History of Macintosh Memory

Addressing modes affect how you use your computer because they determine the amount of application memory you can use on your system. Application memory is the space your system works in when it runs applications, the memory you can see described under “About This Macintosh...” The size of this space limits the total combined size of all the applications you can have open at the same time. The System also uses up application memory.

The amount of RAM that can be “addressed” (used) as application memory on your system is limited by the total number of different memory addresses (locations) that the Macintosh can identify, each with a unique serial number. This identification is performed by sending 1’s and 0’s down a parallel set of data lines called an address bus.

This address bus became an important limitation during the course of the evolution of the Macintosh. Apple’s Macintosh system software was first developed on the Lisa and Mac 128 computers, both of which used the Motorola 68000 microprocessor. Although the internal architecture of the machine used 32–bit address and data paths, this microprocessor only had a 24 line address bus, so no address higher than 224 (about 16 million) was accessible. The top byte (the top eight bits) out of the total of 32 was not used by the 24–bit addressing scheme. So, this 24–bit addressing mode of the Macintosh only supported 16 megabytes of internal addresses.

This 24–bit/16 megabyte limitation was accentuated by another design decision made in the early Mac architecture. Of those 16 megabytes, only the lower 4 were allocated to supporting RAM addresses, hence the “4 megabyte barrier” familiar to users of Compact Macintoshes. The remaining upper 12 megabytes were allocated to addressing the motherboard, video monitors, ROMs, and input/output ports like SCSI and AppleTalk.

The 24–bit addressing mode did not seem like a big limitation on the earliest Macs which had only 128K of RAM. So the unused top byte was put to use by enterprising software engineers to signify various other things. For example, setting the high–order bits of an address could help “tag” (identify) that memory as being temporarily non–relocatable. Thus the hexadecimal address $80001234 represents the same memory location as $00001234, but is just tagged differently. This convention was assumed throughout the operating system, the ROM code, and by the applications which were developed to run on the Mac. And, as you will see, this led to later problems.

Eventually, the 4 megabyte barrier started to become a significant limitation and Apple began to convert the Macintosh hardware, software, and firmware (ROMs) to the 32–bit mode. For the foreseeable future, the internal limitations of RAM addressing would cease to exist. (Unfortunately this memory is also fragmented–at the one gigabyte level. As a result, “only” one gigabyte of application memory can be accessed under System

7’s 32–bit mode. Since, historically, upper level Mac memory needs have doubled every year this should become an issue sometime around 1998!)

The first step of this conversion program was to incorporate a new microprocessor. With the introduction of the Motorola 68020–based Mac II (and the later 68030–based systems), the Macintosh CPU graduated to a 32–bit address bus. The hardware was able to make all 32–bits of each address significant.

The 68020 (and later the 68030 and 68040) Macs also featured an improved 24–bit mode. The ROM addresses occur at 8 megabytes (instead of 4 on the earlier Macs) so these new modular Macs could immediately support up to 8 megabytes of RAM.

Virtual and MAXIMA became the first products to cross the 24–bit mode 8 megabyte barrier by collecting unused space in the upper 8MB and allowing the system to use it in the same way as the original lower 8MB of application RAM. Modular Macs with one NuBus card running a video monitor were able to address up to 14 megabytes of application memory using this approach.

Now the software became a problem. Some of the high–order address bits had been set for non–address purposes, so whenever the hardware used those particular “tagged” addresses the program would fetch information from an incorrect location; i.e. although the hardware was now 32–bit clean, the system, the embedded code in the ROMs, and much of the application software was not. At this point Apple started to clean up their system software and ROM designs, and tried to impress upon third party developers the importance of following Apple guidelines, so the software everyone wrote would work in a full–time 32–bit clean environment when the ROMs and System were ready.

The second step of the conversion program was the new “32–bit clean” ROM designs which were incorporated in the Classic II, LC, LC II, IIci, IIsi, IIvx, IIfx, Quadras, Performas and PowerBooks (except the PowerBook 100). These were first released in 1989 and are capable of accepting full 32–bit addresses. Systems equipped with these ROMs can run in 32–bit mode under System 7 and later systems, provided that the applications and extensions are 32–bit clean. The function of MODE32 is to make this 32–bit mode available for users of Mac II family computers released before the 32–bit clean ROMs were introduced, that is the SE/30, II, IIx, and IIcx. MODE32 contains software that makes these ROMs 32–bit clean.

Apple’s third step was to convert the operating system itself to run in this mode, a feature which is provided as the 32–bit mode option in System 7. One can guess that in some future System the 24–bit option may disappear entirely and 32–bit mode will be the standard. Connectix offers a product called OPTIMA, which creates a 32–bit mode for System 6, but only using Finder.

The last step in moving to 32–bit mode involves the application software. Only “32–bit clean” applications can be used in 32–bit mode–all others will cause a system crash.

Apple made 32–bit mode optional in System 7 because some applications today are incompatible with 32–bit mode and it will probably be a while before the full software base converts. When you want to run applications and/or extensions that are not 32–bit clean, you must switch your system to the 24–bit mode, at which stage you may want to use MAXIMA to enhance the 24–bit mode.