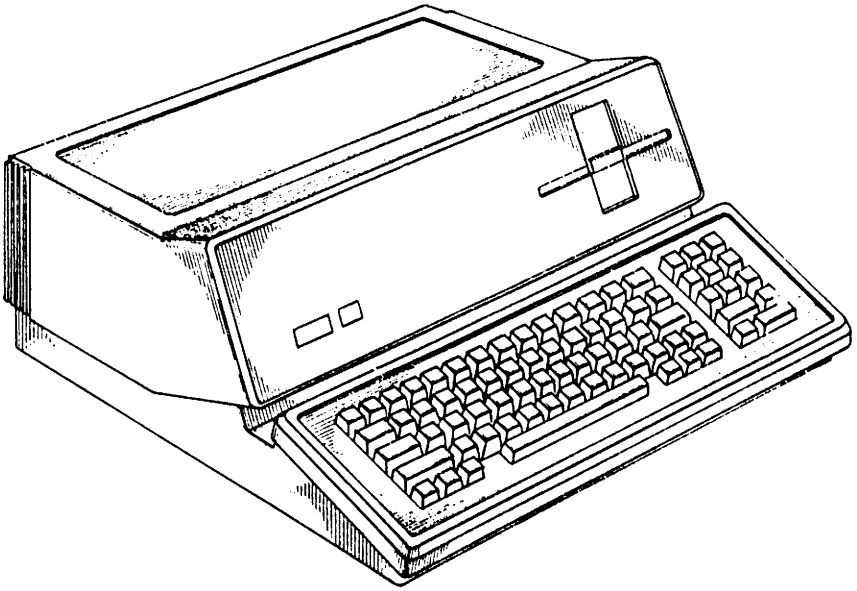
Apple II versus Apple III Hardware Architecture Comments

December 2002

eMail correspondence between

Steven Weyhrich
Rick Auricchio (Apple)
David Craig

Compiled by David T Craig on 05 September 2004



From: Steven Weyhrich
Sent: Saturday, December 14, 2002 2:05 PM
To: David Craig
Subject: CD's

I am sorry I didn't e-mail you sooner to let you know that I did receive the package of CD's that you kindly sent to me. What a treasure trove of information you have collected and organized! I have not yet had an opportunity to look at them in detail, but what I have seen looks pretty neat. You seem to be more of a trivia pack rat than even _I_ am!

I did notice one item: The Byte magazine "letter from Wozniak" does appear to be a joke by the guy writing the column, rather than a true letter from Wozniak. This may not be news to you, but I thought I'd put my two cents worth. :-)

You asked something about whether or not a disassembly of PRODOS had been done anywhere. I had thought that the writers of Beneath Apple ProDOS had published a disassembly listing, but it was only a listing of pertinent parts of the code (what a certain part of the code does, not the disassembly listing of the opcodes). I used to have this, but cannot put my hands on it right now. They also released a couple of updates when there were updates to ProDOS after v1.0, but they gave up after a while; Apple was actively updating ProDOS, whereas DOS 3.3 was left untouched for so long, it was easier to get a handle on what it did and how. Anyway, the only commented disassembly that I ever saw was in the Disassembly Lines column that ran for years in Nibble magazine. The columns were written by a fellow named Sandy Mossberg, who was an M.D. He usually took a part of the BASIC.SYSTEM file, disassembled it (with comments), and then discussed how to modify it to do something differently (such as present a different type of CATALOG listing, for example). I don't recall if he did this with the ProDOS kernel (contained within the PRODOS file), however.

I didn't know that Rich Auricchio was an Apple III programmer; I'd seen his name in some of the source code listings for the Apple IIe and IIc ROM' s that Apple published in the technical reference manuals for those computers, but didn't realize his input on the Apple III. Do you know if the odd bank-switching method for the 80 column text on the Apple IIe was a direct descendant of the code used in the Apple III ROM's? That is, having even columns coming from one bank of RAM, and the odd columns from a different bank. I know the III used the same 6502 processor as in the Apple II, and so could not directly address more than 64K at a time, thus making a bank-switching method necessary to get past the 64K barrier.

