HOW IT WORKS
IC MOS shift registers

Do you know what a MOS shift register is? Do you know how it works? Here are the answers plus how to interface them with other logic families and different applications

by DON LANCASTER

A SHIFT REGISTER IS A DIGITAL DATA STORage device. The data can be the letters to be displayed on a TV screen, numbers in a computer or calculator, intermediate values in a digital filter, or part of an elaborate code or sequence. Shift registers are made up of individual *stages*. Each stage can store one *bit* of information, called a binary 1 or a 0, and usually corresponding to a "yes" or "no" or else perhaps a "present" or "absent" command. Four bits together can represent a decimal number, while six bits together can handle one ASCII character, and so on. In a shift register, the contents can be moved or *shifted* so that the contained information is marched one and only one stage at a time through the device. The shifting process is called *clocking* and one or more clocks are involved in completing the shifting operation.

Figure 1 shows how we might make a shift register out of either a JK or type-D flip-flop. While TTL (Transistor-Transistor logic) devices are shown, we could use any logic family we like. Input data corresponding to a "1" or "0" is presented to the first stage. When the system is clocked, the first bit of data is *entered* and then stored in the *first* stage. On the second clocking, the contents of the first stage get passed on to the second, and the first stage then accepts a new bit of information from the input. The next clocking passes the output of stage 2 on to stage 3, and the output of stage 1 on to stage 2. Finally, stage 1 accepts a new bit of input information.

One more clocking *fills* the register in Fig. 1 as it is only four bits long, and all four stages now have information in them. If we do no more clocking, the register will *keep* the information we sent it. Four more clocking pulses and we can march the data out and use it somewhere else.

So what good is a shift register? We can use it to *store* information. It is a digital *memory*. We can use it to *delay* information. We can use it to *format* information, either in a *buffer* mode where the enter and readout clock rates may be different, or in a *variable-access* mode where we can enter and leave individual stages with data. With certain types of shift registers, we can convert *serial* data to *parallel* form or parallel data (all at once) to serial (one at a time in sequence) form. We can also build counters and sequencers with shift registers. Two popular types are called the *walking ring* computer and the *pseudo random sequence* generator.

Organization

The *organization* of a shift register is decided by how many stages it has and how you can get at the individual stages.

A serial-in-serial-out register gives you the input only to the first stage and the final output of the last stage. it is sometimes called a *serial* register or a SISO (Serial-in-Serial-Out) register. There is no intermediate access.

A SIPO register gives you the outputs of all stages including the last one. The eight-bit 74164 is a typical TTL example. A parallel-in-serial-out or PISO register lets you simultaneously load all the stages but then marches the contents out as a serial-bit string. The TTL 74165 is an eight-bit example of this type.

The most versatile type of shift register would be a PIPO (Parallel-In-Parallel-out) version. Here, you could load data either serially one bit at a time or "broadside" parallel. You could also get all the data out either in broadside parallel all-at-once form, or one bit at a time in serial form. The 74195 is a four-bit TTL package that does this.

You might think that since you could use the PIPO register for everything else anyway that it would be the only way to go. The problem is that you can easily put 2048 shift register stages on a single small chip of silicon. For a 2048-bit PIPO register, you'd need a minimum of 4099 leads for inputs, outputs, clocks, and power supplies. This is a most unwieldy package to say the least, even if we don't worry about the extra circuitry needed for each parallel input. Now the same register can be done SISO in as little as 5 leads.

So, for *short* shift register applications, we have a choice of the four formats. For *long* shift register uses, the only economical way to go is the SISO route. We'll consider everything longer than 24 bits a *long* shift register here. This is often a changeover point. 24 bits or less and you usually use the more flexible and faster TTL registers, often at four or eight stages per package. Above 25 bits, you go to the long serial MOS registers and pick up as many as 2048 bits of storage in a single package.

The majority of registers shift only towards the output and are called *shift right* registers. A very few can also shift back towards the input and are called *bidirectional* or *shift-right-shift-left* devices. These are expensive and not normally available in long lengths. One trick you can do with a *recirculating* register (more on this in a bit) is clock it rapidly ahead one stage less than its length, making it *appear* to back up one, rather than go forward *all but one* of its stages.

Two more things may enter into our register organization. We may have more than one shift register in a single package. One, two, and six registers per package are common. Usually, they have common clocking, but not always. For instance, the Signetics 2518 is a hex 32-bit shift register; the 2519 is a hex 40-bit version. Both have common clocking and a common enter/recirculate control.

