

Betriebssystem-Erwägungen bei Funk-Applikationen von Zynq SoCs

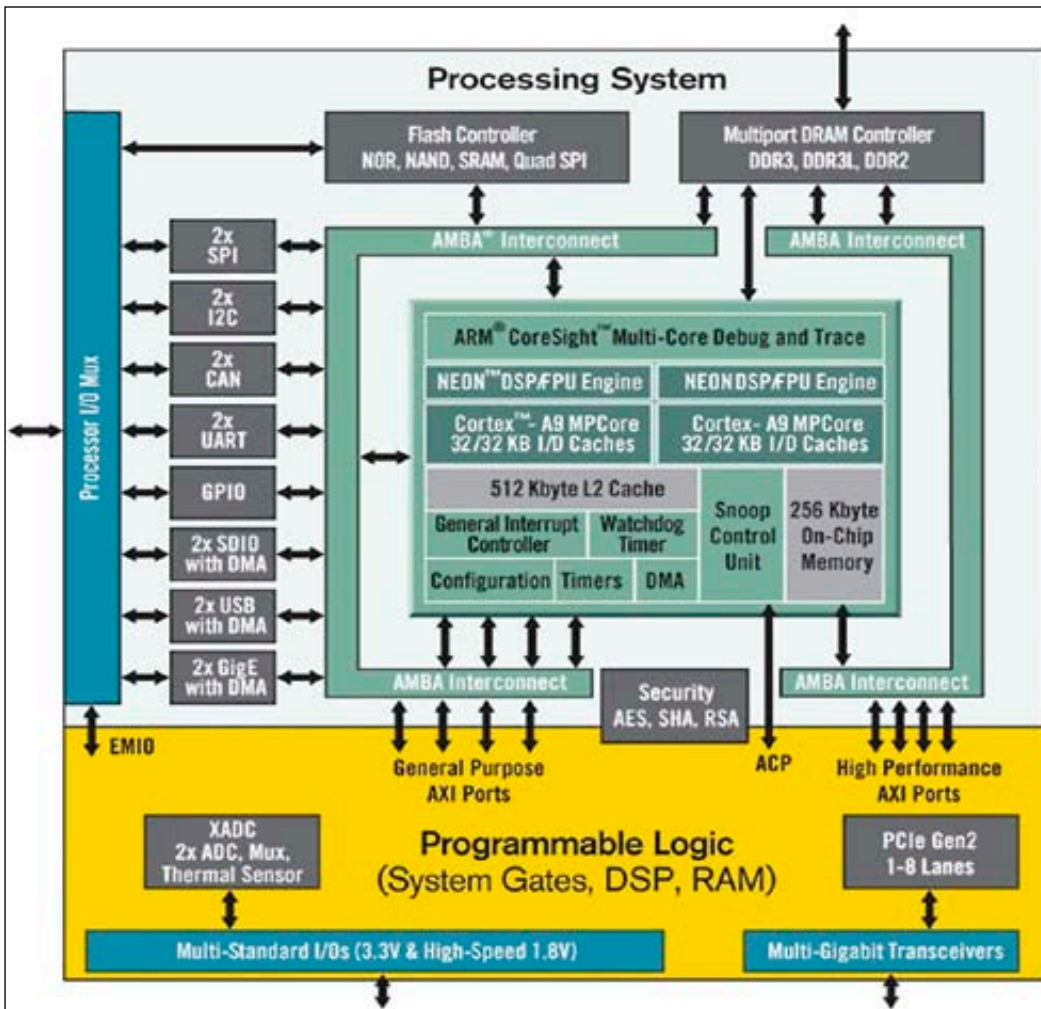


Bild 1. Die Architektur der Zynq All Programmable SoCs.

