A Recipe for Homebrew ECL*

Chuck Hastings

Emitter-coupled logic (ECL) is understood by most computer designers to be the fastest stuff available—which it is—and as too difficult for anyone but the largest companies to design with—which it isn't. This paper will present a brief introduction to ECL and a practical recipe, used once successfully, for designing, building, and troubleshooting a small ECL system with approximately the same level of resources likely to be available in a well-equipped homebrew lab.

*This paper first appeared in the *Conference Proceedings of the Second West Coast Computer Faire,* published by the Computer Faire, Suite 110, 611 Veterans Boulevard, Redwood City, CA 94063. A revised version appeared in the *Computer Music Journal,* Volume 2, issue 1, pages 48-59, 1978; and again with additional revisions in *Foundations of Computer Music,* edited by Curtis Roads and John Strawn, 1985; both the journal and the book are available from the MIT Press, Cambridge, MA.

A Recipe for Homebrew ECL

Chuck Hastings

Why Read this Recipe?

Emitter-coupled logic (ECL) is understood by most computer designers to be the fastest stuff available—which it *is*—and as too difficult for anyone but the largest companies to design with—which it *isn't*. If an appropriate recipe is followed, ECL systems can be developed with very limited resources with as good, or better, chances of technical success as with equivalent transistor-transistor logic (TTL) systems. Thus, homebrew ECL is a viable alternative for applications that require very high-speed processing. Such applications may occur in some technical approaches to music synthesis, speech analysis[1] or simply fireside number crunching involving matrices, partial differential equations, or Fast Fourier Transforms.

Such a recipe isn't written down anywhere—existing ECL tutorials make ECL design sound formidable. However, a careful amateur can achieve a reliable 100-MHz small system today. This paper will present a practical recipe, used once successfully, for designing, building, and troubleshooting a small ECL system with the level of resources available in a well-equipped homebrew lab.

This recipe was developed during the course of one task in a mid-1970's project at Racal-Milgo, then a medium-sized Florida company with no previous ECL systems experience. The circumstances were in many ways quite similar to those of a homebrew project. The outcome of the task was a 24-bit general-purpose stored-microprogram computer, capable of 6 million three-address fixed-point add/subtract/Boolean instructions or 900,000 fixed-point multiply instructions per second, which was completed and subsequently was operated 10 hours a day for several months in a signal-processing system.

Why ECL?

Why ECL? For openers, the established industry-standard 10,000 series ECL (hereafter referred to as "10K") offers at least twice the net speed of Schottky TTL when actually designed into typical systems. Its successor line, 10H000-series ECL (hereafter, in like manner, "10KH") is *again* twice as fast as 10K, and pin compatible with 10K on a part-by-part basis. 10K/10KH provide a more natural and less brute-force approach to high-speed signal transmission than Schottky, and is in a number of respects actually easier to use.

ECL has probably not been considered for many applications where it would have been appropriate, both in industry and more recently in hobby work, because people tend to be scared to death of it. Frankly, ECL has an image problem—see Figure 1. Like many image problems, this one has some basis in truth; but there has been a considerable overlay of exaggeration, distortion, and mythology, which I will do my best to dispel based on the results obtained in one medium-sized computer hardware-development project.

Much of what I have to say concerns a subject euphemistically called "interconnection practice," which means all the things you have to do to keep your logic from being thoroughly confused by its own noise after you turn it on. Except as occasionally noted, all of my remarks concern 10K in a wire-wrap environment. Later on, I'll have a little to say about other ECL families, such as MECL III, PECL III, and Fairchild 100K, and the new 10KH.

A good wire-wrap board, believe it or not, is an excellent signal environment for high-speed logic. I have met people who solemnly claimed that one can't wire-wrap ECL, but it just ain't so. Communications Satellite Corporation, and the Mayo Clinic[8] have both done it for years. I have also

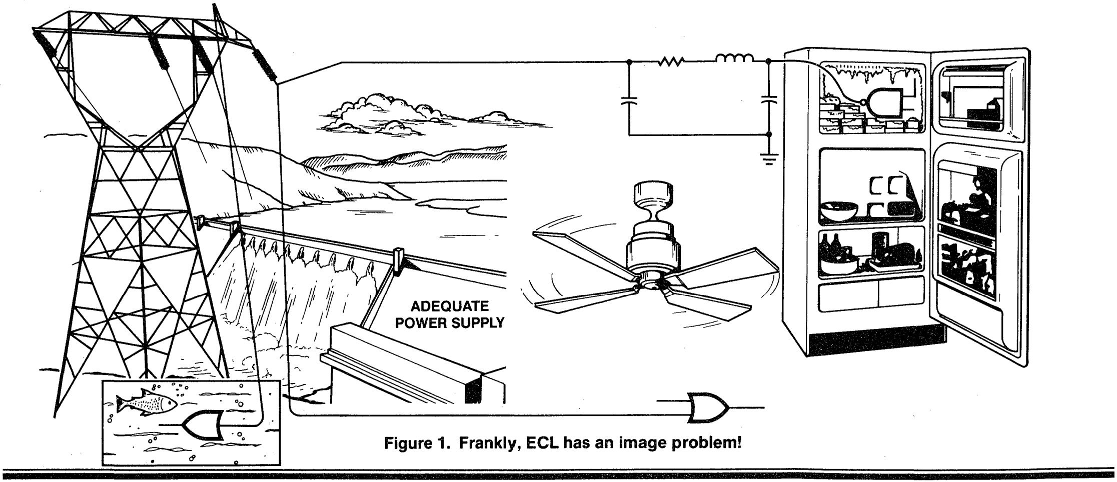

Figure 1. Frankly, ECL has an image problem!

met people who claimed that wire-wrap fabrication was something one does only for prototypes and that it is too expensive to be a manufacturing technique; but Modular Computer Systems in Florida has been cranking out wire-wrapped minicomputers since about the beginning of the 1970s. Much of the wire-wrap equipment used in industry is made by Gardner-Denver, and many hobbyists use Gardner-Denver rechargeable battery-operated wire-wrap guns. There is also a company called OK Machine and Tool Corporation which makes a line of low-cost wire-wrap equipment specifically marketed for use by hobbyists. (See the appendix for a list of addresses.)

