Proseminar Mikroprozessoren Interfacing I/O Devices to the Memory, Processor and Operating System

Von Jan Milz

Gliederung

1. Einführung (What is I/O ?)
2. I/O Performance – Messungen
3. Typen und Eigenschaften von I/O Geräten
4. Busse
5. Functions the OS must provide
6. Giving Commands to I/O Devices
7. Communicating with the Processor – Polling, Interrupts,
8. DMA
9. DMA and Memory
10. Zusammenfassung
11. Future Directions in I/O Systems

Einführung

What are I/O Devices ?

Wir haben gesehen wie ein word oder ein Datenblock durch ein Busprotokoll übertragen wird. Es bleiben weitere Aufgaben offen, die verwirklicht werden muessen, wie z.B. Daten von einem Geraet in die Speicheradresse eines Programms zu transferieren. Folgende Fragen treten auf:

- Wie wird eine Benutzer I/O Anfrage in ein devicecommand umgewandelt und wie erfolgt die Kommunikation?

- Wie werden Daten in und aus dem Speicher gelesen?

- Welche Rolle spielt das Betriebssystem?

Drei Charakteristika eines I/O Systems:

1. Das I/O System teilen sich viele Programme, die auf den Prozessor zurückgreifen.

2. I/O Systeme benutzen Interrupts für I/O Operationen. Da Interrupts auf die Kernel-Ebene zugreifen, muss das OS sie steuern

3. Die Kontrolle eines I/O Gerätes ist sehr komplex, da viele Vorgänge gleichzeitig gesteuert werden muessen und die Anforderungen einer korrekten Kontrolle oft sehr detailliert sind

Functions the OS must provide

Das Betriebsystem spielt eine große Rolle beim handling von I/O Vorgängen. Es ist das **Interface zwischen Hardware und Software**.

Die oben genannten Charakteristika führen zu vielen verschiedenen

Funktionen, die das OS zur bereitstellen muss:

- Das OS garantiert, dass ein Benutzerprogramm nur auf die Teile eines Gerätes zugreift, für die der Benutzer auch die Zugriffsrechte hat.
- Das OS stellt Unterprogramme (routines) bereit, die den Gerätezugriff steuern.
- Das OS steuert die Interrupts die von I/O devices ausgehen.
- Es versucht gleichwertigen Zugriff auf die geteilten I/O Ressourcen zu ermöglichen.

Um diese Funktionen zu Gunsten der Benutzerprogramme zu performen, muss das OS in der Lage sein mit den I/O Geräten zu kommunizieren und Benutzerprogramme am direkten Zugriff hindern.

Drei Arten der Kommunikation werden benötigt:

1. Das OS muss den I/Os Kommandos geben können. Diese sollen nicht nur Operationen wie read oder write enthalten, sondern auch eine Dateiensuche (diskseek) o.ä. implizieren können.

2. Das device muss in der Lage sein, dem OS eine vollständige Operation oder einen Fehler zu melden.

   – beendete Suche

3. Daten muessen zwischen Speicher und I/O device bewegt werden.
   - ein ausgelesener Block während eines diskreads muss von der harddisk in den Speicher bewegt werden.

Giving Comands to I/O Devices

Um einem I/O device einen Befehl zu geben, muss der Prozessor es adressieren. Es gibt zwei Methoden das Device zu adressieren.

- memory-mapped I/O
- special I/O instructions

Durch **memory-mapped I/O** werden Teile des Adressenspeichers genutzt, um durch reads and writes in den Adressen Kommandos an die devices zu realisieren.

Dabei wird z.B. ein write genutzt, um Daten an das Geraet zu senden, wo diese als Kommando interpretiert werden.

Wenn der Prozessor die Adresse und die Daten in den Memory Bus schickt, ignoriert der Speicher die Operation, weil die Adresse einen Teil des Speichers anzeigt, der für I/O reserviert ist.

Der device-controller bemerkt den Vorgang und leitet die Daten (however) an die Geräte weiter.

Benutzerprogramme können hier nicht stören, da der I/O-Adressspeicher durch das OS geschützt ist.

