ASIC Implementation of HDB3 Codec

Meshram Vaibhav Bhimrao, Ramesh T

***Abstract:** This paper demonstrates the working of HDB3 encoder & decoder and also its implementation at chip level. The HDB3 code consist of 3 modules namely violation module, balance module and polarity correction module. The decoder consists of violation detection module, balance detection module and polarity detection module. The encoder design accepts serial data from the information source in binary format. HDB3 encoder encodes the binary data into two bit symbol data. The encoder data is transmitted over a physical channel. At receiver's end when the data is present, the decoder detects the violation symbol and balance symbol using the violation and balance detection module. The polarity is restored by the polarity detection module. The HDB3 codec is a modified AMI generator, the design is targeted on 180nm technology provided by JAZZ foundry.*

The HDB3 codec's front-end design development and verification is carried out using QuestaSim simulator. ASIC implementation of HDB3 codec is done using SYNOPSYS tools.

Index Terms: HDB3 codec.

I. INTRODUCTION

In the domain of digital communication, the base band signal is usually regarded as the code's one kind of electricity expression. But in the actual base band transmission system, not all base bands electricity wave can transmit in the channel, therefore choosing the code that adapts to carry on the the base band transmission system is very important. The HDB3 code is one of a representative codes.

The research about coding in communication industry was an important topic of modern digital communication technology. when the line pattern was selected rightly, it was good for improving the communication quality, ameliorating the transmission performance and extending the transmission distance.HDB3(High Density Bipolar Codes)was called high density bipolar coding ,and it was allowed even when the "0" number up to no more than 3,which was conducive to the recovery of timing signal.

HDB3 code is an improved version of AMI code. It has all the advantages of AMI code i.e. error detection function, no DC component, narrow energy spectrum. Besides it overcomes the difficulty of extracting timing signal when there is a long sequence of zeros in the message to be transmitted. So it has been widely applied to the data transmission of high density information flow and recommended as the interface pattern in the digital transmission of base group, secondary group and third group by the CCITT G.703.Today the HDB3 encoders and decoders are mainly realized by ASIC and peripheral small and medium-sized chips, .the structures of which are all complex. HDB3 used in all levels of the European E-carrier system, the high density bipolar order 3 (HDB3) code replaces any instance of 4 consecutive 0 bits with one of the patterns "000V" or "B00V". The choice is made to ensure that consecutive violations are of differing polarity, i.e. separated by an odd number of normal + or - marks. These rules are applied on the code as it is being built from the original data. Every time there are 4 consecutive zeros in the code they will be replaced by 000−, 000+, +00+ or −00−.

Revised Manuscript Received on 30 August 2012
***Correspondence Author(s)**

Mr. Meshram Vaibhav*, Bhimrao, Department of VLSI Design and Embedded Systems, Visvesvaraya Technological University, UTL Technological Ltd., Bangalore (Karnataka), India.

Mr. Ramesh T., Department of VLSI Design and Embedded Systems, Visvesvaraya Technological University, UTL Technological Ltd., Bangalore (Karnataka), India.

II. BLOCK DIAGRAM ENCODER & DECODER

A. Encoder

The encoder follows the bottom up strategy of the design methodology and is implemented using the verilog HDL. The design consists of the 3 modules namely violation module, balance module and polarity module. Serial bit data is applied to the violation module. The violation module works on 400MHz clock. The violation module generates the output which consists of the violation symbol. The output from the violation module is integrated with balance module. The balance module implements 4 stage pipelining.

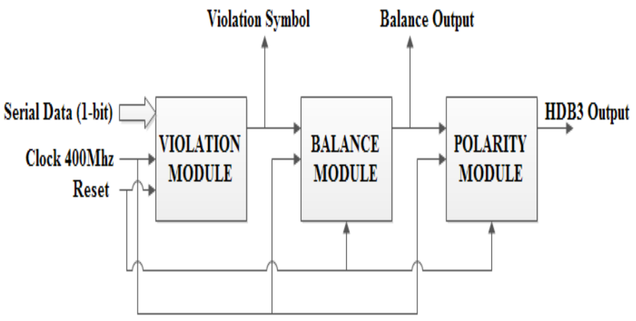


Fig.2.1: Block Diagram of HDB3 Encoder

The pipelining is employed to store the state of balance module over 4 clock cycles. The last module is the polarity module. The polarity module rectifies the wrong polarities associated with the balance output and gives correct HDB3 code.

