Vorlesungsmodul Rechnerarchitektur 1
- VorlMod ReArch1 -

Matthias Ansorg

30. September 2002 bis 26. Mai 2003

Zusammenfassung

Studentische Mitschrift zur Vorlesung Rechnerarchitektur 1 bei Prof. Dr. Lutz Eichner (Wintersemester 2002/2003) im Studiengang Informatik an der Fachhochschule Gießen-Friedberg. Diese Vorlesung ist die ehemalige Vorlesung Netz- und Schaltwerke (PO 1991) und beinhaltet gleichzeitig einige allgemeine Dinge über den Rechneraufbau (ehemals in der Vorlesung »Grundlagen der Informatik«). Für Studenten aus dem 3. Semster ist die Vorlesung eine Prüfungsleistung, für Studenten aus dem 1. Semester eine Studienleistung (da neue PO!).

- **Bezugsquelle:** Die vorliegende studentische Mitschrift steht im Internet zum Download bereit: http://homepages.fh-giessen.de/~hg12117/index.html. Wenn sie vollständig ist, kann es auch über die Skriptsammlung der Fachschaft Informatik der FH Gießen-Friedberg http://www.fh-giessen.de/FACHSCHAFT/Informatik/cgi-bin/navi01.cgi?skripte downgeloadet werden.
- **Lizenz:** Diese studentische Mitschrift ist public domain, darf also ohne Einschränkungen oder Quellenangabe für jeden beliebigen Zweck benutzt werden, kommerziell und nichtkommerziell; jedoch enthält sie keinerlei Garantien für Richtigkeit oder Eignung oder sonst irgendetwas, weder explizit noch implizit. Das Risiko der Nutzung dieser studentischen Mitschrift liegt allein beim Nutzer selbst. Einschränkend sind außerdem die Urheberrechte der verwendeten Quellen zu beachten.
- **Korrekturen:** Fehler zur Verbesserung in zukünftigen Versionen, sonstige Verbesserungsvorschläge und Wünsche bitte dem Autor per e-mail mitteilen: Matthias Ansorg, matthias@ansorgs.de.
- **Format:** Die vorliegende studentische Mitschrift wurde mit dem Programm LyX (graphisches Frontend zu LaTeX) unter Linux erstellt und als pdf-Datei exportiert. Grafiken wurden mit dem Programm xfig unter Linux erstellt und als pdf-Datei exportiert.
- **Dozent:** Prof. Lutz Eichner.
- **Verwendete Quellen:** .
- **Organisatorisches:** Die Vorlesungen finden (Mo) 14:00h und (Di) 08:00h statt; nach 2-3 Wochen finden (Di) 08:00h abwechselnd Vorlesungen und Übungen statt und werden auch Hausübungen ausgeteilt. Die Veranstaltung besteht also aus 3 SWS Vorlesungen und 1 SWS Übungen.
- **Klausur:** Die Hausübungen sind Voraussetzung zur Zulassung zur Klausur vor den Semsterferien, sie müssen also zu 100% richtig und vollständig gelöst werden. Gab es Missverständnisse über Aufgaben, sind vielleicht nur noch 20 von 24 möglichen Punkten Voraussetzung zur Teilnahme zur Klausur. Man kann sich aussuchen, ob man die Klausur vor oder nach den Ferien zu schreiben. Für die Klausur nach den Semsterferien gibt es keine Zulassungsvoraussetzungen! Hier darf jeder mitschreiben, auch wenn er keine Hausübungen abgegeben hat. Um in der Klausur die Note 4 zu erhalten, reicht es aus, die Aufgabentypen der Hausaufgaben zu beherrschen - sie machen 50% der Klausur aus. In der Klausur sind alle schriftlichen Unterlagen als Hilfsmittel zugelassen, jedoch wohl kein Taschenrechner.

Inhaltsverzeichnis

Teil I
Grundbegriffe des Rechneraufbaus

1 Grobstruktur

In einer DIN-Norm ist festgelegt, dass ein Rechner gegliedert ist in:

- Zentraleinheit
  - Main Memory: Hauptspeicher
  - CPU: central processing unit, Prozessor.
  - IO-Unit: input-output unit; Eingabe-Ausgabe-Einheit.
- Peripherie. Sie ist an der IO-Unit angeschlossen. Beispiele:
  - Bildschirm
  - Festplatte
  - Tastatur
  - Scanner

Abbildung 1: Schema einer Festplatte

Abbildung 2: Schema einer Scheibe einer Festplatte

- Drucker
- Diskettenlaufwerk
- Magnetband

2 Wichtige Speichermedien

2.1 Hauptspeicher (MM; main memory)

Der Hauptspeicher besteht aus adressierbaren Zellen zu je 1 Byte, die von 1 beginnend durchnummeriert sind. Das Besondere am Hauptspeicher ist, dass jede Zelle direkt mittels der Adresse zugreifbar ist, im Gegensatz z.B. zu Festplatten. Deshalb heißt er RAM (random access memory): Speicher mit wahlfreiem Zugriff, Direktzugriffsmedium.

2.2 Magnetbandspeicher

Ein Sequentielles Zugriffsmedium; um ein Byte zu lesen, müssen erst alle Vorgängerbytes gelesen werden (sog. sequentielle Zugriffsart). Es wird jedoch nicht ein einzelnes Byte geschrieben oder gelesen, sondern Blöcke und Blocksequenzen von Bytes. Bei Backups werden Blöcke ohne Pause geschrieben, deshalb auch der Name »Streamer«.

2.3 Festplatte

Eine Festplatte besteht aus Zylindern (konzentrische Kreise aus allen Scheiben gleichzeitig) und Sektoren (Kreissektoren aller Scheiben gleichzeitig). Die Schnittmenge eines Zylinders und eines Sektors ist ein Block (fälschlich oft als Sektor bezeichnet). Jeder Block hat i.d.R. 512 Byte (zusätzlich 2 Byte zur Fehlerkorrektur). Die Blocks sind durchnummeriert und direkt adressierbar; man kann jedoch nicht ein einzelnes Byte in einem Block direkt adressieren. Aus der Blocknummer kann man Sektor, Zylinder und Scheibe erhalten, außerdem ob der Block sich auf Ober- oder Unterseite der Scheibe befindet. Alle Lese-/Schreibköpfe sind über eine Nummer auswählbar und werden gleichzeitig bewegt.

Eine Festplatte mit 3 Scheiben wie in Abbildung 1 hat 6 Oberflächen und daher auch 6 auswählbare Lese-/Schreibköpfe (engl. heads). Jeder Zylinder einer Scheibe wie in Abbildung 2 besteht daher aus 6 übereinanderliegenden Kreisen. Eine Diskette hat 16 Zylinder, die Zylinderzahl heutiger Festplatten wird nicht mehr korrekt angegeben, sondern nur noch so dass die Blockberechung durch den Computer korrekt möglich ist.

Ein Block ist durch die CHS-Adresse eindeutig festgelegt: sie besteht aus 3 Zahlen, von denen eine die Zylindernummer (C, cylinder), eine die Kopfnummer (H, head) und eine die Sektornummer (S, sector) angibt. Die Software spricht die Festplatte meist über Blocknummern an, die Festplatte kann diese in die CHS-Adresse umrechnen. Mittels der CHS-Adresse ist die Festplatte ein Quasi-Direktzugriffsmedium: auf jeden Block kann direkt zugegriffen werden, jedoch nicht auf jedes Byte.

3 Die von Neumann-Architektur

3.1 Die Zeit vor von Neumann

1941 Zuse Z3, der erste funktionierende Rechner, auf Basis von Relais. Das Programm bestand hier aus einem achtspurigen Lochstreifen.

1944 Amerikanischer MARK1 von Aiken. Er funktionierte mit Relais und Zahnrädern und war 15m lang. Er beherrschte etwa dasselbe wie die Z3 war jedoch aufgrund der verwendeten Dezimalrechnung 15m lang. Er wurde mit einem 24-spurigen Lochstreifen programmiert.

Abbildung 3: Zentraleinheiten der Rechner vor von Neumann

Abbildung 4: Blockdiagramm der Zentraleinheit der von Neumann-Architektur

1946 ENIAC von Eckert und Mauchly. Funktionierte auf Basis von Röhren und wurde mit Steckern, Kabeln und Schaltern programmiert.

Die Zentraleinheiten der obigen Rechner waren entsprechend Abbildung 3 strukturiert. Das Programm war mit dem Steuerwerk verbunden: ein Lochstreifen wurde manuell durch einen Abtaster geführt, bei Abfühlen eines Loches wurde eine entsprechende Operation vorgenommen. Es gab eine externe Programmsteuerung und Programmierung. Das bedeutete, dass man keine Schleife und keine Unterprogramme programmieren konnte, weil ein Sprung im Programm durch das Programm selbst ja nicht möglich war.

3.2 von Neumann-Architektur

Zur Person:Historisches zur Person :

- Johann (Janos) von Neumann (1903-1957), Ungare
- studierte in Budapest und Zürich
- Universalgenie (Spieletheorie, Quantenmechanik, Beweistheorie, Informatik)
- definierte 1947 das Prinzip der universalen speicherprogrammierbaren Rechenmaschine

Er schaffte die überflüssige dezimale Darstellung im Rechner zugunsten der binären ab. Er fand heraus, dass man Programme auch im Hauptspeicher darstellen kann (Loch entspricht 1, kein Loch entspricht 0), wo auch die Daten dargestellt sind. Mit einem Sprungbefehl sind dann Schleifen möglich! Dies ist die »interne Programmierung«. Die CPU ist jetzt die Zusammenfassung von Operationswerk und Steuerwerk.

Die von Neumann-Architektur wurde zuerst realisiert im EPAC (1946).

