

Implementierungsansätze für ein FPGA basiertes Multiprozessor System

Marco Kirschke

INF-M1 – Anwendung 1 - Wintersemester 2009/2010

25. November 2009

Inhalt

- ▶ Motivation
- ▶ FPGA-based Embedded Systems
- ▶ Entwurfsübersicht zu MPSoC
 - ▶ Entwurfsprozess der Hardware Modellierung
 - ▶ Festlegung der Multiprozessor Architektur
- ▶ Multiprozessor System on Chip mit Altera Nios II
- ▶ Zusammenfassung

Motivation

Zielsetzung für den Masterstudiengang

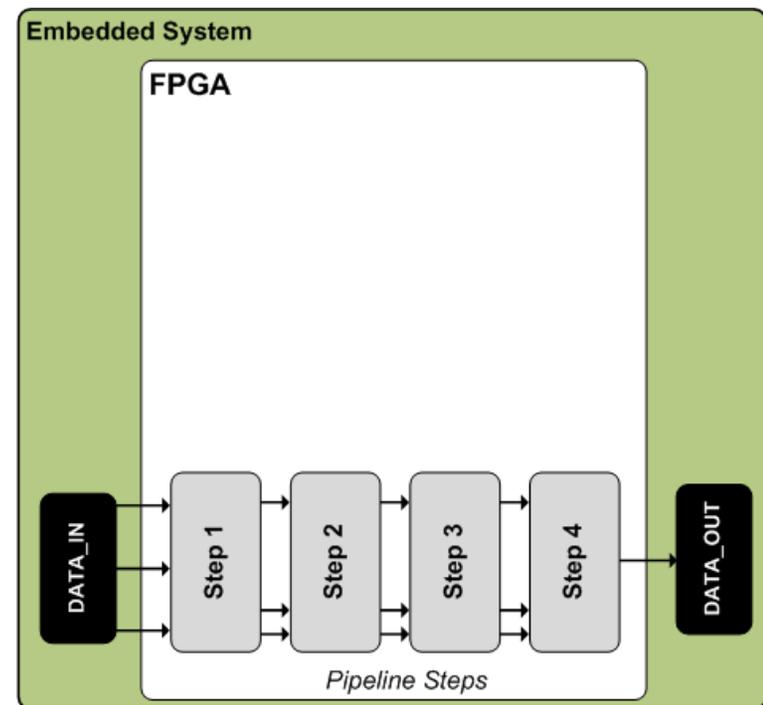
- ▶ Erfahrungen der Bachelor Arbeit ausbauen
- ▶ Masterstudiengang „Verteilte Systeme“
- ▶ Übersicht zu Techniken und Methoden von Multiprozessor Systemen
- ▶ Adaptierung von etablierten Vorgehensweisen auf FPGA basierte Systeme
- ▶ Technologie Grundlagen für ein späteres Arbeitsumfeld

Aufgabengebiete an der HAW

- ▶ SoC Fahrzeug
 - ▶ Multiprozessor System für verschiedene Anforderungen des Carolo Cups
 - ▶ 3D Audio
 - ▶ SoC mit einer Audioquelle ausgelastet
 - ▶ Multiprozessor System für
 - ▶ mehrere Audioobjekte
 - ▶ Klangobjekte im Raum platzieren (Positionsbestimmung)
 - ▶ Verwaltungsprozessor für Szenarien
 - ▶ Living Place Hamburg
 - ▶ Probleme bei der Vielzahl an Sensoren?
-

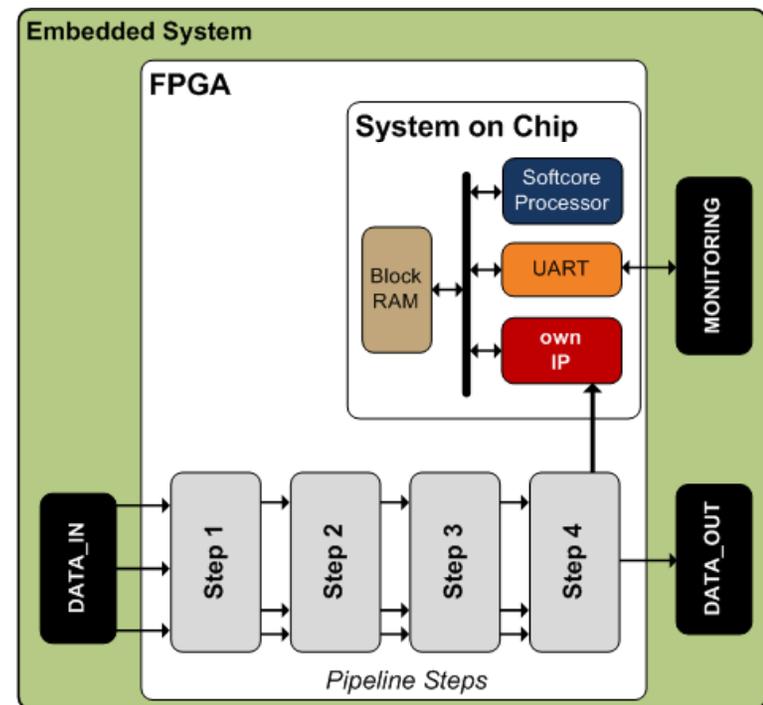
Zielsetzung

- ▶ Entwicklung von Embedded Systems
 - ▶ Einschränkende Randbedingungen
 - ▶ Flexible Plattformen
- ▶ Einsatz von FPGAs



