

Übung 10

Validierung des Steuerwerkes

Testen Sie ihren Prozessorkern mit dem Testspeicher. Schreiben Sie ein Assemblerprogramm, das alle Lade-, Speicher- und Sprungbefehle sowie einige ALU-Befehle enthält. Übersetzen Sie das Programm in die entsprechenden Bitfolgen und tragen es anschließend in den Testspeicher ein. Simulieren Sie das Programm, verfolgen Sie dessen Abarbeitung am Datenpfad und überprüfen Sie den Prozessor auf eine korrekte Funktionsweise hin.

Synthese des Prozessors

Wählen Sie die **entity PROC** als Top-Level-Unit und synthetisieren Sie diese mit Synplify Pro. Modifizieren Sie gegebenenfalls Ihren Quellcode so, dass eine erfolgreiche Synthese möglich ist.

Speicher und LCD-Ansteuerung

Auf der Homepage finden Sie das Modul *periferie.vhd*, sowie ein vorgefertigtes Blockdiagramm *top.bde*.

- Öffnen Sie das Blockschaltbild und fügen Sie Ihre Entity PROC ein.
- Verbinden Sie die beiden Bausteine PROC und PERIFERIE entsprechend.

Die Entity Periferie bietet über folgende Adressen Zugriff auf:

Adresse	Modul	Zugriffsart	Beschreibung
0x0000-0x00FF	BlockRAM 256*16 Bit	Lesen/Schreiben	Hier steht ab Adresse 0x0000 das auszuführende Programm
0x00FF-0xFFDE	unbenutzt		<i>Lesen</i> liefert die Konstante 0, <i>Schreiben</i> wird ignoriert
0xFFDF	Zugriff auf Taster (PB), Dipschalter, Leds	Lesen/Schreiben	<i>Lesen</i> liefert in den Bits 15 bis 8 den Zustand der Dip-Schalter 8 bis 1, in den Bits 7 bis 4 den Zustand der Taster PB4 bis PB1 (0=gedrückt) und in den Bits 3 bis 0 den Zustand der Leds L4 bis L1 (1=leuchtend). <i>Schreiben</i> ändert den Zustand der Leds. Bit 3 bis 0 geben den neuen Zustand der Leds an, alle übrigen Bits haben keine Auswirkung.
0xFFE0-0xFFFF	LCD	Lesen/Schreiben	<i>Schreiben</i> bringt die Bits 7 bis 0 als Ascii-Zeichen auf das LCD (die restlichen Bits werden ignoriert). Dabei steht an der Adresse 0xFFE0 das erste Zeichen links oben. <i>Lesen</i> liefert in den Bits 7 bis 0 das Ascii-Zeichen an der entsprechenden Stelle, Bit 15 bis 8 haben den Wert 0.

Tabelle 13: Zugriff auf die Periferie

- Schreiben Sie ein „Hallo Welt!“ Programm. Kopieren Sie dazu mit dem store-Befehl die ASCII-Zeichen nach Adresse 0xFFE0 ff., wobei die Zeichen bereits im Speicher als Konstante (z.B. als nullterminierender String) abgelegt sein können. Beenden Sie das Programm mit einer Endlos-Warteschleife.
- Setzen Sie den übersetzten Assemblercode in das Quellfile *periferie.vhd* ein. Beachten Sie dabei, dass das Programm ab Adresse 0 im Speicher stehen muss und der Speicher mit absteigenden Adressen initialisiert wird, d.h. der Befehl an Adresse 0 steht im Quellcode als letzter Befehl.
- Synthetisieren und implementieren Sie die **entity top** auf dem FPGA-Board, um das Ergebnis zu testen (vgl. Übungsblatt 2).