

https://de.wikibooks.org/wiki/Assembler-Programmierung_für_x86-Prozessoren/_Befehlsliste

interessant finde ich auch das - dass eben JL und JB nicht dasselbe sind und JA und JG, laut dieser Sammlung. da sich JL und JG auf logische Operationen mit arithmetischen Ergebnissen beziehen, während sich JA und JB auf Arithmetische Operationen beziehen, ... ich würde das hier ernst nehmen... ob das tatsächlich so ist, oder unweit selbst, wenn es wirklich so ist es sich nicht am Ende trotzdem beides für beides verwenden lässt, oder liesse, weil das Ergebnis, vielleicht manchmal vielleicht auch immer, oder meistens, drei unterschiedliche Fälle, das das gleiche wäre, das muss ich mir genauer anschauen...

Okay, sie haben richtig besser auf mein Assembler Skript auf Sie haben vollkommen recht Erstens mal in der ALU ist cmp nun mal also Compare nun mal auch eine Arithmetische Operation das ist logisch das ist Teil sozusagen der ALU Es ist natürlich möglich über den MOV Befehl Und mir ist LODS und STOS des x86 Intel immer bekannt gewesen Es wird allerdings teilweise eben auch, weil der MOV Befehl möglicherweise durch den Assembler selber also durch die x86er Assembler als Gasm tasm und so weiter einfach durch ein LODS und STOS was ja üblich ist, hab ich dir gesagt auch Mips32 atmega8 also Lade und Speicherarchitektur nach Computersystem eins und zwei wird keiner ohne diese Befehle extra machen. Nur ich weiß ja nicht hier könnte man Binärcode Wird daraus heimlich ein LODS oder STOS gemacht wird es bei Intel heißt mir sind diese Befehle Oft werden sie eben von Intel in der Literatur es gab ja eben auch von Anfang den Streit bei dem 8086 alles per mov gemacht Also ist ganz eindeutig, dass ich damit auf Speicher stellen zugreifen kann und das Problem LODSB LODSW LODSD Benutze und inzwischen wohl auch LODSQ ist damit und es hab ich allerdings auch in manchen Büchern gefunden. Also in manchen Büchern wurde der dieser Befehl LODS Wohl durchaus sehr stark benutzt, um Daten zu laden und eben nicht nur dann wenn es um sozusagen speziell die Zeichenketten geht, sondern da gab's es gibt Bücher, die so auch im haben, muss ich trotzdem sagen, wird es oft mit Zeichenketten vorgestellt, was vielleicht Zeichenketten sicher ist. Das ist mit einer Wiederholung nun eingeführt, nämlich mit dem REPZ Befehl. Es stehen die Befehle trotzdem für sich. Die wurden häufig damit eingeführt. MOV es geht auch eben über das MOV was vielleicht der Assembler selber macht, ja, ich wurde oft anders vorgestellt, und ich auch daran gewöhnt, dass weil du noch immer so und es wurde eben immer gesondert vorgestellt auch logisch ist weil das sind ja Speicher Zugriffe und Speicherzugriffe das sind eben oft enthalten, sondern ganze Tabellen sag ich mal und vielleicht wird es einfach den Leuten da muss ich gleich seitig sagen, wie gesagt, mit dem CMP das ist natürlich Teil wenn auch die wieder nichts zu tun haben die keine Sprungbefehle sind es bedingte Verzweigungen den muss Teil der ALU also arithmetisch. Logisch und dazu kommen immer noch diese AAA die ich bishe allerdings wirklich ganz starker Teil müsste man sagen vom Grundwortschatz oder so Aber das Wichtige LODS und STOS wie auf allen Maschinen und nicht nur das, sondern auch nein, mir fällt auf das ist doch nicht das gleiche ist ich nehme sie wieder zurück also ich LODS und STOS nicht mit MOV gleich zu setzen, denn diese haben ja keinen Operanten. Also, ob MOV das gleiche ist nur von Assembler anders übersetzt, nein, weil der Quelle Operant von In diesem Befehl sowie der Ziel Operant ist immer über SI und DI gegeben, immer darüber geht MOV über alle Register angesprochen, alles ansprechen kann. Also ich kann [BX] nehmen und