

I have run across several recent books that are worth a mention. *Who Owns Information?* from Anne Branscomb (Basic Books, 1994) covers everything from Internet to the Dead Sea Scrolls. Focusing on all those radical changes now tearing traditional copyright laws asunder.

While thorough, it is mostly from the viewpoint of a very conservative lawyer. This one would be far more balanced with more input from, say, the *Free Software Foundation*, *Wired Magazine*, or whoever.

For a totally different take on a related topic, read Winn Schwartau's *Information Warfare; Chaos on the Electronic Superhighway* (Thunder's Mouth Press, 1994).

I've long been a Richard Feynman fan. Who was one of my great heros. *No Ordinary Genius: The Illustrated Richard Feynman* has been newly released by Richard Phillips (W. W. Norton, 1994).

A "life and times of" summary that adds to such titles as James Gleick's *Life and Science of Richard Feynman* (Pantheon, 1992).

Or Feynman's own *Surely You're Joking Mr. Feynman* (W.W. Norton, 1985) and *What Do You Care What Other People Think?* (1988).

My other heros? Well, since you asked: Perry Ferrell, Bucky Fuller, Ed Abbey, P.T. Barnum, and William M. Gaines. The last of whom was the greatest philosopher of the twentieth century. Bar none.

Plus, of course, Scrooge McDuck and Gyro Gearloose.

Science Versus Pseudoscience? is an important and easy to read Nathan Aaseng book (F. Watts Press, 1994). One that clarifies several topics we have kicked around here quite a bit. My own definition of pseudoscience is pretty much one of "Looks like a duck – Quacks like a duck".

When you've been at the research game for a while, the pseudoscience stuff gets painfully and laughingly obvious. Such as those dreary chain letters that spend five pages telling you they are not chain letters.

Aaseng is much more specific. In that he lists ten crucial tests to tell the differences between pseudoscience and real science. I've excerpted them here as figure one. The only thing I'd personally append to his list is this: *Science requires that extraordinary claims demand extraordinary proof.*

A second pseudoscience trashing text is *The Hundredth Monkey*, edited by Ken Frazier (Prometheus Books, 1991). Mostly on psychic phenomena and related wonderments.

Magnetic Measurements Handbook is a smallish 95 page self-published book by J. M. Janicke and offered by *Magnetic Research*.

Included are hard-to-find magnetic fundamentals. Along with bunches of construction projects for fluxmeters, Helmholtz coils, and similar stuff.

Magnetic Research also does sell fluxgates and Hall probes.

But note that the Hall devices are *totally* unsuitable for most compass and nav uses. And so is the currently abysmal batch of magnetoresistors. Today, a fluxgate is the *only* way to go. A competing low cost source for fluxgate coils useful for homebrew compass use is *OrthoLogic*.

More on electronic compasses in my *Hardware Hacker* reprints.

Vector-to-step Conversions

Say you want to mill a diagonal cut on a CAD/CAM machine which can only step north-south or east-west. How do you do it? Or you want to show a slanty line on a raster scan display? Or do the same in a laser printer's bitmap? Or wish to map a tumbling 3-D object using rendering animation software?

All of these require *vector-to-step* conversions. One of your most basic and most important needs in all of computer graphics. So, I thought we would review the fundamentals here in figure two.

A *vector* is simply a number that expresses two or more values. As an example, walking ENE for 10.2 miles is a vector quantity.

Vectors can be *resolved* into their underlying components. Ferinstance, going 10.2 miles ENE gets you to the same end point as going 9.4 miles due east and 3.9 miles due north.

Trig can be used to resolve vectors. As an example, ENE is at an angle of 22.5 degrees counterclockwise from east. 10.2 times the *cosine* of 22.5° gets you the 9.4 east component. And 10.2 times the *sine* of 22.5° gets you the 3.9 mile north component.

1. Science is logical and rational.
2. Science makes well-defined claims.
3. Scientific hypotheses are falsifiable.
4. Scientific experiments are repeatable.
5. Science requires that claims be examined by peers.
6. Science views unexplained gaps in theories with suspicion.
7. Science requires caution in examining evidence.
8. Science requires objectivity.
9. Science does not accept coincidence as proof.
10. Science does not accept anecdotal evidence as proof.
11. Extraordinary claims demand extraordinary evidence.