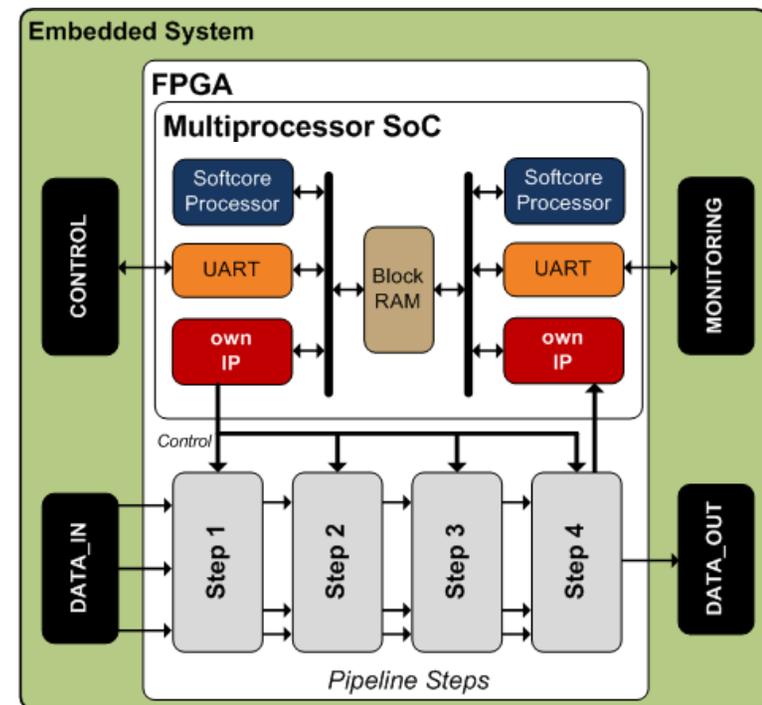
Zielsetzung

- ▶ Entwicklung von Embedded Systems
 - ▶ Einschränkende Randbedingungen
 - ▶ Flexible Plattformen
- ▶ Einsatz von FPGAs
- ▶ Optimierung des Entwicklungsprozesses



Zielsetzung

- ▶ Entwicklung von Embedded Systems
 - ▶ Einschränkende Randbedingungen
 - ▶ Flexible Plattformen
- ▶ Einsatz von FPGAs
- ▶ Optimierung des Entwicklungsprozesses
- ▶ Skalierung des Systems



FPGA-based Embedded Systems

Einsatzgebiete

Automotive



Wireline & Wireless Solutions



Data Storage & Processing



Aerospace & Military



Consumer Electronics



Anbieter



- ▶ Low Cost/Power FPGA
- ▶ XPGA (non-volatile, flash-based FPGA)



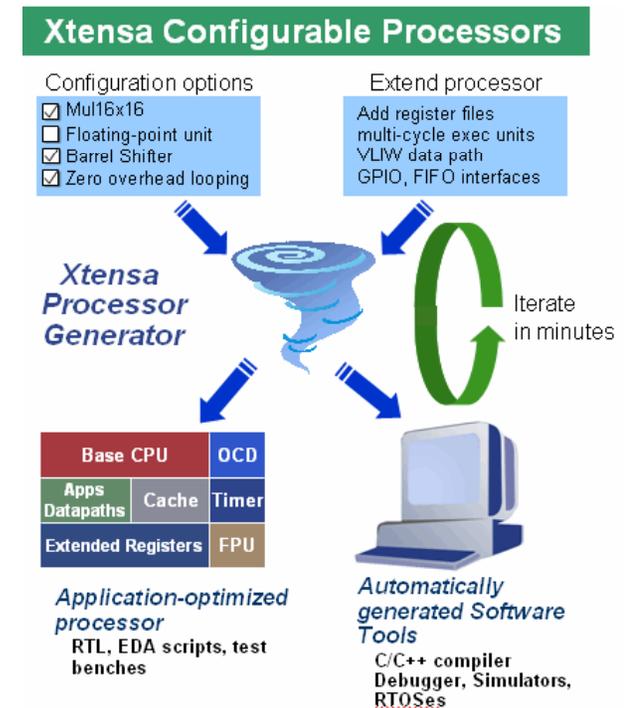
- ▶ Low Cost/Power FPGA
- ▶ Radiation-Tolerant FPGA
- ▶ ARM Prozessoren verfügbar

