Lösung Versuch Nr. 5 Automaten

1: 3-Phasen Takt Generator

1. *Wie viele Zustände muss der Automat mindestens haben und wie viele Flip-Flops werden mindestens für den Zustandsspeicher benötigt?*

   Minimale Anzahl Zustände = 4 Minimale Anzahl Flip-Flops = 2

   Ein Moore-Automaten muss mindestens soviele Zustände haben wie Kompinationen der Ausgangssignale existieren, da die Ausgänge nur eine Funktion des Zustandes sind. Jede Phase (A, B, C) und „aus“ haben andere Ausgangssignale. Dies ergibt mindestens und hinreichend 4 Zustände (jeder Asgangsvektor kommt im Ablauf nur genau einmal vor). Mit 2 Flip-Flops können gerade 4 Zustände codiert werden.

2. *Zeichnen Sie das Zustandsdiagramm für den Automaten auf. Zeichnen Sie darin alle eindeutig durch die Aufgabenstellung gegebenen Übergänge zwischen den Zuständen ein.*

3. *Ergänzen Sie das Zustandsdiagramm mit sinnvollen Übergängen aus der Betriebsart "aus" in die anderen Betriebsarten.*

3-Phasen Takt Generator Zustandsdiagramm

4. *Ordnen Sie jedem Zustand einen Zustandscode zu und stellen Sie die Wahrheits-tabelle für die kombinatorische Transition-Logik auf.*

   Die naheliegendste Art der Zustandscodierung ist

   Zustandscode = Zustandsnummer

```
% transition logic                                  %
% CS = Current State,  NS = Next State             %
 Q1, Q0,  f,  r  % CS %  =>  % NS %  D1, D0 ;
%=================================================%
  0,  0,  0,  0  %  0 %  =>  %  0 %   0,  0 ;
  0,  0,  0,  1  %  0 %  =>  %  3 %   1,  1 ;
  0,  0,  1,  0  %  0 %  =>  %  1 %   0,  1 ;
  0,  0,  1,  1  %  0 %  =>  %  1 %   0,  1 ;
%-------------------------------------------------%
  0,  1,  0,  0  %  1 %  =>  %  0 %   0,  0 ;
  0,  1,  0,  1  %  1 %  =>  %  3 %   1,  1 ;
  0,  1,  1,  0  %  1 %  =>  %  2 %   1,  0 ;
  0,  1,  1,  1  %  1 %  =>  %  1 %   0,  1 ;
%-------------------------------------------------%
  1,  0,  0,  0  %  2 %  =>  %  0 %   0,  0 ;
  1,  0,  0,  1  %  2 %  =>  %  1 %   0,  1 ;
  1,  0,  1,  0  %  2 %  =>  %  3 %   1,  1 ;
  1,  0,  1,  1  %  2 %  =>  %  2 %   1,  0 ;
%-------------------------------------------------%
  1,  1,  0,  0  %  3 %  =>  %  0 %   0,  0 ;
  1,  1,  0,  1  %  3 %  =>  %  2 %   1,  0 ;
```

```
1,  1,  1,  0  %  3 %  =>  %  1 %   0,  1 ;
1,  1,  1,  1  %  3 %  =>  %  3 %   1,  1 ;
```

Wahrheitstabelle für die Transision Logik in AHDL Schreibweise

Achtung: Gegenüber der Vorlesung (Zustandsfolgetabelle) sind die Spalten *Eingang X* (f, r) und *momentaner Zustand* (Q1, Q0) vertauscht angeordnet. Dadurch liegen alle Übergänge aus einem Zustand in aufeinanderfolgenden Zeilen, der Zusammenhang mit dem Zustandsdiagramm wird offensichtlicher und dank der Hilfsspalten unübersehbar.

