HARDWARE HACKER

The Best Mouse Surface Around; Simulati g the Human Brain Using CMOS Inverters; a New A/D Converter

By Don Lancaster

We'll begin with some updates on our previous columns. The part number on the *Hewlett-Packard* shaft encoder integrated circuit is the HCTL-2000. This dude goes by the horrendous name of a quadrature decoder counter interface IC. It automatically provides all the direction sensing, noise elimination, pulse counting and bus interfacing needed to connect a shaft encoder to a microprocessor or a personal computer.

As yet another source on piezoelectric goodies and information, check out *Piezo Electric Products*. Besides some super-quiet cooling fans that use vibrating resonant blades, they are into exotic stuff such as using piezo generators to recover electrical power from car to truck mufflers.

And some more on the SCUZZY interface. A good tutorial series is appearing in *The Computer Journal,* starting with issue #22. Also, NCR has a bunch of plug-and-go modules that will handle most of the SCSI interface firmware for you. Adaptor cards for personal computers start around $300 in single quantities. These prices are certain to drop in the near future.

As always, this is your column and you can get technical help by calling. Use the number shown in the end box. I also have lots of freebies available.

What is the penultimate mouse surface?

Several Tucson cave divers have put me onto a mouse working surface that is much better and much cheaper than just about anything else. It beats most commercial products whiskers down.

So, run down to your friendly local neighborhood divers supply house or scuba shop and pick up a square foot or two of ⅛ inch nylon wetsuit material. The cost is around a dollar per square foot and it even comes in decorator colors.

Round the corners and use the material fuzzy side up.

How does the human brain work?

That's a very good question and a very exciting one as well. There are lots of he vy duty people doing heavy-duty studies on modeling and understanding the human mind. In fact, several firms now h ve ridiculously expensive emulations for u-percomputers and high-end minicomputers that will model the neuron network of the brain.

What does this expensive software d to a dino supercomputer? It converts it i to a model for a group of circuits that ny eighth grader could throw together on a kitchen table, using parts from *Mod rn Electronics* advertisers, and paid for ou of pocket change! So, I'll make this prediction: The next major breakthrough in artifical intelligence will not come from he artificial intelligence hotshots, nor the niversities, and certainly not from the m litary. It will instead be done by two jun or high school students whose only lifet me goal is to embarrass their science teacher.

Let's start with the basics.

The human brain seems to be made up of two basic parts. The "active" par is called a neuron, and looks and beha es suspiciously like a plain old nickel CMOS inverter. The "passive" part is calle a synapse and looks and acts more or ess like a penny resistor.

Neurons are rather small, being some 50 microns in length or something around 0.002 inch. There are bunches of them, ranging from a few thousand for a simple organism, up to as many as a trillion or more in humans and dolphins.

Neurons communicate with each other through the synapses. A single neuron may be connected to as many as 10,000 other neurons with as many synapses.

In computer terms, we seem to have a massively parallel, 3-D asynchronous system. It also seems to be the relative state of the neuron "interconnectedness" that is storing data rather than placing any one piece of information into any one particular place. This goes by the fancy name of "associative memory."

Some very interesting consequences of a massively parallel associative memory are that the zillions of parallel connections more than make up for the relatively slow speed of any individual device. Another consequence is that you may be able to damage or destroy part of the memory and it may recover, much in the same way that you can scratch a hologram without destroying any specific part of its image. Which may explain how stroke victims may eventually regain some abilities, even after obvious and massive brain damage.