Speaking of trivia, take a look at this page:

http://techupdate.zdnet.co.uk/story/0,,t481-s2127406,00.html

It has the lyrics AND an MP3 of the "Apple II Forever!" song, one that I guessing might have been played at the famous Apple II Forever event when the Apple IIc was first introduced in 1984.

Steven Weyhrich <IX0YE>--<
Apple II History
http://apple2history.org

From: David Craig
Sent: Monday, December 16, 2002 9:06 AM
To: 'Steven Weyhrich'
Subject: RE: CD's

Hi Steve,

Glad to hear the Apple II CD collection has some items of interest to you.

> Byte magazine "letter from Wozniak" does appear to be a joke

I am aware of this and believe this letter was in BYTE's April issue which had april fools letters and I believe product announcements.

> ProDOS disassembly

Will look up the Nibble disassembly by Sandy Mossberg.

> Rich Auricchio ... Apple III programmer ... odd bank-switching ...
> 80 column text on the Apple IIe ...
> direct descendant of the code used in the Apple III ROM's?

Yes, the IIe 80 column text handling was the same as the III 80 column text handling. Many of the III's video handling techniques were transported to the IIe and the key people behind the III's implementation (software and hardware) worked on the IIe. Auricchio did the IIe ROM software and Walt Broedner did the IIe hardware (his initials are on the IIe motherboard).

Best in 2003.

Regards,
David Craig

From: Steven Weyhrich
Sent: Monday, December 16, 2002 12:20 PM
To: David Craig
Subject: Re: CD's

On Monday, December 16, 2002, at 10:06 AM, David Craig wrote:

> Yes, the II 80 column text handling was the same as the III 80 column
> text handling. Many of the III's video handling techniques were transported to
> the III and the key people behind the III's implementation (software and
> hardware) worked on the IIe. Auricchio did the IIe ROM software and Walt
> Broedner did the IIe hardware (his initials are on the IIe motherboard).

I see also that Brodener's name is on the Super II prototype motherboard that I have in my possession (see the scan of it on this page:

http://apple2history.org/museum/computers_prototypes/superiiclose.html

Now, HE is someone I'd love to be able to e-mail chat with, to learn more of the story behind the creation of the Apple IIe (everything I do know is taken from magazine articles, books, etc.)

Have a blessed holiday season and New Year yourself!

Steven Weyhrich <IX0YE>--<
Apple II History
http://apple2history.org

From: David Craig <dcraig@cyberwolf.com>
Date: December 16, 2002 5:32:54 PM CST
To: "'sdweyhrich@mac.com'" <sdweyhrich@mac.com>
Subject: Apple II versus Apple III hardware architectures

Hi Steve,

FWIW, here's a description of a key Apple III hardware feature which unfortunately did not make it into the Apple IIe design.

The Apple III supported a new memory addressing mode called EXTENDED ADDRESSING. This allowed a programmer to easily read or write to any memory location in any of the Apple III memory banks. The III supported up to 512K of memory grouped into banks of 32K each. This new address mode worked off of zero page of the system bank, which was always mapped (this was called bank 0) as far as the 6502 CPU was concerned. You placed a 2 byte pointer value into 2 adjacent zero page locations and then in another special page you placed a bank number. For example, to access byte 1000 of bank 5, you would store the address 1000 into say zero page location 25 and 26 and into the special page you would store the bank number 5 into location 25. Then when you executed a zero page indexed instruction such as LDA ($25),Y you were really reading from bank 5. Even Apple III Pascal had access to this feature so you were not tied to using assembly language on the III to have total memory access.

This extended memory architecture formed the heart of the III's operating system's memory access. In its simplest form it formed the foundation for the III's 80 column text and super hi-res graphic features. When transported to the IIe, this memory feature was only partially transported. This resulted in the IIe having a rather simple memory access scheme when compared to the III's more elegant scheme. The IIe's lack of extended addressing also caused Apple to implement the auxiliary memory access function in the IIe's 80-column video firmware programming that Rick Auricchio did.