Es sind mehrere seperate I/O Operationen notwendig, um eine solche Programmabfrage zu verwirklichen.

Der Prozessor muss den Status der Geräte abfragen können.

Beispiel: DEC LP11
Dieser Drucker hat zwei Register. Eines für Statusinformationen und eines für Daten, die gedruckt werden sollen. Das Statusregister enthält ein done-bit, das gesetzt wird, wenn ein Buchstabe gedruckt wurde und ein error-bit, das gesetzt wird, wenn der Drucker Störungen oder kein Papier mehr hat. Alle Daten, die gedruckt werden sollen, werden in das Datenregister geschrieben. Der Prozessor muss warten, bis das done-bit
gesetzt ist, um einen weiteren Buchstaben in den Buffer zu laden. Außerdem muss er das error-bit checken, um zu stoppen, falls ein Fehler auftritt. Jede dieser Aktionen braucht eine Deviceadresse.

Special I/O Instructions:

Diese Instructionen können sowohl die devicenumber als auch das command word spezifizieren. Der Prozessor spricht die Deviceadresse über den I/O bus an.
Um unerlaubten Zugriff zu verhindern, können I/O Instruktionen nur in supervisor oder kernel mode gesteuert werden.

- z.B. Intel 80x86 oder IBM 370 benutzen dedicated I/O instructions.

Der periodische Check eines Statusbits (beim eben genannten Drucker) wird **polling** genannt.

Communicating with the Processor

Polling:

- der einfachste Weg der Kommunikation zwischen device und Prozessor.
- Informationen werden im Statusregister platziert und werden vom Prozessor interpretiert.

→ Der Prozessor hat die gesamte Kontrolle aber auch die ganze Arbeit.

Nachteile:

- Verschwendung von Prozessorleistung, da Prozessoren viel schneller als I/O devices sind.
- Polling muss kontinuierlich weiterlaufen, da keinen Signale vom Gerät selbst ausgehen. Das OS(Prozessor) muss warten bis das Gerät auf den poll reagiert.

Interrupts

- Unterbrechung des Prozessors durch device, das aktiv werden will.
- OS gesteuert.
- Es gibt eine Interrupt-Hierarchy bei vielen parallel auftretenden Interrupts (Prioritäten)

→ Der Prozessor braucht nicht mehr zu pollen, kann sich auf das auszuführende Programm konzentrieren.

Interrupt-Algorithmus:

1) Unterbrechungssignal tritt auf (interrupt recognition).

2) Per Hardware wird geprüft, ob Berechtigung vorliegt. Wenn nicht, interrupt wirkungslos. (interrupt masking)

3) Wenn ja, wird aktives Programm(Zählerstand, Registerinhalte, Status) gesichert. (save status)

4) Unterbrechungsursache wird ermittelt und

   - Der Unterbrecher erhält Rückmeldung, dass der interrupt angekommen ist (interrupt acknowledge)
   - Ein zweites Programm wird gestartet und erledigt notwendige Systemarbeiten. (interrupt sevice routine)

5) Wenn Interrupt erledigt, wird altes Programm wieder geladen und fortgesetzt. (retore and return)

Der Prozessor soll aus der Kette Speicher↔Prozessor↔Device
Entfernt werden.
Zusätzliche hardware wird benötigt:

DMA - Direct memory acces

- Direktspeicherzugriff: unmittelbarer Zugriff eins peripheren Gerätes auf den Speicher.
- Sitzt zwischen I/O controller und Speicher
- Es gibt eine speziellen **DMA-controller**, der ohne die CPU Daten vom Arbeitspeicher über den Datenbus auf ein Gerät oder umgekehrt transportieren kann.
- Interrupts werden nur noch gebraucht, um dem Prozessor Fehler zu melden.

Drei Schritte während eines DMA- Transfers:

1) Der Prozessor teilt dem DMA das device, die auszuführende Operation, die Speicheradresse, aus der die Daten gelesen werden sollen, und die Anzahl der bytes, die transferriert werden sollen, mit.

