On the Implementation of Differential Encoder for Spectral Shaping in 56Kbps Embedded Modems

Davinder Pal Sharma
Department of Physics, University of the West Indies, St. Augustine Campus, Tinidad & Tobago, West Indies
davinder.sharma@sta.uwi.edu

Jasvir Singh
Department of Electronics Technology, Guru Nanak Dev University, Punjab, India
j_singh00@rediffmail.com

ABSTRACT

Present paper deals with the simulation and implementation of two functional units, parse-to-shaping-frame and differential encoder, for spectral shaping in 56Kbps digital modem transmitter. The idea behind spectral shaping is to adapt the shape of the transmitted signal, to conform to the shape or spectral limitations of the channel, without changing the basic pulse shape or peak-to-average-ratio. This unit suppresses the signal component close to dc to minimize the effect of ac couplings or to provide sufficient data transitions for reliable clock recovery. A combined algorithm for implementation of the parse-to-shaping-frame and differential encoder functions utilized in transmitter of 56Kbps digital modem has been presented. An algorithm to perform parse-to-shaping-frame and differential encoding functions has been developed during present study. Proposed algorithm has been simulated and implemented on the Digital Signal Processor. Practical results obtained have been found almost similar to the theoretical and simulated results.

Keywords: Spectral Shaping, 56Kbps Modem, Differential Encoder, Digital Signal Processor.

1 INTRODUCTION

Technologies are changing very frequently in the field of data communication over twisted copper pair (plain old telephone lines). These changes are due to quick advancement in computer and digital communication technologies along with powerful digital signal processing algorithms [1]. In the present cyber age everyone wants to enjoy Internet services like teleconferencing, web-surfing, e-learning, e-banking, online movies and voice-over-telephony at very lower cost, which demands high speed. Even though it has been repeatedly predicted that network access via telephone lines would be replaced by new services based on emerging technologies [2], 56Kbps voice-band modems seem to be the best solution [3], [4] as these modems are still used by the majority of home computer users and small business owners for data communication and network access. In accordance with a study made by Georgia Tech, in 1998 approximately 70% of Internet users were connected to the network with analog voice-band modems and according to the survey conducted by a firm (Jupiter Communication) more than 50 million people in the US alone were using telephone dial up technology to access the Internet in 2001 and the user strength is growing up continuously. The Gartner Group estimates that about 55% of the user were relying on voice-band services even till 2004[5], [6]. Moreover voice-band modem has many advantages over others like they are inexpensive, easy to install, more reliable, widely available and easy in functioning.

Seeing the huge consumer market of voice-band modem for Internet access, present study was carried out on data transmission over analog telephone lines. V.90/V.92 is the current 56Kbps modem standard over the PSTN telephone lines, which uses entirely different technology. Block diagram of 56Kbps modem communication system is shown in Fig. 1. Traditional analog modem like V.34 assumes both the ends of the modem session to have an analog connection to PSTN whereas V.90/V.92 standard assumes one end of the modem section to be purely digital to take the advantage of high-speed connection [7]. Internet Service Providers (ISPs) are already using digital connection at their end. There is only one analog portion on the downstream transmission path (from ISP to DTE) and the upstream data conforms to the V.34 standard. TCM is used in upstream direction whereas in downstream

Pulse Coded Modulation (PCM) as specified by ITU in G.711 recommendation [8] is used in this modem, also known as PCM modem.

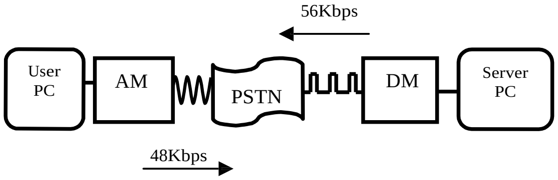


Figure 1: Block diagram of 56Kbps modem communication system

2 DISCRIPTION OF 56Kbps DIGITAL MODEM TRANSMITTER

Transmitter of V.90/V.92 56Kbps digital modem is shown in the Fig. 2. First unit is scrambler whose purpose is to facilitate effective transmission of the data over the telephone channel and to improve the convergence of the adaptive equalization and echo cancellation in the receiver. It helps the receiver to recover the timing information from the received data to facilitate synchronous operation. The downstream encoder in Draft Recommendation V.90/V.92 uses multiple modulus conversion for mapping scheme and convolutional spectral shaping as its spectral shaping scheme. The block diagram in Fig. 3 shows an overview of the downstream encoder and represents one data frame. Data frames in the digital modem have a six-symbol structure (since the robbed-bit signaling pattern repeats every six symbols). Each symbol position within the data frame is called a data frame interval and is indicated by a time index, i = 0, ..., 5.

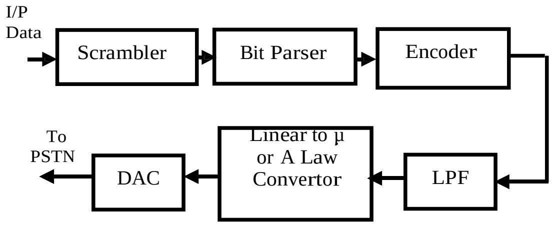


Figure 2: Transmitter of 56Kbps digital modem (server side)

During startup, the following encoder parameters are established:

- C_i equals the positive constellation points for data frame interval i .
- M_i is the number of code points in each constellation C_i .
- K is the number of modulus encoder input data bits per data frame.
- S_r is the number of PCM code sign bits per data frame used as redundancy for spectral shaping.
- S is the number of differential encoder input data bits per data frame, where $S + S_r = 6$.

The positive constellations (C_i) to be used in each data frame interval are specified by the analog modem during training procedures. The signaling rate is determined by the selection of the parameters

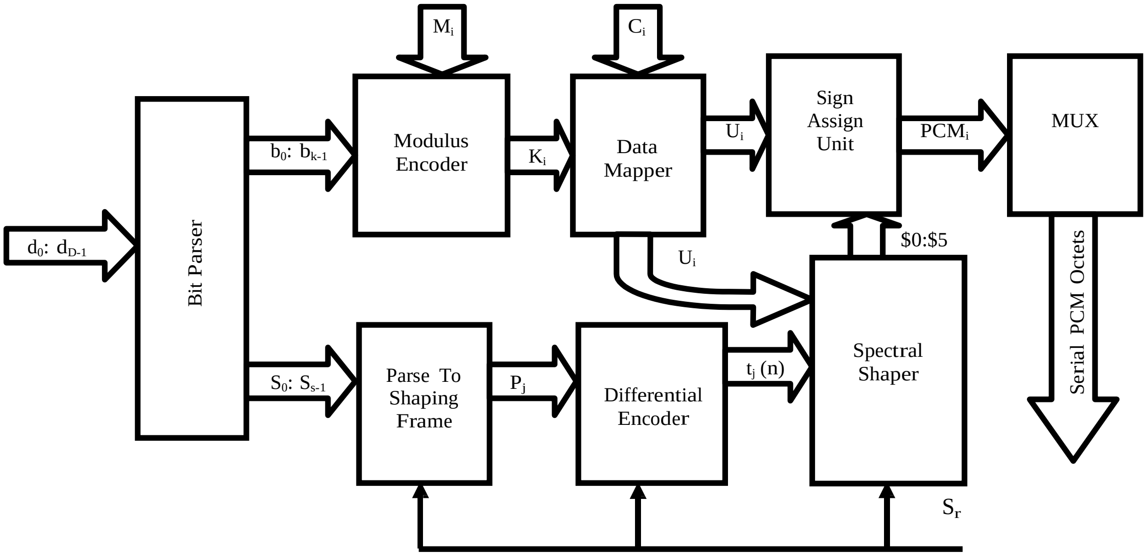


Figure 3: Encoder of 56Kbps digital modem

K and S during the startup phase using formula given by

$$Ds = [(K + S) \times 8000]/(6) \tag{1}$$

Description of each of the components or functional blocks as presented in Fig. 3 is given below [7]:

Bit Parser partitions the block of binary data for one mapping frame into different groups of bits for processing by subsequent stages of the transmitter. It takes bits from scrambled data stream and parses them into two groups, which are fed to two different parts of encoder i.e. to differential encoder and modulus encoder. It takes D (equal to S + K) scrambled input data bits ($d_0 : d_{D-1}$) and parses them into K modulus encoder input bits ($b_0 : b_{K-1}$) and S differential encoder input bits ($s_0 : s_{S-1}$). The modulus encoder takes K bits from the bit parser and maps them into six integers K_i, where i = 1, 2 ,…, 6. Each K_i is an integer between 0 and M_i, where the M_i's are called the mapping moduli and represent the number of elements in each of the PCM code sets defined for data frame interval 0 to data frame interval 5. In order to be able to represent the information in the K bits taken from the parser with these six integers, the values of M_i and K must satisfy the following inequality

$$K \le \prod_{i=0}^{5} (M_i) \tag{2}$$

Each frame interval has an independent mapper associated with it. Each one of them also has a tabulation of M_i PCM codes corresponding to the positive elements of the constellation to be used by it and denoted by C_i. The specific PCM codes that assemble each of the constellations are selected by the analog modem during the startup phase of the communication. It is required that the members of C_i should be labeled in descending order so the label 0 corresponds to the largest PCM code in the constellation and the label M_i correspond to the smallest. The output of each mapper (U_i) is generated by selecting the constellation point in C_i corresponding to K_i.

