

Prüfungsprotokoll Informatik B

Vorlesung: Grundzüge der Informatik B1, B2 (bei Bärbel Mertsching), B3 (bei Bernd E. Wolfinger)
Prüferin: Bärbel Mertsching (Beisitzer: Steffen Schmalz)
Prüfling: Stefan Witt
Zuhörer: 5 an der Zahl (mehr werden nicht zugelassen)
Datum: 16.3.1999 (10³⁰ Uhr in Raum F-330a)
Dauer: ca. 40 Min. (also ca. 10 Minuten überzogen)
Vorbereitung: 13 Wochen, davon 10 intensiv
Literatur: Skript von Mertsching (mehrfach durchgearbeitet),
Schiffmann/Schmitz: Technische Informatik 1,
Tanenbaum: Computernetzwerke,
Duden Informatik
Veranstaltungen: B1-, B2-Praxistutorium,
Projekt Mikrorechnerbau,
Praktikum Elektrotechnik
Note: 1,0

Die Prüfungsatmosphäre ist *sehr* angenehm. Frau Mertsching hat (nach einer Personalausweiskontrolle!) vor der Prüfung zunächst einiges zum Prüfungsablauf gesagt (z.B. daß ich gerne bei Unklarheiten nachfragen darf). Während der Prüfung hat sie sehr oft bestätigende Bemerkungen geäußert („korrekt“, „völlig richtig“) oder augenzwinkernd Anmerkungen hinzugefügt.

Im folgenden sind die Prüfungsfragen nur knapp wiedergegeben. Die Antworten sind nur stichwortartig gefaßt.

Mertsching (M.): Wie kann man arithmetische Operationen in Hardware umsetzen?

Ich (I.): Addition als Schaltnetz realisieren; Subtraktion, Multiplikation und Division auf Addition zurückführen

M.: Wie wird die Subtraktion genau auf die Addition zurückgeführt?

I.: Zweierkomplement (alle Bits invertieren, 1 addieren; Darstellungsart für negative Zahlen) des Subtrahenden bilden und dazuaddieren: Übertrag streichen

M.: Wie wird das Zweierkomplement allgemein gebildet? Was bedeuten die Variablen?

I.: Formel aufgeschrieben (s. Skript) und erklärt

M.: Kann man den Übertrag bei der Subtraktion immer streichen? Was muß man dabei noch beachten? Wie geht das?

I.: immer streichen; Overflow- und Underflowprüfung; durch Prüfung der höchstwertigen Bits

M.: Wie kann man negative Zahlen noch darstellen?

I.: durch Vorzeichen und Betrag (M.: Braucht man dafür mehr Bits als fürs Zweierkomplement? I.: nein, aber 2 Darstellungsarten für die 0); durch Einerkomplement

M.: Kennen Sie noch andere Möglichkeiten?

I.: nein (*im Skript sind doch nur die 3 genannten!*)

M.: Ok. Wozu braucht man Gleitkommazahlen? Wie kann man Gleitkommazahlen allgemein darstellen?

I.: um den Bereich der darstellbaren Zahlen zu erweitern, ohne aufwendige arithmetische Schaltnetze bauen zu müssen; Darstellung: $z = s \cdot m \cdot b^e$

M.: Was haben Sie hierbei aber schon vorausgesetzt? Was muß noch beachtet werden?

I.: der Exponent ist hier nur positiv (also Charakteristik); die Zahl muß noch normiert werden ($0,1 \leq m < 1$)

M.: Wie ist der IEEE-754-Standard aufgebaut?

I.: 32-Bit- oder 64-Bit-Variante (Single- bzw. Double-Precision); bei SP 1 Bit Vorzeichen, 8 Bit Charakteristik und 23 Bit Fraktion

M.: Wie addiert man zwei Gleitkommazahlen? Nehmen Sie z. B. $2 + 4$.

I.: 2 und 4 normiert aufgeschrieben; erklärt: erst Exponenten angleichen (Komma verschieben), dann addieren; Ergebnis normieren und runden (mit Guard-, Roundstelle und Sticky-Bit)

M.: Was ist im IEEE-754-Standard noch enthalten?

I.: Sonderfälle für 0, nicht-normalisierte Zahlen, Overflow, NaNs; Rechnen mit unbestimmten Ausdrücken

M.: auch noch die Rundung. Sprung zur Informationstheorie: Was sind Entscheidungsgehalt, Informationsgehalt, mittlerer Informationsgehalt, Redundanz?

