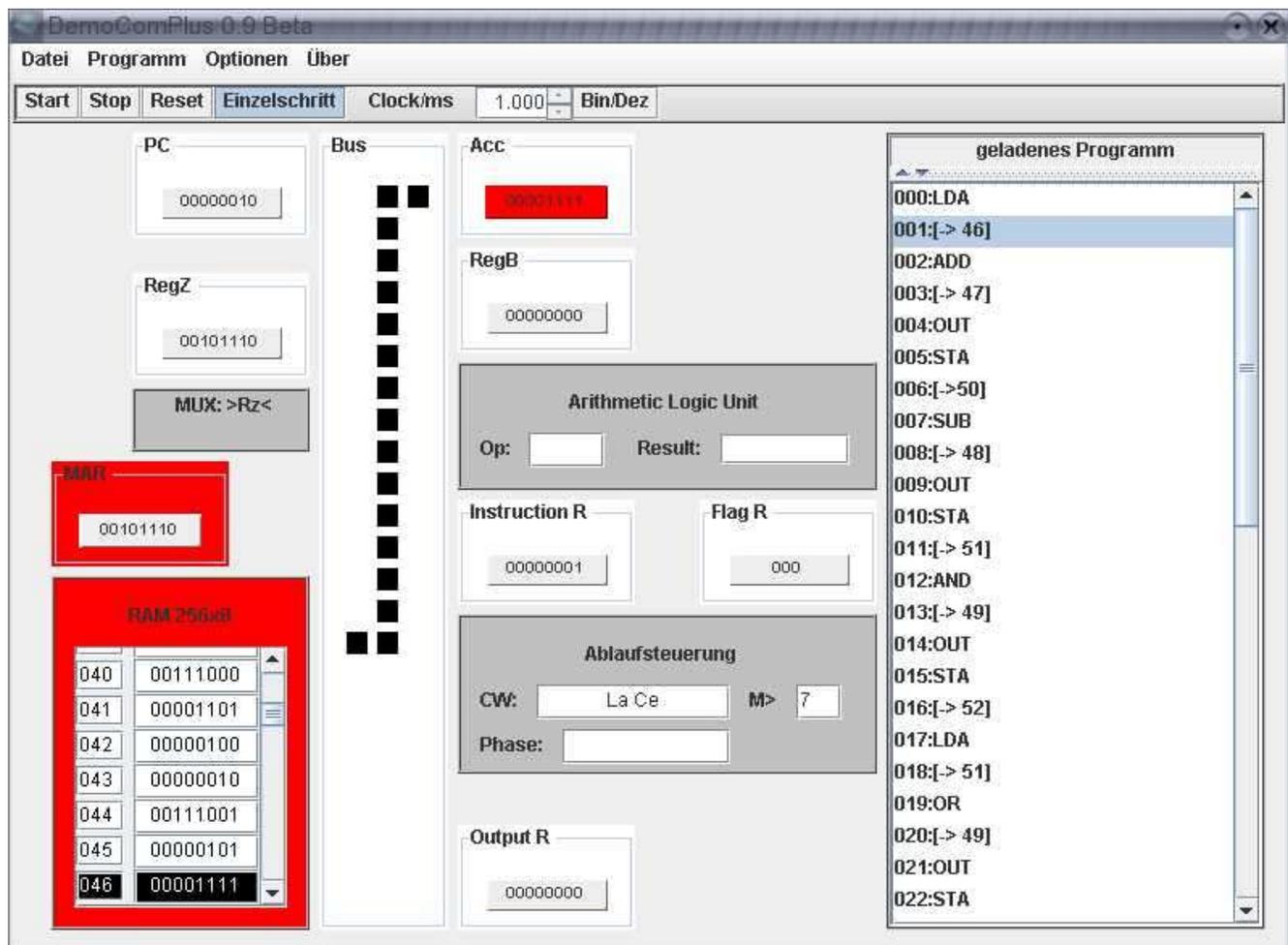


JDemoComPlus

Benutzerhandbuch

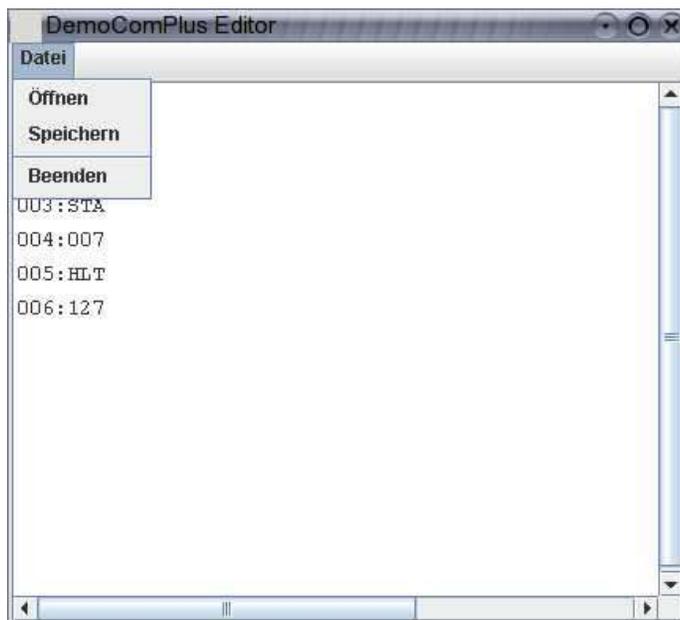
1. Einleitung

JDemoComPlus ist ein Simulator für den DemoComPlus-Demonstrationscomputer. Er stellt die Vorgänge im Inneren des Rechners grafisch dar und ermöglicht das experimentieren mit der DemoComPlus-Architektur. Er richtet sich vor allem an Studenten der Rechnerarchitektur-Vorlesung, um die Inhalte der Vorlesung praktisch nachvollziehen zu können. Ein integrierter Quellcode-Editor und Assembler ermöglichen ein einfaches Übersetzen und Laden eigener Programme.



2. Das BenutzerInterface

Im Hauptbildschirm der Anwendung werden im linken Teil des Bildschirms die wesentlichen aus der Vorlesung bekannten Komponenten des DemoComPlus dargestellt, wobei die Anordnung leicht von der in der Vorlesung eingeführten abweicht. Jede Komponente enthält eine Tooltip-Funktion, die im Falle einer Aktivität der Komponente die entsprechende Funktion Aktion beschreibt, ansonsten die Bezeichnung der Komponente anzeigt. Im rechten Teil wird der vom aktuell geladenen Programm belegte Speicher als Liste dargestellt, hierbei wird