Microcomputer Power

$16.95

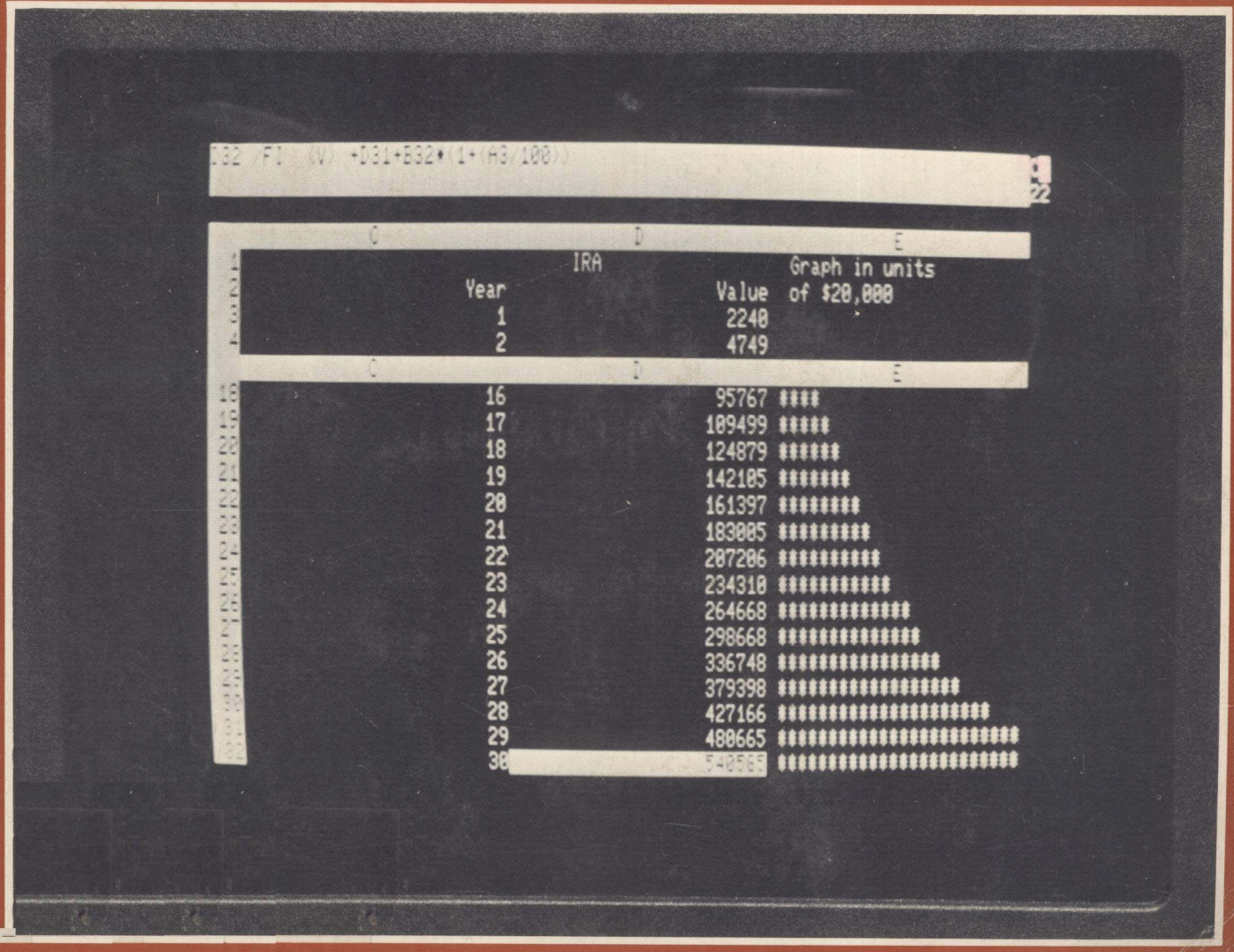
Easy to follow, step-by-step instructions
Use the VisiCalc worksheets to

- calculate sales projections
- evaluate your portfolio
- log your expenses
- compute your IRA account
- determine your net worth

Written for the novice and the professional.

