



Halle 2
Stand 128

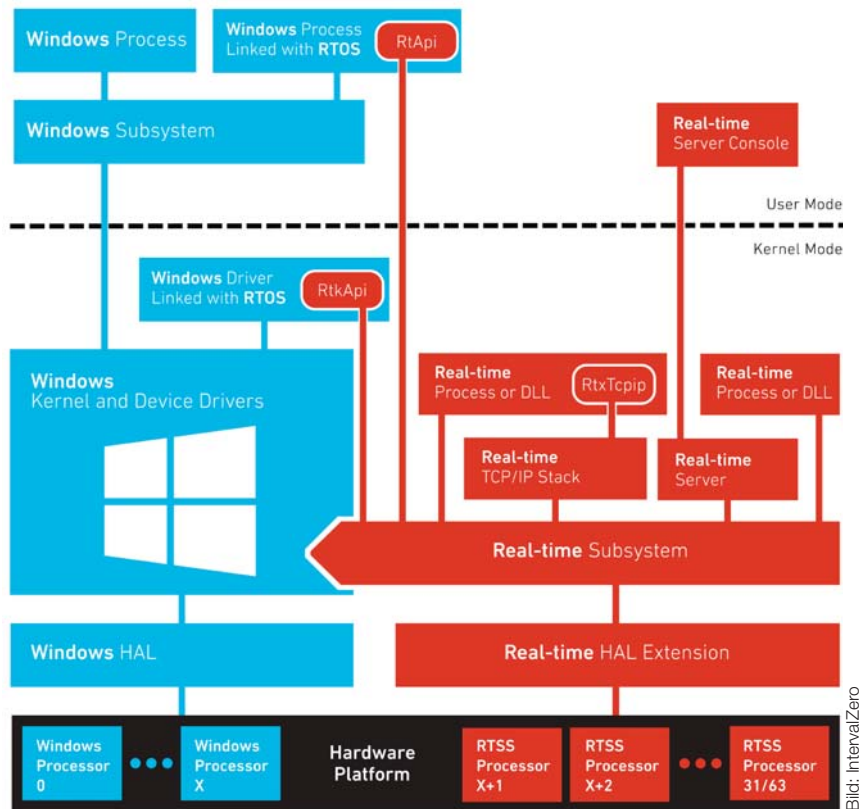


Bild: IntervalZero

Bild 1 | RTX64 stellt eine Architektur bereit, die bisherige Embedded-Umgebungen, die aus DSPs, FPGAs und Micro-Controllern besteht, übertrifft.

Multikernelnutzung

Windows und Echtzeit-OS in einer Hardware

Bisher bestanden Bilderkennungssysteme oft aus zwei Teilen: Einem System zur Erfassung und Aufbereitung und einem zur Steuerung. Heute wächst der Bedarf nach vereinten und integrierten Umgebungen, um Kosten zu verringern und Systeme leichter konfigurierbar zu machen. Das RTX64 System verbindet Echtzeit-Leistung in einer Multikernel-SMP (Symmetric Multi Processing) Umgebung mit Shared Memory, die gleichzeitig die Parallelität von Bilderfassung/-verarbeitung sowie eine Echtzeit-Steuerung zusammen mit einer Windowsumgebung erlaubt.

Solch ein System, basierend auf RTX64, könnte auf einem Multikernel-PC mit mehreren parallelen Echtzeitkernen implementiert werden, mit Microsoft Windows als Schnittstelle zum Menschen, Verbindung zu externen Netzwerken (Internet,...) und programmiert über Microsoft Visual Studio. Die Parallelität wird durch die Fähigkeit von Threads, die auf verschiedenen Kernen laufen und auf Shared Memory zuzu-

greifen verbessert, was mögliche Bild-einzugsprobleme oder das Problem der Datenduplizierung vermeidet.

Windows als Echtzeit Betriebssystem

RTX64 verwandelt Windows in ein voll funktionelles Echtzeit Betriebssystem, das vollständig auf x64 Multicore-Hardware läuft. Es erlaubt den Zugriff auf bis

zu 128Gbytes nicht-paginierter Speicher, abhängig von der tatsächlich abgebildeten physikalischen RAM-Größe. Die 512Gbytes physikalischer Speicher von 64-bit Windows übertrifft bei weitem die 4GByte Limitierung von 32-bit Windows für physikalischen Speicher. Die enorme Menge an verfügbarem Speicher öffnet so die Tür zur Implementierung mächtiger Bildverarbeitungssysteme, die große Mengen an Bilddaten aufnehmen kön-

