Research Journal of Applied Sciences, Engineering and Technology 4(7): 840-845, 2012
ISSN: 2040-7467

Submitted: December 09, 2011 Accepted: January 10, 2012 Published: April 01, 2012

A Single Core Hardware Approach of MPEG Audio Decoder for Real-Time Transmission

M.B.I. Reaz and Mohd Marufuzzaman
Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia

Abstract: The decoding of the voice audio bit stream is an issue in terms of real-time transmission of high quality voice audio over the Internet. A stand-alone chip to perform decoding is a better solution over software approach. The MPEG audio compression provides high compression with minimal loss. This study describes a VHDL model of MPEG audio layer 1 decoder that perform concurrent processing while receiving voice quality audio input bit stream at a constant bit rate and simultaneously producing a stream of 8-bit monopole PCM samples at a constant sampling frequency in real time.

Key words: Audio decoder, FPGA, MPEG, VHDL

INTRODUCTION

The MPEG1 audio becomes a popular algorithm especially in its usage in the Internet for audio data compression due to its high quality with various compression rate, sampling rate and mode (ISO/IEC, 1993; Kiah *et al*., 2011). The MPEG1 is classified into three different layers by its complexity. MPEG1 allows audio signal to be compressed up to 12 times so that it can be transmitting at low bitrate. It can compress 1.5Mbit/sec CD quality audio data into 32 to 448 Kbit/sec for layer 1. It requires supporting sampling rates of 32, 44.1 and 48 Khz. In encoder, the input signal is partitioned into 32 different frequency region and then decimated for Nyquist sampling rate.

LAME developed their own MPEG encoder and decoder for both MPEG1 and MPEG2 that runs independently on its own codes. The developed algorithms are adaptive and automatically perform adjustments to optimize performance (Cheng, 2011). MAD (MPEG Audio Decoder) is another organization that developed MPEG encoder and decoder using fixed-points. MAD provides a 24-bit PCM output to improve the quality of the audio signals. This allows 16-bit PCM applications to use the extra resolution to increase the audible dynamic range through the use of dithering (Leslie, 2011). Enerstam and Peman (1998) presented an MPEG Audio Encoder/Decoder using a DSP chip, the TMS320C6701. Dong *et al*. (2010) presented another DSP based MPEG-1 audio decoder algorithm using floating point implementation. All the designs developed by LAME, MAD and DONG are for software implementation, which allows floating point implementation and the usage of pointers. Similarly, the hardware implementation of the encoder/decoder using a DSP chip by Joakim and Jan also used floating-points in their calculation. However, in VHDL implementation, designs with floating points are not synthesizable. Hence, the implementation in VHDL had to be done using only integers. Furthermore, the range of the integers in VHDL is confined to 2^{31} -1 to -2^{31}. In software implementation, the input is extracted by reading a string of bits stored in a file with a file pointer. Hence, they do not require real-time implementation. However, in VHDL implementation, the input is an input stream of bits at a fixed bit rate of 128 kbits/s. The input is computed in real-time and the output is 8-bit PCM samples at a frequency of 44.1 kHz. As a result, there is a need to perform concurrent processing whereby the input handling, processing and output handling processes run simultaneously.

The Field-programmable Gate Arrays (FPGA) offers a potential alternative to speed up the hardware realization (Coussy *et al*., 2009). From the perspective of computer-aided design, FPGA comes with the merits of lower cost, higher density, and shorter design cycle (Akter *et al*., 2008; Verma *et al*., 2009). It comprises a wide variety of building blocks. Each block consists of programmable look-up table and storage registers, where interconnections among these blocks are programmed through the hardware description language. This programmability and simplicity of FPGA made it favorable for prototyping digital system. FPGA allows the users to easily and inexpensively realize their own logic

Corresponding Author: Mohd Marufuzzaman, Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia, Tel.: Phone: +603-89216316; Fax: +603-89216146

networks in hardware. FPGA also allows modifying the algorithm easily and the design time frame for the hardware becomes shorter by using FPGA (Reaz *et al*., 2007).

