Chapter 5

LINEARIZATION OF TRANSDUCER CHARACTERISTICS

1.3 Introduction

Process signals in nature are normally in analog form and these are quite often non-electrical in nature. The measurement and control by electrical means thus requires transducers that convert the non-electrical signals into equivalent electrical ones. The electrical signals thus generated are further processed and scaled to present either to the galvanometric indicating instruments or to the analog to digital converters (ADC). The ADC is the gateway to digital indicator and today’s state of the art microcomputer based measurement and control systems. Digital technologies have been replacing the analog methods of measurement and instrumentation over the past few decades, because of its inherent noise immunity and ease of data processing. Figure 5.1 illustrates a typical digital measurement and control system.

![Fig. 5.1: Typical Microcomputer Based Measurement and Control System](image)

As in Fig. 5.1, the transducer is used to sense the process variable and to produce equivalent electrical signal. The electrical signal is transmitted through channels to the signal conditioner for processing the signal suitable for conversion into digital data. Digital data is then fed to the controller for further processing to display the process...
value and to generate control output. Control output controls the actual process through *actuator*.

A well defined relationship between the ADC output and the process value is required to be established for meaningful determination of the process value. In many applications, linear behavior is most desirable. When the relation is linear we have only to multiply the sensor output by a fixed constant to get the value of the measured quantity. But if the behavior is nonlinear we have to consult a nonlinear curve or to compute from a nonlinear equation, to determine the actual process value. Therefore, a linear relationship simplifies the hardware. However, a linear relationship is always not available. Non-linearity in instrumentation can result from the measurement principle, from the sensor or from the sensor conditioning [5.1]. In head type flow meters, for example, the measured drop in pressure is proportional to the square of the flow velocity. Hence flow velocity can be obtained by taking square root of the differential pressure signal. Many sensors are linear only in limited measured ranges and others are essentially non-linear. Non-linearity attributed by sensor conditioning is also common; for example, voltage dividers or bridge circuits connecting resistive (linear) sensors produce non-linear outputs (please refer to section 2.2.1). Sometimes, ADCs also introduce non-linearity, for example, the dual slope ADCs are subjected to integrating nonlinearity (please refer to section 4.2.3 for details). It is, therefore, necessary to linearize the transducer output in order to produce a meaningful interpretation of the process value.

Both analog and digital methods are used to linearize sensor outputs. However, digital methods are simple and easy to implement and accurate than their analog counter parts. The author worked on the implementation of some digital methods [5.2], [5.3] which are described in Sections of 5.3.1 and later in this chapter. Before going to the details of these methods, some well known analog and digital linearization techniques are reviewed in Section-5.2.

### 5.2 Analog Techniques

Initially the analog techniques for linearizing sensor characteristics were developed as analog signals were used in instrumentation. Later digital instrumentation techniques
started developing with the popularity of ADCs and microprocessors. This section describes some of the useful analog methods for linearizing sensor characteristics.

5.2.1 Using Diodes

The non-linear signal can be made linear by using special operational amplifier configurations having equal and opposite non-linear characteristics. For example, optoelectronic transducers typically have exponential relationship [5.4] between the output signal and the input light intensity governed by:

\[ V_0 = K \exp(-\alpha Q) \]  

(5.1)

where, \( Q \) is the light intensity, \( V_0 \) is the output signal and \( K \) and \( \alpha \) are constants. The plot of Equation (5.1) depicts the input-output characteristics of a typical light sensor and is presented in Fig. 5.1.

![Input-Output Characteristics of a Typical Light Sensor](image)

Diode forward characteristic, as shown in Fig. 5.2, is also exponential in nature and is given by:

\[ I = I_0 \exp\left(\frac{eV}{kT}\right) \]  

(5.2)

where, \( I \) is diode current, \( V \) is diode bias voltage and other symbols have their usual meanings.
As seen in Fig. 5.1 and Fig. 5.2, the two curves have opposite trends and therefore they can be combined to produce a linear curve. The curves are to be combined such that the voltage output of the light sensor will control the diode forward current and the diode voltage thus produced will be linear with respect to the light intensity. The implementation of this logic can be achieved by placing a diode in the feedback path of the amplifier as shown in Fig. 5.3.

The amplifier output can be obtained by combining Equation (5.1) and Equation (5.2) as follows:

The diode forward current is:

\[
\frac{V_i}{R_i} = \frac{1}{R_i} K \exp(-\alpha Q) = I_0 \exp\left(\frac{eV_0}{kT}\right)
\]  

(5.3)
This may be further reduced to,
\[ V_o = C - BQ \] \hspace{1cm} (5.4)
where, B and C are constants.