Xilinx und Altera

Produktpalette	Xilinx	Altera
<i>High Performance FPGA</i>		
	VIRTEX Family	Stratix Series
<i>Low Cost/Power FPGA</i>		
	SPARTAN Family	Cyclone Series
<i>Entwicklungsumgebung</i>		
	ISE Design Suite	Quartus
<i>System on Chip</i>		
<i>Hardware Software</i>	Xilinx Platform Studio/EDK Xilinx Platform Studio/SDK	SOPC Builder/Quartus NIOS IDE
<i>Soft Processor Cores</i>	PPC 440, PPC405, MicroBlaze	C68000, ARM Cortex, NIOS2

Tensilica

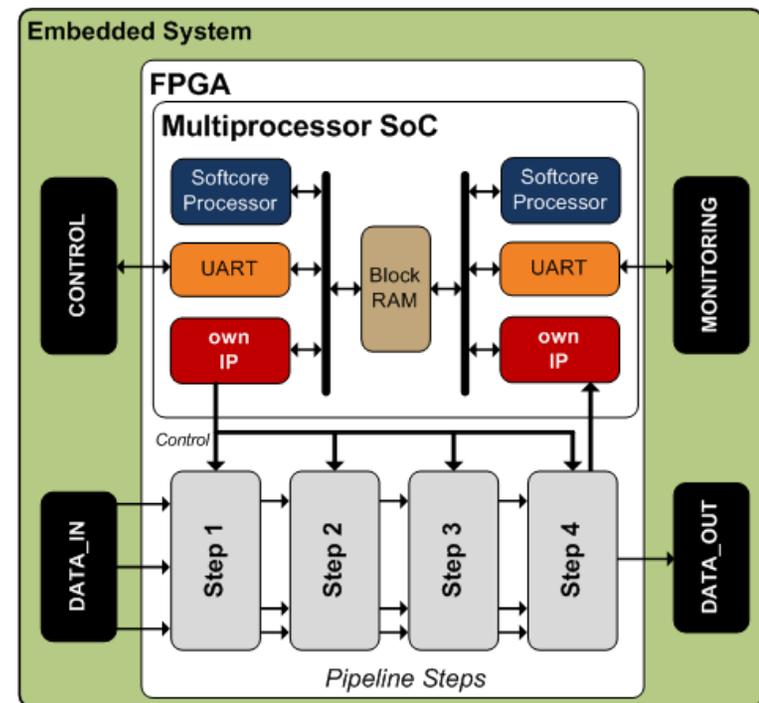
- ▶ Konfigurierbare Prozessoren
- ▶ Xtensa Xplorer IDE zur Konfiguration
- ▶ Kunden können sich auf die Software Entwicklung konzentrieren
- ▶ Bereitstellung von Anwendungen zur HW und SW Erstellung



Entwurfsübersicht Multiprocessor System on Chip

Multiprocessor System on Chip

- ▶ Erzeugung der Komponenten eines MPSoC
 - ▶ starke Abhängigkeit an die Anforderungen des Systems
- ▶ Entwurfsprozess der Hardware Modellierung
- ▶ Festlegung der MP Architektur
- ▶ Zusammenführung der Software und Hardware Komponenten



Entwurfsprozess der HW Modellierung

▶ Entwurfsmethoden

▶ Full-Custom-Entwurf → individueller Entwicklungsansatz

↑ Hohes Optimierungspotential ↓ erhöhter Aufwand

▶ Semi-Custom-Entwurf → Einsatz von Codegeneratoren

↑ verminderte Entwicklungszeit ↓ begrenzte Konfiguration

Entwurfsprozess der HW Modellierung

- ▶ **Vorgehensweise**
 - ▶ Spezifikation des Systems
 - ▶ Testen des Entwurfskonzeptes (*Simulation mit C oder MATLAB*)
 - ▶ Erzeugung der RTL Beschreibung in VHDL oder Verilog
 - ▶ Synthese (*Umsetzung der RTL Beschreibung auf Gatterebene*)
 - ▶ Simulation des Synthesergebnisses (*z.B. in ModelSim*)
 - ▶ Verteilung der Gatterebene auf die HW (*Fitter – Place&Route*)
 - ▶ Erzeugung des „Programming Files“