VisiCalc®
for the TRS-80™ Model II and Model 16 Computers

Edouard J. Desautels



VisiCalc®
for the TRS-80™ Model II and Model 16 Computers

Edouard J. Desautels
University of Wisconsin
Madison

wcb
Microcomputer Power Series
Wm. C. Brown Company Publishers
Dubuque, Iowa

Library of Congress Catalog Card Number: 82–71102

ISBN 0–697–09955–5

Printed in the United States of America

Preface

VisiCalc™ makes a computer as easy to use as a calculator, and it gives you far more power than a calculator does. This little book is intended to show you how you can use VisiCalc effectively. This book is designed so you do not have to have any previous experience with computers to use VisiCalc.

The first chapter sketches the kind of situation in which VisiCalc excels. The next chapter shows you how to fire up the TRS-80 Model II computer. Then we start looking at the specifics of using VisiCalc. Although it is easier to keep reading the chapters in the order in which they appear, you can often skip ahead if you wish to look into some feature of special interest to you. A comprehensive index will help you find your way.

Most of the features of VisiCalc are presented while solving a sequence of realistic problems, such as evaluating the accumulation in an IRA (Individual Retirement Account). Some of these problems are solved a second time, so you can better appreciate the contrast between different approaches.

Almost everyone has to come to grips with numbers, lots of them. VisiCalc is such a powerful assistant in helping you manage numeric information that you should seriously consider equipping yourself with this tool. Perhaps this book will have made it easier for you to make such a decision.

Although this book contains detailed instructions on how you can yourself do everything that is shown, an optional diskette may be used to reduce the typing which would otherwise be required. See the appendix for further information.

Note: the latest model in the TRS-80 family of computers is the Model 16. It is upward-compatible with the Model II, so this book can also be used with the Model 16.

Contents

Chapter 1

WHY USE VISICALC?

Electronic computers have been in use for over 30 years. Why is it that VisiCalc is one of the best-selling computer software packages since the beginning of the computer age? The Time magazine article on VisiCalc gives you some idea of its financial success. What makes VisiCalc so attractive?

The Smash Hit of Software

Daniel Bricklin, 29, and Robert Frankston, 31, a team of new-wave composers, have penned a dynamite disc that has grossed an estimated $8 million. It is not a punk-rock smash, but an unmelodic magnetic number called VisiCalc, the bestselling microcomputer program for business uses. The featherweight sliver of plastic is about the size of a greeting card, but when it is placed in a computer, the machine comes alive. A computer without a program, or "software," is like a $3,000 stereo set without any records or tapes.

Three years ago, Bricklin, then a first-year Harvard Business School student, conceived VisiCalc while struggling with financial-planning problems on his calculator. He enlisted the aid of Frankston, a longtime friend and an expert programmer, to develop a new piece of computer software that would make juggling all those figures easier.

The partnership paid off. Since late 1979 nearly 100,000 copies of nine different versions of VisiCalc have been ordered at prices ranging from $100 to $300. It is far ahead of other business programs like Data Factory and General Ledger, and even outsells the programs for Star Cruiser, Dogfight and other arcade-like computer games.

VisiCalc translates simple commands typed on a keyboard into computer language that the machine then uses to solve problems. It enables a businessman, for example, to manipulate labyrinthine equations to calculate financial trends for his company. If he changes one figure, the machine can tell quickly how that affects the other numbers. A firm that gives its workers a 10% pay hike could estimate how that action would alter its costs, sales, profits, or dividends.

The computer program is being put to a wide range of uses. It helps Allerton Cushman Jr., a New York financial analyst, to project insurance-industry profits during the week and tote up his income taxes on the weekend. The Cabot Street Cinema Theatre in Beverly, Mass., bought VisiCalc to figure out which pattern of movie show times draws the best box-office receipts. An accounting firm in Las Vegas plans to use VisiCalc to tell its gambling-house clients how to position slot machines around the floor to ensure the biggest take. VisiCalc is obviously one composition that is in no danger of fading from the charts.

There can be no doubt that any solution that VisiCalc can produce can also be produced by writing a computer program to generate that solution. For any problem you might wish to solve using VisiCalc, a computer program to solve that problem could be written, using some computer programming language such as BASIC, FORTRAN, COBOL, PASCAL, etc.

