

# Spezialist für FPGAs, Entwicklung von Hardware und Software, Inhouse VHDL/FPGA Training



- Stefan Milch  
Entwicklung von Hard- und Software
- 91080 Uttenreuth
- **Abschluss:** Dipl.-Ingenieur
- **Stunden-/Tagessatz:** 90.00 /Std.  
Festpreis oder nach Aufwand verhandelbar
- **Sprachkenntnisse:** englisch (verhandlungssicher) | französisch (gut) | portugiesisch (verhandlungssicher) | spanisch (gut)
- **Telefonnummer:** +49 1577 / 1725414
- **E-Mail:** stmilch@stmilch.de
- *Letztes Update: 07.01.2019*

## SCHLAGWORTE

Hardware FPGA Schulung (IT Beratung) Projekt Management Projektleitung Hardwareentwicklung µController  
Gerätetreiber VHDL

## SKILLS

Ich bin passionierter Hard- und Software Entwickler. Mein Schwerpunkt sind FPGAs, digital-Hardware und embedded Software.

Ich bin Diplom Ingenieur der Elektrotechnik und habe an der Friedrich Alexander Universität in Erlangen studiert. Mein Schwerpunkt lag dabei auf Nachrichtentechnik. Meinen Studienabschluss habe ich 1990 gemacht, ich bin aber bereits seit 1984 selbständig tätig.

### Hardware Entwicklung

Digital > 20 Jahre, Experte, zuletzt in 2018  
Analog, Leistung > 7 Jahre, Fortgeschritten, zuletzt in 2018  
Digitale Signalprozessoren >20 Jahre, Experte, zuletzt in 2018  
Emulatoren, Debugger > 10 Jahre, Experte, zuletzt in 2018  
FPGA > 20 Jahre, Experte, zuletzt in 2016  
SOC, Zynq > 5 Jahre, zuletzt in 2016  
Embedded Systeme > 20 Jahre, Experte, zuletzt in 2018  
Echtzeitsysteme > 15 Jahre, Experte, zuletzt in 2016

### Software Entwicklung

C++ > 10 Jahre, Experte, zuletzt in 2018  
C > 20 Jahre, Experte, zuletzt in 2018  
Assembler > 20 Jahre, Experte, zuletzt in 2017  
C# 1 Jahr , zuletzt in 2008

MATLAB / Simulink > 3 Jahre, Anfänger, zuletzt in 2008  
Delphi >10 Jahre, Experte, zuletzt in 2017  
JavaScript > 1 Jahr, Anfänger, zuletzt in 2018  
XML > 1 Jahre, Fortgeschrittener, zuletzt in 2018

#### Entwicklungsumgebungen

Xilinx Toolchains (ISE, PlanAhead und Vivado) inkl MicroBlaze und Zynq, Experte, zuletzt 2016  
Quartus Toolchain inkl. SOC, Experte, zuletzt 2015  
Texas Instruments Code Composer, Experte, zuletzt 2018  
Embarcadero RAD Studio (früher Borland), Experte, zuletzt in 2018  
Qt, Anfänger, zuletzt in 2015  
Lattice Diamond, Experte, zuletzt 2013

#### Betriebssysteme

Windows > 10 Jahre, Experte, zuletzt in 2017  
Linux > 5 Jahre, Anfänger, zuletzt in 2016  
Texas Instruments RTOS/BIOS > 3 Jahre, Fortgeschrittener, zuletzt 2018  
OSEK >1 Jahr, Experte, zuletzt in 2008

#### Datenbanken

MySQL  
PostgreSQL  
DBase  
MS-SQL-Server  
Paradox

#### Tools

Git  
Doxygen  
Svn  
MS Office  
Open Office  
CANoe  
CANape  
GENy  
QA-C  
InstallShield

#### Bereich

Meßtechnik >10 Jahre, Fortgeschrittener, zuletzt in 2014  
Automotive > 5 Jahre, Fortgeschrittener, zuletzt in 2008  
Medizintechnik > 1 Jahre, Fortgeschrittener, zuletzt in 1996  
Industrial > 20 Jahre, Experte, zuletzt in 2018

