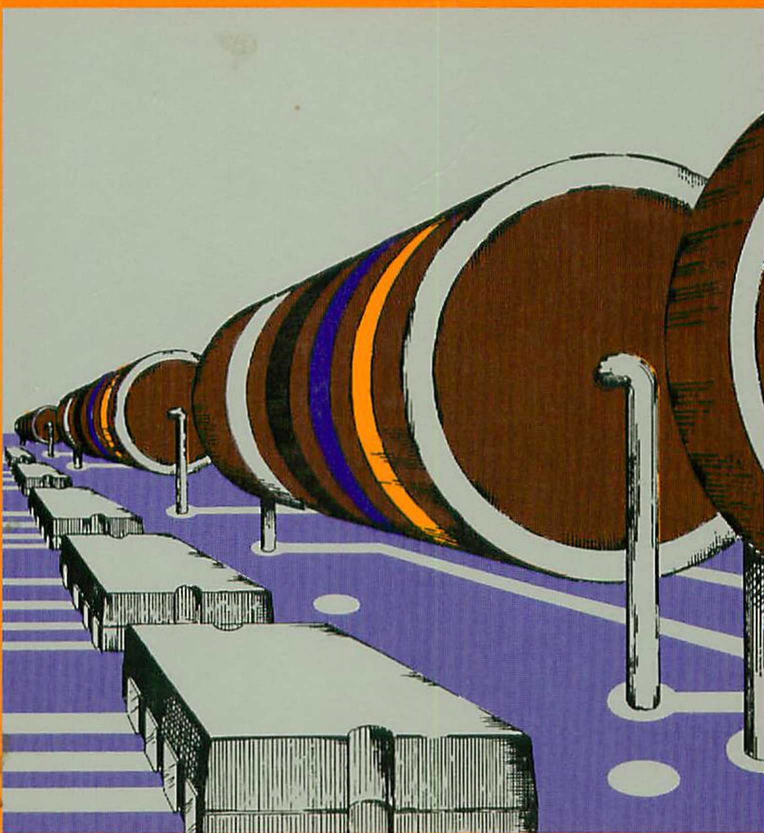
SAMS

Don Lancaster

TTL Cookbook



- TTL applications and projects
- Complete catalog of integrated circuits
- TTL circuit design tips and techniques

TTL Cookbook

By

Don Lancaster

SAMS
A Division of Prentice Hall Computer Publishing
11711 North College, Carmel, Indiana 46032 USA

International Standard Book Number: 0-672-21035-5
Library of Congress Catalog Card Number: 73-90295

95 94 26 25

Interpretation of the printing code: the rightmost double-digit number of the first column is the year of the book's printing; the rightmost double-digit number of the second column, the number of the book's printing. For example, a printing code of 92-23 shows that the twenty-third printing of the book occurred in 1992.

Printed in the United States of America.

Preface

I don't like to revise books. Correct, yes. Revise, no. So I won't.

A book becomes history the instant it appears in print. To tamper with history messes with what others and I were thinking at the time and distorts the way things were, forcing new contexts.

With the *TTL Cookbook,* it is now January 1982, and this book is eight years old. It is also one of the best selling technical paperback classics of all time. And looking around, both nothing has changed and everything has changed.

TTL is still the largest and most popular general purpose logic family, although it now shares the spotlight with an only somewhat smaller and very aggressive CMOS logic family. Most all TTL in use today is the Low Power Schottky or the "74LS" variety. The latest version of this is now called "ALS" for *Advanced Low Power Schottky,* with speed and power advantages. At this writing ALS is new enough that it has not replaced LS yet, but it is only a matter of time.

Surprisingly few new devices have been added to the original TTL list. We now have a handful of new "octal" or eight-wide registers, counters, latches, and bus drivers. These usually come in slightly longer 18, 20, 22 or rarely 24 pin narrow packages. Mercifully, a few of the older TTL device with really atrocious pinouts have been rearranged for sane PC layout. Most notable of these is the 74LS244 bidirectional bus driver, which now has a "straight thru" layout and is called the 74LS640.