That is exactly the reason for using VisiCalc. You <u>don't</u> have to write a program to solve your problem when you use

VisiCalc. You key in the relevant data (there is no way to avoid this completely, no matter what you are using). You then specify how the data items are related, and what answers you want calculated. At this point, the computer expert might object: "This is the same as writing a program". Theoretically that may be true; in practice, it makes all the difference in the world. Writing computer programs can be very tricky and time-consuming, and that is after you have invested your time and effort in learning the programming language.

When you solve a new problem using a computer, you usually have to write a program to solve that problem, using one of the programming languages we mentioned earlier. That usually means you also have to use some kind of editing program just to prepare the program you need. Then you may have to prepare a data file, with the help of the editing program.

The data file is considered input to the program you wrote, and the results produced by your program are called its output. We usually depict the input as flowing into the program which has been loaded into the computer's memory, and show the results produced by your program as flowing out from it, as we see in figure 1.1.

Figure 1.1 Input, Program and Output Relationships

After you examine the output from your program, you may come to one of several conclusions:

(1) great- let's stop computing.
(2) oops!- there must be a "bug" in the program; try to find it and fix it.
(3) it's ok, but what if ...?

Case 3 might involve wondering what would happen if some data item had a slightly larger value, or what would happen if the formula in the program was just a little different.

With VisiCalc, you are more likely to get the first conclusion first. Number two is much less probable, simply because VisiCalc won't let you request many ridiculous computations. VisiCalc really shines in the third situation. You simply change the desired number or formula, and you immediately see the consequences. You don't have to fool with an editing program, or even request that your program be rerun.

With the conventional approach, the input data is laid out for the convenience of the computer program (or whoever wrote the program). The actual layout of the input data has little if any spatial relationship to the results. With VisiCalc, you begin by putting the numbers where they should be at all times. And you

decide where the results should appear, in relationship to the other input. If you should change your mind, it is a simple matter to move things around painlessly. If you then decide to change a number, you simply locate its old value where you expect to find it (not at some strange location chosen for a program's convenience). As soon as you change that number, all the other outputs which depend on that number immediately change.

A Brief Example

The following simple example will give you a better idea of the difference between solving a problem using VisiCalc, versus solving a problem using the conventional computer approach.

Suppose you were a budding author, and had just had your manuscript accepted by a publisher. The publisher might have proposed that you be paid royalties based upon the following sliding schedule:

Earn 5% of selling price, for the first 3,000 copies.
Earn 7% " " next 4,000 "
Earn 8% " " next 5,000 "
Earn 10% " " , for all additional copies.

With VisiCalc, you would lay out the essential data in the form of a table, as we see here in table 1.1.

Rate	Copies
5	3,000
7	4,000
8	5,000
10	?

Table 1.1 Raw Data

You would then make an educated guess as to what number to use for the ? in table 1.1. You might like to project your potential income (before taxes) if the book sold say 50,000 copies, assuming that it sells for $1 per copy. So the ? is replaced by 50,000-(3,000+4,000+5,000), giving us 38,000. You then ask VisiCalc to work out (Rate multiplied by Copies), for each line, which results in table 1.2, since our rate figures are actually percentages.

Rate	Copies	Income
5	3,000	150
7	4,000	280
8	5,000	400
10	38,000	3,800

Table 1.2 Projected Income

Of course, what you really want is a running total, so you ask VisiCalc to produce another column, headed "Total", which is to show, for each row, the sum of the current and all preceding

income figures. That being done, you now see the results in table 1.3.

Rate	Copies	Income	Total
5	3,000	150	150
7	4,000	280	430
8	5,000	400	830
10	38,000	3,800	4,630

Table 1.3 Projected Total Income

This may seem like much ado about nothing, at this point. You could have done the same thing with a calculator, or merely with paper and pencil. True--but having gone this far, you can now begin the "What If ..." phase. As the budding author, you might want to negotiate a better royalty schedule.

