

Aufgabe 1

Lernziele: Machen Sie sich mit dem Soft-Core-Prozessor PicoBlaze von Xilinx und mit seiner Entwicklungsumgebung (Synthesesoftware + Entwicklungsboard) vertraut. Der PicoBlaze ist vollständig in der VHDL-Datei `kcpsm3.vhd` beschrieben und wird in einem Entwurf wie jede andere VHDL-Komponenten durch Instanziierung eingesetzt. Eine synthesesfähige Systembeschreibung, bestehend aus dem PicoBlaze, einem RAM-Block sowie Peripheriekomponenten, steht zur Verfügung. Die Programmierung des Soft-Core-Prozessors erfolgt in seiner Assemblersprache. Ein Assemblerprogramm wird mit dem Programm `KCPSM3.EXE` in eine binäre Version übersetzt, die anschließend zusammen mit der Systembeschreibung für den FPGA-Baustein Spartan3 synthetisiert wird.

Neben dem Umgang mit dem PicoBlaze und seiner Entwicklungsumgebung lernen Sie in dieser Aufgabe auch:

- interrupt-gesteuerte Algorithmen in der Assemblersprache des PicoBlaze zu implementieren,
- einfache Ein-/Ausgabe-Operationen über die IO-Schnittstelle des PicoBlaze durchzuführen und
- Zustandsmaschinen im Assembler zu implementieren.

Als Vorbereitung zur Lösung dieser Aufgabe machen Sie sich mit den Dokumenten *KCPSM3 8-Bit Micro Controller* (`kcpsm3_manual.pdf`) und *PicoBlaze 8-bit Embedded Microcontroller User Guide* (`ug129.pdf`) vertraut.

Systemaufbau. Das Zielsystem setzt sich aus einem Laborrechner (Standard-PC mit einer virtuellen Maschine unter Windows XP) und einem Entwicklungsboard (*Spartan-3 Starter Kit Board*, `ug130.pdf`) von Xilinx zusammen. Der PC wird mit dem Entwicklungsboard einerseits über einen Download-Adapter zwecks der Konfiguration des FPGA-Bausteins und andererseits über ein serielles Datenkabel zwecks des Datenaustausches mit der zu entwickelnden Applikation verbunden. Der FPGA-Baustein wird mit 50 MHz getaktet. Der PicoBlaze braucht immer zwei Takte pro Maschinenbefehl, somit erreicht er bei dieser Taktfrequenz einen maximalen Durchsatz von 25 MIPS. Damit der PicoBlaze im Rahmen dieser Aufgabe serielle Daten korrekt erfassen und ausgeben kann, braucht er einen Timer mit einer vorgegebenen Frequenz. Diese Frequenz beträgt 153,6 kHz und ergibt sich aus dem 16fachen Oversampling der Übertragungsgeschwindigkeit von 9600 Bit/Sek. Der Timer selbst beruht auf einem freilaufenden Frequenzteiler, der bereits in VHDL implementiert und in die Systembeschreibung mit dem PicoBlaze integriert ist. Der Timer erzeugt eine Impulsfolge, die direkt über den Interrupt-Eingang an den PicoBlaze weitergeleitet wird. Die Zeitspanne zwischen zwei Impulsen beträgt 6,51 μ s und stellt somit einen Software-Entwickler vor der Herausforderung, eine Interrupt-Service-Routine im Assembler so effizient zu programmieren, dass sie bis zum Auftreten des nächsten Interrupt-Impulses fertig ist. Denn in dieser Zeitspanne können bis zu 160 Maschinenbefehle ausgeführt werden.

Serielle Datenübertragung. Die asynchrone, bitserielle Datenübertragung zeichnet sich sowohl in Empfangs- als auch Senderichtung durch folgende Merkmale aus:

- Übertragungsgeschwindigkeit: 9600 Bit/Sek mit 16fachen Oversampling
- Übertragungsrahmen: 1 Startbit + 8 Datenbits + 1 Stoppbit (ohne Paritätsbit). Das Startbit wird durch eine 0 und das Stoppbit eine 1 dargestellt. Das LSB (Bit 0) wird immer als erstes Bit und das MSB (Bit 7) als letztes Bit empfangen oder gesendet.
- Eine 0 wird durch Low-Pegel, und eine 1 durch High-Pegel auf der Datenleitung repräsentiert.
- Wenn keine Daten gesendet oder empfangen werden, befindet sich die Datenleitung im Idle-Zustand, also hat sie den High-Pegel.

Auf das Prinzip der asynchronen, bitseriellen Datenübertragung wird in der Übungsstunde detailliert eingegangen.

Applikation. Die Applikation hat die Aufgabe, in Echtzeit

- den über den seriellen Dateneingang RXD bitweise ankommenden Datenstrom vom PC zu erfassen,
- Start- und Stoppbits im Datenstrom zu erkennen,
- die einzelnen Datenbits zu Bytes zusammenzufassen,
- die Bytes auf ihren Inhalt hin zu analysieren, und zwar alle Bytes, die keine Ziffer und keine Buchstaben nach ASCII sind, herauszufiltern,
- die so erfassten Ziffer und Buchstaben um ein Start- und ein Stoppbit zu einem Übertragungsrahmen zu erweitern,
- den Übertragungsrahmen bitweise über den seriellen Datenausgang TXD an den PC zurückzusenden.

Programmierschnittstelle. Alle Ressourcen, auf die ein Software-Entwickler vom Assembler aus direkt zugreifen kann, sind als einzelne Bits zusammen gefasst und über zwei 4-Bit-Adressen $(0000)_2$ und $(0001)_2$ zugänglich. Ein Lesezugriff auf die Adresse $(0001)_2$ führt dazu, dass über die Bitstelle 0 der aktuelle Zustand der Empfangsleitung RXD signalisiert wird, und zwar so, dass eine 0 den Low-Signalpegel und eine 1 den High-Signalpegel anzeigen. Während eines Schreibzugriffs auf die Adresse $(0001)_2$ wird über die Bitstelle 0 die Sendeleitung TXD angesprochen, und zwar so, dass eine 0 zu einem Low-Signalpegel und eine 1 zu einem High-Signalpegel führen. Die untere Tabelle gibt den Überblick über das sog. Memory-Mapping beim Zugriff auf Peripheriekomponenten.

Schnittstellen zwischen PicoBlaze und Peripheriekomponenten

PORT_ID[3..0]	Mode	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Register im IO-Bereich									
0000	Read	-	-	-	-	-	BTN0	LD1	LD0
	Write	-	-	-	-	-	-	LD1	LD0
0001	Read	-	-	-	-	-	-	-	RXD
	Write	-	-	-	-	-	-	-	TXD