The TTL bipolar Programmable Read Only Memory, or PROM, isn't used nearly as much as it was. Smaller 32 × 8 devices have been discontinued by some manufacturers, while others have sharply raised prices. Larger sizes remain available. The reason for the decline is a NMOS family of EPROMs that includes the 2716, 2732, and 2764. Besides being erasable, reusable, denser, and much easier to program, NMOS family EPROMS are now also super cheap. They seem to have run away with almost all the marbles.

Bipolar PROMS using TTL technology do still have one advantage. They are still ten times faster than the 2716, with a typical access time of 50 nanoseconds, compared to the 500 or so of a 2716. So, you might still use bipolar ROMS if speed is essential.

The *Programmable Logic Array,* or PLA, has recently taken off. A wide variety of reasonably priced devices are available, from *Monolithic Memories, National,* and others. These combine gates and registers in a you-program-it array that meets your custom logic needs.

Advantages are far fewer packages and simplified stocking. These PLA's have largely replaced data selector logic designs.

Another thing that hasn't changed is that a thorough understanding of TTL remains central to understanding and using just about everything new and modern in electronics. TTL is very much mainstream and will stay that way for a very long time.

What has changed explosively, though—in case you have been asleep for the last five years—is the microprocessor revolution. Very simply and very bluntly, if it is electronic and doesn't have a microprocessor in it, it is not worth doing. And that applies to ALL electronics, radio, radar, television, communications, power control, hi-fi, video disks and recorders, appliances, cameras, pagers, satellite receivers, data acquisition, process controls. Everything electronic. You name it.

Very few people are dumb enough to still go out and design an electronic something by taking a bushel basket of TTL or other general purpose logic family and then hand-wiring everything together. Today you do everything you can with changeable software, making any remaining hardware very general and flexible.

What does this do to TTL?

Almost all microprocessor systems need some TTL as "glue" to hold them together. Many of the exciting new peripheral add-ons for micros extensively use TTL. These new microprocessor support applications are so exciting and so widespread that they are now the dominant use of TTL chips.

While micros have dramatically increased the use of TTL and made an understanding of TTL even more essential than before, they have also very much narrowed the focus of what is important in TTL and what gets used how. Today, we use TTL bus drivers, registers, latches and low level logic gates extensively with micros. We have to thoroughly understand latches and D-flops, and the logic of handshaking communications between micros and the outside world. We also see new and special TTL devices to handle such things as dynamic refresh and memory management.

But we no longer worry too much about how to build a divide-by-37 counter, since this is better done with software. We no longer would build a TTL clock, digital voltmeter, or frequency counter, except as a student exercise, since you can get ready-to-go chips that do the entire job for you. And building your own TTL central processing unit for a computer, when you can buy an entire microprocessor for $3, is lunacy. Unless, of course, you are the military or into extreme speed.

When I wrote the *TTL Cookbook,* I purposely changed some of the pin callouts on some devices to clarify their use to an absolute beginner. For instance, "1", "2", "4", and "8" was used instead of "A", "B", "C", and "D", "OUT" was used for "Y", and so on. Enables were called enables, even if they were really a strobe or a chip select, and so on. Some uses of some pins were also simplified. On the whole, this has eliminated a lot more confusion than it created, and I am still glad I did it, and no, I won't change things. Same goes double for my not using negative logic symbols.

Taken together, the *TTL Cookbook,* and its companion *CMOS Cookbook* (SAMS 21398) remain essential for your understanding both the hardware background and the technical fundamentals driving the microprocessor revolution.

Rather than add new devices to these books, watch for a new volume in the works called *Don Lancaster's Micro Cookbook, Volume III,* which shows you devices from all the important logic and support families, similar to chapter two of the TTL cookbook.

Don Lancaster
January, 1982

DON LANCASTER heads *Synergetics,* a new-age design and consulting firm involved in microcomputer applications and electronic design. He is well known as the author of the classic *CMOS* and *TTL Cookbooks.* His many other books and hundreds of articles on personal computing and electronic applications have set new standards as understandable, useful, and exciting technical writing. Don's other interests include ecological studies, firefighting, cave exploration, bicycling, and tinaja questing.
Other SAMS books by Don include *Active Filter Cookbook, CMOS Cookbook, Cheap Video Cookbook, Son of Cheap Video, TV Typewriter Cookbook, The Hexadecimal Chronicles, Don Lancaster's Micro Cookbook,* and *The Incredible Secret Money Machine.*

Contents

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

APPENDIX

CHAPTER 1

Some Basics of TTL

A *digital logic family* consists of a group of integrated circuits or other elemental, compatible blocks that can be combined in various ways to perform a series of "yes-no" decisions based on the presence or absence of "yeses" and "nos" on various inputs, and possibly taking into account the history of previous "yeses" and "nos" gone before.