Since most hobbyists probably prefer to have their systems work without a major initial checkout hassle, my interconnection-practice recipe probably errs on the side of overkill. If for some underground entrepreneurial reason you are intensely concerned with the cost of replicating a homebrew ECL system once it is working, you can do a cost-reduction job by deleting some of the practices I am advocating one by one until the system goes bananas. But don't start out doing an el cheapo job—if the system doesn't work at all, you may not have the equipment, resources, or patience to find out why. Big companies do have the luxury of trading off more product-development engineering hours against lower manufacturing costs, but you probably don't. The first time you do it, do it right.

An Astounding Claim

The fear of ECL in the industry is so great that it requires some chutzpah on my part to state straight out that you too can successfully build, debug, and operate ECL systems in your spare bedroom, garage, or rumpus room—just like TTL and metal oxide semiconductor (MOS). You don't have to have the vast resources of a company like Control Data, Univac, IBM, Gould or Burroughs behind you to succeed—or even those of a rather unusual small company such as Cray Research or Denelcor to name two with some obvious ECL expertise.

I make this statement on the basis of successfully developing a medium-sized, high-performance ECL midicomputer under what might be called primitive industrial conditions at a company (Racal-Milgo) having no prior experience building either ECL systems or digital computers. Up until that time, the company management had not particularly understood digital computers, although they did have some expertise in analog computers. The backup resources which one expects to find in place in even a small computer mainframe house simply weren't there.

To top it all off, I myself am a computer systems type—ones and zeroes, architecture, logic design, machine-level software, microprogramming—with very little expertise in, say, linear circuit design or electromagnetic field theory. All the same, with one sharp technician working with me full-time plus part-time help from a few other people, I was able to get a high-performance digital system of about 900 ECL 10K chips developed and operating in about 15 months. (Of course, if I were to design the system today, I would use 10KH chips, which were not available then, wherever possible.) Thereafter, for several months, it was operated many hours a day, five or six days a week, as part of a larger signal-processing system, with very few maintenance problems. If I can do something like that, probably you can too.

The Miami Number Cruncher

The architecture of this midicomputer is not the main point of my presentation, so I'll say just enough about it to put it in perspective. It had a three-address format, with a 48-bit instruction word and a 12-bit data word. Instructions and data came from separate memories with *separate* addressing spaces ("Harvard architecture"). Arithmetic was generally 24-bit twos-complement, with some 12-bit operations also available. The *minor* cycle *(clock interval)* was about 10.17 nanoseconds, which is the reciprocal of the 98.304-MHz basic frequency. One microprogram step required a *major* cycle, consisting of 5 to 12 minor cycles according to a 3-bit microprogrammed field.

Normal execution time for a 24-bit add or subtract instruction was 163 nanoseconds, and a Boolean instruction required one minor cycle less; instructions of both these types required two major cycles. The time of 163 nanoseconds was for a *memory-to-memory* operation, not merely register-to-register, since the main data memory (4K 12-bit words) was comprised of 20-nanosecond-access 1K-by-1-bit ECL memory chips (type 10415A/10146). *Two* copies of *all* main memory words were implemented, in order to avoid the penalty of an extra major cycle on each execution of one of these instructions.

The approximate times for some other 24-bit three-address operations were: 1.1 microseconds for multiplication, 3.5 microseconds for division, and 13 microseconds for the square root of a sum. There were both single-word and block-oriented input and output instructions, and an external command instruction, with a fully asynchronous handshake control philosophy. All instruction sequences were controlled entirely by stored-microprogram techniques.

The computer itself, including both data and instruction memories, occupied three large [418 dual in-line package (DIP) locations] wire-wrap boards mounted in aluminum frames, and drew a little more than 300 watts. It was part of a larger experimental signal-processing system for a proprietary real-time application, and was never intended to be a product in its own right.

Test Equipment

Probably the scale of this machine is larger than should be attempted under home lab conditions. Nevertheless, the only important resources I had that would be difficult to match in a well-equipped homebrew lab were a much larger test equipment budget, and other people to do some of the work.

By far the two most important pieces of test equipment were a Tektronix type 485 portable 350-MHz oscilloscope, and a Data I/O model VI programmable read-only memory (PROM) programmer. The 485 is a marvelous scope, but is much higher in performance than needed for routine measurements, even in ECL work, and is priced (even as used equipment) out of the reach of most hobbyists. I had previously used a Tektronix 150-MHz type 454 scope for TTL work, and this model should be quite adequate for ECL. A 50-MHz or 60-MHz scope such as a Tektronix type 547 or type 453 could be used effectively as long as its limitations were understood and conservative design practices were followed (more on this later).

As for the PROM programmer, this was needed because at that time Miami was, for digital systems work, an isolated area far from the bright lights of technology. In Silicon Valley, Los Angeles, or Boston, an enterprising hobbyist should be able to buy preprogrammed ECL PROMs from a distributor or even a manufacturer, although it may still be a while before they are sold over the counter in every shopping center.

ECL Transmission Lines— Image and Reality

Perhaps the single statement that scared me most, as I embarked on the development of Racal-Milgo's ECL number cruncher, was this one: "In high-speed systems, the inductance, capacitance, and signal delay along interconnections cannot be ignored. The only practical way of dealing with these factors is to treat interconnections as transmission lines."[2] This statement is, of course, literally true in a technical sense, and yet it is enormously misleading. It raises vivid mental images of huge steel towers marching across the wasteland, with long wires dangling from brown insulators in catenary curves (see again Figure 1). It tends to scare the hell out of people who are used to treating logic signals simply as wires from one point to another, as in garden-variety or low-power Schottky TTL.

The truth is that any type of logic operating at relatively high speeds has to be treated with extreme care—not just ECL. I have found that there is essentially no difference between the care that must be taken in designing a *good* high speed Schottky TTL system, with respect to interconnection practice, and that needed in designing a *good* ECL system; but the ECL system will run somewhat more than *twice* as fast, is actually easier to debug and get running, springs fewer nasty surprises on you during the checkout process, and tends on the whole to come closer to treating you right if you have treated it right.

Actually, all that the transmission-line property means in practice is that the last thing attached to each and every signal wire in an ECL computer is a resistor.

If your system is entirely wire-wrapped, as mine was, on a board with good voltage planes (more on that later), the characteristic impedance of each wire is that of a "wire over ground" and is somewhere between 100 and 120 ohms. And if you wade through all the formidable equations in *Z*'s and *i*'s in various handbooks,[10] one of the things you discover is that nothing *really* bad happens—just a few percent reflection—if there is a fair amount of mismatch between the line and the terminating resistor. Because the wire over ground on a wire-wrap board full of other wires is at a varying height anyway, and the characteristic impedance depends on that height, the characteristic impedance of that wire is bound to be "smeared out" and not very precise.

