REAL TIME DIGITAL SIGNAL PROCESSING APPLICATIONS WITH IBM-PC AND TMS 320

Mohamed El-Sharkawy Zhilie Li Jinfeng Wang

Electrical Engineering Department
Bucknell University
Lewisburg, PA 17837
(717) 524-1459

ABSTRACT

In this paper, two real-time digital signal processing applications with IBM personal computer and Texas Instruments TMS320 programmable digital signal processor are presented. In the first application, the IBM personal computer is used to control the Miller Profiler to perform real-time Deep Level Transient Spectroscopy (DLTS) measurements. The second application describes the implementation of a real-time Recursive Adaptive Line Enhancer (RALE) on the Texas Instruments TMS320 programmable digital signal processor. The software logic and flow of each of the adaptive stochastic program, the TMS320 processing program and the IBM control and display program are presented. A schematic diagram of the TMS320-based hardware model is also included.

I. DLTS MEASUREMENTS

In the first application, the IBM-PC and the data translation board DT2805/5716 (Figure 1) are used to control the Miller Profiler to perform real-time Deep Level Transient Spectroscopy (DLTS) measurements [1-4]. The DLTS is a technique to evaluate the relationship between the deep level density \( N_T \) and the distance into the semiconductor \( x_A \) at which the deep level \( E_T \) crosses the bulk Fermi level \( E_F \). A large number of different kinds of semiconductor transient experiments can be conducted, using optical, electrical, or thermal stimulation and exhibiting a wide range of time constants (nanoseconds to hours). Although the possible variations in such experiments, many of them have a common feature; namely the existence of exponential transients as the physical system approaches a steady state following the application of an abrupt disturbance [1]. The transient signal \( V_A(t) \) can be described by the following equation:

\[
V_A(t) = \frac{q}{2\varepsilon_o} \int_0^W N_S(W) \cdot \left( 1 - e^{-(W-x)/\tau} \right) dx_A \left( N_T(W) \right)
\]

where \( \varepsilon_o \) is the permittivity of free space, \( \varepsilon \) is the relative dielectric constant of the semiconductor, \( V_D \) is the zero bias potential difference across the barrier, \( q \) is the electronic charge, \( W \) is the depletion depth, \( x_A \) is the distance into the semiconductor at which the deep level crosses the bulk Fermi level, \( \tau \) is the time constant, \( N_S(W) \) and \( N_T(x_A) \) are the average shallow and deep impurity densities.

To obtain the transient response, the sample under study is briefly forward biased by a command from the IBM-PC then the sample is reverse biased and maintained at a pre-selected depletion depth through automatic adjustment of the bias.

Let

\[
V_A(X_A) = V_B(0) - V_B(\tau)
\]

Using equation (1), equation (2) can be written as:

\[
V_A(X_A) = \left[ \frac{q}{2\varepsilon_o} \right] X_A^2 N_S(W) \left( N_T(W) \right)
\]

where

\[
X_A = W - \lambda(W)
\]

\[\lambda(W) = \frac{2\varepsilon_o (E_F - E_T)}{qN_S(W)} \]

and \( N_S(W) \) is the shallow impurity density at depletion depth \( W \).

The relationship between the deep level density \( N_T \) and \( X_A \) can be written as [5]:

\[
\frac{d V_A(X_A)}{d X_A} = \frac{d}{d X_A} \left( \frac{\varepsilon_o}{q} X_A \right)
\]

Several measurements of \( V_A \) and \( X_A \) are used to calculate the derivative of \( V_A \) with respect to \( X_A \) and to generate the desired graph of the deep level density \( N_T \) versus \( X_A \).

The deep level trap energy level \( E_T \) is calculated using either a proposed real-time algorithm or a real-time cross correlation between the experimental signals and an appropriately synchronized locally generated exponential wave form [1].

In the proposed method, the transient response is modeled by:

\[
V(t) = A e^{-t/\tau} + B
\]

where \( B \) is equal to the preset bias voltage.

