

Funktionsweise eines von-Neumann-Rechners I

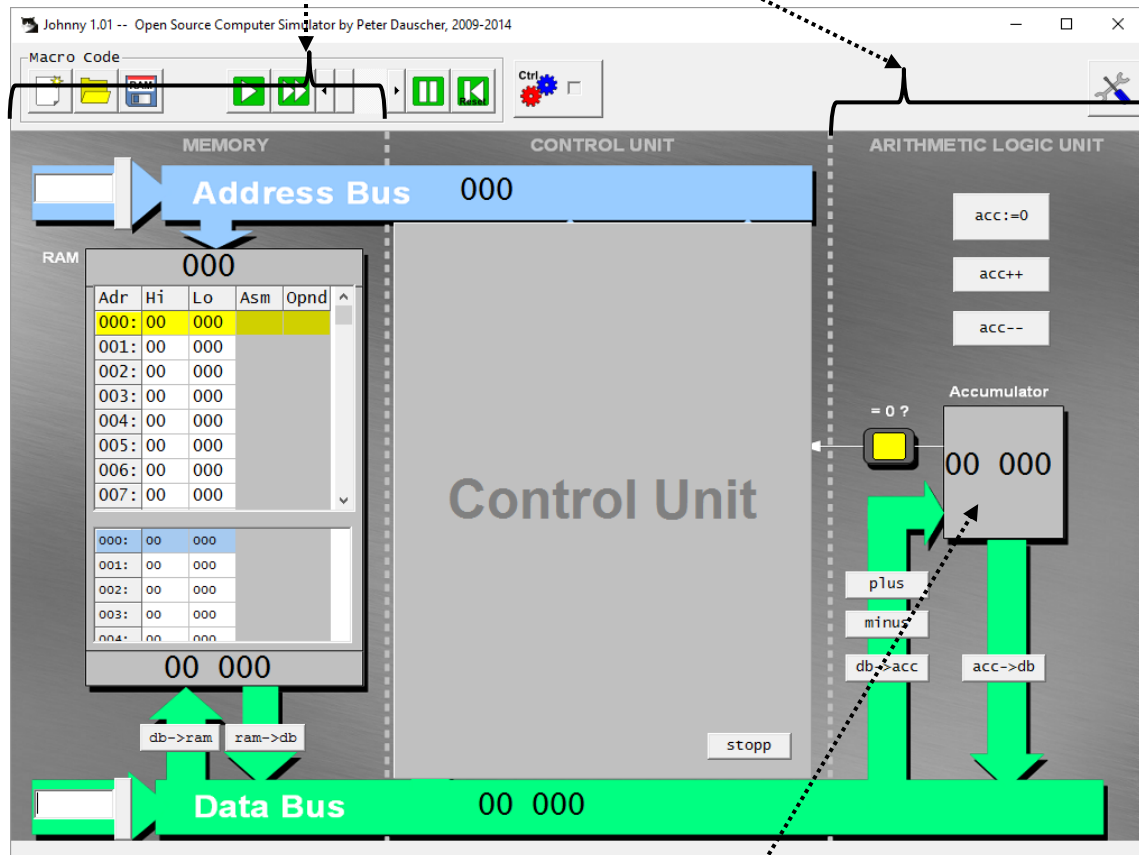
Das Programm Johnny* simuliert einen von-Neumann-Rechner. Das bedeutet, dass du die prinzipielle Funktionsweise nachvollziehen kannst, wie moderne Computer Daten verarbeiten.

- Der Rechner kann nur Zahlen (alphanumerisch codierte Daten) verarbeiten.

Dabei gibt es aber auch kleine Unterschiede zur Wirklichkeit.

- In einem „echten“ Computer werden Daten binär dargestellt. Weil du aber mit Dezimalzahlen besser umgehen kannst als mit Dualzahlen, funktioniert Johnny mit dezimal codierten Daten.

In der Abbildung unten siehst du die Benutzeroberfläche des Programms. Für den Anfang besteht der Rechner vor allem aus dem **Speicher (Memory)** und dem **Rechenwerk (Arithmetic Logic Unit – kurz ALU)**.



Das Rechenwerk beinhaltet einen Speicher, den **Akkumulator**. Dieser kann auf Null oder auf den Wert der Zahl im Datenbus gesetzt sowie der Wert um Eins erhöht oder verringert werden. Auch kann der Wert im Datenbus zu dem Wert im Akkumulator addiert oder davon subtrahiert werden. Dabei kann die Subtraktion auf eine Addition zurückgeführt werden.

- * Johnny ist ein Programm zur Simulation eines von-Neumann-Rechners. Es handelt sich bei JOHNNY um ein Open-Source-Projekt unter der GNU Public License (GPL3). Download der aktuellen Version: <https://sourceforge.net/projects/johnnysimulator/> Hier kann man auch die Projektquellen herunterladen. Unter den folgenden beiden Links sind auch weitere Ausarbeitungen zu finden:

- <http://www.inf-schule.de/rechner/johnny>
- https://wiki.zum.de/wiki/Rechnerarchitektur_mit_Simulator_JOHNNY

Mit freundlicher Genehmigung des Programmautors befindet sich eine Version des Programms im Ordner 18\18-materialien\tools\Johnny\johnny_v_1_01_inst_2014_04_27a\ - das Programm kann durch Öffnen der Datei `johnny.exe` gestartet werden.



