

Konfigurierbare eingebettete Systeme

Beuth Hochschule - Wintersemester 2018/19

Laborübung 1

Gruppe 15:

Omid Rahimian Mashhadi Mat.Nr.: 872958

Torsten Michael Schenk Mat.Nr.: 838995

Abgabedatum: 12.11.2018

Seitenanzahl: 6

Inhaltsverzeichnis

1	Vorwort	3
2	Aufgabe 1	3
2.1	Vorbereitung zur Laborübung	3
3	Aufgabe 2	4
3.1	Beschreibung der Funktion XGpio_DiscreteWrite	4
3.2	Woran/Wo sieht man die Antwort von der Vorbereitung im Code?	4
4	Aufgabe 3	5
5	Glossar	5

1 Vorwort

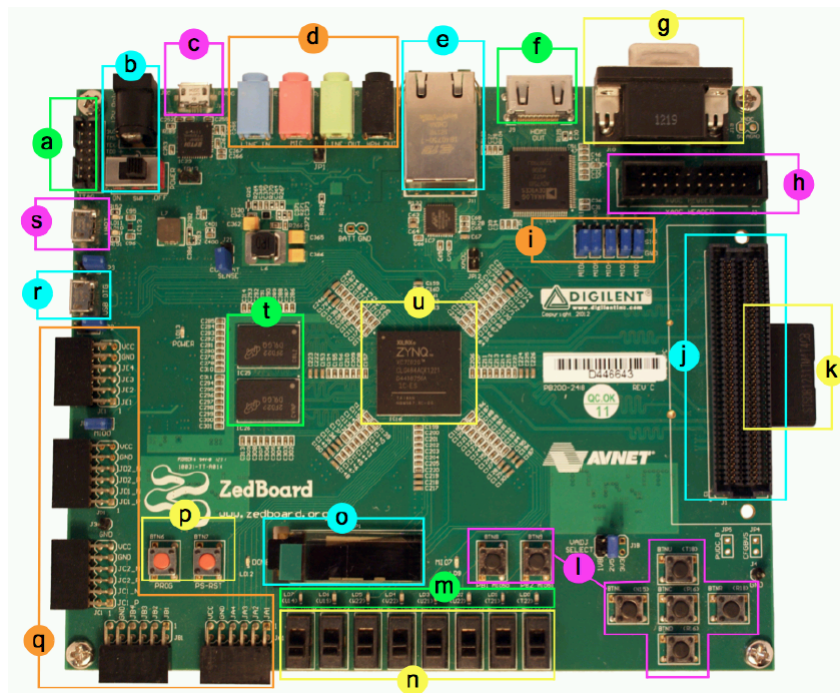
Bei der Recherche zur Bearbeitung der Übungen wurden viele englischsprachige Webseiten zu rate gezogen. Generell kann man sagen, dass englische Fachbegriffe sich im Bereich FPGA und embedded Design etabliert haben, so dass eine Übersetzung eher verwirren als helfen würde. Daher haben wir uns entschieden, die **englischen** Bezeichner und Beschreibungen beizubehalten.

Um Codeabschnitte besser von Beschreibungen besser unterscheiden zu können, wurde eine eigene Schriftart verwendet:

Kommandozeilen Eingaben und Codesnippets werden wie HIER dargestellt.

2 Aufgabe 1

In der Laborübung wurde das ZedBoard Zynq-7000 eingesetzt. Es umfasst als **PL** den Artix-7 FPGA mit 85K Logic Cells (Device Z-7020, Part: XC7Z020) und als **PS** den Dual-core ARM Cortex-A9 MPCore™ mit 866 MHz.



ZedBoard mit Xilinx Zynq-7000 SoC

Wichtigste Anschlüsse für Laborübung 1

- b) Power Supply
- c) USB-JTAG (programming)

2.1 Vorbereitung zur Laborübung

Memory Mapped I/O und isolated I/O sind zwei Methoden um Input-Output Operationen zwischen CPU und der Peripherie auszuführen.

