



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Seminar - SS 12

Erik Andresen

Modellierung einer Fahrspurführung für eine
ARM-MPSoC Plattform mit FPGA-Ressourcen

Inhaltsverzeichnis

1. Einleitung	3
2. Rückblick Projekt 1/2: Pandaboard - Plattform für eingebettete SW-Entwicklung	5
2.1. Vergleich mit Anwendung 2	6
3. ARM-MPSoC Plattform mit FPGA-Ressourcen	7
4. Portierung des HPEC-Fahrzeugs auf den Zynq-MPSoC	9
4.1. Risiken	11
5. Zusammenfassung	12
Literatur	13
A. Beschleunigungsarten über den FPGA im Zynq-MPSoC	15
Glossar	17

1. Einleitung

Im Projekt „High Performance Embedded Computing“ (HPEC) des Department Informatik an der HAW-Hamburg werden Algorithmen für Bildverarbeitung und Regelungstechnik auf eingebetteten Systemen ausgeführt. Als Anwendung dient ein autonom fahrendes Fahrzeug, das einer Fahrspur folgt und dabei Hindernissen ausweichen soll. Ausgeführt wird die Anwendung dabei auf FPGAs, also rekonfigurierbaren Schaltkreisen, die als System-On-Chip (SoC), Beschleuniger-Module und Mikroprozessoren auf einem Chip vereinen: Die Fahrspur der gefolgt werden soll wird mit einer Kamera aufgenommen und von der in der FPGA-Logik realisierten Bildverarbeitungskette identifiziert. Daraus wird die Stellgröße der Regelung für Antrieb und Lenkung errechnet. Als Mikroprozessor kommen in der FPGA-Logik implementierte MicroBlaze IP-Cores zum Einsatz [15][7], die mit einer maximalen Taktfrequenz von 200MHz betrieben werden können: Rechenintensive Algorithmen wie die Bildverarbeitung für die Erkennung der Fahrspur werden im FPGA durch Parallelverarbeitung beschleunigt ausgeführt.

Viele eingebettete Systeme arbeiten unter Batterieversorgung, weswegen der Strombedarf ein wichtiges Kriterium ist. Durch die niedrigere Taktfrequenz liefern Multiprozessorsysteme (MP) mehr Rechenleistung pro Watt als ein vergleichbares Einprozessorsystem. Wegen der zusätzlich geringeren Temperatur von Multiprozessorsystemen werden diese bei eingebetteten Systemen vermehrt eingesetzt [8][4]. Algorithmen mit hohem Rechen- und Arbeitsaufwand werden in eingebetteten Systemen in DSPs und ASICs realisiert, die zusammen mit dem Prozessor in ein System-on-Chip integriert sind [6][22]. Beispiele sind der Tablet-PC „Samsung Galaxy Tab 2“ und das Smartphone „Motorola Droid Bionic“, die beide als Multiprozessor-System-on-Chip (MPSoC), einen OMAP4430 System-On-Chip mit zwei Cortex-A9 ARM-Prozessoren einsetzen: Spielt der Benutzer eines solchen Gerätes ein mit H.264 komprimiertes Video ab, wird dieses Video von einem ASIC des MPSoC dekodiert.

Eine FPGA-MPSoC-Plattform mit zwei bis zu 800MHz getakteten Cortex-A9 Prozessoren im SMP Betrieb wurde von dem FPGA-Hersteller Xilinx mit der Bezeichnung „Zynq-7000“ angekündigt, die ab Sommer 2012 ausgeliefert werden soll [20]. Gegenüber dem Microblaze sind die ARM-Prozessoren nicht durch die Taktrate und Logikzellen des FPGA begrenzt, was die Flexibilität im Hardware/Software-Codesign erhöht. Dies erlaubt die Wiederverwendung von Intellectual Property (IP): Das komplette Hardware-Design für eine Produktreihe wird einmal festgelegt; Kundenspezifische Anforderungen werden in der Software implementiert. Zusätzlich werden die Entwicklungskosten durch die in der Softwareentwicklung kürzeren Entwicklungszeiten und bessere Wartbarkeit gesenkt [13].

Die Technik MPSoC, die im Gegensatz zu allen bisher im Studium[5] behandelten SoCs über mehr als einen Prozessor verfügt, wurde mit der Entwicklungsplattform Pandaboard [12] durchdrungen. Das Pandaboard verfügt mit dem OMAP4430 MPSoC wie der Zynq-7000 MP-SoC über zwei Cortex-A9 Prozessoren. Als Anwendung soll das autonome Fahrzeug unter dem OMAP4430 der Fahrspur folgen, dazu wurden im Rahmen von Projekt 1 und Projekt 2 Fahrzeug-Aktoren und Kamera an das Pandaboard angeschlossen. In weiteren Arbeiten des HPEC-Projektes werden zur Zeit Ausweichassistent und Fahrspurerkennung auf die Pandaboard-Plattform portiert.

Ziel der Masterarbeit ist die Portierung der bisherigen Architektur des autonomen Fahrzeugs auf den eingebetteten Zynq-7000 MPSoC. Dazu werden die eingesetzten Algorithmen zwischen Hard- und Software aufgeteilt: Entscheidungen mit *if*-Bedingungen, z.B. ob der Fahrspur gefolgt, oder ausgewichen werden muss (Abbildung 1) sind eine Aufgabe für die der schneller getaktete ARM-Prozessor besser geeignet ist.

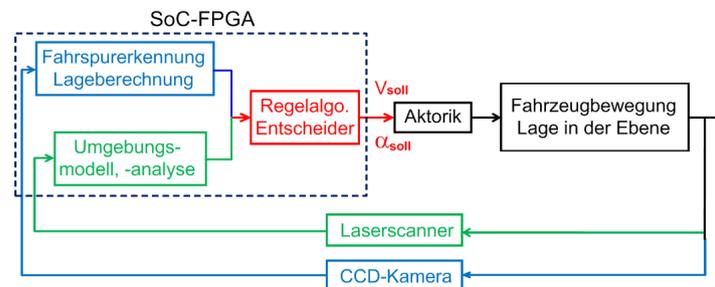


Abbildung 1: Erweiterung der Fahrspurenerkennung des autonomen Fahrzeugs um einen Ausweichassistenten. Je nach Auswertung der Sensoren durch den Entscheidungs-Algorithmus wird die Stellgröße für Antrieb und Lenkung von der Fahrspurenerkennung oder dem Ausweichassistenten übernommen.