I.: Formeln erläutert: $H_0 := \log\left(\frac{1}{p}\right)$, $I_i := \log\left(\frac{1}{p(x_i)}\right)$, $H := \sum_{i=1}^n p(x_i) \cdot I_i$, $R := L - H$ (mit L als mittlerer Codewortlänge)

M.: Ok. Themenwechsel: Wie funktioniert die Netzwerkanalyse im Zeit- und Frequenzbereich?

I.: Zeitbereich: aufstellen der Netzwerkgleichung, das wird eine DGL, deren Grad von der Anzahl der unabhängigen Netzwerkelemente abhängt (auf Nachfrage: 1. oder 2. Grades (\Rightarrow falsch, allgemein n-ten Grades, aber in der Vorlesung waren wir auf den 1. und 2. Grad beschränkt!), lineare DGL); im Frequenzbereich: Transformation der Netzwerkgleichung in den Frequenzbereich durch Ergänzung von Real- oder Imaginärteil, Differentiation wird zu Multiplikation mit $j\omega$, Integration zur Division durch $j\omega$.

M.: Was ist sonst von Vorteil (\Rightarrow Verzicht auf Zeitabhängigkeit habe ich vergessen)? Wie geht die Rücktransformation?

I.: Real- oder Imaginärteil bilden (je nachdem, welcher bei der Hintransformation zugefügt wurde)

M.: Wie wird das Verfahren (Aufstellung der Netzwerkgleichung, Transformation) weiter vereinfacht?

I.: durch Betrachtung der einzelnen Netzwerkelemente: Widerstands- und Leitwertoperatoren

M.: Geben Sie die Strom-Spannungsbeziehung für Widerstand, Kondensator und Spule an.

I.: Impedanzen aufgeschrieben: $\underline{Z} = \frac{u}{i}$ (Widerstand), $\underline{Z} = \frac{1}{j\omega C}$ (Kondensator), $\underline{Z} = j\omega L$ (Spule)

M.: Das haben Sie sich gut angeschaut. (Nach einleitender Erklärung) wie ist denn Speicher aufgebaut? Wie wird er angesteuert?

I.: Meinen Sie Speicherzellen oder –bausteine? (M.: Speicherzellen) es gibt statische und dynamische Speicherzellen; man braucht bei beiden eine Wortleitung zum Ansprechen der Zelle und eine (bzw. zwei) Bitleitungen zum Lesen und Schreiben

M.: Wie sieht eine SRAM-Zelle aus? Vor- und Nachteile?

I.: MOS-SRAM aufgezeichnet (s. Skript); Vorteil: kein Refresh nötig; Nachteil: man benötigt 6 Transistoren, also niedriger Integrationsgrad

M.: Wie sieht eine DRAM-Zelle aus?

I.: aufgezeichnet, Informationsverlust beim Auslesen erläutert

M.: Und wenn man den Arbeitsspeicher eine Stunde lang in Ruhe läßt? Wie oft muß aufgefrischt werden?

I.: der Kondensator entlädt sich; typische Refreshzykluszeit: 1 ms (M.: naja, alle 10 ms (*Skript: 1 ms*))

M.: Wie ist ein von-Neumann-Rechner aufgebaut?

I.: Rechenwerk, Leitwerk, E/A-Prozessor, Speicher, Bus

M.: Wie wird ein Programm abgearbeitet?

I.: 3 Phasen: Holphase (Programmzähler (im Leitwerk) enthält Adresse des Befehls, Befehl ins Instruktionsregister (auch im Leitwerk) laden), Decodierphase (Auswerten des Opcodes und Nachladen der Operanden (entweder sind sie im Adreßfeld oder im Speicher)), Ausführungsphase

M.: Wo stehen die Operanden genau?

I.: abhängig von der genauen Realisierung; im klassischen von-Neumann-Rechner im Registerblock (Rechenwerk)

M.: Wird jeder Befehl auf ALU-Ebene ausgeführt, z.B. ein möglicher Quadratwurzelbefehl (*vgl. Skript*)?

I.: nein, man benutzt dann Mikroprogrammierung (erläutert)

M.: Wechselwirkt das Mikroprogramm mit der ALU? Was sind das Statusregister und Flags?