Fig. 1 – NATHAN AASENG lists these ten key criteria in his *Science Versus Pseudoscience* book. I've appended a popular eleventh test here.

Obviously, your 10.2 mile "direct route" is shorter. And probably a lot easier. A CAD/CAM machine that cuts only east and then travels only north would miss by a country mile in the middle of your path. The trick is to find a set of rules that let just

enough north and east steps be taken in just the right order to approximate the path as closely as you can.

A position in space is said to be absolute if it is measured from some 0,0 reference axis point. A position is relative if it gets measured from the

current point, or from your last used or referred position in space.

A 10.2 mile straight line hike is absolute when you end up 10.2 miles from home. It is relative if today's travel on a one week backpack took you 10.2 straight line miles from this morning's start. Typically, absolute positioning is used for the first point in any path, while relative position is used for all points that follow.

Your first step in a vector-to-step conversion often will be to correct your vector. Usually, you'll want to go from where you thought you were to where you desire to be. But you probably are not where you thought you were. Why? Roundoff errors will change your actual current position from your desired current position. If you do nothing about these errors, they will pile up on you.

Which is one reason you take topo maps on a hike.

Instead, you'll correct your input vector so it routes from where you know you now are to where you want to be. This needs a measurement or else a transform from current device space back into user space. The net result is that your input vector will change slightly in size and direction. Thus "swallowing" previous position errors. Correcting on the fly.

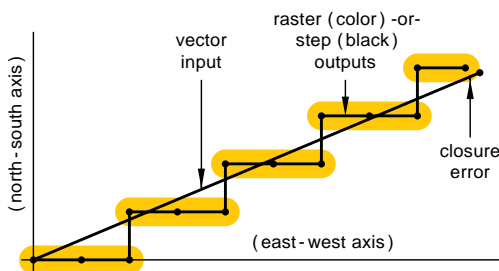
We'll next note that there are eight possible vector-to-step cases. Each of which can end up "alike but different somehow". Let's call east-west our x and north-south our y components. Now, x could be positive or negative. And y could be positive or negative. And the length of x can be greater or less than that of y. Giving us eight total possibilities.

Our next step is to decide which of the eight cases we currently have on hand. Then, we pick the correct case and use it for our actual conversion. If your computer or microcontroller and your available processing time allows you the luxury of "real" trig, you could find the arctan, or "angle whose tangent is", and then integer divide by 45 to get a 0-7 case.

When speed is essential or if your microcontroller isn't too bright, you instead test the signs of x and y plus the absolute lengths of x compared to y. To pick one of the eight cases. The selected case then gets run.

Let's assume the math coordinate

The goal of vector-to-step conversion is to change a diagonal tool path or a slanty line on a screen or printer bitmap into discrete unit steps that travel only in north-south or east-west directions...



First, you correct your input vector so it starts at your actual initial position. This prevents closure errors from piling up. If not already done, you then resolve your input vector into its x and y components. For a vector of length Z and an angle of θ ...

$$x = Z \cos(\theta) \quad \text{and} \quad y = Z \sin(\theta)$$

Next, calculate the slope y/x and save it for later use. Then round x and y off to the nearest integer values to get the actual steps needed. Compare the signs of x and y, and the absolute sizes of x and y to find an octant...

x+	y+	x>y	octant #0	(000 to 045 degrees)
x+	y+	x<y	octant #1	(045 to 090 degrees)
x-	y+	x<y	octant #2	(090 to 135 degrees)
x-	y+	x>y	octant #3	(135 to 180 degrees)
x-	y-	x>y	octant #4	(180 to 215 degrees)
x-	y-	x<y	octant #5	(215 to 270 degrees)
x+	y-	x<y	octant #6	(270 to 315 degrees)
x+	y-	x>y	octant #7	(315 to 360 degrees)

If you are in octant #0, you always step by +x and sometimes step by +y. To determine whether a +y step is needed, multiply the total number of x steps so far by the slope. If the current y total is more than 0.5 steps under this value, also add a new y step.

Here are the rules for the other octants...

- In octant #0, always step by +x and sometimes step by +y.
- In octant #1, always step by +y and sometimes step by +x.
- In octant #2, always step by +y and sometimes step by -x.
- In octant #3, always step by -x and sometimes step by +y.
- In octant #4, always step by -x and sometimes step by -y.
- In octant #5, always step by -y and sometimes step by -x.
- In octant #6, always step by -y and sometimes step by +x.
- In octant #7, always step by +x and sometimes step by -y.

