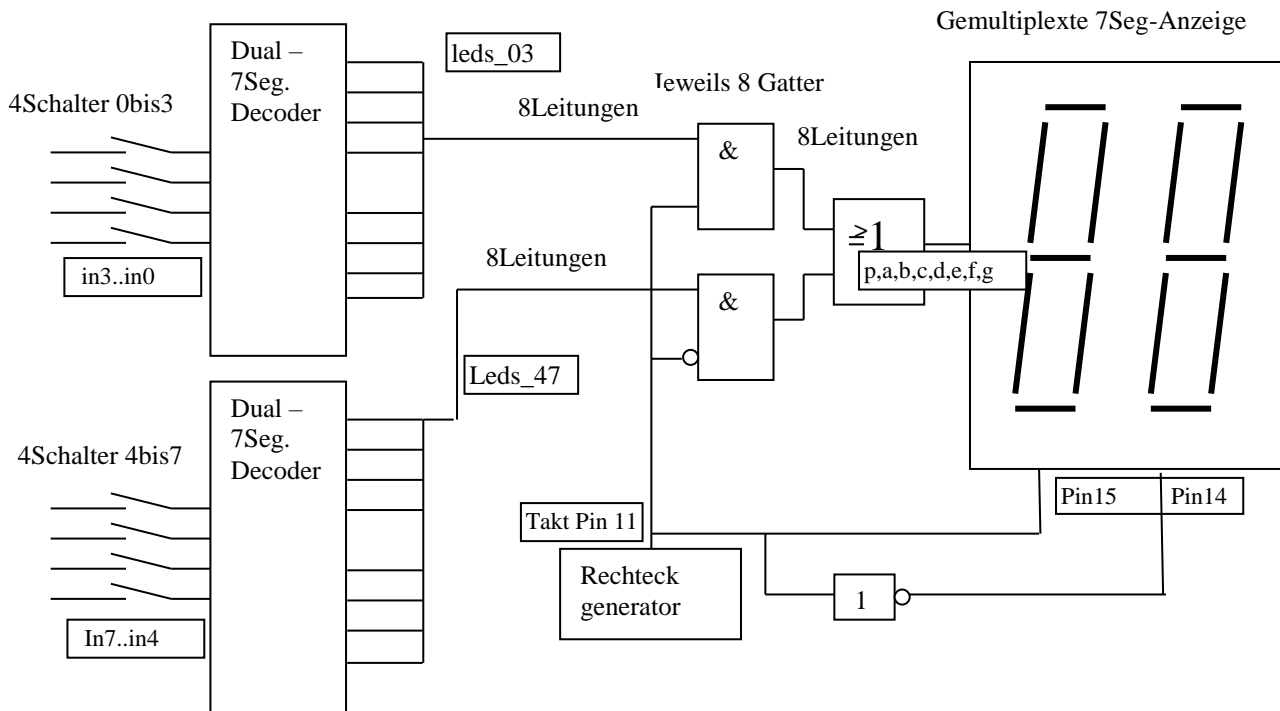




Gemultiplexte 7 Segment-Anzeige Programmierbare Logik

Aufgabenstellung: Sie sollen 8bit als Hex-Zahl auf einer 2 stelligen 7 Segment-Anzeige ausgeben.
Verwirklicht wird diese Aufgabe mit einer programmierbaren Logik.

Das dargestellte Blockschaltbild zeigt die Realisierung mit einer gemultiplexten 7 Segment – Anzeige (Es leuchtet immer nur eine Anzeige). Die Decoderwerte von Schalter 0bis3 werden zuerst auf die Anzeige gegeben, dabei ist die rechte Anzeige eingeschaltet. Danach werden die Decoderwerte von Schalter 4bis7 auf die Anzeige geschaltet, dabei wird die linke Anzeige aktiviert. Dieser Vorgang wiederholt sich ständig. Wenn das Umschalten schnell genug ist, dann sehen wir beide Anzeigen gleichzeitig leuchten, obwohl natürlich immer nur eine Anzeige wirklich eingeschaltet ist.



