

**Dateiname:** DA052\_Andeas\_V

**Titel:**

Optimierung eines PIC-kompatiblen VHDL-Mikrocontrollerkerns

**Bearbeiter:**

Volker Andreas

**Text der Kurzfassung:**

Gegenstand dieser Dokumentation ist die Optimierung eines PIC-kompatiblen-VHDL-Mikroprozessorkerns.

Die für diese Arbeit nötigen Grundkenntnisse werden im Kapitel 3 umrissen. Darin enthalten sind u. a. einige nützliche Tipps im Bezug auf die im Verlauf der Arbeit eingesetzten Software. Auch die Architektur des hier verwendeten FPGAs „Spartan-II“ der Firma Xilinx wird mit ihren wesentlichen Elementen grob umrissen. Den eigentlichen Einstieg in die Materie bildet die überschlägige Beschreibung der Signalverläufe des PIC-Mikrocontrollers. Ein weiterer Punkt ist die Vorstellung der sogenannten Timing und Grouping Constraints.

Im Anschluss an die Grundlagen ist die Erweiterung der Architektur um eine PIC-kompatible UART-Schnittstelle dargestellt. Neben einer ausführlichen Beschreibung der Funktionsweise der hierzu implementierten Module „Baudratengenerator“, „Sender“ und „Empfänger“ findet man in diesem Kapitel Erläuterungen darüber, wie diese Erweiterung in das bestehende Design eingebunden werden kann.

Die weiteren Punkte beschäftigen sich mit der Optimierung des PIC-Mikrocontrollers. In chronologischer Reihenfolge ist dokumentiert, wie im Einzelnen die Optimierung durchgeführt wurde. Den Anfang bildet die Umstellung des Designs auf durchgängige Verwendung der negativen Taktflanke zur Signalübernahme in den synchronen Elementen. Betroffen hiervon sind sämtliche Module (GPR, ROM und PC), die Block-RAM-Komponenten zur Informationsspeicherung verwenden. Während sich die Änderungen im GPR und ROM auf die Negierung des Taktsignals beschränkt, muss im PC der gesamte Stackzugriff neu organisiert werden.

Im nächsten Punkt wird die ALU grundlegend verändert. Die für die arithmetischen und logischen Funktionen zuständigen Schaltungsteile werden weitestgehend gruppiert und damit die Anzahl paralleler Baugruppen reduziert. Der Effekt hiervon ist, dass sich damit die Größe des Ausgangsmultiplexers drastisch verkleinern lässt und man letztendlich Hardwareressourcen einspart. Des Weiteren werden diese Funktionsgruppen an sich daraufhin überprüft, ob sie auch optimal zu der internen Struktur des Spartan-II passen. Dieser realisiert logische Funktionen hardwaremäßig durch die Verwendung von Look Up Tables mit vier Eingangssignalen. Aus diesem Grund können entsprechend entworfene Logikelemente (logische Funktionen, die nicht mehr als vier Eingänge benötigen) sehr effektiv von den Xilinx-Implementierungstools umgesetzt werden. Im Verlauf der Optimierung werden zwei unterschiedliche Varianten

ten der verwendeten arithmetischen Einheit vorgestellt. Die eine setzt sich aus vorgefertigten Xilinx-Bibliothekskomponenten (ADSU4) zusammen, die andere besteht aus einer reinen VHDL-Lösung.

Der Einsatz der renovierten ALU setzt entsprechende Steuersignale voraus. Um den Decodieraufwand so gering wie möglich zu halten werden die Befehlswoorte des PIC-Wortschatzes auf signifikante Eigenschaften hin untersucht, die ein explizites Decodieren unnötig machen. In den Fällen, in denen ein solches Vorgehen nicht zum Erfolg führt, wird das Befehlswort mit geringst möglicher Breite in minimaler Form (durch Minimierung nach Karnaugh) decodiert.

Durch die Verbesserungsschritte hervorgerufene Steigerungen der maximalen Taktfrequenzen zeigten noch ein weiteres Problem auf. Der Watchdog Timer war nicht länger in der Lage den Taktimpulsen zu folgen. Durch kleine Eingriffe in dessen Struktur konnte allerdings dessen maximale Arbeitsgeschwindigkeit hinreichend gesteigert werden.

Daran schließt sich die Einführung von Timing und Grouping Constraints in das Design an. Vorteilhaft an der Vergabe von zeitlichen Randbedingungen ist, dass man damit den Place&Route-Prozess in die Lage versetzt, zeitlich besonders kritische Signalpfade bevorzugt zu behandeln. Die so erreichbaren Ergebnisse sind oft um ein vielfaches besser als die herkömmlich erzielten. Des weiteren lassen sich zeitliche Analysen sehr viel komfortabler durchführen, da durch die Definition von Multi-Cycle Paths und False Paths der Fokus sofort auf die Signale fällt, die zu Verletzungen der zeitlichen Rahmenbedingungen (Setupzeiten u. ä.) führen.

Die hierbei erreichten Einsparungen an Hardwareressourcen und möglichen Steigerungen der Taktfrequenz sind zum Abschluss dieser Arbeit dokumentiert und vergleichend zusammengefasst.