B. Decoder:

It can be deduced from the coding rule of HDB3 code that there must be a non-zero code with the same polarity before each V in the HDB3 code sequence. So the working principle of the decoder is as follows if two adjacent non-zero code in the HDB3 sequence are of the same polarity, then the second one must be a violation code, so convert all the three code before violation to a sequence of "0000", then convert all the remained ±1s to 1s, and 0's to 0's still. When the wrong code is 0, first judge whether it can be replaced by a non-zero code with the same polarity as the first.

If the answer is no, then replace it by a non-zero code with the opposite polarity. When the wrong code is non-zero, first judge whether it can be replaced by a 0. If the answer is no then replace it by a nonzero code with the opposite polarity.

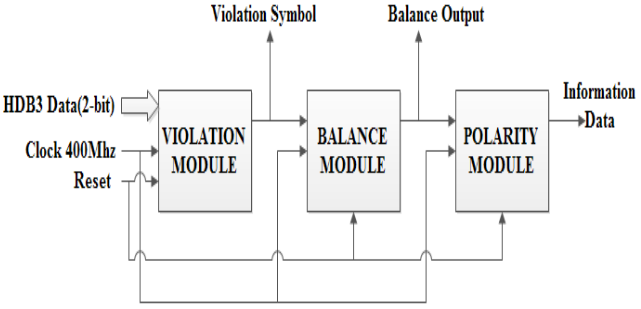


Fig.3.1: Block Diagram of HDB3 Decoder

The modification principle proposed in this design is only one of all the possible ones. When the wrong HDB3 code sequence is corrected based on this principle, there will be more 0's in the decoding results.

III. SPECIFICATIONS

ENCODER

Front-end specification

Input

User data	:	serial data single bit
Frequency	:	400Mhz
Reset	:	Synchronous
Clock	:	Synchronous

Output

HDB3	:	2-bit Symbol

DECODER

Front-end specification

Input

User data	:	2 bit symbol
Frequency	:	400 MHz
Reset	:	synchronous
Design	:	synchronous

Output

Base band signal	:	serial single bit data

Back-end specification for encoder and decoder

Input

STA clock	:	4ns
Input delay clock	:	(0.7 0.8)
Output delay clock	:	(0.5 0.7)
Reset delay	:	(0.5 0.7)
Input clock transition	:	(0.01 0.02)
Output load	:	(0.01 0.04)
Foundry target	:	JAZZ
Technology	:	180nm
Power supply	:	1.8V
IR drop total	:	5% (180mw)

IV. DESIGN AND IMPLEMENTATION

A. HDB3 Encoder Algorithm:

The flowchart presents the method to convert the information signal (data that has to be transmitted) into HDB3 encoded data. The HDB3 encoded data is suitable for transmission over physical channels. The steps of conversion are as follows.

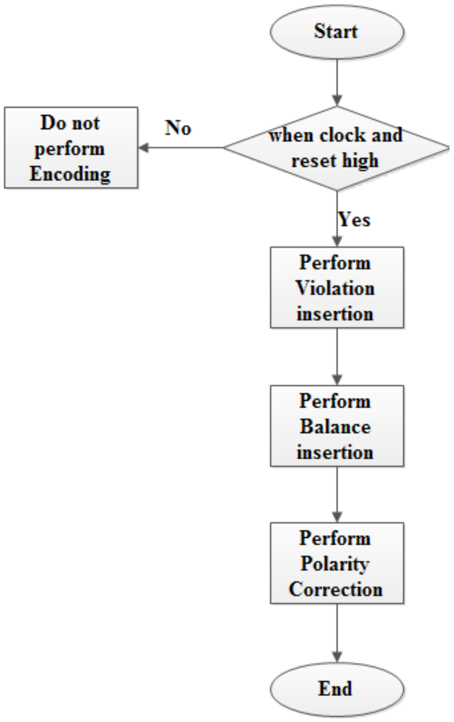

Start
when clock and reset high
No
Do not perform Encoding
Yes
Perform Violation insertion
Perform Balance insertion
Perform Polarity Correction
End

