

Ich begrüße Sie recht herzlich zur Vorlesung Rechnerstrukturen in diesem Wintersemester. Es ist die erste Vorlesung und wir müssen uns leider online treffen, weil eben Corona sich nicht groß verbessert hat treffen. Ich würde vorschlagen, wir gehen erstmal das Prozedere durch und dann sprechen wir über Inhalte. Wenn Sie mich jetzt hier live hören, dann haben Sie es ja geschafft, unsere Vorlesungsseite zu finden. Ich blende die hier nochmal kurz ein. Wir haben also für Sie eine Webseite vorbereitet, wo Sie den Stream sehen. Das ist bei mir jetzt der schwarze Bereich, weil ich den nicht nochmal mitlaufen lasse. Und Sie haben auf dieser zwei Möglichkeiten mit mir zu interagieren. Sie haben die Möglichkeit hier rechts eine Frage zu stellen. Diese, ich ziehe es mal ein bisschen rüber, diese Frage, die wird hier komplett anonym gestellt und wir haben hier einen Tutor mit in der Vorlesung, der versucht dann diese Frage, wenn es eine kurze Frage ist, möglichst sofort zu beantworten textuell. Beantworten lässt, dann wird er diese Frage zu mir durchstellen und aufgrund dieses Streams gibt es zwar eine kleine Zeitverzögerung, aber ich werde natürlich dann versuchen, diese Frage hier live in der Vorlesung direkt zu beantworten. Sie haben hier unten die Möglichkeit zu chatten. Dieser Chat ist nicht ganz anonym, beziehungsweise ist nicht anonym, weil Sie sich da vorher bei StudIP oder über StudIP bei diesem Matrix-Server anmelden müssen. Und dann sind Sie natürlich hier mit Ihrer können mitverfolgen, was gechattet wird. So, also Fragen anonym, Chat nicht anonym und hier findet natürlich der Stream der Vorlesung statt. Okay, gut, dann können wir zur Vorlesung selbst kommen. Auch hier wird es jetzt erstmal ein paar organisatorische Hinweise geben. Und zwar klar, Vorlesung wird gehalten von mir und es ist eine Frontalübung, also eine Hörsaalübung und wir werden uns abwechseln mit Vorlesung und Übung und zwar im Mittel wird es so sein, dass es immer zwei Vorlesungen gibt und dann kommt eine Übung. Das heißt die Termine sind dienstags und freitags und je nachdem, also die ersten beiden Vorlesungen werde ich halten und ob dann Übung oder eine Vorlesung kommt, das entnehmen Sie auf Webseiten. Wir versuchen immer alle Informationen auch in StuttIP zur Verfügung zu stellen. Das heißt, Sie können sich zwar alternativ auf unseren Webseiten und in StuttIP informieren, aber alle wesentlichen Informationen stehen jeweils in StuttIP. So auch eben der Ort der Vorlesung, die findet ja hier online statt und wir haben hier nochmal die Internetadresse, damit Sie diese Seite finden. Gut, alle die jetzt zuhören, die haben ja die Seite offensichtlich schon gefunden. Sie können natürlich alle Materialien zu dieser Vorlesung auf den StudIP herunterladen, das heißt es gibt Folien, es gibt Informationen, Sie kennen das ja inzwischen mit dem StudIP, einfach immer nachschauen und sich auf den Stand halten. Ich habe die Folien im PowerPoint-Format und im PDF-Format zur Verfügung gestellt und kleiner Hinweis noch, ich bin bei der Vorbereitung der jeweiligen Vorlesung immer dabei, so ein bisschen aktuelles Mal und deshalb kann es sein, dass sich im Lauf des Semesters die eine oder andere Folie nochmal ändert. Also ich habe schon mal alle Folien hochgeladen, aber die endgültige Fassung der Folien steht erst am Semesterende zur Verfügung, weil ich eben während des Semesters noch Folien ändere und dann die Änderungen aber auch immer gleich hochlade. Sie finden da natürlich auch in StudIP die Termine, falls Vorlesungen mal ausfallen sollten