Memory Mapped I/O

Es wird der gleiche Adressbus verwendet, um den primären Speicher und den Speicher der Hardwaregeräte anzusteuern. Das bedeutet, die Befehle um bestimmte Bereiche im Speicher anzusprechen, können ebenfalls verwendet werden um die Speicherbereiche der Hardware anzusprechen.

Isolated I/O

Auf der anderen Seite, verwendet Isolated I/O separate Befehle um den primären und den Gerätespeicher anzusprechen. In einem solchen Fall liegen zwei separate Adressbereiche vor. Dies können z.B. separate I/O Pins an der CPU oder ein kompletter eigener Bus sein. Da der primäre Speicherbereich von dem Geräteadressbereich getrennt ist, spricht man von Isolated I/O. Isolated IO benötigt spezielle Befehle um zu schreiben und zu lesen.

3 Aufgabe 2

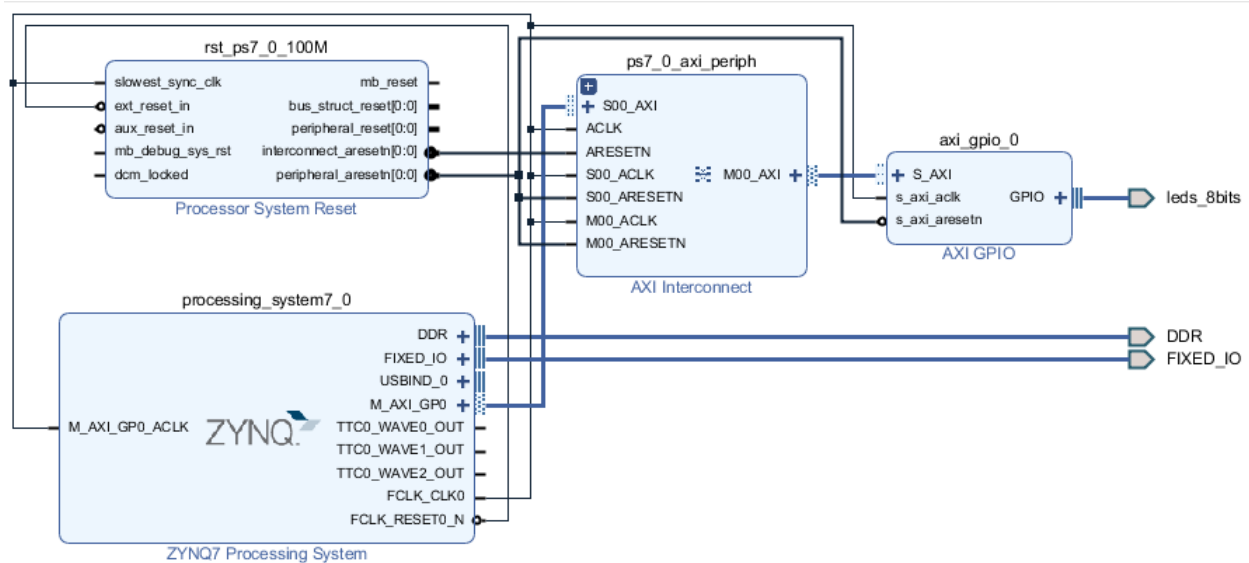
Konfiguration des IP Cores zur Realisierung der Kommunikation zwischen Prozessor und Peripherie als Blockdesign in Vivado 2016.2. Eingebunden in diese wurden zusätzlich die 8Bit-LED-Anzeige und anschließend der Aufbau in VHDL synthetisiert.

3.1 Beschreibung der Funktion XGpio_DiscreteWrite

3 Checks ob Pointer valide, Status Ready, welcher Channel aktiv ist und ob zwei Kanäle von der Hardware unterstützt werden. Danach werden Basispointer und Adressoffset addiert und an XGpio_WriteReg übergeben. Kanäle (Registerbänke) liegen im Speicher nacheinander, ein Umschalten erfolgt über einen Adressoffset.

