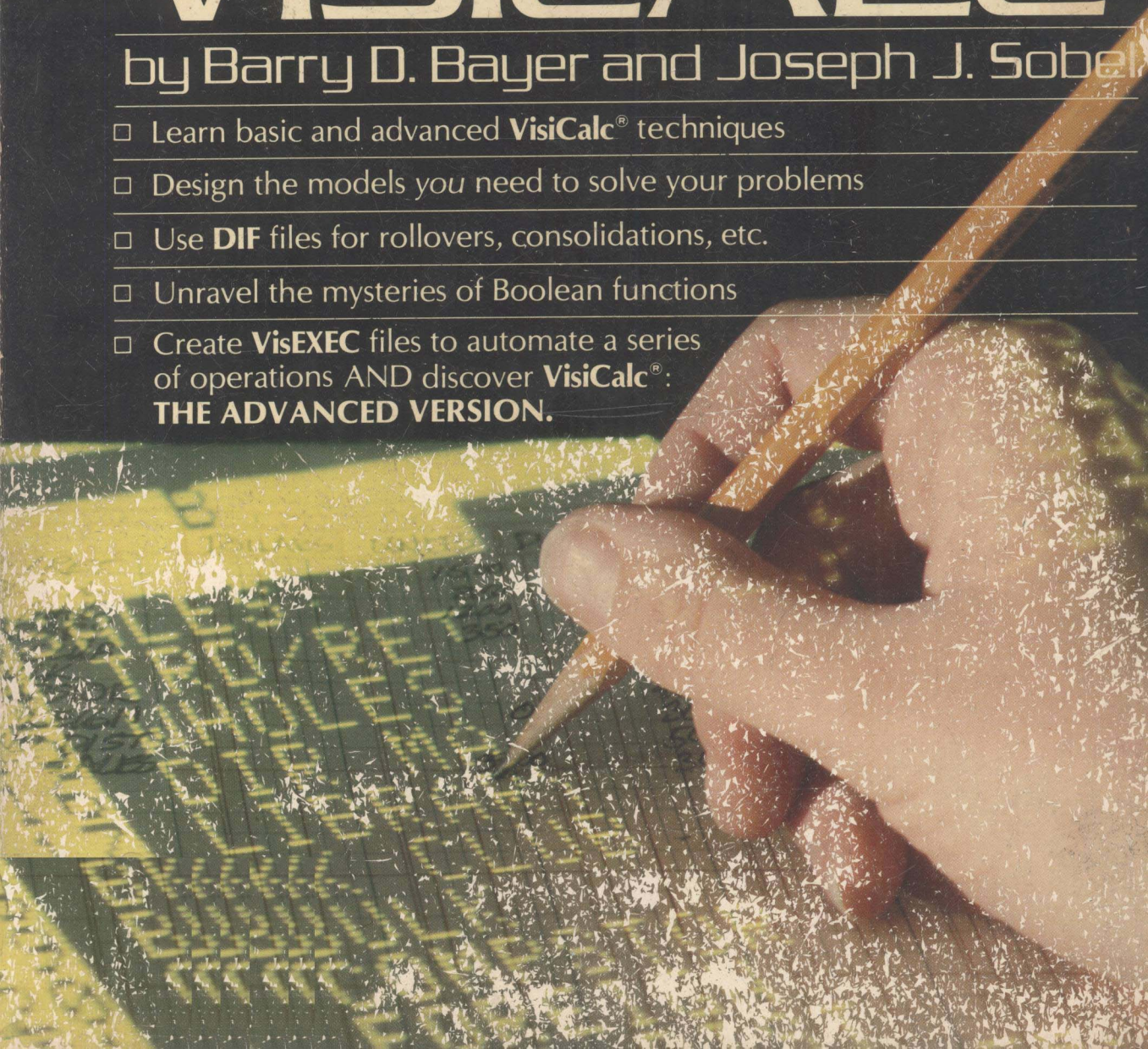


Dynamics of VISICALC[®]

by Barry D. Bayer and Joseph J. Sobel

- ☐ Learn basic and advanced **VisiCalc[®]** techniques
- ☐ Design the models *you* need to solve your problems
- ☐ Use **DIF** files for rollovers, consolidations, etc.
- ☐ Unravel the mysteries of Boolean functions
- ☐ Create **VisEXEC** files to automate a series of operations AND discover **VisiCalc[®]**:
THE ADVANCED VERSION.



