An Efficient FPGA Based Implementation of Multichannel Satellite Modulator

K. K. Vinaymurthi, K. M. Nitin Babu, K. V. Sachin and S. V. Hariprasad

Abstract. This paper discusses the design and successful implementation of a packet based Multichannel satellite modulator using efficient, novel Data buffering technique. The modulator receives data from IP network and transmits it to satellite network. The scheme allows for loss-less transfer of data from Best effort IP based packet network to a circuit switched fixed bit rate narrow band continuous satellite modulator. The multichannel scalable modulator is implemented in XilinxVirtex-6 FPGA. The IP packet processed by TI C6678 DSP reaches FPGA via EMIF interface as RAW chunk of data of multiple channels, which are further, processed and buffered using a novel low latency scheme. The buffered data is serialized and modulated. The modulation chain incorporates [1] INTELSATV.35 Scrambler, Differential encoder, 1/2 rate convolution encoder, RRC filtering and employs BPSK modulation.

I INTRODUCTION

Satellite communications is an integral part telecommunications in today's world. The applications range from television broadcast to military communications. A typical satcom network consists of mainly three components, Remote nodes in the field, Geosynchronous satellites and Earth stations. Depending on the application, communications can be simplex or duplex. The remote nodes deployed along the ground send and receive messages to the satellite sitting in Geosynchronous orbit. The satellite in turn sends and receives the data to and from the earth station.

The earth station can be connected to various other networks via corresponding gateways. Incorporating IP packet processing capabilities into the modulator allows seamless connectivity to external networks. In contrast, a continuous constant rate modulator is preferred for uplink communications as it helps to bring down receiver complexity. Hence the challenge herein lies on how to interwork between the packet based IP network, which is inherently bursty in nature and the fixed bit rate satellite link.

The authors are with C-DOT Bangalore, Hosur Road , Electronics city ph1 ,Bangalore, Karnataka, INDIA nitinkm@cdot.in

II. DESIGN FLOW

II.a SYSTEM OVERVIEW

The system can be mainly divided into three blocks, Packet processing, Data Buffering and Base-band processing as shown in Fig. 1. The packet processing block is responsible for receiving and processing incoming packets, and sending request packets for next data packet. The buffering block mediates the flow of data from packet processing block to the modulation block. It also generates the trigger event for sending request packet. The Base-band block is responsible for performing channel coding and BPSK modulation on the received data.

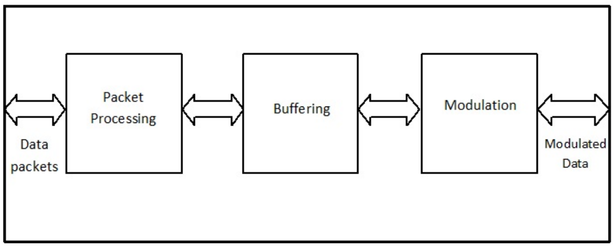


Fig. 1. Top level block diagram.

IIA.1 PACKET PROCESSING

All the packet related processing is done by this block shown in Fig. 2. It sends out request packets and receives data packets from the network. The Ethernet MAC module maintains the physical link and implements up to layer2 over IP protocol stack. The upper layers are processed by the IP/UDP block implemented in software. The data is extracted from the packet and send to buffering block. On top of UDP a proprietary application layer implemented for identifying the data corresponding to different channels. The channel number parameter is included in the packet header for this.

II.a.2 BUFFERING

The buffering block as shown in Fig. 3 acts as the mediator between Packet processing and Base-band Modulator blocks. It buffers the data coming from Packet processing block, serializes it and sends to base-band modulator. There are two buffering stages, data from packet processing block first gets loaded to Buffer I initially. Data loaded in Buffer II is serialized and sent to base-band modulator continuously in linear fashion. When buffer II reading reaches a threshold value near the end, the feedback control generates a trigger to packet processing block resulting in the generation of request packet, which in turn reduces the latency. Reading and writing of buffers is controlled by buffer control module. Once Buffer I is filled and initial half of Buffer II is read, half the data gets transferred from buffer I to Buffer II. Then as the reading of second half is completed, latter half gets transferred. This ensures that data flows to the modulator continuously and there aren’t any read-write collisions reading to corruption of data.