For the maximum possible speed, replace on-the-fly calculations by table lookups, compiling, special hardware, or similar techniques.

Fig. 2 – VECTOR-TO-STEP CONVERSION is one of the most fundamental tools in computer graphics. Here is how do it. Details vary with end use.

orientation, rather than the *geography* one. With east being 0 degrees, north at 90, west at 180, and southeast at 315 degrees. Let's also call any of the eight cases an *octant*. A vector of 10.2 miles ENE is in octant zero.

Now, in octant zero, x absolutely *must* be positive and absolutely *must* be larger than y . And y also *must* be positive. Otherwise, we'd be in some *other* octant. So, in octant zero, we simply use the rule "always step by x and *maybe* step by y ".

Let's assume your corrected vector has been resolved into $x = 9.4$ and $y = 3.9$ components. Since we are only allowed whole *integer* steps, we are not allowed to take 9.4 unit steps. In this case, we will *round down* to nine steps. Thus, we know we want nine x steps. And we want four y steps. The question is *when* to add the y steps to get *closest* to our original vector.

Should the rounding down give you an x value of zero, you exit. The little missed piece gets picked up on the next vector. Exiting on a zero length also gets rid of the possible divide-by-zero in the next step.

Next, you calculate the *slope* or the "slantocity" by dividing the little y number by the bigger x one. Or, in this case y/x . Save this value.

You then step along your x axis, *always* taking one x step. After each x step, you will *compare* your actual y against your desired y position. The desired y position gets obtained by multiplying x times the slope. Any time you end up more than half a step (or 0.5) low, you throw in a y step. Continue this for each needed x step.

Reviewing: For any vector-to-step conversion, you'll first correct your vector so it travels from where you really are to where you want to be. You then pick one of eight octants, based on the size and sign of the x and y components. You'll then pick octant specific code that *always* steps along x but only *sometimes* will step along y . Or vice versa.

Getting Fancy

There are several refinements we can add here. Testing for an 0.5 unit error gets less accurate the slantier you get. But 0.5 is more than good enough for most uses. For the really best fit, you should instead use 0.5 times the *cosine* of the slant angle.

If you can't "afford" a real cosine, $1 - (\text{angle in radians})^2/2$ could fake it pretty good. Cruder useful fakes are $0.5 - (\text{angle in degrees})/300$. Or even letting 0.44 split the difference.

Some CAD/CAM applications will let you move your x and y steppers simultaneously. This is called *stepper phase interleaving*. And means you can travel *east* or *northeast* in octant zero. What you have done is *combine* an east step with a north step into a single northeast one. Eliminating a sharp corner in the process.

Which dramatically improves your resolution, speed, and smoothness of your actual cuts. Figure three shows you the improved results.

But why calculate the obvious over and over again? The foremost rule of all high speed computing is to *never calculate what you can look up!* Any of several *table lookup* techniques can dramatically speed things up for you. But these are hard to understand and are very device specific.

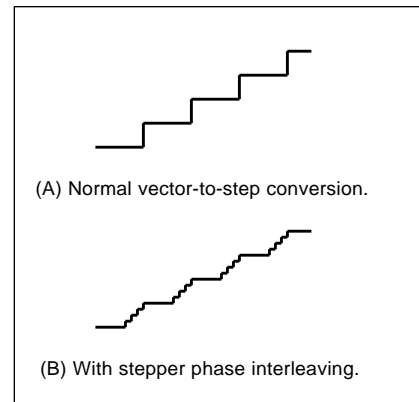


Fig. 3 – CAD/CAM SYSTEMS could apply an -x- and -y- stepper phase interleaving to dramatically improve their apparent resolution.

A PostScript Example

There are all sorts of variations on these vector-to-step schemes. It all depends on the speed required.