Depending on how you interconnect these logic blocks, you can build a computer, a calculator, an electronic music system, a digital voltmeter or counter, a television terminal readout display, a color-tv dot-bar generator, educational demonstrators, or any of thousands of other possibilities. While a single "yes-no" decision by itself usually is not too useful, the proper combination of grouped "yes-no" decisions taken together can represent a number, a word, a command, a musical note, a test signal, or practically anything else you might like.

One digital-logic family that is extremely popular today is called *Transistor-Transistor-Logic*, otherwise known as TTL, T^2L, or "Tee Squared Ell." Important advantages of TTL are its low cost (as low as 30¢ per package in single quantities on the surplus market), its high-speed capability (20 MHz typical; to 125 MHz with special devices), its moderate drive capability and noise immunity, and the industry-wide availability of hundreds of different devices. This gives you a broad selection of simple and elaborate logic blocks that may be directly interconnected to produce more unique functions with fewer packages than virtually any other present logic system.

TTL started out as many device lines by many manufacturers. Today, the bulk of the available and useful devices are in a 5400–7400 numbering system, originally pioneered by Texas Instruments, but now an industry standard. The 7400 line is lower in cost and is useful over a 0- to 70-degree C environment. The 5400 line is a premium military line good from −55 to +125 degrees C. There are a few important non-7400 TTL devices, particularly in the Motorola MC4000 MTTL, the Signetics 8200, and National Tri-State® product lines. Some of these devices offer unique advantages and are widely second-sourced, but even these are beginning to pick up 7400-type number equivalents as they become more of an industry standard, (on page 39).

There is a tendency to assume that most integrated-circuit logic families are pretty much the same, particularly if they work on the same +5-volt supply. This simply is not so, and each logic family has some unique characteristics and use restrictions that are all its own. In the case of TTL, the first-time user is often surprised that the outputs of his circuits never seem to get much more than halfway up to the +5-volt supply, and that unconnected inputs pull themselves up to a positive value. Further, he finds it is very difficult to get the inputs down to ground where they belong for a "low" input condition, unless whatever is pulling them down can *sink* a reasonable amount of current. It turns out there are very good reasons for these design quirks of TTL; most of these peculiarities are necessary for reliable, high-speed operation. Let us take a closer look at a typical TTL device and see what the basic input-output characteristics are.

THE TWO-INPUT, POSITIVE-LOGIC NAND GATE

Fig. 1-1 shows the internal schematic of one-quarter of a 7400 NAND gate. This is one of the most versatile building blocks in the TTL line. In fact, you could build almost any TTL circuit with nothing but this

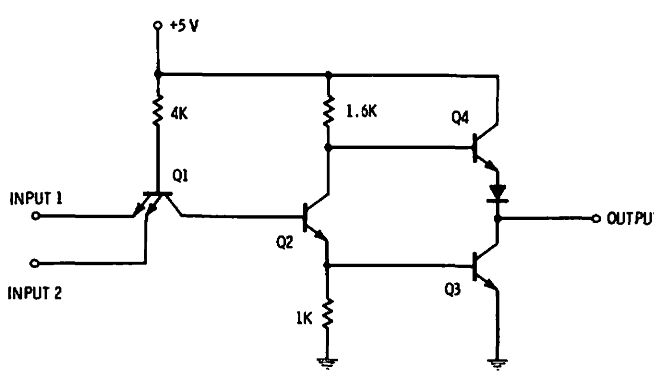


Fig. 1-1. Two-input positive NAND gate.

particular device if you had enough of them and did not mind all the extra packages. We will shortly see that this is an elemental logic block whose output is positive if either input is grounded, and whose output is grounded only if both inputs are positive.