1.8 Grundlagen elektronischer Datenverarbeitung

Arbeitsblatt 05 Funktionsweise eines von-Neumann-Rechners I

Lösungen

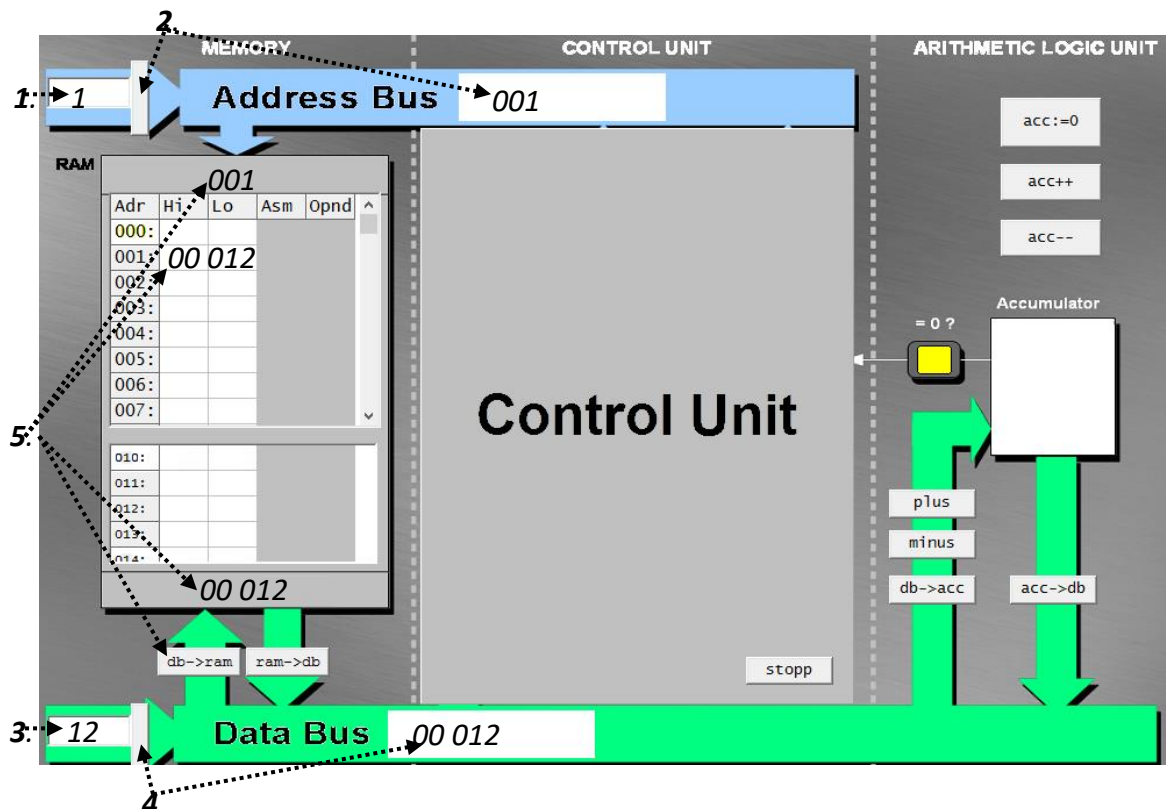
- Der Arbeitsspeicher wird **RAM** genannt (**R**andom **A**ccess **M**emory – sinngemäß übersetzt Speicher für wahlfreien Zugriff). Man kann also beliebig auf jede Zelle des Speichers zugreifen.
 - Der Speicher besteht aus in der Spalte **Adr** fortlaufend nummerierten Speicherzellen.
 - Eine Speicherzelle kann eine Zahl zwischen 0 und 19999 aufnehmen. Diese Zahl wird in **Hi**(gh) und **Lo**(w) aufgeteilt: In die Spalte **Low** werden die Einer, Zehner und Hunderter eingetragen, in **Hi** die Tausender und Zehntausender.
- Um später die Ergebnisse von Rechnungen schneller erkennen zu können, ist hier noch ein zweites Speicherfenster vorhanden.

1. Trage in den Speicher rechts die Zahl 12345 an der Adresse 004 ein.

RAM					
Adr	Hi	Lo	Asm	Opnd	^
000:					
001:					
002:					
003:					
004:	12	345			
005:					
006:					
007:					
...					
010:					
011:					
012:					
013:					
014:					

Wenn in einem von-Neumann-Rechner eine Zahl in den Speicher geschrieben werden soll, ist das etwas umständlicher.

