

SPARXIL: Ein konfigurierbarer FPGA-Coprozessor

Andreas Koch

Abteilung Entwurf integrierter Schaltungen

TU Braunschweig, Gaußstr. 11, 38106 Braunschweig

Tel. 0531/391-2384, FAX 0531/391-5840, a.koch@tu-bs.de

Einleitung

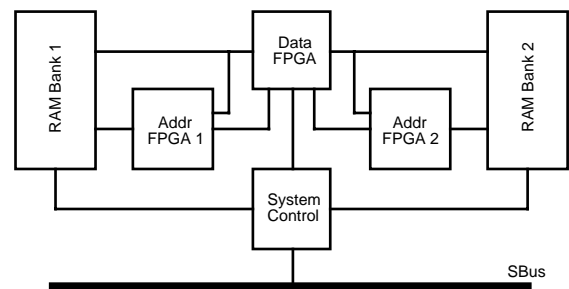
Trotz der ständig steigenden Leistungsfähigkeit moderner CPUs hat die Bedeutung spezialisierter Coprozessoren keineswegs abgenommen. Sie können die CPUs in vielfältiger Art und Weise entlasten. Dazu zählen beispielsweise mathematische Berechnungen, graphische Operationen, Ein-Ausgabeprozesse und spezialisierte Algorithmen für Fuzzy-Logic oder die Simulation neuronaler Netze. Erst durch den Einsatz von Coprozessoren kann in vielen dieser Bereiche die geforderte Leistung erreicht werden.

Es liegt in der Natur der Sache, daß solche Coprozessoren stark spezialisiert sind und nicht alle denkbaren Anwendungen abdecken. Aber auch zur Lösung eines bisher nicht bearbeiteten Problems wünscht man sich gelegentlich spezialisierte Hardware, sei es, um den Zielalgorithmus direkt zu beschleunigen, oder um mit verschiedenen Architekturen zu experimentieren, bevor ein neuer, spezialisierter Coprozessor gefertigt wird. Hier kann ein konfigurierbarer Coprozessor, der individuell an eine gegebene Problemstellung angepaßt werden kann, Abhilfe schaffen. Die neu gewonnene Flexibilität wird dabei durch eine im Vergleich zu konstanten Coprozessoren verminderte Leistung erkauft.

Der SPARXIL Coprozessor

SPARXIL ist ein Vorschlag für einen solchen konfigurierbaren Coprozessor. Die Konfigurierbarkeit wird durch die Verwendung beliebig oft programmierbarer FPGA-Bausteine erreicht, die den Kern des Coprozessors bilden. Da ein einzelner der verwendeten Xilinx XC4010 FPGAs nur Schaltungen minderer Komplexität implementieren kann, werden die Operationen des zu beschleunigenden Algorithmus auf mehrere FPGAs

aufgeteilt. Aus Geschwindigkeits- und Kostengründen wurde bei SPARXIL auf ein variables Kommunikationsnetz zwischen den FPGAs verzichtet. Die Chips sind derart verbunden, daß jeder einzelne im Prozessor für bestimmte Operationen verantwortlich ist. So werden beispielsweise zwei der FPGAs in der Regel als Adreßgeneratoren zur Speicherverwaltung konfiguriert, während das dritte als Datenpfad fungiert.



Der Coprozessor hat zwei eigene, von der CPU unabhängige Speicherbänke je 256Kx32bit. Diese Anordnung erlaubt parallele Operationen sowohl des Coprozessors und der CPU als auch innerhalb des Coprozessors, da das Daten-FPGA über die Adreß-FPGAs gleichzeitig auf beide Speicherbänke zugreifen kann. Um Platz auf den FPGAs zu sparen, wurden Hilfsschaltungen wie die eigentliche RAM Steuerung und eine Byte/Short-Steering-Logik außerhalb der FPGAs aufgebaut.

Die Anbindung an den Hostrechner wird von einem weiteren, fest programmierten FPGA (System Control) verwaltet. Dieses koordiniert neben dem gemeinsamen Zugriff von Host und Coprozessor auf die Speicher auch das Beladen der FPGAs mit Konfigurationsdaten.