DYNAMICS OF

VisiCalc®

Barry D. Bayer

Joseph J. Sobel

DOW JONES-IRWIN ☐ Homewood, Illinois 60430

The financial template discussed in Chapter 6
is available from the authors on disk format.
For further information contact:

Barry D. Bayer
P.O. Box 1506
Homewood, Illinois 60430

© DOW JONES-IRWIN, 1983

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.

From a Declaration of Principles jointly adopted by a Committee of the American Bar Association and a Committee of Publishers.

ISBN 0-87094-391-X

Library of Congress Catalog Card No. 82-73621

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0 K 0 9 8 7 6 5 4 3

DYNAMICS OF

VisiCalc®

PREFACE

In the beginning there was the Apple computer, a checkbook-balancing program, and a bunch of games. With 48K of RAM memory, and BASIC in ROM, the Apple II was a powerful computer, but no one except the computer professional or serious computer hobbist could make it do much. And the typical computer professional was convinced that the Apple was only a toy.

Two Harvard Business School students, Daniel Bricklin and Robert Frankston, knew better; they developed a program that could run on the Apple and which looked like one of their professor's blackboards. Thus was born, for want of a better name, VisiCalc, the VISible CALCulator. At first no one knew what to do with VisiCalc. But as word of the power of this electronic simulation of an accountant's worksheet spread, VisiCalc and the Apple II computer became the preeminent products in their respective fields, and the infant microcomputer industry leaped into the Fortune 500.

We have come to VisiCalc out of a need to use a computer to manipulate data from our respective professions, but without any interest in becoming computer programmers. VisiCalc has worked for us. We have written this book to help make VisiCalc work for you.

The *Dynamics of VisiCalc* was not designed as a cookbook of VisiCalc applications templates nor a tutorial in VisiCalc's basic command keystrokes. Although we have included several VisiCalc templates which you may enter into your computer, it is our hope that you will use this book to enhance your understanding of how to build your own templates to do exactly what you want, exactly the way you want it.

We have not been able to do this alone. The financial model template presented in Chapter 6 and Appendixes C and D was initially suggested by Allen G. Snider of Laventhol & Horwath's Boston office. Many of the other examples come from problems presented by our clients. Parts of this book have appeared in different form in *The Apple Orchard*, *Harvest*, the journal of the Northern Illinois Apple Users Group, and *Desktop Computing*

magazine, and we are indebted to the editors and readers of these publications for their comments and encouragement. We must also thank Jerrold H. Bents, Thomas A. Farin, and L. Darryl Mataya for introducing us to the VisEXEC concepts presented in Versacalc 16!, and the people at Software Arts, Inc., and VisiCorp for their assistance.

This book could not have been written without the cooperation of our families. Our wives Susan and Bette Jo put up with the time taken to plan, write, and revise. Our children, David, Elisabeth, Jonathan, Michael, Stephen, and William, reluctantly stopped using our computers long enough so that we could use them to complete our work.

This book must also be dedicated to Messrs. Bricklin, Frankston, Wozniak, and Jobs, without whom all of this would not even have been a dream.

Barry D. Bayer

Joseph J. Sobel

CONTENTS

PART I

Getting Started

Introduction	2
1. If You Don't Have a Computer	5
An Accountant's Worksheet. Replication. Functioning.	

PART II

Basic Techniques with VisiCalc

2. Basic Template Design	13
Starting Out: <i>VisiCalcers Beware. Save Your Work Often! Back Up Your Disks. Back Up Your Program Disk. Hints for the IBM and Apple User.</i> Planning Your Template: <i>Do It on Paper. Separate Your Input and Output.</i> How to Enter a Template: <i>Windows Make It Easier. Always Use Cell References.</i> Setting the Global Parameters. Global Formats: <i>Global Column Widths. Manual Recalculations. Order of Recalculation.</i> More on Replication. Other Basic Planning Considerations.	
3. The Ins and Outs of VisiCalc: Printing, Saving, and DIF	31
Saving and Loading: <i>Reading a File. Naming Disk Files. Loading a File.</i> Printing: <i>Setup Strings. "Printing" to Disk.</i> Data Interchange Format (DIF): <i>Saving a DIF File. Tuples and Vectors. C R or RETURN.</i>	

