

US\$3.00

SORD WORLD

February
1984

No. 3

PRODUCT REVIEW!
SORD'S M68: Two Computers in One

BIOGRAPHY!
The Revolutionary Mr. Takayoshi Shiina

PIPS APPLICATIONS!
An Automatic Income Statement

SORD WORLD PIPS SCHOOL!
PIPS For Beginners

GUEST COLUMN!
PIPS: A Programmer's Conversion

TABLE OF CONTENTS

WELCOME TO SORD!	1
GUEST COLUMN:	
PIPS: A PROGRAMMER'S CONVERSION	2
PIPS APPLICATION:	
AN AUTOMATIC YEAR-TO-DATE INCOME STATEMENT	4
PRODUCT REVIEW:	
SORD'S M68: PERFORMANCE PAR EXCELLENCE	10
BIOGRAPHY:	
TAKAYOSHI SHIINA, JAPAN'S COMPUTER REVOLUTIONARY	16
SORD WORLD PIPS SCHOOL:	
A PIPS PRIMER	20
SORD'S COMMUNICATIONS PRODUCT LINE	25

PUBLISHER: YASUO TERAZAWA
 EDITOR: MICHAEL UEHARA
 NEW PRODUCTS
 EDITOR: JOHN RAMSAY
 PHOTOGRAPHER: KOJI YOKO
 PROGRAMMER: CLAY ELEMANS

SORD WORLD is a publication of PIPS
 WORLD, INC., Tokyo, Japan. The entire
 contents copyright © by PIPS WORLD,
 INC. All rights reserved. For those in-
 terested in the Japanese publication of
 this magazine, please contact:

PIPS WORLD, INC.
 Naigai Chizu Bldg., 3-22, Kanda
 Ogawa-machi Chiyoda-ku, Tokyo 101
 JAPAN

WELCOME TO SORD!

Welcome to another information-packed edition of SORD WORLD. First of all we have an article by journalist Roy Garner on SORD president, Takayoshi Shiina, the man behind the extraordinary rise of the SORD Computer Corporation from a company with a paid-up capital of less than \$3,000 in 1970 to Japan's fastest growing company of all in 1983. SORD's corporate goal of producing computers which are "user-friendly," reliable, fast, and easily upgradable is something which reflects Mr. Shiina's distinctive personal philosophy. Roy, who has interviewed Mr. Shiina, gives some insights into the unusual mind behind the SORD miracle.

Tom Morrison is a computer journalist and reviewer from California who has been looking at the computer scene in Tokyo recently. Tom has been particularly impressed with the M68, SORD's newest desktop computer with the unique dual CPU system incorporating both an 8-bit and a 16-bit processor. Tom has had a good look at the M68, and has talked with a number of other serious users about the M68's capabilities. His findings on the M68's power, flexibility, and software compatibility will be of interest to all of you who might be facing up to the problems of expanding your system but maintaining compatibility with your old software.

"What surprises me is that this machine has two CPU's...It's hard to believe that a computer can let you use your old software while allowing for updates," Tom said. "This is one machine that will make 8-bit users sit up and look at 16-bit applications a little more seriously," he adds. If you're stuck at the 8-bit stage but need the capabilities of 16-bit power, read Tom's article and see how easily you can have what you need.

Accountant Art Norris provides us with a report from the trenches on the office politics front. Art's top management frequently bombards his accounting section with impossible requests for obscure information which of course they wanted yesterday. (Heard that one before?) But surprisingly enough, Art does not suffer from nerves, ulcers, or working weekends. And the reason is the amazing flexibility of PIPS.

"I'm usually someone who gets pretty worked up over happenings in the office. After seven years of trying to figure out the capricious whims of my co-workers, I finally decided to tackle some of the problems on my own. I spell

relief P-I-P-S."

Using PIPS, Art has developed a method of extracting annual or year to date income statements from quarterly figures at the push of a SORD M68 button. Art's article takes you through the steps involved in this PIPS application.

Guest columnist this issue is Elsa Maybell, a programmer from Baltimore, Maryland, who discovered more than Japan on her recent business trip there.

"I was skeptical about information packages. Most spreadsheets give you a lot of what you don't want and very little of what you do. But PIPS is an exception—the BASIC statements it provides and some of the time saving SORT routines are something people here have wanted for a long time," Elsa told us.

For an insider's view of SORD's activities, we asked Koji Ohta of SORD's Customer Services Division Tokyo Office, to comment on SORD's communications product line. These include the "electronic mailbox" capability of the SOCS wordprocessor, the PCOMM program developed in Australia for transmitting PIPS pages, and the latest improvements to SORD's S-NET networking system.

And this issue we proudly open—THE SORD WORLD PIPS SCHOOL!! "SORD WORLD runs plenty of sophisticated articles on PIPS applications which assume that you already know a lot about PIPS," complained Hiroshi Nemoto, one of SORD's PIPS instructors, "but what about the beginners? I think that one of PIPS' best features is the way in which total beginners can get results after only a few hours training." So we put Hiroshi on the spot straight away and the result is the first in a series of articles on PIPS aimed straight at the novice.

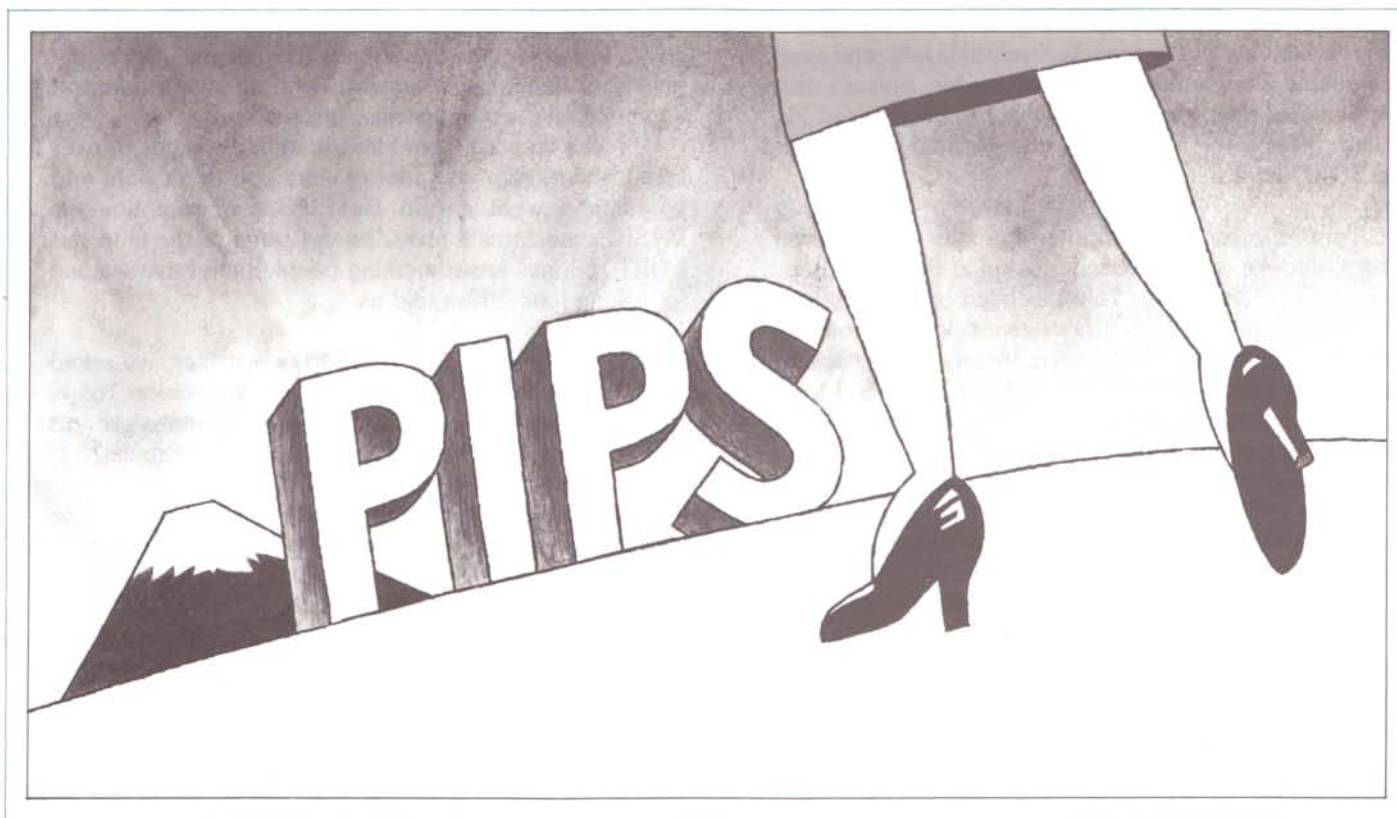
As ever, SORD is working to incorporate the latest developments in computer technology into its great line of products, and so keep you, the user, at the top of the market. SORD WORLD will continue to keep you up with the latest at SORD.

Michael Uehara
Editor

GUEST COLUMN

PIPS: A PROGRAMMER'S CONVERSION

by ELSA MAYBELL



Before last year you couldn't have gotten me to look at PIPS for love or money. I wasn't prejudiced: you couldn't have gotten me to look at any other recent information management packages, either. But it wasn't because I didn't need them. Let me explain.

I'm a professional programmer myself, and I have a bad habit: I find fault with other people's solutions. Even when they work, when they're ready at hand, and when they will save me time and money. Why? Because even if I don't yet know a better way, I'm convinced that there is

one. And every programmer worth his salt is like that.

Back in Baltimore, where I come from, I work in a software house where I supervise the point-of-purchase system section. "Section" is one word for it; "zoo" is another. I know a lot of middle—and lower-middle, and even sub-lower-middle—managers in other lines of work. They're effective and they know their stuff, but they refuse to manage programmers. That's because only programmers can manage programmers. If they're not too busy, of course.

To manage programmers is to manage ten, twenty or a hundred people, each one convinced they've got a unique shortcut that will solve whatever problem is under discussion. The only way to keep the whip hand is to be able to think of something better, fast, and on so many separate occasions that they respect you. It's a battle that you can be ahead in, but that you can never fully win. So I manage twenty inventory-control programmers, most of them occupied in custom adaptations of our standard packages for the obscure or eccentric private purposes of this or that individual customer. I handle them by having some authority—which means being right at least once a day,—a lot of cigarettes, and—I may as well admit it!—a significant weight advantage.

Last year my boss had a new idea. I was to spend less time with people, i.e. less time with a whip and chair, and more time on development. He thought I had it in me to come up with the basic inventory-control modules that would be the heart of our standard packages in the next few years. And he was right about that—but he was wrong about the place the ideas come from. I've never had better ideas than in trying to squelch the latest code-consuming project of one of personnel's whiz-kid acquisitions—not to mention the searing shortcuts and GOTO lover's leaps of my more experienced, hence more dangerous associates.

Still, I faced the electronic broom closet he had prepared for me ("where I could get some peace and quiet") with equanimity. Experimenting in BASIC with only BASIC talking back to you is a more sterile way to go than my boss could understand, but so what? Tedious though it was, I found it an easy—maybe too easy—day's work. BASIC is to me what a piano or a guitar is to other people: an instrument to play on, to try things out, to fumble with, experiment with, until I run across what I'm looking for—and then rapidly carry the idea to its logical conclusion.

So months went by, and then there was more news: our company had made a tieup deal with a sister software outfit in Japan. And they wanted me to visit for six months and contribute my experience about the American wrinkles of inventory control! I was excited about going to Japan; in fact, it was a place I'd always wanted to visit, ever since I was a kid. But I was excited, not about the work opportunity, but about seeing a country that was so different from anything in my own previous experience. And it certainly never occurred to me how much Japan would teach me beyond its traditional aspects. Because it was in Japan that my attitude to what I'd always contemptuously called "spreadsheet programs" was turned on its head.

Japan itself—the people, the culture, the tradition, and—for me, especially—the food, bowled me over. It was all I'd hoped for and more. But in our sister Japanese firm I was in for a big shock. At convenient points on the

research and development floor, there were SORD micro-computers—M23s and some brand new M68s—and on them, always loaded, always ready, was PIPS. What I called a "spreadsheet" program.