In general, I would say that the III had a very elegant hardware architecture compared to the IIe. Apple ported some of this architecture to the IIe, but ended up with a IIe architecture that was (from my perspective) not as elegant as the III and almost kludgely in some regards.

Note that in your Apple II history you say that Walt Broedner came up with this idea during his IIe work. I believe that instead, this idea was done during the III's hardware development which took place in 1978-79 time-frame. Broedner's name is all over the III schematics since he did most of the III's detailed hardware design (Wendell Sander did the high level h/w design and is typically assigned the title of "III designer"). Also, Dick Huston worked on the III's OS (SOS) and later transferred this OS to the IIe where it was called ProDOS. One can therefore say that the III was really just a much more sophisticated Apple II model whose features were only surpassed by the IIgs in the late 1980s, 10 years after the III was designed.

Hope this III history has not bored you.

Best in 2003.

Regards,
David Craig

From: Steven Weyhrich
Sent: Monday, December 16, 2002 5:11 PM
To: David Craig
Subject: Re: Apple II versus Apple III hardware architectures

David,

As usual, you have exceeded my expectations in the usefulness of the information that you provided. As I've told people before, the biggest limitation I see in my Apple II History is the fact that SO MUCH of it is based not on true first-person interviews, but on magazine articles (some of which involved first person interviews), and I know that sometimes what ends up in a magazine is not as accurate as one would like it to be. Info that I have that specifically cites Broedner, Auricchio, Huston, and so on, does so entirely from what was mentioned about or by them in the articles that I do have and have saved over the years (interviews in Byte magazine, probably Call-A.P.P.L.E., and so on. I'll have to review the info you just sent about the III addressing scheme vs the IIe and see if I understand it enough to include such info in that part of the history.

To properly interpret design decisions made when it came to creating the Apple IIe requires knowing the environment at Apple at the time it was being done. If the IIe was actually being designed in the 1981-83 time frame, as Auricchio's web site suggests, this is at the height of Apple's efforts to get the Apple III accepted as a viable computer, and also while certain engineering energies in the company were being occupied with development of the Lisa and Macintosh. With the general attitude towards the Apple II that is supposed to have existed at that time, it doesn't surprise me that they made the IIe addressing scheme kludgy, rather than a more elegant scheme as was used in the III. On the other hand, depending on exactly WHAT 32K part of memory was being swapped in and out as the Apple III made its way through its potential 512K of RAM, the zero page addressing you mentioned may not have been possible on the IIe without radically changing how some of the classic built-in firmware did things (i.e., Applesoft, Integer BASIC, and the Monitor). Swapping in and out the ENTIRE 64K of RAM gave more real estate in which to do programming, but it made it difficult to actually USE that other 64K. You swap out to auxiliary memory, and suddenly your zero page values are different, your Monitor ROM locations are different (unless you took care to copy them all over), etc. You try to use a 32K bank-switching scheme, and now there are zero page locations that are being claimed by more than one program running, and all but the most compulsive programmer would have been pulling their hair out trying to get things to work.

In retrospect (always 20/20, you know), Apple SHOULD have done a couple things, when it comes to the Apple II/III drama:

1) They should have designed the Apple III as they did, but made as many parts of it as possible to be accessible to the Apple II emulator, rather than limiting it to a classic Apple II. If I had been a II Plus owner, who had a language card (extra 16K of RAM), plus a lowercase chip, plus the shift key mod, plus my Corvus hard drive (probably 5

megs!!), plus an 80-column Videx card, etc, etc, and I found that with an Apple III I could not only run ALL of my old software (even the stuff that required a souped up II Plus) but also NEW stuff that ran with that nicely updated Business BASIC, 80 columns, upper and lower case, and the SOS operating system, etc., I would have been interested in upgrading. To lock me out from the most powerful of my software, and FORCE me to buy new would have given me pause as to making a purchase (as it probably did to some II and II Plus owners at the time).