This research is initiated at the System Design Laboratory in the Universiti Kebangsaan Malaysia in the middle of 2010 to develop a new model of MPEG audio layer 1 decoder using VHDL so that new applications such as digital voice memo and communication devices can spur from this technology. The use of VHDL for modeling is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels (architectural, register transfer and logic level) employed in the design (Reaz *et al*., 2006). In the computation of method, the problem is first divided into small pieces, each can be seen as a submodule in VHDL. Following the software verification of each submodule, the synthesis is then activated. It performs the translations of hardware description language code into an equivalent netlist of digital cells. The synthesis helps integrate the design work and provides a higher feasibility to explore a far wider range of architectural alternative. The method provides a systematic approach for hardware realization, facilitating the rapid prototyping of the fuzzy based subway train braking system.

MATERIALS AND METHODS

This research is to develop a new model of MPEG audio layer 1 decoder using VHDL. The decoder is capable of decoding voice audio bit streams into 8-bit PCM samples in the form of standard logic vectors. The decoder is also designed to perform the decoding in real-time that as the bit streams are inputted into the decoder, the corresponding PCM samples will be outputted in real-time. The application of the model can be used in various voice compression devices.

The storage unit of the source is of an output mechanism that provides bits is of bit streams. The output of the decoder is connected to an 8-bit monopole DAC which convert the digital PCM samples into analog signals then the analog signal is converted from electrical signal to audio sound using a speaker.

Implementation choice: In modeling the voice audio decoder, MPEG audio layer 1 is chosen due to its less complexity. As human audible range is 20 kHz, therefore to avoid imaging according to nyquist theory a sampling frequency of 44.1 kHz is selected. The range of bit rates in MPEG audio layer as ISO standards are from 32 to 448 kbps. Since the model is for voice audio bit streams, therefore 128 kbps bit rate is chosen. (120 eight kbps) require a storage device that can stream at 16 kbps. In calculation the integer point is used by upscale the values as the input and output of the decoder is bit streams and 8-bit PCM samples respectively which is a standard logic vector.

To avoid the trigonometric functions in calculations of decoding process a ROM is implemented to store the entire coefficient. This is possible since, the Han Window coefficient is only dependent on the sampling frequency, as the sampling frequency is predetermined.

The model is designed to handle the bit stream and not dependent on the streaming of the input. A separate process that runs concurrently with the decoder unit is designed to handle the bit streams. This allows the decoder to function even the input of the bit streams are not in a constant rate. A separate output unit is designed to run concurrently with the decoder unit and a buffer is used to allow some tolerant in the output flow thus ensure the PCM samples would always be at the correct frequency.

MPEG audio decoding algorithm: The decoding process is the process of expanding the encoded MPEG audio layer 1 format into PCM samples. This decoding process is separated into three blocks, which are the frame unpacking, reconstruction and inverse mapping. The frame unpacking block of the decoding algorithm unpacks and recovers the various piece of information in the frame. The frame unpacking block also does error detection if error-check is applied (i.e., protection bit is cleared). The reconstruction block reconstructs the quantized version of the set of mapped samples using the equation,

$$S'' = \left(\frac{2^{nb}}{2^{nb} - 1} \right) \times \left(S''' + \frac{1}{2^{nb-1}} \right) \quad (1)$$

where,

S'' is the requantized value
S''' is the fractional number
nb is the no. of bits allocated to the samples.

The inverse mapping block transforms these mapped samples back into uniform PCM. The ISO/IEC (1993) is followed for MPEG audio layer decoder.

The MPEG audio layer 1 bit streams are separated into frames. Each frame consists of four parts; the frame header, error checks, audio data and ancillary data. The frame header contains the basic decoding parameter of the stream includes sampling frequency and layer.

The error checks hold the optional 16-bit parity-check word used for error detection within the encoded bit stream. The cyclic redundancy check of the frame is calculated based on the length of the frame and compared to verify the valid frame. The audio data consists of

encoded information to be decoded. It is separated into three sections the bit allocation code, scalefactors code and subband samples.

The bit allocation determines the number of bits that represent the samples in a particular subband. The size of the bit allocations code is 4 bits each, since there are 32 subband, effectively it uses 128 bits. The number of bits allocated for each subband is the integer value of the bit allocation code +1, except for 000 where no bits are allocated and 111 which is invalid. The scalefactors scale the normalized sample value to its actual value for a particular subband is an index based on the ISO. Scalefactors code is only present if the bit allocated to that particular subband is not zero. The subband samples are the normalized value of each sample. Since there are 12 parts of 32 subband samples in one frame, the arrangements of the samples are in parts with the first 32 subband samples followed by the next subband samples. The number of bits used to represent each sample is based on the bit allocated to the subband of the sample (ISO/IEC, 1993). The ancillary data is not defined by the ISO and it is user defined which is not significant in this research frequency.