Für das dargestellte Blockschaltbild ist die Lösung in der Sprache ABEL dargestellt:

```
MODULE seg_GW_OG_2x
```

```
TITLE '2x4bit auf 7Seg'
```

```
" Benutzer Baustein: MACH4A5 der Device Family: Mach4A  
" Codeumsetzer Binaer-Zahl -> 7-Segment-Anzeige  
" Die Anzeige der Bits 0 bis 3 erfolgt auf EinerHex  
" Die Anzeige der Bits 4 bis 7 erfolgt auf Zehner Hex  
" Die 7Segment Anzeige muss gemultiplext werden, die Anzeigen werden über die  
" Pins 14 und 15 ein bzw ausgeschaltet
```

```
DECLARATIONS "***** Ein- und Ausgänge ***Gew Techn Schule_Offenburg***Neumaier**
```

```
//Hinweis: Jumper auf der 7Segment-Platine ziehen. Somit Multiplexbetrieb.
```

```
Takt pin 11; // Ein Taktgenerator an die Logik anschliessen
```

```
in3..in0 PIN 5,4,3,2; "Eingänge Port0 o..3
```

```
in7..in4 PIN 9,8,7,6; "Eingänge Port0 4..7
```

```
p,a,b,c,d,e,f,g PIN 31,30,29,28,27,26,25,24 ISTYPE'BUFFER,COM'; "Anzeige-Ausgänge
```

```
p47,p46,p45,p44,p43,p42,p41,p40 PIN 43,42,41,40,39,38,37,36 ISTYPE'BUFFER,COM'; "LED-Ausgänge
```

```
"ohne Flipflops Port 2
```

```
EinerHex pin 14 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl Port1
```

```
ZehnerHex pin 15 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl
```

```
p1,a1,b1,c1,d1,e1,f1,g1 NODE ISTYPE'BUFFER,COM'; "Ausgänge Decoder1
```



```
P1,A1,B1,C1,D1,E1,F1,G1 NODE ISTYPE'BUFFER,COM'; "Ausgänge Decoder2
"***** SET Deklarationen *****
dipschalter03 = [in3..in0];          "Zusammenfassung der Eingänge
dipschalter47 = [in7..in4];          "Zusammenfassung der Eingänge
leds_aus      = [p,a,b,c,d,e,f,g];  "Zusammenfassung der Ausgänge
leds_03       = [p1,a1,b1,c1,d1,e1,f1,g1]; "Zusammenfassung der Ausgänge
leds_47       = [P1,A1,B1,C1,D1,E1,F1,G1]; "Zusammenfassung der Ausgänge

Port4highnib = [p47,p46,p45,p44]; "Zusammenfassung der Ausgänge
Port4lownib  = [p43,p42,p41,p40];

EQUATIONS "*****
