

Aufgabe 1

Der Entwurf wurde bereits abgegeben. Änderungen seitdem:

- **Alle** Module wurden getaktet.
- Begründung: Zu Beginn waren einige Module zur Bearbeitung von Tastereingaben ungetaktet. Dies führte zu folgenden Fehlermeldungen bei der Place-Pase in Xilinx (hier am Beispiel der Moduswahl):
 - „ERROR Place:1018 - A clock IOB / clock component pair have been found that are not placed at an optimal clock IOB / clock site pair. The clock component <s_modus_BUFGP/BUFG> is placed at site <BUFGMUX_X2Y10>. The IO component <s_modus> is placed at site <J17>. This will not allow the use of the fast path between the IO and the Clock buffer. If this sub optimal condition is acceptable for this design, you may use the CLOCK_DEDICATED_ROUTE constraint in the .ucf file to demote this message to a WARNING and allow your design to continue. However, the use of this override is highly discouraged as it may lead to very poor timing results. It is recommended that this error condition be corrected in the design. A list of all the COMP.PINs used in this clock placement rule is listed below. These examples can be used directly in the .ucf file to override this clock rule. < NET "s_modus" CLOCK_DEDICATED_ROUTE = FALSE; > “
 - Über diesen Error wird in Foren viel und ausgiebig diskutiert, wir haben uns nicht getraut, dem Vorschlag in der Fehlermeldung zu folgen, da im Netz in diversen Forenthreads davon strengstens abgeraten wird, weil man evtl. die Hardware beschädigen kann.
- Die Signalnamen wurden leicht verändert. Die Zuordnung zu den ursprünglich ausgedachten Namen ist offensichtlich.

Quelltexte: Siehe Aufgabe 2

Funktionsweise der Module: Siehe Aufgabe 2

Aufgabe 2

angehängte Dateien:

- Gray.vhd, Gray_tb.vhd : Das Modul zum Modus 1 + Testbench
- Snake.vhd, Snake_tb.vhd : Das Modul zum Modus 2 + Testbench
- GameOfLife.vhd, GameOfLife_tb.vhd : Das Modul zum Modus 3 + Testbench
- Geschwindigkeit.vhd, Geschwindigkeit_tb.vhd : Das Modul zur Geschwindigkeitsänderung + Testbench. Es gibt eine langsamste Geschwindigkeitsstufe, die nicht unterschritten werden kann.
- Modus.vhd, Modus_tb.vhd : Das Modul zur Moduswahl + Testbench. Der Initialmodus ist Modus 1, also Graycode.
- Richtung.vhd, Richtung_tb.vhd : Das Modul zur Richtungsänderung + Testbench. Die Initialrichtung ist 1.
- Mux.vhd : Der Multiplexer, der die LED-Belegung des aktiven Modus weitergibt
- Lauflicht.vhd, Lauflicht_tb.vhd : Die Gesamtschaltung + Testbench
- myDCM.vhd : Der DCM, der die Taktfrequenz wie auf dem letzten Aufgabenblatt verändert (Ausgangsfrequenz 10 MHz).
- in7_out1_mylauflicht.ucf : Die Pinbelegung
- in7_out1_mylauflicht.bit : Der Generierte Bitstream zum Überspielen auf den FPGA

Priorität der Tastendrucke:

Wir haben den Tastendrucke keine unterschiedlichen Prioritäten gegeben. Drückt man beispielsweise gleichzeitig „langsamer“ und „schneller“, geschieht einfach gar nichts.

Bedienungsanleitung: Entspricht den Vorgaben auf dem Übungsblatt.

Funktionsweise der Modi:

Modus 1 (Graycode):

- Zählen im Graycode: Bei Richtung 1 aufwärts, bei Richtung 0 abwärts, bei Überlauf wieder von vorne.
- Auswirkung der DIP-Schalter: Verändern der Schalter stellt aktuell angezeigte Zahl um (auf die Zahl, die durch die Schalterpositionen codiert wird) und macht dann normal weiter.