Die geplante Aufteilung der Arbeiten zwischen den beiden Prozessoren im Master-Slave-Ansatz und dem FPGA zeigt Abbildung 2: Prozessor CPU-1 berechnet die Spurführung aus dem Ergebnis der Bildverarbeitung, die im FPGA durchgeführt wird, während Prozessor CPU-0 den Ablauf der Bildverarbeitung kontrolliert sowie Status- und Debug-Informationen an einen entfernten PC sendet.

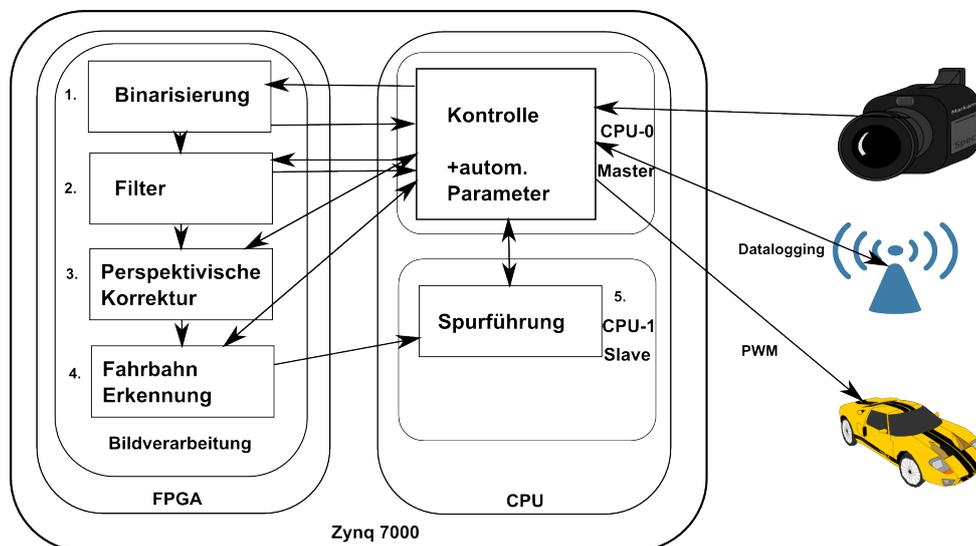


Abbildung 2: HW/SW Co-Design auf der Zynq-MPSoC Plattform: Der FPGA führt die Bildverarbeitung durch, die ARM-Prozessoren steuern die Bildverarbeitung und führen Rechnungen mit Gleitkommazahlen wie die Spurführung aus.

Aufgrund der umfassenden Treiberunterstützung bei der Verwendung von Betriebssystemen wie Linux auf dem Dual-Core Prozessor, kann das Fahrzeug zusätzlich mit neuen Funktionen, wie das Loggen von Betriebswerten über die Funktechniken W-Lan und Bluetooth ausgestattet werden. Eine weitere neue Funktion ist die automatische oder manuelle Einstellung der Bildverarbeitungs-Parameter.

Diese Ausarbeitung ist eine Beschreibung über Ziel und Inhalt der Masterarbeit: Kapitel 2 ist ein Rückblick der vergangenen Arbeiten in Projekt 1, Projekt 2 und der Literaturstudie in Anwendung 2. In Kapitel 3 wird die Architektur der Masterarbeit mit den dazugehörigen Komponenten, Produkten und Werkzeugen vorgestellt. Abschließend wird auf die Risiken eingegangen.

2. Rückblick Projekt 1/2: Pandaboard - Plattform für eingebettete SW-Entwicklung

Die Durchdringung und Aufbereitung einer MPSoC-Technologie für den Einsatz in eingebetteten Systemen wurde durch die Entwicklungsplattform Pandaboard (Abbildung 3) erarbeitet, die über folgende Eigenschaften verfügt:

- OMAP4430 SoC mit Dual-Core 1GHz ARM Cortex-A9 Prozessor und 1GB DDR Arbeitsspeicher.
- SD-Kartenleser zum laden des Betriebssystems.
- Debug-Schnittstellen JTAG und RS-232.
- Kommunikation: W-Lan, Bluetooth und 100MBit/s Netzwerk-Verbindung.
- Ausgabe von Videosignalen mit HDMI- und DVI-Anschluss.
- Kamera-Anschluss nach MIPI CSI-2 Standard.
- Peripherie: Zwei USB 2.0 Host-Anschlüsse sowie „Expansion Connector“ mit GPIOs, I2C und SPI.

Damit das Fahrzeug unter der Durchführung der Algorithmen auf dem Pandaboard autonom einer Fahrspur folgen kann, wurde in Projekt 1 die Ansteuerung der Aktoren und in Projekt 2 das Auslesen der Kamera untersucht. Als Betriebssystem wird auf dem Pandaboard ein von Linaro [9] erweitertes Ubuntu Linux oder Android eingesetzt, da diese die benötigten Treiber enthalten.

Die Ansteuerung der Aktoren mit dem Pandaboard wurde in Projekt 2 untersucht: Die Motoren des Fahrzeugs für Antrieb und Lenkung werden über eine Pulsweitenmodulation (PWM) mit einer Periodendauer von 20ms und einer Impulslänge von 1ms-2ms angesteuert. Der OMAP4430 System-on-Chip kann mit den Timer Pins GPTIMER8 bis GPTIMER11 eine PWM erzeugen; Diese Pins sind jedoch bereits vom USB-Controller belegt und stehen deswegen nicht zur Verfügung [18]. Eine unter dem Linux-Betriebssystem generierte PWM ist wegen des Jitter von 0,7ms nicht geeignet. Die PWM wird deswegen über einen 8-Bit AVR-Mikrocontroller erzeugt, der die Impulslänge für die PWM vom Pandaboard über eine I2C-Verbindung erhält. Für die Anpassung der unterschiedlichen Logikspannung zwischen beiden ICs wird ein Pegelwandler eingesetzt [1].

