

# Programmiermodell 1

Danielle Sandra Mopong Fogue|Moez Rjiba

Dienstag 30.06.2020

- 1 Einführung
- 2 Operationsmodi
- 3 Instruction Set

## Prinzip

- Ein Programmiermodell auch Befehlssatzarchitektur (englisch Instruction Set Architecture, kurz: ISA) ist die abstrakte Beschreibung der Verhaltensweise eines Rechners in Bezug auf seinen Befehlssatz. Dabei wird die Befehlssatzarchitektur so formuliert, dass sie unabhängig von einer bestimmten Implementierung ist. Die ISA ist zudem Teil von Prozessorarchitekturen wie z. B. x86, ARM oder PowerPC und trägt in der Regel deren Namen (z. B. x86-Befehlssatz).

## Prinzip

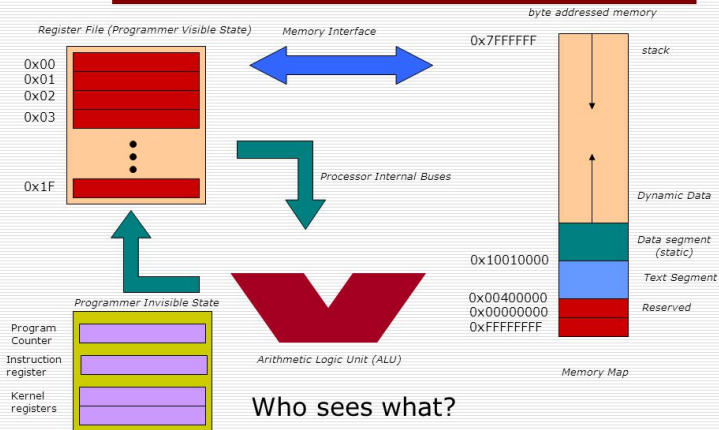
- Eine Befehlssatzarchitektur wird durch eine formale Beschreibung spezifiziert. Sie gibt vor allem einem Assemblersprache-Programmierer die Möglichkeit, das einheitliche Verhalten von Maschinencode für verschiedene Implementierungen einer bestimmten ISA (Mikroarchitekturen oder virtuelle Maschinen) in Bezug auf Register, Datentypen etc. nachzuvollziehen. Damit kann er oder sie binärkompatible Programme für verschiedene Prozessoren erstellen, wenn sie dieselbe Befehlssatzarchitektur verwenden.

## Zusammenfassung:

- Das Programmiermodell beschreibt den Aufbau eines Prozessors als abstraktes Modell. Dazu gehört:
  - Beschreibung der Befehle, die der Prozessor ausführen kann
  - Codierung der Befehle
  - Adressierungsarten
  - Aufbau der Register
  - Zeitverhalten (Anzahl der benötigten Taktzyklen)



## Instruction Set Architecture (ISA)



(60)

- Intel x86 (CISC Architektur)
- ARM (Advanced RISC Machine)
- MIPS (Mikroprozessor ohne verriegelte Pipeline-Stufen)

- ARM Operationsmodi
  - Der ARM-Prozessor hat sieben Operationsmodi

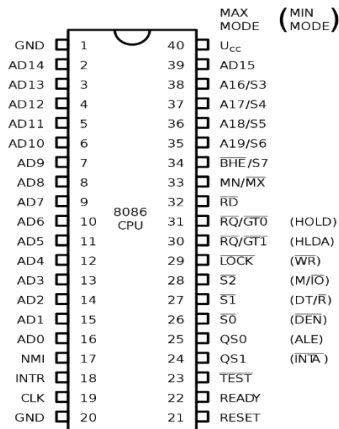
Mode	Privileged	Purpose
User	No	Normal operating mode for most programs (tasks)
Fast Interrupt (FIQ)	Yes	Used to handle a high-priority (fast) interrupt
Interrupt (IRQ)	Yes	Used to handle a low-priority (normal) interrupt
Supervisor	Yes	Used when the processor is reset, and to handle the software interrupt instruction swi
Abort	Yes	Used to handle memory access violations
Undefined	Yes	Used to handle undefined or unimplemented instructions
System	Yes	Uses the same registers as User mode

Table 1: ARM Processor Operating Modes



# Operationsmodi

- x86 Operationsmodi
  - Beispiel Intel 8086 Mikroprozessor mit zwei Operationsmodi