#### Mikrocontroller / DSP

MicroBlaze, Experte, zuletzt in 2016  
ARM > 10 Jahre, Experte, zuletzt in 2014  
Motorola/Freescale > 10 Jahre, Experte, zuletzt in 2008  
Texas Instruments > 20 Jahre, Experte, zuletzt in 2018  
Analog Devices > 10 Jahre, Fortgeschrittener, zuletzt in 2008

#### BUS, Kommunikation

LWL > 2 Jahre, Fortgeschrittener, zuletzt 2014  
GSM > 4 Jahre, Fortgeschrittener, zuletzt 2001  
PCI > 4 Jahre, Fortgeschrittener, zuletzt 2009  
CAN > 1 Jahre, Fortgeschrittener, zuletzt 2008  
TCP/IP > 3 Jahre, Fortgeschrittener, zuletzt 2016  
RS 232, RS 485, RS422, IIC, SPI, USB

#### Funktionen

Projektleitung > 1 Jahre, Fortgeschrittener, zuletzt in 2012  
Applikation > 20 Jahre, Experte, zuletzt in 2018  
HW-Architektur > 20 Jahre, Experte, zuletzt in 2018  
Hardwareentwicklung > 20 Jahre, Experte, zuletzt in 2018  
Konzeptentwicklung > 20 Jahre, Experte, zuletzt in 2018  
Projektmanagement > 20 Jahre, Fortgeschrittener, zuletzt in 2016  
SW-Architektur > 20 Jahre, Experte, zuletzt in 2018  
Softwareentwicklung > 20 Jahre, Experte, zuletzt in 2018  
Test (HW, SW, VHDL) > 20 Jahre, Experte, zuletzt in 2018

#### SoftSkills

Auffassungsvermögen  
Eigenverantwortliches arbeiten  
Flexibilität  
Kundenorientierung

Selbständige Arbeitsweise  
Teamfähigkeit  
Mehrsprachigkeit

Sonstiges  
Digitale Signalverarbeitung

## REFERENZEN

---

[11/2016 - 12/2018]

Überarbeitung einer Steuerung für einen Solarwechselrichter

---

Ausgangspunkt war ein Controller für einen leistungsstarken Solarwechselrichter mit einem älteren Texas-Instruments Delfino DSP als Hauptprozessor. Dessen Ressourcen waren, was Rechenleistung und Speicher angeht, schon voll ausgelastet. Es waren aber Erweiterungen der Firmware notwendig um die Anwendungsmöglichkeiten des Wechselrichters zu erweitern. Zudem war es notwendig, die Verarbeitungsgeschwindigkeit der Regelung zu erhöhen. Es wurde der Hauptprozessor gegen einen Delfino Prozessor neuester Generation ausgetauscht, um so notwendige Änderungen an der Firmware vornehmen zu können. Der neue Prozessor ist im Ggs. zum alten ein Dual-Core-Prozessor.

Meine Aufgaben bestanden in:

- \* Änderung der Schaltung des Controllers
- \* Erstellen des Layouts für die neue Schaltung
- \* Portierung der Firmware auf die neue Hardware
- \* Anpassung der Firmware, so dass sie den zweiten Core sinnvoll nutzen kann
- \* Erweiterung der Firmware entsprechend der Vorgaben des Kunden
- \* Erstellung eines verbesserten Debug-Interfaces für den Prozessor (Schaltplan und Layout)
- \* Erweiterung der Firmware um eine Echtzeitausgabe (RTS) von internen Zuständen auf einem DAC
- \* Erstellung eines Windows-Programms zur Steuerung des Real Time Scopes (RTS)

Stichpunkte

- \* Texas Instruments
- \* Code Composer Studio
- \* Altium Designer
- \* C
- \* svn
- \* git
- \* Embarcadero RAD Studio

[02/2016 - 07/2016]

Implementierung einer FPGA-Firmware für eine Antennensteuerung

---

In einem Folgeprojekt der Antennensteuerung vom Vorjahr wurde die FPGA Firmware erweitert.

Die Peripherie des Microblaze-Prozessors wurde um selbstgeschriebene Module erweitert und ansonsten entsprechend der geänderten Hardware angepasst.