The S differential encoder input bits ($s_0 : s_{S-1}$) are parsed into j = S_r spectral shaping frames of length $6/S_r$. The six $P_j(n)$ bits are then differentially encoded to produce six input sign bits, $t_j(n)$, to the spectral shaper.

The spectral shaping is intended to change the shape of the spectrum of the transmitted signal to make it better suited to the channel used. Spectral shaping affects only the signs of the transmitted PCM symbols. The spectral shaper modifies input sign bits $t_j(n)$ to corresponding PCM code-sign bits ($\$_i$) so as to minimize the spectral shaping metric without violating the constraint specified in V.90/V.92 recommendation [7, 9]. The six sign bits generated by the spectral shaper ($\$_i$) are attached to the six unsigned mapper outputs (U_i) to form the six output symbols (PCM_i), which are then multiplexed to form the stream of PCM octets to be transmitted. This completes the encoding process.

8-bit PCM codes generated by the transmitter arrive at the central telephone office through the internal digital telephone network and are applied to the digital to analog converter in the Codec at the rate of 8000 samples per second. The Codec converts each code to one of 256 voltage levels and passes the resulting staircase waveforms through a low pass filter with a 4kHz cut-off frequency [10].

The linear to µ/A-Law Converter, who expands the 8 encoded PCM bits to 14 bits in accordance with the ITU recommendation G.711 [8]. The procedure of expanding 8-bit input to 14-bit data at transmitter and the compressing the 14-bit data to 8-bit at other end is called Companding. The device, which accomplishes this task, is called CODEC and is generally situated at the central office. A-law is used by European countries whereas in U.S.A. µ-law is popular.

Low pass filter in the modem design is generally used to avoid the aliasing problem caused by ADC in the communication path. To avoid the aliasing problem it should be ensured that the ADC never sees any signals that are to high in frequency. This is also known as anti-aliasing filter. As discussed above this filter has cut-off frequency equal to the bandwidth of the channel used i.e. 4 KHz. The output of filter is connected to the twisted pair (telephone line) through the hybrid circuit installed at the local telephone office of a client.

3 SPECTRAL SHAPING IN 56KBPS DIGITAL MODEM

Spectral shaping affects only the signs of the transmitted PCM symbols. From the six sign bits of each frame, S_r are the redundancy bits and S are the information bits. The number of redundancy bits S_r is determined by the analog modem during the startup procedures; it can take the values 0, 1, 2, or 3. By setting the value of S_r = 0, the spectral shaping capabilities can be disabled [7].

Spectral shaping is used to adapt the shape of the transmitted signal to conform to the shape or spectral limitations of the channel without changing the basic pulse shape or peak to average ratio. Spectral shaping codes achieve this objective by adding some redundant information or modifying the symbol sequence [11]-[13].There are many varieties of spectral shaping codes like those presented in [14]-

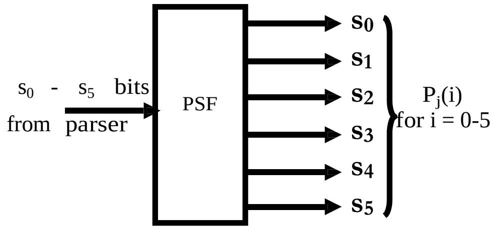
[18] depending on the particular requirements of each application. Typical applications of spectral shaping codes include the suppression of the signal component close to dc to minimize the effect of ac couplings, or to provide sufficient data transitions for reliable clock recovery. Simulation and implementation of functional units such as parse-to-shaping-frame and differential encoder of spectral shaping block of 56Kbps digital modem transmitter is discussed in the subsequent sections.

3.1 Parse to Shaping Frame

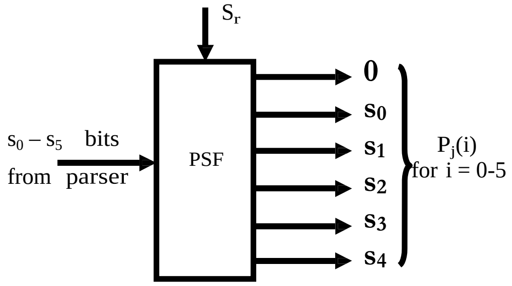
This unit takes S input bits (s_0-s_{S-1}) from the bit parser and parsed them into j = S_r spectral shaping frames of length $6/S_r$. This unit produces six outputs $P_j(n)$ where j(n) represents the n_{th} bit of the j^{th} spectral shaping frame in a data frame. The spectral shaping function depends on selected values of S_r, which may ranges from 0 to 3. S_r bits which are determined during startup procedure selects the value of S bits as $S_r + S = 6$. When $S_r = 0$ & S=6, spectral shaping is disabled and when $S_r = 1$ & S=5 sign bits s_0 to s_4 shall parse to one six-bit shaping frame per data frame according to Table 1. As per ITU Recommendations V.90/V.92, for S_r=2 & S=4, the sign bits s_0 to s_3 shall be parsed into two three-bit shaping frames per data frame and when S_r=3 & S=3, sign bits s_0 to s_2 shall be parsed to three two-bit shaping frames per data frame [7]. Block diagram of parse-to-shaping-frame (PSF) units for different value of S_r is given in Fig. 4. $P_j(i)$ is current shaping frame, $P_{j+1}(i)$ stands for next 1st frame and $P_{j+2}(i)$ stands for next 2nd frame.

Table 1: Parsing process of input sign bits

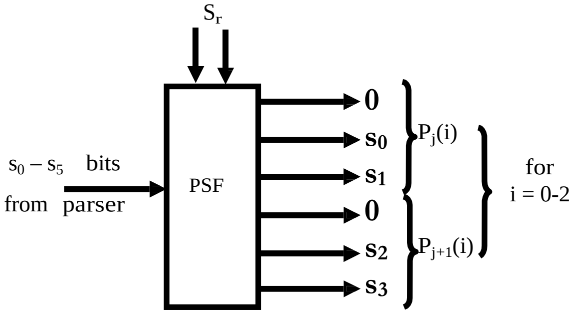
Data frame interval	$S_r = 1, S = 5$	$S_r = 2, S = 4$	$S_r = 3, S = 3$
0	$P_j(0) = 0$	$P_j(0) = 0$	$P_j(0) = 0$
1	$P_j(1) = s_0$	$P_j(1) = s_0$	$P_j(1) = s_0$
2	$P_j(2) = s_1$	$P_j(2) = s_1$	$P_{j+1}(0) = 0$
3	$P_j(3) = s_2$	$P_{j+1}(0) = 0$	$P_{j+1}(1) = s_1$
4	$P_j(4) = s_3$	$P_{j+1}(1) = s_2$	$P_{j+2}(0) = 0$
5	$P_j(5) = s_4$	$P_{j+1}(2) = s_3$	$P_{j+2}(1) = s_2$



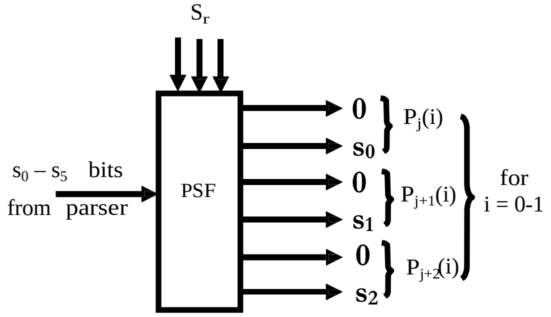
S_r = 0 and S = 6



S_r = 1 and S = 5



S_r = 2 and S = 4



S_r = 3 and S = 3

Figure 4: Block diagram of parse to shaping frame (PSF) units for different values of S_r

3.2 Differential Encoder

The power spectral density (PSD) of a digital communication signal can be controlled and shaped by selecting the transmitted signal pulse and by introducing correlation through coding, which is used to combat channel distortion and noise in transmission. Coding for spectrum shaping is introduced followed by the channel encoding so that the spectrum of the transmitted signal matches the spectral characteristics of a base band or equivalent low-pass channel. Codes that are used for spectrum shaping are generally called either modulation codes or line codes, or data translation codes. Such codes generally place restrictions on the sequence of bits into the modulator and thus introduce correlation and hence memory into the transmitted signal. Modulation codes are usually employed in digital communication over cable systems to achieve spectral shaping and to eliminate or minimize the dc content in the transmitted (or stored) base band signal [19].

