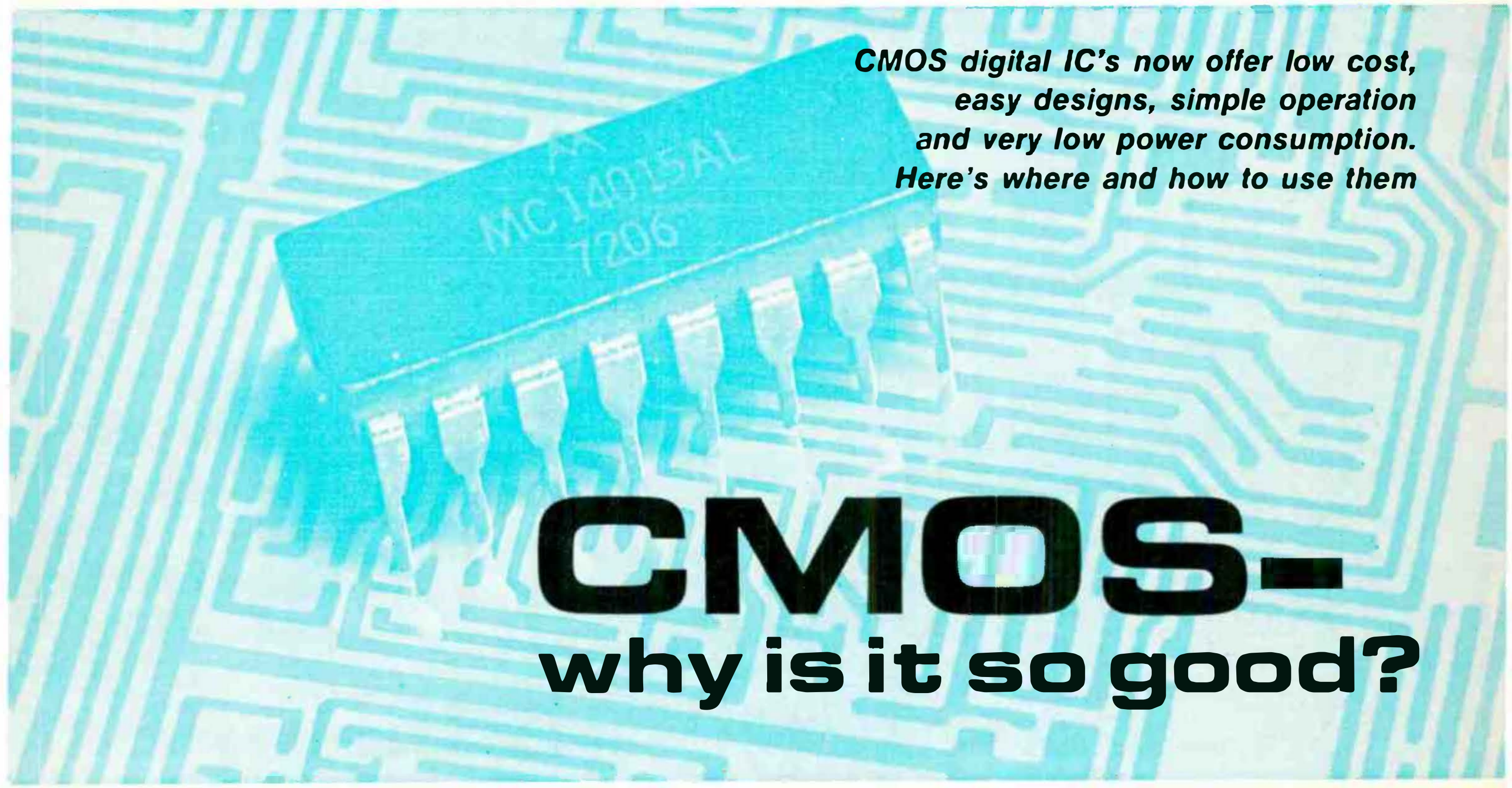


CMOS–why is it so good?

CMOS digital IC's now offer low cost, easy designs, simple operation and very low power consumption. Here's where and how to use them

by DON LANCASTER

CMOS OR COMPLIMENTARY METAL OXIDE Silicon integrated circuits have been around for a number of years. Pioneered by RCA, their high price has kept them from popular use. During that time, their micro power consumption, easy circuit design, and outstanding noise performance has been verified time and time again in many military, industrial, and critical aerospace applications.