Die Software wurde grundlegend überarbeitet. Insbesondere wurde der Kommunikationslayer neu geschrieben um automatisierte Kommunikation zwischen dem PC-Client (Test GUI) und der Steuerung zu ermöglichen. Die Objekte mit denen kommuniziert werden kann, werden aus einer XML-Datei - zusammen mit ihren Attributen - gelesen und daraus Source-Code für PC und Steuerung erzeugt.

Die Test GUI war diesmal Teil des Auftrags und wurde um ein Kommando-Interface via Virtual COM Port erweitert mit dem die Kommunikations-Objekte automatisiert angesprochen werden konnten.

Stichpunkte:

- \* VHDL
- \* XML
- \* Xilinx Spartan 6
- \* Embarcadero RAD Studio
- \* Kommunikation
- \* I2C
- \* Ethernet
- \* git
- \* doxygen

[08/2013 - 10/2016]

Entwicklung einer hochpräzisen Stromquelle für Speicherringe von Teilchenbeschleunigern

---

Um in Speicherringen, wie CERN, die geladenen Teilchen im Kreis rotieren zu lassen werden Ablenkmagnete benötigt. Diese gibt es in supraleitenden und normalen Versionen. Für einen normalen Elektromagneten werden

dazu hochpräzise Stromquellen benötigt. In meinem Projekt war deswegen eine Stromquelle gefordert, die bei bis zu 125 Volt 1000 Ampere mit einer Toleranz von weniger als 50 mA liefern kann.

Diese Stromquelle wurde aus parallel arbeitenden Einzelstromquellen erzeugt. Eine Anforderung war dabei, dass falls eine Stromquelle ausfällt, die anderen Stromquellen sofort die Last der ausgefallenen Stromversorgung übernehmen können.

Meine Aufgabe war es, die Steuerungsplatine für das System und die Steuerungsplatinen für die einzelnen Stromquellen zu entwickeln und die Steuerungskomponenten zu programmieren. Beide Platinen sind FPGA basiert (Xilinx Spartan 6) und enthalten einen Microblaze Prozessor.

Für die Kommunikation habe ich den Hardware-Layer von Ethernet mit einem proprietären Protokoll entwickelt welches den Echtzeit-Rahmenbindungen entspricht. Durch die direkte Kommunikation zwischen dem FPGA auf der Systemsteuerung und den FPGAs in den Stromquellen kann innerhalb von wenigen  $\mu$ s auf den Ausfall einer Stromquelle reagiert werden. Es wurde dadurch auch sichergestellt, dass eventuell auftretende Fehler mit minimaler Verzögerung an die Systemsteuerung übertragen werden können.

Für die Regelung war eine hochgenaue Erfassung des Stroms notwendig. Diese wurde durch einen 18 bit ADC mit nachfolgender Signalaufbereitung realisiert. Durch eine Abtastung auf einem Vielfachen der Regelbandbreite und anschließende Filterung mit Unterabtastung konnten aus dem 18 Bit ADC problemlos die geforderten 20 Bit Auflösung errechnet werden. Die Fehler der Messschaltung konnten durch Kalibrierung und Regelung der Temperatur der Messeinrichtung minimiert werden.

=====  
=====

[01/2015 - 03/2015]

Implementierung einer FPGA-Firmware für eine Antennensteuerung

-----  
Basierend auf einem Microblaze wurde eine Steuerung in ein Spartan 6 FPGA auf einem Enclustra FPGA-Modul entwickelt.

Die Hardware wurde vom Kunden entwickelt. Mein Projektanteil war das FPGA und die Software auf dem Microblaze.

Die Antennensteuerung war mehrere Kilometer von der Zentrale entfernt und über Ethernet angeschlossen. Weitere Peripherie war über IIC, SPI, RS485 und RS232 angeschlossen.

Für die Kommunikation mit der Zentrale habe ich ein Byte-Stream Protokoll entwickelt über das Kommandos gesendet und der Status abgefragt werden konnte. Basierend auf diesem Protokoll konnte auch ein Firmware-update durchgeführt werden. Durch Verwendung von zwei FPGA-Versionen (golden und actual) wurde sicher gestellt, dass bei einem Stromausfall während des Programmierens des Konfigurationsspeichers, das FPGA trotzdem wieder startete und erneut programmiert werden konnte.

Für den Test und die Inbetriebnahme habe ich eine Test GUI für Windows entwickelt, in der alle Funktionen und das Protokoll getestet werden konnten.