In some cases (such as computer animation), your rendering speed is

```
% PostScript Vector-to-step student demo algorithms.

% Copyright c 1994 by Don Lancaster and Synergetics Box 809 Thatcher, AZ 85552
% (602) 428-4073. All commercial rights and all media rights fully reserved.
% Additional support via GENIE PSRT. Consulting services also available.
% Personal use permitted so long as this header remains present and intact.
% Two-way host recordable comm is required.

/curxpos 0 def /curypos 0 def % running output currentpoints

/oct0 {2 copy exch dup 0 eq (0.00001 add) if div /slope exch store pop cvi /#incs exch store 0
0 1 #incs 1 sub (/posn exch def (E) print /curxpos curxpos 1 add store posn 1 add slope mul
0.5 sub 1 index ge {1 add (N) print /curypos curypos 1 add store} if) for pop} def

/oct1 {exch 2 copy exch div /slope exch store pop cvi /#incs exch store 0 0 1 #incs 1 sub
{/posn exch def (N) print /curypos curypos 1 add store posn 1 add slope mul 0.5 sub 1 index
ge {1 add (E) print /curxpos curxpos 1 add store} if) for pop} def

/oct2 {exch 2 copy exch div neg /slope exch store pop cvi /#incs exch store 0 0 1 #incs 1 sub
{/posn exch def (N) print /curypos curypos 1 add store posn 1 add slope mul 0.5 sub 1 index
ge {1 add (W) print /curxpos curxpos 1 sub store} if) for pop} def

/oct3 {2 copy exch div neg /slope exch store pop cvi /#incs exch store 0 0 1 #incs 1 sub
{/posn exch def (W) print /curxpos curxpos 1 sub store posn 1 add slope mul 0.5 sub 1 index
ge {1 add (N) print /curypos curypos 1 add store} if) for pop} def

/oct4 {2 copy exch div /slope exch store pop cvi abs /#incs exch store 0 0 1 #incs 1 sub
{/posn exch def (W) print /curxpos curxpos 1 sub store posn 1 add slope mul 0.5 sub 1 index
ge {1 add (S) print /curypos curypos 1 sub store} if) for pop} def

/oct5 {2 copy div /slope exch store cvi abs /#incs exch store pop 0 0 1 #incs 1 sub (/posn
exch def (S) print /curypos curypos 1 sub store posn 1 add slope mul 0.5 sub 1 index ge {1
add (W) print /curxpos curxpos 1 sub store} if) for pop} def

/oct6 {2 copy div neg /slope exch store cvi abs /#incs exch store pop 0 0 1 #incs 1 sub (/posn
exch def (E) print /curxpos curxpos 1 sub store posn 1 add slope mul 0.5 sub 1 index ge {1
add (E) print /curxpos curxpos 1 add store} if) for pop} def

/oct7 {2 copy exch div neg /slope exch store pop cvi abs /#incs exch store 0 0 1 #incs 1 sub
{/posn exch def (W) print /curxpos curxpos 1 add store posn 1 add slope mul 0.5 sub 1 index
ge {1 add (S) print /curypos curypos 1 sub store} if) for pop} def

/findoctant {2 copy exch 2 copy 0 eq exch 0 eq or {pop 0.000001} if atan cvi 45 idiv
[{oct0}{oct1}{oct2}{oct3}{oct4}{oct5}{oct6}{oct7}{oct0}] exch get} def

/vectortostep {findoactant exec flush } def % main control loop

% //// demo - remove or alter before reuse ////
9.4 3.9 vectortostep quit % should return EENEENEENEENE
```

Fig. 4 – A STEP-TO-VECTOR CONVERTER written in PostScript.

PROGRAMMABLE INTERCONNECT RESOURCES

Aptix
2890 North 1st St
San Jose CA 95134
(408) 428-6200

Atesla
2890 North First Street
San Jose CA 95134
(408) 428-6250

EDN Magazine
275 Washington St
Newton MA 02158
(617) 964-3030

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

Electronic Component News
1 Chilton Way
Radnor PA 19089
(215) 964-4345

Electronic Design
611 Rt #46 West
Hasbrouck Heights NJ 07604
(201) 393-6060

Electronic Products
645 Stewart Ave
Garden City NY 11530
(516) 227-1300

I-Cube
2328-C Walsh Avenue
Santa Clara CA 95051
(408) 986-1077

Lattice Semiconductor
5555 NE Moore Court
Hillsboro OR 97124
(503) 681-0118

Pioneer Semiconductor
2343 Bering Drive
San Jose CA 95131
(408) 435-0800

Quality Semiconductor
851 Martin Ave
Santa Clara CA 95050
(408) 450-8063

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

everything and you have to use every machine language trick you possibly can to make things happen real quick like. Elsewhere (such as CAD/CAM), a more leisurely pace is acceptable, and a powerful and friendly higher level language works just fine.