Equation (7) can be rewritten as:

\[
y(t) = C + D t
\]

where

\[
y(t) = \ln(V(t) - B) - \ln A
\]
method to obtain an approximated value for ratios. A relatively fast and accurate results output port and an IBM-PC.

The RALE hardware module consists of a TMS320C10, shown in Figures 5 and 6. The RALE model has the following specifications:

<table>
<thead>
<tr>
<th>Microprocessor</th>
<th>TMS320C10</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Clock</td>
<td>8 MHz</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>8 kHz</td>
</tr>
<tr>
<td>Memory</td>
<td>1K Words EPROM</td>
</tr>
<tr>
<td>Input/Output</td>
<td>One Parallel Port for Communication with IBM PC. One A/D Converter (ADC80C)</td>
</tr>
<tr>
<td>Software</td>
<td>Two Program for the IBM PC and the RALE Board</td>
</tr>
<tr>
<td>Reset</td>
<td>Boot-up Circuit to run the TMS320 from Program Address Zero</td>
</tr>
<tr>
<td>Power Requirements</td>
<td>+5 V/1.1A, +12 V/10mA, -12 V/-20mA</td>
</tr>
<tr>
<td>Board Dimensions</td>
<td>6.35&quot; x 9.15&quot;</td>
</tr>
</tbody>
</table>

Table 1 shows a comparison between the proposed algorithm and the correlation method for different signal to noise ratios and time constants. The proposed method is much faster than the correlation method. Both methods have similar results for signal to noise ratios larger than 10 dB. The correlation method has a better performance for lower signal to noise ratios. A relatively fast and accurate results can be obtained by first running the proposed method to obtain an approximated value for \( r \) then use this value and the correlation method to obtain an accurate value for \( r \).

II. RECURSIVE ADAPTIVE LINE ENHANCER

The second application describes the implementation of a real-time Recursive Adaptive Line Enhancer (RALE) on the Texas Instruments TMS320 programmable digital signal processor. The RALE algorithm proposed in this paper is based on the adaptive stochastic algorithms recently developed by El-Sharkawy et al. [6-7]. This adaptive stochastic filter avoids both the Strict Positive Real (SPR) condition used as a sufficient condition for convergence and the stability check of the filter parameters at each iteration to adjust the parameters if found unstable. The SPR condition and/or the stability check of the filter parameters limit the real-time implementation of the RALE.

The RALE hardware module consists of a TMS320C10, an analog/digital converter, two EPROMS, a data output port and an IBM-PC. A schematic diagram and close-up photographs of the RALE module are shown in Figures 5 and 6. The RALE model has the following specifications:

- **Microprocessor**: TMS320C10
- **System Clock**: 8 MHz
- **Sampling Frequency**: 8 kHz
- **Memory**: 1K Words EPROM
- **Input/Output**: One Parallel Port for Communication with IBM PC. One A/D Converter (ADC80C)
- **Software**: Two Program for the IBM PC and the RALE Board
- **Reset**: Boot-up Circuit to run the TMS320 from Program Address Zero
- **Power Requirements**: +5 V/1.1A, +12 V/10mA, -12 V/-20mA
- **Board Dimensions**: 6.35" x 9.15"

After the data is processed by the TMS320C10, the coefficients of the transfer function are sent to the IBM-PC. The IBM-PC receives the data through its parallel port, calculates and displays the estimated spectra. All operations are controlled by the software running on the IBM-PC and the RALE board. The TMS320 program is the processing program written in TMS320 assembly language. Its function includes receiving data, processing data through the adaptive stochastic filter, output the estimated coefficients to the IBM-PC, polling the data output port and generating the handshaking signals. The IBM-PC program is the control and display program. Its function includes starting the TMS320, receiving data from the TMS320, calculating and displaying the estimated spectra. The RALE system was tested under different signal to
noise ratio (SNR) and over a wide range of parameter values to illustrate its usefulness. In these tests, the input data y(k) was obtained by adding a white noise v(k) to a sinusoidal input x(k). The sinusoidal input can be represented by

\[ x(k) = u(k)/A(q^{-1}) \]  

(9)

where

\[ A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} \]

and u(k) is an unknown white noise process.

