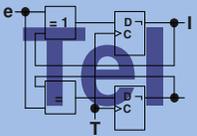


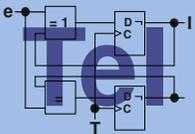
Bussysteme

Ernst-Robert Igel

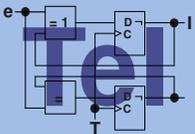
s7745007@mail.inf.tu-dresden.de



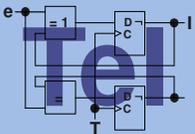
1. Warum Bussysteme?
2. Bus-Strukturen
3. parallele / serielle Busse
4. Buszugriffsverfahren
5. I²C
6. InfiniBand
7. Aussichten

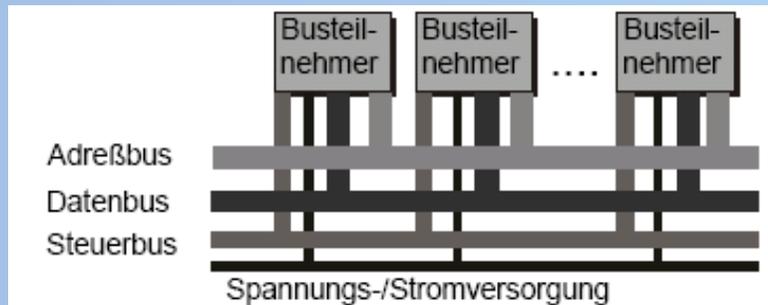


- **Bus - Broadcast and Unknown Server**
- Begriff für Datenverbindung mit mehreren Teilnehmern
- Leitungssystem mit Steuerungskomponenten zum Austausch von Daten und/oder Energie zwischen Hardware
- hunderte verschiedene Bussysteme
 - unterscheiden sich durch: Sicherheit, Kapazität, Latenz, Kosten, Echtzeitfähigkeit, ...
- Einsatzgebiete
 - intern/extern am Computer
 - Steuerung in der Industriefertigung
 - Autos
 - Gebäudetechnik
 - ...



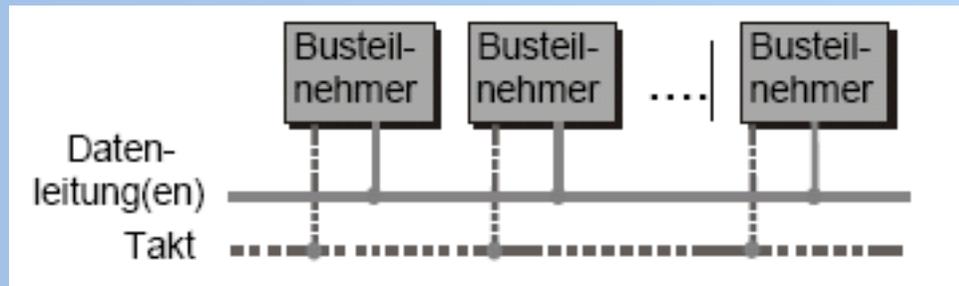
- Vernetzung von Komponenten zum Informationsaustausch
- grundsätzlich Alternative zur Einzelverdrahtung (Zweipunktverbindung)
 - Vorteile:
 - physisch nicht so komplex
 - einfachere Verkabelung
 - + günstiger
 - + leichter
 - Komponenten nicht so ortsfest
 - Nachteile:
 - geringere Ausfallsicherheit
 - Fehlerdiagnose meist schwieriger





- besteht meist aus drei getrennten Teilbussen: Adreßbus, Datenbus, Steuerbus
- Daten werden bit-parallel übertragen
- typischerweise 2^n Datenbits parallel transferiert
- meist auf Leiterplatten als Verbindung genutzt
- als Kabellösung recht aufwendig/teuer
- Datenkonsistenz beispielsweise durch Paritätsbits

- Multiplexbus: Adress- und Datenbus gemultiplext um Leitungen zu sparen
 - Beispiel: PCI-Bus

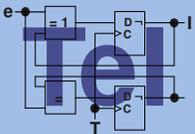


- Übertragung bit-seriell über gemeinsame Leitungen für Adressen, Daten und Steuerinformationen
- Übertragung in Blöcken (Paketen) mit Zieladresse, Steuerinformationen, Prüfsummen, manchmal Senderadresse
- asynchron: ohne Taktleitung, durch Steuerinformationen
 - → meist nur einzelnes Zeichen
- synchron: mit Taktleitung oder durch Code (NRZi, Manchester) synchronisiert
 - “teurer” als asynchron

