

# Musterlösung Übungsblatt 4

## Schaltnetze, VHDL und Speicher aus Flipflops

Vorlesung *Technische Grundlagen der Informatik 1*, Sommersemester 2020

Erstellt von Dr.-Ing. Kristian Ehlers

### Aufgabe 1 Komparator 2

Entwerfen Sie einen Komparator, der es ermöglicht, die zwei positiven 2-Bit Binärzahlen  $a$  und  $b$  miteinander zu vergleichen. Hierbei seien die drei Funktionen  $g$  (greater),  $e$  (equal) und  $l$  (lower) zu definieren, die jeweils den Wert 1 annehmen, sollte  $a > b$ ,  $a = b$  beziehungsweise  $a < b$  gelten.

#### (a) Wahrheitstabelle

Geben Sie die Funktionen in Form einer Wahrheitstafel an.

| $a_{10}$ | $b_{10}$ | $a_1$ | $b_1$ | $a_0$ | $b_0$ | $g$ | $e$ | $l$ |
|----------|----------|-------|-------|-------|-------|-----|-----|-----|
| 0        | 0        | 0     | 0     | 0     | 0     | 0   | 1   | 0   |
| 0        | 1        | 0     | 0     | 0     | 1     | 0   | 0   | 1   |
| 1        | 0        | 0     | 0     | 1     | 0     | 1   | 0   | 0   |
| 1        | 1        | 0     | 0     | 1     | 1     | 0   | 1   | 0   |
| 0        | 2        | 0     | 1     | 0     | 0     | 0   | 0   | 1   |
| 0        | 3        | 0     | 1     | 0     | 1     | 0   | 0   | 1   |
| 1        | 2        | 0     | 1     | 1     | 0     | 0   | 0   | 1   |
| 1        | 3        | 0     | 1     | 1     | 1     | 0   | 0   | 1   |
| 2        | 0        | 1     | 0     | 0     | 0     | 1   | 0   | 0   |
| 2        | 1        | 1     | 0     | 0     | 1     | 1   | 0   | 0   |
| 3        | 0        | 1     | 0     | 1     | 0     | 1   | 0   | 0   |
| 3        | 1        | 1     | 0     | 1     | 1     | 1   | 0   | 0   |
| 2        | 2        | 1     | 1     | 0     | 0     | 0   | 1   | 0   |
| 2        | 3        | 1     | 1     | 0     | 1     | 0   | 0   | 1   |
| 3        | 2        | 1     | 1     | 1     | 0     | 1   | 0   | 0   |
| 3        | 3        | 1     | 1     | 1     | 1     | 0   | 1   | 0   |

#### (b) VHDL - Case-When

Geben Sie eine auf der Wahrheitstafel basierende Beschreibung des Komparators in VHDL an. Nutzen Sie das case-when Statement.

```

34  entity Comparator is
35  Port ( A1, A0, B1, B0 : in  STD_LOGIC;
36        | | | g, e, l    : out STD_LOGIC);
37  end Comparator;
```

```

39  architecture Behavioral of Comparator is
40
41  signal inputs  : STD_LOGIC_VECTOR(3 downto 0);
42  signal outputs : STD_LOGIC_VECTOR(2 downto 0);
43
44  begin
45  inputs <= A1 & B1 & A0 & B0;
46  g <= outputs(2);
47  e <= outputs(1);
48  l <= outputs(0);
49
50  process (inputs)
51  begin
52      case inputs is
53          WHEN "0000" => outputs <= "010";
54          WHEN "0001" => outputs <= "001";
55          WHEN "0010" => outputs <= "100";
56          WHEN "0011" => outputs <= "010";
57          WHEN "0100" => outputs <= "001";
58          WHEN "0101" => outputs <= "001";
59          WHEN "0110" => outputs <= "001";
60          WHEN "0111" => outputs <= "001";
61          WHEN "1000" => outputs <= "100";
62          WHEN "1001" => outputs <= "100";
63          WHEN "1010" => outputs <= "100";
64          WHEN "1011" => outputs <= "100";
65          WHEN "1100" => outputs <= "010";
66          WHEN "1101" => outputs <= "001";
67          WHEN "1110" => outputs <= "100";
68          WHEN "1111" => outputs <= "010";
69      end case;
70
71  end process;
72
73  end Behavioral;

```

(c) DMF

Geben Sie die disjunktiven Minimalformen der Funktionen an.

|          |     |          |    |    |    |
|----------|-----|----------|----|----|----|
|          |     | $a_0b_0$ |    |    |    |
|          | $g$ | 00       | 01 | 11 | 10 |
| $a_1b_1$ | 00  | 0        | 0  | 0  | 1  |
|          | 01  | 0        | 0  | 0  | 0  |
|          | 11  | 0        | 0  | 0  | 1  |
|          | 10  | 1        | 1  | 1  | 1  |

|          |     |          |    |    |    |
|----------|-----|----------|----|----|----|
|          |     | $a_0b_0$ |    |    |    |
|          | $l$ | 00       | 01 | 11 | 10 |
| $a_1b_1$ | 00  | 0        | 1  | 0  | 0  |
|          | 01  | 1        | 1  | 1  | 1  |
|          | 11  | 0        | 1  | 0  | 0  |
|          | 10  | 0        | 0  | 0  | 0  |

$$g = a_1\bar{b}_1 + a_1a_0\bar{b}_0 + \bar{b}_1a_0\bar{b}_0$$

$$e = \bar{a}_1\bar{b}_1\bar{a}_0\bar{b}_0 + a_1b_1\bar{a}_0\bar{b}_0 + \bar{a}_1\bar{b}_1a_0b_0 + a_1b_1a_0b_0$$

