Forschungsbericht SS 2011

Echtzeitfähigkeit mit dem Linux RT-Preempt Patch in FPGA-basierten Prozessorsystemen

Prof. Dr.-Ing. Rainer Bermbach

Einleitung

Viele Anwendungen von Embedded Systemen erfordern sog. Echtzeitfähigkeit, d.h. eine definierte Reaktionszeit auf Ereignisse. Man unterscheidet dabei harte und weiche Echtzeitfähigkeit. Bei harter Echtzeit muss die Reaktion unter allen Bedingungen innerhalb der gegebenen Zeit erfolgen, damit das System nicht fehlerhaft arbeitet. Typische Beispiele bilden Maschinensteuerungen oder bestimmte medizinische Geräte. Weiche Echtzeit „bemüht“ sich, die Aufgaben innerhalb der gesteckten Grenzen zu erledigen. Nichterreichen dieser Werte hat jedoch meist nur eine Art Qualitätsverschlechterung zur Folge wie z.B. bei der Verarbeitung oder Übertragung von Audio oder Video, wo eine kurze Störung wahrzunehmen ist, aber keine weiter gravierenden Fehler entstehen.

Standardbetriebssysteme wie Windows und Linux erlauben keinerlei Echtzeitverarbeitung, weil sie für ganz andere Aufgaben ausgelegt sind. Mit dem Einsatz von Linux in Embedded Systemen wuchs aber der Wunsch nach Echtzeitmöglichkeit für entsprechende Anwendungen. Seit geraumer Zeit gibt es sog. Echtzeiterweiterungen für Linux wie RTLinux, RTAI oder Xenomai. Im Prinzip arbeiten diese alle in gleicher Weise: Ein echtzeitfähiger Betriebssystemkern kümmert sich um die Aufgaben mit harten Echtzeitanforderungen und Linux selbst läuft als eine Task (Aufgabe) des Echtzeitkerns mit niedriger Priorität, d.h. wenn gerade keine Echtzeitaufgaben erfolgen müssen.

Da diese Echtzeiterweiterungen aber völlig andere Programmierschnittstellen mit sich bringen, gibt es auch Bestrebungen, Linux selbst „echtzeitfähiger“ zu machen. Seit der Kernel-Version 2.6 gibt es Verbesserungen wie einen neuen (O(1)) Scheduler und die Wahl zwischen verschiedenen Preemption Models. Mit der Wahl der Konfiguration „config_preempt“ können große Teile des Betriebssystemkerns von Linux unterbrochen (preempted) werden, was ein deterministischeres Zeitverhalten verspricht. Weiterhin sind sog. High-Resolution Timer (HRTIMER) verfügbar, die eine viel höhere zeitliche Auflösung von Ereignissen ermöglichen, als es der bisherige „System Tick“ (Jiffies) erlaubte.

Da es aber auch mit diesen Verbesserungen u.U. schwierig bis unmöglich ist, selbst weiche Echtzeitsysteme unter Linux zu realisieren, wird ein Real-Time Patch (config_preempt-rt; Patch = Erweiterung/Veränderung des Betriebssystemkerns) zur Verfügung gestellt, der solche Applikationen möglich macht. Diese Patches sind für eine Reihe von Prozessorarchitekturen vorhanden, aber längst noch nicht für alle.

Untersuchungen zum Status und den Funktionalitäten des Linux RT-Preempt Patch für PowerPC und MicroBlaze

Eine detailliertere Untersuchung zeigte, dass der RT-Preempt-Patch entgegen anderslautender Hinweise im Web derzeit nur die Architekturen x86, PowerPC, ARM, SPARC und MIPS unterstützt. Die Verfügbarkeit für den MicroBlaze von

Xilinx ist zwar seit einiger Zeit [Siemek2010] angekündigt, war aber zur Laufzeit des Projekts immer noch nicht verfügbar. Eine versuchsweise eigene Anpassung schied aus Aufwandsgründen aus.