Design automation: Many novel programs are developed to automate the design by speeding up the required designing time. Most of the automation used in the development of the decoder involves formatting values as the decoding algorithm involves the usage of constants provided by the ISO standards. These constants are implemented as ROMs thus it is hard-coded into the VHDL model. A program is also designed to convert the PCM output into a wav format to evaluate as an audible output beside absolute values.

A program is developed to extracts the scalefactors from the ISO specifications into VHDL format to form the scalefactors constants up scaled by a factor of 10^4.

Another program is developed to calculate the N_{ik} coefficient since it involves trigonometric functions, which is not supported by VHDL. The N_{ik} coefficients are calculated based on the equation:

$$Nik = \cos\left[(16+i)\times\frac{(2k+1)\pi}{64}\right]$$

for i = 0 to 63 and k = 0 to31 (2)

The calculated values are typically in the range of -1 to 1; thus the value is up scaled by a factor of 10^4.

Another program is developed to extracts the window Coefficient D_i from the ISO specifications into VHDL format to form the D_i constants upscale by a factor of 10^4.

Due to the limitation of file management functions in the VHDL, the test vectors is extracted from a text file. Since the MPEG audio file is typically a binary file, thus MPEG audio file is extracted bit by bit and formatting it as a text file, where each line contain either 1 or 0. By performing the formatting, the testbench read this text file and provide the test vectors to the decoder as a bit streams which is developed using another program.

The output of the simulation is in the form of a text file, where each line will contain the integer value of a PCM sample. In order to perform some auditory analysis on the PCM output, a program is developed to convert the PCM samples into a wav file. The wav format is a 44-byte header contains information such as the number of channels, sampling rate, byte rate and bits per sample.

VHDL describes digital systems by the behavioral model or the structural model thus allows an easy and efficient designing manner. The VHDL model of MPEG audio layer 1 decoder is separated into three sections, which are the VHDL format consideration, the top level design and behavioral description.

VHDL format consideration: VHDL language is capable of designing in hierarchies, reusing components, error management and verification allows describing huge complex circuitry efficiently (Sjoholm and Lindh, 1997).

In designing VHDL model, three packages std_logic_1164, std_logic_arith and std_logic_unsigned is defined from the IEEE library. These packages include definition of standard logic data types, arithmetic operations involving standard logic data types and integer data types and integer to standard logic vector conversion and vice versa. The hift package perform bit-shifting operations on standard logic vector data type. The bit shifting operations is used for bit management and multiplication of power 2 purposes.

Both the entity and architecture of the MPEG audio decoder is ecoder In the entity area, all the input and output ports of the decoder component is defined. There are 4 ports in the ecoder component, which are Clock, Input, Output and Outputready. Both the Clock and Input are of std_logic, a single bit port. The Output is of type std_logic_vector (0 to 7), an 8-bit port. The Outputready is of type std_logic. The architecture section follow the entity section and describe the behavioral relation between the inputs and the outputs.

In the architecture of the VHDL code, it is made up of both concurrent part and also sequential part (processes). The decoder has three processes, nputhandling ainprocessing and utputhandling These three processes are running concurrently with each other (i.e. the second process does not need to wait for the first process to complete before execution). Within the process, the processing is done sequentially. The sensitivity list is a list of signal (concurrent) that the process monitors for. The process only start to process when there is some changes in the signals in the sensitivity list. For example, in the nputhandling process, the sensitivity list has the signal lock this means that whenever the clock signal changes, the process start and