Some things about CMOS seem almost magic. All the inputs are open circuits. As long as the IC doesn't *change* state, it draws essentially *zero* supply power. It's only when you are changing information inside the package that any power is drawn at all, and then the power is drawn only while the change is taking place. CMOS is fantastically forgiving of sloppy power supply design—it works over a 3 to 15-volt range. It slices its logic right down the middle, so it is also forgiving of noise problems. Better yet, it doesn't generate any noise of its own. Its output states look like *resistors,* either a 400-ohm resistor to + or a 400-ohm resistor to ground. Finally, as an experimenter bonus, CMOS is very easy to convert to linear circuitry, particularly crystal oscillators and electrically variable switches and attenuators.

Today, the price of CMOS has dropped to around a dollar per gate package, and around $4 for the fancier versions. Surplus is available for even less, as a quick check of the ads in the back pages of **Radio-Electronics** will verify. Yes, TTL is cheaper, but CMOS is economical enough right now for practically everything you might like to do with it. Better yet, further price reductions are almost a certainty, so now is the time to learn about CMOS and start using it.

Once you get into the designs, you'll find that so much is easier with CMOS—things like circuit and power supply design, noise performance, and your design time, that many times, *today,* CMOS is the cheapest logic you can use on a system basis.

Many CMOS manufacturers are listed in Table 1. Right now, RCA with their 4000 series COSMOS and Motorola with their 14000 and 14500 series MCMOS are leading the pack in having lots of devices widely available. The smaller companies also have many unique IC's offered. Solid State Scientific has many fancy large scale circuits, including a complete micropower clock in one package. Inselek offers ultra fast CMOS, and Harris offers a number of unique devices. National has both the traditional 4000 series devices as well as a line of CMOS with pins identical to the popular TTL 7400 series devices.

There's every indication that the 4000 series of devices will be the industry standard. (Motorola's 14000 series is identical—just subtract 10,000 from the part number). These are propably the best devices to start working with, particularly since they are likely to be available from a wide range of suppliers.

Two myths

Before we go into complete details of what CMOS is and how to use it, let's throw out two myths about CMOS.

Myth 1 says that CMOS is inherently slow. Not true. There is no fundamental reason why MOS circuits should be any slower than bipolar ones. What happens is that many products now offered happen to have higher impedances, trading off supply power for speed of operation. Even so, you can easily run to 5 MHz with the majority of CMOS circuits (except for some specials). One type of CMOS, called SOS for Silicon-on-Sapphire and offered by *Inselek* runs as fast as ordinary TTL, and in fact is the *fastest* logic available *anywhere* on a speed-power basis. Since the majority of circuitry in use runs slower than a few megahertz, particularly experimenter circuits, the trick is to use CMOS where you can and save the high-speed stuff for other families if you have to use them.

Myth 2 says that CMOS is static sensitive and very hard to handle. Again, not true. Virtually all newer CMOS circuits are internally protected six ways from Sunday with resistors and Zener diodes to eliminate any possibility of static damage. A little bit of common sense handling advice still remains—we'll see about it in a minute—, but this is a strictly "just-in-case" type of precaution, and its highly unlikely that you will ever damage a CMOS integrated circuit, unless you are extremely careless.

As with any logic family, there are certain rules of the road that you have to follow in order to get reliable operation. Familiar examples are the tight power supplies and the many 0.01 μF high frequency bypassing capacitors needed to get TTL to work, and the waveform cleanliness and fast falling edges needed with RTL. With CMOS, there are only two precautions you have to worry about. All inputs must always go somewhere and cannot be left floating. Secondly, if you connect a low-impedance piece of test gear to a non-working (power off) CMOS circuit, you could conceivably hurt one of the input diodes. Common sense takes both problems in stride.

Some basics

Before we find out all about CMOS, maybe we'd better review some basics of what digital logic is in the first place. A digital integrated circuit performs simple yes-no or "one-zero" decisions. It provides a "yes" or "no" output or outputs in response to a group of "yes" or "no" commands on its various inputs. Groups of these "yes-no" commands can represent calculator numbers, computer words, or alphanumeric messages. Depending on the internal complexity of a digital IC, we can get anything from a simple combinational decision to a complete calculator in a single package.

The exact value of the input voltage doesn't matter, so long as a "1" is within a guaranteed range of allowable "1" values and a "0" is within a specified range of permissible "0" values.

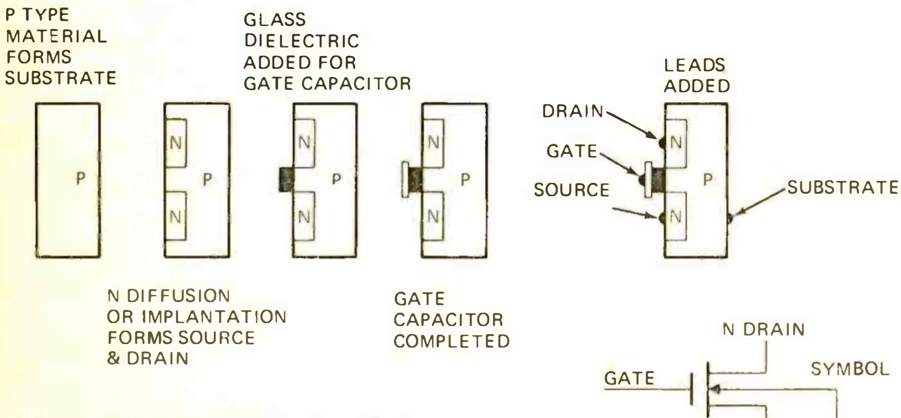


FIG. 1—A MOS TRANSISTOR begins with the substrate slab. Drain and source are added.

