

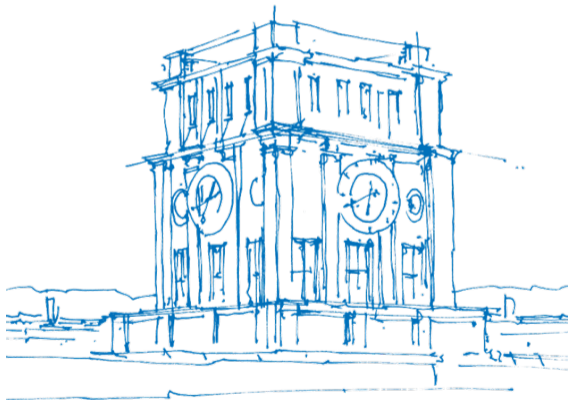
Einführung in die Rechnerarchitektur

Assembler

Lukas Hertel

Lehrstuhl für Rechnerarchitektur und Parallele Systeme
Fakultät für Informatik
Technische Universität München

11. November 2021



TUM Uhrenturm

Hausaufgabe

■ $(00011010)_2 + (00100011)_2 = (?)_2$

■ $(00110110)_2 - (00101010)_2 = (?)_2$

■ $(00011110)_2 \cdot (00000101)_2 = (?)_2$

Themen

■ Tutorblatt

- Registerbenutzung und Konstanten (MOV)
- MOV in ASM vs. Variablen in Hochsprachen
- Arithmetische Begriffe (ADD, SUB, NEG, MUL, DIV, MOD)
- Assembler zu Maschinencode

■ Hausaufgabe

- Festkommarechnung

NASM

Netwide Assembler

- Plattformunabhängiger Assembler für x86 CPU Architektur
- Kommandozeilenwerkzeug
- Manuelles Kompilieren von Code, Syntax
- SASM basiert auf NASM

SASM

SimpleASM

- Grafische Entwicklungsumgebung für ASM
- Kompilieren, Debugging, I/O
- Benötigt Assembler und C-Compiler (z.B. nasm & gcc)
- Link für die Installation

Register

Teilregister von EAX

0000 0000 0000 0000 | 0000 0000 0000 0000

|

|

Aufgabe 2

Unterschied MOV vs. Variablenzuweisung

Aufgabe 3

Arithmetische Befehle

- $\text{ADD } A, B \Rightarrow A = A + B$
 - MOV, SUB, AND, OR
- $\text{NEG } A \Rightarrow A = -A$
 - INC, DEC
- $\text{MUL } A \rightarrow \text{EDX} : \text{EAX} = \text{EAX} * A$
- $\text{DIV } A \rightarrow \text{EDX} : \text{EAX} = \text{EDX} : \text{EAX} / A$
- $\text{IMUL (Signed multiply)}$
 - $A \rightarrow \text{EDX} : \text{EAX} = \text{EAX} * A$
 - $A, B \rightarrow A = A * B$
 - $A, B, C \rightarrow A = B * C$

Aufgabe 4

Von Assembler zu Maschinencode