

1 Tasten entprellen

Die vier Tasten, die auf der Nexys2-Platine vorhanden sind, weisen ein Verhalten auf, das *Prellen* genannt wird. Das bedeutet, dass beim Drücken oder Loslassen einer Taste nicht, wie gewünscht, der Kontakt ein einziges Mal geschlossen bzw. geöffnet wird, sondern aufgrund der federnden Eigenschaften der Taste mehrere Male zwischen beiden Zuständen hin- und herspringen kann.

Durch die Mehrfachauslösung der Tasten kann es zu unerwünschtem Verhalten kommen, weshalb man die Tasten *entprellen* muss. In dieser Aufgabe sollen Sie zunächst das Prellen beobachten und anschließend eine Schaltung entwerfen, die das Problem behebt.

- a) Schreiben Sie ein Modul mit einem Takteingang und einem Eingang für eine Taste, das bei jedem Tastendruck synchron zum Takt einen 4-Bit-Zähler um Eins erhöht. Der Zählerstand soll am Ausgang anliegen.

Um den Tastendruck, d. h. eine steigende Flanke des Tasten-Eingangssignals, zu erkennen, legen Sie ein 2-Bit-Schieberegister an, in das in jedem Takt der aktuelle Wert des Tastensignals eingeschoben wird. Ein Tastendruck liegt also genau dann vor, wenn im Schieberegister die Werte [0,1] gespeichert sind.

Schreiben Sie den Verilog-Code so, dass die Sensitivitätsliste jedes `always`-Blocks (einer für die Erkennung des Tastendrucks und einer für den Zähler) ausschließlich das Taktsignal enthält.

- b) Legen Sie vier Instanzen des Tastendruck-Zählers an und schließen Sie jeweils eine der Tasten an. Geben Sie die Zählerstände auf je einer Ziffer der Siebensegmentanzeige aus. Beobachten Sie, was beim Drücken und Loslassen der Tasten passiert.

Tipp: Um sich Schreiarbeit zu ersparen, können Sie mehrere Instanzen eines Moduls mit dem `generate`-Konstrukt erzeugen. Angenommen, Sie haben das folgende Modul:

```
module mymodule (input signal_in, output signal_out);  
    ...  
endmodule
```

Dann können Sie es folgendermaßen mehrfach mit einer Schleife instanziiieren:

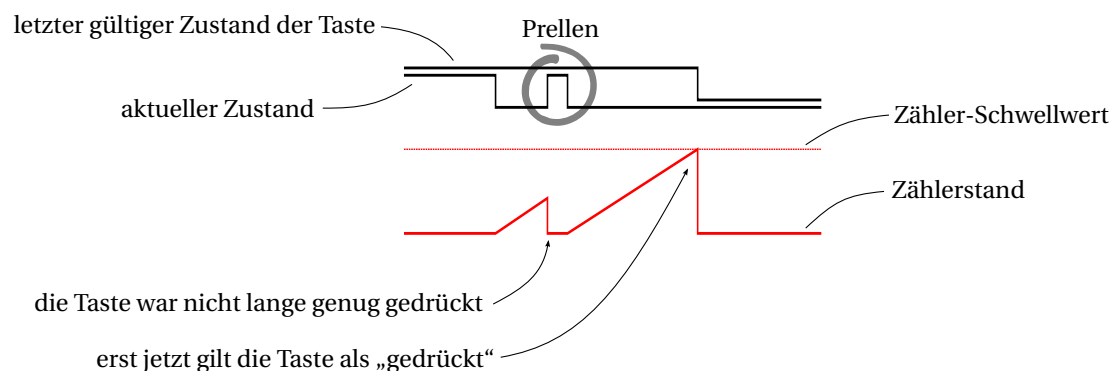
```
wire [3:0] x;  
wire [3:0] y;  
genvar i;  
generate for(i=0; i<4; i=i+1)  
    begin: loop_name  
        mymodule instanz_name  
        (  
            .signal_in(x[i]),  
            .signal_out(y[i])  
        );  
    end  
endgenerate
```

- c) Das Entprellen der Tasten funktioniert nun so: Damit die Taste als „gedrückt“ oder als „losgelassen“ gilt, muss sie für eine gewisse Zeit ununterbrochen in dem jeweiligen Zustand verbleiben. Damit wird verhindert, dass das kurzzeitige Springen zwischen den Zuständen während des Umschaltvorgangs als zusätzlicher Tastendruck interpretiert wird.

Legen Sie also ein neues Modul an, das einen Takteingang, einen Eingang für das Signal der Taste und einen Ausgang für das entprellte Signal hat. In dem Modul soll es ein Register geben, das den letzten gültigen Zustand der Taste speichert (dies ist gleichzeitig das Ausgangssignal des Moduls).

Um das ununterbrochene Verweilen der Taste in einem Zustand zu überprüfen, soll es eine Zählervariable geben, die in jedem Takt um Eins erhöht wird, falls sich der aktuelle Zustand der Taste vom letzten gültigen Zustand unterscheidet. Sobald aber die Taste in den letzten gültigen Zustand zurückspringt, wird der Zähler wieder auf Null gesetzt, so dass der Vorgang von vorne beginnen muss. Erst wenn der Zähler einen bestimmten Wert erreicht, den Sie als Parameter vorgeben, soll der gültige Zustand der Taste aktualisiert werden, indem der Inhalt des Registers invertiert wird.

Dieser Entprell-Mechanismus ist in der folgenden Skizze verdeutlicht:



Bauen Sie nun Ihr Entprell-Modul in die Schaltung ein, die das Drücken der Tasten auf der Siebensegmentanzeige ausgibt. Erhöhen Sie den Entprell-Schwellwert so lange, bis jeder Tastendruck garantiert nur einmal gezählt wird. Welcher Zeitdauer entspricht dies? Was passiert, wenn Sie den Schwellwert zu hoch wählen?