

inVISION

MACHINE VISION · IDENTIFICATION · IMAGING

USB
VISION



POINT GREY
Innovation in Imaging

Nicht alle Board-Level-Kameras sind gleich

Top Innovationen 2015
inVISION prämiert zehn innovative
Produkte und Lösungen

SPS & Bildverarbeitung
Experten diskutieren über Daten-
austausch und Standards

Neue
Rubrik:
**Embedded
Vision**

Marktübersichten
- Zeilenkameras
- CoaXPRESS-Framegrabber

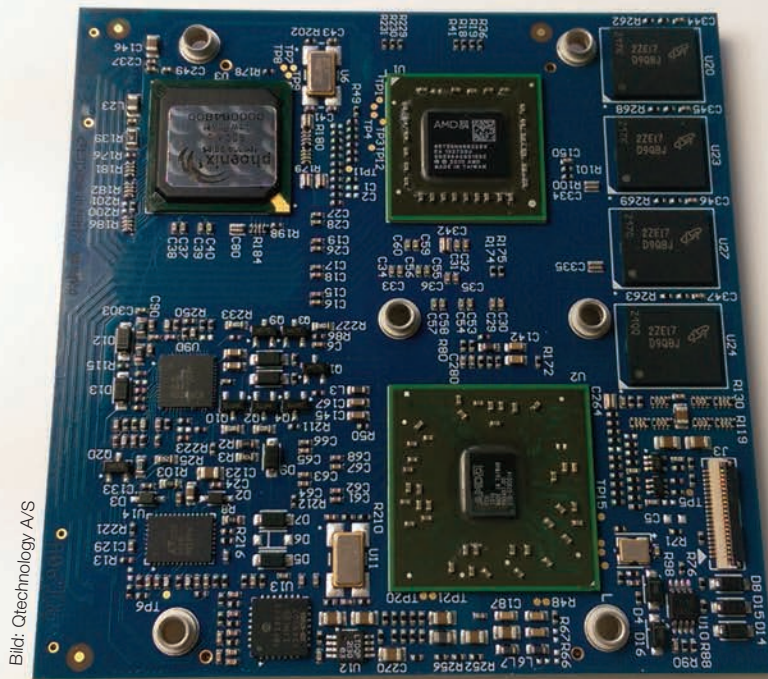


Bild 1 | Das 80x80mm große I/O-Modul enthält eine AMD Embedded G-Series APU T56N und einen FPGA.

APU vs. FPGA

Was setzt sich bei intelligenten Kameras durch?

Die Innovationen bei Smart-Kameras schreiten immer schneller voran. Vision-Systeme werden immer schneller, kleiner und energieeffizienter. Dieser Trend ist besonders ausgeprägt im stark wachsenden Bereich der x86-basierten intelligenten Kameras, bei denen man zugunsten von Designeffizienz, Funktionsvielfalt und Rapid Prototyping auf herkömmliche, sehr begrenzt unterstützte FPGA-basierte Verarbeitungsplattformen verzichtet.

Da der Markt der smarten Kameras hart umkämpft ist, wird die Designflexibilität für die Hersteller intelligenter Kameras immer wichtiger. Sie ermöglicht es, ihre Produkte schneller zur Marktreife zu bringen. Gerade hier zeigt sich aber, dass FPGAs als zentrale Verarbeitungseinheit von Vision-Systemen in die Jahre gekommen sind. Sie verzögern den Entwicklungsprozess auch für erfahrenste Entwicklerteams und es ist sehr aufwendig, sich FPGA-Fachwissen anzueignen. Ältere, rein FPGA-basierte intelligente Kameras verlangen zudem

von Entwicklern geradezu hellseherische Fähigkeiten, was Auswirkungen auf die Effizienz der Entwicklung hat. FPGAs sind komplexe, höchst leistungsfähige Bauteile, die sich an spezifische Zielanwendungen anpassen lassen. In einigen Anwendungsfällen sind sie überzeugende Alternativen zu x86-Prozessoren. Ihr tatsächlicher Wert hängt jedoch wesentlich vom Verständnis des Entwicklers für logische Schaltungen ab. Seitens der Software ist der Wettstreit zwischen FPGAs und vergleichbar leistungsfähigen x86-Plattfor-

men – insbesondere Accelerated Processing Units (APUs) – um einen möglichst einfachen Entwicklungsprozess allerdings nicht ausgeglichen. Bei Konfigurationen mit FPGAs und getrennter CPU müssen bei den meisten Vision-Anwendungen die rechenintensiven Bildverarbeitungsalgorithmen im FPGA implementiert werden. Dies ist eine Einschränkung, da die VHDL (VHSIC Hardware Description Language) und die passenden Kernel-Treiber für jede Zielanwendung eigens entwickelt werden müssen. Bei einem solchen Layout