Suppose that neither input is connected. There is no current out of either emitter of our dual-emitter input transistor, so it does not behave as a transistor. Instead, the base-collector junction is forward biased, and a current flows from the +5-volt supply through the junction diode and into the base of Q2. Q2 will turn on. This also turns on Q3. It robs Q4 of base current, so it simultaneously turns Q4 off. The output goes to within a few tenths of a volt of ground, and we say the output state is *low*. Note that in this condition, the output is capable of absorbing or sinking to ground considerable outside-world current. This is why TTL is called *current-sinking* logic.

If we connect both inputs to a voltage that is more positive than the voltage drop across a few diodes (2.4 volts to be exact), transistor Q1 still does not behave as a transistor, since the emitter-base junction remains reverse biased. While we should never leave a TTL input unconnected because of noise problems, there is no difference between an unconnected input and an input connected to a voltage greater than 2.4 volts but equal to or less than the +5-volt supply.

With both inputs positive or at a positive logic 1, the output of the gate goes to ground. What happens if we ground one input? Q1 now behaves as a transistor, and it turns on, pulling its collector near ground. This turns Q2 off, which then turns off Q3. With Q2 off, the current now can flow through the base junction of Q4. Q4 is now free to turn on. The output swings positive. Note that the output cannot reach the positive supply, for there has to be some voltage drop across the 1.6K resistor, and we get a 0.6-volt drop across the base-emitter junction of Q4 and a second 0.6-volt drop across the output diode. The output typically will only go positive by 3.3 volts or so on a +5-volt supply.

When either input is grounded, the output is forced positive. What if both inputs are grounded? The same thing happens, so the operating circuit rules for the 7400 are as stated in Chart 1-3.

We will find in Chapter 3 that this logical operation is called a positive-logic NAND or a negative-logic NOR function.

When an input is grounded, about 1.6 milliamperes of current has to flow through the grounding lead. If we attempt to ground the input through a resistance, there will be a voltage drop across this resistance,

CHART 1-3. Logic Rules for the NAND Gate

Either or both inputs grounded	Output POSITIVE
Both inputs positive	Output GROUNDED

and the emitter will not be pulled close enough to ground to let the gate operate reliably. The maximum permissible low-state input voltage is around 0.8 volt. At any input above that, the gate will either lose noise immunity or stay in its active region and possibly oscillate. Thus, any resistive connection to ground should be less than 500 ohms (with the regular 7400 TTL family), and any input-low connection must place and hold the input below 0.8 volt positive.

These input and output conditions may seem strange when compared to older logic families. There is one big benefit. When two TTL circuits are cascaded, they "look" at each other transistor-to-transistor. There are no coupling resistances or stored charges to contend with, and the logic family thus turns out to be extremely fast in operation.

A CLOSER LOOK

Circuits in the TTL logic family may be directly interconnected—the output of one to the input of one or more other packages. The drive capability of a digital logic IC is called its *fan-out*. Its input needs needs are called its *fan-in*. The voltage and current conditions needed for a medium-power TTL gate usually are normalized to a fan-in of one unit load. The average TTL gate can drive ten unit loads, and thus has a fan-out of ten. The typical TTL input has a fan-in of one, and all newer TTL devices are designed to have a unity fan-in. Logic blocks with higher fan-outs are usually called *buffers;* these typically have a fan-out of 30. Another type of logic block may be called a decoder-driver or simply a driver. Here the outputs are converted to a group of currents or voltages compatible with some outside-world device, such as a 7-segment readout or a Nixie® tube, and "fan-out" as such has no meaning as these outputs do not drive additional TTLs.

If you exceed the fan-out of a gate, the noise margin first becomes impaired, and then the voltage and current swings become too small to operate all the attached loads reliably. Fortunately, it is only rarely that you have to interconnect more than ten inputs to a single TTL output.

There are usually only two steady-state output conditions on a TTL gate. The *output-low* state is within a few tenths of a volt of ground and is called a positive logic 0. It is capable of sinking considerable current, usually 16 milliamperes. The *output-high* state is usually called a positive logic 1. When it is connected to other TTL circuits, all it has to do is hold the emitters of the various inputs reverse biased. It need only provide for leakage currents. The output-high state is above 2.4 volts and well below the positive supply; 3.3 volts is typical.

Remember that a TTL output-positive logic 0 pulls to ground and sinks a lot of current; a TTL output-positive logic 1 simply provides a positive voltage a little over half the supply voltage.

