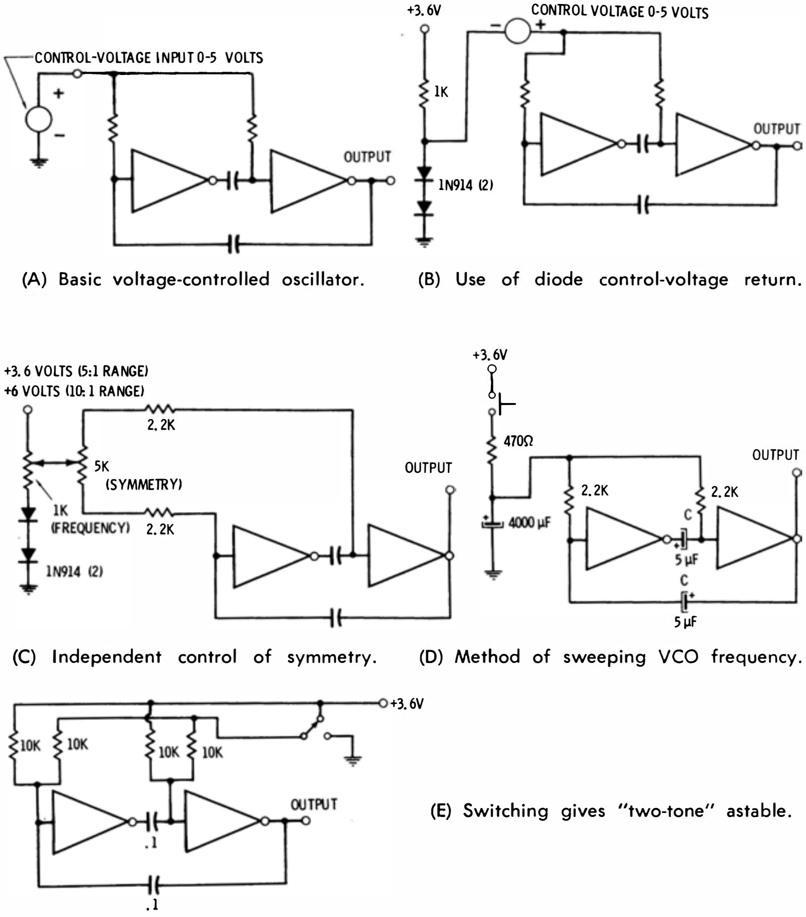
Fig. 3-23. Antistall connections.

For very low control voltages, the frequency may rise again as the voltage is lowered below a critical point. This change is caused when the astable switches to a mode of operation in which base current is provided on a pulse basis through the timing capacitors instead of through the timing resistors. The effect can be eliminated by providing a control voltage greater than the critical value at all times.

In Fig. 3-24B, the input control voltage is returned to two forward-biased silicon diodes. The minimum input voltage seen by the astable is now +1.2 volts, and the control voltage is added to this +1.2 volts, causing the astable to operate through a wide range without the low-voltage turnaround problem. This technique suggests using a potentiometer from the supply to +1.2 volts as a frequency control. An astable whose frequency and symmetry are nearly independent of each other and separately adjustable is shown in Fig. 3-24C.

We can apply a sawtooth to the input control point and produce a swept astable frequency, provided the sweep time is considerably longer than the duration of one cycle at the lowest frequency to be produced. Otherwise, a phase modulation instead of a frequency modulation results. Simply applying a dc voltage through a resistor to a large value electrolytic capacitor (Fig. 3-24D) is an effective but nonlinear way of sweeping an astable. You can use this connection as an electronic panic alarm, a synthetic siren, a wolf whistle, or any similar novelty audio device.

The frequency of the astable also can be changed by switching a timing resistor on each side between ground and the supply voltage (Fig. 3-24E). The result is a two-tone generator whose upper frequency is determined by the parallel combination of the two resistors, and whose lower frequency



(A) Basic voltage-controlled oscillator.

(B) Use of diode control-voltage return.

(C) Independent control of symmetry.

(D) Method of sweeping VCO frequency.

(E) Switching gives "two-tone" astable.

Fig. 3-24. VCO astable operation.

depends on the ratio between the switched and fixed resistors. The switching may be done either mechanically or electronically.

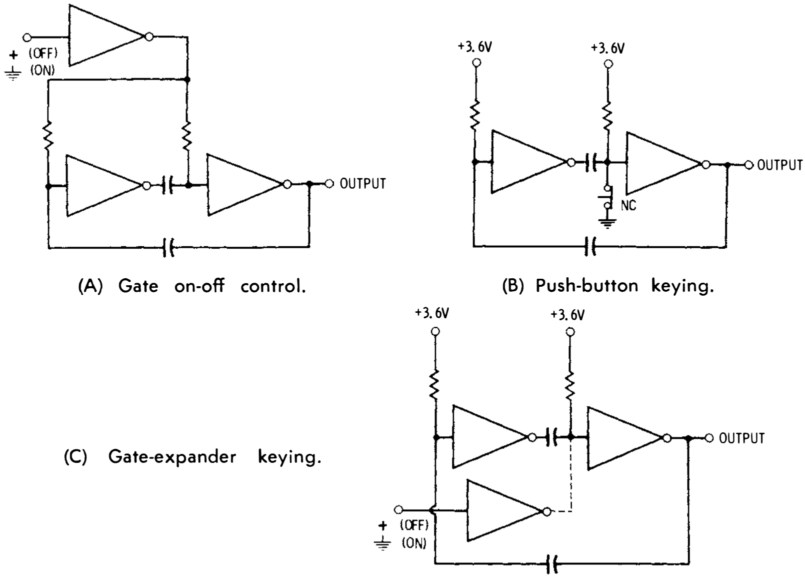
Keying

Sometimes we like to start and stop an astable at will—perhaps manually in an electronic dice game that "rolls" the dice only when a push button is depressed, or perhaps electronically to synchronize the start of the oscillator with some other event. The process is called *keying*, and several circuits are shown in Fig. 3-25.

In Fig. 3-25A, supply power for the recharging resistors is derived from the output of a control gate. A positive input signal *stops* the astable; a grounded one *starts* the astable. In Fig. 3-25B, a push button normally shorts out the input to inverter B. Depressing the button allows the oscillator to run normally; releasing it stops the oscillator. This circuit also guarantees starting, and the push-button bounce will not affect the output waveform if the chosen astable frequency is less than about 2 kHz. In Fig. 3-25C, the same thing is done with a gate expander. This circuit is also sure starting, allowing oscillation when the expander inputs are all grounded and stopping oscillation when the expander input is made positive.

Synchronization and Frequency Division

The astable can be synchronized to narrow timing pulses by using either of the synchronizing circuits of Fig. 3-26. In either circuit, the astable is

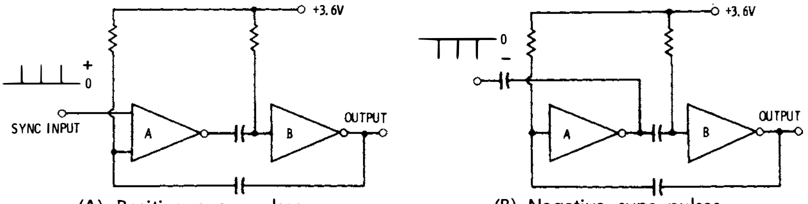


(A) Gate on-off control.

(B) Push-button keying.

(C) Gate-expander keying.

Fig. 3-25. Methods of keying an astable.



(A) Positive sync pulses.

(B) Negative sync pulses.

Fig. 3-26. Synchronization of an astable.

adjusted to be slightly low in frequency. The sync pulses bring about premature state transitions, raising the astable to the desired frequency. In Fig. 3-26A, narrow positive sync pulses are applied to the second input of gate A; in Fig. 3-26B, negative sync pulses are applied to the output of gate A. The first method is preferable, but the circuit in Fig. 3-26B can also be used for frequency division—reliably from 2 through 5, and perhaps with larger divisors if care is exercised.

Chapters 5 and 6 will show that a far more powerful frequency-divider tool is available in RTL—the JK Flip-Flop. Without any external parts, this device can be made to divide reliably by any number; it divides in a manner that is totally independent of frequency and such that it is impossible to "miss" the proper division ratio by one or two counts. While relaxation-oscillator frequency dividers may be built using RTL, they have all the problems discrete and unijunction circuits have—need of adjustment, many parts, critical supply voltages, and the possibility of jumping count and dividing by a new number.

When Astables May Be Used

Astables are used only where frequency stability and accuracy are not highly important. Frequency stability of 20 percent is quite easy to achieve, and 10-percent and 5-percent values are readily attainable. Holding the frequency closer to a desired value than these tolerances is difficult, even with constant-temperature and regulated-supply operation. One percent is the best that can be expected.

If better frequency stability is required, other signal sources must be used. Conventional Hartley and Wien-bridge oscillators may be called upon where the accuracy is to be within the 0.1-to-1 percent range. For greater stability than this, the reference-frequency and time-base techniques described in Chapter 8 are used. With these techniques, adjustable or crystal-controlled square waves of any desired frequency may be produced with 0.01-percent and better accuracy and stability.

When an astable is swept, any of the resistor-capacitor timing schemes (Fig. 3-24D) will result in an exponential rather than a linear sweep. Linearity can be improved considerably by using true current sources instead of recharging resistors. These take the form of upside-down pnp transistors returned to the positive supply, conventional FET circuits, or constant-current diodes.

OTHER MULTIVIBRATOR-TYPE CIRCUITS

There are a few other circuits that are not as extensively known or used as the basic multivibrators, but that can be used to solve several thorny design problems. These circuits include the negative-recovery monostable, the frequency doubler, the Schmitt trigger, the squaring circuit, and the time delay.

Negative-Recovery Monostable

The negative-recovery monostable (Fig. 3-27A) is triggered as an ordinary monostable, but it provides a time delay *after* the *last* trigger pulse arrives. It essentially possesses a negative recovery time, for every time the circuit is triggered, the output of inverter A grounds capacitor C and turns off inverter B. Capacitor C must then recharge through the collector resistor of inverted A until it reaches 0.6 volt positive; then inverter B once again turns on. If the monostable is retriggered *before* the voltage across C has reached +0.6 volt, the charge is dumped, and the capacitor has to start charging again. Only after the last trigger pulse arrives, or only when there is a long enough spacing between trigger pulses, will the voltage reach +0.6 volt and allow the output of inverter B to drop once again to ground.

This circuit may be used for missing-pulse detection, speech analysis, and voice-controlled relay systems. The timing accuracy is relatively poor and dependent upon temperature and supply variations. For a 3.6-volt supply and a medium-power gate, a rough estimate of the equivalent on time is given by the formula:

$$T = \frac{1}{6} C$$

where,

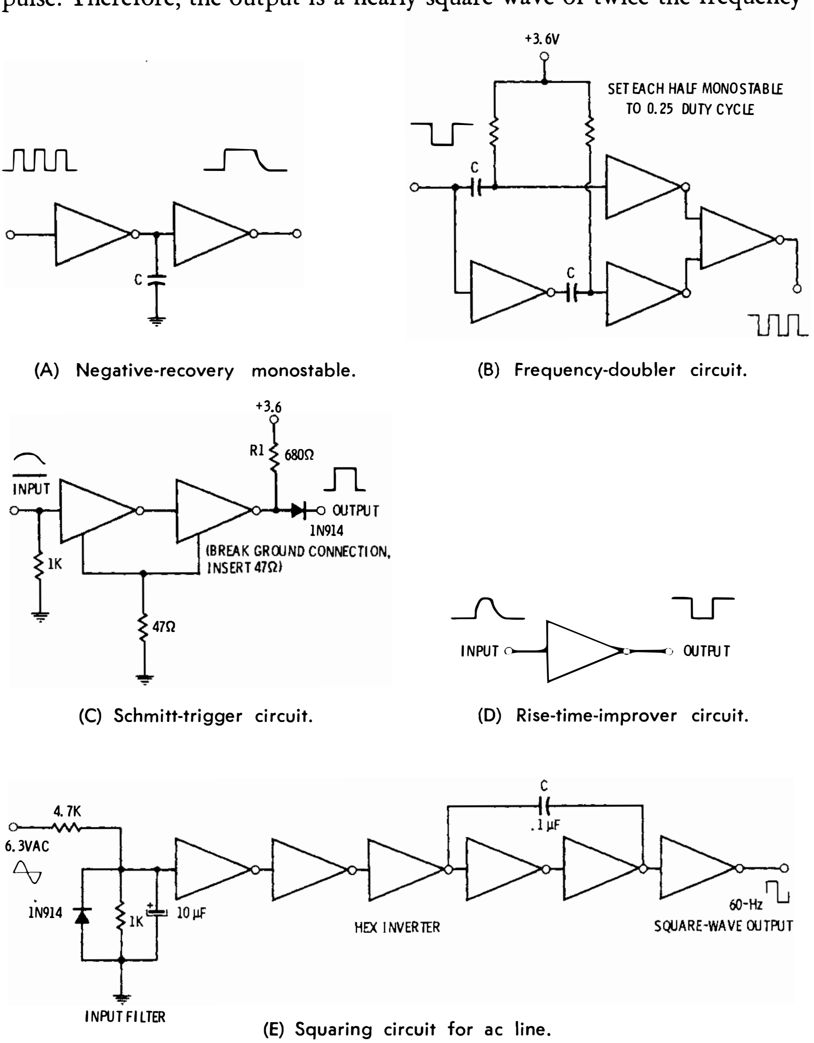
T is the on time in milliseconds,
C is in microfarads.

This on time will also vary with input-pulse duration, and input trigger pulses shorter than 1/10 of the on time should not be used.

This circuit requires considerably more capacitance than an ordinary monostable does, but it requires no recharging resistor and does possess negative recovery time. It has no regeneration or positive feedback, and thus a poor output fall time results. Only the positive-going on time is available; another inverter may be added to get an output that is grounded during the on time. Adding this inverter also improves the rise and fall times.

Frequency Doubler

We can double, quadruple, or octuple the frequency of an input square wave with one, two, or three circuits such as the one shown in Fig. 3-27B. Both the input square wave and its complement (internally generated with the inverter shown) are applied to half monostables which have an on time equal to *one-fourth* of the input period. Each time one of the half monostables delivers a positive-going pulse, the OR gate delivers a negative-going pulse. Therefore, the output is a nearly square wave of twice the frequency



(A) Negative-recovery monostable.

(B) Frequency-doubler circuit.

(C) Schmitt-trigger circuit.

(D) Rise-time-improver circuit.

(E) Squaring circuit for ac line.

Fig. 3-27. Multivibrator-type circuits.

of the input. As the input frequency varies, the symmetry of this doubling process changes somewhat, but operation over a fairly wide frequency range is entirely feasible. To quadruple, set the on time of the second doubler to one-fourth the *output* period of the first doubler, and so on.

Schmitt Trigger

The Schmitt trigger (Fig. 3-27C) is an emitter-coupled type of monostable. No more than a dual two-input gate can be used for this circuit, because the normal emitter ground connection must be broken and the 47-ohm resistor inserted. The Schmitt trigger serves as a level detector with memory. This particular one switches the output positive when the input exceeds 1.5 volts and switches the output to ground when the input goes below 1.2 volts. This circuit is useful for conditioning a noisy input signal or for squaring up a signal with a poor rise time. The hysteresis of the circuit is somewhat load dependent, and resistor R1 should be optimized for a specific load.

Rise-Time Improver, or Squaring Circuit

Every time a signal goes through an inverter, the rise time and fall time improve because of the gain of the inverter as it swings through its active region. As the input swings from 0 to +3.6 volts, nothing happens to the output until the input reaches +0.6 volt; then in the next few tenths of a volt, the output rapidly goes to ground and stays there, even if the input waveform is still a long way from its maximum value.

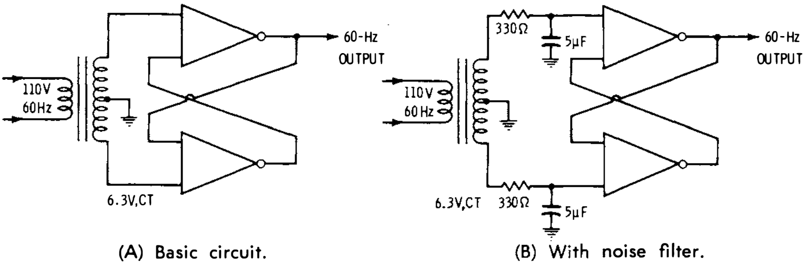
This is a useful technique for "cleaning up" a "sloppy" pulse, eliminating any corner rounding caused by capacitors in monostable circuits, or for use anywhere else we want to obtain faster rise and fall times.

As many inverters as desired can be cascaded. However, the best performance that can be expected is a 20-nanosecond (20 billionths of a second) rise or fall time at the output of the last inverter. One good choice is to use an entire cascaded hex inverter. This circuit is handy for squaring up power-line and other low-frequency audio signals. Chapter 5 will show that JK flip-flops must be triggered by an input signal that abruptly drops once per cycle with a 10- to 100-nanosecond fall time. A hex inverter is ideal for converting the power-line waveform to a square wave of this type (Fig. 3-27E). Usually, it is a good idea to add an RC filter to eliminate any high-frequency noise at the input. Capacitor C in Fig. 3-27E provides a "bootstrapping" type of positive feedback that further improves the output waveshape; it should be used only with inputs of 1 kHz or less.

Both the squaring circuit and the Schmitt trigger are effective and inexpensive means of squaring and conditioning outside-world signals. Both require a volt or two of rms input amplitude. In cases where lower-level signals or particularly noisy signals are encountered, more elegant conditioning circuits, called *comparators*, are needed. These are covered in Chapter 4.

Time Delay

Every time a signal goes through an inverter, it receives around 20 nanoseconds of time delay in addition to the inversion, squaring up, and fan-out improvement. Sometimes one signal will get ahead of another in a digital system. If this happens, we can simply add as many gates as required to put things back in order. An *even* number of gates produces time delay and *no* inversion; an *odd* number gives time delay *with* inversion. As always, all unused gate inputs should be grounded.



(A) Basic circuit.

(B) With noise filter.

Fig. 3-28. Power-line squaring circuits.

Power-Line Squaring Circuit

A set-reset flip-flop may be combined with a center-tapped filament transformer to produce 60-Hz square waves with excellent rise and fall times. Such square waves are often needed for driving JK flip-flops in photo timers, digital volt-ohmmeters, or frequency-counter time bases. In Fig. 3-28A, the positive half-cycles set the flip-flop, and the negative ones reset the flip-flop. Positive feedback caused by the cross coupling produces a snap action with a fall time of about 40 nanoseconds. Available fan-out with a medium-power gate is 14. The power transformer used in the instrument can often do double duty as a timing source.

In Fig. 3-28B, a noise filter has been added for use in extremely noisy areas or if extreme accuracy is required. Normally this filter is not necessary unless there is need for the full ±0.05-percent accuracy the power line nominally provides.

CHAPTER 4

Linear Circuits and Techniques

A *linear* circuit responds in a *proportional* manner to an input signal. Instead of just an on-off or a 1-0 situation, we can have any possible input and output signal levels (within limits) on an instantaneous basis. This chapter will be concerned with several linear applications of the RTL digital circuits. It offers answers to questions such as: How do we convert a digital RTL gate into a linear amplifier so that we can build such things as crystal oscillators, operational amplifiers, comparators, and input amplifiers? How do we convert a variable-amplitude input signal to an on-off digital signal and still obtain good sensitivity and noise rejection? How do we work things backward when we want to convert a digital signal into variable-amplitude form, as in an fm discriminator or a tachometer, dwell meter, or frequency meter?

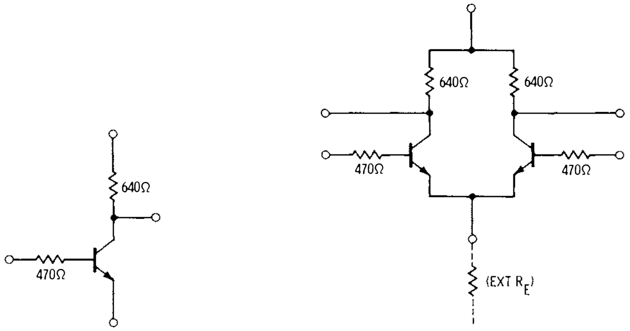
RTL GATES AS LINEAR AMPLIFIERS

While RTL IC's were designed and specified only in terms of their digital on-off capabilities, these same devices also can be thought of as nothing but small plastic boxes full of transistors and resistors. Any external bias connection that forces the transistors into their *active*, or *linear*, region *continuously* will result in ordinary amplification. Further, the digital and linear performance of any semiconductor are related. Since we obtain uniform results *digitally* with RTL, we can expect reasonably consistent linear operation from the same devices—even though the manufacturers do not specify or guarantee this type of performance. While RTL circuits operated in the linear mode are far from ideal, these circuits often serve as economical solutions to linear-amplification problems, particularly when the problems are directly related to a digital system. Also, they may be useful when extremely low cost or low-voltage operation is important.

The usual choice of a linear building block is either the hex inverter or the dual two-input gate. In the case of the two-input gate, one input of each gate is connected permanently to its own emitter, thus disabling the extra inputs. Fig. 4-1 shows two possible configurations around which some practical amplifiers can be built. Fig. 4-1A shows the inverter, or gate, arranged to form a *common-emitter*, single-transistor inverting amplifier. All we need add is suitable bias and input/output bias isolation, and we have an ac amplifier that is useful from very low audio frequencies to several megahertz. In this circuit, the emitters often can be tied directly to ground, and one or two inverters or gates "left over" from digital work elsewhere in a system may be used.

Fig. 4-1B shows a two-transistor amplifier called a *differential amplifier.* This amplifier requires a negative power supply, and the emitters in the IC are removed from ground, but the circuit has important advantages. It will amplify dc and do it stably, it can serve as a limiter, and its gain is easily controlled.

Before looking at these two circuits more closely, it is necessary to take a closer look at the characteristics of RTL when operated in the linear mode.



(A) Common emitter.

(B) Differential circuit.

Fig. 4-1. RTL linear-amplifier configurations.

CHARACTERISTIC CURVES

Since the manufacturers of RTL do not normally specify or guarantee any linear characteristics, it is necessary to use a curve tracer and a bit of lab work to get this type of data. One-sixth of an MC789P hex inverter put on a curve tracer gives a family of curves such as that in Fig. 4-2. The curves are a plot of collector current versus collector voltage, for fixed but incrementally increasing values of base current. From this particular plot, we can tell several things. The inverter breaks down at +7.5 volts or so, drawing heavy current and losing its useful amplifying properties. Below

0.3 volt, the device is saturated, and once again not amplifying properly. Useful amplification occurs between these extremes. You should not use more than a +6-volt supply with an RTL gate for linear operation, nor should you expect linear operation if the output voltage becomes closer than +0.4 volt to ground.

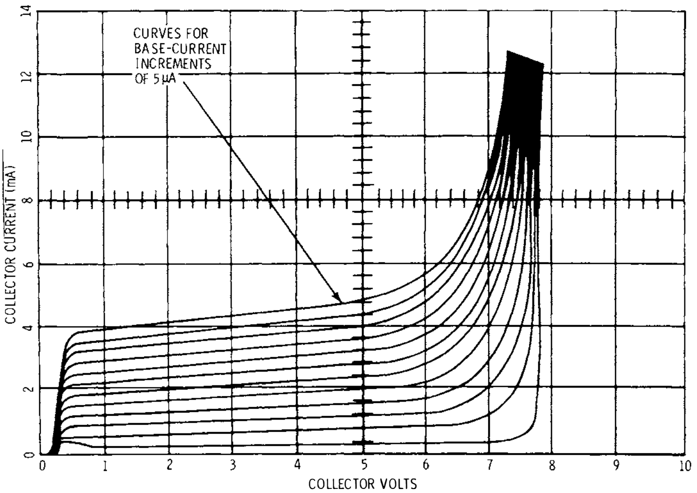


Fig. 4-2. Common-emitter breakdown characteristics of typical RTL medium-power gate.

When run *digitally*, a saturated RTL gate draws around 5 milliamperes of current. We might suspect that the best linear operation would also be obtained for similar collector currents. In Fig. 4-3, the characteristic curves have been expanded to include only the "best" amplifying region—from 0 to 5 volts on the collector, and from 0 to 7 milliamperes of collector current.

Fig. 4-3 could be used for load-line analysis, or other graphic-analysis techniques could be used to find the gain, operating point, power dissipation, and output swing of any amplifier. However, most of the time, we are interested in a specific case. Inside the IC is a 640-ohm collector resistor. This is usually the output resistor and usually the load across which we want the signal to develop. A resistive *load line* can be superimposed on the RTL characteristics. If the inverter current is zero, there is no voltage drop across the resistor, and the entire supply voltage appears from collector to ground. At the other extreme, the inverter would have an emitter-collector voltage drop of zero if it were drawing maximum current (the emitter and collector would act as though shorted together). Under this condition, the entire supply voltage would appear across the *resistor*, and the current would be equal to the supply voltage divided by the resistance of the resistor. Between these extremes, the collector current and voltage are represented by a straight-line plot that obeys Ohm's law. At any given instant,

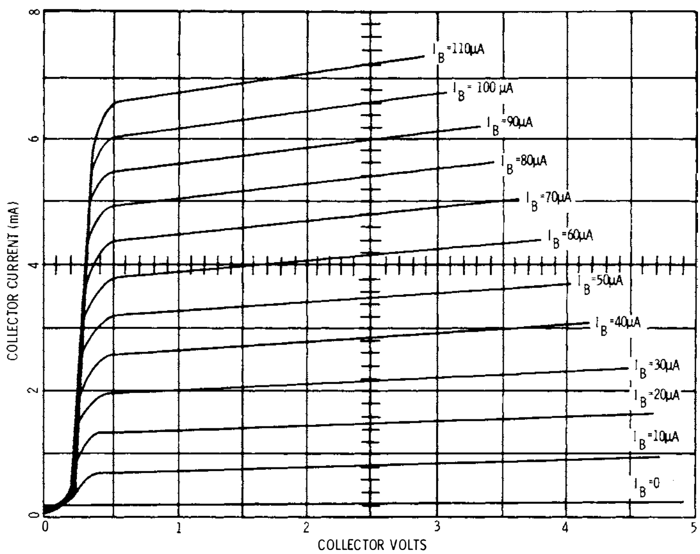


Fig. 4-3. Low-level common-emitter characteristics of typical RTL medium-power gate.

the RTL inverter *must* have its output *both* on its own characteristics *and* on the resistive load line.

In Fig. 4-4, a load line is shown for a 3.6-volt supply and a 640-ohm load resistor. (For a different supply voltage, just use a load line that is *parallel* to the one shown and that intersects the horizontal axis at the supply volt-

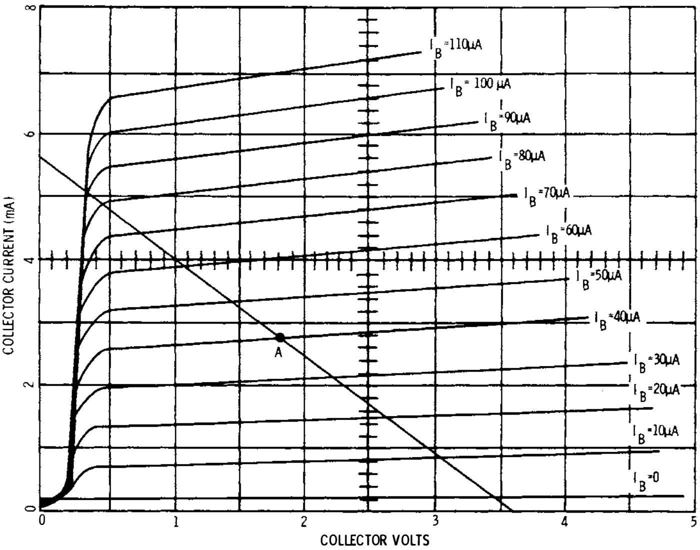


Fig. 4-4. Load line on common-emitter characteristics.

age.) Suppose we use some means to bias the inverter to point A. Point A is on both the load line and the characteristic curves, so it is a stable operating point of the inverter. At point A, 40 microamperes of base current are needed to produce around 2.8 milliamperes of collector current. By a simple division, the dc current gain of the transistor is found to be 70. We can expect around a 2:1 gain variation from device to device, with the normal "dc beta" spreading from about 50 to 100. Gain values of individual inverters or gates in the same package will usually be much closer than those of gates from different packages.

With operation at point A, the output dc level is around 1.8 volts. The maximum peak-to-peak output-voltage swing is about 3 volts, limited on the bottom by inverter saturation and on the top by the collector voltage at zero base current.

We might be tempted to change the load resistor from 640 ohms. If we could *raise* its value, we could increase the voltage gain at the expense of output impedance. One way to do this is to provide supply power to the collector of *another* transistor (with the base shorted to the emitter) and use *both* 640-ohm resistors in series as the load for one inverter to double the gain. (More about this later.) We could also use a gate expander and add a load resistor, but there probably would be little improvement over simply using a transistor instead.

HIGH-POWER OPERATION

We might try shunting the load resistor to obtain more output power, perhaps to try to drive a speaker or a panel lamp. It will not work. Fig. 4-5 shows what happens when you try to raise the output power. The curves become very poor. In particular, the saturation voltage increases and the gain decreases. For instance, at 60 milliamperes of collector current, the saturation drop is about 1.5 volts. When this voltage is subtracted from the available supply voltage, there is little left. Worse yet is the dissipation—around 90 milliwatts per inverter or, if all six are used, around 540 milliwatts per package, much more than the inverter was designed to handle. For higher-power operation, any use of RTL is not suitable at all. Incidentally, this also shows why you cannot directly drive a pilot lamp with a digital gate: the saturation drop becomes so high that there is very little voltage left to run the bulb. The result is either very low brilliance or brilliance that varies greatly from IC to IC.

CLASS-A COMMON-EMITTER AMPLIFIER WITHOUT FEEDBACK

Fig. 4-6 shows how a common-emitter amplifier can be made. To provide the point-A operation of Fig. 4-4, 40 microamperes of base current must be supplied. For this purpose, a bias resistor is connected from the base to the +3.6-volt source. The normal base voltage is around +0.6 volt, leaving

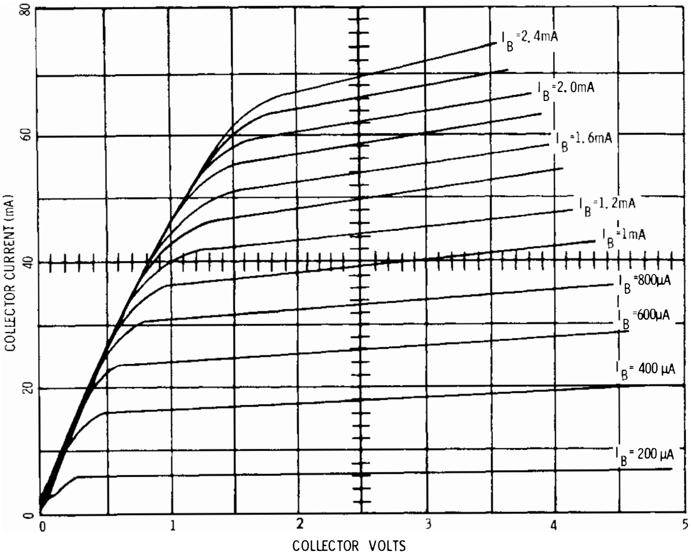
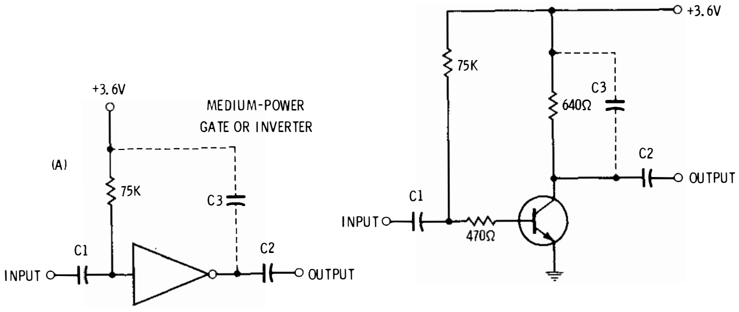


Fig. 4-5. High-level common-emitter characteristics.

3.6 − 0.6, or 3 volts, to "run" the bias resistor. By Ohm's law, the resistance should be 75k. It is necessary to capacitor couple both the input and output of this amplifier to prevent shift of either the input bias current or the dc output level.

The ac gain of this amplifier will be around 30-60, less some internal degeneration. The gain varies with temperature and the particular inverter in use, as does the output operating point. Usually we want better performance than this, but, nevertheless, we have a high-gain amplifier with only a few parts. For optimum output swing, the bias resistor should be adjusted to obtain point-A operation for the particular inverter or gate being used.



(A) Using RTL symbols.

(B) Discrete equivalent.

Fig. 4-6. Class-A amplifier without feedback.

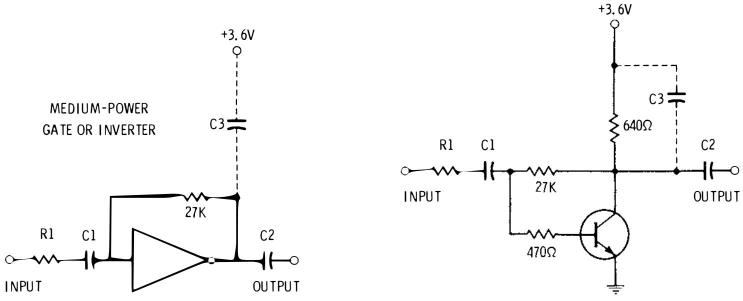
The input impedance of this circuit is around 600 ohms, as is the output impedance. Frequency response extends from a low frequency determined by C1 and C2 to an upper cutoff frequency determined by C3. Without C3, the amplifier is useful to several megahertz. The upper −3-dB cutoff frequency is determined by the frequency at which C3 has a reactance of 640 ohms; the lower −3-dB frequency is that frequency at which C1 has a reactance of about 600 ohms.

CLASS-A COMMON-EMITTER AMPLIFIER WITH FEEDBACK

The circuit of Fig. 4-6 has its limitations. The gain cannot be controlled, and the output operating point requires a custom choice of bias resistor for each individual IC used. Both of these problems are overcome by using a feedback resistor that derives the base current from the *output*, rather than from the supply. This arrangement does two things: It greatly stabilizes the output operating point, for any change in the operating point causes an opposing change in the available base current, returning the output to a reasonably stable point that does not shift excessively with changes of IC's or temperature. As will be shown later, it also permits control of the stage gain, again relatively independently of the particular IC in use and the operating temperature.

Fig. 4-7 shows this feedback connection. For point-A biasing, the output should be at 1.8 volts, and to hold it there, a base current of 40 microamperes is needed. The bias resistor "sees" 1.8 volts, less the 0.6-volt base-emitter drop, or 1.2 volts. By Ohm's law, 1.2 volts ÷ 40 microamperes is about 27k, (nearest standard value), the optimum bias-resistor value. Once again, this is an ac-only amplifier. The input and output must be capacitor coupled to prevent disturbance of the bias levels.

Theoretically, the gain of this circuit should be determined by the ratio of 27K to R1, but internal degeneration of the r_{ie} parameter against the collector load resistor cuts the gain to roughly *one half* that computed from



(A) Using RTL symbols.

(B) Discrete equivalent.

Fig. 4-7. Class-A amplifier with feedback.

Chart 4-1. "Instant Design" Chart for Class-A Amplifier With Feedback (Circuit of Fig. 4-7 Only)

Amplifier Gain of 2 (R1 = 6.2k)					
Lower —3 dB Freq (Hz)	10	100	1k	10k	100k
C1 (μF)	2.4	0.24	0.024	0.0024	0.00024
Upper —3 d3 Freq (Hz)	100	1k	10k	100k	1M
C3 (μF)	6.0	0.6	0.06	0.006	0.0006
Amplifier Gain of 5 (R1 = 2.4k)					
Lower —3 dB Freq (Hz)	10	100	1k	10k	100k
C1 (μF)	6.0	0.6	0.06	0.006	0.0006
Upper —3 dB Freq (Hz)	100	1k	10k	100k	1M
C3 (μF)	6.0	0.6	0.06	0.006	0.0006
Amplifier Gain of 10 (R1 = 1.3k)					
Lower —3 dB Freq (Hz)	10	100	1k	10k	100k
C1 (μF)	12.0	1.2	0.12	0.012	0.0012
Upper —3 dB Freq (Hz)	100	1k	10k	100k	1M
C3 (μF)	6.0	0.6	0.06	0.006	0.0006
Amplifier Gain of 20 (R1 = 640Ω)					
Lower —3 dB Freq (Hz)	10	100	1k	10k	100k
C1 (μF)	24	2.4	0.24	0.024	0.0024
Upper —3 dB Freq (Hz)	100	1k	10k	100k	1M
C3 (μF)	6.0	0.6	0.06	0.006	0.0006

the resistor ratios. For instance, the amplifier has a gain of 10 if R1 is 1.3k, and a gain of 20 if R1 is 640 ohms. For gains less than 10, the gain does not vary significantly as either transistor or temperature is changed. The input impedance of the circuit equals R1, and the output impedance is around 250 ohms for a 27k feedback resistor.

Once again, the frequency response is determined by the capacitors. The lower −3-dB point is determined by the frequency at which the reactance of C1 approximately equals R1. The upper −3-dB point is determined by the frequency at which the reactance of C3 equals 250 ohms or so. By choosing capacitors correctly, you can allow only the desired frequencies to pass through the amplifier, and reject all others which would only add noise to the system.

Chart 4-1 is an "instant design" chart that indicates the design values of R1 and C1 for any desired gain and frequency response. Resistor R1 can be a fixed resistor, the source resistance of the previous stage, or any desired combination of source and fixed resistance. Amplifier stages may be cascaded wherever necessary for more gain. The 640-ohm load resistor sets the maximum gain at 20 or so per stage.

Both types of class-A amplifiers have several things in their favor. They require only one-sixth of a hex inverter, one or two resistors, and two or possibly three capacitors. They operate from a single supply, and if you want, any remaining gates left over in the package may be used either digitally or linearly anywhere you wish. While a 27k feedback resistance gives the maximum possible output swing, lower values can often be used to both extend the frequency response and lower the output impedance of this circuit. With a 10k feedback resistor, the amplifier is useful to 10 MHz, and the output impedance is less than 100 ohms. This is particularly handy when the gates are used in a crystal oscillator that must directly drive a JK flip-flop divider chain or other *digital* load.

DIFFERENTIAL AMPLIFIERS

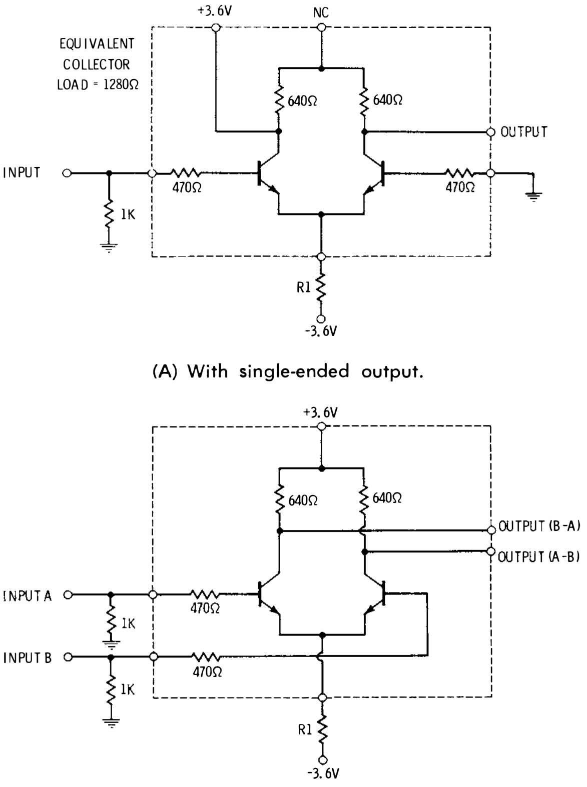
The class-A amplifiers have several limitations. They are ac-only amplifiers that must have capacitor coupling on both input and output. They always invert the signal, and their input impedance is relatively low. If overdriven, they clip or limit nonuniformly, or *asymmetrically.* To amplify dc, a slightly fancier configuration, called the *differential amplifier* is needed. Two "diff amp" circuits are shown in Fig. 4-8.

The differential amplifier requires a negative as well as a positive supply. You can build only *one* differential amplifier per IC package since you must remove the emitters of two transistors from ground. Normally, a dual two-input gate is used, and one transistor on each side is disabled by shorting its base and emitter together.

The differential-amplifier circuits make use of the excellent matching obtainable with transistors that are all on the same IC substrate. The circuit amplifies dc as well as ac. It has two inputs, an *inverting* and a *noninverting* input, and can have both in-phase and out-of-phase (push-pull) outputs, or a single-ended output. These inputs and outputs may be used in any desired combination, at any frequency from dc to several megahertz.

There are some other important advantages. The gain depends on the current through R1. If you like, you can vary this current manually or electronically by changing either the value of R1 or the value of the negative supply voltage. The circuit makes an excellent modulator or agc amplifier when operated in this manner. The input impedance is higher than for the class-A amplifiers, particularly for lower gains. The differential amplifier is largely insensitive to power-supply variations and other *common-mode* input signals. The output voltage clips linearly and symmetrically, and neither transistor can saturate in this circuit, guaranteeing rapid recovery times.

The current supplied by the emitter resistor is nearly constant (for high-quality operation, this resistor is replaced with a true current source). This nearly constant current divides equally between the transistors and, therefore, equally between the load resistors. Any *difference* between the two input voltages upsets this balance, funneling more current to one side and less to the other. Both outputs are automatically out of phase with each other, and the *difference* between the two inputs is amplified and appears as a larger difference in output signals.



(A) With single-ended output.

(B) Differential input and output.

Fig. 4-8. Differential-amplifier circuits.

The circuit can be looked at as an emitter follower driving a grounded-base output stage to get from one input to the *opposite* output, or as a common-emitter stage to get from one input to the output on the same side. The circuit gain is determined by the emitter current and the load resistance; the only effect of changing IC's is to change the input impedance of the amplifier.

Chart 4-2 is an "instant design" chart for differential amplifiers. It permits calculating the gain, input impedance, and peak-to-peak limiting level of either type of differential amplifier.

Chart 4-2. "Instant Design" Chart for Differential Amplifiers

Single-Ended Output (Fig. 4-8A)				
Supply Voltages = +3.6 and −3.6	**Output Impedance = 1.2k**			
R1 (Ohms)	15k	6.2k	3k	1.5k
Emitter Current (mA)	0.2	0.5	1.0	2.0
Voltage Gain (Times)	2.4	6	12	24
(dB)	8	16	22	28
Input Impedance (Ohms)	16k	6.6k	3.3k	1.6k
Peak-to-Peak Limiting Output (Volts)	0.24	0.6	1.2	2.4
Differential Output (Fig. 4-8B)				
Supply Voltages = +3.6 and −3.6	**Output Impedance = 640Ω**			
R1 (Ohms)	15k	6.2k	3k	1.5k
Emitter Current (mA)	0.2	0.5	1.0	2.0
Voltage Gain (Single-Ended) (Times)	1.2	3	6	12
(dB)	2	10	16	22
Input Impedance (Ohms)	16k	6.6k	3.3k	1.6k
Peak-to-Peak Limiting Output (Single-Ended) (Volts)	0.12	0.3	0.6	1.2

As an example of using this chart, suppose we have a 1.5k emitter resistor (R1) and a −3.6-volt supply. The emitter current will be 2 milliamperes, and the gain of a single-ended amplifier (Fig. 4-8A) will be 24, or around 28 dB. The input impedance will be 1.6k, and the limiting output voltage will be 2.4 volts. If we reduce the emitter current to 0.5 milliampere, we need a 6.2k emitter resistor. The gain drops to 6, or 16 dB, while the input impedance goes up to 6.6k, and the output limiting voltage drops to 0.6 volt.

Frequency response of either amplifier may be limited at the high end by a rolloff capacitor, just as in the class-A amplifiers. The upper −3-dB (half-power) frequency is reached when the capacitor reactance equals the output resistance of the differential amplifier.

RESTRICTIONS ON THE DIFFERENTIAL AMPLIFIER

The differential amplifier requires an entire IC package by itself and needs a negative supply. It is usual to ground one input and choose either the inverting or noninverting side for the particular application. Both sides of the amplifier must see identical low-impedance paths to ground, either through the source resistance or through a fixed resistor to ground. You may direct couple to the input so long as the dc input voltage is nearly zero and so long as there is a low impedance looking back into the source.

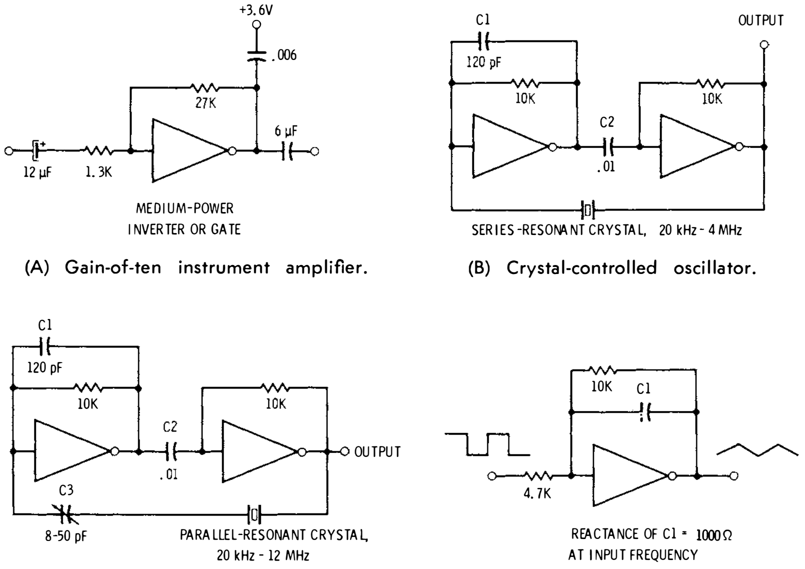
The output of a differential amplifier has a dc offset that varies from one-half to the full amount of positive supply voltage. If you need a dc amplifier in which a grounded input produces a grounded output, you have to add into the output an offset that equals the normal output bias level.

LINEAR APPLICATIONS FOR RTL

The RTL circuits form an impressive and economical lineup of useful amplifiers. Admittedly, better performance can be obtained with ordinary linear IC's, but these often cost more and require higher supply voltages. In general, RTL is used in the linear mode only when the linear operation is in conjunction with a digital system, or when extreme economy or operation from low supply voltages is important. Often in a digital instrument, one and possibly two power supplies may be saved by using RTL in linear operation instead of discrete or linear-IC equivalent circuits.

