

Vorlesung Architekturen und Entwurf von Rechnersystemen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Lösungsvorschlag

Prof. Andreas Koch, Yannick Lavan, Johannes Wirth, Mihaela Damian

Wintersemester 2022/2023
Theorieblatt 3

In dieser Übung wird das Zynq 7000 Reconfigurable System-on-Chip (rSoC) genauer betrachtet.

3.1 Zynq Hardware

Betrachten Sie die Application Processor Unit des Zynq 7000 rSoC (vgl. SoC Foliensatz, Folie 14). Erläutern Sie kurz die Funktion der folgenden Komponenten im Zynq rSoC:

- GIC
- DMAC
- OCM
- SCU

Hinweis: Sie finden genaue Informationen zu jeder Komponente im User Guide des Zynq rSoCs¹.

Lösungsvorschlag:

Der GIC (Generic Interrupt Controller) organisiert die Interrupt Signale zwischen Processing System (PS) und Programmable Logic (PL). Hierbei werden Private Peripheral Interrupts (PPI), Software Generated Interrupts (SGI) und Shared Peripheral Interrupts (SPI) unterschieden. Bei den PPIs handelt es sich um jeweils fünf eigene Interrupts pro CPU, d.h. Interrupts werden immer zu einer bestimmten CPU weitergeleitet. SGIs können verwendet werden, wenn eine CPU einen Interrupt an sich selbst, die andere CPU oder beide gleichzeitig schicken möchte. SPIs können zu einer beliebigen oder beiden CPUs weitergeleitet werden. Die Aufteilung erfolgt vor allem durch den GIC.

Der DMAC (Direct Memory Access Controller) wird für größere Datentransfers zwischen PL und Systemspeicher verwendet. Auf dem Zynq SoC wird hierfür ein 64-bit AXI Master verwendet. Auf dem Zynq SoC können acht parallele Direct Memory Access (DMA) Kanäle betrieben werden. DMA Transfers finden unabhängig von der aktuellen Prozessor-Instruktion statt, wodurch ein höherer Durchsatz möglich ist.

Der On-Chip Memory (OCM) besteht aus 256 kB Random-Access Memory (RAM) und 128 kB Read-Only Memory (ROM) und zwei 64-bit AXI Slave Interfaces. Ein Slave ist für CPU Zugriffe reserviert und der andere wird von allen Mastern in PS und PL geteilt. Der ROM ist für User nicht sichtbar.

Die SCU (Snoop Control Unit) sorgt für die Cache Kohärenz zwischen den Prozessoren und dem L2-Cache.

3.2 DSPs in FPGAs

In der PL des Zynq rSoC befindet sich ein Field-Programmable Gate Array (FPGA). Dort sind Digital Signal Processors (DSPs) verbaut.

- Für welche Zwecke können DSPs genutzt werden?
- Welche Vor- und Nachteile stehen sich bei der Umsetzung mit DSPs und der Umsetzung mit Configurable Logic

¹Zynq 7000 Technical Reference Manual: <https://docs.xilinx.com/v/u/en-US/ug585-Zynq-7000-TRM>

Blocks (CLBs) gegenüber?

Lösungsvorschlag:

- a) Arithmetische Operationen können auf DSPs ausgelagert werden.
- b) CLBs haben den Vorteil, dass man in der Bitbreite sehr variabel ist. Dennoch sorgt die Umsetzung durch Lookup Tables (LUTs) häufig für lange kritische Pfade, wodurch die maximale Taktfrequenz sinkt. DSPs bestehen aus fest verdrahteten Komponenten und können dadurch mit einer höheren Frequenz arbeiten, es wird aber zusätzliche Hardware benötigt, um die Bitbreiten in den CLBs kompatibel zu den Bitbreiten der verwendeten DSPs zu halten. Da die DSPs aber schon auf dem FPGA vorhanden sind, müssen diese nicht erst noch realisiert werden, wodurch mehr CLBs für andere Zwecke genutzt werden könnten.

3.3 Soft Core vs. Hard Core

- a) Wie entsteht der Unterschied in Chipfläche und maximaler Taktfrequenz zwischen Soft Core Prozessoren und Hard Core Prozessoren?
- b) In welchen Szenarien könnte der Einsatz von Softcore Prozessoren trotzdem sinnvoll sein?