2) Der DMA startet die Operation im device und bestimmt den bus. Wenn die Daten verfügbar sind(device oder Speicher), werden sie übertragen. Wenn der Vorgang mehr als einen Transfer benötigt, wird der nächste vorbereitet während der erste läuft. Manche DMAs haben einen eigenen Speicher, um flexibel mit Verzögerungen(delays) umgehen zu können, während sie auf den Bus warten.

3) Wenn der Transfer komplett ist, interruptiert der Kontroller den Prozessor.

DMA and the Memory Sytem

- Beziehung zwischen Speicher und Prozessor ändert sich. Ohne DMA greift der Prozessor direkt auf den Speicher zu.
- Ein neuer Weg zum Speicher entsteht, der nicht über Adressen und oder Cache läuft.
- Übertragungsschwierigkeiten treten auf, die mit Hilfe bestimmter Hard- und Software gelöst werden müssen.

In einem System mit virtuellem Speicher können Daten doppelt auftreten:
Der DMA greift direkt auf den Hauptspeicher zu, während der Prozessor den Cache mit einbezieht. So können gleiche Speicheradressen vom DMA und vom Prozessor unterschiedliche Werte annehmen(stale data problem or coherency problem).

I/O Geräte

I/O-Performancemessungen

Will man die Performance von Disksystemen messen, muß man wissen, daß die Benchmarks, mit denen gemessen wird, von anderen Systemeinheiten beeinflußt werden, z. B. Speicher, Prozessor (im Gegensatz zu Prozessorbenchmarks).
Gemessen wird in MHz = 10^6 HZ.
Die Transferrate wird in MB/s gemessen : 1MB = 10^6B = 1.000.000 B
(Hauptspeicher : 1 MB = 2^20 B = 1.048.576 B).

Supercomputer I/O

Das Maß der Supercomputer I/Os ist die Anzahl der Bytes pro Sekunde zwischen Hauptspeicher und Diskette.

Transaktionprozedur I/O

Der Durchgang wird zeitabhängig gemessen.
Da die Zugänge klein sind, wird die I/ORate als Maß genommen (Maßeinheit: Anzahl der Diskettenaufrufe pro Sekunde).
Die Meßwerte sind kritisch zu betrachten und die Messung ist kostenintensiv.
Beispiel : die Benchmarks TPC-C , TPC-D
Beide stellen die Anfragenbearbeitung gegen die Datenbasis. Gemessen wird die Leistung in Transaktionen / Minute o. Sekunde , wobei ein komplettes System an der Messung beteiligt ist.
Der TPC-D wird bei komplexen Anfragen (typisch bei decision-support-Anweisungen) benutzt.
Der TPC-C (z.B. für Onlinebanking) enthält 9 Typen Databaserecords, 5 verschiedene Transaktionen und eine Benutzersimulation.

Dateiensysteme I/OBenchmarks

Messungen der Unix-Dateisysteme an einer arbeitenden Umgebung ergaben :
- 80% Zugriffe auf Dateien mit weniger als 10 KB
- 90% Zugriffe auf Daten mit sequentieller Adresse
- 67% Leseprozesse, 27% Schreibprozesse, 6%LeseVeränderеSchreibprozesse

Aus den Ergebnissen wurden synthetische Dateisystemen entwickelt => Benchmarks.

Typen und Eigenschaften von I/OGeräten

Es gibt eine sehr große Bandbreite von Typen, die sich nach drei Punkten einordnen lassen:
1. Verhalten : Input (nur lesen), Output (nur schreiben), Storage(lesen u. schreiben)
2. Partner : Mensch oder Maschine
3. Datenrate: höchste Datentransferrate zwischen I/OGerät und Hauptspeicher/Prozessor
   => Höchstrate des I/OGerätes muß berücksichtigt werden

Beispiele für I/OGeräte mit Blick auf Interaktion mit dem Prozessors

Maus

Es gibt zwei Arten von Mäusen, die einen generieren eine Serie von Impulsen, die anderen haben Zähler für die Höhe und Breite. Der Prozessor liest in bestimmten Abständen die Stände und überträgt sie auf den Mauszeiger.
Der Zeiger bewegt sich abrupt, da der Prozessor die Stände aber sehr oft liest, scheint sich der Zeiger stetig zu bewegen.
Beide Arten haben einen oder mehrere Knöpfe, und das System muß merken, wenn sie gedrückt wurden, ev. mit einer Unterscheidung zwischen drücken und gedrückt halten.
Mäuse werden von der Software kontrolliert, sind daher einstellbar.