You often use several shift registers in parallel. For instance, you might use four shift registers to individually handle each bit of a four-bit BCD or binary-coded-decimal digit. Thus each clocking of the register array gets you a whole new decimal number, rather than only ¼ of it The four bits is sometimes called a *word* and sometimes a *byte*. Likewise, an alphanumeric character can be represented by a six bit ASCII character code. Here, we use six registers at once to give us one whole new character on each clocking. Of course, we have to make sure all the registers get clocked exactly alike, for if they didn't, all the data bits would be hopelessly scrambled. This is usually very easy to prevent.

A final feature of a shift register's organization is its *recirculatibility*. Sometimes we might like to look at the contents of a shift register a bit at a time, and then *return* the information back into the same relative slots in the shift register for later use. This is called *recirculation*. Some sort of switching or selection must be provided if you are sometimes going to *enter* new data as opposed to recirculating old data. Some of the long MOS shift registers have an *internal* recirculate logic and are normally used if you need recirculation. We'll see in a minute that recirculation is essential for the *dynamic* registers if you are going to keep the data more than a fraction of a second. Figure 2 shows the logic needed to add an external recirculate to a shift register.

Long MOS shift registers

There's an incredible variety of long shift registers available using several different MOS (Metal-Oxide-Semiconductor)

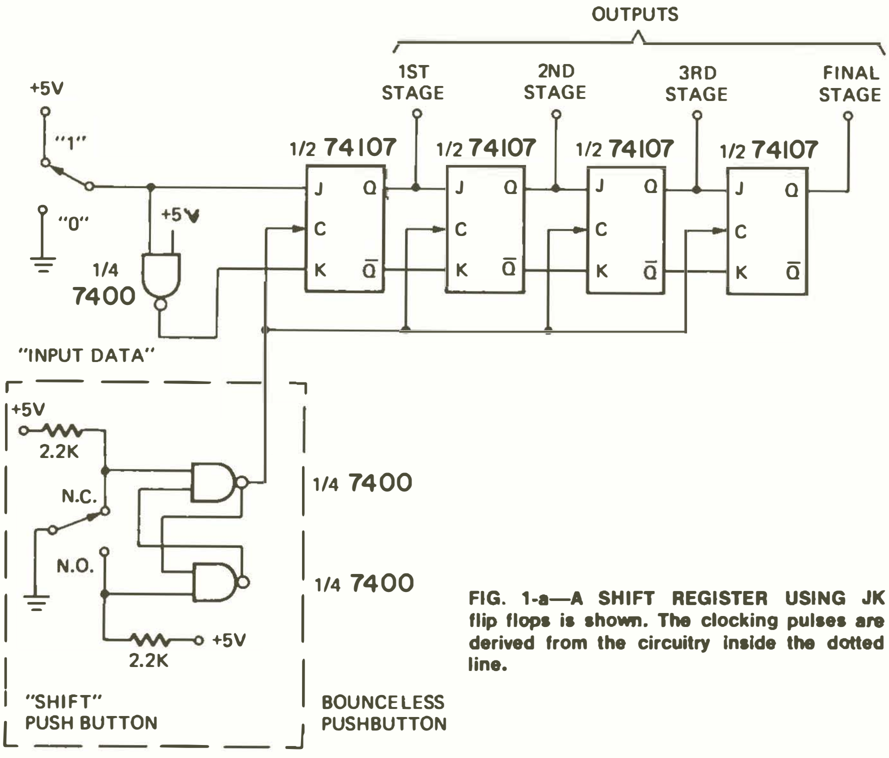


FIG. 1-a—A SHIFT REGISTER USING JK flip flops is shown. The clocking pulses are derived from the circuitry inside the dotted line.