So you begin thinking "What if my first book is not a terrific success? I should ask for a higher percentage for the first few thousand sold, just to be safe." Suppose you settled on 6, 8 and 9%, in place of the 5, 7 and 8 that was offered; you don't want to be greedy. You can now proceed to replace the 5, 7 and 8 by the 6, 8 and 9, and lo and behold, you immediately see the consequences, as shown in table 1.4.

Rate	Copies	Income	Total
6	3,000	180	180
8	4,000	320	500
9	5,000	450	950
10	38,000	3,800	4,750

Table 1.4 Projected Total Using New Rates

You might think about these totals for a while, and because you are now convinced you have a best seller on your hands, perhaps it would be better to focus on negotiating a better top rate than the 10% that was offered. After all, the other rates will make very little difference if you sell 100,000 copies. So you change the 10% to 20% (dreamer) and you decide to increase the 38,000 by 50,000. The results are shown in table 1.5.

Rate	Copies	Income	Total
6	3,000	180	180
8	4,000	320	500
9	5,000	450	950
20	88,000	17,600	18,550

Table 1.5 New Projected Totals

If you were in the publisher's shoes, you would also like to be able to do the same juggling of figures. As the budding author, you have much less experience with the consequences of manipulating either the rates, the sales thresholds, or the number of steps in the schedule. The publishers can probably do it in their heads; you could use a little help.

You could of course have written a computer program to perform these calculations. The data file for this program could very well look like the following list (using the first set of rates and copies):

```
5,3000,7,4000,8,5000,10,38000
```

which is awkward to read and change.
It might have looked like:

```
05 3000      or        05070810
07 4000                03000 04000 05000 38000
08 5000
1038000
```

where you find yourself putting in leading zeroes or leading blanks with the data, because the exact spacing between these numbers may be critical, depending upon the programming language that was used in writing the program. You could easily provide data which looks correct to the naked eye, but which the program takes to be ten times larger (or ten times smaller) than what you had in mind.

The data file

```
5          3000
7         4000
8          5000
10        38000
```

seems to have the right numbers, even though the alignment for the 4000 and the 10 is a little sloppy. Some computer programs would interpret this 4000 as if you had written 40,000 because the actual position of the number on the line was critical. This is much less of a problem with VisiCalc. You immediately see what VisiCalc thinks you just typed. If it is not what you intended, you can change it right away.

What About Big Problems?

Suppose you had a really big problem to solve? Could VisiCalc handle it? Suppose you were going to be a real-estate tycoon, and were working out a ten-year projected statement of cash flow. Such a statement is a table with ten columns of numeric data, and some 33 rows, depending upon how detailed you want to project your cash disbursements. Then we also need some labels to keep track of things. If each column is to hold numbers as large as eight digits, and we leave a little space between columns, for ease of reading, the cash flow table will be about 120 characters wide. Since most computer video display terminals (VDTs) can only display 24 or 25 rows of 80 characters at one

time, you would be hard pressed to squeeze all of the cash flow table onto the VDT screen at one time.

With VisiCalc, you can construct and display tables with as many as 63 columns and 254 rows, and you can "browse" over parts of the table very easily. If the whole table won't fit on your screen, VisiCalc treats your VDT screen as if it were a "window". Your screen window lets you see any part of the table you wish to see, getting as much of it as will fit on your screen at one time. Trying to do this with a conventional computer program would be far more difficult. Figure 1.2 illustrates the idea of a window.

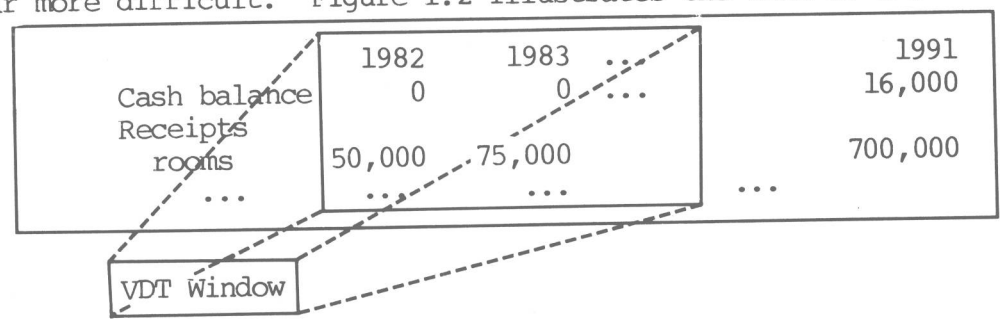


Figure 1.2 VDT Window into a large table

Summing Up

VisiCalc is data-oriented, very much like a calculator. You begin with your own raw data, lay it out on what amounts to an electronic worksheet, using your VDT's screen as an easily erasable worksheet. Then you begin specifying the relationships between your data and the desired results. You build up to the desired end-product in a step-by-step fashion, seeing the results at every step. You always see your input data in the natural spatial relationship it is intended to have with respect to any computed results.

