

4. Übung: Excursion – Xilinx Zynq SoC XC7Z007S

Name(n):

Wie sich bei der Diskussion der 3. Übung herausstellte, müssen bestimmte Eigenschaften der Zielpattform bei der Entwicklung des VP (Virtual Prototype) berücksichtigt werden, damit die Firmware später ohne großen Aufwand auf das Zielsystem portiert werden kann. Daher soll in dieser Übung ein SoC (System-on-a-Chip) basierend auf der IP-Bibliothek von Xilinx, der späteren Zielpattform in die der Cordic-Core eingebunden werden soll, erstellt werden, um sich mit dem System vertraut zu machen.

1 System-on-a-Chip-Design

Mit der Zynq-Reihe stellt Xilinx eine leistungsfähige Plattform, bestehend aus HPS (ARM Cortex-A9) und FPGA [Xil18] zur Verfügung. Die hier verwendete Hardwareplattform Minized¹ ist mit dem XC7Z007S bestückt, welcher nur einen ARM-Prozessor enthält – nähere Informationen zu den Eigenschaften finden sich unter [Avn17a].

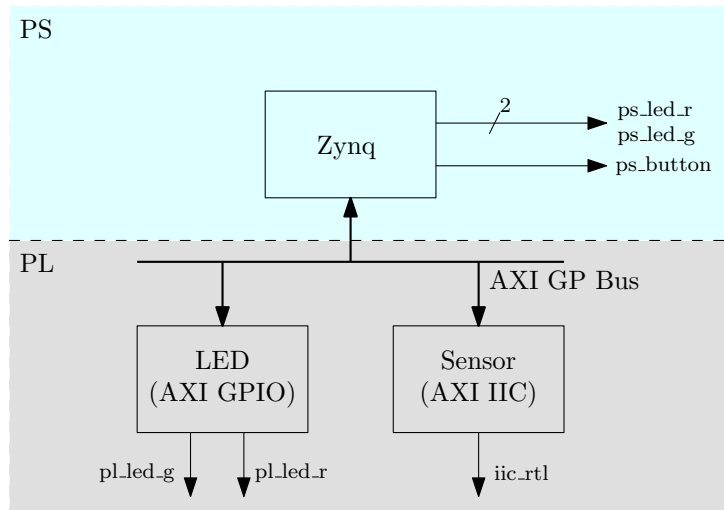
Neben dem Prozessorsubsystem beinhaltet die IP-Bibliothek von Xilinx noch eine Vielzahl weiterer IP-Blöcke, mit denen man auf einfache Weise ein dynamisches embedded System entwickeln kann.

Erstellen wie unten dargestellt basierend auf dem Minized ein System-on-a-Chip, dass folgende Komponenten enthält. Die LED und der Button des Prozessor Systems (PS) sind fix verdrahtet und müssen beim Hardwareentwurf nicht mehr berücksichtigt werden:

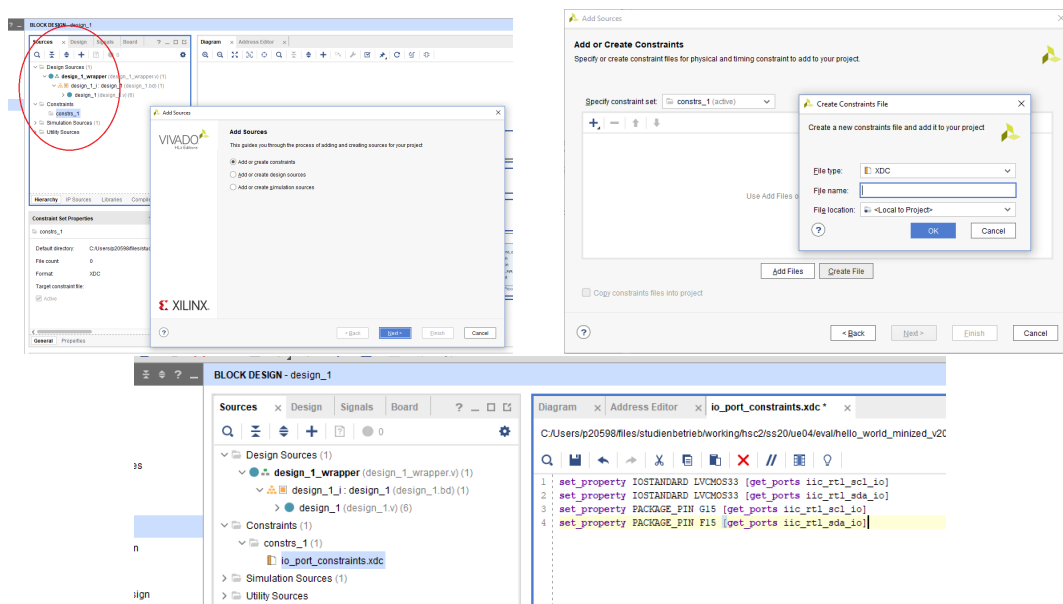
- Plattform: MiniZed (xc7z007sclg225-1).
- Zynq7 Processing System: M AXI GP0 Interface.
- LED (GPIO): axi_gpio, 2-Ports, all output.
- Sensor (I²C): axi_iic (ohne Interrupt).

Die Verbindungen und fehlenden Standard-IP-Blöcke können mit den Design-Assistenten *Run Block Automation* bzw. *Run Connection Automation* hinzugefügt werden. Beachten Sie dass der Design Assistent die Ports erst dann automatisch verbinden kann, wenn diese in der jeweiligen IP aktiviert wurden – AXI GP Interface des Zynq ist standardmäßig deaktiviert. Generelle Hinweise zur Erstellung einer Hardwareplattform basierend auf das Board Minized mit Vivado, bzw. generelles zur Verwendung zum Tool Vivado findet sich in [Avn18, Xil19]

¹www.minized.org



Anmerkung: Die Verbindung des I²C-Controllers mit den entsprechenden Pins des FPGA ist in den Pre-Settings des Boards nicht definiert und muss manuell gemacht werden. Dazu gibt es zwei Möglichkeiten, entweder man fügt unter Source → Constraints ein XDC-File hinzu bzw. erweitert ein bestehendes.

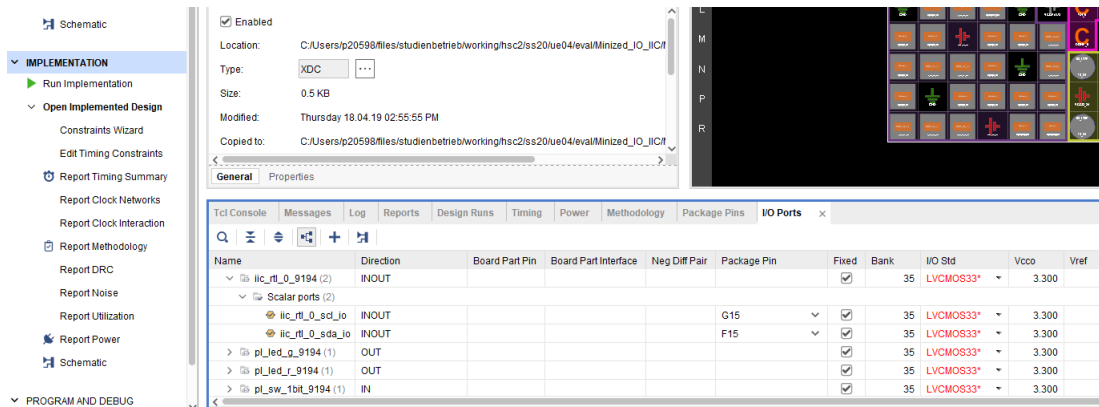


Achtung: der Name der Ports hängt davon ab wie diese im Block Design benannt wurden. Beispielhafte Definition der Spannungspegel und Pin-Verbindung:

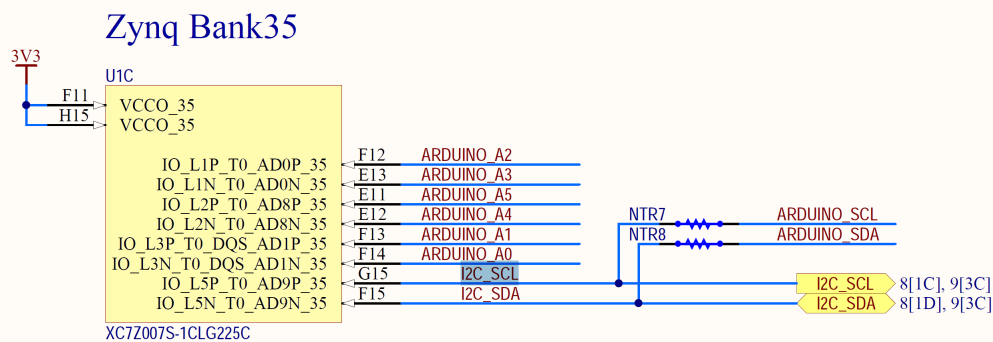
```

1 set_property IOSTANDARD LVCMOS33 [get_ports iic_rtl_scl_io]
2 set_property IOSTANDARD LVCMOS33 [get_ports iic_rtl_sda_io]
3 set_property PACKAGE_PIN G15 [get_ports iic_rtl_scl_io]
4 set_property PACKAGE_PIN F15 [get_ports iic_rtl_sda_io]
  
```

Eine andere Möglichkeit ist die Verwendung des Fensters *IO-Ports*. Hierzu im Flow Navigator den Bereich *Implementation* aktivieren und im Menüpunkt *Window* → *I/O Ports* das Fenster einschalten. Nun können die Einstellungen im unteren Bereich der IDE getroffen werden.



Die Pins an denen der Sensor-IC LIS2DS12TR an den FPGA angeschlossen ist kann der Schematic entnommen werden [Avn17b, S.7].



Generieren Sie den Bit-Stream und exportieren Sie die Plattform für die Verwendung im Xilinx-SDK.

2 Softwareentwicklung

Im Xilinx SDK kann nun basierend auf den Daten die aus Vivado exportiert wurden eine *Hardware Platform Specification* und aufbauen auf diesem ein *BSP* generiert werden, welches die Treiber der Peripherie enthält und mit deren Hilfe eine schnelle Entwicklung einer beliebigen Applikation möglich ist.

2.1 LED Counter

Programmieren Sie mit Hilfe der zweifarbigen LEDs einen Binärzähler der mit 2 Hz zählt. Das Timing soll dabei durch Interrupts gesteuert werden die mittels *Private Timer* des ARM Prozessors erzeugt werden.

2.2 Beschleunigungs- und Temperatursensor

Mittels I²C-Schnittstelle soll der Beschleunigungs- und Temperatursensor alle 2s ausgelesen werden und die Daten mittels `printf(...)` über den UART ausgegeben werden.

Hilfestellungen für die Implementierung (Archiv *SourceFiles*) und nützliche Dokumente finden Sie im Elearning im Odner Unterlagen.

3 Abgabe

Die Abgabe soll folgendes beinhalten:

- Vivado Projekt als Archiv (Project → Archive...)
- SDK
 - Hardware Platform Specification
 - BSP
 - Application

Die Abgabe kann auf mehrere Dateien aufgeteilt werden, wobei alle Dateinamen der üblichen Namenskonvention folgen sollen.

Literatur

[Avn17a] Avnet. *MiniZed Hardware User Guide*, June 2017.

[Avn17b] Avnet. *MiniZed Zynq Development Board*, June 2017.

[Avn18] Avnet. *MiniZed: Creating a Zynq Hardware Platform in Vivado*, June 2018.

[Xil18] Xilinx. *Zynq-7000 SoC Technical Reference Manual*, July 2018.

[Xil19] Xilinx. *Vivado Design Suite Tutorial – Embedded Processor Hardware Design*, June 2019.