zwischen Befehlen, Addressreferenzen und als Variablen genutzten Speicherzellen unterschieden. Die aktuell bearbeitete Speicherzelle wird blau unterlegt, um dem Anwender den Programmfluss anschaulich darzustellen. Die Kontrolle des JDemoComPlus erfolgt über die Toolbar, das Übersetzen und Laden von Programmen erfolgt über die Menüleiste, hierüber kann ebenfalls der integrierte Quellcode-Editor gestartet werden.



Der integrierte Quellcode-Editor des JDemoComPlus

2.1.Funktionen der Toolbar

2.1.1.Kontrolle des Rechners

Die Toolbar enthält vier Buttons, um den Rechner zu kontrollieren. Drückt man den „Run“-Button, beginnt der Rechner zu arbeiten, durch drücken von „Stop“ kann er wieder angehalten werden. Der aktuelle innere Zustand bleibt hierbei erhalten, und durch erneutes drücken von „Run“ kann wieder an der selben Stelle fortgefahren werden. Soll der Rechner in den Anfangszustand zurückversetzt werden, muss der „Reset“-Knopf gedrückt werden. Alle Register werden geleert und der Programmzähler wird auf die Anfangsadresse 0x00 gesetzt. Der Inhalt des RAM-Speichers bleibt jedoch erhalten, ebenso ein eventuell geladenes Programm.

