

Open Source - Mikrocontroller für Mixed Signal ASIC

Embedded Computing Conference
30. August 2011

Michael Roth

Ablauf

- Vorstellung IME
- Motivation
- Vorstellung einiger OpenSource Mikrokontroller
- Evaluation
- Fazit
- Fragen

Unser Institut

- Gegründet 1992 als MICROSWISS Zentrum Nord-Süd, seit 2006 „Institut für Mikroelektronik IME“
- Kompetenzzentrum der FHNW für Mikroelektronik, Embedded Systems-on-Chip, Signalverarbeitung und Übertragungstechnik
- Ausbildung in Bachelor und Master
- produkt-orientierte Projekte mit der Industrie
- 24 Mitarbeiter
- Mitglied von SwissT.net

swissT.net
swiss technology network
Research & Education



Motivation

- Flexibilität => “kleine CPU für kleine Aufgaben”
- Entwicklung wie für einen “gewöhnlichen” Mikrokontroller
- kein externer Prozessor nötig
- Kontrolle über den gesamten Prozess
- Langzeitwartung garantiert
- Konfiguration auf Hardwareebene hilft Chipfläche sparen
- FPGA Soft Core Prozessoren lassen sich aus Lizenzgründen nicht auf ASICs nutzen

Unsere Anforderungen an einen Open Source Mikrokontroller

- Netzliste als HDL-Code verfügbar, kostenlos, kommerziell nutzbar (Killerkriterium)
- Toolchain verfügbar, kostenlos, kommerziell nutzbar
- Core soll klein, erweiterbar und geeignet für low power Projekte sein
- Möglichkeit der Programmierung in C erwünscht
- Status: mind. stabil (Killerkriterium)
- gut dokumentiert

ZPU

Anbieter:	Zylin (www.zylin.com)
Architektur:	32Bit
Toolchain:	gcc
Lizenz:	BSD
Grösse:	442 LUTs (32kB BRAM)
Source:	VHDL, Verilog
fmax:	95MHz

openMSP430

Anbieter:	Open Cores (www.opencores.org)
Architektur:	16Bit RISC; MSP430-Klon
Toolchain:	gcc, eigene
Lizenz:	LGPL
Grösse:	1550LEs, 467FFs @ Cyclone II 8kGates @ 180nm,50MHz
Source:	Verilog
Memory:	extern; max. 63.5kB (von Neumann)
fmax:	45MHz @ Cyclone

8051 IP Core

Anbieter:	Oregano Systems (www.oregano.at)
Architektur:	8Bit CISC (16Bit Adressbus); 8051-Klon
Toolchain:	Keil, IAR, sdcc?
Lizenz:	LGPL
Grösse:	3500LUTs
Source:	VHDL
Memory:	intern: 256B, extern: max. 64kB ROM, max. 64kB RAM (Harvard)
fmax:	50MHz

T48uController

Anbieter:	Open Cores (www.opencores.org)
Architektur:	8Bit CISC; 8048-Klon
Toolchain:	AS
Lizenz:	GPL
Grösse:	738LEs, 112 Memory Bits @ Cyclone
Source:	VHDL
Memory:	intern: 64B RAM, 1024B ROM; max. 256B RAM, 4096B ROM
fmax:	59MHz

Proteus

Anbieter:	Logic Solutions Bründler (www.logicsolutions.ch)
Architektur:	8Bit 16Bit
Toolchain:	HASM, eigene
Lizenz:	Boost
Grösse:	264-1396LEs, 111-549FFs @ Cyclone II
Source:	VHDL
Memory:	max. 1024B, extern: max. 2x64kB
fmax:	>100MHz @ Cyclone II

Mico8

Anbieter:	Lattice (www.lattice.com)
Architektur:	8Bit; Lattice-Äquivalent zu Xilinx PicoBlaze
Toolchain:	gcc
Lizenz:	eigene (LGPL-ähnlich)
Grösse:	199LUTs, 71FFs, 1BRAM @ Spartan3
Source:	Verilog, (VHDL)
Memory:	intern: 32B, extern: max. 4GB
fmax:	>62MHz @ Lattice Devices