I have found the general purpose *PostScript* computing language to be ideal for medium speed conversions. Especially for real time CAD/CAM.

Figure four shows you *PostScript* code that can do these conversions for you. This is a very simple tutorial example that can be (and has been) significantly improved.

With table lookup code, speeds in the 13,000 steps per second range are easily hit. Your results can also get recorded and then be sent out later at arbitrarily high speeds.

Most CAD/CAM systems are cutter head limited to much slower speeds than these. Thus *PostScript* speed is in *no way* a limiting factor.

The key point is that you can now easily apply genuine *PostScript* for ultra low cost homebrew hacker apps. Superbly better than any other way I know of. Particularly for such tricks as microsizing and compensating for tool paths and diameters.

Just how can you use *PostScript*'s

graceful curves and fancy lettering? Curves are simply broken down into lots of individual straight line vectors which are short enough to meet your accuracy goals. *PostScript*'s *charpath* and *flattenpath* operators get used for curve-to-vector conversions.

More examples of vector-to-step conversions are in my [POSTVECT.PS](#), [FLUTWUMP.PS](#) and [FLUTOOLS.PS](#) on www.tinaja.com I've also got some exciting new products in the works that use these concepts. Give me a call if you need advanced details or help with your own project.

The Halogen Cycle

I've gotten several helpline calls over early fusion lamp failures on Canon EX laser engines. Such as the 4M+ or Pro630. The symptom is a darkening of the bulb before it blows. Well, I got some theories here...

A quartz halogen lamp sits inside that bottom roller. And intermittently turns on and off. The light heats the roller, maintaining a regulated 375 F surface temperature.

Despite the fancy name, a quartz halogen lamp is still just a plain old incandescent bulb. It still has a more or less ordinary tungsten filament. The only difference is it gets run

**new from
DON LANCASTER**

ACTIVE FILTER COOKBOOK

The sixteenth (!) printing of Don's bible on analog op-amp lowpass, bandpass, and highpass active filters. De-mystified instant designs. **\$28.50**

CMOS AND TTL COOKBOOKS

Millions of copies in print worldwide. THE two books for digital integrated circuit fundamentals. About as hands-on as you can get. **\$28.50** each.

**INCREDIBLE SECRET
MONEY MACHINE II**

Updated 2nd edition of Don's classic on setting up your own technical or craft venture. **\$18.50**

LANCASTER CLASSICS LIBRARY

Don's best early stuff at a bargain price. Includes the CMOS Cookbook, The TTL Cookbook, Active Filter Cookbook, PostScript video, Case Against Patents, Incredible Secret Money Machine II, and Hardware Hacker II reprints. **\$119.50**

LOTS OF OTHER GOODIES

- Ask the Guru I or II or III \$24.50
- Hardware Hacker II, III or IV \$24.50
- Micro Cookbook I \$19.50
- PostScript Beginner Stuff \$29.50
- PostScript Show and Tell \$29.50
- Intro to PostScript Video \$29.50
- PostScript Reference II \$34.50
- PostScript Tutorial/Cookbook \$22.50
- PostScript by Example \$32.50
- Understanding PS Programming \$29.50
- PostScript: A Visual Approach \$22.50
- PostScript Program Design \$24.50
- Thinking in PostScript \$22.50
- LaserWriter Reference \$19.50
- Type 1 Font Format \$16.50
- Acrobat Reference \$24.50
- Whole works (all PostScript) \$380.00
- Synergetics Surplus Catalog FREE
- Technical Insider Secrets FREE

POSTSCRIPT SECRETS

A Book/Disk combination crammed full of free fonts, insider resources, utilities, publications, workarounds, fontgrabbing, more. For most any *PostScript* printer. Mac or PC format. **\$29.50**

BOOK-ON-DEMAND PUB KIT

Ongoing details on Book-on-demand publishing, a new method of producing books only when and as ordered. Reprints, sources, samples. **\$39.50**

THE CASE AGAINST PATENTS

For most individuals, patents are virtually certain to result in a net loss of sanity, energy, time, and money. This reprint set shows you Don's tested and proven real-world alternatives. **28.50**

BLATANT OPPORTUNIST I

The reprints from all Don's Midnight Engineering columns. Includes a broad range of real world, proven coverage on small scale technical startup ventures. Stuff you can use right now. **\$24.50**