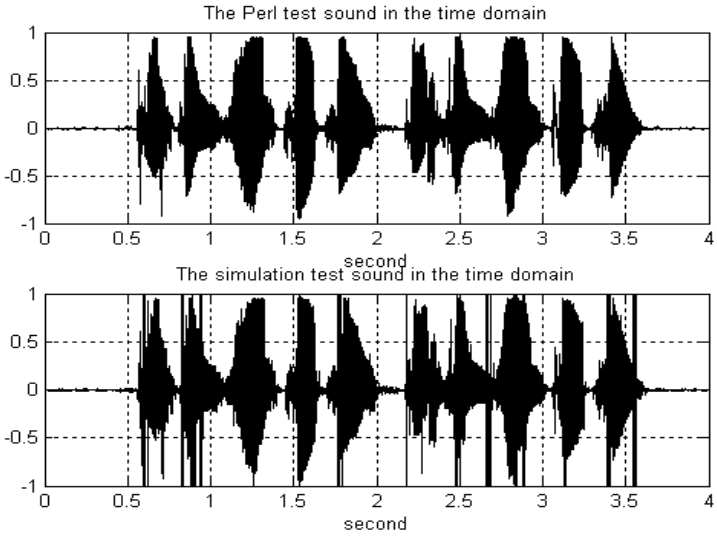

Fig. 1: Time domain waveforms of the Perl output & simulation output

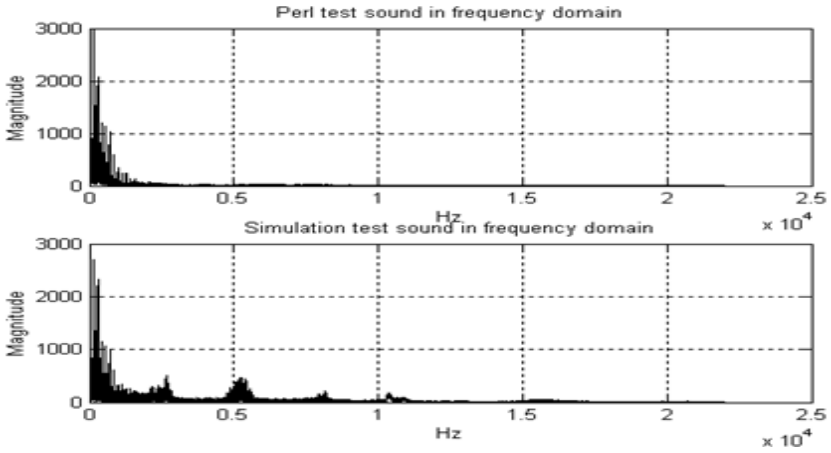

Fig. 2: Frequency domain waveforms of the Perl and simulation output

all the statements in that particular process execute sequentially. For the case of the ainprocessing and utputhandlin their sensitivity list consists of rameready and utputclock signal respectively.

Throughout the design of the decoder, many ROMs and buffer are used. These ROMs and buffer stores integer values, thus arrays of size required by the ROM or buffer is created for this purpose.

Top level design: The decoder has an input of type std_logic for the bit stream and an output of type 8-bit std_logic_vector for the PCM output. A clock signal of type std_logic is added to the decoder to set the bit stream data rate. In order to aid the testing of the MPEG Audio decoder, an utputclock signal of type std_logic is added. This signal toggle whenever the new PCM output is outputted.

Two critical issues are considered to operate the decoder smoothly. First, the decoder is able to constantly accept the continuous bit streams of data even if there is some delay in the processing. Second, the decoder output the PCM samples at a constant rate of 44100 Hz, thus independent of the processing time required. In order to handle these two issues, the input portion, the processing portion and the output portion runs asynchronously and parallel to each other thus three processes nputhandling ainprocessing and utputhandling of the decoder runs concurrently.

Behavioral description: The nputhandling process is designed to handle the continuous input of bit stream. This process performs bit management and frame detection. Once the new frame is detected the whole frame passed to the frame buffer uff2 This buffer is declared as a signal, thus it functions as an intermediate

buffer between the nputhandling and ainprocessing The input is read from the nput port. The input rate is synchronized by the rising edge of an input clock, which runs at the desired data rate. The nputhandling runs concurrently with the other processes in VHDL. An intermediate buffer is placed to pass the frame to the ainprocessing process. The ainprocessing process is the main processing unit of the decoder. At the top level, the ainprocessing process is executed based on the rameready signal, which is supplied by the nputhandling process when a new frame is detected and pushed to the frame buffer.