3.3 Harvard-Architektur

Eine Weiterentwicklung der von Neumann-Architektur, bei der Programm und Daten in zwei verschiedenen Hauptspeichern abgelegt werden. So wird der sog. »von Neumann-Flaschenhals« vermieden, d.h. die Geschwindigkeitsbegrenzung durch Übertragung von Daten und Programmen sequentiell über einen einzigen Systembus.

3.4 RISC und CISC

Ebenfalls eine Weiterentwicklung der von Neumann-Architektur. Während CISC[1]-Rechner der menschlichen Denkweise folgen - möglichst viel mit wenigen Befehlen machen können - und deshalb viele, aber langsame Befehle realisieren, folgen RISC[2]-Rechner dem, was für Computer selbst günstig ist: wenige, schnelle Befehle.

4 Betriebssystem

Das erste Betriebssystem wurde von General Motors erfunden! Bisher wurden Programme auf Lochkarten geschrieben und mit Lochkartenlesern in den Hauptspeicher eingelesen. Dabei wurden die binären Operationscodes geschrieben, was für Menschen natürlich sehr fehlerträchtig ist. Jemand erfand also bald Abkürzungen für diese Maschinenbefehle, d.h. den Assembler, und ein Programm, den Assembler, das die Assembler-Programme assemblierte, also in Maschinensprache umsetzte.

Die Programme wurden jetzt also in Assembler gelocht. Nun wurde der Assembler im Hauptspeicher abgelegt, dann das in Assembler geschriebene Programm eingelesen und vom Assembler direkt in Maschinencode übersetzt und so in Lochkarten gestanzt. Als die höheren Programmiersprachen entwickelt wurden, wurde mit einem analogen Verfahren COBOL bzw. FORTRAN in Assembler übersetzt und dann Assembler in Maschinensprache usw.

[1] complex instruction set computer
[2] reduced instruction set computer

Abbildung 5: Interrupt-Handling im PC

General Motors erfand zuerst Kartensortierer, um Stapel von durcheinandergekommenen, aber nummerierten Lochkarten zu sortieren. Dann erfand General Motors ein Programm, das von einem Magnetband je nach Bedarf einen Assembler oder einen Compiler lud. Dies nannten sie Monitor-Programm. Es war das erste Betriebssystem.

Das Betriebssystem (BS; engl. operating system, OS) ist ein Programmpaket mit folgenden Eigenschaften:

- ermöglicht eine bequeme und wirtschaftliche (d.i. effiziente) Nutzung der Hardware.
- es ist die Schicht zwischen Benutzer und Benutzerprogrammen einerseits und der Hardware andererseits. Dies ist eine sehr weitgehende Abstraktion von der Hardware: das Betriebssystem setzt etwa einen Befehl »Drucke `datei`« in die entsprechende Steuerung der Hardware um.
- es versteckt Einzelheiten der Hardware und stellt dem Benutzer abstrakte Zugriffsmöglichkeiten auf die Peripherie zur Verfügung. Beispiel: Drucken einer Datei auf Mausklick - der Benutzer muss nicht wissen, um welchen Drucker es sich handelt.
- Jedes Peripheriegerät besitzt ein gerätespezifisches Zugriffsprogramm, den sog. (Geräte-)Treiber, der das Schreiben und Lesen der Daten auf diesen Geräten ermöglicht. Durch Steuerleitungen wird festgelegt, ob mit einer Adresse ein Gerät oder der Hauptspeicher angesprochen wird. Beispiel Textverarbeitung: Drücke eine Taste der Tastatur. Das Betriebssystem holt das Zeichen und gibt es auf dem Bildschirm aus. Vergleiche dazu Abbildung 5 - hier bedeuten die Abkürzungen:

  PIC programmable interrupt controller. Er besitzt 16 Eingänge, sog. IRQ-Eingänge (interrupt request), nummeriert von 0 bis 15. An jedem IRQ ist ein Gerät angeschlossen.

  INT Interrupt-Steuereingang des PC

  OS Bereich des Hauptspeichers, in dem die wichtigsten Routinen des Betriebssystems permanent gespeichert sind.

  Prinzipieller Ablauf im Computer nach einem Tastendruck:

  1. Die Tastatur mach einen interrupt request, indem sie ihre Leitung Nr. 1 zum PIC auf logisch 1 setzt. Es handelt sich nur um eine Anforderung, weil der PIC ja gerade mit einer anderen Interruptbehandlung blockiert sein könnte.
  2. Der PIC setzt die Leitung INT (Interrupt-Eingang der CPU auf logisch 1).
  3. Die CPU unterbricht die Arbeit und ruft im Betriebssystem den interrupt handler (IH) auf.
  4. Der interrupt handler holt im PIC ein Byte ab und weiß damit, wer die Unterbrechung ausgelöst hat.
  5. Der interrupt handler ruft den Tastaturtreiber auf. Dieser holt das eingegebene Zeichen von der Tastatur ab, legt es an eine bestimmte Stelle im Hauptspeicher und ruft anschließend den Bildschirmtreiber auf.
  6. Der Bildschirmtreiber gibt das Zeichen auf dem Bildschirm aus.

Bemerkung Das Betriebssystem verwaltet die Hauptspeicher- und CPU-Nutzung der Programme. Beispiel: Um die CPU effizient auszunutzen, befinden sich mehrere Programme geichzeitig im Hauptspeicher. Während ein Proramm z.B. auf eine Eingabe des Benutzers oder Daten von der Festplatte wartet, wird einem anderen Programm die CPU zugeteilt (z.B. Ausdrucken »im Hintergrund«). So können z.B. Großrechner mit 500 Terminals gebaut werden. Programme befinden sich i.d.R. nicht vollständig im Hauptspeicher; Teile befinden sich auf der Festplatte und werden bei Bedarf in den Hauptspeicher nachgeladen. Davon merkt der Benutzer nichts.

Abbildung 6: Typisches Layout eines Motherboards

Zusammenfassung: Die Aufgaben des Betriebssystems

- Es verwaltet die Betriebsmittel (Hauptspeicher, CPU, Festplatte, ...).
- Es steuert die Kommunikation zwischen den Funktionseinheiten (Auflösung von Konflikten bei gleichzeitigem Ressourcenzugriff usw.).
- Es verwaltet Benutzerdaten und Programme (in sog. Dateien, kurz für Datenkartei, in der früher die Lochkartenstapel abgelegt wurden). Benutzerauthentifizierung verhindert Zugriff auf die Daten anderer Benutzer auf demselben Rechner.
- Es stellt dem Benutzer eine bequeme (abstrakte) Oberfläche des Rechners zur Verfügung. Abstraktion bedeutet hier, dass der Benutzer sich mit möglichst wenig Einzelheiten des Rechners beschäftigen muss, d.h. dass er den Rechner einfach bedienen kann. Zu dieser Abstraktion gehört die Erfindung der GUI, zuerst in SMALLTALK und dann auf dem Apple Macintosh realisiert.

5 Einführung in die Rechnerhardware

5.1 Mainboard

Eine große Platine, die kleinere Platinen (Karten, auch Boards) enthält. Die Spieleindustrie ist heute der Hauptgrund, dass so oft schnellere Computer und schnellere Speicher auf den Markt kommen.

In Abbildung 6 verwendete Abkürzungen:

AGP Accelerated Graphics Port. AGP-Grafikkarten besitzen einen eigenen Prozessor und einen eigenen Speicher (Bildspeicher, Videoram), bilden also einen eigenen kleinen Computer, der die CPU entlastet. Der Bildspeicher wird bis 100mal pro Sekunde ausgelesen, was einer Bildwiederholrate von 100Hz entspricht.

PCI Peripheral Component Interconnect. Anschlüsse für moderne Peripherie. Im Prinzip kann hier alles angeschlossen werden, z.B. Grafikkarten (anstelle AGP, evtl. zusätzlich), Soundkarten, TV-Karten, Modems, Netzwerkkarten, SCSI-Adapter (für SCSI-Festplatten usw.), Scanner usw..

AT / ISA Advanced Technology bzw. Industry Standard Architecture. Anschlüsse für die alte Peripherie des IBM PC-AT und Nachbauten, genannt Legacy[3]-Devices. Diese Slots sind zweigeteilt; der PC-XT hatte nur den unteren Teil (XT-Slot mit 8 Bit Busbreite), der PC-AT hatte zusätzlich die AT-Ergänzung. Die Abkürzung dieses PC wird heute noch in vielen Gerätebezeichnungen verwendet: ATX-Board, ATAPI-CD-ROM, ATA-Festplatte. ISA ist die alte IBM-AT-Architektur mit von der sog. ISA-Gruppe (Compaq u.a.) definierten elektrischen Eigenschaften.

Bus	Takt	Breite
AGP	$66,\overline{6}MHz$	$32Bit$
PCI	$33,\overline{3}MHz$	$32Bit$
ISA	$8,\overline{3}MHz$	$16Bit$

Busbreite hier die Datenbusbreite, die Anzahl der gleichzeitig übertragenen Datenbits. In der Regel dauert eine Datenübertragung einen Bustakt. Es gibt im Serverbereich auch PCI-Varianten mit $66,\overline{6}MHz$, $100MHz$ und $64Bit$ Busbreite.

BIOS Basis Input Output System. Ein ROM-Baustein. Das BIOS enthält einfache Routinen, die die Peripherie beim Hochfahren des Systems ermitteln und dann versuchen, das Betriebsystem von Diskette oder Festplatte, CD-ROM oder Netzwerk zu booten. Moderne BIOS sind als Flash-BIOS ausgeführt, die enthaltenen Routinen können also als Software ersetzt werden.