8086 Microprocessor is 40 pin Integrated Circuit (IC)

8086 kann in zwei Modi betrieben werden: Der Maximum-Modus wird auch als MAX-Modus bezeichnet, und der Minimum-Modus oder MIN-Modus

- Der Betrieb im MAX-Modus bedeutet, dass es ein Mehrprozessorsystem unterstützt. Daher kann es mit mehr als einem Prozessor arbeiten. In diesem System kann der Prozessor also seine Arbeit an andere Komponenten übergeben, die sich wie ein Prozessor verhalten können. Wann immer so viele Instruktionen zu diesem Zeitpunkt ausgeführt werden müssen, kann der Prozessor den Systembus an andere Komponenten übergeben.
- Der Betrieb im Minimum-Modus bedeutet, dass es sich nur um ein Ein-Prozessor-System handelt. Nur ein Prozessor, der die gesamte Operation ausführen kann. Wann immer das System also einen einzelnen Prozessor hat, können wir es als Minimum-Modus bezeichnen.

Der Befehlssatz (englisch instruction set) eines Prozessors ist in der Rechnerarchitektur die Menge der Maschinenbefehle, die ein bestimmter Prozessor ausführen kann. Je nach Prozessor variiert der Umfang des Befehlssatzes zwischen beispielsweise 33 und über 500 Befehlen.

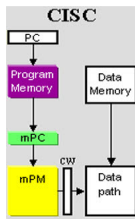
CISC-Prozessoren haben tendenziell größere Befehlssätze als RISC-Prozessoren, gleiches gilt für neuere Prozessoren gegenüber älteren Prozessoren.

- CISC (Complex Instruction Set Computer)
- RISC (Reduced Instruction Set Computer)

# Instruction Set

## CISC (Complex Instruction Set Computer)

- benötigt im Vergleich weniger Programmspeicher, da die einzelnen Befehle komplexer sind und dadurch "mehr" ausführen können
- Meist benötigt ein Befehl mehrere Takte
- Die Realisierung eines CISC Prozessors wird oft mittels Mikrocode realisiert. Zur Ausführung eines Befehl führt das Steuerwerk den entsprechenden Mikrocode aus. Dieser Mikrocode steuert den Datenpfad des Prozessors an. Dadurch können komplexe Befehle realisiert werden.

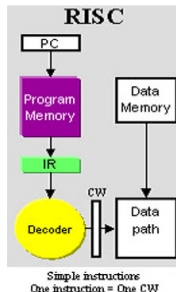


Complex instructions  
One instruction = several CW  
PM standard

# Instruction Set

## RISC (Reduced Instruction Set Computer)

- Für die meisten Prozessoren mit RISC-Befehlssatz gilt:
  - Load-Store Befehle: Speicherzugriffe erfolgen (nur) durch Load bzw. Store Befehle. Dies vereinfacht den Datenpfad
  - Befehlsausführung meist in einem Taktzyklus
  - Sehr viele frei verwendbare Register
  - Befehle haben meist eine feste Länge
  - Steuerwerk ist festverdrahtet (vgl. CISC mit Mikrocode)
  - unabhängige Verarbeitungseinheiten ermöglichen Pipelining



## Befehlsarten

- Der Befehlssatz eines Mikroprozessors setzt sich grob aus wenigen Arten von Befehlen zusammen.
  - Transferbefehlen: werden Daten innerhalb des Systems bewegt. Als Quelle und als Ziel kommen dabei Speicherstellen oder Prozessorregister in Frage (move, load, store, transfer)
  - Arithmetische und logische Befehle: ADD AND
  - Ein-/Ausgabebefehle dienen zum Lesen und Schreiben von Daten über Peripheriegeräte, die Verbindungsmedien zur „Außenwelt des Computers“.
  - Sprungbefehle führen zu Verzweigungen innerhalb eines Programmablaufs, zum Bilden von Schleifen, Reaktionen auf verschiedene Zahlenwerte und Aufruf von Unterprogrammen (Subroutinen). JUMP
  - Stack-Befehle: speichern Daten auf den Stack PUSH

- Links

- <https://csschoolonline.com/8086-microprocessor-operation-modes-features/>
- <https://slideplayer.com/slide/3338248/>
- <https://www.semiversus.com/dic/rechnerarchitektur/programmiermodell.html/>

Danke für ihre Aufmerksamkeit