Durch wiederholtes drücken des „Single Step“-Buttons kann ein Programm in Einzelschritten durchlaufen werden: wird der Button heruntergedrückt wird ein einzelner Mikroschritt ausgeführt und die Anzeige eingefroren, bis der Knopf erneut betätigt wird. Danach befindet sich der Rechner im Stop-Zustand und er kann durch „Run“ normal weiterlaufen, oder es können weitere Einzelschritte durchgeführt werden. Über den Regler Clock kann die Anzeigegeschwindigkeit geregelt werden, in der die Datenbewegungen angezeigt werden.

2.1.2.Anpassen der Darstellung

Nach dem Programmstart werden die Inhalte der Register standardmäßig in Binärdarstellung angezeigt. Um die Anzeige auf Dezimaldarstellung umzuschalten kann der entsprechende Button der ToolBar verwendet werden. Wird die Darstellung verändert, so müssen auch Dateneingaben entsprechend erfolgen, s.u.



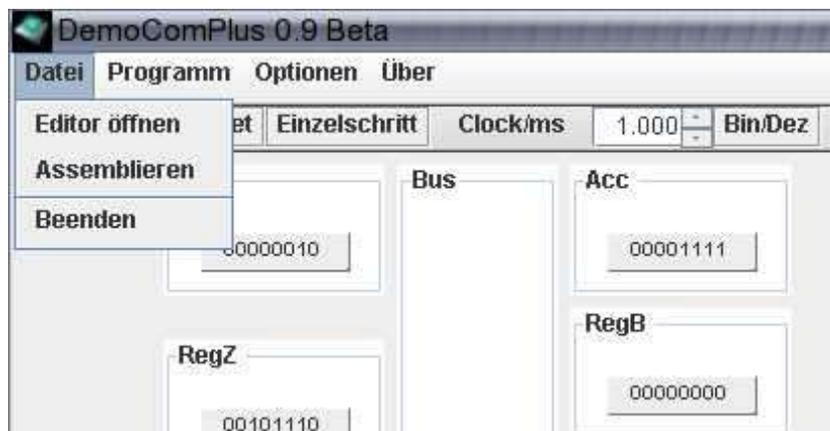
2.2.Funktionen der Menüleiste

2.2.1.Starten des Quellcode-Editors

Der integrierte Quellcode-Editor kann über den Menüpunkt „File“-->“Editor“ aufgerufen werden. Der Editor öffnet ein eigenes Fenster, das einen Textbereich und eine Menüleiste enthält, über die Dateien geladen und gespeichert werden können.

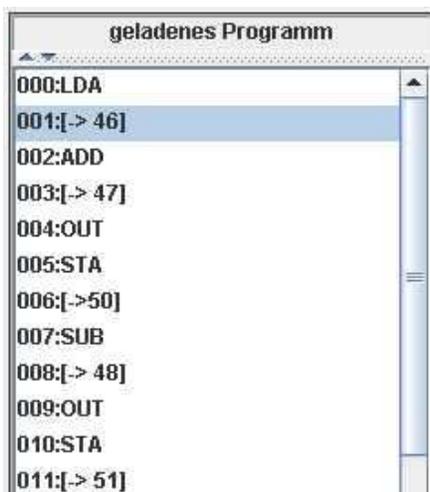
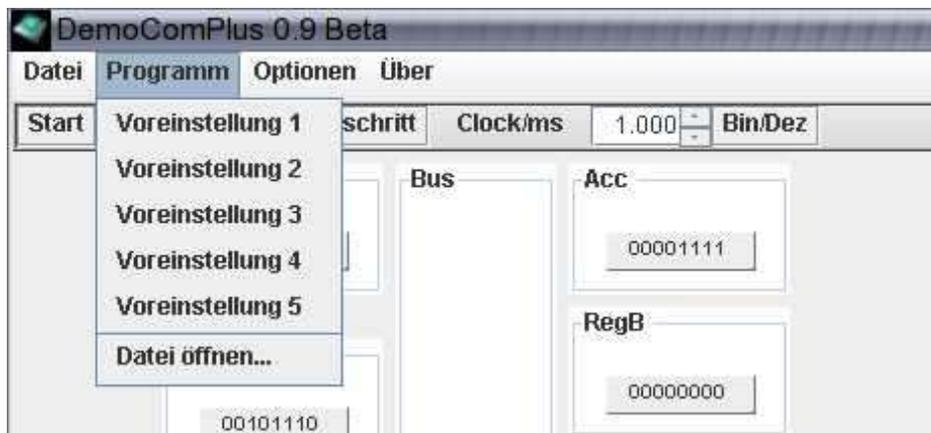
2.2.2.Übersetzen eines Programms