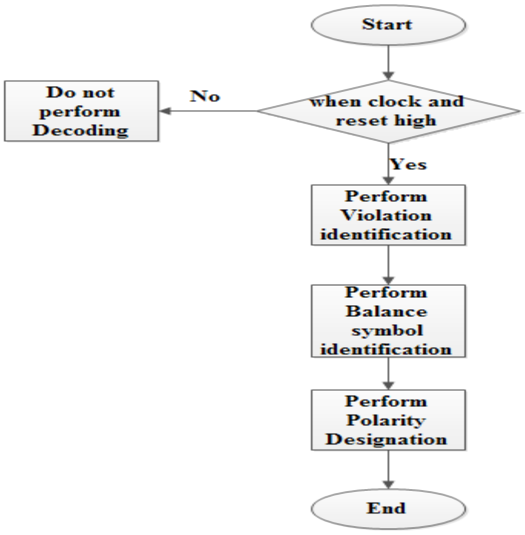
Fig.4.1: Hdb3 Encoding Flow Chart

B. HDB3 DECODER Algorithm:

The Decoder HDB3 working principle is as follows.

It can be deduced from the coding rule of HDB3 code that there must be a non-zero code with the same polarity before each violation code in the HDB3 code sequence. So the working principle of the decoder is as follows.

If two adjacent non-zero code in the HDB3 sequence are of the same polarity, then the second one must be a violation code, so convert all the three code before it to a sequence of "0000". Then convert all the remained ±1s to 1s and 0's to 0's still. The greatest advantage of the decoder in this design over others is that it has the function of decoding as well as error detecting.


Start
when clock and reset high
No
Do not perform Decoding
Yes
Perform Violation identification
Perform Balance symbol identification
Perform Polarity Designation
End

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

V. TESTS AND RESULTS

A. *Simulation and Synthesis results of Encoder*

The following are the simulation results of the violation, balance and HDB3 module. The reset signal is forced to active high by the assertion based test bench. At the positive edge of the clock the algorithm is executed by the MODEL-SIM as shown below.