Magnetische Disketten

Es gibt zwei Arten von Disketten : Floppy Disks (weich) und harte Disketten
Beide Typen beruhen auf dem gleichen Prinzip: eine rotierende Platte mit magnetischer Oberfläche und einem bewegbaren read/write-Kopf, wodurch die Speicherung stromunabhängig ist.
Früher wurden große Disketten mit vielen Bits hergestellt, was die Elektrokosten aufwog, da dieses Modell die niedrigsten Kosten pro Bit hatte.
Heute haben kleinere Disketten Vorteile bei der Leistung und dem Volumen/Byte. Sie sind billiger zu produzieren und lassen sich in hohen Stückzahlen absetzen.
=> 1997 kostete ein MB $0,10 - $0,20 , egal wie groß die Diskette war
Heute ist es weit verbreitet die Disk mit einer konstanten Bitdichte zu bespielen. Das heißt, daß die Dauer in welcher ein Sektor unter den Kopf gelangt, variiert (außen schneller).
Also muß ebenfalls die Rate, mit der gelesen und geschrieben wird, ebenfalls variieren.

Netzwerke.

- einige Eigenschaften : -Entfernung : 10m - 10.000km
  -Geschwindigkeit : 0,001 MB/s - 100 MB/s
  -Topologie : Bus, Ring, Stern, Baum
  -Leitungen : Punkt-zu-Punkt oder geteilte (multi drop) Leitungen

Beispiel: Langstrecken-Netzwerke
Die Entfernung beträgt 10- 10.000 km.
Das erste war ARPANET (Advanced Research Projects Agency of US.Government)
Es schickte 56 Kbits/s und nutzt die schon vorhandenen Punkt-zu-Punkt-Verbindungen, nämlich Telefonleitungen.
Der Hostcomputer kommuniziert mit einem IMP(Interface Message Prozessor) , der teilt die Nachricht in ein Kbit-Pakete, wobei die Adresse in jedem Paket enthalten ist, und schickt sie über verschiedene Kabel zu ihrem Bestimmungsort.
An jedem Knoten werden die Pakete gespeichert, bis sie beim Ziel-IMP wieder zusammengesetzt werden und zum Empfängerhost geschickt werden.
Die meisten Netzwerke nutzen diesen „packet-switched"-Ansatz, bei dem einzelne Pakete sich einen individuellen Weg zum Ziel suchen.
=> ARPANET ist der Vorläufer des Internets.

Es wurde eine Protokoll-Familie (TCP / IP) entwickelt, um verschiedene Netzwerke zu verbinden. Beim IP (Internet Protokoll) tauschen zwei Hosts ihre Adressen aus, aber man hat keine Garantie, daß der Transfer geklappt hat. Beim TCP (Transmission Control Protokoll) ist sicher gestellt, daß alle Pakete ankommen.
Sie arbeiten zusammen : IP-Pakete umschließen die TPC-Pakete

Busse

Sie verbinden die I/OGeräte mit Prozessor und Speicher und die Einheiten untereinander.
Ihr Vorteil ist die Vielseitigkeit, da neue Geräte können leicht angeschlossen werden
Der Nachteil ist, daß ein Engpaß entsteht, der den I/ODurchgang limitiert.

