

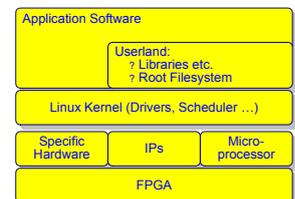
FPGA-basierte Embedded Systeme mit Embedded Linux seit 2009

- flexible Entwicklung durch FPGA-basierte Embedded Systeme (Xilinx Spartan/Virtex)
- verschiedene Prozessorarchitekturen zur Auswahl (PowerPC, SPARC, MicroBlaze)
- benötigte Peripherie als IPs oder spezifische Eigenentwicklungen



- angepasstes Embedded Linux als Betriebssystem

- Kernel mit Treibern und Dateisystem(en)
- Userland / Root Filesystem mit Bibliotheken und Dienstprogrammen
- Bootfunktionalität (Wrapper, Bootloader)



- direktes, applikationsspezifisches Erzeugen des Linux (Linux from Scratch)



- Verwendung von Open-Source-Betriebssystemkomponenten

- Kernel (Vanilla Kernel, Xilinx Kernel = Vanilla Kernel & Xilinx-spezifische Treiber)
- Userland (Buildroot) = Ordnerstruktur, Dienstprogramme (Busybox), Boot- und Init-Skripte
- Bootloader (Das U-Boot)



- Toolchain (Cross Compiler, Binutils)

- Erstellung entweder manuell (aufwändig), automatisiert (Crosstool-ng) oder fertige Binaries
- legt C-Bibliotheken fest (glibc, uclibc, elibc)

- Anpassung und Einsatz von Open-Source-Applikationen (z.B. Telefonanlagensoftware Asterisk, Askozia)



Ansprechpartner: Prof. Dr.-Ing. Rainer Bermbach (r. bermbach@ostfalia.de, 05331 / 939 - 42620)

Ostfalia Hochschule für angewandte Wissenschaften, Standort Wolfenbüttel, Salzdahlumer Str., 46/48, 38302 Wolfenbüttel



FPGA-basierte Embedded Systeme mit Embedded Linux

Rainer Bermbach, Mathias Krüger

Ostfalia Hochschule für angewandte Wissenschaften
Fakultät Elektrotechnik
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel
r.bermbach@ostfalia.de, math.krueger@ostfalia.de

Um den Anforderungen moderner Prototypen gerecht zu werden, setzt man sehr häufig FPGAs als Entwicklungsgrundlage ein. Mit ihrem Einsatz erhält man eine flexible Möglichkeit das Embedded System an die gegebene Zielstellung anzupassen. Diese Flexibilität kann softwareseitig durch den Einsatz eines Betriebssystems unterstützt werden, denn immer mehr moderne Embedded Systeme müssen auch nicht-anwendungsspezifische Aufgaben übernehmen, die sich für alle Embedded Systeme gleichen. Ein sehr weit verbreitetes und leistungsfähiges Betriebssystem ist GNU/Linux, welches aufgrund seiner starken Anpassungsfähigkeit sehr oft im eingebetteten Bereich eingesetzt wird.

Mit dem Einsatz beispielsweise eines FPGA-Entwicklungsboards erhält man eine flexible Grundlage, um u.a. unterschiedliche Prozessorarchitekturen zu implementieren. Xilinx bietet hierbei Unterstützung für PowerPC- und MicroBlaze-Systeme an. Man kann jedoch auch quelloffene Architekturen einsetzen, wie z.B. den Leon 3 (Gaisler Aeroflex). Benötigte Peripherie lässt sich in Form von IPs oder spezifischen Eigenentwicklungen über den Systembus an das Embedded System anbinden.

Neben diversen Distributionen, die den Entwickler beim Erstellen eines Linux-basierten Betriebssystems unterstützen, gibt es auch die Möglichkeit *Linux from Scratch* und somit direkt und applikations-spezifisch zu erstellen. Hierbei muss man die einzelnen Komponenten des Betriebssystems einzeln erzeugen, deren wichtigste der Kernel ist. Der monolithisch aufgebaute Linux-Kernel beinhaltet neben Treibern und Dateisystemen beispielsweise noch die Speicherverwaltung und einen Scheduler. Ein weiterer Bestandteil ist das Userland, welches in einer festen Ordnerstruktur Bibliotheken, Dienstprogramme sowie Boot- und Init-Skripte bereit stellt. Um letztendlich das Betriebssystem in einen lauffähigen Zustand zu bringen, benötigt man noch einen Boot Wrapper oder einen Bootloader.

Jede einzelne Betriebssystemkomponente kann man mit Hilfe von Open-Source-Projekten erstellen. Der Standard-Linux-Kernel ist der Vanilla Kernel. Er wird von Linus Torvalds bereitgestellt und bildet die Grundlage für die meisten anderen Kernel. So bietet z.B. Xilinx einen um spezifische Treiber erweiterten Kernel an. Zum Erstellen des Userlands empfiehlt sich Buildroot. Es erzeugt alle benötigten Teile eines Userland und greift dabei beim Erzeugen der Dienstprogramme auf Busybox zurück. Ein beliebter und leistungsfähiger Bootloader ist *Das U-Boot*, der nicht nur verschiedene Bootquellen nutzen kann, sondern ist auch in der Lage ist, nützliche Debug-Informationen auszugeben.

Zum Erstellen der Software benötigt man verschiedene Tools, die nacheinander durchlaufen werden und somit Toolchain genannt werden. Zu einer Toolchain gehören der Cross Compiler und sämtliche Programme der Binutils. Es gibt verschiedene Wege eine Toolchain zu erhalten. Neben einer sehr aufwändigen manuellen Generierung und der automatisierten Lösung mit Skriptsammlungen wie Crosstool-ng bieten einige Hersteller vorkompilierte Binaries an, die meist gut getestet und somit fehlerärmer sind. Die Toolchain bestimmt auch die eingesetzte C-Bibliothek, gegen die die späteren Anwendungen kompiliert werden sollen. Wichtige Vertreter sind hierbei die glibc, die uclibc und die elibc.

Betriebssystem und Hardware bilden die Grundlage für die Anwendungen des Embedded Systems, wobei auch hier sich ein Blick in die Open-Source-Welt lohnt (beispielsweise Asterisk und Askozia für Telefonanlagen). Somit erweisen sich FPGA-basierte Systeme mit Linux immer häufiger als sehr flexible und leistungsfähige Basis für Entwicklung und Betrieb von modernen Embedded Anwendungen.