SOME TYPICAL RTL LINEAR CIRCUITS

Fig. 4-9A shows a gain-of-ten instrument amplifier with a bandwidth of 10 Hz to 100 kHz and an input impedance of 1.3k. This circuit could see use as an input amplifier in a frequency meter or elsewhere to boost an outside-world analog signal before it goes into a digital system.



(A) Gain-of-ten instrument amplifier.

(B) Crystal-controlled oscillator.

(C) Adjustable crystal oscillator.

(D) Square-triangle wave converter.

Fig. 4-9. Applications of class-A amplifier with feedback.

A crystal oscillator is also a natural application for this amplifier, and two versions appear in Figs. 4-9B and 4-9C. Crystal oscillators are often used in conjunction with digital instruments, perhaps as the beginning of a time base for a frequency counter, as a frequency reference for an electronic stop watch, or as the primary frequency reference in a musician's pitch standard. It is obviously advantageous to work a crystal oscillator into a remaining gate or two left over in a digital system, instead of having to go to discrete transistors and possibly a new supply voltage.

Fig. 4-9B is the simpler of the two circuits, and is useful from 20 kHz up to 4 MHz. It uses a *series*-resonant crystal. The crystal is overdriven, so the output is a high-harmonic square wave. Such a square wave is ideal for driving a counting chain consisting of JK flip-flops, or for producing harmonics for ham-receiver calibration or other communications uses. Capacitor C1 is included to make sure the crystal oscillates at its fundamental frequency. If C1 is too large, the oscillator may squeg, and if C1 is too small, oscillation may take place on the second or third overtone of the crystal. Normally a 120-picofarad capacitor will handle a 100-kHz crystal, and a 22-picrofarad unit will handle crystals in the 1-2 MHz region. Some surplus crystals and unusual-cut crystals may require adjustment of this capacitor and possibly tailoring of C2 for optimum results, but for most crystals, these values are not at all critical.

Sometimes it may be desired to adjust the output frequency slightly by "pulling" the crystal. For an accurate frequency counter, or for accurate communications-receiver calibration, it is necessary to be able to zero-beat the crystal output against a signal from WWV or some other standard. To do this, the adjustable crystal-standard circuit of Fig. 4-9C may be used. This circuit uses a *parallel*-resonant crystal, specified for operation into a 32-picofarad load. The trimmer capacitor becomes the load; if it is more than 32 picofarads, the frequency is low; if it is less than 32 picofarads, the crystal output frequency is high. Capacitor C3 permits adjustment through this range, perhaps obtaining a 200-Hz spread for a 100-kHz crystal.

Either crystal oscillator circuit can directly drive a JK flip-flop without needing a coupling capacitor. This usage will be covered further in later chapters. The circuit in Fig. 4-9C may be used with crystals cut for frequencies up to several megahertz.

A circuit for converting square waves to triangular waves is shown in Fig. 4-9D. Here, a capacitor for feedback and a bias resistor have been added to the gate, and the circuit produces a 200-millivolt triangular-wave output when driven by a digitally derived RTL square wave. The reactance of the capacitor must be 1000 ohms at the fundamental frequency of the input for proper operation. A larger capacitor reduces the available amplitude; a smaller one gives more output amplitude, but the waveform starts to become exponential instead of triangular. This circuit is often used in pulse modulation and in switching-mode regulated power supplies and switching-mode audio amplifiers. By proper gating, you can also produce a linear run-down ramp similar to that obtained from a Miller integrator.

DIFFERENTIAL-AMPLIFIER APPLICATIONS

Fig. 4-10A shows a 455-kHz amplifier with a gain of 30; the circuit uses a dual two-input gate and a split +3.6-volt, −3.6-volt power supply. This type of circuit is useful from audio frequencies to beyond 5 MHz, and may be used as a linear amplifier with low-level signals, as a clipping or limiting amplifier with high-level signals, as a gain-controllable amplifier, or as a modulator. Gain control and modulation are achieved by varying the emitter current. A phase meter or fm discriminator may also be built using a circuit of this type.

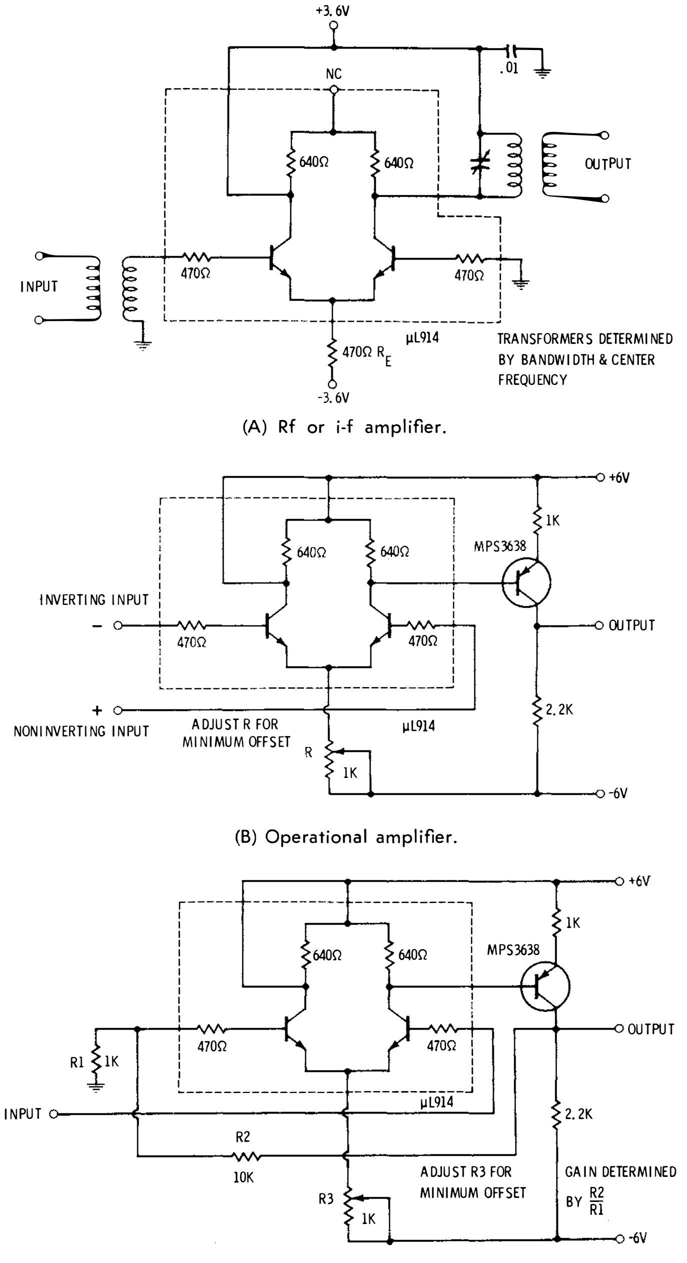
The amplifier shown in Fig. 4-10B may be adjusted to give zero volts dc out for zero volts dc in, with a choice of either inverting or noninverting inputs. By adding feedback from the output to the inverting input, the amplifier can be made quite stable, and the gain will be determined by the ratio of the input and feedback resistors. This arrangement is called an *operational amplifier.* A gain-of-10 noninverting amplifier is shown in Fig. 4-10C. While not nearly as sophisticated as a quality linear operational amplifier would be, this circuit can be built at low cost, and it offers stable, bipolar dc amplification. Resistor R3 is adjusted to bring the output voltage to zero with the input grounded. In this application, +6-volt and −6-volt supplies are used to allow a higher output voltage swing. The circuit is useful to several megahertz, and by adding capacitors to the feedback loop, it can also be used as a linear ramp or sweep generator. We might also make the circuit frequency selective by adding RC, RL, or RLC components to the feedback and input networks.

COMPARATORS

Often, there is need for a means to convert a nondigital input signal into a digital one suitable for being counted or otherwise entered into a digital system. In the last chapter, we saw how several gates or a cascaded hex inverter may be used to perform this function, or how a snap-action Schmitt trigger might also be used.

These previous circuits suffer from several faults. It takes several volts of peak-to-peak input signal to drive these simpler circuits. The input impedance is relatively low and can be nonlinear. If the input is a sine wave, an asymmetrical rectangular wave will result. Such a result may not matter if we are interested only in the presence or the frequency of the input, but if we are trying to measure the time between zero crossings, for example, this performance is clearly unsuitable.

A *comparator* is any circuit that compares two inputs. If input A is larger than input B, a logic-1 output results; if B is larger than A, a logic-0



(A) Rf or i-f amplifier.

(B) Operational amplifier.

(C) Bipolar dc amplifier.

Fig. 4-10. Practical applications of differential amplifier.

output results. Usually a comparator has considerable gain, and a difference of a few millivolts on the input rapidly snaps the output from a 1 to a 0, or vice versa.

A comparator made from a dual two-input gate is shown in Fig. 4-11. The gate serves as a differential amplifier that in turn drives an external differential amplifier, followed by an output emitter follower. The circuit is simply an amplifier with too much gain. If input A is positive with respect to input B, a few millivolts of difference snaps the output several volts in the positive direction, and vice versa.

There are two ways to use this circuit. We can ground one input, and the opposite input will serve as a zero-crossing detector. When a small sine wave is fed in, the output provides a 1 for positive input polarity and a 0 for negative input polarity. The signal loading is the same for both cycle halves, and a symmetrical square-wave output results; it is then possible to measure precisely the time between zero crossings, or otherwise provide a suitable digital output.

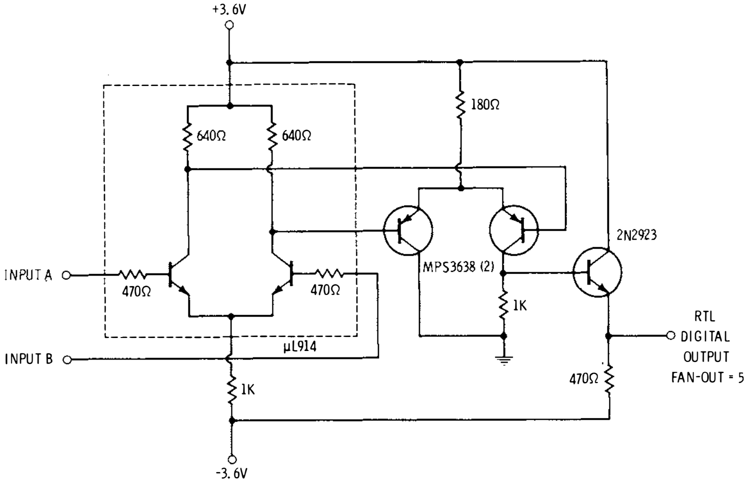


Fig. 4-11. Diagram of a comparator circuit.

Operation with one input grounded works fine for zero-crossing detection, but in the presence of noisy signals, performance can become erratic. Since noise is usually centered about zero, noise variations can often be counted along with the signal. To get around this problem, we can bias the other input enough to insure some noise immunity. For instance, suppose there is 100 millivolts of noise in the system. We set the B input to +0.5 volt dc. The noise by itself never crosses this threshold, but the signal or signal plus noise does. This provides a handy way of selecting the trigger point on a waveform to give the most reliable triggering. If we want to count negative-going signals, perhaps narrow spikes, we would use a negaoutput results. Usually a comparator has considerable gain, and a difference of a few millivolts on the input rapidly snaps the output from a 1 to a 0, or vice versa.

A comparator made from a dual two-input gate is shown in Fig. 4-11. The gate serves as a differential amplifier that in turn drives an external differential amplifier, followed by an output emitter follower. The circuit is simply an amplifier with too much gain. If input A is positive with respect to input B, a few millivolts of difference snaps the output several volts in the positive direction, and vice versa.

There are two ways to use this circuit. We can ground one input, and the opposite input will serve as a zero-crossing detector. When a small sine wave is fed in, the output provides a 1 for positive input polarity and a 0 for negative input polarity. The signal loading is the same for both cycle halves, and a symmetrical square-wave output results; it is then possible to measure precisely the time between zero crossings, or otherwise provide a suitable digital output.

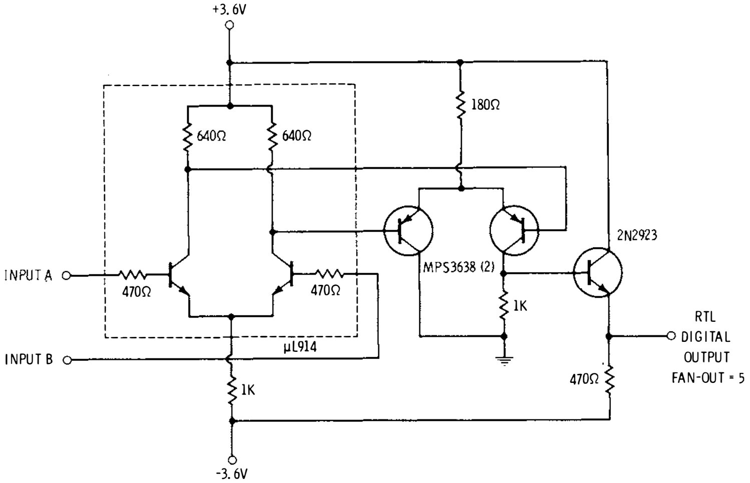
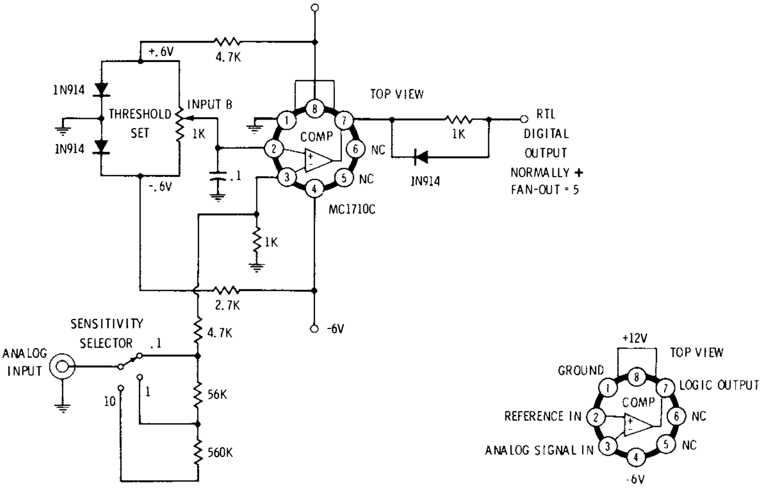


Fig. 4-11. Diagram of a comparator circuit.

Operation with one input grounded works fine for zero-crossing detection, but in the presence of noisy signals, performance can become erratic. Since noise is usually centered about zero, noise variations can often be counted along with the signal. To get around this problem, we can bias the other input enough to insure some noise immunity. For instance, suppose there is 100 millivolts of noise in the system. We set the B input to +0.5 volt dc. The noise by itself never crosses this threshold, but the signal or signal plus noise does. This provides a handy way of selecting the trigger point on a waveform to give the most reliable triggering. If we want to count negative-going signals, perhaps narrow spikes, we would use a negative bias on input B, again with the same results. Output symmetry is lost as the bias level increases.

Commercial comparator units are available. Fig. 4-12 shows the use of such a unit in a practical comparator circuit with variable threshold. The input selector switch adjusts the sensitivity of the device to input levels of 0.1, 1, or 10 volts. A continuous sensitivity control normally is not needed because the comparator will stand considerable variations in input signal level without any adverse affects. Usually, the threshold potentiometer is set near zero, particularly if we want zero-crossing information. With noisy input signals, the threshold is offset enough to get reliable operation. The offset is negative for negative-going input signals, and positive for either bipolar or positive-going inputs. If desired, some positive feedback may be added from the output to input B. This will give a snap action to the input and introduce hysteresis, and it can increase the noise immunity of the comparator operation. The 0.1-μF bypass capacitor should be disconnected from lead 2 if positive feedback is used.



(A) Schematic diagram.

(B) Basing of comparator.

Fig. 4-12. Input circuit for electronic counter.

DUTY-CYCLE INTEGRATORS

Often, we are content to use the output of a digital system in digital form, reading things on a numerical readout, printing answers on a sheet of paper, or perhaps just watching some light bulbs. For some applications, it is desirable to convert any digital duty-cycle variations into a linear form, either to deflect a meter pointer or to generate an audible output. Typical examples of this are tachometers, dwell meters, frequency meters, fm discriminators, and flutter-measurement systems.

A simple method of converting duty-cycle variations into either audio signals or a meter indication is called *duty-cycle integration.* Fig. 4-13 shows the basic concept. Suppose in Fig. 4-13 the switch is closed all the time. A current of 1.0 milliampere exists all the time, and the meter will deflect fully to 1.0 milliampere. On the other hand, if the switch is *open* all the time, the meter will read zero, because the 1.0-milliampere current is never made available to it.

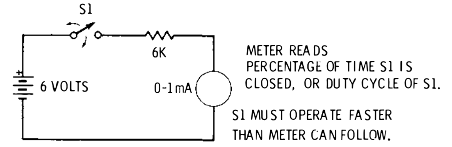

S1
6K
6 VOLTS
0-1mA
METER READS PERCENTAGE OF TIME S1 IS CLOSED, OR DUTY CYCLE OF S1.
S1 MUST OPERATE FASTER THAN METER CAN FOLLOW.

Fig. 4-13. Duty-cycle integrator.

Suppose that S1 is switched on and off in such a way that the switch is on for half the time and off for half the time, on the average. The meter will register 0.5 milliampere; the average current through the meter exactly equals one-half the total available current, because the current is present only half the time. We must open and close the switch rapidly enough that the pointer remains in one position without fluctuation, but above this minimum speed, the meter reading will be *independent* of the *frequency* with which the switch is opened and closed. Thus the meter reads only the percentage of the time the switch is closed, or the *duty cycle* of the switch operation.

We must hold the supply voltage constant while all this is going on, for increasing or decreasing the supply voltage will increase or decrease the meter reading the same way the duty cycle does. Deliberately varying the supply voltage results in a device called a *pulse-height*, *pulse-width multiplier*, in which the meter reading equals the *product* of the duty cycle multiplied by the supply voltage. However, usually we want the output to monitor duty cycle only. This usage requires a regulated power supply, or at the very least, a calibration of the circuit at the exact supply voltage being used.

Dwell Meter

The *dwell time* of a gasoline engine is defined as the length of time the ignition points stay closed; it is usually expressed in degrees of distributor rotation. For instance, in an eight-cylinder engine only 45° of distributor rotation is available for each of the eight cylinders. For any given cylinder, if the points were to stay closed all the time, the dwell angle would be 45°. The dwell angle is a critical adjustment during tune-up, with 13° to 20° being typical optimum dwell angles.

To build a dwell meter, we simply filter the dc at the points (Fig. 4-14) to obtain a spike-free rectangular wave that is positive when the points are open and grounded when the points are closed (assuming a negative-ground electrical system). This point signal is squared up with an inverter to produce a rectangular pulse with good rise and fall time. A second inverter drives a meter through a calibration potentiometer, and the dwell time may be read directly on the meter. A regulated supply, either in the form of a zener diode or internal mercury batteries, is needed to prevent vehicle voltage variations from affecting the meter reading.

To calibrate the meter, the input is grounded, and the meter is adjusted to full scale, or some convenient point (such as the "45" on a 0-50 microampere meter scale) to allow easy reading of the dwell angle. You can also add a new meter scale that goes linearly from 0 to 45°. For four- or six-cylinder engines, the procedure is the same, except that the dwell reading for a given duty cycle is now calibrated on a 0-60° or a 0-90° scale with the dwell angle remaining the same *percentage* it was before. The instrument and the calibration remain the same; only the meter face changes.

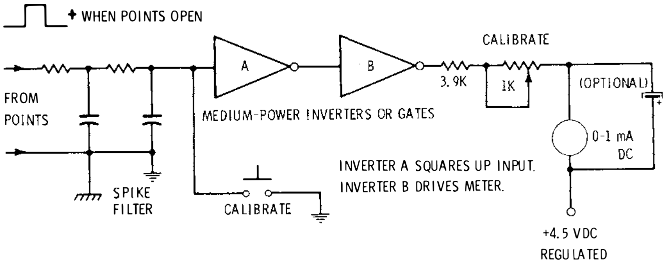


Fig. 4-14. Circuit of a dwell meter.

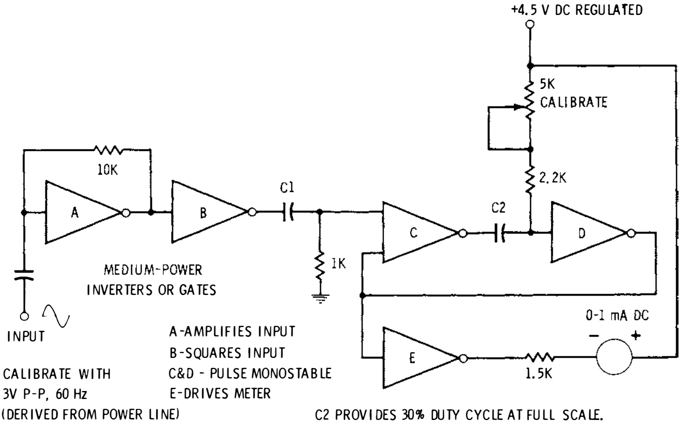
Note that the meter returns to +4.5 volts instead of to ground. When an inverter saturates, its output can come only within 0.2 volt or so of ground, but when an inverter shuts off, its output goes all the way to the supply voltage, without any remaining drop. If the meter were connected to ground, there would be a constant meter current due to the saturation drop that would have to be either allowed for or else bucked out with an external voltage.

The capacitor shown across the meter is optional. It can eliminate meter bobble at low operating speeds, but the response time to a sudden acceleration or change in input speed may be adversely affected.

Frequency Meter

Suppose we trigger a monostable that delivers output pulses of a constant width. The more often we trigger it, the higher the duty cycle will be, and integrating the duty cycle will indicate the input *frequency*. If the indication is to be accurate, the duty cycle of the monostable must be kept below 30° or so, the meter must be isolated from the monostable, and a regulated power supply must be used.

To build a frequency meter, we need only square up an input signal, capacitor couple it to obtain a trigger pulse once each cycle, and then trip a constant-width monostable (Fig. 4-15). The monostable output always stays positive for a fixed time, but how long it remains grounded after that time, and consequently the duty cycle of the output, is determined by the input *frequency.* This duty-cycle output may now be integrated to read out the input frequency directly on a meter. To guarantee linearity, components were chosen so that the monostable never goes above a 30-percent duty cycle; therefore, the meter current for a 100-percent duty cycle should be about three times the normal meter full-scale current.



A-AMPLIFIES INPUT
B-SQUARES INPUT
C&D - PULSE MONOSTABLE
E-DRIVES METER

CALIBRATE WITH
3V P-P, 60 Hz
(DERIVED FROM POWER LINE)

C2 PROVIDES 30% DUTY CYCLE AT FULL SCALE.
C1 PROVIDES PULSE LESS THAN 1/3 MONOSTABLE ON TIME.

Fig. 4-15. Circuit of a frequency meter.

A frequency meter of this type is useful from subaudio frequencies to several megahertz, and multiple ranges may be obtained by switching the monostable capacitor in decade steps. The 60-Hz power line can be used to make the instrument self-calibrating. The accuracy of a frequency indication is in the 2-percent to 5-percent range with selected components; for greater frequency accuracy, the counter techniques of Chapter 8 are used.

Tachometer

A tachometer (Fig. 4-16) is essentially identical to a frequency meter, except for scale calibration. In the case of a gasoline engine, you must consider the number of times the points close during each revolution. On an eight-cylinder, four-stroke-cycle gasoline engine, each cylinder is fired once every two revolutions. Thus the points close four times each revolution. At 600 rpm, for example, four point closures per revolution would give a 40-Hz output signal.

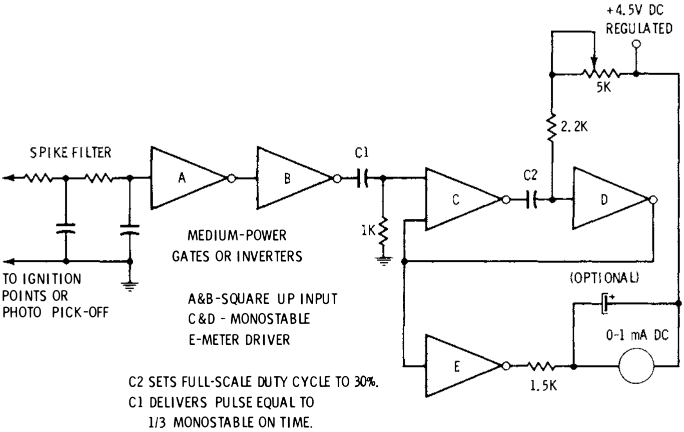


Fig. 4-16. Circuit of a tachometer.

Once again, it is essential to keep the duty cycle below 30 percent, the supply regulated, and the meter isolated from the monostable. Also, it is possible to calibrate with the ac power line.

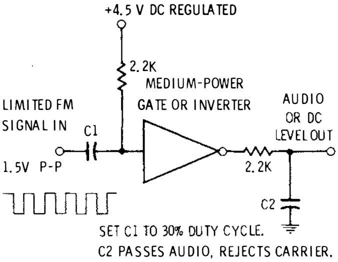
Hex inverters lend themselves easily to use in frequency meters or tachometers. On a tachometer, two inverters are used to condition the input, three for the monostable, and one to drive the meter. The stability, linearity, and temperature performance of this circuit are considerably better than in the normal one- or two-transistor tachometer circuits commonly used, and the overall circuit cost is usually less.

FM Discriminator

Integrator-type circuits can be used to replace the normal Foster-Seeley or ratio-detector frequency discriminators used for frequency demodulation. This substitution provides some unique benefits either for hi-fi listening or for simple and effective approaches to such things as wow and flutter meters, phase meters, and other instruments for listening to or measuring the effects of frequency modulation or frequency shift.

There are several techniques. In the case of an fm receiver (Fig. 4-17), we can take the limited i-f signal, use it to trip a monostable that has a

Fig. 4-17. Wide-band fm discriminator.



fixed pulse width, and then duty-cycle integrate the output with a capacitor of such value that there is no carrier ripple, but that all audio variations are passed on freely. The output signal faithfully follows the audio variations caused by frequency modulation in the input.

There is a major advantage to this technique. The bandwidth of this type of discriminator extends from dc to twice the i-f frequency, meaning the linearity over the normal deviation range is far better than can be obtained with any LC "S-curve" tuning scheme. The resultant distortion is thus far less. Further, no tuning at all is required, and the circuit is permanently aligned. On the debit side are a very low recovered audio signal voltage, the need for good limiting before detection, and the need for excellent supply-voltage filtering.

There are several other fm-discrimination techniques, all of which are variations on the duty-cycle integration theme. A delay line, each side of which drives an RS flip-flop, may be used instead of the monostable. The flip-flop duty cycle may then be integrated to appear as output audio. Another technique uses both conventional and logic techniques simultaneously. A quadrature tank is driven so that it oscillates at the average carrier frequency and then drives an OR gate 90° out of phase with the input signal. The duty-cycle variations of the OR gate are integrated to form audio.

With RTL, it is relatively easy to frequency-discriminate any audio test signal, carrier-communication-system signal, or any other low- or relatively low-frequency signal. Thus the need for bulky high-Q inductors and complex tuning and calibration schemes is eliminated. While RTL also may be used at 4.5 and 10.7 MHz for TV and hi-fi detection, considerably better performance may be obtained at these high frequencies by using either premium IC lines or linear circuits designed specifically for this task.

fixed pulse width, and then duty-cycle integrate the output with a capacitor of such value that there is no carrier ripple, but that all audio variations are passed on freely. The output signal faithfully follows the audio variations caused by frequency modulation in the input.

There is a major advantage to this technique. The bandwidth of this type of discriminator extends from dc to twice the i-f frequency, meaning the linearity over the normal deviation range is far better than can be obtained with any LC "S-curve" tuning scheme. The resultant distortion is thus far less. Further, no tuning at all is required, and the circuit is permanently aligned. On the debit side are a very low recovered audio signal voltage, the need for good limiting before detection, and the need for excellent supply-voltage filtering.

There are several other fm-discrimination techniques, all of which are variations on the duty-cycle integration theme. A delay line, each side of which drives an RS flip-flop, may be used instead of the monostable. The flip-flop duty cycle may then be integrated to appear as output audio. Another technique uses both conventional and logic techniques simultaneously. A quadrature tank is driven so that it oscillates at the average carrier frequency and then drives an OR gate 90° out of phase with the input signal. The duty-cycle variations of the OR gate are integrated to form audio.

With RTL, it is relatively easy to frequency-discriminate any audio test signal, carrier-communication-system signal, or any other low- or relatively low-frequency signal. Thus the need for bulky high-Q inductors and complex tuning and calibration schemes is eliminated. While RTL also may be used at 4.5 and 10.7 MHz for TV and hi-fi detection, considerably better performance may be obtained at these high frequencies by using either premium IC lines or linear circuits designed specifically for this task.

CHAPTER 5

Counting Flip-Flops

The inventions of the JK flip-flop and the Type-D counting flip-flops were important steps forward in computer technology. When these same devices were put in integrated-circuit packages and offered as low-cost distributor stock items, the modern age of digital IC's really began. The whole reason for digital IC's and for RTL is centered on these two devices which so changed the approaches to counting, scaling, and computing that a whole new age of digital instruments and applications became possible.

These counting flip-flops are called *clocked logic elements* in that they can be told ahead of time what to do, and then later on, upon command, they will automatically carry out what they have been previously instructed to do. It is possible to divide down, or scale, any number or any frequency with a counting chain made up of counting flip-flops and few or no discrete components. The division is totally independent of frequency and requires no adjustment, calibration, or component selection. There is no instability or "count jumping" such as may be experienced with relaxation-divider and similar circuits.

With counting flip-flops, division by 2 is automatic, without a need for external steering or pulse-shaping networks. You may count at any desired speed, stopping any time you wish for as long as you wish. With counting flip-flops, you can automatically pass on logic signals from stage to stage in a sequential manner on command, and not worry about transferring the logic signals more than one stage at a time.

You can also build more exotic circuits simply—circuits that get outside-world signals in step with the digital world, numerous decimal counter circuits, circuits to generate a precise one-and-only-one 1-second gate on random command, and many more.

Using the Direct Inputs

The direct inputs immediately flip the JK to one side or the other upon command. They have exactly the same limitations and disallowed-state conditions that we encountered with the RS flip-flop of Chapter 3. Most of the JK flip-flops lack a preset input, so two flip-flops fit in a fourteen-lead package. There is sometimes a way to pick up one or more external preset inputs, as will be described shortly.

If both preclear and preset inputs are available, they obey the following logic rules:

1. If both the preclear and preset inputs remain grounded, the flip-flop will not change state unless acted on by the clocked inputs. In fact, this is the *only* permissible condition of the direct inputs when the clocked inputs are being used. To disable the preclear and preset inputs, you simply ground them.
2. If the preclear input is made positive with the preset input grounded, the flip-flop goes into the state with the $\overline{Q}$ output positive and the Q output grounded. This is just as if the positive voltage on the preclear input were "passed on" to the $\overline{Q}$ output, across the bottom of the JK symbol.
3. If the preset input is made positive with the preclear input grounded, the flip-flop goes into the state with the Q output positive and the $\overline{Q}$ output grounded. This is just as if the positive voltage on the preset input were "passed on" to the Q output, across the top of the JK symbol.
4. If both the preclear input and the preset input are made positive, the flip-flop will go into a disallowed state in which *both* the Q and $\overline{Q}$ outputs are grounded. The *last* direct input to go to ground determines the state the flip-flop will assume. This input condition is normally avoided, as the outputs cease to be complementary under this condition.

To disable any effects of the direct inputs, simply ground them both. A positive voltage on either direct input passes on to the nearest output.

These direct inputs have absolute dominance over the clocked inputs, and thus should never be used simultaneously with the clocked inputs. On a JK flip-flop, the condition of the clocked inputs has no effect on whether the direct inputs will or will not operate.

The foregoing rules can be converted into a negative-logic chart such as the one in Fig. 5-2.

Normally, only a brief pulse is required to preset or preclear a JK flip-flop. A duration of 200 nanoseconds is often sufficient. For slower systems, a 1-to-5 microsecond preset or preclear pulse is commonly used. Usually, you need not worry about multiple preset and preclear input pulses. A bounceless push button normally is not necessary for generating preclear

PRESET	PRECLEAR	Q	$\overline{Q}$	WILL CLOCKED INPUTS OPERATE?
1	1	Q_n	$\overline{Q}_n$	YES
0	1	0	1	NO
1	0	1	0	NO
0	0	1 (DISALLOWED)	1	NO

Fig. 5-2. Negative-logic chart for direct inputs of JK flip-flop.

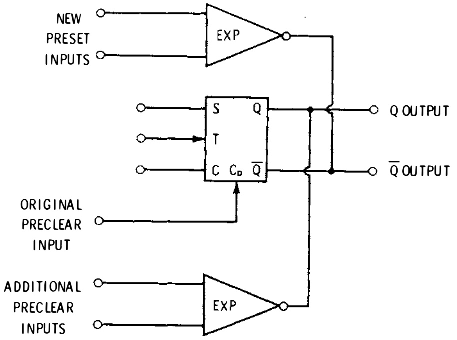
pulses, since preclearing a counter many times achieves the same results as preclearing it once. We must, however, make sure that the preclearing or presetting is complete before the arrival of a new clock or toggle pulse at the clocked inputs.

Adding a Preset Input

A gate expander may be added to many of the JK flip-flops to provide the sometimes absent preset input, or to add fan-in capability to the preclear. This technique works *only* when internal output isolation is not provided in the JK flip-flop. When the technique can be used, you simply attach the gate expander to the opposite output (Fig. 5-3). Thus, for added preset inputs, connect the expander *output* to the $\overline{Q}$ output; for preclear inputs, connect the expander output to the Q output. The expander inputs then become the new preset and preclear inputs.

If you are uncertain whether this technique will work with the particular JK in use, refer to the schematic of that JK. If the actual cross-coupled flip-flop in the schematic goes to the output terminals, the technique works. If there is a single inverter buffer separating the flip-flop from the output, the technique does not work.

There is another simple solution if you need a preset input but do not need a preclear input. A JK flip-flop is a symmetric sort of thing. To get a "free" preset input, simply change the names of everything. The set becomes the clear, the preset is the former preclear, and the Q and $\overline{Q}$ outputs



TECHNIQUE WORKS WITH	TECHNIQUE DOES NOT WORK WITH
μL923	MC722
MC723	MC726
MC790	MC791
MC776	
MC779	
MC787	

Fig. 5-3. Method of adding preset and preclear inputs.

are interchanged. Only the toggle, +, and ground connections do not change. The rules of operation stay the same.

For normal operation, all preset and preclear inputs remain grounded, and they are not used during the normally cyclic operation of a group of JK flip-flops. Usually, they find use only at the beginning or end of a count sequence to force the flip-flops into a desired state.

Using the Clocked Inputs

The clocked inputs are used for "normal" JK operation. As mentioned already, there are three of them, a set, a toggle, and a clear input. Used together, these three inputs can force the JK flip-flop to do nothing, to go into one of two prescribed states, or to go automatically into the other state each time.

Regardless of what is done to the set and clear inputs, nothing will happen to the state of the flip-flop except at the instant the toggle input *abruptly drops* from + to ground (that is, abruptly goes from a negative-logic 0 to a negative-logic 1). At that instant the conditions on the set and clear inputs determine what state the flip-flop will be forced into. We say that the flip-flop is *edge sensitive* to the negative-going clock transition.

Thus, the setting up of what the flip-flop is to do is independent of when the clock or toggle signal actually causes the circuit to do it. The setting-up process is called *conditioning*, and the toggling process is called *clocking*. We condition the flip-flops and then clock them. Nothing can happen to the outputs except during clocking (of course, so long as the direct inputs remain grounded).

The rules are as follows:

1. If the set input is made positive and the clear input is made positive, nothing will happen to the flip-flop as the toggle input suddenly goes from + to ground. The outputs stay the way they were.
2. If the set input is made positive and the clear input is grounded, the flip-flop will go into the state with the Q output positive and the $\overline{Q}$ output grounded when the toggle input suddenly goes from + to ground. In effect, the positive set voltage is passed on to the Q output, directly across the top of the flip-flop symbol.
3. If the set input is grounded and the clear input is made positive, the flip-flop will go into the state with the $\overline{Q}$ output positive and the Q output grounded when the toggle input suddenly goes from + to ground. In effect, the positive clear voltage is passed on to the $\overline{Q}$ output, directly across the bottom of the flip-flop symbol.
4. If both the set and clear inputs are grounded, the flip-flop will automatically *change* output states every time the toggle input abruptly goes from + to ground.

These rules are summarized in the negative-logic chart of Fig. 5-4. The chart shows what happens every time the toggle input goes from + to ground. Note that there are no disallowed states. The 1,1 condition (both set and clear grounded) gives an automatic change of state for each negative clock transition, whereas the 0,0 condition (both set and clear positive) inhibits the JK flip-flop and prevents any change from taking place in the outputs.

Note also that nothing happens at the instant the set or clear inputs are changed, nor does anything happen with the toggle input positive, grounded, or abruptly changing from ground to positive. It is only the negative-going toggle transition that has any effect. Of course, to get a negative toggle transition, somehow the toggle input first must be brought positive. Typical values range from 1.4 to 3.6 volts.

SET	CLEAR	Q	$\overline{Q}$
0	0	Q_n	$\overline{Q}_n$
1	0	1	0
0	1	0	1
1	1	$\overline{Q}_n$	Q_n

Fig. 5-4. Negative-logic chart for clocked inputs of JK flip-flop.

We need not be concerned with the inner workings of a JK flip-flop, since we have no control over what goes on inside it. As a matter of interest, however, a typical internal schematic is shown in Fig. 1-10. There are three essential parts to the operation of the JK. One part is a conventional reset-set flip-flop. Another part is an internal steering network. The third part consists of two transistors designed to act as capacitors when they operate in a *stored-charge* mode. In practice, the steering network responds to both the state the flip-flop is in and the conditions on the set and clear inputs. The steering circuit allows current to pass through only one of the two charge-storage transistors. When the toggle abruptly drops to ground, the stored charge is delivered to the proper side of the flip-flop to bring about the desired transition. The edge-sensitive feature, the automatic steering, and the automatic time delay are all caused by this precise delivery of a critical amount of stored charge.

Toggle-Input Restrictions

There are several important things to watch for if the JK flip-flop is to work properly. The supply voltage must be within one-half volt of +3.6 volts for reliable operation, and the supply must be well bypassed. Neither the fan-out nor the fan-in may be exceeded on any terminal. That is, there must be sufficient fan-out of the devices driving all JK inputs, and the JK outputs must have enough drive capability to handle all loads presented to them.

Most important, the toggle input must drop only once per count, and it must do so very abruptly and noiselessly. The toggle input must fall in less than 100 nanoseconds and more than 10 nanoseconds for proper flip-flop

Table 5-1. Toggle Sources for JK Flip-Flops

GOOD (Proper Operation)	BAD (Erratic or no operation)
Previous JK flip-flop	60-hertz power line
Reset-set flip-flop (Fig. 3-6)	Unconditioned mechanical contact
Bounceless push button (Figs. 3-2B and 3-3C)	Ordinary push button
Sine waves from 100 kHz to 5 MHz	Sine waves above 5 MHz or below 100 kHz
Buffered astable (Fig. 3-21)	Noisy, unconditioned outside-world signals
Schmitt trigger (Fig. 3-27C)	Audio
Hex-inverter squaring circuit (Fig. 3-27E)	Negative-polarity signals
Flip-flop squaring circuit (Fig. 3-28)	Signals that do not exceed 1.4 volts positive or do not drop below 0.4 volt positive
Comparators (Figs. 4-11 and 4-12)	Capacitor-coupled input signals

operation. If this is not the case, either the flip-flop will not operate at all, or it will be very erratic and possibly have a "preferred" state into which it will always fall. Table 5-1 lists some typical suitable and unsuitable trigger sources for JK flip-flops.

Frequency Limitations

Most of the JK flip-flops are specified to operate up to 4.0 MHz, and typically you can get useful performance to 10 MHz. Lower-power JK flip-flops are somewhat more restricted, with a 3-MHz specified toggle rate, and typical operation beyond 4 MHz. Higher-frequency operation is usually obtained at lower supply voltages and with tight supply-voltage regulation. The over-all shape and symmetry of the toggle waveform also come into effect at very high speeds.

The JK flip-flop is a static type of device that remembers what state it is in as long as supply power is applied. Thus there is no low-frequency limit to operation. We can easily toggle once per day, or even once per year if we like. We must be sure that the once-per-year negative-going toggle transition lasts only 100 nanoseconds or less and that no noise appears between toggle pulses, but this is true for any frequency of operation.

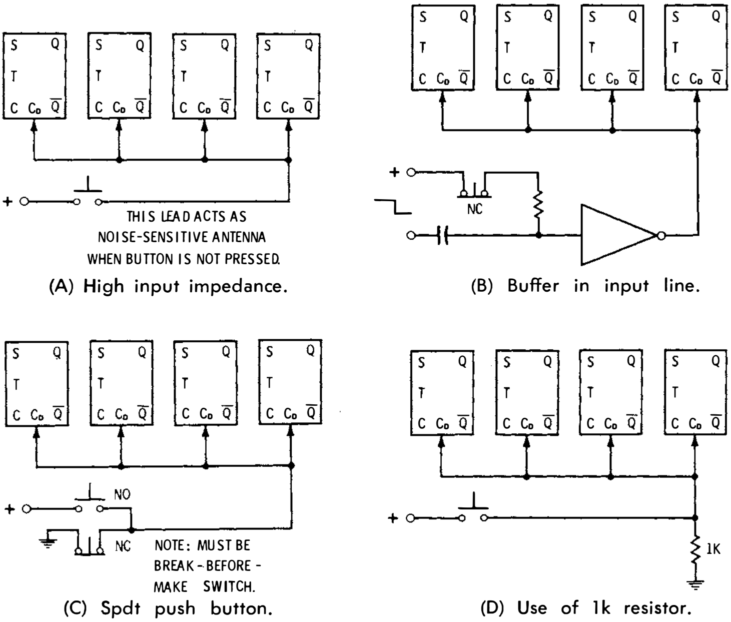
Other JK Restrictions

There are a few other limitations to the RTL JK flip-flop, particularly in high-speed circuits, that must be remembered. Direct set and direct clear pulses should last at least 200 nanoseconds, since any shorter time can cause a miss in resetting an immediately preceding toggle transition.

If you simply connect all the preclear inputs together through a normally open push button to the positive supply voltage (Fig. 5-5A), when the button is *not* pressed everything is tied together with a long lead that behaves as an open-circuit "antenna connection." Any noise present can raise havoc with some or all of the flip-flops in the chain, causing highly erratic operation. The obvious solution to the problem is to be sure a low impedance is *always* presented to the preset and preclear inputs. The low-impedance path may be to ground when the inputs are not being used, or to the positive supply when they are. In Fig. 5-5B, a buffer takes care of the problem. In Fig. 5-5C, an spdt push button is used (it must be a break-before-make switch). In Fig. 5-5D, we simply add a common resistor to ground to provide a low impedance all the time. Unused direct inputs should be grounded.

There is no need to make a preset or preclear input bounceless or noise free, for resetting a counter chain to 00000 once is just as good as doing it a hundred times in a row. However, the final resetting operation must be complete before the next toggling sequence begins. Also, as will be shown later, we have to be careful that presetting or resetting does not bring about a negative toggle transition to the next stage, unless everything is being reset at once and for a relatively long time.

As you cascade many JK flip-flops, you pick up around 100 nanoseconds of delay per stage, and this sometimes can result in the arrival of toggle



(A) High input impedance.

(B) Buffer in input line.

(C) Spdt push button.

(D) Use of 1k resistor.

Fig. 5-5. Methods of providing low input impedance.

signals before the conditioning signals arrive. While this delay must be watched in complicated, high-speed systems, it usually presents no problem in simple, everyday RTL applications. The solution to the problem is to delay the offending signals with a few inverters in cascade or with a monostable of some sort.

The Super JK

JK flip-flops come in several forms, ranging from a low-power version with a limited fan-out, through medium-power versions with fan-out of 10 or 16. Sometimes this is not enough drive capability. By limiting ourselves to one JK per package and providing some extra supply power, we can build a single JK flip-flop that has all five inputs and two buffered, short-circuit-proof outputs with fan-out of 80. This is done with a multipurpose IC consisting of one regular JK flip-flop, two buffers, and a one input expander. The circuit is shown in Fig. 5-6 and is called a *super JK.* This flip-flop has enough fan-out and versatility to handle almost any IC situation. The interconnections expand the ordinary flip-flop to include a preset input and to add one buffer to each output.

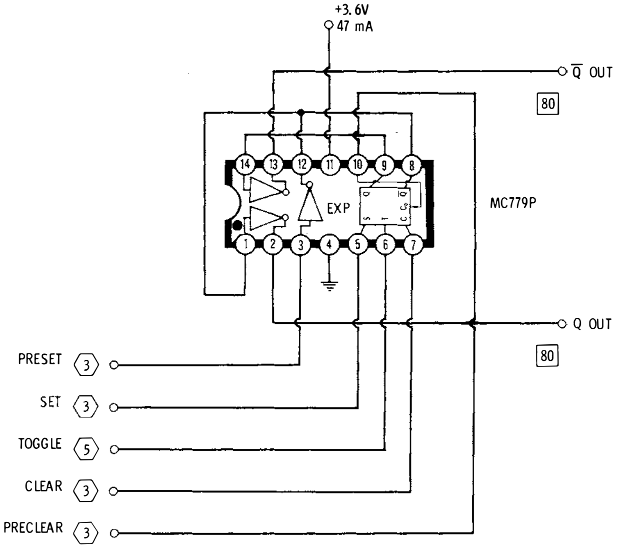


Fig. 5-6. Connections of a "super JK" flip-flop.

USING JK FLIP-FLOPS

There are about eight major basic circuits that can be assembled from one or two JK flip-flops, and there are an unlimited number of ways multiple JK's may be connected together. The eight basic circuit configurations will now be examined in detail.

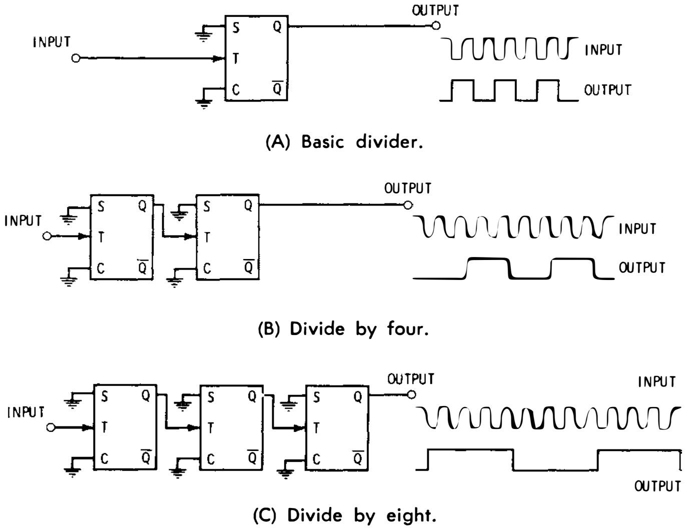
Binary Counter, or Divider

When both the S and C inputs are grounded (Fig. 5-7A), the JK flip-flop toggles once each negative-going clock transition. The Q output first flips "up," and then on the next count it flips "down" again. There are half as many negative transitions on the output as there were on the input, and if a square wave is fed in, the output is a square wave of half the input frequency. The circuit is called a *binary divider* or a *binary counter.* We can always start the binary division process correctly by using the direct inputs to set up the initial state; from there on, the output states alternate. The output square wave depends only on the input negative clock transitions and not on the symmetry of the input, so the output square wave is always symmetric and independent of the input symmetry.