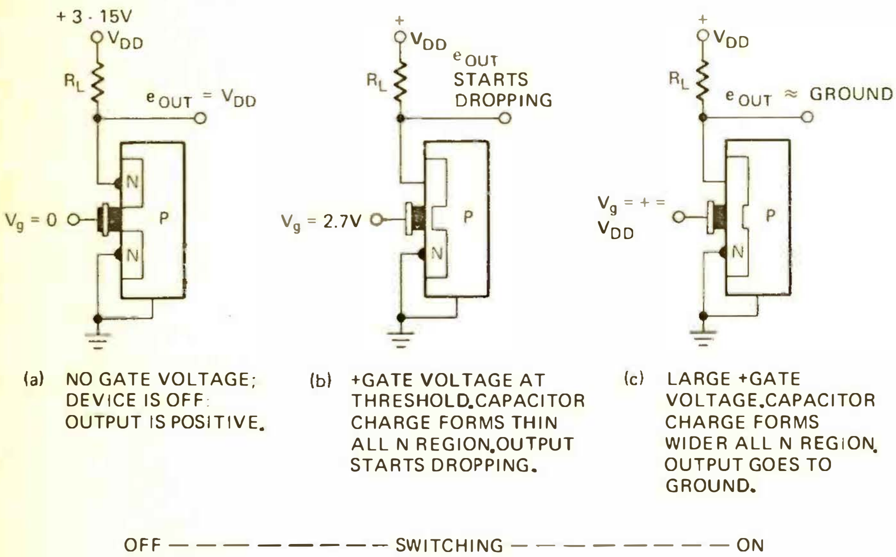


FIG. 2—THREE STAGES OF OPERATION of an n-channel device. Note that the output polarity is opposite that of the input and the device operates as an inverter.

We can usually connect one digital IC to another by direct connection. A given package has a certain drive capability called the *fanout*. Similarly, a given input has a certain load it presents called the *fanin*. These are usually normalized to one *unit* load to simplify things. With CMOS, one package's output can usually drive at least *fifty* other inputs; thus the fanout of CMOS is usually 50.

Our digital IC's also need supply power, often a single positive supply and a ground return. CMOS will work with any supply voltage from 3 to 15 volts. With some logic families, there are limits to the range and the value of the internal impedance of the supplies, particularly at high frequencies. This is particularly true in TTL where the normal circuit operation inherently returns a lot of noise to the supply. This noise must be eliminated before it can hurt the logic performance of another IC down the line.

We'll find out in a bit that CMOS wins on most of these system problems. You can run over a 3 to 15-volt range with a poorly bypassed supply and get away with it. You could, theoretically even use a 15-volt supply with 12 volts of ripple! The ranges of a guaranteed "1" and a guaranteed "0" are even nicer—about half the supply voltage for each. Thus, on a 5-volt supply, a "0" can be anything from around 2.3 volts down to ground, and a "1" can be anything from 2.7 volts up to the positive supply. The logic slices right down the middle. As you add loads to an output, the logic levels don't change like they do with other families—they simply slow down a bit, so the noise performance turns out to be pretty much independent of the loading.

Inside the package

From what we've promised you above, CMOS obviously has to be quite different inside the package than are the common logic families. Let's find out why.

We can start by building an ordinary MOSFET, as shown in Fig. 1. We'll find out shortly that the one we'll start with is an *enhancement mode, N channel* one.

To start, we take a handy bar of p-type silicon, and place two N junction regions in it through diffusion or ion implantation. We call the original bar of p-type material the *Substrate*. The upper N region is the *Drain*, and the lower region is the *Source*.

Next, we put a thin layer (an *extremely* thin layer) of glass or another *nonconductor* dielectric over the substrate between the source and the drain. Then, we add a metal or silicon contact on top of the glass, forming a capacitor. We call this new contact the *Gate*. The Gate has the ability to turn the conducting path between source and drain off and on. Note that there is no DC path from the gate to anything else in the circuit. All the leads are attached by mechanical or non-rectifying, or *ohmic* contacts. And this just about completes the device.