RESOURCE BIN I

A complete collection of all Don's Nuts & Volts columns to date, including a new index and his master names and numbers list. **\$24.50**

FREE SAMPLES

Check Don's Guru's Lair at <http://www.tinaja.com> for interactive catalogs and online samples of Don's unique products. Searchable reprints and reference resources, too. Tech help, hot links to cool sites, consultants. email: don@tinaja.com

FREE US VOICE HELPLINE VISA/MC

SYNERGETICS
Box 809-NV
Thatcher, AZ 85552
(520) 428-4073

much *hotter* for more efficiency. The quartz bulb is more resistant to heat shock than ordinary glass. And some wondrously magic chemistry known as the *halogen cycle* is supposed to make the filament last longer.

Tungsten readily forms compounds with any of the halogen gases. Such as tungsten hexachloride or tungsten hexafluoride. These gases are stable at lower temperatures, but dissociate at very high temperatures.

Now, one big problem with high temperature filament life is that the tungsten literally boils off. This both weakens the filament and blackens the glass, leading to early failure.

Here is how that halogen cycle works: Near the hot filament, ejected tungsten mixes with the dissociated halogen. Away from the hot filament, the two merge to form a stable halide gas. And when the lamp gets turned off, much of the dissociated tungsten deposits *right back* onto the filament! Normally, there is little chance for bulb darkening.

However, there are two important rules for quartz halide lamps: *Never touch them!* and *Always cool them completely!*

Any fingerprints can cause a high temperature spot on the glass, leading to mechanical stress and a possible darkening. And the cooling-off time is *mandatory* for that halogen cycle to complete itself.

So, there might be a supplier or quality control problem. Or maybe somebody who doesn't know better is handling the lamps during assembly or service. But I have the following hunch: That "green" lower energy operation of new laser printers seems *dead wrong* for the halogen cycle!

The *worst* thing you can do to a quartz halogen lamp is to repeatedly and briefly cycle it!

Possible solutions: Run the lamp continuously for an hour every now and then. Maybe even at 120 percent of voltage. See if it lightens up any. Outside of the printer, of course. And in an area where a kilowatt of heat can be safely disposed of. And, of course, *keep fingerprints away!*

Or design a new lamp technology *specifically* for the new green energy cycles. Instead of using a plain old stock copier bulb. Or go to some type of PTC self-regulating beastie.

NAMES AND NUMBERS

Access to Energy

Box 1250
Cave Junction OR 97523
(503) 592-4142

Accu-Weather

619 W College Avenue
State College PA 16801
(814) 234-9601

Basic Books

10 East 53rd Street
New York NY 10022
(212) 207-7057

Bergquist

5300 Edina Industrial Blvd
Edina MN 55439
(612) 835-2322

Design Toscano

17 East Campbell Street
Arlington Heights IL 60005
(800) 525-0733

EITD

2057-2 Aurora Road
Twinsburg OH 44087
(216) 425-9000

Extraordinary Science

PO Box 5636
Colorado Springs CO 80931
(719) 475-0918

Lindsay Publications

PO Box 538
Bradley IL 60915
(815) 935-5353

Magnetic Research

122 Bellevue Avenue
Butler NJ 07405
(201) 838-6348

Motorola

5005 E McDowell Rd
Phoenix AZ 85008
(800) 521-6274

MultiFAX

Route 1 Box 27
Peachland NC 28133
(704) 272-9028

W. W. Norton

500 Fifth Avenue 6th Fl
New York NY 10110
(212) 354-5500

OrthoLogic Corp

2850 S 36th St #16
Phoenix AZ 85034
(602) 437-5520

Prometheus Books

700 E Amherst Street
Buffalo NY 14215
(716) 837-2475

Seismo-Watch Newsletter

PO Box 18012
Reno NV 89511
(800) 852-2960

Stratagene

11011 N Torrey Pines Rd
La Jolla CA 92037
(800) 424-5444

Tektronix

PO Box 500
Beaverton OR 97077
(800) 426-2200

Thunder's Mouth Press

632 Broadway 7th Fl
New York NY 10012
(212) 780-0380

F. Watts Books

95 Madison Avenue
New York NY 10016
(212) 951-2650

ZFX

40 Washington Street
Westborough MA 01581
(800) 876-1232

Programmable Interconnect Resources

The next logical feature beyond programmable integrated circuits is fully *programmable systems*. Where all of the wires *between* chips can be conveniently and quickly rearranged. Obvious uses are for breadboarding, development, and field upgrades.