DIMM Dual Inline Memory Module. Dies sind Speichermodule. Die Anschlüsse heißen Kamm. Auf einem Modul sind 8-9 Speicherchips enthalten und ein winziger SPD-Chip (Serial Presence Detect). Aus diesem Chip liest das BIOS die Speicherdaten aus.

[3] Erbschaft; hier im Sinne von »Altlast, schweres Erbe«.

SIMM Single Inline Memory Module. Sog. PS/2-SIMM, ein Speichermodul bei dem im logischen Sinn nur eine Seite besetzt ist. DIMMs kombinieren zwei solcher Module auf einer Platine.

MM Main Memory. Der Prozessor kann 4 GB adressieren; Linux kann diese 4 GB nutzen, Windows 2000 oder XP max. 2 GB, Win9x max. 512 MB.

CPU Central Processing Unit.

DIP Dual Inline Package[4]. Kleine Schalter zur Konfiguration der Multiplikatoren für Bustakt, CPU-Takt usw.. Der Grundtakt ist dabei der Frontsidebus-Takt (Takt des Busses zwischen CPU und North Bridge). Die Aufgabe der DIPs wird heute oft durch das BIOS übernommen.

NB North Bridge. Bei Intel MHC: Memory Controller Hub. Ein Schnittstellenbaustein, der die gesamte Kommunikation der CPU mit den anderen Komponenten steuert. Früher als Buscontroller bezeichnet. Die Bezeichnung MCH zeigt, dass es sich hier um einen Sternverteiler[5]handelt, der u.a. die Speichersteuerung enthält.

SB South Bridge. Bei Intel ICH: Input Output Controller Hub. Schnittstellenbaustein zur Peripherie.

EIDE Manchmal auch IDE, eigentlich der frühere Name: Intelligent Drive Electronics. Diese Steuerungselektronik war früher im PC, jetzt auf der Festplatte eingebaut. Anschluss für die Festplatten. DOS konnte nur 2 Festplatten und 512 MB pro Festplatte verwalten; der entsprechende Standard IDE wurde also geändert, so dass 4 Festplatten mit mehr Platz verwaltet werden konnten: EIDE, Extended Integrated Drive Electronics. Eine weitere Bezeichnung ist Fast ATA: Fast Advanced Technology Atachment. An den EIDE-Controller können pro Stecker 2 Festplatten und ATAPI-Geräte (CD-ROM, Brenner, DVD, Tape) angeschlossen werden, insgesamt also 4 Geräte. Die IDE-Schnittstellen waren zu Anfang nur für Festplatten gedacht. Früher gab es den SCSI-Bus für Großrechner - jedes Gerät wie Scanner oder Drucker wurden zuerst für SCSI gebaut, waren also für die meisten PC-Besitzer nicht verfügbar. Man nahm also SCSI-Geräte, entfernte Intelligenz und fügte Verpackung hinzu, um sie an den IDE-Anschluss anzuschließen: AT Atachment Package Interface.
Ein großer Nachteil von IDE ist: die Geschwindigkeit der Geräte an einem IDE-Kabel richtet sich immer nach dem langsamsten Gerät. Hat man nur eine Festplatte und ein CD-ROM, schließt man diese natürlich an unterschiedliche Controller an. Weiterer Nachteil: Der Hauptspeicher wird immer als Puffer benutzt, auch wenn zwischen zwei EIDE-Geräten kopiert wird, selbst am gleichen Kabel. SCSI-Geräte können dagegen als Bus-Geräte direkt miteinander kommunizieren, so dass bei Datensicherungen weder Hauptspeicher noch Rechenleistung blockiert werden.

AT Die Buchstaben AT in vielen Bezeichungen wie ATAPI, Fast ATA stammen vom PC AT (»Advanced Technology«) von IBM.

SCSI Small Computer System Interface. Gesprochen »SCASI«; erfunden von Sugart Association als SASI; die Firma wurde aufgekauft und der Bus in SCSI umbenannt, die Amerikaner sprachen es in Anlehnung an SASI aber SCASI aus. Um mehr als 4 Festplatten im PC anzuschließen, verwendet man einen oder mehrere SCSI-Controller in PCI-Slots. Man kann auch einen RAID-Controller verwenden, woran allerdings nur Festplatten angeschlossen werden können. IDE, SCSI und RAID können gleichzeitig in einem System vorhanden sein.
SCSI ist ein Bussystem, genauer ein Netzwerk. Es gibt SCSI in 8, 16 und 32 Bit - hier können 7, 15 und 31 Geräte angeschlossen werden. Je mehr Geräte man anschließt, desto kürzer müssen die verbindenden Kabel sein - deshalb ist es oft besser, mehrere SCSI-Controller zu verwenden. SCSI-Geräte können sein: Drucker, Scanner, CD-ROM, anderer Computer.

RAID Redundant Array of Independent (früher: Inexpensive) Disks. RAID-Verfahren im PC-Bereich:

RAID0 (Stripping). Verteilung der Daten worterise auf verschiedene Festplatten. Bei einer Anforderung liest eine Festplatte zuerst Daten in den Cache, dann wird der Cache ausgelesen, während die andere Platte Daten in den Cache liest. So verdoppelt sich scheinbar die Geschwindigkeit des Plattenzugriffs durch »überlappenden Zugriff«. Bei der Anforderung von Daten liest die Platte die Datenblöcke zuerst in den Festplatten-Cache, die daraus schneller als von der Platte selbst in den Hauptspeicher transportiert werden können. Dies geschieht auf Befehl des Prozessors direkt, denn DMA-Festplatten können Busmaster sein.

[4]Dual bezeichnet hier die Form des Bausteins, nämlich einen Chip mit zwei Reihen Beinchen.
[5]engl. hub, eigtl. Radnabe, an der die Speichen montiert sind.

Abbildung 7: Topologie des USB

RAID1 (Mirroring). Ein System, das die gleichen Daten auf zwei oder mehr Festplatten gleichzeitig schreibt (»spiegelt«) und so die Datensicherheit erhöht. Die Sicherheit besteht in der geringen Wahrscheinlichkeit, dass beide Platten gleichzeitig kaputtgehen.

VLB Vesa Local Bus. Alte Karten, die einen ISA-Steckplatz und eine VLB-Verlängerung belegen. VLB war ein direkter Anschluss an die CPU, es wurden einige Steuerleitungen von ISA mitverwendet.

Boardlayout Es gibt im Wesentlichen zwei Motherboard-Typen:

AT Auch in der kleineren Ausführung BAT (Baby AT). Rein äußerlich daran erkennbar, dass die CPU in der Verlängerung der Slots platziert ist - die Karten durften deshalb eine bestimmte Länge nicht überschreiten. Deshalb wurde ATX entwickelt.

ATX Die CPU sitzt neben den Slots. Neben dieser äußerlichen Änderung wurden hpts. Stromspar- und Stromsteuerungsfunktionen eingebaut, darunter Temperaturfühler für die CPU. Diese Boards passen nicht in die AT-Gehäuse.

Externe Schnittstellen des ATX-Boards

USB universal serial bus. Eine Schnittstelle, die nahezu alle anderen Schnittstellen ersetzen soll. Typisch für USB ist, dass man ein Gerät an das andere anschließen kann. USB ist von der Topologie[6] her kein Bus, sondern ein hierarchischer Baum (vgl. Abbildung 7). Dazu gibt es an der Wurzel des Baums einen Steuerbaustein (»root«-Chip), der alle Geräte im Baum abfragt. An einen USB-Port können Geräte und Hubs angeschlossen werden; Geräte können auch gleichzeitig Hubs sein. Es gibt USB in 2 Geschwindigkeiten:

USB1.1 12MBit/s

USB2.0 480MBit/s. Verfügbar seit Ende 2002. Diese Geschwindigkeit reicht aus, um auch Festplatten usw. daran anschließen zu können.

FireWire Auch IEEE 1394 genannt. Geschwindigkeit bisher 400MBit/s, genormt auch für höhere Geschwindigkeiten. Zu FireWire gibt es im Gegensatz zu USB schon viele Geräte, auch externe Festplatten. FireWire ist eine Standardschnittstelle bei Apple-Computern und damit die Konkurrenz zu USB von Intel. Hinweis: Der ilink an Sony-Digitalkameras ist FireWire.

LAN local area network. Meist ein Anschluss für ein Netzwerk mit Ethernet-Topologie. Das bedeutet: Alle Geräte sind an den Ethernet-Bus angeschlossen, der Ethernet-Bus besteht aus einem einzigen Kabel. Heute bringt man den gesamten Bus in einem Hub unter und schließt die Geräte alle an den Hub an. Switches (aktive Hubs) verhindern z.B. Kollisionen.

COM communication interface. Serielle Schnittstelle. Werden in naher Zukunft wohl nicht mehr vorhanden sein. Alte COM-Schnittstellen haben 25 Pins, neue 9 Pins. Bisher wird hier das externe Modem angeschlossen.

LPT line printer. Die Druckerschnittstelle, auch Parallelschnittstelle genannt.

Joystick Eine Schnittstelle mit 15 Löchern, auch Game Port.

Sound Es können bis zu 5 Aus- und Eingänge mit 3,5mm-Klinkenstecker vorhanden sein.

5.1.1 Arithmetik

Der Prozessor des PC beherrscht 3 Arten von Arithmetik:

- Integer-Arithmetik. Hauptsächlich gedacht für die Adressrechnung im Betriebssystem. In C++ durch den Datentyp int verfügbar.
- Fließkommaarithmetik. In C++ durch den Datentyp float verfügbar.
- Festkommaarithmetik (Dezimalzahlenarithmetik). Wichtig für kaufmännische Anwendungen, in C++ nicht verfügbar, jedoch in COBOL.

[6] Geometrische Struktur eines Netzwerks.