Zur Untersuchung des RT-Preempt-Patch auf FPGA-basierten Prozessorsystemen konnte also nur der Hardcore-Prozessor PowerPC 405 (300 MHz) und nicht der Softcore MicroBlaze herangezogen werden. Für vergleichende Messungen entstand aber ein konventionelles x86-kompatibles Embedded System, das auf einem Geode-Prozessor (500 MHz) basiert.

Ende Juli [Gleixner2011] wurde eine völlig neu überarbeitete Version des RT-Preempt Patch verfügbar, die aber aufgrund ihres Status (Fehlerfreiheit etc.) und der Terminlage nicht im Rahmen des Projektes untersucht werden konnte.

Um weiterhin eine bessere Vergleichbarkeit der Echtzeiteigenschaften zu erhalten, wurde die Echtzeiterweiterung Xenomai miteinbezogen. Xenomai baut eine eigene Echtzeitverarbeitung auf, die sog. Adeos I-Pipe, die sich um alle Interrupts kümmert. Linux läuft unter Xenomai als eigenständige Task mit niedrigerer Priorität, so dass alle zeitkritischen Aktionen von Xenomai behandelt werden können. Xenomai (www.xenomai.org) unterstützt zurzeit die Architekturen ARM, x86, PowerPC, Blackfin, Nios II (ebenfalls ein FPGA-basierter Prozessor, für den jedoch keine Entwicklungsumgebung zur Verfügung steht). Der MicroBlaze wird noch nicht unterstützt, auch wenn Berichte über eine (nicht verfügbare) Portierung zu finden sind [Bétizeau2010]. Der Vorteil von Xenomai liegt in der Flexibilität seiner Programmierschnittstelle (API). Xenomai stellt außer seiner spezifischen API (native) weitere APIs über sog. Skins zur Verfügung, darunter auch eine POSIX-kompatible. POSIX-kompatible Echtzeit-Systemaufrufe sind praktisch, da diese auch von Linux (bei Verwendung der glibc, nicht bei der uclibc) unterstützt werden [Pengu2007].

Eine gute Einführung in die Problematik der Echtzeitverarbeitung sowie die Programmierung unter Linux mit RT-Preempt bzw. Xenomai bieten neben anderen [Opden2011] und [RTWIKI2011a], so dass hier auf eine breitere Ausführung verzichtet wird.

Notwendige Voraussetzungen in Hardware und Software für die Implementation entsprechender Linux-Systeme

Sowohl der RT-Preempt Patch als auch Xenomai benötigt einen hochauflösenden Zähler, vorzugsweise einen sog. Prozesszyklenzähler (TSC – Time Stamp Counter o.ä.), um die verbesserte zeitliche Auflösung realisieren zu können. Während RT-Preempt auch andere Zählertypen (z.B. APIC Timer, HPET) nutzen kann, benötigt Xenomai einen Prozesszyklenzähler. Auch der PowerPC 405 verfügt über einen solchen Zähler, der bei ihm TBR – Time Base Register (64 Bit) heißt. Der MicroBlaze verfügt von Haus aus über keinen solchen Timer. Hier müsste ein entsprechender Timer IP hinzugefügt und für Linux bzw. Xenomai nutzbar gemacht werden.

Auf der Softwareseite müssen die entsprechenden Versionen des RT-Preempt Patch und von Xenomai selbstverständlich mit der passenden Linux-Version kombiniert werden, um Inkompatibilitäten zu vermeiden. Auch für die Interruptverarbeitung von Xenomai (Adeos I-Pipe) muss der Linux-Kernel ja entsprechend gepatcht werden.

Systemimplementationen

Beide Echtzeitsysteme, Xenomai und der RT-Preempt Patch für Linux, wurden auf beiden Hardwareplattformen implementiert. Das FPGA-System basierte auf dem im Labor vorhandenen Board XUPV2P von Xilinx (s. Abbildung1), das einen Virtex II Pro trägt, 256 MByte DRAM besitzt sowie über eine Reihe von Interface-Schaltungen verfügt. Auf dem XUPV2P arbeitet ein PowerPC 405 mit 300 MHz.

