

Versuch V1

C752 Digitaltechnik

Stand: 3. November 2023

Tom Mohr
Martin Ohmeyer

Inhaltsverzeichnis

1	Aufgabe 3	1
1.1	Aufgabe 3.1	1
2	Aufgabe 6	2
2.1	Aufgabe 6.1	2
2.2	Aufgabe 6.2	2
3	Aufgabe 7	3
3.1	Aufgabe 7.1	3
3.2	Aufgabe 7.2	3
3.3	Aufgabe 7.3	3
3.4	Aufgabe 7.4	3
4	Aufgabe 9	4
4.1	Aufgabe 9.1	4
4.2	Aufgabe 9.2	4
4.3	Aufgabe 9.3	4
4.4	Aufgabe 9.4	4
5	Aufgabe 10	5
5.1	Aufgabe 10.1	5
6	Aufgabe 11	6
6.1	Aufgabe 11.1	6
6.2	Aufgabe 11.2	7
6.3	Aufgabe 11.3	8

1 Aufgabe 3

1.1 Aufgabe 3.1

2 Aufgabe 6

2.1 Aufgabe 6.1

2.2 Aufgabe 6.2

3 Aufgabe 7

3.1 Aufgabe 7.1

3.2 Aufgabe 7.2

3.3 Aufgabe 7.3

3.4 Aufgabe 7.4

4 Aufgabe 9

4.1 Aufgabe 9.1

4.2 Aufgabe 9.2

4.3 Aufgabe 9.3

4.4 Aufgabe 9.4

5 Aufgabe 10

5.1 Aufgabe 10.1

```
1  #include <b15f/b15f.h>
2
3  int main()
4  {
5      // Verbindung zum B15-Board aufbauen
6      B15F &drv = B15F::getInstance();
7
8      int mode = 0;
9      while (true)
10     {
11         // DIP Schalter auslesen um aktiven Zustand zu ermitteln
12         mode = drv.readDipSwitch();
13
14         if (mode == 0) // Zustand A
15         {
16             // Digitalen Eingang auf digitalen Ausgang invertieren
17             drv.digitalWrite0(255 - (int)drv.digitalRead0());
18         }
19         else if (mode == 1) // Zustand B
20         {
21             // Lauflicht in Richtung 1
22             for (int output = 64; output > 1; output /= 2)
23             {
24                 drv.digitalWrite0(output);
25                 drv.delay_ms(150);
26             }
27
28             // Lauflicht in Richtung 2
29             for (int output = 1; output <= 128; output *= 2)
30             {
31                 drv.digitalWrite0(output);
32                 drv.delay_ms(150);
33             }
34         }
35     }
36 }
```

6 Aufgabe 11

6.1 Aufgabe 11.1

Ausgewähltes Gatter: AND (DL008D)

6.2 Aufgabe 11.2

Verwendete Gatter: AND (DL008D), NOR (DL002D)

$$\neg((a \wedge b) \vee (c \wedge d))$$

```

1  #include <iostream>
2  #include <string>
3  #include <b15f/b15f.h>
4
5  using namespace std;
6
7  /// @brief Konvertiert eine Zahl von Dezimal in Binär
8  /// @param n Die natürliche Zahl, die konvertiert werden soll.
9  /// @return den Binären Wert von `n`
10 string toBinary(int n)
11 {
12     string r;
13     while (n != 0)
14     {
15         r = (n % 2 == 0 ? "0" : "1") + r;
16         n /= 2;
17     }
18     return r;
19 }
20
21 /// @brief Fordert die Eingabe einer Zahl an und prüft diese auf Kompatibilität
22 /// @param errorOnLastRun `true`, wenn der letzte Aufruf nicht erfolgreich war.
23 /// @return Die korrekte Eingabe als Zahl
24 int inputRequest(bool errorOnLastRun = false)
25 {
26     // Erzeugt eine Fehlermeldung bei unpassender Eingabe
27     if (errorOnLastRun)
28     {
29         cout << "\nDie Eingabe muss eine Ganzzahl sein, welche zur Reihe der
30             ↪ 2er-Potenzen gehört und nicht größer als 256 ist." << endl;
31         cin.clear();
32         cin.ignore(10000, '\n');
33     }
34     cout << "Anzahl der maximalen Zeilen: ";
35     int decimalInput = 0;
36
37     // Eingabe auf Zahl prüfen
38     if (!(cin >> decimalInput))
39     {
40         // Eingabe erneut abfragen
41         return inputRequest(true);
42     }
43
44     int i = 1;
45     // Fehlerhafte Eingabe mit nächst-größerer Zahl korrigieren
46     while (decimalInput <= 256 and decimalInput > i)
47     {
48         i *= 2;
49     }
50

```

6 Aufgabe 11

```
51 // Prüft die eingegebene Zahl auf Gültigkeit
52 return decimalInput == i ? decimalInput : inputRequest(true);
53 }
54
55 int main()
56 {
57     // Verbindung zum B15-Board aufbauen
58     B15F &drv = B15F::getInstance();
59
60     // Die Anzahl der zu testenden Ausgänge abfragen
61     int max = inputRequest();
62     cout << "\n\n";
63
64     for (int i = 0; i < max; i++)
65     {
66         // Aktuellen Testausgang konvertieren
67         string bin = toBinary(i);
68
69         // Führende Nullen ergänzen
70         while (bin.length() < (log(max) / log(2)))
71         {
72             bin = bin.insert(0, "0");
73         }
74
75         // Seperator für die Darstellung einfügen
76         int length = bin.length() * 4 - 3;
77         for (int j = 0; j < length; j += 4)
78         {
79             bin.insert(j, " | ");
80         }
81
82         // Tabellenkopf erzeugen
83         if (i == 0)
84         {
85             string letters = " | a | b | c | d | e | f | g | h |";
86             string header = letters.substr(0, bin.length() * 4 + 2);
87             cout << header << " | A |\n"
88                  << endl;
89         }
90
91         // Ausgang auf Eingang testen
92         drv.digitalWrite0(i);
93         int input = (int)drv.digitalRead0();
94
95         // Ergebnis ausgeben
96         cout << bin << " || " << input << " |" << endl;
97     }
98
99     return 0;
100 }
```

6.3 Aufgabe 11.3