Differential encoding technique has been recommended by the ITU for providing spectral shaping in V.90/V.92 56Kbps digital modem, which is basically Non-Return-to-Zero-Invert-on-ones (NRZI) line coding technique. In this scheme transitions from one amplitude level to another occurs only when a ‘1’ is transmitted. The encoding operation is described mathematically by the relation

$$b_k = a_k \oplus b_{k-1} \tag{3}$$

where $\{a_k\}$ is the binary information sequence into the encoder, $\{b_k\}$ is the output sequence of the encoder and $\oplus$ denotes modulo-2-addition operation. The differential encoding operation introduces memory in the signal. The most direct implementation of the differential encoder is to use an exclusive – OR (XOR) function with a delay in the feedback path as given in the Fig. 5 [19]-[20].

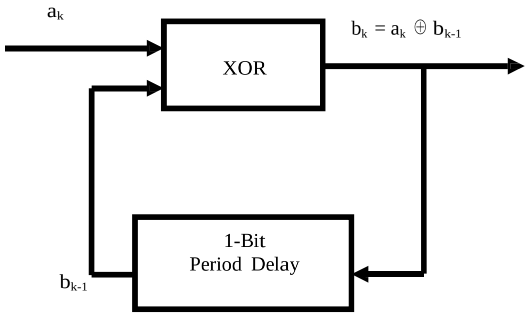


Figure 5: Differential encoder direct impleme - ntation using a XOR function.

For the present case spectral shaping function depends on the selected value of S_r. In the case of S_r=0 and S=6, the PCM code sign bits, $\$_0$ to $\$_5$ may be assigned using input sign bits s_0 to s_5 with the help of following differential coding rules:

$$\$_0 = s_0 \oplus (\$_5 \text{ of the previous data frame}); \text{ and}$$
$$\$_i = s_i \oplus \$_{i-1} \text{ for } i = 1, ..., 5 \tag{4}$$

When S_r=1 and S=5 the odd bits may be differential encoded to produce the output P_j' and a second order differential encoding may be performed to produce the initial shaping sign bit assignment, $t_j(0)$ to $t_j(5)$ using the rule

$$t_j(k) = P_j'(k) \oplus t_{j-1}(k) \tag{5}$$

Finally spectral shaper converts each bit $t_j(k)$ to PCM code sign bit $\$_k$.

For S_r=2 and S=4, after processing through parse to-shaping-frame, the odd bit in each shaping frame may be differentially encoded to produce outputs P_j' and P'_{j+1} and a second order differential encoding may be performed to produce the initial shaping sign bit assignment, $t_j(0)$ to $t_j(2)$ and $t_{j+1}(0)$ to $t_{j+1}(2)$ using the differential encoding rules:

$$t_j(k) = P_j'(k) \oplus t_{j-1}(k)$$
$$t_{j+1}(k) = P'_{j+1}(k) \oplus t_j(k) \tag{6}$$

and finally the spectral shaper converts each $t_j(k)$ bit to PCM code sign bit $\$_k$ and each $t_{j+1}(k)$ bit to PCM code sign bit $\$_{k+3}$.

In the case when S_r=3 and S=3, the odd bit in each shaping frame may be differentially encoded to produce differentially encoded outputs P'_j, P'_{j+1}, and P'_{j+2}. A second order differential encoding may be performed on each shaping frame to produce the initial shaping sign bit assignments $t_j(0)$ to $t_j(1)$, $t_{j+1}(0)$ to $t_{j+1}(1)$, and $t_{j+2}(0)$ to $t_{j+2}(1)$ using the differential encoding rules:

$$t_j(k) = P'_j(k) \oplus t_{j-1}(k)$$
$$t_{j+1}(k) = P'_{j+1}(k) \oplus t_j(k)$$
$$t_{j+2}(k) = P'_{j+2}(k) \oplus t_{j+1}(k) \tag{7}$$

The spectral shaper converts each $t_j(k)$ bit to PCM code sign bit $\$_k$, each $t_{j+1}(k)$ bit to PCM code sign bit $\$_{k+2}$, and each $t_{j+2}(k)$ bit to PCM code sign bit $\$_{k+4}$.

4 ALGORITHM FOR IMPLEMENTATION OF PARSE-TO-SHAPING-FRAME AND DIFFERENTIAL ENCODER

An algorithm to implement spectral shaping unit of the 56Kbps digital modem transmitter is shown in Appendix I, First of all, S sign bits (s_0-s_{1-S}) received from bit parser can be stored at appropriate data memory address (dma) and thereafter various 'dma' can be assigned for the storage of outputs of different stages of spectral shaping unit. After that S_r bit is received from the analog modem during startup operation and then its value can be checked. If it is equal to 0 then a subroutine for parse to shaping frame corresponding to S_r=0 and S=6 is followed otherwise query tasks for S_r=1 and S_r=2 can be performed and then corresponding subroutines LOOP-A, LOOP-B or LOOP-C can be followed.

Upon qualifying query task S_r=0, all the six sign bits (S) s_0-s_5 can be parsed as specified in the ITU Recommendation V.90/V.92. Bit masking technique can be used to perform this parse to shaping frame function and after parsing all the six bits they can be stored at an appropriate 'dma' P_{ji} for further processing by differential encoder. At this stage differential encoding on these parsed bits is performed as per Eq. (4) using the basic structure as given in Fig. 5 and finally obtained sign bits ($\$_0$ - $\$_5$) are stored at 'dma' t_{ji} , where i= 0-5 for j^{th} data frame.

If the analog modem sends S_r=1 during startup procedure then subroutine (a) can be followed. Five sign bits, out of six (s_0-s_5), are parsed as specified in Table 1. In this case bit-masking technique can also be used to perform parse to shaping frame function and after parsing all the six bits they can be stored at an appropriate 'dma' P_{ji} for further processing by differential encoding section. The next task is to perform differential encoding on odd bits of data frame interval 'j' using bits from present and previous data frame 'j-1' after odd bit differential operation, the encoded bits are stored at new 'dma' Pn_{ji}. In the next step, second order differential encoding according to Eq. (5) is performed on Pn_{ji} bits of data frame j and previously second order differential encoding bits. Finally the output of second order differential encoder is stored at 'dma' t_{ji}, which represents the corresponding sign bits and can be used to assign the sign to the PCM code words.

In the case, if query task for S_r=2 get satisfied, subroutine (b) can be used. Initially the data bits received from bit parser are again parsed according to Table 1 using bit masking technique and parsed bits may be stored at 'dma' Pji and P_{jp1i} corresponding to j^{th} and $j+1^{th}$ data frames. Then odd bit differential encoding is performed on odd bits of data frame 'j' and 'j+1' using bits from present (j) previous (j -1) and next (j+1) data frames. Then after encoded bits may be stored at 'dma' Pn_{ji} and Pn_{jp1i} for further processing. Furthermore to achieve sign bits, second order differential encoding may perform as per Eq. (6) using previously encoded bits corresponding to data frame j and j+1 along with the present bits. Process ends with the storage of so obtained sign bits to form PCM code word at 'dma' t_{ji} and t_{jp1i} corresponding to data frame j and j+1. Similar tasks can be performed for the case of S_r=3 using Eq. (7) corresponding to three data frames j, j+1 and j+2 to obtain sign bits to form PCM code words as per subroutine (c).