Problem solving with computers, using the conventional approach, is program-oriented rather than data oriented. Most people who are not computer experts feel more at ease with the data they know well, than with the use of unfamiliar computer programming languages. Most people are familiar with the everyday use of a simple worksheet. VisiCalc combines the ease of using a calculator, and the familiarity of a worksheet, with the power of a computer. It follows that most people will find that VisiCalc provides a very natural, user-friendly way to make the computer work for you.

Coming Up.

Specifics on using VisiCalc: how do you tell VisiCalc to do something? What if you made a mistake? What if you were in the middle of filling out a worksheet and had to leave suddenly: how can you save worksheets, and recall them later? These and many

other features of VisiCalc will be described, discussed and illustrated as we proceed. It goes without saying that you will learn more, faster and better, if you can be using VisiCalc and trying the things we are discussing. We hope however to provide sufficiently detailed examples so that you can follow what is going on even if you don't have immediate access to a computer equipped with VisiCalc.

Chapter 2

GETTING ACQUAINTED WITH THE TRS-80 MODEL II COMPUTER

This chapter deals with the operation of the TRS-80 Model II computer. If you are already familiar with this computer, you might want to merely skim this chapter and proceed to the next. No previous experience with computers is required to use VisiCalc with the TRS-80. Here the rudiments of turning a computer on, and setting it up so you can use VisiCalc effectively will be described. Comments on the Model 16 appear at the end of this chapter.

Physical Components of a TRS-80 Model II

A TRS-80 Model II Computer System has two or more components:

(1) a display console with a diskette drive
(2) a keyboard
(3) optional devices, such as a printer

Figure 2.1 on the next page shows the display console and the keyboard, and it identifies important items. The right-hand part of the display console contains a floppy disk drive. This drive can hold one eight inch diskette. Each diskette can record approximately 500,000 characters of information. Some of these characters may be used to represent computer programs such as VisiCalc, and some of them used to represent your data. The word "byte" is often used in place of the word "character"; for our purposes, these words are equivalent.

You can add up to three more diskette drives on a Model II TRS-80 computer, giving you immediate access to some two million characters of information. You can instead add up to three "hard disks", each of which can hold some eight million characters. These disk drives not only have this much higher information storage capacity, they also let your TRS-80 process data much faster.

Operation of the TRS-80 Model II Computer

We will walk through the steps required to start computing, and we will discuss the purpose of these steps, in case you are wondering about what is going on.

1. Verify the floppy disk drive is empty, by pushing the Drive Release Bar, located on the right, between the power switch and the diskette drive door. Push this bar towards the right. The diskette drive door will then pop open, and if a diskette had inadvertently been left in the drive, it will be ejected. The computer should not have been left with a diskette in it. You