Bild: Qtechnology A/S

Bild 2 | In der intelligenten Kamera von Qtechnology übernimmt eine AMD Embedded G-Series APU die Berechnungen auf den höheren Ebenen. Ein kostengünstiger FPGA nimmt die Sensorsignale entgegen und übernimmt das Bild-Preprocessing.

tenfluss zu steuern. Alle Algorithmen zur Bildverarbeitung (Bildkorrektur, Segmentierung, Merkmalerkennung usw.) müssen deshalb in VHDL im FPGA implementiert werden.

Der entscheidende Faktor

Mit x86-basierten APUs (im konkreten Fall AMD Embedded G-Series T56N APUs) kann im Vergleich ein zuvor unerreichbarer Grad an Entwicklungs-Effizienz erzielt werden. Der wesentliche Grund dafür ist, dass Bildbearbeitungs-Programmbibliotheken wie OpenCV, Matlab oder Halcon aus dem Open-Source-Bereich und von Drittanbietern genutzt werden können. Gleiches gilt für Debugging-Tools für die x86-CPU und OpenCL für den Grafikprozessor. Der x86er kann vollständig mit Standard-Tools geprüft werden, u.a. gibt es Latenz-Analysatoren und Profiler (perf, ftrace), moderne Debugger und integrierte Entwicklungsum-

zieht jede Änderung beim Algorithmus, Datenfluss oder sogar beim Debug- und Prüfungsprozess einen zeitaufwendigen Respin der FPGA-Implementierung nach sich. Zudem muss ein Modell zur Aufteilung der Tasks festgelegt werden, was Auswirkungen auf System-

ebene hat, wenn ein Re-Design erforderlich ist. Oft ist es aber so, dass die bei FPGAs eingebettete CPU nicht leistungsfähig genug ist, um die von den Sensoren erzeugten Daten in Echtzeit verarbeiten zu können, daher wird sie zumeist nur dafür verwendet, den Da-

W3+FAIR

OPTICS, ELECTRONICS & MECHANICS

**DREI BRANCHEN
EIN NETZWERK
EINE MESSE**

**25.+ 26. MÄRZ 2015
RITTAL ARENA WETZLAR
www.w3-messe.de**

KOSTENLOS
Jetzt registrieren!

Hochkarätige Unternehmen und Hidden Champions

Fachseminare und Keynotes powered by Wetzlar Network

Neue Recruiting Lounge

Start-Up Area powered by Hessen-Nanotech/NANORA

Short Courses powered by Optence

SPONSOREN

PARTNER

gebungen (Eclipse, gdb) sowie Power-Analyser (powertop). All dies ist in mächtigen Entwicklungssystemen wie dem Yocto-Projekt zusammengefasst, einem Open-Source-Projekt, das Templates, Tools und Methoden für maßgeschneiderte Linux-basierte Systeme für hardwareunabhängige Embedded-Produkte bereitstellt. Die Anwendungen können ohne Kreuzcompiler oder (meistens kaum gewartete) Support-Pakete für das Board auf jedem Rechner entwickelt und getestet werden. Eine APU, die Kombination einer GPU mit der CPU auf einem Chip, ermöglicht es, rechenintensive Prozesse in der Bildverarbeitung ohne Verzögerungen durch eine Bus-Übertragung von der CPU in die GPU auszulagern. Die CPU kann andere Interrupts mit wesentlich weniger Verzögerung bearbeiten, die Echtzeitleistung des Gesamtsystems steigt und die stetig steigenden Anforderungen heutiger Vision-Systeme an die Rechenleistung werden besser erfüllt. Mit dieser Architektur lässt sich die gesamte Verarbeitungsplattform mit einem standardmäßigen Linux-Kernel verwalten, was bei neuen Kernel-Versionen nur geringen Aufwand verursacht. Die Übertragung auf das Board ist unter jedem Betriebssystem (Linux, Windows) möglich. Um die Entwicklungszyklen noch weiter zu vereinfachen und zu verkürzen, wurde die APU in einen Panel-PC und I/O-Module nach Industriestandards integriert, sodass man Programmcode mit einer einzigen Distribution auf Basis des Yocto-Projekts auf verschiedenen Systemen wiederverwenden kann. Um Standardprotokolle wie Ethercat und Modbus zu unterstützen, genügt es, existierende Programmbibliotheken zu integrieren. Da diese für gewöhnlich in x86-Umgebungen entwickelt werden, müssen sie nicht portiert oder gesondert gewartet werden. Ein zusätzlicher Vorteil der ausgewählten AMD-APU-Plattform ist die Möglichkeit, sie zur Leistungssteigerung mit einem AMD-Radeon-basierten Grafikprozessor in einer MXM-Lösung zu kombinieren. So lässt sich im Bedarfsfall problemlos zusätzliche Leistung für rechenintensive Anwendungen bereitstellen.

