

# WWDC 2020 - ARM Macs?

Beitrag von „guckux“ vom 25. Juni 2020, 20:18

## [Zitat von FriFlo](#)

meint ihr, ein Arm-Chip könnte Real-Time-Anwendungen betreffend besser, schlechter oder gleichwertig zu x86 sein?

tja - ich kann nur meine Meinung kundtun...

Ich schätze RISC wird keine Probleme damit haben, weil seine Maschinenzyklen nicht so lange dauern wie beim CISC. Dadurch würde ich - entsprechende Programmierung des Systems vorausgesetzt - mehr Agilität erwarten.

Letztlich würde ich vermuten, daß es darauf hinausläuft, Kaufe das, was es richtig macht, Bild und Video-Bearbeitung mit entsprechenden Grafikkarten, Audio-Bearbeitung mit entsprechenden Audio-Karten - einen guten DAC kann keine CPU ersetzen...

Schon in den 90igern habe ich entsprechende Systeme kennengelernt (SGI-Workstation für 40,000Mark, plus Videokarte für 80,000Mark und nochmal das gleiche für die Software -> Realtime Videobearbeitung!) ebenso im Musik-Bereich, was für den HomeAnwender ne SoundBlaster16 war, gab es in größerer, qualitativerer Ausführung mit mehr Kanälen und DACs für PCs und Macs...

Der Mac würde dann rein zu einer Management-Station werden...

RISC oder CISC (Bestimmt nicht alles korrekt, ich versuche nur mal einen Einblick/Überblick für "Laien" darzustellen, Fehler bitte mich aufklären - ich lasse mich gerne korrigieren :p 😞)

Es wurde schon erwähnt, daß die x86-Technologie einen Stillstand hatte und man an den GHz Grenzen kratzte, dann hat die israelische Entwicklungsabteilung einen Kniff angewandt - wie hier von [Aluveitie](#) erwähnt und RISC-Technik in die Kerne gebracht - und Intel musste sich dem Anwender gegenüber erklären, wieso plötzlich "lahm getaktete" CPUs (core2duo & Co) mit weniger als 2GHz schneller sind als ihre "Vorgänger", die 4GHz P4...

Der Kern des x86 hat RISC-Charakter, letztlich aber ist es kein RISC, denn es werden mit

Compilern ein lesbarer Code wie Basic, C, Pascal (sogenannte Programmierhochsprachen) in einen Maschinencode mit vielen hundert (Maschinen)-Befehlen übersetzt, welcher dann ausgeführt wird.

Mir fehlen jetzt die Referenzen - vielleicht hat da jemand anders was für mich? - aber ein CISC (x86) hat zB 2000Befehle und ein RISC hat 200. Ein Befehl muss aus dem Speicher geholt werden (lahme Geschichte) um dann intern verarbeitet verarbeitet zu werden (sowas geht schnell). Wenn ich nun 2000 Befehle habe, benötige ich eine gewisse Anzahl von "Wörtern" (Byte) um diese abzubilden - also mehr Wörter für einen Befehl, im Gegensatz zum RISC mit 200 Befehlen, welcher mit weniger Wörtern zurecht kommt. Unterm Strich muss also weniger bei einem RISC zur Befehlsausführung weniger zur CPU übertragen werden. Der Befehl ist kürzer und somit schneller an der Ausführung als bei einem CISC.

Dafür hat der CISC "Spezialisierungen", auf etwas Einfaches reduziert, kann er eine 3x10 direkt berechnen, während ein RISC die 10 nimmt und sie 3mal addieren muss... Im RISC müssen also manche spezialisierte Funktionen / Rechenschritte mit mehr Taktzyklen berechnet werden als in einem CISC.

Deshalb hat sich der RISC auf einen Befehlssatz reduziert, womit 80% der häufigsten Operationen abgedeckt werden. Das spielt er dann aus...

Aus dem praktischen Bereich:

Ich arbeite seit vielen Jahren mit Solaris-Server auf Sparc (RISC) und x86 (CISC). Bei Datenbank Anwendungen hatte der Sparc mit 1.5GHz kein Problem mit dem Xeon 3GHz mit zuhalten oder ihn sogar hinter sich zu lassen, im Gegensatz zB bei manchen (!) Bildoperationen, die waren gleich bis manchesmal schneller beim x86... oder beim Komprimieren - keine Chance des RISC zum CISC...

So, Danke an alle, welche durchgehalten haben, ich hoffe, ich habe denen, die mehr und besseres Wissen in dem Bereich als ich haben nicht zuviel Kopfschmerzen verursacht 😊