Die Auswahl einer Kamera zum Anschluss an das Pandaboard und das Auslesen der Bilder für die Fahrspurerkennung ist Inhalt von Projekt 2. Da das Pandaboard über keine parallele

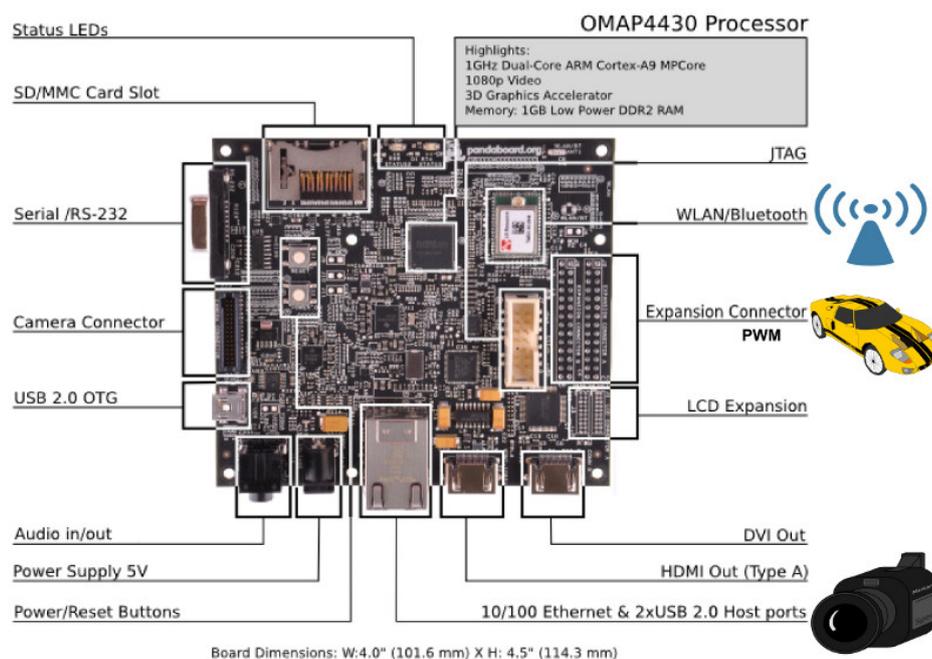


Abbildung 3: Pandaboard Entwicklungsplattform mit Dual-Core ARM Cortex-A9 Prozessor und Peripherie. Die Aktoren des Fahrzeug werden an den „Expansion Connector“ angeschlossen, für die USB-Kamera zur Fahrbahnerkennung gibt es zwei USB 2.0 Host-Ports. Statusmeldungen an die Basisstation lassen sich über die W-LAN und Bluetooth Verbindung senden. Bild: Pandaboard.org

Schnittstelle mit Line und Framevalid verfügt, muss eine andere Kamera, als die im bisherigen Fahrzeug eingesetzte, verwendet werden. USB-Webcams werden von den Android-Versionen standardmäßig nicht unterstützt [17], und für den CSI-2 Anschluss (vgl. Abb. 3) ist keine Hardware vorhanden. Bei dem CSI-2 Anschluss handelt es sich um einen von der Mobile Industry Processor Interface (MIPI) definierten Standard für Kamera-Anschlüsse. Deswegen wird eine IP-Kamera mit MP4-Datenstrom über UDP/RTP verwendet, die Einzelbilder werden von den Video-Beschleunigern des OMAP4430 ohne Verwendung von CPU-Rechenzeit dekodiert. Die Portierung der Fahrspurerkennung auf das Pandaboard unter Linux und Android wird in anderen Arbeiten des HPEC-Projektes durchgeführt.

2.1. Vergleich mit Anwendung 2

Die parallele Programmierung von eingebetteten MPSoC-Plattformen am Beispiel von Spurführungs-Algorithmen wurde in Anwendung 2 [2] untersucht. Spurführungs-Algorithmen wie Follow-the-carrot und Pure-Pursuit, wurden mit modernen Nachfolgern verglichen. Im aktuellen Fahrzeug wird für die Spurführung eine kombinierte Pure-Pursuit/PD-Regelung in der FPGA-Hardware berechnet [15]. Diese Rechnung wird in der Masterarbeit in den ARM-Prozessoren durchgeführt werden, da die ARM-Prozessoren die Berechnungen mit Gleitkommazahlen in einem Koprozessor durchführen.

Ebenfalls wurde eine Literaturstudie zu den Ansätzen für die Programmierung von Multiprozessor-Computern wie MPI und OpenMP durchgeführt. Dabei wurden auch das

Amdahlsche sowie das Gustafsonsche Gesetz ausgewertet, die Aussagen über die Skalierung von Multiprozessor-Computern treffen. OpenMP [11][10] ist ein Programmiermodell um ursprünglich sequentielle C, C++ und Fortran Programme zu parallelisieren [16]. Parallelisiert wird durch minimale Änderungen am sequentiellen Quellcode, damit zeichnet sich OpenMP durch einen hohen Abstraktionsgrad aus: Threads werden nicht vom Programmierer selbst erstellt und verwaltet, sondern es wird angegeben was parallelisiert werden soll. Welcher der Ansätze, ob MPI, OpenMP oder ein Master-Slave-Ansatz für die Multiprozessor-Programmierung in der Anwendung die effizientere ist, wird in der Masterarbeit geprüft werden.

3. ARM-MPSoC Plattform mit FPGA-Ressourcen

Ziel der Masterarbeit ist die Realisierung des autonomen Fahrzeugs mit der auf zwei physischen Cortex-A9 Prozessoren basierenden MPSoC Plattform Zynq-7000 (vgl. Abb. 4) [20] mit folgenden Eigenschaften:

- SMP fähiger Dual-Core Cortex A9 ARM Prozessor mit 32-Bit ARMv7 Befehlssatz
- Je Prozessor zwei 32kB Level-1 Caches, einen für Befehle und einen für Daten sowie eine MMU, Fließkommaeinheit und NEON SIMD-Koprozessor.
- Der 512kB Level-2 Cache wird von den Prozessoren geteilt, die Snoop Control Unit sorgt für die Synchronisation der Caches.
- Physische Peripherie: I2C, SPI, CAN, UART, GPIOs, Gigabit-Ethernet und USB 2.0 Hosts mit DMA, zusätzliche Peripherie kann im FPGA modelliert werden.
- Über die Advanced Mikrocontroller Bus Architecture (AMBA) sind die Prozessoren an die Peripherie und den FPGA angeschlossen.

Der MPSoC ist eine Alternative zu den bisherigen System-on-Chips bestehend aus FPGA und MicroBlaze und Zwei-Chip Designs mit FPGA und Prozessor auf einer Platine. Die mit bis zu 800MHz getakteten ARM-Prozessoren liefern eine höhere Datenverarbeitungsgeschwindigkeit als die mit max 200MHz getakteten MicroBlaze-Prozessoren, was die Flexibilität in der Aufteilung der Aufgaben zwischen Hard- und Software erhöht.

