# **Projekt: von-Neumann-Architektur**

## Lösung zum Arbeitsblatt 1 zum Rollenspiel

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr. der Schachtel** | **Inhalt der Schachtel** | **Macrobefehl** | **Microschritte** |
| 0000 | 0010 0101 | **load 5** | A öffnet Schachtel 5  D transportiert Inhalt von Schachtel 5 zum RK  RK legt den Wert ab; RK hat den (dezimal geschriebenen) Inhalt 23 |
| 0001 | 0100 0110 | **add 6** | A öffnet Schachtel 6  D transportiert Inhalt von Schachtel 6 zum RK  RK addiert den Wert bei sich hinzu; RK hat den (dezimal geschriebenen) Inhalt 39 |
| 0010 | 0011 0111 | **store 7** | A öffnet Schachtel 7  D transportiert Inhalt von RK zu Schachtel 7; in Schachtel 7 ist jetzt die Binärzahl 0010 0111, also dezimal 39 |
| 0011 | 1011 0111 | **out 7** | A öffnet Schachtel 7  D transportiert Inhalt Schachtel 7 zu B  B zeigt Wert (also 39) an |
| 0100 | 1100 0000 | **stop** | Programm wird beendet |
| 0101 | 0001 0111 | Datenschachtel, wird nicht als Befehl interpretiert. |  |
| 0110 | 0000 1000 | dto. |  |
| 0111 | 0000 0000 | dto |  |

(A=Adressierer, D=Datenbote, B=Bildschirm, S=Schachtelzähler, RK=Rechenknecht)

Funktionalität:

Das Programm addiert die Inhalte der Schachteln (dezimal) 5 und 6 und legt die Summe in Schachtel 7 ab. Diese Summe wird angezeigt.

Abkürzen könnte man die Aktion durch den Rechenterm S7 = S5 + S6 im Sinne einer höheren Programmiersprache. In der Schachtel 8 befindet sich dann der Wert (dezimal) 31, bzw. der BitString 0001 1111.