Equation (5.4) now shows that the output varies linearly with the light intensity Q.

### 5.2.2 Segment-wise Linearization

Another important linearization method is Segment-wise (piece-wise) scheme in which gain of the amplifier circuit, processing non-linear input signal, is dynamically adjusted between two limits according to the nature of the input signal. This scheme is a general approach unlike the previous one. Such a linearizing circuit is shown in Fig. 5.4.

The circuit of Fig. 5.4 is a non-inverting amplifier. The general gain expression of such an amplifier is:
\[ G = (1 + \frac{R_f}{R_i}) \] \hspace{1cm} (5.5)
where, \( R_i \) is the effective input resistance connected between inverting terminal (A) and ground. As seen in Fig. 5.4, the input voltage \( v_i \) which is also reflected at the inverting input of the op-amp will determine which of the diodes will be turned on since their
respective cathodes are biased to different voltages by the voltage divider network comprising of $V_r$, $R_5$, $R_6$ and $R_7$. Thus the effective resistance $R_i$ connected between A and ground is $R_1$ in parallel with the combination of all the paths where the diodes are turned on. This sets different gains between different voltage limits. For example, when the input voltage is less than $V_1$, all the resistors except $R_1$ are disconnected from the ground as all the diodes remain off. Thus the gain of the circuit is $R_f/R_1$ for $v_i < V_1$. Now if the input voltage exceeds $V_1$, the diode connected in series with $R_2$ will be on and the gain of the circuit will be increased as the path containing $R_2$ comes in parallel with $R_1$. Similarly, the gain of the circuit will be increased by a step further while the input voltage crosses the next higher voltage limit.

The transfer characteristic of the circuit of Fig. 5.4 is shown in Fig. 5.5. Fig. 5.6 shows how this transfer characteristic can be used to replace a non-linear curve. The same linearizing scheme can be used to fit other non-linear curves by adjusting the breakpoints ($V_1$, $V_2$, $V_3$ etc.) and the segment slopes (through $R_1$, $R_2$, $R_3$, $R_4$ etc.). More breakpoints can also be introduced for smooth curve fitting and hence to achieve more accuracy.

![Graph showing the transfer characteristic](image)

*Fig. 5.5: Transfer Characteristic of the linearizing circuit of Fig. 5.4*
Fig. 5.6: Transfer Characteristic of the linearizing circuit of Fig. 5.4 fitted for replacing a non-linear curve.

5.3 **Digital Linearization Techniques**

The analog methods discussed are complicated in nature and involve precision components for implementation. The precision in these systems primarily depends upon the number of segments that are implemented. More is the number of segments more will be the accuracy. Design of such a circuit requires a number of precession passive and active components along with accurate trimming of break points. This makes the circuit implementation a difficult task. Apart from this they also have the following drawbacks:

- Design complexity increases with the number of segments
- Adjustment of Break-point is tedious and time consuming
- Dependence of the components on temperature and other operating parameters shifts the calibration.

On the other hand, digital methods are accurate and easy to implement. Such implementations in microprocessor based systems are in common use. But for a small, dedicated system, use of a microprocessor along with the necessary peripherals is not always cost effective. Here we present the following linearization techniques which can be implemented without a microprocessor.

- Point to Point Linearization: A Low Cost Digital Technique discussed in details in section 5.3.1 and
5.3.1 Point to Point Linearization: A Low Cost Digital Technique

5.3.1.1 Introduction

It is a simple low cost digital technique offering high accuracy. The basic idea is to digitize the non-linear sensor output by an Analog to Digital Converter (ADC), which is then used to address a ROM storing the actual linear process data in BCD format. The BCD output is then converted into 7-segment display format for driving a 4-digit LED display. Block diagram of the system is presented in Fig. 5.7.

![Block Diagram of the ROM Linearizer](image)

**Fig. 5.7: Block Diagram of the ROM Linearizer**

ROM data storage and process value display are explained in Fig. 5.8. 4-digit data requires two ROM locations for storing the same. Higher bytes and lower bytes of data are stored in separate blocks of the ROM (Fig. 5.8). For example, a process value “1234” corresponding to “000” ADC output are to be stored in ROM such that the lower 2-digit “34” will be in “0000H” location and the upper two digit “12” will be in “1000H”. Therefore, each process value requires two steps for display as all the locations of the ROM share the same data bus (D_7-D_0) to offer their data. These steps are determined by the clock signal. The lower byte (lower two digits) is read from ROM and displayed when clock is low and higher byte (upper two digits) is read and displayed when clock is high. Therefore, the value to be displayed depends upon the ADC output whereas the way it is accessed and displayed depends upon the clock signal.
5.3.1.2 ROM Data Formation