2) When the III was not thriving, taking ALL of the best hardware tricks learned in designing the Apple III and putting them into the IIe, so you have a computer that can easily be upgraded with more RAM, etc., instead of a less well done version.

Apple has likely finally learned that they cannot blow all the old users out of the water, but have to let them run their old stuff. They learned that with the PowerPC introduction, and now also with the OS X introduction; most of the old is compatible, but the new just runs better.

Oh, well, in another space-time continuum... :-)

Oh, yes: about the Super II motherboard. An Apple IIGS programmer named Nathan Mates obtained it once from someone at Apple (you can read the original post Nathan made about it if you go back to that link and click on the "up" link near the top middle of the main part of the page.) Anyhow, Mates withdrew from the Apple IIGS programming world somewhere in 1999, and at some point he e-mailed me and asked if I'd be interested in having it, since he didn't want it any longer. I said "Sure!", and so now it is hanging on my wall at home.

Again, thanks for your help!

Steve

From: David Craig <dcraig@cyberwolf.com>
Date: December 17, 2002, at 04:43 PM
To: "'sdweyhrich@mac.com'" <sdweyhrich@mac.com>
Subject: Re: Apple II versus Apple III hardware architectures

Hi Steve,

I have scanned selected pages from 2 of my Apple III notebooks which explain the Apple II emulation in more detail. Info also discussess the special mode we were discussing. Mode is called FUNNY MODE in a development memo, and is also called SATAN MODE (not devil mode as I incorrectly recalled) by Rick Auricchio who wrote the Apple III's Apple II emulator.

Will email you 2 seperate PDF documents. One contains general III info from an internal Apple reference manual. The other contains internal development memos that includes FUNNY/SATAN mode info.

Here's some more info about the III's memory addressing scheme that may interest you. III supported 2 zero pages and 2 stack pages. The III OS (SOS) used one zero page and one stack page. The III applications used the other zero and stack pages. This was done since both the zero and stack pages are rather small to handle both a complicated OS and complicated applications.

Hope this info is of some use to you.

Regards,
David Craig

From: Steven Weyhrich
Sent: Tuesday, December 17, 2002 4:18 PM
To: David Craig
Subject: Re: Apple II versus Apple III hardware architectures

On Tuesday, December 17, 2002, at 04:43 PM, David Craig wrote:

> These comments would have made a great Apple II position paper for
> Apple's executive staff in the early 1980s.

Ah, but the famous Reality Distortion Field would probably have discounted anything that Jobs did not himself believe in.

I also feel that in addition to having the benefit of hindsight, and knowing what actually DID happen, would have caused ANYONE in Apple's management to know what they SHOULD have done, if they could only go back and do it differently. This goes for the Apple III, and the Apple II in general, and the Macintosh, and the Mac OS, etc., etc. I am personally hopeful that the changes in products that have come about since a more mature (I hope) Steve Jobs returned to Apple will be looked back upon some day, and be viewed as the "right thing" to have done, to save the company and re-build market share.

Of course, the Osborne Computer Company, had they known how disastrous it was going to be to pre-announce the Osborne-2 before they were ready to build it and ship it would do things differently if they could just go back and do it again. :-)

Re: Devil Mode

DAVID! DO REALIZE HOW HARD IT IS FOR ME TO GET ANY REAL WORK DONE WHEN YOU KEEP GIVING ME THESE STUPENDOUS PIECES OF HISTORICAL TRIVIA??? AAAAAUUUUUGGGGHHHH!

(I'm not mad, just excited that I've learned something else, new, that I never knew before).

That's the most interesting thing that I've learned in the past week! I had always KNOWN that there was NO WAY to access the special Apple III features from Apple II emulation mode (BTW, was it called "emulation mode" or something else?) I should have known that if an engineer finds a way to do something, a hacker will find a way around it -- even if it was not a hack but was something it ware really designed to do. I'm surprised some hacker outside of Apple didn't discover it (or perhaps they did -- by now the ROM must have been fully disassembled, even if someone like yourself didn't have an official listing).

