Power and area Efficient Decoder for DAB audio decoder

Pathuri Sateesh
MTech Student
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

M.Shoban Babu
Assistant Professor
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

ABSTRACT: *A new approach to design a DAB audio decoder is introduced to improve the quality of audio at the receiver end. Integrating an MPEG-1 Layer II (MP2) decoder and Advanced Audio Coding Low Complexity (AAC LC) decoder provides basic audio decoding for DAB in FPGA. The audio frames data are generated from DAB channel decoder are stored in RAM. The bit stream Demultiplexer parses the quantized spectrum data in the audio frame and stores them in to the audio RAM. The inverse quantization block reads the quantized spectra from the audio RAM, performs the inverse quantization computations, and writes back the result to the audio RAM. The synthesis filter reads the inverse quantized spectra from the audio RAM, generates the time domain Pulse Code Modulation (PCM) samples, and writes them back to the audio RAM. The whole projects run by using XILINX 12.1 ISE. Finally analyze the power consumption and area occupied by the designed architecture.*

Keywords: *DAB-Digital Audio Broadcasting, AAC LC-Advanced Audio Coding Low Complexity, IQ-Inverse Quantizer, IMDCT-Inverse Modified Discrete Cosine Transform, FPGA-Field Programmable Gate Array*

INTRODUCTION:

Radio broadcasting is one of the most widespread electronic mass media comprising of hundreds of providers, thousands of HF (High Frequency) transmitters and billions of radio receivers worldwide. The new digital radio system, Digital Audio Broadcasting (DAB) has the capability to replace the existing AM and FM audio broadcast services in many parts. Digital audio data need to be compressed for

transmission. Digital audio compression allows the efficient storage and transmission of audio data. The basic task of a perceptual audio coding system is to compress the digital audio data in such a way that the compression is efficient as possible, i.e. the compressed file is as small as possible and the reconstructed (decoded) audio sounds exactly (or as close as possible) to the original audio before compression. Other requirements for Audio compression techniques include low complexity. Perceptual encoding is a lossy compression technique, i.e. the decoded file is not a bit exact replica of the

Existing System:

AAC LC Decoder With the advantages on compression ratio and audio quality, MPEG AAC has been used for a wide range of applications from Internet audio to digital audio broadcasting and multichannel surround sound. Even in handheld products such as mobile phones or music players, they support the decoding of AAC files. However, there are several complex algorithms used in the AAC decoding flow which make it difficult to implement as an efficient design. Moreover, power consumption is also an important issue, particularly for portable devices. There are three different profile defined in AAC. They are the main profile, the low-complexity (LC) profile, and the scalable sampling-rate profile. It allows tradeoffs in audio quality and encoding/decoding complexity for different applications. AAC provides an audio signal that has CD quality at 96-128 kbps/stereo and bit rate 30% lower than that of MP3. Among the three profiles, the LC profile can provides nearly as high audio quality as the main profile but with significant savings in memory and processing requirements. In AAC audio coding, there are two

kinds of audio transport formats. One is Audio Data Interchange Format (ADIF), and the other is Audio Data Transport Stream (ADTS). For the ADIF format, it puts all data controlling the decoder (like sampling frequency, mode, etc.) into a single header preceding the actual audio stream. It is useful for file exchange but does not allow for break-in or start of decoding at any point in time. For the ADTS format, it packs AAC data into frames with headers and allows decoding to begin in the middle of an audio bit stream. AAC is the most advanced MPEG standard for digital audio compression. The design flow of the AAC LC profile is shown in Fig.1. There are several tools used in AAC decoding, which include bit stream parser, Huffman decoding, pulse data decoding, Inverse Quantizer (IQ), rescale, Middle/Side (M/S) and Intensity Stereo, Temporal Noise Shaping (TNS), and Filter bank.

Proposed System

MP2 and AAC LC decoder are both based on the psychoacoustic model for audio compression using a time-tofrequency transformation. So even if two audio decoders have different decoding algorithms, their decoding flow charts are similar. As shown in Fig. 2. It can be seen that both audio decoders have the following blocks bit stream demultiplexer, inverse quantization block, and synthesis filter. These blocks take up most of the computation process for the two audio decoders and lead to low power consumption. The audio frame data are generated from the DAB channel decoder and stored in RAM. A bit payload module conducts the task of reading the desired bits from RAM and transferring them to the MP2 decoder. The bit stream demultiplexer parses the quantized spectrum data in the audio frame and stores them into the audio RAM via the audio RAM interface. The inverse quantization block reads the quantized spectra from the audio RAM, performs the inverse quantization computations, and writes back the result to the audio RAM.

The synthesis filter reads the inverse quantized spectra from the audio RAM, generates the time domain Pulse Code Modulation (PCM) samples, and writes them

back to the audio RAM. The PCM samples are then read out by the Digital Audio Interface (DAI) to drive an audio Digital-to-Analog Converter (DAC) to play the sound. The coefficients and constants needed for each block are stored in the audio ROM. The Processing Unit (PU) is a very compact and fully customized circuit which is composed of adders, D-type flipflops, and multipliers, and is designed especially for MP2 and AAC decoding.