Stichpunkte:

- \* VHDL
- \* Xilinx Spartan 6
- \* Embarcadero RAD Studio
- \* I2C
- ?\* Ethernet
- \* git
- \* doxygen

=====  
=====

[03/2012 - 07/2013]

Entwicklung eines Hochspannungsimpulsgenerators für Linearbeschleuniger

-----  
Um Protonen-, Elektronen- oder Ionenpakete zu beschleunigen werden starke E-Felder benötigt. Diese werden von Klystronen erzeugt, die mit Hochspannungsimpulsen angeregt werden. Dabei werden Spannungspulse bis zu 500 kV bei Stromstärken bis zu 300 Ampere benötigt.

Meine Aufgabe war die Entwicklung einer FPGA-basierten Hardware die aus über hundert einzelnen Komponenten diesen Spannungspuls zuverlässig erzeugt. Durch die Potentialunterschiede können die Module nur über Lichtwellenleiter kommunizieren. Ein integraler Bestandteil des Projektes ist es eine Kommunikationsstruktur zur Verfügung zu stellen, bei der die einzelnen Komponenten, selbst bei den durch die Hochspannungsimpulse erzeugten Störungen kommunizieren können bzw. bei Störungen der Kommunikation zuverlässig arbeiten können.

Aufgrund meiner frühen Einbindung in das Projekt konnte ich das Konzept verbessern und so Entwicklungszeit und Serienkosten signifikant senken.

- Stichpunkte:
- \* VHDL
  - \* Lattice XP2
  - \* Lattice XO2
  - \* Lattice Diamond
  - \* git
  - \* doxygen

=====  
=====  
[09/2010 - 02/2012]  
Portierung einer Steuerung auf eine FPGA-Basierte Hardware  
-----

Eine in die Jahre gekommene Zentrale für eine Alarmanlage wurde auf eine neue Hardware portiert. Die gesamte Digitallogik wurde in ein FPGA integriert. Im ersten Schritt sollte die alte Software - mit minimalsten Anpassungen - auf der neuen Hardware laufen.

Meine Aufgabe bestand im Design und der Implementierung des FPGAs sowie der Beratung und Koordination der Hardware-Entwicklung und dem Layout.

Im zweiten Schritt wurde die Software erweitert um die Anlage auch über TCP/IP bedienen zu können (anstatt RS232). Auch diese Erweiterungen hatten für die alte Software transparent zu sein. Dazu wurde unter anderem das Laufzeitmodell des Prozessors erweitert um parallel zur alten (Assembler-) Software die in C geschriebene TCP/IP-Anbindung zu realisieren.

In weiteren Schritten wurde die Software und das FPGA erweitert:

1. Firmware Download
2. AES-256 Verschlüsselung
3. RSA Verschlüsselung

Weitere Projektdetails:

- \* VHDL
- \* Spartan 6
- \* ChipScope
- \* Subversion
- \* doxygen
- \* TCP/IP
- \* ulp
- \* Treiberentwicklung in C

=====  
=====  
[01/2008 - 08/2010]  
FPGA Entwicklung für einen Messgerätehersteller (Industrie)  
-----

Für eine neue Produktlinie wurde ein Datenflusskonzept entwickelt und in ein Xilinx-FPGA integriert. Zur Erfassung und Ausgabe von Daten wurde ein Framework für standardisierte Module entwickelt und für eines dieser Module das FPGA programmiert.

Das FPGA enthält einen PCI Core, der von mir integriert wurde. Über diesen Core wird das Modul parametrisiert und leitet daraufhin Datenströme zwischen den verschiedenen Schnittstellen hin und her. Die PCI Schnittstelle ist natürlich auch eine mögliche Datenquelle bzw. Datenziel. Der PCI-Transfer erfolgte über Scatter-gather DMA. Andere Schnittstellen waren z.B. ein LWL- und ein EtherCAT-Interface.

Weitere Projektdetails:

- \* VHDL
- \* Spartan 3E
- \* ChipScope
- \* Subversion
- \* doxygen
- \* Treiberentwicklung in C

=====  
=====  
[04/2007 - 10/2008]  
Portierung einer Gerätesteuerung für MBtech/Automotive  
-----

Programmierung eines Steuergerätes für Schubgelenkbusse zur Stabilisierung des Gelenks.