Leistungsoptimierung:

Die Geschwindigkeit der Busse kann nicht stark erhöht werden, da sie hauptsächlich von physikalischen Faktoren, Länge des Busses und Anzahl der I/OGeräte, abhängt.
Die Leistung dagegen ist steigerungsfähig, hat aber ev. negative Auswirkungen.
Beispiel :
Schnellere Antwort auf I/OOperationen erreicht man durch Minimieren der Zeit des Buszugangs. Andrerseits muß für hohe I/ODatenraten die Bandbreite maximiert werden, z. B. durch mehr Buffer und den Transport von größeren Datenblöcken, aber beides verzögert den Zugang des Busses !
=> Die beiden Ziele schneller Buszugang und hohe Bandbreite führen zu entgegengesetzten Designansprüchen
Ein Bus enthält Kontrollleitungen und Datenleitungen.
Kontrollleitungen stellen Anfragen, geben Bestätigungen und identifizieren den Typ der Daten auf der Datenleitung, welche die Information zwischen Quelle und Ziel transportiert.
Informationen bestehen aus Daten, komplexen Kommandos oder Adressen.
Beispiel : Diskette schreibt einen Sektor in den Speicher
Die Datenleitung transportiert sowohl die Speicheradresse als auch die Daten selber (nacheinander). Die Kontrollleitung zeigt die Art der Information an; sie ändert sich also während des Prozesses.
Einige Busse haben zwei Datenleitungen, eine für die Adressen, die andere für die Informationen.
Andere Busse nutzen die Kontrollleitung zusätzlich für ein Protokoll.

Bustypen

1. Prozessor-SpeicherBus
- kurz, schnell, maximiert die Prozessor-Speicherbandbreite
2. I/OBus
- können länger sein, keine festgelegten Endgeräte, viele verschiedene Bandbreiten
3. BackplaneBus
- wurden entworfen, um Prozessor, Speicher und I/OGeräten zu erlauben zu kommunizieren

Einige Hochleistungssysteme nutzen eine Kombination aus allen drei Bussen, wobei der P-S-Bus einen oder mehrere Adapter für den Standardbackplanebus hat , und die I/OBusse an den Backplanebus angestöpselt werden können.
Der Vorteile ist, daß der P-S Bus unabhängig von den anderen schneller gemacht werden kann und eine Ausweitung des Systems unabhängig vom P-S Bus möglich ist.
Beispiel : IBM RS/6000 , Silicon Graphics multiproz

Synchrone und asynchrone Busse

Synchrone Busse haben eine Uhr in der Kontrollleitung und ein festes Protokoll in Abhängigkeit zur Uhr.
Das Protokoll ist einfach umzusetzen, da es wenig Logik benötigt, daher ist der Bus schnell.
Nachteile: 1. Alle Geräte des Busses müssen aufeinander abgestimmt sein.
2. Wegen Taktproblemen können schnelle Busse nicht lang sein.
P-S Busse sind häufig synchron, da sie mit wenigen, nahen Geräten mit hoher Frequenz arbeiten.

Asynchrone Busse sind in der Länge unbeschränkt; es lassen sich viele verschiedene Geräte anschließen.
Zum Koordinieren des Datenaustausches wird ein Handschlagprotokoll benutzt, das heißt, daß beide Geräte erst zum nächsten Schritt übergehen, wenn beide ihre Zustimmung gegeben haben. Realisiert wird es durch zusätzliche Kontrollleitungen.
Asynchrone Busse arbeiten wie abgeschlossene Maschinen, die nur etwas tun, wenn sie wissen, daß der andere einen bestimmten Status hat .
I/OBusse sind oft asynchron, damit neue Geräte leicht angeschlossen werden können.