Most often, the brain works only on a

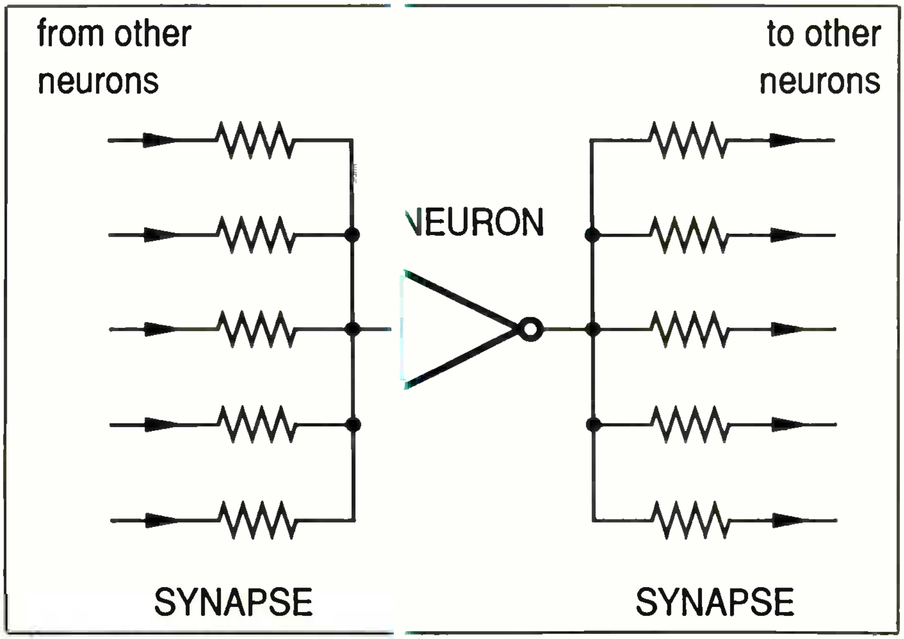

Fig. 1. One pos ible human-brain model.

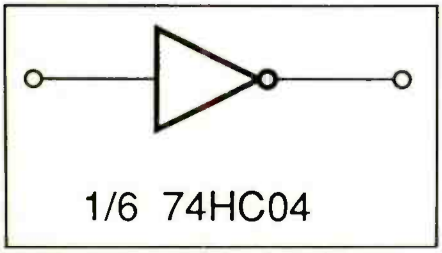

Fig. 2. A CMOS threshold detector.

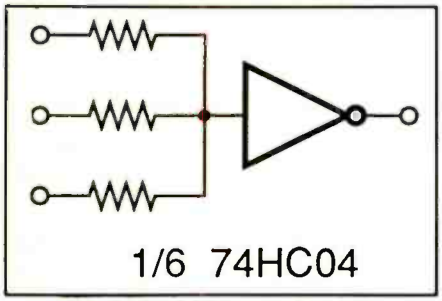

Fig. 3. A CMOS majority gate.

good solution, instead of the best of all possible solutions. You could even argue that the best possible solution to a problem is often both obsessive and energy inefficient. And nature would be on your side.

There's a very famous computer problem called the "traveling salesman routine." In it, a salesman has to visit all the cities on his route. Naturally, he only wants to go a minimum distance to do so. For more than a few dozen cities, the problem solution takes forever, even for supercomputers using the newest algorithms. There's all sorts of uses for this sort of thing, ranging from telephone calls to, unfortunately, star wars.

Here's the neat part: A kitchen table full of resistors and logic gates modeled in a neuron net can solve the same traveling salesman problem that would take a supercomputer months to do. And it does so in a millisecond or less!

The only little gotcha is that the neuron net will come up with a only very good solution, instead of the best possible one. But who cares? Certainly not the traveling salesman.

There's even a new name for human brain modeling and studies. Instead of software and firmware, we now have—*wetware.*

How can we model the brain?

Figure 1 shows part of a neuron net model. For a neuron, we can use a plain old CMOS inverter, say ⅙ of a 74HC04. We'll stick with a +5-volt dc supply, since this seems to be the in thing to do. For a synapse, we'll use plain old resistors. These will often be in the 4.7k to 470k range.

A real neutral net works something like this: Currents routed through several synapse connections will "fire" a neuron if the currents get strong enough. The "fired" neuron pulse is then passed on to yet other neurons through extensive synapse interconnections.