Abbildung 8:

6 Historie der PC-Entwicklung

1981 IBM PC (»personal computer«) mit einem 8086-Prozessor (16Bit-Prozessor), später einem 8088-Prozessor (8Bit-Prozessor, intern 16Bit-Prozessor). Die CPU kontte also intern maximal 16Bit lange Integerzahlen verarbeitet werden. Der Prozessortakt betrug max. 10MHz.
Daten (beim 8086 16Bit breit, beim 8088 8Bit breit) und Adressen (20Bit breit) wurden gemuxt (»gemultiplext«), d.h. abwechselnd über die gleiche Leitung übertragen, so dass 16 der 20 Adressleitungen auch für Datenübertragung verwendet wurden.
Mit 20 Adressbits konnte maximal 2^{20}Byte = 1MByte Hauptspeicher adressiert werden. Dieser erste PC hatte keine Festplatte, sondern ein Magnetband oder ein Diskettenlaufwerk.

1981 Die ersten Grafikkarten für den PC, eine von IBM und eine von Hercules.

1983 IBM PC-XT (»extended technology«). Erweiterungen gegenüber dem IBM PC waren Diskettenlaufwerk und Festplatte standardmäßig. Der XT-Steckplatz (größerer Teil des ISA-Steckplatzes) existiert bis heute - er ist nur 8Bit breit, denn die Peripherie wurde bisher nur mit 8Bit bedient.

1984 IBM PC-AT (»advanced technology«) mit einem 80286-Prozessor. Dies war ebenfalls ein 16Bit-Prozessor, Daten (16Bit breit) und Adressen (24Bit breit) waren jedoch getrennt. Dadurch konnten Daten und Adressen zum Abspeichern in den Hauptspeicher gleichzeitig übertragen werden! Der 80286 war mit bis zu 25MHz deutlich schneller als der 8086.
Mit 24 Adressbits konnte maximal $2^{24}\text{Byte} = 2^4 \cdot 2^{20}\text{Byte} = 16\text{MByte}$ Hauptspeicher adressiert werden. Die elektrischen Eigenschaften des PC-AT wurden erst 1990 von der ISA-Gruppe um Compaq (ohne IBM) genau spezifiziert; der Bus heißt seitdem auch ISA-Bus. Er wurde mit $8,\overline{3}$MHz getaktet und hatte nun auch 16Bit Datenbusbreite bis in die Slots.

1985 Intel entwickelte den 80386-Prozessor. Das war der größte Entwicklungsschritt in der Firma Intel. Der 80386 war ein 32Bit-Prozessor. Daten- und Adressleitungen waren jeweils 32Bit breit (d.h: 4GB Adressraum), der Prozessortakt betrug bis zu $33,\overline{3}$MHz. Das Problem: die AT-Architektur war für den 80386, eine Art »Großrechner im PC« nicht geeignet. Es gab 4 realisierte inkompatible Vorschläge für eine neue Architektur:

- **1987** MicroChannel-Architektur (MCA) von IBM. Es entstand der PS/2-PC - er scheiterte, da er nicht kompatibel war mit der AT-Architektur.
- **1990** EISA (»extended ISA«) von der ISA-Gruppe. EISA ist ein 32Bit-Bus, der zu ISA kompatibel, aber zu teuer war. Bis heute wird EISA in manchen Servern verwendet.
- **1992** VLB (»VESA local bus«) von der VESA, einer Vereinigung von Bildschirm- und Grafikkartenherstellern. Dies war vorübergehend die beste Lösung, denn sie war für Grafikkarten die schnellste und beste Lösung. VLB war aber fpr Prozessoren mit mehr als $66,\overline{6}$Mhz nicht geeignet.
- **1993** PCI-Bus von Intel. Dies war eine preiswerte, zuerst zusätzliche Lösung, die die anderen inkompatiblen Architekturen allmählich verdrängte. Der PCI-Bus war so preiswert, weil er mit nur 32 Leitungen auskam, im Gegensatz zu 64 Leitungen bei den anderen Architekturen: Adressen und Daten wurden durch Burst-Fähigkeit gemultiplext. Außerdem bediente PCI nur 32Bit-Peripherie (keine Abwärtskompatibilität zu XT und ISA-Bus).

7 Kommunikation zwischen den Bausteinen des Mainboards

Definitionen

Chipsatz Bestehend aus North Bridge (bei Intel: MCH) und South Bridge (bei Intel: ICH).

Chipsatz im weiteren Sinne Alle Chips auf dem Mainboard außer Hauptspeicher und CPU.

Abbildung 9:

Kommunikation beim Chipsatz Intel i845D Dies ist ein Chipsatz für Pentium4-Prozessoren und DDR-SDRAM. Der führende Takt ist i.d.R. der FSB-Takt, alle andere Takte inkl. dem Prozessortakt werden daraus generiert. Besprechung der hier verwendeten gängigen Bussysteme:

Bus	Grundtakt	Datenbusbreite
AGP	$66,\overline{6}$MHz	32Bit
PCI	$33,\overline{3}$MHz	32Bit
ISA	$8,\overline{3}$MHz	16Bit
FSB	100MHz, 133MHz	64Bit
M-Bus	100MHz, 133MHz, 166MHz	64Bit
Rambus		16Bit, 32Bit

Der PCI-Bus existiert in sehr unterschiedlichen Varianten, nämlich in $66,\overline{6}$Mhz, 100MHz und 133MHz und mit 64Bit. Schnelle PCI-Busse haben weniger Slots, so dass man ggf. PCI-to-PCI-Bridges verwenden muss, um weitere Slots zu erhalten. Eine gängige Entwicklungsrichtung ist die Einführung serieller Bussysteme, weil hohe Bustakte bei Parallelbussen zu Problemen führen, denn die Bits kommen aufgrund von Unerschieden in der Leitungslänge zu unterschiedlicher Zeit an. Zur Geschwindigkeitssteigerung serieller Busse verwendet man Mehrkanalübertragung (mehrere serielle Kanäle gleichzeitig).

Es gibt Ausnahmen, bei denen ein Takt nicht vom FSB-Takt abgeleitet wird. Solche Takte nennt man asynchron. Die Übertragungsrate ist in der Regel ein Vielfaches des Bustaktes. Moderne Motherboards können im BIOS übertaktet werden, d.h. der FSB-Takt wird erhöht. Welcher Takt im BIOS für FSB eingestellt werden kann hängt davon ab, welche Takte der Prozessor unterstützt.

Der Memory-Bus kann heute, je nach Chipsatz, schneller sein als der FSB; das ringt für die Performance des Rechners eigentlich nichts. Es ist ein Marketingargument, das nicht aufgeht. Empfehlung: man kaufe stets das vorletzte Modell; das ist wesentlich billiger und reicht noch lange aus.

Burst-Mode Eine Blockübertragungsart, zu der alle Busse außer ISA fähig sind. Angefordert wird durch die Adresse der ersten Datums (in Busbreite), es wird daraufhin ein Block von immer 4 Daten übertragen. Empfänger und Sender zählen die Adressen selbst hoch. Beim FSB werden so also 32 Byte in einem Block übertragen. Beim Memory-Bus können die Burst-Anforderungen in bestimmten Situationen so überlagert werden, dass zwischen zwei Bursts keine zeitliche Lücke besteht (aufgrund Adressübertragung). Eine Folge von Bursts erscheint dann als Folge von Bursts in der Form 3-1-1-1-1-1-1-1-1-... . Bezeichnungsbeispiel:

3-1-1-1 Das erste Datum benötigt 3 Takte (wegen Adressübertragung), jedes weitere Datum 1 Takt.

Burstlänge Anzahl der Daten (in Busbreite), die zu einem Burst gehören. Üblich sind Bursts der Länge 4, d.h. 16 Byte bei PCI, 32 Byte bei FSB. Man muss nämlich Puffer in voller Burstlänge zur Verfügung stellen (denn Busse können parallel zueinander arbeiten und müssen ggf. Daten zwischenspeichern), was Geld kostet. Deshalb macht man Bursts nicht beliebig lang.

Übertragungsrate Jeder Bus besitzt einen Grundtakt (sog. Bustakt) und einen Übertragungstakt, der ein ganzzahliges Vielfaches α des Grundtaktes ist. Dieses α heißt Übertragungsrate.

SDR single data rate. Ein Datum (in Busbreite) je Bustakt wird übertragen, d.h. immer bei positiver Taktflanke. Beispiel: AGP 1×.

DDR double data rate. Zwei Daten (in Busbreite) je Bustakt werden übertragen, d.h. sowohl zur positiven als auch zur negativen Taktflanke. Beispiel: AGP 2×

QDR quad data rate[7]. Vier Daten (in Busbreite) je Bustakt werden übertragen, unter Verwendung eines doppelt schnellen Hilfstaktes (vgl. Abbildung 9). Beispiel: AGP 4×

Achtfache Datenrate 8×. Beispiel: AGP 8×.

[7] bei Intel: quad pumped data rate

Abbildung 10: UND, AND

Abbildung 11: ODER, OR

Bezeichnungen hinsichtlich der Übertragungsrate.

FSB

Bezeichnung	FSB-Takt	Übertragungs-Rate	unterstützende Prozessoren
FSB200	100MHz	DDR	AMD: Athlon XP, Duron
FSB266	133MHz	DDR	AMD: Athlon XP, Duron
FSB333	$166,\overline{6}$MHz	DDR	AMD: Athlon XP, Duron
FSB400	100MHz	QDR	Intel Pentium4
FSB533	133MHz	QDR	Intel Pentium4

M-Bus für DDR-SDRAM

Bezeichnung	M-Bus-Takt	Übertragungs-Rate	DIMM-Name	alternativer Name
DDR200	100MHz	DDR	DDR PC200	PC 1600
DDR266	133MHz	DDR	DDR PC266	PC 2100
DDR333	$166,\overline{6}$MHz	DDR	DDR PC333	PC 2700

Die Übertragungsrate in $MByte/s$ für DDR266 beträgt:

- M-Bus-Breite: $64Bit = 8Byte$
- $266,\overline{6}MHz \cdot 8Byte = 2133MByte/s$ (deshalb die Bezeichnung PC 2100)

Die Module für RAMBUS heißen nicht DIMM, sondern RIMM.