Mit der explosiven Ausweitung des drahtlosen Datenverkehrs verstärkt sich der Druck auf die IT-Industrie zur Verbesserung der digitalen Signalverarbeitungs- und Funkssysteme. Derzeit liegt der Fokus auf 4G LTE. 4G-Netzwerke werden in großem Maßstab rund um die Welt installiert. Außerdem sieht man bereits die ersten Forschungs- und Entwicklungsergebnisse für 5G-Netzwerke mit der 1000-fachen Kapazität der 4G-Netze. Diese rasante Technologieentwicklung zeitigt neue und schnell evolviende Anforderungen an die Systemlieferanten – bessere Systemintegration und höhere

System-Performance, niedrigere Materialkosten (BOM), größere Design-Flexibilität, schnellere Time-to-Market und vieles anderes mehr.

Traditionelle ASIC-basierte Bausteine unterstützen Hardware-Lösungen, die zwar die Anforderungen hinsichtlich Leistung und Kosten erfüllen, andererseits aber hohe Einmalkosten (NRE), geringe Flexibilität und lange Entwicklungszeiten bis zur Marktreife bedingen. Um die neuen Anforderungen zu erfüllen und die damit verbundenen Herausforderungen anzugehen, hat Xilinx seine All Programmable SoC (APSoC) Architektur industrieweit eingeführt und

sie erfolgreich in der Zynq-7000 Produktfamilie implementiert.

Auf der Basis der APSoC-Architektur ermöglichen die Zynq-7000-Bausteine breite Produkt-Differenzierungen auf der Systemebene, bessere Integration und Flexibilität durch Hardware-Lösungen, sowie Software- und I/O-Programmierbarkeit (Bild 1). Seit ihrer ersten Ankündigung im Dezember 2011 hat die Zynq-7000-Baustein-Serie in breiten Marktsegmenten wie Telekommunikation, Datenzentren, automotiven und industriellen Anwendungen, sowie im Bereich Aerospace und Defense Eingang gefunden. In der Telekommunikation, speziell in drahtlosen Applikationen, bieten die Zynq-7000-Bausteine besonders markante Vorteile: Ihre integrierte, programmierbare Logik (PL) ist bestens optimiert für die digitale Signalverarbeitung, und das ARM Cortex A9-basierte Prozessor-Subsystem (PS) ermöglicht die äußerst effektive Implementierung von Steuerungsfunktionen für das typische drahtlose Equipment, wie z.B. entfernte Funkstationen und drahtlose Backhaul-Einheiten.

Bei der Erstellung einer drahtlosen Applikation auf der Basis von Zynq APSoC-Bausteinen ist es notwendig, ein geeignetes Betriebssystem (operating system, OS) auszuwählen, das den Anforderungen dieser Applikation gerecht wird. Dies involviert mehrere Schlüsselfaktoren im Hinblick auf die vorgesehenen, oft recht unterschiedlichen drahtlosen Applikationen, die man beachten sollte:

1. Carrier-grade Betrieb: Hohe Systemzuverlässigkeit (99,999%) ist die geltende Anforderung für Systeme auf Betreiber-niveau. Sie definiert den Grad der Verfügbarkeit, die für die betrachtete Einheit notwendig ist. Im Betrieb bedeutet dies, dass die Unterstützung der Systemeigenschaften, wie

