

Einführung in die moderne
Assembler-Programmierung
RISC-V spielerisch und fundiert lernen

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhaltsübersicht

	Vorwort	17
	Thema	19
Teil I	Grundlagen	27
<hr/>		
1	Das Abstrakte	29
2	Hardware	37
3	Code	49
4	Umgebung	53
5	Konzepte	59
Teil II	RISC-V	71
<hr/>		
6	Basen, Module und Profile	73
7	Register	77
8	Adressierungsarten	83
9	Speicher	85
10	Der Befehlssatz	89
11	Die Wahrheit	115

Teil III	Vertiefung	131
<hr/>		
12	Effizienter Code	133
13	Systemaufrufe und Bibliotheken	139
14	Stapel	157
15	Sprünge und Verzweigungen	167
16	Schleifen	179
17	Daten	201
18	Mathe	207
19	Künstliche Intelligenz	219
Teil IV	Projekte	229
<hr/>		
20	Eine minimale Eingabeschleife	231
21	Eine größere REPL	237
Teil V	Anhang	255
<hr/>		
22	Häufige Fehler	257
23	Stilfibel	259
24	Danksagungen	261
25	Literatur	263
	Index	269

Inhaltsverzeichnis

Vorwort	17
Thema	19
Teil I Grundlagen	27
1 Das Abstrakte	29
1.1 Zahlen	29
1.1.1 Addition von Binärzahlen	30
1.1.2 Ein Vorgriff auf die Wortbreite	30
1.1.3 Vorzeichen und Zweierkomplement	31
1.1.4 Vorzeichenerweiterung	32
1.2 Zeichen	33
1.2.1 ASCII	33
1.2.2 Besser als ASCII	34
1.2.3 Zeilenende	35
1.2.4 Stringformate	35
2 Hardware	37
2.1 Ein Überblick	37
2.2 Der Prozessor	38
2.2.1 Aufbau	39
2.2.2 CISC vs. RISC	40
2.2.3 Register	41
2.2.4 Programmzähler	42
2.2.5 Weitere Hardwarebegriffe	42

2.3	Der Arbeitsspeicher	43
2.3.1	Noch mal zur Wortbreite	44
2.3.2	Speicherarchitekturen	46
2.3.3	Ein- und Ausgabe über Adressen	46
2.3.4	Adressierungsarten	47
3	Code	49
3.1	Noch mehr Geschichte	49
3.2	Maschinensprache	50
3.3	Assembler	51
3.3.1	Opcode und Operand	51
4	Umgebung	53
4.1	Werkzeuge	53
4.2	Künstliche Intelligenz	54
4.3	Quellcode	55
4.3.1	Sektionen	56
4.3.2	Datenarten	57
4.3.3	Definitionen	58
5	Konzepte	59
5.1	Sprünge	59
5.2	Verzweigungen	60
5.3	Schleifen	60
5.4	Stapel	61
5.5	Subroutinen	63
5.6	Byte-Reihenfolge	64
5.6.1	Umgekehrte Polnische Notation (UPN)	66
5.7	Cache	66
5.8	Pipeline	68

Teil II	RISC-V	71
6	Basen, Module und Profile	73
6.1	Der Kleine: RV32E	75
7	Register	77
7.1	Die besonderen Fünf	77
7.2	Doppelbelegungen	78
7.3	Vogelfreie und geschützte Register	79
7.4	Register bei RV32E	79
7.5	Register-Synonyme	80
8	Adressierungsarten	83
9	Speicher	85
9.1	Ausrichtung	85
10	Der Befehlssatz	89
10.1	Laden und speichern	89
10.1.1	Load Immediate li	89
10.1.2	Load Address la	90
10.1.3	Move mv	90
10.1.4	Laden aus dem Speicher	91
10.1.5	Kampf der Vorzeichenerweiterung	92
10.1.6	Speicherbefehle	92
10.2	Mathe und Logik	94
10.2.1	Addition und Subtraktion	94
10.2.2	Multiplikation und Division	96
10.2.3	Division	97
10.2.4	Das Modul M bei RV64	98
10.2.5	Logikbefehle	99
10.2.6	Schiebebefehle	100
10.3	Sprünge und Verzweigungen	101
10.3.1	Nur hin: einfache Sprünge	102
10.3.2	There and back again: Subroutinen	103
10.3.3	Verzweigungen	104

10.4	Vergleichen und setzen	107
10.5	Reste	108
10.5.1	Systembefehle	108
10.5.2	Kontroll- und Status-Register	109
10.5.3	Leerbefehl	111
10.5.4	Speicherzugriffe	111
10.5.5	Besondere RV64- und RV128-Befehle	111
10.5.6	Illegale Befehle	112
10.6	Komprimierte Befehle	112
11	Die Wahrheit	115
11.1	Register nach der roten Pille	116
11.2	Mikroanatomie der Befehle	118
11.2.1	Genauerer zur Befehlslänge	120
11.3	Pseudo-Befehle	121
11.3.1	Ladebefehle	121
11.3.2	Verzweigungen	125
11.3.3	Sprungbefehle	126
11.3.4	Weitere Pseudo-Befehle	126
11.4	Macro-Op-Fusion	127
11.5	Zurück ans Licht	129
Teil III	Vertiefung	131
12	Effizienter Code	133
12.1	Das Ziel	133
12.2	Verfahren	134
12.2.1	Strength Reduction	134
12.2.2	Inlining	134
12.2.3	Verzweigungen ersetzen	134
12.2.4	Parallele Befehlsausführung	135
12.2.5	Entkopplung	137
12.2.6	Das große Bild lässt die Kirche im Dorf	138

