Neumi 1.0

Andreas Hofmeister & Patrick Stiegeler

18. August 1998

Zusammenfassung

Bedienungsanleitung zur Animation eines von-Neumann-Rechners

Inhaltsverzeichnis

1 Übersicht

1.1 Was ist Neumi

Wie funktioniert eigentlich so ein Prozessor? In Büchern findet man oft irgendwelche Schaltbilder und Beschreibungen davon, wie so etwas funktionieren soll, so richtig vorstellen kann man sich das aber dann auch nicht immer.

Neumi ist ein Java-Programm (auch als Applet lauffähig), das das Papier lebendig macht. Herr Prof. Ottmann bot uns die Gelegenheit im Rahmen unseres Softwarepraktikums eine Visualisierung der Abläufe in einem einfachen Prozessor zu erstellen. Als Vorlage diente uns die im Buch "Hardware Design" vorgestellte CPU Namens "RESA" (Rechenanlage Saarbrücken), die auch Gegenstand der Vorlesung Technische Informatik II bei uns in Freiburg war.

1.2 Wieso der Name "Neumi"?

Unsere Aufgabenstellung war es, eine Simulation zu erstellen, die die Abläufe in einem "von-Neumann-Rechner" veranschaulicht. Die von uns simulierte RESA-CPU wurde in der Vorlesung "Technische Informatik II" zu "ReTi-II" (Rechner zur Technischen Informatik II) umbenannt. Wir verwenden zwar diese CPU in der aktuellen Version unseres Programmes, doch da das Gewicht mehr auf der Simulation eines "von-Neumann-Rechners" liegt, als auf der Darstellung dieser konkreten CPU, beschlossen wir, nicht "ReTi-II" als Namen zu übernehmen. Wir entschieden uns für "Neumi" (kurz für: "kleiner von-Neumann-Rechner"; außerdem ist uns nichts besseres eingefallen).

2 Installation

2.1 Systemvoraussetzungen

Zum Betrieb von Neumi ist ein System notwendig, auf dem eine Java 1.1 kompatible Umgebung installiert ist. Diese kann entweder von einem Web-Browser bereitgestellt werden, in diesem Fall läuft Neumi als Applet, oder von einem eigenständigen Java-Interpreter.

Außer den Java-Standardklassen, benötigt Neumi auch das AWT (Abstract Windowing Toolkit). Aus diesem Grund läuft Neumi mit alternativen Interpretern, wie etwa kaffee, nicht.

Neumi besitzt zur Anzeige der HTML-Seiten der Dokumentation einen eingebauten HTML-Viewer, dieser ist jedoch noch alles andere als stabil, weshalb es auch die Möglichkeit gibt Netscape oder kdehelp zu verwenden. Nur dann, wenn man den eingebauten Viewer ausprobieren möchte, ist zusätzlich das SWING-Packet von Sun erforderlich.

2.2 Installation als Applikation

Alle benötigten Dateien zu Neumi befinden sich in der Datei "Neumi.jar". Um das Programm ausführen zu können, können Sie entweder

- die Datei Neumi.jar in die Variable CLASSPATH aufnehmen

oder

- die Datei Neumi.jar in einem Verzeichniss auspacken und das Java-Programm NeumiApplet in diesem Verzeichnis starten.

In beiden Fällen wird das Programm mit "java NeumiApplet" gestartet.

2.3 Installation als Applet

Möchten Sie das Programm als Applet starten, brauchen Sie keine Dateien zu installieren. Falls Sie eigene Assembler-Programme schreiben und diese dann abspeichern wollen, so müssen wir Sie hier ein bischen enttäuschen. Um auf Ihre Festplatte zugreifen zu können müßten Sie dem Applet die Erlaubnis hierzu erteilen (dies kann man dann im Browser einstellen). Hierzu wäre es aber notwendig, daß wir unser Applet zertifizieren lassen, was wir aber noch nicht getan haben (kostet Geld!). Eine sinnvolle Alternative ist uns noch nicht bekannt.

3 Bedienung der Simulation

3.1 Die Button-Leiste

Am rechten Rand des Fenster befinden sich eine Reihe von Knöpfen und ein Schieber, mit denen die Animation gesteuert werden kann. Im einzelnen sind dies die folgenden Elemente.

3.1.1 der Start-Knopf

Mit diesem Knopf wird die Animation gestartet. War sie zuvor im Pause-Modus, so fängt sie nun wieder an weiterzulaufen.

3.1.2 der Pause-Knopf

Mit dem Pause-Knopf wird die Animation vorübergehend angehalten. Nach drücken des Start-Knopfes läuft die Animation weiter, der Stop-Knopf hält die CPU endgültig an und versetzt sie in den Anfangs-Zustand.

3.1.3 der Stop-Knopf

Der Stop-Knopf bringt die Animation in den Grundzustand zurück und löscht die Trace-Liste.