We can cascade as many binary dividers as we like, as the divide-by-four and divide-by-eight circuits of Figs. 5-7B and 5-7C illustrate. Later, it will be shown that we can force a binary divider chain to count by any number we like that is equal to or less than its "untampered" capability. Four "untampered" binary counters cascaded count by 16. If we like, we can force them to divide by ten instead. Or, if we wish, we can even force them to divide on command by any number from 1 through 16; the division is reliable and totally independent of frequency or input counting rate.

Shift Register

Suppose a 1 is placed on one clocked input and a 0 is placed on the other (Fig. 5-8A). If the JK is clocked with a negative-going toggle transition,



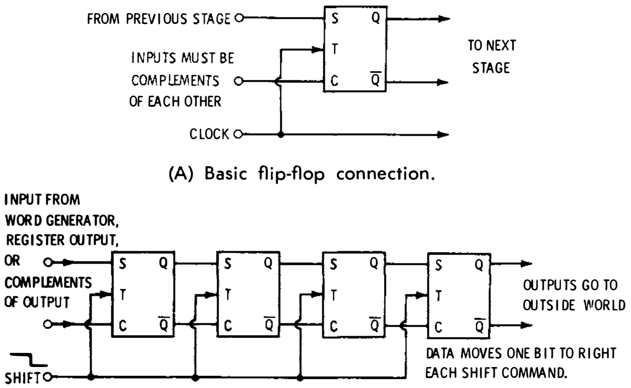
(A) Basic divider.

(B) Divide by four.

(C) Divide by eight.

Fig. 5-7. Binary dividers, or counters.

the same 1 and 0 now appear at the output, both signals apparently "passing through" the flip-flop, from input to output. A cascaded chain of flip-flops operated in this way is called a *shift register.* In a shift register, all of the toggle inputs are connected together and driven *synchronously,* while the output of one flip-flop *conditions* the next one down the line. In a shift register, the original input is required to be a 1 and a 0 or a 0 and a 1. The 1, 1 and 0, 0 input conditions are disallowed under normal circumstances.



FROM PREVIOUS STAGE
INPUTS MUST BE COMPLEMENTS OF EACH OTHER
CLOCK
S
T
C
Q
$\overline{Q}$
TO NEXT STAGE
INPUT FROM WORD GENERATOR, REGISTER OUTPUT, OR COMPLEMENTS OF OUTPUT
SHIFT
OUTPUTS GO TO OUTSIDE WORLD
DATA MOVES ONE BIT TO RIGHT EACH SHIFT COMMAND.

(A) Basic flip-flop connection.

(B) Four-bit shift register.

Fig. 5-8. Shift register.

Every time the shift register is toggled, the information is passed one *bit* to the right. The output may be routed to the outside world. If this is done, the "word" in the shift register appears at the output in *serial* form; that is, each successive bit takes its own turn appearing at the output. We can also connect the shift-register output back to its own input once we have the desired "word" in the register. A shift register connected this way is called a *recirculating register.* If the register is toggled a number of times exactly equal to the number of JK flip-flops present, the "word" in the register will be "marched out" bit by bit in serial form and returned to its initial position to await further use. And, as will be described shortly, we can connect the complements of the shift-register output back to the input, forming a *complementing register,* or a *walking ring* counter.

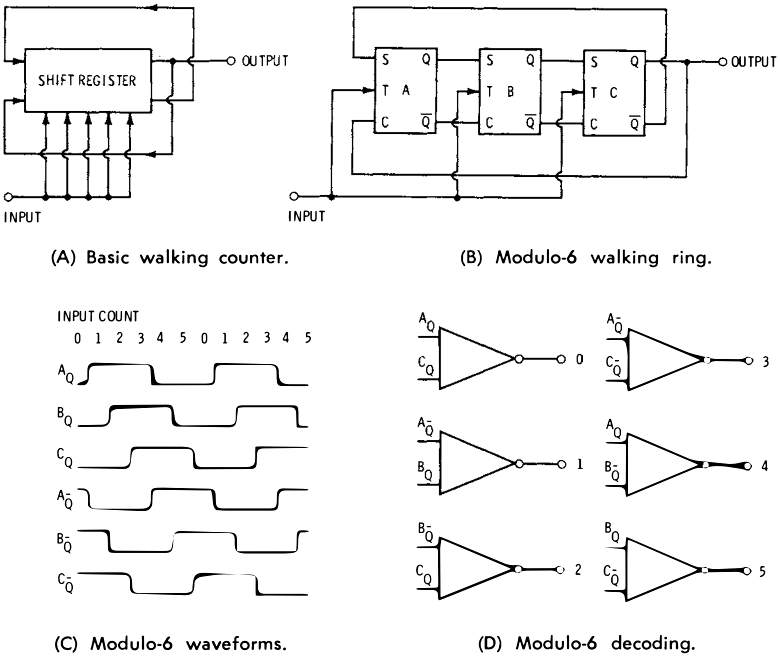
There are many important uses for shift registers. They make excellent frequency dividers, and they can be made into a relatively simple form of up-down, or add-subtract counter. They can perform all sorts of computation, including relatively exotic operations such as squaring, taking square roots, dividing, inverting, integrating, generating sine waves and other functions in pulse-rate form, and many others. They form key parts of a digital replacement for analog computers, called a *digital differential analyzer,* or DDA for short. They can be used to convert *serial* (one at a time) data into *parallel* (all at once) data, and vice versa. Shift registers can be used to generate codes, and they can be used in digital-to-analog and analog-to-digital conversion in more complicated systems.

Walking Ring Counter

The circuit in Fig. 5-9 goes by several different names: walking ring counter, Johnson counter, switchtail ring, complementary ring counter, Grey Code ring, and a half-dozen equally mysterious names. A walking ring counter is a shift register that has its outputs *cross coupled* to its inputs, forming a complementing shift register. The Q output of the last stage goes to the clear input of the first stage. The $\bar{Q}$ output of the last stage goes to the set input of the first stage. All other stage interconnections are made as in an ordinary shift register.

This circuit is basically a frequency divider, with several unique features—most of them good, except for one bad feature that is easily fixed once it is understood. The advantages will be considered first.

The frequency-division ratio equals *twice* the number of shift-register or JK elements used; for example, five flip-flops divide by ten. The output is always a symmetrical square wave, and there is only one stage of delay between the input negative clock transition and the output transition, regardless of the number of stages in use. This is called a *synchronous*



(A) Basic walking counter.

(B) Modulo-6 walking ring.

(C) Modulo-6 waveforms.

(D) Modulo-6 decoding.

Fig. 5-9. Walking ring counter.

counter. There is only one flip-flop that changes state at any given clock negative transition; the rest remain idle. Thus the circuit has definite advantages at very high operating frequencies.

A walking ring also can be used as a mechanical stepping switch is used, to produce outputs or light lamps sequentially. To do this requires *decoding.* Most counter forms require decoding. The binary-to-decimal decoder of Fig. 2-15 was an example of decoding; note that it required ten gates, and that some of the gates needed as many as four inputs. A walking ring counter needs only one two-input gate to decode any single state, regardless of how complex the counter is or how long the sequence is. This fact can result in a very significant saving in cost and complexity for some circuits.

The walking ring counter also can provide "phase shifts"; when the counter is fed at the suitable multiple of a desired frequency, it is possible to get four outputs shifted in phase by 90°, six outputs shifted in phase by 60°, and so on. As will be seen later, it is possible to "trick" the walking ring counter into handling odd as well as even counts simply by changing one connection.

So much for the advantages. We will now take a closer look at a typical counter to see just how all this comes about and why. Then we will turn to the subtle personality problem this counter has and see how it can be solved.

In Fig. 5-9B, a walking ring counter has been formed by building a three-flip-flop shift register and cross coupling the outputs. Suppose this counter is precleared so that the three Q outputs read 111. If the counter is now toggled, the first 1 is passed on to the second stage, and the second 1 is passed on to the third stage. Since the third stage is cross coupled to the first stage, the 0 that was on the complementary, or $\bar{Q}$, output is passed on to the set input of the first stage, and the first-stage Q output becomes 0. One toggling has caused the counter to shift to the 011 state. A second shift causes it to go to 001, and a third causes it to go to 000. On the next count, the last 0 is complemented into a 1, giving 100, followed by 110, and then back to 111. The sequence then repeats. The counts may be listed in sequential order as follows:

111	0
011	1
001	2
000	3
100	4
110	5
111	6 (Back to 0)

The 111 count can be called either 0 or 6; either way, the counter has six states before it repeats, or twice the number of flip-flops used. Further, as the waveforms of Fig. 5-9C show, only one flip-flop changes state on each negative toggle transition, and each output is up for half the time and down for half the time. There are six output symmetrical square waves of one-sixth the input frequency; the phase of each output is shifted by 60°, or offset by one input count, from its neighbor. Each output also has only one negative transition per six input counts.

If we are content with just a divide-by-six circuit, all we need do is take an output lead from any flip-flop. If the counter is to be reset, we probably should pick the last flip-flop in the chain for the output; otherwise, the first time around, the count will be short, and the output will be shifted with respect to our 111 phase of the counter.

If, instead, we want six sequential outputs, we must decode each count state. On count 0 (111), there are two outside ones. This condition happens nowhere else, so to decode count number 0 we use a two-input NAND gate; one input is connected to the Q output of the *first* flip-flop, and the other input is connected to the Q output of the *last* flip-flop. Both of these points go to ground on count 0, and the only time the output of the gate goes high is on count 0.

The next state (011) has a 0 on the first stage and a 1 on the second. This combination is unique, and state number 2 may be decoded with a two-input NAND gate connected to the $\overline{Q}$ output of the first stage and the Q output of the second stage. The output of the gate goes positive only on the 011 coincidence. Since it is not possible to NAND zeros, it is necessary to decode the corresponding 1's; the Q output is chosen to decode a 1, and the $\overline{Q}$ output is chosen to decode the 0 of the desired flip-flop.

All six decoding combinations are shown in Fig. 4-9D. We have the equivalent of a six-point electromechanical stepper that automatically stops on the next step each input count. With preset inputs, the counter may be reset to 111 at any desired time to start over.

This particular modulo-six walking ring counter is much like a die in its action. Two of these circuits and some simple decoding can be used to make an absolutely honest set of electronic dice, useful as a parlor toy or for serious probability studies.

General Decoding—Regardless of how long the walking ring is, you can always decode with two-input NAND gates, and you need only one two-input NAND gate for each point or count to be decoded. The loading on the JK flip-flops will be uniform if each state is decoded; there will be *two* gate inputs connected to each Q and $\overline{Q}$ output of each flip-flop. The technique works for any number of flip-flops. For a 24-point stepper, 12 JK flip-flops and 24 dual two-input gates would be used, for example.

To determine what count goes with which decoding, it is only necessary to write out all the possible states in sequential order. One of these states will be all ones (probably the state to which you will reset). Decode this state with the Q outputs of the first and last flip-flops. One state will be all zeros; it will be located "halfway around" from the all-one state. Decode this state with the $\overline{Q}$ outputs of the first and last flip-flops. Now, look at the remaining states. Each state will have a unique position in which there is a 0 immediately following a 1, or vice versa; this is the combination to be decoded. For instance, on a modulo-12 walking ring counter, the third transition from 111111 will be 000111. To decode this third step, apply the $\overline{Q}$ output of the third flip-flop and the Q output of the fourth flip-flop to a NAND gate; the output of this gate swings positive only on state 000111. All other gates are connected in a similar manner, with each point of connection determined by a unique 01 or 10 somewhere in the count sequence. Remember, to decode a 1 use the Q output, and to decode a 0 use the $\overline{Q}$ output. *Always* use negative logic when designing walking ring counters. Usually resetting the counter puts it in the 111 . . . 11 state, and the count sequence begins from there.

Limitations—For large counts, a large number of flip-flops are needed to get the desired results. For instance, to count to 64, 32 flip-flops are needed. If a straight binary divider is used, only six flip-flops are needed. Nevertheless, if synchronous operation is required and all 64 states are to be decoded, the walking ring counter is sometimes cheaper and simpler to use. For decoding, 64 two-input gates would be needed; with a straight binary divider, 64 *six*-input gates would be needed, bringing about serious wiring and fan-out problems, unless some "x-y addressing" scheme can be used.

Walking ring counters are most popularly used with two, three, five, and six flip-flops, for modulo-4, -6, -10, and -12 counting applications. Higher counts are used only if synchronous operation or reasonable decoding of every state is essential.

Note that all toggle inputs are connected together and driven synchronously. This creates quite a fan-in problem in large-modulo counters, and a buffer on the input is usually essential above modulo six. A buffer may be required on the reset bus for counts above modulo ten.

Disallowed Subroutines—There is one very bad feature of walking ring counters, one that can cause endless hours of trouble if it is not recognized and eliminated. Suppose the modulo-6 walking ring somehow gets into state 101. This condition could develop when power is first applied, or possibly a momentary short circuit or noise pulse could inadvertently put the counter in this state. When the counter is toggled, it goes to 010. When it is toggled again, it goes back to 101. It is now a modulo-2 counter. The counter is dividing by the wrong number, and there is no way for it to put itself back on the right track.

It happens that a cascaded chain of n flip-flops can get into 2^n possible states, but a walking ring counter only needs and uses 2n of these states. Thus there are $2^n - 2n$ wrong counter states the walking ring could get into, and once operating in these wrong states, it will stay there until fixed. On a modulo-6 counter, there are 8 possible states, 6 of which are legitimate. The remaining two are the 101 and 010 of the preceding paragraph. On a modulo-10 walking ring, there are 32 possible states, of which only 10 are legitimate. It turns out that with the remaining states you can build two other modulo-10 counters and one modulo-2 counter, none of which decode properly. We call this cyclic sequence through the wrong states a *disallowed subroutine*, and, clearly, if the counter is to be useful we have to somehow get the walking ring counter working on the right sequence.

Eliminating Disallowed States—The simplest way out is to clear the counter to the 000 . . . 000 or the 111 . . 111 state every time power is applied, and better yet, immediately before each measurement or use. This practice always will effect a cure. Often, a monostable can deliver a brief preset pulse at the beginning of each count sequence. A "panic button" on the reset bus may be added in some applications—if the counter ever goes into a disallowed state, pressing the button re-establishes the desired sequence.

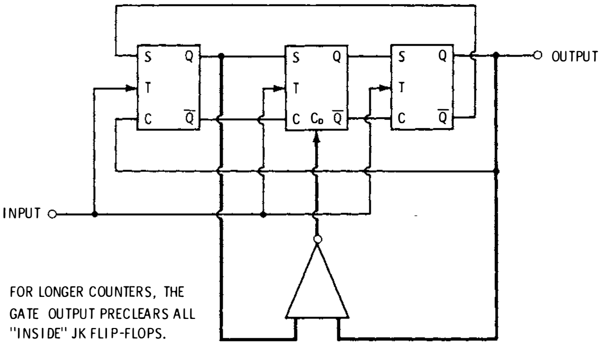

S
Q
T
C
$\overline{Q}$
S
Q
T
C
C_D
$\overline{Q}$
S
Q
T
C
$\overline{Q}$
OUTPUT
INPUT
FOR LONGER COUNTERS, THE GATE OUTPUT PRECLEARS ALL "INSIDE" JK FLIP-FLOPS.

Fig. 5-10. Method for eliminating disallowed subroutines.

To do the job properly and automatically, it is necessary to add an extra NAND gate. There is no possible subroutine on a walking ring counter of any length that does not sooner or later arrive at a state with a 1 at the beginning and a 1 at the end. All that is necessary is to detect this coincidence and, every time it exists, to force all the "inside" flip-flops into the 1 state. This approach is shown in Fig. 5-10. Note that the flip-flops being held at 1 will not change state on the next count in any case, since only the first flip-flop will change. Thus, there will not be a latching problem (see Chapter 6). Also, we now need only reset the outside two flip-flops, as the gate automatically picks up the inside ones as the outside two are reset. Thus a slight savings on reset fan-in may also be achieved this way. Note also that this same gate can provide zero decoding. So, in many applications, no extra parts are needed.

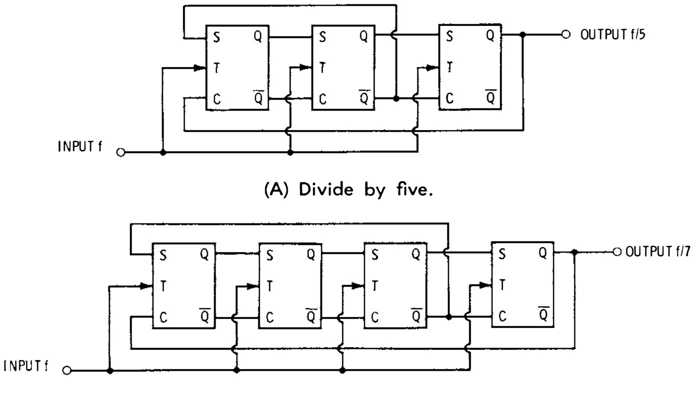
A walking ring counter can be "tricked" into counting to an odd number one less than its normal count by bringing the set input for the first stage from the $\overline{Q}$ output of the next-to-last flip-flop. The odd-length counter has fewer disallowed states than an even-length walking ring counter. Disallowed subroutines cannot occur in rings of length 3,5,7,11, or 13. Typical odd-length counters are shown in Fig. 5-11.

Bucket Brigade

A bucket brigade (Fig. 5-12) is a shift-register configuration that can produce an electronic stepper that is self-decoding and produces a sequential output one point at a time. Basically, this is a "do nothing" circuit: An input signal tells a JK flip-flop to turn itself on; as soon as the JK flip-flop does go on, it realizes its state and enables itself to turn itself off again. Naturally, the JK has to wait for a new negative toggle transition to turn itself off, so the JK remains on for one interval between toggle negative transitions. At the same time, it sets up the next flip-flop down the line to follow the same sequence.

The basic bucket-brigade element is shown in Fig. 5-12A. Assume the clear input is *positive* (0). After toggling occurs, the $\overline{Q}$ output will also be positive (0), and this output will also put a 0 on the set input. With two zeros on the inputs, the flip-flop never changes state after this, regardless of how often it is toggled.

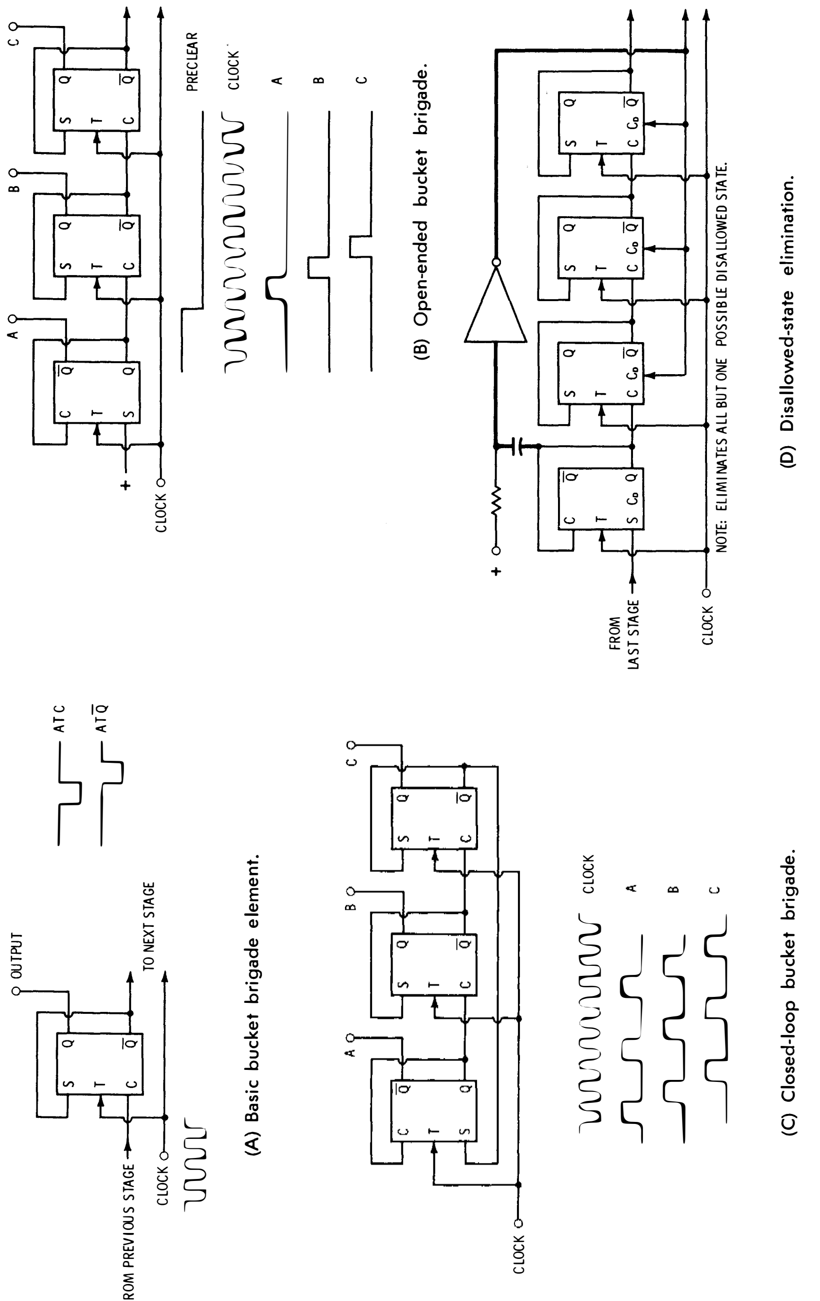
Now, suppose the clear input goes to ground (1). The next negative toggle transition will put the $\overline{Q}$ output at ground (1), and this output places a 1 on the set input. If the 1 on the clear input is now removed, there is a 0 on the clear input and a 1 on the set input. If the clear input stays at 1, there is a 1 on the clear input and a 1 on the set input. Either way, on the next negative toggle transition the flip-flop will go into the opposite state, or the state it was in two counts before. The result of all this is that the $\overline{Q}$ output is normally positive, and goes to ground for the time between toggle negative transitions. At the same time, it enables the next stage down the line to have a grounded $\overline{Q}$ output between the *next* two toggle transitions, and so on. The only limitation on this circuit is that the clear input must become positive again before the *third* toggle transition; otherwise, multiple outputs result.



(A) Divide by five.

(B) Divide by seven.

Fig. 5-11. Odd-length walking ring counters.



(A) Basic bucket brigade element.

(B) Open-ended bucket brigade.

(C) Closed-loop bucket brigade.

NOTE: ELIMINATES ALL BUT ONE POSSIBLE DISALLOWED STATE.

(D) Disallowed-state elimination.

Fig. 5-12. Bucket brigade.

A bucket brigade is built by setting up a chain of flip-flops as shown in Fig. 5-12B. The first flip-flop is connected "upside down" and is preset while the other stages are precleared. As the waveforms in Fig. 5-12B show, each toggle transition passes the on output one stage down the line; the circuit is like a bucket brigade with only one bucket full and the rest empty. After the last transition, the bucket brigade leaves all outputs off until the circuit is preset again. While all this goes on, the respective outputs give sequential positive voltages.

It is possible to close the bucket brigade on itself, as in Fig. 5-12C. Now the circuit behaves as a continuous rotation stepper, with each clock transition moving the step one to the right, 1,2,3,4,5,1,2,3, ... Thus Fig. 5-12B gives a "speak-when-spoken to" type of circuit, while Fig. 5-12C shows a "do-it-yourself-all-the-time" type of circuit.

There are some unique advantages and limitations to the bucket brigade. No decoding at all is required, and only a single flip-flop per stage is needed. Only a single lead is used between stages, simplifying PC layout work. The time with which the input signal goes away is not overly critical, so long as it lasts at least one negative toggle transition and disappears before the third negative toggle transition, so the circuit prevents ever putting out a double pulse. The bucket brigade is open ended—to change the number of stepper points in use, simply add extra flip-flops until the desired count is obtained.

Once again, there are limitations also. Sixty-four steps means 64 flip-flops, but the simplicity of the circuit may justify the expense. Since all toggle inputs are synchronously driven, a buffer on the toggle line is essential. There are even more disallowed states for the bucket brigade than there were for the walking ring counter. Of 2^n possible states only n are useful, the others resulting in no output at all or more than one output on for any given count. Again, the counter should be reset when power is initially applied, and, if practical, once before each count sequence. A mechanical "panic button" may be used also. Remember, the first flip-flop always is connected "upside down." To reset the counter, preset the first stage and preclear all others.

All but one of the disallowed states can be eliminated by detecting a 0 at the first output and using this to force all the other stages into a 1 condition. We must use a monostable and do this on a pulse basis (Fig. 5-12D), for this circuit can cause a latching that will miss counts if we do not. The only way to eliminate the possible 111 . . . disallowed state is to use an n-input NAND gate. This is rarely justified.

Divide-by-Two Synchronizer

Chapter 2 pointed out that simple switches might not be suitable when it is desired to pass only a burst of whole pulses, when the pulses let through must be of a constant width, or when the start of the first pulse let through must occur synchronously with something else in the system.

Any circuit that takes a random outside-world command and puts it in line with internal system timing is called a *synchronizer.* Virtually any synchronization problem may be handled with a single JK flip-flop and one or two external gates.

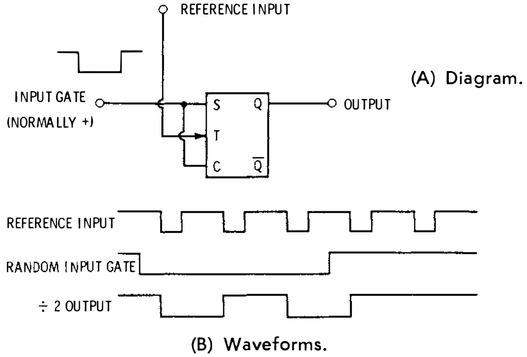
In Fig. 5-13, a single JK flip-flop serves simultaneously as a binary divider and a synchronous gate. The reference signal to be gated or counted is sent to the toggle input; the external random gate command is routed to the set and clear inputs in parallel. If the set and clear inputs are positive, the flip-flop does not change state. If the set and clear inputs are grounded, nothing happens immediately, but there is now a 1,1 condition on the clocked inputs that causes the JK flip-flop to be a binary divider. Starting with the next negative clock transition, the flip-flop behaves as a binary divider, producing an output square wave with one-half the input frequency. Note that we get only "whole lumps" out, regardless of when the input command goes to ground, since nothing can happen except during the negative input transitions.

Whenever the outside-world random command becomes positive again, the action stops. Once again, the binary divider stops on a negative clock transition and not in the middle of a pulse. The net effect is to get out a frequency one-half the input frequency for the synchronization interval that falls within the input command time. This technique is also useful as the first stage of a decimal counter to provide gating.

Straight Synchronizer

There are two forms of straight synchronizer. Neither one divides the reference input frequency by two, and both serve to bring an outside-world gate signal into synchronization with system timing, letting only entire pulses and time intervals pass.

In Fig. 5-14A, an inverter has been added to the input conditioning of the JK flip-flop. Now, if the set input is a 1, the clear input must be a 0,

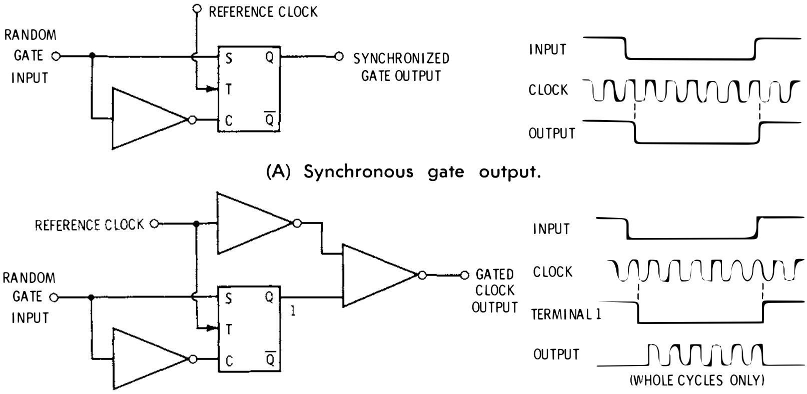


(A) Diagram.

(B) Waveforms.

Fig. 5-13. Divide-by-two synchronizer.

and if the set input is a 0, the clear input must be a 1. The set input is driven with the random outside-world command, and the flip-flop is toggled with system timing pulses. As the waveform diagram shows, we get out a command waveform that goes down on the first negative toggle transition after the input goes down and goes up on the first negative toggle transition after the input does. We get a synchronized output that is perfectly tied into system timing, regardless of when the beginning and the end of the outside-world signal actually occur.



(A) Synchronous gate output.

(B) Synchronous gated clock output.

Fig. 5-14. Straight synchronizers.

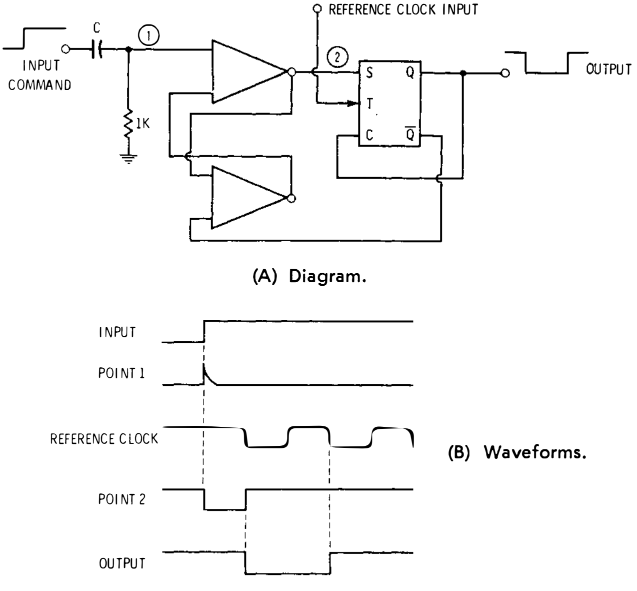
In Fig. 5-14B, a gate has been added to the output. This gate delivers the toggle input to the rest of the system when the input command is being grounded, and stops the toggle input from being passed on when the input command is positive. Once again, we get out only "whole lumps" of the reference frequency—whole lumps that start with the first negative clock transition after the input goes down, and that stop with the first negative clock transition after the input comes up again.

One application for the circuit in Fig. 5-14B is to eliminate apparent count ambiguity on an electronic counter. If an electronic counter is gated with a simple switch, there is a 50-50 chance of turning the switch on either immediately before or immediately after a count goes through. There is a one-count ambiguity, and the last counter digit will "bobble." With a synchronizer, the counter always rounds off this bobble to the next higher digit, eliminating the jitter and apparent display instability. These circuits also see use when it is desired to convert a random event into one that is in step with the rest of the system.

In very high-speed counters, we have to take the time delay of the flip-flop transition and the gate delays into account, or they may shorten or lengthen the first and last gate by several tens of nanoseconds.

One-and-Only-One

How do you build a circuit that provides a precise 1-second on gate whenever you press a button? The answer is once again a form of synchronizer. This one (Fig. 5-15) combines a JK flip-flop with an RS flip-flop to give an output signal that lasts for the duration of two input negative clock transitions. It is triggered by a random, outside-world command.



(A) Diagram.

(B) Waveforms.

Fig. 5-15. One-and-only-one.

In Fig. 5-15A, assume point 2 is positive. Therefore, there is a 0 on the set input, which holds the Q output positive (0). The Q output, in turn, keeps the clear input positive (0), which keeps the flip-flop from doing anything, even though we are feeding in reference toggle pulses whose negative transitions are spaced precisely the desired time width we want as an output.

Now, we feed in a random command to the input. The leading edge of this command causes the RS flip-flop to change state, and point 2 goes to ground. At the next negative toggle transition, the Q output goes to ground. The $\overline{Q}$ output goes positive and resets the RS flip-flop. Meanwhile, the Q output grounds its own clear input. This gives a 0 at the set terminal and a 1 at the clear terminal. The next negative clock transition puts things back the way they originally were, to await a new random command. For each random command, there is one and only one output pulse of a duration exactly equal to the time between two negative toggle transitions.

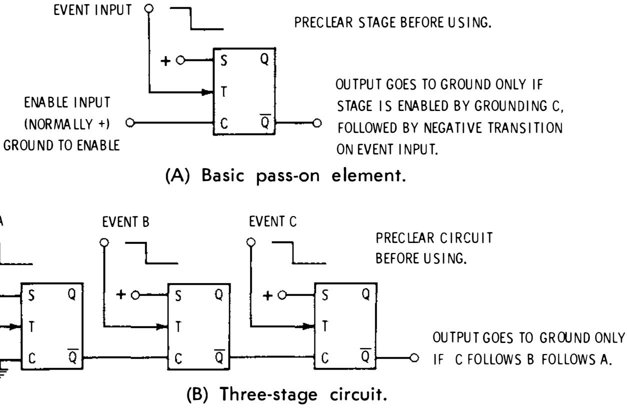
There are many important applications for the one-and-only-one. Generating the timed count commands for an electronic counter is a typical application. Another is taking a random outside-world pulse and getting it into a precise width and time synchronism with internal system timing.

Sequential Pass-On

The sequential pass-on circuit sees use in predetermining counters, electronic locks, and in sequential circuits from which an output is to be obtained only if event A happens before event B before event C. If any event does not happen, or if the events happen in the wrong sequence, an output is never obtained.

The pass-on element is shown in Fig. 5-16A. It is a sort of shift register. Suppose the circuit is initially precleared, making the $\overline{Q}$ output positive. Also suppose the clear input is positive. Nothing happens when the flip-flop is toggled. If the clear input is grounded, the first negative clock transition afterward passes this ground on to the output. Thus all the events happening *before* the input is grounded have no effect on the output; the first event happening after the input is grounded, grounds the output. If there is only one event, it must occur *after* the input is grounded if there is to be an output. If the event happens before input C is grounded, or does not occur at all, we get nothing out.

A three-stage sequential circuit is shown in Fig. 5-16B. Here, to start the action, all the flip-flops are precleared. There is a ground on the clear input of the first stage. If event A happens, it passes on this ground to the second stage. If event B happens after event A, the ground is passed on one stage further. Finally, if C happens, and follows B, the output goes to ground and provides a logic-1 output that can unlatch an electronic lock, sound an alarm, stop a predetermining counter, or make a decade frequency synthesizer stop on the right preselected count.


EVENT INPUT
PRECLEAR STAGE BEFORE USING.
+
S
Q
T
C
Q̄
ENABLE INPUT
(NORMALLY +)
GROUND TO ENABLE
OUTPUT GOES TO GROUND ONLY IF
STAGE IS ENABLED BY GROUNDING C,
FOLLOWED BY NEGATIVE TRANSITION
ON EVENT INPUT.
EVENT A
EVENT B
EVENT C
PRECLEAR CIRCUIT
BEFORE USING.
OUTPUT GOES TO GROUND ONLY
IF C FOLLOWS B FOLLOWS A.

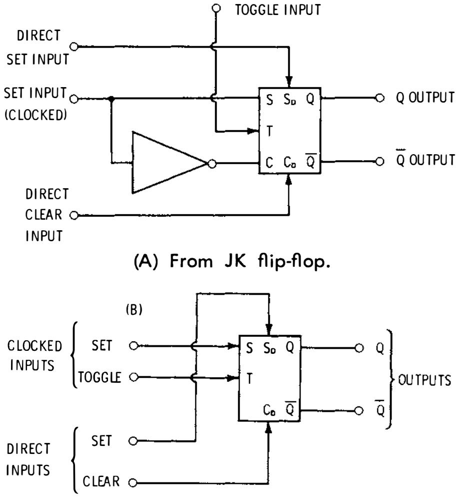
(A) Basic pass-on element.

(B) Three-stage circuit.

Fig. 5-16. Sequential pass-on.

THE TYPE-D FLIP-FLOP

The Type-D flip-flop is somewhat different from the JK flip-flop, and on the whole is slightly less versatile, although it does some of the JK jobs more simply, with lower power, and with fewer stage-to-stage interconnections. Suppose an inverter is added to the input of a JK flip-flop as shown in Fig. 5-17A. Automatically, if there is a 1 on the set input, toggling produces a 1 on the output, and if there is a 0 on the set input, toggling produces a 0 on the output. There is now only one conditioning input. This device is called a Type-D flip-flop. The lead that is saved on the clear terminal can now be used for a preset input, and we still can get two flip-flops in a fourteen-lead package.



(A) From JK flip-flop.

(B) Type-D symbol.

Fig. 5-17. Type-D flip-flop.

The direct set and direct clear inputs behave exactly as they did on the JK flip-flop, except that *they will not work except when the toggle input is high* (+, or 0). This is a peculiarity of the specific way the Type-D configuration is realized in such units as the MC778P.

If we put a 1 on the set input and clock the unit, we get a 1 at the output. If we put a 0 on the input and clock the unit, we get a 0 at the output. The $\overline{Q}$ output, of course, provides the complement each time. There is no way to introduce the 1, 1 and 0, 0 input conditions. Thus we cannot inhibit the Type-D flip-flop and keep it from doing anything. Although we can "trick" the Type-D flip-flop into being a binary divider, we cannot make it start or stop dividing on command, as we could on the JK version; this is the big limitation. Where the Type-D can be used, we gain a simpler circuit layout, dual direct inputs, reduced power consumption, and often an easier-to-implement circuit. It will be shown later that we can make shift registers, walking rings, binary dividers, sequential pass-ons, and one form of synchronizer. Industry uses the Type-D flip-flop mostly for shift registers, particularly where a word is to be entered into a register in parallel form and read out serially. Negative-logic charts for the Type-D flip-flop appear in Fig. 5-18.

DIRECT INPUTS

When the direct inputs are used, the toggle input must be high (+, or 0).

C_D	S_D	Q	$\bar{Q}$
1	1	Q_n	$\bar{Q}_n$
1	0	0	1
0	1	1	0
0	0	1	1
		(Disallowed)	

CLOCKED INPUTS

When clocked inputs are used, direct inputs must be low (grounded, or 1).

Clocked input is entered into flip-flop only on the negative-going toggle transition.

S	Q	$\bar{Q}$
0	0	1
1	1	0

NOTE: Restrictions apply to MC778P dual Type-D flip-flop. For restrictions on any other Type-D unit, consult data sheet.

Fig. 5-18. Negative-logic chart for Type-D flip-flop.

Internally, the actual MC778 differs radically from a JK flip-flop with an inverter on the input, as reference to Fig. 1-11 will show. This is a type of flip-flop called a *master-slave* circuit. Actually there are two related flip-flops, as the logic diagram of Fig. 5-19 shows. When the toggle input goes positive, information is entered into the master flip-flop. When the toggle input goes to ground, the state of the master flip-flop is passed on to the main, or slave, flip-flop. A detailed analysis of the circuit will show that the disallowed-state condition is an essential part of the operation of the master flip-flop.

Type-D flip-flops are considerably more tolerant than JK flip-flops of "sloppy" input signals. This particular flip-flop is still edge sensitive, but since it does not have to rely on stored charge to transfer the logic, the negative clock transitions need not be as sharp as for the JK flip-flop. The clock input voltage only has to fall in less than 100 microseconds, typically.

Since the MC778 is a low-power device, it is limited to a 1.0-MHz maximum operating frequency.

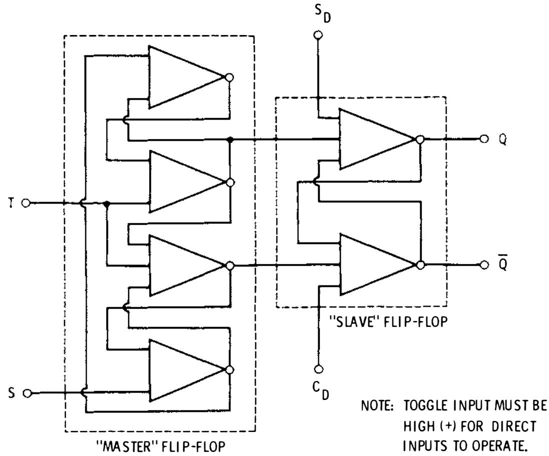

S_D
Q
$\bar{Q}$
T
S
"SLAVE" FLIP-FLOP
C_D
NOTE: TOGGLE INPUT MUST BE HIGH (+) FOR DIRECT INPUTS TO OPERATE.
"MASTER" FLIP-FLOP

Fig. 5-19. Logic-gate equivalent of MC778 Type-D flip-flop.

USING THE TYPE-D FLIP-FLOP

Fig. 5-20 shows several practical Type-D circuits. Fig. 5-20A shows a shift register. Note that both direct inputs are available on each stage and that only one lead between stages is needed now. The information is passed one stage to the right with every clock transition. We can set any word into the register by using the direct inputs any time we wish.

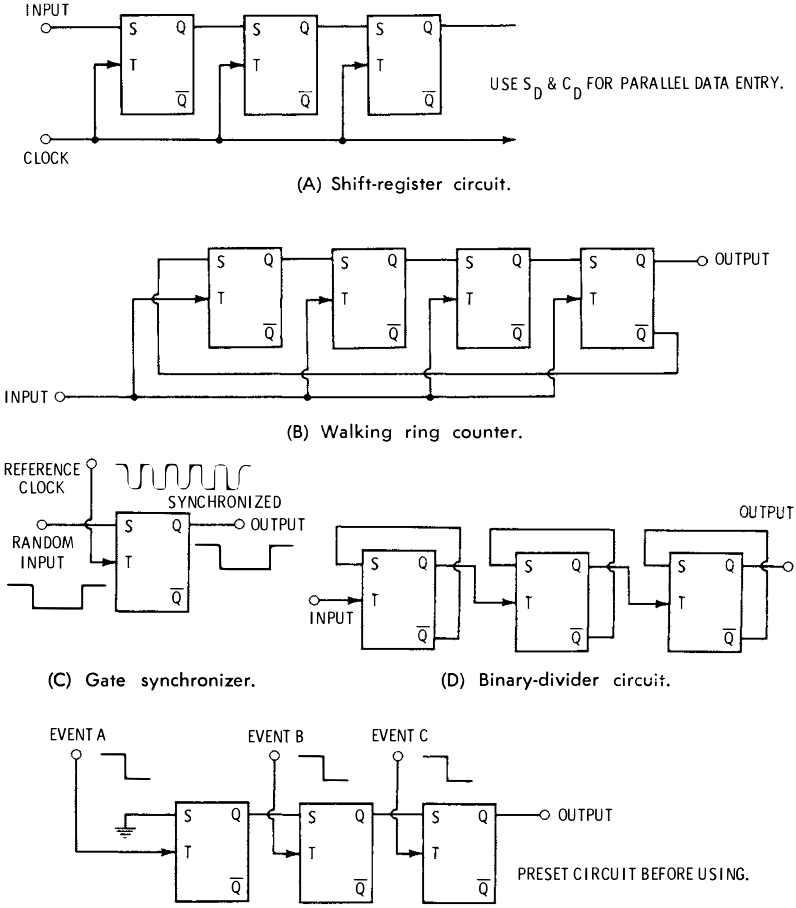
We can leave the shift register open, close it on itself, or have it complement itself. Fig. 5-20B shows a walking ring counter that performs the same way the JK walking ring did, except that now only one lead between stages is needed.

The gate synchronizer circuit is even simpler with the Type-D flip-flop than it was with the JK, since the inverter is apparently built in. Fig. 5-20C gives details.

To build a binary divider, we have to "trick" the Type-D flip-flop into telling itself to go into the opposite state each time. To do this, we externally cross couple the $\bar{Q}$ output to the set input. This connection is shown in Fig. 5-20D.

Fig. 5-20E shows a sequential pass-on circuit using Type-D flip-flops. Other circuits, such as the bucket brigade, certain synchronizers, and the one-and-only-one are not easy to realize with designs in which the Type-D flip-flop is used.

Remember that the direct inputs will not operate properly except when the toggle input is positive. This is an additional constraint on designing and using circuits containing Type-D flip-flops.



(A) Shift-register circuit.

(B) Walking ring counter.

(C) Gate synchronizer.

(D) Binary-divider circuit.

(E) Sequential pass-on.

Fig. 5-20. Applications of Type-D flip-flop.

The choice of Type-D or JK flip-flops for a given circuit is usually easy to make. If you have a repetitive shift-register circuit, need low power, or need minimum interconnections, the type-D flip-flop is often the best choice. Otherwise, use the JK flip-flop. Remember, you can easily convert a JK flip-flop so that it behaves as a Type-D flip-flop (all that is required is an inverter or one more lead from the previous stage), but you can not go backward and make a JK flip-flop from a Type-D flip-flop.

CHAPTER 6

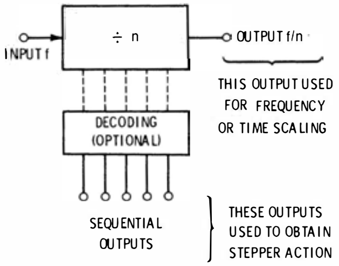
Scaling and Divide-by-n Counting

A *divide-by-n* counter is any circuit that produces an output frequency that is 1/nth the input frequency. The factor can be any whole number, and under certain circumstances n can be a fraction. Fig. 6-1 shows a typical ÷n counter.

There are many uses for ÷n counters. If we square up the 60-Hz powerline waveform and divide by 6, we obtain a 0.1-second square wave useful for time gates in an electronic counter. We can increase the high-frequency range of many digital instruments by a factor of 10, 100, or 1000 with a *decade counter* or *prescaler* added to the input. Electronic digital clocks need circuits that can divide by 10, 12, 2, 6, and 4. Ultra-precise 1-second time gates are obtained by dividing the frequency of a 100 kHz precision crystal clock by 100,000. A musician's pitch reference might start with a 1.0716-MHz crystal. Dividing this frequency by 4096 gives C_4 at 261.6 Hz; dividing it by 2734 gives G_4 at 392.0 Hz.