The most important part of the implementation of the linearization technique is the formation of ROM data. This is explained with the help of a sample data presented in Table-5.1. First two columns of the Table represent the typical characteristic of a hypothetical sensor. As shown in the table, for a process value in the range of 0 to 15, the corresponding sensor output is 0 to 12 mV. The unfilled cells in process value and mV columns represent intermediate values and are not required to be considered for a resolution of ‘1’ of the process value. Analog to Digital Conversion gain is taken as 2. The corresponding ADC count will be 0 to 24, which can access 24 bytes of ROM. Therefore, process values of 0 to 15 have to be stored in 24 locations which results some overlapping locations storing the same process value as shown in Table-5.1.
Table 5.1: ROM data formation Scheme

<table>
<thead>
<tr>
<th>Process Value</th>
<th>Sensor O/P (mV)</th>
<th>ADC COUNT</th>
<th>ROM ADDR.</th>
<th>ROM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HI BYTE</td>
</tr>
<tr>
<td>0000</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0001</td>
<td>1.0</td>
<td>2</td>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0002</td>
<td>2.0</td>
<td>4</td>
<td>4</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0003</td>
<td>3.0</td>
<td>6</td>
<td>6</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0004</td>
<td>3.9</td>
<td>8</td>
<td>8</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0005</td>
<td>4.8</td>
<td>10</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>0006</td>
<td>5.7</td>
<td>11</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>12</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0007</td>
<td>6.5</td>
<td>13</td>
<td>13</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>14</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0008</td>
<td>7.3</td>
<td>15</td>
<td>15</td>
<td>00</td>
</tr>
<tr>
<td>0009</td>
<td>8.1</td>
<td>16</td>
<td>16</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>17</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0010</td>
<td>8.8</td>
<td>18</td>
<td>18</td>
<td>00</td>
</tr>
<tr>
<td>0011</td>
<td>9.5</td>
<td>19</td>
<td>19</td>
<td>00</td>
</tr>
<tr>
<td>0012</td>
<td>10.2</td>
<td>20</td>
<td>20</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>21</td>
<td></td>
<td>00</td>
</tr>
<tr>
<td>0013</td>
<td>10.8</td>
<td>22</td>
<td>22</td>
<td>00</td>
</tr>
<tr>
<td>0014</td>
<td>11.4</td>
<td>23</td>
<td>23</td>
<td>00</td>
</tr>
<tr>
<td>0015</td>
<td>12.0</td>
<td>24</td>
<td>24</td>
<td>00</td>
</tr>
</tbody>
</table>
For an actual sensor operated in its entire range of process value (PV), manual formation of ROM data as shown in Table 5.1 is not feasible. For example, let us consider a type “K” thermocouple used in the temperature range of $0 – 1200^\circ C$. The thermocouple characteristic is non-linear in nature, the different regions of the curve having different slopes. For $1^\circ C$ resolution (including accuracy), the conversion gain (from analog to digital) should be selected such that all the temperature values 0, 1, 2, 3 …… 1200 will find at least one room in the ROM. For higher gain each temperature may find more than one locations in the ROM ensuring smooth transition from one temperature to the other. Also as the temperature is a 4-digit value, two byte locations are needed for storing one value. The two bytes are stored in the ROM in two blocks such that the lower address range (A0-A11) is the same for the two bytes of the same data. The most significant address bit is provided by a clock signal which is also used to enable the corresponding BCD-to-7-segment decoders as shown in Fig.-5.9. Therefore, more than 2400 (1200 x 2) bytes of data are to be formed for programming the ROM for a type “K” thermocouple in the range of $0 – 1200^\circ C$ with a system resolution of $1^\circ C$. This is practically impossible manually. Thus a “C” program is developed which reads sensor information from a data file and generates an output file containing ROM data. It also takes lower and upper range of the process value as inputs. The program is so developed that it always utilizes the entire range of the ADC ($0 – 4096$ counts for ICL7109, a 12-bit ADC). The size of the ROM data produced by the program is always 8KB (4 KB for upper block and 4 KB for lower block). Thus the gain of the circuit that will be required to amplify the sensor output to generate input signal for ADC is dependent on the type of the sensor and the range of the process value. The program calculates the value of the gain to be used and reports the same (product of amplifier gain and ADC conversion factor). It also generates ROM data for the entire 8K. Each data byte is stored in a new line in a disk file. The program also adds “DB” (define byte) to each line preceded by data byte and saves the disk file as an assembly file having extension “asm”. The assembly file is converted into an Intel Hex file with the help of a standard assembler and a linker. Intel Hex file is then directly down loaded into the ROM through a standard PC driven EPROM Programmer. The listing of the “C” program is given in Appendix 4. The program flow is presented in Fig. 5.9(a) and the screen shot of the result after the program is run is given in Fig. 5.9(b).
Fig. 5.9: Program Flow for generating ROM data.