3.1.4 Der Schieberegeler

Mit dem Schieber kann die Geschwindigkeit der Animation eingestellt werden, am oberen Anschlag bewegen sich die Daten in der CPU sehr langsam, steht der Schieber am unteren Ende der Skala, so bewegen sich die Daten deutlich schneller. Die Geschwindigkeit kann jederzeit eingestellt werden, auch wenn die Animation gerade läuft.

3.1.5 Die Trace-Liste

In der Trace-Liste werden die zuletzt ausgeführten Befehle aufgelistet. Diese Liste enthält 20 sichtbare Zeilen. Stehen mehr Befehle in der Liste, so erscheint rechts von der Liste ein Schieber, mit dem man den angezeigten Ausschnitt der Liste verändern kann.

Durch Klicken auf einen Eintrag der Liste, kann die Animation wieder in den Zustand am Anfang des entsprechenden Befehls versetzt werden. Alle folgenden Einträge der Liste werden gelöscht und die CPU beginnt mit der Abarbeitung des ausgewählten Befehls.

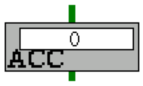
3.2 Besonderheiten bei Web.Browsern

Haben Sie Neumi als Applet gestartet, so ist es leider (noch) nicht möglich eine Datei von Ihrer Festplatte in den RAM-Editor einzulesen, bzw. von dort auf Ihre Festplatte zu sichern. Desweiteren ist es nicht möglich Daten mittels "Cut and Paste" mit anderen Programmen auszutauschen.

4 Baugruppen

Eine CPU setzt sich aus mehreren Komponenten zusammen. Diese werden nach dem Baukasten-Prinzip zusammengesetzt. Von diesen Bausteinen gibt es mehrere Gruppen (die sog. Baugruppen), von denen im folgenden die wichtigsten vorgestellt werden.

4.1 Register



Register sind Zwischenspeicher in der CPU. So werden Register oft als ein Operand bei Berechnungen benötigt. Der Inhalt eines Register wird im Feld oberhalb des Register-Namens angezeigt.

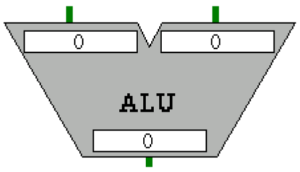
4.1.1 Increment Register

Ein spezielles Register bei der RETII-CPU ist das Register PC (Program Counter). Es enthält die Adresse des gerade ausgeführten Befehls. Zusätzlich zu den Eigenschaften eines normalen Registers, hat dieses Register die Fähigkeit seinen Inhalt zu inkrementieren.

4.1.2 I-Register der RETII CPU

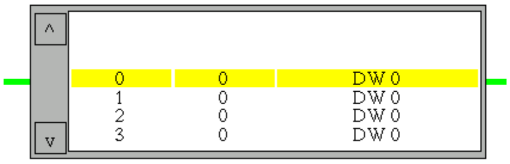
Noch ein Spezialfall der RETII-CPU. Die Befehle dieser CPU enthalten oft einen 24-Bit langen Wert, der je nach Befehl mit oder ohne Vorzeichen verwendet werden soll. Da aber die ALU mit 32 Bit rechnet, muß diese Zahl um die fehlenden acht Bit erweitert werden. Das I-Register besitzt nun die Fähigkeit, eine hierzu benötigte Vorzeichen-Erweiterung (Sign Extension) durchzuführen.

4.2 ALU



Die ALU ist das Herz des Prozessors, hier werden alle Berechnungen durgeführt. Unsere ALU kennt die Operationen Addition, Subtraktion, bitweises UND, ODER und Exklusiv-ODER sowie ALU-NOP (keine Operation). Welche Operation gerade durchgeführt wird, wird oberhalb der Beschriftung “ALU” angezeigt. Wird hier nichts angezeigt, führt die ALU die NOP-Operation durch. Hier wird einfach der linke Operand zum Ausgang kopiert.

4.3 RAM

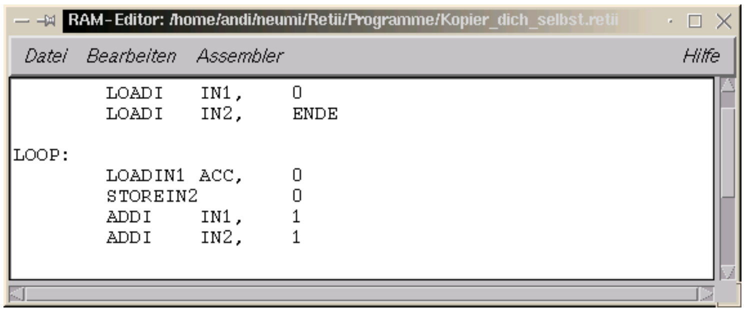


Das RAM ist der Programm- und Daten-Speicher. Es kann 32-Bit-Worte aufnehmen und auf Anfrage wieder ausgeben. Es gibt, wie bei jedem von-Neumann-Rechner, keinen Unterschied zwischen Daten- und Befehls-Worten. Jeder Wert im RAM wird darum sowohl als Zahl wie auch als Mnemonic der Interpretation als Befehl angezeigt.