In some instances, the time rate of firing will change with stimulus intensity. For instance, the color red might fire the red sensors in the eye at a higher rate than the blue or green sensors. In others, synapse pulses have to build up or accumulate until they are strong enough to fire a neuron. Chemical changes are most likely involved, similar to the charging and discharging of a battery. Calcium seems to play an important role. The "interconnectedness" of the neural net will then decide what "thinking" action is to take place.

Probably the best way to start off on all this is with some reading of what others have done and are up to. In particular, check out the May issue of Science 86, and the March 24, 1986 and more recent issues of *EE Times.* For a related subject, check out the April 11, 1986 issue of *Science,* where the complete genetic code for the red, blue, and green vision sensors has been fully cracked for the first time.

There's lots of reasons why you might like to study neuron nets. Firstoff, they are fascinating in themselves and can give you anything from a quick school report to a lifetime of dedicated study.

Second, they can be profitable key to new software that can handle "fuzzy" data that gives today's programs fits. Things such as speech recognition, robotic vision, mailing list cleaners, spelling correctors, and, of course, chess. Plus a great heaping bunch of new applications presently unthunk of.

Third, eventually we will have some way of real-time interacting with the human brain, literally being able to network brilliance, in addition to being able to write

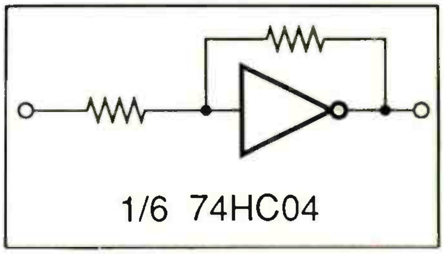

Fig. 4. A CMOS linear amplifier.

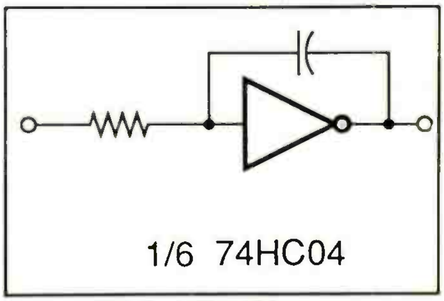

Fig. 5. A CMOS integrator.

"wetware patches" that can "cure" certain forms of mental illness.

Besides, think how much money you could earn by selling little blue boxes to used car salesmen that have three buttons on them marked "enters lot," "gets interested," and "closes sale"?

What can a "neuron" logic inverter do?

You don't have to have an exact model of a neuron or a synapse to study massively parallel 3-D computer architectures that use extensive feedback. Even with plain old CMOS inverters and plain old resistors, you might be able to come up with new ways to thinking and new ways of doing things.

But what can you do with a CMOS inverter? The correct answer, of course, is just about anything. Let's look at a few quick examples.

In Fig. 2, we can use the inverter itself as a threshold detector. Anything less than half the supply voltage drives the output low, while anything more than half the supply voltage drives the output high. Thus, by summing currents above a given threshold would actually "fire" your inverter.

In Fig. 3, the CMOS inverter behaves as a majority gate. The output will go low only if two of the three resistors are positive. Put another way, your neuron fires only if the Ohm's Law voltage produced by net currents from all inputs exceeds the magic "half way up" threshold.

In Fig. 4, we have a linear amplifier that we build by feeding back the inverter's output to its own input. This forms an operational amplifier. The gain in this case equals the ratio of the output resistor to the input resistor. You can extend this to just about anything a "real" op-amp can do. Additional inputs will be summed. Extra resistors from the input to +5 volts or ground will bias the output off-center. With enough bias, you can get half-wave or even full-wave rectification.

Figure 5 is an example of an integrator. The capacitor will build up a charge that represents the past history of input currents. For instance, if the capacitor is first discharged and the input is made positive, you will get an output that starts at the positive supply and linearly ramps downward. One use is for square wave to triangle wave conversion.

A regenerator, or snap-action circuit is shown in Fig. 6. Here we use positive feedback to create hysteresis. This circuit is also called a Schmitt trigger. Use this circuit to change a slowly changing and weak input into a "wall-to-wall" higher power output.