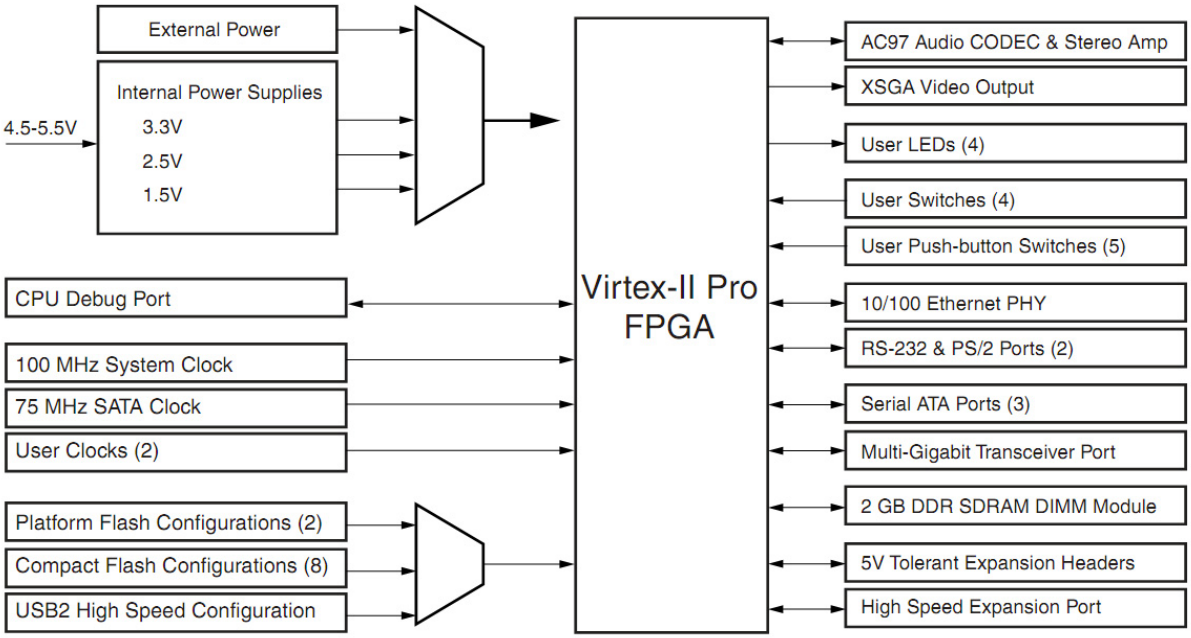

Abb. 1: Blockschaltbild XUPV2P (Quelle: Xilinx)