We can therefore write

\[ y(k) = x(k) + v(k) \]  

(10)

or, using (9), this can be rewritten as

\[ A(q^{-1}) y(k) = u(k) + A(q^{-1}) v(k) \]  

(11)

or equivalently

\[ A(q^{-1}) y(k) = C(q^{-1}) e(k) \]  

(12)

where

\[ C(q^{-1}) e(k) = u(k) + A(q^{-1}) v(k) \]  

(13)

The objective of the RAKE system is to estimate the coefficients of the polynomials A(q^{-1}) and C(q^{-1}) from the input data y(k). The estimated parameters are then used to calculate and display the estimated spectra on the IBM-PC screen.

A brief description of the parameter adaptation algorithm is presented next. A flow chart of the three stage adaptive stochastic filter is shown in Figure 7. For more details see references 6 and 7.

The parameter adaptation algorithm is:

\[ \hat{e}_j(k+1) = \hat{e}_j(k) + \hat{u}_j(k) \hat{e}(k+1)/bt(k) \]  

(14)

and

\[ f_{ij}(k+1) = f_{ij}(k) - \hat{u}_j(k) \hat{u}_j(k)/bt(k) \]  

(15)

\[ 1 = 1, \ldots, 4 \]  

and \[ j = 1, \ldots, 4 \]

where

\[ \hat{u}_j(k) = \sum_{j=1}^{4} f_{ij}(k) \hat{e}_j(k) \]  

(16)

\[ \hat{e}_j(k) = \sum_{j=1}^{4} f_{ij}(k) \hat{e}_j(k) \]

\[ bt(k) = \sum_{j=1}^{4} \hat{u}_j(k) \hat{u}_j(k) \]

\[ \hat{e}_j, \hat{e}_j, \hat{u}_j, \hat{u}_j, \hat{u}_j, \hat{u}_j, \hat{u}_j \] are the estimated parameters \[ a_1, a_2, c_1 \] and \[ c_2 \] and \[ \epsilon \] is the error signal.

Table 2 shows the estimated parameter values for different frequencies and signal to noise ratios. It is noted that the close match between the true frequencies and the estimated frequencies.

REFERENCES

Table 1: Estimated Time Constants

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>SNR (dB)</th>
<th>Estimated ( t ) ms</th>
<th>Estimated Correlation Method</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
<td>0</td>
<td>15333</td>
<td>-10964</td>
<td>1530</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>15333</td>
<td>-12864</td>
<td>2046</td>
</tr>
<tr>
<td>2500</td>
<td>0</td>
<td>-381</td>
<td>-12488</td>
<td>2521</td>
</tr>
<tr>
<td>3000</td>
<td>0</td>
<td>-10869</td>
<td>-12859</td>
<td>3121</td>
</tr>
<tr>
<td>1500</td>
<td>-5.5</td>
<td>11718</td>
<td>-5992</td>
<td>1493</td>
</tr>
<tr>
<td>2000</td>
<td>-5.5</td>
<td>6439</td>
<td>-9245</td>
<td>2078</td>
</tr>
<tr>
<td>2500</td>
<td>-5.5</td>
<td>-727</td>
<td>-8087</td>
<td>2550</td>
</tr>
<tr>
<td>3000</td>
<td>-5.5</td>
<td>-6231</td>
<td>-7322</td>
<td>2959</td>
</tr>
</tbody>
</table>

Table 2: Estimated Parameters and Frequencies for Different Signal to Noise Ratios and Frequencies

Table 2: Estimated Parameters and Frequencies for Different Signal to Noise Ratios and Frequencies.

Figure 1: DLTS Measurements with IBM-PC
Figure 2 Input Waveforms

Figure 3 The Correlator Output as Function of the Input Signal Time Constant

Figure 4 Deep Level Density $N$ Versus the Distance $\lambda$
Figure 5 Hardware Schematic of the Recursive Adaptive Line Enhancer System

Figure 6 Recursive Adaptive Line Enhancer System
Figure 7
Flow Chart of Adaptive Recursive Algorithm