PART III

Advanced Techniques with VisiCalc

- 4. **Advanced Template Design** 45
 More Helpful Hints: *Using Windows. Saving Parts of a Template. Rounding with VisiCalc. Defining Ranges for Easier Alterations. Forward References and Other Misfortunes.* Beginning with Booleans: *More Boolean Functions. The Quasi Booleans.* Using Overlays. Using DIF: *Rolling with DIF. Consolidating with DIF.* VisEXEC Files: *Rolling with VisEXEC. More on VisEXEC.*
- 5. **Enhancements: Hardware and Software to Use with VisiCalc** 73
Software Enhancements. Hardware Enhancements.
- 6. **Designing an Application** 81

PART IV

VisiCalc—The Advanced Version

- 7. **New Features** 103
 The Financial Functions. The Arithmetic Functions. Returning Labels with the Lookup Functions. Date Arithmetic.
- 8. **Attributes** 111
/AE or /GAE Expression. /AH or /GAH Hide. /AM or /GAM Mode: Protect and Unprotect. The Other Modes. Making Your Template User Friendly. */AT Tabs. /AL Labels with Gutters. /AV Values: Displaying Values. /AD Default.*
- 9. **Making Life Easier** 117
 Keystroke Memory. Clearing Memory. Help Screens. Variable Column Widths. Multiple Replications and Other Goodies. Page-Print Formatting. To Buy or Not to Buy?

APPENDIXES

- A. **Command Reference Guide** 127
- B. **Function Reference Guide** 141
- C. **The Blank Model Template** 161

D. DocuCalc Template Printout	167
Glossary	179
Index	185

PART I

Getting Started

INTRODUCTION

You are a businessman, a student, or a teacher. Or perhaps you are an accountant, lawyer, physician, financial analyst, or real estate investor. You have files of numbers, historical data, and future projections which you must organize to assist you in making a decision. You would like to be able to apply varying assumptions to your numbers to see what might happen if interest rates go down, birth rates go up, or M2B stays the same.

You know what formulas and logical relationships your numbers have, and you can organize them with pencil and paper. It is very tedious, however, to write everything down and figure your answers with a pocket calculator. And a change in one assumption will require a complete recalculation, which will take about as long as the first one. Then, when you're finished, you must send your handwritten output to a typist and a proofreader before distribution to your colleagues or clients. And next month, you can look forward to performing this tedious process all over again.

Maybe you have a data-processing department with large, mainframe computers, which could take your data, prepare the appropriate programs, process your data, and return reports to you. In six weeks. Or maybe six months. But you need your answers today.

You have read about the new, powerful microcomputers and might even have purchased one. But the prewritten programs you have purchased don't do exactly what you want to do with your data. Of course you could write a program to do the job in BASIC, Pascal, Forth, APL, or FORTRAN, but you are not a computer programmer and don't want to become one.

This book is written for you!

VisiCalc, a program written by Daniel S. Bricklin and Robert M. Frankston and marketed by VisiCorp, is generally considered one of the most, if not *the* most, successful program written for the microprocessor. Large national accounting firms are rumored to have recommended purchase of a micro, just to gain an environment for use of this program. Originally written for the Apple II but now available for other machines—such as the Atari 800, the IBM Personal Computer, the Apple III, the TRS-80, and Hewlett Packard equipment—VisiCalc probably comes closer to being a universally used and accepted program than anything else except Pac-Man. “VisiClones” or “CalcAlikes,” such as SuperCalc and MagicCalc, now abound, and a spreadsheet program has appeared for the small but mighty Sinclair ZX-81/Timex 1000 twins. With VisiCalc, or another spreadsheet program, a nonprogrammer can manipulate his or her own data and develop very powerful analytic tools.

In this book we will explain, in detail, the workings of VisiCalc's commands and built-in functions. More important, we will show you how you can plan and organize the information you need to do your work by utilizing VisiCalc to its maximum efficiency. In addition, we will point out some of the lesser known and nondocumented “features” of VisiCalc. Many of the techniques we will demon-

strate are usable, not only with VisiCalc, but with all of the spreadsheet programs.

We will not, except as illustration, present specific VisiCalc “templates” or “grids” to perform specific functions. It has been our experience that such “canned” templates usually must be altered to fit the situation, and that mindless entry and alteration of hundreds of VisiCalc cells is usually more difficult (and conducive to error) than thinking the problem through and setting up the template with appropriate replications. Once you really understand how to use VisiCalc, you will be able to cut your development time and tailor your templates to fit your specific needs. Oddly enough, proper design will make your templates easier to revise as your needs change.