One classic system example of ÷n counters is a television-service color-dot-bar generator. These instruments often start with a 189-kHz crystal oscillator; this frequency is equivalent to 12 vertical bars in the TV picture. Dividing by 6 gives 31.5 kHz. A further division, by 2, gives the horizontal sweep frequency of 15,750 Hz, while a division by 525 gives the vertical sweep frequency of 60 Hz. Because one division is even and the other is odd, the successive scan fields are properly interlaced. Usually we accomplish the 525 division by dividing first by 35 and then by 15; this makes it possible to generate 15 horizontal lines at 900 Hz. To obtain a crosshatch pattern, we OR the horizontal and vertical lines; for dots, we AND them in a suitable two-input gate.

RTL counting flip-flops make ÷n counters into relatively simple and easily designed circuits with some important advantages over earlier counters. With RTL, the divide-by-n counter is independent of frequency. There are no adjustments and, at the most, only one noncritical resistor and capacitor external to the IC's themselves. No calibration is required and drift is inherently impossible, even with temperature and supply-voltage variations. The division process is intrinsically stable—there is no way a ÷2193 counter can "jump count" to 2192 or 2194. There are no stability controls and there is no warm-up time.



INPUT f
÷ n
OUTPUT f/n
THIS OUTPUT USED FOR FREQUENCY OR TIME SCALING
DECODING (OPTIONAL)
SEQUENTIAL OUTPUTS
THESE OUTPUTS USED TO OBTAIN STEPPER ACTION

Fig. 6-1. Counter symbol and functions.

The *modulo* of a counter is simply the number, n, by which it is dividing. A *low-modulo* counter divides by n factors from 2 through 9. A *high-modulo* counter divides by any n from 11 up. A *decimal* counter divides by 10. In this chapter, design methods will be given for both low- and high-modulo counters, and fractional ÷n techniques will be examined also. Decimal counting techniques are so numerous and so important that they have been saved for the next chapter.

COUNTER QUALITIES

It will be shown that low-modulo counters fall into a "stock-fix" category, whereas high-modulo counters are designed by using one of several techniques that will handle almost any counting problem. We are by no means limited to a single possible method to divide by a given number. There are many division techniques, and, depending upon the *qualities* we want our counter to have, we have a wide choice of possible circuits. Generally, the more qualities we require in a counter, the more restrictive, and the more expensive and complex, the design becomes. There are 14 important qualities to be considered in counter design, as follows:

Modulo

By what n do we wish to divide? Do we want to divide by more than one n? If we do, how difficult will it be to change from one n to the other?

Decoding

Do we need each state digitally decodable? That is, do we want a different terminal for each counter state, just as an electromechanical stepper has? Decoding always increases the cost. There are certain counters that are self-decoding, counters that are easy to decode, and counters that are hard to decode. We will call a counter easy to decode if it needs only a single two-input gate for each state that must be decoded. Counters that run out of fan-out or that need multiple-input gates will be called hard to decode. In very high-speed counters, it may also be necessary to watch for *overlapping decoding*, in which one output may stay on for an instant after a new one comes on. This condition can create problems in special circuits.

Disallowed States

Every counter, except for a straight binary ripple counter, has certain counts it can conceivably get into that are *not* included in the desired sequence the counter is to go through cyclically. If the counter automatically resumes proper operation after a few counts, we say the counter is *self-clearing.* If the counter does not recover after it goes into the wrong state, we have a *disallowed-state problem*, or a *disallowed subroutine.* One way out of this problem is to reset the counter every time power is applied, and preferably before each count sequence. Otherwise, an autoclear circuit must be added to insure the right count.

Resettability

Being able to return to count zero is unimportant in a ÷35 counter used in a television sync generator; in a ÷10 counter used in a frequency counter, the ability to reset to count zero is absolutely mandatory. Some counting schemes require use of the preclear input on each or on some flip-flops, either to get the desired count or as part of the autoclearing process. If this is the case, either you cannot reset the counter, or you would have to use all flip-flops with both preset and preclear inputs. This limitation prevents you from directly using dual JK flip-flops in some specialized counter configurations. In the last chapter it was shown that gate expanders can be added to certain JK flip-flops to overcome this problem partially. If the counter is to be resettable to any count, you will almost certainly need in every stage flip-flops that have both preset and preclear inputs.

It is necessary that the *state* to which the counter resets is the *first* one in the desired count sequence. Otherwise, the resetting operation can make the first count sequence longer or shorter than the n obtained on sucessive counts.

Output Symmetry

The output can be up for half the time and down for the other half only on an even-modulo counter. Some even-modulo counters are symmetrical by themselves; others can be made that way by using an even or odd counter of *one-half* the desired count and following this counter with a binary ÷2 stage. This arrangement may adversely affect the decodeability of the counter.

A symmetrical output is needed whenever it is to be filtered to give a fundamental-frequency sine wave, or if it is to be used as a precision time gate. In other counters, those in which only decoded outputs are of interest, or those that are only going to provide an output negative transition to toggle another counter, symmetry is of no importance, and it may add to the complexity of the counter design.

Weighting

In a *weighted* counter, each flip-flop always represents so many constant units of output. A flip-flop called "4" in a weighted counter always contributes 4 to the output if it is in the 1 state and nothing to the output if it is in the 0 state. For instance, in a decimal counter weighted 1-2-4-8, if the 1 flip-flop is in the 1 state, the 2 flip-flop is in the 1 state, the 4 flip-flop is in the 1 state, and the 8 flip-flop is in the 0 state, the counter must be on count $1 + 2 + 4 + 0 = 7$.

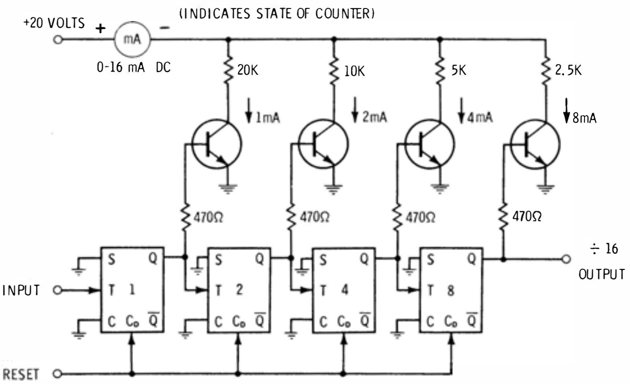


Fig. 6-2. Weighted divide-by-16 counter.

Weighted outputs are particularly useful for converting to an analog output, or for being able to tell instantly what state the counter is in. In our example case, we need only gate 1 mA, 2 mA, and 4 mA to get a 7-mA weighted output that tells us the circuit is on count 7. A 16-bit weighted counter and analog decoder are shown in Fig. 6-2. If desired, weighted outputs can be summed to get a *stairstep* output waveform.

Sometimes two outputs may count for the same weighting. If this is the case, we simply add a prime to the second output to distinguish it from the first output. One form of decimal counter is called a *biquinary* counter and is weighted 1-2-2′-4.

Synchronism

In a *synchronous* counter, those outputs that change state on any given count do so not longer than the delay of one flip-flop after the input to the entire counter falls toward ground. There is never the necessity for a whole chain of flip-flops to trigger each other sequentially, nor are there any gating or reset delays. Synchronous counters are usually the fastest possible, and they are often needed in complex systems, where the ÷n outputs are not allowed to "get behind" the system clock by more than the delay of a single stage. Synchronous counters are usually more expensive and far more complex than nonsynchronous ones.

Speed Limitations

A synchronous counter usually will run at the "wide-open" speed of the IC's themselves. This is *not* true of most other counter types, particularly high-modulo counters. For instance, in one particular type of counter, the sequence may include the sequential triggering of fifteen flip-flops, followed by the triggering of a monostable, followed by a reset and a settling operation. During all this time, no new input negative transitions can be allowed to arrive. Thus, certain counter types cycle considerably more slowly than the capabilities of the individual flip-flops would indicate. With RTL, if operation is to be faster than 500 kHz, it is necessary to check the maximum permissible operating speed of higher-modulo counters to be certain it is well within the total allowed cycle time.

The next chapter will show that one way to get a decimal up-down counter is to use a "9's complement" technique in which all the input signals are multiplied by a factor of nine. This technique can drastically reduce the maximum operating speed, perhaps by a factor of 20. Other shift-register and DDA counting and calculating schemes may require that the shift register be "marched around" once for each input pulse. For long registers, this requirement can drastically limit the operating speeds, often down into the 20-50 kHz region, even though JK flip-flops capable of 8- or 10-MHz operation are being used.

Fan-In

A *low-fan-in* counter has on the input only a single toggle lead to be driven. A *high-fan-in* counter has more than one toggle lead on the input (this is almost always the case with synchronous counters). On high-fan-in counters, it is often necessary to add an input buffer, an added expense that will require supply power. Resettable, high-modulo counters require a buffer on the reset line as well.

A related problem is the cascadeability of the counter. Will it have enough fan-out to drive the next counter down the line, or will a buffer have to be added? We also have to watch the output on cascaded counters. The output line must have one and only one negative transition per count sequence, and this transition must be located the full count cycle "away" from the 0000 count state; otherwise there will be an apparent "phase shift" between the cyclic operations of the two cascaded counters. For instance, on a decimal counter output, we want the negative-going toggle transition at the output to occur only on the zero count following count nine; otherwise we will get carries when we do not want them.

This problem can be serious on some exotic ÷n counter schemes that are currently popular. The problem comes in with an output that has more than one negative transition, or an output that has extra glitches (see below) in it at counts different from the one at which a carry is wanted.

Minimum Counters

A minimum counter uses only the theoretical minimum number of JK flip-flops. This number is always equal to the exponent of the least power of 2 that is equal to or greater than n. For instance, a ÷30 counter can be built with no fewer than the 5 flip-flops it would take to build a 2^5, or modulo-32, counter.

Minimum counters are often the logical choice when your only interest is *scaling*, or producing a ÷n counter with no decoding or symmetry restrictions. While a minimum counter will always be the cheapest one possible, minimum designs are usually difficult to decode and may require a relatively complex PC layout. Most properly designed minimum counters have no disallowed states, but in a new or home-made counter using only JK flip-flops, it is possible that very subtle disallowed-state and glitch conditions could appear.

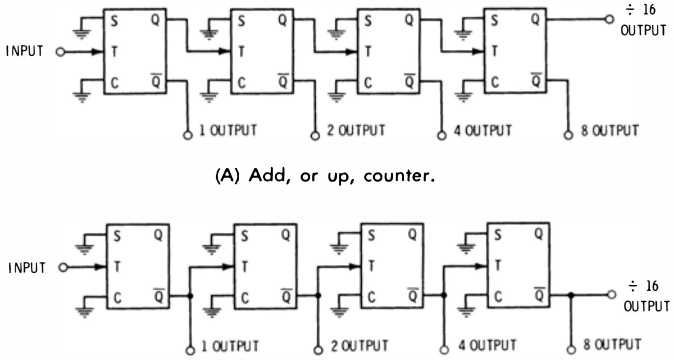
PC Layout

Almost always, a PC board is used for a ÷n counter. Some counter types may require double-sided boards or a large layout with an excessive number of jumpers, adding to either the size or the cost. When many counters are to be built, this factor becomes important.

Direction of Count

An *add*, or *up*, counter counts 0, 1, 2, 3, 4 . . . . A *subtract*, or *down*, counter might count 35, 34, 33, 32 . . . . The counting direction is similar to the direction of rotation of a mechanical stepper and determines the *sequence* in which the outputs are energized. Fig. 6-3 shows a ÷16 up counter and a ÷16 down counter. One goes through its counts in the "normal" manner; the other goes through the sequence "backward." Simply by renumbering the *decoded* outputs, either counter can be used to go through either sequence.

There is an important point here. It usually does not matter what sequence the counter goes through to get where it is going, so long as we know what it is doing and so long as it does the desired job. For most applications, a decimal counter that starts with a binary 15 state and counts down to 6 is just as useful as one that counts in the "regular" way. Requiring a counter to go through a sequence in a specified order is often an artificial, expensive, and unnecessary restriction. We simply pick the proper decoding and labeling on the decoded output terminals. This is particularly



(A) Add, or up, counter.

(B) Subtract, or down, counter.

Fig. 6-3. Negative-logic divide-by-16 counters.

true of a high-modulo scaler. It is easier—and cheaper—to "take the numbers out of the middle," rather than from either end.

Bidirectional Counters

Bidirectional counters operate in either direction, just as a two-coil mechanical stepper does. Bidirectional counters are expensive and normally are avoided wherever possible.

There are several design headaches with a bidirectional counter. You must make sure that a negative toggle transition does not occur when you switch from addition to subtraction. You must make sure that decoding stays the same for each counter state in both the add and subtract mode. You must provide for no count on a simultaneous add and subtract command. And finally, you must provide for negative numbers. This provision often requires an extra flip-flop, called a *sign bit*, which lets you retain meaningful numbers as you subtract, say, a series of 1's from 4. You should get 4, 3, 2, 1, 0, −1, −2 . . . . Without provision for negative numbers, the counter flips up to its *highest* possible count every time you take a 1 from 0. Several practical up-down counters will be shown in the next chapter.

Sometimes it is possible to "trick" an up-only counter into behaving as a bidirectional one, but at a considerable sacrifice of operating speed. On a decimal counter, a "9's complement" technique is used for this purpose (this technique is covered in the next chapter). Basically, for addition the counter is used in the normal manner. For subtraction, the input pulses are multiplied by *one less than n*, and these multiplied pulses are added with the counter. The correct answer results, but it is obtained the "long way around." To borrow, the carries produced by the up counter are compared with the number of input pulses. If a carry is missing, you borrow. The technique is somewhat subtle, but it can be very inexpensive, and its only disadvantage is lack of speed.

Power Level

Long, cascaded ÷n counters can require as much as several amperes of current at 3.6 volts. We have a choice of RTL power levels that may be used, with a compromise among speed, the extra fan-out available for decoding, cost, the number of packages used, and resettability. Table 6-1 lists pertinent factors concerning several available devices.

Table 6-1. Counting Flip-Flops for a ÷n Counter

Device	Type	Stages/ Package	Power/ Stage	Speed	Fan-Out	Presettable?
Super JK	MC779	1	180 mW	4 MHz	80	Yes
Medium-	MC723	1	91 mW	4 MHz	10	With expander
Power JK	μL923	1	91 mW	4 MHz	10	With expander
	MC726	1	100 mW	4 MHz	16	Yes
	MC790	2	91 mW	4 MHz	10	With expander
	MC791	2	95 mW	4 MHz	16	No
Low-Power JK	MC776	2	20 mW	3 MHz	2	With expander
Type D*	MC778	2	20 mW	1 MHz	3	Yes

*Where applicable. Presettable only with toggle input high.

Usually, the medium-power flip-flops represent a good compromise among cost, fan-out, and power consumption. Buffered-output flip-flops such as the MC791 can sometimes be used to save a gate or two in special circuits, but this can introduce other circuit problems. Long cascaded chains of MC791P's or other buffered flip-flops can require very long preset pulses because unwanted pulses can sometimes be propagated through the chain during preset time.

The Type-D flip-flop is also included in Table 6-1, even though its use is limited to certain types of counters. Bear in mind there is an additional restriction on the Type-D flip-flop not found on the JK flip-flop—the direct inputs can be used only when the toggle input is high.

DESIGN PITFALLS

There are some common design errors and assumptions that cause erratic, faulty, or no operation in ÷n counting, and are the source of countless headaches in poor designs. Before building some ÷n counters, it is advisable to look at some good ways *not* to do the job.

By now, it should be obvious that we must provide the proper supply voltage of 3.6 volts, plus or minus one-half volt, well filtered; that we must have proper high-frequency supply bypassing; and that we must have large-diameter wire or wide foil supply and ground runs. The power supply must handle enough current for all the flip-flops—which can be a considerable amount in long-length counters—and its regulation must be good enough that its output voltage (including the effects of ripple) stays within one-half volt of 3.6 volts for all possible counter connections and states.

We must observe the same input-conditioning restrictions for a ÷n counter as we did for a single JK flip-flop. The input must have an abrupt negative-going transition lasting between 10 and 100 nanoseconds. This transition must occur once and only once per desired count, with the transition starting at an amplitude between +1.5 and +4.5 volts and dropping to less than +0.3 volt. *All* input signals must be suitably conditioned to meet these requirements. For very high-speed counters (those operating above 2.0 MHz), the input waveform also should be symmetrical. When Type-D flip-flops are used, conditioning is also required, although the requirements are not as severe. Remember, a Type-D flip-flop cannot be preset or precleared when its clock, or toggle, input is positive.

We also have to watch the fan-in availability and the fan-out capability to make sure these are never exceeded. We must be careful not to short-circuit any output, except on buffered flip-flops such as the MC791 or a super JK. If the counter is to be reset, we have to be certain we have a sufficiently long reset pulse and that we reset to the first state in the ÷n count sequence.

The foregoing are some of the more obvious errors to avoid. Now we turn to some more subtle design mistakes.

Permanent Latching With the Direct Inputs

If the direct inputs of a counter or flip-flop are used at all, they *must* be allowed to go to ground before the next count arrives. Otherwise, the counter either misses a count or two or permanently holds itself in a single state. This condition is called *latching* a counter.

For instance, Fig. 6-4 represents an attempt to build a ÷3 counter by using a ÷4 binary divider and preclear feedback. Suppose we start with the state in which both Q outputs are high, or the 0,0 state with negative logic. Toggling once changes the outputs to 1, 0, and toggling once again gives 0, 1. When the second flip-flop goes into the 1 state, its $\overline{Q}$ output goes positive and preclears the first stage so that the counter assumes the

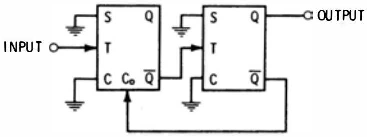


Fig. 6-4. Divider circuit that latches.

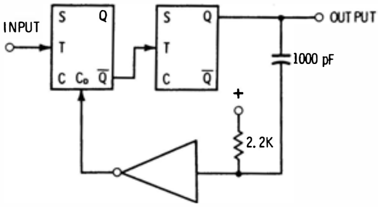


Fig. 6-5. Divider that does not latch.

1,1 state. The counter has gone through four counts in only three clock transitions, and seemingly is ready to go again, but the circuit is not a ÷3 counter. We may have gotten the counter to go through three states, but now the second stage is preventing the first one from changing by holding its preclear input positive. Since the first stage cannot change, neither can the second, and the circuit will now ignore all input toggle transitions. We have latched the counter.

We may solve the problem in one of two ways: we may design a counter that does not require use of the direct inputs, or we may add a monostable or half monostable to generate a *brief* preclear pulse that will disappear *before* the next input negative transition arrives. One divide-by-3 counter that will work (it is not a minimum one) is shown in Fig. 6-5.

Failure to Compensate for a Disallowed Subroutine

A chain of n JK flip-flops can assume any of 2^n possible states. We must always examine all *unused* states to make sure that no combination of unused states can get the counter permanently "in a rut." This problem is solved by designing counters that have no disallowed subroutines, or, if disallowed states or subroutines exist, by resetting the counter every time power is applied—or better yet, just before each use. Another possibility is the "panic button."

If none of these fixes is acceptable, it is necessary to devise logic circuitry that detects at least one state in every possible disallowed subroutine and uses this detected coincidence to jump the counter into the proper sequence. An example of this approach was shown in the modulo-6 walking ring counter of Fig. 5-10. The counter has eight possible states and only six useable ones; the other two form a modulo-2 subroutine. The NAND gate detects the subroutine and forces the counter into the proper state. This technique works on a walking ring of any length by NANDing the two outside 1's and using this coincidence to preclear all inside stages to 1 outputs. For this particular circuit, the flip-flops being precleared do not change state on the next count, so there is no latching problem. Other counter types may require more exotic subroutine-correcting circuitry.

Self-Annihilating Coincidence

The self-annihilating coincidence is a design subtlety that causes many problems, particularly in attempts at decimal counting. This phenomenon causes the possibility that 99 of 100 circuits built will work properly and the last one will positively refuse to work. It may also cause the count to appear to be supply or temperature dependent.

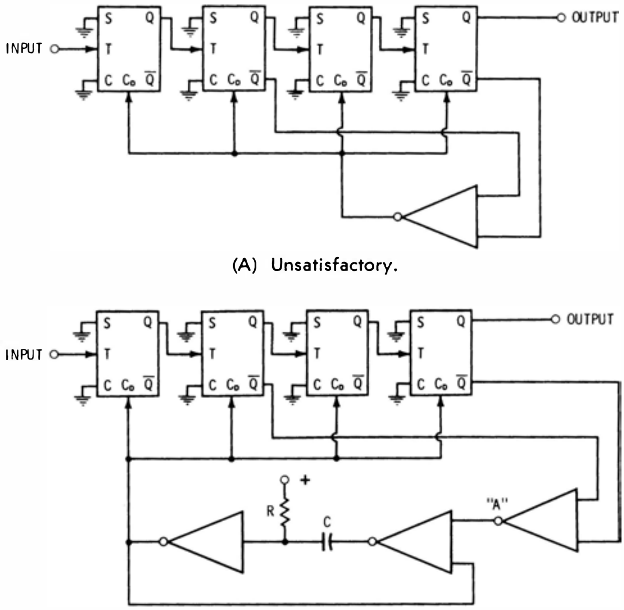
Basically, the output of a gate cannot be allowed to affect the input it is receiving. This point can be illustrated easily as follows. Suppose we build a straight binary ÷16 counter and try to preclear the entire counter to zero on the tenth count (Fig. 6-6A). The first time the $\overline{Q}$ outputs on the second and fourth flip-flops will simultaneously be down is on count 10. On this coincidence, the positive output of the NAND gate is used to preclear the counter back to zero.

This system works so long as each flip-flop requires the same amount of time to preclear. But what if flip-flop No. 4 preclears more quickly than flip-flop No. 2? As soon as flip-flop No. 4 preclears, the $\overline{Q}$ output on this flip-flop goes positive. The output of the NAND gate goes to ground, and the voltage that was supposed to preclear the other three stages disappears. The counter will preclear only those flip-flops it "feels like" preclearing, and practically any count may result.

There are several ways to get around the problem. One is to avoid using a self-annihilating coincidence in a counter. Another is to add a monostable to the NAND-gate output—then when the NAND-gate output goes to ground, the monostable takes over and delivers a positive preclear pulse long enough to guarantee that each individual stage has a chance to preclear fully. There are also more subtle solutions that involve adding extra gates to pick up enough delay to insure proper operation. A suitable monostable circuit is shown in Fig. 6-6B.

Glitch, or Premature Coincidence

In a JK flip-flop, there is a built-in delay of almost 100 nanoseconds between the time the input toggle negative transition occurs and the time the



(A) Unsatisfactory.

(B) Workable.

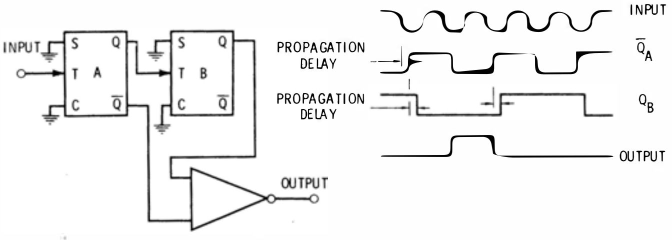
Fig. 6-6. Divide-by-10 circuits.

output completely changes state. While this delay stops any wild races through a cascaded chain of flip-flops that are synchronously driven, it also can produce some gating problems.

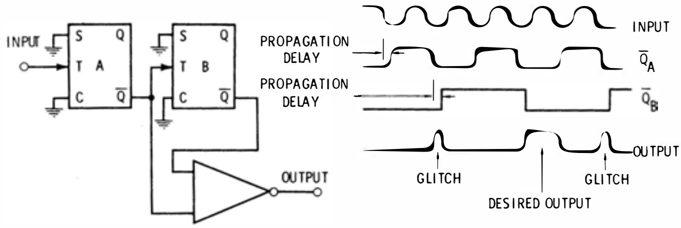
The fact that signals are delayed from each other slightly means there can be an unwanted brief output transition, called a *glitch.* A glitch can take the form of a brief positive-going spike at the output of a gate. The spike rarely lasts for more than 100 nanoseconds. If this output is fed to a light bulb or other outside-world resistive load, it probably would never be noticed. But if the output is intended to preset or preclear a counting chain, or, worse yet, toggle a flip-flop, an unwanted reset or carry would be produced at the wrong time.

A glitch is also called a *premature coincidence*, since the apparent effect is that a gate produces an output far ahead of when it is supposed to. As an example, consider the fragment of a counting chain shown in Fig. 6-7, in which we try to NAND the outputs of two successive JK flip-flops. The waveforms, *including delays*, are shown. In circuit 6-7A, there is no glitch. In circuit 6-7B, however, there is a glitch ahead of the desired output. The difference is that the delayed coincidences overlap in Fig. 6-7A and are separated in Fig. 6-7B. Some apparently similar gating situations may be completely glitch-free or glitch-riddled, depending only on a single gate connection.

How do we eliminate glitches? First, we determine whether there are any, by carefully drawing out all waveforms *including gate and flip-flop delays*, allowing 20 nanoseconds per gate and 100 nanoseconds per flip-flop. Any possibility of glitches will promptly appear.



(A) No glitches produced.



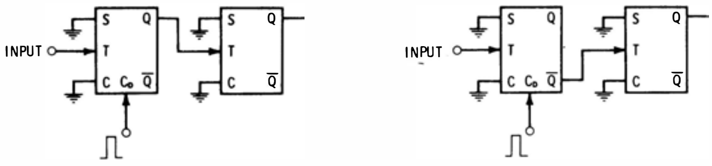
(B) Glitches produced.

Fig. 6-7. Glitches caused by propagation delays.

Glitches can often be eliminated by redesigning the counter so that it has none possible, or the glitches may be routed harmlessly to outside-world loads where they can do no harm. Also, delays—cascaded inverters—may be added to get the offending signals back into line. Sometimes it is possible to gate in the complement of the original input signal so that the gate output is allowed to go positive only long after all possible glitches have occurred. It is also possible to design fully synchronous counters, a more expensive technique that minimizes the glitch problem significantly. But the best solution of all is to avoid using glitch-prone circuits altogether.

Adding More Than is Expected

A brief preclear pulse changes the state of a flip-flop. This is a useful way of adding in counts to shorten the length of a count sequence, but if we ever produce a *carry*, or present a negative toggle transition to the next stage, we will add in more counts than we realize. For instance, in Fig. 6-8A, a brief preclear pulse is delivered to the first flip-flop, driving the Q output to ground if it is not already there. There can be a negative toggle transition for the next stage, and sometimes a carry is produced and sometimes it is not. Fig. 6-8B gets around the problem. Here the same preclear pulse is used in the same manner, but the next stage is toggled from the $\overline{Q}$ output. When we preclear now, the $\overline{Q}$ output can only go or stay positive—no negative transition is possible, and we have prevented adding more than we really wanted to add.



(A) Carry produced.

(B) No carry produced.

Fig. 6-8. Elimination of unwanted carries.

Care must be exercised also whenever the connections to a ÷n counter are changed. If a toggle input abruptly goes from + to ground during the switching process, unwanted counts will be added into the sequence. This is a particularly thorny problem when an up-down counter is switched from add to subtract.

When an entire counting chain is reset by means of the preclear inputs, it does not matter whether unwanted transitions occur or not, so long as the preclear signal lasts longer than the settling time of all the transitions that are produced. In other words, a counter will always preclear if you hold the preclear signal on long enough, even if the counter has been toggled previously.

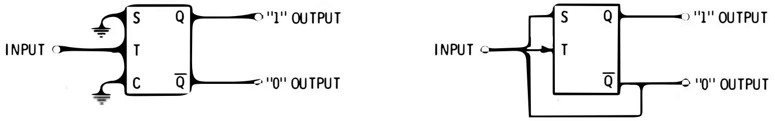
LOW-MODULO COUNTERS

Low-modulo counters fall into a "cookbook," or "stock-fix," category in which you simply pick the right counter to do the desired job. High-modulo counters are designed by picking the *method* that works for the proper modulo and the specific qualities wanted in the counter. The discussion here will not dwell on *why* the low-modulo counters work—operation of most of them will become apparent when the high-modulo methods are described. Any counter can be analyzed by picking a known state (usually the 11111 or 00000 state) and seeing what happens on each successive count to each flip-flop and each gate, including delays. This technique has to work and will show either why a counter will work or that it will not work in the desired manner.

We now turn to the stock-fix, low-modulo counters.

Divide by Two

For division by 2, it is possible to use either of the stock binary divider connections, either by taking the JK flip-flop and grounding the S and C inputs (Fig. 6-9A), or by taking the Type-D flip-flop and cross coupling the $\overline{Q}$ output to the S input (Fig. 6-9B) so that the stage automatically flips with each input count. Either circuit gives one transition each time the clock input comes down, producing a symmetrical output square wave. We can start the action with the direct inputs, putting the flip-flop in the desired initial state.



(A) JK binary.

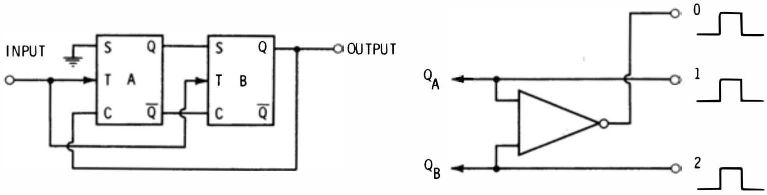
(B) Type-D binary.

Fig. 6-9. Divide-by-2 circuits.

Both circuits are self-decoding, symmetric, synchronous, and weighted (the output counts for "1"). They are minimum, have low fan-in, and have no disallowed states.

Divide by Three

The usual ÷3 counter is the odd-length walking ring of Fig. 6-10A. The lead that is apparently missing is provided inside the IC. This circuit is synchronous, autoclearing, minimum, has a 2:1 asymmetry, and is weighted 1-2. Two of the states are self-decoding; the third requires a single two-input gate as shown in Fig. 6-10B. On the debit side are a high fan-in (two toggle inputs in parallel) and a relatively complex PC layout.

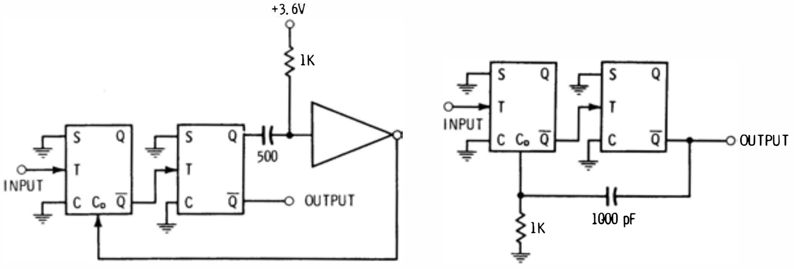


(A) Circuit.

(B) Decoding.

Fig. 6-10. Odd-length walking ring (divide-by-3).

The halfway-addition counter technique (discussed in more detail later) overcomes these debits at the expense of not being synchronous and of requiring one preset input for resettability. The circuit is shown in Fig. 6-11A. On count two, one count is added, shortening the counter from a normal length 4 to a modulo 3. The counter is autoclearing, has a 2:1



(A) With half monostable.

(B) Low-frequency circuit.

Fig. 6-11. Halfway-addition divide-by-3 counters.

asymmetry, and is weighted 1-2. For lower-frequency circuits (100 kHz or less), the half monostable may be eliminated and replaced by an RC network on the $\overline{Q}$ output, as shown in Fig. 6-11B.

Fig. 6-12 shows a ÷3 bucket-brigade counter that is self-decoding and synchronous. It has 5 disallowed states.

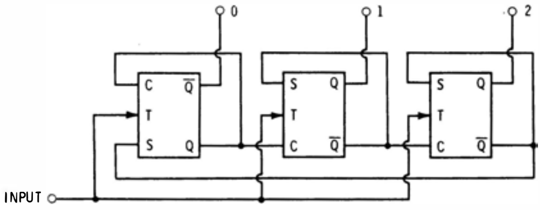


Fig. 6-12. Divide-by-3 bucket brigade.

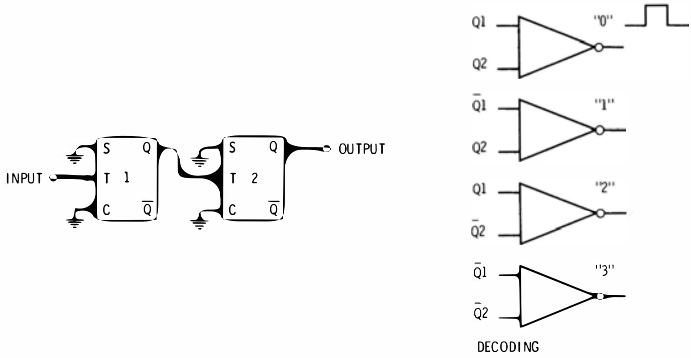
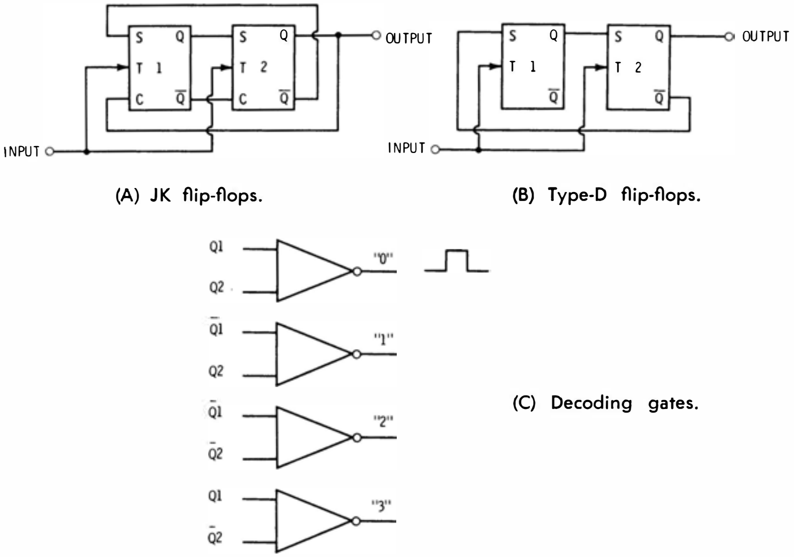


Fig. 6-13. Divide-by-4 binary ripple counter.

Divide by Four

Division by 4 is easy since 4 is a binary power. Fig. 6-13 shows a straight binary ripple counter, which is weighted 1-2, has low fan-in, and is decoded as shown. It is not synchronous.



(A) JK flip-flops.

(B) Type-D flip-flops.

(C) Decoding gates.

Fig. 6-14. Divide-by-4 walking ring counters.

Two walking ring counters are shown in Fig. 6-14A and Fig. 6-14B, the first using JK flip-flops, the second using Type-D flip-flops. They are both synchronous, easy to decode (Fig. 6-14C), unweighted, and have no disallowed states. Fan-in is high with two parallel toggle inputs, and PC layout is more complex than for the straight binary ripple counter.

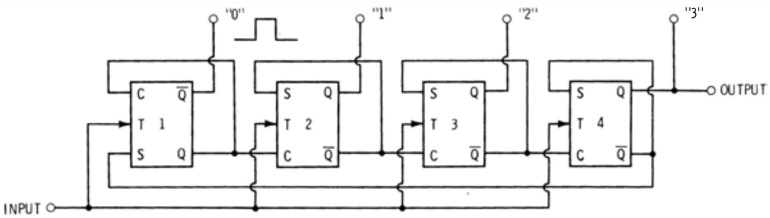


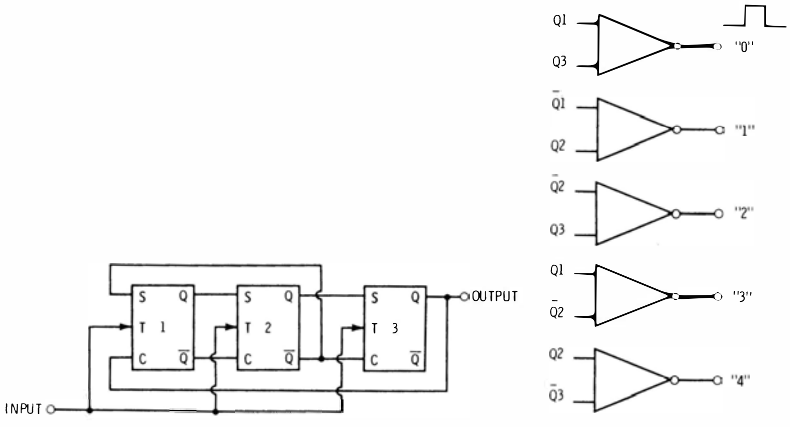
Fig. 6-15. Divide-by-4 bucket brigade.

A modulo-4 bucket brigade appears in Fig. 6-15. It has a high fan-in, is synchronous and self-decoding, and has 12 disallowed states.

Divide by Five

The odd-length walking ring counter of Fig. 6-16 is synchronous, is minimum, and has a 3:2 output asymmetry. There are no disallowed states, but the fan-in is high, and the PC layout is relatively complex. The counter is unweighted but can be decoded easily with five two-input gates as shown.

Fig. 6-17 shows a counter with a 1:4 output asymmetry and a high fan-in (two toggle inputs). It is weighted 1-2-4, is nonsynchronous, autoclearing, but hard to lay out.



(A) Circuit.

(B) Decoding.

Fig. 6-16. Divide-by-5 odd-length walking ring.

Very simple decoding is the prime feature of the quinary counter in Fig. 6-18. Fan-in consists of two parallel toggle inputs. The counter is not synchronous and is weighted 1-1′-2. Two states are self-decoding. Three two-input gates may be used to decode the other three states *if* a buffered flip-flop (MC791P, etc.) is being used. Otherwise, one three-input and two two-input gates are needed. There are no disallowed states.

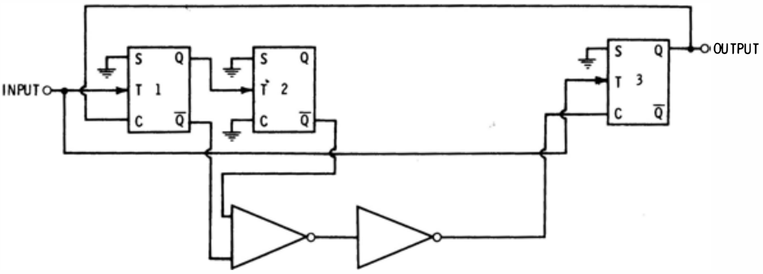
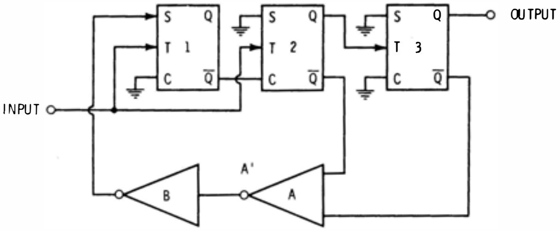
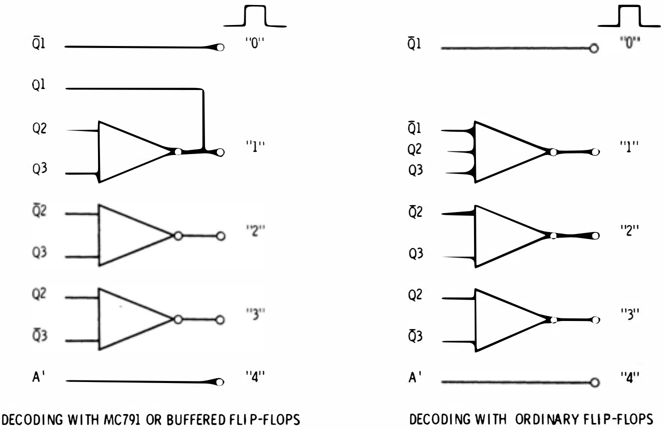


Fig. 6-17. Divide-by-5 circuit.

The self-decoding bucket brigade for modulo 5 is shown in Fig. 6-19. High fan-in, synchronous operation, and self-decoding are combined with 27 disallowed states.



(A) Circuit.


Q̄1
"0"
Q1
Q2
"1"
Q3
Q̄2
"2"
Q3
Q2
"3"
Q̄3
A'
"4"
DECODING WITH MC791 OR BUFFERED FLIP-FLOPS
Q̄1
"0"
Q̄1
Q2
"1"
Q3
Q̄2
"2"
Q3
Q2
"3"
Q̄3
A'
"4"
DECODING WITH ORDINARY FLIP-FLOPS

(B) Decoding.

Fig. 6-18. Quinary counter.

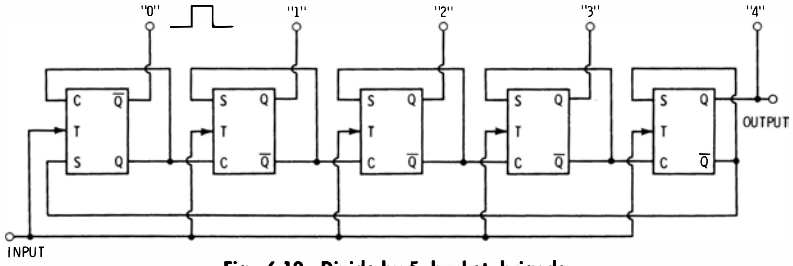
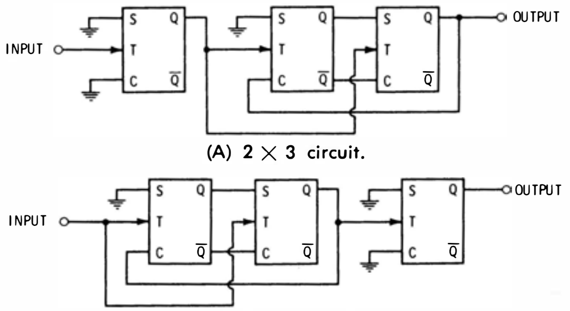


Fig. 6-19. Divide-by-5 bucket brigade.

Divide by Six

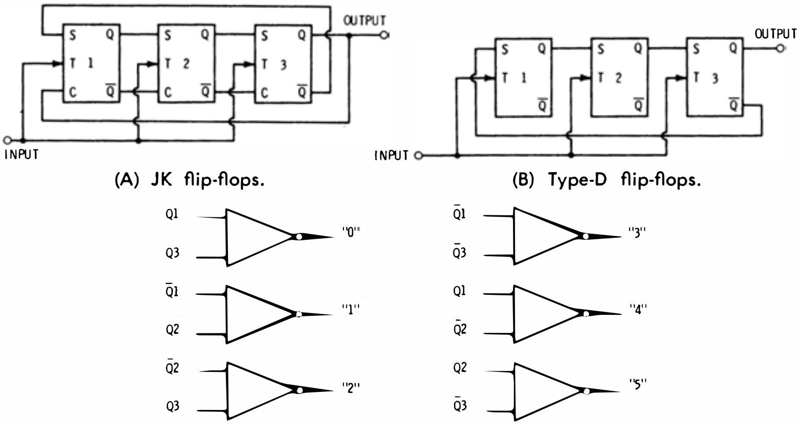
Because 6 can be factored into 2 × 3, we can divide by 6 by building the 2 × 3 counter in Fig. 6-20A or the 3 × 2 counter in Fig. 6-20B. The 2 × 3 counter is weighted 1-2-4 and has a low fan-in and a 1:2 output asymmetry.



(A) 2 × 3 circuit.

(B) 3 × 2 circuit.

Fig. 6-20. Divide-by-6 counters.



(A) JK flip-flops.

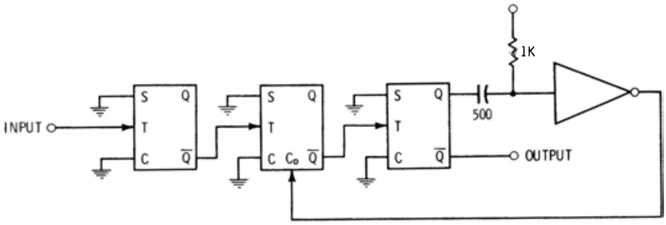
(B) Type-D flip-flops.

(C) Decoding gates.

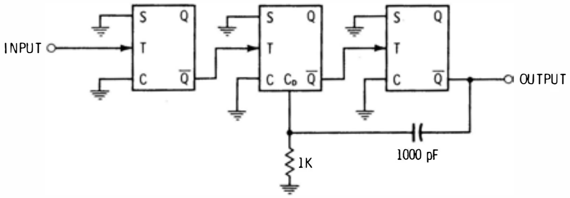
Fig. 6-21. Divide-by-6 walking rings.

The 3 × 2 counter has a higher fan-in, but it produces a symmetrical output. Neither counter is synchronous. Both autoclear, are minimum, and are hard to decode.

The walking ring counters of Fig. 6-21A and Fig. 6-21B are synchronous with a symmetric output. They may be decoded with two-input gates as shown in Fig. 6-21C. The fan-in is high, and there are two disallowed states, which may be removed as shown in Fig. 5-10. Both counters are unweighted.



(A) With half monostable.



(B) Low-frequency circuit.

Fig. 6-22. Halfway-addition divide-by-6 counters.

Fig. 6-22A and Fig. 6-22B show halfway-addition ÷6 counters. They are unweighted, nonsynchronous, asymmetric, and hard to decode. A preset input is required on one flip-flop for resettability. Fan-in is low and PC layout is easy. The counter in Fig. 6-22B is limited to lower speeds.

Fig. 6-23 is the self-decoding, synchronous bucket brigade. Fan-in is high, and there are 58 disallowed states.

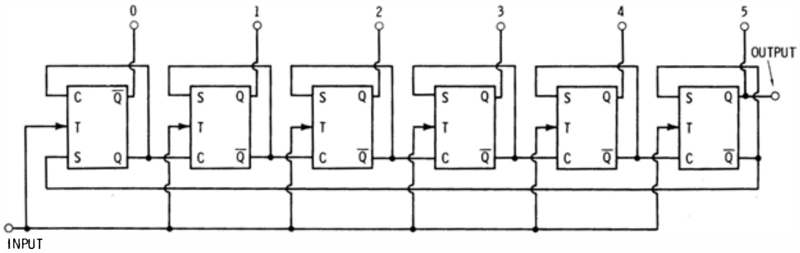
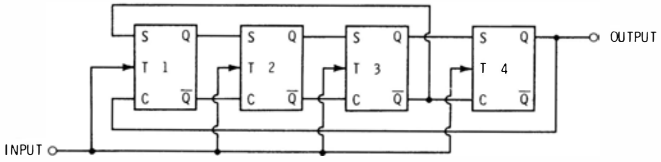
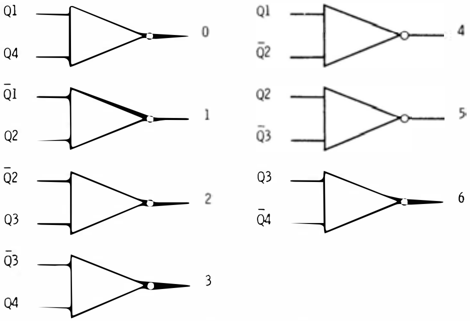


Fig. 6-23. Divide-by-6 bucket brigade.



(A) Circuit.