Modus 2 (Game Of Life):

- Game Of Life im Eindimensionalen: Bei Richtung 1 XOR der Nachbarn, bei Richtung 0 negiertes XOR der Nachbarn.
- verlangsamt nochmal extra (damit man etwas erkennt und es nachverfolgen kann).
- Auswirkung der DIP-Schalter: Setzen der Initialbelegung, startet daraufhin sofort.

Modus 3 (Snake):

- Eine „Schlange“ mit „blinkendem Hinterteil“: Bei Richtung 1 läuft sie nach Rechts, bei Richtung 0 nach Links.
- Auswirkung der DIP-Schalter: Bei Veränderung der DIP-Schalter ändert sich die Länge der Schlange. Die neue Länge entspricht der Anzahl der DIP-Schalter, die auf 1 gesetzt sind, jedoch mindestens 1.

Testen auf dem FPGA: Minus minus. :'-(

Was haben wir u.a. gelernt: „Function“ in VHDL (wurden in „Snake.vhd“ benutzt), lieber alles takten, nächstes Mal weniger Testbenches

Aufgabe 3

	FPGA	CPLD
Vollständiger Name	Field programmable Gate Array	Complex Programmable Logic Device
Hersteller	Xilinx, Altera, Lattice, Atmel Actel,..	Altera, Xilinx, Lattice, Schuricht, Atmel, ...
Leistungsmerkmale	Geschwindigkeit: langsam, Stromverbrauch: hoch, Flexibilität: hoch	Geschwindigkeit: schnell, Stromverbrauch: gering, Flexibilität: gering
Interner Aufbau	Besteht aus vielen kleinen Funktionselementen (CLBs). Es wird lokal verdrahtet, wodurch FPGAs programmierbar sind, aber eine inhomogene Signallaufzeit haben. Die Logik ist mittels LUTs/RAM realisiert. Speicherung auf flüchtigem (SRAM) oder nicht flüchtigem (Antifuse) Speicher.	Besteht aus wenigen Funktionselementen (LABs). Eine Schaltmatrix verbindet LABs untereinander und mit E/A global, was zu festen Signallaufzeiten führt. Speicherung auf nicht flüchtigem Speicher zB.: EPROM
Einsatz	Mobilfunk-Basisstationen, Datenreduktionshardware in der Bildverarbeitung, Ethernet-Echtzeitkommunikation, molekulardynamische Rechnungen	Digitaluhren, PWM-Generator, Speichercontroller für mehrere CPUs, Seriell/Parallel Wandler
Unterschiede	Feinmaschiges Array von Logikblöcken und Flip-Flops. komplexer Aufbau der Logikzellen: wenige große Blöcke auf PAL-Basis Verbindungen: Zentrale globale Verbindung - keine Verdrahtung nötig E/A: Relativ direkte Verbindungen zwischen Makrozellen und Pins. Schneller Signalweg von Logikmakrozellen zu Pins. Signallaufzeiten: homogen, konstant	Einfachere Struktur: konfigurierbare Schaltmatrix Aufbau der Logikzellen: große Anzahl kleiner Blöcke auf LUT-Basis (RAM) Verbindungen: Dezentrale lokale Verbindung - Verdrahtung nötig E/A: Ring aus frei zuordenbaren E/A-Blöcken. Jede Logikzelle kann mit jedem Pin verbunden werden, aber über separate Ausgangsregister vor den Pins.

	Flächennutzung: 40-60% Stromverbrauch: hoch bis sehr hoch	Signallaufzeiten: inhomogen, abhängig vom konkreten Signalweg Flächennutzung: 50-95% Stromverbrauch: gering bis mittel
Programmierungssprachen	VHDL, Abel, grafisch (Altium)	VHDL, Verilog, Abel
Stückzahlen	wenige	viele