Appropriate Resistors

I used two types of resistors: thick-film, which come in 16-pin DIPs costing $1.25 to $2.50 each, depending on quantity, from Beckman, Bourns, and other vendors, with 11 individual resistors per DIP; and ⅛-watt carbon resistors, which are so tiny that the leads can be wire-wrapped around backplane pins. Most of the resistors overall were of the thick-film DIP variety, and the ratio of ECL integrated circuit (IC) DIPs to resistor DIPs was roughly 3:1. There are also single-in-line (SIP) resistor packages, and "active terminators" (Fairchild type 10014) with a nonlinear current-versus-voltage characteristic. In any case, the other end of the resistor is terminated to a supply voltage (V_{TT}) intermediate between the two usual supply voltages (V_{CC} and V_{EE}).

The *Thevenin equivalent* scheme is a second way of terminating a signal line in its characteristic impedance. This approach avoids having a V_{TT} plane at all, and presumably also inflicts less noise from the logic on the main power supplies in some cases; however, it dissipates about 11 times as much extra power per line termination as does the previous method. In the Thevenin equivalent scheme, the termination point for each signal line is connected to both V_{CC} and V_{EE} by resistors whose values are chosen to form a voltage divider (Thevenin network) such that the voltage drop produces V_{TT} at the termination point. Beckman also makes Thevenin network thick-film termination resistor packs, with four such networks per DIP.

To be sure, there are other ways of approaching signal interconnection besides my recipe for terminating each and every signal line in its characteristic impedance, especially if you are (a) skilled in linear circuit design, and/or (b) a masochist. You can simply not terminate the line, and compute out the maximum number of inches or tenths of an inch allowable for line length under each given set of conditions for each signal line. Or you can use *series termination,* in which there is a resistor between the output stage of your gate, or whatever, and the input that is being driven. (There is just one input per line, but with many lines fanning out from one original output.) Possibly in a big company environment where one is using 17-layer etched circuit boards like those used in the Texas Instruments Advanced Scientific Computer, there are real advantages to these schemes. In the wire-wrap world there aren't any, and it is better to terminate each and every signal line in a resistor and then relax, since you have thereby at one stroke slain most of the big, scary goblins of high-speed logic systems—crosstalk, ringing and reflections, limits on line length, and so forth.

Some Good News

And now for some pleasant surprises. First, 10K and 10KH outputs are *open-emitter* and may be tied together in almost the same way as TTL open-collector outputs, but with wire-ORing of outputs viewed as assertive-HIGH and wire-ANDing of outputs viewed as assertive-LOW. However, since the termination resistor has a value determined by the characteristic impedance of the signal line, one no longer has to recompute this value every time the number of driving outputs or the number of driven inputs changes, as one is supposed to do when stringing together open-collector TTL. ECL isn't slowed down much by stringing together open-emitter outputs: roughly 50 picoseconds per additional output. When five or more outputs are strung together, one may start to see minor glitches in the waveform; I never tried that. Stringing together open-emitter outputs turns out to be a valuable technique in ECL, for two reasons: It does an extra level of logic with essentially no extra logic delay and no additional gates, which together with the usual two-rail outputs makes ECL small-scale integration (SSI) much more powerful per gate package than TTL SSI. Also, it allows *in-circuit stimulation* of ECL devices while your system is running, or trying unsuccessfully to run, at full speed; *any* logic point can be tied to a logic "1" source with impunity in order to change what is happening so that you can study it. *This is a very powerful troubleshooting technique.* (It is normally forbidden in TTL troubleshooting because of an unfortunate tendency to melt IC output transistors in totem-pole devices.)

Second, since virtually any ECL IC output stage will drive a 50-ohm line, it will also drive *two* properly terminated 100-ohm lines going to different places, which is very useful, for instance, when driving a lot of memory address lines. By way of comparison, there are only a few TTL devices—the 74S140 dual NAND buffer and the 74128 quad NOR buffer, for instance—that will drive such low-impedance lines.

Third, once you have bitten the bullet and terminated a signal line in its characteristic impedance, you can stop worrying about how long that line is, at least as long as it doesn't go off the board away from the ground plane. The boards I used were roughly a foot wide and almost two feet long, and some signal lines were longer than two feet, which would be rather unacceptable using TTL gates since the usually quoted line length limit is ten inches (For TTL three-state buffers it is much longer.) ECL signals that go off the board should be differential, but even that turns out to be less frightening than it sounds, as will be discussed later on.

Fourth, in ECL the only limitation on fanout that matters is that each additional input connected to a line adds a few picofarads of capacitance, just as additional TTL or MOS inputs do in other systems; and as the number of inputs increases, the rise and fall times lengthen a bit. But instead of a fanout of 10 as for garden-variety gold-doped TTL or high-speed Schottky TTL, or of 21 as for low-power-Schottky TTL, the fanout limit imposed by driving capability is something like 92, which is as good as infinity for most purposes. I never really had to test this proposition out: it usually was not necessary to go beyond driving 10 to 12 loads with one output, except in special situations like driving memory IC address inputs with buffer gates; and even there I stayed conservative.

Voltage Planes and "Positive Earth"

ECL, even the easier-to-use 10K/10KH, should still be built on a *good* board for best results. "Good" here means that the voltage plane or planes occupy *at least 50%* of the available area of the board as it is viewed from above, say by Superman with X-ray vision if the board is multilayer with internal voltage planes. I knew where to get really deluxe boards, from a successor company (Kleffman Electronics, Minnetonka, Minnesota) to one I once worked for, with *four* complete voltage planes, but up until now these boards have not been offered for public sale. However, the designer of these boards, Gary McPherson went into business for himself, so you can try contacting him. (See the appendix at the end of this chapter.) A number of circuit-board companies do now offer wire-wrap breadboards that look satisfactory for ECL, and in some cases state such a design objective (see the list in the appendix). Augat pioneered in this area, with a three-layer board, and boards with a similar design philosophy are now also available from Excel, Garry, Mupac, and SAE. Interdyne has a rather different type of board, which also looks plausible. These boards do, of course, cost more than vector board—probably $200-$300 for one to accommodate 150 or so DIPs. In most cases the DIPs plug directly into the holes in the round pins on the board, and no additional IC sockets are needed. Figure 2 shows an Augat board in local cross-section. Reference 3 is a useful technical note available from Augat on wire-wrapping ECL logic using their boards.