"What good is this toy to you?" I asked them. "You're programmers yourselves!" And at the time, I didn't believe their explanations. How PIPS was so flexible it could even have a role in the first stages of developing sophisticated standard packages. How initial concepts, the rough results of late-night brainstorming, could be set up experimentally on PIPS far faster than if the full BASIC code was written from scratch. How they'd rather do without a blackboard or a pencil and scratchpad than do without PIPS—because PIPS was the first test they could make to see whether a broad new architectural idea was worth going into more deeply or was just another flash in the brainpan.

Back home I would never have dreamed of keeping an American "spreadsheet" on hand to doodle with. What was the point? I knew those things had limitations: it was those limitations that provided a solid basis for custom packaging and our own business.

But then my new colleagues showed me: PIPS is an open piece of software—wide open to BASIC expressions. PIPS had an arsenal of BASIC available for me to use, while PIPS commands do the data setup, the utilities and the housekeeping. In fact, all the tedium of the job. I was left to think—and sketch out the new possibilities that I had been groping for.

PIPS gave me LET statements, with variable assignments employing eleven logical functions. I could use REFR statements to assign values to variables internally without stating them in the autoprogram. And REFW gave me the indexed update capability I was used to. GOTOs and GOSUBs worked just as they do in regular BASIC. I could nest up to three subroutines—and believe me, deeper nesting than that gets messy. FOR...TO...NEXT also was just like home; so were IF/THEN statements. I could keep data items in a separate file and request their retrieval in the autoprogram with a single character; another character would let me manually modify my variables in the course of running the program. And DISP and PRINT were available too, to leave notes to myself wherever I chose to interrupt the PIPS process.

That did it. Now I'm converted to PIPS. Back home in the U.S.A., I twisted my own taskmaster until he made a minor hardware and software investment. Now I've got my in-house system terminal in one corner of my office—and PIPS in the other, always ready, unblinking, waiting for me to lose patience with jamming in the code. I can always swivel my chair—and lose myself in the flight of ideas. And with my productivity up, a lot of those ideas are again coming from the new crop of wiseacres implanted all around me while I was away. May as well face it: without them, what would I really have to think about?

AN AUTOMATIC YEAR-TO-DATE INCOME STATEMENT

by Art Norris

PIPS is not an accounting system, and I am an accountant. But when my other accounting friends are enthusiastically talking software—what they expect from it, or what they're already getting from it, in diminishing the sometimes incredible workload of the accounting department of any high-tech manufacturing firm, they will ask me what piece of software I have to recommend, and I always answer "PIPS." I say that not only for what PIPS can provide in the way of accounting assistance: I've been even more moved by what PIPS offers accountants in the way of relief from top management's sometimes inspired, sometimes harebrained requests in the way of information.

From the point of view of top management in our company—as well as from my own worm's-eye view—PIPS is a miracle for projection, simulation and forecasting. Not only that, it makes them feel they're doing something. And that's great. It means one less thing that I have to do.

Don't get me wrong. I'm a computer believer too. And I've been very impressed at times by the quality of the long-range planning that can emerge from a truly capable executive's brainstorming session with a computerized information management system. But I'm even more impressed by something else. How PIPS can liberate a harried accounting department from the rain of constant demands for information that either long-range planning or half-baked, highly political numbers cookery can require.

In an accounting department, there's continuous tension between the demands from above, from other departments on the same level, from employees with their personal problems—the clarification of a payroll stub, a complaint about the performance of the health insurance vendor—and from the department's own indispensable duties, the things we know must get done. Payroll can't wait, this week or ever. Neither can reports to Federal agencies. And there's the state. And there's the unions, who need more info to square their own health insurance plans for their members.

Into this morass of labor comes a directive from the highest levels. Get out material costs for 1978. What did copper cost then, compared to this year? The head of marketing read in the Wall Street Journal that copper is coming down again. Would costs of materials with copper

components look like those of 1978? And what wrinkles might that create in this year's coming balance sheet? Of course, we aren't supposed to answer that last question. But we have to assemble all the information that will allow our superiors, in their wisdom, to settle the question in case the promised decline actually takes place.

Well, on the Wednesday before the Friday that everyone is waiting for there's no time, really, to offer our support on these recondite matters. But occasionally we have to. With the company's work force standing around, licking their lips. It's hard for the top to understand that there's a crucial difference between a check that's in a person's hand at 11:30 a.m. and one that isn't delivered until 4:20 in the afternoon. But we hear about it—plenty.

Yet I wouldn't blame Hank. Hank is our president, and especially compared to some I've seen elsewhere, I'd rate him a good decision-maker. He bats about .333—that is, every third time he's right. And it's remarkable what good results that will give you in business. His approach is to try to be aware of all the possibilities. Ruling some of them out tends to be someone else's job—but it's a job he allows them to do.

Nor would I blame Martin, our financial controller. Martin is occasionally slow, but absolutely dependable. Curiously, he's the one who seems to manage most often to head Hank off at the pass when it's necessary. Perhaps it's because he is slow, and lets Hank explore whatever new territory he's interested in just enough to help Hank persuade himself out of it, using the information that Hank himself has collected.

And Bernie—well, I would blame Bernie, but he's my direct supervisor, the head of the accounting department. Bernie has trouble telling Hank or Martin no, when Hank has a demand to make out of his latest enthusiasm, or Martin has a demand to make out his concern over Hank's latest enthusiasm.

One day Bernie came to me and said "Art! What am I going to do? Look at this." A memo floated to my desk-top from his nerveless fingers. I won't trouble you with the details. Suffice it to say that it contained a request for lots

of figures, some of which were to be old and some of which were to be imaginary.

"You know, I can't take it any more," Bernie said, looking not at me but deep into the grey finish of a filing cabinet to my right. "Maybe I'll put you in charge of this."

"Bernie, have a seat," I told him. "We're going to talk. The time has come for drastic action. I'm talking about capital investment."

His eyes shifted abruptly from the filing cabinet. "What? What do you mean?" he blurted.

I broke it to him cold. "Bernie, we need to buy upstairs a microcomputer and get them started. There's no other way. They've got too many questions."

"But Art, that means shopping! The microcomputer market is a maze. Which of us is going to put in the six months of research it takes to not get taken?"

"No shopping, Bernie. I'm calling SORD—now. I'll write you out a chit to sign before I go home." He was staggered—he hates to see money go out, he's a good man—but in his eyes was a flicker of desperate hope.

As soon as Bernie was gone, I placed the call. In a few well-chosen words I briefed them on the situation, which they had seen themselves many times before—executive thinking on the loose, dressed to kill with no place to go. What they needed was an M68—no, three M68s—and above all, PIPS to provide them with a little intellectual discipline. I knew PIPS myself from watching it over my fourteen-year-old son's shoulder. He's a manic lepidopterologist with a collection that's forced us to build a loft in his room. With two minor discoveries and a paragraph in *American Lepidopterologist* to his credit, he almost had his entire collection up as a database on some other company's little home computer when the machine broke down on him. But since the SORD M23 and PIPS, he's been producing slick SORTS and Conditional Searches of his 2,200-item collection with nary a whine from his crisp, silent SORD five-inch disk drive.

Two days later the big white SORD boxes were in the office, and the M68s were arrayed in the executive suite—impressive, with their clean lines and smart, dark grey design. "But how to run them?" was Bernie's question, though he was still sighing with relief at the reasonableness of the check he had just signed.

"Not to worry, Bernie. I'll come in this Saturday and load PIPS into one of these beauties. In one day I'll be able to load in an initial data set and create some basic framework with a few simple autoprograms. Then on Monday it'll be ready for Hank."

As good as my word, I gave that Saturday up to in-loading much of the data that had figured most frequently in the Friday-morning memos with which we were so familiar. The piece of that work that I want to offer in this article deserves some attention because it's universal: I'm referring to that longed-for item, the year-to-date income statement. Too often found on the doodle pads and napkins of the people who have to worry about it most, the year-to-date income statement surely deserves the dignity of a PIPS presentation. Not to mention another PIPS attribute: it deserves to be right.

I knew that a full, forty-eight-line balance sheet would be satisfactory from my accountant's point of view, but that Hank would be in too much of a hurry to appreciate it. So I scratched out a shortened, twenty-line version that even I had to admit to myself was basically satisfactory. All the significant moving parts were there, and it provided a pattern that would work in indexing potential performance, which was what Hank would use it for most.

With PIPS running I typed O for Original File. For a title I gave INCOME STATEMENT: YEAR TO DATE and pressed [RETURN]. I only needed two columns: one to provide labeling space for each line, and another for the figure that would go on it. I pressed [RETURN] from the SET: query, giving me a header of four lines. (I later used the DL command to get rid of two of them, as I had no use for them.) Then I used the MT command to lay in the titles of all the lines. The result is shown below.

```
<INCOME STATEMENT: YEAR TO DATE> ( 3.12.83) F24
[ SPRING=89 SUMMER=90 FALL=91 WINTER=92 ]
SALES AND OTHER INCOME
NET SALES
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER
TOTAL SALES AND OTHER INCOME
COSTS AND EXPENSES
COST OF SALES
SELLING, GENERAL AND ADMINISTRATIVE
RESEARCH AND DEVELOPMENT
INTEREST
OTHER
TOTAL COSTS AND EXPENSES
INCOME BEFORE TAXES
INCOME TAXES, CURRENT
```



```

NET INCOME
RETAINED EARNINGS
BALANCE AT BEGINNING OF PERIOD
CASH DIVIDENDS
BALANCE AT END OF PERIOD

```

```

comment :
f50,15,/4

```

The first line under the title refers to the page numbers of the four quarterly income statement pages that I would shortly set up on the PIPS disk. Spring would go on page 89, summer on 90, fall on 91 and winter on 92. This prompt in the page itself would, I planned, be necessary for Hank to run the process of addition that I had in mind.

The four lines ending with SALES AND OTHER INCOME; the seven ending with TOTAL COSTS AND EXPENSES; and the three ending with NET INCOME each comprise a separate section within this twenty-line income statement. On the PIPS screen, I used the COL command to break out these sections in separate colors; for example, the first section was made sky-blue with the instruction R2,R5,16. It's easy to go to town with the COL command, trying various combinations until you strike one that seems precisely right for that table. This INCOME STATEMENT page is especially well suited to COL.

Once you've set a page format up once, you can copy it with a few keystrokes into as many pages as you have need for. So with this INCOME STATEMENT: YEAR-TO-DATE page finished the way I liked it, I kept it in the buffer and PUT it in rapid succession to page 89, page 90, page 91, and page 92.

Now I had five identical pages. I had to at least separate them by title, so that they could be easily

distinguished on the screen. On the other hand, it was essential to keep their formats all the same: the operations I had in mind required this. So I used GET to retrieve page 89 and then typed CT [RETURN] to change its title to QUARTERLY INCOME STATEMENT—SPRING. Not satisfied with that, I used the COL command to color the title line an easily noticeable reverse-field green (COL;R;;R1,1,12;ESC). Then, by GETting pages 90, 91 and 92 in succession, I retitled them QUARTERLY INCOME STATEMENT—SUMMER, FALL, and WINTER, respectively. And on each I used COL to turn the title lines light blue, magenta and white, respectively. Now I tried GETting pages 89 through 93 in rapid succession. They were uniform except for the title line, which flashed in color transformations that made the fact that they were different pages unmistakable.