Turning to Fig. 2, we can hook this MOSFET up in a circuit with a load resistor going to a positive source, perhaps something between 3 and 15 volts. If we ground the gate input, there will be no charge on the gate capacitor, and the drain to substrate circuit will look like a *reverse* biased pn junction, and no drain to source current will flow. So with a grounded gate, the device is *off*.

Suppose we start to make the gate slightly positive. The capacitor will charge up, piling up holes (a lack of electrons) on the input end and piling up electrons on the substrate end. The greater the voltage, the more charge we build up.

The black magic comes in next. Since the substrate was initially p-type material, it is normally lacking some electrons, or normally has an excess of holes. As the gate capacitor starts to charge, the extra charging electrons start to accumulate immediately *under* the gate capacitor. Each electron wipes out a hole on the average, so the material immediately under the gate appears to have *less* holes than it did before we biased it. The area under the gate, which we'll eventually call a *channel* becomes *less* of a P type material than it was, and stays that way so long as the charge remains on the gate capacitor.

If we add yet more positive voltage to the gate, we pick up even more electrons under the gate capacitor, and eventually all the holes are offset by the available electrons. The material immediately under the gate capacitor now looks *intrinsic* or free of either electrons or holes in any excess. The voltage on the gate needed to exactly do this is called the *threshold* voltage and is around 2.7 volts for the 4000 series CMOS.

What if we add more positive gate voltage? Now the excess electrons start piling up since there are no more local holes to combine with. *Immediately under the gate capacitor, the substrate temporarily turns to n-type material, as it has an excess of electrons.* Now, we have all N material going from source to drain. It looks like a plain old junction-free resistor, and conducts current. The more positive the gate voltage, the thicker the N *channel* becomes, and the more current we can draw, limited only by the load resistor bottoming when its voltage drop equals the supply voltage.

Our MOS transistor is normally off, and stays off for negative or zero gate voltage. When the gate voltage reaches a threshold of 3 volts, the source to drain starts conducting and the current increases with increasing positive gate voltage. Since the devise is normally OFF, its called an *enhancement* mode unit as increasing the gate voltage enhances or increases the drain to source current. Its also called a N channel device, because the conducting channel is apparently N material when it exists.

Some features of our transistor should now be obvious. First, the input is always an open circuit, so it never draws any current except when you are charging or discharging the very small gate to substrate capacitor. The input impedance is essentially infinite. Also, when we are conducting, there are no saturated junctions or anything of this sort—all the source to drain looks like is a resistor of around 400 ohms when it is ON, and an almost open circuit when it is off.

Note further that our simple switch works backwards. Make the input positive, and the output goes to ground, and vice versa. This is called a logic *inverter.* We'll shortly see how fancier logic blocks may be built up by suitable series and parallel combinations of inverters. Obviously, if two devices are in series, *both* must be turned on to allow current to flow; if two devices are in parallel, *either* can be turned on to do the same thing.

The gate capacitor turns out to be extremely thin, and its open circuit welcomes the buildup of static electricity. The field strength from static can easily puncture the capacitor and permanently damage it. This is why there was so much static problem with early MOS devices. Practically all devices today have external Zener diodes and resistors to keep static from ever getting close to the gate capacitor.

Complimenting the MOSFET

So far, we've built nothing but a plain old N-channel enhancement mode MOS transistor driving a resistive load. When the transistor is off, the load current is zero, and the output voltage is positive. When the transistor is on, the load current is determined by the supply voltage and the resistor value, and the output drops to ground. Our circuit draws supply power only in one state.

We could use another transistor or a current source for the load resistor, and this is done in ordinary non-complimentary MOS integrated circuits such as are used in character generators, shift registers, and read only memories. CMOS does things differently

Instead of using a load resistor, we build an *exactly opposite* or *complimentary* p-channel enhancement mode transistor and use it as a load. As Fig. 3 shows us, a P-channel enhancement mode device can be connected to the positive supply with a load resistor going to ground. Ground the gate, and the transistor turns on, and the output swings positive. Make the gate positive, and the transistor turns off, and the output drops to ground. So a positive gate voltage will turn on an N-channel device, but turns off a P-channel device. A grounded gate input turns ON a p-channel device and turns OFF an n-channel device.

To build a true CMOS inverter, we simply combine a N-channel and a P-channel device in series as shown in Fig. 4. The two gates are then driven in parallel. To actually do this, takes a bit more construction inside the IC than older logic types needed. The two devices are isolated from each other by several possible means, such as diffusing a "P tub" onto a N substrate and then building the other transistor inside it, or by building everything on non-conducting sapphire or spinel.

When we tie the two together this way, we always have only *ONE* transistor turned on. The steady state output is always either a resistor to ground or a resistor to plus. In neither state is there any internal plus to ground path, and the IC magically seems to never need any supply power!