$$l = \bar{a}_1b_1 + b_1\bar{a}_0b_0 + \bar{a}_1\bar{a}_0b_0$$

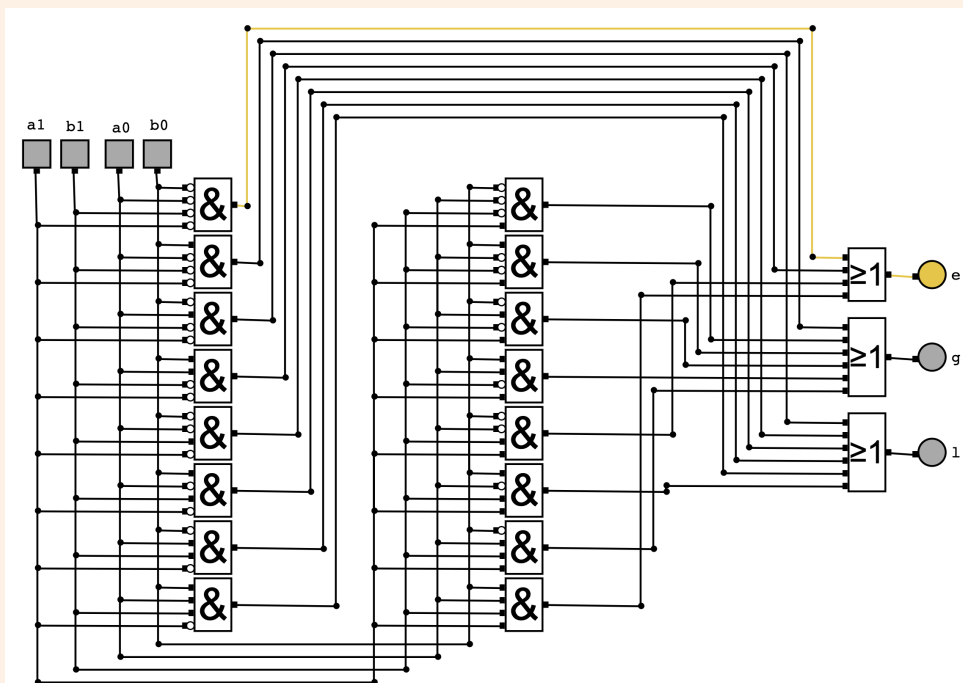
#### (d) VHDL - DMF

Geben Sie eine auf den disjunktiven Minimalformen basierende Beschreibung des Komparators in VHDL an.

```
1  entity Comparator is
2  |   Port ( A1, A0, B1, B0 : in  STD_LOGIC;
3  |   |   |   g, e, l       : out STD_LOGIC);
4  end Comparator;
5
6  architecture Behavioral of Comparator is
7
8  begin
9  |   g <= (a1 and not b1) or (a1 and a0 and not b0) or (not b1 and a0 and not b0);
10 |   e <= (not a1 and not b1 and not a0 and not b0) or (not a1 and not b1 and a0 and b0) or
11 |   |   |   (a1 and b1 and not a0 and not b0) or (a1 and b1 and a0 and b0);
12 |   l <= (not a1 and b1) or (not a1 and not a0 and b0) or (b1 and not a0 and b0);
13 end Behavioral;
```

#### (e) Realisierung mit Hilfe eines Decoders

Realisieren Sie die Funktionen mit Hilfe eines Decoders und ODER-Gattern mit beliebig vielen Eingängen.



## Aufgabe 2 Speicher

Ein Speicher besteht aus einem Adressdecoder und Speicherzellen, die z.B. mit Hilfe von Flipflops realisiert sind. Er besitzt außer den Eingängen für die Adresse noch einen Dateneingang. Wird eine Adresse angelegt, kann in die selektierte Zelle der Wert am Dateneingang gespeichert werden. Die übrigen Zellen bleiben unverändert.

Entwerfen Sie einen 4-Bit Speicher (1-Bit breit mit 4 Speicherstellen) unter der ausschließlichen Verwendung von RS-Flipflops und AND-Gattern ggf. mit negierten Eingängen in LogicCircuits. Die 2-Bit Adressierung soll mit Hilfe eines Decoders realisiert werden. Zusätzlich zu den Adresseingängen verfügt der Speicher über einen 1-Bit breiten Dateneingang.

**(a) Decoder**

Realisieren Sie zuerst den Decoder in LogicCircuits. Dieser verfügt über zwei Adresseingänge und vier Ausgänge, die in Abhängigkeit von den Eingängen beschaltet werden. Es ist stets nur ein Ausgang aktiv.

Lösung ist Teil von Aufgabe c).

**(b) Ansteuergleichungen**

Bestimmen Sie die Ansteuergleichungen der beiden Eingänge eines RS-Flipflops in Abhängigkeit vom Dateneingang  $D$  und der korrespondierenden Ausgabe  $E$  des Adressdecoders.

| E | D | R | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$$\Rightarrow R = \overline{D} \wedge E$$

$$\Rightarrow S = D \wedge E$$

**(c) Speicher**

Realisieren Sie nun den Speicher bestehend aus vier RS-Flipflops in LogicCircuits. Setzen Sie die eben bestimmten Ansteuergleichungen für die R- und S-Eingänge um und nutzen Sie Ihren zuvor entworfenen Decoder. Schließen Sie an dem Ausgang  $Q$  der Flipflops jeweils eine Lampe an.