Then, just to have a full set of data, I entered last year's four quarters. Before doing so, I PUT the four blank versions elsewhere on the disk—they would be used again and again. They were to be one of Hank's playgrounds in PIPS, pages on which he could sketch out possible years again and again. Then he would put his speculative pages into place on the page numbers that I would assign in the autoprogram that would run it all. Then with the WRITE command, I entered the data that fills the four pages you see below:

```

<QUARTERLY INCOME STATEMENT--SPRING>                ( 4.10.83) F24

```

SALES AND OTHER INCOME	
NET SALES	925,766
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER	20,335
TOTAL SALES AND OTHER INCOME	946,101
COSTS AND EXPENSES	
COST OF SALES	656,816
SELLING, GENERAL AND ADMINISTRATIVE	162,019
RESEARCH AND DEVELOPMENT	42,495
INTEREST	38,590
OTHER	6,667
TOTAL COSTS AND EXPENSES	906,587
INCOME BEFORE TAXES	39,514
INCOME TAXES, CURRENT	20,152
NET INCOME	19,362
RETAINED EARNINGS	
BALANCE AT BEGINNING OF PERIOD	44,827
CASH DIVIDENDS	-4,683
BALANCE AT END OF PERIOD	59,506

```

comment :
f50,15,/4

```


<QUARTERLY INCOME STATEMENT--SUMMER>

(25.10.83) F24

SALES AND OTHER INCOME	
NET SALES	1,510,460
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER	33,178
TOTAL SALES AND OTHER INCOME	1,543,638
COSTS AND EXPENSES	
COST OF SALES	1,071,647
SELLING, GENERAL AND ADMINISTRATIVE	264,347
RESEARCH AND DEVELOPMENT	69,334
INTEREST	62,963
OTHER	10,878
TOTAL COSTS AND EXPENSES	1,479,168
INCOME BEFORE TAXES	64,470
INCOME TAXES, CURRENT	32,880
NET INCOME	31,591
RETAINED EARNINGS	
BALANCE AT BEGINNING OF PERIOD	
CASH DIVIDENDS	-7,641
BALANCE AT END OF PERIOD	83,456

comment :

f50.15./4

<QUARTERLY INCOME STATEMENT--FALL>

(25.10.83) F24

SALES AND OTHER INCOME	
NET SALES	1,680,996
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER	36,924
TOTAL SALES AND OTHER INCOME	1,717,920
COSTS AND EXPENSES	
COST OF SALES	1,192,639
SELLING, GENERAL AND ADMINISTRATIVE	294,193
RESEARCH AND DEVELOPMENT	77,162
INTEREST	70,071
OTHER	12,106
TOTAL COSTS AND EXPENSES	1,646,171
INCOME BEFORE TAXES	71,749
INCOME TAXES, CURRENT	36,592
NET INCOME	35,157
RETAINED EARNINGS	
BALANCE AT BEGINNING OF PERIOD	83,456
CASH DIVIDENDS	-8,504
BALANCE AT END OF PERIOD	110,109

comment :

f50.15./4

<QUARTERLY INCOME STATEMENT--WINTER>

(25.10.83) F24

SALES AND OTHER INCOME	
NET SALES	1,120,664
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER	24,616
TOTAL SALES AND OTHER INCOME	1,145,280
COSTS AND EXPENSES	
COST OF SALES	795,093
SELLING, GENERAL AND ADMINISTRATIVE	196,128
RESEARCH AND DEVELOPMENT	51,441
INTEREST	46,714

OTHER	8,070
TOTAL COSTS AND EXPENSES	1,097,447
INCOME BEFORE TAXES	47,833
INCOME TAXES, CURRENT	24,395
NET INCOME	23,438
RETAINED EARNINGS	
BALANCE AT BEGINNING OF PERIOD	110,109
CASH DIVIDENDS	-5,669
BALANCE AT END OF PERIOD	127,878

comment :
f50,15, /4

Now I was ready to write the simple autoprogram that would add up any or all of the quarters and put the results in the YEAR-TO-DATE balance sheet. Typing O for Original File and entering the six-character name of my autoprogram—TODATE—I switched into Edit mode and was ready to compose. Between the PIPS manual

and my own memory of elements of PIPS that my son had demonstrated for me, I attained my goal in just a few minutes. The autoprogram worked, and what was truly shocking, it did a tremendous amount of work with only four lines of code. Here is the autoprogram's complete text:

```
<TODATE>                                ( 3,12,83) N7
G;93;CAL/C;F,;D1,1;P;?;EADD(M5,2)=M5,2;EADD(M6,2)=M6,2;EADD(M7,2)=M7,2
EADD(M9,2)=M9,2;EADD(M10,2)=M10,2;EADD(M11,2)=M11,2;EADD(M12,2)=M12,2
EADD(M13,2)=M13,2;EADD(M14,2)=M14,2;EADD(M15,2)=M15,2;EADD(M16,2)=M16,2
EADD(M17,2)=M17,2;#89M19,2=M19,2;EADD(M20,2)=M20,2;ADD(R17,20,2)=M21,2;ESC
STOP
*
```

The program begins with G;93 (GET page 93). This brings the blank YEAR-TO-DATE INCOME STATEMENT page into the master buffer. Then the calculation mode is switched on for an indefinite series of operations with CAL/C. To make sure that all calculations are displayed in commas, the F, comma instruction is employed. And since all calculations should be in rounded dollars, the decimal instruction is set in advance to D1.1.

Then comes the crucial P. This sets up a mode within a mode, signaling that calculations will be across a block of consecutive pages. The ? mark that follows is the only user input that this autoprogram requires. Here Hank could decide which consecutive quarters he wanted to add together and have appear on page 93. To add together SPRING and SUMMER, he would use a comma, typing 89,90. To add together all four quarters, the syntax would be the same: he would type 89,92.

For this application, I needed to add together not columns, but matrix locations. If I added column 2 across all pages, I would only create more work for myself: the program would have to straighten out lines 19 through 21 after the party was over. Better to add up each matrix location at a time, I thought, as I entered my first addition statement: EADD(M5,2)=M5,2.

In this statement, the E is the indicator that tells PIPS that the following equation applies to the block of pages that was specified right after the P was given. Thanks to the E, PIPS knows that ADD(M5,2)=M5,2 really means "add all the matrix: line 5, column 2 locations together,

and display the result at the same matrix location in the master buffer."

The first three lines of this short autoprogram simply repeat this action for each relevant line—as follows:

```
NET SALES
EADD(M5,2)=M5,2
INTEREST, GAIN ON SECURITIES, DIVIDENDS,
OTHER
EADD(M6,2)=M6,2
TOTAL SALES AND OTHER INCOME
EADD(M7,2)=M7,2
COST OF SALES
EADD(M9,2)=M9,2
SELLING, GENERAL AND ADMINISTRATIVE
EADD(M10,2)=M10,2
RESEARCH AND DEVELOPMENT
EADD(M11,2)=M11,2
INTEREST
EADD(M12,2)=M12,2
OTHER
EADD(M13,2)=M13,2
TOTAL COSTS AND EXPENSES
EADD(M14,2)=M14,2
INCOME BEFORE TAXES
EADD(M15,2)=M15,2
INCOME TAXES, CURRENT
EADD(M16,2)=M16,2
NET INCOME
EADD(M17,2)=M17,2
```


The RETAINED EARNINGS line I simply consolidated with BALANCE AT BEGINNING OF PERIOD which follows it; not good accounting practice, I admit, but suitable for this purpose.

To deal with line 19 (BALANCE AT BEGINNING OF PERIOD) I had to bring over unchanged the figure at the same line of the first quarter, as that was the relevant figure for the beginning of the year. This required a different kind of statement, still legal under the P qualification under which all this addition had been transacted. To specify a matrix position on page 89 (SPRING), I had to use a # mark preceding the page number. So the statement

#89M19,2=M19,2

on the fourth line means "page 89's matrix position line 19, column 2 equals line 19, column 2 in the master buffer."

Line 20, CASH DIVIDENDS, on the other hand, worked like the others: the statement EADD(M20,2)=M20,2 met the case without any difficulty.

The payoff line, BALANCE AT END OF PERIOD,

was a calculation strictly among the figures now in the master buffer that had just been added through by the previous lines of the program. With the statement ADD [R17,20,2=M21,2, rows 17 through 20 of column 2 of the master buffer were added in the buffer's matrix position line 21, column 2—and there was the proverbial bottom line.

It's important to note that while all the calculations in this autoprogram that go across pages use parentheses, all that don't use brackets. Otherwise, the syntax is the same. When you try this yourself, make sure you don't get the two muddled.

Explaining the autoprogram statement by statement, of course, doesn't tell you what a satisfaction it was to run. All it required of the user was that he input the quarters he wanted to see summarized. Then the figures would start to appear from top to bottom in rapid succession, apparently popping up out of nowhere. I put the initial run statement—AU#TODATE—into a function key to make it all even simpler.

```
<INCOME STATEMENT: YEAR TO DATE> ( 3.12.83) F24
[ SPRING=89 SUMMER=90 FALL=91 WINTER=92 ]
SALES AND OTHER INCOME
NET SALES 5,237,886
INTEREST, GAIN ON SECURITIES, DIVIDENDS, OTHER 115,053
TOTAL SALES AND OTHER INCOME 5,352,939
COSTS AND EXPENSES
COST OF SALES 3,716,195
SELLING, GENERAL AND ADMINISTRATIVE 916,687
RESEARCH AND DEVELOPMENT 240,432
INTEREST 218,338
OTHER 37,721
TOTAL COSTS AND EXPENSES 5,129,373
INCOME BEFORE TAXES 223,566
INCOME TAXES, CURRENT 114,019
NET INCOME 109,548
RETAINED EARNINGS
BALANCE AT BEGINNING OF PERIOD 44,827
CASH DIVIDENDS -26,497
BALANCE AT END OF PERIOD 127,878
comment :
f50,15, /4
```

When Monday morning came around, Bernie stuck in his head. He was checking me for fatigue marks. But I had no rings around the eyes to offer him.

"How'd it go? You weren't in here all weekend trying to get the computer ready, were you?"

"No way, Bernie. I went home on Saturday at three o'clock and had an early dinner."

"Does it work?"

"Sure it works. Like a charm. Is Hank in yet?"

He was. We went to see him and explained that the accounting department had taken some steps to provide management with information resources more efficiently than we had done in the past.

Then we led him to the M68. I gave a little speech, explained how free he would be to fiddle with PIPS himself as soon as he was used to it, and put his finger on the function key button.

As the numbers materialized down the brightly colored lines, Hank's eyes were even brighter. I knew I'd succeeded in giving my boss something he'd always wanted: a license to dream.

**PRODUCT
REVIEW**

SORD'S M68 – PERFORMANCE PAR EXCELLENCE



This issue we explore the anatomy of a most unusual and powerful system—the SORD M68. Hardware and software compatibility in a fast and powerful system with the potential and power of 16-bit based architecture allows the SORD M68 to offer more and cost less than all other available systems. In this SORD WORLD exclusive, we explore the systems architecture and extensive software of the M68 from the user's perspective.

When Katherine Curtis, a young Columbus Ohio financial analyst, started looking for a computer, an inveterate hacker friend recommended she carefully examine the performance, features, and price of the SORD M68.

"I was tired of looking at desktop computers that were expensive without offering the performance that 16-bit microprocessor technology and a standard operating system could provide," declared Katherine. "When I saw the M68 I realized I finally had found state-of-the-art hardware that preserved compatibility with my software collection. I didn't want to buy a new computer and have to start from scratch. I had already purchased a lot of expensive software that had proven to be useful in my work. Although that was before I had discovered PIPS III and the large line of friendly and productive software available from SORD, there were still many programs from work and friends which I had to continue using. Even more important was the fact that I had written many of my own FORTRAN, BASIC, and PASCAL programs on my old brand-X 8-bit computer. These programs represented a considerable investment in programming time and I had become very dependent on them for carrying out my everyday work routine. I didn't want to throw them away and I certainly didn't want to have to rewrite them for a different computer. I needed portability so that my old programs would run on my new machine. I also wanted to move up to a more powerful computer system which used a 16-bit microprocessor. The SORD M68 was the only machine that combined the best of all possible worlds—16-bit performance without relegating 8-bit machine programs to the scrap heap. I got performance plus versatility at a price I could afford."

Katherine is just one of many delighted SORD M68 owners around the world. From Austria to Australia, the 16-bit plus 8-bit dual performance of the SORD M68 has proven to be the ideal move up for seasoned microcomputer users and it's also a flexible and friendly introduction to the world of computing for the serious beginner.

What makes the SORD M68 so special? Careful attention to the user and a prescient awareness of hardware and software trends were priorities for SORD engineers and product development teams from the inception of the M68 development project. SORD's aggressive and innovative engineers set out to make a computer that could bridge the 8-bit/16-bit chasm that restricted desktop computing to either the 8-bit or 16-bit mode. The SORD M68 is the only microcomputer that can live comfortably with two different bit-modes without sacrificing performance. Now mainframe performance made possible by 16-bit architecture comes to the desktop at a price that few 8-bit machines can match.

Taking a peek at the SORD M68's internal structure reveals solid engineering that exemplifies the elegant simplicity that has given SORD an international reputation for quality and performance.

