

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 [Xil18b] zur Verfügung. Die hier verwendete Hardwareplattform Minized¹ ist mit dem XC7Z007S bestückt, welcher nur einen ARM-Prozessor enthält – nähere Informationen zu dem Eigenschaften finden sich unter [Avn17a].

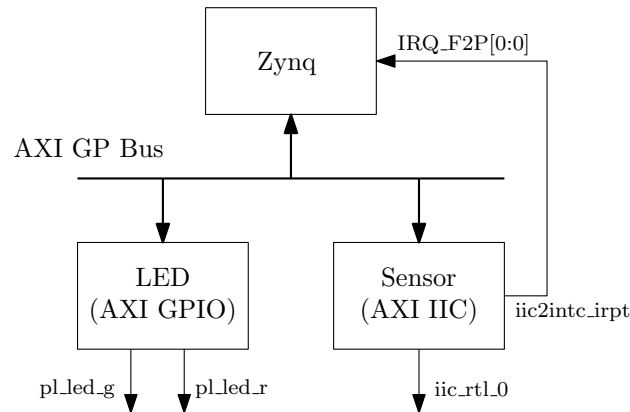
Neben dem Prozessorsubsystem beinhaltet die IP-Bibliothek von Xilinx noch eine Vielzahl weiterer IP-Blöcke, mit denen man in einfacher Weise ein dynamisches embedded System entwickeln kann.

Erstellen Sie basierend auf dem Minized ein, wie in untenstehender Graphik dargestelltes System-on-a-Chip, dass folgende Komponenten enthält:

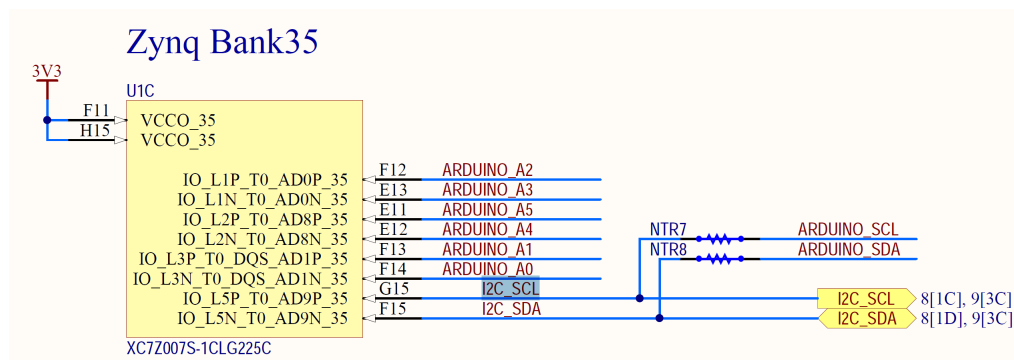
- Plattform: MiniZed (xc7z007sclg225-1).
- Zynq7 Processing System: M AXI GP0 Interface, Fabric Interrupts (PL-PS) IRQ_F2P.
- LED (GPIO): axi_gpio, 2-Ports, all output.
- Sensor (I²C): axi_iic.

Die Verbindungen und fehlenden Standard-IP-Blöcke können mit den Design-Assistenten *Run Block Automation* und *Run Connection Automation* hinzugefügt werden. Hinweise zur Erstellung einer Hardwareplattform basierend auf das Board Minized mit Vivado, bzw. generelles zur Verwendung zum Tool Vivado findet sich in [Avn18, Xil18a]

¹www.minized.org



Hinweis: Der Sensor-IC LIS2DS12TR ist wie folgt angeschlossen [Avn17b]. Nachdem die Implementation erstellt wurde, kann via *Window*→*I/O Ports* die Zuweisung der Pins und Spannungspegel gemacht werden.



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 der HW-Plattform 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 mehrfarbigen LEDs einen Binärzähler der mit 2 Hz zählt. Das Timing soll dabei durch Interrupts gesteuert 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(...)` ausgegeben werden.

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

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.
- [Lat12a] Lattice Semiconductor Corporation. *LatticeMico32 Timer*, December 2012.
- [Lat12b] Lattice Semiconductor Corporation. *LatticeMico32 UART*, December 2012.
- [Lat14a] Lattice Semiconductor Corporation. *LatticeMico32 Software Developer User Guide*, May 2014.
- [Lat14b] Lattice Semiconductor Corporation. *LatticeMico32 Tutorial*, May 2014.
- [Xil18a] Xilinx. *Vivado Design Suite Tutorial – Embedded Processor Hardware Design*, December 2018.
- [Xil18b] Xilinx. *Zynq-7000 SoC Technical Reference Manual*, July 2018.