An ideal interconnect would be dense and cheap, allowing arbitrary connection of anything to anywhere. Connections would permit analog or digital signals to move out in either direction without delays.

Both the "on" resistance and any crosstalk would end up very low. And these devices should consume

zero to little power. They would have to remember all their connections between power downs *and* be fully reprogrammable at any time.

As you might imagine, the semi houses are scrambling to offer new programmable interconnect chips. I have tried to list the key players in this month's resource sidebar.

Unfortunately, nearly all devices offered today miss and miss badly on one or more goals. *Aptix* seems to be at the top of the heap. Their devices are quite dense, extremely flexible, bidirectional analog/digital, and have low on resistance and crosstalk. But they are expensive and have to be reprogrammed each time your power is applied to the system.

Pioneer and *Quality* offer very low cost and easy to use bidirectional bus switches. But their density is very low. Programming is external. *Atesla* has arrays of fuse-blowable direct interconnects. Medium low cost and bidirectional analog/digital. But once blown, the results are permanent and cannot be changed.

Apparently, those programmable logic device folks are relabeling their existing PLA and PLD variations to claim they are interconnects. While dense, reprogrammable, and cheaply priced, the results are often digital

only and one way only. There's also prop delays involved.

One brand new source is *Lattice Semiconductor* with their *Generic Digital Switch* shown in figure five. Densities are offered up to 11 by 11. Any lead can be taught to be an input or a tri-state digital output that can be its input, the complement of its input, a one or a zero. Reprogrammable.

But any A-side input/output can only be routed to B-side pins. These are ideal as dipswitch replacements.

Meanwhile, *I-Cube* is offering what they call FPID devices, short for

NEED HELP?

Phone or write all your US Tech Musings questions to:

Don Lancaster
Synergetics
Box 809-EN
Thatcher, AZ, 85552
(520) 428-4073

US email: don@tinaja.com
Web page: www.tinaja.com

Field Programmable Interconnect Devices. While mostly a PLD style beastie, these *do* also provide for high impedance analog.

Crossbar architecture is used with densities as high as 320 by 320. Since this is static RAM based, it must be reprogrammed every time your power is disconnected. Each pin can be taught to have *eight* different functions. It can be non-buffered for analog signal pass-through. Digital inputs can be buffered or latched. Outputs can be buffered, registered, or held either high or low.

Info on newer devices which are likely to overcome all of the obvious defects of this current crop should show up in the usual places, such as *E.E. Times*, *EDN*, *Electronic Design*, *Electronic Component News*, and in *Electronic Products*.

Programmable logic devices could even be used to downgrade double sided circuit board layouts into single sided ones.

Let's have your thoughts on new hacker uses for these exciting parts. The usual *Incredible Secret Money Machine II* book rewards for your contributions.

New Tech Lit

From *Motorola*, their new *Master Selection Guide*. One unique chip of theirs is the new MC144143 tv closed caption decoder.

Tektronix has a bunch of videos and ap notes that you just may be interested in. One video is on *The VectorScope*, a crucial tool for color tv broadcasting and video production. There's also a pair of vids on *Cable Measurement*, plus others on such specialized topics as *Cable TV Sweep Systems*. Of their many ap notes, you might find their *Spectrum Analyzer*