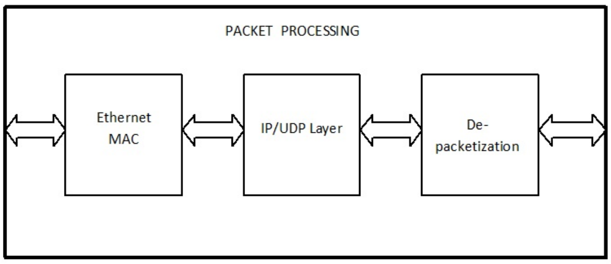


Fig. 2. Packet Processing module.

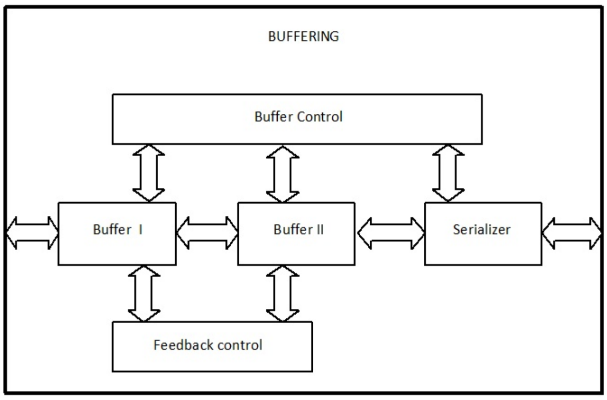


Fig. 3. Buffering module.

II.b.3 BASE-BAND PROCESSING

Multi-channel Baseband modulator receives serial data stream from Buffering block and performs channel coding and modulation on it. The block includes, scrambler, differential encoder, Convolutional encoder, BPSK modulator and RRC filter. The INTELSAT V.35 Scrambler is based on the ITU-T standard for telecommunications is the first module in the modulation chain as in Fig. 4.The scrambled data is differentially encoded to protect from bit reversal during transmission. Data is encoded by (2,1,6) convolutional encoder effectively doubling the bit rate. The encoded bits are mapped to BPSK symbols and up sampled by factor 6 and passed through SRRC filter with roll off factor $\alpha = 0.4$.

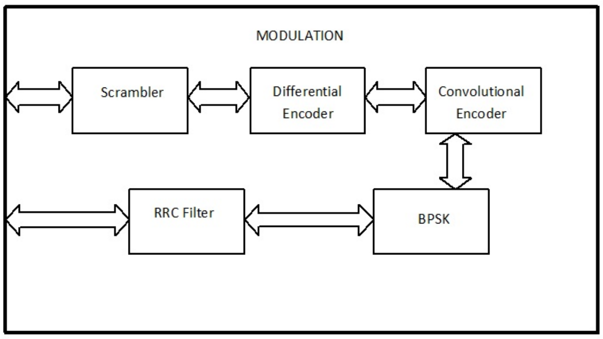


Fig. 4. Baseband modulation module.

III. MATHEMATICAL MODELLING

IP network offers best effort service, so there is no guarantee on the time of arrival of packets [2].The arrival time can vary widely and packets can get dropped in the network leading to packet loss. Therefore the process of arrival of packets can be modelled as a Poisson random process [3].

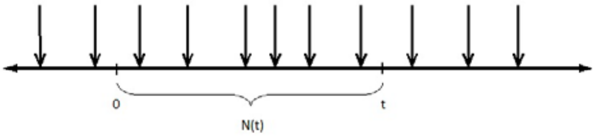


Fig. 5. Poisson arrival process

Let N(t) be the random variable denoting the number of arrivals in time interval (0; t) and λ be the mean number of arrivals in unit time. Then the probability of N(t) = n, P(N(t) = n) is given by

$$P(N(t) = n) = (\lambda t)^n e^{-\lambda} \qquad (1)$$

Considering that packets are arriving continuously, the mean arrivals can be related to the link bandwidth and the packet size. Let R bits/sec be the link bandwidth and L bits be the size of the packet. Then

$$\lambda = \frac{R}{L} \qquad (2)$$

The inter-arrival time between the packets follows exponential distribution. If T be the time interval between arrivals, then the probability that T > t is given by

$$P(T > t) = e^{-\lambda t} \qquad (3)$$

The process of receiving data involves handshaking, where in a request packet is send from the modulator to the information source and data packets are received in response to it. The total waiting time (T_w) is the sum of arrival time of request packet to information source (T_r) and the arrival time of data packet(s) (T_d). The small packet processing time can be neglected.