The SORD M68 is a true 16-bit computer which utilizes a Zilog 80A microprocessor for input/output operations. In the 16-bit mode, the Zilog 80A microprocessor is used as an extremely efficient input/output system. In the 8-bit mode, the Z80A microprocessor is a completely independent 8-bit central processing unit. The extremely efficient nested architecture of the SORD M68 makes it possible for you to have access to two of the most popular personal computing systems which are presently available. By flicking a switch, you can easily change the SORD M68 from an 8-bit machine to a 16-bit machine. The versatility and convenience of this clever system make it possible for the SORD M68 to incorporate the best features of the most popular and performance proven personal computers presently available.

The research and development staff at SORD carefully selected the two microprocessors which are the hearts and minds of the SORD M68. The Motorola 68000 16-bit general-purpose second generation microprocessor provides the SORD M68 with a data processing speed of 10 megahertz—performance comparable to a mainframe computer on a desktop. The 16-bit CPU of the SORD M68 gives the system 256K of random access memory. By using a parity bit, random access memory is expandable to an awesome one megabyte. The direct address space—the number of actual physical locations for data—is 16 megabytes. With memory board expansion the SORD M68 can have 4 megabytes of random access memory. The Z80A 8-bit microprocessor is used as an

PRODUCT REVIEW

extremely efficient input/output processor when the SORD M68 is in the 16-bit mode. This powerful yet flexible design means that bottlenecks in getting information to and from peripheral devices are no longer a problem. You will no longer find yourself waiting for your printer to finish before going onto the next step of your program.

The renowned Z80A microprocessor is the central brain of one of the best 8-bit systems available anywhere. It is the microprocessor of choice for most 8-bit systems today because of its reputation for performance, dependability, and price. The SORD M68 has captured the most desirable features of 8-bit personal computers. It is based on a design which has become the de facto industrial standard. At the same time, features that other companies offer as options are standard on the SORD M68. A wide range of SORD software is based on the Z80A microprocessor. Consequently, the M68 provides the user with a huge package of software from the moment of purchase.

The Z80A microprocessor has established a reputation for efficiency and reliability in the highly competitive world of microelectronics. The Z80A is the silicon soul of many 8-bit microcomputers. The popularity and reputation of Z80-based systems has resulted in a plethora of available software for the SORD M68.

Consider the wide spectrum of software that is available for the SORD M68. There are thousands of games and other entertainment software packages that you can now purchase for 8-bit computers. The fact that the SORD M68 runs four operating systems—CP/M-68K, the UCSD p-SYSTEM, SORD's own KDOS, and SB-80—allows the widest range of software compatibility. With the optional SC88 software card, you can even run software available for computers based on the popular INTEL 8088 16-bit microprocessor. This means that the entire range of software written for IBM personal computers can be used on your SORD M68. The optional SB-80 operating system makes the M68 compatible with a wide selection of CP/M-80 software packages.

SORD's own KDOS operating system allows you to use several different languages when the M68 is in the 8-bit mode. Access to all programmable features of the M68 is available by using RASM—the Relocatable Assembler Language. High-level numerical and general purpose programming is easy with the flexible and full-featured FORTRAN language. Business applications

programming on the M68 is convenient with COBOL, a mainstay language for commercial use which allows the programmer to write source statements in English. BASIC II and BASIC(C) are standard features of the 8-bit M68 mode. For the maximum in user friendly languages, SORD's own PIPS III language is a standard feature of the M68.

The marvelous Motorola 68000 microprocessor is the foundation of many 16-bit microcomputer systems because of its high-performance properties. Only SORD has had the vision and concern for the end user to take the best components that today's microprocessor technology has to offer and create a balanced and innovative product for the bewildered consumer looking for a reliable and enduring microcomputer investment in the shifting sands of contemporary computer technology. There is no need to worry about buying a computer today that will be outmoded tomorrow—the SORD M68 provides the most advanced technology available while preserving the usefulness, reliability, and software support of the 8-bit computer.

Determined to unravel some of the mystery surrounding the art and artifice of contemporary computer technology, SORD WORLD recently paid a visit to Michael P. Rosser, a prominent Tokyo computer consultant for several Japanese high-tech firms, and asked him to explain the significance of using the Motorola 68000 in the SORD M68.

"I think the Motorola 68000 is one of the best 16-bit chips available on the market today," stated Rosser with calm assurance as he sat in his plush office overlooking the Keio University campus in central Tokyo. He continued with a tone of voice which conveyed the experience of a veteran industry observer and microelectronics expert, "The 68000 has become the most popular processor for small computers because it has an enormous capacity for being upgraded. Whoever adopts the 68000 microprocessor as a central component of a computer will be able to move painlessly and smoothly into the next generation of computer technology. Upward compatibility through hardware will give 68000-based machines a definite advantage over other computers that have to rely on software compatibility alone. Computers are compatible only if programs can be successfully run on either one without alteration. Sometimes this compatibility is achieved only through extensive software systems at the cost of com-

puting speed and the availability of memory for other purposes.

Just over the horizon are the Motorola 68000-10 and 68000-20 chips. These chips represent a significant advance over other 16-bit chips. The 68000-10 has virtual memory which allows a programmer to make use of storage resources without having to consider the physical constraints on memory or possible requirements of other applications.

The 68000-20 will feature pure 32-bit input/output operations. The most wonderful aspect of these two developments is that all 68000-series microprocessors are compatible. The fact that SORD has adopted the 68000 shows the company's resolve to maintain program compatibility and portability in the future. Most important of all, the next generation of software will require increased compatibility through hardware—not just software. The SORD M68 is the right choice for anyone wishing to ride the wave of the future. It's amazing that SORD has presented the consumer with a powerful and integrated software and hardware package in the M68 computer at such a low price."

Mr. Rosser's remarks serve to underscore what SORD has known all along—the SORD M68 is the most powerful and versatile system available for its price. SORD products are recognized for built-in quality that begins with system architecture that is powerful, elegant, compatible, and portable. Software support from SORD and compatibility with 8-bit and 16-bit software make the M68 easy to use for both the novice computer user and the seasoned computer professional.

One of the many features that attracted Katherine Curtis to the M68 is the fact that the M68 has four operating systems available. Consequently, the widest possible range of software compatibility is virtually guaranteed. At a time when several different operating systems are vying for the attention of consumers in a very competitive market, SORD offers you the option of choosing the operating system that you want. At present, owning an SORD M68 computer allows you to choose from several standard operating systems. You can choose either the popular CP/M-68 operating system from Digital Research, the UCSD P-system, or SORD'S own KDOS.

The operating system is of critical importance for the efficient utilization of software. A computer is only as efficient as the operating system which supervises and

controls the running of user-oriented programs. There are a number of languages and programs available for the SORD M68 under the CP/M-68K operating system. There is a C-language compiler which gives you access to the powerful C-language which has gained a wide following among old-timer and neophyte programmers alike since its generality makes it convenient and effective for writing entire operating systems. For the computer owner who desires to get into the bits and bytes of his SORD M68, the C-language provides contemporary programming's most widely acclaimed means for writing operating systems as well as major text-processing, numerical, and data base programs.

For number-crunching and other scientific applications, FORTRAN-77 is present on the M68. A microcomputer without FORTRAN would be quite a shock to many scientific programmers as well as those who cut their teeth on FORTRAN a decade ago before the microcomputer revolution. The FORTRAN-77 on the M68 is not the stripped-down version you will find on many other personal computers. SORD has taken special care to make sure that you get the full power and flexibility of FORTRAN-77 and not a watered-down version.

Another high-level language under CP/M on the M68 is the well known APL—whose cryptic acronym is derived from A Programming Language. For those with a mathematical bent, APL is very useful because it has a syntax and character set which is especially appropriate for mathematical applications, particularly those involving arrays. The next time you invert a large matrix, think how easy it would be if you had a high level language like APL available on your microcomputer. The BASIC language was a watershed for ushering in a widespread appreciation of high-level languages for beginning and experienced programmers alike. You have the full power of this extremely popular language on the M68 with BASIC-PLUS. The SORD M68 gives you the full power and flexibility of the BASIC language so this powerful and standard language is available for all of your computing needs.

Wenda Wall programs in BASIC-PLUS on her SORD M68. However, for most of her computing needs she has found PIPS-III to be the friendliest and most productive high level language. Katherine Curtis used to do most of her work using one of the many popular spreadsheet programs. Then PIPS came along and she found

PRODUCT REVIEW

that for financial analysis there was no better way to get to the bottom-line with speed and efficiency.

Pausing to water the plants in her office—her PIPS memorandum informed her this morning that today was the day to water the philodendron—Katherine reflectively stated, “I still have many friends, mostly engineers and scientists, who spend a lot of time crunching numbers with FORTRAN and BASIC. But for my purposes I think that I get more mileage out of PIPS on my SORD M68 than any other program and I find that PIPS makes programming so easy that I can get more done and actually enjoy the time I spend doing it. I’m still glad that I have BASIC-PLUS and FORTRAN-77 on my SORD M68. But sometimes I think that if it weren’t for all the programs my friends write in BASIC, FORTRAN, and PASCAL, it wouldn’t make much difference to me if every program ever written in those languages disappeared without a trace. Of course, I hope they don’t disappear until after I have a chance to find out a more convenient way to get my programming done using PIPs. I still use BASIC and FORTRAN when I have to do a great amount of number crunching, but hopefully those days are over. I was really amazed when I found out that the SORD M68 was capable of using mainframe applications programs as is—no tinkering, dial twiddling, or softcarding necessary. It really was surprising that SORD’s FORTRAN-77, C, and PASCAL are compatible with their mainframe counterparts. Now I can just take canned programs home from the office and put them right on my M68. Of course, I find that I get so much more accomplished with PIPS-III that I very rarely have to take work home. But when important job deadlines are approaching, I know that I don’t have to stay at the office to get everything done. I can take it home and get my financial reporting done in no time on my M68 by using PIPS-III and the SORD word processor. The ultimate electronic office is located right in my den. Teaming up PIPS and the SORD word processor brought a new era of convenience and productivity. Next I plan to use SORD’s S-NET and network with the other members of my company’s corporate finance group so that we can have meetings without leaving our terminals.” Establishing a local area network is easy with the SORD M68.

Harlem Rodriguez is an applications programmer for Katherine’s office and even with many years of mainframe experience he finds that the SORD M68 is the appropriate

vehicle for much of his office systems development.

“The SORD M68 was a real sleeper at first. The people at purchasing recognized a bargain when they saw it. There just isn’t a computer available at its price that has so many standard features, so much flexibility, and such extensive systems compatibility. I use a SORD M68 for much of my work debugging electronic data processing applications. Once I get the programs debugged, we get them up on the mainframe for production runs. Our purchase of a SORD M68 was a doubly cost effective decision. Not only was it twenty to thirty percent cheaper than competing products which didn’t have nearly the computing power but it also saved the firm a bundle because now we don’t have to purchase more time on a mainframe for program development. Since much of my systems development programming is in C, mainframe applications are easily written and debugged on our SORD M68. The SORD product line has allowed our programming section to be at the forefront of EDP technology at a cheaper price than we ever believed to be possible.”

If data communications limitations are a bottleneck for your office automation plans, the SORD M68 provides the hardware that will make your life a little easier. The M68 allows a variety of communications configurations that make it easy for office automation applications. One of the most useful features of the M68 is that bisynchronous communications are possible. With your SORD M68 it will be easy to talk to your mainframe.

The IBM 3270 terminal uses bisynchronous communications which makes it much easier to convey large amounts of information to a mainframe computer. An IBM 3270 emulator is available for the SORD M68. Using the emulator, your SORD M68 can pretend it is an IBM 3270 terminal. This means that your SORD M68 is able to emulate every nuance of the IBM 3270’s repertoire and you are ready to take advantage of a de facto industry standard and plug into the burgeoning systems of networks around the world.

What does this mean for users like Katherine? First of all it means that her SORD M68 is not just an isolated work station. It has the ability to be a powerful data communications center which can be part of a local area network which may include non-SORD computers. Data transferability is easy and much time and effort is saved by eliminating the physical movement of documents.

When Katherine needs a spread sheet from another member of her work group, electronic mail allows her to copy a page or even a series of pages from her partner's PIPS spreadsheet. The powerful SORD M68 system takes the future dream of the electronic office and makes it a present reality. Bisynchronous communications make possible high speed data transmission among members of Katherine's network. At the data transmission speed of one megabyte per second, she doesn't have to wait for data to trickle in from other members of her network.