(*a*) USING "JK" FLIP FLOPS

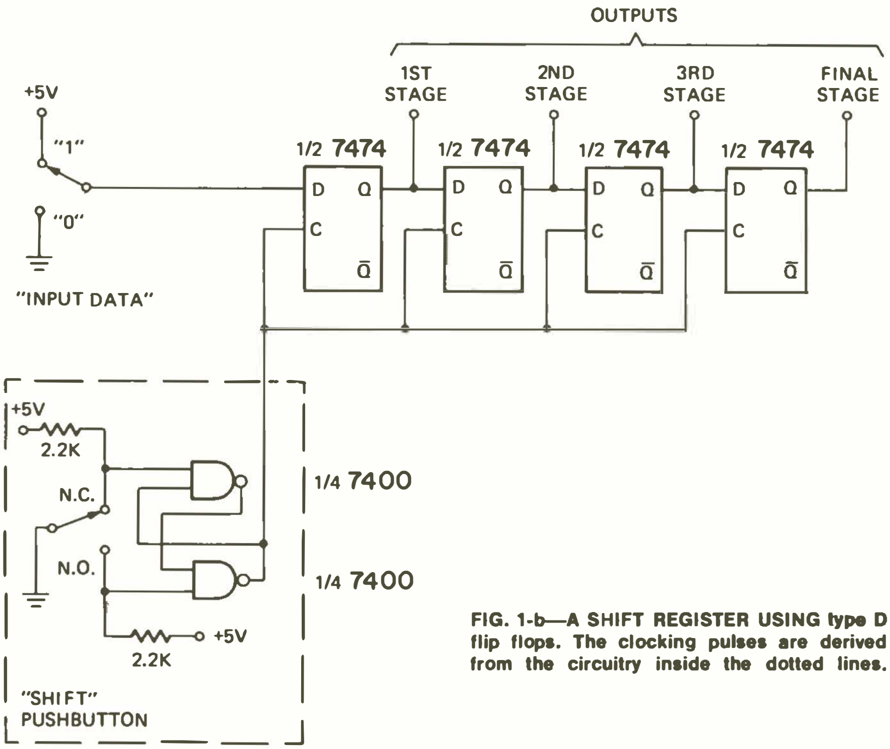


FIG. 1-b—A SHIFT REGISTER USING type D flip flops. The clocking pulses are derived from the circuitry inside the dotted lines.

(*b*) USING "D" FLIP FLOPS

technologies. These range from small 16- and 21-bit versions up to 2048-bit ones in a single package. A brief and more or less random listing is given in Table I, while some of the more prominent manufacturers are listed in Table II. The typical single-unit price varies from around $3 to around $15 per unit and typically runs well under a penny per bit for the longer versions. Some of these have shown up surplus (see back ads of **Radio-Electronics**) for as little as a quarter each for manufacturers seconds. Some of the seconds we tested from the back ads run around a 45% "completely useful" yield. All of these devices are serial-in-serial-out. Typical maximum frequency of operation is 2 or 3 megahertz, although you get much better behavior at a 500 kHz or so rate.

Before you can use any long MOS shift register, you have to ask the following questions:

1. Is the register *static* or *dynamic?*
2. How do you *interface* it with TTL or other logic?
3. What kind of *clock* signals are needed and how many of them?
4. Can it recirculate by itself?
5. Does it have write or read *enables* that lets you combine it with more registers?

Let's take a look at these important concepts in a bit more detail.

Static versus dynamic

Figure 3 shows three different types of shift registers. Our registers of Figs. 1 and 3-a used two flip-flops for storage. They will keep data so long as we apply supply power and are called *static* registers, or sometimes *fully static* registers.

Transformation of information in *any* shift register *has* to be a *two-stage* process or a *two-phase* process. On the beginning of a shift, information is transferred into some form of temporary storage. At the completion of a shift, the information is then sent to a *final* storage. In the case of Fig. 1-a, we have a master (temporary) and a slave (final) storage *within* each JK flip-flop's logic block. The reason for the *necessity* of *two* storage phases per shift is simple—try it with only one, and you get a wild, unchecked race through several stages instead of an orderly progression of one and only one complete stage per clocking.

We don't need a full flip-flop for some applications. Instead, we can use the temporary storage of a capacitor. So, Fig. 3-b shows us a *dynamic* shift register. The capacitor will hold information for us for a reasonably short time, but eventually the leakage will get to us and destroy the information in the cell. Capacitor storage is much simpler and more economical than flip-flops as it usually uses the "free" capacitance found in normay strays. Most dynamic MOS shift registers will hold their information for UP TO one-tenth of a second. Should you fail to clock them in that time, the information is lost.

So, if you are only going to keep your information in your shift register for under a fraction of a second before finally using it, it doesn't matter whether you use a static or a dynamic register. The trouble is that most applications call for data to be reused or held longer than a fraction of a second. So, if you are to use the cheaper, denser dynamic shift registers, you have to move or *refresh* the data a minimum of several dozen times a second. One way to handle the moving of data is to march the information completely once around at least several dozen times per second. In a computer terminal or TV Typewriter, recirculation at the 60 hertz vertical rate is one good approach.