- kann nicht verallgemeinert werden

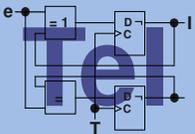
- serielle Busse
 - weniger Leitungen
 - Verbindungen günstiger
 - → meist für “weitere” Entfernungen sinnvoll

- parallele Busse
 - höhere Datenraten
 - weniger aktive Hardware
 - → kurze Verbindungen



- Topologien verlangen Zugriffssteuerung, da nur ein Medium vorhanden ist
- Unterscheidung zwischen kontrolliertem oder zufälligem Buszugriff

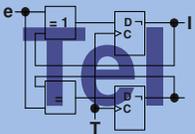
- kontrollierter Buszugriff
 - Sender ist vor dem Senden eindeutig bestimmt
 - allgemein echtzeitfähig, da sich die Zykluszeit berechnen läßt
 - Master/Slave-Verfahren
 - ein Master (aktiv), n Slaves (passiv)
 - Master baut Verbindung zu Slaves auf, diese antworten sofort
 - Slaves sehr günstig, da Intelligenz fast ausschließlich im Master



➤ kontrollierter Buszugriff

- Token-Passing

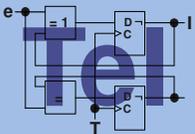
- alle Teilnehmer können Bus und Kommunikationssteuerung übernehmen
- Token wird genau einem Teilnehmer zugeteilt
- nach Datenübertragung wird der Token weiter gereicht
- meist als logischer Ring realisiert
- Teilnehmer müssen Überwachungsfunktionen durchführen
 - + wenn kein Token vorhanden: ein Teilnehmer muss neuen erzeugen
 - + wenn zwei Token vorhanden: ein Token muss gelöscht werden
 - + defekte Stationen müssen aus dem Ring entfernt werden
 - + Möglichkeit im laufenden Betrieb Stationen neu aufzunehmen



- hybride Zugriffsverfahren möglich: PROFIBus (**Process Field Bus**)



- zufälliger Buszugriff
 - mehrere Sender greifen bei Bedarf auf Bus zu (Multi-Master)
 - Mehrfachzugriff muß geregelt werden (Arbitrierung)
 - bitweise Arbitrierung
 - Teilnehmer sendet eigenen Identifier und prüft Ergebnis
 - CSMA (Carriere Sense Multiple Acces)
 - Teilnehmer hört Busleitung ab
 - wartet, falls Leitung belegt
 - Problem, wenn zwei gleichzeitig anfangen zu senden
 - CSMA/CD (Collision Detection)
 - Teilnehmer hören Leitung während des Sendens ab
 - sofortiger Abbruch bei Fehler
 - mit Bus-Arbiter
 - Arbiter entscheidet, wer den Zugriff auf den Bus erhält
 - Beispiel: Stichleitung
 - + Arbiter kann den Bus beliebig an anfordernde Teilnehmer zuteilen
 - + erfolgt bei Mehrfachanforderung meist nach Priorisierung

