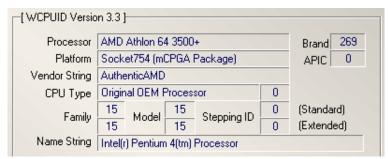
## Playing With MSRs: AMD -> Pentium4

{Written by fOdder} {fOdder (at) flork (dot) dk} [www.reteam.org -/- fOdder.has.it]

(reference the attached amdp4 test.zip file bundled with this paper)



See anything wrong with this image?

I guess everybody are familiar with the "computer type" string under Windows' "system properties" - stuff like "AMD Athlon(tm) 64 Processor 3500+", right before it lists CPU speed and RAM size. This information is returned by the **CPUID** instruction, and should be a pretty failsafe way of determining your CPU, right? After all, the string CPUID returns is hardcoded in the processor?

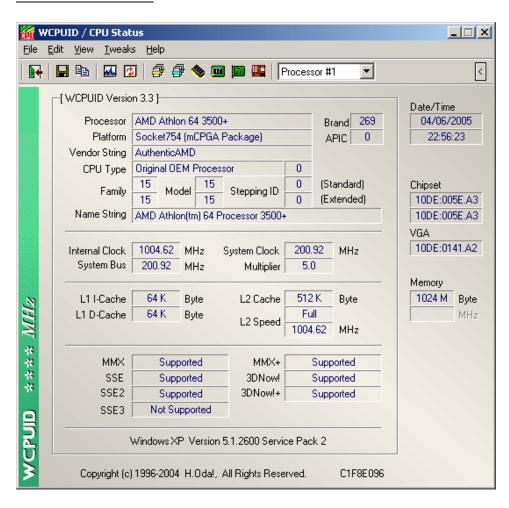
## Guess again.

A while ago, <u>Scali</u> stumbled upon a weird CPUID screenshot, showing a 4200+ processor as a 2200+. Let me quote Scali: At first I thought it was a fake, because I assumed the string was immutable like with Intel. So a 4200+ must display 4200+. But then people claimed that it had to do with the bios... at first I didn't believe them, but eventually I decided to check AMD's docs on CPUID, and it included a whole bunch of crap about how you should program the string in the bios.

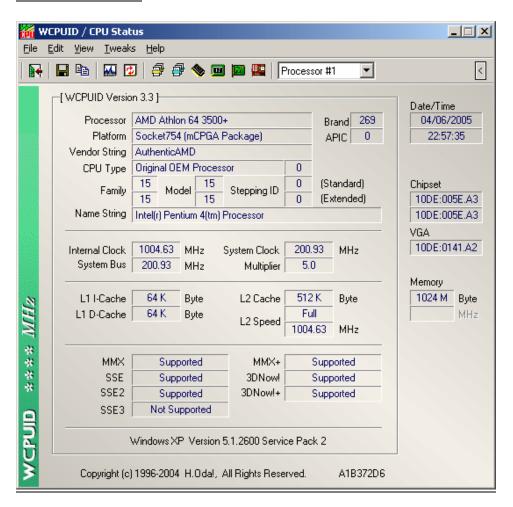
Luckily, I had finished some code to allow ring3 applications to manipulate MSRs a month or so previously, so Scali could quickly hack together an app to test the "CPU renaming". And indeed, AMD processors do allow you to change the CPUID string. It's documented in <a href="MAMD techdoc 20734">AMD techdoc 20734</a>, and I've put a test package with screenshots below. Notice that the screenshots show the internal clock for my AMD64 3500+ as 1004MHz - this is because I've activated the Cool'n'Quiet powersaving mode.

Please note that the demonstration application manipulates **MSRs** - **Model Specific Registers**. This could potentially be dangerous on non-AMD systems, or even AMD systems not supporting this. The code needs ring0 access (administrator privileges on NT, ring0 hack on Win9x). **Running the code is at your OWN responsibility!** 

## Before Screenshot



## After Screenshot



Have fun:)