

Übung 4 Process Structure und State Machines

Jan Grapengeter

May 2, 2019

1 Process Structure

Prozess Statements können benutzt werden, um VHDL Module zu strukturieren, außerdem erlaubt die Prozess Struktur die Verwendung weiterer Statements, die später noch eingeführt werden. Die Syntax für ein Process Statement ist:

```
LABEL(optional):process(sensitivity list(optional))
declarations
begin
SEQUENZ
end process;
```

Aufgabe:

Legen Sie ein neues Projekt an, definieren Sie die Schalter und LEDs des Basys3 als Ein- und Ausgänge. Verbinden Sie zwei Schalter innerhalb eines Prozesses mit einem Und-Gatter und legen Sie das Signal auf eine LED.

2 If Statement

If Statements sind nur innerhalb von Prozessen erlaubt, mit ihnen lassen sich konditionale Strukturen realisieren. Die Syntax für ein If Statement ist:

```
if BEDINGUNG then
SEQUENZ
elsif BEDINGUNG then
SEQUENZ
else
SEQUENZ
end if;
```

Um den Wert eines Signals abzufragen, wird folgende Syntax für die Bedingung verwendet:

if(SIGNAL='1') then (statt 1 könnte hier auch 0 stehen, je nachdem was man abfragen möchte).

Um die Werte eines ganzen Vektors abzufragen, kann folgende Syntax verwendet werden:

VEKTOR(X downto Y)="1111"(Länge der Sequenz von 1 ist X-Y+1)

X ist hier der erste Wert, den man abfragt, Y der letzte, es lassen sich damit auch zusammenhängende Teile eines Vektors abfragen. Innerhalb einer Bedingung können Logikgatter verwendet werden.

Aufgaben:

- 1.: Realisieren Sie Ihre Schaltung aus der vorherigen Aufgabe mit einem If Statement.
- 2.: Realisieren Sie das unten stehende LUT mithilfe eines IF Statements. Verwenden Sie kein Elself Statement.
- 3.: Verbinden Sie nun zusätzlich zwei weitere Schalter mit einer LED wie in der LUT angegeben.

Verwenden Sie diesmal ein Elsif Statement.

4.: Sehen Sie sich Ihre Schaltung in "Elaborated Design" und "Synthesis/Schematic" an und vergleichen Sie die Ergebnisse. Was fällt Ihnen auf?

| LUT Und-Gatter | | |
|----------------|----|---|
| A1 | A2 | E |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Table 1: LUT Und-Gatter

3 Case When Statement

Das Case When Statement ist eine weitere Möglichkeit Bedingungen abzufragen. Die Syntax für das Statement ist:

```
case SIGNAL is
when wert1 =>
SEQUENZ
when wert2 =>
SEQUENZ
end case;
```

"when other" kann als Standardfall eines Case When Statements verwendet werden und wird immer dann aktiv, wenn keiner der anderen Fälle eintritt.

Aufgaben:

- 1.: Realisieren Sie Ihre Schaltung aus der vorherigen Aufgabe mit einem Case When Statement.
- 2.: Sehen Sie sich Ihre Schaltung in "Elaborated Design" und "Synthesis/Schematic" an und vergleichen Sie die Ergebnisse. Was fällt Ihnen auf?
- 3.: Implementieren Sie ein XOR-Gatter mit einem If Statement, einem If Elsif Statement, einem Case When Statement und als einfaches Gatter und vergleichen Sie die Ergebnisse, indem Sie die Schaltung in "Elaborated Design" und "Synthesis/Schematic" ansehen.

4 State Machines

Hier ist das Blockschaltbild eines Endlichen Automaten.

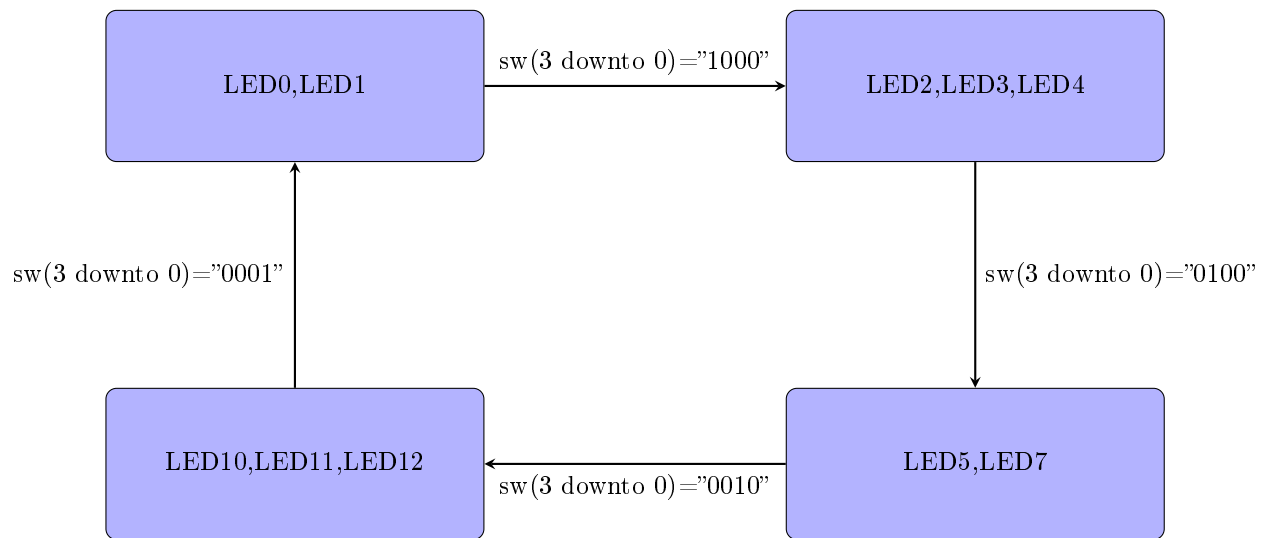


Figure 1: Endlicher Automat

Aufgabe:

- 1.: Implementieren Sie diesen Automaten in VHDL und bringen Sie Ihre Schaltung auf das FPGA. Zu Beginn sollte LED0 leuchten.
- 2.: Welche Probleme treten bei der Implementierung auf?
- 3.: Verwenden Sie Ihr Wissen um If Statements, um Ihren Mehrbit-Volladdierer aus der letzten Übung zu vereinfachen.