Protokoll zum Praktikum Programmierbare Schaltkreise

Fakultät Informatik
TU Dresden

Christian Kroh
Matrikelnummer: 3755154
Studiengang: Informatik (Diplom)
Jahrgang: 2010/2011

Patrick Russell
Matrikelnummer: xxx
Studiengang: Informatik (Diplom)
Jahrgang: 2010/2011

26. November 2013, Dresden

Aufgabe 1 - Binär-Dekoder

1.1 Entwurf

Input 4-Bit Binärzahl durch Schieberegister SW3 ... SW0

Output 7-Segmente Darstellung einer Hexadezimalziffer (8 Einzelsignale = 7 Segmente + 1 Punkt)

Input				Output								
SW3	SW2	SW1	SW0	Hex	A	B	C	D	E	F	G	dot
0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	1	1	1	0	0	1	1	1	1	1
0	0	1	0	2	0	0	1	0	0	1	0	1
0	0	1	1	3	0	0	0	0	1	1	0	1
0	1	0	0	4	1	0	0	1	1	0	0	1
0	1	0	1	5	0	1	0	0	1	0	0	1
0	1	1	0	6	0	1	0	0	0	0	0	1
0	1	1	1	7	0	0	0	1	1	1	1	1
1	0	0	0	8	0	0	0	0	0	0	0	1
1	0	0	1	9	0	0	0	0	1	0	0	1
1	0	1	0	A	0	0	0	1	0	0	0	1
1	0	1	1	b	1	1	0	0	0	0	0	1
1	1	0	0	C	0	1	1	0	0	0	1	1
1	1	0	1	d	1	0	0	0	0	1	0	1
1	1	1	0	E	0	1	1	0	0	0	0	1
1	1	1	1	F	0	1	1	1	0	0	0	1

?? Decoder.vhdl Code

1.2 Auswertung

Ressourcenbedarf

- 7 Logik-Elemente
- 12 Pins

Aufgabe 2 - Hamming-Distanz

2.1 Entwurf

Input 2 4-Bit Werte

- 1.Wert: 4-Bit Binärzahl durch Schieberegister SW3 ... SW0
- 2.Wert: 4-Bit Binärzahl durch Schieberegister SW7 ... SW4

Output 7-Segmente Darstellung einer Hexadezimalziffer (8 Einzelsignale = 7 Segmente + 1 Punkt)

Ansatz SW3 ... SW0 und SW7 ... SW4 logisch xor verknüpfen und Ergebnis direkt auf 7-Segmente Anzeige mappen

(**??** Hamming.vhdl Code)

2.2 Auswertung

Ressourcenbedarf

- 9 Logik-Elemente
- 16 Pins

Aufgabe 3 - Modulo-n-Zähler

3.1 Entwurf

a) Der Zähler ist nach 50 Millionen Schritten zurückzusetzen (50 MHz Takt entspricht 50 Millionen Taktperioden pro Sekunde)

b) Für das Schieberegister ist der Zählerzustand ein Enable-Signal

c)

Input

- 50MHz Takt
- Reset (Schiebeschalter SW0)

Output LED-Zeile

Ansatz 2 Komponenten: Schieber und Zähler

Zähler gibt alle 50-Millionen Taktperioden (50MHz Takt ergibt $50 \cdot 10^6$ Taktschritte pro Sekunde) einen Takt lang ein enable-Signal aus. (**??** Zaehler.vhdl-Code)

Schieber beinhaltet den Zähler als Komponente und verschiebt bei dessen enable-Signal die LED-Anzeige um eine Stelle pro Takt. (**??** Schieber.vhdl-Code)