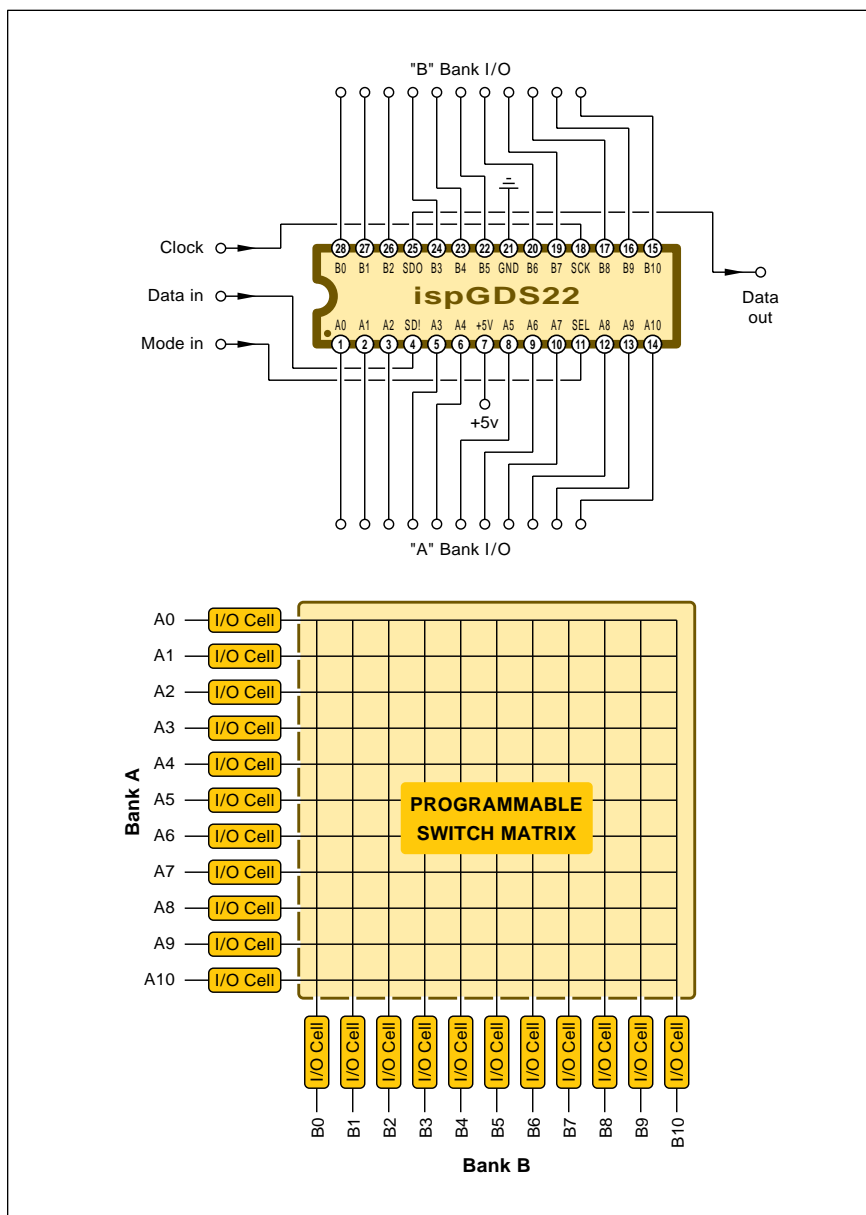


Fig. 5 – GENERIC DIGITAL SWITCH from Lattice Semiconductor can offer medium density programmable interconnects.

Fundamentals to be handy.

Real live human neurons are now offered by *Stratagene* per their inside cover ad in the September 2, 1994 issue of *Science*. If I understand this correctly, these can be hot wired for direct electrical to biological neuron computing experiments. But do not connect these to your cat!

Three weather satellite resources: *MultiFAX*, *Accu-Weather*, and *ZFX*.

From *Harris Publishing*, their new *1995 Electronic Industry Telephone Directory*. If it is not already in my *Synergetics Master List*, the EITD is where I'll always go first.

Access to Energy is a labor-of-love venture that bills itself as a high tech,

free enterprize monthly newsletter.

A freebie *Sil-Pad Sample Kit* from *Bergquist* includes several thermally conductive insulators that are useful for power heatsinking.

A free issue of the *Seismo Watch Newsletter* is available. This covers weekly earthquake info.

The *Extraordinary Science 1995 Resource Guide* is a new catalog of both real science and pseudoscience books, videos, and reprints. Among other strengths, they now stock a definitive Tesla collection. They also now seem to be stocking a lot of the *Lindsay Publications* texts.

A well done collection of Griffins and Gargoyles are newly available

from *Design Toscano*.

For the fundamentals of digital integrated circuits, check into my classic *TTL Cookbook* and my *CMOS Cookbook* texts. More details in my nearby *Synergetics* ad.

Also a reminder that reprints and preprints of all my columns, my free insider secrets catalogs, and instant tech help is on www.tinaja.com

Free catalog requests and technical questions you want answered here or on my web site can also be sent me via my don@tinaja.com email.

As usual, most of these resources I've mentioned appear in the *Names & Numbers* or in the *Programmable Interconnect Resources* sidebars. ♦