Um das Übersetzen eines Programmes auszulösen wählt man in der Menüleiste „File“-->“Assemble“. Es erscheint eine DialogBox, in der die gewünschte Quelldatei ausgewählt werden kann und das Übersetzen erfolgt nach Bestätigen des Dialoges automatisch. Das fertig übersetzte Programm wird als .dca-Datei in dem Verzeichnis der Quelldatei abgelegt und kann nun geladen werden.



2.2.3.Laden eines Programms

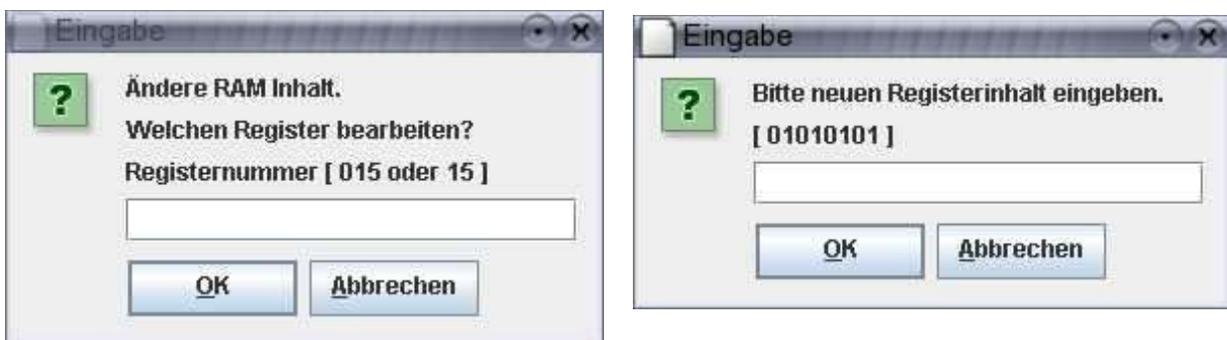
JDemoComPlus enthält 5 eingebaute Beispielprogramme, diese lassen sich über die Menüleistenpunkte „Programm“-->“Preset 1“ bis „Preset 5“ in den Speicher des Rechners laden. Weitere eigene Programme lassen sich über „Programm“-->“Load File“ laden, nach anwählen des Eintrags kann ein zuvor übersetztes Programm (.dca-Dateien) ausgewählt werden. Nachdem ein Programm geladen wurde wird automatisch ein Reset durchgeführt. Das geladene Programm wird auf der rechten Bildschirmseite in einer Liste dargestellt.



Das aktuell geladene Programm wird als Liste dargestellt

2.2.4. Verändern des RAM-Inhaltes

Es besteht ebenfalls die Möglichkeit einzelne Speicherzellen des DemoComPlus-RAM-Speichers zu ändern, was ebenfalls nur im Stop-Zustand möglich ist. Dazu muss in der Menüleiste „Optionen“-->“RAM ändern“ gewählt werden. Es erscheint eine DialogBox, in der zunächst die zu ändernde Speicherzelle gewählt werden muss. Nachdem diese bestätigt wurde erscheint ein weiterer Dialog, in dem der gewünschte neue Wert eingetragen werden kann. Das Eingabeformat ist abhängig von der momentan gewählten Darstellung und im Dialog wird ein entsprechendes Beispiel angezeigt.



Das Eingeben des neuen Wertes kann jederzeit abgebrochen werden. Kann der vom Benutzer eingegebene Wert nicht bearbeitet werden, so wird eine Fehlermeldung ausgegeben und der alte Wert bleibt erhalten.