$$T_w = T_r + T_d \qquad (4)$$

Considering (T_r) and (T_d) to be independent the probability that the total waiting time $T_w > t_w$ is given by,

$$P(T_w > t_w = t_r + t_d) = e^{-\lambda(tr+td)} \qquad (5)$$

For multi-channel implementation with k channels, (T_d) will be sum of inter-arrival times of packets corresponding to each channel (T_{di}).

$$T_d = \sum_{i=1}^{k} T_{di} \qquad (6)$$

Assuming t_{di} are iid (identical and independently distributed)the probability $P(Td > td)$ can be represented as,

$$P(T_d > t_d) = P(T_{di} > t_{di}) = P(T_{di} > t_d /k) \qquad (7)$$

Hence for multi-channel case (5) becomes,

$$P(T_w > t_w) = e^{-\lambda t_r} e^{-\lambda t_d} \qquad (8)$$

From (2) it can be referred that λ = R/L and hence (8) becomes.

$$P(T_w > t_w) = e^{-\frac{R(kt_r + t_d)}{Lk}} \qquad (9)$$

The Buffer is designed as shown in Fig. 4. The width (W) and Depth (D) of the buffer are related to the packet length (L) as L = D x W. The Buffer Stage II as in Fig. 6 has three thresholds

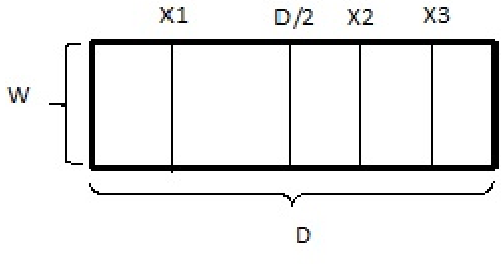


Fig. 6. Buffer Structure

X1, X2 and X3. The buffer reading rate (B_R) is related to the bit rate of the modulator (R_b) as.

$$B_R = \frac{R_b}{W} \qquad (10)$$

The thresholds X1, X2 and X3 are fixed such that when the read pointer reaches X2 a trigger is generated this causes the generation and transmission of request packet to information source. Data is expected to reach Buffer Stage I by the time the read pointer reaches X3. When X3 is reached, half of the data is copied from Stage I to Stage II at a rate (B_W). (B_W) is chosen such that,

$$\frac{D}{2B_W} < \frac{D - X3}{B_R} \qquad (11)$$

The above condition ensures that by the time read pointer reaches the end and goes back to initial point for next read cycle, Data gets updated in the buffer. Similarly when read pointer reaches X1, second half of the buffer gets filled. X1is fixed such that D-X3 = D/2-X1, Hence B_w from (11) holds. The packet has to arrive by the time the read pointer reaches X3 from X2. Hence based on (10) the waiting time available is related to modulator bit rate (R_b) as,

$$t_w = \frac{W(X3 - X2)}{R_b} \qquad (12)$$

Time taken by the request packet to reach information source (t_r) is almost same as the time taken for data packet to reach modulator from source td hence it is a good approximation to take $t_r = t_d = t_w /2$. With this condition (9) is reduced as (13) where in t_w is calculated from (12).

$$P(T_w > t_w) = e^{-\frac{(k+1)Rt_w}{2Lk}} \qquad (13)$$

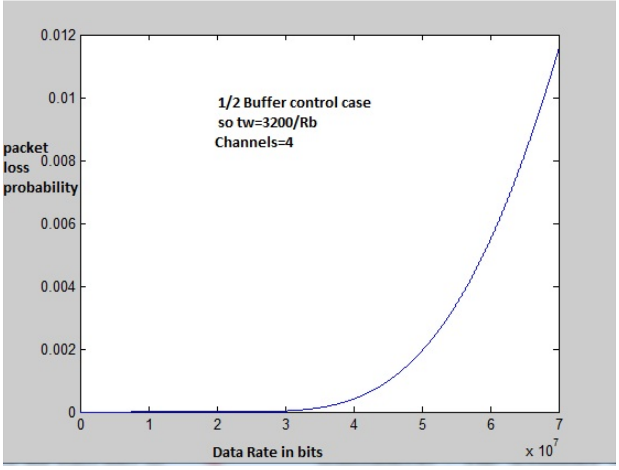


Fig. 7. Data rate vs packet loss probability plot for proposed scheme.