Im Unterschied zu den bisherigen FPGAs mit integriertem PowerPC-Prozessor, verfügt die Zynq-7000 Plattform über physische Peripherie-Elemente wie Bus-Systeme, USB 2.0 Host und Gigabit-Ethernet, die es erlauben, den Prozessor ohne den FPGA zu benutzen: z.B. kann ein Betriebssystem wie Linux, ohne die Verwendung von FPGA-Ressourcen verwendet werden.

Durch die Integration der beiden Komponenten auf einem Chip, sind die Leiterverbindungen gegenüber Zwei-Chip-Designs kürzer, und damit der Bus zwischen FPGA und Prozessor schneller. Gegenüber einer Zwei-Chip-Lösung wird außerdem weniger Strom verbraucht, sowie durch den Wegfall eines Bausteins, die Gesamtbetriebskosten in der Entwicklung gesenkt und Platz im Produkt gespart.

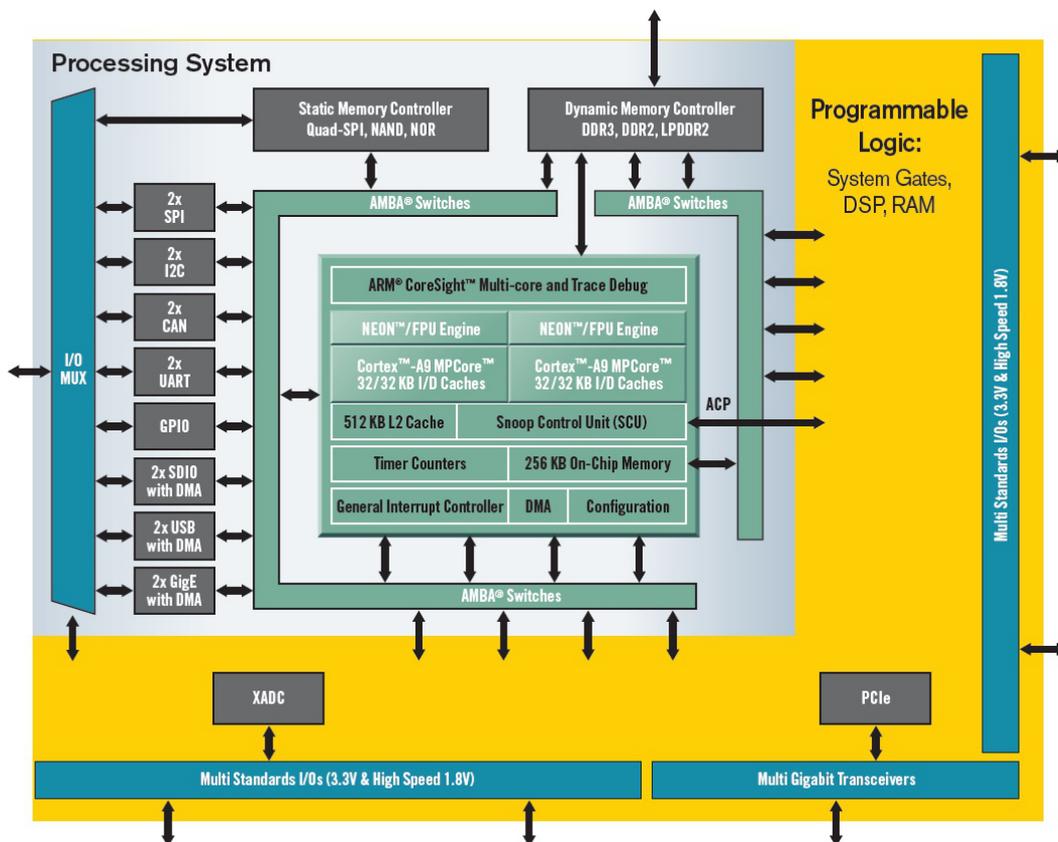


Abbildung 4: Die Zynq-7000 Architektur teilt sich in den Dual-Core Cortex A9 ARM Prozessor mit physischer Peripherie (Processing System) und den FPGA (Programmable Logic) auf. Die ARM Advanced Mikrocontroller Bus Architecture (AMBA) verbindet beide Komponenten miteinander. Über die ARM Advanced Mikrocontroller Bus Architecture (AMBA) ist der FPGA mit den Prozessoren verbunden und kann damit als Peripherie oder Koprozessor arbeiten. Bild: Xilinx.

Als Bussystem für die Kommunikation der Peripherie und des FPGA mit den Prozessoren wird in dem Zynq-MPSoC das standardmäßig in ARM basierten SoCs eingesetzte Advanced eXtensible Interface (AXI) verwendet, das Teil der ARM Advanced Mikrocontroller Bus Architecture (AMBA) ist [3]. Die AXI-Anbindung des FPGA an den Prozessor ist mit bis zu 9600MB/s angegeben [19].

Bei der ARM-Architektur kontrolliert die Snoop Control Unit (SCU) den Inhalt der Prozessor-Caches und sichert deren Konsistenz: Ist der Inhalt eines Caches veraltet, aktualisiert die SCU den Inhalt des Caches; Der Prozessor kann währenddessen andere Aufgaben erledigen bis die SCU per Interrupt mitteilt, daß der Cache aktualisiert wurde. Als Besonderheit bei der Zynq-Plattform dient der Accelerator Coherency Port (ACP) analog dazu, die CPU-Speicher mit dem FPGA-Speicher zu synchronisieren, und wird für den Datenaustausch eingesetzt wenn Algorithmen im FPGA beschleunigt werden.

Für die Softwareentwicklung werden die in der ARM-Welt etablierten Werkzeuge, wie die ARM Entwicklungsumgebung DS-5 (Abbildung 5), der GNU C-Compiler und die Lauterbach Entwicklungs- und Debugwerkzeuge eingesetzt.