Yuan Gu
Senior Corporate Global
Accountmanager bei Xilinx

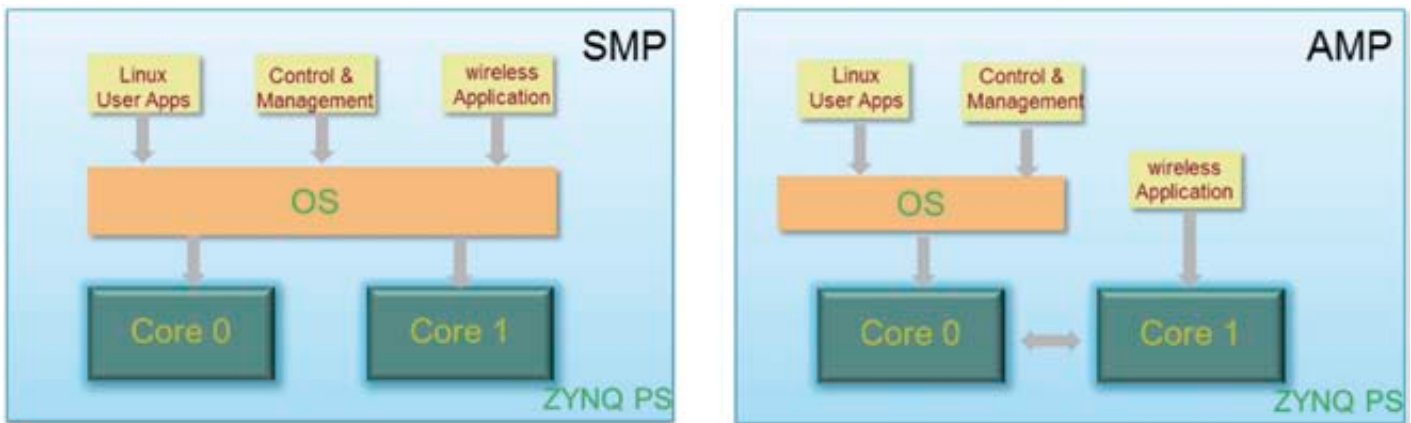


Bild 2. Gegenüberstellung von SMP und AMP.

Kalt/Warm-Restartfähigkeit, die Überwachung, Detektion und das angemessene Handling auftretender Störungen sowie die geforderte Redundanz gewährleistet sind.

2. Echtzeitverarbeitung: Echtzeit in diesem Kontext bedeutet eine vorhersagbare Ansprechzeit, also nicht nur die Aussage „sehr schnell“. Eine entfernte Funkstation stellt andere Anforderungen bezüglich der Echtzeitverarbeitung als die drahtlose Backhaul-Verarbeitung. Das Funk-Equipment ist verarbeitungs-intensiv hinsichtlich des Datenanfalls, und die Prozessoren für die Signalverarbeitung müssen stringente Zeit-Budgets einhalten.

3. Diagnostik: Zur Unterstützung der Diagnostik im Feld und nach einem Ausfall (post mortem) sind umfangreiche Messungen der Performance nötig, und es müssen Logs erstellt und gespeichert werden. Deshalb ist es von besonderer Wichtigkeit, dass eine Reihe von Schlüssel-Indizes einer drahtlosen Applikation im Betrieb überwacht und verwaltet werden. Dazu zählen die Messung der Performance und deren Statistik, CPU-Ausnutzung und Störungs-Monitoring, die Taskwechsel-Funktion des Betriebssystems, Event-History und anderes mehr.

4. Tools und Protokoll-Integration: Eine umfassende Integration des Debugging und der Diagnose-Umgebung neben den spezifischen Netzwerk-Protokoll-Stacks, wie sie von manchen Betriebssystemanbietern

zur Verfügung gestellt werden, ist bei der Entwicklung und Wartung effizienter Systeme von großem Vorteil.

Das Zynq SoC integriert zwei ARM Cortex-A9-Kerne. Das heißt, die Software-Entwickler müssen zwischen den unterstützten Multiprozessor-Architekturen wählen:

- SMP (Symmetric Multi-Processing) oder
- AMP (Asymmetric Multi-Processing).

Wie aus Bild 2 ersichtlich, ist SMP eine Systemarchitektur, in der zwei oder mehr identische Prozessoren auf gemeinsame Ressourcen mit nur einer OS-Instanz zugreifen. Theoretisch behandelt die SMP-Architektur alle Prozessoren unter derselben OS-Instanz gleich. Im Gegensatz dazu arbeiten in der AMP-Architektur die Prozessoren – mit oder ohne OS-Instanz – separat und ohne gegenseitige Erkennung. Der Kern, der ohne Betriebssystem läuft, kann einen Mikrocode ausführen. Dies wird als „bare metal“-Instanz (physische Maschine ohne Virtualisierung) bezeichnet.