Figs. 5 and 6 can be combined to produce a square-wave generator. Do you see how?

Figure 7 is a pulse or edge detector. A single pulse is output every time the input is suddenly brought low.

Finally, Fig. 8 shows how to build a latching memory by connecting two inverters back to back. If one inverter's input is high, it holds the other one low, and vice-versa. And that, of course, is what leads to all of electronic memory.

For more details on gates and inverters, check into both volumes of my *Micro Cookbooks*.

Just as you cannot predict what a car will do by thoroughly studying only its carburetor, the performance of massively parallel inverter-resistor networks will behave differently than the individual circuits studied alone. The magic stuff starts happening when you connect everything to everything else and then let the feedback rip. And therein lies the excitement of some leading-edge research you can do on your own.

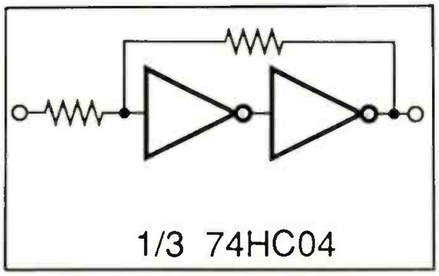

Fig. 6. A CMOS regenerator.

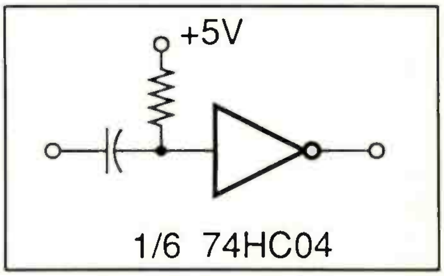

Fig. 7. A CMOS pulser.

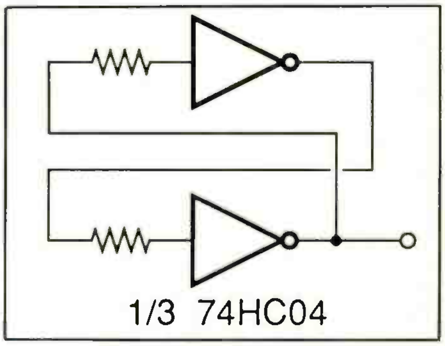

Fig. 8. CMOS memory.

Show me a "neuron" A/D converter

Figure 9 is my adaption of a circuit proposed by John Hopefield of AT&T. It represents a "brand new" way of doing analog to digital, or A/D, conversion. No clocks are needed, no capacitors, nor any precision time or frequency references; yet the circuit needs far fewer inverters than you would use for the usual brute-force or "flash" converter.

Hopefield's theory says that one-way neuron nets work is by seeking minimum energy states. So he set up a feedback network that, through massive feedback, sets up not one but 16 possible "locally minimum" energy states. Each of these states represents one level of a four-bit A/D converter.

The input CMOS inverter acts as an inverting linear amplifier with a gain of somewhat less than one. This unloads, or buffers, the input and changes the outputs to "true" rather than complement values. The gain is less than one so that the inverter still has some drive at both the low and high ends.

The bottom CMOS inverter is our "8" detector. Anything below level eight leaves this inverter low, while anything equal to or above level eight sets this bit high.

The next higher CMOS inverter is our "4" detector. If there were no "synapse" feedback from the 8 output, this detector would go high above level four. But if the