The best way to introduce yourself to the M68 is to take the system for a drive around the network. Touch the ultra-thin keyboard and adjust the angle for maximum comfort. The cylindrical stepped sculptured keys give the keyboard the feeling of a precision instrument. The palm rest helps reduce fatigue that might occur during extended conversations with the M68. Look at the terminal and your eyes benefit from the non-reflective display and a non-interlace system which eliminates image flicker. Adjust the screen for maximum ease of perception with the tilt adjustment that allows you to control the image angle.

"It was what I didn't notice that first impressed me with the user comfort of the M68," explained Katherine. "All the other systems were uncomfortable in some way. With the M68, I could tailor it to the feel that I wanted so that I could sit at my work station for many hours without walking away with glazed eyes and a backache. I never thought word processing could be so easy or so comfortable."

External memory for SORD M68 can have many different configurations. You can choose from either 5.25 or 8 inch floppy disks and hard Winchester disks. The 8 inch Winchester hard disk can store 20 megabytes of data—enough memory to store several unabridged English dictionaries on one disk.

The SORD M68 has several other features that make life easy for the user. A Centronics interface connects to your printer with ease and a built-in DIN5C connector is available for your Light Pen. Floppy and hard disk interfaces use built-in pin connectors. Scientific instruments interface with the SORD M68 through the standard GP- connector. Five expansion slots are available for optional system configurations.

The graphics capabilities of the SORD M68 are an illustrator's dream. A unique Palette function makes possible a range of 16 colors from which the user can

choose. Because the text display and graphics display are separate, you can display and examine your programs while simultaneously looking at the result. High resolution graphics permit you to see the full beauty and texture of the contemporary cornucopia of available software graphics.

In the 16-bit mode, full featured and undiluted versions of popular high-level languages are standard with the M68. When Katherine Curtis got her M68 she was surprised to find out just what she had been missing with her previous stripped-down version of BASIC. Now with BASIC-PLUS she is able to experience the full power of BASIC.

If you want to get started programming from a "structured programming" perspective, there is nothing more valuable than having the high level language PASCAL at your fingertips. The SORD M68 gives you PASCAL—a prima donna programming language which makes structured programming second nature for both beginning and advanced users.

The bottom line with the SORD M68 is performance. With even more features than comparable systems, something made possible by innovative design and quality hardware and software, the SORD M68 still has quite a reasonable price tag. Not only can you run the most popular compatible software on the M68, but you can also communicate with every other major computer system in operation by using SORD'S excellent communications packages.

"The SORD M68 solved my most complex financial problem: how to buy a powerful computer system that could handle all my computing needs that I would not outgrow before I fully amortized my investment," declared Kathy. "Instead of taking my word for it, I advise SORD-WORLD readers to check out the M68 for themselves."

Need we say more? Go to your local SORD dealer and see the most versatile and cost effective 8-bit/16-bit computer system available today.

TAKAYOSHI SHIINA, JAPAN'S COMPUTER REVOLUTIONARY

by Roy Garner



*This article first appeared in
PHP International Edition,
October 1983, and is
reprinted with permission.*

Takayoshi Shiina at his company's showroom



When Takayoshi Shiina gets up to speak before a mixed group of Westerners and Japanese one can almost hear the ice breaking. Journalists, long jaded by the obscure discourse of overcautious business leaders, reach for their notebooks with new-found hope, while the businessmen in the audience put their gin and tonics to one side with looks of surprise and curiosity. The whispered consensus is, "Good heavens, here's someone we can actually understand." Shiina, both literally and figuratively, "speaks our language."

Such is the impact of the youthful and ebullient president of SORD Computer Corporation, a straight-talking extrovert who is at his happiest when he is communicating successfully with as wide a variety of people as possible and, ideally, exchanging views on one of his favorite topics: making computers accessible, easily understandable, and cheaply available to ordinary people. Shiina's name is now firmly associated with his success in the computer field—SORD is claimed to be the only company in the history of Japan to have gone from sales of zero to ¥10 billion (US\$41.6 million) in less than ten years, and its products are renowned for their simplicity and adaptability. Yet this line of business is essentially only the vehicle for the practice of the ideals which Shiina has long adhered to and which are of ultimate importance to him: namely, that people should expect and seek personal fulfillment in their work. Shiina's philosophical and idealistic approach to his work occasionally borders on the whimsical: for example, he labels his basic principles of successful business management his "Eleven Commandments." This leads even liberal observers to treat him, at times, with some skepticism.

Yet his obvious business success, the stimulating atmosphere of his company work places and showrooms, and his position at the center of an industry of paramount importance to contemporary society, make him a widely respected figure in the Japanese business world.

Shiina's meteoric rise to success has been achieved by the application of individual flair and entrepreneurial energies, by acts of assertive independence, and by frequent trappings on convention. All these appear strikingly unJapanese, and he is widely regarded in his own country as a maverick. His ease of dialogue with Westerners in itself also sets him slightly apart.

Underneath these externals, however, Shiina's family and philosophical roots are as traditionally Japanese as one could find—going back, in fact, to the Bushido (way of the warrior) spirit of samurai family ancestors.

Even in Japan, where the education and career paths of the average child are mapped out by anxious parents at birth, one of the most difficult questions a young person ultimately has to face is whether to follow a conventional path through life or whether instead to dare to follow the more difficult path of his, or her, dreams.

For Takayoshi Shiina the choice was apparently never

at issue. From his childhood days in a family of impoverished returnees from China, through to his present position as head of an internationally famous company, Shiina has thrived on his determination to dispense with convention wherever it threatens to inhibit his dogged pursuit of individual fulfillment.

He holds onto the cherished notion that dreams are a vital part of a happy life and, in his own words, doesn't like "people with no dreams and no ambitions."

Shiina's own dreams began to take shape at an early age. Indeed, he claims it was his rigorous childhood experiences that allowed him to firmly establish his own personal aims and philosophy by the time he was just nineteen years old. Some of his earliest memories center on the year 1946, when, as a three-year-old, he and his family were repatriated from Beijing to his father's birthplace in Kujukurihama, in Chiba Prefecture. Times in wartorn China had been very hard, and Shiina reminisces that "even in my mother's womb, I could hear the bombs and the guns." The return to Japan offered little comfort. For one thing, all the family's belongings had been lost or destroyed in China, and, with Japan also in ruins, the family had to struggle just to get by. In addition, the young Shiina found that to be a born-in-China repatriate meant he was considered inferior by his peers. These conditions combined to give Shiina what he describes as "an aggressiveness, a challenge, and a desire for a better future for Japan." But he also acknowledges the debt of gratitude he owes to his mother, who, true to tradition, made great sacrifices to ensure that he obtained the best possible education, and to his maternal uncle, Wakamatsu Ono.

It was undoubtedly Ono who provided the most crucial aspects of Shiina's development, and he, in fact, is still referred to by Shiina as "my guru." Ono was a scholar by nature, and had majored in political science and economics, but he had also spent many years as a railwayman in Manchuria until he returned to Japan to run his own small camera business.

Shiina remembers how "every night after 10 or 11 P.M. he would give me one or two hours of lecturing... every night! He gave me a lot of ideas." Ono talked to Shiina about the meaning of "human beings, religions, philosophies, human relations, and so forth," driven by the desire, as he explained it, to "encourage young people in the reconstruction of Japan."

Shiina attributes Ono's remarkable talents, and depth of vision, to a "Bushido mind and spirit," stemming from his family's proud history as members of Japan's samurai warrior class. As a feature of the family's perceived role of social leadership, many of Shiina's ancestors had gone to work on projects in distant places. In the early nineteenth century, Shiina's great-grandfather went to Okinawa, his grandfather later went to Korea, while Ono, and Shiina's direct family, all ended up in mainland China, a family



Shiina discusses matters with some of the members of his young (average age 27) staff at SORD.

background which helps to explain Shiina's present natural communication skills with non-Japanese businessmen.

Shiina claims that the traditional values and aspirations of his pioneering ancestors, passed down through Ono, "still survive in the mind and give me a system of thinking, help me to challenge new things."

Inspired by Ono, Shiina took an interest in business and worked enthusiastically, from the age of fourteen, in his uncle's camera shop "never taking any salary and enjoying a very good opportunity to learn."

But even then, Shiina's interests were never on a small scale. He disregarded the "small matters" of book-keeping and accounting, and concentrated instead on "the whole range of management strategy and of ambitious planning for the future." Shiina says that "even when I was at university doing an electrical course, I never touched any of the equipment. I was just looking at and 'behind' the people. I always wanted to be a kind of corporate-strategy-type person, never an engineer. Personnel management was a key interest."

This thinking led Shiina to believe strongly in the delegation of work to others, "to find others to do what I cannot." In later years this attitude was to be manifested in Shiina's prowess as one of Japan's first, and most successful, "head-hunters," people who, in the American

mold, lure top figures away from other companies.

"Head-hunting" is no easy task in life-time-employment Japan, but Shiina's magnetic enthusiasm and appeal persuaded Toshiaki Kamijo, the inventor of the Walkman cassette player, to switch from Sony to the SORD fold in November 1981.

Once at university, at the age of twenty, Shiina decided that his time of basic studies and contemplation was passed, and that his next objective was to "test my knowledge and polish up my skills." His first opportunity to experiment in personnel management came after he took up residence in a college dormitory. Shiina noticed that on some waste ground owned by the college, about one kilometer away from the dormitory, there were a few hundred grape vines which had been abandoned by a past landowner.

In a somewhat bizarre venture, Shiina assembled and organized a large group of students, persuaded them of the benefits of having beautiful grape flowers and fruits in front of their own living quarters, and managed to replant all the vines in a single day. Quite what the real value was of shifting a forest's worth of grape vines is beside the point. It was more relevant as Shiina's first public flexing of his entrepreneurial muscles. This, and many other such experiments, helped Shiina to map out what he felt



were the fundamentals of successful business activity. First he would define his objective, then he would establish appropriate motivation for this work force. Next he would endeavor to cultivate a suitable team spirit, and only then go about executing the task. There was one other vital factor, however—aftercare.

This final factor is one which users of SORD computers often cite as being among the company's strongest points. An amusing example of this attention to the customer relates to SORD's early days, when they were attempting to enter the U.S. market. The market assault had mistakenly been started in the middle of the summer vacation, a very slack business period. Shiina visited the new U.S. branch office to encourage his staff and discuss what to do next, and found them virtually at a loss to know how to fill their time. Suddenly the phone rang, and the caller asked if it would be possible to have a minor repair done on his newly purchased machine.

Within thirty minutes the startled customer found the company president, a director, and a chief research engineer on his doorstep. His understandable response was, reportedly, "You Japanese are really something."

Shiina says that "even now I work with the same values," and adds that he is only really satisfied if each ensuing task is another step, a new challenge, beyond the previous venture. Before SORD's genesis, however, Shiina made a couple of false starts. The first consisted of joining a course at the Japanese Self-Defense Force Academy, and the second of signing on with a company he didn't like, only to drop out at the end of the first day with the parting comment to the boss, "You know absolutely nothing about human beings."

The next opening came in October 1963, in a company which was, significantly, handling sales of microcomputers made by the U.S. company, Digital Equipment Corporation (DEC). Shiina's job there quickly led him to believe that the microcomputer field offered just the exciting business opportunity he had always been looking for. Few people in those days understood, or had any experience of, computers, but Shiina, with his knowledge of electronics and his awareness of the needs of the businessman in the office, quickly decided that his new aim in life would be to develop a microcomputer with fully compatible software and hardware (SORD), which would sell for the everyman price of ¥500,000 (US\$2,083). At that time the cheapest machine on the market cost over ¥10 million (US\$41,600).

The rise of the SORD company has all the qualities of a business fairy tale. The enterprise started in a small room equipped with nothing more than a photocopier and a telephone, and capital of around ¥650,000 (US\$2,700). The money was contributed by four friends with computer backgrounds who had listened at the university to Shiina's descriptions of the ideal business venture he one day

planned, and dreamed, he would start.

Shiina's mother acted as managerial partner, and the necessary computer expertise was eventually provided by the same college friends, who soon left their jobs in such prestigious companies as Burroughs and Sharp to assist Shiina in his new computer venture. Suffice it to say that just ten years after the founding of SORD in 1970, an independent research organization found that SORD came top out of 600,000 Japanese companies surveyed as the firm with the fastest growth rate in the previous five-year period. Additionally, SORD's product line features the PIPS (Pan Information Processing System) software, which a recent poll by Japan's leading economics daily, the Nihon Keizai Shimbun, found to be Japan's most popular microcomputer language, used by 22.7 percent of all companies surveyed.