The PU and each block of the audio decoder exchange data via the audio RAM Interface. The results from each block are written to the audio RAM, and then read out by PU for processing. After the PU has finished processing, the results are written back to the audio RAM and wait to be read out by another block.

With this design architecture, the computation circuits of all the blocks are shared with only one PU and thus the chip area can be greatly reduced. Because the computations of each block are fixed, the control logic for the PU is also fixed and can be implemented in ROM to further reduce the chip area.
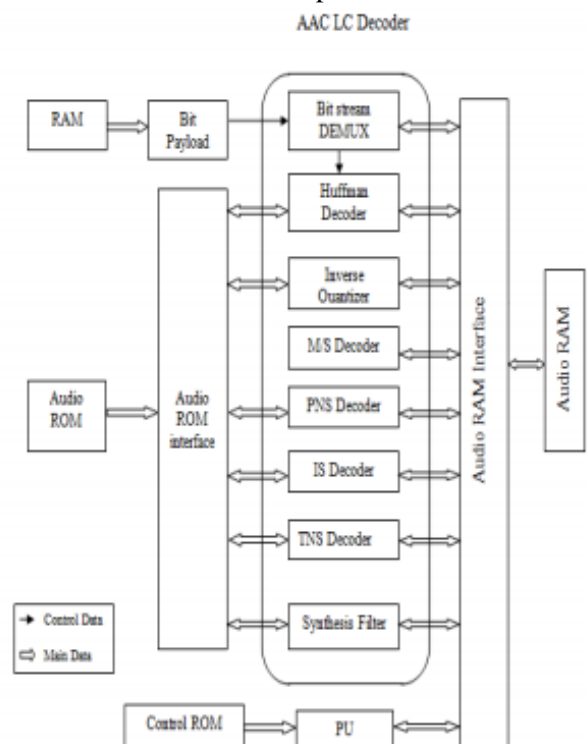

Fig-2: Block Diagram of AAC LC Decoder

RESULTS AND DISCUSSION

DAB audio decoder is developed using MATLAB R2010a software and also run the simulation by using Verilog code in XILINX ISE 12.1 software. Every audio file is first converted into wave file, then that wave file has been given as an input to the MATLAB code and processed, to produce the encoded output. This encoded output is given as input to the Verilog code by using Xilinx ISE 12.1 software in ISim. It produces a decoded data as a simulation result. This result can be verified by using MATLAB. The snapshots of the simulation results are shown below.
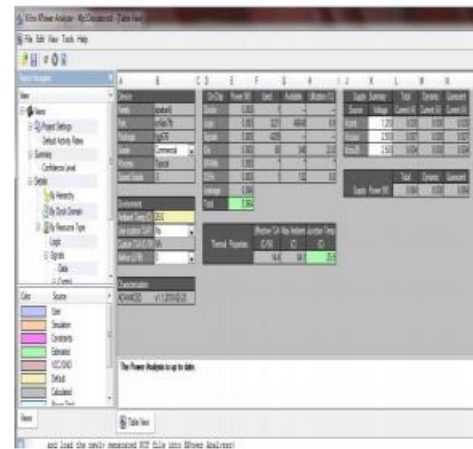
Table- 1: Comparison Results

Analysis	Existing system	Proposed system
Profile	AAC+ MP2	AAC LC + MP2
Architecture	ASIC	FPGA (SPARTAN 6)
Power Consumption of AAC Decoding for DAB Mechanism(mW)	227	64
Process (µm)	0.18 CMOS	FPGA (SPARTAN 6)
Area Utilization (%)	32%	30.7%

From Fig.3 an encoded output from MATLAB is given as a input to FIFO as 32-bit format in Verilog simulation and set positive edge clock, enable, Wfull, FIFO empty and run, get the output as decoded data in wdata [31:0] and that must be equal to the output using MATLAB.
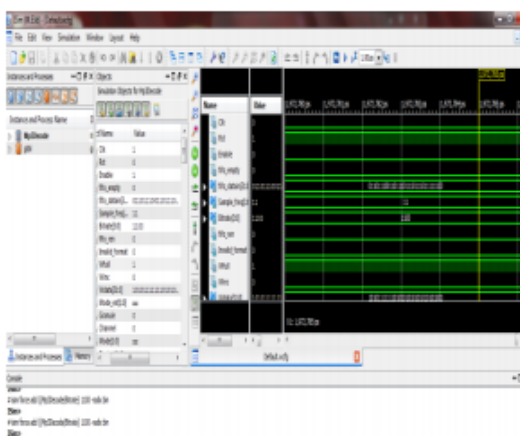

Fig- 3: Xilinx Simulation Result

From Fig. 4. Shows the power analysis report produce by Xilinx ISim after simulation process is completed. Power consumption of designed architecture is 64mW in FPGA Spartan6.