leds_aus= (leds_03&Takt)#(leds_47&!Takt); " Immer nur die Ausgänge von einem Decoder durchschalten
EinerHex=!Takt;
ZehnerHex=Takt;
Port4highnib = dipschalter47;
Port4lownib= dipschalter03;
TRUTH_TABLE "***** Funktionstabelle *****Decoder1*****
(dipschalter03 -> [p1,a1,b1,c1,d1,e1,f1,g1 ]) "Eingänge -> Ausgänge
 0 -> [0, 0, 1, 1, 1, 1, 1, 1]; " LEDs high-aktiv!
 1 -> [0, 0, 0, 0, 0, 1, 1, 0]; " g
 2 -> [0, 1, 0, 1, 1, 0, 1, 1]; " ---
 3 -> [0, 1, 0, 0, 1, 1, 1, 1]; " b | a | f
 4 -> [0, 1, 1, 0, 0, 1, 1, 0]; " ---
 5 -> [0, 1, 1, 0, 1, 1, 0, 1]; " c | d | e
 6 -> [0, 1, 1, 1, 1, 1, 0, 1]; " --- . p
 7 -> [0, 0, 0, 0, 0, 1, 1, 1];
 8 -> [0, 1, 1, 1, 1, 1, 1, 1]; " Fuer Hardware GWS_OG
 9 -> [0, 1, 1, 0, 1, 1, 1, 1];
10 -> [0, 1, 1, 1, 0, 1, 1, 1];
11 -> [0, 1, 1, 1, 1, 1, 0, 0];
12 -> [0, 0, 1, 1, 1, 0, 0, 1];
13 -> [0, 1, 0, 1, 1, 1, 1, 0];
14 -> [0, 1, 1, 1, 1, 0, 0, 1];
15 -> [0, 1, 1, 1, 0, 0, 0, 1];

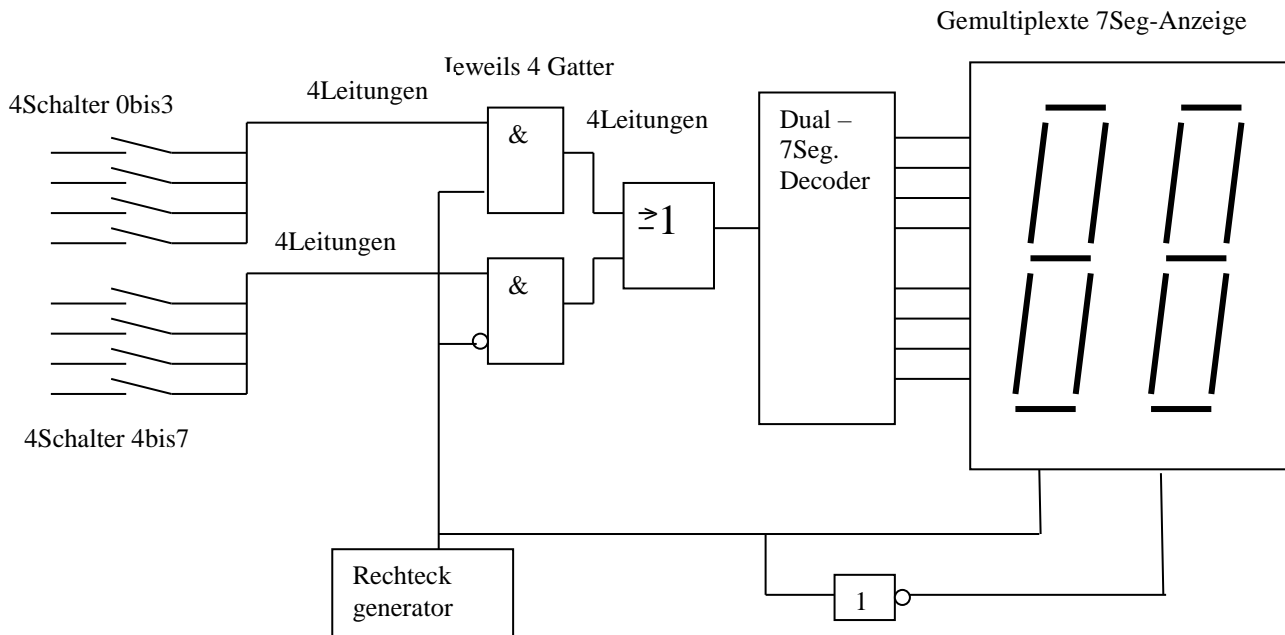
TRUTH_TABLE "***** Funktionstabelle *****Decoder2*****
(dipschalter47 -> [P1,A1,B1,C1,D1,E1,F1,G1]) "Eingänge -> Ausgänge
 0 -> [0, 0, 1, 1, 1, 1, 1, 1]; " LEDs high-aktiv!
 1 -> [0, 0, 0, 0, 0, 1, 1, 0]; " g
 2 -> [0, 1, 0, 1, 1, 0, 1, 1]; " ---
 3 -> [0, 1, 0, 0, 1, 1, 1, 1]; " b | a | f
 4 -> [0, 1, 1, 0, 0, 1, 1, 0]; " ---
 5 -> [0, 1, 1, 0, 1, 1, 0, 1]; " c | d | e
 6 -> [0, 1, 1, 1, 1, 1, 0, 1]; " --- . p
 7 -> [0, 0, 0, 0, 0, 1, 1, 1];
 8 -> [0, 1, 1, 1, 1, 1, 1, 1]; " Fuer Hardware GWS_OG
 9 -> [0, 1, 1, 0, 1, 1, 1, 1];
10 -> [0, 1, 1, 1, 0, 1, 1, 1];
11 -> [0, 1, 1, 1, 1, 1, 0, 0];
12 -> [0, 0, 1, 1, 1, 0, 0, 1];
13 -> [0, 1, 0, 1, 1, 1, 1, 0];
14 -> [0, 1, 1, 1, 1, 0, 0, 1];
15 -> [0, 1, 1, 1, 0, 0, 0, 1];

// Bit fuer die 7Seg Anzeige von 0 bis F Hardware GWS_Offenburg
// 00111111b,00000110b,01011011b,01001111b,01100110b,01101101b,11111101b,
// 00000111b,01111111b,01101111b,01110111b,01111100b,00111001b,01011110b,
// 01111001b,01110001b

END
```



