

Technische Informatik

Zuordnung	Rechner und Netze
Klassenstufe	Kursstufe BF und LF
Bildungsplanbezug	BF, LF
Werkzeug	MICROSIM, Logicsim, MiniJava, Minetest
Autoren	T. Schaller

Inhalt

Das Material beschreibt einen Unterrichtsgang und Hintergrundinformationen zur technischen Informatik beginnend mit der historischen Entwicklung der Rechenmaschine über Transistoren, logische Schaltungen, RS- und D-Flipflop bis zum Halb- und Volladdierer und anschließend zum 8-Bit-Addierwerk.

Themen

- Historische Entwicklung der Rechenmaschine
- Transistoren und CMOS-Schaltungen
- elementare Gatter (NOT, AND, OR, NAND)
- logische Bauteile aus CMOS-Bausteinen
- Torsteuerung
- RS-Flipflop
- D-Flipflop
- Halbaddierer
- Volladdierer
- 8-Bit-Addierwerk
- Registermaschine
- Unterrichtsgang zu Assembler-Programmierung in MICROSIM
- von-Neumann-Rechner

Bildungsplan-Bezüge

3.1.3 Rechner und Netze - Basisfach

- (8) Aufbau, Funktionsweise (unter anderem Befehlszyklus) und Komponenten (unter anderem Adressbus, Datenbus, Rechenwerk, Register, Steuerwerk, Speicherwerk) einer Von- Neumann-Modellmaschine beschreiben

- (9) Mikroprogramme für Assemblerbefehle (zum Beispiel ADD, SUB, JMP) in einer Simulationsumgebung implementieren

3.1.3 Rechner und Netze - Leistungsfach

- (2) Aufbau und Funktion von Halbaddierer und Volladdierer beschreiben und daraus in einer Simulationsumgebung einen Mehrbitaddierer erstellen
- (3) Aufbau und Funktion eines bistabilen Bauteils (zum Beispiel Latch, Flipflop) als Beispiel für einen 1-Bit-Speicher beschreiben und in einer Simulationsumgebung SR-Latch und D-Latch erstellen
- (8) Aufbau, Funktionsweise (unter anderem Befehlszyklus) und Komponenten (unter anderem Adressbus, Datenbus, Rechenwerk, Register, Steuerwerk, Speicherwerk) einer Von- Neumann-Modellmaschine beschreiben
- (9) Mikroprogramme für Assemblerbefehle (zum Beispiel ADD, SUB, JMP) in einer Simulationsumgebung implementieren
- (10) Assembler-Programme für eine Von-Neumann-Modellmaschine (zum Beispiel Multiplikation natürlicher Zahlen) in einer Simulationsumgebung implementieren