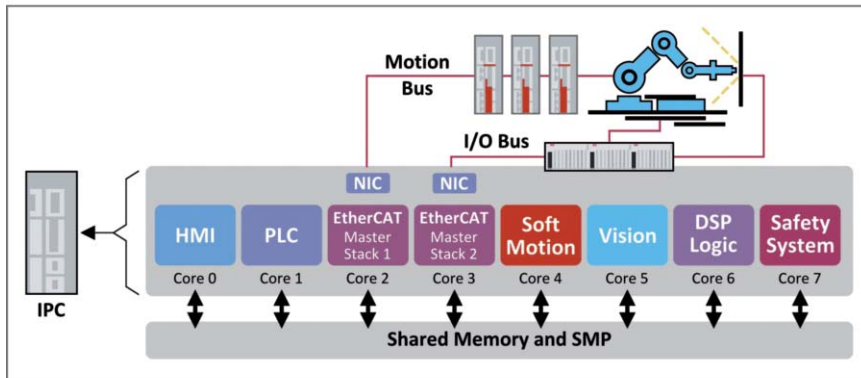


Bild 2 | Die Möglichkeit unterschiedliche Funktionen verschiedenen Kernen zuzuordnen, die über Shared Memory kommunizieren, macht es einfach, verschiedene Werkzeuge auszutauschen, die vom Bildverarbeitungssystem gesteuert werden.

nen und allen im System laufenden Kernen gleichermaßen verfügbar sind. Vor allem erlaubt RTX64 eine Hardware Umgebung aus überall erhältlicher Handelsware in Form von Multicore x64-Geräten. Dies ermöglicht eine Software Umgebung, die Windows mit allen Benutzerschnittstellen, Anwendungen und der Entwicklungsumgebung beherbergen kann. Windows ist zudem verbunden mit der echtzeitfähigen SMP (Symmetric Multi Processing) RTX64 Umgebung, die von eins auf bis zu 63 Kernel skaliert werden kann. Somit können Anwendungen auf einer gemeinsamen Code Basis kompiliert werden, ohne FPGAs oder DSPs zum Ausführen der Logik ein-

zusetzen, deren Code sonst separat kompiliert und mit der Hauptanwendung extra verknüpft werden müsste. RTX64 hat einen Hardware Abstraktionslayer (HAL) der sich vom Windows HAL unterscheidet, aber nebenher arbeitet, und keine Modifikation an Windows erfordert. Beide Systeme operieren Seite an Seite und kommunizieren über existierende Mechanismen. Die RTX64 HAL kann deterministische Echtzeit-Leistung mit Zeitakten bis zu $1\mu\text{s}$ (abhängig von der Hardware Unterstützung) liefern. Der im Echtzeit Subsystem (RTSS) enthaltene Scheduler, kann Threads auf Kerne zuordnen, um symmetrisches Multiprocessing (SMP) zu erreichen, ohne auf Virtua-

lisierung, oder andere komplexe Interprozess-Kommunikationsmechanismen angewiesen zu sein. Der gesamte Speicherraum ist allen Kernen ohne Speicher Partitionierung verfügbar. Das ist ein großer Vorteil für bildverarbeitungsgesteuerte Anwendungen und hochentwickelte industrielle Steuersysteme, die nicht nur visuelle Daten für den Anwender anzeigen, sondern diese auch in Echtzeit verarbeiten, um Bewegungen von Werkzeugen zu steuern, sowie für die Inspektion von erstellten Teile zur Verfügung stellen. Ein so großer Speicherraum, verfügbar auf einer Hardware Plattform, erlaubt es OEMs Software zu entwickeln, die extrem spezialisierte Funktionen enthält, für die ansonsten weitere Hardware Komponenten benötigt wurden. Die Software kann immer wieder repliziert werden und das Team benötigt kein Wissen über spezialisierte Hardware wie DSPs und FPGAs. Der Code existiert in einer vereinheitlichten Code Basis und kann als solcher verwaltet werden.

Mächtige Visionssysteme

Ein RTX64 System ermöglicht die Implementierung eines mächtigen und vielseitigen Bildverarbeitungssystem, das zum größten Teil in der Software festgelegt ist. Die primäre Entscheidung bei der Hardware würde dann die Aus-

wahl der Kamera sein, sowie der speziellen Vorrichtungen, die gesteuert werden sollen, wie z.B. ein Roboterarm, Schweißgerät, Fräswerkzeug oder ein Luftstoß, um aussortierte Teile von einem Förderband zu stoßen. Angesichts der großen Verarbeitungsmöglichkeiten eines Multicore RTX64 Systems ist das sorgfältige Abwägen der Kamerafeatures gegen die Kosten des neuen Systems ein wichtiger Schritt. Das Erledigen von Aufgaben durch die gesteigerte Leistungsfähigkeit kann die Kosten für die benötigte Kamera erheblich verringern. Aus diesem Grund empfiehlt IntervalZero den Schnittstellen Standard GigE-Vision, der schnelle Bildübertragung mit kostengünstiger Standard Ethernet-Kabeln erlaubt und Verbindungen über eine Länge von bis zu 100m zulässt, bzw. bis zu 5.000m mit Glasfaserkabeln. Da GigE Vision eine Standard-Ethernet Verbindung nutzt, ist es nicht auf spezialisierte Hardware wie z.B. einen Framegrabber angewiesen, um Daten an die Anwendung zu übertragen. Darüber hinaus kann es mit einer eigenen IP-Adresse für jede Kamera auch mehrere Kameras auf demselben Netzwerk unterstützen. Des Weiteren setzt GigE Vision auf GenICam auf und kann daher jede beliebige Kamera einbinden, die über GigE-Vision Ethernet- oder EtherCAT-Kabel mit dem RTX64 System verbunden ist. Es bietet etliche Module, einschließlich eines generischen Steuerprotokolls, Namenskonvention, API und einen Transport Layer, der einfach angepasst werden kann, um verschiedene Kamera-/Anwendungskombinationen aufzunehmen.