Lösungsvorschlag:

- a) Soft Core Prozessoren werden auf FPGAs gebaut. Das heißt sie realisieren die Prozessor-Instruktionen durch CLBs und DSPs. Durch die Verwendung von DSPs müssen bestimmte Pfade verwendet werden, um eine korrekte Funktion zu gewährleisten. Zusätzlich hierzu muss die restliche Funktionalität durch eine geschickte Verwendung von CLBs realisiert werden. Hierdurch entsteht ein hoher Flächenverbrauch und langer kritischer Pfad. Hard Core Prozessoren sind festverdrahtet auf einem Silizium Wafer gebaut. Somit können direkt einzelne Gatter/Module verwendet werden, deren Anordnung nochmal in Hinsicht der Flächennutzung und der kritischen Pfade optimiert werden kann. Dadurch entsteht ein kleinerer Flächenverbrauch und ein kurzer kritischer Pfad.
- b) Soft Core Prozessoren können zur Laufzeit ausgewechselt werden. Das macht Sie interessant, wenn man verschiedene Berechnungen auslagern möchte und diese Berechnungen aus einem breiten Anwendungsspektrum kommen. So könnte man z.B. einen Prozessor mit Floating-Point Unit (FPU) ersetzen durch einen Prozessor ohne FPU.

3.4 Empfohlene Bonusaufgabe: Realisierung eines Hardware-Beschleunigers auf dem Zynq rSoC

Sie möchten einen Hardware-Beschleuniger für die Erstellung von Negativ Bildern[1] realisieren. Für die Verwendung des Beschleunigers ist folgender Ablauf vorgesehen:

Ein Bild liegt als Datei auf einem Speichermedium (z.B. SD-Karte). Der User startet ein Programm und gibt dabei den Dateipfad für das gewünschte Bild und einen Dateinamen für das invertierte Bild an. Das Programm soll das gewünschte Bild laden und mit Hilfe des Hardware-Beschleunigers konvertieren. Nachdem die Konvertierung abgeschlossen ist, soll das Bild unter dem angegebenen Dateinamen auf dem Datenträger gespeichert werden. Überlegen Sie sich, wie diese Aufgabe auf dem Zynq SoC umgesetzt werden könnte.

Hinweis: Sie müssen hier keinen Code schreiben. Skizzieren Sie lediglich grob Ihre Ideen. Überlegen Sie zunächst, welche Komponenten im PS (auf den Cortex-A9) und in der PL (FPGA) agieren und wie die Komponenten miteinander interagieren.

Lösungsvorschlag:

Die zwei wesentlichen Komponenten sind die Host-Software im PS und spezialisierte Hardware zur Farbumkehrung in der PL. Die Host-Software lädt das gewünschte Bild in den Speicher und überträgt dann das Bild per AXI4-Lite (langsam) oder über AXI4 und DMA (siehe Aufgabe 1; das PS ist in diesem Fall ein AXI4 Slave) zur PL. In der PL sitzt ein AXI4(-Lite) fähiges Modul, welches die einzelnen Pixel aus dem Speicher entgegennimmt und die Invertierungsoperation ausführt. Anschließend schreibt der Beschleuniger die invertierten Pixel zurück in den Host-Speicher und signalisiert das Ende der Verarbeitung an den Host. Dies könnte durch einen Interrupt oder ein Statusregister, das vom Host gelesen wird, realisiert werden. Im Anschluss liest die Host-Software das Bild aus dem Speicher und schreibt es in eine Datei. Hinweis: In der Realität wird häufig noch ein Treiber im PS verwendet, um den Hardware-Aspekt für den Software-Entwickler wegzuabstrahieren. In diesem einfachen Anwendungsszenario ist aber kein Treiber nötig.

Wichtige Abkürzungen

CLB	Configurable Logic Block
DMA	Direct Memory Access
DSP	Digital Signal Processor
FPGA	Field-Programmable Gate Array
FPU	Floating-Point Unit
LUT	Lookup Table
PL	Programmable Logic
PS	Processing System
RAM	Random-Access Memory
ROM	Read-Only Memory
rSoC	Reconfigurable System-on-Chip

Literatur

[1] [https://de.wikipedia.org/wiki/Invertieren_\(Bildbearbeitung\)](https://de.wikipedia.org/wiki/Invertieren_(Bildbearbeitung)).