Festlegung der Multiprozessor Architektur

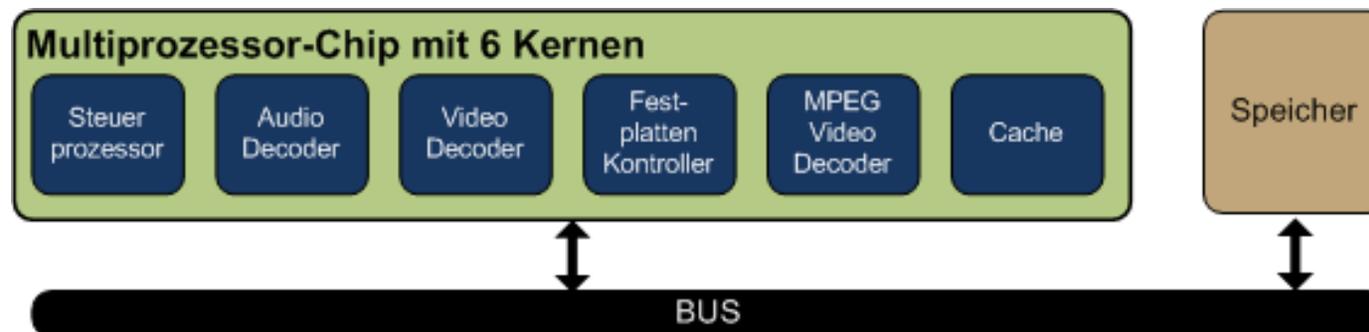
- ▶ Welche Aufgaben übernehmen die Prozessoren?
 - ▶ Wie sind die Prozessoren angeordnet?
 - ▶ Prozessorkoordination in HW oder SW?
 - ▶ Anpassung der HW an die Software des MPSoC?
- oder
- ▶ Anpassung der SW an die Hardware des MPSoC?

Wechsel zu Multiprozessor Systemen

- ▶ wachsende Anforderungen an Rechenleistung
- ▶ Abweichung von Erhöhung der Taktraten
- ▶ Senkung des Energiebedarfs und der Wärmeentwicklung
- ▶ Einsatz von Spezial- bzw. Coprozessoren
 - ▶ Netzwerkprozessoren
 - ▶ Medienprozessoren (Audio/Video)
 - ▶ Kryptographieprozessoren

Gliederung von MP Architekturen

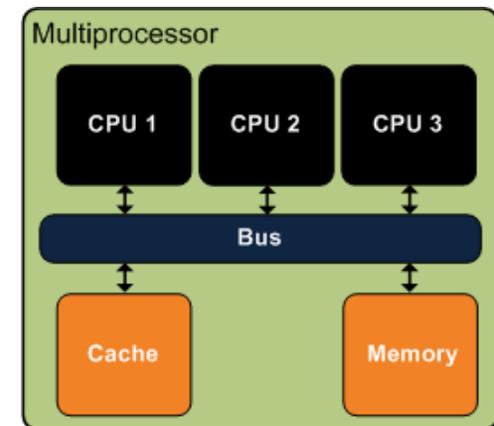
- ▶ Homogene Multiprozessoren
 - ▶ Primär zur Leistungssteigerung
 - ▶ Gemeinsamer Zugriff auf Systemressourcen
- ▶ Heterogene Multiprozessoren
 - ▶ Spezialprozessoren für verschiedene Anforderungen
 - ▶ Beispiel: DVD Player



Anordnung von Prozessoren in MPSoC

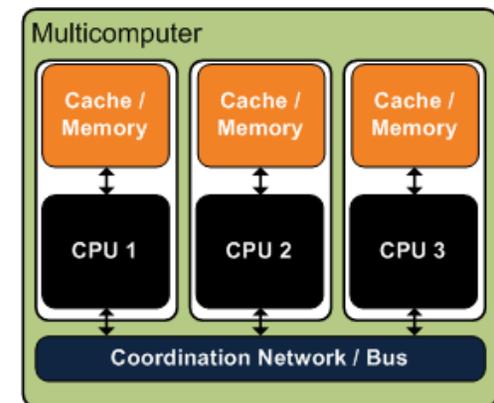
▶ Multiprozessor

- ▶ Mehrere CPUs nutzen einen gemeinsamen Speicher
- ▶ Synchronisation notwendig



▶ Multicomputer

- ▶ CPUs mit eigenem privaten Speicher
- ▶ Nachrichtenaustausch über ein Verbindungsnetzwerk
- ▶ Network on Chips



Multiprozessor

- ▶ Verwendung eines gemeinsamen Speichers
 - ▶ UMA (Uniform Memory Access)
 - ▶ Feste/Vorhersagbare Zugriffszeit auf Speicherreferenzen
 - ▶ NUMA (NonUniform Memory Access)
 - ▶ Speicherreferenzen mit unterschiedlichen Zugriffszeiten
 - ▶ COMA (Cache Only Memory Access)
 - ▶ Adressraum wird auf Cache-Zeilen aufgeteilt
- ▶ Auswirkung auf die Software Erstellung

Software für Multiprozessoren

- ▶ Vorteile für Software Entwickler
 - ▶ Load/Store Befehle für gemeinsam genutzte Speicher
- ▶ Festlegung eines Konsistenzmodells
 - ▶ Speicherzugriffe müssen koordiniert werden
 - ▶ Scheduling Verfahren beachten
- ▶ Parallelen zu Threadbasierter Programmierung
 - ▶ DSM - Distributed Shared Memory
 - ▶ Verwendung eines seitenbasierten virtuellen Adressraumes
 - ▶ MMU erzeugt Seitenfehler → Seiten werden nachgeladen
 - ▶ Anforderungen an das Betriebssystem