IV IMPLEMENTATION

The satellite modulator is implemented on Xilinx Virtex-6 family XC6VLX75T FPGA [8]. It has 74,496 logic cells and5,616 Kb of block RAMs, providing ample resource to implement the required logic. The packet processing block is implemented in TI 6678 DSP. The physical and Link layer processing are done by the integrated Ethernet MAC hardware in the DSP. This element implements up to layer 2 of IP protocol stack as defined in IEEE 802.3-2008specification. The upper layers namely network, transport and application are implemented in software that runs in DSP core. The DSP has a Packet Accelerator Subsystem (PASS) which is used to offload some of the workload allowing accelerated processing of packets. Additional protocols such as ARP (Address resolution Protocol) and ICMP (Internet Control Message Protocol) are also implemented to allow interfacing with other network elements. The data extracted from the packet is send to Virtex-6 FPGA through External Memory Interface (EMIF) between DSP and FPGA. Buffering and modulation is done in FPGA.

IV.a INTERMEDIATE BUFFERING

The buffers are implemented using the block RAMs available in the FPGA with necessary control logic written in VHDL, generated RTL as shown in Fig. 9. The buffer depth is determined by the packet size which is fixed at 800 Bytes. Hence for 4 channels, each channel requires 800 bytes buffer making total memory usage (800X4X2 = 6400Bytes) or 6.4KB.

The latency in the system is directly dependent on t_w defined in (12). The maximum latency with the buffering scheme is half frame duration with t_w maximum, a packet loss probability *vs.* bit rate plot is shown as in Fig. 7. But it can be further reduced to 0:1 times of frame duration or lesser if the buffer write rate is sufficiently higher by order of 10 and if small increase in packet loss probability can be accepted. Compared to the traditional Ping-Pong buffering scheme with buffer stage II divided into two full size buffers where reading and writing are alternated, the new scheme has lesser latency and improved ping performance as shown in Table I without much degradation in packet-loss probability.

IV.b BASEBAND RTL

All the modules are described in VHDL [4], simulated and functionally verified before porting to actual card. The filter is implemented in fixed point arithmetic so that it is hardware realizable [5]. The scrambler is implemented based on theV.35 specification approved by INTELSAT. The convolutional encoder [6] [7] has a rate of 1/2 and constraint length 7. The tap coefficients are (171,133) in octal format which is a widely used standard [8].

Table 1. Comparison between Ping-pong buffering and New scheme.

Scheme	Latency in frame duration	No. of Buffers
Ping-Pong	1	2
New scheme	0.1-0.5	2.

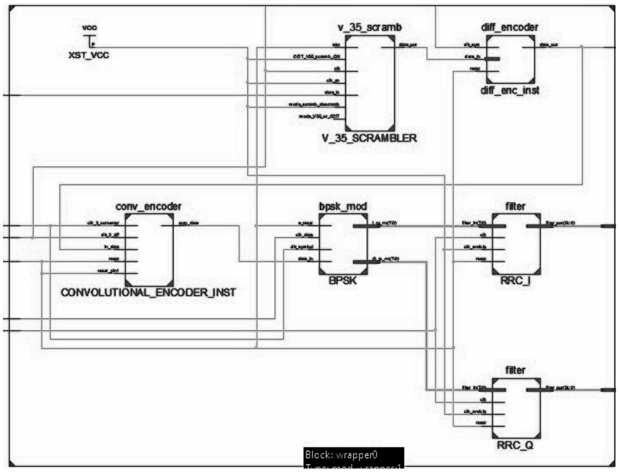


Fig. 8. Baseband modulation RTL.

Xilinx ISE Design Suite is used for code editing, simulation and implementation. Individual modules are synthesized and functionality is verified in the built-in isim simulator in ISE. Modules are implemented providing necessary timing constraints and timing is verified by performing timing analysis. Post placement and routing simulation is performed on the timing met design to verify functionality of the design after implementation. The modules are integrated and the above steps are repeated. Finally bitstream file (.bit) is generated and loaded to FPGA.

