

## Projekt: von-Neumann-Architektur

### Rollenspiel<sup>1</sup> zur Simulation eines einfachen Computers im Sinne der von-Neumann-Architektur

In (fast) allen Computern werden die Daten binär verwaltet und verändert. Ein einfaches Computermodell soll die Funktionsweise erläutern.

1. Der Modellcomputer verfügt über einige Schachteln, die mit Hilfe von Binärzahlen adressiert (also angesprochen) werden können. Als Adresse sind 4 Bit vorgesehen. Also können insgesamt 16 Schachteln verwaltet werden.
2. In jeder Schachtel befindet sich eine mit 8 Bit binär-codierte Zahl.
3. Der Inhalt einer Schachtel kann nur bei geöffneter Schachtel gelesen werden. Ein Wert darf nur in eine geöffnete Schachtel geschrieben werden.

An der Simulation nehmen mehrere „Akteure“ teil:

Name des Akteurs	Beschreibung der Tätigkeit
Adressierer	Kann auf Anweisung des Leiters eine Schachtel öffnen. Er kann den Inhalt nicht lesen. Die Nummer der zu öffnenden Schachtel schreibt ihm der Leiter oder der Schachtelzähler auf einen Zettel.
Datenbote	Er transportiert Zahlen per Zettel zwischen Schachteln und Rechenknecht, kann Zahlen in die Schachteln schreiben und aus ihnen abschreiben (lesen).  Nach jedem Schreib/Lesevorgang schließt er die Schachtel.
Bildschirm	Kann einen beschriebenen Zettel (für die Zuschauer sichtbar) hochhalten.
Schachtelzähler	Merkt sich auf einem Zettel die Adresse der Schachtel mit dem nächsten Befehl. Kann vom Simulationsleiter Schachtelnummern übernehmen oder auf Anweisung seine Nummer um 1 erhöhen. Bei Programmbeginn steht auf seinem Zettel eine 0.
Rechenknecht	Kann vom Datenboten Zahlen übernehmen, dem Datenboten Zahlen übergeben oder eine Zahl zu der Zahl auf seinem Zettel addieren.
Simulationsleiter	Gibt allen anderen die nötigen Anweisungen. Kann Befehle, die ihm der Datenbote aus den Schachteln besorgt, durch Nachsehen in der unten angegebenen Tabelle in die entsprechenden Anweisungen für seine Untergebenen übersetzen.  Wenn er einen Befehl erhalten hat, veranlasst er den Schachtelzähler, dessen Nummer um 1 zu erhöhen.

Tabelle 1: Akteure im Rollenspiel

Liest der Simulationsleiter in der Zelle, deren Adresse der Schachtelzähler momentan verwaltet, eine Binärzahl, zerlegt er sie in zwei Teile. Die ersten 4 Binärstellen codieren einen Befehl (s.u.). Die folgenden 4 Binärstellen stellen die Adresse einer Schachtel dar.

Er benutzt dann diese Tabelle für die Interpretation der Befehle:

Codierung des Befehls	Bedeutung
0010	Lese den Inhalt der Schachtel mit derjenigen Adresse, die als zweiter 4-Bit-Wert angegeben ist, und übergib den Inhalt dem Rechenknecht.  Kurzform: <b>load</b>
0011	Speichere den Inhalt des Rechenknechtzettels in der Schachtel mit derjenigen Adresse, die als zweiter 4-Bit-Wert angegeben ist.  <b>Kurzform: store</b>
0100	Addiere den Inhalt der Schachtel mit der Adresse, die als zweiter 4-Bit-Wert angegeben ist, zum Inhalt des Rechenknechtzettels. Die Summe wird auf dem Rechenknechtzettel (anstatt der bisherigen Zahl) notiert.  <b>Kurzform: add</b>
1011	Übergib den Inhalt derjenigen Schachtel mit der Adresse, die als zweiter 4-Bit-Wert angegeben ist, dem Bildschirm.  <b>Kurzform: out</b>
1100	Beende das Programm. (Der zweite 4-Bit-Wert ist hier irrelevant.)  <b>Kurzform: stop</b>

*Tabelle 2: Befehle*

Die Ausführung des jeweiligen Befehls wird der Simulationsleiter dann mit der Hilfe der oben beschriebenen Akteure erledigen.

### Beispiel:

In einer Schachtel liegt die Binärzahl 00110110. Der Leiter ist gerade dabei, diese Binärzahl zu interpretieren und eine entsprechende Aktion zu veranlassen.

Die ersten 4 Binärstellen (0011) interpretiert er als **store** (siehe obige Tabelle). Die restlichen 4 Binärstellen (0110) bilden die Adresse einer Schachtel.

Jetzt werden folgende Aktionen stattfinden:

1. Der Leiter fordert den Adressierer auf, die Schachtel mit der Adresse **0110** zu öffnen.
2. Der Leiter fordert den Dienstboten auf, vom Rechenknecht dessen 8-Bit-breite Binärzahl zu holen, in die Schachtel mit der Adresse **0110** zu legen und diese Schachtel zu schließen.
3. Der Leiter fordert den Adressierer auf, den Schachtelzähler um 1 zu erhöhen.

Anschließend liest der Leiter den Inhalt der Schachtel mit derjenigen Adresse, die jetzt der Schachtelzähler notiert hat, und interpretiert die in dieser Schachtel abgelegte Zahl.

Die Binärzahl **0011**, als Befehl **store** interpretiert, ist damit zerlegbar in kleine, sogenannte Mikrobefehle, die von den jeweils zuständigen Akteuren ausgeführt werden.

Auf einer höheren Abstraktionsebene kann der Befehl **00110110** auch mit **save 6** bezeichnet ( $0011 \approx \text{store}$  und  $0110_{(\text{bin})} \approx 6_{(\text{dez})}$ ), und beschrieben werden in der Form:

Lege in die Schachtel mit der Adresse  $6_{\text{dez}}$  (bzw.  $0110_{\text{bin}}$ ) die Binärzahl, die zur Zeit der Rechenknecht hat.

**Aufgabe:** Es gibt (siehe Tabelle 2) insgesamt 5 „Makrobefehle“. Zerlegen Sie diese Befehle in Mikrobefehle, indem Sie die folgende Tabelle ergänzen (xxxx ist eine 4-Bit-breite Binärzahl, X ist die zu xxxx gehörige Dezimalzahl):

Makrobefehl	Mikrobefehlsfolge
0010 xxxx	<b>load X</b> <ul style="list-style-type: none"><li>• ???</li></ul>
0011 xxxx	<b>store X</b> <ul style="list-style-type: none"><li>• Schachtel mit Adresse x vom Adressierer öffnen lassen.</li><li>• Wert des Rechenknechtes vom Datenboten in die Schachtel mit Adresse x legen und Schachtel schließen lassen.</li><li>• Befehlszähler um „1“ beim Adressierer erhöhen lassen.</li></ul>
0100 xxxx	<b>add X</b> <ul style="list-style-type: none"><li>• ???</li></ul>
1011 xxxx	<b>out X</b> <ul style="list-style-type: none"><li>• ???</li></ul>
1100 xxxx	<b>stop</b> (Nicht zerlegbar in Mikrobefehle; der Wert von X ist irrelevant)

Tabelle 1: Mikrobefehle