If we ever want the full power-supply voltage as an output, we can use the circuit of Fig. 1-2, where we add a 2.2K pull-up resistor. This is never needed in a TTL-only circuit, but is handy if we are interfacing some other logic family or want the full supply swing for outside-world circuits. Without the resistor, a positive logic 1 is about 3.3 volts with a +5-volt supply.

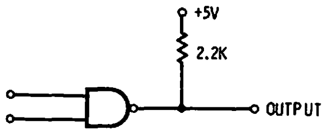


Fig. 1-2. A pull-up resistor may be added to a TTL gate if the full supply-voltage swing is wanted as an output.

Another unique feature of TTL that almost always causes trouble if it is not understood is the way in which the output switches. In the output-low state, only the bottom one of the *totem-pole* pair of transistors conducts. In the output-high state, only the top transistor in the totem-pole output conducts.

During the transition from low to high or vice-versa, both transistors conduct heavily. The instantaneous current is perhaps ten times the normal supply current. This speeds up the switching time of the output stage nicely, but at the same time, it pulls a large *current spike* out of the power-supply lines. This spike could be a fraction of an ampere or more and typically lasts 10 nanoseconds.

Unless a good bypass capacitor is provided immediately beside the IC, this current spike can raise havoc with other ICs in the system! This is an inherent feature of TTL and is one of the key sources of its speed. We will be looking at some bypassing rules soon when we talk about power supplies. Another inherent feature of the current spiking and high-speed potential taken together is that TTL can and usually will behave erratically in "rats nest" breadboarding or in traditional "perf-board" construction unless extreme care is taken with layout and conductor shapes and sizes.

The TTL output swing is from a few tenths of a volt of ground with a large current-sinking capability up to a positive voltage somewhere above half the supply voltage that needs only hold subsequent inputs positive and provide a very small leakage current.

The TTL input swing is also two-state. In the low condition, we have to sink 1.6 milliamperes to ground, and regardless of how we do it, we have to guarantee that the emitter voltage is 0.8 volt or less. In the high condition, all we have to do is hold the input positive above 2.4 volts and provide a very low leakage current. An unattended, unconnected TTL input will pull itself high to the positive logic 1 condition, but will be susceptible to noise and ringing. Good practice calls for an unused input to be tied to +5 volts or tied to a logically similar input. (The military practice of tying to +5 volts via a 1K resistor is usually not justified by the benefits and causes more problems than it solves.) As a practical matter, TTL inputs can usually be floated in a breadboard circuit if there is no lead connected to the package pin, but in final circuit versions, all inputs must be properly terminated.

A few TTL output structures differ from the totem-pole type we have just discussed. One version is called an *open collector* system. A second is called the *Tri-State®* logic system. These are in the minority and are covered in more detail in Chapter 3.

There are also some protective diodes placed on the inputs to all newer TTL gates. When these are added, the actual 7400 schematic looks like Fig. 1-3. Most TTL gates have input clamping diodes as

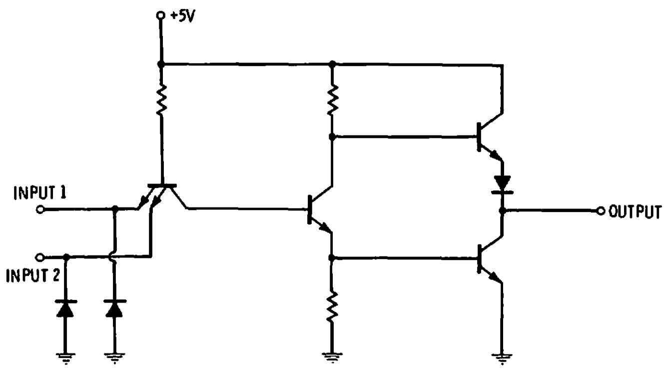


Fig. 1-3. Protective diodes connected to input gates.

shown. The purpose of these diodes is to prevent high-frequency ringing, particularly when longer leads or sharp rise inputs are applied. They improve the high-frequency noise immunity considerably. By the same token, they and the rest of the IC can be immediately destroyed if an input is brought below ground by more than 0.6 volt from a low-impedance source. This can be a problem in interface circuits. Preferably, the inputs should never be allowed to go below ground. If they must be below ground in your particular circuit, be sure to limit the reverse bias current to 10 mA or less with external series resistance. Avoid reverse biasing, if at all possible.

