



Wichtige Rechnerarchitekturen

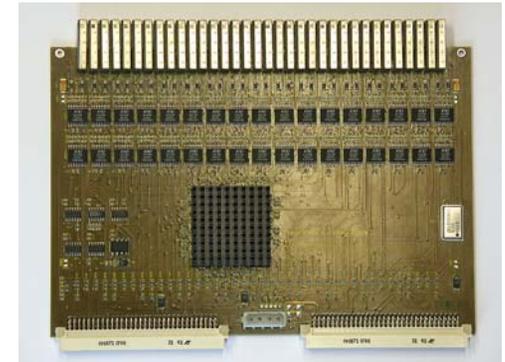
Teil 5

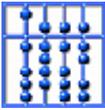
INMOS Transputer, CSP/Occam



INMOS Transputer

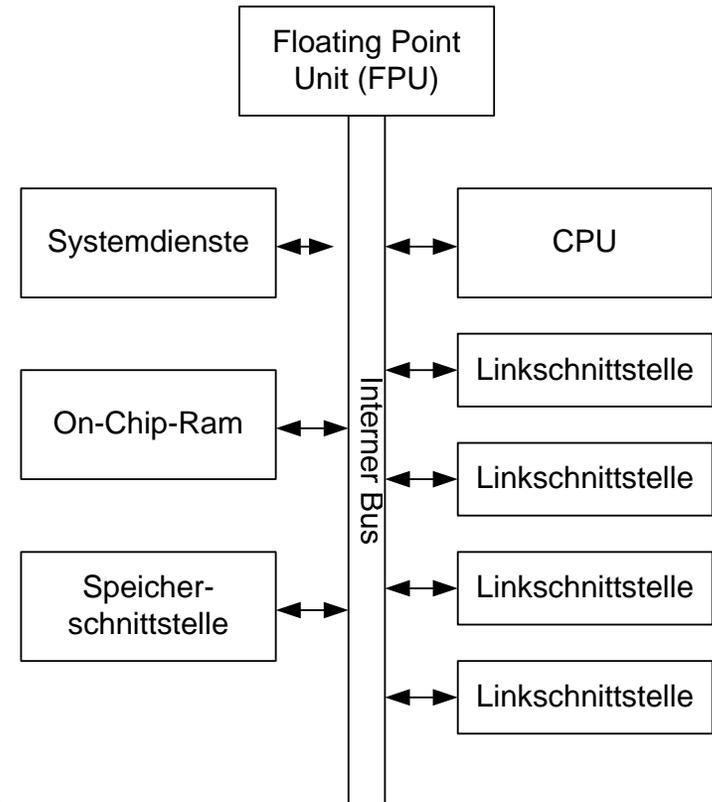
- 1983 vorgestellt von der Firma INMOS (Bristol) (Entwicklung seit 1978)
- Der Name Transputer entstand als Kunstwort aus Transistor und Computer (Idee: ähnlich wie bei Transistoren können mehrere Transputer zu einem leistungsfähigeren System zusammengebaut werden).
- Begriff umfasst eine Klasse von Rechnern, grobe Gliederung in T2xx (16-bit Prozessor), T4xx (32-bit Prozessor), T8xx
- Wesentliches Konzept: Eignung für Cluster \Rightarrow Parallelisierung der Rechenprozesse (Programmiersprache Occam)
- Für das Clusterkonzept wurde eine sehr schnelle Verbindungstechnik entwickelt, die Parallelisierung erfolgte nach dem Rendezvous-Konzept
- Mit geringstem Designaufwand sehr große Cluster aufbaubar
- Siehe <http://vl.fmnet.info/transputer/>