Figure 3-c shows an interesting compromise between static and dynamic registers. Here, we use a capacitor for the temporary storage and a flip-flop for the final storage. This is a compromise that gives us static performance at slightly over half the normal cost. Strictly speaking, this is called a *quasi-static* operation, but practically all the "static" MOS reg-

isters use this technique. There is only one restriction, the clock line must remain in a specified level during the static part of the operation, and there is a *maximum* allowable clock pulse width during the dynamic transfer process.

Interface

Most of the long MOS registers will interface with TTL, DTL, and RTL, but most often a few resistors are needed. You have to read the data sheets very carefully. Unless the data sheet specifically states otherwise, the clock lines are NOT compatible with TTL and take special drive circuitry. More on this in just a bit. Remember that the inputs, enables, recirculates, and output pins can be made TTL compatible, but the clock almost always takes special circuitry.

There are lots of different MOS technologies, and each takes one of the interface circuits shown in Fig. 4. You can usually tell the technology by the supply voltage used or recommended.

If the supplies are ±15 volts, chances are it is a *metal gate* or *high threshold P channel device*. These are the oldest MOS integrated circuits and the hardest to interface. To drive them, you need an *open circuit* TTL logic block that can withstand 15 volts. Suitable devices are the 7406 and 7416. A pull-up resistor is provided to produce the ground and ±15-volt logic inputs. Two resistors are normally used in going from the MOS to TTL, one down to −15 to provide the −1.6 mA needed for a TTL "0", and one series resistor to limit the positive swing to 5 volts or less.

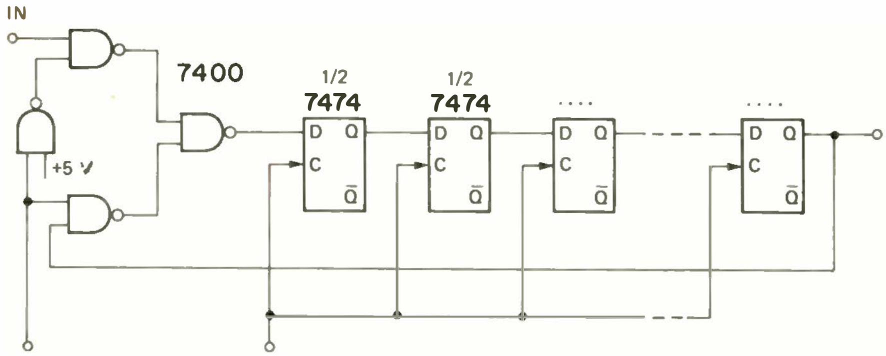
Silicon gate circuits are presently the most common. They have a +5 and −12-volt supply. Usually a 2.2K pull-up resistor is recommended when they are driven by TTL, and their output drive capability depends on the particular output structure used. Often a single 6.8K resistor to −12 volts does the trick.

N-channel circuits often work with a single +5-volt supply and are directly TTL compatible without resistors on output and input. *CMOS* integrated circuits also work off a single +5- to +15-volt supply. At + 5 volts, they are directly TTL compatible on an input, but may not have enough output drive current for regular TTL, so low-power TTL is often used as an output sense amplifier.

Its usually tricky to simultaneously drive another MOS stage along with TTL as the voltage and current swings don't usually work out too well. To get around this, you usually run through a single TTL inverter and use its output to drive the MOS following.

Clocks

More problems happen with long shift registers over clocks and clocking than over any other single difficulty. First and foremost, consult the individual data sheets for the device you are going to use. Unless it specifically says so otherwise (boldly and in large print!), the clock lines are not compatible with TTL. Usually the clock lines need almost the entire supply swing, such as a 16- or 17-volt swing for a silicon gate circuit on +5-, −12-volt power supplies. Further, what-



⏚ = ENTER
+5V = RECIRCULATE

FIG. 2—RECIRCULATING SHIFT REGISTER. Data can be fed from the output to the input.

TABLE I

A FEW OF THE MORE POPULAR LONG MOS SHIFT REGISTERS

ELECTRONIC ARRAYS:

EA1003 Dual 32, static, rec.
EA1004 Dual 100, static
EA1007 Dual 32, static
EA1200 Quad 32, dynamic
EA1203 Variable 1-64 dynamic
EA1210 Dual 526 dynamic
EA1212 Single 512 Dynamic

FAIRCHILD:

3325 Quad 64, Dynamic
3330 480 Bit, Dynamic
3342 Quad 64, Static
3343 Dual 128, Static
3346 Dual 144, Static
3383 Single 256, Dynamic

INTEL:

1402 Quad 256, Dyn, Mpx.
1403 Dual 512, Dyn, Mpx.
1404 Single 1024, Dyn, Mpx.
1405 Single 512, Dyn, Recirc.
1506 Dual 100 dynamic
2401 2048 dynamic, recirc.
2405 1024 dynamic, recirc.