Okay, now this just occurred to me: Do you have in your collection (which I may now have on the CD's you gave me) a memory map for the Apple III, one that shows where SOS lives, and where this 32K switchable bank of memory lives, and (here's what occurred to me) where does an Apple II program execute when running in emulation mode?

Thank you for stimulating my creative juices! Now I have to figure out

how to fit this piece of info into the Apple II and III puzzle I
present in that chapter of the History.

Steven Weyhrich <IX0YE>--<
Apple II History
http://apple2history.org

From: Steven Weyhrich
Sent: Saturday, December 21, 2002 4:34 AM
To: David Craig
Subject: Re: Apple II versus Apple III hardware architectures

Thanks for the info again, David. I'll look over the PDF's and see what I can learn from them. I did get in touch with Rick Auricchio by e-mail, and yes, he does call it "Satan Mode". Interesting things (such as that mode) that programmers tend to come up with to get around the restraints put on them by superiors, eh?

Steve

On Friday, December 20, 2002, at 06:13 PM, David Craig wrote:

> Hi Steve,
>
> I have scanned selected pages from 2 of my Apple III notebooks which explain
> the Apple II emulation in more detail. Info also discussess the special mode
> we were discussing. Mode is called FUNNY MODE in a development memo, and is
> also called SATAN MODE (not devil mode as I incorrectly recalled) by Rick
> Auricchio who wrote the Apple III's Apple II emulator.
>
> Will email you 2 seperate PDF documents. One contains general III info from
> an internal Apple reference manual. The other contains internal development
> memos that includes FUNNY/SATAN mode info.
>
> Here's some more info about the III's memory addressing scheme that may
> interest you. III supported 2 zero pages and 2 stack pages. The III OS (SOS)
> used one zero page and one stack page. The III applications used the other
> zero and stack pages. This was done since both the zero and stack pages are
> rather small to handle both a complicated OS and complicated applications.
>
> Hope this info is of some use to you.
>
> Regards,
> David Craig

From: Rick Auricchio <rick@cfcl.com
Date: Wed Dec 18, 2002 12:08:09 PM US/Central
To: Steven Weyhrich <sdweyhrich@mac.com
Subject: Re: Apple IIe question

[Weyhrich] Do you remember what parts of DOS 3.2.1 and 3.3 you worked on?

DOS 3.2.1 was a patch to fix disk errors. In late '78 and through part of '79, Paul Lutus (developer of AppleWriter) had been experiencing I/O errors on his dual-floppy system. He was one of the few in the world with one, outside of Apple.

One day, while running his test script (thank goodness he had one), I heard the "deselected" drive click during a switch from drive 1 to 2 (or 2 to 1, it doesn't matter). It turns out the head on that drive was stepping a little bit when it shouldn't have, thus positioning itself off-track. I don't recall why a recalibrate didn't recover, but it could be because the head was 1/2 track off and it got *some* data, but failed to get it all. I believe if you could read address mark headers you didn't recalibrate.

I called Woz over, we looked at it, and he found a new capacitor on the motor-control board. Shugart had added this to smooth out the power without Apple's knowledge; he attached a scope and saw the power stayed up on the drive for perhaps 100mSec after it was deselected.

The seek routine immediately began stepping the "new" drive head right after turning off the "old" drive. Because the power didn't shut off immediately to the old drive, its stepper had enough power to click a little bit.

The problem didn't occur on older drives within Apple, because they didn't have that capacitor.

Woz and I added a 150mSec delay to the start of the seek, so there was enough time for the old drive to really die. This delay was invisible, because the seek would still complete before the spindle motor was up to speed.

[Weyhrich] That was DOS 3.2.1.

For DOS 3.3, I integrated Woz's 16-sector disk "core routines."

Shephardson Microsystems did the majority of the work on the original DOS 3 Apple bought DOS from them.

[Weyhrich] Randy Wigginton and Steve Wozniak wrote the RWTS part of it.