Shiina is a firm believer in the positive good which computers offer, and quickly dispels any notions of such machines as dehumanizing or a threat to employment. Those ideas he puts down to misunderstanding and poor product development. He believes the purpose of computerization is ultimately to "save energy and materials and allow the efficient use of resources." Computers will "allow the removal of unpleasant jobs" and complement the change in society toward education-based and service jobs. Accompanying these social trends, Shiina argues that there should be a greater dialogue in the business and political spheres toward the harmonious integration of high technology into the work place. Additionally, Shiina nurses fervent personal hopes for improved international relations between Japan and its neighbors, his contribution to which, he says, is to "encourage U.S. businessmen in Japan." He warns that "success swings from one country to another, so we should always consider each other. I'm a loudspeaker in Japan on this subject." Shiina's other pet field of interest lies in the continuing improvement of the computers themselves. He believes in starting with the user. "Each person is a specialist, so we should give them tools which reflect the common language of these people. It's not just selling packages." Shiina aims for machines which are first of all "user-friendly," secondly reliable, thirdly very fast in operation, and finally upgradable—to meet the opportunities presented by new technological breakthroughs as they arrive in the future. Asked what he feels has been his greatest achievement in life, Shiina answers that it is certainly "working hard, enjoying life, and having established a happy and successful atmosphere of team work among the over 1,000 people now employed in SORD's Japanese and overseas branches."

Shiina adds that "this is my greatest achievement, and it is also my greatest pleasure...after all, I can't carry money into the grave."

Roy Garner is the Japan correspondent for Computing.

THE SORD WORLD PIPS SCHOOL

A PIPS PRIMER

by Hiroshi Nemoto
SORD Instructor Group

STARTING PIPS III



M23 Mark V

From this issue, I'll be giving a first-hand look at SORD's powerful Pan Information Processing System—PIPS, an acronym you will see many times in these pages—for beginners who've had no prior experience with this system. When I first encountered PIPS, I was myself totally unfamiliar with computers. In fact, even in my wildest dreams I could not have imagined operating one—let alone teaching others to use them as I now do. It was PIPS that changed all that.

Since this is for beginners, I'll be keeping it if not actually in words of one syllable, at least on a simple and non-technical level.

PIPS is an information processing package for a computer. So let's do a simple rundown of the various pieces of hardware which make up a computer system. We'll use the SORD M23 Mark V as our example. The main components of any hardware system are (1) the

main unit; (2) the CRT display; (3) the floppy disk drive and its diskettes; and (4) the printer.

(1) Main unit (with typewriter keyboard)

The main unit is composed of the keyboard and the numerous parts which make up the computer. The keyboard is basically an input device; that is, its purpose is to relay data input or commands to the computer.

The parts which make up the main computer unit include servo, memory and calculation functions as determined by the microprocessor, which also locates and classifies the various separate functions.

(2) CRT display

Similar to a TV monitor in appearance, the purpose of the CRT is to provide a visual display of information stored in the computer. It can therefore be considered a form of the computer's output.

(3) Floppy disk drive and diskette

Next, we have a boxlike device known as a floppy disk drive. This magnetically records and stores data on disks or "diskettes." Although some disk drives may be installed internally within the cabinet of the computer, they are still referred to as "external memory devices."

On a system with two disk drives, Drive 1 holds the master file and can be used entirely for memory storage. The other drive, Drive 0, holds the PIPS software and can also be used for a portion of the subfiles.

At present, SORD offers its M23 models with three different size floppy drives.

(4) Printer

The purpose of the printer is to provide a paper printout of the information appearing on the monitor. It is more commonly referred to as "hard copy." As with the display, the printer is an output device.

The number of characters printed per line is variable: the user has the option of selecting 80 (compressed, this can be extended to 132), 120, or 162 characters per line.

The above covers the basic components which make up the M23 system. Other than these there are a number of peripheral devices for such uses as data communications, and so on.

PIPS is designed to be "user-friendly," that is, to help the user and specifically avoid enmeshing him or her in the confusing complexity of computer programming. It is designed for non-experts, to be responsive to people with little or no computer know-how after the minimum of "hands-on" training. There are three elements at the center of this conception of PIPS. I'll explain them in detail in the following paragraphs, but briefly they are (1) data orientation, which makes the data and not the program the master; (2) page structure, making PIPS consistent with conventional business practices; and (3) a natural command system.

(1) Data Orientation

Originally, putting together a program required a horrendous amount of time and expense. In today's world, the time required to respond to changes in a given situation is critical. To process every new situation or new type of information, however, we cannot expect to have a systems engineer or programmer on hand to handle every crisis. The user simply has to be able to go it alone.

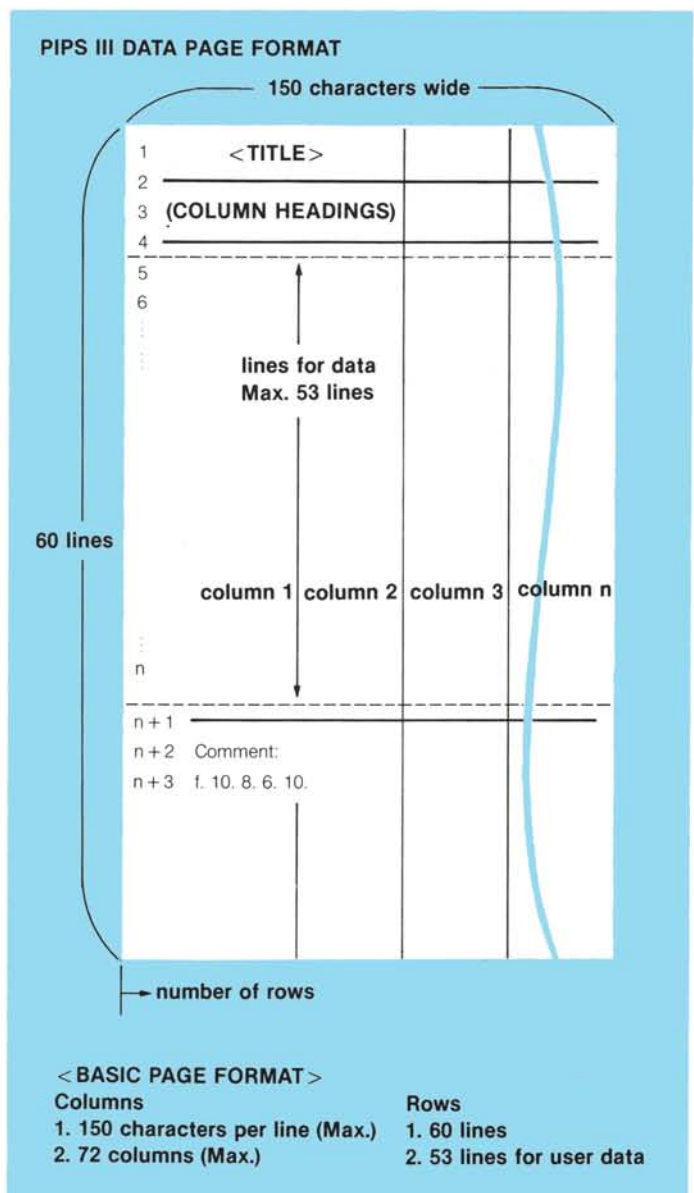
A certain computer analyst in Japan recently wrote "As you know, to build a house, first the architect or contractor must work from a blueprint. To build the house it takes concrete, wood and other materials. But the man who writes the blueprints is not the man who makes the house. Think of the user as the man who has to live in the house. The situation with computers is very similar. Until now, companies have shut their computers away in a separate computer room, where only programmers or system engineers have access to them. But in the future the end user will be the one who makes the program, and we have to consider direct use by the end user himself."

This is where PIPS enters the picture. PIPS is, quite literally, a piece of software developed from the suggestions of users. And it is "data orientated," rather than "program orientated." "Program orientated" means that the program is the master; data can only be entered if it fits into the confines of the particular program being used. "Data orientation" is the opposite; the information is the master and the program responds accordingly. The payoff in terms of flexibility and immediacy is immense. This is reflected in the very brief period of training time needed for the complete neophyte to start using PIPS effectively.

(2) Page structure

SORD has with PIPS tried in every possible way to put the average person at ease and to avoid confronting him or her with unfamiliar situations. So, PIPS is very careful to respect normal business procedures.

Thus, just as the notes, charts, and other materials which find their way into our daily business tend to be in



the form of pages, PIPS also is structured in the form of pages. It's not much different from the bunch of files in your filing cabinet. But it's a lot quicker. Do you have to get up, go to the cabinet, get out the file, take it to your desk, and thumb through it for a minute (or fifteen) to find the page you want? No. You call up your PIPS page without leaving your keyboard. But the process is essentially the same. Only the file is in the computer and the page turns up on the screen. It holds to the format of traditional business procedures. The difference is that unlike conventional filing, no paper is used; instead you store the data electronically.

Take a look now at the format of this page: a page generated by PIPS is basically in the form of a table. Each line of the table is 150 characters wide, and the length of the table is 60 lines. Of this, the format reserves a title section of four lines, and another four-line space at the base for comments, etc. This leaves a maximum of 53 lines for the entry of data.

This page will be inserted in a "master file" containing 96 pages. (There are also 40-page subfiles, but in order to keep our explanation as simple as possible at this point we will confine our explanations to the master file.)

The 96-page files actually represent the number of pages which can be stored on a 8 inch floppy disk. If you were to enter every possible "byte" of data the disk could hold, you would have 96 pages times 53 lines, or approximately 5,000 lines containing roughly 750,000 characters—

Granted, this might be insufficient for certain business operations. But there's a simple and cheap way to expand capacity: more floppy disks.

(3) A natural command system

The final element I want to mention in explaining the basis for PIPS is the command system.

The way we use "command" is not in the military sense; rather, it is the most basic unit by which we may make the computer perform. For example, the command "G" represents "get," in the sense that you want to access and display a particular file from your stored data. Conversely, to return a file to storage, you can command the computer to "put" by typing the "P" command.

By combining these simple to learn commands in the proper sequence, you will find that you are able to perform whatever jobs you need to carry out.

PIPS COMMANDS CHART

BASIC COMMANDS

- O Original. Creates a blank page and allows you to set up the parameters of a new table.
- MT Make Table. Lets you fill a new table with data.

- P Put. Saves the page.
- G Get. Screens the page selected.
- B Shows the master buffer (through which pages are Got or Put to and from external storage).
- L List. Prints out.

BASIC COMMANDS (B)

- GR Graph. Graphs data from a table.
- BT Buffer Titles. Displays directory of pages on disk.
- CT Change Title. Changes a page's title.
- CC Change Comment. Changes a page's comment line.
- DL Delete Line (or group of lines from a table).
- DRL Draw Line. Draws vertical and horizontal lines to make tables clearer in appearance.
- IR Insert Row anywhere on a table.
- RF Reset Format. Aligns a column against L or R margin.
- CF Change Format. Widens or narrows a chosen column.
- NF-N New Format. Alters column parameters without moving data.
- NF-P New Format—Put. Same as above, but also stores previous parameters so that they can be recalled by New Format—Get.
- CA Column Arrange. Rearranges column order. This has 3 branches: D, A and U. D (docking) rearranges columns within one or more tables. A (arrange) sorts for common entries between two tables. U (update) has a mathematical function to update a column in one table by a column in another table.

EDITING COMMANDS

- ED For use on non-table pages. Allows you to type more or less normally. Also has 32 graphic symbols available.
- PF Put Function. Allows you to put a 30-character instruction, made up of PIPS commands, into a function key.

GF	Get Function. Allows you to call a function key without actually pressing it (for instance, in an autoprogram).
SB	Swap Buffers. Moves the contents of the master buffer to the sub buffer, and those of the sub buffer to the master buffer.
BS	Buffer Sub. Displays sub buffer on the screen.
BSn	Shows a part of the sub buffer beginning with a given line number.
TRS	Transfer Sub. Transfers the contents of the master buffer to the sub buffer.
TRM	Transfer Master. Performs the reverse operation.

DATA SEARCH COMMANDS

CA-A	Explained above.
SFN	Links pages into a group which can then be called as a whole in various commands.
SORT	Alphabetizes and enumerates data, changing its order.
CS	Conditional Search. Pulls a smaller data set from a larger one according to user-set criteria.

