

Linux auf dem Nios II Softcore Prozessor

Tobias Klauser <klto@zhaw.ch>

Institute of Embedded Systems
Zürcher Hochschule für Angewandte Wissenschaften

30. August 2011

- 1 Agenda
- 2 Institute of Embedded Systems ZHAW
- 3 Übersicht FPGA und Softcore Prozessoren
- 4 Übersicht Linux
- 5 Linux auf dem Nios II
- 6 Anwendungsbeispiele

Institute of Embedded Systems

- F&E Institut der ZHAW
- Rund 45 Mitarbeiter
- Schwerpunkte
 - ▶ Industrielle Kommunikation
 - ▶ Zeitsynchronisation und Hochverfügbarkeit
 - ▶ System on Chip Design
 - ▶ Wireless Communication, Low Power
 - ▶ Entwicklungsmethoden
- Projekte in Zusammenarbeit mit Industriepartnern
- Für mehr Informationen: <http://ines.zhaw.ch>

Zürcher Hochschule
für Angewandte Wissenschaften



FPGA und Softcore

- Field Programmable Gate Array
 - ▶ Schaltung wird in einer Beschreibungssprache (VHDL, Verilog) codiert
 - ▶ Code wird synthetisiert und FPGA damit programmiert
 - ▶ Änderungen am Design relativ einfach, Custom Systems
 - ▶ Parallelität
- Softcore
 - ▶ Prozessor innerhalb des FPGA implementiert
 - ▶ Fast immer Bestandteil eines System-on-a-Chip
 - ▶ Konfigurierbar (z.B. MMU), erweiterbar (z.B. Custom Instructions)
 - ▶ Proprietär, herstellerabhängig: Nios II (Altera), MicroBlaze (Xilinx), Cortex M1 (ARM)
 - ▶ Open-Source Softcores: Gaisler LEON, OpenRISC, LatticeMico32

Nios II

- 32-bit RISC Softcore-Prozessorarchitektur für Altera FPGAs
- 3 Basisvarianten:
 - ▶ Economy (600-700 LEs)
 - ▶ Standard (1200-1400 LEs)
 - ▶ Fast (2600-3000 LEs)
- Konfigurierbar mit oder ohne Memory Management Unit (MMU)
- Optionale Hardware MUL/DIV Instruktionen, Custom Instructions
- Peripherie (Timer, UART, Ethernet MAC, Flash Controller, ...) als IP über SOPC System konfigurierbar

Wieso ein Betriebssystem?

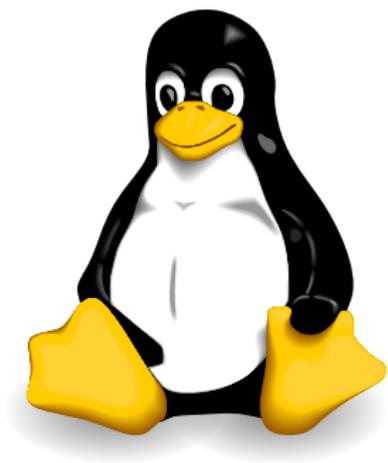
- Multi-Tasking, Multi-User
- Memory Management, Memory Protection (MMU benötigt)
- TCP/IP Stack
- Filesysteme
- Bestehende Software und Treiber

Wieso (Embedded) Linux?

- Skalierbarkeit: Von “Big Iron” bis zum Embedded System
 - ▶ Starke Verbreitung im Bereich Embedded Systems (z.B. Android auf Smartphones)
- Umfangreiches Software-Ökosystem, grosse Community
- Einheitliche Entwicklungsumgebung (Unix-artig, POSIX-Standard)
- Portierungen auf zahlreiche (~ 30) Prozessor-Architekturen
- Niedrige Kosten, freie Lizenzen

Embedded Linux

- Linux Kernel als Kernkomponente eines Linux Systems
- Libraries und Applikationen werden zusätzlich benötigt, um ein komplettes System zu erhalten
- Minimale Voraussetzungen:
 - ▶ 32-bit Prozessor (mit oder ohne MMU)
 - ▶ unterstützte Toolchain (GCC, binutils, ...)
 - ▶ libc (uClibc, glibc, ...)