One of the disconcerting facts about ECL that seems to baffle every person newly introduced to the stuff is that V_{CC}—yes, I *did* say V_{CC}—is normally specified as +0.0 volts, or "positive earth," as British car aficionados say. After all, everyone who has designed TTL or MOS systems *knows* that V_{CC} has to be +5.0 volts and that it is the *other* voltage supply that is at +0.0 volts—why, it is even *called* ground. What, then, is this V_{EE} that is specified as -5.2 volts? Why isn't V_{CC} specified as +5.2 volts and V_{EE} as ground? Certainly the logic doesn't care what the dc potential of various circuit points is relative to Mother Earth, does it? For that matter, why can't ECL run on a 5.0-volt spread between the two main supply voltages as TTL does?

It turns out that when Motorola originally instituted this now-universal +0.0/-5.2 specification, the goal that they were in a subtle way trying to achieve was to get their customers to use the best plane on the board for V_{CC} rather than for V_{EE}, in case there was any difference in the extent of the planes. The circuit properties of ECL are such that the system performance is affected much more by inadequacy of the V_{CC} plane than by, say, a V_{EE} plane that only covers part of the board and shares the same surface with the V_{TT} plane. To keep the internal workings of ECL ICs from being confused by electrical transients due to their own output stages, most of them (except the ones with particularly serendipitous internal layout) have two or even three separate V_{CC} pins. Do not, however, draw the conclusion that you must actually connect these different pins to *different* V_{CC} planes—they *don't* want you to do that, but rather to connect them *separately* to the same V_{CC} plane. It makes sense if you think about it; you don't want the potentials at different V_{CC} pins to diverge—you only want to convey the output switching noise to ground without it going through the tender internal gates.

The Kleffman boards I used had two complete ground planes and two other voltage planes, having been designed to accommodate a mixture of Schottky MSI devices with linears that often required a -5.0-volt supply in addition to the normal TTL supply voltages. I adapted these boards for ECL by using the ground planes for V_{CC} (after all, it is at ground), the TTL V_{CC} planes for V_{EE}, and the -5.0 plane for V_{TT}.

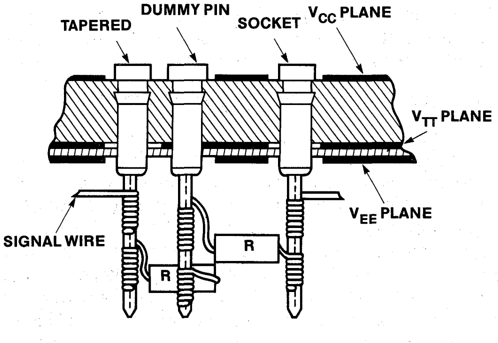

Figure 2. Cross-Section of Augat Wire-Wrap Board

Now that the upside-down supply-voltage polarity issue has been disposed of, you will know what I mean when I state that V_{TT} (or the equivalent individual termination points if the Thevenin-equivalent voltage-divider scheme is used) is normally specified as -2.0 volts.

Decoupling Capacitors

In any high-speed logic system, not just ECL, there should be an easy path for high-frequency noise to get between the two main supply-voltage planes without passing through the logic and confusing the hell out of it on the way. Lower-frequency noise is usually dealt with by connecting a fairly large tantalum electrolytic capacitor, say 22 microfarads or larger, between the two main supply voltages (for ECL, V_{CC} and V_{EE}) at the point where they are brought onto the board, and perhaps at other points on the board also. High-frequency noise is similarly shorted out using little ceramic disk capacitors scattered all over the board, mingled with the semiconductors.

Although I have seen printed recommendations as mild as using an 0.01-microfarad disk capacitor for every few ICs, again—as in the case of signal-line termination—my recipe calls for doing it right everywhere to start with and finishing off the goblins for good. Here, doing it right means using one 0.1-microfarad disk capacitor (ten times as large) for each and every DIP on the board, which thoroughly slays many noise problems otherwise likely to be encountered in either ECL or Schottky systems. AVX (née Aerovox) is the brand I have used, and there are some other vendors also whom I haven't personally calibrated. These capacitors are physically quite small, and cost in the range of 20¢ each in modest quantities.

The reason why I—and other people who write ECL applications notes for semiconductor manufacturers—insist on using *disk* capacitors for this application is that they provide the best practical way to get *just* a capacitor, without at the same time getting an inductor and a resistor willy-nilly into the bargain. The last thing you need is to have all your little decoupling capacitors turn into little tank circuits scattered all over your printed circuit board.

The Kleffman boards I used, and some of the commercially available boards such as Augat's and Interdyne's, provide yet one more weapon in the battle against supply-voltage noise. A pair of supply-voltage planes are physically separated by only a very small thickness—an 0.004-inch mylar layer in the case of the Kleffman boards—so that there is in effect a *distributed* capacitor, sufficiently large to severely restrict the magnitude of the very-highest-frequency noise (say 150 MHz and up), between *all* points on these planes.

Keeping Supply Voltages Smooth

The ever-present possibility of ac noise on the power-supply voltages is probably the real reason for the general industry concern with power-supply-voltage margins in digital logic. It isn't hard today to build a fairly economical power supply with very tight regulation—0.1% to 0.2%—according to what one seasoned power-supply designer once told me. The non-trivial part is getting that precise voltage conveyed to each and every DIP. 10K ECL, by the way, is normally specified as having a ±10% supply-voltage tolerance, and internal voltage compensation. In contrast, normal commercial-grade TTL, and 10KH ECL are specified to tolerate just ±5% supply-voltage misbehavior.

Why, then, are 10K and 10KH specified for a 5.2-volt main-supply-voltage spread rather than a 5.0-volt spread if it is so tolerant of funny power-supply levels? Simply for optimization of other circuit parameters. Under normal circumstances, 10K ECL, and most other ECL, will run perfectly well on a 5.0-volt spread.

Although I did not find it necessary to do this in my system, and I doubt that you will either, it is worth noting that an ECL system can be designed to present an *invariant* load to the power supply—in contrast to a TTL system, since some TTL gate packages may draw up to seven times as much supply current with all outputs LOW as with all outputs HIGH, and thus full-word-complementing operations may result in high-frequency supply-voltage hiccups.