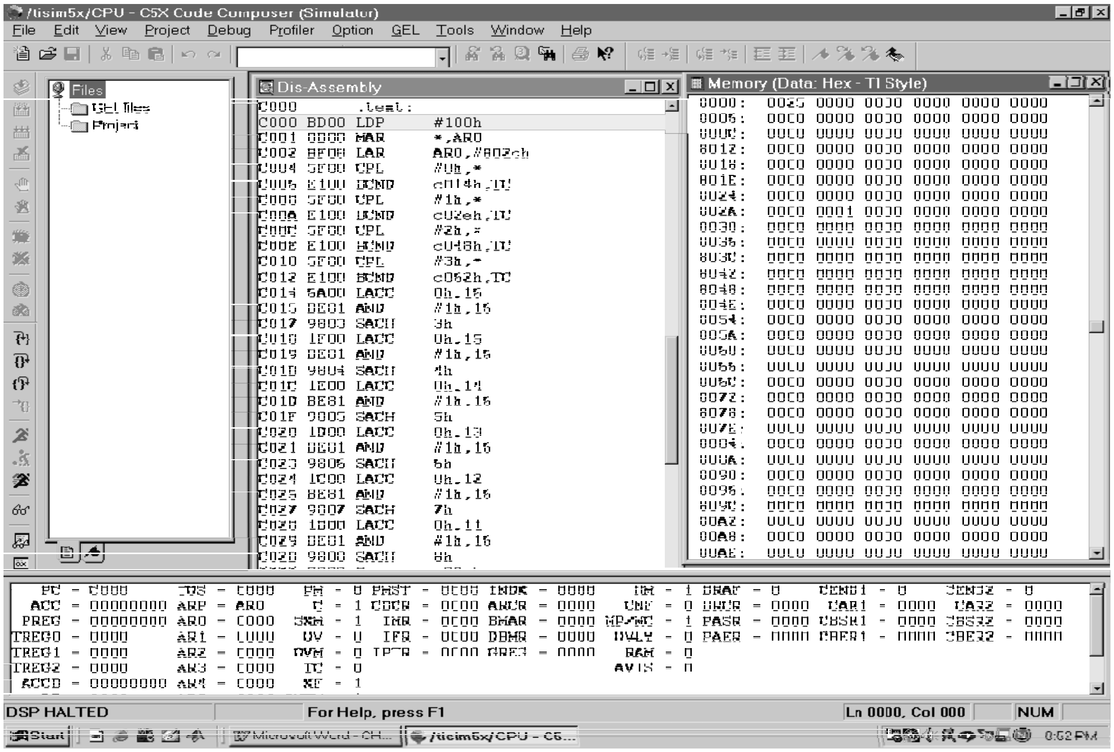
5 DEVELOPMENT OF ASSEMBLY LEVEL PROGRAM FOR PARSE - TO SHAPING - FRAME AND DIFFERENTIAL ENCODER

Assembly level program corresponding to the algorithm discussed in previous section to implement parse to shaping frame and differential encoding functions of 56Kbps digital modem transmitter on TMS320C50PQ57 DSP has been developed during present study. Query task has been implemented with the help of CPL (compare the specified constant with the contents of 'dma' specified by the Auxiliary resister) and BCND (conditional branch, get activated when carry-bit is high) instructions. Parse to-shaping-frame function has been implemented with the help of bit masking technique, which can be implemented with the help of AND instruction with appropriate shift. This is used to AND the content of accumulator with a constant, to mask a particular bit. Differential encoding (both odd bit and second order) function has been implemented with the help of XOR (the content of dma are exclusive-ORed with the contents of the accumulator) instruction. Various ‘dma’, which acts like 1-bit delay along with other supporting instructions, have been used to implement differential encoder as shown earlier in Fig. 5.

6 SIMULATION OF PARSE –TO – SHAPING FRAME AND DIFFERENTIAL ENCODER

For simulation purpose Code Composer Studio (CCS) software package from Texas Instruments, USA has been used. Assembly level program has been converted into the appropriate format for loading into the simulator of CCS using assembler and linker programs. Simulator status before execution of the spectral shaping program for S_r = 0 (stored at 'dma' 802CH) and incoming data 0025H (i.e. s_0 – s_5 having values 1,0,1,0,0,1 respectively) corresponding to data frame j (stored at 'dma' 8000H) along with the previous sign bit $\$_5$ =1(stored at dma 802BH), is shown in Fig.6. Program has been debugged and after execution simulation results have been obtained which are shown in Fig. 7.

Results corresponding to parse to shaping frame have been stored at 'dma' 8003H to 8008H, which are in accordance with the theoretically predicted results as per Table 1. Similarly the outputs of differential

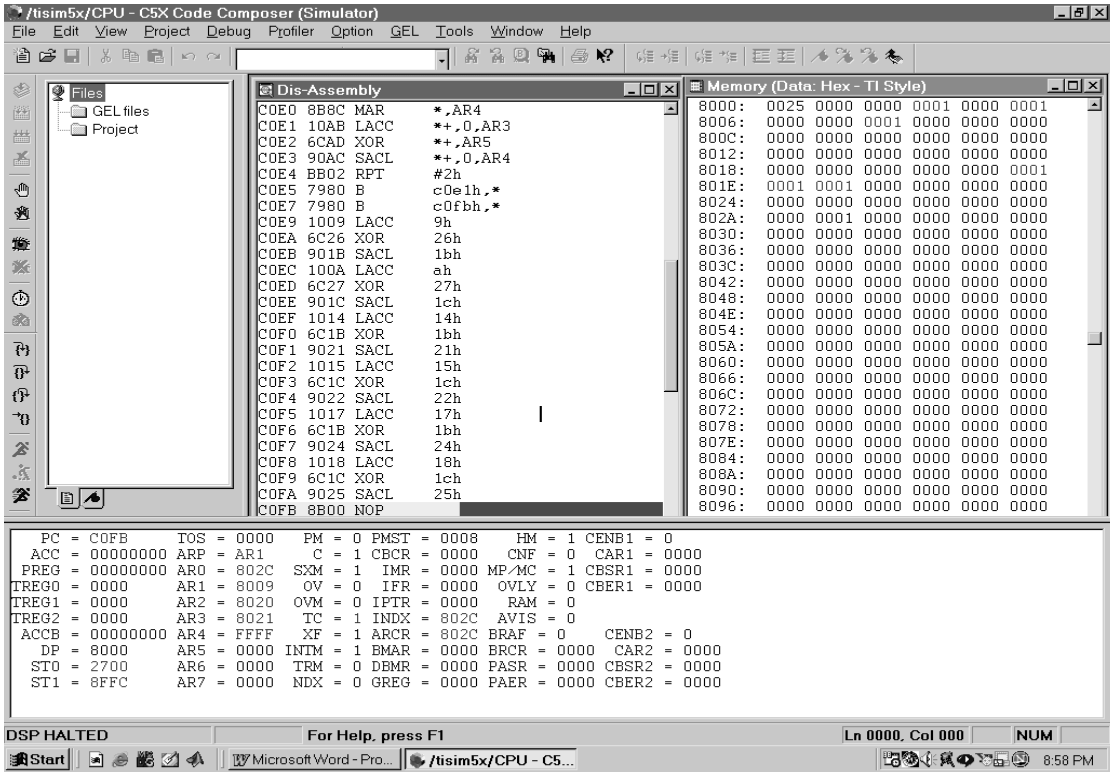


```
C000        .text:
C000 BD00 LDP     #100h
C001 8B88 MAR     *,AR0
C002 BF08 LAR     AR0,#802ch
C004 5F00 CPL     #0h,*
C006 E100 BCND    c014h,TC
C008 5F00 CPL     #1h,*
C00A E100 BCND    c02eh,TC
C00C 5F00 CPL     #2h,*
C00E E100 BCND    c048h,TC
C010 5F00 CPL     #3h,*
C012 E100 BCND    c062h,TC
C014 6A00 LACC    0h,16
C015 BE81 AND     #1h,16
C017 9803 SACH    3h
C018 1F00 LACC    0h,15
C019 BE81 AND     #1h,16
C01B 9804 SACH    4h
C01C 1E00 LACC    0h,14
C01D BE81 AND     #1h,16
C01F 9805 SACH    5h
C020 1D00 LACC    0h,13
C021 BE81 AND     #1h,16
C023 9806 SACH    6h
C024 1C00 LACC    0h,12
C025 BE81 AND     #1h,16
C027 9807 SACH    7h
C028 1B00 LACC    0h,11
C029 BE81 AND     #1h,16
C02B 9808 SACH    8h
```

```
8000:  0025 0000 0030 0000 0000 0000
8006:  00C0 0000 0030 0000 0000 0000
800C:  00C0 0000 0030 0000 0000 0000
8012:  00C0 0000 0030 0000 0000 0000
8018:  00C0 0000 0030 0000 0000 0000
801E:  00C0 0000 0030 0000 0000 0000
8024:  00C0 0000 0030 0000 0000 0000
802A:  00C0 0001 0030 0000 0000 0000
8030:  00C0 0000 0030 0000 0000 0000
8036:  00C0 0000 0030 0000 0000 0000
803C:  00C0 0000 0030 0000 0000 0000
8042:  00C0 0000 0030 0000 0000 0000
8048:  00C0 0000 0030 0000 0000 0000
804E:  00C0 0000 0030 0000 0000 0000
8054:  00C0 0000 0030 0000 0000 0000
805A:  00C0 0000 0030 0000 0000 0000
8060:  00C0 0000 0030 0000 0000 0000
8066:  00C0 0000 0030 0000 0000 0000
806C:  00C0 0000 0030 0000 0000 0000
8072:  00C0 0000 0030 0000 0000 0000
8078:  00C0 0000 0030 0000 0000 0000
807E:  00C0 0000 0030 0000 0000 0000
8084:  00C0 0000 0030 0000 0000 0000
808A:  00C0 0000 0030 0000 0000 0000
8090:  00C0 0000 0030 0000 0000 0000
8096:  00C0 0000 0030 0000 0000 0000
809C:  00C0 0000 0030 0000 0000 0000
80A2:  00C0 0000 0030 0000 0000 0000
80A8:  00C0 0000 0030 0000 0000 0000
80AE:  00C0 0000 0030 0000 0000 0000
```

```
   PC = C000      TOS = C000    PM = 0 PMST = 0C00 INDX = 0000    HM = 1 BRAF = 0    CENB1 = 0    CENB2 = 0
  ACC = 00000000  ARP = AR0      C = 1 CBCR = 0C00 ARCR = 0000   CNF = 0 BRCR = 0000  CAR1 = 0000  CAR2 = 0000
 PREG = 00000000  AR0 = C000   SXM = 1  IMR = 0C00 BMAR = 0000 MP/MC = 1 PASR = 0000 CBSR1 = 0000 CBSR2 = 0000
TREG0 = 0000      AR1 = C000    OV = 0  IFR = 0C00 DBMR = 0000  OVLY = 0 PAER = 0000 CBER1 = 0000 CBER2 = 0000
TREG1 = 0000      AR2 = C000   OVM = 0 IPTR = 0C00 GREG = 0000   RAM = 0
TREG2 = 0000      AR3 = C000    TC = 0                          AVIS = 0
 ACCB = 00000000  AR4 = C000    XF = 1
```