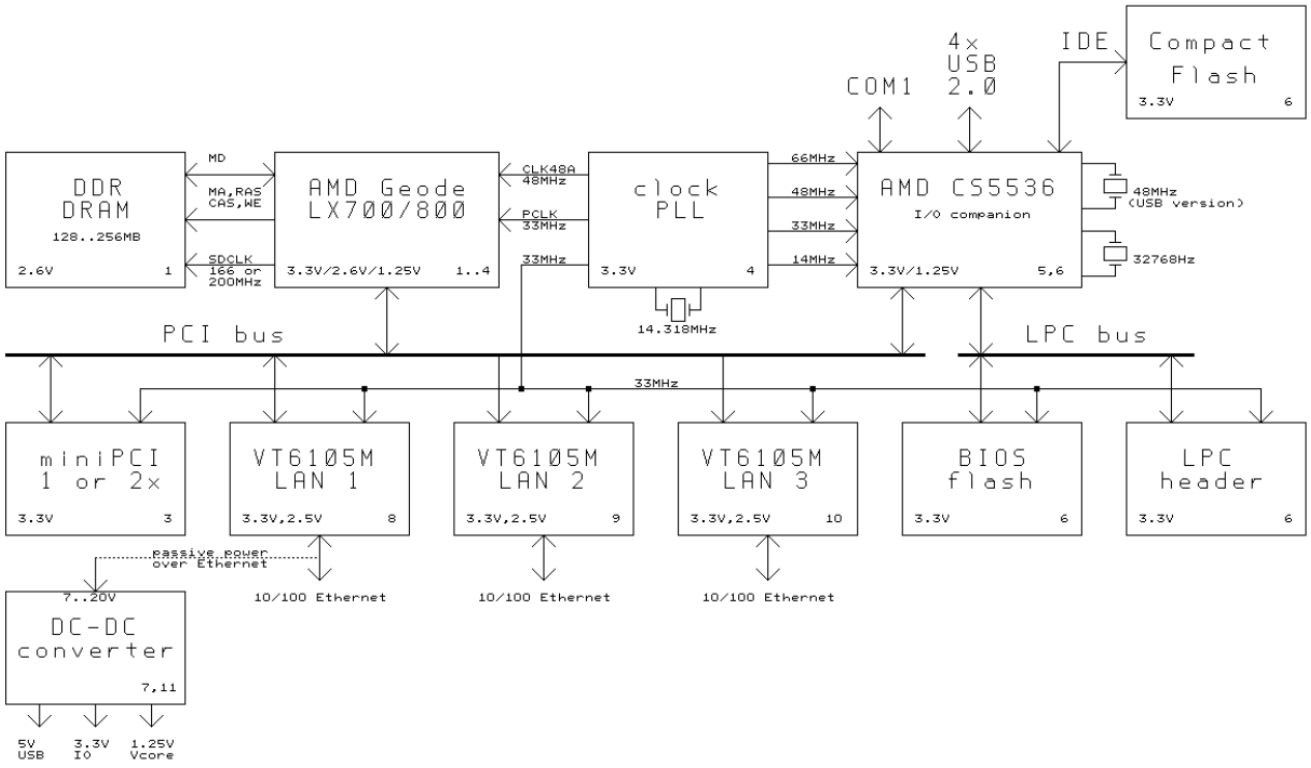

Abb. 2: Blockschaltbild Alix Board (Quelle: PC Engines)

Für das x86-System wurde ein Alix-Board (Alix 2d3, s. Abbildung 2) von PC-Engines eingesetzt, das ebenfalls über 256 MByte DRAM sowie PC-typische Interfaces und Hardware verfügt. Hier arbeitet ein AMD Geode LX800 mit 500 MHz.

Als Entwicklungsrechner zur Implementation der Software fungierte ein normaler PC unter CentOS. Für die FPGA-Systementwicklung kam ISE 10.1 zur Anwendung. Für die Toolchain mit Cross-Compiler, Root-File-System und Bibliotheken kam Buildroot zum Einsatz. Trotz der Vorteile der glibc-Bibliothek speziell auch für Echtzeitanwendungen wurde dennoch die wesentlich kleinere, aber für Embedded Systeme geeignetere uclic-Bibliothek verwendet. Näheres zu Installation, Patchen und Inbetriebnahme beider Systeme sowohl von Linux mit RT-Preempt Patch als auch von Linux unter Xenomai findet sich bei [Ahrend 2011].

Die Applikationsprogramme für die Tests und Messungen bestanden aus einem Kernel-Modul, das den Treiber für die spezifische Hardware (hier ein externer Interrupteingang sowie ein I/O-Port, s.u.) inklusive Interrupt Handler realisierte, und einem Benutzerprogramm im Userspace.

Anpassungen an Treibern und Anwendungsprogrammen

Steht ein Anwendungsprogramm unter Linux zur Verfügung, das mittels RT_Preempt echtzeitfähig gemacht werden soll, so sind nur wenige Änderungen notwendig. Der benötigte hochauflösende HRTIMER, der zeitliche Auflösungen bis in den ns-Bereich gegenüber der üblichen „Jiffy“-Auflösung im 10-ms-Bereich erlaubt, kann ja auch ohne RT-Preeempt Patch schon eingesetzt werden. Es empfiehlt sich, die Priorität des Userspace-Teils des Anwendungsprogramms auf die höchste verfügbare Priorität oder zumindest sehr hoch einzustellen. Im Programm selbst ist es sinnvoll, allen benötigten Speicher vorab zu allokieren und dann das Swapping des Speichers mittels mlockall() zu unterbinden. Selbstverständlich sollte auch das Scheduling-Verfahren auf das echtzeitfähige SCHED_FIFO sowie die Prozesspriorität gesetzt werden. Mithilfe der PID des Echtzeit-Threads kann die standardmäßige Priorität von 50 entsprechend erhöht werden.

