```
// ********************************************************************************************************
// **                                                                                                    **
// **   MLAW2LIN.v - MU-LAW TO LINEAR 2'S COMPLEMENT CODE TRANSLATOR                                     **
// **                                                                                                    **
// ********************************************************************************************************
// **                                                                                                    **
// **                              COPYRIGHT (c) 2001 YOUNG ENGINEERING                                  **
// **                                      ALL RIGHTS RESERVED                                           **
// **                                                                                                    **
// **   THIS PROGRAM IS CONFIDENTIAL AND  A  TRADE SECRET  OF  YOUNG  ENGINEERING.  THE RECEIPT OR       **
// **   POSSESSION  OF THIS PROGRAM  DOES NOT  CONVEY  ANY  RIGHTS TO  REPRODUCE  OR  DISCLOSE ITS      **
// **   CONTENTS,  OR TO MANUFACTURE, USE, OR SELL  ANYTHING  THAT IT MAY DESCRIBE, IN WHOLE OR IN      **
// **   PART, WITHOUT THE SPECIFIC WRITTEN CONSENT OF YOUNG ENGINEERING.                                 **
// **                                                                                                    **
// ********************************************************************************************************
// **   Revision       : 1.0                                                                             **
// **   Modified Date  : 11/01/2001                                                                      **
// **   Revision History:                                                                                **
// **                                                                                                    **
// **   11/01/2001:  Initial design                                                                      **
// **                                                                                                    **
// ********************************************************************************************************
// **                                      TABLE OF CONTENTS                                             **
// ********************************************************************************************************
// **--------------------------------------------------------------------------------------------------**
// **   DECLARATIONS                                                                                     **
// **--------------------------------------------------------------------------------------------------**
// **--------------------------------------------------------------------------------------------------**
// **   FORMAT CONVERSION                                                                                **
// **--------------------------------------------------------------------------------------------------**
// **   1.01:  Mu-Law Input Inversion                                                                    **
// **   1.02:  Mu-Law to Linear Table                                                                    **
// **   1.03:  Linear Output Unbias                                                                      **
// **   1.04:  Linear 2's Complement                                                                     **
// **                                                                                                    **
// ********************************************************************************************************


`timescale 1ns/10ps

module MLAW2LIN       ( DataI, DataO );

   input  [07:00]       DataI;                          // data input - Mu-law

   output [12:00]       DataO;                          // data output - linear 2's complement


// ********************************************************************************************************
// **   DECLARATIONS                                                                                     **
// ********************************************************************************************************

   wire [07:00]         DataI_N;                        // bit-wise inverted input
   wire [12:00]         DataO;                          // linear data output - 2's complement

   reg  [13:00]         BiasedData;                     // linear data - biased
   wire [12:00]         LinearData;                     // linear data - unsigned


// ********************************************************************************************************
// **   FORMAT CONVERSION                                                                                **
// ********************************************************************************************************
// --------------------------------------------------------------------------------------------------------
//      1.01:  Mu-Law Input Inversion
// --------------------------------------------------------------------------------------------------------

   assign DataI_N = ~DataI;

// --------------------------------------------------------------------------------------------------------
//      1.02:  Mu-Law to Linear Table
// --------------------------------------------------------------------------------------------------------

   always @(DataI_N) begin
      casex(DataI_N)
         8'b0111xxxx :  BiasedData = {2'b11,DataI_N[03:00],8'b10000000}; // + full scale
         8'b0110xxxx :  BiasedData = {3'b101,DataI_N[03:00],7'b1000000};
         8'b0101xxxx :  BiasedData = {4'b1001,DataI_N[03:00],6'b100000};
         8'b0100xxxx :  BiasedData = {5'b10001,DataI_N[03:00],5'b10000};
         8'b0011xxxx :  BiasedData = {6'b100001,DataI_N[03:00],4'b1000};
         8'b0010xxxx :  BiasedData = {7'b1000001,DataI_N[03:00],3'b100};
```

```
         8'b0001xxxx :  BiasedData = {8'b10000001,DataI_N[03:00],2'b10};
         8'b0000xxxx :  BiasedData = {9'b100000001,DataI_N[03:00],1'b1}; // + zero

         8'b1000xxxx :  BiasedData = {9'b000000001,DataI_N[03:00],1'b1}; // - zero
         8'b1001xxxx :  BiasedData = {8'b00000001,DataI_N[03:00],2'b10};
         8'b1010xxxx :  BiasedData = {7'b0000001,DataI_N[03:00],3'b100};
         8'b1011xxxx :  BiasedData = {6'b000001,DataI_N[03:00],4'b1000};
         8'b1100xxxx :  BiasedData = {5'b00001,DataI_N[03:00],5'b10000};
         8'b1101xxxx :  BiasedData = {4'b0001,DataI_N[03:00],6'b100000};
         8'b1110xxxx :  BiasedData = {3'b001,DataI_N[03:00],7'b1000000};
         8'b1111xxxx :  BiasedData = {2'b01,DataI_N[03:00],8'b10000000}; // - full scale
      endcase
   end

// ---------------------------------------------------------------------------------------------------
//      1.03:  Linear Output Unbias
// ---------------------------------------------------------------------------------------------------

   assign LinearData = (BiasedData - 33) >> 1;

// ---------------------------------------------------------------------------------------------------
//      1.04:  Linear 2's Complement
// ---------------------------------------------------------------------------------------------------

   assign DataO = LinearData[12] ? {1'b0,LinearData[11:00]} : {1'b1,(~LinearData[11:00] + 1)};

endmodule
```