2.3.Funktionen des Hauptbildschirms

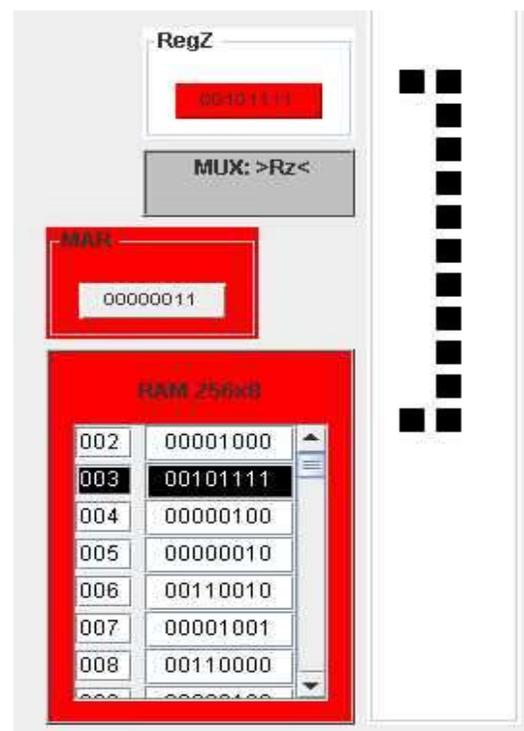
2.3.1.Verändern der Registerinhalte

Befindet sich der Rechner im Stop-Zustand, lassen sich die Inhalte der Register bearbeiten. Mit einem Doppelklick auf den Registerinhalt lässt sich der enthaltene Wert ändern, durch bestätigen mit der Eingabetaste wird der neue Wert übernommen. Wenn binäre Darstellung der Register gewählt wurde muss ein binärer Wert eingegeben werden, bei Dezimaldarstellung muss er dezimal sein.



2.3.2.Anzeige von Zustandsinformation

Aktive Komponenten zeigen meist durch farbliche Veränderung ihren Zustand an. Im nebenstehenden Beispiel sieht man eine Wertübernahme durch RegZ (rotes Inhaltfeld), und Wertübergaben durch MAR und RAM (roter Rahmen). Der Weg der Daten über den Bus wird durch schwarze Kästchen angezeigt. Ebenfalls ist erkennbar, dass der RAM-Speicher die aktuell aktive Zelle in der Liste markiert.



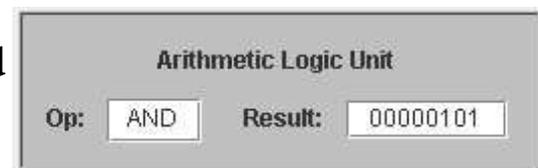
Die Ablaufsteuerung des JdemoComPlus zeigt jeweils das aktuell ausgegebene Kontrollwort, den aktuellen Stand des Mikrozählers sowie die aktuelle Bearbeitungsphase an.



Von der Ablaufsteuerung werden folgende Steuersignale generiert:

- Cp - Program Counter wird um 1 erhöht
- Ep - Enable PC: PC legt aktuellen Zählerstand auf den Bus
- Lp - Load PC: PC wird vom Bus geladen
- Lz - Load Z Register: Z-Register liest vom Bus
- Lm - Load Mar: MAR liest vom Bus
- Ce - Chip Enable: RAM schreibt auf Bus
- We - Write Enable: RAM liest vom Bus
- Li - Load Instruction Reg.: Instruction Register liest vom Bus
- Lb - Load B Register: Register B liest vom Bus
- La - Load Acc: Accumulator liest vom Bus
- Ea - Enable Acc: Accumulator schreibt auf den Bus
- Eu - Enable Unit: ALU schreibt Ergebnis auf Bus zurück
- Lo - Load Output: Output-Register übernimmt Daten vom Bus
- Sa - MUX Schalter
- Lf - Load Flag Register: Flag-Register übernimmt ALU-Flags

Schreibt die ALU das Ergebnis einer Rechenoperation auf den Bus zurück, so wird die aktuell ausgeführte Operation sowie das erhaltene Ergebnis angezeigt.



Beim Interpretieren von Registerinhalten muss beachtet werden, dass der DemoComPlus in 2-complement-Darstellung arbeitet, er erfasst mit 8Bit einen Wertebereich von -127 bis 128. Um die Interpretation zu erleichtern kann auf Dezimaldarstellung umgeschaltet werden, die Registerinhalte müssen somit nicht „von Hand“ interpretiert werden.

