

Prozessor-Simulator

Diplomanden



Michael Schneider



Tobias Petter

Einleitung: Die Funktionsweise von modernen Prozessoren wird im Informatik-Studium an der OST im Rahmen der Module "Betriebssysteme 1 und 2" unterrichtet. Besonders die Kapitel über Assembler und Prozessoren sind komplex und erfordern ein hohes Mass an abstraktem Denkvermögen. Um ein besseres Verständnis bei den Studierenden zu fördern, gab der Studiengang 2020 die Entwicklung eines grafischen Prozessor-Simulators in Auftrag. Mithilfe dessen soll das Nachverfolgen der Arbeitsschritte eines Prozessors leichter möglich sein. Im Zuge unserer Studienarbeit im Herbstsemester 2021 entwickelten wir diesen Simulator weiter und ergänzten ihn um weitere Funktionalität. Das Ziel der vorliegenden Arbeit war, diese Weiterentwicklung abermals fortzuführen. Wir konzentrierten uns dabei auf Verbesserungen der von uns in der Studienarbeit eingefügten Code-Anzeige, um es Nutzern zu erlauben, mit ihrem Code zu interagieren. Der Simulator soll so künftigen Generationen von Studenten noch besser beim Verständnis dieses grundlegenden Stoffes helfen.

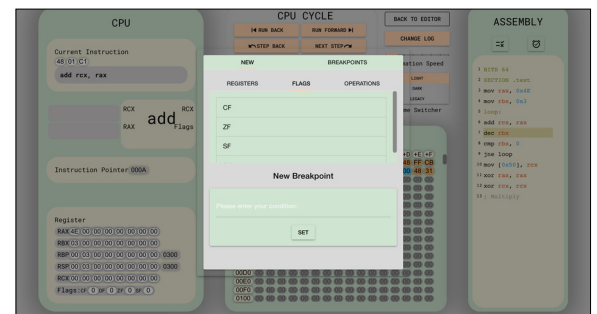
Vorgehen: Der Hauptteil unserer Arbeit bestand darin, das Setzen von bedingten Breakpoints sowie Watchpoints zu ermöglichen. Wir verwendeten dazu einen Expression Parser, um vom Nutzer eingetragene Bedingungen auf ihr Zutreffen zu überprüfen und das Programm gegebenenfalls anzuhalten. Watchpoints funktionieren ähnlich, werden aber nach jedem Schritt automatisch überprüft. Um diese Features einbauen zu können, mussten wir zuerst eine Verbindung zwischen den Instruktions-Bytes im Speicher und den im Frontend angezeigten Textzeilen herstellen. Nachdem wir Zeilen, die nicht auf Instruktionen mappen, aus dem Text im Frontend herausgefiltert hatten, konnten wir unter Verwendung des Disassemblers eine solche Korrelation mit den Daten im Speicher herstellen.

Ergebnis: Es ist uns gelungen, die anfangs stark eingeschränkte Code-Anzeige so auszubauen, dass der Nutzer damit die vollständige Kontrolle über die Ausführung des Programms hat. Es ist möglich, den Simulator schrittweise oder kontinuierlich vorwärts oder rückwärts laufen zu lassen. Bedingte und unbedingte Breakpoints sowie Watchpoints können gesetzt werden und halten die Ausführung des Programms gegebenenfalls an.

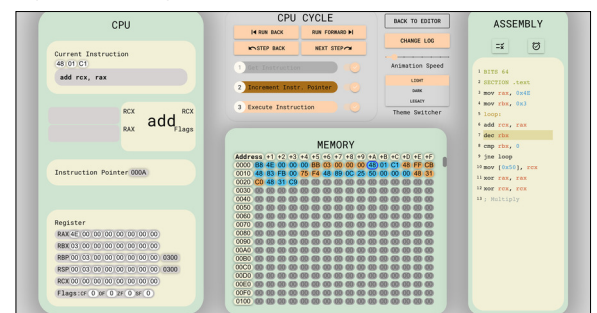
Es ist ausserdem möglich, auf spezifische Codezeilen zu klicken, um den Zustand des Simulators bis zu dieser Zeile vor- oder zurückzuspulen. Weiterhin konnten wir mehrere Verbesserungen beim Informationsgehalt angezeigter Daten einbauen. Wie in einem konventionellen Debugger wird die aktuell ausgeführte Codezeile farblich hervorgehoben. Ausserdem sind die einzelnen kodierten Instruktionen im Speicher jetzt farblich unterschiedlich gekennzeichnet. Der Simulator ist unter einer OpenSource-Lizenz

veröffentlicht und kann unter <http://cpusimulator.pages.gitlab.ost.ch/cpusim-frontend/> verwendet werden.

Das neue Popup zum Setzen von bedingten Breakpoints Eigene Darstellung



Der gesamte Simulator Eigene Darstellung



Referent
Prof. Stefan Richter

Korreferent
Dr. Ettore Ferranti, ABB
Schweiz AG, Baden,
AG

Themengebiet
Software, Application
Design