Figure 6: Simulator status before execution of spectral shaping program for $S_r = 0$



```
C0E0 8B8C MAR     *,AR4
C0E1 10AB LACC    *+,0,AR3
C0E2 6CAD XOR     *+,AR5
C0E3 90AC SACL    *+,0,AR4
C0E4 BB02 RPT     #2h
C0E5 7980 B       c0e1h,*
C0E7 7980 B       c0fbh,*
C0E9 1009 LACC    9h
C0EA 6C26 XOR     26h
C0EB 901B SACL    1bh
C0EC 100A LACC    ah
C0ED 6C27 XOR     27h
C0EE 901C SACL    1ch
C0EF 1014 LACC    14h
C0F0 6C1B XOR     1bh
C0F1 9021 SACL    21h
C0F2 1015 LACC    15h
C0F3 6C1C XOR     1ch
C0F4 9022 SACL    22h
C0F5 1017 LACC    17h
C0F6 6C1B XOR     1bh
C0F7 9024 SACL    24h
C0F8 1018 LACC    18h
C0F9 6C1C XOR     1ch
C0FA 9025 SACL    25h
C0FB 8B00 NOP
```

```
8000:  0025 0000 0000 0001 0000 0001
8006:  0000 0000 0001 0000 0000 0000
800C:  0000 0000 0000 0000 0000 0000
8012:  0000 0000 0000 0000 0000 0000
8018:  0000 0000 0000 0000 0000 0001
801E:  0001 0001 0000 0000 0000 0000
8024:  0000 0000 0000 0000 0000 0000
802A:  0000 0001 0000 0000 0000 0000
8030:  0000 0000 0000 0000 0000 0000
8036:  0000 0000 0000 0000 0000 0000
803C:  0000 0000 0000 0000 0000 0000
8042:  0000 0000 0000 0000 0000 0000
8048:  0000 0000 0000 0000 0000 0000
804E:  0000 0000 0000 0000 0000 0000
8054:  0000 0000 0000 0000 0000 0000
805A:  0000 0000 0000 0000 0000 0000
8060:  0000 0000 0000 0000 0000 0000
8066:  0000 0000 0000 0000 0000 0000
806C:  0000 0000 0000 0000 0000 0000
8072:  0000 0000 0000 0000 0000 0000
8078:  0000 0000 0000 0000 0000 0000
807E:  0000 0000 0000 0000 0000 0000
8084:  0000 0000 0000 0000 0000 0000
808A:  0000 0000 0000 0000 0000 0000
8090:  0000 0000 0000 0000 0000 0000
8096:  0000 0000 0000 0000 0000 0000
```

```
   PC = C0FB      TOS = 0000    PM = 0 PMST = 0008    HM = 1 CENB1 = 0
  ACC = 00000000  ARP = AR1      C = 1 CBCR = 0000   CNF = 0  CAR1 = 0000
 PREG = 00000000  AR0 = 802C   SXM = 1  IMR = 0000 MP/MC = 1 CBSR1 = 0000
TREG0 = 0000      AR1 = 8009    OV = 0  IFR = 0000  OVLY = 0 CBER1 = 0000
TREG1 = 0000      AR2 = 8020   OVM = 0 IPTR = 0000   RAM = 0
TREG2 = 0000      AR3 = 8021    TC = 1 INDX = 802C  AVIS = 0
 ACCB = 00000000  AR4 = FFFF    XF = 1 ARCR = 802C BRAF = 0    CENB2 = 0
   DP = 8000      AR5 = 0000  INTM = 1 BMAR = 0000 BRCR = 0000  CAR2 = 0000
  ST0 = 2700      AR6 = 0000   TRM = 0 DBMR = 0000 PASR = 0000 CBSR2 = 0000
  ST1 = 8FFC      AR7 = 0000   NDX = 0 GREG = 0000 PAER = 0000 CBER2 = 0000
```

Figure 7: Simulator status after execution of spectral shaping program for $S_r = 0$

encoder have been stored at 'dma' 801BH to 8020H which again exactly matches with the theoretically predicted results as per Eq. (5) and hence confirm the successfulness of assembly program corresponding to option S_r=0.

To further confirm the successfulness of developed algorithm and corresponding assembly level program, another case for S_r=3 is considered here. In this case three data frames are required so output of bit parser corresponding to data frame j(000BH), j+1(0002H) and j+2 (0005H) has been stored at 'dma' 8000H, 8001H and 8002H respectively. Another initialization includes S_r = 3 ('dma' 802CH), previous odd bit differential encoding output in data frame interval 1 of data frame j-1 (i.e. $P'_{j-1}(1) = 1$ stored at 'dma' 801AH) and previous outputs of second order differential encoding in data interval 0 & 1 corresponding to data frame j-1 (i.e. $t_{j-1}(0)=0$ at 'dma' 8026H and $t_{j-1}(1) = 1$ stored at 'dma' 8027H). With above initialization the status of the simulator is shown in Fig. 8 and simulation results after debugging and executing the program have been presented in the Fig. 9. Parse to shaping frame results have been stored at 'dma' 8003H to 8008H and results of odd bit differential encoding along with second order differential encoding have been stored at 'dma' 8009H-800AH, 8014h-8015H, 8017-8018H and 801BH-801CH, 8021H-8022H, 8024H-8025H respectively which are in accordance with the theoretically predicted results. Similarly program was simulated for other options S_r=1 and S_r=2 with different outputs of bit parser and absolute performance have been achieved.

7 IMPLEMENTATION OF PARSE - TO SHAPING - FRAME AND DIFFERENTIAL ENCODER ON TMS320C50PQ57 DSP CHIP

Program for parse - to - shaping - frame and differential encoder to perform spectral shaping has been loaded into the DSP Module for its practical implementation using communication software program XTALK provided by VI Microsystems Pvt. Ltd. Same inputs and initialization parameters as used during simulation have been taken again here and it has been observed that practical results are in accordance with the simulated or theoretically predicted results which confirms the successfulness of the present study. Various implementation parameters regarding present implementation of parse to shaping frame and differential encoding functions to perform spectral shaping in the transmitter of 56Kbps digital modem have been given in Table 2.