Fig. 5.9(b): Screen shot of the result of the "C" Program for generating ROM Data
5.3.1.3 Circuit Consideration
The heart of the instrument is the A/D converter. Thus the overall accuracy of the system depends upon the accuracy and stability of the ADC circuit. Different blocks of the circuit are implemented such that 0.1 % or better accuracy may be ensured.

5.3.1.4 Amplifier and ADC
The experimental circuit is designed for type “K” thermocouple. The voltage produced by the thermocouple (0-48.828 mV for 0-1200°C) is not suitable for Analog to Digital Conversion. Thus an amplifier is required to boost up the input voltage suitable for the conversion. Again the cold junction of a thermocouple is kept at ambient temperature in a practical circuit. Therefore mV produced by the thermocouple is less by an amount that would be produced by the same thermocouple while keeping its hot junction in ambient temperature and cold junction in zero degrees Celsius. This is to be compensated by adding the thermocouple absolute mV for ambient temperature to the generated thermo emf. This addition of mV is called Cold Junction Compensation (CJC). For dynamic CJC, however, a second temperature sensor like RTD, thermistor, junction diode or IC temperature sensor is used for sensing ambient temperature and generating the exact shortfall of mV due to the elevation of cold junction temperature instead of keeping it at zero Degrees Celsius. The thermocouple amplifier with automatic cold junction temperature compensating circuit using commonly available LM324 quad operational amplifier used in the design is shown in Fig. 5.10.
Operational Amplifier OA#1 is used to amplify the thermocouple mV; OA#2 is used to interface the ambient sensor for automatic cold junction compensation. The constant current source (I) is used to excite the ambient RTD sensor and is derived from an LM336-2.5V Zenner voltage reference as shown in Fig. 5.11. The operation of the circuit is self explanatory. The value of the current is obtained by dividing the voltage drop appearing across $R_E$ with the value of $R_E$ and is given in Equation (5.6):

$$I = \frac{V_{cc} - V_{E}}{R_E} = \frac{V_{cc} - (V_{cc} - V_Z)}{R_E} = \frac{V_Z}{R_E}$$

This current flows through the load resistance and is independent of both the load and the supply voltage.
The final stage of the amplifier is an adder which adds the uncompensated thermocouple output with the compensating mV generated by the CJC circuit. It also compensates for the offset voltage produced by the resistance of an RTD at zero degrees Celsius.

The choice of ADC depends upon a number of factors like display resolution, process response and of course the cost of the component. As our experimental process value ranges from 0 to 1200°C with 1°C display resolution, the resolution of the ADC should be 11 bit or better. Temperature itself is a slow changing process variable and 4-5 conversions per second is sufficient for most of the cases. Dual slope A/D converters are slow but cheaper and thus are suitable for such applications. Intersil 7109 is a commonly available low cost 12-bit dual slope A/D converter and is chosen for the purpose. Moreover in dual slope ADC, proper component selection can also eliminate the error due to power frequency (50 Hz.) pick-up [5.5].

The stability of reference voltage is a factor in the absolute accuracy of the converter. The resolution of ICL7109 at 12 bits is one part in 4096, or 244ppm. The on-chip reference source of ICL7109 has a temperature coefficient of 80ppm/°C typical [5.6] and will create a one-bit absolute error for a change of 3°C. On the other hand, LM336-2.5V is an adjustable voltage reference having low temperature co-efficient. Temperature coefficient can further be reduced by adding two silicon diodes to generate stable ADC reference voltage as shown in Fig. 5.12. The typical temperature coefficient after trimming the reference voltage at 2.490V between 0-70°C is 1.8 mV or around 10 ppm/°C only. The
reference is further scaled by 1.5K and 1.0K resistors to generate the 1024 mV ADC reference. As ICL7109 produces 2048 counts at its own reference, it defines a scale factor (count to mV ratio) of 2.