Allgemeines, Entscheid

- ARM-Prozessoren nicht als Open Source Klone verfügbar
- AVR- und PIC-Klone haben Anforderungen nicht erfüllt
- keiner der Controller bietet low power Features an

- openMSP430 und Proteus werden weiterverfolgt

Vergleich Proteus ↔ openMSP430

	Proteus	openMSP430
Minimalkonfiguration	Core (8 Bit) 2 GP-Register	Core (16 Bit) Watchdog Timer 14 IRQ 1 NMI 12 GP-Register
Peripherie	«HW-Multiplizierer» GPIO-Ports 0-6 zus. GP-Register Interrupts Timer Debugging Interface PWM UART LCD-Interface Custom Instructions	«HW-Multiplizierer» GPIO-Modul (1-6 8Bit Ports) Timer Serial Debug Interface HW Breakpoints Peripherie-Templates

Ressourcen FPGA

	Proteus	openMSP430
Max. Frequenz (Minimalkonfiguration) @ Cyclone II FPGA	125MHz [BO09]	45MHz [GO10]
Max. Frequenz (Funktionsmuster) @ Cyclone II FPGA	55MHz	35MHz
Ressourcen (Minimalkonfiguration) @ Cyclone II FPGA	264LE, 111FF [BO09]	1'552LE, 467FF [GO10]
Ressourcen (Funktionsmuster) @ Cyclone II FPGA	1'460LE, 615FF	2'400LE, 753FF

Ressourcen ASIC (ohne SCAN Insertion)

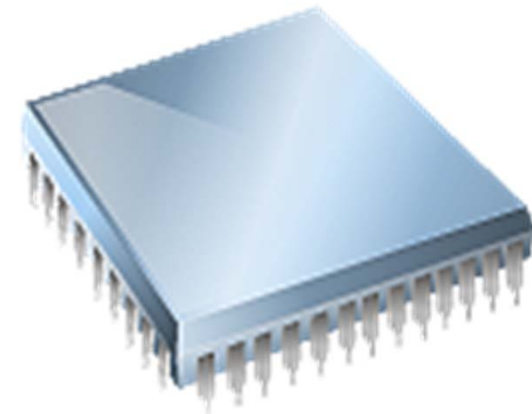
	Proteus	openMSP430
Max. Frequenz (Minimalkonfiguration) @ TSMC 180nm	-	100MHz [GO10]
Max. Frequenz (Funktionsmuster) @ XFAB 350nm	86MHz	34MHz
Ressourcen (Minimalkonfiguration) @ TSMC 180nm	-	10'000 Zellen, 100'000µm ² [GO10]
Ressourcen (Funktionsmuster) @ XFAB 350nm	9'860 Zellen 820'000µm ²	13'740 Zellen 1'140'000µm ²

Toolchain

	Proteus	openMSP430
System Building	Builder	-
Assembler	HASM (ca. EUR 15.-)	GNU Binutils
Kompiler	SDCC	GCC
Debugger	UrJTAG via Cygwin	GDB (CLI) DDD oder Eclipse (GUI)
Debugg-Schnittstelle	JTAG	Serial Link (RS232)
Zusätzliche Hardware	USB-JTAG Adapter (z.B. Amontec JTAGkey Tiny, ca. EUR 29.-)	RS232 Driver/Receiver (z.B. MAX232, ADM3202)
Entwicklungsumgebung	-	Einbindung in Eclipse

Fazit, Ausblick

- einige vielversprechende Projekte
- Informationsbereitschaft, Unterstützung der Autoren erstaunlich gut
- Proteus und openMSP430: ausgereift, gute Toolchain, erfüllen Anforderungen



Fragen?

Michael Roth

Fachhochschule Nordwestschweiz

Institut für Mikroelektronik

michael.roth@fhnw.ch

+41 56 462 46 94

Quellen

- [BO09] Bründler Oliver: Proteus – Realisierung, Rev. 1.0, 08.07.2009
- [GO10] Girard Olivier: openMSP430, Rev. 1.7, 18.08.2010