MOSTEK:

MK1002 Dual 128, Static
MK1007 4 x 80, dynamic

MOTOROLA

MC1141G Triple 66 dynamic
MC1142G Single 200 dynamic
MC1160G dual 100 dynamic
MC1161G Dual 50 bit static
MC2360G Dual 100 Static
MC2361G Dual 128 Static
MC2362G Dual 250 Static
MC2363G Dual 256 Static
MC2380G Dual 100 dynamic

NATIONAL:

MM400 Dual 25 Dynamic
MM402 Dual 50 Dynamic
MM406 Dual 100 Dynamic
MM4001 Dual 64 Dynamic
MM4006 Dual 100 Dynamic
MM4012 Dual 256 Dynamic
MM4013 Single 512, dyn, rec.
MM4105 Quad 64, static
MM5054 Dual 64/72/80 static

SIGNETICS:

2505 Single 512 dyn, rec.
2506 Dual 100, dynamic
2509 Dual 50 Static
2510 Dual 100 Static
2511 Dual 200 Static
2512 Single 1024, dyn, rec.
2518 Hex 32, static, rec.
2519 Hex 40, static, rec.
2521 Dual 128, static
2522 Dual 128, static
2524 Single 512, dyn, rec.
2525 Single 1024, dyn, rec.
2527 Dual 256 static
2528 Dual 250 Static
2529 Dual 240 Static
2532 Quad 80 static
2533 1024 static, rec.

TEXAS INSTRUMENTS:

TMS3000 Dual 25 static
TMS3001 Dual 32 static
TMS3002 Dual 50 static
TMS3012 Dual 128, stat, rec.
TMS3102 Dual 80, static
TMS3112 Hex 32, static, rec.
TMS3113 Dual 133 static, rec.
TMS3304 Triple 66, dynamic
TMS3309 Dual 512, dynamic
TMS3314 Triple 60+4 dynamic
TMS3412 Single 1024 Dynamic

TABLE II

SOME LONG MOS SHIFT REGISTER SOURCES

ELECTRONIC ARRAYS INC.
501 Ellis Street
Mountain View, California 94040

FAIRCHILD SEMICONDUCTOR
464 Ellis Street
Mountain View, California 94040

INTEL CORPORATION
3065 Bowers Avenue
Santa Clara, California 95051

MOSTEK
1215 West Crosby Road
Carrolton, Texas 75006

MOTOROLA SEMICONDUCTOR
Box 20912
Phoenix, Arizona 85036

NATIONAL SEMICONDUCTOR
2900 Semiconductor Drive
Santa Clara, California 95051

SIGNETICS
811 East Arques Avenue
Sunnyvale, California 94086

TEXAS INSTRUMENTS
Box 5012
Dallas, Texas 75222

ever is driving the clock has to drive a bunch of internal switches in a long register, so the clock line capacitance may be several hundred picofarads. Since you need sharp rise and fall times on the clock, it usually takes a special circuit called a *clock driver* to get the job done, as the peak currents involved in charging and discharging the clock line capacitances may be several hundred milliamperes or more. Except for the simplest circuits, a push-pull "totem pole" drive circuit is needed, and a small current limiting resistor (usually 10 ohms) must be provided between the registers and clock lines to prevent short circuit damages and risetimes that raise havoc with the supply lines and decoupling. The clocks must NEVER be allowed to "overshoot" and exceed the positive supply voltage, even briefly for this will destroy or selectively change the information in the register. Clocks must be the proper widths and must not overlap. Where two clocks are used, the "daylight" or space between them is just as important as their widths.

As a general rule, always use clock widths near the *minimum* called for on the data sheets. With most registers, the wider the clock pulses, the more the supply current, and the hotter the IC runs, leading to potential temperature and bit pattern sensitivity problems. Clock widths should be precisely derived from system timing instead of randomly adjusted through monostables or half-monostable pulse shapers, since the position and widths can be quite critical.

On your first design with a new long MOS register, you also have to watch for the number of clocks needed per cycle. Generally static registers need a single clock and each clock pulse advances the information one stage. Static registers are also usually much easier to drive on their clock lines.

Most dynamic registers have two clock lines and need two clock drivers. One clock is the *input* clock; one is the *output* clock. A *pair* of clock pulses is needed to advance the information one stage.