Um eine Applikation unter Xenomai lauffähig zu machen, ist etwas mehr Aufwand nötig. Eine Reihe von nicht voll-echtzeitfähigen Funktionen wie z.B. request_irq() müssen durch ihr Äquivalent aus dem RTDM (Real Time Driver Model) von Xenomai ersetzt werden. Auch die Zuordnung zwischen Kernel-Modul und Userspace-Programm gestaltet sich etwas anders, nämlich durch Übergabe des Gerätenamens und nicht durch Zugriff auf die Gerätedatei selbst. Auch die Applikation wird mit Hilfe einer speziellen Funktion in eine Xenomai-Applikation umgewandelt. Auch das Compilieren benötigt zusätzliche Parameter, da auf die Xenomai-Bibliothek und die Standardbibliothek zugegriffen werden muss.

Stabilität und zum Echtzeitverhalten der Systeme

Bei der Programmierung von Echtzeitsystemen interessiert vorrangig, mit welcher Reaktionszeit des Systems zu rechnen ist, wann auf ein asynchrones externes Ereignis eine erste Reaktion erfolgt. Um dies in den beiden Testsystemen zu untersuchen, wurde im Interrupt ein I/O-Port gesetzt und im anschließend aufgerufenen User-Programm wieder rückgesetzt. Die Zeit vom Auslösen des Interrupts bis zum Setzen des Ports entspricht der sog. Interrupt-Latenz, während die Zeit vom Interrupt bis zum Rücksetzen des Ports der Task-Latenz entspricht. Die Zeiten wurden mit Hilfe eines Speicheroszilloskops bzw. eines Logikanalysators gemessen.

Ein weiterer Test kann mit dem Programm Cyclictest [RTWIKI2011b] durchgeführt werden. Dieser Test für RT-Preempt existiert erfreulicherweise auch für Xenomai. Cyclictest erzeugt periodisch Threads, bei denen die Latenz zwischen geplantem und tatsächlichem Aufruf des Threads gemessen wird. Cyclictest wurde mit verschiedenen Belastungen auf dem Alix-Board

betrieben (5%, 100% Last sowie Anstecken eines USB-Sticks). Während bei einem „normalen“ Linux die minimalen, durchschnittlichen und maximalen Zeiten bei 5% und 100% ganz gut „aussahen“, verursachte das Anstecken des USB-Sticks exorbitant hohe Latenzen (Faktor 1000). Mit aktiviertem RT-Preempt Patch lagen die Werte, auch mit dem USB-Stick, maximal bei 50 µs (durchschnittlich 27-28 µs). Der gleiche Test mit Xenomai lieferte durchschnittlich 11-12 µs und maximal 25 µs, zeigte also wie erwartet niedrigere Latenzen.

Die gleichen Messungen auf dem PowerPC-System ergaben erwartungsgemäß höhere Latenzen (aufgrund der niedrigeren Taktfrequenz und der geringeren Verarbeitungsleistung). Während unter einem „normalen“ Linux Latenzen bis etwa 500 µs auftraten und einige Events überhaupt verpasst wurden, stellten sich unter RT-Preempt maximale Zeiten von etwa 200 µs ein. Auch hier lagen die (maximalen) Messwerte mit Xenomai etwa bei der Hälfte der Werte für RT-Preempt, nämlich bei ca. 100-110 µs.