In this model, the bits shifted left by the corresponding bits extracted using the shifting method. Since the function of power is not available in VHDL, therefore, by shifting the value of "0000000000000001" to the left by "nb" is effectively multiplying 1 by 2^{nb}. Thus, the values of 2^{nb} and $2^{(nb-1)}$ are substituted with two variables where their values are calculated via the shifting method.

In VHDL, integer data type is limited to the range of 2^{31} to -2^{31}. In order to make no overflow, all the values are up scaled by a factor of 10^4. Since, the calculations involve many multiplication operations; the convention is consistent by dividing the result of the multiplication by a factor of 10^4. Thus, in the VHDL codes, the scaling down of values is done in every multiplication operation.

In the VHDL model, the output timing is fixed at the sampling frequency of 44.1 kHz. In order to push the ainprocessing process samples into the utputhandling process a PCM samples buffer (Concurrent) is used, similar to the nputhandling to ainprocessing buffer. The PCM samples buffer provide the function of making the design more tolerant to timing deviation. The size of the buffer is set to 768 (2x384) PCM sample as each frame consists 384 PCM samples thus each time the ainprocessing process output 384 samples. By having the buffer, the ainprocessing process fills up the buffer alternatively from upper 384 to lower 384 PCM samples therefore the timing of the data rate input need not to be exactly matched to the output timing of 44.1 kHz. The data rate only needs an average of desired data rate. The utputhandling process manage the outputting of the PCM samples at a rate of 44.1 kHz by inverting a utputclock signal at a rate of 44.1 kHz effectively the frequency of the utputclock at 22.05 kHz. The sensitivity list of the utputhandling process is added with the utputclock signal therefore everytime the utputclock is toggle (inverted), the utputhandling process start execution. When executed, the process pumps out a PCM sample from the PCM buffer. An internal counter move the pointer of the buffer from 0 to 767 then back to 0 again. As a result, the PCM sample outputted at a constant rate of 44.1 kHz.

After modeling the decoder in VHDL, a testbench program is developed in order to test the functionality of the VHDL model. Three important aspects are checked during the simulation, which are output values, handshaking protocols and timing.

Testbench development: The VHDL model of the decoder was instantiated as a component in the testbench. The testbench is considered as a component with no ports and the decoder is encapsulated by the testbench.

The test vectors used in the testing extracted from a text file, since the number of test vector bits are in terms of a few hundred thousand bits. The td.textio package in the standard library is used to develop testbench to read the test vectors from a file line by line (a bit per line) and also write the output of the VHDL model in to a file line by line (a PCM sample in integer per line).

For inputting of test vectors, the process reads a line from the file, then the value of that line is extracted as integer and converted into td_logic and fed to the nput port of the decoder at every rising edge of the clock which runs at the desired bit rate. After that, the next line is read and the process repeats. For outputting of the decoder to a file is done by first converting the td_logic_vector type value of the utput port into integer data type. It is then written in the file as a new line. This process contains the utputready signal in its sensitivity list thus executes whenever the signal toggles. Effectively, the process write the new PCM sample values to the file once notified by the decoder that the output is ready via the utputready signal.

Simulation verification: When the simulation is ran, all the signals in the design is tracked individually in order to facilitate debugging of bugs in the design. The output values are verified through audio inspection by converting the output file into wav format.

There are two handshaking protocols, one at the nputhandling and ainprocessing processes interface and another at the utputhandling process to the utputready port of the decoder. For the first handshake, when the nputhandling process detects the new frame, it is sent to the frame buffer (concurrent signal) and the rameready signal is toggled. When the ainprocessing process detects the toggling of the rameready signal, it perform the decoding operations and output the 384 samples to the PCM samples buffer (concurrent signal).

For the second handshake, whenever the utputhandling process outputs the new PCM sample, the utputready signal is toggled thus indicate the new PCM samples is outputted.

The outputting of the PCM samples is at a rate of 44.1 kHz, thus the utputready signal is toggle at a rate of 44.1 kHz. As a result the utputready signal is a square-wave of 50% duty cycle and a period of 45.3 ms. On top of that, the input clock is also at a rate equal to the desired bit rate. The bit rate is 128 kbps, thus the input clock is effectively a square-wave of 50% duty cycle and a period of 8.8μs.

