

# The IIGS "peeled"

Volume 1

by :

D. BÄR

D. DELAY

Y. DURANT

J.L. SCHMITT

Eric WEYLAND edited by *Toolbox*

English version by Dagen Brock

This is a very early test work based on a book I like, and nothing more. Please don't send feedback on the book in its current state. I will endeavor to finish the translation and then do an editing pass and perhaps I will decide it is in a finished state. Who knows.

-Dagen

# TABLE OF CONTENTS

*Preface*

*Introduction*

I	65816
II	RAM
III	DRIVE
IV	GRAPHICS
V	GS/OS

## Bibliography

- \*Apple IIgs Firmware reference
- \*Apple IIgs Hardware reference
- \*Exploring Apple GS/OS and PRODOS 8
- \*Programming the 65816

Apple, the Apple logo, Appleshare,  
AppleTalk, Imagewriter,  
Laserwriter, Macintosh and Mac Terminal,  
Finder, Multifinder, Hypercard,  
Prodos, DOS 3.3, GS/OS, Apple II GS,  
Sane, Apple Desktop Bus,  
Appleworks, APW

are registered trademarks of  
Apple Computer, Inc.

High Sierra,  
Amiga,  
Atari,

are registered trademarks.

[ISBN info missing]

[Author info missing]

[Basically a lot of stuff is currently missing]

# Preface

*The Apple //, always and forever!*

Peeling a computer, what a bizarre idea. Today, isn't the world of personal computers split up into two quite distinct categories, the producers (of hardware and software) on the one hand, and the users, or rather the consumers, on the other? We're in the industrial era aren't we?

Personal computing, however, did not start like that at all. At the beginning of the 1980s, it all began with an inventive engineer, handyman, joker and bit of an anarchist, who manufactured a machine in his garage that has since sold millions of units: the handyman was Steve Wozniak, and the machine was very quickly called the Apple //.

A cohort {troop} of students, computing professionals in large systems and various intellectuals then seized the machine. These people were called hackers. They were curious as a cat ("I want to know what this machine has in its belly"), and by the will to master the possibilities, brought by the individuals themselves, by their computer ("I know this machine can do it, and I will not stop until I succeed in making it do it.") They were encouraged by Wozniak and by Apple: Hadn't Wozniak published, and put in the public domain, the original code of the monitor (the part of the program installed in the machine that drives low level functions) of the Apple?

Lots of water has flowed under the bridge since this time: technical progress, on the one hand (it goes very quickly in computing), and the requirements of competition and economic success on the other hand, have professionalized computing. Apple has become a multinational, traded on Wall Street, personal computing has won the enterprise, IBM has become a part.

Apple has made enormous efforts, spent much of the money earned from the elder model (Apple //) to develop the junior (the Macintosh) and to insert it in the enterprise. Many of the hackers became skilled developers, and generally not on the Apple //.

The Apple // itself has changed: it is now called the Apple //GS. It largely resembles its junior, the Macintosh: for example, it comes with an operating system (GS/OS), and with a software toolbox (Toolbox) which makes programming it a lot like a Macintosh.

Apparently there's not a place for Wozniak and the hackers any more. Don't we, of good souls, periodically announce the death of the Apple //?

Yes but look: curiosity, the taste of personal freedom, are things which don't lessen in value. Every individual who touches the keyboard of the Apple // is exposed to this virus (the author of these lines, as well as all those who contributed to this book, were contaminated in this way). More so it exists as a community of users, largely informal but very long-lived, which undertakes to share the knowledge between all those who wish it.

Also, as much as the Apple //GS is a computer of today (16 bits, graphics, sound, etc...), it's always an Apple //: thus one can always, if one wants, if one takes the time and controls one's self, do what we want. It has, for example, a super monitor, which enables us to know at any moment what is happening in our machine. It has, in its ROM, a Visit Monitor accessory which cannot be removed even when rebooting. Wozniak isn't dead, and the Apple // is quite alive.

Those who wrote this work are not professional developers on the //GS. They are users, but of a special kind: they

are people who want to control their machine, and who reach that point.

The personal computer revolution poses an important question, a question of civilization: "If I'm not the master of my machine, I'm just a user, then someone else is the master. And that someone, suddenly, is also master of me."

If, however, I understand exactly what is going on in my computer, I can make it do what I want, and use it as a tool of my own liberation. Instead of depending on the bureaucracies that produced the machine and its software, and doing exactly what they had expected that we do, we show that intelligence of individuals is always higher than that of a machine. We can in particular, which always amazed Wozniak, make it do things the designers never thought of, and which they thought impossible.

To take a somewhat technical example (that you may not understand until after reading this book): I discovered by reading the section on graphics something which I hadn't thought of before.

One of the limitations of the graphics on Apple //GS, is that there is only one graphics page in Super High Resolution, in contrast with HGR graphics on Apple //. This is embarrassing for animations. However, to be compatible with the Apple //, the Apple //GS has a system of echo memory called "shadowing". It has nothing to do with the graphics pages. Except, one author has said, "suppose that I temporarily suspend the shadowing, do I not get an equivalent to a pseudo-page 2?"

This is always the Apple // mentality:

This is not expected in a machine even less in its toolbox. But me, when I want it to do something, well, I'll get there! And when I get there, I'll know everything, including that which even Apple hadn't thought how to do. On other machines, those who would dare violate the prohibited and would have found something like this, would have kept the technique for himself or for sale, expensively. On the Apple //, this is already how we had invented Double High Resolution.

That's the Apple // spirit: the reasons why we chose an Apple //GS as a personal computer are not primarily the techniques (although the GS is an excellent machine). They are neither the most economical nor the least expensive. This is not the boss' machine, this is not the best of breadwinners. This is simply the most versatile machine that we can do what we want: it is an instrument of our liberty. If we happen to criticize Apple in this book, that no one is fooled: it is because we love our GS.