Table 2: Summery of implementation parameters

Function	Data Memory Used (W)	Program Memory Used (W)	Program Execution Time (µs)
Parse to Shaping Frame and Differential Encoding	44	251	Depends upon the value of S_r (1.123 – 1.368)

8 CONCLUSION

Simulation and implementation of functional units such as parse - to - shaping - frame and differential encoder etc. of spectral shaping block of 56Kbps digital modem transmitter have been discussed in the present paper. A combined algorithm for implementation of the parse to shaping frame and differential encoder functions utilized in transmitter of 56Kbps digital modem has been suggested. Parse - to - shaping - frame function has been implemented with the help of bit masking technique where as differential encoder has been implemented with the help of XOR functioning and delay line implementation. Assembly level program corresponding to algorithm developed during present study has been simulated and loaded into the DSP module to implement parse - to - shaping - frame and differential encoding process. Practical results obtained have been found to be same as of theoretical and simulated ones.

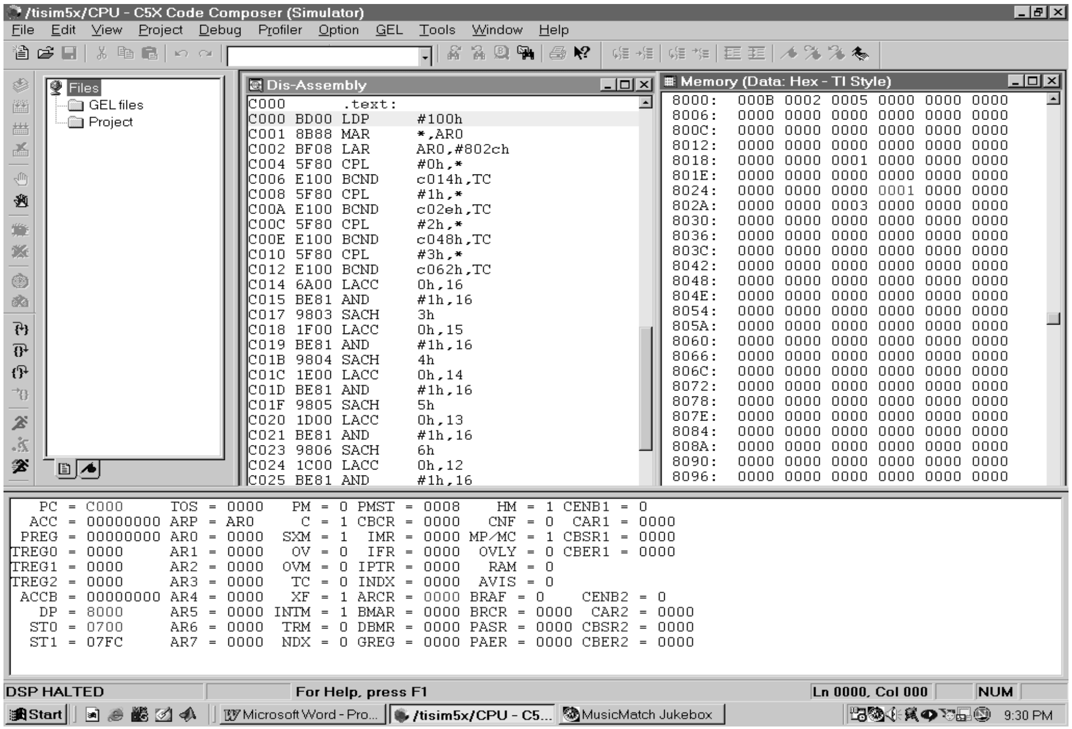
ACKNOWLEDGEMENTS

One of the authors Davinder Pal Sharma is thankful to Guru Nanak Dev University, Amritsar, for providing Research facilities at Department of Electronics Technology for the present research work.

10 REFERENCES

[1] A. A. Gokhale: Introduction to Telecommunication, Thomson Asia Pvt. Ltd. (2001).

[2] D. E. A. Clarke et. al: Emerging Broadband

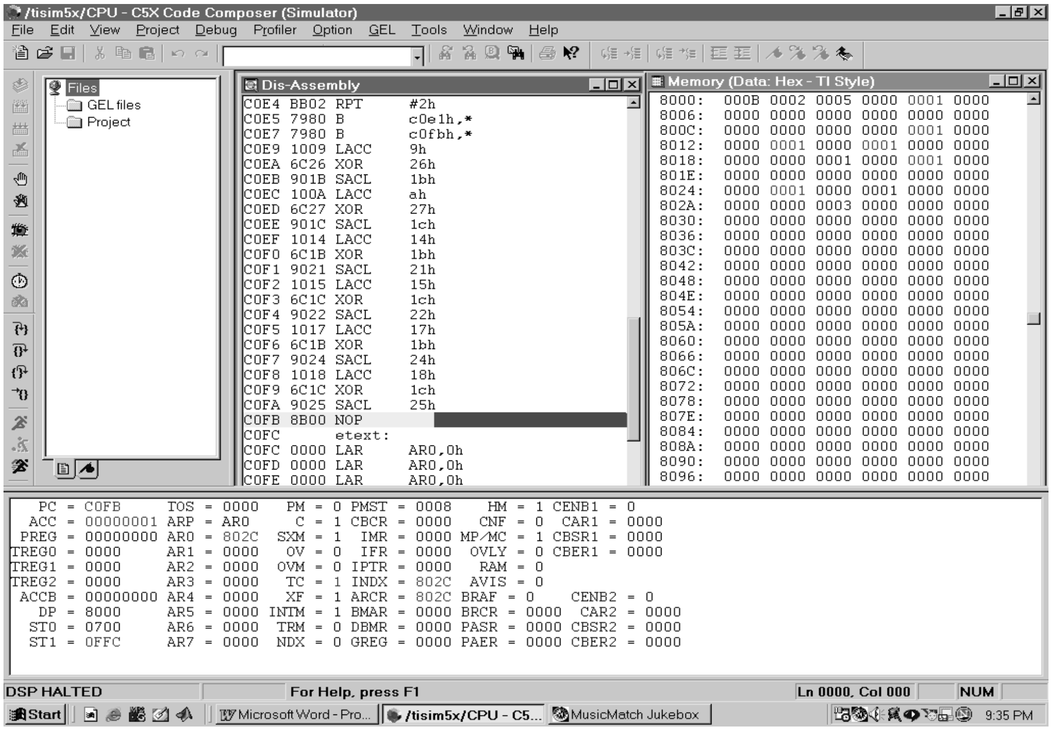


```
C000       .text:
C000 BD00 LDP      #100h
C001 8B88 MAR      *,AR0
C002 BF08 LAR      AR0,#802ch
C004 5F80 CPL      #0h,*
C006 E100 BCND     c014h,TC
C008 5F80 CPL      #1h,*
C00A E100 BCND     c02eh,TC
C00C 5F80 CPL      #2h,*
C00E E100 BCND     c048h,TC
C010 5F80 CPL      #3h,*
C012 E100 BCND     c062h,TC
C014 6A00 LACC     0h,16
C015 BE81 AND      #1h,16
C017 9803 SACH     3h
C018 1F00 LACC     0h,15
C019 BE81 AND      #1h,16
C01B 9804 SACH     4h
C01C 1E00 LACC     0h,14
C01D BE81 AND      #1h,16
C01F 9805 SACH     5h
C020 1D00 LACC     0h,13
C021 BE81 AND      #1h,16
C023 9806 SACH     6h
C024 1C00 LACC     0h,12
C025 BE81 AND      #1h,16
```

```
8000:  000B 0002 0005 0000 0000 0000
8006:  0000 0000 0000 0000 0000 0000
800C:  0000 0000 0000 0000 0000 0000
8012:  0000 0000 0000 0000 0000 0000
8018:  0000 0000 0001 0000 0000 0000
801E:  0000 0000 0000 0000 0000 0000
8024:  0000 0000 0000 0001 0000 0000
802A:  0000 0000 0003 0000 0000 0000
8030:  0000 0000 0000 0000 0000 0000
8036:  0000 0000 0000 0000 0000 0000
803C:  0000 0000 0000 0000 0000 0000
8042:  0000 0000 0000 0000 0000 0000
8048:  0000 0000 0000 0000 0000 0000
804E:  0000 0000 0000 0000 0000 0000
8054:  0000 0000 0000 0000 0000 0000
805A:  0000 0000 0000 0000 0000 0000
8060:  0000 0000 0000 0000 0000 0000
8066:  0000 0000 0000 0000 0000 0000
806C:  0000 0000 0000 0000 0000 0000
8072:  0000 0000 0000 0000 0000 0000
8078:  0000 0000 0000 0000 0000 0000
807E:  0000 0000 0000 0000 0000 0000
8084:  0000 0000 0000 0000 0000 0000
808A:  0000 0000 0000 0000 0000 0000
8090:  0000 0000 0000 0000 0000 0000
8096:  0000 0000 0000 0000 0000 0000
```

```
   PC = C000      TOS = 0000   PM = 0 PMST = 0008    HM = 1 CENB1 = 0
  ACC = 00000000 ARP = AR0     C = 1 CBCR = 0000   CNF = 0  CAR1 = 0000
 PREG = 00000000 AR0 = 0000  SXM = 1  IMR = 0000 MP/MC = 1 CBSR1 = 0000
TREG0 = 0000      AR1 = 0000   OV = 0  IFR = 0000  OVLY = 0 CBER1 = 0000
TREG1 = 0000      AR2 = 0000  OVM = 0 IPTR = 0000   RAM = 0
TREG2 = 0000      AR3 = 0000   TC = 0 INDX = 0000  AVIS = 0
 ACCB = 00000000 AR4 = 0000   XF = 1 ARCR = 0000 BRAF = 0    CENB2 = 0
   DP = 8000      AR5 = 0000 INTM = 1 BMAR = 0000 BRCR = 0000  CAR2 = 0000
  ST0 = 0700      AR6 = 0000  TRM = 0 DBMR = 0000 PASR = 0000 CBSR2 = 0000
  ST1 = 07FC      AR7 = 0000  NDX = 0 GREG = 0000 PAER = 0000 CBER2 = 0000
```