13	Systemaufrufe und Bibliotheken	139
13.1	Systemaufrufe	139
13.1.1	Systemaufrufe bei RARS	140
13.1.2	Systemaufrufe bei Linux	141
13.1.3	Von x86 zu RISC-V	144
13.2	Die C-Bibliothek	145
13.2.1	Ausgabe: puts	145
13.2.2	Ausgabe: printf	146
13.2.3	Eingabe: scanf	146
13.2.4	Umwandlung: strtol	148
13.2.5	Und tschüss: exit	151
13.3	Schummeln macht klug	152
13.3.1	Schummeln mit C	152
13.3.2	Schummeln mit Rust	155
13.3.3	Noch viel mehr schummeln	156
14	Stapel	157
14.1	Stapelnutzung im Alltag	157
14.1.1	I love my sweet 16 stack pointer	158
14.2	Die normative Kraft des C-Moduls	159
14.2.1	Einschub: Die Wahrheit hinter der Wahrheit	160
14.3	Eigene Stapel	162
14.4	Gefahr im Überfluss	164
15	Sprünge und Verzweigungen	167
15.1	Mehrfachverzweigungen	167
15.1.1	Verzweigungsketten	168
15.1.2	Sprungtabellen	169
15.1.3	Dispatch-Tabellen	169
15.1.4	Ein Sonderfall am Rande der Welt	171
15.2	Tail Calls	171
15.3	Funktionsaufrufe	173
15.4	Millicode	174

15.5	Datenübergabe an Subroutinen	175
15.5.1	Das Anhalter-Bit	175
15.5.2	Fragwürdig bis verboten: Datenübergabe im Code	176
16	Schleifen	179
16.1	Allgemeines zu Schleifen	179
16.2	Basis + Index vs. Zeiger	181
16.2.1	Und nun ein kurzer Rant über das Fehlen eines Indexmodus	182
16.3	Techniken für effektive Schleifen	183
16.3.1	Ausrollen der Schleife	183
16.3.2	Eine volle Breitseite Wortbreite	184
16.3.3	Der Sprung in die ausgerollte Schleife	185
16.3.4	Immutables müssen draußen warten	187
16.3.5	Schleifen zusammenfassen	188
16.3.6	Cache as Cache can	188
16.3.7	Unroll and Jam	189
16.3.8	Ein Schritt zurück	190
16.3.9	Der Lohn der Verschwendung	190
16.3.10	Fallbeispiel: Die beste Schleife ist keine	192
16.4	Rekursion	195
16.4.1	Ein heiterer Ausflug zu den wirklich großen Zahlen	198
17	Daten	201
17.1	Strings mit fixierter Länge	201
17.2	Stringtabellen	202
17.3	Bitfelder	202
18	Mathe	207
18.1	Überläufe, Überträge und andere Katastrophen	207
18.1.1	Übertrag bei Addition ohne Vorzeichen	208
18.1.2	Überlauf bei Addition mit Vorzeichen	210
18.1.3	Überlauf bei der Subtraktion ohne Vorzeichen	210
18.2	(M)ultiplikation	211
18.2.1	Schieben als erste Wahl	211
18.2.2	Addieren in der Schleife	211
18.2.3	Wie in der Schule: Shift-Add	212

18.3	Division	212
18.3.1	Division durch Rechtsverschiebung	213
18.3.2	Division durch Subtraktion	214
18.3.3	Fallbeispiel: FizzBuzz	215
19	Künstliche Intelligenz	219
19.1	Allgemeines	219
19.2	Code-Generierung	222
19.3	Code-Analyse	226
19.4	Andere Anwendungen	227
Teil IV	Projekte	229
20	Eine minimale Eingabeschleife	231
20.1	Projektvorschlag: ed	233
21	Eine größere REPL	237
21.1	Ziele	237
21.1.1	Tellerstapeln (unnötig) schwer gemacht	238
21.2	Der Aufbau, iterativ	239
21.2.1	Schritt I: Nur ein Zeichen	239
21.2.2	Schritt II: Eingabe	241
21.2.3	Schritt III: Parsing (provisorisch)	242
21.2.4	Einschub: Byte-wise but time-foolish	244
21.2.5	Schritt IV: Alle Befehle	245
21.2.6	Schritt V: Dictionary	246
21.2.7	Schritt VI: Parsing (jetzt richtig)	247
21.2.8	Schritt VII: Kommandosuche	249
21.3	Nächste Schritte	252
21.4	Projektvorschlag: Forth	252

Teil V	Anhang	255
22	Häufige Fehler	257
23	Stilfibel	259
24	Danksagungen	261
24.1	Colophon	261
25	Literatur	263
	Index	269