Finally, there are a few dynamic *multiplexed* registers such as the *Intel* 1402, 1403, and 1404. These are tricky and hard to use. They contain *two* internal shift registers with a *common* input and output. What is an input clock for one side is the output clock for the other half and vice versa. The data *externally* appears to travel one stage *per* clock pulse, although a *pair* of clock pulses is needed to *complete* each transfer operation. If you are not very careful, you can end up one clock pulse short or long of what you really need, and change the effective register length.

Note that any of these devices can have the clocks spaced out in time. They need not be continuous. They can be in bursts or random, so long as you don't exceed the minimum clock width and "daylight" spacing, and so long as you don't wait

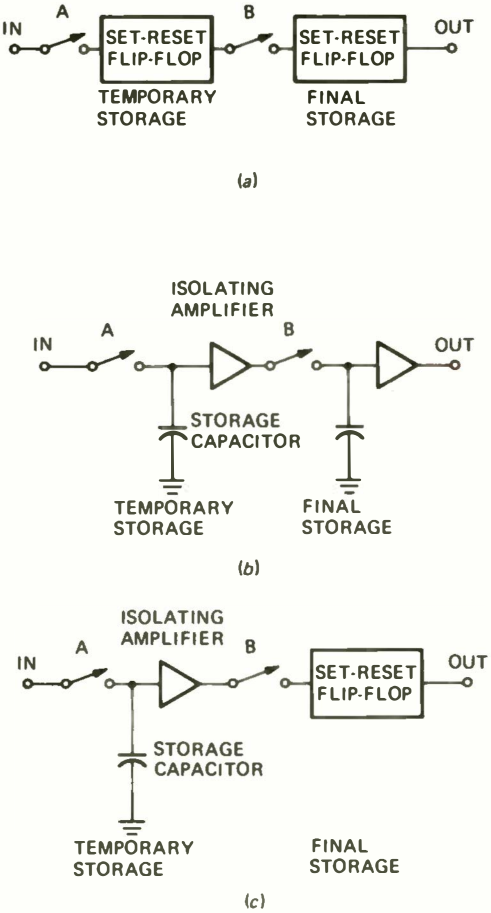


FIG. 3 (top of page)—STATIC shift register. b) DYNAMIC shift register. c) QUASI-STATIC shift register.

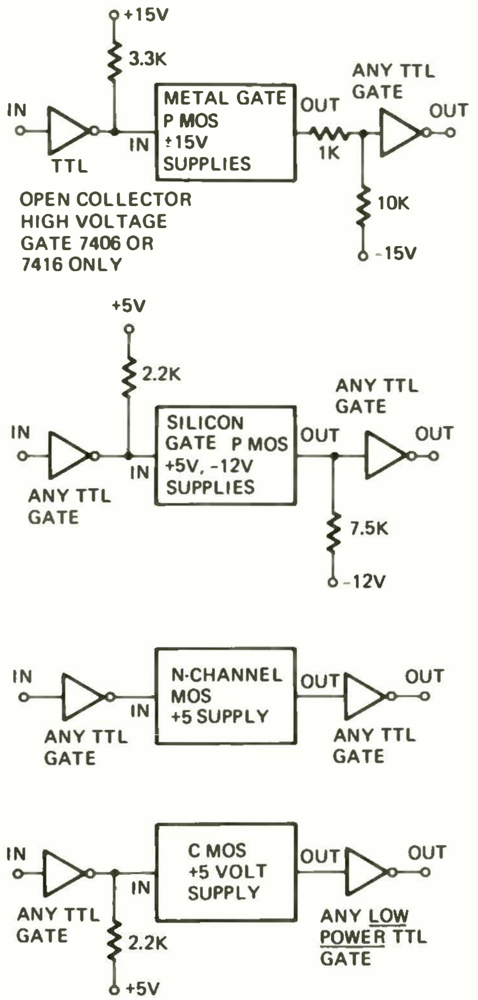


FIG. 4 (bottom of page)—INTERFACING DIFFERENT MOS logic with TTL gates. The type of MOS logic can be identified by the supply requirements.

longer than the dropout time on a dynamic register. Outside of the capacitance you may have to charge and discharge rapidly, *all* of the inputs on any MOS integrated circuit are essentially open circuits and neither source nor sink current.

Enables

An *enable* pin lets you combine either the outputs or inputs of a shift register group without using any fancy selector switches or external logic. Output enables are sometimes called *read enables.* You can combine memories simply by shorting all the outputs together provided you enable only one circuit at a time. Two common types of enables are the *open collector* and the *tri-state*. The latter provides a "1", a "0", or a high-impedance open circuit on command. Write enables also exist, but only on a few of the long registers.