One of the world's fastest computers, the CRAY-1, capable of 138,000,000 floating-point arithmetic operations per second on a sustained basis, was designed according to this invariant-power-supply-load philosophy.[4,5] A number of rather extreme measures were taken in the CRAY-1 to control various types of noise, for obvious reasons. The non-memory portions were designed largely with simple gates (Fairchild type 11C01 essentially align to the more recent Monolithic Memories or Motorola 10H109) having "two-rail" outputs (the output and its complement, on separate leads), with both outputs terminated even if both are not used. In this configuration, each 11C01 presents an invariant load to the power supply. Single-rail output devices such as memory ICs use Thevenin termination. The 72-bit (64 information plus 8 checking), 1,048,576-word CRAY-1 main memory was comprised entirely of 1K-by-1 ECL memory ICs essentially similar (and actually bought to a longer access-time specification) to the ones I used. Of course, I only needed about 200 of them rather than 74,000 or so.

Off-Board Interconnection

Probably one could, at least in some cases, get away with running a properly terminated signal line right off one board onto another, if all the precautions already discussed were taken. I never tried it. In the first place, one can't assume that the voltage-plane potentials on one board *exactly* match those on some other board the way they're supposed to, even if one has used 0.1 microfarad capacitors like popcorn as I have recommended. In the second place, there has to be some way to keep the signal lines at the same characteristic impedance, without discontinuities, as they leap through space between boards—and, worse yet, to keep them shielded from various forms of electromagnetic interference (such as each other) now that they are no longer safely close to a ground plane or other voltage plane.

Again, there is a simple, seemingly drastic, very effective way of solving the problem which pretty well decimates the goblins. As I stated in the previous section, many ECL gates have two-rail outputs. (This may be an unfamiliar idea to TTL chauvinists; except for flipflops, one multiplexer configuration, and a rather little-known two-rail buffer called the 74265, TTL devices don't usually offer this feature.) ECL gate circuit parameters are such that *any* two-rail gate can be used to drive a differential line, with all of the implied advantages of common-mode-noise rejection and insensitivity to temperature and dc-voltage discrepancies between different boards. Since such a line may in principle be as long as a few hundred feet before purely circuit-design-parameter alligators start snapping at you, there is no abrupt length limit of concern to a hobbyist. Of course, remember that a nanosecond is approximately a "light foot" and that electrons in a wire only travel about 2/3 as fast as light travels. Thus you may observe, at least if you can borrow a 485 scope for a while, that each 6 inches to 8 inches of signal line requires another nanosecond for the signal to traverse it, for differential as well as for single-ended signals, I was at one point rather startled to realize that some 15-inch signal lines were actually a *bigger* delay factor in one data path than a whole row of 2-nanosecond buffers with short signal lines coming and going.

At the other end of the differential line, on the other board, one uses a *differential receiver* element with a resistor between the two differential line ends. These elements come in four flavors: three distinct types of triple elements with two-rail outputs (types 10H116 and 10116 for plain vanilla, 10114 for hysteresis, and 10216 for extra blazing speed) and one (types 10H115 and 10115) with quadruple elements with single-rail outputs.

For more details see Reference 6, which also describes how to turn one of the triple two-rail devices into a Schmitt trigger circuit. Two cautions: First, any *unused* elements in a differential-receiver DIP should be "strapped" to force their outputs into one logic state or the other, as otherwise they will hover right at the logic threshold point and the on-chip bias networks will get screwed up and confuse the elements that *are* being used. Also, we found that the resistor values suggested for use with the receiver elements by Reference 6 were not the right ones for our interconnection system, and we wound up using resistors with values close to 300 ohms for all three resistors shown in Figure 3 of that reference. Also each output drives the same number of gate inputs; if necessary, dummy inputs of unused gates are used to "pad out" lines to achieve this balanced position.

Cabling

There are probably other acceptable physical means for getting these differential signals from one board to another, but the one I recommend is *flat ribbon cable,* available in various forms from 3M, Augat, Elco, Spectra-Strip, and probably other companies. Specifically, what I have used is 40-wire 3M cable, which is physically surprisingly small. Many such cables stack neatly in a small thickness, and fairly abrupt turns and at least some limited hinge action are possible. 3M supplies little press-on connectors and a tool to crunch them into place, one at each end of the cable.

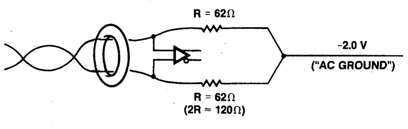

Figure 3. Receiving board termination scheme.

The electrical shielding properties of this cable are *excellent* if one does not get greedy about how many logic signals pass through a single cable. There should be one or more ground wires at each edge of the cable, and alternating signal and ground wires within the cable. This means that one can *only* transmit *nine* differential logic signals in one 40-wire cable, since doing just that according to the recipe demands a *minimum* of 37 wires. The cable format is, of course,

$$G\ G\ S_1\ G\ \overline{S}_1\ G\ \overline{S}_2\ G\ldots G\ S_9\ G\ \overline{S}_9\ G\ G\ G$$

When one stacks several such cables, the signal-ground-signalbar-ground philosophy should also prevail along the "z axis," that is, in the direction perpendicular to the plane of each cable as one goes through successive cables. Thus, in the cables immediately above and below the one whose format has just been given, there would be three edge G's on the left and two on the right, and so forth, so that the signal wires are staggered.

For shipping differential signals around on a board, say from ribbon cable connectors to differential receiver ICs, use *twisted-pair* wire. To make your own, begin with two 15-foot lengths of different colors of wire-wrap wire, secure one end of each wire to something across the room from yourself, clamp the other end of each wire into the chuck of a small electric drill, back up holding the drill until both wires are taut and close together, and gun the drill for a few seconds until the two wires are twisted together for their entire lengths with a satisfactory number of twists per inch. Inevitably there will be a slightly greater density of twists at the end of the wires close to the drill, but don't let that worry you.

Another Off-Board Interconnection Scheme

The following scheme is the work of Norm Winningstad, then of Floating Point Systems, who is expert in such matters. I haven't tried it myself, but there are plausible physical reasons why it should work even better than the previous scheme. It is different in three respects:

First, within each ribbon cable, use a different signal format, namely,

$$G\ G\ G\ S_1\ \overline{S}_1\ G\ G\ S_2\ \overline{S}_2\ G\ G\ldots G\ G\ S_9\ \overline{S}_9\ G\ G\ G$$

This makes good sense: the closer S_j and $\overline{S}_j$ are physically, the more nearly their externally-produced electromagnetic fields cancel. The *z*-axis signal-ground-signalbar-ground approach remains in effect as in my method, since from one cable to another the signals aren't necessarily related in phase and thus won't cancel. *Don't* try to economize by using just one ground wire between each pair of signal wires; having two wires there provides *much* better isolation, since stray fields from each signal-wire pair basically have a whole ground wire to themselves on each side.

