



Ausgabe: 16. November 2015

Abgaben {  Theorie entfällt  
 Praxis 22. November 2015  
Rücksprache 23./24. November 2015

Dies ist das **1. bewertete Aufgabenblatt**.

Beachten Sie die **Abgabetermine** im Kopf dieser Seite.

Die Gesamtpunktzahl, die Sie erreichen, wird direkt zu ihren Modulpunkten gerechnet.

Es werden **80 Punkte** benötigt **um das Praktikum zu bestehen!**

### Aufgabe 1: Taktuntersetzer (5 Punkte)

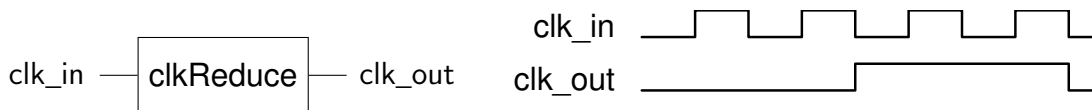


Abbildung 1: Entity clkReduce

Abbildung 2: Taktuntersetzung mit  
divisor = 4

Name	Typ	Art	Beschreibung
divisor	integer	generic	Zählgrenze an der die Transition des Ausgangstaktes geschehen soll
clk_in	std_logic	in	Eingangstakt
clk_out	std_logic	out	Ausgangstakt

In dieser Aufgabe soll eine Taktuntersetzer für die Ansteuerung des 7-Segment-Displays entworfen werden.

Der Display arbeitet mit einer Wiederholungsrate von  $60\text{Hz}$  bis  $1\text{kHz}$ , unsere Zielplattform aber mit  $50\text{MHz}$ . Insofern muss eine Schaltung entwickelt werden, welche aus einem bekannten Eingangstakt `clk_in` mithilfe eines Zählgrenze `divisor` einen Ausgangstakt `clk_out` generiert. Die Frequenz des Ausgangstaktes soll genau um den Faktor `divisor` geringer sein.

1. Implementieren Sie die `architecture behavioral` in der vorgegebenen Datei `clkReduce.vhd`. Nutzen Sie einen `process` zum Zählen.
2. Verifizieren Sie Ihr Design mithilfe der vorgegebenen Testbench `clkReduce_tb`.

---

### Aufgabe 2: Mux4 (3 Punkte)



Abbildung 3: Entity mux4

Entwerfen Sie einen Multiplexer mit 4 Eingängen (a, b, c, d). Die Eingänge und der Ausgang haben eine variable Anzahl von Bits, welche durch den generic N bestimmt wird. Alle Eingänge sind N Bit breit und alle Bits sollen an den N Bit breiten Ausgang weitergeleitet werden.

Nutzen Sie zur Umsetzung ein nebenläufiges Statement!

Der Multiplexer soll eine Fehlerfortpflanzung erlauben ( im Falle das sel nicht aus '0', '1' besteht )

1. Implementieren Sie Ihren Multiplexer in der `architecture behavioral` (Datei `mux4.vhd`).
2. Verifizieren Sie Ihr Design mithilfe der vorgegebenen Testbench `mux4_tb`.

---

### Aufgabe 3: Hochladen auf gitlab (2 Punkte)

1. (2 Punkte) Laden Sie die von Ihnen bearbeiteten Aufgaben auf das gitlab Projekt hoch, welches Ihrer Gruppe zugeordnet wurde.

Nutzen Sie hierbei die bereitgestellten Anleitungen auf *ISIS*.

---

### Literatur

- [1] Mentor Graphics Corporation. *ModelSim SE Reference Manual*, 6.4a edition.
- [2] Mentor Graphics Corporation. *ModelSim SE Tutorial*, 6.4a edition.
- [3] Mentor Graphics Corporation. *ModelSim SE User's Manual*, 6.4a edition.