Teil II
Schaltnetze: Verdrahtete Logik

8 Schaltalgebra

Dies entspricht dem Stoff der mathematischen Aussagenlogik.

8.1 Grundoperationen der Schaltalgebra

8.1.1 UND

ISO-Schaltsymbol Siehe Abbildung 10

Formulierung »$a = 1$, wenn $e_0 = 1$ und «. Im englischen Sprachraum als »Produkt« bezeichnet und deshalb geschrieben als $a = e_0 \cdot e_1$.

logischer Ausdruck $a = e_0 \wedge e_1$

Tabelle

e_1	e_0	a
0	0	0
0	1	0
1	0	0
1	1	1

Abbildung 12: NICHT, NOT

8.1.2 ODER

ISO-Schaltsymbol Siehe Abbildung 11

Formulierung »$a = 1$, wenn mindestens eine 1 anliegt«

logischer Ausdruck $a = e_0 \vee e_1$. Im englischen Sprachraum geschrieben als $a = e_0 + e_1$ und deshalb auch als Summe bezeichnet.

Tabelle

e_1	e_0	a
0	0	0
0	1	1
1	0	1
1	1	1

8.1.3 NICHT

ISO-Schaltsymbol Siehe Abbildung 12. Man unterscheidet interne und externe Zustände der Ausgänge. Der »Inverterkreis« (inverter bubble) invertiert den internen Ausgang und führt zum externen Ausgang. Die internen Zustände sind in der ISO-Kastenlogik all das, was im Kasten selbst geschieht. Die elementaren Gatter in der Realität sind NAND und NOR, d.h. AND und OR bestehen aus NAND bzw. NOR und einem zusätzlichen Inverter!

Formulierung »$a = 0$, wenn $e = 1$«. NOT wird auch Komplement, Negator, Invertierer genannt.

logischer Ausdruck $a = \overline{e}$. Hier handelt es sich im Gegensatz zu AND, OR um einen unären Operator.

Tabelle

e	a
0	1
1	0

8.2 Grundgesetze der Schaltalgebra

Diese Grundgesetze sind mit Hilfe der Tabellen, mit denen die Grundoperationen der Schaltalgebra eingeführt wurden, durch Nachrechnen beweisbar.

1. Assoziativgesetze
$$a \wedge (b \wedge c) = (a \wedge b) \wedge c$$
$$a \vee (b \vee c) = (a \vee b) \vee c$$
2. Kommutativgesetze
$$a \wedge b = b \wedge a$$
$$a \vee b = b \vee a$$
3. Distributivgesetze. Es kann auch auf ganze Teilformeln angewandt werden.
$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$
4. Absorptionsgesetze
$$a \wedge (b \vee a) = a$$
$$a \vee (b \wedge a) = a$$

Abbildung 13: EXOR, XOR

5. Eigenschaften von 0 und 1
   0 ist neutrales Element bezüglich OR:
   $$a \vee 0 = a$$
   1 ist neutrales Element bezüglich AND:
   $$a \wedge 1 = a$$
   Es gibt kein inverses Element:
   $$a \wedge 0 = 0$$
   $$a \vee 1 = 1$$
6. Eigenschaften des Komplements
   $$a \vee \overline{a} = 1$$
   $$a \wedge \overline{a} = 0$$
7. Idempotenz
   $$a \vee a = a$$
   $$a \wedge a = a$$
8. Doppelte Negation
   $$\overline{\overline{a}} = a$$
9. Regeln von DeMorgan
   $$\overline{a \vee b} = \overline{a} \wedge \overline{b}$$
   $$\overline{a \wedge b} = \overline{a} \vee \overline{b}$$

8.3 Verkürzende Schreibweisen

In dieser Veranstaltung gilt: durch Vereinbarung bindet AND stärker als OR. DIN-Normen legen jedoch fest, dass AND und OR gleiche Priorität haben und deshalb die Klammern nicht weggelassen werden dürfen. In dieser Veranstaltung werden Klammern für je ein Gatter verwendet.

$$x_1x_2x_3 \vee x_2\overline{x_3}x_4x_5 \hat{=} (x_1 \wedge x_2 \wedge x_3) \vee (x_2 \wedge \overline{x_3} \wedge x_4 \wedge x_5)$$

Für diese Veranstaltung sind folgende Absorptionsgesetz von großer Wichtigkeit. Mit Rückwärtsanwendung des Distributivgesetzes gilt:

$$wa \vee w\overline{a} = w\,(a \vee \overline{a}) = w1 = w$$
$$(w \vee a) \wedge (w \vee \overline{a}) = w \vee (a \wedge \overline{a}) = w \vee 0 = w$$

8.4 Binäre Schaltzeichen nach DIN 40900 Teil 12

Die Äquivalenz heißt auch NEXOR, weil sie das negierte EXOR ist.

8.4.1 EXOR

ISO-Schaltsymbol Siehe Abbildung 11

Formulierung »$a = 1$, wenn genau eine 1 anliegt«

logischer Ausdruck $a = e_0 \oplus e_1$ Das Zeichen »$\oplus$« wird verwendet, weil XOR die binäre Addition ohne Übertrag (d.h. eine eingeschränkte Addition) ist.

Tabelle

e_1	e_0	a
0	0	0
0	1	1
1	0	1
1	1	0

8.4.2 NOR

Nach DeMorgan gilt folgende Umformung des NOR: $a = \overline{e_1 \vee e_0} = \overline{e_1} \wedge \overline{e_0}$. Somit kann ein NOR aus einem AND mit invertierten Eingängen realisiert werden.

8.4.3 NAND

Nach DeMorgan gilt folgende Umformung des NAND: $a = \overline{e_1 e_0} = \overline{e_1} \vee \overline{e_2}$. Somit kann ein NAND aus einem OR mit invertierten Eingängen realisiert werden.

8.4.4 Verallgemeinerung der Schaltzeichen

AND und OR dürfen auch mehr als 2 Eingänge haben. Schreibweise und Bedeutung des Symbols ändern sich nicht. Die Kastenlogik kann in der Art verallgemeinert werden, dass Schaltzeichen mit beliebigen Beschriftungen möglich sind, die die Anzahl der Eingänge angeben müssen, die 1 sein müssen, damit der Ausgang 1 wird. Dabei verwendet man für die Anzahl der Eingänge in solchen Beschriftungen stets den Buchstaben n, für einen Buchstaben m muss im Anwendungsfall eine feste Zahl eingesetzt werden.

Ein XOR-Element hat immer genau 2 Eingänge. Hat es mehr, bezeichnet man es als m-aus-n-Element.

Ein Paritätsbit ist ein zusätzliches Bit bei der Nachrichtenübertragung zur Fehlerkorrektur. Gerade Parität bedeutet, dass das Paritätsbit die übertragenen Bits auf gerade Anzahl von 1en ergänzt - hier wird das Paritätsbit mit einem Imparitätselement erzeugt. Ungerade Parität bedeutet, dass das Paitätsbit die übertragenen Bits auf ungerade Anzahl von 1en ergänzt - hier wird das Paritätsbit mit einem Paritätselement erzeugt. Mit Paritätsbits kann man lediglich Ein-Bit-Fehler erkennen - das reicht meist aus, weil die Wahrscheinlichkeit, dass gleich zwei Bit auf einmal erfälscht wurden, sehr gering ist.

Bitverfälschungen entstehen durch Zerfall von Atomen in der Chip-Hülle oder durch störende Spannungen bei der Übertragung.

8.5 Logische Funktionen und Schaltnetze

Logischer Ausdruck Sei $X = \{x_1, x_2, \ldots, x_n\}$ eine Menge binärer Variablen (d.h. x_i kann den Wert 0 oder 1 annehmen). 0, 1, x_i sind (atomare) logische Ausdrücke. Sind a und b logische Ausdrücke, so auch $\overline{a}$, (a), $a \vee b$, $a \wedge b$.

Logische Funktion Eine Abbildung von einer Menge m-stelliger Binärwörter (m fest!) in die Menge $B = \{0, 1\}$ heißt logische Funktion oder Schaltfunktion. Logische Funktionen werden durch Tabellen oder logische Ausdrücke definiert.

Priorität logischer Operationen

1. Invertierung
2. Klammerausdrücke
3. AND
4. OR

Beispiele für logische Funktionen

1. $$f(x, y, z) = xy(\overline{x} \vee y) \vee z\overline{y}$$

2.

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

8.5.1 Umsetzung logischer Funktionen in verdrahtete Schaltnetze

Jede logische Funktion kann in Hardware (verdrahtete Schaltnetze) oder Software (programmierte Schaltnetze, Programm mit Speicher) umgesetzt werden. Alles, was mit Hardware realisierbar ist, ist auch in Software realisierbar.

Beispiel Die Klammern sind, von innen nach außen, Bauanleitungen zum Aufbau der Schaltung.

Teil III
Rechnerarchitektur 1 bei Prof. Dr. Bernd Müller

9 Einführung in die Rechnertechnik

9.1 Meilensteine der Computerarchitektur

- Mechanische Computer
  - Blaise Pascal
  - Konrad Zuse
  - Babbage usw.
  - ...
- 1. Generation: Vakuumröhren (1945-1955)
  - von-Neumann-Maschine
- 2. Generation: Transistoren (1955-1965)
- 3. Generation: Integrierte Schaltungen (1965-1980)
- 4. Generation: VLSI Integration (1980-?). Integration mehrerer 10^6 Transistoren auf einem Chip.