Zürcher Hochschule
für Angewandte Wissenschaften

zhaw School of
Engineering
InES Institute of
Embedded Systems

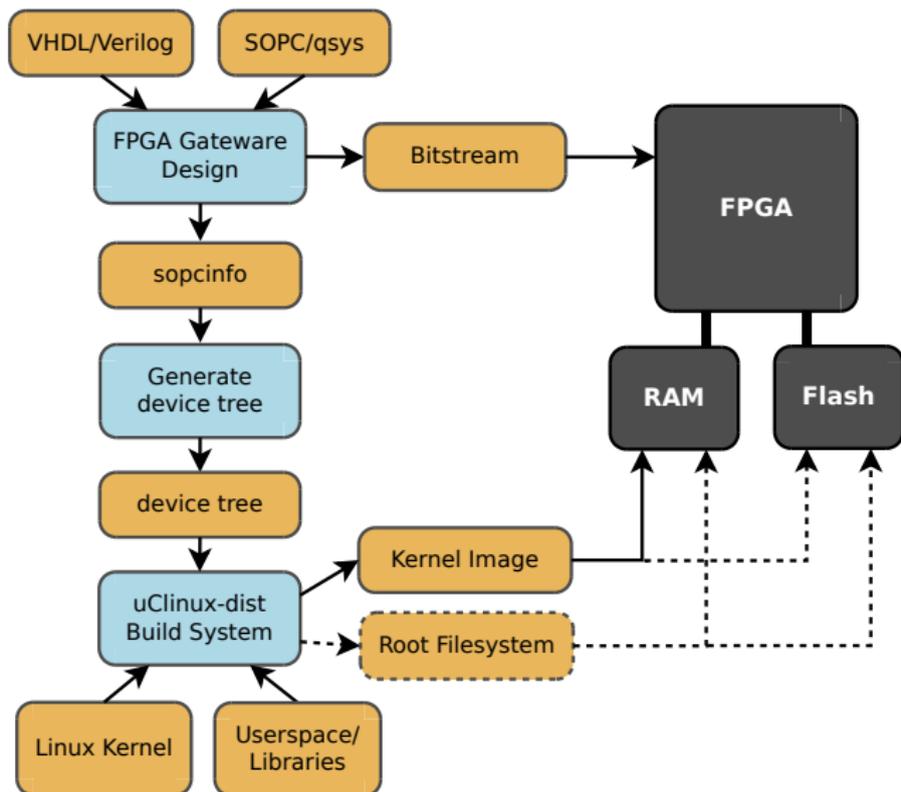
Linux/uClinux auf dem Nios II

- Linux unterstützt Nios II mit *und* ohne MMU
 - ▶ Erster Port von Microtronix für Nios II NOMMU (Kernel 2.6.11)
 - ▶ Separater Port von Wind River für Nios II MMU (Kernel 2.6.21)
- Open Source Nios II Linux (alterawiki.com/wiki/Linux)
 - ▶ MMU und NOMMU Ports zusammengeführt
 - ▶ Zahlreiche Verbesserungen, Anpassungen an aktuelle Linux Kernel Versionen
 - ▶ Integration in Mainline (kernel.org) geplant
 - ▶ Verwendet uClinux-dist als Distribution
- Kommerzielle Anbieter
 - ▶ Auf Basis von Open Source Nios II, eigene Distribution
 - ▶ Wind River, Timesys, SLS

Nios II mit oder ohne MMU?