Multicomputer

- ▶ Prozessoren verwalten ihren privaten Speicher
 - ▶ Nachrichtenaustausch über Verbindungsnetze
 - ▶ Einsatz verschiedener Topologien
 - ▶ Stern, Baum, Ring, Gitter, Würfel, etc.
 - ▶ Routing der Nachrichten nötig
 - ▶ Aufwendigere Software Entwicklung
 - ▶ Verteilung der Informationen mit Send/Receive Methoden
 - ▶ Synchroner oder gepufferter Nachrichtenaustausch
 - ▶ MPI - Message Passing Interface
-

Auswahl eines Betriebssystems

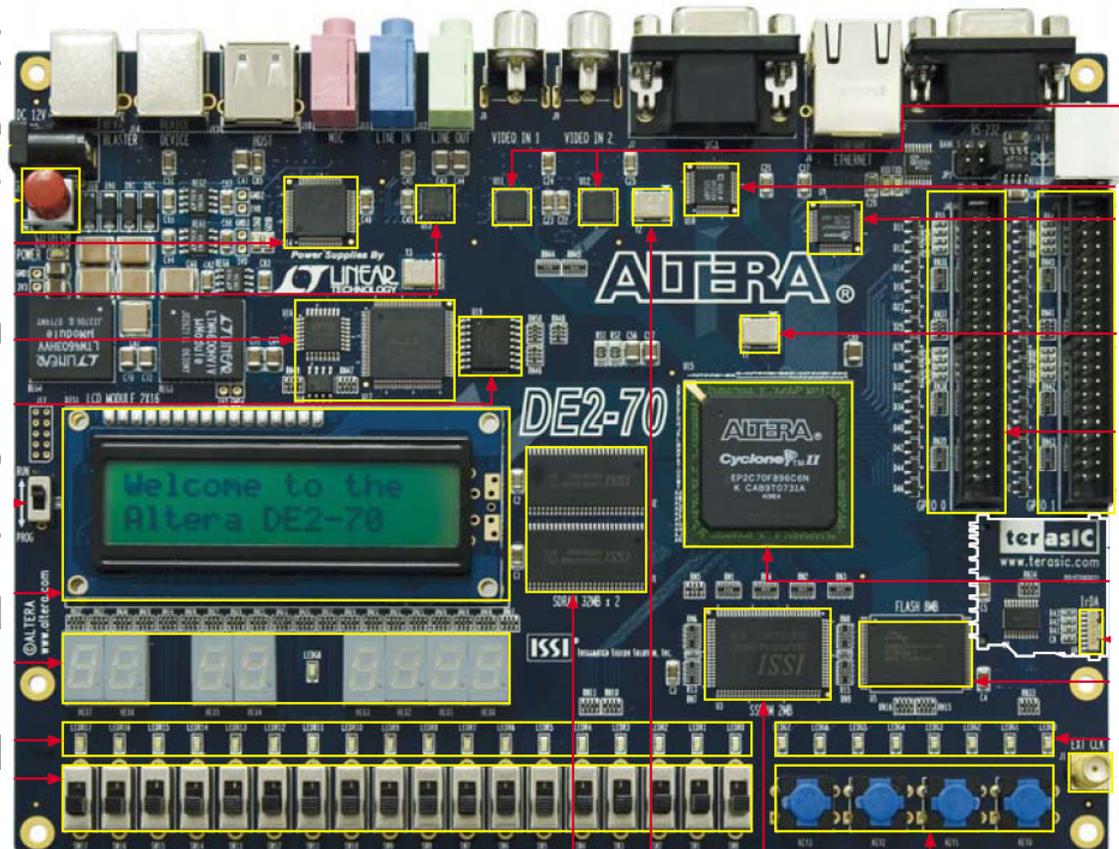
- ▶ **Auswahlkriterien für ein RTOS**
 - ▶ Architektur des MPSoC
 - ▶ Unterstützung von Standard Bibliotheken
 - ▶ Ressourcenverbrauch
 - ▶ Verfügbarkeit / Integration von Treibern für Peripherie
- ▶ **Xilinx und Altera FPGAs unterstützen**
 - ▶ μ C/OS II
 - ▶ BlueCat Embedded Linux / Wind River Linux
 - ▶ ThreadX