2. Starte das Programm Johnny. Du benötigst am PC aber auch dieses Arbeitsblatt und einen Stift. Zunächst soll die Zahl **12** in die Speicherzelle **001** geschrieben werden. Führe die dazu erforderlichen Schritte am PC aus und ergänze die angezeigten Werte handschriftlich in der Grafik unten. Die Reihenfolge der Schritte ist auf dem Arbeitsblatt nummeriert.
- Trage dazu **1.** die Adresse 001 in das Adressfeld  ein und kopiere sie **2.** durch Klick auf  in den Adressbus.
- Dann muss **3.** die Zahl 12 in das Eingabefeld  unten links neben dem Datenbus geschrieben und **4.** durch Klick auf die Schaltfläche  in den Datenbus kopiert werden.
- Zum Schluss überträgt **5.** der Rechner die Zahl durch Klick auf  aus dem Datenbus in die Speicherzelle, die in dem Adressbus eingetragen ist.





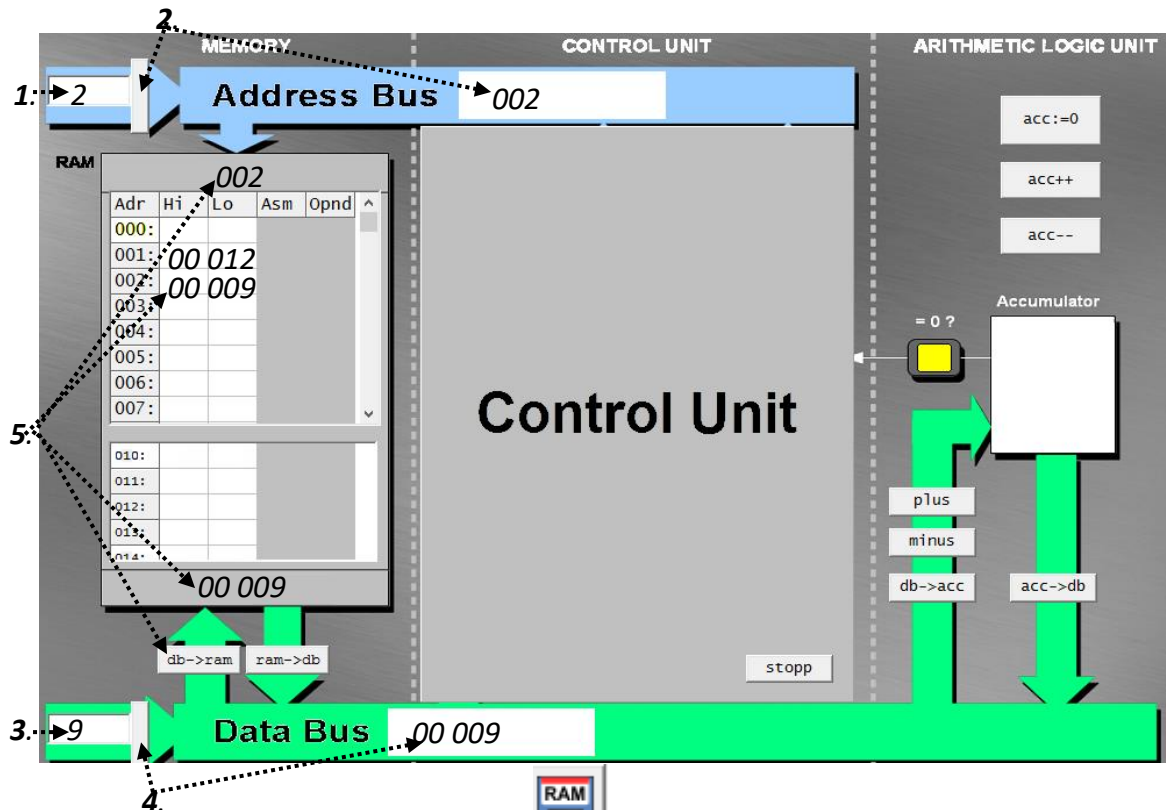
1.8 Grundlagen elektronischer Datenverarbeitung

Arbeitsblatt 05 Funktionsweise eines von-Neumann-Rechners I

Lösungen

3. Trage die Zahl **9** in die Speicherzelle **002** ein.

Führe die dazu erforderlichen Schritte am PC und handschriftlich in der Grafik unten aus und nummeriere auf dem Arbeitsblatt die Reihenfolge der Schritte.



4. Speichere den RAM mit der Schaltfläche  und beende Johnny.
(18\18-materialien\johnny-dateien\01-12und9imRAM.ram)

Wie gesagt, werden alle Daten in einem „echten“ Computer binär codiert. Johnny arbeitet nur deshalb mit Dezimalzahlen, weil das wesentlich übersichtlicher ist. Für den Menschen ist es im Prinzip egal, in welchem Zahlensystem er rechnet.

5. Stelle die Dezimalzahlen 12 und 9 im Dualsystem dar.

$$12_{10} = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 1100_2 \quad | \quad 9_{10} = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 1001_2$$

6. Rechne $12 + 9$ schriftlich im Dezimalsystem und im Dualsystem.

Dezimalsystem:	Dualsystem:
$\begin{array}{r} 12 \\ + 9 \\ \hline 21 \end{array}$	$\begin{array}{r} 1100 \\ + 1001 \\ \hline 10101 \end{array}$

Ergebnis: $12_{10} + 9_{10} = 21_{10}$ und $1100_2 + 1001_2 = 10101_2$

7. Begründe, warum der Computer nur im Dualsystem rechnen kann.

Logische Schaltungen basieren auf Ein-Aus-Schaltern.