Fig- 4: Power analysis report

CONCLUSIONS

Thus the real time MP2 and AAC decoding system supporting LC profile for DAB audio decoder was designed. The output of this project shall be processed by many blocks before hear the audio services. To verify the designed system by using simulation. The result of simulation showed in both Xilinx ISE 12.1 ISim and MATLAB R2010a software provide the same decoded output. Then the power consumption and area occupied by the architecture have been analyzed and at the same time maintaining the high audio quality. Future work of this project is to implementing this designed architecture using FPGA SPARTAN6. And also measure the audio quality of AAC LC decoder by using PSNR.

REFERENCES:

[1] Guoyu Wang, Hongsheng Zhang, Mingying Lu, Chao Zhang, Tao Jiang and Guangyu Guo, „Low-cost Low-power ASIC solution for both DAB+ and DAB audio decoding‟ IEEE Transactions on VLSI systems, pp. 1-9, 2013.

[2]. Chao-Tang Yu, Kai-Sheng Yang and Yu-Pin Chang, „DAB Channel Decoder Implementation Using FPGA and Its Testing Platform Buildup‟, IEEE Transaction on VLSI system, pp. 424-429, 2005.

[3]. Herre J., „Temporal noise shaping, quantization and coding methods in perceptual audio coding‟ AES

17th International Conference High Quality Audio Coding, pp. 312–315, 1999.

[4]. Jae-Sik Lee, Jong-Hoon Jeong and Tae-Gyu Chang, „An Efficient method of Huffman decoding for MPEG-2 AAC and its performance analysis" IEEE Transactions on Speech and Audio Processing, vol. 13, No. 6, pp. 1206–1209, 2005.

[5]. Lu M. Zhang H. Wang G. Guo G. and JiangT., „Optimization and hardware implementation of the AAC inverse quantization algorithm for portable DAB+ receivers"

International Journal Digital Content Technology, vol. 6, No. 6, pp. 125–132, 2012. [6]. Shen-ChuanTai, Chuen-Ching Wang and Chih-Ying Lin, „FFT and IMDCT circuit sharing in DAB receiver" IEEE Transactions on Broadcasting, vol. 49, No. 2, pp. 124–131, 2003.

[7]. Tsung-HanTsai, Chun-Nan Liu, Hsueh-Yi Lin, HsingChuang Liu and Chia-Ying Wu, „A 1.4 MHz 0.21 mW MPEG-2/4 AAC single chip decoder" IEEE Custom Integrated Circuits Conference(CICC), pp.653–656, 2009.

[8]. Tsung-HanTsai, Chun-Nan Liu, „Low power system design for MPEG-2/4 AAC audio decoder using pure ASIC approach" IEEE Transactions on circuits and systems, vol. 56, pp. 144-155, 2009.

[9]. Yan M. Hu G. and Wang J. (2009) „Design and implementation of MPEG-4 AAC decoder on ARM embedded system for CMMB receiver" International Conference Manage. Service Sci., pp. 1–3.

[10]. L. Joohyun, K. Sungdo, and K. Jinkyu, "A 159.2 mW SoC implementation of T-DMB receiver including stacked memories," in Proc. IEEE Custom Integr. Circuits Conf., Sep. 2008, pp. 679–682.

[11].. H. Zhang, M. Lu, and G. Wang, "An ASIC implementation of MPEG audio decoders," in Proc. 7th Int. Conf. ASIC, Oct. 2007, pp. 754–757.

[12]. P. Liu, L. Liu, N. Deng, X. Fu, J. Liu, Q. Liu, G. Zhang, and B. He, "VLSI implementation for portable application oriented MPEG-4 audio codec," in Proc. IEEE Int. Symp. Circuits Syst., May 2007, pp. 777–780.

[13]. M. Lu, H. Zhang, G. Wang, T. Jiang, and G. Guo, "Optimization and hardware implementation of the AAC inverse quantization algorithm for portable DAB+ receivers," Int. J. Digit. Content Technol. Appl., vol.6, no. 6, pp. 125– 132, Apr. 2012.

[14]. M. C. Hans and V. Bhaskaran, "A compliant MPEG-1 layer II audio decoder with 16-B arithmetic operations," IEEE Signal Process. Lett, vol. 4, no. 5, pp. 121–122, May 1997.

[15]. D. Schulz, "Improving audio Codecs by noise substitution," J. Audio Eng. Soc., vol. 44, no. 7, pp. 593–598, Jul. 1996.

[16]. T.-H. Tsai and C.-N Liu, "Low-power system design for MPEG-2/4 AAC audio decoder using pure ASIC approach," IEEE Trans Circuits Syst I, Reg. Papers, vol. 56, no. 1, pp. 144–155, Jan. 2009.

[17]. K. Seong-Do and O. Seung-Hyeub, "A CMOS single chip receiver with RF front-end and baseband processor for band-III T-DMB/DAB applications," in Proc. 11th Int. Conf. Adv. Commun. Technol., pp. 1238–1241, 2009.

[18]. M. Yan, G. Hu, and J. Wang, "Design and implementation of MPEG-4 AAC decoder on ARM embedded system for CMMB receiver," in Proc. Int. Conf. Manage. Service Sci., 2009, pp. 1–3.

[19]. C. W. Ryu, D. H. Lee, H. J. Chi, K. S. Kwan, T. H. im, and J. S. Park, "Design of digital audio DSP core," in Proc. 1s