![ADC Reference Circuit](image)

*Fig. 5.12: ADC reference using LM336-2.5V with temperature co-efficient trimming arrangement.*

### 5.3.1.5 ROM

The ICL7109 ADC can address 4096 locations. For a 4-digit display, two 8-bit locations are necessary to store a process value. Therefore, two 4KB blocks of ROM are required to store 4096 points of the process value. Thus an 8KB EPROM 2764A is chosen. Moreover, if the full range of the ADC (4096 counts) be utilized for a process value in the range of 0-1000 units, each unit corresponds to nearly four ROM locations. This reduces the effects of non-linearity and inaccuracy of the ADC and the amplifier four folds to the results.
5.3.1.6 Display
4-digit 7-segment display is chosen as the temperature range is 0 – 1200°C. Lower two displays (#1 and #2) are fed by the lower 4KB ROM data and upper two displays (#3 and #4) are fed by the upper 4KB ROM data. Data bus D3-D0 supplies data for display #1 and #3 and D7-D4 on the other hand carries data for display #2 and #4.

5.3.1.7 Display Driver
Each LED of a 7-segment display requires 7-8 mA of current for acceptable intensity and thus LED drivers are required to drive the display. CD4511 is a commonly used BCD to 7-Segment decoder driver IC and used to display the BCD data stored in ROM into the 7-segment display. Four such ICs are used to drive 4-digit display unit.

5.3.1.8 Clock Generator
The same data bus is used to drive different displays. For example D0-D3 drives display #1 and #3 and D4-D7 drives display #2 and #4. Thus the displays are to take data in a time multiplexed manner. A clock generator is used to multiplex displays. When clock is high it drives display #1 and #3 and when low it drives display #2 and #4. Therefore, clock and inverted clock are necessary for multiplexing. IC 7414, hex Schmitt-trigger inverter, is used for the purpose. One of the inverters left unused in 7414 is used to generate the clock by connecting a capacitor in the input of a Schmitt-trigger inverter and a resistance in the feedback loop. The duty cycle of the clock is made nearly 50% for uniformity of the displays. Fig. 5.13 depicts the clock generator.

![Clock Generator](image)

*Fig. 5.13: Clock Generator used for display multiplexing*
Capacitor charging and discharging time and the frequency \([5.7]\) of oscillation as shown in Fig. 5.14 are given by,

\[
t_1 = RC \ln \frac{V_{T+}}{V_{T-}}
\]

\[t_2 = RC \ln \frac{V_{CC} - V_{T-}}{V_{CC} - V_{T+}}\] \[5.8\]

\[
f = \frac{1}{RC \ln \frac{V_{T+}(V_{CC} - V_{T-})}{V_{T-}(V_{CC} - V_{T+})}}\] \[5.9\]

### 5.3.1.9 Hardware Implementation

Fig. 5.15 shows the detailed schematic circuit diagram. The thermocouple processing circuit including cold junction compensation as shown in schematic circuit diagram (Fig. 5.10) is designed around the quad op-amp LM324 (U8). The excitation current for the Pt-100 RTD employed for ambient temperature sensing for automatic cold junction compensation as shown in schematic circuit in Fig. 5.11 is designed around the
operational amplifier U8d. The reference voltage of the ADC is generated by ZD1 (LM336-2.5V).

The clock and the inverted clock as shown in the schematic circuit of Fig. 5.13 are implemented using the gates U7d and U7e of 74LS14, a hex Schmitt Trigger inverter. Other two inverters U7c and U7a are used to drive LED5 and LED6 indicating polarity and over range of the ADC ICL7109 respectively. The digital outputs (B12 to B1) of the ADC are connected to the address lines (A11 to A0) of the 8 KB EPROM 2764A. The most significant address line A12 of 2764A is driven by the clock generator. The data outputs (D7 to D0) of the 2764A are connected to the inputs of four 4511s, BCD to 7-Segment decoder drivers (U3 to U6), to drive the 4-digit 7-Segment Display Unit. All the display segments are individually connected with the decoder outputs through current limiting resistances to ensure uniform intensity.

The Single Sided PCB layout of the circuit is shown in Fig. 5.16 (not in scale). An accuracy of 1°C is aimed for type ‘K’ thermocouple calibrated in the temperature range of 0-1000°C. Therefore special attention has been given for layout of the analog section of the ADC. The different ground lines for analog circuit, digital logic and display are made separate on the PCB and connected at different pins of the connector so that all would be tied together at the supply end only. This is to minimize the interference of current spikes on the analog circuit arising from digital logic and display. Bypass capacitors are placed on the supply line near all the ICs to make the supply stable. Ground and supply lines are made wider to minimize the impedance to avoid unnecessary voltage drops along the lines. They are also laid side by side as far as practicable to improve capacitive coupling between them to bypass the noise more effectively.
Fig. 5.15: Detailed Schematic Circuit Diagram of the ROM Linearizer
Fig. 5.16: The Layout of the ROM Linearizer in a Single Sided PCB.
5.3.1.10 Results

The performance of the circuit is checked against two reference temperatures viz. at the melting point of ice (0\(^{\circ}\)C) and at the boiling point of water (100\(^{\circ}\)C) and found no deviations. Performance of the instrument throughout the range is verified by simulating the mVs of a type “K” thermocouple from a reference voltage source. The experimental results are presented in Table 5.4.