Im Allgemeinen bietet SMP eine einheitliche OS-Plattform für Applikationen auf einer höheren Ebene. Ein Software-Architekt muss sich also nicht mit der Ressourcenteilung auf beide Kerne und deren Kommunikationen befassen, wenn er eine Applikation auf diesem Betriebssystem erstellt. Außerdem bedingt SMP einen gewissen Performance-

Overhead, der die Leistung von zeitkritischen drahtlosen Applikationen beeinflussen kann. Im Vergleich mit SMP hat die leichtgewichtige AMP-Software mit einer OS-Instanz wenig oder keinen Overhead. Doch sie benötigt ein sorgfältig konzipiertes Kundensoftware-Design mit geteilten Prozessor-Ressourcen und Kommunikation zwischen den Prozessoren.

Mehrere drahtlose Schlüssel-Applikationen lassen sich sehr effizient in einem der Zynq APSoC-Bausteine implementieren. Dazu zählen Funkeinheiten und drahtlose Backhaul-Systeme. Jede drahtlose Applikation stellt unterschiedliche Anforderungen an die Performance und braucht das Betriebssystem zur Unterstützung verschiedener Eigenschaften. Die Funk-Applikation ist ein gutes Beispiel für den Einsatz des Zynq-Bausteins zur Implementierung einer voll integrierten Hardware- und Software-Lösung, die die gesamte Verarbeitung im digitalen Front-End übernimmt.

Das digitale Front-end ist der Hauptbestandteil einer typischen Applikation in einer entfernten Funkstation (remote radio head, RRH) für drahtlose 4G-Netzwerke. Hier lassen sich die Systemanforderungen aufteilen in Signalverarbeitung und Steuerung. Im Bereich der Signalverarbeitung kann man Zynq zur Implementierung von Filtern mit hoher Abtastrate für die digitale Auf- und Abwärts-Konvertie-

rung einsetzen, und ebenso zur Reduzierung des Scheitelfaktors und für die digitale Vorverzerrung (pre-distortion, DPD). DPD ist ein spezieller Fall: Sie benötigt sowohl Zynq PS, als auch Zynq PL.

Die DPD-Verarbeitung lässt sich aufteilen in den High-speed Datenpfad und den Update-Pfad. Der Update-Pfad dient zur periodischen Aktualisierung der Filterbank-Koeffizienten; er ist gut geeignet zur Implementierung auf dem ARM Cortex-A9-Kern. Typischerweise wird eine Koeffizienten-Aktualisierung innerhalb einiger, oder einiger zehn Millisekunden ausgeführt. Wegen der arithmetischen Komplexität der Berechnungen müssen eventuell auch der A9-Kern und die eingebettete NEON SIMD Vektor-Recheneinheit herangezogen werden, um die erforderliche hohe Performance zu erbringen. Zusätzlich kann der Zynq PL die Hardware-Beschleunigung von Funktionen mit starker Abhängigkeit vom Prozessor-Takt unterstützen. Also werden der Zynq PL, der ARM A9-Kern und der NEON Coprozessor oft gemeinsam und kollaborativ eingesetzt.

Der Steuerzweig des Funksystems wird typischerweise zur erstmaligen Kalibrierung, Konfigurierung, Alarmauslösung, Scheduling und Message-Terminierung auf den Netzen eingesetzt. In einer Funkanwendung erfordert dies meist keine besonders hohe Performance, kann also leicht von einem ein-