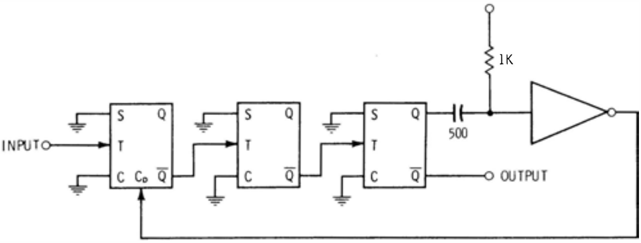


(B) Decoding.

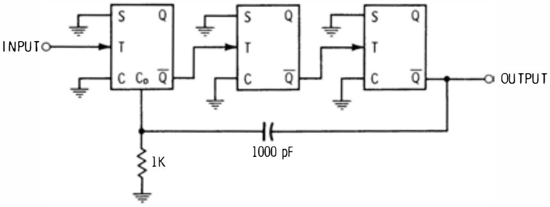
Fig. 6-24. Divide-by-7 odd-length walking ring.

Divide by Seven

The odd-length walking ring of Fig. 6-24A is synchronous, has no disallowed states, and has a 3:4 output asymmetry. Fan-in is high, the counter is unweighted, PC layout is difficult, and decoding is easy (Fig. 6-24B).



(A) With half monostable.



(B) Low-frequency circuit.

Fig. 6-25. Halfway-addition divide-by-7 counters.

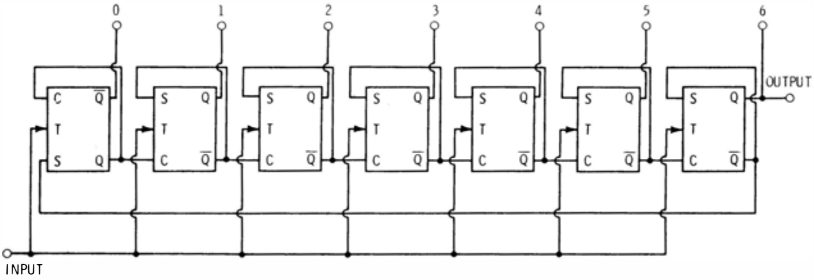


Fig. 6-26. Divide-by-7 bucket brigade.

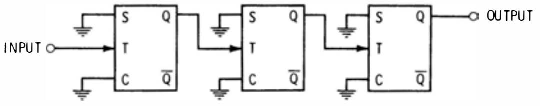
The half-way addition counters of Fig. 6-25 have low fan-in, an easy PC layout, and no disallowed states. They are unweighted, asymmetric, hard to decode, and not synchronous. Reset is possible with one preset input.

The bucket brigade is shown in Fig. 6-26. High fan-in and 121 disallowed states are combined with a weighted output that is also synchronous and self-decoding.

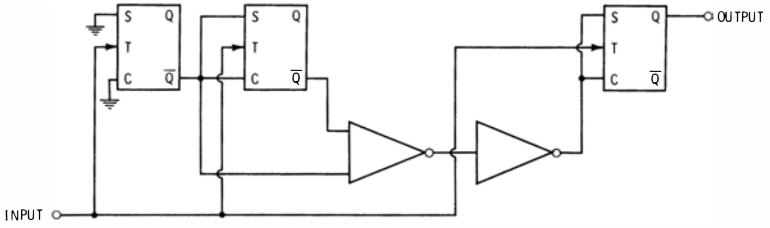
Divide by Eight

Eight is a binary power, so division by 8 is relatively easy. Fig. 6-27A shows a straight binary ripple counter; it is weighted 1-2-4, is symmetric, has low fan-in, and is easy to lay out. It is nonsynchronous and hard to decode. A fully synchronous ÷8 counter is shown in Fig. 6-27B, and an up-down binary counter is shown in Fig. 6-28. The latter two circuits may be expanded for any binary power: 16, 32, 64, etc.

The walking ring counters of Figs. 6-29A and 6-29B are synchronous, symmetric, and easily decoded (Fig. 6-29C). There are 8 disallowed states,

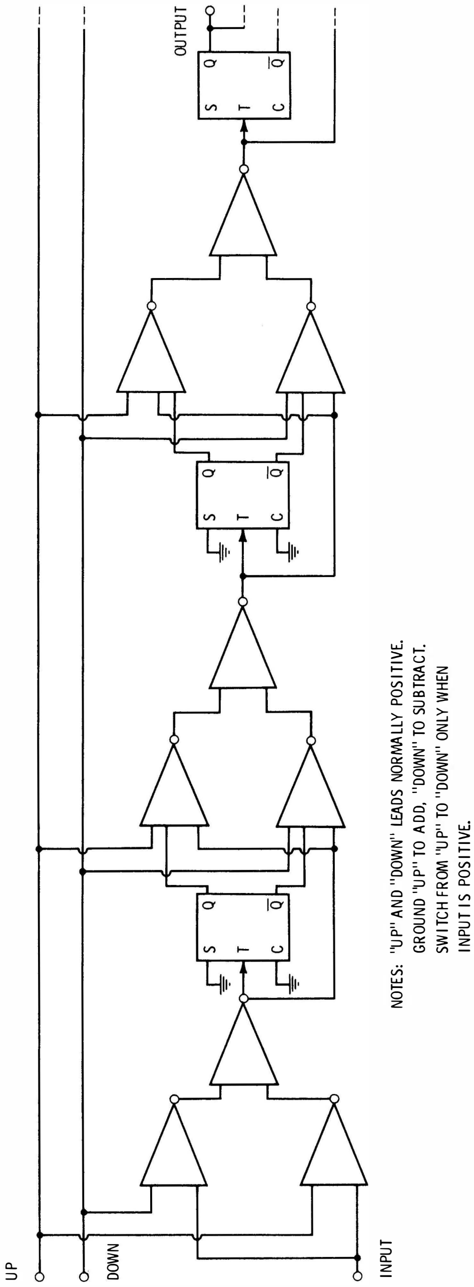


(A) Binary ripple counter.



(B) Synchronous binary.

Fig. 6-27. Divide-by-8 counters.

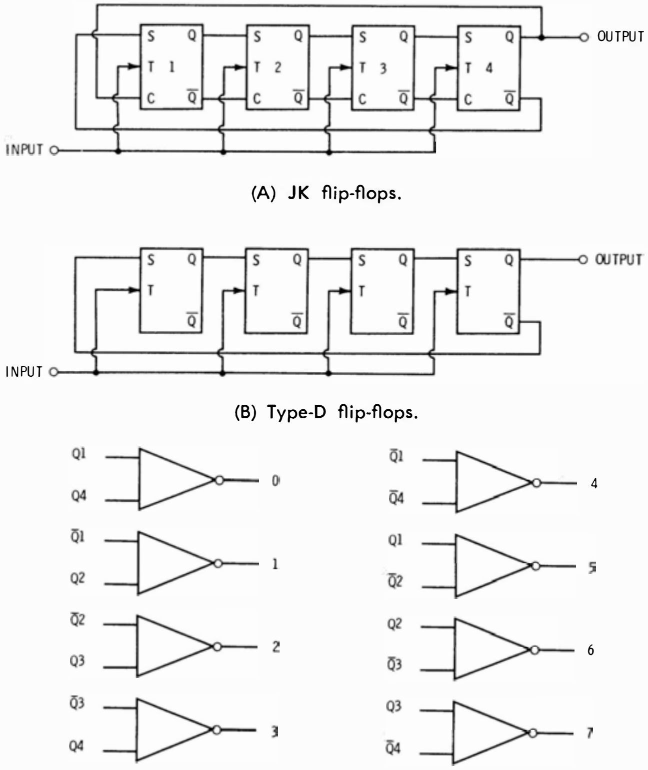


NOTES: ''UP'' AND ''DOWN'' LEADS NORMALLY POSITIVE.
GROUND ''UP'' TO ADD, ''DOWN'' TO SUBTRACT.
SWITCH FROM ''UP'' TO ''DOWN'' ONLY WHEN
INPUT IS POSITIVE.

Fig. 6-28. Binary add-subtract counter.

corrected by NANDing the outside two states and using this NAND coincidence to preclear the inside two flip-flops. Fan-in is high.

The self-decoding bucket brigade appears in Fig. 6-30. As usual, fan-in is high, and the counter is synchronous, self-decoding, and weighted. There are 248 disallowed states.



(A) JK flip-flops.

(B) Type-D flip-flops.

(C) Decoding gates.

Fig. 6-29. Divide-by-8 walking ring counters.

Divide by Nine

The function of the 3 × 3 circuit in Fig. 6-31 is rather obvious. The circuit is autoclearing, has a 2:1 output asymmetry, is hard to decode, and is not synchronous.

The odd-length walking ring (Fig. 6-32A) is fully synchronous and relatively easy to decode (Fig. 6-32B), but it has a disallowed 3-count subroutine and high fan-in. Output asymmetry is 5:4.

The nonsynchronous circuit of Fig. 6-33 has an 8:1 output asymmetry and has no disallowed states. Decoding is difficult, and the fan-in requirement is high (two toggle inputs).

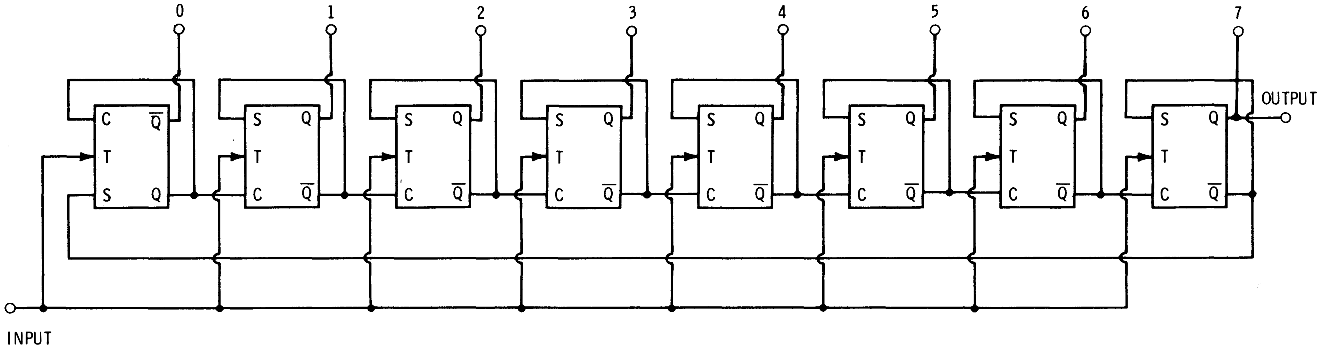


Fig. 6-30. Divide-by-8 bucket brigade.

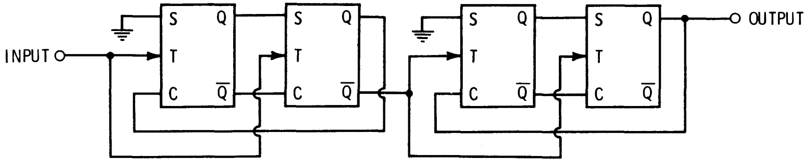
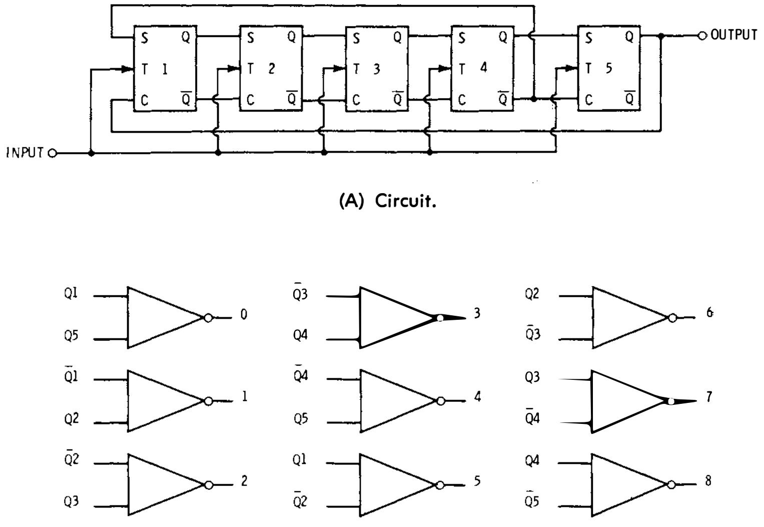


Fig. 6-31. Cascaded divide-by-3 counters.



(A) Circuit.

(B) Decoding.

Fig. 6-32. Divide-by-9 odd-length walking ring.

Fig. 6-34 shows the synchronous, self-decoding, weighted bucket brigade. The circuit is expensive, has a very high fan-in, and has 503 disallowed states.

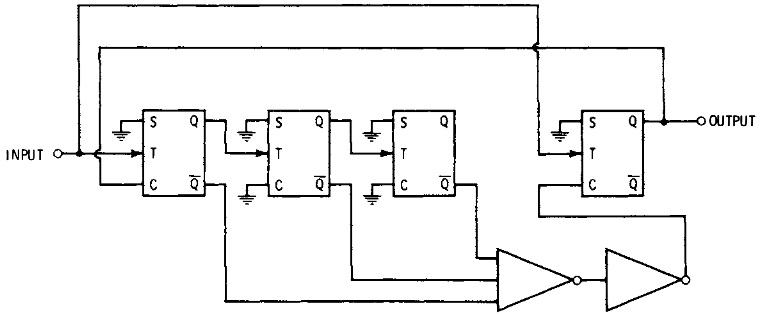


Fig. 6-33. Even-odd reduction (divide by 9).

HIGH-MODULO COUNTERS

Above modulo 10, we usually pick a counting *method* that provides the desired counter qualities, and then apply the method to the desired n. We already have a good stock of counter methods, but there are a few more that should be considered. The following is a look at all the approaches we have to the generalized ÷n counter.

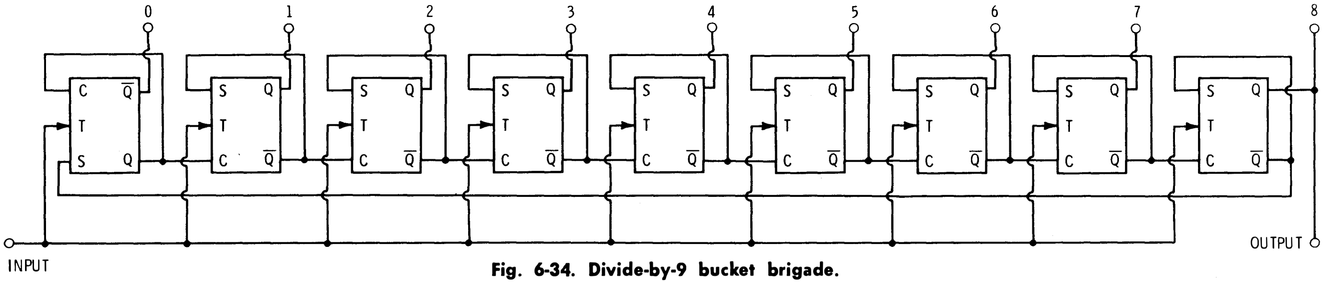


Fig. 6-34. Divide-by-9 bucket brigade.

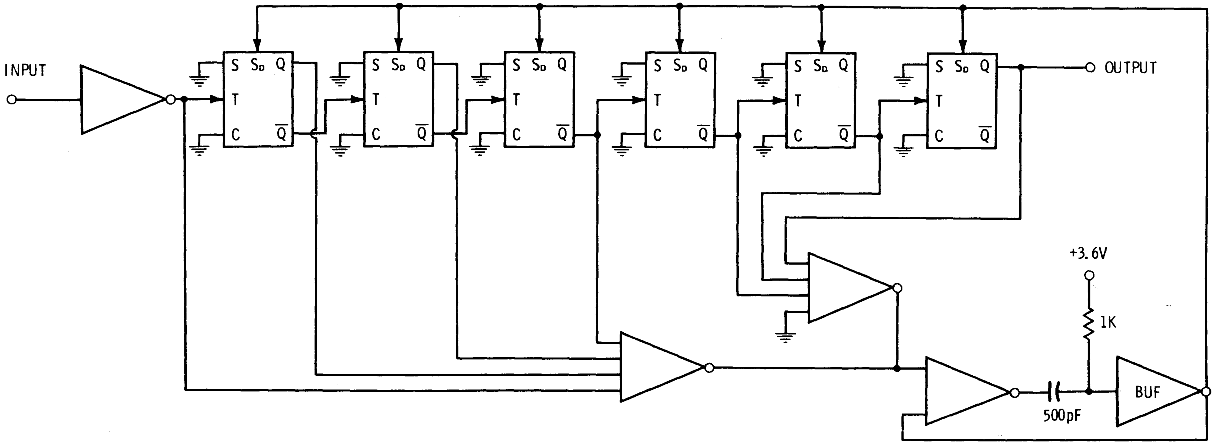


Fig. 6-35. Divide-by-35 counter using decode-and-reset technique.

Factoring

Factoring is an obvious counting technique, but is also a limited and relatively weak method. If n can be factored, a ÷n counter can be made by cascading counters corresponding to the factors. For instance, a ÷18 counter can be built by using a ÷6 and a ÷3 counter in cascade, a ÷9 and a ÷2 counter in cascade, or two ÷3 counters and one ÷2 counter in cascade. The technique never works with prime n, and always works with even n. We can always reduce an even n by putting a binary divider stage either ahead of or behind the chain. A binary divider at the input permits a lower fan-in and can retain weighting; a binary stage at the output guarantees output symmetry.

We retain all the qualities of the factor counters, except the output is no longer synchronous. Factored counters usually are hard to decode and usually are unweighted. They are rarely minimum. A ÷125 counter can be built with seven flip-flops, while three cascaded ÷5 counters require at least nine flip-flops. One specialized advantage of factored counters is the availability of intermediate frequencies. For instance, a factored ÷18 counter driven at 18 kHz could provide output frequencies of 9 kHz, 3 kHz, and 1 kHz simultaneously, while other, nonfactored ÷18 counters could not provide the 9-kHz and 3-kHz outputs.

Bucket Brigade

Bucket brigades are always weighted, synchronous, and self-decoding. It is easy to change the count length, simply by adding stages or by deriving a reset pulse at a desired maximum count. Counters of this type offer a straightforward PC layout. However, they can also be quite expensive, bulky, and power consuming for a large n, since one JK flip-flop per stage is needed. A ÷125 counter would require 125 JK flip-flops. Fan-in would be extremely high, as all toggle inputs are driven synchronously in parallel. There are also $2^n - n$ disallowed states, all but one of which are relatively easy to remove. The parts count for this circuit is very high. Typical bucket-brigade counters are shown in Figs. 6-15, 6-19, and 6-23.

Walking Ring

The walking-ring counting method was covered in Chapter 5. The walking ring counter is synchronous, gives a symmetric output, and may be decoded easily with two-input gates. It provides phase-shifted outputs if desired. It only works on even counts, and for a ÷n counter, n/2 flip-flops are needed. If there are n flip-flops, there are $2^n - 2n$ disallowed states, easily compensated for as in Fig. 5-10 by detecting the "outside" 1's and forcing the "inside" flip-flops into the 1 state.

For large counts, the walking ring counter is far from minimum, has a very high fan-in, and consumes considerable supply power. The Type-D walking ring counter is easily converted into an add-subtract counter (details of this conversion will appear in the next chapter). The count length is hard to change, particularly on JK walking rings. Typical walking ring counters are shown in Fig. 6-21.

Odd-Length Walking Ring

All of the advantages of the conventional walking ring carry over to the odd-length version. Here, one feedback path is started one stage early. Synchronous operation and easy decoding are retained, but the symmetric output is lost: the output symmetry is $(n+1)/2:(n-1)/2$. There are no disallowed states for rings of length 3, 5, 7, 11, and 13. Other rings have subroutines that are removed in the usual way. With prime-length walking rings, the subroutines will be the same length as the desired sequence for lengths greater than modulo 13. It takes $(n+1)/2$ flip-flops to build the counter. Fan-in is high, and the counter is far from minimum for large n. It is very difficult to change n. Typical odd-length walking ring counters are shown in Figs 6-16 and 6-24.

Decode and Reset

Decode and reset is another relatively obvious technique. A straight binary ripple counter longer than n is built. State n is decoded with a NAND gate, and the output of the NAND gate trips a monostable that resets the counter to state zero. Advantages of this technique are that the counter is easily resettable, binarily weighted, and nearly minimum. The disadvantages are ones of design—it is quite easy to develop glitches that reset the counter too early, a self-annihilating coincidence, or a permanent latching. It is extremely difficult to change count, and a multiple-input gate and full monostable, not required on other counter types, are needed. Decoding and PC layout both are difficult.

Fig. 6-35 shows a ÷35 counter using this technique. The first binary power greater than 35 is $2^6 = 64$. We build a six-stage binary-ripple up counter that would normally count to 64. To the counter, we add a seven-input NAND gate to detect the 100011 coincidence. To avoid glitches and premature reset during the settling time, we also add the *complement* of the input clock to make sure resetting can only occur in the *middle* of an input count, a time long after all settling glitches have disappeared. The NAND output is used to generate a reset pulse that is applied to the preset input of each stage. We can reset the counter to zero at any time by using an extra input on the monostable.

Encode and Count Down

There are several possible encode-and-count-down techniques. Basically, at the beginning of the count sequence we "write in" n in binary form on a binary-ripple down counter. We do this with the preset and preclear inputs. We then count the circuit down to zero and detect the 0000 . . . 0 coincidence. We can change n easily by changing the number to which we preset and preclear. The initial write-in pulse must be long enough to overcome any ripple carries and unwanted counts.

The encode-and-count-down circuit can be made as a one-shot counter, or the zero coincidence may be used to re-enter n again so that the counter cycles continuously.

The counter is weighted and nearly minimum, but it is not synchronous and is hard to decode. There are no disallowed states, and the counter autoclears. This type of counter usually is customized for any particular application. When designing a counter of this type, you must provide for unwanted carries, early glitch outputs, and self-annihilating coincidences in the counter design.

Encode and Count Up

To encode and count up, we build a binary-ripple up counter, and preset and preclear the *difference* between n and the maximum count sequence. When the counter overflows, we either stop the counting or generate a write-in pulse to re-enter the difference between n and the binary length. Design problems and available qualities are pretty much the same as for a decode-and-count-down counter.

Halfway Addition

The three previous methods are plagued with design problems that are nonexistent in the halfway-addition counter. A halfway-addition counter is also a binary ripple counter, but it takes its extra counts out of the *middle* of the count sequence. No multiple gates are required, glitches are inherently impossible, there is no possibility of latching, the counter autoclears, and the design is nearly minimum. It is usually the lowest-cost counter you can build for a particular n.

Halfway-addition counters are useful for large n, particularly where scaling is the only quality needed. The halfway-addition counter offers low fan-in, low cost, and an extremely easy PC layout. On the debit side, the counter is unweighted, asymmetric, and hard to decode. It is not synchronous. The count may be changed easily, by changing wire jumpers, programming slide switches, or using multiple selector switches.

The following are the rules for designing a halfway-addition counter:

1. Find the smallest number k for which 2^k is greater than n.
2. Build a binary ripple counter k stages long, arranged so that any stage can be precleared without producing a carry to the next stage.
3. Add to the output stage a half monostable that generates a positive pulse only on count $2^k/2$. This pulse is generated *halfway* through the untampered count length.
4. Find $2^k - n$ and convert it to binary form.
5. Connect the halfway-addition pulse from the half monostable to the required preclear inputs.

The application of these rules can be demonstrated through the design of a ÷77 counter. The first power of 2 greater than 77 is 128. Seven stages are required to build a ÷128 ripple counter. Each stage of the ripple counter is toggled from the $\overline{Q}$ output of the preceding stage. This arrangement makes it possible to preclear any stage at any time without producing a carry, since all a preclear input pulse can do is drive the $\overline{Q}$ output positive and never to ground. A half monostable is added to the final Q output, since this output drops abruptly on count k/2. The output of the half monostable gives a positive halfway-addition pulse.

Now, $128 - 77 = 51$. We have to add 51 counts halfway through the count sequence, or immediately after count 64. In binary form, 51 is $32 + 16 + 2 + 1$. To complete the ÷77 counter, we simply connect the half-monostable output to the stages that correspond to 32, 16, 2, and 1. The complete counter is shown in Fig. 6-36A.

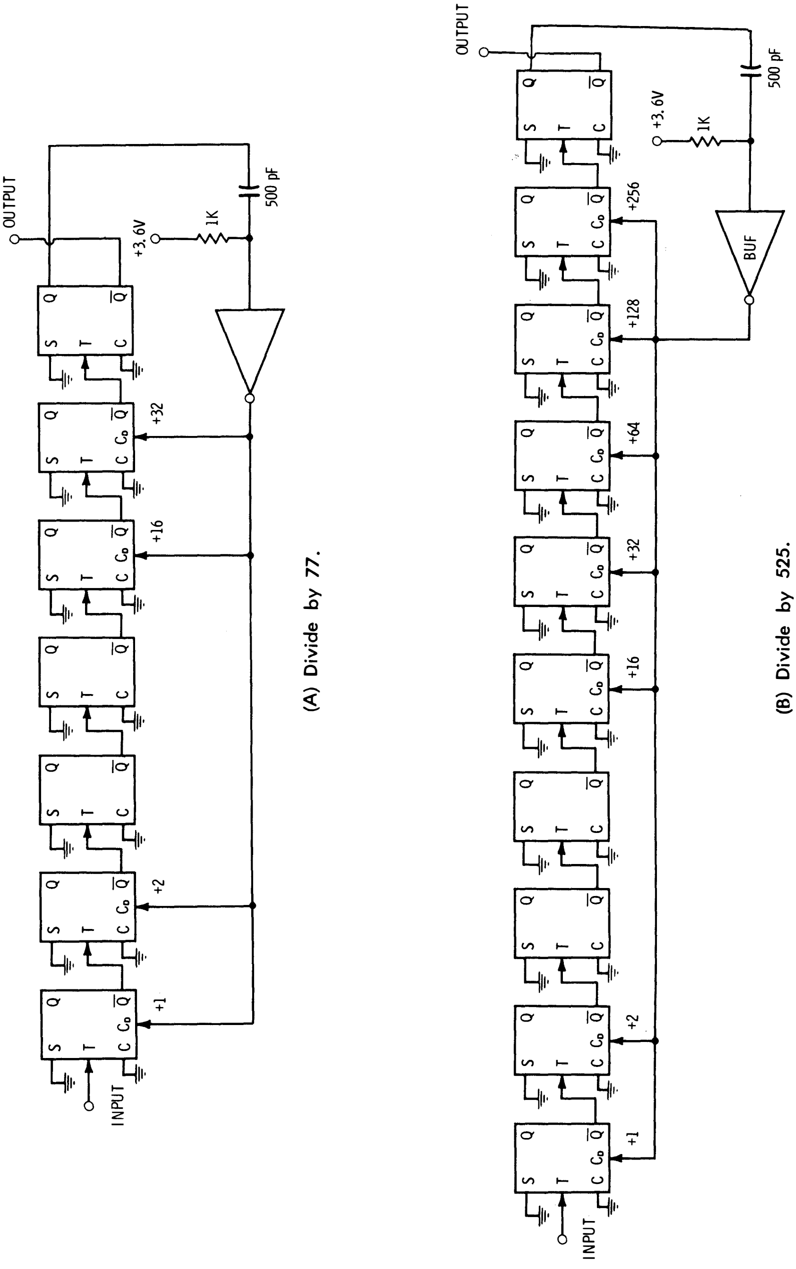
If the halfway-addition pulse is applied only to one or two stages, and if the input frequency is less than 100 kHz, a resistor-capacitor pulse network on the $\overline{Q}$ output can be used instead of the half monostable. Circuits of this type appear in Figs. 6-22B and 6-25B.

For long counts, sometimes more fan-out is needed for the preclearing operation than a half monostable can supply. If this is the case, a buffer half monostable may be used instead. Fig. 6-36B shows a ÷525 counter in which this technique is used.

The n of the counter is controlled through the choice of which preclear inputs are connected to add in the difference between the desired count and the untampered length of the counter. Fig. 6-37 shows a ÷n halfway-addition counter which may be programmed to divide from 9 through 16. The counter can be made to divide from 1 through 16 if provision is made for the self-annihilating coincidence that occurs if the last stage tries to add to itself. This correction requires a full monostable and is shown in Fig. 6-37B. The slide switches shown may be replaced with a four-deck selector switch. The resistors on the preclear inputs are necessary if any lead length is required; they help avoid problems caused by noise pickup, as pointed out in connection with Fig. 5-5.

The halfway-addition counter can be reset only if a preset input is available on each stage that has a preclear input that is used for the halfway-addition process.

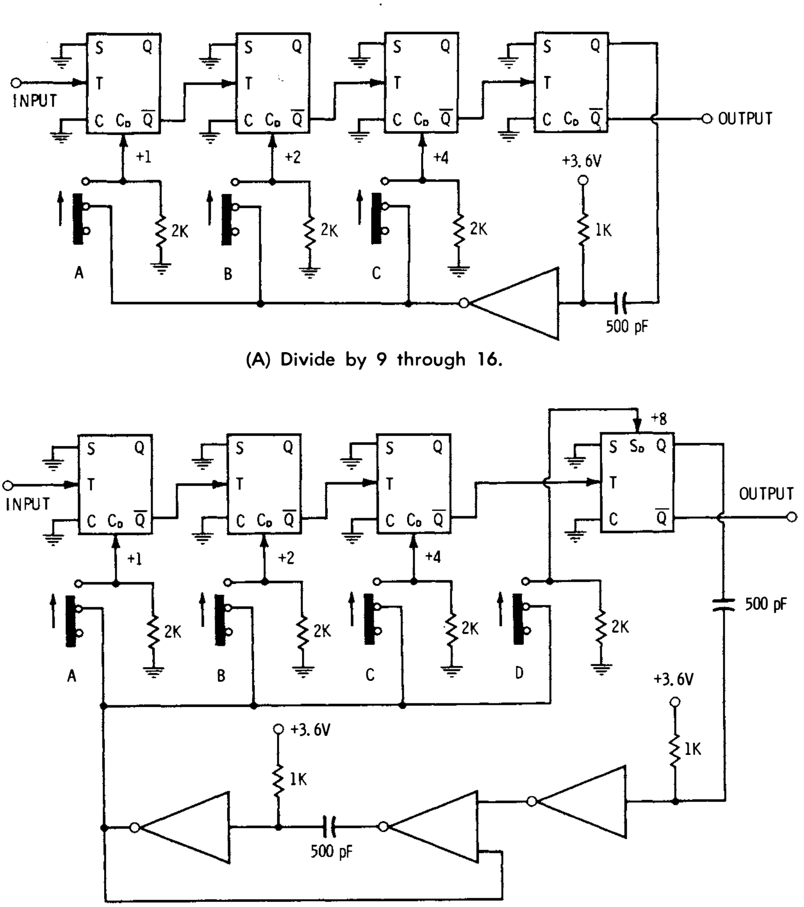
One other limitation of the halfway-addition counter is its relatively low speed. The counter is far slower than the wide-open capabilities of the JK flip-flops used, particularly on high counts. On count k/2, each flip-flop must sequentially toggle the next one, taking up 100 nanoseconds per stage. The final flip-flop must trip the half monostable and generate at least a 200-nanosecond addition pulse. The preclear inputs require another 100 nanoseconds to settle down and to do the preclearing. All of this activity has to happen before the next input pulse arrives. You can predict the maximum frequency of operation by multiplying the number of stages by



(A) Divide by 77.

(B) Divide by 525.

Fig. 6-36. Halfway-addition counter circuits.



(A) Divide by 9 through 16.

(B) Divide by 1 through 16.

Division Ratio	Circuit A Close Switches	Circuit B Close Switches	Division Ratio	Circuit A Close Switches	Circuit B Close Switches
1	———	A B C D	9	A B C	A B C
2	———	B C D	10	B C	B C
3	———	A C D	11	A C	A C
4	———	C D	12	C	C
5	———	A B D	13	A B	A B
6	———	B D	14	B	B
7	———	A D	15	A	A
8	———	D	16		

(C) Switch settings.

Fig. 6-37. Selectable-modulo halfway-addition circuits.

100 nanoseconds and adding 300 nanoseconds to the product; the result is the minimum *period* corresponding to the highest possible operating frequency. This estimate will usually be quite conservative, since the entire 100 nanoseconds of delay time per flip-flop is not used up before the next one starts toggling.

Phase-Shift Ring

A walking ring counter has available n negative-going transitions, spaced one count apart around the count sequence. We can select any desired negative-going transition and do any of several possible things with it. Counters using the phase-shifted negative-going output transitions are called *phase-shift rings.*

For example, we could reset our counter with a pulse from a monostable triggered by the selected transition; the result is a programmable ÷n counter whose count may be changed simply with a single-pole selector switch. The same counter may also be decoded. It is easily reset and it is synchronous. It may be set up to run continuously by using output-derived reset pulses, or a set-reset flip-flop or some similar command gate may be added to make it count out n and stop, one time only.

We also can use the negative-going transitions to advance a sequential pass-on circuit to build a programmable decade (or other n) divider, in which the units, tens, and hundreds are set separately on individual selectors. This arrangement makes possible a predetermining counter, a photo or interval timer that counts power-line cycles, or a scaler for a frequency synthesizer. For these applications, no decoding at all is needed, and a single-deck selector switch with very simple wiring does the entire job. Thus the phase-shift ring is the easiest counter on which to change n. The next chapter will show just how this technique may be applied to the design of practical circuits.

Other Counter Methods

Most of the reasonable and all of the simple counter methods have been covered, but there are many others. These other techniques are often subtle and frequently are plagued with design problems. Most of these problems have been mentioned already. They may be summarized as follows:

1. There can be no permanent disallowed states or unfixed disallowed subroutines.
2. Although a counter may cyclically go through n counts, there is no guarantee that the "output" will have one and only one positive pulse, or one and only one negative transition, spaced precisely n counts away from the initial 000 state.
3. The output of a coincidence must never be used to disturb the stages creating that coincidence. Also, we must never permanently latch a counter.

4. It is necessary to design around glitches and premature gate outputs caused by glitches.
5. We have to watch for all the qualities we want in a counter, particularly decodeability, ease of PC layout, and cost.

Most of the remaining counter techniques cannot meet these requirements directly, or they are relatively complex or of limited use. Each will be considered briefly.

By Guess—This term might be applied to any counter design in which the S and C inputs on various JK flip-flops are changed in tune with the states on the other flip-flops, to bring about the desired inhibiting, binary transition, or state transfer. Out of all this comes a cyclic ÷n counter. When the method works, you can often come up with a minimum and low-cost design. You literally keep playing around with the counter states until you find the combination that works.

Karnaugh Mapping—A *Karnaugh map* is a cyclic state diagram that turns the by-guess method into a somewhat predictable, but complex, design technique. The main problem with Karnaugh mapping is that stage and gate delays are not immediately obvious, sometimes causing serious glitch and premature-coincidence problems. Karnaugh mapping does point out possible subroutines and disallowed states reasonably well. Details on this technique are available in many good computer books.

Shift-Register Counters—There are many other possible shift-register counters besides the bucket brigade, the phase-shift counter, and the conventional and odd-length walking rings. A straight n-count shift register with a single 1 in it behaves as an n-point electromechanical stepper. The register can be closed on itself for continuous operation. Another shift-register possibility is to take the output of the register and use it to determine whether the first register stage is going to inhibit or binary divide, resulting in a cyclic sequence. Feedback from intermediate stages may be added also. Often, an output with a single negative-going transition may not exist, and decoding and PC layout can be quite complex. These counters are usually unweighted and usually have fixable disallowed subroutines.

Even-Odd Reduction—In even-odd reduction, we combine *even*-length counters with JK flip-flops designed to "add one" to the count sequence. For instance, a ÷37 counter is a ÷36 counter cascaded with an "add-one" circuit. The ÷36 counter is an even ÷4 counter cascaded with a ÷9 counter. The ÷9 counter is a ÷8 counter cascaded with an "add-one" circuit. Using this technique, we can build up any count length in a counter that is often nearly minimum. The technique is interesting, and sometimes the circuit can be built with no gates. However, counters of this type are unusually hard to lay out, and it is almost impossible to change the count sequence. The counter is normally self-clearing, and decoding is usually difficult. Two even-odd reduction counters for division by 5 and 9, respectively, are shown in Figs. 6-17 and 6-33.

FRACTIONAL-MODULO COUNTERS

There are several techniques for fractional n that are sometimes put to use. A fractional-n counter divides by some *ratio* of two whole numbers, providing frequencies related by, say 3/2, 17/19, or 3:4:5.

Least Common Multiple

In this method, we find the least common multiple of the desired frequencies and use an input clock frequency equal to this multiple. For instance, to generate 5, 4, 3, 2, and 1 kHz, we start with a 60-kHz clock. We divide by 12 to get 5 kHz, by 15 for 4 kHz, by 30 for 2 kHz, and by 60 for 1 kHz.

Sometimes the least common multiple is too high to be reasonable. If this is the case, it is often possible to *approximate* the desired frequency to within a tolerable accuracy and still do the job. For instance, on a musician's pitch reference, we might start with a 1.0716-MHz frequency. Division by 4096 results in a precise C_4 at 261.6 Hz. It is not possible to obtain A_4 at 440 Hz exactly, but a division by 2436 gives 439.9 + Hz, a frequency that is sufficiently accurate.

If you want two frequencies related by 49/35, pick a frequency that is 35 times frequency A. Divide by 35 to get A and by 49 to get B. This method gives the desired ratio exactly.

Multiply and Divide

To get two frequencies related by 3/2, 3/4, etc., use one or several frequency-doubler stages (Fig. 3-27B) and then divide by the required n. For instance, to get 500 Hz and 333 Hz, double the 500 Hz to 1 kHz, and divide by 3.

This technique is somewhat limited to certain fractional ratios, but it does work when a higher-frequency clock simply is not available. It also works only over a limited frequency range if symmetry and phase are important. This limitation occurs because the time constants in the multiplier become greater or less than 25 percent of the input period as the frequency is changed. Thus the technique is best used for single-frequency applications, or for operation over a relatively narrow frequency band.

Averaging

Sometimes we can alternately *change* the divisor and obtain an output that has an *average* number of pulses per second equal to the desired frequency. The output waveforms are never symmetric, and the output has a *variable* pulse rate but a frequency *average* over a long time equal to the desired frequency.

There are many possibilities with this technique. For instance, in the modulo-10 minimum-hardware circuit of Chapter 7, the first flip-flop divides by 2 and the last flip-flop divides by 2; the middle two must be divid-

FRACTIONAL-MODULO COUNTERS

There are several techniques for fractional n that are sometimes put to use. A fractional-n counter divides by some *ratio* of two whole numbers, providing frequencies related by, say 3/2, 17/19, or 3:4:5.

Least Common Multiple

In this method, we find the least common multiple of the desired frequencies and use an input clock frequency equal to this multiple. For instance, to generate 5, 4, 3, 2, and 1 kHz, we start with a 60-kHz clock. We divide by 12 to get 5 kHz, by 15 for 4 kHz, by 30 for 2 kHz, and by 60 for 1 kHz.

Sometimes the least common multiple is too high to be reasonable. If this is the case, it is often possible to *approximate* the desired frequency to within a tolerable accuracy and still do the job. For instance, on a musician's pitch reference, we might start with a 1.0716-MHz frequency. Division by 4096 results in a precise C_4 at 261.6 Hz. It is not possible to obtain A_4 at 440 Hz exactly, but a division by 2436 gives 439.9 + Hz, a frequency that is sufficiently accurate.

If you want two frequencies related by 49/35, pick a frequency that is 35 times frequency A. Divide by 35 to get A and by 49 to get B. This method gives the desired ratio exactly.

Multiply and Divide

To get two frequencies related by 3/2, 3/4, etc., use one or several frequency-doubler stages (Fig. 3-27B) and then divide by the required n. For instance, to get 500 Hz and 333 Hz, double the 500 Hz to 1 kHz, and divide by 3.

This technique is somewhat limited to certain fractional ratios, but it does work when a higher-frequency clock simply is not available. It also works only over a limited frequency range if symmetry and phase are important. This limitation occurs because the time constants in the multiplier become greater or less than 25 percent of the input period as the frequency is changed. Thus the technique is best used for single-frequency applications, or for operation over a relatively narrow frequency band.

Averaging

Sometimes we can alternately *change* the divisor and obtain an output that has an *average* number of pulses per second equal to the desired frequency. The output waveforms are never symmetric, and the output has a *variable* pulse rate but a frequency *average* over a long time equal to the desired frequency.

There are many possibilities with this technique. For instance, in the modulo-10 minimum-hardware circuit of Chapter 7, the first flip-flop divides by 2 and the last flip-flop divides by 2; the middle two must be dividing by 2-1/2, or 5/2. Actually, the two inside stages together divide by 2 and then by 3, by 2 and then by 3, and so on, such that the *average* number of their output counts is related to their input by 5/2 even though the *actual* 5/2 frequency is not really present. (There is a real ÷10 frequency at the output of the whole decimal counter.)

The averaging technique is useful whenever pulse rates are being used and we are interested only in getting so many counts over a relatively long period of time. In some VCO circuits, a local oscillator and an up-down counter sometimes can be added to smooth out the pulse-rate variations and get the actual frequency desired. The latter approach is often expensive and complex.

CHAPTER 7

Decimal Counting

Machines can count by two's, by eight's, or by any other radix, but people count by ten's. As a result, there are today more different applications for decimal, or decade, counters than for any other type of counter or scaler.

Decimal counters developed more slowly than other computer circuits because of the relative complexity and inefficiency of the base-10 number system. In fact, for many years, decimal techniques were avoided by the computer people. Today, this is no longer true, and there is a wide variety of practical, low-cost decimal techniques easily realizable with RTL IC's.

DECIMAL COUNTER APPLICATIONS

Where do we use decimal, or decade, counters? We have the same choices of counter qualities for the base ten as for any other n. Any of the high-modulo techniques of the last chapter can be used to build a decimal counter. The trouble is that decimal counters are rarely isolated deep inside an electronic system. Generally, a decimal counter has to do something besides count. It might be called on to provide a unique output waveform, drive a visual readout or a printing hammer, or stop on a preselected count. Before a decimal counter is built, it is necessary to determine what else the counter must do, over and above the normal counter qualities. We now turn to eight major applications of decimal counters to see just what the "something else's" are.

Scale Frequency or Time

In decimal scaling of frequency or time, the primary concern is getting an output frequency one-tenth of the input frequency, or a time reference ten times longer than the input period. For instance, we might like to use a premium IC decade divider to bring a 30-MHz signal down to 3 MHz where it may be measured or counted easily with RTL. Decade scaling is usually far less costly than the comparable task of direct high-frequency measurement, and it is far more versatile and easier to use than zero beating or frequency conversion of high-frequency signals.

Another obvious application would involve taking a 0.1-second time gate in an electronic counter and then multiplying it by 10 or 100 to get an equivalent 1- or 10-second time reference.

Usually, a decade scaler does not have to be resettable, synchronous, decodeable, or weighted. Sometimes, all that is needed is a single negative-going output transition for every ten negative-going input transitions. Other times, a symmetric output is required, particularly if it is to be filtered to get a sine wave, or if a specified time width combined with a precise 50-percent duty cycle is needed. In still other cases, a specific asymmetrical output is required. Most often, we need several scalers, and cost is an important consideration. Then, a counter such as the modulo-10, minimum-hardware circuit (Fig. 7-11) is ideal.

Count and Decode into Ten Output Lines

In this application, the requirement is for ten distinct, sequential outputs that might serve as an electronic stepper, drive a printing hammer, or power certain types of readout. Usually it is necessary to be able to reset this type of counter, and carry output is often needed.

There may be a demand for considerable power on the output lines, sometimes far more than is available directly from a flip-flop or a gate. Driver transistors or possibly SCR's may have to be used. A high current capability in the on state may be needed, or perhaps a high-voltage blocking ability in the off state, and sometimes both. Control of disallowed states usually is also a must to prevent more than one output from being powered at any one time. Walking ring counters are often an attractive solution to this problem.

Count and Drive a Visual Readout

For a visual readout, we need a resettable decimal counter with a carry output, but we also require the counter to show what count it is on by means of its own visual readout. This type of counter is often used in digital voltmeters, universal frequency counters, and other digital instruments. It represents the most popular type of decimal counter.

Usually it is not necessary to decode all ten states individually, particularly if the chosen readout can be suitably "bent" to fit the chosen counter. An effort is always made to minimize the total system complexity by picking a counter sequence and a decoding system that result in the lowest overall cost. Going from counter to ten lines to encoder to readout is often an unduly complex and expensive route.

Cost is usually a major restriction of this counter type, as the *total* count-decode-readout *system* must be considered, and, usually, three to seven of these systems are used simultaneously in a digital instrument.

A 1-2-2′-4 counter, biquinary decoder, and one of the three readouts shown in Figs. 7-17, 7-19, and 7-21 provide an attractive solution to this problem. This counter works equally well with meter, incandescent, *Nixie*, or 7-bar readouts.

Count, Decode, and Produce an Analog Output

Another type of counter gives an output voltage or current *quantized* into ten discrete and stable steps. Counters may be cascaded to produce 100 or 1000 discrete steps, with correspondingly tighter restrictions on the current summing and step tolerance.

This decimal counter is used to convert a digital count into a proportional analog voltage, to drive a meter readout, or to produce a stair-step output waveform. If an outside-world output is obtained, this counter may be called a *digital-to-analog converter*, or D/A for short.

For this application, use is made of a weighted counter that usually has to be resettable and produce a carry. While the 1-2-4-8, 1-2-4-5, and 1-1′-2-5 weighted counters will work, there is a slight advantage to using a 1-2-2′-4 counter.

Count and Stop on a Predetermined Count

A counter that stops on a predetermined count might be used to count bottle caps, time photographs, or control an industrial process. Usually, there are many cascaded counters, and each counter has an associated selector switch. The desired number is dialed on the selector switches, and when the counter reaches that number it stops counting and shuts down whatever it is controlling.

For a photo timer, the power-line waveform may be squared and divided in frequency by 6 to get 0.1-second timing pulses. The time interval may be selected with switches, and in this way exposure time may be set to any preselected value. Usually a readout is not needed.

A readout is needed if the device is used for controlling an industrial process such as making bottle caps. The desired number of bottle caps is "dialed in," and the counters stop when they reach the proper number. Should the bottle-cap machine stop before the counter does, the readout tells how many more caps are needed to fill the order.

Either application requires a resettable counter and some means to provide a coincidence or a sequential pass-on as the desired number is reached. Economy is usually a factor, and we would particularly like to get by with single-pole selector switches, because they are usually far cheaper and easier to wire than the multiple-pole variety. A carry output obviously is needed, at least on the lower-order decades.

When a readout is required, the walking-ring predetermining counter and sequential pass-on of Fig. 7-9 is a good choice. When a readout is not necessary, the phase-shift ring of Fig. 7-10 may be used to save several flip-flops.

