

Anwendungsspezifisch optimierte Schaltungsimplementierung für XILINX-FPGA

Andreas Wassatsch; Hagen Ploog; Steffen Dolling; Dirk Timmermann

Einleitung

Durch die wachsenden Anforderungen an die schnelle Verfügbarkeit von Prototypen, die Minimierung der Realisierungskosten und die Anpassungsfähigkeit bestehender Verfahren für die Signalverarbeitung, rücken Implementationen der entsprechenden Algorithmen in FPGA-Bausteinen immer weiter in den Vordergrund. Dieses trifft ebenso für die Kleinserienproduktion zu, da sich der finanzielle und zeitliche Aufwand für die Umsetzung eines als Prototyp vorliegenden Designs, sei es in Form eines konventionellen Schaltungsaufbaus, als auch in Form einer FPGA-Entwicklungslösung, in einen anwendungsspezifischen Schaltkreis (ASIC) als nicht rentabel erweisen. Dieser Trend wird weiterhin durch das stetig verbesserte Preis/Leistungsverhältnis aktiv unterstützt. Die FPGA-Hersteller partizipieren dabei von der schnell fortschreitenden Entwicklung der Technologien zur Schaltkreisherstellung im Bereich der Mikroprozessoren.

Neben den FPGA-Reihen der Hersteller wie Actel, Altera, Atmel, Lucent und Motorola [1] zählt die 4000E/EX Reihe der Firma XILINX zu den führenden bezüglich der Einsatz- und Typenvielfalt.

Im weiteren soll gezeigt werden, wie durch eine geeignete Zerlegung des betrachteten Verarbeitungsalgorithmus eine leistungsfähige und zugleich kostengünstige Lösung erzielt werden kann.

1 Besonderheiten der XILINX 4000E/EX Serie

Nur durch die genaue Kenntnis der speziellen Komponenten jeder FPGA-Reihe ist eine optimale Abbildung der benötigten Schaltungsfunktionen möglich. Die in der 4000E/EX Reihe zu Verfügung stehende Grundfunktionalität besteht in einer als konfigurierbaren Logik Block (CLB) bezeichneten Zelle (Abbildung 1)[2]. Der CLB bietet drei frei belegbare Funktionsgeneratoren F, G und H, sowie Speicherelemente an. Die Generatoren F und G können jeweils aus den vier Eingangssignalen jede boolesche Verknüpfung erzeugen. Das gleiche gilt für den Generator H, jedoch nur mit drei Eingangsgrößen.

Die Auswahl der einzelnen Komponenten erfolgt während der den SRAM basierten FPGA's typischen Initialisierungsphase, in der die Konfiguration von einem geeigneten Bootmedium (ROM, μ P, ...) geladen wird.

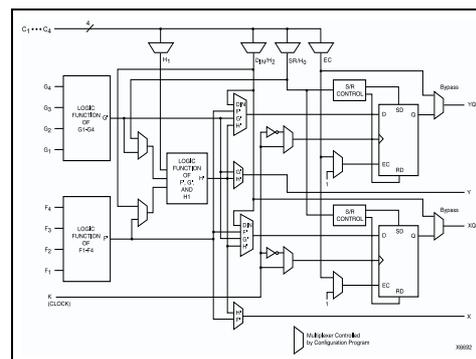


Abbildung 1: 4000E/EX CLB

1.1 Fast Carry Logic

Die CLB-Struktur der 4000E/EX erlaubt die Nutzung einer als Fast Carry Logic bezeichneten Konfiguration von Volladdierern, mittels der sich die auf einen Übertrag basierenden Addiererkonzepte, wie Ripple Carry (RC), aber auch Carry Save (CS), optimal abbilden lassen. Sie ist gekennzeichnet durch die Nutzung direkter Verbindungen zwischen den CLB's ohne den Umweg über die normalen Verbindungswege. Es muß hierbei erwähnt werden, daß sich der Ressourcenbedarf bei einer Implementierung eines CS-Addierers anstelle eines RC-Addierers nicht wie erwartet um den Faktor 2 erhöht. Architekturbedingt vergrößert sich dieser Faktor mindestens auf den Wert 4. Somit verbleibt die Anwendung einer CS-Schaltungsstruktur nur auf Geschwindigkeit orientierte Applikationen mit unkritischen Ressourcenbedarf beschränkt. Der erhöhte schaltungstechnische Aufwand für die in der konventionellen Schaltungswelt schnellere Konfiguration eines nichtredundanten Addierers, wie z.B. Carry-Select oder Carry-Skip, ist der wesentliche Grund, der den ebenso erwarteten Geschwindigkeitsvorteil je nach Konstellation in Richtung 0 reduziert. Ursache für dieses Verhalten ist das beschränkte Fassungsvermögen eines CLB's in Bezug auf die in ihm zu implementierende Logik. So können die benötigten Multiplexer nicht mit in die CLB's der Addierer gepackt werden. Daher ist das Carry-Out-Signal gezwungen, den CLB zu verlassen, und über die Interconnections den Weg in einen anderen CLB zu suchen. Dieses ergibt somit zusätzliche Verzögerungen, die den eigentlich erwarteten Geschwindigkeitsgewinn wieder aufzerren.

