

Linux auf FPGAs

Massgeschneiderte Computersysteme

Christoph Zimmermann
Marc-André Beck

Berner Fachhochschule
MedOnStream

1. März 2008

Gliederung

- 1 GNU/Linux in eingebetteten Systemen
 - Einsatzort
 - Vorteile
 - Distribution
- 2 Einführung Programmable Logic Devices
 - Was macht diese Chips so attraktiv?
 - Einsatzbereich FPGA
 - Wann lohnt sich der Einsatz von FPGAs?
- 3 freie Software - freie Hardware
 - Datenbusse
 - Prozessoren
 - Verschmelzen von Hard- und Software
- 4 Demonstration

Gliederung

- 1 GNU/Linux in eingebetteten Systemen
 - Einsatzort
 - Vorteile
 - Distribution
- 2 Einführung Programmable Logic Devices
 - Was macht diese Chips so attraktiv?
 - Einsatzbereich FPGA
 - Wann lohnt sich der Einsatz von FPGAs?
- 3 freie Software - freie Hardware
 - Datenbusse
 - Prozessoren
 - Verschmelzen von Hard- und Software
- 4 Demonstration

Einsatzort

Gerät

Router

Navigationssysteme

Access Points

Mobiltelefone

VoIP Telefone

Beispiel

Linksys WRT54G

TomTom

FON2100

Motorola RAZR2 V8

Siemens Gigaset C455 ip

Vorteile von Linux

- Herstellerunabhängig
- Einfach zum portieren
- Treiberunterstützung
- effiziente Softwareentwicklung
- skalierbar

Freie Metadistributionen

Eine Auswahl:

- μ Clinux
- Gentoo Embedded
- OpenEmbedded

Gliederung

- 1 GNU/Linux in eingebetteten Systemen
 - Einsatzort
 - Vorteile
 - Distribution
- 2 Einführung Programmable Logic Devices
 - Was macht diese Chips so attraktiv?
 - Einsatzbereich FPGA
 - Wann lohnt sich der Einsatz von FPGAs?
- 3 freie Software - freie Hardware
 - Datenbusse
 - Prozessoren
 - Verschmelzen von Hard- und Software
- 4 Demonstration

Was sind Programmable Logic Devices?

Definition (Die Idee)

Die Hardware passend für die Anwendung zu programmieren
und nicht Software auf einer fixen Hardware zu programmieren.
So etwas wie ein *General Purpose Chip*.

Entwicklungsgeschichte Überblick

- 1970 ROMs optimiert als Look-up-table
- 1973 FPLA von Signetics
- 1978 PAL
- 198? CPLD
- 1984 FPGA
- 1998 Erster 1 Mio. Gatter FPGA

Praktischer Vergleich der verschiedenen PLD Typen

- PAL typischerweise als Adressdecoder.
- CPLD z.B. als Bus Koppler, Speicher Controller oder Antriebsregler.
- FPGA: Video Encoder, Software Defined Radio, Neuronale Netzwerke, Ein-Chip-Systeme, etc

Praktischer Vergleich der verschiedenen PLD Typen

- PAL typischerweise als Adressdecoder.
- CPLD z.B. als Bus Koppler, Speicher Controller oder Antriebsregler.
- FPGA: Video Encoder, Software Defined Radio, Neuronale Netzwerke, Ein-Chip-Systeme, etc

Praktischer Vergleich der verschiedenen PLD Typen

- PAL typischerweise als Adressdecoder.
- CPLD z.B. als Bus Koppler, Speicher Controller oder Antriebsregler.
- FPGA: Video Encoder, Software Defined Radio, Neuronale Netzwerke, Ein-Chip-Systeme, etc

Attraktiv weil...

- Design Wiederverwendung (IP Cores)
- Kleine Entwicklungsrisiken verglichen mit ASIC
- Sehr gut test- und verifizierbar
- An zukünftige Anforderungen anpassbar

Attraktiv weil...

- Design Wiederverwendung (IP Cores)
- Kleine Entwicklungsrisiken verglichen mit ASIC
- Sehr gut test- und verifizierbar
- An zukünftige Anforderungen anpassbar

Attraktiv weil...