Figure 8: Simulator status before execution of spectral shaping program for S_r=3



```
C0E4 BB02 RPT      #2h
C0E5 7980 B        c0e1h,*
C0E7 7980 B        c0fbh,*
C0E9 1009 LACC     9h
C0EA 6C26 XOR      26h
C0EB 901B SACL     1bh
C0EC 100A LACC     ah
C0ED 6C27 XOR      27h
C0EE 901C SACL     1ch
C0EF 1014 LACC     14h
C0F0 6C1B XOR      1bh
C0F1 9021 SACL     21h
C0F2 1015 LACC     15h
C0F3 6C1C XOR      1ch
C0F4 9022 SACL     22h
C0F5 1017 LACC     17h
C0F6 6C1B XOR      1bh
C0F7 9024 SACL     24h
C0F8 1018 LACC     18h
C0F9 6C1C XOR      1ch
C0FA 9025 SACL     25h
C0FB 8B00 NOP
C0FC       etext:
C0FC 0000 LAR      AR0,0h
C0FD 0000 LAR      AR0,0h
C0FE 0000 LAR      AR0,0h
```

```
8000:  000B 0002 0005 0000 0001 0000
8006:  0000 0000 0000 0000 0000 0000
800C:  0000 0000 0000 0000 0001 0000
8012:  0000 0001 0000 0001 0000 0000
8018:  0000 0000 0001 0000 0001 0000
801E:  0000 0000 0000 0000 0000 0000
8024:  0000 0001 0000 0001 0000 0000
802A:  0000 0000 0003 0000 0000 0000
8030:  0000 0000 0000 0000 0000 0000
8036:  0000 0000 0000 0000 0000 0000
803C:  0000 0000 0000 0000 0000 0000
8042:  0000 0000 0000 0000 0000 0000
8048:  0000 0000 0000 0000 0000 0000
804E:  0000 0000 0000 0000 0000 0000
8054:  0000 0000 0000 0000 0000 0000
805A:  0000 0000 0000 0000 0000 0000
8060:  0000 0000 0000 0000 0000 0000
8066:  0000 0000 0000 0000 0000 0000
806C:  0000 0000 0000 0000 0000 0000
8072:  0000 0000 0000 0000 0000 0000
8078:  0000 0000 0000 0000 0000 0000
807E:  0000 0000 0000 0000 0000 0000
8084:  0000 0000 0000 0000 0000 0000
808A:  0000 0000 0000 0000 0000 0000
8090:  0000 0000 0000 0000 0000 0000
8096:  0000 0000 0000 0000 0000 0000
```

```
   PC = C0FB      TOS = 0000   PM = 0 PMST = 0008    HM = 1 CENB1 = 0
  ACC = 00000001 ARP = AR0     C = 1 CBCR = 0000   CNF = 0  CAR1 = 0000
 PREG = 00000000 AR0 = 802C  SXM = 1  IMR = 0000 MP/MC = 1 CBSR1 = 0000
TREG0 = 0000      AR1 = 0000   OV = 0  IFR = 0000  OVLY = 0 CBER1 = 0000
TREG1 = 0000      AR2 = 0000  OVM = 0 IPTR = 0000   RAM = 0
TREG2 = 0000      AR3 = 0000   TC = 1 INDX = 802C  AVIS = 0
 ACCB = 00000000 AR4 = 0000   XF = 1 ARCR = 802C BRAF = 0    CENB2 = 0
   DP = 8000      AR5 = 0000 INTM = 1 BMAR = 0000 BRCR = 0000  CAR2 = 0000
  ST0 = 0700      AR6 = 0000  TRM = 0 DBMR = 0000 PASR = 0000 CBSR2 = 0000
  ST1 = 0FFC      AR7 = 0000  NDX = 0 GREG = 0000 PAER = 0000 CBER2 = 0000
```

Figure 9: Simulator status after execution of spectral shaping program for S_r=3

Access Technologies, British Telecommunication Technology Journal, Vol. 16, pp. 187-195 (Oct. 1998).

[3] V. U. Reddy: Voice-band Modems: A Device to Transmit Data over Telephone Networks - Part (I): Basic Principles of Data Transmission, Resonance, Vol. 6, pp. 56-65 (May 2001).

[4] V. U. Reddy: Voice-Band Modems: A Device to Transmit Data over Telephone Networks - Part (II): Advanced Ideas which made High Data Rates possible, Resonance, Vol. 6, pp. 60-70 (June 2001).

[5] F. Adams et. al: Today's Access Technologies, British Telecommunication Technology Journal, Vol. 16, pp. 21-33 (Oct. 1998).

[6] Georgia Tech Studies: (1998). http://www.gvu.gatech.edu/user_surveys/survey-1998-10/graphs/technology/q01.htm.

[7] ITU-T Recommendation V.90: A Digital and Analog Modem pair for use on the PSTN at Data Signaling Rates of up to 56000 bits/s Downstream and 33600 bits/s Upstream, ITU-T V Series Recommendations (1998).

[8] Recommendation G.711: Pulse Coded Modulation (PCM) of Voice Frequencies, ITU-T G Series Recommendations (1988).

[9] L. Brown: PCM Modem Design: V.90 Characteristics, Communication System Design Magazine, Vol. 6 (1998).

[10] S. A. Tretter: Constellation Shaping, Nonlinear Precoding and Trellis Coding for Voice band Telephone Channel Modems, Kluwer Academic Publishers (2002).

[11] H. Kobyashi: A Survey of Coding Schemes for Transmission or Recording of Digital Data, IEEE Trans. Comm. Tech., Vol. COM-19, pp 1067-1100 (Dec. 1971).

[12] A. Croiser: Introduction to Pseudoternary Transmission Codes, IBM Journal of Research & Development, Vol. 14, pp. 354-367 (July 1970).

[13] E. Gorog: Redundant Alphabets with Desirable Frequency Spectrum Properties, IBM Journal of Research & Development, Vol. 12, pp 234-241 (May 1968).

[14] H. Waldman, C. A. M. Pingarilho: Spectral Shaping Codes” Proc. of IEEE ISIT, pp 209 (June, 1994).

[15] H. Waldman, C.A.M. Pingarilho: Coding for Spectral Shaping, Proc. of Global Telecommunications Conference, pp. 132-135 (Nov. 1994).

[16] A. K. Khandani,: Constellation Shaping as a Geometrical Approach to Solving a Constrained Optimization Problem, Proc. of IEEE ISIT, pp. 330 (June 1994).

[17] M. Campanella, G. Garbo, G. Mamola: A Design Technique for Spectral Shaping in CPM Systems, IEEE Transactions on Communications, Vol. 45 (May 1997).

[18] J.K. Cavers, R.F. Marchetto: A New Coding Technique for Spectral Shaping of Data, IEEE Transactions on Communications, Vol. 40 (Sept. 1992).