MPSoC mit 3 Nios II Prozessoren

terasic Development Board DE2-70

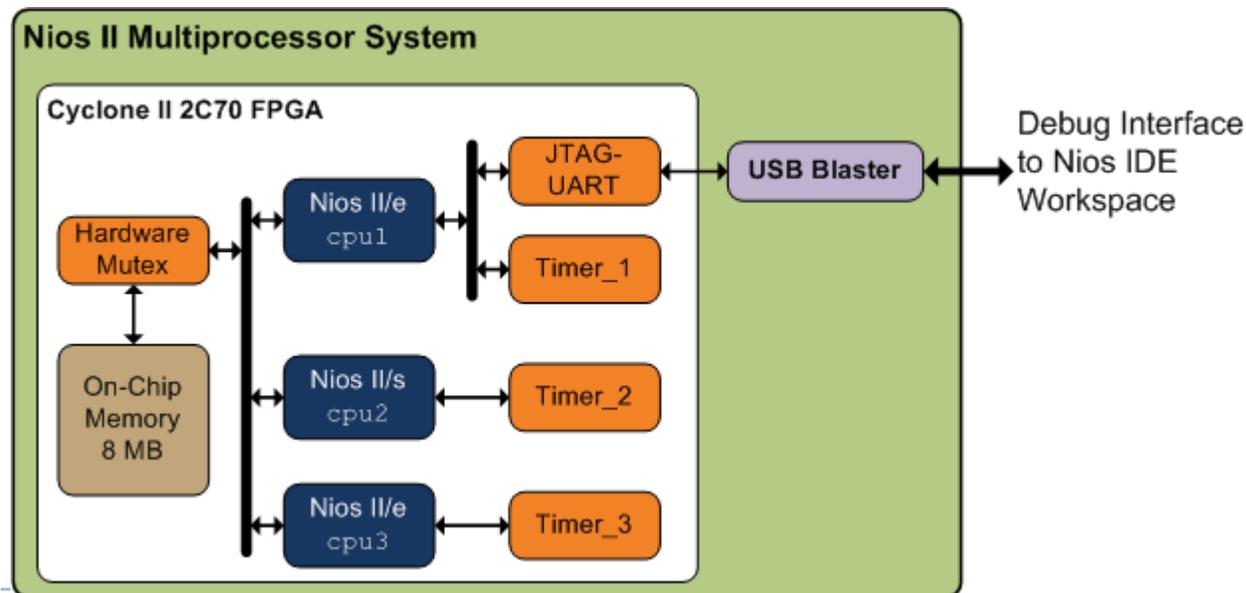
► Entwicklungsboard

- ▶ Altera Cyclone II 2C70
- ▶ Two 32-Mbyte SDRAM
- ▶ 8-Mbyte Flash Memory
- ▶ 50-MHz oscillator
- ▶ 24-bit CD-quality audio
- ▶ VGA DAC ADV7123
- ▶ 2 TV Decoder (NTSC/
- ▶ 10/100 Ethernet Contr
- ▶ USB Host/Slave Cont
- ▶ RS-232 transceiver
- ▶ PS/2 mouse/keyboard
- ▶ IrDA transceiver



Nios II Multiprocessor System

- ▶ „Hello World“ Beispiel mit 3 Nios II Prozessoren
- ▶ Zugriff auf On-Chip Memory über Hardware Mutex
- ▶ CPU1 behandelt Ausgabe über JTAG UART



SOPC Builder und Quartus II

The screenshot displays the Altera SOPC Builder interface for a Cyclone II device. The main window is titled "Altera SOPC Builder - DE2_70_SOPC.sopc*" and shows a "System Contents" view with a "Component Library" on the left and a "System Generation" view on the right. The "System Generation" view includes a "Target" section with "Device Family: Cyclone II" and a "Clock Settings" table.

Name	Source	MHz
clk_50	External	50,0
pll_c0_system	pll.c0	100,0
pll_c1_memory	pll.c1	100,0
pll_c2_audio	pll.c2	18,181818

Below the clock settings is a table listing system components:

Use	Conne...	Module Name	Description	Clock	Base	End	Tags	IRQ
<input checked="" type="checkbox"/>		cpu1	Nios II Processor	pll_c0_syst...				
		instruction_master	Avalon Memory Mapped Master				IRQ 0	IRQ 31
		data_master	Avalon Memory Mapped Master					
		jtag_debug_module	Avalon Memory Mapped Slave					
<input checked="" type="checkbox"/>		cpu1_timer	Interval Timer	pll_c0_syst...	0x09608800	0x09608fff		
<input checked="" type="checkbox"/>		cpu2	Nios II Processor	pll_c0_syst...	0x00000800	0x00000fff		
<input checked="" type="checkbox"/>		cpu2_timer	Interval Timer	pll_c0_syst...	0x00001000	0x0000101f		
<input checked="" type="checkbox"/>		cpu3	Nios II Processor	pll_c0_syst...	0x00001800	0x00001fff		
<input checked="" type="checkbox"/>		cpu3_timer	Interval Timer	pll_c0_syst...	0x00001020	0x0000103f		
<input checked="" type="checkbox"/>		timer_stamp	Interval Timer	pll_c0_syst...	0x09609440	0x0960945f		
<input checked="" type="checkbox"/>		pio_green_led	PIO (Parallel I/O)	pll_c0_syst...	0x096094e0	0x096094ef		
<input checked="" type="checkbox"/>		pio_red_led	PIO (Parallel I/O)	pll_c0_syst...	0x096094f0	0x096094ff		
<input checked="" type="checkbox"/>		pio_switch	PIO (Parallel I/O)	pll_c0_syst...	0x09609500	0x0960950f		
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART	pll_c0_syst...	0x096095a8	0x096095af		
<input checked="" type="checkbox"/>		uart	UART (RS-232 Serial Port)	pll_c0_syst...	0x09609460	0x0960947f		
<input checked="" type="checkbox"/>		pio_button	PIO (Parallel I/O)	pll_c0_syst...	0x09609510	0x0960951f		
<input checked="" type="checkbox"/>		lcd	Character LCD	pll_c0_syst...	0x09609520	0x0960952f		
<input checked="" type="checkbox"/>		sdram_u1	SDRAM Controller	pll_c0_syst...	0x04000000	0x05ffff		
<input checked="" type="checkbox"/>		sdram_u2	SDRAM Controller	pll_c0_syst...	0x06000000	0x07ffff		
<input checked="" type="checkbox"/>		ssram	Cypress CY7C1380C SSRAM	pll_c0_syst...	0x09200000	0x093ffff		
<input checked="" type="checkbox"/>		tristate_bridge_ssram	Avalon-MM Tristate Bridge	pll_c0_syst...				
<input checked="" type="checkbox"/>		cfi_flash	Flash Memory Interface (CFI)	pll_c0_syst...	0x08800000	0x08ffff		
<input checked="" type="checkbox"/>		tristate_bridge_flash	Avalon-MM Tristate Bridge	pll_c0_syst...				
<input checked="" type="checkbox"/>		sysid	System ID Peripheral	pll_c0_syst...	0x09609590	0x09609597		
<input checked="" type="checkbox"/>		pll	PLL	clk_50	0x09609480	0x0960949f		
<input checked="" type="checkbox"/>		SEG7	SEG7_IF	pll_c0_syst...	0x096094a0	0x096094bf		
<input checked="" type="checkbox"/>		message_buffer_mu...	Mutex	pll_c0_syst...	0x096095b8	0x096095bf		
<input checked="" type="checkbox"/>		message_buffer_ram	On-Chip Memory (RAM or ROM)	pll_c0_syst...	0x09609000	0x096093ff		

At the bottom of the window, there are informational messages:

- Info: pio_switch: PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simulation.
- Info: pio_button: PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simulation.

SOPC Builder und Quartus II

The screenshot displays the Quartus II software interface. The main window shows a 'Compilation Report - Flow Summary' for a project named 'DE2_70_NIOS_DEVICE_LED'. The report is organized into a tree view on the left and a detailed summary on the right.

Task List:

Task	Time
Compile Design	00:05:44
Analysis & Synthesis	00:02:11
Edit Settings	
View Report	
Analysis & Elaboration	
Partition Merge	
Netlist Viewers	
RTL Viewer	
State Machine Viewer	
Technology Map Viewer (Post-Mapping)	
Design Assistant (Post-Mapping)	
Edit Settings	
View Report	
I/O Assignment Analysis	
Early Timing Estimate	
Fitter (Place & Route)	00:03:09
Assembler (Generate programming files)	00:00:11
Classic Timing Analysis	00:00:13
EDA Netlist Writer	
Program Device (Open Programmer)	

Flow Summary:

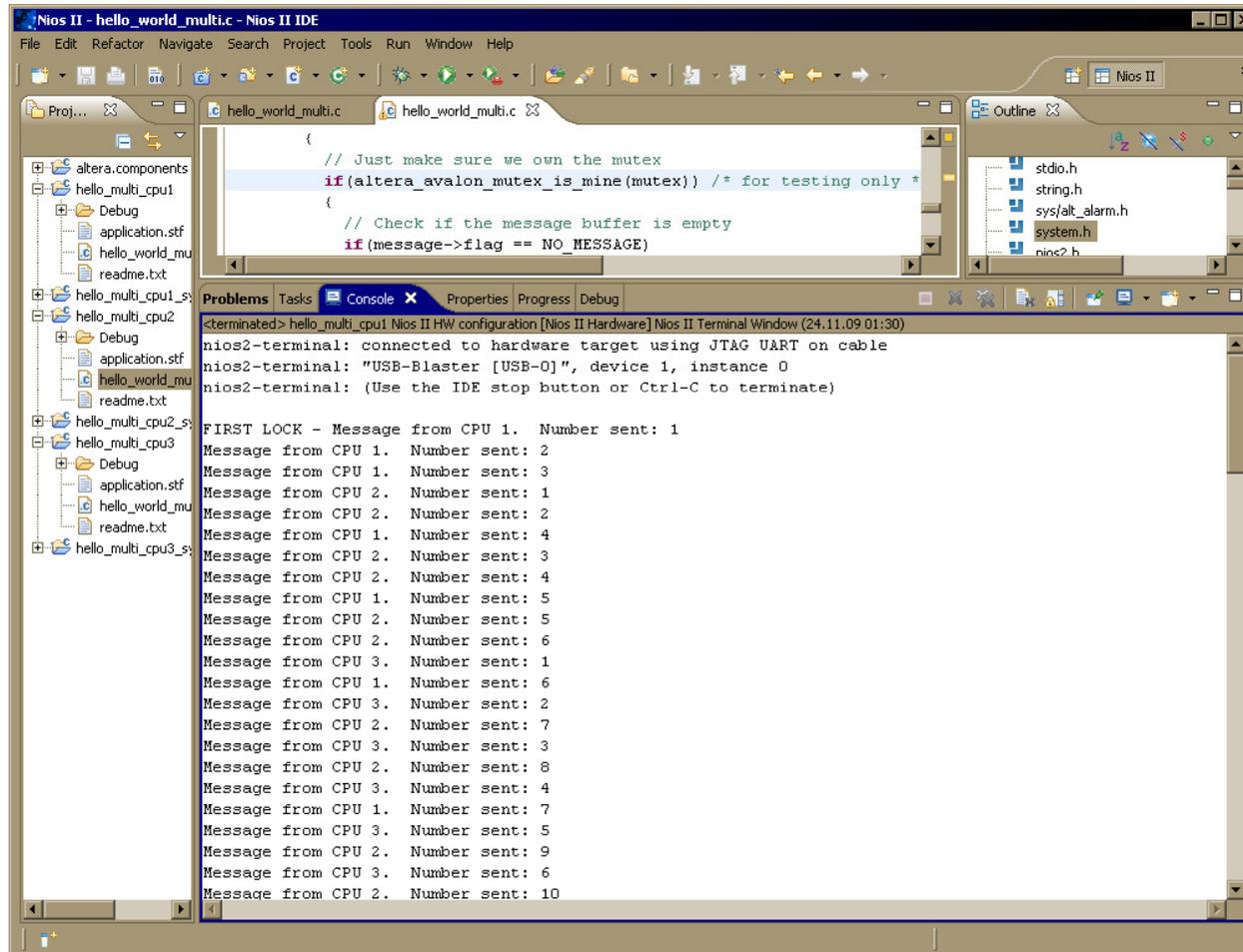
Flow Status	Successful - Tue Nov 24 00:49:12 2009
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	DE2_70_NIOS_DEVICE_LED
Top-level Entity Name	DE2_70_NIOS_DEVICE_LED
Family	Cyclone II
Device	EP2C70F896C6
Timing Models	Final
Met timing requirements	No
Total logic elements	8,916 / 68,416 (13 %)
Total combinational functions	7,984 / 68,416 (12 %)
Dedicated logic registers	4,238 / 68,416 (6 %)
Total registers	4500
Total pins	406 / 622 (65 %)
Total virtual pins	0
Total memory bits	73,728 / 1,152,000 (6 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	1 / 4 (25 %)

The bottom of the window shows a message log with the following text:

```

Info: Ended Full Compilation at Tue Nov 24 00:49:15 2009 W. Europe Standard Time
Info: SRAM Object File E:/Daten/Studium/DEV/tut_multiproc/DE2_70_NIOS_DEVICE_LED_time_limited.sof contains time-limited megafunction that sup
Info: SRAM Object File E:/Daten/Studium/DEV/tut_multiproc/DE2_70_NIOS_DEVICE_LED_time_limited.sof contains time-limited megafunction that sup
Info: Started Programmer operation at Tue Nov 24 00:50:05 2009
  
```


Nios IDE zur Softwareerstellung





Zusammenfassung

Zusammenfassung

- ▶ Implementierungsansätze für MPSoC
- ▶ High Performance Embedded Systems durch Verwendung von FPGAs
- ▶ Optimierung des Entwicklungsprozesses durch Integration von Software Komponenten
- ▶ Auslagerung der Anforderungen auf unterschiedliche Prozessoren

Vielen Dank für Ihre
Aufmerksamkeit.

Literatur

- [1] **Altera. 2007.** Creating Multiprocessor Nios II Systems *Altera:Tutorial*. [Online] 5. Dezember 2007. [November 2009.] www.altera.com/literature/tt/tt_nios2_multiprocessor_tutorial.pdf
- [2] **Wolf, W. 2007.** *High Performance Embedded Computing*. San Francisco, CA: Morgan Kaufmann, 2007. 0-12-369485-0
- [3] **Tanenbaum, A. . 2006.** *Computerarchitektur*. München: Pearson Studium, 2006. 3-8273-7151-1.
- [4] **Rauber, T. und Rüniger, G. 2008.** *Multicore: Parallele Programmierung*. Heidelberg: Springer, 2008. 3-540-73113-9.
- [5] **Jerraya, A. und Wolf, W. 2005.** *Multiprocessor Systems-on-Chips*. San Francisco, CA: Morgan Kaufmann, 2005. 0-12-385251-0
- [6] **lenne, P. und Leupers, R. 2007.** *Customizable Embedded Processors*. San Francisco, CA: Morgan Kaufmann, 2007. 0-12-369526-0
-