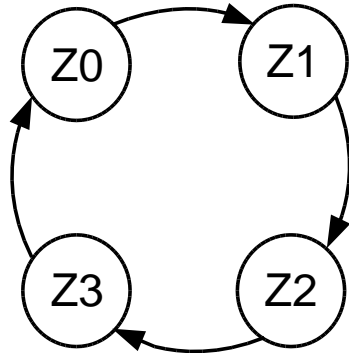


Zähler von 0 bis 3

Zustandsübergangsdiagramm



Zustandsübergangstabelle

aktueller Zustand	folgender Zustand
Z0	Z1
Z1	Z2
Z2	Z3
Z3	Z0

Zustandskodierung Dual

Zustand	Q1	Q0
Z0	0	0
Z1	0	1
Z2	1	0
Z3	1	1

Beschreibung des Zählers in Abel-Syntax

```

MODULE Counter
  Takt    PIN 11;      // Takt für Zustandsspeicher
  Q1,Q0   PIN 15,14  ISTYPE 'REG'; //
  Zustandsspeicher

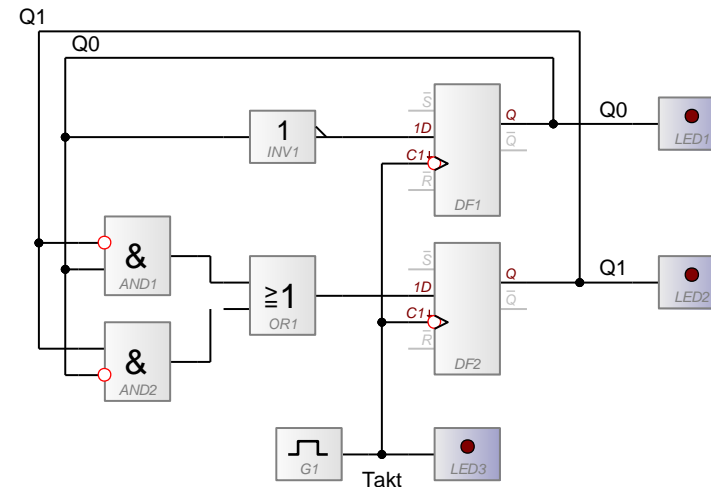
  // Takt für die D-FFs
  EQUATIONS
  Q1.clk = Takt;
  Q0.clk = Takt;

  TRUTH_TABLE
  ([Q1,Q0]:>[Q1,Q0]);
  [ 0, 0]:>[ 0, 1];
  [ 0, 1]:>[ 1, 0];
  [ 1, 0]:>[ 1, 1];
  [ 1, 1]:>[ 0, 0];

  // alternative Beschreibung des Zählers mit
  // EQUATIONS
  // Q1 := (Q1 & !Q0) # (!Q1 & Q0);
  // Q0 := !Q0;

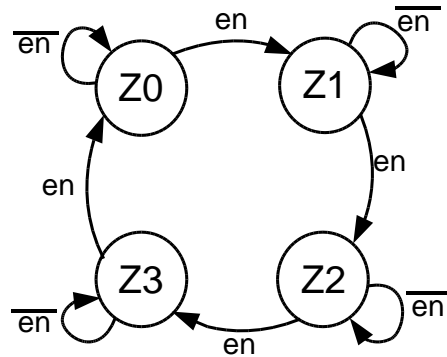
  END
  
```

Simulation mit ProfiLab



Zähler von 0 bis 3 mit Freigabesignal

Zustandsübergangsdiagramm



Zustandsübergangstabelle

Eingangssignal en	aktueller Zustand	folgender Zustand
0	Z0	Z0
1	Z0	Z1
0	Z1	Z1
1	Z1	Z2
0	Z2	Z2
1	Z2	Z3
0	Z3	Z3
1	Z3	Z0

