

Übung zur Vorlesung Einführung in Computer Microsystems

Prof. Dr. A. Koch
Thorsten Wink



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Sommersemester 11 Übungsblatt 1

Die folgenden Aufgaben sollen in der HDL Verilog bearbeitet werden. Zur Simulation können Sie XILINX ISE verwenden. Es ist als WebPack-Edition frei verfügbar und auch auf den Poolrechnern der RBG installiert. Dort kann es einfach mit dem Befehl `ise` gestartet werden. Ein Tutorial zur Installation und Benutzung finden Sie auf unserer Webseite. Wir empfehlen, Version 11 zu verwenden.

Aufgabe 1.1 Zähler in Verilog

- Beschreiben Sie einen 4-Bit-Zähler in Verilog HDL. Der Zähler hat einen Eingang für den Takt (mit `clk` bezeichnet). Der Zählerstand wird mit dem Ausgang `count` ausgegeben.
- Der Zähler soll um einen Eingang für ein `enable`-Signal erweitert werden. Es wird nur gezählt, wenn `enable` 1 ist.
- Der Zähler soll mit einem synchronen Reset (`sreset`) erweitert werden, so dass mit steigende Taktflanke der Zähler auf 0 zurückgesetzt wird wenn `sreset`=1.
- Der Zähler soll mit einem asynchronen Reset (`areset`) erweitert werden, so dass unabhängig vom Takt der Zähler auf 0 zurückgesetzt wird.
- Über eine Leitung `set` und einen 4-Bit-Dateneingang `value` soll der Zähler synchron auf den Wert von `value` gesetzt wird, sobald `set` 1 ist.
- Der Zähler soll nur bis zu einem Wert `max` zählen, der über einen zu definierenden Parameter gesetzt werden kann. Ist kein Parameter beim Modulaufruf angegeben, soll wie bisher ohne einen Schwellwert gezählt werden.
- Schreiben Sie einen Testrahmen für die letzte Teilaufgabe, so dass `max` = 5. Zu Beginn sollen alle Eingangssignale auf 0 liegen. Nach 7 ns soll ein synchroner Reset erfolgen, danach soll der Startwert 3 gesetzt werden und der Zähler gestartet werden. Geben Sie ein Timing-Diagramm an, bei dem die Werte für `clk` und `count` zu sehen sind.

Aufgabe 1.2 Paritätsbit

Schreiben Sie ein Verilog-Modul, das zu einem übergebenen Bitstring von n Bits ein Paritätsbit hinten anhängt, welches 1 ist, wenn die Anzahl der Einsen im Bitstring ungerade ist, und das 0 ist, wenn die Anzahl der Einsen im Bitstring gerade ist. Der so entstandene neue Bitstring soll der Ausgang des Moduls sein. Wird kein Parameter angegeben, so soll die Bitbreite des Ausgangs 9 Bit betragen.

Aufgabe 1.3 Fragen

- Wie können Werte an Wire-Variablen zugewiesen werden? Geben Sie ein Beispiel an. Können Wires Werte speichern?
- Wie können Zuweisungen an Signale verzögert werden?
- Wie unterscheiden sich `initial` und `always`?

Übung zur Vorlesung Einführung in Computer Microsystems

Aufgabe 1.4 Addierer/Subtrahierer

Gegeben sind folgende Verilog HDL Module:

```
module HalfAdder(
    input A, B,
    output Sum, Carry
);

    assign Sum = A ^ B;
    assign Carry = A & B;
endmodule

module FullAdder(
    input A, B, CarryIn,
    output Sum, CarryOut
);

    wire sum1, carry1, carry2;

    HalfAdder ha1(A.(A), B.(B), Sum.(sum1), .Carry(carry1));
    HalfAdder ha2(A.(CarryIn), B.(sum1), .Sum(Sum), .Carry(carry2));

    assign CarryOut = carry1 | carry2;
endmodule
```

- Konstruieren Sie mit Hilfe der beiden obigen Module einen 4-Bit Ripple-Carry Addierer mit den Eingängen A und B und dem Ausgang Sum. Schreiben Sie eine Testbench, welche die Additionsfunktion testet und führen Sie eine Verhaltenssimulation durch.
- Erweitern Sie den 4-Bit Addierer aus a) um eine Subtraktionsfunktion. Ein zusätzliches Eingangssignal Sub soll von Addition auf Subtraktion umschalten. Der „-“-Operator darf dazu *nicht* verwendet werden. Erweitern Sie Ihre Testbench aus a) um den zusätzlichen Test der Subtraktion. Testen Sie auch negative Differenzen.

Aufgabe 1.5 Tageberechnung

Schreiben Sie ein Verilog-Modul, das als Eingabe den Monat erwartet (binär codiert) und als Ausgabe das Ergebnis liefert, ob der Monat 31 Tage hat oder nicht.

Durch Bearbeitung und Abgabe der Hausaufgaben können Sie einen Notenbonus für die Prüfung CMS erwerben. Es gelten folgende Regeln:

- Die Bearbeitung ist in Gruppen zu 2 Studierenden möglich.
- Gruppenübergreifendes Erarbeiten von Lösungsideen ist erlaubt. Es muss jedoch jede Gruppe eine eigene Lösung abgeben. Wir werden dies kontrollieren, bei Plagiaten werden ALLE Bonuspunkte ALLER beteiligten Gruppen aberkannt!
- Es kann maximal ein Notenbonus von 3 Notenschritten erreicht werden.
- Der Bonus wird nur angerechnet, wenn die Klausur auch ohne den Bonus bestanden ist!!!
- Um den Bonus zu erhalten, muss die Abgabe über das Moodle-System bis zum Abgabetermin hochgeladen werden. Genaue Infos dazu werden rechtzeitig auf der Webseite und im Forum bekannt gegeben. Quellcode muss als

Übung zur Vorlesung Einführung in Computer Microsystems

Textdatei abgegeben werden, schriftliche Ausarbeitungen müssen im pdf-Format eingereicht werden. Bei Gruppen muss nur eine Abgabe erfolgen, alle Dokumente müssen die Namen und Matrikelnummern BEIDER Gruppenteilnehmer enthalten.

Diese Hausaufgaben müssen bis 6.5.11, 18:00 über das Moodle-System abgegeben werden.

Hausaufgabe 1.1 Mittelwert (5 Punkte)

- a) Schreiben Sie ein Verilog-Modul, welches aus vier 16-Bit breiten Zahlen in Zweierkomplement-Darstellung den Mittelwert berechnet. Achten Sie auf eine möglichst effiziente Implementierung!
- b) Schreiben Sie eine Testbench für Ihr Modul. Testen Sie darin alle Fälle, die Ihrer Meinung nach nötig sind, um einen fehlerfreien Betrieb zu gewährleisten. Begründen Sie jeden Eingabedatensatz mit erwartetem Ergebnis und dem Grund für diesen Testfall.

Hausaufgabe 1.2 Paralleler Multiplizierer (5 Punkte)

Konstruieren Sie aus den Modulen von Aufgabe 1.4 einen voll parallelen 4-Bit Multiplizierer für *positive* Zahlen. Der „*“-Operator darf dazu *nicht* verwendet werden. Wieviele Bits werden für das Ergebnis benötigt? Testen Sie die Funktion des Multiplizierers durch Simulation mit einer Testbench.

Plagiarismus

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Weitere Infos unter www.informatik.tu-darmstadt.de/plagiarism