Erhöhen der Busbandbreite
Viel der Bandbreite wird durch die Wahl zwischen asynchron und synchron entschieden.
1. Wird die Datenbusbreite erhöht, dauert der Transfer von mehreren Wörtern weniger Buszyklen.
2. Durch getrennte Adress- und Datenleitungen können Adresse und Information in einem Zyklus geschickt werden.
3. Blocktransfers : Man erlaubt dem Bus viele Wörter ohne Adresse oder Bestätigung zwischendurch in Back-to-Back Bussen zu transportieren
Die Nachteile sind mehr Busleitungen, erhöhte Komplexität und / oder erhöhte Antwortzeit bei Anfragen, die ev. warten müssen, während ein längerer Transfer läuft

Verteilen des Buszugriffs
Wie kann ein Gerät sich einen Bus reservieren ?
Entscheidend bei großen I/OSystemen ist es, daß die I/OGeräte ohne große Einmischung des Prozessors arbeiten
Es wird also eine Kontrollinstanz benötigt, die den Zugriff auf den Bus regelt.
Der Busmaster kontrolliert alle Anfragen und schickt die Busse los.
Der Prozessor muß trotzdem in der Lage sein, Busse zu rufen oder zu schicken, wobei er dann
der Busmaster ist => einfachstes System
Nachteil : Der Prozessor ist an jeder Transaktion beteiligt.
Bei mehreren möglichen Busmastern muß entschieden werden, wer den Bus zuerst benutzen darf.

Busvermittlung (Arbitration)
Es gibt viele verschiedene Schemata für die Vermittlung, z. B. durch spezielle Hardware oder ausgeklügelte Protokolle.
Ein Gerät signalisiert eine Anforderung; nach Gebrauch des Busses wird wieder ein Signal gegeben, daß der Bus frei ist.
Die meisten Multiplemasterbusse haben ein extra Leitungen für Anfragen und Freigabe
Haben nicht alle Geräte ihre eigenen Anfragenleitungen, braucht man eine zusätzliche Releaseline.

Allgemein berücksichtigen Schemata zwei Faktoren :
1. Priorität : die Geräte bekommen verschiedene Prioritäten zugewiesen
2. Fairness : auch das Gerät mit der niedrigsten Priorität soll irgendwann bedient werden

Vermittlungsschemata können in vier Klassen eingeteilt werden :
1) Daisy Chain Arbitration

Die Grant(=Freigabe)leitung läuft vom Vermittler zum Gerät mit höchster Priorität, dann zum nächsten, bis zum letzten.
Möchte ein Gerät den Bus benutzen, schickt es ein Signal über die Anfrageleitung und wartet auf eine Übertragung auf der Grantline, die anzeigt, daß der Bus zugeteilt ist.
Dann wird das Grantsignal unterbrochen und die Anfrageleitung freigegebenen.
Das Gerät benutzt den Bus und setzt danach die Releaseline zum Signalisieren von : Fertig!
Dadurch, daß die Grantline unterbrochen wird, verhindert man, daß Geräte hoher Priorität einem Gerät niedriger Priorität, das glaubt die Freigabe erhalten zu haben, den Bus weg-schnappen können.
Fairness ist nicht implementiert.

2) Die zentrale parallele Vermittlung nutzt multiple Anfrageleitungen, wodurch die Geräte ihre Anforderungen unabhängig stellen. Der zentrale Vermittler sucht ein aus und teilt diesem den Bus zu.
Der PCI, ein Standard-Backplanebus nutzt dieses Schema.

3) Die geteilte Vermittlung durch Selbsternennung nutzt ebenfalls multiple Anfrageleitungen. Jedes Gerät, das den Bus nutzen will, sendet einen Identifizierungscode zum Bus und kommt unabhängig von der Priorität an die Reihe.

4) Bei der geteilten Vermittlung durch Kollisionsentdeckung fordert jedes Gerät unabhängig den Bus an. Gehen mehrere Anfragen auf einmal ein, kommt es zur Kollision, die nach einem festgelegtem Schema abgebaut wird.

<u>Busstandards</u>
Die meisten Computer lassen sich erweitern, daher hat die Industrie verschiedene Busstandards
geschaffen. Manchmal wird ein I/OBus einer besonders populären Maschine de facto Standard, wie z. B. : IBM PC-AT Bus, oder ein Standard wird von einer kleinen Gruppe für ein bestimmtes Problem festgelegt, z. B. SCSI (Small Computer System Interface)
- PCI (ursprünglich von Intel entwickelt) nutzt einen general purposed Backplanebus
- SCSI hat Backplanebusse und P-S Busse