Jede Zeile im Anzeigefeld des RAM ist in drei Spalten unterteilt. In der ersten Spalte steht die Speicher-Adresse, in der zweiten der dort gespeicherte Wert (notiert als Hexadezimal-Zahl) und in der dritten die entsprechende Kodierung dieses Wertes als Mnemonic. Das Zuordnen von Wert zu Menmonic ist dabei vom Befehlssatz der dargestellten CPU abhängig.

Am linken Rand der Anzeige stehen zwei Knöpfe zur Verfügung, mit deren Hilfe man den sichtbaren Speicher-Ausschnitt verändern kann. Dies erleichtert oftmals das Testen eines eben selbstgeschriebenen Programmes.

Klickt man mit der Maus auf das RAM, so wird der RAM-Editor angezeigt.



Mit dessen Hilfe können eigene Programme für die CPU geschrieben und nach erfolgreicher Assemblierung in das RAM geladen werden. Auch der umgekehrte Weg, den RAM-Inhalt zu dissassemblieren, ist möglich. Die Programme können dann auf Festplatte gespeichert und auch wieder von dort gelesen werden. Eine Übersicht über das Schreiben von Programmen gibt das Hilfe-Menü.

A Properties

Das Verhalten und das Erscheinungsbild der Animation lassen sich über die Datei “Neumi.properties” steuern. Jeder Eintrag in dieser Datei besteht aus Zeilen der Form

```
<Parameter> = <Wert>
```

Die folgendeTabelle enthält die möglichen Einträge in der Datei, deren Auswirkungen und den jeweiligen Default-Werten. Die einzelnen Einträge besitzen dabei folgende mögliche Belegungen:

- alle **Font**-Angaben: “Helvetica”, “Courier”, “TimesRoman”
- alle **Font.Style**-Angaben: “PLAIN”, “BOLD”, “ITALIC”. Kombinationen werden durch Leerzeichen getrennt angegeben, z.B. “BOLD ITALIC”.
- alle **Font.Size**-Angaben: positive Integer-Zahl (obere Grenze wissen wir grad nit)
- **Animator.FontAdjust**: Integer-Zahl
- **HTMLViewer**: “netscape”, “kdehelp”, “swing”.
- **DocumentRoot**: URL, die zu den Hilfe-Dateien führt (kann auch mit “file://” beginnen).

Eintrag	Vorgabe	Wirkung
Schriften allgemein		
Font	Helvetica	Default-Wert für Schriften
Font.Size	14	Default-Wert für die Schrift-Größe
Font.Style	PLAIN	Default-Wert für den Schrift-Stil
Editor-Font		
Editor.Font	Courier	Schrift, die der RAM-Editor benutzt
Editor.Font.Style	PLAIN	Zugehöriger Schrift-Stil
Editor.Font.Size	<Font.Size>	Größe der Schrift
Menü-Font		
Menu.Font	Helvetica	Schrift für die Menüs des RAM-Editors
Menu.Font.Style	ITALIC	Zugehöriger Schrift-Stil
Menu.Font.Size	<Font.Size>	Größe der Schrift
Button-Font		
Button.Font	Helvetica	Schrift für Buttons in Dialogen
Button.Font.Style	BOLD	Zugehöriger Schrift-Stil
Button.Font.Size	<Font.Size>	Größe der Schrift
Font für Dialoge		
Dialog.Font	Helvetica	Schrift für Dialoge
Dialog.Font.Style	<Font.Style>	Zugehöriger Schrift-Stil
Dialog.Font.Size	<Font.Size>	Größe der Schrift
Dialog.Heading.Font	<Font>	Schrift für Dialog-Überschriften
Dialog.Heading.Font.Style	BOLD ITALIC	Zugehöriger Schrift-Stil
Dialog.Heading.Font.Size	14	Größe der Schrift
Animation		
Animator.Font	TimesRoman	Schrift für die Objekte der Animation
Animator.FontAdjust	-2	Korrektur-Offset für die Berechnung der Schriftgrößen für die Objekte der Animation. Falls die Beschriftungen der Objekte ungewöhnlich groß oder klein sein sollten, kann man mit diesem Wert etwas experimentieren.
Html-Viewer Vorgaben		
HTMLViewer	netscape	Programm zur Anzeige von HTML-Dokumenten (Hilfe-Texte...). Als Applet gestartet wird dieser Wert ignoriert. Es wird der HTML-Viewer benutzt, in dem das Applet läuft.
DocumentRoot	–	Pfad zu den Dokumenten, die mit dem HTMLViewer angezeigt werden sollen. Sollte bei der Installation gesetzt werden. Bei Applets wird der Wert ignoriert.