- Design Wiederverwendung (IP Cores)
- Kleine Entwicklungsrisiken verglichen mit ASIC
- Sehr gut test- und verifizierbar
- An zukünftige Anforderungen anpassbar

Attraktiv weil...

- Design Wiederverwendung (IP Cores)
- Kleine Entwicklungsrisiken verglichen mit ASIC
- Sehr gut test- und verifizierbar
- An zukünftige Anforderungen anpassbar

Einsatzbereich FPGA

Typische Produktbereiche sind:

- Telekommunikation
- RADAR, SONAR
- Bild-/Videoverarbeitung
- High Performance Computing
- Kryptographie
- Luft und Raumfahrt

Gliederung

- 1 GNU/Linux in eingebetteten Systemen
 - Einsatzort
 - Vorteile
 - Distribution
- 2 Einführung Programmable Logic Devices
 - Was macht diese Chips so attraktiv?
 - Einsatzbereich FPGA
 - Wann lohnt sich der Einsatz von FPGAs?
- 3 freie Software - freie Hardware**
 - Datenbusse
 - Prozessoren
 - Verschmelzen von Hard- und Software
- 4 Demonstration

- FPGAs ermöglichen freie Hardware zu erstellen.
- FPGAs sind aber unfrei.

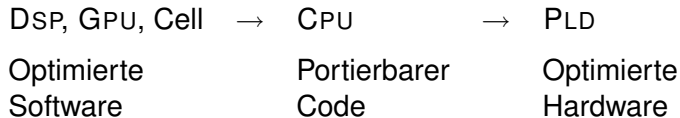
- FPGAs ermöglichen freie Hardware zu erstellen.
- FPGAs sind aber unfrei.

Datenbusse

Bus	Organisation	Lizenz
Wishbone	OpenCores	Public Domain
Coreconnect	IBM	Proprietär, frei nutzbar
AMBA	ARM	Proprietär, frei nutzbar

Prozessoren

Prozessor	Bus	Lizenz
OpenRISC	Wishbone	LGPL
AEMB	Wishbone	LGPL
S1 (T1)	Wishbone	GPL
LEON	AMBA	GPL dual



Gliederung

- 1 GNU/Linux in eingebetteten Systemen
 - Einsatzort
 - Vorteile
 - Distribution
- 2 Einführung Programmable Logic Devices
 - Was macht diese Chips so attraktiv?
 - Einsatzbereich FPGA
 - Wann lohnt sich der Einsatz von FPGAs?
- 3 freie Software - freie Hardware
 - Datenbusse
 - Prozessoren
 - Verschmelzen von Hard- und Software
- 4 **Demonstration**