Programmable Divide

A scaler may be built with a predetermining selector switch. A division ratio from 1 through 10 is selected by means of the switch, and the output frequency follows suit. These scalers are used in frequency synthesizers, certain types of timers, or anywhere else it is desired to control the ratio of two frequencies manually. Usually, weighting, decoding, reading out, or resetting is not of concern. The programmable-divide walking ring circuit of Fig. 7-8 is often the best choice for this particular problem, if division from 1 to 10 is of interest. For division from 1 to 100 or 1 to 1000, the phase-shift ring is best.

Add-Subtract Count

An add-subtract counter goes in either direction and might be used in a machine-tool positional control, adding machine, totalizer, or desk calculator. A readout and resettability usually are requirements. The counter has two inputs, a count input and add-subtract input. The add-subtract input determines whether the next series of counts will be added or subtracted. Input counts are then added to or removed from the previous tally in the counter as directed. It is of utmost importance that the tally in the counter does not change as a switch is made from an add command to a subtract command. Any form of add-subtract counter is expensive and complex, and should be avoided if at all possible.

When speed is not a factor, the simplest way out is to use the 9's complement add-subtract adapter that converts any up-only counter into an add-subtract one. Details of this method are given at the end of this chapter. When a high-speed "real" add-subtract counter is required, the synchronous add-subtract walking ring of Fig. 7-7 is often the cheapest way out.

Accumulate Arithmetic

The accumulate-arithmetic decimal counter is used in the heart of a digital computer or desk calculator, or in specialized military computers. The *sequence* of states the counter goes through must be specified exactly, so that accompanying logic circuits that perform addition, subtraction, multiplication, division, or square-root extraction may be added to make use of the specified sequence. This counter is almost always weighted and may have to be synchronous. Usually, there is need for the ability to write in a number using the direct-set and direct-clear inputs.

There are many choices of counter sequences that may be used, depending on the rest of the arithmetic logic. Four suitable counters that will be discussed in this chapter are the 1-2-4-8, the 1-2-4-5, the 1-2-2′-4, and the 1-1′-2-5 counters. Other arithmetic counters, such as the "excess 3" counter or the self-complementing counters like the 1-2-4-2′ counter, are too numerous and specialized to cover here.

SOME PRACTICAL DECIMAL COUNTERS

In designing a decimal counter, it is necessary to look at the total picture. Thus, proper design involves not just how to count by ten, but how to get the simplest, best-performing *system* possible, considering the counter, decoding, readout, output drive requirements, cost, and complexity, *plus* all the ordinary counter qualities talked about in the last chapter.

There are really two "families" of decade counters available, those that use shift-register techniques, and those that use binary-division techniques. Generally, the shift-register types require more parts and power, and they may need input buffers. For programmable division, making a predetermined count, and addition-subtraction counting, the shift-register counters are more economical to use than the binary types. Shift-register counters are also synchronous.

The binary types are often cheaper, have no disallowed states, are simpler in design, and consume less power. On the debit side, they are difficult to decode into ten lines, to use for addition-subtraction counting, making a predetermined count, or for programmable division.

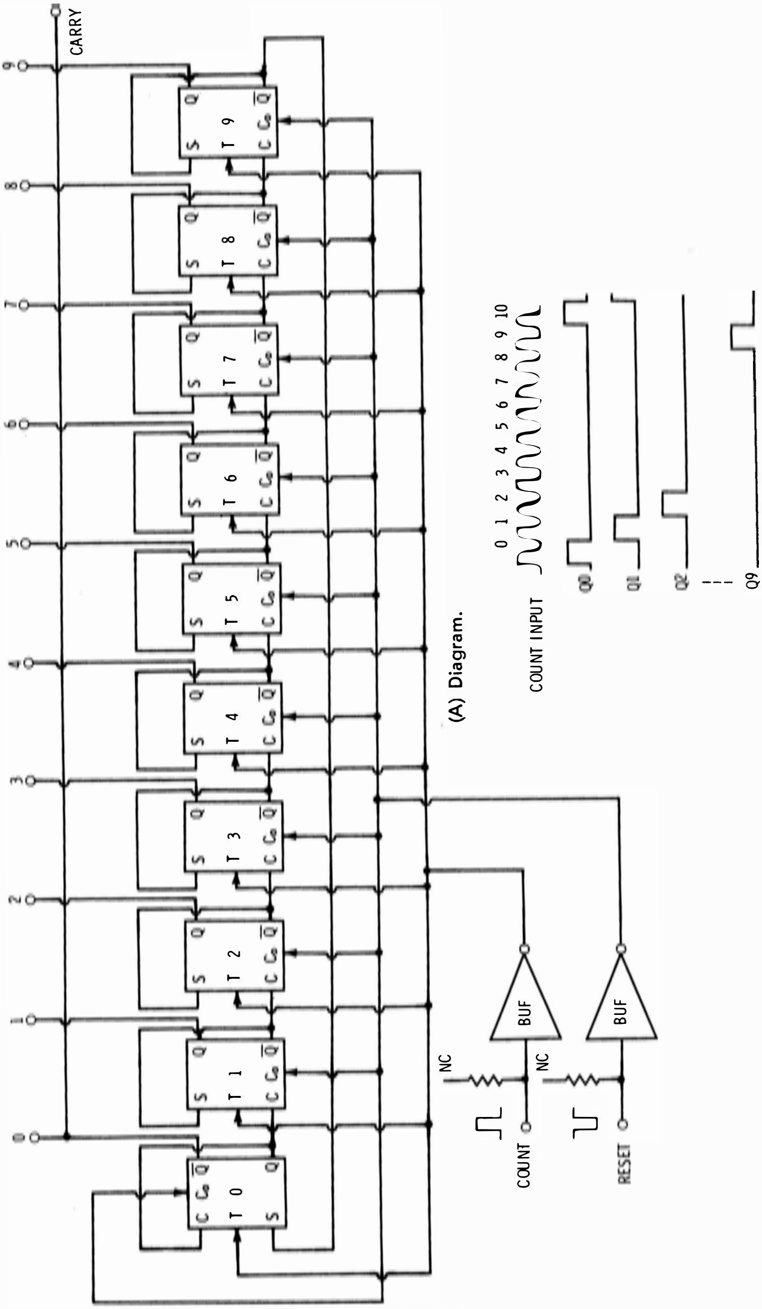
Ten-Count Bucket Brigade

As with any other bucket brigade, the counter in Fig. 7-1 is self-decoding, synchronous, and weighted. We automatically get out a one-in-ten decoding without any additional parts. All outputs are normally grounded (negative-logic 1). The energized output goes positive on the selected count. Because an inverting buffer is normally used to drive the ten synchronous toggle inputs, *positive* input pulses are normally used, and the counter shifts on the positive-going input transition. The carry output is asymmetric with a 10-percent duty cycle.

Some applications are shown in Figs. 7-2 through 7-4. In Fig. 7-2A, individual lamps are used as readouts. High-voltage transistors are used to drive neon indicators, and high-current, high-gain transistors are used to drive incandescent lamps. The base resistor (R) may be eliminated if buffered flip-flops such as the MC791 or MC726 are used, and if only a single transistor base is driven by each output.

In Fig. 7-2B, a *Nixie* tube or other gas-filled readout is used. Here, ten high-voltage transistors are needed. Instead of using ten base resistors, however, we can use a single common emitter resistor, saving nine parts. Fig. 7-2C shows how a 7-bar readout is driven. A logic translator converts the 1-in-10 code into a 7-bar code that lights the proper segments for each number. The translator can take the form of a diode array or a commercial decoding IC. If a commercial IC is used, make sure it accepts the 1-in-10 code.

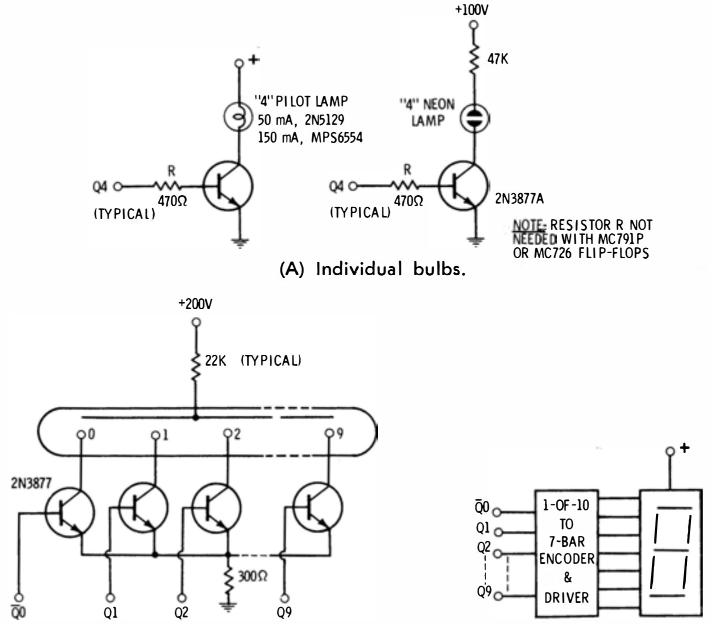
Fig. 7-3 shows a way to shorten or predetermine the count. The outputs of successive stages are NANDed together to predetermine a counter. Each stage produces a grounded output from its $\overline{Q}$ terminal (Q terminal for



(A) Diagram.

(B) Waveforms.

Fig. 7-1. Bucket-brigade decade counter.



(A) Individual bulbs.

(B) Nixie readout.

(C) 7-bar readout.

Fig. 7-2. Readouts for bucket-brigade counter.

state zero) on the selected count. When all decades have the proper outputs *simultaneously* grounded, the NAND output goes positive and produces a control output.

A programmable-divide counter is made by detecting the trailing edge of the maximum selected count, and triggering a half monostable which in turn resets the counter and produces an output pulse. Details are shown in Fig. 7-4. The Q outputs are selected, rather than the $\overline{Q}$ outputs as was the case in the predetermining counter.

With this circuit, a half monostable is all that is needed for counts ÷2 through ÷10. Bypassing the entire counter will provide a ÷1 action, if also needed.

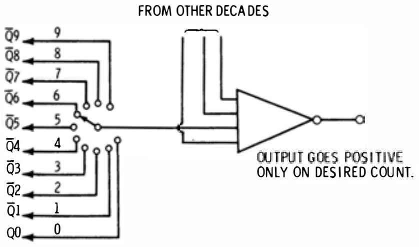


Fig. 7-3. Method of obtaining output on predetermined count.

The output of the monostable or half monostable becomes the carry for the next stage. It will be highly asymmetric, being positive for a few microseconds and grounded for the remainder of the time. Buffers are normally used to get sufficient fan-out.

The ten-count bucket brigade is not used often, since everything it can do can be done more easily with lower supply power by a walking ring counter. In the bucket brigade, there are 1014 disallowed states that are eliminated by resetting before every count sequence. The ten-count bucket brigade is directly equivalent to a mechanical 10-position stepper without any additional decoding.

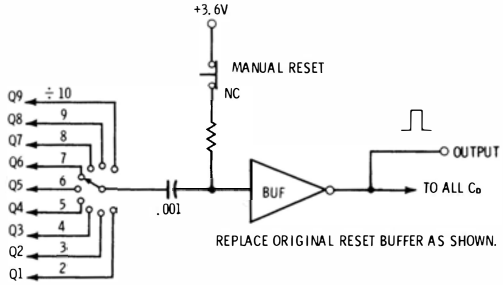


Fig. 7-4. Method of obtaining programmable-divide counter.

Walking-Ring Decade Counter

The synchronous divide-by-ten counter in Fig. 7-5 requires five flip-flops and a buffer, and produces a symmetrical output square wave of one-tenth the input frequency. Decoding requires 10 two-input gates, and the fan-out of each terminal is constant, so there are no loading problems. Either JK or Type-D flip-flops may be used. (This type of counter was covered in detail in Chapter 5.) Since a count-input buffer is normally used, the count usually changes on the positive-going input transition.

The gate connections for decoding are shown in Fig. 7-6A, and 7-6B shows how the "0" decoding can be used simultaneously to eliminate any disallowed subroutines and ease the reset fan-in. Without this circuit, there are 32 possible states, grouped into the legitimate ten, two disallowed ten-state routines, and a disallowed two-state routine. No additional parts are required to autoclear the counter when this technique is used if the "0" output is to be decoded.

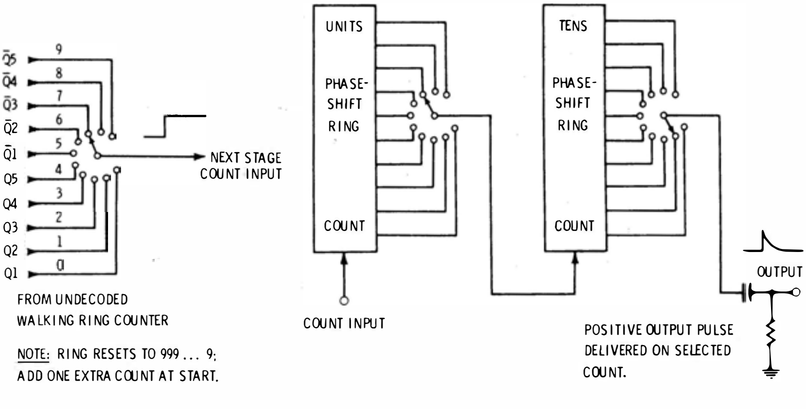
Add-Subtract Walking Ring

The walking ring counter can be converted into an add-subtract synchronous decimal counter, to produce one of the simpler and cheaper versions of a true up-down counter. With Type-D flip-flops (Fig. 7-7A), 16 extra gates are needed. With JK flip-flops (Fig. 7-7B) 21 extra gates are sequential pass-on advances to the last step and produces an output. We can decode and indicate all the intermediate steps if we like.

There is one problem that arises in this circuit when a zero count is selected. Possibly a transition to the sequential pass-on will not occur, or it may occur before it can be effective. To get around this, *all* the zero positions on the selector switches are routed to a signal that is *higher* in frequency than the input count signal. Now, if any zeros are missing, the high-frequency signal provides the needed transitions, and the circuit works on all counts. It even shuts itself down automatically should you set it to 000 and tell it to count.

Phase-Shift Ring

The phase-shift ring is one of the simplest programmable decimal counters. The technique requires only the basic walking ring counter and a single-pole, ten-position selector switch for each decade. The phase-shift ring makes use of the fact that there are ten positive-going output transitions, one for each count, that may be selected and routed as a carry to the next stage (Fig. 7-10A). In effect, we are producing a carry on a count other than ten. The first time around, the counter divides by the number it is supposed to be predetermining; after that, it decade divides.



(A) Basic connection.

(B) 0-100 divider.

Fig. 7-10. Phase-shift ring counter.

While this counter is not well known, it is ideal for all programmable scaling operations, such as digital photo timers, frequency synthesizers, or digital desk calculators. There are two limitations to the circuit: First, because the circuit counts in an obscure manner to get the final result, it is impossible to decode or indicate intermediate states. Secondly, there is a problem in obtaining the correct count; the problem normally is fixed by resetting the counter to 999 initially, and then adding a free count. This provides the zero transition that may be needed, just as a high-frequency signal was used to get the same effect with the predetermining walking ring counter.

Cascaded phase-shift ring stages divide by the *sum* of their individual settings and not the product of the divisors. For instance, ÷3, ÷9, and ÷1 scalers cascaded would divide by 27, while three cascaded phase-shift decade rings set to 3, 9, and 1 would divide by 391, making it possible to dial in a division ratio directly, even if the desired ratio is in the tens of thousands.

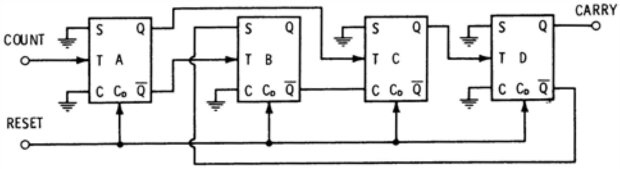
As an example of how the phase-shift counter works, suppose we build a two-decade version, cascading the second decade so that it is toggled by the *selected* positive transition of the first decade (Fig. 7-10B). Suppose we set the counter for 17. The sequence goes like this: Reset to 99. Add a free count; this gives a 90. (No carry is produced because the selector switch is set to carry on count No. 7, not on count No. 0.) Now we start counting: 90, 91, 92, 93, 94, 95, 96, 07. Here there was a carry because the selector switch routed the positive-going transition to the next stage. We continue counting: 08, 09, 00 (again no carry), 01, 02, 03, 04, 05, 06, 17. Once again, count No. 7 on the first counter gave a carry that jumped the second counter to state 1. The tens selector switch is positive on 1, and a positive-going output transition appears at this point. This transition is the output, and in spite of all the wild numbers in between, it took exactly 17 counts to get the output.

The extra free count at the beginning is easy to obtain. If this idea is bothersome, a few changes in the units decade get around the problem, but then the units decade becomes slightly different from the others. Another solution is to set the number in one low, or else tolerate the extra count needed to get an output. When the circuits of Fig. 7-10 are used with the basic walking ring counter of Fig. 7-5, the reset will *automatically* produce the 999 count. The extra pulse can often be gotten by applying an input start command through a half monostable, or it may be obtained by some similar means.

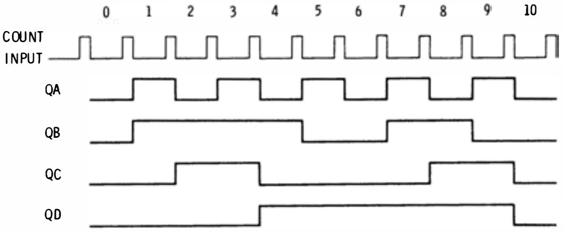
Modulo-Ten Minimum-Hardware Counter

The modulo-ten minimum-hardware counter (Fig. 7-11A) is the first of the binary-divider style of counters to be covered. It is the simplest possible decade counter, for it uses only four JK flip-flops and no gates. It gives an output waveform that has a 3:2 asymmetry and one negative-going transition for a carry on the tenth count. It is easily resettable, but is not synchronous.

This circuit is difficult to decode and is unweighted, so it is limited pretty much to scaling and decade dividing. The design is a "by-guess" one, in which the JK flip-flops are connected to inhibit each other on certain counts, as the waveforms show (Fig. 7-11B). The circuit autoclears, and there are no permanent disallowed states or subroutines.



(A) Diagram.



(B) Waveforms.

Fig. 7-11. Modulo-10 minimum-hardware divider.

Inverted Modulo-Ten Minimum-Hardware Counter

The modulo-ten minimum-hardware circuit can be "turned inside out" by putting the binary divider that was on the input on the output (Fig. 7-12). This arrangement provides a symmetrical output waveform, but it requires two complementary inputs. The two complementary inputs may be obtained by using the Q and $\overline{Q}$ outputs of a previous stage, or by adding an inverter to get the same effect.

A good application for the inverse circuit is as a time base for an electronic counter. One possibility is shown in Fig. 7-13. This circuit receives a 10-Hz input signal, either from the power line and a divide-by-six

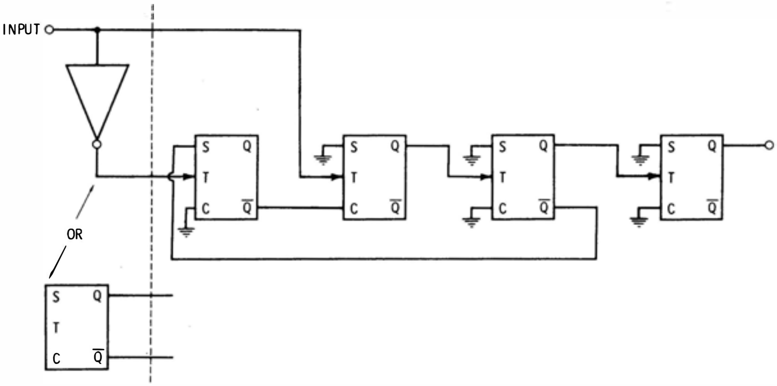
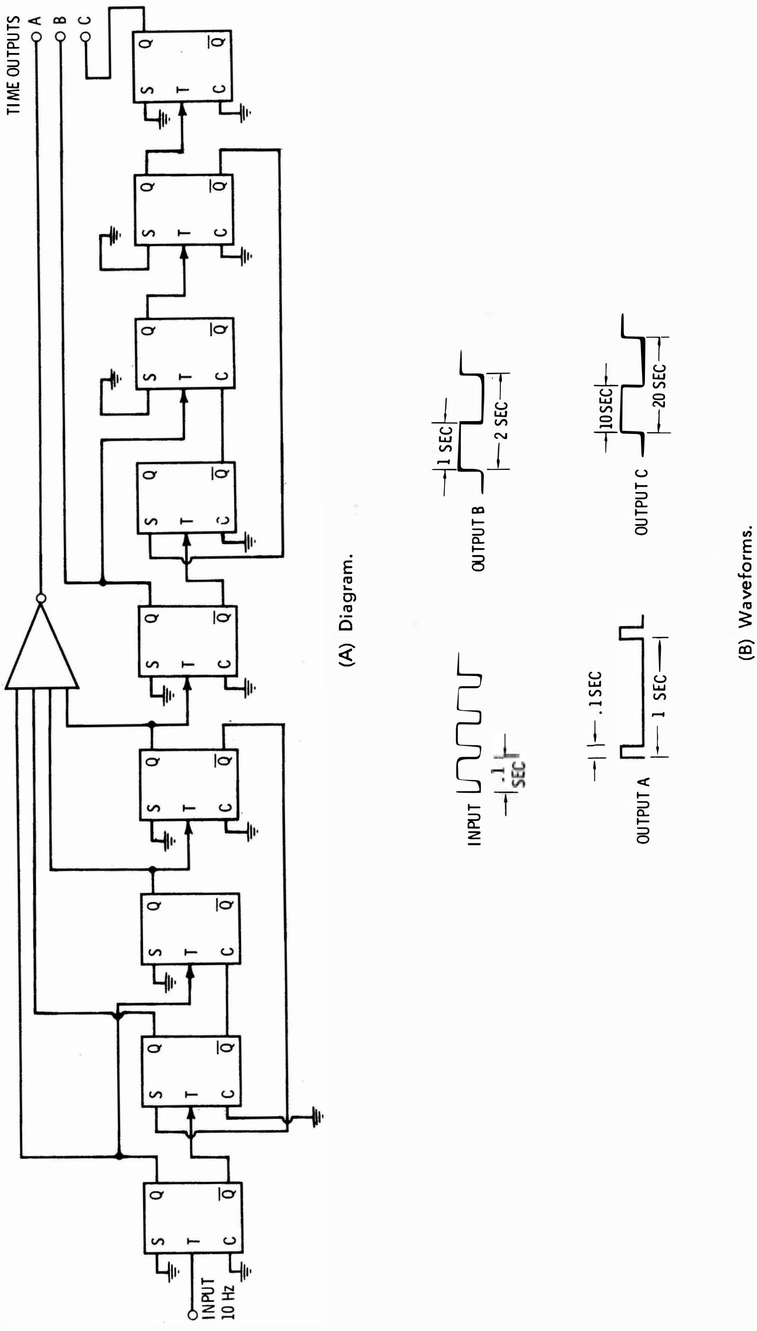


Fig. 7-12. Inverse modulo-10 minimum circuit.



(A) Diagram.

(B) Waveforms.

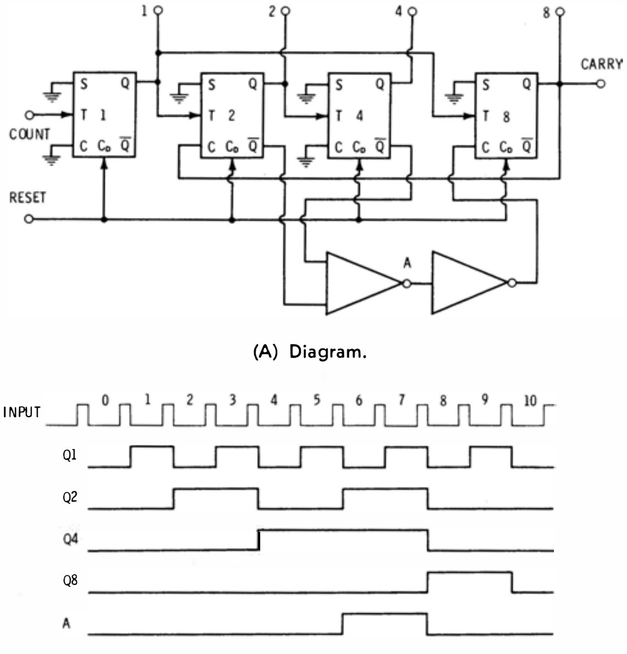
Fig. 7-13. Gate generator for electronic counter.

counter, or from a higher-frequency crystal standard and a divider chain. Two inverse modulo-ten minimum circuits, a divide-by-two flip-flop, and one four-input gate then give three output timing waveforms that will provide fully automatic electronic counter operation. The circuit provides a 0.1-second measure/0.9-second display; a 1-second measure/1-second display; and a 10-second measure/10-second display output.

1-2-4-8 BCD Decimal Counter

The 1-2-4-8 BCD decimal counter is the "classic" binary-divider type of decimal counter. Perhaps more problems have cropped up through poor versions of the 1-2-4-8 circuit than with any other single digital circuit. There are many ways to get the 1-2-4-8 weighting into a counter, most of them poor. The obvious decode-and-reset method has several disadvantages —it slows the counter operation, it interferes with external resetting, and it requires a full monostable to prevent any self-annihilating coincidences.

Fig. 7-14A shows a binary-division type of decimal counter that is weighted 1-2-4-8, is easily resettable, provides a carry output, and has no disallowed subroutines, races, or self-annihilating coincidences. It consists of four flip-flops, a two-input gate, and an inverter. Input fan-in is low, and the counter may be decoded into ten lines by use of the circuit of Fig.



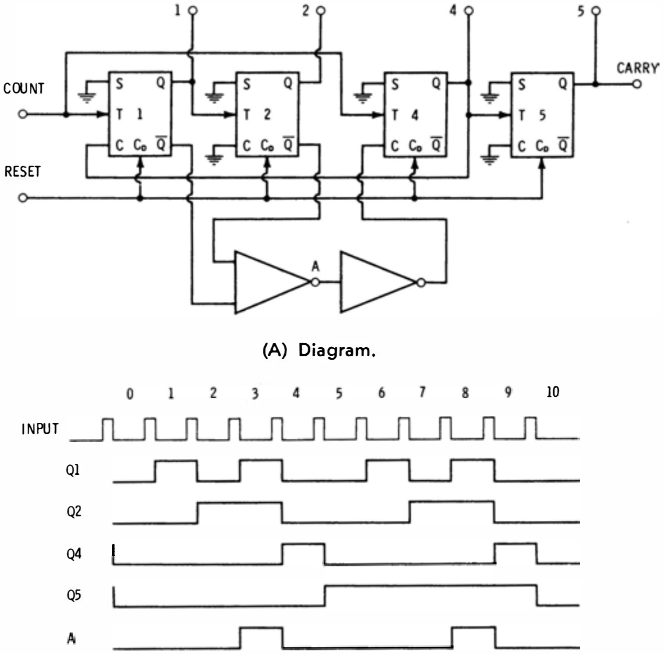
(A) Diagram.

(B) Waveforms.

Fig. 7-14. The 1-2-4-8 BCD counter.

2-15. Shortly, we will examine a better and simpler decoding method for readouts.

The counter is really a divide-by-two binary stage followed by a divide-by-five counter. The divide-by-five counter is a divide-by-four counter and an "add-one" circuit. The divide-by-four circuit operates normally for four input counts. The fourth count toggles the add-one circuit, which turns itself on and inhibits the divide-by-four counter for the next count and then turns iteslf back off. The result is divide-by-five action, with everything returning to the starting point on the fifth count.



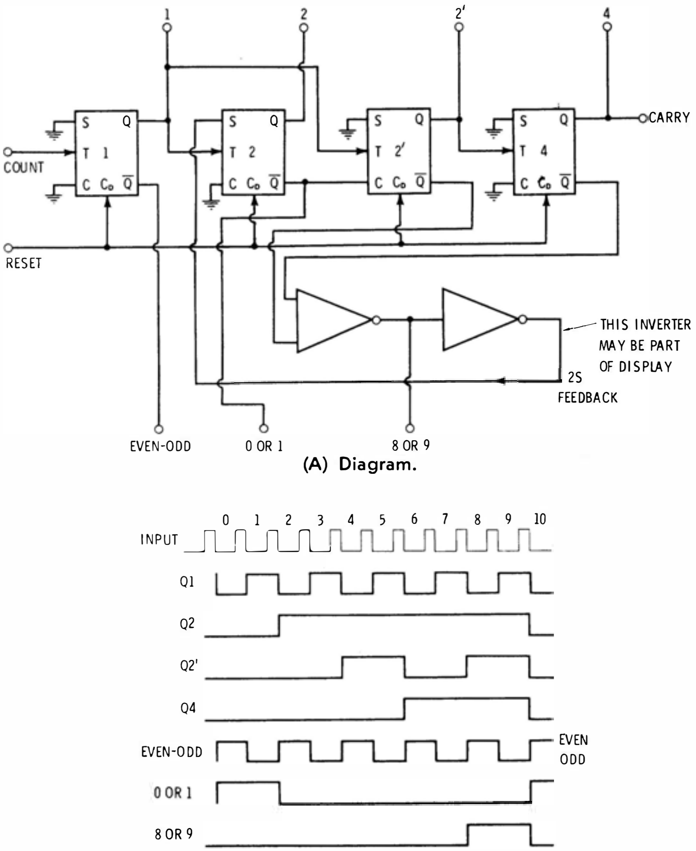
(A) Diagram.

(B) Waveforms.

Fig. 7-15. The 1-2-4-5 counter.

As the waveforms in Fig. 7-14B show, we get the 1-2-4-8 weighting, with positive logic available on the Q outputs and negative logic available on the $\overline{Q}$ outputs. The counter runs as fast as the first-stage flip-flop permits. For medium-power RTL, operation can be at least as fast as 4.0 MHz, and operation beyond 10 MHz usually can be achieved if you are very careful about supply voltages, filtering, and input-waveform shape and symmetry.

Single-package 1-2-4-8 decimal RTL counters are just becoming available at reasonable prices as this is written. Sometimes $\overline{Q}$ outputs are not available on the single-package counters.



(A) Diagram.

(B) Waveforms.

Fig. 7-16. The 1-2-2′-4 decimal counter.

1-2-4-5 Decade Counter

We can interchange the divide-by-two and the divide-by-five sections on the 1-2-4-8 counter, just as we can invert the modulo-ten minimum-hardware circuit. Now the divide-by-five portion goes around twice for every ten counts, and the output weighting is 1-2-4-5. The output is symmetrical, and the fan-in is doubled because two toggle inputs are being driven in parallel. This counter is shown in Fig. 7-15.

1-2-2′-4 Decade Counter

The 1-2-2′-4 decade counter (Fig. 7-16) is a slightly different weighted counter also built from four flip-flops, one gate, and one inverter. As with the other weighted binary-division decimal counters to be discussed, this one is resettable and has no disallowed states. It is not synchronous, and the output asymmetry is 3:2. The counter contains a divide-by-two section that divides the input *five* times around for ten counts in. This section is followed by a divide-by-five counter that cycles once around for each ten counts in.

This particular counter can be used easily to drive an analog or meter output, an incandescent display, a *Nixie* readout, or a 7-bar readout. By careful choice of decoding and driving techniques, we sometimes can get by with a single quad two-input gate for all the decoding. This method is vastly simpler than the decoding scheme shown in Fig. 2-15.

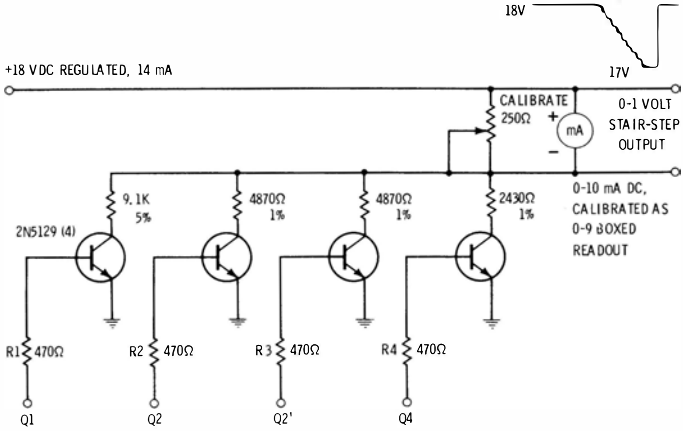


Fig. 7-17. Meter readout for 1-2-2′-4 counter.

The techniques discussed here also can be applied to the other binary-division weighted decimal counters, simply by changing a few connections. The techniques should also show how much more complex and undesirable it is to decode into ten lines before driving a readout. The methods to be covered are far simpler and cheaper.

For instance, to obtain an analog stair-step output or to drive a meter readout, we can use the circuit of Fig. 7-17, which requires no decoding at all. Each of the Q outputs of the counter drives a gating transistor through a current-limiting resistor. The collector current of each transistor is determined by the regulated supply voltage and the value of the collector resistor. These resistors are weighted in value to provide currents in the ratio of 1:2:2:4. These currents are summed, either in a meter or in a low-value output resistor, and the total current indicates the count. The supply voltage must be well regulated and must be high enough that any variation in drop across the meter as it changes counts, or any saturation change in the transistors, does not significantly affect the accuracy of the weighted currents that are being produced. While a vertical boxed meter readout is far from ideal, a decimal counter, decoder, and analog meter readout can be built quite economically.

Generally, we like to use a more "watchable" readout, such as an in-line incandescent display, a *Nixie* readout, or a 7-bar readout, since they produce their own light and are much easier to view than a meter pointer is. To drive these circuits, we need to decode the counter outputs. Usually, this is easiest to do by going first to a biquinary code from the original 1-2-2′-4 code of the counter.

A binquinary code has six lines. These lines are "weighted" in the following manner:

First line	Even or Odd
Second line	Zero or One
Third line	Two or Three
Fourth line	Four or Five
Fifth line	Six or Seven
Sixth line	Eight or Nine

Note that each output really does two things at once. We can get a biquinary output code much more simply than a one-in-ten decoding.

The numerals to be driven are grouped into even and odd groups. The "even" command turns on, or tries to turn on, all the even numerals. The "odd" command (the complement of the even one) tries to turn on all the odd numerals. The other five biquinary lines are only energized one at a time; each allows power to reach only one selected numeral, and the desired numeral lights. For instance, on count No. 7, the even-or-odd bus will try to light numerals 1, 3, 5, 7, and 9. The six-or-seven bus will try to light numerals 6 and 7. Only bulb No. 7 receives power through both commands, and only bulb 7 lights.

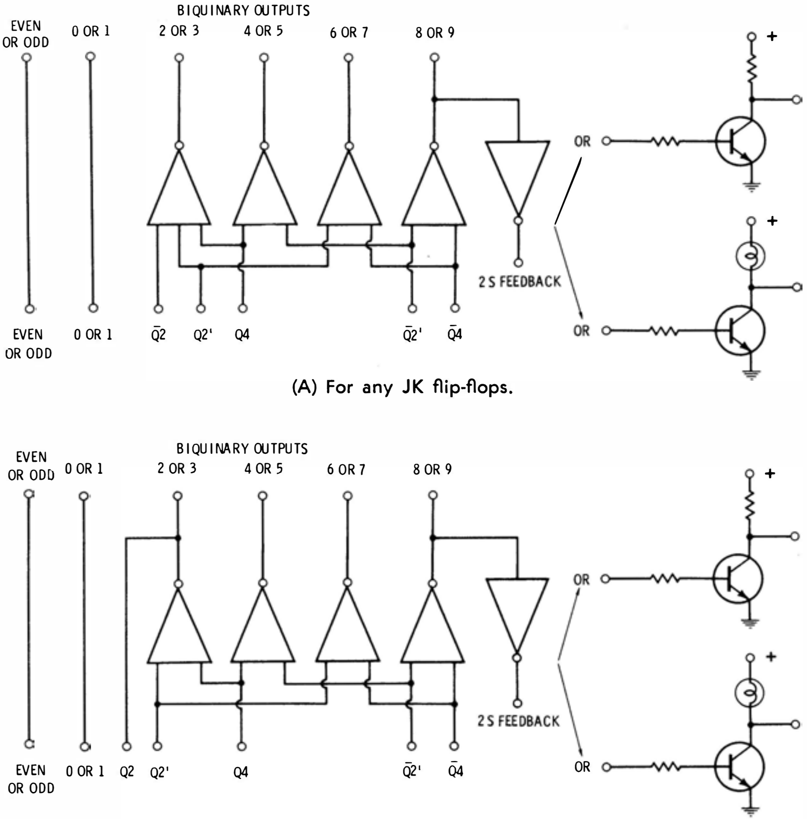
We can either control B+ with the even-odd command and control the grounding of selected numeral pairs, or we can put both commands in series at the ground end of the readout. It is necessary to watch for unexpected paths through series combinations of off numerals. These paths do not occur in *Nixie* readouts, and they are easily eliminated with diodes or extra transistors in an incandescent-lamp readout.

First, we have to get the biquinary coding. Fig. 7-16A shows that we already have an even-or-odd output, a zero-or-one output, and an eight-or-nine output as part of the feedback decoding. The rest of the states are obtained as shown in Fig. 7-18.

Three more gates are needed to get the other states. If regular JK flip-flops are used, one of the gates has to have three inputs, but, as Fig. 7-18B shows, if buffered flip-flops are used, a single quad two-input gate is sufficient to produce the entire biquinary decoding.

We do, of course, still need the inverter for the 2S feedback needed to run the counter. We can build this with a transistor and two resistors, or we can use part of the display itself "free," if we are careful. For instance, Fig. 7-19 shows an incandescent 0-9 staggered in-line readout driven by the 1-2-2′-4 counter and biquinary decoder. The three transistors at the left supply +6 volts to the even or odd power bus, depending on the even-or-odd output from the decoder. The five lower npn transistors drive pairs of bulbs. The pnp transistors below the bulbs act simply as dual diodes, isolating any undesired paths caused by series combinations of off bulbs. A total of 13 economy plastic transistors is needed for the display. Two resistors added to the 8-or-9 stage provide the inverted 2S feedback needed to run the counter.

Fig. 7-20 shows a commercial kit version of this counter-decoder-driver-readout. At the present state of the art, this unit costs less than a single plane or gas-filled readout, but somewhat more than a meter readout. Maximum counting speed is typically 10 MHz, and the circuit needs only about 750 milliwatts of total supply power. Since no high voltage is needed for the display, battery-powered instruments using this counter combination are practical.



(A) For any JK flip-flops.

(B) For buffered flip-flops.

Fig. 7-18. Biquinary encoders for 1-2-2′-4 counters.

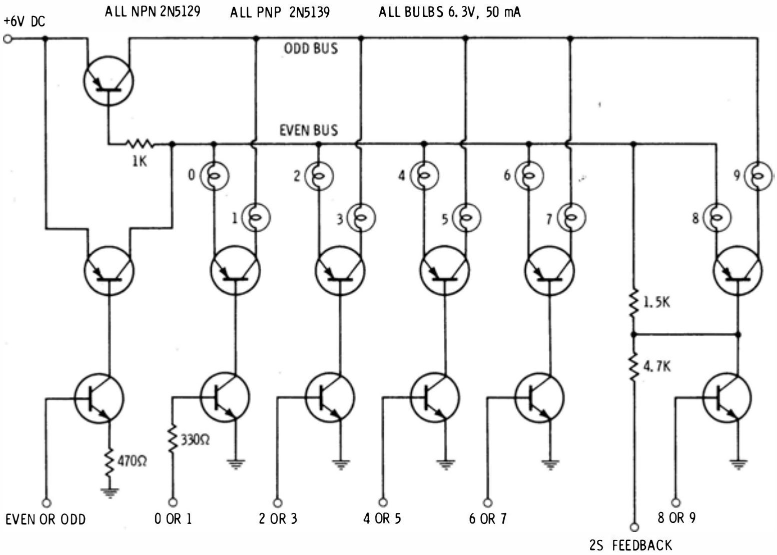
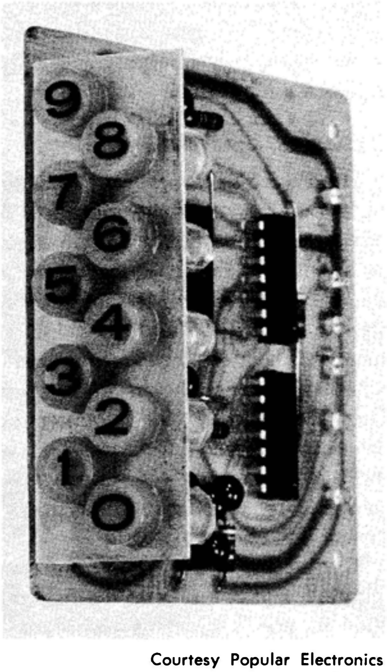


Fig. 7-19. Incandescent-lamp 0-9 display.

Fig. 7-20. Unit containing counter, decoder, and readout.



Courtesy Popular Electronics

When a single-numeral *Nixie*-type readout is mandatory, the more expensive circuit of Fig. 7-21 may be used. The ten transistors driving the indicator must be high-voltage ones. Because of the blocking effect of the off cathodes, no undesired paths exist. There is no way to get a "free" 2S inverter, so one must be added as shown. This circuit also requires 750 milliwatts of power, but only half of it has to come from the 200-volt display power supply. The *Nixie* may be blanked with an additional npn transistor between the even-odd transistors and ground.

At this writing, single-RTL-IC *Nixie* decoder/drivers are just becoming available at reasonable cost. These devices directly drive the indicator tube without external transistors.

Fig. 7-22A shows the additional decoding needed to drive a 7-bar readout, and Fig. 7-23A shows the decoding required for a 9-bar readout. (An 8-or-9 input is not needed; this state is sensed automatically when all the other number inputs are at their low level.) The 9-bar readout is slightly more complex, but it has the advantage of presenting more attractive and uniform 1's and 4's. Drive transistors are needed to match either readout type. A slight simplification is possible if feedback from the display is available.

1-1′-2-5 Decimal Counter

The 1-1′-2-5 counter is simply a 1-2-2′-4 counter "turned inside out" so that it consists of a divide-by-five section driving a divide-by-two section (Fig. 7-24). We gain a symmetrical output, but now need a higher fan-in at the count input because two toggle inputs are being driven in parallel in this configuration.

Choice of Counter Type

The choice of which weighted binary-divider type of decade counter to use depends upon the application. Obviously, if the weighting is important, as would be the case in an arithmetic accumulator, our choice is made for us. If low fan-in is important, the 1-2-4-8 and the 1-2-2′-4 counters would be considered. If symmetrical output is important, the choice narrows to the 1-2-4-5 and the 1-1′-2-5 counters. If we are driving an analog output or a meter, we might like to prevent pegging the meter if the counter momentarily gets in the wrong state when power is first applied. The 1-2-2′-4 and 1-1′-2-5 counters can only give a maximum of a weighted "9" output, never any more. The 1-2-4-5 counter can give a "12," while the 1-2-4-8 is worse yet—it can give a "15." Any of the four counters can be made to encode into biquinary form, but the required connections change for each weighting.

The 1-2-4-8 and 1-2-2′-4 counters may be synchronously gated by obtaining the S and C inputs to the first stage from an external gate input, as was done in Fig. 5-13. This technique will not work with the 1-1′-2-5 and 1-2-4-5 counters.