Ein vorhandener Knickschutz-Algorithmus musste auf eine neue Hardware und Softwareplattform portiert werden. Die neue SW musste MISRA-konform unter dem Betriebssystem OSEK implementiert werden.

Zielprozessor war ein S12X-Prozessor von Freescale (früher Motorola).

Mein Verantwortungsbereich war:

- \* Installation und Konfiguration von OSEK-OS und CANbedded von Vector
- \* Erstellung von Treibern für die neue Hardware
- \* Bereitstellung von Support zur Inbetriebnahme der neuen Hardware
- \* Entwicklung der Taskstruktur
- \* Portierung des Knickschutzalgorithmus
- \* Integration der UDS-Diagnose, eines Fehlerspeichers und der Störungsüberwachung.
- \* Erstellung einer Testumgebung für die Modultests auf dem Target-Prozessor

Verwendete Tools waren unter anderem:

- \* COSMIC C-Compiler/Debugger
- \* Lauterbach Emulator/Debugger
- \* Eclipse
- \* doxygen
- \* Subversion/Tortoise
- \* OIL-Konfigurator von Vector (OSEK-OS)
- \* GENy von Vector (Konfigurator für CANbedded)
- \* Subversion/Tortoise/Trac
- \* Matlab
- \* Visual Studio (C++)
- \* CANoe
- \* CANape
- \* QA-C

=====

[04/2006 - 03/2007]

Erstellung eines Internetportals für eine Immobiliendatenbank.

Es wurde eine ASP.NET Applikation erstellt, welche mittels SQL-Datenbank den Datenbestand und die Präsentation von Immobilien verwaltet.

Die Aufgaben waren:

- \* Design des Web-Interfaces
- \* Design der Datenbank
- \* Implementierung und Test
- \* Administration des Servers
- \* Erstellung von PC-Tools für die komfortable Datenaufbereitung und -transfer

Die verwendeten Tools waren:

- \* Borland Developer Studio
- \* PostgreSQL

=====

[01/2006 - 03/2006]

Design einer PCI-Karte mit Digital Down Converter für die digitale Signalverarbeitung.

Die Aufgaben waren:

- a) Hardware
  - \* Bauteilauswahl
  - \* Konzepterstellung
- b) FPGA:
  - \* Entwicklung eines Konzeptes zum optimalen Datendurchsatz
  - \* Design und Implementierung eines Interfaces zu einem Local-Bus
  - \* Design und Implementierung eines Interfaces zu einem Puffer-DRAM
  - \* Design und Implementierung eines recht aufwändigen Signalverarbeitungsalgorithmus in einem möglichst kostengünstigen FPGA

Branchen:

Telekommunikation

Die verwendeten Tools waren:

Xilinx ISE

ModelSim

=====  
=====  
[01/2005]

Tuner Fernsteuerung  
-----

Erstellung einer Bibliothek die einen Rhode&Schwarz Satellitentuner mit einfachen C-Funktionen steuern lässt. Der Tuner wird mittels IEEE488 bedient. Entsprechend der Anforderungen des Kunden wurde die Funktionalität des Tuners auf mächtigere Funktionen abgebildet. Ein Eventhandling wurde implementiert.

Die verwendeten Tools waren:

C-Compiler

IEEE488 Library

=====  
=====  
[2002 - 2004]

Entwicklung eines Radarechosimulators für Bosch/Automotive  
-----

Machbarkeitsanalyse. Es war zu verifizieren ob der Algorithmus überhaupt so realisierbar war.

Systemdesign, Hardware-Architektur, Software-Architektur: Aus den Ergebnissen der Machbarkeitsanalyse und den vorgegebenen Randbedingungen (integration in ein DSP-Netzwerk mit PC-Anbindung) wurden die Notwendigen Hardware und Software Komponenten spezifiziert.

Hardware-Entwicklung: Zwei Platinen waren zu entwickeln. Dokumentation, Bausteinauswahl, Schaltplaneingabe, Layout und Kontroller der Fertigungsunterlagen sowie die anschließende Integration und Test waren mein Aufgabenbereich.