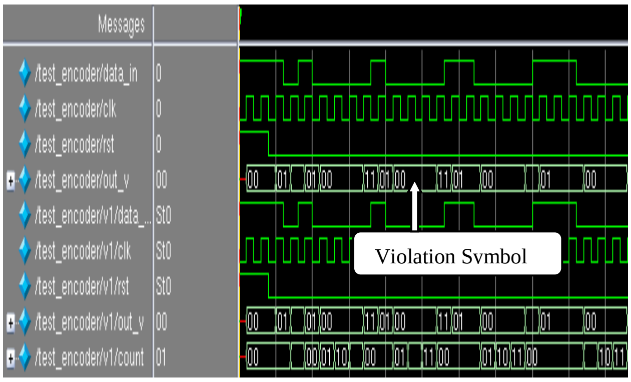


Fig.5.1: Violation Simulation

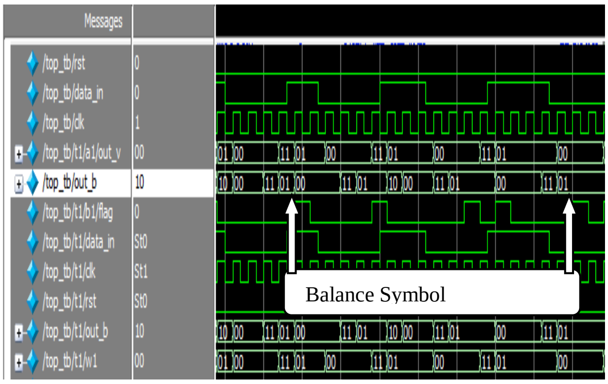


Fig.5.2: Balance Simulation

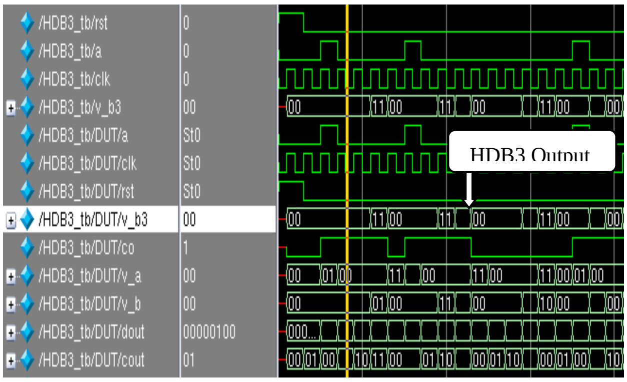


Fig.5.3: HDB3 Output

Post Synthesis Simulation:

The process of synthesis is to convert the functional code to a register transfer logic which shows the physical connection of the devices (transistors). The synthesis is done using the 2009 version Design compiler by Synopsys and ensured that pre synthesis and post synthesis simulation result matches. The following is the post synthesis simulation results of the violation, balance and HDB3 module. As the reset signal is forced to active high by the assertion based test bench. At the positive edge of the clock the algorithm is executed.

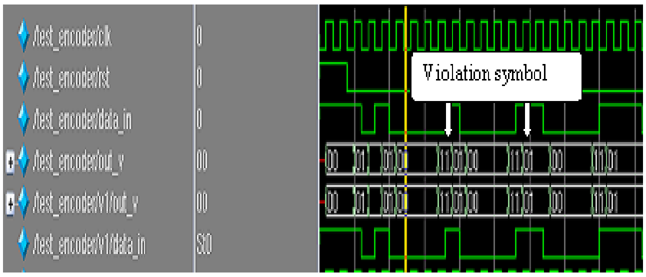


Fig.5.4: Synthesis Violation Simulation

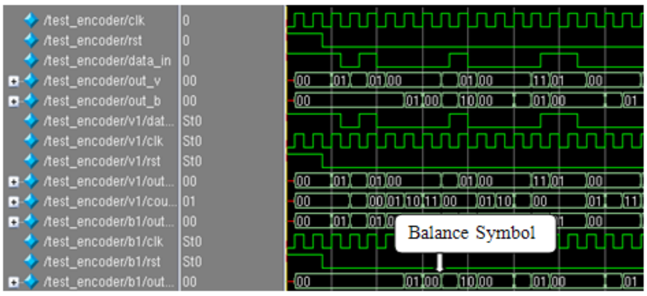


Fig.5.5: Synthesis Balance Simulation

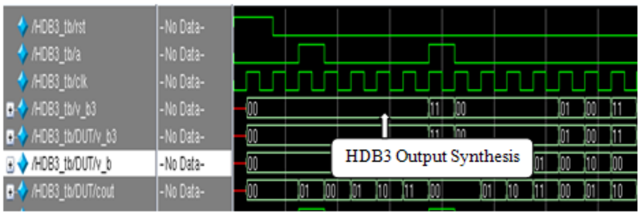


Fig.5.6: Synthesis Hdb3 Simulation

Set up analysis

****

Report: timing
VIOLATION-path type full
-delay type max
-max paths 1
Design: HDB3
Version: C-2009.06-SP2

****

Start point: rst (input port clocked by clk)
Endpoint: dout_reg [2]
(Rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Data required time 2.73
Data arrival time -1.29

Slack (MET) 1.44

Hold Analysis:

****

Report: timing
-path type full
-delay type min
-max paths 1
Design: HDB3
Version: C-2009.06-SP2

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

```
 ****************************************

Start point: cout_reg [1]
(Rising edge-triggered flip-flop clocked by clk)
Endpoint: cout_reg [1]
(Rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min
--------------------------------------------------------------
Data required time                              0.07
Data arrival time                              -0.16
 --------------------------------------------------------------
Slack (MET)                                     0.09
```

RTL View of the Encoder:

The following shows the representation of the netlist obtained using the DC_SHELL RTL viewer.