I.: ja, der Statusvektor liegt als Teil des Zustandsvektors am Mikroprogrammsteuerwerk an; Statusregister: spezielles Register des Rechenwerks, das in Abhängigkeit des ALU-Ergebnisses gesetzt wird; Flags: Bits des Statusregisters (z.B. Z-Flag, N-Flag)

M.: Wie reagiert das Programm auf das Statusregister?

I.: Mikro- oder Makroprogramm? (M.: beide!) Mikroprogramm muß so codiert sein, daß der Zustandsvektor in Abhängigkeit vom Statusregister auf den entsprechenden nächsten Befehl verweist; im Makroprogramm kann durch bedingte Sprünge auf Flags reagiert werden

M.: Was für 2 grundsätzliche Möglichkeiten gibt es beim Prozessorentwurf?

I.: RISC- und CISC-Prozessoren (erläutert)

M.: Welche Erweiterungen des von-Neumann-Rechners gibt es?

I.: Interrupts, Erweiterungen der Busse, (nach kurzem Überlegen, währenddessen ergänzt sie: größere Adreßwortbreite) Speicherhierarchie

M.: Wie sieht die aus?

I.: Cachespeicher, Hauptspeicher, Hintergrundspeicher (Festplatten, als „neueres Medium“ CD-ROM), Archivspeicher

M.: Betrachten wir den Cache. Was ist das für Speicher? Wozu braucht man den?

I.: schneller Zwischenspeicher (Befehle und Daten); um die relativ langsamen Hauptspeicherzugriffe seltener durchführen zu müssen

M.: Realisiert man den mit SRAM oder DRAM? Warum?

I.: mit SRAM, weil schneller (z.B. in schneller ECL-Technik); Diskussion (=> nur schneller, weil für SRAMs kein Refresh nötig ist!)

M.: Kommen wir zur Abtastung (*zeichnet ein aperiodisches Signal $u(t)$*). Was muß man dabei beachten?

I.: das Abtasttheorem $f_a \geq 2 \cdot f_g$; (M.: nach wem benannt?) benannt nach Shannon

M.: Wie kann man die Frequenzen des Signals untersuchen? Formel?

I.: man muß das Signal in den Frequenzbereich transformieren; da es sich um ein aperiodisches Signal handelt,

benutzt man dann die Fouriertransformation; Formel $\underline{F}(j\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$

M.: jetzt haben Sie $u(t)$ zu $f(t)$ gemorpht, aber egal; wie sieht das qualitativ als Graph aus?

I.: skizziert, aber dabei schon das Frequenzspektrum periodisch gezeichnet

M.: nein, das Frequenzspektrum tritt nur im Basisband auf, erst nach der Abtastung ist es periodisch. Was ist ω ? Wo ist f_g ? Was passiert bei Nichtbeachtung des Abtasttheorems?

I.: ω ist die Kreisfrequenz ($\omega = 2\pi f$, $f = \frac{1}{T}$); f_g zeigt (s. Skript), Aliaseffekte erläutert (Überlappung

der periodischen Abtastspektren, statt der tatsächlich vorhandenen Frequenz wird eine niedrigere gemessen, Zeitbereich nicht rekonstruierbar)

M.: Ok. Erläutern Sie das TCP/IP-Referenzmodell.

I.: zentrale Protokolle: IP als Vermittlungsprotokoll, TCP als Transportprotokoll (Ende-zu-Ende-Sicht), darauf aufbauend diverse Anwendungsprotokolle (Beispiele auf Nachfrage: HTTP, FTP, SMTP,...)

M.: Was liegt unter dem Internetprotokoll?

I.: keine verbindlichen Protokolle auf der Bitübertragungsschicht und Sicherungsschicht (einfach Host-an-Netz-Schicht)

M.: Nehmen wir eine Twisted-Pair-Verbindung zwischen 2 Rechnern; wie kann IP darauf zugreifen?

I.: mit Bitübertragungs- und Sicherungsprotokoll (z.B. Ethernet und HDLC)

M.: HDLC ist aber ein typisches OSI-Protokoll; ich meinte die Medienzugriffsschicht (*MAC-Teilschicht, steht (noch) nicht im Skript, Anm.!*). Was ist Ethernet?

I.: Rechnerverbindung über Bustopologie, CSMA/CD als Kollisionskonfliktlösungsalgorithmus: „horchen“, senden wenn Bus frei, Zurückziehen bei Kollision und zufällige Zeit warten (*steht auch nicht im Skript!*)