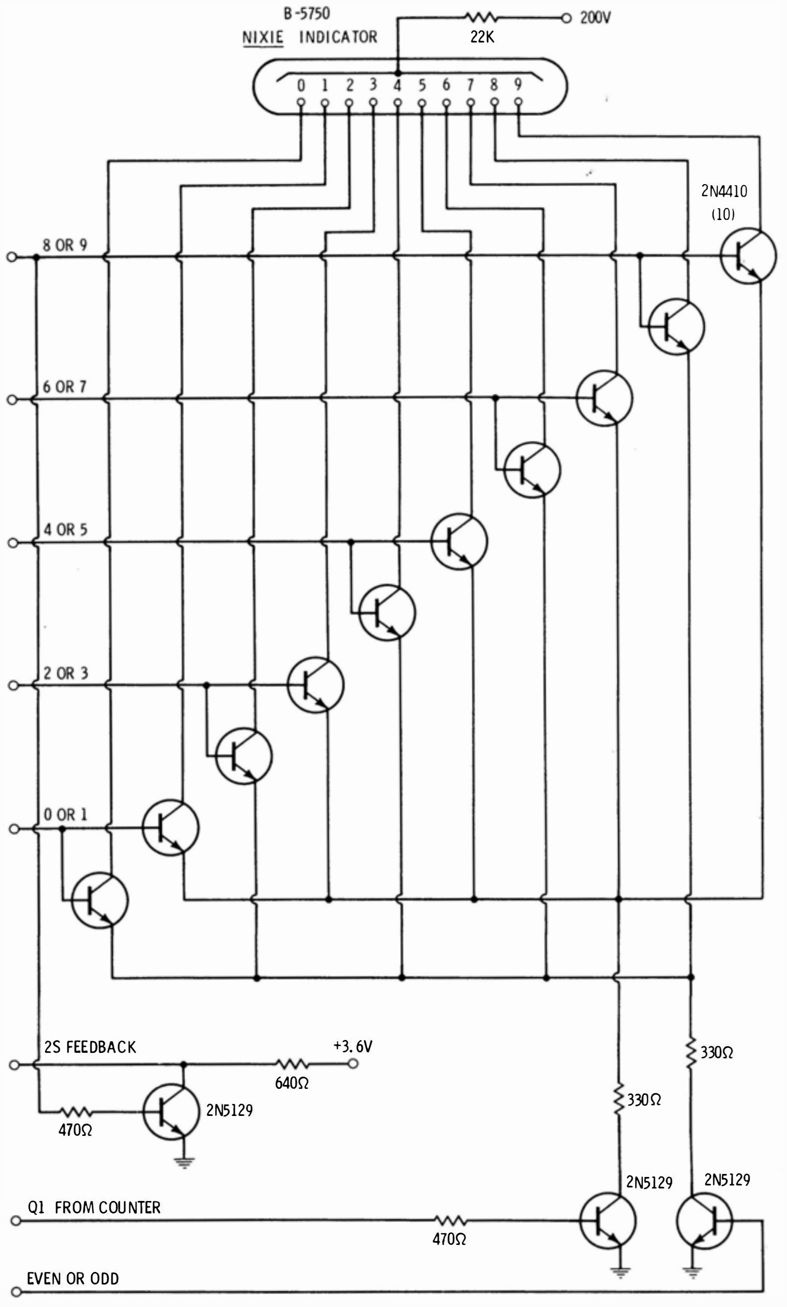
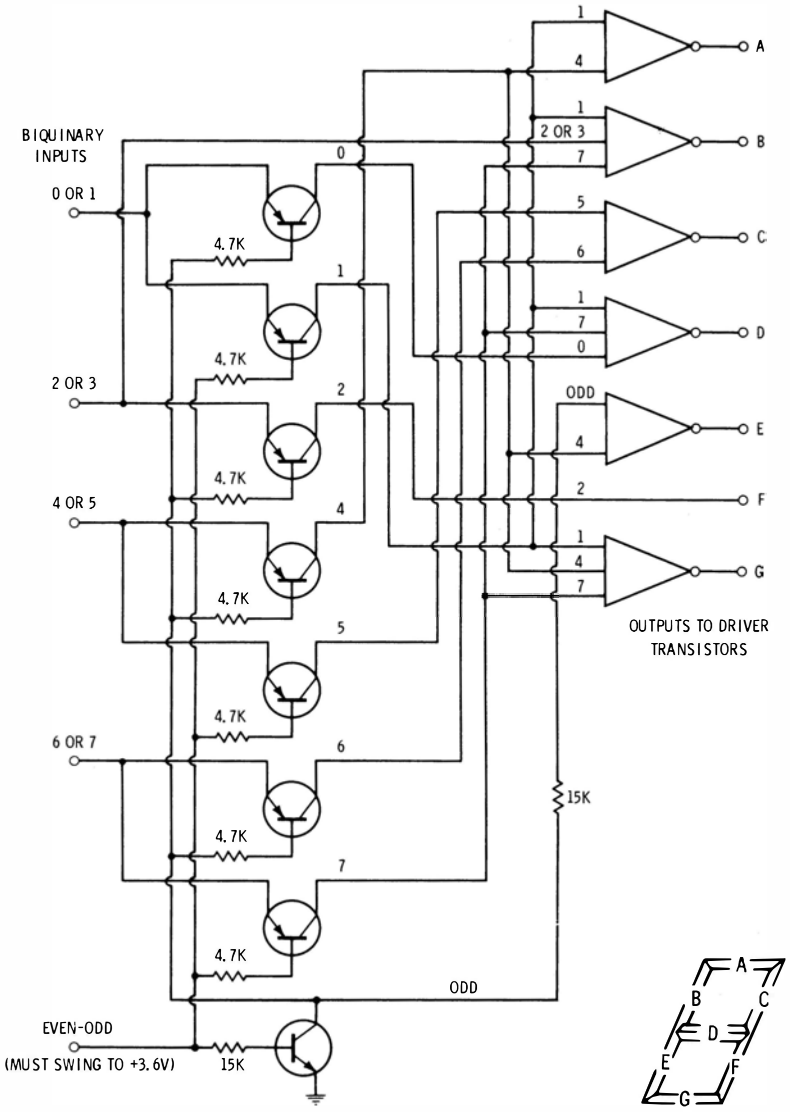


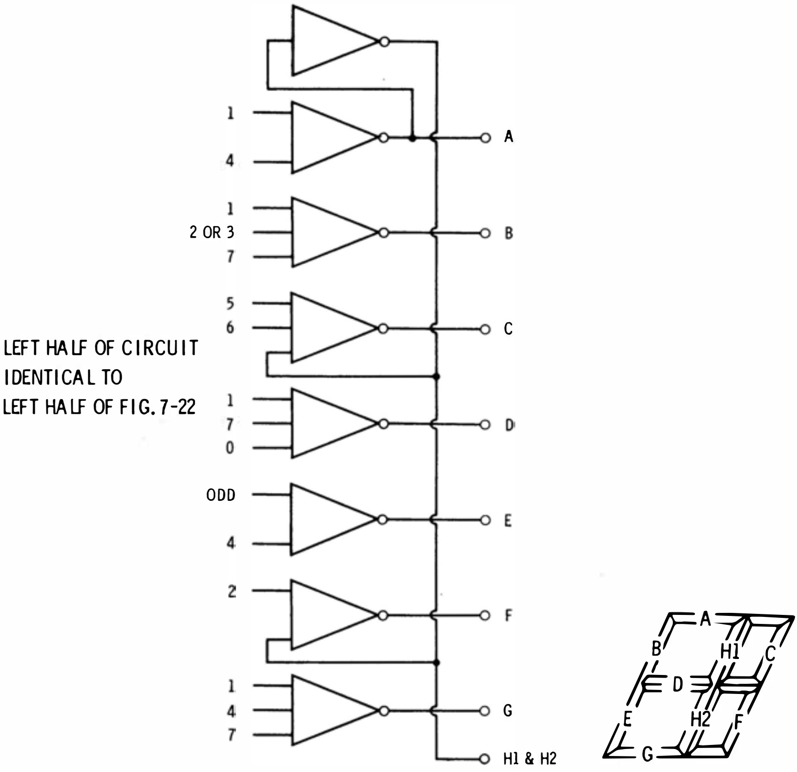
Fig. 7-21. Nixie indicator in 0-9 display.



(A) Circuit.

(B) Readout.

Fig. 7-22. Decoding for 7-bar readout and biquinary input.



(A) Circuit.

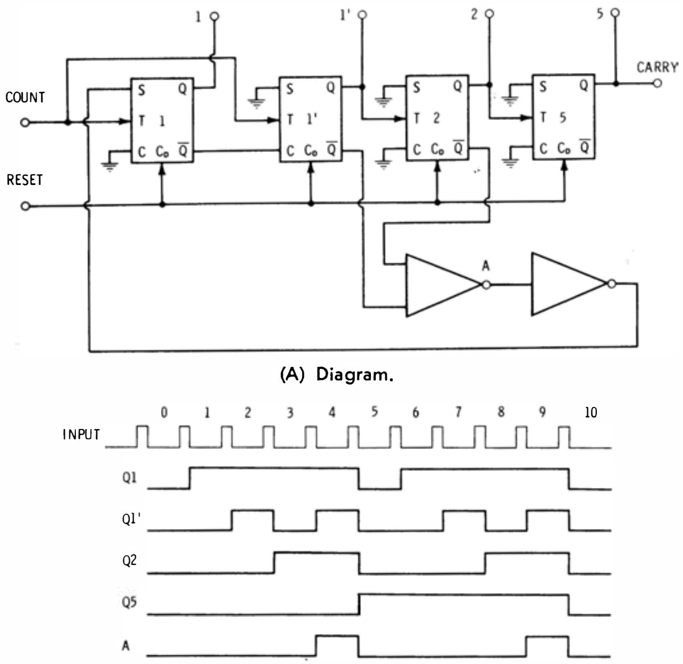
(B) Readout.

Fig. 7-23. Modification of Fig. 7-22 for 9-bar readout.

A 9'S COMPLEMENT UP-DOWN COUNTER

"True" add-subtract, or up-down, decimal counters are quite complex; Fig. 7-7 actually shows one of the simpler circuits. The complexity is caused by the need to switch from "add" to "subtract" without changing the count or decoding on the counter, and without producing a carry or a negative-going toggle transition to any of the flip-flops used. We also have to be able to distinguish a carry from a borrow output. While a fully integrated, single IC counter is the obvious solution to this problem, such counters are still too costly for many applications. We also compromise on decodeability when we build a true up-down counter, and sometimes we cannot use the low-cost biquinary decoding, or the particular type of weighting we want.

Because of the complexity of the add-subtract problem, there is a tendency to look for some way of getting the same apparent results without the cost and complexity of a real up-down counter. The desired results may be obtained by using a *nines-complement add-subtract adapter.* While this



(A) Diagram.

(B) Waveforms.

Fig. 7-24. The 1-1′-2-5 decimal counter.

adapter is a relatively complex device, it will apparently convert any reasonable up-only decade counter into an up-down counter. Since the adapter uses only the count and carry terminals of the up-only counter, any desired normal decoding or readout may be used.

The price paid for this convenience is speed. The adapter slows things down considerably; the particular adapter to be discussed here runs at only a 205-kHz maximum count rate. While it needs no flip-flops, the circuit does require five IC's, some resistors, and some capacitors. Nevertheless, this approach presently is far cheaper, simpler, and easier to use than any true up-down counter would be.

To convert an up-only decimal counter into an add-subtract, or up-down, decimal counter:

1. Provide an input circuit that gives *nine* output pulses for every *subtract* input pulse, and *one* output pulse for every *add* input pulse. Route the output of this circuit to the count input on the decimal counter.
2. When *adding*, route the *carry* pulses to the next stage in the normal manner.

3. When *subtracting*, route a *borrow* pulse to the next stage only when a carry is *not* produced after an input pulse.

Maximum count speed using this method is determined by how fast the nine output pulses can be generated and how fast a borrow signal can be recognized and produced.

One adapter and one up-only counter are used for each decade. A typical adapter circuit is shown in Fig. 7-25. Fig. 7-26 shows the waveforms, and Fig. 7-27 shows how to interconnect the adapters and the up-only counters in a complete add-subtract system.

The circuit operates as follows. Any input pulse generates a single, controlled-width pulse in the master monostable. If we are subtracting, the master-monostable output is doubled in frequency three times, to produce eight output pulses. Meanwhile, the master monostable also drives an add-pulse generator which generates one output pulse. The one output pulse is spaced precisely so that it fits in an empty time slot between counts 4 and 5 of the eight pulses produced by the frequency octupler. The eight pulses and the single pulse are combined in the add/subtract combiner, which gives the needed 9 counts out to the counter in response to a single input pulse.

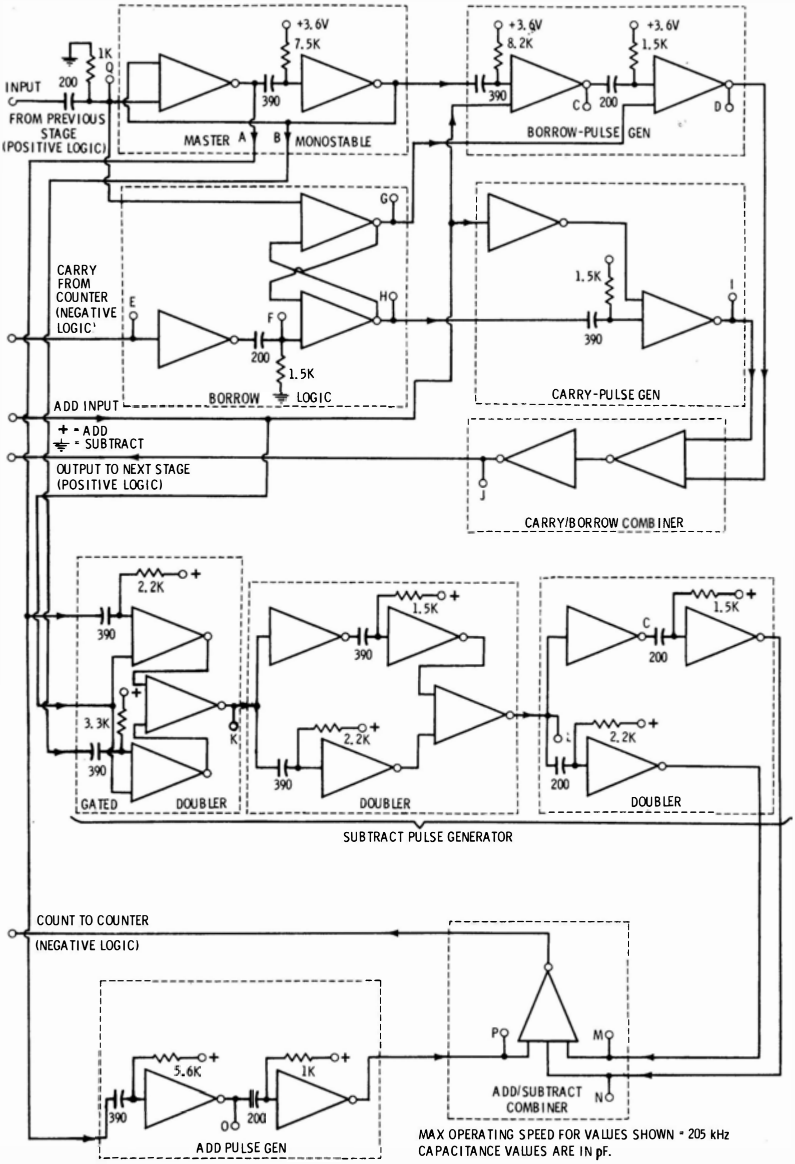
The nine pulses rapidly run the counter ahead nine counts. When this is done, the counter *appears* to have backed up one count, and *appears* to be subtracting. It goes forward so rapidly that the counting action is not apparent, and the counter appears to subtract instantaneously.

When *adding*, we place a positive add command into the adapter. This command *inhibits* the first doubler in the octupler, and prevents the eight pulses from being produced. The single pulse from the add-pulse generator gives only a single count out of the add/subtract combiner in response to a single input pulse. Thus *subtract* (a grounded add input) gives nine count pulses for every input pulse, and *add* (a positive add input) gives one count pulse for every input pulse.

Half the problem has been solved. The other half of the circuit provides the carry and borrow logic. To borrow, we set a borrow-logic flip-flop as an input pulse arrives. *If* a carry is produced on or immediately after the nine generated count pulses, the flip-flop is reset and nothing happens. If the carry is *not* produced, the control flip-flop lets the borrow-pulse generator produce a borrow pulse that is routed to the output. The borrow-pulse generator operates long enough after the ninth count that all counter delays in carrying are picked up. It operates only during subtraction, and only then when a carry from the counter is not produced on the nine pulses following an input count.

To carry, we automatically generate an output pulse *every* time a carry output is produced by the counter. This is done with the carry-pulse generator. A carry pulse is generated only during addition, and only if a carry pulse comes in from the decimal counter.

The carry and borrow pulses cannot occur at the same time. A borrow occurs only during subtraction; a carry occurs only during addition. These two separate possible pulses are combined in the carry-borrow combiner, producing the output to the next stage. The next stage behaves in the same



MAX OPERATING SPEED FOR VALUES SHOWN = 205 kHz
CAPACITANCE VALUES ARE IN pF.

Fig. 7-25. Diagram of add-subtract adapter.

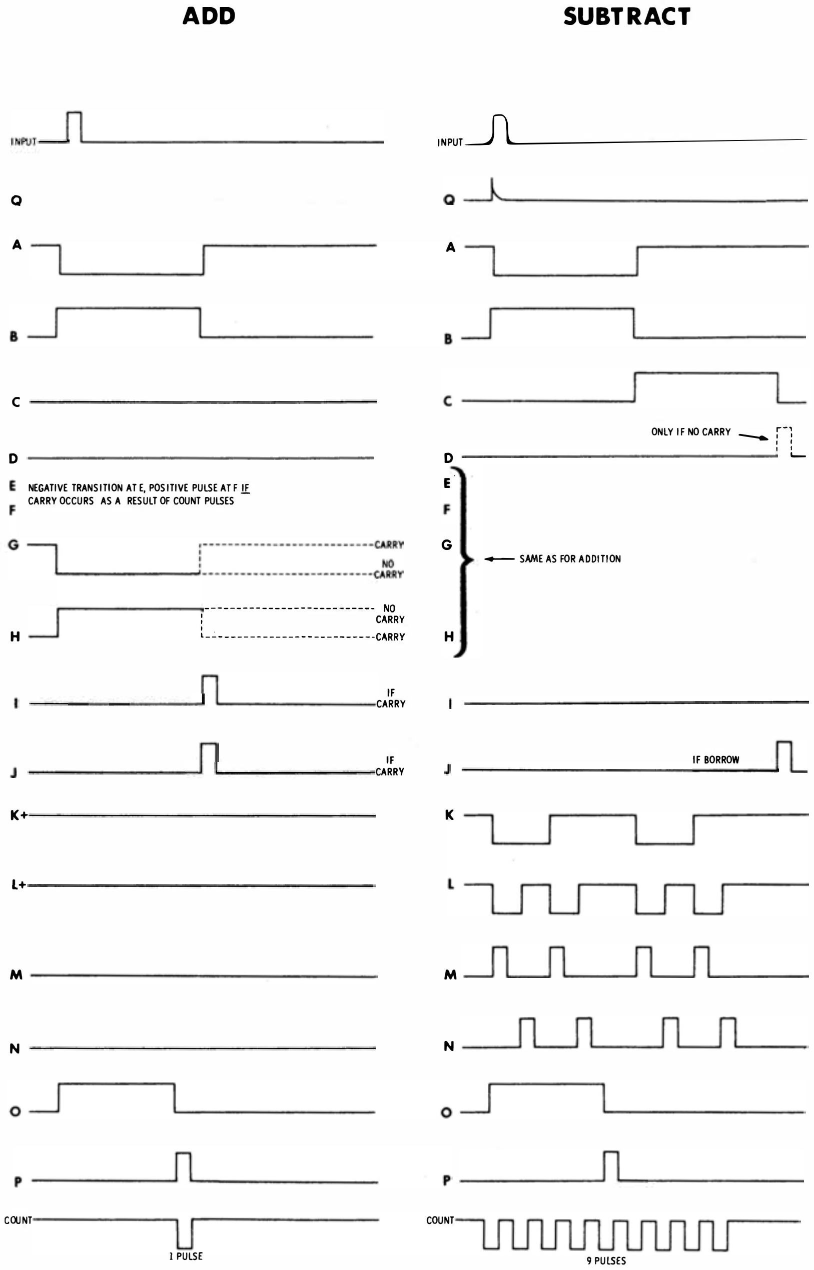


Fig. 7-26. Waveforms of add-subtract adapter.

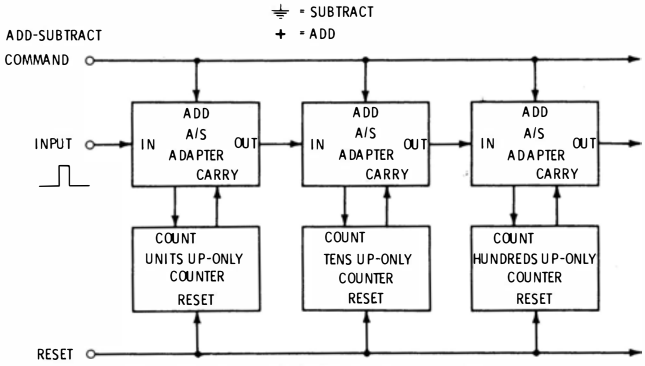


Fig. 7-27. Block diagram of up-down counter.

manner, subtracting if a borrow is sent it, and adding if a carry is sent it, since it is responding to the same add/subtract command the first stage is.

This type of counter is slow, but it is relatively inexpensive and it permits use of virtually any up-only counter, except a phase-shift ring, and any desired weighting and decoding. During subtraction, all decoded outputs will be flashed briefly. This effect will not be noticeable in visual readouts, but if an electrical readout is needed, it will be necessary to blank the outputs during the subtraction process.

The circuit shown uses positive logic and positive pulses on the input and output lines, and the normal negative logic transitions are routed to the up-only counters. If you prefer negative logic throughout, add an inverter to the input terminal, and remove the inverter from the output terminal. The circuit may be built with two hex inverters, three quad two-input gates, and a single transistor.

CHAPTER 8

Digital Instruments and Other RTL Applications

We have covered all the individual circuits needed to build almost any electronics project with RTL. The question now is how to interconnect the basic circuits to make useful and functional electronic systems.

This chapter deals with three kinds of applications. The first is *digital* instruments. While anything built with RTL is really some sort of digital instrument, we will apply the term only to those projects that have numeric displays. This category includes events counters, electronic stop watches, digital voltmeters, etc. A look at the speed and accuracy limitations inherent in digital displays and instruments of this type will be included.

Secondly, we will turn to unusual applications of RTL in some relatively complex electronic gear. These applications use RTL to reduce system cost or to provide performance where no reasonable electronic system was available previously to do the job. Examples will be a precision timer, electronic dice, a musician's pitch reference, a television-service dot-and-bar generator, an electronic calculator, and some simple power controls.

Finally, the book will conclude with some suggestions on how to design you own simple RTL projects—low-cost hobby, service, or science-fair projects that may be built easily in a few evenings.

Most of the circuits covered in this chapter will be shown in block-diagram form, instead of by complete schematics. This approach will make it possible to talk about many more applications than could be included if complete constructional design information were given for each one. Details of most of the individual building blocks have been covered in previous chapters.

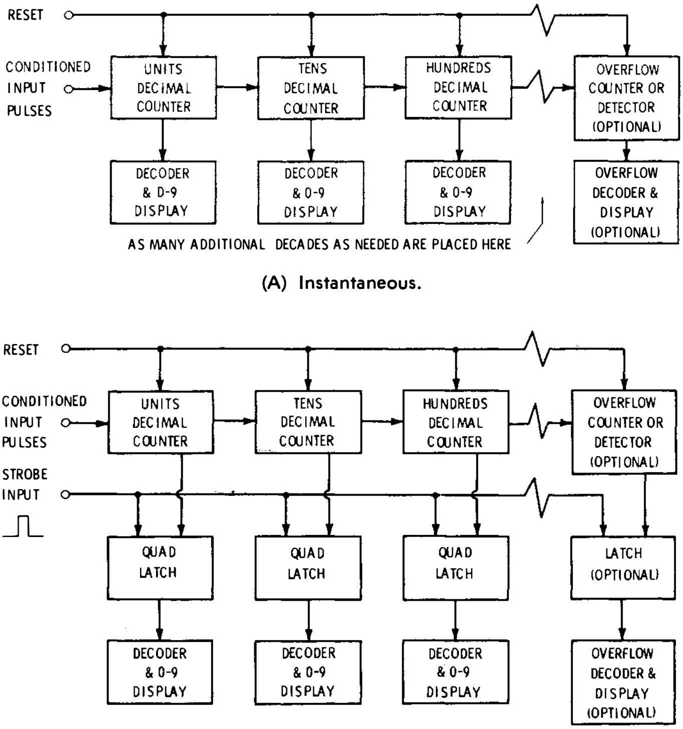
Several of the projects in this chapter have appeared in *Popular Electronics* magazine and are included here with the permission of the publisher.

Most of the projects in this chapter are available in kit form. One source of these kits is Southwest Technical Products Inc., Box 16297, San Antonio, Texas 78216.

DIGITAL INSTRUMENTS

The heart of any digital instrument is a *digital display.* A digital display is a series of cascaded, resettable decimal counters with their readouts, arranged so that you can read the *total* number of input counts accumulated since the display was reset last. Fig. 8-1 shows two popular types of digital displays.

Fig. 8-1A shows an *instantaneous* digital display. In this display, conditioned input pulses drive a series of cascaded decimal counters. Each decimal counter has its output decoded and sent directly to a 0-9 indicator. During the counting operation, an instantaneous display will be a blur, or, on slower counts, the numerals will "bounce along" with the input counts. It is only when the counting sequence is finished that the instantaneous display stops and indicates an answer.



(A) Instantaneous.

(B) Strobed.

Fig. 8-1. Two types of digital display.

The *strobed* display of Fig. 8-1B always indicates an answer. Between each counter and its decoder is added a memory circuit called a quad latch. Upon a strobe command, the particular state of the counter is transferred and held in the quad latch. The digital display then always reads the last answer, even when the counter is working on a new count.

An instantaneous display is always more economical than a comparable strobed display, but it is rather slow in that it must take time out from measurement to give enough time for reading and interpreting the answer. The problem of blurred or changing numbers during measurement can be overcome sometimes by *blanking* the display, turning it off entirely during the measuring interval. The strobed display needs only a microsecond or so to transfer the information from the counter to the latch. Thus, a strobed display is faster, and we can make continuous measurements. The price paid for this capability is in the extra flip-flops required for each quad latch. Three quad two-input gates and an inverter are needed for each decade. (At the time of this writing, a one-IC RTL quad latch has been newly announced.)

Only three decades are shown in Fig. 8-1. As many decades as needed may be added by connecting the carry output of each decade to the count input of the next decade.

Fig. 8-1 also shows an *overflow counter*, or *overflow detector*. This device goes onto the end of the counting chain and can do two things. It can tell when the "full-scale" capacity of the counter has been exceeded, and it can extend or double the display capability without the need for an additional decade. Usually it adds "half" a digit. A digital display with a "2½" digit range usually covers 0-159 or 0-199; a "3½" digit range usually covers 0-1599 or 0-1999.

An overflow indication is far more than a convenience, for we otherwise have no way of telling when the counter spills. For instance, a 1534 counter reading may really be 651,543; the two most significant digits of the measurement are lost.

Overflow counters take several forms. We can use a dual flip-flop, running the first as a binary divider and the second as a sequential pass-on. The 0 and 1 indicators may be driven from the $\overline{Q}$ and Q outputs of the first stage, and a red X or overrange light may be driven from the Q output of the second stage. The second stage is designed to latch on the first overflow, and stay lighted regardless of the number of extra counts after that.

A second form of overflow counter that manv commercial instrument makers use is one in which the *final* decade counter is really a base-sixteen counter. The decoding is arranged to drive *both* a 0-9 and a 0-1 display. This arrangement requires a decoder that produces a 1 output for both counts 1 and 11, a two for both 2 and 12, and so on.

Both of these schemes provide "more" instrument without much extra cost. Instead of a ten-digit readout on the last stage, a single extra-long neon-lamp "1" or a single pilot lamp is used.

A digital display by itself can *not* measure frequency, time, or voltage. All it can do by itself is totalize input pulses. Only if and when the input pulses stop arriving does the digital display stop and settle down to a continuous reading. Shortly, we will discuss how to make the input count pulses a function of frequency, time, or voltage. First, however, it is necessary to turn to an important and often overlooked limitation on digital displays, the speed-accuracy product.

SPEED-ACCURACY PRODUCT

Just as there is a limit to the gain and frequency response in an analog instrument, there is a similar limitation in accuracy and response speed in a digital instrument. This limitation is called the *speed-accuracy product.* It reveals two things: The more accurate we want a measurement to be, the longer we have to take; and there is a fundamental limit to the *product* of speed and accuracy that is determined by how many decades are in use, and by how often the input count pulses are arriving.

The *accuracy* of a digital display by itself is plus zero, minus one count. If the input counts are turned on and off at random, this accuracy drops to plus or minus one count. If a synchronous gate is used to turn the input counts on and off, an accuracy of plus zero, minus one count is retained. Thus, there are two good reasons for using a synchronous gate—it doubles the accuracy, and it eliminates any last-digit fluctuation.

The accuracy is in terms of the *actual reading*, and not the full-scale capability. For instance, on a three-digit display, a 100 reading will only be accurate to −1, +0 percent, while a 999 reading will be accurate to nearly −0.1, +0 percent.

The other side of the accuracy coin is dependent on the accuracy of the input pulses being received. If we are just synchronously counting pulses, there should be no trouble, but if we are attempting to measure frequency, time, or voltage, the problem is more difficult. *The number read on a digital display can be no more accurate than the accuracy of the input counts fed to the display.*

For instance, if we build a digital voltmeter with a 1-percent-accurate voltage-to-frequency converter, we could use 75 decades in a digital display and still have no better than the 1-percent accuracy of the converter. If we build a universal counter that uses the power line to control a gate, we can only get at best the ±0.05-percent frequency accuracy of the power-line voltage, again regardless of the number of decades. If we build a ballistic-velocity meter or chronometer, the accuracy will be no better than the accuracy to which we measure the screen spacings and the stability of the screen spacings, even if we use many decades and a crystal frequency reference calibrated against WWV.

The important point here is that you cannot arbitrarily increase the accuracy of a digital instrument simply by adding more decades. The economical thing to do is to *match* the accuracy of the digital display to the accuracy of the input count signals. More decades are harder to read, they give meaningless digits, and they cost money.

The speed half of the speed-accuracy product depends on the maximum input count rate and the number of decades being used. With a 100-kHz count rate, a five-digit display requires one second maximum to indicate the right answer. A seven-digit display requires a minute and forty seconds, and a nine-digit display takes a total of two hours, forty-six minutes, and forty seconds to produce the right answer. This time requirement is the second price that must be paid for extra digits. There is a certain maximum counting speed; that speed divided into the total number of counts possible with the number of decades in use determines the maximum time needed to reach the correct answer.

Frequently, the maximum clock rate is limited by factors other than the "wide-open" 4-10 megahertz maximum counting speed of a single RTL flip-flop. These limitations take the form of the speed with which a voltage-frequency conversion can be made, the ripple times through cascaded counting chains, the response time of an input transducer, etc. Since the accuracy is usually limited by external circuits anyhow, you should always use the minimum possible number of decades in a digital display that will permit doing the required job. Besides its obvious economy, this approach makes the results available more quickly.

While there are subtle methods of "beating" the speed-accuracy limitations, such methods usually lead to complex solutions to a measuring problem. For instance, we can accurately measure the frequency of a noise-free, low-frequency sine wave by zero-crossing detection. By starting and stopping a high-frequency reference oscillator, we can measure the period (and hence the frequency) of the low-frequency signal within a single cycle, perhaps to an accuracy that would require thousands of cycles of direct frequency measurement. Also, the digital displays in this book are *serial* systems. The computer people sometimes use a parallel system which *simultaneously* works on tens, hundreds, and thousands, instead of letting the carry from the units operate the tens, etc. While this technique is of great value for computers, few instruments can use it, and, in general, parallel digital circuits are considerably more complex and expensive than serial circuits.

COMPOSITE DIGITAL INSTRUMENTS

Digital techniques can be applied to a number of measurement problems. A dozen of the most common applications are described in this section.

Events Counter, or Arithmetic Totalizer

One of the simplest digital instruments is diagrammed in Fig. 8-2. Basically, it consists of a digital display, input conditioning, and a reset

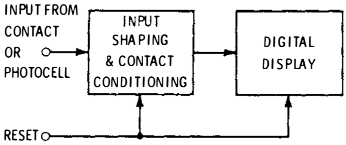

INPUT FROM CONTACT OR PHOTOCELL
INPUT SHAPING & CONTACT CONDITIONING
DIGITAL DISPLAY
RESET

Fig. 8-2. Digital events counter.

push button. This instrument *counts* input pulses. When the pulses are not coming in, the counter *displays* the total accumulated. The digital display can be reset to 0000 at any time; the reading any time after reset tells the total number of events since reset.

There is a wide choice of input-conditioning circuits, as discussed in Chapter 3. Regardless of the method selected, we must be sure there is one and only one abrupt negative-going input transition for each desired input count. If photocell inputs are used, it is necessary that the light not be 120-hertz modulated by the power line; such modulation can cause false counts under marginal light levels. Either use a lamp with a very long time constant (photoflood, etc.), or use dc lamp power.

Generally, the frequency response of the inputs should be limited to just barely more than is needed to accept the input counts, and the signals can be conditioned from there. This technique can help keep high-frequency noise out of the system.

Period or Time Measurement

The *time* during which an input signal is present is measured by the type of digital circuit in Fig. 8-3. We start with a stable frequency reference, perhaps a crystal oscillator using one of the circuits of Chapter 4. We reset the digital display. When the event to be timed *starts*, a gate opens, and pulses at the reference frequency are fed into the digital display. When the event stops, the pulses are stopped, and the digital display reads the total number of pulses that occurred *during* the event we are trying to time. The digital readout multiplied by the *period* of the reference signal gives the elapsed time. For instance, if we use a 100 kHz clock, the reference period is 10 microseconds. A reading of 797 means it took 7970 microseconds, or 7.97 milliseconds, for the measured event to occur.

Input conditioning once again is required, but since a gate and not a flip-flop is being operated with the input signal, the problem is not severe.

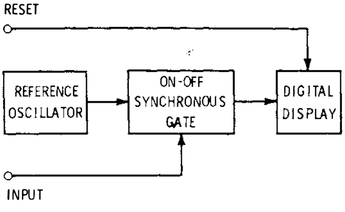


Fig. 8-3. Measurement of period, or time.

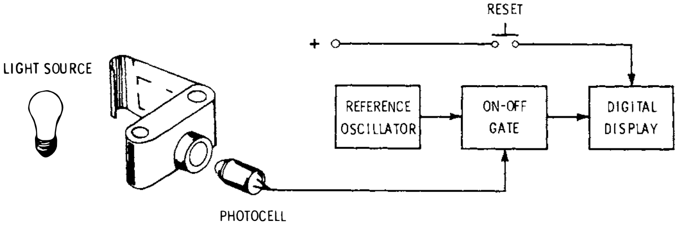


Fig. 8-4. Tester for photographic shutters.

We simply have to be certain that the input voltage rises in less than one reference period, stays up with no bounce for the rest of the measurement interval, and then returns to ground in less than one reference period. Any "holes" in the "on" response will reduce the apparent time measurement.

Photographic-Shutter Tester

Fig. 8-4 shows a practical example of a time-measuring digital instrument. It permits accurate measurement of the length of time a camera shutter remains open. A photocell is placed on one side of the shutter, and a photoflood lamp is placed on the other side so that the photocell is illuminated only through the shutter. A good-quality, high-speed, silicon photocell and a suitable conditioning circuit should be used. The reference frequency depends on the number of decades and the range of the shutter we wish to test. For shutter times of 0.1 to 1 second and a three-place digital display, use a 1-millisecond reference. Increase the reference speed accordingly when measuring faster shutter speeds. The instrument will have a 1-percent worst-case accuracy, far better than the capabilities of many makeshift shutter testers. For more accuracy, go to a four-place display and a reference frequency that is ten times faster. A number of reference frequencies are easily obtained with a 100-kHz oscillator and a few decade dividers.

Measurement of Time Between Events

We can measure the time *between* events instead of the duration of a single event by going to a start-stop gate instead of an ordinary duration gate (Fig. 8-5). Here, the first event starts the counting action, which con-

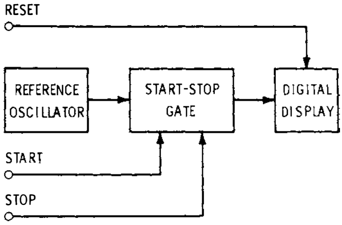


Fig. 8-5. Measurement of time interval between events.

tinues after the first event has ended. The second event stops the counting, and the digital display reads the total time between the two events.

To perform this type of measurement, we usually use a set-reset flip-flop that is leading-edge triggered; one side is driven from the start input and one from the stop. The output of the flip-flop drives either a conventional or a synchronized switch; this switch then connects the reference-oscillator pulses to the digital display.

Ballistic Chronometer

A practical application of the time-between-events digital instrument is the accurate measurement of the speed of a bullet (Fig. 8-6). We place two screens a known distance apart and fire a round through them. Depending on the type of screens used, the bullet either momentarily makes contact or permanently breaks contact as it passes through. In either case, the counting action is started by the first screen and stopped by the second. A simple calculation then yields the velocity of the bullet.

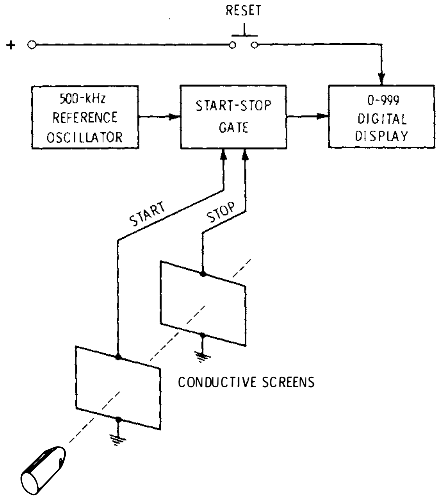


Fig. 8-6. Method for measurement of ballistic velocity.

For instance, suppose the reference frequency is 500 kHz and the screen spacing is 4 feet. Suppose the test round produces a reading of 676. Each reference-frequency pulse represents 2 microseconds, so a total of 1352 microseconds, or 1.352 milliseconds, must have elapsed. If the bullet went 4 feet in 1.352 milliseconds, it must have gone 1 foot in 0.338 millisecond, or an equivalent velocity of $1/(0.338 \times 10^{-3}) = 2960$ feet per second.

There are several precautions that must be taken with this instrument. The first screen must be far enough away from the muzzle of the pistol or rifle to avoid any blast effects. Also, the distance between the screens must be measured with extreme accuracy, or the overall accuracy of the measurement will be limited severely. For instance, to get a 0.1-percent instrument accuracy, the screen spacing must be accurate to ⅓₂ inch. In addition, the screens have to remain stationary to the same tolerance as the bullet passes through them, and a newly broken screen must not produce any contact bounce.

The commercial kit instrument of Fig. 8-7 covers velocities of 1000 to 5000 feet per second on two overlapping scales, with a rated accuracy of 0.2 percent. It is either battery or ac operated, works with either screen type, and tests its own batteries and screens.

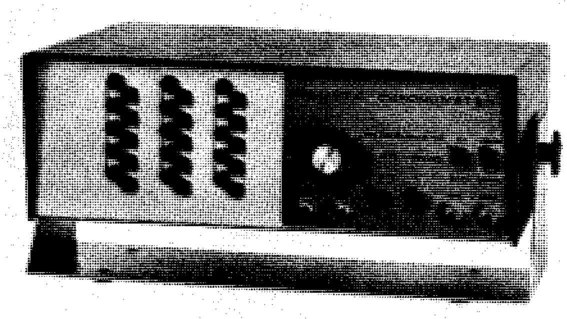


Fig. 8-7. Kit-type ballistic chronometer.

Similar techniques with slower reference frequencies may be used to measure sports-car or dragster velocities. Usually, a photoelectric pick-off is used, but a break-wire trip may serve the same purpose. The technique works equally well with measurement of time-to-quarter-mile or terminal velocity at the quarter-mile trap.

Electronic Stop Watch

Instruments for measurement of event duration and the time between events may be combined into one universal circuit that provides a wide range of reference frequencies and gives a choice of duration or start-stop gates. The version shown in Figs. 8-8 and 8-9 offers a choice of six reference frequencies from 50 kHz down to 1kHz, with corresponding time periods of 0.02 to 1.0 millisecond. This range allows ballistic-velocity measurement, photographic-shutter testing, physics experiments, and any other accurate timing measurements in the range from 20 microseconds to 1 second. The same circuit also counts events, acts as a divide-by-10, divide-by-100, or divide-by-1000 scaler, and can be used to generate three-digit random numbers.

By the addition of suitable divide-by-ten and divide-by-six circuits, the basic instrument can be modified to handle seconds and minutes. When so many decades are involved, the limiting accuracy will most likely be determined by the accuracy of the inputs and not by the total display reading.

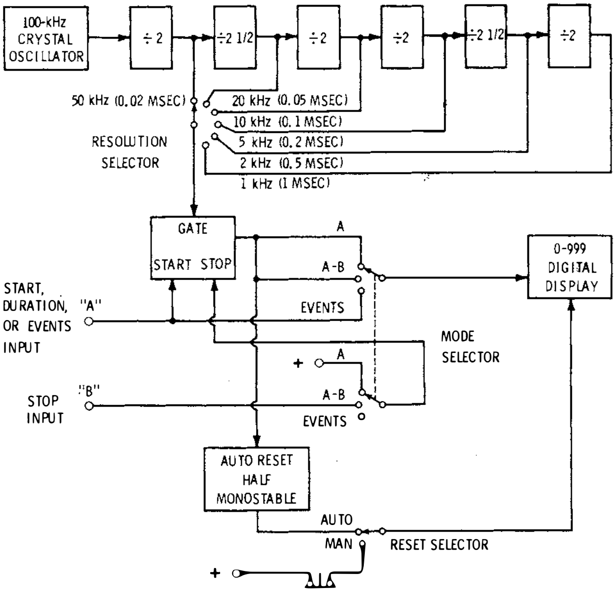
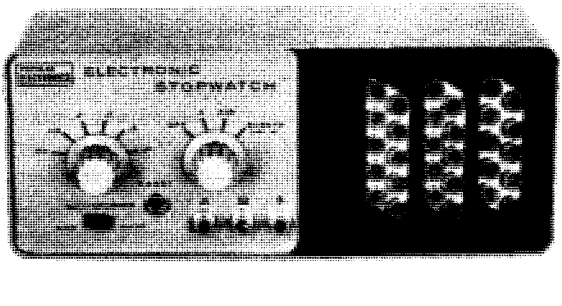


Fig. 8-8. Circuit of electronic stop watch.

Digital Clock

A much more reasonable approach to a digital clock or to measurement of long time intervals is to start with a 60-hertz power-line reference, divide by six to get tenths of a second, by ten to get seconds, by ten again to get tens of seconds, by six again to get minutes, and so on (Fig. 8-10). While a crystal reference could be used stop-watch fashion, it is far more expensive, and practically all time-measurement applications are accuracy limited by factors other than the ±0.05-percent tolerance the power line normally provides. Thus, unless you absolutely need much better than



Courtesy Popular Electronics

Fig. 8-9. An electronic stop watch.

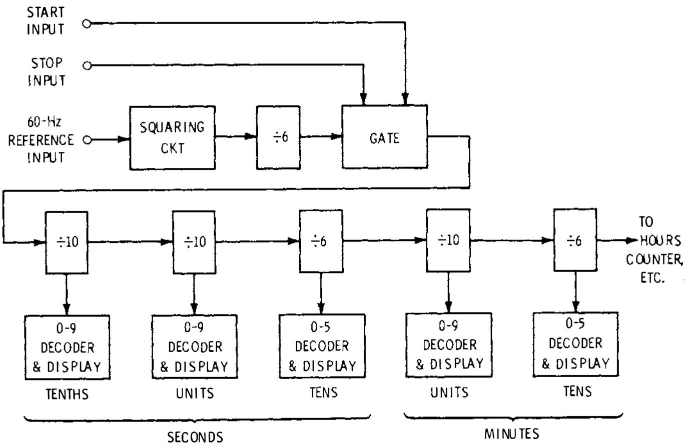


Fig. 8-10. Circuit of a digital clock.

±0.05-percent accuracy, the extra decade dividers and crystal reference only add substantially to the circuit cost without providing anything better in the way of performance.

The completed instrument can be "free-run" as an ordinary clock, or gated stop-watch fashion to measure either the duration of an event or the time between events. Sports-car rallies are a good application for instruments of this type.

Frequency Counter

A frequency counter (Fig. 8-11) measures events *per unit time.* It is probably the most popular single digital instrument. To build one, we need a precision time reference, usually derived from a crystal if we need exceptional accuracy, or from the power line if a nominal ±0.05-percent accuracy is acceptable. A switch is opened for a time interval determined by the reference generator. During the time the switch is open, the input counts are routed to the digital display, and the display then indicates the events

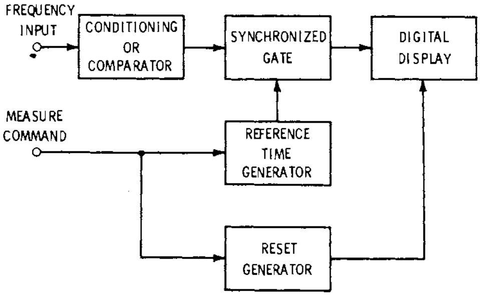


Fig. 8-11. Circuit of a frequency counter.

that have occurred during the reference time. This operation is exactly backward from that of the electronic stop-watch circuit, for in a counter we control the switch with the reference and count the input, while on the stop watch we counted the reference and controlled the switch with the input signal.

The usual choices of time reference are 0.1, 1, and 10 seconds, with the longer times being reserved either for very accurate measurements or for low-frequency measurements.

The digital-display reading *divided* by the reference time gives the frequency. For instance, a reading of 2843 equals 28.43 kilohertz if you are using a 0.1-second time gate, 2.843 kHz if you are using a 1-second time gate, and 284.3 hertz if you are using a 10-second time gate.

Conditioning of the input is particularly critical because the input counts are fed directly to the digital display. While high-level squaring circuits may be used, they are normally fairly low in impedance and require considerable input signal level. Consequently, a comparator is almost always used, particularly when low-level or noisy signals are involved.

Universal Frequency Counter

A universal frequency counter (Fig. 8-12) is simply a frequency counter that makes available the widest possible choice of input frequency ranges. The commercial kit instrument of Fig. 8-13, for example, has five frequency ranges from 0-200 hertz through 0-2 megahertz, as well as provision for counting events or for use of an externally provided gate signal. It has a comparator input and provides 0.1-, 1-, and 10-second gate-time

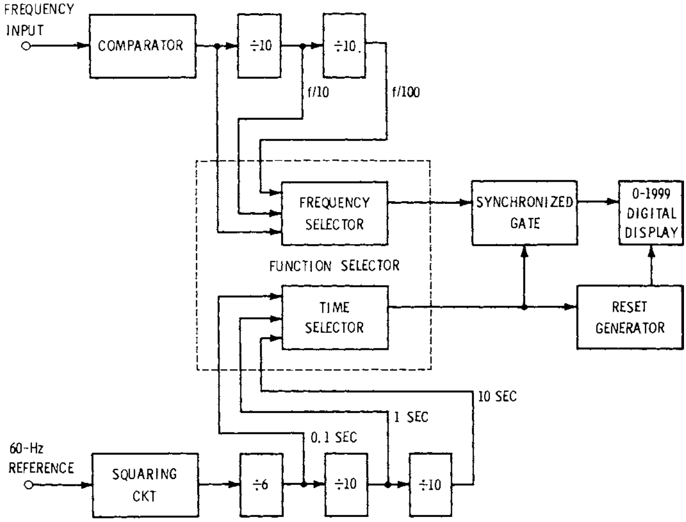
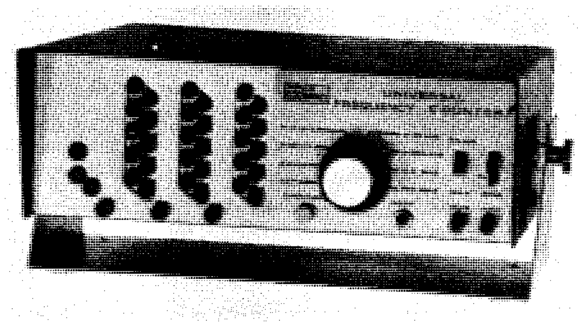


Fig. 8-12. Circuit of a universal frequency counter.



Courtesy Popular Electronics

Fig. 8-13. Kit version of universal frequency counter.

references. The 0-1999 digital display is compatible with the nominal power-line accuracy from which the time gates are derived. Higher input frequencies may be measured with input decade dividers.

If greater than power-line accuracy is needed, an electronic stop-watch type of crystal reference and divider may be used to provide a more stable reference time source.

Digital Voltmeter, Gated-Oscillator Type

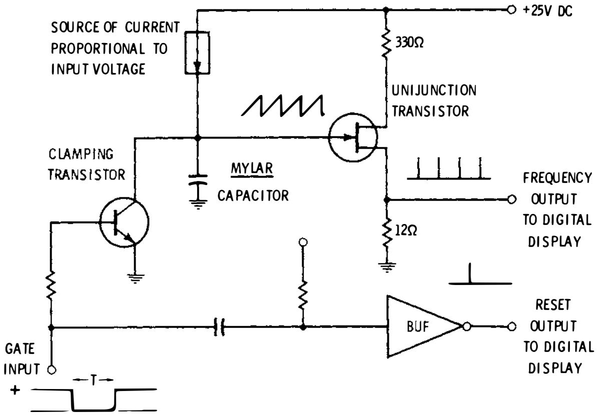
One of the fastest-developing digital instruments today is the digital voltmeter, along with its companion digital multimeters and digital panel-meter replacements. Only very recently have such instruments become economical, and at present, only the use of RTL lets you build digital instruments that compare in price to their older, less accurate, analog equivalents.