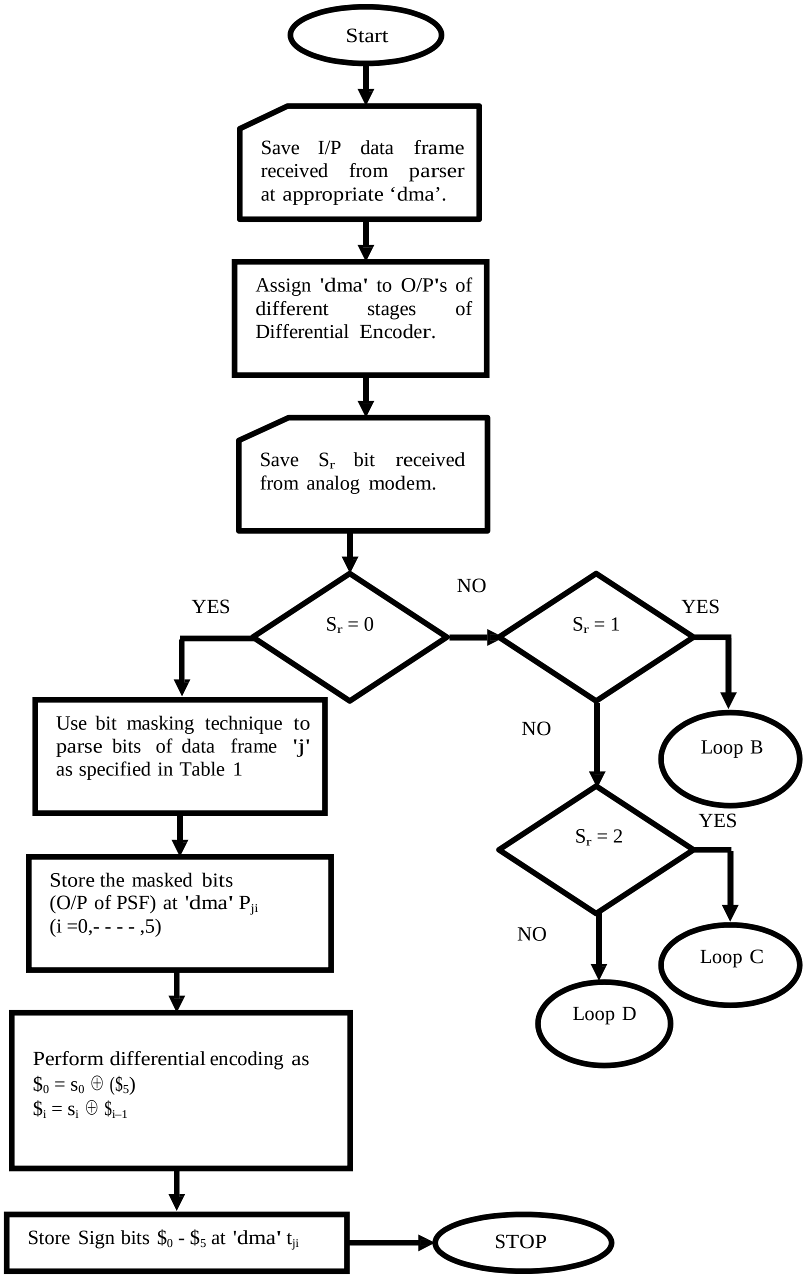
[19] J. G. Proakis, D.G. Manolakis: Digital Communication, McGraw Hill Inc. (2000).

[20] H. Shankar, P. V. Wagt: Implementing a High-Speed Differential Encoder, Inphi Corporation, White Paper.

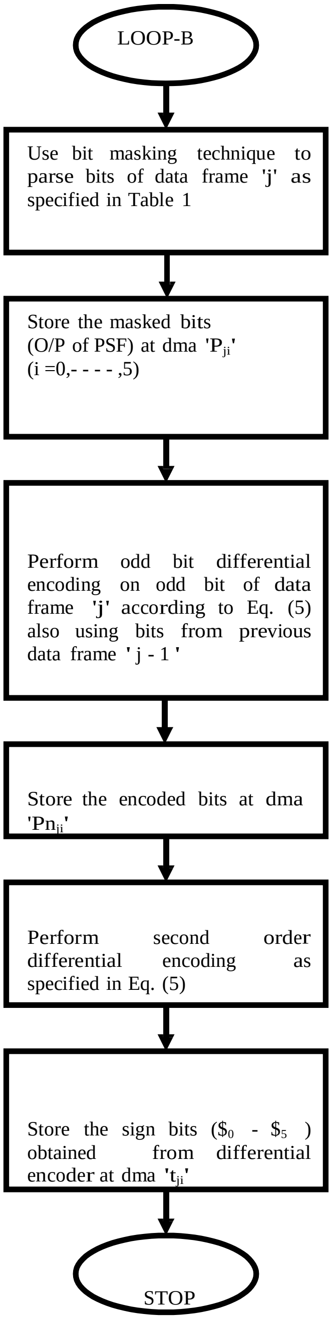
[21] L. M. Caraballo: System Level Design and Simulation of a PCM Voiceband Modem Compliant with the ITU V.90 Standard, Texas A & M University, M. S. Thesis (May 2000).

APPENDIX – I

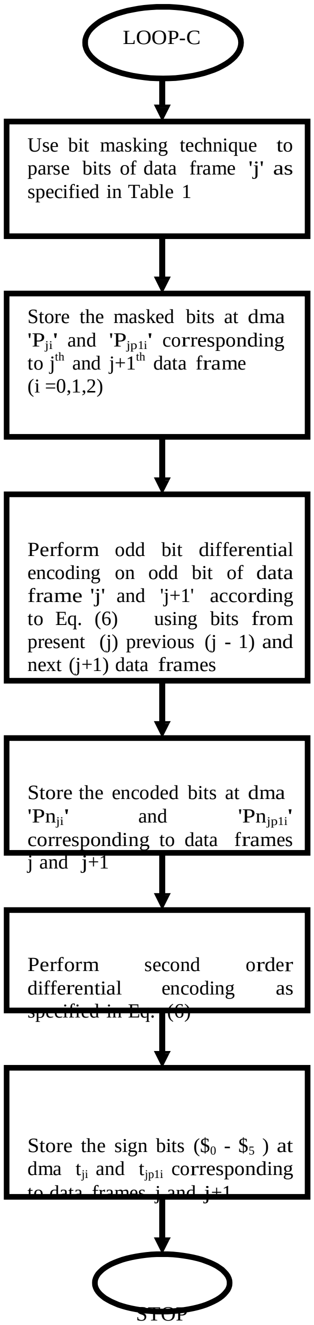
Main algorithm to perform spectral shaping in 56Kbps digital modem transmitter



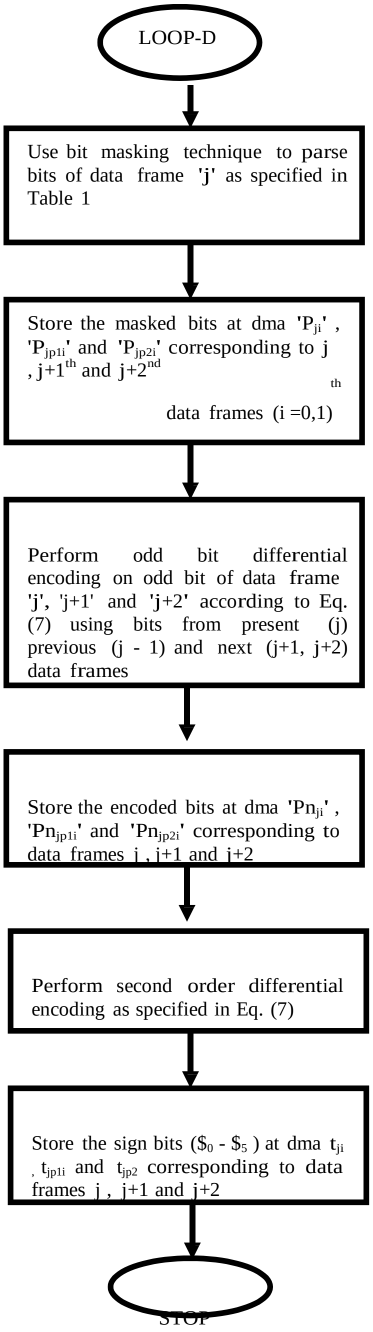
APPENDIX – I (contd…..)



(a) Subroutine to perform spectral shaping corresponding to S_r=1 and S=5



(b) Subroutine to perform spectral shaping corresponding to S_r=2 and S=4



(c) Subroutine to perform spectral shaping corresponding to S_r=3 and S=3