CALCULATION COMMANDS

CAL	Calculation mode. Most of the math in PIPS is done in this mode.
CAL/C	Calculate/Continue. Leaves the mode on until released.
FO	Figure Buffer Operation. Allows easy transfer of columns of data between master buffer and figure buffer (which provides extra, off-table calculation space).
D	Decimal. Sets how many decimals of calculation results are to be displayed.
ACM	Accumulate. Calculates a running total of a column.
MAR	Matrix Read. Picks up the figure in a selected "cell" of a table.
MAW	Matrix Write. Lays down the read figure in a cell of another table.
BF	Display Figure Buffer.

The above table lists all the main commands used for PIPS. As you can see, there are about 60. Your first reac-

tion is probably, "Do you mean I have to memorize all of them?" Let me reassure you: it's not hard. In fact, about 20 main commands are all you really need to get started using PIPS.

Most of the PIPS commands, which you don't need at this point, are for later when you are ready to carry out highly complex, demanding operations.

Computers require painstaking programming. Figuratively speaking, to insert a screw with a screwdriver using the average program written under BASIC, you would have to write something like this:

```
hold screw in left hand
move screw to hole
insert screw into hole
hold screwdriver in right hand
set tip of screwdriver into groove on head of screw
turn screwdriver
```

In comparison, with PIPS all you would have to say is "Put in a screw." And that's it!

Now to give you an idea of how powerful PIPS really is, let's look at a sample of a program from a certain microcomputer hobby magazine.

The program is designed to give a comparison of sales figures for the past two years, showing (1) market share; (2) net growth; and (3) growth rate.

Under BASIC, it takes 163 lines to create the entire program. This is a huge volume, and for a novice or someone like myself (I'm already over 40 years old), this is murder!

How would we obtain the same information using PIPS? Well, we would need—only seven lines! Simply incredible! Unbelievable! Fantastic! (Excuse me for getting excited—sometimes I let myself get carried away.)

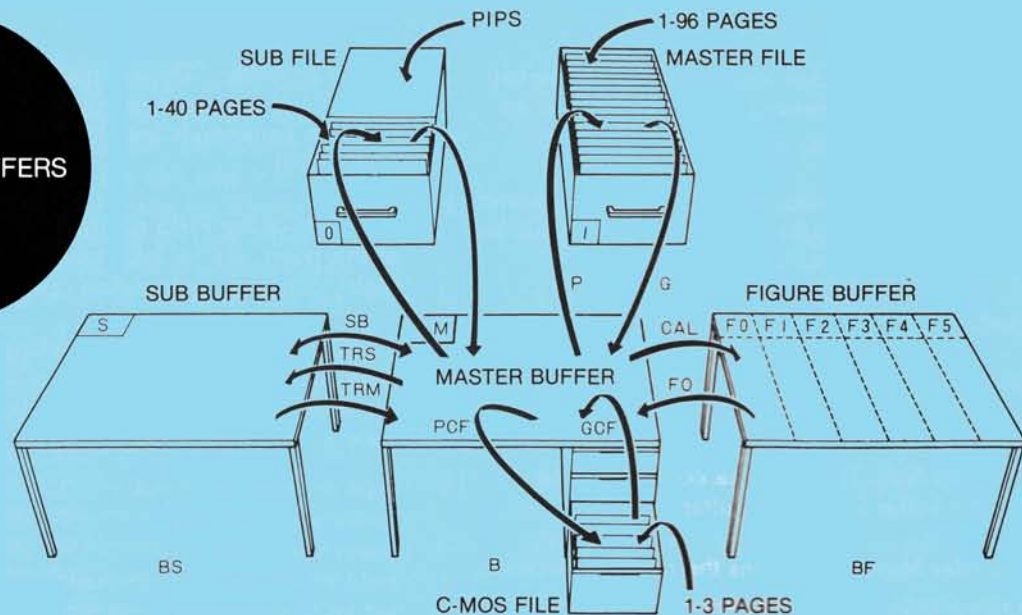
What it boils down to is that PIPS helps free you from dependence on others, making self-reliance truly possible. And in the complex world of computers, this must stand as being thoroughly remarkable.

The Relationship Between Files and Buffers.

Now let's take a look at one of the most important aspects of operating PIPS: the relationship between files and buffers. If this is not learned correctly at the very beginning, later on the user will find himself missing out on many essential points.

Take a look at the accompanying figure. This is a simple illustration of how files and buffers interrelate to one another during PIPS operations. First, the files: hypothetically, we can consider these to be identical to a filing cabinet we might encounter in everyday work at the office.

FILES AND BUFFERS



Figuratively speaking, PIPS III is made up of three files. Two of these are the master file and sub file, which are stored on the floppy disk as we mentioned previously. The master file is where the data is stored. The sub file can be regarded as an auxiliary file which is used while work is in progress. The third file is what we call the C-MOS file. Unlike the other two, it is stored internally, in an IC (integrated circuit) in the computer.

Because the digital information only has to travel an infinitesimally short distance within the IC itself, entry and retrieval of data from this file is extremely fast. To give you an idea of its speed, let's consider this page. To write or store the data on this page, the master or sub file would require a processing time of from two to five seconds. The C-MOS file requires no waiting time at all; it would process the information almost instantaneously.

However, should the AC power to the computer be cut, the data in the C-MOS file would be lost. The data therefore only remains in the file as long as the computer is turned on, which is in most cases until the end of the workday. The advantage of a C-MOS file is therefore to enhance the computer's processing speed.

As shown in the graph, the C-MOS operates in a manner similar to the drawer in a desk. Think of it like this: when you arrive at the office in the morning, you would take the files that you plan to work on for that day, and place them on your own desk. Then there is no need to be constantly making trips back and forth between your desk and the filing cabinet. Of course at the end of the day you would return the files to the cabinet before leaving the office.

Think of the buffer as your work area. Explaining how the buffer relates to computer operation is a little more complex. In most cases the purpose of a buffer is to provide a link between two separate computer devices, or

between a device and a human operator. It is a kind of intermediate damper, or shock-absorber, if you will.

For PIPS, the buffer can be thought of as a "place" where data is processed in between files and the outside. Another way of looking at it is to think back to your work desk. You remove necessary files from your file cabinet, and then spread them out on top of the desk. The buffer can be viewed as the spreading out of data.

Now try to imagine your desk as if it were divided into three segments. First, in the center of the desk is your "master file." Most of the work you do is at the desk, and depending on what the work entails, it is probably necessary to arrange and expand a number of documents. To make that job somewhat easier, there is also an area on the desk for a "sub file."

Let's say that in the course of your work, some sort of calculation has to be carried out. Even after getting your results, you might also want to retain the calculations temporarily, just like you would on a scratchpad. And you need another place on the desk for this, which we can call your "figure buffer."

Unfortunately, our desk—the buffer—has one disadvantage; due to the limitations of the CRT display, it is not possible to view all three buffers simultaneously. So one thing you have to do is learn to perform the equivalent of changing the channels on a TV. This is where the various editing commands come in. If you want to see what is in the master buffer, type B and RETURN. (You can think of the RETURN key as the command EXECUTE.) To see the sub buffer, type B S (buffer-sub). To check figures, type B F (buffer-figure, followed by RETURN. The list of Editing commands contains various other commands for swapping and transferring between buffers. They're not complicated—remember it's really no different from your desk. This is just one of the ways in which PIPS is "natural" and reflects the office procedures you have been following for years.

SORD'S COMMUNICATIONS PRODUCT LINE

by Kohji Ohta SORD Customer Services

SOCS WORD PROCESSOR INTERNATIONAL COMMUNICATIONS AT YOUR FINGERTIPS

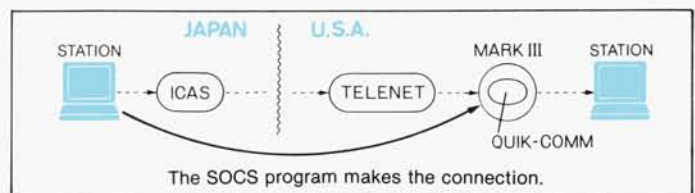
Up to now, international business has relied mainly on three forms of communication: international telephone calls, telexes, and fax systems. Each system, however, has disadvantages which sometimes outweigh the advantages. Take the telephone, for example. While it is nice for "personal" contact, it is expensive and you can never be sure that the person you want to call will be there (unless a person-to-person call is made, which ups the bill even more). Telexes and fax systems solve the problem of price, but in this day and age, who wants a hard copy? Hard copies are fine if the information received is to be read and filed away. But if the information must be processed in any way, it means re-keying the information into a computer. Optical Character Readers could solve the problem, but at present they are still too expensive for the everyday personal computer user.

Solving these disadvantages and bringing new advantages to international communications is the "electronic mailbox" system.

What is an "electronic mailbox?"

Although the phrase sounds rather electronically mystifying, the actual concept is not very difficult to grasp. In essence, an electronic mailbox works the same way as an ordinary post office mailbox. Once a mailbox is established for you at the post office, messages are placed in

this mailbox and wait to be picked up at your earliest convenience. Electronic mail works in the same way except that the mail is stored in a host computer instead of at the post office. Another difference is that you don't have to drive down to the post office to get your letters. Retrieving your mail is as simple as loading the SOCS (SORD Overseas Communications System) WP software into your M23 or M68, hooking up your SORD computer to a modem or acoustic coupler, and dialling a phone number.



Before this electronic mail system can work for you, however, your mailbox must be reserved in a HOST computer. There are a variety of host computers which offer mailbox services, but the one that the SOCS program works with is General Electric's Mark III QUIK-COMM program. This mailbox service covers over forty countries and offers not only storage and editing capabilities but record keeping features as well. There's no need to date or number your messages as QUIK-COMM automatically attaches the date and time to your message and assigns it a serial number which makes reference at a later date simple and convenient.

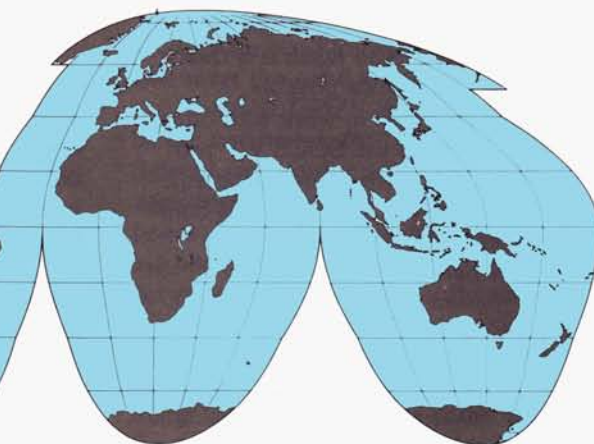
TELENET ETC. COUNTRIES

- Australia
- Austria
- Canada
- Denmark
- England
- Finland
- France
- Holland
- Ireland
- Mexico
- Norway
- Puerto Rico
- Saudi Arabia
- Sweden
- U.S.A.
- West Germany



MARK III NETWORK COUNTRIES

- Argentina
- Bahrain
- Belgium
- Bermuda
- Chile
- Dominica
- Hong Kong
- Israel
- Italy
- Kuwait
- Luxembourg
- New Zealand
- Philippines
- Portugal
- Singapore
- Spain
- Taiwan
- United Arab Emirates
- Switzerland



The message you send can be accessed from any of forty different countries. This points to another big advantage of using the QUIK-COMM service. With a telex or fax, your message is sent to only one place at a time. Imagine having to send the same message to forty different countries! With a telex or fax, it means sending the same tape or piece of paper through the machine forty times!! And with the speed of telex and fax systems as they are, that task could easily take an hour or more. Add line busy situations and interruptions to let other people send "just a short little message" (after all, there's usually only one telex or fax in a company and it's a rare day when anyone can monopolize those machines for more than just a few minutes) and you've got a simple task that could just cost someone the better part of a day!! With the QUIK-COMM program, it's just a simple matter of using a GROUP command to send the same message to a whole list of people.

Have you ever wondered what messages lie waiting for you at the home office while you are away on a business trip? Or have you ever wanted to spend most of the day at home but still keep up on what's happening at the office? It's all possible with an M23P, an acoustic coupler, and the SOCS Word Processor software.

What is the SOCS WP software?