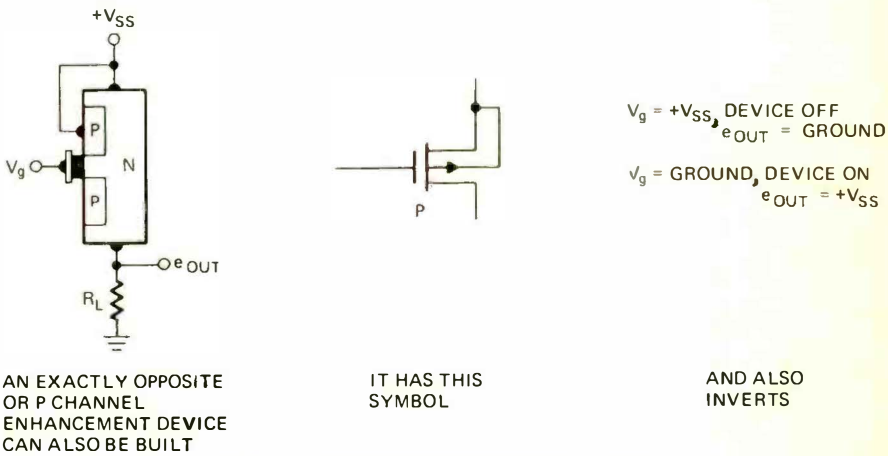


FIG. 3—THE P-CHANNEL ENHANCEMENT MODE DEVICE complements the n-channel FET in Fig. 2. Its construction begins with an n-type slab or substrate.

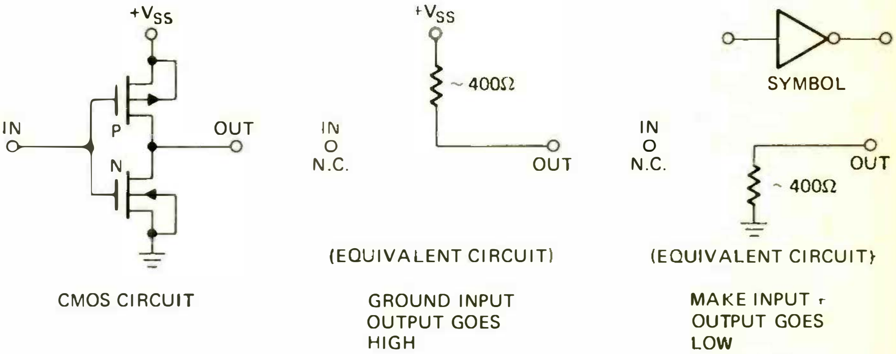


FIG. 4—THE CMOS DEVICE is made by combining two n-channel and p-channel FET's. The circuit configuration shown is that of an inverter.

Generally, the output of one CMOS stage drives the input of the next, which is an open circuit, so the load also draws no current. Apparently, we have a logic family that *never* needs any power at all. Can this be?

Obviously not, for we would have some sort of perpetual motion circuit that violates a bunch of laws of information theory and thermodynamics. It *has* to take some power to transfer information. In CMOS, this power gets used in charging and discharging the input capacitance of the next stage. During the charging time, current flows from the positive supply into the gate capacitor of the following stage. During the discharge time, the current from the gate capacitor of the stage following is discharged to ground.

The gate capacitor is a very small one If we don't charge or discharge it very often, we can literally run our CMOS off a damp blotter battery. As we increase the operating frequency however; we charge and discharge the capacitor more often and the *average* supply power goes up. When we get to 5 megahertz or so, the total supply power gets up to roughly what the other logic families need. At low frequencies, CMOS takes very little supply power and offers fantastic power supply savings. As you increase frequency, the power needs proportionately increase to the point where there is little power savings above 5 MHz. This shows why the CMOS watch circuits draw so little battery current as the majority of the circuitry changes at a very slow rate. In fact, practially all of the CMOS power is used in the first four divider stages following the crystal on a typical watch; the rest is utterly negligible.

Our basic CMOS inverter (Fig. 4) then consists of a N-channel transistor on the bottom and a P-channel one on top. Both gates are connected together. Ground the input, and the N-channel job turns off and the P-channel one turns on, and the inverter's output goes positive. Make the input gates positive, and this time, the N-channel device turns on and the P-channel device turns off, and the output goes to ground. The output of our inverter always looks like a 400-ohm resistor, either to + or to ground; the input is always an open circuit.

With a fairly high supply voltage and our simple inverter, the decision between a "1" and a "0" is usually made halfway up, a point at which both transistors are moderately conducting. As the supply voltage is lowered, the "1"-"0" decision gets a bit sloppier and wider, but still a bunch of noise immunity is offered.

