DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Konzeptionierung eines Simulators für 8-bit Prozessoren

Studienarbeit

Bachelor of Science

Studiengang Informationstechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Andreas Schmider, Nico Schrodt

Abgabedatum 30. November 2021

Bearbeitungszeitraum	2 Semester
Kurs	TINF19B3
Betreuer der Ausbildungsfirma	Prof. Dr.-Ing. Kai Becher

Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

Konzeptionierung eines Simulators für 8-bit Prozessoren

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, 30. November 2021

Ort, Datum Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

1 Einführung

Test

1.1 Ziel der Arbeit

In dieser Arbeit soll ein Simulationsprogramm geschrieben werden, mit dem mehrere unterschiedliche 8-Bit Prozessoren simuliert werden können. Dazu sollen die Grundlegenden Eigenschaften in kurzen Tutorials erläutert werden. Ebenfalls soll es eine interaktive Einweisung geben wie der Simulator verwendet werden kann.

1.2 Theoretische Grundlagen

1.2.1 Architektur eines Prozessors

1.2.2 Befehlsformate

1.2.3 CISC und RISC

1.2.4 Parallelität nach Flynn

1.2.5 Evolution der Prozessoren

1.2.6 Unterschiede 8-bit, 16-bit, 32-bit und 64-bit Prozessoren

1.3 Auswahl der Werkzeuge

1.3.1 Programmiersprache

1.3.2 GUI-Umgebung

1.3.3 Programmierumgebung

2 Projektplanung

Platzhalter

2.1 Zeitplan

2.2 Auswahl geeigneter Varianten

2.3 Intel 8080

Der Intel 8080 verwendet fünf Arten von Befehlen:

- Data Transfer Group
- Arithmetic Group
- Logical Group
- Branch Group
- Stack, I/O and Machine Control Group

Die Befehle der Data Transfer Group bewegen Daten zwischen Registern und/oder Speicher wie zum Beispiel mit den MOV Befehlen. In der Arithemtic Group werden, wie der Name schon sagt, Befehle mit arithmetische Operationen wie ADD (Addition) verwendet. In der Logischen Gruppen werden logische Operation wie ORA (Oder) verwendet. In der Branch Group liegen die Befehle, welche den Standardmäßigen Programmfluss ändern und das Programm nicht zwangsläufig in der nächsten Zeile fortgesetzt wird (JZ (Jump on Zero)). In der letzten Gruppe liegen die Befehle, die Eingänge und Ausgänge beachten oder den Stack bearbeiten. [Intel 8080 S.46 (4-1)]

Der Intel 8080 ist in der Lage Befehle, die aus einem, zwei oder drei Bytes bestehen, auszuführen. Dabei gibt das erste Byte immer den Opcode oder Operation Code an. In Byte zwei und drei werden nur Daten oder Adressen gespeichert. Dabei werden die zwei Byte großen Adressen so gespeichert, dass das niederwertige Byte vor dem höherwertigem gespeichert wird. Die Adressen können dabei über vier verschiedene Modi verwendet werden.

- Direct
- Register
- Register Indirect

- Immediate

Bei "Direkt"wird der Wert in dem Speicher mit der angegebenen Adresse verwendet. Hier werden das Low-Byte im zweiten und das High-Byte im dritten Byte gespeichert. Bei "Register"wird auf ein oder zwei Register verwiesen und verhält sich wie bei Direkt. Bei "Register Indirect"wird der Wert aus der Adresse aus dem zweiten und dritten Byte des Befehls gelesen. Dieser Wert wird als Adresse verarbeitet und erst der Wert aus dieser Adresse ist der zu verwendete Wert. Bei Ïmmediateßteht im zweiten und/oder dritten Byte ein Wert mit dem gearbeitet wird (Lowbyte im zweiten Byte). [Intel 8080 S.47 (4-2)]

Bei Interrupts und Branch Befehlen gibt es nur den "Directünd "Register indirect"Modus [Intel 8080 S.47 (4-2)].

Der Prozessor besitzt fünf Condition Flags. Das Zero flag, das angibt ob das Ergebnis eines Befehls den Wert 0 hatte. Das Sign flag, welches angibt ob Bit 8, das Most Signifikant-Bit, des letzten Ergebnisses den Wert 1 hat. Das Paritäts flag, welches gesetzt ist, wenn das letzte Ergebnis einen Modulo 2 Wert von 0 hat, also der Wert gerade ist. Das Carry flag, das einen Übertrag bei einer Addition oder einen Abzug bei einer Subtraktion oder Vergleich anzeigt und noch das Auxiliary Carry flag, welches ebenfalls einen Übertrag oder Abzug anzeigt aber zwischen dem vierten (Bit 3) und fünften Bit (Bit 4). [Intel 8080 S.47f (4-2)]

2.3.1 Data Transfer Group

Für den Prozessor sind 13 Befehle aus dieser Gruppe bekannt. Bei keinem dieser Befehle werden die Condition Flags gesetzt oder zurückgesetzt.

- Move Register
- Move from Memory
- Move to Memory
- Move immediate
- Move to Memory Immediate
- Load register pair immediate
- Load Accumulator direct
- Store Accumulator direct

- Load H and L direct
- Store H and L direct
- Load Acumulator indirect
- Store Accumulator indirect
- Exchange H and L with D and E

2.3.2 Arithmetic Group

- Add Register
- Add Memory
- Add immediate
- Add Register with carry
- Add Memory with carry
- Add immediate with carry
- Sustract Register
- Subtract Memory
- Subtract immediate
- Subtract Register with borrow
- Subtract Memory with borrow
- Subtract immediate with borrow
- Increment Register
- Increment Memory
- Decrement Register
- Decrement Memory
- Increment register pair
- Decrement register pair
- Add register pair to H and L
- Decimal Adjust Accumulator

2.3.3 Logical Group

- AND Register
- AND Memory
- AND immediate
- Exclusive OR Register
- Exclusive OR Memory
- Exclusive OR immediate
- OR Register
- OR Memory
- OR immediate
- Compare Register
- Compare Memory
- Compare immediate
- Rotate left
- Rotate right
- Rotate left through Carry
- Rotate right through Carry
- Complement Accumulator
- Complement Carry
- Set Carry

2.4 Beispielprozessor 2

2.5 Beispielprozessor 3 ...

3 Umsetzung

Platzhalter

3.1 Abstraktion der Architektur

3.1.1 ALU

3.1.2 Akkumulator

3.1.3 Instruction Register

3.1.4 DataBus

3.1.5 etc.

3.2 Ablauf der Simulation

3.3 Tutorials

4 Fazit und Ausblick

Platzhalter

Literatur

[1] Google: https://www.google.com