

Inhaltsverzeichnis / Gliederung Vorlesung „Rechnerarchitekturen“

1. Zahlensysteme, Zahlendarstellung, Codes

- 1.1 Allgemeines Bildungsgesetz für Zahlensysteme / Umwandlungen
- 1.2 Dualzahlenarithmetik
 - 1.2.1 Addition
 - 1.2.2 Multiplikation
 - 1.2.3 Subtraktion / Komplementbildung
 - 1.2.4 Division
- 1.3 Hexadezimalsystem
 - 1.3.1 Rechnen im Hexadezimalsystem
- 1.4 Zahlendarstellung
- 1.5 Codes

Übung Zahlensysteme

Übung Codes

2. Halbleiterspeicher

- 2.1 Festwertspeicher
 - 2.1.1 ROM
 - 2.1.2 PROM
 - 2.1.3 EPROM
 - 2.1.4 EEPROM
 - 2.1.5 Flash EPROM
- 2.2 Schreib-/Lesespeicher
 - 2.2.1 SRAM
 - 2.2.2 DRAM
 - 2.2.3 Geschwindigkeitssteigerung beim DRAM-Zugriff

Übung Speicherbausteine

- 2.3 Anwendungen von Halbleiterspeichern
 - 2.3.1 Speichersysteme
 - 2.3.2 Spiegelung
 - 2.3.3 Banking
 - 2.3.4 Umcodierer
 - 2.3.5 Zustandsautomaten


Übung Speicherdekodierung + Umcodierer + Zustandsmaschine

3. Grundstruktur eines Digitalrechners

- 3.1 Aufbau eines einfachen Rechners
- 3.2 Control Unit
- 3.3 Charakterisierende Rechnereigenschaften
- 3.4 Befehls- und Adressierungsarten

4. Embedded Controller der 80186.Familie

- 4.1 Historie
- 4.2 Architekturüberblick
- 4.3 Prozessorkern
 - 4.3.1 Registersatz
 - 4.3.2 Speicheradressierung
 - 4.3.3 Stack
- 4.4 Kommunikation mit dem Speicher und externer Peripherie
 - 4.4.1 Multiplexbus
 - 4.4.2 Speicherinterface
 - 4.4.3 Aufbau eines Buszyklus
 - 4.4.4 Datenbus-Pufferung
- 4.5 Grundfunktionen für den Betrieb des Mikroprozessors
 - 4.5.1 Taktversorgung
 - 4.5.2 Reset-Erzeugung
 - 4.5.3 Power Management
- 4.6 Integrierte Peripherieschaltungen
 - 4.6.1 Peripheral Control Block
 - 4.6.2 Chip Select Unit
 - 4.6.3 I/O-Ports
 - 4.6.4 Serielle Schnittstelle

 Sprung nach Kap. 5

*

- * 4.6.5 Timer und Counter
- * 4.6.6 Interrupt Controller
- * 4.6.7 DMA Controller
- * 4.6.8 Sonstige Peripherie

5. Programmierung in Assembler

- 5.1 Entwicklung eines Assemblerprogramms
- 5.2 Aufbau eines Assemblerprogramms
- 5.3 Befehlssatz
 - 5.3.1 Adressierungsarten
 - 5.3.2 Befehle zum Datentransfer
 - 5.3.3 Befehle für arithmetische Operationen
 - 5.3.4 Logikbefehle
 - 5.3.5 Befehle zum Verarbeiten von Zeichenketten
 - 5.3.6 Sprünge und Unterprogrammaufrufe
 - 5.3.7 Befehle zur Kontrolle des Prozessors

Übung Assemblerprogrammierung

- 5.4 Das Arbeiten mit Unterprogrammen
 - 5.4.1 Aufbau und Ablauf von Unterprogrammen
 - 5.4.2 Unterprogrammbeispiel
 - * 5.4.3 Lokale Variablen und Parameterübergabe über den Stack
 - * 5.4.4 Unterprogrammbeispiel mit Stack-Parameterübergabe → Rücksprung nach Kap. 4.4
- 5.5 Strukturierte Programmierung

*: je nach zur Verfügung stehender Zeit