Second, use the termination scheme illustrated in Figure 3 on the receiving board, rather than the one described by Reference 6. Any of the four aforementioned types of 10K or 10KH differential receiver elements will work. The *bifilar ferrite toroid* is optional. ("Bifilar" means that both wires go through it; a "ferrite toroid" is, of course, a small doughnut-shaped magnetic core.) If it is used, however, it will nicely throttle unwanted common-mode noise transients while letting the desired signal pass almost unscathed, since the latter produces almost no *net external* electromagnetic field to sweep across the toroid—at least, if the toroid is physically close enough to the –2 volt termination point so that the signals on both sides of the differential line are still essentially in phase at that point, and hence are equal and opposite.

Third, use flexible plastic sheets between successive stacked ribbon cables to increase the *physical distance* between their wires. Note that this plastic is not a conductor—it isn't intended to be. Shielded ribbon cable is a different game, outside the scope of this discussion; for one thing, the stuff with built-in shielding is physically much stiffer and doesn't bend as readily.

Keeping It All Cool

If one firmly grasps a 1K-by-1 ECL memory IC (type 10415A/10146) after the computer has been running for a few minutes, one can get second-degree burns. This IC type dissipates as much as 3/4 watt, and we measured ceramic DIP case temperatures as high as 60°C. Up-down counters (type 10H136/10136) and hex D-flipflops (type 10H186A/10186) also run pretty hot, although not quite that hot. Oddly enough, 256-by-4 PROMs (type 10149) run much cooler—about 45°C. The average power dissipation for all ICs in the entire midicomputer, including SSI and medium-scale integration (MSI) types as well as large-scale integration (LSI) types such as the 10149 and 10415A, was about 1/3 watt. Of course, probably about 1/10 of that was not dissipated as heat, within the ICs themselves but rather within the termination resistor DIPs.

Despite all that, we encountered no problems attributable to heat. The ICs simply sat out in the open, on large boards that were mounted vertically like pages of a book on a central vertical post, free to flop back and forth through a small arc since they were interconnected by ribbon cable as described. Although we had a forced-air cooling scheme figured out, we never had to use it and relied purely on convection and radiant cooling. I also found that ECL ICs, once installed and running properly, very rarely died of natural causes, at least as compared with TTL ICs in similar applications. Probably the very high percentage (about 78) of voltage plane on our boards helped a lot to conduct the heat efficiently away from the ICs. Also, Augat has done studies which show that wire-wrapped boards run cooler than etched boards with equivalent circuitry on them, partly because wire-wrap pins and wires offer a lot of surface area from which to lose heat. Today Augat offers a similar "book" packaging system.

An ECL system installed within a closed metal cabinet, particularly if the boards are mounted horizontally, should doubtless be cooled by some more active technique, such as forced air. As for what the big-machine people do, Control Data's big computers are Freon-cooled, with lots of little pipes running along chassis structural members. The CRAY-1 uses not only Freon cooling but heavy gauge heat-conductive copper sheets. (And, even though the CRAY-1 mainframe itself is physically small enough to fit into your spare bedroom, the auxiliary cooling apparatus might drive you out of the rest of the house.) Some large IBM computers use chilled-water cooling. Some day, an ingenious hobbyist trying to cool a really massive homebrew ECL system may wind up using the refrigeration unit from a used Sears Coldspot, but open-rack convection cooling or forced air should do the trick for most systems.

There were a few Saturdays when we worked on the Racal-Milgo system, and Plant Engineering forgot to turn on the air-conditioning until the middle of the morning; and in Miami during the summer an unventilated room is bad news for people as well as for computers. The system didn't run too well on those days until the air-conditioning had been on long enough to pull the temperature in the lab down below, say, 90°F. However, we really had no reason to believe it was the ECL that was giving problems. We were using a semi-homebrew IM6100-based, PDP-8-compatible microcomputer to control the larger ECL machine, which is also a likely technique for a hobbyist who already has one or two other microcomputers. Most of the system reliability problems that we were actually able to pin down turned out to be trouble with the 2102-type MOS memory ICs used with the IM6100, and the specific symptom of trouble we had on those hot Saturday mornings was usually inability to load the ECL midicomputer instruction memory from the IM6100. Nevertheless, if you do choose to rely on open-rack convection cooling, your ECL homebrew number cruncher may run a bit better if you keep your spare bedroom at a temperature that you also find agreeable.

If Your Scope Isn't Fast Enough

ECL 10K output stages have a nominal logic swing from about –0.900 volts (considered to be a "logic 1" or "HIGH") down to about –1.750 volts ("logic 0" or "LOW"), with the logic threshold being around –1.290 volts. (*Now* you know what these negative numbers *really* mean.) If you have followed the interconnection recipe of the preceding paragraphs faithfully, you should see picture-book square waveforms everywhere, although they do look just a bit cleaner at the end of a signal line close to the termination resistor than at points along the way. Even with a 485 scope, which shows every little wiggle, the ECL waveforms I observed looked very clean compared to the grassy ones sometimes seen in high-speed TTL systems.

If you must use a slower scope for economic reasons, you will, of course, observe even cleaner waveforms (which aren't exactly real) with slightly rounded corners, which may not deceive you very much about anything essential as long as you remember *why* it is that they look so clean. The principal danger is that, now and then, there will be a glitch on some signal line that is insufficiently wide to show up on the scope, or at least to look to you as if it is of sufficient magnitude to reach the logic threshold—when, all the while, here is an up-down counter (type 10H136/10136) or hex D-flipflop (type 10H186A/10186) or other edge-sensitive device whose clock input is connected to that signal line, which sure is acting as if it is getting an edge at just about that time. Probably it is, and you just can't see it because your scope has smoothed it out for you. Realizing that such must be the case, you have your choice of (a) getting hold of a more expensive scope, or (b) "reading between the lines" of what your humbler scope is telling you, terminating that clock line better, and seeing if the problem doesn't then go away. Or perhaps the clock-line glitch is really due to a switching hazard in the logic you designed to control that clock line.

Product Families

To oversimplify things a bit in a manner meaningful to a ones-and-zeroes type like me, ECL is one type of *current-mode logic.* The state of an ECL gate is determined by which of the two output legs the main current is being steered through, and the resulting voltages at the output points are interesting side effects but are not the basic switching phenomenon.