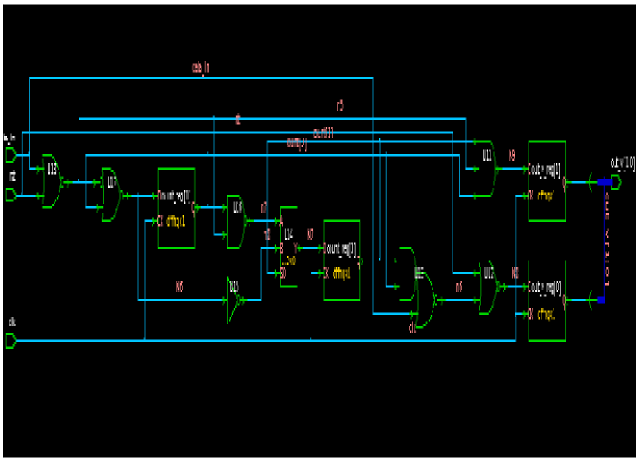


Fig.5.7: Violation RTL View

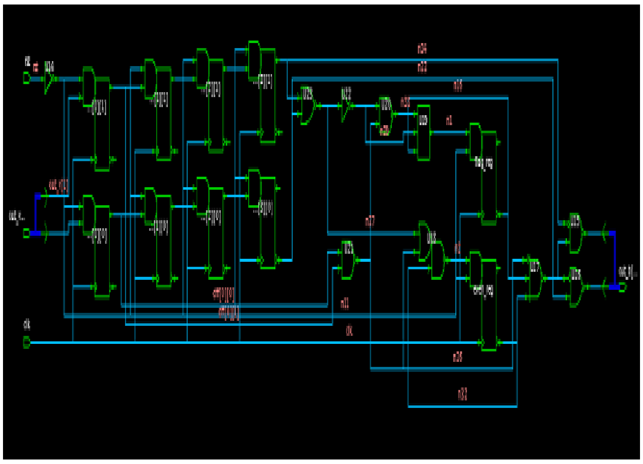


Fig.5.8: Balance RTL View

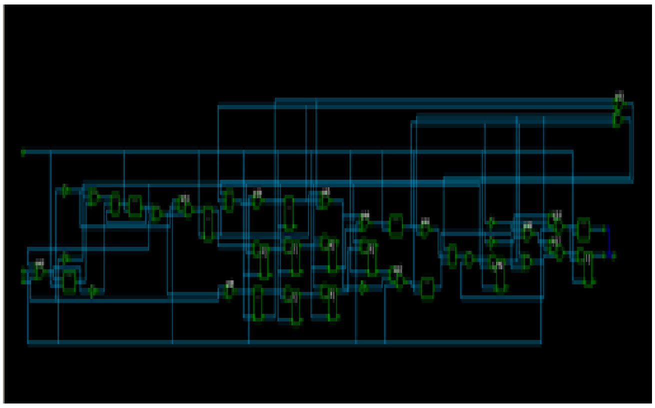


Fig.5.9: Hdb3 RTL View

Physical Design Layout:

The logic synthesis of the code was Carry the net list through a Physical Design flow which consist of the floor plan, placement, clock tree synthesis, and route, then A GDSII layout is generated for the design, which is programmed to physical design.

