```
// VerilogA for ece1396h, bin2ther, veriloga

`include "constants.vams"
`include "disciplines.vams"
`include "discipline.h"
`include "constants.h"

// $Date: 1997/08/28 05:54:32 $
// $Revision: 1.1 $
//
//
// Based on the OVI Verilog-A Language Reference Manual, version 1.0 1996
//
//



//--------------------
// bin2ther_ideal
//
// -  Ideal binary to thermometer code decoder
//
// vin:        [V,A]
// vclk:       [V,A]
// vd0..vd6:   data output terminals  [V,A]
//
// INSTANCE parameters
//    tdel, trise, tfall = {usual} [s]
//    vlogic_high = [V]
//    vlogic_low  = [V]
//    vtrans_clk  = clk high to low transition voltage [V]
//    vref        = voltage that voltage is done with respect to [V]
//
// MODEL parameters
//    {none}
//
// This model is ideal in the sense that there is no mismatch modeled.
//

module bin2ther(vd6, vd5, vd4, vd3, vd2, vd1, vd0, vin2, vin1, vin0);
electrical vd6, vd5, vd4, vd3, vd2, vd1, vd0, vin2, vin1, vin0;
parameter real trise = 0 from [0:inf);
parameter real tfall = 0 from [0:inf);
parameter real tdel = 0 from [0:inf);
parameter real vlogic_high = 5;
parameter real vlogic_low  = 0;
//parameter real vtrans_clk     = 2.5;
//parameter real vref        = 1.0;

real vd[0:6];

   analog begin

                vd[6] = 0;
                if (V(vin2) == vlogic_low && V(vin1) == vlogic_low && V(vin0) == vlogic_low) begin
                     vd[6] = vlogic_low;
                     vd[5] = vlogic_low;
                     vd[4] = vlogic_low;
                     vd[3] = vlogic_low;
                     vd[2] = vlogic_low;
                     vd[1] = vlogic_low;
                     vd[0] = vlogic_low;
                end
                else if (V(vin2) == vlogic_low && V(vin1) == vlogic_low && V(vin0) == vlogic_high) begin
                     vd[6] = vlogic_low;
                     vd[5] = vlogic_low;
                     vd[4] = vlogic_low;
                     vd[3] = vlogic_low;
                     vd[2] = vlogic_low;
                     vd[1] = vlogic_low;
                     vd[0] = vlogic_high;
                end
                else if (V(vin2) == vlogic_low && V(vin1) == vlogic_high && V(vin0) == vlogic_low) begin
                     vd[6] = vlogic_low;
                     vd[5] = vlogic_low;
                     vd[4] = vlogic_low;
                     vd[3] = vlogic_low;
                     vd[2] = vlogic_low;
                     vd[1] = vlogic_high;
```

```
                  vd[0] = vlogic_high;
            end
            else if (V(vin2) == vlogic_low && V(vin1) == vlogic_high && V(vin0) == vlogic_high) begin
                  vd[6] = vlogic_low;
                  vd[5] = vlogic_low;
                  vd[4] = vlogic_low;
                  vd[3] = vlogic_low;
                  vd[2] = vlogic_high;
                  vd[1] = vlogic_high;
                  vd[0] = vlogic_high;
            end
            else if (V(vin2) == vlogic_high && V(vin1) == vlogic_low && V(vin0) == vlogic_low) begin
                  vd[6] = vlogic_low;
                  vd[5] = vlogic_low;
                  vd[4] = vlogic_low;
                  vd[3] = vlogic_high;
                  vd[2] = vlogic_high;
                  vd[1] = vlogic_high;
                  vd[0] = vlogic_high;
            end
            else if (V(vin2) == vlogic_high && V(vin1) == vlogic_low && V(vin0) == vlogic_high) begin
                  vd[6] = vlogic_low;
                  vd[5] = vlogic_low;
                  vd[4] = vlogic_high;
                  vd[3] = vlogic_high;
                  vd[2] = vlogic_high;
                  vd[1] = vlogic_high;
                  vd[0] = vlogic_high;
            end
            else if (V(vin2) == vlogic_high && V(vin1) == vlogic_high && V(vin0) == vlogic_low) begin
                  vd[6] = vlogic_low;
                  vd[5] = vlogic_high;
                  vd[4] = vlogic_high;
                  vd[3] = vlogic_high;
                  vd[2] = vlogic_high;
                  vd[1] = vlogic_high;
                  vd[0] = vlogic_high;
            end
            else if (V(vin2) == vlogic_high && V(vin1) == vlogic_high && V(vin0) == vlogic_high) begin
                  vd[6] = vlogic_high;
                  vd[5] = vlogic_high;
                  vd[4] = vlogic_high;
                  vd[3] = vlogic_high;
                  vd[2] = vlogic_high;
                  vd[1] = vlogic_high;
                  vd[0] = vlogic_high;
            end
            else begin
                  vd[6] = vd[6];
                  vd[5] = vd[5];
                  vd[4] = vd[4];
                  vd[3] = vd[3];
                  vd[2] = vd[2];
                  vd[1] = vd[1];
                  vd[0] = vd[0];
            end

      //
      // assign the outputs
      //

      V(vd6) <+ transition( vd[6], tdel, trise, tfall );
      V(vd5) <+ transition( vd[5], tdel, trise, tfall );
      V(vd4) <+ transition( vd[4], tdel, trise, tfall );
      V(vd3) <+ transition( vd[3], tdel, trise, tfall );
      V(vd2) <+ transition( vd[2], tdel, trise, tfall );
      V(vd1) <+ transition( vd[1], tdel, trise, tfall );
      V(vd0) <+ transition( vd[0], tdel, trise, tfall );

   end
endmodule
```