

Implementierung eines Mikrorechners in VHDL auf einem FPGA

Markus Schneider

21. November 2016

Inhaltsverzeichnis

Abkürzungsverzeichnis	3
Abbildungsverzeichnis	4
Tabellenverzeichnis	5
1 Einleitung	6
1.1 Was ist ein FPGA?	6
1.2 Beschreibung des genutzten FPGA und Entwicklungsboards	7
1.3 Einführung in VHDL	7
2 Hardware	8
2.1 Aufbau	8
2.1.1 Steuerwerk	8
2.1.2 Rechenwerk	9
2.1.3 Dekodierer	10
2.1.4 Programmzähler	11
2.1.5 Stack	12
2.1.6 Arbeitsspeicher	13
2.1.7 Registerbelegung	14
2.2 Befehlssatz	14
2.3 Ein- und Ausgabe	14
2.3.1 Eingabe	14
2.3.2 Ausgabe	14
3 Implementierung in VHDL	15
3.1 Schemata	15
3.2 Beschreibung wichtiger Module	16
3.2.1 top – Verbindung mit der Hardware	16
3.2.2 core – Topmodul der CPU	17
3.2.3 memory_control – Speichercontroller	18
4 Fazit	19
4.1 Umfang und Aufwand	19
4.2 Ziele	20

Abkürzungsverzeichnis

EPU Educational Processing Unit

CPU Central Processing Unit

FPGA Field-programmable gate array

IC Integrated Circuit

LUT Look-Up-Table

Abbildungsverzeichnis

Tabellenverzeichnis

2.1 Registerbelegung 14

1 Einleitung

Diese Dokumentation beschreibt den Aufbau und die Funktionsweise der Educational Processing Unit (EPU). Das Projekt kam dadurch zustande, dass die Struktur und die Arbeitsweise eines Computers, insbesondere der Central Processing Unit (CPU) besser verstanden werden soll. Um dieses Ziel zu erreichen, wurde die EPU gebaut, da sie als lehrreicher Mikrorechner, wobei der Hauptteil der EPU nur aus der CPU besteht, die Funktionsweise und den Aufbau eines Alltagscomputer erklärt und somit Verständnis für die Komplexität unserer heutigen Rechner einbringt.

1.1 Was ist ein FPGA?

Ein Field-programmable gate array (FPGA) ist ein Integrated Circuit (IC), welcher zum Aufbau digitaler Schaltungen dient. Er besteht meist aus mehr als 100.000 Logikblöcken [1, S. 8]. Ins Deutsche übersetzt bedeutet FPGA soviel wie ‘im Feld programmierbare [Logik-]Gatter-Anordnung’, wobei ‘im Feld’ sich dabei auf den Konsumenten bezieht.

Die Logikschaltungen eines FPGA sind entweder über elektronische ‘Schalter’ der Konfiguration entsprechend verknüpft oder es werden sogenannte Look-Up-Tables (LUTs) benutzt, mit denen die Logikfunktion explizit realisiert werden kann. Eine LUT kann verschiedene kombinatorische Funktionen (NAND, XOR, AND, NOT, Multiplexer, etc.) aus den Eingangssignalen realisieren. Die meisten LUTs besitzen zwischen 4 und 6 Eingangssignale. Es ist auch möglich, mehrere LUTs in Serie zu schalten und die Limitierung durch die Eingangssignale zu verhindern [2].

Da der FPGA aber nach Verlust des Stromanschlusses die Konfiguration der Logikelemente nicht von selbst speichert, wird meist zusätzlich noch ein Flash-Speicher verbaut, damit nach einem Stromverlust die alte Konfiguration wieder neu geladen werden kann. Dies hat auch den Vorteil, dass dadurch der Status des FPGA zurückgesetzt wird und somit ein ‘Neustart’ schnell möglich ist.

Anders als bei üblicher Programmierung von Computern kann bei einem FPGA keine herkömmliche Programmiersprache verwendet. Das ‘Programmieren’ wird üblicherweise als Konfiguration bezeichnet und wird mithilfe einer Hardwarebeschreibungssprache wie z.B. VHDL oder Verilog erledigt.

1.2 Beschreibung des genutzten FPGA und Entwicklungsboards

1.3 Einführung in VHDL

2 Hardware

2.1 Aufbau

2.1.1 Steuerwerk

2.1.2 Rechenwerk

2.1.3 Dekodierer

2.1.4 Programmzähler

2.1.5 Stack

2.1.6 Arbeitsspeicher

2.1.7 Registerbelegung

Die EPU besitzt 16 Register, welche durch Selektion von $\log_2(16) = 4$ Adressbits angesprochen werden. Mithilfe der Tabelle 2.1 soll eine Übersicht aller Register dargestellt werden.

Selektion	Name	Zweck
0000	R0	Akkumulator
0001	R1	Allgemeine Verwendung
0010	R2	Laufvariable
0011	R3	Datenregister
0100	R4	Allgemeine Verwendung
0101	R5	Allgemeine Verwendung
0110	R6	Allgemeine Verwendung
0111	R7	Allgemeine Verwendung
1000	R8	Allgemeine Verwendung
1001	R9	Allgemeine Verwendung
1010	R10	Allgemeine Verwendung
1010	R11	Allgemeine Verwendung
1100	R12	Allgemeine Verwendung
1101	R13	Allgemeine Verwendung
1110	FLA	Flagregister
1111	ID	Interruptdaten

Tabelle 2.1: Registerbelegung

2.2 Befehlssatz

2.3 Ein- und Ausgabe

2.3.1 Eingabe

2.3.2 Ausgabe

3 Implementierung in VHDL

3.1 Schemata

3.2 Beschreibung wichtiger Module

3.2.1 top – Verbindung mit der Hardware

3.2.2 core – Topmodul der CPU

3.2.3 memory_control – Speichercontroller

4 Fazit

4.1 Umfang und Aufwand

4.2 Ziele

Literaturverzeichnis

- [1] Christopher Isreal und Marcel Jaoks. Implementierung einer 4-Bit MiniCPU in VHDL auf einem FPGA, Juli 2009.
- [2] FPGA - Mikrocontroller.net. <https://www.mikrocontroller.net/articles/FPGA#Aufbau>. Eingesehen am 21.11.2016.