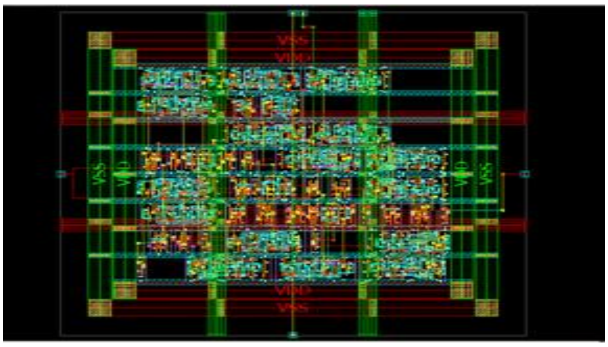


Fig.5.10: Layout View of HDB3

B. Simulation and Synthesis results of Decoder

The following are the simulation result of the HDB3 Decoder module. As the reset signal is forced to active high by the assertion based test bench. At the positive edge of the clock the algorithm is executed by the MODEL-SIM as shown below.

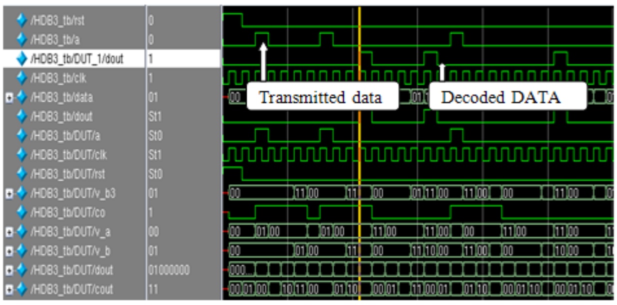


Fig. 5.11 Hdb3 Decoder Output

Post Synthesis Simulation:

The process of synthesis is to convert the functional code to a register transfer logic which shows the physical connection of the devices. The synthesis is done using the 2009 version Design compiler by Synopsys. The pre synthesis and post synthesis simulation result matches. The following is the post simulation result of the violation, balance and HDB3 module. As the reset signal is forced to active high by the assertion based test bench. At the positive edge of the clock the algorithm is executed.

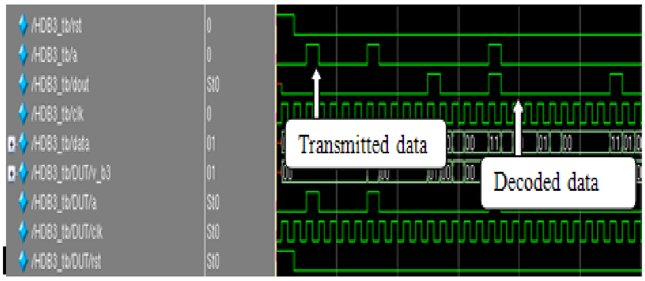


Fig. 5.12 Post Synthesis HDB3 Decoder Output

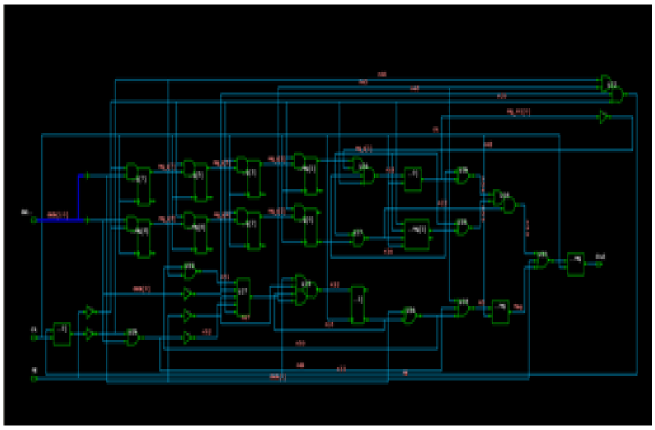
Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Setup analysis:

```
****************************************
  Report: timing
    -path type full
    -delay type max
    -max paths 1
 Design: decode
 Version: C-2009.06-SP2
****************************************
  Start point: data [1] (input port clocked by clk)
  Endpoint: reg_m_reg [0]
 (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max
  ---------------------------------------------------------------
   Data required time                          3.52
  Data arrival time                           -1.95
  ---------------------------------------------------------------
        Slack (MET)                           1.57
```

RTL View of the Encoder:

The following shows the representation of the netlist obtained using the DC_SHELL GUI viewer



5.13 HDB3 Decoder RTL View

Physical Design Layout:

The logic synthesis of the code was Carry the net list through a Physical Design flow which consist of the floor plan, placement, clock tree synthesis, and route, then A GDSII layout is generated for the design, which is programmed to physical design. The figure 5.14 shows the layout diagram of the Decoder.

