Tutorium zu Computer-Engineering im SS19 Termin 2

Jakob Otto

HAW Hamburg

1. April 2019

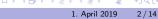


Ablauf

- Praktikum
 - ▶ Praktikum Aufgabe 1







Aufgabenzettel





UCF-Datei





kleine Wiederholung

- UCF-Datei mappt ein- und Ausgänge auf pins
- Format:

```
NET <port-name> LOC = <pin> | IOSTANDARD=LVCMOS33;
```

• Als Beispiel:

```
NET "clk" LOC = "V10" | IOSTANDARD=LVCMOS33;
```



5/14



1. April 2019

Woher die ganzen pins?!

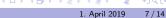
- Normalerweise aus irgendwelchen Datenblättern..
- Hier: Aus der CE_Board-Doku
- Weitere Doku zum Nexys2-board gibts HIER





Ideen zur Praktikumsaufgabe





Tick-Generator

- erzeugt ticks mit einer gewünschten Frequenz auf basis eines gegebenen Taktes
- kann Frequenz nur absenken!
- im Grunde zählt der Tickgenerator nur Takte
- Ticks werden wie ein druck eines buttons interpretiert.





Tickgenerator-idee I

```
entity TickGen is
    port(
        tick : out std_logic;
        clk : in std_logic
    );
end entity TickGen;
```

Tickgenerator-idee II

```
tickGen: process (clk) is
  constant maxValue : integer := 25000;
  variable count : integer range 0 to maxValue := 0;
  variable tick_v : std_logic;
begin
  if (rising edge(clk)) then
    count := count + 1;
    if (count = maxValue) then
      count := 0;
      tick v := '1':
    else
      tick v := '0':
    end if:
  end if:
  tick <= tick v;
end process tickGen;
```

Auswahl der Anoden

- um alle 7 Seg. Displays zu nutzen müssen die jeweiligen Anoden angesteuert werden.
- Dazu nutzt ein zyklisches shiftregister!
- shiftregister soll '1' mit frequenz des generierten ticks shiften.
- Dadurch werden nacheinander die einzelnen Anoden durchgeschaltet.





```
shift:
process(shiftRegister_cs) is
begin
    shiftRegister_ns <= shiftRegister_cs(shiftRegister_cs'left-1 deend process shift;
shiftregister sollte mit "1110" initialisiert werden!</pre>
```



Anode ist low-aktiv

Lookup table

- Die nibble werden mit 4 bit codiert
- zum Anzeigen müssen die nibble auf die Kathodenausgänge umgesetzt werden.
- Dazu ist ein lookup-table sinnvoll
- Die Ausgangswerte werden dazu in einem lookup-table hinterlegt werden
- ullet \rightarrow case-when!





Lookup table

```
case nibVal_v is
  when "0000" =>
    segments_v := "10000001";
  when "0001" =>
    segments_v := "11001111";
  when "0010" =>
    segments_v := "10010010";
  -- usw
end case;
```