Table 5.4: Experimental Results of the ROM Linearizer

Ambient temperature: 31\(^{\circ}\)C.
Corresponding Thermocouple type “K” emf: 1.244 mV.

<table>
<thead>
<tr>
<th>SL NO</th>
<th>TEMP (°C)</th>
<th>T/C O/P wrt 0°C (mV)</th>
<th>T/C O/P wrt 31°C (mV actually fed)</th>
<th>OBSERVED TEMP (°C)</th>
<th>DEVIATION (°C)</th>
<th>ACCURACY OF RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>1.244</td>
<td>0.000</td>
<td>31</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>4.095</td>
<td>2.851</td>
<td>100</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>8.137</td>
<td>6.893</td>
<td>201</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>12.207</td>
<td>10.963</td>
<td>301</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>16.395</td>
<td>15.151</td>
<td>401</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>500</td>
<td>20.640</td>
<td>19.396</td>
<td>501</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>600</td>
<td>24.902</td>
<td>23.658</td>
<td>601</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>700</td>
<td>29.128</td>
<td>27.884</td>
<td>701</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>800</td>
<td>33.277</td>
<td>32.033</td>
<td>801</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>900</td>
<td>37.325</td>
<td>36.081</td>
<td>901</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>995</td>
<td>41.074</td>
<td>39.830</td>
<td>995</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

5.3.1.11 Conclusion

From the experimental data it is seen that the accuracy aimed for is achieved. The circuit is tested with thermocouple input. However, the circuit can accept any sensor. For a different sensor, the data of the ROM are to be generated and downloaded in the
EPROM. Also a different gain of the amplifier may be set. The interactive executable program generates the ROM data, as well as the new gain of the circuit to be adjusted.

5.3.2 A CPLD Implementation of a Circuit for Linearizing Transducer Characteristics

5.3.2.1 Introduction

Programmable Logic Devices (PLDs) are now becoming popular in industry because of its reduced design cycle and ease of program development through the Integrated Development Environment (IDE). Different PLD vendors are now also adding many useful flavors in their PLDs along with the basic design resources like memory, microprocessor, and other peripherals in the form of soft and hard IP cores. Different synthesis, optimization, simulation and implementation tools available in IDE make the complex digital system design a simple task. Some of the prominent PLD vendors are Xilinx, Altera, Actel who manufacture different families of PLDs. One family of PLDs differs from the other in terms of available resources and internal architecture. The author implemented a segment-wise linearization scheme in a Xilinx CPLD.

The design is described through Very high speed integrated circuit Hardware Description Language (VHDL) and is implemented in a Xilinx 9500 series CPLD (Complex Programmable Logic Device) chip. The Xilinx CPLDs use electrically erasable flash programmable devices to hold the configuring bit stream and hence are non-volatile. Besides, these devices are in-circuit programmable and hence can be reconfigured for different sensors without removing the circuit from the working system.

5.3.2.2 Linearizing Technique Used

Segment-wise Linearization Technique is adopted here as shown in Fig. 5.17. Process Value is plotted against digitized sensor output. The top most curve is the characteristics of a representative sensor, which is non-linear in nature. The entire curve is replaced by a number of short linear segments having known slope and end coordinates. The number of such segments determines the system accuracy. Naturally higher the number of segments, higher is the accuracy. As shown in Fig. 5.17 for a particular input $D_m$, the errors
produced by single-segment and three-segment systems are \( e_1 \) and \( e_2 \) respectively, where \( e_2 \) is much less than \( e_1 \).

![Fig. 5.17: Segment-wise Linearization Technique](image)

### 5.3.2.3 Implementation

Most important part of implementation of the linearizer is to locate the input data in the proper segment and then to calculate the process value from the straight line equation of that segment. As depicted in Fig. 5.17, if \( D_{in} \) be the input data we can find out by successive comparisons with breakpoints \( C_1, C_2, C_3 \) etc., that it is in the 2\(^{nd} \) segment having end co-ordinates \((C_1,T_1)\) and \((C_2,T_2)\). Then the process value \((T)\) can be calculated from the equation of the 2\(^{nd} \) straight line, i.e.

\[
T = \frac{T_2 - T_1}{C_2 - C_1} (D_{in} - C_1) + T_1 = S_2 (D_{in} - C_1) + T_1
\]

where, \( S_2 = (T_2-T_1)/(C_2-C_1) \) is the slope of 2\(^{nd} \) segment.