MMU	NOMMU
<i>mehr</i> LEs im FPGA	<i>weniger</i> LEs im FPGA
Virtual Memory, separater Adressraum für jeden Prozess	ein globaler Adressraum für alle Prozesse und Kernel
Shared Libraries	Libraries werden statisch zu jedem Binary gelinkt
ELF Fileformat	Flat Fileformat
keine Änderungen an Userspace-Software notwendig	u.U. Anpassungen an Software notwendig (z.B. kein <code>fork()</code>)
langsamer	schneller

Development Workflow



Linux Image & Root Filesystem

- Kernel Image als komprimiertes ELF-Binary
 - ▶ Kann direkt in RAM geladen und gestartet werden
 - ▶ Wird u-boot verwendet, muss das Image konvertiert werden
- Root Filesystem in einem `initramfs` an Kernel angehängt
- Keine Persistenz über Reboot hinweg
 - ▶ Alternative: Root FS separat in Flash speichern
 - ▶ NFS (Network Filesystem) verwenden

Device Tree (devicetree.org)

- Strukturierte Beschreibung der Hardware
- Integration mit Linux:
 - ▶ Als Bestandteil des Kernel Images
 - ▶ Im Flash (nur mit u-boot)
 - ▶ Laden über TFTP (nur mit u-boot)
- Linux extrahiert Device Tree, lädt Treiber entsprechend
- Von zahlreichen Linux-Ports eingesetzt (ARM, PowerPC, MicroBlaze, OpenRISC)

```
cpu_0: cpu@0x0 {
    compatible = "ALTR,nios2-9.1";
    clock-frequency = <50000000>;
    dcache-size = <2048>;
    icache-size = <4096>;
    ALTR,implementation = "fast";
    ALTR,has-div;
    ALTR,has-mul;
    ALTR,reset-addr = <0xc4000000>;
    ...
};

uart_0: serial@0x6000000 {
    compatible = "ALTR,uart-9.1";
    reg = <0x6000000 0x20>;
    interrupt-parent = <&cpu_0>;
    interrupts = <2>;
    current-speed = <115200>;
    clock-frequency = <50000000>;
};
```

Zürcher Hochschule
für Angewandte Wissenschaften

Bootprozess

- ① Laden des FPGA-Bitstreams
- ② Laden des Linux Images inkl. Reset des Nios
- ③ Nios beginnt mit Ausführung des Linux Kernels
- ④ Linux Kernel erkennt Hardware anhand des Device Tree und lädt entsprechende Treiber
- ⑤ `initramfs` wird durch den Kernel ins RAM entpackt und gemountet
- ⑥ Init-Prozess (`/sbin/init`) wird ausgeführt
- ⑦ Optional: Mounten von von Flash- bzw. Netzwerk-Filesystemen
- ⑧ Weitere Userspace-Prozesse (Daemons etc.) werden durch Init gestartet

Anwendungsbeispiel - Redundantes Kommunikationsnetzwerk

- Nios II mit MMU (anfangs ohne MMU)
- Ethernet-basiert, optional SHDSL
- Parallel Redundancy Protocol
- Software-Komponenten (COTS und Eigenentwicklung) u.a. für SHDSL, Webserver, Remote Update, SNMP, NTP
- Treiber für eigene IP-Komponenten

Anwendungsbeispiel - Übertragung von HD-Videodaten

- Nios II ohne MMU
- Gigabit-Ethernet
- USB Host- und Device-Controller
- Software-Komponenten (COTS und Eigenentwicklung) u.a. für Session-Setup, Konfiguration, Webserver
- Treiber für eigene IP-Komponenten und Hardware

Zusammenfassung

- Nios II als flexibel konfigurierbarer Softcore-Prozessor innerhalb eines System-on-a-Chip
- Linux bietet umfangreiche Hardware- und Software-Unterstützung
- Softcore-System und Linux sind nachträglich einfach anpass- und erweiterbar
- Integration von bestehender Linux/Unix-Software mit kleinen oder geringen Anpassungen möglich

Danke für Ihre Aufmerksamkeit

Tobias Klauser
Institute of Embedded Systems
Zürcher Hochschule für Angewandte Wissenschaften
klto@zhaw.ch

Weitere Informationen zu Nios II Linux:
<http://www.alterawiki.com/wiki/Linux>