Although QUIK-COMM is a good program, it can be made even better through the use of our SOCS WP program. Ordinarily, characters entered from the keyboard are sent directly to the Mark III HOST computer. If mistakes are made while entering data, they must be corrected using the QUIK-COMM editing functions. Using these editing functions, however, means not only mastering one more piece of software but also an increase in on-line costs since there is a charge for each editing function used. The SOCS WP uses the SORD WP off-line to create and edit a document. When the document is finished errorfree, it is sent to the HOST computer using the SOCS telecommunication function. The result is a clean document transmitted at the lowest price possible.

The SOCS WP can also be made to work with other mailbox systems, too. Included with the SOCS WP program is the BASIC telecommunications source program. Anyone familiar with BASIC and our asynchronous communications TBASIC subroutines can easily tailor the package to fit other mailbox networks.

PCOMM PIPS COMMUNICATION THE EASY WAY

Leave it to the Australians to come up with a nice neat way to handle PIPS communication. Mitsui Computer Systems, of Sydney, Australia, has developed a PCOMM program which can transfer/receive PIPS pages

to/from another terminal without the user ever feeling he is leaving the PIPS system and entering the communications mode. This not only banishes the feeling that the user is entering something "alien" but also allows him to look at the PIPS pages for confirmation first before actually sending the data.

Below is an example of what the screen might look like after entering the "DK #PCOMM" command from the Select Command mode of PIPS.

```

M0000....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*
1 : (PCOMM MASTER BUFFER) ( 0. 0. 0) F7
2 :
3 : ASYNCHRONOUS COMMUNICATIONS
4 :
5 :
6 : comment : PCOMM Program Revision 00B (c) S.FLYNN 1983
7 : f30,30,30,30,29,
8 :
9 :
10 :
11 :
12 :
13 :
14 :
15 :
16 :
17 :
18 :
19 :
20 :
    * Select Command = 0
60:7 * Select Command =
    0 S R T MT G P D L CM BT CON AUI PIPS

```

At the bottom of the familiar PIPS page are the commands that can be used in the PCOMM program. As can be seen, most of the commands ("0," "MT," "G," "P," etc.) are the same as those used in the ordinary PIPS system. The "S" and "R" commands are for sending or receiving PIPS pages.

The "T" command stands for terminal mode and is one of the commands that really makes this program special. With this command, the user can not only enter terminal mode (the mode in which one terminal is directly communicating with another) at any time, but can also escape from this mode at the simple touch of a button. Being able to go "offline" at will enables the user to look at his PIPS directory, get pages into the buffer, or do any other preparatory work even in the middle of a communication session. Being able to switch back and forth from terminal mode to PIPS command line in a matter of seconds has proved to be quite a time saver.

The "D" command is for scrolling around the buffer.

The "CON" command is another attractive feature. This command is for setting up the configuration necessary for communications (Code, Parity Check, Stop Bits, etc.). Usually these parameters need only be set once. How-

ever, with the present Time Sharing System with which this PCOMM is being used, it is necessary to set and re-set one of the parameters, XON/XOFF, during the communication session. The convenience of being able to set the configuration from the command mode can be appreciated if you can imagine a situation in which you have two printers (for example, a daisy wheel and a dot matrix). Imagine also that their baud rates are different. Some of the pages can be printed with one printer but others must be done with the other. You can see the situation where a few pages are printed, PIPS is ENDED, the baud rate is changed, PIPS is started up again, a few pages are printed, PIPS is ENDED, the baud rate is changed, etc. Now, imagine that you could change the baud rate with a SET command and you will realize how valuable it is to be able to change the configuration from the command line.

The last and most important command of the PCOMM program is the "AU #" command. Like the ordinary "AU #" command, it allows the user to string a line of commands together. Also included in the AUTO program are special commands for communications control; DELAY, RETRY, etc. Wild cards can also be used to check incoming data. With wild cards for data check and communication control commands, it is easy to make programs for automatic logon, transmission of data, and logoff. An auto-dial function is also planned for the future which will make the transmission of PIPS pages completely automatic.

CONTACT: STEVE FLYNN
MITSUI COMPUTER SYSTEMS,
AUSTRALASIA
P.O. BOX 234
FRENCH'S FOREST NSW 2086
AUSTRALIA

P-NET SYSTEM PIPS LOCAL AREA NETWORK

Ever wish you didn't have to yell across the room to someone to press the space bar every time you need to send data from one machine to another with the PIPS TD command? Ever dream of a low cost PIPS network?

If those are your worries and wishes, the P-NET system was made for you. The P-NET system utilizes two programs (a file server and a terminal simulator) along with the S-NET hardware to connect SORD machines into a PIPS Local Area Network. The network can consist of a simple connection between two machines to a total system of 64 machines.

The File Server Program

The file server program is used to turn a SORD machine into a central PIPS HOST (a file cabinet from which other terminals can get or put PIPS pages). This file server

program is equipped with security features so that only terminals which are authorized to do so may access the pages in the file cabinet. Authorized terminals can also be limited as to which pages they may access for reading and updating. A certain terminal may, for example, be able to look at any page in the central file but only be allowed to put certain pages from its own file. These security measures can not only keep certain pages confidential but can guard against mistakenly writing over valuable pages.

Once the file server program is started, no operator intervention is required. The HOST always sits ready waiting for access commands from other terminals and responds to those commands automatically.

The Terminal Simulator Program

The terminal simulator program is used to send/receive pages to/from the HOST. This program can be started from the Select Command mode of PIPS-III. The commands used are "CN" and "DN" for logging onto and off the HOST computer, "RX" and "TX" for receiving/transferring pages, "XB" to look at the HOST title file, "MX" for message exchange between the host and terminal, "G" for getting a PIPS page onto the display for confirmation, "RPI" to return to PIPS, and "END" for ending the operation altogether.

M243/M243EX PDAT FILE SERVER SNET PIPS-III

The PDAT FILE SERVER and SNET PIPS-III system is a more sophisticated version of the P-NET system providing greater security and ease of use.

The management functions of the PDAT file server make it the real master keeper of a PIPS hard disk. Besides having functions to set passwords and authorize terminals for access, the HOST has the capability to delete protections should the terminal forget its own password. There is also a function to backup the hard disk onto floppies whenever the need arises.

The SNET PIPS-III program offers convenience as all the commands for SNET use can be entered from the Select Command mode. Unlike the P-NET system, the user does not have to exit from PIPS-III to send or receive pages. Sending or receiving pages is done simply by prefixing page numbers with an H, which stands for HOST. To get page one from HOST, the user just enters the command "G;H1."

SNET PIPS-III also provides a page LOCK function to prevent confusion arising from two or more terminals trying to update the same page. By locking certain pages, the terminal which is accessing those pages can ensure that no other terminal will access and update the pages while it is in the middle of processing them. File sharing is carried out systematically and in a worry-free fashion.



SORD M23/M23P WORD PROCESSING AUDIO TRAINING PACKAGE

As everyone, knows, employee training can be a tremendous job requiring either the valuable hours of another employee who acts as the trainer or time away from the office while the employee attends a training course. To cut down on this loss of time, Mitsui Computer Systems of Sydney, Australia, has developed an AUDIO TRAINING PACKAGE. This splendid package includes a manual in a presentation binder, audio cassette tapes, WP diskettes, and even transparent labels which can be stuck onto, on the side of, or above the function keys and ten keys to make learning the WP key positions quick and easy.

The presentation binder itself should make any typist's eye light up. As all typists know, it is no easy task to learn from or type the exercises out of a manual which can't be stood upright. It is, in fact, a cursing experience. With a presentation binder, reading and typing is a breeze as it is so easy to move one's eyes from the manual to the display. (I wish I had this when I was learning how to use the WP!!)

The diskettes are also special: the WP has been developed in Australia so that some functions can be done more easily than the present WP which is supplied by SORD. Underline, for example, is done using the Glossary function instead of the cumbersome zero format line.

The cassette tape is, of course, the best part of the package. The voice on the tape is smooth and friendly.

The text starts from the very basics of the computer, explaining the system and how to take care of diskettes and proceeds to give very good lessons not only in how to operate the computer (press this button, press that button) but also the basics of word processing (manipulating text, making the job easier, etc.).

Other software produced in Australia includes an M23 Color Screen Editor which makes the production of exciting full color blinking screens a real snap. Moving around the screen is just a matter of pressing cursor arrows and colors. Blinking and other functions can be enabled/disabled just by pressing escape and another key. This screen editor really makes work fun.

Color games have also been developed. The color games come as a set of three volumes that range from animation-type games to word games to strategy games. Not being an arcade person myself, I rather enjoy the word and strategy games, especially when I am looking for another way to do something and need a diversion so that I can start afresh. The most interesting game is the "problem solver." A session with this computer psychiatrist can really help to get rid of what bugs me.

CONTACT: MARIA BUSUTTIL
MITSUI COMPUTER SYSTEMS,
AUSTRALASIA
P.O. BOX 234
FRENCH'S FOREST NSW 2086
AUSTRALIA

To subscribe to SORD WORLD follow the table below to determine your rate.
All rates are based on one year (six issues) subscriptions. Sorry, but no refunds
are allowed.

ASIA (Including Korea, South East Asia, and the Philippines) = \$28.00

NORTH AMERICA = \$30.00

EUROPE (Other) = \$31.00

**All rates are in US dollars. Please send check or international
money order with the card below to:**

**PIPS WORLD, INC.
Naigai Chizu Bldg.
3-22, Kanda Ogawa-machi
Chiyoda-ku, Tokyo 101
JAPAN
Tel.: (03) 295-6713**

Japan Price = ¥750

Postage = ¥200

Yearly Subscription (six issues) = ¥4,500 (Postage Included)

**For readers in Japan payment should be made by sending cash
by registered letter to the above address or by mail transfer to**

Tokyo 0-83552 (Postal Acct. #.)

PIPS WORLD, INC.

PLEASE send me one year (six issues) of SORD WORLD.

NAME:

ADDRESS:

CITY:

STATE:

COUNTRY:

POSTAL CODE:

No.1 (Oct. '83),

No.2 (Dec. '83)

No.3 (Feb. '84),

No.4 (Apr. '84)

FROM WHICH ISSUE?

www.sord.co.nz

The SORD M68 Multi-Smart Mega System

Does tomorrow's expansion dilemma have you—and your company—worried? The M68 was designed today to meet the fast paced demands of tomorrow's office automation.

It not only holds its own as a stand-alone, but can serve as a satellite for a larger mainframe.

The SORD M68

gives you two CPUs instead of one.

The M68000 microprocessor supports the 16-bit mode; the Z80A microprocessor is your 8-bit computer. One CPU backs up the other.

The SORD M68 opens up the 16-bit world—and preserves your 8-bit software investment.

It has 1MB of RAM expandable to 4 mega-byte and a choice of four

operating systems: CP/M-80*, CP/M-68K*, UCSD p-System* or KDOS.

Automate your office with small-computer software, big-computer power.

Run PIPS, SORD's information management system, and topflight SORD word processing—or run the vast software library available on the CP/M-80* (8-bit) or CP/M-68K* (16-bit) operating systems. And with S-NET, SORD's economical networking system,

you can link M68s at a fraction of the price of other networks. **Download from your present mainframe for easy development programming.**

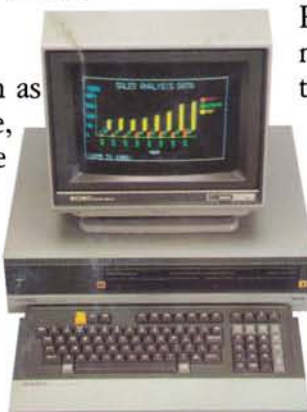
The M68 can stand in as an intelligent terminal to

your company's full-scale system, running mainframe-standard FORTRAN, BASIC, PASCAL, COBOL and C (depending on the mode and operating system selected).

The M68 is truly affordable—costing less than many smaller machines. Because at SORD, economy is also part of the future we're planning.

SORD welcomes all OEM questions

* CP/M-80 and CP/M-68K are trademarks of Digital Research Inc. UCSD p-System is a trademark of the Regents of the University of California.



Computer innovator **SORD** SORD COMPUTER CORPORATION

HEAD OFFICE: Saito Bldg. 2F., 14-6, Kyobashi 3 Chome, Chuo-ku, Tokyo 104, Japan. Telephone: (03) 562-6061. Telex: 2522745 SORDIN J

www.sord.co.nz