GECKO3 Openhardware Plattform

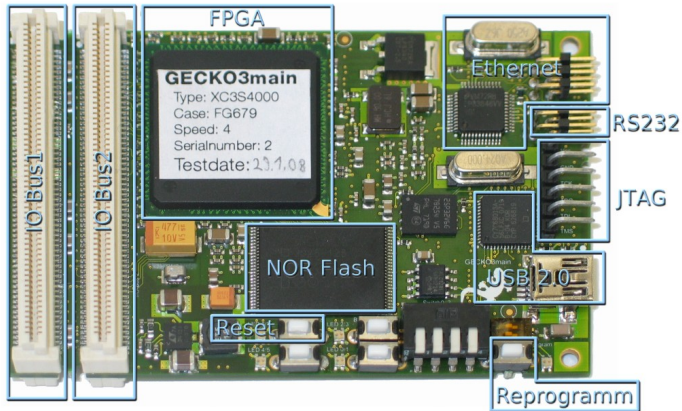


Abbildung: Das GECKO3main Modul

GECKO3main Blockdiagramm

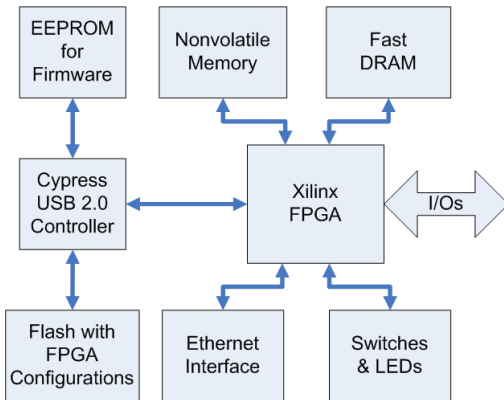


Abbildung: Blockdiagramm des GECKO3main Moduls

Aufbau des Demonstrationssystems

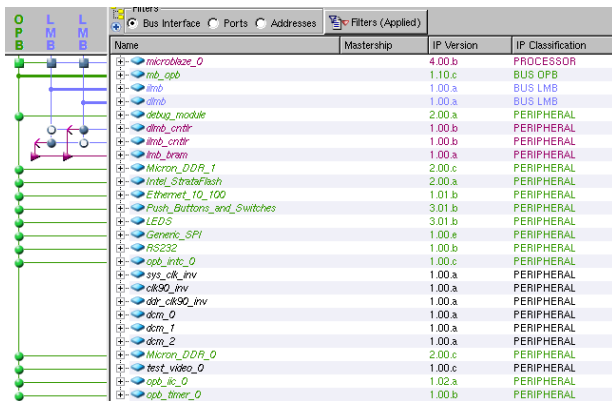
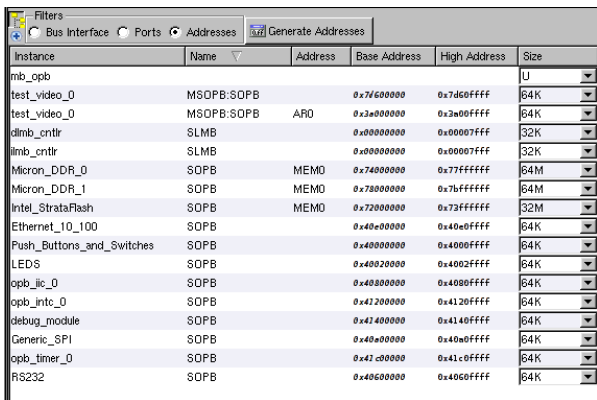


Abbildung: Aufbau des Demonstrationssystems.

Zu sehen sind die verwendeten Cores und die Struktur der Busse

Aufbau des Demonstrationssystems



The screenshot shows a software interface with a 'Filters' section at the top containing radio buttons for 'Bus Interface', 'Ports', and 'Addresses', along with a 'Generate Addresses' button. Below this is a table with the following columns: Instance, Name, Address, Base Address, High Address, and Size. The table lists various hardware components and their memory addresses and sizes.

Instance	Name	Address	Base Address	High Address	Size
mb_opb					U
test_video_0	MSOPB:SOPB		0x7f600000	0x7d60ffff	64K
test_video_0	MSOPB:SOPB	AR0	0x3a000000	0x3a00ffff	64K
dlimb_cntrl	SLMB		0x00000000	0x00007fff	32K
lmb_cntrl	SLMB		0x00000000	0x00007fff	32K
Micron_DDR_0	SOPB	MEM0	0x74000000	0x77ffffff	64M
Micron_DDR_1	SOPB	MEM0	0x78000000	0x7bffffff	64M
Intel_StrataFlash	SOPB	MEM0	0x72000000	0x73ffffff	32M
Ethernet_10_100	SOPB		0x40e00000	0x40e0ffff	64K
Push_Buttons_and_Switches	SOPB		0x40000000	0x4000ffff	64K
LEDS	SOPB		0x40020000	0x4002ffff	64K
opb_iic_0	SOPB		0x40800000	0x4080ffff	64K
opb_intc_0	SOPB		0x41200000	0x4120ffff	64K
debug_module	SOPB		0x41400000	0x4140ffff	64K
Generic_SPI	SOPB		0x40a00000	0x40a0ffff	64K
opb_timer_0	SOPB		0x41c00000	0x41c0ffff	64K
RS232	SOPB		0x40600000	0x4060ffff	64K

Abbildung: Aufbau des Demonstrationssystems. Zu sehen sind die Adressbereiche der Cores

Fragen?

Fragen?

labs.ti.bfh.ch/gecko/
christoph.zimmermann@bfh.ch
info@medonstream.com