Applications

We only have enough room to quickly run down some obvious applications of long shift registers. Two important ones were shown in the TV Typewriter story (**Radio-Electronics,** September 1973). Six recirculating 512-bit registers were used as a main memory character store and a final hex 32-bit shift register was used as a line register needed for formatting the dot matrix characters.

Pocket calculators and computers use long shift registers for number and program storage. Often, they are combined with internal multiplexing, calculation, and control circuitry into a single package.

Some music synthesizers use long shift registers as tune computers or composer storage. Several far out tricks that can be done with them is the separation of pitch and tempo, and the ability to play an upside down scale, or a reversed or backwards score. To reverse a shift register, you simply run it ahead N-1 clock pulses as fast as you can go. For instance, a 512-bit shift register can be clocked ahead 511 bits in well under a millisecond, and it appears to have backed up one slot at the end of the burst.

Long shift registers are ideal for sequence generation of noise that repeats for cryptography, computer security, music, and audio testing applications.

Long shift registers make good *buffers* or *data concentrators.* Input information can be loaded into a shift register at a random, slow, or asynchronous outside-world rate and then transferred to the rest of your circuit later on synchronously at high speed.

You can build an electrically variable delay line out of long shift registers. The clocking controls the delay time independently of the input data frequencies. You can get a delay to risetime ratio of 500:1 out of a 1024-bit register, something that's hard to do with analog delay lines. Speech compression (for talking book tapes and records), vibrato (for music synthesizers), and spectrum translation are three typical use examples.

In fancier circuits, shift registers are used as the key element in digital filters,

(continued on page 97)

AUDIO FEEDBACK CIRCUITS

(continued from page 87)

Both the boost and cut circuits are in the operational amplifier circuit and Equation 10 does apply. Converting R1, R3 and R1 mathematically from a "tee" to a "delta" configuration to facilitate analysis, will yield a corner boost frequency at f_{obb} and a corner cut frequency at f_{obc}. They are both equal to 1/6.28C5 (R1 + 2R4).

The intermediate settings of the control will yield intermediate amounts of treble boost and cut. As was the case with the bass control, the corner frequency is shifted away from the center frequency when less boost or cut is required at the upper ends of the band. The setting of the control will not affect the center or low frequency regions of the band.

The value of C5 was set at about 100-pF, so it would not load the input circuit excessively and yet be large enough not to be affected by stray capacitances in the circuit.

f_{obb} was chosen for about 16 dB of boost at 10,000 Hz. An approximate curve used to determine the corner frequency is shown in Fig. 9. At the maximum setting of the control, $f_{obc} = f_{obb} = 1.5$ kHz. Since R1 and C5 are already known, R_4 is calculated to be about 500,000 ohms.

R5 must be made as small as practical when compared to the reactance of C5 at the highest audio frequency that must be boosted. A 500,000-ohm linear center-tapped potentiometer was found to be satisfactory.

A low-gain amplifier or lower impedance bipolar transistor are frequently used in the feedback tone control circuit in place of the JFET. As these components cause the operational amplifier to differ radically from the ideal, the components must change from the calculated values to produce results similar to those outlined above. The circuit should be designed in the laboratory in this case. Since the function of each component has been detailed, the effects of changing a component is known and the design procedure does not have to be haphazard.

A complete tone control circuit has been drawn in Fig. 10 showing the bass and treble controls. The following factors affecting the various functions of the control should be noted.

The amount of boost and cut produced by the treble control is affected by R4 and C5. Make either component larger if more treble action is required. To a lesser degree, increasing R1 increase the amount of treble boost, while increasing R3 affects the size of the treble cut.

As for the bass circuit, C3 and R3 must be increased to further emphasize the boost while C2 and R1 must be increased to accentuate the cut. R-E

MOS SHIFT REGISTERS

(continued from page 62)

correlators, and Fourier series calculators. And, as a final and obvious application, shift registers are being used to replace magnetic discs as medium-speed, high-density storage systems for computers. These are often called *silicon disc* files.