Since both transistors are never simultaneously conducting very heavily, there is no current surge that gets thrown back onto the supply power line as there is with TTL and some DTL circuits. This greatly eases the power supply and decoupling design problems.

Interface and fanout

As we add more CMOS inputs to an output, only the load capacitance changes, since the gates are all open circuits. The voltage levels do not change. You can drive at least 50 gates or more with one output lead, and the noise immunity and signal levels stay the same pretty much independent of the number of new devices you hang onto an output.

The story changes a bit when you actually try to draw some load current to interface the outside world or some other logic family. While the ON resistance is around 400 ohms, there is a current limiting that takes place under heavy load or short circuit conditions, and the available source or sink current is typically a bit less than one milliampere. This is more than enough to interface regular MOS, RTL, most discrete circuitry, and *low power* TTL, but it is not quite enough to *reliably* interface regular TTL. To drive TTL, you have to use a CMOS buffer such as the MC14049 or MC14050, or another circuit that provides at least 1.6-mA output current. With a normal device, you can treat the output as a 1-mA current source or sink and proceed accordingly. Some sort of current amplification is recommended for driving LED's or lamps, while liquid crystal and some flourescent displays are directly compatible.

You can apparently short circuit CMOS continuously without harm, at least at room temperature. This is handy for electronic music keying and building bounceless pushbuttons. Other interface techniques are easy to work up.

Coming from the outside world to CMOS is a slightly different story. The open circuit inputs make things relatively easy. All you have to do is never go below ground or above the positive supply with an input. A voltage near ground will be read as a "0" and near the positive supply will be read as a "1". With a +3.6 or +5 volt supply, you can directly interface DTL or RTL. With TTL, the output guaranteed "1" is usually only half the supply voltage, so a simple pullup resistor of 2.2 to 10K should be added. *NEVER let the input go above positive or below ground* particularly from a low impedance, as we'll shortly see that this can hurt CMOS—in fact its about the only way you can really damage it.

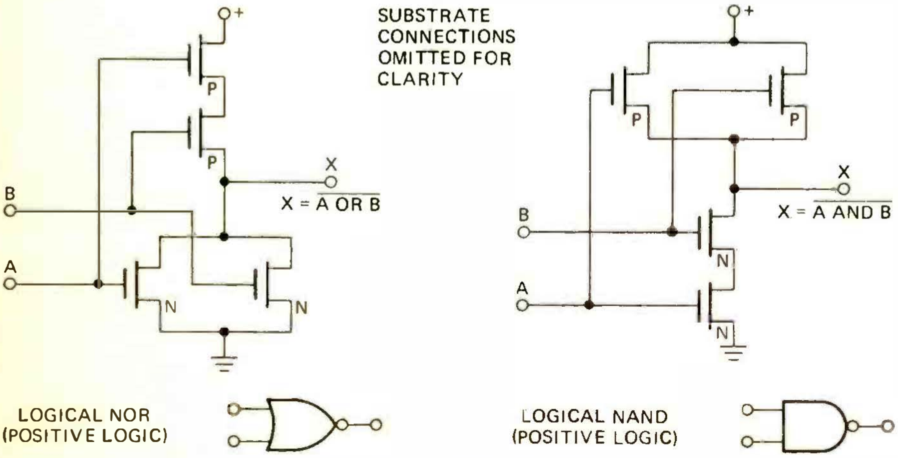


FIG. 5—MORE COMPLEX CMOS LOGIC is easily built up by combining p- and n- channel devices. Shows are connections used to form logic NOR and NAND circuits.

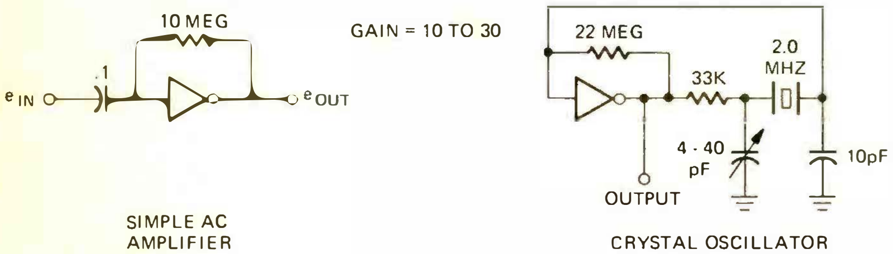
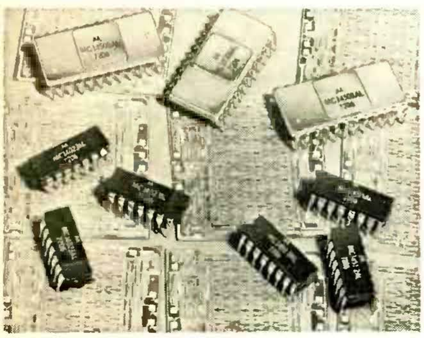


FIG. 6—LINEAR APPLICATIONS OF CMOS need only the addition of a feedback resistor. Here are two examples—a simple ac amplifier and a crystal oscillator.