9.2 Kurze Geschichte der Rechnerentwicklung

(vgl. Tabelle im Skript GrdlgInf von Prof. Schmitt)

1990: Der RS8000 von IBM war die erste superskalare Maschine, d.h. hier werden mehrere ALUs in einem Chip integriert.

Di e Intel-Familie

- Intel 4004: Taktfrequenz $0,1$MHz. 4 Bit Datenbusbreite, d.h. 640 Byte sind adressierbar.
- Intel 8008: Erster Mikroprozessor mit einem 8-Bit-Datenbus.
- Mit der heutigen Datenbusbreite von 32 Bit können 4GB Hauptspeicher adressiert werden.

9.3 Moore's Law für Intel-Prozessoren

9.4 Aufbau eines Rechnersystems

Koponenten:

- Eingabegeräte
- Ausgabegeräte
- Speichermedien
- Netzwerkanbindung
- eigentlicher Rechner

Alle Komponenten sind an einem Bussystem angeschlossen, sowohl CPU als auch RAM und Controller für die verschiedenen Peripheriegeräte wie Keyboard, Maus und Festplatte.

9.5 Datenpfad in der von-Neumann-Maschine

Die Werte in Eingaberegistern müssen vor der Ausführung der arithmetischen Operation in die ALU-Eingangsregister transferiert werden. Das ALU-Ausgangsregister kann identisch sein mit einem ALU-Eingangsregister. Die Übertragungen zwischen den verschiedenen Registern werden von einem Leitwerk durchgeführt.

9.6 Funktionale Anforderungen an den Systembus

Ein Busteilnehmer will

- entweder mit einem anderen Teilnehmer Daten austauschen
- oder er fordert eine bestimmte Dienstleistung an

Folgende Anforderungen sind zu realisieren

- Busanforderung und Busarbitrierung. Es muss verhindert werden, dass zwei Teilnehmer gleichzeitig auf den Bus schreiben, weil dann die Daten nicht mehr identifiziert werden können.
- Interruptanforderung und -verarbeitung. Damit soll der Computer auf asynchrone (»überraschende«) Ereignisse wie Tastatureingaben reagieren können.

Gruppen des Leitungssystems:

Versorgungsbus. Für Spannungsversorgung und Taktleitungen. Hauptspannungen +12V, +5,5V, +3,3V. Wenn z.B. bei serieller Datenübertragung negative Spannungen benötigt werden, so erzeugt man sie direkt auf dem jeweiligen Controller mit einem Spannungswandler.

Adressbus. Dient der Auswahl der am Bus angeschlossenen Geräte.

Datenbus. Busbreite 1 Bit bei SPS, 8, 16, 32 oder 64 Bit). Bei 64 Bit Busbreite werden nicht alle Leitungen realsiert.

Steuerbus. Überträgt die zur Erfüllungd es Busprotokolls notwendigen Steuersignale.

9.7 Softwarearchitektur

Schichtenweise von oben nach unten:

- Anwendung
- Standardbibliotheken
- Systemaufruf-Schnittstelle
- Dateisystem
- Basisfunktionen
  - CPU-Verwaltung
  - Prozessverwaltung
  - Hauptspeicherverwaltung
  - Cache-Verwaltung
  - IPC: inter process communication
- Hardware-Abstraktionsschicht. Dies soll gewährleisten, dass ein Betriebssystem auf unterschiedlichster Hardware laufen kann.
- Hardware

9.8 Der Personalcomputer

10 Grundkonzept der von Neumann-Architektur

Nach von Neumann besteht eine Rechenanlage aus 5 Bausteinen:

- Leitwerk (engl. control unit)
- Rechenwerk (engl. arithmetic logic unit). Kann logische und arithmetische Verknüpfungen durchführen.
- Hauptspeicher (engl. memory), der sowohl die Daten als auch die Anweisungen enthält.
- Eingabewerk (engl. input)
- Ausgabewerk (engl. output)

Nach von Neumann werden alle Operationen im Akkumulator durchgeführt, ebenso finden alle Eingaben und Ausgaben darüber statt. Bei heutigen Rechnerarchitekturen wird dieses Konzept nicht mehr in dieser Form angewandt.

Rechen- und Leitwerk werden realisiert durch ein verallgemeinertes komplexes Schaltwerk, die CPU (engl. für central processing unit). Es muss Datenübertragung zwischen den Komponenten durch einen Bus möglich sein. Die Datenbusbreite ist meist gleich der Wortbreite des Prozessors. Der Bus besteht aus Daten-, Adress- und Steuerleitungen. Über die Steuerleitungen wird gesteuert, wer Daten senden und empfangen darf. Auf den Hauptspeicher kann im Normalfall nur byteweise zugegriffen werden. Bit: engl. für binary digit, binäre Ziffer.

Bestandteile des Leitwerks:

- Befehlszähler. Er enthält immer die Speicheradresse des nächsten auszuführenden Befehls.
- instruction register. Enthält den Befehl vor der Ausführung.
- Ablaufsteuerung. Holt einen Befehl und führt ihn aus und wiederholt diesen Vorgang.

Im Assembler eines Prozessors gibt es Befehle MOV o.ä., mit denen man einen Wert aus einem Prozessorregister auf ein E/A-Gerät schreiben kann, d.h. in das Register dieses Geräts, einen E/A-Port. Ein sog. Operationsschritt eines Prozessors dauert im Normalfall einen Taktimpuls. Ein ADD ax Befehl benötigt z.B. vier Operationsschritte, ein ADD M (aus dem Hauptspeicher holen, zum Akku addieren) z.B. 7 Operationsschritte, d.i. 2 Operationszyklen. Eine Taktfrequenz von 2400MHz heißt also nicht, dass $2,4 \cdot 10^9$ Befehle pro Sekunde ausgeführt werden können. Im Prozessor gibt es shift left und shift right -Befehle, zur Multiplikation und Division mit 2.

Bestandteile des Rechenwerks:

- ALU
- Registerblock. Nach der Anzahl der Bits zur Adressierung von Maschinenbefehlen, die in einem Maschinenwort vorgesehen sind, richtet sich die Anzahl der Register.
- Statusregister. Es enthält z.B. das Carrybit, wenn ein Überlauf stattgefunden hat. Es enthält je nach Operation weitere Informationen wie: erster Operand ist kleiner / größer als zweiter Operand, Operanden sind gleich usw. Das Statusregister ist mit Datenleitungen mit dem Leitwerk verbunden. Die einzelnen Bits des Statusregisters werden auch Flags genannt; so gibt es das Carry-Flag, das Divide-by-Zero-Flag usw.
- Shifter. In manchen Prozessoren hier eingebaut.

Das Rechenwerk ist ein »universelles Operationswerk«, mit dem elementare arithmetische und logische Operationen ausgeführt werden können. Register dienen zur Speicherung von Operanden; auf sie ist ein exrem schneller Zugriff möglich. Heute kann jedes Register als Ergebnisregister verwendet werden, es gibt kein speziell ausgezeichnetes Akkumulatorregister mehr.

Das Leitwerk ist ein umschaltbares Steuerwerk; die Umschaltung zwischen verschiedenen Steuerungsalgorithmen geschieht durch den Operationscode (Opcode) der Maschinenbefehle.

Bei der HARVARD-Arcitektur, die ein paar Jahre nach der von-Neumann-Architektur entwickelt wurde, gibt es getrennte Speicherbereiche für Maschinenbefehle und Daten, zur Behebung des von-Neumann-Flaschenhalses. So nämlich ist der Zugriff auf Befehle und Daten gleichzeitig möglich. Nahezu alle Mikrocontroller sind nach dieser Architektur aufgebaut.

Typische Arten von Maschinenbefehlen:

- Arithmetische und logische Befehle
- Ein- / Ausgabebefehle
- Datentransferbefehle

Maschinenbefehle werden üblicherweise durch Mnemonics dargestellt.

11 Das Betriebssystem

Möglichkeiten der Definition:

- Ein Betriebssystem ist ein komplexes Programmpaket, das dem Benutzer eines System ermöglicht, die Hardware eines Systems effizient zu nutzen. Benutzer können Anwender, Administratoren oder auch Anwendungsprogramme sein.
- Die Gesamtheit der Software, die die Ausführung von Programmen ermöglicht. Ein Betriebssystem kann Dienste bereitstellen zur Verwaltung von Betriebsmitteln, zur Ein- / Ausgabesteuerung und zur Datenverwaltung. (IEEE Standard Glossary of Information Science).

Hauptaufgaben des Betriebssystems:

- Dienstleistungen für den Benutzer
  - Bequemer Zugang
  - Abstraktion von der Hardware
  - Verwaltung der Daten
- Automatische Steuerung des Rechners
  - Verwalten der Ressourcen des Rechners
  - Steuerungd es zeitlichen Ablaufs der Programme
  - Koordination gleichzeitig stattfindender Aktovotäten

Was sind die Ressourcen eines Rechners?