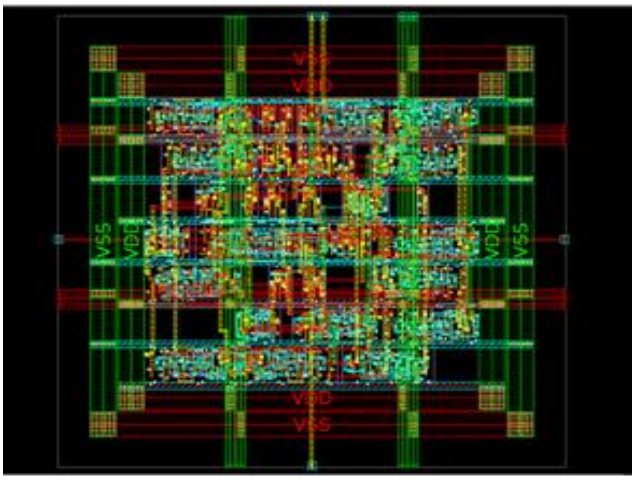


Fig. 5.14 Layout View of Decoder

Application of HDB3 CODEC

1. The HDB3 codec designed are used in encoding and decoding the serial data while transmitting E1 frame over a physical channel.
2. The HDB3 is also used in the transmission of the data in the E3 Carrier which implies, in the PABX or in the phone exchange.
3. HDB3 is also used in some of the routers in which E3 ports where present but these routers where not common.
4. HDB3 is also used in the ATM (Asynchronous transfer modes) of communication. Since timing signals information can be extracted from HDB3 codes the design can also be implemented asynchronously.
5. The ATM has interfaces with E1 and E3 switches thus the use of HDB3 is justified here.
6. HDB3 encoding is also used in the DS3.
7. The HDB3 encoding and decoding is also implemented in ADSL.

VI. CONCLUSION

The HDB3 codec basically consist of three modules namely violation, balance and polarity. These modules are independently developed and integrated to form the HDB3 encoder. The encoder design works at 400 MHz as lower frequency of operation with upper limit as 0.775 GHz. The coder accepts the serial data and gives the coded output as 2-bit symbol. In the similar fashion the decoder accept the 2-bit symbol and outputs a single bit data. The HDB3 coder and decoder adhere to the ITU G.703 standards. The coder and decoder consume 300mw of power with supply voltage of 1.8v at the maximum allowable voltage drop of 5% that of the power supply. The layouts are done using 180nm technology. The soft core can be used in the transmission of data in the OSI protocol in the physical layer whenever the E1 or E3 frames are being transmitted. The individual modules are highly flexible and modular but may not grantee exact synthesis and timing closure when synthesized with other unknown modules and technology apart from 180nm.

VII. ACKNOWLEDGMENT

First of all, my humble and sincere thanks to Mr.Venkatesh Prasad, CEO of RV-VLSI design center, for permitting to do this project at his institute till completion.

I am grateful to Mr. Purushotham Sannakariyappa, Design Engineer, and RV- VLSI design center, to guide me from the commencement to the end of the project it was under your guidance I was able to finish the project in the stipulated time. I would also like to thank Mr. Kaushik (Senior Design Engineer) Mr. Vinay Hulikal, Mr. Srinath (Physical Design Engineer). Their teaching is excellent.