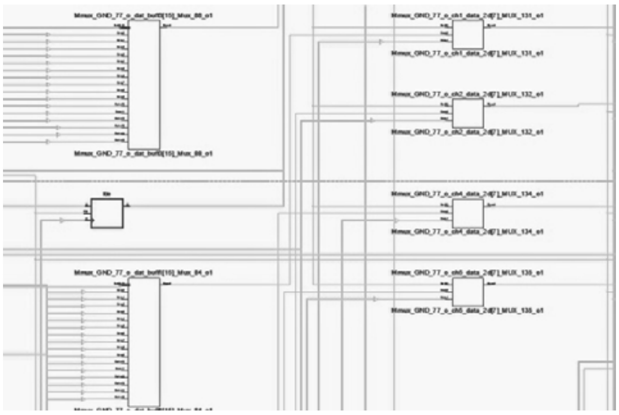


Fig. 9. Buffering module RTL.

V. VERIFICATION AND RESULTS

The last stage of Circuit verification is done directly in the FPGA itself using Integrated Logic Analyzer (ILA) IP core. ILA core is added to the design and implementation steps are repeated. The signals that need to be monitored are tapped by ILA, and these can be observed in PC using chipscope analyzer tool in ISE. The tool connects to FPGA via JTAG emulator and displays the internal signals real-time. After completing on hardware debugging and verification, the core is removed and the design is re-implemented. The final bitstream is loaded in flash by creating .mcs file and writing it into SPI flash connected to the FPGA using the impact tool in ISE.

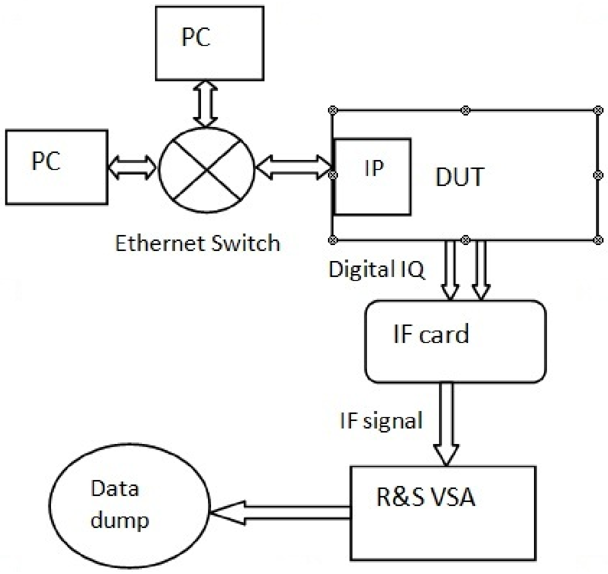
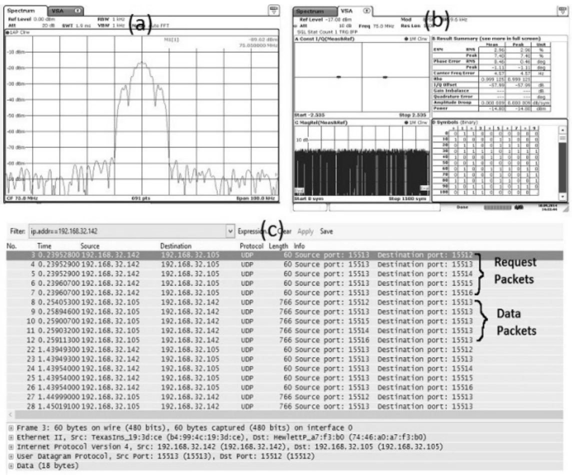


Fig. 10. Test setup.

System Performance analysis test setup is shown as in Fig.10, Data to be modulated comes from different PCs to same destination port of the DUT all connected through an Ethernet switch. Wireshark is used to analyze the packets sent through the network as in Fig. 11(c). After the base band processing of 4 channels the resultant IQ samples from Fig. 8 are sent for digital up-conversion. The up-converted samples are sent to IF card for converting to analog domain using a dual DAC with channels spaced accordingly so that adjacent channel interference is minimum. The IF signal centered at 4 different frequencies is further given to R&S Vector signal Analyzer for signal demodulation. The Analyzer checks modulated spectrum centered at 75MHz and correspondingly demodulated the signal and displays constellation, EVM and phase measurements all within tolerable limits. Decoded bits are cross verified with the data that has been sent, results are captured and verified for each channel separately as in Fig. 11(a)(b).