VARIETY OF MCMOS IC's made by Motorola. Many varieties are currently available.

Building some logic

There's really not too much you can do with just inverters, although they do have a few handy applications. Fig. 5 shows how we can combine series and parallel arrays of MOS transistors to perform logic. In the NOR circuit of 2A, making the inputs both positive pull the output to ground. So does making either input positiive. Only when both inputs are grounded, does the output swing positive. This is known as a positive logic NOR circuit. To build the familiar OR circuit, you simply add an inverter to the output—making either input positive gets us a positive output.

The series combinations present as real problem, for since they all have to be on anyway to conduct current to an output load, the sequence in which they turn on doesn't matter, and an on device has no voltage drop across it, or at least very little. One minor effect is that the threshold voltages will shift slightly for the differing transistor positions. This is a minor effect and is detailed on most data sheets.

The NAND circuit is an upside down NOR one. Both inputs must be positive to force the output to ground. Add an inverter and we have an AND circuit in which both inputs have to be positive to get a positive output.

More complex logic is easily built up with proper series and parallel combinations. Two NAND gates back-to-back form a set-rest flip-flop. These may be cascaded with a CMOS circuit called a *transmission* gate to form a *master-slave* flip flop which in turn can be used for binary division, decimal counting, and all the more familiar MSI logic applications.

Table II is a more or less random selection of the hundred or so CMOS integrated circuits available today. These will give you an idea of what is on the market and may represent a good choice for initial experiments. Two devices that are particularly interesting that have no equivalents in the older logic families are the MC14016 and MC14046. The former is a quad switch. It can be used for digital or analog signal transmission, and it doesn't matter what you call the input or the output, since the ON equivalent circuit is a resistor and the OFF equivalent circuit is nothing. Tie four of these together, and you can just as easily select one of four input signals and route it to a single output, or use one input signal to go to zero, one, two, three or four places at once. Thus, with this CMOS package, there is no difference between a data selector and a data distributor.

The MC14046 is a phase lock loop circuit; unlike the older PLL's this one will work and track and lock over a 1000:1 frequency range, making it a top contender for electronic music, digital tachometers, frequency multipliers, and things like that. Like many of the older PLL's, it has a maximum frequency of 500 kHz.

Some precautions

As with any logic family, there are several things to watch out for to keep out of trouble. These are surprisingly easy and simple with CMOS.

Rule 1 is that *all* inputs must go somewhere. This can be either to a logically similar input or connected to positive or ground as needed to get the right function. The reason is simple—a floating input is an open circuit that can pick up hum and

(continued on page 88)

CMOS
(continued from page 36)

noise and wander around from a one to a zero and back again. If you find an experimental circuit that works for a few seconds or maybe even half a minute and then quits, chances are there is a floating input messing things up.

Rule 2 involves connecting test equipment. If you ever apply test signals from a low-impedance generator to a turned OFF piece of CMOS (One with the supply power removed), you can drive the protecting diodes into conduction. Get above 50-mA through the diode and you kill the IC. The way around this is to make sure that no input or output can deliver more than 10-mA or so under short or reverse supply conditions: this will protect everything. A good practice is to leave a K resistor in series with all input test signals, particularly if they come from a low-impedance source.

Rule 3 involves a CMOS bug that is being eliminated in newer designs. Its called SCR latchup, and can be caused by a momentary input signal transient or reverse polarity connection. The whole IC literally turns on as a silicon controlled rectifier and draws a bunch of current—like half an amp or more. If you can shut things down in time, there is no damage done; otherwise the chip self-destructs. This normally would only happen if you were very sloppy about testing—if a piece of CMOS ever feels hot, *disconnect power IMMEDIATELY,* and things should get better. With a reasonable amount of care in your experiments, this will never happen.

Rule 4 says to not go out of your way to static damage the devices. Most IC's are properly protected against normal handling and in-circuit operation, but go along with the game anyway. Keep unused devices in their protective foam or aluminum carriers. Return them to conductive foam afterwards. *NEVER USE STYROFOAM to store* CMOS! Don't solder CMOS into a circuit until all other parts are soldered in place, and then do the soldering quickly with a SMALL IRON. Above all, never probe around sloppily on a live circuit or attempt to make circuit changes with the supply power applied.

This may seem like a bunch of don'ts, but if you have had any experience at all with the older logic families, you'll have to agree that CMOS has the least hassle associated with its use.