OTHER LOGIC BLOCKS

While the two-input NAND gate is an extremely versatile logic block, there are hundreds of others available. A one-input gate is called an *inverter* or a *noninverting buffer* or *driver*, depending on whether the output follows or complements the input. In complementary circuits, the output will be a zero when the input is a one and vice versa. Other two-input gates are available that perform different logic on the two inputs. These include the AND, OR, NOR, NAND, and EXCLUSIVE OR gates and are detailed in Chapter 3. We can have more than two logic inputs, with three-, four-, and eight-input NAND gates being common.

Gate-only circuits usually operate immediately and without regard to the history of the ones and zeros fed to the device. If history is important, we get into memory and counting devices, such as *JK Flip-Flops, D Flip-Flops, Latches,* and so on. Gates and the simple flip-flops are usually called SSI, short for *small-scale integration.*

One step beyond SSI is MSI or *medium-scale integration.* Here gates and flip-flops are suitably interconnected to produce system blocks that are called shift registers, counters, adders, decoders, data selectors, distributors, and memories. When an MSI device is aimed at one specific special-purpose task, it will have some functional name instead, such as a parity generator, a rate multiplier, a priority encoder, or a decoder-driver. The majority of TTL devices are MSI.

LSI stands for *large-scale integration* and is usually reserved for entire systems on a chip. The only common TTL LSI chip is the 74181 Arithmetic Logic Unit, used for central calculations in a computer or calculator.

In Chapter 2, we will take a closer look at the more popular TTL devices and see what they do and how to use them. One of the big benefits today in TTL is the incredible variety of MSI devices available.

PACKAGES

Some TTL is available in Mil-spec ceramic flat packages. These are expensive and very hard to use. The majority of devices are available in the common 14- and 16-pin DIP or dual in-line package, where plastic and more expensive ceramic versions are common. A very few TTL devices have too many input pins for the 16-pin package; these are reserved for the larger 24-pin DIP.

The pin conventions of the three common packages are shown in Fig. 1-4. The supply connections usually, but not always, go on diagonally opposite pins. On a 14-pin package, pin 7 is ground and 14 is +5 volts. On a 16-pin package, pin 8 is ground, and 16 is +5 volts. Pin 12 is ground and pin 24 is +5 volts on the 24-pin package. There are just enough exceptions to these supply rules, particularly with early devices, that you should double check each device before wiring it.

Usually, more than one logic block goes into a package. For instance, a quad 2-input gate contains four separate 2-input gates in a single package. A dual 4-input gate holds two separate 4-input gates in the same package. Except for the supply and ground connections, the individual blocks are completely independent. They may be used

