

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [Straße Hausnummer] [PLZ Ort] |  | T.[Telefon]  F.[Fax] |  | [E-Mail-Adresse]  [Internetadresse] |

Inhaltsverzeichnis

1. [Zusammenfassung 2](#_Toc447794794)
2. [Aufgabenbeschreibung 2](#_Toc447794795)

[Speichwerk](#_Toc447794796)

[Speichwerk](#_Toc447794797)

[\_Toc447794798](#_Toc447794798)

# Prinzipieller Aufbau des Leitwerks

Hier soll der generelle Aufbau des Leitwerks veranschaulicht werden, wie er in der Von-Neumann-Architektur vorzufinden ist. Die einzelnen Elemente des Leitwerks werden in den folgenden Unterkapiteln erklärt.

OPCode

HighByte

LowByte

Befehlsdecoder

Steuerwerk

BZ - Register

MUX

SAR

+1

StatusFlags

Steuersignale

Vom Rechenwerk

Zum Rechenwerk

Datenbus

Adressbus

*Befehlsregister*

## Befehlsregister

OpCode, High-byte und Low-Byte ergeben zusammen das Befehlsregister. Insgesamt umfasst jedes Register 8 Bit, somit ist das komplette Befehlsregister 24 Bit breit. Das Befehlsregister ist an den Datenbus angeschlossen.

### OPCode

Der OpCode beschreibt die den auszuführenden Befehl. Dieser ist kodiert und wird an den Befehlsdecoder weitergegeben. Da der OpCode maximal 8 Bit lang sein kann, besitzt der Befehlssatz maximal 256 Befehle.   
In diesem Projekt wird jedoch nur ein kleiner Befehlssatz codiert werden, der die nötigsten Befehle unterstützt.

### High-Byte und Low-Byte

Diese Erweiterung des Befehlsregisters wird nur bei Befehlen verwendet, die eine Adresse benötigen. Also jegliche Art von Jump-Befehlen und der Move-Befehl. Da der Adressbus aus 16 Bit Breit ist, also aus 2 Byte müssen 2 Register à 1 Byte vorgesehen werden. Dies kommt daher, dass der Datenbus nur eine Breite von 8 Bit besitzt.

## Befehlsdecoder und Steuerwerk

Der Befehlsdecoder liest den OpCode ein und entscheidet je nach OpCode um welchen Befehl es sich handelt. Hier kann schon eine Schätzung gemacht werden, ob es sich um einen Sprungbefehl handelt, und somit High-Byte und Low-Byte benötigt werden.   
Das Steuerwerk liest die Statusregister ein und entscheidet ausgehend von dem decodierten Befehl und den Statusregistern welche Steuersignale an das Rechenwerk gesendet werden sollen.   
Weiterhin steuert das Steuerwerk das Source-Adress-Register und den MUX, auf diese beiden Elemente wird später noch eingegangen.

# Aufgabenbeschreibung

Spezifikationen des Prozessors:

Datenbus: 16 Bit  
Adressbus: 8 Bit

**Befehlsausführung im Leitwerk:**

1. Befehl aus dem Speicher ins Befehlsregister laden (FETCH CYCLE)
2. Befehlsentschlüsselung (DECODE CYCLE)
3. Laden der Operanden (Operand FETCH)
4. Befehlsausführung (Execution Cycle)
5. Rückschreiben der Ergebnisse (Write Back)

I

II

III

IV

IV

Jedoch muss vor dem „normalen“ Programmablauf einen geeigneter Einsprung gemacht wird, d.h. der Befehlszähler (BZ) bzw. das Source-Adress-Register (SAR) muss mit der entsprechenden Einsprungadresse in der das Programm, welches ausgeführt werden soll initialisiert werden.