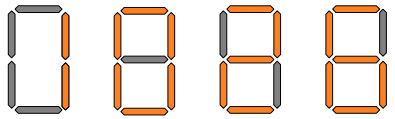
7-Segmentanzeige



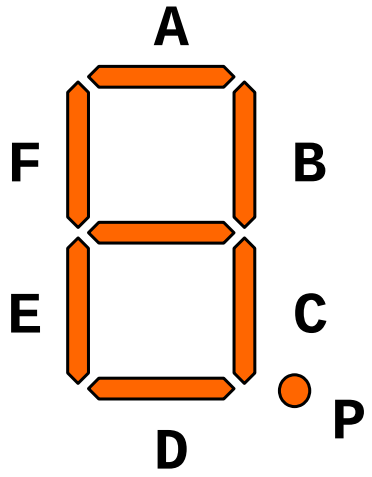
7-Segment Anzeige mit 4 Leitungen/Bits steuern

Eine 7-Segmentanzeige soll mindestens alle Ziffern 0-9 darstellen können. Dafür brauchen wir mindestens 4 Bits. Wegen 2^4=16 bleiben noch 6 Zustände übrig.

Es ist praktisch, wenn die Anzeige noch ein Minus „-“, ein „E“ und „r“ für „Er“=Error anzeigen kann.

1) Füllen Sie die Tabelle auf der nächsten Seite aus, so dass für jedes Bitmuster der Eingabevariablen W-Z die entsprechende Ziffer angezeigt wird.

   Dabei ist wichtig: Im Fall des Basys3 Boards schaltet man die einzelnen Segment mit 1=aus und 0=ein.

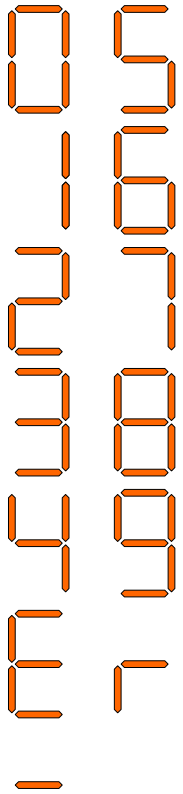


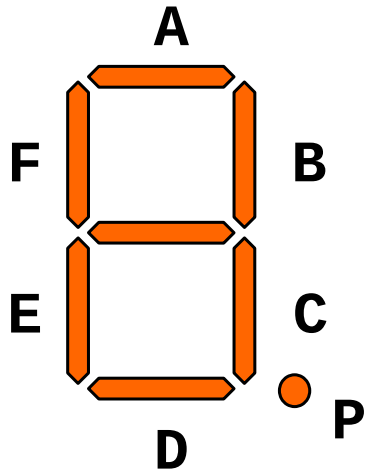
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

7-Segmentanzeige

Wahrheitstabelle (0 = ein/1 = aus)

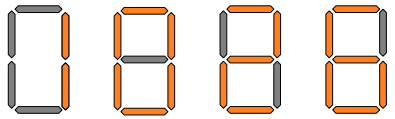
Z	Y	X	W	A	B	C	D	E	F	G	CHAR
0	0	0	0	–	–	–	–	–	–	–	0
0	0	0	1	–	–	–	–	–	–	–	1
0	0	1	0	–	–	–	–	–	–	–	2
0	0	1	1	–	–	–	–	–	–	–	3
0	1	0	0	–	–	–	–	–	–	–	4
0	1	0	1	–	–	–	–	–	–	–	5
0	1	1	0	–	–	–	–	–	–	–	6
0	1	1	1	–	–	–	–	–	–	–	7
1	0	0	0	–	–	–	–	–	–	–	8
1	0	0	1	–	–	–	–	–	–	–	9
1	0	1	0	–	–	–	–	–	–	–	E
1	0	1	1	–	–	–	–	–	–	–	r
1	1	1	1	–	–	–	–	–	–	–	–





© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

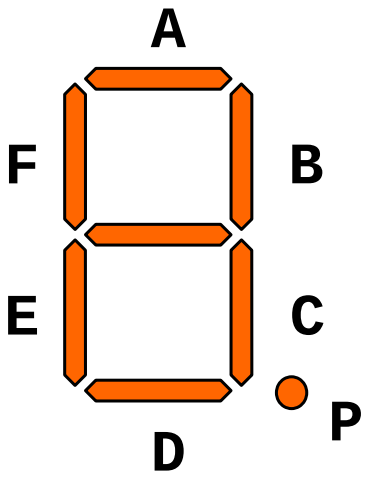
7-Segmentanzeige



7-Segment Anzeige mit 4 Leitungen/Bits steuern

2) Vereinfachen Sie die 7 Logikfunktionen für jedes Segment. Sie können dabei arbeitsteilig vorgehen.

3) Schreiben Sie nun ein Modul, das eine 4-Bit Ziffer auf einer Anzeige ausgibt. Testen Sie das Modul in einem Hauptmodul in dem Sie Schalter verwenden. (siehe nächste Seite)



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

7-Segmentanzeige

```
module sev_seg(
    input HW_switch_0, HW_switch_1, HW_switch_2,
    input HW_switch_3, HW_switch_4, HW_switch_5,
    input HW_switch_6,

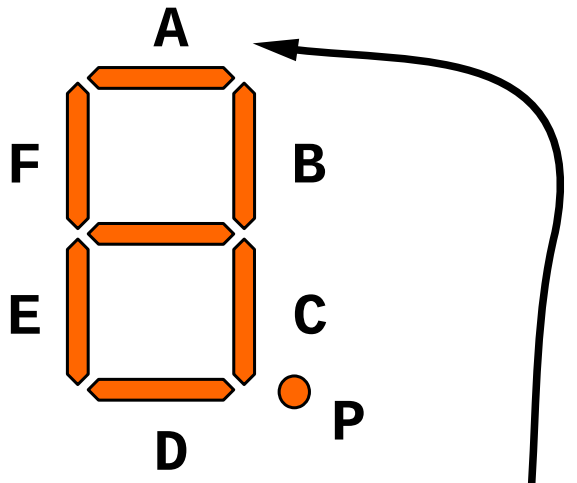
    // Segmenete A-G
    output SSD_CA, SSD_CB, SSD_CC,
    output SSD_CD, SSD_CE, SSD_CF, SSD_CG,

    // Digits 0-3
    output SSD_AN0, SSD_AN1, SSD_AN2, SSD_AN3
    );

    // Test der 7-Segment Anzeige

    assign SSD_AN0 = 0; // 1 Stelle ein
    assign SSD_AN1 = 1; // 2 Stelle aus
    assign SSD_AN2 = 1; // 3 Stelle aus
    assign SSD_AN3 = 1; // 4 Stelle aus

    assign SSD_CA = HW_switch_0;
    assign SSD_CB = HW_switch_1;
    assign SSD_CC = HW_switch_2;
    assign SSD_CD = HW_switch_3;
    assign SSD_CE = HW_switch_4;
    assign SSD_CF = HW_switch_5;
    assign SSD_CG = HW_switch_6;
endmodule
```



Portnamen:

```
output SSD_CA,
…
output SSD_CG,
```
Segmente

```
output SSD_AN0,
…
output SSD_AN3
```
Digits

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.