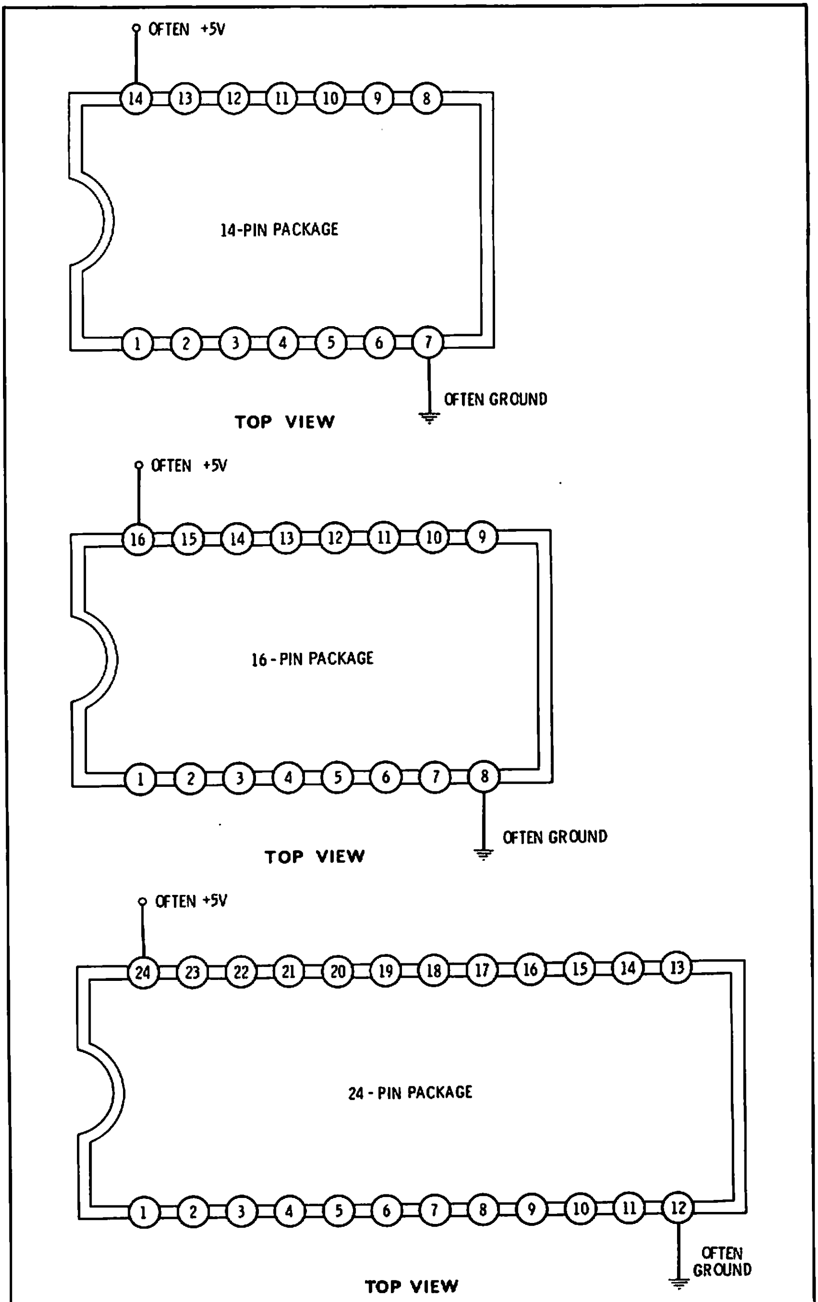


Fig. 1-4. TTL package numbering.

together or in totally separate circuits. Other common examples are hex inverters with six 1-input gates per package, and dual counting flip-flops, which contain two individual bistable logic blocks that can count and shift as well as remember data.

TYPES OF TTL AVAILABLE

There are several different *subfamilies* of TTL that trade off speed, power, and additional complexity for special uses. The TTL we have talked about so far is usually known as *regular* TTL. The other subfamilies are called *Low-Power* TTL, *High-Power* TTL, *Schottky* TTL, and *Low-Power Schottky* TTL. Table 1-1 compares the typical characteristics of each subfamily, while Table 1-2 shows the usual part-numbering system to identify a family of a given temperature range. Finally, Table 1-3 is a rough "rule of thumb" for the tradeoffs you can expect in terms of speed and power as you change subfamilies.

Table 1-1. Typical Characteristics of TTL Subfamilies

Family	Gate Propagation Time	Power per Gate	Maximum Counter Frequency
Regular TTL	10 nanoseconds	10 milliwatts	35 megahertz
High-Power TTL	6 nanoseconds	22 milliwatts	50 megahertz
Low-Power TTL	33 nanoseconds	1 milliwatt	3 megahertz
Schottky TTL	3 nanoseconds	19 milliwatts	125 megahertz
Low-Power Schottky TTL	10 nanoseconds	2 milliwatts	45 megahertz

Table 1-2. TTL Subfamily Numbering

Family	−55° to +125° C	0° to +70° C
Regular	5400	7400
High-Power	54H00	74H00
Low-Power	54L00	74L00
Schottky	54S00	74S00
Low-Power Schottky	54LS00	74LS00

Regular TTL

Regular TTL is normally the widest available and the lowest-priced type of TTL, and it has far and away the greatest variety and second-sourcing. A typical gate-propagation time is 10 nanoseconds; this is the time it takes for a logic change at a gate input to appear as a logic change on the output. Around 10 milliwatts per gate is needed, and counting flip-flops go as high as 35 MHz.