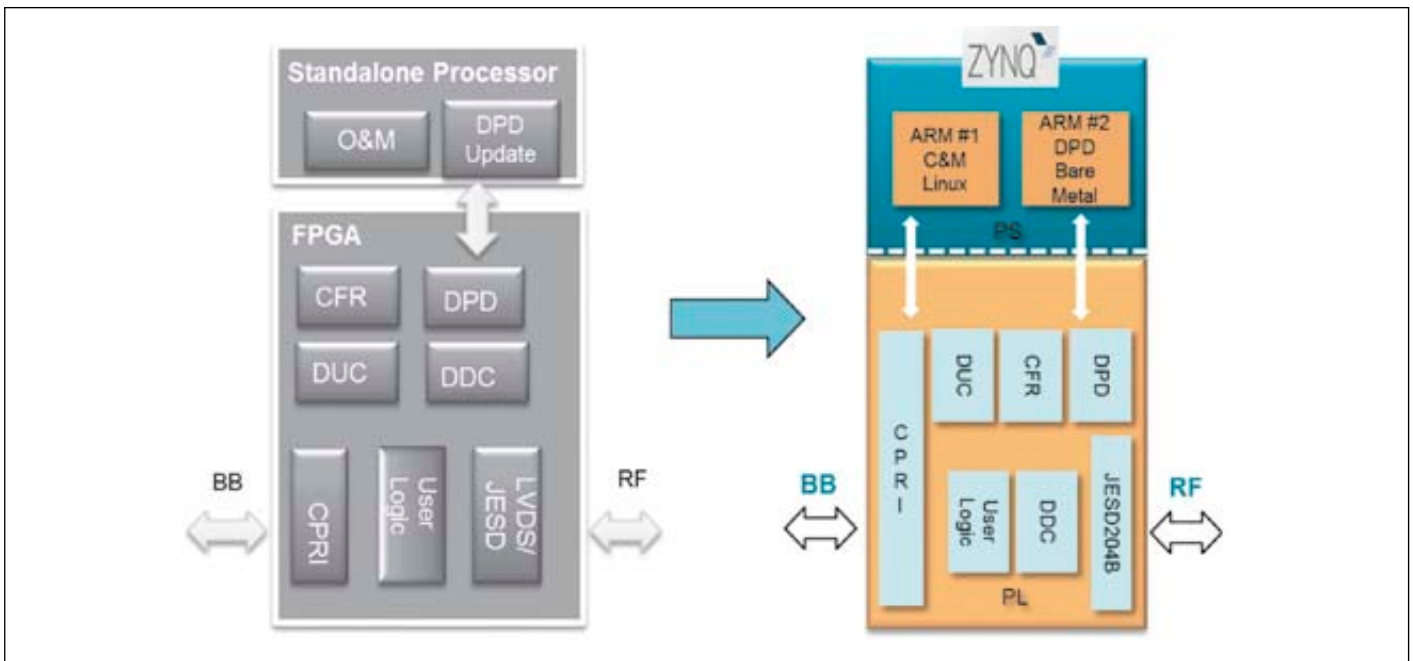


Bild 3. Migration einer diskreten Lösung auf die Zynq-Plattform.

zigen ARM A9-Kern im Zynq erledigt werden.

Die richtige Auswahl der passenden Architektur zur Unterstützung sowohl der DPD-Applikation, wie der Steuerverarbeitung ist sehr wichtig, denn sie definiert die System-Performance, die Zuverlässigkeit und Einfachheit der Wartung.

Eine oft getroffene Wahl der Architektur für Funk-Applikationen ist der AMP-Modus. Diese Struktur verwendet einen ganzen ARM-Kern für die DPD-Verarbeitung, wenn sie im Bare-Metal-Modus läuft. Dabei bietet sie mehr Headroom für den Zeitbedarf der Aktualisierung der DPD-Koeffizienten. Alle anderen Applikationen, wie Steuerung und OAM-Routinen (operation, administraton, maintenance), laufen auf dem zweiten OS-gesteuerten ARM A9-Kern. In dieser Architektur, in der das Betriebssystem nur die Steuerung eines der beiden ARM-Kerne übernimmt, muss ein Kommunikationskanal zwischen den Applikationen auf den beiden separaten Kernen etabliert werden, etwa durch die Nutzung des On-chip-Speichers oder geteilter Speicherbereiche.