(a)
(b)
(c)
Filter: ip.addr==192.168.32.142
Expression... Clear Apply Save
Request Packets
Data Packets

```
No.  Time       Source          Destination     Protocol Length Info
   3 0.23952800 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15512
   4 0.23952900 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15513
   5 0.23952900 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15514
   6 0.23960700 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15515
   7 0.23960700 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15516
   8 0.25405300 192.168.32.105  192.168.32.142  UDP        766 Source port: 15512  Destination port: 15513
   9 0.25894600 192.168.32.105  192.168.32.142  UDP        766 Source port: 15513  Destination port: 15513
  10 0.25900700 192.168.32.105  192.168.32.142  UDP        766 Source port: 15515  Destination port: 15513
  11 0.25903200 192.168.32.105  192.168.32.142  UDP        766 Source port: 15514  Destination port: 15513
  12 0.25911300 192.168.32.105  192.168.32.142  UDP        766 Source port: 15516  Destination port: 15513
  22 1.43949300 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15512
  23 1.43949300 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15513
  24 1.43954000 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15514
  25 1.43954000 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15515
  26 1.43954000 192.168.32.142  192.168.32.105  UDP         60 Source port: 15513  Destination port: 15516
  27 1.44999000 192.168.32.105  192.168.32.142  UDP        766 Source port: 15512  Destination port: 15513
  28 1.45019100 192.168.32.105  192.168.32.142  UDP        766 Source port: 15513  Destination port: 15513

⊞ Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: TexasIns_19:3d:ce (b4:99:4c:19:3d:ce), Dst: HewlettP_a7:f3:b0 (74:46:a0:a7:f3:b0)
⊞ Internet Protocol Version 4, Src: 192.168.32.142 (192.168.32.142), Dst: 192.168.32.105 (192.168.32.105)
⊞ User Datagram Protocol, Src Port: 15513 (15513), Dst Port: 15512 (15512)
⊞ Data (18 bytes)
```

Fig. 11. Test results

VI. CONCLUSION & FUTURE WORK

A Multichannel Satellite modulator based on Intelsat earth station standard has been designed and implemented. The devised novel buffering technique allows virtually loss-less, Low latency data transfer from packet switched IP network to circuit switched Satellite network by minimizing packet loss probability. The design flow presented in this paper can also be used in any other variable delay to fixed delay system with bitrates varying from Kbps to Mbps based on packet loss Probability requirement. The 4 channel modulator circuitry occupies 2472 slices in Xilinx Virtex-6 FPGA which is around 21% of total logic available.

Current design makes use of DSP to process packets. In future implementation we plan to implement the packet processing in FPGA itself. It will make use of Ethernet MAC IP in Virtex-6 to implement Link and physical layers and a soft-core Microblaze processor to implement the software based upper layers. This allows the entire solution to be implemented in single FPGA.

We take this opportunity to thank our Project Director Mr.Soundarakumar M and Executive Director Mr.VipinTyagi for their valuable suggestions and support throughout the project and keeping the morale high.

REFERENCES

1 Intelsat IESS specification: Restricted access [Online].Available:http://www.intelsat.com/tools-resources/library/iess-documents/

2. James F. Kurose ,KeithRoss,*Computer Networking: A Top-Down Approach Featuring the Internet 2nd Addison-Wesley Longman Publishing* Co., Inc. Boston, MA, USA(2002)

3. Allen, A. O. *Probability, statistics, and queueing theory with computerscience applications*, 2nd ed. Academic Press, Inc., Boston, MA, (1990).

4. F. Scarpino, *VHDL and AHDL Digital System Implementation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall PTR, (1998)

5. NitinBabu K M and Vinaymurthi K K, *GMSK Modulator for GSMsystem an economical implementation on FPGA, ICCSP (2011),* DOI:10.1109/ICCSP.2011.5739302. pp. 208-212 5

6. H.L. Lou. Implementing the Viterbi algorithm.*IEEE Signal processingmagazine* **3**

7. Simon Hykin,Communication Signals, Fourth edition. **4**

8. Xilinx Inc. ,Virtex FPGA Family: Complete Datasheet [Online]. Available:http://www.xilinx.com/ 8