Schlußfolgernd kann festgehalten werden, für eine beliebige kombinatorische Schaltung ist es das Hauptziel bei der FPGA-Implementierung, den Signalpfad auf so wenig wie möglich Übergänge zwischen den CLB's zu reduzieren.

1.2 Pipelinebare Strukturen

Aus der Abbildung 1 der Struktur eines CLB's läßt sich erkennen, daß die zwei primären Ausgangssignale sowohl ungepuffert als auch über ein Speicherelement, daß wahlweise die Funktionalität eines Registers oder die eines Latches realisieren kann, zu den Ausgängen geführt werden können. Damit bietet sich sogleich die Möglichkeit der Realisierung gepipelinerter Verarbeitungsabläufe an. Die Positionierung der Pipelineregister kann wahlweise direkt in der Schaltungsbeschreibung erfolgen oder mittels geeigneter Software-Tools [3] nachträglich eingefügt werden. Für eine bessere Kontrolle der implementierten Pipelinestruktur ist die erste Variante zu bevorzugen. Nach erfolgreicher Realisierung der Grundfunktionalität der Verarbeitungspipeline kann auf dann auch ein Pipeline-Retiming, d.h. eine Neuordnung der Registerzeilen zur gleichmäßigen Verteilung der Laufzeiten im Signalpfad, angewendet werden, um so die Performance weiter zu verbessern.

1.3 RAM-Strukturen

Jeden CLB kann die Funktionalität eines RAM-Blockes zugewiesen werden, wobei verschiedene Konfigurationsmodi Anwendung finden können. So ist zum Beispiel die Realisierung eines begrenzten RAM-Bereiches für einen im FPGA implementierten Mikrocontroller [4] denkbar. Durch den Dual-Port-Mode ist auch eine Nutzung als FIFO möglich, mit dessen Anwendung im Vergleich zu einer Realisierung mit CLB-Registern ein enormes Optimierungspotential steckt.

1.4 Input/Output-Blocks

Auch der Funktionalität der Ein-/Ausgabe-Blöcke sollte bei der Implementierung Beachtung geschenkt werden. Wie in Abbildung 2 [2] der Grundstruktur eines IOB's der 4000E-Serie zu erkennen ist, stehen neben der optionalen Treibersstärke, Tri-State Ausgang und Pull-Up/-Down Widerstände auch zwei getrennte Speicherelemente zur Verfügung. Diese können so zum Beispiel für die Realisierung eines Ein- bzw. Ausgabe-Registers verwendet werden, womit ihre Implementation ohne Nutzung von CLB-Ressourcen möglich ist.

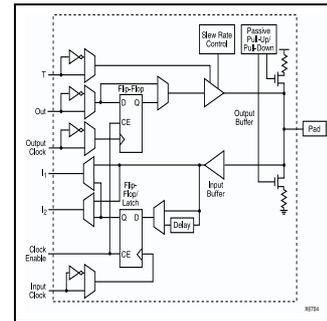


Abbildung 2: IOB 4000E

Features wie die separaten Takt- und Resetsignalnetze, die Möglichkeit der FPGA-internen Realisierung von Bussystemen mittels Tri-State-Treibern und die Wahl besonders schneller Verbindungswege für weit auseinanderliegende Schaltungsblöcke stehen dem Schaltungsdesigner ebenso zu Verfügung.