5. Stellen Sie die Wahrheitstabelle für die kombinatorsiche Ausgangslogik auf.

```
% output logic                    %
% CS %  Q1, Q0  => !nA, !nB, !nC;
%================================%
%  0 %   0, 0   =>   0,   0,   0;
%  1 %   0, 1   =>   1,   0,   0;
%  2 %   1, 0   =>   0,   1,   0;
%  3 %   1, 1   =>   0,   0,   1;
```

Wahrheitstabelle für die Output Logik in AHDL Schreibweise

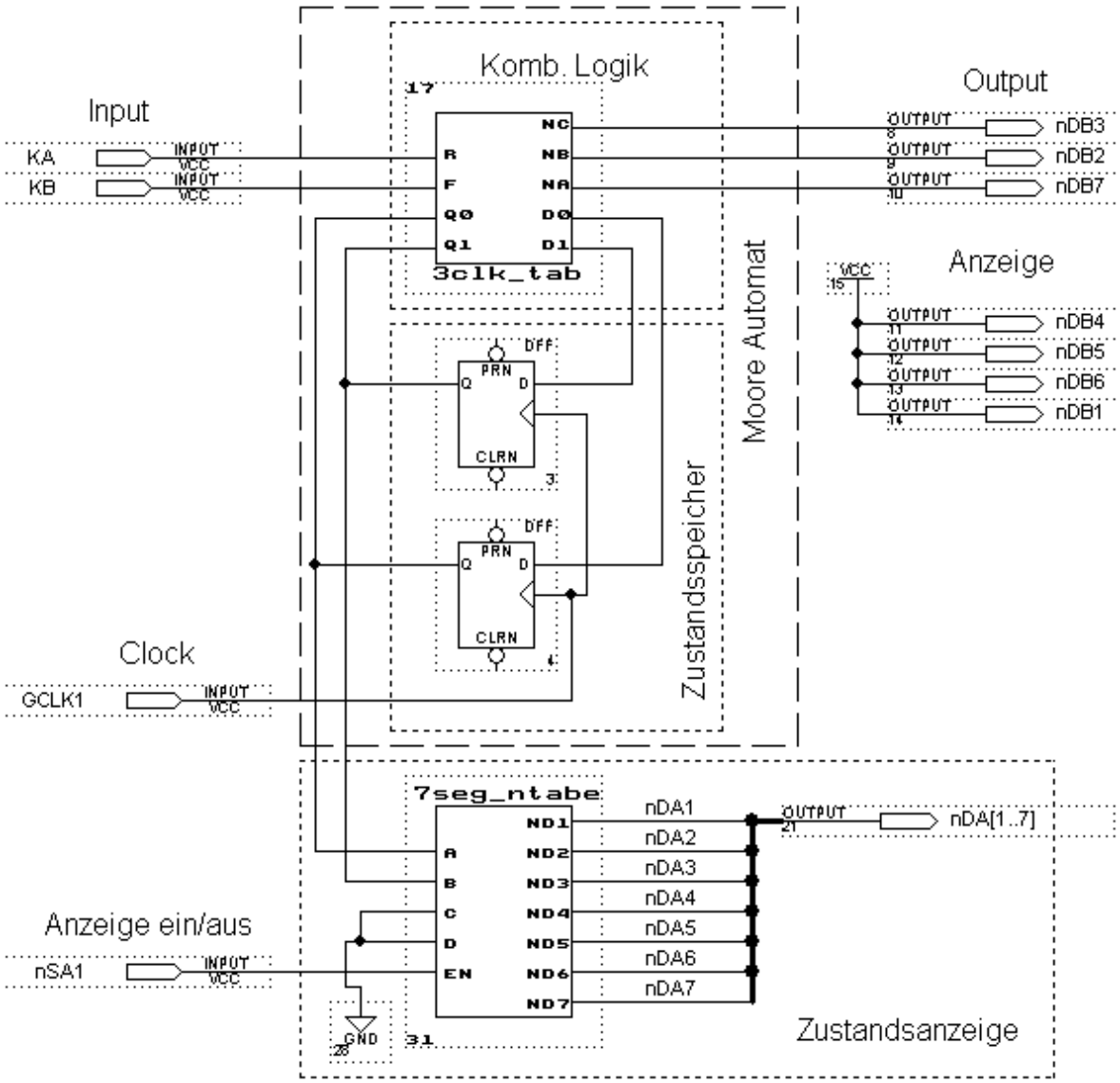
6. Realisieren Sie den 3-Phasen Taktgenerator auf dem PLD-Board.