Diese Daten, die den Coprozessor auf die aktuell zu bearbeitende Aufgabe spezialisieren, werden nicht direkt vom Host in die FPGAs geladen, da die relativ

langsame Ladegeschwindigkeit der FPGAs den Host zu lange aufhalten würden. Der Host überträgt stattdessen eine oder mehrere Konfigurationen schnell in eine weitere Speicherbank, aus dem dann das Controller-FPGA die vom Benutzer konfigurierbaren Bausteine belädt. Diese Gliederung erlaubt auch, daß sich der Coprozessor ohne Intervention des Hosts aus der Bibliothek der übertragenen Konfigurationsdaten selbst neu beladen kann. Beispielsweise können die unterschiedlichen Phasen eines Bilderkennungsalgorithmus vom Host als Einzelkonfigurationen zusammenhängend auf den Coprozessor übertragen werden. Nach dem Start des Coprozessors lädt das Controller-FPGA zunächst die erste Konfiguration und läßt diese ablaufen. Bei deren Ende leitet das Data-FPGA automatisch das Nachladen der Konfiguration der zweiten Phase des Algorithmus ein usw. Auf diese Weise kann die auf den FPGAs zur Verfügung stehende Rechenkapazität besser genutzt werden.

Da die maximale Taktrate eines FPGA-basierten Coprozessors in hohem Maß von der Komplexität der in den FPGAs implementierten Schaltung abhängt, ist die Taktfrequenz von 100 kHz - 80 MHz frei programmierbar und kann so optimal an die aktuelle Konfiguration angepaßt werden.

Einbindung in das Gesamtsystem

SPARXIL kommuniziert nicht direkt mit der CPU, sondern ist über den Peripheriebus des Hostrechners angeschlossen. Bei der als Hostrechner verwendeten SPARCstation findet die Kommunikation über den SBus statt. Der Coprozessor ist als Slave ausgeführt, der selbst keine Transaktionen einleiten kann, sondern von der CPU angesprochen werden muß. SPARXIL kann nur durch Auslösen eines Interrupts die Aufmerksamkeit des Hosts auf sich ziehen. Die Anbindung an den SBus ist aber hinreichend schnell, um auch Burst-Transfer des Hosts entgegenzunehmen.

Softwareschnittstelle

SPARXIL wird vom Host als Smart-Memory angesprochen. Die auf dem Coprozessor befindlichen Speicher werden mittels der Host-MMU in den virtuellen

Adreßraum des Benutzerprozesses ein-geblendet und können dann transparent innerhalb von Programmen manipuliert werden. Zur Steuerung des Coprozessors steht eine Bibliothek bereit, die Funktionen wie Start, Laden und Stoppen des Coprozessors sowie Statusabfragen bereitstellt.

Anwendungen

Für den Coprozessor wurde bisher eine Anwendung implementiert. Es handelt sich dabei um den Kern einer spezialisierten DES Bibliothek. Dabei wurde eine Beschleunigung um den Faktor 2.8 gegenüber einer handoptimierten Assemblerlösung erzielt. Dieser Wert ist aber nicht das Maximum der erzielbaren Beschleunigung, da die verwendeten CAD Werkzeuge (Xilinx XACT/XBLOX) nicht auf die Coprozessor-Architektur optimieren. Um diesem Zustand abzuhelpfen, entwickeln wir derzeit spezialisierte Werkzeuge, die bei der Übertragung eines Entwurfs in ein FPGA die Architektur des Gesamtsystems berücksichtigen.

Ausblick

Obwohl die bestehende SPARXIL-Architektur schon Leistungssteigerungen gegenüber der Host-CPU ermöglicht, bietet sie hinreichend Anhaltspunkte für Verbesserungen. So sind beispielsweise eine schnellere DVMA-Master-Anbindung an den SBus ebenso denkbar wie die Integration eines schnellen programmierbaren Kommunikationsnetzes zwischen FPGAs und Speicher. Da kompliziertere arithmetische Operationen auf FPGAs in der Regel nur ineffizient implementiert werden können, ist auch die Hinzunahme spezialisierter arithmetischer Einheiten wie Multiplizierer oder Memory-Mapped-FPUs vielversprechend (quasi als Co-Coprozessor).