Sie sollen nun die Aufgabe nach folgendem Blockschaltbild lösen und ein Programm dafür entwickeln. Sie ändern das gegebene Programm entsprechend ab.



Weitere Aufgaben:

- Ersetzen Sie die Schalter 0bis3 und 4bis7 durch 2 BCD-Zähler (0 bis 15). Somit haben Sie einen 2 stelligen Zähler von 0 bis FF in Hex.
- Ersetzen Sie die Schalter 0bis3 und 4bis7 durch 2 BCD-Zähler (0 bis 9). Somit haben Sie einen 2 stelligen Zähler von 0 bis 99 dezimal.
- Wenn Sie den Takt 1Sekunde machen, dann haben Sie einen Sekundenzähler. Sie benötigen dafür einen zweiten Rechteckgenerator.
- Ändern Sie den Zähler so, dass er bei 59 wieder bei 0 beginnt.
- Erweitern Sie die Sekundenanzeige um 2 weitere Stellen für die Minutenanzeige. Beachten sie untenstehendes Blockschaltbild für die 4x7Seg-Anzeige.
- Erweitern Sie die Uhr zu einer Stunden- und Minutenanzeige. Teilen Sie dazu den Takt intern durch 60. Dann können die Uhr mit 50Hz takten. Diese Frequenz können sie aus den 50 Hz der 230V/400V Netzspannung generieren. Wie ?
- Dokumentieren Sie den Werdegang ihrer Uhr und präsentieren Sie ihre Lösung. Vielleicht gelingt es ihnen auch die untenstehenden Fragen zu integrieren!

Wie könnte man die Uhr stellen??

Wie könnte man einen Wecker kreieren, der immer bei einer festen Zeit Alarm gibt?

Wie könnte man den Wecker einstellbar machen?

Sie sehen: Fragen über Fragen!!

Lösungsvorschlag 1 Aufgabe 2stellig, nur ein 7seg-Decoder

Lösung mit 1 Zähler 0 bis 15 und 4 Schalter

```

MODULE seg_GW_OG_2x
TITLE '2x4bit auf 7Seg'
"Codeumsetzer Binaer-Zahl -> 7-Segment-Anzeige
" Die Anzeige der Bits 0 bis 3 erfolgt auf EinerHex
" Die Anzeige der Bits 4 bis 7 erfolgt auf Zehner Hex
" Die 7Segment Anzeige muss gemultiplixt werden, die Anzeigen werden über die
" Pins 14 und 15 ein bzw ausgeschaltet

DECLARATIONS "***** Ein- und Ausgänge ***Gew Techn Schule_Offenburg***Neumaier**

//Hinweis: Jumper auf der 7Segment-Platine setzen. Somit die letzte Anzeige
//dauerhaft aktiv. Kein Multiplexbetrieb. Nur die Datenverbindung notwendig.
Takt pin 33;
takt2          PIN 11; "takt zaehler
in7..in4       PIN 9,8,7,6;
bit3..bit0     node ISTYPE'BUFFER,REG'; "Zählerausgänge
               "Eingänge Port0 4..7 ersetzt mit bit3..bit0 ->4bit Dualzähler
               p,a,b,c,d,e,f,g PIN 31,30,29,28,27,26,25,24 ISTYPE'BUFFER,COM'; "Anzeige-Ausgänge
               "ohne Flipflops Port 2
EinerHex pin 14 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl
ZehnerHex pin 15 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl
dec3, dec2, dec1, dec0 NODE ISTYPE'BUFFER,COM'; "Ausgänge Decoder1
dipschalter03 = [bit3..bit0];          "Zusammenfassung der Eingänge
dipschalter47 = [in7..in4];          "Zusammenfassung der Eingänge

dipschalter = [dec3, dec2, dec1, dec0];"Zusammenfassung der Ausgänge
ctr = [bit3..bit0]; "Takteingang Y0

EQUATIONS  `4bit Dualzähler in Hochsprache
ctr := ctr + 1;
ctr.c = takt2;

EQUATIONS "*****
dipschalter= (dipschalter03&Takt)#(dipschalter47&!Takt); " Immer nur die Ausgänge von einem
" Decoder durchschalten

EinerHex=!Takt;
ZehnerHex=Takt;
TRUTH_TABLE "***** Funktionstabelle *****Decoder1*****

(dipschalter -> [p, a, b, c, d, e, f, g ]) "Eingänge -> Ausgänge
0 -> [0, 0, 1, 1, 1, 1, 1, 1]; " LEDs high-aktiv!
1 -> [0, 0, 0, 0, 0, 1, 1, 0]; " a
2 -> [0, 1, 0, 1, 1, 0, 1, 1];" ---
3 -> [0, 1, 0, 0, 1, 1, 1, 1];" b | g | f
4 -> [0, 1, 1, 0, 0, 1, 1, 0];" ---
5 -> [0, 1, 1, 0, 1, 1, 0, 1];" c | d | e
6 -> [0, 1, 1, 1, 1, 1, 0, 1];" --- . p
7 -> [0, 0, 0, 0, 0, 0, 1, 1];
8 -> [0, 1, 1, 1, 1, 1, 1, 1];" Fuer Hardware GWS_OG
9 -> [0, 1, 1, 0, 1, 1, 1, 1];
10 -> [0, 1, 1, 1, 0, 1, 1, 1];
11 -> [0, 1, 1, 1, 1, 1, 0, 0];
12 -> [0, 0, 1, 1, 1, 0, 0, 1];
13 -> [0, 1, 0, 1, 1, 1, 1, 0];
14 -> [0, 1, 1, 1, 1, 0, 0, 1];
15 -> [0, 1, 1, 1, 0, 0, 0, 1];

