



HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HOF

SEMINARARBEIT

**Aufbau und Funktionsweise eines
Prozessors**

Marco Vogel

unter Aufsicht von
Stefan Müller

7. November 2017

Aufbau und Funktion eines Prozessors

Marco Vogel

7. November 2017

Inhaltsverzeichnis

1	Motivation	4
2	Informationsverarbeitung	4
2.1	Binäre Darstellung von Zahlen	4
3	Logische Schaltglieder	4
3.1	AND-Gatter	4
3.2	OR-Gatter	4
3.3	NOR-Gatter	4
3.4	XOR-Gatter	4
3.5	NOT-Gatter	4
3.6	Flip-Flops	4
4	Prozessorarchitekturen	4
4.1	Von-Neumann Architektur	4
4.2	Harvard Architektur	4
4.3	CISC-Prozessoren	4
4.4	RISC-Prozessoren	4
5	Aufbau und Funktion	4
5.1	Steuerwerk	4
5.2	Register	5
5.2.1	Universalregister	6
5.2.2	Spezialregister	6
5.3	Arithmetisch Logische Einheit	7
5.3.1	ALU-Konfigurationen	7
5.3.2	Arithmetische Operationen	7
5.3.3	Logische Operationen	7
5.4	Memory Management Unit(evtl)	7
5.5	Bussysteme	7

6 Speicher	7
6.1 RAM/ROM	7
6.2 Stack	7
7 Befehlsausführung	7
7.1 Befehlszyklus	7
7.2 Schleifen	7
8 Besondere Ausführungsarten	7
8.1 Interrupts	7
8.2 Exceptions	7
8.3 Subroutinen	7
9 Planung und Entwurf eines Prozessors	7
9.1 Befehlsbreite	7
9.2 Befehlssatz	7
9.3 Speicher	7
9.3.1 RAM/ROM	7
9.3.2 Stack	7
10 Implementierung einer Prozessorsimulation in Logisim	7
10.1 Logisim	7
10.2 Prozessor Komponenten	7
10.3 Ausführung eines Assemblerprogrammes	7

Abbildungsverzeichnis

1	Darstellung des RegisterwerkTODO	5
---	--	---

1 Motivation

2 Informationsverarbeitung

2.1 Binäre Darstellung von Zahlen

3 Logische Schaltglieder

3.1 AND-Gatter

3.2 OR-Gatter

3.3 NOR-Gatter

3.4 XOR-Gatter

3.5 NOT-Gatter

3.6 Flip-Flops

4 Prozessorarchitekturen

4.1 Von-Neumann Architektur

4.2 Harvard Architektur

4.3 CISC-Prozessoren

4.4 RISC-Prozessoren

5 Aufbau und Funktion

5.1 Steuerwerk

Jeder Prozessor besitzt einen gewissen Umfang ihm zur Verfügung stehender Befehle. Diese Befehle werden als Bitmuster oder Mnemonic dokumentiert. Das Steuer-

werk analysiert das Bitmuster welches aus dem Speicher zur Ausführung übergeben wird und vergleicht es mit den bekannten Bitmustern der Opcode-Befehle. Sollte eine Übereinstimmung gefunden werden wird ein Signal, welches dem dekodierten Befehl entspricht, an die angebundenen Hardware der CPU übergeben (ALU bzw. Register). Diese benutzen dieses Signal daraufhin zur weiteren Befehlsausführung.[?]

