

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

### **Arbeitsblatt 4 zum Schachtelcomputer**

Bisher kann der Schachtelcomputer ausschließlich lineare Programme abarbeiten. Um Programme mit Fallunterscheidungen und Schleifen abarbeiten zu können, sind zwei neue Befehle vorgesehen. Nehmen wir außerdem weitere Befehle zum „Rechnen“ (Subtrahieren, Multiplizieren, ganzzahlig Dividieren, ganzzahlige Restbildung) hinzu, ergibt sich die folgende erweiterte Tabelle:

<b>Codierung des Befehls</b>	<b>Bedeutung</b>	<b>Kurzform</b>
<b>0000</b>		<b>data</b>
<b>0001</b>	Der Adressteil wird als positive Zahl interpretiert und in den AKKU transportiert.	<b>lint</b>
<b>0010</b>	Lese den Inhalt der Speicherzelle mit der angegebenen Adresse und speichere den Wert im AKKU	<b>load</b>
<b>0011</b>	Speichere den Inhalt des AKKUs in der Speicherzelle mit der angegebenen Adresse.	<b>save</b>
<b>0100</b>	Addiere den Inhalt der Speicherzelle mit der angegebenen Adresse zum Inhalt des AKKUs, speichere die Summe im AKKU.	<b>add</b>
<b>0101</b>	Subtrahiere den Inhalt der Speicherzelle mit der angegebenen Adresse vom Inhalt des AKKUs, speichere die Differenz im AKKU.	<b>sub</b>
<b>0110</b>	Multipliziere den Inhalt des AKKUs mit dem Inhalt der Speicherzelle mit der angegebenen Adresse	<b>mul</b>
<b>0111</b>	Dividiere den Inhalt des AKKUs ganzzahlig durch den Inhalt der Speicherzelle mit der angegebenen Adresse	<b>div</b>
<b>1000</b>	Der Inhalt des AKKUs wird zum Rest der ganzzahligen Division von Inhalt von AKKU und Inhalt der Speicherzelle mit der angegebenen Adresse	<b>mod</b>
<b>1001</b>	Der Programmzähler wird auf den Wert der Speicherzelle mit der angegebenen Adresse gesetzt.	<b>jmp</b>
<b>1010</b>	Der Programmzähler wird auf den Wert der Speicherzelle mit der angegebenen Adresse gesetzt., falls der Inhalt des AKUs den Wert 0000 hat. Ansonsten wird das Programm an der folgenden Adresse fortgesetzt.	<b>jeq</b>
<b>1011</b>	Gib den Inhalt der Speicherzelle mit der angegebenen Adresse auf einem Ausgabemedium aus.	<b>out</b>
<b>1100</b>	Beende das Programm. (Die Adresse ist hier irrelevant)	<b>stop</b>

Tabelle 1: geänderte Befehlskodierungen

Bisher hat der Schachtelcomputer maximal 16 Schachteln (da der 4-Bit-breite Adressteil eines Befehls maximal 16 Adressen codieren kann). Damit auch Programme mit mehr als 16 Befehlen zu verwalten sind, soll der Adressteil mit 12 Bit codiert werden, so dass jetzt maximal 4096 Adressen (Schachteln) angesprochen werden können.

Wenn man also z.B. den Wert, der in der Zelle  $20_{\text{dezimal}} = 0000\ 0001\ 0100_{\text{binär}}$  enthalten ist, zum Wert, der auf dem Zettel vom Rechenknecht steht, addieren möchte, wird der Befehl codiert mit

0100 0000 0001 0100

Die Schachteln enthalten jetzt also 16-Bit-breite Binärstrings. Die ersten 4 Bit codieren den Befehl, die folgenden 12 Bit die angesprochene Adresse.

### **Aufgabe 1:**

In zwei Zellen des Schachtelcomputers befindet sich jeweils eine Zahl (in binärer Codierung). Der größte gemeinsame Teiler dieser beiden Zahlen soll errechnet werden.

Entwerfen Sie dazu ein geeignetes von-Neumann-Programm.

### **Aufgabe 2:**

Erweitern Sie Ihren Modellrechner so, dass bedingte und unbedingte Sprünge sowie die fünf arithmetischen Operationen verarbeitet werden können.