Correct. RWTS, Read-Write Track/Sector, was the main entry point of the "core routines," the floppy driver code.

[Weyhrich] I was also told that Dick Huston did work on modifying DOS 3.x.

Dick maintained DOS in general, fixing bugs. DOS 3.1 had plenty; DOS 3.2 fixed a lot of them. Dick also knew the core routines, so he perhaps helped with those too.

[Weyhrich] Also, legend had it (at least at one time) that there was no assembly source file for DOS (at least on the Apple II), but that it was patched via the mini-assembler. Is there any truth in that?

Not true, but close. DOS source lived on an S-100 Z80 system, but I forget what kind. That's where Dick made the changes. Some time after I got the assembler in good shape on the Apple II, DOS got moved there. I don't recall who moved it over. Could've been me, could've been Dick, or perhaps John Arkley.

GameBasic on the Apple I, predecessor of Integer Basic, was hand-entered in hex by Woz at parties. The Mini-Assembler came later than that...

[Weyhrich] David Craig told me that there was a secret in the Apple III called "Devil Mode"

"Satan Mode," a name I made up. Andy Hertzfeld discovered that you could use the /// as a II with access to the various features, IF you didn't require ALL the hardware compatibility that the "Apple II switch" enabled.

Initial attempts to use it were risky, because there were occasional things you'd forget about and you'd just crash. He then created a "Satan Mode Boot" diskette, which I later used to run the SubLogic FS-1 flight simulator at the increased processor speed.

I don't believe anyone bothered to try using bank-switching, extended indirect addressing, or 80-column video in Satan Mode; it was a novelty that was neat but didn't become very popular.

Once you flipped the /// into II Emulation Mode, you couldn't flip it out without a reset, nor could you access any of the advanced features.

[Weyhrich] [made floppy driver] "correct", was it an issue of functioning correctly

Yes. It was in ROM so the system could boot. Dick Huston fixed "the last bug" and offered a $50 challenge to find another. Two days later I smugly handed him about seven bugs! I declined the reward, but he insisted on paying.

One of the bugs caused multiple drives to fail, so I got the task of fixing the driver. Dick was beginning on ProDos---the Disk Division was hot to sell the 5MB Profile hard drive for the II machines---so it wasn't like they took him off the floppy and gave it to me.

I knew how the core routines worked on the II, and the /// was similar, but I had not actually done a floppy driver in its entirety. I spent a few weeks reading his code, understanding what it had to do (and how it did or didn't do the job). A lot of his code, in tight assembly fashion, was intertwined. The read and write paths, for example, came through the same block of code, with tests all over saying "reading?" or "writing?" to branch here and there within the block.

I went so far as to completely flowchart a driver, realizing that it

was faster and smaller to simply write multiple routines. Then create subroutines from the duplicated code. This is straightforward design, nothing special.

I then began coding and had the driver operational in a couple of weeks. Coding and testing went quickly once problems had been thought-out in the design phase. Separating fundamental things like the read and write path made it easier to debug; with a common path you can't debug the first write when it's going to do loads of reads first!

The ROM-based driver was fine for booting, because it didn't have to handle multiple drives, and, if there was a Write bug we wouldn't be doing writes. So we kept the ROM as is and made the floppy driver installable. When SOS loaded, it had a ram-based floppy driver.

[Weyhrich] Do you recall if the driver functioned differently from the one in DOS 3.x on the Apple II?

The driver was essentially identical in function, though the /// could have four floppy drives connected. The API was different, since SOS used the unix-style read/write/open/close/ioctl paradigm. The core routines were rewritten, but they were still about the same as those Woz had written in the II. They would be considered derivative work if you pushed on the copyright, for sure.

[Weyhrich] Were you the sole author of [///] SCP [system configuration program]?

Yes.

I also did much (or all?) of "Selector," which was a menu-based program launcher for the ///. I don't know how much of it I did, but I know I did the visual stuff. All that text-based folder-image scrolling and drawing.