Schema folgt auf der nächsten Seite mit integrierter Lösung der Aufgabe 7. Da die Transition- und Ausgangslogik mite je einer Tabelle im gleichen File (als ein Subdesign) definiert wurden, ensteht nur ein Symbol für beide.

Fakultative Aufgaben:

7. *Über einen 7-Segment Decoder auf der Anzeige DA den Zustand anzeigen.*

Ist bereits im Schema auf der nächsten Seite integriert. Der 7-Segment Decoder ist als Luxusausführung mit einem Enable Eingang erweitert, damit die Anzeige mit dem Schalter SA1 ausgeschaltet werden kann.



3-Phasen Takt Generator als Moore Automat mit fakultativer Zustandsanzeige

.

```
% 3clk_tab.tdf  3-Phase Clock Generator %
% Transision- and Output- Logic tables  %
% 20.12.02 Zi %
SUBDESIGN 3clk_tab
( Q1, Q0, f, r        : INPUT;
  D1, D0, nA, nB, nC  : OUTPUT; )
BEGIN
TABLE  % transition logik %
% CS = Current State,  NS = Next State %
 Q1, Q0,  f,  r  % CS %  =>  % NS %  D1, D0 ;
%=========================================%
  0,  0,  0,  0  %  0 %  =>  %  0 %   0,  0 ;
  0,  0,  0,  1  %  0 %  =>  %  3 %   1,  1 ;
  0,  0,  1,  0  %  0 %  =>  %  1 %   0,  1 ;
  0,  0,  1,  1  %  0 %  =>  %  2 %   1,  0 ;
%-----------------------------------------%
```

```
  0,  1,  0,  0  %  1 %  =>  %  0 %   0,  0 ;
  0,  1,  0,  1  %  1 %  =>  %  3 %   1,  1 ;
  0,  1,  1,  0  %  1 %  =>  %  2 %   1,  0 ;
  0,  1,  1,  1  %  1 %  =>  %  1 %   0,  1 ;
%-----------------------------------------%
  1,  0,  0,  0  %  2 %  =>  %  0 %   0,  0 ;
  1,  0,  0,  1  %  2 %  =>  %  1 %   0,  1 ;
  1,  0,  1,  0  %  2 %  =>  %  3 %   1,  1 ;
  1,  0,  1,  1  %  2 %  =>  %  2 %   1,  0 ;
%-----------------------------------------%
  1,  1,  0,  0  %  3 %  =>  %  0 %   0,  0 ;
  1,  1,  0,  1  %  3 %  =>  %  2 %   1,  0 ;
  1,  1,  1,  0  %  3 %  =>  %  1 %   0,  1 ;
  1,  1,  1,  1  %  3 %  =>  %  3 %   1,  1 ;
END TABLE;
TABLE   % output logic %
% CS = Current State %
% CS %  Q1, Q0  =>  !nA,  !nB,  !nC;
%  0 %   0, 0   =>   0,    0,    0;
%  1 %   0, 1   =>   1,    0,    0;
%  2 %   1, 0   =>   0,    1,    0;
%  3 %   1, 1   =>   0,    0,    1;
END TABLE;
END;
```

AHDL Text-Design-File 3clk_tab.tdf mit Transition- und Output- Logik

Ergänzung für Spezialisten: Entwurf als Medwedjew Automat

Der Taktgenerator kann auch als Medwedjew-Automat (alos ohne Ausgangslogik) entworfen werden. Dies ist möglich, weil jedes Muster der Ausgangssignale nur einmal vorkommt. Als Zustandsspeicher müssen dabei 3 Flip-Flops eingesetzt werden, da jeder der 3 Ausgänge direkt mit einem Flip-Flop Ausgang verbunden werden muss.