The generalized equation of the \( n^{th} \) segment can be established as:

\[
T = \frac{T_n - T_{n-1}}{C_n - C_{n-1}} (D_{in} - C_{n-1}) + T_{n-1} = S_n (D_{in} - C_{n-1}) + T_{n-1}
\]  \( (5.10) \)
Similar equations can also be set up for other segments. For implementation of these equations in CPLD, the constants $S_n$ (slope), $T_n$ (temp offset) and $C_n$ (count offset) of the $n^{th}$ segment as indicated in Equation (5.10), are to be selected for the computation. These values are stored in different register banks ($C_n$, $S_n$, and $T_n$) in the CPLD and are selected using Multiplexers M1, M2, and M3 as shown in Fig. 5.18. The control input of the multiplexers is determined from the result of comparison of Din with the break points. To compute the final results, arithmetic building blocks like multipliers, adders and substrators are required (Fig. 5.18), which are available in the Xilinx Xc9500 series of CPLDs.

![Fig. 5.18: Block Diagram of the Linearizer of equation (5.10)](image)

As seen in the block diagram, $C_x$, $S_x$ and $T_x$ are the register blocks used to store the different sets of constants of the short linear segments. The COMPARATOR is the control circuit which compares the input data $Din$ with $C_1$, $C_2$, $C_3$ etc. in succession to determine the segment to be considered presently and generates control signals for the multiplexers (M1, M2, M3) to select proper set of constants required to calculate the
process value. Subtraction, Multiplication and addition as required in Equation (5.10) are performed by the Subtractor, the Multiplier and the Adder respectively (Fig. 5.18).

### 5.3.2.4 Software Modeling

Implementation of any digital system in programmable logic devices (PLD) needs describing the hardware to the PC based CAD tool. The description of the Hardware can be input through State Diagram, Boolean expression, Schematic Circuit Diagram or in a Hardware Description Language (HDL). Schematic circuit diagram may be quite complex for a large circuit. On the other hand if the circuit is described in a structured language then it is easier to follow the complex logic, loop may be used to describe repetitive blocks. This makes analysis for next level of processing simpler. The Hardware Description Language VHDL is used here to describe the circuit of Fig. 5.18.

Xilinx *Integrated Software Environment (ISE)* VLSI Design CAD Tool is used for circuit modeling, simulation and implementation. The linearizer is designed for PT100 sensor for a temperature range of 0 to 250°C. Break Points are calculated by a program ‘LNZ.FOR’ developed in FORTRAN for an accuracy of 0.2°C giving rise to four linear segments. Segment constants so obtained are listed in Table-5.2.

*Table-5.2: Segment Constants*

<table>
<thead>
<tr>
<th>Break Points</th>
<th>Tx (°C)</th>
<th>Cx (Count)</th>
<th>Sx</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>070</td>
<td>073</td>
<td>0.95312500</td>
</tr>
<tr>
<td>#2</td>
<td>140</td>
<td>145</td>
<td>0.97265625</td>
</tr>
<tr>
<td>#3</td>
<td>213</td>
<td>218</td>
<td>0.99609375</td>
</tr>
<tr>
<td>#4</td>
<td>250</td>
<td>255</td>
<td>0.99609375</td>
</tr>
</tbody>
</table>

The listing of the FORTRAN program is given in Appendix 5. It is an interactive program and asks for input data file name, output file name, operating range, accuracy required, and gain of the circuit (including the conversion gain of the ADC) required for generating full scale ADC output. A screen shot of the program while running is shown in Fig. 5.19.
The input is supplied through 8-bit binary data. The circuit is designed to produce the temperature ($^\circ$C) in the form of an 8-bit output data. The multiplier block of course takes two 8-bit numbers and produces a 16-bit product. However, the slope data of Table–5.2 is appropriately scaled so that the higher 8-bits of the product are only significant. The LSB of the result thus obtained is rounded off using bit-7 of the product.