- Hardwareressourcen
  - Prozessor. Das Betriebssystem muss in Multiprozessorsystemen alle Prozessoren beschäftigen.
  - Hauptspeicher
- Softwareressourcen
  - Daten
  - Anwendugen
  - Prozesse
  - Treiber
  - Protokollstapel (schichtenweise aufgebaut wie bei TCP/IP)

Das erste Betriebssystem

- Programmierer ist das erste Betriebssystem
- Operator
- Batchsysteme mit Monitorprogrammen
- Multiprorammierung
- Time-Sharing

Abbildung 14: Hardwarezugriff durch Anwendungsprogramme

- Personal-Computer
- Spezialsysteme
  - Echtzeitsysteme
  - verteilte Systeme
  - Handy-Betriebssysteme

Virtuelle Maschine. Ein Betriebssystem liegt als Software-Schicht über der Hardware und verdeckt diese. Aufwändige und immer wiederkehrende Sequenzen von Befehlen, um die Hardware direkt anzusprechen, werden durch kurze und mächtige Befehle (sog. Systemaufrufe) an die virtuelle Maschine ersetzt. Die virtuelle Maschine abstrahiert von Hardware-Details. Beispiel: Betriebssystemaufrufe von DOS, Ansteuerung der Hardware unter DOS:

DOS: disk operating system, z.B. MSDOS. Der Betriebssystemaufruf geschieht vom Anwendungsprogramm aus durch softwaremäßig ausgelöste Interrupts, z.B. Int 20h, Int 21h. Das Betriebssystem spricht die Hardware dann dierekt an oder verwendet seinerseits BIOS-Aufrufe. Anwendungsprogramme können theoretisch auch direkt über das BIOS auf die Hardware zugreifen, sogar direkt auf die Hardware - das ist aber nicht der normale Weg. Mit Interrupts kann man also sowohl auf Hardware, BIOS, als auch auf Betriebssystemfunktionen zugreifen. Das Einlesen eines Zeichens geschieht z.B. hardwareunabhängig durch einen Betriebssystemaufruf mit Int 21h.

Sichtweisen des Betriebssystems

- Sichtweise des Benutzers eines Rechners. Er verlangt eine effiziente, einfache Bedienung.
  - MMI (man machine interface)
  - GUI (graphical user interface)
- Sichtweise des Anwendungsentwicklers. Eher an Funktionsbibliotheken interessiert.
  - API (application programming interface).

11.1 Prozesse

Auf einem Betriebssystem laufen i.A. mehrere Programme gleichzeitig (Multiprogramming). Ein Prozess ist dann definiert als ein Programm, das sich gerade in Ausführung befindet. Wenn gleichzeitig mehrere Nutzer ein System nutzen können, heißt das »Multiuserfähigkeit«. Prozesse sind recht komplexe Obekte:

- Prozesse können parallel existieren
- Prozesse können auf gemeinsame Ressourcen zugreifen (synchronisiert durch das Betriebssystem), z. B. Nutzung gemeinsamer Speicherbereiche.
- Prozesse können neue Prozesse erzeugen (child-process)
- Prozesse können untereinander kommunizieren (interprocess communication)
- Prozesse können beendet werden.

Die Prozessverwaltung ist Aufgabe des Betriebssystems. Es muss jedem Prozess Programmspeicher zur Verfügung stellen. Darin befinden sich die Maschinenbefehle (CODE-Segment), statische Daten in einem reservierten Speichersegment, Anforderung weiteren Speichers (HEAP-Segment, STACK-Segment; dynamisch verwaltet, d.h. Allozierugng und Freigabe zur Laufzeit). Das Betriebssystem stellt einen virtuellen Speicher zur Verfügung, d.h. man kann mit vollen 32bit-Adressen arbeiten, als hätte man 4GB Hauptspeicher. Das Betriebssystem sorgt dafür, dass der benutzte Speicher auch real beeitgestellt wird und ggf. der Inhalt von Hauptspeicher auf Festplatte ausgelagert wird. Aufgaben des Betriebssystems:

- Prozessmanagement. Aufgabe ist die fehlerfreie, zeitlich verschränkte Ausführung der Prozesse.

Abbildung 15:

- Erschaffen und Beenden von Prozessen
- Unterbrechen aktiver Prozesse
- Einhalten vorgegebener Antwortzeiten
- IPC: inter process communication
- Verwaltung der Prozesse nach Priorität

  Hintergrundprozesse. Niedrigste Priotität. Hier kann sich das Betriebssystem »Zeit lassen« - es sind Aufgaben, die ohnehin einige Zeit beanspruchen, z.B. Lesen und Schreiben auf die Festplatte.

  Vordergrundprozesse. Höhere Priorität, hier geht es um Interaktion mit dem Benutzer.

  Echtzeitprozesse. Höchste Priorität, denn hier geht es um zeitkritische Wechselwirkung mit äußeren Prozessen.

- Kommunikation und Synchronisation. Nicht nur einzelne Prozesse müssen untereinander kommunizieren, sondern auch mehrere Rechner.
  - Interprozesskommunikation.
    - Zwei Prozesse in einem Rechner können über Signale kommunizieren (unterschiedliche Signale für unterschiedliche Ereignisse).
    - Zum Datenaustausch ist ggf. ein gemeinsamer Speicherbereich erforderlich. Das Betriebssystem muss die zeitliche Abfolge synchronisieren.
    - Auch eine Datei oder eine Pipe eignet sich zum Informationsaustausch.
    - Nachrichten (messages)
  - Kommunikation meherer Rechner, verteilte Systeme. Ebenfalls Aufgabe eines modernen Betriebssystems.
    - Auch hier kommunizieren zwei Prozesse miteinander, nur befinden sie sich auf unterschiedlichen Rechnern. Möglichkeiten der Kommunikation sind Dateien, Pipes, Nachrichten.

- Ressourcen-Management
  - Peripheriegeräte verwalten
  - Zugriff auf Dateien mit Hilfe eines Dateisystems koordinieren. Das Dateisystem beginnt bei einem Hauptverzeichnis, unter dem sich hierarchisch gegliedert Dateien und weitere Verzeichnisse befinden können. Jede Datei wird mit einem Pfad durch diesen Verzeichnisbaum bezeichnet. Dei Bedienung des Dateisystems soll vom Betriebssystem von der Hardware abstrahiert ermöglicht werden.
  - Verplanen und Bereitstellen des benötigten Hauptspeichers. Wenn der physisch vorhandene Hauptspeicher nicht ausreicht, müssen Bereiche in eine Swap-Datei ausgelagert und später wieder eingelagert werden. Der Hauptspeicher ist physisch byteweise organisiert, zur besseren Verwaltung und zum Auslesen wird er jedoch meist seitenweise verwaltet (Größe einer solchen »page« etwa 4 KB).

- Benutzerschnittstelle. Schnittstelle ist eigentlich die Trennung von zwei (Software, Harware)-Schichten. Schichtenweiser Aufbau:
  - Tastatur, Bildschirm. Zwischen diesen Komponenten und dem Kommandointerpreter liegt die Benutzerschnittstelle.
  - Kommandointerpreter (shell). Die Eigenschaften der Schnittstelle hngen vom verwendeten Kommando-Interpreter ab, unter *x-Systemen z.B. csh, bash, tcsh, corn.
  - Betriebssystemkern (kernel). Die Schnittstelle zwischen Kommandointerpreter und Betriebssystemkern ist standardisiert.

- Netzwerkdienste

Abbildung 16:

- »Netzwerke verbinden verteilte Systeme«.
- Im Netzwerk besteht eine Sammlung von Prozessen auf verschiedenen Computern, die also keinen gemeinsamen Speicher teilen.
- Kommunikationsdienste realisieren die Kommunikation der Prozesse. Für den Benutzer sollen diese Dienste transparent (unsichtbar) sein - er soll sich keine Gedanken darüber machen, wo z.B. im Netz seine Dateien gespeichert werden.
- Netzbetriebssysteme ermöglichen die gemeinsame Nutzung aller im Netz vorhandenen Ressourcen.

Arten von Prozesszuständen:

- Durch Interrupts können eigentlich blockierte Prozesse wieder aktiviert werden; effizienter als Polling.

Klassifizierung von Betriebssystemen

- Monolithische Systeme
- Geschichtete Systeme
- Dienstorientierte Systeme. Kleiner Kernel (Micro-Kernel) und Anwendungen in priviliegiertem Modus, die als Server für Dienstleistungen fungieren.

Teil IV
Aufgabensammlung

Es folgen Aufgaben von Prof. Dr. Eichner mit Quellenangabe und Lösung. Die Aufgaben sind thematisch geordnet und nicht nach ihren Quellen, so dass man diese Aufgabensammlung sehr gut verwenden kann, um Lösungsraster für die Aufgaben in der Klausur aufzufinden. Die Gliederung in Kapitel setzt sich aus der Gliederung in [7] und den durch dieses Dokument gemachten Zusätzen zusammen.

12 Aufgaben zu »Grundbegriffe des Rechneraufbaus«

12.1 Grobstruktur

12.1.1 Wissensfragen zur PC-Hardware

12.1.2 Wissensfragen zur PC-Hardware

12.2 Hauptspeicher (MM; main memory)

12.3 Magnetbandspeicher

12.4 Festplatte

12.5 Die von Neumann-Architektur

12.6 Betriebssystem

12.7 Einführung in die Rechnerhardware

12.8 Historie der PC-Entwicklung

12.9 Kommunikation zwischen den Bausteinen des Mainboards

13 Aufgaben zu: »Schaltnetze: Verdrahtete Logik«

13.1 Schaltalgebra

13.1.1 Beweis einer Formel mit Formeln (Einzeiler)

13.1.2 Beweis des Assoziativgesetzes

Quelle: [8, H1.1]. Beweisen Sie das Assoziativgesetz: $a \vee (b \vee c) = (a \vee b) \vee c$.