Es ergeben sich total 8 Zustände des Automaten. Vier davon werden nicht benützt. Achtung: Bei einem seriösen Etwurf muss von jedem der unbenützten Zustände ein unbedingter Übergang auf einen der benützten vorgesehen werden, sonst könnte sich der Automat bei einer Störung in den unbenützten Zuständen “aufhängen”.

Bei der Realisierung in einem PLD werden für den Medwedjew-Automaten sogar weniger Resourcen verbraucht, nämlich nur 3 PAL-Zellen (4% unseres PLDs), gegenüber 5 (7%) beim Entwurf mit minimaler Anzahl (zwei) Flip-Flops und Ausgangslogik (2 Zellen für die Transitionlogik mit Flip-Flops und zusätzlich 3 weitere für die Ausgangslogik).

Achtung: werden die unbenützten (dunklen) Segemente der Anzeige richtigerweise auf Vcc gelegt, so wird dazu für jedes Segment eine weitere PAL-Zelle belegt, deren Logik und Flip-Flop gar nicht gebraucht wird!

Den Medwedjew-Automaten haben David von Allmen und 2 weitere Gruppen im HS08 realisiert.

2: Audio Sequence Player

8. *Wie funktioniert der programmierbare Frequenzteiler (Zähler) im Ton Generator (Bild 2.1) und welche Aufgabe hat das T-Flip-Flop am Ausgang?*

   Der Zähler wird über den asynchronen Loadeingang (LDN, aktiv low) auf den Eingängen (A..H) angelegten Teilerwert gesetzt und zählt von diesem anschliessen rückwärts bis auf 0. Dieser (End-)Wert wird mit dem AND-Tor mit invertierten Eingängen detektiert. Der Zähler wird dabei über den Inverter wieder neu auf den Anfangswert gesetzt.

   Am Ausgang des AND-Tores entsteht nur ein kurzer Impuls. Mit dem T-Flip-Flop wird daraus ein Rechteck mit halber Frequenz erzeugt. Das Flip-Flop wechselt den Zustand bei einer aktiven Taktflanke nur dann, wenn T gleich 1 ist, also nur beim Zählerstand 0.

9. *Entwerfen Sie das Schema des Noten Generators (note_gen.gdf).*

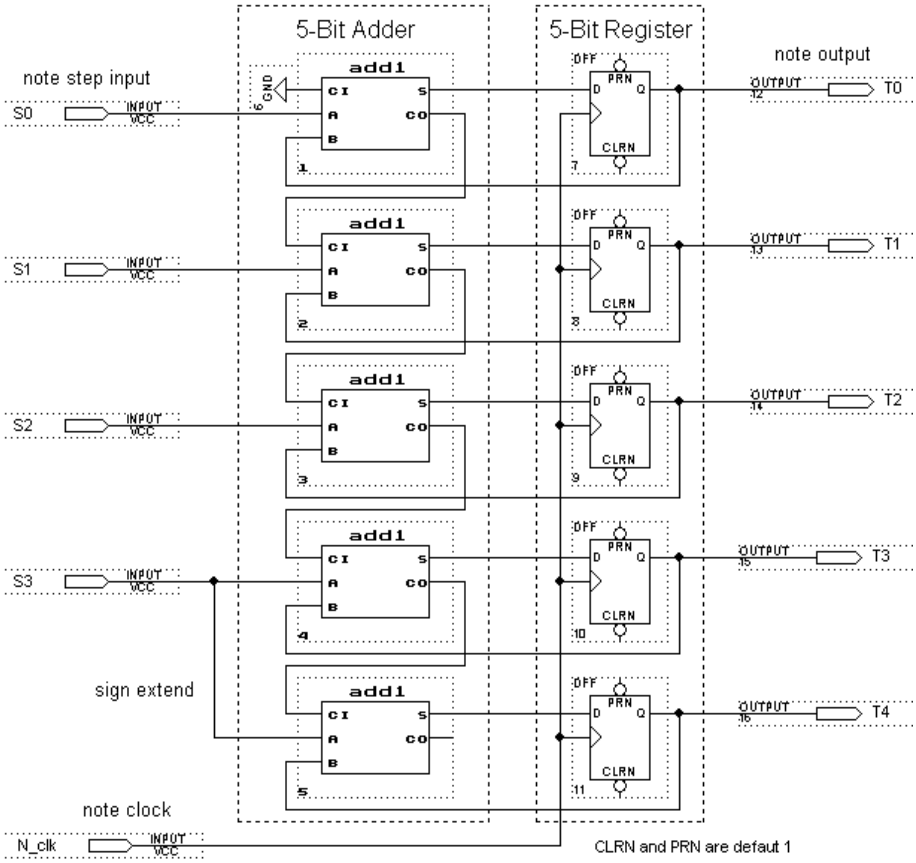
   Siehe Schema auf der nächsten Seite.