5.3.2.5 VHDL Description of the Circuit

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity linearizer is
    Port ( Din: in std_logic_vector(7 downto 0);
          Dout: out std_logic_vector(7 downto 0));
end linearizer;
architecture behavioral of linearizer is
--segment equations
```
--any unknown temperature in the 1st segment can be found using
--equation:
--\[ t = t_0 + \frac{dt}{dc}(c-c_1), \]
where \((c_0,t_0), (c_1,t_1)\) etc. –are the break points.
--For pt100 RTD, gain of the amp is chosen such that ADC will produce
--a count of 255 for 250° C.
--Further, the accuracy for segmentation is taken as 0.2° C which is
--well within the resolution of the system (1° C).
--Slopes of the straight line segments are always less than one
--(as change in temperature “\(dt\)” is always less than the corresponding
--change in input data “dc” and thus expressed in 8-bit binary fraction):

constant slope0: std_logic_vector(7 downto 0) :=b"1111_0100";
--0.953125d
constant slope1: std_logic_vector(7 downto 0) :=B"1111_1001";
--0.97265625d
constant slope2: std_logic_vector(7 downto 0) :=B"1111_1111";
--0.99609375d
constant slope3: std_logic_vector(7 downto 0) :=B"1111_1111";
--099609375d
--temperature offsets of the break points:
constant t0: std_logic_vector(7 downto 0) :=B"0000_0000"; --0d
constant t1: std_logic_vector(7 downto 0) :=B"0100_0110"; --70d
constant t2: std_logic_vector(7 downto 0) :=B"1001_0001"; --140d
constant t3: std_logic_vector(7 downto 0) :=B"1101_0101"; --213d

--ADC counts at the break points:
constant c0: std_logic_vector(7 downto 0) :=B"0000_0000"; --0d
constant c1: std_logic_vector(7 downto 0) :=B"0100_1001"; --73d
constant c2: std_logic_vector(7 downto 0) :=B"1001_0001"; --145d
constant c3: std_logic_vector(7 downto 0) :=B"1101_1010"; --218d
--constant c4: std_logic_vector(7 downto 0) :=B"1111_1111"; --255d
signal cur_slope: std_logic_vector(7 downto 0);
signal cur_cx: std_logic_vector(7 downto 0);
signal cur_tx: std_logic_vector(7 downto 0);
signal result: std_logic_vector(15 downto 0);

begin
  P1:process(Din)
    begin
      if Din < c1 then
        cur_slope <= slope0;
        cur_cx <= c0;
        cur_tx <= t0;
      elsif Din < c2 then
        cur_slope <= slope1;
        cur_cx <= c1;
        cur_tx <= t1;
      elsif Din < c3 then
        cur_slope <= slope2;
        cur_cx <= c2;
        cur_tx <= t2;
      else
        cur_slope <= slope3;
        cur_cx <= c3;
        cur_tx <= t3;
      end if;
    end process P1;
  result <= (Din – cur_cx) * cur_slope;
  Dout <= cur_tx + result(15 downto 8) + result(7);
end behavioral;
5.3.2.6 Experimental Data

The circuit is implemented in a VLSI Trainer Kit having Xilinx CPLD XC9536 (44 pin). The experiment is carried out by supplying input data through an 8-way DIP switch. The experimental results are presented in the Table-5.3 below:

<table>
<thead>
<tr>
<th>Input Count (Din)</th>
<th>Output Temp (Dout) in °C</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX</td>
<td>Corresponding Temp (°C)</td>
<td>HEX Value</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>00</td>
</tr>
<tr>
<td>19</td>
<td>23.67</td>
<td>18</td>
</tr>
<tr>
<td>36</td>
<td>51.38</td>
<td>33</td>
</tr>
<tr>
<td>4B</td>
<td>71.55</td>
<td>48</td>
</tr>
<tr>
<td>64</td>
<td>95.77</td>
<td>60</td>
</tr>
<tr>
<td>82</td>
<td>125.05</td>
<td>7D</td>
</tr>
<tr>
<td>96</td>
<td>144.70</td>
<td>91</td>
</tr>
<tr>
<td>AF</td>
<td>169.47</td>
<td>AA</td>
</tr>
<tr>
<td>C8</td>
<td>194.21</td>
<td>C3</td>
</tr>
<tr>
<td>E1</td>
<td>219.56</td>
<td>DC</td>
</tr>
<tr>
<td>FF</td>
<td>250.00</td>
<td>FA</td>
</tr>
</tbody>
</table>

5.3.2.7 Conclusion

- The above experimental table shows that the temperature accuracy is limited by the resolution of the system (1 °C).
- Higher accuracy can be achieved by either increasing the number of input bits or by decreasing the temperature range.
- The binary output of the linearizer representing temperature in °C can be displayed in 7-segment display using binary to BCD converter followed by a BCD to 7-segment decoder. The whole conversion mechanism can also be implemented in the CPLD itself producing 7-segment output directly.

Both the methods described in this chapter are simple, low cost and easy to implement. However, the ROM Linearizer requires display driver, clock generator, EPROM and other components to be added externally to form the system. On the other hand, in the CPLD based Linearizer, all the logical circuits can be implemented within the CPLD itself.