3. Die Assemblersprache des integrierten Assemblers

Der JdemoComPlus verfügt über einen integrierten Assembler, um einfacher eigene Programme einsetzen zu können. Dem Benutzer steht eine primitive Programmiersprache zur Verfügung, die im folgenden erklärt werden soll.

3.1. Bedeutung der einzelnen Befehle:

BEFEHL	SYNTAX	OPCODE	BEDEUTUNG
LDA	LDA <ADDRES>	00000001	LADE DATA VON RAM[ADDRES] IN DEN ACC
STA	STA <ADDRES>	00000010	SCHREIBE DATA VON ACC IN RAM[ADDRES]
ADD	ADD <ADDRES>	00001000	ADD DATA VON RAM[ADDRES] UND ACC
SUB	SUB <ADDRES>	00001000	SUB RAM[ADDRES] VON ACC
OR	OR <ADDRES>	00001011	OR* ZW. RAM[ADDRES] UND ACC
AND	AND <ADDRES>	00001010	AND* ZW. RAM[ADDRES] UND ACC
XOR	XOR <ADDRES>	00001100	XOR* ZW. RAM[ADDRES] UND ACC
INV	INV	00001101	INV DER BITS VON ACC
DEC	DEC	00001110	ACC UM 1 VERRINGERN
INC	INC	00001111	ACC UM 1 ERHOEHEN
JNZ	JNZ <ADDRES>	00010000	JUMP TO ADDRES IF ZRO NICHT GESETZT
JZ	JZ <ADDRES>	00010001	JUMP TO ADDRES IF ZRO GESETZT
JNG	JNG <ADDRES>	00010010	JUMP TO ADDRES IF NEG GESETZT
JCY	JCY <ADDRES>	00010011	JUMP TO ADDRES IF CRY GESETZT
JMP	JMP <ADDRES>	00010100	JUMP TO ADDRES
OUT	OUT	00000100	DATA VON ACC IN OUTPUT REGISTER
HLT	HLT	00000101	ANHALTEN DES RECHNERS

3.2.Format einer Quelldatei

Quelldateien für den Assembler sind sehr einfach aufgebaut. Als Beispiel soll folgendes Quellprogramm dienen:

```
000:LDA      #Lade Accumulator mit
001:006     #Zieladresse 6
002:INC     #inkrementiere Accumulator
003:STA     #speichere Accumulator nach
004:006     #Zieladresse 6
005:HLT     #halte den Rechner an
006:-127    #adresse 6 <wertzelle>
```

Die Funktion des Programms ist einfach: es wird der Inhalt einer Speicherzelle in den Accumulator geladen, dieser Wert anschliessend um 1 erhöht und danach an die ursprüngliche Stelle zurückgeschrieben.

Jede Zeile beginnt mit einer Speicherzellennummer von 0-255, sie gibt an in welche Speicherzelle die entsprechende Zeile geladen werden soll. Nach einem Doppelpunkt ist nun entweder ein Befehl oder ein Wert erlaubt, Werte werden dezimal mit optional führenden Nullen eingegeben, erlaubter Bereich ist -127 bis 128. Steht ein Wert hinter einem Befehl dessen Syntax eine Adresse erfordert, so übersetzt der Assembler die folgende Zeile als Adressangabe, ansonsten wird der Wert als Speicherzelle für Nutzdaten betrachtet. Diese Unterscheidung ist zwar für den DemoComPlus unwesentlich da er die entsprechenden Unterschiede nicht kennt, erleichtert jedoch dem Anwender das Unterscheiden zwischen als Adresszeiger genutzten Speicherzellen und als Wertspeicher genutzter. Wird ein Addresswert erwartet ist der Gültigkeitsbereich 0-255, alle Angaben in ungültigen Bereichen führen zu einem Parserfehler.

Quelldateien dürfen auch Kommentare enthalten, sie werden durch das #-Zeichen eingeleitet und der Rest einer Zeile nach dem Zeichen wird als Kommentar aufgefasst.