

Das nächste Kapitel dieser Vorlesung ist überschrieben mit Entwurf des Befehlsatzes Instruction Set Design oder auch ISA Instruction Set Architecture. Eigentlich neben der Entwicklung der reinen Hardware das wichtigste Thema für den Rechenarchitekten. Es geht um den Entwurf des Befehlsatzes. Um das zu verstehen, müssen wir wieder auf ein paar Voraussetzungen aus der Grundlagenrechenarchitektur-Vorlesung zurückgreifen. Sie sollten also wissen, was eine von Neumann-Architektur ist. Sie sollten eine grobe Ahnung davon haben, was ein Maschinenbefehl eigentlich ist. Was wir in diesem Kapitel Ihnen vermitteln wollen, ist, dass wir die Wechselwirkungen besser verstehen, die bestehen zwischen dem Befehlsatz, zwischen dem Compiler und der Architektur. Also der Befehlsatz ist ja die Fragestellung, ist das jetzt schon Hardware, Maschinenbefehle oder ist es schon Software? Der Compiler muss ihr Anwendungsprogramm in Maschinenbefehle übertragen oder die steuern die ganze Architektur, die ich gebaut habe. Dementsprechend sitzen wir genau an der Schnittstelle zwischen Hardware und Software. Und worum geht es jetzt in diesem Kapitel genau? Wir wollen uns anschauen, welche unterschiedlichen Möglichkeiten wir haben, Befehlsformate zu definieren. Wir wollen schauen, wie häufig die, also wir klassifizieren die Befehle erstmal also was sind richtig wichtige Befehle, welche Befehle sind vielleicht nicht ganz so wichtig. Ganz wichtig ist natürlich immer der Zugriff auf den Speicher, der ja möglichst schnell und möglichst einfach gestehen soll. Dementsprechend kann man sich sehr viele Gedanken machen zu Adressierungsmodi und zum Speicherzugriff. Wir wollen schauen, wie wichtig sind Konstanten in Anwendungsprogrammen und wie spiegelt sich das dann wieder im Befehlsatz, diese Konstanten. bisschen problematisch insbesondere wenn ich parallel denke also wenn ich an pipeline denke kontrollbefehle es ist auch ist ja nichts anderes letztendlich als als ein sprungbefehl ist es auch immer ein problem also irgendwelche verzweigungen die ich da habe in meinem programm dann wie eng hängt eigentlich ist der compiler an der hardware ich der compiler arbeitet erst mal gegen eine ist diese virtuelle Maschine jetzt schon identisch mit der Hardware? Das muss ich mir auch genau Gedanken machen. Dann haben wir heutzutage im wesentlichen noch Registermaschinen und ich habe eine beschränkte Anzahl von Registern und wie werden diese Register verwaltet? Wie viele Register brauche ich überhaupt? Was ist das Optimum an Registern? Auch das muss ich irgendwie ermitteln und dann ganz zum Schluss uns betrachten Compileroptimierungen. Wir werden dieses Kapitel es schaffen, aber schauen wir mal wie weit wir kommen. Ok, also mein Befehlssatz, mein Instruction Set ist die Schnittstelle