TABLE ONE
SOME CMOS MANUFACTURERS

HARRIS SEMICONDUCTOR
Box 883
Melbourne, Florida, 32901
305-727-5430

INSELEK
743 Alexander Road
Princeton, N.J., 08540
609-452-2222

MOTOROLA SEMICONDUCTOR PROD.
Technical Information Center
Box 20912
Phoenix, Arizona, 85036
602-244-6900

NATIONAL SEMICONDUCTOR
2900 Semiconductor Drive
Santa Clara, California, 95051
408-732-5000

RCA COSMOS DIVISION
Rte 202
Sommerville, N.J. 08876
201-526-3001

SOLID STATE SCIENTIFIC
Montgomeryville Ind. Park
Montgomeryville, Pa. 18936
215-855-8400

Some linear tricks

One nice experimental thing about CMOS is that you can convert an inverter into an amplifier simply by connecting a 10-megohm resistor from output to input. Any gate can also be converted by suitable termination of unused inputs. Fig. 6-a shows the basic amplifier which has a gain of 10 to 30 along with a high input impedance and a pretty wide output swing. This is handy for amplifying and limiting test signals and inputs, and anywhere else you might like to do something analog in a predominately digital system. One very handy application is the crystal oscillator in Fig. 6-b. It's one of the simplest logic oscillators you can build and one of the best performing owing to the high circuit impedances. A CMOS buffer stage should be added if you want to reach the outside world with this circuit.

There's also a bunch of unique analog switching you can do with circuits like the MC14016, and you can even use the MC14049 and MC14050 as hex, bilateral, symmetrical electrically variable resistors, provided you add a couple of resistors and work with low level signals. This is particularly useful for percussion keying in electronic music. At six notes per package, that's only two IC's per octave needed for a high performance true two quadrant multiplier.

TABLE II
A FEW TYPICAL CMOS DEVICES

CD4001	(MC14001)	Quad Nor Gate
CD4007	(MC14007)	Dual Uncommitted CMOS pair w/inverter
CD4011	(MC14011)	Quad Nand Gate
CD4013	(MC14013)	Dual D Flip Flop
CD4016	(MC14016)	Quad Analog/Digital bilateral switch
CD4023	(MC14023)	Seven stage binary divider
CD4026	(MC14026)	Decade Counter/7 segment decoder
CD4046	(MC14046)	Phase Lock Loop
CD4049	(MC14049)	Hex inverter buffer
CD4050	(MC14050)	Hex non-inverting buffer

Learning more about CMOS

We're not going to give you any circuits here, mostly because we are out of space. Maybe you can show us some. We can suggest three good ways to get more information and more experience with CMOS:

1. Get the data sheets and data books from the Manufacturers of table 1. Everyone listed offers some sort of book or data file on CMOS. One of the oldest and best is the RCA COSMOS Integrated Circuits Manual and normally costs $2.50. Everybody else on the list will be more than happy to send you something—provided you request it in a professional way. Absolutely type or phone your request; if possible use a business letterhead. Another route is to use the bingo cards from the dozens of electronic trade magazines—available at a library if you can't personally qualify.

2. Get some CMOS and hook it up. A good choice might be two each of the MC14001, MC14011, and MC14007, and one each of the MC14013, MC14016, and MC14046. Even at list prices, this assortment should be under $20, and much less as surplus. Prices are sure to drop. Be sure to watch the Market Center ads in **Radio-Electronics** for CMOS bargains.

3. Watch Radio-Electronics for applications ideas. Steve Leckerts CMOS clock in the April 73 issue was the first major CMOS advanced experimenter project. Many of the plug ins in the Digital Grinchwal series of test equipment (starting November 1972) used or will use CMOS. And, of course, if you come up with a good circuit on your own that other advanced experimenter's might be interested in, we'd probably like to publish it and pay you for it to boot.

Regardless of where you go for more information, now is the time to learn about CMOS, for no other logic now available has as attractive a combination of features, particularly suited for advanced experimental uses. R-E

TAPE PLAYER WON'T CHANGE TRACKS

This auto-tape player will not change tracks. Even with the panel pushbutton, nothing happens.—W.H., Dunlap, Iowa

Most of these use a solenoid to move the head for track-changing. Check the dc voltage across the solenoid terminals while holding the TRACK CHANGE button down. If you get voltage, disconnect the solenoid and check it for continuity. In several of these, you'll find a diode shunted across the coil, for transient suppression. If it shorts, the solenoid won't work.

IONIZERS AND AIR-CLEANERS

I want to build some things like ionizers, and electrostatic air cleaners. Where can I find parts for these?—A.B., Fremont, Ohio

The Triad Transformer Co., 305 N. Briant, Huntington, Ind. 46750, makes quite a few special high-voltage power supplies for such things. They'll be happy to send you a bulletin on them. As a matter of fact, the high-voltage power supply of a junked TV set could be used too.