Zu den bereits unterstützten High-Level-Betriebssystemen gehören Linux, Android und MS-Windows Embedded Compact 7, zu den unterstützten Echtzeitbetriebssystemen zählen

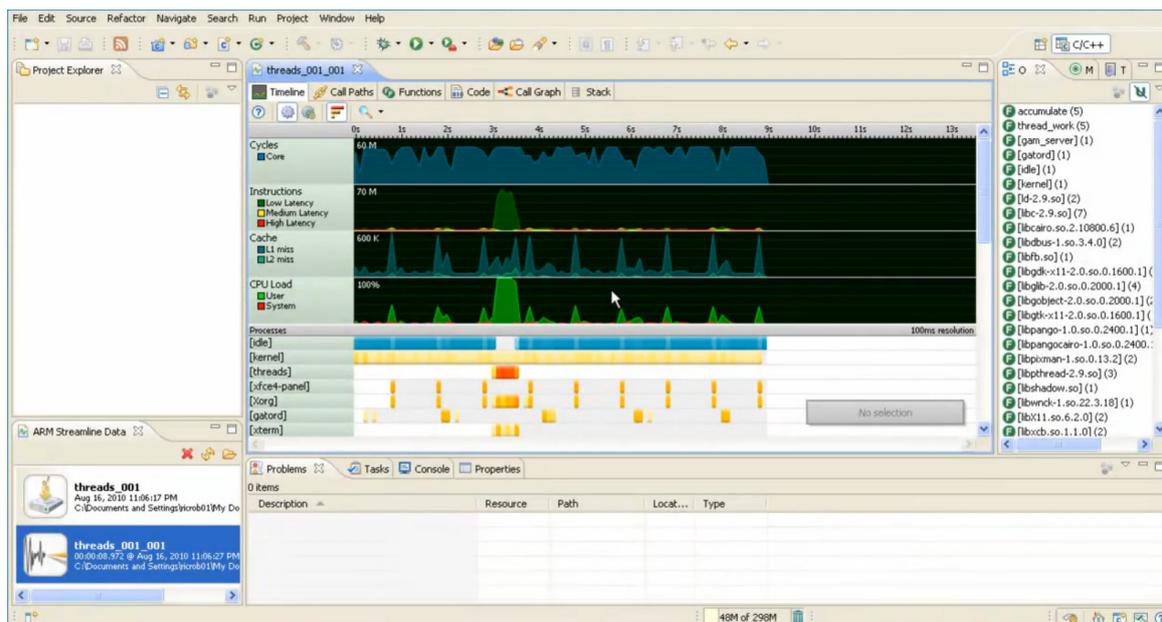


Abbildung 5: ARM DS5 Entwicklungsumgebung im Debugmodus. Zu sehen ist die Auslastung einzelner Prozesse unter Linux auf dem Prozessor. Bild: ARM

Micrium μ C/OS-III, FreeRTOS, QNX und T-Kernel. Dabei können die Betriebssysteme sowohl im SMP-Betrieb, als auch im AMP-Betrieb arbeiten. Im AMP-Betrieb können pro Prozessor unterschiedliche Betriebssysteme laufen, wie Linux auf einem Prozessor und QNX auf dem anderem.

Als Zynq-7000-Entwicklungsplattform wird das von den Firmen Digilent und Avnet entwickelte Zedboard [23] (Abbildung 6) eingesetzt. Es verfügt über:

- Zynq-7000 System-On-Chip, die ARM Prozessoren lassen sich mit bis zu 667MHz takten.
- 512MB DDR3 Arbeitsspeicher.
- SD-Kartenleser zum laden des Betriebssystems.
- Debug-Schnittstellen: Integrierten USB-JTAG, JTAG-Anschluss und RS-232.
- Kommunikation: Gigabit-Netzwerk Verbindung.
- Ausgabe von Videosignalen über einen HDMI-Anschluss.
- Peripherie: USB 2.0 Host sowie FMC- und Pmod-Anschlüsse um auf die FPGA-Pins zuzugreifen.

4. Portierung des HPEC-Fahrzeugs auf den Zynq-MPSoC

Ziel der Masterarbeit ist die Realisierung des Kamera-basierten Spurführungssystems mit dem Zynq-MPSoC in folgenden Schritten:

1. Auswahl eines Betriebssystems für die zwei ARM Cortex-A9 Prozessoren.

2. Synchronisationskonzepte für die Software-Threads untereinander und mit dem FPGA entwickeln.
3. Hardware/Software-Partitionierung
4. Microblaze PLB-Interface der Bildverarbeitungskette durch AXI Interface des Zynq-7000 ersetzen.
5. Spurführung in Software realisieren.
6. Neue Funktionen: Loggen von Betriebswerten und Echtzeitbildspeicherung, Ausweichassistent anbinden.

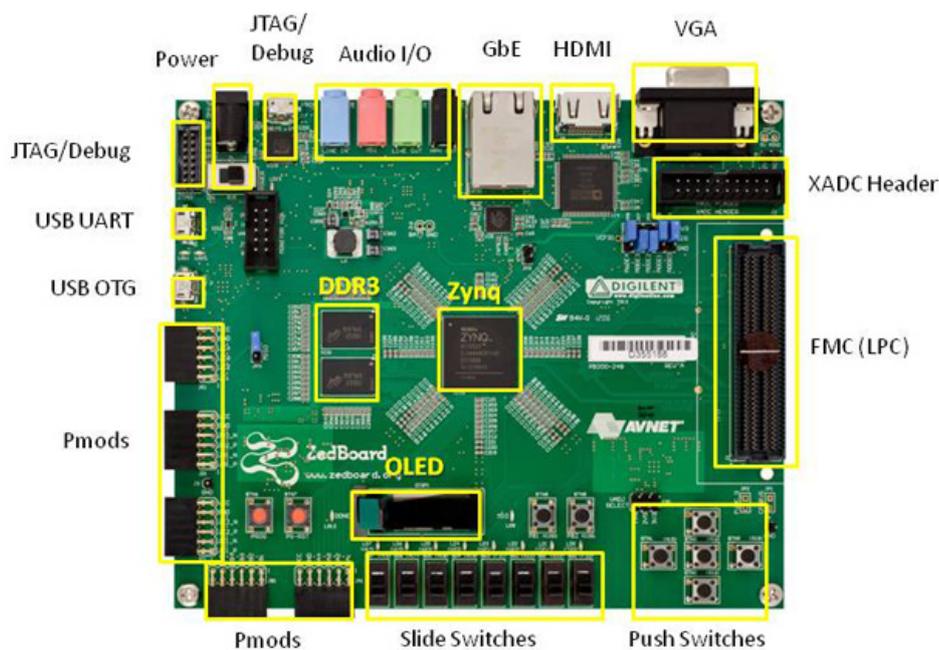


Abbildung 6: Zynq-7000 Entwicklungsplattform Zedboard: Über die Pmods und FMC-Anschlüsse lassen sich Kamera und Fahrzeug-Aktoren an die Pins des Zynq-SoC anschließen. Bild: Zedboard.org

Im aktuellen Fahrzeug-System findet die Spurführung ohne Beteiligung der MicroBlaze-Mikrocontroller in der FPGA-Logik statt [7], die MicroBlaze-Mikrocontroller übernehmen die Kommunikation und die Verarbeitung der Ausweichsensorik. Für das Senden von Debug-Informationen an einen PC ist die Bildverarbeitungskette über einen in der FPGA-Logik realisierten Processor Local Bus (PLB) an die MicroBlaze-Prozessoren angeschlossen.