Damit lassen sich genau zwei Zustände darstellen, die 0 und die 1.

Mit den 10 Ziffern des Dezimalsystems zu rechnen, wäre technisch enorm aufwändig und ist auch nicht erforderlich, weil ein Computer mit der 0 und der 1 sehr gut rechnen kann.

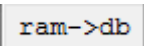


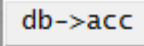
1.8 Grundlagen elektronischer Datenverarbeitung

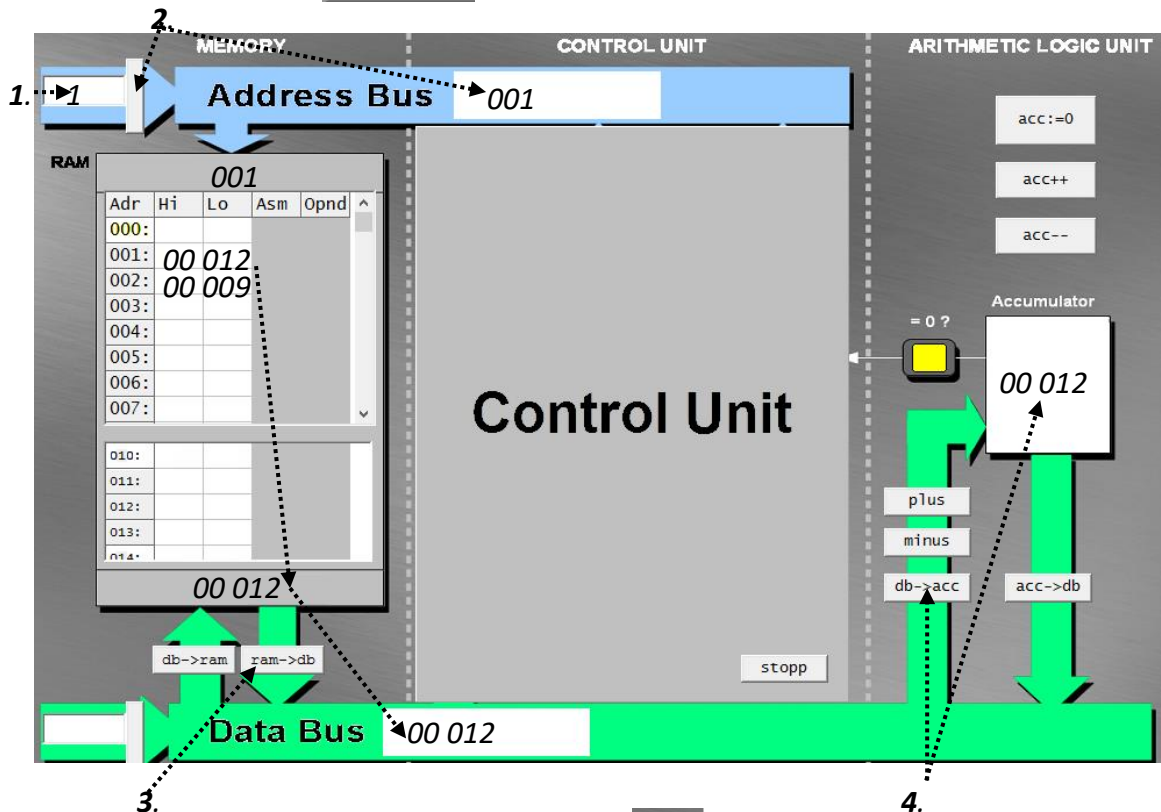
Jetzt wollen wir herausfinden, wie Johnny die Rechnung $12 + 9$ ausführt. Dafür wird das **Rechenwerk (ALU)** benötigt.

8. Öffne den zuletzt gespeicherten RAM mit der Schaltfläche . Jetzt sind die Busse wieder leer.

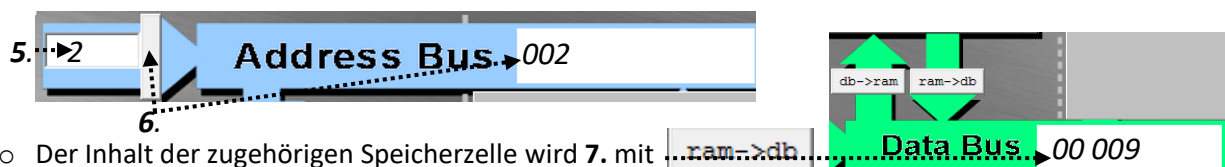
Die Zahlen in den Speicherzellen mit den Adressen 001 und 002 sollen addiert werden.

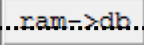
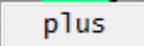
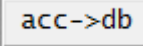
Jetzt muss **1.** und **2.** der Adressbus mit der ersten Adresse belegt werden. Dann wird **3.** der Inhalt der Speicherzelle mit der Schaltfläche  auf den Datenbus kopiert.