My humble and sincere thanks to Dr. V .Venkateswarlu, Principal and HOD, VTU Extension Centre, UTL Technologies Ltd, Bangalore.

I would like to express my heartfelt thanks to Prof. Ramesh.T, Lecturer, UTL technologies, VTU Extn Center, Dr. Siva S Yellampalli for helping and understanding me.

REFERENCES

1. Xiaohua Wang, Ning Tang, Songqing Zhou, Yanggang Yin, “ASIC design of the HDB3 encoder chip”, 2011 Fourth International Conference on Intelligent Computation Technology and Automation.
2. Yang Zhang, Xiumin Wang, Yuduo Wang, “A New Design of HDB3 Encoder and Decoder based on FPGA”, 2009 Ninth International Conference on Hybrid Intelligent Systems
3. Zhang Chan-sen, Xu Qi, “A Design of HDB3 CODEC based on FPGA”, 2010, IEEE.
4. Daniel R. Hicks, “Tale of an IC Design Engineer-HDB3 encoder/decoder circuit”.
5. H.Kobayashi, “A Survey of Coding Schemes for Transmission or Recording of Digital Data,”IEEE Trans. on Communications, Vol. COM-19, p. 1087 (Dec. 1971).
6. A. Brosio, U. DeJulio, V. Lazzari, R. Ravaglia, and A. Tofanelli, “A Comparison of Digital Subscriber Line Transmission Systems Employing Different Line Codes,” IEEE Trans. on Communications, Vol. COM-29 (11), p. 1581 (Nov. 1981).
7. R.F. Lyon, “Two-Level Block Encoding for Digital Transmission,” IEEE Trans. on Communications, Vol. COM-21 (12), p. 1438 (Dec. 1973).
8. E. Kretzmer, “Generalization of a Technique for Binary Data Communication,” IEEE Trans. on Communication Tech., Vol. COM-14, (Feb. 1966).
9. Rung-Bin Lin and Meng-ChiouWu. A new statistical approach to timing analysis of VLSI circuits. In VLSI ’98: Proceedings of the Eleventh International Conference on VLSI Design: VLSI for Signal Processing, page 507, Washington, DC, USA, 1998. IEEE Computer Society.
10. Anirudh Devgan and Chandramouli Kashyap. Block-based static timing analysis with uncertainty. In ICCAD ’03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design, page 607, Washington, DC, USA, 2003. IEEE Computer Society.
11. Aseem Agarwal, Kaviraj Chopra, and David Blaauw. Statistical timing based optimization using gate sizing. In DATE ’05: Proceedings of the conference on Design, Automation and Test in Europe, pages 400–405, Washington, DC, USA, 2005. IEEE Computer Society.
12. J. L. Neves and E. G. Friedman. Design methodology for synthesizing clock distribution networks exploiting nonzero localized clock skew.

    IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 4(2):286–291, June 1996.
13. J. N. Franklin and J. R. Pierce, “Spectra and Efficiency of Binary Codes without DC,” IEEE Trans. on Communications, Vol. COM-20 (6), p. 1182 (Dec. 1972).
14. Assuncao P,"Post-processing of MPEG - 2 coded video for transmission at lower bit rates," IEEE IntConfAcoust, Speech, Signal Processing, 1996 (4):1998 - 2001.
15. R. H. J. M. Otten, Automatic Floorplan Design, Proc. 19th ACM/IEEE Design Automation Conf. (1982), pp. 261-267.
16. Lou Scheffer. Explicit computation of performance as a function of process variation. In TAU ’02: Proceedings of the 8th ACM/IEEE international workshop on Timing issues in the specification and synthesis of digital systems, pages 1–8, New York, NY, USA, 2002. ACM Press.
17. S. R. Nassif. Modeling and analysis of manufacturing variations. In Custom Integrated Circuits, 2001, IEEE Conference on., pages 223–228, San Diego, CA, May 2001.
18. Ngspice Circuit Simulator. http://ngspice.sourceforge.net.
19. http://www.cpe.ku.ac.th/~nguan.