In der Masterarbeit wird der Spurführungs-Algorithmus auf den ARM-Prozessoren in Software unter Verwendung der Fließkommaeinheit realisiert. Die Bildverarbeitung wird weiterhin in der FPGA-Logik durchgeführt, wird aber von dem bisherigen PLB im MicroBlaze-SoC an die AMBA des Zynq-MPSoC angepasst. Durch die Anbindung der Bildverarbeitungskette an die ARM-Prozessoren kann der Videostrom in allen Zwischenschritten, z.B. über die Gigabit-Netzwerk-Controller zur Analyse an einen PC gesendet werden (Abb. 7). Für die Anbindung der im FPGA synthetisierten Bildverarbeitungskette an die ARM-Prozessoren, stehen verschiedene Beschleunigungsarten zur Verfügung [19], die in den Abbildungen 9, 10, 11 und 12 im Anhang A erläutert sind. Für die Bildverarbeitungskette wird die dazu passende optimale Beschleunigungsvariante ausgewählt.

Als Betriebssystem ist ein Linux mit Echtzeit-Erweiterungen aufgrund der Unterstützung durch Treiber, wie Gigabit-Ethernet und SW-Bibliotheken (OpenMP, MPI) im SMP-Betrieb auf beiden Prozessoren vorgesehen. Für die Partitionierung der Aufgaben zwischen beiden Prozessoren unter dem SMP-Betriebssystem sind OpenMP oder MPI geeignet.

Eine andere Konzeptvariante ist, Linux auf einem Prozessor und ein Echtzeitbetriebssystem auf dem anderen Prozessor laufen zu lassen. Der Prozessor mit dem Echtzeitbetriebssystem übernimmt die zeitkritischen Kernaufgaben für die Fahrzeugsteuerung, wohingegen der zweite Prozessor Kommunikationsaufgaben, wie Loggen von Betriebswerten und das Senden von Videoströmen über Gigabit-Ethernet übernimmt.

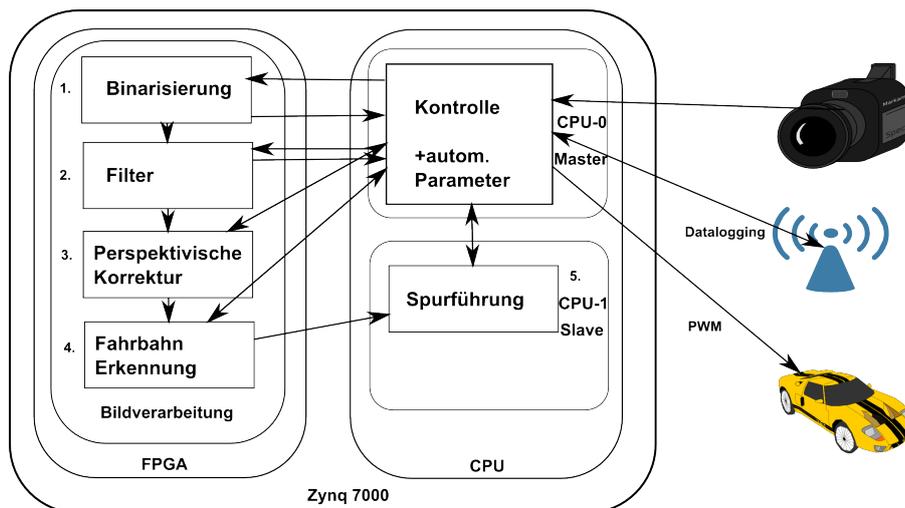


Abbildung 7: Partitionierung des Hardware/Software-Codesign für die eingebettete Plattform: Die Bildverarbeitungskette ist im FPGA realisiert, die ARM-Prozessoren steuern die Bildverarbeitung und führen Rechnungen mit Gleitkommazahlen wie die Spurführung aus.

Die einzelnen in der Masterarbeit durchzuführenden Schritte zeigt Abbildung 8.

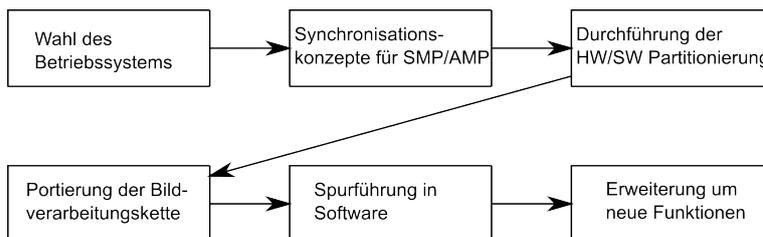


Abbildung 8: Arbeitsschritte in der Masterarbeit (Siehe Text)

4.1. Risiken

Nach mehrfachem Verzug [14] soll die Zynq-Plattform ab Sommer 2012 ausgeliefert werden. Die Auslieferung kann weiter verzögert werden, so daß die Zynq-Prozessoren nicht rechtzeitig zur Verfügung stehen würden. Für diesen Fall wird weiterhin mit dem Panda-board gearbeitet, oder alternativ kann der Dual-Core ARM im Emulator QEMU [21], oder die gesamte Zynq-Plattform in der Cadence Virtual Plattform getestet werden.

Als neues Produkt ist die Zynq-Plattform kaum getestet, Fehler in den Hard- und Software-Tools können vermehrt vorkommen. Da die in der Industrie bereits erprobten ARM-Prozessoren mit AXI-Bus verwendet werden, und mit den Standard ARM-Entwicklungswerkzeugen gearbeitet wird, sollte dieses Risiko geringer sein. Für die Migration von dem bisherigen PLB auf den neuen AXI-Bus, wird die komplette Bildverarbeitungskette angepasst, was einen hohen zeitlichen Aufwand bedeutet und damit ein Risiko darstellt. Ebenso muss für jede Aufgabe die richtige Beschleunigungsart evaluiert werden.

Der Speichercontroller im Zynq-SoC wird von dem FPGA und dem Prozessor gemeinsam genutzt, was sich bei Auslastung als Engpass herausstellen könnte: Sowohl der FPGA für die Verarbeitung des Videostroms, als auch der Prozessor für Programme und Daten, greifen über den Speichercontroller auf die Speicher zu.