Das Gleichheitszeichen zwischen den beiden Formeln ist bewiesen, wenn sie denselben Werteverlauf haben (also wenn $(a \vee (b \vee c)) \leftrightarrow ((a \vee b) \vee c)$ eine Tautologie ist):

a	b	c	$a \vee (b \vee c)$	$(a \vee b) \vee c$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

13.1.3 Beweis der Regel von DeMorgan

13.2 Binäre Schaltzeichen

13.2.1 Schaltzeichen aus Tabelle

13.2.2 Schaltung in Schaltzeichen (multiple choice)

13.2.3 Ausgangswert bei gegebener Eingangsbelegung

13.2.4 Bezeichung von Schaltzeichen / Tabelle in Schaltzeichen

13.3 Logische Funktionen und Schaltnetze

13.3.1 Logischen Ausdruck in Schaltung, ohne Optimierungen

13.3.2 Schaltnetz in logischen Ausdruck, ohne Optimierungen

13.3.3 Aufwand und Geschwindigkeit eines Schaltnetzes

13.3.4 logischen Ausdruck in Schaltnetz, ohne Optimierungen; Aufwand und Geschwindigkeit eines Schaltnetzes; Schaltnetz in logischen Ausdruck

13.3.5 Schaltung in logischen Ausdruck, ohne Optimierungen

13.3.6 Logischen Ausdruck in Schaltung, ohne Optimierungen

13.4 Normalformen

13.4.1 Funktionen in eine Normalform, Funktionen in Schaltnetze

13.4.2 Minterm- und Maxtermdarstellung in Lang- und Kurzform ermitteln

13.4.3 Normalformtyp und Länge einer Schaltfunktion

13.5 KV-Diagramme

13.5.1 Primterme über KV-Diagramm, Arten dieser Primterme

13.5.2 DMFn durch KV-Diagramm, Funktion im KV-Diagramm gegeben

13.5.3 Disjunktive Minimalformen aus KV-Diagramm

13.5.4 Mintermdarstellung aus KV-Diagramm

13.5.5 Funktion in KV-Diagramm darstellen

13.5.6 Durch KV-Diagramm: Kernprimterme, eliminierbare Primterme, disjunktive Minimalformen

13.6 Verfahren von Quine-McCluskey

13.6.1 AND/OR-Minimalformen über Primtermtabellen aus Primtermen

13.6.2 Primterme einer Funktion mit Quine-McCluskey-Verfahren

13.6.3 Disjunktive Minimalformen aus Primtermen über Primtermtabelle

13.6.4 Primterme einer Funktion mit Quine-McCluskey-Verfahren

13.6.5 Minimale disjunktive Normalformen durch Primtermtabelle

13.6.6 Quine-McCluskey-Verfahren für Funktion mit 4 Variablen

13.7 Umsetzung unvollständig definierter Funktionen in Schaltnetze

13.8 Realisierung von Schaltnetzen durch Decoder-Coder-Kombinationen

13.8.1 Code-Umsetzung aus verdrahteter Schaltung ablesen

13.8.2 Decoder zu einem Code-Umsetzer mit Enable entspr. Tabelle verdrahten

13.8.3 Tabelle in verdrahtete Decoder-Codierer-Kombination umsetzen

13.8.4 4 Decoder zu einem Code-Umsetzer verdrahten

13.8.5 Funktionsbündel aus Tabelle in Decoder-Codierer-Kombination und ROM umsetzen

13.8.6 Tabelle als verdrahtete Decoder/Codierer-Kombination

13.8.7 In Tabelle gegebenen Coder als verdrahtetes Schaltnetz realisieren

13.8.8 In Tabelle gegebenen Coder als verdrahtetes Schaltnetz realisieren

13.9 Datenwegschaltungen

13.9.1 Demux und Adressierer kombinieren zu einem 1-zu-12-Demux

13.9.2 1-zu-4-Dmux als 2-zu-4-Decoder betreiben

14 Aufgaben zu: »Schaltnetze: Programmierte Logik«

14.1 Realisierung von Schaltnetzen in ROMs

14.1.1 Funktionsbündel in einem angegebenen ROM realisieren

14.1.2 Tabelle in einem angegebenen ROM realisieren

14.1.3 Funktionsbündel passend umformen und im gegebenen ROM realisieren

14.2 Zusammensetzung von ROMs

14.2.1 In zusammengesetzte ROM realisierte Funktionen in KDNF angeben

14.3 Realisierung von Schaltnetzen in PLAs

14.3.1 Funktionen in PLA realisieren, mit Optimierungen

14.3.2 In einer gegebenen PLA realisierte Funktionen in DNF angeben

14.3.3 Tabelle in einer angegebenen PLA realisieren, ohne Vereinfachungen

14.3.4 In PLA realisierte Funktionen als DNF angeben

14.3.5 In Tabelle gegebene Funktion in angegebener PLA realisieren

14.4 Kommerzielle PLAs

14.4.1 Funktionsbündel im IFL F82S1523 (kommerzieller PLA) umsetzen

14.5 Realisierung von Schaltnetzen in PALs

14.5.1 Funktion in angegebener PAL darstellen

14.6 Generic Array Logic (GAL)

14.6.1 Fragen zu OLMC13 eines GAL 16V8

15 Aufgaben zu: »Schaltwerke«

15.1 Allgemeines Schaltwerksmodell

15.2 Moore-Schaltwerke

15.2.1 Übergangsgraph eines Moore-Schaltwerks nach Vorgaben entwickeln

15.2.2 Übergangsgraph in mikroprogrammiertes Moore-Schaltwerk umsetzen

15.2.3 Übergangsgraph aus Moore-MPS mit MUX ablesen

15.2.4 Übergangsgraph in mikroprogrammiertes Moore-Schaltwerk mit und ohne MUX umsetzen

15.3 Ein serieller Vergleicher als Moore-Schaltwerk

15.4 Mealy-Schaltwerke

Literatur

[1] A. S. Tanenbaum, J. Goodman: »Computerarchitektur«; Prentice Hall 1999.

[2] P. Hermann: »Rechnerarchitektur«; Vieweg 1998.

[3] P. Pernards: »Digitaltechnik 1 und 2«; Hüthig-Verlag 1995 und 2001.

[4] W. Schiffman, R. Schmitz: »Technische Informatik 1 und 2«; Springer-Verlag.

[5] Hans-Peter Messmer: »PC-Hardware«; Addison-Wesley; 6. Auflage 2000; 59,95 EUR. Es gibt gebrauchte Exemplare für 30 EUR bei Amazon http://www.amazon.de. Dieses Buch ist für den Stoff zu PC-Hardware in dieser Vorlesung zu empfehlen. Es ist wohl das beste Buch zu PC-Hardware.

[6] Prof. Dr. Müller: »Umdruck zur Vorlesung Rechnerarchitektur 1«. Kann in der Veranstaltung Rechnerarchitektur 1 bei Prof. Müller für 2,50 EUR erworben werden und enthält im Anhang das gesamte Skript von Professor Eichner [7]! Die Veranstaltung bei Prof. Müller ist vergleichbar mit der bei Prof. Eichner: einige Teile werden fehlen, etwas wird zusätzlich vorhanden sein.

[7] Prof. Dr. Lutz Eichner: »Notizen zur Vorlesung Netz- und Schaltwerksentwurf; WS 2000/2001«. Das »offizielle Skript« dieser Veranstaltung. Es existiert nicht in digitaler Form, sondern wird manchmal, insbesondere auf Nachfragen, von Prof. Eichner als Kopien ausgegeben und ist außerdem als Anhang in [6] enthalten. Vorsicht: im Skript von Prof. Eichner können noch einige (Druck-)Fehler enthalten sein.

[8] Prof. Dr. Lutz Eichner: Hausübungen zur Veranstaltung Rechnerarchitektur 1 im Wintersemester 2002/2003. Im Internet erhältlich auf der Homepage von Prof. Dr. Eichner http://homepages.fh-giessen.de/~hg56.

[9] »Netz- und Schaltwerksentwurf; 17. April 1997«; Methodenerklärung zur Veranstaltung »Netz- und Schaltwerksentwurf« bei Prof. Dr. Lutz Eichner. 27 Seiten. Erhältlich in der Skriptsammlung der Fachschaft MNI der FH Gießen http://www.fh-giessen.de/FACHSCHAFT/Informatik/cgi-bin/navi01.cgi?skripte. Direkter Link http://www.fh-giessen.de/FACHSCHAFT/Informatik/data/skripte/nus.zip. Wenn [7] verfügbar ist, ist dieses Dokument nicht mehr notwendig, zumindest nicht in der Klausur.

[10] »Netz- und Schaltwerksentwurf Kurzanleitung«. 9 Seiten. Dokument in 9 TIFF-Dateien mit einem »GEC«-Logo. Ursprüngliche Quelle im Internet leider nicht mehr auffindbar, möglicherweise http://www.fh.gecfilm.de. Daher zur Verfügung gestellt als http://matthias.ansorgs.de/InformatikDiplom/Modul.ReArch1.Eichner/Kurzanleitung.zip.

[11] Prof. Dr. Lutz Eichner: »Fach: Netz- und Schaltwerke; Klausur im Januar 1999«. Enthalten in der Klausurensammlung der Fachschaft MNI der FH Gießen http://www.fh-giessen.de/FACHSCHAFT/Informatik/cgi-bin/navi01.cgi?klausuren als in PDF oder AmiPro. Direkter Link zur PDF-Datei: http://www.fh-giessen.de/FACHSCHAFT/Informatik/data/klausuren/nus9901.pdf.

[12] Prof. Dr. Lutz Eichner: »Fach: Netz- und Schaltwerke; Klausur im März 1999«. Enthalten in der Klausurensammlung der Fachschaft MNI der FH Gießen http://www.fh-giessen.de/FACHSCHAFT/Informatik/cgi-bin/navi01.cgi?klausuren als in PDF oder AmiPro. Direkter Link zur PDF-Datei: http://www.fh-giessen.de/FACHSCHAFT/Informatik/data/klausuren/nus9903.pdf.