und so wie Sie das gewohnt sind und auch hier finden Sie alle Details in StudIP. Zum Beispiel werden wir ein Repetitorium am Ende des Semesters veranstalten und Termine und ähnliches dann eben entsprechend in StudIP. Gut, worum geht es jetzt inhaltlich in dieser Vorlesung? Wir setzen ja zum Hören dieser Vorlesung voraus, dass Sie schon die Vorlesung Grundlagen der Rechnerarchitektur besucht haben. Architektur, ja gut, da haben wir versucht, Ihnen zu erklären, wie so ein Prozessor denn funktioniert. Und wir haben begonnen mit ein paar Voraussetzungen in der Informationstheorie und haben dann uns schrittweise so einen Rechner gebaut, in dem wir eben Execution Unit, Control Unit, Speicher und das Zusammenspiel derselben besprochen haben. Wir haben dann schon ein bisschen an Performance gedacht und einführend mal über Pipeline gesprochen und über Performance haben damit die Grundlagen gelegt, die Sie eigentlich schon brauchen, um jetzt dann diese Vorlesung vollständig zu verstehen. Hier jetzt in der Vorlesung Rechnerstruktur, warum geht es da? Wir wollen uns erstmal die Ziele einer Rechnerarchitektur anschauen. Also welche Vorgaben, welche Ziele habe ich, wenn ich Rechnerarchitektur betreibe? Es geht natürlich sehr schnell dann um Performance, also um Leistung und Möglichkeiten zur Leistungssteigerung. sehen dass jedes einzelne detail sehr wichtig ist in bezug auf die leistung also muss man erst mal gedanken machen wie soll denn mein befehl satz aussehen also welche befehle erlaube ich wie konstruiere ich einen befehl satz was sind die eigenschaften des befehl satz ist was sind wichtige eigenschaften da geht es dann im kapitel befehl satz design drüber dann haben wir das kapitel alu design da geht es ein bisschen detizierter darum wie man eine ALO, Performance, also geeignet für gute Performance Design. Datenpfad, da geht es natürlich sehr intensiv darum, um Pipelining, welche Stufen kann ich alles definieren, wie arbeiten diese Stufen zusammen und das ist quasi eine direkte Fortsetzung der Pipeline-Grundlagen aus der Grundlagen-Rechner-Architektur. Wir werden uns den Speicherzugriff ein bisschen genauer betrachten müssen, Caches, also wie groß mache ich einen Cache, warum ist die Cache-Hierarchie, wie sie heutzutage aussieht, das wollen wir uns alles genauer anschauen. Und dann geht es schon ein bisschen in die Parallelität, wir sprechen über Superskalarität, also das ist eher so Parallelität bei Monoprozessoren und diese gestrichelte Linie, die zeige ich dann im Übergang zum zweiten Teil der Vorlesung, da geht es dann um parallele Rechnerarchitekturen wie man auch performance möglicherweise modellieren kann und dann sprechen über multi core und multi threading und am schluss dann über synchronisation gut begleitende literatur gibt es natürlich auch zu dieser vorlesung und zwar ja das wichtigste steht gleich an erster stelle da oben das ist computer architecture quantitative approach das ist ein buch von des Rechenarchitekten. Also wir haben daraus einige Beispiele entnommen und ein Teil der Vorlesung basiert auch ein bisschen auf diesem Buch. Also da steht eigentlich alles drin, was für einen Rechenarchitekten so wichtig ist. Da gibt es dann auch noch ein bisschen eine Fortführung zu diesem Buch, das heißt dann Computer Organization and Design, ebenfalls von David Patterson und John Hennessy. Ein richtig dicker Wälzer. Sie sehen ja, das nochmal wichtige Informationen für Rechnerarchitekten. Zum Thema Mikrocontroller und Mikroprozessoren gibt es auch noch eine Buchempfehlung, wo man sich dann eben

mal Details zur grundsätzlichen Funktionsweise von Mikroprozessoren nochmal holen kann. Gut, worum geht es jetzt genau bei der Rechnerarchitektur? Ja, wir haben jetzt schon mal ganz grob über Stichpunkte gesprochen, müssen wir zuerst sprechen, um dann auch ein bisschen genauer zu verstehen, ja was ist denn eigentlich jetzt genau Rechnerarchitektur? Jeder kann sich unter Architektur was vorstellen, wenn es um den Bau geht, wenn es um Häuser geht, aber was sind denn die Aufgaben des Rechnerarchitekten, ist möglicherweise noch nicht so ganz klar und deshalb wollen wir uns erstmal anschauen, welche Ziele, welche Vorgaben habe ich denn eigentlich als Rechnerarchitekt? So, dafür brauchen wir dann natürlich das entsprechende nächste Kapitel. Dieses Kapitel ist überschrieben mit dem Thema Ziele der Rechnerarchitektur. Als Voraussetzungen, um dieses Kapitel zu verstehen, müssen Sie wie bei allen anderen Kapiteln dann auch natürlich die Grundlagen der Rechnerarchitektur gehört haben, weil wir einfach Begriffe verwenden, die da einführend definiert werden. Da müssen Sie sich was darunter vorstellen können, sonst verstehen Sie natürlich einige Sachen nicht, die ich dann hier erkläre. Worum geht es? Wir möchten als Ziel für dieses Kapitel ausgeben, dass wir eine Motivation geben für eine performancebasierte, quantitative Sicht der Rechnerarchitektur. Also es geht um Leistung, es geht um Leistungssteigerung, um die S-Kurve, was diese S-Kurve bedeutet für den Rechnerarchitekten, was sie beschreibt, die auch für uns als Rechnerarchitekten gelten. Also wir können natürlich die Physik nicht außer Kraft setzen. Wir sind an die Gesetze der Physik natürlich gebunden. Welche Optionen haben wir als Entwickler? Und dann unter diesen Vorkenntnissen nochmal betrachten die Inhalte der Vorlesung Rechnerstruktur. Okay, ja, Leistung, Performance, was ist das eigentlich? Formel, die Leistung ist definiert als Arbeit durch Zeit. Und ganz allgemein gesprochen gilt natürlich diese Formel auch in der Rechnerarchitektur. Also da sind wir erstmal im einfachen Ansatz für die Definition der Leistung, unterscheiden wir uns als Rechnerarchitekten überhaupt nicht von anderen Disziplinen. Ja, wie wird bei uns, also in der Rechnerarchitektur, wie wird hier die Es geht natürlich hauptsächlich hier um Technologie erstmal. Also wenn Sie überlegen, Grundlagen der Rechnerarchitektur haben wir das besprochen, so die ersten Rechner waren mechanisch, dann kamen elektrische, elektronische Rechner dazu, das heißt es wurde viel mit Relays, mit Röhren gearbeitet, dann wurden irgendwann die Transistoren erfunden und die integrierten Schaltkreise, also ICs erfunden und da gibt es natürlich unterschiedliche Schaltungstechnologien, etwas langsamere, also zum Beispiel ECL Technologie sind bipolare Transistoren, das ist natürlich