To measure voltage, we must somehow *convert* that voltage to a series of pulses, with either the total number of pulses or the frequency of the pulses proportional to the input voltage. This process is called *voltage-to-frequency conversion*, or V/F conversion for short. The result of the V/F conversion is then routed to a digital display. System constants are chosen to make the displayed number identical to the numerical value of the input voltage.

Unless we are trying to build a combination digital instrument, the gating and sequencing circuitry for a digital voltmeter can be far simpler and cheaper than that required for a universal frequency counter. Simply building a V/F adapter and adding it to the front of a universal counter, while it certainly will work, is a very expensive route to follow.

In general, V/F conversion tends to be expensive and complex, particularly if we need a great deal of accuracy or have to operate at high speeds. Today, there are two relatively economical techniques for V/F conversion, using the *gated oscillator* and the *dual-slope integrator*. The gated oscillator is usually the less expensive of the two and can be made to reject all 60-hertz hum and noise quite easily. It always measures for a constant time.

On the debit side, its accuracy is limited to the 1-percent range, and it requires both calibration and zero controls. The dual-slope integrator is the standard circuit used in the more expensive digital voltmeters. It offers better accuracy (to 0.01 percent) than the gated oscillator, has better linearity, needs no zero adjustment, and cancels its own nonlinearities. Its disadvantages are a higher cost, measurement for a nonconstant time, and the inability to reject line-borne 60-hertz hum and noise completely. The dual-slope system also needs a reference frequency, but the reference frequency does not have to be exceptionally stable for proper operation.



+25V DC
SOURCE OF CURRENT PROPORTIONAL TO INPUT VOLTAGE
330Ω
UNIJUNCTION TRANSISTOR
CLAMPING TRANSISTOR
MYLAR CAPACITOR
FREQUENCY OUTPUT TO DIGITAL DISPLAY
12Ω
BUF
RESET OUTPUT TO DIGITAL DISPLAY
GATE INPUT
+
T

Fig. 8-14. Simplified schematic of gated-oscillator V/F converter.

One typical gated oscillator is shown in Fig. 8-14. A gated oscillator is simply a voltage- or current-controlled oscillator whose frequency is made proportional to the input voltage or current, and whose output is turned on only for a specified gate time. The number of output pulses depends on both the input voltage and the gate duration; by selecting the correct circuit parameters, we can get 135 counts out for 1.35 volts in, and so on.

Usually the oscillator is turned on and off with a power-line-derived gate. When the oscillator runs, the digital display accumulates the next reading. When the oscillator stops, the digital display shows this reading. Just before the oscillator is allowed to start again, a brief reset pulse erases the old answer. In fact, it is possible to erase the display after the oscillator starts, providing the erasure is completed before the oscillator has a chance to put out its first pulse. This method is considerably simpler, for all we have to do is half-monostable the leading edge of the "measure" gate to derive a reset pulse. This same technique works on universal counters and other digital instruments also.

The circuit shown is a unijunction-transistor relaxation oscillator. It is driven by a constant-current source that delivers a current proportional to the system input voltage. The oscillator is allowed to run only for a precise time determined by the gate input, when the clamping transistor is off. The *frequency* of oscillation is determined by the input current. *How long* oscillation takes place is determined by the input gate. The total number of pulses produced is determined both by the input current and the gate duration. This total number of pulses is adjusted to correspond to the input voltage.

The circuit is calibrated by adjusting either the timing capacitor or the proportionality between the input voltage and the current supplied to the oscillator. Zero adjustment is obtained by introducing some leakage current around the proportional-current source. The short-term stability of this circuit can approach 0.1 percent, but both zero and calibration potentiometers are needed to reach this stability.

By using a signal derived from the power line, dividing by four and decoding one of four, we can produce a gate that measures the input for 16.7 milliseconds, and then displays for 50 milliseconds, updating the readings 15 times a second. The result is an apparently continuous display. The choice of a line-locked gate means we average 60-hertz hum for one whole cycle each time. The one-cycle time average of a sine wave is zero; the same is true for the harmonics. Thus line-locked gating is essentially blind to any input hum or noise, a very desirable feature in low-level measurement systems.

Digital Voltmeter, Dual-Slope-Integrating Type

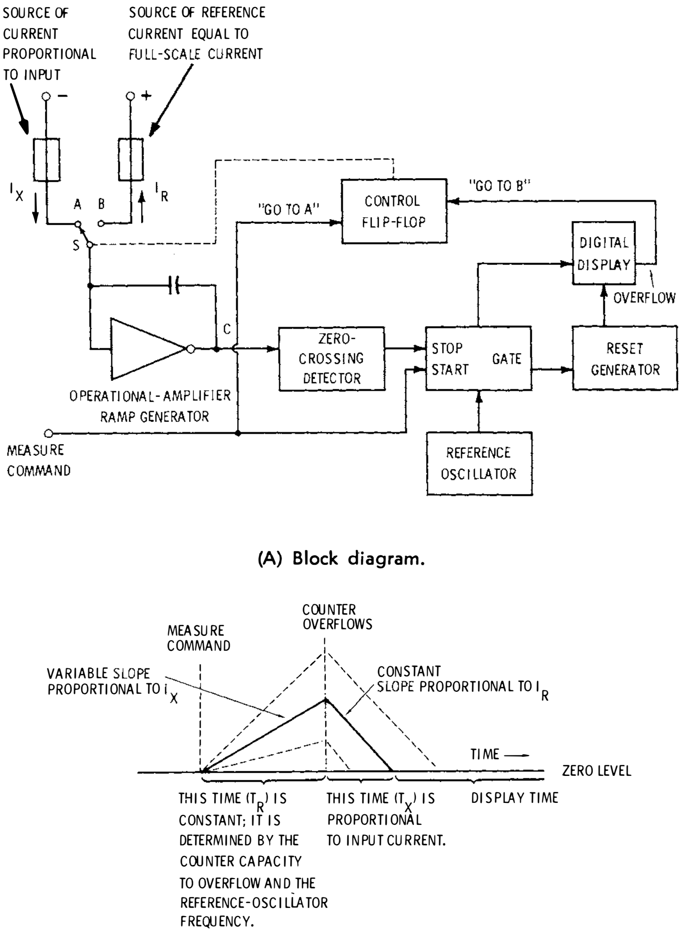
The dual-slope integrator provides a more complex and more accurate method of V/F conversion. Besides a digital display, we need an operational-amplifier ramp generator, a reference-frequency oscillator, and a precision reference-current source.

Briefly, the circuit of Fig. 8-15A works as follows. While the digital display is showing the previous answer, the reference oscillator is disconnected from the digital display. The control flip-flop holds control switch S in position B. The output of the ramp generator rests at ground.

Now, a measure-command pulse arrives. This pulse immediately opens the start-stop gate, and the reference-oscillator pulses are fed to the digital display. At this instant, the control flip-flop switches to position A. The digital display starts accumulating counts, and the operational-amplifier ramp generator starts building a positive ramp (Fig. 8-15B).

The digital display is allowed to run until it overflows. At the instant of overflow, a known time T_R has elapsed, since a known number of reference pulses (equal to the counter capacity) have been obtained from a source of known frequency. During this known time, the input current (I_X) has caused the output of the operational-amplifier ramp generator to reach some positive voltage. The greater I_X is, the more positive this voltage is.

Now, at the instant of overflow, the control flip-flop switches S to position B. The reference oscillator is still allowed to drive the digital display, and the digital display simply starts counting over again. A reference cur-


SOURCE OF CURRENT PROPORTIONAL TO INPUT
SOURCE OF REFERENCE CURRENT EQUAL TO FULL-SCALE CURRENT
−
+
I_X
I_R
A
B
S
C
"GO TO A"
CONTROL FLIP-FLOP
"GO TO B"
DIGITAL DISPLAY
OVERFLOW
ZERO-CROSSING DETECTOR
STOP
START
GATE
RESET GENERATOR
OPERATIONAL-AMPLIFIER RAMP GENERATOR
MEASURE COMMAND
REFERENCE OSCILLATOR
COUNTER OVERFLOWS
MEASURE COMMAND
VARIABLE SLOPE PROPORTIONAL TO I_X
CONSTANT SLOPE PROPORTIONAL TO I_R
TIME —
ZERO LEVEL
THIS TIME (T_R) IS CONSTANT; IT IS DETERMINED BY THE COUNTER CAPACITY TO OVERFLOW AND THE REFERENCE-OSCILLATOR FREQUENCY.
THIS TIME (T_X) IS PROPORTIONAL TO INPUT CURRENT.
DISPLAY TIME

(A) Block diagram.

(B) Integrator action.

Fig. 8-15. Dual-slope-integrator V/F converter.

rent, I_R, of opposite polarity to I_X, is fed to the ramp generator, and the ramp generator starts integrating back toward zero. When the ramp voltage reaches zero, a time T_X will have elapsed. When the voltage reaches zero, the start-stop gate turns off the reference-oscillator pulses, and the digital display reads the time needed for the ramp generator to integrate back down to zero.

There is a simple relationship between the times and the currents:

$$\frac{I_X}{I_R} = \frac{T_X}{T_R}, \quad \text{or} \quad T_X = T_R \frac{I_X}{I_R}$$

This formula shows that the output run-down time (T_X) is proportional to the ratio of the input currents (T_R is constant). Since the reference current (I_R) is known, the output count follows the input current.

Better yet, any nonlinearities in the ramp generator cancel themselves, for whatever nonlinearities are produced on the way up to a positive voltage are eliminated on the way back down. This is the big advantage of dual-slope integration—all we need is a ramp generator that returns to its starting point; the voltage produced at any instant is not particularly important. A linearity of 0.01 percent can be obtained with this technique.

A second desirable feature is that the operation is independent of the reference-oscillator frequency. Usually we can get by with an astable reference oscillator instead of a precision crystal source. A reasonable short-term stability is required from the reference oscillator, but this requirement is fairly easy to meet.

The input is always scaled so that I_X is *less than* I_R. This makes T_X less than T_R so that we always get a less-than-overflow reading on the counter. The digital display is reset the instant a new measure command arrives.

Digital Multimeter

The commercial kit multimeter in Figs. 8-16 and 8-17 measures dc volts and ohms to 1-percent accuracy. It measures 0-2, 0-20, and 0-200 volts dc, as well as resistance from 1 ohm to 200,000 ohms on four ranges. It may be calibrated with an internal mercury reference cell.

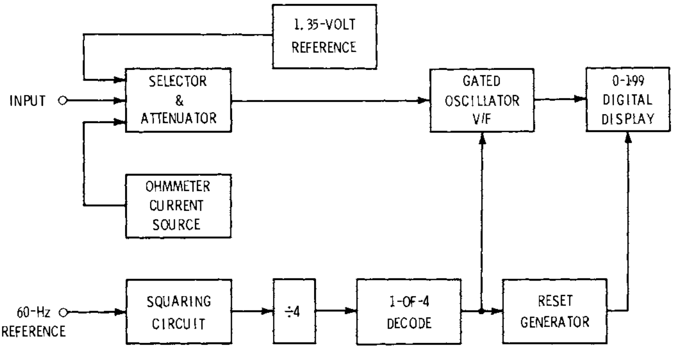
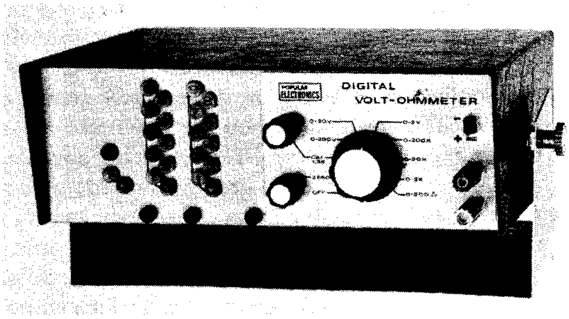


Fig. 8-16. Diagram of digital multimeter.

Resistance is measured by providing a known, fixed current, and then measuring the voltage drop produced across the resistor under test. The maximum metering current is 10 milliamperes, allowing the safe measurement of delicate solid-state components. The actual value of the resistance appears on the digital display.

A gated-oscillator V/F converter is used, synchronously run with the power line to eliminate effects of 60-hertz hum.



Courtesy Popular Electronics

Fig. 8-17. Kit version of digital multimeter.

Newer versions of this circuit offer a full *Nixie* display and are somewhat smaller.

SOME RELATIVELY COMPLEX RTL APPLICATIONS

We turn now to some newer and better places to put RTL to use—as replacements for circuits that were too complex or otherwise restricted in performance when built with older circuitry. In each case, RTL greatly reduces cost, improves performance, or both.

The important thing to note in these applications is that we do not simply take each bit and piece of an old application and attempt to digitize it. We build instead an entirely new *system* that provides the same or better *results* as the conventional circuit did, but is designed to obtain those results in the way that best suits RTL.

Precision Timer

Photographic and other seconds/minutes timers usually depend on large electrolytic capacitors for their time-interval determination. Precision resistors and relatively complex switches are also normally used. In time, the capacitor and trip-point values drift, and the long-term accuracy cannot be relied upon. Worse yet, the long time intervals are often not as accurate as the short ones, and in spite of a three- or four-decade selector assembly, it is often difficult to get more than a few percent accuracy for long time intervals.

More satisfactory results are obtainable by using RTL digital techniques to count pulses derived from the power line instead. The long-term stability is much improved, there is no need for any calibration or precision components, and the circuit costs turn out about the same.

We square up the power-line signal and divide by six to get 0.1-second timing pulses (Fig. 8-18). We then use programmable phase-shift ring counters with their single-pole selector switches to select the tenths, sec-

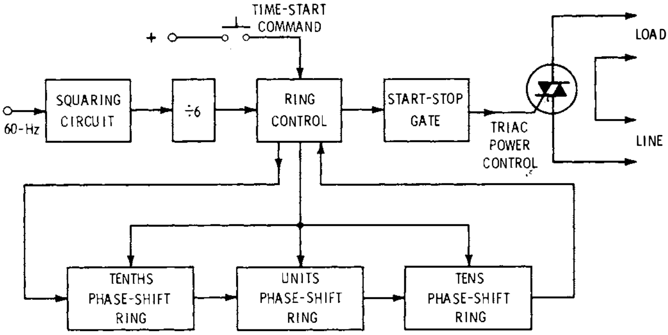


Fig. 8-18. Timer with power-line reference.

onds, and tens of seconds. A start-stop gate connected to a *Triac* provides solid-state control of 110 volts ac without any mechanical contacts. Several kilowatts can be handled if necessary.

The phase-shift ring is normally held in the 999 state. When a time-start command arrives from either a manual button or foot switch, the ring control adds one count and then starts counting the 0.1-second pulses. At the same time, the *Triac* is turned on and provided with continuous gate drive. When the proper count is reached (corresponding to the selected time), the ring is reset and the *Triac* gate drive is removed. Timing is thus accomplished with line accuracy, no calibration, and no internal load-current-carrying contacts.

Electronic Dice

A pair of electronic dice (Fig. 8-19) is useful for parlor games, particularly when small children are playing. The same novelty item is quite useful

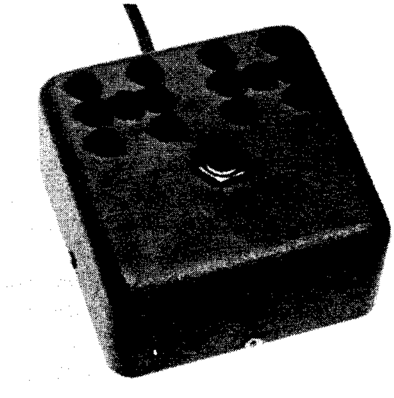


Fig. 8-19. RTL electronic dice.

Courtesy Popular Electronics

for promotion or displays, and for science-fair demonstrations of probability or chance.

While many "electronic-dice" circuits have appeared in the hobby magazines, most of them have one or both of two serious faults. Either the numbers are not weighted in true dice fashion with true dice odds, or while the numbers are claimed to be "random," they are *not* equiprobable. With RTL, we can overcome both of these objections and build an instrument that in every way duplicates the honest throw of an honest pair of dice.

We should first note that throwing two six-sided dice simultaneously is the same, with respect to probability, as throwing one 36-sided die, each side of which is identified with one of the 36 possible and equiprobable two-die combinations.

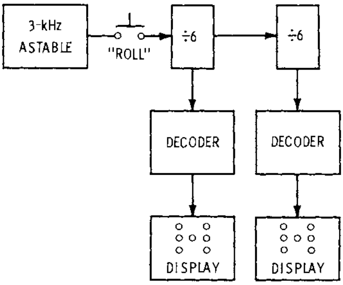
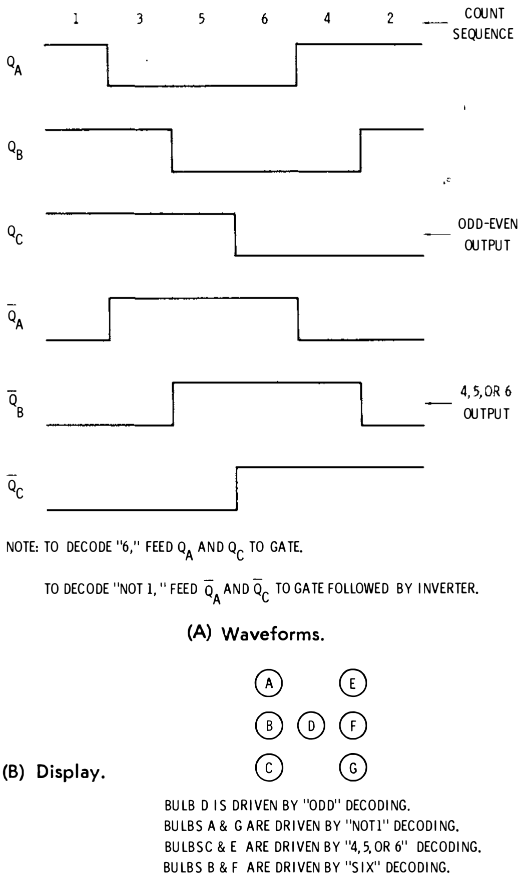


Fig. 8-20. Diagram of RTL electronic dice.

The circuit (Fig. 8-20) consists of two cascaded modulo-six walking ring counters. The counters are driven by an astable multivibrator when the roll button is depressed. The frequency is made high enough that depressing the button for a few tenths of a second cycles the first counter hundreds of times, and the second one dozens of times. When the button is released, the counters stop in one of 36 random and equiprobable states. Just after the roll button is depressed, a very brief reset pulse is generated to get the two counters into the 000 state to eliminate the disallowed states. Thus the reading is cleared before the dice roll once again, and the previous reading has no effect on the new one.

We could decode each counter into six lines and then re-encode each of the six lines into a group of pilot lights that represent the familiar spot patterns, but there is a simpler and more direct method that decodes directly into spot patterns. Since any count sequence can be used, we can call the sequence 1-3-5-6-4-2. Examination of the walking-ring timing waveforms of Fig. 8-21A reveals that an even-odd output and a 4, 5, or 6 output already are available. Two more gates and an inverter can be added to get a "NOT 1" and a "6" output. These decodings are all that are needed.

The bulbs of each die are arranged into a 7-bulb H, as shown in Fig. 8-21B. The center bulb is driven only on odd counts. Two diagonally opposite bulbs are driven on the NOT 1 counts, and the remaining two diag-



NOTE: TO DECODE "6," FEED Q_A AND Q_C TO GATE.

TO DECODE "NOT 1," FEED $\bar{Q}_A$ AND $\bar{Q}_C$ TO GATE FOLLOWED BY INVERTER.

(A) Waveforms.

(B) Display.

BULB D IS DRIVEN BY "ODD" DECODING.
BULBS A & G ARE DRIVEN BY "NOT1" DECODING.
BULBSC & E ARE DRIVEN BY "4, 5, OR 6" DECODING.
BULBS B & F ARE DRIVEN BY "SIX" DECODING.

Fig. 8-21. Decoding of electronic dice.

onally opposite bulbs are driven on the 4, 5, or 6 output only. The final two horizontal bulbs are driven by the 6 decoding. Although seven bulbs are in use, a maximum of six are permitted to light at any time, resulting in the familiar die patterns.

Convergence Generator for Color-Television Servicing

The dot-bar generator for color-television servicing is an instrument in which RTL can be used to considerable advantage. Most television service generators use unijunction-transistor, tube, or IC relaxation dividers. As a result, they need calibration and regulated supply voltages, and, worst of all, they have a tendency to be unstable and often require long warm-up periods. Further, many parts are needed for each relaxation divider stage, including expensive potentiometers and precision capacitors. The remedy is obvious: Replace the relaxation dividers with RTL scaling circuits, giving absolute stability the instant the power is applied, and this at one-half component per stage. We can now operate with ordinary high-current D cells, and obtain a totally portable, hum-free instrument.

Fig. 8-22 outlines the details in simplified form. The heart of the instrument is a 189-kHz crystal-controlled oscillator, which runs the timing chain, provides a color-rainbow gate, and produces vertical lines. A second crystal oscillator runs at 3.563795 MHz and provides color information. A final crystal oscillator, built with a high-frequency transistor, generates the rf carrier (usually on channel 2, 3, or 4). An optional 4.5-MHz sound-subcarrier oscillator may also be added.

The 189-kHz frequency is divided by six, and in the process a six-level, uniformly spaced gray-scale signal is produced with a weighted divider. The result of division by six is a signal at 31.5 kHz, which is the dividing point between horizontal- and vertical-synchronization channels. A further division by two provides horizontal sync at 15,750 hertz; a division by 35 results in horizontal bars at 900 hertz; and vertical sync at 60 hertz results from a final division by 15. A synchronizer following the 15,750-hertz divider allows horizontal bars to be produced only on even fields.

All these signals are combined in a diode rf modulator, driven by an RTL gate expander and some resistors. Horizontal and vertical sync are shaped in half monostables and are fed to the modulator continuously.

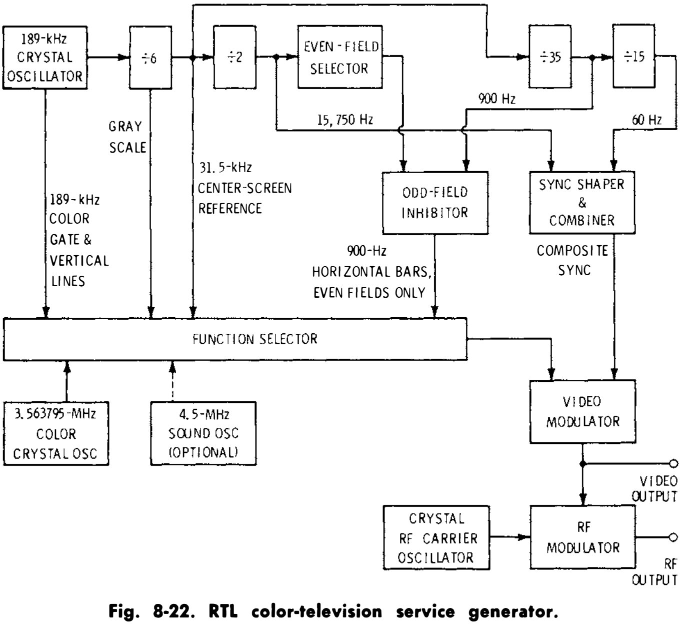
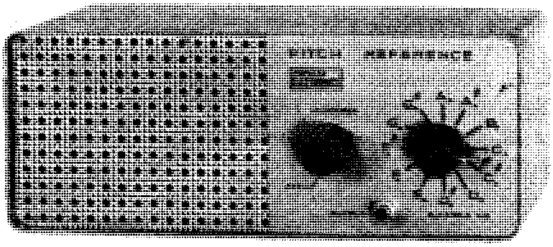

189-kHz CRYSTAL OSCILLATOR
÷6
÷2
EVEN-FIELD SELECTOR
÷35
÷15
900 Hz
15,750 Hz
60 Hz
GRAY SCALE
31.5-kHz CENTER-SCREEN REFERENCE
189-kHz COLOR GATE & VERTICAL LINES
ODD-FIELD INHIBITOR
SYNC SHAPER & COMBINER
900-Hz HORIZONTAL BARS, EVEN FIELDS ONLY
COMPOSITE SYNC
FUNCTION SELECTOR
3.563795-MHz COLOR CRYSTAL OSC
4.5-MHz SOUND OSC (OPTIONAL)
VIDEO MODULATOR
VIDEO OUTPUT
CRYSTAL RF CARRIER OSCILLATOR
RF MODULATOR
RF OUTPUT

Fig. 8-22. RTL color-television service generator.



Courtesy Popular Electronics

Fig. 8-23. Kit-type electronic pitch reference.

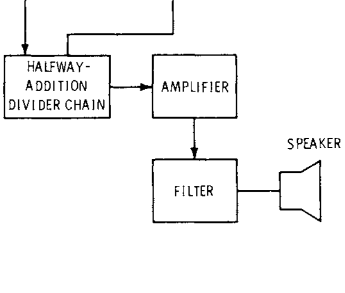
Horizontal and vertical lines are obtained by half-monostable shaping and switch selection. Combining the two in an OR circuit provides a crosshatch, and combining them in an AND circuit provides dots. Color is injected with the subcarrier oscillator, and the back porch is added with another half monostable. Additional switching makes the other service functions available from the instrument.

Musician's Pitch Reference

While many mechanical, electronic, and electromechanical tuning aids for pianos and organs exist, they are mostly quite expensive and relatively difficult to use. RTL may be used in a "hands off" frequency synthesizer that approximates the twelve notes of the middle octave to quite acceptable accuracy, and at a cost comparable to that of a good set of tuning forks. Unlike the forks, the RTL version provides a continuous output, with volume that is adjustable and pitch that is selectable and changeable at any time. One version is shown in Fig. 8-23.

The circuit (Fig. 8-24) consists of a 1.0716-MHz crystal oscillator and a halfway-addition divider chain that is switch programmable to divide by the factors shown in Table 8-1. Each division produces a square-wave output that approximates the desired frequency to an accuracy of about one-

Fig. 8-24. Block diagram of an electronic pitch reference.


1.0716 - MHz CRYSTAL OSCILLATOR
NOTE SELECTOR
HALFWAY-ADDITION DIVIDER CHAIN
AMPLIFIER
SPEAKER
FILTER

half cent (±0.03 percent), based on the equally tempered scale with $A_4 = 440.0$ hertz. The square-wave output is filtered to recover a low-distortion fundamental sine wave, which is sent to a speaker.

If desired, an offset oscillator may be inserted between the crystal reference and the divider chain. This oscillator would permit tuning slightly sharp or flat, and would let you "stretch" a piano keyboard in the same manner a professional tuner does.

Table 8-1. Frequencies of Pitch Reference

Note	Freq (Hz)	Division Ratio
C_4	261.6	4096
$C\sharp_4$	277.2	3866
D_4	293.7	3650
$D\sharp_4$	311.1	3444
E_4	329.6	3250
F_4	349.2	3068
$F\sharp_4$	370.0	2896
G_4	392.0	2734
$G\sharp_4$	415.3	2580
A_4	440.0	2436
$A\sharp_4$	466.2	2298
B_4	493.9	2170

With the RTL reference, all notes are tuned to unison, by listening only to fundamental beats that even an untrained listener can detect readily.

Desk Calculators

There are two RTL approaches to small computers, the conventional shift-register memory approach, and the newer pulse-rate computational approach.

In the shift-register approach, RTL and interface circuits are combined with MOS shift-register memories. This type of computer follows the conventional "big-machine" organization of serial storage registers, time-strobed output displays, and synchronous data transfer. Its advantages are that many decades of accuracy can be obtained, and that the addition, subtraction, and multiplication are particularly fast and easy to implement. The system lends itself well to automatic programming.

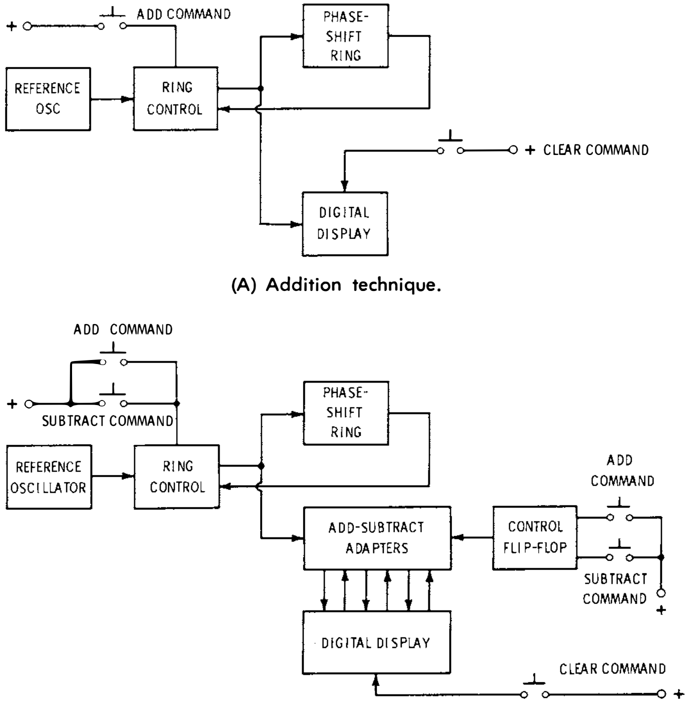
The pulse-rate computational approach is relatively new. Normally it is limited by its operating speed to four or five significant digits, but the big advantage of the pulse-rate machine is the ease and speed with which you can take square roots, square, divide, and find sines, cosines, exponentials, logarithms, and other functions that are extremely complex and time-consuming to find with the shift-register memory approach.

Details of either system would require an entire book, but we can show here the basic addition and subtraction processes involved in the pulse-rate system.

Two counters are needed. One is a conventional digital display, and one is a programmable phase-shift-ring divider (Fig. 8-25A). A reference oscillator is connected to both the programmable divider and the digital display only for the number of counts selected on the programmable divider. When the compute button is pressed, the selected number of counts is automatically added to the digital-display tally. For addition, the number is simply "dialed in" and the compute button operated.

To subtract, we need an add-subtract digital display most easily obtained with an add-only display and a string of add-subtract adapters (Fig. 8-24B). Now, in the add position, the selected counts are added to the display total; in subtract they are removed.

Thumbwheel switches can be used instead of a keyboard, providing a low-cost memory function. While addition and subtraction are not overly convenient with such an arrangement, this type of machine organization "sets up" the ability to work with the more complex functions much



(A) Addition technique.

(B) Addition/subtraction.

Fig. 8-25. RTL pulse-rate calculator techniques.

more simply than any other computer approach. For instance, all we have to do is bring out a separate lead from the phase-shift ring and the digital display, and feed each separate lead a pulse train *related* by the function of interest. If the signals on the two leads are related as x and x^2, we can obtain either the square or square root of a number. If the signals are related as cos ψ and ψ, we can find either cosines or angles. With this approach, square roots are no harder to find than squares; division is just as easy as multiplication; and so on.

The additional circuitry needed for these functions is not complex or costly. It uses only a circuit called a *rate multiplier* (or two for the sines and cosines) and some conventional counters.

AC Power Controls

The trigger circuits for semiconductor line-operated power controls are often unduly complex and expensive if made of discrete parts, and the prices of many special IC power-control modules are still too high for many applications. A single RTL IC can often replace considerable other circuitry. These circuits are particularly useful when a gate-controlled switch is to be operated with a low-level dc control voltage; when an inductive load needs continuous gate drive; when there are multiple controlled circuits, such as a theater dimmer or a home psychedelic music-lighting center; when RFI-free "zero-point" switching is needed; or when a touch-on, touch-off latch is needed.

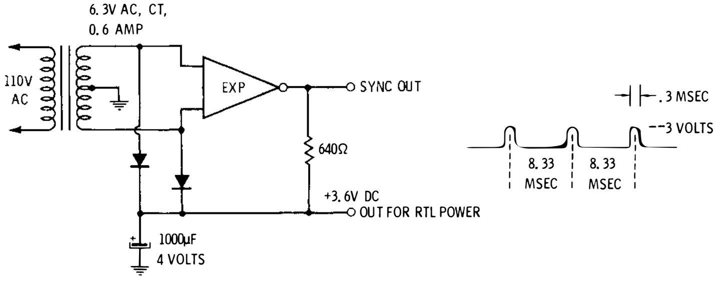


Fig. 8-26. Line-zero synchronizer circuit.

The key to the whole process is the synchronizer of Fig. 8-26. The 60-hertz line zeros are applied to a NAND gate and become narrow positive synchronizing pulses that occur only as the ac line voltage goes through zero. Thus, one quarter of an IC can substitute for a bridge rectifier and UJT assembly plus a zener, or substitute for trigger diodes.

Since the positive pulses arrive only when the line voltage is zero, they may be used to reset a ramp generator coincident with each zero, independent of the output loading or current. They also may be used directly to switch a gate-controlled device during the zero crossing for RFI-free heater operation, with a mechanical, low-current thermostat contact or other electrical control source.

A proportional load-current control is shown in Fig. 8-27. This circuit responds to a high-impedance 0-2 volt dc control signal to control the load power proportionally and nearly linearly. Operation is completely independent of the load, and the gate of the *Triac* always sees a low impedance because it is either clamped to ground before it is turned on or held to a safe and constant positive gate current after it is turned on.

The circuit makes use of a single quad gate expander. One gate generates a line sync pulse for each zero crossing (that is, one at the beginning of each half cycle of line voltage). When each pulse occurs, a second gate "dumps" the charge on a capacitor so that a ramp generator (which uses a third gate) can begin integrating downward from the supply voltage to zero voltage. The fourth gate inverts the ramp-generator output and drives the *Triac.* Note that the *Triac* is completely buffered from the ramp generator.

The higher the input voltage is, the faster the ramp runs down, and the earlier in the half cycle the *Triac* is turned on. The earlier the *Triac* is turned on, the longer it stays on, and the greater the average load current is. Thus the average load power depends on the input voltage.

Note that the RTL common return is at ac line potential, and that isolating inputs must be provided if there is any danger of shock. This, of course, is true of any SCR or *Triac* control circuit.

Other Applications

There are many other RTL applications in which it is possible to gain the same benefits illustrated in these examples—better stability, better performance, lower cost, and simpler overall design. Other potential applications include electronic games, sports-car-rally computers, magnetometers, or any other device in which we can use a combination of scaling and counting circuits, pulse generators, and logic circuits to perform a useful or novel function.

SOME SIMPLE RTL APPLICATIONS

The preceding section has shown a number of rather complex applications. But what can RTL do that is simple and cheap? Plenty. This section lists ten simple applications that lend themselves well to construction projects. These devices could be useful as service aids, test equipment, teaching machines, science-fair projects, or electronic novelties. Instead of specific circuits, only suggestions on what to build and how to go about it are given. All the circuits and information needed are in this book. By designing the circuits yourself, you should be able to construct improved, customized versions of these projects.

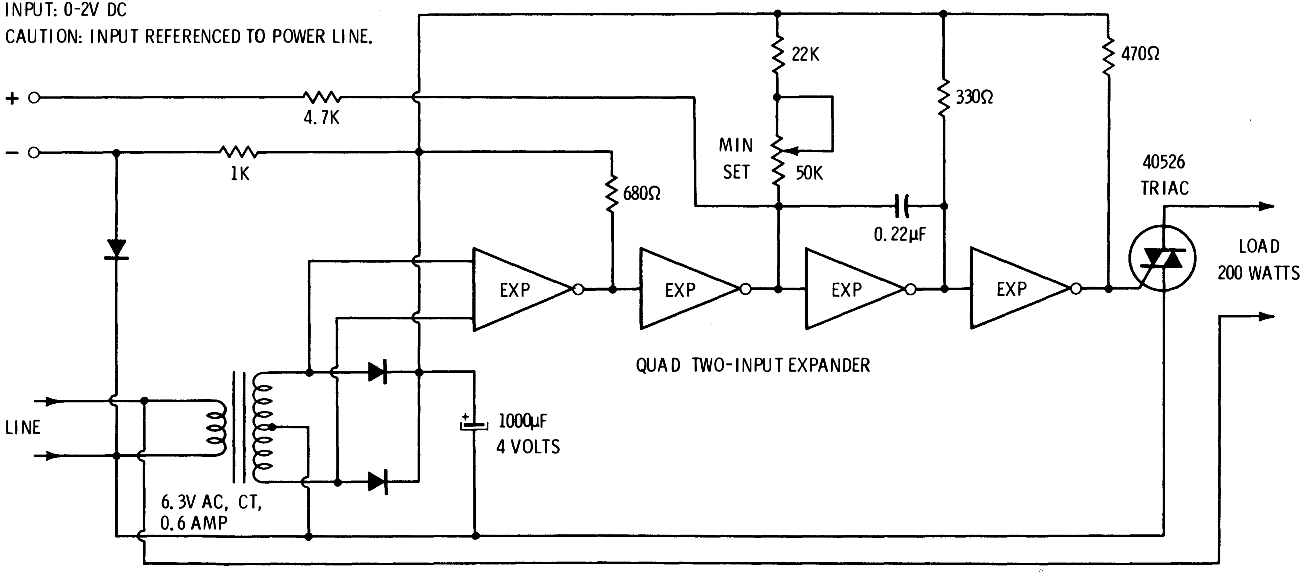
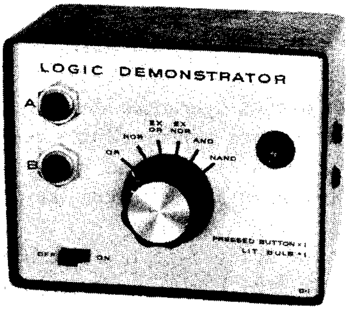


Fig. 8-27. Diagram of proportional power control.

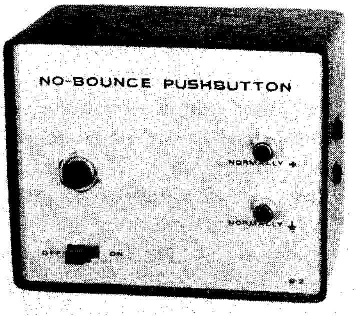
The logic demonstrator of Fig. 8-28 may be used to verify the logic relationships of Chapter 2. It can serve as a teaching aid or a science-fair project. It can be made from a quad two-input gate, a lamp driver, two push buttons, a selector switch, and two D cells. Call a pushed button or a lighted lamp a "1," and arrange the switching to get the AND, NAND, OR, NOR, EXCLUSIVE OR, and EXCLUSIVE NOR functions.

The bounceless push button (Fig. 8-29) is the most basic piece of digital test equipment. It can operate from an internal battery, and consists basically of a push button and a single two-input gate. It gives an output pulse that permits stepping any digital circuit or instrument through its paces on a one-by-one basis. If you do any digital experiments at all, this should be your first piece of test equipment.



Courtesy Southwest Technical Products Inc.

Fig. 8-28. Logic demonstrator.



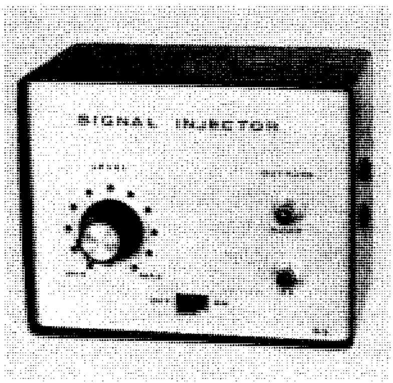
Courtesy Southwest Technical Products Inc.

Fig. 8-29. Bounceless push button.

The signal injector in Fig. 8-30 is built with a dual buffer. Cross-couple the low-level outputs and build a 1-kHz astable; the output will be a 1-kHz short-circuit-proof, adjustable audio square wave. Couple only the sharp trailing edge to the rf terminal with a small capacitor to obtain a "mosquito" type of signal injector for all sorts of radio and television service work.

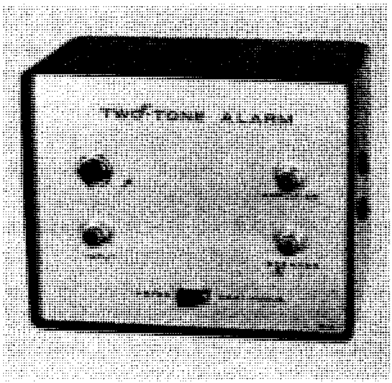
The two-tone alarm of Fig. 8-31 is made from a hex inverter and a driver transistor. It consists of two astables, one that operates at a 5-hertz rate and another that is switched between a 500-hertz and a 1-kHz rate by the 5-hertz astable. The variable tone signal that results can be used as an electronic doorbell, an alarm, an aural indicator, or a "panic" or novelty device. It can drive a speaker loudly.

There is often need for a test square wave of perfect symmetry, adjustable from 10 hertz to 100 kilohertz with controllable amplitude. To generate such a signal, use a single multipurpose IC, letting one buffer and one inverter generate signals over a 10:1 frequency range. This astable may be potentiometer controlled and operated in conjunction with switched capacitors for the decade selection. Use the flip-flop to obtain output symmetry,



Courtesy Southwest Technical Products Inc.

Fig. 8-30. Signal injector.



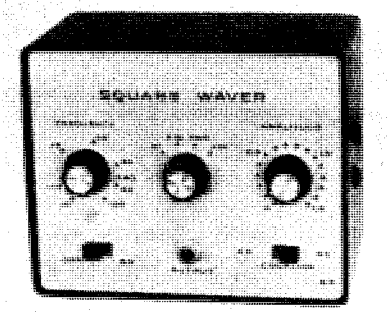
Courtesy Southwest Technical Products Inc.

Fig. 8-31. Two-tone alarm.

and the other buffer to achieve short-circuit-proof operation. An output blocking capacitor can be switched in or out to give a balanced or a positive-only square wave. The circuit easily drives a speaker, and it can be powered from two internal D cells. The finished unit might look like the one in Fig. 8-32.

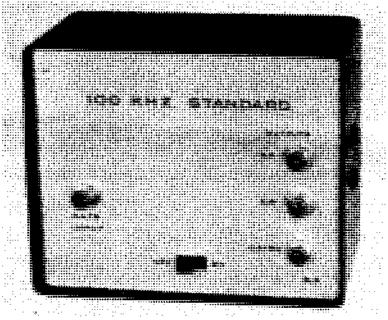
To build a 100-kHz standard such as the one in Fig. 8-33, take one 100-kHz crystal, a trimmer, and a quad two-input gate. The resulting general-purpose frequency standard can be calibrated against WWV and used as the basis for an electronic stop watch or for digital experiments. It can be used to generate harmonics for accurate receiver calibration. Use two gates for the oscillator, and one for an output buffer. Gate the final one for digital control, and capacitor couple the leading edge of the unkeyed output with a 50-pF capacitor to get the rf harmonics out.

Another practical application is to use the hex-inverter circuit of Chapter 4 to build up a one-IC tachometer/dwell meter combination with per-



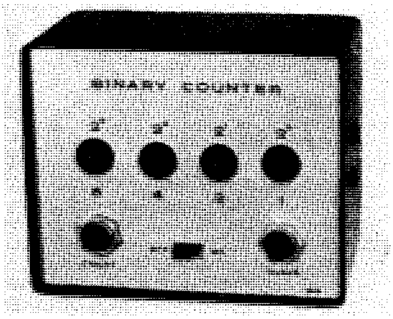
Courtesy Southwest Technical Products Inc.

Fig. 8-32. Square-wave generator.



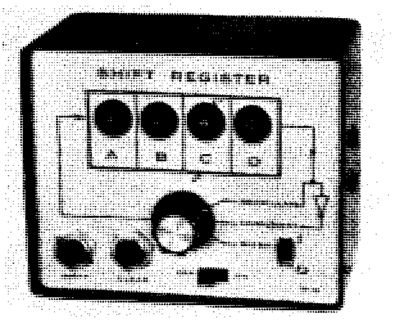
Courtesy Southwest Technical Products Inc.

Fig. 8-33. A 100-kHz standard.



Courtesy Southwest Technical Products Inc.

Fig. 8-34. Binary counter.



Courtesy Southwest Technical Products Inc.

Fig. 8-35. Shift register.

formance far better than many "simpler" (and more expensive) discrete circuits give.

If you are interested in measuring frequency only approximately, say to 3-percent accuracy, build the circuit shown in Fig. 4-15. It costs less than a counter, can have a very wide range, and can be calibrated against the power line. An ac supply is practical for this unit, since there is already a power-line connection for calibration.

The binary counter of Fig. 8-34 can serve as a good science-fair project or a good way to become familiar with the workings of a JK flip-flop. Use two dual flip-flops, connected as a binary divider, driven from a bounceless push-button circuit made from a dual two-input gate. Use four driver transistors and four indicator lamps to show the output states.

The same parts used in the binary counter can be connected as a shift register instead (Fig. 8-35). Arrange the input to accept manually entered "1's" and "0's," to recirculate, or to complement the output. This device provides a study or teaching aid, or once again, a good science-fair project. The complementing position can be used to demonstrate walking ring counters and the disallowed-state problems and solutions, as well as regular shift techniques.

Both of these final projects require a relatively large amount of supply power. Either a line-operated supply or extra heavy-duty alkaline D cells should be used with them.

Index

RTL
Cookbook

Over the past decade, digital integrated circuits have been brought to a high state of development in order to meet the requirements imposed by modern electronics technology. These devices have paved the way for many of the complex electronic marvels of our age. At the same time, they have made possible improved performance, easier design and construction, and reduced cost of many everyday electronic devices.

This book, concerned with the family of digital-logic integrated circuits called Resistor-Transistor Logic, or RTL, has been designed to provide information that can be put to practical use in the building of electronic devices. Applications ranging from simple switching through digital measuring circuits are covered fully, yet without involved mathematical developments. The author has been careful to point out unexpected complications that might arise in the reduction of theory to practice, and where appropriate, he has included tabular design data or simple formulas. In addition to coverage of a multitude of digital applications, there is a full chapter on linear operation of RTL integrated circuits. Numerous illustrations are combined with a readable text to present the subject in a way that makes it easy to understand.

For the experimenter, this book provides an understanding of RTL ICs as they are used in everyday electronics projects. For the technician, it shows how these circuits work and how to design with them. And in addition, for the engineer, it provides in convenient form circuits that can be adapted easily for use as building blocks in larger systems.

Don Lancaster is head of Synergetics-Arizona, an electronics design and consulting firm. He has written numerous articles on electronics applications, both for technical journals and for hobby magazines. His nonelectronic interests include ecological studies, firefighting, cave exploration, and bicycling. His other SAMS books include *TTL Cookbook, User's Guide to TTL* (wallchart), *Active-Filter Cookbook, TV Typewriter Cookbook, CMOS Cookbook, The Big CMOS Wallchart, The Cheap Video Cookbook,* and *The Incredible Secret Money Machine.*