Kommunikation über EtherCAT

Wie auf der Kameraseite bringt EtherCAT Vorteile auf Seiten des Steuerkreises, der auch alte Feldbus Komponenten wie Profibus oder CANopen einbinden kann. EtherCAT läuft in der Software unter RTX64, ohne dass eine spezialisierte Karte in den Systembus gesteckt

werden muss. Unabhängig davon, ob es auf einem oder mehreren Prozessorkernen läuft, kommuniziert es direkt mit jedem Netzwerk Schnittstellen Chip, der im System genutzt wird. Das individuelle Gerät kann während der EtherCAT-Konfiguration gewählt werden. Natürlich können auch andere Schnittstellen, wie USB oder PCI Express, leicht und kostengünstig integriert werden. Der Vorteil ist, dass mit diesen Schnittstellen verbundene Maschinen oder Geräte von der Steuerkreis-Software verwaltet werden, die unter RTX64 läuft, welches wiederum selbst von der Bildverarbeitung gesteuert wird, die auf dem gleichen System läuft. Bildverarbeitung und Motion Control sind somit integrale Bestandteile des Systems. Zur gleichen Zeit jedoch sind Steuer- und Bewegungsfunktionen – dank der Multicore-Funktionalität – modular und können auf einem zugeordneten Kern laufen, der mit dem Bildverarbeitungssystem über Shared Memory kommuniziert. Wenn der gleiche Bohrer genutzt werden soll, um z.B. andere Arbeitsabläufe an einem Werkstück auszuführen, kann die Anwendung durch einfaches Nachrüsten der Softwareteile angepasst werden. Wenn es nötig ist, kann auch die gesamte Anwendung ausgetauscht und ein anderes Werkzeug gewählt werden, aber dieses würde immer noch die gleiche RTX64 Windows-basierte Hardware-/Betriebssystem Umgebung nutzen. Der Multicore Charakter des RTX64 Systems erlaubt Parallelität zwischen verschiedenen Funktionen, so dass sie sich nicht gegenseitig beeinflussen. Ein Interrupt, der für die Steuerung gedacht ist, wird daher die Bildverarbeitung nicht beeinträchtigen, die auf einem anderen Kern läuft. Datenerfassung und Bedienerschnittstelle, die unter Windows laufen, funktionieren reibungslos. Abhängig davon, wie viele 64-bit Kerne ins System eingebunden sind, kann auch ein Teil der Bildverarbeitung zwischen einer bestimmten Anzahl von Kernen aufgeteilt werden. Mit Windows als Bedienerschnittstelle ist es zudem möglich auf das Internet oder andere

Anwendungen direkt zuzugreifen oder Bilder anzuzeigen, mit denen das Bildverarbeitungssystem arbeitet.

Softwareentwicklung unter Windows

Basierend auf der Microsoft Visual Studio Umgebung kann die Softwareentwicklung komplett unter Windows mit Sprachen wie C/C++ und Suiten von Editoren, Debuggern und Analyse-Werkzeugen stattfinden. Außerdem steht eine ganze Reihe von Softwarebibliotheken zur Verfügung, die speziell auf die Bildverarbeitung abzielen, und die Funktionen wie Kantenfindung, Objekt-/Merkmalkennung, Extraktion...beinhalten. IntervalZero kooperiert mit einer Anzahl von Lieferanten, um deren Bibliotheken auf RTX64 zu portieren. Das Wesen eines bildgeführten Servosystems ist es, dass es nicht auf Sensoren und Positionierungsmechanismen angewiesen ist, deren Parameter präzise definiert werden müssen. Ein Gestell zum Halten eines Teiles würde z.B. eine Spannvorrichtung benötigen, die definiert sein müsste, um das Teil zu laden und in einer exakten Position zu halten. Ein Bohrer müsste programmiert werden, um basierend auf der Teileposition die exakten Stellen zu finden. Für ein weiteres Teil müssten Arbeitsablauf, Positionierung und Steuerung neu definiert werden. Mit einem Bildverarbeitungssystem braucht das Teil nicht so präzise platziert zu werden, weil das System die Koordinaten in einem angelernten Muster mit dem aufgenommenen Bild in Einklang bringt, um das Werkzeug mit der nötigen Genauigkeit zu steuern. Verschiedene Abläufe könnten in der Lage sein, das selbe Positionierungsgerät zu nutzen, und natürlich den gleichen menschlichen Bediener. ■

www.intervalzero.de

Autor | Daron Underwood, CTO, IntervalZero