In TTL and MOS, on the other hand, the output *voltage* states are where the action is, and the currents tag along after the voltage as interesting (and often inconvenient) side effects. In Schottky TTL logic, for instance, the voltage states you actually see on a scope are about +0.2 volts for a logic 0 and +4.1 volts for a logic 1. Incidentally since high speed Schottky TTL rise times are probably a bit *faster* than the 3.5-nanosecond to 4-nanosecond times characteristic of ECL 10K; you may notice that the "voltage slew rate," or whatever that parameter should be called, is many times greater for Schottky TTL, and is in fact roughly equal to the corresponding rate for the very fastest ECL families.

There are lots of custom families made by IC manufacturers for direct sale to large computer companies that fall into the general category of current-mode logic. However, most of these are not available for sale to the general public. There are seven product families, more specifically considered to be ECL, that are (or have been) sold to all comers:

- ECL I, contemporary with diode-transistor logic (DTL) and now obsolete and not used in new designs.
- ECL II, contemporary with and somewhat akin to H-series TTL, and recently also declared obsolete by its manufacturer (Motorola).
- ECL III, extremely fast, but with only a modest selection of SSI and MSI types.
- ECL 10K, the one used in the Miami number-cruncher.
- ECL 10KH, fully compatible with ECL 10K, but with twice the speed, 75% better noise margins, and voltage compensation.
- ECL 95K, much like 10K, but a marketplace also-ran.
- ECL 100K, compatible in speed with ECL 10KH, but not compatible with ECL 10K.

The first five of these families were introduced by Motorola, and the last two by Fairchild. Monolithic Memories and Motorola both supply 10KH parts. Fairchild, Motorola, and Plessey also have various other ECL products that are not really organized into families—for instance, Fairchild's 11C01 OR/NOR gate (used in the CRAY-1), which is a 100K-technology device offered in a non-100K package. The last five families are, or can be made to be, electrically compatible so that with some care devices may be mixed in a system.

The Even Faster Stuff

A hobbyist willing to hand-solder, rather than wire-wrap, all interconnections to that part of his system might with due care succeed in making some limited use of ECL III and/or ECL 100K.[10] ECL 10KH is about as fast as these two, but its slower edge speeds make it easier to wire-wrap. (A few turns of wrapped wire, it turns out, can function all too well as an inductor when hit with the 900- or 700-picosecond edges respectively characteristic of those families, and the resulting impedance discontinuities make reflections.) The technology required to build a system of any size using one of these families, however, remains more difficult at the present time than the technology for ECL 10K. References 7, 8, and 9 describe a practical approach to wire-wrapped use of ECL 10KH and 100K.

There *are* two ECL III devices of some interest to a hobbyist: type 1648, which is called a voltage-controlled oscillator; and type 1658, which is called a voltage-controlled multivibrator. There is no announced ECL 10K or 10KH product matching either of these descriptions, and they are useful. I used a 1648 once in an otherwise all-TTL system to provide a hand-adjustable system clock source for testing "clock margin"—that is, how fast the system could be made to go before it failed. The 1648 worked fine after being well shielded.

Those ECL 10K parts having type numbers of form 102XX (106XX in military temperature) form a subfamily with appreciably different properties. Hand-soldered connections are also advisable when using any of these. They are quite a lot faster than their normal ECL 10K equivalents, and consume essentially no more power, so they sound very much like 10KH parts. Alas, this greater speed has been obtained by cutting very short the leisurely rise and fall times deliberately designed into normal ECL 10K. To oversimplify a bit, normal ECL 10K rise and fall times are perhaps 3.5 to 4 nanoseconds—much longer than the nominal logic delay of 2 nanoseconds or so; whereas 102XX rise and fall times are probably not much different from the logic delay, which I have observed to be about 1.25 nanoseconds. Reluctantly, I concluded that system noise problems would be minimized by restricting the use of 102XX parts to those situations where that last ounce of speed is really required, for instance in the clock generation circuits. The only example of proven crosstalk trouble in the Racal-Milgo midicomputer was due to an unnecessarily long wire being driven by a type 10212 buffer; it was eliminated by relocating a few ICs so that that buffer was closer to its load.

10KH parts, like 10K parts, have longer rise and fall times deliberately designed in, in the range of 1.5 nanoseconds.

One other simple and plausible use of ECL 10KH is for building high-speed, fixed-frequency oscillators. Two type 10H105 OR/NOR gates on the same IC in series make a dandy oscillator (see Figure 4), which should generate frequencies of at least 100 MHz depending on which sample you use. Don't try to build this type of oscillator with a single gate—it won't even oscillate, but just hang in there with its output at about the threshold voltage. If swapping ICs around doesn't get you the frequency you are shooting for, try hanging very small capacitors on gate outputs to slow them down. (Caution: If you insist on using ECL III instead of ECL 10KH, the speed of oscillation may depend on the ambient temperature.)

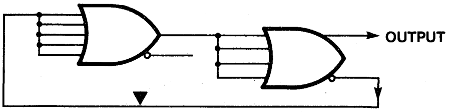

Figure 4. Fixed-Frequency Oscillator.

Logic-Drawing Conventions

If one wishes to think "assertive-HIGH"—that is, the more positive of the two output voltage states represent a 1 and the negative state represents a 0—then it follows that in TTL logic the simplest and most natural gate structure is the NAND gate, whereas in ECL logic it turns out to be an OR/NOR gate. It is fairly clearcut in both cases what is really the simplest circuit for the silicon people to build.

Since the ORing together of "minterms" seems to be more natural to human psychology than the ANDing together of "maxterms," part of learning to use ECL consists of learning to use "mixed-logic" conventions in some form. These conventions allow you to think of—and draw—a given physical gate circuit as performing either an OR function or an AND function, and then to consider as a separate issue the assertiveness of that gate's input and output signals.

Soaking up the mixed-logic viewpoint should also be part of learning to design with TTL, but unfortunately it isn't always.[10,12] In fact, many erudite polemics have been written defending older and less general viewpoints that I feel are now best understood as evolutionary stages on the way to the full-blown mixed-logic viewpoint. One of these older viewpoints is the one that results in the dozens of busy little black and white triangular flag symbols on the inputs and outputs of ECL parts as drawn on many manufacturers' data sheets.

I can only say that once I forced myself to give up my Bronze-Age viewpoint and learned to use mixed-logic conventions, within a week I was wondering why I had ever used anything else. If one uses an additional logic symbol to denote "psychological inversion"—implying that although the electrical polarity of some signal (say, for example, $\overline{\text{BANANAS}}$) *hasn't* changed one's perception of its meaning or "psychological polarity" *has* changed (in the example, to YES WE HAVE NO $\overline{\text{BANANAS}}$), then logic drawings can be made almost as semantically precise and self-checking as logic equations. This is *not* an academic exercise; it helps you spot real mistakes before you wire the machine up wrong and waste a lot of time trying to figure out why it isn't working.