Zustandskodierung Dual

Zustand	Q1	Q0
Z0	0	0
Z1	0	1
Z2	1	0
Z3	1	1

Beschreibung des Zählers in Abel-Syntax

```

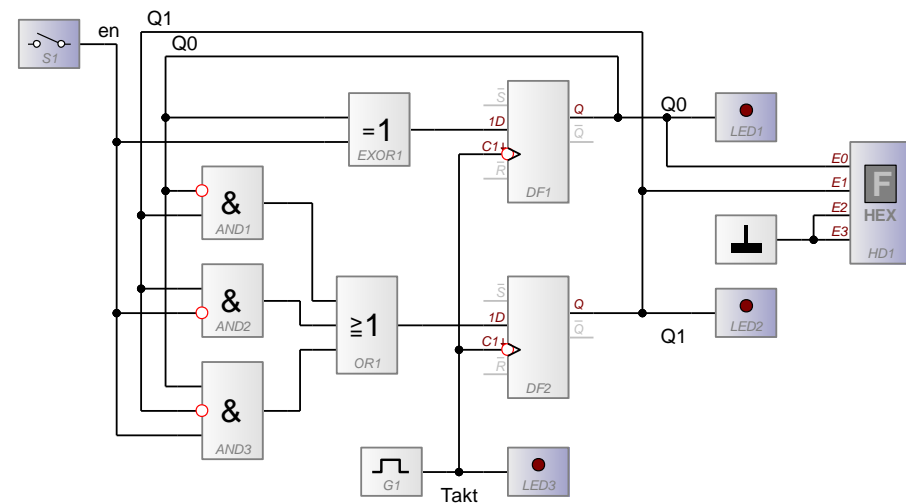
MODULE CounterEn
Takt  PIN 11;    // Takt für Zustandsspeicher
en    PIN 2;    // en = 1 Freigabe für Zähler
Q1,Q0 PIN 15,14 ISTYPE 'REG'; // Zustandsspeicher

// Takt für die D-FFs
EQUATIONS
Q1.clk = Takt;
Q0.clk = Takt;

TRUTH_TABLE
([en,Q1,Q0]:>[Q1,Q0]);
// Zähler gesperrt
[ 0, 0, 0]:>[ 0, 0];
[ 0, 0, 1]:>[ 0, 1];
[ 0, 1, 0]:>[ 1, 0];
[ 0, 1, 1]:>[ 1, 1];
// Zähler freigegeben
[ 1, 0, 0]:>[ 0, 1];
[ 1, 0, 1]:>[ 1, 0];
[ 1, 1, 0]:>[ 1, 1];
[ 1, 1, 1]:>[ 0, 0];

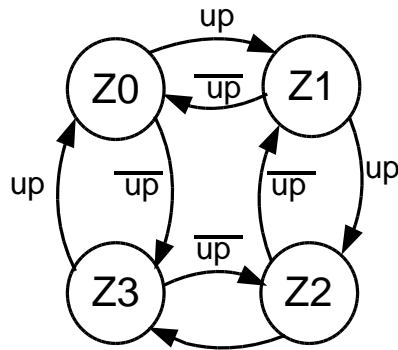
// alternative Beschreibung des Zählers mit
// EQUATIONS
// Q1 := !Q0 & Q1 # Q1 & !en # Q0 & !Q1 & en;
// Q0 := !Q0 & en # Q0 & !en;
END
    
```

Simulation mit ProfiLab



Zähler von 0 bis 3 mit wählbarer Richtung

Zustandsübergangsdiagramm



Zustandsübergangstabelle

Eingangssignal up	aktueller Zustand	folgender Zustand
0	Z0	Z3
1		Z1
0	Z1	Z0
1		Z2
0	Z2	Z1
1		Z3
0	Z3	Z2
1		Z0

Zustandskodierung Dual

Zustand	Q1	Q0
Z0	0	0
Z1	0	1
Z2	1	0
Z3	1	1

Beschreibung des Zählers in Abel-Syntax

```

MODULE CounterUp
Takt PIN 11; // Takt für Zustandsspeicher
up PIN 2; // up = 0 -> rückwärts
// up = 1 -> vorwärts
Q1,Q0 PIN 15,14 ISTYPE 'REG'; // Zustandsspeicher

// Takt für die D-FFs
EQUATIONS
[Q1..Q0].clk = Takt;

TRUTH_TABLE
([up,Q1,Q0]:>[Q1,Q0]);
// rückwärts
[ 0, 0, 0]:>[ 1, 1];
[ 0, 0, 1]:>[ 0, 0];
[ 0, 1, 0]:>[ 0, 1];
[ 0, 1, 1]:>[ 1, 0];
// vorwärts
[ 1, 0, 0]:>[ 0, 1];
[ 1, 0, 1]:>[ 1, 0];
[ 1, 1, 0]:>[ 1, 1];
[ 1, 1, 1]:>[ 0, 0];

// alternative Beschreibung des Zählers mit
// EQUATIONS
// Q1 := (!Q0 & Q1 & up) # (Q0 & !Q1 & up) #
// (Q0 & Q1 & !up) # (!Q0 & !Q1 & !up);
// Q0 := !Q0;
END
  
```

Simulation mit ProfiLab