Because Thomas Root did a great job with a smart-terminal interface for the console, a lot of the animation was done with a stream of control characters! I'd just package up viewport-setting controls, scrolling operations, and so forth, and the animation would happen in the driver.

[Weyhrich] I know that making the changes in the ESC(ape) cursor movement code to allow the use of the arrow keys was a BIG improvement for me, as I did a lot of Applesoft programming at the time the IIe came out, and it just made more sense than did the IJKM keys. Also, having the inverse "+" appear (when using the 80-column firmware) to indicate that you were in cursor movement mode was a BIG help also.

I forgot about that stuff. Remember uppercase-restrict? That was something that I recalled from the Xerox CP-V timesharing system: if you typed in lowercase, the system would upshift it to interpret it. I think that went in too.

[Weyhrich] Did you see any difference in the corporate culture when you returned for the second time, or for the third time? Was it very different as a (I presume) bigger company? Is there a time of working at Apple that you feel was most enjoyable for you?

My times were: Mar 79-Jan 83, Sep 85-Oct 92, Sep 95-Oct 01.

The first stint was when we had 75 people in Engineering. It was a

blast, because I'd worked with a friend building a 6502 computer from scratch in 1975. I bought my Apple II in 1977, and the learning environment in the early Apple days was great.

Andy Hertzfeld joined about six months after I did. He and I had been writing articles for Micro Magazine, and we both wrote an article showingan ONERR-GOTO patch for Integer Basic. We "recognized" each other from that. Alan Watson, who may still be at Apple writing technical manuals, also wrote an article on which DRAMs would work in the II. At the time he was working for Fairchild or another chip manufacturer (Fairchild?). I mentioned the article when we were introduced.

The second stint was after I'd learned UNIX and C. Five of us started the A/UX project. There I got to learn unix in great detail, wrote more drivers for various Macs around the Mac-II timeframe, and we did lots of good stuff with A/UX.

The third stint, after Taligent, was working on the CHRP project. I then moved to the Rhapsody project shortly after the Gil Amelio layoff binge. There, I worked with former-A/UXers and former-Taligent folks along with the NeXT people.

As time went on, more after 1995, you could see the big-corporate changes. Benefits became weaker and cost more, even though the employees were older with families (when we were young and single, we didn't need all the good benefits!)

Corporate bean-counters began cutting expenses as they always do: they cut what's easy, but not what really COUNTS.

For example, at one point in the early 90s, someone decided to cut out the T-Shirt budget. Why? Because it was an easy-to-spot line item on someone's budget. Never mind the morale backlash. That got overturned at some point, but not after it had done some morale damage.

You see this all the time in organizations where the finance people are outof touch with the organization. They cut a line item without really knowingwhether it'll make a difference. You have to cut the EXPENSIVE stuff, not a little cheap thing.

For example, they'd cut some minor thing that people liked, then the Repro/Printing department would print thousands of notices about changingout the Xerox machines for another brand. Instead of putting one notice atthe machine they plastered them all over the building. Waste.

The Telecom people would print thousands of GLOSSY card posters telling usthey're adding a new prefix for extensions (974-xxxx and now 862-xxxx). BIGexpense, and for what?

Anyway, I'm rambling...it's just that I see this stupidity all over. Someone wants to cut expenses and he begins cutting things without firstlooking at where the money is actually going.

--
- rick

-

-- -
Rick Auricchio rick@cfcl.com http://www.cfcl.com/rick
Acoustic Legacy Studios Cambria, CA USA 93428 805-927-7305
Years ago, I sent my mojo in for repairs. I still don't know if it's workin'.

From: David Craig
Sent: Friday, December 27, 2002 10:54 AM
To: 'sdweyhrich@mac.com'
Subject: Rick Auricchio and AppleWorks

Hi Steve,

Just read the "selector" part of Rick Auricchio's email to you and it jogged my memory about something Rick told me several years ago.

Rick said that he created a windows-based text library for the Apple III (or maybe II) that Rupert Lissner used for his AppleWorks program (called III EZ-Pieces on the III originally).