NAMES AND NUMBERS

Computer Journal
190 Sullivan Crossroad
Columbia Falls, MT 59912
(406) 257-9119

EE Times
600 Community Drive
Manhasset, NY 11030
(516) 365-4600

Hewlett Packard
1820 Embarcadero Road
Palo Alto, CA 94303
(415) 857-8000

NCR Electronics
3718 N. Rock Road
Witchita, KS 67226
(800) 325-SCSI

Piezo Electric Products
212 Durham Avenue
Metuchen, NJ 08840
(201) 548-2800

Science Magazine
1333 H Street NW
Washington, DC 20005
(202) 326-6400

Science 86 Magazine
1333 H Street NW
Washington, DC 20005
(202) 326-6731

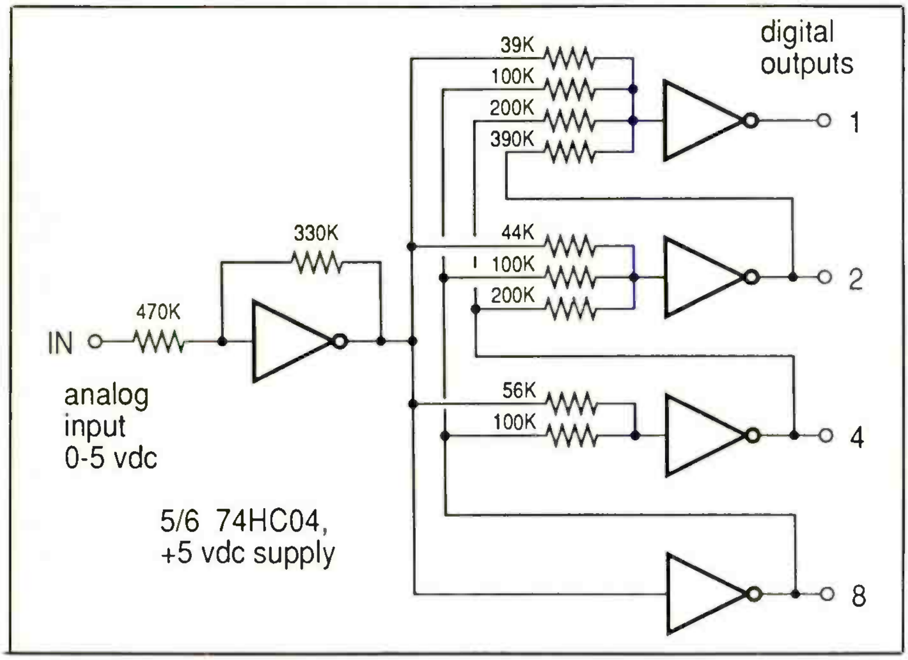

Fig. 9. A novel "neuron-like" A/D converter.

input level is above level 8, then 8 levels worth of current are resistively subtracted from it. Thus, the "4" output will go high for levels 4, 5, 6 and 7. It will seperately go high for levels 12, 13, 14, and 15.

The "2" detector has two feedback paths, one from the "4" and one from the "8" outputs. Follow the bouncing ball, and you will find this output going high for counts 2, 3, 6, 7, 10, 11, 14 and 15.

Finally, the "1" detector has three feedback paths at its input that subtract 2, 4, or 8 levels as needed. The "1" output will be high only on input levels 1, 3, 5, 7, 9, 11, 13, or 15.

What good is this circuit? If in fact, this turns out to be an accurate model of how the brain "thinks," then we are probably looking at an all-time winner. Right now, you have a simple, easily understood, and cheap way of handling a limited-resolution A/D converter with nothing but CMOS inverters and resistors. There's lots of times when you might like to have a simple "volume-control" input to a digital project, for which this circuit is nearly ideal.

You can easily extend the concept to more bits, but you eventually will need lots of very precise value resistors, so today's "heavyweight" A/D conversion schemes have nothing to fear.

How about a new contest?

OK. A free *SAMS* book to the first 10 *Modern Electronics* hardware hackers that come up with some interesting and useful "neuron" circuits that use nothing but CMOS inverters, resistors, and capacitors to simulate something in possibly the same way the human brain does.

As with the earlier contests, the overall winner gets an all-expense paid tinaja quest for two (FOB Thatcher, AZ) and maybe some cash-type money if the idea is good enough for a *Modern Electronics* article. So, take off your thinking cap and show me how to simulate it. Send you entries directly to me.

NEED HELP?

Phone or write your **Hardware Hacker** questions directly to:

Don Lancaster
Synergetics
Box 809
Thatcher, AZ 85552
(602) 428-4073