Getting started

If you are new to shift registers, pick up a few of the bargain surplus units and try experimenting with them. You'll get best results if you stick with the static units at first and avoid the older metal gate ±15 volt circuits as they are hard to interface. Remember to pick up several units at once if you are buying seconds. Above all, have the exact data sheet on hand, and if possible, some application notes as well. Be sure to have your power supplies well decoupled and regulated and make sure your clock lines and drivers *exactly* meet the specified requirements. Keep your clock pulse widths down around the minimum recommended values to minimize internal heating and try to derive the clock widths and spacing from digital logic and timing rather than using adjustable monostable delays. R-E

PHOTO ETCH™

PRINTED CIRCUIT KIT

Makes circuits THREE WAYS

1 FULL SCALE ARTWORK MASTER

2 MAGAZINE ART ILLUSTRATION → FILM POSITIVE → NEGATIVE → DEVELOPED PHOTO RESIST IMAGE ON COPPER CLAD CIRCUIT BOARD → ETCHED CIRCUIT BOARD

3 DIRECT ETCH DRY TRANSFERS APPLIED TO COPPER CLAD BOARD

NO CAMERA DARKROOM FILM CUTTING TRACING

USES DATAK'S POS-NEG™ PROCESS The revolutionary photographic way that makes PERFECT printed circuits from original art or a printed page.

KIT CONTAINS: 5" x 6" steel printing frame, 4 sheets 5" x 6" photocopy film, yellow filter, chemicals for 1 pint film developer and 1 pint film fixer, 5" x 6" copper clad board, 3" x 4½" copper clad board, spray can of photo etch resist, 1 pint resist developer, 2 sheets 8½" x 11" layout film, 1 roll 1/16" printed circuit tape, 1 roll 1/32" printed circuit tape, 8 sheets dry transfer direct etch PC patterns including pads, transistors, round can and flat pack ICs, DIP ICs, edge card connectors, lines, circles, jogs etc., ½ lb anhydrous ferric chloride to make 1 pint etchant, instructions.

ER-4 COMPLETE PHOTO ETCH SET	$24.95
ER-2 PC patterns and tapes—refill	3.39
ER-3 ¼ pound dry etchant—refill	1.25
ER-5 6 sheets photocopy film—refill	3.39
ER-6 Film process chemicals—refill	1.79
ER-7 Photo resist spray, 2.5 oz.—refill	2.95
ER-8 Resist developer, 16 oz. can—refill	2.95

AT YOUR DISTRIBUTOR OR DIRECT

the DATAK corp.
65 71st St. • Guttenberg, N. J. 07093

Circle 79 on reader service card

Accuracy like a VTVM... Convenience like a VOM...

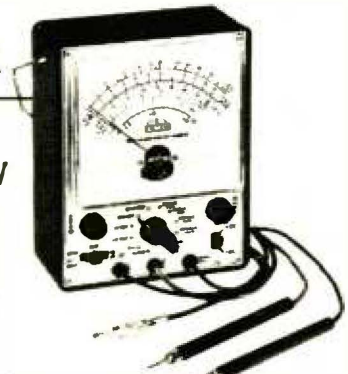
NEW BATTERY-OPERATED FET SOLID-STATE VOLT-OHMMETER #116

Easy-to-build KIT

$36.90 #116K

Factory-Wired & Tested

$49.49 #116W



Now you can get all the benefits of a VTVM (laboratory accuracy, stability and wide range) but with its drawbacks gone: no plugging into an AC outlet, no waiting for warm-up, no bulkiness. New Field Effect Transistor (FET) design makes possible low loading, instant-on battery-operation and small size. Excellent for both bench and field work.

Compare these valuable features:
• High impedance low loading: 11 megohms input on DC, 1 megohm on AC • 500-times more sensitive than a standard 20,000 ohms-per-volt VOM • Wide-range versatility: 4 P-P AC voltage ranges: 0-3.3, 33, 330, 1200V; 4 RMS AC voltage ranges: 0-1.2, 12, 120, 1200V; 4 DC voltage ranges: 0-1.2, 12, 120, 1200V; 4 Resistance ranges: 0-1K, 0-100K, 0-10 meg., 0-1000 meg.; 4DB ranges: −24 to +56DB.

Sensitive easy-to-read 4½" 200 microamp meter. Zero center position available. Comprises FET transistor, 4 silicon transistors, 2 diodes. Meter and transistors protected against burnout. Etched panel for durability. High-impact bakelite case with handle useable as instrument stand. Kit has simplified step-by-step assembly instructions. Both kit and factory-wired versions shipped complete with batteries and test leads. 5¼"H x 6¾"W x 2⅞"D. 3 lbs.

Send FREE catalog of complete EMC line and name of nearest distributor. RE-12

Name______

Addr______

City______

State______ Zip______

EMC

ELECTRONIC MEASUREMENTS CORP.
625 Broadway, New York, N.Y. 10012