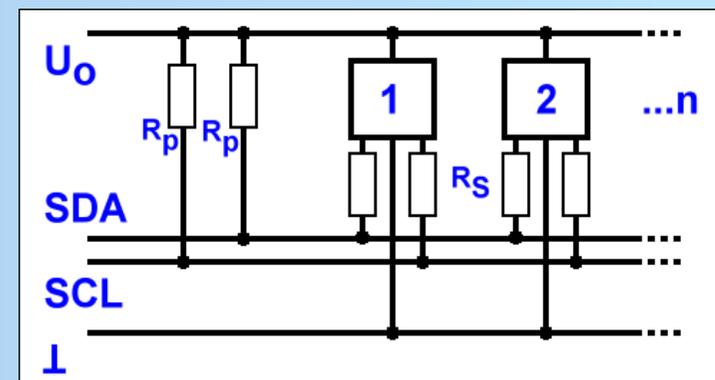


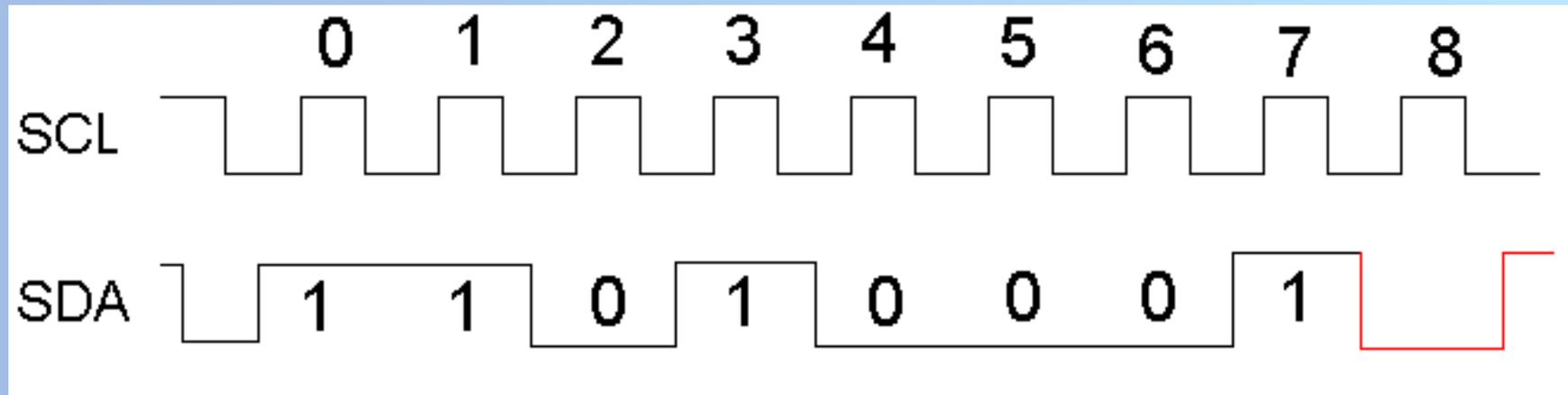
I²C - Inter-Integrated Circuit

- von Philips entwickelt, um Chips in Fernsehgeräten zu verbinden
- serielles Bus-System (Multi-Master bzw. Master-Slave)
- nur zwei Leitungen (SDA, SCL)

- Master sendet Adresse (mit R/W-Bit) und Daten
- Slave sendet bei Lesezugriffen Daten
- Daten werden byteweise quittiert
 - vom Slave nach Adress- und Schreibdaten
 - vom Master nach Lesedaten

- Geschwindigkeit max. 3,4 Mbit/s (meist nur 100 kbit/s)
- keine Prüfung auf Fehler während der Datenübertragung
- sehr günstige Komponenten

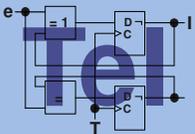




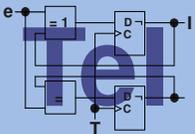
- Leitungen führen ohne Zugriff logisch 1
- während SCL logisch 1, signalisiert Pegeländerung an SDA Übertragungsbeginn
- während der Übertragung Pegeländerung an SDA nur bei SCL logisch 0 erlaubt
- nach acht Bit sendet DER Slave Quittung, indem er SDA auf logisch 0 zieht

- Vereinigung von "Future I/O" (Compaq, IBM, HP) "Next Generation I/O" (Intel, Microsoft, Sun)
- Einsatzgebiete: Netzwerkverbindungen, Bussystem
 - im Moment nur als Cluster-Verbindungstechnologie
- pro Kanal Datenübertragungsraten von 2,5 Gbit/s vollduplex (DDR Variante 5 Gbit/s)
- transparente Bündlung von Kanälen möglich
- besonders niedrige Latenz (ca. 3 Mikrosekunden)
- Übertragung über Kupferleitungen

- spezielle Verbindungsmodi, um Latenz niedrig zu halten:
 - Zugriff auf Hauptspeicher eines anderen Knotens, ohne Betriebssystem oder Prozessor der Gegenseite zu benutzen: RDMA (Remote DMA)



- immer wichtigere Rolle bei “einfacher” Vernetzung
 - Automobilindustrie
 - Hausautomatisierung
- Standardisierung von Bussen und Schnittstellen
- spezielle Bussysteme für noch schnellere Übertragungen
- echtzeitfähiges Ethernet (zum Beispiel Industrial Ethernet) wird als Standard vielfach komplexe, teure Busse in der Industrie verdrängen



Vielen Dank!