Whenever you use VisiCalc, keep in mind that there are two principal keys to it. The first is an understanding of VisiCalc’s capabilities and how to use them. But almost more important is an understanding of your own applications. In this book, we will show you how to combine your knowledge of the problems you must solve with your ever-increasing VisiCalc skills to produce exactly the correct template.

Chapter 1 is an introduction to VisiCalc for those who don’t have a computer and have never used a spreadsheet program.

Chapter 2 will give you a thorough grounding in how to design a VisiCalc template, while Chapter 3 explains how VisiCalc communicates with the outside world through your disk drives and printer.

Chapter 4 contains expert-level hints and techniques about template design. Chapter 5 reviews hardware and software peripherals which can be used to make your work with VisiCalc even easier and more productive.

In Chapter 6 we take a real business problem: the hurried creation of a business plan for submission to a banker in support of a loan application. You may never have had a similar problem, but we think that you’ll be able to apply the design process outlined in Chapter 6 to develop your own planning techniques.

Chapters 7, 8, and 9 discuss the new Advanced Version of VisiCalc, which we find much more powerful and easier to use than the current standard version.

We hope that Appendix A, a comprehensive listing of VisiCalc commands, and Appendix B, a reference guide to VisiCalc’s built-in functions, will own a place on your bookshelf as references in addition to serving as useful guides during the reading of this book.

If you own neither computer nor spreadsheet program and are not sure whether VisiCalc could be useful to you, we hope to help you answer that question. Work through this book, concentrating on the template examples and ignoring the specifics that relate to the computer itself. You will, at the very least, come away with a good understanding of what VisiCalc can and cannot do. Then you will be able to decide if a spreadsheet program is for you.

Having noted that this book will be valuable to readers without computers, we should add that it is not intended as a guide for raw beginners with computers. If you own a computer and VisiCalc, we assume that you can turn on your com-

puter, boot (start up) VisiCalc, and format data disks. We also assume that you have at least read through the manual that comes with VisiCalc and that you have your VisiCalc reference card at hand. If you have that minimum of training, we can help you to unlock the dynamics of VisiCalc.

We will often distinguish between a template *user* and a template *designer*. By *designer* we mean the person who enters the formulas and formats into the VisiCalc workspace. By *user*, we mean a person who loads a predesigned VisiCalc template and uses it to manipulate the data that he or she enters. We intend this book to make you a *designer*, or at the very least, an informed template *user*.

In this book, we have used brackets with bold type to indicate special function keys. They are listed below, along with the functions they control.

[RETURN] or **[ENTER]**

[ESC] (escape)

[CTRL] (control)

[REPT] (repeat)

[/] (to start command)

[;] (to change windows)

[!] (to recalculate)

[RESET]

We are not programmers—we are professionals, a lawyer and an accountant. We began by studying our computers' instruction manuals just to get the devices working to begin with. We need our information processed our way, and we have neither the time, money, skill, nor inclination to prepare our own programs or to hire consultants. With VisiCalc, we have been able to get our computers to work for us. In this book we will show you how to use VisiCalc to get your computer to work for you.

CHAPTER 1

If You Don't Have a Computer

AN ACCOUNTANT'S WORKSHEET

Imagine an accountant's worksheet. You know what we mean. It is ruled off into rows and columns, and it is printed on very good paper so that errors can easily be erased. The sheet probably has room for 40 rows and 10 or 12 columns, and it can be used for everything from calculating trial balances to posting batting averages, from scribbling luncheon orders to creating models of international market interactions.

VisiCalc is basically an electronic simulation of that same accountant's worksheet. But VisiCalc has 63 columns lettered A to BK and 254 rows numbered 1 to 254. Each cell is named for its coordinate column and row designation. For example, the cell in the upper left-hand corner of the sheet is called A1; the cell in the upper right-hand corner is BK1; the lower left-hand corner cell is A254; and the cell in the lower right-hand corner is BK254. You can make entries to any cell, just as on your paper worksheet. But if you make a mistake with VisiCalc, you can correct it without an eraser, simply by covering the cell with the cursor and typing in the new data or formula.

Because of memory limitations in most microcomputers, not all rows and all columns may be used at the same time. However, even the smallest of the microcomputers will have available memory for hundreds of cells, and this should be plenty of space for most of the applications that we will discuss in this book.

The VisiCalc user can enter three types of information into any cell: labels, numbers, or formulas. A label is simply the name of something, consisting of normal alphanumeric characters. Labels are useful for heading columns or rows,

leaving notes for the user, and putting lines, borders, and the like into report formats. Labels are mostly ignored by VisiCalc when it does its calculations. Typical labels are SALES, PROFITS, TOTAL, ———.