Therefore, Rick can claim fame to AppleWorks' user interface!

You may want to verify this with Rick if it interests you.

Regards,
David Craig

From: David Craig
Sent: Friday, December 27, 2002 9:13 AM
To: 'Steven Weyhrich'
Subject: RE: Apple IIe question

Hi Steve,

Thanks for forwarding Rick's email about his recollections of early Apple history. His comments are always fascinating reading and contain a cornucopia of historical and technical facts. I'm amazed that he can remember such small items as a disk capacitor -- I have a hard time remembering my name at times, let alone something that happened 20 years ago :-)

You asked about Apple II DOS and its development history. Check out the following web site for all the details. If you can get the source for DOS 3.1 from PAUL LAUGHTON that is something I would like to see.

> Paul Laughton
> paul@laughton.com
> http://www.laughton.com/Apple/Apple.html
>
> From: Paul Laughton
> Reply To: paul@laughton.com
> Sent: Friday, December 22, 2000 5:51 AM
> To: David Craig
> Subject: Re: Apple II DOS
>
> David,
>
> I still have the source code listing for Apple DOS. It is not exactly the same as the released DOS 3.1. There were a few bugs fixed by Apple after I provided them with the code and before they released.
>
> The listing is on the old style of computer multi-fold, print paper (very wide). I don't know exactly how it would get copied on today's xerox machines. It would be quite expensive since each of the hundred or so pages would have to be carefully hand positioned on the copy machine. If you remain interested, I can get a quote from a local copy shop. The book, Inside Apple DOS, does contain a significant part of the source (Randy Wiggington gave the authors an "illegal" copy.)
>
> Back when I worked with Apple I had a copy of the AppleSoft listing. Both in the original form as recieved from Microsoft and as modified by Randy. It was interesting to see how Gates and Allen wrote their code. I thought I did a much better job with the Atari Basic :-)
>
> Merry Christmas,
> Paul
>
> I have these vague plans to donate my Apple material to a museum one of these days.

Rick mentioned DOS residing originally on a non-Apple machine. You may be interested to know that Apple III SOS and Apple III Business BASIC were developed on the Apple II and the sources never resided on the Apple III. Apple's early publications were also done on a non-Apple (see Jef Raskin's Mac and Me history for the details).

Also, if you want to contact DICK HUSTON about his Apple work (DOS, SOS, ProDOS, Apple II peripheral card firmware, Apple II ROM firmware) here's his email address:

jrhuston@znet.com

Regards,
David Craig

From: Steven Weyhrich
Sent: Saturday, December 21, 2002 4:34 AM
To: David Craig
Subject: Re: Apple II versus Apple III hardware architectures

Thanks for the info again, David. I'll look over the PDF's and see what I can learn from them. I did get in touch with Rick Auricchio by e-mail, and yes, he does call it "Satan Mode". Interesting things (such as that mode) that programmers tend to come up with to get around the restraints put on them by superiors, eh?

Steve

From: David Craig
Sent: Monday, December 23, 2002 3:20 PM
To: 'Steven Weyhrich'
Subject: RE: Apple II versus Apple III hardware architectures

Steve,

>>>
Interesting things (such
as that mode) that programmers tend to come up with to get around the
restraints put on them by superiors, eh?
<<<

In terms of the Apple II emulation on the Apple III, I don't believe this "satan mode" was made in order to "get around the restraints put on them by superiors". For backwards compatibility reasons, there had to be a way for the Apple III to be put into an Apple II mode and the emulator software had to enable this mode. A soft-switch is most likely responcible for this mode.

I looked for the Apple II emulation mode soft-switch and could not find a specific reference to it. I did see in the Apple III schematic a pin on the VIA chip labeled AIISW which seems to be the magic switch. Given that Apple did not document this in its internal documents tells me they did not want others to know about this switch so that intermediate modes such as "satan mode" would not be accessible to 3rd party developers.

Merry Christmas and best in 2003.

Regards,
David Craig

###