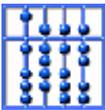




Architektur

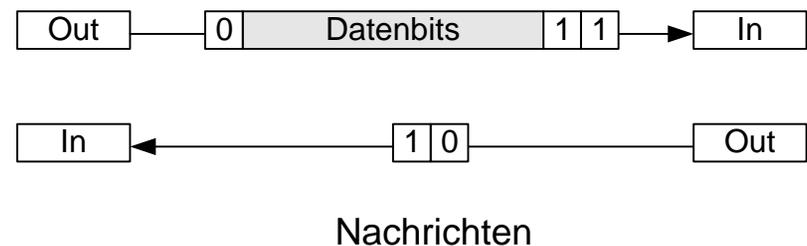
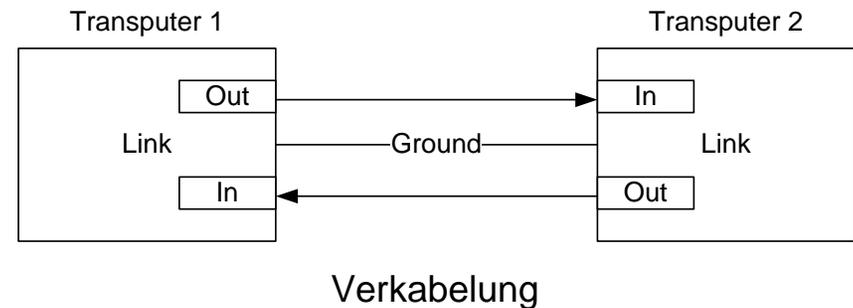
- Interner Bus: je nach Taktrate des Transputers konnten Transferraten von 80-120 MB/s erreicht werden
- CPU: bis zu 30 MHz Taktfrequenz, Prozessverwaltungsmechanismen sind bereits in der Hardware eingebaut (sehr schnell)
- On-Chip-RAM: maximal 4 kB groß, Zugriffszeit 33 ns, zusätzlich existiert eine Schnittstelle für externen Speicher (ab T4 bis zu 4 GB groß)
- Links: jeder Transputer besitzt 4 serielle, bidirektionale Links, über welche mit bis zu 4 anderen Transputern Daten ausgetauscht werden können, die Schnittstellen arbeiten parallel zur CPU (fest verdrahtete Steuerung), maximale Übertragungsraten 20 MB/s
- 64-Bit-Gleitkommaeinheit: bei allen Transputern der Klasse T8 an internen Bus angeschlossen

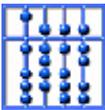




Kommunikation zwischen Transputern

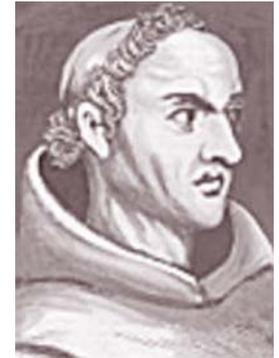
- Die Übertragung erfolgt in jede Richtung über eine separate Datenleitung.
- Der Versand erfolgt dabei über ein festes Protokoll: jedem zu übertragenden Datenbyte werden zwei Bit auf High-Pegel (11) vorangestellt und am Ende wird ein Low-Pegel-Bit (0) angehängt. Jede Nachricht wird dabei mit einem Acknowledge-Signal beantwortet, die Daten sendende Schnittstelle wird bis zum Empfang dieses Signals blockiert (synchrone Kommunikation, Rendezvous-Konzept).





Occam

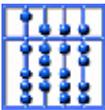
- Als Programmiersprache wurde Occam verwendet, mit der man parallel Abläufe festlegen konnte.
- Als Namenspate fungierte der Philosoph William of Ockham. Sein Postulat „*Dinge sollten nicht komplizierter als unbedingt notwendig gemacht werden*“ war Motto der Entwicklung.
- Occam basiert auf dem Modell CSP (communicating sequential processes) von C.A.R. Hoare; siehe auch CCS (Calculus of Communicating Systems) von R. Milner
- Occam ist eine Sprache, die die parallele Ausführung von Aktionen direkt mit einbezieht
- Die Kommunikation zwischen den einzelnen Prozessen erfolgt synchron über unidirektionale Kanäle.
- Die Realisierung auf dem Transputer ist 1:1. Als Kanal zwischen zwei Prozessen auf unterschiedlichen Transputern kann ein (halber) Link benutzt werden. Befinden sich die beiden Prozesse auf einem Transputer, so kann der Kanal über Speicherplätze simuliert werden.
- Siehe <http://vl.fmnet.info/occam/>



William of
Ockham



C.A.R. Hoare



Occam

- Code wird in Occam zu Blöcken zusammengefasst, indem die einzelnen Zeilen alle gleichweit eingerückt werden
- Eine Anweisung wird durch das Ende der Zeile beendet
- Sprachelemente:

- Eingabe ? :

```
keyboard ? c
```
- Ausgabe !:

```
screen ! c
```
- Sequentielle Ausführung SEQ:

```
SEQ
  x:=1
  y:=2
```
- Parallele Ausführung PAR:

```
PAR
  keyboard ? x
  screen ! y
```
- Alternative Ausführung ALT*:

```
ALT
  x<10 & chan1 ? y
    screen ! y
  x<20 & chan2 ? y
    screen ! y
```

*Bei der ALT kann für jeden Block eine Bedingung, sowie eine Eingabe (beide optional) angegeben werden. Es wird derjenige Block ausgeführt, dessen Bedingung wahr ist und auf dem Daten eingehen. Trifft dies für mehrere Blöcke zu, so wird ein Block gewählt und ausgeführt.