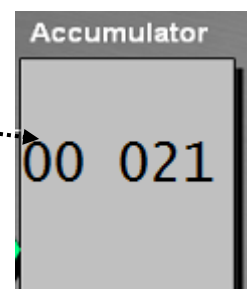
- Der wichtigste Bestandteil der ALU ist der **Akkumulator**. Hier werden Rechenergebnisse gespeichert.
- Mit der Schaltfläche  wird **4.** der Inhalt des Datenbusses in den Akkumulator kopiert.

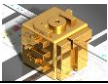


- Mit der Kontrolllampe links vom Akkumulator  wird angezeigt, ob dieser den Wert 0 hat. Die Lampe wechselt jetzt die Farbe auf grau.
- Dann muss **5.** und **6.** Der Adressbus mit der Adresse des zweiten Summanden belegt werden.

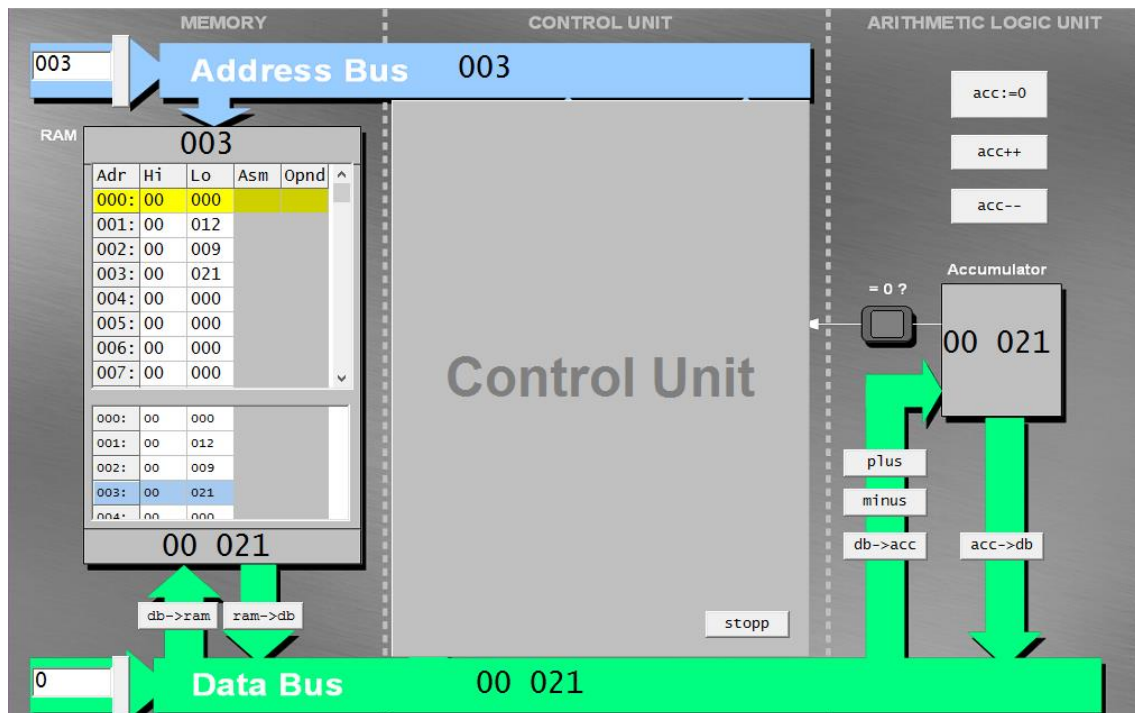


- Der Inhalt der zugehörigen Speicherzelle wird **7.** mit  auf den Datenbus kopiert.
- Jetzt kann der Inhalt des Datenbusses mit der Schaltfläche  mit dem aktuellen Wert des Akkumulators addiert werden.
- Da man das Ergebnis später eventuell nochmal verwenden möchte, ist es sinnvoll, die Zahl in der Speicherzelle mit der Adresse 003 zu sichern. Dazu ist der Inhalt des Akkumulators mit  auf den Datenbus zu kopieren, die Adresse 003 im Adressbus einzutragen und der Inhalt des Datenbusses in die zugehörige Speicherzelle zu kopieren.