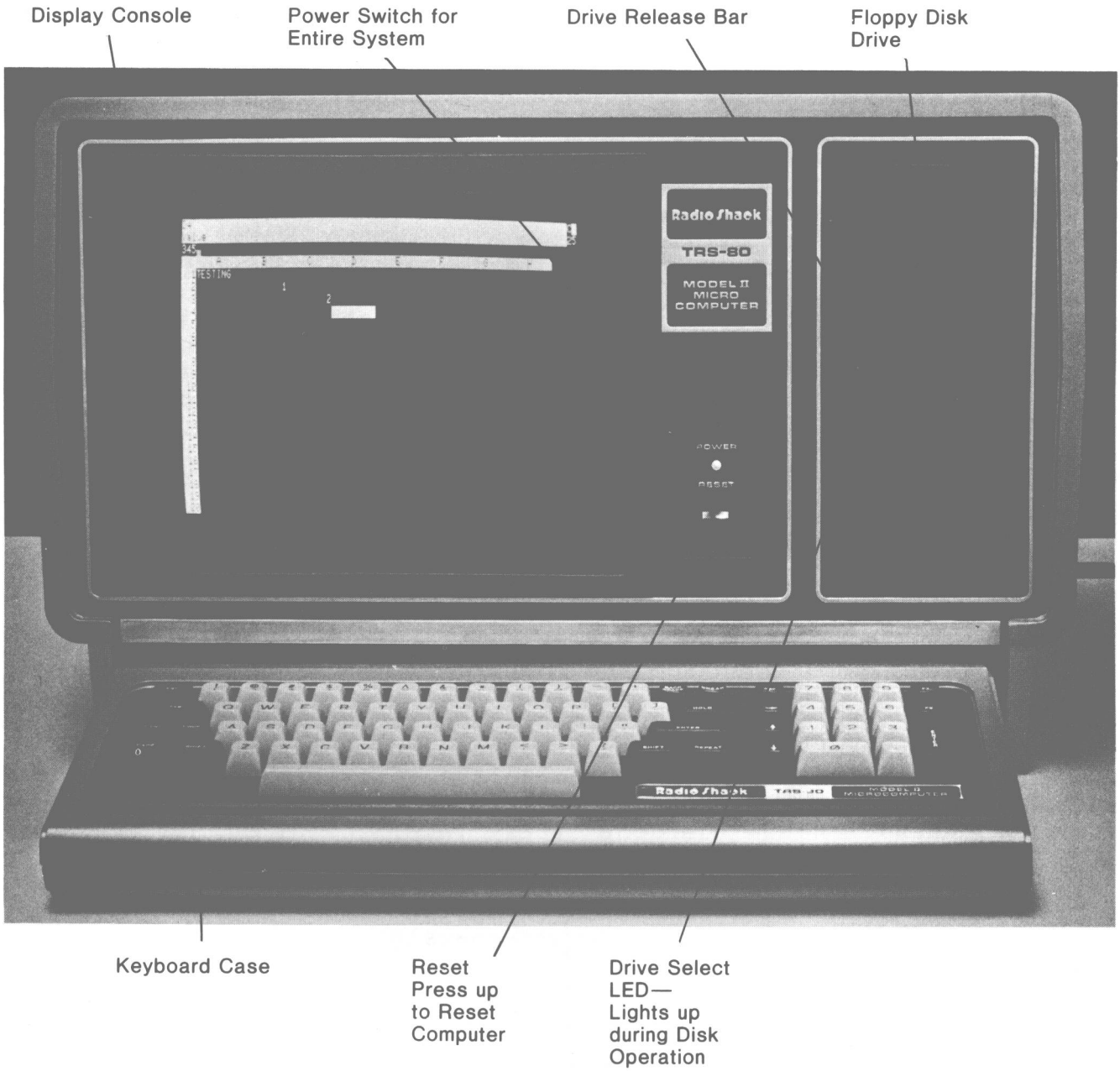


Figure 2.1 Physical Components of a TRS-80 Model II

should not power up or power down the computer while it has a diskette in it. Doing so could damage the drive, or the diskette, or both.

2. Turn the computer on, by pushing the switch labeled POWER up, towards the position engraved ON (barely visible, since it is in black, on a black background). If you have the disk expansion unit which adds one, two, or three extra disk drives, you must turn its power on, even if you do not plan to use any of the extra drives that day. Failure to turn on the disk expansion unit can lead to destroying the information on the diskette placed in the built-in drive, drive 0 (I can attest to this from sad personal experience). If you inadvertently find yourself in this predicament (diskette in drive 0, with its power on, but the expansion unit turned off), remove the diskette from drive zero before you turn on the expansion unit.