A number is—well, a number. Like 1, 2.3142342, 6.00. Numbers are also digits strung together with numerical operators +, -, * (for multiplication), and / (for division). Numbers are used by VisiCalc when calculating.

Formulas combine numbers, references to other cells, and many built-in commands. A typical formula is "1.1*A2". This formula, when placed in Cell A3, would tell VisiCalc to take whatever value was in Cell A2, multiply it by 1.1, and place the answer into A3. As a result, A3 would always be 10 percent larger than A2.

If A4 were specified as "1.1*A3", A5 as "1.1*A4", and A6 as "1.1*A5", the number in A6 could represent the number in A2 compounded over four periods of time. For example, "1000" (dollars) placed into A2 would yield "1464.10" if compounded at 10 percent simple interest for four years. If you have a hand calculator with exponents, this is the same as the formula

$$((1000*(1.1))^4)$$

where "⁴" means "to the fourth power". We can change this example slightly, so that the compounding factor "1.1" would instead be contained in cell A1; it could thus be used as a variable. Our formula may now refer to Cell A1, and Cell A6 would be shown as "(A5*A1)". Figure 1.1 shows a VisiCalc Screen displaying the example we have just discussed.

Cell A2 contains the value 1000. The formulas in Cells A3, A4, A5, and A6 are quite similar; they each reference the compounding value in Cell A1, and they each multiply this compounding value by the value in the preceeding cell in order to compound the earlier value. If you think of the templates that you have created on paper, you will realize that cells are often related in similar ways. When such relationships occur, we can replicate the formula in a cell to a series of other cells, without being required to enter the formula directly into each cell.

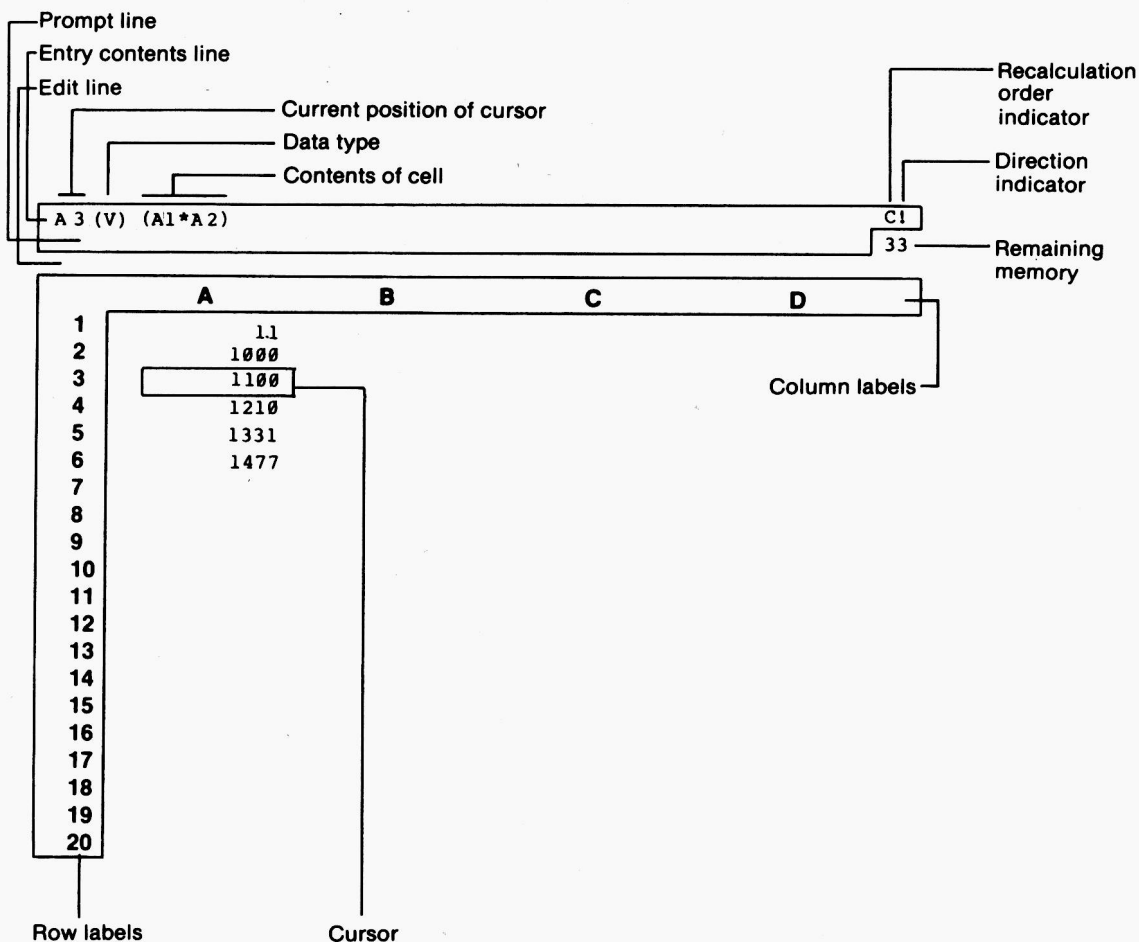
After entering the formula "(A1*A2)" into Cell A3, the VisiCalc user can enter the command "/R" to begin a sequence which will replicate the formula in Cell A3 into Cells A4, A5, and A6—or, in typical VisiCalc terminology, from the "Source Range A3...A3" into the "Target Range A4...A6".