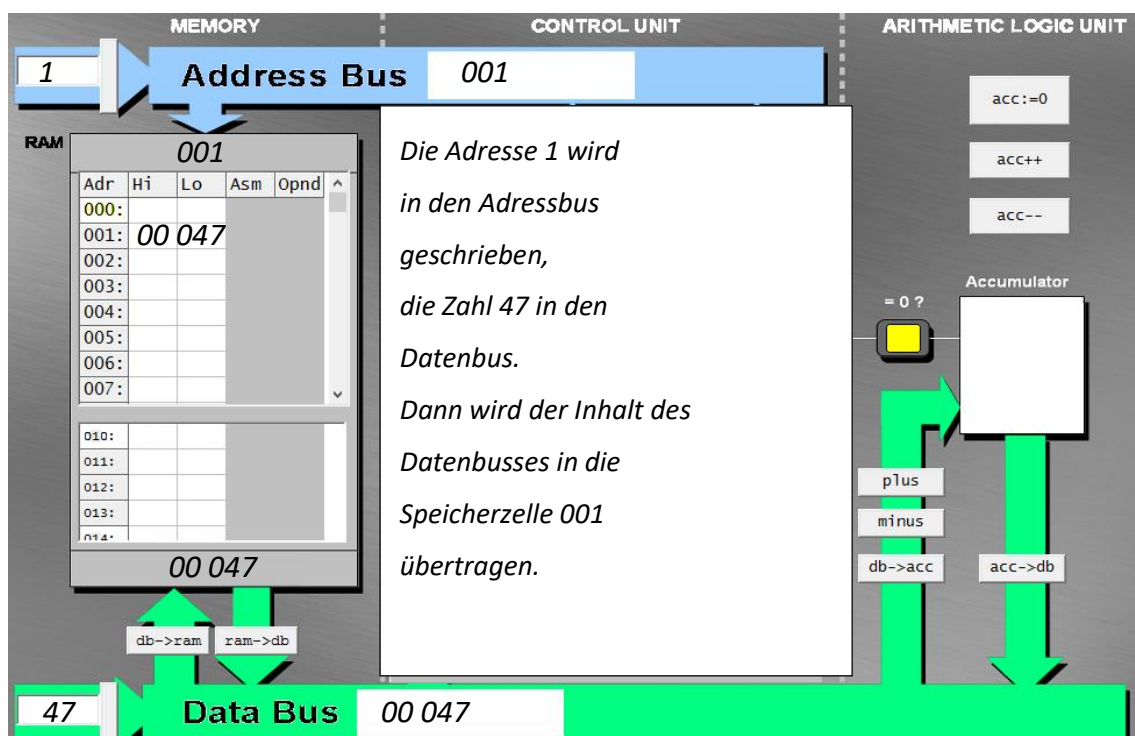


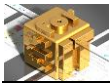
Das Ergebnis sieht dann in der Simulation so aus: (18\18-materialien\johnny-dateien\02-12plus9.ram)



9. Die beiden Zahlen 47 und 11 sollen in die Speicherzellen 001 und 002 geschrieben und dann mit Hilfe der Simulation eines von-Neumann-Rechners addiert werden. Das Ergebnis soll in die Speicherzelle 003 geschrieben werden.

- Zuerst muss der *Akkumulator* durch Klick auf RESET  und der *RAM* mit  gelöscht werden.
- Führe die dazu erforderlichen Schritte mit dem Programm *Johnny* am PC aus.
- Ergänze die angezeigten Werte handschriftlich. Die Schritte werden beispielhaft in dem Fenster für das Steuerwerk (*Control Unit*) der folgenden Grafik erläutert.