RESULTS AND DISCUSSION

The analysis and discussion of each stage of the design flow is carried out in three different angles audio

inspection, time domain waveform and frequency domain waveform. Comparison is made between the wav outputs of the stages. After obtaining the wav file from the conversion program, a first level audio inspection is performed. From the auditory inspection, it is concluded that there are many spikes in the sound, but there is no attenuation and frequency shift distortion heard. Since the Perl model result is considered more accurate, the comparison is done between the VHDL simulation results and the Perl model results. In order to analyze the noise injected by the VHDL model, an analytical comparison is done on the time and frequency domain waveforms using Matlab.

From the time domain waveforms in Fig. 1, it is verified that there are spikes being injected by the VHDL model as observed during auditory inspection. Although, there are spikes in the waveform, the general waveform of the sound is still dominantly observed. Thus, some filtering can be used to remove the spikes and satisfactory results can be obtained. These spikes are due to the lack of precision available in the VHDL model, as the integer range is limited.

From the frequency domain waveforms in Fig. 2, spikes injected is observed as high frequency components. There are four major spikes in the frequency domain waveform of the VHDL simulation result, which occurs at around 2200, 5000, 8000 and 12000Hz, respectively. Since, the human voice frequency response dominate mainly at 2000Hz, thus a low-pass filter is able to remove the spikes with a cut-off frequency at 2000Hz. As conclusion, the results of the VHDL simulation was acceptable although distortion was suffered because these spikes.

CONCLUSION

A new model of MPEG Audio Layer 1 decoder using VHDL is successfully designed to get continuous MPEG audio layer 1 bit streams as input and output the decoded 8-bit PCM samples as a standard logic vectors. The decoder is able to function normally, even if there is some deviation of data rate in the input. The decoder portrays some errors and limitation such as the spikes at the output. These spikes are due to the limited precision possible of the decoder. In order to solve the problem, a low pass filter is required after the decoding of the output samples. Thus, this problem does not pose a serious threat to the design.

REFERENCES

Akter, M, M.B.I. Reaz, F. Mohd-Yasin and F. Choong, 2008. Hardware implementations of image compressor for mobile communications. J. Commun. Technol. Electron., 53(8): 899-910.

Cheng, M., 2011. The L.A.M.E. (Lame Ain an MP3 Encoder) Project. Retrieved from: http://lame.sourceforge.net/

Coussy, P., D.D. Gajski, M. Meredith and A. Takach, 2009. An introduction to high-level synthesis. IEEE Design Test Comp., 26(4): 8-17.

Dong, Z., G. Cao and Y. Wang, 2010. Design of DSP-Based Digital Audio Processing System and Implementation of MPEG-1 Audio Layer III Decoding Algorithm. International Conference on Intelligent Computation Technology and Automation (ICICTA). pp: 23-26.

Enerstam, J. and J. Peman, 1998. Hardware implementation of MPEG audio real-time encoder. M.A. Thesis, Luleo University of Technology, Luleo, Sweden. Retrieved from: http://www.mp3-tech.org/programmer/docs/report.zip.

ISO/IEC International Standard CD 11172-3, 1993. Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s-Part 3, Audio.

Kiah, M.L.M., B.B. Zaidan, A.A. Zaidan, A.M. Ahmed and S.H. Al-Bakri, 2011. A review of audio based steganography and digital watermarking. Int. J. Phys. Sci., 6(16): 3837-3850.

Leslie, R., 2011. MAD: MPEG Audio Decoder. Retrieved from: http://www.mars.org/home/rob/proj/mpeg/.

Reaz, M.B.I., F .Choong and F. Mohd-Yasin, 2006. VHDL modelling for classification of power quality disturbance employing wavelet transform artificial neural network and fuzzy logic: Simulation. Trans. Soc. Modelling Simulation Int., 82(12): 867-881.

Reaz, M.B.I., F. Choong, M.S. Sulaiman and F. Mohd-Yasin, 2007. Prototyping of wavelet transform artificial neural network and fuzzy logic for power quality disturbance classifier. J. Electric Power Comp. Syst., 35(1): 1-17.

Sjoholm, S. and L. Lindh, 1997. VHDL for Designers. Prentice Hall.

Verma, A., S. Dhingra and M.K. Soni, 2009. Design and synthesis of FPGA for speed Control of induction motor. Int. J. Phys. Sci., 4(11): 645-650.