5. Zusammenfassung

Im HPEC-Projekt werden Algorithmen für Bildverarbeitung und Regelungstechnik auf eingebetteten Systemen ausgeführt, als Anwendung dient ein autonom fahrendes Fahrzeug, das einer Fahrspur folgt und dabei Hindernissen ausweichen soll. Die Erkennung der von einer Kamera aufgenommenen Fahrspur, wird im FPGA durch Parallelverarbeitung, beschleunigt berechnet.

Die kombinierte ARM-FPGA-Plattform Zynq-7000 wird von Xilinx im Sommer 2012 eingeführt: Durch die gegenüber dem MicroBlaze schnelleren ARM Prozessoren soll die Flexibilität im Hardware/Software-Codesign erhöht werden, was die Entwicklungskosten senkt und die Wartbarkeit des eingebetteten Produktes erhöht. Deswegen wird in der Masterarbeit das bestehende HPEC-Fahrzeug mit der Zynq-Plattform realisiert.

Für die Durchdringung und Aufbereitung der MPSoC-Technologie, wurden Arbeiten auf der Pandaboard OMAP4430-Plattform durchgeführt, die ebenfalls wie der Zynq-MPSoC über einen Dual-Core Cortex-A9 verfügt. Diese Arbeiten werden zur Zeit in Abschlussarbeiten des HPEC-Projekt fortgeführt.

Vor der Realisierung der Algorithmen auf dem Zynq-MPSoC werden Überlegungen für die Hardware/Software Partitionierung durchgeführt: Algorithmen bestehend aus der Berechnung von Gleitkommazahlen, werden in der Fließkommaeinheit des Prozessors und Algorithmen mit hohem Datendurchsatz, wie die Bildverarbeitung im FPGA berechnet. Sequentielle Aufgaben werden in Software gelöst.

Es wird eine Software-Architektur entworfen, bei der die beiden Prozessoren bestmöglich genutzt werden. Dazu stehen SMP-Lösungen mit z.B. OpenMP und MPI oder eine AMP-Lösung mit zwei unterschiedlichen Betriebssystemen zur Auswahl. Die Bildverarbeitung wird von dem Microblaze-SoC auf den ARM-MPSoC portiert. Für die Plattform spezifischen Anpassungen stehen mehrere Beschleunigungsarten zur Auswahl (Anhang A).

Danach können neue Funktionen wie das Loggen von Informationen über eine Funkverbindung und der Ausweichassistent in Software implementiert werden. Zum Schluss wird in der Masterarbeit eine Gegenüberstellung mit der MicroBlaze-Lösung durchgeführt.

Literatur

- [1] ANDRESEN, Erik: Inbetriebnahme einer ARM Cortex-A9 Dualcore-Plattform und Erzeugung von PWM-Signalen für die Ansteuerung von Aktoren / HAW Hamburg, Department Informatik. 2012. – Master-Ausarbeitung Projekt 1
- [2] ANDRESEN, Erik: Parallele Programmierung von Eingebetten SMP-Plattformen am Beispiel von Spurführungs-Algorithmen / HAW Hamburg, Department Informatik. 2012. – Master-Ausarbeitung AW2
- [3] ARM: *AMBA Open Specifications*. – URL <http://www.arm.com/products/system-ip/amba/amba-open-specifications.php>
- [4] CROWLEY, Patrick: The future in your pocket. In: *SIGCOMM Comput. Commun. Rev.* 38 (2008), März, Nr. 2, S. 61–64. – URL <http://doi.acm.org/10.1145/1355734.1355744>. – ISSN 0146-4833
- [5] HAW HAMBURG, Department I.: *Studiengang technische Informatik*. – URL http://www.informatik.haw-hamburg.de/technische_informatik.html
- [6] KHADDOUR, M. ; WANG, Z. ; HAMMAMI, O.: Implementing block cipher on embedded multiprocessors platform. In: *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on*, april 2009, S. 193 –198
- [7] KIRSCHKE, Marco: *FPGA-basierte MPSoC-Plattform zur Integration eines Antikollisions-systems in die Fahrspurführung eines autonomen Fahrzeugs*, HAW Hamburg, Department Informatik, Diplomarbeit, 2012
- [8] LIN, Tzong-Yen ; HUNG, Yu-Ting ; CHANG, Rong-Guey: Efficient Hardware/Software Partitioning Approach for Embedded Multiprocessor Systems. In: *VLSI Design, Automation and Test, 2006 International Symposium on*, april 2006, S. 1 –4
- [9] LINARO: *Webauftritt der Linaro Organisation*. – URL <http://www.linaro.org>
- [10] OPENMP: *OpenMP - Community Seite*. – URL <http://www.compunity.org>
- [11] OPENMP: *OpenMP Webauftritt*. – URL <http://www.openmp.org>
- [12] PANDABOARD.ORG: *PandaBoard*. – URL <http://pandaboard.org>
- [13] RIEMENSCHNEIDER, Frank: *(Fast) eineiige Zwillinge - Alteras Antwort auf Zynq*. – URL http://www.elektroniknet.de/bauelemente/technik-know-how/halbleiter/article/88868/0/Fast_eineiige_Zwillinge_-_Alteras_Antwort_auf_Zynq/
- [14] SANTARINI, Mike: Xilinx Architects ARM-Based Processor-First, Processor-Centric Device. In: *Xcell Journal* 71 (2010). – URL <http://www.xilinx.com/publications/xcellonline/>
- [15] SCHNEIDER, Christian: *Ein SoC-basiertes Fahrspurführungssystem*, HAW Hamburg, Department Informatik, Diplomarbeit, 2011
- [16] SIMON HOFFMANN, Rainer L.: *OpenMP Eine Einführung in die parallele Programmierung mit C/C++*. Springer, 2008. – ISBN 978-3-540-73122-1

-
- [17] TEXAS INSTRUMENTS: *Building Android L27 for PandaBoard, with USB camera support.* – URL http://omappedia.org/wiki/Building_L27.IS.2.M1_for_PandaBoard,_with_USB_camera_support
- [18] TEXAS INSTRUMENTS: *OMAP4430 Technical Reference Manual.* – URL http://focus.ti.com/pdfs/wtbu/OMAP4430_ES2.x_PUBLIC_TRM_v0.zip
- [19] XILINX: *OMAP4430 Technical Reference Manual.* – URL http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf
- [20] XILINX: *Zynq-7000 Extensible Processing Platform.* – URL <http://www.xilinx.com/Zynq>
- [21] XILINX: *Zynq 7000 QEMU Modell.* – URL <http://wiki.xilinx.com/zynq-qemu>
- [22] YUE-LI, Hu ; QIAN, Ding: Design of an architecture for multiprocessor system-on-chip (MPSoC). In: *High Density Microsystem Design and Packaging and Component Failure Analysis, 2006. HDP'06. Conference on*, june 2006, S. 63 –66
- [23] ZEDBOARD.ORG: *ZedBoard: Zynq Evaluation & Development Board.* – URL <http://www.zedboard.org>