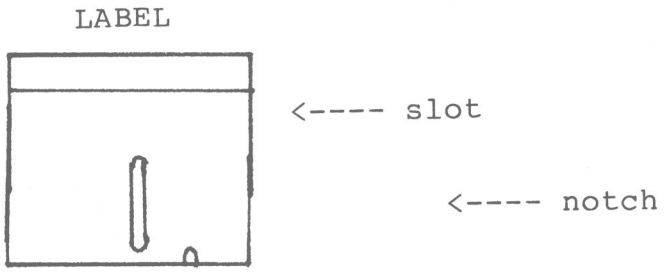
3. When you turn the power on, an indicator light just below the power switch should start to glow. If it does not, turn the power switch off and check that the power cable is properly secured at both ends (at the rear of the display console, and at the wall power outlet). Then repeat step 2.

4. Within a few seconds after you turn the power on, the console display will show the message

INSERT DISKETTE

If you do not see such a message, and the power-on light is glowing, then push the RESET switch which is just below the POWER switch. You should then see the "INSERT DISKETTE" message.

5. Find the diskette labeled VisiCalc. Remove it from its protective jacket. All diskettes have an oval slot in the rigid black square envelope which protects the flexible (floppy) disk magnetic recording media, as shown below:



If you see a small notch to the right of the oval slot, half-way between the slot and the edge of the diskette, then the computer will refrain from changing any information on that diskette; the diskette is said to be "write-protected".

If on the other hand you see a piece of foil in the same position covering the notch, the information on that diskette can

be changed, and new information added. In such a case, we can say that this diskette is "write-enabled".

If you are inserting your one and only VisiCalc diskette into the diskette drive, make sure its write-protect notch is exposed. That will prevent accidental overwriting of your master copy. If you are inserting a copy of your VisiCalc diskette, then you presumably want to be able to change things on it, add new data, etc., so cover its write-protect notch with the gummed foil that is provided. We will discuss how you can copy a diskette shortly.

When you insert the diskette into the diskette drive, the label side should be facing away from the console display; the label should be facing to the right. Gently push the diskette into the drive door slot until you feel it click into place. Never try to remove a diskette by pulling it out. Push the drive door shut (push the door's edge toward the left, until you hear it click). If you then decide to remove the diskette, push the Release Bar to the right, when the Drive Select light in the Release Bar is off.

6. The computer will sense that the diskette has been inserted after you shut the diskette drive door. The computer will then begin reading information from that diskette. Whenever the computer is actively using the diskette, the Drive Select light within the Drive Release Bar will glow. Never remove a diskette while this light is on.

7. Within a few seconds, your display screen should show the Tandy Corporation logo (a huge T over a huge C, followed by a request to enter the date:

KEY IN DATE MM/DD/YYYY

You must respond by typing the two digits for the current month (e.g. 09 for September), a slash, then the two digits for the day of the month (e.g. 29), another slash, then the four digits for the current year (e.g. 1982).

Figure 2.2 shows you the keyboard we will be working with. It is very much like a typewriter keyboard, with the addition of a numeric keypad on the right, plus a few extra keys that will prove to be very useful as we go on.

Figure 2.2 TRS-80 Model II Keyboard

You can correct an error while typing a response by pushing the BACKSPACE key one or more times, and retyping the correct information. When the date appears to be correct, push the ENTER key. Whenever you use the keyboard, you can type using either of the two ENTER keys, and you can use either set of digit keys (those at the top of the usual typewriter keyboard, or those on the numeric keypad at the right). So a typical keyin would leave the screen showing

KEY IN DATE MM/DD/YYYY 09/29/1982

You had to push the ENTER key to cause TRSDOS to accept your response, but as a general rule no special character will be displayed at the point you pushed the ENTER key. The computer will next ask you to enter the time:

INPUT THE TIME HH.MM.SS

Since VisiCalc does not pay any attention to the time of day (it does need the correct date), you can avoid responding by simply pushing the ENTER key. Otherwise type the hour in, using the 24-hour-day format (e.g. 1 P.M. is recorded as 13, 2 P.M. as 14, etc.). Follow the two digits for hours by a period, type two digits for the minutes, another period, and two more digits for the seconds. You can omit the seconds if you wish. For instance, you could enter the time 3:46 P.M. by typing

15.46 ENTER

The computer will then keep track of the correct time. (note: whenever you see the word ENTER included among other items to be typed, simply push the key labelled ENTER--do not type the 5 letters E, N, T, E, and R.)