3.2 Auswertung

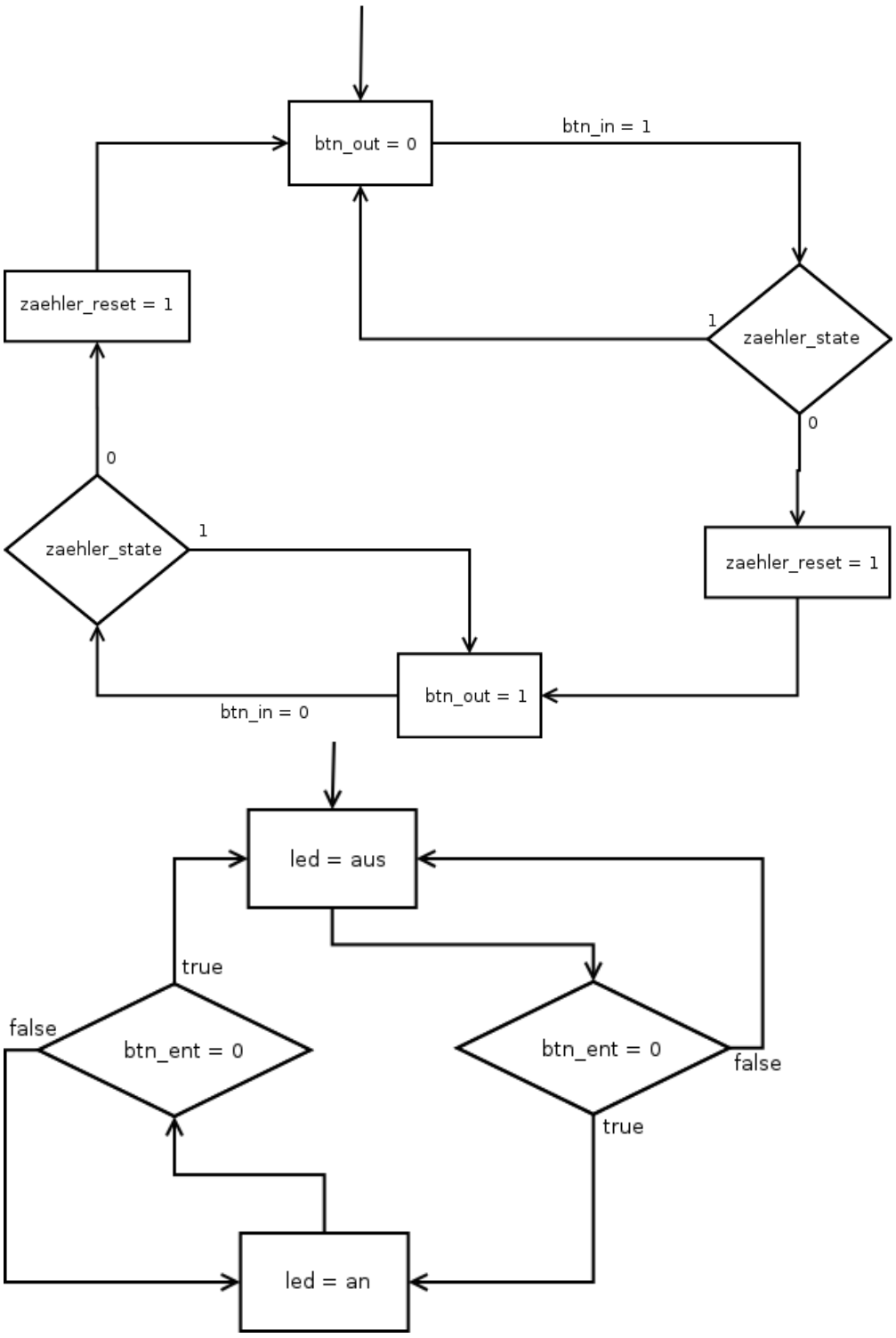
Ressourcenbedarf

- 60 Logik-Elemente
- davon 38 dedizierte Logik-Elemente
- 12 Pins
- maximale Taktfrequenz von 250 MHz

Aufgabe 4 - Entprell-Automat

4.1 Entwurf

State-Machine-Charts



LED enthält die Komponente Entprellung und verbindet die Ein- und Ausgangssignale. (**??** LED.vhdl-Code)

Entprellung enthält die Komponente Zaehler, der bei der Veränderung des Eingangsignals gestartet wird und für 3ms weitere Änderungen ignoriert. (**??** Entprellung.vhdl-Code)

Zaehler implementiert einen Zähler, der durch ein Signal definierte Schritte zählt. Ausgegeben wird der aktuelle Zustand des Zählers. Eingegeben ein Reset-Signal. (**??** Zaehler.vhdl-Code)

4.2 Auswertung

Ressourcenbedarf

- 86 Logik-Elemente
- davon 79 dedizierte Logik-Elemente
- 44 Register
- 3 Pins
- maximale Taktfrequenz von 178 MHz

Aufgabe 5 - HALLO-Anzeige

5.1 Entwurf

zu a) Es müssen 5 Zeichen kodiert werden (H, A, L, O, Leerzeichen).

$ld\,5 = 3$

Daher werden für eine Binärkodierung mindestens 3 Bits benötigt.

Input			Output								
BIT2	BIT1	BIT0	CHAR	A	B	C	D	E	F	G	DOT
0	0	0		1	1	1	1	1	1	1	1
0	0	1	H	1	0	0	1	0	0	0	1
0	1	0	A	0	0	0	1	0	0	0	1
0	1	1	L	1	1	1	0	0	0	1	1
1	0	0	O	0	0	0	0	0	0	1	1

b) Für das Schieberegister ist der Zählerzustand ein Enable-Signal

c)

5.2 Auswertung

Ressourcenbedarf

- 73 Logik-Elemente
- 61 Register
- 34 Pins
- maximale Taktfrequenz von 262 MHz

Aufgabe 6 - Stoppuhr

6.1 Entwurf

6.2 Auswertung

Ressourcenbedarf

- 163 Logik-Elemente
- davon 121 dedizierte Logik-Elemente
- 35 Pins
- maximale Taktfrequenz von 225 MHz

Anhang

7.1 01-Aufgabe Code

Listing 1: VHDL-Code Decoder.vhdl

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity Decoder is
6
7    port (
8      sw : in std_logic_vector(3 downto 0);
9      cc : out std_logic_vector(7 downto 0));
10
11 end Decoder;
12
13 architecture Dec1 of Decoder is
14 begin
15
16   ------------------------------------------------------------------
17   -- Outputs: 4 bit breites Wort in Hexadezimale 7-Segment Anzeige
18   ------------------------------------------------------------------
19   with sw select
20     cc <= "00000011" when "0000",
21         "10011111" when "0001",
22         "00100101" when "0010",
23         "00001101" when "0011",
24         "10011001" when "0100",
25         "01001001" when "0101",
26         "01000001" when "0110",
27         "00011111" when "0111",
28         "00000001" when "1000",
29         "00001001" when "1001",
30         "00010001" when "1010",
31         "11000001" when "1011",
32         "01100011" when "1100",
33         "10000101" when "1101",
34         "01100001" when "1110",
35         "01110001" when "1111";
36 end dec1;
```