```

END



Lösung mit 2 Zählern jeweils 0 bis 15

```

bit3..bit0  node ISTYPE'BUFFER,REG'; "Zählerausgänge
            "Eingänge Port0 4..7
    p,a,b,c,d,e,f,g PIN 31,30,29,28,27,26,25,24 ISTYPE'BUFFER,COM'; "Anzeige-Ausgänge
            "ohne Flipflops Port 2
EinerHex pin 14 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl
ZehnerHex pin 15 ISTYPE'BUFFER,COM'; "Anzeige-Auswahl
    dec3, dec2, dec1, dec0 NODE ISTYPE'BUFFER,COM'; "Ausgänge Decoder1

dipschalter =[dec3, dec2, dec1, dec0];

    dipschalter03 = [bit3..bit0];          "Zusammenfassung der Eingänge
dipschalter47 = [bit7..bit4 ];          "Zusammenfassung der Eingänge
ctr = [bit3..bit0]; "Takteingang Y0
ctrz = [bit7..bit4]; "

EQUATIONS
    ctr := ctr + 1;
    ctr.c = takt2;

    ctrz := ctrz + 1;
    ctrz.c = !bit3;

EQUATIONS "*****
dipschalter= (dipschalter03&Takt)#(dipschalter47&!Takt); " Immer nur die Ausgänge von einem
Decoder durchschalten
EinerHex=!Takt;
ZehnerHex=Takt;
TRUTH_TABLE "***** Funktionstabelle *****Decoder1*****

(dipschalter -> [p, a, b, c, d, e, f, g ]) "Eingänge -> Ausgänge
    0 -> [0, 0, 1, 1, 1, 1, 1, 1]; " LEDs high-aktiv!
    1 -> [0, 0, 0, 0, 0, 1, 1, 0]; " a
    2 -> [0, 1, 0, 1, 1, 0, 1, 1];" ---
    3 -> [0, 1, 0, 0, 1, 1, 1, 1];" b | g | f
    4 -> [0, 1, 1, 0, 0, 1, 1, 0];" ---
    5 -> [0, 1, 1, 0, 1, 1, 0, 1];" c | d | e
    6 -> [0, 1, 1, 1, 1, 1, 0, 1];" --- . p
    7 -> [0, 0, 0, 0, 0, 1, 1, 1];
    8 -> [0, 1, 1, 1, 1, 1, 1, 1];" Fuer Hardware GWS_OG
    9 -> [0, 1, 1, 0, 1, 1, 1, 1];
    10 -> [0, 1, 1, 1, 0, 1, 1, 1];
    11 -> [0, 1, 1, 1, 1, 1, 0, 0];
    12 -> [0, 0, 1, 1, 1, 1, 0, 0, 1];
    13 -> [0, 1, 0, 1, 1, 1, 1, 0];
    14 -> [0, 1, 1, 1, 1, 1, 0, 0, 1];
    15 -> [0, 1, 1, 1, 0, 0, 0, 1];

END

```