A. Beschleunigungsarten über den FPGA im Zynq-MPSoC

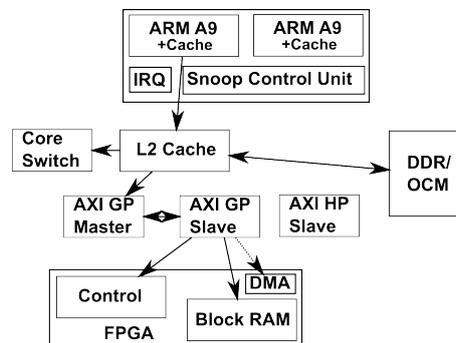


Abbildung 9: Beschleunigung mit kopieren der Daten durch CPU oder DMA-Controller: Die Daten werden von der CPU zwischen dem FPGA-Block-RAM und dem CPU-Speicher kopiert. Da hier keine Ereignisse des DMA-Controllers behandelt werden und in der FPGA-Logik nur die Anbindung an den AXI-Slave implementiert wird ist diese Art der Beschleunigung einfach zu verwenden. Für den Kopiervorgang wird CPU-Zeit benötigt, der DMA-Controller kann wahlweise mit eingebunden werden. Transferraten sind auf maximal 25MB/s begrenzt.

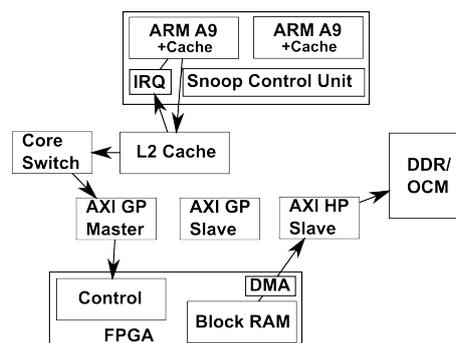


Abbildung 10: Beschleunigung mit Steuer- und Datenpfad: Der FPGA-Block-RAM wird direkt über das AXI High-Performance (HP) Interface mit dem CPU-Speicher verbunden. Die daraus resultierende Übertragungsrates von 1,2GB/s eignet sich z.B. für Videoströme. Der General-Purpose Master wird für den Kontrollfluss benutzt, der CPU kann per Interrupt mitgeteilt werden wenn Daten bereit stehen. Nachteil ist die Belegung der FPGA-Ressourcen und die höhere Latenz der AXI-HP Anbindung.

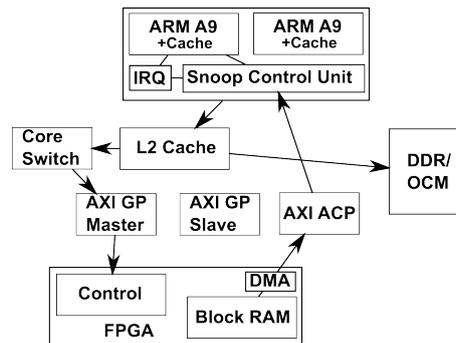


Abbildung 11: Beschleunigung mit dem Accelerator Coherency Port (ACP): Im Unterschied zu der vorherigen Variante, ist der FPGA-Block-Ram über den ACP mit der Snoop Control Unit, und damit mit den Level 1 und Level 2 Caches der Prozessoren verbunden. Da die Daten bereits in dem CPU-Cache vorliegen, kann die CPU mit einer geringeren Latenz auf die Daten zugreifen. Bei gleicher Übertragungsrage, wie über das AXI HP-Interface belegt die AXI-ACP Anbindung mehr FPGA-Ressourcen. Durch den direkten Zugriff auf den CPU-Cache, kann die CPU ausgebremst werden da weniger Cache für Programme zur Verfügung stehen. Durch die geringe Cachegröße ist diese Variante für kleinere Datenmengen geeignet.

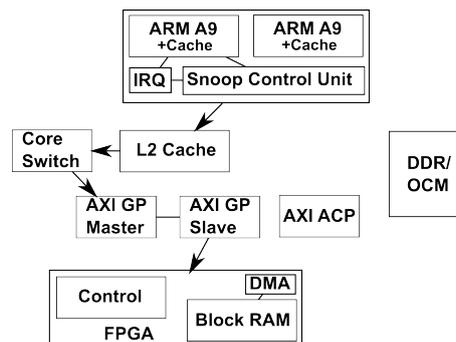


Abbildung 12: Beschleunigung mit FPGA als Koprozessor: Über das General Purpose-Interface hat der FPGA direkten 32-Bit DMA Zugriff auf die Register des Prozessors, der FPGA kann damit als Co-Prozessor der CPU arbeiten. Die CPU kann also Daten in einem Register ablegen, auf die der FPGA eine Operation wie das Ändern der Byte-Reihenfolge anwendet. Nach Abschluss der Operation kann die CPU über einen Interrupt benachrichtigt werden. Eignet sich für max. 32 Bit Daten, hat aber eine geringe Latenz.

Glossar

AMBA Advanced Mikrocontroller Bus Architecture

AMP Asymmetric multiprocessing

ARM 32-Bit RISC Prozessor Architektur

ASIC Application specific integrated circuit

AXI Advanced eXtensible Interface

DSP Digitaler Signalprozessor

FPGA Field Programmable Gate Array

HPEC High Performance Embedded Computing

IC Integrierter Schaltkreis

IP Intellectual Property

Linaro Eine 2010 von den Firmen ARM, Freescale, IBM, Samsung, ST-Ericsson und Texas Instruments gegründete gemeinnützige Organisation, mit dem Zweck die Entwicklung von Open-Source-Software für die ARM-Plattform voranzubringen.

MIPI Mobile Industry Processor Interface, eine 2003 von den Firmen ARM, Intel, Nokia, Samsung, STMicroelectronics und Texas Instruments gegründete Organisation um die Schnittstellen in mobilen Geräten zu vereinheitlichen.

MMU Memory Management Unit

MP Multiprozessor

MPSoC Multiprozessor-System-on-Chip

PLB Processor Local Bus

SCU Snoop Control Unit

SIMD Single Instruction, Multiple Data

SMP Symmetric multiprocessing

SoC System-on-Chip

Xilinx Ein führender Hersteller von FPGAs.