   Man überlege sorgfältig, Wie das fünfte (höchstwertige) Bit des Addierer-eingangs beschaltet werden muss.

   Zum Expandieren einer 2er Komplementzahl muss das Vorzeichenbit in alle höherwertigeren Bits (hier nur 1, das fünfte) kopiert werden.

10. *Was für ein Automatentyp ist der Notengenerator (note_gen)?*

    Es ist ein **Medwedjew** Automat, ein Sonderfall des **Moore** Automaten.



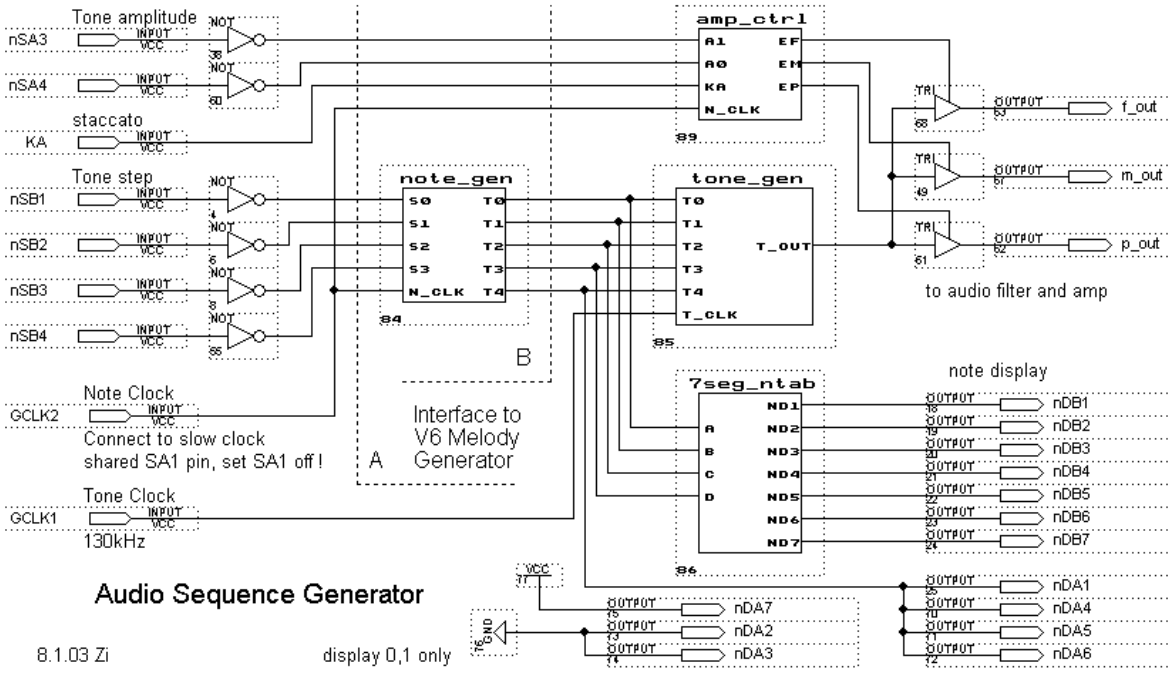
Noten Generator

Fakultative Aufgabe:

11. *Entwerfen Sie an Stelle des Läutstärke-Kontrollblocks (vol_ctrl) einen Amplituden Kontrollblock (amp_ctrl)*

```
% Amplitude control logic table %
% 11.7.02, 24.9, 10.1.03 Zi     %
SUBDESIGN amp_tab
(   A1, A0   : INPUT;    % amplitude code             %
       KA    : INPUT;    % staccato key               %
    n_clk    : INPUT;    % note clock                 %
 ef, em, ep  : OUTPUT; ) % enable forte, mezzo, piano %
BEGIN
TABLE
 KA, n_clk, A1, A0   =>  ef, em, ep  ;   % column titles %
  0,    x,    0,  0   =>   0,  0,  0  ;   % off      %
  0,    x,    0,  1   =>   0,  0,  1  ;   % piano    %
  0,    x,    1,  0   =>   0,  1,  0  ;   % mezzo    %
  0,    x,    1,  1   =>   1,  0,  0  ;   % forte    %
  1,    0,    x,  x   =>   0,  0,  0  ;   % staccato %
  1,    1,    0,  0   =>   0,  0,  0  ;   %  off     %
  1,    1,    0,  1   =>   0,  0,  1  ;   %  piano   %
  1,    1,    1,  0   =>   0,  1,  0  ;   %  mezzo   %
  1,    1,    1,  1   =>   1,  0,  0  ;   %  forte   %
END TABLE;
END;
```

Amplituden Kontroll- Tabelle (amp_tab.tdf)



Audio Sequenz Generator mit Amplituden Kontroll Block und Noten Anzeige

Achtung, böser Fehler:

Wenn die Ausgänge des Notengenerators falsch vom Addierer anstatt richtig dem Zustandsspeicher (Q der D-FlipFlops) abgenommen werden, kann nicht mehr compiliert werden (can not fit!).