Im Interrupt-Test mit dem IO-Port auf dem FPGA-Board mit dem PowerPC zeigte sich interessanterweise, dass die Latenzen mit RT-Preempt höher waren als ohne. Dies ist auf die Umsetzung von Interrupts in entsprechende Threads zurückzuführen. Wichtig war dabei auch, der Task und der Interrupt Service Routine eine hohe Echtzeitpriorität zuzuweisen. Damit konnten maximale Werte von etwa 120 µs für die Interrupt-Latenz und etwa 330 µs für die Task-Latenz erreicht werden.

Das Konzept von Xenomai zeigte sich auch in diesem Fall erwartungsgemäß überlegen. Hier ergaben sich maximale Zeiten von 28 µs (Interrupt-Latenz) bzw. 156 µs (Task-Latenz) mit sehr geringen Jitter-Werten.

Die Stabilität der Systeme erwies sich nach anfänglichen Schwierigkeiten als hervorragend. Es gab keinerlei Abstürze oder ähnliches. Die Software von RT-Preempt und auch von Xenomai scheint vergleichsweise ausgereift zu sein. Wie das mit der Version 3 des RT-Preempt Patch unter Linux 3.x aussieht, ist noch zu überprüfen.

Auswirkungen der Echtzeitmöglichkeiten auf reale Applikation wie die Open-Source-Telefonanlage Asterisk konnten noch nicht getestet werden, da erst zu einem sehr späten Zeitpunkt Asterisk (in einem anderen Projektkontext) überhaupt lauffähig gemacht werden konnte. Eine andere Anwendung konnte aber eindeutig von den Echtzeiteigenschaften durch RT-Preempt profitieren. Ein Embedded Board empfing Netzwerkpakete auf seinen Ethernet-Schnittstellen und versah die Pakte mit einem (relativen) Zeitstempel, bevor die Pakete weitergesendet wurden. Mit Hilfe des RT-Preempt Patch konnten die Echtzeitvorgaben des Vorhabens mühelos eingehalten werden.

Literatur

Ahrend2011: Ahrend, B.: *Untersuchung verschiedener Echtzeitansätze für ein Embedded Linux*. Diplomarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandete Wissenschaften, 2011.

Bétizeau2010: Bétizeau, M.; Boucart, Ch.-E.; Lemetayer J.-M.; Pasquier K.: *Portage d'un noyau Temps Réel XENOMAI sur processeur softcore XILINX Microblaze*. http://uuu.enseirb.fr/~kadionik/se/projets_avances/0910/rapport_sujet1_E3_0910.pdf

Gleixner2011: Gleixner, Th.: *Announce 3.0-rc7-rt0 based RT patch*. http://lkml.org/lkml/2011/7/19/309

Opden2011: Opdenacker, M.; Petazzoni, T.; Chanteperdrix, G.: *Real-time in embedded Linux systems*. Free Electrons, 2011

http://free-electrons.com/doc/embedded_linux_realtime.pdf

Pengu2007: Pengutronix: *POSIX Echtzeit: Kernel 2.6 und der Preempt-RT*. http://www.lug-erding.de/vortrag/Linux_Echtzeit-02-Programmierung.pdf

RTWIKI2011a: rt.wiki.kernel.org: *RT_PREEMPT_HOWTO*. https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO *)

RTWIKI2011b: rt.wiki.kernel.org: *Cyclictest*. https://rt.wiki.kernel.org/index.php/Cyclictest *)

Siemek2010: Siemek, M.: *MicroBlaze RT*. http://eeek.borgchat.net/lists/linux-rt-users/msg05692.html

** (derzeit –10/2011– aufgrund von Umstrukturierungen auf kernel.org nicht erreichbar)*

Kontaktdaten

Ostfalia Hochschule für angewandte Wissenschaften
Fakultät Elektrotechnik
Prof. Dr.-Ing. Rainer Bermbach
Salzdahlumer Straße 46/48
38302 Wolfenbüttel
Telefon: +49 (0)5331 939 42620
E-Mail: r.bermbach@ostfalia.de
Internet: www.ostfalia.de/pws/bermbach