Offene Standards

Mit APUs anstatt FPGAs als zentrale Verarbeitungseinheit in intelligenten Kameras können offene Entwicklungstools wie OpenCL genutzt werden. Diese nicht-proprietären Programmierplattformen ermöglichen es dem Hersteller und seinen Kunden, höhere Leistungen in der Parallelverarbeitung zu erzielen. Dies bei gleichzeitiger Skalierbarkeit über ein breites Spektrum von Vision-Systemen und auf der Basis des gleichen Sourcecodes. Die Möglichkeit, einen zukunftsicheren, portablen Quellcode zu entwickeln und zu pflegen, steigert die Effizienz der Programmierung immens. OpenCL ist ein offener, lizenzfreier Programmierstandard für die Parallelverarbeitung auf heterogenen Systemen. Er bietet Entwicklern von Vision-Systemen eine plattformübergreifende, nicht-proprietäre Lösung, um ihre Anwendungen auf populären Plattformen wie APUs, Multicore-Prozessoren und GPUs zu verbreiten. Eine einheitliche Sprache genügt für alle Parallelverarbeitungs-Elemente innerhalb eines Systems. Entwickler arbeiten mit einem abstrakten Plattformmodell, das die unterschiedlichen Architekturen in gleicher Weise darstellt; das Ausführungsmodell unterstützt zudem die Parallelität von Daten und Tasks über heterogene Architekturen hinweg. Syntax und Workflow-Attribute von OpenCL lehnen sich an die Software-Welt an, was die Sprache wesentlich eingängiger macht als VHDL. OpenCL kann prinzipiell auch für FPGA-basierte intelligente Kameras implementiert werden. Es erfordert jedoch tiefgreifende Kenntnisse der FPGA-Technologie, um eine höhere Programmiersprache in die Hardwarebeschreibungssprache (HDL) der FPGAs umzuwandeln – der Markt für solche 'OpenCL-nach-HDL-Entwicklungsumgebungen' ist sehr überschaubar. Auch hier gilt: Um FPGAs effektiv einzusetzen, sind hochspezifische Kenntnisse von Nöten, die Tool-Unterstützung ist dürftig und die Entwicklungszyklen oft langwierig.

Gut angebunden

Durch ihre PC-gleiche Flexibilität in Verbindung mit branchenüblicher, für x86er-Prozessoren optimierter Software, eröffnen x86-basierte Vision-Systeme vielfältige Möglichkeiten zur Effizienzsteigerung bei Entwicklung, Distribution und Administration. Dadurch eignen sich die intelligenten Kameras z.B. zum Aufbau neuronaler Netzwerke für anspruchsvollen Anwendungen in Industrien, die Echtzeit-Daten in höchsten Frequenzen zu verwertbaren Informationen aufbereiten. Weiterhin ermöglicht die x86-Unterstützung eine enge Integration in IT-Netzwerke und x86-basierte Prozessleitsysteme, woraus sich zusätzliche Vorteile für in diesen Netzwerken gehostete Anwendungen ergeben. So kann man problemlos Remote-Management und -Debugging realisieren und damit den Kunden-Service weiter verbessern. Wohlgedemert, die hier beschriebene Applikation verzichtet nicht vollkommen auf FPGAs in intelligenten Kameras. Statt der ursprünglichen rein FPGA-basierten Architektur verwendet Qtechnology aber eine Kombination aus APU und kostengünstigen FPGAs. Der FPGA nimmt die Signale entgegen und übernimmt das Bild-Preprozessing. Die FPGA-Funktionalität ist über das Standard-Video-Aufnahme-API für Linux Video4Linux2 (V4L2) angebunden. Jede Standard-Anwendung (Skype, Ekiga usw.) und -Bibliothek (OpenCV, Gstreamer) kann auf die Sensoren zugreifen. Die weitere Verarbeitung erfolgt mit Standard-Bildverarbeitungsbibliotheken oder benutzerdefinierten OpenCL-Programmen. Eingriffe seitens des Hardwareentwicklers sind nicht notwendig. ■

www.qtec.com

Autoren | Kristian Gløde Madsen, Geschäftsführer, Qtechnology
Ricardo Ribalda, Firmware-Entwickler, Qtechnology