Such devices as the Motorola MC4016 and MC4018 and the Signetics 8280 and 8288 counters were originally "non-7400" devices. They now have 7400 equivalents and are *essentially* identical to regular 7400 TTL. They may have a slightly different fan-in requirement and the output positive-voltage swing may be slightly different, but the

Table 1-3. A Comparison of TTL Subfamilies

Family	Speed	Power
Regular	X1	X1
High-Power	X2	X2
Low-Power	X1/10	X1/10
Schottky	X3.5	X2
Low-Power Schottky	X1	X1/5

devices can usually be freely mixed with traditional 7400 devices. Fan-out is normally ten regular TTL devices.

Low-Power TTL

Low-power TTL exchanges power consumption for speed and is identified by an *L* in the part number. For instance, a 74L00 is a low-power, commercial version of the 7400 regular TTL NAND gate. There is roughly a 10:1 tradeoff in the low-power version—$\frac{1}{10}$ the speed to counters at $\frac{1}{10}$ the power, although the simpler gates run $\frac{1}{4}$ the speed on $\frac{1}{10}$ the power. Flip-flops and counters have a maximum toggle frequency of 3 MHz or so. Within the low-power subfamily, the fan-out remains ten, but a low-power TTL gate can drive only one regular TTL gate. While the 54L00 and 74L00 series TTL do offer low-power consumption, many of their advantages are being pre-empted by the CMOS logic families, particularly the RCA 4000 series COSMOS and the Motorola MC14000 series McMOS lines.

High-Power TTL

The high-power TTL devices are designated with an *H* in the part number. 74H00 is the equivalent of a 7400 gate, and so on. Typically you get twice the speed for twice the power. Counters are good to 50 MHz. Within the high-power subfamily, the fan-out remains at 10, but the fan-in is typically 1.3 times regular TTL loads. Thus, a regular TTL gate can drive at most only 7 high-power TTL inputs. High-power TTL is largely being replaced by the newer Schottky TTL which is faster and draws less supply power. Quite a few high-power devices remain available. One advantage they do have over the Schottky devices is that the outputs are "quieter," a handy feature in high-speed digital-to-analog converters, but hardly useful elsewhere.

Schottky TTL

Schottky TTL is an improved version of TTL that has a better speed/power tradeoff than the older types. To do this, Schottky diodes (a fast diode with a 0.3-volt forward drop) are placed across most of the transistors in the basic TTL gate. This prevents the transistors from saturating and thus eliminates any storage-time delays inside the transistors. The part numbers have an *S* in them, as in a 74S00. Propagation delays of 3 nanoseconds are combined with flip-flops that can run at 125 MHz.

Where high speed is essential, Schottky TTL is a logical choice. Its competitor is MECL and other emitter-coupled logic families which in general are much faster, but considerably more difficult to use.

A high-speed, unsaturated logic family such as Schottky TTL presents serious restrictions in the type and quality of test equipment you must have to work with it intelligently. A 60-MHz triggered oscilloscope is essential and a 120-MHz one is preferable. As might be expected, Schottky devices are much more critical as to layouts and supply decoupling than ordinary TTL because of their higher speed. Nevertheless, where high speed is essential, they are often the simplest solution to system problems in the 30- to 120-MHz range.

Low-Power Schottky TTL

Devices such as the 74LS00 are emerging as a more recent variation on TTL. The low-power Schottky TTL family is slightly faster than regular TTL, but requires only ⅕ the power. It does this by using the Schottky diodes to eliminate storage-time effects, but then raises the circuit impedance levels to slow things down to normal and pick up power savings. For many applications, this represents a near-optimum combination of values. Being newer and inherently more complex than regular TTL, the LS series is higher priced. As of this writing, it also does not have as extensive a variety of devices or suppliers as the regular 5400/7400 series TTL does.

Which Family?

While all these variations of TTL are available as design options, the regular-power traditional TTL remains the cheapest and usually the widest available choice. Very often, it is also the best overall design choice. 74S is essentially replacing 74H, and 74L is being challenged by CMOS. 74LS will represent a good choice when it is cost competitive with regular TTL and has enough different devices readily available.

The best rule today seems to be to choose regular TTL unless you have a specific speed problem, and then choose 74S. Remember that for all but the simplest high-speed systems, you will need a high-