Dies ist besonders wichtig für einige relevante Steueraufgaben,

beispielsweise die Überwachung der Funktion des DPD-Moduls. Derartige Lösungen zur Kommunikation zwischen den Prozessen (inter-process communication, IPC) sind nicht standardisiert. Sie müssen im AMP-Modus separat entwickelt werden.

Die SMP-Architektur ist sehr geradlinig konzipiert, mit nur einer OS-Instanz zur Steuerung beider ARM-Kerne – und damit aller Applikationen. IPC, Debugging und Tool-Chain-Unterstützung laufen alle unter demselben Betriebssystem. Um sicherzustellen, dass der DPD-Applikation genügend Ressourcen zugeteilt werden, lassen sich spezifische Verfahren wie Core Affinity oder Interrupt Shielding in der Software-Applikation einsetzen. Im ersten Fall läuft die DPD-Applikation nur auf einem Kern, und potenziell teilen sich keine anderen Tasks die Ressourcen (außer dem OS Scheduler Overhead). Im zweiten Fall werden die Interrupt-Dienste (außer denen, die von der DPD-Applikation ausgelöst werden) auf den zweiten Kern geleitet. Somit werden die Ressourcen von der DPD-Applikation voll ausgelastet.

So gesehen ist das Zynq APSoC die ideale Plattform zur Unter-

stützung sowohl der AMP-, als auch der SMP-Architektur. Wie in Bild 3 gezeigt integriert Zynq zwei ARM-Prozessoren, einen 12,5-Gb/s SerDes und einen 500-MHz+ DSP mit höherer Zuverlässigkeit. Er liefert damit die gesamte digitale Front-end Funktionalität, wie DPD, CFR, DUC/DDC und CPRI/JESD-Schnittstellen. Diese Lösung umgeht die Notwendigkeit der Schnittstellen zwischen den Prozessoren und separatem FPGA. Sie vereinfacht somit das PCB-Design.

Die Migration von einer diskreten Multi-Chip-Lösung zur integrierten Single-Chip-Lösung auf der Zynq-Plattform ist relativ einfach und geradlinig. Xilinx bietet einen glatten Migrationspfad auf Zynq durch die Bereitstellung umfangreicher Hardware- und Software-Lösungen. Darin eingeschlossen ist eine IP-Bibliothek der digitalen Signalverarbeitung für DUC, DDC, CFR und DPD. Zusätzlich werden multiple Betriebssystem-Lösungen unterstützt, einschließlich Bausteintreiber, Boot Loader, BSP-Template und den gebräuchlichen Tools. Mit einer erfolgreichen Migration auf die Zynq-Plattform kann diese Lösung die System-Performance

signifikant erhöhen, den Leistungsverbrauch senken und die Materialkosten niedrig halten.

Fazit

Der Beitrag erläutert die Schlüsselanforderungen bei der Auswahl des Betriebssystems für drahtlose Applikationen, die Architektur der Implementierung und die Kriterien hinsichtlich AMP versus SMP bei deren Einsatz auf einem Xilinx Zynq-7000 Baustein.

Diese fortschrittlichen Bausteine von Xilinx erbringen bei der Entwicklung von Equipment für die Telecom-Infrastruktur eine wesentlich verbesserte Performance, leichtere System-Integration, geringere Materialkosten und geringeren Leistungsverbrauch bei hoher Zuverlässigkeit und kürzerer Time-to-Market. Daneben ermöglichen sie die volle Programmierbarkeit sowohl der Software wie der Hardware. Damit können die Entwickler ihre Systeme schneller fertig stellen, aber auch weiterhin Aktualisierungen des Equipments nach dessen Installation im Feld vornehmen. Damit umgehen sie das mit dem Einsatz von Bausteinen wie ASSPs und ASICs verbundene Risiko. ◀