FPGA und CPLD-Entwicklung: 3 Xilinx FGAs (Spartan2 und Virtex II) und 3 Xilinx CPLDs wurden von mir in VHDL entwickelt und Integriert.

Eine IO-Platine wurde von mir spezifiziert und Integriert.

Software-Entwicklung: Verschiedene Software von Mikrocode ( Floating-Point FPGA), Assembler (Floating Point DSP), C (Controller DSP) bis C++ (Bedien-Client, Server und Testprogramme/Tools) wurde von mir entwickelt.

Der Simulator wurde von mir vor Ort integriert und die Abnahme durchgeführt.

Da sich die Anforderungen beim Kunden geändert hatten, wurde war ein Redesign notwendig welches ich auch komplett übernommen habe.

Ein Redesign der Signalverarbeitung für den TigerShark-Prozessor wurde von mir durchgeführt.

Branchen:

Automobilbranche

DV-Umfeld:

32-bit, Applikation, Assembler, Automotive, CAN, CANalyzer, CANopen, CPLD, Digital, Digitale Signalprozessoren, Digitale Signalverarbeitung, Echtzeitsysteme, FPGA, Hardwareentwicklung, HW-Architektur, Modelsim, Softwareentwicklung, SPI, SW-Architektur, Texas Instruments, VHDL, Windows, Xilinx ISE

=====  
=====  
[01/1990-06/2003]

Digitale Signalverarbeitung  
-----

Ich war über lange Zeit verantwortlich für einen Teil eines Datenflussgetriebenen Systems zur digitalen Signalverarbeitung.

SW-Architektur:

Spezifikation eines Datenflussgetriebenen Signalverarbeitungssystems

Spezifikation von Modulatoren/Demodulatoren auf Basis dieses Systems

Applikation:

Entwicklung von Editoren zur Verwaltung/Debugging dieses Systems

SW-Entwicklung:

Entwicklung von diversen Signalverarbeitungsmodulen.

Branchen:

Telekommunikation

Die verwendeten Tools waren:

TMS320C30-Assembler  
TMS320C30-C Compiler  
Borland C++  
Rcs  
Cvs

=====  
=====

[1995 - 1996]

Entwicklung einer Platine für einen Magnetresonanztomographen

-----  
Als Teil eines Teams hatte ich die Hardware zu designen zu integrieren und zu dokumentieren.  
Hardware-Architektur: Anhand vorgegebener Schnittstellen und Anforderungen hatte ich die Hardware zu designen. Ein ASIC was auf dieser Platine integriert werden sollte wurde von mir spezifiziert.  
Hardware-Entwicklung: Dokumentation, Testcases, Bauteilauswahl, Schaltplaneingabe, Überwachung des Layouts und die Integration war mein Teil der Aufgaben.  
FPGA-Entwicklung: Ein komplexes FPGA, welches verschiedenste Abläufe zu steuern hatte wurde von mir in VHDL entwickelt.

Branchen:  
Medizintechnik

DV-Umfeld:  
32-bit, FPGA, Hardwareentwicklung, HW-Architektur, Texas Instruments, VHDL

Digitale Signalverarbeitung

-----  
SW-Architektur:  
Spezifikation eines Datenflussgetriebenen Signalverarbeitungssystems  
Spezifikation von Modulatoren/Demodulatoren auf Basis dieses Systems  
Applikation:  
Entwicklung von Editoren zur Verwaltung/Debugging dieses Systems  
SW-Entwicklung:  
Entwicklung von diversen Signalverarbeitungsmodulen.

Branchen:  
Telekommunikation

DV-Umfeld:  
32-bit, Applikation, Assembler, C, C++, Digitale Signalprozessoren, Digitale Signalverarbeitung, Softwareentwicklung, SW-Architektur

Portierung von Software für ein Mobiltelefon

-----  
Vorhandene Software war für ein Mobiltelefon zu portieren. Der eingesetzte Signalprozessor war proprietär und mit wenig Ressourcen ausgestattet. Es wurde dementsprechend auch in einem Assembler programmiert der proprietär war.  
Jede Routine war für sich zu portieren und durch eine geeignete Testumgebung den Nachweis zu erbringen, dass die Portierung gelungen war.

Branchen:  
Telekommunikation

DV-Umfeld:  
Assembler, Softwareentwicklung, Softwaretest