2 Anwendungsbeispiel: Implementierung des CORDIC-Verfahren

Bei der Entwicklung des CORDIC-Core [5] stand als primäres Designziel die Implementierung in einem möglichst kleinen und damit preiswerten FPGA-Baustein im Vordergrund. Somit fiel die Entscheidung bei der Wahl des Addiererkonzeptes auf eine RC-Implementation, da diese sich besonders ressourcenschonend in der gegebenen CLB-Struktur abbilden läßt. Zur Einhaltung der geforderten Taktrate war damit nur der Weg über eine FPGA-gerechte Beschreibung und Synthese möglich, um über die optimale Anordnung der CLB's zur Ausnutzung der Fast-Carry-Logik die Designvorgabe zu erreichen. Durch die Unempfindlichkeit der Zielapplikation gegenüber einer Latenzzeit für die Generierung der Ausgangsdaten, wurde eine gepipelinierte Abarbeitung des Verfahrens implementiert, wodurch zugleich die Einhaltung der Taktratenforderung als auch die Anforderungen bezüglich des Datendurchsatzes erfüllt werden konnte. Mit der Optimierung des zugrundeliegenden mathematischen Verfahrens war der Einsatz der FIFO-Konfiguration möglich (Abbildung 3), womit im Z-Pfad eine Einsparung von 52% gegenüber der Realisierung mittels der normalen CLB-Register-Implementierung erreicht wurde. Ein Retiming in den letzten Iterationsstufen führte zu der Einsparung von drei Pipelineregisterzeilen, womit zugleich neben dem Effekt der Einsparung von CLB-Registerressourcen auch die Latenzzeit um 18% verringert werden konnte. Die optionale Ausnutzung der Registerfunktionalität der IO-Blöcke bietet die Möglichkeit der Anpassung der Design-Cores an die jeweilige Schaltungsumgebung. Eine Einsparung von Verdrahtungsressourcen brachte die Verwendung des vorhandenen separaten Taktsignalnetzes und der völlige Verzicht auf die Rücksetzbarkeit des Designs. Dieses war auf Grund der reinen Pipelinestruktur des Cores möglich, da sich die Schaltungsstruktur designbedingt nach 14 Takten in einem definierten Zustand befindet.

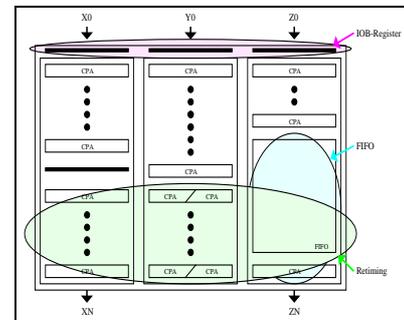


Abbildung 3: Cordic-Pipeline

Durch die beschriebenen Einsparungen konnte eine Reduzierung der Designgröße erreicht werden, die es ermöglicht eine Implementierung unter den Kosten eines

gleichfunktionalen ASIC-Bausteins bei gleichzeitig größerer Anpassungsfähigkeit zu erstellen.

3 Designdarstellung

Für die Beschreibung der Schaltungsimplementation wurde VHDL ((Very High Speed Integrated Circuits) Hardware Description Language) [3] verwendet, um so einen möglichst vom benutzten Designtool unabhängigen Beschreibungsstil verwenden zu können. Daher wurde auch für die FPGA-abhängige Implementierung von Grundstrukturen, wie die der RC-Addierer, ein FPGA-spezifisches Package geschrieben, und in der eigentlichen Designbeschreibung nur auf die so geschaffenen Konstrukte zurückgegriffen. Hiermit ist ein schneller Wechsel der Zielarchitektur, d.h. die Wahl einer anderen FPGA-Familie bzw. die Abbildung auf eine ASIC-Library, ohne Änderungen an der eigentlichen Funktionsbeschreibung möglich.

4 Zusammenfassung

Mit der in diesem Beitrag behandelten Anwendung konnte gezeigt werden, daß durch eine anwendungsspezifisch optimierte Schaltungsimplementierung der Einsatz von FPGA's im Bereich der Einzel- und Kleinserienfertigung wirtschaftlich interessant wird. Dieser Trend wird sich durch ständige Weiterentwicklungen durch die FPGA-Hersteller, wie die neue XILINX Spartan-FPGA-Familie [6], und der moderaten Preispolitik bei der benötigten Entwicklungssoftware fortsetzen. Hiermit eröffnet sich vor allem kleineren Unternehmen der Übergang zu einer zeitgemäßen Form der Produktentwicklung und damit zu der Ablösung kostenintensiver Entwicklungs- und Fertigungsstrategien.

Literatur

- [1] Marktübersicht: Programmierbare Logik-ICs. Markt&Technik, (32), 96.
- [2] XILINX: The Programmable Logic Data Book, 1996.
- [3] Synopsys: Synopsys Online Documentation Version 1997.08, 1997.
- [4] D.Dannhäuser: Inbetriebnahme und Untersuchungen zur optimalen Implementierung des RUN4-Mikrocontrollers in einem FPGA. Kleiner Beleg, 1998.
- [5] A.Wassatsch: CORDIC CORE mdC11V16S. Institut MD, 1997.
- [6] Den ASIC-Markt im Visier. Markt&Technik, (4), 98.

Verfasser

Dipl.-Ing. Andreas Wassatsch; Dipl.-Ing. Hagen Ploog;
Dipl.-Ing. Steffen Dolling; Prof.Dr. Dirk Timmermann
Universität Rostock; FB Elektrotechnik und Informationstechnik
Institut für Angewandte Mikroelektronik und Datentechnik
R.-Wagner Str. 31
18119 Rostock-Warnemünde
e-mail: wa11@e-technik.uni-rostock.de