



Grundlagen der Rechnerarchitektur

SS 2008 – Übungsblatt 11

Ausgabe: 11. Juli 2008

Aufgabe 1. *byte ordering*

Was ist unter „little endian“ (little end first, least significant byte first) und „big endian“ (big end first, most significant byte first) zu verstehen? Erläutern Sie das NUXI-Problem beim Datentransfer zwischen verschiedenen Computern.

Wo — außer beim Netzwerk-Datentransfer — kann das „byte ordering“-Problem im DV-Alltag Probleme bereiten?

Aufgabe 2. *CSMA/CD*

Wofür steht das Akronym *CSMA/CD*? Erläutern Sie das damit bezeichnete Verfahren am Beispiel des Funkverkehrs mehrerer Funkstationen, die gemeinsam nur eine Funkfrequenz benutzen.

Aufgabe 3. *CISC*

Diskutieren Sie CPUs mit „reinem“ Hardware-Decoder (ohne Mikroprogramm), und vergleichen Sie sie mit mikro- oder gar nano-programmierten CPUs.

Was sind „wait states“ und welche Vor- bzw. Nachteile sind damit verbunden? Warum treten die negativen Auswirkungen von Warteperioden auf aus dem Speicher benötigte Datenworte in modernen Multitasking-Betriebssystemen weniger stark in Erscheinung?

Was ist unter der Bezeichnung „Minimierung der semantischen Lücke“ zu verstehen? Wo wurde dieser Ausdruck benutzt und weshalb kam es zu diesem „Ziel“?

Aufgabe 4. *RISC/CISC*

Zitieren Sie das Designprinzip von RISC-CPU. Was bedeutet RISC und aus welchem Grund trat das RISC-Designprinzip in Konkurrenz zum seinerzeit vorherrschenden CISC-Designprinzip?

Was ist ein μ P-Speicher? In welchen Typen von CPUs tritt er auf und wo ist er innerhalb der CPU lokalisiert?

Diskutieren Sie mindestens sechs Eigenschaften, die RISC-CPU's von den zu Zeiten der Einführung von RISC-Rechnern üblichen CISC-CPU's typischerweise unterscheiden.

Warum sind bei RISC-Rechnern *optimierende Compiler* noch wichtiger als bei CISC-Rechnern?

Aufgabe 5. *Pipeline*

Beschreiben Sie in eigenen Worten das Prinzip einer Befehls-Pipeline. Was ist in diesem Zusammenhang unter den Begriffen Latenz und (Befehls-)Durchsatz zu verstehen. Welche Probleme können beim Betrieb einer Pipeline entstehen?