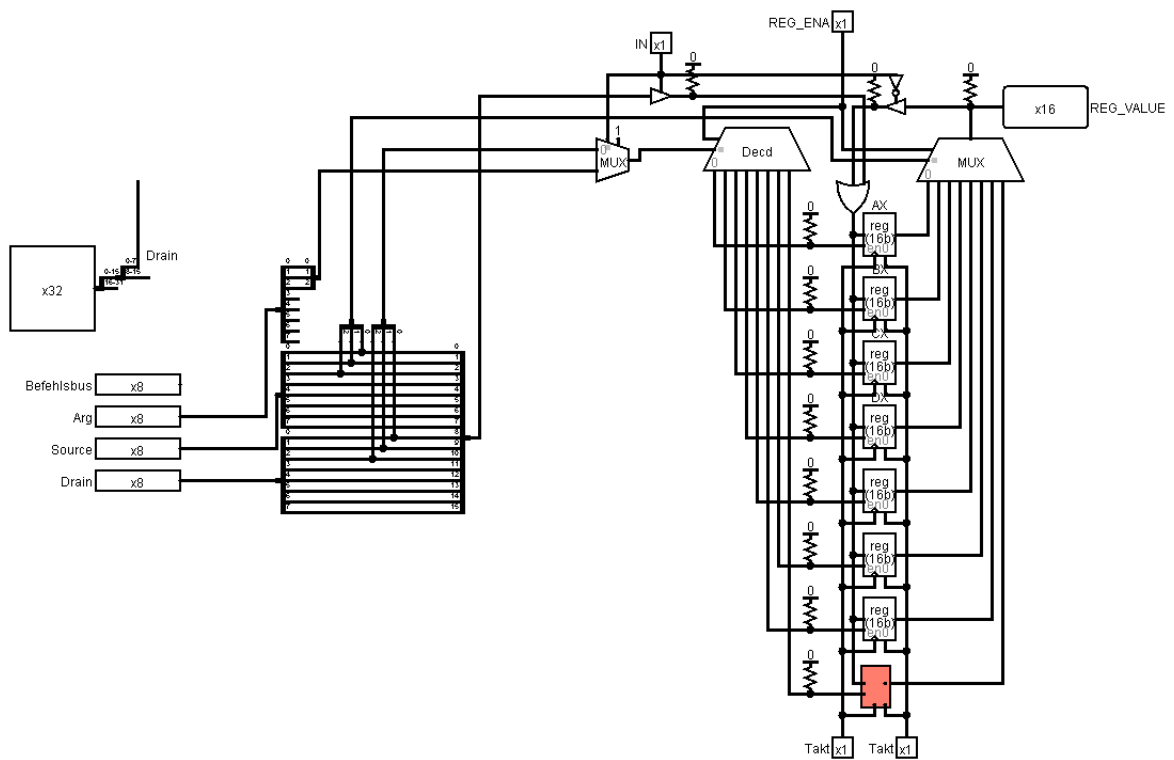


Abbildung 1: Darstellung des RegisterwerkTODO

5.2 Register

Register sind die schnellste Speichereinheit innerhalb einer CPU. Prozessoren besitzen eine vielfach höhere Ausführungsgeschwindigkeit als Arbeitsspeicher. Die CPU müsste ohne Register viele Taktzyklen auf Daten warten bevor sie diese verarbeiten könnte. Register bieten deshalb die Möglichkeit, sehr kleine Datenmengen mit einer sehr geringen Latenz prozessorintern lesen und schreiben zu können. Übliche Registergrößen sind

8,16,32 oder 64 Bit.[?] Sie werden aus Flip-Flops aufgebaut welche jeweils genau ein Bit speichern können, das heißt ein 64 Bit Register besteht aus 64 gemeinsam gesteuerten Flip-Flops.[?] Diese Art der Datenspeicherung hat allerdings auch einige Nachteile. So verbrauchen Register sehr viel Energie und Platz auf dem Prozessordie, es werden deshalb keine großen Speichermengen zur Verfügung gestellt. (Nachteile evtl streichen)

5.2.1 Universalregister

Es werden zwei Arten von Registergruppen unterschieden. In einem Universalregister kann ein Programm Werte und Variablen abspeichern. Sie stehen außerdem einem Programmierer von außen offen, das heißt er kann auf jedes Universalregister direkt zugreifen und seinen Wert verändern.

5.2.2 Spezialregister

Spezialregister werden von einer CPU für interne Zwecke genutzt. Oft sind in Prozessoren ähnliche Spezialregister zu finden.

Der StackPointer(SP) ist ein Register welches auf die aktuelle Position des Stacks im Speicher zeigt. Wenn der Befehl zur Speicherung eines Werts auf dem Stack ausgeführt wird inkrementiert die CPU automatisch, durch die interne Verschaltung des SP, den Wert des StackPointers. Dadurch zeigt das Register immer auf die nächste freie Speicheradresse im Stack.

Der InstructionPointer(IP) enthält die Adresse des nächsten Befehls im Programmspeicher der ausgeführt werden muss. Auch er wird nach der Abarbeitung eines Befehlszyklus als letzter Schritt inkrementiert. Dieses Register bietet allerdings die Möglichkeit einen anderen Wert zu laden. Das wird zur Realisierung von Sprüngen innerhalb des Programmcodes benötigt.

Das Statusregister(SR) werden zur Ausführung von bedingten Sprunganweisungen gebraucht. Sie werden auch Flagregister genannt da die ALU, in Abhängigkeit der zuletzt ausgeführten Rechenoperation, einzelne Bit(Flags) setzen kann. Auf die einzelnen Flags und ihre Bedeutung wird im Abschnitt der ALU näher eingegangen

5.3 Arithmetisch Logische Einheit

5.3.1 ALU-Konfigurationen

5.3.2 Arithmetische Operationen

5.3.3 Logische Operationen

5.4 Memory Management Unit(evtl)

5.5 Bussysteme

6 Speicher

6.1 RAM/ROM

6.2 Stack

7 Befehlsausführung

7.1 Befehlszyklus

7.2 Schleifen

8 Besondere Ausführungsarten

8.1 Interrupts

8.2 Exceptions

8.3 Subroutinen

9 Planung und Entwurf eines Prozessors

9.1 Befehlsbreite

9.2 Befehlssatz

9.3 Speicher

9.3.1 RAM/ROM