Temperierte Halbton Stimmungstabelle

```
% Stimmungstabelle temperiert Halbton        %
% 5 bit Halbton > 8 bit Tonefrequenz Divider %
% 5.7.02, 9.7, 6.1.03 Zi                     %
SUBDESIGN th_tab
(     T0, T1, T2, T3, T4  : INPUT;
      D0, D1, D2, D3, D4, D5, D6, D7 :OUTPUT; )
BEGIN
 % Stimmungstabelle 5 bit Halbton > 8 bit Frequenzteiler %
 % c1 = 261Hz, divider clock input 130kHz                %
TABLE
%  D  H % T4,T3,T2,T1,T0  => D7, D6, D5, D4, D3, D2, D1, D0 ; % H   D   Ton  %
%  0 00 %  0, 0, 0, 0, 0  =>  1,  1,  1,  1,  1,  0,  0,  0 ; % F8 248  c1   %
%  1 01 %  0, 0, 0, 0, 1  =>  1,  1,  1,  0,  1,  0,  1,  0 ; % EA 234  cis1 %
%  2 02 %  0, 0, 0, 1, 0  =>  1,  1,  0,  1,  1,  1,  0,  1 ; % DD 221  d1   %
%  3 03 %  0, 0, 0, 1, 1  =>  1,  1,  0,  1,  0,  0,  0,  1 ; % D1 209  dis1 %
%  4 04 %  0, 0, 1, 0, 0  =>  1,  1,  0,  0,  0,  1,  0,  1 ; % C5 197  e1   %
%  5 05 %  0, 0, 1, 0, 1  =>  1,  0,  1,  1,  1,  0,  1,  0 ; % BA 186  f1   %
%  6 06 %  0, 0, 1, 1, 0  =>  1,  0,  1,  0,  1,  1,  1,  1 ; % AF 175  fis1 %
%  7 07 %  0, 0, 1, 1, 1  =>  1,  0,  1,  0,  0,  1,  1,  0 ; % A6 166  g1   %
%  8 08 %  0, 1, 0, 0, 0  =>  1,  0,  0,  1,  1,  1,  0,  0 ; % 9C 156  gis1 %
%  9 09 %  0, 1, 0, 0, 1  =>  1,  0,  0,  1,  0,  0,  1,  1 ; % 93 147  a1   %
% 10 0A %  0, 1, 0, 1, 0  =>  1,  0,  0,  0,  1,  0,  1,  1 ; % 8B 139  ais1 %
% 11 0B %  0, 1, 0, 1, 1  =>  1,  0,  0,  0,  0,  0,  1,  1 ; % 83 131  h1   %
% 12 0C %  0, 1, 1, 0, 0  =>  0,  1,  1,  1,  1,  1,  0,  0 ; % 7C 124  c2   %
% 13 0D %  0, 1, 1, 0, 1  =>  0,  1,  1,  1,  0,  1,  0,  1 ; % 75 117  cis2 %
% 14 0E %  0, 1, 1, 1, 0  =>  0,  1,  1,  0,  1,  1,  1,  0 ; % 6E 110  d2   %
% 15 0F %  0, 1, 1, 1, 1  =>  0,  1,  1,  0,  1,  0,  0,  0 ; % 68 104  dis2 %
% 16 10 %  1, 0, 0, 0, 0  =>  0,  1,  1,  0,  0,  0,  1,  1 ; % 62  98  e2   %
% 17 11 %  1, 0, 0, 0, 1  =>  0,  1,  0,  1,  1,  1,  0,  1 ; % 5D  93  f2   %
% 18 12 %  1, 0, 0, 1, 0  =>  0,  1,  0,  1,  1,  0,  0,  0 ; % 58  88  fis2 %
% 19 13 %  1, 0, 0, 1, 1  =>  0,  1,  0,  1,  0,  0,  1,  1 ; % 53  83  g2   %
% 20 14 %  1, 0, 1, 0, 0  =>  0,  1,  0,  0,  1,  1,  1,  0 ; % 4E  78  gis2 %
% 21 15 %  1, 0, 1, 0, 1  =>  0,  1,  0,  0,  1,  0,  1,  0 ; % 4A  74  a2   %
% 22 16 %  1, 0, 1, 1, 0  =>  0,  1,  0,  0,  0,  1,  1,  0 ; % 46  70  ais2 %
% 23 17 %  1, 0, 1, 1, 1  =>  0,  1,  0,  0,  0,  0,  1,  1 ; % 42  66  h2   %
% 24 18 %  1, 1, 0, 0, 0  =>  0,  0,  1,  1,  1,  1,  1,  0 ; % 3E  62  c3   %
% 25 19 %  1, 1, 0, 0, 1  =>  0,  0,  1,  1,  1,  0,  1,  1 ; % 3B  59  cis3 %
% 26 1A %  1, 1, 0, 1, 0  =>  0,  0,  1,  1,  0,  1,  1,  1 ; % 37  55  d3   %
% 27 1B %  1, 1, 0, 1, 1  =>  0,  0,  1,  1,  0,  1,  0,  0 ; % 34  52  dis3 %
% 28 1C %  1, 1, 1, 0, 0  =>  0,  0,  1,  1,  0,  0,  0,  1 ; % 31  49  e3   %
% 29 1D %  1, 1, 1, 0, 1  =>  0,  0,  1,  0,  1,  1,  1,  0 ; % 2E  46  f3   %
% 30 1E %  1, 1, 1, 1, 0  =>  0,  0,  1,  0,  1,  1,  0,  0 ; % 2C  44  fis3 %
% 31 1F %  1, 1, 1, 1, 1  =>  0,  0,  1,  0,  1,  0,  0,  1 ; % 29  41  g3   %
END TABLE;
END;
```

Temperierte Halbton Stimmungstabelle, AHDL Text-Design-File th_tab.tdf