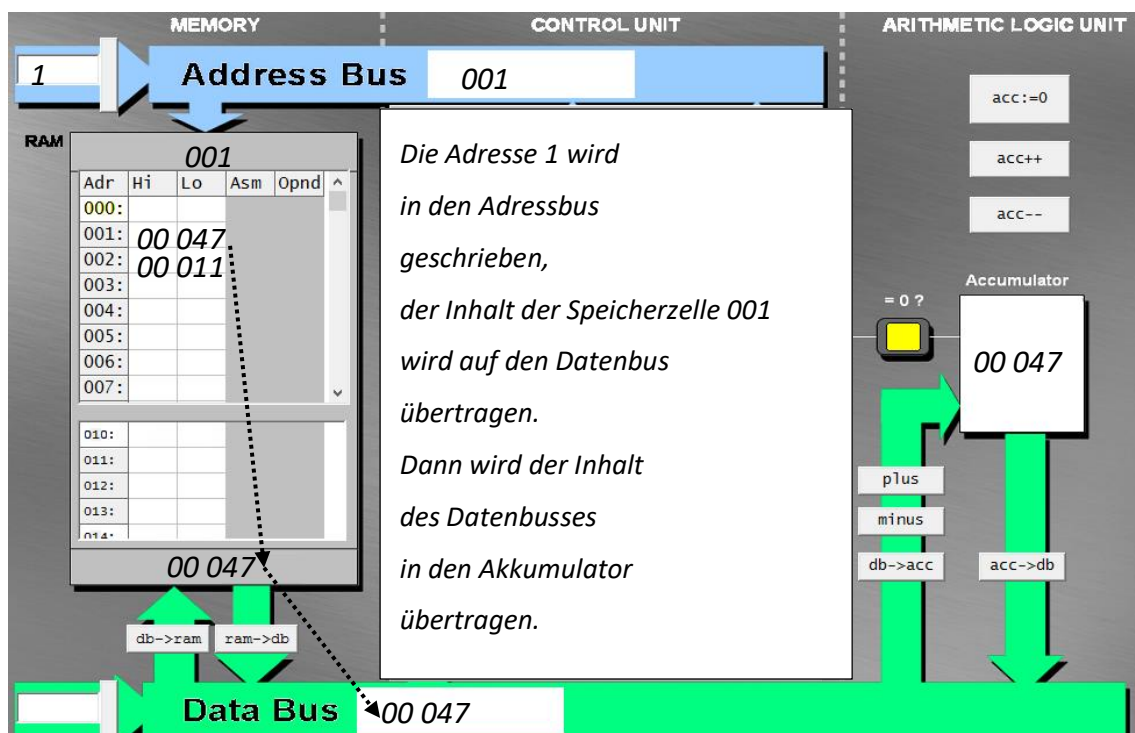
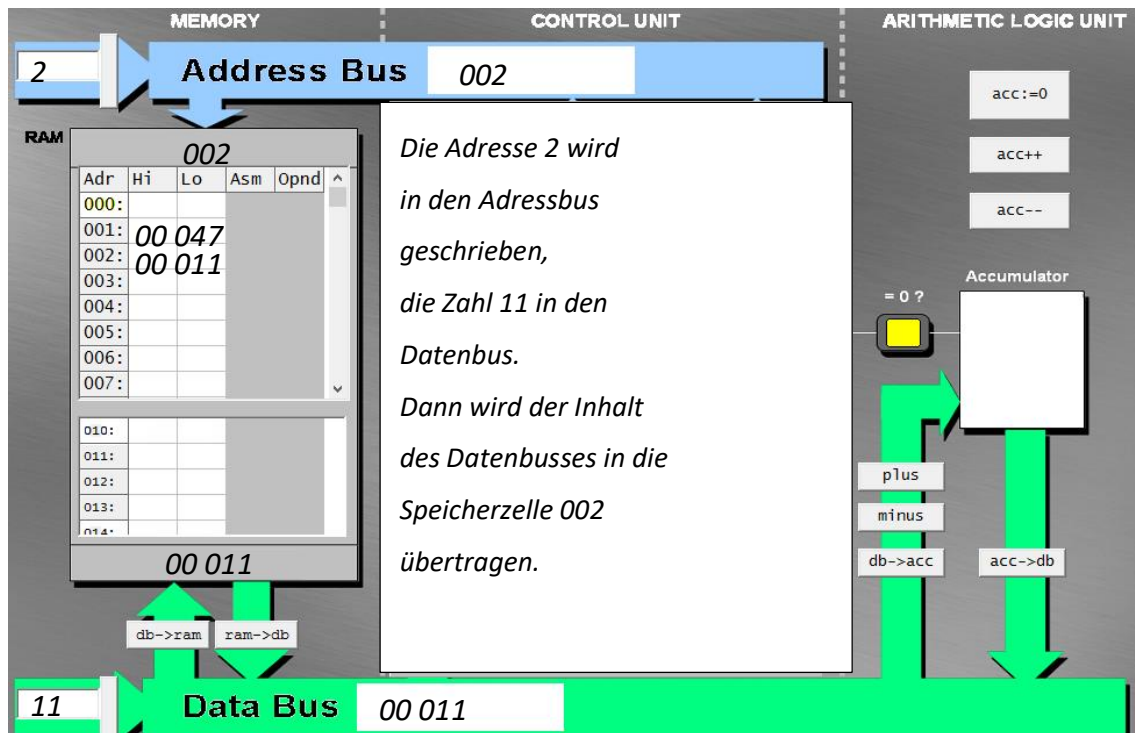


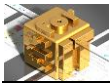
1.8 Grundlagen elektronischer Datenverarbeitung

Arbeitsblatt 05 Funktionsweise eines von-Neumann-Rechners I

Lösungen

10. Ergänze die angezeigten Werte für die weiteren Schritte und erläutere die Schritte in dem Fenster für das Steuerwerk (Control Unit):





11. Stelle die Dezimalzahlen 47 und 11 im Dualsystem dar.

$$47_{10} = 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 101111_2$$

$$11_{10} = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 1011_2$$

12. Rechne $47 + 11$ schriftlich im Dezimalsystem und im Dualsystem.

Dezimalsystem:

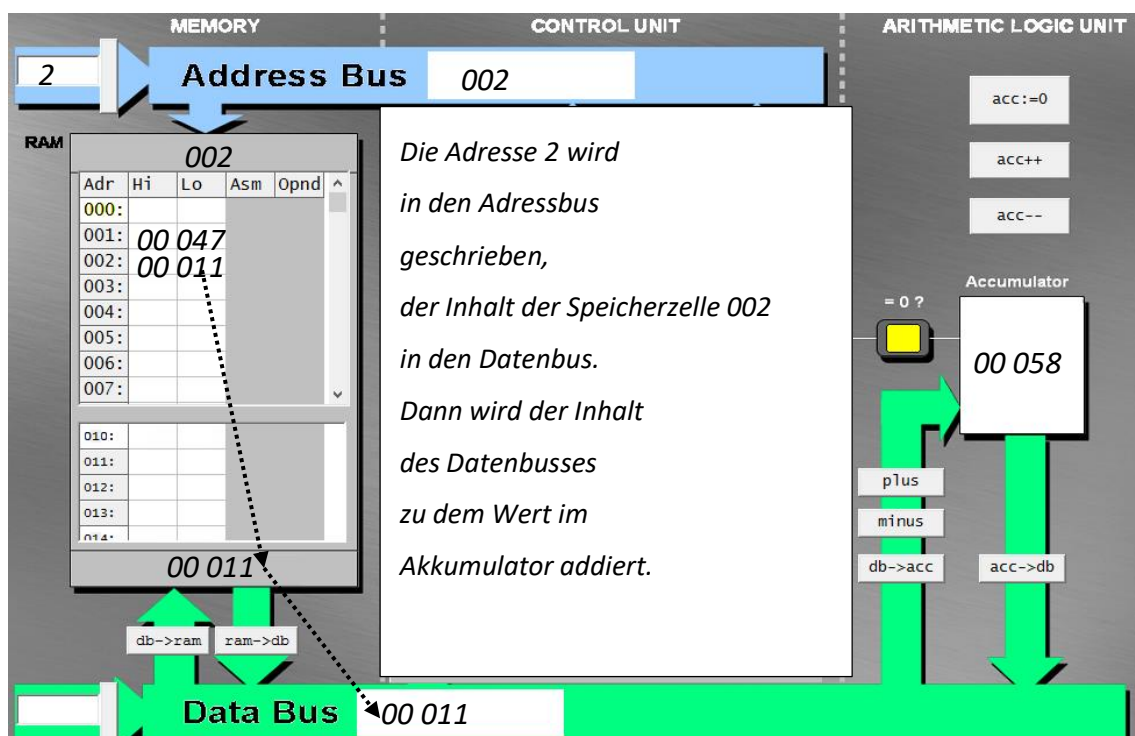
$$\begin{array}{r} 47 \\ + 11 \\ \hline 58 \end{array}$$

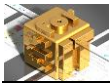
Dualsystem:

$$\begin{array}{r} 101111 \\ + 1011 \\ \hline 111010 \end{array}$$

Ergebnis: $47_{10} + 11_{10} = 58_{10}$ und $101111_2 + 1011_2 = 111010_2$

13. Ergänze die angezeigten Werte für die weiteren Schritte und erläutere die Schritte in dem Fenster für das Steuerwerk (Control Unit):

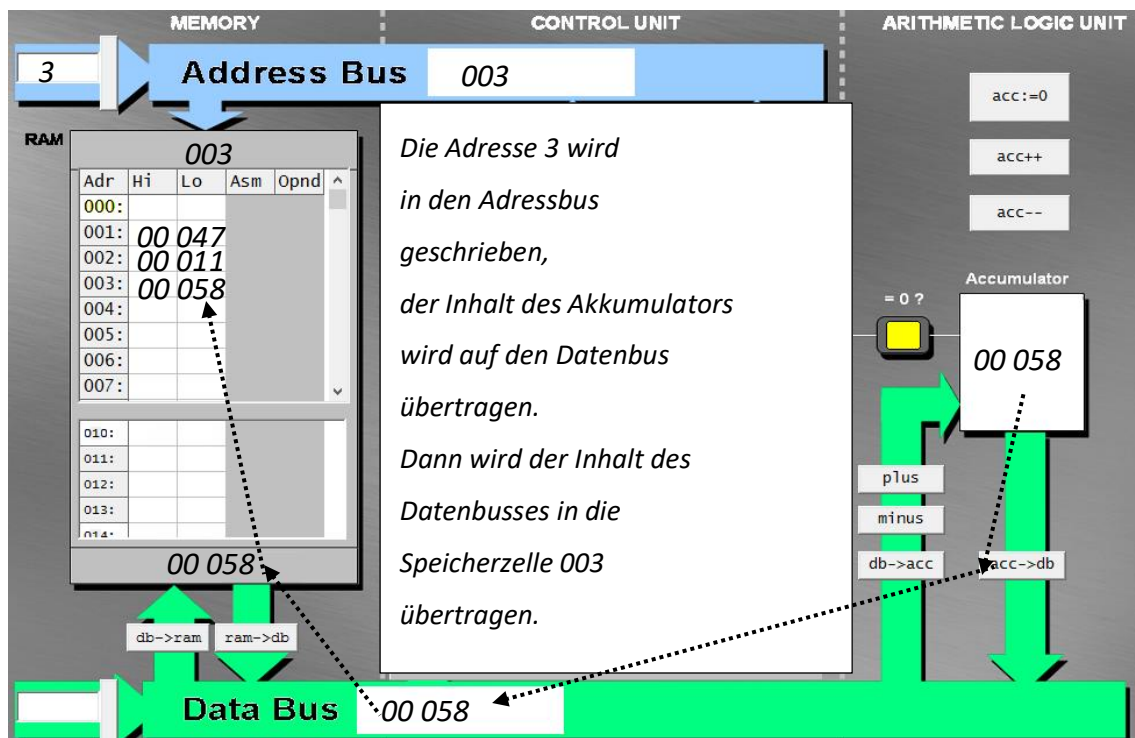




1.8 Grundlagen elektronischer Datenverarbeitung

Arbeitsblatt 05 Funktionsweise eines von-Neumann-Rechners I

Lösungen



Das Ergebnis sieht dann in der Simulation so aus: (18\18-materialien\johnny-dateien\ 03-47plus11.ram)