REPLICATION

The operation of the REPLICATE command is difficult to describe, but the replication concept is so important to VisiCalc that we should conduct a preliminary discussion of the subject here.

The REPLICATE command, initiated by "/R", permits you to copy the contents of one or more cells to another group of one or more cells. The cells that are copied are known as the *source range*, and the cells that are copied *to* are the *target range*. The source range may consist of a single cell or a series of adjacent

FIGURE 1.1: AN ANNOTATED VISICALC SCREEN



cells in the same row or column. The target range may consist of a single cell, a series of adjacent cells in the same row or column, or even a rectangular area. We'll save the more complicated types of replications for Chapter 2 and Appendix A, but replicating the example discussed above is rather simple.

Place the cursor on Cell A3, and type "/R". VisiCalc's edit line will read "A3" and the prompt line will now read

REPLICATE:SOURCE RANGE OR RETURN

Press [RETURN], and the "CARRIAGE RETURN" key, [ENTER], or whatever your computer calls it, and the prompt line will now read

REPLICATE:TARGET RANGE

and the edit line will read

A3 ... A3:

This means that the source range has been entered and begins with Cell A3; it also ends with Cell A3. (In other words, the source range is the Cell A3.)

Type in "A4.A6" and hit another [RETURN]. The prompt line will now say

REPLICATE: N=NO CHANGE, R=RELATIVE

The edit line will now read

A3:A4...A6:A1

and A1 will be highlighted in inverse video or some other fashion.

The A4...A6 means that the target range has been identified, beginning with Cell A4, and ending with Cell A6. The "A1" begins the formula in Cell A3.

After identifying the source and target ranges, VisiCalc will highlight each formula reference in the source range and ask whether the reference should be replicated "relative" or "no change." If "no change," the target cell will reference exactly the same cell as the source cell; if "relative," the target cell will contain a reference with the same location relationship to the target cell as the source cell's reference to the source cell. Thus if Cell D4 contained a reference to Cell C3, the target cell E4 should contain a relative reference to Cell D3 (one up, one to the left).

As we have seen above, the A2 reference should always be to one cell above the current cell, and is therefore relative. The A1 reference, on the other hand, is always to A1, and should therefore be "no change."

As soon as you type the "N" for the A1 reference, "*A2" is added on the edit line and the highlighting will move from A1 to A2, requesting, in effect, "R" or "N" for the A2 reference. Hit "N", and you will see VisiCalc calculate each cell in the target range, making separate entry into each target cell unnecessary. This is not terribly impressive if you are using only four cells. But what if you wanted a year-by-year analysis for each of the next 20 years? Or month-by-month, for that matter? This ability to avoid the entry of large groups of cells is one of the most important features of VisiCalc. "REPLICATE," as previously stated, is a *very* important command.

Of course, the real power of VisiCalc comes now, when you can ask "What if . . . we start with \$8765.32?" Or "What if . . . we compound at 12 percent per year?" Try putting "8765.32" into Cell A2, and "1.12" into Cell A1 for the answer to these and other fascinating questions.

FUNCTIONING

We now understand how a VisiCalc spreadsheet (we will be calling it a grid, worksheet, or template) is set up, how to replicate, and the value of using cell references. But before we really get into exploitation of the dynamics of VisiCalc, we should really talk about VisiCalc's built-in functions.