3.2 Woran/Wo sieht man die Antwort von der Vorbereitung im Code?

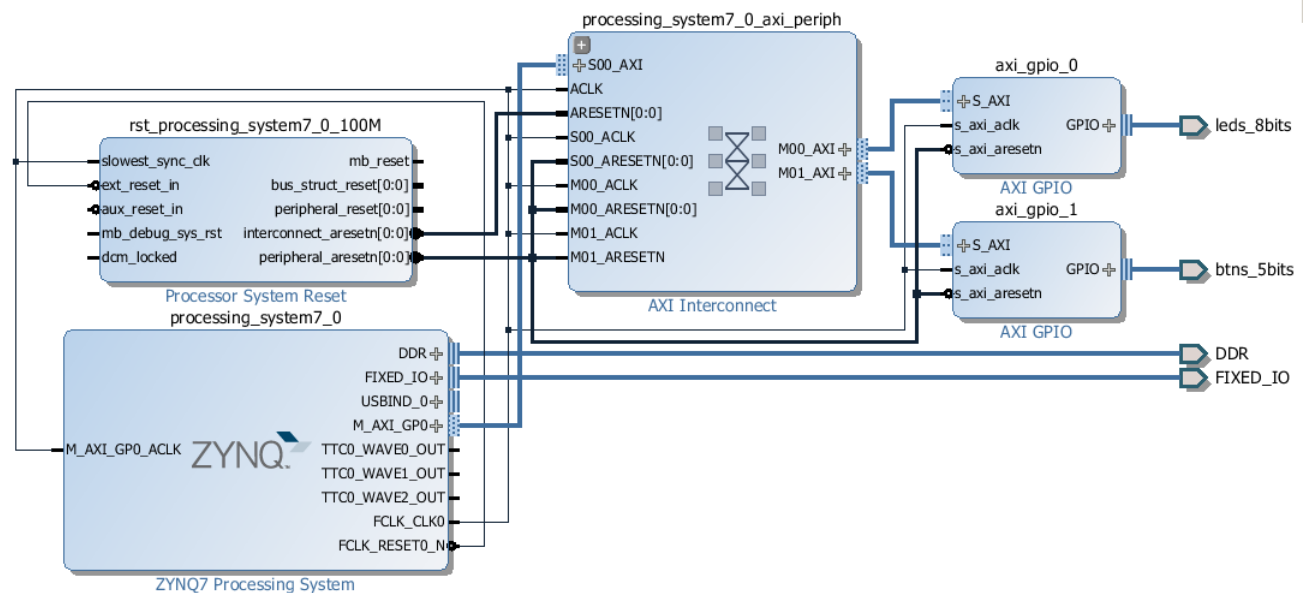
In der Funktion XGPIO_Initialize wird über die DeviceID ein Instanz-Zeiger (InstancePtr) erzeugt und initialisiert. Dieser Instanz-Zeiger wird später verwendet, um die genaue Adresse zu berechnen. Die Offsets können zum konfigurieren der I/Os als (Ein- oder Ausgang) und zum setzen der einzelnen Bits verwendet werden.



Vivado: Schemaplan der IP-Cores

4 Aufgabe 3

Konfiguration des IP Cores zur Realisierung der Kommunikation zwischen Prozessor und Peripherie als Blockdesign in Vivado 2016.2. Eingebunden in diese wurden zusätzlich die 8Bit-LED-Anzeige und die Push-Buttons. Der Aufbau wurde anschließend in VHDL synthetisiert.



Vivado: Schemaplan der IP-Cores

5 Glossar

Beschreibung der wichtigsten Abkürzungen, die in der Übung verwendet werden.

Glossar

PS Processing System, z.B. PS memory

PL Programmable Logic, z.B. PL memory

AXI Advanced eXtensible Interface as protocol for Intellectual Property (IP) cores