Reference 13 is an eminently sane paper on this whole dogma-ridden subject; I have relied on it for many years for guidance as to logic drawing conventions. Recently there have been more papers in the same vein. I have just one minor quibble with the author: the symbol suggested there for psychological inversion, a small line drawn across the signal line at right angles, tends not to show up too well on blueline copies. A little solid triangle or arrowhead drawn next to the signal line shows up much better, and there is a hole of the right shape on most templates. (See Figure 4 again.)

Figure 5 shows the same physical gate element, one of the two-input elements from a type 10H105 triple OR/NOR gate part, drawn first with assertive-HIGH inputs as an OR/NOR gate and second with assertive-LOW inputs as an AND/NAND gate. You get the idea. The pins are all still in the same relative positions. If one uses the standard inversion bubbles correctly, one doesn't really need the additional symbols such as triangular flags—they in fact introduce one too many degrees of freedom and just confuse the issue. I suggest the proper use of mixed-logic conventions as a sort of software adjunct to the rest of my homebrew ECL recipe.

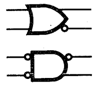

Figure 5. Two representations of the same gate.

Finale

I have used references sparingly, because many of them tend to give ECL system design an air of awesome complexity and desperate peril, and this is exactly what I am trying to tell you isn't necessarily so. However, if you get seriously into ECL you probably will want to get hold of Monolithic Memories 1985 LSI Datebook, which has full specifications of its available ECL 10KH parts. Good luck, and may the Force be with you.

Acknowledgements

There are several people who were at Racal-Milgo when I was, without whose efforts the number cruncher would never have happened and I wouldn't have been able to tell you about it. Two deserve particular mention. Dick Joerger was the "sharp technician" referred to in this paper and he built most of the machine with his own hands; today he is a senior engineer. Rick Johnston (now of GE) did the preliminary design of the control section and improved it greatly from what I had originally planned; he made it a pipelined, overlapped, stored-microprogram machine with a self-restarting main timing circuit.

There are also three electronics industry veterans without whose wise opinions I would not have had the temerity to try to build the machine out of ECL. They are Stan Bruederle of Dataquest (at that time of Signetics), Rob Walker of LSI Logic (at that time still at Fairchild), and Norm Winningstad of Floating Point Systems. Everything each of them told me about ECL turned out to be absolutely correct, and much of it appears in this paper. If I am wildly off-base on any topic, it's because I didn't get their complete message.

References

1. Cann, R. 1979, 1980. "An analysis/synthesis tutorial." *Computer Music Journal* 3(3): 6–11, 3(4):9–13, 4(1): 36–42.
2. Fairchild Semiconductor. 1977. The ECL Databook. Fairchild Semiconductor, Mountain View, Calif.
3. Doucet, L. 1974. Packaging High-Speed ECL Integrated Circuits. Augat, Inc., Attleboro, Mass.
4. Russell, R. 1978. "The CRAY-1 computer system." *Communications of the Association for Computing Machinery* 21(1): 63–72.
5. IEEE Computer Society. 1976. "CRAY-1: The smaller supercomputer." *IEEE Computer* 3: 53.
6. Blood, W. 1974. Interfacing with MECL 10,000 Integrated Circuits. Application note AN-720. Motorola Semiconductor Corp., Phoenix, Ariz.
7. Gilbert, B., and L. Harris. 1980. "Advances in processor architecture, display, and device technology for biomedical image processing." *IEEE Transactions on Nuclear Science* NS-27(3): 1197–1206.
8. Gilbert, B., L. Krueger, and R. Beistad. 1980. "Design of prototype digital processors employing subnanosecond emitter-coupled logic and rapid fabrication techniques." *IEEE Transactions on Components, Hybrids, and Manufacturing Technology* CHMT-3(1): 125–134.
9. Gilbert, B. 1982. "Architectures, tools, and technologies for the development of special-purpose processors." In S. I. Kartashev and S. V. Kartashev, eds. *Supersystems of the Eighties*, vol. 2 of *Modular Computers and Systems.* Englewood Cliffs, N. J.: Prentice-Hall.

10. Blood, W. 1980, *The MECL System Design Handbook,* Motorola Semiconductor Corp., Phoenix, Ariz.

11. Grossman, M. 1978. "Fast logic no problem for wire-wrapped boards." *Electronics.* 6 December 1973.

12. Grossman S. 1973 "ECL, TTL Get Together on same board." *Electronics* 6 December 1973.

13. Kintner, P. 1971. "Mixed logic: A tool for design simplification." *Computer Design* 8: 55–60.

Appendix

Here is a list of vendors of various items that you will need, arranged in alphabetical order by topic. My listing them here does not imply any warranty that you will find them utterly perfect, but I have had positive dealings with many of them.

ECL-Grade Logic Breadboards

Augat, Inc.
33 Perry Avenue
Attleboro, MA 02703

Excel Products Company, Inc.
401 Joyce Kilmer Avenue
New Brunswick, NJ 08903

Garry Manufacturing Company
1010 Jersey Avenue
New Brunswick, NJ 08902

Interdyne, Inc.
14761 Califa Street
Van Nuys, CA 91411

Gary C. McPherson
P. O. Box 1044
Minnetonka, MN 55343

Mupac Corporation
646 Summer Street
Brockton, MA 02402

Stanford Applied Engineering, Inc. (SAE)
340 Martin Avenue
Santa Clara, CA 95050

Flat Ribbon Cable

Augat (see above)

Elco Corporation
2250 Park Place
El Segundo, CA 90245

Eltra Spectra-Strip
7100 Lampson Avenue
Garden Grove, CA 92642

3M Company
Industrial Electrical Products
3M Center
St. Paul, MN 55101

Ceramic Disk Capacitors

AVX Ceramics
P. O. Box 867
Myrtle Beach, SC 29577

Centre Engineering
2820 East College Avenue
State College, PA 16801

Resistor Packages

Beckman Instruments, Inc.
2500 Harbor Boulevard
Fullerton, CA 92634

Bourns, Inc.
1200 Columbia Avenue
Riverside, CA 92507

ILC Data Device Corporation
Airport International Plaza
Bohemia, Long Island, NY 11716

Wirewrap Equipment

Gardner Denver Company
1333 Fulton Street
Grand Haven, MI 49417

OK Machine and Tool Corporation
3455 Conner Street
Bronx, NY 10475