

Übungsblatt 5

(10 Punkte)

Abgabe bis spätestens Montag, 10. November 2014, 12:00 Uhr

Hinweise: Sie können eine beliebige VHDL Entwicklungsumgebung, wie beispielsweise *VHDL Simili* (Win/Linux) oder *GHDL/GTKWave* (Win/Linux/Unix), verwenden, um Ihre Modelle zu verifizieren. Für Letzteres findet sich eine Vorlage auf der Übungswebsite zum herunterladen. *Ubuntu*-Nutzer können das PPA *ppa:pgavin/ghdl* verwenden. Unter Windows können sie *ghdl* und *gtkwave* unter Cygwin betreiben. Die binaries dazu finden Sie auf der GHDL Homepage.

5.1 Zeitsemantik in VHDL-Simulationen (4 Punkte)

In der Vorlesung wurden einige wichtige Begriffe im Zusammenhang der VHDL-Semantik erläutert.

1. Erklären sie ausführlich, was ein *Deltazyklus* ist und wozu er benötigt wird. Können Deltazyklen in realer Hardware auftreten?
2. Das VHDL *inertial timing*-Modell wird standardmäßig für Eintragungen von Transaktionen in Eventlisten verwendet. Beschreiben Sie dessen Semantik und erläutern Sie den technischen Hintergrund.
3. Simulieren Sie das folgende Beispiel und zeichnen sie die jeweiligen Pegelverläufe. Hat das *inertial timing*-Modell konkrete Auswirkungen? Welcher Unterschied ergibt sich zum *transport*-Modell?

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use work.all;
4
5 entity e_inertial is
6 end e_inertial;
7
8 architecture rtl of e_inertial is
9     signal u, v, w : std_logic;
10 begin
11     p1: process
12     begin
13         u <= '0', '1' after 10 ns, '0' after 15 ns, '1' after 20 ns,
14             '0' after 25 ns, '1' after 30 ns;
15         wait;
16     end process p1;
17
18     p2: process (u)
19     begin
20         v <= u after 4 ns;
21         w <= u after 6 ns;
22     end process p2;
23
24 end rtl;
```

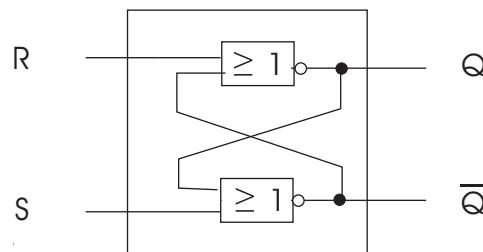
5.2 VHDL-Syntax (3 Punkte)

In der Vorlesung wurde ein Volladdierer in der Hardware-Beschreibungssprache VHDL entworfen. Überlegen Sie, wie man diesen Volladdierer zu einem N-Bit-Addierer mit 'ripple carry'-Berechnung verallgemeinern kann:

1. Zeichnen Sie die allgemeine Struktur eines N-Bit-Addierers unter der Verwendung von Volladdierern. Nutzen Sie dabei die Regularität der Verbindungen der 'inneren' Volladdierer aus.
2. Geben Sie für N=4 Bit eine strukturelle Beschreibung eines N-Bit-Addierers in VHDL an. Verwenden Sie als Bausteine die in der Vorlesung vorgestellten Volladdierer.

5.3 VHDL-Semantik (3 Punkte)

Ein *RS NOR Latch* ist ein bistabiles Speicherelement, das aus zwei rückgekoppelten NOR-Bausteinen aufgebaut ist:



Geben Sie eine Verhaltensbeschreibung des Bausteins in VHDL an. Nehmen Sie dabei an, dass die verwendeten *NOR*-Bausteine eine Verzögerung von je $1ns$ haben. Verwenden Sie zur Darstellung aller Signale den Typ *Bit*. Zeichnen Sie die Signalverläufe, wenn Ihr Baustein mit folgenden Stimuli versehen wird:

S <= '0' after 0 ns, '1' after 5 ns, '0' after 10 ns
 R <= '0' after 0 ns, '1' after 15 ns, '0' after 20 ns

Was ist bei der Initialisierung der Signale zu beachten? Sind in diesem Beispiel Delta-Zyklen für die Simulation notwendig? Begründen Sie Ihre Antwort.

Allgemeine Hinweise: Die Übungstermine und weitere Informationen finden Sie über

<http://ls12-www.cs.tu-dortmund.de/daes/de>

Die Übungszettel werden in der Regel dienstags ausgegeben (Homepage der Übung) und müssen bis zum darauf folgenden Montag bearbeitet werden. Die Abgaben können in den beschrifteten Briefkasten vor dem Sekretariat des LS12 eingeworfen werden oder per Email (PDF) an den entsprechenden Übungsgruppenleiter gesendet werden.